Evaluating Physical-Layer BLE Location Tracking Attacks on Mobile Devices

Hadi Givehchian*, Nishant Bhaskar*, Eliana Rodriguez Herrera, Héctor Rodrigo López Soto,
Christian Dameff, Dinesh Bharadia, and Aaron Schulman

UC San Diego

2022 IEEE Symposium on Security and Privacy (SP) | 978-1-6654-1316-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/SP46214.2022.9833758

***Abstract*—Mobile devices increasingly function as wireless tracking beacons. Using the Bluetooth Low Energy (BLE) protocol, mobile devices such as smartphones and smartwatches continuously transmit beacons to inform passive listeners about device locations for applications such as digital contact tracing for COVID-19, and even finding lost devices. These applications use cryptographic anonymity that limit an adversary's ability to use these beacons to stalk a user. However, attackers can bypass these defenses by fingerprinting the unique physical-layer imperfections in the transmissions of specific devices.**

We empirically demonstrate that there are several key challenges that can limit an attacker's ability to find a stable physical layer identifier to uniquely identify mobile devices using BLE, including variations in the hardware design of BLE chipsets, transmission power levels, differences in thermal conditions, and limitations of inexpensive radios that can be widely deployed to capture raw physical-layer signals. We evaluated how much each of these factors limits accurate fingerprinting in a large-scale field study of hundreds of uncontrolled BLE devices, revealing that physical-layer identification is a viable, although sometimes unreliable, way for an attacker to track mobile devices.

I. Introduction

The mobile devices we carry every day, such as smartphones and smartwatches, increasingly function as wireless tracking beacons. These devices continuously transmit short-range wireless messages using the Bluetooth Low Energy (BLE) protocol. These beacons are used to indicate proximity to any passive receiver within range. Popular examples of such beacons include the COVID-19 electronic contact tracing provided on Apple and Google Smartphones [10] as well as Apple's intrinsic Continuity protocol, used for automated device hand-off and other proximity features [1].

However, by their nature, BLE wireless tracking beacons have the potential to introduce significant privacy risks. For example, an adversary might stalk a user by placing BLE receivers near locations they might visit and then record the presence of the user's beacons [3], [37]. To address these issues, common BLE proximity applications cryptographically anonymize and periodically rotate the identity of a mobile device in their beacons. For instance, BLE devices periodically re-encrypt their MAC address, while still allowing trusted devices to determine if these addresses match the device's true MAC address [6]. Similarly, COVID-19 contact tracing applications regularly rotate identifiers to ensure that receivers cannot link beacons from the same device over time [2].

While these mechanisms can foreclose the use of beacon content as a stable identifier, attackers can bypass these countermeasures by fingerprinting the device at a lower layer. Specifically, prior work has demonstrated that wireless transmitters have imperfections introduced in manufacturing that produce a unique physical-layer fingerprint for that device (e.g., Carrier Frequency Offset and I/Q Offset). Physical-layer fingerprints can reliably differentiate many kinds of wireless chipsets [14], [9], [18], [35], [29], [21], [28], [8], including a recent attempt to distinguish 10,000 WiFi [19] chipsets.

To the best of our knowledge, no prior work has evaluated the practicality of such physical-layer identification attacks in a real-world environment. Indeed, prior to BLE tracking beacons, no mobile device wireless protocol transmitted frequently enough—especially when idle—to make such an attack feasible. Additionally, there is no existing BLE fingerprinting tool that can measure the physical-layer imperfections in BLE transmissions (i.e., CFO and I/Q offset) accurately. Prior techniques for fingerprinting either provide low precision fingerprints because they use short duration (e.g., transient) signal features, or provide high precision fingerprints but require long duration signal features which exist only in protocols like WiFi but not in BLE. Our first contribution is a tool that uses a novel method to recover these imperfections by iteratively adding imperfections to a re-encoded clean copy of a received packet, until they match the imperfections of the received packet over the air (Section III).

Our next contribution is an evaluation of how practical it is for an attacker to track BLE-beaconing devices using their RF fingerprint. Namely, using lab-bench experiments, we identify four primary challenges to identifying BLE devices in the field: (1) BLE devices have a variety of chipsets that have different hardware implementations, (2) applications can configure the BLE transmit power level, resulting in some devices having lower SNR BLE transmissions, (3) the temperature range that mobile devices encounter in the field can introduce significant changes to physical-layer impairments, and (4) the low-cost receivers that an attacker can use in the wild for RF fingerprinting are not significantly less accurate than the tools used in prior studies [9].

Our final contribution is a set of field experiments to evaluate how significantly these challenges diminish an attacker's ability to identify mobile devices in the field. We leverage the fact that BLE tracking beacons are already used on many mobile devices to perform an uncontrolled field study where we evaluate the feasibility of tracking BLE devices when they are operating in public spaces where there are hundreds of other nearby devices. To the best of our knowledge, our work he first to evaluate the feasibility of an RF fingerprinting

DOI 10.1109/SP46214.2022.00030

attack in real-world scenarios.

We show that even when there are hundreds of devices we encountered in the field, it is still feasible to track a specific mobile device by its physical-layer fingerprint. However, we also observe that certain devices have similar fingerprints to others, and temperature variations can change a device's metrics. Both of these issues can lead to significant misidentification rates. In summary, we find that physical layer tracking of BLE devices is indeed feasible, but it is only reliable under limited conditions, and for specific devices with extremely unique fingerprints, and when the target device has a relatively stable temperature. The dataset and code that we used to perform this evaluation can be found at:

https://github.com/ucsdsysnet/blephytracking.git

II. BLE Device Tracking Threat Model

In this section we describe the threat model of location privacy attacks on BLE-enabled mobile devices. Then, we demonstrate how location privacy attacks are a significant threat today because popular mobile devices continuously, and frequently, transmit BLE advertisements.

A. Threat model: Passively fingerprinting BLE transmissions

An attacker wants to detect when their target—a user with a mobile device—is present at a specific location (e.g., a room in a building). To do so, first the attacker must isolate the target to *capture a fingerprint* of its wireless transmissions. Then it must find features that uniquely identify the target, namely the unique physical-layer features of the device's BLE transmitter hardware. Then, the attacker sets up a receiver in the location where they want to see if the transmitter is there and *passively sniffs* for the target's BLE transmissions. They will know when the target device is near the receiver when it captures one or more packets that matches the target's physical layer fingerprint. The more frequently the BLE device transmits, the more likely the attacker is to receive a transmission if a user passes by. Also, the more accurate the fingerprinting technique is, the better the attacker can differentiate the target from other nearby devices. Fingerprinting bypasses MAC address randomization [7], [26], BLE's existing defense against tracking.

To perform a physical-layer fingerprinting attack, the attacker must be equipped with a Software Defined Radio sniffer: a radio receiver capable of recording raw I/Q radio signals. Although, as we show in Section IV-D, it is sufficient to use a modest hobbyist-level SDR (∼$150).

B. Extent of threat: Popular mobile devices are vulnerable

Increasingly, mobile devices are adding BLE beacons to provide new features. Most notably, during the COVID-19 pandemic, governments have installed software on iPhones and Android phones to send constant BLE advertisements for digital contact tracing: devices listen for nearby transmissions to determine if and for how long another device was nearby. Also, Apple and Microsoft operating systems have recently added BLE beaconing to their devices for inter-device communication features: lost device tracking, and seamless user switching between devices (e.g., Apple's Continuity Protocol, Microsoft's Universal Windows Platform) [5]. Therefore, BLE beacons are now common on many mobile platforms, including: phones, laptops, and smartwatches.

Product	*OS*	*# of adverts/minute*
iPhone 10	iOS	872
Thinkpad X1 Carbon	Windows	864
MacBook Pro 2016	OSX	576
Apple Watch 4	iOS	598
Google Pixel 5*	Android	510
Bose QC35	Unknown	77

*Only beacons with COVID-19 contact tracing enabled.

TABLE I: BLE beaconing behavior of popular mobile devices.

Fingerprinting and tracking a BLE device requires the device to act like a tracking beacon: it must transmit continuously and frequently. We observed the BLE behavior of popular devices to determine if they transmit continuously, and how frequently they transmit if they do. Specifically, we isolated six popular devices in a Faraday cage—ensuring they were the source of the transmissions—and we used an SDR sniffer to collect all BLE advertisements (i.e., BLE beacons) transmitted on any of the three advertising channels.

Mobile devices send BLE beacons continuously: We observed continuous BLE beaconing from all the six mobile devices shown in Table I. Even when all of these mobile devices have their screens off (e.g., they are in their user's pocket) they all continuously transmit BLE beacons. Indeed, this is a feature that is necessary for the proper function of the BLE-based applications on these devices (e.g., contact tracing). Continuous beaconing is a significant new threat compared to the behavior of other protocols on mobile devices that only transmit intermittently (e.g., periodic WiFi scanning).

Mobile devices send hundreds of BLE beacons per minute: Table I also shows the average number of BLE beacons (i.e., BLE advertisements) we observed per minute from each device. We observe that all of these devices transmit frequently—hundreds of packets per minute—even when the device is otherwise idle (e.g., screen off). Transmitting hundreds of advertisements per minute makes it feasible to produce a physical-layer fingerprint quickly: even if the device is in range of the sniffer for a few seconds (Section V).

III. BLE Tracking Toolkit

In this section, we describe a toolkit to evaluate if an attacker can perform a BLE tracking attack based on physical-layer fingerprinting. First, we describe how BLE produces a similar physical-layer fingerprint to other wireless protocols. Then, we describe the unique challenges of fingerprinting BLE transmissions, and therefore why existing fingerprinting techniques do not work on BLE transmissions. Next, we describe a new approach to fingerprinting BLE devices using a novel joint imperfection estimation technique. Finally, we describe how an attacker can use a sniffer to track a specific device by detecting if its fingerprint matches one of the BLE devices nearby the sniffer.

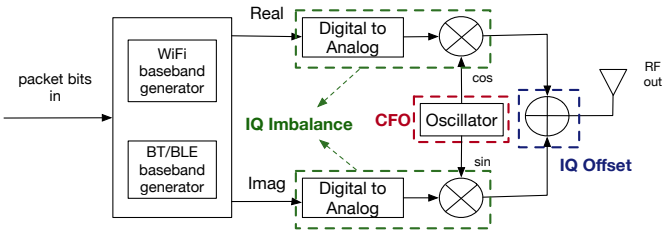

Fig. 1: Architecture of WiFi/BLE combo chipsets

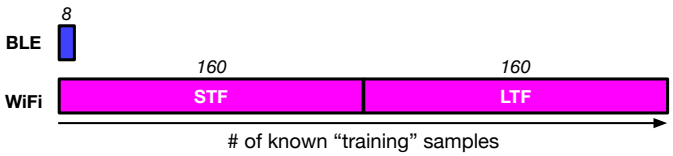

Fig. 2: Length of known samples in BLE and WiFi packets.

A. BLE has WiFi-like signal imperfections

Physical layer fingerprinting relies on each BLE radio having unique hardware imperfections introduced by manufacturing variations in its transmitter chain. Different types of imperfections are introduced by different transmitter architectures. Therefore, we need to understand the architecture of a typical BLE chipset to understand what imperfections we need to fingerprint.

We investigated the architecture of several BLE chipsets used in popular mobile devices, and found that WiFi and BLE are often integrated into the same device. Also, internally, they share the same 2.4 GHz I/Q frontend. (Figure 1). This architecture, known as a "combo chipset" is desirable for mobile devices because it reduces the device's overall size and power consumption, and it serves as a point to synchronize both protocols' 2.4 GHz transmissions, so they do not interfere with each other.

A consequence of this hardware design choice is that BLE transmissions contain the same hardware imperfections as WiFi. The imperfections are introduced by the shared I/Q frontend of the chipset (Figure 1). They result in two measurable metrics in BLE and WiFi transmissions: *Carrier Frequency Offset (CFO)* and *I/Q imperfections*, specifically: I/Q offset and I/Q imbalance. Prior work demonstrated that these metrics are sufficient to uniquely fingerprint WiFi devices [9]. The following describes how each of these metrics are calculated and how they result from manufacturing variations:

CFO: It is an offset in the carrier frequency generated by the RF frontend's local oscillator. The carrier frequency is ideally exactly the center frequency of the channel in use. However, imperfections in the radio's local oscillator, a crystal oscillator, yields a unique CFO added to every transmission. Crystals cut in different ways yield different tolerances in how much an individual crystal's frequency can deviate from the true value it was to produce for. This imperfection manifests as CFO because the local oscillator is mixed with the baseband signal (e.g., WiFi or BLE) in the RF frontend, so it can be transmitted; thereby, carrying the crystal's imperfection as a feature in the transmission.

I/Q imperfections: These are a result of the following two phenomena. *I/Q Offset* is created by two different imperfections in the RF frontend: (1) the carrier frequency signal leaking through the mixer into the transmitted signal, or (2) the baseband signal having a DC offset. I/Q offset results in a fixed complex term added to each received I/Q sample (i.e., a shift in the center of the constellation). *I/Q Imbalance* occurs because of a mismatch between the parallel analog components of RF chain in I (in-phase) and Q (quadrature) signal paths. This results in asymmetry in the phase and amplitude of received I/Q samples.

B. BLE is more difficult to fingerprint than WiFi

Measuring transmitter imperfections is significantly more challenging for BLE transmissions than it is for WiFi transmissions. The problem is, BLE signals are Gaussian Frequency Shift Keying (GFSK) waveforms that do not require accurately correcting CFO and I/Q imperfections for decoding. Conversely, WiFi signals are wideband multi-carrier waveforms, therefore their decoding algorithm requires accurately correcting for CFO and I/Q imperfections for decoding.

As a result of this issue, BLE packets contain fewer known "training" samples used for measuring imperfections than WiFi (Figure 2). BLE packets have only 8 training samples, while WiFi packets have 320 training samples. BLE receivers use these symbols to implement very coarse grained CFO correction. Namely, they average the two frequencies (symmetric positive and negative frequencies are used to represent 0 and 1 symbols) in BLE's training sequence to produce a coarse grained average CFO [34]. Indeed, with only 8 samples at BLE's 1 MHz sample rate, the theoretical limit of CFO accuracy is 2 kHz assuming 3 degree phase noise, even at high SNR (40 dB).

To make matters worse, inaccurate coarse compensation of CFO significantly affects our ability to measure I/Q imperfections. Inaccurately compensating CFO will result in time-dependent phase shift distorting the I/Q constellation, making it challenging to accurately estimate I/Q imperfections.

Prior fingerprinting techniques developed for other protocols are not capable of overcoming these challenges. Therefore, prior approaches can not be used to fingerprint BLE [39], [9], [23], [17], [35], [27], [31], [16].

Another approach to fingerprinting BLE transmissions would be to use neural networks. Although neural networks can address these challenges, we did not use them in this work because of the following limitations: (1) Neural networks make it difficult to determine the significance and distinguishably of each type of hardware imperfection (e.g., CFO, I/Q offset), (2) Neural networks also can overfit to a specific bit pattern in a packet, rather than the transmitter imperfections. This is problematic because BLE advertisements do not have a stable bit pattern: MAC addresses change every 15 minutes. (3) In our preliminary experiments with neural nets, we found they require significantly more training data than the conventional sification we describe in this work.

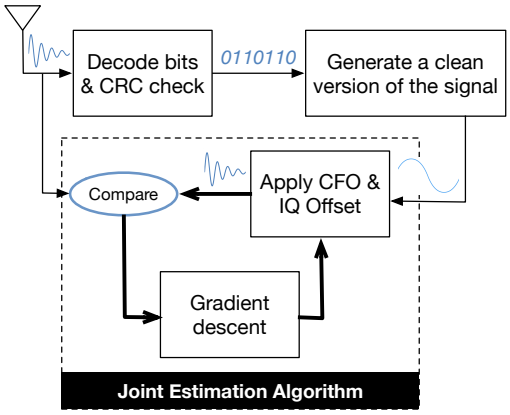

Fig. 3: Our new BLE imperfection estimation method.

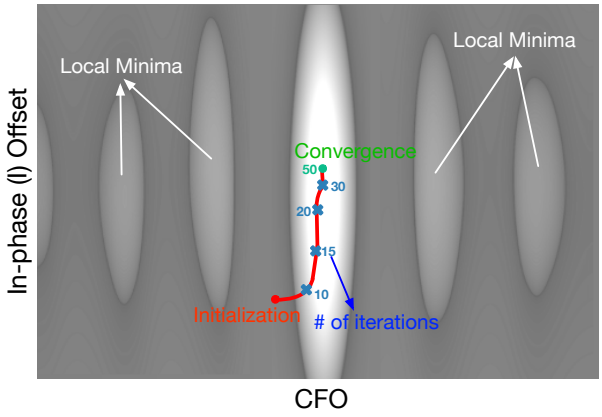

Fig. 4: Process of jointly estimating CFO and I/Q offset.

C. Accurate measurement of BLE's CFO & I/Q imperfections

The fingerprinting methodology we present in this work (Figure 3) is the first physical-layer identification method that can accurately estimate CFO and I/Q imperfections of BLE signals. To fingerprint BLE, we need to overcome the following two fundamental challenges:

Accurate estimation of CFO (and I/Q imperfections) is not feasible with BLE's short training symbol sequence (preamble). Instead of relying on the 8 known samples, we can utilize the entire packet (∼370 samples). This results in a theoretical CFO measurement precision of about 40 Hz compared to 2 kHz from BLE's coarse CFO estimation. The next question is: how can we leverage the entire decoded packet to estimate CFO and I/Q imperfections accurately?

First, we decode the entire received BLE packet (for packets with a valid CRC) and reconstruct a clean BLE waveform (Figure 3 top). Then, we leverage the clean reconstructed BLE waveform and distort it iteratively, we do so until we find CFO and I/Q imperfection estimates where the reconstructed signal matches the original received signal (Figure 3 bottom). Brute force search for the optimal CFO and I/Q imperfections requires significant computational complexity because we need to search all possible values. We use optimization techniques to make this more efficient. The primary insight of our optimization approach is as follows: estimating I/Q imperfections of a received signal depends on the estimates of the signal's CFO. Therefore, as we get closer to an accurate estimate of CFO, we reduce the search space to find accurate I/Q imperfections. We build on this insight and present a joint CFO and I/Q imperfection estimation methodology.

Jointly estimating CFO and I/Q: Let y be the captured complex baseband BLE signal (normalized by the average amplitude). In a GFSK modulated signal, ideally we have $Real\{y\} = cos(\omega(t)t)$ and $Imag\{x\} = sin(\omega(t)t)$ where $\omega(t)$ is the baseband frequency of the signal which is generated according to the GFSK modulation. However, the presence of hardware imperfections will slightly change the baseband signal. We first decode the signal to obtain the sequence of bits and then, we make $\omega(t)$ according to GFSK modulation. Let y' be the model of the imperfect signal. With the effects of CFO and I/Q imperfections, the baseband signal becon

$$y'(t) = A \times \big[(1-\frac{\epsilon}{2})cos(\omega(t)t - \frac{\phi}{2}) + I + j\big((1+\frac{\epsilon}{2})sin(\omega(t)t + \frac{\phi}{2}) + Q\big)\big] \times e^{j(\phi_o + 2\pi f_o t)}$$

where f_o, ϕ_o, A, $\frac{1-\epsilon}{1+\epsilon}$, ϕ, I and Q denote CFO, phase offset, normalized amplitude of the signal, I/Q amplitude imbalance, I/Q phase imbalance, I offset and Q offset, respectively. The goal is to choose the value of these variables in such a way that $||y' - y||^2$ is minimum and as a result, y' is as close as possible to the captured signal y. Therefore, we must solve the following optimization problem:

$$min_{f_o,\phi_o,A,\epsilon,\phi,I,Q} F = ||y' - y||^2 = |Real\{y'\} - Real\{y\}|^2 + |Imag\{y'\} - Imag\{y\}|^2$$

However, this problem is not convex, and the objective function has several local minima (Figure 4). To avoid optimization techniques ending up in a local minimum, we initialize the variables properly. This increases the chance of finding the global minimum. Although theoretically we aren't guaranteed to reach the global minimum for arbitrary optimum numbers of these variables, we found in practice the initialization helps us reach the optimum value.

To initialize CFO, we start by taking the average of frequencies in the preamble. Then we compensate the initial CFO in the signal to get the signal $z = ye^{-2\pi f_o t}$. To estimate initial I/Q imperfections, we use the I/Q constellation of the GFSK signal. The I/Q constellation of an ideal GFSK signal is a circle centered at $(0, 0)$. However, I/Q imperfection will change this constellation. Specifically, I/Q offset shifts the center of the constellation, and I/Q imbalance changes the shape from a circle to a tilted ellipse. As a result, to get an initial estimation of I/Q imperfections, we fit an ellipse to the 2-dimensional points $(Imag\{z\}, Real\{z\})$ by minimizing the Least Square Error. The center of the ellipse will provide the initial I/Q offset, and initial I/Q imbalance can be obtained from the ratio of minor and major diameter and rotation angle of the ellipse.

Although these initial estimations are close to optimum, they are not accurate. As mentioned earlier, this CFO initialization ased on an 8 symbol preamble and therefore not accurate.

Moreover, an inaccurate CFO estimate will cause a time-dependent phase shift which distorts the I/Q constellation. Therefore, the initial I/Q offset and imbalance estimation will also have errors. Consequently, we employ optimization techniques to jointly estimate these imperfections using the initial estimates.

We chose gradient descent to solve the optimization problem, as it ensures that we move towards the optimal values after each step. Specifically, we use a quick form of gradient descent, Nesterov Accelerated Gradient Descent (NAG) to move from the initialization towards the optimum values of $f_o, \phi_o, A, \epsilon, \phi, I, Q$ by minimizing F in the mentioned optimization problem. NAG adaptively adjusts the parameter update at each step, so that we move faster towards the optimal value at the start but slow down as we get close to the minima.

Figure 4 demonstrates the process of how we start from the initial estimations of CFO and I/Q imperfections, then move toward the optimal values of CFO and I/Q imperfections using gradient descent (the red line), and finally converge to the accurate estimations of CFO and I/Q imperfections in a few iterations. Since this optimization problem is not convex, it is still possible we end up converging to a local optima. Therefore, if after convergence, the average of F was not less than a certain threshold (determined according to SNR), we adjust the initialization values by a fixed step and repeat the gradient descent process.

The proposed optimization based estimation ensures accurate estimation with fine granularity as it keeps moving towards the optimum with adaptive steps and removes the mutual effect of mismatch in estimating these imperfection parameters. Moreover, the objective function of the optimization is chosen as the summation of all samples across the packet, which diminishes the impact of the channel and provides fine-grained estimates of the CFO and I/Q imperfections.

Evaluating CFO estimation accuracy: To evaluate the accuracy of our new fine-grained fingerprinting algorithm compared with BLE's default coarse CFO estimation, we compute the standard deviation of CFO measured for 100 packets from each device in a set of 100 different BLE transmitters observed in the field. Figure 5 shows the CDF of the standard deviation of the CFO across all transmitters, for both techniques. We see that our fine-grained CFO estimation significantly reduces the standard deviation of CFO estimation for all devices. This reduces the within-device variance, making fingerprints easier to distinguish.

Summary: We demonstrate that it is feasible to estimate CFO and I/Q imperfections of WiFi/BLE combo chipsets accurately; even though BLE does not have the rich signal features that are present in WiFi.

D. Fingerprinting and identifying a target device

With fine-grained estimates of the CFO and I/Q imperfections in BLE transmissions, we can now execute a BLE location tracking attack. The first step in the attack is capturing BLE packets. We use an SDR to capture raw I and Q samples of nearby BLE transmissions. The captured BLE packets are processed in two stages—fingerprinting and identification

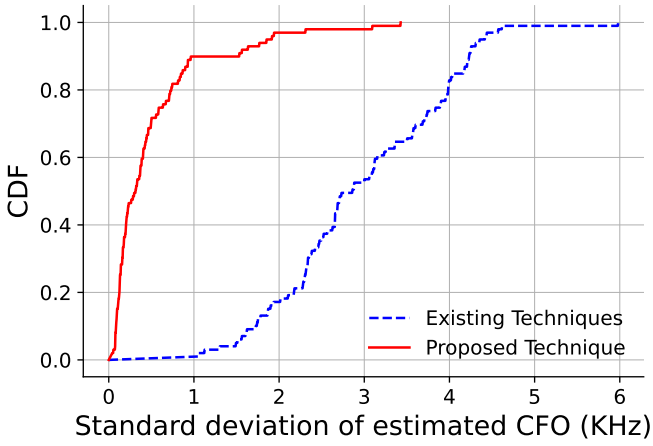

Fig. 5: Comparing the CFO estimation of existing coarse-grained techniques with our proposed technique.

the fingerprinting stage the target device is briefly isolated, and we capture a number of packets that we use for building a fingerprint for the device (i.e., training packets). The latter stage identifies if a captured BLE packet matches the fingerprint of the target device.

Fingerprinting Stage: For each packet from a device D, we extract CFO and I/Q imperfections using the algorithm described in III-C. Let $x_1, ..., x_N$ be the CFO and I/Q imperfection feature vectors for N training packets we have received from device D. We calculate the mean μ_D and covariance matrix Σ_D of $X = [x_1 \quad ... \quad x_N]$. μ_D and Σ_D together with a threshold that will be defined later is considered the profile of device D.

Identification Stage: In identification stage, we want to decide whether a packet x_t was transmitted by device D, indicating that the target device is near the SDR. To do so, we compute the Mahalanobis distance to the profile of device D:

$$distance(x_t, \mu_D, \Sigma_D) = \sqrt{(x_t - \mu_D)^T \Sigma_D^{-1} (x_t - \mu_D)}$$

This distance metric measures how close the features of the new packet are to the profile of device D. In addition to μ_D and Σ_D, we define a threshold $thresh$ as the profile of the device. Whenever $distance(x_t, \mu_D, \Sigma_D) < thresh$ for packet x_t, we identify the packet as being transmitted by the target device D. The threshold can be chosen in two ways. One way would be to choose a threshold that guarantees a certain FNR in a validation set. Another way can be to pick a threshold that minimizes $FPR^2 + FNR^2$, so that their values are balanced. In this paper, we use these two methods for selecting the threshold depending on the goal of the experiment.

Additionally, since the MAC address of every BLE device is stable for a limited duration of time, we can receive multiple packets that we know belong to the same BLE device. As a result, we can use multiple packets to identify a BLE device, reducing inter-packet noise. One way that we found most effective to use multiple packets was to first average the feature tor x for all packets from the same BLE device, and then

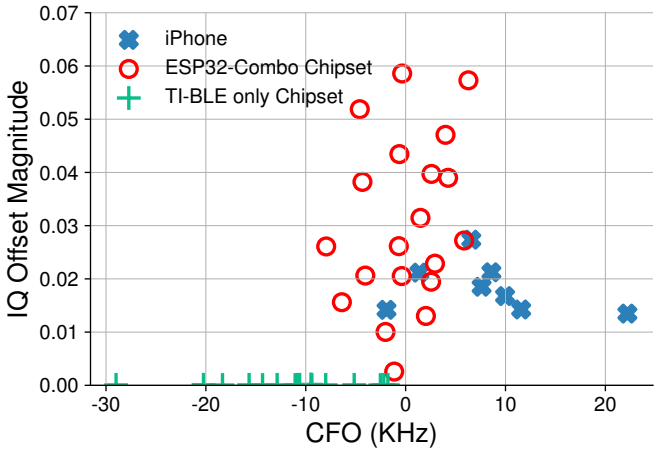

Fig. 6: Comparing the fingerprints of 48 BLE chipsets

compute the Mahalanobis distance. This averaging reduces the effect of small deviations in the imperfections across packets.

Summary: We identify a device based on the Mahalanobis distance to its previously recorded hardware imperfection fingerprint. Also, since BLE devices have temporarily stable identifiers in their packets, we can identify a device based on the average over multiple packets, increasing identification accuracy.

IV. Challenges

There are five primary challenges that limit the effectiveness of tracking BLE devices based on their physical-layer fingerprint. For each challenge, we perform controlled experiments or theoretical analysis to investigate how significantly they affect fingerprinting accuracy in practice. We found that BLE tracking attacks are likely to be feasible in practice. However, the attacker's ability to track a specific device will vary depending on several factors that are out of their control.

A. Uniqueness of BLE fingerprints

BLE transmitters must have unique imperfections if an attacker wants to differentiate their target from other nearby devices. To evaluate how similar BLE fingerprints are in practice, we compare the fingerprint of many devices across three different popular BLE chipsets. Specifically, we captured the fingerprint of eight recent iPhones with WiFi+BLE combo chipsets, 20 ESP32 WiFi+BLE microcontroller chipsets, and 20 TI CC2640 BLE-only chipsets used in low-power devices (e.g., fitness trackers). We captured 100 packets using a high-quality SDR (USRP N210) from each of these devices in a controlled environment (i.e., an RF isolation chamber). We computed the fingerprint of each device across all 100 packets using the methodology described in the previous section.

Figure 6 shows the mean of the fingerprint metrics for each of the 48 devices. We plot only the CFO and I/Q offset metrics to simplify the visualization, adding I/Q imbalance does not change the conclusions of the experiment. Overall, most of the 48 devices have unique fingerprints. However, there are a few devices that have similar fingerprints, mak them more difficult to uniquely identify. The distribution of device fingerprints also appears to be dependent on the chipset. Namely, there are striking differences in how the I/Q offset metric is distributed between different chipsets. For instance, the ESP32 devices have a much larger range of I/Q offsets than the iPhones, which may be because ESP32s are low-end chipsets compared to the high-performance WiFi+BLE combo chipsets used in iPhones.

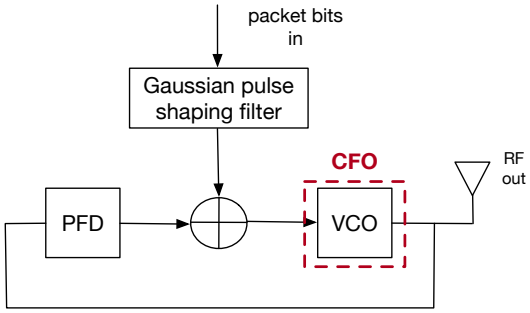

Fig. 7: TI's BLE-only transmitter. This is not an I/Q modulator.

Surprisingly, the TI BLE-only chipsets all have negligible I/Q offset. Recall in Section III, we described how unlike WiFi, BLE is not an inherently I/Q modulated protocol; therefore, the TI's BLE-only chipset may have I/Q offset because it may not use an I/Q modulator. We confirmed this suspicion by finding a technical report that describes the TI BLE chipset radio architecture: it uses a PLL-based (non-I/Q) modulator [36].

Summary: An attacker's ability to uniquely identify a target device's fingerprint depends on the BLE chipset it is using, as well as the chipsets of the other devices nearby. Distinguishing devices with the same chipset is likely more difficult than distinguishing devices with different chipsets. This may make tracking attacks difficult in practice because targets are likely to use the same popular devices (e.g., iPhone).

B. Temperature stability of BLE fingerprints

A device's BLE fingerprint must be stable to track over time across multiple locations. However, a device's CFO may drift when the temperature of the device changes. CFO is a product of imperfections in the crystal oscillator used to generate the transmitter's center frequency (e.g., 2.480 GHz), and the frequency error of a crystal oscillator has a well-defined relationship with its temperature called the "Bechmann curve". The relationship between temperature changes and I/Q imperfections is not as well understood as with CFO.

Smartphones are particularly exposed to temperature variations. Their internal temperature can significantly change due to internal components heating up (and cooling down) when activity changes, and they also experience a variety of ambient temperatures [20]. However, it is possible that smartphones do not have instability in their BLE transmissions. The impact of temperature on CFO is dependent on the cut angle and face of the crystal [12], and smartphones may use high-quality crystals that have less frequency drift due to temperature changes. Also, smartphones may use temperature compensated crystals as they may be required for high-data rate cellular
ımunication chipsets.

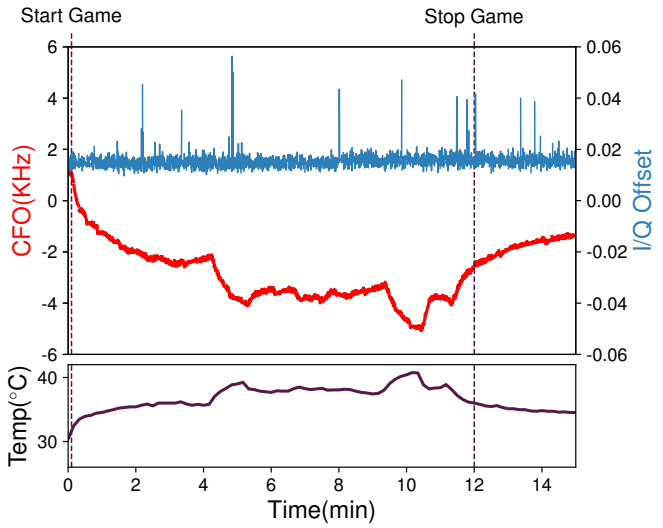

Fig. 8: Metric stability while playing a GPU-intensive game

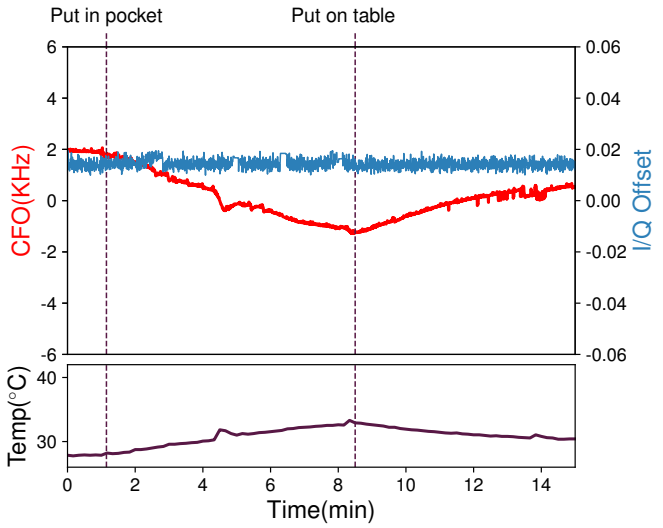

Fig. 9: Metric stability while putting the phone in a pocket

We performed controlled experiments to observe how temperature affects CFO and I/Q offset of a typical smartphone. We tested the effects of internal components changing temperature by playing a graphics-heavy game (Asphalt 9), and the effects of ambient temperature by putting an idle phone into a user's pants pocket. Our test device was a common smartphone, a Moto G6, and it was running a COVID–19 contact tracing app to generate BLE transmissions. Each test ran for 15 minutes. During the tests we captured the fingerprint metrics from each BLE packet with a USRP N210. Simultaneously, we also captured readings from all the internal temperature sensors of the device. We only present the temperature sensor data that most closely correlated with the changes in CFO, which was the Power Management Integrated Circuit's temperature sensor.

Figures 8 and 9 show the per-packet variation in CFO and IQ offset during the 15-minute tests. We do not show the variation in I/Q imbalance as it as we found it has a similar relationship to temperature as I/Q offset. For the game experiment (Figure 8), we observe that the CFO has a linear relationship to the changes in temperature. When the game begins, the CFO increases, and when the game ends, it decreases. At the peak internal temperature (+10°C above baseline), we observe a significant CFO deviation (7 kHz). For the in-pocket experiment (Figure 9), the peak change in CFO is much less than the game experiment (2 kHz). However, it is still significant enough to introduce confusion with other devices that have similar I/Q metrics (Figure 6). Finally, figures 8 and 9 both show that I/Q offset (and I/Q imbalance which is not shown) does not correlate with temperature.

Summary: Device temperature changes significantly change the CFO a smartphone, but not the I/Q imperfections. If an attacker tries to track a device when it is under heavy use, it will need to allow for significant differences in CFO from the initial fingerprint, which may result in increased confusion with other nearby devices. Also, putting an idle device in a user's pocket changes the CFO significantly enough to ca confusion as well. Ideally, an attacker would both get an initial fingerprint, and try to identify the device, in the of the most common use case for the device: idle in the user's pocket.

C. Differences in BLE transmitter power

BLE transmit power affects how far away an attacker can track a target. If some devices have lower transmit power, it is more difficult for an attacker to capture their beacons. One may assume that all similar devices (e.g., smartphones) would use similar transmit power—especially when they are running the same popular app. In particular, we would expect similar transmit power for the same contact tracing apps, where transmit power correlates with distance where the contact occurred. However, transmit power is configurable: BLE APIs on mobile devices allow applications to set their beacon transmit power to match the needs of the application.

We measured the received SNR of BLE beacons from several popular smartphones while they were running the Apple/Google COVID–19 contact tracing app. The measurement was performed with a USRP N210, and all the phones were placed at the same distance (15 feet) from the USRP. We performed this measurement on five different phones, running latest version of iOS and different versions of Android. We installed the same official California COVID–19 contact tracing app on all the devices. Then, we averaged the SNR over 100 received packets from each of the devices.

Figure 10 shows that the iPhone 8 has an SNR 10 dB higher than all other Android phones we tested. Therefore, the iPhone's BLE beacons are likely to be received considerably farther away than the other devices. Anecdotally, we observed that an iPhone's COVID–19 contact tracing beacons 7 meters farther than any of the Android devices we tested*.

Summary: There can be significant differences in BLE transmit power across devices, and even across apps running on devices. We observed that iPhones transmit COVID–19

ncluding other versions of the iPhone available at the time (e.g., Xr).

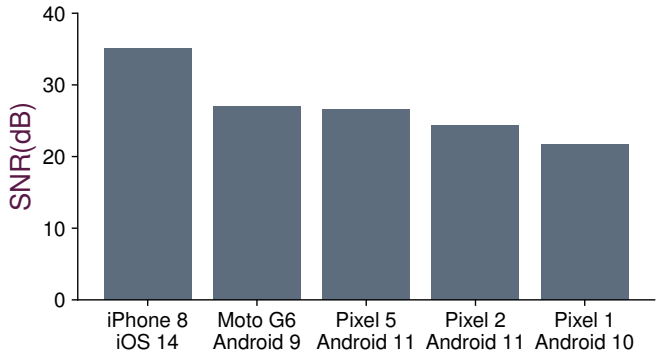

Fig. 10: SNR of COVID contact tracing beacons across devices

contact tracing beacons with significantly higher power than Android devices. Consequently, attackers may be able to track iPhones from a farther distance than Android devices.

D. Quality of an attacker's sniffer radio

Physical-layer fingerprinting attacks can require an expensive high-quality Software-Defined Radio (SDR) to execute. The problem is, an SDR's receiver chain adds signal imperfections to the transmitted signals. If the SDR's imperfections are unstable, they can make it difficult to identify a device based on its previously captured fingerprint. On the other hand, the more expensive the required SDR is, the fewer locations an attacker can deploy them to track their target.

Recently, several low-cost SDRs have become popular among hobbyists. However, the stability of their receivers' imperfections are unknown. We evaluate if one of the least expensive SDRs has sufficient imperfection stability for BLE device tracking.

We compared the fingerprinting metrics captured by a high-end SDR, USRP N210 ($3,400), and a low-end SDR, LimeSDR-Mini ($179). To make the comparison fair, we sent BLE packets from a single iPhone device to both SDRs simultaneously. We computed the average and standard deviation of our metrics to evaluate if the two devices observe the same absolute imperfections, and if they have similar metric stability. Similar to prior experiments, we captured 100 beacons to compute these distributions.

CFO: The USRP observed a mean of -4.78 kHz and a standard deviation of 102 Hz, while the Lime-SDR observed a lower mean of -8.07 kHz but with a similar standard deviation of 114 Hz. The difference is in the mean CFO is likely due to manufacturing variations in the SDR's crystal oscillators. Both radios however use a TCXO-based oscillator, therefore their CFO measurements will be stable even if the SDR's temperature changes.

I/Q metrics: A similar conclusion can be drawn about the differences between the observed I/Q metrics. The USRP observed an average I/Q offset magnitude of 0.0145 and standard deviation of 0.0017. While the Lime-SDR observed an average of 0.0203 but with a similar standard deviation 0.0030. The I/Q imbalance was surprisingly similar across both devices, with a mean amplitude of 0.991 for the USRP and 0.987 for the Lime-SDR, the corresponding standard deviations were similar too (0.0016 and 0.0021).

Summary: Attackers can use lower-cost ($179) hobbyist-grade SDRs to do physical-layer attacks, but they will likely have to calibrate the differences between their SDRs before they deploy them.

E. Mobility of target device

Physical-layer tracking would be impossible if the BLE fingerprint of BLE device changes as it moves from one physical location to another. Specifically, fingerprints may change due to differences in the target's physical environment (e.g., multipath in one room vs. another), and differences in motion of the target (e.g., walking vs. driving).

Physical environment: A change in the physical location of the target can alter the received signal's SNR due to changes multipath conditions. However, we observed that this appears to have an insignificant impact on BLE fingerprinting metrics. In Section V-C, we will demonstrate that we can accurately identify 17 target devices across different locations. Furthermore, Figures 12 and 11 show the that above a certain minimum SNR ($\sim$10 dB), changes in SNR do not impact identification accuracy

Speed of Motion: A moving BLE device may experience a velocity-dependent frequency offset due to the Doppler effect [41]. While this may cause a slight drift in the CFO of the BLE target device, the impact is not significant for the frequencies that BLE operates at.

For example, if a BLE device is moving at a velocity of 80 kilometers per hour, and the receiver is stationary, the Doppler frequency offset at 2.4 GHz is about 180 Hz. This is only $\tilde{5}$0% of the median of standard deviation of CFO for BLE devices we observed in the field (Figure 5). Therefore, even at relatively high speed motion, the Doppler shift doesn't impact an attacker's ability to track devices.

Summary: Changing location, or speed, of BLE device has an insignificant impact on the attacker's ability to accurately fingerprint and identify a target device.

V. Field Evaluation

Several of the challenges described in the previous section raise the possibility that there are realistic scenarios where an attacker may falsely identify their target is present when it is not (False Positive), or falsely identify their target is not present when it is (False Negative). Determining how often these errors happen in practice requires a field study. Fortunately, BLE devices constantly beacon, and these beacons contain an anonymous identifier that is stable for 15-minutes. We leverage these properties of BLE to perform a large-scale uncontrolled field study of how severely misidentification errors manifest in real-world environments.

To begin with, we assess how well our BLE tracking toolkit works, even though devices may not have unique fingerprints, and their fingerprint can be affected by temperature variations. We end with two case studies describing how well the end-to-end attack works in the field over multiple days. To the best of our knowledge, this is the first uncontrolled experiment to

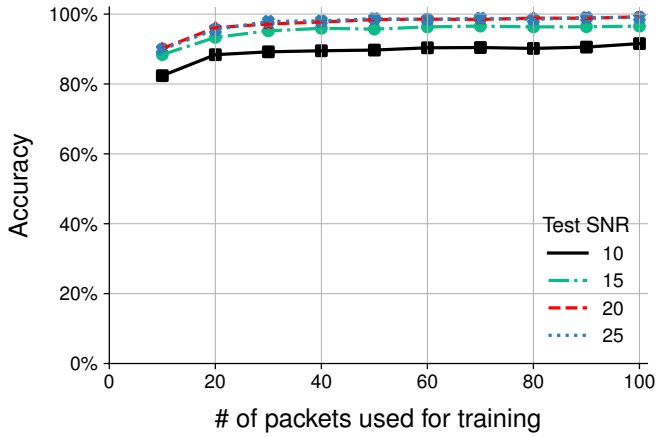

Fig. 11: Identification accuracy with different training sizes

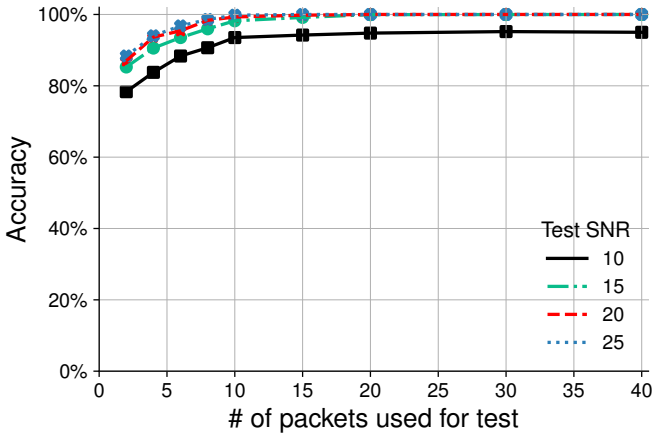

Fig. 12: Identification accuracy with different test sizes

evaluate the effectiveness of a physical-layer tracking attack in practice.

Data Collection

We collected two datasets of BLE beacons from uncontrolled mobile devices. The first dataset was collected in public places that were likely to contain many stationary BLE-enabled mobile devices, including: six coffee shops, a university library, a food court. We set up a USRP N210 in each of these locations for approximately one hour, and opportunistically collected BLE beacons. We observed hundreds of packets from 162 unique devices across all the locations. We used this dataset to evaluate the false positive (and false negative) rate of our BLE tracking toolkit. The second dataset was collected in a facility where many unique devices passed briefly within range of our USRP N210. We observed dozens of packets from 647 unique devices over the course of 20 hours of data collection. We used this dataset to evaluate the uniqueness of BLE physical-layer fingerprints across a large number of devices.

Ethical Considerations: Our data collection is completely passive, and we only capture BLE advertisement packets (i.e., beacons) that devices already broadcast indiscriminately with the intention of being received by any nearby device. Many of these packets originated from pervasive BLE applications like contact tracing and device discovery. To ensure we only capture BLE advertisement packets, we configured our SDR to only capture BLE advertisement frequencies and mask off non-advertisement channels [22]. Furthermore, we ensure that in the decoding stage only undirected advertising packets are passed on to the analysis phase.

The device fingerprints we produce as part of the analysis in this work cannot be directly linked to individual people. Moreover, the BLE advertising packets from which we produce these fingerprints do not reveal any personally identifiable information about the user of the transmitting device. We only performed full identification and tracking on 17 devices that we controlled. According to our university's IRB office, this work does not qualify as human subjects research.

Data Analysis

We fingerprint and identify devices using our BLE tracking toolkit described in Section III. To apply this methodology on field-collected datasets, we first had to determine how many packets an attacker needs to receive from each device to accurately fingerprint and identify it. We found this threshold by performing a controlled experiment using 20 ESP32 BLE chipsets. We tested in varying SNR conditions from 10 to 30 dB—exactly what an attacker would typically see in the field—to see if the number of packets needed for fingerprinting and identification increases when beacons have poor SNR. Next, we identified each of the 20 devices using the algorithm described in Section III-D. We split the captures used for training and test as follows: 80% of the beacons were used for training (i.e., fingerprinting), and 20% for testing (i.e., identifying). We trained with beacons at three SNR values: $\{10, 15, 25\}$ dB. Then, we ran identification tests with beacons that had $\{10, 15, 25\}$ dB SNR independently. We evaluated the identification accuracy of different training sizes with a test size of 10 packets.

Figure 11 shows the accuracy of identifying the devices compared to the number of training packets used for building the device fingerprints. For all SNR values, having 50 packets for training is sufficient. Many BLE devices transmit significantly more than 50 beacons a minute (Table I); therefore we estimate an attacker only needs to isolate a mobile device for at most one minute to get enough packets to fingerprint it.

Figure 12 shows the accuracy of classifying the devices compared to the number of packets used (the number of training packets is fixed to 50 per device). Across the tested SNRs, an attacker only needs 10 packets to accurately identify a device. For the rest of the field study, we use 50 packets to fingerprint a device, and 10 packets to identify a device.

A. False Positives and False Negatives

In the following experiments, we evaluate the likelihood that our BLE tracking toolkit confuses a device that is not a target with a target (False Positive), and the likelihood that it does identify a target when it is present (False Negative).

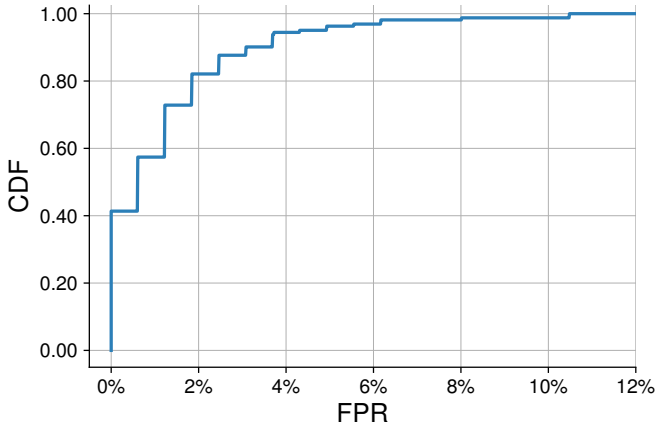

Fig. 13: Dist. of FPR a device when comparing with all others

Given the absence of ground truth of device identities in our dataset, we relied upon the fact that BLE devices have stable MAC addresses for ∼15 minutes (after with they re-randomize the MAC address). Therefore, we used the MAC as ground truth that multiple packets received were from the same device. However, a device's MAC address can be randomized during our data collection, causing us to incorrectly treat the same physical-layer fingerprint as two devices. We mitigated this problem by only considering devices that we observed during one contiguous period of time in each location where we did not observe any new devices, nor any devices that appear to stop transmitting. This filtering left us with 162 devices to use for our false positive and false negative evaluation.

We consider every device (MAC address) $i \in \{1, 2, 3, ..., 162\}$ as a target, and we train our classifier to find that device's fingerprint (Section III-D). Then, for each of the other devices, we run the classifier to see if it identifies them as the target (i) device. If it does, then that is considered a *false positive*. The number of false positives for target device i divided by the total number of devices is the False Positive Rate (FPR) for device i. Next, we fingerprint each target i and run the classifier to see if it fails to identify each device as itself. Each instance of this is a *false negative*. We repeat this process for all the 162 devices (each time one of them is selected as the target), and divide the result by the total number of devices to compute the total False Negative Rate (FNR). We observe our classifier achieves a 2.5% FNR across all 162 devices.

Figure 13 shows the distribution of FPR for each of the 162 devices. The median FPR of a device is only 0.62%. Moreover, 40% of the devices were not confused with any other device (zero FPR), which implies many devices seen in the field have unique physical-layer fingerprints. Owning a device with unique imperfections makes someone particularly vulnerable to BLE tracking attacks. We also observed a small fraction of devices had an FPR as high as 10%.

1) How imperfections contribute to identification: Next, we evaluate how each of the imperfections contribute to identification. Table II shows the FPR and FNR when using CFO, I/Q offset and I/Q imbalance separately, and all together,

Features used	FPR	FNR
CFO only	2.42%	2.45%
I/Q offset only	19.84%	2.39%
I/Q imbalance only	32.53%	1.52%
All Features	**1.21%**	**2.53%**

TABLE II: Hardware imperfection-specific FPR and FNR.

Devices Compared	FPR	FNR
Only Apple Products	1.91%	2.40%
Only other Products	1.15%	2.94%
Apple vs other	0.15%	—
All Devices	**1.21%**	**2.53%**

TABLE III: Manufacturer-specific FPR and FNR.

repeating a similar experiment as we used to compare device manufacturers. CFO contributes the most to identification, as it can have a wider range of values for different devices compared to I/Q imperfections. I/Q imperfections alone have a much higher FPR, but they can resolve the confusion between devices with similar CFO. This same phenomena is also visible our controlled lab experiments (Figure 6) where some devices have CFO values close to each other, but their difference in I/Q imperfection makes them distinguishable. Also, recall that temperature can cause variation CFO while it does not have any notable impact on I/Q imperfections. As a result, I/Q imperfections can help identify the target when it experiences temperature changes.

2) Effect of device model: Based on our controlled experiments (Section IV-A), we expect devices from the same manufacturer to be more likely to be confused than devices of different manufacturers. To test this hypothesis, we used the technique proposed in [11] to distinguish Apple products in our dataset from other devices. About 76% (123 devices) in the dataset are Apple products. The prominence of Apple products in the dataset is likely because Apple enables their BLE-based device handoff service by default on many of their mobile products, including iPhones and Apple Watches.[†]

Table III shows the FPR and FNR of Apple products compared with other products. As expected, the FPR when comparing Apple devices with other Apple devices (1.91%) is greater than the median FPR when comparing across all devices (0.62%). Also, the FPR and FNR when comparing Apple products with other devices is close to zero. This appears to confirm our hypothesis that devices from the same manufacturer are more likely to be similar to each other than devices from different manufactures.

3) Effect of temperature: The temperature of the devices we observe in the field were unlikely to experience significant temperature changes during the course of our data collection. Therefore, we perform a model-based simulation to evaluate the effect of temperature changes on FPR and FNR. Recall that temperature changes affect CFO because of the well-documented relationship between frequency drift of crystal oscillators and their temperature (Section IV-A). Using the curves in [12], we calculate the change in CFO (Δf) as temperature drifts further from the temperature baseline when the device was fingerprinted ($\Delta T\ ^\circ C$). To ensure the target

[†]We collected this dataset before COVID–19 contact tracing launched.

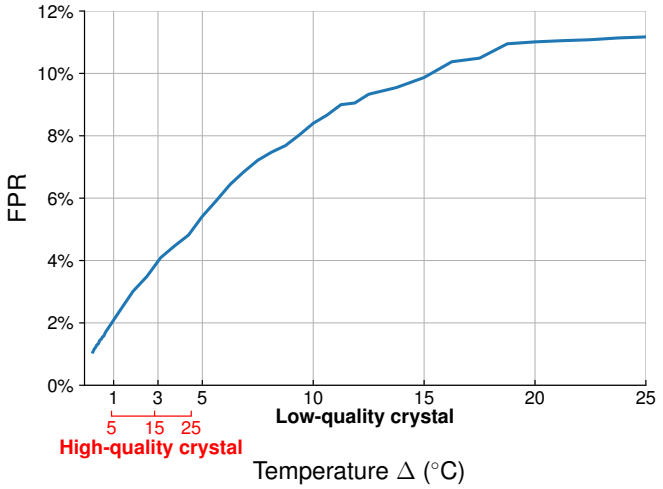

Fig. 14: How oscillator temperature changes affect FPR.

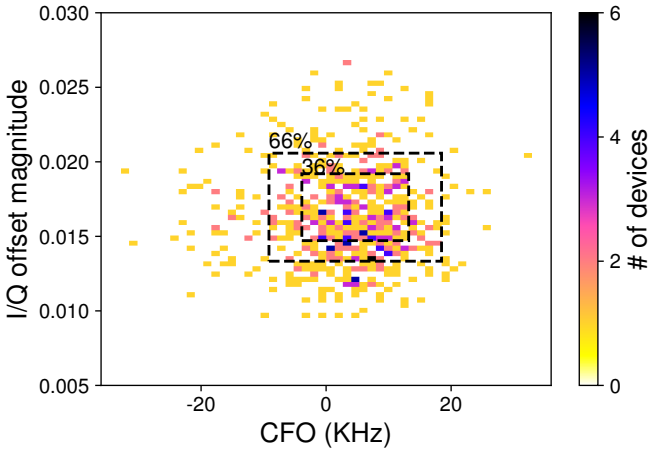

Fig. 15: Histogram of imperfections across 647 BLE devices.

is not missed even if the temperature changes are as large as ΔT $^{\circ}C$, we modified the classifier to accept the device as the target even if the CFO of the device is Δf away from the fingerprinted CFO of the target. The consequence of increasing the range of acceptable CFO values is that it increases the chance of observing a device whose CFO falls in the acceptable range, resulting in an increase in FPR.

Figure 14 presents the FPR as the change in temperature increases. We present the results for both high-quality and low-quality crystals (i.e., different cutting accuracies), as the type of crystal depends on the specific device being targeted. Temperature change causes significantly less change in CFO (and thus less increase in FPR) for high-quality crystals (0 minute cutting accuracy) compared to low quality crystals (8 minute cutting accuracy). For low-quality crystals, FPR increases rapidly as the temperature increases. If the change in temperature is too significant ($25^{\circ}C$), CFO becomes useless for identification: the FPR is the same as if we only used IQ offset and IQ imbalance. In summary, temperature changes can severely limit an attacker's ability to track a target device.

B. Uniqueness of imperfections

Recall that across the 162 devices observed in our first field evaluation dataset, we found ∼40% of the devices to be uniquely identifiable. However, is natural to ask, is the same true at large scale? If the attacker were to observe several hundred devices over multiple days, will we see a similar fraction of devices that are uniquely identifiable?

To answer this question, we performed a larger-scale field data collection. We placed an SDR at the exit of a room where *hundreds of different devices* passed by each day. We recorded the Apple/Google COVID–19 Exposure Notification BLE beacons transmitted by those devices over the course of l0 hours on two days, separated by one week to limit the number of duplicate devices. We computed the mean CFO and mean I/Q offset magnitude for each BLE MAC address we observed in the beacons. The mean hardware imperfections are representative of the fingerprint of the BLE device. To red the chance that we observed the same device with two or more different MAC addresses, we filtered out devices which were observed for a duration longer than three minutes[‡].

We observed 647 unique MAC addresses across the two 20 hours of data collection. Figure 15 shows the 2-Dimensional histogram of the fingerprints of these devices, namely their CFO and I/Q offset magnitude. The number of histogram bins were chosen so that the number of bins (2500) is significantly larger than the total BLE devices observed. Each bin represents a CFO range of ∼1.3 kHz, and an I/Q offset magnitude range of 0.00516. Devices that fall in the same bin are considered to have indistinguishable hardware imperfections. We also show the bounds of the 2D histogram that cover 36% ($\sim\sigma$) and 67% ($\sim 2\sigma$) of the devices (σ because imperfections tend to be normally distributed).

We found that 47.1% (305) of the devices were unique. This confirms that even in a larger data set, ∼40% of devices are uniquely distinguishable. We also observed that devices with overlaps did not overlap with many other devices. For instance, 15% (97) of the devices had similar imperfections with only one other device.

C. Case Study 1: Temporal tracking of many targets

Next, we conduct an experiment to evaluate how well our toolkit can track 17 controlled targets over time, in real world environments. These controlled targets are listed in Table IV. Each target is isolated in an office to capture 50 packets to train the classifier with its fingerprint.

False Negative dataset: Between 2–7 days after we fingerprinted the targets, we individually took them to a different location, and we captured their packets using a USRP N210 sniffer placed 10 ft away from the targets. We did not strictly force the targets to have the same temperature in the office and food court, but both environments were air-conditioned indoor buildings and there was nominal activity on the targets.

False Positive dataset: We evaluated the FPR for these targets using a trace from a coffee shop from our field datasets,

[‡]Apple rotates addresses every 15 mins and Android every 10 mins.

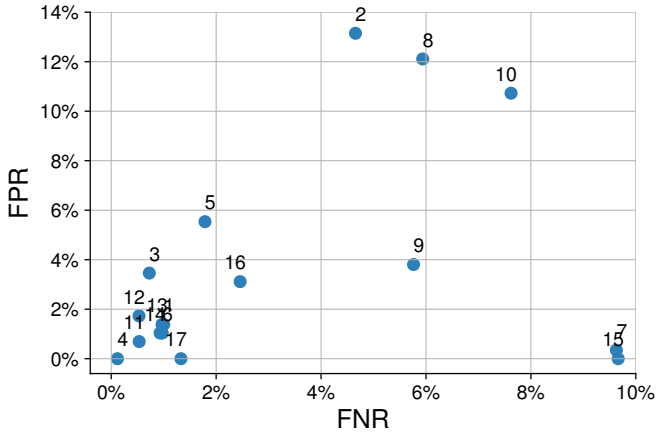

Fig. 16: FNR–FPR for 17 controlled targets.

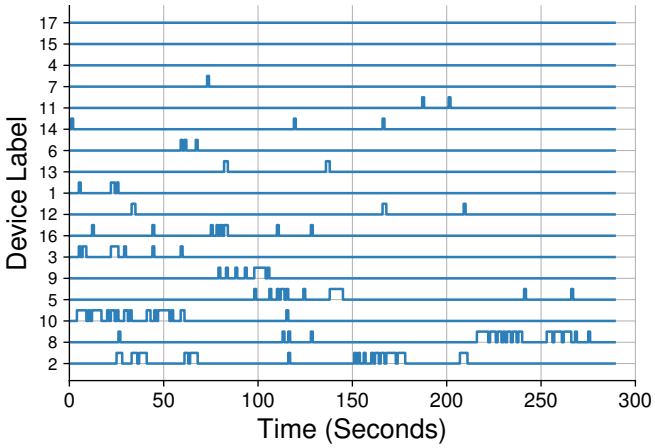

Fig. 17: FPR occurrences over time for each of the 17 targets.

#: Device	#: Device	#: Device
1: iPhone 10	7: iPhone 10	13: MacBook Pro
2: iPhone 8	8: iWatch	14: Thinkpad
3: iPhone 11	9: iPhone 10	15: AirPod
4: Bose Headset	10: iPhone 8	16: Pixel 2
5: iWatch	11: iPhone 10	17: Pixel 5
6: iPhone 8	12: iWatch	

TABLE IV: 17 target devices used for this experiment and their label numbers that are used in Figures 16 and 17.

because we knew the 17 controlled devices were not present during that experiment.

Temporal FNR and FPR: We calculate the FNR and FPR over time, in each 10 second interval of the captures. In each time interval, we provide 10 packets from each MAC address to the classifier to determine if it matches any of the 17 targets' fingerprints. The FNR is the fraction of intervals where the target was present, but was not identified, and the FPR is the fraction of intervals where the target was not present, but was mistakenly identified.

Results: Figure 16 shows the average FNR and FPR for these 17 targets. The average FNR of these controlled targets is 3.21% and the average FPR is 3.5%. Although there are a few devices with high FNR and FPR, most devices have distinguishable hardware imperfections, resulting in low FNR and FPR.

Figure 17 shows the temporal patterns of false positive occurrences for each of the 17 targets in one of the field traces. Each time there is a bump in a device's horizontal line, it means that at least one device was mistakenly identified as being the target during that time interval. We observe that false positives are sometimes short-lived, but often they last for longer than one 10-second interval, possibly indicating a device with similar hardware imperfections came within range of the sniffer.

D. Case Study 2: Tracking a person

Finally, we describe an end-to-end tracking attack we executed on a controlled target (a volunteer who uses an iPhone). The attacker first carries their SDR sniffer close to the target device to obtain the device's physical-layer fingerpr

Simultaneously, the attacker scans for nearby BLE devices using a commonly available BLE scanner phone app, and they record the MAC address of the BLE device with the highest observed signal strength, which is the nearest device (i.e., the target's phone). Later, they use this MAC address to pick out the target device's packets from the raw sniffer capture. Then, they feed these packets into the BLE tracking toolkit to train its classifier with the target device's fingerprint.

After creating the fingerprint, the attacker tracks their target by placing an SDR and laptop close to their target's home. The attacker can determine when the target is home by observing when the classifier running on the laptop indicates the packets received by the SDR match the target device's fingerprint. The attacker tracks their target for one hour, during which the target walks inside and outside the house 2 times. Figure 18 shows the number of unique MAC addresses observed every ten seconds during this hour. There are approximately 30 other devices nearby that could be confused with the target.

The blue bar shown in Figure 19 shows the ground truth of when the person was inside the house during this hour. The attacker's identification toolkit runs once every 10 seconds, and the red bar shows the time durations during which the tracking toolkit thinks the person was present. The bars perfectly match except for immediately prior to minute 10, where the toolkit falsely detects the presence of the target for 50 seconds, even though it had not yet actually returned.

VI. Countermeasures

BLE location tracking based on hardware impairments cannot be defended against by simple software/firmware update mechanisms. These manufacturing variation based properties are baked into the RF signal chain.

One possible defense against this attack requires us to rethink the design of a BLE chipset's signal chain. We envision adding a random time-varying extra frequency offset the crystal oscillator. This would cause the CFO measured at the receiver to also be time-varying and unpredictable. Fortunately, since BLE has a large CFO tolerance (150 kHz [30]), an extra

quency shift will not impact packet decoding.

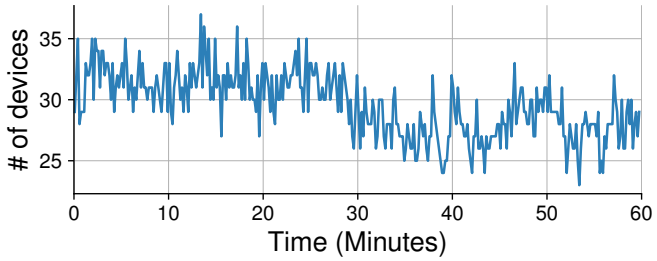

Fig. 18: Number of unique MAC addresses observed over time while tracking the target.

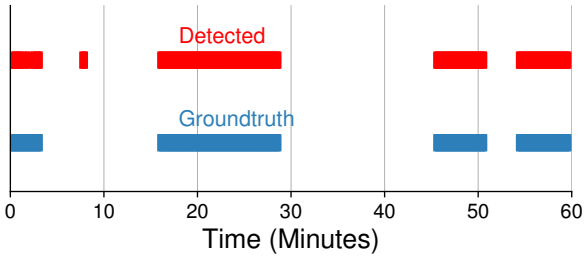

Fig. 19: The blue bar represents the time that the target was present, the red bar represents the time that our tracking toolkit detected the presence of the target.

We also envision another defense that does not require hardware modification. In Section IV-B we observed that CFO changes significantly when a device's internal components heat up and cool down. Internal component temperature depends on the workload running on the phone: a time-varying workload can result in a time-varying CFO. We envision a defense in which a background process runs a computation, and keeps randomly changing the computation in line with the MAC address changes. Unfortunately, a constantly changing workload can result in a constantly changing battery consumption. Worse still, if the device temperature remains constantly elevated, the battery life also decreases over time [20].

VII. Related Work

BLE MAC-Layer Fingerprinting

At its most basic level, BLE's design frustrates MAC-layer fingerprinting. Although BLE advertisements contain a full 6-byte MAC address that is unique to the advertising device, the BLE protocol also has built-in cryptographic MAC randomization. Fortunately, prior work found (and we confirmed) that mobile devices are properly implementing BLE's MAC address randomization [5], [26]. Namely, they found devices are following the BLE specification and periodically (every 10–15 minutes) randomizing their MAC addresses [6].

However, several papers have performed privacy attacks by deriving identifiers from the packet contents of beacons that were not reset properly after the MAC was randomized, for both WiFi [15], [26] and BLE [32], [33], [5], [25], [11] radios. However, all of these attacks fall short as they either require the receiver to continuously listen to beacons from the target devices, or fundamentally rely on identifiers that can easily be removed through simple software updates. This limits the attacker's ability as they must persistently follow a target to track it. Thus, link layer techniques don't provide persistent identifiers that can be utilized for long term tracking of devices.

Physical-layer Fingerprinting

RF fingerprinting using hardware impairments is a well studied field. Researchers have analyzed various hardware impairment based signal properties such as CFO, I/Q offset/imbalance, signal transients and others [9], [39], [17], [23], [35], [24], [4], and leveraged various statistical methods, in recent times deep learning approaches [16], [42], [27] to fingerprint these properties. For instance, the transient portion of the signal has been proposed as a unique signature to classify different wireless devices [38], [13] even Bluetooth signals [18]. However, the transient portion of BLE and Bluetooth signals is only about 2 microseconds and contains insufficient information to uniquely identify a device among tens of devices. Modulation-shape features have also been explored for RF fingerprinting devices such as RFID transponders [14]. However, the Gaussian shape in GFSK modulation of BLE signals is generated digitally in most personal electronic devices such as phones, and thus, cannot be used as a unique fingerprint. In the WiFi literature, CFO and I/Q imperfections (I/Q origin offset and I/Q imbalance) are two well recognized features which have been shown to be the most separable features for WiFi fingerprinting [9].

BLE hardware in mobile devices are similar in architecture and suffer from the same hardware impairments as WiFi radios. Despite that, other than a few efforts at coarse CFO extraction utilizing specialized hardware (CC2400) [34], [40], there exists limited work in RF fingerprinting of these BLE chipsets. This is primarily because the techniques to extract these properties rely upon the presence of long known sequence of bits and pilots, a convenience not provided in simple BLE transmissions. Even if the WiFi techniques were utilized for BLE signals, they would yield coarse estimates of these persistent identifiers, which are not particularly useful when fingerprinting a large amount of devices. Furthermore, to be able to utilize any RF fingerprinting technique as a privacy attack, we need to have evidence that it works in real world settings. Unfortunately, all prior work in RF fingerprinting has been performed in controlled environmental settings with a defined set of devices. We design a technique to extract the hardware impairments such as CFO and I/Q offset from BLE signals at a fine granularity. We were then able to collect a massive dataset of BLE devices in the wild and analyze their RF fingerprints to evaluate the potentials and limitations of the physical-layer fingerprinting privacy attack in the wild. We also demonstrated the feasibility of a location privacy (tracking) attack utilizing these physical-layer parameters in realistic scenario.

VIII. Conclusion

In this work, we evaluated the feasibility of physical-layer tracking attacks on BLE-enabled mobile devices. We found that many popular mobile devices are essentially operating as tracking beacons for their users, transmitting hundreds of BLE beacons per second. We discovered that it is indeed feasible to get fingerprints of the transmitters of BLE devices, even though their signal modulation does not allow for discovering of these imperfections at decoding time. We developed a tool that automates recovering these features in transmitted packets.

Then, we used this tool to determine what challenges an attacker would face in using BLE to track a target in the wild. We found that attackers can use low-cost SDRs to capture physical-layer fingerprints, but those identities may not be easy to capture due to differences in devices' transmission power, they may not be stable due to temperate variations, and they may be similar to other devices of the same make and model. Or, they may not even have certain identifying features if they are developed with low power radio architectures. By evaluating the practicality of this attack in the field, particularly in busy settings such as coffee shops, we found that certain devices have unique fingerprints, and therefore are particularly vulnerable to tracking attacks, others have common fingerprints, they will often be misidentified. Overall, we found that BLE does present a location tracking threat for mobile devices. However, an attackers ability to track a particular target is essentially a matter of luck.

IX. Acknowledgements

We would like to thank our shepherd and the anonymous reviewers from IEEE S&P 2022 for their insightful comments. We also thank the reviewers from MobiSys 2021 and USENIX Security 2020 whose feedback lead to this manuscript. Also, thanks to Stefan Savage for his helpful comments. This work was supported by in part by Qualcomm's Innovation Fellowship, and a gift from Amateur Radio Digital Communications.

References

[1] Apple Inc. Use Continuity to connect your Mac, iPhone, iPad, iPod Touch, and Apple Watch. https://support.apple.com/en-us/HT204681.

[2] Apple Inc. / Google Inc. *Exposure Notification - Bluetooth Specification*, Apr. 2020. v1.2.

[3] Apple Inc. / Google Inc. *Exposure Notification - Frequently Asked Questions*, Sept. 2020. v1.2.

[4] M. Azarmehr, A. Mehta, and R. Rashidzadeh. Wireless Device Identification using Oscillator Control Voltage as RF Fingerprint. In *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pages 1–4, April 2017.

[5] J. K. Becker, D. Li, and D. Starobinski. Tracking Anonymized Bluetooth Devices. *Proceedings on Privacy Enhancing Technologies*, 2019(3):50 – 65, 2019.

[6] Bluetooth SIG. Bluetooth Technology Protecting Your Privacy. https://www.bluetooth.com/blog/bluetooth-technology-protecting-your-privacy/, Apr. 2015.

[7] Bluetooth SIG. Bluetooth Technology Protecting Your Privacy. https://www.bluetooth.com/blog/bluetooth-technology-protecting-your-privacy/, Apr. 2015.

[8] K. Bonne Rasmussen and S. Capkun. Implications of Radio Fingerprinting on the Security of Sensor Networks. In *2007 Third International Conference on Security and Privacy in Communications Networks and the Workshops - SecureComm 2007*, pages 331–340, 2007.

[9] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless Device Identification with Radiometric Signatures. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, MobiCom '08, page 116–127, New York, NY, USA, 2008. Association for Computing Machinery.

[10] California Health Care Foundation. Preliminary Research suggests COVID-19 Warning App has slowed Transmission of the Virus. https://www.chcf.org/blog/preliminary-research-suggests-covid-19-warning-app-slowed-transmission-virus/.

[11] G. Celosia and M. Cunche. Discontinued Privacy: Personal Data Leaks in Apple Bluetooth-Low-Energy Continuity Protocols. *Proceedings on Privacy Enhancing Technologies*, 2020(1):26–46, 2020.

[12] CTS Corporation. Crystal Basics. https://www.ctscorp.com/wp-content/uploads/Appnote-Crystal-Basics.pdf.

[13] B. Danev and S. Capkun. Transient-based Identification of Wireless Sensor Nodes. In *2009 International Conference on Information Processing in Sensor Networks*, pages 25–36, April 2009.

[14] B. Danev, T. S. Heydt-Benjamin, and S. Čapkun. Physical-Layer Identification of RFID Devices. In *Proceedings of the 18th Conference on USENIX Security Symposium*, SSYM'09, page 199–214, USA, 2009. USENIX Association.

[15] J. Freudiger. How Talkative is your Mobile Device?: An Experimental Study of Wi-Fi Probe Requests. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '15, pages 8:1–8:6, New York, NY, USA, 2015. ACM.

[16] S. Gopalakrishnan, M. Cekic, and U. Madhow. Robust Wireless Fingerprinting via Complex-Valued Neural Networks. *arXiv preprint arXiv:1905.09388*, 2019.

[17] J. Hall, M. Barbeau, and E. Kranakis. Enhancing Intrusion Detection in Wireless Networks using Radio Frequency Fingerprinting. In *Communications, internet, and information technology*, pages 201–206, 2004.

[18] J. Hall, M. Barbeau, and E. Kranakis. Detecting Rogue Devices in Bluetooth Networks using Radio Frequency Fingerprinting. In *In IASTED International Conference on Communications and Computer Networks*. Citeseer, 2006.

[19] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis. Deep Learning for RF Fingerprinting: A Massive Experimental Study. *IEEE Internet of Things Magazine*, 3(1):50–57, 2020.

[20] S. Kang, H. Choi, S. Park, C. Park, J. Lee, U. Lee, and S.-J. Lee. Fire in Your Hands: Understanding Thermal Behavior of Smartphones. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery.

[21] I. O. Kennedy, P. Scanlon, and M. M. Buddhikot. Passive Steady State RF Fingerprinting: A Cognitive Technique for Scalable Deployment of Co-Channel Femtocell Underlays. In *2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 1–12, Oct 2008.

[22] M. Khazraee, Y. Guddeti, S. Crow, A. C. Snoeren, K. Levchenko, D. Bharadia, and A. Schulman. Sparsdr: Sparsity-proportional backhaul and compute for sdrs. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '19, page 391–403, New York, NY, USA, 2019. Association for Computing Machinery.

[23] M. Köse, S. Taşcioğlu, and Z. Telatar. Wireless Device Identification using Descriptive Statistics. *Communications Fac. Sci. Univ. of Ankara Series A2-A3*, 57(1):1–10, 2015.

[24] P. Liu, P. Yang, W. Song, Y. Yan, and X. Li. Real-time Identification of Rogue WiFi Connections using Environment-Independent Physical Features. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 190–198, April 2019.

[25] J. Martin, D. Alpuche, K. Bodeman, L. Brown, E. Fenske, L. Foppe, T. Mayberry, E. Rye, B. Sipes, and S. Teplov. Handoff All Your Privacy – A Review of Apple's Bluetooth Low Energy Continuity Protocol. *Proceedings on Privacy Enhancing Technologies*, 2019.

[26] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown. A Study of MAC Address Randomization in Mobile Devices and When it Fails. *Proceedings on Privacy Enhancing Technologies*, 2017(4):365–383, 2017.

[27] K. Merchant, S. Revay, G. Stantchev, and B. Nousain. Deep Learning for RF Device Fingerprinting in Cognitive Communication Networks. *IEEE Journal of Selected Topics in Signal Processing*, 12(1):160–167, Feb 2018.

[28] A. Nicolussi, S. Tanner, and R. Wattenhofer. Aircraft Fingerprinting Using Deep Learning. In *2020 28th European Signal Processing Conference (EUSIPCO)*, pages 740–744, 2021.

[29] A. C. Polak, S. Dolatshahi, and D. L. Goeckel. Identifying Wireless Users via Transmitter Imperfections. *IEEE Journal on Selected Areas in Communications*, 29(7):1469–1479, August 2011.

[30] Y. Rekhter and T. Li. Core Specification 5.3. Technical report, Bluetooth SIG, July 2021.

[31] P. Robyns, E. Marin, W. Lamotte, P. Quax, D. Singelée, and B. Preneel. Physical-Layer Fingerprinting of LoRa Devices Using Supervised and Zero-Shot Learning. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '17, page 58–63, New York, NY, USA, 2017. Association for Computing Machinery.

[32] M. Ryan. Bluetooth: With Low Energy Comes Low Security. In *Presented as part of the 7th USENIX Workshop on Offensive Technologies*, Washington, D.C., 2013. USENIX.

[33] D. Spill and A. Bittau. Bluesniff: Eve meets Alice and Bluetooth. In *Proceedings of the first USENIX workshop on Offensive Technologies*, page 5. USENIX Association, 2007.

[34] W. Sun, J. Paek, and S. Choi. CV-Track: Leveraging Carrier Frequency Offset Variation for BLE Signal Detection. In *Proceedings of the 4th ACM Workshop on Hot Topics in Wireless*, HotWireless '17, page 1–5, New York, NY, USA, 2017. Association for Computing Machinery.

[35] W. C. Suski II, M. A. Temple, M. J. Mendenhall, and R. F. Mills. Using Spectral Fingerprints to Improve Wireless Network Security. In *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, pages 1–5, Nov 2008.

[36] TechInsights. Texas Instruments CC2640R2F SimpleLink Bluetooth Low Energy Wireless MCU RF Architecture Report. Technical report, TechInsights, 02 2018.

[37] C. Troncoso, M. Payer, J.-P. Hubaux, M. Salathé, J. Larus, E. Bugnion, W. Lueks, T. Stadler, A. Pyrgelis, D. Antonioli, L. Barman, S. Chatel, K. Paterson, S. Čapkun, D. Basin, J. Beutel, D. Jackson, M. Roeschlin, P. Leu, B. Preneel, N. Smart, A. Abidin, S. Gürses, M. Veale, C. Cremers, M. Backes, N. O. Tippenhauer, R. Binns, C. Cattuto, A. Barrat, D. Fiore, M. Barbosa, R. Oliveira, and J. Pereira. Decentralized Privacy-Preserving Proximity Tracing, 2020.

[38] S. Ur Rehman, K. Sowerby, and C. Coghill. Rf Fingerprint Extraction from the Energy Envelope of an Instantaneous Transient Signal. In *2012 Australian Communications Theory Workshop (AusCTW)*, pages 90–95, Jan 2012.

[39] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir. Fingerprinting Wi-Fi Devices Using Software Defined Radios. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '16, pages 3–14, New York, NY, USA, 2016. ACM.

[40] J. Wu, Y. Nan, V. Kumar, M. Payer, and D. Xu. BlueShield: Detecting Spoofing Attacks in Bluetooth Low Energy Networks. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, pages 397–411, San Sebastian, Oct. 2020. USENIX Association.

[41] F. Xiong and M. Andro. The Effect of Doppler Frequency Shift, Frequency Offset of the Local Oscillators, and Phase Noise on the Performance of Coherent OFDM Receivers. Technical report, NASA, 2001.

[42] J. Yu, A. Hu, F. Zhou, Y. Xing, Y. Yu, G. Li, and L. Peng. Radio Frequency Fingerprint Identification based on Denoising Autoencoders. In *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–6, Oct 2019.