WAC workshop 2020

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
  - CVE-2019-9506: https://www.kb.cert.org/vuls/id/918987/
- 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]

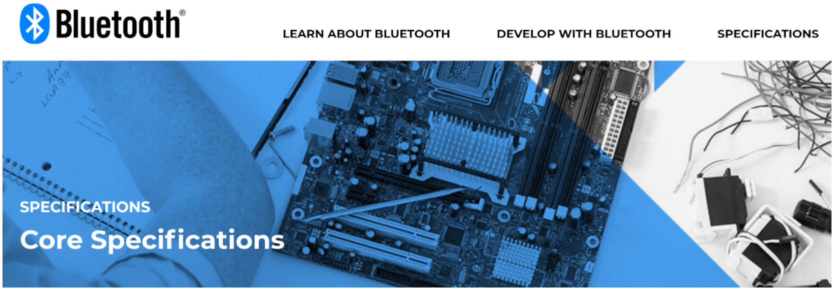
Part 1: Introduction about Bluetooth

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

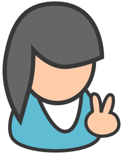
Bluetooth Standard

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

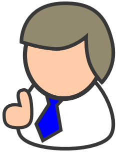


https://www.bluetooth.com/specifications/bluetooth-core-specification/

Bluetooth Security: Pairing and Secure Sessions

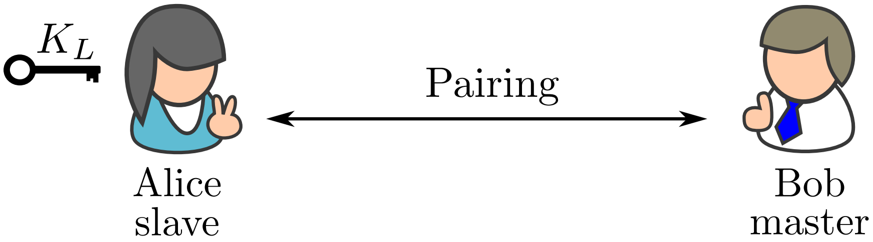


Alice
slave

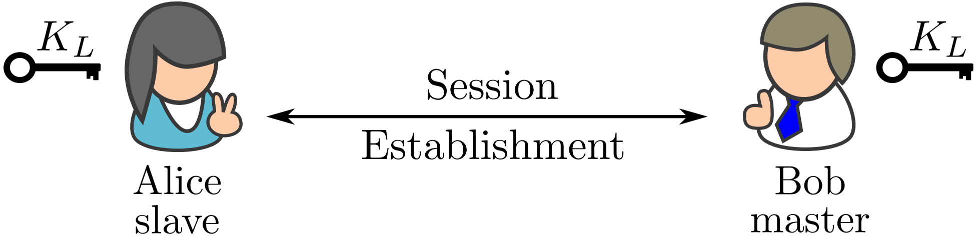


Bob
master

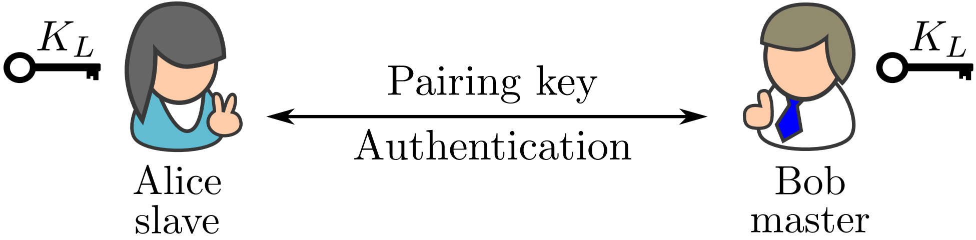
Bluetooth Security: Pairing and Secure Sessions



Bluetooth Security: Pairing and Secure Sessions

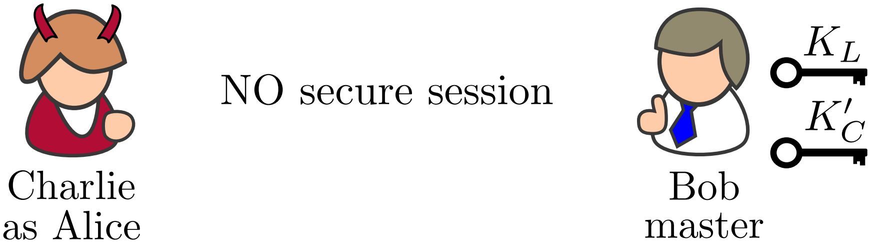


Bluetooth Security: Pairing and Secure Sessions

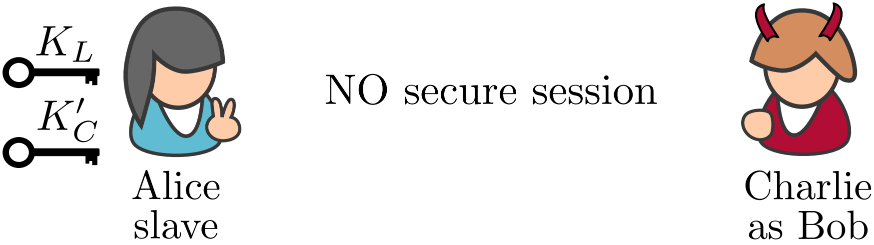


Bluetooth Security: Pairing and Secure Sessions

Bluetooth Security: Pairing and Secure Sessions


Charlie
as Alice
NO secure session
Bob
master
K_L
K'_C

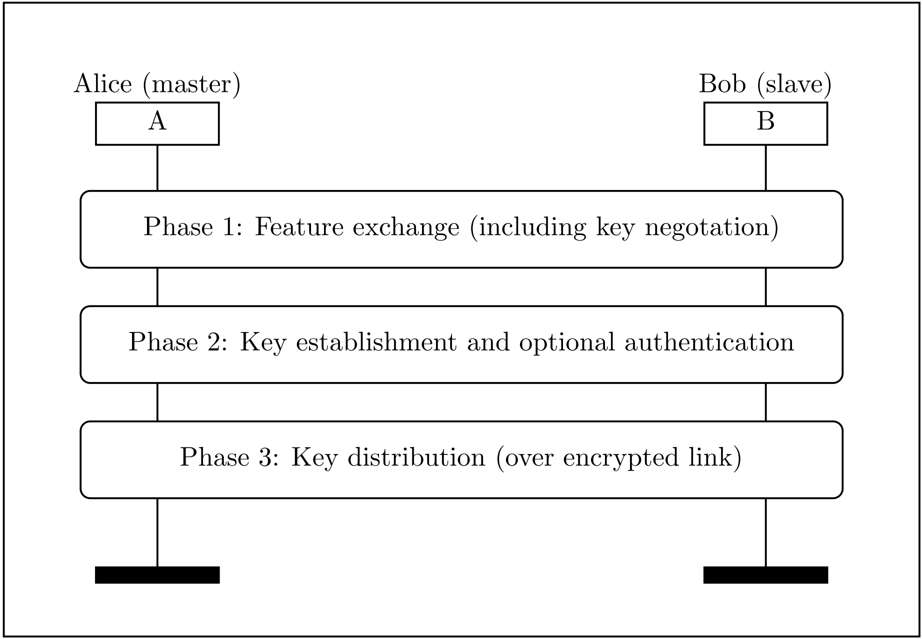
Bluetooth Security: Impersonation and MitM



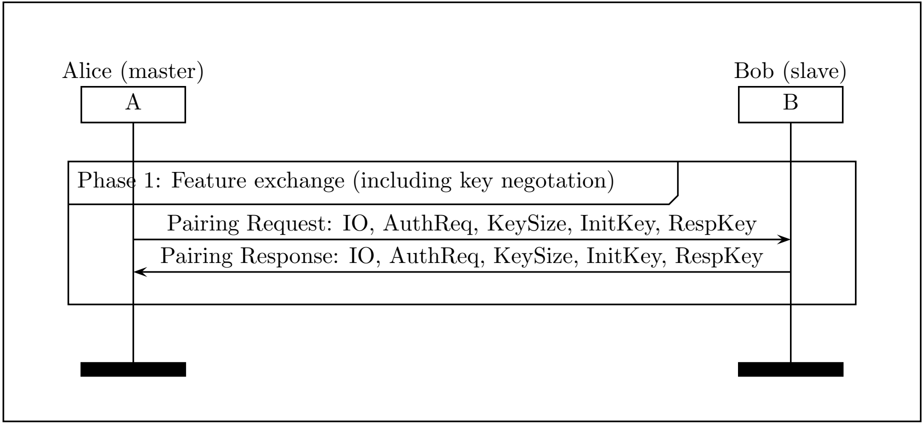
Bluetooth Security: Impersonation and MitM

Part 2: KNOB Attack on BLE

BLE Pairing

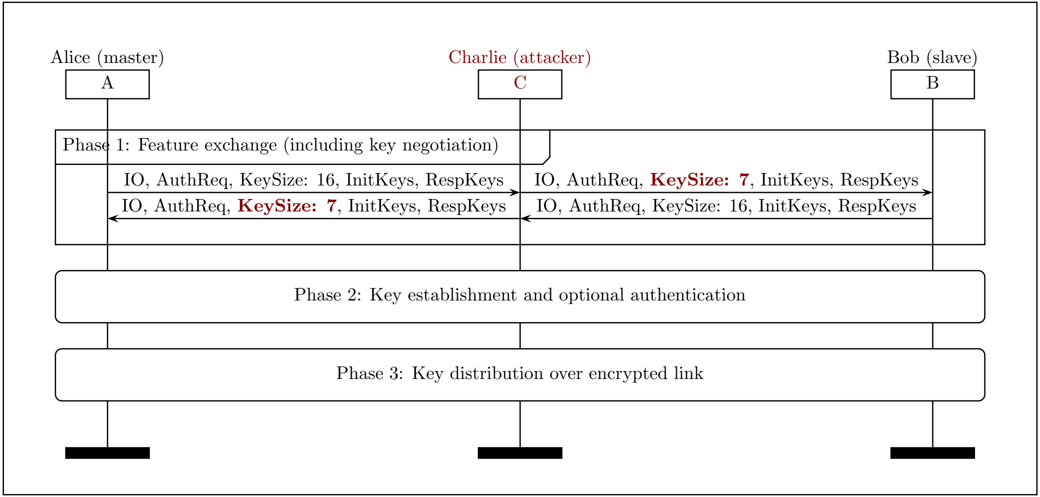

Alice (master)
Bob (slave)
A
B
Phase 1: Feature exchange (including key negotation)
Phase 2: Key establishment and optional authentication
Phase 3: Key distribution (over encrypted link)

Issues with BLE Pairing (Key Negotiation)



- Issues
  - KeySize negotiation is **not protected**, i.e. no integrity, no encryption
  - KeySize values (pairing key strenght) 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)

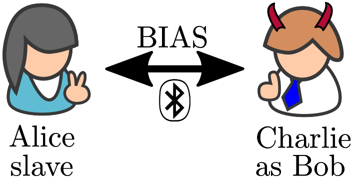
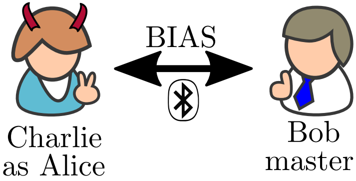
Bluetooth Authentication Mechanisms

- Legacy Secure Connection (LSC) authentication
  - Unilateral, challenge-response
- Secure Connection (SC) authentication
  - Mutual, challenge-response
- LSC or SC negotiated during secure session establishment

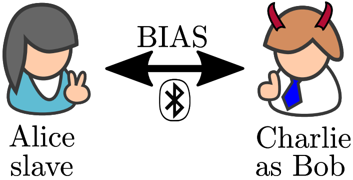
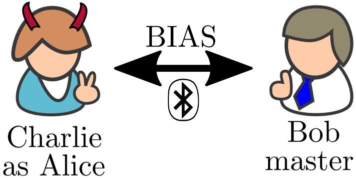
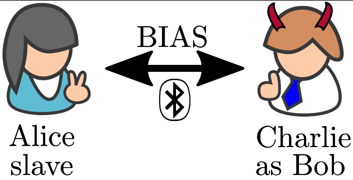
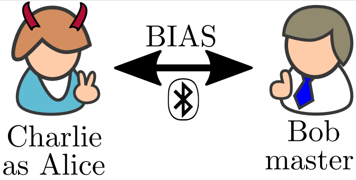
BIAS Attacks on Bluetooth Session Establishment

BIAS Attacks	Master Impersonation	Slave Impersonation
Legacy Secure Connections		
Secure Connections		

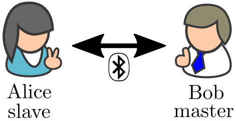
BIAS Attacks on Bluetooth Session Establishment

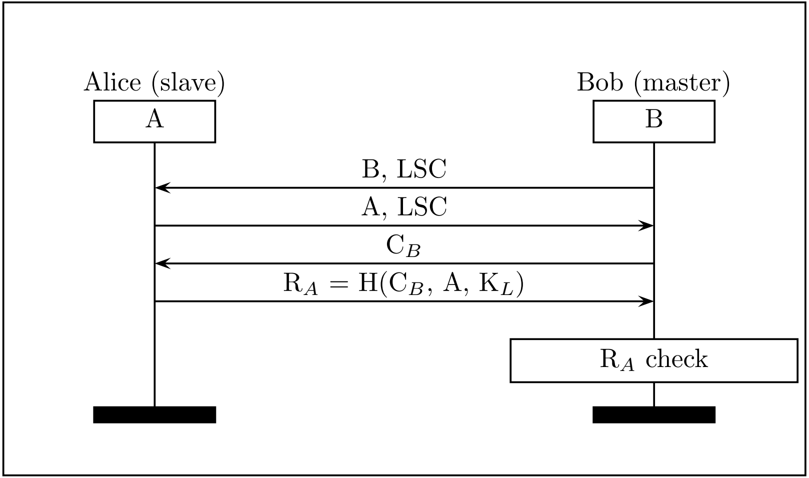
BIAS Attacks	Master Impersonation	Slave Impersonation
Legacy Secure Connections		
Secure Connections		

BIAS Attacks on Bluetooth Session Establishment

BIAS Attacks	Master Impersonation	Slave Impersonation
Legacy Secure Connections		
Secure Connections		

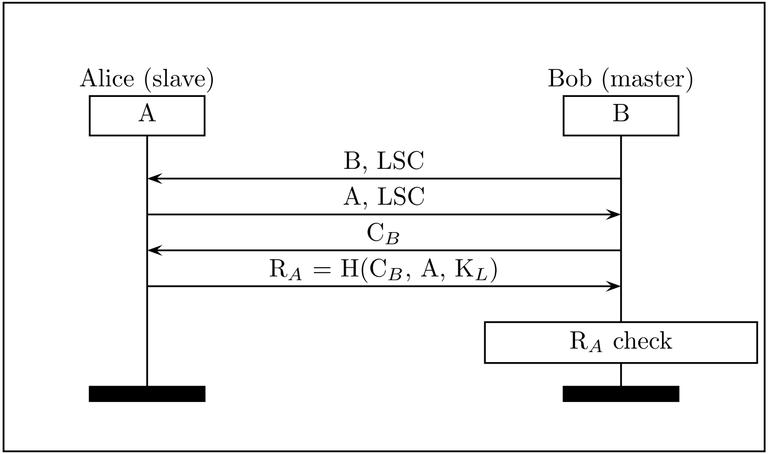
Legacy Secure Connection (LSC) Authentication



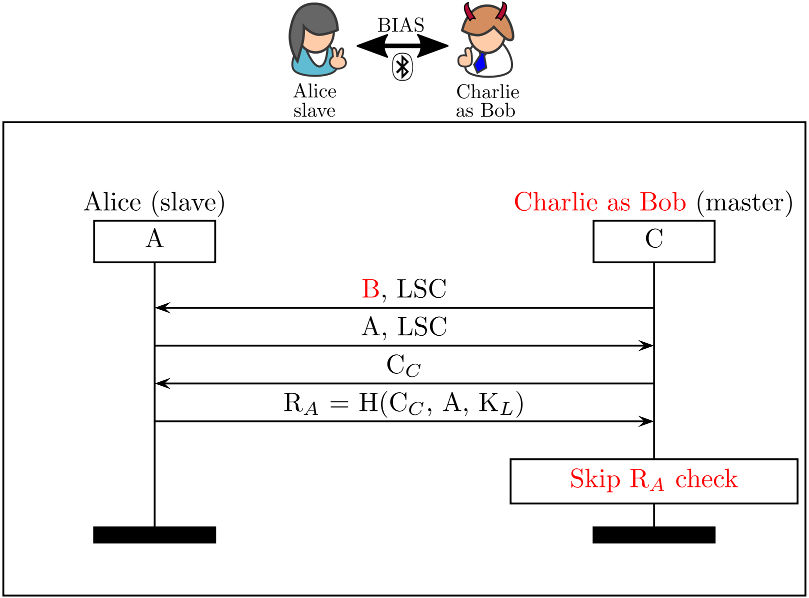


Issues with LSC Authentication

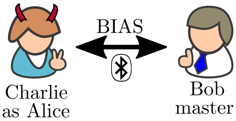
- LSC authentication is **not used mutually** for session establishment
- A device can **switch authentication role**

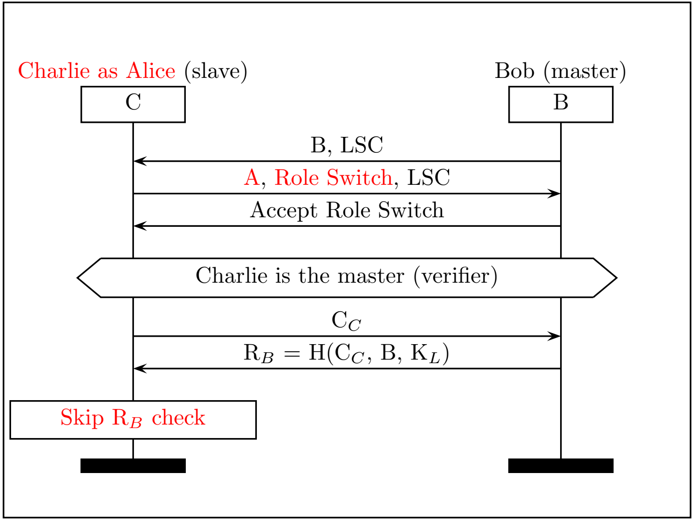


BIAS Attack on LSC: Master Impersonation

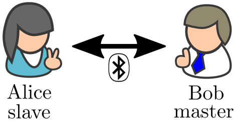


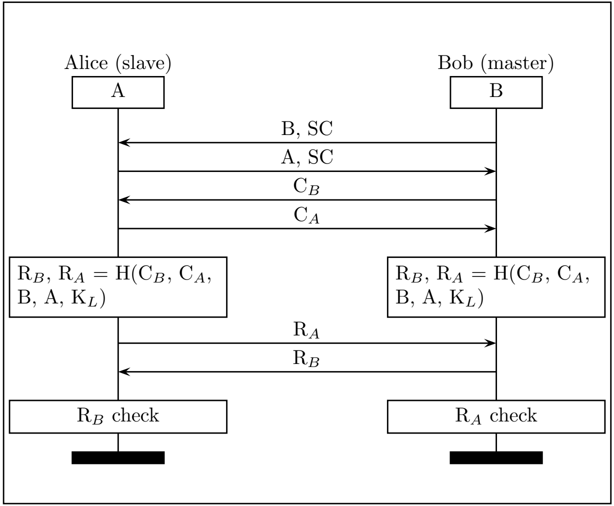
BIAS Attack on LSC: Slave Impersonation





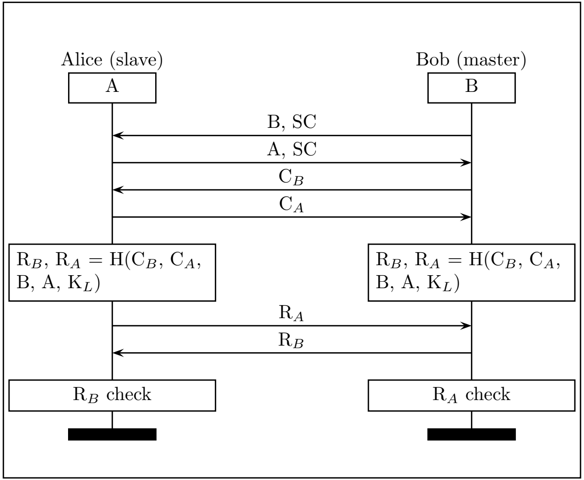
Secure Connections (SC) Authentication



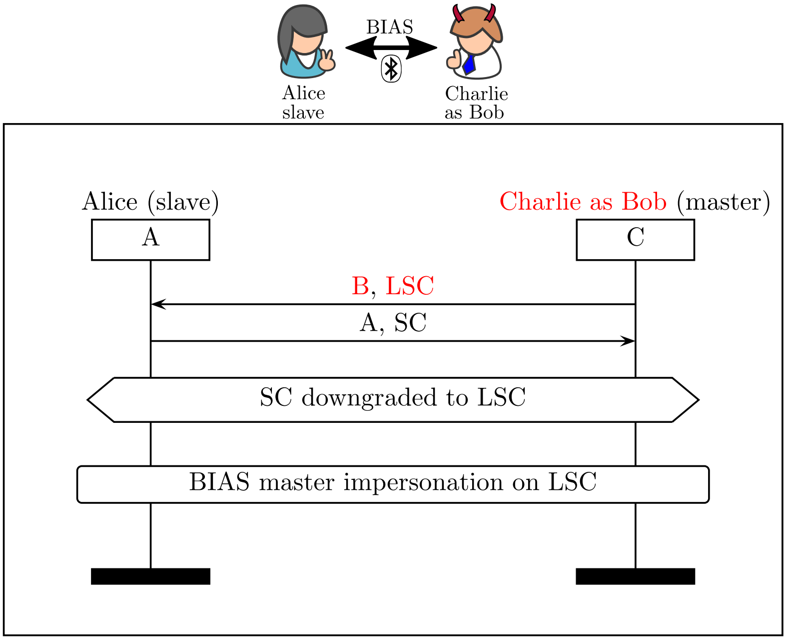


Issues with SC Authentication

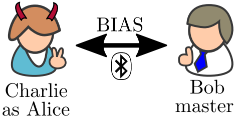
- SC negotiation **is not integrity-protected**
- SC support is **not enforced** for pairing and session establishment

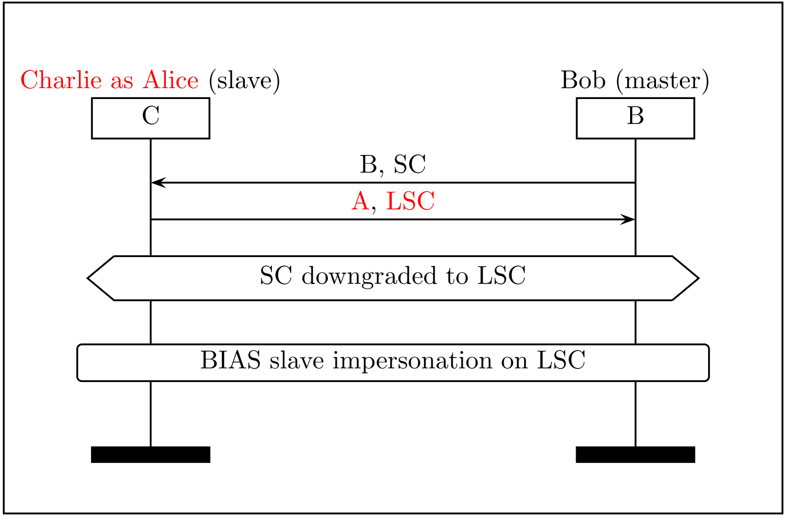


BIAS Attack on SC: Master Impersonation



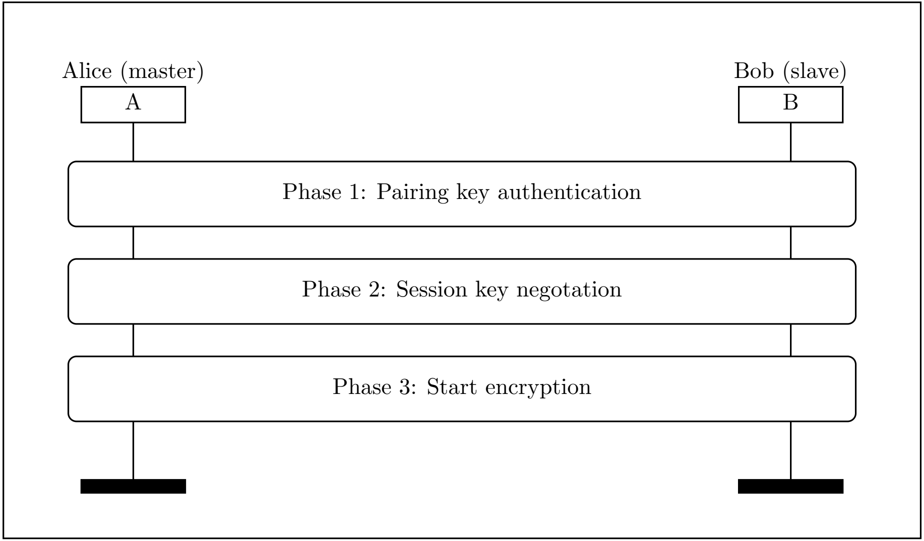
BIAS Attack on SC: Slave Impersonation



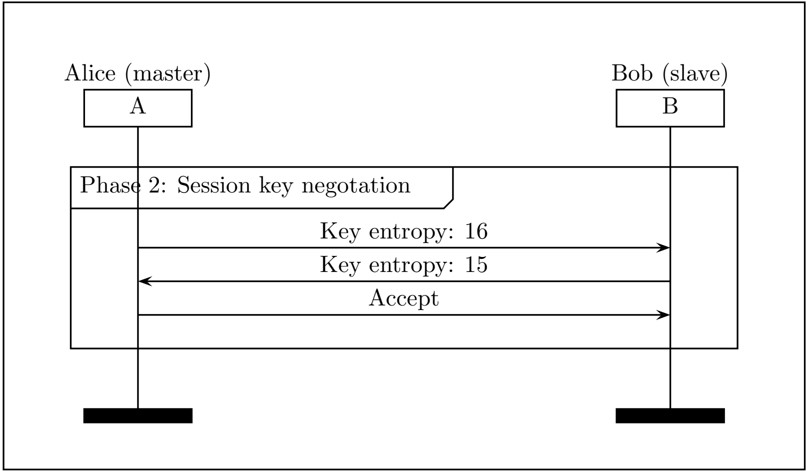


Part 2: KNOB Attack on BT

BT Session Establishment: Overview

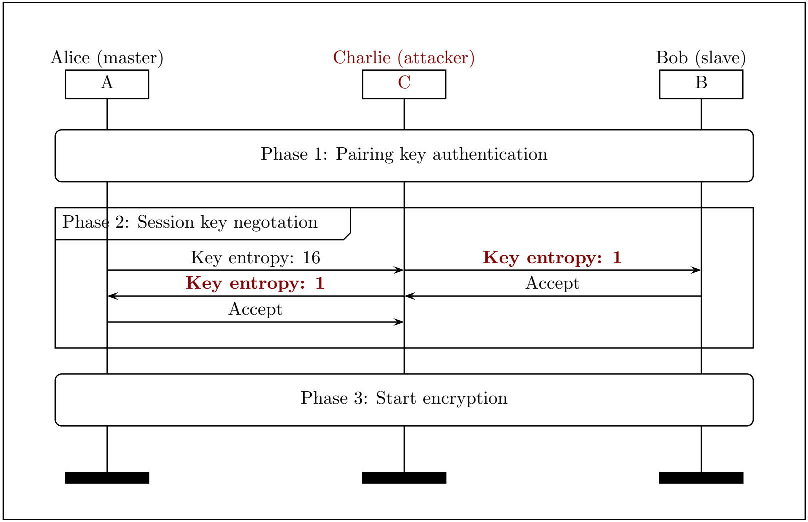


BT Session Establishment: Session Key Negotiation



- Issues
  - Key entropy negotiation is **not protected**, i.e. no integrity, no encryption
  - Key entropy values between **1 byte** and 16 bytes

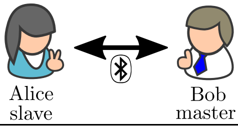
KNOB Attack on BT

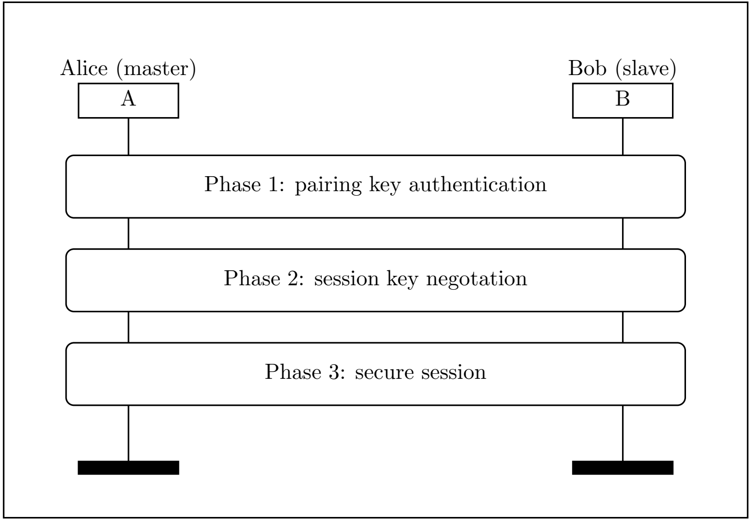


- KNOB attack on BT
  - Downgrade BT session key entropy to 1 bytes
  - Brute-force the session key and break BT security

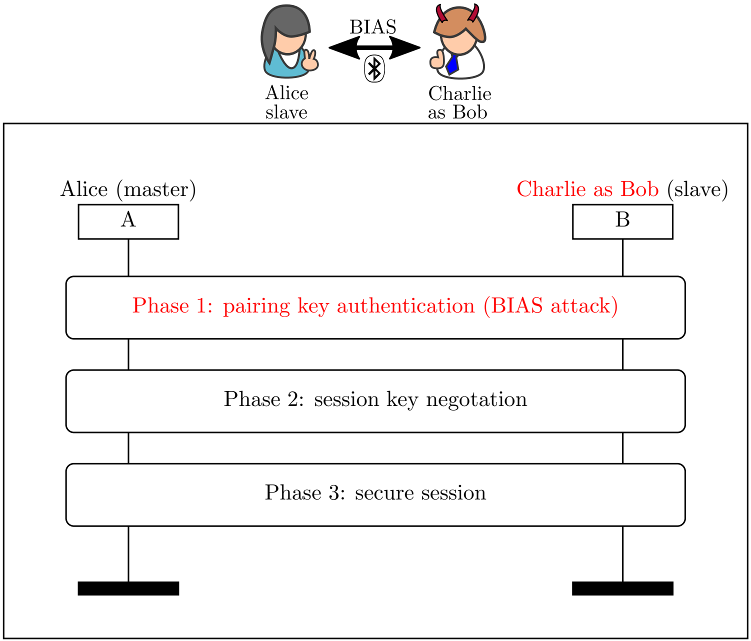
Part 3: BIAS + KNOB

BIAS + KNOB: Break Bluetooth Session Establishment

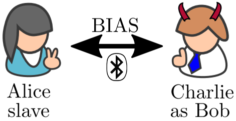


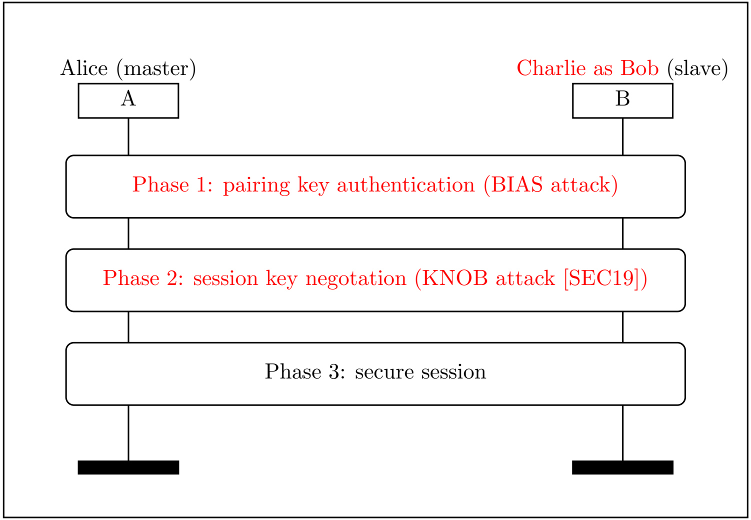


BIAS + KNOB: Break Bluetooth Session Establishment

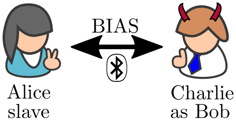


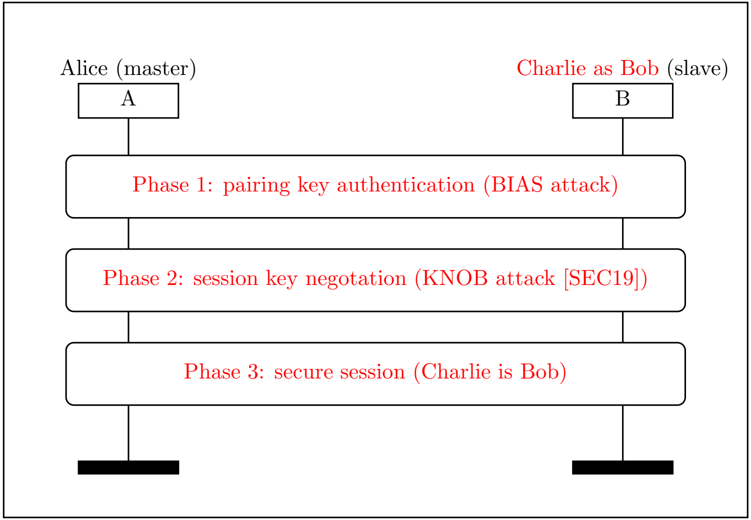
BIAS + KNOB: Break Bluetooth Session Establishment



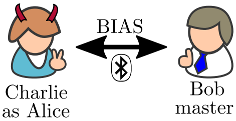


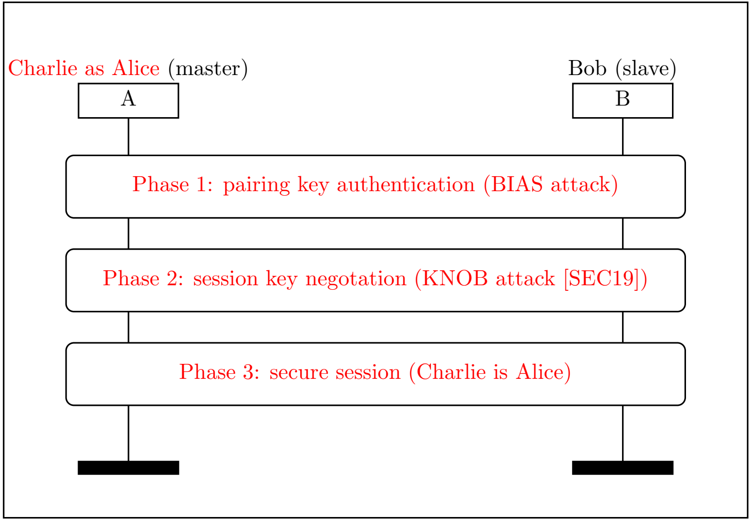
BIAS + KNOB: Break Bluetooth Session Establishment





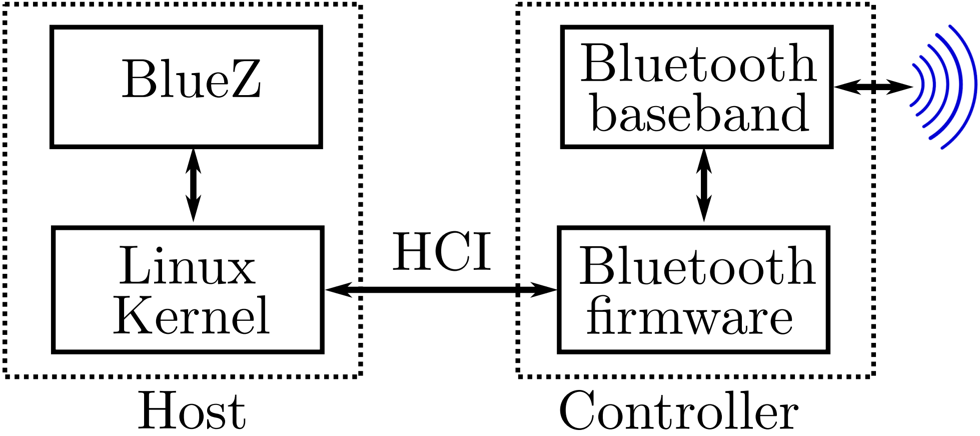
BIAS + KNOB: Break Bluetooth Session Establishment





Part 3: Implementation

Host, Controller, and Host Controller Interface (HCI)



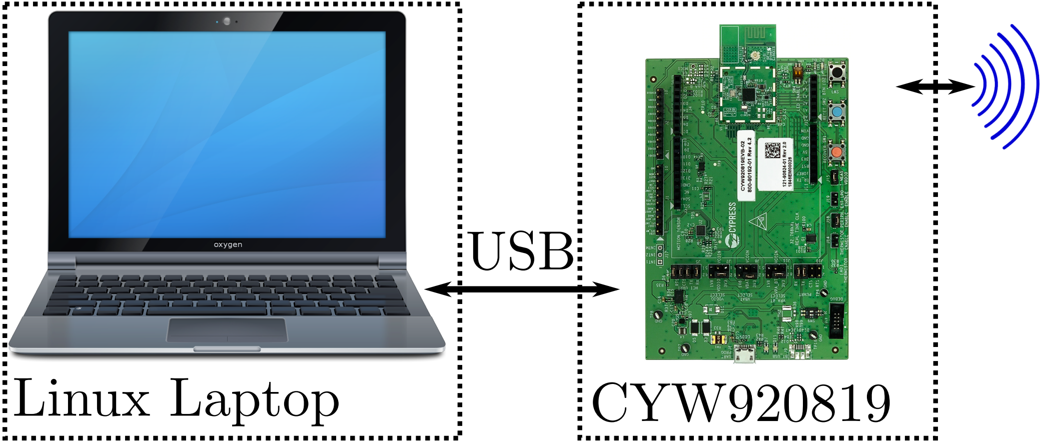
Implementation of KNOB Attack on BLE

- 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`

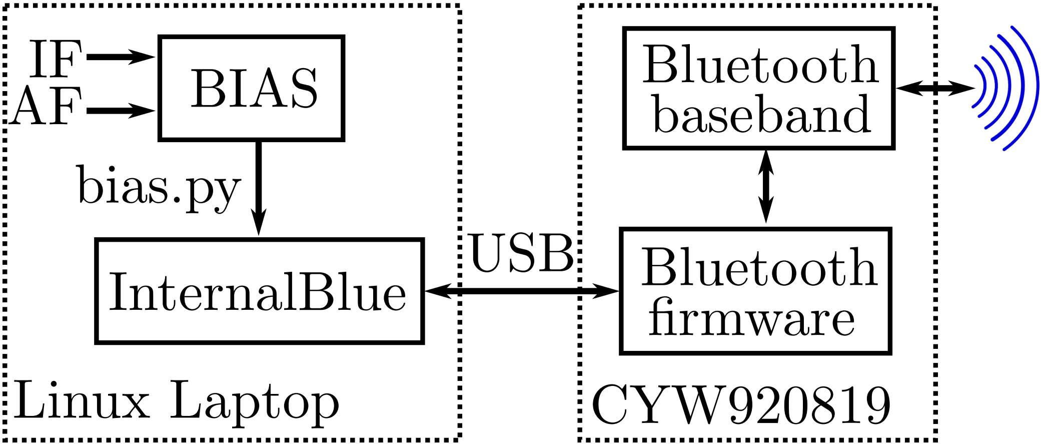
- Custom user-space BLE stack
  - Based on `PyBT` (`https://github.com/mikeryan/PyBT`)
  - That is based on `scapy` (`https://scapy.net`)

Implementation of BIAS Attacks on BT



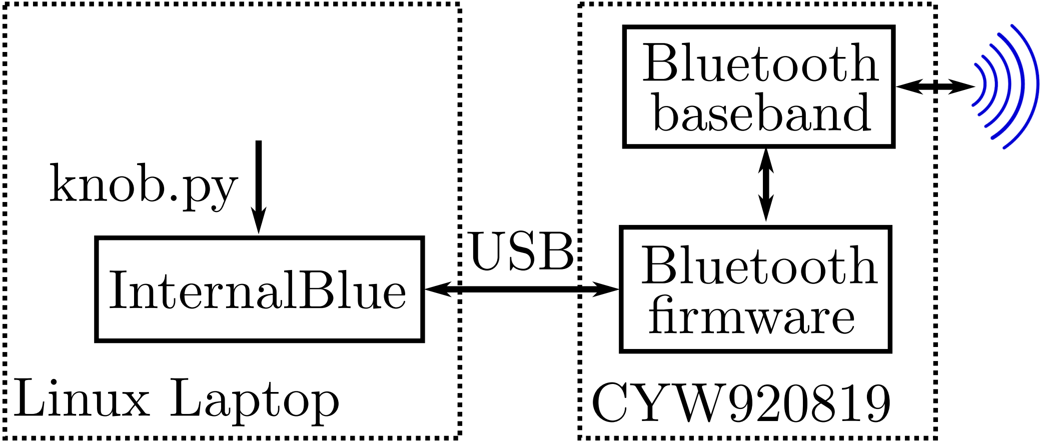
https://github.com/francozappa/bias
https://github.com/seemoo-lab/internalblue

Implementation of BIAS Attacks on BT



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)

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

Evaluation: BIAS on BT (31 devices, from 2020)

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

Evaluation: BIAS on BT (31 devices, from 2020)

		LSC		SC	
Chip	Device(s)	MI	SI	MI	SI
Bluetooth v4.1					
CYW4334	iPhone 5s	●	●	-	-
CYW4339	Nexus 5, iPhone 6	●	●	-	-
CYW43438	RPi 3B+	●	●	●	●
Snapdragon 210	LG K4	●	●	●	●
Snapdragon 410	Motorola G3, Galaxy J5	●	●	●	●
Bluetooth v$\leq$ 4.0					
BCM20730	ThinkPad 41U5008	●	○	-	-
BCM4329B1	iPad MC349LL	●	●	-	-
CSR 6530	PLT BB903+	●	●	-	-
CSR 8648	Philips SHB7250	●	●	-	-
Exynos 3470	Galaxy S5 mini	●	●	-	-
Exynos 3475	Galaxy J3 2016	●	●	-	-
Intel 1280	Lenovo U430	●	●	-	-
Intel 6205	ThinkPad X230	●	●	-	-
Snapdragon 200	Lumia 530	●	●	-	-

Evaluation: KNOB on BT (38 devices, from 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: 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 $\leq$ 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

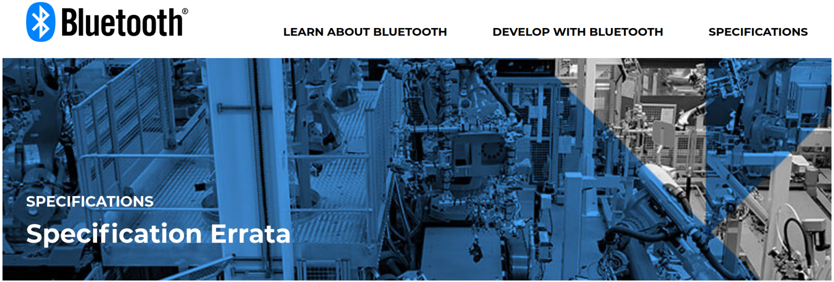
Counter KNOB Attacks on BT and BLE

- 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

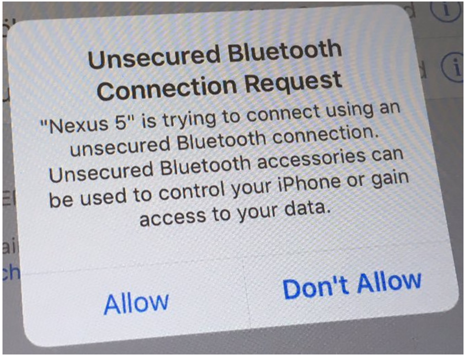
Bluetooth SIG amended the standard (2019-08-13)

- 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



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

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
  - https://www.bluetooth.com/learn-about-bluetooth/bluetooth-technology/bluetooth-security/bias-vulnerability/
- Vendors
  - ????
- Bottom line
  - No concrete mitigations put in place

P3: Conclusion

KNOB and BIAS Attacks Recap

- KNOB attack on BLE
  - Compute BLE paring 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, BLESA, 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`