Tracking without Traces

Fingerprinting in an Era of Individualism and Complexity

Florian Adamsky[1], Stefan Schiffner[2], and Thomas Engel[2]

October 22, 2020

University of Applied Sciences Hof[1]
University of Luxembourg[2]

Outline

Introduction

What is Fingerprinting (FP)?

- Uniquely identifying software or hardware
- Collecting attributes and creating a unique fingerprint
- We distinguish between:

  Active FP requires interaction with the Software or Hardware

  Passive FP requires no interaction with the Software or Hardware

Use cases for fingerprinting

Security

Positioning

Privacy Attack

Use cases for fingerprinting

Security

Positioning

Privacy Attack

Use cases for fingerprinting

Security

Positioning

Privacy Attack

Use cases for fingerprinting

Security

Positioning

Privacy Attack

Software Attributes

Desktop Browser Fingerprinting

Table 1: Example of a browser fingerprint

Attribute	Value
User-agent	Mozilla/5.0 (Linux x86 64) Gecko Firefox/67.0
HTTP ACCEPT Headers	text/html, /; q=0.01 gzip, deflate, br en-US,en;q=0.5
Time Zone	-480 (UTC+8)
Screen Size and Color Depth	1920x1080x24
Plugins	Shockwave Flash, ...
System Fonts	Andale Mono, Arial, Bitstream Vera Sans Mono, ...
Language	en-US
Platform	Linux x86 64

Mobile Browser Fingerprint

- Similar to a desktop browser fingerprint, but there are no plugins
- User-agent is more unique compared to desktop browser:

Example (User-agent strings on a mobile device)

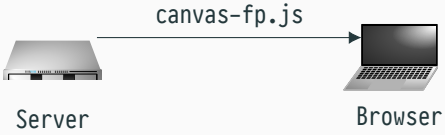
```
Mozilla/5.0 (iPhone; CPU iPhone OS 8_1_1 like Mac OS X) AppleWebKit/600.1.4 (KHTML,
like Gecko) Mobile/12B436 [FBAN/FBIOS;FBAV/20.1.0.15.10;
FBBV/5758778;FBDV/iPhone7,2;FBMD/iPhone;FBSN/iPhoneOS;FBSV/8.1.1;FBSS/2;
FBCR/vodafoneUK ;FBID/phone;FBLC/en_GB;FBOP/5]
```

Canvas Fingerprinting

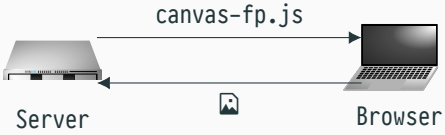
Server

Browser

Canvas Fingerprinting



Canvas Fingerprinting



Canvas Fingerprinting (cont.)

Figure 1: Example from [2].

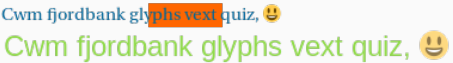


Figure 2: Example from Firefox on Linux.

Figure 3: Example from Firefox on MacOS.

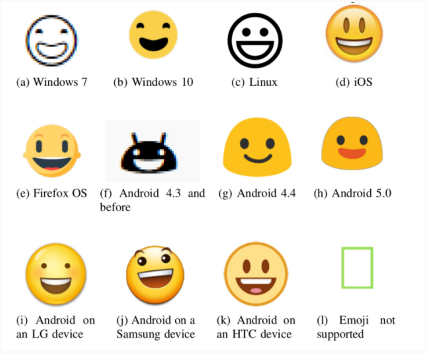


Figure 4: Different smileys on different operation systems. Source: [2]

WebGL Fingerprinting

- Some browsers publish the graphic card that is used to render WebGL
- Some browsers also publish the driver for the graphic card:

Example of WebGL Vendor and Renderer	
WebGL Vendor:	Intel
WebGL Renderer:	Mesa Intel(R) UHD Graphics 620 (...)

Figure 5: Example of WebGL Data.

Figure 6: Example of my Browser Fingerprint created on `https://amiunique.org/` [2]

Hardware Attributes

What is Device Fingerprinting?

- Creating a unique fingerprint from a device
- Combination of different metrics (attributes)
- Smartphone is getting more and more our unique ID

Smart Bins in London (2012)

Figure 7: Smart Bin in London with targeted advertising. Source: zdnet.com

Smart Bins in London (2012)

Figure 7: Smart Bin in London with targeted advertising. Source: zdnet.com

Smart Bins in London (2012)

Figure 7: Smart Bin in London with targeted advertising. Source: zdnet.com

Smart Bins in London (2012)

Figure 7: Smart Bin in London with targeted advertising. Source: zdnet.com

Crystal Oscillator

- Crystal oscillator are needed to get the transmission frequency
- Due to the manufacturing process, there tiny differences between the crystals
- These differences can be measured with high precision measurement devices
  - Radio Frequency Fingerprinting (RFF)

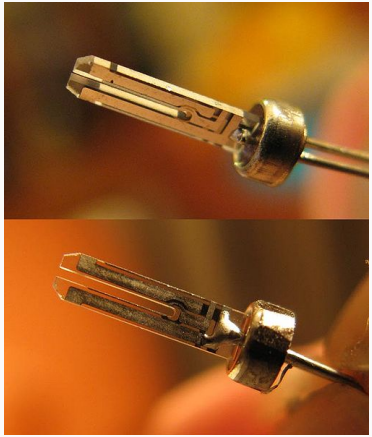


Figure 8: Crystal Oscillator

Device Fingerprinting mit CSI (2018)

Accurate and Efficient Wireless Device Fingerprinting Using Channel State Information

Jingyu Hua*†, Hongyi Sun*†, Zhenyu Shen*†, Zhiyun Qian‡ and Sheng Zhong*†
*State Key Laboratory for Novel Software Technology, Nanjing University, China
†Department of Computer Science and Technology, Nanjing University, China
Email: huajingyu2012@gmail.com, hysquall@gmail.com, shenzhenyuyu@gmail.com, sheng.zhong@gmail.com
‡University of California, Riverside, USA
Email: zhiyunq@cs.ucr.edu

***Abstract*—Due to the loose authentication requirement between access points (APs) and clients, it is notoriously known that WLANs face long-standing threats such as rogue APs and network freeloading. Take the rogue AP problem as an example, unfortunately encryption alone does not provide authentication. APs need to be equipped with certificates that are trusted by clients ahead of time. This requires either the presence of PKI for APs or other forms of pre-established trust (e.g., distributing the certificates offline), none of which is widely used. Before any strong security solution is deployed, we still need a practical solution that can mitigate the problem. In this paper, we explore a non-cryptographic solution that is readily deployable today on end hosts (e.g., smartphones and laptops) without requiring any changes to the APs or the network infrastructure. The solution infers the Carrier Frequency Offsets (CFOs) of wireless devices from Channel State Information (CSI) as their hardware fingerprints without any special hardware requirement. CFO is attributed to the oscillator drift, which is a fundamental physical property that cannot be manipulated easily and remains fairly consistent over time but varies significantly across devices. The real experiments on 23 smartphones and 34 APs (with both identical and different brands) in different scenarios demonstrate that the detection rate could exceed 94%.**

***Index Terms*—device fingerprinting, attack detection, authentication, wireless networks**

I. Introduction

WiFi networks are attracting various kinds of attacks due to their extremely high popularity. Among these attacks, rogue Access Points (i.e., rogue APs) and WiFi Freeloading are the most common which bring significant security and privacy threats. A rogue AP is a device set up by an adversary to mimic the legitimate AP in public places such as coffee shops and shopping malls. It usually uses the same identifiers (basic service set identifier (BSSID) and service set identifier (SSID)) as the original AP. Once users are fooled to connect to it, the adversary could launch man-in-the-middle attack and eavesdrop all the network communications. It has been estimated that almost 20% of corporations have rogue APs in their networks [1]. WiFi freeloading refers to that an unauthorized user compromises or bypasses the authentication of an AP and then gets into the private WLAN for free. Note that a freeloader may be stealing more than just bandwidth considering that he has become an insider of the network.

A key point to defend against these attacks is a strong mutual authentication mechanism between clients and APs. In fact, the wireless security enhancement 802.11i RSNA (Robust Security Network Association) does provide optional mutual authentication using traditional cryptographic methods (i.e., digital certificates), which can make both attacks less likely if implemented properly. Unfortunately, as mentioned by Jana et al. [2], wireless networks using 802.11i RSNA still suffers from vulnerabilities due to several practical issues. For instance, as the signal strength is the only criteria for clients to select AP in the current implementation, users can be tricked to connect to a fake AP with a higher signal strength than that of the real one but does not support any security measures such as RSNA. Even worse, as the management and distribution of digital certificates are extremely cumbersome, most networks simply choose to support only user authentication but never AP authentication. As a result, it is still easy for adversaries to deploy fake APs. For the freeloading attack, although most networks authenticate users, the authentication is based on human-determined passwords which are usually quite weak and can be easily compromised or disclosed especially when all users share one password (e.g., WPA2-PSK).

Motivated by the above, researchers have proposed noncryptographic solutions based on device fingerprinting recently. These solutions are not meant to replace cryptographic solutions. Rather, they aim to provide an extra layer of security in light of the difficulties in adopting cryptographic solutions. Their common idea is to identify physical characteristics of the hardware that can be extracted remotely to fingerprint wireless devices [2]. An example scenario where this technique is helpful is the following: when a user goes to a coffee shop visiting frequently, without fingerprinting techniques, he will easily be tricked to connect to a rogue AP with the same identity as the real one. With fingerprinting, if the fingerprint (remembered by the client) changes, the user could be alerted that he is likely connecting to a rogue AP.

While a promising direction, there is no real-world deployment of such fingerprinting techniques to our knowledge. This is due to several important practical issues. First, the solution may require special hardware which hinders the

‡ Hua was supported in part by NSFC-61300235. This work was supported in part by NSFC-61472054, NSFC-61402221, the Jiangsu Province Double Innovation Talent Program, and NSFC-61321491. (Corresponding author: Sheng Zhong.)

POSTER: WLAN Device Fingerprinting using Channel State Information (CSI)

Florian Adamsky
SnT, University of Luxembourg
florian.adamsky@uni.lu

Tatiana Retunskaia
CSC, University of Luxembourg
tatiana.retunskaia.001@student.uni.lu

Stefan Schiffner
SnT, University of Luxembourg
stefan.schiffner@uni.lu

Christian Köbel
Honda R&D Europe
christian_koebel@de.hrdeu.com

Thomas Engel
SnT, University of Luxembourg
thomas.engel@uni.lu

ABSTRACT

As of IEEE 802.11n, a wireless Network Interface Card (NIC) uses Channel State Information (CSI) to optimize the transmission over multiple antennas. CSI contain radio-metrics such as amplitude and phase. Due to scattering during hardware production these metrics exhibit unique properties. Since these information are transmitted unencrypted, they can be captured by a passive observer. We show that these information can be used to create a unique fingerprint of a wireless device, based on an little as 100 CSI packets per device collected with an off-the-shelf Wi-Fi card. For our proof of concept we captured data from seven smartphones including two identical models. We were able to identify more than 98% when using out-of-the-box Random Forest (RF).

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**;

ACM Reference Format:
Florian Adamsky, Tatiana Retunskaia, Stefan Schiffner, Christian Köbel, and Thomas Engel. 2018. POSTER: WLAN Device Fingerprinting using Channel State Information (CSI). In *WiSec '18 Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks, June 18–20, 2018, Stockholm, Sweden.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3212480.3226099

1 INTRODUCTION

Smart devices, such as smartphones or smartwatches, are our constant companion and most of them have wireless LAN (WLAN) integrated. There is an ongoing research effort to collect information from a WLAN device to generate a unique fingerprint. In the security arms race, a unique device fingerprint of a smart device can be used on both sides. On the one hand, it can be used as a security mechanism to authenticate a device or a person. On the other hand, it can be used to track devices and therefore invade user's privacy.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WiSec '18, June 18–20, 2018, Stockholm, Sweden
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-5731-9/18/6…$15.00
https://doi.org/10.1145/3212480.3226099

Starting from IEEE 802.11n, WLAN has support for Multiple Input, Multiple Output (MIMO), which allows to receive and to send information over multiple antennas. To optimize the transmission over multiple antennas and to adapt it to current channel conditions, IEEE 802.11n uses physical information about the wireless signal (CSI).

In this work, we show our preliminary results to create a unique fingerprint with RF based on CSI. These CSI can be obtained by a passive observer. There is no need to be associated with the smartphone.

2 EXPERIMENTAL RESULTS

Every wireless NIC that supports IEEE 802.11n measures and receives CSI. Thus our experiments can be translated to any standard hardware. For more convenience, we used a Intel 5300 NIC, since there is a modified firmware and driver available to extract the CSI easily, cf. [1]. The CSI contain the hardware timestamp, frame counter, number of receiving antenna, number of sending antennas, Received Signal Strength Indication (RSSI) of each antenna, noise, Automatic Gain Control (AGC), permutation matrix, rate, and the amplitude and phase for the first 30 subcarries in form of complex matrix. We have captured the CSI with 1000 ICMP echo replies from seven smartphones, including two identical models, which are listed in Table 1.

Table 1: List of our tested smartphones.

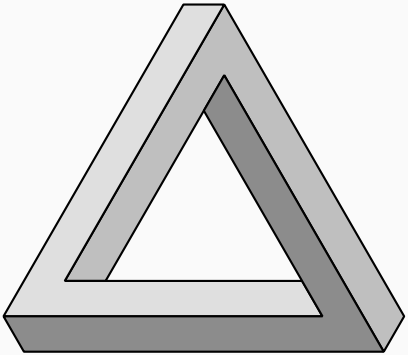
Abbreviation	Brand of Smartphone	Operation System
Asv1	Asus Zenphone	Android 5.0
Asv	Huawei P8 Lite	Android 6.0
CLA	LG G5	Android 7.0
FLO	LG G5	Android 7.0
RIO	Samsung S8	Android 7.0
SAS	Nexus 6P	Android 8.1
WLA	Huawei Honor 8	Android 7.0

After we have obtained the data and converted it to format that *Weka* can process, we used the machine learning algorithm RF to distinguish all phones. In our experiment, we trained our model with 10%, 33%, and 66% of the data and validated it with the rest. In the first experiment, we used all features that are available from CSI. In the second one, we only used the phase, to investigate if

Countermeasures

Countermeasures

- Active Countermeasure will enhance fingerprintabilty
- Peter Eckersley [1] called this phenomena the Paradox of Fingerprintable Privacy Enhancing Technologies



ePrivacy Regulation

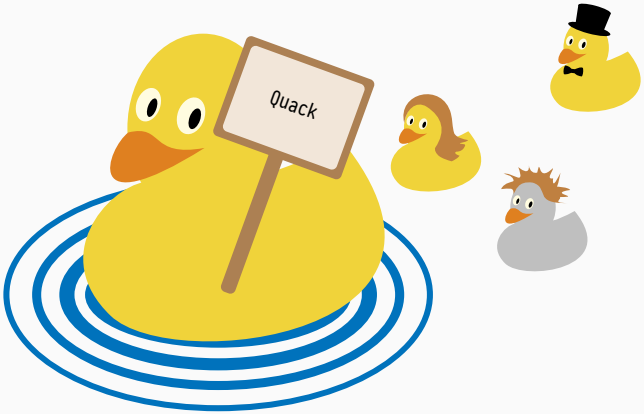
GDPR and ePrivacy

GDPR

ePrivacy

Information Society Service $\neq$ Communication Service

Duck Test



Transparency of data usage enabling fingerprinting

> *Protection of information transmitted to, stored in and related to processed by and collected from users' terminal equipment.*
> *The use of processing and storage capabilities of terminal equipment and the collection of information from end-users' terminal equipment, including about its software and hardware, other than by the user concerned shall be prohibited, except on the following grounds:*
>
> *ePrivacy Regulation Proposal Art 8.*

Exceptions

EU Parliament

- Exceptions are very specific
  - (b) the user has given his of her specific consent;
  - (c) it is strictly technically necessary (...)
  - ...

Exceptions

EU Parliament

- Exceptions are very specific
  - (b) the user has given his of her specific consent;
  - (c) it is strictly technically necessary (...)
  - ...

Council of EU

- Reduce the exceptions drastically
- Introduce concept of legitimate interest

GDPR and ePrivacy

Conclusion

- Fingerprinting is off-the-shelf technology
- Technical countermeasures might be counterproductive
- Further dual use and hard to detect
- hard for individuals to self defend
- Need for clear rules

Questions?

References

[1] Peter Eckersley. “How Unique Is Your Web Browser?” In: *Proceedings of the 10th Privacy Enhancing Technologies Symposium (PETS 2010)*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 1–18. DOI: 10.1007/978-3-642-14527-8_1. (Visited on 10/24/2019).

[2] P. Laperdrix, W. Rudametkin, and B. Baudry. “Beauty and the Beast: Diverting Modern Web Browsers to Build Unique Browser Fingerprints”. In: *2016 IEEE Symposium on Security and Privacy (SP)*. May 2016, pp. 878–894. DOI: 10.1109/SP.2016.57.

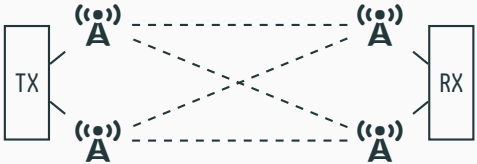
WLAN Standard since 2009

- IEEE 802.11b only supports Single-Input and Single-Output (SISO)

WLAN Standard since 2009

- IEEE 802.11b only supports Single-Input and Single-Output (SISO)

- IEEE 802.11n and higher supports Multiple-Input and Multiple-Output (MIMO)



Channel State Information (CSI)

- Information about the channel
- These information are needed to optime the transmission for the current channel conditions
- Every IEEE 802.11n NIC collects CSI
- A one bit transmission in a 3×3 MIMO scenario would look like the following

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 0.8 & 0 & 0 \\ 0.7 & 0 & 0 \\ 0.9 & 0 & 0 \end{bmatrix}$$

- In reality the CSI matrix contains complex numbers (amplitude and phase)