Attack-Aware Synchronization-Free Data Timestamping in LoRaWAN

CHAOJIE GU, LINSHAN JIANG, RUI TAN, and MO LI, Nanyang Technological University, Singapore

JUN HUANG, Massachusetts Institute of Technology, United States

Low-power wide-area network technologies such as LoRaWAN are promising for collecting low-rate monitoring data from geographically distributed sensors, in which timestamping the sensor data is a critical system function. This paper considers a synchronization-free approach to timestamping LoRaWAN uplink data based on signal arrival time at the gateway, which well matches LoRaWAN's one-hop star topology and releases bandwidth from transmitting timestamps and synchronizing end devices' clocks at all times. However, we show that this approach is susceptible to a *frame delay attack* consisting of malicious frame collision and delayed replay. Real experiments show that the attack can affect the end devices in large areas up to about $50,000\,\mathrm{m}^2$. In a broader sense, the attack threatens any system functions requiring timely deliveries of LoRaWAN frames. To address this threat, we propose a LoRaTS gateway design that integrates a commodity LoRaWAN gateway and a low-power software-defined radio receiver to track the inherent frequency biases of the end devices. Based on an analytic model of LoRa's chirp spread spectrum modulation, we develop signal processing algorithms to estimate the frequency biases with high accuracy beyond that achieved by LoRa's default demodulation. The accurate frequency bias tracking capability enables the detection of the attack that introduces additional frequency biases. We also investigate and implement a more crafty attack that uses advanced radio apparatuses to eliminate the frequency biases. To address this crafty attack, we propose a pseudorandom interval hopping scheme to enhance our frequency bias tracking approach. Extensive experiments show the effectiveness of our approach in deployments with real affecting factors such as temperature variations.

CCS Concepts: • **Networks** → *Sensor networks*; • **Security and privacy** → *Distributed systems security*.

Additional Key Words and Phrases: Low-power wide-area networks, LoRaWAN, data timestamping, wireless security.

ACM Reference Format:
Chaojie Gu, Linshan Jiang, Rui Tan, Mo Li, and Jun Huang. 2019. Attack-Aware Synchronization-Free Data Timestamping in LoRaWAN. *ACM Trans. Sensor Netw.* 1, 1, Article 1 (January 2019), 32 pages. https://doi.org/0

1 INTRODUCTION

Low-power wide-area networks (LPWANs) enable direct wireless interconnections among end devices and gateways in geographic areas of square kilometers. It increases network connectivity

as a defining characteristic of the Internet of Things (IoT). Among various LPWAN technologies (including NB-IoT and Sigfox), LoRaWAN, which is an open data link layer specification based on the LoRa modulation scheme [31], offers the advantages of using license-free ISM bands, low costs for end devices, and independence from managed cellular infrastructures.

LoRaWAN is promising for the applications of collecting low-rate monitoring data from geographically distributed sensors, such as utility meters, environment sensors, roadway detectors, industrial measurement devices, etc. All these applications require data timestamping as a basic system service, though they may require different timestamp accuracies. For instance, data center environment condition monitoring generally requires sub-second accuracy for sensor data timestamps to capture the thermodynamics [14]. Sub-second-accurate timestamps for the traffic data generated by roadway detectors can be used to reconstruct real-time traffic maps [26]. In a range of industrial monitoring applications such as oil pipeline monitoring, milliseconds accuracy may be required [3]. In volcano monitoring, the onset times of seismic events detected by geographically distributed sensors require sub-10 milliseconds accuracy to be meaningful to volcanic earthquake hypocenter estimation [24].

There are two basic approaches, namely, *sync-based* and *sync-free*, to data timestamping in wireless sensor networks (WSNs). In the sync-based approach, the sensor nodes keep their clocks synchronized and use the clock value to timestamp the data once generated. Differently, the sync-free approach uses the gateway with wall time to timestamp the data upon the arrival of the corresponding network frame. Based on various existing distributed clock synchronization protocols, multi-hop WSNs mostly adopt the sync-based approach. The sync-free approach is ill-suited for multi-hop WSNs, because the data delivery on each hop may have uncertain delays due to various factors such as channel contention among nodes.

In contrast, LoRaWANs prefer the sync-free approach for uplink data timestamping. Reasons are two-fold. First, different from multi-hop WSNs, LoRaWANs adopt a one-hop gateway-centered star topology that is free of the issue of hop-wise uncertain delays. Specifically, as the radio signal propagation time from an end device to the gateway is generally in microseconds, the LoRaWAN frame arrival time can well represent the time when the frame leaves the end device. As a result, timestamping the uplink data at the gateway can meet the milliseconds or sub-second timestamping accuracy requirements of many applications. Second, if the sync-based approach is adopted otherwise, the task of keeping the end devices' clocks synchronized at all times and the inclusion of timestamps in the LoRaWAN data frames will introduce communication overhead to the narrowband LoRaWANs (a detailed analysis can be found in §3.2). Therefore, performance-wise, the sync-free approach well matches LoRaWANs' star topology and addresses its bandwidth scarcity.

However, LoRaWAN's long-range communication capability also renders itself susceptible to wireless attacks that can be launched from remote and hidden sites. The attacks may affect many end devices in large geographic areas. In particular, the conventional security measures that have been included in the LoRaWAN specifications (e.g., frame confidentiality and integrity) may be inadequate to protect the network from wireless attacks on the physical layer. Therefore, it is of importance to study the potential wireless attacks against the sync-free data timestamping, since inaccurate and even incorrect timestamps significantly undermine the value of the data. For example, when applying LoRa for transmitting detected earthquake events, tiny timestamping errors will lead to earthquake hypocenter estimation errors [24]. In this paper, we consider a basic threat of *frame delay attack* that directly invalidates the assumption of near-zero signal propagation time. Specifically, by setting up a *collider* device close to the LoRaWAN gateway and an *eavesdropper* device at a remote location, a combination of malicious frame collision and delayed replay may introduce arbitrary delays to the deliveries of uplink frames. Although wireless jamming and replay have been studied extensively, how easily they can be launched in a coordinated manner

to introduce frame delay and how much impact (e.g., in terms of the affected area) the attack can generate are still open questions in the context of LoRaWANs.

This paper answers these questions via real experiments. Our measurements show that LoRa demodulators have lengthy vulnerable time windows, in which the gateway cannot decode either the victim frame or the collision frame, and raises no alerts. Thus, it is easy to launch stealthy attacks by exploiting the vulnerable time windows. In particular, as the attack does not breach the integrity of the frame content and sequence, the attack cannot be solved by cryptographic protection and frame counting. Our experiments in a campus LoRaWAN show that, a fixed setup of a collider and an eavesdropper can subvert the sync-free data timestamping service for end devices in a large geographic area of about $50,000\,\mathrm{m}^2$. In a broader sense, this attack threatens any system functions that require timely deliveries of uplink frames in LoRaWAN. Note that this attack is valid but marginally important in short-range wireless networks (e.g., Zigbee and Wi-Fi) because of the limited area affected by the attack and the difficulty in controlling the attack radios' timing. Differently, it is important to LoRaWANs because it can affect large geographic areas, and the timing of the attack radios can be easily controlled due to LoRaWAN's long symbol times.

Therefore, an upgraded sync-free timestamping approach that integrates countermeasures against the attack and meanwhile preserves the bandwidth efficiency is desirable. Moreover, it should only require changes to the gateway. In this paper, we aim to develop an awareness of the attack by monitoring the end devices' radio frequency biases (FBs). Due to the manufacturing imperfections of the radio chips' internal oscillators, each radio chip generally has an FB that is the difference between the frequency of the carrier signal emitted by the chip and the nominal value. A change of FB detected by the gateway suggests the received frame may be a replayed one since the adversary's replay device superimposes its own FB onto the replayed signal. To access the physical layer, we integrate a low-cost (US\$25) software-defined radio (SDR) receiver [4] with a commodity LoRaWAN gateway to form our LoRa TimeStamping (LoRaTS) gateway. We develop time-domain signal processing algorithms for LoRaTS to estimate the FB. Experiments show that (i) with a received signal-to-noise ratio (SNR) of down to $-18\,\mathrm{dB}$, LoRaTS achieves an accuracy of $120\,\mathrm{Hz}$ in estimating FB, which is just 0.14 parts-per-million (ppm) of the channel's central frequency of $869.75\,\mathrm{MHz}$; (ii) the frame replay by an SDR transceiver introduces an additional FB of at least 0.24 ppm. Thus, LoRaTS can track FB to detect the replay step of the frame delay attack. Note that the detection does not require uniqueness or distinctiveness of the FBs across different LoRa transceivers because it is based on changes of FB.

In summary, LoRaTS supports the bandwidth-efficient sync-free timestamping and requires no modifications on the LoRaWAN end devices. It is a low-cost countermeasure that increases the cost and technical barrier for launching effective frame delay attacks since the attackers need to eliminate the tiny FBs of their radio apparatuses. LoRaTS strikes a satisfactory trade-off between network efficiency and the security level required by typical LoRaWAN applications. If the attackers employ expensive radio apparatuses to eliminate the tiny FBs, we further develop an approach based on the pseudorandom number generator to counteract the zero-FB attack. With the zero-FB attack countermeasure deployed, although the end device needs to follow a transmission schedule determined by the pseudorandom number generator, the security level of the system is further improved.

This paper makes the following contributions:

- We implement the frame delay attack against LoRaWANs. Simulations and experiments show the large sizes of the geographic areas vulnerable to the attack.
- Based on an analytic model of LoRa's chirp spread spectrum (CSS) modulation, we design a time-domain signal processing pipeline to accurately estimate end devices' FBs.

- Extensive experiments in indoor and outdoor environments show that LoRaTS can detect the frame delay attacks that introduce additional FBs.
- We implement the zero-FB attack and propose a Pseudorandom Interval Hopping scheme to counteract the zero-FB attack.

The rest of this paper is organized as follows. §2 reviews related work; §3 describes sync-free data timestamping; §4 studies the attack; §5 presents LoRaTS and uplink frame arrival time detection approach; §6 studies LoRa's FB and uses it to detect attack; §7 studies zero-FB attack and countermeasure; §8 presents experiment results; §9 concludes this paper.

2 RELATED WORK

Improving LoRaWAN's communication performance has received increasing research. The Choir system [18] exploits the diverse FBs of the LoRaWAN end devices to disentangle colliding frames from different end devices. Choir uses the dechirping and Fourier transform processing pipeline to analyze FB, which does not provide sufficient resolution for detecting the tiny extra FB introduced by attack (see details in §6.1). In this paper, based on an analytic model of LoRa's CSS modulation, we develop a new time-domain signal processing algorithm based on a least squares formulation to achieve the required resolution. The Charm system [16] exploits *coherent combining* to decode a frame from the weak signals received by multiple geographically distributed LoRaWAN gateways. It allows the LoRaWAN end device to use a lower transmitting power. Several recent studies [20, 27] have devised various backscatter designs for LoRa to reduce the power consumption of end devices. All the studies mentioned above focus on understanding and improving the data communication performance of LoRaWAN [16, 18], or reducing power consumption via backscattering [20, 27]. None of them specifically addresses efficient data timestamping, which is a basic system function of many LoRaWAN-based systems.

LongShoT [28] is an approach to synchronize the LoRaWAN end devices with the gateway. Through low-level offline time profiling for a LoRaWAN radio chip (e.g., to measure the time delays between hardware interrupts and the chip's power consumption rise), LongShoT achieves sub-50 microseconds accuracy, which is echoed by our results on the accuracy of estimating signal arrival time using a different approach. LongShoT is designed for the LoRaWAN systems requiring tight clock synchronization. Differently, we address data timestamping and focus on the less stringent but more commonly seen milliseconds or sub-second accuracy requirements. Our sync-free approach releases the bandwidth from frequent clock synchronization operations.

Security of LoRaWAN is receiving research attention. In [11], Aras et al. discuss several possible attacks against LoRaWAN, including key compromise and jamming. The key compromise requires prior physical attack of memory extraction. In [12], a selective jamming attack against certain receivers and/or certain application frames is studied. Different from the studies [11, 12] that do not consider the stealthiness of jamming, we consider stealthy frame collision. From our results in §4.2, the selective jamming in [12] cannot be stealthy because it cannot start jamming until the frame header is decoded and the corruption of payload must lead to integrity check failures. In [29], Robyns et al. apply supervised machine learning for end device classification based on the received LoRa signal. From our measurements, the dissimilarity between the original and the replayed signals is much lower than that among the original signals from different end devices. Thus, the supervised machine learning is not promising for attack detection.

Device identification based on radiometric features has been studied for short-range wireless technologies. A radiometric feature is the difference between the nominal and the measured values of a certain modulation parameter. The work [13] studied the radiometric features of IEEE 802.11 radios, including symbol-level features regarding signal magnitude and phase, as well as the

frame-level feature regarding carrier frequency. In LoRaWAN, the received signal strength is often rather low due to long-distance propagation or barrier penetration. As such, the signal magnitude radiometric feature cannot be used as a radiometric feature. As the phase of LoRa signal is arbitrary, it cannot be employed as a radiometric feature too. In this paper, we show that the bias of the LoRa signal's carrier frequency from the nominal value is an effective radiometric feature. This feature can be used to counteract the frame delay attack. Based on LoRa's CSS modulation, we develop a lightweight algorithm that can estimate this feature from the received LoRa signal. It requires a low-cost SDR receiver, unlike the expensive vector signal analyzer [5] used in the work [13].

3 DATA TIMESTAMPING IN LORAWAN

3.1 LoRaWAN Primer

LoRa is a physical layer technique that adopts CSS modulation. LoRaWAN is an open data link specification based on LoRa. A LoRaWAN is a star network consisting of a number of *end devices* and a *gateway* that is often connected to the Internet. Gateways are often equipped with GPS receivers for time keeping. The transmission direction from the end device to the gateway is called *uplink* and the opposite is called *downlink*. LoRaWAN defines three classes for end devices, i.e., Class A, B and C. In Class A, each communication session must be initiated by an uplink transmission. There are two subsequent downlink windows. Class A end devices can sleep to save energy when there are no pending data to transmit. Class A adopts the ALOHA media access control protocol. Class B extends Class A with additional scheduled downlink windows. However, such scheduled downlink windows require the end devices to have synchronized clocks, incurring considerable overhead as we will analyze shortly. Class C requires the end devices to listen to the channel all the time. Clearly, Class C is not for low-power end devices. In this paper, we focus on Class A, because it is supported by all commodity platforms and energy-efficient. To the best of our knowledge, no commodity platforms have out-of-the-box support for Class B that requires clock synchronization.

3.2 Advantages of Sync-Free Timestamping

Data timestamping, i.e., to record the *time of interest* in terms of the wall clock, is a basic system function required by the data collection applications for monitoring. For a sensor measurement, the time of interest is the time instant when the measurement is taken by the end device. Multi-hop WSNs largely adopt the *sync-based* approach. Specifically, the clocks of the WSN nodes are synchronized to the global time using some clock synchronization protocol. Then, each WSN node can timestamp the data using its local clock. WSNs have to adopt this approach due primarily to that the multi-hop data deliveries from the WSN nodes to the gateway in general suffer uncertain delays. Thus, although the clock synchronization introduces additional complexity to the system implementation, it has become a standard component for systems requiring data timestamping. However, the clock synchronization introduces considerable communication overhead to the bandwidth-limited LoRaWANs.

We present an example to illustrate the overhead to maintain sub-10 milliseconds (ms) clock accuracy in LoRaWANs. Typical crystal oscillators in microcontrollers have drift rates of 30 to 50 ppm [19]. Without loss of generality, we adopt 40 ppm for this example. With this drift rate, an end device needs 14 synchronization sessions per hour to maintain sub-10 ms clock accuracy. These 14 sessions represent a significant communication overhead for an end device. For instance, in Europe, a LoRaWAN end device adopting a spreading factor of 12 can only send 24 30-byte frames per hour to conform to the 1% duty cycle requirement [21]. Although the synchronization information may be piggybacked to the data frames, a low-rate monitoring application may have to send the frames more frequently just to keep time. In addition, the data frames need to include data
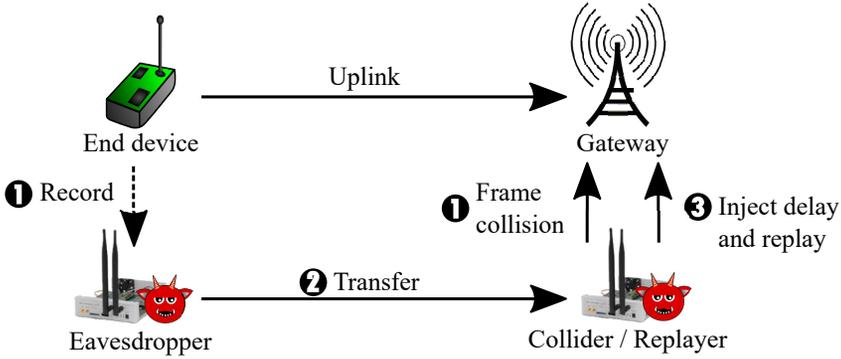
Fig. 1. Steps for implementing frame delay attack.

timestamps, each of which needs at least a few bytes. This is also an overhead given the bandwidth scarcity.

To efficiently utilize LoRaWAN's scarce bandwidth and exploit its star topology, the sync-free timestamping approach can be adopted. In this approach, an end device transmits a sensor reading once generated. Upon receiving the frame, the gateway uses the frame arrival time as the data timestamp. The signal propagation time from the end device to the gateway, which is often microseconds, can be ignored for millisecond-accurate timestamping. Compared with the sync-based approach, this sync-free approach avoids the communication overhead caused by the frequent clock synchronization operations and the transmissions of timestamps. Thus, the sync-free approach is simple and provides bandwidth-saving benefit throughout the lifetime of the LoRaWANs.

4 SECURITY OF SYNC-FREE TIMESTAMPING

The long-range communication capability of LoRaWAN enables the less complex and bandwidth-efficient sync-free timestamping. However, it may also be subject to wireless attacks that can affect large geographic areas. Having understood the benefit of sync-free timestamping, we also need to understand its security risk and the related countermeasure for achieving a more comprehensive assessment on the efficiency-security tradeoff. A major and direct threat against the sync-free approach is the *frame delay attack* that manipulates the frame delivery time to invalidate the assumption of near-zero signal propagation delay. We define the attack as follows.

Frame delay attack: The end device and gateway are not corrupted by the adversary. However, the adversary may delay the deliveries of the uplink frames. The malicious delay for any uplink frame is finite. Moreover, the frame cannot be tampered with because of cryptographic protection.

The attack results in wrong timestamps under the sync-free approach. This section studies the attack implementation (§4.1), investigates the timing of malicious frame collision (§4.2), and studies the size of the vulnerable area in which the end devices are affected by the attack (§4.3).

4.1 Attack Implementation

4.1.1 Implementation steps. Fig. 1 illustrates the attack implementation. The adversary sets up two malicious devices called *eavesdropper* and *collider* that are close to the end device and the gateway, respectively. The attack consists of three steps. ❶ At the beginning, both the eavesdropper and the collider listen to the LoRa communication channel between the end device and the gateway. Once the collider detects an uplink frame transmission, it transmits a collision frame. In §4.2, we
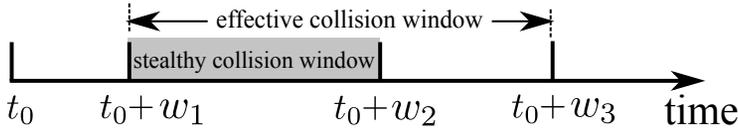
Fig. 2. Collision attack time window.

will investigate experimentally a stealthy collision method such that the victim gateway does not raise any warning message to the application layer. Meanwhile, once the eavesdropper detects an uplink frame transmission, it records the radio waveform of the frame. Note that the collider may choose a proper transmitting power of the collision frame such that the collision can affect the victim gateway, while not corrupting the radio waveform recorded by the eavesdropper. ❷ The eavesdropper sends the recorded radio waveform data to the collider via a separate communication link that provides enough bandwidth. ❸ After a time duration of τ seconds from the onset time of the victim frame transmission, the collider replays the recorded radio waveform. Thus, in this paper, the *collider* and the *replayer* refer to the same attack device. The above collision-and-replay process does not need to decipher the payload of the recorded frame; it simply re-transmits the recorded radio waveform. As the gateway cannot receive the original frame and the integrity of the replayed frame is preserved, the gateway accepts the replayed frame even if it checks the frame integrity and frame counter. The attack introduces a delay of τ seconds to the delivery of the frame.

We discuss several issues in the attack implementation. First, using a normal LoRaWAN frame to create malicious collision is more stealthy than brute-force jamming, since it may be difficult to differentiate malicious and normal collisions. Brute-force jamming can be easily detected and located. Second, as the adversary delays the uplink frame, how does the adversary know in time the direction of the current transmission? In LoRaWAN, the uplink preamble uses up chirps, whereas the downlink preamble uses down chirps. Thus, the adversary can quickly detect the direction of the current transmission within a chirp time. From our results in §4.2, a time duration of one chirp for sensing the direction of the transmission does not impede the timeliness of the collision attack. Third, to increase the stealthiness of the replay attack, the replayer can well control the transmitting power of the replay such that only the victim gateway can receive the replayed frame. Fourth, the attack does not require clock synchronization between the eavesdropper and the collider.

4.1.2 Discussion on a simple attack detector. A simple attack detection approach is to perform round-trip timing and then compare the measured round-trip time with a threshold. However, this approach has the following three shortcomings. First, it needs a downlink transmission for each uplink transmission, which doubles the communication overhead. LoRaWAN is mainly designed and optimized for uplinks. For instance, a LoRaWAN gateway can receive frames from multiple end devices simultaneously using different spreading factors, whereas it can send a single downlink frame only at a time. This is because Class A specification requires that any downlink transmission must be unicast, in response to a precedent uplink transmission. Thus, the round-trip timing approach matches poorly with the uplink-downlink asymmetry characteristic of LoRaWAN. Second, with this simple attack detection approach, it is the end device detecting the attack after receiving the downlink acknowledgement. The end device needs to inform the gateway using another uplink frame that is also subject to malicious collision. Third, as the attacks are rare (but critical) events, continually using downlink acknowledgements to preclude the threat is a low cost-effective solution. In summary, this simple round-trip timing countermeasure is inefficient and error-prone.

4.2 Timing of Malicious Frame Collision

In this section, we study the timing of effective malicious frame collision. When investigating the geographic area affected by the attack, the ratio between the powers of the victim signal and the collision signal also needs to be considered. §4.3 will jointly consider the collision timing and the signal power ratio. We set up two SX1276-based LoRa nodes as the transmitter and the receiver, which are separated by about 5 m. We use a third LoRa node as the collider against the receiver. The distance between the collider and the receiver is about 1 m. Although the quantified results obtained based on SX1276 are chip specific, the qualitative results (i.e., the trend) are consistent with the general understanding on wireless demodulation. Thus, the qualitative results provide general insights and implications. The gateway-class iC880A LoRaWAN concentrator and an open-source LoRa demodulator that we use in §4.3 also exhibit similar trend. In practice, the adversary may conduct experiments similar to those presented below to obtain the required attack timing once they know the model of the victim LoRa chip.

From our experiments, there are three critical time windows (denoted by w_1, w_2, and w_3) after the onset time of the victim transmission (denoted by t_0). These time windows are illustrated in Fig. 2. If the onset time of the collision frame is in $[t_0, t_0 + w_1]$, the receiver most likely receives the collision frame only; if it is in $[t_0 + w_1, t_0 + w_2]$, the receiver receives neither frame and raises no alerts; if it is in $[t_0 + w_2, t_0 + w_3]$, the receiver reports "bad frame" and yields no frame content; if it is after $t_0 + w_3$, the receiver can receive both frames sequentially. Therefore, the time window $[t_0 + w_1, t_0 + w_2]$ is called *stealthy collision window* and the $[t_0 + w_1, t_0 + w_3]$ is called *effective collision window*. Note that we view the "bad frame" situation as effective attack, because the receiver cannot differentiate malicious and normal collisions based on the warning message.

Our experiments measure w_1, w_2, and w_3 under a wide range of settings including spreading factor and the payload size. Table 1 summarizes the results. From the results for w_1, the collision should start after the 5th chirp of the victim frame transmission. Explanation is as follows. (Note that as the demodulation mechanism of used SX1276 is proprietary and not publicly available, our explanations in this section are based on general understanding on wireless demodulation.) First, the receiver has not locked the victim frame's preamble until the 6th chirp. If the collision starts before the 5th chirp of the victim frame, the receiver will re-lock the collision frame's preamble with higher signal strength, resulting in reception of the collision frame. Second, the receiver locks the victim frame's preamble from the 6th chirp and simply drops any received radio data without reporting any error if any of the last three chirps (i.e., the 6th, 7th, and 8th chirps) of the preamble and/or the frame header are corrupted. For the latter case of frame header corruption, the radio chip cannot determine whether itself is the intended recipient and hence drops the received data. Thus, the collision should start after the 5th chirp of the victim frame.

We can also see that w_2 increases exponentially with the spreading factor. This is because: i) the total time for transmitting the preamble and frame header increases exponentially with the spreading factor; ii) corruption of the payload after the frame header leads to integrity check error and the "bad frame" message. The w_3 is roughly the time for transmitting the victim frame. Thus, if the collision onset time is after $t_0 + w_3$, both the victim and collision frames can be received.

The above experiments show that, there is a time window of more than 20 ms for the collision to corrupt the preamble partially and the frame header such that the victim simply drops the received data and raises no alerts. Collision starting in this window is stealthy. There is also an effective attack window of more than 100 ms. It is not difficult to satisfy such timing requirements using commodity radio devices.

Table 1. Collision time windows for SX1276.

Spreading factor S	Chirp time	Preamble time	Payload (byte)	w_1	w_2	w_3
7	1.024	8.2	10	5	28	141
			20	5	38	156
			30	6	41	165
			40	6	54	178
7	1.024	8.2		6	41	165
8	2.048	16.4	30	10	82	208
9	4.096	32.8		22	156	274

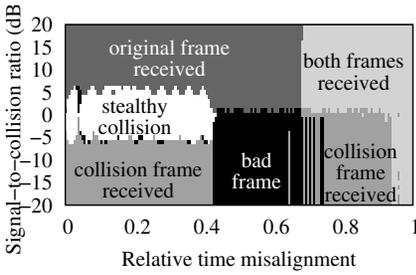* Unit for chirp time, preamble time, w_1, w_2, w_3 is millisecond.

Fig. 3. Result of `gr-lora`'s demodulation under collision with different signal-to-collision ratios and relative time misalignments.
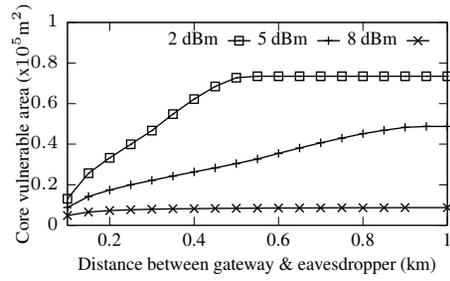

Fig. 4. Core vulnerable area vs. distance between gateway and eavesdropper under various collision powers.

4.3 Size of Vulnerable Area

In this section, through simulations and extensive experiments in a campus, we show that by setting up a collider and an eavesdropper at fixed locations, the frame delay attack can affect many end devices in a geographic area. The simulations based on realistic measurements with an open-source LoRa demodulator and a path loss model [15] provide insights into understanding the vulnerable area. The experiments in the campus further capture other affecting factors such as terrain and signal blockage from buildings. In this section, the *core vulnerable area* refers to the geographic area in which the end devices are subject to stealthy collision and successful eavesdropping; the *vulnerable area* additionally includes the area in which the end devices are subject to the collision causing "bad frame" reports and successful eavesdropping.

4.3.1 Simulations. To study the vulnerable area, we need to consider the signal path loss and the ratio between the powers of the victim signal and the collision signal at the receiver. We call this ratio *signal-to-collision ratio* (SCR). To characterize attack timing, we define *relative time misalignment* (RTM) as $\frac{\text{collision time lag}}{\text{frame time}}$, where the collision time lag is the time lag of the collision onset from the victim signal onset. In our simulation, the victim and collision frames have identical length but different payload contents. We generate the I and Q waveforms of these two frames using LoRa signal model. We superimpose the two frames' signals to simulate collision. Moreover, we scale the amplitudes of the two signals and time-misalign them to create certain SCR and RTM. The sum signal is processed using an open-source LoRa demodulator `gr-lora` [2]. Fig. 3 shows the
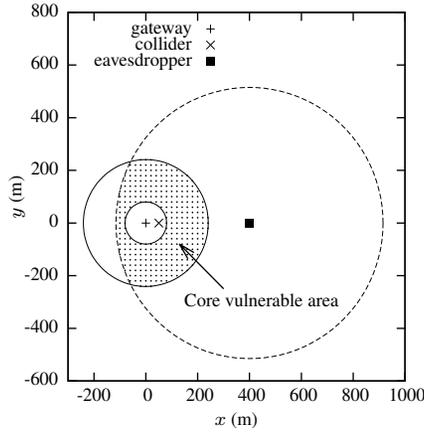
Fig. 5. The core vulnerable area (i.e., the shaded area) with gateway at $(0,0)$, collider at $(50,0)$, and eavesdropper at $(400,0)$. Collider's and victim end device's transmitting powers are $2\,\text{dBm}$ and $14\,\text{dBm}$, respectively. End devices in the ring centered at $(0,0)$ are subject to stealthy collision; end devices in the dashed circle are subject to successful eavesdropping.
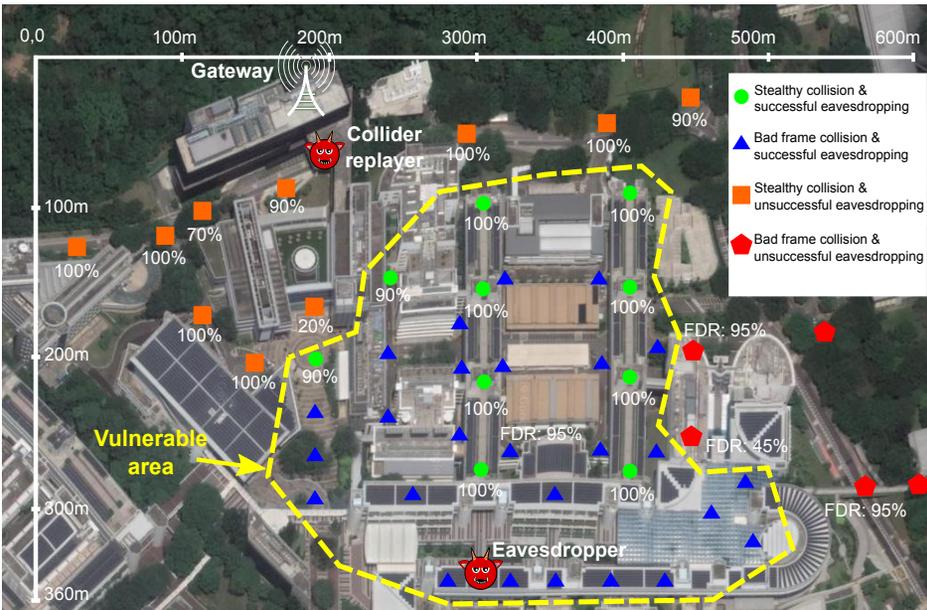

Fig. 6. Vulnerable area of a campus LoRaWAN. A gateway and a USRP-based eavesdropper are deployed on the rooftops of two buildings. A collider is deployed on an overhead bridge. We carry an end device to each of the marked locations and conduct an attack experiment. The four point shapes represent four types of attack outcomes. (Satellite image credit: Google Map)

demodulation results under various SCR and RTM settings. We can see that if RTM is less than 0.4 and SCR at the gateway is within $[-6\,\text{dB}, 6\,\text{dB}]$, the collision is stealthy. The eavesdropped frame can be demodulated if SCR at the eavesdropper is greater than $6\,\text{dB}$.

We adopt a LoRa signal path loss model for urban areas proposed in [15] based on real measurements. The details of the model can be found in [15]. The frame delay attack is successful if the attacker can control RTM below 0.4 and satisfy the following two conditions:

$$-6\,\text{dB} \leq P_v - L_{v,g} - (P_c - L_{c,g}) \leq 6\,\text{dB}, \tag{1}$$

$$6\,\text{dB} \leq P_v - L_{v,e} - (P_c - L_{c,e}), \tag{2}$$

where the subscripts v, g, c, and e respectively denote the victim end device, the gateway, the collider, and the eavesdropper; P_x denotes the transmitting power of device x; $L_{x,y}$ denotes the path loss from device x to y. Eq. (1) is the condition for stealthy collision; Eq. (2) is the condition for successful eavesdropping. The SCR thresholds of $6\,\text{dB}$ and $-6\,\text{dB}$ in Eqs. (1) and (2) are from Fig. 3. Note that our modeling of successful eavesdropping in Eq. (2) only considers the case that the signal from the collider at the eavesdropper has a power much higher than the noise floor, so that we can ignore the impact of noise on the eavesdropping.

Fig. 5 shows an example of the areas defined by Eqs. (1) and (2). The collider's and end device's transmitting powers are $2\,\text{dBm}$ and $14\,\text{dBm}$. The gateway's altitude is $25\,\text{m}$; the collider, eavesdropper, and end devices have an identical altitude of $0\,\text{m}$. As shown in Fig. 5, the ring centered at the gateway is defined by Eq. (1); the disk area in the dashed circle is defined by Eq. (2). Thus, the overlap between the ring and the disk is the core vulnerable area, which is $62,246\,\text{m}^2$. Then, we vary the distance between the gateway and the eavesdropper (denoted by d_{ge}) and the P_c setting. Fig. 4 shows the resulting core vulnerable area. We can see that the core vulnerable area in general increases with d_{ge} and becomes flat after d_{ge} exceeds a certain value. Moreover, among the three P_c settings (i.e., 2, 5, and 8 dBm), $P_c = 2\,\text{dBm}$ gives larger core vulnerable areas. Reason of the above two observations is that the eavesdropper can achieve a larger eavesdropping area due to the weaker collision signal received by the eavesdropper. The core vulnerable area saturates because the eavesdropping area in the dashed circle illustrated in Fig. 5 covers the entire ring area when d_{ge} exceeds a certain value. Note that when d_{ge} is very large, the noise power dominates and the core vulnerable area shrinks to zero.

The above simulation results suggest that the location of the gateway is the key information that the adversary needs to obtain. Based on that, the adversary can plan the placement of the collider and eavesdropper to affect a large geographic area. For the LoRaWANs adopting multiple gateways, the adversary can place a collider close to each of the gateways. In practice, the locations of the gateways can be obtained by the adversary in various ways (e.g., social engineering) and should not be relied on for the security of the system.

4.3.2 Experiments in a campus LoRaWAN. We conduct a set of experiments in an existing campus LoRaWAN to investigate the vulnerable area in real environments. The LoRaWAN's gateway covers the area shown in Fig. 6 that has a number of multistory buildings. The gateway, which consists of an iC880a LoRaWAN concentrator board, a Raspberry Pi, and a high-gain antenna, is located on the rooftop of a building. Both the collider and the eavesdropper consist of a laptop computer and a USRP N210 each. The collider is placed on an overhead bridge attached to the gateway's building. The horizontal distance between the gateway and the collider is about $50\,\text{m}$. The eavesdropper is placed on the rooftop of another building that is about $320\,\text{m}$ from the gateway's building. We carry an SX1276-based LoRaWAN end device to each of the locations marked in Fig. 6, measure the frame delivery ratio (FDR), and perform an attack experiment. The measured FDRs at all the visited locations are 100%, except the four locations labeled with non-100% FDRs. Thus, the gateway can cover the accessible area shown in Fig. 6.

In each attack experiment, the end device's and the collider's transmitting powers are $14\,\text{dBm}$ and $8\,\text{dBm}$, respectively. All malicious collisions are effective. The outcomes can be classified into four

categories, which are the combinations of the collision results (stealthy collision or "bad frame") and eavesdropping results (successful or unsuccessful). In Fig. 6, we use four point shapes to represent the four attack outcomes. The percentage below a location is the ratio of stealthy collisions. We can see that, at most locations close to the gateway and collider, the malicious collisions are stealthy. At the locations in the bottom most part of Fig. 6, the collisions cause gateway's bad frame reports. There is a transit region in the middle of Fig. 6, in which the collision outcomes are mixed. Note that the visited locations shown in Fig. 6 are on the rooftops, in semi-outdoor corridors, or in indoor environments. The indoor/outdoor condition may affect the collision outcome type. At the locations in the area enclosed by the dashed polygon, the gateway can decode the frame that is recorded by the eavesdropper and then replayed by the collider, suggesting that the eavesdropping is successful. Thus, this area is the vulnerable area caused by the attack setup, which is about $50,000\,\text{m}^2$.

Note that the demodulation mechanism of the iC880a concentrator is proprietary and can be different from the open-source LoRa demodulator we used in §4.3.1. The actual signal propagation behaviors in the campus LoRaWAN can be much more complex than the model used in §4.3. However, the simulation result (Fig. 5) and real experiment result (Fig. 6) show similar patterns, i.e., the eavesdropping area is around the eavesdropper and the core vulnerable area is a belt region between the gateway and the eavesdropper. Thus, our modeling and simulations in §4.3 provide useful understanding on the LoRaWAN vulnerability.

5 LoRaTS GATEWAY

As shown in §4, a fixed setup of a collider and an eavesdropper can subvert the sync-free timestamping for many end devices in a large geographic area. This section presents the LoRaTS gateway that supports the bandwidth-efficient sync-free timestamping as an advantage throughout the network lifetime and develops awareness of the frame delay attack.

5.1 LoRaTS Gateway Hardware

To detect the attack, we integrate an SDR receiver with a LoRaWAN gateway to monitor the physical layer. Various cheap (US$25 only [10]) and low-power SDR receivers are available now. In this paper, we use RTL-SDR USB dongles based on the RTL2832U chipset [4], which were originally designed to be DVB-T TV tuners. The RTL-SDR supports continuous tuning in the range of [24, 1766] MHz, which covers the LoRaWAN bands. It can operate at $2.4\,\text{Msps}$ reliably for extended time periods. Thus, the sampling resolution is $1/2.4\,\text{Msps} = 0.42\,\mu s$. Our research is conducted based on a LoRaTS hardware prototype that integrates a Raspberry Pi, an iC880a LoRaWAN concentrator, and an RTL-SDR USB dongle. Fig. 7 shows the prototype. An $868\,\text{MHz}$ antenna is used with the RTL-SDR to improve signal reception.

The SDR receiver is used to capture the radio signal over a time duration of the first two preamble chirps of an uplink frame. The first sampled chirp is used to extract an accurate timestamp (cf. §5.3), whereas the second sampled chirp is used to extract the FB of the transmitter (cf. §6). The accurate timestamp is a prerequisite of the FB estimation. As only two chirps' radio waveform is analyzed, the Raspberry Pi suffices for performing the computation. Instead of using RTL-SDR, a full-fledged SDR transceiver (e.g., USRP) can be used to design a customized gateway with physical layer access. However, this design loses the factory-optimized hardware-speed LoRa demodulation built in the iC880a concentrator. Moreover, full-fledged SDR transceivers are often 10x more expensive than LoRaTS. The low-cost, low-power, listen-only RTL-SDR suffices for developing the attack detector.
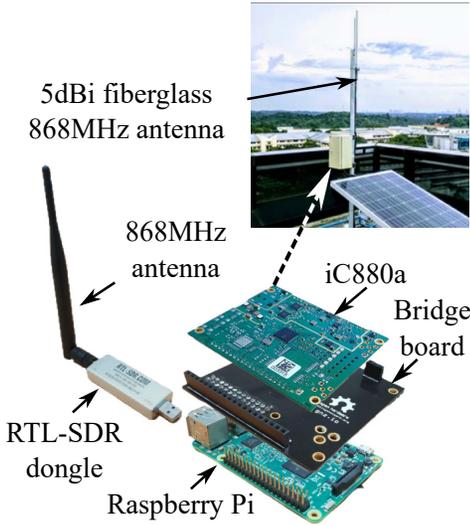
Fig. 7. LoRaTS hardware prototype consisting of Raspberry Pi, iC880a concentrator, bridge board, RTL-SDR USB dongle.
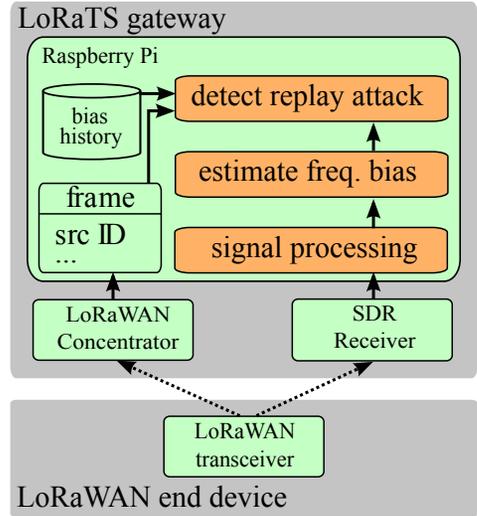

Fig. 8. LoRaTS software. Bottom part is end device; upper part is gateway; solid arrows are local data flows; dashed arrows are transmissions.

5.2 LoRaTS **Gateway Software**

The upper part of Fig. 8 illustrates the software architecture of LoRaTS to detect the attack. It is based on the results in the subsequent sections of this paper. The uplink transmission from the end device is captured by both the gateway's LoRaWAN concentrator and the SDR receiver. The LoRaWAN concentrator demodulates the received radio signal and passes the frame content to the Raspberry Pi. Signal processing algorithms are applied on the LoRa signal after down-conversion by the SDR receiver to determine precisely the arrival time of the uplink frame, estimate the transmitter's FB, and detect whether the current frame is a replayed one. The replay detection is by checking whether the estimated FB is consistent with the historical FBs associated with the transmitter ID contained in the current frame. Thus, the gateway is aware of the attack and can take necessary actions. Note that LoRaTS uses the SDR receiver to obtain FBs, rather than to decode the frame.

The LoRa communication is processed by the LoRa concentration board (i.e., iC880A), which is standalone hardware. The Raspberry Pi runs the concentration board's driver and forwards the packet data produced by the driver to our LoRaTS software stack. Thus, the LoRaTS software stack and the driver-forwarder pipeline concurrently run on the Raspberry Pi. As long as the computing resources of the Raspberry Pi are not exhausted, the LoRaTS software stack's computation and the communication will not have unhealthy contention. Note that the computing resources of the Raspberry Pi are sufficient to avoid the exhaustion.

5.3 **Signal Modeling and Uplink Frame Arrival Timestamping**

In this section, we present the modeling of the LoRa signal reception and our approach of detecting the onset time of the first preamble chirp. They form a basis for developing the FB estimation algorithms in §6.
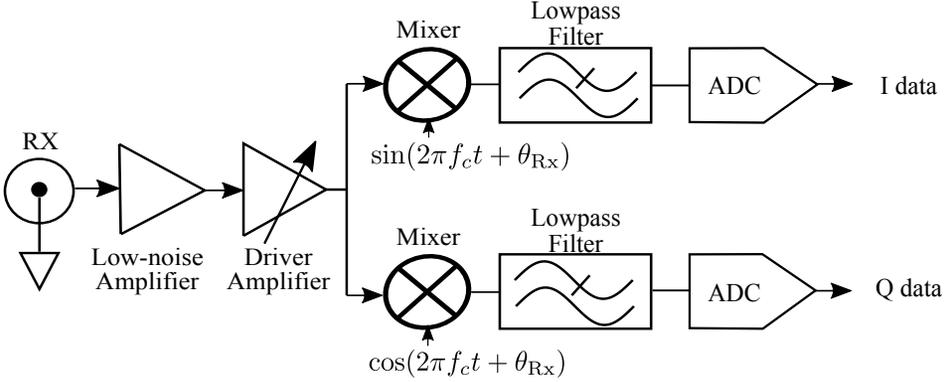
Fig. 9. Analog signal processing in SDR receiver.

5.3.1 Derivation of I and Q Components of LoRa Signal. Fig. 9 illustrates the essential analog signal processing steps of most SDR receivers to yield the in-phase (I) and quadrature (Q) components of the received radio signal. The SDR receiver generates two unit-amplitude orthogonal carriers $\sin(2\pi f_c t + \theta_{\mathrm{Rx}})$ and $\cos(2\pi f_c t + \theta_{\mathrm{Rx}})$, where f_c is a specified frequency and θ_{Rx} is the phase of the two self-generated carriers. The f_c can be set to be the central frequency of the used LoRa channel. The I and Q components, denoted by $s_I(t)$ and $s_Q(t)$, are

$$s_I(t) = s(t) \cdot \sin(2\pi f_c t + \theta_{\mathrm{Rx}})$$

$$= \frac{A(t)}{2} \left(\cos\left(2\pi \int_0^t f(x)\mathrm{d}x - 2\pi f_c t + \theta_{\mathrm{Tx}} - \theta_{\mathrm{Rx}} \right) \right. \tag{3}$$

$$\left. - \cos\left(2\pi \int_0^t f(x)\mathrm{d}x + 2\pi f_c t + \theta_{\mathrm{Tx}} + \theta_{\mathrm{Rx}} \right) \right), \tag{4}$$

$$s_Q(t) = s(t) \cdot \cos(2\pi f_c t + \theta_{\mathrm{Rx}})$$

$$= \frac{A(t)}{2} \left(\sin\left(2\pi \int_0^t f(x)\mathrm{d}x - 2\pi f_c t + \theta_{\mathrm{Tx}} - \theta_{\mathrm{Rx}} \right) \right. \tag{5}$$

$$\left. + \sin\left(2\pi \int_0^t f(x)\mathrm{d}x + 2\pi f_c t + \theta_{\mathrm{Tx}} + \theta_{\mathrm{Rx}} \right) \right), \tag{6}$$

The high-frequency components in Eqs. (4) and (6) are removed by the low-pass filters of the SDR receiver. Thus, the I and Q components after the filtering, denoted by $I(t)$ and $Q(t)$, are given by Eqs. (3) and (5). They can be rewritten as

$$I(t) = \frac{A(t)}{2} \cos \Theta(t), \quad Q(t) = \frac{A(t)}{2} \sin \Theta(t),$$

$$\Theta(t) = 2\pi \int_0^t f(x)\mathrm{d}x - 2\pi f_c t + \theta, \quad \theta = \theta_{\mathrm{Tx}} - \theta_{\mathrm{Rx}}.$$

The continuous-time $I(t)$ and $Q(t)$ are then sampled by the analog-to-digital converters (ADCs) to yield the I and Q data. For simplicity of exposition, the analysis in this paper is performed in the continuous-time domain.

5.3.2 CSS reception using SDR receiver. A chirp is a finite-time signal with time-varying frequency that sweeps the channel's bandwidth. Specifically, it can be expressed as $s(t) = A(t)\sin\left(2\pi \int_0^t f(x)\mathrm{d}x + \right.$
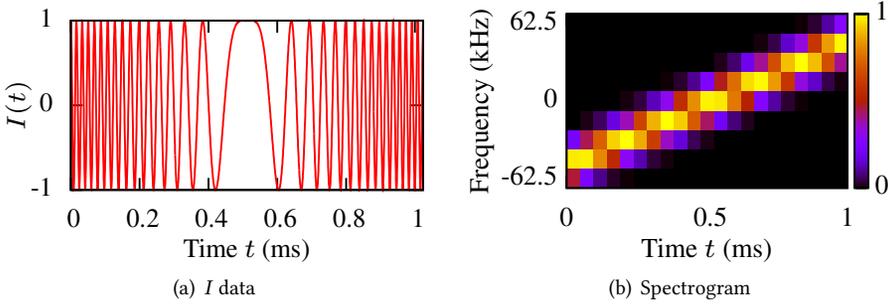
(a) I data

(b) Spectrogram

Fig. 10. I data (θ_{Tx}=θ_{Rx}) and spectrogram of a preamble chirp.

(a) I data with different phases.

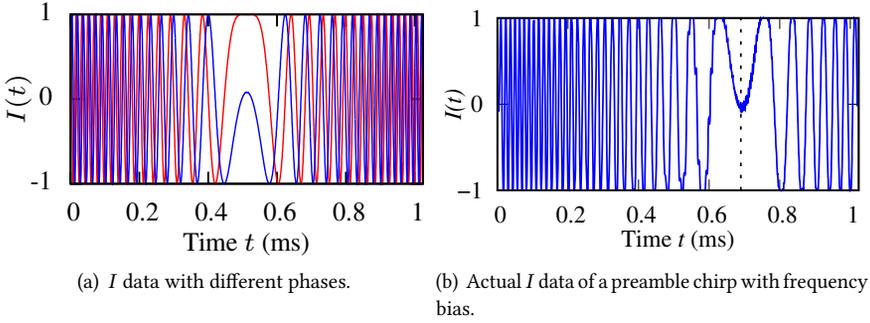(b) Actual I data of a preamble chirp with frequency bias.

Fig. 11. LoRa's I signal waveform is affected by the initial phase and frequency bias (FB).

θ_{Tx}), where $A(t)$ and $f(t)$ denote the instantaneous amplitude and frequency of the chirp at the time instant t, $\theta_{\mathrm{Tx}} \in [0, 2\pi)$ is the transmitter's phase that is usually unknown.

The SDR receiver generates two unit-amplitude orthogonal carriers $\sin(2\pi f_c t + \theta_{\mathrm{Rx}})$ and $\cos(2\pi f_c t + \theta_{\mathrm{Rx}})$, where f_c is the central frequency of the channel and θ_{Rx} is the phase of the two self-generated carriers. After mixing the received signal with the two orthogonal carriers and applying low-pass filtering, which are standard operations of SDR, the SDR receiver yields the I and Q components that can be expressed as $I(t) = \frac{A(t)}{2} \cos \Theta(t)$ and $Q(t) = \frac{A(t)}{2} \sin \Theta(t)$, where the angle $\Theta(t) = 2\pi \int_0^t f(x)\mathrm{d}x - 2\pi f_c t + \theta_{\mathrm{Tx}} - \theta_{\mathrm{Rx}}$.

A LoRaWAN uplink preamble consists of eight up chirps by default [31]. For a preamble chirp, $f(t) = \frac{W^2}{2^S} \cdot t - \frac{W}{2} + f_c$ for $t \in \left[0, \frac{2^S}{W}\right]$, where W is the channel bandwidth, $S \in \{6, 7, \ldots, 12\}$ is the *spreading factor*, and $\frac{2^S}{W}$ is the *chirp time*. The $f(t)$ increases linearly from $(f_c - W/2)$ Hz to $(f_c + W/2)$ Hz over a chirp time. The angle of the preamble chirp can be derived as $\Theta(t) = \frac{\pi W^2}{2^S} t^2 - \pi W t + \theta_{\mathrm{Tx}} - \theta_{\mathrm{Rx}}$. In this paper, we use a channel with $f_c = 869.75\,\mathrm{MHz}$ and $W = 125\,\mathrm{kHz}$. Fig. 10 shows the I data and the spectrogram of an ideal preamble chirp. The parameters for generating Fig. 10 are $A(t) = 2$, $\theta_{\mathrm{Tx}} - \theta_{\mathrm{Rx}} = 0$, and $S = 7$. Thus, the chirp time $\frac{2^S}{W}$ is $1.024\,\mathrm{ms}$. To generate the spectrogram, we apply the short-time fast Fourier transform (FFT) with 2^S-point Kaiser window and 16-point overlap between two neighbor windows. Thus, the spectrogram consists of 20 power spectral densities (PSDs) over the chirp time of $1.024\,\mathrm{ms}$.
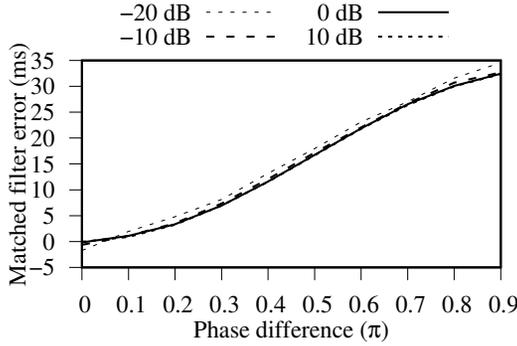
Fig. 12. The matched filter error with different phase differences. Different lines represent different SNR.
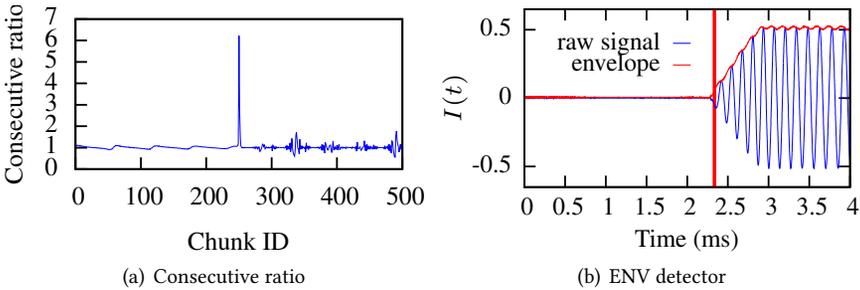

(a) Consecutive ratio

(b) ENV detector

Fig. 13. The intermediate and final results of envelope detector.

5.3.3 Preamble onset time detection. Detecting the onset time of the preamble is non-trivial. In this section, we discuss the matched filter approach and its inefficacy. Then, we present three other candidate methods.

Matched filter is a widely adopted symbol detection technique. Its principle is to slide a template signal over the incoming signal to detect the existence of the template signal's pattern in the incoming signal. Therefore, a basic assumption of the matched filter is that the I/Q signal of the received symbol has the same or similar pattern as the template signal. In the current context, this assumption is valid if the receiver is phase-locked to the transmitter (i.e., $\theta_{Rx} = \theta_{Tx}$). However, as LoRa adopts time-varying frequency, it is difficult for the SDR receiver to estimate the transmitter's phase θ_{Tx}. As a result, the phase difference $\theta_{Tx} - \theta_{Rx}$, which is a critical factor affecting the pattern of $I(t)$ and $Q(t)$, will be random. Fig. 11(a) shows the ideal $I(t)$ traces of the preamble chirp when the phase difference is 0 and π, respectively. The waveform shapes are different. Thus, we cannot define a template signal to achieve efficient matched filtering. In addition, we conduct a numeric experiment to assess how the phase difference affects the matched filtering performance. Fig. 12 presents the errors of the matched filter in determining the onset time of a preamble chirp ($S = 12$) using templates with different phase differences. Different curves correspond to different SNR. We can see that, compared with the noise level, the phase difference has a much more significant impact on the performance of the matched filter. The phase difference may result in up to tens of milliseconds error in onset time estimation. In contrast, the detectors presented in the rest of this section can achieve microseconds accuracy.

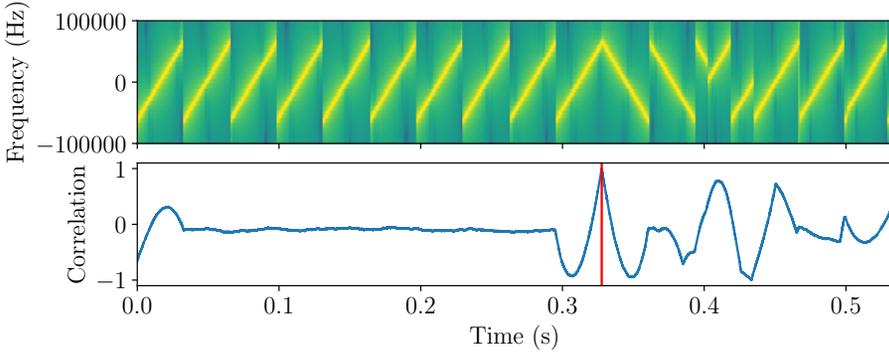For LoRaTS, we consider three parameter-less detectors:

Fig. 14. Correlation detector. The upper part shows the spectrogram of a LoRa frame. The bottom part shows the normalized correlation coefficient between the spectrogram and a locally generated hill peak pattern.

Envelope (ENV) detector: First, we apply the Hilbert transform to extract the amplitude envelope of the *I* or *Q* signal. Fig. 13(b) shows the extracted amplitude envelope for *I* data. We adopt the *folding* technique [25, 35] to detect the signal onset time from the amplitude envelope. Specifically, we evenly divide the envelope into chunks of equal length (e.g., 200 samples). Then, we calculate the sum of the absolute values of the amplitudes of all samples in each chunk, which is referred to as *trunk sum*. Lastly, we compute the ratio between the trunk sums of any two consecutive trucks to generate a ratio sequence. As shown in Fig. 13(a), the ratio sequence has a single peak. The detector yields the peak's time instant as the preamble onset time. The red vertical line in Fig. 13(b) indicates the detected onset time.

Correlation (CORR) detector: The Start Frame Delimiter (SFD) in a LoRa frame consists of two and a quarter down chirps. SFD is used by LoRa receiver for synchronization, because the junction of the up chirp before SFD and the first down chirp of SFD presents a salient hill peak as shown in the upper part of Fig. 14. We can compute the correlation between the spectrograms of the received LoRa signal and a locally generated hill peak template. The maximum of the correlation trace gives the time instant of the hill peak, which can be used to infer the onset time of the LoRa frame. The bottom part of Fig. 14 shows the normalized correlation coefficient trace and the detected hill peak time represented by the red vertical line.

AIC detector: The autoregressive Akaike Information Criterion (AIC) algorithm [30] was originally developed to estimate the arrival time of seismic waves with an accuracy of a single sampling point. As the *I* and *Q* signals are similar to the seismic waves [24], the AIC is a promising solution for our problem. It works as follows. For each point of the signal as an onset time candidate, two autoregressive models are constructed for the signal segments before and after the onset time candidate. The candidate that gives the largest dissimilarity between the two autoregressive models is yielded as the final result. From Fig. 15, AIC can detect the onset time from the signal with a smooth start. From the results in [30], AIC's detection results have a bias of 4 samples. With a sampling rate of 2.4 MHz, the bias is $\mathbb{E}[\epsilon] = \frac{4}{2.4\,\text{Msps}} = 1.67\,\mu s$ only, where ϵ represents onset time detection error.

5.3.4 Evaluation. We conduct experiments to evaluate the performance of the three detectors presented above. As AIC is nearly unbiased [30], we primarily assess the root-mean-square deviation (RMSD), which characterizes the consistency of the detection results. Due to the difficulty in obtaining the ground truth of the preamble arrival time, we indirectly estimate the RMSD as follows.
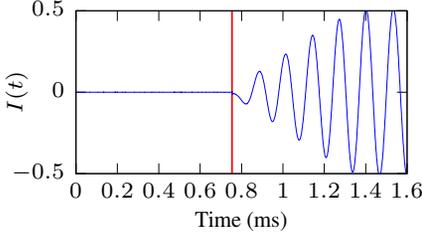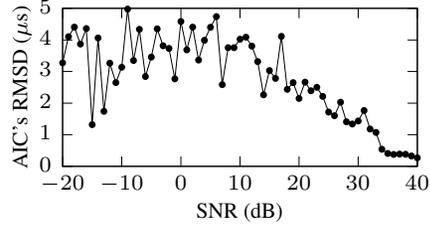
Fig. 15. AIC detector

Fig. 16. AIC's RMSD vs. SNR.

We place two LoRaTS nodes A and B close to each other such that the signal propagation time is near-zero. Node A initiates a round-trip communication; each of them detects the onset times for both its transmitted and received signals, generating four onset times totally. Note that a LoRaTS node's SDR receiver can also capture the signal transmitted by the node's LoRa radio and detect the onset time. Denote by Δ the measured round-trip time based on the detected onset times; denote by ϵ_X^{Tx} and ϵ_X^{Rx} the node X's unknown onset time detection errors for its transmitted and received signals. For the two close nodes A and B, $\Delta = \epsilon_A^{Rx} - \epsilon_A^{Tx} + \epsilon_B^{Rx} - \epsilon_B^{Tx}$ and $\text{RMSD}(\epsilon) = \frac{1}{2}\text{RMSD}(\Delta)$ if the errors are independent and identically distributed. Measurements show that $\text{RMSD}(\epsilon)$ is $1.21\,\mu s$, $0.64\,\mu s$, and $0.33\,\mu s$ for ENV, CORR, and AIC, respectively. Thus, AIC achieves more consistent detection results. Then, we evaluate the impact of random noises on AIC's $\text{RMSD}(\epsilon)$. We artificially add zero-mean Gaussian noises to the collected high-SNR I and Q traces. Then, we apply AIC on the noise-added traces to detect the preamble onset time. Fig. 16 shows the results. Note that the SNR range in Fig. 16 can cover realistic SNRs, e.g., $13\,dB$ to $-1\,dB$ in a multistory building (cf. §8). From Fig. 16, the AIC's $\text{RMSD}(\epsilon)$ is less than $5\,\mu s$ when the SNR is down to $-20\,dB$. Thus, AIC achieves robust onset time detection in the presence of strong noises. The rest of this paper uses AIC.

6 FRAME DELAY ATTACK DETECTION

Internal oscillators for generating carriers generally have FBs due to manufacturing imperfection. This section develops algorithms for estimating LoRa transmitters' FBs based on LoRa's CSS modulation and use them to detect the frame delay attack. Note that the existing FB estimation algorithms developed for other radios cannot be ported to LoRa due to different modulation schemes. For instance, the FB estimation for OFDM [34] is apparently not applicable for LoRa CSS. As discussed later, LoRa demodulation's built-in FB estimation technique does not provide sufficient resolution. Thus, highly accurate FB estimation for LoRa CSS is a non-trivial problem.

6.1 FB Estimation

This section describes algorithms for estimating the transmitter's FB based on an up chirp in the preamble. First, we analyze the impact of the transmitter's and SDR receiver's FBs (denoted by δ_{Tx} and δ_{Rx}) on the I and Q traces. The up chirp's instantaneous frequency accounting for δ_{Tx} is $f(t) = \frac{W^2}{2^S} \cdot t - \frac{W}{2} + f_c + \delta_{\text{Tx}}$, $t \in \left[0, \frac{2^S}{W}\right]$. The two local unit-amplitude orthogonal carriers generated by the SDR receiver are $\sin(2\pi(f_c + \delta_{\text{Rx}})t + \theta_{\text{Rx}})$ and $\cos(2\pi(f_c + \delta_{\text{Rx}})t + \theta_{\text{Rx}})$. After mixing and low-pass filtering, the I and Q components of the received up chirp can be derived as $I(t) = \frac{A(t)}{2}\cos\Theta(t)$ and $Q(t) = \frac{A(t)}{2}\sin\Theta(t)$, where the angle $\Theta(t)$ is given by

$$\Theta(t) = \frac{\pi W^2}{2^S}t^2 - \pi W t + 2\pi\delta t + \theta_{\text{Tx}} - \theta_{\text{Rx}}, \quad \delta = \delta_{\text{Tx}} - \delta_{\text{Rx}}. \tag{7}$$

When $\delta = 0$, the axis of symmetry of $I(t)$ is located at the midpoint of the preamble chirp time. As shown in Fig. 11(b), a negative δ causes a right shift of the axis of the symmetry in the time domain, whereas a positive δ causes a left shift.

For a certain SDR receiver, the FB estimation problem is to estimate δ from the captured I and Q traces. We do not need to estimate δ_{Tx}, because for a certain SDR receiver with a nearly fixed δ_{Rx}, a change in δ indicates a change in δ_{Tx} and a replay attack. In fact, FB estimation is a prerequisite of LoRa demodulation. Now, we discuss the incompetence of the LoRa demodulators' built-in FB estimation technique for attack detection. LoRa's CSS scheme evenly divides the whole channel bandwidth of W Hz into 2^S bins, where S is the spreading factor. The starting frequency of a bin corresponds to a symbol state. Since the preamble chirp linearly sweeps the channel bandwidth, its starting frequency can be viewed as the FB. LoRa demodulation firstly applies *dechirping* and then FFT to identify the preamble's and any data chirp's starting frequency bin indexes. The difference between the two indexes is the symbol state. As FFT achieves a resolution of $\frac{1}{x}$ Hz using x seconds of data, the Fourier transform of a chirp with length of $\frac{2^S}{W}$ seconds has a frequency resolution of $\frac{W}{2^S}$ Hz. This is also the resolution of the built-in FB estimation. Thus, for low spreading factor settings, the resolution may be poor. For instance, when $S = 7$ and $W = 125\,\mathrm{kHz}$, the resolution is $976.56\,\mathrm{Hz}$. As we will show in §6.2, this near-1 kHz resolution is insufficient to detect attacks that introduce sub-1 kHz FBs. The colliding frame disentanglement approach Choir [18] also uses the dechirping-FFT pipeline to analyze FB. Thus, it is subject to insufficient resolution. To achieve higher resolutions, this section presents two time-domain approaches, i.e., linear regression and least square, which are designed based on Eq. (7).

The FB estimation problem is essentially a parameter estimation problem based on noisy data since the wireless channel is inevitably subjected to various noises. Different from the methodology of developing the optimal parameter estimation algorithm based on detailed assumptions about the channel, this paper uses the two widely adopted parameter estimation methods (i.e., linear regression and least squares) based on key insights obtained from the signal model. In particular, the least squares method is a rule-of-thumb approach to reduce the impact of noises. We also conduct extensive evaluations to compare the two approaches' performance in the presence of various noise levels.

6.1.1 Linear regression approach. Eq. (7) can be rewritten as $\Theta(t) - \frac{\pi W^2}{2^S}t^2 + \pi Wt = 2\pi\delta t + \theta$, which is a linear function of t with $2\pi\delta$ as the slope. Thus, the slope can be estimated by linear regression based on the data pairs $(t, \Theta(t) - \frac{\pi W^2}{2^S}t^2 + \pi Wt)$, where $t \in \left[0, \frac{2^S}{W}\right]$, $\Theta(t) = \mathrm{atan2}(Q(t), I(t)) + 2k\pi$, and $k \in \mathbb{Z}$ rectifies the multi-valued inverse tangent function $\mathrm{atan2}(\cdot, \cdot) \in (-\pi, \pi)$ to an unlimited value domain. The rectification is as follows. The k is initialized to be 0 when $t = 0$. As t increases, if $\mathrm{atan2}(Q(t), I(t))$ jumps from $-\pi$ to π, k decreases by one; if $\mathrm{atan2}(Q(t), I(t))$ jumps from π to $-\pi$, k increases by one. Note that the traces $I(t)$ and $Q(t)$ where $t \in \left[0, \frac{2^S}{W}\right]$ are the segments of the captured I and Q signals starting from the preamble onset time detected by the AIC algorithm and lasting for a chirp time duration of $\frac{2^S}{W}$ seconds.

Note that the $I(t)$ and $Q(t)$ are the I and Q data traces captured by the SDR receiver for a complete preamble chirp. The preamble onset time detected by AIC is used to segment the I and Q traces to chirps. From our measurements, the first preamble chirp, in general, has an increasing amplitude $A(t)$ after the onset time (as shown in Fig. 15), which generates a negative impact on the linear regression accuracy. As the second preamble chirp has a more stable $A(t)$, we use the second chirp for the linear regression. As the linear regression approach has a closed-form formula to compute δ, it has a complexity of $O(1)$.
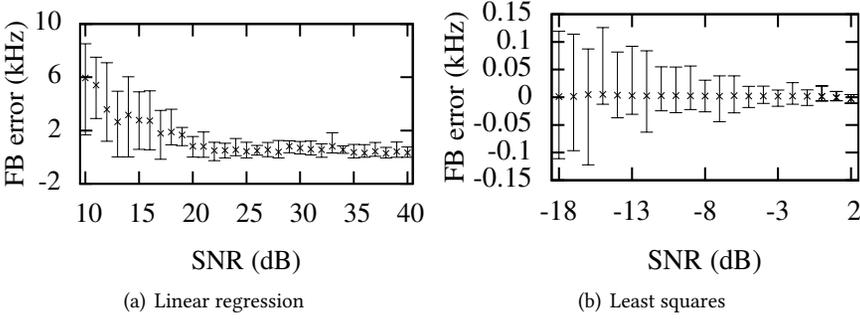
(a) Linear regression

(b) Least squares

Fig. 17. FB estimation errors vs. SNR.

6.1.2 Least squares approach. The LoRa signals can be very weak after long-distance propagation or barrier penetration. The LoRa's demodulation is designed to address low SNRs. For SX1276, the minimum SNRs required for reliable demodulation with spreading factors of 7 to 12 are -7.5 dB to -20 dB [1]. We aim at extracting FB at such low SNRs. We solve a least squares problem:

$$\underset{\theta_{\mathrm{Tx}}-\theta_{\mathrm{Rx}}\in[0,2\pi),\delta}{\arg\min}\sum_{t\in[0,2^S/W]}(Q(t)-A\sin\Theta(t))^2+(I(t)-A\cos\Theta(t))^2,$$

where $Q(t)$ and $I(t)$ are the received Q and I traces; $\Theta(t)$ is given by Eq. (7); $A\sin\Theta(t)$ and $A\cos\Theta(t)$ are the noiseless Q and I templates. The above formulation requires that the Q and I templates have an identical and constant amplitude A. As the second preamble chirp can meet this requirement, we use it for FB estimation. The A can be estimated as the square root of the difference between the average powers of the LoRa signal and the pure noise. We use a `scipy` implementation of the differential evolution algorithm [32] to solve the least squares problem. Raspberry Pi uses about 0.7 seconds to solve it. We use *Memory Profile* [7], a Python module for monitoring memory usage of Python programs, to profile our algorithm on a Raspberry Pi 3 Mod B. The memory usage is 77.191 MiB. The Raspberry Pi 3 Mod B has 1 GB memory, which is sufficient for the computation.

6.1.3 Performance comparison. We compare the FB estimation accuracy of the linear regression approach and the least squares approach. Fig. 17 shows the results. For each SNR setting, 20 LoRa I and Q traces with random FBs are generated using the signal model in Eq. (7). We also generate 20 noise traces; the magnitude of the noise is controlled to achieve the specified SNR. In Fig. 17, each error bar showing the 20%- and 80%-percentiles is from the 20 FB estimation results performed on the sum signals of the generated ideal LoRa signals and noise. From Fig. 17(a), the linear regression approach can achieve low FB estimation errors when the SNR is very high (e.g., 40 dB). However, it performs poorly for low SNRs. This is caused by the susceptibility of the inverse tangent rectification to noises. Specifically, as the inverse tangent rectification is based on a heuristic to detect `atan2`'s sudden transitions between $-\pi$ and π, large noises lead to false positive detection of the transitions. Differently, the least squares approach maintains the FB estimation error within 120 Hz (i.e., 0.14 ppm), when the SNR is down to -18 dB. Thus, the rest of this paper adopts the noise-resilient least squares approach, though it is more compute-intensive.

6.1.4 FB measurements for 16 end devices. We use an RTL-SDR to estimate the FBs of 16 SX1276-based end devices. In each test for an end device, the distance between the end device and the RTL-SDR is about 5 m. The error bars labeled "original" in Fig. 18 show the results. We can see that the FBs for a certain node are stable, and the nodes generally have different FBs. The absolute FBs
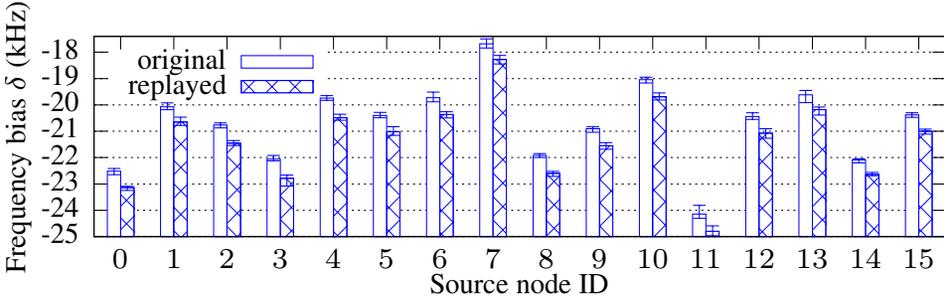
Fig. 18. FBs estimated from the original LoRa signals from 16 end devices and those replayed by a USRP-based replayer. The error bar shows mean, minimum, and maximum of FBs in 20 frame transmissions.
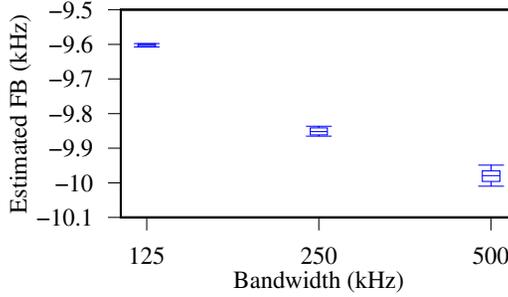

Fig. 19. FBs estimated from an end device with different bandwidth settings. Each box plot shows minimum, maximum, median, 25%, and 75% percentiles.

are from $17\,\mathrm{kHz}$ to $25\,\mathrm{kHz}$, which are about $20\,\mathrm{ppm}$ to $29\,\mathrm{ppm}$ of the nominal central frequency of $869.75\,\mathrm{MHz}$. Some nodes have similar FBs, e.g., Node 3, 8, and 14. Note that the detection of the replay attack is based on the fact that the replayed transmission has a different FB. In other words, the attack detection does not require distinct FBs among different end devices. From Fig. 18, we also observe that all nodes have negative FB measurements, which means that $\delta_{\mathrm{Tx}} < \delta_{\mathrm{Rx}}$, where δ_{Tx} and δ_{Rx} are the unknown FBs of the end device and the RTL-SDR. Note that as the RTL-SDR is a low-cost device, it may have a large FB causing the negative relative FB measurements.

6.1.5 Impact of coding rate and bandwidth settings on FB estimation. In this paper, we use a single preamble chirp to estimate the FB. The coding rate, which is related to the forward error correction (FEC) for payload, is irrelevant to the preamble chirps. Therefore, the coding rate is irrelevant to our FB estimation method. Regarding signal bandwidth, as our FB estimation approach is developed based on the signal model with the bandwidth as a parameter, the approach is valid under any bandwidth setting. We conduct an experiment to check the impact of different bandwidth settings on the FB estimation. We set $S = 12$, $f_c = 869.75\,\mathrm{MHz}$, coding rate $= 4/5$ while changing the bandwidth W to all possible settings, i.e., $125\,\mathrm{kHz}$, $250\,\mathrm{kHz}$, and $500\,\mathrm{kHz}$. Fig. 19 presents the FB estimates for an end device with different bandwidth settings. We can see that the estimation result varies slightly with the bandwidth setting. Thus, LoRaTS should re-profile the FB when an end device changes its bandwidth at run time.

Fig. 20. USRP N210 with GPSDO. The GPSDO module is marked with a red rectangle.

6.2 Replay Attack Detection

The replayer also has an FB. The error bars labeled "replayed" in Fig. 18 show the FBs estimated from the LoRa signals received by the LoRaTS's SDR receiver when a USRP replays the radio waveform captured by itself in the experiments presented in §6.1. Compared with the results labeled "original", the FBs of the replayed transmissions are consistently lower. This is because the USRP has a negative FB. The average additional FBs introduced by the replayer range from -543 to -743 Hz, i.e., 0.62 to 0.85 ppm of the channel's central frequency. Thus, with the FB estimation accuracy of 0.14 ppm achieved under low SNRs (cf. §6.1.2), the additional FBs caused by the replay attack can be detected.

Based on the above observation, we describe an approach to detect the delayed replay. LoRaTS maintains a database of the FBs of the nodes with which it communicates. This database can be built offline or at run time using its SDR receiver in the absence of attacks. To address the end devices' time-varying radio frequency skews due to run-time conditions like temperature, LoRaTS can continuously update the database entries based on the FBs estimated from recent frames. To decide whether the current received frame is a replayed frame, the LoRaTS gateway checks whether the FB of the current received frame is within the acceptable FB range of the end device based on the database. This detection approach is applied after the LoRaTS gateway decodes the frame to obtain the end device ID. The FB estimated from a frame detected as a replayed one should not be used to update the database.

This detection mechanism forms the first line of defense against the frame delay attacks that introduce extra FBs. It gives awareness of the attack that is based on the logistics of collision and record-and-replay. Note that with the knowledge of our detector, the attackers may invest more resources and efforts to hide their radiometrics. Thus, our FB-based attack detector forces the attackers to hide their radiometrics with increased cost and technical barriers.§7 will discuss the approach to eliminate the extra FBs and a further countermeasure to deal with the more crafted attack.

7 ZERO-FB ATTACK AND COUNTERMEASURE

7.1 Implementation of Zero-FB Attack

To bypass the FB-based attack detector described in §6.2, the adversary needs to precisely calibrate its eavesdropper and replayer to have FBs lower than the resolution of our FB estimation algorithm, e.g., 0.14 ppm as shown in §6.1.2. We call this attack *zero-FB attack*. The zero-FB attack's calibration requires a highly accurate frequency source operating at the channel frequency, which is non-trivial. The GPSDO module of USRP provides a GPS-locked reference clock of 10 MHz with 0.025 ppm accuracy [6]. In this section, we investigate whether we can implement the zero-FB attack using GPSDO-equipped USRP. Specifically, as shown in Fig. 20, we add a GPSDO module to the USRP-based eavesdropper and replayer, respectively. We use this improved attack apparatus to launch the frame delay attack. At the same time, we use LoRaTS to estimate the FBs of the original frame and the replayed frame for attack detection. Based on the experiment results, the additional FBs introduced by the attack are less than 0.035 ppm. The result is consistent with the accuracy level of the GPSDO's clock. Moreover, as this additional FB is below the resolution of our FB estimation algorithm (i.e., 0.14 ppm), our proposed FB-based attack detector is incapable of effectively detecting the attack. However, we stress that the attack setup is costly – each GPSDO unit costs about US$1,800 and the attack apparatus enhancement costs about US$3,600 in total, since both the collider and the eavesdropper need the GPSDO module. Therefore, we use a US$25 RTL-SDR to enforce the attacker to invest US$3,600 more in order to win the attack-defense chase.

7.2 Countermeasure

To detect the zero-FB attack, we propose a Pseudorandom Interval Hopping (PIH) scheme. The key idea of PIH is that, if the end device and the gateway achieve an agreement on the inter-frame intervals that are pseudorandom, the gateway can check the actual inter-frame intervals against the prior agreed intervals to detect the delays introduced by the attacker. To achieve the prior agreement, the gateway and the end device can establish a secret symmetric key by a key establishment protocol such as Diffie-Hellman's and use a pseudorandom generator seeded with the symmetric key to generate the inter-frame intervals. As the seed is confidential, the attacker can hardly predict the future pseudorandom inter-frame intervals based on the observed intervals. The end device will follow the pseudorandom sequence to regulate the transmission interval in transmitting the data frames. Now, we discuss several issues related to PIH.

Dealing with the waiting time at the end device. PIH is an add-on to sync-free timestamping for enhancing security. It does not require clock synchronization between the end device and the gateway because the gateway only checks the transmission interval between two consecutive frames. When the end device generates a new data record, it needs to wait for the next scheduled transmission time. Therefore, the end device needs to inform the gateway of the waiting time for correct data timestamping. The details of dealing with this issue are as follows. A LoRaWAN end device records the times of interest (e.g., the time instants when new sensor data records are acquired) in terms of its unsynchronized clock. Right before sending a number of data records using a frame, the device replaces the records' times of interest in its local clock with their elapsed times up to the present, form the frame, and transmit it immediately. We assume that the waiting time from the generation to the transmission of the data records is short to ensure limited local clock drift and limited bits to represent the elapsed times. For instance, to enforce an upper bound of 10 ms clock drift under a drift rate of 40 ppm, the waiting time needs to be within 4.1 minutes. Thus, the sync-free synchronization can compensate for the arbitrary transmission interval introduced by the PIH. Note that the waiting time also needs to account for attack detection capability that is discussed shortly.
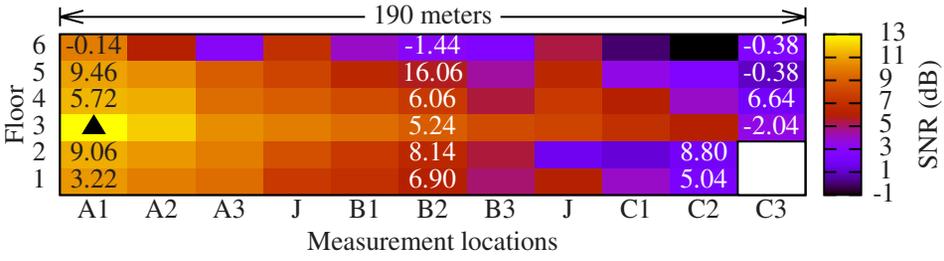
Fig. 21. SNR survey in a building (lateral view) with 3 sections (A, B, C) and 2 junctions (J). The triangle represents the fixed node. The number in a cell is round-trip time measurement in μs excluding propagation time but containing onset time detection error when mobile node is in the cell.

Setting of maximum inter-frame interval. Due to the end device's clock drift, the inter-frame interval may not exactly follow the prior agreed interval. To deal with this issue, the gateway can set a threshold to tolerate some deviations. If more deviation is allowed, the zero-FB attack can introduce some delay below the allowed deviation and remains undetected. If less deviation is allowed, the gateway may generate excessive false alarms, and the end device may need to use frequent synchronization sessions to calibrate its clocks, which introduces communication overhead. Thus, the threshold setting should be studied to achieve a good balance between security and efficiency under the PIH scheme. We provide an analysis as follows to guide the setting of the maximum inter-frame interval. The timing error of the end device, denoted by E, can be represented by $E = t \cdot r$, where t and r denote the maximum interval and clock drift rate, respectively. For example, if the r is 40 ppm (same setting as §3.2) and the maximum interval t is 30 minutes, the maximum timing error is 72 ms. Since we assume that the gateway has UTC, the allowed deviation can be determined based on the timing error of the end device. Reversely, we can determine the maximum interval based on a certain allowed deviation. For instance, if we set the allowed deviation to be 10 ms, the interval should be no greater than $\lceil \frac{E}{r} \rceil$ (i.e., $\lceil \frac{10\,\text{ms}}{40\,\text{ppm}} \rceil = 25\,\text{s}$). The pseudorandom generator needs to follow a uniform distribution, where the upper bound is the maximum interval. Moreover, the clock drift rate of the end device with Temperature Compensated Crystal Oscillator (TCXO) is low (e.g., 1 ppb). The parameter settings will be more flexible on these end devices with TCXOs. We note that the total clock drift of the end device over multiple uplink transmissions is irrelevant to the PIH approach that only checks the interval between any two consecutive uplink transmissions.

Transmission skipping, timeliness improvement, and frame losses. Sometimes the end device does not have data pending transmission at a scheduled transmission time or has data that is to be transmitted as soon as possible. To improve the efficiency and timeliness of PIH, we can divide the time into shorter time slices and allow skipping transmissions at the scheduled time instants. In this way, the real-time performance can be improved. In the case of a lossy link, the gateway can check the frame counter in the uplink frame and check the sum of these inter-frame intervals.

8 EXPERIMENTS

8.1 Experiments in a Multistory Building

LoRaWAN can be used for indoor applications, such as utility metering. We conduct a set of experiments to investigate the feasibility of attack and effectiveness of our attack detector in a concrete building with six floors. The building has three sections and two section junctions along its long dimension of 190 meters. Fig. 21 illustrates a lateral view of the building. First, we survey
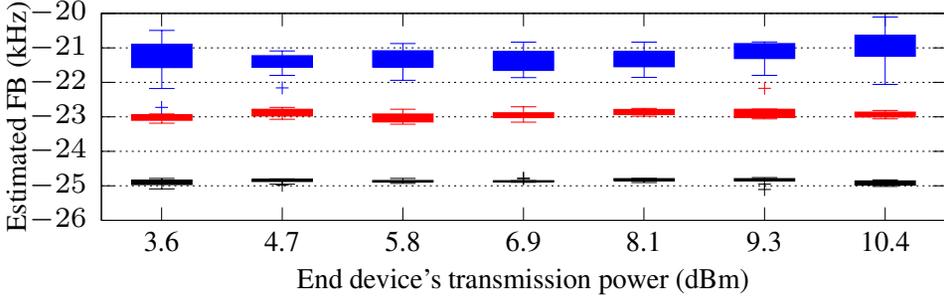
Fig. 22. Estimated FB vs. transmitting power of the end device. Each box plot shows min, max, 25% and 75% percentiles. (1) Bottom row: end device to eavesdropper; (2) Mid row: end device to LoRaTS gateway; (3) Top row: replayer to LoRaTS.

the SNR inside the building to understand the signal attenuation. We deploy a fixed LoRaWAN transmitter in Section A on the 3rd floor. Then, we carry an SDR receiver to different positions inside the building to measure the SNR. At each position, we first profile the noise power and then measure the total power when the fixed node transmits. In each section, we measure three positions. The heat map in Fig. 21 shows the SNR measurements. We can see that the SNR decays with the distance between the two nodes. The SNRs are from -1 dB to 13 dB. Then, we conduct the following experiments. By default, we set $S = 12$, $f_c = 869.75$ MHz, $W = 125$ kHz, coding rate $= 4/5$.

Attack experiments: We deploy an iC880a-based gateway and an SX1276-based end device in Section A1 of the 3rd floor and Section C3 of the 6th floor, respectively. The LoRa signals are significantly attenuated after passing through multiple building floors. If the end device adopts a spreading factor of 7, it cannot communicate with the gateway. A minimum spreading factor of 8 is needed for communications. We deploy two USRP N210 stations as the eavesdropper and the collider, next to the end device and the gateway, respectively. We set the transmitting power of the end device and the collider to be 14 dBm. The malicious collision is stealthy to the gateway; the eavesdropping is successful. Thus, the frame delay attack can be launched in this building.

Onset time detection: We replace the iC880a gateway with our LoRaTS gateway and move the end device in the building. The number shown in a cell of Fig. 21 is the measured round-trip time Δ in μs excluding the propagation delay when the end device is at the corresponding location. Note that the propagation delay is calculated based on the estimated Euclidean distance between the gateway and the end device. As Δ contains an onset time detection error, it may become negative. Then, we compute the RMSD of all Δ measurements shown in Fig. 21. From the analysis in §5.3.4, the AIC's average $\mathrm{RMSD}(\epsilon)$ in this building is $2.4\,\mu$s only. This result is consistent with that in Fig. 16.

Impact of transmitting power on FB estimation: Fig. 22 shows the estimated FBs versus the end device's transmitting power under different settings. The bottom row of black box plots is the FBs estimated by the eavesdropper when the end device transmits the uplink frame with different transmitting powers. The middle row of red box plots is the FBs estimated by the LoRaTS gateway in the absence of the frame collision and replay attacks. Thus, the FBs estimated by the eavesdropper and the LoRaTS gateway are different. This is because that as analyzed in §6.1, the estimated FB δ contains the transmitter's and receiver's FBs δ_{Tx} and δ_{Rx}. Note that the eavesdropper and the LoRaTS gateway, in general, have different FBs. From Fig. 22, the end device's transmitting power has little impact on the FB estimation.

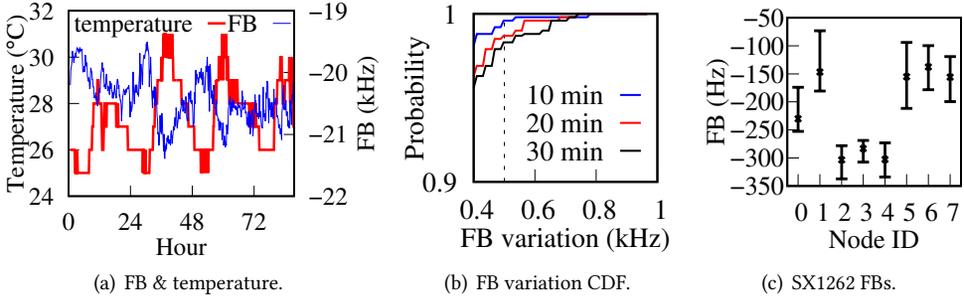(a) FB & temperature.   (b) FB variation CDF.   (c) SX1262 FBs.

Fig. 23. (a), (b): Temporal stability of SX1276's FB over 87 hours. (c): The FBs of eight SX1262 LoRa chips with TCXOs.

Additional FB introduced by replayer: In Fig. 22, the top row of blue box plots are the FBs estimated by the LoRaTS when the replayer replays the radio waveform recorded by the eavesdropper. When the end device adopts a higher transmitting power, the replayed signal also has higher power. By comparing the middle and the top rows, we can see that the replay attack introduces an additional FB of about 2 kHz, which is 2.3 ppm of the LoRa channel's central frequency. Therefore, FB monitoring can easily detect the replay attack. Compared with the results in Fig. 18 showing additional FBs of 0.62 to 0.85 ppm, the FBs in this set of experiments are higher. This is because we use two different USRPs as the eavesdropper and replayer; their FBs are superimposed.

8.2 Temporal Stability of FB

FB can be affected by ambient conditions such as temperature. We continuously track the FB of an SX1276-based end device for 87 hours to study its temporal stability. We place the end device with a temperature sensor in a semi-outdoor corridor with time-varying temperature. The end device transmits 10 frames every 10 minutes to the LoRaTS gateway, resulting 1,440 frames per day. Fig. 23(a) shows the end device's temperature and FB traces. The Pearson correlation between FB and temperature is -0.78. Moreover, the FB has transient variations that can be caused by interference from other communication systems operating in neighbor frequency bands. As LoRaTS detects the attack based on the changes of FB, such transient variations may cause false alarms. Fig. 23(b) shows the CDFs of the maximum FB variation if the end device transmits a frame every 10, 20, and 30 minutes. If the attack detection threshold is 500 Hz based on our previous measurements of the additional FB introduced by the attack, from the CDFs, the false alarm rate (i.e., the probability that the FB variation exceeds 500 Hz) is about 0.4%, 1.3%, and 1.7% for the three frame interval settings.

The SX1276 used in this paper does not have TCXO. For LoRa radios with TCXO, the false alarm rate can be further reduced. To verify this, we deploy eight SX1262-based end devices. SX1262 is the next-generation LoRa chip equipped with TCXO [8]. In Fig. 23(c), each error bar shows the 10%- and 90%-percentiles of 150 FB estimation results. We can see that the TCXO can significantly shrink the fluctuation. Specifically, from our measurements, the FB variations are no greater than 250 Hz. In contrast, from Fig. 23(a), without TCXO, the FB fluctuation range is up to 1 kHz. Thus, with a detection threshold of 500 Hz, the false alarm rate of our approach for a system based on SX1262 will be near-zero.

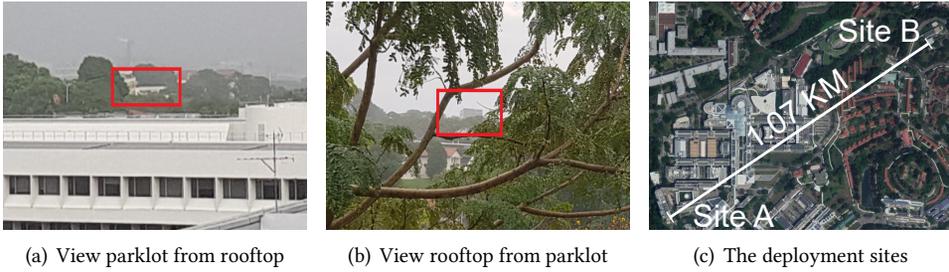(a) View parklot from rooftop   (b) View rooftop from parklot   (c) The deployment sites

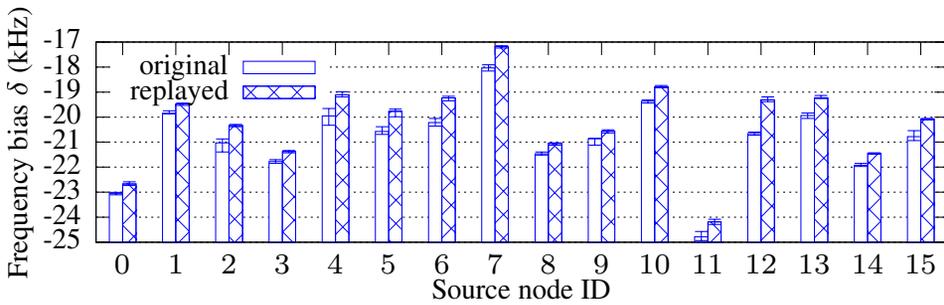Fig. 24. Pictures taken at the two sites (Site A: rooftop, Site B: parklot).

Fig. 25. LoRaTS's FB estimates when the distances between the gateway and the end devices are about 1.07 km.

8.3 Outdoor Experiments

Outdoor experiments with longer distance: In this set of experiments, we deploy SX1276-based end devices in an outdoor parking lot. The LoRa parameters for the end devices are the same as we used in the indoor experiments. We replace the iC880a-based gateway shown in Fig. 6 with a LoRaTS gateway. The distance between the end device and the LoRaTS gateway is about 1.07 km. Fig. 24 shows the pictures taken at the two sites. The circled construct in a figure is the building where the other site is located in. The collider shown in Fig. 6 is also used in this set of experiments. The eavesdropper is deployed at a location about 200 m from the end device. When the transmitting powers of the end device and the collider are 14 dBm and 8 dBm, respectively, we can successfully launch the frame delay attack. We also use the round-trip timing approach discussed in §5.3.4 to evaluate AIC's performance. The measurements show that AIC's $\mathrm{RMSD}(\epsilon)$ is 1.29 μs only. This result is better than that obtained in the multistory building because the LoRa signal suffers significant attenuation in the indoor environment. Then, we investigate the additional FBs introduced by the replay attack. Fig. 25 shows LoRaTS's FB estimates for the frames transmitted by 16 end devices and the corresponding replays. The extra FBs introduced by the attack is up to 1.76 ppm. Thus, the attack can be detected.

Impact of gateway heights on FB estimation: The device height may affect the communication performance due to the different path losses. To evaluate the impact of gateway height on FB estimation, we use two LoRaTS gateways to receive the LoRa signal simultaneously in this experiment. We set a *static gateway* at a fixed height ($5m$), relative to the end device, and carry the other gateway called *moving gateway* to different heights. As shown in Fig. 26, the blue box plots and the red box plots show the FB estimation results from the static and moving gateway,
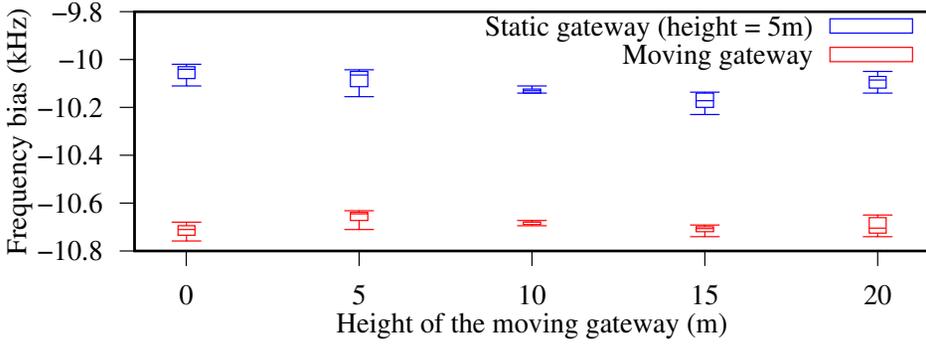
Fig. 26. FB estimation results from the *static gateway* and the *moving gateway*. The *static gateway* is placed at a fixed heights ($5m$) while the *moving gateway* is carried to different heights. These two LoRaTS gateways receive the LoRa signal simultaneously. Each box plot shows minimum, maximum, median, 25%, and 75% percentiles.
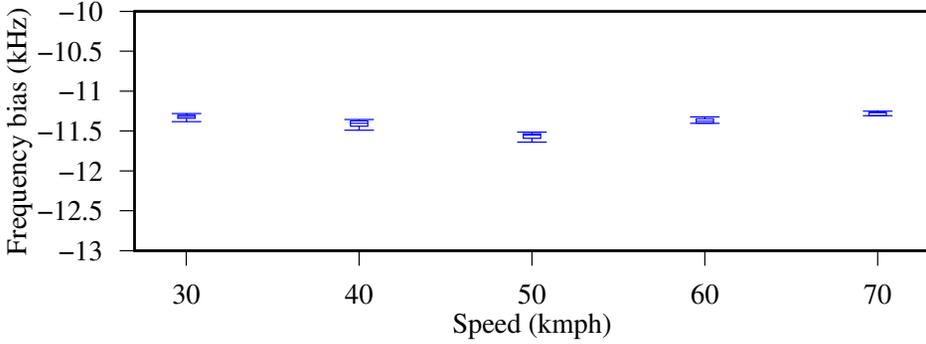

Fig. 27. FB estimation results of an end device moving at different speeds. Each box plot shows minimum, maximum, median, 25%, and 75% percentiles.

respectively. Each box plot shows minimum, maximum, median, 25%, and 75% percentiles. From the experiment results, we can see that the gateway's height does not have a noticeable impact on the performance. The FB estimations from the two gateways are different, because the estimated FB contains both the transmitter's and receiver's FBs, as analyzed in §6.1.

Impact of end device movement on FB estimation: When there is a relative movement between a transmitter and a receiver, the received signals have frequency shifts caused by the Doppler effect. Previous studies have shown that LoRa modulation is robust against the Doppler effect [17]. We now analyze the impact of the Doppler effect on the FB estimation. When the end device moves at a velocity v relative to the gateway, the FB caused by the Doppler effect, denoted by Δf_d, is $\frac{v}{c}f_{chirp}$, where the c is the speed of light and f_{chirp} is the instantaneous frequency of the chirp [23]. When the end device moves at a speed of 70 km/h, the additional FB caused by the Doppler effect is about 50 to 60 Hz. We conduct a set of experiments to understand the impact of movement speed on the FB estimation. Specifically, the experiments are conducted on a road section. We set up a LoRaTS gateway at the road side and carry an end device in a car. The car passes the gateway at different speeds of 30, 40, 50, 60, and 70 km/h. The FB esitmation results are shown in Fig. 27. Each box plot shows minimum, maximum, median, 25%, and 75% percentiles of the FB estimates.
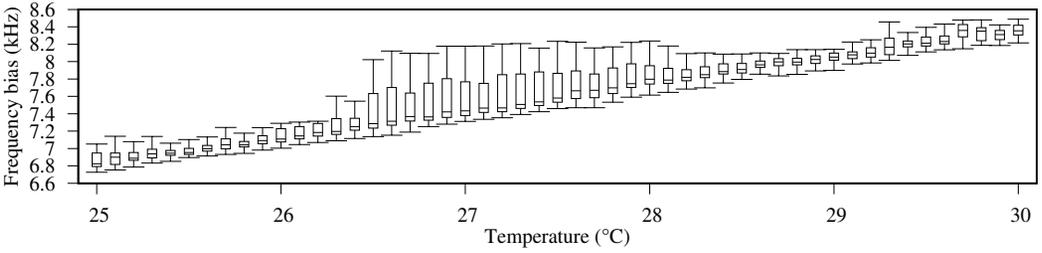
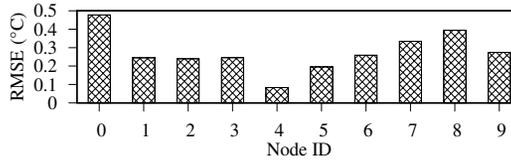Fig. 28. Boxplot of the temperature vs. frequency bias of an SX1276-based end device.

Fig. 29. The root-mean-square errors (RMSEs) of linear regression models for 10 SX1276-based end devices.

From the results, we can observe that the FB estimation results for a certain movement speed are stable, with a maximum fluctuation of 145 Hz. Moreover, the FB fluctuation is about 400 Hz across all speeds. This cross-speed fluctuation is caused by the variation of the ambient temperature of the end device during the experiment process. From our experiment results regarding the impact of temperature on FB that will be presented shortly in §8.4, the 400 Hz FB variation observed in Fig. 27 corresponds to a 0.5°C temperature variation around 26°C (cf. Fig. 28). Note that 26°C is the cabin temperature we set on the car's air conditioning system. As cars' air conditioning systems often have ±1°C control errors [33], we cannot completely eliminate the impact of temperature variations in this set of experiments. In Fig. 27, we do not see the monotonic relationship between FB and movement speed, because the impact of the movement speed (i.e., up to 60 Hz additional FB according to our analysis) is much smaller than the impact of the temperature variation in the car.

8.4 Leveraging Temperature for FB Monitoring

As discussed in §8.2, an end device without TCXO needs to transmit frames periodically to help the gateway track its FB in the presence of temperature variations. However, for the end devices that do not have pending application data for transmission, transmitting dummy frames for the purpose of FB tracking wastes energy. To mitigate this issue, we propose to leverage temperature to lower the transmission frequency. In this approach, the end device needs to be equipped with a temperature sensor and attach the temperature reading in the uplink frame. When the uplink frame is received by the LoRaTS gateway, the gateway will first estimate the FB of the end device. Then, the LoRaTS estimates the temperature based on a known temperature-FB model and compares the estimated temperature with the actual temperature reading in the payload. If the difference between the estimated value and the sensor reading exceeds a pre-defined threshold, the gateway declares the detection of the stealthy frame delay attack.

To evaluate this approach, we deploy ten SX1276-based end devices, each equipped with an AM2302 temperature-humidity sensor [9]. Each end device transmits a frame, including the reading from the temperature-humidity sensor periodically. Once upon receiving the frame, the gateway timestamps the data, records the temperature, and estimates the FB. Thus, for each end device, we can get a pairwise temperature-FB trace based on collected data. Fig. 28 shows the boxplot
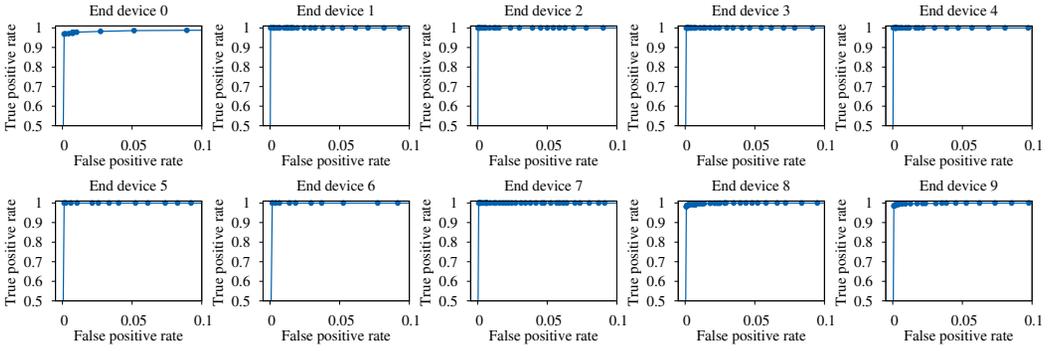
Fig. 30. ROCs for linear regression models for 10 end devices.

of the pairwise data collected by an end device over 24 hours. We can see that the frequency bias increases with the temperature and follows a linear trend. This is because the frequency-temperature characteristics of the crystal oscillator without the temperature compensation exhibit a linear relationship in this temperature range (i.e., 25°C to 30°C) [22]. Thus, we apply linear regression to model the relationship during the modeling phase and use the model to estimate the temperature according to the FB at run time. We use 2,000 pairwise data points as the training data set in the experiment. Fig. 29 shows the root-mean-square errors (RMSEs) of the linear regression models for all the ten end devices. We can see that the RMSEs are below 0.5°C. Fig. 30 shows the receiver operating characteristic (ROC) curves of using the models to detect the frame delay attacks on the frames transmitted by the ten end devices. Different data points on a ROC curve are results based on different temperature detection thresholds. Let N denote the total number of tests for an end device. Accordingly, let N_{TA} and N_{FA} denote the total numbers of true alarms and false alarms at the gateway side, respectively. The true positive rate and the false positive rate are measured by N_{TA}/N and N_{FA}/N, respectively. For each end device, our approach achieves empirical true positive rates of 100%, subject to a false positive rate upper bound of 1%.

Although this approach needs an additional temperature sensor for each end device, it further lowers the requirement of the communication frequency to save energy in the long term. Note that temperature sensors are often available in many monitoring applications. To capture the runtime affecting factors introduced by different environments and wireless conditions, the network can perform *in situ* temperature-FB profiling based on the collected FB and temperature data over a certain time duration. The resulting temperature-FB model thus captures the run-time affecting factors. To prevent the attacker from misleading the gateway to build a false profile, the network operator should perform the profiling in a short attack-free time period with close supervision, e.g., one day in each season. With the closely supervised profiling in a short period of time, we can make sure that the network can mitigate the impact of attack throughout longer periods of time.

9 CONCLUSION

This paper shows that sync-free data timestamping, though bandwidth-efficient, is susceptible to the frame delay attack that can be implemented by a combination of frame collision and delayed replay. Experiments show that the attack can affect many end devices in a large geographic area. To gain attack awareness, we design a gateway called LoRaTS that integrates a low-power SDR receiver with a commodity LoRaWAN gateway. We develop efficient time-domain signal processing algorithms to estimate the FBs of the end devices. The least squares FB estimation algorithm achieves high resolution and can uncover the additional FBs introduced by the attack. We also

consider a more skillful and resourceful attacker who eliminates the additional FBs. We propose a pseudorandom interval hopping scheme to counteract the zero-FB attacks. In summary, with LoRaTS, we can achieve efficient sync-free data timestamping with awareness of frame delay attack.

ACKNOWLEDGMENTS

REFERENCES

[1] 2018. http://modtronix.com/prod/components/wireless/sx1276.pdf.
[2] 2019. https://github.com/rpp0/gr-lora.
[3] 2019. Fog Computing for Industrial Automation. http://bit.ly/2uOYp9z.
[4] 2019. RTL-SDR. https://www.rtl-sdr.com/.
[5] 2019. Vector Signal Analyzer. https://www.keysight.com/en/pc-2416877/vector-signal-analyzer?cc=US&lc=eng.
[6] 2020. GPSDO. http://bit.ly/2FL8Fl5.
[7] 2020. *Memory Profiler*. https://github.com/pythonprofilers/memory_profiler
[8] 2020. *Wireless SX1262 LoRa Module*. http://www.ebyte.com/en/product-view-news.aspx?id=437
[9] 2020. *Wireless SX1262 LoRa Module*. https://www.adafruit.com/product/393
[10] Amazon. 2020. RTL-SDR Blog R820T2 RTL2832U 1PPM TCXO SMA Software Defined Radio (Dongle Only). https://www.amazon.com/dp/B0129EBDS2.
[11] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes. 2017. Exploring the Security Vulnerabilities of LoRa. In *CYBCONF*. 1–6.
[12] Emekcan Aras, Nicolas Small, Gowri Sankar Ramachandran, Stephane Delbruel, Wouter Joosen, and Danny Hughes. 2017. Selective Jamming of LoRaWAN using Commodity Hardware. In *MobiQuitous*.
[13] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. 2008. Wireless device identification with radiometric signatures. In *MobiCom*.
[14] Jinzhu Chen, Rui Tan, Yu Wang, Guoliang Xing, Xiaorui Wang, Xiaodong Wang, Bill Punch, and Dirk Colbry. 2012. A high-fidelity temperature distribution forecasting system for data centers. In *IEEE RTSS*.
[15] Silvia Demetri, Marco Zúñiga, Gian Pietro Picco, Fernando Kuipers, Lorenzo Bruzzone, and Thomas Telkamp. 2019. Automated estimation of link quality for LoRa: a remote sensing approach. In *IPSN*.
[16] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe. 2018. Charm: Exploiting Geographical Diversity Through Coherent Combing in Low-Power Wide-Area Networks. In *IPSN*.
[17] Alexander A Doroshkin, Alexander M Zadorozhny, Oleg N Kus, Vitaliy Yu Prokopyev, and Yuri M Prokopyev. 2019. Experimental study of LoRa modulation immunity to Doppler effect in CubeSat radio communications. *IEEE Access* 7 (2019), 75721–75731.
[18] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. 2017. Empowering Low-Power Wide Area Networks in Urban Settings. In *SIGCOMM*.
[19] T. Hao, R. Zhou, G. Xing, and M. Mutka. 2011. WizSync: Exploiting Wi-Fi infrastructure for clock synchronization in wireless sensor networks. In *RTSS*.
[20] Mehrdad Hessar, Ali Najafi, and Shyamnath Gollakota. 2019. NetScatter: Enabling Large-Scale Backscatter Networks. In *NSDI*.
[21] The European Telecommunications Standards Institute. 2017. Technical characteristics for Low Power Wide Area Networks Chirp Spread Spectrum (LPWAN-CSS) operating in the UHF spectrum below 1 GHz.
[22] MinQiang Li, XianHe Huang, Feng Tan, YanHong Fan, and Xun Liang. 2005. A novel microcomputer temperature-compensating method for an overtone crystal oscillator. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 52, 11 (2005), 1919–1922.
[23] Jansen C Liando, Amalinda Gamage, Agustinus W Tengourtius, and Mo Li. 2019. Known and unknown facts of LoRa: Experiences from a large-scale measurement study. *ACM Transactions on Sensor Networks (TOSN)* 15, 2 (2019), 1–35.
[24] Guojin Liu, Rui Tan, Ruogu Zhou, Guoliang Xing, Wen-Zhan Song, and Jonathan M Lees. 2013. Volcanic earthquake timing using wireless sensor networks. In *IPSN*.
[25] RVE Lovelace, JM Sutton, and EE Salpeter. 1969. Digital search methods for pulsars. *Nature* 222, 5190 (1969), 231–233.
[26] Seri Oh, Stephen G Ritchie, and Cheol Oh. 2002. Real-time traffic measurement from single loop inductive signatures. *Transportation Research Record* 1804, 1 (2002), 98–106.

[27] Yao Peng, Longfei Shangguan, Yue Hu, Yujie Qian, Xianshang Lin, Xiaojiang Chen, Dingyi Fang, and Kyle Jamieson. 2018. PLoRa: a passive long-range data network from ambient LoRa transmissions. In *SIGCOMM*.

[28] Ceferino Gabriel Ramirez, Anton Sergeyev, Assya Dyussenova, and Bob Iannucci. 2019. LongShoT: long-range synchronization of time. In *IPSN*.

[29] Pieter Robyns, Eduard Marin, Wim Lamotte, Peter Quax, Dave Singelée, and Bart Preneel. 2017. Physical-layer fingerprinting of LoRa devices using supervised and zero-shot learning. In *WiSec*.

[30] Reinoud Sleeman and Torild Van Eck. 1999. Robust automatic P-phase picking: an on-line implementation in the analysis of broadband seismogram recordings. *Physics of the earth and planetary interiors* 113, 1-4 (1999), 265–275.

[31] N. Sornin, M. Luis, T. Eirich, T. Kramp, and O. Hersent. 2016. LoRaWAN™Specification (V1.0.2).

[32] Rainer Storn and Kenneth Price. 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization* 11, 4 (1997), 341–359.

[33] Yi Xie, Zhaoming Liu, Jiangyan Liu, Kuining Li, Yangjun Zhang, Cunxue Wu, Pingzhong Wang, and Xiaobo Wang. 2020. A Self-learning intelligent passenger vehicle comfort cooling system control strategy. *Applied Thermal Engineering* 166 (2020), 114646.

[34] Yingwei Yao and Georgios B Giannakis. 2005. Blind carrier frequency offset estimation in SISO, MIMO, and multiuser OFDM systems. *IEEE Transactions on Communications* 53, 1 (2005), 173–183.

[35] Ruogu Zhou, Yongping Xiong, Guoliang Xing, Limin Sun, and Jian Ma. 2010. ZiFi: wireless LAN discovery via ZigBee interference signatures. In *MobiCom*.