A Synchronization Front-End for LoRa Self-Jamming Operation on SDR Platforms

Clément Demeslay, Roland Gautier, Philippe Rostaing, Cristina Despina-Stoian

A Synchronization Front-End for LoRa Self-Jamming Operation on SDR Platforms

Clément Demeslay
Lab-STICC, CNRS UMR 6285
University of Brest
6 avenue Le Gorgeu, Brest, France
clement.demeslay@univ-brest.fr

Roland Gautier
Lab-STICC, CNRS UMR 6285
University of Brest
6 avenue Le Gorgeu, Brest, France
roland.gautier@univ-brest.fr

Philippe Rostaing
Lab-STICC, CNRS UMR 6285
University of Brest
6 avenue Le Gorgeu, Brest, France
philippe.rostaing@univ-brest.fr

Cristina Despina-Stoian
Department of Communication and Information Technology
Military Technical Academy "Ferdinand I"
Bucharest, Romania
cristina.despina@mta.ro

***Abstract*—Long Range (LoRa) has become recently a front-runner in the Low-Power Wide-Area Networks (LPWAN) solutions applied to low-energy and low-cost Internet of Things (IoT) transceivers. LoRa security mainly relies on the application layer, where few security is present at the physical layer. Combining security of both layers would increase the overall LoRa security. We proposed in that sense a LoRa self-jamming scheme that enables secure and covert communications by adding jamming symbols when transmitting. The scheme has been validated in simulations. This article continues the work by validating the scheme on real-world Software Defined Radio (SDR) equipment. We develop in that sense a dedicated synchronization front-end, crucial part of the receiver to enable proper demodulation. Each step of the front-end is detailed and results from SDR transmissions are provided to assess demodulation ability. Results show proper demodulation of the transmitted symbols and very good adequacy between expected and actual front-end behavior on SDR. This paves the way for real-world application of the LoRa self-jamming scheme.**

***Index Terms*—LoRa, SDR, USRP, synchronization, self-jamming**

I. Introduction

In the past few years, Long Range (LoRa) has become a front-runner in the Low-Power Wide-Area Networks (LPWAN) solutions applied to low-energy and low-cost Internet of Things (IoT) transceivers and is increasingly implemented to achieve practical solutions in a wide range of fields such as agro-informatics [1] or smart home design [2].

The increasing number of LoRa transceivers gives more and more opportunities for malicious entities to disrupt or eavesdrop LoRa communications. LoRa security mainly relies on the application layer e.g. [3], [4], while few security is present at the physical layer. Reinforcing security of the LoRa physical layer in conjunction with existing application layer security would increase overall LoRa security.

We proposed in that sense in [5] an ally friendly jamming solution inspired by the philosophy proposed in [6], [7], but dedicated to LoRa communications while taking into account the unique LoRa demodulation challenges of our proposed solution. It is based on a self-jamming scheme that tackles a potential eavesdropper by adding deliberately unknown jamming symbols by the latter whose demodulation capability is highly impacted. The legitimate receiver, however, leverages these perfectly known jamming symbols to efficiently demodulate and obtain a processing gain. This scheme exhibits very good performances in simulations.

In this article, we propose to expand the work by assessing the self-jamming scheme on real-world Software Defined Radio (SDR) equipment. To do so, we design a dedicated synchronization front-end, crucial part of the receiver to enable demodulation. Based on comprehensive figures, we present all the steps of the front-end: signal detection, coarse and fine time synchronization, and final synchronization in the LoRa frequency domain. SDR transmissions results are provided and show: 1) very good adequacy between expected and actual synchronization front-end behavior on SDR, and 2) proper payload symbol demodulation on SDR with unaltered self-jamming properties.

The remainder of the paper is as follows. In Section II, we give basics on the LoRa modulation. In Section III, we present the LoRa self-jamming scheme [5]. In Section IV, we give knowledge on LoRa synchronization to ease the understanding of the proposed synchronization front-end. The synchronization front-end is introduced in Section V. Then, we validate the front-end in Section VI by providing results of real-world transmissions performed on SDR platforms.

II. LoRa basics

In literature, LoRa waveforms are of the type of Chirp Spread Spectrum (CSS) signals. These signals rely on sine waves with instantaneous frequency that varies linearly with time over frequency range $f \in [-B/2; B/2]$ with $B \in \{125, 250, 500\}$ kHz and time range $t \in [0; T]$ with T the symbol period. This basic signal is called an *upchirp*

979-8-3503-5595-6/24/$31.00 ©2024 IEEE

or *downchirp* when frequency respectively increases or decreases over time. A LoRa symbol consists of SF bits with $SF \in \{7, 8, \dots, 12\}$, leading to an M-ary modulation with $M = 2^{SF} \in \{128, 256, \dots, 4096\}$. In the discrete-time signal model, the Nyquist sampling rate is usually used to minimize computation resources: $F_s = B = 1/T_s$. A mathematical expression of LoRa waveform sampled at $t = kT_s$ has been derived in [8]:

$$x_a[k] = e^{2j\pi k\left[\frac{a}{M} - \frac{1}{2} + \frac{k}{2M}\right]}, \quad k \in \{0, 1, \dots, M-1\}. \tag{1}$$

We may see that an upchirp is actually a LoRa waveform with symbol index $a = 0$, written $x_0[k]$. Its conjugate $x_0^*[k]$ is then a downchirp. Reference [9] proposed a simple and efficient solution to demodulate LoRa signals. In Additive White Gaussian Noise (AWGN) channel, the demodulation process is based on the Maximum Likelihood (ML) detection scheme. The received signal is:

$$r[k] = x_a[k] + w[k] \tag{2}$$

with $w[k]$ a complex AWGN with zero-mean and variance $\sigma^2 = E[|w[k]|^2]$, with $E[.]$ denoting the expectation operator.

The ML detector aims to select index $\widehat{a}$ that maximizes the scalar product $\langle r, x_n \rangle$ for $n \in \{0, 1, \dots, M-1\}$ defined as:

$$\langle r, x_n \rangle = \sum_{k=0}^{M-1} r[k] x_n^*[k] \tag{3a}$$

$$= \sum_{k=0}^{M-1} \underbrace{r[k] x_0^*[k]}_{\tilde{r}[k]} e^{-j2\pi \frac{n}{M} k} \tag{3b}$$

$$= \mathrm{DFT}\{\tilde{r}[k]\}[n] = \tilde{R}[n] \tag{3c}$$

with $\mathrm{DFT}\{x[k]\}[n]$ the Discrete Fourier Transform (DFT) of $x[k]$. The DFT operator can be implemented very efficiently with a Fast Fourier Transform (FFT) algorithm. This considerably reduces the computation complexity of the demodulator. The demodulation stage proceeds with two simple operations:

- multiply the received signal by the downchirp $x_0^*[k]$, also called dechirping process,
- compute $\tilde{R}[n]$, the DFT of $\tilde{r}[k]$ and select the discrete frequency index $\widehat{a}$ that maximizes $\tilde{R}[n]$.

This way, the dechirp process merges all the signal energy onto a unique frequency bin a and can be easily retrieved by taking the magnitude of $\tilde{R}[n]$. The symbol detection is then:

$$\widehat{a} = \arg\max_n \quad \left|\tilde{R}[n]\right|. \tag{4}$$

The Signal-to-Noise Ratio (SNR) is defined as:

$$SNR = \frac{P_s}{\sigma^2} = \frac{1}{\sigma^2} \tag{5}$$

with $P_s = E[|x_a[k]|^2]$.

III. LoRa self-jamming scheme [5]

A. Modulation

The self-jamming scheme is presented in Fig. 1. It consists of transmitting $U - 1$ jamming symbols that are superimposed to the symbol of interest and perfectly known by the legitimate receiver but not by a potential eavesdropper. There are in total U transmitted symbols. The legitimate receiver has the ability to recover the information but the eavesdropper will not. In fact, the latter will see U symbols at random DFT locations and will have strong difficulties to correctly demodulate without the knowledge of the jamming symbols. Furthermore, each symbol has a fraction of the total available power i.e. $P_{jam} = P_s/U = 1/U$, making it difficult to make a proper detection as the LoRa DFT magnitudes are progressively reduced with U, leading to a higher AWGN sensitivity. It can be seen as a spread spectrum approach but only performed in the LoRa frequency domain i.e. the LoRa DFT. The signal bandwidth remains unchanged with B bandwidth.

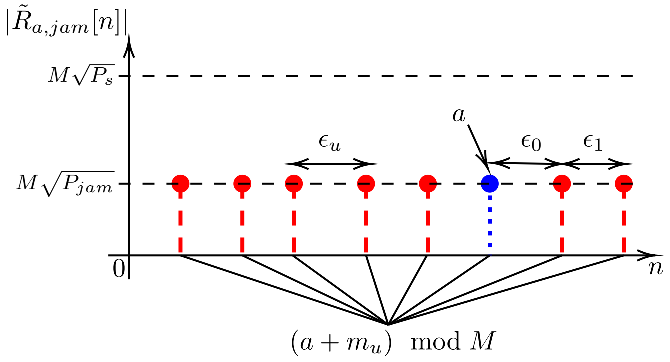

Fig. 1. Illustration of the LoRa self-jamming scheme in the LoRa DFT frequency domain.

The jamming symbols are defined as:

$$a_{u,jam} = (a + m_u) \mod M, \quad m_u \in \{0, 1, \dots, M-1\} \tag{6}$$

with $a_{u,jam}$ in (6) the u-th jamming symbol having the relative delay m_u from the desired symbol symbol a. The first jamming symbol $a_{0,jam}$ corresponds to the symbol of interest to be retrieved by the receiver. Then, $m_0 = 0$ for $u = 0$. The transmitted waveform is then:

$$x_{a,jam}[k] = x_a[k] \underbrace{\sqrt{P_{jam}} \sum_{u=0}^{U-1} e^{2j\pi k \frac{m_u}{M}}}_{S_{jam}[k]} \tag{7}$$

with $S_{jam}[k]$ in (7) the self-jamming component. We denote as $\mathbf{m}$ the vector of delays m_u and ϵ_u the value difference between the $(u+1)$-th and u-th jamming symbols as:

$$\begin{aligned} \epsilon_u &= (m_{u+1} - m_u) \mod M \\ &= (a_{u+1,jam} - a_{u,jam}) \mod M. \end{aligned} \tag{8}$$

From (7), the self-jamming signal is in time domain the perfect superposition of the $U-1$ jamming symbols over the symbol of interest. The symbol duration remains unchanged with T duration.

In the LoRa standard, the frame has a preamble to perform the necessary processing before demodulating the payload i.e. frame detection and synchronization. The preamble consists of N_{up} upchirps followed by N_{ID} network identification symbols and $N_{down} = 2.25$ downchirps [10]. There are N_d payload symbols in the frame. In [5], the frame format is modified by without loss of generality 1) ignoring the networking identification symbols as those are not used and 2) setting the constraint $N_{down} = N_{up}$ for better synchronization performances. For the default value $N_{up} = 8$, this leads to a preamble duration of 16 symbols now instead of 10.25 (56% longer). The spectral efficiency of the self-jamming scheme will be lower than that of the legacy LoRa scheme.

B. Demodulation

We denote as $r_{jam}[k] = x_{a,jam}[k] + w[k]$ the received LoRa self-jamming signal. The legitimate receiver can retrieve the desired symbol with the following simple cross-correlation approach [5]:

$$\widehat{a} = \arg\max_{v} \quad F_{cc}[v] \tag{9}$$

with:

$$F_{cc}[v] = \left| \mathrm{DFT}\left\{ \sum_{u=0}^{U-1} r_{jam}[k] x^*_{m_u}[k] \right\}[v] \right|. \tag{10}$$

The received signal in (10) is projected onto each jamming symbol $x^*_{m_u}[k]$ instead of the unique reference downchirp $x^*_0[k]$, as in (3b). This receiver is named the cc receiver (cross-correlation receiver).

Fig. 2 shows an example of the demodulation of the symbol $a = 0$ with the cc receiver for $U = 8$, $\mathbf{m} = [0\ 14\ 32\ 39\ 52\ 67\ 83\ 87]$, $SF = 7$, with AWGN noise ($SNR = 0$ dB). The left side of the figure shows the LoRa DFT and the right side the cc output. We clearly see the symbol of interest at $v = a = 0$ in the cc output.

IV. Basics on LoRa synchronization

The major desynchronizations that LoRa signals encounter in practice are the Carrier Frequency Offset (CFO) i.e. residual carrier signal in baseband and the Sampling Time Offset (STO) i.e. demodulation window reference shift between the transmitted and the receiver. These two desynchronizations have a strong impact on LoRa demodulation and thus need to be compensated before demodulation. In what follows, we give basics on LoRa CFO and STO synchronization of literature (concepts and state-of-the-art algorithms that will be exploited in this article), plus specific synchronization elements dedicated to the LoRa self-jamming scheme that will be used many times throughout the article.

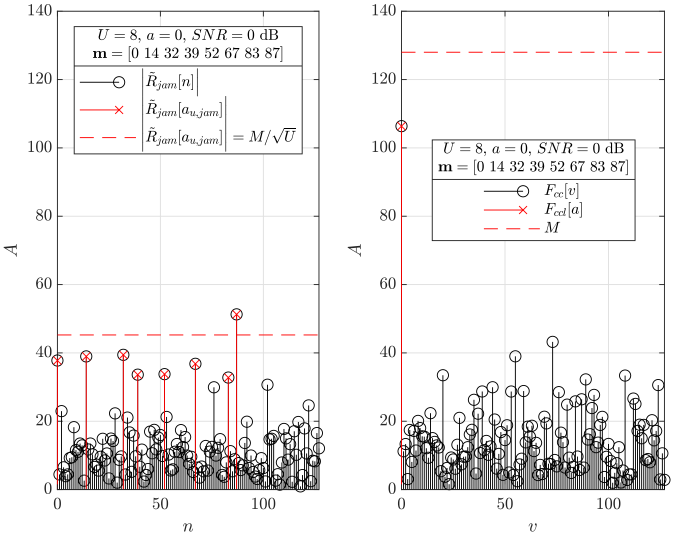

Fig. 2. Illustration of the LoRa DFT (left) and cc receiver output (right) with AWGN noise ($SNR = 0$ dB), $U = 8$ and $SF = 7$.

A. CFO and STO impact on LoRa demodulation

1) CFO impact on LoRa demodulation: The CFO is translated in LoRa terms as:

$$CFO = \frac{\Delta_f}{B/M} = (CFO_{int} + CFO_{frac}) \tag{11}$$

with Δ_f in (11) the CFO expressed in Hz, CFO_{int} and CFO_{frac} the integer and fractional part of LoRa DFT bins shift. For example, $\Delta_f = 1.2$ kHz leads to $CFO_{int} = 1$ and $CFO_{frac} \approx 0.228$ for $SF = 7$ and $B = 125$ kHz. CFO_{int} shifts the main peak by CFO_{int} positions in the LoRa DFT i.e. $\widehat{a} = (a + CFO_{int}) \mod M$ and CFO_{frac} spreads the energy over neighbor bins. The closer to 0.5 CFO_{frac}, the higher the energy spread, in both left and right neighbor bins. The same impact is experienced when using the cc receiver.

2) STO impact on LoRa demodulation: The STO is a demodulation window time reference shift, noted τ:

$$STO = \frac{\tau}{T_s} = (STO_{int} + STO_{frac}). \tag{12}$$

For example, $\tau = 99.2\ \mu$s leads to $STO_{int} = 12$ and $STO_{frac} = 0.4$. It has an analogue impact on the DFT and the cc receiver as the CFO and is different at each transmission since the transmitter and the receiver are not synchronized with each other. The STO is modeled as a uniform random distribution between 0 and $M - 1$.

B. CFO_{frac} estimator form literature

CFO_{frac} can be estimated independently from the other desynchronizations i.e. independently from CFO_{int}, STO_{int} and STO_{frac}, thanks to the algorithm in [11]. It is based on the well-known Three Spectral Line (TSL) approach which evaluates the energy asymmetry between adjacent bins generated by the CFO fractional part in the frequency domain. It has very good AWGN resiliency and is barely impacted by the presence of self-jamming symbols. This estimator will then be considered for the proposed synchronization front-end.

C. The ccs demodulator

To synchronize itself, the cc receiver needs to demodulate the signal with the LoRa self-jamming receiver in (10). It has correct operation provided that the signal is synchronized in time i.e. $STO = 0$. At synchronization stage, the received signal is unfortunately desynchronized with unknown STO. The cc receiver will not work properly. We propose then a modified version of the cc receiver, called ccs (cross-correlation synchronization) for working during synchronization stage:

$$\widehat{a} = \arg\max_{v} \quad F_{ccs}[v] \tag{13}$$

with:

$$F_{ccs}[v] = \sum_{n=0}^{M-1} \left| \tilde{R}_{jam}[n]\tilde{X}_{0,jam}[(n-v) \mod M] \right| \tag{14}$$

and:

$$\tilde{R}_{jam}[n] = \text{DFT}\left\{ r_{jam}[k]x_0^*[k] \right\}[n] \tag{15}$$

$$\tilde{X}_{0,jam}[n] = \text{DFT}\left\{ x_{0,jam}^*[k] \right\}[n]. \tag{16}$$

D. The STO_{frac} candidate estimator

There is a robust STO_{frac} estimator in literature [10]. It is also based on the TSL approach as used in the CFO_{frac} estimator and shows very good performances. However, the latter does not work properly when applied on LoRa self-jammed waveforms.

We decide instead to use a solution to estimate STO_{frac} based on a candidate approach. We define the set of STO_{frac} candidates as $\mathcal{S}_{STO_{frac}}$ with C, the number of candidates. We select then the candidate that, after STO_{frac} correction, minimizes the energy spread in the normalized ccs output (having $\max_{v} \quad F_{ccs}[v] = 1$). The energy spread is evaluated by computing the magnitude distance between the bin of interest and its highest left or right neighbor bin:

$$\delta(STO_{frac}^{cand}) = |1 - A_{max}| \tag{17}$$

with:

$$A_{max} = \max \quad \{A^-, A^+\} \tag{18}$$

$$A^- = F_{ccs}^{up}[(\widehat{a}_{up} - 1) \mod M]/F_{ccs}^{up}[\widehat{a}_{up}] \tag{19}$$

$$A^+ = F_{ccs}^{up}[(\widehat{a}_{up} + 1) \mod M]/F_{ccs}^{up}[\widehat{a}_{up}] \tag{20}$$

$$\widehat{a}_{up} = \arg\max_{v} \quad F_{ccs}^{up}[v] \tag{21}$$

$$F_{ccs}^{up}[v] = \sum_{n=0}^{M-1} \left| \tilde{P}_{jam}[n]\tilde{X}_{0,jam}[(n-v) \mod M] \right| \tag{22}$$

and:

$$\tilde{P}_{jam}[n] = \frac{1}{N_{up}} \sum_{i=0}^{N_{up}-1} \tilde{R}_{jam}^{(i)}[n] \tag{23}$$

the averaged upchirp symbols of the preamble. $\tilde{R}_{jam}^{(i)}[n]$ in (23) denotes the DFT of the i-th preamble upchirp symbol ($a = 0$). The more the distance, the less the energy spread.

STO_{frac} is finally estimated as:

$$\widehat{STO}_{frac} = \arg\max_{STO_{frac}^{cand} \in \mathcal{S}_{STO_{frac}}} \quad \delta(STO_{frac}^{cand}) \tag{24}$$

Fig. 3 illustrates an example of the estimator execution for two STO_{frac} values ($STO_{frac} = 0.3$ and $STO_{frac} = 0.8$). For $STO_{frac} = 0.3$, the candidate $STO_{frac}^{cand} = 0.3$ maximizes the magnitude distance since there is no more fractional STO ($STO_{frac} - STO_{frac}^{cand} = 0$). For $0.3 \leq STO_{frac}^{cand} \leq 0.8$, the distance decreases since at these candidate values, the fractional STO is actually increasing. The minimum distance is reached when $STO_{frac}^{cand} = 0.8$ as it corresponds to have maximum fractional STO ($STO_{frac} - STO_{frac}^{cand} = -0.5$). The behavior is opposite for $STO_{frac} = 0.8$ since there is 0.5 difference between $STO_{frac} = 0.3$ and $STO_{frac} = 0.8$.

The set $\mathcal{S} = \{0, 0.1, \ldots, 0.9\}$ of STO_{frac} candidates ($C = 10$) are chosen for the synchronization front-end.

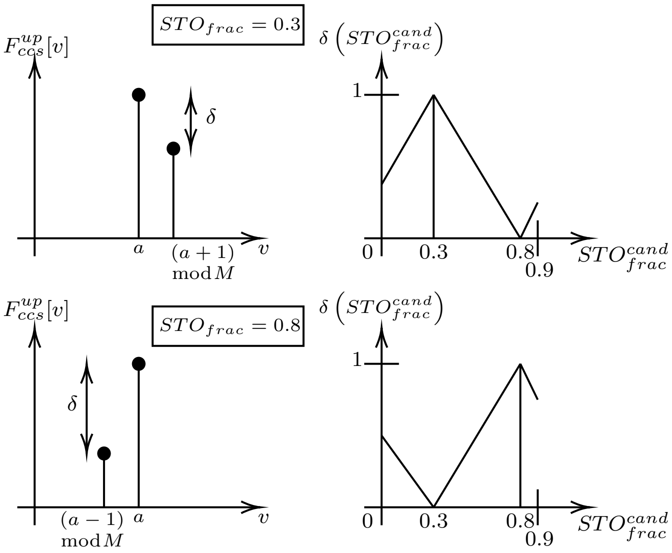

Fig. 3. LoRa self-jammed signal detection principle.

V. LoRa self-jamming synchronization front-end for SDR operation

A. Front-end overview

The LoRa self-jamming synchronization front-end is summarized in Fig. 4. There are 4 steps in total, each one is denoted with a circled number, in the chronological order. The principle is as follows. The first step is to decide whether the received signal is a pure AWGN one (no activity on the channel) or the LoRa self-jammed of interest. Once the signal of interest is detected, the receiver performs a coarse time synchronization by estimating the beginning of the frame at the LoRa symbol level (Step 2). It continues with an initial fine time synchronization at sample level in Step 3. Then, it performs the final time and frequency synchronization (CFO_{int} and STO_{int}) in Step 4. In the next sub-sections, we detail each step.

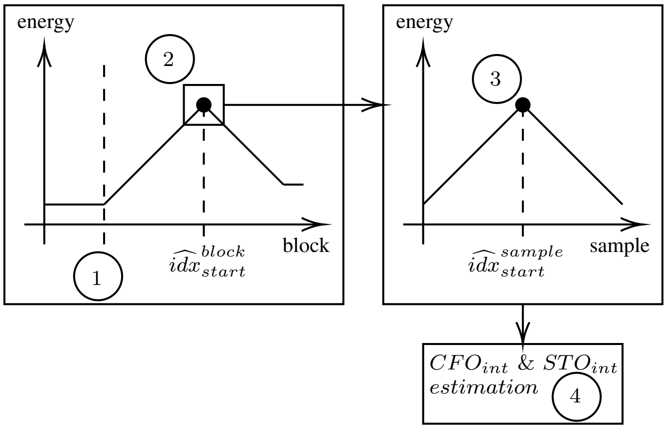
energy
2
block
$\widehat{idx}^{block}_{start}$
1
energy
3
sample
$\widehat{idx}^{sample}_{start}$
CFO_{int} & STO_{int} estimation
4

Fig. 4. Overview of the LoRa self-jamming synchronization front-end.

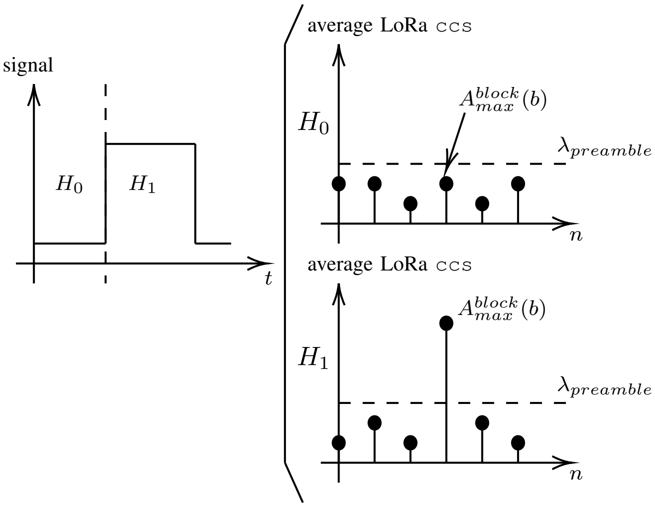

Fig. 5. LoRa self-jammed signal detection principle.

B. Step 1 : LoRa self-jammed signal detection

In Step 1, the receiver must decide whether the received signal is a pure AWGN one (noted as hypothesis H_0) or the LoRa self-jammed of interest (noted as hypothesis H_1), as illustrated in Fig. 5. To do so, the receiver extracts energy in the ccs output. In H_0, only AWGN is present, bins amplitudes are relatively low. In H_1, ccs concentrates the energy in a unique bin. Then, following a decision threshold, noted $\lambda_{preamble}$, the receiver can detect the presence of the LoRa self-jammed signal. The following steps are required to make the decision efficient:

- 1.a: To determine a well suited threshold value, the received signal is normalized to the estimated AWGN variance i.e. $r[k] = r[k]/\widehat{\sigma}^2$. The receiver can estimate the variance during silence periods and update it only from time to time since the AWGN variance only slightly drifts over time [12].
- 1.b: The ccs output is produced from N_{up} averaged blocks of M samples (interpreted as preamble upchirp symbols). This reduces AWGN variance and improves signal detection in low SNR conditions. We use b notation to index averaged blocks.
- 1.c: The receiver extracts the maximum magnitude from the ccs output, noted $A^{block}_{max}(b)$.
- 1.d: If $A^{block}_{max}(b) > \lambda_{preamble}$, then the LoRa self-jammed signal is detected. It repeats steps 1.a-1.d with the next averaged block (shifting by M samples), otherwise. For simplicity purpose, the threshold is found empirically in the performed SDR transmissions in Section VI.

C. Step 2 : Coarse time synchronization at symbol level

Once the LoRa self-jammed signal is detected, the receiver continues to produce averaged blocks, following sub-steps 1.b and 1.c of Step 1. It performs a coarse time synchronization by estimating on which averaged block the preamble starts (presence of preamble upchirp symbols). The index is noted idx^{block}_{start}. As the receiver shifts by M samples between averaged blocks, it can actually be seen as a moving average process, as shown in Fig. 6 ($N_{up} = N_{down} = 2$ for figure clarity). The red, blue and black blocks denote the upchirp, downchirp and payload section of the frame, respectively. The energy is maximized when the current averaged block is derived from the N_{up} preamble upchirp symbols, at $b = idx^{block}_{start}$. As the fractional part of the CFO and STO spreads the energy over neighbor bins, those must mitigated to maximize energy concentration and thus enabling proper preamble start detection. It is performed by using the CFO_{frac} estimator of literature [11] and the STO_{frac} candidate estimator presented in Section IV-D.

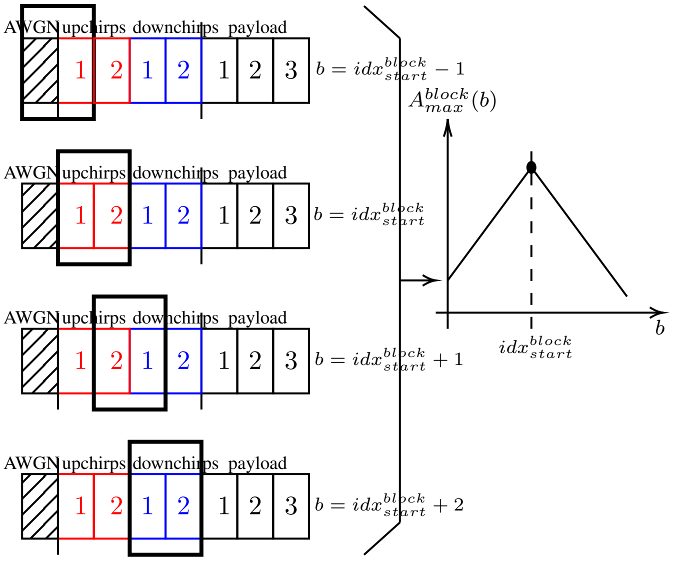
AWGN upchirps downchirps payload
1 2 1 2 1 2 3
$b = idx^{block}_{start} - 1$
AWGN upchirps downchirps payload
1 2 1 2 1 2 3
$b = idx^{block}_{start}$
AWGN upchirps downchirps payload
1 2 1 2 1 2 3
$b = idx^{block}_{start} + 1$
AWGN upchirps downchirps payload
1 2 1 2 1 2 3
$b = idx^{block}_{start} + 2$
$A^{block}_{max}(b)$
idx^{block}_{start}
b

Fig. 6. Moving average process for coarse time synchronization at symbol level.

D. Step 3 : Initial fine time synchronization at sample level

The receiver performs next an initial fine time synchronization at sample level. It performs the same operations as in

Step 2, except that *i)* CFO_{frac} and STO_{frac} need not be re-estimated as the new estimates would be the same as the ones derived at LoRa symbol level in Step 2, and *ii)* the receiver now shifts by one sample between averaged blocks.

The blocks are indexed with β notation, and $A_{max}^{sample}(\beta)$ and idx_{start}^{sample} notations are used to differentiate from LoRa symbol level (M samples) in Step 2. Again, the energy is maximized when the current averaged block is derived from preamble upchirp symbols exactly aligned i.e. $\beta = idx_{start}^{sample}$. idx_{start}^{sample} is searched around idx_{start}^{block}, from $\beta = \widehat{idx}_{start}^{block} - M$ to $\beta = \widehat{idx}_{start}^{block} + M$ (see the box next to Step 2 label in Fig. 4). The search space contains then $2M+1$ samples.

E. Step 4 : Final time and frequency synchronization

The last step is to perform the final time and frequency synchronization i.e. estimating CFO_{int} and STO_{int} (CFO_{frac} and STO_{frac} already estimated in Step 2). It is estimated by leveraging both upchirp and downchirp preamble symbols of the frame [10] but using the ccs output [5] instead of the regular LoRa DFT. Since the idx_{start}^{sample} estimation is not perfect (likely $\widehat{idx}_{start}^{sample} \neq idx_{start}^{sample}$), the cc receiver can not be used as the latter is very sensitive to any time delay. This leads to:

$$\widehat{STO}_{int} = (\widehat{a}_{up} - \widehat{CFO}_{int}) \mod M \tag{25}$$

$$\widehat{CFO}_{int} = \frac{(\widehat{a}_{up} + \widehat{a}_{down}) \mod M}{2} \tag{26}$$

with:

$$\widehat{a}_{up} = \arg\max_{v} \quad F_{ccs}^{up}[v] \tag{27}$$

$$\widehat{a}_{down} = \arg\max_{v} \quad F_{ccs}^{down}[v] \tag{28}$$

and $F_{ccs}^{up}[v]$ and $F_{ccs}^{down}[v]$ in (27) and (28), the preamble upchirp symbols ccs outputs averaged together and the downchirp preamble symbols ccs outputs averaged together, respectively.

The STO_{int} estimation will actually depend on the estimated idx_{start}^{sample} value. For the sake of clear explanation, we suppose no integer CFO i.e. $\widehat{CFO}_{int} = CFO_{int} = 0$ (in practice, CFO_{int} is corrected before STO_{int} estimation). Let us denote the difference between the estimated and actual idx_{start}^{sample} by:

$$c = \widehat{idx}_{start}^{sample} - idx_{start}^{sample}. \tag{29}$$

Depending on c, the demodulation behavior will not be the same. Fig. 7 illustrates this for the two cases $c < 0$ and $c > 0$. When $c < 0$, it means that the estimated start sample will be before the actual one. The receiver will then be ahead of time (sign $-$ in the figure). This implies that the residual sample time delay will shift bins to the left in the ccs output. That is, $c = -2$ will produce the main bin at $v = M - |c| = M - 2$. The behavior is opposite when $c > 0$ i.e. the receiver is lagging behind (sign $+$) and the residual sample time delay will shift bins to the right in the ccs output ($c = 3$ leads to the main bin at $v = c = 3$).

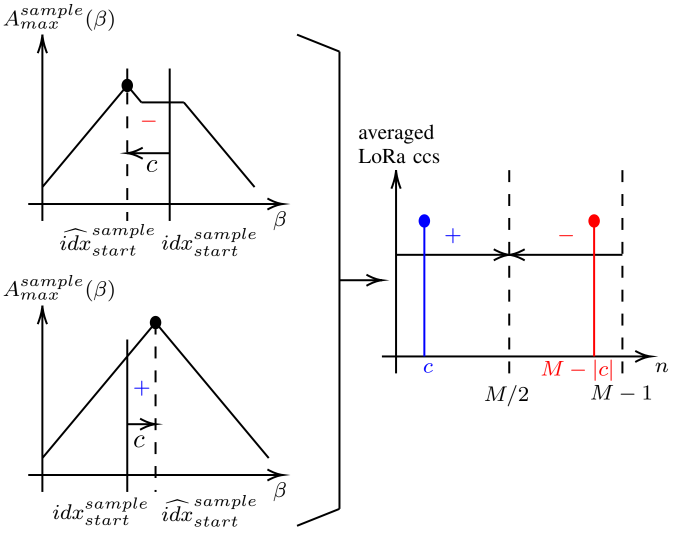

Fig. 7. Illustration of the impact of imperfect initial fine time synchronization on STO_{int} estimation.

This can be seen as shifting decision boundaries in the ccs output:

$$\begin{cases} c \text{ samples to the left} & \text{if } 0 \leq \widehat{STO}_{int} < M/2 \\ |c| \text{ samples to the right} & \text{if } M/2 \leq \widehat{STO}_{int} \leq M-1 \end{cases}$$

If $c = 0$, the initial fine time synchronization perfectly estimated the beginning of the frame and thus $\widehat{STO}_{int} = 0$. To be able to correctly determine the correction direction ($-$ or $+$), the constraint $|c| \leq M/2$ must be satisfied i.e. consistent initial fine time synchronization. Simulations show that this constraint is satisfied more than 99% of the time for $SNR \geq -4$ dB and up to $U = 30$ jamming symbols.

Once CFO_{frac}, CFO_{int}, STO_{frac} and STO_{int} are estimated, the receiver can time and frequency synchronize the frame and proceed to the payload demodulation with the cc receiver.

VI. SDR EXPERIMENTS RESULTS

A. Test-bed

In this section, we present examples of LoRa self-jammed frames synchronization and demodulation on SDR platforms. The test bed is shown in Fig. 8. We used SDR devices from Ettus Research, the B210 Universal Software Radio Peripheral (USRP) platform. Two USRP are used: one for transmission (left in the figure), one for reception (right in the figure), and are placed at a distance of approximately 50 cm. Two VERT900 antennas are used, with dual band 824-960/1710-1990 MHz operation [13]. Thus, these antennas are suited for LoRa operations (e.g. 868.1 MHz LoRa carrier frequency in Europe).

The SDR parameters used for LoRa self-jammed transmissions are reported in Table I.

We use $SF = 7$, one frame is transmitted each time having $N_{up} = 8$ preamble upchirp symbols, $N_{down} = N_{up}$ downchirp

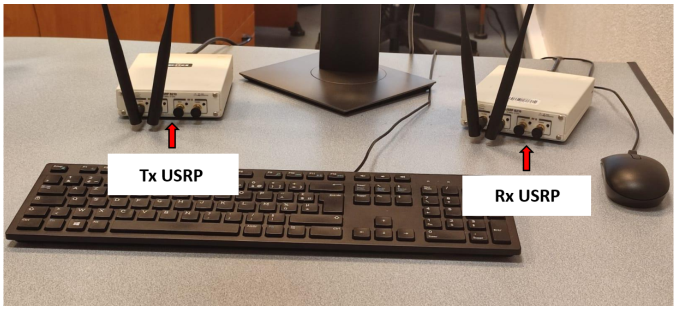

Fig. 8. SDR test-bed with two Ettus Research USRP B210.

SDR parameter	Value
Carrier frequency (F_c)	868.1 MHz
Transmission gain (G_{T_x})	40 dB
Receive gain (G_{R_x})	0 dB
Estimated SNR	$\widehat{SNR} > 15$ dB (for proper synchronization and demodulation visualization)

TABLE I
SDR PARAMETERS USED FOR LORA SELF-JAMMED TRANSMISSIONS.

preamble symbols, and $N_d = 500$ payload symbols. $SF = 7$ is the most common value for LoRa transmissions since it gives the lowest computation burden for still a very good cell coverage, at least 10 km [14]. Increasing SF will improve demodulation performance ($\approx$ 2.5 SNR dB gain for each SF increment), without changing the self-jamming properties. $N_{up} = 8$ is the default number of upchirp preamble symbols [15]. There are 500 payload symbols, which is quite a big value for LoRa (usually lower). The transmission time is then longer, enabling us to assess demodulation stability over time. The frame is synchronized and demodulated with no symbol demodulation error.

Two transmissions are performed: $U = 5$ or $U = 30$ jamming symbols. The transmission with $U = 30$ will be compared to the one $U = 5$, to evaluate the synchronization front-end and demodulation behavior for higher U values, where one might expect potential edge effects on SDR platforms.

B. Results

Table II reports execution time (in ms) of frame demodulation of the two aforementioned transmissions ($U = 5$ and $U = 30$). Execution time is decomposed as execution time of the four steps of the synchronization front-end (S1 to S4 columns in the table), execution time of synchronization (S1+S2+S3+S4 columns), execution time of payload demodulation (payl. column), and execution time of frame demodulation (sync. + payl. column). The code is executed in MATLAB environment without any compilation, with an i7 13700K CPU and 32GB @3600 MHz RAM. Step 1 (signal detection) is performed over the 16 blocks preceding the beginning of the preamble. We also added execution time of a LoRa legacy frame ($U = 1$) applying the same synchronization front-end. The ccs receiver is replaced with the legacy non-coherent receiver in (4) and the cc receiver also reduces to the legacy non-coherent receiver (4) in this case since $U = 1$.

	S1	S2	S3	S4	sync.	payl.	sync. + payl.
$U = 5$	16	13	10	3	42	17	59
$U = 30$	16	13	10	3	42	80	122
LoRa legacy ($U = 1$)	15	12	9	3	39	7	46

TABLE II
EXECUTION TIME (MS) OF SELF-JAMMED ($U = 5$ AND $U = 30$) AND LORA LEGACY ($U = 1$) FRAMES DEMODULATION.

From Table II, we see that synchronization front-end execution time is the same between $U = 5$ and $U = 30$ with about 42ms. In fact, all processing is done without prior knowledge on U. Then, it does not depend on U. The execution of the synchronization front-end is slightly faster with the LoRa legacy frame. Indeed, the cross-correlation process in (14) is implemented with highly optimized FFT algorithms, leading to close computation complexity to the legacy receiver (4). Note that payload execution time for $U = 5$ is much lower compared to $U = 30$ (17 vs. 80). This comes from the summation term over U in (10). The legacy receiver having $U = 1$, the latter gives the fastest execution with 7ms. Since having higher U is more desirable, security is enabled at a cost of a higher delay for the receiver. The self-jamming scheme should then be dedicated to applications where delay and energy consumption are less critical. Demodulating 500 payload symbols with $U = 5$ gives about the same execution time as demodulating 100 payload symbols with $U = 30$ (17ms vs. 16ms (80/5=16, linear behavior)). Then, to keep equivalent time execution, frames may be shorter when using higher U values.

Fig. 9 shows the execution of the synchronization front-end of the $U = 5$ transmission in subplots a) to c) and the cc output before and after synchronization in subplots d) to f). For signal detection, we set the threshold value at $\lambda_{preamble} = 6.8$ for $U = 5$, a value enabling proper signal detection on SDR. The computation of the theoretical value for given U and false alarm probability could be an investigation path for future research. We have $B = 250$ kHz bandwidth.

From Fig. 9, we can conclude:

- Subplot a): We clearly see the progressive energy maximization thanks to the moving average process (see Fig. 6). The signal is detected at the 65-th block (Step 1) and the preamble start at symbol level is detected at the 73-th block (coarse time synchronization in Step 2). Note the amplitude exceeding 1000 due to AWGN normalization in sub-step 1.a of Step 1 ($\widehat{\sigma}^2 = 1.23 \times 10^{-8}$), and higher amplitudes in downchirp and payload section ($b > 80$) than in pure AWGN section ($b < 65$).
- Subplot b): We see the expected behavior of Step 3. The energy is maximized at $\beta = 77$ i.e. $\widehat{idx}_{start}^{sample} = 77$. This leads to $\widehat{STO}_{int} = 0$ thus $c = 0$ (perfect initial fine time synchronization in Step 3).
- Subplots c) and d): We show the LoRa DFT of the 2-th payload symbol ($a = 63$) before and after synchronization. The estimated synchronization parameters are

shown in subplot d). Note the low estimated CFO with $\widehat{CFO}_{int} = 0$ and $\widehat{CFO}_{frac} \approx 0.08$ (An actual $\Delta_f \approx$ 160 Hz). We clearly see that without synchronization, the demodulation is impossible as bins of interest are flooded. After synchronization, the $U = 5$ jamming symbols are clearly visible.

- Subplots e) and f): We show the cc output of the 2-th payload symbol before and after synchronization. Clearly, the synchronization enables proper demodulation as the symbol energy is concentrated in the bin of interest at $v = a$.

Fig. 10 shows the same results as in Fig. 9, but for $U =$ 30 jamming symbols and $B = 125$ kHz bandwidth. We set the threshold value at $\lambda_{preamble} = 14$, a value also enabling proper signal detection on SDR for $U = 30$.

From Fig. 10, we can conclude:

- Subplot a): We see normal behavior, with overall lower amplitudes than $U = 5$ (≈ 600 vs. ≈ 1100). This comes from increasing U.
- Subplot b): We see normal behavior. $\widehat{STO}_{int} = 125$ leading to $c = -3$ this time ($\widehat{idx}_{start}^{sample} = 87$ vs. $idx_{start}^{sample} = 90$). Note that as U grows, the energy tends to flatten around idx_{start}^{sample}. The increased ambiguity is mitigated thanks to the final STO_{int} estimation.
- Subplots c) and d): As U is much higher, even with synchronization, it is much more difficult to distinguish jamming symbols (due to the fact that the transmission is performed at constant total transmit energy for both $U = 5$ and $U = 30$). This brings to light the efficiency of the self-jamming scheme.
- Subplots e) and f): The total energy after synchronization is lower than that of $U = 5$ (≈ 90 for $U = 30$ vs. ≈ 111 for $U = 5$). This is inherent to the cc receiver structure as the main amplitude progressively decreases with U increasing. The cc receiver proves here its relevance and robustness as even for high U value, the energy concentration ability is unaltered on SDR platforms.

Fig. 11 shows the evolution over the 500 payload symbols of the main and direct neighbor cc output bin amplitudes, for the aforementioned SDR transmissions performed at $U = 5$ and $U = 30$. The amplitudes are noted V^- (left direct neighbor), V (bin of interest), and V^+ (right direct neighbor). We see from the figure the consistency of the bin of interest amplitude, with low spread around average value (≈ 112 for $U = 5$ and ≈ 91 for $U = 30$). The energy residual in direct neighbor bins is also low (< 22 for both $U = 5$ and $U = 30$), showing good CFO_{frac} and STO_{frac} synchronization since these two desynchronizations spread the energy over left and right neighbor bins.

VII. Conclusion

In this article, we presented a synchronization front-end for enabling proper demodulation of LoRa self-jammed waveforms designed to enhance the security of LoRa communications by adding jamming symbols when transmitting. The front-end is step-by-step detailed with comprehensive explanations and illustrations. To validate the front-end, real-world LoRa self-jammed transmissions are performed on Ettus B210 SDR platforms. Results show proper demodulation of the transmitted symbols and very good adequacy between expected and actual front-end behavior on SDR. This demonstrates the viability of the LoRa self-jamming scheme on real-world equipment.

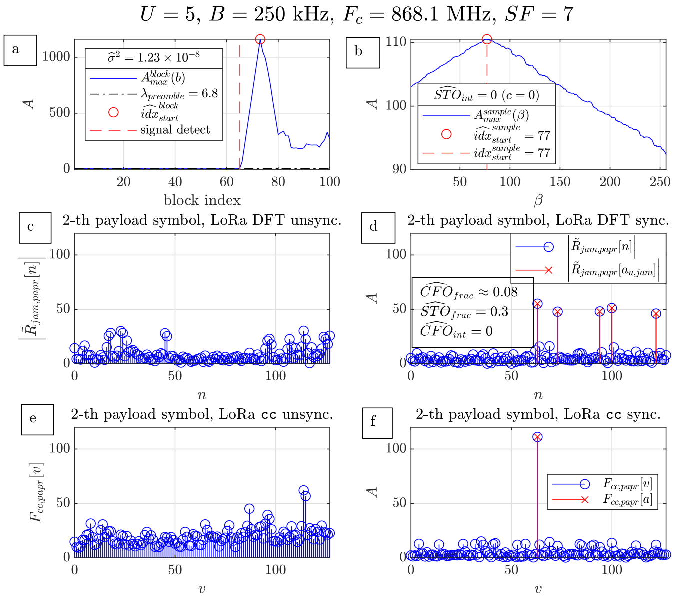

Fig. 9. LoRa self-jammed frame synchronization and demodulation on SDR platform ($U = 5$).

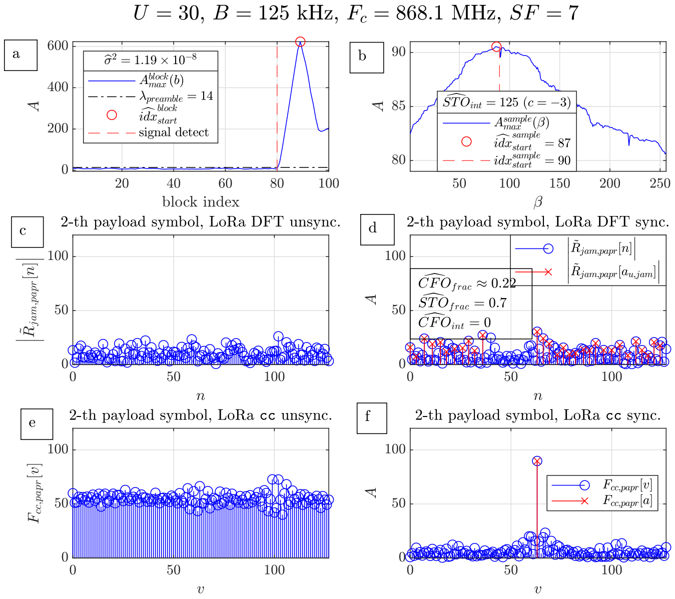

Fig. 10. LoRa self-jammed frame synchronization and demodulation on SDR platform ($U = 30$).

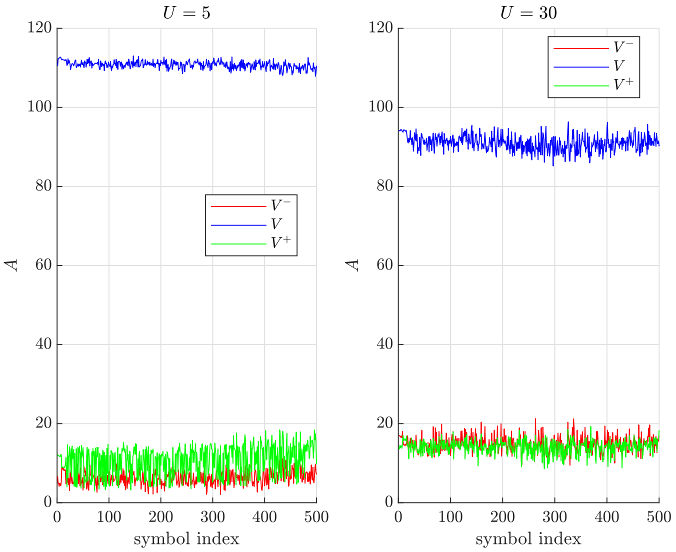

Fig. 11. Bin of interest and direct neighbor bins amplitude evolution (cc output) over payload symbols, for the two transmissions performed at $U = 5$ and $U = 30$.

Acknowledgments

This work is jointly supported by the Brest Institute of Computer Science and Mathematics (IBNM) CyberIoT Chair of Excellence of the University of Brest, the "Pôle d'Excellence Cyber" and the DEI service of the University of Brest for the international mobility grant. We also would like to thank the Military Technical Academy of Bucharest for their valuable contribution to this study. This work is also supported in part by the European Regional Development Fund (ERDF), the French Ministry of Higher Education and Research, the Brittany Region, Brest Métropole, the departmental council 29 through the CPER projects "SpaceTechDroneTech" and "Industrie du Futur".

References

[1] A. Gehani, S. Harsha, R. Raghav, M. Sarkar, and C. Paolini, "Application of 915 MHz band LoRa for agro-informatics," in *2021 Wireless Telecommunications Symposium (WTS)*, pp. 1–4, 2021.

[2] S. Opipah, H. Qodim, D. Miharja, Sarbini, E. A. Z. Hamidi, and T. Juhana, "Prototype design of smart home system base on LoRa," in *2020 6th International Conference on Wireless and Telematics (ICWT)*, pp. 1–5, 2020.

[3] A.-U.-H. Ahmar, E. Aras, D. T. Nguyen, S. Michiels, W. Joosen, and D. Hughes, "Cram: Robust medium access control for lpwan using cryptographic frequency hopping," in *2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 95–102, 2020.

[4] K.-L. Tsai, F.-Y. Leu, I. You, S.-W. Chang, S.-J. Hu, and H. Park, "Low-power aes data encryption architecture for a lorawan," *IEEE Access*, vol. 7, pp. 146348–146357, 2019.

[5] C. Demeslay, R. Gautier, P. Rostaing, G. Burel, and A. Fiche, "A Novel Scheme for Discrete and Secure LoRa Communications," *Sensors*, vol. 22, no. 20, p. 7947, 2022.

[6] W. Shen, P. Ning, X. He, and H. Dai, "Ally Friendly Jamming: How to Jam Your Enemy and Maintain Your Own Wireless Connectivity at the Same Time," in *2013 IEEE Symposium on Security and Privacy*, pp. 174–188, 2013.

[7] Y. Guo and Z. Liu, "Time-Delay-Estimation-Liked Detection Algorithm for LoRa Signals Over Multipath Channels," *IEEE Wireless Communications Letters*, vol. 9, no. 7, pp. 1093–1096, 2020.

[8] M. Chiani and A. Elzanaty, "On the LoRa Modulation for IoT: Waveform Properties and Spectral Analysis," *IEEE Internet of Things Journal*, vol. 6, pp. 8463–8470, May 2019.

[9] L. Vangelista, "Frequency Shift Chirp Modulation: The LoRa Modulation," *IEEE Signal Processing Letters*, vol. 24, pp. 1818–1821, December 2017.

[10] M. Xhonneux, A. Orion, D. Bol, and J. Louveaux, "A Low-Complexity LoRa Synchronization Algorithm Robust to Sampling Time Offsets," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[11] J. Tapparel, O. Afisiadis, P. Mayoraz, A. Balatsoukas-Stimming, and A. Burg, "An Open-Source LoRa Physical Layer Prototype on GNU Radio," in *2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1–5, 2020.

[12] V. D. Pham, P. H. Do, D. T. Le, and R. V. Kirichek, "LoRa Link Quality Estimation Based on Support Vector Machine," *DCCN 2021*, 9 2021.

[13] Ettus, "VERT900 Antenna." https://www.ettus.com/all-products/vert900/.

[14] Murata, "What is the realistic range of LoRa? What is the actual range that can be achieved?."

[15] C. Bernier, F. Dehmas, and N. Deparis, "Low complexity LoRa frame synchronization for ultra-low power software-defined radios," *IEEE Transactions on Communications*, pp. 1–1, February 2020.