A review of the BIAS and KNOB attacks on Bluetooth Classic and Bluetooth Low Energy

Daniele Antonioli
Who Am I

- Daniele Antonioli
  - Postdoc at EPFL
  - I like cyber-physical and wireless systems, protocol analysis, applied crypto, ...
  - Twitter: @francozappa
  - Website: https://francozappa.github.io

- I work in the HexHive group led by Mathias Payer
  - System security e.g., Bluetooth security and DP3T
  - More: https://hexhive.epfl.ch/
BIAS and KNOB attacks on Bluetooth

- Key Negotiation Of Bluetooth (KNOB) Attack
  - Exploits Bluetooth’s key negotiation

- Bluetooth Impersonation AttackS (BIAS)
  - Exploits Bluetooth’s key authentication
  - CVE-2020-10135: https://kb.cert.org/vuls/id/647177/

- KNOB and BIAS attacks are standard-compliant
  - Billions of vulnerable devices
  - E.g. smartphones, laptops, tablets, headsets, cars, ...
Talk Outline

• Talks has three parts
  ▶ Part 1: Introduction about Bluetooth and its security mechanisms
  ▶ Part 2: High level description of the BIAS and KNOB attacks
  ▶ Part 3: Attacks’ implementation, evaluation and countermeasures

• Related work by Nils Tippenhauer, Kasper Rasmussen, and myself
  ▶ “The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR” [SEC19]
  ▶ “Key Negotiation Downgrade Attacks on Bluetooth and Bluetooth Low Energy” [TOPS20]
  ▶ “BIAS: Bluetooth Impersonation AttackS” [S&P20]
Part 1: Introduction about Bluetooth
Bluetooth Classic and Bluetooth Low Energy

• Bluetooth
  ▶ Pervasive wireless communication technology

• Bluetooth Classic (BT)
  ▶ High-throughput services
  ▶ E.g., audio, voice

• Bluetooth Low Energy (BLE)
  ▶ Very low-power services
  ▶ E.g., wearables, contact tracing
Bluetooth Standard

- Bluetooth Standard
  - Complex documents (Bluetooth Core v5.2, 3.256 pages)
  - Custom security mechanisms (pairing, secure sessions)
  - No public reference implementation

https://www.bluetooth.com/specifications/bluetooth-core-specification/
Bluetooth Security: Pairing and Secure Sessions

Alice
slave

Bob
master
Bluetooth Security: Pairing and Secure Sessions

Alice slave

Pairing

Bob master

$K_L$
Bluetooth Security: Pairing and Secure Sessions

$K_L$

Alice slave

Session Establishment

Bob master

$K_L$
Bluetooth Security: Pairing and Secure Sessions

Alice slave

Bob master

$K_L$

Pairing key

Authentication
Bluetooth Security: Pairing and Secure Sessions

Alice slave

Bob master

Session key

Negotiation

$K_L$
$K'_C$

$K_L$
$K'_C$
Bluetooth Security: Pairing and Secure Sessions

Alice slave $\rightarrow$ Secure session $\rightarrow$ Bob master

$K_L$ $\rightarrow$ Alice slave $\rightarrow$ $K'_C$

$K_L$ $\rightarrow$ Bob master $\rightarrow$ $K'_C$
Bluetooth Security: Impersonation and MitM

Charlie as Alice

NO secure session

Bob master

$K_L$

$K'_C$
Bluetooth Security: Impersonation and MitM

$K_L$  
$K_C'$

Alice
slave

NO secure session

Charlie
as Bob
Bluetooth Security: Impersonation and MitM

$K_L$  
$K'_C$

Alice slave

NO MitM

$K_L$  
$K'_C$

Bob master
Part 2: KNOB Attack on BLE
BLE Pairing

Alice (master) A

Bob (slave) B

Phase 1: Feature exchange (including key negotiation)

Phase 2: Key establishment and optional authentication

Phase 3: Key distribution (over encrypted link)
Issues with BLE Pairing (Key Negotiation)

- Issues
  - KeySize negotiation is **not protected**, i.e. no integrity, no encryption
  - KeySize values (pairing key strength) between **7 bytes** and 16 bytes
KNOB Attack on BLE

- **KNOB attack on BLE**
  - Downgrade BLE pairing key to 7 bytes of entropy
  - Session keys will inherit 7 bytes of entropy
  - Brute-force the session key and break BLE security
Part 2: BIAS Attack on BT
BIAS Attacks Introduction

- BIAS attacks target BT secure session establishment
  - Not pairing

- Assumptions for Alice and Bob
  - Securely paired in absence of Charlie
  - Share a strong pairing key (e.g. 16 bytes of entropy)
Bluetooth Authentication Mechanisms

• Legacy Secure Connection (LSC) authentication
  ▶ Unilateral, challenge-response

• Secure Connection (SC) authentication
  ▶ Mutual, challenge-response

• LSC or SC negotiated during secure session establishment
<table>
<thead>
<tr>
<th>BIAS Attacks</th>
<th>Master Impersonation</th>
<th>Slave Impersonation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy Secure Connections</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secure Connections</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
BIAS Attacks on Bluetooth Session Establishment

<table>
<thead>
<tr>
<th>BIAS Attacks</th>
<th>Master Impersonation</th>
<th>Slave Impersonation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Legacy Secure Connections</td>
<td>Alice slave</td>
<td>Charlie as Bob</td>
</tr>
<tr>
<td>Secure Connections</td>
<td>BIAS</td>
<td>Bob master</td>
</tr>
</tbody>
</table>

Daniele Antonioli (@francozappa)
# BIAS Attacks on Bluetooth Session Establishment

<table>
<thead>
<tr>
<th>BIAS Attacks</th>
<th>Master Impersonation</th>
<th>Slave Impersonation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Legacy Secure Connections</strong></td>
<td>Alice slave ← BIAS → Charlie as Bob</td>
<td>Charlie as Alice ← BIAS → Bob master</td>
</tr>
<tr>
<td><strong>Secure Connections</strong></td>
<td>Alice slave ← BIAS → Charlie as Bob</td>
<td>Charlie as Alice ← BIAS → Bob master</td>
</tr>
</tbody>
</table>
Legacy Secure Connection (LSC) Authentication

\[ R_A = H(C_B, A, K_L) \]
Issues with LSC Authentication

- LSC authentication is **not used mutually** for session establishment
- A device can **switch authentication role**

![Diagram showing the authentication process involving Alice (slave) and Bob (master).]

\[ R_A = H(C_B, A, K_L) \]

\[ R_A \text{ check} \]
BIAS Attack on LSC: Master Impersonation

Alice (slave)  Charlie as Bob (master)

A  C

B, LSC  A, LSC

C

R_A = H(C_C, A, K_L)

Skip R_A check
BIAS Attack on LSC: Slave Impersonation

Charlie as Alice (slave)

Bob (master)

B, LSC

A, Role Switch, LSC

Accept Role Switch

Charlie is the master (verifier)

\[ C_C \]

\[ R_B = H(C_C, B, K_L) \]

Skip \( R_B \) check
Secure Connections (SC) Authentication

Alice (slave)

A

Bob (master)

B

B, SC

A, SC

C_B

C_A

R_B, R_A = H(C_B, C_A, B, A, K_L)

R_B check

R_A check

R_B, R_A = H(C_B, C_A, B, A, K_L)

R_A

R_B
Issues with SC Authentication

- SC negotiation **is not integrity-protected**
- SC support is **not enforced** for pairing and session establishment

\[
R_B, R_A = H(C_B, C_A, B, A, K_L)
\]

Alice (slave) \(A\) \hspace{2cm} Bob (master) \(B\)

- Alice (slave) sends \(A, SC\) to Bob (master)
- Bob (master) sends \(B, SC\) to Alice (slave)
- \(C_B\) and \(C_A\) are exchanged

\[
R_B, R_A = H(C_B, C_A, B, A, K_L)
\]

- Alice (slave) sends \(R_B\) to Bob (master)
- Bob (master) sends \(R_A\) to Alice (slave)

**RB check** \hspace{2cm} **RA check**
BIAS Attack on SC: Master Impersonation

Alice (slave)

\[ A \]

\[ \text{SC downgraded to LSC} \]

\[ \text{BIAS master impersonation on LSC} \]

Charlie as Bob (master)

\[ C \]

\[ \text{A, SC} \]

\[ \text{B, LSC} \]
BIAS Attack on SC: Slave Impersonation

Charlie as Alice (slave)  Bob (master)
C  B

B, SC  A, LSC

SC downgraded to LSC

BIAS slave impersonation on LSC
Part 2: KNOB Attack on BT
BT Session Establishment: Overview

Alice (master)  
A

Bob (slave)  
B

Phase 1: Pairing key authentication

Phase 2: Session key negotiation

Phase 3: Start encryption
BT Session Establishment: Session Key Negotiation

Alice (master)

A

Bob (slave)

B

Phase 2: Session key negotiation

Key entropy: 16

Key entropy: 15

Accept

• Issues
  ▶ Key entropy negotiation is not protected, i.e. no integrity, no encryption
  ▶ Key entropy values between 1 byte and 16 bytes
• KNOB attack on BT
  ▶ Downgrade BT session key entropy to 1 bytes
  ▶ Brute-force the session key and break BT security
Part 3: BIAS + KNOB
BIAS + KNOB: Break Bluetooth Session Establishment

Alice (master)

Bob (slave)

Phase 1: pairing key authentication

Phase 2: session key negotiation

Phase 3: secure session
BIAS + KNOB: Break Bluetooth Session Establishment

Alice (master)

A

Charlie as Bob (slave)

B

Phase 1: pairing key authentication (BIAS attack)

Phase 2: session key negotiation

Phase 3: secure session
BIAS + KNOB: Break Bluetooth Session Establishment

Phase 1: pairing key authentication (BIAS attack)
Phase 2: session key negotiation (KNOB attack [SEC19])
Phase 3: secure session
BIAS + KNOB: Break Bluetooth Session Establishment

Phase 1: pairing key authentication (BIAS attack)

Phase 2: session key negotiation (KNOB attack [SEC19])

Phase 3: secure session (Charlie is Bob)
BIAS + KNOB: Break Bluetooth Session Establishment

Charlie as Alice (master)  
Bob (slave)

Phase 1: pairing key authentication (BIAS attack)
Phase 2: session key negotiation (KNOB attack [SEC19])
Phase 3: secure session (Charlie is Alice)

Daniele Antonioli (@francozappa)
Part 3: Implementation
Host, Controller, and Host Controller Interface (HCI)

```
+--------+     +------------------+
| BlueZ  |     | Bluetooth baseband |
|        |     | Bluetooth firmware |
|        |     |                    |
+--------+     +------------------+

Linux Kernel

Host

HCI

Controller
```

Daniele Antonioli (@francozappa)
Implementation of KNOB Attack on BLE

- Security Manager Protocol (SMP) manipulation
  - Implemented in the BLE host (OS)

- Custom Linux kernel
  - net/bluetooth/smp.c: SMP_DEV(hdev)->max_key_size = 7
  - See https://github.com/francozappa/knob/tree/master/ble

- Custom user-space BLE stack
  - Based on PyBT (https://github.com/mikeryan/PyBT)
  - That is based on scapy (https://scapy.net)
Implementation of BIAS Attacks on BT

Linux Laptop

https://github.com/francozappa/bias

https://github.com/seemoo-lab/internalblue

CYW920819
Implementation of BIAS Attacks on BT

IF
AF

BIAS
bias.py
InternalBlue
Linux Laptop

USB

Bluetooth baseband
Bluetooth firmware
CYW920819

https://github.com/francozappa/bias
https://github.com/seemoo-lab/internalblue
Implementation of KNOB Attack on BT

https://github.com/francozappa/knob
https://github.com/seemoo-lab/internalblue
Patch for the KNOB Attack on BT

#!/usr/bin/python2
addr_Lmin = "0x20118a"  # addr RE from firmware
addr_Lmax = "0x20118b"  # addr RE from firmware
internalblue.writeMem(addr_Lmin, "\0x01")  # 1 byte of entropy
internalblue.writeMem(addr_Lmax, "\0x01")  # 1 byte of entropy
Part 3: Evaluation
### Evaluation: KNOB on BLE (19 devices, from 2019)

<table>
<thead>
<tr>
<th>Device</th>
<th>OS (BLE Host)</th>
<th>Role</th>
<th>LTK Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BLE Secure Connections (Bluetooth ≥ 4.2)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Garmin Vivoactive 3</td>
<td>Proprietary</td>
<td>Peripheral</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Google Pixel 2</td>
<td>Android</td>
<td>Central</td>
<td>7 bytes</td>
</tr>
<tr>
<td>LG K40</td>
<td>Android</td>
<td>Central</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Samsung Gear S3</td>
<td>Tizen OS</td>
<td>Peripheral</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Thinkpad X1 3rd</td>
<td>Linux</td>
<td>Central</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Thinkpad X1 6rd</td>
<td>Linux</td>
<td>Central</td>
<td>7 bytes</td>
</tr>
<tr>
<td>TI CC1352R</td>
<td>TI RTOS</td>
<td>Central</td>
<td>7 bytes</td>
</tr>
<tr>
<td><strong>BLE legacy security (Bluetooth 4.0 and 4.1)</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comet Blue thermostat</td>
<td>Unknown</td>
<td>Peripheral</td>
<td>7 bytes</td>
</tr>
<tr>
<td>EDIFIER R1280DB speaker</td>
<td>Unknown</td>
<td>Peripheral</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Fitbit Charge 2</td>
<td>Fitbit OS</td>
<td>Peripheral</td>
<td>7 bytes</td>
</tr>
<tr>
<td>ID115 HR Plus</td>
<td>Unknown</td>
<td>Peripheral</td>
<td>7 bytes</td>
</tr>
<tr>
<td>LG Nexus 5</td>
<td>Android</td>
<td>Central</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Logitech MX Anywhere 2S</td>
<td>Nordic</td>
<td>Peripheral</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Motorola G3</td>
<td>Android</td>
<td>Central</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Samsung Galaxy J5</td>
<td>Android</td>
<td>Central</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Samsung TV UE48J6250</td>
<td>Tizen OS</td>
<td>Peripheral</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Xiaomi Mi band</td>
<td>Proprietary</td>
<td>Peripheral</td>
<td>7 bytes</td>
</tr>
<tr>
<td>Xiaomi Mi band 2 (x2)</td>
<td>Proprietary</td>
<td>Peripheral</td>
<td>7 bytes</td>
</tr>
</tbody>
</table>
# Evaluation: BIAS on BT (31 devices, from 2020)

<table>
<thead>
<tr>
<th>Chip</th>
<th>Device(s)</th>
<th>LSC MI</th>
<th>LSC SI</th>
<th>SC MI</th>
<th>SC SI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bluetooth v5.0</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apple 339S00397</td>
<td>iPhone 8</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>CYW20819</td>
<td>CYW920819EVB-02</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Intel 9560</td>
<td>ThinkPad L390</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Snapdragon 630</td>
<td>Nokia 7</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Snapdragon 636</td>
<td>Nokia X6</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Snapdragon 835</td>
<td>Pixel 2</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Snapdragon 845</td>
<td>Pixel 3, OnePlus 6</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td><strong>Bluetooth v4.2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apple 339S00056</td>
<td>MacBookPro 2017</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Apple 339S00199</td>
<td>iPhone 7 Plus</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Apple 339S00448</td>
<td>iPad 2018</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>CSR 11393</td>
<td>Sennheiser PXC 550</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Exynos 7570</td>
<td>Galaxy J3 2017</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Intel 7265</td>
<td>ThinkPad X1 3rd</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Intel 8260</td>
<td>HP ProBook 430 G3</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Daniele Antonioli (@francozappa) - A review of the BIAS and KNOB attacks on Bluetooth Classic and Bluetooth Low Energy
Evaluation: BIAS on BT (31 devices, from 2020)

<table>
<thead>
<tr>
<th>Chip</th>
<th>Device(s)</th>
<th>LSC MI</th>
<th>LSC SI</th>
<th>SC MI</th>
<th>SC SI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bluetooth v4.1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CYW4334</td>
<td>iPhone 5s</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CYW4339</td>
<td>Nexus 5, iPhone 6</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CYW43438</td>
<td>RPi 3B+</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Snapdragon 210</td>
<td>LG K4</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Snapdragon 410</td>
<td>Motorola G3, Galaxy J5</td>
<td>●</td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td><strong>Bluetooth v≤ 4.0</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BCM20730</td>
<td>ThinkPad 41U5008</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>BCM4329B1</td>
<td>iPad MC349LL</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CSR 6530</td>
<td>PLT BB903+</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CSR 8648</td>
<td>Philips SHB7250</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Exynos 3470</td>
<td>Galaxy S5 mini</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Exynos 3475</td>
<td>Galaxy J3 2016</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Intel 1280</td>
<td>Lenovo U430</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Intel 6205</td>
<td>ThinkPad X230</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Snapdragon 200</td>
<td>Lumia 530</td>
<td>●</td>
<td>●</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
## Evaluation: KNOB on BT (38 devices, from 2019)

<table>
<thead>
<tr>
<th>Chip</th>
<th>Device(s)</th>
<th>$K'_C$ Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bluetooth version 5.0</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apple A1865</td>
<td>iPhone X</td>
<td>1 byte</td>
</tr>
<tr>
<td>Apple 339S00428</td>
<td>MacBookPro 2018</td>
<td>1 byte</td>
</tr>
<tr>
<td>Mediatek MT6762</td>
<td>LG K40</td>
<td>3 bytes</td>
</tr>
<tr>
<td>Snapdragon 660</td>
<td>Xiaomi MI A2</td>
<td>1 byte</td>
</tr>
<tr>
<td>Snapdragon 835</td>
<td>Pixel 2, OnePlus 5</td>
<td>1 byte</td>
</tr>
<tr>
<td>Snapdragon 845</td>
<td>Galaxy S9</td>
<td>1 byte</td>
</tr>
<tr>
<td><strong>Bluetooth version 4.2</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apple 339S00045</td>
<td>iPad Pro 2</td>
<td>1 byte</td>
</tr>
<tr>
<td>BCM43438</td>
<td>RPi 3B, RPi 3B+</td>
<td>1 byte</td>
</tr>
<tr>
<td>BCM43602</td>
<td>iMac MMQA2LL/A</td>
<td>1 byte</td>
</tr>
<tr>
<td>CSR 11393</td>
<td>Sennheiser PXC 550</td>
<td>1 byte</td>
</tr>
<tr>
<td>CSR 11836</td>
<td>Bose SoundLink revolve</td>
<td>1 byte</td>
</tr>
<tr>
<td>CSR 12942</td>
<td>Sony WH-100XM3</td>
<td>1 byte</td>
</tr>
<tr>
<td>Exynos 7570</td>
<td>Galaxy J3 2017</td>
<td>1 byte</td>
</tr>
<tr>
<td>Intel 7265</td>
<td>Thinkpad X1 3rd, Dell Latitude E7250</td>
<td>1 byte</td>
</tr>
<tr>
<td>Intel 8260</td>
<td>HP ProBook 430 G3</td>
<td>1 byte</td>
</tr>
<tr>
<td>Intel 8265</td>
<td>Thinkpad X1 6th</td>
<td>1 byte</td>
</tr>
<tr>
<td>Snapdragon 625</td>
<td>Xiaomi Mi Max 2</td>
<td>1 byte</td>
</tr>
</tbody>
</table>
### Evaluation: KNOB on BT (38 devices, from 2019)

**Bluetooth version 4.1**
- BCM4339 (CYW4339)  Nexus 5, iPhone 6  1 byte
- Snapdragon 210  LG K4  1 byte
- Snapdragon 410  Motorola G3, Galaxy J5  1 byte

**Bluetooth version ≤ 4.0**
- Apple W1  AirPods  7 bytes
- BCM20730  Thinkpad 41U5008  1 byte
- BCM4329B1  iPad MC349LL  1 byte
- Broadcom 8721  Anker A7721, Thinkpad KT-1255  1 byte
- Broadcom 20702  MacBookAir Mid 2012  1 byte
- CSR 6530  Plantronics BackBeat 903+  1 byte
- CSR 8648  Philips SHB7250+  1 byte
- Exynos 3475  Galaxy J3 2016  1 byte
- Intel Centrino 6205  Thinkpad X230  1 byte
- Snapdragon 200  Lumia 530  1 byte
- Snapdragon 615  Galaxy A7  1 byte
- Snapdragon 800  LG G2  1 byte
Part 3: Countermeasures
Counter KNOB Attacks on BT and BLE

• Legacy-compliant
  ▶ Set minimum entropy value to 16 bytes
  ▶ Enforce key entropy of 16 bytes

• Non legacy-compliant
  ▶ Integrity protect key negotiation
  ▶ Remove entropy negotiation feature
Bluetooth SIG amended the standard (2019-08-13)

- Erratum 11838: Encryption Key Size Updates
  - Mandatory only for recent Bluetooth versions: 4.2, 5.0, 5.1, 5.2
  - BT minimum entropy value now is 7 bytes, BLE stays the same

KNOB on BT: Apple mitigation

- Notify the user if key entropy is lower than 7 bytes
  - Accept any entropy value if user presses Allow (once)

- Shifting responsibilities to users is bad!
  - Users do not care, accidentally press, are tricked to press
KNOB on BT: Google and Linux mitigation

- OS patch
  - Checks entropy and terminates the session if entropy is less than 7 bytes
  - Uses HCI Read Encryption Key Size command

- Shifting responsibilities to the OS can still be bad!
  - Malicious OS can still negotiate 1 byte of entropy
Counter BIAS Attacks on BT

- Use LSC authentication mutually during session establishment
- Integrity-protect session establishment with the pairing key
- Enforce SC support across pairing and session establishment
BIAS: Bluetooth SIG and Vendors Response

- Bluetooth SIG

- Vendors
  - ???

- Bottom line
  - No concrete mitigations put in place
P3: Conclusion
KNOB and BIAS Attacks Recap

- KNOB attack on BLE
  - Compute BLE pairing key and all derived session keys

- BIAS attacks on BT
  - Establish BT secure sessions while impersonating any Bluetooth device

- KNOB attack on BT
  - Compute BT session keys

- KNOB + BIAS on BT
  - Break BT secure sessions while impersonating any Bluetooth device
Lessons Learned

• Choose wisely your standard-compliant security mechanism
  ▶ E.g. Is entropy negotiation really needed?
  ▶ E.g. Is unilateral authentication acceptable?

• Standard compliant attacks are very effective
  ▶ 1 vuln = billions of vulnerable devices

• Standard compliant attacks are difficult to patch
  ▶ Updating the standard != patching devices
Open Problems with Bluetooth Security

- BT and BLE allow to negotiate keys with very low entropy (e.g., 1 byte)

- BT and BLE entropy negotiations are not protected and do not provide any runtime benefit

- Most devices are still vulnerable to standard-compliant attacks (KNOB, BIAS, invalid curves, legacy pairing, BLESAn, NiNo, . . .)

- Bluetooth SIG has no bug-bounty program (good for black-hats, bad for white-hats)
This is it. Thanks for your attention!

- Related work (by Daniele Antonioli, Nils Tippenhauer, and Kasper Rasmussen)
  - BIAS: Bluetooth Impersonation AttackS [S&P20]
  - Key Negotiation Downgrade Attacks on Bluetooth and Bluetooth Low Energy [TOPS20]
  - The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR [SEC19]

- Try the attacks yourself!
  - https://github.com/francozappa/knob
  - https://github.com/francozappa/bias