Jamming of LoRa PHY and Countermeasure

Ningning Hou, Xianjin Xia, Yuanqing Zheng
The Hong Kong Polytechnic University, Hong Kong, China
ningning.hou@connect.polyu.hk, {xianjin.xia, yqzheng}@polyu.edu.hk

***Abstract*—LoRaWAN forms a one-hop star topology where LoRa nodes send data via one-hop up-link transmission to a LoRa gateway. If the LoRa gateway can be jammed by attackers, the LoRa gateway may not be able to receive any data from any nodes in the network. Our empirical study shows that although LoRa physical layer (PHY) is robust and resilient by design, it is still vulnerable to synchronized jamming chirps. Potential protection solutions (*e.g.*, collision recovery, parallel decoding) may fail to extract LoRa packets if an attacker transmits synchronized jamming chirps at high power. To protect the LoRa PHY from such attacks, we propose a new protection method that can separate LoRa chirps from jamming chirps by leveraging their difference in the received signal strength in power domain. We note that the new protection solution is orthogonal to existing solutions which leverage the chirp misalignment in time domain or the frequency disparity in frequency domain. We conduct experiments with COTS LoRa nodes and software defined radios. The results show that synchronized jamming chirps at high power can jam all previous solutions, while our protection solution can effectively protect LoRa gateways from the jamming attacks.**

I. Introduction

Low-power wide-area networks such as LoRaWAN are emerging technologies that enable long-range low-power wireless communication for battery-powered sensor nodes [1]–[4]. A LoRa node is expected to transmit LoRa packets with the communication range upto 10 km using AA batteries for ten years and enables innovative applications [5]–[7] (*e.g.*, smart electricity metering, smart homes, supply chain, and health care).

LoRa adopts chirp spread spectrum (CSS) modulation in physical layer (PHY), which is known to be resilient and robust to interference and noise. Benefiting from the long communication range, LoRaWAN forms a one-hop star topology, where a large number of LoRa nodes can send packets via one-hop up-link transmissions to a LoRa gateway, which greatly simplifies the network protocol design and facilitates data collection. In such a star topology, however, if a LoRa gateway is jammed by malicious attackers, the LoRa gateway may not be able to receive LoRa packets from any nodes in the network, leading to single point of failure. Neighbor gateways could help receive the packets in this case, but those gateways can also be under jamming attacks.

We note that wireless jamming has been extensively studied in literature [8] and LoRa jamming has also been attracting attention in both academia and industry recently. Some previous works [9]–[11] have demonstrated that it is indeed possible to jam LoRa nodes to some extent by emitting various jamming signals, while other measurement studies [1], [12], [13] show that LoRa nodes are inherently resilient and robust to interference and can even support parallel transmissions by resolving collisions. To better understand the LoRa demodulation under jamming attacks, we conduct experiments with COTS LoRa nodes and software defined radios. Our empirical study indicates that jamming attacks (*e.g.*, random interference and jamming chirps) may not necessarily affect packet receptions at LoRa gateways, meaning that LoRa by design is resilient to a certain type of jamming attacks and intentional interference.

By conducting deep analysis, however, we notice that if the jamming chirps are well-aligned with LoRa chirps, LoRa gateways cannot extract the LoRa chirps from jamming chirps any more. As such, a malicious attacker can send synchronized chirps at high power to jam LoRa chirps, which leads to dramatic performance degradation of LoRa communication. We note that existing collision recovery solutions (*e.g.*, FTrack [12], mLoRa [14]) cannot resolve collisions caused by synchronized jamming chirps, since the LoRa chirps and the jamming chirps are aligned and thus cannot be separated in the time domain. Frequency domain collision recovery solutions (*e.g.*, Choir [13]) cannot help either since attackers can send the jamming chirps at the same frequency of LoRa chirps.

To further enhance the LoRa PHY against synchronized jamming chirps, we propose a new protection method that separates LoRa chirps from jamming chirps by leveraging their difference in signal strength. We note that the new protection method is orthogonal to existing solutions which leverage timing information (*e.g.*, chirp boundary misalignment) or frequency information (*e.g.*, frequency disparity). As such, our protection method can be integrated with existing collision recovery solutions and complement each other.

We implement our protection method and conduct experiments with COTS LoRa nodes as well as software defined radios. The experiment results show that well-synchronized jamming chirps at high transmission power can jam all previous solutions with very high success rates, while our protection method can effectively protect LoRa gateways from all known LoRa jamming attacks including synchronized jamming chirps.

The key contributions of this paper can be summarized as follows.

- We investigate the vulnerability of current LoRaWAN physical layer under jamming attacks. We expose the risk of LoRa gateways under the attack of synchronized jamming chirps, which could lead to single point of failure in LoRaWAN.
- We propose a new collision recovery method as a countermeasure against the attack of synchronized jamming chirps by leveraging the difference in signal strength of jamming chirps and LoRa chirps.
- We conduct comprehensive experiments with COTS LoRa nodes as well as software defined radios under various experiment settings. The experiment results demonstrate the effectiveness of our protection method against jamming attacks.

II. Background and System Model

A. LoRa PHY: Chirp Spread Spectrum

LoRa adopts Chirp Spread Spectrum (CSS) modulation in physical layer. In CSS, a chirp signal sweeps through

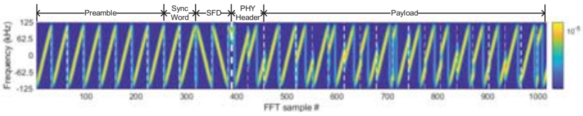


Fig. 1. **LoRa packet structure.**

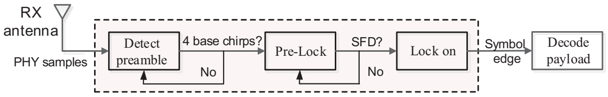


Fig. 2. **Locking process at LoRa receiver.**

and wraps around a predefined bandwidth with the instant frequency increasing (up-chirp) or decreasing (down-chirp) linearly at a constant rate over time. LoRa uses different initial frequencies of chirps to modulate symbols. An up-chirp with the initial frequency of $f_0 = -BW/2$ is named base chirp. LoRa uses N different initial frequencies to represent $SF = \log_2 N$ bits. Such a procedure can be represented as follows.

$$S(t, f_{sym}) = e^{j2\pi(\frac{k}{2}t+f_0)t} \cdot e^{j2\pi f_{sym}t} = C(t) \cdot e^{j2\pi f_{sym}t} \quad (1)$$

where f_{sym} denotes the initial frequency of the up-chirp (*i.e.*, encoded symbol). $C(t) = e^{j2\pi(\frac{k}{2}t+f_0)t}$ represents the raw chirp signal (termed *base chirp*); f_0 and k denote the initial frequency and increasing rate of the chirp, respectively.

B. LoRa Packet Structure

Fig. 1 shows the PHY samples of a LoRa packet collected with software defined radios (SDR). A LoRa packet starts wtih several identical up-chirps as preamble and 2 sync word symbols followed by 2.25 start frame delimiter (SFD) as illustrated in the figure. In explicit header mode, physical header and payload follow the SFD in a LoRa packet. LoRa packets can have a varied number of preamble (*e.g.*, >4 up-chirps), but sync word and SFD are mandatory. On receiving incoming LoRa signals, a LoRa receiver first detects the preamble and then detects the SFD to determine the starting point of physical header and payload.

C. LoRa Packet Detection and Demodulation

LoRa packet reception process involves several key steps as illustrated in Fig. 2. First, the LoRa receiver detects the arrivals of LoRa packets by detecting the preamble consisting of more than 4 up-chirps. The preamble detection can be performed by correlating the received PHY samples with an up-chirp generated locally at the LoRa receiver [15]. More than 4 consecutive spikes in correlation results indicate the arrival of one LoRa packet. One may also detect the preamble by tracking the continuity of frequency after multiplying the incoming PHY samples with down-chirps [12]. After successful preamble detection indicating the arrival of a LoRa packet, the LoRa receiver needs to accurately detect the SFD so as to determine the chirp boundaries of PHY header and payload. To this end, the LoRa receiver multiples the incoming PHY samples with an up-chirp and monitor continuous frequency for 2.25 chirp duration to determine the chirp boundary of the first chirp in PHY header and payload. After successfully locking on the chirp boundaries, the LoRa receiver can demodulate the chirps and decode incoming packets.

To demodulate a received chirp within a demodulation window, the LoRa receiver first multiplies the received signal with the *conjugate of the base chirp* denoted as $C^{-1}(t)$ and performs Fast Fourier Transform (FFT) on the multiplication results. After that, the LoRa receiver searches for the spike in the FFT bins (which indicates the initial frequency) and thereby demodulate symbol. The demodulation process is as follows

$$S(t, f_{sym}) \cdot C^{-1}(t) = e^{j2\pi f_{sym}t} \quad (2)$$

The FFT of $e^{j2\pi f_{sym}t}$ produces one spike in the FFT bins, indicating the initial frequency of f_{sym} [12].

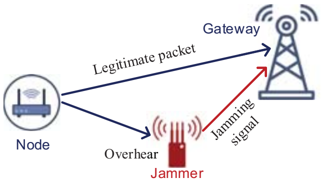


Fig. 3. **Attack model.**

D. System Model and Assumptions

Fig. 3 illustrates the jamming model, which consists of a LoRa node (which sends LoRa packets), a LoRa gateway (which receives LoRa packets), and a malicious jammer (which aims to jam the LoRa communication).

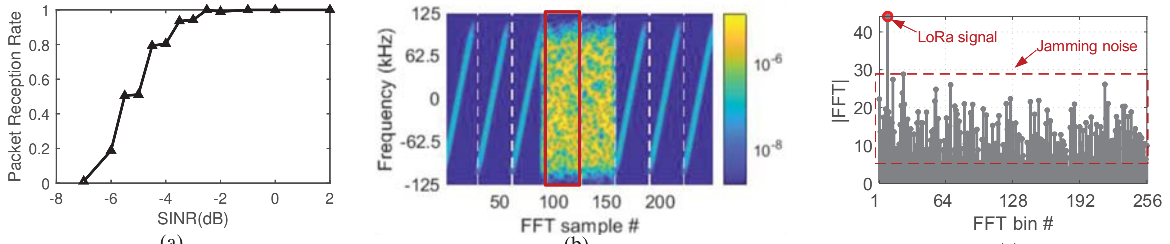
We assume that the LoRa gateway is equipped with software defined radio (SDR) to measure physical layer samples for collision recovery. We note that LoRa gateway can use low-cost receive-only SDR (*e.g.*, RTL-SDR dongle) since it only needs to receive rather than transmit radio signals. For the downlink from the LoRa gateway to the LoRa node, the gateway can use COTS LoRa modules for transmission.

We assume that the jammer is equipped with software defined radio (*e.g.*, USRP N210) for sniffing incoming LoRa packets and generating jamming radio signals accordingly. The jamming radio can be random Gaussian noise or LoRa signals. In LoRaWAN, LoRa nodes typically adopt low duty cycle mode (*e.g.*, 1% duty cycle). As such, if a jammer constantly emits jamming signals at high transmission power, the jammer can be easily detected and located. Therefore, we consider a jammer that adopts reactive jamming where the jammer stays quiet when the channel is idle, and starts emitting jamming signals when it detects on-going LoRa communication to selectively jam the LoRa communication. The objective of the jammer is to jam the communication between LoRa nodes and a LoRa gateway. We assume that the jammer aims to jam a specific gateway rather than all gateways in a network.

On the other hand, we want to design and implement countermeasure to protect the communication by enhancing the LoRa gateway against the jammer. Ideally, the countermeasure should not require any modification to the LoRa node to support a large number of already deployed COTS LoRa nodes.

III. Empirical Study of LoRa Jamming

LoRa jamming has been attracting wide attention due to the potential risk of single point failure under jamming attacks. Previous works [9]–[11] have demonstrated that it is indeed possible to jam LoRa nodes to some extent by emitting various jamming signals, while other measurement studies [1], [12], [13] show that LoRa nodes are inherently resilient and robust to interference and can even support parallel transmissions by resolving collisions. In the following, we conduct empirical study to evaluate the impact of a variety of prior jamming attacks to the LoRa communication.



(a) (b) (c)

Fig. 4. **Jamming with Gaussian noise.** (a) Packet Reception Rate of LoRa node under different SINR. (b) Spectrum of LoRa base chirps under Gaussian noise attack. (c) FFT after dechirp operation of chirp in red box in (b).

A. *Prior Jamming Attacks and Empirical Study*

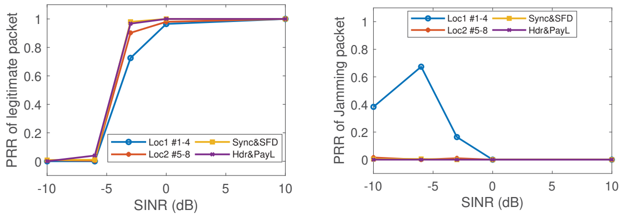
1) Jamming LoRa with Gaussian Noise: Gaussian noise has been commonly used to jam wireless communication systems. In the following, we test if LoRa communication can be jammed using Gaussian noise and evaluate the impact of Gaussian noise to LoRa communication. To this end, we use a software defined radio to emit Gaussian noise in the same frequency band as the LoRa communication. We vary the transmission power of the Gaussian noise jammer and measure the packet reception rate (PRR) under different Signal-to-Interference-plus-Noise Ratio (SINR). In the experiment, we keep the transmission power of LoRa node unchanged and both the LoRa node and the LoRa gateway remain static.

As shown in Fig. 4(a), we can observe that LoRa node can achieve almost 100% PRR even when SINR is $-2\ dB$ and it can still achieve almost 80% PRR when SINR decreases to $-4\ dB$. Intuitively, $0\ dB$ means that the signal strength of LoRa node is comparable with the interference and noise, while a negative SINR means that the received LoRa signal at the gateway is even weaker than the interference and noise.

The reason why LoRa can still receive packets even with negative SINR is that LoRa adopts CSS modulation, which is inherently robust to interference and noise. Fig. 4(b) plots the spectrum of LoRa chirps (preamble part) under the Gaussian noise attack. In Fig. 4(b), we see that the LoRa chirps are totally submerged by the Gaussian noise. Fortunately, if we apply the demodulation operation (*i.e.*, multiplying with down-chirp and FFT), we can still see a spike in the FFT bins corresponding to the correct initial frequency as shown in Fig. 4(c). That is because after the de-chirp operation, the power of Gaussian noise will still be distributed to all FFT bins, while the LoRa chirp will concentrate into one FFT bin corresponding to the initial frequency of the up-chirp.

As a matter of fact, the LoRa node can adopt a more conservative parameter setting (*e.g.*, spreading factor, bandwidth) to further enhance its robustness against interference and noise. If the jammer emits Gaussian noise at higher transmission power, it may cause performance degradation but the jammer can be detected due to the high transmission power, which is restricted by regulation. This experiment demonstrates that unlike other wireless technologies, LoRa PHY is inherently robust to Gaussian noise to some extent in practice.

2) Jamming LoRa with Chirps: Recent work [9] proposes to jam LoRa nodes with LoRa packets and cause collisions to legitimate LoRa communication. The prior work sets the maximum transmission power of jammer, while legitimate LoRa node may transmit at a lower transmission power to reduce power consumption. We evaluate the impact of jamming chirps to LoRa chirps in the collisions. The jamming LoRa packet is in the same packet structure as the legitimate LoRa packet as illustrated in Fig. 1. In this experiment, both legitimate transmitter and jammer are configured to use the same SF and bandwidth. We note that if they adopt different parameter settings, as LoRa gateways support parallel transmissions of LoRa packets with different parameter settings, legitimate packets can be received by gateways [1]. After setting the same parameters (*e.g.*, spreading factor, bandwidth, central frequency), we vary the transmission power of the jammer and evaluate the impact of jamming chirps under different SINR.



(a) (b)

Fig. 5. **Jamming packets collide with different parts of LoRa packets under different SINRs:** (a) Packet Reception Rate of legitimate packets and (b) Packet Reception Rate of jamming packets.

As we described in Section II-C, LoRa demodulation process involves several key steps including preamble detection, frame alignment, and chirp demodulation in demodulation windows. As such, we consider the following four scenarios where jamming chirps collide with the different parts of LoRa packets: 1) collision with the first four base chirps; 2) collision with the last four base chirps; 3) collision with sync word and SFD; and 4) collision with PHY header and payload. Fig. 5 shows the experiment results, from which we have the following key observations.

First, to jam LoRa signal with COTS LoRa node, the power of jamming packets need to be orders of magnitude higher than that of legitimate LoRa packets (*e.g.*, SINR $\leq -3\ dB$). If the received signal strength is comparable with the jamming signal (*e.g.*, SINR $\geq 0\ dB$), the legitimate packets can still be received correctly with high PRR (*e.g.*, $\geq 96.5\%$).

Second, the LoRa receiver is not likely to receive the late coming jamming packets. That is because the LoRa receiver is more likely to detect and lock on the preamble and SFD of the legitimate packets that arrive earlier than the jamming packets. Yet, we do observe the capture effect where jamming packets colliding at the first four base chirps of legitimate packet with strong signal strength are selected and demodulated (*e.g.*, SINR $\leq -3\ dB$).

Third, the impact of collision at PHY header and payload seems weaker than the collision at the preamble. Referring to Fig. 6, let us see how the collision at the PHY header and payload part would influence the demodulation of legitimate chirps in demodulation windows. Suppose the legitimate chirp

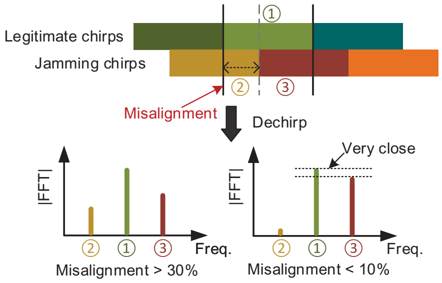


Fig. 6. **Demodulation example:** Chirps misaligned with demodulation window will have part of its power split out.

(①) collides with jamming chirps (② and ③) as illustrated in the figure. In the demodulation of PHY header and payload, a LoRa receiver multiplies the PHY samples in the demodulation window with a down-chirp, and performs the FFT on the multiplication result. Due to the collision of jamming chirps in the demodulation window, the FFT operation will generate three spikes as illustrated in the figure. Because the jamming chirps are misaligned with the legitimate chirp, the power of jamming chirps will be divided into two demodulation windows and their corresponding spikes would be lower than that of legitimate one. As such, we see that LoRa nodes can tolerate collisions at PHY header and payload with jamming chirps with comparable or even slightly stronger signal strength. However, if the jamming chirps and the legitimate chirps are well aligned (*e.g.*, $< 10\%$ misalignment), the spikes of jamming chirps within the demodulation windows could become higher than those of legitimate chirps. In this case, the legitimate nodes will be jammed, since the LoRa receiver demodulates the jamming chirps but not the legitimate chirps.

B. Prior Collision Recovery Methods as Countermeasures

We can draw strength from the recent advances in LoRa collision recovery and parallel transmissions to protect LoRa communication against jamming attacks. For example, recent works show that some LoRa collisions can be resolved by separating the LoRa chirps of different LoRa nodes in the time domain [12], [13], [16]–[18] and in the frequency domain [13].

For example, LoRa collision recovery schemes (*e.g.*, FTrack [12], mLoRa [14]) can resolve the collisions of multiple LoRa nodes as long as their chirp boundaries are misaligned in the time domain. FTrack [12] detects the continuity of chirps within a demodulation window to recover collisions. Referring to Fig. 6, we see the frequency of legitimate chirp continuously increases while the frequency of jamming chirps are not continuous within the demodulation window due to the chirp boundary misalignment. If the jamming chirps and legitimate chirps are well-aligned in the time domain, the FFT spikes of jamming chirps and legitimate chirps will be very close to each other. In this case, if the jamming chirps are slightly stronger than legitimate chirps, those collision recovery schemes cannot resolve the collisions.

Frequency domain collision recovery schemes (*e.g.*, Choir [13]) separate LoRa collisions by leveraging the frequency differences of colliding nodes due to their hardware imperfection. For example, Choir [13] notices that the fractional part of initial frequencies of different LoRa nodes are unique, which can be used as physical layer fingerprints. As such, Choir can group different chirps according to fractional parts and thereby separate colliding LoRa chirps. If the frequency of jamming chirps are synchronized with the legitimate chirps (*e.g.*, emitting jamming chirps with the same fractional part of initial frequency), those collision recovery schemes cannot separate the legitimate chirps from jamming chirps.

In summary, prior collision recovery methods cannot separate legitimate LoRa chirps from jamming chirps if the jamming chirps are aligned with the legitimate chirps in the time domain and the frequency domain. In this case, if the power of a jamming chirp is higher than that of a legitimate chirp, LoRa receivers will demodulate jamming chirps within demodulation windows rather than legitimate chirps.

IV. Defeating Prior Countermeasures with Synchronized Jamming Chirps

As we described in Section III-B, in order to attack a legitimate LoRa node, an attacker needs to emit jamming chirps that satisfy the following three conditions. Otherwise, prior countermeasures can protect the legitimate LoRa node by separating legitimate chirps from jamming chirps.

A. Necessary Conditions of Jamming against Prior Countermeasures

C-1: Jamming chirps should be well-aligned with legitimate LoRa chirps in time domain. Prior collision recovery and parallel decoding methods (*e.g.*, FTrack [12], mLoRa [14]) separate LoRa collisions in time domain. As such, if jamming chirps are not aligned with legitimate chirps, the jamming chirps can be separated in the time domain.

C-2: Jamming chirps should mimic legitimate LoRa chirps in frequency domain (e.g., central frequency). Frequency domain collision recovery schemes (*e.g.*, Choir [13]) separate LoRa collisions by leveraging the frequency differences of colliding nodes. To jam a LoRa node protected by the frequency domain collision recovery schemes, a jammer needs to synchronize the jamming chirps in frequency domain with the LoRa node.

C-3: Jamming chirps should have a higher power than legitimate LoRa chirps at a LoRa receiver. If the power of a jamming chirp is weaker than that of a legitimate chirp, the LoRa receiver can correctly detect the initial frequency of the legitimate chirp.

We note *C2* (*i.e.*, frequency condition) and *C3* (*i.e.*, power condition) are relatively easy to satisfy. For example, a jammer can measure the frequency of a legitimate preamble and extract the fractional part of frequency. After that, the jammer can emit jamming chirps with the same fractional part, which can defeat the frequency domain collision recovery scheme (*e.g.*, Choir [13]). To increase the power of jamming chirps at receiver, a jammer can increase the transmission power and get closer to the LoRa receiver.

However, *C1* (*i.e.*, timing condition) can be a bit challenging to satisfy because of signal processing delay caused by software defined radios, different communication distance between the LoRa node and the LoRa receiver, *etc*. As such, jamming chirps may not be well-aligned with legitimate chirps in the time domain. In this case, the power of jamming chirps will be divided into two adjacent demodulation windows. Moreover, the time domain collision recovery schemes can separate the legitimate chirps from the misaligned jamming chirps.

B. Jamming with Synchronized Chirps

We illustrate the basic jamming workflow as shown in Fig. 7. A LoRa jammer hears LoRa packets over the air. Upon detecting a valid LoRa preamble, it will attempt to lock on

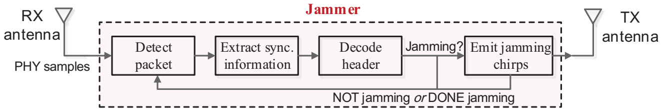


Fig. 7. **The general workflow of LoRa jammer.**

the packet by extracting synchronization information. After that, it can identify and interpret the packet header like a normal receiver. If the packet is transmitted by a targeted node, the jammer will emit synchronized chirps to jam the legitimate packet. Specifically, to launch an effective jamming with well-synchronized chirps, the jammer needs to take all time/frequency offsets (*i.e.*, jamming conditions) into account and carefully compensate them before sending jamming chirps in real time. We present the key steps to generate synchronized chirp jamming in the following.

1) Accounting for propagation delay: Basically, the emitted jamming chirps are required to closely align with the chirps of a legitimate packet when received at a gateway. The communication distance between jammer and gateway and the corresponding propagation delay affects the arrival time of jamming chirps at the gateway. We notice that as LoRa typically adopts narrow bandwidths (*i.e.*, $\leq$ 500 kHz), the sampling interval of LoRa receiver is relatively large (*e.g.*, $>$ 2 μs). The signals arrived within 2 μs (which corresponds to a communication distance of 600 m) are aligned to the same PHY sample. In practice, the jammer can emit jamming chirps within 600 m away from the gateway to mitigate the influence of propagation delay.

2) Compensating carrier frequency offset (CFO): When jammer hears the preamble of a legitimate packet, it detects chirp boundaries from the preamble and aligns jamming chirps to legitimate chirps. Intuitively, the jammer can detect chirp edges by correlating the received preamble that is composed of successive base chirps with a locally generated base-chirp. However, the detected edges may not correspond to the correct chirp edges due to carrier frequency offset between the legitimate node and the jammer. As a result, the frequency offset translates into corresponding time offset for chirp signals [12], [16]. To be specific, let Δf_{cfo} denote the CFO. The received preamble chirps can be represented as

$$R_{pre}(t) = h \cdot e^{-j\Delta f_{cfo}t} \cdot C(t) \tag{3}$$

where $C(t)$ denotes the base up-chirp of preamble transmitted by the legitimate node, and h is the channel between the node and the jammer. If we directly correlate $R_{pre}(t)$ with a local base chirp $C(t)$, the detected chirp edge would be $\Delta t = \frac{2^{SF}}{BW^2}\Delta f_{cfo}$ away from the real edge, as illustrated in Fig. 8(a). According to our measurements, this edge offset Δt can be as large as ten samples in practice. As such, a jammer must compensate the timing offset caused by CFO and align jamming chirps to correct edges.

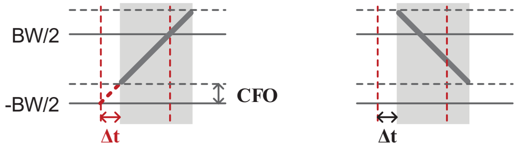
Firstly, a jammer needs to estimate CFO from the received signal. We exploit the SFD that comes after preamble (see Fig. 1) for CFO estimation. In particular, a received SFD chirp can be represented as:

$$R_{sfd}(t) = h \cdot e^{-j\Delta f_{cfo}t} \cdot C^{-1}(t) \tag{4}$$

By multiplying Eq. (3) with Eq. (4), we obtain

$$R_{pre}(t) \cdot R_{sfd}(t) = h^2 \cdot e^{-j2\Delta f_{cfo}t} \tag{5}$$

We perform FFT (Fast Fourier Transform) on Eq. (5) and the resulting FFT peak indicates the value of Δf_{cfo}. We use Δf_{cfo} to compute the corresponding chirp edge offset $\Delta t = \frac{2^{SF}}{BW^2}\Delta f_{cfo}$, which is finally used to infer the correct chirp edge from detected edges.



(a) (b)

Fig. 8. **CFO affects edge detection:** (a)Detected edge vs. real edge of base up-chirp in preamble. (b)Extracted SFD down-chirp with edge offset Δt.

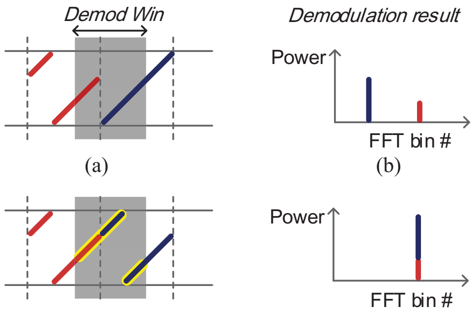
As we may detect incorrect chirp boundaries from the received preamble due to CFO, one may wonder how to extract the correct preamble chirp and SFD chirp for CFO estimation. As a matter of fact, we can first perform correlation detection on the received preamble to coarsely detect the boundary timing of chirps with a time offset, as illustrated by red dashed lines in Fig. 8. We use the coarsely detected timing to identify SFD chirps. We note that the extracted preamble base-chirp and SFD down-chirp have the same offset (*i.e.*, Δt) with their real edge timing, as illustrated in Fig. 8. As a result, the extracted chirps in Eq. (5) for CFO estimation are actually $R_{pre}(t-\Delta t)$ and $R_{sfd}(t-\Delta t)$, rather than the ideal $R_{pre}(t)$ and $R_{sfd}(t)$. As the edge time offset (Δt) translate into frequency offset Δf_{edge} for the up-chirp and an opposite frequency $-\Delta f_{edge}$ for the down-chirp, we have $R_{pre}(t-\Delta t) \cdot R_{sfd}(t-\Delta t) = R_{pre}(t) \cdot R_{sfd}(t)$. In summary, the above CFO estimation method (*i.e.*, Eq .(5)) still holds with the time offset in preamble and SFD detection.

3) Compensating hardware and software delay: The jammer also needs to process received signal and react in real time. It imposes a strict constraint on the processing latency (termed *jamming delay*). We use a software defined radio (*i.e.*, USRP N210) as hardware and use the open-source GNU Radio (GR) as software to perform jamming on-line. In particular, we list the main contributors of jamming delay as follows.

- Data transfer: The delay of data transfers between different components, *e.g.*, from USRP Rx buffer to data processing blocks as well as from blocks to USRP Tx when emitting jamming chirps.
- Scheduling: The latency of OS (*i.e.*, operating system) and GR scheduling.
- Signal processing: The latency of signal processing including preamble detection, packet decoding, synchronization of jamming chirps, *etc*.

We note that as signal processing is generally performed on PCs with powerful CPUs, the processing latency is relatively short (*e.g.*, tens of μs on our Intel i5 PC). In comparison, the air time of LoRa packet is of 2 $\sim$ 3 orders of magnitude longer. For instance, the transmission time of a typical LoRa chirp with $SF = 8, BW = 250$ kHz is about 1 ms (*i.e.*, 100× longer than signal processing). Theoretically, this would leave a sufficient amount of time for a jammer to finish signal processing and generate jamming chirps in real time.

On the other hand, we empirically observed that the GR scheduling and data transfers exhibit time uncertainty in practice. The latency varies randomly from 100 μs to $10,000$ μs in our measurements. We configure the GR scheduler with a Single-Thread-Scheduler mode (*i.e.*, STS) to reduce the processing latency and time variation. We also configure



(a) (b)

(c) (d)

Fig. 9. **Jamming without synchronization:** (a-b) Non-identical jamming chirps and demodulation result vs. (c-d) Identical jamming chirps and the demodulation result. When consecutive jamming chirps are identical, the samples from adjacent chirps form a complete chirp in the demodulation window which well-aligns with legitimate chirp.

the buffer size of inter-block data transfer to fit the size of LoRa chirps. As a result, the end-to-end jamming latency becomes rather stable (*e.g.*, 500 μs in our setting), which can be measured and compensated before sending jamming chirps.

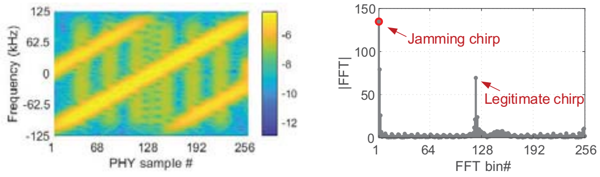
In order to align a jamming chirp with a legitimate chirp, the jammer needs to infer which sample is currently transmitting in the air (*i.e.*, the front wave of legitimate packet). To this end, the jammer continuously receives samples of legitimate packet using USRP, which buffers the received samples and reports them when the buffer is full. In practice, the number of reported samples in every buffer and the corresponding timestamp can vary due to the uncertainly in GR scheduling. To address this problem, the jammer can estimate the current transmitting sample in the air with the latest received buffer size and its timestamp. By further counting in the processing latency, the jammer can determine the time compensation for precise alignment of jamming chirps with legitimate chirps.

C. Jamming with identical Consecutive Chirps

The synchronized jamming approach satisfies all conditions listed in Section IV-A. A jammer can properly choose jamming chirps to mimic the payload of a legitimate packet, and employ synchronized jamming to defeat the existing collision recovery strategies. However, the synchronized jamming approach requires careful calibration and strict timing requirement to align jamming chirps with legitimate chirps. In the following, we demonstrate that it is possible to jam in a lightweight manner without strict synchronization (*e.g.*, delay compensation).

If we perform jamming without synchronization, the emitted jamming chirps are likely to misalign with chirps of legitimate packet. Suppose a gateway uses a time domain collision recovery scheme to protect legitimate packets from jamming attacks. Let us consider a demodulation window that is aligned with a legitimate chirp but not jamming chirps. As illustrated in Fig. 9(a), since the demodulation window spans across two adjacent jamming chirps, jamming signals within this demodulation window would experience a sudden change in frequency at chirp boundary. As a result, after demodulation, there will be two FFT spikes at different FFT bins (Fig. 9(b)).

However, if the two adjacent jamming chirps are the same, their frequency would experience no sudden change at the jamming chirp boundary (see Fig. 9(c)). As a result, both the jamming chirp and the legitimate chirp exhibit frequency continuity within the demodulation window, meaning that the power of consecutive jamming chirps will concentrate in the demodulation window, as if one jamming chirp is well-aligned with the window, as illustrated in Fig. 9(c) and (d). As such, a jammer can emit the same consecutive chirps to defeat existing countermeasures without synchronizing to legitimate chirps.



(a) (b)

Fig. 10. **Jamming power is higher than the power of legitimate packet:** (a)Received signal power of jamming chirp vs. legitimate chirp. (b)FFT magnitude of demodulated jamming chirp vs. legitimate chirp.

However, COTS LoRa radio interleaves the payload data to avoid successive identical symbols in the PHY layer. While we can observe two consecutive chirps with the same initial frequency in practice, we seldom observe more than three identical symbols appearing successively in the payload of packets transmitted by COTS LoRa nodes. As such, a jammer can emit two consecutive chirps with the same initial frequency as jamming chirps.

We note that the consecutive chirp pattern still differs from the random chirp pattern of a normal packet payload. Existing time domain collision recovery schemes can be adapted to discern a consecutive jamming attack by detecting chirp's consecutive patterns. As a result, the consecutive jamming approach may not be as effective as the synchronized jamming approach against existing countermeasures. Note that there can be other variations of jamming methods. For example, a jammer can transmit consecutive SFD chirps to interfere the locking process at a receiver. Similar to identical consecutive chirps, this method can be easily detected though.

V. Countermeasure

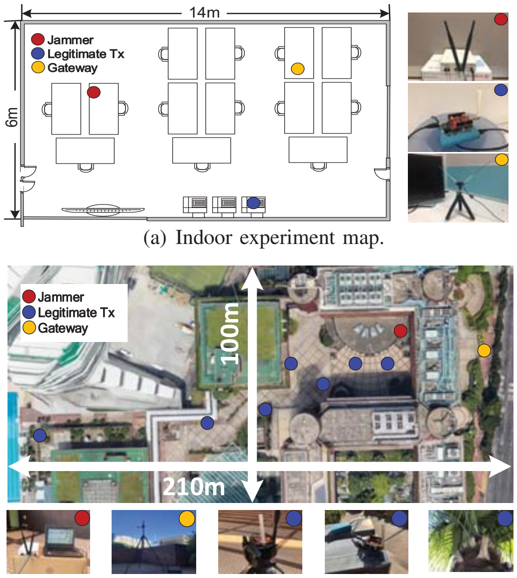
In the previous section, we reveal that current LoRaWAN suffers the risk of synchronized jamming attacks. In this section, we present a new countermeasure to protect LoRaWAN against synchronized jamming attacks.

Recall from the jamming conditions in Section IV-A (*i.e.*, *C-3*), in order to successfully jam a LoRa packet, it requires the power of a jamming chirp to be higher than the power of a legitimate chirp in a demodulation window, as illustrated in Fig. 10(a). Essentially, we can expect a discrepancy of FFT magnitude between the jamming chirp and the legitimate chirp after demodulation, as shown in Fig. 10(b). This motivates us to differentiate a legitimate chirp from a jamming chirp by checking their received signal strength in *power domain*, which complement the conventional collision recovery schemes examining time and frequency domain.

The received signal strength (*i.e.*, RSS) of LoRa packet can be affected by many factors (*e.g.*, transmit power, communication distance, receiver gain, *etc.*), but most of those factors are generally invariant during the transmission of a packet. For instance, the transmit power of a LoRa node can be adapted for each packet transmission, but will remain the same during the packet transmission. Besides, in our target scenarios, LoRa nodes generally remain stationary or move at a low speed. More importantly, since the LoRa PHY (*i.e.*, CSS) does not modulate the amplitude of LoRa chirps, the power level of LoRa chirps from the same packet would remain pretty stable and share high similarity. In addition, as a selective jammer starts jamming after interpreting the header of a legitimate packet, it leaves the packet preamble intact. As such, a receiver (*i.e.*, gateway) can measure the RSS from the preamble of a legitimate packet and use the measured RSS to help extract legitimate chirps from jamming chirps.

Finally, we present a RSS-based LoRa decoder as a countermeasure to the synchronized jamming attack. The decoding process generally works as follows. A receiver first detects the preamble of LoRa packet. In addition to extracting symbol timing (*i.e.*, chirp edges) from preamble as in a standard LoRa decoder, we also measure the RSS of preamble chirps. We then employ the same method of a standard decoder to locate and demodulate symbol chirps in the payload. In each demodulation window, we can obtain the interleaved FFT results of demodulated legitimate and jamming chirps, as shown in Fig. 10(b). Different from a standard decoder that selects the highest FFT peak as demodulation result, we pick the FFT peak with a magnitude that can best match the RSS measured from preamble as the demodulation result of legitimate chirp. We iteratively apply this method to demodulate all legitimate chirps and feed demodulated symbols into a standard decoder to produce the payload data of legitimate packet.

We note that if the RSS of jamming chirps and the RSS of legitimate chirps are very close, our RSS-based protection method alone cannot separate the legitimate chirps from the jamming chirps. In practice, it can be very challenging for a jammer to tune the transmission power of jamming chirps so that the RSS of jamming chirps can be received by a LoRa gateway at the similar RSS of legitimate chirps. Note that there is no feedback to the jammer from either the legitimate LoRa node or the LoRa gateway. Besides, in case of transmission failure because of jamming attack, a LoRa node would retransmit at different transmission power. Since our RSS-based protection method is orthogonal to the existing collision recovery methods which leverage the time and frequency domain information, those existing methods can be used in parallel to enhance protection method.



(a) Indoor experiment map.

(b) Outdoor experiment map.

Fig. 11. **Experiment layout.**

VI. Implementation and Evaluation

A. Implementation and Setup

We implement the jamming attack and corresponding countermeasure in real-world. We conduct experiment and evaluation in both indoor and outdoor environment. Specifically, as shown in Fig. 11, the indoor test bed spans $14 \times 6\ m^2$ and it is a typical office room with rich multipaths. The outdoor test bed spans $210 \times 100\ m^2$ and it is an urban outdoor environment with many skyscrapers. We use a COTS LoRa node (*i.e.*, LoRa shield, which consist of HopeRF's RFM96W transceiver module embedded with the Semtech SX1276 chip) as the legitimate transmitter and put it at different places (blue dots in Fig. 11(a) and Fig. 11(b)). A low-cost receive-only RTL-SDR dongle (*i.e.*, yellow dot) is used as the LoRa gateway to record the PHY samples. We implement the jamming process on a USRP N210 to work as a jammer (*i.e.*, red dot). For performance evaluation, we develop the standard LoRa demodulator and our own countermeasure in MATLAB to process PHY samples received by RTL-SDR dongle. All devices work at 915 MHz band. If not specified, we configure the spreading factor, code rate, and bandwidth of the LoRa chirp signal to 8, $4/8$, and 250 KHz, respectively.

To evaluate the impact of jamming attack and the effectiveness of our countermeasure, we implement the following two schemes: **1) Victim**: Legitimate LoRa communication (uses standard LoRa demodulation) under jamming attack, which is used to evaluate the impact of jamming attack; and **2) Protege:** The victim protected by our countermeasure against jamming, which is used to demonstrate the effectiveness of our countermeasure.

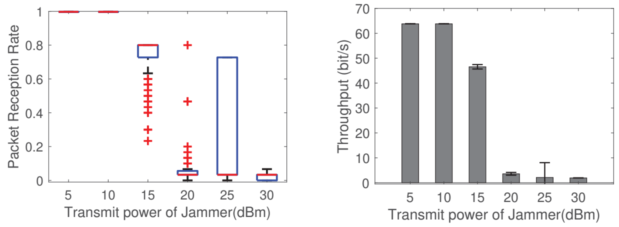
We use the following metrics to evaluate the performance. **1) PRR:** Packet Reception Rate (PRR) is the ratio of correctly received packets over the transmitted legitimate packets. **2) SER:** Symbol Error Rate (SER) is the ratio of incorrectly demodulated symbols; and **3) Throughput:** It quantifies the successfully received bits per unit time. We also compare our countermeasure with FTrack [12] and Choir [13] against jamming attack.

B. Basic Performance

1) Impact of Jamming Attack: In this experiment, the legitimate transmitter sends LoRa packets every 2 seconds. The payload length of each packet is set to 30 with the lowest transmission power of (5 dBm) in indoor environment. We keep the three players (*i.e.*, in Fig. 11(a)) static and vary the transmission power of jammer from 5 dBm to 30 dBm. In each scenario, we conduct over 120 measurements.

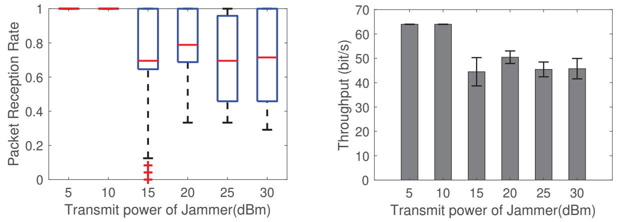
Fig. 12 shows the victim's PPR and throughput under jamming with different transmission power. We observe that when the jamming power is relatively small ($5 \sim 10\ dBm$), the PRR of Victim is almost 100%, meaning that the jamming attack has no impact on the LoRa communication due to its low jamming power. With further increase of jamming power (15 dBm), victim's PRR begins to decrease rapidly. When jamming power is 20 dBm or higher, the PRR decreases and almost all packets will be jammed by the attacker. Accordingly, the throughput of victim drops drastically when the jamming power is $20 \sim 30\ dBm$. This result reveals that the LoRa communication is vulnerable to synchronized jamming attack with a relatively high transmission power and the performance of LoRa communication can be substantially affected.

2) Performance of Countermeasure: In this experiment, we evaluate the performance of countermeasure. We use the same



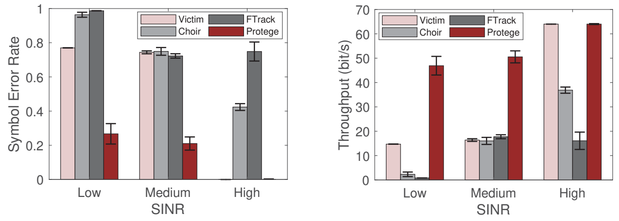
(a) (b)

Fig. 12. **Jammer performance with different transmission power. Victim's (a) PPR and (b) Throughput.**



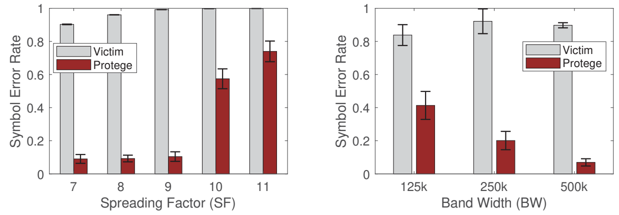
(a) (b)

Fig. 13. **Countermeasure performance with different transmission power. Protege's (a) PRR and (b) Throughput.**



(a) (b)

Fig. 14. **Performance comparison of Victim, Choir, FTrack, and Protege under different SINRs: (a) Symbol Error Rate (SER) and (b) Throughput.**



(a) (b)

Fig. 15. **Impact of (a) SF and (b) BW on Symbol Error Rate (SER) of Victim and Protege.**

setting as in subsection VI-B1. Fig. 13 presents the results. The average PRR and throughput of protege is higher than 70% across all transmission power of jammer. In comparison with Fig. 12, the overall PRR and throughput of protege are much higher than those of victim, especially when jamming power is higher than 15 dBm. The throughput of protege is 20× higher than that of victim when jamming power is 25 dBm, and 23× when jamming power is 30 dBm. This is because when transmission power is higher than 15 dBm, the SINR at receiver is low ($-10 \sim -5\ dB$). In this case, the power of legitimate chirps is weaker than that of jamming chirps, leading to incorrect demodulation results of victim. In contrast, our countermeasure can leverage the difference in received signal strength and separate the legitimate chirps from the jamming chirps. The experiment results indicate that our countermeasure can protect the LoRa gateway against such synchronized jamming attacks.

3) Comparison with Existing Countermeasures: In the following, we compare victim and protege with two typical collision recovery and parallel decoding methods, *i.e.*, FTrack and Choir. We compare these four methods in low ($-10 \sim -5\ dB$), medium ($-5 \sim\ 5\ dB$), and high ($5 \sim 10\ dB$) SINR scenarios. Each scenario includes over 120 measurements.

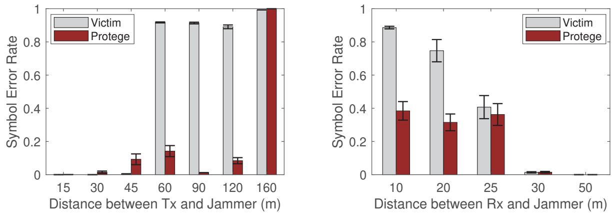
We plot the SER and throughput in Fig. 14. We observe that victim, Choir and protege have lower SER as SINR becomes higher. However, FTrack has over 72% SER in all scenarios. This is because FTrack distinguishes colliding chirps by using frequency tracks caused by time misalignment of two chirps. However, jammer in this paper synchronizes jamming chirps with legitimate chirps, making it hard for collision recovery method which uses timing information to separate. Since Choir disentangles colliding chirps by leveraging the disparity in frequency domain, higher signal strength benefits its performance. We can also see that protege has best performance in terms of SER and throughput in all SINR scenarios. Specifically, in low SINR scenario, our countermeasure (*i.e.*, protege) only has 26% SER in low SINR scenario while FTrack and Choir have SER of 96.38% and 98.7% respectively, even higher than that of victim (77%) using standard LoRa demodulation. In high SINR scenario, Choir and FTrack still have very high SER and low throughput, while protege and victim have almost 0 SER and 100% throughput. This experiment demonstrates that our RSS-assisted countermeasure outperforms all existing countermeasures.

C. Impact of LoRa Configuration

In this subsection, we examine the impact of LoRa packet configuration on the performance of jamming attack and our countermeasure strategy. We adopt the same experiment settings as in Section VI-B. Due to page limit, we only present the results of high jamming power ($\geq 20\ dBm$).

We first study the impact of LoRa spreading factor (SF). In this experiment, we fix the bandwidth to 250 kHz and vary SF from 7 to 11. We compare the PHY layer symbol error rates of standard demodulation method (*i.e.*, victim) and our countermeasure strategy (*i.e.*, protege) in Fig. 15(a). As expected, the SER of victim stays at high level (*e.g.*, $> 90\%$) for all SFs due to the high jamming power. In contrast, the protege can decode legitimate packets with SER lower than 10% when $SF = 7 \sim 9$. We observe that the SER of protege increases dramatically to higher than 60% as SF increases to 10 and 11. This is because the frequency gap between LoRa symbols becomes narrower as SF increases. As such, a larger SF will generally make the demodulation more vulnerable to jamming attacks.

We next evaluate the SERs of victim and protege in various bandwidth (BW) of LoRa packet. We set $SF = 8$ and change BW from 125 kHz to 500 kHz in the experiment. In Fig. 15(b), we see that as compared to the high SERs of standard demodulation method (*i.e.*, victim), our countermeasure strategy can correctly demodulate legitimate chirps with $SER < 20\%$ when $BW \geq 250\ kHz$. This is because wider bandwidth corresponds to larger frequency gap between LoRa symbols. As such, a wider bandwidth will generally make the demodulation more robust to jamming attacks.



(a)

(b)

Fig. 16. **Impact of (a) Distance between Tx and Jammer and (b) Distance between Rx and Jammer on SER of Victim and Protege.**

D. Impact of Jamming Distance

We perform testbed experiments in an outdoor environment as shown in Fig. 11(b). Unless otherwise specified, we adopt the default LoRa packet configuration of $SF = 8$, $BW = 250\ kHz$. In the first experiment, we place the jammer at a fixed distance ($15\ m$) to the gateway and keep them static. We place the legitimate LoRa node at different locations to evaluate the effective jamming range. The transmit power of jammer is fixed to $20\ dBm$. We configure the legitimate node with the maximum transmit power (*i.e.*, $23\ dBm$) and change node locations with distances of $15 \sim 160\ m$ to the gateway. We present the SER results of standard demodulation method (*i.e.*, victim) and our countermeasure strategy (*i.e.*, protege) in Fig. 16(a). We observe that both victim and protege can correctly demodulate legitimate packets when the node is within $45\ m$ from the gateway because of the high SINR of packets (*i.e.*, higher signal power than jamming power). When the distance is between $60 \sim 120\ m$, the SER of victim increases dramatically $\geq 80\%$, because the signal power of legitimate packets falls below the jammer power. Protected by our countermeasure, protege can still correctly demodulate packets when the distance is between $60 \sim 120\ m$. When the distance further increases to $160\ m$, the received signal strength of legitimate packets becomes too weak, leading to almost 100% SERs for both strategies.

In the second experiment, we keep the gateway and the legitimate node at fixed locations and move the jammer to evaluate the jamming performance at different distances. The legitimate node transmits with the maximum power (*i.e.*, $23\ dBm$). The distance between the legitimate node and the gateway is $20\ m$. We configure the jammer with the TX power gain of $80\ dB$. In Fig. 16(b), we find that the SER of standard demodulation method (*i.e.*, victim) decreases with jamming distance, because the power strength of jamming chirps become weaker as the distance increases. The SERs of protege are higher than 30% when jamming distance $\leq 25\ m$, because of the comparable signal power between jamming chirps and legitimate chirps. When the distance $\geq 30\ m$, both victim and protege can correctly demodulate the symbols since the power of jamming chirps becomes too weak in this range.

In summary, when a jammer is very close to the gateway, victim's performance will be dramatically affected by the jammer. With our countermeasure, protege can still demodulate some of the symbols correctly. Note that the LoRa PHY adopts error correction code to correct symbol errors in practice.

VII. Related Work

A variety of LPWAN technologies such as SigFox [19], NB-IoT [20], LTE-M [21] and LoRa [22] have been proposed to support wide area network connection for IoT devices. Prior efforts [1], [23]–[27] are devoted to the measurement study and performance analysis of LoRa, such as packet air-time [28], [29], power consumption [1], [30], coverage [31], [32], PHY security [33], *etc*. Based on these measurements, many strategies [4], [34]–[40] have been proposed to optimize LoRa communications and applications.

Wireless jamming has been extensively studied in literature [41]–[43]. Recent works study the impact of jammer to LoRaWAN and propose countermeasures. LoRaTS [44] studies the attack-aware data timestamping in LoRaWAN, which can protect LoRaWAN against frame delay attack. Aras *et al.* [45] identify a few security vulnerabilities of LoRa including encryption key extraction, jamming attacks, and replay attacks. Aras *et al.* [9] use commodity LoRa nodes as jammers to selectively jam LoRa packets. Previous collision recovery and parallel decoding schemes can help mitigate the impact of those jammers. Unlike those works, we study the impact of synchronized jamming chirps and propose countermeasure to protect against such new jamming attacks.

Collision recovery and parallel demodulation schemes can be used to solve collisions of LoRa chirps and jamming signals and thereby protect LoRa communication under jamming attacks. Choir [13] differentiates LoRa chirps by examining the frequency differences between different LoRa nodes. Choir groups different chirps according to different frequencies and separates colliding LoRa chirps. Other works leverage the misalignment of colliding packets in the time domain to separate colliding chirps. FTrack [12] detects the continuity of chirps within a demodulation window to recover collisions. mLoRa [14] derives the time offset between colliding packets based on preamble detection results and obtains collision-free PHY samples. CoLoRa [17] groups LoRa chirps to their corresponding LoRa nodes by examining the power level of the same frequency in different demodulation windows. NScale [16] amplifies the time offsets between colliding packets with non-stationary signal scaling. Those previous works mainly resolve collisions by leveraging the time domain and the frequency domain information. Our protection mechanism complements and enhances the previous works. We extract legitimate chirps from jamming chirps by examining their corresponding received power in demodulation windows.

VIII. Conclusion

In this paper, we reveal the vulnerability of LoRa PHY under the attack of synchronized jamming chirps. The insight of the jamming attack is that a well-synchronized jamming chirp cannot be separated from a legitimate LoRa chirp in the time domain. As a result, most existing protection methods cannot protect the LoRa PHY against such synchronized jamming chirps. To enhance the LoRa PHY, we propose a novel countermeasure, which leverages the difference between the received signal strength of legitimate chirps and jamming chirps in the power domain. The protection method can complement and enhance existing collision recovery schemes which leverage the chirp misalignment in time domain or the frequency disparity in frequency domain.

Acknowledgement

This work is supported by the National Nature Science Foundation of China under grant 61702437 and Hong Kong GRF under grant PolyU 152165/19E. Yuanqing Zheng is the corresponding author.

References

[1] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, “Known and unknown facts of lora: Experiences from a large-scale measurement study,” *ACM Trans. Sen. Netw.*, vol. 15, no. 2, Feb. 2019.

[2] J. P. S. Sundaram, W. Du, and Z. Zhao, “A survey on lora networking: Research problems, current solutions and open issues,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, 2019.

[3] W. Gao, W. Du, Z. Zhao, G. Min, and M. Singhal, “Towards energy-fairness in lora networks,” in *IEEE ICDCS'19*, 2019.

[4] X. Xia, Y. Zheng, and T. Gu, “Litenap: Downclocking lora reception,” in *IEEE INFOCOM'20*, 2020.

[5] Semtech lora applications. [Online]. Available: https://www.semtech.com/lora/lora-applications

[6] F. Zhang, Z. Chang, K. Niu, J. Xiong, B. Jin, Q. Lv, and D. Zhang, “Exploring lora for long-range through-wall sensing,” *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, no. 2, Jun. 2020. [Online]. Available: https://doi.org/10.1145/3397326

[7] L. Chen, J. Xiong, X. Chen, S. I. Lee, K. Chen, D. Han, D. Fang, Z. Tang, and Z. Wang, “Widesee: Towards wide-area contactless wireless sensing,” in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, ser. SenSys '19, 2019, p. 258–270.

[8] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, “A survey on jamming attacks and countermeasures in wsns,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, 2009.

[9] E. Aras, N. Small, G. S. Ramachandran, S. Delbruel, W. Joosen, and D. Hughes, “Selective jamming of lorawan using commodity hardware,” in *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, 2017, pp. 363–372.

[10] A. Rahmadhani and F. Kuipers, “When lorawan frames collide,” in *Proceedings of the 12th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, 2018, pp. 89–97.

[11] K. Mikhaylov, R. Fujdiak, A. Pouttu, V. Miroslav, L. Malina, and P. Mlynek, “Energy attack in lorawan: experimental validation,” in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019, pp. 1–6.

[12] X. Xia, Y. Zheng, and T. Gu, “Ftrack: Parallel decoding for lora transmissions,” in *Proceedings of the 17th Conference on Embedded Networked Sensor Systems*, 2019, pp. 192–204.

[13] R. Eletreby, D. Zhang, S. Kumar, and O. Yağan, “Empowering low-power wide area networks in urban settings,” in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 309–321.

[14] X. Wang, L. Kong, L. He, and G. Chen, “mlora: A multi-packet reception protocol in lora networks,” in *2019 IEEE 27th International Conference on Network Protocols (ICNP)*. IEEE, 2019, pp. 1–11.

[15] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, D. Fang, and K. Jamieson, “Plora: Passive long-range data networks from ambient lora transmissions,” in *ACM SIGCOMM'18*, 2018.

[16] S. Tong, J. Wang, and Y. Liu, “Combating packet collisions using non-stationary signal scaling in lpwans,” in *ACM MobiSys'20*, 2020.

[17] S. Tong, Z. Xu, and J. Wang, “Colora: Enable muti-packet reception in lora,” in *IEEE INFOCOM'20*, 2020.

[18] Z. Wang, L. Kong, K. Xu, L. He, K. Wu, and G. Chen, “Online concurrent transmissions at lora gateway,” in *IEEE INFOCOM'20*, 2020.

[19] SigFox. (2019, Jan.) Sigfox overview. [Online]. Available: https://www.sigfox.com/en/sigfox-iot-technology-overview

[20] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, and A. Ghosh, “Nb-iot system for m2m communication,” in *Proceedings of IEEE Wireless Communications and Networking Conference*, ser. (WCNC 2016), Apr 2016, pp. 1–5.

[21] M. Lauridsen, I. Z. Kovacs, P. Mogensen, M. Sorensen, and S. Holst, “Coverage and capacity analysis of lte-m and nb-iot in a rural area,” in *Proceedings of IEEE 84th Vehicular Technology Conference*, ser. (VTC-Fall 2016), Sep 2016, pp. 1–5.

[22] L. Alliance. (2019, Jan.) Lorawan for developer. [Online]. Available: https://lora-alliance.org/lorawan-for-developers

[23] A. Rahmadhani and F. Kuipers, “When lorawan frames collide,” in *Proceedings of the 12th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WiNTECH'18)*. New York, USA: ACM, Nov 2018, pp. 89–97.

[24] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, “Understanding the limits of lorawan,” *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, Sep. 2017.

[25] D. Bankov, E. Khorov, and A. Lyakhov, “On the limits of lorawan channel access,” in *Proceedings of the 2016 International Conference on Engineering and Telecommunication (EnT)*, Nov 2016, pp. 10–14.

[26] M. C. Bor, U. Roedig, T. Voigt, and J. M. Alonso, “Do lora low-power wide-area networks scale?” in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'16)*, Nov 2016, pp. 59–67.

[27] J. Haxhibeqiri, F. V. den Abeele, I. Moerman, and J. Hoebeke, “Lora scalability: A simulation model based on interference measurements,” *Sensors*, vol. 17, no. 6, p. 1193, Mar 2017.

[28] U. Noreen, A. Bounceur, and L. Clavier, “A study of lora low power and wide area network technology,” in *Proceedings of the 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, Oct 2017, pp. 1–6.

[29] A. Lavric and V. Popa, “A lorawan: Long range wide area networks study,” in *Proceedings of the 2017 International Conference on Electromechanical and Power Systems (SIELMEN)*, Oct 2017, pp. 417–420.

[30] T. Bouguera, J.-F. Diouris, J.-J. Chaillout, R. Jaouadi, and G. Andrieux, “Energy consumption model for sensor nodes based on lora and lorawan,” *Sensors*, vol. 18, no. 7, p. 2104, Jul 2018.

[31] J. Petajajarvi, K. Mikhaylov, A. Roivainen, T. Hanninen, and M. Pettissalo, “On the coverage of lpwans: range evaluation and channel attenuation model for lora technology,” in *Proceedings of the 2015 14th International Conference on ITS Telecommunications (ITST)*, Dec 2015, pp. 55–59.

[32] J. Petäjäjärvi, K. Mikhaylov, M. Pettissalo, J. Janhunen, and J. Iinatti, “Performance of a low-power wide-area network based on lora technology: Doppler robustness, scalability, and coverage,” *International Journal of Distributed Sensor Networks*, vol. 13, no. 3, pp. 1–16, Mar 2017.

[33] N. Hou and Y. Zheng, “Cloaklora: A covert channel over lora phy,” in *The 28th IEEE International Conference on Network Protocols (ICNP'20)*, 2020.

[34] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe, “Charm: Exploiting geographical diversity through coherent combining in low-power wide-area networks,” in *Proceedings of the ACM/IEEE IPSN'18*, 2018, p. 60–71.

[35] A. Gadre, R. Narayanan, A. Luong, A. Rowe, B. Iannucci, and S. Kumar, “Frequency configuration for low-power wide-area networks in a heartbeat,” in *Proceedings of the USENIX NSDI'20*, 2020, pp. 339–352.

[36] A. Gadre, F. Yi, A. Rowe, B. Iannucci, and S. Kumar, “Quick (and dirty) aggregate queries on low-power wans,” in *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2020, pp. 277–288.

[37] A. Balanuta, N. Pereira, S. Kumar, and A. Rowe, “A cloud-optimized link layer for low-power wide-area networks,” in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 247–259.

[38] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, “Lmac: Efficient carrier-sense multiple access for lora,” in *The 26th Annual International Conference on Mobile Computing and Networking (MobiCom'20)*, 2020.

[39] M. Hessar, A. Najafi, and S. Gollakota, “Netscatter: Enabling large-scale backscatter networks,” in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, 2019, pp. 271–284.

[40] R. Nandakumar, V. Iyer, and S. Gollakota, “3d localization for sub-centimeter sized devices,” in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018, pp. 108–119.

[41] W. Xu, W. Trappe, Y. Zhang, and T. Wood, “The feasibility of launching and detecting jamming attacks in wireless networks,” in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, pp. 46–57.

[42] Y. Zou, J. Zhu, X. Wang, and L. Hanzo, “A survey on wireless security: Technical challenges, recent advances, and future trends,” *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.

[43] Y. Liu, H.-H. Chen, and L. Wang, “Physical layer security for next generation wireless networks: Theories, technologies, and challenges,” *IEEE Communications Surveys and Tutorials*, vol. 19, no. 1, pp. 347–376, 2017.

[44] C. Gu, L. Jiang, R. Tan, M. Li, and J. Huang, “Attack-aware data timestamping in low-power synchronization-free lorawan,” in *IEEE ICDCS'20*, 2020.

[45] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes, “Exploring the security vulnerabilities of lora,” in *CYBCONF'17*, 2017.