# Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms

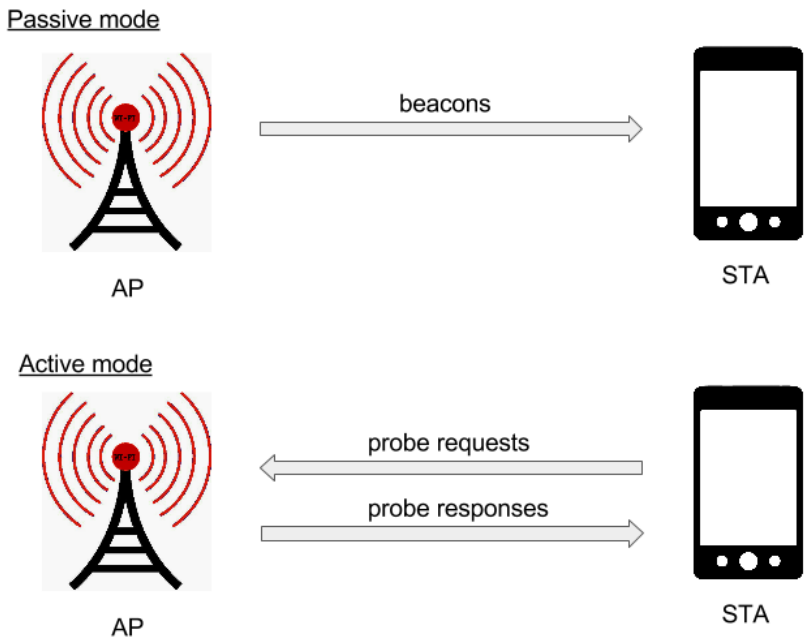
Mathy Vanhoef[†], Célestin Matte[‡], Mathieu Cunche[‡], Leonardo S. Cardoso[‡], Frank Piessens[†]

[†] iMinds-Distrinet, KU Leuven , [‡] Univ Lyon, INSA Lyon, Inria, CITI, France, Région Rhône-Alpes funding

AsiaCCS - June 1st 2016

# Introduction - Wi-Fi service discovery

- Wi-Fi infrastructure mode is asymmetric: access point (AP, "server") and stations (STA) ("clients")
- Stations discover APs by either:
  - listening for incoming `beacon` frames (passive mode),
  - sending `probe request` frames (active mode).
    - Less energy-consuming
    - Containing a unique identifier: the MAC address



# Introduction - Tracking



- What: getting the knowledge of a device's presence over time
- Who: businesses, intelligence services, nasty neighbours, employers...
  - Many retail tracking start-ups: Nomi, Euclid, Purple WiFi...
- Privacy issue: no consent nor awareness

# Introduction

- MAC address randomization proposed to prevent tracking
  - Being deployed in major OSes
    - iOS 8, Android 6, Windows 10, Linux kernel 3.18
- Is it enough to prevent tracking ?
- Maybe not. We show that:
  - Probe requests contain a lot of other information that can be used to fingerprint devices
  - Probe requests contain predictable fields
  - We can force a device to reveal its real MAC address through active attacks
  - All of this allows an attacker to track devices without the use of a stable link-layer identifier

# Introduction - attacker model


- Attacker capabilities
  - Monitoring wireless channels
  - Injecting 802.11 frames (for active attacks only)
- Attacker objectives
  - Tracking devices
  - $\equiv$ Group frames belonging to the same device
- Link-Layer identifier is assumed to change periodically

# Introduction - datasets

Table: Details of the probe requests datasets.

| Dataset | `Lab` | `Train-station` | `Sapienza`[1] |
|---|---|---|---|
| #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 |

[1] **sapienza-probe-requests-20130910**.

# Outline

Part 1/4: Information Elements

# Fingerprinting using Information Elements

- Reminder: Wi-Fi service discovery of (unassociated) devices
- Information elements (a.k.a. tagged parameters, or tags)
  - Indicates the support of capabilities
  - Ex. `Supported Rates`, `High Throughput capabilities` and `Interworking Capabilities`
- High diversity in term of values and in term of information elements present in probe requests
  - Idea: Exploit this diversity to fingerprint devices

# Fingerprinting using Information Elements

```
▼Tag: HT Capabilities (802.11n D1.10)
  Tag Number: HT Capabilities (802.11n D1.10) (45)
  Tag length: 26
 ▼HT Capabilities Info: 0x100c
    .... .... .... ...0 = HT LDPC coding capability: Transmitter does not support receiving LDPC coded packets
    .... .... .... ..0. = HT Support channel width: Transmitter only supports 20MHz operation
    .... .... .... 11.. = HT SM Power Save: SM Power Save disabled (0x0003)
    .... .... ...0 .... = HT Green Field: Transmitter is not able to receive PPDUs with Green Field (GF) preamble
    .... .... ..0. .... = HT Short GI for 20MHz: Not supported
    .... .... .0.. .... = HT Short GI for 40MHz: Not supported
    .... .... 0... .... = HT Tx STBC: Not supported
    .... ..00 .... .... = HT Rx STBC: No Rx STBC support (0x0000)
    .... .0.. .... .... = HT Delayed Block ACK: Transmitter does not support HT-Delayed BlockAck
    .... 0... .... .... = HT Max A-MSDU length: 3839 bytes
    ...1 .... .... .... = HT DSSS/CCK mode in 40MHz: Will/Can use DSSS/CCK in 40 MHz
    ..0. .... .... .... = HT PSMP Support: Won't/Can't support PSMP operation
    .0.. .... .... .... = HT Forty MHz Intolerant: Use of 40 MHz transmissions unrestricted/allowed
    0... .... .... .... = HT L-SIG TXOP Protection support: Not supported
 ▼A-MPDU Parameters: 0x19
    .... ..01 = Maximum Rx A-MPDU Length: 0x01 (16383[Bytes])
    ...1 10.. = MPDU Density: 8 [usec] (0x06)
    000. .... = Reserved: 0x00
 ▶Rx Supported Modulation and Coding Scheme Set: MCS Set
 ▶HT Extended Capabilities: 0x0000
 ▶Transmit Beam Forming (TxBF) Capabilities: 0x0000
 ▶Antenna Selection (ASEL) Capabilities: 0x00
```

Figure: Example of the `HT_Extended_capabilities` Information Element

# Fingerprinting using Information Elements

Empirical evaluation using the datasets

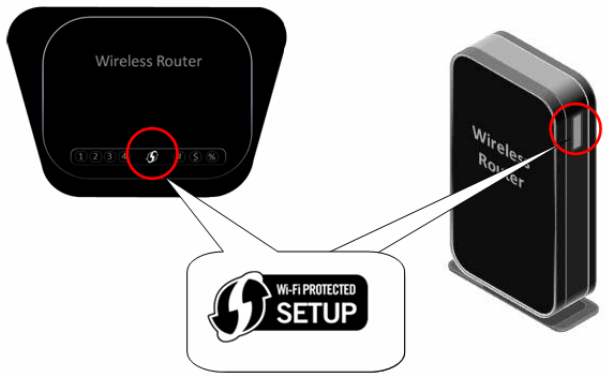
- Considered metrics
  - Fraction of affected devices
  - Entropy: amount of identifying information
- Single Information Elements
  - Can provide up to 5.24 bits of entropy
  - Stable for more than 95% of the devices (no change over time)
  - Some IE are found in almost all devices (`Supported rates`)
  - Ex. `HT capabilities info` (`Train-station` dataset) : 4.74 bits of entropy, 90% of devices affected, stable for 95.9% devices
- Global fingerprint (most common IEs)
  - Entropy : 7.03 bits (`Train-station`)
  - Enough to uniquely identify 1 device among 128 (on average)

See full details in Table 2 of the paper.

# Fingerprinting using Information Elements

Wi-Fi Protected Setup (WPS)

- Information element dedicated to WPS
  - Includes a UUID field
- Universally Unique Identifier UUID
  - A unique identifier *by definition*
  - Generally derived from the MAC address[2]
  - Could be reversed to reveal the original MAC
- Re-identification attack on the datasets
  - UUID derived from the real Wi-Fi MAC address in 75% of the cases



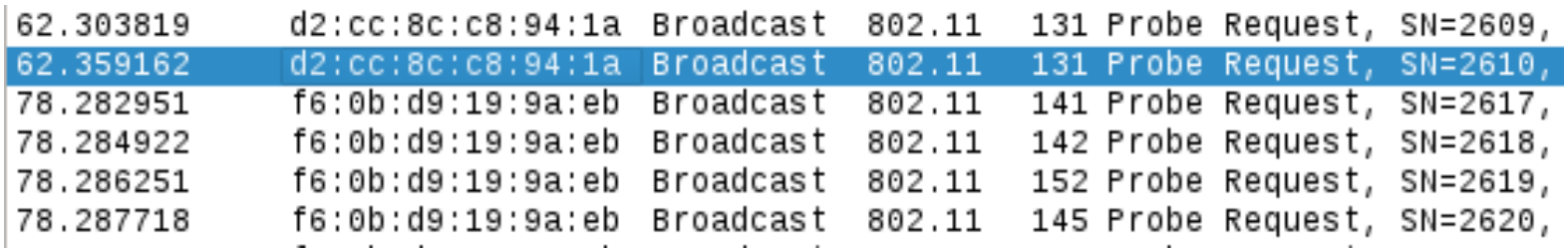

[2] **rfc4122**.

# Outline

Part 2/4: Predictable fields

# Predictable fields

- Predictable fields in 802.11 frames
  - Fields with a content that can change over time
  - Value in a given frame can be predicted from the previous frames
  - Example: `Sequence Number` field
    - Incremented for each frame
    - Not reset when MAC address is changed in iOS[3]
    - Can be used to trivially defeat MAC address randomization



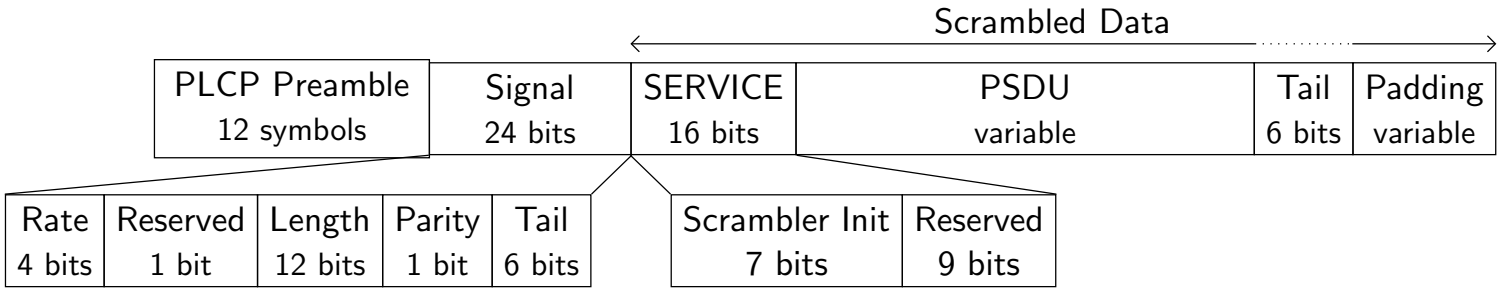
| | | | | |
|---|---|---|---|---|
| 62.303819 | d2:cc:8c:c8:94:1a | Broadcast | 802.11 | 131 Probe Request, SN=2609, |
| 62.359162 | d2:cc:8c:c8:94:1a | Broadcast | 802.11 | 131 Probe Request, SN=2610, |
| 78.282951 | f6:0b:d9:19:9a:eb | Broadcast | 802.11 | 141 Probe Request, SN=2617, |
| 78.284922 | f6:0b:d9:19:9a:eb | Broadcast | 802.11 | 142 Probe Request, SN=2618, |
| 78.286251 | f6:0b:d9:19:9a:eb | Broadcast | 802.11 | 152 Probe Request, SN=2619, |
| 78.287718 | f6:0b:d9:19:9a:eb | Broadcast | 802.11 | 145 Probe Request, SN=2620, |

Figure: Example of a device not resetting its sequence number counter when changing its MAC address

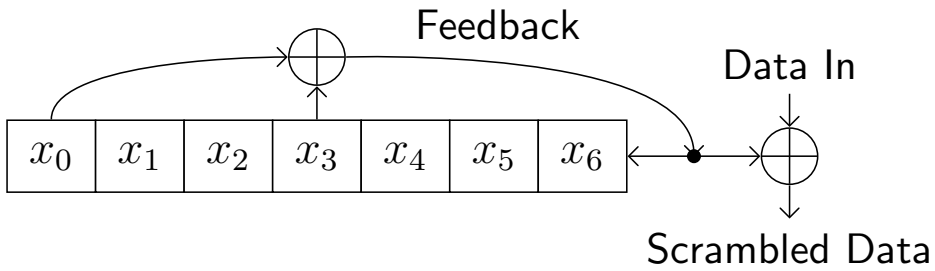
[3] **freudiger-wisec2015**.

# Predictable scrambler seed

- Scrambler in OFDM frames of 802.11 PHY
  - Used to improve frame retransmission
  - Seed contained in the 7 first bits of `SERVICE` field
  - Seed should be different for each frame

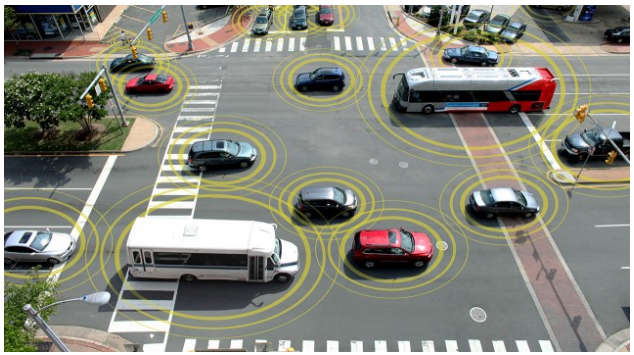



- Scrambling sequence generated by a Linear Feedback Shift Register (LFSR)
  - Seed sets the initial state of LFSR



# Predictable scrambler seed

- Scrambler seeds can be predictable
  - Bloessl et al. showed that it is the case for two prototype implementation of 802.11p[4] (vehicular networks)



  - Possible because no specification in the standard on how to generate the seeds
  - Implementation choice taken by the vendor
- What about commodity 802.11 implementations ?

[4] **bloessl-icnc2015**.

# Predictable scrambler seed

- Study of scrambler seeds in 802.11 commodity hardware
  - Wait, it's a physical layer field
  - Experimental setup
    - GNU-Radio implementation of 802.11 based on `gr-ieee802-11`[5]
    - USRP N210
    - (awesome) Faraday room from FIT CortexLab[6]
    - 11 Wi-Fi commodity hardware

[5] **srif-bloessl2013**.

[6] `http://www.cortexlab.fr/`

# Predictable scrambler seed

- Observed behaviors
  - Constant seed, or limited to a small set (bug ?)
  - Incremental: seed value is incremented by one at each frame
  - Freewheeling: State of the LFSR at the end of a frame is reused for the next frame
- Same as 802.11p: this field can be used to group frames

# Outline

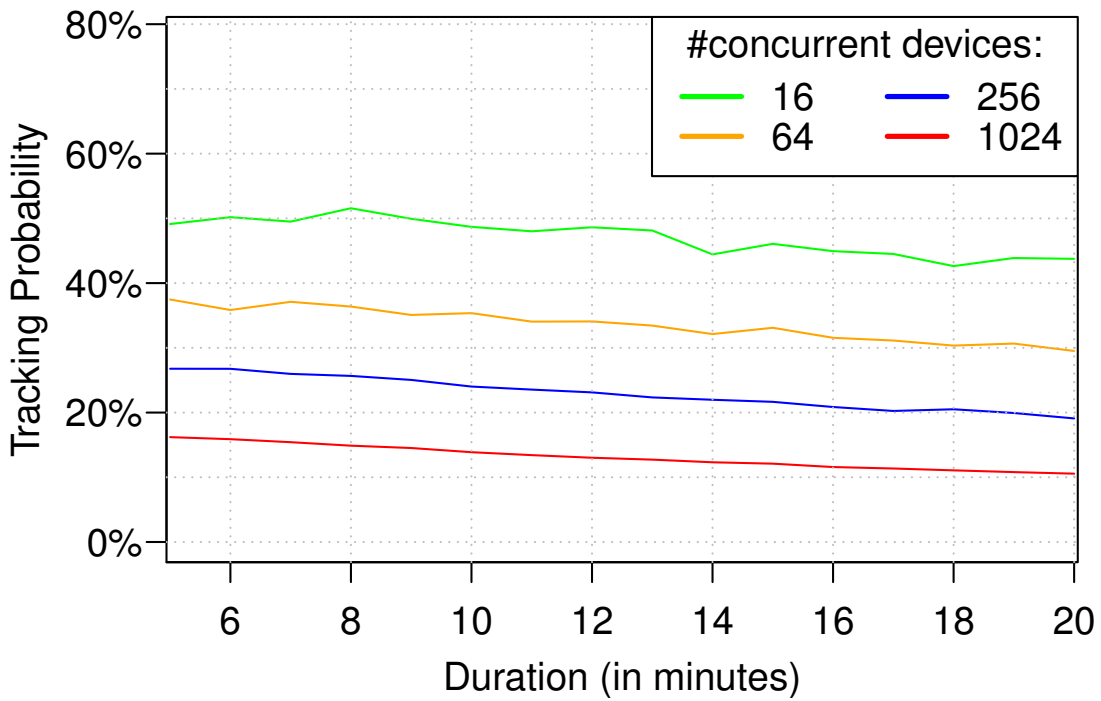
Part 3/4: Tracking algorithm

# Tracking algorithm

- Tracking algorithm:
  - Aim: group probe requests from the same device
  - Use IE fingerprints to create clusters
  - Distinguish devices within these clusters using predictable fields

# Tracking algorithm

- Performances:
  - Strict evaluation conditions: a device is correctly tracked if its frames are grouped in a single cluster containing no frame from other devices
  - Remove obvious identifiers (WPS, SSIDs list)
  - We manage to track 30% of devices during 20 minutes beyond 64 concurrent devices



# Outline

Part 4/4: Active attacks

# Active attacks

Aim: force a device to reveal its actual MAC address

- Karma attack
  - Method:
    - Create fake APs with popular SSIDs (network names)
    - Trigger authentication/association from STA
    - STA switch back to their real MAC when connecting to AP
  - Experiment:
    - Broadcast beacons for the top-5 known SSIDs at the train station
    - Triggered association requests for 17.4% of seen devices

# Active attacks

- Exploiting Hotspot 2.0
  - Method:
    - Enable Wi-Fi roaming
    - STA sends ANQP queries to AP to retrieve list of available services
    - (We show that) STA switches back to their real MAC address when querying
    - Queries also contain predictable counter that could help tracking
  - Experiment:
    - Deployed a fake HS2.0 AP at the train station during two 20-minute sessions
    - 5.25% and 16.37% of stations sent AQNP queries.

# Countermeasures

- Information elements in probe requests
  - Are they really needed? Before association?
  - In all frames by default
  - Remove them or restrict to a bare minimum
- Scrambler seed and other predictable fields
  - Reset to a random value upon MAC address change
  - Unpredictable scrambler seeds
    - Use a crypto PRNG to generate seeds
    - Or chipsets allowing a reset of the seed
- Active attacks
  - HS2.0: Keep random MAC address when sending ANQP queries
  - Use a pseudonym MAC address for associations (Windows 10 model: one pseudonym per network)

# Conclusion

Context:

- MAC address randomization during Wi-Fi service discovery deployed to prevent tracking
- Is it enough?

We showed that:

- Probe request frames contain enough information to fingerprint devices
- Probe request frames contain predictable fields
- Active attacks can reveal the original MAC address

Discussion:

- Not enough specifications, too many details left to vendors' decision
- Privacy not taken into account in specifications
- IEEE 802 privacy study group[7]

[7] http://www.ieee802.org/PrivRecsg/

# Backup slide 1: entropy table

| 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% | - | - | - |

# Backup slide 2: algorithm results with scrambler seed

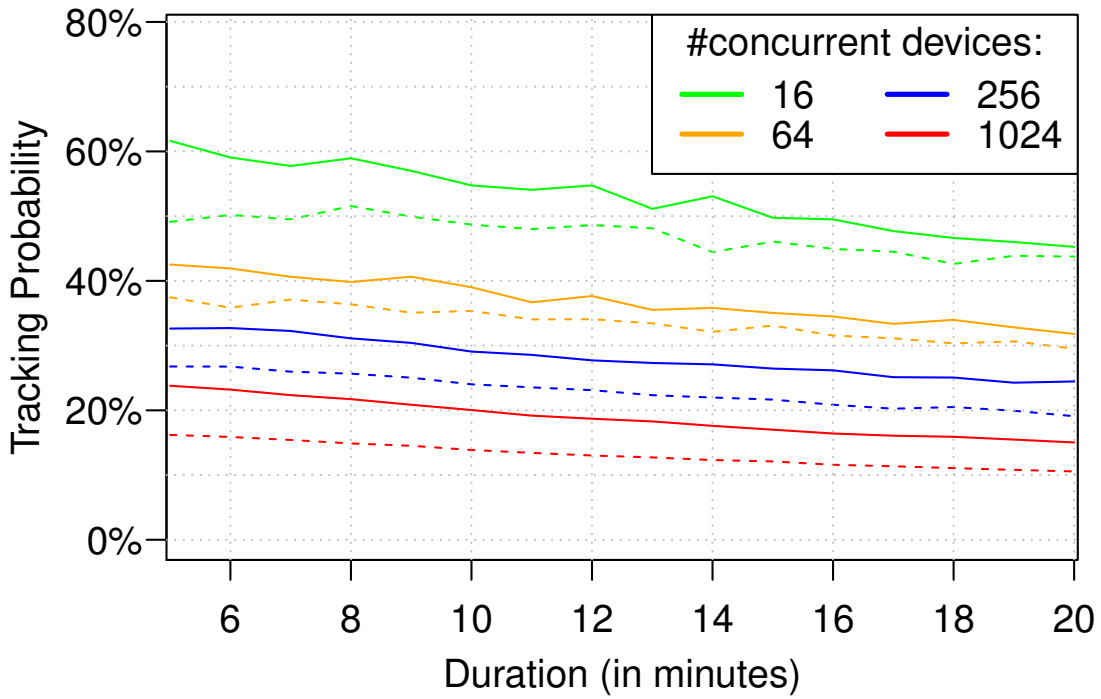
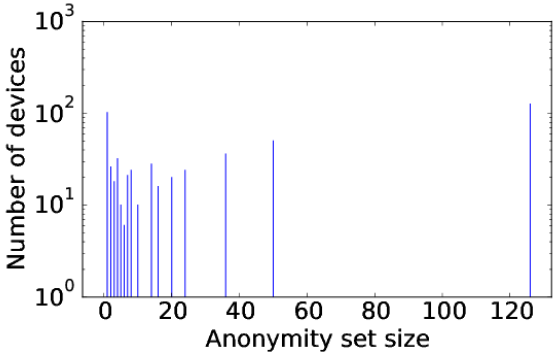
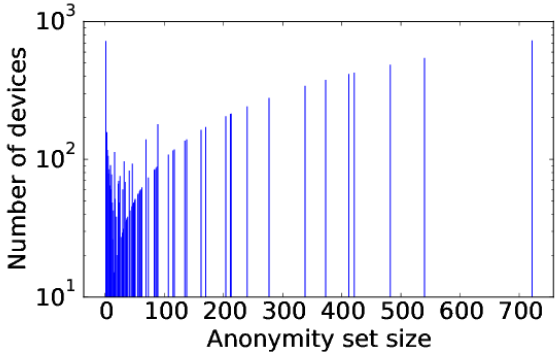



Figure: Performances of the tracking algorithm using the scrambler seed

# Backup slide 3: algorithm results with scrambler seed



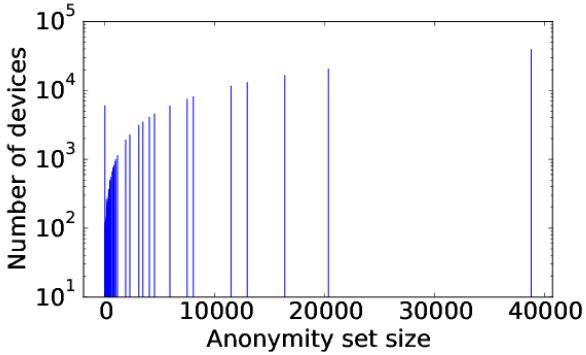
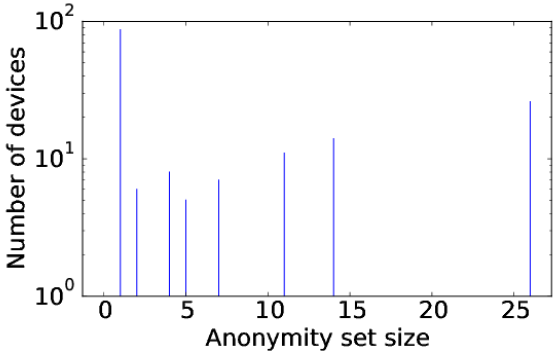
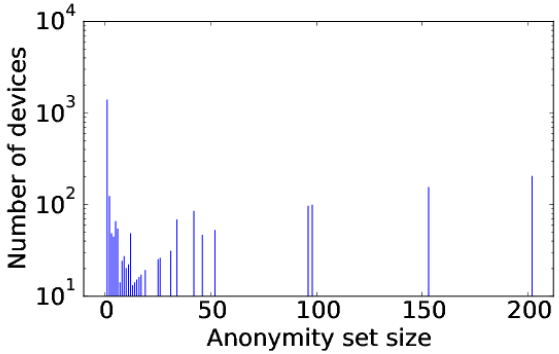




Figure: IE Anonymity sets (Lab, Train station, Sapienza)



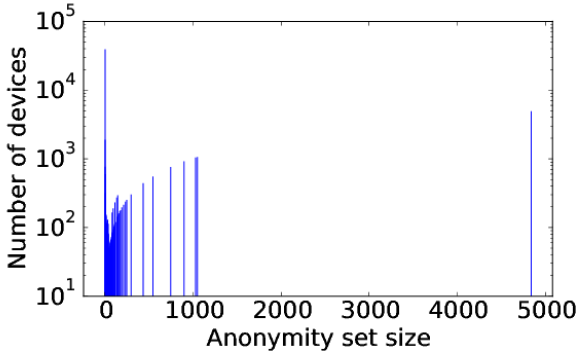




Figure: SSID Anonymity sets (Lab, Train station, Sapienza)