BLE Location Tracking Attacks by Exploiting Frequency Synthesizer Imperfection

Yeming Li, Hailong Lin, Jiamei Lv*, Yi Gao, and Wei Dong*
The State Key Laboratory of Blockchain and Data Security
College of Computer Science, Zhejiang University
Email:{liymemnets, linhl, lvjm, gaoy, dongw}@zju.edu.cn

***Abstract*—In recent years, Bluetooth Low Energy (BLE) has become one of the most wildly used wireless protocols and it is common that users carry one or more BLE devices. With the extensive deployment of BLE devices, there is a significant privacy risk if these BLE devices can be tracked. However, the common wisdom suggests that the risk of BLE location tracking is negligible. The reason is that researchers believe there are no stable BLE fingerprints that are stable across different scenarios (e.g., temperatures) for different BLE devices with the same model. In this paper, we introduce a novel physical-layer fingerprint named *Transient Dynamic Fingerprint* (TDF), which originated from the negative feedback control process of the frequency synthesizer. Because of the hardware imperfection, the dynamic features of the frequency synthesizer are different, making TDF unique among different devices, even with the same model. Furthermore, TDF keeps stable under different thermal conditions. Based on TDF, we propose *BTrack*, a practical BLE device tracking system and evaluate its tracking performance in different environments. The results show *BTrack* works well once BLE beacons are effectively received. The identification accuracy is 35.38%-57.41% higher than the existing method, and stable over temperatures, distances, and locations.**

I. Introduction

Bluetooth Low Energy (BLE) [1] has become one of the most popular wireless protocols because of its cheap, low-energy, and wide adaptation nature. It is common for users to carry one or more BLE devices nowadays. According to the Special Interest Group (SIG) research [2], the annual shipments of BLE-compatible devices exceed 4 billion in 2022. BLE devices continuously transmit beacons to support applications and services (e.g., Apple iBeacon [3] and Google Eddystone [4]). Each BLE device can broadcast 77-872 beacons per minute [5].

However, by their nature, BLE beacons have the potential to introduce significant privacy risks. Therefore, the topic of BLE device tracking and its countermeasures are extensively studied in the literature. Adversaries can achieve BLE device tracking by eavesdropping that BLE beacons [6], [7]. To solve this, manufacturers have adopted the MAC address randomization strategy to maintain anonymity [8]–[10]. The BLE advertisers periodically change their MAC addresses (10min-15min), so the adversary is hard to identify whether the received beacon is advertised by the target or not. Approaches based on software fingerprints [11], [12] can bypass the MAC randomization, they typically require a very long eavesdropping time (up to hours) and can be fixed with software updates. Many recent approaches use physical-layer fingerprints because they have the potential to compromise all the software-level security mechanisms [13], [14]. The physical-layer fingerprints are caused by manufacturing imperfections, so they can not be easily fixed or modified. There are three main types including (1) Fingerprints based on Carrier Frequency Offset (CFO) [14], [15]. (2) Fingerprints based on I/Q imperfections [16]. (3) Fingerprints based on Transient delay [13]. Despite all these efforts, BLE location tracking attacks are still considered unreliable and impractical, as illustrated by a recent empirical study [5]. The authors in [5] point out that (1) the CFO is hard to distinguish devices with the same model and it is sensitive to different thermal conditions [5], [17]. (2) The I/Q imperfection only exists in I/Q modulation architecture, but most of the BLE-only devices use frequency synthesizer-based modulation architecture, which inherently has no I/Q imperfection. (3) The transient delay is identical among devices with the same model. To date, the common wisdom is that BLE location tracking is not practical and the risk is negligible [5].

This paper intends to answer one *essential* question, i.e., *is BLE location tracking still possible?* Our study reveals that we can extract robust physical-layer fingerprints from imperfections in the frequency synthesizer which is a prevalent component in all kinds of BLE chips [18], [19]. Specifically, the frequency synthesizer is a negative feedback control system that is used to generate target frequency, and hardware imperfections can affect the dynamics of the control system (e.g., response time, overshoot). We name the novel fingerprint as *Transient Dynamic Fingerprint* (TDF). Although frequency synthesizer imperfection is well recognized by the hardware community, its use for physical-layer fingerprinting and device tracking has long been *overlooked* in the research community. The reason is two-fold. First, it is hard to extract stable fingerprints from the frequency synthesizer. The reason is it requires fine-grained output frequency variation during the transient delay (14μs-32μs). Owing to the inherent time-frequency resolution constraints of the Fourier Transform, it is infeasible to capture the precise relationship of frequency variations with time. Second, it is time-consuming to process the TDF fingerprints for all BLE beacons received from crowded

This work is supported by the National Natural Science Foundation of China under Grant No. 62072396 and 62272407, the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under grant No. 2023C01033, the National Key R&D Program of China (2020YFB1313501), and the National Youth Talent Support Program. Jiamei Lv and Wei Dong are the corresponding authors.

advertising channels.

In this paper, we first study the impacting factors and properties of TDF. TDF is related to several electronic components in the frequency synthesizer, including various hardware imperfections. We find that TDF is *unique* even among devices with the same model. Besides, TDF is independent of the temperature-sensitive crystal oscillator [17], and TDF is relatively *stable* with varying temperatures. We then propose *BTrack*, a BLE device identification system for user's location tracking. *BTrack* incorporates novel techniques to address the challenges we have described earlier. First, we propose a robust fingerprint extraction method based on the phase change of the transient signal (instead of Fourier Transform) in order to capture the fine-grained frequency variations with time. A DNN-based classifier is then used to extract TDF from the phase change. Second, to achieve device tracking in a timely manner, we propose a bi-variate model filter to effectively filter out those beacons transmitted by devices of different models from the target device.

We implement *BTrack* with USRP B210 and evaluate with 24 popular BLE chips in four different environments. In total, over 39,000 signal traces are collected for evaluation. *BTrack* works well (82.93%-97.45% detection rate) once the BLE beacons are effectively received. Compared with the state-of-the-art approaches, *BTrack* can achieve 35.38%-57.41% higher identification accuracy and the performance is stable over varying temperatures, distances and environments. To the best of our knowledge, this is the first work to exploit frequency synthesizer imperfections for device identification and location tracking. Our contributions can be summarized as follows:

- We propose a novel physical-layer fingerprint called TDF, and fully study its properties, uniqueness, and stability.
- We propose *BTrack*, a practical device identification system for location tracking, illustrating that the TDF can pose a great threat to users' location privacy.
- We implement *BTrack* and open source our code and dataset[1], and extensive experiments are conducted.

II. Background and Preliminary

In this section, we first show the BLE location tracking attacks threat model. Then, we introduce the preliminary of the BLE and discuss the reason why the existing BLE fingerprint is not feasible in some conditions.

A. Location Tracking Threat Model

Consider a scenario that an adversary wants to track someone, detecting exactly when they are present at a specific location. The target user may carry BLE devices (e.g., earphones, wearables), which continuously advertise BLE beacons (77∼872 beacons per minute [5]). The adversary is equipped with an SDR to capture the raw signal of BLE beacons. First, it collects some beacons transmitted by the target BLE device in an isolated environment. Then, it extracts physical-layer fingerprints that can uniquely represent the target device. After that, the adversary can deploy BLE sniffers at interest locations to capture BLE beacons. Upon receiving a BLE beacon, the sniffers extract the physical-layer fingerprint. If one or more BLE beacons match the physical-layer fingerprint of the target BLE devices, the adversary can infer the presence of the target user at the corresponding location.

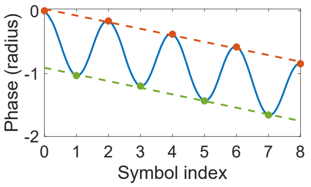


Fig. 1: CFO measurement.

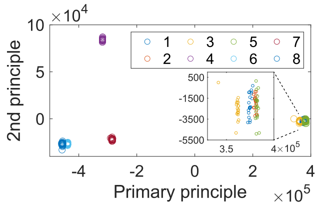


Fig. 2: PCA result of CFO.

B. BLE Preliminary

BLE utilizes Gaussian Frequency Shift Keying (GFSK) modulation [1]. The GFSK is a form of continuous phase modulation and the base-band signal in the I/Q form is $S_{\text{tx}}(t) = e^{j\phi(t)}$. Where the $\phi(t)$ is the phase of the signal at time t. For transmission of bit 1, the frequency is positive and the phase increases, and vice versa.

Then, we introduce three well-established physical layer fingerprints and outline their respective limitations:

Carrier Frequency Offset (CFO): The CFO is the frequency difference between the carrier frequencies of the transmitter and the receiver. Before transmission, the base-band signal is loaded on the carrier wave with frequency f_{ctx} and the BLE signal over the air is $e^{j[\phi(t)+2\pi f_{\text{ctx}} t]}$. At the receiver side, the receiver first removes the carrier wave according to its local oscillator with frequency f_{crx}, and the received base-band signal is $e^{j[\phi(t)+2\pi f_{\text{CFO}} t]}$, where $f_{\text{CFO}} = f_{\text{ctx}} - f_{\text{crx}}$. Since the hardware imperfection, the carrier frequencies of each device are different. The center frequency of the received base-band signal contains a frequency bias rather than zero, which is known as the CFO. CFO has been widely used in IEEE 802.15.4 and WiFi device identification [20]–[22]. BLE exhibits a wide tolerance for CFO (up to 150 kHz) and there have been related works that utilize CFO for Bluetooth device identification [13], [14].

However, the CFO has two shortcomings that limit its usage in practical BLE device identification. First, CFO is sensitive to thermal conditions [5]. When the room temperature or the computation workload of the BLE devices changes, the CFO of BLE devices will change as well and cause identification failure. Second, CFO is incapable to identify different devices of the same model. We apply a preliminary experiment to illustrate that. We use the preamble to calculate the CFO. The preamble over the air is 0b01010101 and the signal phase during the preamble is shown in Fig. 1. With the CFO, all peaks and troughs are linear increases or decreases. We perform linear fits to the peaks and troughs, respectively. The average slope a indicates the phase error induced by the CFO, which can then be calculated as $a/2\pi$. We collect the CFO of eight nRF52840 chips [23]. The CFOs on 40 BLE

[1] https://github.com/sada45/BTrack.

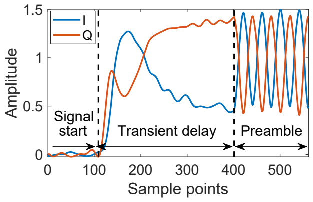
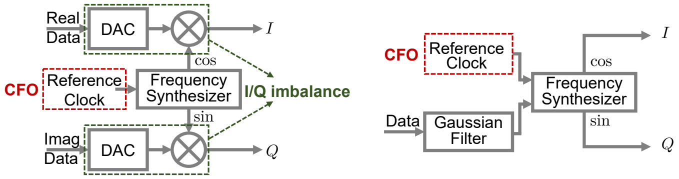


Fig. 3: Signal amplitude.

Chips	Avg. Transient Delay (μS)	Standard Deviation
nRF52840-1	14.80	1.59
nRF52840-2	14.77	1.75
nRF52840-3	14.63	1.67
CC2650-1	24.76	0.96
CC2650-2	24.79	0.72
CC2650-3	24.90	0.50

TABLE I: Start-up delay.



(a) I/Q modulation-based

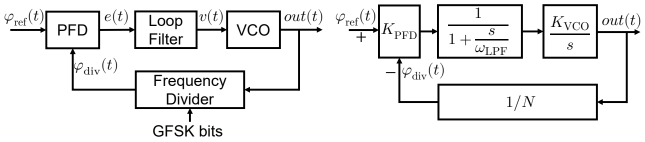
(b) Frequency synthesizer-based

Fig. 4: Two architectures of BLE RF front-end.

channels are considered as a feature vector of a device. Fig. 2 shows the result of Principal Component Analysis (PCA). Four nodes are gathered together, and two of them are totally overlapped. Utilizing CFOs of all 40 channels would not provide significant benefits as the explained variance ratio of the primary principal component is 98.89%.

Transient delay: There is a delay between the signal amplitude rising and the start of BLE data transmission. Fig. 3 shows a raw signal sample collected from a nRF52840 chip with 20MHz sampling rate. The existing work [13] takes the transient delay as a physical-layer fingerprint to identify BLE devices. However, the transient delay is different among different models but similar between devices with the same model. Tab. I shows the average transient delay and standard deviation of six BLE chips. Given that the standard deviation is significantly larger than the average transient delay difference, it becomes difficult to discern whether the variations between the measured and average values are attributable to fluctuations or disparities between different devices. Therefore, the transient delay can not be used to identify individual devices with the same model.

I/Q Imperfection: There are DC value and phase differences between the in-phase (I) and quadrature (Q) paths in the BLE modulation architecture. A straightforward architecture is shown in Fig. 4(a). The I and Q paths are separately controlled to generate the signal. Because of the imperfection of the Digital-Analog Converters (DAC) and mixers, the I and Q signal have different DC values and phases. This architecture is widely used for WiFi/BLE combo chips [5]. However, the GFSK is not an inherently I/Q modulation method. The frequency synthesizer-based modulation architecture (Fig. 4(b)) is a more stable and cost-efficient choice. Data bits directly control the frequency synthesizer to generate the target frequency in real-time and directly transmit the I and Q paths. Since there is no DAC and mixer needed in this architecture, it has no I/Q imperfection and the cost is lower. This modulation architecture is wildly used in BLE chips, especially for BLE-only chips [19], [24], [25]. Therefore, the I/Q imbalance can not be used to identify the BLE chips with frequency synthesizer-based modulation architecture.


$\varphi_{\text{ref}}(t)$
PFD
$e(t)$
Loop Filter
$v(t)$
VCO
$out(t)$
$\varphi_{\text{div}}(t)$
Frequency Divider
GFSK bits
$\varphi_{\text{ref}}(t)$
+
K_{PFD}
$\frac{1}{1+\frac{s}{\omega_{\text{LPF}}}}$
$\frac{K_{\text{VCO}}}{s}$
$out(t)$
−
$\varphi_{\text{div}}(t)$
$1/N$

(a) Functional block diagram.

(b) Transfer function.

Fig. 5: The architecture of frequency synthesizer, which is a negative feedback control system.

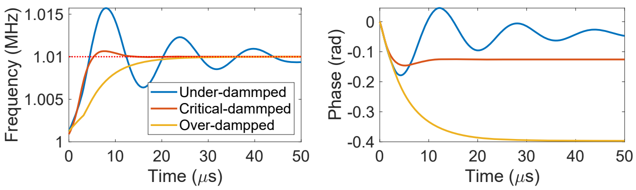
III. Fingerprinting with Frequency Synthesizer Imperfection

In the paper, we propose a novel physical-layer fingerprint named Transient Dynamic Fingerprint, which is caused by frequency synthesizer imperfections. The frequency synthesizer is a negative feedback control system that is used to generate target frequency. The output frequency variations show the dynamics of the control system, which are affected by hardware imperfections and unique among different devices even with the same model. In this section, we start by introducing the basic architecture of the frequency synthesizer. Then, we show the uniqueness of TDF. After that, we show this fingerprint is stable under different thermal conditions.

A. Frequency Synthesizer Modeling

The frequency synthesizer is used to generate a range of frequencies from a single reference frequency. The most common ones are Phase Locked Loop (PLL) based frequency synthesizers. It is a negative feedback control system and the functional diagram is shown in Fig. 5(a). A typical frequency synthesizer consists of a Phase Frequency Detector (PFD), loop filter, Voltage Control oscillator (VCO), and a frequency divider. The frequency divider is in the feedback loop and the frequency of the output signal $f_{\text{out}}(t)$ is divided with N to get the divided frequency $f_{\text{div}}(t)$. The divided signal is then input into the PFD with the reference signal. The PFD generates an error signal that can be presented as $e(t) = K_{\text{PFD}}(\varphi_{\text{ref}}(t) - \varphi_{\text{div}}(t))$. Where the K_{PFD} is the PFD gain, $\varphi_{\text{ref}}(t)$ and φ_{div} are the phase of the reference signal and divided signal, respectively. The loop filter is used to remove the noise, which can maintain stability in the loop and improve the overall performance of the synthesizer. The output of the loop filter $v(t)$ is the control voltage of the VCO, and the output frequency and phase of VCO are $f_{\text{out}}(t) = f_{\text{free}} + v(t)K_{\text{VCO}}$ and $\varphi_{\text{out}}(t) = \int_0^t f_{\text{out}}(\tau)d\tau$. Where f_{free} is VCO free-running frequency, K_{VCO} is VCO gain.

Before transmitting BLE data, the chip should first let the frequency synthesizer generates the stable carrier frequency f_c. This is exactly what the RF chip does during the transient delay. First, the divisor of the frequency divider is set to $N = f_c/f_{\text{ref}}$. Then, the frequency synthesizer circuit is closed and a negative feedback control process starts. When the frequency reaches the steady state, the output frequency equals the carrier frequency, and the VCO control voltage should be $(f_c - f_{\text{free}})/K_{\text{VCO}}$. We further analyze how the frequency synthesizer reaches the steady state after the circuit is turned on. Fig. 5(b) shows the transfer function of the frequency



(a) Frequency changes

(b) Phase changes

Fig. 6: Simulate results of frequency synthesizer dynamics with different damping factors.

synthesizer in the S-domain. The PFD and frequency divider are linear components, therefore, the transfer functions are K_{PFD} and $1/N$, respectively. The loop filter is a Low-Pass Filter (LPF) and we take the first-order LPF for example. The transfer function can be presented as:

$$H_{\text{LPF}}(s) = \frac{1}{1 + \frac{s}{\omega_{\text{LPF}}}}.$$

Where the ω_{LPF} is the angular frequency of the cut-off frequency of the loop filter. The output phase of VCO is the integral of its output frequency and its transfer function is:

$$H_{\text{VCO}}(s) = \frac{K_{\text{VCO}}}{s}.$$

Therefore, the transfer function of frequency synthesizer is:

$$\begin{aligned} H(s) &= \frac{K_{\text{PFD}} H_{\text{LPF}}(s) H_{\text{VCO}}(s)}{1 + \frac{1}{N} K_{\text{PFD}} H_{\text{LPF}}(s) H_{\text{VCO}(s)}} \\ &= \frac{\omega_{\text{LPF}} K_{\text{PFD}} K_{\text{VCO}}}{s^2 + \omega_{\text{LPF}} s + \frac{\omega_{\text{LPF}}}{N} K_{\text{PFD}} K_{\text{VCO}}}. \end{aligned} \tag{1}$$

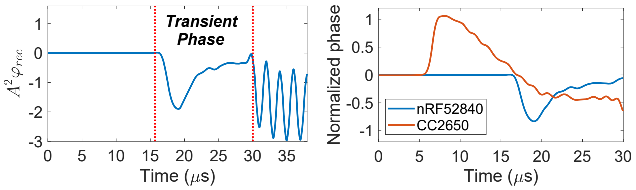
It can be presented as a second-order oscillator equation:

$$H(s) = N \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \tag{2}$$

Where ω_n is the natural frequency of the system, and ζ is the damping factor. To simplify, we consider the first N as a linear component that will not change the non-linear features of the frequency synthesizer. ω_n and ζ can be formally presented as:

$$\omega_n = \sqrt{\frac{1}{N}\omega_{\text{LPF}} K_{\text{PFD}} K_{\text{VCO}}},\ \zeta = \frac{1}{2}\sqrt{\frac{N\omega_{\text{LPF}}}{K_{\text{PFD}} K_{\text{VCO}}}}. \tag{3}$$

The damping factor is a crucial characteristic of control systems, describing the system dynamics. More specifically, it shows the vibration attenuation when this system is perturbed. We apply a simulation that a frequency synthesizer that generates 1.01MHz target frequency using a VCO with 1MHz free running frequency. Fig. 6(a) shows the variation of frequency over time. While the damping factor is small, the system is under-damped and the frequency will fluctuate and slowly stabilize to the target frequency. As for the over-damped system, though there are no fluctuations, it takes a long period to get to the target frequency. Manufacturers want frequency synthesizers to work as the critical-damped system. Therefore, the frequency synthesizer can quickly reach the target frequency. Manufacturers will set a fixed transient delay for each model according to the typical time required for frequency synthesizers to reach the stable stage.



(a) Amplitude-weighted phase

(b) Transient phase

Fig. 7: The received transient phase.

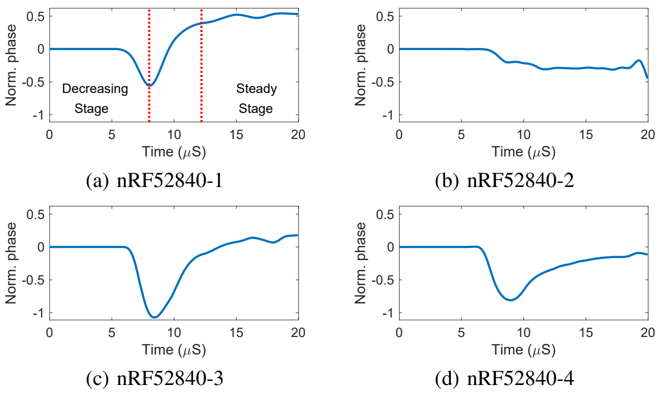
B. Transient Dynamics Fingerprint

In this section, we illustrate that the TDF for each device is unique and can be used for device identification. According to Eq. 3, the damping factor is related to ω_{LPF}, K_{PFD}, and K_{VCO}. Because of the hardware imperfection, the parameters of each functional block in the frequency synthesizer differ among devices, leading to different damping factors. Therefore, the dynamics of the frequency synthesizers can be different. Our key idea to identify BLE chips is extracting the dynamics features of the frequency synthesizer during the transient, we name it the TDF. To extract the TDF, we should first acquire the output of the VCO. An intuitive way is applying Short-Time Fourier Transform (STFT) to get the output frequency of the VCO with time like Fig. 6(a). However, the STFT exhibits a trade-off between frequency and time resolution. More specifically, with a larger STFT window size, the frequency resolution is high but unable to capture the rapid changes in the signal over time. With a smaller window, the time resolution is high but unable to capture the small fluctuations of frequency. Therefore, it is impossible to get the fine-grained frequency changes versus time. Inspired by the BLE demodulation method, we utilize the phase change to extract the TDF. At the receiver side, the received phase is:

$$\varphi_{\text{rec}}(t) = \int_0^t 2\pi(f_{\text{out}}(\tau) - f_{\text{crx}})d\tau. \tag{4}$$

Fig. 6(b) shows the simulation result of the received signal phase while the carrier frequency of the receiver is 1.01MHz. The received phase precisely reflects the frequency changes of the VCO output frequency, and we call the phase change during the transient delay as the *transient phase*.

To get the transient phase, we first measure the CFO with the BLE preamble and compensate each raw signal. After that, we get the phase change during the transient delay from the raw I/Q signal. A straightforward method is calculating the phase with $\varphi_{\text{rec}}(t) = \arctan(Q(t)/I(t))$. However, when there is no signal in the channel, even the noise with a small amplitude can cause severe fluctuations in the phase. Therefore, we calculate the amplitude-weighted phase $A^2\varphi_{\text{rec}}$ (Fig. 7(a)), where A is the signal amplitude. Then, since the amplitude of BLE signals decays with distance, a normalization is required. We set the unit scale of normalization to the absolute value of amplitude-weighted phase difference caused by one symbol in the BLE preamble. Fig. 7(b) shows the transient phases of nRF52840 and CC2650 on the BLE channel 37 (2.402GHz). The phase of nRF52840 initially



(a) nRF52840-1

(b) nRF52840-2

(c) nRF52840-3

(d) nRF52840-4

Fig. 8: Transient phases of different nRF52840 chips.

TABLE II: Feature vectors of nRF52840 chips.

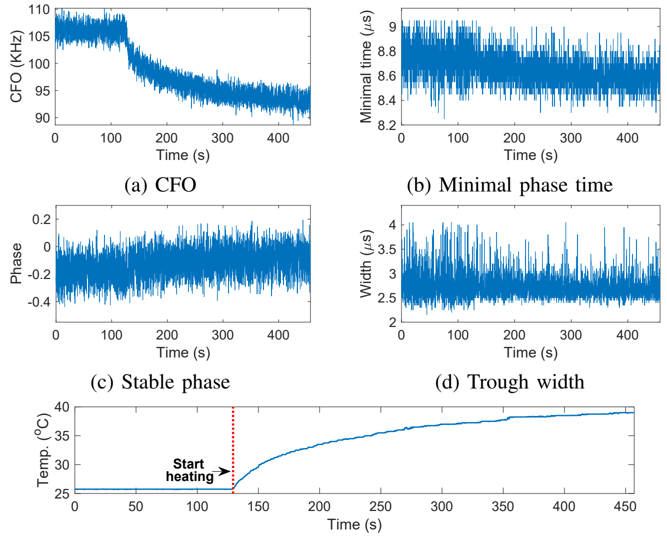
Chip	Min. phase	Min. phase time	Stable phase	Overshoot	Trough width
nRF52840-1	-0.54	7.98μS	0.60	-1.13	2.43μS
nRF52840-3	-1.00	8.53μS	0.16	-0.25	2.59μS
nRF52840-4	-0.82	8.78μS	-0.13	0.16	2.56μS

decreases because the free-running frequency of its VCO is lower than the receiving frequency. In contrast, the CC2650 exhibits the opposite behavior as the free-running frequency of its VCO is higher. The phase changes of different models are totally different, and the TDF is significantly different. So it would be easy to identify different models with TDF. We focus on the question that, whether TDF can be used to identify devices with the same model.

Fig. 8 show the transient phase of four nRF52840 chips. The hardware imperfection of the second one is significant, it has become an over-damped system. The others are all critical-damped systems but they exhibit noticeable differences in their specific details. We extract five features to describe the frequency synthesizer dynamics as the TDF. First, for the decreasing stage, we take the minimal phase and the corresponding time. Second, we take the average phase of the stable stage as the stable phase. Then, we extract the overshoot, which is an important feature to measure the dynamics of a control system [26]. We take the ratio of the stable and the minimal phase as the overshoot. Tab. II shows the TDFs of those three critical-damped chips. There are significant differences between them and it is possible to identify devices with the same model based on the TDF.

C. Temperature Stability

Although the carrier frequency with CFO is generated by the frequency synthesizer, it is not caused by the frequency synthesizer itself. When the frequency synthesizer reaches the stable stage, the output frequency should always be the Nf_{ref}, while the N is fixed for each target frequency. The CFO originated from the inaccuracy of the reference frequency, which is generated by the crystal oscillator. The crystal oscillator is highly sensitive to thermal conditions changes [17], [27]. So the CFO is sensitive to the temperature. Eq. 2-3 shows that the transfer function and damping factor are independent of the reference signal. Therefore, TDF is more stable with temperatures than CFO-based fingerprints.



(a) CFO

(b) Minimal phase time

(c) Stable phase

(d) Trough width

(e) Temperature

Fig. 9: Feature stability with temperature changes.

We further apply a preliminary to illustrate that. We extract TDFs by placing an nRF52840 chip in a room with 26°C room temperature. A heating plate is put on the device to change the chip temperature. We record the chip temperature with the internal temperature sensor. Fig. 9 shows the CFO, minimal phase time, stable phase, trough width, and chip temperature with the time. The CFO suffers from serious changes when the temperature changes. The average and standard deviation of CFO at the first 100s are 106.07KHz and 1.40, respectively. But for the last 100s, they are 93.528KHz and 1.34. The TDF (i.e., Fig. 9(b,c,d)) have minor changes with the temperature. For example, the standard deviation of the trough width is 0.345 and the difference between the average values before and after heating is only 0.098. Therefore, the TDF is more stable to the temperature changes than the CFO.

IV. System Design

We have shown that TDF can be used for device identification and is stable to temperature changes. There are two major issues that need to be resolved before the TDF can be applied in practice. First, due to the high density of BLE devices operating in the wireless channel, the receiver can receive numbers of BLE beacons from other devices, which significantly increases the computational overhead. Second, there are significant differences in the transient phases of different chip models, and it is tedious to manually analyze and extract the TDF for each model. To solve these, we propose *BTrack*, a practical device identification system with device TDF. In this section, we provide an overview of the *BTrack* system design, followed by a detailed introduction to the significant system modules.

A. System Overview

In Fig. 10, we show the bird-eye view of *BTrack*'s architecture. The BLE device identification involves four key stages:

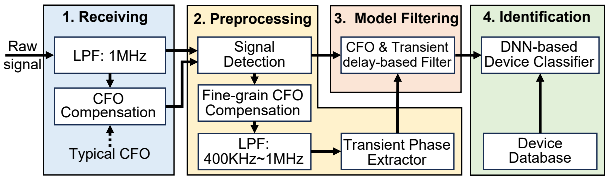


Fig. 10: System Overview of *BTrack*.

(1) Signal receiving: *BTrack* uses an SDR to collect the raw signal with 20MHz sampling rate. To make *BTrack* successfully receive BLE beacons, a low pass filter with 1MHz cut-off frequency is applied to remove the high-frequency noise. Then, *BTrack* compensates the raw signal with the typical value of the target device model.

(2) Preprocessing: This module extracts the CFO, transient delay, and transient phase. *BTrack* detects the BLE preamble. With the preamble, the accurate CFO for each BLE packet can be estimated, and a fine-grain CFO compensation can be applied. After that, *BTrack* can apply another low-pass filter with a lower cut-off frequency since the CFO is eliminated. It is worth mentioning that, a lower cut-off frequency can remove more noise and increase the identification range, but some of the TDF can also be filtered out. An appropriate cutoff frequency should be selected. Finally, the transient phase can be extracted with the method mentioned in Sec. III-B.

(3) Model filtering: Due to the high density of BLE devices operating in the wireless channel, *BTrack* can receive a large number of packets from non-interest devices. This module filters out those BLE beacons transmitted by devices of different models from the target device. Since the CFO and transient delay are significantly different among models, the model filter uses them to identify the model of the devices. We illustrate the details in Sec. IV-B

(4) Identification: The classification model takes the transient phase as input and decides whether this sample is transmitted by the target devices. The details are presented in Sec. IV-C.

B. Bi-variate Model Filter

According to our preliminary experiments in Sec. II-B, the chips with the same model have identical transient delays. As for CFO, the values are different but similar among individual devices with the same model. The basic idea of the front filter is utilizing the transient delay and CFO to implement a chip model-level classifier. If the received signal exhibits significantly different transient delay and CFO, the signal can be ignored. An intuitive way to achieve this is to calculate the 2-D distance of transient delay and CFO between the received signal and the average values of the target model. However, we find that the CFO of the same model devices can form multiple clusters, it is improper to use one average value to cover all devices. For example, in all of our nRF52840 chips, there exist Rev.0 (QIAAC0) and Rev.1 (QIAAD0). The average CFO of Rev.0 and Rev.1 are 108.60KHz and -10.67KHz, respectively. The standard deviations are 3.58 and 8.92. In Fig. 2, the four chips on the right are Rev.0, and the left four are Rev.1.

To solve this, we propose a bi-variate model filter. It generates the profiles of transient delay and CFO for each chip model. Each transient profile is presented as $[\mu_\mathrm{t}, \sigma_\mathrm{t}]$, where μ_t is the average transient delay, and σ_t is the standard deviation. The CFO can have multiple clusters, so the CFO profile is $[(\mu_\mathrm{c}^1, \dots \mu_\mathrm{c}^n), (\sigma_\mathrm{c}^1, \dots \sigma_\mathrm{c}^n)]$, where μ_c^i and σ_c^i are the average CFO and standard deviation of i-th cluster. We use the mean shift algorithm to cluster the device CFOs. The reason is that the mean shift does not need the user to specify the number of clusters. By setting the radius of the kernel (20 in the experiment), it can automatically learn the number of clusters present in the data. Once a signal has arrived, *BTrack* first calculates the 1-D Mahalanobis distance with the transient delay profile. If the threshold is larger than the threshold (3 in the experiment), the signal is discarded. Otherwise, *BTrack* will further calculate the Mahalanobis distance between each cluster of CFO. The filter has a larger threshold of the CFO since it is sensitive to different thermal conditions (i.e., 10 in the experiment). If the signal belongs to any of the CFO clusters, it will be recognized as being transmitted from a device of the same model as the target.

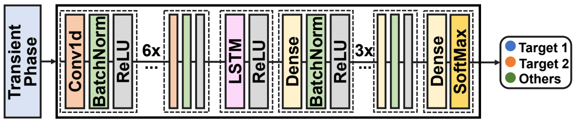


Fig. 11: Architecture of DNN-based classifier.

C. Device Identification

We have extracted some features which are enough to identify nRF52840 in Sec. III-B. However, we still use a DNN-based classifier to identify devices with the transient phase. The reason is three-fold. First, the trend of transient phases has significant differences across different models, the users have to re-analyze the transient phase of each model to extract model-specific TDFs, decreasing the scalability of the system. For example, in Fig. 7, the transient phase of CC2650 and nRF52840 are totally different. The minimal phase and the corresponding time used to identify nRF52840 are incompatible with CC2650. Second, it is tedious to extract general features for each model, since there are some anomalous devices with significantly different transient phases in the same model (e.g., nRF52840-2 in Fig. 8). Third, our key idea is extracting TDFs from the transient phase to infer the hardware imperfections of frequency synthesizers. Neural networks can comprehensively explore and extract pertinent features, thereby enhancing the identification performance.

Fig. 11 shows the architecture of our DNN-based classifier. We use seven-layer CNNs and one LSTM fully exploit the spatial and temporal features of the input, followed by four dense layers and one SoftMax layer to output the labels of the input sample. To train the DNN model, two sets of transient phase samples are required. The first one is collected from the target devices. The second one is acquired from the device database, and all of the samples are labeled as “others”. The traces in the database is pre-collected from other devices with the same model. The “others” set should be diverse and cover as many potential cases as possible. Besides, it should not include any similar devices in the “target” set. However, it is

Chip model	Target/ Others/ Alien
Nordic nRF52840	2/6/4
Dialog DA14580	2/4/1
TI CC2650	1/3/1

TABLE III: Chip models.

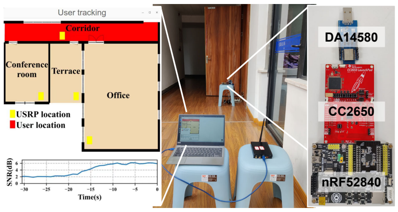


Fig. 12: The setup for the evaluation.

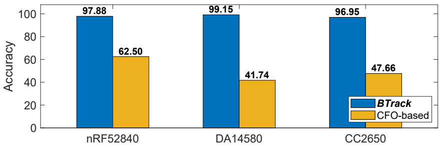
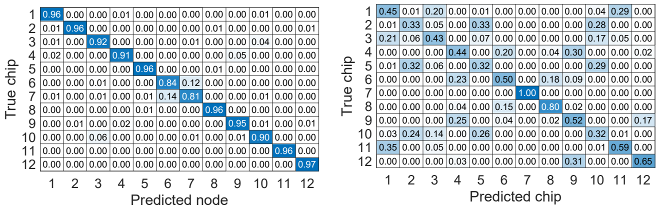


Fig. 13: Identification accuracy.



True chip \ Predicted node	1	2	3	4	5	6	7	8	9	10	11	12
1	0.96	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00
2	0.01	0.96	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01
3	0.01	0.00	0.92	0.00	0.00	0.00	0.01	0.00	0.00	0.04	0.00	0.00
4	0.02	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.96	0.00	0.00	0.01	0.00	0.00	0.00	0.00
6	0.00	0.00	0.01	0.00	0.00	0.84	0.12	0.00	0.00	0.00	0.00	0.00
7	0.01	0.00	0.01	0.00	0.01	0.14	0.81	0.00	0.00	0.01	0.00	0.00
8	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.96	0.00	0.00	0.00	0.00
9	0.00	0.01	0.00	0.02	0.00	0.00	0.00	0.00	0.95	0.01	0.00	0.01
10	0.00	0.00	0.06	0.00	0.00	0.00	0.01	0.00	0.01	0.90	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.96	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.97

(a) *BTrack*.

True chip \ Predicted chip	1	2	3	4	5	6	7	8	9	10	11	12
1	0.45	0.01	0.20	0.00	0.01	0.00	0.00	0.00	0.00	0.04	0.29	0.00
2	0.01	0.33	0.05	0.00	0.33	0.00	0.00	0.00	0.00	0.28	0.00	0.00
3	0.21	0.06	0.43	0.00	0.07	0.00	0.00	0.00	0.00	0.17	0.05	0.00
4	0.00	0.00	0.00	0.44	0.00	0.20	0.00	0.04	0.30	0.00	0.00	0.02
5	0.01	0.32	0.06	0.00	0.32	0.00	0.00	0.00	0.00	0.29	0.00	0.00
6	0.00	0.00	0.00	0.23	0.00	0.50	0.00	0.18	0.09	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
8	0.00	0.00	0.00	0.04	0.00	0.15	0.00	0.80	0.02	0.00	0.00	0.00
9	0.00	0.00	0.00	0.25	0.00	0.04	0.00	0.02	0.52	0.00	0.00	0.17
10	0.03	0.24	0.14	0.00	0.26	0.00	0.00	0.00	0.00	0.32	0.01	0.00
11	0.35	0.00	0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.59	0.00
12	0.00	0.00	0.00	0.03	0.00	0.00	0.00	0.00	0.31	0.00	0.00	0.65

(b) CFO-based.

Fig. 14: Matching matrix. *BTrack* utilizes TDF to achieve accurate device identification, which is unique and distinguishable among devices even with the same chip model.

still possible that some devices (known as alien devices) are not included in the database before, which may cause false positive alarms. To overcome this, we apply a classification probability threshold. If the maximum probability corresponds to the target devices, we check whether this probability is higher than the threshold. If the probability is higher than the threshold, the sample is identified as the target device. Otherwise, this sample is identified as the "others". In *BTrack*, we select the threshold as 0.8 empirically.

V. Evaluation

In this section, we evaluate the identification accuracy of *BTrack*. Then, we evaluate the temperature stability and robustness of *BTrack*. After that, we show a case study to track moving users. Finally, we evaluate the system overhead.

A. Implementation

We implement *BTrack* with USRP B210. It is connected to a laptop with i7-10750H and GTX1650ti for hardware configuration and data processing. It scans on channel 37 to capture the advertising packet transmitted by BLE devices. In the evaluation, we focus on identifying chips with the same model. We divide each model of chips into three sets: target, others, and alien. The "target" and the "others" sets are used for training and validating our DNN-based classifier. The "alien" set is only used for evaluation. We choose three popular models of BLE chips from three representative manufacturers and the number for each group is shown in Tab. III. The nRF52840 [23] is the class-leading BLE chip produced by Nordic Semiconductor, which has the highest market share among all manufacturers (41% [28]). The DA14580 [29] is a successful commercial chip known for its low power consumption and has been widely used in a large number of wearable devices. The CC2650 [30] is a famous BLE chip provided by well-established manufacturer TI. We apply the experiment at the third floor of a building (Fig. 12), which contains an office, a conference room, a terrace, and a corridor.

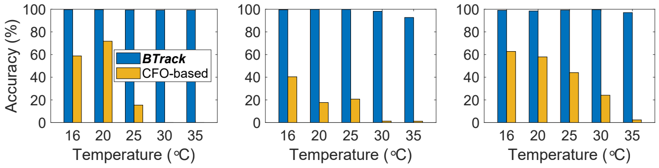
For comparison, we implement the state-of-the-art CFO-based BLE chip identification method proposed in IEEE S&P 2022 [5]. Specifically, we first calculate the CFO profile of each chip in the "target" and the "others" groups. Once a BLE packet is received, we calculate the Mahalanobis distance between its CFO and all the known CFO profiles. If the distance is larger than the threshold, it does not belong to any known device, we label it as "others". Otherwise, the BLE packet is labeled as the type with the minimal distance. The threshold given in [5] is 2, but we set it to 3 since the CFO is sensitive to temperature and we will test its performance in different temperatures.

B. Identification Accuracy

First, we show the identification performance of *BTrack*. We let the DNN-based classifier identify each device and compare it with the CFO-based method. We collect 512 signal traces for each device in the conference room with 16°C temperature and high Signal-to-Noise Ratio (SNR). 80% of the traces for training and 20% for evaluation. Fig. 13 shows the performance of each chip model. *BTrack* achieves 96.95%-99.15% accuracy, 35.38%-57.41% higher than the CFO-based method. The reason is that *BTrack* achieves device identification with TDF, which varies among devices even with the same chip model. But the CFOs of the same model devices are quite similar, making it hard to identify individual devices. To further illustrate that, we show the matching matrix of 12 nRF52840 chips in Fig. 14. Each element of the i-th row and j-th column in the matrix indicates the average matching rate between the i-th chip and the j-th profile in the training dataset. With TDF, there are only two nodes that show a slight similarity (i.e., the 6, 7-th chip). In practice, the adversary needs to ensure that the "others" set chosen from the device database does not contain devices that are similar to those in the "target" set. As for the CFO-based fingerprint, the matching ratio is small and discrete, making the identification accuracy low. There is one chip that with a high matching ratio (i.e., the 7-th chips) since its CFO is far from typical values. Specifically, the revision of the 7-th chip is Rev.1, and the average CFO is 8KHz, while the CFOs of the other Rev.1 chips are between -8KHz and -14KHz.

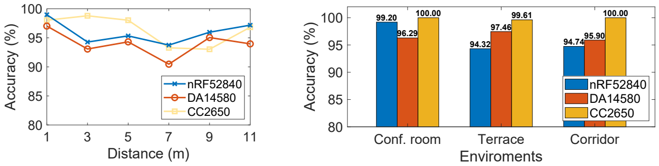
C. Temperature Stability

We train the *BTrack* with the traces collected at 16°C and use an air conditioner to control the room temperature to 20°C, 25°C, 30°C, and 35°C to simulate typical indoor temperatures for each season. At each temperature, we collect 512 signal traces transmitted from the device in the "target" set. Fig. 15 shows the identification accuracy of the "target" devices. The



(a) nRF52840 (b) DA14580 (c) CC2650

Fig. 15: The temperature stability of *BTrack*. The identification accuracy is stable for temperature changes.



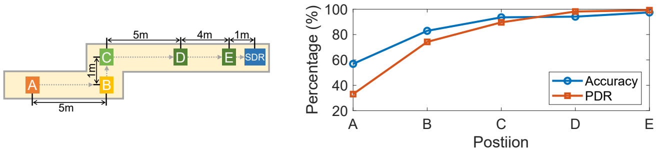
(a) Distance stability (b) Environmental stability

Fig. 16: The distance and environmental stability. The performance is stable with distances and environments.

results show that *BTrack* achieves 92.70%-99.80% accuracy at all temperatures. The CFO-based method totally failed when the temperature was higher than 30°C. The reason is the CFO changes significantly with the temperature, exceeding the distance threshold. For example, the average CFO and standard deviation of one nRF52840 chip at 16°Care 109.96KHz and 1.33, and the average CFO becomes 99.37KHz at 35°C. The difference between the average CFO is 10.59KHz with 7.96 Mahalanobis distance, much larger than the threshold. Similarly, the CFO of DA14580 changes up to 6.52KHz, and 9.05KHz for CC2650. For nRF52840, the accuracy of the CFO-based method increases at 20°C. We find that the CFO has a non-monotonic relation with the temperature. There is a turning point for CFOs of nRF52840 chips at 20°C, i.e., the CFO raised a bit at 20°C and then decreased with temperature increase. At this temperature, the difference between each chip is the largest, leading to higher accuracy. For CC2650, the accuracy to identify the "target" device with CFO (62.75%) is higher than the identification accuracy in Fig. 13. The reason is the CFO of the "target" device is -7.59KHz. Two other devices have very similar CFOs (-7.50 and 7.45KHz), and we remove them from the "others" group of the CFO-based method.

D. System Robustness

We first show the robustness regarding distance. Since the typical communication range of BLE is 10m [31], [32], we place the receiver 1∼11m away from the devices. The data collection is performed in a 26°C office, which contains at least 20 PCs, 20 smartphones, and three servers. We train the classifier with signal traces collected at a 1m distance. Fig. 16 shows the identification accuracy of the target devices with different distances. *BTrack* can achieve 90.47%-98.98% accuracy in all distance. The accuracy first decreases at 1m-7m, then increase a little bit after 9m. The reason is these two locations are near the office door, away from the large number of interference sources in the office. Meanwhile, *BTrack* maintains a very low False Positive Rate (FPR) for all the non-interest devices (<2.77%). For alien devices, the FPR of nRF52840 and CC2650 are lower than 0.62%. The DA14580 at 9m has a bit higher FPR (7.44%) and the FPR of other scenarios is less than 4.08%.



(a) User trace (b) Accuracy

Fig. 17: The accuracy for *BTrack* tracking moving users. *BTrack* works well at LOS and even with slight blockage.

TABLE IV: Runtime of each stage in *BTrack* (μs).

Transient Phase len.	**1. Receiving**	**2. Pre-processing**	**3. Model Filtering**	**4. Identification**	**Total**
20μs	505.99	2282.60	1.95	177.52	2966.22
30μs	524.98	2423.21		218.88	3167.18
40μs	533.67	2612.96		254.93	3401.67

To show the system's robustness in different environments, we collect signal traces of target devices from three other places: a conference room, a terrace, and a corridor. The distance between the receiver and the devices is 1m. We train the DNN-based classifier with the dataset collected from the office. *BTrack* achieves 94.32%-100% accuracy across different environments. Therefore, *BTrack* is stable with different environments.

E. Case Study

We also evaluate the *BTrack* with two mobile users at a corridor with a corner (Fig. 17(a)). Each user carries a nRF52840 and moves from location A to location E. Fig. 17(b) shows the Packet Delivery Ratio (PDR) and detection rate of *BTrack*. Since BLE is a short-range communication protocol and location A is a Non-Line-Of-Sight (NLOS) scenario. The signal transmitted from here is highly distorted and the PDR of BLE beacons is only 33.09%. With those successfully received BLE beacons, *BTrack* can still achieve a 56.90% detection rate. Consider that mobile devices can transmit about 77-872 BLE beacons per minute [5], *BTrack* can still detect the appearance of the target within 0.37-4.14s. Once there is no blockage, there are significant increases of *BTrack* performance, i.e. the detection rate increases 10.65% from location B to location C. The *BTrack* works well (detection rate $\geq$ 93.59%) once there exists Line-Of-Sight (LOS) path to the receiver.

F. System Overhead

We evaluate the runtime of the four stages in Fig. 10 and Tab. IV shows the time for each stage to process one signal trace. The feature length should adapt to the length of the transient delay. The time for packet filtering is fixed since it only takes the values of CFO and transient delay. The preprocessing stage takes the majority of the runtime since it contains lots of time-consuming operations, such as low-pass filtering, convolution for finding preambles, etc. The model filter can effectively block the signals from different model chips. We collect 1,000 BLE beacons from the office and the model filter can filter out almost all of them.

VI. Countermeasures

We propose countermeasures to TDF-based BLE location tracking from hardware and external interference aspects.

Hardware: To completely resolve this issue, hardware modifications are required. For the PLL-based frequency synthesizer, the most cost-efficient way is adding a switch between the VCO and antenna. Therefore, during the transient delay, the switch is opened, so the output frequency of VCO will not leak into the wireless channel. After the frequency synthesizer is stabled at the target frequency, the switch is closed so the chip can normally transmit signals. Another solution is using different types of frequency synthesizers. Direct Digital Synthesizer (DDS) [33], [34] is a viable alternative to the PLL-based synthesizer. It uses a digital circuit to directly generate the target frequency without the negative feedback control process. It is worth mentioning that, since the DDS requires high-speed DACs, the cost and power consumption of the chip will increase. This is the reason why commercial BLE chips still prefer the PLL-based frequency synthesizer.

External interference: We also propose a defense that does not need to change the hardware of those existing BLE chips. To fool the adversary, the user can carry an SDR to transmit interferences. The SDR must continuously generate interferences so its transient phase change will not be detected. It is synchronized with the target devices, when the target is transmitting data, it generates high-frequency noise so the receiver can use a low-pass filter to remove the noise and normally receive the data. The rest of the time, the SDR generates low-frequency noise, so that the adversary can not capture a clean transient phase for device identification.

VII. Discussion

The frequency synthesizer is a necessary part of all wireless chips. For low-power wireless protocols, this fingerprint is commonly present. The LoRa chips from Semtech (SX1276/77/78/79) all use PLL-based frequency synthesizers and the typical transient delay is 60μs [35]. The IEEE 802.15.4 chips also use PLL-based frequency synthesizers, such as the Atmel AT86RF231 [36]. Therefore, the TDF exists in these chips. Fortunately, unlike the BLE, these low-power protocols are typically used in IoT devices, and few users carry these devices around them. Therefore, it is difficult to achieve location tracking attacks with these low-power wireless devices. As for those high-speed wireless protocols (e.g., WiFi), they tend to use the DDS or high-performance PLL-based frequency synthesizer to reduce the transient delay and achieve high throughput. The transient delays of these WiFi chips are too short to be used for device identification. For example, we measure the transient delay of the BCM43455 [37] on Raspberry 4B, the transient delay is 5.86μs, much shorter than low-power wireless protocol chips (e.g., DA14580 has 32.64μs transient delay). Besides, the mobile devices typically work as the WiFi client, which will not continuously broadcast beacons. Therefore, it is difficult for the adversary to achieve location tracking attacks with the WiFi signal.

VIII. Related Work

Software fingerprint: The BLE devices will periodically change their MAC address to achieve anonymity. However, the existing works show that some software identifiers remain consistent. J. Becker et al find the token for generating random MAC addresses is unique and remains static [11]. The Generic Attribute (GATT) profile can also be used to create a fingerprint that can be exploited to circumvent anti-tracking features of the BLE standard [12]. However, these methods require the adversary to continuously collect broadcast packets from the device for a period of time. Besides, these software-level identifiers can be removed by software updates. Therefore, it is difficult to use the software fingerprint to achieve location tracking attacks.

Radio Frequency fingerprint: The Radio Frequency Fingerprint (RFF) originated from hardware imperfections and is introduced during the manufacturing process. These imperfections deviate slightly from their nominal specifications and also vary among different devices. Researchers have extensively studied and proposed many RFFs for different wireless technologies as device identifiers. Existing fingerprints include carrier frequency offset (CFO) [38]–[40], clock skew [41], IQ imperfections [22], [42], etc. The CFO and clock skew originated from the imperfection of the crystal oscillator, which is sensitive to temperature and can not be used to identify the same model devices. The I/Q imperfections only exists in I/Q modulation RF front-ends. Therefore, it can not be used to identify the BLE-specific chips. Some of the works focus on utilizing the transient portion of BLE [43]–[45]. They detect the envelope of the transient portion of the RF signal and extract coarse-grained features (e.g., skewness, deviation of the amplitude envelope, and transient delay.). However, these methods require very high-performance equipment (e.g., spectrum analyzer) since the frequency of the received intermediate frequency signal is about 450MHz [46]. Besides, the coarse-grained features are insufficient to uniquely identify numbers of devices [5]. In this paper, we propose the TDF of the PLL-base frequency synthesizer. This fingerprint originated from the negative feedback control process of the frequency synthesizer, which contains the dynamic feature of the system. The TDF is unique among devices, even for the same model devices.

IX. Conclusion

In this paper, we first propose TDF, which originated with the negative feedback control process of the PLL-based frequency synthesizer and can be affected by hardware imperfections. Our analysis and preliminaries show that TDF is unique and stable under different thermal conditions. With TDF, we propose *BTrack*, a practical BLE location tracking system. Compare with the existing CFO-based method, *BTrack* has 35.38%-57.41% higher identification accuracy and is stable with different temperatures, distances, and environments. This system shows the threat of BLE location tracking attacks is still substantial and we propose possible countermeasures.

References

[1] SIG, "Core specification 5.4," https://www.bluetooth.com/specifications/specs/core-specification/, 2023.

[2] SIG, "Bluetooth 2022 market update," https://www.bluetooth.com/2022-market-update/, 2022.

[3] Apple, "ibeacon," https://developer.apple.com/ibeacon/, 2023.

[4] Google, "Eddystone," https://github.com/google/eddystone, 2023.

[5] H. Givehchian, N. Bhaskar, E. R. Herrera, H. R. L. Soto, C. Dameff, D. Bharadia, and A. Schulman, "Evaluating physical-layer BLE location tracking attacks on mobile devices," in *Proc. of IEEE SP*. IEEE, 2022, pp. 1690–1704.

[6] D. Chen, K. G. Shin, Y. Jiang, and K.-H. Kim, "Locating and tracking BLE beacons with smartphones," in *Proc. of ACM CoNEXT*, 2017, pp. 263–275.

[7] C. Troncoso, M. Payer, J.-P. Hubaux, M. Salathé, J. Larus, E. Bugnion, W. Lueks, T. Stadler, A. Pyrgelis, D. Antonioli *et al.*, "Decentralized privacy-preserving proximity tracing," *arXiv preprint arXiv:2005.12273*, 2020.

[8] S. Akiyama, R. Morimoto, and Y. Taniguchi, "A study on device identification from BLE advertising packets with randomized mac addresses," in *Proc. of IEEE ICCE-Asia*. IEEE, 2021, pp. 1–4.

[9] G. Kalantar, A. Mohammadi, and S. N. Sadrieh, "Analyzing the effect of Bluetooth low energy (BLE) with randomized mac addresses in iot applications," in *Proc. of IEEE iThings and IEEE GreenCom and IEEE CPSCom and IEEE SmartData*. IEEE, 2018, pp. 27–34.

[10] SIG, "Bluetooth technology protecting your privacy," https://www.bluetooth.com/blog/bluetooth-technology-protecting-your-privacy/, 2015.

[11] J. K. Becker, D. Li, and D. Starobinski, "Tracking anonymized bluetooth devices." *Proc. Priv. Enhancing Technol.*, vol. 2019, no. 3, pp. 50–65, 2019.

[12] G. Celosia and M. Cunche, "Fingerprinting bluetooth-low-energy devices based on the generic attribute profile," in *Proc. of ACM IoT S&P*, 2019, pp. 24–31.

[13] É. Helluy-Lafont, A. Boé, G. Grimaud, and M. Hauspie, "Bluetooth devices fingerprinting using low cost SDR," in *Proc. of IEEE FMEC*. IEEE, 2020, pp. 289–294.

[14] J. Huang, W. Albazrqaoe, and G. Xing, "BlueID: A practical system for Bluetooth device identification," in *Porc. of IEEE INFOCOM*. IEEE, 2014, pp. 2849–2857.

[15] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, "Fingerprinting wi-fi devices using software defined radios," in *Proc. of ACM WiSec*, 2016, pp. 3–14.

[16] P. Hao, X. Wang, and A. Behnad, "Relay authentication by exploiting I/Q imbalance in amplify-and-forward system," in *Proc. of IEEE GLOBECOM*. IEEE, 2014, pp. 613–618.

[17] C. Corporation, "Application note crystal basics," https://www.ctscorp.com/wp-content/uploads/Appnote-Crystal-Basics.pdf, 2023.

[18] K. S. Mohamed and K. S. Mohamed, "Hardware realization of GFSK-based Bluetooth modem," *Bluetooth 5.0 Modem Design for IoT Devices*, pp. 45–73, 2022.

[19] B. Huff and D. Draskovic, "A fully-integrated bluetooth synthesizer using digital pre-distortion for pll-based gfsk modulation," in *Proc. of IEEE RFIC Symposium*. IEEE, 2003, pp. 173–176.

[20] W. Yan, T. Voigt, and C. Rohner, "RRF: A robust radiometric fingerprint system that embraces wireless channel diversity," in *Proc. of ACM WiSec*, 2022, pp. 85–97.

[21] J. Hua, H. Sun, Z. Shen, Z. Qian, and S. Zhong, "Accurate and efficient wireless device fingerprinting using channel state information," in *Proc. of IEEE INFOCOM*. IEEE, 2018, pp. 1700–1708.

[22] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. of ACM MobiCom*, 2008, pp. 116–127.

[23] "nRF52840 Objective Product Specification v0.5," https://infocenter.nordicsemi.com/pdf/nRF52840_OPS_v0.5.pdf, 2016.

[24] M. H. Perrott, "Fast and accurate behavioral simulation of fractional-n frequency synthesizers and other pll/dll circuits," in *Proc. of ACM DAC*, 2002, pp. 498–503.

[25] M. H. Perrott, T. L. Tewksbury, and C. G. Sodini, "A 27-mw cmos fractional-n synthesizer using digital compensation for 2.5-mb/s GFSK modulation," *IEEE journal of solid-state circuits*, vol. 32, no. 12, pp. 2048–2060, 1997.

[26] Y. Diao, J. L. Hellerstein, S. Parekh, R. Griffith, G. E. Kaiser, and D. Phung, "A control theory foundation for self-managing computing systems," *IEEE journal on selected areas in communications*, vol. 23, no. 12, pp. 2213–2222, 2005.

[27] F. L. Walls and J. R. Vig, "Fundamental limits on the frequency stabilities of crystal oscillators," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, vol. 42, no. 4, pp. 576–589, 1995.

[28] Nordic Semiconductor, "Q2 & first half report 2023," https://www.nordicsemi.com/-/media/Investor-Relations-and-QA/Quartely-Reports/2023/Q2_Quarterly_Report_2023.pdf, 2023.

[29] Dialog, "DA14580," https://www.renesas.com/us/en/document/dst/da14580-datasheet, 2022.

[30] Texas Instruments, "CC2650 SimpleLink Multistandard Wireless MCU," https://www.ti.com/lit/ds/symlink/cc2650.pdf?ts=1690185806084&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCC2650, 2016.

[31] D. K. Singh and R. Sobti, "Wireless communication technologies for internet of things and precision agriculture: A review," in *Proc. of IEEE ISPCC*. IEEE, 2021, pp. 765–769.

[32] Z. K. Farej and A. W. Talab, "Extended range evaluation of a ble mesh network for control application," in *Proc. of IEEE ICCITM*. IEEE, 2021, pp. 31–35.

[33] M. Zhang, S. Chen, J. Zhao, and W. Gong, "Commodity-level BLE backscatter," in *Proc. of ACM MobiSys*, 2021, pp. 402–414.

[34] A. Bonfanti, F. Amorosa, C. Samori, and A. L. Lacaita, "A DDS-based PLL for 2.4-GHz frequency synthesis," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 50, no. 12, pp. 1007–1010, 2003.

[35] Semtech, "SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver," https://semtech.my.salesforce.com/sfc/p/#E0000000JelG/a/2R0000001Rc1/QnUuV9TviODKUgt_rpBlPz.EZA_PNK7Rpi8HA5..Sbo, 2020.

[36] Microchip, "At86rf231 - complete datasheet,," https://ww1.microchip.com/downloads/en/DeviceDoc/doc8111.pdf, 2023.

[37] Cypress Semiconductor, "Bcm43455 datasheet," https://www.alldatasheet.com/datasheet-pdf/pdf/828943/CYPRESS/BCM43455.html, 2015.

[38] J. Hua, H. Sun, Z. Shen, Z. Qian, and S. Zhong, "Accurate and efficient wireless device fingerprinting using channel state information," in *Proc. of IEEE INFOCOM*, 2018, pp. 1700–1708.

[39] L. Peng, A. Hu, J. Zhang, Y. Jiang, J. Yu, and Y. Yan, "Design of a hybrid rf fingerprint extraction and device classification scheme," *IEEE internet of things journal*, vol. 6, no. 1, pp. 349–360, 2018.

[40] P. Liu, P. Yang, W.-Z. Song, Y. Yan, and X.-Y. Li, "Real-time identification of rogue WiFi connections using environment-independent physical features," in *Proc. of IEEE INFOCOM*. IEEE, 2019, pp. 190–198.

[41] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz, "On the reliability of wireless fingerprinting using clock skews," in *Proc. of ACM WiSec*, 2010, pp. 169–174.

[42] P. Liu, P. Yang, W.-Z. Song, Y. Yan, and X.-Y. Li, "Real-time identification of rogue WiFi connections using environment-independent physical features," in *Proc. of IEEE INFOCOM*, 2019, pp. 190–198.

[43] J. Hall, M. Barbeau, E. Kranakis *et al.*, "Enhancing intrusion detection in wireless networks using radio frequency fingerprinting." *Communications, internet, and information technology*, vol. 1, 2004.

[44] S. U. Rehman, K. Sowerby, and C. Coghill, "RF fingerprint extraction from the energy envelope of an instantaneous transient signal," in *Proc. of IEEE AusCTW*. IEEE, 2012, pp. 90–95.

[45] M. Barbeau, J. Hall, and E. Kranakis, "Detection of rogue devices in bluetooth networks using radio frequency fingerprinting," in *Proc. of Citeseer CCN*. Citeseer, 2006, pp. 4–6.

[46] B. Danev and S. Capkun, "Transient-based identification of wireless sensor nodes," in *Proc. of IEEE IPSN*. IEEE, 2009, pp. 25–36.