Wi-Fi tracking : Fingerprinting attacks and counter-measures

Célestin Matte

N°d'ordre NNT : 2017LYSEI114

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
INSA Lyon

Ecole Doctorale ED 512
InfoMaths

Spécialité/ discipline de doctorat :
Informatique

Soutenue publiquement le 07/12/2017, par :
Célestin Matte

Wi-Fi Tracking: Fingerprinting Attacks and Counter-Measures

Devant le jury composé de :

Nom, prénom	grade/qualité	établissement/entreprise	
Nguyen, Benjamin	Professeur des universités	INSA Centre Val de Loire	Rapporteur.e
Rasmussen, Kasper	Associate professor	University of Oxford	Rapporteur.e
Chrisment, Isabelle	Professeur des universités	Université de Lorraine	Présidente
Risset, Tanguy	Professeur des universités	INSA Lyon	Examinateur
Neumann, Christoph	Principal scientist	Technicolor	Examinateur
Minier, Marine	Professeur des universités	Université de Lorraine	Directrice de thèse
Cunche, Mathieu	Maître de conférences	Insa Lyon	Co-directeur de thèse

Cette thèse a été préparée à

CITI Laboratory (Center of Innovation in Telecommunications and Integration of service) - INSA Lyon
Bâtiment Claude Chappe
6 avenue des Arts
F-69621 Villeurbanne
France

et dans le cadre de

Équipe Inria Privatics
Centre Inria Grenoble Rhône-Alpes
Inria – antenne Lyon-La Doua
Bâtiment CEI-2
56 Boulevard Niels Bohr
69100 Villeurbanne

sur un financement de

Région Rhône-Alpes

Département FEDORA – INSA Lyon - Ecoles Doctorales – Quinquennal 2016-2020

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
CHIMIE	**CHIMIE DE LYON** **http://www.edchimie-lyon.fr** Sec : Renée EL MELHEM Bat Blaise Pascal 3e etage secretariat@edchimie-lyon.fr Insa : R. GOURDON	M. Stéphane DANIELE Institut de Recherches sur la Catalyse et l'Environnement de Lyon IRCELYON-UMR 5256 Équipe CDFA 2 avenue Albert Einstein 69626 Villeurbanne cedex directeur@edchimie-lyon.fr
E.E.A.	**ELECTRONIQUE, ELECTROTECHNIQUE, AUTOMATIQUE** **http://edeea.ec-lyon.fr** Sec : M.C. HAVGOUDOUKIAN Ecole-Doctorale.eea@ec-lyon.fr	M. Gérard SCORLETTI Ecole Centrale de Lyon 36 avenue Guy de Collongue 69134 ECULLY Tél : 04.72.18 60.97 Fax : 04 78 43 37 17 Gerard.scorletti@ec-lyon.fr
E2M2	**EVOLUTION, ECOSYSTEME, MICROBIOLOGIE, MODELISATION** **http://e2m2.universite-lyon.fr** Sec : Sylvie ROBERJOT Bât Atrium - UCB Lyon 1 04.72.44.83.62 Insa : H. CHARLES secretariat.e2m2@univ-lyon1.fr	M. Fabrice CORDEY CNRS UMR 5276 Lab. de géologie de Lyon Université Claude Bernard Lyon 1 Bât Géode 2 rue Raphaël Dubois 69622 VILLEURBANNE Cédex Tél : 06.07.53.89.13 cordey@univ-lyon1.fr
EDISS	**INTERDISCIPLINAIRE SCIENCES-SANTE** **http://www.ediss-lyon.fr** Sec : Sylvie ROBERJOT Bât Atrium - UCB Lyon 1 04.72.44.83.62 Insa : M. LAGARDE secretariat.ediss@univ-lyon1.fr	Mme Emmanuelle CANET-SOULAS INSERM U1060, CarMeN lab, Univ. Lyon 1 Bâtiment IMBL 11 avenue Jean Capelle INSA de Lyon 696621 Villeurbanne Tél : 04.72.68.49.09 Fax :04 72 68 49 16 Emmanuelle.canet@univ-lyon1.fr
INFOMATHS	**INFORMATIQUE ET MATHEMATIQUES** **http://edinfomaths.universite-lyon.fr** Sec :Renée EL MELHEM Bat Blaise Pascal, 3e étage Tél : 04.72. 43. 80. 46 Fax : 04.72.43.16.87 infomaths@univ-lyon1.fr	M. Luca ZAMBONI Bâtiment Braconnier 43 Boulevard du 11 novembre 1918 69622 VILLEURBANNE Cedex Tél :04 26 23 45 52 zamboni@maths.univ-lyon1.fr
Matériaux	**MATERIAUX DE LYON** **http://ed34.universite-lyon.fr** Sec : Marion COMBE Tél:04-72-43-71-70 –Fax : 87.12 Bat. Direction ed.materiaux@insa-lyon.fr	M. Jean-Yves BUFFIERE INSA de Lyon MATEIS Bâtiment Saint Exupéry 7 avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél : 04.72.43 71.70 Fax 04 72 43 85 28 Ed.materiaux@insa-lyon.fr
MEGA	**MECANIQUE, ENERGETIQUE, GENIE CIVIL, ACOUSTIQUE** **http://edmega.universite-lyon.fr/** Sec : Marion COMBE Tél:04-72-43-71-70 –Fax : 87.12 Bat. Direction mega@insa-lyon.fr	M. Philippe BOISSE INSA de Lyon Laboratoire LAMCOS Bâtiment Jacquard 25 bis avenue Jean Capelle 69621 VILLEURBANNE Cedex Tél : 04.72 .43.71.70 Fax : 04 72 43 72 37 Philippe.boisse@insa-lyon.fr
ScSo	**ScSo*** **http://ed483.univ-lyon2.fr/** Sec : Viviane POLSINELLI Brigitte DUBOIS Insa : J.Y. TOUSSAINT Tél : 04 78 69 72 76 viviane.polsinelli@univ-lyon2.fr	M. Christian MONTES Université Lyon 2 86 rue Pasteur 69365 LYON Cedex 07 Christian.montes@univ-lyon2.fr

*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

Remerciements

Je tiens à remercier toutes les personnes qui ont participé de près ou de loin à cette thèse et m'ont soutenu durant cette période, en particulier...

...mon encadrant, Mathieu Cunche, pour ses conseils avisés, son dynamisme et ses nombreuses propositions concernant la thèse, la liberté d'organisation qu'il m'a laissée, tout le travail scientifique et administratif fourni, ainsi que sa confiance;

...ma directrice de thèse, Marine Minier, pour la relecture incroyablement rapide de ce manuscrit;

...mon troisième encadrant, Franck Rousseau, pour son suivi depuis Grenoble;

...les coauteurs des articles publiés, pour leur sérieux et leur travail;

...l'ensemble des membres du laboratoire, et en particulier les adeptes du *coin café*, haut lieu de riches discussions politiques, ou non;

...l'intégralité de la section danse-études de l'INSA de Lyon, qui a comblé ces trois années de projets dansés passionnants, de potins et de week-end boueux. Je remercie en particulier Delphine Savel pour toute l'énergie dépensée dans l'organisation de cette section;

...les danseuses et les techniciens qui m'ont suivi jusqu'au bout de mon projet d'art numérique, une expérience riche en apprentissage et en émotions;

...les membres d'un certain salon Jabber, toujours à l'avant-garde;

...mes parents et grands-parents, pour leur soutien permanent;

...Nagham, pour la relecture volontaire d'une thèse bien loin de son domaine;

...sans compter toutes les personnes que je ne peux mentionner par souci de brièveté.

Abstract

Wi-Fi Tracking: Fingerprinting Attacks and Counter-Measures

The recent spread of everyday-carried Wi-Fi-enabled devices (smartphones, tablets and wearable devices) comes with a privacy threat to their owner, and to society as a whole. These devices continuously emit signals which can be captured by a passive attacker using cheap hardware and basic knowledge. These signals contain a unique identifier, called the MAC address. To mitigate the threat, device vendors are currently deploying a countermeasure on new devices: MAC address randomization. Unfortunately, we show that this mitigation, in its current state, is insufficient to prevent tracking.

To do so, we introduce several attacks, based on the content and the timing of emitted signals. In complement, we study implementations of MAC address randomization in some recent devices, and find a number of shortcomings limiting the efficiency of these implementations at preventing device tracking.

At the same time, we perform two real-world studies. The first one considers the development of actors exploiting this issue to install Wi-Fi tracking systems. We list some real-world installations and discuss their various aspects, including regulation, privacy implications, consent and public acceptance. The second one deals with the spread of MAC address randomization in the devices population.

Finally, we present two tools: an experimental Wi-Fi tracking system for testing and public awareness raising purpose, and a tool estimating the uniqueness of a device based on the content of its emitted signals even if the identifier is randomized.

Keywords: Computer science, Privacy, Network, Fingerprinting, Wi-Fi, Tracking, Probe request, MAC address randomization

Résumé

Traçage Wi-Fi : Attaques par Prise d'Empreinte et Contre-Mesures

Le récent développement des appareils portatifs possédant une interface Wi-Fi (smartphones, tablettes et "wearables") s'accompagne d'une menace sur la vie privée de leurs utilisateurs, et sur la société toute entière. Ces appareils émettent en continu des signaux pouvant être capturés par un attaquant passif, à l'aide de matériel peu coûteux et de connaissances basiques. Ces signaux contiennent un identifiant unique appelé l'adresse MAC. Pour faire face à cette menace, les acteurs du secteur déploient actuellement une contre-mesure sur les appareils récents: le changement aléatoire de l'adresse MAC. Malheureusement, nous montrons que cette mesure, dans son état actuel, n'est pas suffisante pour empêcher le traçage des appareils.

Pour cela, nous introduisons plusieurs attaques basées sur le contenu et la répartition temporelle des signaux. En complément, nous étudions les implémentations du changement aléatoire de l'adresse MAC sur des appareils récents, et trouvons un certain nombre de manquements limitant l'efficacité de ces implémentations à prévenir le traçage.

En parallèle, nous effectuons deux études de terrain. La première s'attaque au développement des acteurs exploitant les problèmes cités plus haut afin d'installer des systèmes de traçage basés sur le Wi-Fi. Nous listons certaines de ces installations et examinons plusieurs aspects de ces systèmes : leur régulation, les implications en terme de vie privée, les questions de consentement et leur acceptation par le public. La seconde étude concerne la progression du changement aléatoire d'adresse MAC dans la population des appareils.

Finalement, nous présentons deux outils : le premier est un système de traçage expérimental développé pour effectuer des tests et sensibiliser le public aux problèmes de vie privée liés à de tels systèmes. Le second estime l'unicité d'un appareil en se basant sur le contenu des signaux qu'il émet, même si leur identifiant est modifié.

Mots-clés : Informatique, Vie privée, Prise d'empreintes (fingerprinting), Wi-Fi, Traçage, Probe request, changement aléatoire de l'adresse MAC

Contents

List of Figures

List of Tables

List of acronyms

ANQP Access Network Query Protocol, page 47

AP Access Point, page 21

CID Company ID, page 23

CNIL *Commission nationale de l'informatique et des libertés*, page 29

CTS Clear to Send, page 20

DSSS Direct-sequence spread spectrum, page 24

FCS Frame Check Sequence, page 24

FN False Negative, page 117

FNR False Negative Rate, page 117

FP False Positive, page 117

FPR False Positive Rate, page 117

FTC Federal Trade Commission, page 70

GSM Groupe Spécial Mobile / Global System for Mobile Communications, page 49

HT High-Throughput, page 25

ICMP Internet Control Message Protocol, page 40

IE Information Element, page 24

IFAT Inter-Frame Arrival Time, page 24

IMSI International Mobile Subscriber Identity, page 49

LA Locally Administered, page 22

LAN Local Area Network, page 40

NFC Near-Field Communication, page 51

NIC Network Interface Controller, page 22

OFDM Orthogonal Frequency Division Multiplexing, page 41

OSR Overall Success Rate, page 117

OUI Organisationally Unique Identifier, page 22

PDU Protocol Data Unit, page 17

PNL Preferred Network List, page 24

PPDU PLCP Protocol Data Unit, page 24

PSDU PLCP Service Data Unit, page 24

RF Radio-Frequency, page 38

RFID Radio-frequency identification, page 50

RSSI Received signal strength indicator, page 25

RTS Request to Send, page 20

SIM Subscriber Identity Module, page 49

SSID Service Set Identifier, page 22

STA Station, page 21

TMSI Temporary Mobile Subscriber Identity, page 49

TN True Negative, page 117

TNR True Negative Rate, page 117

TP True Positive, page 117

TPR True Positive Rate, page 117

TSF Timing Synchronization Function, page 40

UUID Universally Unique Identifier, page 101

WEP Wired Equivalent Privacy, page 24

WPS Wi-Fi Protected Setup, page 25

Glossary

Frame: On OSI's MAC layer, "messages" (PDU) are called *frames*.

Access Point (AP): A device providing access to a Wi-Fi network to other devices, after an association procedure.

Probe request: A frame emitted by Wi-Fi-enabled devices to discover surrounding Access Points (networks).

Probing: The act of sending probe requests.

Active scanning: One of the two possible methods used by Wi-Fi-enabled mobile devices to discover surrounding APs, in which devices send probe requests. The other one is *passive scanning*, in which devices listen for APs' network advertisements.

Associated/Unassociated: A Wi-Fi-enabled device which went through the whole association procedure with an AP is *associated* (with that AP), and has access to the latter's network.

MAC address: The unique identifier of a Wi-Fi card added in every frame, including probe requests.

Randomization: A generic term we use in this thesis to describe MAC address randomization, i.e. the fact of frequently changing the MAC address of a device to a random one.

Locally Administered bit (LA bit): A bit in the MAC address indicating whether this address is the original one provided by the manufacturer of the Wi-Fi card or a manually-set address.

Global/local address: A MAC address whose Locally Administered bit is not set to 1 is a global address. Usually, it means that the device using this address kept the original (unique) MAC address provided by the Wi-Fi card's manufacturer. Conversely, a MAC address whose LA bit is set to 1 is a local address.

Random address: An address which has been changed by MAC address randomization.

Usually, this is a local address (LA bit set to 1), but not always.

Information Element (IE) / tags: Fields added to probe requests, containing information about the emitting device's capabilities

Service Set Identifier (SSID): A human-readable name identifying a network. E.g.: "FreeWifi", "eduroam", "McDonald's Pyongyang", "FBI surveillance van", etc. SSID is a mandatory Information Element in probe requests.

Broadcast/directed probe request: A probe request is *broadcast* if its SSID tag is null, *directed* otherwise.

Preferred Network List: The list of networks a device knows, i.e. networks to which it associated with in the past, plus manually-added networks.

Burst: A set of probe requests sent within a short timeframe (less than 100ms) during active scanning. On a given channel, a burst often consists in either one probe request for each SSID in the device's PNL (up to a certain length), or a unique broadcast probe request.

Hidden Access Point: An AP which does not advertise its presence except when it receives a directed probe request containing its related SSID.

Chapter I

Introduction

I.1 Introduction

The recent wide-scale spread of Wi-Fi-enabled mobile devices came with a privacy threat to their owner. In order to discover surrounding networks, these devices continuously emit signals. These signals contain a number uniquely identifying the emitting device: the MAC address. As a consequence, emitted signals may be collected by a passive attacker, which can then obtain sensitive information such as the presence of the device along time. This constitutes a privacy issue for its owner, who can be tracked by any passive attacker. Such a design flaw is exploited by real-world actors, from government agencies for surveillance [71, 179] to retailers for statistics on customers [26] or cities for urban planning [115].

Fortunately, device vendors (including Wi-Fi chipset manufacturers and device developers) are currently (as of 2017) working on the deployment of a countermeasure to this unique-identifier issue: MAC address randomization [206, 189]. As its name suggest, the idea is to frequently change the device's MAC address with a new, random identifier.

In this thesis, we study the effectiveness of these implementations, by introducing new attacks showing that this measure, in itself, possesses some limitations.

This current chapter first introduces some background concepts mandatory to understand this document in section I.2. Section I.3 presents datasets used in the different experiments. Finally, section I.4 presents the structure of the whole document.

I.2 Background

In this section, we introduce core domains approached in this thesis: Wi-Fi, device fingerprinting, ubiquitous computing and privacy.

I.2.1 IEEE 802.11 Wi-Fi

We focus most of our work on one popular set of wireless protocols: the Wi-Fi technology, or more formally, IEEE 802.11 Wi-Fi [97]. The latter is a set of specifications on both the physical and the data-link layer of the OSI model. Wi-Fi comes in the form of a dozen protocols essentially varying in terms of used frequency, data rates and modulation. In itself, the Wi-Fi name is a trademark of the Wi-Fi Alliance non-profit organization [1]. Wi-Fi is mostly used to deploy short-range wireless networks (several dozens to hundreds of meters), such as enterprise or home networks. With the deployment of ubiquitous computing and the Internet of Things, it also takes its place as one of the possible standards to communicate with a wide range of small devices not possessing a wired interface (e.g. an Ethernet port), from smartphones to internet-connected baby-monitors. Wireless communications are often more convenient than wired ones for a number of reasons: easy deployment, reduced infrastructure cost, devices mobility, etc.

While using a wide range of already existing protocols from the physical to the Logical Link Control layer, the Wi-Fi standard extensively defines its own MAC layer protocol (802.11). At this level of the OSI layer, messages (or more precisely, the PDU) are called frames. The 802.11 protocol stack is displayed in a simplified form in figure I.1.

Frame types: The 802.11 protocol defines several types of frames (Type field in figure I.5): management, control, and data. The management frames which are relevant to this thesis are:

- *Probe requests*: sent by stations during active scanning (see below). When a probe request includes a non-null SSID, it is called a *directed* probe request and only APs related to this SSID are expected to respond. When the SSID is null, the probe request is a *broadcast* one, and all surrounding APs are expected to respond.
- *Probe responses*: replies to a probe request.
- *Beacons*: sent by APs to advertise their presence, for passive scanning.
- *Request to Send (RTS) and Clear To Send (CTS)* frames, seldom mentioned in

1. It is a play-on-word on Wireless and Hi-Fi and does not really mean anything else.

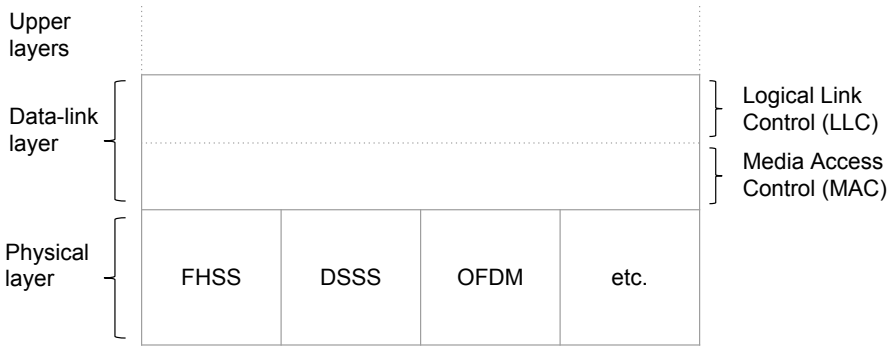


Figure I.1 – The 802.11 protocol stack (simplified).

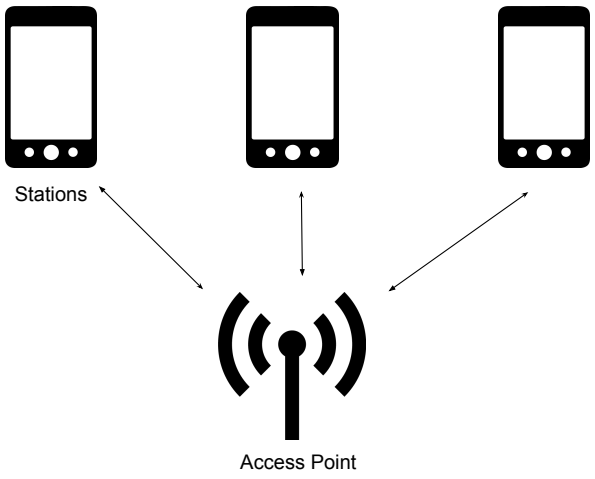


Figure I.2 – Wi-Fi Infrastructure mode.

this thesis, are used by devices as a way to reduce frame collisions.
The format of a Wi-Fi frame is schematized in figure I.5.

Topology: Several network modes can be used to build a Wi-Fi topology: infrastructure, ad-hoc, bridge or range-extender. The most used mode for mobile devices is infrastructure. In the latter, one device (the Access Point (AP)) acts as a switch for other devices, the stations (STA) (see figure I.2).

Service discovery: In order to join a Wi-Fi network, a station must go through an authentication and association mechanism. As long as this mechanism is not fully performed, the device is *unassociated* and can only exchange management frames with the AP and other devices. In order to discover surrounding APs, Wi-Fi-enabled devices employ either an *active* or a *passive* service discovery mode, as shown in figure I.3. In the active mode, Wi-Fi-enabled devices broadcast management frames known as *probe requests*, to which

surrounding APs reply with a *probe response*. These probe requests might contain names (Service Set Identifiers or SSIDs) of networks the device wants to connect to. In the passive mode case, devices passively listen to *beacons*, broadcast by APs, that announce the characteristics of the corresponding Wi-Fi network. Active service discovery is generally employed by mobile devices, because of its reduced energy consumption and faster speed, compared to the passive one. It's also the only way to discover hidden APs, i.e., APs which do not advertise their presence using beacons or by responding to broadcast probe requests. Active scanning also have other minor uses, such as speeding up reassociation, useful in VoIP calls [127], or when the device in a node of an ad hoc network [131].

Hidden Access Points: *Hidden* Access Points deserve some attention. They will only reveal their presence to devices sending probe requests including their SSID [1]. This mechanism makes it complicated to remove direct probe request as a countermeasure to some issues presented in this thesis (fingerprinting using SSIDs or probe requests timing). When a network's name is manually entered in a device's PNL, the device considers it as a network reachable through a hidden AP, and will therefore use directed probe requests to search for it.

Channels: On the physical layer, Wi-Fi devices commonly operate either on the 2.4 GHz or the 5 GHz frequency bands. In Europe, these frequency bands are subdivided in 13 overlapping bands from 2.400 to 2.4835 GHz, and 19 non-overlapping bands from 5.150 to 5.725 GHz (subdivided in 8 and 11 bands for regulation purpose). As an AP operates on a single band, stations usually broadcast probe requests on all available channels in order to discover all networks. By far, the most used channels are 1, 6 and 11 [2].

MAC address: In order to communicate, devices address each other on the MAC layer using a 6-byte globally-unique identifier called the MAC address. As seen in figure I.4, the first three bytes (prefix) of this address are an Organizationally Unique Identifier (OUI) which has to be bought by vendors from the IEEE Registration Authority in order to be used, so as to ensure the global uniqueness of MAC addresses. The last three bytes are the Network Interface Controller (NIC). One very specific bit of the MAC address is the seventh bit of the first byte of the OUI: the Locally Administered bit (LA bit). If set to 1, it indicates that the MAC address has been changed by the administrator of the device and is not guaranteed to be unique. It is unclear whether MAC address randomization during active scanning should set the LA bit to 1, as, to our knowledge, no document mentions this case explicitly [96]. A prefix whose LA bit is set to 1 is called

1. They may send empty beacons, but sending a probe request including their SSID is still necessary.
2. `https://wigle.net/stats#octetstats`, consulted on 2017.07.21

a Company ID (CID), and is automatically purchased by manufacturers along with their OUI equivalent [96]. For simplicity, we will use the *OUI* term when referring to both OUI and CID. It is possible for companies to buy private OUIs, which are not publicly tied to the company's name [96].

While other forms of MAC addresses exist (e.g. a 64-bit one used for IPv6), we only refer in the thesis to the 48-bit ones used in IEEE 802.11 in 2017.

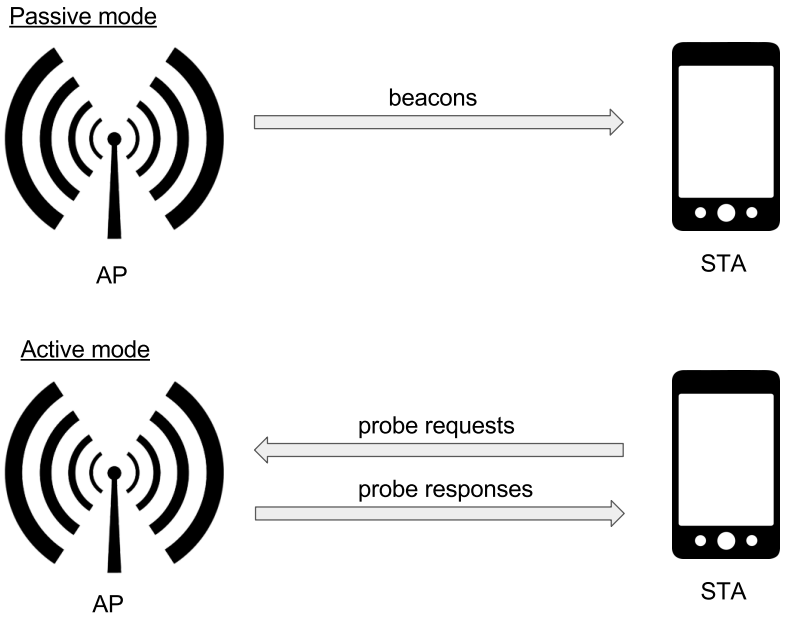


Figure I.3 – Both service discovery modes in IEEE 802.11. In the passive mode, access points broadcast beacons. In the active mode, the station broadcasts probe requests and the access points reply with probe responses.

Bursts of probe requests: For most devices, probe request frames are sent across the different channels in groups (bursts) within a small timeframe (usually less than 100ms

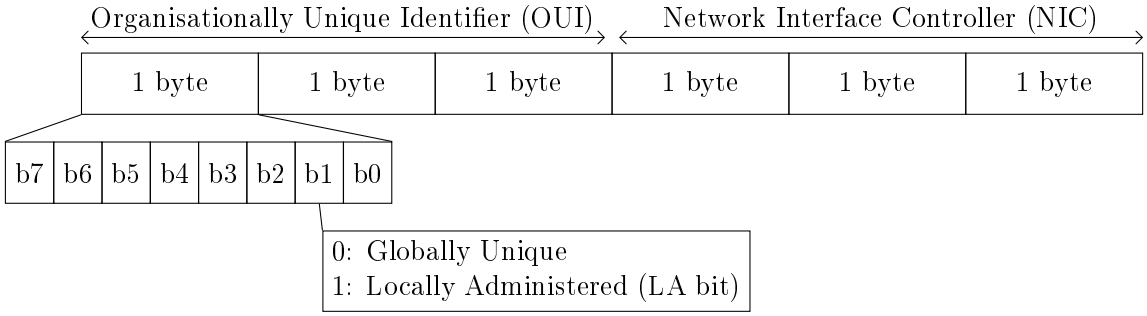


Figure I.4 – MAC address format.

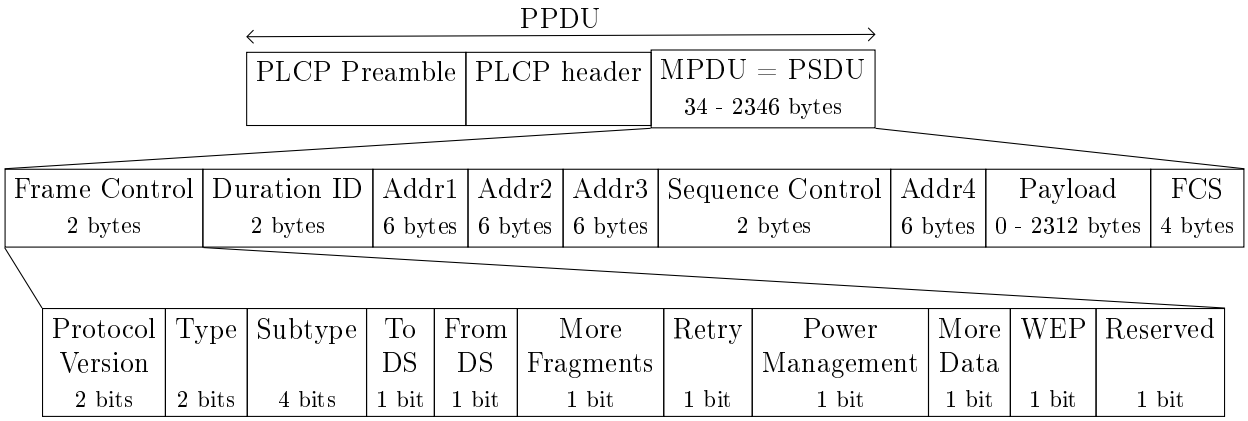


Figure I.5 – Wi-Fi frame format. The first row corresponds to the physical layer, while the second and third rows correspond to the Data Link layer. The PLCP header contains the Signal and Service fields seen in figure II.2. The sizes of the PLCP fields are not indicated because they depend on the protocol (OFDM, DSSS...). The *Addr* fields have different meaning depending on the frames type and subtype; they usually represent source and destination addresses. Depending on the frame type, the payload can include data, fixed and tagger parameters (Information Elements).

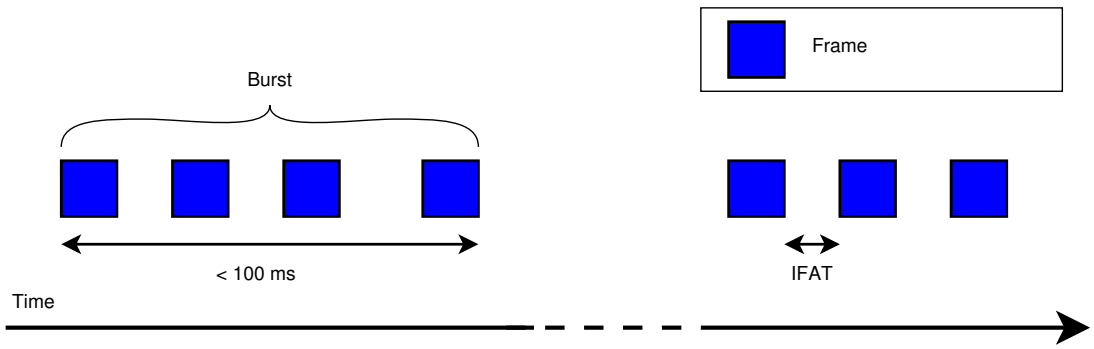


Figure I.6 – Transmission sequence of probe request frames with Inter-Frame Arrival Time (IFAT) within a burst, i.e. a group of frames sent by a device within a time window smaller than 100 ms.

for a single channel, less than 500ms in total) during an active scanning event. Each frame of the group contains a different searched network name (SSID). A burst usually contains the names of all the networks the device previously connected to (up to a certain length [16]). This list of networks is called its Preferred Network List (PNL). Such groups of frames are called *bursts* (see Figure I.6). We call the time difference between two frames Inter-Frame Arrival Time (IFAT).

Information Elements: Probe requests include information in their frame body under the form of *Information Elements* (*IEs*) [97, §8.4.2], also called *tagged parameters*, or *tags*. Most of these IEs (except SSID and supported rates) are not mandatory and are used to advertise the support of various functionalities. They are generally composed of several subfields whose size can range from one bit to several bytes. Because they are mostly

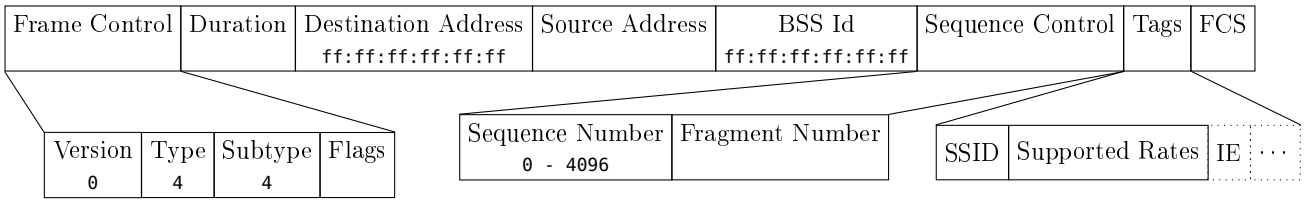


Figure I.7 – Probe request frame format. Destination (or receiver) address and BSS Id are usually left with broadcast values. Source (or transmitter) address is filled with the MAC address of the sender. SSID and supported rates are the only mandatory Information Element. SSID can be left empty to force all nearby APs to respond ("broadcast" or "null" probe). The supplementary HT Control field introduced in 802.11n for 5 GHz Wi-Fi is rarely encountered. Figure I.5 gives more details about the content of the flag field. Indicated values are fixed values for this kind of frames.

optional, these IEs are not included by all devices and the set of IEs can therefore vary from one device to another, depending on the configuration and capabilities of the device. Examples of such IEs include:

- the `HT capabilities` tag advertises capabilities for the High-Throughput 802.11n standard.
- The `WPS` tag advertises support of the Wi-Fi Protected Setup standard and includes various information, including identifiers (see section V.4).
- The `Supported Rates` tag indicated the data exchanges rates supported by the device's wireless card. This tag is completed by the `Extended Supported Rates` if more than 8 rates are supported.

Signal strength: As Wi-Fi frames are exchanged on the physical layer through radio signals, the strength at which each frame is received conveys interesting information (for instance, it indicates to end user how functional their Wi-Fi connection is). Signal strength of received Wi-Fi frames is usually represented using a measure called Received signal strength indicator (RSSI). The values range from -100 to 0 dBm[1].

I.2.2 Device fingerprinting

Fingerprinting consists in collecting enough information to identify or classify a target based on some observable features. The goal is to construct a stable pseudo-identifier of this target, if enough identifying information is available. Targets can have many forms: web browser [50], vehicles [220], passers-by using their gait [124], IoT devices [188], etc.

Fingerprinting applications are multiple: tracking [77], identity spoofing detection [55, 14],

1. decibel-milliwatts

secure localization [165], access control [197], multiple identity (cloning) detection [110] or malfunction detection [205]. In this thesis, we focus on the tracking application, rendered possible by the identifying capabilities of fingerprints.

It has been shown that all layers of the OSI model can be used to fingerprint mobile devices, because protocols have various implementations, or because they all leak identifiers [11]. Some of them tend to be more exploitable because they carry more permanent identifiers, because they require cheaper hardware to monitor, or simply because upper layers do not fit the use case. For instance, fingerprinting everyday-carried mobile devices is better done on the MAC layer because unassociated devices do not send upper layers information, and the physical layer requires more expensive hardware to sniff. For this reason, we will focus in this thesis on fingerprinting of mobile devices not associated to an AP, i.e., not producing information above the MAC layer. As many devices keep sending probe requests when associated [62] (see section IV.4), studied techniques can also be applied to associated devices. We put this strong limitation because everyday-carried devices are usually not associated to an AP at places where Wi-Fi trackers may be installed (shops, street, etc.), and most of the techniques *a fortiori* work if the phone is associated.

To give some proof that associated devices are more easily trackable, let's mention some techniques fitting this use case. Application-layer traffic contains a plethora of implicit and explicit identifiers [11]. Other protocols such as so-called *Zeroconf* ones, who aim to automatically configure devices, also broadcast plenty of information [11].

I.2.2.1 Core concepts

Features: A fingerprint is a set of information (in vector form) identifying a device. In order to be part of a fingerprint, interesting elements have to be selected and turned into features. The latter constitute individual values which can help identifying devices. For instance, in the case of web browser fingerprinting, features include the user agent, the announced language or the full list of system fonts. As detailed in related chapters (V and VI), examples of features in the case of unassociated mobile device fingerprinting are the content of the various fields, or the bursts' length.

Entropy: Entropy is a measure to quantify the amount of information brought by an element (taking discrete values [1]) in a dataset. The entropy of an element i is computed

1. When the element is a continuous variable, variance is used instead.

as follows:

$$H_i = -\sum_{j \in E_i} f_{i,j} * \log_2 f_{i,j} \tag{I.1}$$

where E_i is the domain of possible values for element i and $f_{i,j}$ is the frequency of the value j for the element i in the dataset.

Entropy's unit is bit of information (or bit of entropy). n bits of information bring enough information to distinguish up to 2^n individuals. For instance, a fingerprinting technique bringing 7 bits of information will be able to uniquely identify, on average, one device in a population of a maximum of 128 devices.

All features have a different level of entropy, and are therefore more or less useful to form a fingerprint. Danev et al. list properties that features must fulfill to be good candidates for a fingerprint [40]:

— universality: the feature should exist for every device,
— permanence: the feature should not change over time,
— collectability: the feature should be extractable for every device.

Moreover, the whole set of features should identify devices individually, at best.

Unique fingerprints: A "perfect" (unique) fingerprint would be a set of features which identifies a device uniquely, i.e., only this device possesses these values for these features in a given dataset. The biggest the entropy, the more devices in a dataset will have such a unique signature (set of features).

Classification: When a fingerprint does not contain enough information to uniquely identify targets, one may resort to classification techniques. The goal of classification it to associate *labels* to a set of targets, i.e., group targets according to certain criteria. For instance, a device could be associated to its Operating System, or to a version of its Wi-Fi card's driver.

Once data about a target is turned into a vector form representing the features, the classifier estimates which label can be applied to the tested vectors, i.e., decides to which previously known target each piece of data can be linked to, if any.

To detail more, classification techniques aim to link vectors (sets of features) to labels (for instance, a device identifier or model). They can be either supervised or unsupervised (clustering). In the former case, a classifier must be trained against a labeled dataset, i.e., the classifier must learn possible features values of the targets in order to be able to

recognize these targets later. In the latter case, the classifier does not have access to a labeled training set. It groups vectors using various techniques often leveraging knowledge on the dataset, such as the number of labels or statistics property of the dataset. A plethora of mathematical methods exist to perform these two kinds of classification, each having their own interest and drawbacks. Interested readers may refer to the very clear survey on supervised learning by Kotsiantis et al. [117].

A good example to understand device fingerprinting is the Panopticlick [1] tool by Eckersley, designed to make web browser fingerprinting understandable by the general public [50]. The tool gathers a dozen collectable features and displays their entropy to the users, along with the browser's global uniqueness.

I.2.3 Ubiquitous computing

Ubiquitous computing, more commonly called "the Internet of Things", designates the recent proliferation of computing devices present "everywhere", as opposed to desktop computing.

These often small devices are everyday objects that did not possess advanced computing and network capabilities in the past, such as light bulbs, thermostats, or most importantly, mobile devices such as fitness trackers and smartphones.

A core element of ubiquitous computing is the strong network connectivity of its components. Wi-Fi is one of the wireless communication standards that have been adopted in this context. One of the reasons of its success was its already wide deployment for common home networks for desktop computers, which allow an easy integration of these small computing objects to users' networks.

As many of these devices are Internet-connected, ubiquitous computing introduces new privacy issues [2]. For instance, mass surveillance is made possible by the fact that some of these objects (smartphones, fitness trackers...) are permanently carried by their users.

Several works already focused on these problems. As early as 2007, Saponas et al. noticed that devices which were not yet called "smart", already caused privacy issues. Notably, fitness sport kits already made tracking possible by leaking a personal identifier [178].

1. `https://panopticlick.eff.org/`, consulted on 2017.06.22
2. Not to mention the security issues.

Similarly, Aura et al. studied information leaked by mobile devices when associated on potentially open networks [11]. They found out that many protocols, such as DNS, Kerberos, DHCP and many others leaked personal identifiers. A more recent study found similar results, such as applications leaking IMEI and IMSI in HTTP traffic [103].

Smartphones are certainly the most spread and best representative ubiquitous computing device. They are permanently carried by their owners and possess many network interfaces and sensors. Besides, they carry a plethora of personal information: location history, pictures, contact list, etc. These factors generate many privacy issues that lead to a number of publications. To name a few, Spensky et al. give a holistic view of which component of a mobile device can access which private data [192]. Backes et al. build a mobile application to statically spot privacy-violating information flows [12]. Achara et al. showed that the list of installed applications can be used to build a unique device identifier in 99% of the cases [2].

I.2.4 Privacy

Privacy is a large field spreading from formal models to applied attacks, e.g. cryptanalysis. The concept is difficult to define, as different cultures or even individuals have different opinions on what is personal or sensitive information. In the end of the XIX$^{\text{th}}$ century, an influential article defined it as "the right to be let alone" [207]. Nowadays, privacy is more generally defined as the ability for individuals to choose which personal information they make accessible to which other people [211]. Other theories of privacy exist, defining it for instance as "accessibility to others", "a way to temporarily escape competition in a capitalistic society" or "a protection against over-exploitation" [184, 7, 65]. The right to privacy is stated in the Universal Declaration of Human Rights [10]: "No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honor and reputation.".

In Europe, the general population gained interest in privacy issues after the creation and exploitation of public records by the Nazis before and during the Second World War to track Jewish people. This history raised concern in the population about filing of individuals. This ultimately gave birth to privacy-defending national independent institutions (data protection authorities), such as the *Commission nationale de l'informatique et des libertés* (CNIL) in France, the Federal Commissioner for Data Protection and Freedom of Information in Germany, or the Information Commissioner's Office in the United King-

dom. On the European scale, these institutions are grouped in the Article 29 Working Party (Art. 29 WP)[1].

But why is privacy even important in a society? On a global scale, having personal information about a group of individuals is a form of power. It indeed allows one to exert targeted pressure on selected individuals, which can be used to unlawfully gain influence. As a result, it is essential for a correct separation of powers to prevent entities to have such a power, or to make sure that sufficient countervailing power exists. Moreover, individuals under permanent surveillance will modify their behaviour towards increased obedience, which leads to a normalization of behaviours and thoughts, as developed by Foucault [60].

Historically, privacy was always linked to new technologies. The "right to be let alone" formula was stated as a reaction to the apparition of new photographic technologies and their use by the gossip press. Privacy's focus moved towards personal data in the end of the 20th century, when processing and storage of data became cheap [198].

Nowadays, the privacy threat slowly expands to the physical world, because of the spread of everyday-carried mobile devices continuously emitting trackable signals. Wi-Fi-tracking systems primarily collect mobility data on individuals. Such data is very sensitive in terms of privacy. The uniqueness of mobility traces is so high that only four spatio-temporal points are sufficient to identify 95% of individuals in a population of 1.5 million individuals [44]. In other words, knowing a few locations of an individual can be sufficient to identify them uniquely. Location information is not only sensitive because it gives information about the presence of a person at a given place at a time, but also because of the semantics of the visited locations, i.e., which kind of places were visited, and in which order [4]. One may not want this kind of information to be made public, for a wide range of reasons. It may reveal information about their consumption (which shops do they go to?), personality (what places do they visit for leisure?), sexuality (do they frequent gay bars?), relationships (which persons do they meet?), specific sensitive behaviours, etc.

I.2.5 Physical tracking

One possible application of Wi-Fi-enabled devices fingerprinting is physical tracking. With the advent of ubiquitous computing, such devices are carried all the time by most individuals, which makes it possible to indirectly track their owner in the physical world.

1. Full members list can be retrieved here: `http://ec.europa.eu/justice/data-protection/article-29/structure/members/index_en.htm`(consulted on 2017.05.29)

In section II.4, we will describe physical tracking and its applications using Wi-Fi, Bluetooth, and various other radio technologies.

I.3 Used Datasets

Due to strong legislation on personal data around the world, obtaining datasets of real-world probe requests is difficult [1]. The best option is to use the pseudonymized Sapienza dataset published by Barbera et al. [15]. This dataset was collected in Rome in early 2013 and covers different use cases, including a train station, a mall, a university, and a political meeting. A total of more than 8 millions of probe requests sent by over 160 000 devices form the dataset. However, because of its old age, this dataset contains barely any random MAC address: 0.2% of the MAC addresses have their LA bit set to 1 (see section IV.3). As a consequence, we cannot use it as is as a real-world dataset for our randomization-related works.

More recently, Robyns et al. published several datasets recorded in late 2015 and early 2016. The `Belgium` dataset contains timing information about 28 millions of management frames, among which 200 000 are probe requests. In this dataset, 66% of the MAC addresses are random (see section IV.3). While the related `Glimps2015` dataset does not contain any timing information, this dataset contains `libpcap`'s radiotap information, including the TSF [2] timestamp for each frame. This timestamp acts as a normal timestamp indicating the time at which the first bit of a frame reaches the wireless card, but the state of the timer may change brutally due to TSF mechanisms. We reconstructed a normal timeline by detecting and softening these brutal changes.

Besides these public datasets, we collected our own sets of data. In order to limit privacy risks when analyzing them, we restricted the captured to probe requests, which means that no network data was collected. During the Middleware conference in December 2014, we installed sensors for a couple of hours. This constitutes the `Middleware2014` dataset. The `Train station` dataset was captured around one large train station in Lyon in October 2015. The `Lab` dataset is a 5-day-long capture in October 2015 in our laboratory.

1. For readers interested in getting access to supplementary datasets, let's note that some authors worked on large private datasets and may be worth contacting to work on such datasets [209, 131, 180, 151]. The `Sigcomm2004` and `Sigcomm2008` also contain probe request traces of devices of voluntary participants, but MAC addresses are entirely anonymized and the LA bit is not maintained [182, 176]. Readers interested in anonymizing a dataset of network traces may look at the tcpmkpub tool [163].
2. Time Synchronization Function

Name	Time	Place	Situation	MAC addr.	probe requests	Source
Sapienza	2013.02 - 2013.05	Rome	mix	160 000	8 000 000	[15]
Middleware2014	2014.12	Bordeaux	hotspot	900	140 000	personal
Lab	2015.10	Lyon	local AP	1 300	120 000	personal
Train station	2015.10 - 2015.11	Lyon	street	9 700	110 000	personal
Glimps2015	2015.12	Belgium	local AP	83 000	120 000	[175]
Belgium	2016.01 - 2016.02	Belgium	hotspot	3 700	200 000	[175]
Martin	2015.01 - 2016.12	Maryland	street	2 600 000	66 000 000	[131]
Madeira	2015.12 - 2017.06	Madeira	hotspot	13 000 000	300 000 000	not public

Figure I.8 – Used datasets

All of these datasets (including our own ones) are anonymized using similar methods: replacing the NIC bytes of the MAC addresses with pseudonyms, and removing SSIDs or replacing them with pseudonyms. As discussed in section V.4, we identified an issue with the Sapienza dataset: the WPS IE is not anonymized. We informed its authors about the issue.

Other datasets are:

— The Martin is a huge collection of several dozens of millions of probe requests over a period of almost two years. Even though we did not get access to this dataset, we list it here because we use its statistics in section IV.3.

— The Madeira dataset is a long-term permanent capture running in many cities of the Madeira Island since December 2015 [1]. We got access to the database table containing MAC addresses details (OUI and a hashed version of the address).

These datasets are summarized in figure I.8. We distinguish the situation of these datasets into several cases:

— *hotspot*: the recording matches what one would obtain by recording traffic at a public hotspot, i.e., users stay for some time (from minutes to hours) and the turn-over is high,

— *local AP*: what one would obtain by recording traffic at a home AP: users stay for a long time (possibly the whole capture), and the turn-over is low,

— *street*: what one would obtain by recording traffic in a street or along a road: most users are seen very briefly and the turn-over is high,

— *mix*: a combination of above cases. [2]

To give more details, we compare the fraction of vendors (according to OUIs) in the different datasets we have access to (in terms of number of non-random MAC addresses, identified by their LA bit) in figure I.9. This gives an overview of the distribution of

1. http://beanstalk.m-iti.org/category/tracking-platform/, consulted on 2017.07.18
2. The Sapienza contains all these cases in different capture files.

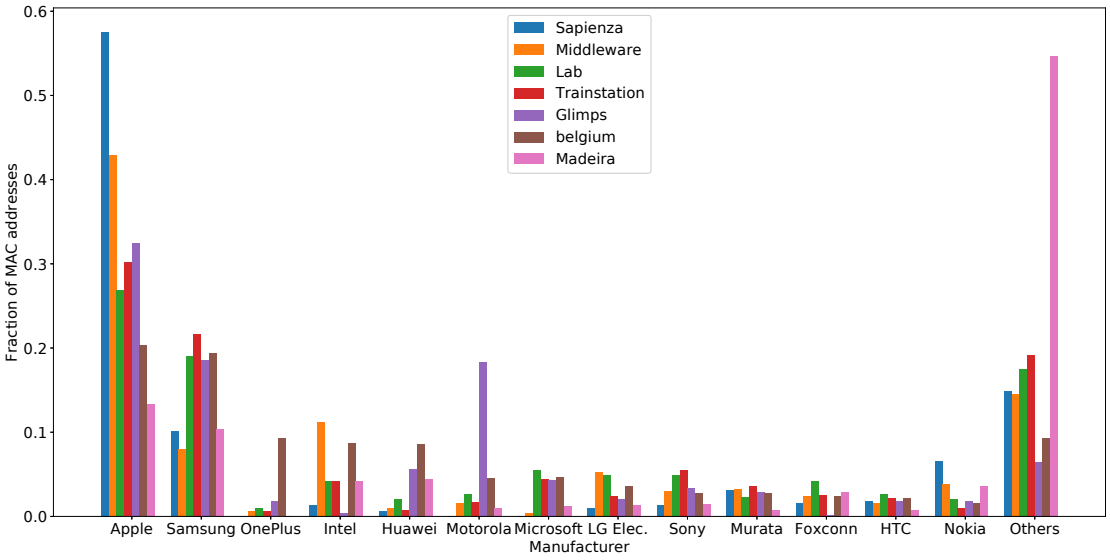


Figure I.9 – Fraction of non-random MAC addresses belonging to most-spread manufacturers. Represented manufacturers are those for which at least one dataset had 3% of its MAC addresses belonging to it.

devices' vendors in the different datasets. Apple arrives first in all datasets, as the company manufactures all iOS devices, unlike Google and its Android OS. Large differences of appearance of a given vendor across several datasets can be attributed to their dates. Lower scores in all popular manufacturers for the `Madeira` can be explained by a hardly quantifiable number of random addresses not setting the LA bit to 1, and using already-attributed OUIs [1]. We're unsure about this source of incoherent addresses: it could be explained by some unidentified device models performing MAC address randomization in a yet-unobserved manner, or by an attacker deliberately spamming the systems with fake addresses [2].

I.4 Document structure

We start with a state of the art in chapter II. We follow and discuss the current spread of commercial Wi-Fi-tracking systems in chapter III. In chapter IV, we study the deployment of MAC address randomization, including its spread in the wild, and example of

1. The large number of such addresses using OUIs of small companies not manufacturing any mobile devices tends to make us believe that these are not genuine OUIs.

2. The attacker hypothesis seems more credible when we consider the fact that almost no MAC addresses in this dataset (0.05%) use both an unregistered OUI and a LA bit not set to 1.

implementations in recent devices. Chapters V and VI present two attacks weakening the effectiveness of MAC address randomization. Chapter VII presents a Wi-Fi tracking system that we developed for experiments and raising public awareness about the privacy issues of Wi-Fi tracking. Chapter VIII concludes the document.

Chapter II

State of the Art

This section details the state of the art in the different domains and techniques approached in this thesis. We first introduce scanning behaviours in section II.1. Then, we explain why and how this scanning becomes a privacy issue in section II.2. We move on to reviewing the various device fingerprinting techniques in section II.3. Section II.4 presents physical tracking techniques along with their applications. Section II.5 then discusses some possible countermeasures to privacy issues leveraged by Wi-Fi-based tracking. We finish with a short mention of some artists who tackled the same problem as we do with a fully different approach in section II.6.

II.1 Active scanning behaviours

Since Probe Requests emitted by mobile devices are at the center of this thesis, we start by reviewing works that have studied how probe requests are generated by mobile phones, and noted particularities in the probing behaviours of different devices.

Conditions of emission

It was first noted by Franklin et al. that devices keep sending probe requests even when associated to an AP [62].

Freudiger listed several factors impacting probing behaviour: screen state, charging state, airplane mode, Wi-Fi setting screen open, Bluetooth activation, and proximity of a known

network. He extensively tested the influence of the number of known networks (length of the PNL). He studied probing behaviours of several popular devices of different operating systems using multiple antennas, calculating the average number of sent probe requests for each of them [63].

Activating a device's screen triggers a burst of probe requests for many devices. Jamil et al. exploited this to estimate the state of a phone's screen depending on its probe requests [104].

Frequency of emission

Lim et al. studied probing patterns in the `Sapienza` dataset, and found different probing frequency depending on the context. They noted that some devices (using Android below 2.3.7) almost never send any probe request. They also calculated the average number of devices sending probe requests depending on the size of a temporal window. They found that, for instance, 85% of devices have an 80% probability of sending probe requests in a 3-minute timeframe [126].

Devices operating on battery power may probe less often than when connected to a charger [204].

Abedi et al. compared probing rates of both Wi-Fi and Bluetooth, and concluded that Wi-Fi devices broadcast theoretically 10 times and empirically 8 times more MAC addresses than Bluetooth devices, on average [1].

Channels

Waltari and Kangasharju made some interesting discoveries while studying channel-changing patterns of probe requests [204]. They noticed some interesting probe request behaviours, such the fact that some devices avoid channels forbidden on the U.S. environment even while running in a another country, or the fact that all devices do not send exactly a number of probe requests which is a multiple of the number of channels. Some devices start a burst by probing on channel 1 while others start on channel 6. All tested devices switch channel by either incrementing or decrementing the channel number. They also noticed some randomness in the timing pattern of some devices. While the reason for this is left to identify, it might be an attempt to deal with the issue of fingerprinting using timing of probe requests, introduced in chapter VI.

While studying probing behaviours on multiple channels, Corbett et al. find channel

differences. Notably, they conclude that, out of channels 1 to 5, channel 4 is the most stable and regular channel in terms of probing behaviours [34].

Content

Gentry and Pennarun noticed several precise behaviours: for many devices, the "current channel" Information Element is only added in directed probe requests, OS updates can change IEs and the IE fingerprint of a given device is different in the 2.4 GHz and 5 GHz bands [72].

Barbera et al., in a study of the distribution of Preferred Network List's lengths, noted that many vendors limit the number of sent probe requests to 16. As a result, few devices send more than 16 different SSIDs [16].

II.2 Probe requests as a source of privacy breach

Wi-Fi probe requests are broadcast at high frequency by many devices, ranging from one every few seconds to every few minutes [63]. As a source of information constantly leaked by devices, they can be used to infer more information on these devices and their owner. For instance, Jamil et al. detected user behaviour regarding their device's usage based on the frequency of sent probe requests [104]. Redondi et al. presented a method to determine whether a device sending probe requests is a smartphone or a laptop [172]. Socio-economical status can also be extracted from vendor ID embedded in the MAC address [16]. On a security perspective, Robyns et al. used databases of collected probe requests to infer statistics about device models in order to estimate the number of devices affected by a security vulnerability [174].

Many devices still add SSIDs (network names) to their probe requests for various reasons. These SSID leaks are an important source of privacy breach. Semantics of these names can be exploited to reveal information about a person or a group of persons which can be sensitive, such as their home address, social links [37, 77] or the global social structure of a crowd [16]. Network names can be as sensitive as "Juvenile Detention Classroom" [162]. Di Luzio et al. used SSIDs contained in probe requests to infer the origin location of participants of a conference, using the public database WiGLE [1] to get geographic location out of SSID names [49].

1. `https://wigle.net/`, consulted on 2017.08.02

II.3 Devices fingerprinting

Devices fingerprinting was presented in the introduction. This section will present subcases of this concept: first, physical components fingerprinting. Then, Wi-Fi-based techniques will be detailed.

II.3.1 Physical components fingerprinting

Due to slight variations during the manufacturing process along with their aging, components of a device possess singularities which make them all behave in a slightly different way. While this usually does not disrupt normal functioning of the component, this can be exploited to fingerprint the latter.

In this section, we present Radio-Frequency (RF) fingerprinting, which exploits differences in the manufacture of radio components, before detailing clock skew fingerprinting, a technique to fingerprint a device using the state of its clock.

To be complete, we have to briefly mention that other components of a device can be fingerprinted. For instance, Das et al. abuse imperfections in microphones and speakers to fingerprint devices using audio signals [42].

II.3.1.1 Radio-frequency fingerprinting

Exploiting the physical characteristics of radio emitters is an old technique, which can be traced back as early as World War II. It was also used during the Vietnam War, when Morse code signals were fingerprinted so as to determine whether they came from an allied or enemy source [107]. Radio operators analyzed signal frequency and amplitude in order to determine similarities and derive a potential identifier. The technique was not entirely reliable and was used in conjunction with other identifying tools.

The historical approach to Radio-Frequency (RF) fingerprinting is transient analysis. Transient is the period during which a radio transmitter activates before sending payload signal. Signal sent during that period depends on the transmitter, and can thus be used as a fingerprint [195]. A plethora of new approaches have been tested to perform RF fingerprinting. Other signal parts can be exploited, such as near-transient regions, or part

of the data itself. Different mathematical tools can extract features out of waveforms [40]. Apart from the signal's waveforms, the modulation domain can also be used [23].

Precise components of a RF device can be targeted, such as the power amplifier and the digital-to-analog converter [167]. The clock can also be a target, as presented in the next section.

RF fingerprinting has recently been studied to specifically fingerprint mobile devices. For instance, Kaplan et al. published a patent to fingerprint mobile phones using a frequency-based analysis of the analog signal [110]. Brik et al. showed that the network card of a wireless device could be fingerprinted [23]. However, a common factor to these works is the fact that used hardware is extremely expensive, compared to the cost of a network card necessary to fingerprint radio emitters. For instance, Brik et al. use an Agilent 89641S vector signal analyzer, which costs tens of thousands of dollars [23].

The identity-spoofing detection capability of fingerprinting can be used to detect MAC address spoofing. This has been done using RF fingerprinting, for instance by Hall et al. [83]. Closely related, pseudonym use (using various identifiers) can be broken. A tool by Brucker uses carrier frequency offset to break MAC address randomization as one of its examples [1]. Carrier frequency offset, another possible feature of RF fingerprinting, designates difference of frequency between a sender and a receiver carrier. It is caused by imperfections in both carrier's oscillator, and the Doppler effect if they are moving.

RF fingerprinting is sometimes called physical-layer fingerprinting. Interested readers may refer to the complete survey of RF fingerprinting by Danev et al. [40].

II.3.1.2 Clock skew

Most devices possess an internal clock that is used to keep track of passing time. Due to hardware imperfections, these clocks tend to slowly derive from the globally accepted universal time in a hardly-predictable manner. Variations reach a magnitude of several parts-per-million (ppm), i.e., 10^{-6} seconds.

Kohno et al. were the first one to use clock skew to fingerprint devices [113]. They introduced the idea of fingerprinting a *physical device* (device hardware) instead of an operating system or a driver, using the clock skew as an identifier. They showed that

1. `https://rftap.github.io/blog/2016/09/01/rftap-wifi.html`, consulted on 2017.07.02

a device's clock skew derives slowly over time (0 to 4 ppm in a 24-hour period), while variations across devices are significantly high enough to allow fingerprinting (a range of about 600 ppm). Their technique works even if the target and the attacker are separated by several routers, or located in distant cities.

For a clock skew fingerprinting attack to work, one needs access to some kind of timestamp from the target device. Several timestamp sources were studied in the literature:

— ICMP [RFC792] timestamps [113, 35, 128],
— TCP [RFC1323] timestamps [113, 150],
— Timing Synchronization Function (TSF) [97, §10.1] timestamps [105].

The first two timestamp sources are based on protocols from the network layer or above. TSF timestamp are only sent by APs, in beacons and probe responses. As a consequence, an unassociated mobile device does not produce any of them. While mobile devices have been shown to be fingerprintable using clock skews [35], current state of the art does not allow one to fingerprint unassociated mobile devices using clock skews alone. We did not find any way to render this possible. Desmond et al. attempted to estimate clock skews from IFATs, but it seems the difference of magnitude between both values is too high [48]. A potential solution to this issue was found by Huang et al. [93]. Using the Bluetooth protocol, they showed that fingerprinting unpaired devices is possible using the temporal characteristics of their frequency hopping patterns. However, this technique will only work if target devices have their Bluetooth interface activated, and are paired to a Bluetooth LAN.

II.3.2 Wi-Fi-based fingerprinting

This section presents the core of our topic: Wi-Fi-based fingerprinting. We first present a technique based on the physical layer, then compare the numerous MAC-layer-based works.

II.3.2.1 Wi-Fi-based physical layer fingerprinting: scrambler seed

Fingerprinting can be based on data fields of the physical layer. For instance, a Wi-Fi specific field, the scrambler seed, can be used to identify an 802.11p device.

One of the modulation technique used for the 802.11 physical layer is called Orthogonal

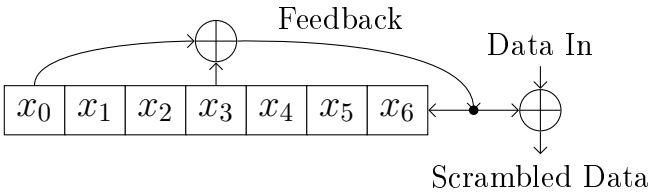


Figure II.1 – The scrambler used in 802.11 frames.

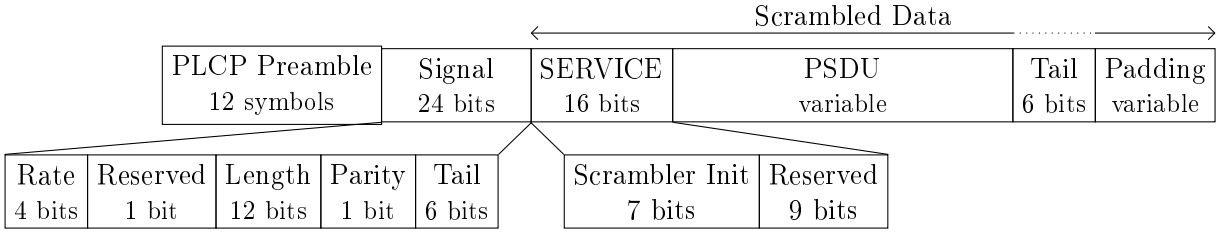


Figure II.2 – OFDM frames format.

Frequency Division Multiplexing (OFDM). In order to mitigate transmission errors by removing repetitive patterns in transmitted data, this data is scrambled. To do so, data is XORed with a bit sequence generated by a Linear Feedback Shift Register (LFSR), as represented in figure II.1. Figure II.2 shows the layout of OFDM-encoded frames. The seed of the scrambler can be reconstructed from the Scrambler Init field.

Bloessl et al. were the first to introduce the idea of tracking devices using the scrambler seed. They did not work on mobile device, but on network cards supporting 802.11p, the standard for Wireless Access in Vehicular Environments (WAVE) [19]. They showed that the scrambler seed field is changing in a predictable manner in studied implementations. This can be leveraged to link consecutive frames sent by a device, even if these frame do not contain any identifier of that device.

While not detailed in this manuscript, we extended this previous work by showing that previous attack worked on network cards used by popular mobile devices [200]. This work was itself extended by Vo-Huu et al., who used scrambler seed in conjunction with carrier frequency offset and other features to show the feasibility of tracking Wi-Fi-enabled devices using commodity software-defined radio platforms [203].

II.3.2.2 Wi-Fi-based MAC-layer fingerprinting

In this section, we first present implementation fingerprinting, then focus on the more precise device-specific techniques.

Implementation fingerprinting: Fingerprinting a unique device is a tough question, as it requires finding a behaviour which varies across all devices. It is often easier to find differences in the implementations of a mechanism, i.e., fingerprinting a device's "model", or more precisely, the combination of its operating system, its wireless card's driver and chipset (whichever are involved). For instance, drivers can be fingerprinted based on their implementation of the exchange change rate adaptation algorithm [33, 149], their reaction to malformed frames [20], or the channel-changing patterns of probe requests [204]. Some authors even limit themselves to distinguishing broad classes of devices, such as smartphones and laptops [172].

A pioneering work in implementation fingerprinting is the 2006 article by Gopinath et al. who studied the implementation of the random backoff algorithm used for CSMA/CA. They noticed that norms weren't enforced in all implementations, and that deviations to these norms could be used to fingerprint the network card driver of the device [76]. They also studied the enforcement of other parts of the 802.11 standard:

— the respect of the *duration* field and the behaviour when facing incorrect values,
— how the transmission rate changes depending on network load,
— wait time before attempting a reauthentication after receiving a deauthentication frame.

The duration field was also studied in other works [25, 51]; but as this field rarely contains any non-zero value for unassociated devices, this technique has little chance to work in such a case.

Another pioneering work, closer to our own work, is the one by Franklin et al., which showed that the temporal characteristics of probe requests (i.e., IFAT values) can be exploited to build a fingerprint of a device's network card driver [62]. They built devices signatures by calculating the distribution of IFATs. We make use of this technique in chapter VI. Closely, Gao et al. fingerprinted APs by studying the delay between packets that they retransmitted (Packets Inter-Arrival Time) [70].

Ellch tested several implementation details to fingerprint drivers: respect of the RTS/CTS window, behaviour when AP changes the source address in an association reply (association redirection) and using the duration field as a feature [51].

Martin et al. tried to extract more information from a MAC address that simple vendor name. The found out that MAC address allocation are usually contiguously assigned to distinct models, but no assignment standard exist between constructors. They also observe a complexity in OUI allocation, with some OUIs being shared by different vendors

and conversely, some device models spanning on several OUIs [133].

Corbett et al. studied implementations of active scanning in terms of temporal behaviour, on multiple channels. They apply a spectral analysis, analyzing timing of probe requests with a different approach than the one we use in chapter VI. They transform IFATs into spectral signatures using power spectral densities [34].

Device fingerprinting: Some works try to overcome the limitation of fingerprinting to implementations, and find some device-specific techniques.

Some of these techniques are only functional for associated devices. Aura et al. listed many protocols leaking identifiers on a local network, mainly because of service discovery [11]. Pang et al. tracked devices associated to a public hotspot using implicit identifiers within their traffic: network destinations, SSIDs in probe requests, size of broadcast packets and options of the MAC header (fields such as "more fragments", "retry", "power management", and "order") [161]. Gentry and Pennarun fingerprinted chipsets using content of both probe requests and association requests frames. To distinguish between devices of the same model, they added the hostname broadcast in DHCP requests or DNS-Service Discovery requests, which requires devices to be associated [72]. Siby et al. classified all kinds of wireless devices monitored using different interfaces, using sent-to-received ratio, along with sent and received volumes as classifying features. They, however, ignored management frames, rendering their technique only useful for associated devices [188].

Desmond et al. first fingerprinted devices using timing of their probe request, considering both IFAT and clock skews estimated out of them. Their technique required the device's traffic to be recorded for one to 2 hours for the technique to be successful, despite a low number of devices. Due to the large difference of magnitude between IFAT and clock skew values, it is not clear whether the latter truly have an impact [48].

Neumann et al. studied passive fingerprinting techniques of Wi-Fi-enabled devices which can be performed with a simple network card [152]. Among the different Wi-Fi-related parameters studied, they figured out that the more efficient ones at classifying devices are network transmission time and IFATs. Other studied parameters are transmission rates switching behaviour, frame size, medium access time (which depends on the random backoff) and transmission pattern of probe requests. Using these parameters, they reach a unique identification rate of more than 50% (with 10% of false positives) in a dataset gathered at a conference. This work is close to what we do in chapter VI, but does not focus on defeating the MAC address randomization technique. We go further than them

on two points: we limit ourselves to probe request frames and test several algorithms for frames grouping. Interestingly, they mention IFAT of "Data null function" frames, which are only sent by associated devices. Such frames could be used to counterbalance the fact that associated devices often send less probe requests than unassociated ones.

Information Elements: Information Elements were briefly mentioned by several authors: Gopinath et al. focused on vendor-specific tags [76] and Pang et al. mentioned that they could help constituting implicit identifiers, without further proof [161]. We develop this in chapter V. Following our article on fingerprinting devices using the content of their probe requests, Robyns et al. used the same IE-related fingerprinting technique. Moreover, they studied which frames used by unassociated devices could be leveraged to increase devices' trackability using active attacks (see section II.4.1.1) [175]. Gentry and Pannarun's work (presented above) [72] includes the calculation of the IE signature (fingerprint) of over a thousand device models, which they later published [1].

Table II.1 summarizes characteristics of these previously presented articles.

Xu et al. performed a full survey of fingerprinting in wireless networks in 2016 [218].

II.4 Physical tracking

Whether it is through websites or mobile applications, user tracking is a common thing in the digital world. This practice of monitoring users' activity for analytics or profiling purposes has recently been extended to the physical world, where radio and video technologies now allow to accurately detect, recognize and categorize human activities [155, 151, 67].

We define physical tracking as the activity consisting in following the whereabouts of a person along time in the physical world. Tracking can have diverse applications, from locating war opponents (see examples in section II.4.2.4) to performing location analytics of customers in a store (see section II.4.2.1). Tracking is an issue when it is performed without the consent of the targeted person. It then consists, as such, in a violation of the person's privacy.

This section will present various tracking technologies, focusing mainly on Wi-Fi. Then, we discuss some real-world evidences of physical tracking.

1. `https://github.com/NetworkDeviceTaxonomy/wifi_taxonomy`, consulted on 2017.08.04

Article	Passive	Unass.	STA	Dist. dev	Used technique
[76]		✓	✓		Norm respect
[62]	✓	✓	✓		Probe requests timing
[32]	✓		✓		Rate switching
[25]	✓		✓		duration field
[51]		✓	✓		RTS/CTS window, assoc. redir.
	✓		✓		Duration field
[161]	✓		✓	✓	Implicit identifiers
[11]	✓		✓	✓	Leaked identifiers
[20]		✓	✓		Malformed frames
[34]	✓	✓	✓		Spectral analysis (IFAT)
[48]	✓	✓	✓	✓	IFAT + clock skew
[70]					delay between packets
[149]			✓		Rate adaptation
[152]	✓	✓	✓	✓	Many features
[204]	✓	✓	✓		Multi-canal probe requests analysis
[72]	✓		✓	✓	Probe requests, association frames & DHCP hostname
[133]	✓	✓	✓		OUI decomposition
[172]	✓	✓	✓		Probing frequency and RSSI
[188]	✓		✓	✓	Traffic volume analysis
[175]		✓	✓	✓	IE + active attacks

Table II.1 – Summary of the different Wi-Fi-based fingerprinting works (sorted by year). Second column indicates whether the method is passive. Third one indicates whether the method can be applied on devices unassociated to an AP. Next one indicates whether the attack can be used to fingerprint stations (as opposed to AP-only), and next column indicates whether the method can distinguish individual devices instead of fingerprinting a shared characteristic (such as the driver's model).

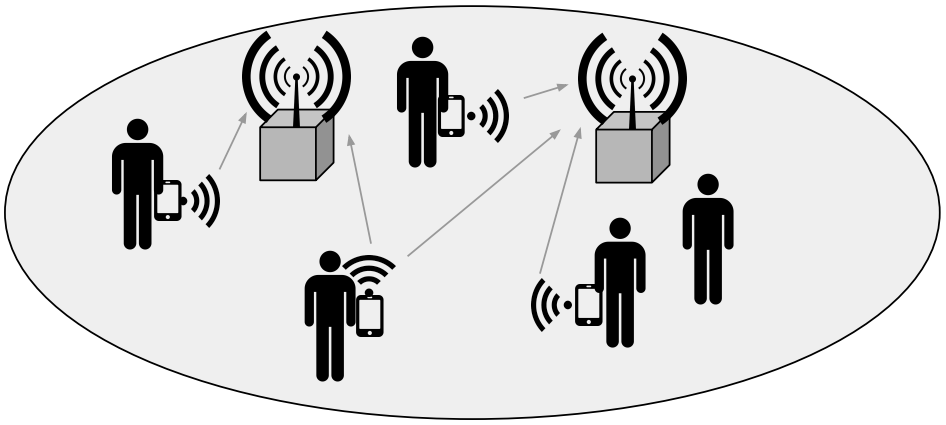


Figure II.3 – A Wi-Fi tracking system.

II.4.1 Technologies

II.4.1.1 Wi-Fi-based tracking

Radio-based physical tracking relies on sniffers that collect identifiers contained in messages emitted by radio-enabled devices [63]. These identifiers are used to detect users' presence and estimate their mobility. Because Wi-Fi is included in many portable devices and relatively easy to sniff, it is the main radio technology used in the physical tracking industry.

Deploying passive sensors, one is able to collect frames of all nearby activated devices. These sensors can be cheap, as many commercial off-the-shelf Wi-Fi cards can be turned into so-called *monitor mode*, which allows them to collect frames even if they are not addressed to them [1]. Because Wi-Fi-enabled devices permanently perform active scanning, most of them can be detected by a passive tracking system (see discussion in the introduction). Figure II.3 schematizes such a Wi-Fi tracking system.

Many tracking systems have been developed along the years. Musa and Eriksson built a Wi-Fi tracking system able to estimate user trajectories, and using active attacks to increase devices' trackability [151]. Cuthbert and Wilkinson built an open-source distributed tracking system called Snoopy [2], able to perform some active attacks [214]. O'Connor built another Wi-Fi tracking system based on cheap hardware [159].

One may argue that tracking a device is not the same as tracking an individual. Cunche introduced two tracking techniques based on Wi-Fi, to retrieve the link between a MAC address and the person owning the related device. The first one exploits rare SSIDs registered in the device's PNL. The attacker guesses which SSIDs may be linked to a person (using their semantics or the physical location of the related network), and sends these SSIDs in beacons so that the target device replies, giving its MAC address. In the second one, the attacker follows a person until only one MAC address has been seen during the whole session, hinting that this address is the target's device's one [36]. Wilkinson proposed related "use cases" using that link retrieval, e.g., finding a spy, a celebrity or a criminal [214].

Similarly, we introduced an attack to make the link between a MAC address and an

1. To be precise, the monitor mode prevents the card from dropping frames whose destination address is not either their own MAC address, a broadcast or a multicast address.
2. `https://github.com/sensepost/Snoopy`, consulted on 2017.07.21

account on a social media platform. To reach this goal, we created a fake environment of APs to trick a device's positioning system, so that the user posts information on the platform associated with a fake, identifying location [137].

Indoor tracking: One application of tracking is indoor tracking, i.e., collecting information so as to reconstruct the trajectory of a target within a small place. Obtaining precise location information using Wi-Fi in a small place is a difficult task. Estimating the location of a device using signal strength isn't reliable, as the RSSI-to-distance correlation is weak [177, 66, 41]. It is indeed affected by factors such as propagation path or whether the device is shielded (e.g., placed in a pocket) [209]. Weppner et al. proposed a solution using several sensors and filtering noisy recordings, achieving a precision of around 10 meters [209]. A better solution is to the more reliable Channel State Information (CSI) instead. A tool by Halperin et al. makes it possible to access it using an off-the-shelf network card [84]. CSI is more precise measurement than RSSI because it contains channel information for each subcarrier instead of a global measurement. This measure has been used to fingerprint a person [217] using their gait [124] or behaviour inside a shop [222], among others. We had to face the difficulty of performing indoor tracking for some installations of the Wombat tool (described in chapter VII).

Active attacks: While we so far mostly considered passive tracking attacks, some active attacks can increase the probability to track nearby devices by forcing them to reply to queries.

Cunche et al. [36] proposed a method to force devices to reveal their presence using commonly used SSIDs. Using most-spread SSIDs is sufficient to cover a large fraction of devices [37, 200]. In a later work, we studied another attack alongside this one. This second technique relied on the new 802.11u standard, commonly referred to as Hotspot 2.0, where we showed that Linux and Windows send Access Network Query Protocol (ANQP) requests using their real MAC address [200].

Musa et al. introduced a technique using forged RTS frames to force a device to respond with a CTS frame, revealing its presence in the process [151]. However, the paper announces disappointing results. Our own tests confirm that most devices only reply to these forged frames if the latter are received within a short period of time after the emission of a frame. Thus, this technique appears inefficient to enhance tracking capabilities. They also consider setting the tracker in a man-in-the-middle position by emulating an AP using a popular SSID, to leverage the frequent emission of null frames from the associated device to increase its trackability.

Similarly, Martin et al. performed tests to attempt to defeat randomization [131]. Studying some control and management frames, they found that only RTS frames trigger a response when sent to a device in an unassociated and unauthenticated state. Worryingly, they noticed that Android devices using location services do respond to RTS frames even if their Wi-Fi interface is disabled or if they are in airplane mode. Unlike previous work and our own tests, they claim to reach a success rate of 100% with the RTS attack. Martin et al. also noted that some unstudied link-layer protocol such as Cisco Discovery Protocol (CDP) or Link Layer Discovery Protocol (LLDP) could be leveraged to collect more information on mobile devices. [133].

Robyns et al. pushed the topic further by exploring many different possibilities to enhance tracking [175]. Among others, they used action frames to trigger a response from nearby devices. Their results show that the most promising ones for active tracking are Generic Advertisement Service requests (such as ANQP queries used in the HotSpot 2.0 protocol). These frames can be broadcast, require no previous knowledge about tracked devices, and trigger an immediate response. However, support by the device is required. As a result, they appear to be a good candidate to enhance tracking.

II.4.1.2 Bluetooth-based tracking

Bluetooth is a set of standards for short-range radio transmission in the 2.4 GHz band. In this technology, LANs are constituted of a master device communicating with up to seven slaves. Devices that joined a network are *paired* (equivalent of *associated* in Wi-Fi). Due to its popularity in mobile devices, Bluetooth is often presented alongside Wi-Fi in this thesis.

Similarly to Wi-Fi, Bluetooth can be exploited for passive tracking. Bluetooth network cards also possess a MAC address, usually different from the Wi-Fi one (often one digit away [131]). Bluetooth got a major update with the release of Bluetooth 4.0, which introduced Bluetooth Low Energy (BLE), a new set of protocols based on a different stack aiming to reduce energy consumption. This version of the technology introduced randomization of MAC addresses [215]. This is later discussed in section II.5.3.1.

Unsurprisingly, we find some Bluetooth-based tracking systems very similar to the Wi-Fi-oriented ones. In fact, some of them can handle both protocols [214]. Holeman made a tool to collect Bluetooth devices' location [91] and Bugher described a procedure to make your own [24]. Albazrqaoe et al. worked on a Bluetooth-based traffic sniffing system using

2 antennas working simultaneously that defeats the channel-hopping protection against sniffing [6]. Issoufaly and Tournoux proposed a BLE-based tracking system, exploiting the observation that randomization in BLE was rarely used [101].

Bluetooth is especially popular in wearable devices. For instance many fitness trackers use Bluetooth as a way to exchange data with their owner's smartphone or computer. It was remarked as early as 2007 that these devices leaked their Bluetooth MAC address [178]. As of 2016, they still pose a wide range of privacy issues comparable to smartphones. They often transmit location information to the companies' servers, do not use MAC address randomization, lack efficient security protections, and have unclear policies with regards to resale of personal information to third parties [87, 56].

II.4.1.3 Cellular tracking

Using expensive hardware, different cellular technologies can be tracked as well.

The old GSM radio technology (2G) has been thought to prevent a passive attacker from tracking cellphones. The latter possess a permanent identifier linked to their SIM card, called IMSI. These identifiers are exchanged as infrequently as possible, i.e., upon reaching the network (on system startup). They are quickly replaced with a pseudonym called Temporary Mobile Subscriber Identity (TMSI), which will be frequently changed. The association between a SIM card and a TMSI is kept by the infrastructure in a database called the Visitor Location Register. This old example of pseudonym implementation could be taken as an example for the current deployment of MAC address randomization.

Recent development in cryptanalysis weakened the security of this technology. Cryptographic algorithms in the GSM norm (A5/1 and A5/2) are now considered broken and supposedly decryptable on-the-fly[1]. More recent cellular technologies such as UMTS[2] and LTE[3] (3G and 4G) integrate more advanced protections against attacks, such as mutual authentication for LTE.

However, jamming frequencies used by these technologies force devices to switch back to GSM as a fallback solution. This widespread attack is used by what is commonly called

1. https://www.washingtonpost.com/business/technology/by-cracking-cellphone-code-nsa-has-capacity-for-decoding-private-conversations/2013/12/13/e119b598-612f-11e3-bf45-61f69f54fc5f_story.html, consulted on 2017.06.26
2. Universal Mobile Telecommunications System
3. Long Term Evolution

IMSI catchers. They exploit the fact that the 2G protocol does not allow a station to identify the base station to place themselves in a man-in-a-middle position using a fake 2G base stations. Stations connect to this fake base station after other protocol's frequencies have been jammed. Then, they exploit the weak cryptography of the protocol to decrypt communications, and use requests to track the device's location in real time.

A current hot topic in the security of cellular infrastructure is the flaws in the GSM's communication protocol between operators called Signaling System #7 (SS7). These flaws allow an unauthenticated attack to directly perform queries in the Visitor Location Register. This can be used to obtain location information or eavesdrop communications of any device provided their IMSI or a TMSI is known [53]. As of July 2017, these security holes are still widely open [154].

To sum up, a wide range of flaws in the different cellular protocols and their implementations allow tracking of cellular devices.

II.4.1.4 Other radio-based tracking

Other radio technologies can be exploited to perform different kinds of tracking. We list the most-used ones.

Beacons: Beacons are sometimes considered a form of tracking, despite requiring strong user cooperation. They are small devices broadcasting information such as their position or other announcements. They can be used for mobile devices to get their precise indoor position. Applications also react when some beacons are seen, for instance to display location-aware advertisement. With this technology, no signal emitted by devices is collected. Tracking can only be performed on the device-side, for instance by an application sending this information on a remote server.

Radio-frequency identification (RFID): RFID is a set of standards for short to middle-range (from centimeters to hundred meters) radio communication, depending of the used frequency. RFID has a variety of applications suitable for tracking: company badges and various tags or cards, pet microchips, passports or clothes. RFID requires some kind of cooperation from the tracked persons since they have to carry the RFID device. In the case of cars, RFID is used for electronic toll collection to reduce delays on toll roads. RFID devices can be placed in tires, or in parking passes [169].

Near-Field Communication (NFC): NFC is a short-range radio communication technology. Similarly to RFID, NFC technology is used in various cards or tags, as well as mobile phones. While range is supposed to be limited to about 10-20 centimeters, appropriate hardware can eavesdrop communications from 20-30 cm, possibly even more [116].

ANT: A proprietary technology operating on the 2.4 GHz band. Its primary application is to be incorporated in sport and fitness devices to communicate with a smartphone, which makes it a good candidate for individuals tracking. Some other applications fit that purpose as well: watches, heart rate monitors and cadence meters. To our knowledge, this possibility has rarely been studied or even mentioned for tracking or analytics [214], most probably due to its minor popularity compared to its competitors.

Tire-pressure monitoring system (TPMS) TPMS reports tire pressure to the car's driver using radio technology (315 MHz) that can be eavesdropped [169].

As a side-note, let's mention some non-radio technologies which are used in the real-world to track or identify individuals:

— Intelligent Video Analytics, which automatically analyzes content from video cameras [1],
— Automatic License Plate Recognition, an application of previous technology on vehicles [164],
— Voice recognition, used for instance in criminal cases [111],
— Stylometry, to identify the author of a text [22].

II.4.1.5 Cross-technologies tracking

Some authors studied tracking methods combining several technologies.

Martin et al. studied the ability for an attacker having access to Wi-Fi and GSM network traces to find the correlation between a MAC address and an IMEI [132]. Nguyen et al. use computer vision to make the link between a device and its owner, combining visual and RF technologies [155]. O'hanlon et al. found flaws in Wi-Fi authentication protocols that allow one to retrieve a target's IMSI using either passive or active attacks, effectively building a Wi-Fi-based IMSI catcher [160]. A public tool by Seiwert combines information

1. More information on this technology and its privacy implications can be found in a FTC forum: `https://www.ftc.gov/news-events/events-calendar/2011/12/face-facts-forum-facial-recognition-technology`, consulted on 2017.07.26

sent by application-layer protocols and probe requests' SSIDs to elaborate a map of places previously visited by a target iPhone device [183].

II.4.2 Applications of physical tracking

Nowadays, physical tracking systems are deployed in shopping centers [67], urban transportation systems, highways or ring roads [61]. See chapter III for a broader review of real-world Wi-Fi tracking installations.

Despite some efforts from the industry and close surveillance from data protection authorities, users' privacy is still in jeopardy [46, 191, 86]. This is aggravated by the fact that this technology is not well known by the general public, usually not aware that such systems exist and that passers-by may have been tracked.

II.4.2.1 Wi-Fi-based physical analytics

Physical analytics is a set of technologies aiming at collecting statistics about passers-by in a place. It has a lot to do with its web equivalent, web analytics. For instance, a city council may want to obtain statistics about mobility patterns, or a shop manager may want to get statistics about its customers in terms of peak traffic periods or popular paths inside the shop to increase sales.

Wi-Fi analytics is a currently expanding technology to achieve these goals, taking advantage of the fact that most customers carry mobile devices sending identifying information to track them individually. Wi-Fi analytics can be used to make statistics about crowd density [209], pedestrian flows on a city scale [66], population dynamics [115], customers in a shop or a shopping mall [46], or road traffic [219]. See section III.3 for real-world examples of such applications.

Wi-Fi analytics has gathered some interests from the scientific community. Chon et al. built a Wi-Fi "crowdsensing" tracking system, i.e. using two dozen individual mobile phones to massively gather Wi-Fi frames in a city [29]. Weppner et al. made a similar system to track Bluetooth scans [210]. Depatla et al. proposed a less intrusive approach to people counting: using RSSI modifications between two antennas [47]. While promising in terms of privacy, this technique is tested on a group of 9 people, and no indication is given on its scalability on bigger crowds. Zeng et al. went as far as estimating shoppers'

behaviour using the signal strength of their mobile device. However, it only works if a single shopper is in range and requires cooperation of the user's personal device [222]. Fawaz et al. proposed an indoor-tracking system in which users can define privacy preferences which define when their location is revealed in exchange of a reward [57].

Some authors have tackled the privacy problems in analytics and came up with some privacy-preserving location analytics solutions. These solutions trade a little accuracy against privacy guarantees, such as unlinkability with other datasets (which includes not storing identifiers). These methods usually imply using a data structure containing a reduced amount of information, and providing methods to probabilistically evaluate their actual content. Such data structures may be Linear Counting sketches [108] or Bloom filters [5].

II.4.2.2 Profiling

Physical tracking can have more applications than just getting location information about a person. By keeping location information along time, or crossing it with other sources of information (e.g., bought products in a shop), trackers can create a profile for each person, i.e., a list of its usual behaviours [214].

This is highly valuable for advertisers, which can use this information to place targeted advertisement. This advanced form of tracking is even more troublesome privacy-wise, as it means that private information is kept for an extended period of time and shared with third parties [46]. Moreover, such technologies possess inherent privacy design flaws: feedback loops that normalize certain categories of behaviours in the population, and potential for unjustified discrimination [198]. These issues grew concerns when bins including Wi-Fi tracking systems were installed in London [194].

This kind of tracking can also be leveraged for surveillance, in order to keep track of the whereabouts of individuals, or detect suspect behaviours [181].

II.4.2.3 Vehicles tracking

A wide-spread application of analytics is to obtain information on vehicles using sensors places along the road. This can be used to estimate real-time vehicle delay, travel time and origin-destination data [219].

Using portable sensors in Maryland and Delaware, Sharifi et al. calculated the vehicles detection rate. It turns out only 2 to 8% of vehicles are detected by the system [186]. While Bluetooth is commonly used to track vehicles, Luber and Junghans wondered to which extent Wi-Fi could replace it to calculate travel time and mean travel speed. They obtain slightly less precise results using Wi-Fi, because Wi-Fi devices detection is much lower than Bluetooth ones (1% compared to 6.5%) [129].

II.4.2.4 Other uses of physical tracking

Evidences exist that mobile phones tracking is performed by states-sponsored agencies to retrieve stolen phones [1], journalists [92] or war opponents [2]. The NSA [3] is suspected of performing cellphone tracking using a wide range of methods: massive cellular tracking by tapping directly into network links between providers [179], Wi-Fi-based tracking using pods mounted on drone to monitor data from both routers and mobile devices on a city scale (i.e., wardriving) [71, 179], tracking using location information sent in cleartext by applications over the Internet, tracking using location information of the Wi-Fi networks devices log into [181], plus some programs involving explicit hacking of the target phone. They're even suspected of using tracking for assassination by drones [179]. Wardriving using drones had been demonstrated before [208]. It's even been shown that device localization is possible [3].

Use of IMSI catchers by government agencies is public notoriety and has even been made legal in some countries such as Germany [4] and France [5].

Using RFID chips to track people has long been a dystopian scenario. A company in the U.S. recently made a public statement that their employee could be "voluntarily" chipped [6]. In fact, inserting RFID chips in uniforms has already been done to track schoolchildren [7] [8].

1. https://www.cnet.com/news/russian-police-spy-on-peoples-mobile-data-to-catch-thieves/, consulted on 2017.07.21
2. https://www.crowdstrike.com/blog/danger-close-fancy-bear-tracking-ukrainian-field-artillery-units/, consulted on 2017.07.21
3. National Security Agency
4. https://www.gesetze-im-internet.de/stpo/__100i.html, consulted on 2017.07.25
5. https://www.legifrance.gouv.fr/affichTexteArticle.do?idArticle=JORFARTI000032627262&cidTexte=JORFTEXT000032627231, consulted on 2017.07.25
6. https://www.prlog.org/12653576-three-square-market-microchips-employees-company-wide.html, consulted on 2017.08.25
7. https://www.theguardian.com/uk/2007/nov/24/schools.education, consulted on 2017.07.31
8. http://www.zdnet.com/article/schoolchildren-to-be-rfid-chipped/, consulted on 2017.07.31

II.5 Countermeasures to Wi-Fi-based tracking

Information leakage due to active service discovery of Wi-Fi-enabled devices poses important privacy issues, introduced in section II.2. Various countermeasures have been proposed to solve this issue, from privacy-preserving service discovery to the use of temporary identifiers in probe requests. Some solutions have focused on implementing privacy-preserving Wi-Fi analytics solutions not requiring a wide modification of every Wi-Fi-enabled device.

II.5.1 Privacy-preserving service discovery

Several systems have been proposed to handle service discovery in a more privacy-friendly way. An early public-key-based solution was speculated by Greenstein et al., who dismissed it because of its computationally-expansive cost [77]. Pang et al. proposed a mechanism adding confidentiality to various service discovery protocols. To achieve this, they use various cryptographic methods, including public and symmetric key protocols, and anonymous identity-based encryption [162]. This mechanism was extended in a later article by similar authors, who proposed an alternative to the full 802.11 link-layer protocol obfuscating all transmitted bits, including identifiers. This article notably adds a mechanism to obfuscate identifiers in data packets [78].

A more basic approach to increase privacy of service discovery is simply to make use of it only when deemed useful. While manually turning Wi-Fi on when necessary can be cumbersome, a more practical solution is geofencing. This technique activates the Wi-Fi interface only when the device is in a location close to the AP's one. A close technique has been proposed by Kim et al., which adapted probing to the location context, sending only SSIDs of APs known to be present at this location [112]. Wi-Fi Geofencing has been spotted in the developer preview of Android 8.0 [1].

To fix the problem of human-readable network names in probe requests, Lindqvist et al. proposed a system using a shared-key challenge-response protocol to discover APs, including hidden APs [127]. We note that for this system to work, hidden APs have to respond to all probe requests, thus revealing their presence permanently. This defeats the purpose of using hidden APs.

1. `http://www.androidpolice.com/2017/04/04/android-o-feature-spotlight-automatically-enable-wifi-youre-near-saved-network/`, consulted on 2017.07.21

Wu et al. proposed a system suitable for any service discovery protocol. Similarly to previous article, service advertisement messages (probe requests in our case) are protected by identity-based encryption (a generalization of public-key encryption) [216].

II.5.2 Mix zones

Preventing trajectory tracking can be done using path confusion techniques, such as mix zones. The idea is to set zones where user's traces get mixed, so that a user cannot be linked to a full trajectory [90] [1]. For instance, users may exchange their pseudonyms through cryptographically-protected messages, as proposed by Freudiger et al. in the case of vehicular networks [64]. Conversely, while studying mix zones in the case of vehicular tracking using predictable fields, Bloessl et al. found that tracking was still possible with good results. In this specific case, an attacker possesses sufficient information for this, such as used lane, speed, and the predictable field [19].

II.5.3 Pseudonyms

A *pseudonym* is a temporary identifier used by a device to avoid leaking its unique and permanent identifier. Examples of pseudonyms are TMSIs in cellular technologies or random MAC addresses in Wi-Fi or Bluetooth.

While pseudonyms protect users against the possibility of tracking using explicit identifiers such as MAC addresses, it is often an insufficient technique. An attacker can still form *implicit identifiers* [77]. They may be fields changing in a predictable way (e.g. sequence numbers [63] or scrambler seed [19]) or any field bringing some bits of entropy. Implicit identifiers can also take the form of a group of individual pieces of information which, alone, are not unique, but reach a high probability of uniqueness when taken as a group (*meta-identifier*).

Side information can be used to defeat pseudonym uses on a short timescale. Information from other layers can be used to track devices [19], movement patterns [220], as well as signal strength [13, 55, 187] [2]. Sequence numbers have been shown to be an efficient way to detect pseudonym use, in the specific case of spoofing [82], and in the case of MAC

1. This technique is used in the real-world by Taliban leaders to defeat NSA tracking of their SIM cards [179]
2. Another source estimated that the false positive rate is too high, around 20-30% [81].

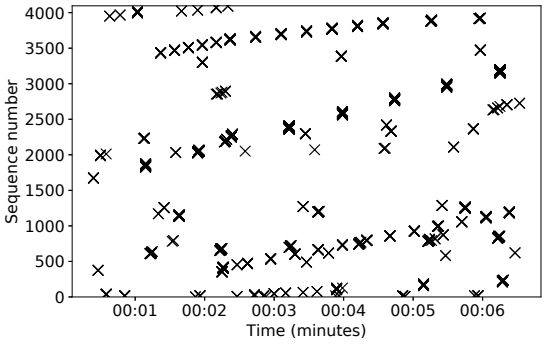


Figure II.4 – Sequence numbers wrt. time in part of the Lab dataset. Many frequency patterns clearly appear.

address randomization [63]. In the latter case, Freudiger also noted that some IEs could be used for this purpose.

The fact that sequence numbers can be leveraged to defeat MAC address randomization is worth some more words. As seen in figure II.4, a lot of probe requests can be grouped trivially visually because of this issue. To our knowledge, no algorithm automating this task has been published.

Vehicle paths are highly predictable as they mostly follow a straight line path. As a consequence, this information can be leveraged to track these devices despite their pseudonym use [59, 213]. We're unsure whether these techniques could be applied on devices traversing a less predictable path, such as smartphones. Some works seem to hint in this direction [94].

Pseudonyms have been especially studied and extended in the case of vehicular networks. They've been, for instance, combined with mix zones [52], silence periods [94, 106], and dynamic modification of the signal strength [106] to increase their effectiveness as a privacy protection feature. Förster et al. evaluated location privacy brought by pseudonyms in such a case [59]. Liao and Li proposed *synchronous* changes of pseudonyms, in which a vehicle advertises its plan to change its pseudonym at a certain time, so that other vehicles do so at the same time [125]. These works introduce solutions which can be adapted to similar problems in other technologies, such as Wi-Fi or Bluetooth.

II.5.3.1 Wi-Fi MAC Address Randomization

Originally, having stable and globally unique identifiers made it easier to avoid any kind of collisions on identifiers on MAC-layer protocols. For that matter, the IEEE made it mandatory for any company selling components operating on that layer to purchase an Organizationally Unique Identifier (OUI), and to use it to attribute unique addresses to every component. While they avoid errors due to collisions for many protocols, unique identifiers have a drawback: they allow devices to be easily identifiable by anyone. This constitutes a privacy issue in some cases. When it comes to 802.11 service discovery, the tracking threat is high, because it is permanently performed by everyday-carried devices. On the other hand, the collision problem is minor because it would not disrupt the protocol. For these reasons, the MAC address randomization technique was introduced, primarily in this use case. See section IV.2 for a discussion on the different implementations of the technique.

The IEEE standard advocates that devices doing so set the Locally-Administered bit of the MAC address to 1 [96] (see section I.2.1), but observations in the wild show that this is not always the case, as later discussed in sections IV.3 and IV.4.

In 2005, the idea of MAC address randomization was already tested, for both unassociated and associated devices. Gruteser and Grunwald tested an implementation and concluded that it indeed reduced the average tracking time, but found that address collisions could be a problem in large networks [81]. To avoid this issue, Jiang et al. proposed using a unique global *join address* along with a nonce longer than a MAC address to distinguish users. Then, the random MAC address used during the association is assigned by the AP [106].

Bernardos et al. studied various implementations of automatic or manual MAC address changing tools. They were part of an IETF experimentation to make the first tests of real-world use of MAC address randomization[1]. These tests are useful to verify that possibly unpredicted side behaviours, such as the exhaustion of DHCP addresses pools, do not happen in practice. They caused few, minor issues [18].

1. On this occasion, they also produced a tutorial to randomize MAC addresses on several systems: `https://oruga.it.uc3m.es/802-privacy/index.php/MAC_address_change_tutorial`, consulted on 2017.07.27

MAC address randomization in Bluetooth

Due to its strong parallel with Wi-Fi, we mention how MAC address randomization is designed in the Bluetooth technology.

Bluetooth integrated MAC address randomization earlier than Wi-Fi in their standard, starting from Bluetooth 4.0 (Low Energy) and improved in Bluetooth 4.2 [215]. Similarly to Wi-Fi, unassociated devices ("unpaired" devices, in the Bluetooth terminology) can frequently change their MAC address to a pseudonym. This policy is rarely enforced in the wild [101]. For instance, Siby et al. noticed that a Bluetooth smart lock did not randomize its MAC address [188]. Pseudonyms are changed based on a timer, and possibly other triggers, such as switching the device on and off [215].

Randomization is more advanced in Bluetooth than in Wi-Fi. Notably, it is not limited to unassociated devices. When pairing, Bluetooth devices exchange an Identity Resolution Key (IRK), a cryptographic key specially dedicated to reversing the pseudonym MAC address into the actual MAC address of the paired device.

Another mechanism exists to avoid leaking original MAC addresses: reconnection addresses. Bluetooth devices can directly advertise their wish to pair with a device they previously paired with, using requests containing both MAC addresses. In this case, they do not use their own MAC address, but a reconnection address which can also be resolved back to the original address using the IRK.

MAC address randomization in Bluetooth Low Energy is not perfect, and some flaws allow an attacker to retrieve the actual address of a device. While paired Bluetooth devices stop broadcasting their MAC address, it's possible to force them to send it again by sending deauthentication requests or jamming used frequencies [56]. A similar problem as the one described in chapter V has been noted for Bluetooth: some devices advertise side-information while randomizing their MAC address, including a UUID, i.e., a unique identifier. A tool called Blue Hydra [1] has been developed to display all these broadcast information sent by nearby devices [69].

1. `https://github.com/pwnieexpress/blue_hydra`, consulted on 2017.07.14

II.5.4 Lack of options for consumer-side countermeasures

Privacy-focused individuals may wonder what actions they can take to avoid being subjected to tracking because of insufficient privacy protection in their personal devices. This is, however, a difficult task.

Turning off Wi-Fi on one's personal device may seem sufficient to avoid Wi-Fi tracking. However, we published a tech report showing that the situation is more complicated. On Android, 3 different options influence probing behaviours, including one option allowing devices to send probe requests even if the Wi-Fi switch is off. Troublesomely, some devices do not even allow this option to be deactivated [144].

Setting phones in plane mode appears not to be a completely satisfying solution either. iOS devices starting from version 8.3 keep receiving GPS data, even while in plane mode. Starting from iPhone 6, NFC remains enabled and will keep emitting signals [1]. Huang and Snowden built a tool to detect when a device emits data even while in plane mode [92].

To sum up, popular smartphones possess flaws which make deactivation of trackable signals complicated. Regular consumers lack better options than not carrying such smartphones so as not to be tracked.

Another way to see the privacy issue in physical analytics solutions is to oppose them. Going in this direction, the Valora tool generates fake probe request traffic to disrupt existing installations [2].

II.6 Art

Privacy researchers are not the only persons trying to gather attention on the privacy issues of mobile devices leaking information. To end this bibliography on a lighter note, let's mention artists who have made some work on this topic.

Wi-Fi Whisperer by Kyle McDonald and Surya Mattu use sniffers to capture traffic, and whisper extracted information in a creepy voice [3] [4]. Cracken by Kévin Ardito is an

1. https://www.macobserver.com/analysis/ios-airplane-mode-gps-nfc/, consulted on 2017.07.26
2. https://github.com/antoinet/valora, consulted on 2017.08.02
3. https://soundcloud.com/kyle-mcdonald/whisper, consulted on 2017.07.28
4. https://www.wired.com/2016/06/wifi-whisperer-stalks-phones-data-creepiest-way-possible/,

installation looking like a cyborg tentacle which reacts to Wi-Fi data (presumably probe requests) to "allow an organic, pathetic visualization tinted with Japanese pornography" [1]. PRISM: The Beacon Frame by Julian Oliver is a project aiming at replicating what actual spying equipments look like, despite the lack information about them. It contains both a Wi-Fi tracker and an IMSI catcher, which are used to collect sensitive information about the crowd. This information is then display through a prism in a "rich and exploitative light show" [2].

consulted on 2017.07.28

1. See http://www.makery.info/2017/03/14/au-mirage-festival-on-a-touche-du-doigt-le-virtuel/, consulted 2017.07.28
2. https://julianoliver.com/output/the-beacon-frame, consulted on 2017.08.23

Chapter III

Overview of real-world deployment of physical analytics systems

This chapter studies the real-world deployment of physical analytics systems. Starting with a few real-world examples, it then discusses various aspects of such systems: privacy implication, regulation, consent, public acceptance, and engineering aspects.

III.1 Introduction

Physical analytics systems are booming in various places of the world. Due to the recent proliferation of Wi-Fi-enabled mobile devices (detailed in this chapter), many companies have seen this as an opportunity to gather statistics about clients, populations or vehicles and developed systems for that matter.

We give an overview of the current deployment of Wi-Fi and Bluetooth physical analytics systems in its various forms throughout the world. We do not mean to be exhaustive, as we primarily focus on French and English-speaking resources (thus excluding many local press articles). We start by discussing the recent evolution of trackable devices in section III.2. Then, we list some installations in section III.3. In section III.4, we discuss their actual privacy guarantees. Section III.5 lists details of regulations and their applications in various countries. Section III.6 talks about consent and its possible implementations. Finally, in section III.7, we give evidences of the generally negative acceptance of tracking systems in public opinion.

Due to the close proximity of Bluetooth and Wi-Fi analytics systems, we group both of these technologies in this chapter.

III.2 Evolution of the wireless landscape

First, let's study the evolution of the number of everyday-carried Wi-Fi-enabled mobile devices to justify the wide-scale privacy threat of tracking.

III.2.1 Number of devices

Worldwide, more than 7 billion cellular subscriptions[1] are active in 2015 [100]. Wide disparities exist, as mobile broadband subscriptions range from 86.7 for 100 inhabitants in developed countries, to 12.1 for 100 inhabitants in the lesser developed countries in 2015. Mobile cellular subscriptions keep increasing, and reach more than 100% in several parts of the world, indicating that some people are using more than one device [99].

The previously presented figure of 86.7 broadband subscriptions for 100 inhabitants in developed countries also indicates a new trend: in these countries, a huge majority of people possesses an Internet-enabled device[2]. In 2016, in France, 77% of people aged 18-75 declare possessing a smartphone [45]. As these devices integrate software necessary for internet usage (web browsing, emails, messaging...), most broadband-generation devices also integrate Wi-Fi hardware, to allow for a faster and cheaper Internet access than the cellular technologies.

Adoption of Wi-Fi-enabled mobile devices has developed rapidly in the last few years. For instance, in the U.S., the rate of adults owning a smartphone skyrocketed from 35% in 2011 to 68% in 2015 [8], reaching 77% in latest surveys [166]. Similarly, tablet computer ownership rose from 3% in 2010 to 51% in 2016. France exhibits a similar trend: smartphone ownership rate rose from 29% in 2012 to 65% in 2016 [9].

1. This encompasses all kinds of cellphones, not only (Wi-Fi-enabled) smartphones.
2. Broadband allows a fast Internet access.

III.2.2 Detection possibilities

Despite their wide spread, are these devices good candidates for tracking? In this section, we will discuss the number of devices which satisfy the conditions to be *detectable*: carried by their owner, active, possessing a functioning Wi-Fi interface, and actively sending frames.

A study in the U.S. shows that 94% of smartphone owners carry their phone frequently, and 82% turn them off either rarely or never [171]. However, all of these devices are not detectable at all times: all devices do not send Wi-Fi frame periodically, or the Wi-Fi interface can be temporarily deactivated or malfunctioning. In 2017, penetration rate of mobile telephony is greater than 100% in many parts of the world [98].

Estimating how many of these devices are trackable is an unanswered question. To our knowledge, no public study of Wi-Fi activation rate on mobile devices exists. Besides, we showed that devices whose Wi-Fi switch is off can still be lead to emit Wi-Fi signals [144].

Estimating the ratio of devices over the number of people in a population is a tough question. A number of biases exist, i.e. depending on the country [100, 98], the population's distribution of ages [171], education [166], developed environment [166], income [193] or socio-professional category [98], etc [153]. Different figures have been presented regarding the number of trackable devices. In 2013, a Wi-Fi tracking company put forward the figure of 40% to 60% of a mall's visitors, depending on the location (city) [85]. In [209], authors calculate the ratio of devices over population size in a car manufacturer exhibition, using a ground-truth obtained via camera detection. Their results indicate that the targeted population is on average 1.5 times more numerous than the number of devices, this factor varying from 1.0 to 2.6. Experimental results indicate potential large differences in this factor in similar events. Performing captures in 10 security conferences around the world in 2012-2013, Wilkinson recorded factor ranging from 0.44 to 3.75 between the number of devices and the number of conference attendees [1] [214]. A study in 2015 in Manhattan [115] using census and administrative data from several sources as ground-truth values found results within ±15% of these census data. Their estimate is even between 2 to 5% of the counts of the most reliable sources, according to them. As they do not adapt the count of Wi-Fi-enabled devices to the estimated population count, this suggests a close one-to-one correspondence between population count and Wi-Fi-enabled devices count. This result is quite surprising, considering a survey in the same city in the same year

1. Average: 1.58; Standard deviation: 1.11

which found that only 79% of the population owned a smartphone [153]. This rate lower than 1 may be compensated by people carrying more than one device. It must be noted that this study gathers all conditions for statistics using Wi-Fi-enabled devices counting to give reliable results: number of passers-by is always great enough for results not to be affected by an important standard variation. Moreover, smartphone ownership in this city is high even among groups usually affected with a low ownership rate, such as low-income or old people [153]. According to a presentation slide for the tracking installation in the Railway museum of London (see link below), this installation reaches a 96% correlation between the number of visitors (for which they have ground truth) and the Wi-Fi-based counts, despite the fact that only 53% of visitors carry a Wi-Fi-enabled device.

Abedi et al. compared Wi-Fi and Bluetooth regarding the ability to monitor people. Their conclusion is that, due to differences in transmission range, popularity, probing rate and default configuration in popular devices, Wi-Fi is more suitable for this application. In their experiment, among around a thousand detected MAC address from both protocols, more than 90% of them are Wi-Fi addresses [1]. On a dataset of more than 6 000 addresses, Shauer et al. obtained a similar ratio of 94% in favor of Wi-Fi addresses [180].

All these numbers show that the possibility of tracking individuals on a massive scale has recently become a very real possibility, and therefore an issue.

III.3 Fields of application

Despite strong regulations in many countries (detailed in section III.5), Wi-Fi and Bluetooth analytics installations slowly develop in many infrastructures. This section lists a few evidences of real-world analytics solutions.

Road traffic analytics:

- A Bluetooth-based vehicle tracking system in Houston provides real-time traffic information to the general public [1].
- A tracking system based on both Bluetooth and Wi-Fi is installed on Lyon's ring road [79].
- A Bluetooth-based traffic detector is installed in Maryland [219].

Retail analytics:

1. `http://www.houstontranstar.org/faq/trafficTech.aspx`, consulted on 2017.06.06

— In May 2017, CB Insights identified several dozens of start-ups working on location analytics for retailers [26]. See also the article by Demir et al. listing analytics companies [46].
— In *La Défense* in Paris, a shopping mall recently installed a Wi-Fi analytics system without prior notice [1].
— In the US, a company called Nomi sold Wi-Fi analytics systems to approximately 45 clients in 2013. Some clients deployed these systems in multiple locations. According to the FTC, these different installations collected information on no less than 9 millions of individual devices between January 2013 and September 2013 [58].
— The Norwegian Data Protection Authority published a report in 2016 about tracking in public spaces, reminding concepts of consent, pseudonyms, etc. [43]. The report compares 4 tracking technologies: Wi-Fi tracking, Bluetooth tracking, beacons and Intelligent Video Analytics. The report gives two examples of Wi-Fi tracking installations in Norway:
  — at Hamar's *CC Stadion* shopping center,
  — at the Oslo airport, a Wi-Fi tracking system estimates waiting time to pass security check, using one AP before and after customs.
— A Wi-Fi and Bluetooth-based tracking system is operational in Amsterdam's airport since mid-2017 [2].
— Apart from retail stores, Wi-Fi tracking is also used in other closed places. For instance, two museums in London now have permanent Wi-Fi tracking installations to get information such as the most visited rooms, or for "security concerns" [3].

Crowd analytics / population statistics:

— In the U.K., the Transport for London corporation tracked subway commuters for one month in the end of 2016 [156].
— Similarly, a wide 6-month experiment was performed in New York, using 53 APs to collect Wi-Fi information, in order to know more about population dynamics [115].
— Several Wi-Fi tracking systems are installed in various French cities: Niort since

1. http://tempsreel.nouvelobs.com/rue89/rue89-nos-vies-connectees/20170711.OBS1939/vous-etes-reste-22-minutes-chez-l-opticien-jeudi-et-le-centre-commercial-le-sait.html, consulted on 2017.07.14
2. https://www.schiphol.nl/en/privacy-policy/, consulted on 2017.09.13
3. http://www.gizmodo.co.uk/2017/04/exclusive-heres-what-museums-learn-by-tracking-your-phone/, consulted on 2017.09.08

March 2017 [1], Rennes since February 2017 [2]. In these hybrid cases, crowd analytics is used for retail analytics, on a larger scale.

— The Chinese government made an announcement in 2011 about the installation of a tracking system in Beijing, targeting no less than 17 millions cellphones [109]. This raised international concern about the surveillance capabilities and other possible abuse of such a system [173]. We're not sure about the technology used in this case.

— In Singapore, a company is using drones to perform wardriving with the intent to create user profile to serve advertisements. In 2015, they claimed they had no less than 530 millions user profiles [3].

Surveillance:

— Some of the evidences of uses of state-sponsored physical tracking presented in section II.4.2.4 involve Wi-Fi-based systems. Notably, wardriving is performed using drones, to monitor populations on a city scale [179].

III.4 Privacy aspect in real-world installations

A general claim for these installations is that they are actually *more* privacy-preserving that other systems, such as licence plate or face recognition because of the encryption used before storing data. However, this encryption is usually weak, taking the form of the creation of a single pseudonym which can be as weak as a salt-less hashing [58]. Demir et al showed how risky this approach is. Salt-less hashing of MAC addresses can be reversed within seconds using a modern GPU, because of the small address space of used MAC addresses, mainly if the reversal attack is limited to allocated OUIs. Hashing using a salt may not be a better-suited solution, as the salt needs to be stored by the system to hash addresses on-the-fly. As a consequence, compromise of a system usually includes the compromise of the salt [46]. Kumar when further in this analysis, taking into account the OUI semantics: OUIs registered by vendors not producing mobile devices can be ignored as well [120].

1. http://www.lanouvellerepublique.fr/Deux-Sevres/Communes/Niort/n/Contenus/Articles/2017/03/17/Wifi-VIP-la-publicite-directe-sur-les-smartphones-3035563, consulted on 2017.06.06

2. http://www.20minutes.fr/rennes/2011831-20170210-rennes-capteurs-wifi-suivre-clients-centre-ville, consulted on 2017.08.26

3. https://venturebeat.com/2015/02/23/drones-over-head-in-las-valley-are-tracking-mobile-devices-locations/, consulted on 2017.07.27

Besides, data is often stored for long periods of time for debugging or analysis purposes. For instance, unlike originally planned, the Maryland installation ended up keeping perpetual online archives [219]. The Lyon installation stores identifiers for more than 1 month [79], and the one in *la Défense* for 6 months.

A 2014 analysis by Demir et al. showed how insufficient the privacy policies applied by Wi-Fi tracking companies were at that time. For many of them, they found a combination of long retention periods, data storage delegated to third-parties, weak hashing and absence of opt-out system [46].

Even more questionably, Wi-Fi tracking is performed by some public Wi-Fi providers. Some of them exploit their man-in-the-middle position to collect information of questionable interest such as age, gender, and photo on social media [119].

Despite their sensitive operations, some tracking systems eventually turn out to be poorly secured, exposing users to potential privacy breaches. Cerrudo reversed-engineered wireless sensors used worldwide for vehicle counting and found alarming vulnerabilities: lack of encryption and authentication, unencrypted and unsigned firmware updates [27].

User notifications aren't always correctly done. For instance, this was one of the topics of an administrative complaint by the FTC towards the Nomi company in 2015 [58].

Some Wi-Fi analytics systems go further into tracking by combining multiple technologies to track consumers, a technique sometimes labeled *convergence* [190]. For instance, one company goes as far as crossing at least 8 sources of information, from video camera footage to payment card data [1]. This is highly troublesome in terms of privacy, because a lot of privacy-sensitive information can be gathered from these multiple sources. Crossing them together bring even more meaningful information.

Privacy guarantees in these systems are complicated to enforce, and weaken their usefulness for advertisers. For instance, calculating the revisit rate of customers implies that visitors' information is kept for an extended period of time using pseudonyms, at best. While pretending to anonymize collected data, the Niort system (see above) is able to calculate this revisit rate.

1. `https://retailnext.net/en/how-it-works/`, consulted on 2017.07.16

III.5 Regulation

Due to these privacy issues, Wi-Fi tracking and analytics is often limited by regulation entities.

In France, the CNIL published detailed rules that retail Wi-Fi tracking installations must respect to be authorized [31]:

— data must be deleted when the device owner leaves the shop (it can be aggregated),
— or a strong collision rate must be ensured, i.e. recorded identifiers must correspond to several devices (without precise numbers about this minimum collision rate).

Companies not respecting these rules can be fined or have their installations rejected or forbidden. For instance, the CNIL rejected in July 2015 a proposition of an installation in *la Défense* in Paris [1], on the basis that pseudonymization is not sufficient to provide anonymity. To be more precise, they noted that MAC addresses hashing with a salt does not prevent the processing manager to cross recordings with other sources of information, or to infer other information about users. Despite the fact that a counter-procedure by the company was rejected [2], an installation is present as of July 2017 (see section III.3).

In Sweden, the Swedish Data Protection Authority amended a company to modify or cease a Wi-Fi tracking installation in the city center of Västerås because MAC addresses were stored in cleartext [3]. The installation was later accepted in exchange of modifications in the systems: MAC addresses are deleted after a few seconds, and information about the system is displayed on boards in the city and on the company's website [4].

In the United States, the Federal Trade Commission (FTC), while not as strict as in other countries, is also careful about the privacy aspect of tracking installations [74]. The Electronic Frontier Foundation (EFF), an international non-profit digital rights group, also reminded that it might be illegal to capture MAC addresses [86].

In the U.K., Wi-Fi tracking bins were removed due to privacy concern after an order from the City of London Corporation [194].

1. https://www.legifrance.gouv.fr/affichCnil.do?oldAction=rechExpCnil&id=CNILTEXT000031159401, consulted on 2017.07.14
2. http://arianeinternet.conseil-etat.fr/arianeinternet/getdoc.asp?id=209297&fonds=DCE, consulted on 2017.07.14
3. http://www.datainspektionen.se/press/nyheter/2015/besoksflodena-i-vasteras-mats-for-noggrant-/, consulted on 2017.06.07 (in Swedish)
4. http://www.datainspektionen.se/press/nyheter/2015/gront-ljus-for-besoksmatning-i-vasteras/, consulted on 2017.06.07 (in Swedish)

III.6 Consent in physical tracking

Consent is an important aspect for privacy protection. It basically states that users give their agreement to share information about them. As we defined privacy in section I.2.4 as the ability to choose which personal information someone shares with whom, there cannot be privacy without consent.

Two approaches exist for a user to indicate whether they want to be part of a system:

- Opt-in: the users explicitly announce that they want to be part of the system (and are therefore not part of the system by default).
- Opt-out: the users are considered part of the system by default, and can indicate that they do not want to.

To prevent oneself from being tracked, real-world installations offer either an opt-out system, or nothing at all. For instance, we did not find any possibility to opt out of the existing vehicle tracking systems listed above. When existing, the opt-out system is sometimes poorly advertised [74]. In the Nomi-FTC case, one of the complaints was that the opt-out was global to all installations of Nomi's system, and not store-relative, and not possible inside stores. We found no evidence of existing systems using an opt-in method. When tracking is performed by Wi-Fi providers, information of the fact that users will be tracked may be hidden in the contract's details, sometimes even erroneously pretending that keeping location information is a legal requirement [119]. While this can be considered as a form of opt-in, one must remember that these terms are almost never read entirely (if at all) by users [157].

Opt-out mechanisms typically involve a webpage on which the user needs to enter their device address [1] (see Figure III.1). In the Niort installation, users can opt out by scanning a QR code [2]. In *La Défense*, users have to send an email to oppose tracking. A problem with all of these systems is that it requires the user to be able to access its MAC address. While this is already a complex operation for a non-tech-savvy user, it is (almost) impossible for devices such as fitness trackers, which do not provide easy access to this piece of information. In a shopping center in Paris, it's even been reported that customers are invited to turn Wi-Fi off altogether so as not to be tracked [3]. This is also written

1. `https://smart-places.org/`, `http://flux-data-vision.com/optout.html`for the Niort installation, both consulted on 2017.06.06
2. We assume this QR code redirects to the related opt-out webpage.
3. `http://www.lefigaro.fr/secteur/high-tech/2017/08/02/32001-20170802ARTFIG00264-le-bhv-aspire-les-donnees-de-ses-clients-mais-il-est-loin-d-etre-le-seul.php`, consulted on 2017.08.04

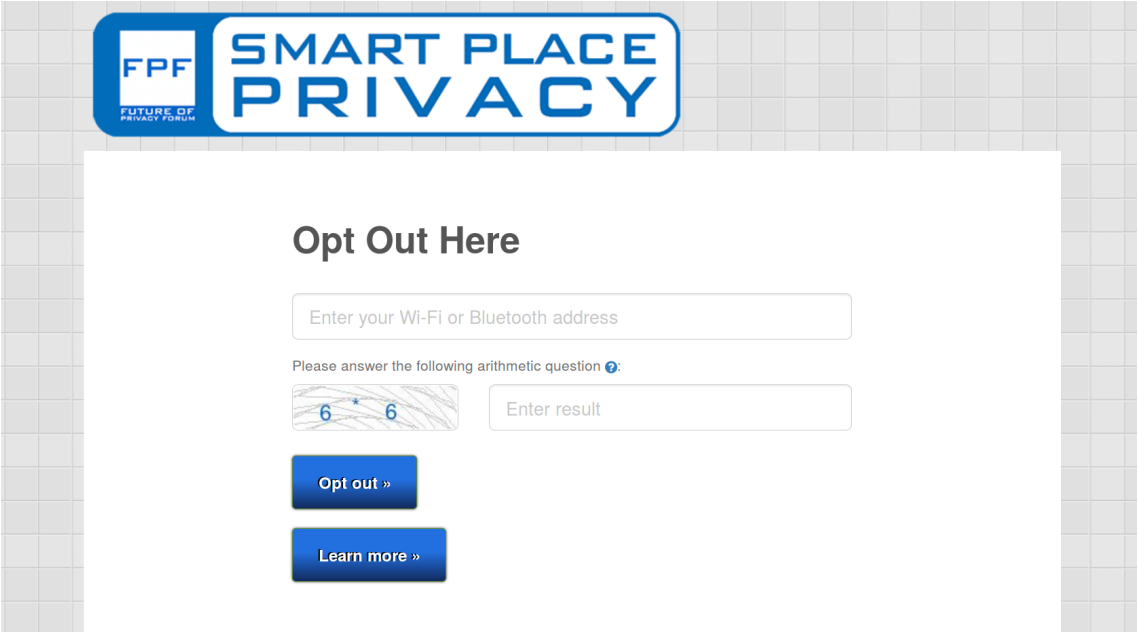


Figure III.1 – Screenshot of the opt-out webpage set up by the Future of Privacy Forum: `https://optout.smart-places.org/`. Users entering the address of their device opt out of Wi-Fi analytics systems deployed by most major Wi-Fi analytics companies in the US.

in the privacy policies of Amsterdam's airport's system (see link above). This method is not sufficient: we showed that Android devices having the "always allow scanning" option activated will keep sending probe requests even if Wi-Fi is deactivated. Moreover, we have observed a device not proposing any way to deactivate this option [144].

One of the problems in Wi-Fi analytics is that Wi-Fi frames are not stopped by walls. A tracking system will then record information on people not entering the place (e.g. walking in nearby streets). This is problematic when entering a place is considered a form of consent (which justifies the opt-out strategy) [1].

Tracking using other technologies than Wi-Fi or Bluetooth may be even more problematic regarding the consent question. Soltani made a summary of consent and notice in various technologies [191].

Possible improvements to opt-out mechanisms are discussed in section VII.2.

1. Some companies actually advertise this issue as a feature: `http://www.libelium.com/products/meshlium/smartphone-detection/`, consulted on 2017.07.27

III.7 Public acceptance

Wi-Fi tracking systems are generally not accepted by the population. A 2014 survey by OpinionLab on 1000 customers found a rejection rate of tracking in retail stores of 80% [158]. Primary cited concerns are "data security" and "spying". Fawaz et al. found a similar result of 70% of rejection in a survey on 200 Amazon Mechanical Turk participants [57]. The latter study found that only 10% of participants accepted full gathering of their Wi-Fi broadcast information. While modifying the question to state that the store explicitly asks user consent for tracking, these numbers moved to respectively 61% and 15%, which suggest that consent plays a key role regarding user acceptance of tracking.

When the public is aware about existing Wi-Fi tracking installations, strong concern often rises from local association or political parties, as shown in the Niort case [1] or for an installation in Rennes [2]. The latter has lead to a suspension of the installation until the CNIL gives its opinion [3]. In the U.S., customers got unnerved about tracking in retail stores [30]. Politician interventions against Wi-Fi analytics lead to the redaction of a code of conduct [4] for mobile location analytics [86, 134]. The latter advocates use of an opt-out system and explicit notice.

It can be noted that common users may have a bias towards accepting systems they do not understand, or if they do not understand the extent of leaked information. Kowitz and Cranor showed how users changed their attitude when shown some information leaked by their personal device on a local network [118].

Acceptance of physical tracking is best understood when compared to the perception of other forms of tracking, notably online (web) tracking. Studies have shown that this form of tracking is widely rejected. A study found that 66% of adult Americans reject targeted advertisement, and would not allow advertisers to track them online if they had a choice. This number rises to 86% when they're explained how data is collected, and to 90% if targeting is the result of their offline activities [196]. Another study found a similar rejection rate of online tracking of 68% [170]. It can be noted, however, that users' behaviour often differs from their privacy statements [17], and that they often have strong

1. http://deuxsevres.eelv.fr/12/wifi-vip-un-dispositif-couteux-qui-porte-atteinte-a-la-vie-privee-des-niortais-e-s/, consulted on 2017.06.06
2. http://www.ouest-france.fr/bretagne/rennes-35000/commerce-rennes-dominique-fredj-demissionne-du-carre-rennais-4860890, consulted on 2017.06.07
3. http://www.20minutes.fr/rennes/2027787-20170309-rennes-commercants-reportent-mise-service-capteurs-wifi-suivant-smartphones, consulted on 2017.07.14
4. https://fpf.org/wp-content/uploads/10.22.13-FINAL-MLA-Code.pdf, consulted on 2017.06.08

misconceptions about tracking [148].

III.8 Conclusion

We saw in this chapter various real-world examples of Wi-Fi-based tracking systems. Most of them present shortcomings in terms of privacy, when it comes to gathering user consent, presenting an easy-to-use opt-out system, or simply following regulation guidelines.

We state that all these systems should be built following the privacy-by-design principle. The latter is a core concept when it comes to building privacy-preserving systems. It basically states that systems should integrate privacy mechanisms in their core structure, and not treat it as an optional feature. Basic concepts that ubiquitous computing-related systems must integrate are clear notices, explicit user consent, adequate security and anonymity [122].

Some readers may be interested in the engineering aspect of physical tracking system. A part of Young's report on the Bluetooth-based traffic detector in Maryland [219] gives interesting insights about the problematics of a permanent deployment of such a system. The solutions to these issues chosen by this system's designers are the following. For the necessary resistance to temperature and humidity extremes, the sensors respect the NEMA TS2 standard, a standard for traffic control assemblies. To get a sufficient source of energy, each sensor is equipped with a 30 Watt solar panel. For data communication, cellular communication using a GDM modem is used. The final cost of the installation is 7 200$ per sensor, including hardware, installation, and data cost. On the whole, the life-cycle of the system is 5 years. Additionally, Grolleau published a report discussing various aspects of Lyon's ring road's system. Similarly to the previous one, sensors are powered with 70 Watt solar panels [79]. More information on this installation can be found in a presentation by Purson et al. [61].

Chapter IV

MAC address randomization

This chapter focuses on MAC address randomization, a tracking-prevention technique currently under deployment by most core mobile device vendors. The chapter starts with an overview of the adoption of this technique in most-spread mobile Operating Systems. Then, two experiments are performed. The first one studies the spread of randomized addresses used in the wild (using available datasets), according to one indicator that randomization is used, the Locally Administered bit. The second one is a case study of 6 devices supposedly implementing the technique. We look at how these devices do so, and what flaws limit the effectiveness of these implementations at preventing tracking.

IV.1 Introduction

MAC address randomization is a technique respecting the privacy-by-design paradigm. It consists in changing the hardware identifier of a device's Wi-Fi card to a temporary and random one. It has long been promoted before vendors started actually implementing it in commercial devices [122, 81].

During the course of this PhD study, devices started using the technique, as a protection against tracking. While the possibility of tracking had been known for a long time [77, 178], the proliferation of Wi-Fi-enabled permanently-carried mobile devices made it a wide-scale issue (see section III.2.1).

MAC address randomization requires support by the driver of the network card, as well as

the operating system. Firmware support is not necessary as most cards already support MAC address change. However, for a correct integration, including per-burst change and correct addresses randomness, support of this component is required [1].

IV.2 OS Adoption of Randomization

In practice, MAC address randomization implies that probe requests no longer use the real MAC address of the device. For example, a new MAC address can be used at each scan iteration, where one scan iteration consists in sending probe requests on all usable channels. However, since a (draft) specification on MAC address randomization does not yet exist, iOS, Windows, and Linux all implemented their own variants of randomization. This raises the question whether their implementations actually guarantee privacy.

IV.2.1 iOS

Apple added MAC address randomization to its devices starting from iOS 8 [189]. In iOS 8, randomized addresses are only used while unassociated and in sleep mode [63], or associated and waking up from sleep mode [221] [2]. iOS 9 was extended to also use randomization in what Apple calls location and auto-join scans [189]. Based on our own experiments, this means that randomization is now also used when the device is active, i.e., when the screen is turned on. Martin et al. noted that, surprisingly, iOS 10 devices add a vendor-specific IE indicating that the probe requests is emitted by an iOS 10 device. When randomizing, iOS devices use random CIDs [3], and the addresses seem uniformly distributed [131].

1. See for instance comment added in Linux kernel's commit effd05ac479b.
2. A rumor (unconfirmed since its publication) claimed that the randomization algorithm in iOS 8 is backdoored, so that Apple can sell a way to track devices despite randomization: `https://medium.com/@morid1n/apple-to-license-algorithm-which-will-allow-the-tracking-of-random-mac-addresses-generated-by-ios-8-5f572eb5c9b2`, consulted on 2017.05.25
3. As a reminder, a CID is a OUI whose LA bit is set to 1 (see section I.2.1).

IV.2.2 Android

Android 6.0 uses randomization for background scans if the driver and hardware support it [1]. Support was also added to Android 5.0 in an incremental patch [131] [2]. Although Android versions before 5.0 do not support randomization, several applications supporting this feature have been released [3]. Common features of these applications are a periodical update of the MAC address to a random value, but also the manual modification of this address by the user. Note that these applications require root privilege to operate, which reduces their usability for the average user.

Android devices randomize MAC addresses using a specific CID: `DA:A1:19`. This CID was introduced in Android's source code as early as September 2014 [4] [5]. Due to the open source format of Android's source code, manufacturers often make change to the OS. As a consequence, all Android devices do not have the same behaviours. The Nexus 6 device by Motorola has been reported to use a different CID when randomizing: `92:68:C3`. Some other Motorola devices change their MAC address without setting the LA bit [131].

Troublesomely, Martin et al. notices that Android devices tend to send probe requests using their global address when manipulated: when turning on their screen or when the phone receives a call [131]. We make the same observation later in this chapter (see section IV.4).

IV.2.3 Windows

Microsoft supports randomization since Windows 10 [206]. Enabling randomization is possible if the hardware and driver support it. Interestingly, not only does Windows use random addresses for probe requests, it also uses a random address when connecting

1. `https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html`, consulted on 2017.06.09
2. See current source code `https://android.googlesource.com/platform/frameworks/opt/net/wifi/+/android-5.0.1_r1/service/java/com/android/server/wifi/WifiStateMachine.java`, consulted on 2017.07.29
3. Such as Pry-fi (`https://play.google.com/store/apps/details?id=eu.chainfire.pryfi`) or Wifi Mac Changer (`https://play.google.com/store/apps/details?id=com.wireless.macchanger`)
4. `https://android.googlesource.com/platform/frameworks/opt/net/wifi/+/099e31798ed1c675a9ad654debac96f975ebcc82%5E1..099e31798ed1c675a9ad654debac96f975ebcc82/`, consulted on 2017.07.14
5. Some early implementations were buggy. For instance, user saw the Nexus 9 randomizing its MAC address using the incorrect `00:90:4C` OUI: `https://forum.xda-developers.com/nexus-9/orig-development/kernel-fire-ice-t2930451/page205`, consulted on 2017.07.14

to a network. To ensure the client always uses the same address when connecting to a particular network, a per-network address is calculated as follows [95]:

$$addr = \text{SHA-256}(SSID, macaddr, connId, secret)[:6] \tag{IV.1}$$

Here, *SSID* is the name of the network, *macaddr* the original MAC address, and *connId* a parameter that changes if the user removes (and re-adds) the network to its PNL. The *secret* parameter is a 256-bits cryptographic random number generated during system initialization, unique per interface, and kept the same across reboots. Bits in the most significant byte of *addr* are set so it becomes a locally administered, unicast address. This hash construction is similar to the generation of IPv6 interface identifiers as proposed in RFC 7217 [RFC7217]. It assures that systems relying on fixed MAC addresses continue to work as expected, e.g., when authentication is performed based on the MAC address. Users can also manually instruct the OS to daily update the per-network address randomly.

IV.2.4 Linux

Linux added support for MAC address randomization during network scans in kernel version 3.18. The address should be randomized for each scan iteration [80]. The `mvm` module of the `iwlwifi` driver supports randomization since kernel 3.18. The `brcmfmac` driver added support for this in kernel 4.5.

On the software side, randomization can be supported by `wpa_supplicant`, the most spread software implementing Wi-Fi on Linux devices [1]. Support was introduced in 2015 with version 2.4 [2].

The privacy-oriented Linux distribution Tails [3] does not support MAC address randomization during network scans. Instead, it generates a (new) random MAC address at boot. This random address keeps the first 3 bytes of the original address, the Organization Unique Identifier (OUI), and only randomizes the last three bytes. While not as optimal as periodical address changes, it does prevent tracking over extended periods of time.

1. This is not the case for all Linux devices: Android devices do not seem to take wpa_supplicant's configuration into account regarding randomization [131].
2. `https://w1.fi/cgit/hostap/plain/wpa_supplicant/ChangeLog`, consulted on 2017.07.29
3. `https://tails.boum.org`, consulted on 2017.06.09

IV.3 Locally administered addresses use historic

Devices using other MAC addresses than their attributed one are supposed to set the locally administered bit of this address to 1 (see section I.2.1). While all firmware implementing randomization do not respect this standard, it constitutes a good indicator of MAC address randomization use among time. Apart from randomization, this bit is only set when users manually change their MAC address, or for rarely-used protocols (e.g. Wi-Fi Direct). We note, however, that some common devices (Nintendo devices), operate in P2P using an address with LA bit set to 1 [131]. Numbers presented later in this section indicate that, at a time when randomization was not integrated by vendors, LA bit use was anecdotal.

As no public study following the evolution of MAC address randomization exists, we compare LA bit use in datasets from different times and locations in table IV.1. This table lists LA bit use and the fraction of unallocated OUIs in randomized addresses in the different datasets (see section I.2.1 for details on OUI allocation).

As we cannot reliably group probe requests from devices using MAC address randomization, we cannot obtain an exact per-device count. As a solution to this issue, we use both a per-MAC-address count and a per-probe-request count to compute the results. The former gives information on the actual number of detected MAC addresses, while the latter more accurately matches a per-device count. We could not compute the per-probe requests counts for 2 datasets because we did not have access to them. We must note that the per-MAC-address counts are expectedly highly variable: as devices using randomization use multiple MAC addresses, a small number of devices can drastically increase the amount of received randomized addresses. This is especially true if devices are monitored for a long period. For example, compare both counts for the `Middleware2014` dataset.

This table indicates a clear evolution of LA bit use among time. Anecdotal in 2013, it impacted an important portion of probe requests in the beginning of 2016. Surprisingly, a very high amount of randomized addresses in most datasets use an unallocated OUI. Vendors are supposed to use registered OUIs when setting local addresses [96] [1]. In fact, the equivalent CID of an OUI is automatically bought (see section I.2.1). This might indicate that most of these received probe requests come from implementations of MAC address randomization not respecting the 802.11 norm regarding MAC address use, such

1. This document, however, does not explicitly mention systematic randomization of addresses during active scanning. To our knowledge, no official document does so.

as manually-installed randomization applications. An example of this would be a device manually configured using the `macchanger` tool with its `-b` option ("burned-in-address" option, i.e., do-not-set-LA-bit option). The per-probe request counts indicate that these random addresses actually account for a very small fraction of devices, except in the `Glimps2015` dataset. This exception is not really surprising, considering the fact that devices are seen for a short time in this dataset.

In the `Martin` dataset, a progression of several Android-related CIDs used for MAC address randomization raises the fraction of registered OUI to 4.4%. However, the release of the iOS 10 OS around the same period (2016-09) also plays in favor of the former trend of unallocated OUI use for randomization, as iOS 10 devices use unregistered OUIs when using random addresses (see section IV.2). In 2012, Musa et al. remarked a similar but slightly different trend: in a 9-month public street deployment, 14% of the observed unique MAC addresses (over 60 000) used an unallocated OUI, whose LA bit is not set [151]. We did not observe this trend in any of the datasets we have access to: this ratio barely reaches 1% in one of them, and lies between 0.0 and 0.1% in the other ones.

Table IV.1 – Fraction of MAC addresses having a Locally Administered bit set to 1 over the total number of MAC addresses, in different datasets. "LA bit %" columns indicate the fraction of MAC addresses having their LA bit set to 1. The "Unalloc." column indicates fraction of these random addresses also using an unallocated OUI. Results are displayed by MAC addresses counts, and by probe requests count.

Dataset				Results			
				Per MAC addr.		Per probe requests	
Time	Name	MAC addr.	Probe req.	LA bit %	Unalloc.	LA bit %	Unalloc.
13.02-13.05	`Sapienza`	160 000	8 000 000	0.2%	33%	0.2%	13.5%
14.12	`Middleware2014`	900	140 000	47%	99.8%	1.5%	99.8%
15.10	`Lab`	1 300	120 000	14%	100%	1.7%	100%
15.10-15.11	`Train station`	9 700	110 000	23%	97.2%	10.0%	89.1%
15.12	`Glimps2015`	83 000	120 000	66.2%	99.0%	57.7%	98.9%
16.01-16.02	`Belgium`	3 700	200 000	48.8%	99.3%	2.8%	99.3%
15.01-16.12	`Martin`	2 600 000	66 000 000	53.8%	95.5%		
15.12-17.06	`Madeira`	13 000 000	300 000 000	99.8%	99.4%		

IV.4 Case study: analysis of Randomization implementations

We studied probing and MAC address randomization patterns of several phones performing MAC address randomization. Our goal was a better understanding of probing patterns and random MAC address use and change frequency. These factors influence the usefulness of MAC address randomization in terms of privacy. For instance, the question of whether random addresses change in every burst or not impacts our ability to perform timing attacks, as described in chapter VI. During the study, we identified flaws in the randomization pattern, leading to reused addresses in several devices. We contacted vendors of affected components to signal the issues.

Studied devices' models are: Nexus 6P, Nexus 5X, OnePlus 3, iPhone 6, iPhone 7 and iPad 2. All of these devices are so-called "flagships", i.e., expensive models made by vendors to demonstrate and advertise their OS capabilities. We only consider such devices because more efforts are put into integrating new capabilities (OS and chipset) into these models than for cheaper ones. As a result, they were the only devices handling MAC address randomization at the time of the study.

We give more details on the tested devices:

— The Nexus 6P is a phone manufactured by Huawei and developed by Google. It was released in September to December of 2015 (depending on the country) and originally ran Android 6.0. On this device, MAC address randomization is handled by the BCM4358 chipset [1].

— The Nexus 5X is a model manufactured by LG Electronics and co-developed by Google. It was released at the same time as the Nexus 6P, and can be considered a slightly cheaper version of the latter. For this device, MAC address randomization is handled by the QCA6174 chipset manufactured by Qualcomm [2].

— The OnePlus 3 model A3000 is a phone developed and manufactured by the OnePlus company, and released in mid-2016. It uses the same Wi-Fi chipset as the Nexus 5X (QCA6174) [3].

— iPhone 6 and iPad 2 are two popular devices developed and manufactured by Apple.

1. https://www.ifixit.com/Guide/Vue+%C3%A9clat%C3%A9e+du+Nexus+6P/51660, consulted on 2017.07.16

2. https://www.ifixit.com/Guide/Vue+%C3%A9clat%C3%A9e+du+Nexus+5X/51318, consulted on 2017.07.16

3. https://forum.xda-developers.com/oneplus-3t/help/wifi-chip-oneplus-3tac6174-t3509914, consulted on 2017.07.16

Name	OS on release	Release	Developer	Manufacturer	Chipset
Nexus 6P	Android 6.0	2015-09	Google	Huawei	BCM4358 (Broadcom)
Nexus 5X	Android 6.0	2015-09	Google	LG Electronics	QCA6174 (Qualcomm)
OnePlus 3	OxygenOS	2016-06	OnePlus	OnePlus	QCA6174 (Qualcomm)
iPad 2	iOS 4.3	2011-03	Apple	Apple	BCM43291HKUBC (Broadcom)
iPhone 6	iOS 8	2014-09	Apple	Apple	339S0228 (Murata)
iPhone 7	iOS 10.0	2016-09	Apple	Apple	339S00199 (Murata)

Table IV.2 – Characteristics of the studied devices

To handle Wi-Fi, the iPad 2 uses the Broadcom BCM43291HKUBC chipset [1] while the iPhone 6 uses Murata chipset of reference 339S0228 [2].

— iPhone 7 is the successor of the iPhone 6, still relying on a wireless chipset manufactured by the same company, of reference 339S00199 [3].

The device characteristics are summarized in table IV.2.

IV.4.1 Protocol

Due to the difficulty of finding and obtaining some access to recent devices, and because different models have different behaviours, we had to adapt our captures to the different situations. As a result, the methodology varies across devices. Our ability to make precise measurements and to filter captures in a reliable way varies across the cases.

Captures: Depending on available hardware, we performed the captures using different numbers and models of antennas. We recorded all captures using one or several instances of the `tshark` program operating on different channels, including both 2.4 GHz and 5 GHz bands. These channels were fixed for a given interface, except for the OnePlus A3000 capture which took advantage of IoTScanner's channel-hopping feature [188]. Note that an interface capturing on a defined channel often receives frames from other channels [4]. Recording hardware included multiple TP-Link TL-WN722N Wi-Fi USB dongles, a laptop's Intel Wireless-AC 8260 network card, and a "Wi-Pi" USB Wi-Fi dongle. Most devices did not have an external source of Internet connection. The iPhone 6 did have a

1. `https://fr.ifixit.com/Teardown/iPad+2+Wi-Fi+EMC+2415+Teardown/5071`, consulted on 2017.07.16
2. `https://www.ifixit.com/Teardown/iPhone+6+Teardown/29213`, consulted on 2017.07.16
3. `http://www.techinsights.com/about-techinsights/overview/blog/apple-iphone-7-teardown/`, consulted on 2016.07.16
4. During our captures, we sometimes captured the same frame using interfaces on channels as far as 1 and 9.

Device	Channels	Hardware	External connection
Nexus 6P	1, 5, 9, 13, 36, 64	TP-Link, Intel 8260 (chan 36)	No
Nexus 5X	1, 6, 11	TP-Link, built-in Wi-Fi card	No
OnePlus A3000	channel-hopping	TP-Link	Unknown
iPad 2 and iPhone 6	1	Intel 8260	No
iPhone 7	Unknown	Wi-Pi	Yes

Table IV.3 – Captures

Case Name	Description	Duration
untouched	The phone is turned off, unassociated and not manipulated.	23h
associated	The phone is associated to an access point but not manipulated nor moved	13h30
manipulated	The phone is manipulated every 5-10 minutes, associated to an AP but not moved (except when manipulated). Each time the phone is manipulated, it is turned on, unlocked, and a random app is opened.	4h
moving	The phone is not associated to an AP. It is placed in a person's pocket and not manipulated while the person moves in a small room.	1h40

Table IV.4 – Description of the different captures of the Nexus 6P.

SIM card, but was inside a Faraday cage. Information about the captures is summarized in table IV.3.

Use cases: For some devices, we performed several captures, corresponding to different device usages. These use cases are described in table IV.4 for the Nexus 6P and in table IV.1 for the Nexus 5X. During its 36-hour long capture, the OnePlus 3 was unassociated to an AP, and its usage is unknown (but mostly untouched). The iPad 2, the iPhone 6 and the iPhone 7 were manipulated during the captures: these were quick captures we made in a Faraday cage in November 2015, when iOS 9 had just been released and we wanted to test MAC address randomization, which was advertised by Apple as being improved compared to iOS 8 [221]. The iPhone 7 was a 20-minute capture performed while we had a temporary access to a device of this model.

Case Name	Description	Registered networks	Duration
random uses	Phone usage during the capture is unknown.	1 hidden network in PNL	7h30min
untouched	Phone turned off, unassociated and not manipulated	No hidden network in PNL	40min

Figure IV.1 – Description of the different captures of the Nexus 5X.

Filtering captures: Most of the captures were made in a noisy radio environment. As a consequence, recorded captures included probe requests originating from devices different from the ones we targeted. We needed to filter these captures to only keep the target's traffic. As we studied devices who apply techniques (e.g., randomization) to precisely avoid identification, this is not a trivial task.

For the iPad 2 and iPhone 6 captures, we had access to a Faraday cage and could thus easily filter captures from ambient noise. Due to the difficulty of filtering captures performed outside of a shielded environment, we had to resort to other techniques for the other devices.

For the Nexus 6P and the Nexus 5X, we exploited the hidden network mechanism: devices do not remove the SSID of a manually-added network (i.e. hidden network) from randomized probe requests. Our procedure consists in manually adding a new network with a unique SSID, that we're later able to use for filtering. We note that this may cause devices to artificially send more probe requests than they would during normal operations, as it modifies their PNL with a type of network they would not search for if they do not know other hidden networks.

This approach could not be used for the iPhone 7: we could not add a non-existing hidden network to the PNL of the device, because the latter did not keep it registered if it did not find the network immediately. As iOS devices randomize both parts of the MAC address (including their OUI), we filtered the results by keeping all probe requests using a random address, excluding OUI registered by companies different from Apple. The latter condition removes several dozen bursts sent by a Motorola device (according to the OUI). Note that this filtering process does not guarantee that we only keep probe requests from the studied device or even from a device of the same model or vendor.

The OnePlus 3 uses random address. We filter the capture file by keeping only probe requests sent using the `DA:A1:19` OUI. No probe request is sent using the device's global address. A study of the sequence numbers of the result indicates that no external device pass through this filtering process.

We summarize the different filtering approaches and their limitations, in table IV.5.

Approach	Devices	Theoretical limitations
Add specific network to PNL, keep MAC addresses using this SSID at least once	Nexus 6P, Nexus 5X (random uses)	May miss bursts sent using global addresses only, and may increase number of sent probe requests
Keep only probe requests having Google's random OUI `DA:A1:19` or target's global address	OnePlus 3	May include probe requests from other recent Android devices
Faraday cage	iPad 2, iPhone 6	None (requires access to a Faraday cage)
Keep only random addresses, excluding OUIs registered by company not manufacturing the target's model	iPhone 7	May include probe requests from other recent devices of the same manufacturer, or even from different vendors
No filtering	Nexus 5X (untouched)	May include probe requests from other devices
Building a meta-identifier out of Information Elements (see chapter V)	None	Collisions may occur, thus mistaking other devices' probe requests for target's ones
Using sequence numbers	None	Difficulty to build a reliable protocol: many sequence numbers are missed, and different devices' sequences may be mixed

Table IV.5 – Filtering approaches

IV.4.2 Results

In this section, we present the results of the different tests. We first focus on device-specific results, then present findings which are common to several devices: OUI, sequence numbers, MAC address change frequency, and a flaw in the function generating random addresses.

IV.4.2.1 Device-specific results

Nexus 6P

Random addresses: In the *associated* case, no random address is used.

Probing frequency: In the *manipulated* case, the phone keeps probing every 30 seconds, plus around 1 second afterwards. Other probes are triggered by phone manipulation,

Case	Probe frequency	MAC address change
untouched	regular with unexpected changes and some noise	every burst
associated	regular for 1 hour, then punctual	none
manipulated	regular pattern along with erratic behaviour	every burst
moving	similar to *untouched*, behaviour change after some time	every burst

Table IV.6 – Results for the Nexus 6P.

especially turning on the screen.
In the *untouched* and *moving* cases, the frequency of probing changes inexplicably several times during the capture. See table IV.6 for more details. Considering the experiment configuration, such behaviour is unclear. It seems the device regularly sends probe requests faster for a full cycle of 4096 sequence numbers.
In the *associated* case, the phone exhibited different behaviours among several tests:

— probing with its actual MAC address, with a null SSID, every 5 minutes, then every 6 minutes after some time,
— probing very sparsely (less than one burst per hour), using either its global MAC address and a null SSID or the AP's SSID, or random MAC addresses and non-null SSIDs (this might be due to the phone reassociating with the AP for any reason),
— combination of above behaviours.

All of these behaviours can all be observed in figure IV.2.

Irregular behaviours: We noticed a handful of unexpected and unexplained behaviours:

— Even in the *untouched* case, some timing irregularities happen. Background applications may be responsible for randomly provoking a burst of probe requests.
— Sometimes, the MAC address is changed in the middle of a burst.
— Sometimes, the MAC address changes very quickly. In one case, a random MAC was kept only 1 second even though the device was untouched.
— During a burst, the device sends probe requests on all channels, from 1 to 64. At the end of the burst, one supplementary probe request is sent back on channel 1 (and only on this channel), 3 seconds after the end of the burst. We assume this behaviour to be a bug, a kind of mis-synchronization between the process handling probe sending and the one handling channel changing. See bold example in Appendix A. For this reason, we constructed figure IV.2 using probe requests on channel 9 only.
— Independently from the use cases, we noticed this behaviour: when the phone has its screen turned on but is not manipulated, probe requests using a global address are sent every 15 seconds, even if the phone is locked.

Some of these behaviours have an impact on the effectiveness of randomization. The

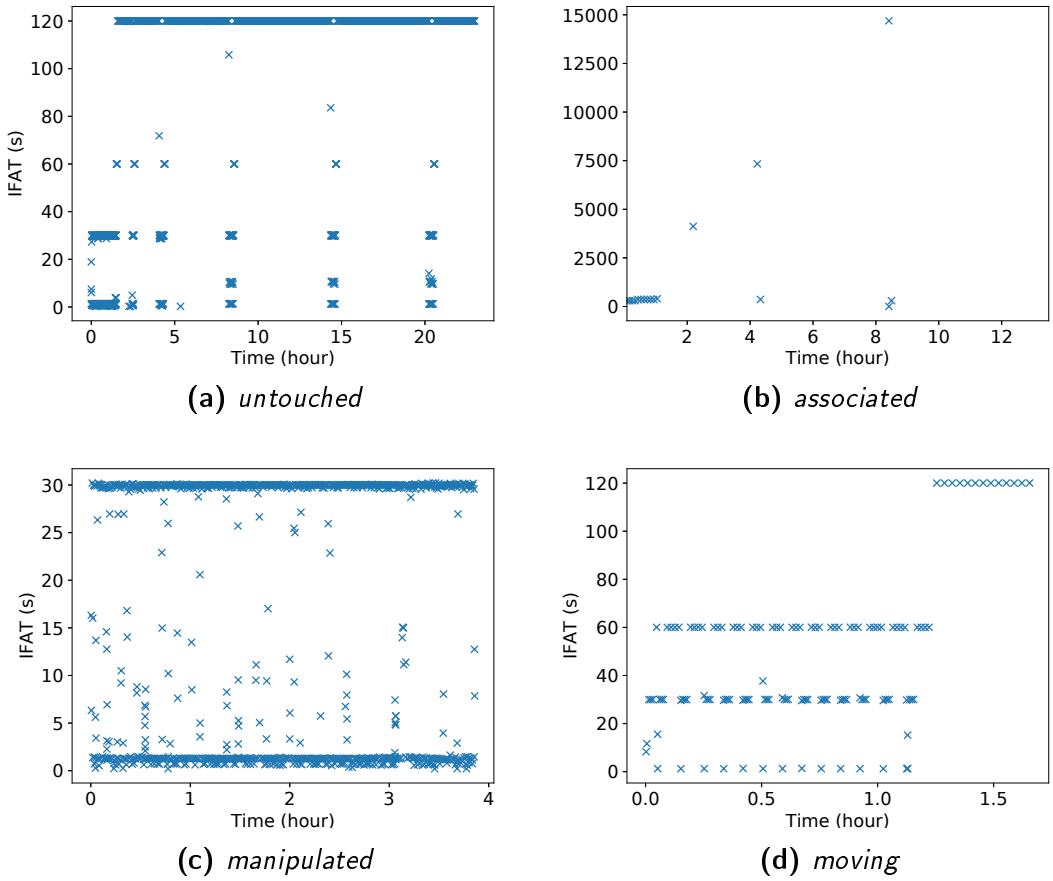


Figure IV.2 – Inter-Burst Arrival Time of probe requests on channel 9 during Nexus 6P's captures.

different use cases all produce different probe schemes. This result, already noticed by Freudiger [63], can be used to identify a device's use depending on its probing behaviour, as shown in the case of screen activation state by Jamil et al. [104].

Channels: The device sends probe requests from channel 1 to 64 successively. On channel 36 to 64, we observed probing frequency patterns similar to the ones in 2.4 GHz channels. Following a seemingly non-systematic pattern, the device may go up to channel 100 (5.5 GHz) (see Figure IV.3). No probe requests were seen on above channels (104-165) even though they're allowed in Europe. Based on sequence numbers, we can safely conclude that no probe requests are sent on these channels.

Full burst: Recording a full burst on all channels provides some interesting information. Notably:

— channel change causes a slight delay in successive probe requests,

— a full burst takes less than 100ms on a single channel, but about 500ms on all

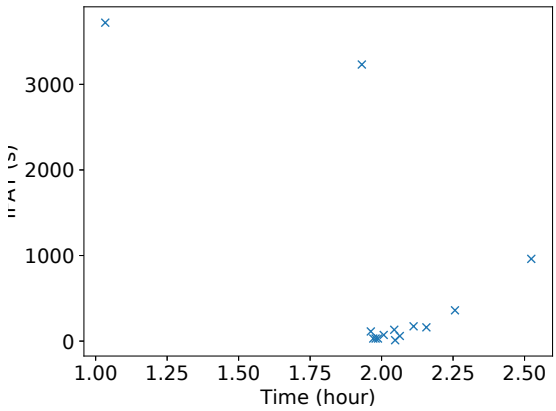


Figure IV.3 – Inter-Burst Arrival Times of probe requests on channel 100 in the *untouched* case.

channels.

See Appendix A for an example of a full burst recorded on 5 simultaneous channels.

Summary: This device exhibits behaviours limiting the effectiveness of randomization: it notably keeps probing using its global address in some cases (when associated, when the screen is on). While the device exhibits regular timing patterns, these patterns change according to the phone's usage and other unexplained factors. Probing pattern in some 5 GHz channels is irregular.

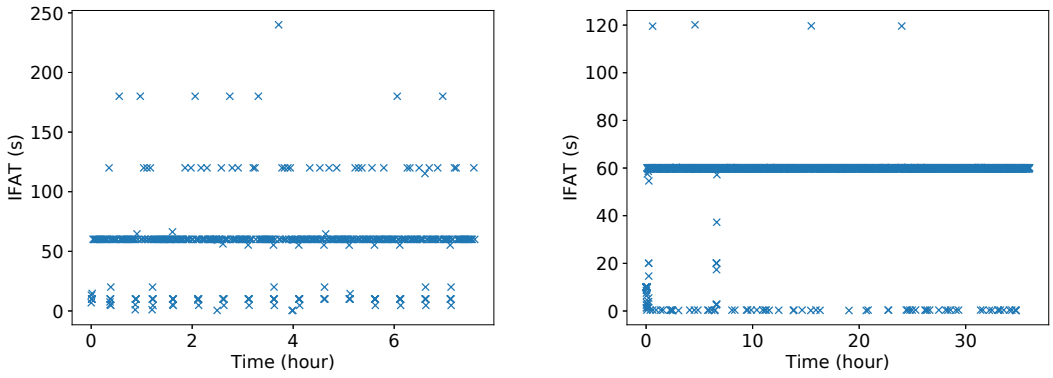
Nexus 5X

We notice some irregularities:

- Sequence numbers only go up to 2047 instead of the standard 4095.
- In the *random uses* case, the device sends probe requests using its regular MAC address every exact 30 minutes, plus punctually and irregularly at some other times. This behaviour is not seen in the *untouched* case.
- In the *random uses* case, all probe requests using a random address are directed ones (the bursts only contain the probe request looking for our manually-added local AP). This means that without this SSID, the device might not send any random probe request at all. Bursts using the global address contain both directed and broadcast probe requests.

These irregularities, especially the first and the second ones, have an impact on privacy. They may be used to fingerprint this device's model.

Probe request timings for this device are plotted in figure IV.4. We only plot the *random cases* case since the other one cannot be filtered reliably. In this figure, we acknowledge



(a) Nexus 5X **(b)** OnePlus 3

Figure IV.4 – Inter-Burst Arrival Time of probe requests during Nexus 5X's *random uses* and OnePlus 3's captures.

a relatively stable probing period of 60 seconds (multiples can be attributed to missed frames in the capture file), along with frequent switches to a 10-second frequency. Local variations might be attributed to phone manipulations. Although it's still left to be shown, we believe a regular timing in the probing pattern may help defeating MAC address randomization [1].

Summary: An irregularity in sequence numbers makes this device subject to fingerprinting. The global address is broadcast in some cases for unknown reasons. Probing pattern is very regular.

OnePlus 3

As for previous devices, the device randomizes its MAC address using a different address for every burst. The global MAC address is never used, and no SSID is added to any probe request. A burst is sent every 10 seconds at first, then the probing frequency changes after 5 minutes. Then, probe requests are sent every exact 60 seconds, with some bursts being sent after a delay of less than 500ms. These behaviours can be seen in figure IV.4.

Summary: The global address is never used. Probing pattern is very regular.

iPad 2 and iPhone 6

We did not manage to make the iPhone 6 send any probe request using a random MAC address, despite trying different combinations of configurations (cellular data and GPS

1. We only use intra-burst timing in chapter VI.

activation, etc.) and manipulations (untouched for several minutes, manipulated, etc.).

The iPad 2 was seen sending such probe requests, while both untouched and manipulated. However, the behaviour was not systematic and the conditions to make the device start using random addresses are unknown. All bursts using random addresses added the full PNL of the device in the probe requests' SSIDs, along with broadcast probe requests within the same bursts. Adding the whole PNL to random addresses defeats the purpose of randomization.

Difference of behaviours regarding MAC address randomization between both devices might have something to do with the iPhone not finding cellular network or any Internet connectivity inside the Faraday cage. MAC address randomization is notably used in iOS 9 to report scan results per location to Apple servers [1]. Being less expected than phones to have a permanent connectivity, iPads might have such services configured differently. Moreover, while some previous works managed to trigger randomization in iOS devices, others obtained the same result as ours [18].

Summary: We could not make the iPhone 6 use any random address. The iPad 2 sometimes uses random addresses under unknown conditions, sending its whole PNL alongside.

iPhone 7 (Partial results)

The device connected automatically to a local open network, and stopped sending any probe request using a random address. Once the network was removed from its PNL, it started sending probe requests using random addresses. It is unknown whether a user action originally associated the phone with the open network.

When associated with an open network, no probe request passing the filtering process was recorded. All probe requests were broadcast and used random addresses.

Summary: All probe requests use random address. The phone stops sending any probe request once associated.

1. `https://support.apple.com/en-gb/HT203033`, consulted on 2017.07.16

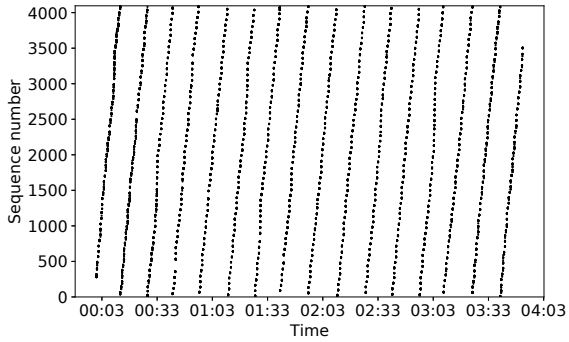


Figure IV.5 – Sequence numbers among time in Nexus 6P's *manipulated* case.

IV.4.2.2 CID use in randomization

All tested Android devices use Android's random OUI (CID) when randomizing the MAC address: `DA:A1:19`, instead of the OUI of their global (real) address. When randomizing, iOS devices randomize the whole address, and do not use a registered Apple OUI. These observations are coherent with the state of the art.

IV.4.2.3 Sequence numbers

For the Nexus 6P, the sequence numbers are normal, i.e., contiguous from 0 to 4095, as seen in figure IV.5. For the Nexus 5X and the OnePlus A3000, sequence numbers are contiguous, but the cycle stops at 2047.

Sequence number behaviours on iOS devices are more complex. When using random addresses, sequence numbers are contiguous. However, devices seem to maintain a different counter for sequence numbers added to probe requests sent using global addresses. Moreover, this counter sporadically decreases by several hundreds, and never seems to go further than about 350 as a consequence. This behaviour is a good thing in terms of privacy: maintaining different counters prevents any link from being made using sequence numbers between global and random addresses. However, we would advocate a stronger anonymization patterns for the sequence numbers: they should either be totally random, or maintaining a fixed value (e.g. 0). To our knowledge, sequence numbers have no use in probe requests, and randomizing them would not disrupt any protocol.

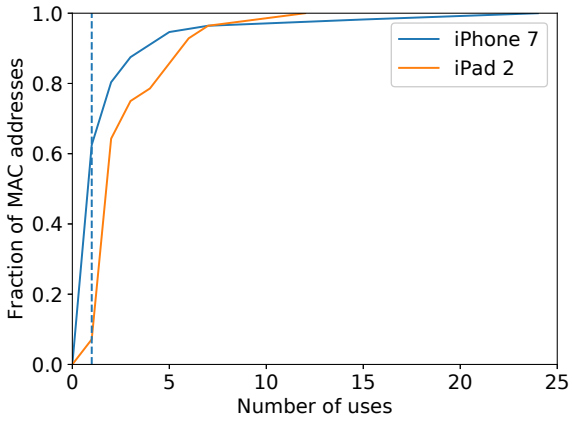


Figure IV.6 – CDF of the number of successive bursts using the same random address for the iPhone 7 and the iPad 2.

IV.4.2.4 MAC address change frequency

In the Nexus 6P, Nexus 5X and OnePlus A3000, random addresses last for only one burst. While we do see addresses used among several consecutive bursts, we attribute these to the MAC address reuse discussed above.

The situation is less clear for the iPhone 7, in which 43% of addresses are seen for longer than one burst. According to statistical tests performed by Martin et al., iOS devices do not exhibit a collision rate that indicates a flaw in their randomization function [131]. Thus, we cannot attribute this behaviour to MAC address reuse. iOS 9 devices uses random addresses during several bursts. This observation is confirmed by the captures of the iPad 2, in which even random addresses are used for a single burst. CDF for both devices are plotted in figure IV.6.

IV.4.2.5 Random MAC addresses reuse

In the Nexus 6P, the Nexus 5X and the OnePlus A3000, we noticed an irregularity in the way random MAC addresses are selected. A lot of them are reused, i.e., the device uses an address, changes to one or several other addresses, then switches back to the previous address. We assume this is a flaw in the random MAC addresses generator.

For the Nexus 6P, over a 23-hour period, out of the 332 recorded random addresses, 159 (48%) are reused at least once. 115 are reused more than once, and one of them is used as much as 27 times. Considering the fact that the NIC part of the MAC address can

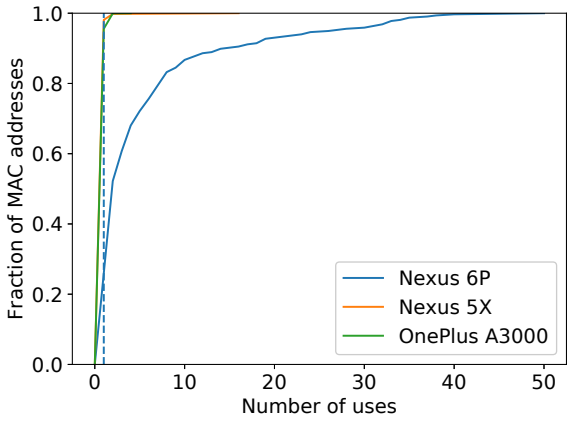


Figure IV.7 – CDF of the fraction of MAC addresses that are used more than n times.

take 2^{24} possible values, the probability of having so many collisions is extremely small.

In fact, the expected number of collisions c between n values on an address space of size D is a balls-into-bins problem [114]. The expected fractions of bins containing j balls can be estimated using the following function:

$$f(j) = \frac{1}{j!} \left(\frac{n}{D}\right)^j e^{-\frac{n}{D}} \tag{IV.2}$$

Thus, c's estimation is given by considering bins containing more than 1 ball:

$$c = (1 - f(0) - f(1)) * D \tag{IV.3}$$

In our case, $n = 332$ and $D = 2^{24}$. The average expected number of collisions c is 0.003. Hence, the selection process does not follow a uniform distribution.

We also observe some random addresses reuses in the Nexus 5X and the OnePlus A3000: 5 out of the 473 MAC addresses (1.1%) are reused (once) (rsp. 27 out of 2168 (1.2%)). In this situation, the expected average number of collisions (using equation (IV.3)) is $c = 0.007$ (rsp. 0.140). See figure IV.7 for the distribution of reused addresses in the three devices.

For all affected devices, we did not find patterns in the way addresses are reused, or differences between reused or non-reused addresses. Also, the fraction of reused addresses over the total number of random addresses does not increase over time, while the number

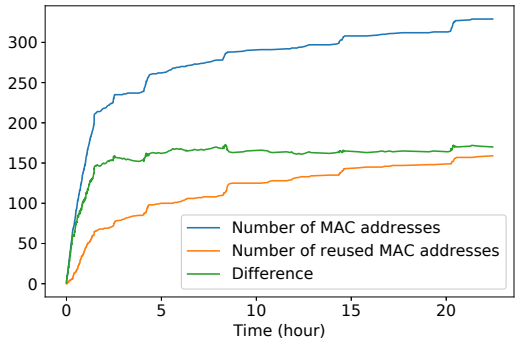


Figure IV.8 – Evolution of the number of both addresses and reused addresses among time in Nexus 6P's *untouched* case.

of addresses does. In other words, both the number of reused addresses and the number of not-reused addresses increase over-time (see figure IV.8). This indicates that devices do not have a fixed subset of reusable addresses.

Repetition of the reused addresses along time: We performed another 2-hour long capture 2 days later, in order to determine whether these reused addresses had some temporal locality. Results indicate that this is not the case, as 71 of the 332 random addresses of the first capture are seen again in the later one. In a new 2-day-long capture 6 days later, after rebooting the phone, a lot of these addresses are still reused: among the 323 MAC addresses seen in this new capture, 181 (56%) are shared with the first capture. Some of them are reused up to 62 times during that period of 2 days.

Cross-devices address reuse: The Nexus 6P shares no common address with the other 2 devices. However, the other two devices share 9 random MAC addresses (out of 2168 for the OnePlus 3 and 473 for the Nexus 5X). These 9 addresses are *not* reused addresses for these devices, i.e., each device uses them once. Using the previous formula, the expected number of collisions for $2168 + 473$ addresses is much lower: 0.21. As a reminder, the Nexus 6P use the BCM4358 Wi-Fi chipset, while the other two devices share the QCA6174 chipset.

These observations suggest that the set of reused MAC addresses might be used as a pseudo-identifier for the device. One could select the most reused addresses of a device, and use them as pseudo-identifiers to attribute them to this device in later captures. As these sets seem influenced by the chipset model, this would constitute an implementation fingerprinting. However, this requires confirmation by testing this hypothesis using a larger dataset.

	Nexus 6P	Nexus 5X	OnePlus 3	iPad 2	iPhone 6	iPhone 7
Random MAC address	●	●	●	●	○	●
Random sequence numbers	○	○	○	◐	◐	◐
No SSID in random probes	○	○	○	○	○	
Global address not leaked*	○	○	●	○	○	
Address changed each burst	●	●	●	○		○
No reused addresses	○	○	○			
Random OUI**	◐	◐	◐	●		●
No regular timing pattern	○	○	○	○	○	

Table IV.7 – Summary of the different characteristics of the devices having an impact on a correct implementation of MAC address randomization, in order of importance. ●: implemented; ○: not implemented; ◐: partially implemented; empty case: data insufficient to make a conclusion. *The difference in this row between the Nexus 5X and the OnePlus 3 may be attributed to the difference in the use cases, as their chipset is the same. **We put "partially implemented" for Android devices because they use a common OUI which leaks their OS but not their model and actually hides the manufacturer. One may argue that this is sufficient for a correct randomization implementation.

Conclusion: To sum up, we observe unusual and time-independent address reuse at a high rate for the Nexus 6P, and a lower rate for the Nexus 5X and the OnePlus A3000. This indicates a flaw in the randomization function generating random MAC addresses, weakening the effectiveness of MAC address randomization. A device may be tracked according to its subset of frequently reused addresses. Devices using equal chipsets are seen sharing some random addresses, while devices using different Wi-Fi chipsets do not. This may indicate that this address reuse is at least partly model-specific. Confirmation and study of device-specificity of these reused addresses are left for future work.

This issue has been reported to Qualcomm[1], reported and acknowledged by Broadcom. We also reported it to Google engineers. They acknowledged having addressed some issues of address reuse, but it is not clear whether we encountered the same issue.

IV.4.2.6 Summary

We summarized our observations on the different implementations of MAC address randomization in table IV.7. What appears in this table is that there still is a long road to go for all devices to perfectly implement randomization. Android and iOS devices both made advances in different directions. While some efforts have been made on iOS for modifying sequence numbers' behaviours, Android happens to change the address at every burst.

1. Issue QPSIIR-522.

IV.4.3 Conclusion

Studying implementation of MAC address randomization in recent devices, we observed a diversity of behaviours, ranging from different probing frequency to various randomization schemes. We found bugs in the several chipsets used by Android devices weakening the randomness of the temporary MAC addresses. Comparing the different features which should be correctly implemented for MAC address randomization to properly protect the user's privacy, we observed a wide variety of shortcomings in all devices.

We state the difficulty of correctly implementing MAC address randomization. The 802.11 norms must be strictly followed in order to prevent fingerprinting due to model specificities. We propose guidelines for a correct implementation of MAC address randomization in the conclusion of this manuscript.

Chapter V

Devices Fingerprinting Using Probe Requests Content

This chapter studies how fingerprinting of a Wi-Fi-enabled device can be performed despite the use of MAC address randomization, by leveraging the content of its probe requests, including Information Elements (IEs) and SSIDs. A particular IE, the WPS UUID, is studied, because it can uniquely identify a device and be leveraged to recover its original MAC address. This technique is particularly efficient at fingerprinting devices: depending on the dataset, the average entropy ranges from 5.5 to 7.0 bits. We then present a tool whose aim is to compute how unique a device is, according to its Information Elements.

V.1 Introduction

It's been showed several times before that Wi-Fi-enabled devices can be tracked using a fingerprint of various features (see section II.3.2.2). In this chapter, we focus on probe request frames, so as to find methods working for both associated and unassociated Wi-Fi-enabled devices. We study the content of such frames, and compute the amount of information which can be extracted out of them. The main goal of this work is to show and quantify the trackability of unassociated Wi-Fi-enabled mobile devices.

Dataset	`Lab`	`Train station`	`Sapienza`
#MAC addr.	500	10 000	160 000
#Probe Req.	120 000	110 000	8 million
Time frame	Oct '15	Oct/Nov '15	Feb/May '13
Location	Lab	Train Station	Rome

Table V.1 – Details of the datasets of probe requests used for this study.

V.2 Datasets

Throughout the study presented in this chapter, we used 3 of the datasets introduced in section I.3 as they're the only ones for which we have sufficient information: the `Train station`, `Lab` and `Sapienza` datasets.

In all datasets, we removed probe requests sent from locally administered addresses. These are either random MAC addresses, or specially assigned ones, and in general do not remain constant. Since we use MAC addresses as unique devices identifiers to check the performance of our algorithms, they would distort our results. Finally, based on sequence numbers and device-specific IEs, we detected and removed a few devices that kept their OUI, but randomized the NIC part of their MAC address (thus not respecting the LA bit convention).

Table V.1 summarizes the characteristics of those datasets after this cleaning pass.

V.3 Fingerprinting using Information Elements

In this section, we study how much identifying information can be found in the body of probe requests besides MAC addresses and sequence numbers. In particular, we study the data carried in the frame body of probe requests in the form of Information Elements (IEs), and show that it can be used to fingerprint and identify devices. These IEs, introduced in section I.2.1, are fields added to every management frames in order to advertise the support of various functionalities by the device.

Element	Entropy (bits)			Stability			Affected devices		
	Lab	Station	Sapienza	Lab	Station	Sapienza	Lab	Station	Sapienza
HT capabilities info	3.94	4.74	3.35	96.0%	95.9%	99.6%	90.9%	90.0%	81.1%
Ordered list of tags numbers	4.23	5.24	4.10	93.6%	94.2%	91.2%	100%	100%	100%
Extended capabilities	2.59	2.57	0.064	98.5%	99.4%	99.9%	55.4%	51.3%	0.6%
HT A-MPDU parameters	2.59	2.67	2.54	97.8%	99.1%	99.7%	90.9%	90.0%	81.1%
HT MCS set bitmask	1.49	1.43	1.16	97.6%	99.0%	99.9%	90.9%	90.0%	81.1%
Supported rates	1.18	2.10	1.36	98.2%	95.9%	99.8%	100%	99.9%	100%
Interworking - access net. type	1.08	1.11	0.006	99.6%	99.6%	100.0%	47.5%	46.1%	0.04%
Extended supported rates	1.00	1.77	0.886	98.0%	96.3%	99.4%	99.1%	72.6%	99.7%
WPS UUID	0.878	0.788	0.658	98.2%	99.2%	99.6%	8.4%	5.5%	3.6%
HT extended capabilities	0.654	0.623	0.779	97.8%	98.9%	99.9%	90.9%	90.0%	81.1%
HT TxBeam Forming Cap.	0.598	0.587	0.712	97.8%	98.9%	99.9%	90.9%	90.0%	81.1%
HT Antenna Selection Cap.	0.579	0.576	0.711	98.0%	98.9%	99.9%	90.9%	90.0%	81.1%
Overall	5.48	7.03	5.65	92.5%	90.7%	88.8%	-	-	-

Table V.2 – Analysis of the Information Elements of probe requests in the considered datasets. For each item: the entropy brought by the element, the percentage of devices for which this item is stable over time, and the percentage of devices that include this item in their probe requests.

V.3.1 Entropy

We manually select a set of Information Element, based on their presence in most devices' probe requests. We evaluate the quantity of information brought by these different elements using the three datasets introduced in section V.2. Following the approach of Panopticlick [50], we empirically evaluate the amount of information provided by each element by computing its entropy in the datasets. As a reminder, the entropy of an element i is computed as follows:

$$H_i = -\sum_{j \in E_i} f_{i,j} * \log_2 f_{i,j} \tag{V.1}$$

where E_i is the domain of possible values for element i and $f_{i,j}$ is the frequency of the value j for the element i in the dataset. We consider the absence of an element as a possible value.

Results of our analysis of the IEs are presented in table V.2. The *Entropy* column presents the amount of identifying bits provided by the elements. The *Stability* column presents the fraction of devices for which the value of the element remains constant throughout the datasets. Finally, the *Affected Devices* column presents the fraction of devices that include this IE in their probe requests.

What appears in this table is that all of these elements are stable for most devices over the observation period. Since most of these IEs reflect intrinsic capabilities of the device,

there is no reason for them to change over time. Upon further inspection, it appears that elements which are not stable over time are generated by a small group of devices. Most of the studied IEs are present in almost all devices. For instance, the `HT capabilities` tag, used to advertise capabilities for the High-Throughput 802.11n standard, is the most useful one for fingerprinting. This tag includes a lot of subfields whose values vary from one device to another, providing a lot of identifying information.

There is a high diversity in the amount of information provided by the selected elements. For instance, the `HT capabilities info` provides up to 4.74 bits of entropy, while the `HT Antenna Selection Capabilities` provides only 0.711 bit in the best case. This difference can be explained by a larger element (in term of bits), and also by a variance of the value of this element.

Some differences between the datasets are likely due to their age. In particular, some features were not wide-spread yet when the `Sapienza` dataset was produced in 2013. Back then, few devices had an `Extended Capabilities` IE, while it is wide-spread as of now. Apart from this, the three datasets display the same trends for all the elements.

The *Overall* row presents the information for all the selected IEs considered together. We can observe that for 88.8% to 93.8% of devices, the included IEs as well as their values do not change over time. More importantly, the amount of information brought by all the IEs together is above 5.4 bits in all three datasets.

Note that the WPS element is not stable for all devices. This does not mean that its content varies over time, but that it is intermittently included by some devices, since we consider the lack of an element as a possible value. When the WPS element is present, it always has the same content.

The full IE list, calculated using the Panoptiphone tool (see section V.6) can be found in Appendix C.

Martin et al. noted that most Android devices (but no iOS device) use a different set of IEs when they use a random address or their real one [131]. While this is not noticeable in results in our old datasets, this may have an impact on the stability of IEs in more recent captures.

V.3.2 Anonymity sets

To further study the impact of these IEs, we evaluate their usefulness as a device identifier. For each IE fingerprint, we form a set of all devices sharing this fingerprint (called an anonymity set) and compute its size. Figure V.1 shows the distribution of the set sizes. The three datasets exhibit a similar distribution. First, we can observe that there is a significant number of devices alone in their set (leftmost impulse), which means that they have a unique fingerprint. Then, there is a large number of small groups, meaning that although these devices cannot be uniquely identified by the IE fingerprint, they are in a small anonymity set. Finally, there is a small number of large sets, meaning that a large number of devices share the same fingerprint.

This last case is likely caused by highly popular device models: they are found in large numbers and share the same characteristics. A corollary of this observation is that the identifying potential of IEs is reduced for such device models.

These results show that the IEs can serve as a unique identifier for some devices and that, for the rest of them, it can be used as a first step toward full identification.

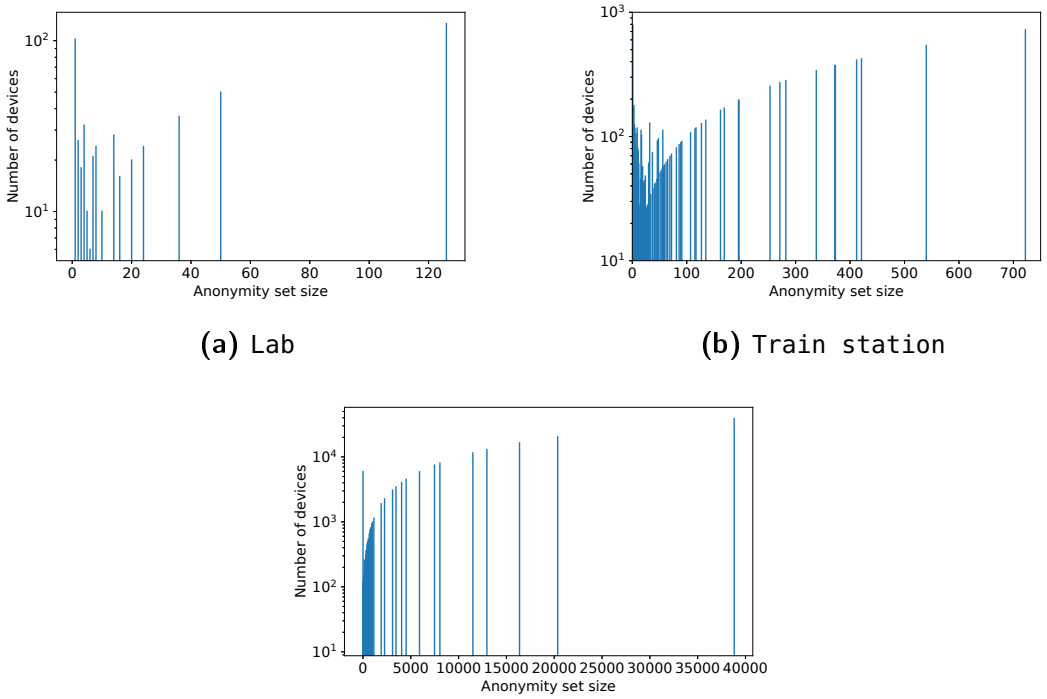
V.4 Wi-Fi Protected Setup (WPS) UUID

One of the IEs found in probe requests is dedicated to Wi-Fi Protected Setup (WPS), a protocol simplifying device pairing. We show that the unique identifier contained in this IE can be used to reveal the real MAC address of the device.

In our datasets (see table V.3), between 3.7% and 8.6% of devices broadcast at least one probe request with such an IE. One notable field of this IE is the Universally Unique Identifier (UUID) of the device, which is by definition identifying.

There is no official specification for the generation of the UUID, but the Wi-Fi Alliance recommends following the specification of RFC 4122 [RFC4122] and to derive it from the MAC address of one of the device's interfaces [212, §3.19]. More specifically, RFC 4122 specifies that the UUID should be derived from the truncation of the digest obtained from a cryptographic hashing of the MAC address.

On Linux, `wpa_supplicant` is responsible for the addition of the WPS element. It gen-



(a) `Lab` (b) `Train station`

(c) `Sapienza`

Figure V.1 – Number of devices that share the same IE fingerprint with a group (i.e., anonymity set) of various sizes.

Algorithm 1: WPS UUID generation in wpa_supplicant

Input: MAC: MAC address of an interface
Returns: 16-byte WPS UUID

$salt \leftarrow$ `0x526480f8c99b4be5a65558ed5f5d6084`
$UUID \leftarrow$ SHA-1($MAC, salt$)
$UUID[6] \leftarrow (5 \ll 4) \mid (UUID[6]$ & `0x0f`$)$
$UUID[8] \leftarrow$ `0x80` $\mid (UUID[8]$ & `0x3f`$)$

return $UUID[:16]$

erates the UUID by computing the SHA-1 hash of the MAC address with a fixed seed, before truncating it. The full algorithm is shown in Algorithm 1. Demir et al. showed that hashed MAC address are reversible through brute-forcing, due to their relatively small address space [212]. Hence it is possible to recover the MAC address that was used to generate the UUID. In other words, if the UUID is calculated in this manner, it leaks the real MAC address.

We calculated the UUID based on the MAC address as described in Algorithm 1 for the `Train station` and `Lab` datasets. This revealed that roughly 75% of all devices

using the WPS IE indeed derive the UUID from the MAC address (see table V.3). For the `Sapienza` dataset, which preserves only the OUI part of the MAC addresses, we attempted to recover the original MAC address by testing all possible values for the last three bytes of the address (together with the given OUI). This proved extremely successful, as this yielded a result for 92% of the devices. Because we do not have access to the original MAC addresses, we cannot guarantee that all of the recovered addresses are the one used as the Wi-Fi MAC address. Indeed, RFC 4122 recommends using the address of one of the interfaces, meaning other MAC addresses, such as the Bluetooth one, can be used. In a later work extending our own, Martin et al. found that, in their dataset, all devices using the `DA:A1:19` and `92:68:C3` OUIs use their actual MAC address to generate WPS's UUID [131]. It's still unclear whether all devices do the same. The details of WPS UUIDs provenance can be found in section V.4.1. We informed the authors of the `Sapienza` dataset about these de-anonymization issues.

Using the same method, we tested our own datasets again, this time exhaustively testing all possible values for the last three bytes of the MAC address, while keeping the advertised OUI. This uncovered 7 new MAC addresses for the `Train station` dataset, and none for `Lab`. These 7 addresses are all one bit away from the Wi-Fi MAC address of the device, indicating that they are the address of another interface (e.g., the Bluetooth address). We also found a few devices using bogus UUIDs (`12:34:56`... or `00:00:00`...). We conclude that, at the exception of devices using bogus UUIDs, the WPS element is a unique identifier in all our datasets. Moreover, the UUID field of the WPS element can be used to reveal the real MAC address of a device.

Martin et al. reproduced this attack on their more recent dataset, limiting themselves to UUID added in frames using an address having an Android random CID [1]. First, they noted that 29% of these include the WPS IE, which constitute a progression compared to table V.2 (even though this table considers *all* addresses). Building a 2.5 TB hash table of addresses owned by manufacturers implementing randomization, they're able to reverse any UUID in less than 1s. They validate the hypothesis that reversed addresses are actual global addresses, at least for Android devices for which they have both random and global addresses [2] [131].

1. i.e., either `DA:A1:19` or `92:68:C3`.
2. We assume they use the UUID itself to link both addresses to a device.

Table V.3 – Results of the WPS UUID re-identification attack

Dataset	Clients with WPS a tag	Successfully reversed UUIDs
`Lab`	8.4%	76.1% (35/46)
`Train station`	5.5%	73.9% (391/529)
`Sapienza`	3.6%	92.0% (5378/5844)

V.4.1 Types of WPS UUIDs in the `Sapienza` dataset

We go even further and attempt to fully determine the provenance of all UUIDs in the `Sapienza` dataset. Among the 5844 UUIDs:

- 5378 (92.0%) use Algorithm 1 using the OUI (or NIC) of the Wi-Fi MAC address used when probing,
- 266 (4.6%) are fixed and unrelated to a MAC address. There are 4 possible values:
  - `00:00:00:00:00:00:00:00:00:00:00:00:00:00:00:00`
  - `12:34:56:78:9a:bc:de:f0:12:34:56:78:9a:bc:de:f0`
  - `12:34:56:78:9a:bc:de:f0:12:34:56:78:9a:bc:de:f1`
  - `22:21:02:03:04:05:06:07:08:09:0a:0b:0c:0d:0e:0f`
- 154 (2.6%) set the LA bit on their own MAC address before using Algorithm 1, even though they send probe requests without this bit set. We estimate that the reason for this is that `wpa_supplicant`[1] uses the MAC address of another interface. In particular, devices such as the Nexus 4 create a second (virtual) Wi-Fi interface for P2P (Wi-Fi Direct). The MAC address of this interface equals the address of the real interface, but with the local bit set. Additionally, these devices start `wpa_supplicant` and pass this virtual interface as the first parameter to `wpa_supplicant`. This causes `wpa_supplicant` to derive the UUID based on the MAC address of the virtual P2P interface.
- 24 (0.4%) display the MAC address directly and complete with fixed bytes, e.g. `00:a0:96:01:27:84` → `00:00:00:00:00:00:10:10:80:00:`**`00:a0:96:7b:fd:91`**,
- 9 (0.2%) display the MAC address directly and complete with seemingly random bytes, e.g. `00:90:a9:01:27:62` → `0a:f6:73:74:`**`00:90:a9`**`:c5:9e:05:01:69:0a:05:2d:60` (we do not have the NIC part in this dataset),
- 2 (0.0%) use an invalid format (wrong UUID version number),
- 1 (0.0%) advertise a name-based UUID using MD5 hashing (we do not know if this is actually the case)

1. or another component using the same algorithm with the same salt

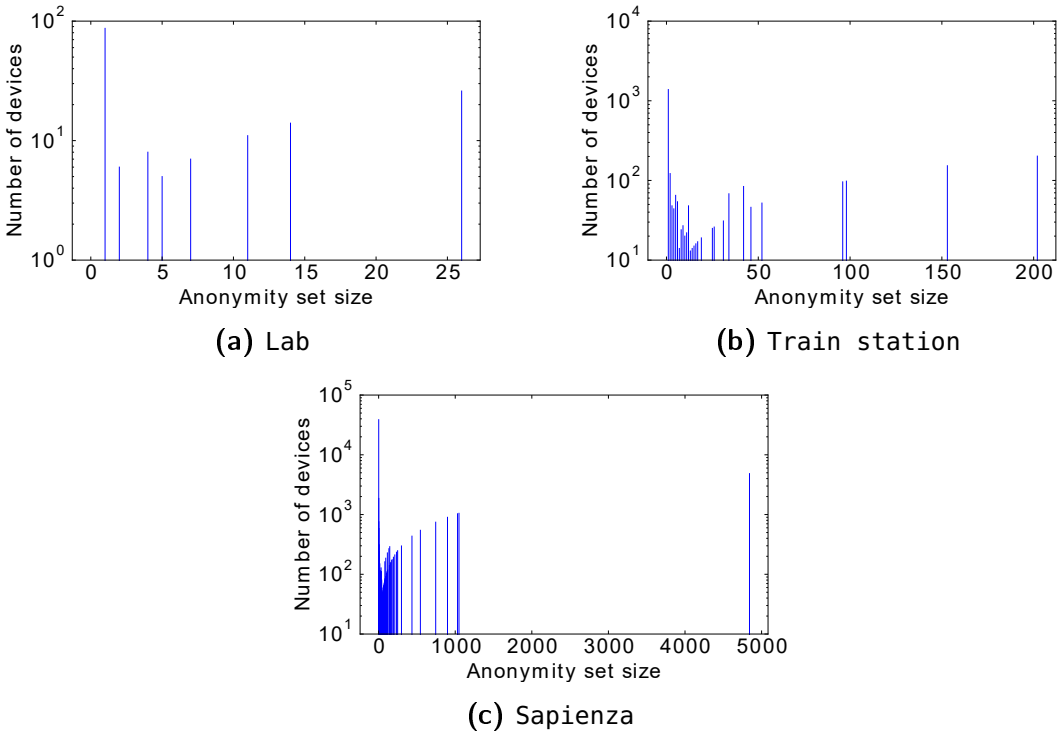


Figure V.2 – Number of devices that share the same SSID fingerprint with a group (i.e., anonymity set) of various sizes.

— 1 (0.0%) is random (UUID version number 4),
— 1 (0.0%) uses Algorithm 1 on an unknown, unallocated MAC address,
— 8 (0.1%) are left to identify: they advertise a name-based (SHA-1) UUID (version number 5), but attempting to apply Algorithm 1 on all possible MAC addresses does not yield any result. Based on the device name in the WPS IE, these are all MacBooks. These devices probably use a custom algorithm, or `wpa_supplicant`'s algorithm with a different salt.

In other words, if we want to retrieve MAC addresses:

— 95.2% of UUID can be reversed,
— 4.6% cannot be reversed,
— we do not know whether the remaining 0.2% can be reversed.

V.5 Fingerprinting using SSIDs

Probe requests include a Service Set Identifier (SSID) element, which is used to specify a network searched by the device. It's been shown in the past that SSIDs can be leveraged

to track devices [77, 37]. We go further by showing that the SSID fingerprint, i.e., the list of SSIDs searched by a device, can be a unique identifier. Devices including this element send multiple probe requests within a burst to cover all the SSIDs in their Preferred Network List (one probe for each network). During each scan iteration, devices send an ordered burst of probe requests over a small timeframe. Unlike other IEs, collecting SSIDs searched by a device therefore requires to collect the information contained in several probes sent during a burst.

Although the practice of putting SSIDs in probe requests is progressively abandoned for obvious privacy reasons, especially by devices using MAC address randomization, it is still observed for a number of reasons. First, some active devices are not up-to-date and are still running an OS that does not include this privacy-enhancing modification. Second, using a probe request with an SSID is the only way to discover a hidden Access Point. No matter how up-to-date the OS is, a device with manually-configured hidden networks will broadcast the corresponding SSIDs. Finally, we have observed that some recent devices like the iPad 2 running iOS 9.1 or the OnePlus One running Android 5.1.1 broadcast probe requests with SSIDs when waking up from sleep mode (see section for the former IV.4). We conjecture that this is because some OSes, as a way to speed up the network-reactivation process, offer separate APIs to initiate background and on-demand (wake up) scans.

In our datasets, we found that 29.9% to 36.4% of devices broadcast at least one SSID. Among these, 53% to 64.8% broadcast a unique list of SSIDs. Therefore, this list can be used as an additional unique identifier to track devices.

Using the same method as for IEs, we computed the distribution of anonymity sets for SSIDs. The results are shown in figure V.2. For readability, we removed the empty SSID list, corresponding to devices which do not broadcast any SSID. They would add a single large anonymity set on the far right of all figures.

As for IE fingerprints, the three datasets exhibit a similar distribution. For instance, in the `Lab` dataset, the far left bar indicated that 87 SSID fingerprints are unique, while the bar at the opposite side of the figure indicates that 26 devices share the same fingerprint. Apart from these extreme values, it appears that the anonymity sets of devices sending SSIDs have a small size ($< 6\%$ of the number of devices). This makes the SSID fingerprint a good tool for identifying and tracking devices.

V.6 Application: a tool to calculate device uniqueness

We introduce Panoptiphone [146], a user-friendly tool to shed light on the trackability of Wi-Fi-enabled devices, even when they are using industry-standard techniques such as MAC address randomization. We developed this tool to raise awareness on the necessity to make deeper modifications on the Wi-Fi 802.11 protocol regarding information contained in probe requests that simple identifier randomization.

Panoptiphone is inspired by the web browser fingerprinting tool Panopticlick [50], and aims to show the identifying information that can be found in the frames broadcast by a Wi-Fi-enabled device. Information is passively collected from devices that have their Wi-Fi interface enabled, even if they are not connected to an Access Point. *Panoptiphone* uses this information to create a fingerprint of the device and empirically evaluate its uniqueness among a database of fingerprints. The user is then shown how much identifying information their device is leaking through Wi-Fi and how unique it is.

V.6.1 Uniqueness evaluation

The goal of *Panoptiphone* is to exhibit the trackability of a device by evaluating its uniqueness. This evaluation is based on the fingerprint built using the IEs found in the probe requests sent by this device. The uniqueness is evaluated with regard to a database of fingerprints. Following the approach of Panopticlick, we consider two metrics to evaluate this uniqueness: the anonymity set size that corresponds to the number of devices that are sharing the same fingerprint, and the entropy that quantifies the amount of identifying information provided by IEs. They are computed as described earlier in this chapter (sections V.3.1 and V.3.2).

V.6.2 The Panoptiphone tool

The *Panoptiphone* tool is based on a three-step process. First, radio signals emitted by a device are captured through a Wi-Fi interface in monitor mode, then the resulting data is analyzed to evaluate the uniqueness of the device, and finally the result is displayed as a feedback to the user. The architecture of the tool is presented on figure V.3.

To capture data, our tool only requires a Wi-Fi card supporting monitor mode. On a

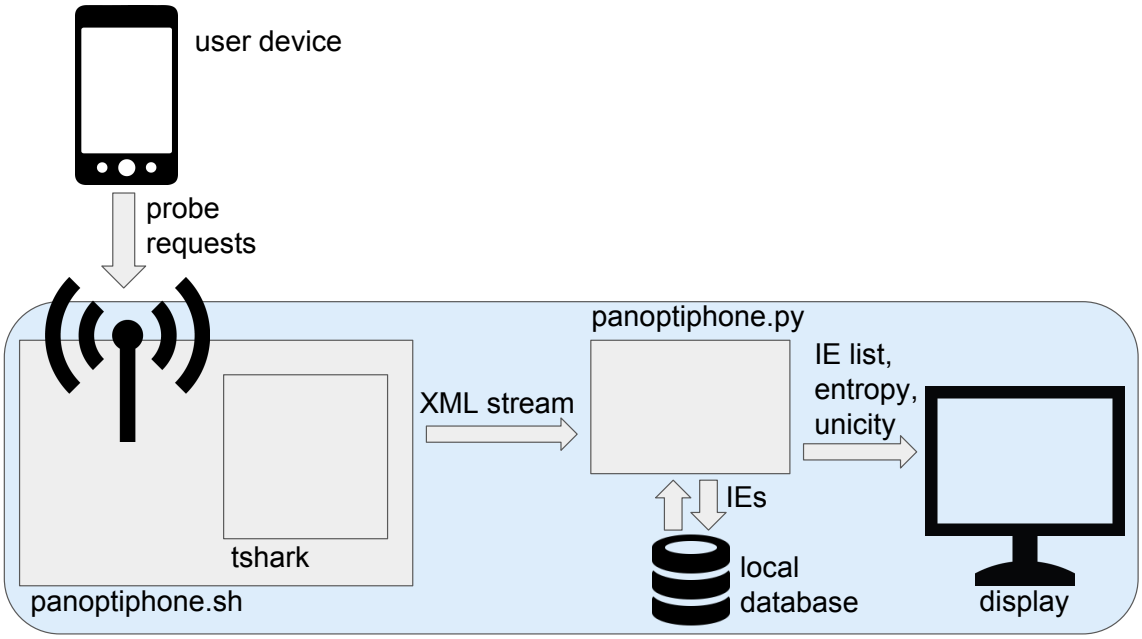


Figure V.3 – Architecture of the Panoptiphone system.

modern Linux system, this is the case for most basic off-the-shelf cards. Using an external USB dongle can simplify the estimation of proximity of users' devices, but is not necessary. Adjacent devices are detected using signal strength.

The tool is composed of two scripts. The first one, `panoptiphone.sh`, is a small bash script configuring the Wi-Fi interface and launching `tshark` with appropriate options. Its output can then be parsed in real-time by the second script, `panoptiphone.py`. The latter is a python script making the computations using and storing information in a local database, and displaying results. The display includes the list of Information Elements, which are presented using their `libpcap` name, along with metrics for each element.

We rely on a database of fingerprints obtained from the `Sapienza` dataset [15], composed of 8 millions of probe requests from 160 000 devices. Pending on the user consent, our tool can add a fingerprint of tested devices to the database.

The only information captured by our tool (in its current form) is the IEs contained in probe requests, sent by devices having an enabled Wi-Fi interface. Traffic data sent by associated devices, timing information or physical-layer information are not considered.

Once the fingerprint is captured, the privacy metrics (anonymity set size and entropy) are computed for each IE as well as for the whole fingerprint. The result of this analysis is then displayed to the user.

V.6.3 Privacy-protection measures

It is often non-trivial to manipulate private data while disclosing privacy breaches. In order to guarantee the privacy of our tool's users, we keep as little necessary information as possible. In particular, we do not keep association between the different IEs, except for the global fingerprint, which is kept SHA256-hashed. Thus, the only information that can be obtained out of it is whether a full global fingerprint has already been seen. Furthermore, we encrypt elements which are direct identifiers or contain private information: MAC addresses, WPS's UUIDs and SSIDs.

In real-time mode, the tool only detects devices in a range close to the antenna (a few centimeters), to ensure only agreeing participants will have their data collected.

V.6.4 User Interaction

Panoptiphone users are able to test the uniqueness of their Wi-Fi-enabled device. By bringing their device close to the antenna of the system, they trigger a capture event that captures the fingerprint of their device, which will be processed by the tool to compute the uniqueness of the device. The result of this process is displayed as a feedback to the user on a screen. Appendix B presents an example of several commands and their output, starting with an example output of the real-time mode.

The tool has several additional features such as the display of the global statistics of the fingerprints stored in the database as well as specific IEs.

V.7 Conclusion

In this chapter, we showed several techniques to fingerprint mobile devices by exploiting the content of their probe requests. This method weakens MAC address randomization, whose purpose is to avoid the possibility of tracking. Notably, this technique can successfully bypass Microsoft's implementation of randomization, supposed to avoid identification of devices when they associate to different networks.

One application of this fingerprinting technique could be to use it as the basis of a tracking algorithm, as presented in the original article [200]. This algorithm would track devices

over time using their fingerprints, therefore effectively defeating the purpose of MAC address randomization.

This work lead to modifications in the future Android 8.0, which will restrict Information Elements in probe requests to SSIDs and DS parameters set [89].

Chapter VI

Devices Fingerprinting Using Probe Requests Timing

This chapter studies how fingerprinting of a Wi-Fi-enabled device can be performed despite the use of MAC address randomization, by solely leveraging the timing of its probe requests. We compare several state-of-the-art algorithms and metrics, using both supervised and unsupervised learning. This technique, due to stronger conditions, appears harder to implement and less efficient than the one presented in the previous chapter. This work demonstrates the feasibility of the technique, and constitutes a first step towards a more efficient solution. Additionally, some clustering algorithms appear to be good at estimating the number of emitting devices in a capture of identifier-free probe requests.

VI.1 Introduction

We showed in previous chapter that probe requests contain enough information to form a fingerprint of a device, even without a reliable link-layer identifier. In the present chapter, we use even stronger conditions: we suppose that we don't have access to data-link layer information, except the randomized MAC address. In other words, we go further than this previous chapter, supposing the demonstrated flaws were fixed, and devices stopped adding identifying information in the content of probe requests.

Instead, we study the feasibility of tracking devices based on timing information only. More particularly, we exploit the fact that frames sent by Wi-Fi devices follow regular

patterns that can be used for time-based fingerprinting [62]. Our goal is to group identifier-free frames, ideally creating clusters of frames corresponding to their emitting devices.

The difficulty of such a technique, compared to classical fingerprinting, is that we only have a small number of frames to fingerprint a device. In many implementations, random MAC addresses change after a small number of frames have been sent (see chapter IV). As a result, we can only gather a small amount of timing information for each random MAC address. Our solution has to work with a small quantity of information. With our attacker model, we do not have previous knowledge about the communicating devices (including their number), and we do not have their fingerprints before the attack. We have to build an attack that can reliably group frames from an unknown number of devices. We investigate several techniques, from custom-made techniques to state of the art clustering and supervised learning algorithms.

In a previous work, we studied the question with the assumption that a device would send several bursts of probe requests with the same pseudonym [143]. This assumption tends to not be true anymore for recent devices, as shown in section IV.4 and in Android O privacy changes [89]. Thus, this chapter focuses on fingerprinting devices using timing information contained in a *single* burst of probe requests.

This chapter presents the following contributions. We compare several algorithms able to weaken MAC address randomization, in the sense that we are able to group frames from the same device despite the use of random MAC addresses. This attack only relies on timing information and is able to reach an accuracy (average of true positive plus true negative rates) of up to 66%.

Terminology: We remind the following definitions:

— a *burst* is a group of probe request frames sent by a device within 100ms. it typically corresponds to a scan event,

— Inter-Frame Arrival Time (IFAT) is the time difference between two frames. As this work only considers pseudonyms used in a unique burst, IFAT is only used here to designate timing of frames within the same burst.

VI.2 Threat model

We consider an attacker able to monitor the wireless signals in the vicinity of the target, using one off-the-shelf Wi-Fi card. This attacker has access to the timing information and the MAC address of each probe request frame, but not more. We make this assumption to consider a situation where the information leakage in Wi-Fi passive discovery has been fixed, as the latter has already been shown to allow device tracking in chapter V. The goal of this attacker is to classify the signals of devices in the crowd even though they use MAC address randomization, and to track individual devices among extended periods of time.

VI.3 Methodology

In this section, we present the methodology of the different tests we performed.

VI.3.1 Timing features

We define a handful of features, i.e., measurable values that can help fingerprinting devices. Their individual importance is later evaluated.

- **IFAT:** as a given burst will contain several IFATs values, we'll have a distribution of IFATs for each burst. Adapting the procedure introduced by Franklin [62], we turn these values into features by cutting the distribution of IFATs within a burst in 10 bins of 10ms each. This gives us 10 features: the percentage of IFATs within each bin.
- **Burst length:** the total duration of the burst, in μs.
- **Number of frames:** the number of frames per burst.

Each burst is turned into a vector of these features, which is used to feed the clustering and supervised learning algorithms.

Burst signature: Our incremental algorithm is using its own set of features and distance, because of its structural difference with the other tested algorithms. It is solely making use of IFAT-based signatures, slightly different from the features presented above. We

build a database of time-based signatures for the different MAC addresses, as described in [62]. We compute the distribution of IFATs in 10 bins, as described above. But here, we also compute the fraction and mean value of IFATs in each bin, which constitutes the signature $\mathcal{S}(G)$ for a burst G. For G, let F_b^G be the fraction of frames in a bin b, and M_b^G the mean IFAT value in bin b. Let $\mathcal{B}$ be the set of all possible bins, the signature $\mathcal{S}$ of group G is given by:

$$\mathcal{S}(G) = \{F_b^G, M_b^G | b \in \mathcal{B}\}$$

VI.3.2 Grouping algorithms

The objective of the attack is to defeat the use of pseudonyms at the link layer by grouping the frames belonging to a device. This task is performed by a grouping algorithm that will rely on the timing features and distances introduced in previous section. We envision several state-of-the-art algorithms, chosen for their ease of implementation and popularity.

VI.3.2.1 Incremental algorithm

We adapt our custom algorithm described in our previous article [143] and turn it into a threshold-based clustering algorithm (algorithm 2). It takes as input a capture of probe requests and outputs a mapping between the frames and a set of clusters. Ideally, each cluster should correspond to a distinct device and its associated bursts. The mapping is obtained by clustering frames that appear to originate from the same device based on timing information.

Algorithm description: For each burst, we calculate the distance between the signature of this group and every other known signature. If at least one of these distances is below a given threshold t, we choose the burst having the signature yielding minimal distance, and put the new burst in the chosen burst's cluster. Otherwise, we estimate the burst to belong to a new device, and add its signature to the database (i.e., create a new cluster).

As a result, we obtain an incremental learning algorithm: new bursts can be later tested by our algorithms, which will decide whether they are linked to a previous cluster or if they belong to a new one.

Algorithm 2: Incremental clustering algorithm

Input: $\mathcal{G}$: bursts, identified by their MAC address
t: distance threshold
d: a distance function
Returns: $\mathcal{C}$: dictionary of clusters

$\mathcal{C} \leftarrow \emptyset$
$\mathcal{D} \leftarrow \emptyset$ `// Database of signatures`
foreach $\mathcal{B} \in \mathcal{G}$ **do**
  $\mathcal{S} \leftarrow$ signature($\mathcal{B}$)
  $D \leftarrow \mathcal{D} \cup \mathcal{S}$
foreach $\mathcal{B} \in \mathcal{G}$ **do**
  $d_{min} \leftarrow min(d(\mathcal{S}, \mathcal{S}')$ where $\mathcal{S}' \in \mathcal{D})$
  if $d_{min} < t$ **then**
    $\mathcal{C}[\mathcal{S}'.mac].add(\mathcal{B}.mac)$
  else
    $\mathcal{C}[\mathcal{B}.mac] \leftarrow \mathcal{B}.mac$
return $\mathcal{C}$

VI.3.2.2 Clustering

We try several state-of-the-art clustering algorithms. We use two popular algorithms: DBSCAN [54] and k-means [130]. The former is a density-based algorithm which automatically determines the number of clusters, and ignores noise. The latter requires the user to provide the expected number of clusters. For the sake of evaluation, we will assume that this number is known, keeping in mind that the algorithm is not useable as-is. We also consider the mean shift algorithm, a centroid-based algorithm [28].

VI.3.2.3 Supervised learning

Supervised learning can also be adapted to our case: we adapt the random forests algorithm [88] to match our problem. To do so, we train a classifier on a set of labels, and use that trained classifier to link bursts of another dataset.

When given a vector v, a trained random forest classifier returns a table t_v indicating the probabilities that v belongs to each of its known labels (p_{v,l_1}, p_{v,l_2}, etc.). To estimate whether two vectors v_1 and v_2 belong to a common label, we feed both vectors to the trained classifier, and use both probability tables to compute the following number p:

$$p = \sum p_{v_1,l_i} * p_{v_2,l_i}$$

We then compare p to a threshold to decide whether both vectors can be linked.

VI.3.3 Distances

We consider several state-of-the-art distances (or metrics) for the algorithms handling distances between vectors, i.e., the various clustering algorithms, including our own [1]. We need these metrics to compare vectors and evaluate the probability that they represent frames emitted by the same device.

Euclidian distance: a simple distance that calculates the shortest path between two vectors. Euclidian distance D_E between two vectors v and w of size n is defined by:

$$D_E = \sqrt{\sum_{n}^{i=1}(v_i - w_i)^2}$$

Cosine distance [2]: a metric based on the cosine similarity, i.e., the cosine angle between two vectors. The cosine distance D_c between two vectors v and w of size n is defined by:

$$D_c = \frac{\sqrt{\sum_{n}^{i=1} v_i w_i}}{\sqrt{\sum_{n}^{i=1} v_i^2 \sqrt{\sum_{n}^{i=1} w_i^2}}}$$

Franklin's distance: The last considered distance is a modification of the one used by Franklin to fingerprint device drivers [62]. We modify Franklin's distance formula so that it respects the symmetric property of a distance [3]. This distance is closely tied to the way the signatures are built in the incremental algorithm, and is therefore only used for this one. The distance between two bursts v and w is based on their signatures. We calculate

1. The random forests algorithm does not need metrics because it bases its decisions on the feature's entropy or variance.

2. This is actually not a distance since it does not respect the triangle inequality, but it is often called a distance for convenience.

3. To do so, we apply the following modification: instead of multiplying the difference of the means by the percentage (fraction) of a single device, we multiply by the mean of the percentages of both compared devices.

the distance D_F using the following formula:

$$D_F = \sum\nolimits_{b \in \mathcal{B}} (|F_b^w - F_b^v| + \frac{(F_b^v + F_b^w)}{2} * |M_b^w - M_b^v|)$$

Fractions and means are set to 0 if the bin is empty.

VI.4 Evaluation protocol

In this section, we present our evaluation protocol: used performance metrics and datasets, features evaluation, parameters selection and validation.

VI.4.1 Evaluation metrics

To evaluate results, we adapt the de-anonymization attack using linkage tests described by Sharad and Danezis [185]. We randomly select pairs of bursts from different devices, and an equal number of pairs of bursts sent from the same device. We then test whether the different classifiers are able to correctly determine whether they were sent by the same device or not. Our null hypothesis is: "Given two bursts A and B, burst A and burst B come from the same device". Then, the four possible cases are:

- True Positive (TP): two bursts are correctly labelled as coming from the same device,
- True Negative (TN): two bursts are correctly labelled as coming from different devices,
- False Positive (FP): two bursts are labelled as coming from the same device but come from different devices,
- False Negative (FN): two burst are labelled as coming from different devices but come from the same device.

TPR, TNR, FPR and FNR refer to True Positive Rate, and so on.

Parameters are evaluated using the Overall Success Rate (OSR) [121], which, in our case, can be calculated by:

$$OSR = \frac{TPR + TNR}{2}$$

Name	Duration	MAC addr.	probe requests
`Lab`	6 days	550	120 000
`Lab` cut	1h20	67	10 000
`Lab` cut2	6 min	24	1 000
`Belgium`	11 days	1 900	200 000
`Belgium` cut	16h	273	10 000
`Belgium` cut2	15 min	65	1 000

Figure VI.1 – Datasets used to test the timing attack.

VI.4.2 Datasets

To evaluate the performances of our grouping algorithms, we rely on real-world datasets of probe requests. In these datasets, we remove any probe request using a random address (identified by their LA bit), and simulate the use of random MAC address by replacing the real MAC address of each device by a random pseudonym changing every burst. The aim of this procedure is to obtain the link between random addresses and their related device.

We also remove bursts consisting in a single probe request. Such bursts do not contain enough information, as no IFAT or burst length can be calculated. In all datasets, the quantity of such bursts amounts from 13 to 33% of the groups.

We test all algorithms on datasets of various durations and amounts of devices. These datasets are summarized in table VI.1.

VI.4.3 Validation: cross-checking

While we already used different devices for training and testing the supervised learning algorithms, we push the validation further by using a different dataset for both parts. All tests of the random forests on the `Lab` dataset are cross-checked using a part of the `Belgium` dataset of similar size, and vice versa. This evaluation is made besides of the unique dataset evaluation.

VI.5 Results

In this section, we present the results of our tests.

VI.5.1 Features importance

We first evaluate importance of the different chosen features. Using a tree-based algorithm (random forest) has the advantage of furnishing several ways to easily perform feature selection [73]. We use the mean decrease impurity method. In each constructed tree, how much variance (or entropy) reduction each feature brings can be calculated. Averaging among all trees in the forest, we can obtain a score of the importance of each feature in the classification task [21].

What appears when testing feature importance on the "`Belgium` cut" dataset is that the burst length feature is by far the most important one (69%) and could probably reach good results alone. In second come the IFATs features (23% combined), and the number of frames arrive in the last position (8%). On the larger `Belgium` dataset, the burst length feature scores even higher (86%), for a combined score of 11% for IFATs and 3% for burst length.

VI.5.2 Algorithms

We then study the quality of the considered algorithms. Results are presented in table VI.3. As a reminder, in our experimental conditions, an algorithm replying randomly to the segment linkage test would have a TPR, a TNR and an OSR of 0.5. Some results could not be computed because related algorithms are too computationally-expensive.

DBSCAN gives surprisingly consistently good results for a clustering algorithm. K-means, despite being given more information in our tests (number of clusters), performs poorly. It, however, always returns a high TNR, which may be useful for applications requiring a low false positive rate. Mean shift barely performs better than a random choice. Random forests results are not as good as DBSCAN's ones, and vary depending on the dataset size. The algorithm, however, present the advantage to be computationally cheaper than every other ones. Our incremental algorithm reaches results equivalent to DBSCAN's ones, with a constantly high TNR but a better TPR when the dataset is larger.

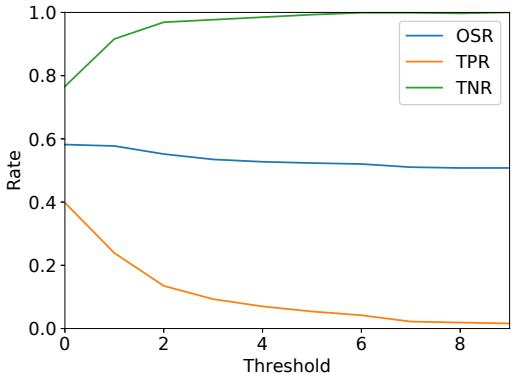


Figure VI.2 – Results of the random forest segment linkage varying the threshold parameter. OSR is slightly better for a threshold of 0, and the TPR is the highest, at the price of a lower TNR.

Dataset	DBSCAN	k-means	Mean shift	Rand Forest	Inc. alg.	Rand Forest Cross-check
`Lab`	0.78/0.64	0.11/1.00	0.88/0.12	0.40/0.76	0.46/0.94	0.42/0.75
`Lab` cut	0.43/0.94	0.18/0.97	0.87/0.08	0.65/0.60	0.30/0.96	0.62/0.45
`Lab` cut2	0.42/0.96	0.29/0.93	0.88/0.18	0.88/0.29	0.26/0.95	0.89/0.33
`Belgium`				0.32/0.75	0.45/0.96	0.31/0.77
`Belgium` cut	0.29/0.88	0.10/0.98	0.82/0.18	0.49/0.68	0.30/0.92	0.58/0.59
`Belgium` cut2	0.36/0.93	0.14/0.97	0.88/0.15	0.80/0.27	0.35/0.93	0.98/0.06
Avg. OSR	**0.66**	0.57	0.50	0.57	**0.65**	0.56

Figure VI.3 – Results of the different algorithms. Indicated valued are: TPR / TNR.

Cross-validation: Cross-checking tests tends to give poor results on small datasets, as the probability to encounter devices sending probe requests having close characteristics is small. On larger datasets, results are equivalent to the ones obtained using the same dataset for training and testing. Further tests give results similar to the ones obtained on large datasets when the training dataset is large and the testing dataset in small (i.e., more consistent results).

VI.5.3 Parameters

Varying different parameters, we evaluate their optimal values on the `Lab` dataset. We choose values maximizing the OSR, while all other parameters are fixed.

We obtain the following results:

— Threshold value for the random forests yields best results if the threshold is left at

0 (figure VI.2).

— Metrics: the cosine distance metrics gives a better OSR than the Euclidian distance for DBSCAN on all datasets (+0.05 on average). However, the latter consistently reaches a TNR of more than 93%, which may be useful if one wants to avoid false positives. Nonetheless, we perform tests using the cosine distance. For our incremental algorithm, the latter gives poor results (0.0 TNR), so we stick to our custom distance.

VI.5.4 Devices count estimation

We also explore the idea of counting devices based on probe request timing. The wide deployment of MAC address randomization has made it difficult to estimate the number of devices based on their identifier. We use the ability of some clustering algorithms (DBSCAN, mean shift, and our incremental algorithm) to estimate the number of clusters and compare it to our ground-truth value.

As our aim is that each cluster contains the bursts of an individual device, we use the capacity of evaluating the number of clusters of several algorithms to count devices. This involves all tested clustering algorithms except k-means. Table VI.4 compares the clusters counting capabilities of the different algorithms. We compute the relative errors e between estimated number of devices n_e and real number of devices n_r:

$$e = \frac{|n_r - n_e|}{n_r}$$

The mean shift algorithm tends to highly underestimate the number of clusters in all cases. DBSCAN reaches very good estimations for small to medium datasets, underestimates the number of clusters by a factor to 3 to 4 on the large `Lab` dataset, but highly underestimates the number of clusters by a factor of 30 for the largest one. The reason for this is unknown. While not perfect, this still provides a reliable estimation of the number of devices without the use of an identifier for small to large datasets. Our own algorithm drastically overestimates the amount of clusters in all cases.

Dataset	DBSCAN	Mean shift	Inc. Algo
`Lab`	73%	97%	4953%
`Lab` cut	**11%**	87%	1275%
`Lab` cut2	**0%**	83%	270%
`Belgium`	97%		8339%
`Belgium` cut	38%	90%	2628%
`Belgium` cut2	35%	79%	674%

Figure VI.4 – Relative error of the estimated number of clusters.

VI.6 Limitations

While this technique is able to distinguish between devices of different models, devices of the same models might only have different fingerprints if the length of their PNL differs. More work is required to estimate whether used features are affected by device configurations, so as to determine whether this technique alone can effectively create fingerprints of individual devices.

While performances of our algorithms are no perfect, they show that this approach is promising and open opportunities for future works. Optimizing performances are left for future research and engineering works. Potential improvements could be done by testing more algorithms, metrics, and tweaking features.

VI.7 Conclusion

Summary: We have discussed how timing of probe requests alone, in its various forms (inter-frame arrival time, burst duration) could be used to cluster bursts, effectively weakening MAC address randomization. Unlike previous work, we did this while considering that a pseudonym lasts only for a single burst. We studied a range of algorithms, including clustering and supervised learning algorithms to do so. Our results show that this approach is promising. There is a gain compared to a random oracle, demonstrating a potential for better results. Moreover, some algorithms tend to be good at estimating the number of clusters, which may reveal a good opportunity to perform device counting despite the wide deployment of MAC address randomization.

Future work: Possible extensions to this work could exploit the fact that probe requests are often periodic, i.e., considering IFATs between bursts. Moreover, the simple attacker

model could be further extended by considering advanced techniques and using several sensors. Adding the location information obtained by different sensors has potential to improve the OSR of our algorithms.

As Martin et al. figured out, the WPS flaw described in section V.4 can be exploited to safely link random addresses to the same device [131]. This would give a ground truth to evaluate our technique on actual random addresses used in the wild. This would provide an additional validation on real-world random addresses, we requires an access to a dataset possessing such information.

Countermeasure: A countermeasure to this attack would involve either regularizing or randomizing the timing pattern of emitted probe request. This is not an easy task. A first step towards achieving this would be to only send a unique frame for each burst. This can be easily done by limiting probing to broadcast probe requests. That way, devices would only send a single probe request containing a unique SSID on a given channel. The problem with such a countermeasure is that hidden APs could not be discovered anymore, and that channel switch duration would still influence the features. Adding a random delay between successive probe requests could be another potential solution.

Chapter VII

Implementation: Wombat

The chapter presents a tracking system that we developed for experiments and awareness-raising demonstrations. This system tracks users by collecting signals emitted by their Wi-Fi-enabled mobile devices, and presents information that is was able to collect to them. We describe the system implementation, present how we use it to raise public awareness about privacy issues related to Wi-Fi tracking, then discuss a new, simple and user-friendly method to allow users to opt out of such systems.

VII.1 Wi-Fi Tracking System Implementation

We developed an experimental Wi-Fi-based physical tracking system that can be used for demonstration purposes. This system effectively tracks users through their Wi-Fi-enabled mobile devices and can then show them the type and amount of data that has been collected. In addition, this experimental platform is also used to deploy and test privacy-enhancing features for physical tracking systems.

We used the system in many different contexts, including a several-month installation at the `Cité des Sciences et de l'Industrie` (City of Sciences and Industry) in Paris. The system is called Wombat, which wasn't chosen to mean anything[1], but which one could expand in a periphrasis such as "Wi-Fi Omniscient Monitoring system for Basic or Advanced Tracking".

1. The animal is cute, though.

VII.1.1 Details of Implementation

Wombat is a fully functional Wi-Fi tracking platform supporting three main features: collection, storage/processing, query/output. These three features are implemented through a distributed infrastructure composed of:

— **Sensor nodes**: small devices with wireless monitoring capabilities. They collect information sent on wireless channels and forward it to the server.

— **Central server**: the central entity of the system. It receives data sent by sensor nodes and then stores it in an internal data structure. It is also in charge of answering queries related to the stored data.

To ensure communication between the sensor nodes and the server, the *Wombat* system relies on a wired Ethernet connection. In addition, Wombat can be enriched with a *user interface* and an *opt-out node*:

— **User interface**: a device in charge of displaying detailed information about a tracked device or general statistics (see figure VII.2). The device to display can be specified manually by its MAC address or through proximity detection.

— **Opt-out node**: an element in charge of implementing an opt-out mechanism for users refusing to be tracked by the system (see section VII.2).

A description of the Wombat system with all its components is presented on figure VII.1.

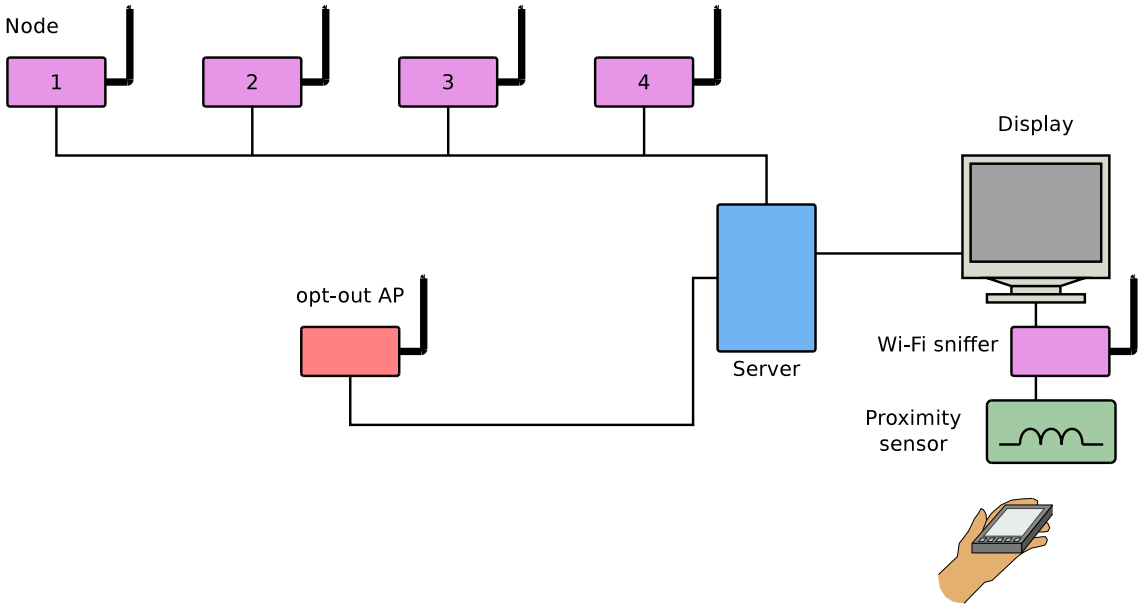


Figure VII.1 – Architecture of the Wombat system in a demonstration configuration.

On the hardware side, nodes and servers are implemented on Raspberry Pi machines.

Nodes use Wi-Fi USB dongles to collect signals. Even though Raspberry Pi model 3 possesses a wireless card, the latter cannot be turned into monitor mode.

On the software side, node and servers use the Arch Linux operating system. One of the constraints imposed by the *Cité des Sciences et de l'Industrie* for the installation is that the machines could be turned off without notice. Such a constraint is not trivial, because Raspberry Pis are not made to bear such usage: their SD card can easily get corrupted, which makes the operating system impossible to boot. Among the different possibilities to handle unexpected shutdowns, we chose the one which does not require supplementary hardware (and thus, cost). As corruption happens when the card runs out of energy during a write operation, we make sure the SD card is always booted in read-only mode. To test whether the system is indeed resistant to SD-card corruption, we perform extensive tests, detailed in section VII.1.2.

Concerning the front-end, the system possesses several possibilities:

— the front-end developed by the Fleur de Papier company [1] for the *Cité des Sciences et de l'Industrie*,

— a simpler front-end developed by ourselves for earlier installations.

The server also handles two modes to return results to the front-end:

— a query mode, where the front-end queries for data about a specific MAC address,

— a blind mode, where the server possesses its own Wi-Fi USB dongle, used to detect device proximity. Thanks to the latter, it sends data about detected close-range devices.

The query mode is mainly used for testing and simple temporary installations, while the blind mode is used for long-term installation, such as the one for the *Cité des Sciences et de l'Industrie*.

Our front-end constructs a timeline of the different places visited by a user, as seen in figure VII.2.

VII.1.2 Testing Implementation Resilience

We performed extended resiliency tests. In order to set up the system in permanent installations, we had to make sure it could work properly in such situations, where stability is necessary.

1. `http://www.fleurdepapier.com/`

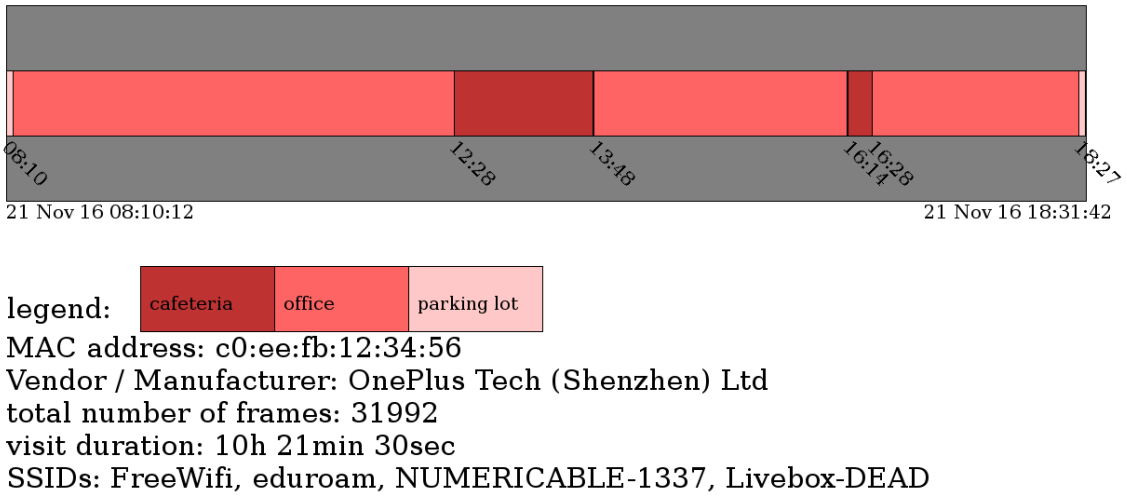


Figure VII.2 – Our front-end for the Wombat project (simulated output).

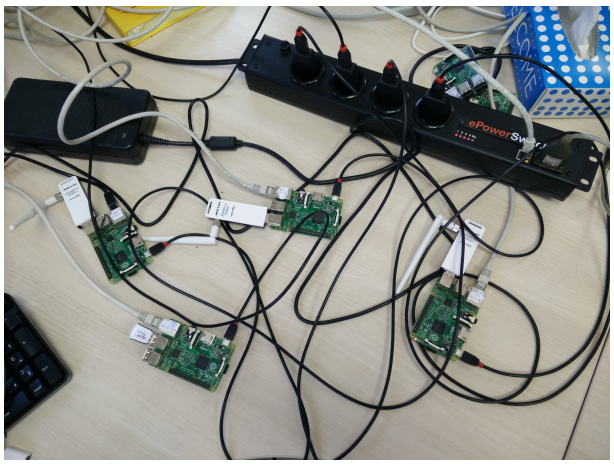


Figure VII.3 – Photography of the testing infrastructure.

To make sure the installation is resistant to sudden turn-off, we automated sudden reboot using a remotely-controllable power switch (model: ePowerSwitch 4). We plugged 4 nodes and performed a series of reboots. We rebooted all machines when a central server received a notification by all 4 machines, indicating that they had captured at least one probe request. Thus, we verified that system reboot worked end-to-end, and that no part started failing during the process. The installation can be seen on picture VII.3.

We performed a series of 700 reboots of the 4 nodes, lasting approximately 12 hours. During all these reboots, none of them failed to restart or to capture probe requests. The whole system took between 30 seconds to a maximum of 3 minutes to reboot completely,

i.e., go through the whole booting process, including network interfaces activation and time synchronization through NTP, up to reporting frames to the server. It must be noted that these tests were performed in an early phase of development. Latest versions of the system would complete the whole initialization sequence in a faster and less variable time as it does not rely on unpredictably slow NTP clients anymore.

As a second series of tests, we made sure turning off the power during boot time could not corrupt the SD card. To do so, we used our power switch to reboot the machines 50 times after 1, 2, 5 and 10 seconds. Even after these 400 reboots, all machines continued to work properly.

As a result, we can safely assume that our system is resistant to sudden reboots. After several months of service in the `Cité des Sciences`, we can conclude that this assumption is true.

VII.1.3 Possible evolutions

Different evolutions can be considered for the Wombat project.

Privacy-preserving features: The first obvious evolution is to integrate privacy-preserving techniques to the system, notably using privacy-preserving data structures, as introduced by Alaggan et al. [5].

Wireless network: Secondly, our system requires a wired link between the nodes and the server. Introducing the possibility to communicate using a wireless link (such as a mesh network) would make deployment easier and would allow more configuration possibilities.

Indoor tracking: More advanced techniques could be implemented to improve the precision of indoor positioning in small places. For instance, CSI could be used instead of RSSI (see section II.4.1.1). More advanced techniques can also involve advanced statistical methods, or learning a mapping of signal strengths in a room [41].

Active tracking: Lastly, we could extend tracking methods by considering active monitoring. The wombat system is purely passive as it never sends any frame to the tracked devices. Several techniques introduced in section II.4.1.1, such as exploiting the RTS/CTS mechanism or announcing popular SSIDs could be used to implement active tracking. As most commercial systems use only passive tracking, this would be a great innovation.

While more intrusive than passive tracking, one could imagine a technique in which a device only replies to requests if its owner explicitly allowed it to, introducing a form of opt-in.

Other technologies: On a long-term vision, we could also consider adding support for other tracking technologies described in section II.4.1.4.

VII.2 Privacy-enhancing feature: opt-out mechanism

VII.2.1 Current opt-out mechanisms and their limitations

Wi-Fi tracking systems have been criticized because they are collecting users' data without their consent. As a result, opt-out mechanisms to allow concerned users to escape tracking have been deployed [68]. These opt-out mechanisms typically involve a webpage on which the user needs to enter its device address (see section III.6).

Even though it represents a step toward increased user control, this kind of approach presents several issues, mainly related to their usability. The main issues are the following:

1. In order to find their MAC address, users need to navigate deeply into their device's settings, which can be a difficult task for non-tech-savvy users.

2. Users need to manually enter this 16-character-long identifier on the opt-out webpage, which can be a cumbersome task.

3. Subscribing to the opt-out mechanisms means that the device identifier will be sent to a third party which will store it indefinitely.

4. Multiple tracking systems may use different opt-out databases and thus require users to go through this process for each system.

5. Some devices (e.g. fitness trackers) do not provide an access to their MAC address.

It is likely that these usability issues will deter users from using this opt-out mechanism, thus preventing them from protecting their privacy. A more usable opt-out solution is therefore required.

VII.2.2 A Wi-Fi-based opt-out mechanism

We propose to use Wi-Fi as a vector to transmit the opt-out decision, by leveraging core Wi-Fi elements. More specifically, on the tracking system side, the opt-out mechanism is implemented by a primitive [1] Access Point (AP), with which Wi-Fi devices willing to opt out must associate. This idea was first mentioned by Soltani [191].

The network name (SSID) advertised by this AP is explicitly indicating the purpose of the network: opting out of a Wi-Fi tracking system. For instance, this SSID can be "`Opt-Out Wi-Fi tracking`" or "`Do not track`".

A device whose owner wants to opt out will associate with this AP. Upon such an event, the device will contact the AP in order to proceed through the association protocol. During this process, the AP will learn the MAC address of the device by parsing received frames. From this point on, the AP can consider that the corresponding device wants to opt out of the tracking system, and can thus add this address to a local blacklist. This list is maintained locally, and an expiry delay can be configured on the server so that blacklisted identifiers are not kept indefinitely.

For the user, the opt-out procedure can be summarized as follow:

1. Open the Wi-Fi network manager;
2. Identify and select the opt-out network;
3. Connect to the opt-out network.

From a user point of view, this opt-out mechanism involves a small number of simple tasks with which most users are familiar: identifying and connecting to a Wi-Fi network. Thus, we provide a user experience which won't discourage users from actually opting out of the system.

Wi-Fi-based opt-out is supported by all Wi-Fi devices providing a user interface, which is the case for the majority of devices carried around by people (smartphones, tablets, laptops, etc.). Moreover, it has the big advantage of not requiring any software or hardware modification.

On the Wi-Fi tracking system side, only minor modifications must be performed: a prim-

1. This AP is primitive because it does not provide any service other than announcing its presence and allowing devices association. In particular, it does not provide IP connectivity, i.e. no network connection is possible. Because of this lack of network connectivity, most devices will disconnect automatically after a certain period of time.

itive AP must be deployed and must be linked to the Wi-Fi tracking system in order to report opting out MAC addresses.

Another advantage of this method is its persistence and its seamless nature: each time the device will detect an opt-out AP using the same SSID, it will automatically notify its willingness to opt out without requiring any user intervention. Indeed, as the device has already been successfully connected to the opt-out network, the latter is kept in its PNL. As a consequence, next time the device will come in range of an AP advertising this opt-out SSID, it will associate with this AP, effectively indicating its intent to opt out.

A global opt-out mechanism for Wi-Fi tracking could be implemented if all stakeholders agree on a common SSID. This mechanism could then be seen as an equivalent of the web-based *Do-Not-Track* mechanism [147] for the physical world.

A limitation of this approach is that it could help an attacker to perform a wide-scale karma attack [39]. To prevent this, OSes could be instructed not to actually associate with networks having a "Do Not Track" SSID.

VII.3 Application: raising user awareness

Users are generally not aware that Wi-Fi tracking is possible and are even less aware that it is actually used by commercial entities. This can be explained by the fact that tracking is performed using radio signals, a technology that leaves no visible traces. Moreover, the visual notifications displayed by trackers are generally obscure or hidden.

For the sake of transparency, it is therefore important to show people that such technologies exist, and to explain their principles and their capabilities. Wombat has been developed in this spirit: to raise user awareness by demonstrating a real-world Wi-Fi tracking system.

Although Wombat lacks some of the functionalities found in industrial Wi-Fi tracking systems, it features their core functionalities: device identification, detection and itinerary tracking. These functionalities are sufficient to present the principle of a Wi-Fi tracking system and to initiate a discussion on the corresponding privacy issues. The Wombat system has been used during demonstrations addressed to different types of audiences: researchers, students, industrials, and the general public (see section VIII.4.3).

The demonstration scenario in the *Cité des Sciences* is the following. Visitors exploring the exhibition are tracked through Wi-Fi signals emitted by their personal devices. At the entrance, they are notified of the system's presence and of the opt-out mechanism. In the last part of the visit, they are presented a user interface on which they can explore the information that has been collected on them. Through a proximity sensor combined with a Wi-Fi interface, the system detects the device that is placed on the stand. From there, an interactive screen displays the collected data: identifier, brand of the device, name of the networks searched by the device, and an approximate representation of the user itinerary inside the exhibition.

This last demonstration has the potential to enlighten a large number of individuals from the general public. People aware of the potential privacy issues will be more inclined to adopt solutions to protect their privacy, and to ask for better privacy protections, either legal or technical.

Deployments of the Wombat system are listed in section VIII.4.3.

VII.4 Conclusion

We introduced Wombat, an experimental Wi-Fi tracking system. We showed how it can be used as a demonstration tool in order to raise user awareness. Then, we discussed how this platform can be used as a basis to develop and test privacy-preserving mechanisms. The first one of them, a Wi-Fi-based opt-out mechanism, has been presented. It has the advantage of being easy to implement and to use be end users.

We envision developing the demonstrative aspects of Wombat, by including other radio technologies, improving the trajectory reconstruction algorithm, and extending the user interface. We also plan to integrate other privacy-preserving features, such as privacy-preserving analytics [5].

Chapter VIII

Conclusion

VIII.1 Summary

In this thesis, we studied Wi-Fi-enabled mobile devices fingerprinting. We notably followed the evolution of a technique currently under massive deployment, called MAC address randomization. We have studied the deployment of this technique, the implementation flaws and limitations in several devices, and presented several classes of fingerprinting methods to bypass it. The first one leverages content of probe request frames sent by mobile devices, including their Information Elements and SSIDs. One of these Information Elements, the WPS UUID, appears to uniquely identify devices, and can be used to recover the device's original MAC address despite randomization. The second one uses their timing, including burst duration, number of frames and the distribution of Inter-Frame Arrival Time. We also presented a tracking system we developed for public awareness raising and experiments.

Our results show that there still is a long way to go before MAC address randomization is correctly implemented in consumer devices. Such an implementation is difficult to reach, because cooperation between components manufactured by different actors is needed. Besides, implementing a solid PRNG generating randomized addresses according to a uniform distribution, although needed, appears not to be reached in some implementations.

VIII.2 Perspectives

The perspectives opened by this thesis are the following. The discussion on real-world Wi-Fi tracking systems gives pointers to regulation entities and concerned citizens regarding the potential and actual abuse of such technologies. The study of the implementation of MAC address randomization, along with the two presented attacks, gives developers some guidelines on the different aspects of such implementations that have to be considered to provide correct privacy protection against tracking. This opportunity was already seized by Google [89]. Advanced attacks leveraging timing of probe requests can be developed, for instance including inter-bursts timings. Moreover, the counting capabilities of some clustering algorithms could be further studied as a potential way to estimate a crowd size despite the spread of MAC address randomization as a privacy-preserving measure. Finally, the developed tracking system is now a fully-functional platform, on which privacy-protecting techniques will be tested, and which will certainly be reused for demonstrations in the future.

VIII.3 Guidelines for MAC address randomization

Considering flaws presented in chapters IV, V and VI, we propose the following guidelines for a correct implementation of MAC address randomization:

1. The MAC address must be changed in every burst of probe requests.
2. Probe requests must be devoid of unnecessary Information Elements.
3. In particular, SSIDs must always be null (see below for the hidden AP case).
4. Sequence numbers must be randomized at the beginning of every burst of probe request, or maintain a fixed value (0).
5. The function generating the random addresses must be of cryptographic level so as to guarantee unlinkability between frames, as its purpose is to protect privacy-sensitive information. It must have a reliable entropy source, as MAC address changes can be very frequent.
6. It must be ensured that the global address is never used for active scanning.
7. The OUI part of the MAC address should preferably be randomized as well. To do so, randomized OUI should have their LA bit set to 1 and should not be registered by a company.

8. Efforts must be done to break timing patterns of probe requests: for instance, random delay can be added between bursts.

Additionally, we advise dropping the support of hidden Access Points. They hinder a correct implementation of randomization, as devices need to add identifying SSIDs to their probe requests to discover them. They constitute a bad security practice anyway, as they're based on the broken security-by-obscurity concept. They should be replaced with Access Points using a strong encryption scheme and strong passwords.

VIII.4 Summary of contributions

VIII.4.1 List of publications

Peer-reviewed conferences

1. Célestin Matte, Mathieu Cunche, Franck Rousseau, and Mathy Vanhoef. Defeating MAC address randomization through timing attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, pages 15–20. ACM, 2016
2. Célestin Matte and Mathieu Cunche. Demo: Panoptiphone: How unique is your Wi-Fi device? In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, pages 209–211. ACM, 2016
3. Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo Cardoso, and Frank Piessens. Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In *AsiaCCS*, May 2016
4. Célestin Matte, Jagdish Prasad Achara, and Mathieu Cunche. Device-to-identity linking attack using targeted Wi-Fi geolocation spoofing. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, page 20. ACM, 2015

Articles in general-public technical journals (see section VIII.4.3)

1. Célestin Matte. Transfert de style : et si Van Gogh peignait Tux ? *GNU/Linux Magazine France*, 202, March 2017

2. Célestin Matte and Mathieu Cunche. Traçage Wi-Fi : applications et contre-mesures. *GNU/Linux Magazine France*, HS 84, May 2016
3. Célestin Matte. Fingerprinting de smartphones : votre téléphone est-il traçable ? *MISC - Multi-Systems & Internet Security Cookbook*, 81, September 2015

Additional presentations

1. Célestin Matte and Mathieu Cunche. Wombat: An experimental Wi-Fi tracking system. In *Atelier sur la Protection de la Vie Privée*, 2017
2. Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo Cardoso, and Frank Piessens. Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms. In *Atelier sur la Protection de la Vie Privée*, 2016
3. Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo Cardoso, and Frank Piessens. Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms. In *CITI Lab. PhD Day*, 2016
4. Célestin Matte and Mathieu Cunche. Beam me up, Scotty: identifying the individual behind a MAC address using Wi-Fi geolocation spoofing. In *1er Colloque sur la Confiance Numérique en Auvergne*, 2014
5. Célestin Matte and Mathieu Cunche. Beam me up, Scotty: identifying the individual behind a MAC address using Wi-Fi geolocation spoofing. In *Atelier sur la Protection de la Vie Privée*, 2014

Other publications

1. Célestin Matte, Mathieu Cunche, and Vincent Toubiana. Does disabling Wi-Fi prevent my Android phone from sending Wi-Fi frames? Research Report RR-9089, Inria - Research Centre Grenoble – Rhône-Alpes; INSA Lyon, August 2017
2. Célestin Matte, Marine Minier, Mathieu Cunche, and Franck Rousseau. Poster: Privacy-preserving Wi-Fi tracking systems. In *CITI Lab. PhD Day*, 2015

Software production

1. Wombat, an experimental tracking system (not public yet).
2. Panoptiphone [146], a tool calculating a device's uniqueness, on the model of Panopticlick [50].

Contributions to standardization bodies

1. Mathieu Cunche, Juan Carlos Zuniga, Mathy Vanhoef, and Célestin Matte. Privacy issues in 802.11 networks. `https://mentor.ieee.org/802.11/dcn/16/11-16-1492-00-0wng-privacy-issues-in-802-11-networks.pptx`, 2016
2. Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo Cardoso, and Frank Piessens. Tracking 802.11 stations without relying on the link layer identifier. `https://mentor.ieee.org/privecsg/dcn/16/privecsg-16-0003-00-0000-tracking-802-11-stations-without-relying-on-the-link-layer-identifier.pdf`, 2016

VIII.4.2 Impact on the industry

We were contacted by Google regarding our articles on bypassing of MAC address randomization. The company explicitly cited our collaboration for the improvement of randomization in Android O [89]. Modifications related to our work are the following [1]:

— For each Wi-Fi scan while it is disconnected from an access point, the phone uses a new random MAC address (whether or not the device is in standby).
— The initial packet sequence number for each scan is also randomized.
— Unnecessary Probe Request Information Elements have been removed: Information Elements are limited to the SSID and DS parameter sets.

We contacted chipset vendors regarding bugs described in chapter IV. Broadcom acknowledged working on fixing both these issues and those described in our article [200]. Qualcomm confirmed the random address reuse vulnerability and rated the issue as high (issue QPSIIR-522), and files the other bugs as issues QPSIIR-894 and QPSIIR-895. We also reported the address reuse bug to our contact at Google for the record.

Results of our AsiaCCS paper were presented [200] at several sessions of the IEEE 802.11 privacy group, on both a teleconference [199] and on a live session in San Antonio [38], to propose the inclusion of privacy recommendations in 802.11 specifications. To be precise, this entity does not have any legal power, but its regulations recommendations are de facto globally followed by the industry.

All of this shows the impact of our work on influential real-world actors. The Google privacy team was quite reactive and contacted us itself. For all other actors to implement our proposed countermeasures correctly, some standardization effort is necessary.

1. Copied verbatim from the blog post.

VIII.4.3 Popularization

One of the roles of scholars is to broadcast the results of their research to the wider public, and to bring their advanced technical point of views to public debates. This is especially true in our domain, where research is applied by essence. In the privacy domain, it is essential for both technical and legal actors to work together, their complementarity being the key to insure good privacy guarantees to the citizens of society. Moreover, legislators lack resources to hire technical specialists for every new technology. It is our duty as researchers to work with them and help them understand implications of technical issues and legal decisions. Similarly, the general public does not have all the keys to understand problems risen with new technologies. Bringing light on these issues is sometimes a necessary step in fixing them, as political deciders may not be aware of these problems, or not be interested in investing time and money in solving issues no one knows about.

Thus, part of the work carried out during the PhD was made with the aim of popularization. Different groups of public where targeted: wide public, technically-advanced people, other scientists, companies, regulation entities.

Reaching the public is one of the aims of the Wombat tool, described in section VII. As of May 2017, several of its instances are permanently deployed:

1. at the *Cité des Sciences et de l'Industrie* in Paris,
2. in a showroom of the laboratory.

La Cité des Sciences et de l'Industrie is a museum dedicated to science popularization in France welcoming more than 3 millions of visitors each year. The Wombat installation is part of a one-year-long exhibition (April 2017 - March 2018) on data and digital technologies called *Terra Data*[1]. The latter discusses technical and societal aspects of big data. More precisely, it explains basic concepts such as algorithms, data, personal information, etc. Wombat is deployed all over the exhibition using 9 sensor nodes, a server node and an opt-out node. It is accompanied by a user interface developed by a third party. It tracks visitors of the exhibition, so as to present them information of their concern: estimated covered path inside the exhibition, information about their device, previously used Wi-Fi networks, etc. We provide an opt-out system for users unwilling to be tracked. This installation aims to give a real-life example of privacy issues related to Wi-Fi tracking, along with introducing the opt-out mechanism. We believe an example has more impact to the public if people are struck by the display of their own personal

1. http://www.cite-sciences.fr/fr/vous-etes/enseignants/votre-visite/expositions/terra-data/

data. This exhibition has been relayed in the French press [1].

To reach companies, we performed several demos using the Wombat project along with other demos unrelated to this PhD:

1. during several demos aimed at the industry during the Citi-SPIE chair opening ceremony and as a consequence of the latter,

2. at the *Salon Internet des Objets 2017* (Internet of Things Exhibition 2017) [2],

Reaching companies is important for our research to have an impact on the industry. For ethical reasons, we ignored a request from a company which contacted us to demand us to help them setting tracking systems up.

We also published articles is non-scientific technical journals, targeted at technically-advanced readers. The first one presents the different possible methods to track devices [135], as presented in section II.3. The second one presents practical techniques to track devices, and to prevent oneself against tracking [141] [3]. The point of these articles is to render our research topics (state of the art and technical details) more reachable for a non-scientific public. The `panoptiphone` demo is targeted at more or less the same public, as it displays advanced technical information.

Demos were also made for an entity having some kind of legal power: the CNIL organization. The latter is commissioned by the French government to fix rules regarding new technologies, and especially databases containing personal information. It produced a list of rules related to Wi-Fi tracking [31] and has the power to fine companies which do not respect them. We discussed technical aspects regarding Wi-Fi tracking (MAC address randomization, bypass methods, opt-out strategies). To this day, we remain in contact with them.

Popularization about Wi-Fi tracking issues has thus been made to all kind of public.

1. Among others: `https://www.franceinter.fr/emissions/grand-angle/grand-angle-05-avril-2017`, `http://www.francetvinfo.fr/partenariats/exposition-terra-data-nos-vies-a-lere-numerique-a-la-cite-des-sciences-et-de-lindustrie_2134591.html`, `http://www.lemonde.fr/big-browser/article/2017/04/20/l-expo-terra-data-decode-pour-vous-le-monde-des-donnees_5114357_4832693.html`, `http://www.leparisien.fr/flash-actualite-culture/voyage-dans-le-monde-fascinant-et-effrayant-des-donnees-04-04-2017-6823692.php`, all consulted on 2017.09.08

2. `http://www.sido-event.com/`, consulted on 2017.09.22

3. As a side note, I also published an article about a topic unrelated to my thesis: style transfer using neural networks [136].

VIII.4.4 Our work in the press

Some of our works have been mentioned in general-public press articles.

Our AsiaCCS article [200] was seldom mentioned in the press [1], along with our IEEE presentation [2].

Our recent technical report related to the trackability of devices whose Wi-Fi interface is deactivated [144] has been relayed a lot in the press. It's been relayed by different famous French new technologies news websites [3], a science news website [4], and a consumer protection association [5]. On the international side, a Hacker News post gathered over a hundred comments [6].

VIII.5 Concluding remarks

The battle for end-user privacy should not be fought on the sole technical side. Regulations entities can have a strong role and impact. Clear and strong rules must be defined and enforced regarding the various tracking technologies. The current deployment of many Wi-Fi-based analytics systems is slowed down by entities such as the CNIL. It is our duty as privacy researchers to work along with them in order to provide them valuable technological inputs regarding different aspects of these systems.

Privacy researchers also have a role towards the general public, usually ill-prepared to defend themselves against abusive use of technologies. Along with the technical and regulation side, researchers have to help the general public understand the implications of new technologies, so as to allow public debate not to be biased towards a blind acceptance of new technologies due to advanced marketing strategies or other kinds of propaganda.

1. https://securityintelligence.com/news/mac-address-randomization-gets-clobbered/, https://www.theregister.co.uk/2017/03/10/mac_address_randomization/, consulted on 2017.09.08
2. https://www.bleepingcomputer.com/news/security/researchers-break-mac-address-randomization-and-track-100-percent-of-test-devices/, consulted on 2017.09.08
3. Among others: http://www.01net.com/actualites/sur-android-le-wi-fi-peut-vous-tracer-meme-s-il-est-desactive-1245292.html, http://www.zdnet.fr/actualites/android-desactiver-le-wi-fi-n-empeche-pas-d-etre-espionne-39856640.htm, http://www.commentcamarche.net/news/5870290-meme-desactive-le-wi-fi-reste-tracable, consulted on 2017.09.08
4. https://www.sciencesetavenir.fr/high-tech/meme-coupe-le-wi-fi-sous-android-peut-suivre-le-telephone_116061, consulted on 2017.09.08
5. https://www.quechoisir.org/actualite-smartphones-android-meme-une-fois-le-wi-fi-desactive-vous-etes-piste-n46076/, consulted on 2017.09.08
6. https://news.ycombinator.com/item?id=15141077&goto=news, consulted on 2017.09.08

Bibliography

[1] Naeim Abedi, Ashish Bhaskar, and Edward Chung. Bluetooth and Wi-Fi MAC address based crowd data collection and monitoring: benefits, challenges and enhancement. In *Australasian Transport Research Forum (ATRF), 36th, 2013, Brisbane, Queensland, Australia*, 2013.

[2] Jagdish Prasad Achara, Gergely Acs, and Claude Castelluccia. On the unicity of smartphone applications. In *Proceedings of the 14th ACM Workshop on Privacy in the Electronic Society*, pages 27–36. ACM, 2015.

[3] V Acuna, Abhaykumar Kumbhar, Edwin Vattapparamban, F Rajabli, and I Guvenc. Localization of wifi devices using probe requests captured at unmanned aerial vehicles. In *Wireless Communications and Networking Conference (WCNC), 2017 IEEE*, pages 1–6. IEEE, 2017.

[4] Berker Agır, Kévin Huguenin, Urs Hengartner, and Jean-Pierre Hubaux. On the privacy implications of location semantics. *Proceedings on Privacy Enhancing Technologies*, 2015.

[5] Mohammad Alaggan, Mathieu Cunche, and Marine Minier. Privacy-preserving t-incidence for wifi-based mobility analytics. In *7e Atelier sur la Protection de la Vie Privée (APVP'16)*, 2016.

[6] Wahhab Albazrqaoe, Jun Huang, and Guoliang Xing. Practical bluetooth traffic sniffing: Systems and privacy implications. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services*, pages 333–345. ACM, 2016.

[7] Thomas Allmer. A critical contribution to theoretical foundations of privacy studies. *Journal of Information, Communication and Ethics in Society*, 9(2):83–101, 2011.

[8] Monica Anderson. Technology device ownership, 2015. http://www.pewinternet.org/2015/10/29/technology-device-ownership-2015/, consulted on 2017.05.15, 2015.

[9] ARCEP. L'État d'internet en france 2017, 2017.

[10] UN General Assembly. Universal declaration of human rights. *UN General Assembly*, 1948.

[11] Tuomas Aura, Janne Lindqvist, Michael Roe, and Anish Mohammed. Chattering laptops. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 167–186. Springer, 2008.

[12] Michael Backes, Sven Bugiel, Erik Derr, Sebastian Gerling, and Christian Hammer. R-droid: Leveraging android app analysis with static slice optimization. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 129–140. ACM, 2016.

[13] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. Ieee, 2000.

[14] M Barbeau, J Hall, and E Kranakis. Detection of rogue devices in bluetooth networks using radio frequency fingerprinting. In *proceedings of the 3rd IASTED International Conference on Communications and Computer Networks, CCN*, pages 4–6, 2006.

[15] Marco V. Barbera, Alessandro Epasto, Alessandro Mei, Sokol Kosta, Vasile C. Perta, and Julinda Stefa. CRAWDAD dataset sapienza/probe-requests (v. 2013-09-10). Retrieved 10 November, 2015, from, http://crawdad.org/sapienza/probe-requests/20130910, September 2013.

[16] Marco V Barbera, Alessandro Epasto, Alessandro Mei, Vasile C Perta, and Julinda Stefa. Signals from the crowd: uncovering social relationships through smartphone probes. In *Proceedings of the 2013 conference on Internet measurement conference*, pages 265–276. ACM, 2013.

[17] Bettina Berendt, Oliver Günther, and Sarah Spiekermann. Privacy in e-commerce: stated preferences vs. actual behavior. *Communications of the ACM*, 48(4):101–106, 2005.

[18] Carlos Bernardos, Juan Carlos Zúñiga, and Piers O'Hanlon. Wi-Fi internet connectivity and privacy: hiding your tracks on the wireless internet. In *IEEE CSCN*, 2015.

[19] B. Bloessl, C. Sommer, F. Dressler, and D. Eckhoff. The scrambler attack: A robust physical layer attack on location privacy in vehicular networks. In *ICNC*, 2015.

[20] Sergey Bratus, Cory Cornelius, David Kotz, and Daniel Peebles. Active behavioral fingerprinting of wireless devices. In *Proceedings of the first ACM conference on Wireless network security*, pages 56–61. ACM, 2008.

[21] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[22] Michael Brennan, Sadia Afroz, and Rachel Greenstadt. Adversarial stylometry: Circumventing authorship recognition to preserve privacy and anonymity. *ACM Transactions on Information and System Security (TISSEC)*, 15(3):12, 2012.

[23] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. Wireless device identification with radiometric signatures. In *MobiCom*, 2008.

[24] Grant Bugher. Detecting bluetooth surveillance systems. In *DEFCON*, 2014.

[25] Johnny Cache. Fingerprinting 802.11 implementations via statistical analysis of the duration field. *Uninformed. org*, 5, 2006.

[26] CB Insights. The store of the future: 150+ startups transforming brick-and-mortar retail in one infographic, 2017.

[27] Cesar Cerrudo. Hacking us traffic control systems. In *DEFCON*, 2014.

[28] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.

[29] Yohan Chon, Suyeon Kim, Seungwoo Lee, Dongwon Kim, Yungeun Kim, and Hojung Cha. Sensing wifi packets in the air: practicality and implications in urban mobility monitoring. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 189–200. ACM, 2014.

[30] Stephanie Clifford and Quentin Hardy. Attention, Shoppers: Store Is Tracking Your Cell. `http://www.nytimes.com/2013/07/15/business/attention-shopper-stores-are-tracking-your-cell.html`, 2013.

[31] CNIL. Mesure de fréquentation et analyse du comportement des consommateurs dans les magasins. http://www.cnil.fr/linstitution/actualite/article/article/mesure-de-frequentation-et-analyse-du-comportement-des-consommateurs-dans-les-magasins/, consulted on 2017.05.05, 2014.

[32] Cherita Corbett, Raheem Beyah, and John Copeland. A passive approach to wireless nic identification. In *Communications, 2006. ICC'06. IEEE International Conference on*, volume 5, pages 2329–2334. IEEE, 2006.

[33] Cherita L Corbett, Raheem A Beyah, and John A Copeland. Passive classification of wireless nics during rate switching. *EURASIP Journal on Wireless Communications and Networking*, 2008(1):495070, 2007.

[34] Cherita L Corbett, Raheem A Beyah, and John A Copeland. Passive classification of wireless nics during active scanning. *International Journal of Information Security*, 7(5):335–348, 2008.

[35] Marius Cristea and Bogdan Groza. Fingerprinting smartphones remotely via icmp timestamps. *IEEE Communications Letters*, 17(6):1081–1083, 2013.

[36] Mathieu Cunche. I know your MAC address: Targeted tracking of individual using Wi-Fi. *Journal of Computer Virology and Hacking Techniques*, pages 1–9, 2013.

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

[38] Mathieu Cunche, Juan Carlos Zuniga, Mathy Vanhoef, and Célestin Matte. Privacy issues in 802.11 networks. https://mentor.ieee.org/802.11/dcn/16/11-16-1492-00-0wng-privacy-issues-in-802-11-networks.pptx, 2016.

[39] Dino A Dai Zovi and Shane A Macaulay. Attacking automatic wireless network selection. In *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, pages 365–372. IEEE, 2005.

[40] Boris Danev, Davide Zanetti, and Srdjan Capkun. On physical-layer identification of wireless devices. *ACM Computing Surveys (CSUR)*, 45(1):6, 2012.

[41] Davide Dardari, Pau Closas, and Petar M Djurić. Indoor tracking: Theory, methods, and technologies. *IEEE Transactions on Vehicular Technology*, 64(4):1263–1278, 2015.

[42] Anupam Das, Nikita Borisov, and Matthew Caesar. Do you hear what i hear?: Fingerprinting smart devices through embedded acoustic components. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 441–452. ACM, 2014.

[43] Datatilsynet. Tracking in public spaces, 2016.

[44] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376, 2013.

[45] Deloitte. Usages mobiles. `https://www2.deloitte.com/fr/fr/pages/technology-media-and-telecommunications/articles/usages-mobiles-2016.html`, 2016.

[46] Levent Demir, Mathieu Cunche, and Cédric Lauradoux. Analysing the privacy policies of Wi-Fi trackers. In *Proc. of the 2014 workshop on physical analytics*, 2014.

[47] Saandeep Depatla, Arjun Muralidharan, and Yasamin Mostofi. Occupancy estimation using only wifi power measurements. *IEEE Journal on Selected Areas in Communications*, 33(7):1381–1393, 2015.

[48] Loh Chin Choong Desmond, Cho Chia Yuan, Tan Chung Pheng, and Ri Seng Lee. Identifying unique devices through wireless fingerprinting. In *Proceedings of the first ACM conference on Wireless network security*, pages 46–55. ACM, 2008.

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

[50] Peter Eckersley. How unique is your web browser? In *Privacy Enhancing Technologies*, 2010.

[51] Jonathan P Ellch. Fingerprinting 802.11 devices. Technical report, NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 2006.

[52] Karim Emara, Wolfgang Woerndl, and Johann Schlichter. CAPS: context-aware privacy scheme for vanet safety applications. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, page 21. ACM, 2015.

[53] Tobias Engel. SS7: Locate. track. manipulate. In *Chaos Communication Congress*, 2014.

[54] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[55] Daniel B Faria and David R Cheriton. Detecting identity-based attacks in wireless networks using signalprints. In *Proceedings of the 5th ACM workshop on Wireless security*, pages 43–52. ACM, 2006.

[56] Zack Fasel and Erin Jacobs. I fight for the users, episode i - attacks against top consumer products. In *DEFCON*, 2016.

[57] Kassem Fawaz, Kyu-Han Kim, and Kang G Shin. Privacy vs. reward in indoor location-based services. *Proceedings on Privacy Enhancing Technologies*, 2016(4):102–122, 2016.

[58] Federal Trade Commission. Complaint. `https://www.ftc.gov/system/files/documents/cases/150423nomicmpt.pdf`, consulted on 2017.07.14, 2015.

[59] David Förster, Frank Kargl, and Hans Löhr. A framework for evaluating pseudonym strategies in vehicular ad-hoc networks. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, page 19. ACM, 2015.

[60] Michel Foucault. *Surveiller et punir. Naissance de la prison.* Editions Gallimard, 2014.

[61] Atec ITS France, editor. *Evaluations simultanées de différentes technologies innovantes de recueil de données trafic pour le calcul de temps de parcours en temps réel*, 2015.

[62] Jason Franklin, Damon McCoy, Parisa Tabriz, Vicentiu Neagoe, Jamie V Randwyk, and Douglas Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *USENIX Security*, 2006.

[63] Julien Freudiger. How talkative is your mobile device? An experimental study of Wi-Fi probe requests. In *WiSec*, 2015.

[64] Julien Freudiger, Maxim Raya, Márk Félegyházi, Panos Papadimitratos, and Jean-Pierre Hubaux. Mix-zones for location privacy in vehicular networks. In *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, 2007.

[65] Christian Fuchs. Towards an alternative concept of privacy. *Journal of Information, Communication and Ethics in Society*, 9(4):220–237, 2011.

[66] Yuki Fukuzaki, Masahiro Mochizuki, Kazuya Murao, and Nobuhiko Nishio. A pedestrian flow analysis system using Wi-Fi packet sensors to a real environment. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 721–730. ACM, 2014.

[67] Brian Fung. Wi-Fi tracking in retail stores. `http://www.washingtonpost.com/blogs/the-switch/wp/2013/10/19/how-stores-use-your-phones-wifi-to-track-your-shopping-habits/`, consulted on 2014.04.01, 2013.

[68] Future of Privacy Forum. Opt out of smart store tracking. `https://optout.smart-places.org/`, consulted on 2015.04.02, 2014.

[69] Sean Gallagher. Hands-on: Blue hydra can expose the all-too-unhidden world of bluetooth. `https://arstechnica.com/information-technology/2016/09/hands-on-blue-hydra-can-expose-the-all-too-unhidden-world-of-bluetooth/`, consulted on 2017., 2016.

[70] Ke Gao, Cherita Corbett, and Raheem Beyah. A passive approach to wireless device fingerprinting. In *Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on*, pages 383–392. IEEE, 2010.

[71] Barton Gellman and Ashkan Soltani. NSA tracking cellphone locations worldwide, Snowden documents show. *The Washington Post*, 2013.

[72] Denton Gentry and Avery Pennarun. Passive taxonomy of wifi clients using mlme frame contents. *arXiv preprint arXiv:1608.01725*, 2016.

[73] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010.

[74] Megan Geuss. Creepy but legal phone-tracking company gets wrist slap for empty privacy promise. `https://arstechnica.com/tech-policy/2015/04/creepy-but-legal-phone-tracking-company-gets-wrist-slap-for-empty-privacy-promise/`, consulted on 2017.06.08, 2015.

[75] F. Gont. A method for generating semantically opaque interface identifiers with ipv6 stateless address autoconfiguration (slaac). RFC 7217, IETF, 2014.

[76] KN Gopinath, Pravin Bhagwat, and K Gopinath. An empirical analysis of heterogeneity in ieee 802.11 MAC protocol implementations and its implications. In *Proceedings of the 1st international workshop on wireless network testbeds, experimental evaluation & characterization*, pages 80–87. ACM, 2006.

[77] Ben Greenstein, Ramakrishna Gummadi, Jeffrey Pang, Mike Y Chen, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. Can ferris bueller still have his day off? protecting privacy in the wireless era. In *HotOS*, 2007.

[78] Ben Greenstein, Damon McCoy, Jeffrey Pang, Tadayoshi Kohno, Srinivasan Seshan, and David Wetherall. Improving wireless privacy with an identifier-free link layer protocol. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 40–53. ACM, 2008.

[79] Guillaume Grolleau. La captation bluetooth au service des aménagements urbains, 2015.

[80] Emmanuel Grumbach. iwlwifi: mvm: support random MAC address for scanning. Linux commit `effd05ac479b`, 2014.

[81] Marco Gruteser and Dirk Grunwald. Enhancing location privacy in wireless LAN through disposable interface identifiers: A quantitative analysis. *Mobile Networks and Applications*, 10(3):315–325, 2005.

[82] Fanglu Guo and Tzi-cker Chiueh. Sequence number-based MAC address spoof detection. In *Recent Advances in Intrusion Detection*, pages 309–329. Springer, 2006.

[83] Jeyanthi Hall, Michel Barbeau, and Evangelos Kranakis. Enhancing intrusion detection in wireless networks using radio frequency fingerprinting. In *Communications, Internet, and Information Technology*, pages 201–206, 2004.

[84] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review*, 41(1):53–53, 2011.

[85] Quentin Hardy. Technology turns to tracking people offline. `https://mobile.nytimes.com/blogs/bits/2013/03/07/technology-turns-to-tracking-people-offline/?referer`, 2013.

[86] Parker Higgins and Lee Tien. Mobile tracking code of conduct falls short of protecting consumers. `https://www.eff.org/deeplinks/2013/10/mobile-tracking-code-conduct-falls-short-protecting-consumers`, 2013.

[87] Andrew Hilts, Christopher Parsons, and Jeffrey Knockel. Every step you fake–a comparative analysis of fitness tracker privacy and security. *Open Effect Report*, 2016.

[88] Tin Kam Ho. Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE, 1995.

[89] Giles Hogben. Changes to device identifiers in android o. `https://android-developers.googleblog.com/2017/04/changes-to-device-identifiers-in.html`, consulted on 2017.06.09, 2017.

[90] Baik Hoh and Marco Gruteser. Protecting location privacy through path confusion. In *Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005. First International Conference on*, pages 194–205. IEEE, 2005.

[91] Ryan Holeman. The bluetooth device database. In *DEFCON*, 2013.

[92] Bunnie Huang and Edward Snowden. Against the law: Countering lawful abuses of digital surveillance, 2016.

[93] Jun Huang, Wahhab Albazrqaoe, and Guoliang Xing. Blueid: A practical system for bluetooth device identification. In *INFOCOM, 2014 Proceedings IEEE*, pages 2849–2857. IEEE, 2014.

[94] Leping Huang, Kanta Matsuura, Hiroshi Yamane, and Kaoru Sezaki. Enhancing wireless location privacy using silent period. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 2, pages 1187–1192. IEEE, 2005.

[95] Christian Huitema. Experience with MAC address randomization in Windows 10. In *93th Internet Engineering Task Force Meeting (IETF)*, July 2015.

[96] IEEE. Guidelines for use organizationally unique identifier (oui) and company id (cid). `http://standards.ieee.org/develop/regauth/tut/eui.pdf`, consulted on 2017.05.28, 2014.

[97] IEEE 802.11 Working Group and others. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Std 802.11-2012, 2012.

[98] Insee. Tableaux de l'économie française, 2017.

[99] International Telecommunication Union. Ict facts and figures. https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf, 2013.

[100] International Telecommunication Union. Ict facts and figures. https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf, 2015.

[101] Taher Issoufaly and Pierre Ugo Tournoux. Bleb: Bluetooth low energy botnet for large scale individual tracking. In *Next Generation Computing Applications (NextComp), 2017 1st International Conference on*, pages 115–120. IEEE, 2017.

[102] Van Jacobson, Robert Braden, and David Borman. Tcp extensions for high performance. RFC 1323, IETF, 1992.

[103] Sakshi Jain, Mobin Javed, and Vern Paxson. Towards mining latent client identifiers from network traffic. *Proceedings on Privacy Enhancing Technologies*, 2016(2):100–114, 2016.

[104] Shuja Jamil, Sohaib Khan, Anas Basalamah, and Ahmed Lbath. Classifying smartphone screen on/off state based on wifi probe patterns. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 301–304. ACM, 2016.

[105] Suman Jana and Sneha Kumar Kasera. On fast and accurate detection of unauthorized wireless access points using clock skews. In *MobiCom*, 2008.

[106] Tao Jiang, Helen J Wang, and Yih-Chun Hu. Preserving location privacy in wireless lans. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 246–257. ACM, 2007.

[107] Reginald Victor Jones. *Most secret war*. Penguin UK, 2009.

[108] Michael Kamp, Christine Kopp, Michael Mock, Mario Boley, and Michael May. Privacy-preserving mobility monitoring using sketches of stationary sensor readings. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 370–386. Springer, 2013.

[109] Cecilia Kang. China plans to track cellphone users, sparking human rights concerns. http://voices.washingtonpost.com/posttech/2011/03/china_said_it_may_begin.html, consulted on 2017., 2011.

[110] Dmitry Kaplan and David M Stanhope. Waveform collection for use in wireless telephone identification, December 7 1999. US Patent 5,999,806.

[111] Lawrence George Kersta. Voiceprint identification. *Nature*, 196(4861):1253–1257, 1962.

[112] Yu Seung Kim, Yuan Tian, Le T Nguyen, and Patrick Tague. LAPWiN: Location-aided probing for protecting user privacy in Wi-Fi networks. In *Communications and Network Security (CNS), 2014 IEEE Conference on*, pages 427–435. IEEE, 2014.

[113] Tadayoshi Kohno, Andre Broido, and Kimberly C Claffy. Remote physical device fingerprinting. *Dependable and Secure Computing, IEEE Transactions on*, 2(2):93–108, 2005.

[114] Valentin Fedorovich Kolchin, Boris Aleksandrovich Sevastyanov, and Vladimir Pavlovich Chistyakov. *Random allocations*. Winston, 1978.

[115] Constantine E Kontokosta and Nicholas Johnson. Urban phenology: Toward a real-time census of the city using Wi-Fi data. *Computers, Environment and Urban Systems*, 64:144–153, 2017.

[116] Henning Siitonen Kortvedt. Securing near field communication, 2009.

[117] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[118] Braden Kowitz and Lorrie Cranor. Peripheral privacy notifications for wireless networks. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*, pages 90–96. ACM, 2005.

[119] Krowdthink. They know where you are - an investigation into the contracts, policies and practices of mobile and Wi-Fi service providers in relation to location tracking, 2016.

[120] Amrit Kumar. *Security and Privacy of Hash-Based Software Applications*. PhD thesis, Université Grenoble Alpes, 2016.

[121] Vincent Labatut and Hocine Cherifi. Evaluation of performance measures for classifiers comparison. *arXiv preprint arXiv:1112.4133*, 2011.

[122] Marc Langheinrich. Privacy by design—principles of privacy-aware ubiquitous systems. In *International conference on Ubiquitous Computing*, pages 273–291. Springer, 2001.

[123] P. Leach, M. Mealling, and R. Salz. A universally unique identifier (UUID) URN namespace. RFC 4122, IETF, July 2005.

[124] Yan Li and Ting Zhu. Gait-based Wi-Fi signatures for privacy-preserving. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 571–582. ACM, 2016.

[125] Jianxiong Liao and Jianqing Li. Effectively changing pseudonyms for privacy protection in vanets. In *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on*, pages 648–652. IEEE, 2009.

[126] Roman Lim, Marco Zimmerling, and Lothar Thiele. Passive, privacy-preserving real-time counting of unmodified smartphones via zigbee interference. In *Distributed Computing in Sensor Systems (DCOSS), 2015 International Conference on*, pages 115–126. IEEE, 2015.

[127] Janne Lindqvist, Tuomas Aura, George Danezis, Teemu Koponen, Annu Myllyniemi, Jussi Mäki, and Michael Roe. Privacy-preserving 802.11 access-point discovery. In *WiSec*, 2009.

[128] Li Lu, Runzhe Wang, Jing Ding, Wubin Mao, Wei Chen, and Hongzi Zhu. Fastid: An undeceived router for real-time identification of wifi terminals. In *IFIP Networking Conference (IFIP Networking), 2015*, pages 1–9. IEEE, 2015.

[129] Andreas Luber, Marek Junghans, Sascha Bauer, and Jan Schulz. On measuring traffic with Wi-Fi and Bluetooth. In *18th ITS World Congress*, 2011.

[130] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[131] Jeremy Martin, Travis Mayberry, Collin Donahue, Lucas Foppe, Lamont Brown, Chadwick Riggins, Erik C Rye, and Dane Brown. A study of MAC address randomization in mobile devices and when it fails. *arXiv preprint arXiv:1703.02874*, 2017.

[132] Jeremy Martin, Danny Rhame, Robert Beverly, and John McEachen. Correlating gsm and 802.11 hardware identifiers. In *Military Communications Conference, MILCOM 2013-2013 IEEE*, pages 1398–1403. IEEE, 2013.

[133] Jeremy Martin, Erik Rye, and Robert Beverly. Decomposition of MAC address structure for granular device inference. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*, pages 78–88. ACM, 2016.

[134] Jennifer Martinez. Franken still unsatisfied with Euclid's privacy practices. `http://thehill.com/policy/technology/291299-franken-still-unsatisfied-with-euclids-privacy-practices`, consulted on 2017.05.17, 2013.

[135] Célestin Matte. Fingerprinting de smartphones : votre téléphone est-il traçable ? *MISC - Multi-Systems & Internet Security Cookbook*, 81, September 2015.

[136] Célestin Matte. Transfert de style : et si Van Gogh peignait Tux ? *GNU/Linux Magazine France*, 202, March 2017.

[137] Célestin Matte, Jagdish Prasad Achara, and Mathieu Cunche. Device-to-identity linking attack using targeted Wi-Fi geolocation spoofing. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, page 20. ACM, 2015.

[138] Célestin Matte and Mathieu Cunche. Beam me up, Scotty: identifying the individual behind a MAC address using Wi-Fi geolocation spoofing. In *1er Colloque sur la Confiance Numérique en Auvergne*, 2014.

[139] Célestin Matte and Mathieu Cunche. Beam me up, Scotty: identifying the individual behind a MAC address using Wi-Fi geolocation spoofing. In *Atelier sur la Protection de la Vie Privée*, 2014.

[140] Célestin Matte and Mathieu Cunche. Demo: Panoptiphone: How unique is your Wi-Fi device? In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, pages 209–211. ACM, 2016.

[141] Célestin Matte and Mathieu Cunche. Traçage Wi-Fi : applications et contre-mesures. *GNU/Linux Magazine France*, HS 84, May 2016.

[142] Célestin Matte and Mathieu Cunche. Wombat: An experimental Wi-Fi tracking system. In *Atelier sur la Protection de la Vie Privée*, 2017.

[143] Célestin Matte, Mathieu Cunche, Franck Rousseau, and Mathy Vanhoef. Defeating MAC address randomization through timing attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec)*, pages 15–20. ACM, 2016.

[144] Célestin Matte, Mathieu Cunche, and Vincent Toubiana. Does disabling Wi-Fi prevent my Android phone from sending Wi-Fi frames? Research Report RR-9089, Inria - Research Centre Grenoble – Rhône-Alpes; INSA Lyon, August 2017.

[145] Célestin Matte, Marine Minier, Mathieu Cunche, and Franck Rousseau. Poster: Privacy-preserving Wi-Fi tracking systems. In *CITI Lab. PhD Day*, 2015.

[146] Célestin Matte. Panoptiphone. `https://github.com/Perdu/panoptiphone`, consulted on 2017. doi: `https://doi.org/10.5281/zenodo.1044394`, 2017.

[147] Jonathan Mayer, Arvind Narayanan, and Sid Stamm. Do Not Track: A Universal Third-Party Web Tracking Opt Out. Internet-Draft draft-mayer-do-not-track-00, Internet Engineering Task Force, March 2011. Work in Progress.

[148] William Melicher, Mahmood Sharif, Joshua Tan, Lujo Bauer, Mihai Christodorescu, and Pedro Giovanni Leon. (do not) track me sometimes: Users' contextual preferences for web tracking. *Proceedings on Privacy Enhancing Technologies*, 2016(2):135–154, 2016.

[149] Mariyan Mirza, Paul Barford, Xiaojin Zhu, Suman Banerjee, and Michael Blodgett. Fingerprinting 802.11 rate adaption algorithms. In *INFOCOM, 2011 Proceedings IEEE*, pages 1161–1169. IEEE, 2011.

[150] Steven J Murdoch. Hot or not: Revealing hidden services by their clock skew. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 27–36. ACM, 2006.

[151] ABM Musa and Jakob Eriksson. Tracking unmodified smartphones using Wi-Fi monitors. In *Proceedings of the 10th ACM conference on embedded network sensor systems*, pages 281–294. ACM, 2012.

[152] Christoph Neumann, Olivier Heen, and Stéphane Onno. An empirical study of passive 802.11 device fingerprinting. In *Distributed Computing Systems Workshops (ICDCSW), 2012 32nd International Conference on*, pages 593–602. IEEE, 2012.

[153] New York City Department of Consumer Affairs. New york city mobile services study: Research brief, 2015.

[154] Lily Newman. An open-source toolkit to help patch cell networks' critical flaw. `https://www.wired.com/story/ss7-flaw-open-source-toolkit/`, consulted on 2017.07.27, 2017.

[155] Le Nguyen, Yu Seung Kim, Patrick Tague, and Joy Zhang. Identitylink: User-device linking through visual and rf-signal cues. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, Sept 2014.

[156] James O Malley. Here's what tfl learned from tracking your phone on the tube. `http://www.gizmodo.co.uk/2017/02/heres-what-tfl-learned-from-tracking-your-phone-on-the-tube/`, consulted on 2017., 2017.

[157] Jonathan A Obar and Anne Oeldorf-Hirsch. The biggest lie on the internet: Ignoring the privacy policies and terms of service policies of social networking services. *The 44th Research Conference on Communication, Information and Internet Policy 2016*, 2016.

[158] OpinionLab. New study: consumers overwhelmingly reject in-store tracking by retailers. `https://www.opinionlab.com/newsmedia/new-study-consumers-overwhelmingly-reject-in-store-tracking-by-retailers/`, consulted on 2017.06.08, 2014.

[159] Brendan O'Connor. CreepyDOL: Cheap, distributed stalking. In *BlackHat*, 2013.

[160] Piers O'Hanlon, Ravishankar Borgaonkar, and Lucca Hirschi. Mobile subscriber wifi privacy. *arXiv preprint arXiv:1703.02874*, 2017.

[161] Jeffrey Pang, Ben Greenstein, Ramakrishna Gummadi, Srinivasan Seshan, and David Wetherall. 802.11 user fingerprinting. In *MobiCom*, 2007.

[162] Jeffrey Pang, Ben Greenstein, Srinivasan Seshan, and David Wetherall. Tryst: The case for confidential service discovery. In *HotNets*, 2007.

[163] Ruoming Pang, Mark Allman, Vern Paxson, and Jason Lee. The devil and packet trace anonymization. *ACM SIGCOMM Computer Communication Review*, 36(1):29–38, 2006.

[164] Chirag Patel, Dipti Shah, and Atul Patel. Automatic number plate recognition system (anpr): A survey. *International Journal of Computer Applications*, 69(9), 2013.

[165] Neal Patwari and Sneha K Kasera. Robust location distinction using temporal link signatures. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 111–122. ACM, 2007.

[166] Pew Research Center. Mobile fact sheet. `http://www.pewinternet.org/fact-sheet/mobile/`, consulted on 2017.05.15, 2017.

[167] Adam C Polak, Sepideh Dolatshahi, and Dennis L Goeckel. Identifying wireless users via transmitter imperfections. *IEEE Journal on Selected Areas in Communications*, 29(7):1469–1479, 2011.

[168] Jon Postel et al. Rfc 792: Internet control message protocol. RFC 792, IETF, 1981.

[169] Punkingmonkey. The road less surreptitiously traveled. In *DEFCON*, 2013.

[170] Kristin Purcell, Joanna Brenner, and Lee Rainie. Search engine use 2012. *Pew Internet & American Life Project Washington*, 2012.

[171] Lee Rainie and Kathryn Zickhur. Americans' views on mobile etiquette. `http://www.pewinternet.org/2015/08/26/chapter-1-always-on-connectivity/`, 2015.

[172] Alessandro Enrico Cesare Redondi, Davide Sanvito, and Matteo Cesana. Passive classification of Wi-Fi enabled devices. In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 51–58. ACM, 2016.

[173] Rainey Reitman. China deputizes smart phones to spy on beijing residents' real-time location. , consulted on 2017., 2011.

[174] Pieter Robyns, Bram Bonné, Peter Quax, and Wim Lamotte. Poster: Assessing the impact of 802.11 vulnerabilities using wicability. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 217–218. ACM, 2016.

[175] Pieter Robyns, Bram Bonné, Peter Quax, and Wim Lamotte. Noncooperative 802.11 MAC layer fingerprinting and tracking of mobile devices. *Security and Communication Networks*, 2017, 2017.

[176] Maya Rodrig, Charles Reis, Ratul Mahajan, David Wetherall, John Zahorjan, and Ed Lazowska. CRAWDAD dataset uw/sigcomm2004 (v. 2006-10-17). Downloaded from `http://crawdad.org/uw/sigcomm2004/20061017`, October 2006.

[177] Piotr Sapiezynski, Radu Gatej, Alan Mislove, and Sune Lehmann. Opportunities and challenges in crowdsourced wardriving. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference*, pages 267–273. ACM, 2015.

[178] T Scott Saponas, Jonathan Lester, Carl Hartung, Sameer Agarwal, and Tadayoshi Kohno. Devices that tell on you: Privacy trends in consumer ubiquitous computing. In *Usenix Security*, volume 3, page 3, 2007.

[179] Jeremy Scahill and Glenn Greenwald. The NSA's secret role in the U.S. assassination program. *The Intercept*, 2014.

[180] Lorenz Schauer, Martin Werner, and Philipp Marcus. Estimating crowd densities and pedestrian flows using Wi-Fi and bluetooth. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 171–177. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[181] Bruce Schneier. Csec surveillance analysis of ip and user data. `https://www.schneier.com/blog/archives/2014/02/csec_surveillan.html`, consulted on 2017.09.02, 2014.

[182] Aaron Schulman, Dave Levin, and Neil Spring. CRAWDAD dataset umd/sigcomm2008 (v. 2009-03-02). Downloaded from `http://crawdad.org/umd/sigcomm2008/20090302`, March 2009.

[183] H Seiwert. isniff gps: Passive sniffing tool for capturing and visualising wifi location data disclosed by ios devices. `https://github.com/hubert3/iSniff-GPS`, consulted on 2017.05.25, 2012.

[184] Sebastian Sevignani. The problem of privacy in capitalism and the alternative social networking site diaspora. *tripleC: Communication, Capitalism & Critique. Open Access Journal for a Global Sustainable Information Society*, 10(2):600–617, 2012.

[185] Kumar Sharad and George Danezis. An automated social graph de-anonymization technique. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pages 47–58. ACM, 2014.

[186] Elham Sharifi, Masoud Hamedi, Ali Haghani, and Hadi Sadrsadat. Analysis of vehicle detection rate for bluetooth traffic sensors: A case study in maryland and delaware. In *18th World Congress on on Intelligent Transport Systems*, 2011.

[187] Yong Sheng, Keren Tan, Guanling Chen, David Kotz, and Andrew Campbell. Detecting 802.11 MAC layer spoofing using received signal strength. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008.

[188] Sandra Siby, Rajib Ranjan Maiti, and Nils Tippenhauer. Iotscanner: Detecting and classifying privacy threats in iot neighborhoods. *arXiv preprint arXiv:1701.05007*, 2017.

[189] Katie Skinner and Jason Novak. Privacy and your app. In *Apple Worldwide Dev. Conf. (WWDC)*, June 2015.

[190] Ashkan Soltani. Technological overview. https://www.ftc.gov/system/files/documents/public_events/182251/mobiledevicetrackingseminar-slides.pdf, consulted on 2017., 2014.

[191] Ashkan Soltani. Privacy trade-offs in retail tracking. https://www.ftc.gov/news-events/blogs/techftc/2015/04/privacy-trade-offs-retail-tracking, consulted on 2017.05.17, 2015.

[192] Chad Spensky, Jeffrey Stewart, Arkady Yerukhimovich, Richard Shay, Ari Trachtenberg, Rick Housley, and Robert K Cunningham. Sok: Privacy on mobile devices–it's complicated. *Proceedings on Privacy Enhancing Technologies*, 2016(3):96–116, 2016.

[193] Statista. Share of adults in the united states who owned a smartphone from 2011 to 2013, by household income. https://www.statista.com/statistics/195006/percentage-of-us-smartphone-owners-by-household-income/, 2013.

[194] The Guardian. City of london corporation wants 'spy bins' ditched. https://www.theguardian.com/world/2013/aug/12/city-london-corporation-spy-bins, consulted on 2017.05.16, 2013.

[195] J Toonstra and W Kinsner. A radio transmitter fingerprinting system odo-1. In *Electrical and Computer Engineering, 1996. Canadian Conference on*, volume 1, pages 60–63. IEEE, 1996.

[196] Joseph Turow, Jennifer King, Chris Jay Hoofnagle, Amy Bleakley, and Michael Hennessy. Americans reject tailored advertising and three activities that enable it. *Departmental Papers (ASC)*, 2009.

[197] Oktay Ureten and Nur Serinken. Wireless security through rf fingerprinting. *Canadian Journal of Electrical and Computer Engineering*, 32(1):27–33, 2007.

[198] Niels Van Dijk. Property, privacy and personhood in a world of ambient intelligence. *Ethics and information technology*, 12(1):57–69, 2010.

[199] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo Cardoso, and Frank Piessens. Tracking 802.11 stations without relying on the link layer identifier. https://mentor.ieee.org/privecsg/dcn/16/privecsg-16-0003-00-0000-tracking-802-11-stations-without-relying-on-the-link-layer-identifier.pdf, 2016.

[200] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo Cardoso, and Frank Piessens. Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms. In *AsiaCCS*, May 2016.

[201] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo Cardoso, and Frank Piessens. Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms. In *Atelier sur la Protection de la Vie Privée*, 2016.

[202] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo Cardoso, and Frank Piessens. Why MAC address randomization is not enough: An analysis of Wi-Fi network discovery mechanisms. In *CITI Lab. PhD Day*, 2016.

[203] Tien Dang Vo-Huu, Triet Dang Vo-Huu, and Guevara Noubir. Fingerprinting Wi-Fi devices using software defined radios. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 3–14. ACM, 2016.

[204] Otto Waltari and Jussi Kangasharju. The wireless shark: Identifying wifi devices based on probe fingerprints. In *Proceedings of the First Workshop on Mobile Data*, pages 1–6. ACM, 2016.

[205] Bingchen Wang, Sigeru Omatu, and Toshiro Abe. Identification of the defective transmission devices using the wavelet transform. *IEEE transactions on pattern analysis and machine intelligence*, 27(6):919–928, 2005.

[206] Winkey Wang. Wireless networking in Windows 10. In *Windows Hardware Engineering Community conference (WinHEC)*, March 2015.

[207] Samuel D Warren and Louis D Brandeis. The right to privacy. *Harvard law review*, pages 193–220, 1890.

[208] Michael Weigand, Renderman, and Mike Kershaw. Build your own UAV 2.0 - wireless mayhem from the heavens. In *DEFCON*, 2010.

[209] Jens Weppner, Benjamin Bischke, and Paul Lukowicz. Monitoring crowd condition in public spaces by tracking mobile consumer devices with wifi interface. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, pages 1363–1371. ACM, 2016.

[210] Jens Weppner, Paul Lukowicz, Ulf Blanke, and Gerhard Tröster. Participatory bluetooth scans serving as urban crowd probes. *IEEE Sensors Journal*, 14(12):4196–4206, 2014.

[211] Alan F Westin. Privacy and freedom. *Washington and Lee Law Review*, 25(1):166, 1968.

[212] Wi-Fi Alliance. *Wi-Fi Simple Configuration Protocol and Usability Best Practices for the Wi-Fi Protected Setup Program, v2.0.1*, April 2011.

[213] Björn Wiedersheim, Zhendong Ma, Frank Kargl, and Panos Papadimitratos. Privacy in inter-vehicular networks: Why simple pseudonym change is not enough. In *Wireless On-demand Network Systems and Services (WONS)*, pages 176–183. IEEE, 2010.

[214] Glenn Wilkinson. Digital terrestrial tracking: The future of surveillance. *DEFCON*, 22, 2014.

[215] Martin Woolley. Bluetooth technology protecting your privacy. `https://blog.bluetooth.com/bluetooth-technology-protecting-your-privacy`, consulted on 2017.05.28, 2015.

[216] David J Wu, Ankur Taly, Asim Shankar, and Dan Boneh. Privacy, discovery, and authentication for the internet of things. In *European Symposium on Research in Computer Security*, pages 301–319. Springer, 2016.

[217] Tong Xin, Bin Guo, Zhu Wang, Mingyang Li, Zhiwen Yu, and Xingshe Zhou. Freesense: Indoor human identification with Wi-Fi signals. In *Global Communications Conference (GLOBECOM), 2016 IEEE*, pages 1–7. IEEE, 2016.

[218] Qiang Xu, Rong Zheng, Walid Saad, and Zhu Han. Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys & Tutorials*, 18(1):94–104, 2016.

[219] Stanley Young. Bluetooth traffic detectors for use as permanently installed travel time instruments. *State Highway Administration Research Report*, 2013.

[220] Bin Zan, Zhanbo Sun, Macro Gruteser, and Xuegang Ban. Linking anonymous location traces through driving characteristics. In *Proceedings of the third ACM conference on Data and application security and privacy*, pages 293–300. ACM, 2013.

[221] Zebra technologies. Analysis of ios 8 MAC randomization on locationing. `http://mpact.zebra.com/documents/iOS8-White-Paper.pdf`, 2015.

[222] Yunze Zeng, Parth H Pathak, and Prasant Mohapatra. Analyzing shopper's behavior through wifi signals. In *Proceedings of the 2nd workshop on Workshop on Physical Analytics*, pages 13–18. ACM, 2015.

Appendix A

Full Burst

This Appendix describes a full burst of probe requests recorded on 4 simultaneous interfaces set on different channels. Note that an interface may record frames sent on different channels (for instance, frame with sequence number 4021 is seen by interfaces on channels 1 and 5). All probe requests were sent using the same random MAC address.

time	SN	channel	SSID
0.000000	4001	1	nexus 6P
0.002190	4002	1	toto
0.004746	4003	1	
0.038640	4004	1	nexus 6P
0.041442	4005	1	toto
0.044120	4006	1	
0.102281	4011	1	toto
0.174707	4021	1	toto
0.277277	4036	1	toto
3.120354	**4073**	1	nexus 6P
0.044149	4006	5	
0.102304	4011	5	toto
0.104485	4012	5	
0.124419	4014	5	toto
0.131066	4015	5	nexus 6P
0.133774	4016	5	toto
0.136479	4017	5	
0.149190	4018	5	nexus 6P
0.158757	4019	5	toto
0.164263	4020	5	nexus 6P
0.174730	4021	5	toto
0.177289	4022	5	
0.219703	4028	5	
0.290583	4038	5	nexus 6P
0.330351	4044	5	toto
0.332410	4045	5	
0.341055	4046	5	nexus 6P
0.343223	4047	5	toto
0.350583	4048	5	
0.216875	4027	9	toto
0.234982	4029	9	nexus 6P
0.250164	4030	9	toto
0.252366	4031	9	
0.261443	4033	9	toto
0.264212	4034	9	
0.274448	4035	9	nexus 6P
0.277300	4036	9	toto
0.279977	4037	9	
0.290602	4038	9	nexus 6P
0.293365	4039	9	toto
0.295610	4040	9	
0.308012	4041	9	nexus 6P
0.311961	4042	9	toto
0.450759	4067	64	nexus 6P
0.450820	4068	64	toto
0.451464	4069	64	
0.465958	4070	64	nexus 6P
0.465961	4071	64	toto
0.466727	4072	64	

Appendix B

Example of a Panoptiphone session

```
$ ./panoptiphone.sh wlan0 # Live capture
Capturing on 'wlan0'
MAC address: c0:ee:fb:75:0d:59 (OnePlus Tech (Shenzhen) Ltd)
One in 13654.92 devices share this signature
Field                            | Entropy | One in x devices have this value | value
wps.uuid_e                       |  0.528  |                          5606.000 |
wlan_mgt.tag.number              |  0.483  |                        163812.000 | 0,1,50,3,45,221,127
wlan_mgt.supported_rates         |  0.304  |                        163793.000 | 2,4,11,22
wlan_mgt.extended_supported_rates |  0.302  |                        162962.000 | 12,18,24,36,48,72,96,108
wlan_mgt.ht.capabilities.psmp    |  0.301  |                        162962.000 | 0x0000012c
wlan_mgt.ht.ampduparam           |  0.000  |                             1.000 | 0x00000003
[...]
total                            |  3.489  |

$ python panoptiphone.py -d # dump database
163858 devices in the database
Information element | Entropy | Aff dev | Number of values
wlan_mgt.tag.length |   3.959 |  99.97  |  417
wlan_mgt.tag.number |   3.046 |  99.97  |  414
wlan_mgt.ssid       |   3.695 |  99.97  |  20592
[...]
total               |   5.834 |    -    |  163858
29171 devices (17.80%) are unique in the database

$ python panoptiphone.py -v wlan_mgt.txbf.txbf # list possible values of a field
Value     | Number of times seen
0;0       | 115512
0         | 17353
FFFFFFFF  | 4
```

Figure B.1 – Example output of several commands of Panoptiphone.

Appendix C

Full Information Elements List

This appendix lists all Informations Elements encountered in the `Sapienza` dataset. Unlike Table V.2, this presents all subfields without any grouping, yielding a different presentation of the results. First column gives IEs using their `libpcap` name. Second column gives the entropy, as calculated in section V.3. Third column gives the fraction of devices possessing related IE in their probe requests. Last column indicates the number of possible values for this element.

Information Element (`libpcap` name)	Entropy	Aff dev	Values
wlan_mgt.tag.length	3.959	99.97	417
wlan_mgt.ssid	3.695	99.97	20592
wlan_mgt.tag.number	3.046	99.97	414
wlan_mgt.tag.vendor.data	2.518	77.41	95
wlan_mgt.ht.ampduparam.maxlength	2.411	81.09	9
wlan_mgt.tag.vendor.oui.type	2.128	79.84	33
wlan_mgt.tag.oui	2.126	79.84	38
wlan_mgt.ht.capabilities.sm	1.958	81.09	7
wlan_mgt.ht.capabilities.amsdu	1.897	81.09	5
wlan_mgt.ht.capabilities.short20	1.845	81.09	5
wlan_mgt.ht.capabilities.dsscck	1.844	81.09	5
wlan_mgt.ht.capabilities.rxstbc	1.795	81.09	7
wlan_mgt.ds.current_channel	1.681	52.91	17
wlan_mgt.supported_rates	1.343	99.96	36
wlan_mgt.ht.ampduparam.mpdudensity	1.334	81.09	9
wlan_mgt.ht.capabilities.green	1.278	81.09	5
wlan_mgt.ht.mcsset.highestdatarate	1.277	81.09	10
wlan_mgt.ht.mcsset.rxbitmask.8to15	1.251	81.09	5
wlan_mgt.ht.mcsset.txsetdefined	1.248	81.09	4
wlan_mgt.ht.capabilities.short40	1.224	81.09	5
wlan_mgt.ht.mcsset.rxbitmask.16to23	1.193	81.09	5
wlan_mgt.htex.capabilities.htc	1.184	81.09	5
wlan_mgt.ht.capabilities.width	1.183	81.09	5
wlan_mgt.ht.capabilities.txstbc	1.176	81.09	5
wlan_mgt.htex.capabilities.rdresponder	1.171	81.09	4
wlan_mgt.ht.mcsset.rxbitmask.32	1.171	81.09	5
wlan_mgt.ht.capabilities.ldpccoding	1.170	81.09	5
wlan_mgt.ht.mcsset.rxbitmask.0to7	1.167	81.09	5
wlan_mgt.htex.capabilities.transtime	1.165	81.09	4
wlan_mgt.htex.capabilities.mcs	1.165	81.09	4
wlan_mgt.ht.mcsset.txrxmcsnotequal	1.155	81.09	4
wlan_mgt.ht.capabilities.40mhzintolerant	1.154	81.09	5
wlan_mgt.txbf.txss	1.153	81.09	5
wlan_mgt.txbf.txbf	1.153	81.09	4
wlan_mgt.txbf.fm.uncompressed.maxant	1.153	81.09	4
wlan_mgt.txbf.fm.compressed.tbf	1.153	81.09	4
wlan_mgt.txbf.mingroup	1.153	81.09	4
wlan_mgt.txbf.fm.uncompressed.tbf	1.153	81.09	4
wlan_mgt.txbf.fm.compressed.maxant	1.153	81.09	4

wlan_mgt.txbf.csi.maxrows	1.153	81.09	4
wlan_mgt.txbf.rxss	1.153	81.09	4
wlan_mgt.txbf.channelest	1.153	81.09	4
wlan_mgt.txbf.fm.compressed.bf	1.153	81.09	4
wlan_mgt.txbf.fm.uncompressed.rbf	1.153	81.09	4
wlan_mgt.ht.mcsset.txmaxss	1.153	81.09	4
wlan_mgt.ht.mcsset.rxbitmask.33to38	1.153	81.09	3
wlan_mgt.txbf.calibration	1.153	81.09	3
wlan_mgt.htex.capabilities.pco	1.153	81.09	3
wlan_mgt.txbf.csinumant	1.153	81.09	3
wlan_mgt.txbf.rxndp	1.153	81.09	3
wlan_mgt.ht.mcsset.txunequalmod	1.153	81.09	3
wlan_mgt.txbf.txndp	1.153	81.09	3
wlan_mgt.asel.txcsi	1.153	81.09	3
wlan_mgt.ht.mcsset.rxbitmask.24to31	1.153	81.09	3
wlan_mgt.ht.mcsset.rxbitmask.39to52	1.153	81.09	3
wlan_mgt.asel.reserved	1.153	81.09	3
wlan_mgt.asel.csi	1.153	81.09	3
wlan_mgt.asel.if	1.153	81.09	3
wlan_mgt.ht.capabilities.lsig	1.153	81.09	3
wlan_mgt.asel.txif	1.153	81.09	3
wlan_mgt.asel.rx	1.153	81.09	3
wlan_mgt.txbf.impltxbf	1.153	81.09	3
wlan_mgt.txbf.rcsi	1.153	81.09	3
wlan_mgt.txbf.csi	1.153	81.09	3
wlan_mgt.asel.sppdu	1.153	81.09	3
wlan_mgt.txbf.reserved	1.153	81.09	3
wlan_mgt.ht.capabilities.psmp	1.153	81.09	3
wlan_mgt.asel.capable	1.153	81.09	3
wlan_mgt.ht.mcsset.rxbitmask.53to76	1.153	81.09	3
wlan_mgt.ht.capabilities.delayedblockack	1.153	81.09	3
wlan_mgt.ht.ampduparam.reserved	1.153	81.09	3
wlan_mgt.wfa.ie.type	1.045	29.78	13
wlan_mgt.vs.pren.type	0.875	70.50	2
wlan_mgt.extended_supported_rates	0.866	99.45	27
wps.uuid_e	0.614	3.42	5355
wps.device_name	0.338	3.49	234
wps.length	0.329	3.50	132
wps.model_name	0.327	3.48	184
wps.model_number	0.322	3.48	130
wps.manufacturer	0.303	3.48	53
wps.config_methods	0.274	3.50	29
wps.primary_device_type	0.269	3.50	19
wps.config_methods.virt_display	0.254	3.50	15
wps.config_methods.phy_display	0.254	3.50	15
wps.rf_bands	0.244	3.50	13
wps.config_methods.virt_pushbutton	0.232	3.50	15
wps.config_methods.pushbutton	0.232	3.50	15
wps.type	0.230	3.50	17
wps.config_methods.keypad	0.229	3.50	11
wps.config_methods.label	0.228	3.50	11
wps.request_type	0.225	3.50	12
wps.vendor_extension	0.224	3.48	12
wps.ext.id	0.224	3.48	11
wps.ext.len	0.223	3.48	10
wps.config_methods.display	0.223	3.50	11
wps.device_password_id	0.222	3.50	11
wps.config_methods.phy_pushbutton	0.222	3.50	11
wps.version	0.222	3.50	11
wps.configuration_error	0.221	3.50	10
wps.association_state	0.221	3.50	10
wps.config_methods.usba	0.221	3.50	10
wps.config_methods.nfcext	0.221	3.50	10
wps.config_methods.ethernet	0.221	3.50	10
wps.config_methods.nfcint	0.221	3.50	10
wps.config_methods.nfcinf	0.221	3.50	10
wps.vendor_id	0.221	3.48	11
wps.ext.version2	0.220	3.48	10
wps.primary_device_type.category	0.125	1.50	14
wifi_p2p.p2p_capability.device_capability	0.121	1.28	22
wifi_p2p.listen_channel.channel_number	0.117	1.28	12
wifi_p2p.p2p_capability.device_capability.concurrent_operation	0.114	1.28	15
wifi_p2p.p2p_capability.device_capability.client_discoverability	0.107	1.28	12
wifi_p2p.p2p_capability.device_capability.service_discovery	0.107	1.28	12
wps.primary_device_type.subcategory_telephone	0.102	1.26	12
wifi_p2p.listen_channel.country_string	0.102	1.28	15
wifi_p2p.p2p_capability.group_capability	0.102	1.28	12
wifi_p2p.p2p_capability.group_capability.intra_bss_distribution	0.101	1.28	11
wifi_p2p.length	0.101	1.28	11
wifi_p2p.type	0.101	1.28	11
wifi_p2p.p2p_capability.device_capability.invitation_procedure	0.101	1.28	11
wifi_p2p.p2p_capability.group_capability.persistent_reconnect	0.101	1.28	11
wifi_p2p.p2p_capability.group_capability.group_limit	0.101	1.28	10
wifi_p2p.p2p_capability.group_capability.cross_connection	0.101	1.28	10
wifi_p2p.p2p_capability.group_capability.persistent_group	0.101	1.28	10

wifi_p2p.p2p_capability.device_capability.device_limit	0.101	1.28	10
wifi_p2p.p2p_capability.group_capability.group_formation	0.101	1.28	10
wifi_p2p.listen_channel.operating_class	0.101	1.28	10
wifi_p2p.p2p_capability.device_capability.infrastructure_managed	0.101	1.28	10
wifi_p2p.p2p_capability.group_capability.group_owner	0.101	1.28	10
wlan_mgt.extcap.b0	0.054	0.56	3
wlan_mgt.tag.data	0.052	0.37	181
wlan_mgt.extcap.b1	0.050	0.56	3
wlan_mgt.extcap.b4	0.050	0.56	3
wlan_mgt.extcap.b2	0.050	0.56	3
wlan_mgt.extcap.b3	0.050	0.56	3
wlan_mgt.extcap.b5	0.050	0.56	3
wlan_mgt.extcap.b6	0.050	0.56	3
wlan_mgt.extcap.b7	0.050	0.56	3
_ws.expert.message	0.049	0.55	3
_ws.expert.severity	0.049	0.55	3
_ws.expert.group	0.049	0.55	3
wlan_mgt.tag.data.undecoded	0.035	0.37	3
wps.primary_device_type.subcategory_computer	0.023	0.22	2
wlan_mgt.tag.length.bad	0.019	0.18	2
wps.uuid_r	0.015	0.08	131
wlan_mgt.vs.nintendo.service	0.012	0.07	70
wlan_mgt.vs.nintendo.consoleid	0.011	0.06	90
wlan_mgt.vs.nintendo.length	0.010	0.07	17
wlan_mgt.vht.mcsset.rxmcsmap.ss1	0.010	0.08	3
wlan_mgt.vht.mcsset.txmcsmap.ss1	0.010	0.08	3
wlan_mgt.vht.mcsset.rxhighestlonggirate	0.010	0.08	2
wlan_mgt.vht.capabilities.maxmpdulength	0.010	0.08	2
wlan_mgt.vht.reserved	0.010	0.08	2
wlan_mgt.vht.capabilities.rxldpc	0.010	0.08	2
wlan_mgt.vht.mcsset.txmcsmap.ss8	0.010	0.08	2
wlan_mgt.vht.mcsset.rxmcsmap.ss8	0.010	0.08	2
wlan_mgt.vht.mcsset.rxmcsmap.ss3	0.010	0.08	2
wlan_mgt.vht.capabilities.rxpatconsist	0.010	0.08	2
wlan_mgt.vht.mcsset.rxmcsmap.ss7	0.010	0.08	2
wlan_mgt.vht.mcsset.rxmcsmap.ss6	0.010	0.08	2
wlan_mgt.vht.capabilities.mubeamformer	0.010	0.08	2
wlan_mgt.vht.mcsset.txmcsmap.ss2	0.010	0.08	2
wlan_mgt.vht.capabilities.txstbc	0.010	0.08	2
wlan_mgt.vht.capabilities.supportedchanwidthset	0.010	0.08	2
wlan_mgt.vht.capabilities.soundingdimensions	0.010	0.08	2
wlan_mgt.vht.capabilities.vhthtc	0.010	0.08	2
wlan_mgt.vht.capabilities.short80	0.010	0.08	2
wlan_mgt.vht.capabilities.linkadapt	0.010	0.08	2
wlan_mgt.vht.capabilities.maxampdu	0.010	0.08	2
wlan_mgt.vht.capabilities.vhttxopps	0.010	0.08	2
wlan_mgt.vht.capabilities.subeamformer	0.010	0.08	2
wlan_mgt.vht.capabilities.txpatconsist	0.010	0.08	2
wlan_mgt.vht.capabilities.rxstbc	0.010	0.08	2
wlan_mgt.vht.mcsset.rxmcsmap.ss5	0.010	0.08	2
wlan_mgt.vht.capabilities.subeamformee	0.010	0.08	2
wlan_mgt.vht.mcsset.rxmcsmap.ss4	0.010	0.08	2
wlan_mgt.vht.mcsset.txhighestlonggirate	0.010	0.08	2
wlan_mgt.vht.capabilities.beamformerants	0.010	0.08	2
wlan_mgt.vht.mcsset.txmcsmap.ss3	0.010	0.08	2
wlan_mgt.vht.capabilities.short160	0.010	0.08	2
wlan_mgt.vht.capabilities.mubeamformee	0.010	0.08	2
wlan_mgt.vht.mcsset.rxmcsmap.ss2	0.010	0.08	2
wlan_mgt.vht.mcsset.txmcsmap.ss5	0.010	0.08	2
wlan_mgt.vht.mcsset.txmcsmap.ss4	0.010	0.08	2
wlan_mgt.vht.mcsset.txmcsmap.ss7	0.010	0.08	2
wlan_mgt.vht.mcsset.txmcsmap.ss6	0.010	0.08	2
wlan_mgt.vs.nintendo.type	0.008	0.07	3
wlan_mgt.tag.request	0.006	0.05	5
wps.ext.request_to_enroll	0.005	0.04	2
wlan_mgt.tag.interpretation	0.003	0.02	16
wlan_mgt.extcap.b19	0.003	0.02	3
wlan_mgt.extcap.b9	0.002	0.02	2
wlan_mgt.extcap.b25	0.002	0.02	2
wlan_mgt.extcap.b38	0.002	0.02	2
wlan_mgt.extcap.b39	0.002	0.02	2
wlan_mgt.extcap.b30	0.002	0.02	2
wlan_mgt.extcap.b31	0.002	0.02	2
wlan_mgt.extcap.b32	0.002	0.02	2
wlan_mgt.extcap.b33	0.002	0.02	2
wlan_mgt.extcap.b34	0.002	0.02	2
wlan_mgt.extcap.b36	0.002	0.02	2
wlan_mgt.extcap.b37	0.002	0.02	2
wlan_mgt.extcap.b29	0.002	0.02	2
wlan_mgt.extcap.b28	0.002	0.02	2
wlan_mgt.extcap.b20	0.002	0.02	2
wlan_mgt.extcap.b12	0.002	0.02	2
wlan_mgt.extcap.b27	0.002	0.02	2
wlan_mgt.extcap.b24	0.002	0.02	2
wlan_mgt.extcap.b15	0.002	0.02	2

wlan_mgt.extcap.b26	0.002	0.02	2
wlan_mgt.extcap.b23	0.002	0.02	2
wlan_mgt.extcap.b8	0.002	0.02	2
wlan_mgt.extcap.b22	0.002	0.02	2
wlan_mgt.extcap.b18	0.002	0.02	2
wlan_mgt.extcap.b21	0.002	0.02	2
wlan_mgt.extcap.b35	0.002	0.02	2
wlan_mgt.extcap.b14	0.002	0.02	2
wlan_mgt.extcap.b13	0.002	0.02	2
wlan_mgt.extcap.b10	0.002	0.02	2
wlan_mgt.extcap.b11	0.002	0.02	2
wlan_mgt.extcap.b16	0.002	0.02	2
wlan_mgt.extcap.b17	0.002	0.02	2
wlan_mgt.rsn.gcs.type	0.002	0.01	3
wlan_mgt.rsn.pcs.count	0.002	0.01	2
wlan_mgt.rsn.version	0.002	0.01	2
wlan_mgt.rsn.capabilities.jmr	0.002	0.01	2
wlan_mgt.rsn.capabilities.peerkey	0.002	0.01	2
wlan_mgt.rsn.capabilities.mfpc	0.002	0.01	2
wlan_mgt.rsn.capabilities.mfpr	0.002	0.01	2
wlan_mgt.rsn.pcs.oui	0.002	0.01	2
wlan_mgt.rsn.akms.type	0.002	0.01	2
wlan_mgt.rsn.capabilities.gtksa_replay_counter	0.002	0.01	2
wlan_mgt.rsn.capabilities.ptksa_replay_counter	0.002	0.01	2
wlan_mgt.rsn.gcs.oui	0.002	0.01	2
wlan_mgt.rsn.capabilities.preauth	0.002	0.01	2
wlan_mgt.rsn.pcs.type	0.002	0.01	2
wlan_mgt.rsn.akms.count	0.002	0.01	2
wlan_mgt.rsn.akms.oui	0.002	0.01	2
wlan_mgt.rsn.capabilities.no_pairwise	0.002	0.01	2
wps.primary_device_type.subcategory_displays	0.002	0.01	2
wlan_mgt.extcap.b62	0.001	0.01	3
wlan_mgt.extcap.b48	0.001	0.01	2
wlan_mgt.extcap.b40	0.001	0.01	2
wlan_mgt.extcap.b45	0.001	0.01	2
wlan_mgt.extcap.b47	0.001	0.01	2
wlan_mgt.extcap.b44	0.001	0.01	2
wlan_mgt.extcap.b46	0.001	0.01	2
wlan_mgt.extcap.o7	0.001	0.01	2
wlan_mgt.extcap.serv_int_granularity	0.001	0.01	2
wlan_mgt.extcap.o8	0.001	0.01	2
wlan_mgt.extcap.b63	0.001	0.01	2
wlan_mgt.extcap.b61	0.001	0.01	2
wlan_mgt.ric_desc.rsrc_type	0.001	0.00	9
wifi_display.subelem.dev_info.max_throughput	0.001	0.01	4
wifi_display.subelem.dev_info.content_protection	0.001	0.01	3
wifi_display.subelem.dev_info.control_port	0.001	0.01	3
wifi_display.subelem.session.reserved	0.001	0.01	2
wifi_display.subelem.dev_info.coupled_sink_by_sink	0.001	0.01	2
wifi_display.subelem.session.tdls_persistent_group	0.001	0.01	2
wifi_display.subelem.session.audio_unsupp_pri_sink	0.001	0.01	2
wifi_display.subelem.dev_info.pc	0.001	0.01	2
wifi_display.subelem.length	0.001	0.01	2
wifi_display.subelem.dev_info.wsd	0.001	0.01	2
wifi_display.subelem.session.tdls_persistent_group_reinvoke	0.001	0.01	2
wifi_display.subelem.id	0.001	0.01	2
wifi_display.subelem.dev_info.time_sync	0.001	0.01	2
wifi_display.subelem.dev_info.available	0.001	0.01	2
wifi_display.subelem.dev_info.type	0.001	0.01	2
wifi_display.subelem.session.audio_only_supp_source	0.001	0.01	2
wifi_display.subelem.dev_info.coupled_sink_by_source	0.001	0.01	2

FOLIO ADMINISTRATIF

THESE DE L'UNIVERSITE DE LYON OPEREE AU SEIN DE L'INSA LYON

NOM : Matte
(avec précision du nom de jeune fille, le cas échéant)

DATE de SOUTENANCE : 07/12/2017

Prénoms : Célestin Léon Charles

TITRE : Wi-Fi tracking : Fingerprinting Attacks and Counter-Measures

NATURE : Doctorat

Numéro d'ordre : 2017LYSEI114

Ecole doctorale : InfoMaths

Spécialité : Informatique

RESUME :

Le récent développement des appareils portatifs possédant une interface Wi-Fi (smartphones, tablettes et « wearables ») s'accompagne d'une menace sur la vie privée de leurs utilisateurs, et sur la société toute entière. Ces appareils émettent en continu des signaux pouvant être capturés par un attaquant passif, à l'aide de matériel peu coûteux et de connaissances basiques. Ces signaux contiennent un identifiant unique appelé l'adresse MAC.
Pour faire face à cette menace, les acteurs du secteur déploient actuellement une contre-mesure sur les appareils récents: le changement aléatoire de l'adresse MAC.
Malheureusement, nous montrons que cette mesure, dans son état actuel, n'est pas suffisante pour empêcher le traçage des appareils.

Pour cela, nous introduisons plusieurs attaques basées sur le contenu et la répartition temporelle des signaux. En complément, nous étudions les implémentations du changement aléatoire de l'adresse MAC sur des appareils récents, et trouvons un certain nombre de manquements limitant l'efficacité de ces implémentations à prévenir le traçage.

En parallèle, nous effectuons deux études de terrain. La première s'attaque au développement des acteurs exploitant les problèmes cités plus haut afin d'installer des systèmes de traçage basés sur le Wi-Fi. Nous listons certaines de ces installations et examinons plusieurs aspects de ces systèmes : leur régulation, les implications en terme de vie privée, les questions de consentement et leur acceptation par le public. La seconde étude concerne la progression du changement aléatoire d'adresse MAC dans la population des appareils.

Finalement, nous présentons deux outils : le premier est un système de traçage expérimental développé pour effectuer des tests et sensibiliser le public aux problèmes de vie privée liés à de tels systèmes. Le second estime l'unicité d'un appareil en se basant sur le contenu des signaux qu'il émet, même si leur identifiant est modifié.

MOTS-CLÉS : Informatique, Vie privée, Prise d'empreintes (fingerprinting), Wi-Fi, Traçage, Probe request, changement aléatoire de l'adresse MAC

Laboratoire (s) de recherche : CITI

Directeur de thèse: Marine Minier
Composition du jury :

Rapporteurs

Civilité	Nom	Prénom	Grade / Qualité	Etablissement	Email
M.	NGUYEN	Benjamin	Professeur des Universités	INSA Centre Val de Loire	benjamin.nguyen@insa-cvl.fr
M.	RASMUSSEN	Kasper	Associate Professor	University of Oxford	kasper.rasmussen@cs.ox.ac.uk

Membres du jury comprenant également les rapporteurs s'ils en font partie

Civilité	Nom	Prénom	Grade / Qualité	Etablissement	Email
M.	NGUYEN	Benjamin	Professeur des Universités	INSA Centre Val de Loire	benjamin.nguyen@insa-cvl.fr
M.	RASMUSSEN	Kasper	Associate Professor	University of Oxford	kasper.rasmussen@cs.ox.ac.uk
MME	CHRISMENT	Isabelle	Professeur des Universités	Université de Lorraine	isabelle.chrisment@loria.fr
M	RISSET	Tanguy	Professeur des Universités	INSA-LYON	tanguy.risset@insa-lyon.fr
M.	NEUMANN	Christoph	Principal Scientist	Tecnicolor	christoph.neumann@technicolor.com
MME	MINIER	Marine	Professeur des Universités	Université de Lorraine	marine.minier@loria.fr
M.	CUNCHE	Mathieu	Maître de conférences	INSA-LYON	mathieu.cunche@insa-lyon.fr

Président de jury : Isabelle Chrisment