Evaluating Physical-Layer BLE Location Tracking Attacks on Mobile Devices

***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 [9] 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], [36]. 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 [13], [8], [17], [34], [27], [20], [26], [7], including a recent attempt to distinguish 10,000 WiFi [18] chipsets.

However, 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 fingerprinting techniques 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 searching for a set of imperfections added 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 evaluating 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) operating systems configure the BLE transmit power level differently resulting in some devices having lower SNR fingerprints (3) the temperature range that mobile devices experience in the field introduces significant changes to hardware impairments, (4) the low-power receivers that an attacker would use in the wild for RF fingerprinting are less accurate than the tools used in prior studies [8].

Our final contribution is evaluating 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, and that common BLE identifiers are stable for 15 minutes, to perform an uncontrolled field study where we evaluate the feasibility of tracking a single BLE device while operating in public spaces where there are hundreds of other nearby devices, such as coffee shops and libraries. To the best of our knowledge, our work is the first to evaluate the feasibility of an RF fingerprinting attack in real-world scenarios.

We show that across over 100 devices observed in 6 locations, it is feasible to track a specific mobile device by its physical-layer fingerprint. However, we also observe that certain devices are much more similar to others, and temperature variations can change a device's metrics, both issues can lead to significant misidentification rates. In summary, we find that physical layer tracking of BLE devices using low-cost receivers 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 [31], [24], 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 ($\sim$\$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 two 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 smart watches.

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.

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.

Mobile devices send BLE beacons continuously: We observed continuous BLE beaconing from all ~~of~~ the six mobile devices shown in Table I. Even when all of these mobile devices have their ~~screen~~ 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 quickly 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 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 device matches one of the BLE devices nearby the sniffer.

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

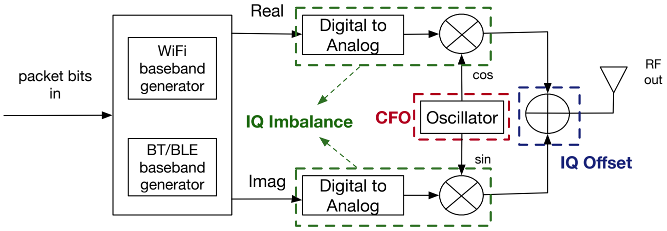

Fig. 1: Architecture of WiFi/BLE combo chipsets

of imperfections are introduced by different transmitter architectures. Therefore, we need to understand the architecture of 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 [8].

The following describes how each of these metrics are calculated and how they result from manufacturing variations:

CFO 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 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 the 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.

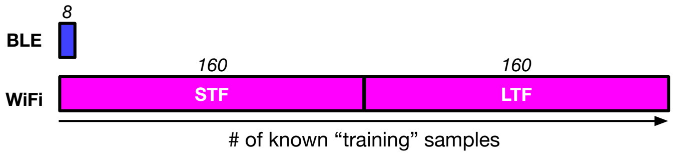

Fig. 2: Length of known samples in BLE and WiFi packets.

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, the structure of BLE's baseband signals are so simple that a receiver does not need to directly measure the CFO and ~~IQ~~ I/Q imperfections accurately. BLE signals are simple narrowband Gaussian Frequency Shift Keying (GFSK), so a receiver does not need to precisely correct imperfections to decode received packets correctly. Conversely, WiFi signals are wideband multi-carrier waveforms, therefore their decoding algorithms requires correcting for CFO and ~~IQ~~ I/Q imperfections to decode packets correctly.

~~As a result of this of this~~ Due to 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 do implement very course grained CFO correction using the small number (i.e., 8) of known samples in the preamble of each packet. 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 course grained average CFO [33]. These CFO estimates are course grained because they only rely on 8 samples in the preamble. Indeed, with only 8 samples, the theoretical limit of CFO accuracy 2 kHz assuming 3 degree phase noise. Moreover, inaccurate ~~coarse-grained~~ coarse compensation of CFO significantly affects our ability to measure I/Q imperfections, since inaccurately compensating CFO will result in time-dependent phase shift distorting the I/Q constellation.

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

Comparison with other fingerprinting methods: The fingerprinting methodology described in this section is the first physical-layer identification method that can accurately estimate CFO and I/Q imperfections of BLE signals. Prior fingerprinting techniques were developed for other protocols that do not have these two fundamental fingerprinting challenges (e.g., WiFi), therefore, ~~thy~~ they can not be used to fingerprint BLE [38], [8], [21], [16], [34], [25], [29], [15].

Although neural networks can address these challenges, we did not use them in this work because of the following reasons: (1) They limited our ability to perform a detailed evaluation of the strengths and limitations of each of the individual types of hardware imperfections that we can use for device identification. Specifically, neural networks make it difficult to determine the significance and distinguishably of each type of hardware imperfection (e.g.~~CFO, IQ~~, CFO, I/Q offset). (2) Neural network training can overfit to a specific bit

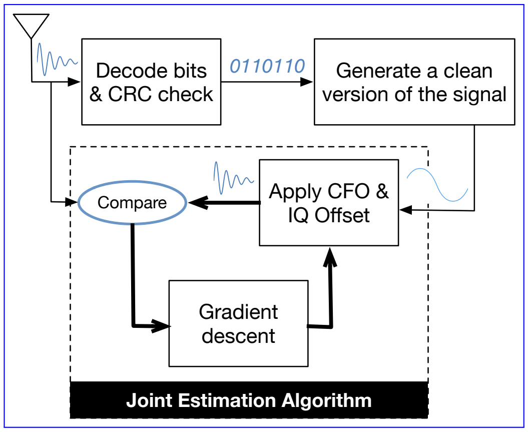

Fig. 3: ~~Overview of our~~ Our new BLE imperfection ~~measurement~~ estimation method.

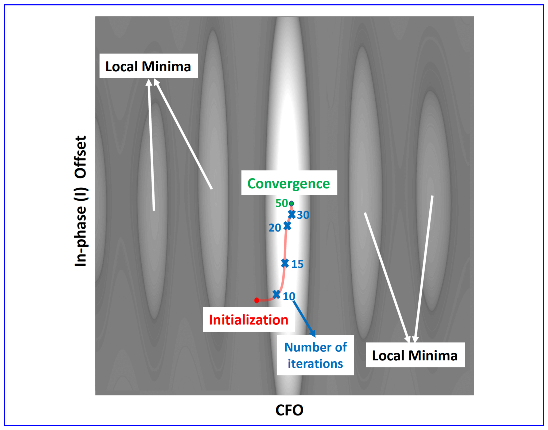

Fig. 4: How the optimization based algorithm converges to ~~the accurate hardware imperfection values such as~~ jointly estimate CFO and ~~IQ~~ I/Q offset.

pattern in a packet, rather than the transmitter imperfections. This is a problem because BLE advertisements do not have a stable bit pattern: MAC addresses change every 15 minutes. (3) Our preliminary experiments with neural nets also indicated that they require significantly more training data than the conventional classification we describe.

Our methodology: We demonstrate it is possible to fingerprint BLE by overcoming the two primary challenges of BLE fingerprinting with two novel fingerprinting techniques (Figure 3). First, to overcome the challenge where measuring I/Q imperfections requires accurately measuring CFO, we present a novel approach that jointly estimates CFO and I/Q imperfections. Second, to accurately estimate CFO and I/Q imperfections with only BLE's short known training symbol sequence, we decode the entire received BLE packet and reconstruct a clean BLE waveform (only when the CRC check passed). Therefore, instead of relying on the 8 known samples in the preamble, we can utilize the entire packet ($\sim$370 samples). These additional samples provide a theoretical CFO measurement precision of about 40 Hz compared to 2 kHz from ~~coarse-grained~~ coarse estimation.

Then, we distort the waveform with different CFO and I/Q imperfections, until we find CFO and I/Q estimates such the reconstructed signal matches the original received signal. Brute force search for these hardware imperfection parameters and trying all possible values has a huge computational complexity as the search space for these imperfection parameters is vast. As a result, we use optimization techniques to efficiently move towards the optimal value of these parameters.

1) Jointly estimating CFO and I/Q: Let $y = Real\{y\} + jImag\{y\}$ be the captured baseband 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 aforementioned hardware imperfections will slightly change the 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. Considering the effects of CFO, ~~IQ offset and IQ~~ I/Qoffset and I/Q imbalance, we can write

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

where f_o, ϕ_o, A, $\frac{1-\epsilon}{1+\epsilon}$, ϕ, I and Q denote CFO, phase offset, normalized amplitude of the signal, IQ amplitude imbalance, IQ 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 as shown in Figure 4. Consequently, any optimization technique may end up in a local optima. To avoid this, we initialize the variables properly to increase the chance of finding the global minimum significantly. Although theoretically it will not guarantee ending up in the global minimum for arbitrary optimum numbers of these variables, we found that in practice we will reach the optimum value with this initialization in practical conditions.

To initialize CFO, 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 $(0, 0)$ since the phase changes according to GFSK modulation but the amplitude is always constant. However, I/Q imperfection will change this constellation. Specifically, I/Q offset will shift the center of the circle as it is equivalent to adding a fixed complex term to the ideal signal, and IQ imbalance will change the shape from a circle to a tilted ellipse. As a result, to get an initial estimation of IQ 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 IQ offset and initial IQ imbalance can be obtained from the ratio of minor and major diameter and rotation angle of the ellipse.

Although, these initial estimations provide an initialization close to optimum, they are not accurate. As mentioned earlier, this initialization of CFO is not accurate and robust enough as it only relies on an 8-microsecond preamble. Also, mismatch in CFO compensation will cause a time-dependent phase shift which distorts the I/Q constellation. Therefore, the initial IQ offset and imbalance estimation will also have errors. Consequently, we employ optimization techniques to jointly estimate hardware imperfection parameters accurately and robustly.

Finding these optimal values can be extremely computationally expensive. Indeed, the naive approach would be to brute force try all possible values for CFO, IQ offset and IQ imbalance to find the optimal values. Instead, we used gradient descent to solve the optimization problem, as it ensures that we move towards the optimal values after each step. Specifically, we use a widely-used fast 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 an instance of how we start from the initial estimations of CFO and IQ imperfections, then move toward the optimal values of CFO and IQ imperfections using gradient descent (the red line), and converge to the accurate estimations of CFO and IQ imperfections in a few iterations.

However, as mentioned earlier, this optimization problem is not convex, and we may converge to the local optima. Therefore, if after convergence, the average of F was not less than a certain threshold which is determined according to SNR, we add certain steps to the first initialization and repeat the aforementioned ~~gradients~~ 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 optimization is chosen as the summation of all PHY samples across the packet, which diminishes the impact of AWGN and provide more robust information and granularity.

Evaluating CFO estimation accuracy: To evaluate the accuracy of our new fine-grained fingerprinting algorithm compared to ~~coarse-grained~~ coarse BLE CFO estimation, we compute CFO for 100 packets from 100 of BLE transmitters observed in the field.Figure 5 shows the CDF of the standard deviation of CFO for both techniques. We see that our fine-grained CFO estimation significantly reduced the standard deviation of CFO estimation for all devices. This reduces the within-class variance and makes these devices have significantly more unique fingerprints.

Summary: For the first time we showed that it is feasible to estimate CFO and IQ imperfections of WiFi/BLE combo chipsets accurately based on the simple BLE signal itself; in other words, without needing the rich signal features that are present in WiFi.

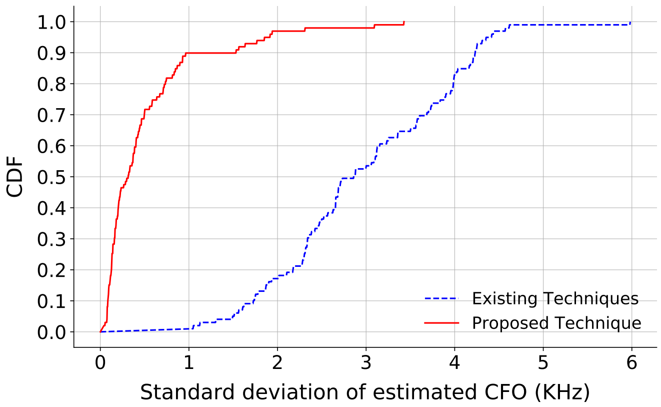

Fig. 5: Comparing the CFO estimation of existing coarse-grained techniques with our proposed technique.

2) Profiling and identifying the device: The first step in deploying our RF fingerprinting attack is to capture the BLE signal. We use an SDR to capture raw I and Q samples of BLE. Next, we use the captured signal to fingerprint the device. The entire processing flow can be divided into two stages, Fingerprinting Stage and Identification Stage. In the former stage, the device is isolated, and we capture a number of packets from the target device to build a profile for the device (training packets). The latter stage, employs this profile to identify the device when the MAC addresses is changed.

Fingerprinting Stage: For each packet from a device D, CFO and IQ imperfections can be extracted with a high resolution using algorithm described in III-C. Let $x_1, ..., x_N$ be the CFO and IQ 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 with a new MAC address belongs to device D, indicating that the target device is present. 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 is a way to measure 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 $distance(x_t, \mu_D, \Sigma_D) < thresh$, packet x_t belongs to the target device D and device D is identified. Otherwise, packet x_t belongs to some other device in the world that we are not looking for. This threshold is chosen by using the validation set. One could choose a threshold that guarantees a certain level of FNR according to the validation set. Another way is to pick a threshold that balances both FPR and FNR. To obtain that goal, we pick the threshold that minimizes $FPR^2 + FNR^2$ so that neither FNR nor FPR gets

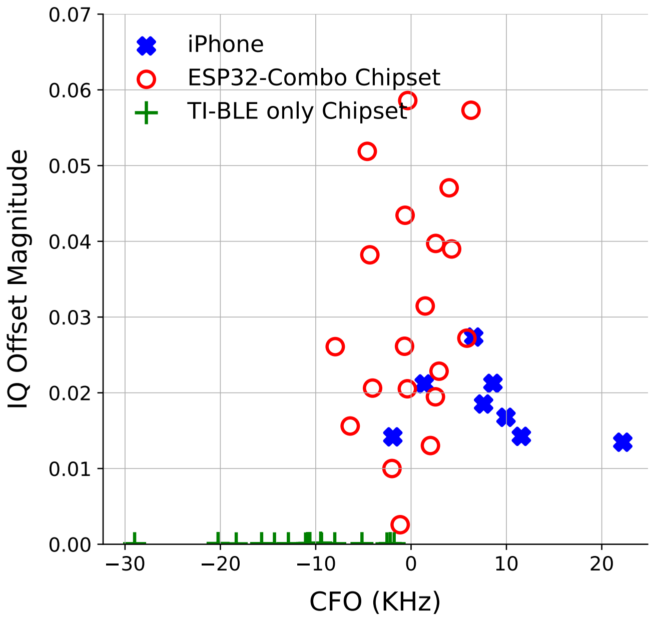

Fig. 6: Comparing the fingerprints of 48 BLE chipsets

much larger than the other. In this paper, we use these two methods for selecting the threshold depending on the goal of analysis in different experiments.

Moreover, since the MAC address is fixed for a period of time, we receive a number of packets with the same MAC address which we know belong to the same device. As a result, we can make a decision about the identity of the MAC address instead of the individual packets. One way that we found most effective, was to first average the feature vector x for all packets with the same MAC address and then compute the Mahalanobis distance. This would further reduce the tolerance due to estimation error and inherent tolerance of features.

IV. Challenges

In this section, we describe the four primary challenges that can limit the effectiveness of physical-layer BLE tracking attacks. We also use controlled experiments to investigate how significant these issues are in practice. We found that although BLE fingerprinting is likely to be feasible, several common situations can make BLE identification significantly more difficult.

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 48 devices with three different popular chipsets. Specifically, 8 recent iPhones, 20 ESP32 WiFi/BLE microcontrollers, and 20 TI CC2640 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). Then, we computed the fingerprint metrics using the tool described in the previous section, and averaged the metrics over all 100 packets to remove any transient noise effects.

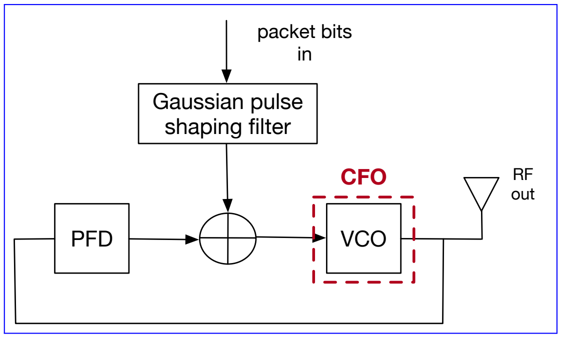

Fig. 7: TI's BLE-only transmitter. It is not an I/Q modulator.

Figure 6 shows the fingerprint metrics for each of the 48 devices. We plot only CFO and I/Q offset 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. There only are a few devices with similar signatures. However, the distribution of device fingerprints is not uniform:; each device model appears to be drawing from a different distribution of fingerprints. All devices appear to have similar range of CFO values, but the I/Q offset metric appears to be extremely model dependent. In particular, the ESP32 devices have a much larger range of I/Q offset fingerprints than the iPhones, which may be reasonable because ESP32s are very low cost chipsets compared to the high-speed WiFi+BLE combo chipsets used in iPhones.

The most surprising model-dependent metric is the non-existent I/Q offset of the TI low power BLE-only chipsets. The primary difference between the TI chipset and the other models tested, is that the TI can only transmit BLE (not WiFi). Recall in Section III, we described how unlike WiFi, BLE is not an inherently I/Q modulated protocol; therefore, the BLE-only chipset does not require the use of an I/Q modulator. We confirmed this suspicion by finding a TI application note describing the TI BLE chipset radio architecture, TI describes how their low power transmitter uses a PLL-based FSK modulator [35].

Summary: An attacker's ability to differentiate a particular target's unique fingerprint depends on the model of BLE chipset it is using, as well as the model of chipset the other devices nearby are using. Distinguishing devices with the same model chipset is likely more difficult than distinguishing only on device of one model near devices of only other models. This may make this attack more difficult because targets are likely to use popular devices (e.g., iPhone). Although, if the target happens to be a device that is not common in that environment, they are likely to have a much more unique fingerprint.

B. *Temperature stability of BLE fingerprints*

A device's BLE fingerprint must be stable to track it over multiple locations and days. However, transmitter impairments — CFO in particular — may drift if the temperature the device is operating in changes. CFO is a product of imperfections in the crystal oscillator used to generate the BLE transmission's

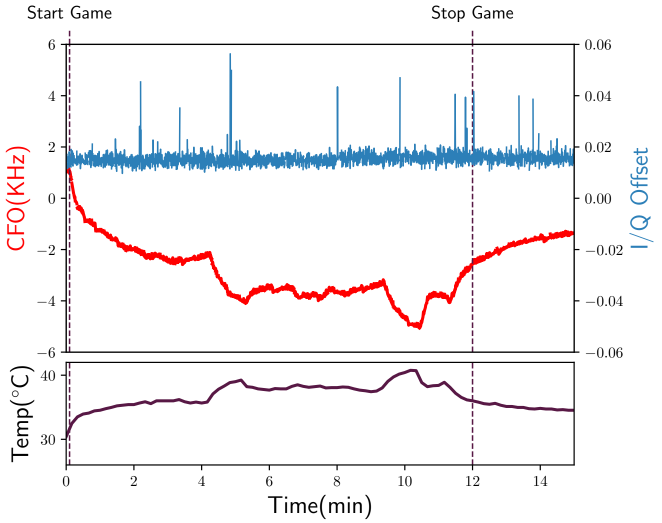

Fig. 8: Metric stability while playing a GPU-intensive game

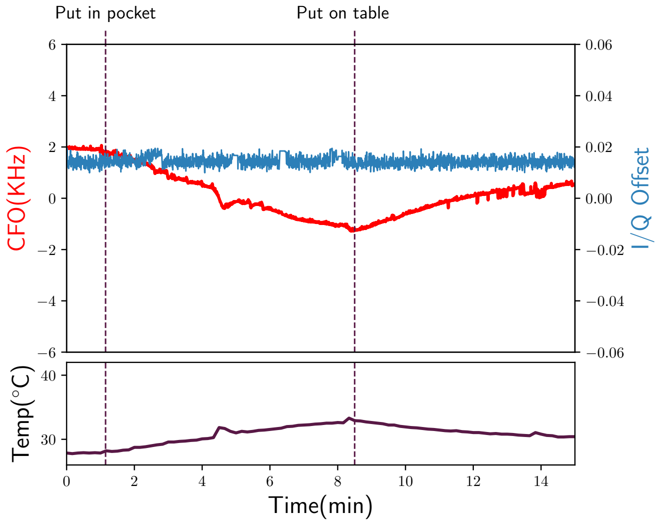

Fig. 9: Metric stability while putting the phone in a pocket

center frequency. A crystal oscillator's frequency error has a well-defined relationship with its temperature. The model of this relationship is called the "Bechmann curve". This relationship depends on the type and specifications of the crystal oscillator. It is possible that BLE devices do not have this instability, especially if they use the same temperature compensated timing source as is used for its high-data rate cellular communication. Also, even if they are using uncompensated crystals for BLE, the impact of temperature significantly depends on the cut angle and face of the crystal [11]. The relationship between these temperature changes and the I/Q modulation metrics is not as well understood.

Smartphones are a particularly concerning device for this issue. The internal temperature of smartphones can significantly change based on device activity, and also with variation in ambient temperature [19].

To investigate the impact of changes in temperature, we put a Motorola Moto G6 phone in two common use cases that may affect temperature: playing a graphics-heavy game (Asphalt 9), and putting an idle phone into a user's pants pocket. This phone was running a COVID contact tracing app to generate BLE beacons, and the phone was kept in normal operation state (WiFi, LTE, Bluetooth all on). Each test ran for 15 minutes, and we captured the fingerprint metrics with a USRP N210. We also captured all of the internal temperature sensor data from the device, and we report on the sensor that most closely correlated with the changes in CFO, which was the sensor in the Power Management Integrated Circuit.

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.

In the game experiment (Figure 8), we observe that the CFO changes significantly, and with a linear relationship to the changes in temperature. As the game begins and ends, the CFO increases, and it decreases when the game ends. At the peak temperature (+10°C above baseline), we observe a significant CFO variation of up to 7 kHz.

The idle experiment (Figure 9), is likely to represent the more common use case: a significant fraction of a phone's lifetime is likely to be spent idle on tables or in users' pockets. The peak change in CFO is much less significant than the game experiment (2 kHz vs. 7 kHz). However, it is still significant enough to introduce confusion with other devices that have similar I/Q metrics, or no I/Q metrics (see Figure 6).

Figures 8 and 9 both show that I/Q offset (and I/Q imbalance which is not shown), do not present a correlation with temperature.

Summary: Device temperature appears to significantly change the fingerprint metrics of a device. 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. Putting an idle device in a user's pocket does change the metrics significantly enough to cause confusion as well. Ideally, an attacker would get an initial fingerprint of the most common use case for the device: idle in the user's pocket. There is another subtlety though. A device's internal temperature will always tend toward the ambient temperature. Although small changes in ambient temperature will not cause a significant difference in CFO; if a device experiences a significant change in ambient temperature, the attacker would have to acquire a new initial fingerprint.

C. Differences in BLE transmitter configurations

The transmit power of BLE beacons affects how far a target can be when an attacker receives their beacons. The BLE APIs on mobile devices often allow software to configure the beacon transmit power to a level that matches the needs of their application. Any difference in transmit power between devices may make it more difficult for an attacker to determine where their target was when they observed their beacons. One may assume that all similar devices, especially when they are running the same popular app, would use the same transmit power. This is particularly expected for contact tracing apps,

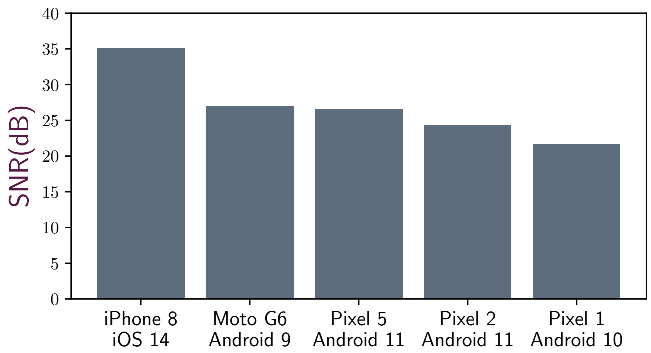

Fig. 10: Average received SNR across devices running BLE contact tracing

where transmit power correlates with distance where the contact occurred.

We measured the received power of 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 15 feet away from the radio inside an office building. We observed 5 different phones, running latest version of iOS and different versions of Android. We installed the same official state COVID-19 contact tracing app on all the devices. Then, we averaged the SNR over 100 beacon packets from each of the devices.

Figure 10 shows that the iPhone has an average SNR 10 dB higher than all other Android phones we tested. This resulted in a significant difference in distance at which beacons were detectable by the USRP. Anecdotally, we observed that this resulted in iPhones being detectable over 7 meters farther the Android devices.

Summary: For an attacker attempting to track a target, there may be differences in software configuration depending on specifically what device they are tracking, even if the device is running an app they have tested on a different device. We observed a significantly reduced effective range for Android than iPhones. Consequently, attackers needs to get closer to the Android devices to track them.

D. Quality of an attacker's sniffer hardware

Physical-layer fingerprinting attacks can require an expensive high-quality Software-Defined Radio (SDR) to execute. The more expensive the required SDR is, the fewer locations an attacker can deploy them to track their target. Recently, several low-cost hobbyist SDRs have become widely available. Therefore, attackers may try to deploy these SDRs to increase their coverage.

We evaluate the difference in SDR quality by comparing the metrics we observe while receiving the same beacons from one target. Specifically, we send BLE packets from a single iPhone device and measure the same metrics with both an expensive USRP N210 ($3,400) and a hobbyist LimeSDR-Mini ($179). We compare the average value and standard deviation of these metrics to evaluate if the two devices reach the same value, and to see if they provide similar metric stability. Similar to the other 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 much lower mean of -8.07 kHz but with a similar standard deviation of 114 Hz. The difference is likely due to the fact that the Lime-SDR uses a lower-end oscillators. Both radios however use a TCXO-based oscillator, therefore they both will have stability over temperature ranges.

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 sniffers before they deploy them.

E. Mobility of target device

For tracking, the BLE fingerprint of a mobile device should not vary as the target moves from one physical location to another. We carry our mobile devices (particularly smartphones) with us at all times. Consequently, the attacker's ability to track the BLE device may be affected by changing ambient environment (different physical location) and also by the speed of motion of target (walking, running, travelling in car etc.)

Physical location: A change in the physical location of the BLE device can cause SNR variation due to change in multipath conditions of the ambient environment. However, in our study we observed that this has minimal impact on the ability to track the BLE devices. In Section V-C, we observed a low FPR-FNR rate in tracking our 17 target devices, even though the location changed. Furthermore, Figure 12 and Figure 11 show the accuracy of classifying 20 ESP32 chipsets across a range of SNR values (10 – 30 dB). We observe that above a certain minimum SNR (1̃0 dB), the variation in SNR does not impact classification accuracy

Speed of Motion: A moving BLE device may experience an additional velocity-dependent frequency offset due to the Doppler effect [40] While this may cause a drift in the measured CFO of the BLE target device, the impact is not significant. For example, if the BLE transmitter at 2.4 GHz is moving at a velocity of ~~50 miles~~ 80 kilometers per hour and the receiver is stationary, the Doppler frequency offset is only about 180 Hz. In comparison, the mean of standard deviation of CFO for all devices we observe in our field study (Section V-C) is about 330 Hz Therefore even at relatively high speed motion, the Doppler shift shouldn't impact tracking ability.

Summary: Changing location or change in speed of motion of the BLE device has little to no impact in the attacker's ability to accurately fingerprint and identify the 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, these beacons have 15-minute stability in their anonymous identifier. We leverage these properties of BLE to perform a large-scale uncontrolled data collection and analysis of how severely misidentification errors manifest in real-world environments. In particular, we assess how useful BLE fingerprints are even though devices are unevenly unique, and fingerprints can be affected by temperature variations. We end with two case studies showing how well the end-to-end attack works in field, even over multiple days. To the best of our knowledge, this is the first uncontrolled experiment to evaluate the effectiveness of a physical-layer tracking attack.

Data Collection

We collected a large dataset of BLE beacons from uncontrolled wireless personal devices that happened to be running a BLE beaconing app. We chose to collect these data sets in public places[1] that are likely to contain a large number of BLE-enabled mobile devices: 6 coffee shops, a university library, and a food court. We ~~setup~~ set up a USRP N210 in each of these locations for approximately one hour, and opportunistically collected beacons in raw signal captures. Given the absence of ground truth, we relied upon the fact that the randomization of MAC addresses occurs every 15 minutes. This gave us periods of time ~~where~~ when we knew the packets received were from the same device, because it had a temporarily stable identifier. Even though we were analyzing captures from many real-world devices, we were careful to only analyze BLE beacons and their corresponding physical-layer fingerprint.

Ethical Considerations: We ensure that our data collection is completely passive. We only receive BLE advertisement packets that devices already broadcast with the intention of being received by other nearby devices. These broadcasts are sent by existing applications like contact tracing, device discovery, Continuity etc. To ensure we only capture these BLE packets, we only capture BLE advertisement frequencies and mask off non-advertisement channels. Furthermore, we ensure that in the decoding stage only undirected advertising packets are processed.

Moreover, the device fingerprints we collected in the field cannot be directly linked to individual people. The BLE packets we obtain these fingerprints are advertisement messages that do not reveal any personally identifiable information about the user of the transmitting device. The only devices that we performed full identification and tracking on were the 17 devices that we controlled ourselves.

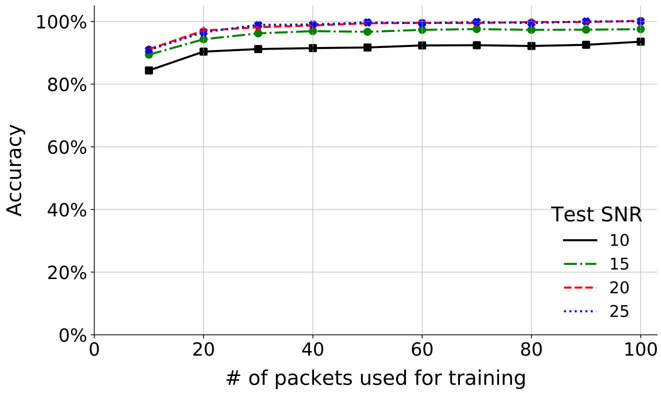

Fig. 11: Classification accuracy with different training sizes

Finally, we also discussed the details of this research with our IRB office, and were told that our experiments do not qualify as human subjects research.

Data Analysis Methodology

In order to analyze this dataset, we must first determine how many packets an attacker needs to receive from a device in to accurately perform the initial fingerprinting and eventual identification. To find this threshold, we performed a controlled experiment using a representative set of BLE transmitters, namely the 20 ESP32 chipsets. We tested 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 increases when the beacons have poor quality. We then classified each of the 20 devices for each chipset using the algorithm described in Section III-C2. We split the captures used for training and test as follows: 80% of the beacons were used for training, and 20% for testing. We then trained the classifier with all three SNR values in the set $\{10, 15, 25\}$ dB, and we ran tests with beacons that had $\{10, 15, 25\}$ dB SNR independently. We evaluated the training accuracy with a test size of 10 packets. Note that we are only looking for a conservative number of packets to use for classification, not an optimal one.

Figure 11 shows the training accuracy of classifying the devices compared to the number of packets used for building the fingerprint of devices. For all SNR values, having 50 packets for training is sufficient. As described in Section **??**, many BLE devices transmit significantly more than 50 beacons a minute. Consequently, we need to isolate a mobile device for at most 1 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 SNR values, it shows that 10 packets can accurately identify devices. In summary, for the field study, we use 50 packets to fingerprint a device, and 10 packets to identify a device.

The final component of our methodology is identifying what packets correspond to the same transmitter. The problem is, the MAC address of a device may change over due to MAC address randomization, causing us to treat the same physical-layer fingerprint as two devices. To mitigate this

[1] Data collection occurred prior to the outbreak of COVID-19.

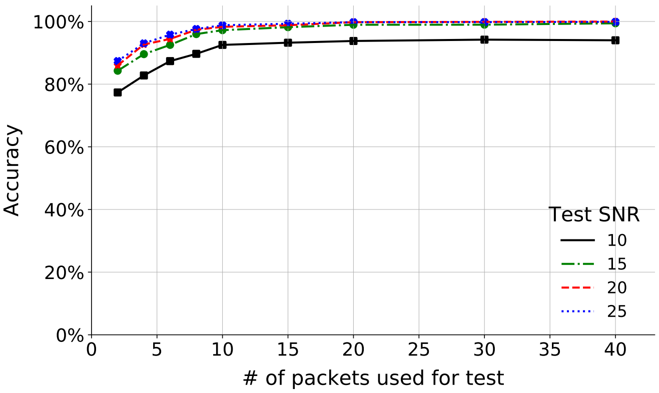

Fig. 12: Classification accuracy with different test sizes

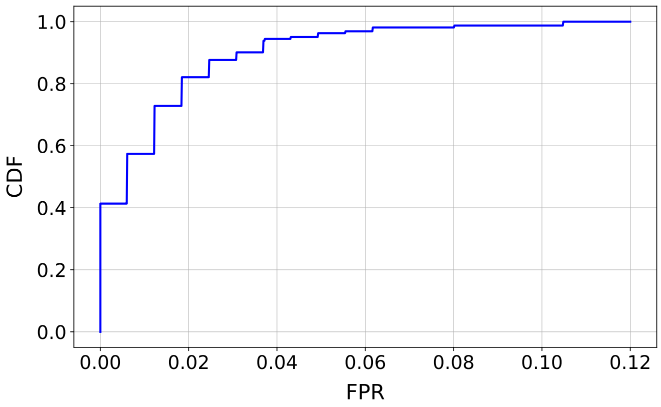

Fig. 13: Dist. of FPR a device when comparing with all others

Devices Considered	FPR (percent)	FNR (percent)
Apple Products	1.91	2.40
Not Apple Product	1.15	2.94
Apple vs Not Apple	0.15	–
All Devices	**1.21**	**2.53**

TABLE II: FPR and FNR comparison for Apple products and other devices.

problem, for the data we collect at a single location, we only consider devices that we have observed during a contiguous period of time when no device we observe changes its MAC address (and thus appears to stop transmitting). We combine the final set of 162 devices observed across all locations for our analysis.

A. *Uniqueness of devices in the wild*

In the following experiments, we evaluate the uniqueness of the devices we observed in the field. Namely, we evaluate the likelihood that we confuse a device that is a not our target, with our target (False Positive). First we select a threshold for the classifier by taking a device (MAC address) $i \in \{1, 2, 3, ..., 162\}$, we fingerprint it, then finding a threshold where the False Negative Rate (FNR) using the rest of the packets from device i is low (approximately 2%). Then, for each of the remaining devices, we use 10 packets and compute the metrics and compare with the fingerprint of i device. If it is identified as the device i, then it is considered as a false positive. The average of all these false positives is considered as the False Positive Rate (FPR) for device i. We compute the FPR for all $i \in \{1, 2, 3, ..., 162\}$ devices by repeating the same calculations.

Figure. 13 shows the distribution of FPR for all 162 devices. The median FPR is only 0.62. Moreover, we see for more than 40 percent of the devices, there is no confusion (zero FPR) even across a dataset as large as 162 other devices. These particular devices are those that have a unique and separable hardware imperfections; for instance, a relatively large CFO or IQ offset or IQ imbalance. Owning a device with one of those outlier imperfections is a serious privacy threat. On the other end of the spectrum, there are also devices with extremely commonly observed hardware imperfections; for instance both CFO and IQ offset are small. The FPR for these devices are even as high as 0.1.

Effect of device model: Based on our controlled experiments, devices from the same model are more likely to be confused than devices of different models. We used the technique proposed in [10] to distinguish Apple products' beacons from other devices. About 76 percent (123 devices) of the dataset are Apple products, making them the majority of all observed devices. This is likely because of Apple's default enabling of their Continuity service.[2]

We repeat the same FPR experiment as above, however, this time we break up the results based on whether the device is an Apple product or not in Table II. The FPR between Apple devices (first row) is greater than the average FPR we reported before and the FPR between Apple product and the devices that are not Apple products is very small (third row). The reason could be that as one might expect, the hardware imperfections of the devices from the same make are more likely to be close to each other than the devices with different make and model. In fact, the controlled experiments in Section IV-A also showed that the hardware imperfection distributions of devices from the same make and model could be similar.

~~What imperfections contribute~~ How each imperfection contributes to identification~~?~~: Finally, we evaluate what imperfections contribute to identification. Table III shows the FPR and FNR when using CFO, ~~IQ offset and IQ~~ I/Qoffset and I/Q imbalance separately and together by repeating the same experiment as before. CFO contributes the most as it can have a wider range of values for different devices compared to ~~IQ~~ I/Q imperfections. However, CFO is not sufficient to get the best result. ~~IQ~~ I/Q imperfections resolve the confusion in some cases in which the two device have similar CFO values and reduce the FPR from 2.42 percent to 1.21 percent. This also can be observed in our controlled lab experiments in Figure 6 where some devices have CFO values close to each other, but their difference in ~~IQ~~ I/Q imperfection helps us with distinguishing those devices.

Recall that temperature can cause variation CFO while it does not have any notable impact on ~~IQ~~ I/Q imperfections. As

[2] We collected this dataset before COVID-19 contact tracing launched.

Features used	FPR (percent)	FNR (percent)
Baseline CFO	7.41	3.49
CFO only	2.42	2.45
IQ offset only	19.84	2.39
IQ imbalance only	32.53	1.52
All Features	**1.21**	**2.53**

TABLE III: Separability of each feature alone. CFO contributes the most but IQ imperfections could further improve the identification.

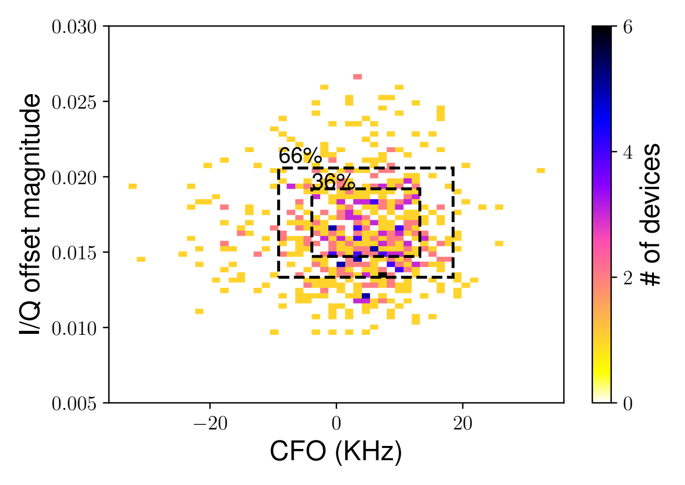

Fig. 14: Histogram of imperfections across 647 BLE devices.

a result, ~~IQ~~ I/Q imperfections can help in situations in which the target might experience temperature changes. Moreover, the FPR and FNR when only using CFO computed by our method, is better than FPR and FNR when use the baseline CFO computed by existing techniques described before. This significantly impacts the identification when we aim at identifying a device in the presence of a large set of other devices. In fact, as discussed in Section IV-A having a method to measure hardware imperfections with high precision is necessary to RF fingerprinting; otherwise the device with close fingerprints will be easily confused.

Uniqueness of imperfections in a large number of devices: Recall that across the 162 devices observed in our field evaluation dataset, we found ∼40% of the devices to be uniquely identifiable. Therefore, the hardware impairments like CFO and I/Qoffset and imbalance are unique enough to identity those devices. However, is natural to ask, is the same is 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 entry/exit of a hallway where *hundreds of different people* passed by each day. Each person passed by the SDR once, other than the people who worked at the facility. We recorded the Apple/Google COVID–19 Exposure Notification BLE beacons transmitted by the devices carried by these people. Our recordings were for a total of 20 hours on two days separated by one week to ensure there were as few duplicates as possible. We computed the average CFO and average I/Qoffset magnitude for each random BLE MAC address we observed in the beacons. To reduce 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 3 minutes[3].

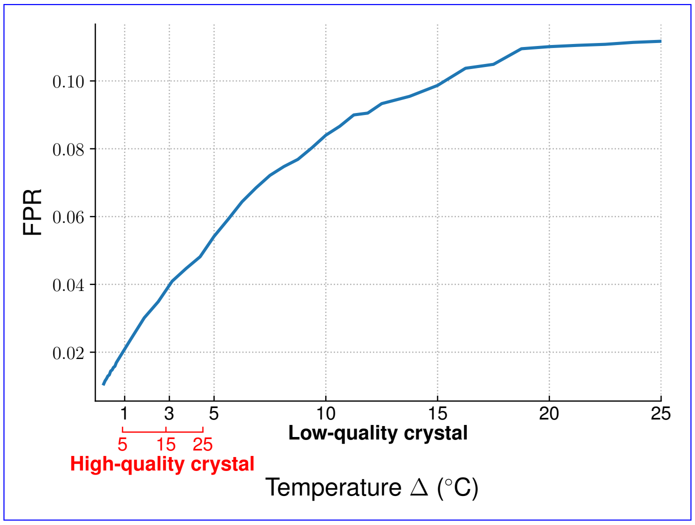

Fig. 15: The effect of crystal oscillator temperature change on FPR at a fixed FNR, for a high quality and a low quality crystal oscillator. The CFO of a low quality crystal with 8 minute cutting accuracy is significantly changed by temperature, resulting a drastic increase in FPR; while the same change in temperature has much less impact on a high quality crystal

We observed 647 unique MAC addresses that were observed for less than three minutes across the two 20 hours of data collection. Figure 14 shows the 2-dimensional histogram of the CFO and I/Qoffset magnitude of these devices. The number of histogram bins were chosen so that each bin represents a CFO range of twice the median of standard deviation (∼1.3 kHz) of devices observed in the earlier field evaluation. Devices that fall in the same bin are considered overlapped and indistinguishable from their hardware imperfections. Since manufacturing variations follow a normal distribution, we also show the bounds of the 2D imperfection histogram that cover 36% (∼σ) and 67% (∼2σ) of the devices. We found that 47.1% (305) of the devices were not similar to any other observed device. This confirms that even in a larger data set, hardware imperfections are uniquely distinguishable for ∼40% of devices. We also observed that devices with overlaps did not overlap with a significant number of devices. For instance, 15% (97) of the devices had similar imperfections with only one other device.

B. Temperature effects on identifiability

During the maximum of 15 minutes that we observe devices in the field, the temperature of some of these devices may change due to the change in activity level on the device (such as calling or opening apps on the phone), most devices will

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

maintaining roughly the same temperature. Consequently, the effect of temperature on the ability to identify the devices is not well-represented in the field data. We now evaluate what would be the effect of temperature changes on the ability to identify the devices in the field.

Recall from Section IV-Ai that temperature changes can significantly affect CFO of the device even up to several kHz, while it does not affect IQ imperfections noticeably. The impact of temperature on frequency drift on crystal oscillators is well documented, for instance [11] describes the Bechmann curve (frequency drift versus temperature) for popular AT-cut crystals for different crystal cutting accuracies. Generally speaking, three observations can be made from this curve. First, the frequency drift (which results in changing CFO) increases as the temperature changes more and more. Second, in the temperature interval that personal electronic devices may experience, frequency drift is linearly dependent on the temperature changes. Third, crystals with different cutting accuracies experience different amount of frequency drift for the same amount of change in temperature.

When the device temperature changes, the CFO of the device changes and the device will have a different CFO compared to when it was fingerprinted. Consequently, the device will not be identifiable resulting in false negative. To mitigate this, we should extend the CFO values that is expected to be received from the target device. For instance, assume $\Delta T°C$ temperature change causes Δf kHz change in CFO. Assume a device was fingerprinted in $T_0^\circ C$ and the CFO was recorded as f_0 kHz. In order to make sure (ignoring other factors that may affect RF fingerprints) that our identification system can tolerate up to $\Delta T°C$ temperature change, and we will identify the device if its temperature is anywhere between $[T_0 - \Delta T, T_0 + \Delta T]$, we must accept that any CFO value between $[f_0 - \Delta f, f_0 + \Delta f]$ kHz matches the CFO of our device. The consequence of extending the acceptable boundaries of CFO is that other devices with CFO in this range might be mistakenly identified as our target (if their IQ imperfections are also close to each other). This will increase the false positive rate (FPR) of our identification system. In fact, as ΔT increases, it will cause more change in CFO which forces us to extend the CFO boundaries in order to maintain the same false negative rate (FNR). By extending the CFO boundaries, there will be more and more devices out in the field that are not our target but their CFO fall inside the accepted CFO boundary of our target, resulting in higher FPR.

Now we use our field data to analyze the effect of temperature change on identifying the device. We follow the same approach as described in Section V-A. The difference is that this time we use the analysis discussed above to set the threshold or boundaries for the target. For any temperature change of $\Delta T°C$, we use the curves in [11] to find the corresponding change in CFO Δf. We set the threshold for CFO equal to Δf so that we make sure our identification system can identify the target if the temperature changes up to $\Delta T°C$. For other imperfections (IQ offset and imbalance), we chose the threshold the same way as before as they are not noticeably affected by temperature according to our controlled lab experiments. Figure 15 represents FPR for different values of ΔT for two crystal with different cutting accuracies. As mentioned earlier, temperature change causes less change in CFO for high quality crystals with high cutting accuracy compared to low quality crystals with low cutting accuracies. As a result, if we assume the device is using a high quality crystal, the FPR will increase less as we don't need to push CFO boundaries too much to handle possible changes in temperature. Figure 15 demonstrates this fact assuming a very high quality crystal with 0 minute cutting accuracy is used as well as what would happen if a very low quality crystal with 8 minute cutting accuracy is used.

According to Figure 15, the temperature can potentially increase FPR for a fixed FNR rapidly for low quality crystals and make the identification less accurate. This can affect the ability of attacker in some scenarios if the victims use their personal electronic devices to run heavy apps quite often or if the device was fingerprinted when its temperature was not in a normal temperature such as the room temperature. Furthermore, not that for low quality crystals, when the change in temperature is too much, CFO almost does not help in identification as we had to push the CFO boundaries to include almost all possible CFO values seen for the devices in the field. Consequently, the FPR is the same as if we only use IQ offset and IQ imbalance.

C. Case Study 1: Many targets

In this section, we conduct an experiment to demonstrate one of the scenarios that this attack might be deployed. We have 17 targets outside the set of devices collected in the wild. These targets are listed in Table IV that we want to identify. Each target is isolated in an office for a while to get at least 50 packets for training and 50 packets for the evaluation set. These targets are profiled using the training and evaluation data and the profile of each device is stored and used in the rest of the study to compute FNR and FPR.

Between 2-7 days later, we took these targets to a food court and turn on our sniffer 10 ft away from the targets. This data is used to compute the FNR for these 17 targets using the stored profile for each of the targets. We did not strictly force the targets to have the exact same temperature in the office and food court, but both environments were air-conditioned indoor buildings and there was a normal level of activity on the target devices. We also use the captures from the aforementioned coffee shop data (this time we don't filter out any MAC address, and we consider all MAC addresses even with a very small number of packets) in which none of the targets are present, to compute the FPR for these targets. The way that FNR and FPR are calculated is that, in each 10 seconds of the captures, we see if there is any MAC address (at least one MAC address) whose hardware imperfections matches the stored profile of a specific target. If there exist such MAC address, then we have identified that specific target device in that second. Then FNR and FPR are computed depending on whether that target was actually present there or not. In fact FNR here is the percentage of time a specific target device is there, but we think it's not, and FPR is the

Label: Device	Label: Device	Label: 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 other figures

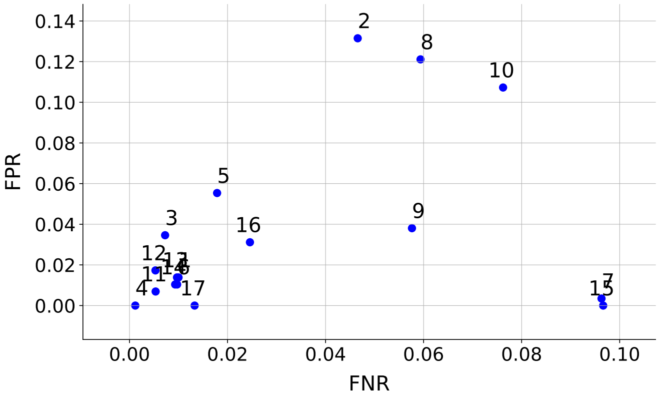

Fig. 16: FNR-FPR for 17 targets. Label numbers are assigned in Table IV.

percentage of time that a specific target device is not present, but we think it is.

Results: Figure 16 presents FNR-FPR for these 17 targets. The average FNR of these targets is 3.21 percent and the average FPR is 3.5 percent. As shown in the figure, although there are a few devices with high FNR and FPR, most devices have distinguishable and persistent hardware imperfections, resulting in low FNR and FPR. If an individual owns one of these well-identifiable device, then they could be in a serious threat of being identified by the attackers at their desired location.

Figure 17 demonstrates the false positive occurrences for all of our targets in the longest coffee shop capture. Each time there is a bump, it means that at least one device was detected as the corresponding target falsely. As we observe, most of the false positives continue for a while, possibly because a device with a similar hardware imperfections as the target entered the coffee shop and left after a while. There also exists a very few false positive occurrences that last for a few seconds. This is because sometimes we did not get enough packets from a device in 10 seconds and the hardware imperfections of the very few noisy packets looked like our target. However, after getting more packets from that device, it turns out we were wrong and false positive is resolved in the next time slots. Finally, its worth mentioning that on average, we saw 18 unique MAC addresses in each 10 seconds and overall we saw 259 unique MAC addresses during this 48 minute capture at a coffee shop.

D. *Case Study 2: Tracking one person*

In this section, we deploy and evaluate a typical scenario that our RF fingerprinting attack might take place. To obtain the target's BLE fingerprint, we (the attacker) get close to the target (a volunteer who uses an iPhone). We scan for nearby BLE devices using a commonly available BLE scanner phone app, and simultaneously capture raw signals of all BLE packets with an SDR. When we are close to the target, we record advertising address of the BLE device with the highest observed signal strength (our target's phone). We can then use this address to filter out target device's packets in our raw signal capture, and use those to derive the fingerprint.

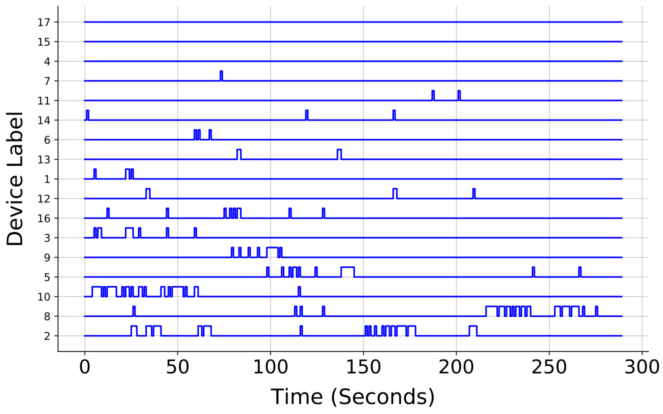

Fig. 17: Each line represents FPR occurrences over time for one of the 17 targets. Label numbers are assigned in Table IV.

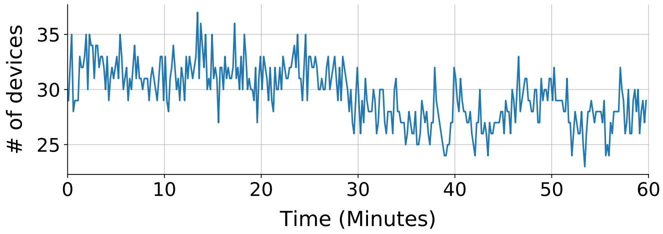

Fig. 18: Number of unique MAC addresses observed over time during the experiment of tracking Pixel 5

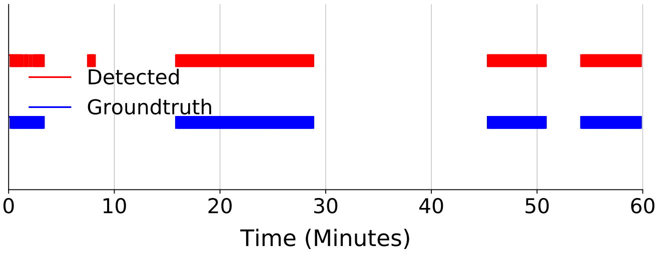

Fig. 19: The blue bar represents the time that the Pixel 5 target was present, and the red bar represents the time that our algorithm detected the presence of the Pixel 5

After obtaining the fingerprint we place our SDR close to the house (tracking location) of our target, and then track the target's presence at the location. We run this tracking for one hour, during which the person walks inside and outside the house 2 times. Figure 18 shows the number of unique MAC addresses observed during this time. As we count the number of MAC addresses in a short duration of every 10 seconds, this also represents the number of devices observed over time.

The blue bar shown in Figure 19 demonstrates the periods of time at which the person was inside the house and the signal of their phone was stronger than the power level threshold set by the sniffer. Our identification system makes an attempt every 10 seconds and the red bar in the same plot shows the time durations which our identification system thinks the person was present. The bars perfectly match except for immediately prior to the 10th minute, where we falsely detect the presence of the target for 50 seconds, even though it had not 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; as long as the device transmits the signal will have CFO and I/Q offset and imbalance.

A possible defense against this attack requires us to rethink the design of the BLE chipset's signal chain. Particularly, we envision a BLE transmitter circuit VCO that adds an extra frequency offset beyond the fixed offset due to the crystal oscillator. The extra frequency offset is time-varying and randomly changing during run-time. Consequently, the CFO measured at the receiver will also be time-varying and not stable. Since BLE has a large CFO tolerance (150 kHz [28]), an extra frequency shift will not impact the decoding of packet at the receiver. However, our attacker who relies on a stable CFO to track the target, will no longer be able to utilize these hardware impairments as a reliable identifier.

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], [24]. 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 [14], [24] and BLE [30], [32], [5], [23], [10] 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 attack is extremely limited, as it requires an attacker to persistently follow a target to track it; therefore, if the attacker misses a MAC randomization cycle it can no longer identify the target. 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, ~~IQ~~ I/Q offset/imbalance, signal transients and others [8], [38], [16], [21], [34], [22], [4], and leveraged various statistical methods, and in recent times deep learning approaches [15], [41], [25] to fingerprint these properties. For instance, transient portion of the signal has been proposed as a unique signature to classify different wireless devices [37], [12] even Bluetooth signals [17]. 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 [13]. 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 ~~IQ imperfections (IQ~~ I/Qimperfections (I/Q origin offset and ~~IQ~~ I/Q imbalance) are two well ~~recongized~~ recognized features which have been shown to be the most separable features for WiFi fingerprinting [8].

BLE hardware in mobile devices are similar in architecture and suffer from the same hardware impairments as WiFi radios. Despite that, ~~except~~ other than a few efforts at coarse ~~grained~~ CFO extraction utilizing specialized hardware (CC2400) [33], [39], 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 ~~IQ~~ 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 a realistic scenario.

VIII. Conclusion

In this work, we evaluated the feasibility of physical-layer location attacks on mobile devices with BLE. 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 features if they are developed with low power radio architectures. Evaluating the practicality of this attack in the field, particularly in busy settings such as coffee shops, we find that certain devices are particularly similar to others, and will often be misidentified, protecting their users. Other devices have extremely identifiable characteristics, and may even be identified if temperature shifts significantly. Overall, we found that BLE does pose a location privacy threat to mobile devices, but an attackers ability to track users is essentially a matter of luck, or that there are not many nearby devices.

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] 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.

[8] 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.

[9] 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/.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

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

[27] 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.

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

[29] 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.

[30] 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.

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

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

[37] 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.

[38] 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 &#38; Privacy in Wireless and Mobile Networks*, WiSec '16, pages 3–14, New York, NY, USA, 2016. ACM.

[39] 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.

[40] 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.

[41] 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.