When LoRa Meets EMR: Electromagnetic Covert Channels Can Be Super Resilient

Cheng Shen*, Tian Liu*, Jun Huang†✉, Rui Tan‡
*Peking University, †Massachusetts Institute of Technology
‡Nanyang Technological University

2021 IEEE Symposium on Security and Privacy (SP) | 978-1-7281-8934-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/SP40001.2021.00031

***Abstract*—Due to the low power of electromagnetic radiation (EMR), EM convert channel has been widely considered as a short-range attack that can be easily mitigated by shielding. This paper overturns this common belief by demonstrating how covert EM signals leaked from typical laptops, desktops and servers are decoded from hundreds of meters away, or penetrate aggressive shield previously considered as sufficient to ensure emission security. We achieve this by designing EMLoRa – a super resilient EM covert channel that exploits memory as a LoRa-like radio. EMLoRa represents the first attempt of designing an EM covert channel using state-of-the-art spread spectrum technology. It tackles a set of unique challenges, such as handling complex spectral characteristics of EMR, tolerating signal distortions caused by CPU contention, and preventing adversarial detectors from demodulating covert signals. Experiment results show that EMLoRa boosts communication range by 20x and improves attenuation resilience by up to 53 dB when compared with prior EM covert channels at the same bit rate. By achieving this, EMLoRa allows an attacker to circumvent security perimeter, breach Faraday cage, and localize air-gapped devices in a wide area using just a small number of inexpensive sensors. To countermeasure EMLoRa, we further explore the feasibility of uncovering EMLoRa's signal using energy- and CNN-based detectors. Experiments show that both detectors suffer limited range, allowing EMLoRa to gain a significant range advantage. Our results call for further research on the countermeasure against spread spectrum-based EM covert channels.**

I. Introduction

Electromagnetic (EM) covert channel is a well documented threat of data exfiltration where the attacker manipulates the electromagnetic radiation (EMR) of an infiltrated system to encode and leak sensitive information [1], [2], [3], [4], [5], [6], [7]. Unlike conventional covert channels that hide data in legitimate network traffic [8], [9], [10], [11], [12], EM covert channels do not require a compromised protocol stack or network interface, and thus can circumvent a variety of network security measures, such as traffic monitors, firewalls, data diodes, and even air-gaps, where network interfaces are disabled or removed to physically isolate sensitive data from the outside world.

However, due to the low power of EMR, the threat of EM covert channels has been limited to only short-range attacking scenarios. Prior studies show that the EMR of typical CPU-memory systems can be decoded within only 5 m [13], which demands the attacker to be in an unrealistically close proximity to the infiltrated system. Prior attempts to extending communication range rely on directional log-periodic antennas (LPAs) [2]. However, LPAs are bulky, require a line-of-sight signal path, must be precisely pointed to the infiltrated system, and therefore will easily violate the stealthy requirement of covert channel.

In addition to path loss, covert EM signals are subject to significant attenuation caused by shielding. In particular, sensitive government and enterprise systems are typically shielded by metal enclosures compliant with emission security (EMSEC) standards. For example, the TEMPEST standard developed by NATO and the U.S. NSA requires sensitive systems to be protected by "a minimum of 100 dB insertion loss from 1 KHz to 10 GHz" [14], [15], [16]. In comparison, experimental studies show that the EMR of typical CPU-memory systems is only 15 dB above the noise floor [13]. This means that EM signals will be 85 dB (i.e., about 2×10^9 times) weaker than noise after penetrating a TEMPEST shield.

In this paper, we challenge the common belief about the limit of EM covert channels. We ask if an attacker can decode EM covert signals despite deep attenuation, thereby achieving long range or even breaching aggressive shielding previously considered as sufficient to ensure EMSEC. To this end, we revisit the design of EM covert channels with *LoRa* [17] – a state-of-the-art wireless technology that enables kilometer-level connectivity for energy-starved IoTs using *chirp spread spectrum* [18]. The wide success of LoRa in wireless networks raises a natural security question – *"Is it feasible to exploit conventional EM sources as LoRa-like radios to fundamentally escalate the threat of EM covert channels?"*

To demonstrate this feasibility, we develop *Electromagnetic LoRa, EMLoRa* – a super resilient EM covert channel that exploits memory as a LoRa-like radio. EMLoRa's transmitter is a user-space malware that has access to the sensitive data of the infiltrated system. It encodes sensitive data by shaping memory EMR into LoRa-like chirps, which are modulated by manipulating the bit flow written into the memory. EMLoRa's receiver is an inexpensive, portable software radio, which decodes EM chirps to exfiltrate sensitive data from a long distance or behind an aggressive shield.

Different from conventional EM covert channels, EMLoRa represents the first attempt to designing an EM covert channel using spread spectrum technology. However, the CPU-memory system as a covert radio has a unique set of characteristics and brings key challenges that prevent the straightforward application of spread spectrum. From a system angle, the malware transmitter of all EM covert channels faces uncontrollable

✉Corresponding author: junhuang@mit.edu.

DOI 10.1109/SP40001.2021.00031

CPU contentions from legitimate system processes, which pose a fundamental challenge to precisely modulate covert signals, especially for CSS that require accurate control of signal shape. However, all previous EM covert channels assume an ideal host environment, where the transmitter exclusively occupies the victim system's CPU and ignores the interference of contentions. From a signal processing angle, EMLoRa must handle the complex spectral characteristics of memory EMR, and prevent adversarial detectors who know the design and modulation parameters of EMLoRa from demodulating covert signals and achieving the same gain of attenuation resistance. In this paper, we demonstrate how attackers circumvent these challenges to make EMLoRa feasible, reliable, and stealthy.

We have implemented EMLoRa by employing BladeRF [19] – an inexpensive and portable software radio – to receive covert EM signals. Extensive experiments on desktops, laptops, and servers show that EMLoRa's signal can be decoded from up to 250 m away, or penetrate EM shield of up to 78 dB without requiring specialized radio accessories such as high-gain LNA and directional antenna (which, if available, will provide an additional gain of 22 to 70 dB at the receiver side), thus posing a serious threat to the 100 dB bar specified by the TEMPEST. Further, we demonstrate how EMLoRa enables attacks in three previously impossible scenarios.

• *Wide-area data exfiltration.* Security-critical organizations typically enforce a perimeter around classified area wherein unauthorized devices are exclusively prohibited. By achieving long-range, EMLoRa enables wide-area data exfiltration *over* security perimeter by allowing the attacker to deploy receivers in public areas outside the perimeter. Real-world experiments demonstrate how EMLoRa signals penetrate walls and be decoded in a public area 120 m away from the building. We then further conduct high-fidelity ray tracing-based emulations, which show how receivers deployed in Maryland Avenue Linear Park and Dongdan Sports Center successfully decode EMLoRa signal leaked from the U.S. Department of Social Security and the Ministry of Commerce of China, respectively.

• *Penetrating Faraday cage.* With EMLoRa, we demonstrate data exfiltration from systems shielded in Faraday cages. A recent study show that magnetic covert channel can penetrate Faraday cages thanks to the immunity of magnetic field to electromagnetic shielding [20]. However, the magnetic covert channel requires a line-of-sight path and the range is only 100cm to 150cm due to the weak amplitude of magnetic field. In comparison, EMLoRa can penetrate a Faraday cage even when the receiver and the shielded system are deployed in different rooms separated by multiple concrete walls.

• *Localization of air-gapped devices.* We demonstrate that, once infiltrated by EMLoRa, the victim device can be wirelessly localized in a wide area using just a small number of inexpensive sensors. This is achieved by exploiting the memory of the infiltrated device as a beacon signal transmitter. Previous wireless localization systems all rely on radios, such as Bluetooth, Wi-Fi, or GPS modules. In contrast, EMLoRa is the first system that can localize a device even when it is air-gapped to hide location.

In addition, we discuss how to uncover EMLoRa signals using both energy- and convolutional neural network (CNN)-based detectors. We show that the both detectors suffer limited range, allowing EMLoRa to gain a significant advantage. The result calls for further research on countermeasures against spread spectrum-based EM covert channels.

In sum, this paper makes the following contributions.

- We explore the limit of EM covert channels by developing EMLoRa, a super resilient EM covert channel that exploits memory as a LoRa-like radio. To the best of our knowledge, EMLoRa is the first attempt to designing an EM covert channel using spread spectrum technology.
- We evaluate the performance of EMLoRa based on extensive experiments on laptops, desktops and servers. We then further demonstrate how EMLoRa enables data exfiltration and device localization attacks in three previously impossible scenarios.
- We develop energy- and CNN-based detectors to uncover EMLoRa signals, and then conduct experiments to reveal their limitations.

II. Related Work

Covert/side channels. To circumvent network security measures, recent data exfiltration attacks exploit physical covert channels generated by magnetic field [20], thermal emanation [21], or backscattering ambient signals transmitted by nearby wireless networks [22]. However, these attacks suffer from very short communication range (i.e., typically less than 2m). Acoustic covert channels based on near-ultrasonic emissions can achieve 20m [23]. Unfortunately, near-ultrasound is partially audible, especially to youths.

EMLoRa is most related to prior attacks based on EMR. EM side channels aim to extract information (such as cryptographic keys) from unintentional EMRs [24], [25], [26], [27]. More recently, researchers leverage EM side channels of CPU and memory for attestation [28], memory profiling [29], malware detection [30], [31], [32], neural network reverse engineering [33], and wireless eavesdropper detection [34].

Different from these *passive* analysis, EM covert channels aim to leak secret information by *actively* manipulating EMR. For example, Kuhn et al. [1] demonstrate an EM covert channel that manipulates display content to modulate the EMR of monitor. The resulted communication channel achieves long range due to high power of monitor EMR. However, the manipulated display content can be easily noticed by the user of monitor. Recent studies exploit the EMRs of peripheral [3], power management unit [4], and memory [2], which ensure stealthy communication but suffer from very low EMR power. In particular, previous studies of EM covert channels commonly use binary modulation to simplify the design. This allows them to sidestep a set of key challenges, such as handling the complex spectrum characteristics of EMR and tolerating internal resource contentions that interfere with

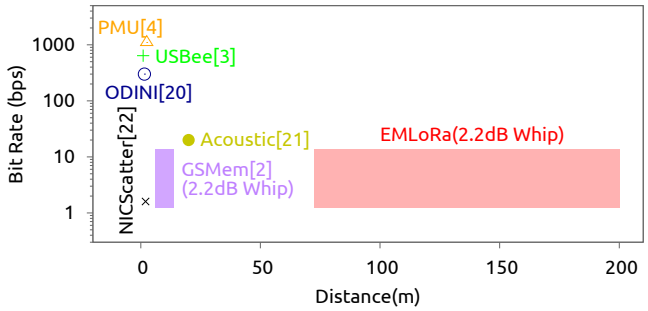


Fig. 1: Comparison between EMLoRa and related work.

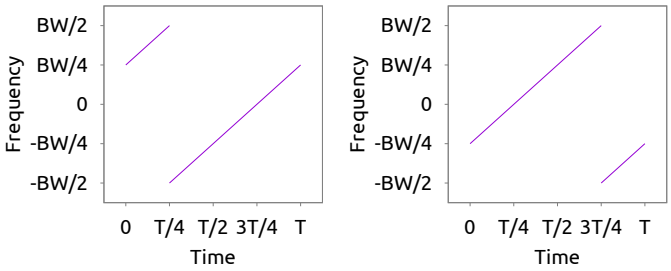


Fig. 2: Two baseband LoRa chirps cyclically shifted by $\frac{T}{4}$ and $\frac{3T}{4}$, respectively, under a spreading factor of 2.

precise shaping of EM signal[1]. However, binary modulation schemes are well-known to be vulnerable against attenuation.

Compared with prior physical and EM covert channels, EMLoRa aims at an unexplored challenging trade-off between bit rate and attenuation resistance, as shown in Fig. 1. By achieving this trade-off, EMLoRa enables previously impossible attacks, such as exfiltration of short secrets (plain password, MD5, disk encryption key, etc.) within a couple of minutes over a long range or through an aggressive shielding. It is worth noting that, although the bit rate of EMLoRa is lower than some of existing short-range covert channels, enabling spread spectrum allows EMLoRa to significantly improve the trade-off between bit rate and range. For example, when equipped with the same type of antenna (2.2 dB whip), EMLoRa improves the range by more than 20x when compared with GSMem [2] at the same bit rate.

BitJabber [5] – a study parallel to our work – demonstrated a fast EM covert channel by upgrading the binary FSK used by conventional EM covert channels to M-FSK. A common feature of BitJabber and EMLoRa is to explicitly address spread spectrum clocking, which leads to improved communication performance. However, BitJabber focuses on fast transmission in short-range scenarios. It requires the receiver to reverse spread spectrum clocking, which is impossible when received signals are deeply buried by noise.

LoRa. LoRa is a state-of-the-art wireless standard for low-power wide-area networking (LPWAN). Recently, significant research has been devoted to improve the range and scalability of LoRa. For example, Choir [35] exploits the frequency offsets between LoRa transmitters to separate collided LoRa packets. FTrack [36] leverages both time and frequency domain features to enable parallel decoding of LoRa signal. Charm [37] enhances the coverage of LoRa-based LPWANs using distributed MIMO and coherent combining. Inspired by LoRa's super resilience to attenuation, recent studies transplant LoRa to backscatter systems, which enable long range communication at a very low energy cost [38], [39].

Different from prior studies of LoRa, we explore the security implication of LoRa on EM covert channels. We show that the power spectrum of EMR presents complex characteristics, which invalidate trivial transplantation of standard LoRa encoding/decoding algorithms used on conventional radios. This paper tackles this challenge and demonstrate the first malicious use of state-of-the-art spread spectrum technology on EM covert channels.

[1]For example, binary amplitude-shift keying (ASK) encodes zeros and ones using different power levels. Therefore, an ASK receiver only needs to know the total energy of signals received on the channel, rather than caring about fine-grained spectrum characteristics, e.g., the distribution of signal energy across frequency. As such, the ASK receiver is also less sensitive to signal distortions occurred within the channel bandwidth.

III. A LoRa Primer

LoRa uses *chirp spread spectrum (CSS)* at its PHY. In this section, we present a brief primer on CSS modulation and demodulation.

Modulation. CSS modulates signals as up-chirps whose frequency linearly increases with time over a predefined bandwidth. Mathematically, an up-chirp can be expressed as,

$$C(t) = e^{j2\pi t(f_0 + \frac{BW}{2T}t)}, \tag{1}$$

where f_0 is the initial frequency, BW and T are chirp bandwidth and duration, respectively.

CSS represents different bits by introducing different *cyclic time shifts* to the *base chirp* defined in Eqn. 1. Specifically, encoding an N-bit symbol needs 2^N cyclic time shifts, where N is termed as the *spreading factor*. Mathematically, a chirp cyclically shifted by ΔT can be expressed as,

$$S(t, \Delta T) = C(t)e^{j2\pi t\phi(t,\Delta T)},$$

where

$$\phi(t, \Delta T) = \begin{cases} (1 - \frac{\Delta T}{T})BW & 0 \le t < \Delta T \\ -\frac{\Delta T}{T}BW & \Delta T \le t < T. \end{cases}$$

Demodulation. Denote a chirp submerged under noise as,

$$S'(t, \Delta T) = S(t, \Delta T) + N(t),$$

where $N(t)$ is noise. During demodulation, the receiver first *dechirps* the received signal by multiplying $S'(t, \Delta T)$ with $\overline{C}(T)$, i.e., the conjugate of the base chirp. The result can be expressed as,

$$S'(t, \Delta T)\overline{C}(t) = e^{j2\pi t\phi(t)} + N(t)\overline{C}(t). \tag{2}$$

The receiver then takes an FFT on Eqn. 2, which will produce two peaks at $BW\frac{\Delta T}{T}$ and $BW\frac{T-\Delta T}{T}$ in frequency domain, where peak frequencies can be used to decode the data represented by ΔT. In comparison, the noise term in Eqn. 2, i.e., $N(t)\overline{C}(t)$, will be *spread* over BW (because time-domain multiplication is equivalent to frequency-domain convolution),

leading to a low energy intensity. As a result, the signal to noise ratio (SNR) of the received chirp will be significantly boosted, allowing decoding even when the received chirp is deeply drowned in noise.

IV. EMLoRa: An Overview

In this section, we first discuss the attack model of EMLoRa and then highlights its key designs.

A. Attack Model

Transmitter. EMLoRa shares the same assumption with conventional EM covert channels at the transmitter side (i.e., requiring only a user-space malware with no root privilege). We assume the malware has already been planted to the victim's system. This can be done by social engineering, file sharing, or bundling a legitimate software [40]. Alternatively, the malware can be installed by a malicious owner of a shared, corporate, or public computer. We further assume that the malware has access to sensitive/private data of the host. To achieve this, it may masquerade as a legitimate software or exploit a side-channel [41], [42].

Receiver. EMLoRa features a fundamentally escalated threat model at the receiver, where it allows the attacker to decode covert signals from a long distance away or behind an aggressive shielding. We assume EMLoRa's receiver roughly knows the memory frequency of the transmitter. Specifically, memory frequency is an integer multiple of two base frequencies including 100 MHz and $133\frac{1}{3}$ MHz. In practice, the clock frequency of a modern DDR has 13 possible values ranging from 200 MHz of DDR2-400 to 1600 MHz of DDR4-3200. To determine which frequency is being used by the transmitter, EMLoRa can simply scan these channels to search for EM chirps. However, due to minute manufacture variance, memory frequencies differ across individual memories. We will discuss fine-grained frequency acquisition in section V-C.

B. Challenges and Key Designs

To communicate over an EM covert channel using a spread spectrum scheme like LoRa, EMLoRa faces a set of unique challenges. In the following, we discuss these challenges and sketch our solutions.

- **Challenge**: Designed as a user-space malware, EMLoRa's transmitter suffers jitters and CPU contentions from legitimate processes, which interfere with EMLoRa's memory control, leading to distortion of EM chirps.

  Solution: At the transmitter side, EMLoRa employs a jitter-resilient modulation scheme to shape memory EMR through approximate chirp synthesization. To tolerate CPU contention, EMLoRa monitors the intensity of CPU contention at run-time and then adaptively trades-off bit rate with signal resilience by tuning chirp shapes.
- **Challenge**: Unlike standard radios whose carrier signals have a steady frequency, the frequency of memory EMR is constantly oscillating due to the effect of spread spectrum clocking. In particular, spread spectrum clocking disperses EMR energy over frequency, which not only reduces the signal-to-noise ratio (SNR) of EMLoRa signal significantly but also invalidates trivial transplantation of standard LoRa encoding/decoding algorithms.

  Solution: At the receiver side, we propose a novel use of folding [43] – a signal processing algorithm previously used in large radio telescopes to search for weak signals – to mine and fuse noise-buried EM signals dispersed by spread spectrum clocking. In addition, EMLoRa exploits unique spectrum characteristics of memory EMR to reduce signal processing overhead in the synchronization phase.
- **Challenge**: By using a spread spectrum scheme, EMLoRa's signal will be much more visible in wireless spectrum such that an adversarial detector knowing EMLoRa's design and modulation parameters can use the same receiving algorithm to decode the covert signal, thereby defeating the purpose of stealth communication.

  Solution: To prevent adversarial detectors from demodulating covert signals, EMLoRa uses a secret sequence shared by transmitter and receiver to permutate signals within an EM chirp. Notice that synchronizing with a permutated chirp will increase signal processing overhead, we propose a simple signal processing method that reduces synchronization complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$, where n is the chirp length.

V. Design of EMLoRa

This section presents the design of EMLoRa in detail. We first model memory as a complex radio, and then base on the model to design EMLoRa's transmitter and receiver. After that, we discuss how to adapt EMLoRa based on CPU contention, and hide its signal from adversarial detectors.

A. Memory as a Complex Radio

To underpin the design of EMLoRa, we model memory system as a radio to characterize the spectrum of memory EMR. At a high-level, memory clock acts like a local oscillator, and its EMR provides a carrier that can be modulated by memory activities. Memory bus radiates the modulated EMR as an antenna, thereby creating a covert channel. While previous studies have explained how memory activity modulates memory clock EMR [13], they assume a simple clock with a fixed, constant frequency. In practice, memory systems by default use *spread spectrum clock (SSC)* to meet electromagnetic compatibility regulation. In the following, we focus on modeling the complex spectral characteristics resulted by the interaction between SSC and memory activities.

Modeling SSC. A SSC is generated by modulating the frequency of a simple clock (e.g., a square wave), which causes clock energy to spread over frequency, leading to a reduced EMR intensity. Specifically, a SSC can be expressed as,

$$s_{\text{ssc}}(t) = \cos(2\pi f_0 t + \frac{\Delta f}{f_m} \sin(2\pi f_m t)), \tag{3}$$

where f_{clk} is the frequency of the simple clock, f_m and Δf are the modulating frequency and peak frequency deviation,

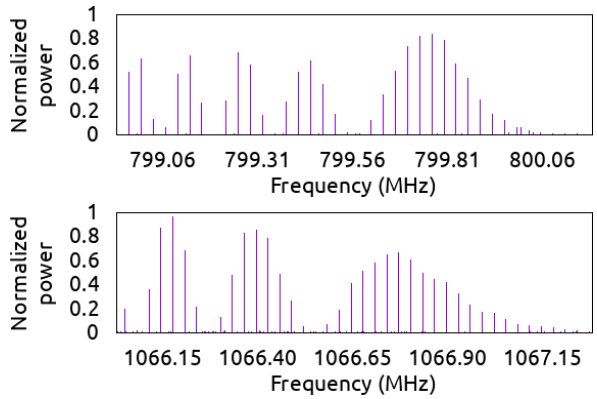


Fig. 3: The power spectra of memory clock EMRs of a DDR3-1600 (top) and a DDR4-2133 (bottom).

respectively. Mathematically, the spectrum of Eqn. 3 can be calculated as [44],

$$\|\mathcal{F}_{ssc}(f)\| \stackrel{\text{def}}{=} \|\sum_{n} J_{\text{n}}(\frac{\Delta f}{f_m})(\delta(f - f_{clk} + nf_m) - \delta(f - f_{clk} - nf_m))\|. \tag{4}$$

where $J_{\text{n}}(\cdot)$ is the Bessel function of the first kind, $\delta(\cdot)$ is the Dirac delta function.

Eqn. 4 indicates that the energy of a SSC is non-zero only at $f_{clk} \pm nf_m$. In other word, a SSC is the sum of a series of *sub-clock* components distributed at $f_{clk} \pm nf_m$. As two examples, Fig. 3 plots the spectra of memory clock EMRs for a DDR3-1600 and a DDR4-2133[2]. We observe that memory clock spectrum demonstrates a series of energy peaks, where each peak corresponds to a sub-clock. We further notice that all peaks are confined in an 1 MHz frequency range at the lower side-band of f_{clk}. This is because of the band-pass filter (BPF) of the memory clock generator, which removes frequency components higher than f_{clk} and limits the maximum frequency deviation of SSC to maintain reliable synchronization between memory and memory controller.

As a result, we can re-write the SSC as,

$$s_{\text{ssc}}(t) = \sum_{n=0}^{N} A_{\text{ssc}}(n)\sin(2\pi f_{ssc,n}t), \tag{5}$$

where $A_{\text{ssc}}(n)$ and $f_{ssc} = f_{clk} - nf_m$ are the amplitude and frequency of the n-th sub-clock, respectively; N is the total number of sub-clocks contained in the memory clock, which depends on the pass-band of the clock generator's BPF.

Modulating SSC. Consider a series of periodic memory accesses performed at clock edges. The corresponding current flow and the produced EMR can be approximated as a square wave sampled at clock edges. Specifically, the square wave can be expressed using its Fourier series as follows,

$$s_{\text{acc}}(t) = \frac{2A_{\text{acc}}}{\pi}\sum_{m}\frac{\cos((2m-1)2\pi f_{\text{acc}}t)}{2m-1}.$$

where f_{acc} and A_{op} denote the frequency and amplitude of the current produced by periodic memory accesses, respectively.

Because memory accesses are performed at clock edges, in effect, they introduce an amplitude modulation to the current of clock signal. From a radio perspective, this is equivalent to frequency mixing, where the clock provides a local oscillator and the current of periodic memory access can be treated as an input signal. Because $s_{\text{ssc}}(t)$ has multiple sub-clocks and $s_{\text{acc}}(t)$ is composed of multiple harmonics, AM modulation will occur between *each pair* of sub-clock and harmonic. This can be expressed as,

$$s_{\text{mod}}(t) = s_{\text{ssc}}(t)s_{\text{acc}}(t) = \sum_{n}\sum_{m}\frac{A_{\text{ssc}}(n)A_{\text{acc}}}{(2m-1)\pi}s_{\text{img}}(n,m,t), \tag{6}$$

where

$$s_{\text{img}}(n,m,t) = \sin(2\pi(f_{\text{ssc},n} + (2m-1)f_{\text{acc}})t) + \sin(2\pi(f_{\text{ssc},n} - (2m-1)f_{\text{acc}})t)$$

is a pair of *mirror images* generated by modulating the n-th sub-clock of $s_{\text{ssc}}(t)$ using the m-th harmonic of $s_{\text{acc}}(t)$.

The model given in Eqn. 6 indicates that, by modulating memory clock with periodic memory accesses, each sub-clock will be associated with a pair of mirror images as well as their harmonics. Specifically, for the n-th sub-clock at $f_{\text{ssc},n}$, the mirror images corresponding to the i-th harmonic of f_{acc} will appear at $f_{\text{ssc},n} \pm (2i-1)f_{\text{acc}}$.

B. EMLoRa Transmitter

The goal of EMLoRa's transmitter is to exploit the AM modulation effect of periodic memory activity to synthesize EM chirps. In the following, we develop a malware transmitter and then base on the model established in section V-A to summarize the spectrum characteristics of generated signal.

Synthesizing EM chirps. We develop EMLoRa's transmitter by augmenting the square wave EMR generator [13] (i.e., the squareWave function of Fig. 5). As the code shows, squareWave writes two equal-size byte blocks into memory,[3] creating a current flow that approximates a square wave. The size of byte block determines wave frequency.

A naive method of generating an EM chirp is to iteratively call squareWave with linearly increasing frequency. Unfortunately, we find that chirp energy generated in this way is very low because of inaccurate timing. Specifically, the getTime function in Fig. 5 introduces jitters such that each generated wave is subject to a random frequency error. As a result, chirp energy will disperse in frequency domain.

EMLoRa addresses this problem through *approximate chirp synthesization*. Specifically, it approximates a chirp as a sequence of discrete frequency levels, as illustrated in Fig. 4. When synthesizing an EM chirp, it generates multiple square waves at each frequency level for a certain dwell time. At each frequency level, because square waves are independently affected by jitters, the average of their actual frequencies tends

[2]The clock frequency of a DDR is the half of memory speed, i.e., the f_{clk} of DDR3-1600 and DDR4-2133 are 800 MHz and 1067 MHz, respectively

[3]To directly write memory, one can use _mm_stream_si128 – a user-space function – to bypass cache.

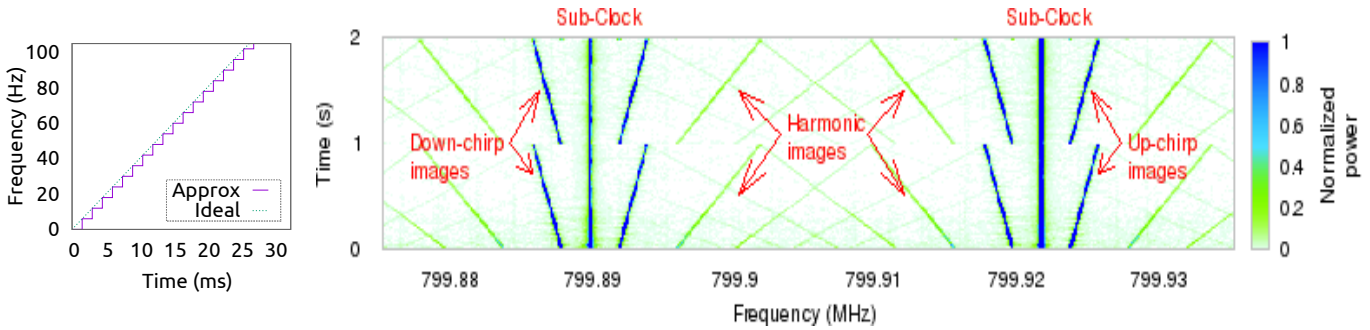


Fig. 4: Approximate chirp synthesization (left) and the spectrum of two EM chirps modulated around two sub-clocks (right).

```
1  // Return system time in nano-seconds
2  static inline uint64_t getTime() {
3    struct timespec t;
4    clock_gettime(TIME_ABSOLUTE, &t);
5    return t.tv_sec*NSEC_PER_SEC+t.tv_nsec;
6  }
7  // Generate a square wave EMR
8  static inline void squareWave(float wavePeriod){
9    uint64_t start = getTime();
10   while (getTime() < start+wavePeriod/2)
11     _mm_stream_si128(&reg, reg_one);
12   while (getTime() < start+wavePeriod)
13     _mm_stream_si128(&reg, reg_zero);
14 }
15 // Generate a chirp based on given data and
       spreading factor (SF)
16 void approxChirp(float freqLow, float freqHigh,
       int data, int stepSize, int SF) {
17   const float dwellTime = 1e6; //ns
18   const float BW = freqHigh-freqLow;
19   const float shift = BW*data/pow(2, SF);
20   for (int level = freqLow; level < freqHigh;
         level+=stepSize) {
21     float wavePeriod = 1e9/((level+shift)%BW);
22     uint64_t t0 = getTime();
23     while (getTime() < t0+dwellTime)
24       squareWave(wavePeriod);
25   }
26 }
```

Fig. 5: The code of EMLoRa transmitter.

toward a normal distribution, whose mean equals the intended frequency level, and the variance decreases as the number of square waves increases (as established by the Central Limit Theorem). Therefore, the longer the dwell time, the stronger the EMR energy at each frequency level. Based on empirical observation, we find that a dwell time of 1.5*ms* suffice to accumulate a sufficient level of energy. We then set the step size between frequency levels to 6 Hz to limit the discontinuity of the generated chirp.

We determine EMLoRa's modulation parameters, including chirp bandwidth and starting frequency, based on empirical measurements. Unlike conventional radios, EMLoRa's malware transmitter suffers jitters caused by the inaccurate user-space timer and the contention of legitimate system processes. Specifically, when modulating EM waves, the ratio between jitter and wave period increases with EM wave frequency, introducing a higher amount of signal distortion. As a result, the upper-bound frequency of EMLoRa signal is inherently limited by jitters. Moreover, we observe that the power of synthesized EMR decreases as the intended frequency level increases. This is because, at higher frequency levels, the impact of jitter gets relatively amplified, which intensifies the dispersion of EMR energy. Meanwhile, in order to avoid self interference, EMLoRa should insert a *guard band* to separate sub-clock and the synthesized chirps. Based on experiment results, we set chirp bandwidth to 2000 Hz, and insert a 2000 Hz guard band between sub-clock and chirps.

Spectrum of EM chirps. Fig. 4 shows the power spectrum consisting of two EM chirps synthesized by the malware transmitter of EMLoRa. As characterized by the model established earlier, the spectrum consists of three components, including,

- *Sub-clocks*. Consecutive sub-clocks of SSC are separated by a constant interval of f_m in frequency domain, where f_m is the modulation frequency of the SSC generator.
- *Mirror chirp images*. Each sub-clock component is associated with a pair of mirror chirp images, including an up-chirp and a down-chirp.
- *Harmonic chirp images*. Each up- and down-chirp image have a sequence of harmonic chirp images. For the n-th harmonic chirp, the bandwidth is n times wider, and the power degrades exponentially fast as n increases.

Encoding and framing. Same as LoRa, EMLoRa encodes data by cyclically shifting the base EM chirp. As shown in the code of approxChirp in Fig. 5, the transmitter first calculates a shift based on the given data and spreading factor (line 20), then shifts the base chirp cyclically during approximate chirp synthesization (line 22).

The transmitter then groups encoded EM chirps into a frame. Each frame begins with a preamble chirp, followed by a group of data chirps and a parity chirp. The preamble is a base chirp carrying no data. It allows the receiver to detect and synchronizae with the frame.

C. EMLoRa Receiver

At a high-level, EMLoRa's receiver works in two steps, including *(i) synchronization*, where it detects and locates the first signal sample of an incoming frame, and *(ii) demodulation*, where it demodulates data chirps to extract bits. In both phases, EMLoRa leverages the unique spectral characteristics of EMLoRa signal to boost SNR and reduce signal processing overhead. In the following, we describe the design of EMLoRa receiver in detail.

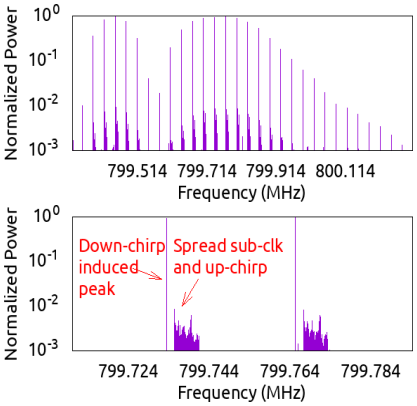


Fig. 6: The spectrum of EMLoRa after dechirping using a base down-chirp.

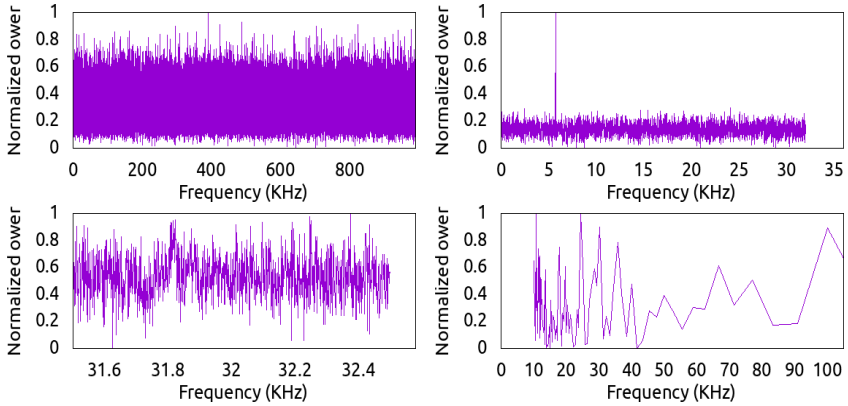


Fig. 7: Comparison between folding, auto-correlation, and FFT when detecting dechirped peaks in the presence of significant noise.

Dechirp. In both synchronization and demodulation phase, EMLoRa dechirps received signals like standard LoRa. Specifically, it first collects a window of signals, where the window size equals chirp duration. Then, it multiplies received signals with the conjugate of a base chirp and then takes FFT.

However, unlike standard LoRa, EMLoRa signal consists of sub-clocks, mirror chirps, and harmonic images, which will be transformed in different ways after dechirp. As an example, Fig. 6 plots the spectrum of EMLoRa' signals after dechirping using a base down-chirp. As shown in the figure, dechirp transforms all down-chirp images into frequency-domain peaks. In comparison, other signal components and noise are spread over frequency, which leads to a significantly reduced energy intensity, yielding a boosted SNR. The case of decrhiping using a base up-chirp will be similar.

Synchronization Suppose an EMLoRa frame is incoming and its preamble is Δt ahead the first signal sample of the receiver's dechirp window. When Δt is smaller than chirp duration, a part of the preamble will fall into the current dechirp window. The goal of synchronization is to detect the presence of preamble and estimate Δt. A naive solution is to search the entire dechirp window, which however would incur a significant cost.

To reduce signal processing overhead, EMLoRa leverages the unique symmetric property of mirror chirps to walkaround searching. Specifically, EMLoRa first dechirps received signals using *both* up- and down-chirps. Then, each pair of mirror chirps will be transformed into two peaks. For the mirror chirps associated with the n-th sub-clock, the pair of dechirped peaks will appear at $f_{ssc,n} \pm (\frac{BW}{T}\Delta t + G)$, where BW and T are chirp bandwidth and duration, respectively; G is the gap band inserted between chirps and sub-clocks; and $f_{ssc,n}$ is the frequency of the n-th sub-clock. If the receiver can determine the frequency interval between dechirped peaks of mirror chirps, then Δt can be directly estimated as $\Delta t = \frac{T}{2BW}\Delta f$, where Δf is the frequency interval between dechirped peaks.

However, under severe attenuation, dechirped peaks will be deeply buried underneath noise, which presents a significant challenge to EMLoRa receiver. To detect the incoming preamble, the receiver needs to fuse dechirped peaks distributed across frequency to boost SNR.

To tackle this challenge, EMLoRa leverages the fact that adjacent dechirped peaks are separated by f_m, thereby demonstrating periodicity in frequency-domain. Based on this observation, EMLoRa performs *folding* – a fast algorithm originally used to amplify periodic astronomical signals received by large radio telescopes [45], [46]. Suppose $\mathcal{P}$ represents the series of N signals and $\mathcal{P}[i]$ ($i \in [1, N]$) is the amplitude of the ith signal. To search for a signal with a period of K, the spectrum is first divided into small windows of K points and then added in a window-wise fashion as,

$$\mathcal{F}_K[i] = \sum_{j=0}^{\lfloor \frac{N}{K} \rfloor - 1} \mathcal{P}[i + j * K].$$

After folding, the energies of periodic signals will be fused while the sum of random noise will have a lower gain.

EMLoRa applies folding over the dechirped spectrum to fuse the energy of dechirped peaks. Because the f_m of transmitter is unknown, EMLoRa folds the dechirped spectrum at all possible f_m to search for fused peaks. We note that the search-induced computational overhead is low because folding involves only simple arithmetic. Moreover, our empirical measurements show that the f_m of SSC generators typically varies between 30 KHz to 35 KHz, which allows EMLoRa to confine search in a small range.

Fig. 7 compares the results of folding, auto-correlation, and FFT when searching for dechirped peaks in the presence of significant noise. As shown in Fig. 7(c) and Fig. 7(d), auto-correlation and FFT performs poorly. Specifically, FFT fails to identify the dechirped peaks due to its poor resolution. Auto-correlation identified a peak but the amplitude is extremely weak and thus is susceptible to noise. As a comparison, Fig. 7(b) plots the result when folding the dechirped spectrum at the transmitter's f_m, where the fused peaks can be clearly identified. Once the pair of fused peaks corresponding to both mirror chirp images are identified, EMLoRa can directly calculate the Δt of the incoming preamble based on the frequency interval between fused peaks.

It is worth noting that EMLoRa's receiver need not perform syhchronization for each received signal sample. Instead, syn-

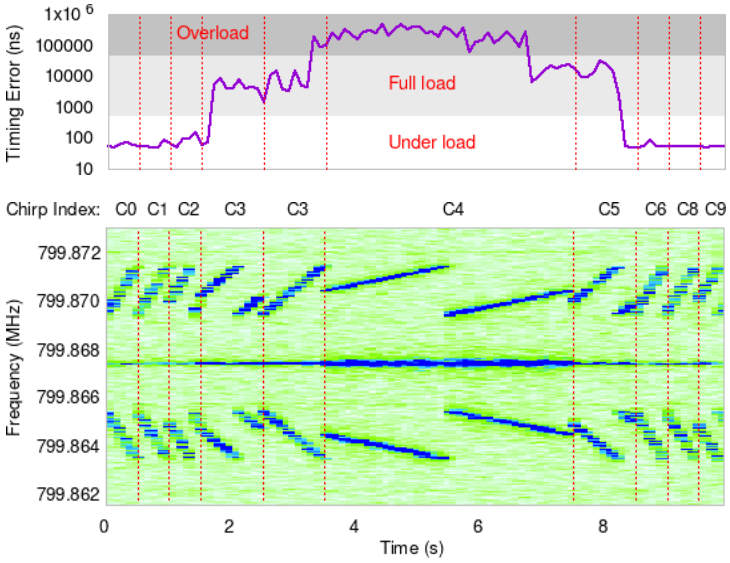


Fig. 8: An example of EM chirp adaption under varying CPU load. EMLoRa transmits a long chirp between 3.5 s and 7.5 s to resist the high timing error caused by intensive CPU contention.

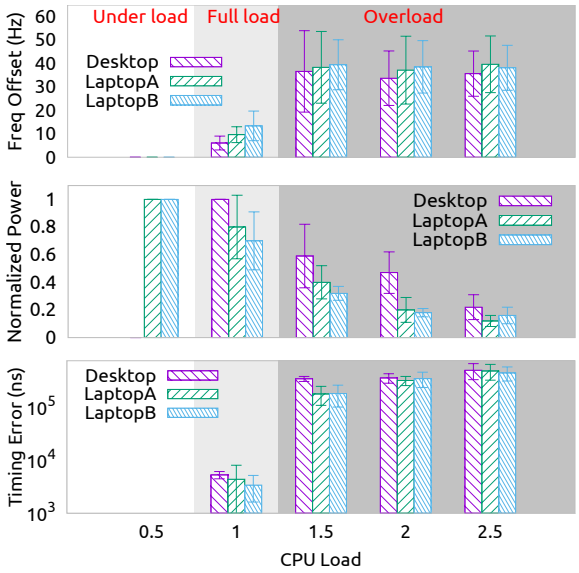


Fig. 9: Impacts of CPU contention on the frequency error (top), power (middle), and timing error (bottom) of EM chirps.

chronizing every half chirp duration is enough to ensure no frame is missed.

Demodulation. After synchronization, EMLoRa demodulate data chirps like a standard LoRa receiver except the following signal processing to boost SNR. First, EMLoRa fuses the energy of mirror chirps, which are separated by $2G$ in frequency, where G is the size of guard band. To this end, EMLoRa first dechirps using up- and down-chirp, and then shifts the obtained spectra by $\pm G$ and sums them together. Second, like what it did in the synchronization phase, EMLoRa fuses dechriped peaks distributed over spectrum by performing folding at f_m. Note that EMLoRa need not search all possible f_m in the demodulation phase, because f_m is already determined after synchronization.

In CSS, the drift of signal frequency does not affect demodulation as long as it is smaller than $\frac{BW}{2}SF$, where BW is the chirp bandwidth and SF is the spreading factor. We observe that memory frequency drift between consecutive chirps is insignificant when compared with this threshold. To mitigate long term effect during the entire packet cycle, EMLoRa receiver tracks accumulated frequency drift using standard phase-locked loops.

Frame decoding. Using the above method, the receiver decodes incoming chirps one by one until the fused peak drops to below the noise floor of fused spectrum, which indicates the end of frame. The receiver then checks the integrity of the frame using the parity bits received in the last chirp.

D. Adaption to CPU Contention

CPU contentions of legitimate processes may block the malware transmitter of EMLoRa, introducing significant jitters and frequency errors to distort EM chirps. Next, we explore the feasibility of tackling this issue with a rate adaption layer atop EMLoRa's transmitter and receiver. It is worth noting that, unlike CPU contention, we observe that the contention over memory is only transient and thus has much less impact on EMLoRa. This should be attributed to multi-level cache and the high speed of memory bus, which effectively resolves memory contentions incurred by legitimate processes. For this reason, we only consider CPU contention in the design of rate adaption.

In the following, we begin by conducting benchmarks to characterize the impacts of CPU contention on EM chirps. Based on the results, we engineer a rate set that provides different trade-offs between bit rate and signal resilience, and then augment EMLoRa to adapt rate. Fig. 8 shows an example of EM chirp adaption under varying CPU load. We next discuss how EMLoRa achieves this in detail.

Impacts of CPU contention. To understand the impacts of CPU contention, we run benchmarks on two laptops and one desktop that differ in memory type, frequency, and the number of CPU cores (as listed in Table II). We use stress to control CPU load, which is calculated as the ratio between the numbers of processes and CPU cores. We then study the distortion of EM chirps by measuring their frequency errors and power degradation.

Interestingly, we find that CPU contention affects frequency error and chirp power in different ways. As shown in Fig.9, chirp frequency error increases from 0 Hz to 35.7 Hz as CPU load increases from 0.5 to 1.5. However, further increasing CPU load will not see a commensurate increase of frequency error. In particular, we observe that chirp frequency error is bounded by 50 Hz on all the three systems. In contrast, as Fig. 9 shows, chirp power continues to degrade with the increase of CPU load.

We believe that the different impacts on frequency error and chirp power should be attributed to CPU scheduling strategy. Specifically, chirp frequency error is caused by jitters, which

TABLE I: Adaptive rate set.

Contention Level	Chirp Duration (s)	SF	Bitrate (bps)
Underload	0.5	7	14
Fullload	1	6	6
Overload	4	5	1.25

is determined by the *waiting time* of EMLoRa's malware. In contrast, chirp power is affected by *CPU utilization*, which determines how long the malware occupies memory bus. We conjecture that scheduling algorithms on commodity CPUs are tasked to maintain predictable and consistent waiting time for processes when CPU is overload, thus imposing a bound on chirp frequency error. In contrast, the CPU utilization of the malware continues to decrease as CPU load increases, yielding a degraded chirp power.

Engineering rate set. Based on benchmark results, we qualitatively classify CPU contention into three levels including *underload*, *fullload*, and *overload*, and then engineer a corresponding rate set, as defined in Table I. The basic rate for underload is determined by empirically tuning the balance between bit rate and attenuation resilience. When CPU becomes fullload or overload, EMLoRa increases chirp duration to compensate the loss of emission power, and decreases spreading factor to tolerate frequency error. Specifically, it is easy to compute that, a k times longer chirp duration improves signal resilience by k times. On the other hand, to tolerate a frequency error of ϵ, the maximum spreading factor should be $\lfloor \log_2 \frac{BW}{2\epsilon} \rfloor$, where BW denotes chirp bandwidth. Based on this observation, we set spreading factors to 6 and 5 under fullload and overload, respectively. The chirp duration is set to 1s and 4s, which are 2x and 8x longer than that of the basic rate. This allows EMLoRa to resist a CPU load of up to 2x. We note that higher CPU loads are not common in practice.

Run-time adaption. We then augment EMLoRa's malware transmitter to classify CPU contention level and adapt bit rate. Specifically, EMLoRa adapts rate based on the timing error of square waves. As show in Fig. 9, timing error is strongly correlated with CPU contention, making it a good metric to navigate rate selection. To measure timing error, EMLoRa instruments the squareWave function of the malware to measure the timing error of each generated square wave. The measured timing error is then compared with predefined thresholds to determine CPU contention level, which guides the malware to select a rate for the next chirp. To determine the thresholds of timing errors, EMLoRa can either actively measures it by stimulating the CPU, or passively profiles timing error distribution of the infected system.

When demodulating an incoming data chirp of an unknown rate at the receiver side, EMLoRa first dechirps it using all three rates and then examines which one yields the highest energy. This allows EMLoRa to bypass explicit rate indication and thus avoid protocol overhead.

E. Combating Adversarial Detectors

We next discuss how to prevent adversarial detectors who know the design and modulation parameters of EMLoRa from demodulating covert signals and achieving the same gain of attenuation resistance. The basic idea is to permutate signals of an EM chirp based on a secret sequence shared by EMLoRa's transmitter and receiver. For each chirp, a different sequence is used to avoid time-domain correlation, which may yield a correlation peak that is detectable. Such secret sequences can be hard coded into EMLoRa's malware before planting into the victim system. Without knowing the secret sequence, the adversarial detector cannot recover EM chirps and thus will fail to demodulate EMLoRa's chirps.

TABLE II: Configurations of infiltrated systems.

	Desktop	LaptopA	LaptopB	Server
Memory	DDR3	DDR4	LPDDR3	DDR3
Freq. (MHz)	1333	2400	2133	1333
Mem. Channel	2	2	2	4
CPU	Pen G3420	i7-3520	i5-8520U	Xeon E5
Cores	2	8	8	8
Shield	ABS plastic	ABS plastic	Al alloy	Al alloy

However, synchronizing with a permutated preamble chirp will increase the complexity of signal processing. For example, consider an incoming preamble $s(t)$, which partially falls into the current dechirp window of EMLoRa. Denote the permutated chirp as $x(t)$. A naive method of synchronization is to compute the cross-correlation between $s(t)$ and $x(t)$ at all possible timing shifts, which will incur a signal processing overhead of $\mathsf{O}(n^2)$, where n is the chirp length. Specifically, the correlation can be expressed as,

$$(x \star s)[n] \stackrel{\text{def}}{=} \sum_{k=-\infty}^{\infty} x[k] s^*[n+k] \tag{7}$$

To reduce synchronization overhead, we notice that the convolution of $s(t)$ and $x(t)$ is,

$$(x * s)[n] \stackrel{\text{def}}{=} \sum_{k=-\infty}^{\infty} x[k] s[n-k] \tag{8}$$

Let $g[k] = s^*[-k]$, we have,

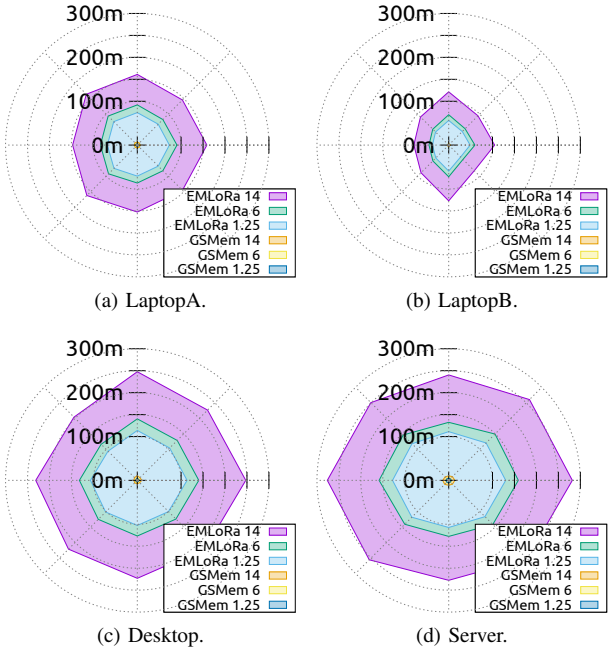
$$(x * g)[n] \stackrel{\text{def}}{=} \sum_{k=-\infty}^{\infty} x[k] s^*[n+k] \stackrel{\text{def}}{=} (x \star s)[n] \tag{9}$$

As a result, we can bypass exhaustive cross-correlation by computing Eqn. 9 and then taking FFT, which effectively reduces signal processing overhead to $\mathsf{O}(n \log n)$.

VI. EMLoRa Performance

This section evaluates the performance of EMLoRa. We implement EMLoRa's receiver using BladeRF [19] – an inexpensive and portable software radio equipped with a 2.2 dBi whip antenna. We plant the malware transmitter into four devices of different configurations, including two laptops, one desktop, and one server, as listed in Table II. When transmitting covert signals, EMLoRa's malware exploits all available memory channels of the infiltrated system. The payload size of each packet is set to 84 bits.

We compare EMLoRa with GSMem [2] – an EM covert channel that modulates memory EMR using binary on-off keying. For a fair comparison, we let EMLoRa and GSMem



(a) LaptopA. (b) LaptopB.

(c) Desktop. (d) Server.

Fig. 10: Indoor communication range of EMLoRa.

transmit at the same bit rate in all experiments. We note that the performance of GSMem reported in this section is worse than that in [2]. This is because we do not employ the high-gain directional antenna used in [2].

Communication Range We first evaluate the communication range of EMLoRa when receiving from different directions. We define communication range as the maximum distance from which at least 90% packets can be correctly received. Fig. 10 compares the communication ranges of EMLoRa and GSMem in indoor environments at different bit rates. We first measure received signal power in a large office room where the longest line-of-sight path is about 15m. We then fit measurement results in to a path loss model to estimate the maximum communication range. We observe that receiver direction has a non-negligible impact on communication range. As an example, when receiving signals at 1.25 bps from the left side of Laptop-B, EMLoRa achieves a communication range of 130 m. In comparison, the range is only 70 m when receiving from the back side. This should be attributed to the shape and material of the shield cases of particular computers. We observe that the server has the longest communication range, which achieves 250 m when receiving from the front side. We also observe that EMLoRa significantly extends communication range when compared with GSMem under the same bit rate. Specifically, at a bit rate of 1.25 bps, the range improvement ratios are 23.2x, 22.7x, 33.4x, and 21.7x on desktop, laptop-A, laptop-B, and server, respectively.

We then evaluate the range of EMLoRa in an outdoor environment. We place the receiver in the front side of the transmitter, and then measure its packet error rate while moving the receiver away until the error rate exceeds 10%. As shown in Table III, the maximum outdoor range of EMLoRa

TABLE III: Outdoor communication range of EMLoRa.

Bit Rate	EMLoRa			GSMem		
	DDR3	DDR4	LPDDR3	DDR3	DDR4	LPDDR3
14bps	72.5m	51m	40m	3.5m	2.75m	1.75m
6bps	86m	60m	48m	4.25m	3.75m	2.5m
1.25bps	137m	95m	73.5m	8m	6.25m	4m

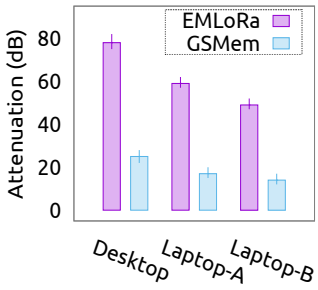


Fig. 11: The maximum attenuation resistance level.

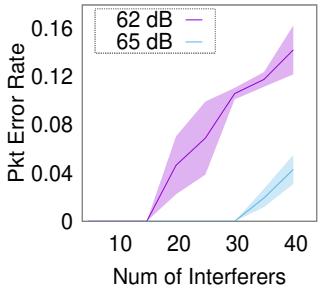


Fig. 12: PER v.s. the number of interfering devices.

is between 40 m to 72.5 m when transmitting at 14 bps. In comparison, the range of GSMem at the same bit rate is only 1.75 m to 3.5 m. By using a lower chirp rate of 1.25 bps, the range of EMLoRa can be further extended to 73.5 m to 137 m. We find that the performance of EMLoRa in outdoor is slightly worse than that observed in indoor. This is because, in indoor environments, signals reflected from walls will be combined at the receiver, which helps EMLoRa boost SNR.

Attenuation resistance. We next evaluate the performance of EMLoRa under attenuation. We note that accurate control of EMR attenuation is difficult because we cannot connect an attenuator to the 'antenna' of the emitter, i.e., the memory bus. To address this issue, we emulate a given level of attenuation by mixing an equivalent amount of white noise to the EMR signal received in close proximity to the transmitter.

Fig. 11 compares EMLoRa and GSMem in terms of the maximum attenuation resistance, which is defined as the maximum attenuation level at which a 90% packet delivery rate can be maintained. The bit rate is set to 14 bps in this experiment. As the figure shows, the maximum attenuation endurance differ across the desktop and the two laptops, which is because of their different EMR power and shielding materials. We observe that EMLoRa outperforms GSMem by 35 dB to 53 dB. In particular, when receiving from the desktop, EMLoRa can resist up to 78 dB attenuation, which is 22 dB from the 100 dB bar set by the TEMPEST. We note that the performance of EMLoRa can be further improved by using specialized radio accessories such as high-gain LNA and directional antenna, which will provide an additional gain of 22 to 70 dB, thus posing a serious threat to the TEMPEST.

Impact of ambient Interference We next stidu the performance of EMLoRa in a crowded environment where devices having the same memory frequency introduce EM interference. We employ Laptop-A as the transmitter and deploy it in a library computer room that has 40 identical desktops, all having the same memory frequency as Laptop-A. We then turn on desktops one by one and let them actively write their

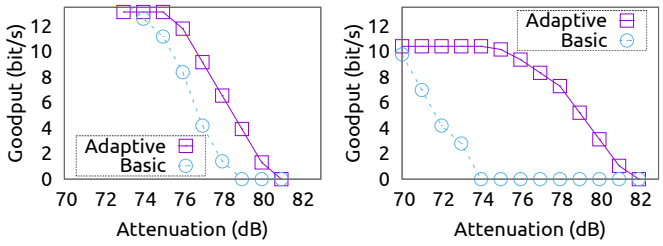


Fig. 13: The impact of CPU contention on EMLoRa (left:video playing, right:coin mining).

memories to emit strong interfering EMR. We then study the impact on EMLoRa receiver. Before measuring EMLoRa's chirp error rate and packet error rate, we add a certain amount of white noise equivalent to 62 dB attenuation to study the impact of EM interference on EMLoRa's attenuation resilience.

Fig. 12 plots the packet error rate of EMLoRa as the number of interfering devices increases. We observe that EMLoRa is fairly resilient to EM interference. In particular, when all 40 interfering devices are active simultaneously, the packet delivery rate of EMLoRa is consistently higher than 95%. This is because CCS is inherently resilient to narrow-band interference such as memory EMR. After dechirp, the sub-clocks of interfering memory EMRs will be spread over chirp bandwidth, thereby substantially reducing their interference with EMLoRa.

Impact of CPU contention. Finally, we study the performance of EMLoRa under CPU contention. In this experiment, the desktop that has the least number of cores in our device set is employed as the transmitter, because it is expected to suffer the most intensive CPU contention. We study two types of CPU workloads, including video streaming and Monero coin mining, where the former represents a common workload on PC, and the latter features a computation-intensive task. We observe that their average CPU load are 74% and 92% on the desktop, respectively. Fig. 13 shows the goodputs of EMLoRa as a function of attenuation when rate adaption is on and off, respectively. The goodput is calculated as the product between packet size and the number of correct packets delivered in a unit time. We observe that rate adaption effectively maintains attenuation resistance under CPU contention. Specifically, it provides about 2 dB and 5 dB gain over the basic EMLoRa for video streaming and Monero coin mining, respectively.

VII. EMLoRa Enabled Attacks

In this section, we demonstrate data exfiltration and localization attacks against air-gapped devices in three previously impossible scenarios.

A. Wide-Area Data Exfiltration

EMLoRa enables wide-area data exfiltration, which allows an attacker to circumvent security perimeter by deploying the receiver in public areas outside the perimeter. In the following, we conduct both real-world experiment and ray tracing-based emulations to demonstrate this attack.

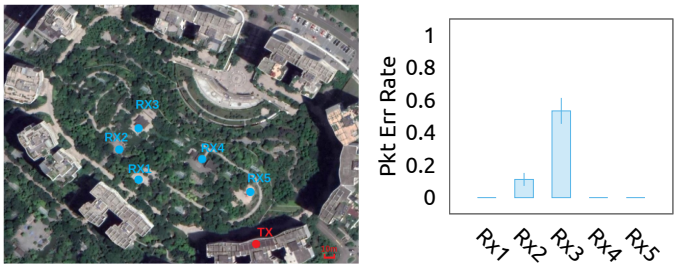
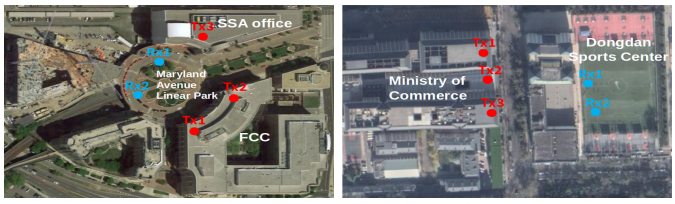
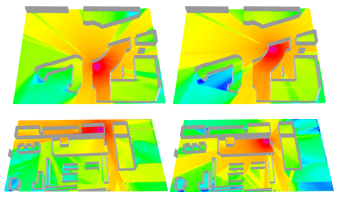


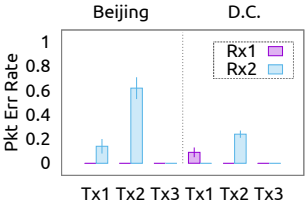
Fig. 14: The experiment setup and results of wide-area data exfiltration.



(a) Emulated scenarios (left: Washington DC; right: Beijing).



(b) Selected ray tracing results in Washington DC (top) and Beijing (bottom).



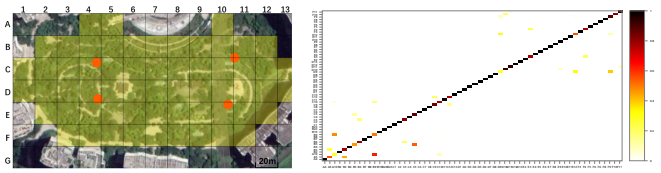
(c) Packet error rates.

Fig. 15: Emulation of wide-area data exfiltration.

Fig. 14 shows the setup of the real-world experiment. We employ a DDR3-1600 laptop as the EMLoRa transmitter, which is placed within a building with reinforced concrete walls. Five receivers are deployed at different locations 30 m to 120 m away from the building. We then let the transmitter send packets at 1.25 bps. We observe that only RX2 and RX3 (located at 117 m and 120 m away from the building, respectively) experienced packet losses. Specifically, the packet error rates at RX2 and RX3 are 11.2% and 53.4%, respectively. The poor performance of RX3 is due to the long distance and the obstruction of trees on the signal propagation path.

We then further demonstrate the wide-area data exfiltration attack by conducting ray tracing-based emulations in two scenarios. In the setup of emulation, two EMLoRa receivers are deployed in the Maryland Avenue Linear Park and Dongdan Sports Center to receive EMLoRa signals leaked from the U.S. Department of Social Security and the Ministry of Commerce of China, respectively, as shown in Fig. 15. In each scenario, we place transmitters in three selected locations within the buildings of the U.S. DSS and the MoC of China, and then deploy two receivers in public areas. We emulate the wireless communication of EMLoRa using WinProp [47] – a 3D ray tracing tool that emulates complex signal fadings caused by the blockage and reflection of ground and walls. We then measure the packet error rate at EMLoRa receivers.

As shown in Fig. 15, in the Beijing scenario, EMLoRa suffers a relatively high packet loss when receiving from TX2 at RX2. By further looking into the result, we find that the loss



(a) Experiment site. (b) Confusion matrix of locations.

Fig. 16: Outdoor device localization using EMLoRa.

is caused by the block of a building that cuts-off the line-of-sight path between TX2 and RX2. In comparison, the EMLoRa deployed around the U.S. DSS can decode all transmitters' packets except suffering low packet error rates of about 10% when receiving from TX1 at RX1, and 20% when receiving from TX2 at RX2.

B. Air-Gapped Device Localization

We then develop a device localization attack that exploits EMLoRa to transmit beacon signals, which allow an attacker to localize an air-gapped device using just a small number of EMR sensors. We assume that the attacker can deploy EMLoRa receivers in the target area in advance and conduct measurements to collect location fingerprints. To demonstrate this attack, we deploy four EMLoRa receivers in an outdoor area of about 35000 m^2, as shown in Fig. 16. The outdoor area is divided into 91 grids of 400 m^2. We then train a localization system using labeled location fingerprints, which are the vectors of normalized signal powers received by the four location sensors. We then place and move a laptop of DDR3-1600 around the area to test localization accuracy.

We observe that localization accuracy increases with the number of used location sensors. When all the four location sensors are used, localization accuracy can be improved to above 90%. We then further plot the confusion matrix between girds in Fig. 16b. We observe that, when localization error occurs, the distance between the wrong and the ground-truth grid is typically within 2 grids.

C. Faraday Cage Penetration

We next demonstrate through-wall data exfiltration from an aggresively shielded server, whose configuration is reported in Table II. We place the server in a Faraday cage made of thick Aluminum foil, which provides about 70 dB attenuation. Our experiment is conducted in an indoor area of 1600 ft^2, where we deploy the victim system in room D, as shown in Fig. 17. We then place the EMLoRa receiver in the hall and other rooms separated by concrete walls. All doors are closed during experiment.

Fig. 17 shows EMLoRa's goodputs measured at different locations in the presence/absence of the Faraday cage. We observe that, without the shielding of Faraday cage, leaked data can be received at all locations except A, which demonstrates as a dead spot. The goodput varies between 11.7 bps to 5.2 bps across locations, allowing the attacker to receive a 256-bit encryption key within 21.8 to 49.2 s. When the victim system is enclosed in the Faraday cage, location B and I turn to new dead spots, and the goodputs at location C, G, H and M drop substantially. Despite that, EMLoRa maintains a goodput higher than 8.6 bps at all other locations.

While we focus on Faraday cage made of Aluminum foil in this experiment, we note that our results can be generalized to other shielding settings. No matter what materials are used to implement the Faraday cage, the effectiveness of shielding can be quantified by the attenuation level in dB, which we have evaluated in detail in Fig. 11.

VIII. EMLoRa Detector

In this section, we explore the feasibility of uncovering EMLoRa's signals using energy- and CNN-based detectors.

Energy-based detector. Due to the permutation of EM chirp signals, the adversarial detector cannot dechirp EMLoRa's signals. Instead, the energy-based detector monitors the signal power in the frequency band utilized by EMLoRa, and then triggers an alert whenever the signal power is higher than the noise floor. We note that this detection method has two limitations. First, it requires a prior knowledge of EMLoRa's frequency utilization. Second, it may produce false alarms because legitimate system processes may also produce memory EMR in the same frequency band. Nevertheless, studying the performance of this detector helps us understand the upper-bound of detection rate that can be achieved by an energy-based detector.

CNN-based detector. Unlike the energy-based detector, the CNN-based detector assumes no prior knowledge about EMLoRa's frequency utilization. Moreover, it aims to differentiate the memory EMRs of EMLoRa and legitimate system processes through supervised learning. We engineer the CNN-based detector by customizing the neural network developed in [31], a three-layer CNN designed to detect the EMR produced by Rowhammer. This design choice is motivated by the similarity between Rowhammer and EMLoRa, as both of them involve intensive periodic memory operation.

We train the CNN by feeding it with labeled EMR spectra produced by EMLoRa and legitimate system processes. We obtain such spectra by performing FFT on signal windows of 0.05s, and then fold obtained spectra to fuse memory EMRs dispersed by spread spectrum clocking. To further suppress noise, we combine every 20 spectra into a training sample. Fig. 18 shows four examples of training spectrum measured for EMLoRa, video player, and Web browser, and Ubuntu Bionic Beaver. We find that the training typically converges within 10 rounds. To test the robustness of the detector against attenuation, we add white noise to each testing spectrum, and tune the amount of noise to study the impact on detection rate.

Evaluation. As shown in Fig. 19, we find that the energy- and the CNN-based detectors can accurately identify EMLoRa's signals only when the attenuation level is lower than 12 and 18 dB, respectively. However, the performance of both detectors degrades quickly as the level of attenuation increases. In particular, the performance of the CNN-based detector drops faster than that of the energy-based detector because it relies on fine-grained spectrum features. Even for the energy-based

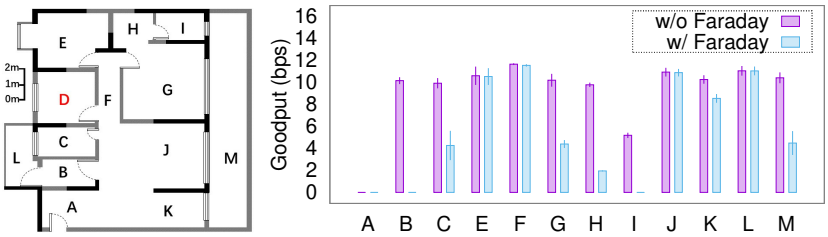


Fig. 17: Faraday cage penetration using EMLoRa.

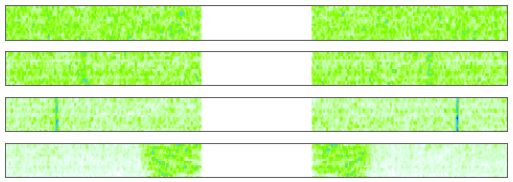


Fig. 18: From top to bottom: frequency spectrum features of Ubuntu Bionic Beaver, video player, Google Chrome, and EMLoRa.

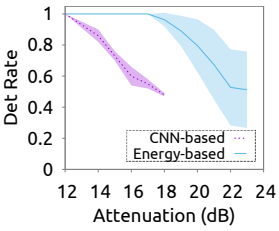


Fig. 19: Performance of energy- and CNN-based detector.

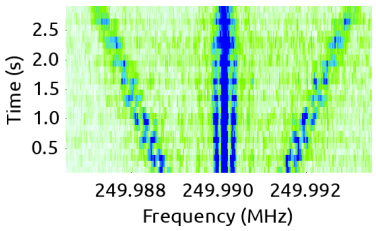


Fig. 20: Mirror chirp images generated by modulating the clock EMR of USB.

detector, its detection rate drops to below 50% when attenuation exceeds 22 dB. In comparison, EMLoRa can survive up to 78 dB attenuation, which allows it to gain a significant range advantage. These results call for further research on the countermeasure against spread spectrum-based EM covert channels.

IX. Discussion

Calibration of modulation parameters. In the current prototype of EMLoRa, modulation parameters (e.g., chirp bandwidth and bit rates) are determined based on a small set of devices, including a desktop, two laptops, and a server, which use DDR3, DDR4, and LPDDR of different memory clock frequencies. To ensure the reliability of covert communication, the parameters are conservatively calibrated based on the LPDDR3 laptop that has the weakest EMR, therefore may result in performance loss when applied on devices that have stronger EMR. We note that if the configuration of the device-to-infiltrate can be known through reconnaissance, device-specific parameter calibration will further improve EMLoRa's performance. For example, by increasing the basic rate on devices that have stronger EMR power, EMLoRa will achieve a higher throughput. Fine-grained parameter calibration on a larger device set is left for our future work.

Comparison with other spread spectrum schemes. Among widely used spread spectrum techniques, frequency-hopping spread spectrum (FHSS) and time-hopping spread spectrum (THSS) are primarily used for anti-jamming, which differ from EMLoRa's design goal of achieving attenuation resilience. Direct-sequence spread spectrum (DSSS) is most relevant to CSS as both of them are robust against noise. However, compared with CSS, a key disadvantage of DSSS is that it requires a strong synchronization between transmitter and receiver [48], and therefore is typically outperformed by CSS when transmitting power is low.

Generalization to other EM sources. Although we focus on memory EMR in this paper, the method of EMLoRa can be generalized to other EM sources. To demonstrate this, we modify the malware transmitter of USBee [3] to modulate the clock EMR of USB bus using CSS. To shape EM chirps, we manipulate the bit flow written into USB to control the frequency of EMR. As shown in Fig. 20, the modulated spectrum demonstrates mirror chirp images associated with clock component, which is identical to the pattern shown in Fig. 4 and thus can be decoded using the receiver developed in section V-C.

Jamming EMLoRa signals. Beyond detecting EMLoRa signals, a natural countermeasure against EMLoRa is to jam the frequency band around the memory clock frequency to degrade the SINR at the EMLoRa receiver. However, we note that the memory EMRs and their harmonic components typically leak into licensed bands. This means that the countermeasure have to jam licensed bands in order to completely block EMLoRa signals, which will not only violate spectrum regulation, but also interfere with legitimate wireless devices. As a result, naive jamming-based countermeasure is inapplicable. Considering EMLoRa's super resilience against attenuation, developing effective countermeasures will be an important direction of future work.

X. Conclusion

This paper leverages LoRa to revisit EM covert channel attacks. We present EMLoRa – the first EM covert channel attack that is super resilient to attenuation. Experiment results show that EMLoRa significantly extends communication range and can penetrate aggressive shielding previously considered as sufficient to ensure emission security. By achieving this, EMLoRa enables previously impossible attacks against air-gapped devices, and gains a significant range advantage over energy- and CNN-based covert signal detectors. These results call for further research on the countermeasure against spread spectrum-based EM covert channels.

Acknowledgment

We are grateful to anonymous reviewers and our shepherd, Aanjhan Ranganathan, for their insightful comments. This research was supported, in part, by funds from BvTech S.p.A. and the members of the Cybersecurity at MIT Sloan (CAMS) consortium (https://cams.mit.edu)

REFERENCES

[1] M. G. Kuhn and R. J. Anderson, "Soft tempest: Hidden data transmission using electromagnetic emanations," in International Workshop on Information Hiding, 1998.

[2] M. Guri, A. Kachlon, O. Hasson, G. Kedma, Y. Mirsky, and Y. Elovici, "Gsmem: Data exfiltration from air-gapped computers over gsm frequencies," in USENIX Security Symposium, 2015.

[3] M. Guri, M. Monitz, and Y. Elovici, "Usbee: Air-gap covert-channel via electromagnetic emission from usb," in Annual Conference on Privacy, Security and Trust (PST), 2016.

[4] N. Sehatbakhsh, B. B. Yilmaz, A. Zajic, and M. Prvulovic, "A new side-channel vulnerability on modern computers by exploiting electromagnetic emanations from the power management unit," in International Symposium on High-Performance Computer Architecture (HPCA), 2020.

[5] Z. Zhan, Z. Zhang, and X. Koutsoukos, "Bitjabber: The world's fastest electromagnetic covert channel," in International Symposium on Hardware Oriented Security and Trust (HOST), 2020.

[6] M. Guri, G. Kedma, A. Kachlon, and Y. Elovici, "Airhopper: Bridging the air-gap between isolated networks and mobile phones using radio frequencies," in International Conference on Malicious and Unwanted Software (MALWARE), 2014.

[7] R. Callan, A. Zajic, and M. Prvulovic, "A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events," in IEEE/ACM International Symposium on Microarchitecture (Micro), 2014.

[8] B. W. Lampson, "A note on the confinement problem," Communications of the ACM, 1973.

[9] P. Peng, P. Ning, and D. S. Reeves, "On the secrecy of timing-based active watermarking trace-back techniques," in IEEE Symposium on Security and Privacy (S&P), 2006.

[10] S. Cabuk, C. E. Brodley, and C. Shields, "Ip covert timing channels: design and detection," in ACM SIGSAC Conference on Computer and Communications Security (CCS), 2004.

[11] K. S. Lee, H. Wang, and H. Weatherspoon, "Phy covert channels: Can you see the idles?" in USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2014.

[12] G. Shah, A. Molina, M. Blaze et al., "Keyboards and covert channels." in USENIX Security Symposium, 2006.

[13] A. Zajic and M. Prvulovic, "Experimental demonstration of electromagnetic information leakage from modern processor-memory systems," IEEE Transactions on Electromagnetic Compatibility, 2014.

[14] U. A. Force, "Emission security," http://cryptome.org/dodi/2013/afssi-7700.pdf, 2009.

[15] ——, "Product delivery order requirements package checklist," http://netcents.af.mil/shared/media/document/AFD-140107-011.pdf/, 2014.

[16] N. I. Assurance, "Tempest equipment selection process," http://www.ia.nato.int/niapc/tempest/certification-scheme/, 1981.

[17] L. Alliance, "Lora," https://www.lora-alliance.org/, 2016.

[18] "Lora modulation basics," http://www.semtech.com/images/datasheet/an1200.22.pdf, 2016.

[19] "Bladerf," https://ww.nuand.com.

[20] M. Guri, B. Zadov, and Y. Elovici, "Odini: Escaping sensitive data from faraday-caged, air-gapped computers via magnetic fields," IEEE Transactions on Information Forensics and Security, 2019.

[21] M. Guri, M. Monitz, Y. Mirski, and Y. Elovici, "Bitwhisper: Covert signaling channel between air-gapped computers using thermal manipulations," in IEEE 28th Computer Security Foundations Symposium, 2015.

[22] Z. Yang, Q. Huang, and Q. Zhang, "Nicscatter: Backscatter as a covert channel in mobile devices," in ACM International Conference on Mobile Computing and Networking (MobiCom), 2017.

[23] M. Hanspach and M. Goetz, "On covert acoustical mesh networks in air," arXiv preprint arXiv:1406.1213, 2014.

[24] M. Alam, H. A. Khan, M. Dey, N. Sinha, R. Callan, A. Zajic, and M. Prvulovic, "One&done: A single-decryption em-based attack on openssl's constant-time blinded rsa," in USENIX Security Symposium, 2018.

[25] G. Camurati, S. Poeplau, M. Muench, T. Hayes, and A. Francillon, "Screaming channels: When electromagnetic side channels meet radio transceivers," in ACM SIGSAC Conference on Computer and Communications Security (CCS), 2018.

[26] W. Van Eck, "Electromagnetic radiation from video display units: An eavesdropping risk?" Computers Security, 1985.

[27] D. Agrawal, B. Archambeault, J. R. Rao, and P. Rohatgi, "The em side—channel(s)," in International Workshop on Cryptographic Hardware and Embedded Systems (CHES), 2002.

[28] N. Sehatbakhsh, A. Nazari, H. Khan, A. Zajic, and M. Prvulovic, "Emma: Hardware/software attestation framework for embedded systems using electromagnetic signals," in IEEE/ACM International Symposium on Microarchitecture (Micro), 2019.

[29] N. Sehatbakhsh, A. Nazari, A. Zajic, and M. Prvulovic, "Spectral profiling: Observer-effect-free profiling by monitoring em emanations," in IEEE/ACM International Symposium on Microarchitecture (Micro), 2016.

[30] A. Nazari, N. Sehatbakhsh, M. Alam, A. Zajic, and M. Prvulovic, "Eddie: Em-based detection of deviations in program execution," in The International Symposium on Computer Architecture (ISCA), 2017.

[31] Z. Zhang, Z. Zhan, D. Balasubramanian, B. Li, P. Volgyesi, and X. Koutsoukos, "Leveraging em side-channel information to detect rowhammer attacks," in IEEE Symposium on Security and Privacy (S&P), 2020.

[32] Y. Han, S. Etigowni, H. Liu, S. Zonouz, and A. Petropulu, "Watch me, but don't touch me! contactless control flow monitoring via electromagnetic emanations," in ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017.

[33] L. Batina, S. Bhasin, D. Jap, and S. Picek, "Csi nn: Reverse engineering of neural network architectures through electromagnetic side channel," in USENIX Security Symposium, 2019.

[34] C. Shen and J. Huang, "When fish takes bite: Detecting wireless eavesdroppers by stimulating memory emr," in USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2021.

[35] R. Eletreby, D. Zhang, S. Kumar, and O. Yagan, "Empowering low-power wide area networks in urban settings," in Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM), 2017.

[36] X. Xia, Y. Zheng, and T. Gu, "Ftrack: parallel decoding for lora transmissions," in ACM Conference on Embedded Networked Sensor Systems (SenSys), 2019.

[37] A. Dongare, R. Narayanan, A. Gadre, A. Luong, A. Balanuta, S. Kumar, B. Iannucci, and A. Rowe, "Charm: exploiting geographical diversity through coherent combining in low-power wide-area networks," in The International Conference on Information Processing in Sensor Networks (IPSN), 2018.

[38] V. Talla, M. Hessar, B. Kellogg, A. Najafi, J. R. Smith, and S. Gollakota, "Lora backscatter: Enabling the vision of ubiquitous connectivity," CoRR, 2017.

[39] Y. Peng, L. Shangguan, Y. Hu, Y. Qian, X. Lin, X. Chen, D. Fang, and K. Jamieson, "Plora: A passive long-range data network from ambient lora transmissions," in Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM), 2018.

[40] F. Salahdine and N. Kaabouch, "Social engineering attacks: A survey," Future Internet, 2019.

[41] P. Kocher, J. Horn, A. Fogh, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher et al., "Spectre attacks: Exploiting speculative execution," in IEEE Symposium on Security and Privacy (S&P), 2019.

[42] M. Lipp, M. Schwarz, D. Gruss, T. Prescher, W. Haas, A. Fogh, J. Horn, S. Mangard, P. Kocher, D. Genkin et al., "Meltdown: Reading kernel memory from user space," in USENIX Security Symposium, 2018.

[43] R. Zhou, Y. Xiong, G. Xing, L. Sun, and J. Ma, "Zifi: wireless lan discovery via zigbee interference signatures," in ACM International Conference on Mobile Computing and Networking (MobiCom), 2010.

[44] B. Boashash, "Time-frequency signal analysis and processing: A comprehensive reference," 2003.

[45] R. Lovelace, J. Sutton, and E. Salpeter, "Digital search methods for pulsars," Nature, 1969.

[46] D. H. Staelin, "Fast folding algorithm for detection of periodic pulse trains," Proceedings of the IEEE, 1969.

[47] "Winprop," https://altairhyperworks.com/product/FEKO/WinProp-Propagation-Modeling.

[48] M. J. Abbas, M. Awais, and A. U. Haq, "Comparative analysis of wideband communication techniques: Chirp spread spectrum and direct sequence spread spectrum," in 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2018.