Secure Data Timestamping in Synchronization-Free LoRaWAN

Chaojie Gu  
School of Computer Science and Engineering  
Nanyang Technological University  
Singapore  
gucj@ntu.edu.sg

Rui Tan  
School of Computer Science and Engineering  
Nanyang Technological University  
Singapore  
tanrui@ntu.edu.sg

Jun Huang  
Center for Energy-efficient Computing and Applications  
Peking University  
China  
jun.huang@pku.edu.cn

ABSTRACT

Low-power wide-area network technologies such as LoRaWAN are important for achieving ubiquitous connectivity required by the Internet of Things. Due to limited bandwidth, LoRaWAN is primarily for applications of collecting low-rate monitoring data from geographically distributed sensors. In these applications, sensor data timestamping is often a critical system function. This paper considers a synchronization-free approach of timestamping the uplink data at the LoRaWAN gateway, which can give milliseconds accuracy. Its key advantages are simplicity and no extra overhead, commensurate with the scarce communication resources of LoRaWAN. However, we show that this low-overhead approach is susceptible to a frame delay attack that can be implemented by a combination of stealthy jamming and delayed replay. To address this threat, we propose a SoftLoRa gateway design that integrates a commodity LoRaWAN gateway with a low-power software-defined radio receiver to track the inherent frequency biases of LoRaWAN end devices. With a set of efficient signal processing algorithms that are designed based on LoRaWAN’s modulation method, our frequency bias estimation achieves a resolution of 0.14 parts-per-million (ppm) of the channel’s central frequency. This resolution is sufficient to detect the attack that introduces an additional frequency bias of one or more ppm. We evaluate our approach in various indoor and outdoor environments. In summary, this paper presents an attack-aware and low-overhead approach to timestamping the data generated by LoRaWAN end devices.

1 INTRODUCTION

Low-power wide-area networks (LPWANs) enable direct wireless interconnections among end devices and gateways in geographic areas of up to tens of square kilometers [19]. It will increase the network connectivity as a defining characteristic of the Internet of Things (IoT). Among various LPWAN technologies (including NB-IoT and Sigfox), LoRaWAN [22], which is an open data link layer specification based on the LoRa physical layer technique, offers the advantages of using license-free ISM bands (e.g., EU 868 MHz and US 915 MHz), low costs for end devices, and independence from managed infrastructures (e.g., cellular networks).

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 IoT measurement devices, and etc. Most of these real-world monitoring applications require the sensing data to have timestamps in the global time, though the timestamps do not have to be highly accurate such as microseconds level. For instance, in both indoor and outdoor environment condition monitoring, seconds accuracy for sensor data timestamps will be sufficient due to the slow dynamics of the environment condition. Second-accurate timestamps for the traffic data generated by roadway detectors can be used to reconstruct real-time traffic maps well. In a range of industrial monitoring applications such as oil pipeline monitoring, milliseconds accuracy is sufficient.

Conventionally, to perform the data timestamping in a wireless sensor network (WSN) for data collection, the clocks of the sensor nodes need to be synchronized, such that the nodes can timestamp their data once generated. To achieve clock synchronization, each WSN node can be equipped with a GPS receiver for accessing the global time. However, GPS receivers consume excessive power and may not work in indoor environments. Thus, various WSN clock synchronization protocols based on message exchanges have been developed. Different from the above synchronization-based approach, the synchronization-free approach uses the gateway with a globally synchronized clock to timestamp the data upon the arrival of the corresponding network packet. However, in multi-hop WSNs, this synchronization-free approach may perform unsatisfactorily, because the data delivery on each hop may have uncertain delays due to various factors such as channel contention among nodes.

Differently, the synchronization-free approach is desirable for uplink data timestamping in LoRaWANs. Reasons are two-fold. First, in contrast to the 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 times from the end devices to the gateway are generally in microseconds, the LoRaWAN frame arrival time instant can well represent the time instant that the frame leaves the end device. As a result, timestamping the uplink data at the gateway can meet the timestamping accuracy requirements of many real-world applications. Second, if the synchronization-based approach is adopted otherwise, the task of keeping the end devices’ clocks synchronized and the inclusion of timestamps in the LoRaWAN data frames will introduce considerable overhead to the narrowband LoRaWANs. Given the above reasons, the synchronization-free approach of timestamping the uplink data at the gateway is lightweight, implementation-friendly, and efficient due to its simplicity.

Despite the prospect of the synchronization-free approach owing to its low overhead, in this paper, we take an adversary perspective to examine this approach under the LoRaWAN context and seek to improve its security. This is because that wrong timestamping can also lead to undesirable consequences. The long-range communication capability of LoRaWAN, though increasing connectivity, renders the communications susceptible to wireless attacks that...
can be launched from remote and hidden sites. The LoRaWAN specifications only define conventional frame confidentiality, integrity, and device authentication measures. These conventional security measures may be inadequate to protect the network from wireless attacks on the physical (PHY) layer of LoRaWAN.

In this paper, we consider a basic threat of frame delay attack described in the RFC 7384 [15] that will subvert the synchronization-free uplink data timestamping approach. Our experiments based on a commodity LoRaWAN platform show that there is a time window of tens of milliseconds (ms) after the onset of a legitimate frame transmission for implementing stealthy jammers. The jammed victim gateway cannot decode any frame and raises no alerts to the operating system (OS). The lengthy time window makes the stealthy jamming easily achievable using commodity LoRa devices. Based on the above, we have implemented the frame delay attack through a combination of stealthy jamming and frame replay that can introduce arbitrary delays to the deliveries of LoRaWAN frames. This will subvert the synchronization-free data timestamping approach that assumes near-zero signal propagation times. As the attack does not breach frame integrity, it cannot be solved by cryptographic protection and conventional security measures such as frame counting.

To address the above attack, we explore the LoRaWAN transmitters’ traits that can be extracted from their transmitted signals. A recent study [8] exploited the LoRaWAN transmitters’ distinct frequency biases (FBs) in generating the chirps to disentangle colliding frames. The biases are mainly caused by the manufacturing imperfections of the transmitters’ internal oscillators. Inspired by this, we inquire whether the replay step of the frame delay attack introduces extra detectable FBs. Different from [8] that only needs a coarse-grained analysis to detect multiple peaks in the frequency domain, we will need high-precision FB estimation to detect a small extra FB introduced by the malicious replayer.

To this end, we propose a SoftLoRa gateway that integrates a low-power software-defined radio (SDR) receiver [2] with a commodity LoRaWAN gateway to capture and analyze the received radio signals. Compared with an alternative solution of using a single full-fledged SDR transmitter such as USRP to demodulate and analyze the received signals in software, our SoftLoRa gateway is advantageous in its efficient hard-speed demodulation while its SDR receiver is used for defense only. Based on LoRa’s Chirp Spread Spectrum (CSS) modulation method, we develop a set of signal processing algorithms for SoftLoRa to estimate the FB. From our experiments with 16 LoRaWAN end devices, we show that (i) with a received signal-to-noise ratio (SNR) of down to \(-25\) dB, SoftLoRa achieves a resolution of 120 Hz in estimating the transmitter’s FB, which is just 0.14 parts-per-million (ppm) of the channel’s central frequency of 869.75 MHz; (ii) the frame replay by a USRP introduces an additional FB of at least 543 Hz (i.e., 0.62 ppm), exceeding SoftLoRa’s FB estimation resolution of 0.14 ppm. Thus, SoftLoRa can track FB to detect the replay step of the frame delay attack. Note that the detection does not require uniqueness or distinctiveness of LoRaWAN end devices’ FBs, because it detects the FB changes caused by the replay, rather than identifies the transmitter.

In this paper, we make the following contributions:

- We implement a stealthy frame delay attack with quantified parameter ranges against a commodity LoRaWAN platform, alerting to the insecurity of any system functions that rely on timely delivery of frames such as the gateway’s uplink data timestamping.
- We design time-domain signal processing algorithms for accurately estimating LoRaWAN end devices’ FBs. The FB estimation requires accurate timestamping of the signal arrival. We achieve microseconds signal timestamping accuracy, improving our understanding on the timestamping accuracy for narrowband LoRaWAN signals and also echoing the results in [18] that are obtained using a different approach.
- With the accurate FB estimation, our SoftLoRa gateway can reliably detect the frame delay attack. The SoftLoRa gateway enables an attack-aware lightweight approach to timestamping the data generated by LoRaWAN end devices that run little or even no code for timestamping. The simplicity, low overhead, and attack awareness are highly desired in real-world systems.

The rest of this paper is organized as follows. §2 reviews related work. §3 details the synchronization-free data timestamping in LoRaWAN. §4 studies the frame delay attack. §5 presents the SoftLoRa gateway design. §6 develops the microseconds-accurate signal timestamping needed by FB estimation. §7 studies LoRa’s FB and uses it to counteract the frame delay attack. §8 presents the results of the experiments in real environments. §9 concludes this paper.

2 RELATED WORK

The communication performance of LoRaWAN has received increasing research. A theoretical capacity analysis for LoRaWAN is presented in [14]. LoRaWAN’s communication performance is profiled via field measurements [13, 17, 26]. Marcelis et al. [13] propose a coding scheme for data recovery. The Choir system [8] exploits the diverse FBs of the LoRaWAN end devices to decode colliding frames from different end devices. However, it does not develop an FB estimation algorithm. The Charm system [7] 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 transmission power. Several recent studies [11, 16, 24, 27] have devised various backscatter designs for LoRa to reduce the power consumption of end devices. However, all these existing studies focus on understanding and improving the data communication performance of LoRaWAN [7, 8, 13, 14, 17, 26], or reducing power consumption via backscattering [11, 16, 24, 27]. None of them specifically addresses efficient data timestamping, a basic system function of many LoRaWAN-based systems.

LongShoT [18] 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. LongShoT...
LoRa 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. Moreover, as we will discuss in §3.2, the timestamping approach based on the prior clock synchronization will introduce considerable overhead. If highly accurate (e.g., microseconds accuracy) timestamping is not needed, it is wise to adopt the synchronization-free approach with the security enhancement presented in this paper.

Security of LoRaWAN has received limited research. In [4], Aras et al. discuss several possible attacks against LoRa, including key compromise, frame replay, and jamming. The first two attacks need prior physical attacks such as memory extraction and node reset. Their jamming simply aims at subverting the victim receiver’s frame decoding. In this paper, we additionally examine the timing of the jamming such that the victim gateway does not alert the OS. In [5], a selective jamming attack against certain receivers and/or certain application frames is studied. From our results in §4.3, the selective jamming in [5] cannot be stealthy because it cannot start jamming until the frame header is decoded. As a result, the selective jamming will corrupt the payload, leading to integrity check failures and alerts. In [20], Robyns et al. apply deep learning for LoRa transmitter identification based on the received baseband signal. Their approach can only identify the source transmitter as one of the considered transmitters that the trained deep model captures. It cannot be used to detect the malicious replayer that is in general out of the deep model.

3 DATA TIMESTAMPING IN LORAWAN

3.1 LoRaWAN Primer

LoRa is a PHY layer technique that adopts a CSS modulation and works in ISM bands (e.g., US 915 MHz and EU 868 MHz). 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 will require the end devices to have synchronized clocks, incurring considerable overhead as we will analyze shortly. Class C requires the end devices to be in the listening mode all the time. Clearly, Class C is not for low-power IoT objects. In this paper, we focus on the energy-efficient Class A, because it is supported by all commodity LoRaWAN platforms. To the best of our knowledge, no commodity LoRaWAN platforms have out-of-the-box support for Class B – the system developers will need to engineer the needed clock synchronization first.

3.2 Sync-Based vs. Sync-Free Timestamping

Data timestamping, i.e., to record the time of interest in terms of wall clock that is meaningful to the data, is a basic system function required by the monitoring data collection applications based on LoRaWAN. For a sensor measurement, the time of interest is the time instant when the measurement is taken by the end device. WSNs largely adopt the synchronization-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 time delays. Thus, although the clock synchronization introduces additional complexity to the system design, it has become a standard component for systems requiring data timestamping. However, in LoRaWANs with much less communication capacity due to their narrowband nature, the overhead of the clock synchronization cannot be ignored.

Now, we present an example to illustrate. Assume we have a microseconds or milliseconds accurate clock synchronization approach for LoRaWAN. Typical crystal oscillators found in microcontrollers and personal computers have drift rates of 30 to 50 ppm [10]. Without loss of generality, we adopt 40 ppm for the following calculation. Under this drift rate, an end device will need 14 synchronization sessions per hour to ensure a sub-10ms clock error. These synchronization sessions may 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 specified by The European Telecommunications Standards Institute [12]. Although the synchronization information may be piggybacked to the data frames as in [18], a low-rate monitoring application may have to send the frames more frequently just to satisfy the time keeping requirement. In addition, the timestamps for data records also occupy frame payload space. For instance, if each frame with 30-byte payload contains an eight-byte timestamp [18] for the data in the frame, 27% of the effective bandwidth is used to convey timestamps. From the above example, we can see that the clock synchronization service and the data timestamps may consume a significant fraction of LoRaWAN’s communication capacity.

In this paper, we consider a synchronization-free approach. Specifically, a LoRaWAN end device still records the times of interest in terms of its unsynchronized clock. Right before sending a number of data records using a frame, the device replaces the 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 buffer 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 buffer time needs to be within 4.1 minutes. As a result, 18 bits will be sufficient to represent an elapsed time with 1 ms resolution. Since the one-hop signal propagation time from the end device to the gateway is negligible for millisecond-level systems, the gateway can easily reconstruct the global timestamps based
on the frame arrival time and the elapsed times contained in the frame. Compared with the synchronization-based approach, this synchronization-free approach avoids the communication overhead and implementation complexity caused by clock synchronization. It can also reduce the frame payload use for time information (e.g., 18 bits for elapsed time versus 8 bytes for complete timestamp). In particular, if the end device can immediately transmit any newly generated sensor/event data, the elapsed time payload is even not needed.

The synchronization-free approach is lightweight and commensurate with LoRaWAN’s scarce communication resources. Its accuracy can be affected by (1) the delay from the end device’s application code requesting frame transmission to the actual emission of radio signal by the LoRaWAN chip and (2) the gateway’s accuracy in timestamping the radio signal arrival. An existing study [9] shows that these issues cause a sum uncertainty of about 3 ms only. By resorting to lower-level accesses to the radio chip such as in [18], the uncertainty will be further reduced. Thus, the synchronization-free approach can achieve ms or sub-ms timestamping accuracy.

4 SECURITY OF SYNCHRONIZATION-FREE TIMESTAMPING

The synchronization-free approach brings various benefits including implementation simplicity and bandwidth usage saving. Due to the broadcast and long-range nature of LoRaWAN communications, it is also important to investigate the security aspect. In this section, we consider a frame delay attack that maliciously manipulates the propagation time. Thus, the attack will directly affect the synchronization-free timestamping approach that assumes near-zero signal propagation time. In this section, §4.1 defines the threat model; §4.2 presents the implementation of the attack; §4.3 experimentally investigates several important parameter settings to implement the attack; §4.4 discusses a simple attack detection approach and its shortcomings.

4.1 Frame Delay Attack

If an adversary introduces a malicious time delay to the deliveries of the uplink frames from end nodes, the timestamps generated by the gateway will be compromised. We formally define the threat model as follows.

**Definition 1 (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 from end nodes. The malicious delay for any frame is finite. Moreover, the frame cannot be tampered with because of cryptographic protection.

4.2 Implementation of Frame Delay Attack

This section presents a practical implementation of the frame delay attack via a combination of jamming and replay. §4.2.1 presents the implementation principle; §4.2.2 discusses several practical issues in implementing the attack.

4.2.1 Implementation principle. Fig. 1 illustrates the attack implementation. The adversary sets up two malicious devices called eavesdropper and replayer that are close to the end device and the gateway, respectively, to delay the delivery of the uplink frame. The

**Figure 1: Implementing frame delay attack by stealthy jamming and replay.**

attack consists of three steps. (1) At the beginning, both the eavesdropper and the replayer listen to the LoRa communication channel between the end device and the gateway. Once the replayer detects an uplink frame transmission from the end device to the gateway, it jams the gateway’s frame reception by transmitting a jamming frame. In §4.3, we will investigate experimentally a stealthy jamming method such that the victim gateway based on an off-the-shelf LoRa radio does not raise any warning message to the application layer. Meanwhile, once the eavesdropper detects an uplink frame transmission from the end device to the gateway, it records the radio waveform of the frame. Note that the replayer may properly control the transmission power of the jamming such that the jamming frame can jam the victim gateway, while not corrupting the radio waveform recorded by the eavesdropper. When the replayer is far away from the eavesdropper, delicate transmission power control of the replayer may be waived because the jamming signal will be weak at the eavesdropper after propagation attenuation. This is experimentally demonstrated in §8. (2) The eavesdropper sends the recorded radio waveform data to the replayer. (3) After a time duration of $\tau$ seconds from the onset time of the legitimate frame transmission, the replayer replays the recorded radio waveform received from the eavesdropper. The above jam-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 is unaware of the earlier jamming and the integrity of the replayed frame is preserved, the gateway will accept the replayed frame even if it checks the cryptographically protected check sum and frame counter. The above process introduces a delay of $\tau$ seconds to the delivery of the frame.

Although the above descriptions focus on an end device, the attack setup illustrated in Fig. 1 can affect the uplink frames from many end devices close to the eavesdropper, as long as the strength of the signal from an end device at the eavesdropper is much higher than that of the jamming signal from the replayer.

4.2.2 Several practical issues. To increase the stealthiness of the replay attack, the replayer can well control the transmission power of the replay such that only the victim gateway can receive the replayed frame. Although the volume of the recorded radio waveform data is often large, the eavesdropper can transmit the data to the jammer via a separate communication link (e.g., LTE).

As the adversary should delay 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 §4.3, the jamming should start after several
chirps and before tens of chirps of the frame transmission. Thus, a
time duration of one chirp for sensing the direction of the transmis-
sion will not impede the timeliness of the jamming. And the end
device identification can be achieved by using the uplink frame’s
source node ID (if not encrypted) or extracting the end device’s
frequency trait (see §7.2).

4.3 Attack Experiments
We conduct experiments to investigate the parameter settings for
implementing the stealthy jamming attack. In the experiments, we
set up two RN2483-based LoRa nodes as the end device and the
gateway. We use a third LoRa node as the jammer. The distance
between the gateway and the end device is about 5 m; the distance
between the jammer and the gateway is about 1 m. We primarily
investigate the timing of successful stealthy jamming. From our
experiments, there are three critical time windows (denoted by \(w_1\),
\(w_2\), and \(w_3\)) after the onset time of the legitimate frame transmis-

Figure 2: Jamming attack time window.

sion from the end device to the gateway (denoted by \(t_0\)). These time
windows are illustrated in Fig. 2. If the onset time of the jamming
is in \([t_0, t_0 + w_1]\), the gateway most likely receives the jamming
frame from the jammer only; if it is in \([t_0 + w_1, t_0 + w_2]\), the gate-
way can receive neither frames and the gateway’s RN2483 raises
no alerts; if it is in \([t_0 + w_2, t_0 + w_3]\), the RN2483 reports frame cor-
ruption; if it is after \(t_0 + w_3\), the gateway can receive both frames
sequentially. Therefore, the time window \([t_0 + w_1, t_0 + w_2]\) is called
effective attack window. The jamming attack with an onset time in
this window is stealthy to the gateway.

We measure \(w_1\), \(w_2\), and \(w_3\) under various settings for the spreading
factor and the payload size of the legitimate frame. Table 1 sum-
marizes the results. From the results for \(w_1\), the jamming should
start after the 5th chirp of the legitimate frame transmission. This
is because: i) the gateway’s LoRa chip has not locked the legitimate
preamble until the 6th chirp and it will re-lock the jamming frame’s
preamble due to higher signal strength if the jamming frame starts
before the 5th chirp of the legitimate frame; ii) the gateway’s LoRa
chip locks the legitimate preamble from the 6th chirp and will sim-
ply drop 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 hardware cannot determine
whether itself is the intended recipient and therefore drops the re-
ceived data. Thus, the stealthy jamming should start after the 5th
chirp of the legitimate frame transmission.

We can also see that \(w_2\) increases exponentially with the spreading
factor. This is because: i) the total time for transmitting the pre-
amble and frame header increases exponentially with the spreading
factor; ii) corruption of the payload after the frame header will lead
to an integrity check error and a warning message. The \(w_3\)
is roughly the time for transmitting the legitimate frame. Thus, if
the jamming onset time is after \(t_0 + w_3\), both the legitimate and the
jamming frames can be decoded.

The above experiments show that, there is a time window for
the jamming to corrupt the preamble partially and the frame header
such that the victim simply drops the received data and raises no
alerts. Jamming starting in this window is stealthy.

4.4 Discussion on a Simple Attack Detector
A simple attack detection approach is to perform round-trip tim-
ing and then compare the measured round-trip time with a thres-
hold. However, this approach will need 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 multi-
ple 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 trans-
mision must be unicast, in response to a precedent uplink trans-
mission [22]. Thus, the round-trip timing approach matches poorly
with the uplink-downlink asymmetry characteristic of LoRaWAN.
Moreover, as the frame delay attacks will be rare events, continu-
ously using downlink acknowledgements to preclude the threat is
a low cost-effective solution. Differently, in this paper, we will de-
sign advanced signal processing algorithms that run at the gate-
way to analyze the received radio signals and detect the frame de-
lay attack. Our technique will be a cost-effective solution for the
awareness of the attack existence without introducing any commu-
nications overhead or any modifications to the hardware/software
of the end devices.

5 SoftLoRa Gateway
This section presents the SoftLoRa gateway used to achieve secure
data timestamping in LoRaWANs.

5.1 SoftLoRa Hardware
To develop the attack detection capability, we integrate an SDR
receiver with a LoRaWAN radio to monitor the LoRa PHY layer.
Various cheap (US$25 only [2]) and low-power SDR receivers are
widely available now. In this paper, we use RTL-SDR USB dongles
based on the RTL2832U chipset [2], which were originally designed

| Spreading factor \(S\) | Chirp time | Preamble time | Payload (byte) | \(w_1\) | \(w_2\) | \(w_3\) |
|----------------------|-----------|---------------|----------------|--------|--------|--------|
| 7                    | 1.024     | 8.2           | 10             | 5      | 28     | 141    |
| 8                    | 2.048     | 16.4          | 20             | 5      | 38     | 156    |
| 9                    | 4.096     | 32.8          | 30             | 6      | 41     | 165    |

* Unit for chirp time, preamble time, \(w_1\), \(w_2\), \(w_3\) is millisecond.
to be DVB-T TV tuners. The RTL-SDR supports continuous tuning in the range of [24, 1766] MHz, which covers the LoRaWAN bands (i.e., 430, 433, 868, 915 MHz). It can operate at 2.4 Mspss reliably for extended time periods. Thus, the sampling resolution is 1/2.4 Mspss = 0.42 μs.

The research of this paper is conducted based on a SoftLoRa hardware prototype that integrates a Raspberry Pi 3 Model B single-board computer (as the host), a Cooking Hacks LoRaWAN shield [1] (as the LoRa transceiver), and an RTL-SDR USB dongle (as the SDR receiver). Fig. 3 shows the prototype. The LoRaWAN shield consists of a Microchip RN2483 chipset, an 868 MHz antenna, and a general-purpose input/output (GPIO) interfacing circuit. RN2483 is based on Semtech SX1276, a major commodity LoRa chip on the current market. Mounted on the host via GPIO pins, the shield can be controlled using a C++ library from Cooking Hacks. An 868 MHz antenna is also integrated with the RTL-SDR to improve signal reception. The RTL-SDR is plugged into a USB port of the gateway’s host computer.

The SDR receiver will be used to capture the radio signal over a time duration of the first two CSS chirps of an uplink frame. The first sampled chirp is used to extract PHY-layer timestamp (cf. §6), whereas the second sampled chirp is used to extract the FB of the transmitter (cf. §7). The microseconds-accurate PHY-layer timestamp is a prerequisite of the FB estimation. As only two chirps’ radio waveform will be analyzed, SoftLoRa will have manageable computation overhead, which can be performed by embedded computing boards such as Raspberry Pi.

An alternative approach is to adopt a full-fledged SDR transceiver (e.g., USRP) to design a highly customized LoRaWAN gateway with PHY signal analysis capability. However, this approach will lose the factory-optimized hardware-speed LoRa demodulation built in the commodity LoRaWAN platforms. Moreover, full-fledged SDR transceivers (e.g., USRP N210) are often 10x more expensive than

![Figure 3: SoftLoRa hardware prototype consisting of Raspberry Pi (host), LoRaWAN shield (LoRa transceiver), RTL-SDR USB dongle (SDR receiver).](image)

![Figure 4: Software architecture of SoftLoRa gateway. Bottom part is end device; upper part is the gateway; solid arrows are local data flows; dashed arrows are transmissions.](image)

![Figure 5: Analog signal processing in SDR receiver.](image)

SoftLoRa. The low-cost, low-power, listen-only RTL-SDR suffices for developing the frame delay attack detector.

### 5.2 CSS Reception using SDR Receiver

This section models the LoRa signal reception using the SDR receiver. It will be a basis for understanding the challenges of achieving PHY-layer timestamping in §6 and developing accurate FB estimation algorithms in §7.

LoRa adopts CSS modulation. A chirp is a finite-time band-pass signal with time-varying frequency that swaps the whole bandwidth of the communication channel in a linear or non-linear manner. Let $A(t)$ and $f(t)$ denote the instantaneous amplitude and frequency of the chirp at the time instant $t$. Thus, the chirp, denoted by $s(t)$, is

$$s(t) = A(t) \sin \left(2\pi \int_0^t f(x)dx + \theta_{Tx}\right),$$

where $\theta_{Tx} \in [0, 2\pi)$ is the LoRa transmitter’s phase that is usually unknown.

Fig. 5 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 $\sin(2\pi f_c t + \theta_{Rx})$ and $\cos(2\pi f_c t + \theta_{Rx})$, where $f_c$ is a specified frequency and $\theta_{Rx}$ 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

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

$$= \frac{A(t)}{2} \left(\cos \left(2\pi \int_0^t f(x)dx - 2\pi f_c t + \theta_{Tx} - \theta_{Rx}\right) - \cos \left(2\pi \int_0^t f(x)dx + 2\pi f_c t + \theta_{Tx} + \theta_{Rx}\right)\right), \quad (1)$$

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

$$= \frac{A(t)}{2} \left(\sin \left(2\pi \int_0^t f(x)dx - 2\pi f_c t + \theta_{Tx} - \theta_{Rx}\right) + \sin \left(2\pi \int_0^t f(x)dx + 2\pi f_c t + \theta_{Tx} + \theta_{Rx}\right)\right), \quad (2)$$

The high-frequency components in Eqs. (2) and (4) 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
We perform PHY-layer signal timestamping by detecting the onset of the LoRaWAN transceiver demodulates the received radio signal and passes the frame to the gateway’s computer host. PHY-layer signal processing algorithms are applied on the LoRa signal after down-conversion by the SDR receiver to pick precisely the arrival time of the radio signal (i.e., PHY timestamping), 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 biases associated with the transmitter ID claimed in the current frame. The gateway will be aware of such frame replay attack and drop the replayed frame. Note that SoftLoRa gateway uses the SDR receiver to obtain FBs, rather than to decode the frame.

In this rest of this paper, we will present the signal processing algorithms for PHY signal timestamping in §6, FB estimation and attack detection in §7. Note that microseconds-accurate PHY signal timestamping is a prerequisite of the FB estimation.

6 SIGNAL TIMESTAMPING FOR LORA

In this section, we present our LoRa signal timestamping approach on SoftLoRa (§6.1) and evaluation results (§6.2). We aim to achieve microseconds accuracy in timestamping the LoRa signal.

6.1 LoRa Signal Timestamping using SDR Receiver

We perform PHY-layer signal timestamping by detecting the onset time of LoRa frame preamble. With preamble onset timestamp, the FB estimation algorithm can select the right segments of I and Q data to work on. In this section, we first model the preamble’s I and Q data received by the SDR receiver and then discuss the preamble onset time detection.

6.1.1 Preamble received by SDR receiver. In LoRaWAN, by default, the preamble of an uplink frame (from end device to gateway) or a downlink frame (from gateway to end device) consists of eight up or down linear chirps, respectively [22]. Let $f_c$ and $W$ represent the central frequency and bandwidth of the used LoRa channel. In all numerical examples and experiments of this paper, we use a LoRaWAN channel with $f_c = 869.75$ MHz and $W = 125$ kHz.

The instantaneous frequency of an up chirp increases linearly with time, from the lowest frequency (i.e., $f_c - W/2$) to the highest frequency (i.e., $f_c + W/2$) of the channel. It is given by $f(t) = W^2/τ^2 \cdot t - W/2 + f_c$ for $t \in \left[0, \frac{2\tau}{\pi W}\right]$, where $S$ is the spreading factor and $\frac{2\tau}{\pi W}$ is the chirp time. The $S$ is an integer within $[6, 12]$. A larger spreading factor increases the chirp time and thus decreases the data rate when the chirps are used to encode data. However, longer chirp times increase the SNR at the receiver and thus the communication range. By following the analysis in §5.2, the I and Q components of the received up chirp are given by $I(t) = \frac{A(t)}{2} \cos \Theta(t)$ and $Q(t) = \frac{A(t)}{2} \sin \Theta(t)$, where the instantaneous angle $\Theta(t)$ is $\Theta(t) = \frac{2W^2}{\pi^2} t^2 - \pi W t + \theta$. The analysis for the down chirp with linearly decreasing frequency is similar; we do not elaborate here.

Fig. 6 shows the I data and the spectrogram of an ideal up chirp sampled at 2.4 Msps. The parameters of the up chirp are $A(t) = 2$, $\theta = 0$, and $S = 7$. Thus, the chirp time $\frac{2\tau}{\pi W}$ is 1.024 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 over the chirp time of 1.024 ms.

6.1.2 Preamble onset time detection. Detecting the onset time of the preamble is non-trivial. To understand the challenges, in this section, we discuss two possible methods and their inefficacy. Then, we present two other promising methods.

As the up chirp exhibits a clear time-frequency pattern as shown in Fig. 6, a possible approach to locating the first up chirp of the preamble is to analyze the spectrogram of the received I and Q data. However, the spectrogram inevitably has reduced time resolution. For instance, the time resolution of the spectrogram in Fig. 6 is $1024 \mu s/20 \approx 50 \mu s$, which impedes high-resolution PHY-layer timestamping.
AIC is a promising solution for our problem. The algorithm 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 onset time estimate. The vertical line in Fig. 9(b) represents the onset time detected by the AIC detector.

Note that as both the envelope detector and the AIC detector formulate the onset time detection as optimization problems, they do not need any detection threshold.

Figure 9: Preamble onset time detection results.

Matched filter is a widely adopted symbol detection technique for constant carrier frequency modulation schemes, e.g., ASK and PSK. As a coherent detection technique, the matched filter requires that the receiver is phase-locked to the transmitter (i.e., $\theta = 0$) to achieve the best symbol detection accuracy. However, as LoRa adopts time-varying frequency, it is difficult for the SDR receiver to estimate the transmitter’s phase $\theta_T$. In fact, low-end SDR receivers such as the RTL-SDR used in this work do not provide phase-lock capability. As a result, the phase difference $\theta$, which is a critical factor affecting the shape of $I(t)$ and $Q(t)$, will be random. Fig. 7 shows the ideal $I(t)$ traces of the up chirp when the $\theta$ is 0 and $\pi$. The waveform shapes are different. Thus, it is impossible to define a template shape for the matched filter to work. Moreover, as analyzed in §7, the FB of the LoRa transmitter will significantly alter the shapes of the $I$ and $Q$ signals. Fig. 8 shows the actual $I$ trace captured by the SDR receiver. The dip center shift is caused by the FB. Thus, due to the random $\theta$ and the LoRa transceiver’s FB, the matched filter is not promising.

To investigate the signal timestamping of LoRa, we consider two time-domain signal processing techniques:

**Envelope detector:** First, we apply the Hilbert transform to extract the amplitude envelope of the $I$ or $Q$ signal. Then, the sample with the largest ratio between its envelope amplitude and the previous sample’s envelope amplitude is yielded as the preamble onset. Fig. 9(a) shows the extracted amplitude envelope and the ratio curve over time. We can see that the maximum ratio well indicates the onset time of the preamble.

**AIC detector:** The autoregressive Akaike Information Criterion (AIC) algorithm [21] has been widely adopted to estimate the arrival time of seismic wave with an accuracy of a single sampling point. As the $I$ and $Q$ signals are similar to the seismic waves, the AIC is a promising solution for our problem. The algorithm 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 onset time estimate. The vertical line in Fig. 9(b) represents the onset time detected by the AIC detector.

Table 2: Error upper bound ($\mu s$) by envelope (ENV) detector and AIC.

| SNR (dB) | ENV | AIC |
|---------|-----|-----|
| 0       | 5.4 | 1.0 |
| -10     | 4.5 | 1.3 |
| 0       | 4.8 | 1.5 |
| 10      | 5.2 | 0.8 |
| 20      | 1.9 | 1.7 |
| 30      | 5.2 | 1.5 |
| 40      | 6.3 | 1.3 |
| 50      | 5.2 | 1.9 |

Figure 10: AIC timestamping error vs. received SNR.

6.2 Signal Timestamping Accuracy Evaluation

We conduct a set of experiments to evaluate the accuracy of our PHY-layer signal timestamping. The accuracy of signal timestamping is restricted by the sampling rate of the SDR. When the real onset time is between two consequent samples, the real onset time is unknown while the range can be confirmed. Thus, we use the upper bound of the signal timestamping error to evaluate our approach. In §8, we will also measure this error metric in a multistory building and a campus, where the signal propagation will be affected by the noise and the attenuation with distance.

Tables 2 shows the error upper bound $\Delta t$ measured for the envelope detector and the AIC detector when operating on the $I$ and $Q$ data in ten experiments, respectively. We can see that the AIC detector achieves higher accuracy. In particular, the timing errors of the AIC detector are less than 2 $\mu s$.

Then, we evaluate the impact of random noises on AIC’s signal timestamping accuracy. We artificially add zero-mean Gaussian noises to the collected high-SNR $I$ and $Q$ traces. Then, we apply the AIC detector on the noise-added traces to detect the LoRa signal onset time. The SNR in dB is defined as $10 \log_{10} \frac{\text{signal power}}{\text{noise power}}$. Fig. 10 shows the results. From our measurements in a multistory building (cf. §8), the received SNR ranges from 13 dB to –1 dB. From Fig. 10, the average timestamping error is expected to be within 20 $\mu s$. This will be confirmed by real experiments in §8. When the SNR is –20 dB, which is the lower limit for reliable demodulation [3], from Fig. 10, the average error will be within 25 $\mu s$.

7 FREQUENCY BIAS-BASED FRAME DELAY ATTACK DETECTION

Internal oscillators for generating carriers generally have frequency biases (FBs) of one to hundreds of ppm, due to manufacturing imperfection. This section develops algorithms for estimating LoRa transmitters’ FBs. We also investigate whether such FBs can be used to detect the replay attacks.
7.1 FB Estimation

This section describes algorithms for estimating the transmitter’s FB based on an up chirp in the preamble. The method for a down chirp is similar. First, we analyze the impact of the transmitter’s and SDR receiver’s FBs (denoted by $\delta_{Tx}$ and $\delta_{Rx}$) on the $I$ and $Q$ traces. The up chirp’s instantaneous frequency accounting for $\delta_{Tx}$ is

$$f(t) = \frac{W^2}{2^5} \cdot t - \frac{W}{2} + f_c + \delta_{Tx}, \quad t \in \left[ 0, \frac{2^5}{W} \right].$$

The two local unit-amplitude orthogonal carriers generated by the SDR receiver are $\sin(2\pi (f_c + \delta_{Rx})t + \theta_{Rx})$ and $\cos(2\pi (f_c + \delta_{Rx})t + \theta_{Rx})$. Following the analysis in §5.2, the $I$ and $Q$ components of the received up chirp can be derived as $I(t) = \frac{W(t)}{2\pi} \cos(\Theta(t))$ and $Q(t) = \frac{W(t)}{2\pi} \sin(\Theta(t))$, where the instantaneous angle $\Theta(t)$ is given by

$$\Theta(t) = \pi \frac{W(t)}{2^5} t^2 - \pi Wt + 2\pi \delta t + \theta, \quad \delta = \delta_{Tx} - \delta_{Rx}. \quad (5)$$

The difference between the transmitter’s and receiver’s FBs, i.e., $\delta$, affects the $I$ and $Q$ waveforms. Fig. 11 shows the numerical results of $I$ traces when $\delta = -25$ kHz and $\delta = +25$ kHz. The non-zero $\delta$ shifts the axis of symmetry of the $I$ trace. We observed this in the actual $I$ data shown in Fig. 8.

For a certain SDR receiver, the FB estimation problem is to estimate $\delta$ from the captured $I$ and $Q$ traces. We do not need to estimate $\delta_{Tx}$, because for a certain SDR receiver with a nearly fixed $\delta_{Rx}$, a change in $\delta$ indicates a change in $\delta_{Tx}$ and a replay attack. In what follows, we describe two approaches based on linear regression and least squares formulations. The least squares approach keeps robust when the SNR is very low but has higher computation overhead.

7.1.1 Linear regression approach. From Eq. (5), the $I$ traces are obtained by linear regression based on the data pairs $(t, \Theta(t) - \pi W t + \delta t + \theta, \frac{W(t)}{2\pi})$, where $t \in \left[ 0, \frac{2^5}{W} \right]$. The rectification is as follows. The $k$ is initialized to be 0 when $t = 0$. As $t$ increases, if atan2($Q(t), I(t)$) jumps from $-\pi$ to $\pi$, $k$ decreases by one; if atan2($Q(t), I(t)$) jumps from $\pi$ to $-\pi$, $k$ increases by one. Note that the traces $I(t)$ and $Q(t)$ where $t \in \left[ 0, \frac{2^5}{W} \right]$ 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 $\frac{2^5}{W}$ seconds.

Fig. 12 shows the intermediate results of the FB extraction. Fig. 12(a) shows the real $I(t)$ and $Q(t)$ traces of an up chirp emitted by an RN2483-based LoRa transmitter and captured by the SDR receiver. Fig. 12(b) shows the $\pi W t + \delta t + \theta, \frac{W(t)}{2\pi}$ obtained by rectifying the result in Fig. 12(b) with $2k\pi$. Fig. 12(d) shows $\Theta - \pi W t + \delta t + \theta, \frac{W(t)}{2\pi}$ We can see that it is indeed a linear function of time $t$, which conforms to our analysis. By applying linear regression to the result in Fig. 12(d), the FB $\delta$ (i.e., the slope of the fitted line divided by $2\pi$) is estimated as $-22.8$ kHz. Note that the nominal central frequency is $869.75$ MHz. The FB is merely $26$ ppm of the central frequency.

We use an SDR receiver to estimate the FBs of 16 RN2483-based LoRa transmitters. In each test for a LoRa transmitter, the distance between the transmitter and the SDR receiver is about 5 m. The error bars labeled “original” in Fig. 13 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 are from $17$ kHz to $25$ kHz, which are about $20$ ppm to $29$ ppm of the nominal central frequency of $869.75$ MHz.

Some nodes have similar FBs, e.g., Node 3, 8, and 14. Recall our discussion in §4.2.1 that the adversary may wish to identify the nodes based on FBs to selectively attack a certain node. To address the issue of similar FBs, the adversary may jointly use the FBs
and the received signal strengths that are affected by the transmitters’ geographic locations to fingerprint the transmitters. Differently, the detection of the replay attack is based on the fact that the replayed transmission will have a different FB. In other words, the attack detection does not require distinct FBs among different transmitters.

As the linear regression approach has a closed-form formula to compute δ, it has a complexity of $O(t)$ in the search of the δ solution. However, the inverse tangent rectification is susceptible to small errors, hence the requirement is met when the second chirp of the preamble is used. The FB estimated from a frame that is detected to be a replayed one should not be used to update the database. Through the FB monitoring, the SoftLoRa gateway can detect the replay attack and will not use the replayed frame to do data timestamping. Thus, the data timestamps will not be spoofed by the frame delay attack.

We discuss two notes about the detection mechanism. First, to bypass the above detection mechanism, the attacker will need SDRs with FBs within 0.14 ppm (i.e., the SoftLoRa’s FB estimation accuracy). However, as RF devices typically have FBs of one to tens of ppm, it is difficult to bypass the detection mechanism. Second, the detection does not require uniqueness of the FBs across different LoRa transceivers, because it is based on changes of FB.

8 PERFORMANCE EVALUATION

8.1 Experiments in a Multistory Building

We conduct experiments in a concrete building with six floors. Along its long dimension of 190 meters, the building has three sections and two section junctions. Fig. 15 illustrates a lateral view of the building. We survey the SNR inside the building. We deploy a fixed LoRaWAN transmitter in Section A on the 3rd floor, as illustrated by the triangle in Fig. 15. Then, we carry a mobile SoftLoRa 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. We use the method described in §7.1.2 to compute SNR. In each section, we measure traces are scaled to achieve different SNRs. From Fig. 14, we can see that the FB estimation errors are below 120 Hz (i.e., 0.14 ppm), when the SNR is down to –25 dB for both types of noises.

7.2 Replay Attack Detection

The malicious replacer also has an FB. We use a USRP N210 SDR transceiver to build a replacer. The error bars labeled “replayed” in Fig. 13 show the FBs estimated from the LoRa signals received by the SoftLoRa’s SDR receiver when the USRP replays the radio waveform captured by itself in the experiments presented in §7.1. 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 in §7.1.2, the additional FBs caused by the replay attack can be detected.

Based on the above observation, we describe an approach to reliably detect the replay attacks. We assume that a SoftLoRa gateway has a database of the FBs of the nodes with which it communicates. This database can be built offline or at runtime using its SDR receiver in the absence of attacks. To address the neighbors’ time-varying radio frequency skews due to run-time conditions like temperature, the SoftLoRa gateway 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 SoftLoRa gateway checks whether the FB of the current received frame is within the FB range of the claimed source node in the database. This detection approach is applied after the SoflLoRa gateway decodes the frame to obtain the claimed source node ID. The FB estimated from a frame that is detected to be a replayed one should not be used to update the database. Through the FB monitoring, the SoftLoRa gateway can detect the replay attack and will not use the replayed frame to do data timestamping. Thus, the data timestamps will not be spoofed by the frame delay attack.

We discuss two notes about the detection mechanism. First, to bypass the above detection mechanism, the attacker will need SDRs with FBs within 0.14 ppm (i.e., the SoftLoRa’s FB estimation accuracy). However, as RF devices typically have FBs of one to tens of ppm, it is difficult to bypass the detection mechanism. Second, the detection does not require uniqueness of the FBs across different LoRa transceivers, because it is based on changes of FB.
A1 of the 3rd floor and Section C3 of the 6th floor, respectively. The LoRa signals are significantly attenuated after passing through the air. We conduct the following experiments. Default settings used in our experiments are: spreading factor $S = 12$; $f_c = 869.75$ MHz; $W = 125$ kHz.

### 8.1.1 Full implementation of frame delay attack

We deploy a commodity LoRaWAN end device and a SoftLoRa gateway 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 due to the signal attenuation. A minimum spreading factor of 8 is needed for reliable data communications between them. In rural environments, this spreading factor can be used to achieve communication ranges of three to four kilometers [14]. Thus, our multistory building environment creates realistic challenges similar to those caused by long geographic distances. Following the attack scheme in Fig. 1, we deploy two USRP N210 stations as the eavesdropper and the replayer, next to the end device and the gateway, respectively. We fully implement the attack steps described in §4.2.1. We set the transmission power of the master to the maximum level, i.e., 15. The transmission power of the jammer signal is 14.1 dBm. After crossing multiple building levels, the jamming signal arriving at the eavesdropper is weak. As a result, when the replayer replays the jamming signal, the eavesdropper is unable to capture the jamming signal due to the weak jammer. The replayer can successfully decode the jamming signal, making the replay attack stealthy to the gateway. We fully implement the attack steps described in §4.2.1. We set the transmission power of the master to the maximum level, i.e., 15. The transmission power of the jammer signal is 14.1 dBm. After crossing multiple building levels, the jamming signal arriving at the eavesdropper is weak. As a result, when the replayer replays the jamming signal, the eavesdropper is unable to capture the jamming signal due to the weak jammer. The replayer can successfully decode the jamming signal, making the replay attack stealthy to the gateway. This experiment shows the credibility of the frame delay attack in a building.

### 8.1.2 Signal timestamping accuracy

We measure the signal timestamping error metric defined in §6.2 when the mobile node is at different locations in the building. Note that this signal timestamping is a prerequisite of the subsequent FB estimation. The numbers shown in the cells of Fig. 15 are the measured timestamping error in $\mu$s when the mobile node is at the corresponding locations.

### 8.1.3 Impact of transmission power on FB estimation

Fig. 16 shows the estimated FBs versus the end device’s transmission power under different settings. The bottom row of black box plots are the FBs estimated by the eavesdropper when the end device transmits the uplink frame with different transmission powers. The middle row of red box plots are the FBs estimated by the SoftLoRa gateway in the absence of the jamming and replay attacks. Thus, the FBs estimated by the eavesdropper and the SoftLoRa gateway are different. This is because that here we use two different USRPs as the eavesdropper and replayer. Their FBs are superimposed. The top row shows the two sites (Site A and B) for deploying the end device and the SoftLoRa gateway are different. This is because that as analyzed in §7.1, the estimated FB contains the transmitter’s and receiver’s FBs $\delta_{Tx}$ and $\delta_{Rx}$. Note that the eavesdropper and the SoftLoRa gateway in general have different FBs. From Fig. 16, the end device’s transmission power has little impact on the FB estimation.

### 8.1.4 Additional FB introduced by replayer

In Fig. 16, the top row of blue box plots are the FBs estimated by the SoftLoRa gateway when the replayer replays the radio waveform recorded by the eavesdropper. We set the gain of the replayer’s USRP to be 0 dB. When the end device adopts a higher transmission power, the replayer also has higher power. By comparing the middle and the top rows, we can see that the replayer attack introduces an additional FB of about 2 kHz, which is 2.3 ppm of the LoRa channel’s central frequency. Therefore, the FB monitoring can easily detect the replay attack. Compared with the results in Fig. 13 showing additional FBs of 0.62 to 0.85 ppm, the FBs in this set of experiments are higher. This is because that here we use two different USRPs as the eavesdropper and replayer; their FBs are superimposed.

### 8.2 Long-Distance Experiments in a Campus

We deploy a LoRaWAN end device and a SoftLoRa gateway in a campus, which are separated by an Euclidean distance of 1.07 km. The signal’s one-way propagation time is 3.57 $\mu$s. The end device and the SoftLoRa gateway are deployed on the roof top of a building and an open stair case of another building, respectively. Fig. 17(c) shows the two sites (Site A and B) for deploying the end device and the SoftLoRa gateway at the campus. Site A is at the roof top of a
building, whereas Site B is in an open stair case of another building. Fig. 17 shows the pictures taken at the two sites. The circled construct in a figure is the building where the other site is located in. We conducted four tests to evaluate the signal timestamping error. It rained heavily during the tests. The measured error upper bounds during the four tests are $3.52 \text{µs}, 2.27 \text{µs}, 6.43 \text{µs}, 0.23 \text{µs}$. We can see that SoftLoRa gateway achieves microsecond signal timestamping accuracy over a distance of one kilometer. This will ensure accurate FB estimation.

9 CONCLUSION

This paper considers the security of a synchronization-free data timestamping approach for LoRaWANs. Specifically, the timestamping of data is performed by the LoRaWAN gateway based on the frame arrival time. The low communication overhead of this approach makes it suitable for the bandwidth-limited LoRaWANs for collecting low-rate data from geographically distributed end devices. However, we show that this approach is susceptible to a basic threat of frame delay attack that can be implemented by a combination of stealthy jamming and delayed replay. As the attack does not need to break the cryptographic protection of the frame, existing security measures prescribed by the LoRaWAN gateway cannot counteract this threat. To address this attack, we propose a gateway design called SoftLoRa that integrates a low-power SDR receiver with a LoRaWAN gateway. We design efficient time-domain signal processing algorithms based on the CSS modulation method to estimate frequency biases of the end devices. Our algorithms achieve a resolution of 0.14 ppm of the carrier frequency, which is sufficient to uncover the additional frequency biases introduced by the replay step of the frame delay attack. In summary, with SoftLoRa gateway, we present a cost-effective defense approach for the low-overhead synchronization-free data timestamping approach against the frame delay attack.

ACKNOWLEDGMENTS

This research was funded by a Start-up Grant at Nanyang Technological University (NTU). We acknowledge Zhenyu Yan for assistance in conducting the long-distance experiments in the NTU campus.

REFERENCES

[1] 2018. LoRaWAN Shield for Raspberry Pi. https://www.cooking-hacks.com/lorawan-shield-for-raspberry-pi-868-mhz-shee-socket.
[2] 2018. RTL-SDR. https://www rtl-sdr.com/.
[3] 2018. SX1276 Datasheet. http://modtronix.com/prod/components/wireless/sx1276.pdf.
[4] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes. 2017. Exploring the Security Vulnerabilities of LoRa. In The 3rd IEEE International Conference on Cybernetics (CYBCONF) 1-6.
[5] Emekcan Aras, Nicolas Small, Gowri Sankar Ramachandran, Stephane Delbruel, Wouter Joosen, and Danny Hughes. 2017. Selective Jamming of LoRaWAN using Commodity Hardware. In The 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous).
[6] Charles Byers. [n. d. ] Fog Computing for Industrial Automation. https://www.controleng.com/articles/fog-computing-for-industrial-automation/.
[7] Adwait Dongare, Revathy Narayanan, Akshay Gade, Anh Long, Artur Balanuta, Swarun Kumar, Bob Iannucci, and Anthony Rowe. 2018. Chaff: Exploiting Geographical Diversity Through Coherent Combining in Low-Power Wide-Area Networks. In The 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN).
[8] Rashad Eletreby, Dianzu Zhang, Swarun Kumar, and Osman Yaygan. 2017. Empowering Low-Power Wide Area Networks in Urban Settings. In ACM SIGCOMM.
[9] Chaojie Gu, Rui Tan, Xin Lou, and Dusit Niyato. 2018. One-Hop Out-of-Band Control Planes for Low-Power Multi-Hop Wireless Networks. In The 17th Annual IEEE International Conference on Computer Communications (INFOCOM) 1187–1195.
[10] Tian Hao, Ruogu Zhou, Guoliang Xing, and Matt Mutka. 2011. WiSync: Exploiting Wi-Fi infrastructure for clock synchronization in wireless sensor networks. In The 32nd IEEE Real-Time Systems Symposium (RTSS). IEEE, Vienna, Austria, 1379–1392.
[11] Mehrdad Hessari, Ali Najafi, and Shyamnath Gollakota. 2019. NetScanner: Enabling Large-Scale Backscatter Networks. In The 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI).
[12] The European Telecommunications Standards Institute. [n. d. ] Technical characteristics for Low Power Wide Area Networks Chip Spread Spectrum (LPWAN-CCS) operating in the UHF spectrum below 1 GHz. https://www.etsi.org/deliver/etsi_tr/105300_105399/105352/01_01_00/tr_105352v010101p.pdf.
[13] Paul J Marceles, Vijay S Ras, and R Venkatesha Prasad. 2017. DaRe: Data recovery through application layer coding for LoRaWAN. In The 2nd ACM/IEEE International Conference on Internet-of-Things (IoTD).
[14] K. Mikhailov, Juha Petarajärvi, and T. Hämänen. 2016. Analysis of Capacity and Scalability of the LoRa Low Power Wide Area Network Technology. In The 22nd European Wireless Conference 1–6.
[15] T. Mizrahi. 2014. Security Requirements of Time Protocols in Packet Switched Networks. https://tools.ietf.org/html/rfc7384.
[16] Yao Peng, Longfei Shangguan, Yue Hu, Yujie Qian, Xianshang Lin, Xiaojialiang Chen, Dingyang Fang, and Kyle Jamieson. 2018. PiLoRa: a passive long-range data network from ambient LoRa transmissions. In ACM SIGCOMM. ACM, 147–160.
[17] T. Petriäg, M. Goessens, L. Nuaymi, L. Toutain, and A. Pélou. 2016. Measurement, performance, and analysis of LoRa FABIAN, a real-world implementation of LPWAN. In IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC) 1–7.
[18] Cefirmino Gabriel Ramirez, Anton Sergeyev, Assya Dyussenova, and Bob Iannucci. 2019. LongShoT: long-range synchronization of time. In Proceedings of the 16th International Conference on Information Processing in Sensor Networks. ACM, 289–300.
[19] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. 2017. Low power wide area networks: An overview. IEEE Communications Surveys & Tutorials 19, 2 (2017), 855–873.
[20] Pieter Robyns, Eduard Marin, Wim Lamotte, Peter Quax, Dave Singelée, and Bart Hermans. 2017. Selective Jamming of LoRaWAN using Commodity Hardware. In The 27th European Wireless Conference (EW). Mobile and Wireless Communications (WiSec). ACM.
[21] Reinoud Steeman and Torild Van Eck. 1999. Robust automatic b-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.
[22] N. Sornin, M. Luis, T. Eirich, and T. Kramp. 2016. LoRea: A Backscatter architecture that enables Large-Scale Backscatter Networks. In The 3rd Annual LPWAN Conference PREDNET wildlife animal tracking project. In Proceedings of the 1st Annual LPWAN™Specification (V1.0.2).
[23] 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.
[24] Vamsi Talla, Mehrdad Hessari, Bryce Kellogg, Ali Najafi, Jordu R Smith, and Shyamnath Gollakota. 2017. LoRa Backscatter: Enabling the vision of ubiquitous connectivity. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT) 1, 3 (2017), 105.
[25] Eileen Hui Lian Tay. 2018. Cybersecurity of long-range wide-area networks.
[26] Viktor Toldov, J.P. Meijers, Riaan Wolhuter, Nathalie Mutin, and Laurent Clavier. 2016. Performance evaluation of LoRa radio solution for FREDNET wildlife animal tracking project. In 1st Annual LPWAN Conference.
[27] A. Vardhaney, Oliver Harms, Carlos Perez-Penichet, Christian Rohner, Frederik Hermans, and Thiemo Voigt. 2017. LoRea: A Backscatter architecture that achieves a long communication range. In The 15th ACM Conference on Embedded Networked Sensor Systems (SenSys). ACM. 18.
[28] Jiashui Zhu. 2017. Cybersecurity of long-range wide-area networks.

Figure 17: Pictures taken at the two sites.