Title

Evaluating Large-Scale Wireless Scanning for Securing Wireless Access Links in Urban Areas

Author

Bhaskar, Nishant

Publication Date

2023

UNIVERSITY OF CALIFORNIA SAN DIEGO

Evaluating Large-Scale Wireless Scanning for Securing Wireless Access Links in Urban Areas

A dissertation submitted in partial satisfaction of the requirements for the degree Doctor of Philosophy

in

Computer Science (Computer Engineering)

by

Nishant Bhaskar

Committee in charge:

Professor Aaron Schulman, Chair
Professor Patrick Pannuto
Professor Stefan Savage
Professor Geoffrey Voelker
Professor Xinyu Zhang

2023

The Dissertation of Nishant Bhaskar is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

DEDICATION

To my mother, my pillar of strength

EPIGRAPH

Every time we think we have measured
our capacity to meet a challenge,
we look up and we're reminded that that
capacity may well be limitless.

"Josiah Bartlett" (Martin Sheen),
The West Wing

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Across the PhD journey, I went through several trials and tribulations, enjoying the highs of success and enduring the despair that came with failure. While the decision to start the quest was one of my own, it would have been nigh impossible to reach this destination if not for the support of so many of my colleagues, my friends and my family. I will be remiss not to thank everyone who made this journey much simpler and enjoyable.

To begin with, I want to convey my deepest gratitude to my advisor Aaron Schulman. Aaron has been not just a mentor, but one of the big inspirations in my life, both professional and personal. He constantly motivated me to take on challenging, but extremely rewarding research problems, and provided any and all support I needed to succeed in tackling these tricky problems. Along the way, he taught me all facets of the research methodology, from ideation to experiment design, and a particular emphasis on presentation. When required, he wouldn't shy away from even sitting down and teaching me about a topic, classroom style. The learnings I got from him were not limited to research, I also learned a great deal about teaching and mentorship through discussion with him. Most importantly, he always ensured that I receive all the credit and accolades for the successes, while ensuring that I never shoulder the blame solely for the failures. Beyond just the professional relationship, Aaron was always there to support during my trying and challenging times, always encouraging me and believing in me. He truly has shaped not only the wireless security researcher that I have become, but also the person that I am today.

During my PhD journey, I also had the privilege of interacting with and learning about so many facets of academia from several faculty members. I want to thank Dinesh Bharadia for teaching me so much about wireless communication systems. Through numerous collaborations, and via innumerable discussions with him, I learned a lot about building wireless systems and analyzing the wireless physical layer. I want to thank Pat Pannuto for the enthralling discussions, collaborations, and even teaching opportunities on a subject close to my heart — low power embedded systems. A big thank you to Michael Taylor for providing me my first foray into research and introducing me to the research process in my years as a Masters student. I want

to thank Kirill Levchenko for providing me an introduction to security research, and teaching me early on the importance of spending time designing research infrastructure. I want to thank Christian Dameff for teaching me so much about medical cybersecurity and vulnerability research, for providing me with access to resources that were crucial to my research projects, and for the infectious enthusiasm that he would always bring to the room. I want to thank Stefan Savage for providing me the most amazing big picture insights about security research, and particularly for providing me with so much guidance about the legal aspects of cybersecurity research.

I want to thank all the members of my committee Pat Pannuto, Geoffrey Voelker, Stefan Savage, and Xinyu Zhang, for their invaluable advice and comments that helped shape my dissertation.

The PhD also offered me the opportunity to work with and learn from so many amazing student collaborators. I want to thank Shelby Thomas, Pulit Bhatnagar, Maxwell Bland, Hadi Givehchian, Moein Khazraee, Raghav Subbaraman, Samuel Crow, Manideep Dunna, Dhananjay Jagtap, and Alexander Redding for the informative discussions on a myriad of technical topics, and the collaborative work that made much of this PhD dissertation possible.

Through the highs and lows of this grad school experience, the CSE building has been my home, and the CSE community has been my family. I was lucky to get to contribute to many aspects of the department such as Chez Bob, Social Hour, Visit Day that make this department a unique place to be in. I want to thank the faculty, staff and students who have been the social pulse of this department over the years. In particular, I want to thank Geoffrey Voelker, Pat Pannuto, Sorin Lerner, Jennifer Folkestad, Tierra Terell, Jocelyn Bernardo, Matthew Scott, Dhiman Sengupta, Gautam Akiwate, Sunjay Cauligi, John Renner, Joe DeBlasio, Craig Disselkoen, Ariana Mirian, Jennifer Chien.

Finally, this journey would have been impossible without the support of incredible friends that I made along the way. My friends across BSG and The Foundry ensured coming into the office every day, an enjoyable experience. I want to thank Shelby Thomas, Rajdeep Das, Vector

Li, Lixiang Ao, Liz Izhikevich, Evan Johnson, Moein Khazraee, Xiaochu Liu, Luis Vega, Lu Zhang, Anuj Rao, Michael Barrow for making the office a safe space to discuss ideas, learn and grow as both a researcher and a human. The amazing group of friends I made in SysNet over the years, who helped me grow both personally and professionally — Gautam Akiwate, Ariana Mirian, Anil Yelam, Chengcheng Xiang, Tianyin Xu, Stewart Grant, Audrey Randall, Alex Liu, Sunjay Cauligi. Friends from my MS days who kept in touch all these years and always encouraged and celebrated my endeavors — Pulkit Bhatnagar, Abhinav Garg, Abhijit Tripathy, Anuj Rao. My friends from my MS days who stayed back in San Diego, and were my go to for anything and everything — Pulak Sarangi, Anant Dhayal, Suganth Krishna, Mainak Biswas, Thyagarajan Venkatanarayan. My Amdavadi group of friends who kept me connected to my Gujarati roots — Aditi Mavalankar, Vraj Shah and Aashka Vora. A friend group that really developed in the last couple of years of my PhD, but were instrumental in really getting me over the line, and have since became an indispensable part of my journey — Sampada Kallol, Janet Johnson, Dhiman Sengupta, Gokce Sarar, Shravan Narayan. And finally, my friends from undergrad and my past work life who even till this day are some of my biggest cheerleaders, people I can always rely upon for advice whenever I need it the most — Gaurav Karwa, Jagdish Ghuge, Shashank Maheshwari, Manali Shinde.

Last but by no means the least, I want to thank my family, and in particular my mother Nutan Kumari and my father Bhanu Pratap who supported and encouraged me at every step of this long process. I owe a lot to my mother, who was my inspiration when I started the PhD program, and has been a pillar of strength for me across the highs and lows of this journey.

Chapter 3, in part, is a reprint of the material as it appears in *Usenix Security Symposium 2019*. Nishant Bhaskar, Maxwell Bland, Kirill Levchenko, and Aaron Schulman. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in part, is a reprint of the material as it appears in *IEEE Symposium on Security and Privacy 2022*. Hadi Givehchian, Nishant Bhaskar, Eliana Rodriguez Herrera, Hector Rodrigo Lopez Soto, Christian Dameff, Dinesh Bharadia, Aaron Schulman. This work was supported in part by a gift from Amateur Radio Digital Communications. The dissertation author was the co-primary investigator and author of this paper.

Chapter 5, in part, is currently being prepared for submission for publication of material. Nishant Bhaskar, Raghav Subbaraman, Sam Crow, Moein Khazraee, Dinesh Bharadia, Aaron Schulman. This work was supported in part by National Science Foundation grants NSF-2239163 (CAREER), and NSF-2213689. The dissertation author was the primary investigator and author of this material.

VITA

2012	Bachelor of Engineering in Electronics and Instrumentation, Birla Institute of Technology and Science Pilani
2012–2015	Embedded Application Engineer, Texas Instruments Inc.
2015–2017	Teaching Assistant, Computer Science and Engineering University of California San Diego
2017–2023	Research Assistant, University of California San Diego
2019	Master of Science in Computer Science (Computer Engineering), University of California San Diego
2023	Doctor of Philosophy in Computer Science (Computer Engineering), University of California San Diego

PUBLICATIONS

Givehchian, H., Bhaskar, N., Redding, A., Zhao, H., Schulman, A., Bharadia, D., 2024. *Practical Obfuscation of BLE Physical-Layer Fingerprints on Mobile Devices*. In 2024 IEEE Symposium on Security and Privacy (IEEE S&P 2024)

Nikoofard, A., Givehchian, H., Bhaskar, N., Schulman, A., Bharadia, D. and Mercier, P.P., 2022. *Protecting Bluetooth User Privacy through Obfuscation of Carrier Frequency Offset*. In IEEE Transactions on Circuits and Systems II: Express Briefs.

Subbaraman, R., Bhaskar, N., Crow, S., Khazraee, M., Schulman, A. and Bharadia, D., 2022, June. *Observing wideband RF Spectrum with low-cost, resource limited SDRs*. In Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services (MobiSys 2022).

Givehchian, H., Bhaskar, N., Herrera, E.R., Soto, H.R.L., Dameff, C., Bharadia, D. and Schulman, A., 2022, May. *Evaluating Physical-layer BLE Location Tracking Attacks on Mobile Devices*. In 2022 IEEE Symposium on Security and Privacy (IEEE S&P 2022).

Jagtap, D., Bhaskar, N. and Pannuto, P., 2021, June. *Century-scale Smart Infrastructure*. In Proceedings of the Workshop on Hot Topics in Operating Systems (HotOS 2021).

Bhaskar, N., Bland, M., Levchenko, K. and Schulman, A., 2019, August. *Please Pay Inside: Evaluating Bluetooth-based Detection of Gas Pump Skimmers*. In USENIX Security Symposium (USENIX Security 2019).

ABSTRACT OF THE DISSERTATION

Evaluating Large-Scale Wireless Scanning for Securing
Wireless Access Links in Urban Areas

by

Nishant Bhaskar

Doctor of Philosophy in Computer Science (Computer Engineering)

University of California San Diego, 2023

Professor Aaron Schulman, Chair

Modern society relies upon the safe and secure operation of wireless communication links in several computing systems, from personal devices to public infrastructure. These wireless access links utilize Bluetooth and WiFi radios, and enable users to remotely access and monitor computing systems, conveniently and at a safe distance. For instance, equipment on the grid can be remotely accessed, and COVID exposure information can be obtained at a safe distance using wireless links.

Unfortunately, these wireless access links have also made our computing systems less secure — attackers can gain unauthorized access, or remotely track these systems through these

legitimate wireless links. Furthermore, attackers even implant their own illicit wireless links to gain access to personal equipment and critical infrastructure alike, e.g., payment card skimmers at gas stations.

In order to secure these wireless access links, we need to understand if attackers are gaining unauthorized access by hiding illicit links, and if attackers are performing targeted attacks on popular wireless access links. Wireless scanning-based auditing can be a potential solution to develop insights about the above security and privacy problems. However, there are several challenges to utilizing wireless scan information for this auditing, that bring to question the feasibility of wireless scanning as a security approach. In particular, wireless scans provide limited information and the wireless access links are extremely diverse, making targeted auditing of particular wireless link a needle in a haystack problem. Furthermore, these links are spread across large metropolitan areas needing us to do wardriving wireless scanning, but the existing scanning tools are extremely slow to discover all devices, making it tough to reliably scan for all wireless access link real-world locations.

In this dissertation, I perform several large scale field measurement studies of real-world wireless access links, by performing wireless scanning based auditing across entire metropolitan areas. I study actual security and privacy scenarios to demonstrate the feasibility of wireless scanning based targeted auditing as a tool to defend against attacks on wireless access links. I also analyze the practical challenges and limitations of performing such targeted auditing, from the perspectives of attackers and defenders. In summary, I defend the following thesis statement: *To defend wireless access links spread across urban areas, it is feasible to: 1) use link layer scan information to identify illicit wireless links, 2) use physical layer information in wireless signals to attack a target wireless device, and 3) scan reliably for all wireless access links when wardriving using low-cost commodity hardware*

Chapter 1

Introduction

Modern society relies upon the safe and secure operation of wireless communication links that are part of many computing systems. From personal devices like smartphones, to public infrastructure like grid equipment and streetlights, wireless access links (e.g. Bluetooth and WiFi) are used in a myriad of applications across metropolitan areas.

The wireless access links allow users within wireless range, a connection (e.g. classic Bluetooth connections) or a connectionless (e.g. Bluetooth LE beacons) data exchange mechanism to the computing systems. They are not connected onto any centralized network (e.g. the Internet), and are similar to wireless ad-hoc links. To enable this ad-hoc link formation, these links are typically scannable and connectable by the user device. Amongst the wireless technologies, Bluetooth and WiFi are particularly popular for wireless access links, because of their widespread availability on user devices (e.g. smartphones). Today these links are not only integrated onto newer computing systems, but even being retrofitted onto legacy computing systems (such as grid infrastructure) These are so ubiquitous that its common to see hundreds of such links at any public location in urban areas.

These wireless access links enable remote access and passive monitoring of the computing systems, conveniently and at a safe distance. For example, a maintenance worker no longer needs to climb up the pole near a dangerous high voltage line to physically access a circuit breaker; they can remotely connect to its Bluetooth interface and run diagnostics at a safe distance [70]. During the COVID-19 pandemic, information about exposure to COVID was conveniently communicated through wireless links; smartphones used continuously transmitting BLE beacons to alert users of potential exposure when they were at an unsafe distance from an infected individual.

While integrating wireless access links has made it convenient to access computing systems safely, it has also made them less secure and private. Attackers can now misuse these wireless access links to gain unauthorized access, or even remotely track our computing devices; and they can do these attacks covertly from a distance, without risk of being caught. For example, researchers have demonstrated it is possible to remotely access and control urban infrastructure such as traffic lights at intersections [50], and circuit breakers on power lines [41]. They have also demonstrated covert tracking of individuals by simply listening to wireless signal transmissions from their personal mobile devices [37]. Furthermore, attackers are exploiting the ubiquity of wireless devices, by implanting their own illicit wireless access links to gain illegitimate access to critical infrastructure and even unsuspecting people. For instance, criminals have been installing Bluetooth radios in payment terminals at gas stations and ATMs to commit millions of dollars in fraud [108]. Also, stalkers have been covertly placing AirTags onto unsuspecting victims, to follow and track them [16]. Therefore, learning how to protect these urban scale wireless access links has become critical to our public health and safety, economy and even national security.

Securing these wireless links requires us to have a comprehensive understanding of the potential attack surface, across entire urban areas, and then build proactive defenses. More specifically, we need to understand if attackers are gaining unauthorized access to computing systems by hiding illicit wireless links, and if attackers are performing targeted attacks on popular wireless access links. To do so, we need to perform wireless scanning-based targeted auditing of all such wireless links across entire metropolitan areas.

Scanning based auditing is a standard mechanism for securing interconnected computing devices on an organizational network (or even on the Internet). Network monitors run scans on a network to enumerate information about every connected computing device. This information helps us identify vulnerabilities that can be targeted by attackers in legitimate network connected devices, as well as any illegitimate devices installed by attackers on the network.

Despite its promise, its unclear whether scanning based auditing mechanisms is an effective security approach for this non-interconnected "network" of wireless access links in urban areas. There are multiple challenges that bring to question the feasibility of wireless scans in understanding the threats of illicit wireless links or targeted attacks on legitimate wireless links.

1.1 Challenges

The wireless access links spread across urban areas are extremely diverse. They are used in a wide variety of computing devices across urban areas, from personal user smart devices to public grid infrastructure. Also, there are a diverse set of manufacturers of the boards and modules used in these wireless links, resulting in a diverse set of links at any public location. Not only this, but even modules from same manufacturer can have diversity in the wireless chipsets used, and even physical hardware architecture variations within the same module. Consequently, it is likely to see several computing devices using the same type of wireless access link in public place, or the same type of computing device using different types of wireless links across an urban area.

However, unlike conventional network scanning, wireless scanning captures minimal information at the link and physical layers. At the link layer, wireless scans reveal basic pieces of information like MAC address, friendly names, device types and geospatial locations. At the physical layer, a few basic device identifiers such as Carrier Frequency Offset (CFO), I/Q imbalance/offset may also be obtained. In addition, the availability of information is unreliable — scans may not always obtain all information such as names, and properties such as MAC address, CFO, I/Q may change over time and with varying environmental conditions.

Consequently, the limited information in wireless scans coupled with the diversity of wireless access links has made targeted auditing using wireless scans of any particular wireless link a needle in a haystack problem. For example, at a public location its unclear if we can uniquely differentiate a Bluetooth module in an illicit gas pump skimmer, from the same module used in a streetlight and a speed sign. In other words, **wireless access links are hidden in the noise of several other wireless devices in public places**.

In addition, these wireless access links are utilized in applications that are spread across metropolitan areas. Performing a comprehensive targeted audit of wireless links will require us to perform data collection across entire cities and countries. This can be achieved through wardriving-based wireless scanning field data collection campaigns

Unfortunately, existing wireless scanning tools are extremely slow to reliably discover all wireless devices while wardriving. Wireless scan protocols are fundamentally designed for wireless devices that send requests/receive responses and beacons across multiple scan channels sequentially to enumerate information about surrounding devices. This sequential nature of wireless scanning makes it extremely inefficient for wardriving applications — for example, classic Bluetooth scans need 10 seconds to reliably discover all devices in range, and yet a vehicle driving at 50 mph will be in a typical range of 100 m for only about 4 seconds. As a consequence, **using current wireless scanning tools for targeted auditing may miss wireless access links in the collected scan information**.

1.2 Thesis

The above challenges bring to question whether wireless scanning is a feasible approach for defending this “network" of non-Internet connected wireless access links spread across urban areas. We don’t know if wireless scanning provides reliable information to identify illicit links in computing systems, or understand if attackers can perform targeted tracking of common wireless access links.

In this dissertation, I perform several large scale field measurement studies of real-world wireless access links, by performing wireless scanning based auditing across entire metropolitan areas. I study actual security and privacy scenarios to demonstrate the feasibility of wireless scanning based targeted auditing as a tool to defend against attacks on wireless access links. I also analyze the practical challenges and limitations of performing such targeted auditing, from the perspectives of attackers and defenders.

In summary, I defend the following thesis statement:

To defend wireless access links spread across urban areas, it is feasible to: 1) use link layer scan information to identify illicit wireless links, 2) use physical layer information in wireless signals to attack a target wireless device, and 3) scan reliably for all wireless access links when wardriving using low-cost commodity hardware

Wireless access links found in urban areas use various types of communication protocols — WiFi, Bluetooth, Zigbee and others For this dissertation, I focus my field studies and tool design for Bluetooth-based links — both Bluetooth Low Energy and Bluetooth Classic. Indeed, Bluetooth is the most ubiquitous wireless ad-hoc technology in use today, across personal devices, and public facing infrastructure. In fact, its ubiquity and ease-of-use is what even prompted criminals to use it in gas pump skimmers. Despite the focus on Bluetooth scanning, the lessons from my dissertation are universal across all types of wireless protocols, as scan information and scan methods are very similar across protocols. Therefore, the described body of knowledge applies widely to the entire "network" of non-interconnected wireless access links found in urban areas.

1.3 Contributions and Organization

The remainder of this dissertation is organized as follows.

In Chapter 2, I present an overview of prior literature in the domain of wireless access link auditing. I survey papers in the area aimed at wireless device identification using information from wireless scans, at both the link-layer and physical-layer.

In Chapter 3, I describe a field measurement study to understand the effectiveness of using link-layer information from smartphone Bluetooth scans for the detection of Bluetooth-based credit card skimmers at gas stations. We designed a custom Bluetooth scanning app called Bluetana, and provided that to field investigators, who collected Bluetooth scan data across 1185 gas stations in multiple states in the US. I analyze this scan data to understand how the scan properties of detected skimming devices look like. I also analyze whether these skimming devices are actually "hidden in the noise" of the several legitimate Bluetooth devices seen at a gas station. Finally, I highlight several operational lessons we learned during the course of the study, about the benefits and challenges of using wireless scanning for detecting illicit devices.

In Chapter 4, I present another field measurement study to understand whether it's feasible for an attacker to use physical-layer information in wireless signals to perform targeted tracking of a wireless personal device. I capture BLE beacon signals captured in real-world locations like food courts, library and public facility. We do a detailed discussion of the real-world challenges an attacker faces in using these signals to perform physical-layer tracking. I then analyze whether the hardware properties of the BLE transmitter derived from these signals can be used to track a particular BLE access link. Finally, I analyze if a particular wireless link has unique imperfections or is "hidden in the noise" of the several BLE-enabled wireless links seen in public locations.

In Chapter 5, I describe the design of a wireless scanning tool aimed at ensuring we can reliably scan for all wireless links in range, even when performing large-scale driving experiments in which devices are in range for only few seconds I present the design of a new parallel Bluetooth scanning protocol, that reduces the time taken to ensure we can reliably scan for all Bluetooth devices down to a few seconds. I then present hardware challenges with implementing such a scanning protocol on low-cost SDR hardware, such as PAPR distortions and limited bandwidth. Finally, I present solutions to these hardware challenges based on the details of the classic Bluetooth scanning protocol.

Chapter 3, in part, is a reprint of the material as it appears in *Usenix Security Symposium*

2019. Nishant Bhaskar, Maxwell Bland, Kirill Levchenko, and Aaron Schulman. The dissertation author was the primary investigator and author of this paper.

Chapter 4, in part, is a reprint of the material as it appears in *IEEE Symposium on Security and Privacy 2022*. Hadi Givehchian, Nishant Bhaskar, Eliana Rodriguez Herrera, Hector Rodrigo Lopez Soto, Christian Dameff, Dinesh Bharadia, Aaron Schulman. The dissertation author was the co-primary investigator and author of this paper.

Chapter 5, in part, is currently being prepared for submission for publication of material. Nishant Bhaskar, Raghav Subbaraman, Sam Crow, Moein Khazraee, Dinesh Bharadia, Aaron Schulman. The dissertation author was the primary investigator and author of this material.

Chapter 2

Related Work

In this chapter I present prior literature in the domain of wireless access link identification. I explore various papers aimed at targeted device identification or auditing using link layer and physical layer information from wireless scans. While the upcoming chapters tend to focus on Bluetooth-based wireless access link, in this chapter I look at prior work in device identification for Bluetooth as well as WiFi links.

2.1 Wireless access link identification

The link layer and physical layer contain several pieces of information that can be used for wireless access link 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.

2.1.1 Identifying information at the Link Layer

Identifying information at the link layer is primarily due to the differences in how manufacturers implement WiFi and Bluetooth specifications. Packet contents at link layer are transmitted in the clear (despite authentication and encryption at higher layers). The information available from wireless scans (obtained through device discovery packets) 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.

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. Following are the device discovery packets utilized in WiFi and Bluetooth protocols, and the information they provide for targeted auditing:

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.

Classic Bluetooth FHS responses. Classic Bluetooth scanners send inquiry requests to which Bluetooth devices respond with Frequency Hopping Synchronization (FHS) packets. In particular, these packets provide the MAC address and Class-of-Device of the responding device. In order to obtain the device Bluetooth name, the scanner must send a second packet (Remote Name Request) to which the Bluetooth access link respond with its friendly name.

Bluetooth LE 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, along with 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 [9]. 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.

2.1.2 Identifying information at the 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

Table 2.1. Summary of targeted device identification techniques in prior literature

Technique	*Citations*
Link Layer	
Packet Contents	[49, 109, 76, 77, 101, 91, 22, 75]
Packet Timing	[64, 58, 13, 56, 48, 33, 78, 47]
Physical Layer	
Signal Strength	[46, 21, 99, 51, 32]
Channel State	[97, 113, 60, 72]
Hardware Imperfections	[53, 52, 103, 28, 111, 73, 55]

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.

2.2 Survey of literature

Table 2.1 shows a summary of the device identification techniques we discuss in this chapter, grouped according to the identifying information utilized. Following is an overview of the literature across the various techniques.

2.2.1 Link Layer

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.

Wi-Fi. Freudiger et al. [49] 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 aggregation process used in packet linkage at receiver) is manufacturer dependent, and can also be exploited as an identifier.

Vanhoef et al. [109] analyzed the effectiveness of using WiFi probe IEs as a device identifying feature. By analyzing the Sapienza dataset [19] (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 [109] 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. [76] performed a 2 year probe request data collection from multiple phones, and attempted to verify the observations in [109]. 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. [77] 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 [76] 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 [24].

For classic Bluetooth data packets, Spill et al. [101] 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. [91] 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 [112, 9]. 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. [22] 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 doesn't 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 [9]. This protocol enables synchronization between multiple Apple devices using different BLE advertisement messages. In fact, [22] used the Nearby and Handoff messages in particular for the analysis in their paper. Martin et al. [75] performed a detailed analysis of the Continuity Protocol, and found several 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.

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 [85].

Kohno et al. [64] 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 from this work, Jana et al. [58] 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 [64] 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 [58] is limited by the accuracy of receive time measurement. Arackparambil et al. [13] 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. [56] 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. [48] 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. [33] 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. [42] 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 [48]. Matte et al. [78] 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. [47] 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.

2.2.2 Physical Layer

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. [46] 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 [32, 21] can be used to solve this problem. Bauer et al.[21] attempted to cluster the signalprint vector for a transmitter use a k-means clustering 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. [99] later observed that most 802.11 APs implement antennae diversity. Because of this RSS distributions following a Gaussian Multi-Modal pattern ([46, 32, 21] 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.[51] 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.

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.

[113, 97] used CFR measurements to localize a WiFi transmitter in an indoor setting. Sen et al. [97] 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.

[60, 72] 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. [60] 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.

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 components of the transmitter uniquely.

Hall et al. [53] 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. [52] further used the same detection mechanism as [53] 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. [103] 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 [53, 52]. 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 [28, 31]. Brik et al. [28] 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. [111] 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 [55, 73]. Liu et al. [73] 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. [102] 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.

Chapter 3

Link-Layer Wireless Scan Information for Identifying Illicit Wireless Links

The ubiquity of wireless access links has made it easier for attackers to attack public infrastructure. Today, criminals are implanting illicit wireless links to gain covert unauthorized access to gas pumps [14, 15]. These illicit wireless links are "hidden in the noise" of tens of others such links at public locations. In this chapter, I describe the results of a 19-month metropolitan scale field measurement study to understand the feasibility of using link-layer information from smartphone Bluetooth scans to identify these illicit links. In particular, this field study was aimed at defending against a type of illicit link — Bluetooth-based payment card skimmers.

Payment card skimming attacks at gas pumps have reached alarming levels. In 2018, law enforcement officials recovered 972 skimmers from gas pumps in Florida [15] and 148 skimmers from Arizona [14] alone. Based on industry estimates, a single skimmer can capture 30–100 credit cards per day [5] and each card, based on estimates from law enforcement officials, nets the criminal an estimated $500 [6], resulting in a daily loss of $15,000–50,000 per day of operation for each skimmer.[1] Less is known about how long a skimmer remains in operation, but allowing for even one day of operation per skimmer, 2018 losses exceed $16 million across these two states.

[1] In Section 3.1.2, we compare these quoted estimates to other sources, and find them to be in agreement.

Gas pumps are an ideal skimming target. Gas pumps have relatively weak security: their payment circuitry can be accessed with universal keys or crowbars, and reading payment data is as easy as tapping into a ribbon cable (Section 3.1.1). Gas pump skimmers can be hidden inside of a gas pump enclosure, making them difficult to detect. As a result, inspectors have resorted to manually opening the pumps to inspect their wiring for skimmers. Gas pump skimming has become so pervasive that the Arizona Department of Agriculture, Weights and Measures Division (AZWMSD) now checks for skimmers while doing routine inspections.[2] From 2016 to 2018, the AZWMSD looked for skimmers in 7,325 gas station inspections. Inspectors found skimmers in only 1.5% of these inspections.

Unfortunately, Law Enforcement (LE) rarely catch criminals while they are collecting payment data from gas pump skimmers. The reason is, many gas pump skimmers are equipped with Bluetooth connectivity [65, 66, 67, 68]. This allows criminals to remain in their car while wirelessly retrieving card payment data. While Bluetooth is a vital tool for criminals to exfiltrate data from gas pumps, it also could be an opportunity to make it easier to detect skimmers.

In this measurement study, we evaluate the effectiveness of using Bluetooth scans from a smartphone to detect these payment card skimmers. Indeed, if a skimmer can be detected with a smartphone, then authorities can discover and remove skimmers passively and quickly while they visit a gas station for other reasons. We built a smartphone application to perform this study, called Bluetana. Bluetana collects all Bluetooth scan data that is available via the Android Bluetooth APIs. We equipped 44 volunteers in six U.S. states with smartphones running Bluetana. Our volunteers have collected wireless scans at 1,185 gas stations, where they observed a total of 2,214 Bluetooth devices. In these scans, Bluetana detected a total of 64 skimmers installed at gas stations in Arizona, California, Nevada, and Maryland, and it was the sole source of information that led law enforcement to find 33 skimmers.

Our study is the first comprehensive look at how skimmers can appear in Bluetooth scans. Namely, we observe that it is feasible to differentiate skimmers from other common Bluetooth

[2] For example, the “Vapor Recovery Inspection Pre-Test Checklist” has a checkbox for “Checked for Skimmers”.

devices that appear in Bluetooth scans at gas stations (e.g., vehicle telemetry collectors). Using a combination of Bluetooth scan link-layer information fields such as Class-of-device, MAC address and Device name, we were able to uniquely identify skimmers at gas stations. We also find that signal strength is a reliable way to determine if a Bluetooth device is located near a gas pump, and thus could be a skimmer.

Our study reveals several problems with consumer Bluetooth-based skimmer detection applications [93, 2, 100]: (1) there are many legitimate products that appear at gas stations that use the same Bluetooth modules as known skimmers; therefore, MAC address-prefix based detection may lead to false positives, (2) there are many Bluetooth modules used in skimmers that do not comply with IEEE MAC assignment requirements. We also debunk advice on how to find skimmers with Bluetooth scans from authorities [4] and viral information from social media [74]. For instance, none of the skimmers we found using Bluetooth scans have a name that is a long string of letters and numbers.

Performing this in-depth study brought to light several important operational lessons learned about the importance of detecting skimmers with Bluetooth. Using Bluetooth scans, officials detected skimmers while driving by gas stations that they otherwise would not have inspected. We also witnessed several instances where an inspector tried to find a skimmer, but could not find it on their first pass looking inside a gas pump. However they persisted and found it based on the knowledge that a suspected skimmer had appeared in Bluetooth scans. Surprisingly, we observed that there are skimmers installed in the same gas station, or city, that have very similar MAC addresses—indicating their source is a single criminal entity. We even found skimmers installed hundreds of miles away that had surprisingly close MAC addresses.

The rest of the chapter is organized as follows: Section 2 provides background on internal gas pump skimming: their construction, monetary incentive, and prevalence in the wild. Section 3 is an overview of our large-scale Bluetooth scan collection methodology. In Section 4, we present the results of our study: what the skimmers we detected look like, how they compare to skimmers recovered independently by Law Enforcement, and whether they are well hidden in

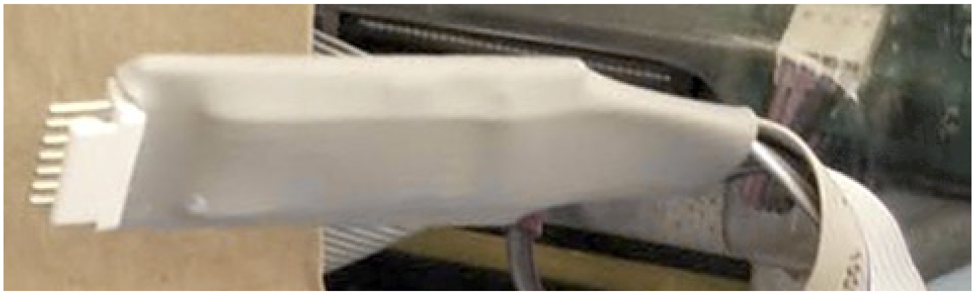


Figure 3.1. An internal Bluetooth-based skimmer wrapped in gray tubing to blend in with the cabling inside the fuel pump. This skimmer was detected by Bluetana in Tempe, AZ.

the Bluetooth environment. In Section 5, we present possible counter measures to the Bluetooth detection. In Section 6 we present the operational lessons we learned about skimming and criminal investigation procedure, while performing our large scale measurement study. Section 7 is related work, and we conclude in Section 8.

3.1 Background

Skimmers are illicit devices that capture credit card magnetic stripe data when a card is used at a point-of-sale (PoS) terminal or automatic teller machine (ATM). External skimmers use a magnetic head concealed in a false faceplate to read the magnetic stripe of a card as it is inserted into the real card reader. However, this paper is concerned with a newer class of skimmers, called *internal skimmers*, that are installed entirely inside a PoS terminal or ATM, leaving no visual evidence of its presence [94]. Internal skimmers are attached inline to the cable that connects the card reader to the main circuit board of the PoS terminal, tapping into the data and drawing power. To make data collection easier, many internal skimmers include a Bluetooth-to-serial module that allows the perpetrator to covertly collect the “skimmed” card data from a safe distance. These skimmers are built using commodity hardware with a total unit cost of $20 or less.

Fuel pumps with a built-in PoS terminal have become a very popular target for such internal skimmers: they are unattended, easy to access, and have poor physical security, which

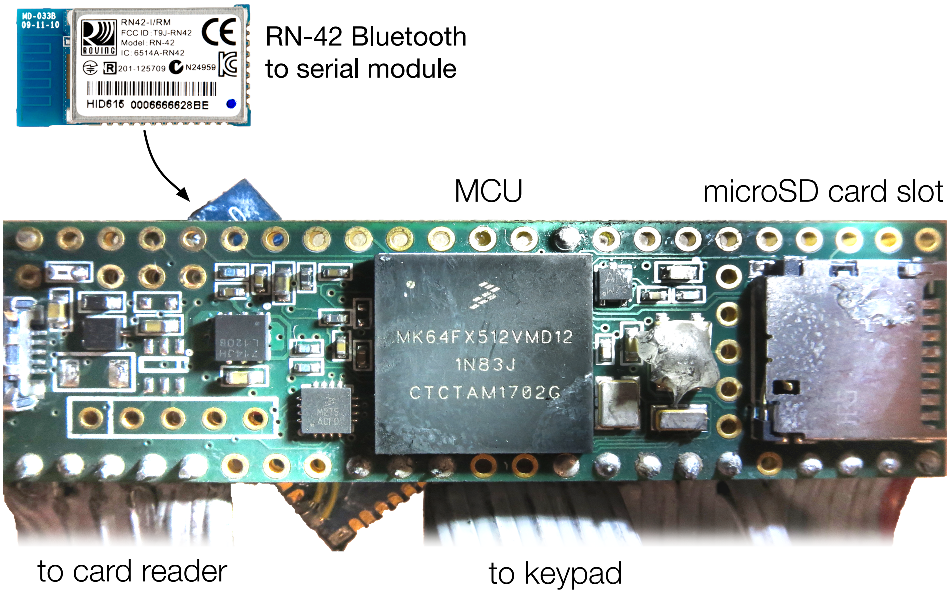


Figure 3.2. Parts of a typical internal Bluetooth-based fuel pump skimmer. This skimmer was detected by Bluetana.

make it easy to install a skimmer without being noticed. In a typical installation scenario, an attacker positions a van at a fuel station to block the station attendant's view of the target pump (Excerpt in A.2), opens the fuel pump using a common master key or crowbar, and clips a discreet gumstick-sized skimmer to the ribbon cable between reader and main circuit board using a vampire clip (Figure 3.1). The entire process to install skimmer can take less than 10 seconds [1]. The perpetrator can then return to the station with a smartphone, and without leaving their vehicle, connect to the skimmer using Bluetooth and download the card data.

3.1.1 Internal Bluetooth Skimmers

The subject of our study are *internal, Bluetooth-based skimmers* that are installed in fuel pump PoS terminals. Figure 3.2 shows a typical Bluetooth skimmer, recovered from a fuel station in Southern California. This skimmer consists of a "Teensy" development board with an ARM Cortex-M4F microcontroller and a Roving Networks RN-42 Bluetooth-to-serial module. It also includes connectors for tapping into the wiring inside the pump (not shown).

Connections. In the figure, the ribbon cable on the left intercepts or replaces the ribbon cable that connects the magnetic stripe reader to the PoS terminal main board. The skimmer also uses this connection for power: the power and ground pins of the Teensy (on far left of board, not visible in Figure 3.2) are connected to power and ground on the card reader cable. The ribbon cable on the right intercepts or replaces the ribbon cable from the PoS keypad. This allows the perpetrator to capture additional card verification data, namely the debit card PIN or credit card billing ZIP Code. Availability of a PIN code with a stolen debit card in particular, can increase its value five-fold on the black market (Table 3.1). However, not all skimmers capture keypad data.

Most gas station skimmers read the unencrypted data pulled from magnetic stripe readers. Card issuers feel that removing sensitive data from the magnetic stripe on cards will help to solve the problem [86]. Newer literature has demonstrated attacks on chip payment systems [18, 26], and law enforcement in Latin America have begun to find EMV skimmers that are Bluetooth enabled [69, 3].

Controller board. The skimmer pictured in Figure 3.2 used a Teensy microcontroller development board equipped with a 120 MHz ARM Cortex-M4F microcontroller made by Freescale Semiconductor. By using a development board, a skimmer requires only rudimentary electronic assembly: soldering wires to the development board.

However, skimmers have also been found using what appeared to be fully custom-designed boards. These are compact, making them better for hiding in the dispenser. Examples of micro-controllers used in recovered skimmers include Microchip PIC18F4550 [2] and Atmel XMEGA128A4U [3].

Storage. The Teensy board also has a microSD card slot for additional data storage. Skimmers built on custom PCBs have also used flash and EEPROM ICs for storage. The storage capacities vary across designs, with examples using the PCT25VF032B (32-Mbit) [3] and M25P16VP (16-Mbit) [2].

Bluetooth module. The skimmer shown in Figure 3.2 uses a Roving Networks RN-42 module, an inexpensive Bluetooth-to-serial module found in many skimmers. In Section 3.2.1 we describe characteristics of popular Bluetooth-to-serial modules used in recovered skimmers for wireless data exfiltration. On the Bluetooth side, a Bluetooth-to-serial module provides a Serial Port Peripheral interface, which most operating systems recognize as a Bluetooth modem and instantiate a serial device for it. Operating systems will create a corresponding serial device, allowing user-space applications, namely a criminal's card dumping application, to communicate with the module. On the hardware side, a Bluetooth-to-serial module provides a TTL-level receive and transmit pin, allowing it to interface to any microcontroller UART. The module this allows even the simplest microcontroller to communicate via Bluetooth with a host device. The 2.4GHz Bluetooth antenna is included on the module's circuit board (exposed area to the left of the metal shield for the module shown in Figure 3.2), so the antenna is also hidden.

Bluetooth-to-serial modules generally require no configuration, however, most can be reconfigured using Hayes-style modem AT commands. In Section 3.3.1 we describe the configuration capabilities of popular modules. Notably, all of the Bluetooth-to-serial modules we found in skimmers support changing the device MAC address, Bluetooth device name, changing the pairing password, and the ability to become non-discoverable once paired.

3.1.2 Economics of Carding

Stealing and monetizing stolen credit and debit card data, called *carding* by its practitioners, is a well-studied form of financial fraud, however, reliable estimates of losses resulting from a single skimmer are difficult to find. To the criminal operating a skimmer, the expected revenue per skimmer breaks down as:

$$W = (\text{card value}) \times (\text{cards per day}) \times (\text{days deployed}).$$

Of these, we found published estimates for only the first two quantities, and very little about skimmer lifetimes. Here, we summarize the available data with the goal of estimating the losses incurred by a single skimmer.

Card value. To monetize stolen credit card data, skimmer installers have two options: sell the data on the black market, or cash out the cards on themselves. Based on our survey of sites selling stolen card data, black market prices for stolen cards fall in the $10–220 range, depending on whether the card is a debit or credit card, and whether it comes with a PIN (for debit) or billing ZIP code (for credit). Table 3.1 provides a summary of these prices with references.

Criminals can also cash out the cards themselves. Debit cards with a PIN are often cashed out by withdrawing money from an ATM, while credit cards are often cashed out by purchasing high-value merchandise (e.g. iPhones) and re-selling them. Reported cash-out values for debit and credit cards range between $400 and $1,000, depending on credit limit associated with the card. We also conducted a survey of cash-out values reported in court documents involving skimmers.[3] Several cases reported specific cash-out values, rather than ranges. The debit card cash-out values were $1132 [80], $444 [34] $665 [8], $1354 [7]. The credit card cash-out values were $362 [98] and $400 [12].

Losses due to credit and debit card fraud are borne largely by banks and merchants. This is likely because consumer liability for fraud in the U.S. is limited to $50 for credit cards, and $50 or more for debit cards (depending on how quickly the consumer reports the fraud). Industry estimates for losses per-card incurred by banks are $650 for debit cards and, $1,003 for credit cards [1, 17]. The U.S. Sentencing Commission estimates per-card losses at $500 or more.

Cards per day. The number of cards a skimmer captures each day depends on the number of transactions at that pump, which will vary by station. Rippleshot, a payment fraud prevention service, states: “a single compromised pump can capture data from roughly 30–100 cards per day” [5]. The lower end Rippleshot’s estimate agrees with the estimate of 20–50 cards per day we received from U.S. law enforcement agents. In addition, we found two court documents

[3] We surveyed only documents available without fee from Court Listener.

Table 3.1. Value of stolen credit and debit cards.

Scheme	*Value*	*Reference*
Black market price		
Debit, no PIN	$20–30	[79, 96, 44, 88]
Debit with PIN	$110–220	[71, 96, 88]
Credit, no ZIP	$10–25	[79, 96, 44, 88]
Credit with ZIP	$25–60	[79, 96, 44, 88]
Cash-out value		
Credit or Debit (standard)	$400–800	[40, 83, 39, 110]
Credit (premium)	$1,000	[83, 92, 43]
Bank and merchant loss		
Credit	$1,003	[1]
Debit	$650	[17]
Consumer liability		
Debit (> 60 days)	unlimited	15 USC 1693g
Debit (< 60 days)	max $500	15 USC 1693g
Debit (< 2 days)	max $50	15 USC 1693g
Credit	max $50	15 USC 1643
Prosecuted loss		
Credit or debit	$500	[6]
Court documents		
Credit	$362–400	[80, 34, 8, 7]
Debit	$665–1132	[12, 98]

that report criminals captured 25 [12] and 30 [8] cards per day. We also studied 10 skimmers recovered from the field, which we were told were used and wiped daily. We found an average of 20 cards per skimmer, divided evenly between debit and credit cards.[4]

Days deployed. Internal skimmers are not limited by battery life and can remain in operational indefinitely, because they draw power from the PoS circuitry, Skimmer lifetime, then, is limited only by how long they can remain undetected. Unfortunately, there is little reliable data on this. Our only direct experience is our discovery of a pair of skimmers that remained undetected for six months (Section 3.2.1). However, LE informed us that criminals may leave skimmers in gas pumps after only a few days of retrieving card data and moving on to another location. Given the

[4] These skimmers were provided to us because they were removed by the station owner, rather than LE, making them unsuitable for use as evidence.

Table 3.2. Prevalence of skimming in three regions of the U.S.

Location & Year	*Recovered skimmers*	*Skimmed stations*	*Skimmers / station*	*Skimmers / 10^6 people*
San Diego				
FY 2018	42	11	3.2	11.9
Arizona				
2016	88	54	1.6	4.3
2017	57	46	1.2	2.7
2018	148	86	1.7	6.9
All	293	134	2.2	14.0
Florida				
2016	207	162	1.3	10.0
2017	650	432	1.5	31.1
2018	972	524	1.8	45.6
All	1,829	1,029	1.7	87.4

very limited data available on skimmer lifetimes, we instead consider skimmer value *per day of operation*.

Cashout success rate. Our analysis of court documents revealed that criminals are often unsuccessful when trying to cash out a skimmed card. This may be due to a variety of reasons, such as the following: incorrectly reading card data, hitting daily withdrawal limits, and activating fraud alerts. Several cases mentioned that criminals were not successful in cashing all skimmed cards. One case mentions a specific cashout success rate of 47% [7].

Total skimmer value. Finally, we estimate the range of per-day revenue from a skimmer based on the prior figures. Our low-end estimate is $4,253 (25 cards per day, cashout of $362 per card, and 47% cashout success rate), and our high-end estimate is $63,638 (100 cards per day, $1,354 cashout per card, and cashout success rate of 47%).

3.1.3 Skimmers Recovered in the Wild

To understand the prevalence of skimmers in the wild, we obtained data on recovered skimmers from three regions in the United States: San Diego and Imperial counties of California, with a combined population of 3.5 million; the state of Arizona, with a population of 7 million inhabitants; and the state of Florida, with a population of 21 million inhabitants. Table 3.2 summarizes the statistics. We note that these numbers do not represent *all* recovered skimmers. For San Diego and Imperial counties, our statistics represent the number of skimmers found by or reported to a U.S. federal law enforcement agency. For Arizona and Florida, our statistics represent skimmers found by or reported to the AZWMSD and the Florida Department of Agriculture and Consumer Services.

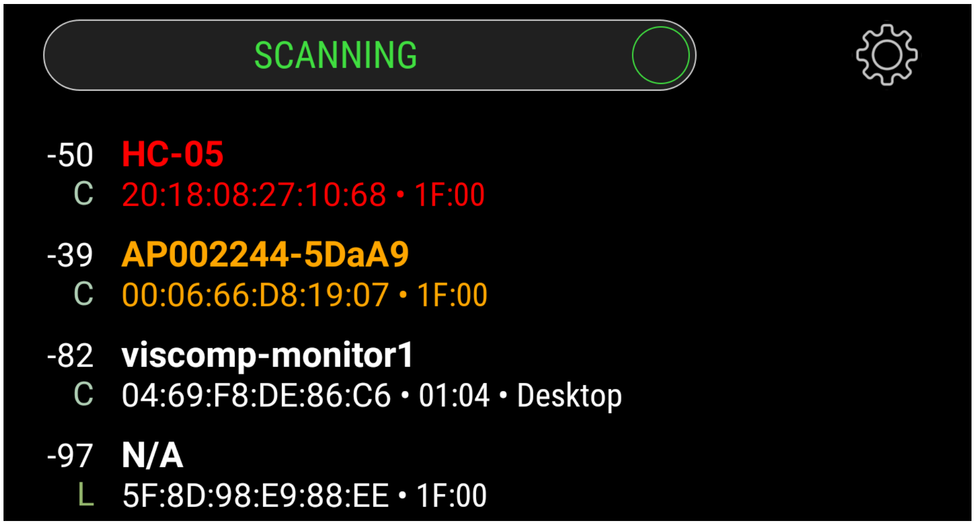
The number of recovered skimmers has increased from 2016 to 2018 in both Florida and Arizona. The total number of skimmers recovered in 2018 across the three geographic regions is significant: if each skimmer operated for just one day, we estimate their total monetary impact would be $17.43 million. Yet, as the skimmers-per-million people number shows, the possibility of an average consumer encountering a skimmer at a gas station is quite small.

3.2 Data Collection Methodology

Driven by the observation that skimmers are hard to find—few pumps in San Diego, Arizona, and Florida have been found to have skimmers installed in them (Table 3.2)—we created a tool, called Bluetana, to evaluate the effectiveness of Bluetooth-based skimmer detection. We begin by presenting an overview of the tool and the data it collects. Then we describe how Bluetana identifies suspicious devices and directs users to collect additional data. Finally, we discuss how we retroactively inspect data to find skimmers.

3.2.1 Crowdsourcing Bluetooth Scanning

We developed Bluetana, an Android-based measurement tool that officials and volunteers use to scan for skimmers at gas stations. Bluetana scans for nearby Bluetooth—both Classic and Bluetooth Low Energy (BLE)—devices every 5 seconds using Android's Bluetooth API. It collects the Bluetooth scans and geolocation data, and uploads this data to a secure database over a cellular link. Bluetana collects all the Bluetooth scan data that Android makes available,



```
-50  HC-05
  C  20:18:08:27:10:68 • 1F:00

-39  AP002244-5DaA9
  C  00:06:66:D8:19:07 • 1F:00

-82  viscomp-monitor1
  C  04:69:F8:DE:86:C6 • 01:04 • Desktop

-97  N/A
  L  5F:8D:98:E9:88:EE • 1F:00
```

Figure 3.3. The Bluetana user interface. Bluetana highlights suspicious devices, inspiring users to collect more signal strength samples, and even perform inspections.

including Device name, MAC Address, Class-of-Device[5], and signal strength (RSSI).

How we visited 1200 gas stations. We outfitted 44 volunteers and inspectors in six U.S. states (CA, AZ, MD, NC, NV, IL) with low-end smartphones running Bluetana in kiosk mode (they could not close the application). We selected officials who frequent gas stations as part of their daily job duties. Primarily, they were Weights and Measures inspectors.

Indicating suspicious devices to inspire data collection

The Bluetana display shows a list of Bluetooth devices detected during scanning. When Bluetana detects a potential skimmer, it indicates this to the user by highlighting the device record (Figure 3.3). The Bluetooth scan profile of the modules that have been found in skimmers inform which devices we highlight in Bluetana.

Skimmers recovered by LE are often found to use CSR (Qualcomm) chip-set-based Bluetooth modules. Our highlighting procedure primarily looks for the default Bluetooth profile of these modules—with the exception of the Device Name which can be missing due to poor signal strength, and modified by criminals in an attempt to hide the device (Section 3.3). The

[5] Class-of-Device is twenty four bits indicating the device's intended use, such as *smartphone* or *speaker*.

factory default Bluetooth scan profile (i.e., MAC prefix, Device Name, and Class-of-Device) of these modules are as follows:

Mod.	MAC Prefix	Dev. Name	Class of Dev.
RN	`00:06:66`	“`RBNT-*`”	Uncategorized
HC	*Various*	“`HC-05/06`”	Uncategorized

Bluetana chooses a highlight color via a three-step decision process, depicted in Figure 3.4. First, the app checks the device’s class. All skimmers studied within this work, whether discovered by Bluetana or not, had a device class of *Uncategorized*. If the device class is not uncategorized, the data is saved for later analysis. The device’s MAC prefix is then compared against a “hitlist” of prefixes used in skimming devices recovered by law enforcement. If the device has a MAC that is not on this hitlist, it is unlikely to be a skimmer, and the app highlights the record yellow. Next, if the device name matches a common product using the same MAC prefix, the record highlights in orange. If all three fields (MAC prefix, Class-of-Device, and Device Name) indicate the device is likely to be a skimmer, Bluetana highlights the record in red. The highlighting procedure is the result of a year of refinements based on our experience finding skimmers in the field, and Bluetana includes a remote update procedure to account for these incremental changes.

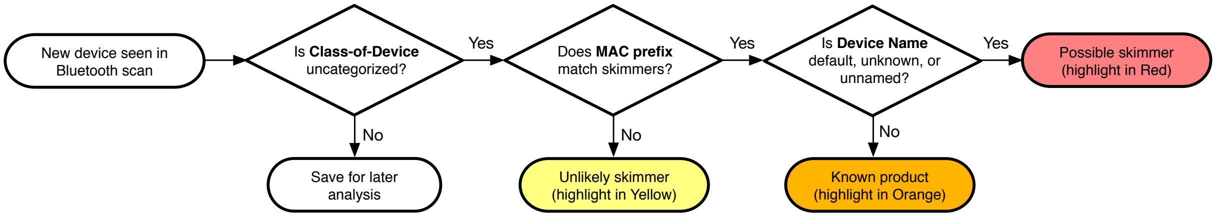
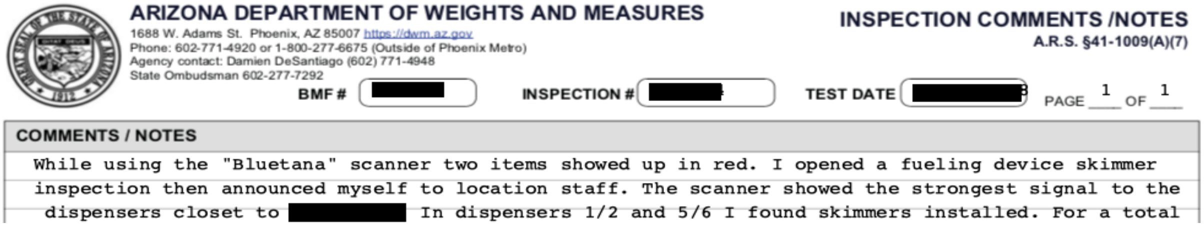

New device seen in Bluetooth scan
Is **Class-of-Device** uncategorized?
Yes
No
Save for later analysis
Does **MAC prefix** match skimmers?
Yes
No
Unlikely skimmer (highlight in Yellow)
Is **Device Name** default, unknown, or unnamed?
Yes
No
Known product (highlight in Orange)
Possible skimmer (highlight in Red)

Figure 3.4. The procedure Bluetana uses for highlighting suspicious devices.

This simple highlighting proved to be vital to our data collection. Red serves as a cue to perform signal strength localization: it directed our users to collect more samples of signal strength to determine if a device is located in the gas pump area—and is therefore likely to be a skimmer. In several cases, Bluetana highlighting a device in red was the only reason officials performed manual skimmer inspections: out of the 64 skimmers we found, 33 were recovered because an official started an inspection only after noticing a device was highlighted in red in Bluetana.

In one instance, an Arizona Weights and Measures inspector was driving by a gas station when two red highlighted devices appeared in Bluetana. He made an unscheduled stop at the gas station, performed a skimmer inspection, and discovered two skimmers. Figure 3.5 shows a portion of the official Arizona inspection report documenting this incident.



ARIZONA DEPARTMENT OF WEIGHTS AND MEASURES
1688 W. Adams St. Phoenix, AZ 85007 https://dwm.az.gov
Phone: 602-771-4920 or 1-800-277-6675 (Outside of Phoenix Metro)
Agency contact: Damien DeSantiago (602) 771-4948
State Ombudsman 602-277-7292

INSPECTION COMMENTS /NOTES
A.R.S. §41-1009(A)(7)

BMF # [redacted] INSPECTION # [redacted] TEST DATE [redacted] PAGE 1 OF 1

COMMENTS / NOTES

```
While using the "Bluetana" scanner two items showed up in red. I opened a fueling device skimmer
inspection then announced myself to location staff. The scanner showed the strongest signal to the
dispensers closet to [redacted] In dispensers 1/2 and 5/6 I found skimmers installed. For a total
```

Figure 3.5. Bluetooth scanning helps inspectors find more skimmers because they detect skimmers when driving by a gas station.

Bluetana's highlighting procedure is more comprehensive than other skimmer detection apps on the Play Store. Scaife et al. [93] investigated the behavior of these apps and found that they flag skimmers based on either MAC prefix or Device Name. These apps would miss skimmers with non-standard MAC prefixes or customized (missing) device names which Bluetana was able to find (Section 3.3.1). Bluetana also found legitimate devices that would be considered skimmers by these apps (Section 3.3.2).

Identifying skimmers after data collection

During the study, we manually examined every Classic Bluetooth device observed at a gas station visit in real time (as Bluetana users upload their scan data). At the beginning of our study, we relied primarily on the signal strength of the device to determine if it was a suspected skimmer. By the nature of being installed inside a gas pump, the Bluetooth signal of a skimmer is strongest in the pump area. Other devices that we suspected to be skimmers all had a low signal strength in the pump area, because aside from the cars parked at the pumps, the only

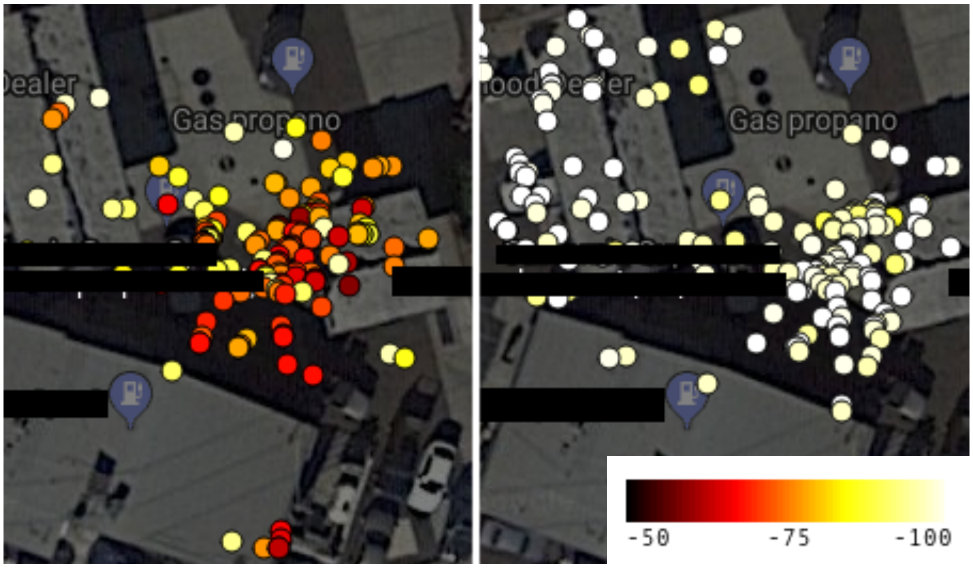


Figure 3.6. RSSI data overlaid on satellite view of a gas pump. Device on left has high RSSI near the gas pump and is likely a skimmer, device on the right is not.

places where a Bluetooth device would be located in the pump area would be inside the pump. Combining the signal strength and geolocation with satellite imagery of the gas station, we were able to easily detect when the signal was emanating from inside a gas pump (example shown in Figure 3.6). While at a gas station, Bluetana users also noticed this by moving toward the pump area to see if the device's signal strength increases.

If we saw any suspicious devices in the dataset, we alerted officials that they should inspect the pumps at the station in question. Initially, we did not know which of these devices were skimmers: many initial inspections we requested turned up empty-handed. However, as the study progressed, we improved our understanding of the profile of skimmers.

A natural experiment observing deployment duration

Having a database of all prior scans made it possible for us to look for skimmers that we may have missed in the past. In particular, looking back in at the database led to us to discover two skimmers that we had initially missed. A retroactive analysis of two stations discovered skimmers that were still operating even though we first detected them *six months* earlier. This natural experiment is likely the first concrete data on how long skimmers can be installed without being found in a routine or complaint-induced pump inspection.

3.2.2 Limitations

Selection bias

We designed our data collection to look for a specific type of gas pump skimmer: one that uses a Classic Bluetooth module, and is discoverable in Bluetooth scans. Our contacts in LE confirmed that this type of skimmer has been found in gas stations across the entire U.S. They also reported that these skimmers are particularly common in Arizona and California; therefore, these states were the focus of our study.

The results of our study may not be representative of the nature of gas pump skimming across the country. Criminals in other regions may evade Bluetooth-based detection by using alternate exfiltration methods (e.g., Bluetooth Low Energy and SMS), or configurations (e.g., non-discoverable mode). We outline these countermeasures in Section 3.4.

Bluetana does not connect to devices

We could collect more data about Bluetooth devices by trying to connect to them. This could be useful for conclusively detecting a skimmer or collecting information about the type of Bluetooth device. By sending commands that skimmers are known to respond to, Bluetana would be able to see if the device responds equivalently to known skimmers. This is precisely what one of the current Bluetooth skimmer scanning applications on the Play Store does.

This practice may seem innocuous, but our discussions with law enforcement indicate that this could overwrite information critical to future investigations. The problem is, internal registers in many skimmer Bluetooth modules records the last-paired MAC address. This information can be used to link a suspect possessing a smartphone or laptop with their skimmers. The typical forensic evidence collection performed by law enforcement on skimmers includes collecting the last-paired MAC address [95].

3.3 Results

In this section, we present the results of our 19 month study of Bluetooth devices observed with Bluetana at 1,185 gas stations across six U.S. states (CA, AZ, NV, MD, IL, NC). During the course of this study, Bluetana detected 64 skimmers operating in 34 gas stations; all of the skimmers were removed from the pumps by local and federal law enforcement agents. Bluetooth scanning is a surprisingly effective way of detecting skimmers: in Arizona, Bluetana has detected skimmers at 1.58% of the 491 stations it scanned, and routine inspections by state inspectors had a similar detection rate of 1.5% from 2016 to 2018.

The primary result of this study is as follows: there are distinct characteristics of the 64 internal skimmers detected by Bluetana that differentiate them from the 2,562 other Bluetooth devices that Bluetana found at gas stations (e.g., car stereos). Namely, these skimmers were predominately using the default Bluetooth module configuration. Additionally, we discovered that some criminals use a custom Device Name in an apparent attempt to hide their skimmers from Bluetooth scans. These custom Device Names stand out, making them easier to differentiate from other devices.

3.3.1 What Do Skimmers Look Like in Scans?

We begin by presenting how skimmers we observed appear in Bluetooth scans. We describe the properties of two sets of skimmers: 64 skimmers that we detected in the field during the course of this study, as well as 23 skimmers that were independently recovered by two LE agencies. The 23 skimmers recovered independently by LE have similar characteristics to the 64 that Bluetana detected in the field. The Bluetooth characteristics of these skimmers are detailed in Table 3.3. We now analyze the following properties: Class-of-Device, MAC prefix, and Device Name.

Table 3.3. Bluetooth scan properties of skimmers observed during our study. The exact Device Names are not shown, instead we describe the names we found.

	# of skimmers	
Bluetooth Scan Property	*Bluetana*	*LE*
Class-of-Device		
Uncategorized	64	23
Manufacturer (MAC prefix)		
Roving Networks		
00:06:66	45	13
Shenzhen Bolutek		
98:D3:31	1	
Unknown		
20:13:04	1	
20:17:11	1	
20:18:01	2	
20:18:04	1	
20:18:07	1	
20:18:08	4	10
20:18:09	4	
20:18:10	1	
20:18:11	2	
98:D3:35	1	
Device Name		
Default	36	23
[Law enforcement]	2	
[Mobile phone]	4	
[Indescript object]	2	
[Numerical]	2	
Unnamed	18	
Total	64	23

All of the skimmers are "Uncategorized" Class-of-Device

Class-of-Device is primarily used to select the icon that indicates the category of a device in a Bluetooth scan (e.g., Headphones). Bluetooth modules used in skimmers analyzed in this study (i.e., HC and RN), have an "Uncategorized" Class-of-Device assigned by default. Changing Class-of-Device on these modules is trivial: the modules provide a serial command to set it. Despite this, criminals do not appear to be modifying the Class-of-Device on any of the skimmers we observed: all of the 87 skimmers detected by Bluetana and recovered independently by LE used the default "Uncategorized" device class.

MAC prefixes are often manufacturer defaults

Bluetooth module manufacturers burn a MAC address into the module's EEPROM. Although it is possible to change the MAC with a SPI-based reprogramming of the CSR chip's EEPROM, we have not observed any skimmers that have a modified MAC. The first three bytes (prefix) of the MAC address typically correspond to the manufacturer of the device.

Although MAC address prefixes are often assigned by IEEE (e.g., all of the RN Bluetooth modules have the same manufacturer MAC prefix) the HC modules have a wide variety of MAC prefixes. Of the HC modules we observed, only one has a MAC prefix assigned by the IEEE. This could make it significantly more difficult to detect an HC-equipped skimmer. However, looking at of the MAC prefixes of the skimmers that we observed, a clear pattern emerges: manufacturers appear to be burning module manufacture date into the first four bytes of the MAC address in the following format: `YY:YY:MM:(DD)`.

Device names are often default, occasionally customized

Device Names allow users to identify their devices in Bluetooth scans. They are assigned a factory default value by the manufacturers, and are modifiable by users. Most of the skimmers we observed had a default Device Name: namely, all of the skimmers provided by LE, and more than half the skimmers we detected in with Bluetana. A skimmer with a default Device Name looks innocuous, because some legitimate products using the same modules are also shipped with the default module name (Section 3.3.3). Occasionally, we found that criminals set a custom device name on their skimmers. This appears to be an attempt to make the skimmer look less suspicious. Bluetana detected custom-named skimmers with a variety of names. The custom names of skimmers discovered by Bluetana had variety: some were random strings of numbers, and others masqueraded as LE.

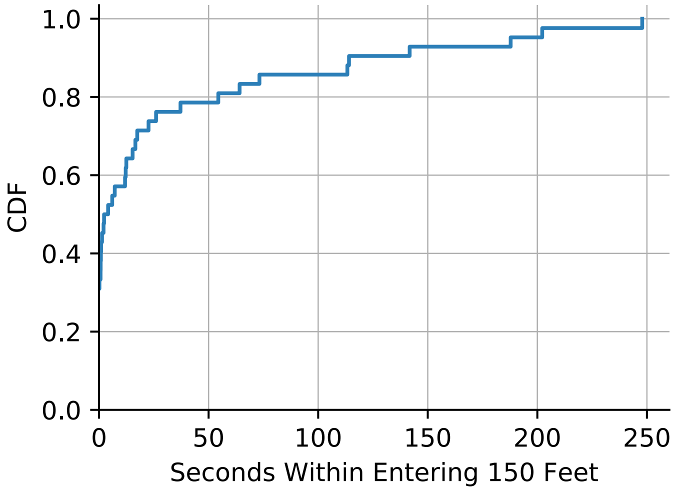


Figure 3.7. Skimmers are detected within a minute of passing near a gas station.

Bluetana did not detect a Device Name for several skimmers. This is expected because the device sends its MAC and Class-of-Device in the first scan response packet; it sends the device name in a subsequent packet (that may be missed).

Skimmers are detected within one minute

Bluetooth scanning has the benefit of detecting some skimmers without manually inspecting each of the pumps. However, attenuation from a gas pump's metal enclosure, may limit the range that Bluetooth scans are effective. We analyzed the scans from Bluetana to see how long an official had to spend at a gas station before they detected the skimmers installed there (Figure 3.7). The median time to detection was 3 seconds, and 80% of the skimmers were detected within one minute. This is a 99% decrease in search time compared to the average of 30 minutes that inspectors take to check a gas station for skimmers.[6], This result indicates that inspectors can quickly stop at gas stations to check for Bluetooth-detectable internal skimmers.

Table 3.4. On average there are two Classic Bluetooth devices seen at each gas station; infrequently, there are skimmers.

		Devices Observed				
State	*Stations*	*#*	*Avg.*	*Std.*	*Days*	*Skimmers*
CA	571	1148	2.01	1.94	152	22
AZ	491	1140	2.32	2.03	130	36
NV	38	93	2.45	3.44	21	4
MD	23	42	1.83	1.86	14	2
IL	18	37	2.06	2.01	13	0
NC	10	20	2	1.67	10	0

3.3.2 Are Skimmers Distinguishable in Scans?

Next, we evaluate if the skimmers detected by Bluetana were clearly distinguishable from the other devices observed at gas stations. The primary result of this study is that these skimmers were not hidden well. Many of these skimmers use the default configuration of their Bluetooth modules. Legitimate devices using the same Bluetooth modules may have some default parameters, and a few have all of parameters set to the default. We conclude that by combining multiple characteristics: MAC prefix, Class-of-Device, and Device Name, there are only a small number of devices that could be confused with skimmers.

This study also reveals that when criminals creatively modify their skimmer's Device Name, it makes detection easier. We also found that criminals could improve how they hide skimmers in Bluetooth scans. For example, they could change the Class-of-Device to hide as a more popular device (e.g., a smartphone).

Dataset Overview

Over the course of the 19 month study, Bluetana users visited 1,185 gas stations across six states (Table 3.4). During these visits, Bluetana detected a total of 64 skimmers—all of which were recovered by officials. These skimmers were in the presence of 2,214 other devices. On average, Bluetana saw 2.2 devices per station ($\sigma = 2.05$). Given that there are only a small

[6] Source: discussions with inspectors.

number of Bluetooth devices seen per station, it may seem likely that these devices are all skimmers. However, only a small fraction (3.98%) of these devices matched the characteristics of the skimmers we observed during the course of our study.

We performed this study on Classic Bluetooth devices only. We did not include BLE because we are not aware of any internal gas station skimmers using BLE modules. However, we observed a large number of BLE devices at gas stations; therefore, switching skimmers to BLE modules may make them more difficult to detect with scanning tools like Bluetana (Section 3.4.1).

For this analysis, we only include the scan data that is collected the first time a Bluetana user visits a station. Restricting the dataset in this way ensures fairness in our results. Analyzing all inspections may bias our observation of what Bluetooth devices tend to be found at gas stations to those that were visited multiple times. Specifically, we only analyze scans performed the first time Bluetana is near a gas station (within 150 feet) for at least 30 seconds and up to 5 minutes. 22 out of 64 of the skimmers were detected on subsequent visits to gas stations, so they are not included in this analysis.

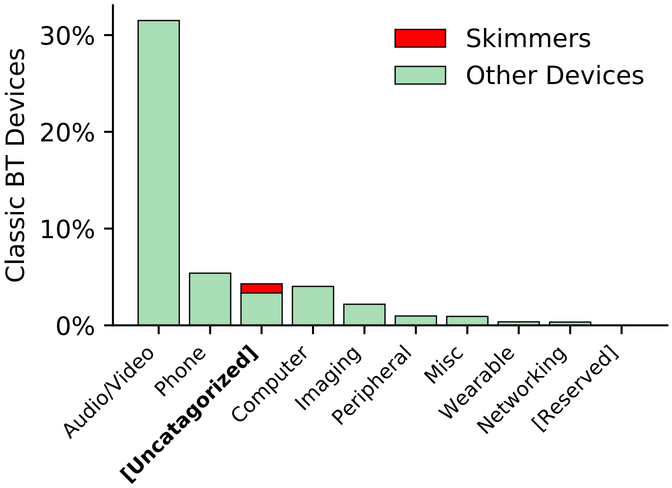


Figure 3.8. Skimmers appear in the third most common class of Bluetooth devices.

Skimmers are Uncategorized, but so are other devices

The only Bluetooth property that is common among all skimmers we observed is that they have an Uncategorized Class-of-Device. Figure 3.8 shows the distribution of Bluetooth device classes at gas stations. Uncategorized devices are the third most common Class-of-Device found by Bluetana (8.4% of devices). Although, out of the 1,185 gas stations that Bluetana users visited, Uncategorized devices were only observed at 143 gas stations (12.1%).

Other devices use the same modules as skimmers

Within the set of Uncategorized devices, we next look at the distribution of their MAC prefixes (Figure 3.9). We find that the Bluetooth modules used in skimmers are also used in many other legitimate devices. Specifically, more than half of the RN modules seen at gas stations were in skimmers, but there were many other devices that had RN modules. This is an important observation because a popular detection application, SkimPlus [100], only flags skimmers based on a hitlist of MAC prefixes [93]; it may incorrectly flag legitimate devices as skimmers.

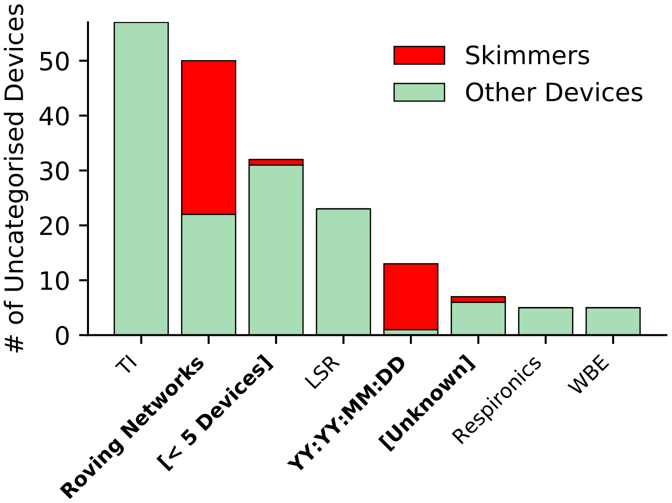


Figure 3.9. Many other devices appear to be using the same Bluetooth modules as skimmers.

The devices observed with MAC prefixes that were in the `YY:YY:MM:DD` format (likely HC modules) were mostly skimmers. There were many devices that had IEEE assigned MAC prefixes that were infrequently seen at gas stations (< 5 Devices). Only one of these devices was a skimmer. Also, there were many devices with MAC prefixes unknown to the IEEE, but not in the date format, only one of these devices was a skimmer. Overall, 102 devices out of 187 Uncategorized devices matched the MAC prefixes of Bluetana-observed skimmers. This reduces the number of stations where Bluetana detected skimmers to 79 out of the 143 stations where it found Uncategorized devices.

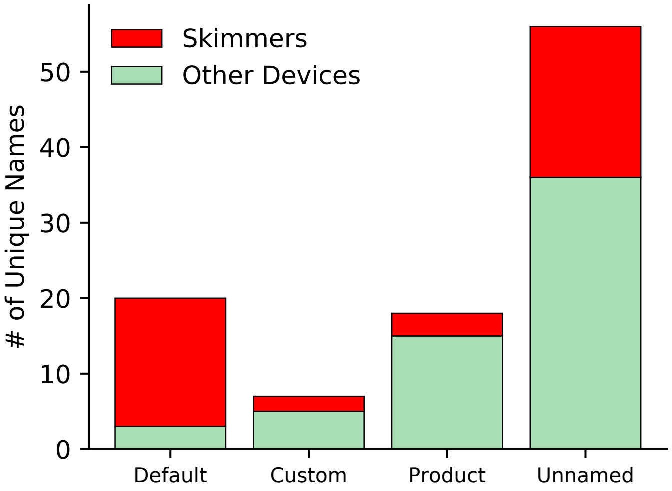


Figure 3.10. Default and custom names distinguish skimmers from legitimate devices.

Default- and custom-named modules are often skimmers

Finally, we investigate if skimmers can be differentiated from other devices by their Device Name. The remaining 102 devices are Uncategorized and their MAC prefixes are either: Roving Networks, `YY:YY:MM:DD`, Unknown, or seen on less than five devices. Only 42 of these devices were confirmed to be skimmers.[7] In Figure 3.10, we divide the remaining devices by their category of Device Name, including: *unnamed*, manufacturer *default*, known legitimate *product*, and *customized*. Devices observed by Bluetana with default names were often skimmers. Custom named devices were not common at gas stations but had a higher probability of being skimmers.

[7] We do not include 22 of the Bluetana-detected skimmers in this analysis because they were not detected on the first visit to a gas station.

Three skimmers were disguised as products, however all three were distinguishable because their names were popular smartphones, which should not have the MAC prefix of Bluetooth-to-Serial modules. Bluetana missed capturing the Device Name for many of the skimmers, as well as other devices that it detected.

3.3.3 Accuracy of Bluetooth-based Detection

To evaluate the accuracy of Bluetooth-based detection, we analyze Bluetana scan data collected during inspections in Arizona. Specifically, there was a 7-month time period in which Bluetana was used by many of the Arizona inspectors (October 7, 2018 – May 7, 2019), and we compare the reports filed during these inspections with the scan data that Bluetana collected.

Missed skimmers

During this time period, there were 27 inspections where skimmers were found while an inspector was running Bluetana. A total of 42 skimmers were recovered during these inspections, of which Bluetana was able to detect 36. Therefore, Bluetana missed detecting 14.3% of the total skimmers recovered during these inspections.

We do not know exactly why Bluetooth-based scanning missed these skimmers. Half of the missed skimmers were from inspections where Bluetana detected other skimmers at the gas station. It is likely that these missed skimmers were not powered on due to improper installation. The remaining missing skimmers may have been built with alternate exfiltration methods, such as SMS [93], or even require physical recovery [94].

Incorrectly detected skimmers

Bluetana highlighted a device in red during 45 Arizona inspections where no skimmer was found. There were 757 total inspections where inspectors used Bluetana[8], Bluetana may have incorrectly detect skimmers in 5.9% of inspections.

[8] This includes both routine and complaint/prior knowledge triggered inspections

Incorrectly identifying skimmers is likely due to the fact that RN and HC modules are used in a variety of legitimate products, some of which are seen in and around gas stations. We found RN and HC modules in radar-based speed limit signs, weather sensors [82] automotive diagnostic scanners, scales [81] and fleet tracking systems [106]. Some of these devices have Device Names that clearly indicate what product they are, but would be confused with skimmers if the Device Name is missing. Unfortunately, several of these products also use the default Device Names on their Bluetooth modules (*RNBT-xxxx* or *HC-05*). These legitimate devices will look exactly like skimmers. In such cases, inspectors will need to rely on RSSI localization to determine if these devices are located inside a gas pump.

3.4 Countermeasures and Responses

This work is a single snapshot in an evolving landscape of attacks on payment systems. While Bluetana has proven effective at finding Bluetooth skimmers, it by no means represents the last move in the cat-and-mouse game. In the remainder of this section, we discuss what the next few steps in this arms race might look like. That is, given that inspectors and volunteers are using Bluetana, what can be the skimmer installers' next move, its cost, and what our response might be. It is possible for a determined and resourceful criminal to implement the countermeasures that we will be describing (particularly non-discoverable mode).

3.4.1 Switching to Bluetooth Low Energy

We have observed that by switching to BLE, criminals have *many* more places to hide. Figure 3.11 shows the cumulative distribution of the number of BLE and Bluetooth devices we saw at each fuel station. Under the filtering of Section 3.3, over 8,000 unique BLE devices were seen, making the ratio of Classic to BLE approximately 1:4.

Cost to attacker. There is almost no cost to criminals in switching their Bluetooth modules to BLE. In fact, newer EMV skimmers discovered in other countries are BLE enabled [69].

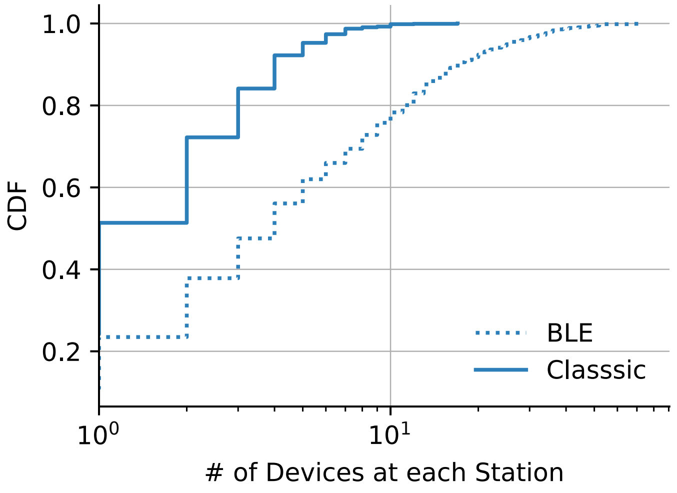


Figure 3.11. BLE devices are more common than Classic.

However, none of our contacts in law enforcement have encountered BLE-based gas station skimmers. It is possible that there is simply no incentive to switch: the same reason criminals have not yet adapted to masking their Bluetooth device class.

Response. BLE devices may be harder to differentiate due to the higher number of devices at each gas station and a lack of distinguishing features. With more sophisticated filtering techniques, it may still be possible to isolate BLE skimmers within this larger set of devices. One possibility is automated RSSI localization to the fuel dispenser location, a possible subject of future research.

3.4.2 Non-Discoverable Skimmers

The most natural way to evade discovery via Bluetooth would be to put the module in non-discoverable mode. When a Bluetooth device is non-discoverable, it does not respond to normal Bluetooth scans. Instead, it only responds to paging packets specifically addressed to its MAC address.

Cost to attacker. Non-discoverability would make exfiltration more difficult for criminals. One possibility is creating a pre-paired data collection device. However, we have been informed by law enforcement that the individual who installs the skimmer is often independent from the individual responsible for data recovery (called a “mule”). The criminal would not be able to send a mule to recover card data without first delivering them the device. Alternately the criminal could record the MAC address of the skimmer Bluetooth module. This would require careful bookkeeping and the use of tools that support the creation of a non-discoverable connection.

Response. It is still possible to discover a non-discoverable device. For a small set of target address ranges, e.g., `00:06:66` used by Roving Networks modules, we believe it would be practical to attempt to guess all 16.8 million possible addresses. Prior work has shown that it is possible to discover any non-discoverable device via brute force in 18.64 hours; knowledge of OUI would ideally allow us to reduce this search time [35]. Unfortunately, this requires specialized hardware, rather than an inexpensive Android phone.

3.4.3 Impersonating Common Benign Devices

Another natural response to Bluetana would be to change the MAC address and name of the device to that of a common benign device, such as a mobile phone or a Bluetooth-enabled car entertainment system. This would make the skimmer appear innocuous to Bluetana.

Cost to attacker. Reprogramming the MAC address on the CSR-based Bluetooth modules, which include the Roving Networks and HC-05 and HC-06 modules, cannot be done using the AT commands used to change device name and pairing. Instead, the skimmer installer would need to re-flash the CSR firmware using a special programming cable. While, in principle, not difficult, it would require an additional degree of sophistication than programming a simple micro-controller development board. The skimmer installer could also change the device name but not the MAC address, say, to one of the known benign devices using the same module, something that us possible to do by issuing AT commands from the micro-controller to the module. While this may cause Bluetana to detect these as a skimmer, signal strength can still be used to identify location of the module.

Response. Because Bluetana collects all Bluetooth data, we can identify skimmers retroactively when we learn of a new MAC address and name used by known skimmers. Thus, if attacks switch to impersonating benign devices, we can update the Bluetana highlighting mechanism to identify those devices as suspicious. This would result in additional inspections, but would still provide significant gain over the state of the art.

3.4.4 Using Non-Bluetooth Communications

During discussions with law enforcement agencies tasked with identifying skimmers, we were told about skimmers that use GSM modems or WiFi as an alternative to Bluetooth. In the case of WiFi, we believe that the Bluetana methodology will still be effective. GSM poses a more serious challenge for detection.

Cost to attacker. While using GSM would avoid detection using Bluetana, it creates an additional trail of evidence linking the perpetrator to the skimmer. Law enforcement officers could obtain information about the SMS recipient through subpoenas, so receiving the SMS messages on another phone on a US carrier, for example, would be easy to trace.The perpetrator would need to use an SMS service that would not expose his/her identity.

Response. In addition to legal tools available to law enforcement to trace SMS messages, a GSM modem could be detected using a Software-Defined Radio.

3.4.5 Attacker Bottlenecks

The attacker (skimmer installer) has several practical ways to evade detection using Bluetana. Each of these, however , has an additional cost in terms of effort, risk exposure, or expertise. We do not yet have a strong understanding to which of these costs attackers are most sensitive. Indeed, the very low price of stolen credit card numbers, compared to their potential cash out value (Table 3.1) suggests that the bottleneck in the carding value chain is *not* getting card information but cashing out cards. Thus, while Bluetana may raise the cost for attackers, we do not believe that it will raise it so much as to make fuel dispenser skimming unprofitable.

3.5 Operational Lessons Learned

While performing the Bluetana study, we learned several lessons about the operational use of Bluetooth scanning for skimmer detection. In this section, we provide an overview of two most important lessons we learned.

3.5.1 Bluetooth Helps During Inspections

Criminals hide skimmers in the crevices of gas pumps to avoid detection during inspections. We witnessed several instances where investigators were unable to locate skimmers via physical inspection alone. In one incident, Bluetana flagged four devices at a station; however, no skimmers were located. This result led officials more experienced in skimmer recovery to perform a second thorough inspection of the station. These officials located all four skimmers. The evidence provided by Bluetana forced them to continue the inspection, instead of abandoning it and leaving the devices in the field.

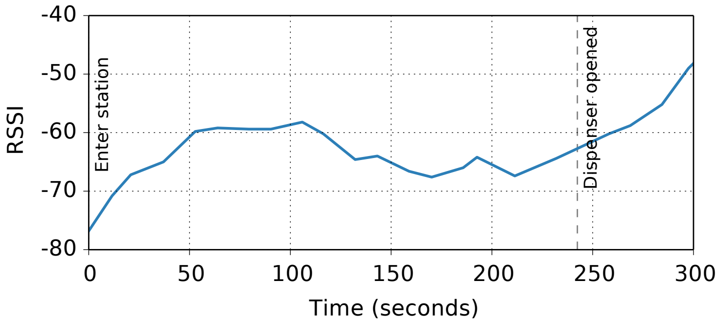


Figure 3.12. Opening of the gas pump enclosure results in a significant jump in observed Bluetooth signal strength from a skimmer.

Figure 3.12 demonstrates an instance of how the signal strength measurements helped inspectors determine which pump had a skimmer. When the gas pump's metal door was opened, the signal strength increased significantly, prompting inspectors to look carefully for the skimmer in that pump.

3.5.2 MAC Addresses May Indicate the Source

Network equipment vendors (e.g., Bluetooth module manufacturers) tend to allocate MAC addresses sequentially by production time [77]. Therefore, if two devices have similar MAC addresses, they are likely part of the same batch of devices sold. This information can be used to associate skimmer Bluetooth modules to the same board designer or crew.

Table 3.5. Several geographically separated skimmers had similar MAC addresses.

	Group				
	1	**2**	**3**	**4**	**5**
Skimmers	3	5	6	4	3
Gas stations	2	2	5	4	2
Min. difference in MACs	1	4	9	10	4
Closest MAC distance (in miles)	0	17	59	203	448

We group the skimmers found by Bluetana with the same first 5 bytes of MAC address. Table 3.5 shows five such groups. We list the difference in MAC address and the geographic distance between the closest MACs in each group. Skimmers in group 1 and 2 were recovered at gas stations in the same county, separated by at most 17 miles. From LE sources, we know that criminals often plant skimmers across multiple stations in a given city/county, and the MAC address data collected indicates this. Groups 3-5 are the most interesting, as the closest MACs in the same group are in stations across different counties. The closest MACs in group 5 are at stations separated by 448 miles. This may seem surprising, but LE informs us that skimmer crews avoid detection by migrating from city to city.

3.6 Future Work and Conclusion

As new skimmer detection tools gain popularity, criminals will adapt skimming designs to evade detection. We expect future skimmers will use techniques such those described in Section 3.4. Similar to Bluetana, future work in this area should emphasize designing easy-to-deploy systems for detecting skimmers, and evaluating their effectiveness with large-scale studies.

Push-back from banks and card issuers has led to wide-scale adoption of EMV in retail PoS systems. However, EMV adoption in gas stations across the U.S. has been slow due to high costs. Therefore, Visa and Mastercard have pushed the EMV adoption deadline for gas stations from 2017 to October 2020 [45]. As gas stations begin migrating to EMV, skimmers targeting EMV will become more common. Future research should focus on the detection of EMV "shimmers" that are gaining in popularity.

Finally, we believe gas pump skimming is the harbinger of an era of attacks using illicit wireless access links. For example, there is an internal Bluetooth-based implant for unlocking door access control systems [20]. Future work should also identify other systems that are vulnerable to using such illicit links.

In this chapter, we presented results of a 19-month-long measurement study of Bluetooth scanning as a mechanism to detect illicit internal gas pump skimmers. Our evaluation showed that link layer characteristics of Bluetooth-based internal skimmers can be distinguished from other Bluetooth devices commonly seen at gas stations. We detected, and LE recovered, 64 skimmers at 34 gas stations across four states in the U.S. For 33 of the detected skimmers, Bluetana was the only source of information that prompted investigators to conduct an inspection. In conclusion, link-layer information revealed in Bluetooth scans is effective at detection of illicit wireless access links at public locations, even in the presence of tens of other Bluetooth devices.

Chapter 3, in part, is a reprint of the material as it appears in *Usenix Security Symposium 2019.* Nishant Bhaskar, Maxwell Bland, Kirill Levchenko, and Aaron Schulman. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Physical-Layer Wireless Signal Information for Tracking a Mobile Device

Wireless scans also reveal physical-layer information from the wireless signal, that can uniquely identify a wireless transmitter [29, 59] Attackers can misuse these wireless transmitter properties to perform targeted tracking of wireless devices. In this chapter, I describe the results of a field measurement study to understand the feasibility of such a targeted tracking attack in real-world public locations, when the target wireless access link is “hidden in the noise” of hundreds of other wireless devices. In particular, the field study was aimed at demonstrative effectiveness of a tracking attack on a specific type of wireless access link — Bluetooth LE enabled personal mobile devices.

The mobile devices we carry every day, such as smartphones and smartwatches, increasingly function as wireless tracking beacons. These devices continuously transmit short-range wireless messages using the Bluetooth Low Energy (BLE) protocol. These beacons are used to indicate proximity to any passive receiver within range. Popular examples of such beacons include the COVID-19 electronic contact tracing provided on Apple and Google Smartphones [30] as well as Apple’s intrinsic Continuity protocol, used for automated device hand-off and other proximity features [9].

However, by their nature, BLE wireless tracking beacons have the potential to introduce significant privacy risks. For example, an adversary might stalk a user by placing BLE receivers near locations they might visit and then record the presence of the user's beacons [11, 107]. To address these issues, common BLE proximity applications cryptographically anonymize and periodically rotate the identity of a mobile device in their beacons. For instance, BLE devices periodically re-encrypt their MAC address, while still allowing trusted devices to determine if these addresses match the device's true MAC address [25]. Similarly, COVID-19 contact tracing applications regularly rotate identifiers to ensure that receivers cannot link beacons from the same device over time [10].

While these mechanisms can foreclose the use of beacon content as a stable identifier, attackers can bypass these countermeasures by fingerprinting the device at a lower layer. Specifically, prior work has demonstrated that wireless transmitters have imperfections introduced in manufacturing that produce a unique physical-layer fingerprint for that device (e.g., Carrier Frequency Offset and I/Q Offset). Physical-layer fingerprints can reliably differentiate many kinds of wireless chipsets [38, 29, 54, 104, 87, 62, 84, 27], including a recent attempt to distinguish 10,000 WiFi [59] chipsets.

However, no prior work has evaluated the practicality of such physical-layer identification attacks in a real-world environment. Indeed, prior to BLE tracking beacons, no mobile device wireless protocol transmitted frequently enough—especially when idle—to make such an attack feasible. In contrast, today it is common to find tens and hundreds of personal devices transmitting these BLE beacons at all public locations – office buildings, public library, coffee shops and others. For an attacker, even with a precise fingerprint, locating one target device in such public locations is a needle in a haystack problem — we don't know the limitations to uniquely differentiating one BLE device in this "noise" of several other BLE devices.

In this chapter, we take an empirical measurement approach to understanding the practicality of this tracking threat. We develop a technique to estimate high precision fingerprints from BLE beacons. We then perform BLE beacon data collection in lab on devices that we control, and also uncontrolled data collection of BLE beacons from mobile devices seen at a variety of public locations. Using our fingerprint technique, we analyze these beacons from real-world devices to understand the scope of this privacy threat, and how likely is an attacker at being successful in locating their target. In particular the contributions of our work are as below:

1. Using lab-bench experiments on BLE devices we own, we identify four primary challenges to identifying BLE devices in the field: (1) BLE devices have a variety of chipsets that have different hardware implementations, (2) applications can configure the BLE transmit power level, resulting in some devices having lower SNR BLE transmissions, (3) the temperature range that mobile devices encounter in the field can introduce significant changes to physical-layer impairments, and (4) the low-cost receivers that an attacker can use in the wild for RF fingerprinting may be significantly less accurate than the tools used in prior studies [29].

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

Through these empirical studies, we show that even when there are hundreds of devices we encountered in the field, it is still feasible to uniquely identify a specific mobile device by its physical-layer fingerprint. However, we also observe that certain devices have similar fingerprints to others, and temperature variations can change a device's metrics. Both of these issues can lead to significant confusion in distinguishing mobile devices. In summary, we find that physical layer tracking of BLE devices is indeed feasible, but it is only reliable under limited conditions, and for specific devices with extremely unique fingerprints, and when the target device has a relatively stable temperature.

4.1 Background

In this section, we define the threat model for a tracking attack on a BLE based mobile device. Following that we provide details on how extensive the threat is by exploring how all popular personal mobile devices are continuously and frequently transmitting BLE beacons.

4.1.1 Threat model: Passive fingerprinting of BLE mobile devices

We consider an attacker that intends to detect when the target – a particular user possessing the target mobile device — is at a specific location (e.g. a room in a building or a crowded public place). The attacker must possess a software-defined radio (SDR) to capture the raw I/Q data of the BLE beacons transmitted by nearby mobile devices. Even though a lot of SDR tools are expensive, we show in Section 4.3.4 that a modest hobbyist-level SDR ($\sim$\$150) is sufficient for the attacker.

The attacker first captures a fingerprint of their target's mobile device. They do so by getting close to them and capturing BLE beacons from their mobile device. Then they use these BLE beacons to estimate unique physical-layer properties of the BLE transmitter hardware, such as CFO and I/Q offset – these define the fingerprint of the target mobile device.

Armed with the fingerprint of their target, the attacker sets up the receiver at the eventual attack location where they want to track the target. The attacker captures beacon packets from all mobile devices at the location, estimates the physical-layer properties and compares it to the target fingerprint. If the fingerprint matches, the attacker knows that the target is at the location. The more frequently the BLE device transmits, the more likely the attacker is to receive a transmission if a user passes by. Also, the more accurate the fingerprinting technique is, the better the attacker can differentiate the target from other nearby devices.

Table 4.1. BLE beaconing behavior of popular mobile devices.

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

*Only beacons with COVID-19 contact tracing enabled.

4.1.2 Extent of threat: Popular mobile devices are vulnerable

Increasingly, mobile devices are adding BLE beacons to provide new features. Most notably, during the COVID-19 pandemic, governments have installed software on iPhones and Android phones to send constant BLE advertisements for digital contact tracing: devices listen for nearby transmissions to determine if and for how long another device was nearby. Also, Apple and Microsoft operating systems have recently added BLE beaconing to their devices for two inter-device communication features: lost device tracking, and seamless user switching between devices (e.g., Apple's Continuity Protocol, Microsoft's Universal Windows Platform) [23]. Therefore, BLE beacons are now common on many mobile platforms, including: phones, laptops, and smartwatches.

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

1. **Mobile devices send BLE beacons continuously:** We observed continuous BLE beaconing from all the six mobile devices shown in Table 4.1. Even when all of these mobile devices have their screens off (e.g., they are in their user's pocket), they continuously

transmit BLE beacons. Indeed, this is a feature that is necessary for the proper function of the beacon applications such as contact tracing. Continuous beaconing is a significant new threat compared to the behavior of other protocols on mobile devices that only transmit intermittently (e.g., periodic WiFi scanning).

2. **Mobile devices send hundreds of BLE beacons per minute:**

   Table 4.1 also shows the average number of BLE beacons (i.e., BLE advertisements) we observed per minute from each device. We observe that all of these devices transmit frequently—hundreds of packets per minute—even when the device is otherwise idle (e.g., screen off). Transmitting hundreds of advertisements per minute makes it feasible to produce a physical-layer fingerprint quickly: even if the device is in range of the sniffer for a few seconds (Section 4.4).

4.2 BLE Tracking Toolkit

In this section I present a high-level overview of the algorithm we use in this work to estimate the physical-layer properties of the BLE transmitters. The details of the algorithm are outside the scope of this dissertation, and therefore I only present the high level intuition to obtaining high-precision fingerprints from BLE beacons.

The hardware imperfections that lead to the fingerprint arise from the underlying manufacturing variations in BLE transmitter hardware. These manufacturing variations lead to non-idealities in the received BLE beacon signals, which we can measure to derive the fingerprint. In particular, most mobile devices feature an integrated single-chip WiFi+ BLE transmitter, which has a shared I/Q frontend. Therefore, the BLE transmissions are impacted by the same hardware imperfections as the WiFi transmitter. For our work, we explore the following specific hardware imperfections:

1. **CFO:** The Carrier Frequency Offset (CFO) is a shift in the carrier frequency from the

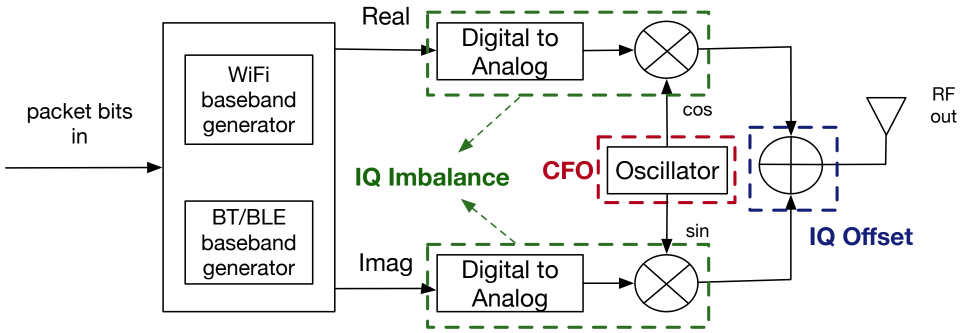


Figure 4.1. Architecture of WiFi/BLE combo chipsets

ideal channel value. This arises due to the frequency error of the crystal oscillator that is used to generate the carrier signal that feeds into the mixer in the RF frontend.

2. **I/Q imperfections :** I/Q Offset happens due either the leakage of the carrier signal onto the transmitter output due to non-idealities of the mixer hardware, or due to an DC offset on the baseband signals. I/Q Imbalance is a deviation in amplitude and phase of transmitted signal, due to the mismatch between similar analog components on the in-phase and quadrature-phase paths.

Figure 4.1 shows the architecture of a typical BLE transmitter, and the sources of the imperfections described above.

Unfortunately, we can't reuse techniques from prior work on WiFi physical-layer fingerprinting to measure these properties precisely for Bluetooth LE. Prior techniques rely on the presence of a long known sequence or preamble as a reference to measure the signal distortions due to CFO and I/Q imperfections accurately BLE has a very short preamble and that leads to extremely inaccurate estimates of CFO and I/Q from prior techniques.

However, a key insight about BLE decoding helps us. Unlike WiFi, BLE uses simpler GFSK modulation and does not require us to compensate CFO and I/Q imperfections before decoding. Consequently, we can decode the entire BLE beacon packet and obtain the full bit sequence correctly. This bit sequence can be used to create a reference signal which is much longer (packet length), and that can provide us improved estimates for our fingerprint.

With a longer reference signal available as a starting point, the fingerprint estimation algorithm estimates the hardware imperfections. Starting with the initial pure signal, the algorithm iteratively adds CFO and I/Q imperfections until our pure starting signal looks similar to the received signal. To do so it models the imperfection estimation as an optimization problem, with the BLE signal modelled under impact of the imperfections. Using this approach, we were able to achieve high precision estimates as compared to using just 8 bits of preamble. Furthermore, the estimates over a packet are obtained as an average across all the raw samples in the packet, which minimizes impact of SNR changes, resulting in robust estimates of CFO and I/Q imperfections.

Finally, for the actual tracking attack the attacker estimates CFO and I/Q from multiple beacon packets from the same device. The actual fingerprint is represented as a distribution of CFO and I/Q across multiple packets. When actually tracking the target at the destination, the statistical distance of the distributions of a newly observed device and target are compared against a threshold.

4.3 Real-world challenges to physical-layer identification

Using our high-precision fingerprint technique, we perform an empirical analysis in lab conditions to understand the limitations of this physical layer identification. There are five primary challenges that limit the effectiveness of tracking BLE devices based on their physical-layer fingerprint. For each challenge, we perform controlled experiments or theoretical analysis to investigate how significantly they affect fingerprinting accuracy in practice, and in turn the ability of an attacker to uniquely identify their target. We found that BLE tracking attacks are likely to be feasible in practice. However, the attacker's ability to identify a specific device reliably will vary depending on several factors that are out of their control.

4.3.1 Uniqueness of BLE fingerprints

BLE transmitters must have unique imperfections if an attacker wants to differentiate their target from other nearby devices. To evaluate how similar BLE fingerprints are in practice, we

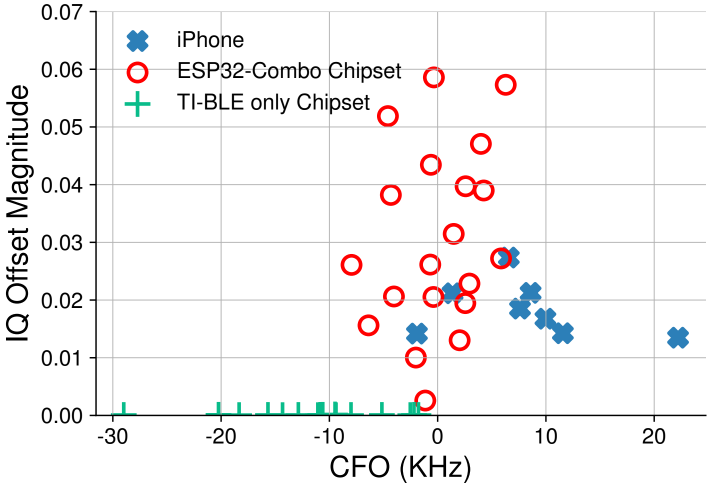


Figure 4.2. Comparing the fingerprints of 48 BLE chipsets

compare the fingerprint of many devices across three different popular BLE chipsets. Specifically, we captured the fingerprint of eight recent iPhones with WiFi+BLE combo chipsets, 20 ESP32 WiFi+BLE microcontroller chipsets, and 20 TI CC2640 BLE-only chipsets used in low-power devices (e.g., fitness trackers). We captured 100 packets using a high-quality SDR (USRP N210) from each of these devices in a controlled environment (i.e., an RF isolation chamber). We computed the fingerprint of each device across all 100 packets using the methodology described in the previous section.

Figure 4.2 shows the mean of the fingerprint metrics for each of the 48 devices. We plot only the CFO and I/Q offset metrics to simplify the visualization, adding I/Q imbalance does not change the conclusions of the experiment. Overall, most of the 48 devices have unique fingerprints. However, there are a few devices that have similar fingerprints, making them more difficult to uniquely identify. The distribution of device fingerprints also appears to be dependent on the chipset. Namely, there are striking differences in how the I/Q offset metric is distributed between different chipsets. For instance, the ESP32 devices have a much larger range of I/Q offsets than the iPhones, which may be because ESP32s are low-end chipsets compared to the high-performance WiFi+BLE combo chipsets used in iPhones.

Surprisingly, the TI BLE-only chipsets all have negligible I/Q offset. Recall in Section 4.2,

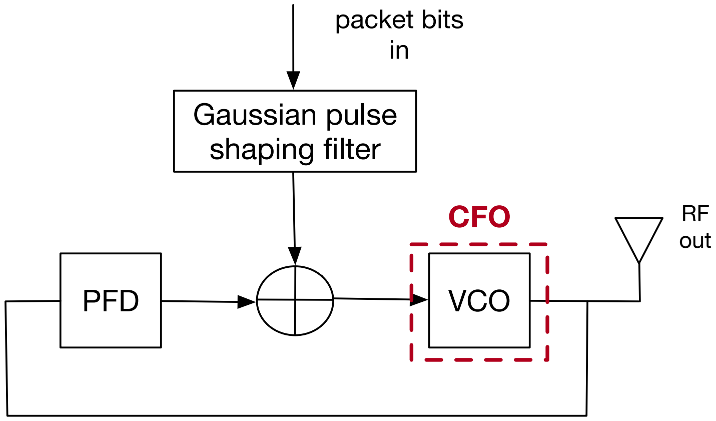


Figure 4.3. TI's BLE-only transmitter. This is not an I/Q modulator.

we described how unlike WiFi, BLE is not an inherently I/Q modulated protocol; therefore, the TI's BLE-only chipset may have I/Q offset because it may not use an I/Q modulator. We confirmed this suspicion by finding a technical report that describes the TI BLE chipset radio architecture: it uses a PLL-based (non-I/Q) modulator [105].

Summary

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

4.3.2 Temperature stability of BLE fingerprints

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

Smartphones are particularly exposed to temperature variations. Their internal temperature can significantly change due to internal components heating up (and cooling down) when activity changes, and they also experience a variety of ambient temperatures [61]. However, it is possible that smartphones do not have instability in their BLE transmissions. The impact of temperature on CFO is dependent on the cut angle and face of the crystal [36], and smartphones may use high-quality crystals that have less frequency drift due to temperature changes. Also, smartphones may use temperature compensated crystals as they may be required for high-data rate cellular communication chipsets.

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

Figure 4.4 shows the per-packet variation in CFO and IQ offset during the 15-minute tests. We do not show the variation in I/Q imbalance as it as we found it has a similar relationship to temperature as I/Q offset. For the game experiment, we observe that the CFO has a linear relationship to the changes in temperature. When the game begins, the CFO increases, and when the game ends, it decreases. At the peak internal temperature (+10°C above baseline), we observe a significant CFO deviation (7 kHz). For the in-pocket experiment, the peak change in CFO is much less than the game experiment (2 kHz). However, it is still significant enough to introduce confusion with other devices that have similar I/Q metrics (Figure 4.2). Finally, figure 4.4 show that I/Q offset (and I/Q imbalance which is not shown) does not correlate with

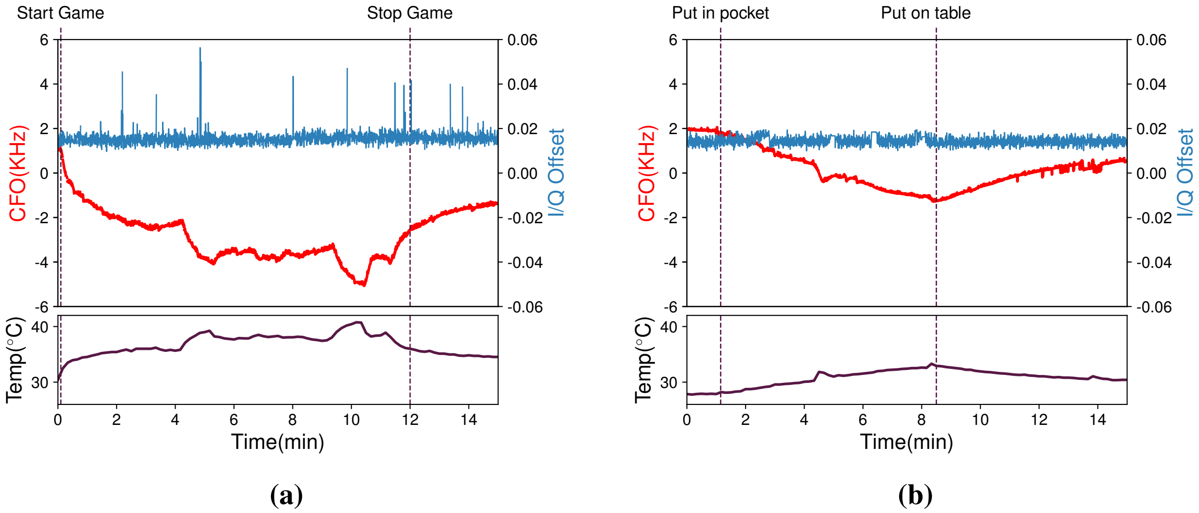


Figure 4.4. Stability of CFO and I/Q offset when (a) playing a GPU-intensive game and (b) putting the phone in a pocket

temperature in both the cases.

Summary

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

4.3.3 Differences in BLE transmitter power

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

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

Figure 4.5 shows that the iPhone 8 has an SNR 10 dB higher than all other Android phones we tested. Therefore, the iPhone's BLE beacons are likely to be received considerably farther away than the other devices. Anecdotally, we observed that an iPhone's COVID–19 contact tracing beacons 7 meters farther than any of the Android devices we tested[1].

Summary

There can be significant differences in BLE transmit power across devices, and even across apps running on devices. We observed that iPhones transmit COVID–19 contact tracing beacons with significantly higher power than Android devices. Consequently, attackers may be able to track iPhones from a farther distance than Android devices.

4.3.4 Quality of an attacker's sniffer radio

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

[1] Including other versions of the iPhone available at the time (e.g., Xr).

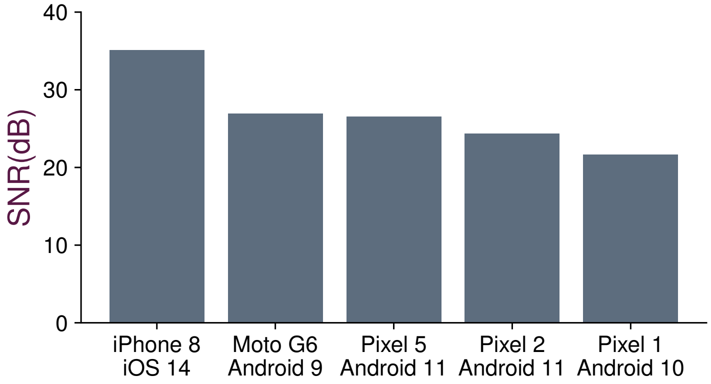


Figure 4.5. SNR of COVID contact tracing beacons across devices

their target.

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

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

CFO

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

I/Q metrics

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

Summary

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

4.3.5 Mobility of target device

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

Physical environment

A change in the physical location of the target can alter the received signal's SNR due to changes multipath conditions. However, we observed that this appears to have an insignificant impact on BLE fingerprinting metrics. We have observed through experiments that above a certain minimum SNR ($\sim$10 dB), changes in SNR do not impact identification accuracy.

Speed of Motion

A moving BLE device may experience a velocity-dependent frequency offset due to the Doppler effect [114]. While this may cause a slight drift in the CFO of the BLE target device, the impact is not significant for the frequencies that BLE operates at.

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

Summary

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

4.4 Field Evaluation

Several of the challenges described in the previous section raise the possibility that there are realistic scenarios where an attacker may not be successful in identifying their target device. Determining how often these errors happen in practice requires us to do a field study in real-world locations. Fortunately, BLE devices constantly beacon, and these beacons contain an anonymous identifier that is stable for 15-minutes. We leverage these properties of BLE to perform a large-scale uncontrolled field study of how likely is it for an attacker to be confused when searching for a target device.

To begin with, we assess how well our BLE tracking toolkit works, even though devices may not have unique fingerprints, and their fingerprint can be affected by temperature variations. We then provide a multi-day uncontrolled field study that shows the uniqueness of CFO and I/Q offset for mobile devices when observing several hundreds of mobile devices. To the best of our knowledge, this is the first uncontrolled experiment to evaluate the effectiveness of a physical-layer tracking attack in practice.

Data Collection

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

Ethical Considerations

Our data collection is completely passive, and we only capture BLE advertisement packets (i.e., beacons) that devices already broadcast indiscriminately with the intention of being received by any nearby device. Many of these packets originated from pervasive BLE applications like contact tracing and device discovery. To ensure we only capture BLE advertisement packets, we configured our SDR to only capture BLE advertisement frequencies and mask off non-advertisement channels [63]. Furthermore, we ensure that in the decoding stage only undirected advertising packets are passed on to the analysis phase.

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

Data Analysis

We fingerprint and identify devices using our BLE tracking toolkit described in Section 4.2. We first determined how many packets from one device are needed in order to obtain the fingerprint and then identification. To do this, we performed controlled experiments using off-the-shelf ESP32 devices at different SNR values and different number of packets observed. The observation was that to establish a fingerprint the attacker need only 50 packets from the target BLE device, and this is sufficient even at low SNR values. We also observed that once the fingerprint is established, a device can be identified or tracked by observing only 10 packets. Considering that most mobile devices transmit several hundreds of BLE beacon packets per minute, an attacker doesn't need a lot of time to perform high-precision identification.

4.4.1 False Positives and False Negatives

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

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

We consider every device (MAC address) $i \in \{1, 2, 3, ..., 162\}$ as a target, and we train our classifier to find that device's fingerprint (Section **??**). Then, for each of the other devices,

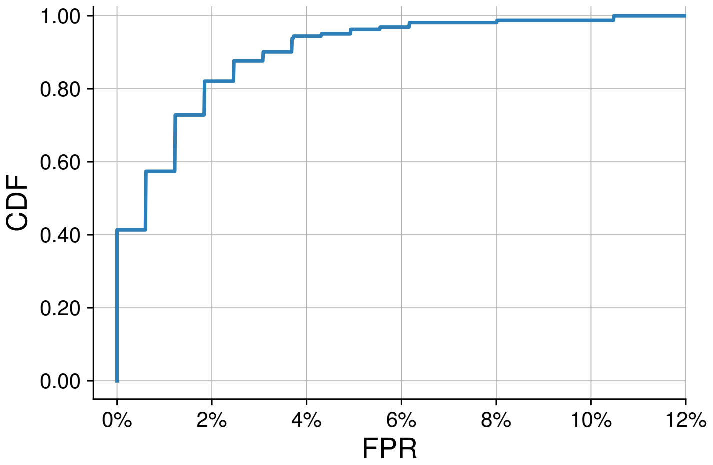


Figure 4.6. Dist. of FPR a device when comparing with all others

we run the classifier to see if it identifies them as the target (i) device. If it does, then that is considered a *false positive*. The number of false positives for target device i divided by the total number of devices is the False Positive Rate (FPR) for device i. Next, we fingerprint each target i and run the classifier to see if it fails to identify each device as itself. Each instance of this is a *false negative*. We repeat this process for all the 162 devices (each time one of them is selected as the target), and divide the result by the total number of devices to compute the total False Negative Rate (FNR). We observe our classifier achieves a 2.5% FNR across all 162 devices.

Figure 4.6 shows the distribution of FPR for each of the 162 devices. The median FPR of a device is only 0.62%. Moreover, 40% of the devices were not confused with any other device (zero FPR), which implies many devices seen in the field have unique physical-layer fingerprints. Owning a device with unique imperfections makes someone particularly vulnerable to BLE tracking attacks. We also observed a small fraction of devices had an FPR as high as 10%.

Effect of temperature

The temperature of the devices we observe in the field were unlikely to experience significant temperature changes during the course of our data collection. Therefore, we perform a model-based simulation to evaluate the effect of temperature changes on FPR and FNR. Recall that temperature changes affect CFO because of the well-documented relationship between frequency drift of crystal oscillators and their temperature (Section 4.3.1). Using the curves in [36], we calculate the change in CFO (Δf) as temperature drifts further from the temperature baseline when the device was fingerprinted (ΔT °C). To ensure the target is not missed even if the temperature changes are as large as ΔT °C, we modified the classifier to accept the device as the target even if the CFO of the device is Δf away from the fingerprinted CFO of the target. The consequence of increasing the range of acceptable CFO values is that it increases the chance of observing a device whose CFO falls in the acceptable range, resulting in an increase in FPR.

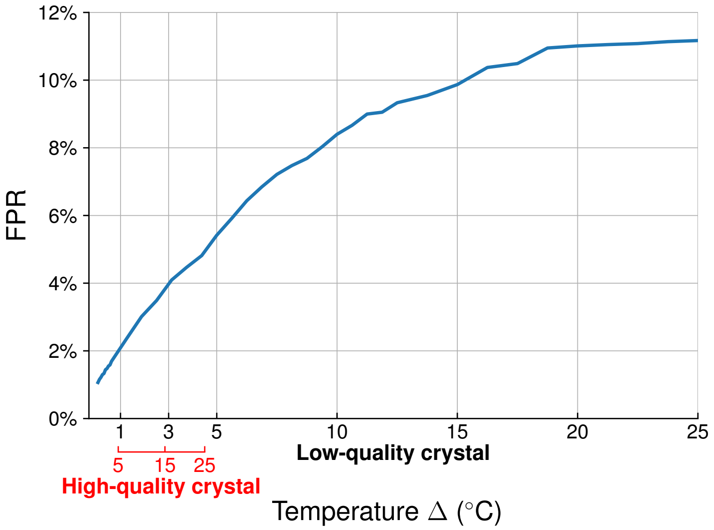


Figure 4.7. How oscillator temperature changes affect FPR.

Figure 4.7 presents the FPR as the change in temperature increases. We present the results for both high-quality and low-quality crystals (i.e., different cutting accuracies), as the type of crystal depends on the specific device being targeted. Temperature change causes significantly less change in CFO (and thus less increase in FPR) for high-quality crystals (0 minute cutting accuracy) compared to low quality crystals (8 minute cutting accuracy). For low-quality crystals, FPR increases rapidly as the temperature increases. If the change in temperature is too significant (25°C), CFO becomes useless for identification: the FPR is the same as if we only used IQ offset and IQ imbalance. In summary, temperature changes can severely limit an attacker's ability to track a target device.

4.4.2 Uniqueness of imperfections

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

To answer this question, we performed a larger-scale field data collection. We placed an SDR at the exit of a room where *hundreds of different devices* passed by each day. We recorded the Apple/Google COVID–19 Exposure Notification BLE beacons transmitted by those devices over the course of l0 hours on two days, separated by one week to limit the number of duplicate devices. We computed the mean CFO and mean I/Q offset magnitude for each BLE MAC address we observed in the beacons. The mean hardware imperfections are representative of the fingerprint of the BLE device. To reduce the chance that we observed the same device with two or more different MAC addresses, we filtered out devices which were observed for a duration longer than three minutes[2].

We observed 647 unique MAC addresses across the two 20 hours of data collection. Figure 4.8 shows the 2-Dimensional histogram of the fingerprints of these devices, namely their CFO and I/Q offset magnitude. The number of histogram bins were chosen so that the number of bins (2500) is significantly larger than the total BLE devices observed. Each bin represents a CFO range of ∼1.3 kHz, and an I/Q offset magnitude range of 0.00516. Devices that fall in the same bin are considered to have indistinguishable hardware imperfections. We also show the bounds of the 2D histogram that cover 36% (∼σ) and 67% (∼2σ) of the devices (σ because imperfections tend to be normally distributed).

We found that 47.1% (305) of the devices were unique. This confirms that even in a larger data set, ∼40% of devices are uniquely distinguishable. We also observed that devices

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

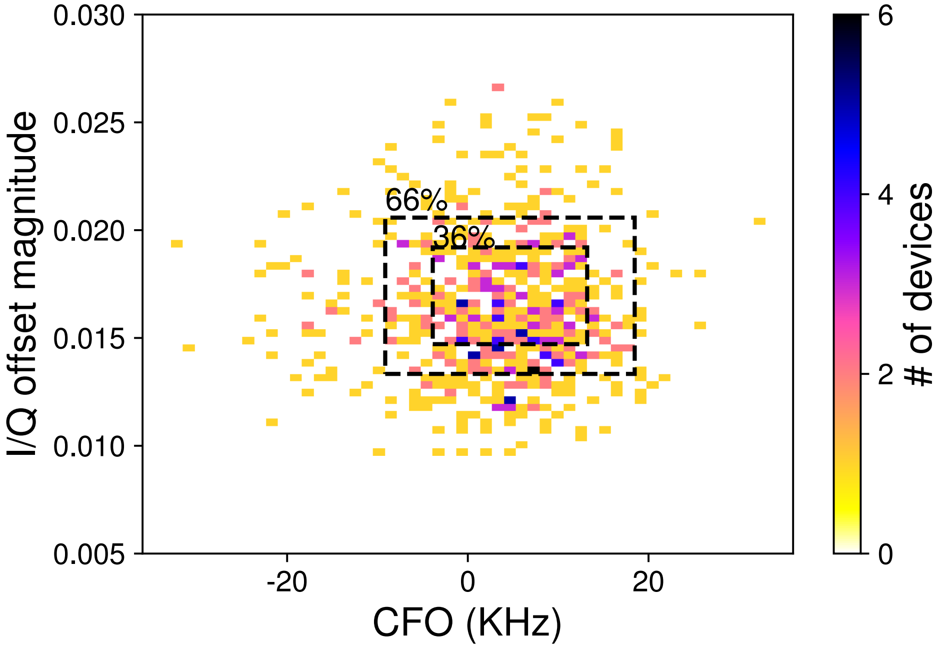


Figure 4.8. Histogram of imperfections across 647 BLE devices.

with overlaps did not overlap with many other devices. For instance, 15% (97) of the devices had similar imperfections with only one other device.

4.5 Conclusion

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

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

Chapter 4, in part, is a reprint of the material as it appears in *IEEE Symposium on Security and Privacy 2022*. Hadi Givehchian, Nishant Bhaskar, Eliana Rodriguez Herrera, Hector Rodrigo Lopez Soto, Christian Dameff, Dinesh Bharadia, Aaron Schulman. The dissertation author was the co-primary investigator and author of this paper.

Chapter 5

Reliable Scanning of Wireless Links Using Low-cost Commodity Radios

5.1 Introduction

In Chapters 3 and 4, I demonstrated that information from wireless scans, at the link and physical layers, can be used to perform targeted auditing of wireless access links, even in the presence of several other such links at public places. However, this targeted auditing requires that we actually have the tools to reliably obtain wireless scan data for every single wireless link in range. This is a challenge because auditing entire metropolitan areas requires wardriving data collection, but that means scanners will be in range for only a few seconds. In this chapter, I explore whether we can design tools that make it feasible to scan reliably for all wireless links, using just low-cost commodity hardware. In particular, the designed tool perform efficient scanning of classic Bluetooth wireless access links.

Classic Bluetooth access links are distributed throughout large urban areas. Performing a comprehensive security audit of this "network" of non-Internet connected links requires us to perform wireless scanning over large geographic areas. Unfortunately conventional classic Bluetooth scanning is a slow process[89]. Bluetooth scanning requires us to sequentially send device discovery (inquiry) packets on certain frequency channels and then wait for responses in a time slotted ALOHA manner [90]. This process has to be done for 32 inquiry request/response channels spread across a bandwidth of 76 MHz in the 2.4 GHz ISM band. Additionally, Bluetooth devices turn on their receivers for only short durations periodically (10 ms every 1.24 sec) to save power. This requires that the scanning must be sequentially repeated several times across all the 32 channels, to ensure that every device sends a response. Consequently, it takes at least 10.24 seconds to scan for every classic Bluetooth device within wireless range. In noisy environments, this duration can get even longer (upto 40.96 seconds).

The slow speed of single-channel Bluetooth scanning makes existing scanning tools like smartphones infeasible for reliably auditing all wireless links when wardriving. For example, at a driving speed of 50 mph, scanning for Bluetooth devices within a typical urban range of 100 meters requires us to finish the scan in less than 4 seconds. Since current single-channel scanners need $\sim$ 10 seconds to finish a scan, they will likely end up missing several devices in the scan data. Consequently, we are left with an impossible choice – we cover an entire metro area while continuously wardriving but miss Bluetooth devices, or we discover all Bluetooth devices in a smaller geographic area by doing stop-and-go scanning.

Modern software-defined radios can be used to design faster Bluetooth scanning tools, but they are limited by their cost and portability. Indeed, a portable low-cost scanning tool lets us deploy hundreds of them to cover an entire urban area. SDRs speed up the scan process by transmitting inquiry requests and receiving responses in parallel across multiple narrowband classic Bluetooth channels. The received raw response signals can be backhauled to the host computer, where they can be decoded to retrieve the information about the devices being queried. There also exist several low-cost portable SDR options such as PlutoSDR that provide an on-board processor and don't need a separate host computer.

Unfortunately, the bandwidth requirements of a parallel multi-channel Bluetooth scanning limit the choice of SDR hardware. Classic Bluetooth scanning requires us to send multiple inquiry requests and receive inquiry responses across channels spread over an analog bandwidth of 76 MHz in the 2.4 GHz band. This further requires a network backhaul rate of $\sim$2.5 Gbps, which can only be supported using a 10G ethernet link. This makes low-cost SDR options such as PlutoSDR unsuitable for a parallel multi-channel Bluetooth scanning application Higher end

SDRs can be used for this purpose, but they are often neither portable nor low-cost.

In this chapter, we present an initial exploration into the design of a low-cost multi-channel Bluetooth scanning tool, aimed at performing fast wireless audits across urban areas. The key insight that helps us in the design of this tool is that classic Bluetooth uses separate request and response channels. There are 32 inquiry request channels, and corresponding one-to-one mapped (but different) 32 response channels. We can therefore send inquiry packets rapidly across all channels, and responses will arrive at defined independent response frequencies after a Bluetooth protocol defined time period (time slot). The specific contributions of this work are:

1. We present a Bluetooth-protocol compliant multi-channel scan algorithm to reduce the scan time to under 10 seconds.
2. We present a novel technique for enabling low analog bandwidth SDRs (61.44 MHz) to access Bluetooth channels outside of their receive bandwidth

The rest of the chapter is organized as follows: Section 5.2 provides a background on the Bluetooth device discovery process, and prior work in speeding up Bluetooth scans. Section 5.3 presents the design of the multi-channel scan algorithm, resolving hardware issues with multi-channel scan, and enabling out-of-band channel access for low-cost PlutoSDR. Section 5.4 provides initial testing results on the speed-up provided by the new scanning algorithm, and we conclude in Section 5.5.

5.2 Background

5.2.1 Classic Bluetooth device discovery

This section describes the Bluetooth scanning approach as described in the Bluetooth specification. This is the process used by current protocol-compliant scanning tools such as smartphone apps. Classic Bluetooth device discovery follows a time-slotted ALOHA approach. The scanner sends inquiry request packets on certain frequency channels and then listens for

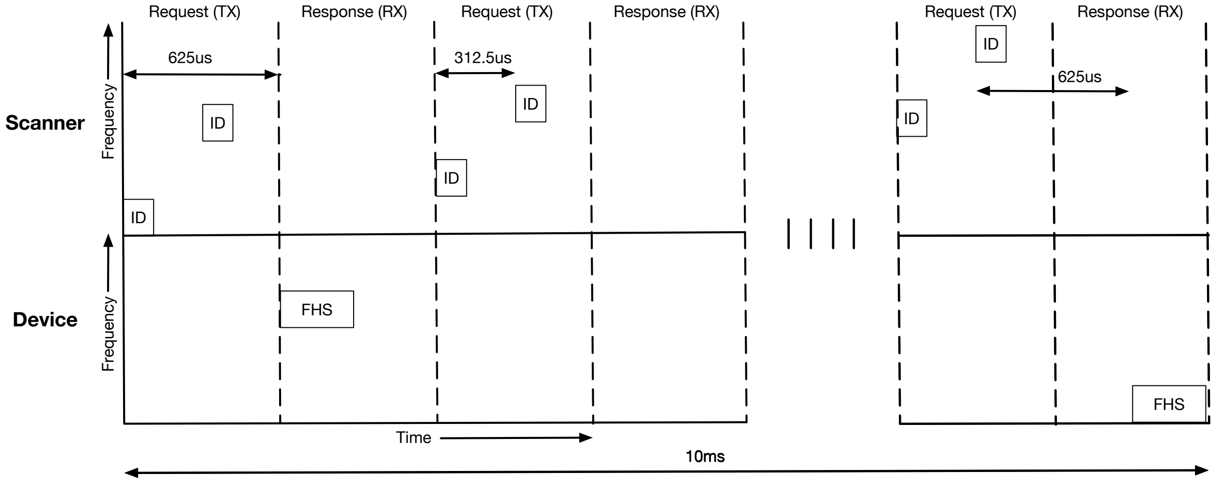


Figure 5.1. Classic Bluetooth device discovery process

responses in alternating 625μs time slots. Every request time slot, the scanner sends two inquiry (ID) requests on two different channels, switching to a different pair of channels in the next request time slot. Devices hear for ID packets and then respond with inquiry response (FHS) packets that contain information that we intend to collect for device auditing (MAC address, device type and other fields). Devices respond back with the FHS packets exactly 625 μs after they hear the ID packet, on the corresponding response channels.

There are 64 non-overlapping 1 MHz channels used for discovery – 32 request channels, and corresponding 32 response channels with a one-to-one mapping. This ensures the scanner and device both know exactly which channels to receive/transmit ID/FHS packets on. Figure 5.2 shows the mapping of request and response channels. The scanner sequentially cycles through 16 of the 32 channels, sending ID packets on two channels every request time slot. This full set of 16 channels comprises one inquiry train.

Bluetooth scanning is slow

The Bluetooth inquiry process was designed for low-power devices, and for avoiding interference in the 2.4 GHz band. A Bluetooth device doesn't listen constantly for ID packets; a typical device will only listen for ID packets on one frequency channel for 11.25 ms in a 1.28 second interval. At the end of 1.28s interval, the device switches the frequency channel in order

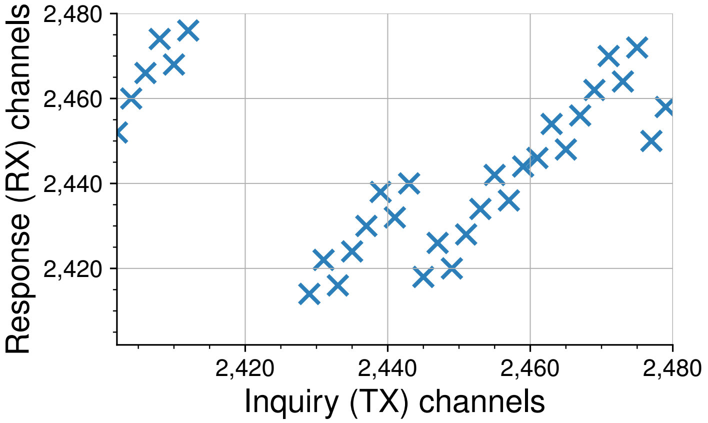


Figure 5.2. Mapping of inquiry request and response frequency channels for classic Bluetooth

to minimize impact of potential interference. Consequently, scanners need to repeat the inquiry train several times to ensure that when a device actually wakes up it receives an ID packet. In particular, scanners repeat an inquiry train of 16 channels 256 times (2.56s in total), and then move onto the other train of 16 channels. In order to minimize collisions, the trains also swap one frequency member every 1.28s. Consequently, the specification mentions that the inquiry process must run for at least 10.24s in a noise-free environment, and at least 40.96s in a noisy environment to guarantee we receive FHS packet from every device. This scan speed is very slow, resulting in likely missed devices when driving around with a mobile scanner.

5.3 Designing a multi-channel BT scanner

5.3.1 A multi-channel Bluetooth scanning protocol

As discussed in section 5.2, conventional Bluetooth scanning using narrowband scanner (e.g. smartphone) can take from 10.96s – 40.96s, which is very slow for wardriving applications. This slow speed is a consequence of the sequential nature of transmitting ID packets, and the fact that the Bluetooth devices are sleeping for majority of their time.

We define a new fast multi-channel scanning protocol, that can be implemented on wideband SDRs. The following key insights help us define this faster scanning protocol:

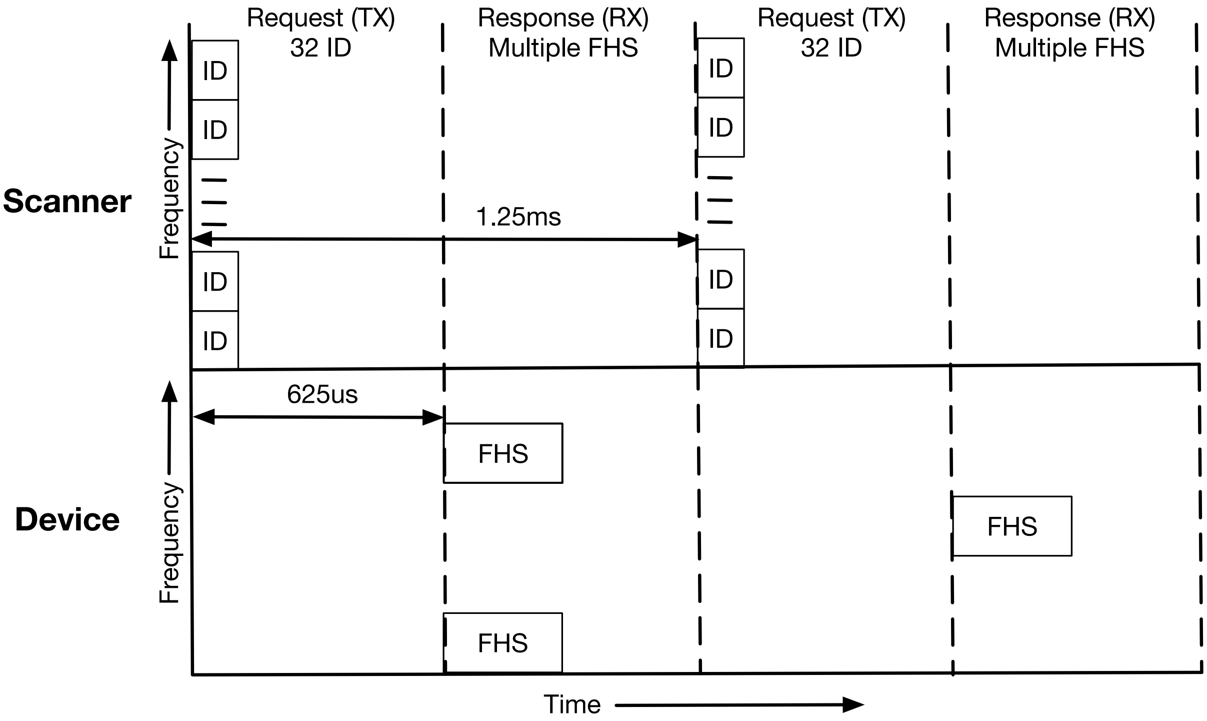


Figure 5.3. Multi-channel scan process and timing

1. If a Bluetooth device is awake and hears an ID packet, it will reliably respond with a FHS packet exactly 625μs later.

2. A Bluetooth device will listen for ID packets for 11.25ms in a 1.28s duration.

Instead of a train of ID packets spread over 10ms, we transmit ID packets on all channels in a burst at the start of every request time slot, using the TX chain of the SDR. Figure 5.3 shows the packet transmissions and timing for this new scanning process. We ensure tight timing control to ensure that the next burst is transmitted exactly two slots later (1.25ms later). Every Bluetooth device that is listening, will respond back with a FHS packet at the start of the response time slot. These packet signals can be received by the wideband RX chain of our SDR, and decoded to obtain the relevant auditing information.

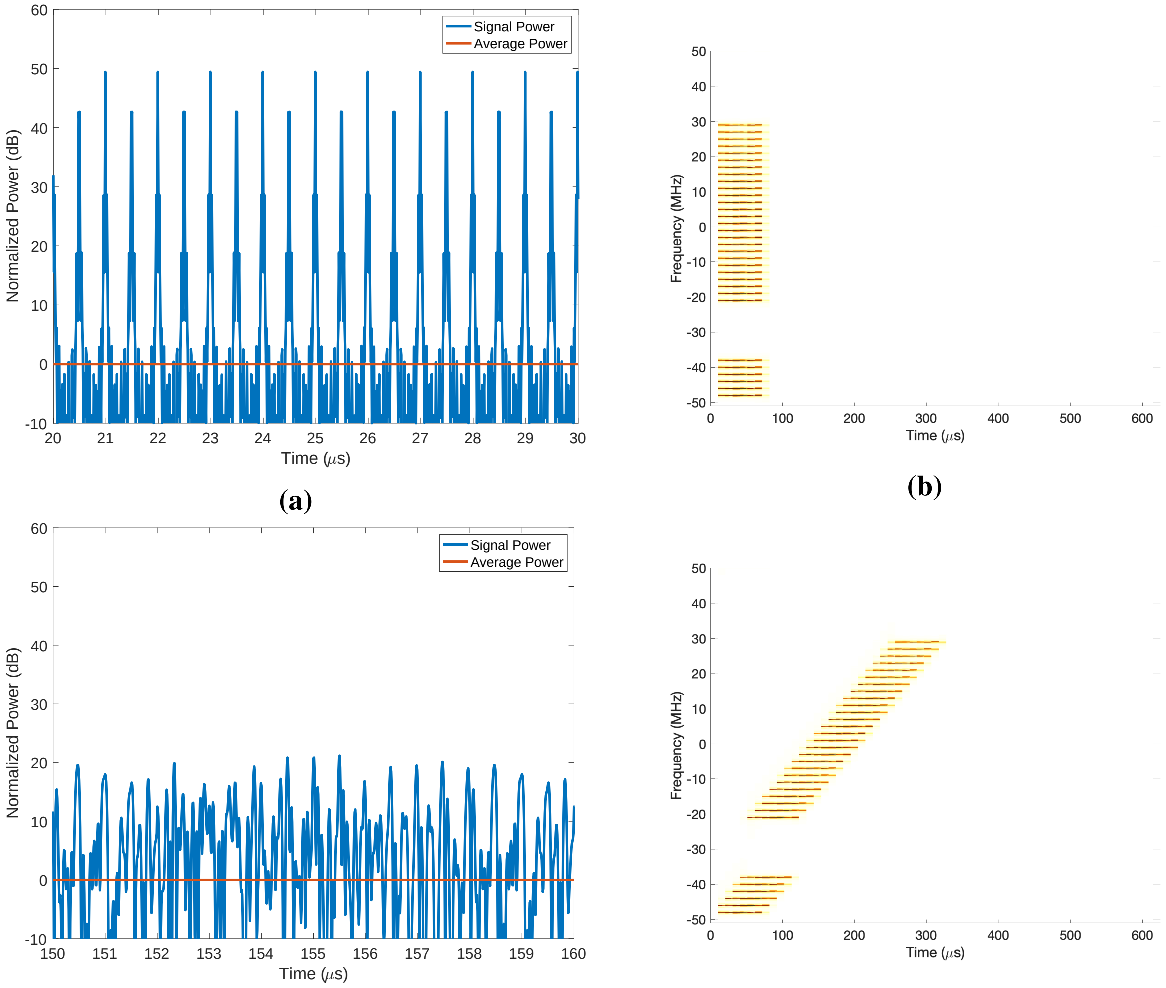
We need to repeat this burst a certain number of times to ensure every sleeping device wakes up and responds. Since typical Bluetooth devices wake up once in a 1.28s interval (for 11.25ms interval), repeating the bursts for a duration of 1.28s should ensure we are able to get responses from every surrounding device. In a noisy environment, we may need to repeat these bursts for double the time, or 2.56s. This is significant theoretical improvement over the scanning speed possible with conventional scanning tools.

Handling PAPR issues

While our ID packet burst strategy improves the scanning speed, we cannot use this packet burst directly on SDR hardware. In particular, ID packets have the exact same packet contents, resulting in the same waveform being transmitted on all frequency channels at the exact same time instant. This results in the overall wideband signal having pulses in the time domain, which is problematic for the analog hardware of our SDR. These pulses result in a very high Peak-to-Average-Power-Ratio (PAPR), and these get significantly distorted due to non-linearities of the programmable amplifier (PA). In addition, high PAPR signals are difficult to represent using the limited resolution of the DACs. This PAPR problem is commonly encountered with radios transmitting OFDM signals, which is very similar to the multi-channel ID packets burst for our scenario.

To reduce the PAPR of the overall wideband signal, we stagger the ID packets on consecutive frequency channels by a small time duration. This breaks the frequency repetition, and prevents formation of pulses in time domain. We offset each channel's packet in time by a $N * t_{stagger} \mu$s, where N is the frequency channel index. Figure 5.4 qualitatively shows how our staggering the ID packets reduces the peak power of the signal, making it more amenable to transmission without distortion.

The value of $t_{stagger}$ is chosen with a few considerations. Firstly, most radios are designed to handle PAPR of 15dB. We choose a stagger time to bring the PAPR of the signal below 15 dB and ensure that the quantization noise is minimized. Secondly, the value of time offset should be such that we can maintain the tight request and response slot boundaries. FHS responses are received exactly 625μs after ID packets and a typical FHS packet is 366μs in length. In order to ensure that we can receive this FHS response within the 625μs response slot boundaries,



(a) **(b)** **(c)** **(d)**

Figure 5.4. Reducing PAPR by staggering ID packets. Subfigures (a) and (c) show the time domain wideband signal with and without the stagger, and (b) and (d) are the spectral plots showing the time shift of consecutive ID packets. The peak power is significantly reduced with the stagger.

our stagger time must be so that the last ID packet is transmitted $\leq 259\mu$s. This can be done for a $t_{stagger} \leq 8.35\mu$s. In Figure 5.5, we show the relationship between $t_{stagger}$ and the PAPR. $t_{stagger} = 8.2\mu$s is chosen for our system, as it gives the lowest PAPR within the constraints mentioned above.

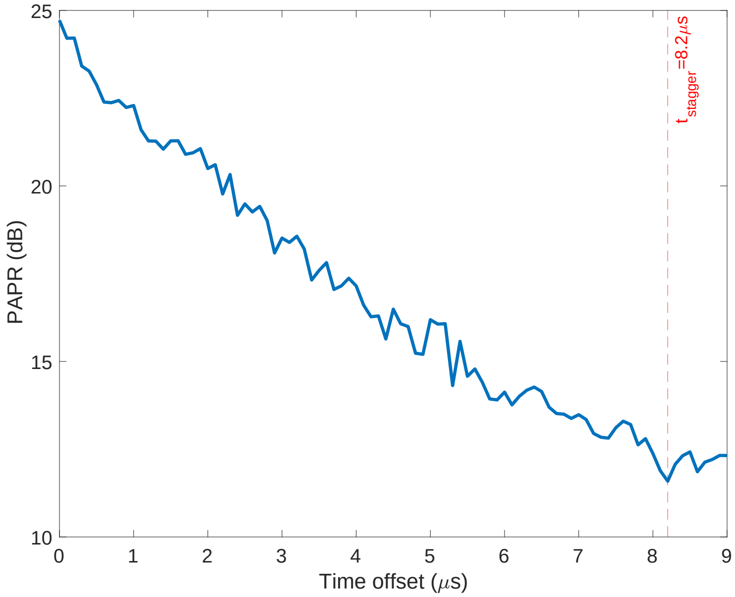


Figure 5.5. Variation of PAPR with the stagger time offset of ID packet burst. Dotted line shows the chosen stagger value for our system

5.3.2 Multi-channel scanning on low-end commodity SDR

Implementation of the multi-channel scanning protocol requires a hardware platform that supports a wide analog bandwidth to cover the entire 76 MHz band, on both transmit and receive sides. It also should have processing capability to process raw FHS signals at any response frequency, decode and extract information from the FHS packets. There are several commodity SDR platforms, e.g. USRP x300, that support a wide analog bandwidth (upto 160 MHz), and a large network bandwidth (10Gbps ethernet link) to offload processing on a separate computer. But such platforms costs several thousand dollars and need a separate computer for processing and decoding, making them not portable or scalable and therefore unsuitable for large scale empirical wireless auditing.

Instead, we chose to implement our system on the Analog Devices ADALM-PLUTO (PlutoSDR) software radio. The PlutoSDR is a popular example of a new class of modern SDRs – low-cost, standalone units with on-board processing capability. It costs only $230 and includes analog front-end(filter, amplifiers), ADCs, DACs, FPGA as well as an ARM processor (667

MHz dual-core) and 512 MB memory. The processor on the Pluto runs Linux and can support on-board custom signal processing and packet decoding. The PlutoSDR is an ideal platform in terms of cost and portability to implement our wardriving scanning tool.

However, there are certain hardware challenges that we have to overcome for using the PlutoSDR for our multi-channel Bluetooth scanning tool. Firstly, the Pluto has a limited analog bandwidth of 56 MHz on the receive side and 40 MHz on the transmit side. Since the Bluetooth request and response channels are spread across a 76 MHz band, the PlutoSDR will be unable to send ID packets and receives FHS packets across the entire Bluetooth band. Second, even for the response channels it can sense, PlutoSDR's sampling rate of 61.44 million samples per second results in 245 MB of sample data per second on the receive side. The built-in processor is not powerful enough to filter and decode in real-time this large amount of data. In the following subsections, we provide some initial ideas on overcoming these hardware design challenges.

Extending the analog bandwidth

Classic Bluetooth has request channels spread across 76 MHz (2.402 GHz – 2.477 GHz), and response channels spread across 63 MHz (2.414 GHz – 2.476 GHz). However, PlutoSDR only supports an analog bandwidth of 56 MHz on transmit side, and 40 MHz on receive side, resulting in some Bluetooth channels being out-of-band. As a consequence, the PlutoSDR won't be able to query and listen on all channels which will result in an unwanted increase in scan time.

To extend the analog bandwidth of the Pluto, a key insight comes from how request/response channels are distributed in the 2.4GHz band. Request and response channels are interspersed across the band, and the time slotted ALOHA nature of Bluetooth scanning means that request channels are free during response time slots and vice-versa. Therefore, during response slot if we can create an image of the out-of-band channels onto the free in-band channels, we can easily receive those responses. Similarly, in inquiry slot we can transmit on the free in-band channels and create images onto the necessary out-of-band inquiry channels. Figure 5.7

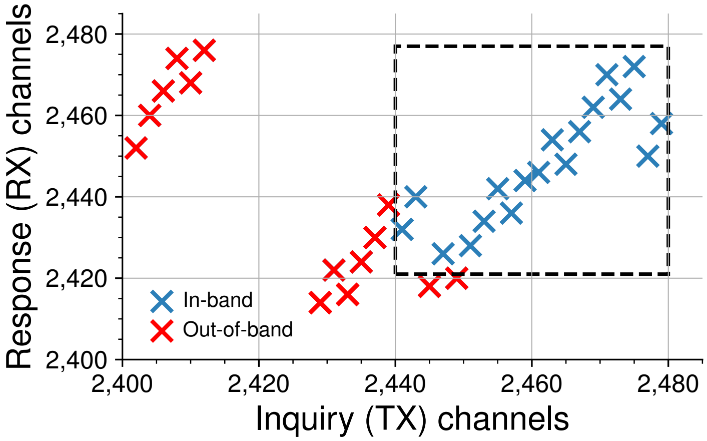


Figure 5.6. Map of request/response channels showing out-of-band channels for PlutoSDR. The dotted lines represent the analog bandwidth limits of the Pluto.

demonstrates this idea.

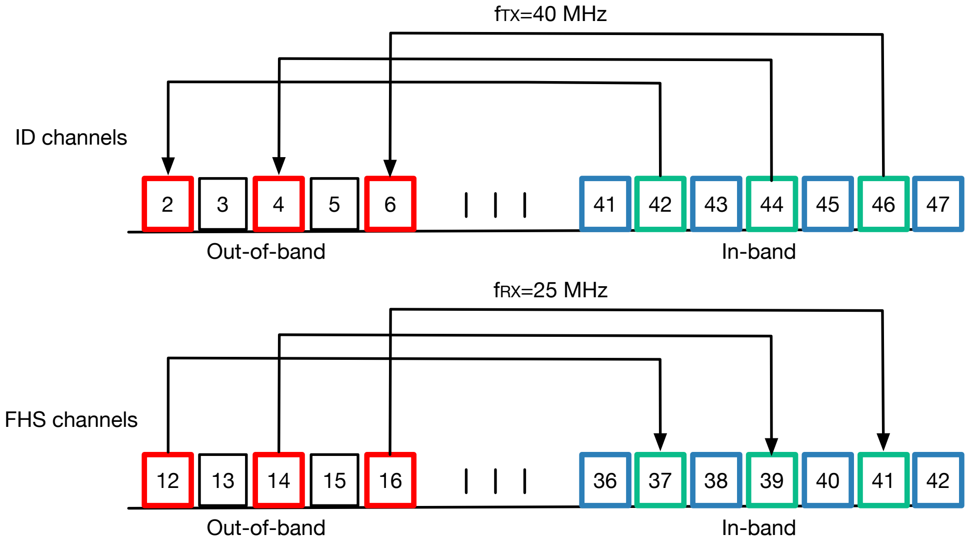
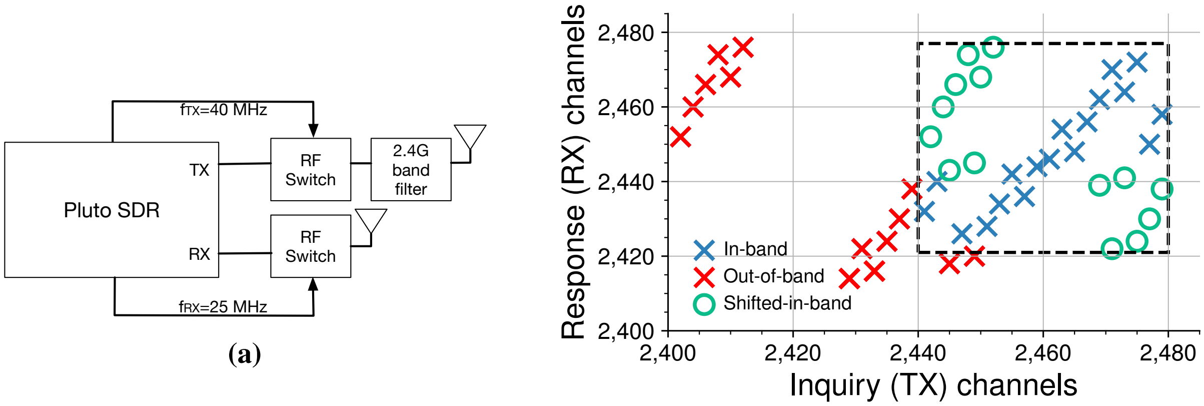


Figure 5.7. Utilizing unused request and response channels for moving out-of-band channels in-band and vice versa

To create these images, we can use an analog frequency mixer. However, that would require us to generate the appropriate frequency shifts using an additional voltage controlled oscillator, increasing the system cost and complexity. Instead, for our design we utilized an RF switch as a mixer [115, 116] and generate the necessary TX and RX shifting frequencies (switch control input) directly from the PlutoSDR's FPGA. We use a shift frequency $f_{TX} = 40$MHz for

ID packets, and $f_{RX} = 25$MHz for FHS packets. On the transmit side, the use of a square wave as a switch input will generate harmonics even outside the 2.4GHz. We use a 2.4 GHz band filter to remove these unwanted images. Figure 5.8 shows our system block diagram and the remapped inquiry and response channels in our final system hardware.



(a)

(b)

Figure 5.8. Extending the analog bandwidth of PlutoSDR. (a) shows the system block diagram and (b) shows the new map of request/response channels with shifted images

Minimizing data backhaul using SparSDR

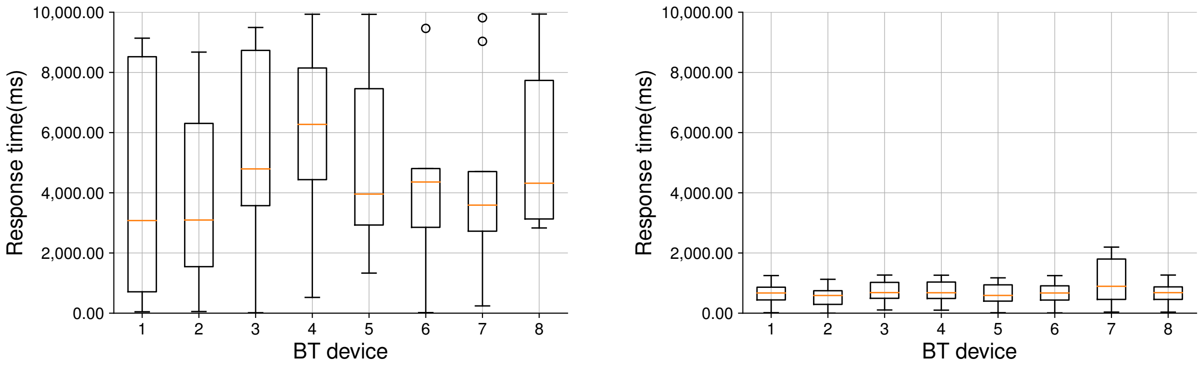
The PlutoSDR needs to listen for responses on channels spread across a wideband (56 MHz). Typically, this is done by running the analog to digital converters at a very high sampling rate (61.44 Msps), receive the raw signal samples, and then decode them per channel to obtain the Bluetooth device information (MAC address, device type). Unfortunately, this high sampling rate will result in almost 245 MB of raw sample data that needs to be processed. The PlutoSDR processor is not powerful enough to this in real-time, nor does it have sufficient on-board memory to store this temporarily. Worse, the majority of this compute is unnecessary because (1) the response channels only occupy 32 MHz of the total receive bandwidth, and (2) FHS packets are not received all the time, most of the time we only get useless signals or noise.

To solve this problem, we utilized our previous project SparSDR [63]. SparSDR provides the ability to compress the spectrum in both frequency and time. It lets us channelize the spectrum by masking out the frequency channels which are unused, allowing us to only receive signals in the 32 MHz of actual response channels. Additionally, it lets us threshold the signal level in the individual channels, ensuring that we only need to spend time processing signals that are above a certain power level, and not waste resources processing noise samples. Coupled with the fact that actual spectrum occupancy is very low even in noisy environments, SparSDR helps us drastically reduce the amount of data the processor needs to handle. The processor now only needs to process when there are valid signals above a certain power on the response channels.

5.4 Evaluation

We perform an initial experiment to evaluate the speedup in discovering classic Bluetooth devices of our new multi-channel scanning protocol and bandwidth extension, over a conventional smartphone scanner. For our target we chose the HC-05, a popular commodity off-the-shelf Bluetooth module used in several applications in an urban area. In fact this module is so popular that we even criminals use it to build credit card skimmers We took 8 HC-05 devices and put them in an RF isolation box, isolating them from WiFi and other wireless signals. We connected a USRP x300 in the box to sense all the requests transmitted by our tools, and the responses sent by the HC-05 devices. The USRP constantly records the entire 80 MHz Bluetooth band to a computer, capturing every single packet that is sent by any device in the setup. We first put a smartphone with a Bluetooth scanning Android app (we used Bluetana from chapter) in the box to use as the scanning tool. We run scanning app in constant scanning mode (with default scan time settings) to discover all the eight HC-05 devices. We record all classic Bluetooth packets using the x300 for 41 seconds ($> 40.96s$), and then process and decode the recorded requests and responses. We then perform the same experiment using a PlutoSDR with our bandwidth extension hardware, and running our multi-channel scanning protocol using a simple onboard software.

Figure 5.9 shows the distribution of time to receive a response from each of the eight



(a) **(b)**

Figure 5.9. Empirical observations of Bluetooth device discovery time for (a) smartphone-based scanning tool and (b) our improved low-cost scanning tool. Our tool significantly improves scan time, making it more effective for wardriving.

devices, across 10 tries of the experiment for both the smartphone scanner and our low-cost scanning tool. We observe that the average time of discovery across the 8 devices improves from 4.88s to 0.71s for our low-cost scanner. The variation in scan time from one trie to the next is significantly lower for our fast scanning tool. Furthermore, we observe that the worst case scan time for any Bluetooth device goes down from 9.93s to 2.03s, making this tool extremely reliable and effective for wardriving data collection.

5.5 Conclusion

In this chapter, we presented the design of a low-cost multi-channel fast Bluetooth scanning tool. We developed a multi-channel scanning protocol and resolved hardware challenges of PAPR and analog bandwidth limitations of PlutoSDR. These measures allowed us to perform this wideband multi-channel scanning on a low-cost commodity tool such as the PlutoSDR. The new scanner significantly speeds up classic Bluetooth device discovery over conventional single-channel smartphone scanners. By finding Bluetooth devices faster and more reliably, our tool significantly reduces the chances of missing devices in Bluetooth scans during wardriving. Our scanner therefore enables truly comprehensive urban scale audits of our distributed network of "wireless ad-hoc" links.

Chapter 5, in part, is currently being prepared for submission for publication of material. Nishant Bhaskar, Raghav Subbaraman, Sam Crow, Moein Khazraee, Dinesh Bharadia, Aaron Schulman. The dissertation author was the primary investigator and author of this material.

Chapter 6

Conclusion

Wireless access links are an important component of various electronic computing systems in urban areas. Securing these critical wireless links against attackers is crucial to the safe operation of our computing systems. In this dissertation, I explored real-world security and privacy problems associated with these wireless access links, by utilizing information collected from wireless scans. In particular, through targeted auditing of wireless access links, I investigated if attackers are hiding illicit wireless links in public computing infrastructure, and if attackers are carrying out targeted attacks on popular personal wireless links. During the course of multiple large-scale field measurement studies across urban areas, I demonstrated that indeed we can reliably perform targeted auditing of wireless access links in public places even with limited information in wireless scans, and that we can reliably scan for all observable wireless devices while wardriving by using low-cost commodity hardware. I defended the following thesis: *To defend wireless access links spread across urban areas, it is feasible to: 1) use link layer scan information to identify illicit wireless links, 2) use physical layer information in wireless signals to attack a target wireless device, and 3) scan reliably for all wireless access links when wardriving using low-cost commodity hardware*

By utilizing link layer information from wireless scans, I demonstrated that defenders can detect illicit wireless links in public infrastructure. In Chapter 3, I performed a field measurement study over multiple states in the US to investigate the problem of illegal Bluetooth-based payment card skimmers at gas stations. We built a Bluetooth scanning app (Bluetana), that was used by investigators to collect scan data across 1185 gas stations, and detect and recover 64 skimmers over a 19-month period. I observed that link layer characteristics of Bluetooth devices in skimmers can distinguish them from other benign devices at gas stations. In particular, predominately criminals utilize popular commodity modules with default Bluetooth properties. Interestingly, I observed that criminals changing link-layer properties to hide the skimmers had a counter effect, as the skimmers became more conspicuous. Finally, there are benign computing systems in and around gas stations that utilize similar Bluetooth modules as skimmers, that may cause confusion in detection. Overall though, Bluetooth scanning based detection is an effective means of defending against the threat of these illicit wireless links.

By utilizing physical layer information from wireless signals, I demonstrated that attackers can perform targeted tracking of wireless personal devices. In Chapter 4, I performed multi-day field measurement studies across several real-world public locations, to investigate the possibility of physical-layer tracking attack on BLE-enabled wireless access links. Specifically, I examined the scenario in which an attacker uses the unique hardware imperfections of the BLE radio obtained through Bluetooth scanning, to perform targeted identification of a victim's personal device (e.g. smartphone). I observed that a number of BLE access links have very distinguishable hardware fingerprints, that makes it feasible to track them even in the presence of hundreds of other devices in public locations. As an example, I observed that 47.1% of the 657 BLE-enabled smartphones that were seen at a public facility were distinguishable from the other devices. However, I also observed that several real-world factors that limit an attacker's effectiveness of such physical-layer tracking. For instance, certain devices may have common fingerprints and can be easily confused; device and ambient temperature may impact the measurement of these hardware fingerprints. Overall, physical layer information derived from Bluetooth scans can be used by an attacker to target a particular wireless access link, but their effectiveness is limited by several real-world constraints.

By performing multichannel scanning on commodity radio receiver, I demonstrate

that it is feasible to reliably scan all wireless access links even when wardriving. Finally, in Chapter 5, I presented a new Bluetooth scanner design, that enables faster enumeration of classic Bluetooth access links compared to a smartphone-based scanner. Specifically, we implement a scanning tool on a low-cost SDR platform that reduces theoretical maximum Bluetooth scan time, ensuring reliable auditing of Bluetooth access links. To do this, I designed a multichannel Bluetooth scan protocol that sends/receives Bluetooth inquiry requests in parallel on all channels. I also resolved hardware limitations of low-cost PlutoSDR to perform this parallel request/response, namely reducing the PAPR by staggering transmitted inquiry packets, and using unused response channels with an RF switch to transmit inquiry request packets on channels out-of-band for PlutoSDR. I observed that compared to a smartphone scanning app, our implementation reduced the average scan time for real-world commodity Bluetooth devices by a factor of 7x, allowing reliable scanning of access links even if in range for short time. Overall, we can reliably scan for all wireless access links even while wardriving, by performing multichannel scanning using a low-cost commodity radio receiver.

In summary, in the course of all this work, I have demonstrated that wireless scanning based auditing is a feasible and reliable mechanism for securing diverse wireless access links spread across urban areas.

6.1 Future Directions

The work in this dissertation is an initial foray into the vast but relatively unexplored attack surface of wireless access links in urban areas. These wireless links are used in critical computing systems unknown to us, and are potential targets or even under attack unbeknownst to us. Through my work I have shown that wireless scanning based auditing is a practical and insightful mechanism to reason about the security and privacy of these links. Furthermore, I have shown that tools for performing effective wardriving-based auditing, can be made low-cost and therefore accessible to the community at large. Beyond the tools and techniques themselves, the field measurement campaigns I undertook provided immense insights into attacker behavior and their limitations and brought to light real-world threats that plague urban wireless access links.

This dissertation has laid the groundwork for several possible directions for future work. Credit card skimmers and wireless personal devices are just a couple of examples of wireless access links. In reality, wireless access links are utilized in many types of public infrastructure all throughout urban areas. It is necessary for public utility and safety that this infrastructure be operated safely. And yet despite their criticality, these wireless links are often undocumented. This makes it difficult to reason about the potential security problems such as unauthorized access, or even illicit links. A potential direction for future work can involve performing a field measurement study similar to Chapter 3, albeit at a much bigger scale and targeting all types of wireless access links (Bluetooth, WiFi, Zigbee) A metropolitan scale wireless scanning-based auditing of all wireless access links in public infrastructure can help reveal potential weak links for unauthorized access and also reveal illicit wireless links that have implanted inside the equipment. This enumeration can be followed by security analysis of the potential weak wireless access links, and then designing proactive defenses to prevent such misuse of the wireless links.

Another research direction that can be greatly beneficial is auditing of non-discoverable devices. Indeed, during this work I targeted wireless access links that could be scanned for and enumerated. While this represents the vast majority of wireless access links in use, there are other devices which are set to be non-scannable or non-discoverable. This feature is a common security mechanism provided by all wireless protocols, to ensure that once a device is associated with a trusted device — e.g., Bluetooth access links is paired to a smartphone — it should be not be visible in scans to any other scanners. This ensures that only trusted devices attempt a connection to the device. While this feature has not been in use for licit wireless access links, because of the additional burden of always maintaining the one device that is associated with the non-scannable link, this feature has been used by criminals to hide their illicit wireless access links from being detected. We don't currently have reliable techniques (other than brute force) to detect such non-discoverable devices. However, this problem has a different dimension that we must take into account — the ethical dimension. Indeed, if we were to design a technique to bypass non-discoverability, we will be bypassing a key security feature used by a lot of legitimate devices. Therefore, we need to design techniques that help us perform auditing of these links, while at the same time maintain anonymity of legitimate wireless access links.

Finally, in Chapter 5 I demonstrated the design of a multichannel Bluetooth scanner that can perform fast and reliable scanning of all Bluetooth links around. Another research direction can be to extend this scanner to support multiple wireless protocols such as WiFi and Zigbee. Indeed, WiFi, Zigbee and Bluetooth all share the 2.4 GHz ISM band, and therefore wireless signals received at our SDR can capture packets from all these protocols. Furthermore, a multi-protocol scanner will be immensely useful for comprehensive auditing of all possible wireless links seen across urban areas. However, with the additional burden of backhauling and processing additional packets from multiple protocols, it is possible that we may overwhelm the PlutoSDR processor resulting in packet losses. Furthermore, it is tricky to perform timed ALOHA style transmit and receive for certain protocols, when we must also schedule other protocols. That said, the benefits of such a multi-protocol scanner make this an attractive research direction to investigate into.

Appendix A

Payment Card Skimmer Court Cases

The appendix contains excerpts from various public court documents related to cases of credit card skimming. These excerpts provide anecdotal data about the monetary impact of the skimmer problem.

A.1 Cashout Value

USA v. Hristov et al [80]

"...Bank of America suffered a loss of $33,000 with 36 compromised customer accounts. Citizens Bank suffered a loss of $91,580 with 74 compromised customer accounts ..."

USA v. Cristea et al [34]

"...Altogether, on February 21,2016, FBI surveillance observed Cristea, Co-conspirator #1, and Co-conspirator #2 go to approximately 12 different locations, where, according to CardTronic's records, they withdrew at least $7,000 from at least 18 First National Bank accounts ..."

USA v. Khasanov et al [7]

"...USPS agents thereafter conducted record checks on the purchased USPS money orders and discovered that 10 of the 57 money orders had been purchased with 5 payment numbers issued by Citibank ..."

Date	Location of USPS	Amount
Aug 4 2017	Waldorf, MD	$2,904.80
Aug 7 2017	Washington, DC	$1,492.80
Aug 7 2017	McLean, VA	$1,400.00
Aug 7 2017	Washington, DC	$1,803.20
Aug 7 2017	Hyattsville, MD	$792.05

USA v. Aqel [98]

"...the Probation Officer also notes that the actual loss to victims was $8,327.58. Id. Similarly, the Probation Officer notes that while Mr. Aqel possessed 120 stolen credit card numbers, only 23 of those numbers were used to make purchases ..."

USA v. Rodriguez et al [57]

"...Between on or about July 7, 2016, and on or about July 20, 2016, Defendant ... attempted to conduct approximately 133 retail transactions totaling in excess of $27,000 ... using approximately 90 counterfeit access devices re-encoded with credit/debit account information that were obtained by a skimming device placed on the point of sale terminal of a gas pump ..."

Application for Search Warrant, 2:18mj1277[12]

"...On April 14, 2016, a man (later identified as Estrada) used a fraudulent Visa credit card and a fraudulent MasterCard to purchase two $300.00 gift cards from the Kohl's store ..."

USA v. Konstantinov et al [8]

"...In total, the defendants compromised approximately 524 debit card accounts and made approximately 779 fraudulent withdrawals, totaling $348,376.80 ..."

A.2 Credit/Debit cards per skimmer per day

Application for Search Warrant, 2:18mj1277 [12]

"...On September 9, 2016, an employee at Jilly's Mobil ...reported to Detective Craig Meyer that he had found what appeared to be a skimmer on pump #8 ...Detective Meyer downloaded and exported the data stored on the skimmer taken from Jilly's Mobil pump #8. The results showed data for 221 victim credit card accounts ...

...Detective Meyer reviewed the video surveillance footage for Jilly's Mobil from September 1, 2016. At 1:38 PM on September 1st, a red Ford Explorer drove to pump 8. The Ford Explorer was positioned in a manner whereby the opened passenger door blocked the view of the gas pump by the store employee inside the Jilly's Mobil ..."

Bibliography

[1] Arizona Department of Agriculture, Weights and Measures Service Division . Data Skimmers in Motor Fuel Dispensers. https://agriculture.az.gov/sites/default/files/Skimmer%20Presentation%20%28Website%20Edition%29.pdf, September 2017.

[2] Nate Seidle . Gas Pump Skimmers . https://learn.sparkfun.com/tutorials/gas-pump-skimmers, September 2017.

[3] Nick Poole . Credit Card Skimmers Evolved: Shimming . https://www.sparkfun.com/sparkx/blog/2673, April 2018.

[4] Office of Minnesota Attorney General Keith Ellison . ATM and Gas Pump Skimmers . https://www.ag.state.mn.us/Brochures/pubATMSkimmers.pdf.

[5] Rippleshot . State of Card Fraud: 2018. https://www.aba.com/Products/Endorsed/Documents/Rippleshot-State-of-Card-Fraud.pdf, 2018.

[6] United States Sentencing Commission . Guidelines Manual . https://guidelines.ussc.gov/gl/%C2%A72B1.1, 2018.

[7] Affidavit in Support of Criminal Complaints and Arrest Warrants, USA v. Khasanov et al, 1:18cr149. US District Court for the Eastern District of Virginia. https://www.courtlistener.com/recap/gov.uscourts.vaed.385830/gov.uscourts.vaed.385830.2.0.pdf, January 2018.

[8] Appeal from the US District Court for the Eastern District of Oklahoma, USA v. Konstantinov et al, 6:13cr62. United States Court of Appeals for the Tenth Circuit. https://www.ca10.uscourts.gov/opinions/14/14-7050.pdf, June 2015.

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

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

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

[12] Application for Search Warrant, 2:18mj1277. US District Court for the Eastern District of Wisconsin. https://www.courtlistener.com/recap/gov.uscourts.wied.84529/gov.uscourts.wied.84529.1.0.pdf, July 2018.

[13] Chrisil Arackaparambil, Sergey Bratus, Anna Shubina, and David 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.

[14] Arizona Department of Agriculture. Credit Card Skimmers. https://agriculture.az.gov/weights-measures/fueling/credit-card-skimmers, February 2019.

[15] Kyle Arnold. Florida gas pump thefts rise as credit-card skimmers get more savvy. https://www.orlandosentinel.com/business/consumer/os-bz-credit-card-skimmers-20181108-story.html, November 2018.

[16] ArsTechnica. Stalkers' "chilling" use of AirTags spurs class-action suit against Apple. https://arstechnica.com/tech-policy/2022/12/apple-airtags-are-now-the-weapon-of-choice-for-stalkers-lawsuit-says/, December 2022.

[17] ATM Industry Association. Global Fraud and Security Survey - 2017. https://www.ncr.com/company/blogs/financial/how-much-does-atm-crime-cost, January 2018.

[18] Hagai Bar-El. White Paper: Known Attacks Against Smartcards. Technical report, Discretix Technologies Ltd., 2005.

[19] Marco V. Barbera, Alessandro Epasto, Alessandro Mei, Sokol Kosta, Vasile C. Perta, and Julinda Stefa. CRAWDAD dataset sapienza/probe-requests (v. 2013-09-10). Downloaded from https://crawdad.org/sapienza/probe-requests/20130910, September 2013.

[20] Mark Bassegio and Erik Evenchick. Breaking access controls with BLEKey. https://www.blackhat.com/docs/us-15/materials/us-15-Evenchick-Breaking-Access-Controls-With-BLEKey-wp.pdf, August 2015.

[21] Kevin Bauer, Damon McCoy, Ben Greenstein, Dirk Grunwald, and Douglas Sicker. Physical layer attacks on unlinkability in wireless LANs. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 108–127. Springer, 2009.

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

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

[24] Nishant Bhaskar, Maxwell Bland, Kirill Levchenko, and Aaron 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, August 2019. USENIX Association.

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

[26] Mike Bond, Omar Choudary, Steven J Murdoch, Sergei Skorobogatov, and Ross Anderson. Chip and Skim: Cloning EMV Cards with the Pre-play Attack. In *Proc. IEEE Symposium on Security and Privacy*. IEEE, 2014.

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

[28] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho 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.

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

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

[31] 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.

[32] 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.

[33] Cherita L. Corbett, Raheem A. Beyah, and John A. Copeland. Passive Classification of Wireless NICs During Active Scanning. *Int. J. Inf. Secur.*, 7(5):335–348, September 2008.

[34] Criminal Complaint, USA v Cristea et al, 4:16cr182. US District Court for the Southern District of Texas. https://www.courtlistener.com/recap/gov.uscourts.txsd.1357299.1.0.pdf, April 2016.

[35] Daniel Cross, Justin Hoeckle, Michael Lavine, Jason Rubin, and Kevin Snow. Detecting non-discoverable bluetooth devices. In *International Conference on Critical Infrastructure Protection*, pages 281–293. Springer, 2007.

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

[37] Boris Danev, Thomas S. Heydt-Benjamin, and Srdjan Čapkun. Physical-layer identification of rfid devices. In *Proceedings of the 18th Conference on USENIX Security Symposium*, SSYM'09, page 199–214, USA, 2009. USENIX Association.

[38] Boris Danev, Thomas S. Heydt-Benjamin, and Srdjan Čapkun. Physical-Layer Identification of RFID Devices. In *Proceedings of the 18th Conference on USENIX Security Symposium*, SSYM'09, page 199–214, USA, 2009. USENIX Association.

[39] The Ultimate Instore Carding by n3d from Darknet. http://wickybay.com/2017/10/ultimate-instore-carding-n3d-darknet/.

[40] DbaseJob. Carding!!! How To Make Your First Money. https://prvtzone.ws/threads/carding-how-to-make-your-first-money.5052/#post-20315.

[41] DEFCON23. Karl Koscher - Sniffing Scada. https://www.youtube.com/watch?v=4vPptUmyv4U.

[42] Loh Chin Choong Desmond, Cho Chia Yuan, Tan Chung Pheng, and Ri Seng 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.

[43] Tutorial Carding with Dumps. https://honeymoney24cc.com/cardingwithdumps.

[44] CC Dumps Shop. https://dumps.to/, February 2019.

[45] Electronic Transactions Association. ETA Statement on Visa and Mastercard's EMV Liability Shift Date Changes. https://www.electran.org/eta-statement-on-visa-and-mastercards-emv-liability-shift-date-changes/, 2016.

[46] Daniel 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.

[47] Kassem Fawaz, Kyu-Han Kim, and Kang G. Shin. Protecting privacy of BLE device users. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 1205–1221, Austin, TX, August 2016. USENIX Association.

[48] Jason Franklin, Damon McCoy, Parisa Tabriz, Vicentiu Neagoe, Jamie Van Randwyk, and Douglas 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.

[49] Julien 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.

[50] Branden Ghena, William Beyer, Allen Hillaker, Jonathan Pevarnek, and J. Alex Halderman. Green lights forever: Analyzing the security of traffic infrastructure. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, San Diego, CA, August 2014. USENIX Association.

[51] 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.

[52] Jeyanthi 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.

[53] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis. Detection of transient in radio frequency fingerprinting using signal phase. pages 13–18, 2003.

[54] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis. Detecting Rogue Devices in Bluetooth Networks using Radio Frequency Fingerprinting. In *In IASTED International Conference on Communications and Computer Networks*. Citeseer, 2006.

[55] 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.

[56] 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.

[57] Indictment, USA v. Rodriguez et al, 1:17cr417. US District Court for the Northern District of Ohio. https://www.courtlistener.com/recap/gov.uscourts.ohnd.237118.1.0.pdf, October 2017.

[58] Suman Jana and Sneha Kumar 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.

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

[60] 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.

[61] Soowon Kang, Hyeonwoo Choi, Sooyoung Park, Chunjong Park, Jemin Lee, Uichin Lee, and Sung-Ju Lee. Fire in Your Hands: Understanding Thermal Behavior of Smartphones. In *The 25th Annual International Conference on Mobile Computing and Networking*, MobiCom '19, New York, NY, USA, 2019. Association for Computing Machinery.

[62] I. O. Kennedy, P. Scanlon, and M. M. Buddhikot. Passive Steady State RF Fingerprinting: A Cognitive Technique for Scalable Deployment of Co-Channel Femtocell Underlays.

In *2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 1–12, Oct 2008.

[63] Moein Khazraee, Yeswanth Guddeti, Sam Crow, Alex C. Snoeren, Kirill Levchenko, Dinesh Bharadia, and Aaron Schulman. SparSDR: Sparsity-Proportional Backhaul and Compute for SDRs. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '19, page 391–403, New York, NY, USA, 2019. Association for Computing Machinery.

[64] 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.

[65] Krebs on Security. Skimmers Siphoning Card Data at the Pump. https://krebsonsecurity.com/2010/07/skimmers-siphoning-card-data-at-the-pump/, July 2010.

[66] Krebs on Security. Pro-Grade Point-of-Sale Skimmer. https://krebsonsecurity.com/2013/02/pro-grade-point-of-sale-skimmer/, February 2013.

[67] Krebs on Security. Gang Rigged Pumps With Bluetooth Skimmers. https://krebsonsecurity.com/2014/01/gang-rigged-pumps-with-bluetooth-skimmers/, January 2014.

[68] Krebs on Security. Tracking a Bluetooth Skimmer Gang in Mexico. https://krebsonsecurity.com/2015/09/tracking-a-bluetooth-skimmer-gang-in-mexico/, September 2015.

[69] Krebs on Security. ATM 'Shimmers' Target Chip-Based Cards. https://krebsonsecurity.com/2017/01/atm-shimmers-target-chip-based-cards/, January 2017.

[70] Schweitzer Engineering Laboratories. SEL-2925 Bluetooth Serial Adapter. https://selinc.com/products/2925/.

[71] Legitshop. Trusted Dumps with PIN. https://legitshop.org/, February 2019.

[72] 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.

[73] 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.

[74] Dan MacGuill. Can a Mobile Phone's Bluetooth Sensor Be Used to Detect Card Skimmers? https://www.snopes.com/fact-check/bluetooth-gas-pump-skimmers/, 2019.

[75] Jeremy Martin, Douglas Alpuche, Kristina Bodeman, Lamont Brown, Ellis Fenske, Lucas Foppe, Travis Mayberry, Erik Rye, Brandon Sipes, and Sam 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.

[76] Jeremy Martin, Travis Mayberry, Collin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Erik C Rye, and Dane Brown. A study of MAC Address randomization in mobile devices and when it fails. *Proceedings on Privacy Enhancing Technologies*, 2017(4):365–383, 2017.

[77] Jeremy Martin, Erik Rye, and Robert Beverly. Decomposition of MAC address structure for granular device inference. In *Proc. Annual Computer Security Applications Conference (ACSAC)*, 2016.

[78] Célestin Matte, Mathieu Cunche, Franck Rousseau, and Mathy 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.

[79] Meccadumps. Buy Dumps CVV online Fullz Verified seller . https://meccadumps.net/, February 2019.

[80] Memorandum and Order, USA v. Hristov et al, 1:10cr10056. US District Court for the District of Massachussetts. https://www.courtlistener.com/recap/gov.uscourts.mad.127405/gov.uscourts.mad.127405.62.0.pdf, April 2011.

[81] Mettler-Toledo. BC Shipping Scale Service Manual. https://thescalestore.com/manuals/Mettler-Toledo-BC-User-Manual-1.pdf, August 2015.

[82] MH Corbin Highway Information Systems. Surface Scan. http://mhcorbin.com/Portals/0/MH%20Corbin%20Surface%20Scan%20User%20Manual%20v1.1%20(002)%20new%20cover.pdf, January 2018.

[83] Everything you need to know about instore carding. http://wickybay.com/2017/11/everything-need-know-instore-carding/, November 2017.

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

[85] Vern 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.

[86] PCI Security Standards Council. PCI DSS Quick Reference Guide. https://www.pcisecuritystandards.org/documents/PCI_DSS-QRG-v3_2_1.pdf, 2018.

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

[88] PRTSHIP. DUMPS. https://prtship.com/forums/dumps.6/.

[89] Yakov Rekhter and Tony Li. Core Specification 5.3. Technical report, Bluetooth SIG, July 2021.

[90] Lawrence G. Roberts. Aloha packet system with and without slots and capture. *SIGCOMM Comput. Commun. Rev.*, 5(2):28–42, apr 1975.

[91] Mike 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.

[92] Santander Bank. What is my debit card spending/withdrawal limit? https://customerservice.santanderbank.com/app/answers/detail/a_id/3713/kw/atm%20withdraw/r_id/102441.

[93] N. Scaife, J. Bowers, C. Peeters, G. Hernandez, I. N. Sherman, P. Traynor, and L. Anthony. Kiss from a rogue: Evaluating detectability of pay-at-the-pump card skimmers. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1208–1222, Los Alamitos, CA, USA, may 2019. IEEE Computer Society.

[94] Nolen Scaife, Christian Peeters, and Patrick Traynor. Fear the Reaper: Characterization and Fast Detection of Card Skimmers. In *Proc. USENIX Security*, 2018.

[95] Scientific Working Group on Digital Evidence. Best Practices for Examining Magnetic Card Readers. https://www.swgde.org/documents/Current%20Documents/SWGDE%20Best%20Practices%20for%20Examining%20Magnetic%20Card%20Readers.

[96] Sell CVV (CC). https://sellcvvdumps.shop/.

[97] Souvik Sen, Božidar Radunovic, Romit Roy Choudhury, and Tom 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.

[98] Sentencing Memorandum of the United States, USA v. Aqel, 2:14cr270. US District Court for the Southern District of Ohio. https://www.courtlistener.com/recap/gov.uscourts.ohsd.178108/gov.uscourts.ohsd.178108.47.0.pdf, November 2015.

[99] 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.

[100] Skim Plus (Bluetooth Skimmer Detection). https://play.google.com/store/apps/details?id=com.rs.skimplus.beta, 2018.

[101] Dominic Spill and Andrea Bittau. Bluesniff: Eve meets alice and bluetooth. In *Proceedings of the first USENIX workshop on Offensive Technologies*, page 5. USENIX Association, 2007.

[102] Weiping Sun, Jeongyeup Paek, and Sunghyun 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.

[103] 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.

[104] 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.

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

[106] Teletrac. Teletrac Drive User Guide. http://community.teletrac.com/teletrac.com/assets/2014-04-23_android%20tablet%20user%20guide.pdf, January 2014.

[107] Carmela Troncoso, Mathias Payer, Jean-Pierre Hubaux, Marcel Salathé, James Larus, Edouard Bugnion, Wouter Lueks, Theresa Stadler, Apostolos Pyrgelis, Daniele Antonioli, Ludovic Barman, Sylvain Chatel, Kenneth Paterson, Srdjan Čapkun, David Basin, Jan Beutel, Dennis Jackson, Marc Roeschlin, Patrick Leu, Bart Preneel, Nigel Smart, Aysajan Abidin, Seda Gürses, Michael Veale, Cas Cremers, Michael Backes, Nils Ole Tippenhauer, Reuben Binns, Ciro Cattuto, Alain Barrat, Dario Fiore, Manuel Barbosa, Rui Oliveira, and José Pereira. Decentralized Privacy-Preserving Proximity Tracing, 2020.

[108] US Attorney's Office. Fourth Defendant Sentenced to Prison for Multi-Million Dollar Gas Pump Skimming Scheme. https://www.justice.gov/usao-nv/pr/fourth-defendant-sentenced-prison-multi-million-dollar-gas-pump-skimming-scheme, June 2023.

[109] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S. Cardoso, and Frank 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.

[110] VICE. Gangs on the Dark Web: Credit Card Scammers. https://www.youtube.com/watch?v=jT-jmq8KBw0, June 2018.

[111] Tien Dang Vo-Huu, Triet Dang Vo-Huu, and Guevara Noubir. Fingerprinting Wi-Fi Devices Using Software Defined Radios. In *Proceedings of the 9th ACM Conference on Security &#38; Privacy in Wireless and Mobile Networks*, WiSec '16, pages 3–14, New York, NY, USA, 2016. ACM.

[112] 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/, October 2018.

[113] 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.

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

[115] Pengyu Zhang, Dinesh Bharadia, Kiran Joshi, and Sachin Katti. Hitchhike: Practical backscatter using commodity wifi. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, SenSys '16, page 259–271, New York, NY, USA, 2016. Association for Computing Machinery.

[116] Pengyu Zhang, Colleen Josephson, Dinesh Bharadia, and Sachin Katti. Freerider: Backscatter communication using commodity radios. In *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '17, page 389–401, New York, NY, USA, 2017. Association for Computing Machinery.