Decoding LoRa Packets under Collaborative Jamming Attacks

Md Ashikul Haque
University of Texas at Dallas
Richardson, Texas, USA

Abusayeed Saifullah
University of Texas at Dallas
Richardson, Texas, USA

Abstract

This paper addresses the vulnerability of LoRa communications to wireless jamming attacks and proposes an effective anti-jamming technique. Mitigating jamming in a LoRa network is challenging as its devices have limited computation power and energy. The state-of-the-art work addresses jamming by positioning at least three gateways in a line and exploiting spatio-temporal offsets across them. It is effective only against a single jammer. When multiple attackers collectively jam a signal, the variability in arrival times of jamming signals across gateways results in differing combined jamming signals at each gateway, rendering the offset exploitation technique impractical. We propose a new technique to handle collaborative jamming where multiple attackers collectively jam a signal. Our idea is to first disentangle the combined jamming signals by transmitting a known signal that remains orthogonal to the LoRa signal on the same channel but collides with the jamming signals. This method is link layer-agnostic, entails no overhead at the LoRa nodes, and enables packet decoding under both a single jammer and multiple jammers on a channel, effectively combating various types of jamming – reactive, proactive, or random. We have implemented our anti-jamming system at the LoRa gateway and conducted experiments under various jamming scenarios on LoRa nodes. The results show that it improves packet reception rate and per packet energy consumption by up to 100× (40.7% vs 0.4%) and 136× (4.52 mJ vs 615 mJ), respectively, in the harshest three-jammer outdoor setup with 50 m average jammer distance to the gateway;

CCS Concepts

• **Networks** → **Wireless access networks**; *Network experimentation*; • **Security and privacy** → *Denial-of-service attacks.*

Keywords

LoRa, LPWAN, collaborative jamming, anti-jamming, wireless security

1 Introduction

Jamming is a kind of denial of service attack in which malicious devices block legitimate communication by causing intentional interference in networks. Its impact on wireless networks extends across economic, social, and military realms. In 2020, 85% of cargo truck thefts in Mexico reportedly involved wireless jamming [25]. A notable incident occurred in March 2022 when SpaceX's Starlink faced a jamming attack [18]. Reliable communication is essential during emergencies, but jamming disrupts this, putting emergency responses and the sharing of vital information at risk. Safeguarding wireless communication by mitigating this serious threat is therefore critical at both national and global scales.

In this paper, we aim at fortifying LPWAN (low-power wide-area network) communications against wireless jamming attacks. LPWANs, designed for low-power, low data rate communication over long distances, are transforming the IoT landscape. The demand for IoT applications is rapidly growing, with an estimated 29 billion IoT devices projected by 2030 [34]. LoRa, a leading LPWAN technology [11, 15, 19], is globally deployed with hundreds of millions of devices across every inhabited continent, supporting over 600 use cases such as ship monitoring, asset anti-theft, vaccine temperature monitoring, and workplace CO_2 level monitoring [22]. ABI Research predicts that LoRa will constitute over 50% of all LPWAN connections by 2026 [33]. Given LoRa's widespread adoption and diverse applications, we propose an anti-jamming technique specifically tailored for LoRa.

Although LoRa is known to be resistant to low Signal-to-Noise Ratio (SNR), it may fail to decode a signal if interfered with by a significantly stronger signal. In various scenarios, e.g., in battlegrounds, remote deployments, and densely crowded places, jammers can transmit such strong jamming signals and they may not be traced. Besides, in any scenario, if multiple jammers collaboratively send jamming signals, each using a moderate transmission power, a LoRa signal can get completely buried. Recent research has already shown that LoRa communications are vulnerable to jamming attacks leading to significant packet loss, delays, and faster battery drain [1, 23].

In a recent research [14], the authors observed that severe jamming drastically reduces the packet reception rate from a LoRa node to below 1% with a single jammer and to 0% with multiple jammers, even under increasing transmission (Tx) power and spreading factor (a LoRa parameter that enhances reliability). With its rapid expansion and widespread adoption, LoRa thus faces a growing challenge from jamming that demands effective mitigation strategies. Due to its extensive coverage, LoRa signals can be detected over long distances, rendering them susceptible to significant jamming from multiple sources. A critical vulnerability lies in the fact that jamming a small portion of the spectrum can disrupt a large number of LoRa devices. Specifically, in LoRa networks, all end devices communicate directly with a gateway, making the gateway a single point of failure; jamming at this point can disable the entire network.

Mitigating jamming in a LoRa network is extremely challenging as the devices have low computation power and limited energy typically supplied by small and independent batteries. Existing work for LoRa mostly examine jamming impact [1, 16, 17, 23]. Several studies address collision issues between LoRa packets or aim to decode signals only under low interference power [4, 7, 21, 35, 36, 38, 39]. These are not suitable for mitigating jamming as the jammers can send any jamming signal and the target LoRa signal quality may reach far below the SNR required for reception. A recent study proposed in [13] addresses jamming by strategically positioning at least three gateways in a line and exploiting spatio-temporal offsets across the gateways. It is effective only against a single jammer. When multiple attackers collectively jam a signal,

the variability in arrival times of jamming signals across gateways results in differing combined jamming signals at each gateway, rendering the offset exploitation technique impractical.

In this paper, we propose a new technique to handle collaborative jamming where multiple attackers collectively jam signals on a channel in a LoRa network. The core idea behind our technique is to first disentangle the combined jamming signals (sent from multiple jammers) that jam a LoRa packet at the gateway. This is achieved by transmitting a known signal that remains orthogonal to the LoRa signal on the same channel but collides with the jamming signals. Upon disentangling, the combined jamming signal is exploited and subtracted from the collided signal to recover a jammed LoRa packet. This proposed method is link layer-agnostic, entails no overhead at the LoRa nodes, and enables packet decoding when facing attacks from both a single jammer and multiple jammers on a channel, effectively combating various types of jamming – reactive, proactive, or random.

We have implemented our anti-jamming system at a LoRa gateway based on USRP B200 [8] using GNU Radio [9]. Outdoor experiments were conducted to evaluate its performance under various jamming scenarios in a deployment of ten LoRa nodes. The results show that our technique enhances packet reception rates by up to 100× (40.7% vs 0.4%) and reduces energy consumption per packet by up to 136× (4.52 mJ vs 615 mJ) compared to LoRaWAN in the harshest three-jammer outdoor setup with 50 m average jammer distance to the gateway. These gains correspond to the harsher evaluated points rather than every operating point.

In the rest of the paper, Section 2 discusses related work. Section 3 provides an overview of LoRa, and our jamming and system model. Section 4 introduces the core concept for decoding jammed signals, while Section 5 presents the design of the anti-jamming system. Section 6 presents the experimental results. Section 7 concludes the paper.

2 Related Work

Prior work has extensively studied the effects, detection, and mitigation of jamming in wireless networks [10, 28]. Common defenses include spread spectrum, adaptive transmit power, frequency hopping, coding, and covert-channel techniques [3, 5, 20, 24, 26, 27, 29, 30, 37]. However, under strong or agile jammers these approaches often degrade substantially. LR-FHSS provides resilience through subcarrier hopping but uses a different physical layer than the standard CSS used in LoRa, so its mechanism does not directly apply to commodity LoRa.

Several LoRa-specific techniques (e.g., mLoRa [36], FTrack [39], and CoLoRa [35]) focus on resolving collisions between valid LoRa packets and thus do not address adversarial (and potentially non-LoRa) jamming. Other works improve LoRa robustness using multi-gateway diversity, learning-based decoding, or retransmissions [4, 7, 21, 38], but they still require sufficiently high post-interference SNR (around -35 dB) to decode, which sustained jamming can violate.

Recent LPWAN work has examined LoRa jamming threats and their impact [1, 12, 13, 16, 17, 23, 40]. The countermeasure in [16] targets synchronized chirp-based jamming and relies on alignment and power-difference assumptions, and it does not address collaborative jamming.

Prior work in SNOW [12, 31, 32] and single-jammer LoRa solution [13] do not extend to collaborative jamming, since multiple independent jammers create gateway-dependent composite interference that prevents exploiting stable offsets. A recent multi-attacker LoRa approach [14] relies on implicit symbol synchronization to keep jamming energy distinguishable in FFT bins, but it assumes LoRa-form jamming and targets uplink only. In contrast, we handle multiple jammers whose waveforms can be non-LoRa and time-varying, and our design applies to both uplink and downlink.

3 Background and System Model

3.1 LoRa Overview

A LoRa system primarily consists of end-devices (nodes) and gateways. Nodes are typically battery-powered with limited energy, while gateways are line-powered and relay packets between nodes and a network server. LoRa uses Semtech's Chirp Spread Spectrum (CSS) modulation with configurable parameters such as carrier frequency, bandwidth (BW), spreading factor (SF), and transmit power. The SF values ($6 \leq s \leq 12$) determine the number of chips per symbol (2^s); higher SF improves sensitivity and range at the cost of a lower data rate.

Modulation. CSS uses a base *up-chirp* whose instantaneous frequency sweeps linearly across the channel bandwidth. A data symbol is encoded by a cyclic time (equivalently, frequency) shift of this base chirp; the shift index corresponds to a value in $\{0, \ldots, 2^s - 1\}$.

Demodulation. After detecting the preamble, the receiver *dechirps* each symbol by multiplying the received up-chirp with a corresponding base *down-chirp*. An FFT over the symbol window concentrates energy into a single bin whose index reveals the transmitted symbol. The decoder then applies FEC and CRC to correct residual errors and verify packet integrity.

3.2 Jamming Attack Model

Jamming disrupts wireless communication by intentionally transmitting interfering signals on the same channel. We consider a LoRa network where a channel may face *proactive*, *reactive*, or *random* jamming. In *proactive jamming*, an attacker predicts a legitimate transmission and transmits at the right time to interfere it. In *reactive jamming*, an attacker waits until it detects a legitimate signal and then transmits to interfere it. In *random jamming*, the attacker transmits at random times. In all cases, jammers do not encode data, they only aim to disrupt reception.

We consider both a single jammer and *collaborative jamming*, where multiple attackers jam the same channel, making mitigation harder. Jammers may be at different locations, may coordinate, and may start transmitting simultaneously or at different times. Each jammer can transmit a different waveform and can vary it across transmissions. We assume a small number of jammers per channel (e.g., up to four), since many jammers would be costly and can fully bury the LoRa signal, leaving little practical room for recovery.

A jammer need not be energy-limited in our model. The energy-budget intuition only explains why some attackers may prefer efficient disruption; our design and evaluation also consider strong powered jammers through higher transmission powers. A jammer can transmit long packets or multiple consecutive packets, but

excessively long signals are unnecessary because the goal is simply to bury the target LoRa packet during its airtime.

We focus on separating a LoRa packet from jamming signals. Handling collisions between multiple LoRa packets is a different problem, typically addressed by existing MAC mechanisms or collision recovery techniques. The present paper implements and evaluates the uplink case at the gateway. Extending the full recovery pipeline to simple end devices for downlink is outside the current scope.

4 Underlying Idea for Decoding a Jammed LoRa Signal

In collaborative jamming, multiple attackers from different physical locations collectively jam a channel, making its mitigation much harder. Each jammer can send a different signal and can also vary its signal across transmissions. The jamming signals from different jammers reach different gateways at varying times, causing the combined signals to differ across the gateways. Therefore, if we apply the approach in [13] developed for a single jammer scenario to combat collaborative jamming, it will not work. This is because the approach in [13] relies on the fact that the jamming signal remains almost same across different gateways which may be true for a single jammer scenario but does not hold under collaborative jamming. We conducted an experiment and the results are illustrated in Figure 1, which shows the poor performance of [13] against multiple jammers as the packet reception rate remains close to 1%. Hence, to address collaborative jamming, we will develop an entirely new technique capable of combating both single and multiple jammers.

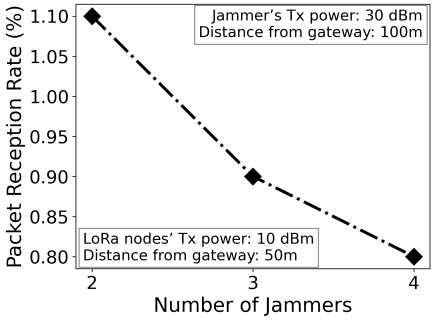


Figure 1: Performance of [13] varying number of jammers.

As mentioned above, collaborative jamming poses a more challenging scenario, leaving no option to exploit their temporal and spatial variation across gateways to combat jamming. Therefore, our proposed approach will first recover the part of the combined jamming signals (from multiple jammers) that interfere/jam a LoRa packet and then subtract it from their collided signal to recover the LoRa packet. This is done by transmitting a known signal that will remain orthogonal to the LoRa signal on the same channel but collide with the jamming signals.

Suppose one LoRa node's signal, say X_{lora}, is jammed by multiple jammers on a channel. Let X_{jammers} denote the combined signal of the jammers on the channel. Now $X_{\text{lora}} + X_{\text{jammers}}$ will represent all

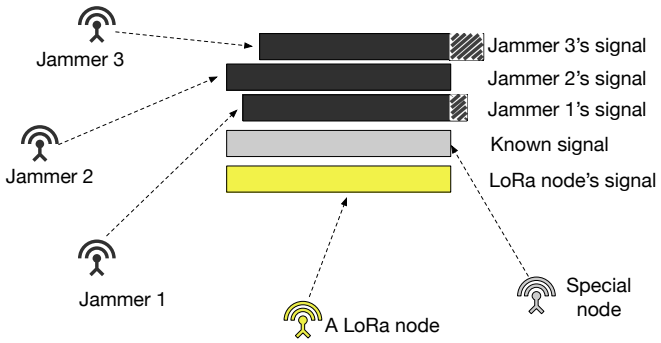


Figure 2: A scenario that shows a LoRa signal, the known signal, and three jammers' signals collided at the gateway.

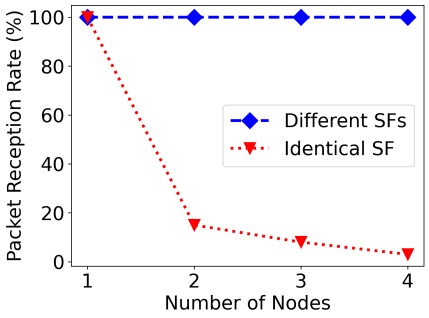


Figure 3: Orthogonality of SFs in LoRa.

collided signals. Our recovery technique is based on the following idea. If we know X_{jammers}, we can subtract it from $X_{\text{lora}} + X_{\text{jammers}}$ and recover the desired LoRa signal (X_{lora}). Suppose, X_{jammers} remains unchanged across transmissions. Then after sending X_{lora} signal, the LoRa node can send a known signal X_{known} that is known to the gateway in the next transmission. Assuming X_{jammers} is the same across these two cases, the collided signals in the second case will be given by $X_{\text{known}} + X_{\text{jammers}}$. The gateway now can subtract X_{known} from $X_{\text{known}} + X_{\text{jammers}}$, and recover X_{jammers}. Then from the first case where it received $X_{\text{lora}} + X_{\text{jammers}}$, it can subtract X_{jammers} and recover X_{lora}. Now if we consider X_{jammers} changes across transmissions, this will not work. If we want to recover in this way, we have to get $X_{\text{known}} + X_{\text{jammers}}$ and $X_{\text{lora}} + X_{\text{jammers}}$ from the same transmission. To make this possible, we shall apply the orthogonality of the spreading factors to receive $X_{\text{lora}} + X_{\text{jammers}}$ and $X_{\text{known}} + X_{\text{jammers}}$ after making a transmission from a LoRa node as described below.

In this description, we use the term '*LoRa signal/Tx*' to indicate the LoRa transmission that we want to receive and/or recover from the interference of the jammers. The known signal that we transmit to recover this LoRa packet is also a LoRa signal but will be called '*known signal/Tx*' to avoid any confusion. We use a special node that will transmit the known signal on a channel at the same time (after detecting the LoRa node's signal) when the LoRa node transmits on that channel. We conducted an experiment where a varying number of LoRa nodes transmitted packets simultaneously on the

same channel using both different and identical SFs, and the results are illustrated in Figure 3, which demonstrates the orthogonality of SFs in LoRa. Therefore, the LoRa node and the special node will use two different SFs and will not collide as LoRa SFs are orthogonal to each other on the same channel. In the current design, this argument is used under the same BW setting and with a reserved SF for the known signal. Since LoRa SFs are only quasi-orthogonal in practice, mixed-BW or slope-matched BW/SF combinations are outside the evaluated scope. We need a single special node for the entire network. Note that a LoRa network typically uses multiple gateways. One option could be to use the target channel (that we consider for anti-jamming) of one gateway to send the known signal while the gateway continues to operate on other channels for its regular services. This ensures that the special node does not introduce significant overhead. Using a USRP device as the special node offers a more cost-effective and flexible solution, which is why we shall choose this option. The configuration of the special node will be detailed in the next section.

Figure 2 shows a scenario where the LoRa signal, the known signal, and the signals from three jammers collide at the gateway. All of these collided signals received at the gateway can be represented by $X_{\text{lora}} + X_{\text{known}} + X_{\text{jammers}}$. In this signal, the gateway can ignore the parts of X_{jammers} (shown in gray/white strip color) that do not collide with the known or the desired LoRa signals. Since X_{lora} and X_{known} were transmitted on different SFs, they do not collide with each other. In other words, in absence of X_{jammers}, the gateway could receive X_{lora} and X_{known} correctly as they are on two different SFs. Therefore, upon receiving the collided signal $X_{\text{lora}} + X_{\text{known}} + X_{\text{jammers}}$, the gateway multiplies with the known signal's down-chirp and gets $X_{\text{known}} + X_{\text{jammers}}$ signal. From this signal, we can now subtract the known signal X_{known} and get the jammers' signals X_{jammers}. After that, the gateway multiplies the collided signal $X_{\text{lora}} + X_{\text{known}} + X_{\text{jammers}}$ with the X_{lora} signal's down-chirp and gets $X_{\text{lora}} + X_{\text{jammers}}$ signal. From this signal, the gateway can now subtract X_{jammers} and will get X_{lora}. In absence of jammer signals, the gateway receives $X_{\text{lora}} + X_{\text{known}}$ and can simply subtract the known signal X_{known} to recover the desired signal X_{lora}. If the network does not experience jamming, the special node can be turned off. It can be turned on again if the network starts experiencing jamming.

This approach is effective against conventional jammers because both the known signal and the LoRa signal are transmitted **simultaneously on the same frequency channel**. A jammer aiming to disrupt that frequency band by transmitting interfering noise will affect both signals as they occupy the same spectral resources at the same time. This allows the impact of the jamming signal to be measured using our known signal and subsequently removed from the target LoRa signal.

Addressing Variable Signal Lengths. We shall use the known signal length equal to maximum possible LoRa packet length. Therefore, the LoRa nodes can use packets of variable length and they will fully overlap with the known signal. However, the combined jamming signal length can be much bigger. Therefore, on either ends there can be collision-free fragments and they can be ignored to apply our proposed decoding theory. In proactive jamming, there is always a fragment of the combined jamming signal in the beginning that does not collide with the LoRa signal (as the latter has not yet started). In both proactive jamming and reactive jamming, there can be collision-free fragment of (either the LoRa signal or the jamming signal).

Based on the above idea, we describe our system for handling collaborative jamming in the following section. While the subtraction principle is not inherently limited to uplink, this paper implements and evaluates uplink recovery at the gateway. Downlink recovery at constrained end devices is left for future work.

5 Design of the Anti-Jamming System

The proposed anti-jamming system is designed to decode packets in LoRa networks by utilizing a special LoRa node and a gateway. The special node transmits a LoRa signal with a unique spreading factor (SF) upon detecting an incoming LoRa packet, while the gateway employs a novel dual-demodulation pipeline to decode the LoRa signal despite the presence of multiple jammers. This section details the design and operational principles of both special node and decoder at gateway.

The system utilizes the technique discussed above to decode packets even when multiple jammers are transmitting. Our first goal is to find the jamming signal with the help of a special node's signal and use that information to decode the LoRa packet. To achieve this, we use a special node placed near the gateway and modify its MAC layer accordingly. We ensure that the special node always transmits a known signal simultaneously with the LoRa signal using the same channel but a different SF.

At the gateway, we introduce two decoders as illustrated in Figure 4. The first decoder's responsibility is to find the jamming signal and pass that information to the second decoder. The second decoder then uses the information from the first decoder to decode the symbols of the LoRa packet. Note that the second decoder uses the signal received by the Rx (receive) antenna and the knowledge obtained from the first decoder. If we peek into the first decoder, we will see it uses a modified traditional LoRa decoder. It multiplies each segmented symbol with the segmented base down-chirp of the known signal's SF and applies FFT on that. From its prior knowledge of the special node's signal, it knows how the segmented FFT bins should look. By leveraging this knowledge, it isolates the energy peaks contributed by the jammers. This information is passed to the second decoder.

The second decoder first utilizes the signal received by the Rx antenna and multiplies each symbol with the base down-chirp of the LoRa signal's SF. It applies FFT on the multiplied signal and gets the energy in the FFT bin of each symbol. The FFT bin contains both LoRa signal energy and jamming signal energy. There is no energy from the special node's signal as it is orthogonal to the LoRa signal. Since we already have information about the energy contributed by the jamming signal in the FFT bins from the first decoder, we can subtract those and get the energy only contributed by the LoRa signal. We can then select the FFT bin with the highest peak and decode that as our symbol. This process continues until the entire LoRa packet is decoded.

5.1 Challenges

This design poses some serious challenges that need to be addressed to have a functional system. Firstly, how do we transmit a known

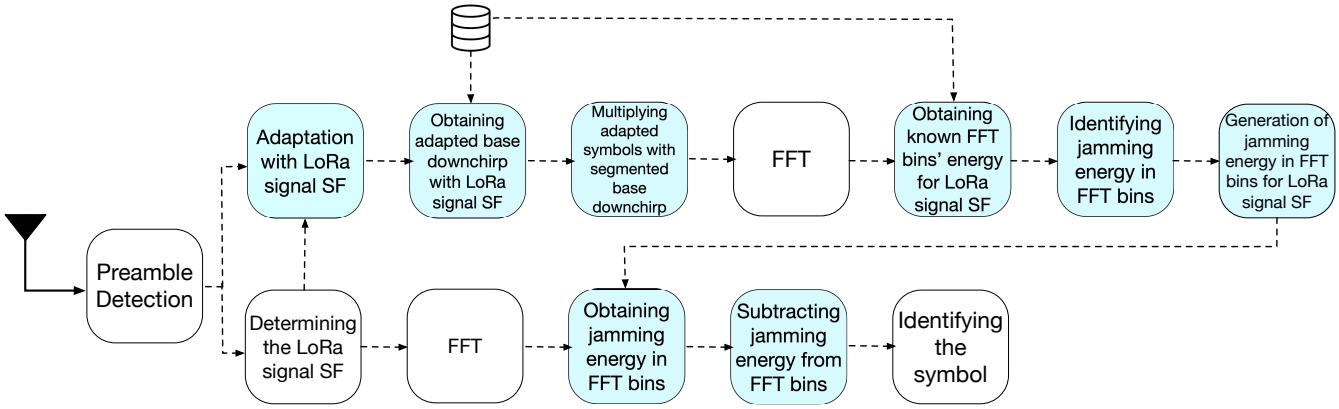


Figure 4: Illustration of workflow in anti-jamming decoder.

signal simultaneously with a LoRa signal? Secondly, how do we address unequal symbol lengths across SFs? Thirdly, how do we address known samples that are in multiple demodulation window?

5.1.1 *How to Transmit a Known Signal Simultaneously with a LoRa Signal?*

Here, the special node (transmitting the known signal) and the LoRa node (transmitting the desired signal) use the same channel with different SFs. To detect LoRa node transmissions, the special node listens for uplink transmissions in the medium. CSS modulation in LoRa makes RSSI-based detection impractical, as signals may be below the noise floor. Instead, we use the carrier activity detection (CAD) feature of LoRa chips, which is not utilized in LoRaWAN. CAD probes for packet preambles in a channel with spreading factor s and bandwidth BW for a fixed duration $(2^s + 32)/BW$, which is roughly one LoRa symbol plus a fixed CAD overhead rather than exactly two full symbols. Upon detecting activity, the special node immediately transmits the known signal. While continuous probing raises energy concerns, the special node can be line-powered due to its proximity to the gateway.

A potential concern with placing the special node near the gateway is the near-far problem, where the strong signal from the special node could inadvertently jam the weaker, distant LoRa signal. However, this is mitigated by the fundamental orthogonality of Spreading Factors in LoRa. As demonstrated by our experiment shown in Figure 3, signals with different SFs do not interfere with each other during the decoding process. Even if the known signal is received with much higher power, the gateway's demodulator for the LoRa node's SF will not get any FFT peak from it, effectively isolating the two signals.

The special node's placement ensures the known signal reaches the gateway simultaneously with the LoRa signal. LoRa's narrow bandwidth results in large sampling intervals; for example, 8 μs for a bandwidth of 125 kHz, meaning signals transmitted within 8 μs align to the same sample. With a bandwidth $\leq$ 500 kHz and a sampling interval $\geq$ 2 μs, a special node placed within 2 μs (600 meters) avoids propagation delay.

We empirically measure the latency of the special node's detection of X_{lora} and transmission of the X_{known} signal. In our setup, we use a USRP B200 connected to a system with AMD 7800x3d processor as the special node, and the average latency is approximately 300 μs, which is low enough that the X_{known} signal can always reach the gateway either before or simultaneously with X_{lora}'s payload. We configure the transmission of X_{known} to coincide with the arrival of X_{lora}'s payload at the gateway. Constant listening through CAD, enabled by line power and proximity to the gateway, consistently ensures that X_{known} reaches the gateway at the same time as X_{lora}'s payload.

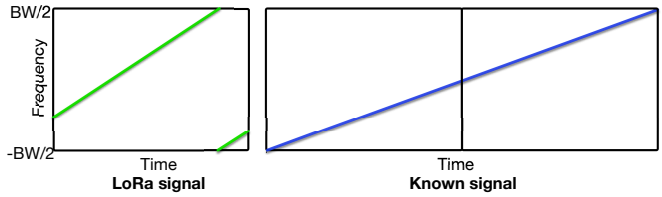


Figure 5: An illustration of unequal symbol windows.

5.1.2 *How to Address Unequal Symbol Lengths across SFs?*

In the collided signal ($X_{\text{lora}} + X_{\text{known}} + X_{\text{jammers}}$), X_{lora} and X_{known} have orthogonal SFs, while the jammers' signals can vary. Multiplying the collided signal by the base down-chirps of X_{known} and X_{lora} results in different symbol windows due to their distinct SFs. As shown in Figure 5, varying SFs lead to different symbol windows and sample counts. In the example, the LoRa transmitter operates at an SF one less than the known signal, halving the symbol window length and sample count. Since SF represents the number of chips per symbol (2^s for SF s), an FFT on entire known signal symbol window aggregates energy from jamming signals irrelevant to decoding X_{lora}.

We address this challenge by splitting or aggregating the X_{known} according to X_{lora}'s SF. As illustrated in Figure 5, if the SF of X_{lora} and X_{known} are s and $(s+1)$ respectively (resulting in a double symbol window size for X_{known}), we split the X_{known} symbols in half to match the symbol window of X_{lora}. We also need to consider the base down-chirp of X_{known}, because it will be multiplied with the X_{known} symbols. We split or aggregate the base down-chirp in the same manner as the X_{known} symbols. As depicted in Figure 5, the base down-chirp is similarly split in half to match the X_{lora} symbol

window. This ensures the demodulation process of X_{known} is not disrupted while matching X_{lora}. This depiction and explanation exemplify the case when X_{known} has an SF one higher than X_{lora}. The same principle applies when X_{known}'s SF is more (resulting in more splits). If X_{known}'s SF is lower than X_{lora}, we aggregate X_{known} symbols by SF difference to match the symbol length.

5.1.3 ***How to Address the Changing Energy Distribution across FFT Bins for Different SFs?*** When a jamming signal is multiplied by the base down-chirp of a different spreading factor (SF), the energy distribution across FFT bins changes significantly. This shift complicates the accurate identification and subtraction of jammer signals, as energy peaks associated with the jamming signal no longer align consistently across different SFs. Without addressing this discrepancy, it becomes challenging to accurately isolate and subtract the jamming energy, potentially degrading system performance.

To solve this, we employ a lightweight generative model trained specifically to learn the transformation of energy distribution in FFT bins from one SF to another. We generate the training data by multiplying the same jamming signal with base downchirps of different SFs and subsequently applying FFT operations to capture the energy distribution in FFT bins. The resulting FFT energy distributions serve as ground truth labels for training the model. During testing, given the FFT energy distribution for one SF, the generative model quickly predicts the corresponding distribution for another SF.

5.1.4 ***How to Address Samples that Overlap between Different Demodulation Window?*** While segmenting or aggregating X_{known} symbol aids in resolving the unequal symbol length issue, it introduces a new obstacle. As illustrated in Figure 6, the next demodulation window may contain X_{known} symbols from the previous demodulation window, which will get included during multiplication with the respective base down-chirp samples and subsequent FFT application. This results in energy being distributed across different demodulation windows. For instance, in the depicted example, the symbol of the first demodulation window is a split zero, and the new demodulation window is also a split zero. Consequently, we must account for these split energies in multiple FFT bins due to segmentation when subtracting to obtain the jamming energy peaks.

To address this challenge, we consider all splitting or aggregating of X_{known} symbols. We capture all symbols in our known LoRa packet and create splitting or aggregating for all SFs other than its own. For example, if our known LoRa packet comprises N symbols and SF $s+1$, the total duration for this known LoRa packet is proportional to $N \times 2^{(s+1)}$. We calculate the total packet duration and symbol duration for all other SFs. Given the limited number of SFs (ranging from 6 to 12, resulting in a count of 7), we precompute all segmentations and store them in a database. Upon calculation, we obtain the respective FFT bin values and leverage these values for subtraction to determine the jamming energy in the FFT bins during demodulation of X_{lora}. As shown in (C) and (F) of Figure 6, we split an X_{known} symbol FFT window into half and store them for future usage. This approach also mitigates the issue associated with

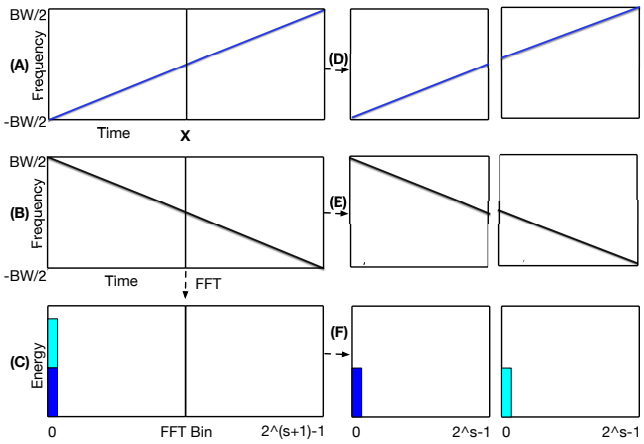


Figure 6: An illustration of overlapping samples.

changing the FFT projection of the known LoRa signal. As adjustments in the number of FFT bins may project energy into a lower or higher number of bins, risking energy leakage to neighboring bins, the aforementioned approach proactively addresses this concern by considering all SF combinations and their respective FFT bin values beforehand. Note that by doing the opposite (expanding FFT bins and adding FFT energy of aggregated symbols), we can address this issue when X_{known}'s SF is lower. Furthermore, we precompute the splitting or aggregating of base down-chirp for all other SFs as illustrated in (B) and (E) of Figure 6 and store them in a database.

5.2 Special Node Configuration

We currently use one special node for the jammed channel being protected. In a multi-gateway deployment, one can associate one special-node instance with each protected gateway region. In our prototype, a USRP B200 serves as the special node because it offers better implementation flexibility.

To achieve the desired functionality, we configure the MAC layer of the special node to continuously listen for incoming LoRa packets on the channel specified by the gateway using CAD. The gateway marks a channel as jammed based on repeated losses and abnormal received-energy patterns, then instructs the special node to monitor that channel. Because this decision is reactive, some packets can be lost before mitigation starts.

Once the special node detects an incoming LoRa packet, it must transmit its own signal using a different SF than that of the detected packet. Determining the SF of the incoming packet by analyzing each symbol's time length on air would introduce delays, which could be detrimental to the system's effectiveness in a jamming scenario. To circumvent this issue, we reserve one SF for the special node; in our default experiments, the LoRa nodes use SF 8 and the special node uses SF 7. By using a unique SF that no other nodes in the network employ, we guarantee that the special node's transmission remains orthogonal to any LoRa node's signal in the network.

With the above techniques and the observations from our empirical results discussed in Section 5.1.1, we ensure that X_{known} arrives at the gateway simultaneously with X_{lora}'s payload. Furthermore, it becomes easier to align the beginning of X_{known}'s symbol window with X_{lora}'s symbol window due to the placement of the special node, which eliminates propagation delay. This alignment helps the gateway process the signal with lower delay, as it removes the need for the gateway to perform alignment. The signal transmitted by the special node is predetermined, meaning that the symbols in the signal are already known to the gateway. Moreover, the gateway has all the combinations of SF and its resultant FFT bin energy from offline computation. By utilizing this known signal, the gateway can separate the jamming signals from the legitimate LoRa node's signal, thereby aiding in the recovery of the original communication. This capability significantly improves the system's effectiveness in maintaining reliable communication despite the presence of jammers.

5.3 Safe Guard for Proactive Jamming

The proposed anti-jamming system remains resilient against sophisticated proactive jammers. This type of jammers attempt to overpower X_{lora} by predicting transmissions and sending high-power jamming signals. Such jamming can obscure the LoRa signal, preventing the special node from even detecting it. Our proactive-jamming safeguard is therefore a hardened operating mode rather than a claim of full unmodified LoRaWAN compatibility. To counter this, our system employs two key safeguards:

First, a longer preamble for X_{lora} allows the special node to accumulate the energy of preamble upchirps into a single FFT bin, enabling the detection of even weak X_{lora} signals against strong jamming. Although this accumulation could introduce a slight alignment delay, we empirically adjust the preamble length to maintain synchronization. Our experiments show that accumulating the energy of 24 base upchirps into one FFT bin reliably detects X_{lora} even at an SNR as low as -35 dB. Accordingly, in this hardened anti-jamming mode we use a preamble length of 32 symbols in low-SNR scenarios. This intentionally deviates from the common shorter LoRaWAN preamble because the special node must reliably detect very weak packets before transmitting X_{known}.

Second, a sliding window technique allows for dynamic accumulation in preamble detection. The special node begins with an accumulation window of two symbols and incrementally adds new upchirp windows, doubling the accumulation until it detects a peak in the FFT bin or the window reaches $\frac{3}{4}\times$ the max preamble length. This dynamic approach ensures timely detection, allowing the special node to detect signals without requiring prolonged accumulation in every case.

Once detection occurs, the remaining $\frac{1}{4}$ of the preamble length (i.e., 8 symbols) provides ample time for the special node to transmit X_{known} in alignment with X_{lora}'s payload at the gateway. This ensures proper alignment for effective decoding despite processing and transmission delays.

These precautions are not necessary when dealing with a reactive jammer, as by the time the reactive jammer's signal X_{jammer} reaches the special node, it has already detected the X_{lora} signal. However, this design remains robust and seamlessly effective against random jammers as well.

5.4 Decoder Design

The decoder at the gateway in our system is designed to decode the LoRa node's signal while mitigating the effects of jamming using the known signal transmitted by the special node, as illustrated in Figure 4 and Figure 7. In Figure 4, our designed blocks are highlighted in light blue, while the standard LoRa decoder blocks are shown in a neutral color. The gateway employs a novel dual-demodulation pipeline that processes both the known and unknown signals to effectively isolate and decode the LoRa signal.

5.4.1 Signal Processing Overview. The gateway receives three types of signals: the combined jamming signal X_{jammers} from multiple jammers, the known signal X_{known} from the special node, and the desired LoRa signal X_{lora} from the node. Notably, X_{known} and X_{lora} are orthogonal and do not interfere with each other. The primary objective of the gateway is to decode X_{lora} while using X_{known} from the special node as a reference to mitigate the effects of jamming.

Traditionally, a LoRa gateway employs a single demodulation pipeline as described in Section 3.1. However, our design introduces a dual-demodulation pipeline to handle both X_{known} (from the special node) and X_{lora} (from the LoRa node). This dual approach is critical for separating X_{jammers} from X_{lora}. After detecting the signal preamble (as described in Section 5.3), the gateway creates two signal copies: one for the first demodulator to identify the jamming signal and another for the second demodulator to decode the LoRa signal. Both pipelines begin processing immediately.

5.4.2 First Demodulation Pipeline: Processing the Known Signal. Upon receiving the detected signal, the first demodulator adapts (splits or aggregates) the signal ($X_{\text{lora}} + X_{\text{known}} + X_{\text{jammers}}$) based on the SF of X_{lora} obtained from the second demodulator, as depicted in (A) of Figure 7. It then retrieves the adapted base down-chirp of X_{known} for the SF of X_{lora} from the database, as shown in (B) of Figure 7. The adapted signal is multiplied by the adapted base down-chirp, and an FFT is performed on the multiplied signal to analyze energy in the frequency domain, as depicted in (C) of Figure 7. Since the adapted base down-chirp corresponds to the X_{known} signal, the FFT bins contain the energy of the X_{known} + X_{jammers} signal, excluding the X_{lora} signal.

Although energy from X_{lora} is still present, multiplying by the base down-chirp of X_{known} under a different SF and the same BW prevents it from coherently combining into a peak. Instead, it spreads across FFT bins as a small residual term, which is why our evaluation keeps the known and desired signals on distinct SFs under a fixed bandwidth.

The demodulator then retrieves the precomputed FFT bin energy for the adapted symbol of X_{known} from the database, as illustrated in Figure 7 (D). It subtracts this energy from the FFT bin of the received signal, resulting in the energy distribution in FFT bins contributed by X_{jammers}, as illustrated in Figure 7 (E). The pre-trained generative model takes the energy distribution of X_{jammers} in FFT bins as input and generates its corresponding distribution for the SF of X_{lora}, as illustrated in Figure 7 (F). This energy distribution in FFT bins is then shared with the second demodulator.

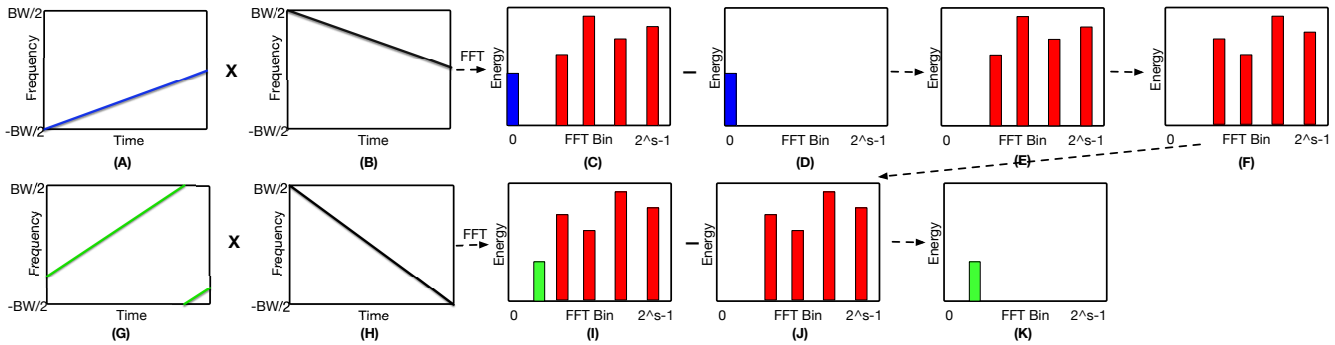


Figure 7: Illustration of anti-jamming decoder at signal level.

5.4.3 Second Demodulation Pipeline: Decoding the LoRa Signal. The second demodulator first sends the SF of X_{lora} to the first demodulator. It then multiplies the signal with the base down-chirp of X_{lora} and applies an FFT to the multiplied signal. Since the base down-chirp corresponds to X_{lora}, the FFT bins contain the energy of the $X_{\text{lora}} + X_{\text{jammers}}$ signal, excluding the X_{known} signal, as depicted in (I) of Figure 7. The second demodulator waits until the first demodulator shares the energy of X_{jammers} in FFT bins, as illustrated in Figure 7 (J). After obtaining this information, it subtracts the jamming signal's energy from the FFT bins of the $X_{\text{lora}} + X_{\text{jammers}}$ signal, as illustrated in Figure 7 (J) subtracted from (I). This subtraction results in FFT bins containing only the energy of X_{lora}, as illustrated in Figure 7 (K). By detecting the peak in FFT bins, the symbol of X_{lora} is decoded.

5.4.4 Detailed Steps and Algorithms. The identification of FFT peaks is a critical step in the demodulation process. The known signal's predetermined symbols allow the gateway to accurately map FFT peaks to specific symbols. This involves calculating the expected FFT peaks for each symbol of the known signal, matching the observed FFT peaks in the received signal to the expected peaks, and subtracting the identified peaks from the total FFT peaks to isolate the jamming signal components.

By repeating the peak identification and subtraction process for all symbols of the known signal, the gateway can reconstruct the jamming signal. This involves aggregating the FFT peaks attributed to the jamming signal across all symbols, creating a comprehensive profile of the jamming signal in the frequency domain, and using this profile to subtract the jamming signal components from the composite signal in the second pipeline.

After isolating the LoRa signal peaks, the gateway decodes the LoRa signal by mapping the isolated FFT peaks to the corresponding LoRa symbols, reconstructing the original LoRa packet from the decoded symbols, and validating the integrity of the reconstructed packet using error-checking mechanisms such as cyclic redundancy check (CRC). We empirically measure the processing latency of our decoder using a USRP B200 connected to a system with an AMD 7800X3D processor. We find that the average processing delay for a 32-byte packet with SF 8 is approximately 12 ms, which is only 1 ms higher than the 11 ms average processing delay observed with a traditional LoRa decoder using the same B200 and 7800X3D system.

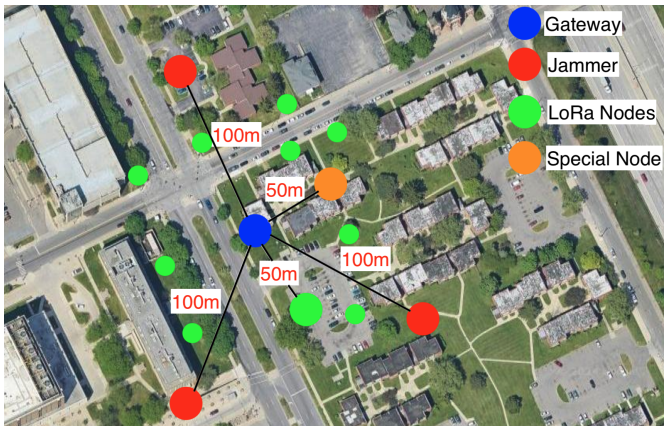


Figure 8: Default Setup.

This low latency is due to pre-calculated downchirps, symbols, and FFT bins of known signal for different SFs.

The dual-demodulation pipeline offers several advantages. By accurately identifying and subtracting the jamming signal components, the gateway enhances the reliability of LoRa communication in jamming-prone environments. The use of a known signal from the special node provides a robust reference point, facilitating the effective separation of the jamming signal from the LoRa signal. Additionally, the orthogonality between the known signal and the LoRa signal ensures minimal interference, aiding in the accurate decoding of the LoRa signal. Finally, the pre-computed components of X_{known} reduce the anti-jamming decoder's latency, demonstrating its efficacy.

6 Experiments

This section describes the experimental setup and evaluates our anti-jamming system.

6.1 Setup

We conduct the experiment outdoors in a suburban metropolitan area using five USRP B200s [8] implemented in GNU Radio [9]. These serve as one LoRa gateway, three jammers, and one special node. For the LoRa nodes, we utilize five Dragino LoRa shield [6]

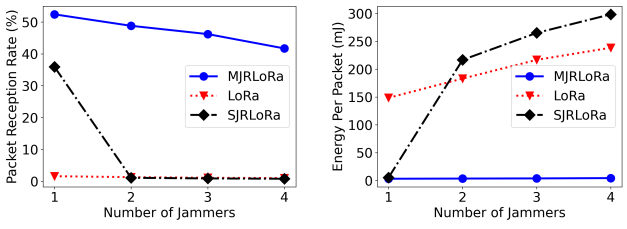
paired with an Arduino Uno R3 [2] and five Raspberry Pi with LoRa HATs based on the SEMTECH SX1276 LoRa transceiver. The jammers are reactive in all setups except when we explicitly vary jamming type, and in the four-jammer experiment the extra jammer is placed on the same circular trajectory around the gateway at the same radius but a different angular position.

In all setups, the LoRa nodes operate on separate channels between 902-915 MHz with a spreading factor of 8 (except in varying-SF configurations), a bandwidth of 125 kHz, and a coding rate of 4/5. The special node uses a spreading factor of 7 (except in varying-SF configurations) across all setups. Each plotted point uses 1000 packets from the affected LoRa node, sent once every 10 seconds; the LoRa packet size is 32 bytes and the known packet is 48 bytes. Each jammer transmits a channel-occupying interference waveform for 2 seconds, which is sufficient to cover a 32-byte LoRa packet at every evaluated SF. In reactive mode this waveform is triggered by packet detection, in proactive mode it is aligned with the scheduled packet transmission, and in random mode it starts at random times. Default transmission powers are 10 dBm for LoRa nodes, 25 dBm for the special node, and 30 dBm for jammers.

The jammers collectively jam one channel at a time, meaning one node is affected at a time in the default setup. We report averages over repeated runs while repositioning jammers on a circular path whose radius matches the specified jammer distance. We first compare against SJRLoRa [13], which by design uses three gateways. This comparison tests whether that prior single-jammer recovery principle extends to collaborative jamming rather than equalizing hardware cost. Because SJRLoRa fundamentally targets the single-jammer case, LoRaWAN is used as the primary baseline afterwards.

6.2 Experimental Result

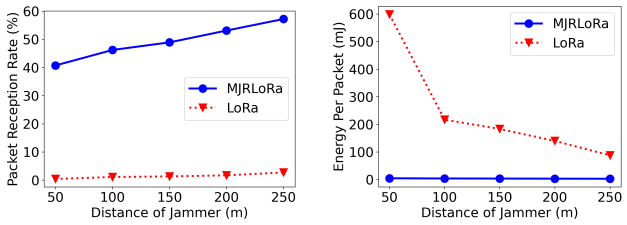
We report ***Packet Reception Rate (PRR)*** and ***Energy Per Packet (EPP)***, where PRR = $\frac{N_{\text{rx}}}{N_{\text{tx}}}$ and EPP = $\frac{E_{\text{tx}}+E_{\text{rtx}}}{N_{\text{rx}}}$. Higher PRR and lower EPP indicate better performance. EPP is important here because LoRa nodes are battery-powered and poor reception quickly translates into repeated transmissions and higher delivery cost per successfully received packet. Our system, named 'MJRLoRa' (Multi-Jamming Resilient LoRa), is compared with 'SJRLoRa' (Single-Jamming Resilient LoRa) [13] and 'LoRa' (LoRaWAN baseline) in the results and discussion.



(a) Packet Reception Rate (b) Energy Per Packet

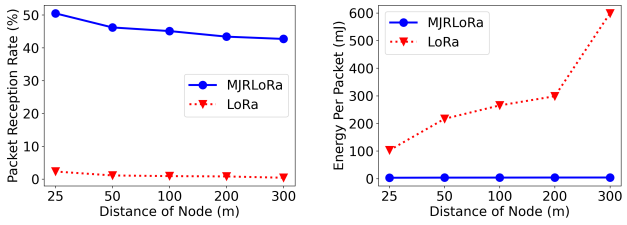
Figure 9: Performance varying number of jammers.

6.2.1 Varying Number of Jammers. In this setup, we vary the number of jammers from one to four, using both SJRLoRa and LoRa as baselines. For SJRLoRa, three USRP B200s serve as LoRa gateways because that prior design relies on multi-gateway diversity. As shown in Figure 9, once more than one jammer is present, the PRR of both baselines stays around 0.8%–1.4%, while their EPP rises to about 184–299 mJ. SJRLoRa remains competitive in the single-jammer case, but its performance collapses with multiple jammers because the composite jammer signal is no longer consistent across gateways. In contrast, MJRLoRa maintains useful recovery performance up to four jammers. For the remaining outdoor experiments, we use three jammers as the default setup.



(a) Packet Reception Rate (b) Energy Per Packet

Figure 10: Performance varying avg. distance of jammers.

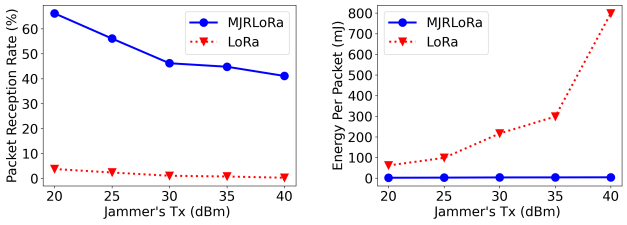


(a) Packet Reception Rate (b) Energy Per Packet

Figure 11: Performance varying distance of node.

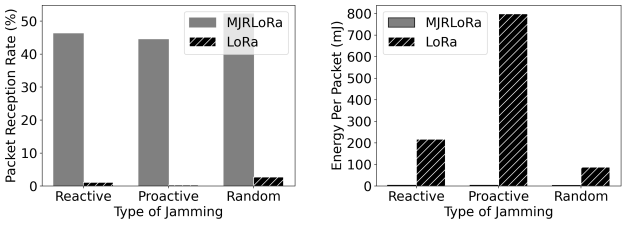
6.2.2 Varying Average Distance Between Jammers and Gateway. In this setup, we vary the average distance between three jammers and the gateway from 50 m to 250 m. Figure 10(a) shows that PRR improves for both MJRLoRa and LoRa as the jammers move farther from the gateway. At 250 m, MJRLoRa reaches 58.21% PRR, while LoRa remains at 2.79%. In this three-jammer setup, at the 50 m average jammer-distance point, MJRLoRa achieves 40.7% PRR while LoRa achieves 0.4%, yielding the highest PRR gain of 101.75 times (40.7% vs 0.4%). Figure 10(b) shows the same trend in energy cost, and at the same point, MJRLoRa requires 4.52 mJ per packet while LoRa requires 615 mJ, yielding the highest EPP reduction of 136.12 times (4.52 mJ vs 615 mJ).

6.2.3 Varying Distance Between Node and Gateway. In this setup, we vary the distance between one LoRa node and the gateway from 25 m to 300 m. As shown in Figure 11(a), PRR decreases for both MJRLoRa and LoRa as the node moves farther from the gateway because the desired signal becomes weaker. However, MJRLoRa remains around the 50% range over much of this sweep, whereas LoRa



(a) Packet Reception Rate (b) Energy Per Packet

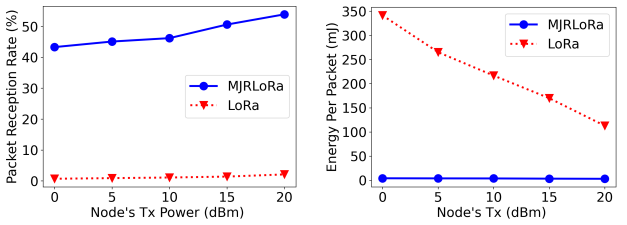
Figure 12: Performance varying Tx of jammers.



(a) Packet Reception Rate (b) Energy Per Packet

Figure 14: Performance varying type of jamming.

stays near zero to a few percent. At 25 m, MJRLoRa achieves 50.74% PRR compared to 2.38% for LoRa. At the farthest evaluated node distance of 300 m in this three-jammer setup, MJRLoRa achieves 43.7% PRR while LoRa achieves 0.41%, yielding a PRR improvement of 106.75 times (43.7% vs 0.41%). At the same point, MJRLoRa requires 4.30 mJ per packet while LoRa requires 625 mJ, yielding an EPP reduction of 145.25 times (4.30 mJ vs 625 mJ).



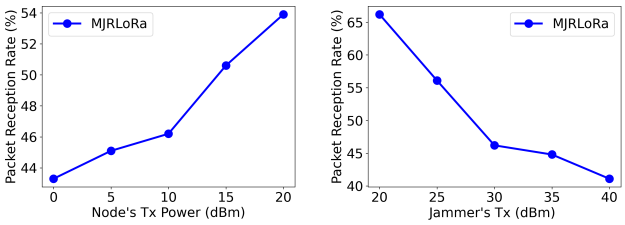
(a) Packet Reception Rate (b) Energy Per Packet

Figure 13: Performance varying Tx of node.

6.2.4 Varying Transmission Power of Jammers. For this setup, we vary jammer transmission power from 20 dBm to 40 dBm while the LoRa node uses 10 dBm. As shown in Figure 12(a), PRR declines for both MJRLoRa and LoRa as jammer power increases. The key point is that MJRLoRa still sustains PRR above 40% even at 40 dBm jammer power, whereas LoRa drops to 0.32%. At the highest evaluated jammer power of 40 dBm in this three-jammer setup, MJRLoRa achieves 42.29% PRR while LoRa achieves 0.32%, yielding a PRR improvement of 137.67 times (42.29% vs 0.32%). At the same point, MJRLoRa's EPP is 4.34 mJ while LoRa's EPP is 798.57 mJ, yielding an EPP reduction of 184.01 times (4.34 mJ vs 798.57 mJ).

6.2.5 Varying Transmission Power of Node. For this setup, we vary the LoRa node transmission power from 0 dBm to 20 dBm while the jammers use 30 dBm. As illustrated in Figure 13(a), PRR increases for both MJRLoRa and LoRa as the desired signal becomes stronger. MJRLoRa reaches 53.42% PRR at 20 dBm, compared with 2.18% for LoRa. At the weakest desired-signal point of 0 dBm, the relative gain is larger because the LoRa baseline degrades more sharply.

6.2.6 Varying Type of Jamming. In this setup, we vary the type of jamming among reactive, proactive, and random. To mimic a proactive jammer, we schedule the jamming waveform to begin together with the LoRa transmission; for reactive jamming, the waveform is triggered after packet detection; and for random jamming, it starts at random times. Figure 14 shows that MJRLoRa maintains similar PRR and EPP for reactive and proactive jamming, while LoRa performs worse against proactive jamming because more packets are suppressed before normal reception can proceed. MJRLoRa performs somewhat better against random jamming because the jammer waveforms often do not fully overlap the LoRa packet.



(a) Varying Node's Tx (b) Varying Jammer's Tx

Figure 15: Performance varying the channel of jammers.

6.2.7 Varying Channel of Jammers. In this setup, the jammers move across channels to jam different channels over time while still attacking only one channel at a time. Since all nodes transmit on different channels, each node is affected only when the jammer selects its channel. Figure 15 shows behavior broadly consistent with Figure 13(a) and Figure 12(a), indicating that the design remains effective when the attack target changes over time.

6.3 Discussion

From the experimental results, it is clear that our approach effectively mitigates collaborative jamming. The packet reception rate (PRR) stays between 40% and 70% across scenarios. Although this may appear low, it is a major improvement over existing techniques, which often achieve no more than 1% PRR in critical real deployments, causing unnecessary battery drain in LoRa nodes. Our PRR is computed at the MAC layer using CRC checks, where packets with symbol errors above the coding rate threshold are discarded.

In the future, we plan to develop a more robust anti-jamming strategy and extend our method to scenarios where jammers collaboratively target multiple channels at once. This introduces challenges such as multi-channel detection and simultaneous known-signal transmission, requiring further research.

7 Conclusion

Jamming poses a significant threat to low-power wide-area network (LPWAN) communications due to their reliance on centralized gateways. In this paper, we have addressed the vulnerability of LPWAN, specifically LoRa, to wireless jamming attacks by proposing a new anti-jamming method. It entails transmitting a known signal orthogonal to the LoRa signal on the same channel to disentangle and subtract combined jamming signals, thereby recovering the original packet. This method is link layer-agnostic, entails no overhead at the LoRa nodes, and enables packet decoding even when facing attacks from a single jammer or multiple jammers on a channel. We have implemented and evaluated our anti-jamming system using USRP and COTS LoRa devices in outdoor experiments. Results demonstrate up to a 100× improvement in packet reception (40.7% vs 0.4%) and a 136× reduction in energy consumption (4.52 mJ vs 615 mJ) compared to standard LoRaWAN in the harshest three-jammer outdoor setup with 50 m average jammer distance to the gateway.

Acknowledgments

The work was supported by the US National Science Foundation through grants CNS-2601685 and CNS-2602744, and by the US Office of Naval Research through grant N00014-23-1-2151.

References

[1] Emekcan Aras, Nicolas Small, Gowri Sankar Ramachandran, Stéphane Delbruel, Wouter Joosen, and Danny Hughes. 2017. Selective jamming of LoRaWAN using commodity hardware. In *MobiQuitous*. 363–372.
[2] Arduino. [n.d.]. Arduino Uno Rev3.
[3] Roberta Daidone, Gianluca Dini, and Marco Tiloca. 2014. A Solution to the GTS-based Selective Jamming Attack on IEEE 802.15.4 Networks. *Wirel. Netw.* 20, 5 (July 2014), 1223–1235.
[4] Adwait Dongare, Revathy Narayanan, Akshay Gadre, Anh Luong, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. 2018. Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks. In *IPSN*. IEEE, 60–71.
[5] Salvatore d'Oro, Laura Galluccio, Giacomo Morabito, Sergio Palazzo, Lin Chen, and Fabio Martignon. 2014. Defeating jamming with the power of silence: A game-theoretic analysis. *IEEE transactions on wireless communications* 14, 5 (2014), 2337–2352.
[6] Dragino. [n.d.]. GPS/LoRa Shield.
[7] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. 2017. Empowering low-power wide area networks in urban settings. In *SIGCOMM*. 309–321.
[8] Ettus Research. [n.d.]. USRP B210.
[9] GNU Radio Project. [n.d.]. GNU Radio.
[10] Kanika Grover, Alvin Lim, and Qing Yang. 2014. Jamming and anti–jamming techniques in wireless networks: a survey. *International Journal of Ad Hoc and Ubiquitous Computing* 17, 4 (2014), 197–215.
[11] Md Ashikul Haque, Aakriti Jain, Venkata Prashant Modekurthy, and Abusayeed Saifullah. 2026. Enabling Cross Technology Communication from LR-FHSS to LoRa. In *SenSys*. 1–13. https://doi.org/10.1145/3774906.3802771
[12] Md Ashikul Haque and Abusayeed Saifullah. 2023. A Game-Theoretic Approach for Mitigating Jamming Attacks in LPWAN. *EWSN* (2023).
[13] Md Ashikul Haque and Abusayeed Saifullah. 2024. Handling jamming attacks in a LoRa network. In *2024 IEEE/ACM Ninth International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, 146–157.
[14] Md Ashikul Haque and Abusayeed Saifullah. 2025. Mitigating Jamming Attacks in LoRa Networks: A Defense Strategy against LoRa-Based Jammers. In *MobiHoc*. 51–60.
[15] Md Ashikul Haque, Abusayeed Saifullah, and Haibo Zhang. 2025. Deep reinforcement learning based coexistence management in lpwan. In *IEEE INFOCOM 2025-IEEE Conference on Computer Communications*. IEEE, 1–10.
[16] Ningning Hou, Xianjin Xia, and Yuanqing Zheng. 2021. Jamming of LoRa PHY and countermeasure. In *INFOCOM*. IEEE, 1–10.
[17] Chin-Ya Huang, Ching-Wei Lin, Ray-Guang Cheng, Shanchieh Jay Yang, and Shiann-Tsong Sheu. 2019. Experimental evaluation of jamming threat in LoRaWAN. In *VTC2019-Spring*. IEEE, 1–6.
[18] IEEE Spectrum. [n.d.]. Satellite Jamming.
[19] Aakriti Jain, Md Ashikul Haque, Abusayeed Saifullah, and Haibo Zhang. 2024. Burst-MAC: A MAC Protocol for Handling Burst Traffic in LoRa Network. In *2024 IEEE Real-Time Systems Symposium (RTSS)*. 148–160. https://doi.org/10.1109/RTSS62706.2024.00022
[20] Loukas Lazos, Sisi Liu, and Marwan Krunz. 2009. Mitigating Control-channel Jamming Attacks in Multi-channel Ad Hoc Networks. In *WiSec*. 169–180.
[21] Chenning Li, Hanqing Guo, Shuai Tong, Xiao Zeng, Zhichao Cao, Mi Zhang, Qiben Yan, Li Xiao, Jiliang Wang, and Yunhao Liu. 2021. Nelora: Towards ultra-low snr lora communication with neural-enhanced demodulation. In *SenSys*. 56–68.
[22] LoRa Alliance. [n.d.]. LoRaWAN.
[23] Konstantin Mikhaylov, Radek Fujdiak, Ari Pouttu, Voznak Miroslav, Lukas Malina, and Petr Mlynek. 2019. Energy attack in LoRaWAN: Experimental validation. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*. 1–6.
[24] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou. 2009. A survey on jamming attacks and countermeasures in WSNs. *IEEE Communications Surveys Tutorials* 11, 4 (2009), 42–56.
[25] Overhaul. 2023. Mexico Q1-2023 Cargo Theft Report.
[26] Konstantinos Pelechrinis, Marios Iliofotou, and Srikanth V Krishnamurthy. 2010. Denial of service attacks in wireless networks: The case of jammers. *IEEE Communications surveys & tutorials* 13, 2 (2010), 245–257.
[27] K. Pelechrinis, C. Koufogiannakis, and S. V. Krishnamurthy. 2009. Gaming the jammer: Is frequency hopping effective?. In *2009 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*. 1–10.
[28] Hossein Pirayesh and Huacheng Zeng. 2022. Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey. *IEEE communications surveys & tutorials* 24, 2 (2022), 767–809.
[29] A. Proano and L. Lazos. 2012. Packet-Hiding Methods for Preventing Selective Jamming Attacks. *IEEE Transactions on Dependable and Secure Computing* 9, 1 (2012), 101–114.
[30] D. R. Raymond and S. F. Midkiff. 2008. Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses. *IEEE Pervasive Computing* 7, 1 (2008), 74–81.
[31] Abusayeed Saifullah, Mahbubur Rahman, Dali Ismail, Chenyang Lu, Ranveer Chandra, and Jie Liu. 2016. SNOW: Sensor Network over White Spaces. In *SenSys*.
[32] A. Saifullah, M. Rahman, D. Ismail, C. Lu, J. Liu, and R. Chandra. 2018. Low-Power Wide-Area Network Over White Spaces. *IEEE/ACM Transactions on Networking* 26, 4 (Aug 2018), 1893–1906. https://doi.org/10.1109/TNET.2018.2856197
[33] Semtech. [n.d.]. LoRa Modem Design Guide.
[34] Statista. 2021. Number of Internet of Things (IoT) Connected Devices.
[35] Shuai Tong, Zhenqiang Xu, and Jiliang Wang. 2020. Colora: Enabling multi-packet reception in lora. In *INFOCOM*. IEEE, 2303–2311.
[36] Xiong Wang, Linghe Kong, Liang He, and Guihai Chen. 2019. mlora: A multi-packet reception protocol in lora networks. In *ICNP*. IEEE, 1–11.
[37] Anthony D Wood, John A Stankovic, and Gang Zhou. 2007. DEEJAM: Defeating energy-efficient jamming in IEEE 802.15. 4-based wireless networks. In *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE, 60–69.
[38] Xianjin Xia, Qianwu Chen, Ningning Hou, Yuanqing Zheng, and Mo Li. 2023. XCopy: Boosting Weak Links for Reliable LoRa Communication. *MobiCom* (2023).
[39] Xianjin Xia, Yuanqing Zheng, and Tao Gu. 2019. FTrack: Parallel decoding for LoRa transmissions. In *SenSys*. 192–204.
[40] Dejun Yang, Guoliang Xue, Jin Zhang, Andrea Richa, and Xi Fang. 2013. Coping with a smart jammer in wireless networks: A Stackelberg game approach. *IEEE Transactions on Wireless Communications* 12, 8 (2013), 4038–4047.