Hardwear.io Virtual Con 2020

From the Bluetooth Standard to Standard-Compliant 0-days

Daniele Antonioli and Mathias Payer

EPFL

Who We Are

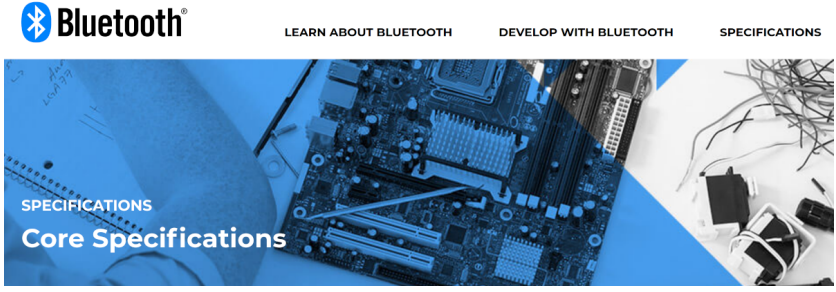
- Daniele Antonioli
  - Security researcher, Postdoc at EPFL
  - @francozappa
  - More: https://francozappa.github.io

- Mathias Payer
  - Security researcher, Professor at EPFL
  - @gannimo
  - More: https://nebelwelt.net/

- We are researchers in the HexHive group
  - System security topics
  - More: https://hexhive.epfl.ch/

Bluetooth Standard

- Bluetooth Standard
  - Complex document (Bluetooth Core v5.2, 3.256 pages)
  - Specifies *Bluetooth Classic (BT)* and *Bluetooth Low Energy (BLE)*



https://www.bluetooth.com/specifications/bluetooth-core-specification/

Standard-Compliant 0-days

- Standard-compliant 0-day (security vulnerability)
  - Unknown and/or unaddressed
  - Agnostic to hardware, and software implementation details
  - Very effective (1 vuln = all standard-compliant devices are exploitable)
  - Difficult to patch (firmware upgrades, device recall)

Key Negotiation of Bluetooth (KNOB) Attacks

- KNOB attacks on Bluetooth Low Energy (BLE) and Bluetooth Classic (BT)
  - Exploiting standard-compliant 0-days in Bluetooth **key negotiation**

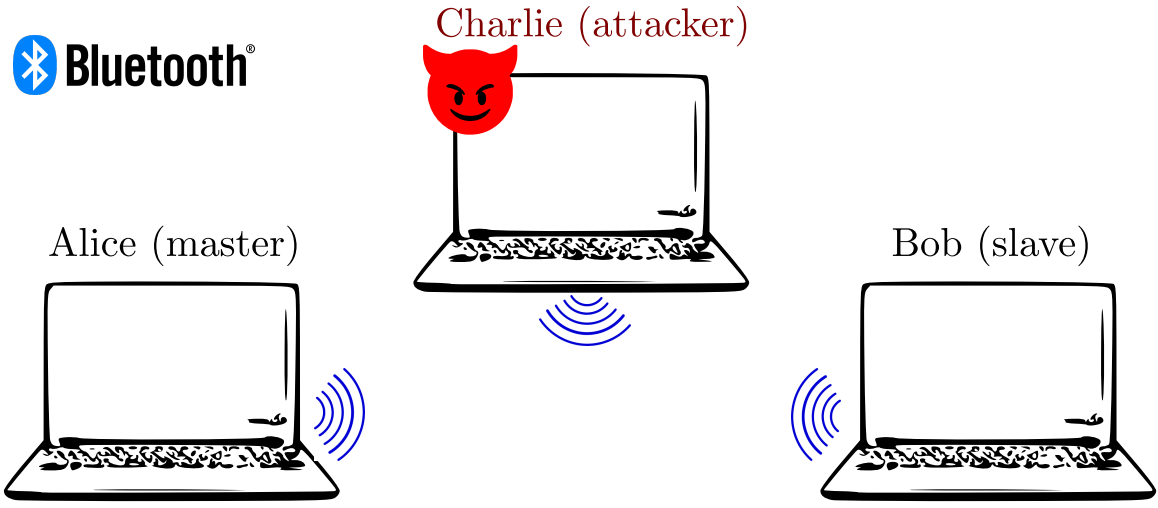
- Related work (cc: Nils Tippenhauer and Kasper Rasmussen)
  - "The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR" [SEC19]
  - "Key Negotiation Downgrade Attacks on Bluetooth and Bluetooth Low Energy" [TOPS20]

Bluetooth Security

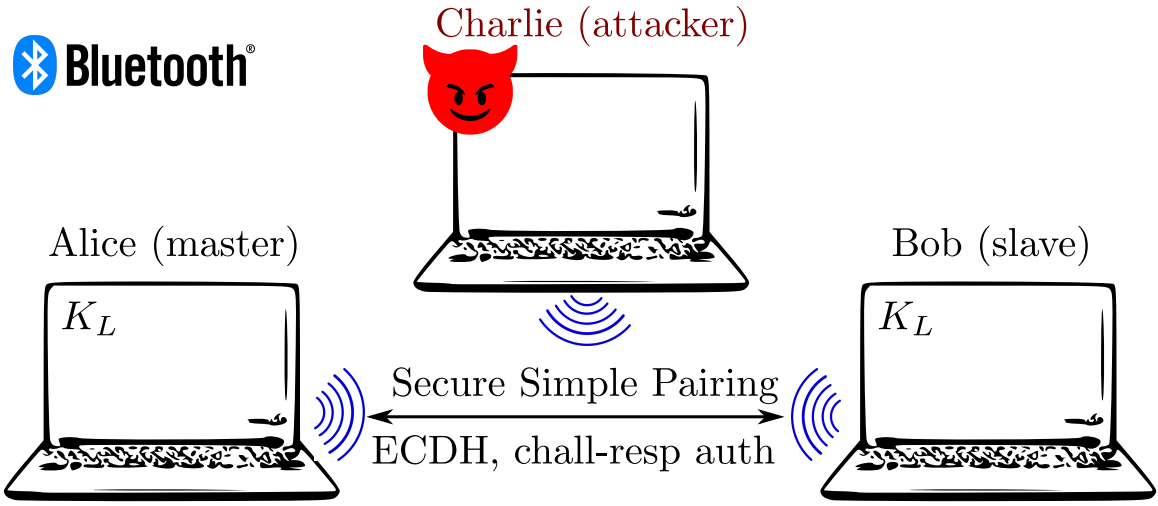
Bluetooth Security Overview

- **Pairing**
  - Establish a long term key (SSP based on ECDH)
- **Secure session establishment**
  - Establish a session key (derived from pairing key)
- Security mechanisms
  - Association: protect against man-in-the-middle attacks
  - **Key negotiation**: negotiate a key with variable entropy (strength)

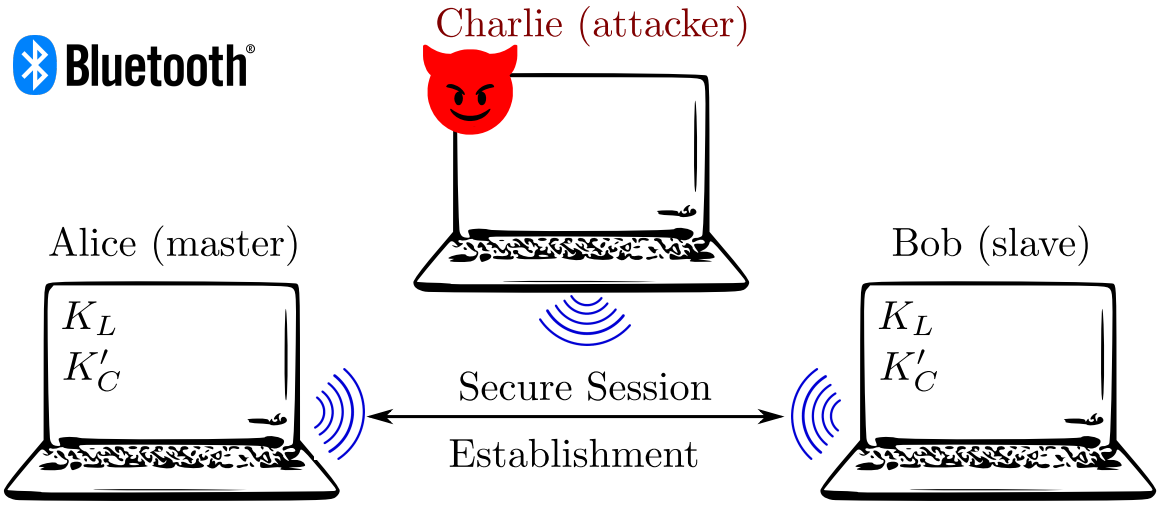
Bluetooth Threat Model



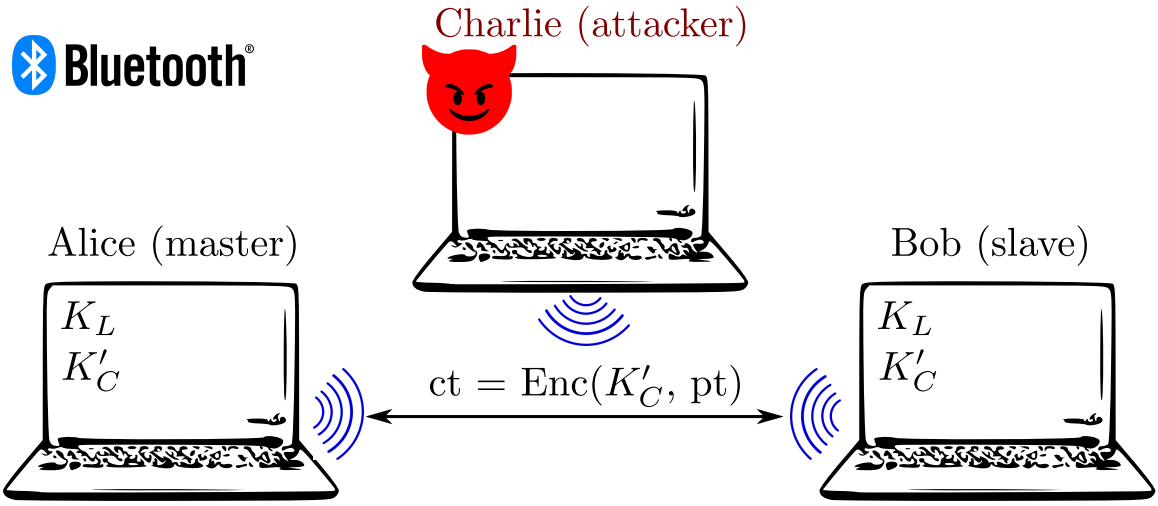
Bluetooth Threat Model



Bluetooth Threat Model

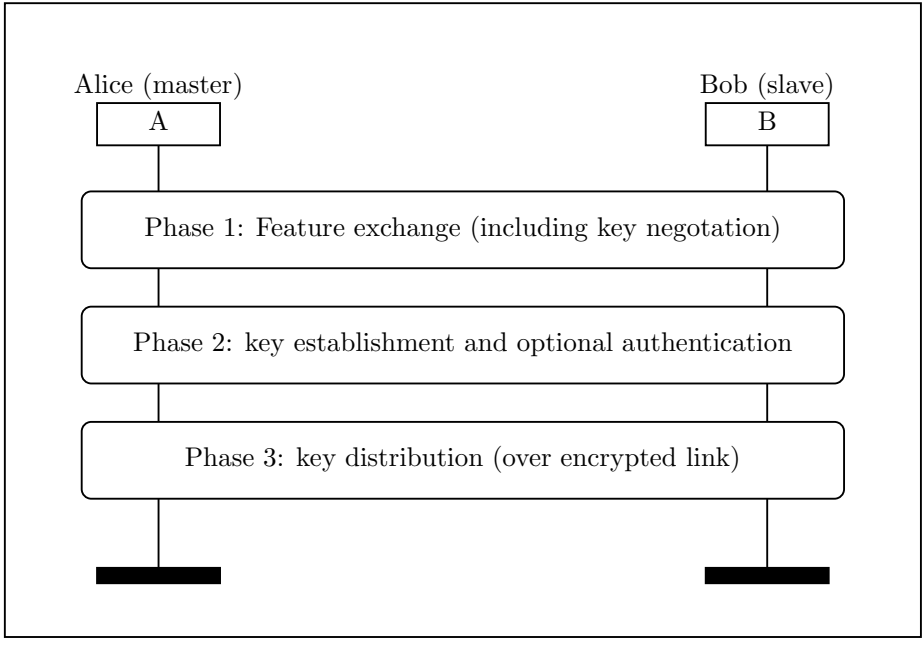


Bluetooth Threat Model

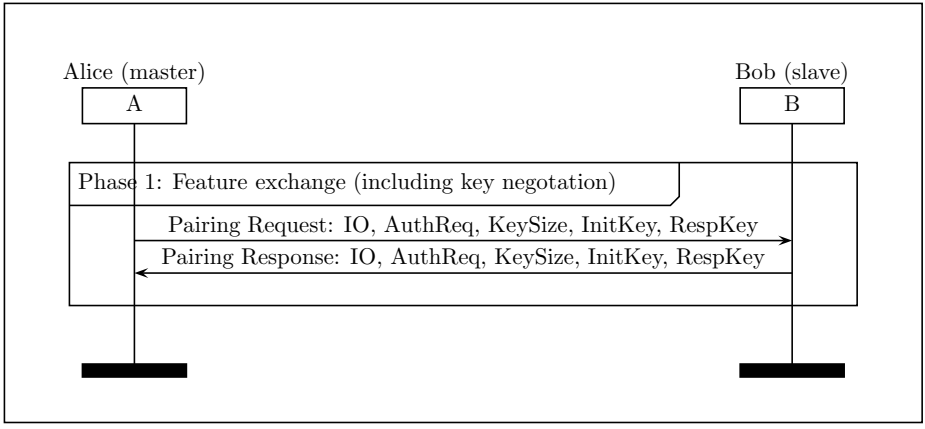


KNOB attack on BLE

BLE Pairing: Overview

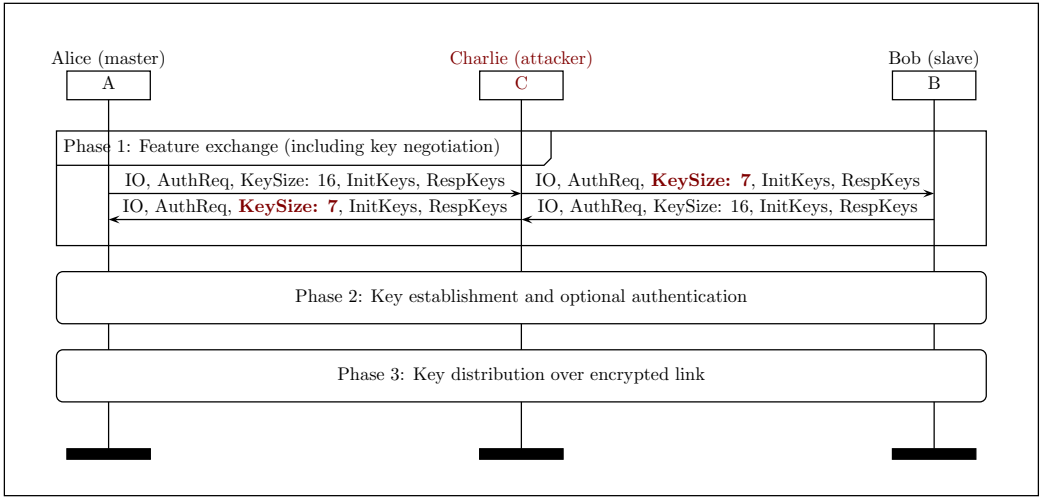


BLE Pairing: Key Negotiation



- Key negotiation issues (standard-compliant 0-days)
  - KeySize negotiation is **not protected**, i.e. no integrity, no encryption
  - KeySize values between **7 bytes** and 16 bytes

KNOB Attack on BLE Feature Exchange



- KNOB attack on BLE pairing
  - Attacker downgrades KeySize to 7 bytes
  - Victims' pairing and session keys have 7 bytes of entropy
  - Attacker brute-forces the low-entropy keys

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`
- Custom user-space BLE stack
  - Based on `PyBT` (`https://github.com/mikeryan/PyBT`)
  - That is based on `scapy` (`https://scapy.net`)

Evaluation of BLE KNOB Attack (19 devices, from Oct 2019)

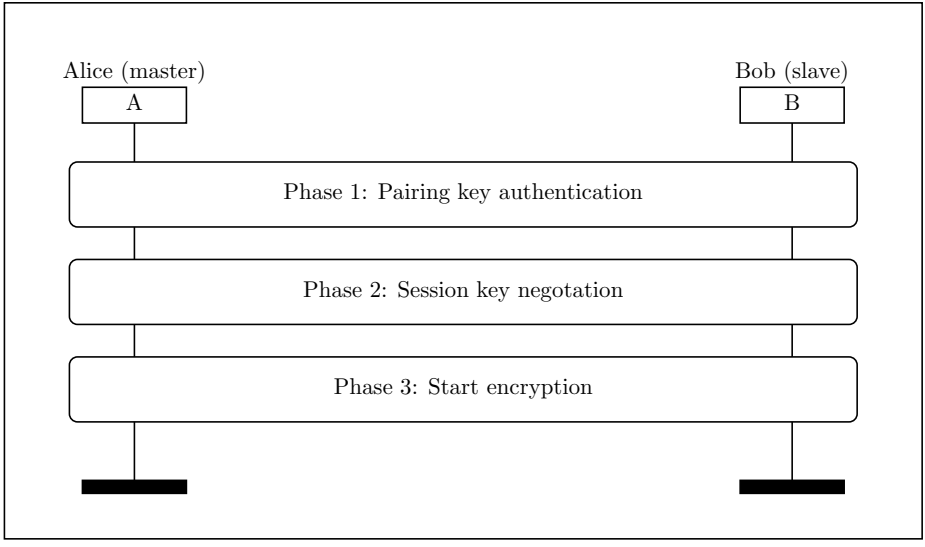
Device	OS (BLE Host)	Role	LTK Entropy
BLE Secure Connections (Bluetooth ≥ 4.2)			
Garmin Vivoactive 3	Proprietary	Peripheral	7 bytes
Google Pixel 2	Android	Central	7 bytes
LG K40	Android	Central	7 bytes
Samsung Gear S3	Tizen OS	Peripheral	7 bytes
Thinkpad X1 3rd	Linux	Central	7 bytes
Thinkpad X1 6rd	Linux	Central	7 bytes
TI CC1352R	TI RTOS	Central	7 bytes
BLE legacy security (Bluetooth 4.0 and 4.1)			
Comet Blue thermostat	Unknown	Peripheral	7 bytes
EDIFIER R1280DB speaker	Unknown	Peripheral	7 bytes
Fitbit Charge 2	Fitbit OS	Peripheral	7 bytes
ID115 HR Plus	Unknown	Peripheral	7 bytes
LG Nexus 5	Android	Central	7 bytes
Logitech MX Anywhere 2S	Nordic	Peripheral	7 bytes
Motorola G3	Android	Central	7 bytes
Samsung Galaxy J5	Android	Central	7 bytes
Samsung TV UE48J6250	Tizen OS	Peripheral	7 bytes
Xiaomi Mi band	Proprietary	Peripheral	7 bytes
Xiaomi Mi band 2 (x2)	Proprietary	Peripheral	7 bytes

KNOB attack on BT

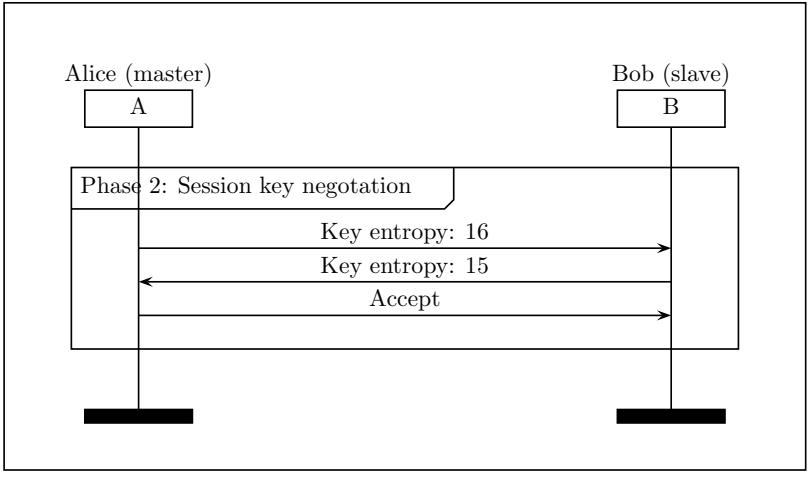
BT Pairing

- Alice and Bob
  - Securely paired over BT in absence of Charlie
  - Share a strong pairing key (16 bytes of entropy)

BT Session Establishment: Overview

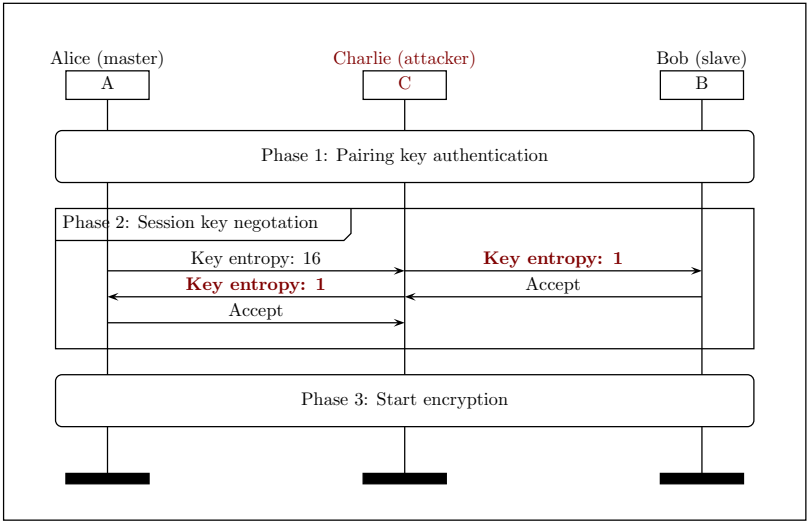


BT Session Establishment: Session Key Negotiation



- Key negotiation issues (standard-compliant 0-days)
  - 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 Session Key Negotiation



- KNOB attack on BT secure session establishment
  - Attacker downgrades key entropy to 1 bytes
  - Attacker brute-forces the low-entropy key

Implementation of KNOB Attack on BT

- Link Manager Protocol (LMP) manipulation
  - Implemented in the BT controller (firmware)

- Custom version of internalblue
  - RE Nexus 5 BT firmware
  - Write ARM patches for LMP
  - Patch Nexus 5 at runtime

Evaluation of BT KNOB Attack (38 devices, from Jun 2019)

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

Evaluation of BT KNOB Attack (38 devices, from Jun 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

KNOB Attacks Countermeasures

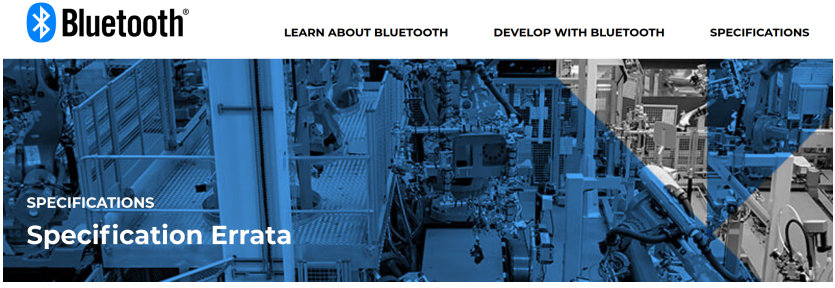
Our countermeasures for BT and BLE

- 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

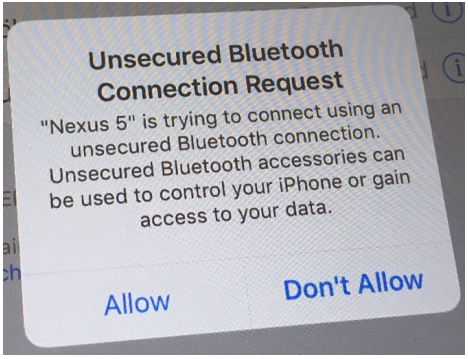
Bluetooth SIG amended the standard (2019-08-13)

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



https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=470741

KNOB on BT: Apple mitigation



https://twitter.com/seemoolab/status/1169363042548760577/photo/1

- 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

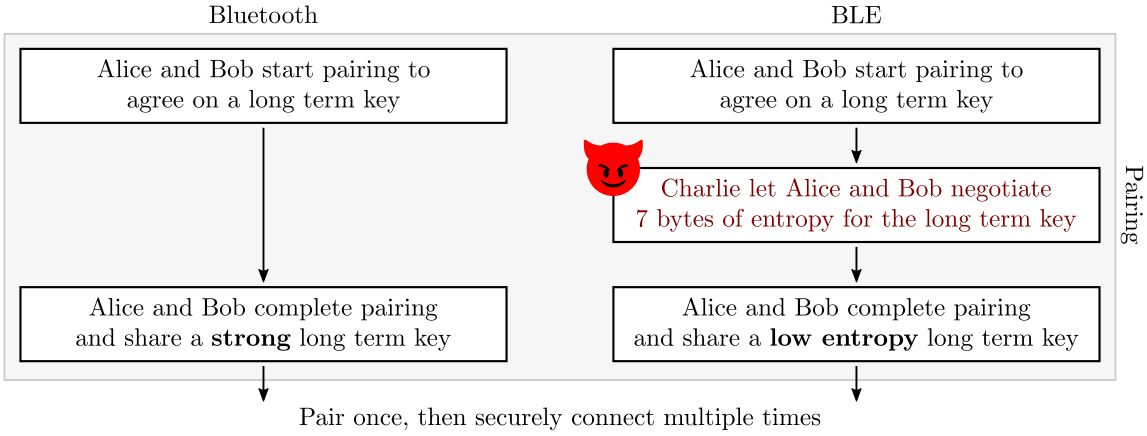
KNOB on BT: Google and Linux mitigation

BlueZ
Official Linux Bluetooth protocol stack

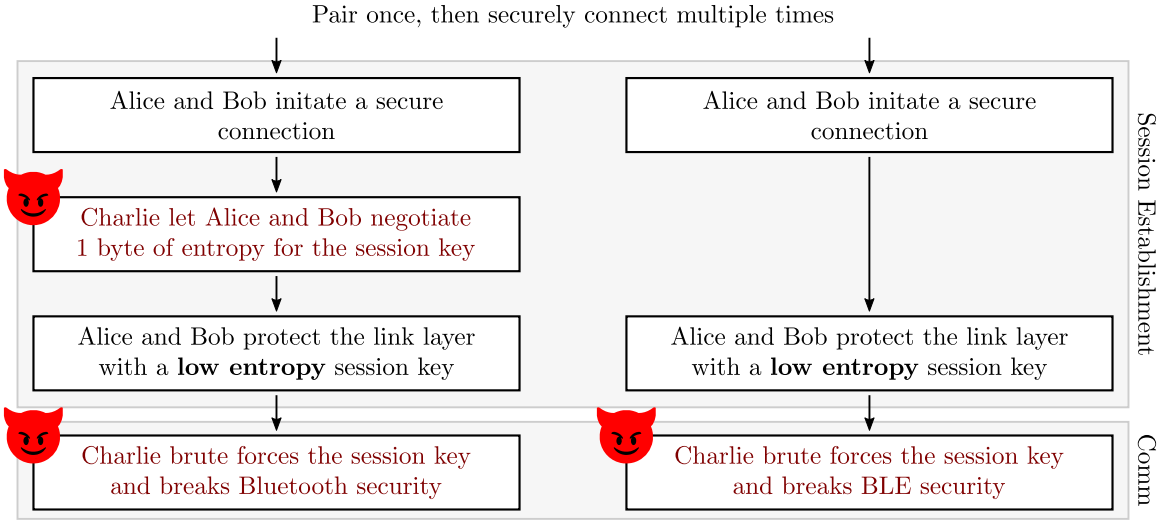
- OS patch
  - Checks entropy and terminates the session if entropy is less than 7 bytes
  - Uses *HCI Read Encryption Key Size* command
- Shifting responsibilities to the OS can still be bad!
  - Malicious OS can still negotiate 1 byte of entropy

Conclusion

KNOB BT vs. BLE: Pairing



KNOB BT vs. BLE: Secure Session Establishment



Current State of Bluetooth security

- 7 bytes of entropy for a key is too low (comparable to DES)
- BT and BLE key negotiations remain un-protected
- Entropy negotiation does not provide runtime benefits (key size stays constant)
- Most of the BT devices are still vulnerable to the 1 byte downgrade

From the Bluetooth Standard to Standard-Compliant 0-days

- Bluetooth Standard
  - Specifies *Bluetooth Classic (BT)* and *Bluetooth Low Energy (BLE)*
- Standard-compliant 0-days (vulnerabilities)
  - Very effective and difficult to patch
- **Key Negotiation of Bluetooth (KNOB) attacks on BT and BLE**
  - More info at `https://knobattack.com`
  - Try it yourself at `https://github.com/francozappa/knob`