ILLEGAL 3863 INSTRUCTIONS

Find My (?:Device)? 101

The technologies behind Bluetooth location trackers

Agenda

Bytes	Description	Requirement
0-5	MAC address	REQUIRED
6-8	Flags TLV; length = 1 byte, type = 1 byte, value = 1 byte	OPTIONAL
9-12	Service Data TLV; length = 1 byte, type = 0x16, value = 0xFCB2	REQUIRED
13	Network ID	REQUIRED
14	Near-owner bit (1 bit, least significant bit) + reserved (7 bits)	REQUIRED
15-36	Proprietary company payload data	OPTIONAL

ble 1: Location-Enabled Payload Format

```
Bluetooth Low Energy Link Layer
  Access Address: 0x8e89bed6
> Packet Header: 0x2340 (PDU Type: ADV_IND, ChSel: #1, TxAdd: Random)
  Advertising Address: 4a:3c:7e:91:95:b2 (4a:3c:7e:91:95:b2)
v Advertising Data
  v Flags
      Length: 2
      Type: Flags (0x01)
      000. .... = Reserved: 0x0
      ...0 .... = Simultaneous LE and BR/EDR to Same Device Capable (Host): false (0x0)
      .... 0... = Simultaneous LE and BR/EDR to Same Device Capable (Controller): false (0x0)
      .... .1.. = BR/EDR Not Supported: true (0x1)
      .... ..1. = LE General Discoverable Mode: true (0x1)
      .... ...0 = LE Limited Discoverable Mode: false (0x0)
  v Service Data - 16 bit UUID
      Length: 25
      Type: Service Data - 16 bit UUID (0x16)
      UUID 16: Google LLC (0xfeaa)
      Service Data: 40e634838959dc0b03a426189d5f5941f245b76275a4
  CRC: 0x98795d
```

What?!

- Problem statement
- Functional goals

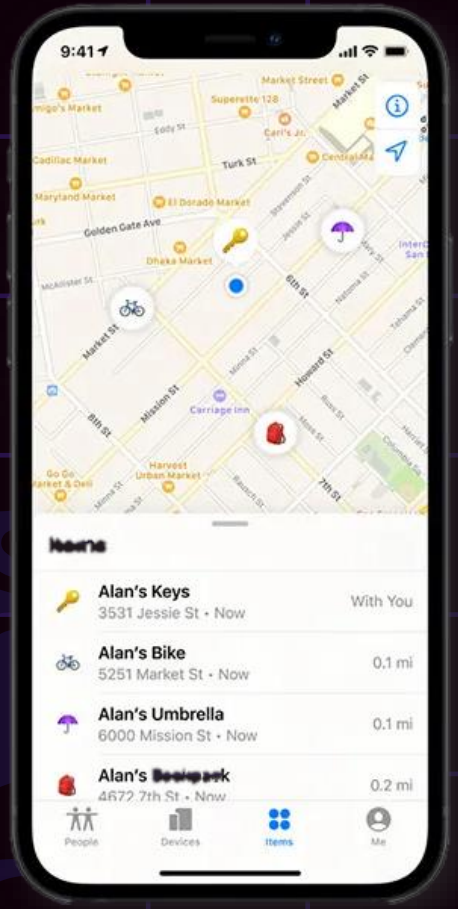
How?

- Protocol description(s)
- Standards and publications

And now?

- Radio sniffing
- Software reverse engineering

Active Bluetooth beacons: Emit locally, find globally



Actively powered beacons, unidirectional, emit local radio signals

Global network makes device position available to authorized owner

(no advertisement or trademark infringement intended. yada yada, etc. pp.)

Technological Convergence of BLE trackers

2011: Apple “Find My Friends”

2009: Apple “Find My iPhone”

2019: Apple “Find My”

GSM+GPS trackers

2024: Google “Find My Device”

2010: Bluetooth Low Energy (BLE)

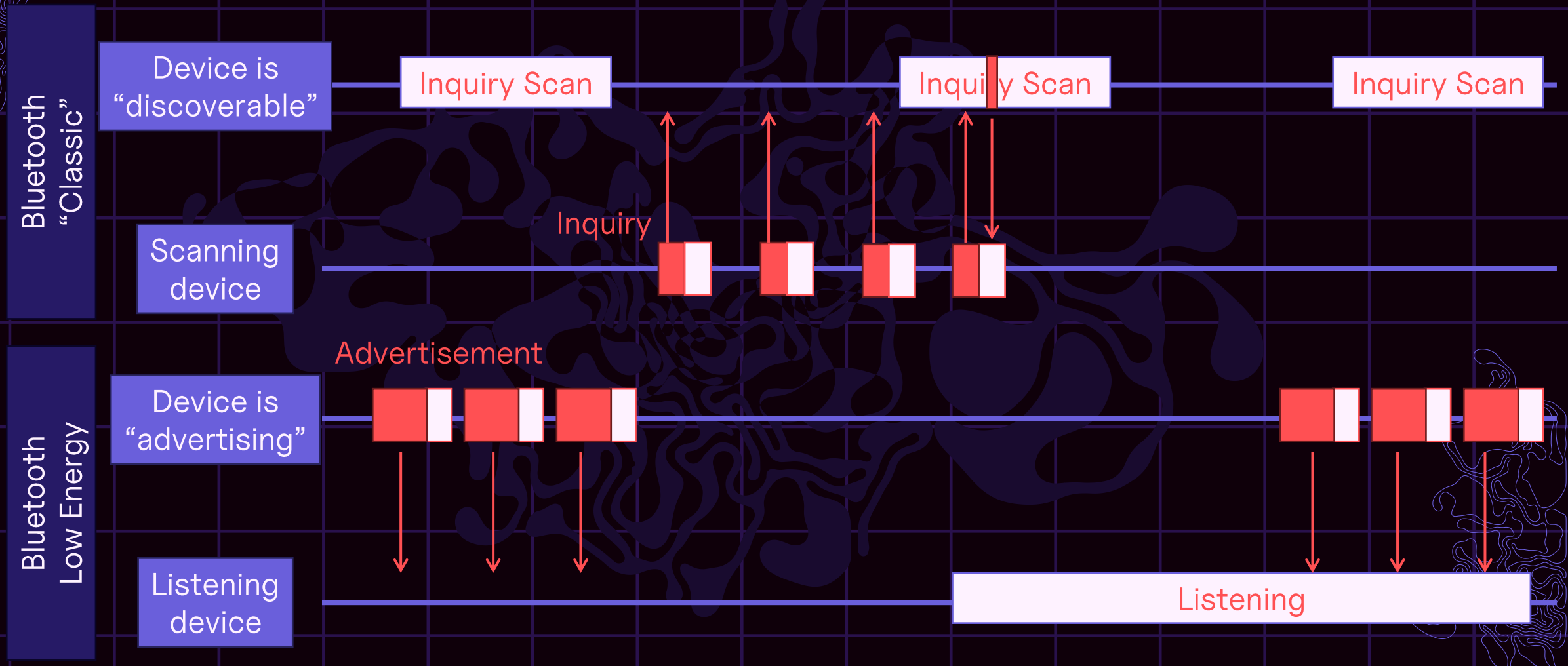
2015: Tile (“Community find”)

2013: Apple iBeacon

2013: Tile (last detected)

2015: Google Eddystone

Bluetooth: Discovery procedures



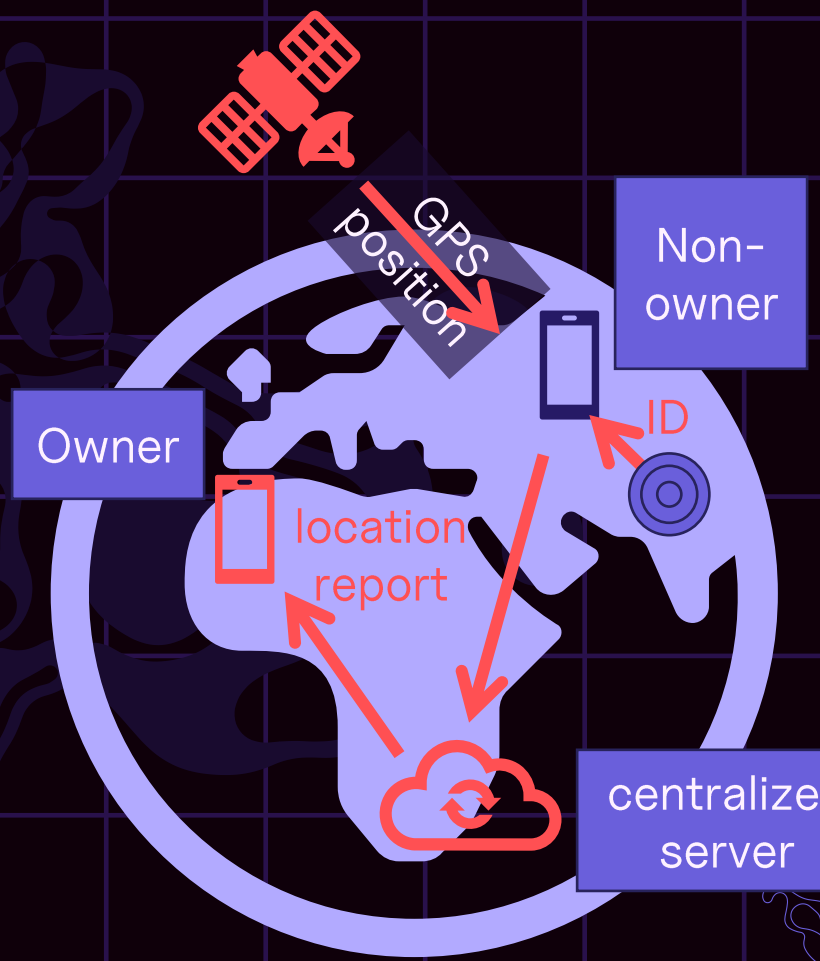
(Not to scale)

Construct a tracking network

BLE advertisements contain source MAC address and arbitrary advertisement information (up to 31 bytes)

Common use cases: iBeacon/Eddystone

Trivial to construct a centralized tracking network (e.g. Tile “Community find”)

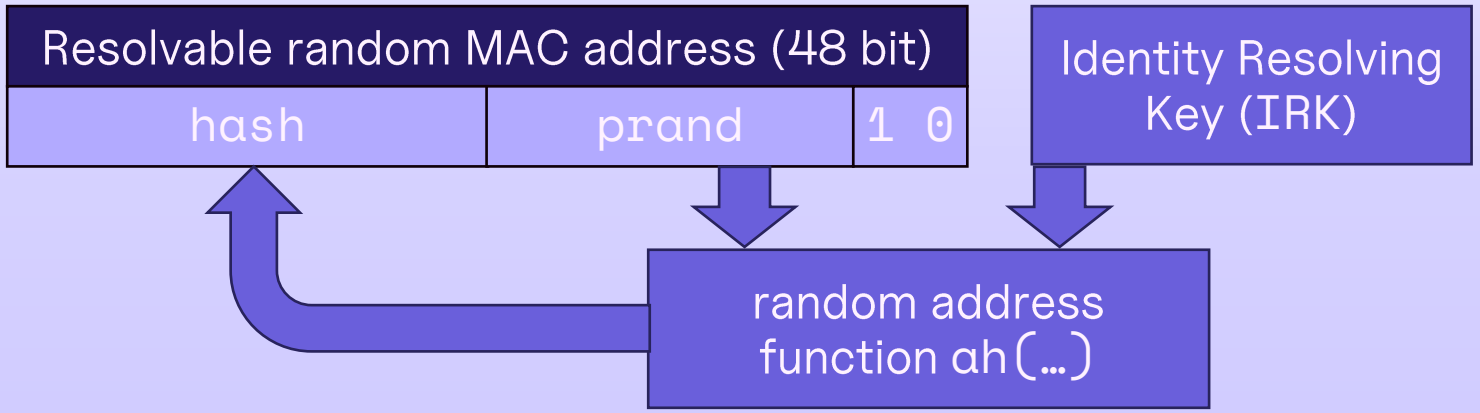


Trivial tracking, Attacker model

Privacy issue: globally unique MAC address can be
- recognized across sightings
- correlated across receivers

Partial solution: Random MAC addresses, optionally “resolvable”.

Resolvable Private Address generation



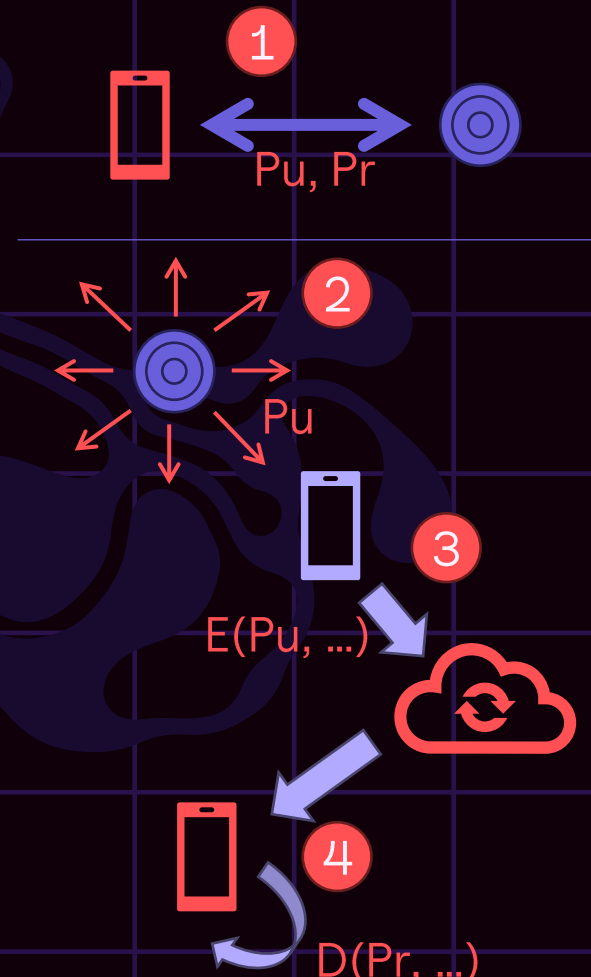
Attacker model
- Assume the attacker has one or more radios capable of receiving BLE advertisements
- Attacker radios can cooperate, possibly globally

Attacker goals
- Track device (= person?) from location to location
- Observe dwell time at a static location

Note: Trivially applies to centralized server operator

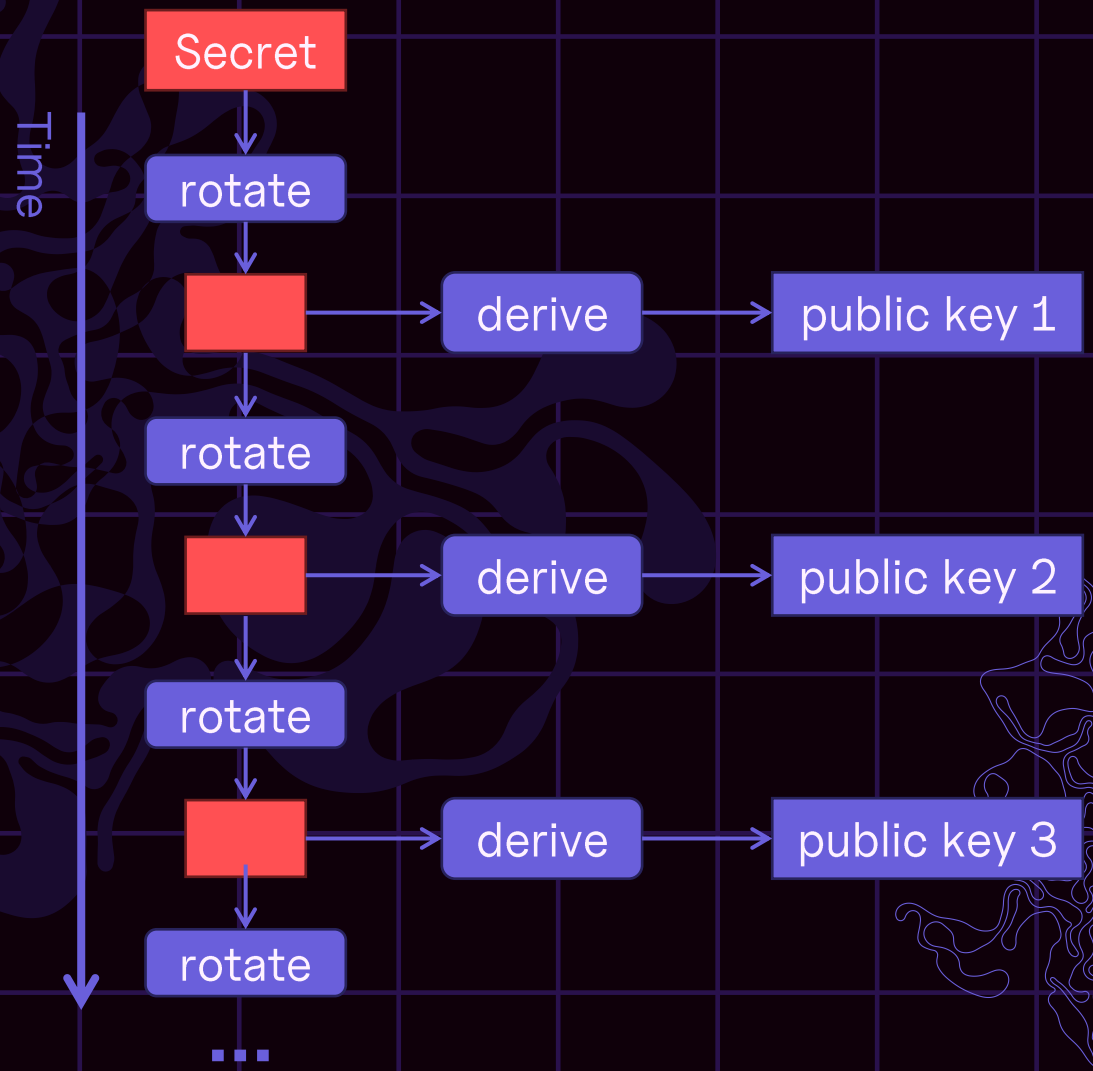
Core idea: Encrypt location reports

1. On accessory initialization:
   Setup keys between owner device and accessory
2. In BLE advertisement:
   Broadcast public key
3. Non-owner devices:
   Encrypt location with public key and upload to server
4. Owner device:
   query for location reports, download and decrypt



Core idea: Rotate public key

- A static broadcast public key would, in effect, be a static identifier
- Countermeasure: Accessory changes public key in pre-defined intervals
- Owner device can reproduce same sequence
- Non-owner devices are oblivious this is going on



“Two of every kind”

Apple/iOS

- “Find My” network (FMN)

Google/Android

- “Find My Device” network (FMDN)

Also ran

- Tile
- Samsung

Apple: “Find My”

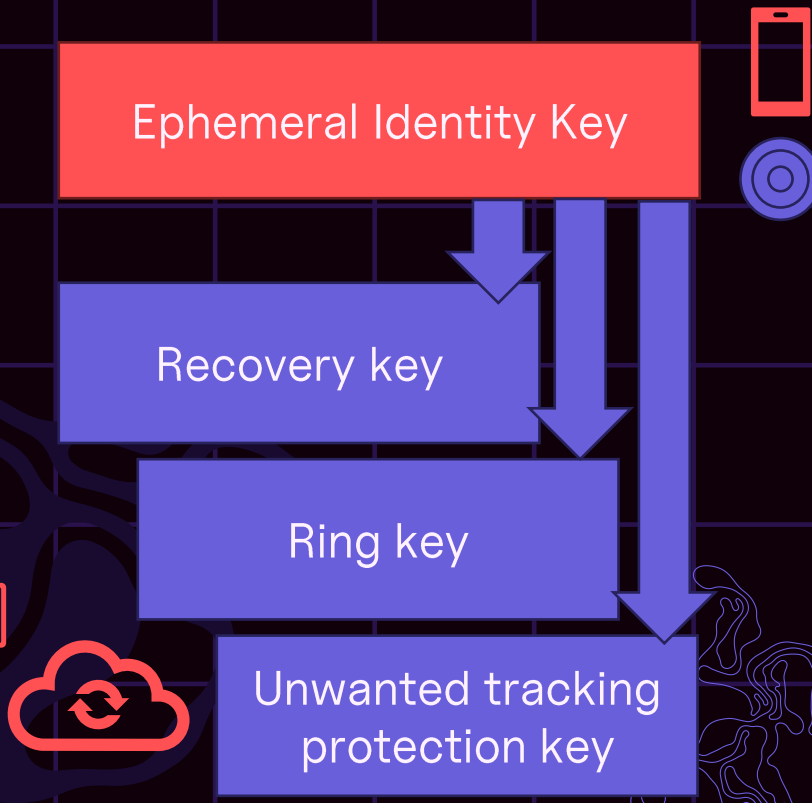
- NIST P-244 curve, 28 byte public key
- Use “random” source MAC address, for parts of the public key

Bytes	Content
0-5	BLE address `((pi[0] | (0b11 << 6)) || pi[1..5])`
6	Payload length (30)
7	Advertisement type (0xFF, manufacturer specific)
8-9	Company ID (0x004C)
10	OF type (0x10)
11	OF data length (25)
12	Status (e.g. battery level)
13-34	Public key bytes `pi[6..27]`
35	Public key bits `pi[0] >> 6`
36	Hint

- Alexander Heinrich, Milan Stute, Tim Kornhuber, Matthias Hollick. **Who Can *Find My* Devices? Security and Privacy of Apple's Crowd-Sourced Bluetooth Location Tracking System.** *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2021. doi:10.2478/popets-2021-0045
- Alexander Heinrich, Milan Stute, **OpenHaystack**, https://github.com/seemoo-lab/openhaystack

Google: “Find My Device”

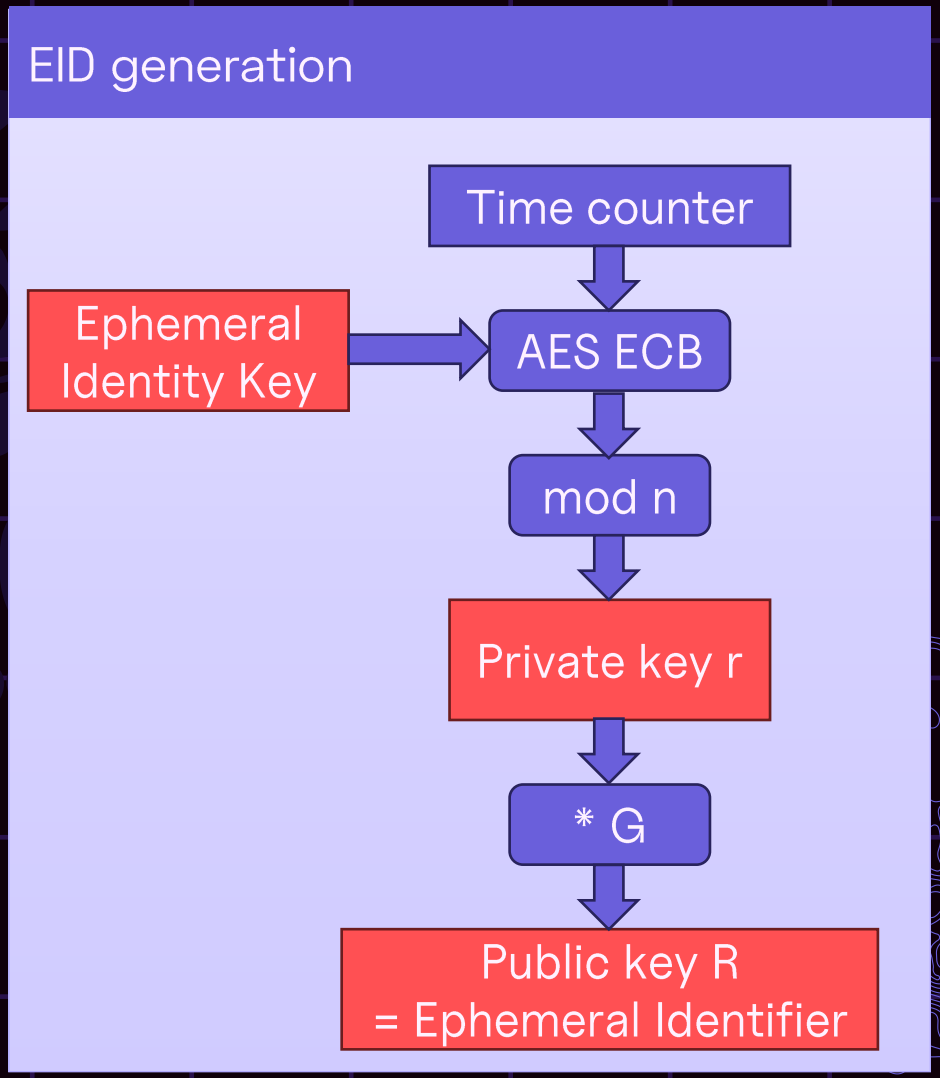
- Extension to Google Fast Pair Service (GFPS)
- Secret key `EIK` (Ephemeral Identity Key) chosen by owner phone
- Derived keys stored on backend:
  - Recovery key: `SHA256(EIK || 0x01)[:8]`
  - Ring key: `SHA256(EIK || 0x02)[:8]`
  - Unwanted tracking protection key: `SHA256(EIK || 0x03)[:8]`



Find My Device Network Accessory Specification, https://developers.google.com/nearby/fast-pair/specifications/extensions/fmdn

“Find My Device” advertisements

- Curve: SECP160R1 (20 bytes), or SECP256R1 (32 bytes, BLE 5 extended advertising)
- Extension of Eddystone format, UUID 0xFEAA
  - Eddystone: Types `0x00`, `0x10`, `0x20`, `0x30`
  - Find My Device type: `0x40` , or `0x41` (unwanted tracking protection)
- 20 bytes ephemeral ID == public key, 1 byte hashed flags



Abusing BLE trackers for stalking

- Low cost and high accessibility mean increased potential for non-consensual abuse of BLE trackers
- Reaction: Apple/Google trackers monitor connection to owner device and initiate countermeasures when away for too long
- Countermeasures:
  - Emit sound periodically, or on movement
  - Reduce identifier rotation frequency
  - Allow easier detection and pinpointing of tracker

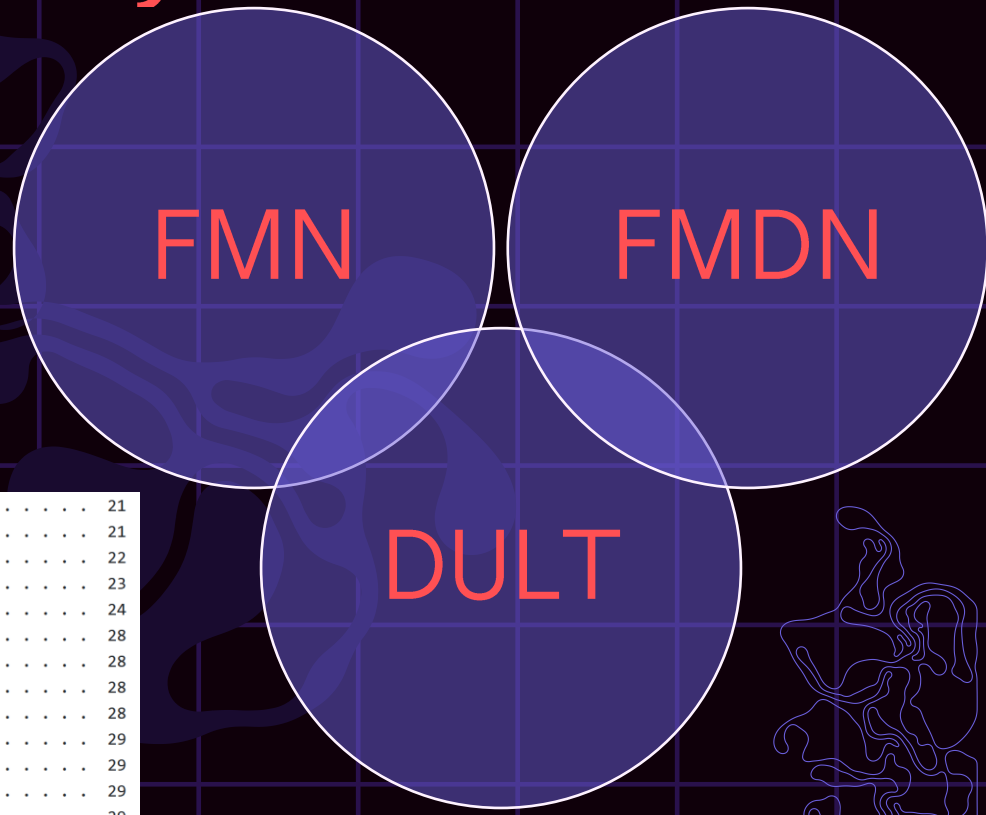
See also:

- **Escaping Big Brother (or Your Ex)**, erlern, day 2

media.ccc.de

“Detecting Unwanted Location Trackers” (DULT)

- IETF WG specifies threat model and behavior for tracker (-like) devices
- Command for “non-owner” devices are available unauthenticated, if device is not in “near-owner” state



Detecting Unwanted Location Trackers Accessory Protocol , https://datatracker.ietf.org/doc/draft-ietf-dult-accessory-protocol/

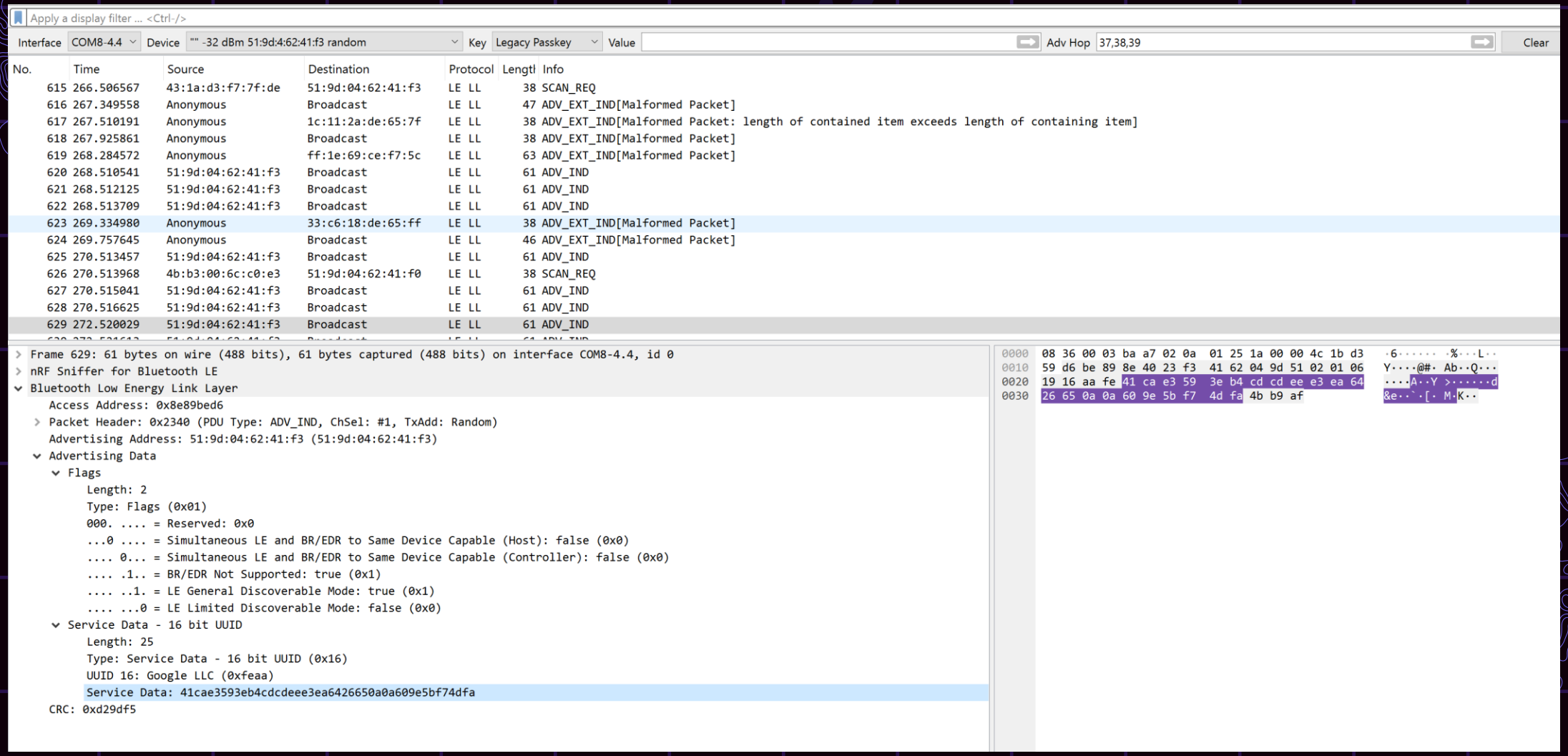
SDR sniffing with rad10

```
^Cfelix@flx-vivobook-asuslaptop-x415ja-r465ja:~/BTLE/host/build$ ./btle-tools/src/btle_rx  | grep --line-buffered aafe4
0032000us Pkt002 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569f9ee05 CRC0
0032977us Pkt030 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:519d046241f3 Data:0201061916aafe414d65d3e96bd71a813bff14d5c7ae971df328811286 CRC0
0131169us Pkt042 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569e9ee01 CRC1
0098087us Pkt067 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:519d046241f3 Data:0201061916aafe414d65d3e96bd71a813bff14d5c7ae971df328811286 CRC0
0033610us Pkt086 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569f9ee05 CRC0
0032613us Pkt139 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94549f9ee05 CRC1
0556880us Pkt158 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:519d046241f3 Data:0201061916aafe414d65d3e96bd71a813bff14d5c7ae971df328811286 CRC0
0000784us Pkt203 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569f9eecd CRC1
0032437us Pkt223 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79ceb8ce92d3c22d68ac8d94569f9ee05 CRC1
0098258us Pkt290 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:519d046241f3 Data:0201061916aafe414d65d3e96bd75a813bff14d5c7ae971dd328811286 CRC1
0163826us Pkt304 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8d7126055c Data:0201061916aafe400e7ed99eb79eeb8de92d1c22d68ac8d94569f9ee05 CRC1
0262727us Pkt391 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569f9ee05 CRC0
0032244us Pkt419 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569f9ee05 CRC1
0196449us Pkt444 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:519d046241f3 Data:0201061916aafe414d65d3e96bd71a813bff14d5c7ae971df328811286 CRC0
0066313us Pkt518 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569f9ee05 CRC0
0130473us Pkt558 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569b9276a CRC1
0032711us Pkt584 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569f9ee05 CRC0
0032732us Pkt596 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569f9ee05 CRC0
0000328us Pkt660 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8cc92d1c22d68ac8d96469f9ee05 CRC1
0294961us Pkt675 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:519d046241f3 Data:0201061916aafe414d65d3e96bd71a813bff14d5c7ae971df328811286 CRC0
0032748us Pkt698 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569f9ee05 CRC0
0032528us Pkt709 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:519d046241f3 Data:0201061916aafe414d65d3e96bd71a813bff14d5c7ae971df328811286 CRC0
0032771us Pkt720 Ch37 AA:8e89bed6 ADV_PDU_t0:ADV_IND T1 R0 PloadL35 AdvA:5b8df126055c Data:0201061916aafe400e7ed99eb79eeb8ce92d1c22d68ac8d94569f9ee05 CRC0
```

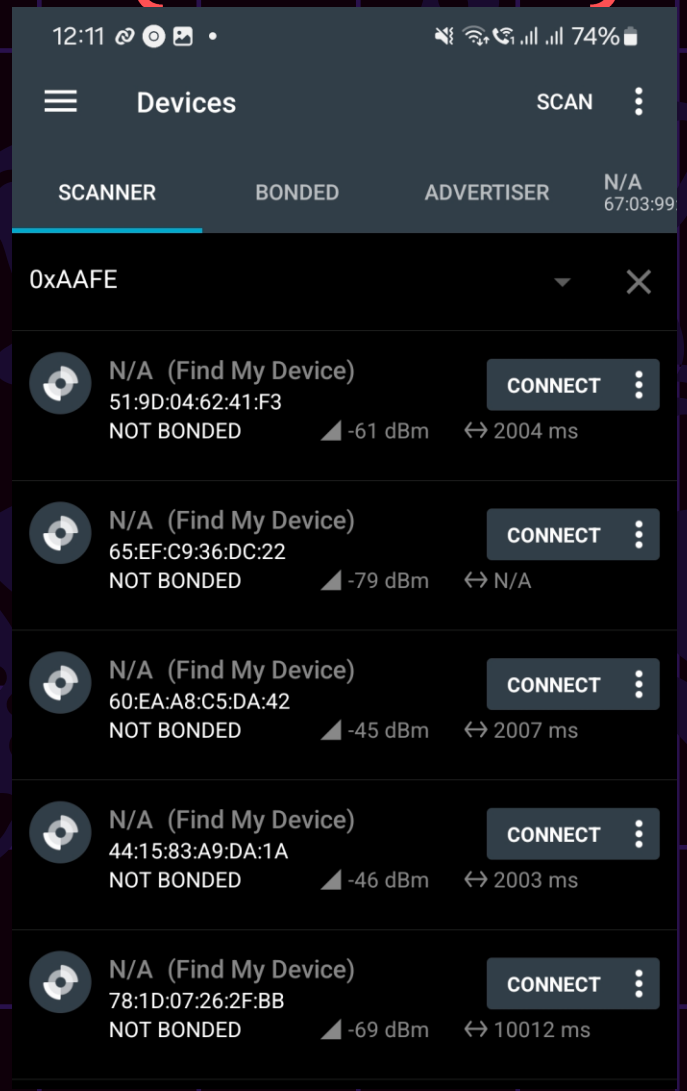
https://github.com/JiaoXianjun/BTLE

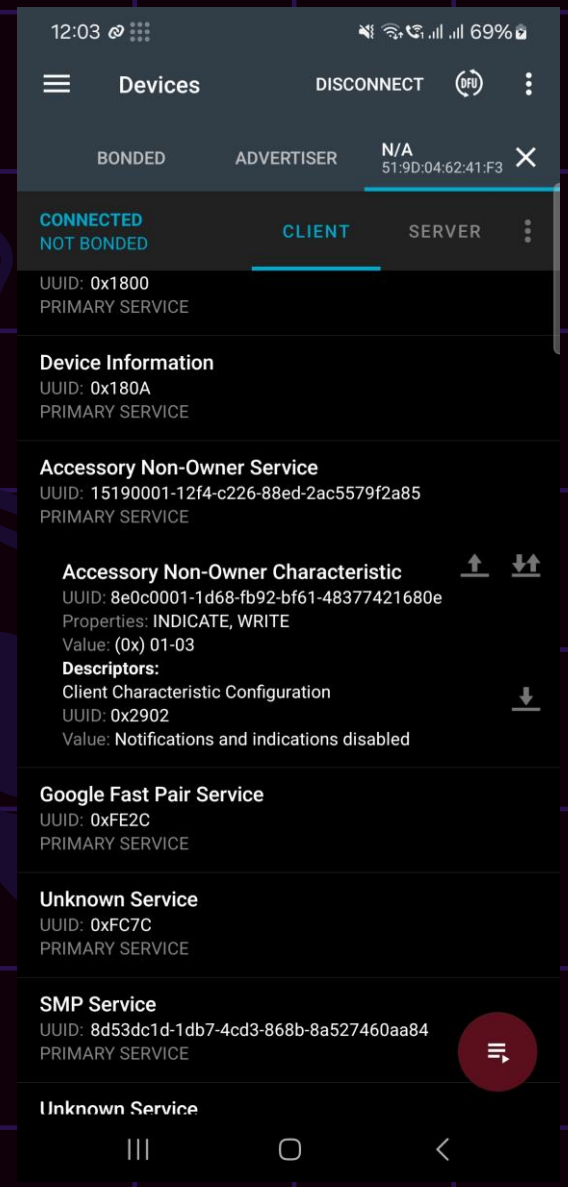
BTLE sniffer on bladeRF or HackRF, such as rad1o (the CCCamp 2015 badge)

Nordic Sniffer: nRF52 SDK



nRF Connect (Android)





Android “Find My Device” app

METHOD: POST +

URL

+ https://android.googleapis.com/nova/nbe_list_devices

HEADERS

+ accept-language: en-DE
+ authorization: Bearer ya29.m.CoYCAQ1IaZhgd1oXk2vBMGarYXZxgswqpNDaOxIuSBoO--2-51pJYIaOEWa_5BM1ekI06mel_Bph4NEeKZ4Eltauf6CL1Ns5YDXSn-

- Handled by com.google.android.gms and fmd
- HTTPS public key pinning

2 kB Protobuf RESPONSE BODY


```
247 "1": {
248   "1": "670d244d-0000-219e-b338-30fd381716e4"
249 },
250 "2": {
251   "1": "moto tag15",
252   "2": 22,
253   "9": {
254     "1": "https://lh3.googleusercontent.com/f-yUCKet_o8g0Izb5iscX4P03SoVtMFzTjzryxKuJuAUKkccGrnjrgI1qVyGCdClho0-loDcPLMQvZRJwE3c",
255     "2": 1
256   },
257 "12": {
```

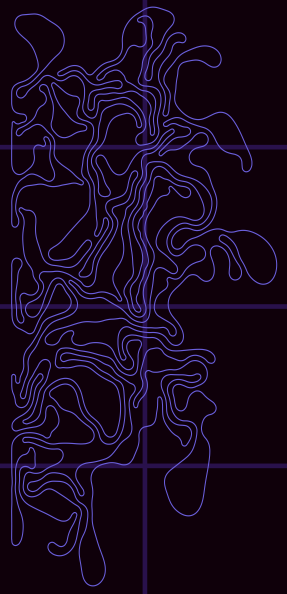
Call for Reverse Engineering

- Research question: Communication between A H E of own Find My Device accessories
- Is there anything in the lower layer radio protocol that might compromise anonymity?

https://github.com/leonboe1/GoogleFindMyTools
by Leon Böttger, December 2024

User Story

As a
Me
I want to
be able to derive the broadcast keys of my own devices
So that I
can have my Home Assistant detect whether the devices are at home



ILLEGAL 3863 INSTRUCTIONS

The End

Signal henryk.42

threema://36C48UCS

https://chaos.social/@henryk

See also:

- **From fault injection to RCE: Analyzing a Bluetooth tracker**, Nicolas Oberli, day 1

media.ccc.de