A survey of techniques in passive identification of wireless personal devices and the implications on user tracking

Nishant Bhaskar
University of California San Diego

Abstract

In today's world, wireless personal devices such as mobile phones, fitness trackers, medical devices etc. are ubiquitous. These devices provide a convenient user experience by continuously transmitting different wireless messages. Unfortunately this convenience comes at a cost, as a passive eavesdropping adversary can capture these transmissions, and use the information to uniquely identify the devices, and further to track the device owners. This privacy leakage problem is more acute at the link and physical layers, because existing wireless security and privacy mechanisms fail to protect information at these layers.

In this survey, I analyze existing literature to understand how our wireless devices can be identified using techniques at the link and physical layers. For an adversary, the choice of a particular technique is a tradeoff decision. I present a set of heuristics to compare the identification techniques that highlights these tradeoffs. Finally, an adversary that is successful in device identification can further use the captured transmissions to perform user tracking. I present several examples from literature that highlight the extent of this user privacy leakage.

1 Introduction

Wireless personal devices such as mobile phones, fitness trackers, personal medical devices etc. are widely used in today's world. To provide a seamless experience to the end user, these devices continuously send wireless transmissions containing several pieces of information. These transmissions can be either for indicating their presence to other wireless devices, or to send data to specific devices. For instance, continuous WiFi probe requests (upto 2000 probes an hour [21]) ensures your mobile devices are always connected to the highest strength access point, Apple devices continuously transmit Bluetooth Low Energy (BLE) advertisements (200 times a minute [41]) to enable the Continuity protocol etc.

Unfortunately this convenient experience comes at a cost. These wireless packets can be sniffed/eavesdropped by a completely passive observer, and various features can be extracted from them to uniquely identify your wireless device. Moreover, device identity leakage is only the first step, as the adversary can then perform more egregious privacy violation like tracking the device owner, both physically and behaviorally. In essence, our wireless personal devices are homing beacons, that have put a target on our backs for all types of adversarial privacy leakage.

These privacy concerns are hard to resolve at the wireless protocol level, due to the fundamental nature of wireless communication. Encryption standards protect post-authentication data payloads, but the link layer headers still contain unique MAC addresses for identification. Furthermore, at the pre-authentication stage, wireless devices transmit information in device discovery packets in the clear, simply to enable detection by other wireless devices. In recent times, this problem is compounded by the use of these unencrypted discovery packets to transmit data (e.g., Bluetooth Low Energy advertisements in Apple devices). While MAC address randomization can protect this unique identifier, the other pieces of information in these packets have been exploited by several papers to create different types of identifiers. This problem is worse at the physical layer, wherein features identifying the transmitter can be derived, by the mere presence of a transmission.

In today's world, passive eavesdropping based device identification and user tracking is not just a cautionary tale, but a reality. Large scale passive wireless data collection efforts have been undertaken by research groups, which have resulted in huge databases, some of which are available to the public [51, 4, 46]. While these databases are partly anonymized, they still reveal personal user information. For instance, I can run a search to see all households using a Bluetooth CPAP machine, indicating there is a sleep apnea patient in the house. Several industry players have been also found to passively collect wireless traffic secretly with the aim of identifying and tracking users, primarily for targeted

marketing [48, 37, 36, 66] . Understanding in depth how your device can be identified and tracked, is therefore of immediate importance.

In this survey, I present several different techniques in research literature for wireless device identification at the link and physical layer through passive eavesdropping of packets. I only consider papers identifying WiFi and Bluetooth (both classic and Low Energy) transmitters, as these are the most popular wireless protocols for personal devices. These techniques have their own pros and cons, and no one technique can be used in all situations. However, broadly these techniques can apply to almost all existing wireless protocols.

For an adversary, the choice of a device identification technique is a tradeoff decision, based on the pros and cons. Depending on the use case and intended goal, one or several techniques may be used. For instance, analyzing device discovery packet contents can be done using commodity off-the-shelf wireless radios, but the identifier may change with a software update. On the other hand, transmitter imperfections are immune to changes in software, but require the use of special software radios for data collection and analysis. To compare the techniques, I use a set of heuristics that broadly fall in three categories – universality, stability and practicality – following the definitions in [62].

An adversary who has successfully identified your device, is well on their way to achieving their ultimate goal of tracking the device user. In fact, physical and link layer information used for identification is more than sufficient to track and monitor the users – physical location, user behavior or even body movement. I will present several papers, which showcase the extent of this privacy leakage. Majority of these papers rely on unique MAC addresses as a device identifier. While address randomization is available, the device identification techniques I will discuss show that it is not a deterrent to privacy violation.

The rest of the paper is organized as follows: Section 2 provides a formal definition of the passive wireless eavesdropping threat model. In Section 3 I define the various sources of information that are available at the link and physical layer for wireless device identification. I also define the taxonomy for classifying the papers in device identification, as a set of key questions that need to answered to understand the tradeoffs between the techniques. In Section 4 I present the various papers grouped according to techniques in physical and link layer, and present a comparison based on the taxonomy. In Section 5, I look at the consequences of successful device identification, by presenting several papers aiming to performing user tracking in several ways. Section 6 identifies future research in the area from several viewpoints, and we conclude in Section 7.

2 Threat Model

The definition of our threat model follows the privacy metrics defined in [62]. We consider the case of an wireless personal device that is transmitting messages intended for certain receiver(s). The eavesdropper is a *passive local adversary* scanning for *observable data*, in a way that the wireless nodes are unaware of its presence. This means that the eavesdropper will be passively listening to the transmitted packets, but will never inject/spoof packets with the aim of deception nor will it ever send a request prompting the transmitter/receivers to send a response. Additionally, the eavesdropper will not in any way prevent the intended receivers from listening to the packets (e.g., by injecting noise in the communication channel). Also the adversary will make no assumption about the security of the system, i.e., transmitted packets may or may not be fully/partially encrypted.

3 Wireless device identification

An adversary as described in Section 2 can collect the wireless transmissions from these personal devices, and use various techniques to extract features that represent a unique fingerprint/identity for the transmitter. This process is referred to as wireless device identification. On successfully deriving the identity of the wireless personal device, the adversary can track the device owner either physically or behaviorally.

The link layer and physical layer contain several pieces of information that can be used for device identification. These can be unique device identifiers, payload of transmitted packet or even a physical property of the transmitter/-transmission. Next, I present some sources of deriving identifying information that are used in the surveyed literature.

3.1 Identifying information in wireless signals

3.1.1 Link Layer

Device identification information at link layer is primarily due to the differences in how manufacturers implement Wi-Fi and Bluetooth specifications. Packet contents at link layer are transmitted in the clear (despite authentication and encryption at higher layers). The information available in device discovery packets (probe requests, probe beacons, BLE advertisements) and link layer headers in data packets can be utilized for identifying devices. In addition, link layer handles the actual transmission and scheduling of all these packets, and therefore certain timing side channels exist which can be utilized to obtain packet timing specific properties.

Device discovery packets are particularly useful for an adversary to derive identifying information, due to their continuous and periodic availability and variety of information fields. A large body of work in the area of wireless device identification utilizes these packets.

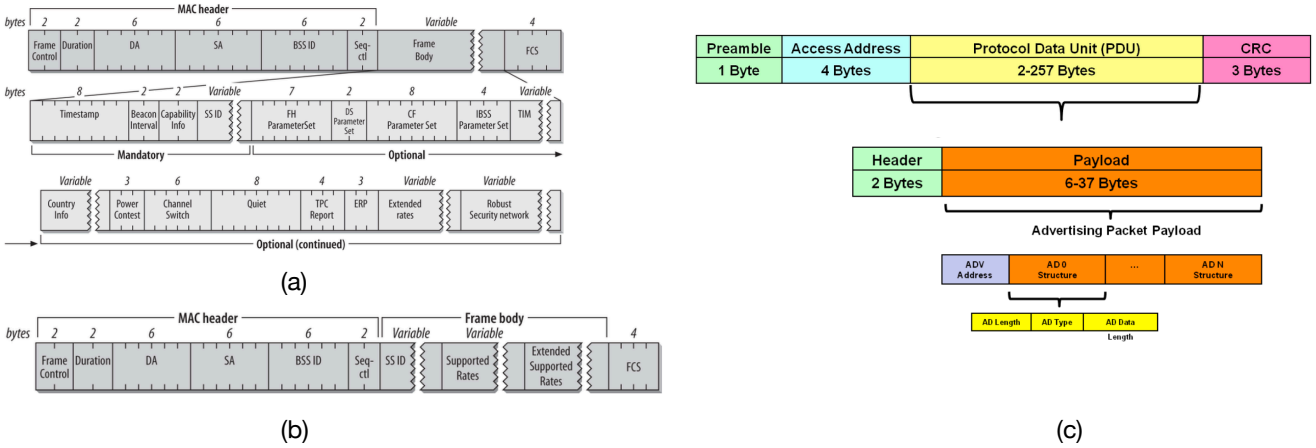

Figure 1: Packet structures of (a) Probe beacon [22] (b) Active probe request [22] and (c) BLE advertisement [45]

Device discovery packets. Prior to authentication and forming a wireless connection, devices need to discover each other. Devices do so by broadcasting link layer device discovery packets, containing information to identify the device (such as MAC address, name) and the features/capabilities it offers.

These device discovery packets are fundamental to WiFi/Bluetooth operation. However, they exist at the pre-authentication stage, and therefore are transmitted in the clear. A passive eavesdropper can obtain direct device identifiers (MAC address, name) or indirect identifiers (features/-capabilities) from these packets. Figure 1 shows the structures of device discovery packets exploited by papers in this survey, and the identifying information they provide. These are as follows:

- **Probe beacon**: Probe beacons are broadcast packets transmitted by the access point(AP). In passive device discovery mode, stations(Wi-Fi clients) sniff for these packets and identify nearby APs and their capabilities. These packets contain informations such as MAC address in the header and several information elements(IE) in the payload. The various IEs can be leveraged to create a distinct identity for an AP. In addition, beaconing interval and timestamp information can be used to profile rate of arrival of beacon packets.

  These IEs include mandatory fields like service set identifier (SSID), beaconing interval, timestamp of transmission of packet, sequence number of frame and capability information such as type of access point infrastructure, security protocols and type of physical layer. Additionally, these packets may contain several optional IEs for other capabilities.

- **Active probe request**: WiFi stations can also perform active device discovery, in which they send probe request to specific SSID to solicit a probe response. Mobile devices in particular use this mode, as it power saving. The directed active probe request contains MAC address of station and SSID of destination AP, along with other mandatory IE fields. Similar to probe beacon, IEs like sequence number and capabilities can be used to develop a device identity. In addition, stations send out bursts of several probe requests, with each request containing a SSID previously connected to. Therefore an eavesdropper can create a list of SSIDs, called Preferred Network List (PNL) for every MAC address, by listening to a probe burst. Furthermore, timing analysis based identity information can be extracted between probe bursts, and between packets in a burst.

- **BLE advertisement**: Advertisements are continuously broadcast by Bluetooth Low Energy slave devices, so that master devices can find them. An advertisement may be directed or undirected, and connectable or non-connectable, resulting in 4 combinations. Different types of advertisements are used in different scenarios. An advertisement contains the advertiser Bluetooth address, alongwith a tag-length-value structure with different data types like Universally Unique Identifier (UUID), and complete Local Name. While device address changes with MAC randomization, the UUID can act as an identifier. Furthermore, advertisement interval can be profiled to uniquely identify the device.

  In recent times, companies like Apple and Microsoft have been using BLE undirected, non-connectable advertisements as a conduit for transferring device event information [1]. These advertisements contain various different data type structures used to represent different types of events. The variety of information and frequency of transmission of these unencrypted data packets, make these packets a serious privacy concern.

3.1.2 Physical layer

The physical layer is responsible for converting the WiFi and Bluetooth packets into the analog signal, and then transmitting the signal over the wireless medium. This exposes several fingerprinting features that inherently describe the behavior of the radio chipset. Importantly, this physical layer information is independent of the type of packet and information contained in it, making it a more lethal weapon for an adversary aiming to identify your device.

Physical layer identifying information may be retrieved by analyzing signal propagation through the wireless medium. This can be done by measuring the received power of the signal which includes effects of attenuation during propagation (received signal strength), or by measuring the effects of signal propagation on the wireless channel itself (channel state information)

Physical layer identification can also be done by analyzing the received signal's non-ideal properties, in either the transient or the steady state part of the signal. These are caused by inherent hardware defects in the transmitter, and are the best identifier for a particular wireless device.

3.2 A taxonomy for comparison

An adversary will choose a device identification technique tailored to their specific monitoring use-case. Therefore, we need a point of comparison for understanding the constraints under which a technique can be used. Accordingly I present a series of fundamental questions to define the heuristics used to compare the literature in the area. The goal is to understand the tradeoffs in using the different techniques. These heuristics fall under the three broad property groups as defined in [62] – *usability*, *stability*, *practicality*

Through these questions, we compare the different real-world constraints under which a technique may or may not be useful. An thing to note is that while accuracy in identifying devices is important, it can vary depending on the implementation specifics (hardware design, software for data capture and analysis, testbed used etc.) of the used technique. Accuracy doesn't offer us a basic understanding into how important a technique is, and therefore we don't use that as a comparison point.

Following are the heuristics used to compare literature in the survey:

- *Works for all device roles? (Role)* : When devices are connected, they assume master/slave roles based on whose clock the two devices are synced. Certain techniques, such as packet timing based, are used and effective only for one type of device role and not universally for all devices.

- *Features stable with changing environment? (Environment)* : Wireless signal features are influenced by changing environmental conditions due to multipath propagation, drifts in temperature, signal absorption etc. Certain techniques, such as signal strength and channel state, depend strongly on the wireless medium and are not stable with changing environment.

- *Features stable with software updates? (Software)* : Software/firmware upgrades in the field are commonly done, and may result in changes in observed protocol features. Certain techniques, predominately link layer methods, depend on specific software implementations, and are not stable to software upgrades.

- *Cheap data collection equipment? (Cost)* : The cost of a passive receiver is important, as any effective data identification system is a network of multiple receivers. Cheap receivers includes most wireless NICs in personal computers, and low cost embedded radios (I consider sub-$100 to be low cost). Certain techniques, such as hardware imperfection based, require the use of costlier sofware defined radios(SDRs) or vector signal analyzers (VSA), and are not practical in terms of cost.

- *Proven to work outside controlled environment? (Outdoor)* : An adversary who needs to track a large population, would require flexible techniques that work well not just in controlled laboratory conditions, but even uncontrolled outdoor environments. Certain techniques, predominately the physical layer methods, have only been shown to work in indoor environments, and are not practical in an outdoor environment.

4 Survey of literature

Table 1 shows a comparison of the device identification techniques according to the taxonomy we defined. We can observe the tradeoff decision in choosing a particular technique.

At a high level, link layer methods offer less stability as they are affected by software upgrades, but are practical in cost and have been proven to reliably work even in uncontrolled environments. Comparitively, physical layer techniques, and in particular hardware imperfections based techniques, are extremely stable and universal such that the derived device identifier is the best representation for the wireless transmitter across all techniques. These techniques are therefore a bigger privacy threat. Unfortunately, capturing and analyzing signal information at physical layer generally involves costly equipment like SDRs. In addition, no literature has demonstrated these techniques working outside lab environments. Consquently, link layer techniques are more practical, and seem to dominate in literature for device and user tracking. In the coming sections, I summarize the literature across the various techniques we have discussed.

		Universality	*Stability*		*Practicality*	
Technique	*Citations*	*Role*	*Environment*	*Software*	*Cost*	*Outdoor*
Link Layer						
Packet Contents	[21, 60, 42, 43, 56, 52, 7, 41]	Yes	Yes	No	Yes	Yes
Packet Timing	[34, 31, 2, 29, 20, 13, 44, 19]	No	Yes	No (Arrival time) Yes (Clock skew)	Yes	Yes
Physical Layer						
Signal Strength	[18, 6, 55, 23, 11]	Yes	No	Yes	Yes	No
Channel State	[54, 65, 32, 38]	Yes	No	Yes	Yes	No
Hardware Imperfections	[25, 24, 58, 9, 61, 39, 28]	Yes	Yes	Yes	No	No

Table 1: Summary of techniques for device identification grouped by networking layer

4.1 Link Layer

4.1.1 Packet contents

We observe that all types of Wi-Fi and Bluetooth devices transmit link layer information continuously. This may be data traffic, or periodic device discovery packets. In fact due to constant availability of device discovery packets, most papers using packet contents technique use these type of packets. By examining the contents of these packets, wireless device identifiers can be derived. While all packets contain a MAC address that uniquely identifies the transmitter, MAC randomization has made it a less potent target. Instead the following papers look at other information fields transmitted in these packets, and use combinations of these fields for deriving unique device identifiers

In terms of taxonomy, packet contents based techniques are universal in all device roles. The derived device identifiers are stable to changes in environment, but can change significantly (or even become non-existent) with software changes. Finally, data collection can be done practically using low-cost off-the-shelf radios, and the several papers have used this technique in a public outdoor location successfully.

WiFi. Freudiger et al. [21] captured probe requests from iOS 8.1.3 and Android 5.0.1 devices and analyzed the MAC addresses. They observed that probe requests from several randomly generated private MAC addresses can be linked together because sequence number increments at a known rate. Further on, they saw these mobile devices reveal their unique actual MAC address in probe requests that are transmitted when the phone screen is active. Therefore on observing over a long period of time, and using sequence number information, an entire set of random MAC addresses can be associated to the actual MAC address. Lastly, they observed that vendor specific information (such as aggregration process used in packet linkage at receiver) is manufacturer dependent, and can also be exploited as an identifier.

Vanhoef et al. [60] analyzed the effectiveness of using WiFi probe IEs as a device identifying feature. By analyzing the Sapienza dataset [4] (dataset of probe requests with actual MAC addresses), they observed most (93.8%) devices don't change the IE fields over time, thereby making it a feasible feature to exploit. However, the level of separation was limited to device models (as IEs from same model are similar). For similar device separation the authors relied upon using sequence numbers and probe arrival times as features.

Further on, they noticed some implementation flaws in WiFi stacks, which can be misused for device tracking. Firstly, for 75% of probe requests, the WPS UUID was derived from the actual MAC address and a fixed salt using SHA256 which meant the actual MAC address can easily be reverse engineered. Secondly, the scrambling mechanism is used to ensure an even spread of 1s and 0s across the OFDM spectrum. This scrambling is done based on a seed value that should be pseudorandom but instead is highly predictable, and can be reversed to be used as a device identifier.

While the techniques introduced in [60] are expected to work despite MAC randomization, they never actually performed analysis on a dataset have randomized MAC addresses in the probe requests. Martin et al. [42] performed a 2 year probe request data collection from multiple phones, and attempted to verify the observations in [60]. They observed that the WPS IE field is not readily available in most devices and therefore UUID derivation is not possible. Instead, they proposed using the IEEE company identifier (the private random MAC addresses are derived from those) to identify the manufacturer, following which sequence numbers can be used to separate similar devices. An important observation was that association/authentication frames increment the same sequence number as probe requests, and also reveal the actual MAC address, making them an important tool in revealing the device identity.

In another paper, Martin et al. [43] derived and analyzed their actual MAC addresses for the devices in the above dataset which had a WPS IE field. They observed that for a given manufacturer, the pattern of assigning MAC addresses is related to the specific model of the wireless device. Therefore, by decomposing actual MAC addresses, a device's manufacturer and specific model can also be figured out. Following this, techniques similar to [42] can be used to separate similar devices.

Bluetooth. Unlike WiFi, Bluetooth classic device discovery packets contain very few information fields for deriving identifiers. In certain specific cases, device identification works well if the goal is to identify a particular class of devices [8].

For classic Bluetooth data packets, Spill et al. [56] solved the master device identification problem, by quite literally extracting packet contents and other Bluetooth properties. By reverse engineering Bluetooth packet contents in real time, they were able to obtain the MAC address, clock bits (and therefore the hopping sequence) and the whitening sequence for the Bluetooth device. Ryan et al. [52] further extended this work to be able to identify and track BLE devices. They also made an interesting observation that BLE devices follow a simple channel hopping mechanism (increment by fixed number) unlike classic Bluetooth, and easily reversible whitening, making their Bluetooth properties easily derivable and identification straightforward.

In recent times the use of BLE advertisements for inter-device message passing has become the norm [64, 1]. All major hardware vendors use a combination of advertisements to provide a seamless experience to the user. But all they end up doing is providing a huge number of packets for the passive eavesdropper to exploit.

Becker et al. [7] observed that major operating systems like iOS, MacOS, Windows 10 and several smartwatch/fitness trackers OSes, continuously send BLE advertisements. While they use periodically changing MAC addresses the payload doesnt change or changes asynchronously to the MAC address. This allowing us to continuously identify and track these devices by observing MAC address and payload identifiers at the same time. In the most egregious case, Windows 10 devices can be tracked indefinitely using this algorithm.

Looking specifically at Apple devices, these continuous BLE advertisements can be attributed to Apple's Continuity Protocol [1]. This protocol enables synchronization between multiple Apple devices using different BLE advertisement messages. In fact, [7] used the Nearby and Handoff messages in particular for the analysis in their paper. Martin et al. [41] performed a detailed analysis of the Continuity Protocol, and found several different features across different packets of the protocol, that can be used for tracking of not only the device, but also reveal user information. For example, device tracking is possible with Handoff messages as they use a sequence number that increments independent of MAC address randomization. Also, Nearby messages never stop transmitting and have a 4-byte data field that remains constant for one or two frames after MAC randomization.

4.1.2 Packet Timing

The link layer is also responsible for deciding the specific time scheduling properties of the various transmissions. For example, device discovery packets are scheduled at certain intervals of time, and the exact time instants are decided by link layer based on channel conditions; Bluetooth data transmit/receive is performed in tightly defined time slots, and is affected by the transmitter clock drift etc.

Packet timing techniques measure these specific timing properties, and use the timing information as features for identifying particular transmitters. In particular the papers I surveyed measure two types of transmitter identifying properties – the drift of the transmitter source clock, time between periodic packet transmissions. Again due to their continuous and periodic nature, device discovery packets feature in most of the literature.

In terms of taxonomy, clock skew measurement has only been shown to work for master devices whereas inter-arrival time has been shown to work primarily for slave devices. Packet timing techniques offer similar environmental stability and practicality as packet contents based techniques,i.e., they are stable to environment changes, and data collection is practically possible outdoors and using low-cost commodity radios. Finally clock skew methods are immune to software upgrades, but inter-arrival time based techniques are not.

Clock Skew. Physical clocks are not ideal and have imperfections. Therefore, the use of a clock source for link layer timing will result in drift from ideal timing values. Clock skew is a measure of that drift, defined as the rate of change of clock offset over time [50].

Kohno et al.[34] were among the earliest to identify the opportunity with using clock skew as a device fingerprint. They observed that network stacks attach TCP timestamps to TCP/ICMP packets at time of sending a packet. Using these timestamps, and measuring the packet receive time they computed the clock skew fingerprint.

Drawing inspiration fromt this work, Jana et al. [31] explored 802.11 network stacks for timing based identification. They observed that AP beacon/probe response packets contain a Time Synchronization Function (TSF) timestamp. They used this timestamp, and measured receive time using the *do_gettimeofday* Linux function, to obtain the clock skew. They estimated the variation in skews of multiple APs in a residential setting.

Using link layer timestamp was advantageous because TCP timestamping [34] requires AP to be associated with some stations. Additionally, APs (whether associated or not) are always sending probe beacon/request responses, and therefore continuous tracking is possible.

However, the skew measurement in [31] is limited by the accuracy of receive time measurement. Arackparambil et al. [2] suggested the use of TSF timestamp (microsecond resolution) on the receive side as well. This provides a 5x lower variance on offset measurements as compared to using the Linux function. They also suggested that line fitting error must be included, to handle fabricated skews (A scenario which Jana et al. had not anticipated).

However, all these methods relied on values reported by the transmitter. Not only are these limited by several transmitter factors (network stack, OS etc.). Bluetooth does not provide such timestamps, so a different approach was needed for skew measurement. Huang et al. [29] observed that Bluetooth defines transmit/receive slot boundaries, and skew will manifest as a drift from these boundaries. They clustered the arrival times of the preambles to generate a fingerprint for a device, and then checked statistical distance of any new cluster to verify if the same.

Huang et al. observed that real Bluetooth radios follow clock skew bounds ($\leq$ 20 ppm), whereas noise is randomly distributed in a short time period. This can be used to filter out noise from legitimate preambles. The linear relation of clock offset over time also meant that they didn't need any knowledge of transmit time, or even time slot boundaries to perform the clustering of preambles.

Inter-packet arrival time. The periodic and continuous nature of device discovery mechanisms exposed another feature – inter-packet arrival time This feature exists because wireless transmitters schedule the probe/advertisement packets at different rates, depending on the wireless stack implementation. This implementation difference can provide fine-grained separation between transmitters.

Franklin et al. [20] fingerprinted Wi-Fi device drivers by binning frequency of probe request arrival times for different (NIC drivers,host OS) combinations. Accuracy of fingerprinting was verified by comparing signatures of 30 minute traces against the database. The intuition was that a particular driver will have defining probe transmit times signature when observed over a long time (in their case 12 hours).

Corbett et al. [13] used frequency domain analysis to differentiate between different NICs, by considering inter-arrival time series data and computing power spectral density They observed that the 50 frequencies with highest power is a defining identifier for classifying NICs. The advantage with frequency domain analysis was that minute timing variations can be captured even with a short trace(e.g.,variation due to rate switching).

However, for similar devices (use the same driver and OS), very high resolution time measurements are required for measuring inter-arrival time differences. Instead, Loh et al. [16] proposed to use the bursty nature of probes, by using inter-probe burst arrival time for identification. By clustering together bursts using a variance threshold, they were able to even able to differentiate similar transmitters with high accuracy. They also observed that inter-burst intervals reduce measurement requirement (resolution of minute-order required), but increase data collection time.

These papers, though either don't explicitly address MAC randomization, or just assume each transmitter has a unique constant MAC address [20]. Matte et al. [44] presented a technique that works with minimal number of packets, and demonstrated proper functioning even with randomization. They combined information from both inter-probe arrival time and inter-burst arrival time to create burst sets grouped using nearest neighbors methods. With this method, they required only 4 groups of bursts per transmitter to achieve high accuracy in device identification. This means that the fingerprint can be derived in the time duration in which a device has a constant MAC address, making this method practical.

In the world of BLE advertisements, Fawaz et al. [19] quantified exact absolute time instants when specific BLE devices would advertise. They used this knowledge to jam advertisements from BLE transmitters, to prevent adversarial tracking. Because devices sense channel and random backoff before choosing to advertise, the likelihood of blocking inocuous advertisements is low. For example, in the common case of advertising time of 1.024 s, less than 30% of inocuous advertisements were blocked, while achieving 100% success in jamming advertisements of upto 10 target devices.

4.2 Physical Layer

4.2.1 Signal strength

Signal propagation through the medium has several effects such as attenuation, scattering, fading etc. Received Signal strength (RSS) is a measure of the received signal power of a wireless transmission. It is a function of the transmitter's distance to receiver as well as channel conditions. A number of papers have attempted to use a series of signal strength measurements indoors, to identify individual transmitters at specific locations.

In terms of taxonomy, signal strength based techniques are universal in all device roles. Being a physical medium based technique, signal strength is stable to changes in software but is heavily influenced by environmental changes. Finally, while data collection can be done using commodity radios, this technique has only been practically proven to work effectively for indoor or enclosed environments.

Faria et al. [18] combined RSS readings for the same transmitter from multiple 802.11 receivers. They used differential values (with respect to the highest RSS for a transmitter) to improve robustness to varying transmission levels. The intuition was that differential RSS reading from closely located transmitters differ by atmost a maximum threshold, whereas different physically separated transmitters differ by atleast a minimum threshold. By varying threshold values and applying different matching rules, they obtained high accuracy in differentiating transmitters separated by 7m, using a network of only 12 receivers.

RSS readings from a stationary transmitter are environment depedent, therefore using absolute values can lead to erroneous results. RSS clustering approaches [11, 6] can be used to solve this problem. Bauer et al.[6] attempted to cluster the signalprint vector for a transmitter use a k-means clus-

tering approach, to combat the noisy environment sources. Requiring just 3 receivers, they were able to obtain upto 77% accuracy in differentiating transmitters separated by 3.5m, even if there were upto 25 transmitters. They observed that even if transmitters were operating at different power levels, the reduction in accuracy was minimal.

Sheng et al. [55] later observed that most 802.11 APs implement antennae diversity. Because of this RSS distributions following a Gaussian Multi-Modal pattern ([18, 11, 6] assumed a simple Gaussian distribution). This GMM nature of RSS can provide more fine-grained features for fingerprinting APs. By using a mixture model to cluster per-frame signalprints, they achieved high detection accuracies using just 7 monitors. Additionally, they observed that the RSS distributions thus modeled are stable over time, despite changing multi-path effects.

Unfortunately, RSS based fingerprinting doesn't work in the presence of mobile transmitters, as signal strength values change drastically. In specific scenarios though, if we had a good estimation of the motion of transmitter/receiver, the device identification can be used to distinguish from transmitters with a different relative motion. Ghose et al.[23] exploited this aspect to design a RSS based authneticater for 802.11 networks. By using a helper device as a wand waved around the device to be authenticated, they observed definite RSS fluctuations Even if the MAC address was spoofed, spoofing the relative motion to the helper is not possible. Additionally, because of close relative proximity to the device, the variation in RSS had a higher roll-off rate, as compared to a snooper that was further away.

4.2.2 Channel State

The major drawback with RSS measurements is variations due to multipath shadowing. These variations are not only over distance but also over time, even over a relatively stable link condition. Comparitively, channel state information (CSI) can separate multipath components, and therefore provide a more fine grained fingerprint based on wireless medium conditions. In the case of WiFi networks, presence of multiple subcarriers results in a large feature set for CSI based device identification. Majority of the work in this area is aimed at Wi-Fi transmitters. These channel state measurements can be performed in the time domain (Channel Impulse Response) or in the frequency domain (Channel Frequency Response).

In terms of our taxonomy, channel state exhibits the same tradeoffs as RSS, i.e., works for all device roles, stable to changes in software but not to environmental changes, can be collected using low-cost radios but impractical to use in an outdoor environment.

[65, 54] used CFR measurements to localize a WiFi transmitter in an indoor setting. Sen et al. [54] used channel frequency response (CFR) measurements from multiple WiFi subcarriers to perform localization. They observed that CFRs vary significatly temporally and with environment changes, but were relatively immune to human movement. Further on CFR reported by different APs for same physical location are diverse and that can be exploited to improve classification. For training, they created a CFR cluster based map of individual 1m x 1m location. For inference, the CFR of packets received from closest AP are checked for similarity distance and then group to a certain CFR cluster (and thereby to a location spot). By receiving beacon packets for 1s at a location, they were able to localize to 1m x 1m spot upto 85%, even if beacons were received from only AP.

[32, 38] utilized CIR measurements to localize Wi-Fi transmitters in an indoor environment instead. Fundamentally CIR is time domain representation of CFR, and provides more spatial information Jin et al. [32] derived CIR by taking inverse fourier transform (IFT) on the receiver's channel estimation (CFR vector), and then reducing number of samples based on system bandwidth required. They utilized non-parametric kernel regression for localization using a logarithmic scale for the approximated CIR vector. The log scale ensures that large delay ACIR elements also contribure fairly to location estimation. They obtained high accuracy in classifying positions even with increased bandwidth. Most importantly, they obtained higher accuracy with just two APs, than a RSS based scheme with 4 APs. Additionally, even with 7 people in the environment, they saw minimal degradation in accuracy performance, which was seen with CFR based studies.

4.2.3 Hardware imperfections

The hardware components of RF signal chain typically have certain manufacturing imperfections, which in turn introduce non-idealities in the transmitted signal. These non-idealities may manifest themselves through transients in the signal, or through an error/offset in the steady state signal itself. Measuring these hardware imperfections can be used to identify the individual transmitters. As these features represent the hardware design of the radio itself, they are the most ideal representation of a transmitter.

In terms of taxonomy, hardware imperfection based techniques can be used universally for any device role. Hardware imperfections are also completely stable in value to both environmental changes and changes to the software. The biggest problem with hardware imperfections as a device identifier is that data collection requires the use of costly SDR or VSA, which makes it less practical to deploy at scale. As perhaps a consequence of this, there exist no work which demonstrates these methods to work outdoors.

Transient signal. When a radio is turned on, there is a short tranient phase before the control loops in the power amplifier and phase locked loop settle. The characteristics of the signal generated at this stage, can identify the hardware com-

ponents of the transmitter uniquely.

Hall et al. [25] used phase characteristics to detect and record transients from Bluetooth radios, unlike previous approaches that used signal amplitude. Phase characteristics are preferrable because they are less susceptible to noise. Also, the slope of phase becomes linear at start of transient, making detection easier. The difference in phase variance for each portion of the unwrapped phase signal was used to create a fingerprint for classification of radios.

Hall et al. [24] further used the same detection mechanism as [25] to retrieve transients for WiFi radios. They measured amplitude, phase and frequency component (using Discrete Wavelet Transform). Using statistical measures on these properties, they obtained a series of 10 properties as a feature vector for fingerprinting using a Bayesian Filter. They achieved 94-100% accuracy in classifying radios, including those from same manufacturer.

Suski II et al. [58] analyzed the effectiveness of amplitude and phase transient detection mechanisms. By computing variance in transiert start estimation error, they realized that amplitude-based methods provide better noise resistance. This observation was in contrast to previous work [25, 24]. To create a classification fingerprint, they used the power spectral desnity sequence. This fingerprint was matched to a cluster using cross-correlation, with a threshold. With these methods they achieved upto 80% accuracy, even with SNR down to 6 dB.

Steady-state signal. Once the control loops in the transmitter hardware have settled, the signal is in steady state, and actual packet reception can begin. A number of papers have attempted to analyze the non-idealities of the received steady state signal.

Some initial papers attempted to do a comprehensive evaluation of various hardware imperfection induced properties [9, 10]. Brik et al. [9] analyzed the properties of frequency error, SYNC correlation, I/Q offset, magnitude and phase error from several 802.11 NICs. Using values averaged over 20 frames, they created a feature vector and used SVM classifier to bin the signals. They achieved phenomenal accuracy of $\geq 99\%$ accuracy in classification, and worst case-similarity at 17%. Additionally, the values were stable to changes in channel conditions and distance from receivers.

Unfortunately, such high accuracy results have not been repeatable since. Vo-Huu et al. [61] hypothesize this was due to a very stable test environment and the use of vector signal analyzer instead of SDR. They attempted to perform classification of modulation features using SDRs. They used a combination of carrier frequency offset, sampling frequency offset, transient and scrambler seed measurements in a short time duration (to ensure MAC address randomization doesn't kick in) and compute similarity distance. While they achieved high accuracy for comparing two different make of radios, classification accuracy was low when testing similar make devices. However the measurements were stable across several days of observation.

Recent work has also attempted in extracting environment independent modulation properties from the channel state information itself [28, 39]. Liu et al. [39] extracted the phase error due to I/Q imbalance from the channel state information. Their filtering was based on the intuition that variance of phase gradients is lower for actual signals even with a varying environment. Modulation properties extracted thus, exhibit similar time and environment invariance.

Work in Bluetooth modulation feature extraction has been limited to detection of presence of wireless transmitters in a noisy environment Sun et al. [57] designed CV-Track to observe variation in CFO values, to detect presence of a BLE signal. The idea was that a BLE packet, even if partially corrected, will result in constant CFO values, if there are overall equal number of 1s and 0s. To distinguish transmissions from multiple beacons, they combined packet CFO values with the inter-arrival time of beacons. The intuition was that frequency mismatch between two transmitters remains constant for a time period longer than a single packet duration.

4.3 Discussion

For an adversary aiming to identify a wireless personal device, there are a number of features to exploit in the wireless transmission, both at the link and physical layer. These features can be used inspite of higher layer security and privacy protocols in place, making these particularly egregious. While many of these papers were aimed at designing intrusion/adversary detection mechanisms, the methods can be easily exploited for adversarial device identification.

The choice of a particular source of information for an adversary boils down to a tradeoff between practicality in the field vs univerality and stability of the feature. Physical layer methods identifying hardware imperfections offer the "golden" device identifier, i.e., the derived identifier offers high universality and stability. That said, they require the use of specialized equipment like signal analyzers or software radios, making them hard to deploy at a large scale (difficult to collect, aggregate and analyze data).

For these very reasons, we see most real world deployments typically use link layer methods to collect and analyze wireless device data. In fact, the multitude of identification information in the link layer packet contents itself is sufficient to identify individual user behaviour in real world scenarios. In the next section, we look at existing literature that showcases the extent of information leakage from the users of these wireless personal devices.

5 Tracking the device owner

In the previous section we saw that passive sniffing of wireless transmissions provides sufficient information for an adversary to identify wireless personal devices uniquely. An

adversary who is successful at device identification, can then proceed to perform more egregious privacy leakage on the device owner. Fundamentally, passive eavesdropping of link layer information (both packet contents and packet timing) can be used in a variety of ways to track the device owner. This tracking can be location or physical tracking, and also behavioral tracking.

In this section, I present several papers that demonstrate as examples to show the various different ways by which user privacy can be compromised by passive sniffing of Wi-Fi and Bluetooth personal devices. In today's world an average person will have multiple such devices on their person, and consequently this information leakage represents an immediate privacy threat that we must address.

5.1 User social relation tracking

With wide deployment of APs, mobile OSs implemented directed active probing of PNL, as a much more power efficient device discovery mechanism. While this provided seamless connectivity to the user, periodic transmission of probe requests also leaked information about the social and demographic background of large crowds.

Cunche et al. [14] were the first to explore this hypothesis. By using similarity metrics from record linkage problems, they were able to observe links between multiple mobile devices (Devices basically operating in a similar location). The important intuition was that while calculating device linkage, higher weight was assigned to SSIDs common to two devices, but not seen frequently across the dataset. For example, a public hotspot is common across the dataset and has a lower weight in linking two devices, as opposed to a home AP.

Barbera et al. [5] took these metrics to a huge scale by large scale temporal passive probe requests collection at events like political rallies, Vatican Pope announcements and public locations like train stations. With a massive dataset of 11 million probes from 160K+ devices, they were essentially able to establish linkages in a bipartite graph that mirrored commercial social networks. From just SSID and OUI information, they revealed several social properties – age groups and political affiliations likely to buy a phone brand, languages people speak and diurnal patterns of students visiting university.

Luzio et al. [17] went one step ahead (and potentially scarier) with the same dataset, and linked each AP to the geographic location based on Wigle data. This revealed a history of locations where a user has been in the past, and were able to make predictions on what cities and locations are they most likely from. They also observed that by eavesdropping probes in election rallies of the major political parties, they were able to closely predict eventual voting statistics as reported in the actual election. This essentially translates to the ability to monitor and predict crowd behaviour providing an avenue for monitoring public at large.

5.2 User location tracking

The continuous probing/advertisement behaviour of the wireless devices allows for position and movement tracking of mobile devices using a network of stationary or mobile passive eavesdoppers (wardriving).

Musa et al. [47] designed a system to collect probe requests by a series of monitors placed along a road. For a moving transmitter, variation in probe intervals, multipath fading, Doppler effect might make it difficult to obtain probe requests. However, by knowing the spatial road networks and traversable road segments, it is possible to combine both detection and non-detection of probes to obtain a very accurate vehicle trajectory. In fact, by just placing monitors 460m apart across a 2.7km road, they were able to estimate trajectory points of moving vehicle to within 67m. Additionally, they also pointed out that more transmissions can be solicited from WiFi devices by using active RTS and null frame injection methods.

Issoufaly et al. [30] proposed a similar technique for tracking BLE device advetisements, by proposing that such tracking can be easily done using smartphones. They proposed building a tracking app that can be deployed in several smartphones, which serve as a BLE Botnet to track personal wireless devices. Korolova et al. [35] lend credence to this idea, by observing that Android and iOS leak consistent device identifiers for an observed BLE device, to all apps on the same phone. Further on, by exploiting inherent backward compatibility mechanisms present in these OSs, an app can run a BLE scan without needing explicit user permission. Interestingly such techniques are similar to wardriving mechanisms that exist in the wild today. In fact, Wigle already has a public WiFi and Bluetooth (BLE and Classic) wardriving database in place.

With the introduction of privacy mechanisms (MAC address randomization) in both WiFi and Bluetooth standards, such tracking mechanisms should have ceased to be effective. Unfortunately, [30, 15, 27] analyzed several innately personal BLE devices (popular fitness trackers) and observed that they still continue to use persistent static addresses for advertisements. Das et al. [15] collected BLE advertising packets at a gym for 8 consecutive days. They observed 95% devices use an $advertisinginterval \leq 8s$, and continuously keep on advertising when not connected to the phone. This results in large packet avaialability to analyze. Identification of fitness trackers was intuitive, as they used tracker model name in the device name. They observed that 89% of trackers didn't change their MAC address (use static addresses). Consequently, they were able to observe 24/99 identified trackers at the gym on several different days, demonstrating ease of long term tracking

5.3 User behavioral tracking

There is a fundamental expectation that data traffic payloads are encrypted, and beyond the master's device identifier (like MAC address), dont reveal any other piece of information. Unfortunately, several different papers have proven this assumption to be simplistic. A number of papers have performed timing analysis on network traffic to reveal user information like browsing patterns (WiFi) and even physical activities (Bluetooth),

WiFi traffic reveals browsing behaviour. The major use case for Wi-Fi networks is to enable Internet access to the variety of mobile and personal devices connected to it. While wireless encryption standards prevent an eavesdropper from directly observing the payloads, a traffic analysis reveals side channel information about the websites a user is browsing to.

Pang et al. [49] looked at analyzing unique network destinations visited by a user, when connected to public hotspots. The idea was similar to PNLs, in that a certain individual is more likely to visit certain websites more frequently, a feature that differs across users. They were able to achieve high accuracy in grouping network destinations from an individual user. By combining specific information in link layer headers, a higher accuracy can be achieved in associating browsing behaviour with individual users. While this works well in the case of a public access point, information beyond link layer is unavailable for private hotspots.

Zhang et al. [67] investigate user online activity leakage by passively sniffing encrypted WiFi data traffic from a private hotspot. Using timing analysis on network traffic, they extracted features such as data rate, inter-frame timing, frame size etc. They used a SVM classifier with these features for several online applications such as browsing, online video streaming, BitTorrent, chatting etc. They were able to achieve high accuracy not only in detecting individual application traffic, but also able to achieve clear separation when multiple applications were running concurrently. The scariest thing was that they achieved over 90% accuracy in classification, passively eavesdropping for just a minute.

Recently, several papers [63, 3] have applied similar traffic analysis technique as [67] to infer specific app usage on smartphones by people. Atkinson et al. [3] used several statistical measures of packet size and inter-packet arrival rate to fingerprint top 34 free apps on Google Play Store. These apps spanned several categories such as news, retail, lifestyle, health, travel, entertainment, social media etc. Using Random Forest based classifier, they achieved a 98% accuracy in identifying specific apps from traffic. Armed with this they attacked 7 specific users and were able to construct user personas, requiring just 5 seconds of individual app data in real time to confirm app usage. They were also able to make simultaneous detections for different users in a noisy enterprise network environment.

Fitness tracker traffic reveals physical activity. With the increasing user base for Bluetooth Low Energy devices, privacy leakage for wireless data traffic has assumed a new form. These BLE radios are used in personal devices like fitness trackers, mice etc, and therefore privacy leakage amounts to leaking information about the physical activities of the human.

Das et al. [15] further observed that performing traffic analysis on fitness tracker data on popular fitness trackers like Fitbit can provide a coarse-grained estimate of user activity. A feature vector for physical activity classification contained payload data rate, number of empty packets and number of start packets. Using this, they achieved 97.6% accuracy in classifying 4 types of user activities - sleeping,sitting,walking,running.

Further on, they observed high correlation for payload data rate and empty packet count with the accelerometer data. By plotting empty packet count vs payload data rate for 5 different users, they observed non-overlapping clusters for each user, indicating that these parameters even reveal the specific style of motion of an individual user.

Even worse, Classen et al. [12] observed recently that all Fitbit models feature a *Live mode* to enable real time user activity display on smartphone. In this mode, once the tracker and smartphone have authenticated, the tracker switches to sending unencrypted data packets to smartphone over the BLE link. This *Live mode* data packet contains information such as distance travelled, elevation, step count and heart rate, all in the clear for an eavesdropper to record.

5.4 Discussion

As we can see in this section, there is a large amount of user identifying information available for an adversary at the physical and link layers. Therefore, an adversary who is succesful at fingerprinting a wireless transmitter can utilize the collected information to identify the device owner.

Interestingly, the papers aimed at user privacy leakage use information exposed through link layer, and not the physical layer. While ease of data collection is a possible reason for this, there is another important factor. The user identifying features exist because of specific implementation and usage of wireless radios, as defined in the firmware/software stack. Therefore, these properties only exist at the link layer and not physical layer.

Among the different type of data/control packets that have been exploited across the papers, probe request (WiFi) and advertisements (Bluetooth LE) are the worst offendors. These packets are transmitted frequently by all personal wireless devices, are unencrypted and contain a number of information fields, that can be misused to conveniently identify both the device and the user.

Fundamentally, it boils down to our design choices. The prevalence of wireless transmissions from personal devices

is to provide convenience to users. For instance, continuous WiFi probe requests ensures your mobile devices is always connected to the highest strength acces point, Apple devices continuously transmit advertisements to enable the Continuity protocol etc. At the same time, the amount of information contained in these packets makes it easy for an adversary to identify the devices and the users. Therefore there is a need to rethink our design choices to prevent this privacy leakage.

Additionally, I believe wireless product designers should not keep repeating the same privacy mistakes, especially with personal devices. For instance, Saponas et al. [53] pointed out that the Nike+iPod sport kit launched in 2006, used persisent device identifiers that can result in user tracking. 12 years on, fitness trackers have now evolved to wrist-worn devices using Bluetooth, and yet expose the same vulnerability [12]. We must understand the risks of identity and user privacy leakage from these wireless personal devices, and make better design choices moving forward.

6 Directions for Future Research

Future research in this area spans multiple different directions.

Practical physical layer device identification. Hardware imperfection based techniques provide an extremely stable and universal device identity. And yet, as we saw in Section 5, none of the device and user tracking methods use this technique. This is primarily due to the cost of the SDRs or signal analyzers needed for acquiring the signal. There have been several efforts in recent times to design and improve efficiency of SDR captures [26, 33] and crowdsourced efforts at spectrum analysis [59]. Future work can look to expand on these efforts to enable practical physical layer identification in outdoor environments, especially if we can enable such data collection and analysis from existing mobile devices [40]. Such research has great value for designing robust intrusion detection systems.

Potential privacy concerns with BLE directed advertisements. BLE has the possibility of performing directed advertising (very similar to WiFi directed probe request), in which advertisements containing specific MAC addresses are transmitted. Fundamentally, the devices that a personal BLE device would connect are more personal than say a WiFi AP, i.e., while PNL based fingerprinting was limited to social linkage assessment in WiFi, in BLE similar techniques may yield a more unique and individualistic fingerprint. Future work needs to analyze how widespread the usage of directed advertisements are, as they can lead to severe privacy violations.

Wireless privacy leakage in medical devices. In this survey we have seen that privacy leakage at the physical and link layer is a concern with wireless personal devices. With the advent of personalized medical equipment such as CPAP machines, insulin pumps etc., passive eavesdropping based information leakage becomes a real and immediate concern. While HIPAA regulations are in place to ensure private patient data confidentiality, data leakage by such wireless devices needs further analysis. Future work needs to analyze the wireless traffic from such devices, to see if confidential patient information is being leaked by such devices.

7 Conclusion

In this survey I presented existing literature exploring techniques that a passive eavesdropper can use to identify wireless personal devices at the link and physical layers. I also presented a body of work which shows that an adversary can then use the same information to track the device owner - both physically and behaviorally. At the basic level, the constant transmission of packets rich in device and user specific information by wireless radios – especially device discovery packets – leads to an always available source of information that can be exploited by a passive eavesdropper for privacy leakage.

We analyzed the tradeoff decision involved in choosing an identification technique. Broadly, a hardware imperfection based physical layer identity is a "golden" representation of the wireless transmitter, but is harder to capture compared to packet contents based link layer technique. Understanding these tradeoff decisions, and the sources of information is important to design better intrusion detection systems. Future research must also look into reducing the availability of information for a passive eavesdropper to reduce such privacy leakage.

References

[1] Apple. Use Continuity to connect your Mac, iPhone, iPad, iPod touch and Apple Watch. https://support.apple.com/en-us/HT204681, Apr. 2019.

[2] C. Arackaparambil, S. Bratus, A. Shubina, and D. Kotz. On the Reliability of Wireless Fingerprinting Using Clock Skews. In *Proceedings of the Third ACM Conference on Wireless Network Security*, WiSec '10, pages 169–174, New York, NY, USA, 2010. ACM.

[3] J. S. Atkinson, J. E. Mitchell, M. Rio, and G. Matich. Your wifi is leaking: What do your mobile apps gossip about you? *Future Generation Computer Systems*, 80:546 – 557, 2018.

[4] M. V. Barbera, A. Epasto, A. Mei, S. Kosta, V. C. Perta, and J. Stefa. CRAWDAD dataset sapienza/probe-requests (v. 2013-09-10). Downloaded from https://crawdad.org/sapienza/probe-requests/20130910, Sept. 2013.

[5] M. V. Barbera, A. Epasto, A. Mei, V. C. Perta, and J. Stefa. Signals from the crowd: Uncovering social relationships through smartphone probes. In *Proceedings of the 2013*

Conference on Internet Measurement Conference, IMC '13, pages 265–276, New York, NY, USA, 2013. ACM.

[6] K. Bauer, D. McCoy, B. Greenstein, D. Grunwald, and D. Sicker. Physical layer attacks on unlinkability in wireless LANs. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 108–127. Springer, 2009.

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

[8] N. Bhaskar, M. Bland, K. Levchenko, and A. Schulman. Please Pay Inside: Evaluating Bluetooth-based Detection of Gas Pump Skimmers. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 373–388, Santa Clara, CA, Aug. 2019. USENIX Association.

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

[10] A. Candore, O. Kocabas, and F. Koushanfar. Robust stable radiometric fingerprinting for wireless devices. In *2009 IEEE International Workshop on Hardware-Oriented Security and Trust*, pages 43–49, July 2009.

[11] Y. Chen, W. Trappe, and R. P. Martin. Detecting and Localizing Wireless Spoofing Attacks. In *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 193–202, June 2007.

[12] J. Classen, D. Wegemer, P. Patras, T. Spink, and M. Hollick. Anatomy of a Vulnerable Fitness Tracking System: Dissecting the Fitbit Cloud, App, and Firmware. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2(1):5:1–5:24, Mar. 2018.

[13] C. L. Corbett, R. A. Beyah, and J. A. Copeland. Passive Classification of Wireless NICs During Active Scanning. *Int. J. Inf. Secur.*, 7(5):335–348, Sept. 2008.

[14] M. Cunche, Mohamed Ali Kaafar, and R. Boreli. I know who you will meet this evening! Linking wireless devices using Wi-Fi probe requests. In *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–9, June 2012.

[15] A. K. Das, P. H. Pathak, C.-N. Chuah, and P. Mohapatra. Uncovering Privacy Leakage in BLE Network Traffic of Wearable Fitness Trackers. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, HotMobile '16, pages 99–104, New York, NY, USA, 2016. ACM.

[16] L. C. C. Desmond, C. C. Yuan, T. C. Pheng, and R. S. Lee. Identifying unique devices through wireless fingerprinting. In *Proceedings of the First ACM Conference on Wireless Network Security*, WiSec '08, pages 46–55, New York, NY, USA, 2008. ACM.

[17] A. Di Luzio, A. Mei, and J. Stefa. Mind your probes: De-anonymization of large crowds through smartphone WiFi probe requests. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[18] D. B. Faria. Detecting identity-based attacks in wireless networks using signalprints. In *in Proceedings of WiSe'06: ACM Workshop on Wireless Security*, pages 43–52, 2006.

[19] K. Fawaz, K.-H. Kim, and K. G. Shin. Protecting privacy of BLE device users. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 1205–1221, Austin, TX, Aug. 2016. USENIX Association.

[20] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Van Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, USENIX-SS'06, Berkeley, CA, USA, 2006. USENIX Association.

[21] 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.

[22] M. Gast. *802.11 wireless networks: the definitive guide*. " O'Reilly Media, Inc.", 2005.

[23] N. Ghose, L. Lazos, and M. Li. SFIRE: Secret-Free-in-band Trust Establishment for COTS Wireless Devices. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 1529–1537, April 2018.

[24] J. Hall. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. In *In Proceedings of the 3rd IASTED International Conference on Communications, Internet and Information Technology (CIIT*, pages 201–206. Kranakis, 2004.

[25] J. Hall, M. Barbeau, and E. Kranakis. Detection of transient in radio frequency fingerprinting using signal phase. pages 13–18, 2003.

[26] M. Hessar, A. Najafi, V. Iyer, and S. Gollakota. TinySDR: Low-Power SDR Platform for Over-the-Air Programmable IoT Testbeds, 2019.

[27] A. Hilts, C. Parsons, and J. Knockel. Every step you fake: A comparitive analysis of fitness tracker privacy and security. Open Effect Report. Available at https://openeffect.ca/reports/Every_Step_You_Fake.pdf, 2016.

[28] J. Hua, H. Sun, Z. Shen, Z. Qian, and S. Zhong. Accurate and efficient wireless device fingerprinting using channel state information. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pages 1700–1708, April 2018.

[29] J. Huang, W. Albazrqaoe, and G. Xing. BlueID: A practical system for Bluetooth device identification. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 2849–2857, April 2014.

[30] T. Issoufaly and P. U. Tournoux. BLEB: Bluetooth Low Energy Botnet for large scale individual tracking. In *2017 1st International Conference on Next Generation Computing Applications (NextComp)*, pages 115–120, July 2017.

[31] S. Jana and S. K. Kasera. On Fast and Accurate Detection of Unauthorized Wireless Access Points Using Clock Skews. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, MobiCom '08, pages 104–115, New York, NY, USA, 2008. ACM.

[32] Y. Jin, W. Soh, and W. Wong. Indoor localization with channel impulse response based fingerprint and nonparametric regression. *IEEE Transactions on Wireless Communications*, 9(3):1120–1127, March 2010.

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

[34] T. Kohno, A. Broido, and K. C. Claffy. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2(2):93–108, April 2005.

[35] A. Korolova and V. Sharma. Cross-App Tracking via Nearby Bluetooth Low Energy devices. In *Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, CODASPY '18, pages 43–52, New York, NY, USA, 2018. ACM.

[36] D. Kravets. An Intentional Mistake:The anatomy of Google's Wi-Fi sniffing debacle. https://www.wired.com/2012/05/google-wifi-fcc-investigation/, May 2012.

[37] M. Kwet. In Stores, Secret Surveillance Tracks your every Move. https://www.nytimes.com/interactive/2019/06/14/opinion/bluetooth-wireless-tracking-privacy.html, June 2019.

[38] F. J. Liu, Xianbin Wang, and H. Tang. Robust physical layer authentication using inherent properties of channel impulse response. In *2011 - MILCOM 2011 Military Communications Conference*, pages 538–542, Nov 2011.

[39] 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.

[40] D. Mantz, J. Classen, M. Schulz, and M. Hollick. InternalBlue - Bluetooth Binary Patching and Experimentation Framework. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '19, pages 79–90, New York, NY, USA, 2019. ACM.

[41] 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(4):34–53, 2019.

[42] 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.

[43] J. Martin, E. Rye, and R. Beverly. Decomposition of MAC address structure for granular device inference. In *Proc. Annual Computer Security Applications Conference (ACSAC)*, 2016.

[44] C. Matte, M. Cunche, F. Rousseau, and M. Vanhoef. Defeating MAC Address Randomization Through Timing Attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, WiSec '16, pages 15–20, New York, NY, USA, 2016. ACM.

[45] Microchip. Bluetooth Low Energy Packet Types. https://microchipdeveloper.com/wireless:ble-link-layer-packet-types, Oct. 2019.

[46] Mimezine. WiGLE. https://wigle.net, 2003.

[47] A. B. M. Musa and J. Eriksson. Tracking Unmodified Smartphones Using Wi-Fi Monitors. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12, pages 281–294, New York, NY, USA, 2012. ACM.

[48] J. O'Malley. TfL is going to track all London Underground users using Wi-Fi. https://www.wired.co.uk/article/london-underground-wifi-tracking, May 2019.

[49] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 User Fingerprinting. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*, MobiCom '07, pages 99–110, New York, NY, USA, 2007. ACM.

[50] V. Paxson. On calibrating measurements of packet transit times. In *Proceedings of the 1998 ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '98/PERFORMANCE '98, pages 11–21, New York, NY, USA, 1998. ACM.

[51] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, J. Zahorjan, and E. Lazowska. CRAWDAD dataset uw/sigcomm2004 (v. 2006-10-17). Downloaded from https://crawdad.org/uw/sigcomm2004/20061017, Oct. 2006.

[52] 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.

[53] T. S. Saponas, J. Lester, C. Hartung, S. Agarwal, T. Kohno, et al. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In *USENIX Security Symposium (USENIX Security 2007)*, 2007.

[54] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. You are facing the mona lisa: Spot localization using phy layer information. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, pages 183–196, New York, NY, USA, 2012. ACM.

[55] Y. Sheng, K. Tan, G. Chen, D. Kotz, and A. Campbell. Detecting 802.11 MAC Layer Spoofing Using Received Signal Strength. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, pages 1768–1776, April 2008.

[56] 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.

[57] 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, pages 1–5, New York, NY, USA, 2017. ACM.

[58] 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.

[59] B. Van den Bergh, D. Giustiniano, H. Cordobés, M. Fuchs, R. Calvo-Palomino, S. Pollin, S. Rajendran, and V. Lenders. Electrosense: Crowdsourcing spectrum monitoring. In *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 1–2, March 2017.

[60] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens. Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, pages 413–424, New York, NY, USA, 2016. ACM.

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

[62] I. Wagner and D. Eckhoff. Technical privacy metrics: A systematic survey. *ACM Comput. Surv.*, 51(3):57:1–57:38, June 2018.

[63] Q. Wang, A. Yahyavi, B. Kemme, and W. He. I know what you did on your smartphone: Inferring app usage over encrypted data traffic. In *2015 IEEE Conference on Communications and Network Security (CNS)*, pages 433–441, Sep. 2015.

[64] K. Wiggers. Why Android Nearby, iBeacons and Eddystone failed to gain traction. https://venturebeat.com/2018/10/27/why-android-nearby-ibeacons-and-eddystone-failed-to-gain-traction/, Oct. 2018.

[65] L. Xiao, L. Greenstein, N. Mandayam, and W. Trappe. Fingerprints in the ether: Using the physical layer for wireless authentication. In *2007 IEEE International Conference on Communications*, pages 4646–4651, June 2007.

[66] D. Yanofsky. Google can still use Bluetooth to track your Android phone when Bluetooth is turned off. https://qz.com/1169760/phone-data/, Jan. 2018.

[67] F. Zhang, W. He, X. Liu, and P. G. Bridges. Inferring Users' Online Activities through Traffic Analysis. In *Proceedings of the Fourth ACM Conference on Wireless Network Security*, WiSec '11, pages 59–70, New York, NY, USA, 2011. ACM.