



University of
Zurich^{UZH}

MAC Scavenger

A Passive Method Handling MAC Randomization on Mobile Devices

*Lenz Baumann
Zurich, Switzerland
Student ID: 10-741-577*

Supervisor: Bruno Rodrigues
Date of Submission: August 5, 2020

Abstract

Device fingerprinting is a controversial topic in the area of mobile devices. On the one hand, it can be a useful tool to derive mobility metrics in public spaces by capturing wireless signals emitted by these devices. On the other hand, is the possible privacy violation of the people carrying these devices. To prevent smartphones and other wireless-transmitting devices from being tracked while they move into Wi-Fi zones, a technique called MAC randomization is becoming popular to fully or partially modify the unique identifier of the physical network adapter by randomly generated values.

This thesis investigates strategies to circumvent such randomization strategies by applying signal strength-based localization to distinguish different devices otherwise non-distinguishable by only looking at the information they contain. It shows that even though in general very unreliable, RSSI values are an appropriate and easily accessible measure that can be used to localize a device in time and space. Based on this finding, the MacScavenger is described: A system that monitors devices over time, distinguishes different devices from each other, locates them in space, and can identify the amount of devices present.

Zusammenfassung

Das Identifizieren von mobilen Endgeräten anhand von sogenannten "Fingerabdrücken" ist weitverbreitet aber umstritten: Auf der einen Seite ist es eine relative einfache Möglichkeit auf öffentlichen Plätzen oder an Veranstaltungen Mobilitätsdaten zu erheben. Auf der anderen Seite stellt eine solche Datenerhebung eine Verletzung der Privatsphäre dar, da insbesondere anhand der Media Access Control (MAC) Adresse eine Person überwacht werden kann. Um diese Form der Überwachung zu verunmöglichen, etablierte sich in den letzten Jahren eine Strategie zur Randomisierung der MAC Adresse, welche diese in gewissen Situationen mit einer Zufallszahl ersetzt.

Diese Masterarbeit untersucht Strategien und Ansätze, welche diese Form der Anonymisierung zu umgehen vermögen. Als Lösung des genannten Problems wird ein kombinierter Ansatz vorgeschlagen: Gerätespezifische Informationen, welche in Probe Requests enthalten sind, werden durch die Position des Gerätes als zusätzliche Information ergänzt. Die vorliegende Arbeit vermag zu zeigen, dass einerseits der vielfach kritisierte Ansatz der RSSI-basierten Lokalisierung mehr als ausreichend ist um in diesem Zusammenhang von Nutzen zu sein und es andererseits die vorgeschlagene Kombination in der Tat vermag, Geräte voneinander zu unterscheiden. Das implementierte System *MacScavenger* vermag verschiedene Geräte voneinander zu unterscheiden, völlig ungeachtet der Tatsache, ob diese Geräte ihre MAC Adresse randomisieren oder nicht. Weiter ist es erfolgreich in der Lage die ungefähre Anzahl an Geräten im überwachten Raum abzuschätzen.

Acknowledgements

I like to thank Bruno Rodrigues, Prof. Burkhardt Stiller, and the CSG Team for their constant support and all their helpful input. Furthermore, I want to thank Simon Tuck and Thomas Bocek and the Livealytics team for making this collaboration possible. Furthermore, I would like to thank Cyrill Halter for his helpful inputs and the leads, he was always more than willing to share with me. Last but not least, I want to thank my fiancée for her constant support.

Contents

Abstract	i
Zusammenfassung	iii
Acknowledgements	v
1 Introduction	1
1.1 Motivation and Description of Work	1
1.2 Problem Description	1
1.3 Goals	2
1.4 Report Outline	3
2 Background and Related Work	5
2.1 Wireless Networks in General	5
2.1.1 Background	5
2.1.2 Properties	5
2.1.3 Wireless Medium	7
2.1.4 Nomenclature in Wireless Networks	8
2.1.5 IEEE	8
2.2 MAC Address	8
2.2.1 Background	8
2.2.2 The MAC Identifier	9
2.2.3 The Organizationally Unique Identifier (OUI)	9

2.2.4	The Company ID (CID)	10
2.2.5	Locally Unique Identifiers	10
2.3	The IEEE 802.11 Wireless Standard	11
2.3.1	General	11
2.3.2	802.11 Components	11
2.3.3	The 802.11 MAC Layer	11
2.3.4	802.11 Network Operations	13
2.3.5	802.11 MAC Frames	13
2.3.6	Probe Requests and Information Elements (IE)	16
2.4	MAC Randomization	17
2.5	Related Work	18
2.5.1	Current Situation in popular Operating Systems	18
2.5.2	Different Strategies to Circumvent MAC Address Randomization and their Flaws	19
2.5.3	Existing systems for De-Randomization	23
2.5.4	Discussion	23
3	MacScavenger	25
3.1	Proposed De-Anonymization Process	25
3.1.1	Localization	25
3.1.2	Kalman Filtering	27
3.1.3	Naming	27
3.2	Discussion of Proposed Approach	28
3.3	Requirements	28
3.4	Assumptions	29
3.5	Design Considerations	29
3.6	System Topology	30
3.6.1	Shell	30
3.6.2	Data Gathering Component	30

3.6.3	Data Analysis Component	31
3.7	Implementation	31
3.7.1	Languages and Frameworks	32
3.7.2	System Components	33
3.7.3	Main Processes	37
4	Evaluation	43
4.1	Experiment 1: Standalone Localization	43
4.1.1	Basic Idea	43
4.1.2	Methodology and Setup	43
4.1.3	Results	44
4.1.4	Conclusion	45
4.2	Experiment 2: Joint Localization	45
4.2.1	Basic Idea	45
4.2.2	Methodology and Setup	45
4.2.3	Results	46
4.2.4	Conclusion	47
4.3	Experiment 3: MacScavenger at Messe Basel	48
4.3.1	Basic Idea	48
4.3.2	Methodology and Setup	48
4.3.3	Results	48
4.3.4	Conclusion	49
5	Summary and Conclusion	53
5.1	Possible Future Improvements	54
	Bibliography	60
	Abbreviations	61
	List of Figures	62

List of Tables	64
List of Algorithms	65
A MacScavenger Github Repository	69
B Additional Files	71

Chapter 1

Introduction

1.1 Motivation and Description of Work

Passive wireless sensing is used commonly to gather information from devices inside a network. For instance, it is key for strategic business planning, visualizing, and tracking the flow of people that can be of interest. Monitoring human behavior can give an advantage by knowing precisely at any given time how many people gather at what position inside a building. Business owners can, for example, arrange their products corresponding to the points of interest inside their stores. The analysis of wireless signals emitted by portable devices such as smartphones, laptops, and tablets enables the extraction of the positional data from those devices. In public locations, security measures such as emergency routes can be enhanced by analyzing crowd behavior. Passive wireless sensing is a technology with high potential when thinking of statistical data, but is also prone to raise privacy issues since citizens unknowingly provide sensor data from their devices.

Nonetheless, as digitization is advancing, more and more parts of our daily lives are becoming transparent to all kinds of eyes and ears. The different devices carried around as telephones, laptops, watches and even hearing aids are exposing information about their carriers. As a central piece of information the MAC address can be used to link and connect different information, as it allows a unique identification of every device participating in modern data transmission. To protect the privacy of the users and prevent the abuse of the MAC address, manufacturers started to render the MAC address anonymous through a process called MAC randomization (or MAC address randomization).

1.2 Problem Description

When a device applies MAC address randomization, it constantly changes its MAC address. Therefore, it becomes useless as a unique identifier used in any device fingerprinting approach. This basic problem generates a variety of other problems:

- It becomes difficult to track a device in space over time, as it lacks a non-changing element on which it can be identified.
- It becomes difficult to count the number of devices the monitored traffic is originating from, as the MAC address is constantly changing, creating many "fake" devices that will bias the results.
- MAC randomization is nothing universal, nor is it defined in any standard by the Institute of Electrical and Electronics Engineers (IEEE). Therefore, it is possible, that one device is changing its MAC address constantly, while another device is revealing its true MAC address without any randomization.

This thesis revolves around these problems by discussing various approaches for device fingerprinting that attempt to overcome the difficulties imposed by MAC randomization. Furthermore, it proposes its approach, that can handle MAC randomization by fulfilling a variety of goals.

1.3 Goals

The goal of this work is to discuss the different possibilities to circumvent the difficulties introduced with MAC randomization and propose possible solutions. A possible solution to deal with MAC address randomization in the context of device tracking and counting, is designed in the form of a system, able to count and track devices in time and space and to distinguish devices from each other without actively interacting with the devices themselves. For the proposed system, three specific goals are defined in the following way:

- The **primary** goal is the ability to tell if captured traffic is originating from the same device or not, regardless of the information revealed by its randomized or non-randomized MAC addresses present in the data.
- The **secondary** goal is to count the number of devices present by knowing which portions of traffic originate from the same device and hence correcting the bias introduced by MAC randomization.
- The **third** goal, is to localize devices in space. By knowing how many devices are present and by being able to distinguish devices, it becomes possible to track a single device in the covered area over time.

Following these goals, the proposed system is completely passive and uses a combined approach of Received Signal Strength Indicator (RSSI) value-based localization and the Information Elements (IE) transmitted in every 802.11 probe request frame. In a series of experiments, this thesis shows not only the feasibility of the proposed approach, but also the strengths of the MacScavenger system compared to other systems.

1.4 Report Outline

The thesis is structured the following way: Chapter 2 introduces MAC randomization and its background before discussing a range of approaches to deal with MAC randomization, along with their strengths and weaknesses. Chapter 3 describes the principles and main ideas of the proposed system as well as its implementation and topology. Chapter 4 evaluates the proposed approach and the MacScavenger system itself. Chapter 5 summarizes the contribution of this thesis to the research on MAC randomization and outlines possible future research and improvements.

Chapter 2

Background and Related Work

This chapter describes the necessary background, the underlying principles, and the technologies required to understand the main issues and questions targeted in this thesis. The Section *Related Work* introduces the current state of research, as well as existing ways and approaches to deal with MAC randomization. It leads over to the next chapter describing the proposed system.

2.1 Wireless Networks in General

2.1.1 Background

While the first wireless transmissions were conducted in the middle of the 19th century[1], it took almost a hundred years more until in 1971 the first Wireless Local Area Network (WLAN) was deployed at the University of Hawaii (ALOHAnet [2]), towards different islands to be able to connect to the main computer on the main island [3].

2.1.2 Properties

As can be seen in the case of the ALOHAnet, WLANs complement but cannot replace fixed networks [4]. They are, however, a suitable choice in areas that are difficult to install cables or in environments where the mobility and flexibility of the user are prioritized [4].

Therefore, the many advantages of wireless communication over traditional fixed network infrastructures are accompanied by multiple drawbacks, as shown in the following paragraphs:

Strengths

- **Mobility:** Wireless communication enables higher flexibility concerning data access even while in motion, as the data transmission does not rely on cables but *e.g.* Radio Frequency (RF) or infrared as medium [4][5][6].
- **Ease and Speed of Deployment:** Wireless connections provide a straightforward deployment and operation, such as to add and remove components from the network, as no cabling is required. Especially when installing cables is difficult or even impossible (as with the ALOHAnet) wireless networks offer a huge advantage [4].
- **Flexibility:** As no re-cabling is necessary, easy expansions of the existing infrastructure are possible on-the-fly and with minimal effort. The infrastructure of a wireless network does not change qualitatively, whether you are connecting one or a million users [4]. For instance, to offer service in a specific area, all you need is base stations and antennas [4][5].
- **Costs:** Due to its flexibility and universal applicability, the ease of deployment, the simple interfaces, and the extensible nature of wireless networks, the costs to deploy such networks are really low. Additionally, traditional costs to maintain cabled networks are no longer present. As an amplifying factor, costs for the necessary hardware to build and deploy wireless network infrastructure are low when compared to the overall costs a cabled network produces [4][5].

Weaknesses

- **Lack of physical boundaries:** Network security for *fixed networks*, protects fixed and well-defined paths. Hence, protecting the network's physical infrastructure, such as the building itself or the wiring cabinets, suffices to ensure the network's integrity. *Wireless networks* operate in an open medium, that does transmit data using a designated link with a particular encoding and modulation. Therefore, signals can be sent or received by anybody in possession of the radio techniques [4].
- **Dynamic physical medium:** Wired networks are very predictable and controllable: It is easy to increase their capacity by adding the switches or routers and by patching new network cables without complicating the network. Wireless networks, however, are much more dynamic. *E.g.* radio waves bounce off or penetrate walls and often behave unexpectedly as they suffer from several propagation issues that may interrupt the radio link. Lacking a reliable network medium, wireless technologies are required to compensate with careful frame validation to guard against frame loss, resulting in complex protocols and slow data transmission [4].
- **Speed** Another, closely related, difficulty that arises due to the medium itself is the impossibility to increase the speed of transmission: Its not possible to increase stability and transmission strength and increase the volume of transmitted data at the same time. As the radio band is fixed, it can only transmit more data at the cost of losing efficiency [4].

- **Limited Coverage** Wireless mediums are broadcast mediums [4]. Therefore, when one station transmits, all other stations must listen. Access Points (AP)s, however, allow a fixed amount of transmission capacity per AP, that must be shared by all attached users. Adding capacity, therefore, requires the network administrator to add access points while simultaneously reducing the coverage area of the other access points [4].
- **Security:** Wireless networks require additional strong encryption and authentication features to prevent traffic interception or injection, as 802.11 did not define the necessary security protocols right from the start [4][5]

2.1.3 Wireless Medium

Wireless networks need a medium that offers broad areal coverage as well as the ability to penetrate obstacles. While early wireless networks used infrared light as medium [6], that was blocked by walls, today, most, if not all, wireless technologies nowadays use radio waves (Radio Frequency)[4].

Radio waves are split into different bands defined by the frequency range the waves have. Therefore, wireless devices are constrained to operate in a certain frequency band. Each band has an associated bandwidth, that contains the amount of frequency space in the band [4]. The available radio spectrum allocation is rigorously controlled by regulatory authorities through licensing. The table 2.1 describes the different bands and their frequency range.

Table 2.1: The different licensed and unlicensed radio frequency bands by [4]

Band	Frequency Range
UHF ISM	902-928 MHz
S-Band	2-4 GHz
S-Band ISM	2.4-2.5 GHz
C-Band	4-8 GHz
C-Band satellite downlink	3.7-4.2 GHz
C-Band radar (weather)	5.25-5.925 GHz
C-Band ISM	5.725-5.875 GHz
C-Band satellite uplink	5.925-6.425 GHz
X-Band	8-12 GHz
X-Band radar (police/weather)	8.5-10.55 GHz
Ku-Band	12-18 GHz
Ku-Band radar (police)	13.4-14 GHz & 15.7-17.7 GHz

The Industrial, Scientific and Medical (ISM) bands are set aside specifically for equipment related to industrial, scientific, and medical use¹. And, while most bandwidths require expensive licensing and registration, the ISM bands are free to use, provided that the

¹Also the microwave oven operates in the 2.4 Giga Herz (GHz) ISM band as this frequency is considered as the most effective for heating water [7]

devices comply with power constraints [4]. 802.11 wireless networks operate in the ISM bands along with many other devices commonly used in wireless network infrastructure [4].

2.1.4 Nomenclature in Wireless Networks

Over time, different names for the same technologies have evolved. While some use these names interchangeably, each term highlights a different aspect of wireless technology and, therefore, has a slightly different meaning:

- **Wireless Ethernet:** Highlights the shared lineage with the traditional wired Ethernet, of whom it is considered a "branch" [4].
- **Wi-Fi:** Evolved from the certification program for interoperability run by the Wi-Fi Alliance², the major trade association of 802.11 equipment vendors [4].
- **WLAN:** This term is referring to any kind of Wireless Local Area Network infrastructure and would not need to be Wi-Fi interoperable. However, due to the dominance of the 802.11 standard, most modern WLAN are 802.11 based and comply with the Wi-Fi standard [4].

2.1.5 IEEE

The IEEE³ traces its origins back to 1884, when it was founded under the name of *American Institute of Electrical Engineers (AIEE)* with the vision to "support professionals in their nascent field and to aid them in their efforts to apply innovation for the betterment of humanity" [9].

Today the IEEE focuses mainly on standardizing electrical equipment, including communication technology: It is responsible for elaborating the standards that define all 802.11 based communication, and, therefore, all wireless processes relevant to this work. It is also responsible for the assignment of unique identifiers (such as *e.g.* the MAC address) in a way that makes the assignment available to interested parties [10].

2.2 MAC Address

2.2.1 Background

Originally, Xerox Corporation was the registration authority for Ethernet parameters that assigned an ID called *Block ID* [11]. As of today, this task is done by the IEEE

²The Wi-Fi Alliance, also known as the Wireless Ethernet Compatibility Alliance (WECA), tests products for compliance with the 802.11 standard [8]

³pronounced "Eye-triple-E"[9]

Registration Authority (IEEE RA). It is responsible for the assignment of address blocks to organizations in exchange for a fee, to guarantee the uniqueness of devices [4]. These address block assignments are publicly available. Even though it is possible to acquire a non-public listing option, according to [12] this possibility is not used by any major company.

2.2.2 The MAC Identifier

Even though the unique identifier of a device is commonly called *MAC address*, according to the IEEE documentation [13], the terms MAC or MAC-48, are deprecated and replaced by the term EUI-48 (Extended Unique Identifier 48) [11]. And while many alternatives exist to the EUI-48 as basis for a MAC address assignment, such as EUI-64 or Extended Local Identifier (ELI), in most cases the EUI-48 is used. To minimize complexity of the possible types of address block assignments through the IEEE, it is assumed that a EUI-48 is used and a fully qualified 24-bit Organizationally Unique Identifier (OUI) is purchased through the assignment of an MA-L address block [14].

An EUI-48 is structured into two 3-octet blocks: A 24-bit OUI assigned to the manufacturer by the IEEE RA and a 24-bit additional identifier assigned by the OUI holder, called the Company ID (CID). Each OUI and CID are unique with respect to all assigned OUI's by the IEEE RA and all CID's defined by the OUI holder. Both OUI and the CID can be represented as a base-16 number of six hexadecimal digits. Alternatively they can be represented as octets separated by hyphens. However, the IEEE RA prefers the hexadecimal representation.

2.2.3 The Organizationally Unique Identifier (OUI)

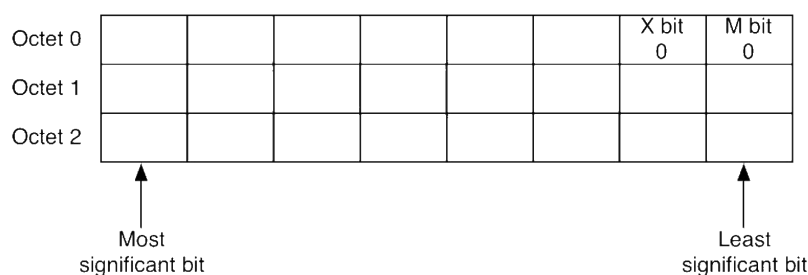


Figure 2.1: The Organizationally Unique Identifier (OUI). Figure by [13].

The OUI is a 24-bit (three-octet) sequence: The first octet is the initial (most significant) octet. The least and second significant bits of octet 0 are designated as M-bit (least significant bit) and X-bit (second least significant bit). In the OUI both M and X bit have value 0.

When an EUI-48 is used as a basis for a MAC address assignment (for example, an IEEE 802 network address), the two least significant bits of the initial octet (Octet 0 = first octet of the OUI) are used for special purposes [13]:

- The *group*-bit, also *I/G*-bit or *M*-bit (c.f. Figure 2.1) has the value 01 (= 00000001)
- The *local*-bit, also *U/L*-bit or *X*-bit (c.f. Figure 2.1) has the value 02 (= 00000010)

OUIs and longer MAC prefixes are assigned with the *local*-bit zero and the *group*-bit unspecified. Multicast identifiers may be constructed turning the *group*-bit to one and the Unicast identifiers may be constructed by leaving the *group*-bit zero.

2.2.4 The Company ID (CID)

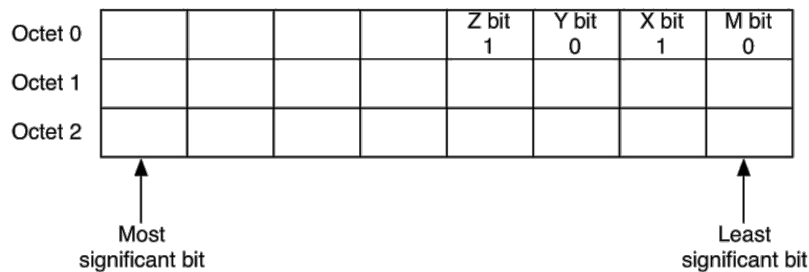


Figure 2.2: The Company ID (CID). Figure by [13].

The CID is a 24-bit (three-octet) sequence that has a different structure than the OUI: Octet 0 is the initial (most significant) octet. The four least significant bits of octet 0 are M (least significant bit of octet 0), X (second least significant bit of octet 0), Y, and Z. In the CID, the M, X, Y, and Z bits have the values 0, 1, 0, and 1, respectively [13]. The representation of the CID is equal to the previous identifiers: It can be represented as base-16 number of six hexadecimal digits or as octets separated by hyphens.

2.2.5 Locally Unique Identifiers

In addition to the globally unique EUI-48-based MAC addresses, modern devices use locally assigned addresses which are distinguished by a 1 in the U/L-bit of the first octet of the OUI. Local MAC addresses are used in contexts such as multi-Service Set Identifiers (SSID)s, configured APs, mobile device-thethered hotspots and Peer-to-Peer (P2P) services [15][4].

For globally unique EUI-48 identifiers assigned by the owners of an OUI or longer prefixes, the *Local* bit is zero. If the *Local*-bit is one, the identifier is under the control of the local network administrator, and the identifier is not globally unique anymore. However, the holder of an OUI loses its authority over MAC addresses generated based on the OUI, if the local bit is on. If that is the case, it is a locally unique identifier [15][4].

2.3 The IEEE 802.11 Wireless Standard

2.3.1 General

The 802.11 standard ”specifies a set of MAC and physical-layer protocols for implementing WLAN” [16]. It is part of the 802 family, that represents multiple networking standards, as seen in Figure 2.3.

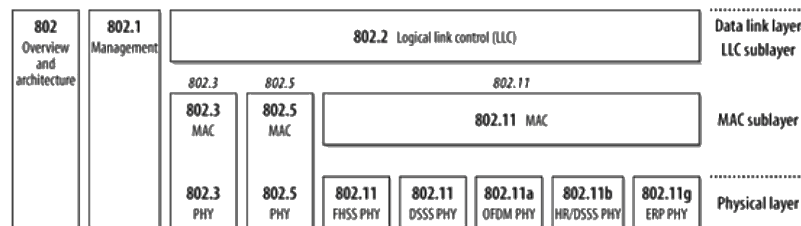


Figure 2.3: The IEEE 802 family and its relation to the OSI model [4].

The IEEE 802.11 specifications focus on the two lowest layers of the Open Systems Interconnection (OSI) model since they incorporate both physical and data link components (layers 1 and 2 *c.f.* Figure 2.4). The MAC-layer provides a set of rules determining how to access the medium and send data, but the details of transmission and reception are left to the PHY layer (layer 1) [4].

2.3.2 802.11 Components

Basic components of 802.11 are the *stations*, the *access points*, the *wireless medium* and the *distribution system* (*c.f.* Figure 2.5). *Stations* are computing devices with wireless network interfaces. They do not necessarily need to be mobile devices [4]. *Access Points* works as bridges between the wireless medium and the rest of the world. They convert wireless frames to frames that can be transmitted over fixed network connections [4]. The *wireless medium* is the medium the data is transmitted on. By far the most popular are RF physical layers [4]. The *distribution system* is responsible for the communication between the access points. When several APs are connected to form a large covered area, they need to be able to communicate with each other to track the movement of the mobile devices [4].

2.3.3 The 802.11 MAC Layer

The 802.11 MAC layer can be seen as a special kind of link layer that uses the 802.11.2/LLC encapsulation also used by wired network communication [4]. However, as the 802.11 MAC-layer allows for mobile network access, a number of additional features were incorporated into it. As a result, the 802.11 MAC specifications are not straightforward in contrast to other IEEE 802 MAC specifications [4].

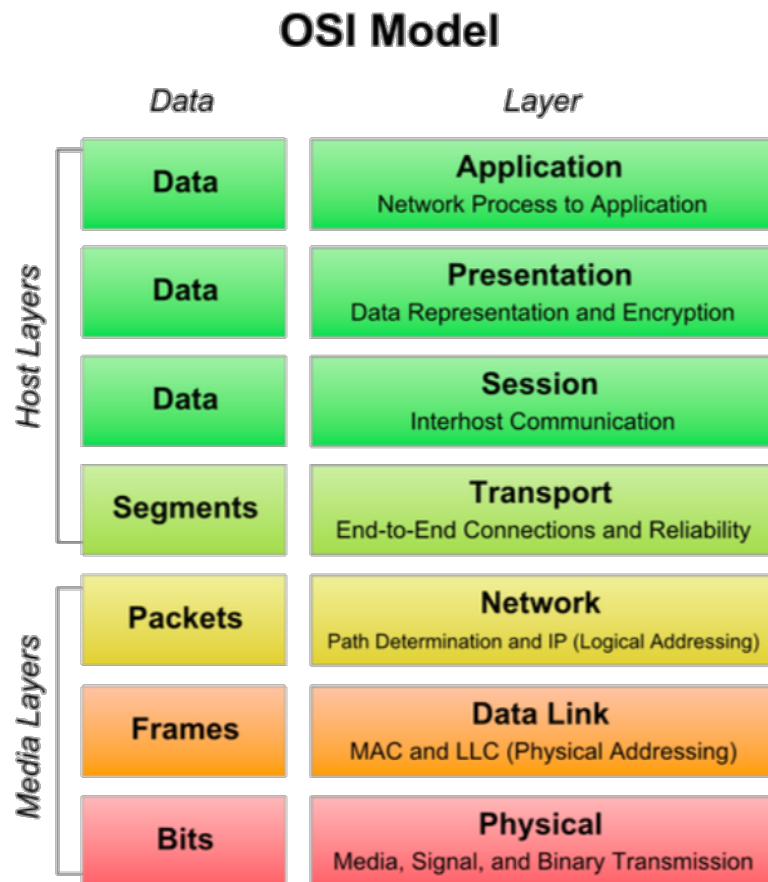


Figure 2.4: The OSI model [17].

As the wireless medium introduces many complex problems, compared to wired networks, the 802.11 MAC layer has to deal with a variety of challenges:

RF Link Quality: Radio Links are less stable than traditional Ethernet cable links due to the medium they use. Therefore, RF-based transmissions are bound to interfere with each other at some point, especially because most devices use the same unlicensed ISM RF bands (*cf.* Table 2.1). To work around these disturbances during the signaling process, 802.11 introduces multiple measures to increase the link stability. One such measure is *positive acknowledgement*. With positive acknowledgment, every successfully transmitted frame must be acknowledged by the receiver. If any part of the transfer - either the transfer of the original data or the transfer of the acknowledgment - fails, the frame is considered lost [4].

Data Transmission Speed: The quality of the radio link also has a strong influence on the network transmission speed. A high data link stability results in poor data transmission speed and vice versa. Therefore, stations must implement methods for determining when to switch the data range in response to changing conditions. Besides, it must be possible to run several different data rates at the same time with different stations [4].

Hidden Node Problem: In wireless networks, nodes may not always know about each other: Hence, it can occur that two nodes, that both do not see each other, can see the

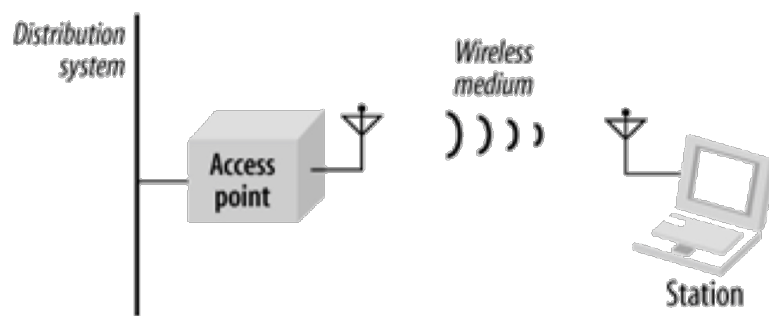


Figure 2.5: Components of 802.11 Local Area Network (LAN)s [4].

third node. If they simultaneously start sending data to the third node, this node may fail to make sense of the data. Such collisions resulting from hidden nodes are hard to detect in wireless environments because wireless transceivers are generally half-duplex [4]: They do not transmit and receive at the same time. To prevent collisions, 802.11 allows stations to use special frames to clear out an area for a specific time interval: Request-to-Send (RTS) and Clear-to-Send (CTS) frames. The RTS frame reserves the radio link for transmission and silences stations that hear it. If the targeted station receives an RTS it responds with a CTS. The CTS frame silences again all stations in the immediate vicinity, but this time from the receivers point of view [4]. Once the RTS/CTS exchange is complete, stations can transmit frames without worrying about interference from any hidden nodes. The transmission ends with the positive acknowledgment [4].

2.3.4 802.11 Network Operations

In the 802.11 standard, stations are identified by 48-bit IEEE MAC addresses (*cf.* Subsection 2.2.2). Conceptually, frames are delivered based on the MAC address. However, as frame delivery is unreliable with the medium 802.11 uses, 802.11 incorporates many mechanisms to ensure reliability, such as additional services and complex framing, described in Subsection 2.3.3 [4].

Nine services are defined by 802.11, that each vendor has to implement to create functional software and hardware. Only three thereof are used for moving data, while six are management operations that allow the network to keep track of mobile nodes and deliver frames accordingly [4]. Table 2.2 summarizes theses nine services.

2.3.5 802.11 MAC Frames

To be able to fulfill all the services in Table 2.2, 802.11 defines three types of frames [4]:

- **Management Frames:** Management frames take the biggest percentage of all 802.11 frame types as they are responsible for identification, association/disassoci-

Service	Station or Distribution Service	Description
Distribution	Distribution	Used in frame delivery to determine destination address in infrastructure networks
Integration	Distribution	Frame delivery to an IEEE 802. LAN outside the wireless network
Association	Distribution	Used to establish the AP which serves as the gateway to a particular mobile station
Reassociation	Distribution	Used to change the AP which serves as the gateway to a particular mobile station
Disassociation	Distribution	Removes the wireless station from the network
Authentication	Station	Establishes station identity (MAC address) prior to establishing association
Deauthentication	Station	Used to terminate authentication and by extension association
Transmit Power Control (TPC)	Station/Spectrum Management	Reduce interference by minimizing station transmit power
Dynamic Frequency Selection (DFS)	Station/Spectrum Management	Avoids interfering with radar operation in the 5 GHz band

Table 2.2: Overview of network services according to [4]

ation processes and the overall management of the network [4]. In the context of this work the most important management frame is the **Probe Request**.

- **Control Frames:** Control frames assist in the delivery of data frames. They administer access to the wireless medium and provide MAC-layer reliability functions [4]. Typical control frames are RTS, CTS and Acknowledgement (ACK) frames.
- **Data Frames:** Data frames carry higher-level protocol data in the frame body. Depending on the type of service provided by the AP, different data frame types are used.

Structure of a MAC frame

The 802.11 MAC frames do not include some of the classic Ethernet frame features, such as *e.g.* type/length fields or the preamble (c.f. Figure 2.6). Instead, the "preamble is part of the physical layer and encapsulation details such as type and length are present in the header of the data carried in the 802.11 frame" [4].

The 802.11 MAC frame consists of the following fields:



Figure 2.6: Generic 802.11 MAC frame. Figure by [4].

- **Frame Control Field:** Each frame starts with a two-byte Frame Control subfield with the components shown in Figure 2.7. Among others, the frame control field includes the protocol version, type and sub-type fields, a fragmentation flag, and a retry flag [4].
- **Duration/ID Field:** The duration/id field, follows the frame control field and has several uses (*cf* Figure 2.8). In most cases it indicates the number of microseconds, the medium is expected to remain busy for the transmission currently in progress [4].
- **Address Fields:** Contain up to four address fields for usage in different purposes: Generally, address 1 is used for the receiver, address 2 for the transmitter, and address 3 for filtering by the receiver. Addresses have the form of MAC addresses (*cf*. Section 2.2.2), defined by the I/G and the U/L bits [4].
- **Sequence Control Field:** Used for both de-fragmentation and discarding duplicate frames. It is composed of a 4-bit fragment number field and a 12-bit sequence number field as can be seen in Figure 2.9. The sequence number subfield acts as a modulo-4096 counter of the frames transmitted.
- **Frame Body:** The frame body or data field, moves higher-level payload from station to station. Originally, 802.11 can transmit frames with a maximum payload of 2304 bytes of higher-level data. While Ethernet does pad its payload to a minimum length, 802.11 does not [4].
- **Frame Check Sequence:** As with the Ethernet, the 802.11 frame closes with a Frame Check Sequence (FCS) also called Cyclic Redundancy Check (CRC). This frame sequence allows it to determine whether a sequence was damaged in transit or not. However, different than in Ethernet connections, in 802.11 networks, even a positive CRC of the FCS must still be positively acknowledged.

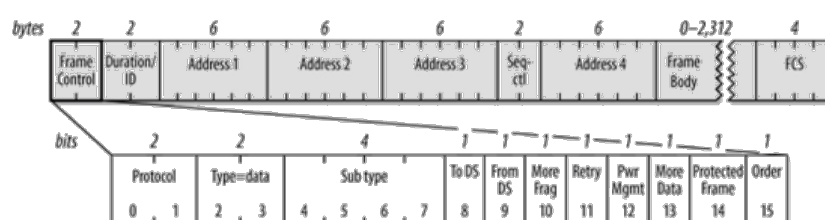


Figure 2.7: Frame control field. Figure by [4].

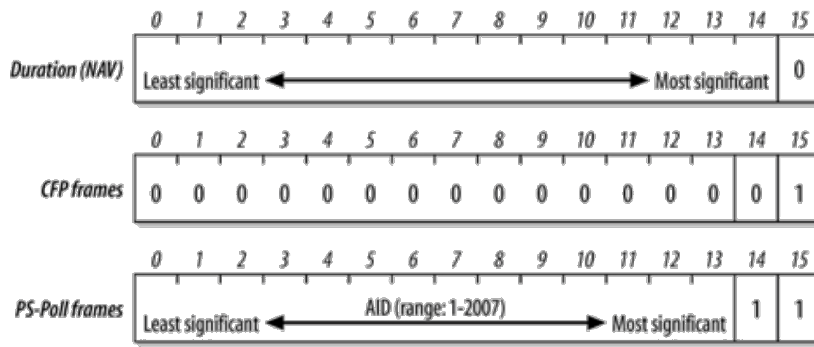


Figure 2.8: Duration/ID field. Figure by [4].

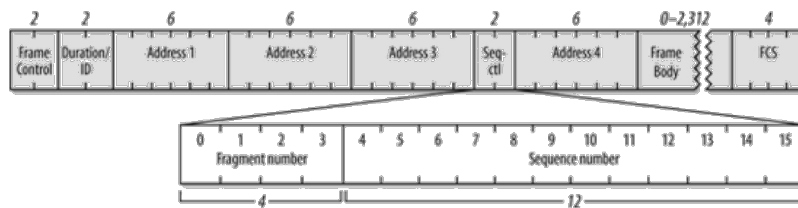


Figure 2.9: Sequence control field. Figure by [4].

2.3.6 Probe Requests and Information Elements (IE)

802.11 probe requests are management frames [18], that follow the structure shown in Figure 2.10. The frame consists of various fixed-length header fields and a body of variable length consisting of a varying amount of IE: Little pieces of information, that reveal certain properties about the sender. Each IE consists of a 1 octet Element ID field, a 1 octet Length field and a variable-length element-specific Information field [13]. An exhaustive list of IEs can be found in [13].

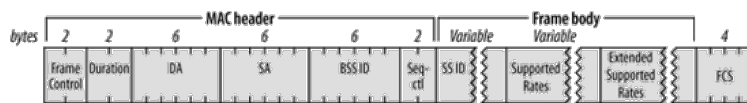


Figure 2.10: Probe Request Structure. Figure by [4].

Probe Requests are broadcast in bursts (*cf.* Figure 2.11) at a high frequency by many devices containing a 802.11 capable network card. Their purpose is to discover APs to associate with [15][4]. The amount of bursts, the amount of probe requests contained in each burst, and the exact timing between each packet, depends on several factors such as screen state, charging state, airplane mode, Wi-Fi setting screen open, Bluetooth activation, and proximity of a known network [15].

In the context of this work, probe request frames are the most important frames as they are the prime target for passive monitoring of network devices. As a source of information constantly leaked by devices, they can be used to infer more information on these devices

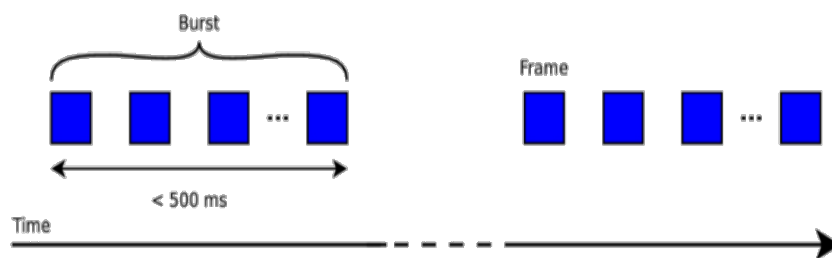


Figure 2.11: Probe Request Bursts. Figure by [19].

and their carriers [15]. Therefore, it is possible to passively monitor the medium for such packets that do not only contain device-specific information but also the MAC address for globally unique identification.

2.4 MAC Randomization

Having globally unique identifiers makes it easier to avoid any kind of collisions on the MAC-layer protocol. For that matter, the IEEE made it mandatory for any company selling components operating on that layer to purchase an OUI, combine it with an CID and to use it to assign a unique address to every component [15]. However, this approach has a major drawback: As shown in Subsection 2.3.6, packets can be captured and their unique identifier can be extracted, allowing it for anyone to track devices and their activities. This potential violation of each users privacy, lead to the introduction of the *MAC randomization* (also: *MAC address randomization*) technique [15].

The MAC randomization, is a technique that allows devices to replace their MAC address with a random value in order to prevent observers from "singling out their traffic or physical location from other nearby devices" [12]. These, locally assigned, addresses are used *e.g.* when devices are performing active scans by broadcasting probe requests.

Using a locally assigned MAC address, however, is only possible in disassociated state. As soon as the station associates with an AP, it is forced to reveal its real MAC address [12]. As shown in Subsection 2.2.5, locally assigned addresses are under the authority of the network administrator and do not need to be globally unique. Hence, they can be generated in several ways as long as they do not violate IEEE 802.11 standards, such as using an OUI assigned to another manufacturer.

As this process of randomizing the MAC address is not standardized, multiple ways exist to render the MAC address identifier anonymous. At the same time, most of the applied techniques have flaws at different levels of their implementation. The field of research into de-anonymizing MAC addresses, therefore, is enormous and shall be discussed in more detail in the next section.

2.5 Related Work

This section starts with an overview of the different strategies applied by different manufacturers and operating systems. Then it overviews the countless different ideas and approaches to deal with MAC randomization. Further, it introduces software systems that attempt to overcome MAC randomization. After a final discussion, it leads over to the next chapter, proposing a basic approach.

2.5.1 Current Situation in popular Operating Systems

MAC randomization-related issues are not a specific problem of mobile devices. Most operating systems that use 802.11 for wireless connections rely on MAC randomization when in non-associated state [15]. However, the fundamental principle of tracking devices is to be able to create a mapping from devices to people. Therefore, in this context, mobile devices are the most relevant types of devices, as they are closely attached to their carrier and rarely leave the side of the person who owns them. On the contrary, they can be thought of a representative of the person who carries them.

Provided that, it is most interesting to investigate how different mobile Operating System (OS)s behave, regarding MAC randomization. In the following, the use of MAC randomization in the most common⁴OSs is discussed.

Android

Android uses randomization from version 6.0 on for background scans if the driver and hardware support it [22]. Although Android versions before 6.0 do not support randomization, several applications supporting this feature have been released (*e.g.* [23]). Those applications allow the user to replace the MAC address by a random value. Such operations, however, usually require root privilege to work, which reduces their impact on the average user [22].

iOS

Apple introduces MAC address randomization with iOS 8 [12]. Additionally, [12] found out, that MAC addresses derived from iOS Apple devices do not share any common OUI. Exceptions are the correct values for the U/L and the I/G bits. Statistical tests, however, show, that Apple MAC addresses are uniformly random distributed, which leads to the conclusion, that iOS implements truly randomized MAC addresses and does violate the IEEE assigned number space, by using address space assigned to other companies[12].

In iOS 10, a vendor-specific IE was added to all transmitted probe requests. This trivializes the identification of iOS 10 Apple devices. However, this specific IE field never changes across Apple devices and does not allow to cluster or identify specific MAC addresses [12].

⁴According to various online sources, as by 2019, iOS and Android share around 98% of the market for mobile OSs[20][21].

2.5.2 Different Strategies to Circumvent MAC Address Randomization and their Flaws

The family of de-anonymization strategies is large and complex, the classification proposed in this work, therefore, neither exhaustive nor exclusive. Hence, other hierarchies and structures would be possible as well. In the context of this work, only approaches are discussed, that attempt to **passively** identify a **specific device** by creating a **unique fingerprint**. Approaches such as device classification, device inference, or approaches that actively interact with the devices are omitted.

Fingerprinting describes the process of generating a unique fingerprint for a device. This process uses data collected about a device and tries to extract or create a unique identifier for each device, such that it can be distinguished from other devices [15]. A perfect fingerprint is unique and able to successfully identify a device from all other existing devices. In practice, however, this is unlikely to be achieved.

The discussed approaches below all attempt to create unique fingerprints, differing only in the kind of information used.

Physical Fingerprinting

Due to slight variations during the manufacturing process along with their specific age, components of mobile devices possess singularities which make them all behave in a slightly different way. While this fact usually does not disrupt the normal functioning of the component, it can be exploited to fingerprint the device [15]. This PHY-level data contains many fields that can be exploited. To minimize complexity, a non-exhaustive list of features derived from PHY-level data is presented:

- **Carrier Frequency Offset:** This occurs due to small differences in the carrier frequency of the transmitting and receiving radio. This leads to a time-varying phase offset across each frequency band. These differences vary across transmitters and receivers. Applied *i.a* by [24], [25] and [26].
- **Scrambler Seed:** According to the IEEE 802.11a/g standard, a Wi-Fi transmitter shall generate a new random scrambling seed for every transmission of a PHY-layer frame to mitigate errors in the transmission. Due to its specific implementation however, the seed used by the scrambler can be reconstructed and be used to identify specific transmitters, as the seeds differ from transmitter to transmitter [26]. The scrambler is considered to be a feature of the 802.11 Wi-Fi physical layer and not on the same level as other physical features such as e.g. wave modulation according to [15]. Applied *i.e* by [26] and [22].
- **Wave form and wave modulation:** The 802.11 standard defines two popular modulation techniques: Direct-Sequence Spread-Spectrum (DSSS) and Orthogonal Frequency-Division Multiplexing (OFDM) [22]. These techniques are used to increase the robustness of data transmission and to optimize data transmission or

receipt. To do so, they adapt the waveforms and the frequency of the RF [18]. Devices are distinguished using differences between amplitude, the phase, and the frequency of the waves themselves [27]. Applied *i.a* by [26] and [22].

- **Clock Skew Variation:** Relies on small irregularities in the hardware clock of the transmitter. Based on Transmission Control Protocol (TCP) timestamps, these irregularities or offsets can be used to fingerprint devices. Applied *i.a* by [28] and [29]

MAC-layer-based Fingerprinting

In contrast to physical-layer fingerprinting, MAC layer fingerprinting does not rely on raw radio signals. Instead, one can use off-the-shelf Wi-Fi hardware to process the radio signal and analyze the received MAC-layer data. To do so, network interfaces are typically configured to run in "monitor mode": This mode allows to capture frames regardless of their destination MAC address [27].

MAC-layer based fingerprinting, makes use of the features present within frames at the MAC link layer. For the purpose of this work, we split the different features into the groups *content-based*, *timing-based* and *geospatial-based* and elaborate in more detail on the three groups below:

Content-based De-Anonymization

Includes approaches that make use of specific fields present in 802.11 network frames. All content-based approaches necessarily rely on 802.11 packets such as management or control frames. Typical fields used for content-based fingerprinting are:

- **Sequence Number:** Some OSs do not reset their sequence number between probe request bursts and therefore reveal that they come from the same source [15]. Hence, rendering MAC randomization useless. Applied by *i.a.* [12] and [22]
- **MAC address:** The MAC address is a common identifier for identifying a device, as it is "disclosed by devices on a regular basis, even when the device is not associated with an AP" [27] (*cf.* Probe Requests 2.3.6). Its global uniqueness makes it a perfect match for tracking a device [27]. However, it is due to this aspect, that MAC address randomization started to come into being. Applied by *i.a.* [30] and [19].
- **Header Information:** Even if the information present in the and through the MAC address directly is undisclosed due to MAC address randomization, the header contains several other relevant pieces of information, such as *e.g.* the *sequence number* (*cf.* Subsection 2.3.3) [27]. Applied by *i.a.* [22], [19] and [27]
- **Per-Bit MAC Header Analysis:** Many approaches use a relatively small amount of information (mostly specific fields) and discard all other information. Another approach is to use the complete header information and learn which bits of the

overall frame are useful for fingerprinting and which are not [27]. Using any suitable machine learning approach leads in the best case to an increase in information extraction while at the same time decreasing the amount of useless data. Applied by *i.a.* [27].

- **Information Elements (IE)** (cf. Section 2.3.6): IEs are the fields embedded in probe requests and beacon frames. They contain information needed to establish a connection between a client and an AP. Typical entries are the supported data rates, known SSIDs, security and communication network capabilities, vendor specific data and others. As IEs contain many hard- and software related information, they showed to be a reliable source for fingerprinting. Even if the MAC address changes between frame bursts, the IE of the frames themselves are not changing. Applied by *i.a.* [22] and [19].
- **MAC Layer Management Entity (MLME)**: The Wi-Fi MLME comprises a number of different types of packets used in the operation of the Wi-Fi network. Most Wi-Fi MLME frames consist of a set of fixed parameters which are always present followed by tagged parameters. The information present in the MLME can be exploited in a similar manner as with the Probe Requests [31]. Applied by *i.a.* [31].
- **Wi-Fi Protected Setup (WPS)**: One of the IE found in probe requests is dedicated to WPS, a protocol simplifying device pairing. Using this field, it is possible to reveal the true not randomized MAC address of a device as it contains the Universally Unique Identifier (UUID). The UUID, however, is per specification calculated by applying a hash-digestion function that digests the real MAC address and outputs a hash. If the function is known, the MAC randomization process can be reverted and the real MAC address can be revealed. This technique is called UUID reversal technique[32]. Applied by *i.a.* [32] and [22].

Timing-based De-Anonymization

In active scanning, devices send probe request frames to detect APs within their transmission range and to provoke a response. For each channel, the client broadcasts a probe request and starts a timer. If the timer reaches a threshold, the channel becomes idle and the client scans the next channel. Then, the client waits until another timer reaches its threshold and processes the received probe response frames, and then scans the next channel [33]. Further specifications of the active scanning function is not provided in the IEEE 802.11 standard. As a result, implementing active scanning within wireless drivers has become a poorly guided task. This has led to the development of many drivers that perform probing using slightly different techniques [33]. These implementation differences along with additional firm- or hardware related differences, cause timing variations within the behavior of devices [27]. Most common timing-based approaches are listed below:

- **Channel Switching Time**: When a device is scanning for nearby Wi-Fi networks, it sends one or more probe requests on all available channels. This means that during a network scan, the Wi-Fi radio is constantly switching channels. Timing

variations related to the ratio of the time a device stays on a channel to the time it takes to switch a channel can be exploited for fingerprinting. Applied by *i.a.* [34].

- **Probe Request Timing:** Relies on information revealed not from the content, but from the timings of probe requests [15][27] such as:
 - *Inter Burst Intervals:* Measures the time between two bursts of Probe Requests as a feature. Applied by *i.a.* [30] and [19] .
 - *Burst Length Feature:* Uses the length of Probe Request Bursts by measuring the time interval between the first and the last frame of a burst. Applied by *i.a.* [30] and [19].
 - *Inter Frame Arrival Time (IFAT):* The IFAT measures the time between the arrival of each frame within the same burst of the same MAC address. Applied by *i.a.* [35], [32], [33] and [36].
- **Time of Flight:** This approach uses the time of flight of a packet to localize its sender via a tri- or multilateration in space. Applied by *i.a.* [25].

Geo-spatial-based De-Anonymization

Geo-spatial-based approaches attempt to localize the sender of a signal in space. Two common basic strategies are known:

- **Geo-spatial Fingerprinting:** By dividing the space at hand into small units and by fingerprinting each unit specifically, it is possible to match the received signal with the different fingerprints to localize a device in space. Applied by *i.a.* [37] and [19].
- **Direct Localization:** By using different receivers that cover the area, it is possible to localize a device in space by applying mathematical operations such as tri- or multilateration⁵.

Apart from the localization strategies themselves, two common measures of the signal received are RSSI and CSI:

The **Received Signal Strength Indicator (RSSI)** characterizes the attenuation of radio signals during propagation and has been adopted in a large body of localization systems. Although the RSSI can achieve meter-level localization accuracy in simple environments, "it suffers from dramatic performance degradation in complex situations due to multipath fading and temporal dynamics" [39]. RSSI values can be measured simply via the signals transmitted from wireless communication devices. Therefore, localization based on the RSSI will not increase complexity or hardware costs, which makes it the

⁵Multilateration is a method to determine the location of a movable vehicle or stationary point in space using multiple ranges between the vehicle/point and multiple spatially-separated known locations [38]

most common method for localization in wireless sensor networks [37]. Applied by *i.a.* [37] and [40].

Different from the RSSI, **Channel State Information (CSI)** is a powerful feature of the physical layer that can discriminate multipath characteristics. In a conceptual sense, channel response is to RSSI what a rainbow (color spectrum) is to a sunbeam, where components of different wavelengths are separated [39]. As Channel State Information (CSI) is able to provide more stable and more fine-grained signal information, it is the preferred choice for most approaches. Applied by *i.a.* [37] [25][41][39][42][43][44].

2.5.3 Existing systems for De-Randomization

- **Wombat**: The Wombat system by [15] is a network analysis tool created to raise awareness of the information leaked by network signals. It collects and extracts all possible data, such as place, MAC address, and visit duration and presents it to the user [15].
- **PARADIS**: The PARADIS system uses a timing-based approach to uniquely identify and differentiate between different CIDs. It uses a specially designed sensor, that can measure the wave modulation of wireless frames [24].
- **NiFi**: The NiFi system is a passive Wi-Fi-based user identification system relying on differences in the signal sequences induced by the user behavior [45].
- **nexmon**: Nexmon is not an actual system to circumvent de-randomization, but rather a tool to extract CSI values for each packet on the fly [46].
- **Wobly**: Wobly is a CSI based tool, that measures gait patterns and uses these to identify different people within the covered area [47].
- **PriLA**: PriLA is a location-based authentication system that verifies a users' location information based on carrier frequency offset data [48].

2.5.4 Discussion

It does not matter whether the final goal is classification, device or people inference, localization, de-anonymization, physical tracking or non-device based surveillance: All of the discussed approaches can be categorized into one or multiple of the following four possible weaknesses:

1. They rely on **specific data fields** that are not assumed to be stable over time or universally equal from device to device (could render useless after the next soft- or firmware upgrade). This is the case with all approaches that rely on data fields from MAC-layer frames, especially from IEs.

2. They rely on **non-commodity hard- or firmware**: While many approaches use CSI values that are not accessible from all Network Interface Card (NIC)s, others like physical-layer fingerprinting techniques, use extremely expensive hardware.
3. They pose **strong limitations** to their results regarding external factors or the number of devices that can be analyzed in parallel. This applied to most RSSI based approaches.
4. They rely on an idea, that is **possible in theory**, but impossible in real life due to an enormous amount of practical problems (*c.f.* time of flight)

And even though, physical component fingerprinting raises some kind of exception to this conclusion, due to its extremely high accuracy (*cf.* [24], [26] and [27]), it also relies on complex and expensive hardware not accessible to most people. For instance, [24] use an Agilent 89641S vector signal analyzer, which costs multiple times more than off-the-shelf 802.11 Wi-Fi Network Cards used by other fingerprinting techniques⁶.

Based on the analysis of the strengths and weaknesses of the approaches, the following conclusions can be drawn:

1. It is not possible to create a system, that circumvents MAC address randomization completely: A successful approach, therefore, takes into account, that its results are an approximation and its bearing limited.
2. A successful system should not rely on a specific field or piece of data produced by the device itself. Instead, it should use external information like signal strength (RSSI).
3. Instead of relying on one piece of information alone, a successful system combines multiple data sources to increase accuracy and stability.

⁶This is an enormous understatement: The follow-up model (according to [49]) in used condition costs on Ebay 106'000 Dollars [50]

Chapter 3

MacScavenger

3.1 Proposed De-Anonymization Process

The system proposed in this work uses a multi-step approach, that combines signal strength-based device localization with IEs. It, therefore, combines an external feature, with a feature, that is content-based and device-specific.

Instead of relying solely on probe request features, and struggle with the potentially small variance of IEs, the system tries to increase the informative value of the IE, by adding the location estimate as an extra source. Through this mechanism, it becomes possible to tell whether two probe requests are possibly originating from one or multiple devices and ultimately to count present devices. The necessary steps to make these decisions are described in Algorithm 1.

The primary goal of the proposed system is, therefore, to create a location estimate, that is as accurate as possible. Because the better the system can locate devices, the better it can distinguish devices based on their IEs.

3.1.1 Localization

To localize a device, the RSSI values received at the different APs can be used to estimate an distance in meter using the log distance path-loss model.

$$PL = PL_0 + 10 * \gamma * \log_{10} \frac{d}{d_0} + X_g \quad (3.1)$$

The basic equation of the log distance path-loss model sets the path-loss(PL), represented by the RSSI value, in relation to several other factors [51]:

- PL_0 : The path-loss at reference distance d_0
- γ : The path-loss exponent, a factor to incorporate different environmental factors

Algorithm 1: Decision making process of the proposed system

Result: Circumvent MAC address randomization based bias

```

while new packets are captured do
  if packet is captured by multiple APs then
    extract MAC address of sender and IEs;
    if database contains combination of MAC address and IEs then
      | The device is not doing MAC address randomization;
    else
      if database contains equal IEs then
        | Create location estimate;
        | Compare current location with last location;
        if locations close enough then
          | Devices are the same device doing MAC address randomization;
          | Add MAC address as alias to previous MAC address.
        else
          | Devices are not the same;
          | Add to database as new device;
        end
      else
        | Add data to database;
      end
    end
  else
    | Discard packet;
  end
end

```

- d : The length of the path
- d_0 : The reference length
- X_0 : A normal (or Gaussian) random variable with zero mean, reflecting the attenuation caused by path fading mechanisms.

Applying the Formula 3.1 to the measured RSSI values at the different receivers, results in an approximation of the distance the sender has to each AP. Combining these distance estimates with a mathematical approach called *multilateration*, the sender's position can be estimated. Multilateration combines the multiple distances between a device with unknown location and multiple spatially separated APs with known location [38] to estimate the location of the unknown device. The mathematical concept used in multilateration is expressed in Equation 3.2.

In the case of three receivers, the formula looks the following way:

$$(x_i - x_u)^2 + (y_i - y_u)^2 = r_i^2 \text{ for } i \text{ in } 1 \dots 3 \quad (3.2)$$

When the three equations are rearranged, they show a linear equation in (x_u, y_u) . Written in vector form the equation is:

$$2 \begin{bmatrix} x_3 - x_1 & y_3 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix} \begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} (r_1^2 - r_3^2) - (x_1^2 - x_3^2) - (y_1^2 - y_3^2) \\ (r_2^2 - r_3^2) - (x_2^2 - x_3^2) - (y_2^2 - y_3^2) \end{bmatrix} \quad (3.3)$$

The equation 3.3 can then be solved by (x_u, y_u) .

Often, however, multiple measurements exist for the same position and more than the minimum of three APs are receiving a signal from the same sender. The system profits from the excess of data by incorporating as much information as possible into the multi-lateration process. The emerging prediction error, resulting from an over-specified system of equations, is minimized by using a non-linear least-squares approximation. This solution allows to fit a set of m observations (more than three RSSI values) with a model that is non-linear in n unknown parameters [52]. In this process, all unknown parameters, including parameters from the path-loss formula, can be approximated.

3.1.2 Kalman Filtering

A Kalman Filter is an algorithm that calculates a joint probability distribution, combining measurements and a theoretical model on-the-fly and in real-time [53][54]. Kalman Filtering, therefore, allows to combine the measured data points with an expected behavior, such as, in the case of this work, a moving device.

By assuming the device to behave in a certain way (move at a certain speed) and therefore excluding irrational behavior (jump from point A to point B in no time), the location estimates can be corrected. Additionally, it can also be used to apply filtering mechanisms such as a moving average in order to reduce noise.

Both applications of the Kalman Filter - noise reduction and path smoothing - are helpful in the context of the proposed system and were thoroughly tested. However, in real-life (i) paths are not always available and (ii) the data is sparse. Therefore, neither filtering nor smoothing, were applied in the final version of the proposed system.

3.1.3 Naming

The system is named *MacScavenger*, which is a combination of two words: The word MAC refers to the fact that the system solves a problem introduced by MAC randomization. The word *scavenger*, meaning *hunter*, *scrounger* or even *sweeper*, refers to the fact that the issue at hand is neither easy nor trivial and all we are able to do is *hunt* for data, *scrounge* pieces of information and *sweep* the area for useful clues, that can later be combined into a useful pattern. From now on the system is referred to by its name, *MacScavenger*.

3.2 Discussion of Proposed Approach

Apart from the many advantages, that the proposed approach has compared with other ways to handle MAC address randomization, there exist a couple of disadvantages that are discussed briefly below:

- **Instability of RSSI:** One of the main reasons, most systems do not rely on the RSSI values (*c.f.* [25] [39] [55] [46] etc.) for localization is the fact, that it is error-prone and unstable (*c.f.* Subsection 2.5.2) . However, as the prime goal of the proposed system is not to localize, but to distinguish devices from each other, the problem becomes less relevant, and only in the worst case, this becomes an issue. In our case, it is the IE that identifies the device in the first place [22]. Localization is only used in case, the IE identification fails due to devices that are too similar and have the same IEs.
- **Probe request-based approach:** The proposed system relies mainly on probe requests. If no such frames are transmitted, the system is rendered useless. However, all systems, except the ones measuring pure physical data, rely on some type of MAC frame. Also, compared to other frame types, probe requests do not require association or interaction with the device and have a high transmission rate (*cf.* Subsection 2.3.6) compared to other frame types (*cf.* MLME in Subsection 2.5.2). Additionally, probe requests are transmitted in bursts, which increases the chance of a single packet being monitored by multiple receivers (*cf.* Figure 2.11).

It can be said, that compared to other proposed approaches and concerning the goals specified in 1.3, the proposed system is a valid attempt to handle the difficulties imposed by MAC randomization.

3.3 Requirements

Based on the requirements described in Section 1.3 the following list of requirements is derived:

- The MacScavenger system can distinguish two devices based on the probe requests they emit. Using the IEs, contained in the probe request frame body, the MAC address, and the RSSI values, the system can tell, whether two devices are the same or not.
- Using snapshots of measurements over time the MacScavenger system can categorize newly seen devices into *four* categories by combining knowledge from the current snapshot and the aggregation of past snapshots:
 1. **Already seen:** Devices that can be identified as known, based on their IEs and their location.

2. **Not seen before:** Devices that have yet unknown IEs.
 3. **Randomizing:** Devices that are randomizing their MAC addresses
 4. **Non-Randomizing:** Devices that are not applying MAC address randomization
- The MacScavenger system can localize every device it sees. Based on the entirety of the location estimates, it is possible to reenact the behavior of each identified device.
 - The MacScavenger system is wisely managing the accumulated data load by ensuring, data storage and minimal data loads. This ensures that the load posed on *e.g.* expensive mobile connections, is minimized, while at the same time a system failure will not result in total loss of data.

3.4 Assumptions

Based on the basic principles of the proposed system, a variety of assumptions are made for the system to be functional:

- **Device Density:** It is assumed that one device represents one person, which is usually the case for mobile phones (one person typically carries one mobile phone). Due to the fact, that the system is counting devices, people carrying multiple devices, break with these assumptions. In that case, it is no longer possible to infer the number of people present. It is further assumed that all devices are ideally in a non-associated state, producing a high rate of probe requests.
- **Well covered area:** The system needs to be evenly distributed in the area that is under surveillance. If the monitoring devices can not cover the area completely or have too much overlapping space, the localization fails.
- **Gait Pattern:** It is assumed, that people carrying devices are moving around in space at a relatively slow pace. Even though the MacScavenger system can be configured to different walking speeds, the localization approach fails if the position of the devices changes quicker than the monitor nodes receive new probe requests.
- **Disturbances:** It is assumed that no interference with any other system exists. This includes any system, that produces probe requests, modifies existing packets or that does any sort of radio wave disturbance.

3.5 Design Considerations

The MacScavenger system is using a *sync* device to aggregate and evaluate the data gathered by the *monitor* nodes. This sync is also responsible for reducing data volume and safely store the data. Even though the sync creates a slightly more complex system topology it has several advantages:

- It allows us to minimize the amount of data, that is stored or sent over possibly expensive connections to the cloud by aggregating the data on-the-fly and by discarding non-relevant information.
- By taking over most of the computationally intensive operations, the requirements on hard- and software posed on the monitor nodes are reduced to a minimum.

The MacScavenger system consists of two independent logical components:

- **Data Gathering:** This component is responsible to coordinate the data gathering process as well as the aggregation and storage of all relevant data.
- **Data Analysis:** This component is responsible to read in the data and interpret it based the relevant features.

Additionally, a specially designed *shell* allows the user to interact with the application and the functionality provided by the two components. The user can start, stop and observe the monitoring process or trigger the analysis using different parameters

Furthermore, all components of the MacScavenger system are set up as a streaming application, containing a pipeline. This pipeline consists of different I/O, aggregation, and analysis operations. This setup allows easy configuration and adaption of the MacScavenger systems behavior by "plugging" in and out different operators to the pipeline.

3.6 System Topology

3.6.1 Shell

The entry point of the MacScavenger is the shell. It provides an interface for interacting with both the data gathering and the data analysis component. Both components are completely separated and do not share any functionality. This independence ensures the complete interoperability of the MacScavenger system in case of external data.

3.6.2 Data Gathering Component

The data gathering component is based on a client-server architecture, with the sync as client and monitors nodes as servers. The client connects to the servers, starts different processes, and fetches data. The shell only interacts with the sync and outsources all interaction with the monitor nodes to the sync node.

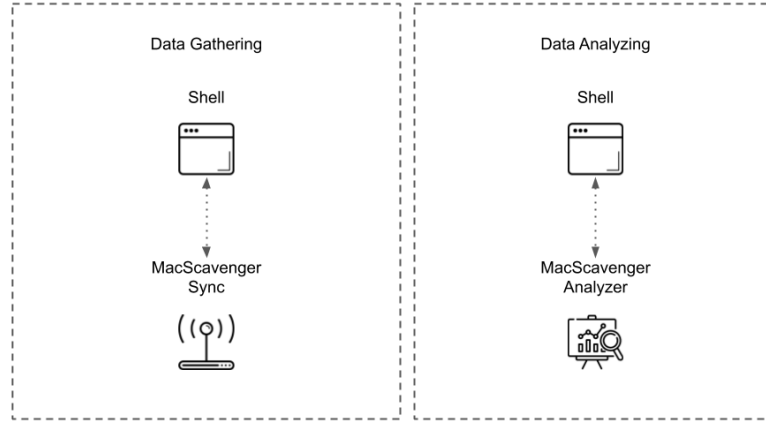


Figure 3.1: The Interaction of the Shell with the data gathering and the data analysis components

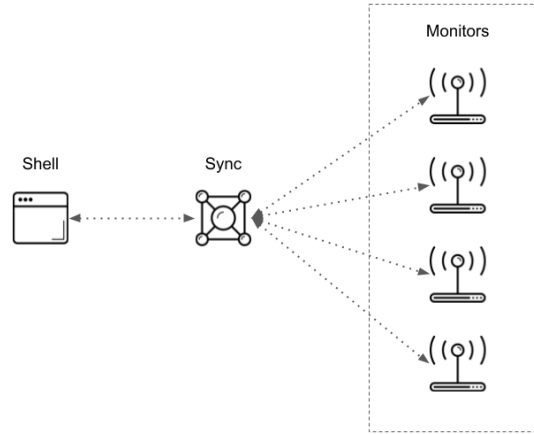


Figure 3.2: The topology of the MacScavenger system regarding the Data Gathering Component

3.6.3 Data Analysis Component

The data analysis component is using a specially designed pipeline that gets the raw data as input and pipes it through a set of operators to extract knowledge. To involve the data analysis component the shell delegates the analysis to the analysis component. In this case, no part of the data gathering component, such as the sync, is used.

3.7 Implementation

This section discusses the components described in the previous section and their functionality in more detail, also regarding the used soft- and required hardware¹.

¹The MacScavenger system does not include any hardware components itself and is fully operational on any system that runs python3. However, due to the necessity to monitor network traffic, several

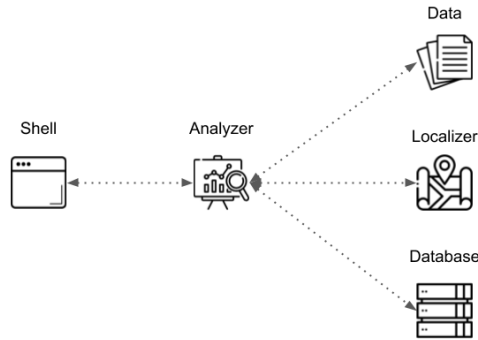


Figure 3.3: The topology of the MacScavenger system regarding the Data Analysis Component

3.7.1 Languages and Frameworks

This subsection describes all implementation relevant details, that apply to all components of the MacScavenger system

Python

The MacScavenger system is written in *python*. Python is a "high-level general-purpose programming language" [56]. It has multiple paradigms and allows object-oriented, functional-oriented, and aspect-oriented programming. Furthermore, python data types are strongly and dynamically typed [57]. Python strives for a simple, easy to read syntax and grammar and embraces a "there should be one—and preferably only one—obvious way to do it" design philosophy [58].

The actual strength of python lies in its extensibility, which allows easy use of external packages, that add programmable interfaces to existing applications[57]. It was chosen as the basic language for the MacScavenger due to its easy readability, its simple syntax, and its enormous worldwide open-source community.

For python, version 3 was chosen, not only because it includes many built-in features such as simple interfaces for asynchronous programming, but also because python 2 has lost support by January 1 2020 [59].

Streaming

All components of the MacScavenger system follow a strict streaming approach. Streams are possibly infinite sequential data structures, that compute their elements only on-demand [60]. This approach allows us to handle a potentially infinite amount of data, such

requirements are posed towards external hardware components such as the NIC of the used devices.

data resulting from a monitoring process. As the basic data used by the MacScavenger system is such a stream, each component of the MacScavenger system consists of such a pipeline to deal with the data.

MacScavenger uses the Python package "streamz", which offers lightweight support for building pipelines to "manage continuous streams of data" [61].

Sockets and Socket Servers

A socket is the combination of an Internet Protocol (IP) address and a port number [62]. In a more narrow sense, a socket is an endpoint of a two-way communication link between two programs running on the network [63].

The python *socketserver* framework builds upon these sockets and offers both TCP- as well as User Datagram Protocol (UDP) servers. The MacScavenger uses multithreaded TCP socket servers listening on a specific port [64]. Whenever the server is connected to a device (MacScavenger sync), the server (MacScavenger monitor) will open a thread for each open connection.

MongoDB

MongoDB is a cross-platform NoSQL document-based database, that uses JSON-like documents with optional schemas. It offers flexible ways to store information in documents, ad-hoc queries, indexing, and various ways of aggregating data [65]. Each database is organized in collections that store different documents.

MongoDB is used in the MacScavenger system to store and complement already seen combinations of IEs and MAC addresses with their location estimate.

Python Command Line Module

The Python Command-Line Module *cmd* is a simple framework for writing line-oriented command interpreters [66]. It offers multiple possibilities to create a command-line interface to interact with any sort of python code.

cmd was used to create the MacScavenger shell. A command-line interface to interact with the MacScavenger system and its components.

3.7.2 System Components

Gathering Component

The gathering component consists of two types of nodes: A MacScavenger sync node and multiple MacScavenger monitor nodes. Both need to interact with each other to gather valid data.

MacScavenger Sync The MacScavenger sync node is responsible for coordinating the data gathering process by interacting with the user on one side and with the monitor nodes on the other side.

The management of all the monitor nodes is achieved through a list, that contains all necessary information about every monitor node: MAC address, port, degree of activity, and transmitted data. Using this data the sync node is then able to establish a connection to each MacScavenger monitor node through a TCP streaming socket.

Once the data gathering process is started the MacScavenger sync receives data from all monitor nodes in parallel. The received data is then piped directly into the sync nodes streaming pipeline, which applies several operators: One is responsible for the reassembly of the data received over the TCP socket and the other one for RSSI noise reduction. In the end, the pipeline stores data in a database in json format. The complete process is shown in Figure 3.4.

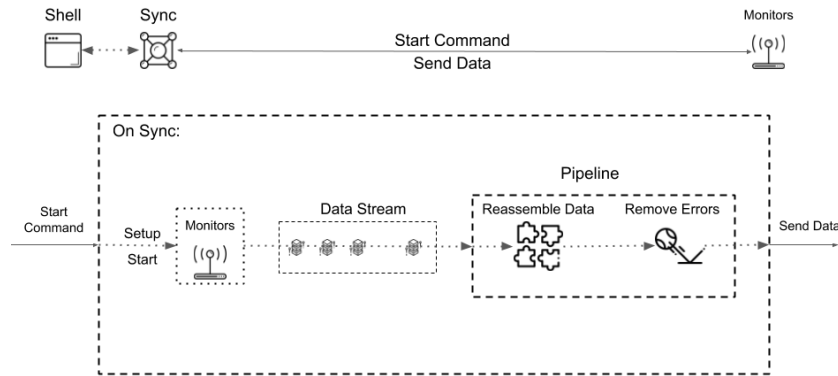


Figure 3.4: The Data Gathering Pipeline on MacScavenger Sync node

MacScavenger Monitor Each MacScavenger monitor node is set up as a TCP socket-server, containing a stream pipeline configured to transform and transmit the data back to the sync node as shown in Figure 3.5. After it starts up, it awaits one of three commands triggered by the MacScavenger sync:

1. **Setup:** This command will check the host system for available network cards and sets them up in monitor mode.
2. **Start Monitoring:** This command starts the packet dumping process. The main work hereby is executed by *tshark*, a network protocol analyzer[67], that dumps all probe requests received at the specified NIC. The monitor node pipeline extracts the relevant data fields such as MAC address, timestamps, RSSI values and IEs, and sends them back to the MacScavenger sync. In parallel, the NIC is forced to change its listening channel every 0.25 seconds to monitor all available channels.

3. **Stop Monitoring:** This command stops the channel switching and the packet dumping, setting the monitor node back to its listening state. In this state, it awaits again one of the three commands.

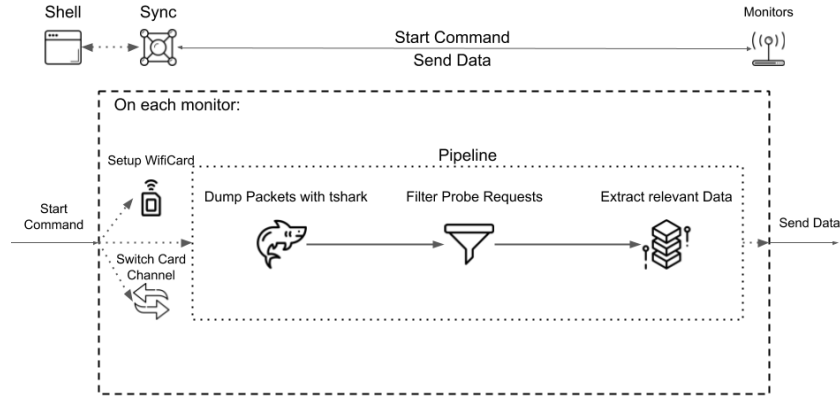


Figure 3.5: The Data Gathering Pipeline on MacScavenger Monitor nodes

Analysis Component

The MacScavenger analysis component comes into play once a valid data source exists. The core of the analysis component is a stream pipeline, consisting of multiple steps. Each step transforms, interprets, or stores the data as shown in Figure 3.6. Most data transformations are executed using pandas DataFrames².

The multiple steps are shown in the following list:

1. **Loading Data:** The data is loaded from a valid data source. Either the data is produced by the MacScavenger itself or it is external data, that has been formatted in the right way.
2. **Parsing Data:** The timestamps of each received probe request are parsed and formatted in the right way.
3. **Splitting Up Data:** As the data, itself is not known beforehand, the received probe requests rate can be very high (many packets in a short interval). Therefore, the data is split up into small consecutive windows of a specified length.
4. **Intersection of Data:** All packets that were received by less than a specified amount of monitor nodes are discarded. All other packets are aggregated, such that for each probe request, the RSSI values, measured at different monitor nodes, are stored.

²DataFrames are a 2-dimensional labeled data structure with columns of different types. They allow vectorized data transformations and aggregations to increase speed [68].

5. **Localizing Data:** Based on the different RSSI values, each packet is localized using Multilateration (*c.f.* Subsection 3.1.1).
6. **Interpreting Data:** By checking the database for similar or equal occurrences, the MacScavenger analyzer can tell, whether the origin device is applying MAC address randomization or not, and how many devices are present in the monitored area.
7. **Store Data:** All knowledge extracted in the previous step is added to the database.

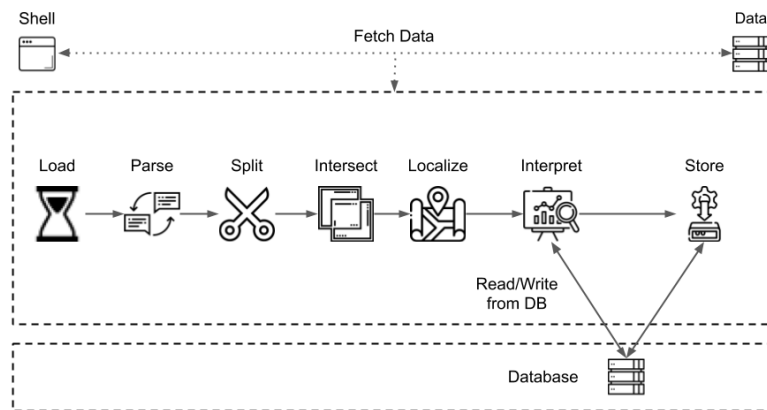


Figure 3.6: The Data Analysis Pipeline

MacScavenger Shell

Via the shell the user accesses all of MacScavenger functions. The shell is built using the Python *cmd* framework (*c.f.* Subsection 3.7.1), which includes automatic built-in standard behavior, such as *help* functions. The MacScavenger Shell offers the following commands:

- **load:** The load command is used to load a configuration file, that includes a list of configuration data for monitor nodes such as name, IP address, and port. The load command allows the user to add a list of pre-configured monitor nodes in an easy and fast way. The configuration file can be created manually by writing the data in the correct form into a file or by using the *save* command. The load command can be used with or without a parameter: If the list needs to be written into a configuration file with a specific name, this can be done by specifying a name. If no name is specified, the list is written to the default configuration file, overwriting previous content.
- **setup:** The setup command is used to trigger the sync node to initialize the network cards on all known monitor nodes.
- **save:** The save command writes a list of configured monitor nodes to the configuration file for later use by the *load* command. If no specific configuration is specified

the save command will load the default configuration, if a name is given and a valid configuration file is known under that name, this configuration is loaded.

- **ls:** The ls command shadows the Unix command Listing (ls). It shows detailed information on the loaded devices, such as their degree of activity, the amount of data received from the device, whether they are dead or alive and whether their network card has been initialized to work in monitor mode.
- **start:** The start command initializes and begins the data gathering process. The chain of events including the sync node and all monitor nodes is described in more detail in Figures 3.5 and 3.4.
- **stop:** The stop command stops the data gathering process described in Subsection 3.7.2.
- **add:** The add command allows the user to add devices on the fly in case a new device needs to be used for monitoring, which is not yet part of any configuration file. The add command expects three additional parameters *name*, *IP address* and *port*.
- **rm:** The rm (remove) command allows the deletion of devices from the list of monitoring devices. The command requires the devices *name* as a parameter.
- **clear:** The clear command removes all monitoring devices from the list.
- **analyze:** The analysis command initializes the data analysis process. The command leads the user through the process of configuration, where the user can enter various parameters such as the *data path*, *monitor node information* or the *assumed walking speed* of the people. After finishing the analysis it outputs a summary of the analysis in the shell.
- **exit:** Shuts down the MacScavenger shell.
- **help:** The help command prints out a list of all possible commands. Using the help command with one of the possible commands as a parameter outputs the use of the specified command.

3.7.3 Main Processes

The following section will outline the core processes of the MacScavenger, including the interactions between the components involved. Each process will be outlined in a sequence diagram.

Basic Shell Operations

When the user starts the MacScavenger shell, it loads its previous commands and prompts a welcome message. The user is now able to enter the various commands mentioned in Section 3.7.2. At the same time, the MacScavenger sync is initialized.

If a configuration file is loaded the shell first discards all previously-stored devices and then parses the YAML-based configuration file. It accesses the list of devices, including their parameters and credentials, such as name, IP address, and port number, and forwards it to the sync component, which in turn stores all those devices in its internal list. As soon as this process is done, the shell prints a short overview of the loaded devices to the user.

If at any time an overview is requested by the user, a loop is triggered: The shell requests the latest information about the currently known monitors from the MacScavenger sync in short time intervals, presenting to the user a self-updating list on the command line, that adapts to all changes occurring in the background. By pressing *ctrl+c* the loop is exited and the normal command-line interface is available again.

Calling the exit command writes all commands used in this session into the history file and exits the process. In the next session, these commands can be accessed by pressing the *up* or *down* keys on the keyboard.

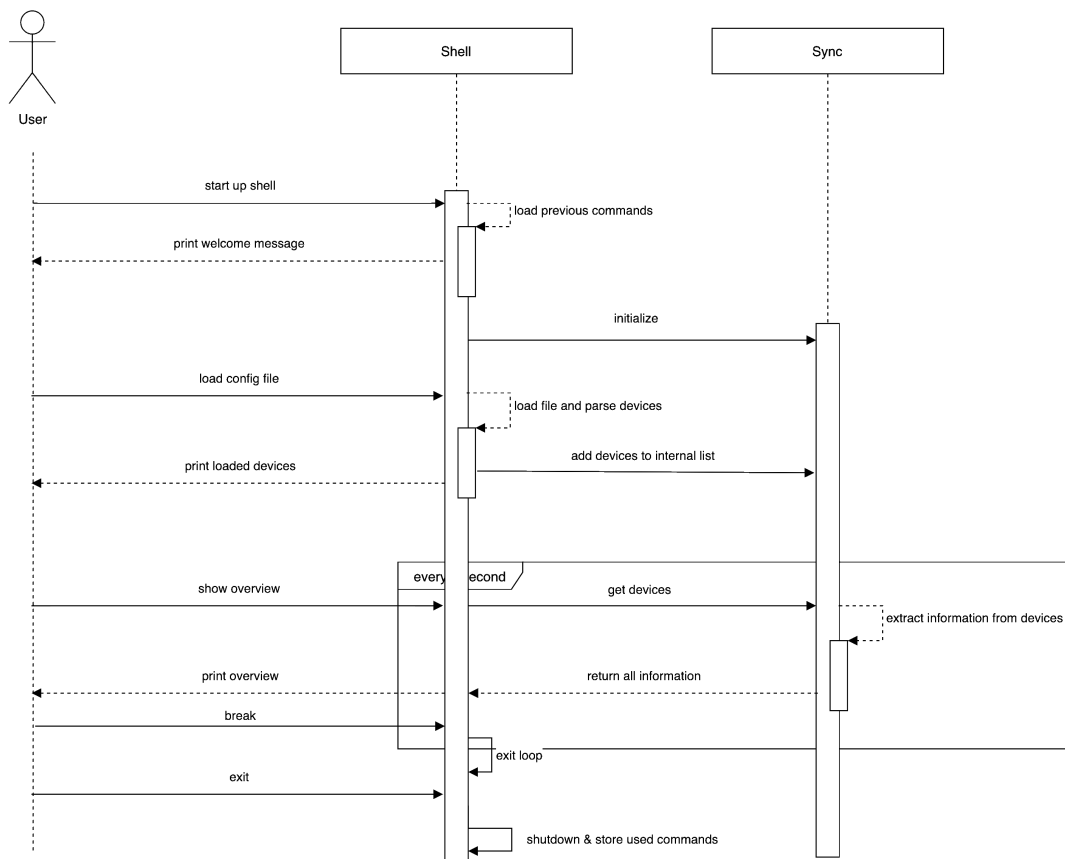


Figure 3.7: Sequence Diagram of the Basic Shell Operations

Setup Monitors

After the user has added different devices, either from a configuration file or manually via the command line, the devices need to be checked and configured to be ready to run the operations used in the data gathering procedure.

As soon as the shell receives the setup command, it delegates it on to the MacScavenger sync. The sync then iterates over currently known devices and attempts to connect with each device. As soon as a connection is established, the sync sends the setup command to the monitor node.

Each node receiving the setup command then attempts to set its NIC to monitor mode. Whether this operation is successful or not is then communicated back to the sync.

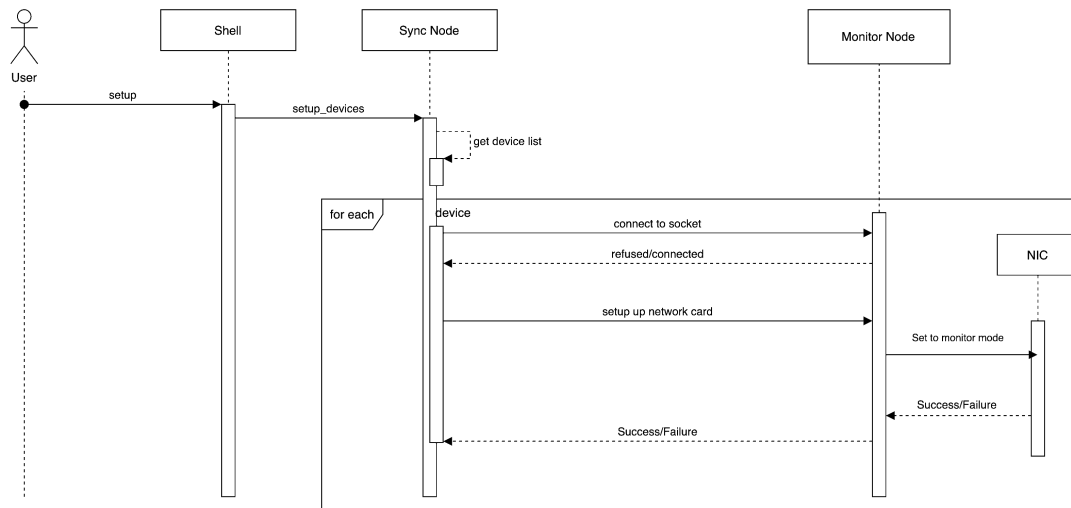


Figure 3.8: Sequence Diagram of the Initialization of the Monitor Nodes NICs

Data Gathering Process

The data gathering process is kicked off by the *start* command received by the shell. The shell immediately delegates this command to the sync, which iterates over its internal list of devices and tells each device to monitor the network traffic. If the connection attempt fails, the sync will retry after a short timeout until the connection can be established.

As soon as a monitor node receives the command to monitor, it starts a special routine, which forces the NIC to switch the network channel every 250ms. Meanwhile, the actual packet dumping is delegated to *tshark*, which dumps and filters all probe requests. The monitor then directly transmits all data back to the sync in a streaming-like manner.

The sync unites all incoming data streams from the different monitor nodes and stores the data in the specified form. This can be any valid database, implementing the necessary interface.

Data Analysis Process

The data analysis process is started via the shell. In an interactive process, the shell first asks the user to specify a data path and then requests a set of parameters to be specified. While a set of default parameters are available, it is possible to specify every parameter

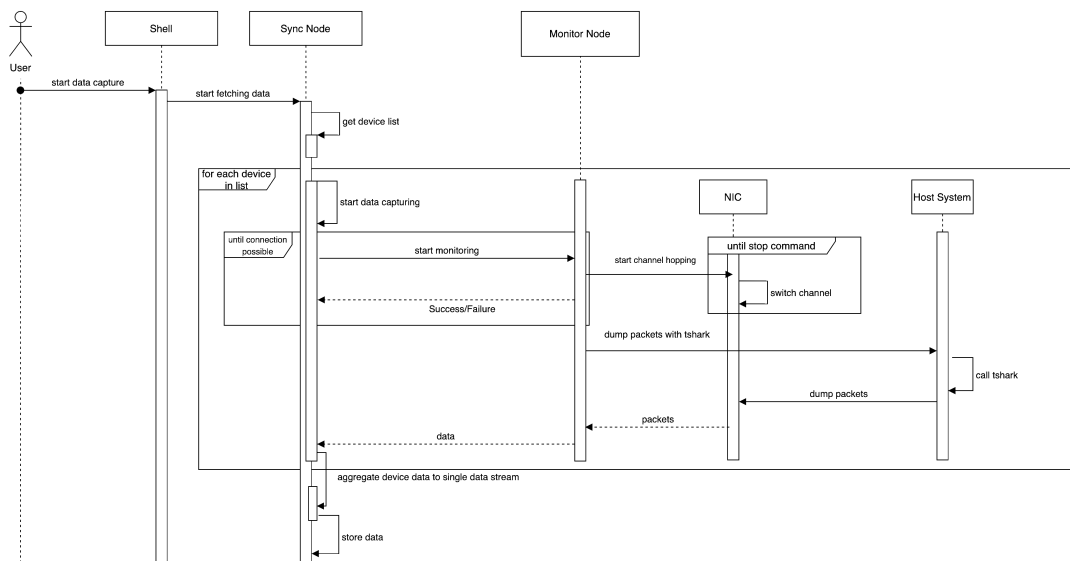


Figure 3.9: Sequence Diagram of the Data Gathering Process

used in the analysis process. Available parameters are: The *interval size*, the data stream is split into, the *assumed walking speed* of the people, an *in-burst threshold*, the *minimum amount of APs* a packet needs to be captured by and the *verbosity* of the analysis process. Once all parameters are known, the shell requests an analysis of the specified data with the user-defined parameters from the data analyzer.

The data analyzer, pipes the data as a stream through its pipeline consisting of a set of operators. First, the data is loaded and parsed, then it is split into equally sized time windows, is filtered and aggregated as described in Section 3.7.2.

Each packet is then sent to a special *localizer* component, that looks at the measured RSSI values from all monitor nodes and attempts a location estimate. This estimate is then used by the data analyzer to categorize the packet: By querying the database, it can decide whether this packet has an already known source or not. All insights acquired in this step are again written to the database.

After all, data has been analyzed, the shell requests a summary from the data analyzer component. After querying the database it passes an aggregation of all knowledge it gathered analyzing the data, back to the Shell, that in turn prints it to the user onto the command line.

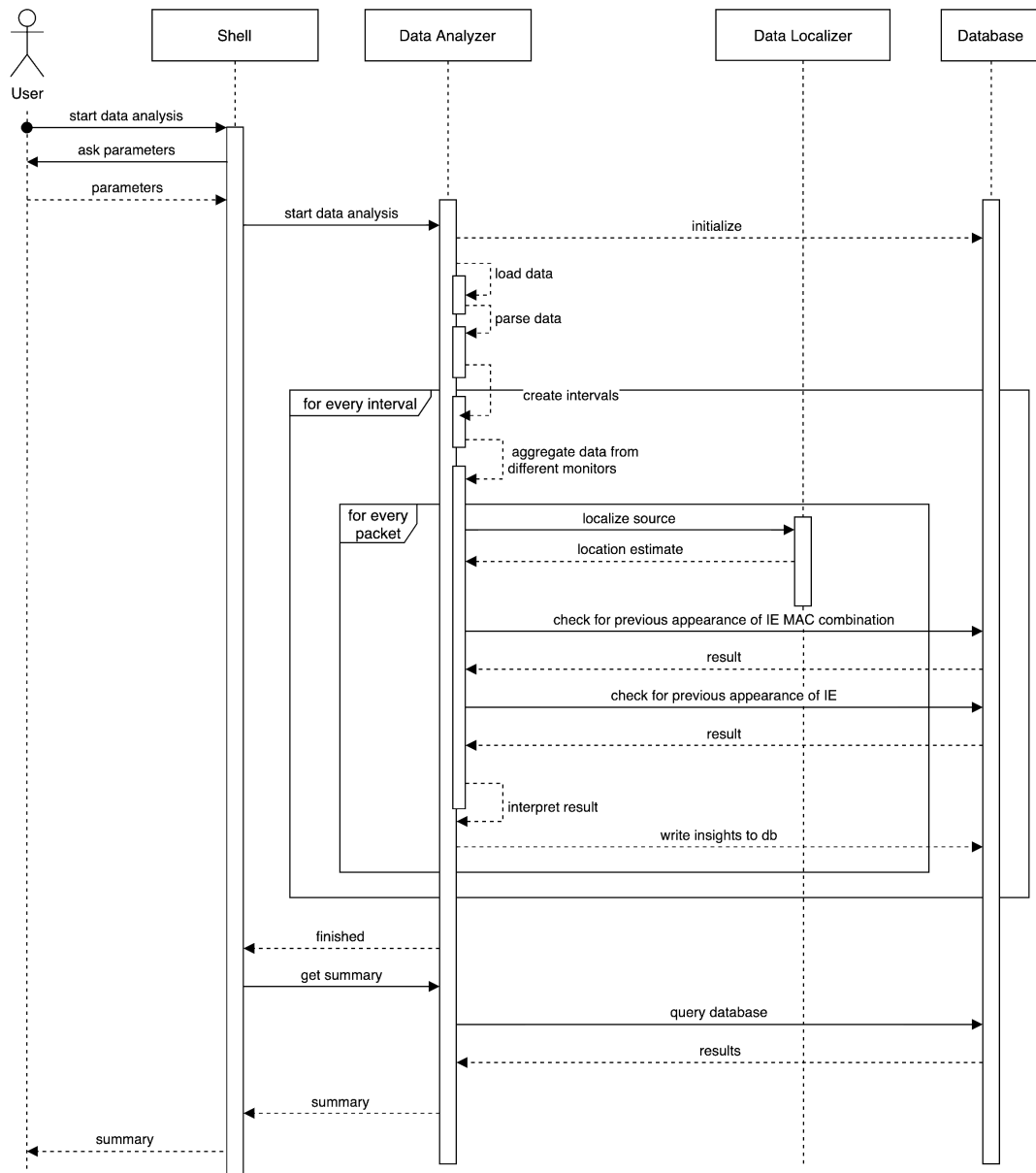


Figure 3.10: Sequence Diagram of the Data Analysis Process

Chapter 4

Evaluation

This chapter describes three experiments, which were conducted to (i) test the localization-based approach of MAC address de-anonymization and to (ii) test whether the proposed system is working as expected according to the goals in Section 1.3. As the experiments focus on different aspects, they differ in their exact setups and specific goals.

4.1 Experiment 1: Standalone Localization

4.1.1 Basic Idea

Experiment 1 was conducted as the first attempt to localize a device based on the RSSI values it emits. By measuring the RSSI values of a designated sender on four different monitor nodes, it was the goal of this experiment to locate the device at different points in space within the covered area. This goal is congruent with goal number 2 specified in Section 1.3.

4.1.2 Methodology and Setup

The experiment was conducted on the 20th of April in Winterthur, ZH on an open field space with good weather and temperatures around 20 degrees Celsius [69]. Four monitor devices were placed at the corners of an 8 by 5-meter grid, while a sender was moving one meter at a time, halting one minute per grid point.

The sender was a *Raspberry Pi 3* with *Alfa-AWUS036NHA Wifi-Adapter*. By using packet injection and the python package *scapy*, artificial probe requests were generated and transmitted at a high rate. Each probe request contained a global sequence number and a burst number for later identification. The monitors were four *Spitz GL-Routers*. All components were connected via cables and a switch.

Starting with the first grid point, the sender started to send probe requests for one minute. At the same time, each monitor was doing a packet dump using the *tshark* framework. After each minute, the sender was moved forward to the next grid point in the direction of the red arrow visible in Figure 4.1 until all 48 grid points were reached. The four corner positions were not covered by the sender as they were already blocked by the monitors themselves.

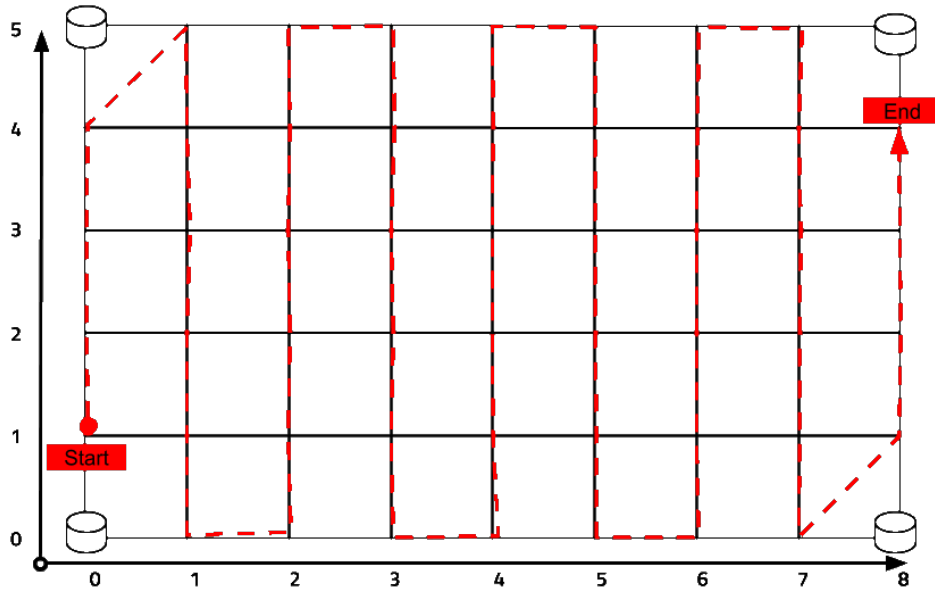


Figure 4.1: Methodology of Experiment 1: At each point in the grid, the sending device was transmitting probe requests that could be captured by the four monitoring devices in the corners.

As the sender was moving on a specified path, a Kalman Filter was used to smooth out the location estimates.

4.1.3 Results

The experiment is evaluated by measuring the deviation of the estimated position of the sender from the true position of the sender. At each point on the grid, this deviation is calculated and, finally, averaged overall steps.

The overall error in meters was 87 meters without Kalman Filtering, 58 meters applying Kalman Filter based smoothing. This results in a per step error of 1.7 meters or 1.1 meters with the Kalman Filter. Including the maximal possible distance in meters (10.8), this results in an error percentage of 0.15 percent (0.1 with Kalman Filter). The deviation of the unfiltered location estimates ranges from 0.03 to 5 meters.

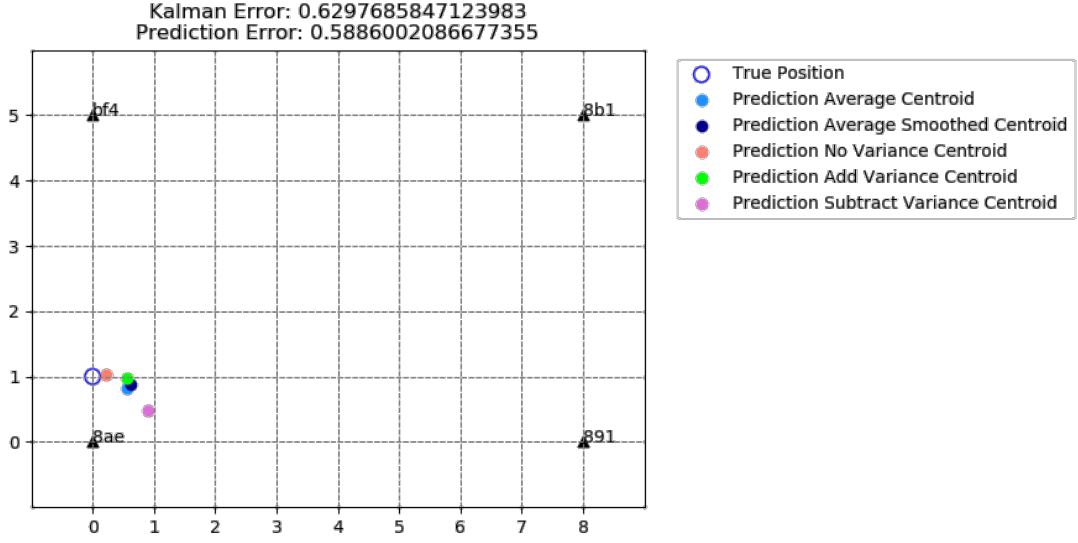


Figure 4.2: Result of Experiment 1 on position (0,1): The true position is shown with the empty big blue circle. The different estimates of the unfiltered (light blue), Kalman Filter based (dark blue) and variance based approaches (orange, green, violet) are shown as filled dots.

4.1.4 Conclusion

The RSSI-based localization of a transmitting device using probe requests was successful, as the average deviation was very low (around 1 meter per position using the Kalman Filter).

4.2 Experiment 2: Joint Localization

4.2.1 Basic Idea

Experiment 2 was a joint experiment: It included RSSI measurements from both Wi-Fi probe requests as well as *Bluetooth*-based measurements. The goal of this experiment was to compare and combine, the location estimates of both types of frames and to gain insights, whether a combined approach is fruitful or not. The experiment was conducted similar to experiment 1 but with an additional Bluetooth device in place, fewer measurement points, and different hardware. Similar to experiment1, experiment 2 aimed also at fulfilling the goal number one, specified in Section 1.3.

4.2.2 Methodology and Setup

The experiment was conducted on the 25th of June, 2020 in Zürich, ZH in an urban environment on a concrete balcony. The weather was hot and dry and the temperatures were around 27 degrees Celsius [70].

Similar to experiment 1, a sender was moving through the grid of a specified area in the same way as shown in Figure 4.1. However, in experiment 2, the area was only 5 by 5 meters wide, resulting in only 32 measurement points.

The sender was the same device used in experiment 1. Also, the same hardware and software hooks were used for the probe request creation and injection. However, as monitoring devices four *ASUS Tinkerboards* were used. Each of them was equipped with an external network card for packet monitoring (*Ralink RT5572N USB dongle*) and additional hardware for Bluetooth packet capturing (*Ubertooth One*). The packet dumping of the Probe Requests was again conducted using *tshark*.

4.2.3 Results

The experiment was evaluated in the same way experiment 1 was evaluated. Experiment 2 showed less accurate location estimates, for both Bluetooth and Wi-Fi-based localization.

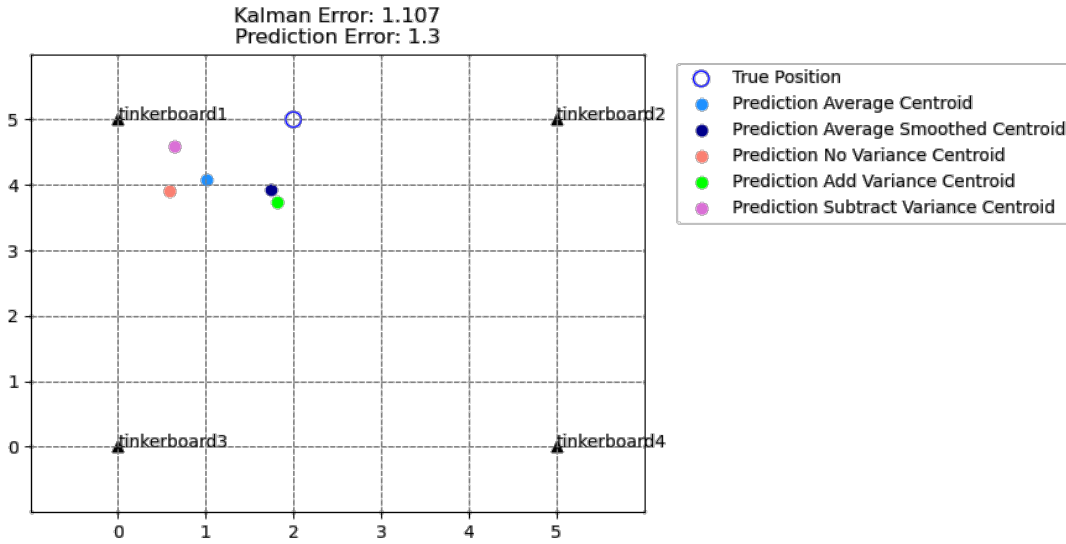


Figure 4.3: Result of Experiment 2 at position (2,5). For legend *c.f.* Figure 4.2.

The overall error in meters for the Wi-Fi-based localization was 72 meters without Kalman Filtering, 68 meters applying Kalman Filter based smoothing. This results in a per step error of 2.2 meters or 2.1 meters with the Kalman Filter. Including the maximal possible distance in meters (8.4), this results in an error percentage of 0.26 percent (0.25 percent with Kalman Filter). The deviation of the unfiltered location estimates ranges from 0.44 to 4.5 meters. See Figure 4.3

The overall error in meters for the Bluetooth based localization was 68 meters. This results in a per step error of 2.1 meters. The deviation of the unfiltered location estimates ranges from 0.8 to 3.8 meters. Including the maximal possible distance in meters (8.4), this results in an error percentage of 0.25 percent. Due to the limited time (60 seconds) used for the measurement, not enough data was collected to apply the Kalman smoothing. The Bluetooth-based localization is described in more detail in [71].

4.2.4 Conclusion

Compared with the experiment 1, experiment 2 produced slightly less accurate results. This fact can not only be seen in the higher per step error but also the fact, that the Kalman Filtering did only have a minor impact: The error does not source primarily in the variance of the different types of estimates (they all tend to be quite close together), instead, the location estimates are in most cases completely off in their approximation. This can be seen well in Figure 4.4.

Possible reasons for the lower overall accuracy could be the environment (concrete floor), the heat (27 degrees Celsius) and the different hardware used (ASUS Tinkerboard with Ralink RT5572N wifi adapter). Especially, the environmental aspects are known factors for changing the measured RSSI values [72].

Additionally, it is possible and based on the visuals in Figures 4.4 and 4.3 an educated guess, that the monitor on the bottom left corner was producing questionable measurements. This behavior could have been provoked by the direct sun light exposure of this monitor compared to all other monitors [72].

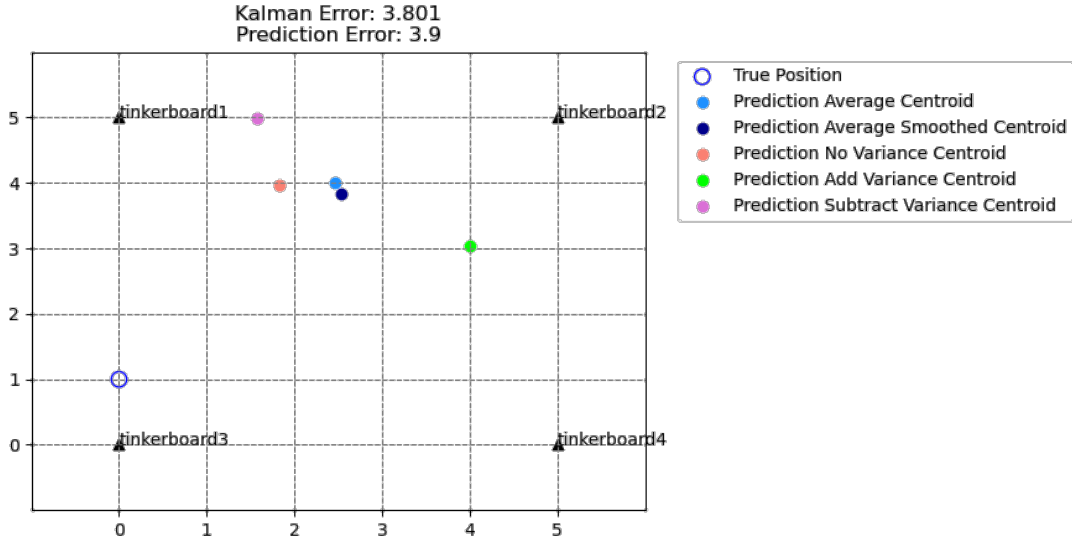


Figure 4.4: Result of step 1 of Experiment 2: the different variations of the location estimate are close together, but very far apart from the true location of the sender. For legend *c.f.* Figure 4.2.

It is not possible to tell, whether the different behavior of experiment 1 in comparison to experiment 2 comes from the location (field vs. concrete) or the different hardware. However, the success of the first experiment could only be repeated partially: First, the Bluetooth-based approach had hardware-based failures, and secondly, the Wi-Fi-based approach had a higher error rate than in the previous experiment. Therefore, based on the results of experiment 2, a combined approach of Wi-Fi- and Bluetooth-based localization is unlikely to improve the results.

4.3 Experiment 3: MacScavenger at Messe Basel

4.3.1 Basic Idea

The basic idea of experiment 3 was to apply the approach of localization-based MAC address de-anonymization to real-life data. Therefore, experiment 3 differed in various aspects from the previous two experiments:

1. The experiment was not conducted under ideal conditions but rather during the real-life event "Reloading Live - So machen wir Veranstaltungen sicher – ein realer Showroom"¹ at the Messe Basel on June 2nd, 2020.
2. For the data gathering and the data analysis processes, the MacScavenger was used.
3. The goal was to do a device/people count instead of verifying the location estimates. Therefore, the effectiveness of using the combined approach of IEs and location estimate was evaluated using the system proposed in this work

Experiment 3, therefore, was the ultimate test for the usefulness of the MacScavenger system, congruent with goals two and three (*c.f.* Section 1.3). However due to practical reasons, no sync could be placed. Therefore, the monitor node code was slightly adapted to start and stop at specified times and to store all data locally. The core processes of both gathering and analysis components were left untouched.

4.3.2 Methodology and Setup

The Livealitics stand at the "Reloading Live" event was placed as shown in Figure 4.5 and covered an area of 4 by 4 meters. At four opposed points at the border of this area, four MacScavenger monitor nodes were placed at a height of around two meters above the floor. The same devices as in experiment 2, including all hard- and software components, were used as monitor nodes.

The monitoring process began at 11:00 AM and was automatically stopped at 14:00. However, due to unnoticed cabling issues, one of the four access points had no power and could, therefore, not monitor.

4.3.3 Results

Since the MacScavenger can run analyses using numerous parameters and different thresholds specified by the user, varying results can be obtained. For the analysis of experiment 3, the following parameters were chosen:

¹In English: "Reloading Live - Ways to make your events more secure - A real-life showroom"

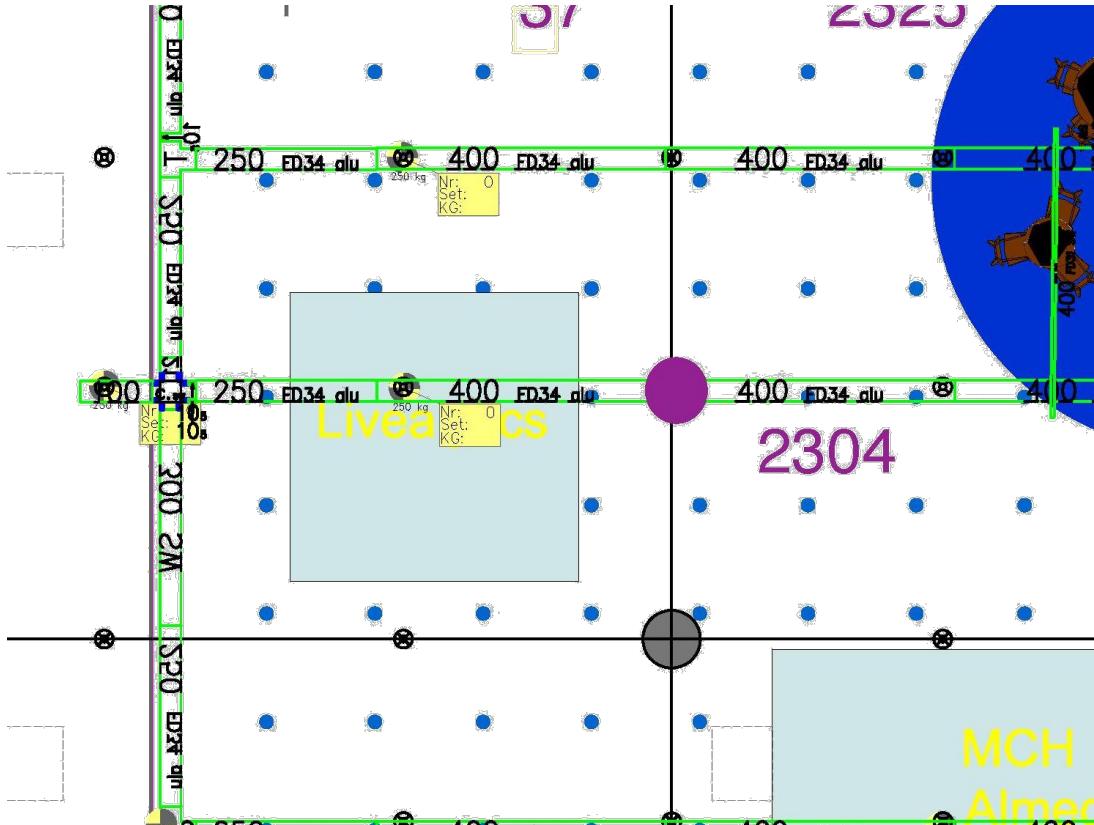


Figure 4.5: A map of the lifelytics stand at the "Reloading Live" event

- Interval Size: 10s
- Assumed Walking Speed: 2km/h
- In Burst Threshold: 1s
- Minimum AP detection rate: 3

The MacScavenger detected approximately 425 different devices on site. 370 devices were seen multiple times, while 55 appeared just once. Additionally, of all detected devices, the MacScavenger was able to identify 301 devices applying MAC randomization and 69 devices that were not. Overall, 708 different MAC addresses were captured.

4.3.4 Conclusion

The results obtained by the MacScavenger system in experiment 3 are very hard to evaluate because there is no actual control data. However, two external data sources are available:

- **Ticketing System:** The ticketing system counts the number of people entering and the number of people leaving the site.

- **Ceiling Camera:** The ceiling camera keeps track of the number of people entering and leaving the area of the Livealytics stand.

Both external evaluation sources, however, had a slightly different focus than the MacScavenger: The ticketing system was counting all people entering and leaving the building and therefore did keep track of the overall event. The ceiling camera was limited to the exact borders of the stand and did not include any person that stood outside of this area. The MacScavenger, however, was neither able to track the overall event as it only covered a small area nor was it limited to the stand only, as it captures probe requests also from outside the borders of the Livealytics stand. A simple evaluation of results obtained by the MacScavenger system using data from the ticketing system or the ceiling camera is thus difficult.

Nevertheless, under the assumption that all people present on site during the time of monitoring, at least once visited the Livealytics² stand or came close enough to be monitored, a comparison of the MacScavenger results and the ticketing information can be made.

By looking at the amount of first time entrances in Figures 4.6 and 4.7, an overall count of first entrances of 566 people can be made. This corresponds to the number of total visitors that entered the site for the whole day. This count, however, includes not only visitors but also all staff and the exhibitors. The number of visitors during the whole day was 360, as can be seen in Figure 4.7.

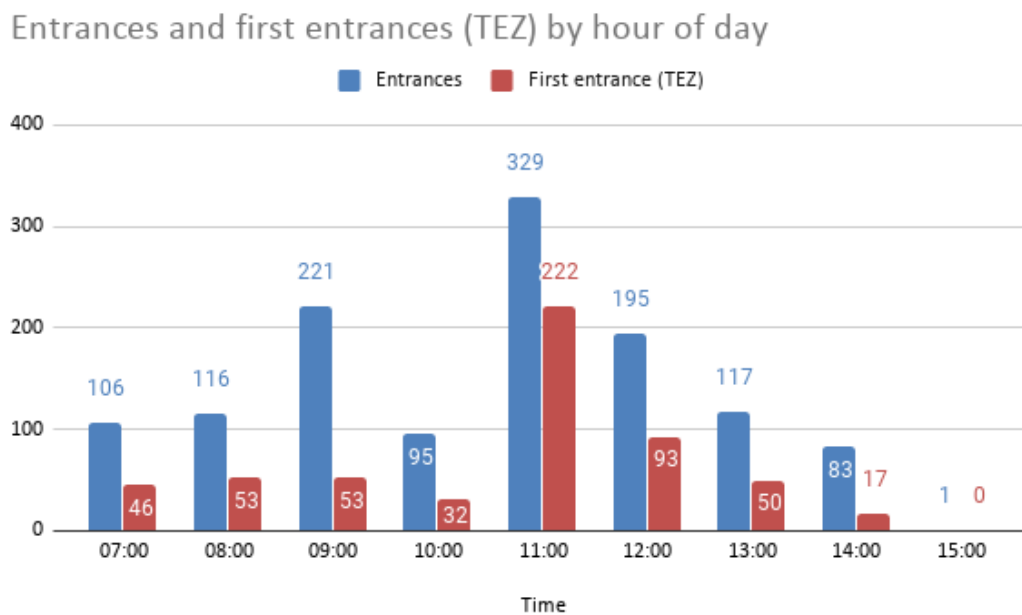


Figure 4.6: The amount of first time entrances over time

Both retrieved validation numbers - the number of total people (566) and the number of all visitors (360) - fall short to measure the number of people that passed by the Livealytics stand due to the following reasons:

²Given the fact that an event like the "Reloading Live" is visited by people to check out the different stands, this assumption is very likely.

Time	Aussteller	Besucher	Presse/Media	Entrances	Aussteller	Besucher	Presse/Media	First entrance (TEZ)
07:00	105	1		106	45	1		46
08:00	115		1	116	52		1	53
09:00	208	2	11	221	41	2	10	53
10:00	85	6	4	95	23	6	3	32
11:00	93	227	9	329	10	211	1	222
12:00	77	118		195	11	82		93
13:00	67	50		117	3	47		50
14:00	72	11		83	6	11		17
15:00	1			1				
16:00								
17:00								
18:00								
Grand total	823	415	25	1263	191	360	15	566

Figure 4.7: The different types of users and their count at different times of the day

- The number of total people is too high: As most other exhibitors do not move around to visit other stands and most staff is standing at the entry the number of people passing by close is likely to be lower.
- The number of visitors is too low: As not only visitors passed by the Liveanalytics stand, but also a certain amount of exhibitors and staff, this actual number is likely to be higher. Additionally, many devices part of the equipment emits probe requests as well.

The amount of devices counted by the MacScavenger system (425) is somewhere in the middle between 566 and 360 and, therefore, is very well approximating the number of people present at the event. Especially, when the amount of detected MAC addresses is compared to the amount of devices identified, the contribution of the MacScavenger can be seen.

Additionally, the amount of people seen multiple times using MAC address randomization is 370 which is almost congruent with the number of visitors.

Chapter 5

Summary and Conclusion

Passive device fingerprinting is a controversial topic as it allows for tracking any device and also its carrier without any interaction between the monitoring and the monitored device. To protect users from unwanted surveillance, manufacturers started masking the main identifying property of modern network transmission: The MAC address of the NIC. By replacing the real MAC address with an alias in specific situations, *MAC address randomization* made a simple MAC address-based tracing impossible. This led to a completely new field of research, that tries to find ways to overcome the difficulties induced by this technique.

This thesis investigated the most common approaches to tackle MAC address randomization and discussed their proposed mechanisms by highlighting strengths and weaknesses. This work showed that no approach can be (i) effective, (ii) stable over time and device, and (iii) easy to deploy at the same time. On the one hand, while physical layer based approaches obtain very high precision, they suffer from high costs and complex hardware operation, that is necessary to investigate the network traffic in that way. On the other hand, MAC layer-based or content-based approaches are easy to deploy as they rely on off-the-shelf hardware and software, but suffer from unstable results regarding different devices and obtain only low precision in identifying devices.

To overcome this trade-off, this work focused on a combined approach: It attempts to use the stability and device-independence of a location-based approach and combines it with the informative value of a content-based approach. The system aimed primarily at the ability to distinguish two devices from each other, even if both devices are using MAC address randomization. To test and evaluate the proposed system, a series of experiments was conducted.

The evaluation showed, that RSSI-based localization performs better than anticipated and it suffices as a proxy to distinguish different devices from each. It demonstrated that the proposed combined approach is able to approximate the number of devices present with a high level of accuracy (425 predicted people, 360 to 566 present people) and that it is able to clean the bias resulting from MAC randomization when counting people.

5.1 Possible Future Improvements

The two values used for the proposed system - the RSSI value and the IEs - are both, the topic of their own fields of research. Improvements to the system proposed in this work would, therefore, mainly include in-depth research on the behavior of these two parameters.

Even though various experiments were conducted to evaluate RSSI values for their use in localization, more research is needed to precisely assess their behavior under different environments' conditions. Possible insightful experiments would include tests of the localization approach on bigger and broader areas, including multiple devices once, more than just 4 APs, or testing areas with obstacles. Moreover, the formation and creation of IEs as well as their contents are for the most part unknown as well. A thorough investigation of different device types, manufacturers, and OSs would be fruitful.

A variety of improvements to the proposed system itself are also imaginable: A more responsive user interface that would offer more possibilities to the user, better error handling of the monitor nodes in case of unexpected behavior, and also improvements to the algorithms used for localization would greatly improve speed, stability, and ease of use.

One final area for possible future improvement would be to introduce a variety of experiments to evaluate different hardware types. As this thesis has demonstrated, the hardware itself most likely has a significant impact on the accuracy of the localization and, ultimately, on the accuracy of the predictions of the system.

Bibliography

- [1] P. Hindle, “History of Wireless Communications,” 7 2015. [Online]. Available: <https://www.microwavejournal.com/articles/24759-history-of-wireless-communications>
- [2] F. F. Kuo, “Computer Networks - the ALOHA System,” *Journal of Research of the National Bureau of Standards*, vol. 86, no. 6, 8 1981.
- [3] Johns Hopkins School of Public Health, “Wireless Networking - History of Wireless.” [Online]. Available: <https://web.archive.org/web/20070210131824/http://www.jhsph.edu/wireless/history.html>
- [4] M. S. Gast, *802.11 Wireless Networks: The Definitive Guide, Second Edition*. O’Reilly Media, Inc., 2005.
- [5] J. Salazar, *Wireless Networks*, 1st ed. Czech Technical University of Prague, 2017. [Online]. Available: <http://www.techpedia.eu>
- [6] J. M. Kahn and J. R. Barry, “Wireless Infrared Communications,” *IEEE*, 1997. [Online]. Available: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=554222>
- [7] Keith Gibbs, “Microwave ovens and resonance in molecules,” 2020. [Online]. Available: http://www.schoolphysics.co.uk/age16-19/Waveproperties/Waveproperties/text/Microwave_ovens/index.html
- [8] E. Griffith, “WECA becomes Wi-Fi Alliance,” 10 2002. [Online]. Available: <http://www.internetnews.com/wireless/article.php/1474361/WECA-becomes-Wi-Fi-Alliance.htm>
- [9] IEEE Standards Association, “History of IEEE.” [Online]. Available: <https://www.ieee.org/about/ieee-history.html>
- [10] IEEE, “IEEE SA - Standards Group MAC Address.” [Online]. Available: <https://standards.ieee.org/products-services/regauth/grpmac/index.html>
- [11] D. E. Eastlake 3rd and J. Abley, “IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters,” pp. 1–27, 2013. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc7042.txt>
- [12] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, “A Study of MAC Address Randomization in Mobile Devices and When it Fails,” *Proceedings on Privacy Enhancing Technologies*, no. 4, pp. 365–383, 2017.

- [13] IEEE Standards Association, “Guidelines for Use of Extended Unique Identifier (EUI), Organizationally Unique Identifier (OUI), and Company ID (CID),” IEEE, Tech. Rep., 2017. [Online]. Available: <https://standards.ieee.org/develop/regauth/tut/eui.pdf>
- [14] I. SA, “MA-L Block Assignment.” [Online]. Available: <https://standards.ieee.org/products-services/regauth/oui/index.html>
- [15] C. Matte, “Wi-Fi Tracking: Fingerprinting Attacks and Counter-Measures,” Ph.D. dissertation, INSA Lion, 2017.
- [16] Wikipedia, “IEEE 802.11.” [Online]. Available: https://en.wikipedia.org/wiki/IEEE_802.11
- [17] Melissa Brown, “OSI Model: Part One,” 9 2018. [Online]. Available: https://medium.com/@melissabrown_44103/osi-model-part-one-d60e363390ac
- [18] IEEE, “IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, pp. 1–3534, 12 2016.
- [19] M. Cunche and C. Matte, “On Wi-Fi Tracking and the Pitfalls of MAC Address Randomization,” Ph.D. dissertation, Université Lion.
- [20] S. O’Dea, “Mobile OS market share 2019 | Statista,” 2 2019. [Online]. Available: <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>
- [21] StatCounter Global Stats, “Mobile Operating System Market Share Worldwide,” 6 2019. [Online]. Available: <https://gs.statcounter.com/os-market-share/mobile/worldwide>
- [22] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, and F. Piessens, “Why MAC Address Randomization is Not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms,” in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 413–424. [Online]. Available: <https://doi.org/10.1145/2897845.2897883>
- [23] Osama Abukmail, “Wifi Mac Changer.” [Online]. Available: <https://play.google.com/store/apps/details?id=com.wireless.macchanger>
- [24] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, “Wireless device identification with radiometric signatures,” in *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2008, pp. 116–127.
- [25] D. Vasisht, S. Kumar, and D. Katabi, “Decimeter-Level Localization with a Single WiFi Access Point,” in *Proceedings of the 13th Usenix Conference on Networked Systems Design and Implementation*, ser. NSDI’16. USA: USENIX Association, 2016, p. 165–178.

- [26] T. Vo-Huu, T. Vo-Huu, and G. Noubir, "Fingerprinting Wi-Fi Devices Using Software Defined Radios," in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, 2016, pp. 3–14.
- [27] P. Robyns, B. Bonné, P. Quax, and W. Lamotte, "Noncooperative 802.11 MAC Layer Fingerprinting and Tracking of Mobile Devices," *Security and Communication Networks*, vol. 2017, p. 6235484, 2017. [Online]. Available: <https://doi.org/10.1155/2017/6235484>
- [28] S. Jana and S. K. Kasera, "On Fast and Accurate Detection of Unauthorized Wireless Access Points Using Clock Skews," in *MobiCom*, San Francisco California, 9 2008.
- [29] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Computer Society Press*, vol. 2, no. 2, 5 2005.
- [30] M. Cunche, "I know your MAC Address : Targeted tracking of individual using Wi-Fi," *Journal of Computer Virology and Hacking Techniques*, no. 4, 2014.
- [31] D. Gentry and A. Pennarun, "Passive Taxonomy of Wifi Clients using MLME Frame Contents," 2016.
- [32] C. Matte, M. Cunche, F. Rousseau, and M. Vanhoef, "Defeating MAC Address Randomization Through Timing Attacks," in *ACM WiSec 2016*, Darmstadt, Germany, 7 2016. [Online]. Available: <https://hal.inria.fr/hal-01330476>
- [33] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. van Randwyk, and D. Sicker, "Passive data link layer 802.11 wireless device driver fingerprinting," *15th USENIX Security Symposium*, pp. 167–178, 2006.
- [34] F. Goovaerts, G. Acar, R. Galvez, F. Piessens, and M. Vanhoef, "Improving Privacy Through Fast Passive Wi-Fi Scanning," in *Secure IT Systems*, A. Askarov, R. R. Hansen, and W. Rafnsson, Eds. Cham: Springer International Publishing, 2019, pp. 37–52.
- [35] K.-J. Djervbrant and A. Häggström, "A Study on Fingerprinting of Locally Assigned MAC-Addresses," Ph.D. dissertation, Halmstad University, Halmstad, 2019. [Online]. Available: <http://hh.diva-portal.org/smash/get/diva2:1324343/FULLTEXT02.pdf>
- [36] S. Aneja, N. Aneja, and M. S. Islam, "IoT Device Fingerprint using Deep Learning," *CoRR*, vol. abs/1902.0, 2019. [Online]. Available: <http://arxiv.org/abs/1902.01926>
- [37] A. Zhang, Y. Yuan, Q. Wu, S. Zhu, and J. Deng, "Wireless Localization Based on RSSI Fingerprint Feature Vector," *International Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–7, 2015.
- [38] Wikipedia, "True Range Multilateration." [Online]. Available: https://en.wikipedia.org/wiki/True_range_multilateration
- [39] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor Localization via Channel Response," *ACM Comput. Surv.*, vol. 46, no. 2, 12 2013. [Online]. Available: <https://doi.org/10.1145/2543581.2543592>

- [40] W. Zegeye, S. Amsalu, Y. Astatke, and F. Moazzami, “WiFi RSS fingerprinting indoor localization for mobile devices,” in *IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2016, pp. 1–6.
- [41] Y. Ma, G. Zhou, and S. Wang, “WiFi Sensing with Channel State Information: A Survey,” *ACM Comput. Surv.*, vol. 52, no. 3, 6 2019. [Online]. Available: <https://doi.org/10.1145/3310194>
- [42] C. Feng, S. Arshad, and Y. Liu, “MAIS: Multiple Activity Identification System Using Channel State Information of WiFi Signals,” in *12th International Conference, WASA*, Guilin, China, 2017, pp. 419–432.
- [43] H. Liu, Y. Wang, J. Liu, J. Yang, Y. Chen, and H. V. Poor, “Authenticating Users Through Fine-Grained Channel Information,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 251–264, 2 2018.
- [44] W. Xi, J. Zhao, X. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang, “Electronic frog eye: Counting crowd using WiFi,” in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 4 2014, pp. 361–369.
- [45] L. Cheng and J. Wang, “How Can I Guard My AP? Non-Intrusive User Identification for Mobile Devices Using WiFi Signals,” in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 91–100. [Online]. Available: <https://doi.org/10.1145/2942358.2942373>
- [46] F. Gringoli, M. Schulz, J. Link, and M. Hollick, “Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets,” in *Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization*, ser. WiNTECH ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 21–28. [Online]. Available: <https://doi.org/10.1145/3349623.3355477>
- [47] Y. Li and T. Zhu, “Gait-Based Wi-Fi Signatures for Privacy-Preserving,” in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ser. ASIA CCS ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 571–582. [Online]. Available: <https://doi.org/10.1145/2897845.2897909>
- [48] W. Wang, Y. Chen, and Q. Zhang, “Privacy-Preserving Location Authentication in Wi-Fi Networks Using Fine-Grained Physical Layer Signatures,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1218–1225, 2 2016.
- [49] Keysight Technologies, “89641S VXi-Based 6.0 GHz RF Vector Signal Analyzer [Obsolete] | Keysight.” [Online]. Available: <https://www.keysight.com/en/pd-1000004548%3Aepsg%3Apro-pn-89641S/vxi-based-60-ghz-rf-vector-signal-analyzer?cc=US&lc=eng>
- [50] Keysight Technologies, “Keysigt N9020B-550 50GHz MXA signal analyzer, 160M, PreAmp, CAL, Warranty LOADED for sale online.” [Online]. Available: <https://www.ebay.com/c/17017481240>

- [51] Mathuranathan, “Log Distance Path Loss or Log Normal Shadowing Model - GaussianWaves,” 9 2013. [Online]. Available: <https://www.gaussianwaves.com/2013/09/log-distance-path-loss-or-log-normal-shadowing-model/>
- [52] Wikipedia, “Non-linear least squares.” [Online]. Available: https://en.wikipedia.org/wiki/Non-linear_least_squares
- [53] R. E. Kálmán, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960. [Online]. Available: <http://www.cs.unc.edu/~welch/kalman/media/pdf/Kalman1960.pdf>
- [54] Wikipedia, “Kalman filter.” [Online]. Available: https://en.wikipedia.org/wiki/Kalman_filter
- [55] H. Liu, Y. Wang, J. Liu, J. Yang, and Y. Chen, “Practical User Authentication Leveraging Channel State Information (CSI),” in *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, ser. ASIA CCS ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 389–400. [Online]. Available: <https://doi.org/10.1145/2590296.2590321>
- [56] Dave Kuhlman, “A Python Book: Beginning Python, Advanced Python, and Python Exercises,” 4 2012. [Online]. Available: https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python_book_01.html
- [57] Chris M, “BeginnersGuide/Overview,” 9 2019. [Online]. Available: <https://wiki.python.org/moin/BeginnersGuide/Overview>
- [58] Python.org, “The Zen of Python,” 8 2004. [Online]. Available: <https://www.python.org/dev/peps/pep-0020/>
- [59] Python.org, “Sunsetting Python 2.” [Online]. Available: <https://www.python.org/doc/sunset-python-2/>
- [60] P. L. Bewig, “SRFI 41: Streams,” 10 2007. [Online]. Available: <https://srfi.schemers.org/srfi-41/srfi-41.html>
- [61] Matthew Rocklin, “Streamz,” 2017. [Online]. Available: <https://streamz.readthedocs.io/en/latest/index.html>
- [62] Cisco Systems, *Cisco Networking Academy Program CCNA 1 and 2 Companion Guide*. 201 West 103rd Street, Indianapolis, IN 46290 USA: Cisco Press, 2003.
- [63] Oracle, “What Is a Socket?” 2019. [Online]. Available: <https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>
- [64] Python.org, “socketserver — A framework for network servers.” [Online]. Available: <https://docs.python.org/3/library/socketserver.html>
- [65] MongoDB, “What Is MongoDB?” [Online]. Available: <https://www.mongodb.com/what-is-mongodb>

- [66] Python.org, “cmd - Support for line-oriented command interpreters.” [Online]. Available: <https://docs.python.org/3/library/cmd.html>
- [67] Wireshark, “tshark.” [Online]. Available: <https://www.wireshark.org/docs/man-pages/tshark.html>
- [68] Pandas, “Pandas DataFrames.” [Online]. Available: https://pandas.pydata.org/pandas-docs/stable/getting_started/dsintro.html
- [69] AccuWeather, “Winterthur, Zürich, Schweiz Wetter monatlich.” [Online]. Available: <https://www.accuweather.com/de/ch/winterthur/316623/april-weather/316623>
- [70] AccuWeather, “Zürich, Zürich, Schweiz Wetter monatlich.” [Online]. Available: https://www.accuweather.com/de/ch/zurich/8002/june-weather/405979_pc?year=2020
- [71] C. Halter, “BluePIL: Fully Passive Identification and Localization of Bluetooth Devices in Near-Real-Time,” Ph.D. dissertation, University of Zürich, 2020.
- [72] J. Luomala and I. Hakala, “Effects of temperature and humidity on radio signal strength in outdoor wireless sensor networks,” in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2015, pp. 1247–1255.

Abbreviations

ACK Acknowledgement

AIEE American Institute of Electrical Engineers

AP Access Points

CID Company ID

CRC Cyclic Redundancy Check

CSI Channel State Information

CTS Clear-to-Send

DFS Dynamic Frequency Selection

DSSS Direct-Sequence Spread-Spectrum

ELI Extended Local Identifier

FCS Frame Check Sequence

GHz Giga Herz

IE Information Elements

IEEE Institute of Electrical and Electronics Engineers

IEEE RA IEEE Registration Authority

IFAT Inter Frame Arrival Time

IP Internet Protocol

ISM Industrial, Scientific and Medical

LAN Local Area Network

ls Listing

MAC	Media Access Control
MLME	MAC Layer Management Entity
NIC	Network Interface Card
OFDM	Orthogonal Frequency-Division Multiplexing
OS	Operating System
OSI	Open Systems Interconnection
OUI	Organizationally Unique Identifier
P2P	Peer-to-Peer
RF	Radio Frequency
RSSI	Received Signal Strength Indicator
RTS	Request-to-Send
SSID	Service Set Identifiers
TCP	Transmission Control Protocol
TPC	Transmit Power Control
UDP	User Datagram Protocol
UUID	Universally Unique Identifier
WECA	Wireless Ethernet Compatibility Alliance
WLAN	Wireless Local Area Network
WPS	Wi-Fi Protected Setup

List of Figures

2.1	The Organizationally Unique Identifier (OUI). Figure by [13].	9
2.2	The Company ID (CID). Figure by [13].	10
2.3	The IEEE 802 family and its relation to the OSI model [4].	11
2.4	The OSI model [17].	12
2.5	Components of 802.11 LANs [4].	13
2.6	Generic 802.11 MAC frame. Figure by [4].	15
2.7	Frame control field. Figure by [4].	15
2.8	Duration/ID field. Figure by [4].	16
2.9	Sequence control field. Figure by [4].	16
2.10	Probe Request Structure. Figure by [4].	16
2.11	Probe Request Bursts. Figure by [19].	17
3.1	The Interaction of the Shell with the data gathering and the data analysis components	31
3.2	The topology of the MacScavenger system regarding the Data Gathering Component	31
3.3	The topology of the MacScavenger system regarding the Data Analysis Component	32
3.4	The Data Gathering Pipeline on MacScavenger Sync node	34
3.5	The Data Gathering Pipeline on MacScavenger Monitor nodes	35
3.6	The Data Analysis Pipeline	36
3.7	Sequence Diagram of the Basic Shell Operations	38
3.8	Sequence Diagram of the Initialization of the Monitor Nodes NICs	39

3.9	Sequence Diagram of the Data Gathering Process	40
3.10	Sequence Diagram of the Data Analysis Process	41
4.1	Methodology of Experiment 1: At each point in the grid, the sending device was transmitting probe requests that could be captured by the four monitoring devices in the corners.	44
4.2	Result of Experiment 1 on position (0,1): The true position is shown with the empty big blue circle. The different estimates of the unfiltered (light blue), Kalman Filter based (dark blue) and variance based approaches (orange, green, violet) are shown as filled dots.	45
4.3	Result of Experiment 2 at position (2,5). For legend <i>c.f.</i> Figure 4.2.	46
4.4	Result of step 1 of Experiment 2: the different variations of the location estimate are close together, but very far apart from the true location of the sender. For legend <i>c.f.</i> Figure 4.2.	47
4.5	A map of the lifealytics stand at the "Reloading Live" event	49
4.6	The amount of first time entrances over time	50
4.7	The different types of users and their count at different times of the day	51

List of Tables

2.1	The different licensed and unlicensed radio frequency bands by [4]	7
2.2	Overview of network services according to [4]	14

List of Algorithms

1	Decision making process of the proposed system	26
---	--	----

Appendix A

MacScavenger Github Repository

The code as well as a few simple instructions for the *MacScavenger* can be found on Github via the following link: <https://github.com/HolzmanoLagrene/MacScavenger.git>. The repository is maintained by **HolzmanoLagrene**. For more information please see the *README*-file.

Appendix B

Additional Files

The following list of files is available under the following **link**.

1. Master Thesis as PDF (mac-scavenger-a-passive-method-handling-MAC-randomization-on-mobile-devices.pdf)
2. Master Thesis as Latex source code (mac-scavenger-a-passive-method-handling-MAC-randomization-on-mobile-devices.zip)
3. Reference to *MacScavenger* Github Repository (macscavenger-github-repository-reference.txt)
4. *MacScavenger* source code as per 5.08.2020 (macscavenger-github-repository.zip)
5. Datasets of all three experiments (experiments.zip)
6. Intermediate Presentation (intermedpresentation.pptx)