USENIX 2019 @ Santa Clara, US

The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR

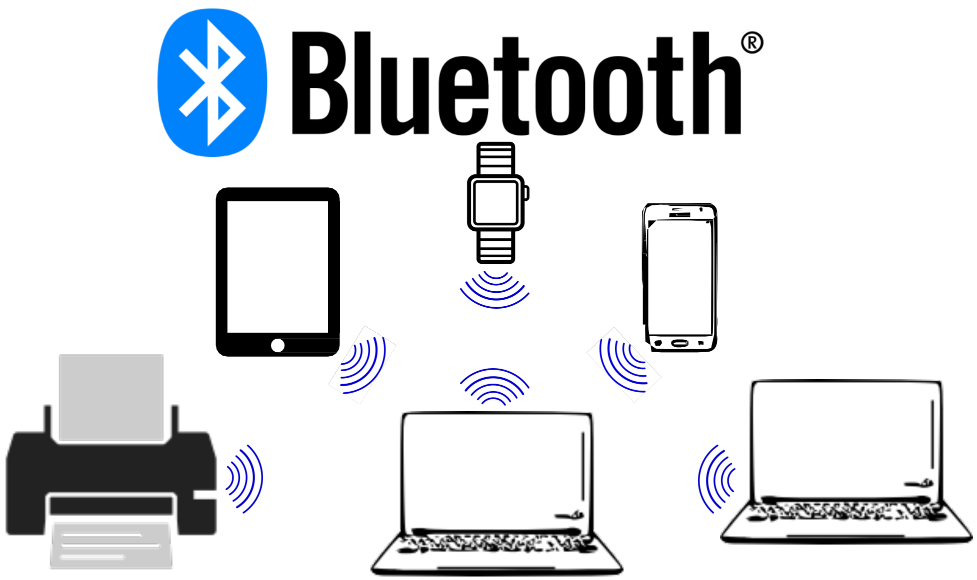
Daniele Antonioli[1], Nils Ole Tippenhauer[2], Kasper Rasmussen[3]

[1]Singapore University of Technology and Design (SUTD)
[2]CISPA Helmholtz Center for Information Security
[3]University of Oxford

Presenter: Kyeong Tae Kim
Slides mostly borrowed from Daniele Antonioli

Bluetooth

- Pervasive wireless technology for personal area networks
- E.g., mobile, automotive, medical, and industrial devices



This paper not about BLE, but about Bluetooth Classic!

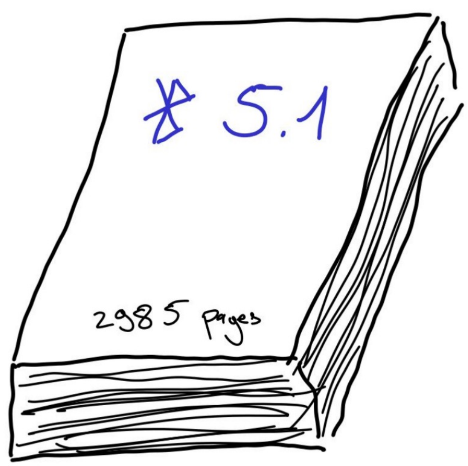
BLE (Bluetooth Low Energy)

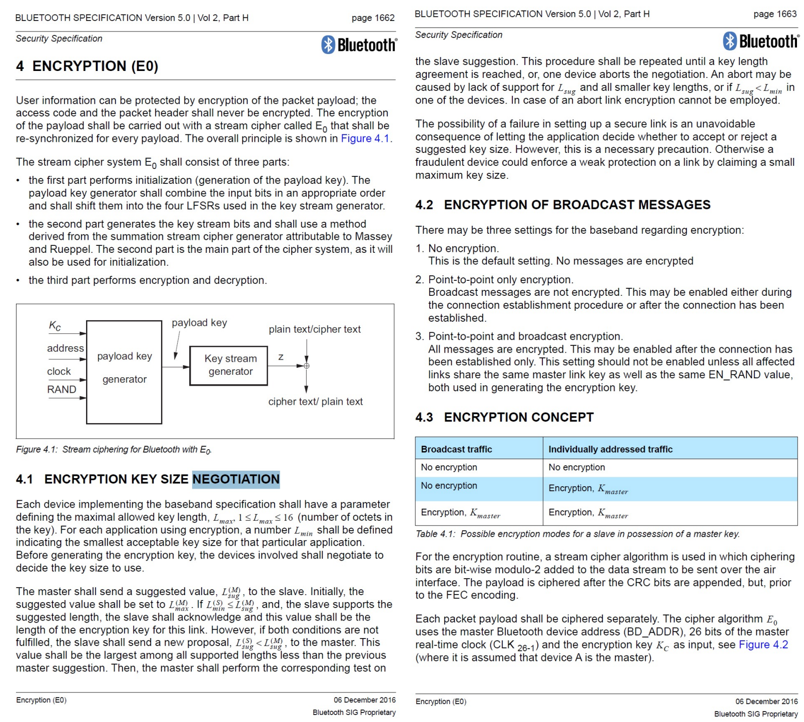
- Started at Bluetooth 4.0 specification
- Open source implementation available in Linux drivers
- Security at various perspectives has been explored on BLE
  - Mike Ryan, **Bluetooth: With Low Energy comes Low Security** (USENIX WOOT ‘13)
  - Kassem Fawaz, Kyu-Han Kim, Kang G. Shin, **Protecting Privacy of BLE Device Users** (USENIX Security ‘16)

This paper not about BLE, but about Bluetooth Classic!

Bluetooth BR/EDR (or Classic)

- Open but complex specification
- No public reference implementation
- Unexplored!!
- Uses custom security mechanisms at the link layer





BLUETOOTH SPECIFICATION Version 5.0 | Vol 2, Part H page 1662

Security Specification

Bluetooth

4 ENCRYPTION (E0)

User information can be protected by encryption of the packet payload; the access code and the packet header shall never be encrypted. The encryption of the payload shall be carried out with a stream cipher called E_0 that shall be re-synchronized for every payload. The overall principle is shown in Figure 4.1.

The stream cipher system E_0 shall consist of three parts:

- the first part performs initialization (generation of the payload key). The payload key generator shall combine the input bits in an appropriate order and shall shift them into the four LFSRs used in the key stream generator.
- the second part generates the key stream bits and shall use a method derived from the summation stream cipher generator attributable to Massey and Rueppel. The second part is the main part of the cipher system, as it will also be used for initialization.
- the third part performs encryption and decryption.

Figure 4.1: Stream ciphering for Bluetooth with E_0.

4.1 ENCRYPTION KEY SIZE NEGOTIATION

Each device implementing the baseband specification shall have a parameter defining the maximal allowed key length, L_{max}, $1 \leq L_{max} \leq 16$ (number of octets in the key). For each application using encryption, a number L_{min} shall be defined indicating the smallest acceptable key size for that particular application. Before generating the encryption key, the devices involved shall negotiate to decide the key size to use.

The master shall send a suggested value, $L_{sug}^{(M)}$, to the slave. Initially, the suggested value shall be set to $L_{max}^{(M)}$. If $L_{min}^{(S)} \leq L_{sug}^{(M)}$, and, the slave supports the suggested length, the slave shall acknowledge and this value shall be the length of the encryption key for this link. However, if both conditions are not fulfilled, the slave shall send a new proposal, $L_{sug}^{(S)} < L_{sug}^{(M)}$, to the master. This value shall be the largest among all supported lengths less than the previous master suggestion. Then, the master shall perform the corresponding test on

Encryption (E0) 06 December 2016
Bluetooth SIG Proprietary

BLUETOOTH SPECIFICATION Version 5.0 | Vol 2, Part H page 1663

Security Specification

Bluetooth

the slave suggestion. This procedure shall be repeated until a key length agreement is reached, or, one device aborts the negotiation. An abort may be caused by lack of support for L_{sug} and all smaller key lengths, or if $L_{sug} < L_{min}$ in one of the devices. In case of an abort link encryption cannot be employed.

The possibility of a failure in setting up a secure link is an unavoidable consequence of letting the application decide whether to accept or reject a suggested key size. However, this is a necessary precaution. Otherwise a fraudulent device could enforce a weak protection on a link by claiming a small maximum key size.

4.2 ENCRYPTION OF BROADCAST MESSAGES

There may be three settings for the baseband regarding encryption:

1. No encryption.
   This is the default setting. No messages are encrypted
2. Point-to-point only encryption.
   Broadcast messages are not encrypted. This may be enabled either during the connection establishment procedure or after the connection has been established.
3. Point-to-point and broadcast encryption.
   All messages are encrypted. This may be enabled after the connection has been established only. This setting should not be enabled unless all affected links share the same master link key as well as the same EN_RAND value, both used in generating the encryption key.

4.3 ENCRYPTION CONCEPT

Broadcast traffic	Individually addressed traffic
No encryption	No encryption
No encryption	Encryption, K_{master}
Encryption, K_{master}	Encryption, K_{master}

Table 4.1: Possible encryption modes for a slave in possession of a master key.

For the encryption routine, a stream cipher algorithm is used in which ciphering bits are bit-wise modulo-2 added to the data stream to be sent over the air interface. The payload is ciphered after the CRC bits are appended, but, prior to the FEC encoding.

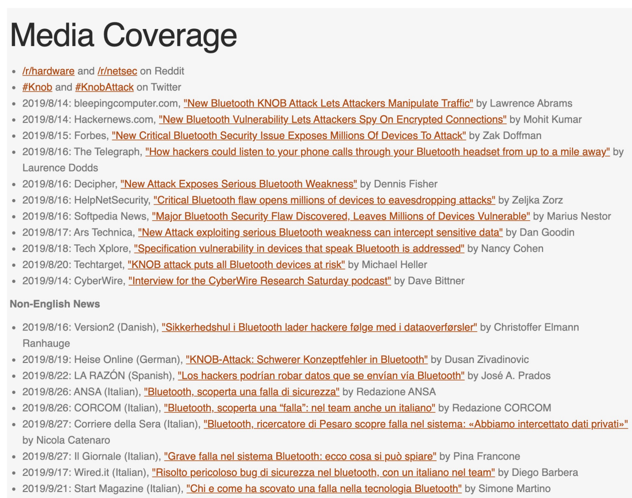
Each packet payload shall be ciphered separately. The cipher algorithm E_0 uses the master Bluetooth device address (BD_ADDR), 26 bits of the master real-time clock (CLK_{26-1}) and the encryption key K_C as input, see Figure 4.2 (where it is assumed that device A is the master).

Encryption (E0) 06 December 2016
Bluetooth SIG Proprietary

Introduction

The attacker completely breaks Bluetooth classic security without being detected.

- 1 Byte key guessing is enough!!
- Affects any standard compliant Bluetooth device
  - 14 vulnerable chips (Intel, Broadcom, Apple, and Qualcomm)
  - 21 vulnerable devices



Media Coverage

- /r/hardware and /r/netsec on Reddit
- #Knob and #KnobAttack on Twitter
- 2019/8/14: bleepingcomputer.com, "New Bluetooth KNOB Attack Lets Attackers Manipulate Traffic" by Lawrence Abrams
- 2019/8/14: Hackernews.com, "New Bluetooth Vulnerability Lets Attackers Spy On Encrypted Connections" by Mohit Kumar
- 2019/8/15: Forbes, "New Critical Bluetooth Security Issue Exposes Millions Of Devices To Attack" by Zak Doffman
- 2019/8/16: The Telegraph, "How hackers could listen to your phone calls through your Bluetooth headset from up to a mile away" by Laurence Dodds
- 2019/8/16: Decipher, "New Attack Exposes Serious Bluetooth Weakness" by Dennis Fisher
- 2019/8/16: HelpNetSecurity, "Critical Bluetooth flaw opens millions of devices to eavesdropping attacks" by Zeljka Zorz
- 2019/8/16: Softpedia News, "Major Bluetooth Security Flaw Discovered, Leaves Millions of Devices Vulnerable" by Marius Nestor
- 2019/8/17: Ars Technica, "New Attack exploiting serious Bluetooth weakness can intercept sensitive data" by Dan Goodin
- 2019/8/18: Tech Xplore, "Specification vulnerability in devices that speak Bluetooth is addressed" by Nancy Cohen
- 2019/8/20: Techtarget, "KNOB attack puts all Bluetooth devices at risk" by Michael Heller
- 2019/9/14: CyberWire, "Interview for the CyberWire Research Saturday podcast" by Dave Bittner

Non-English News

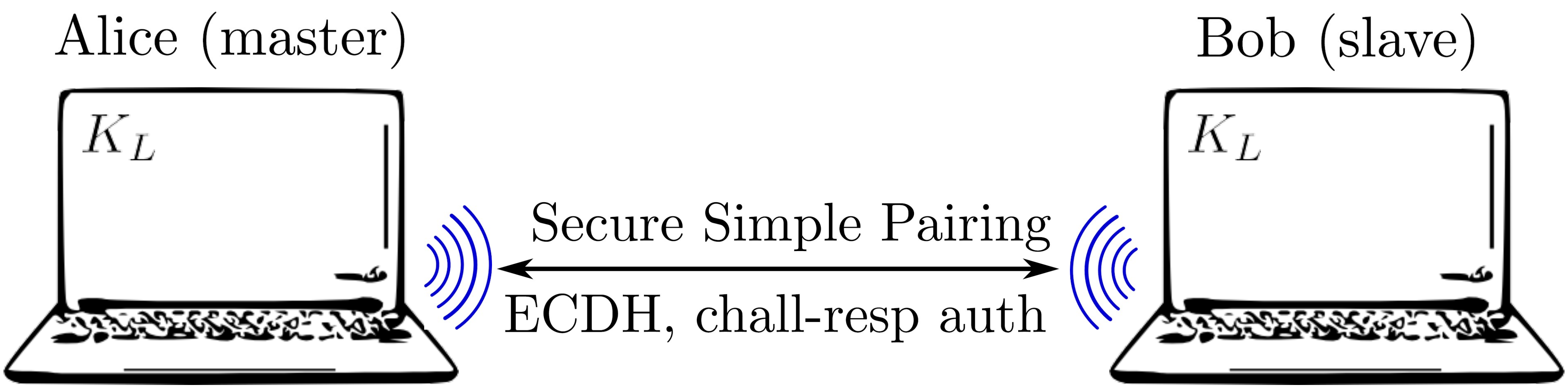
- 2019/8/16: Version2 (Danish), "Sikkerhedshul i Bluetooth lader hackere følge med i dataoverførsler" by Christoffer Elmann Ranhauge
- 2019/8/19: Heise Online (German), "KNOB-Attack: Schwerer Konzeptfehler in Bluetooth" by Dusan Zivadinovic
- 2019/8/22: LA RAZÓN (Spanish), "Los hackers podrían robar datos que se envían vía Bluetooth" by José A. Prados
- 2019/8/26: ANSA (Italian), "Bluetooth, scoperta una falla di sicurezza" by Redazione ANSA
- 2019/8/26: CORCOM (Italian), "Bluetooth, scoperta una “falla”: nel team anche un italiano" by Redazione CORCOM
- 2019/8/27: Corriere della Sera (Italian), "Bluetooth, ricercatore di Pesaro scopre falla nel sistema: «Abbiamo intercettato dati privati»" by Nicola Catenaro
- 2019/8/27: Il Giornale (Italian), "Grave falla nel sistema Bluetooth: ecco cosa si può spiare" by Pina Francone
- 2019/9/17: Wired.it (Italian), "Risolto pericoloso bug di sicurezza nel bluetooth, con un italiano nel team" by Diego Barbera
- 2019/9/21: Start Magazine (Italian), "Chi e come ha scovato una falla nella tecnologia Bluetooth" by Simone Martino

Bluetooth Security Mechanisms

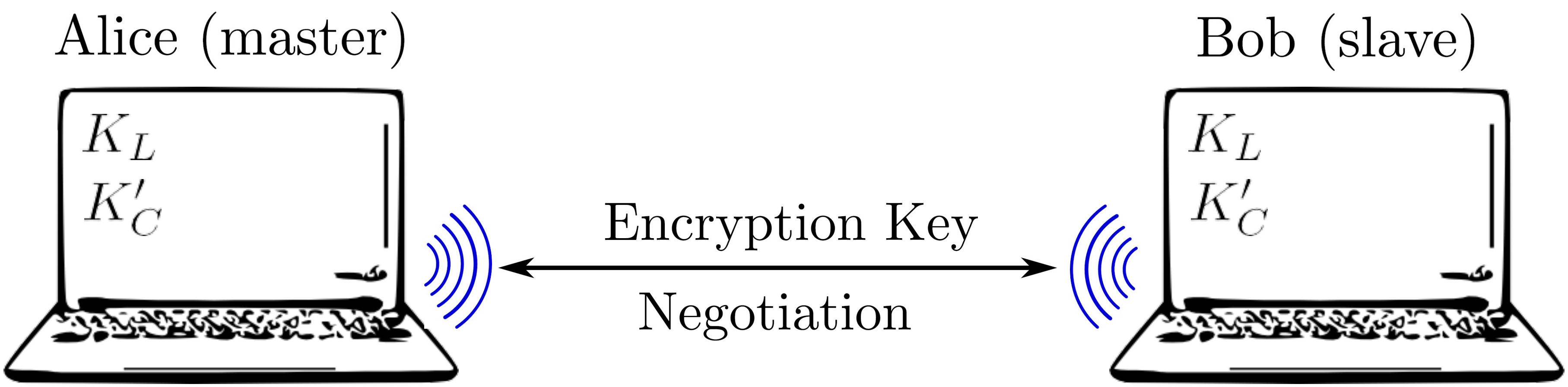
Bluetooth®

Bluetooth Security Mechanisms

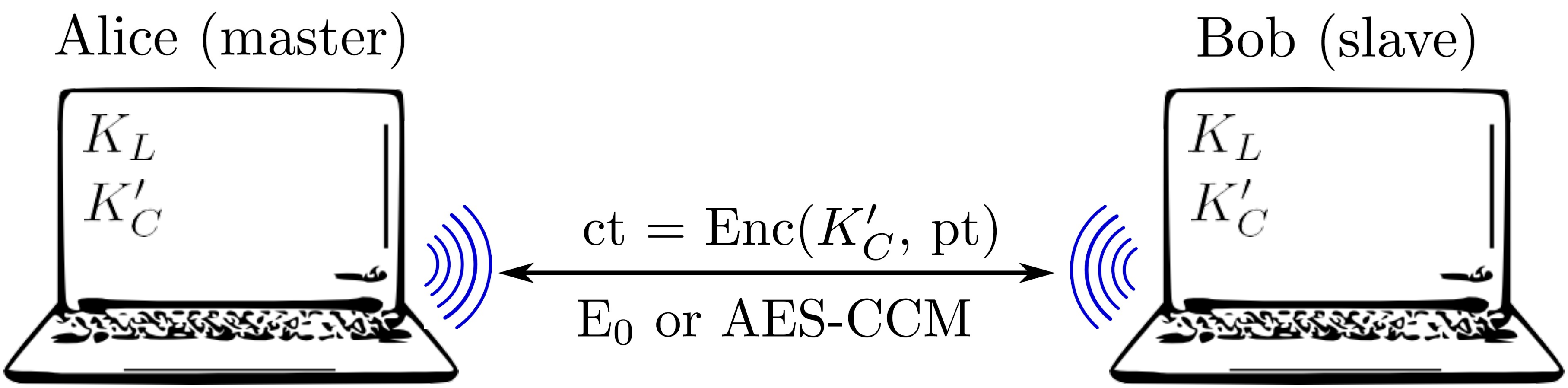
Bluetooth®



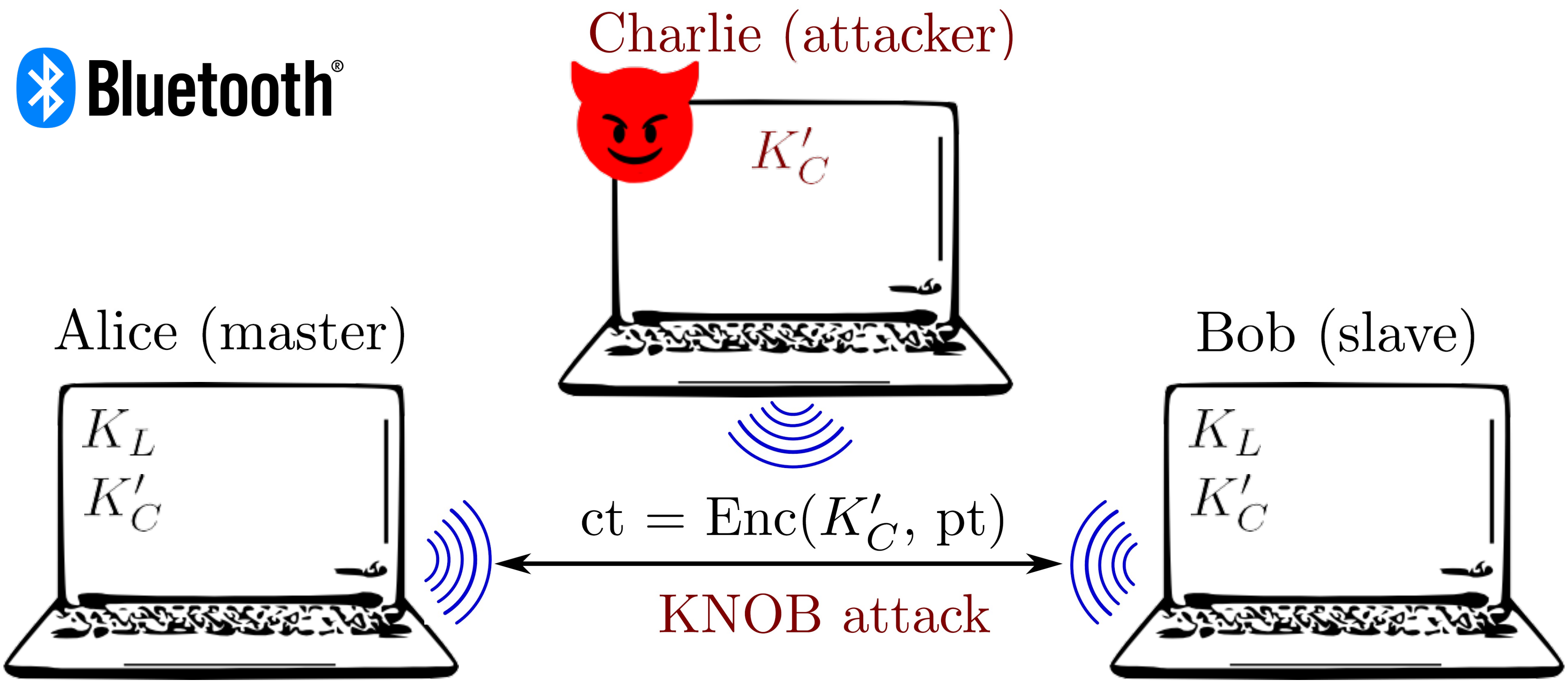
Bluetooth Security Mechanisms



Bluetooth Security Mechanisms

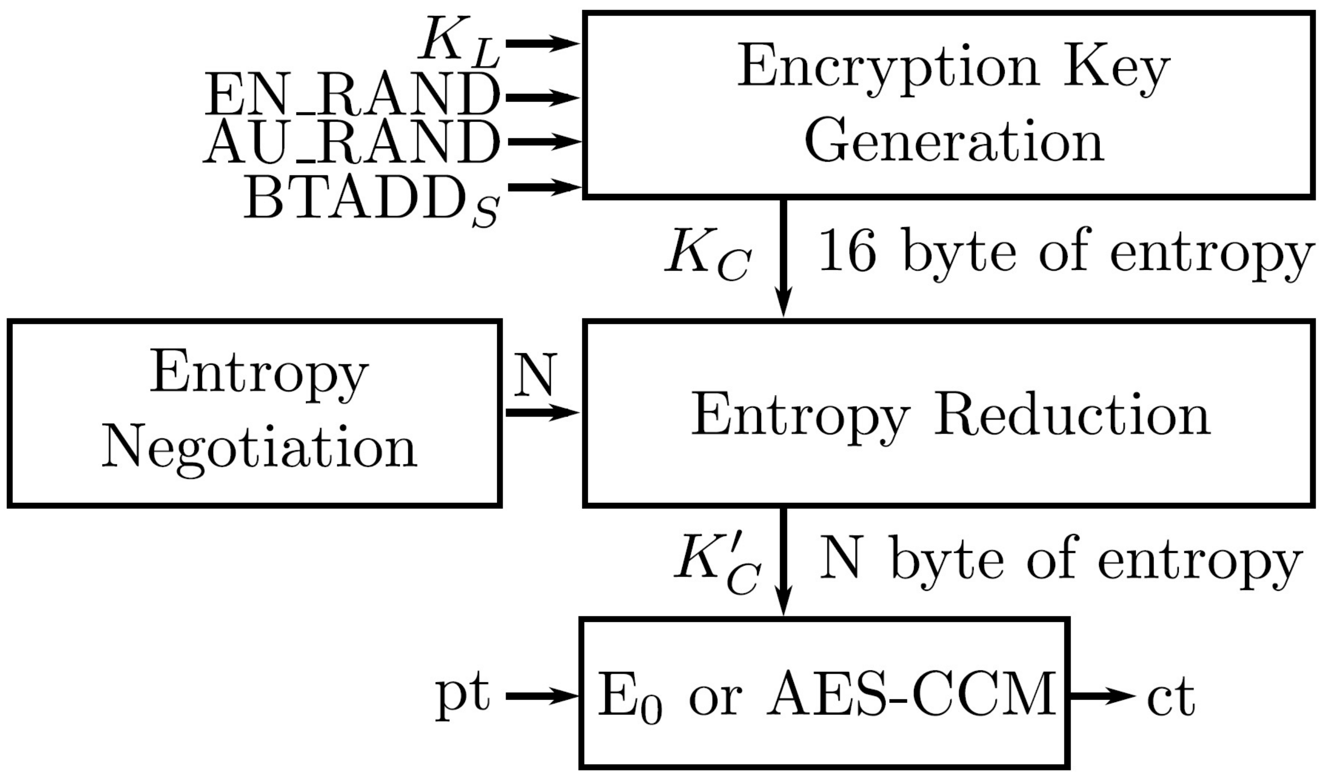


Bluetooth Security Mechanisms



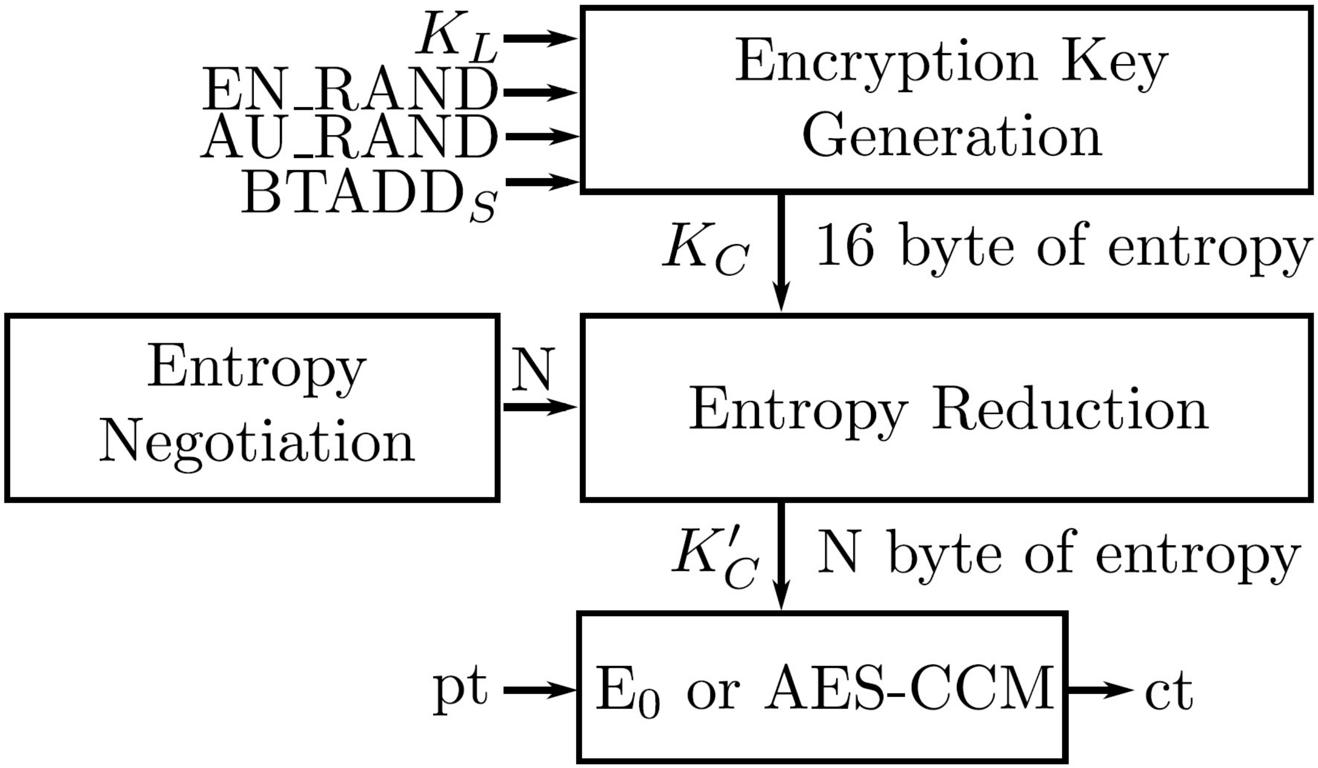
Encryption Key Negotiation Of Bluetooth (KNOB)

- Paired devices negotiate an encryption key (K'_C) upon connection



Encryption Key Negotiation Of Bluetooth (KNOB)

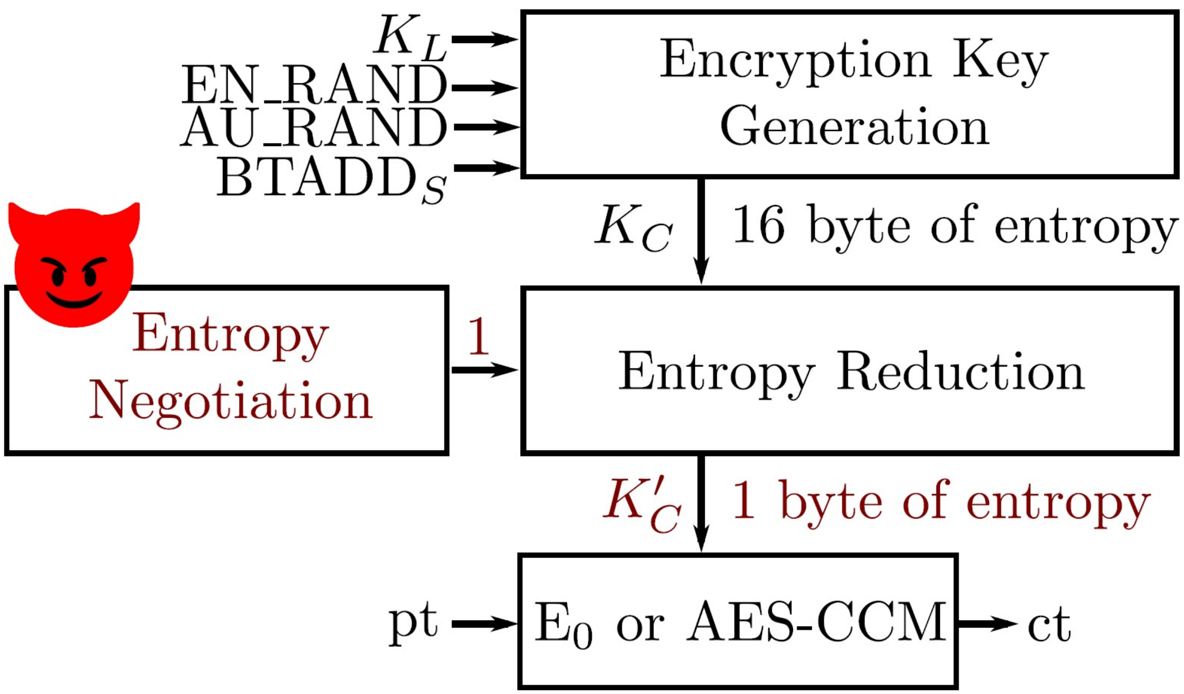
- Paired devices negotiate an encryption key (K'_C) upon connection



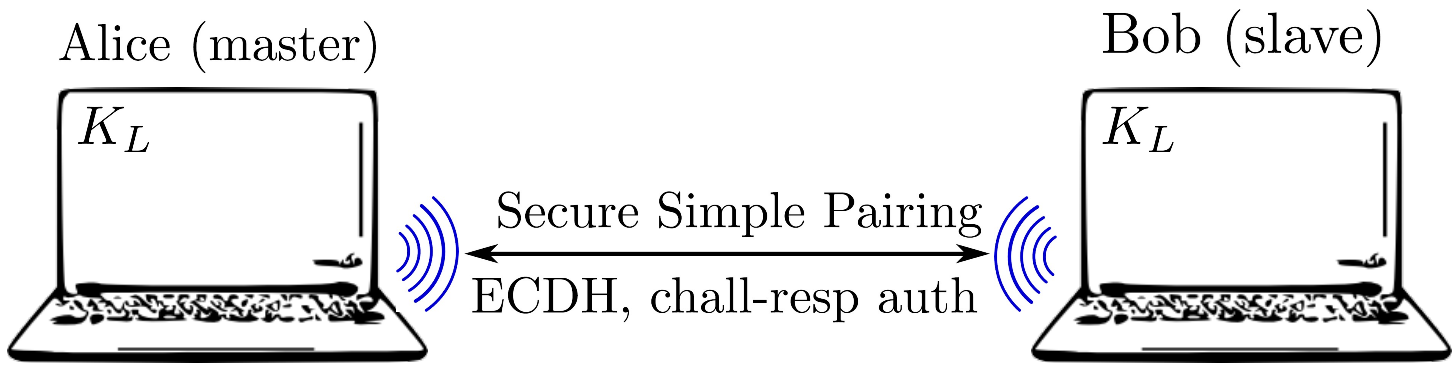
Bluetooth allows K'_C with 1 byte of entropy and does not authenticate Entropy Negotiation

Our Contribution: Key Negotiation Of Bluetooth (KNOB) Attack

- Our **Key Negotiation of Bluetooth (KNOB) attack** sets N=1, and brute forces K'_C
  - Affects *any* standard compliant Bluetooth device (architectural attack)
  - Allows to *decrypt all traffic* and *inject valid traffic*
  - Runs in *parallel* (multiple links and piconets)

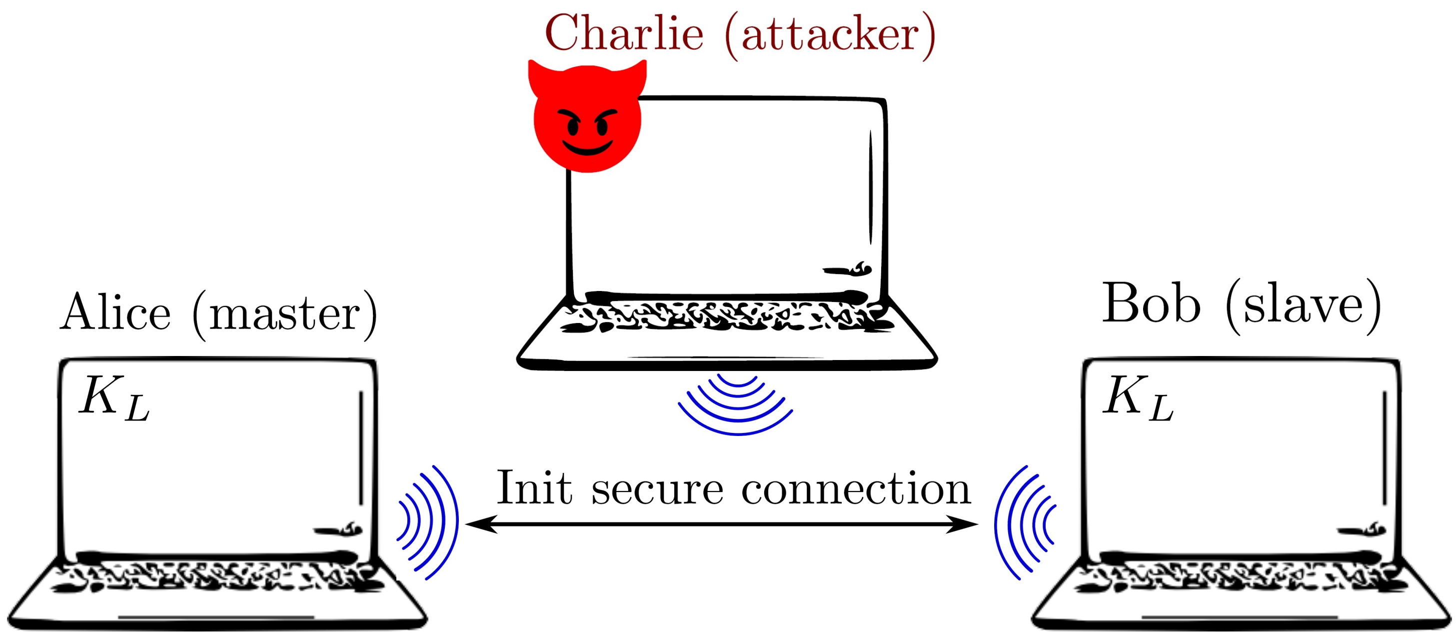


KNOB Attack Stages



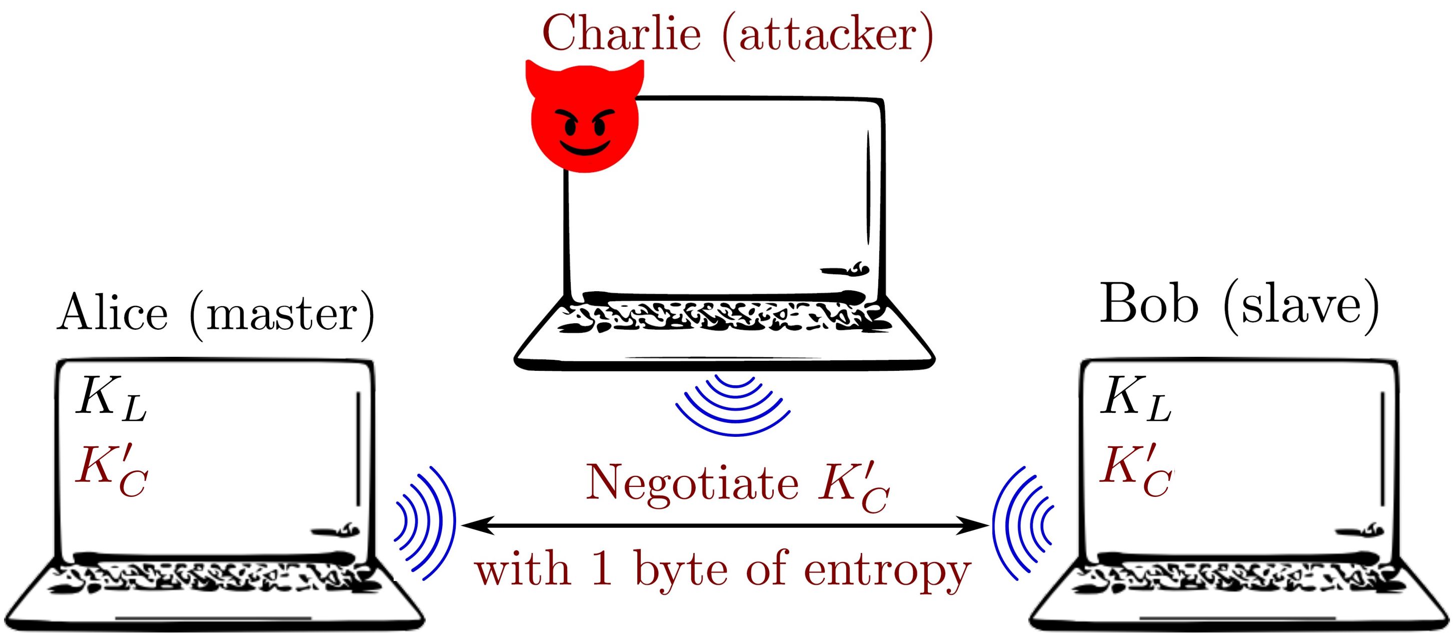
1. Alice and Bob securely pair in absence of Eve

KNOB Attack Stages



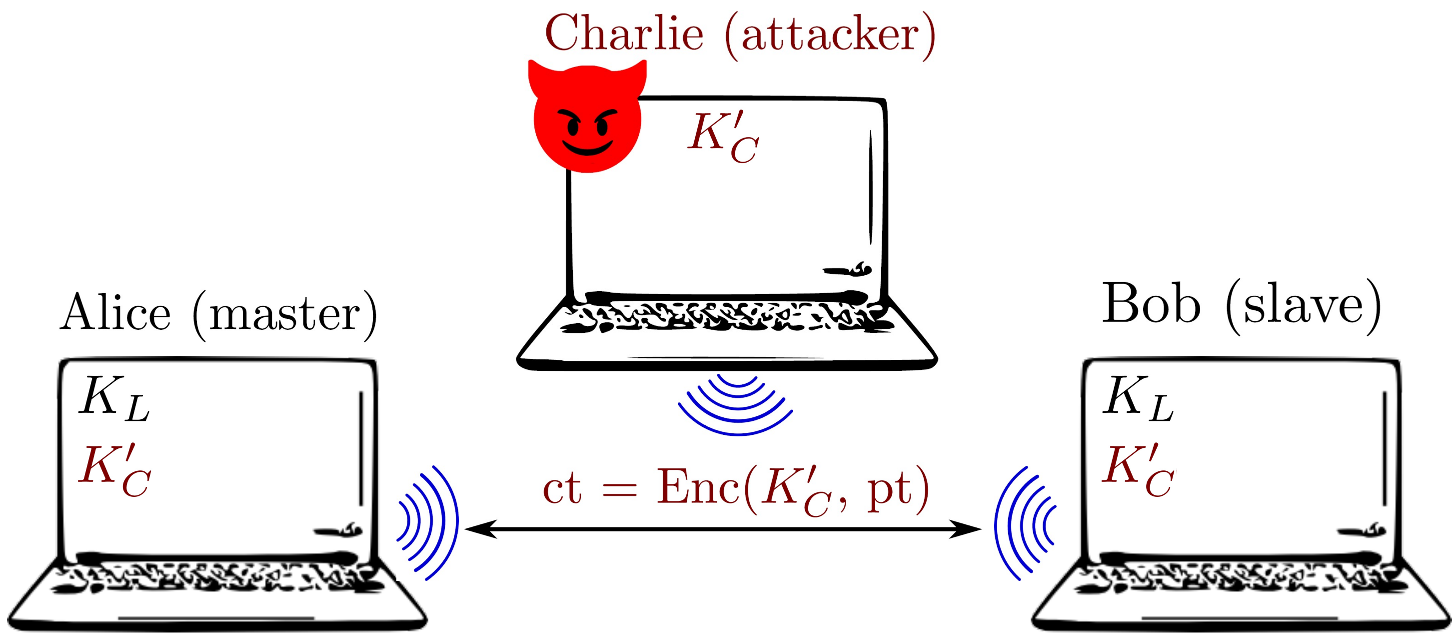
1. Alice and Bob securely pair in absence of Eve
2. Alice and Bob initiate a secure connection

KNOB Attack Stages



1. Alice and Bob securely pair in absence of Eve
2. Alice and Bob initiate a secure connection
3. Charlie makes the victims negotiate an encryption key with 1 byte of entropy

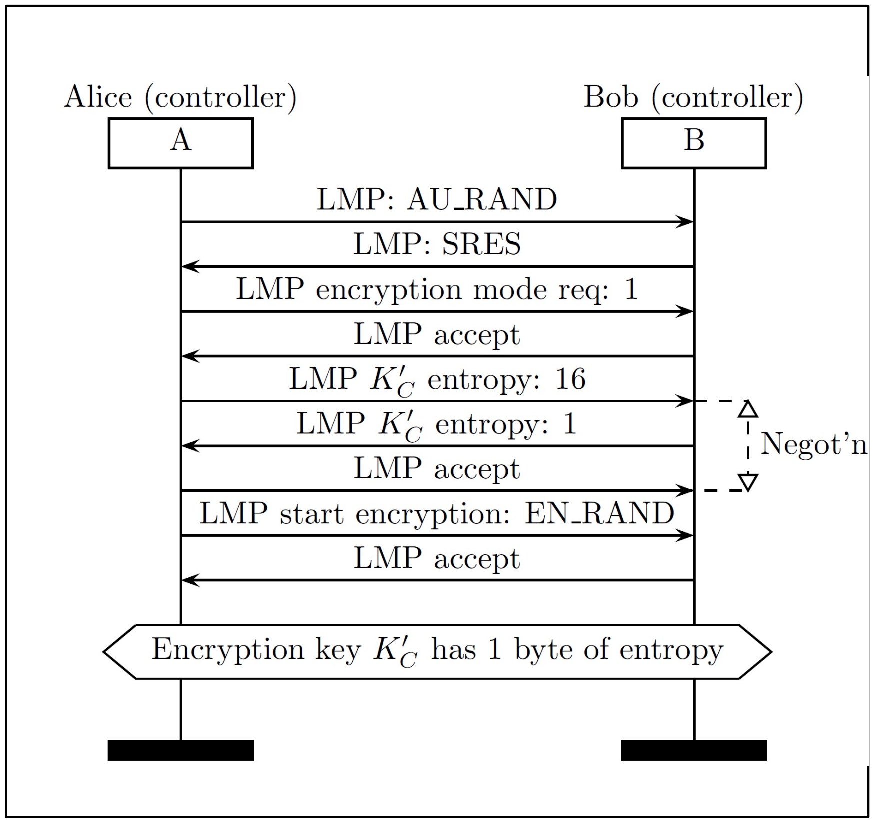
KNOB Attack Stages



1. Alice and Bob securely pair in absence of Eve
2. Alice and Bob initiate a secure connection
3. Charlie makes the victims negotiate an encryption key with 1 byte of entropy
4. Charlie eavesdrop the ciphertext and brute force the key in real time

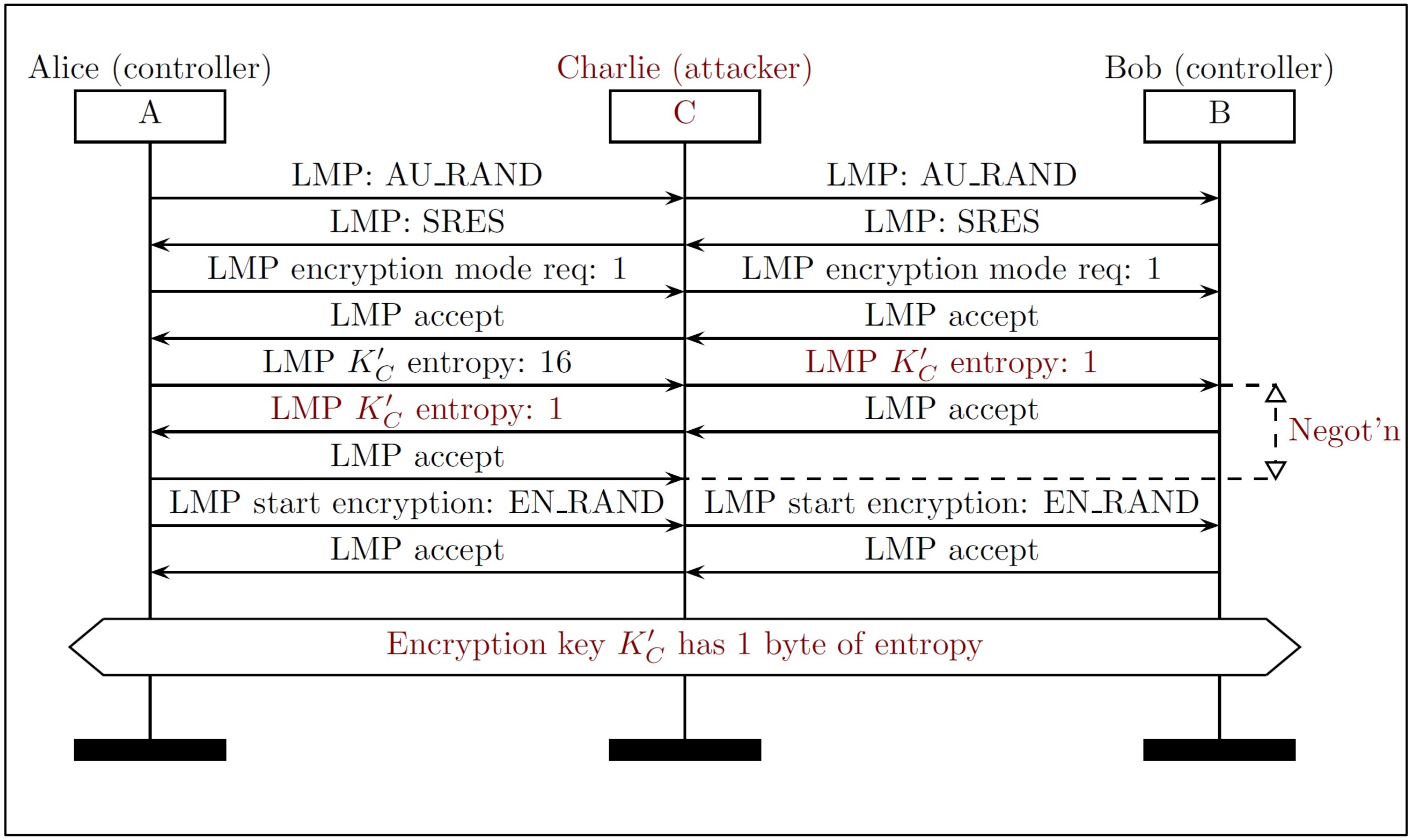
Bluetooth Entropy Negotiation

- Entropy negotiation is **neither integrity protected** nor encrypted
  - N between 1 and 16

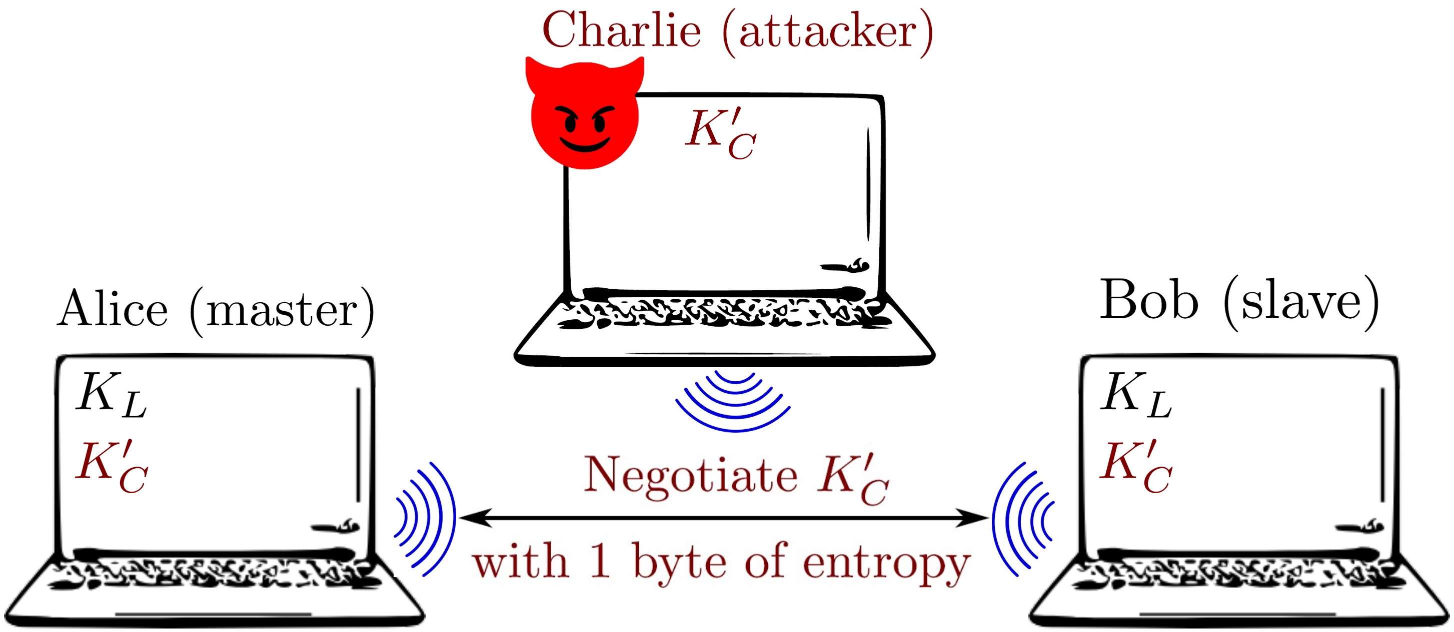


Adversarial Bluetooth Entropy Negotiation

- Charlie sets N=1 (K'_C's entropy), LMP is neither integrity protected nor encrypted



Brute Forcing the Encryption Key (K'_C) in Real Time



- Alice and Bob use an encryption key (K'_C) with 1 Byte of entropy
  - Charlie brute forces K'_C within 256 candidates (in parallel)
- K'_C space when entropy is 1 byte
  - AES-CCM: `0x00 ... 0xff`
  - E_0: (`0x00 ... 0xff`) x `0x00e275a0abd218d4cf928b9bbf6cb08f`

KNOB Attack Scenario

- Attacker decrypts a file exchanged over an encrypted Bluetooth link
  - Victims: Nexus 5 and Motorola G3
  - Attacker: ThinkPad X1 and Ubertooth (Bluetooth sniffer)

Vulnerable chips and devices (Bluetooth 5.0, 4.2)

Bluetooth chip	Device(s)	Vulnerable?
Bluetooth Version 5.0		
Snapdragon 845	Galaxy S9	✓
Snapdragon 835	Pixel 2, OnePlus 5	✓
Apple/USI 339S00428	MacBookPro 2018	✓
Apple A1865	iPhone X	✓
Bluetooth Version 4.2		
Intel 8265	ThinkPad X1 6th	✓
Intel 7265	ThinkPad X1 3rd	✓
Unknown	Sennheiser PXC 550	✓
Apple/USI 339S00045	iPad Pro 2	✓
BCM43438	RPi 3B, RPi 3B+	✓
BCM43602	iMac MMQA2LL/A	✓

✓ = Entropy of the encryption key (K'_C) reduced to 1 Byte

Vulnerable chips and devices (Bluetooth 4.1 and below)

Bluetooth chip	Device(s)	Vulnerable?
Bluetooth Version 4.1		
BCM4339 (CYW4339)	Nexus5, iPhone 6	✓
Snapdragon 410	Motorola G3	✓
Bluetooth Version ≤ 4.0		
Snapdragon 800	LG G2	✓
Intel Centrino 6205	ThinkPad X230	✓
Chicony Unknown	ThinkPad KT-1255	✓
Broadcom Unknown	ThinkPad 41U5008	✓
Broadcom Unknown	Anker A7721	✓
Apple W1	AirPods	*

✓ = Entropy of the encryption key (K'_C) reduced to 1 Byte
* = Entropy of the encryption key (K'_C) reduced to 7 Byte

KNOB in Bluetooth core spec v5.0 (page 1650)

*"For the encryption algorithm, **the key size (N) may vary between 1 and 16 octets (8-128 bits)**. The size of the encryption key is configurable for two reasons. The first has to do with the many **different requirements imposed on cryptographic algorithms in different countries** - both with respect to export regulations and official attitudes towards privacy in general. The second reason is to **facilitate a future upgrade** path for the security without the need of a costly redesign of the algorithms and encryption hardware; **increasing the effective key size is the simplest way to combat increased computing power at the opponent side**."*

https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=421043

KNOB Attack Disclosure and Countermeasures

- Responsible disclosure with CERT and Bluetooth SIG (`CVE-2019-9506`)
  - KNOB discovery in May 2018, exploitation and report in October 2018
  - Many industries affected, e.g., Intel, Broadcom, Qualcomm, ARM, and Apple
- *Legacy compliant* countermeasures
  - Set 16 bytes of entropy in the Bluetooth firmware
  - Check N from the host (OS) upon connection
  - Security mechanisms on top of the link layer
- *Non legacy compliant* countermeasures
  - Secure entropy negotiation with K_L (ECDH shared secret)
  - Get rid of the entropy negotiation protocol

Related Work

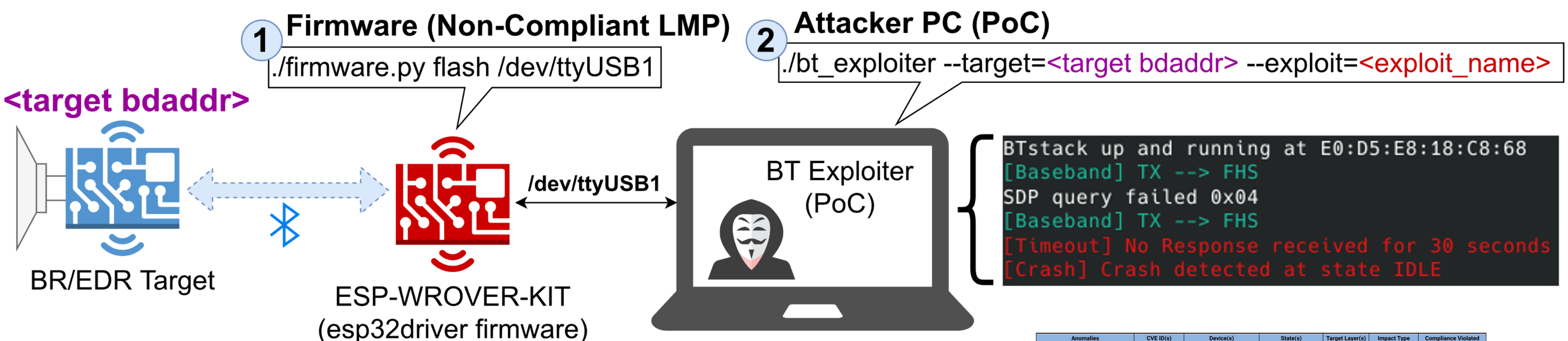
The security and privacy guarantees of Bluetooth were studied since Bluetooth v1.0.

- Several attacks on the Secure Simple Pairing (SSP) protocol
- Several attacks on various implementations of Bluetooth (Android, iOS, Windows, Linux)
- Several attacks on security of the ciphers used by Bluetooth

→ KNOB attack works regardless of security guarantees / target platform / cipher !

Follow Up Study

Matheus E. Garbelini et al., **"BRAKTOOTH: Causing Havoc on Bluetooth Link Manager"** (White Paper 2021), https://asset-group.github.io/disclosures/braktooth/



BT SoC Vendor	BT SoC	Dev. Kit / Product	Sample Code
Bluetooth 5.2			
Intel	AX200	Laptop Forge15-R	N.A
Qualcomm	WCN3990	Xioami Pocophone F1	N.A
Bluetooth 5.1			
Texas Instruments	CC2564C	CC256XCQFN-EM	SPPDMMultiDemo
Zhuhai Jieli Technology	AC6366C	AC6366C_DEMO_V1.0	app_keyboard
Bluetooth 5.0			
Cypress	CYW20735B1	CYW920735Q60EVB-01	rfcomm_serial_port
Bluetrum Technology	AB5301A	AB32VG1	Default
Zhuhai Jieli Technology	AC6925C	XY-WRBT Module	N.A
Actions Technology	ATS281X	Xiaomi MDZ-36-DB	N.A
Bluetooth 4.2			
Zhuhai Jieli Technology	AC6905X	BT Audio Receiver	N.A
Espressif Systems	ESP32	ESP-WROVER-KIT	bt_spp_acceptor
Bluetooth 4.1			
Harman International	JX25X	JBL TUNE500BT	N.A
Bluetooth 4.0			
Qualcomm	CSR 8811	Laird DVK-BT900-SA	vspspp.server.at
Bluetooth 3.0 + HS			
Silabs	WT32i	DKWT32I-A	ai-6.3.0-1149

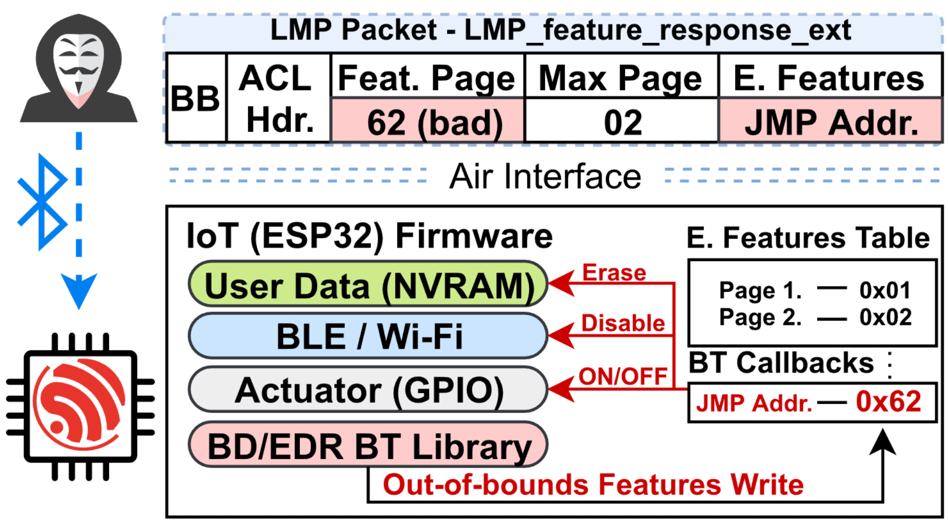
Anomalies	CVE ID(s)	Device(s)	State(s)	Target Layer(s)	Impact Type	Compliance Violated
8.1 **V1** Feature Pages Execution	CVE-2021-28139	ESP-WROVER-KIT	*Feature Exchange*	LMP	ACE / Deadlock	[V.1] Part E, Sec. 2.7
8.2 **V2** Truncated SCO Link Request	CVE-2021-34144	AC6366C_DEMO_V1.0	***After Paging***	LMP	Deadlock	[V.2] Part E, Sec. 2.7
8.3 **V3** Duplicated IOCAP	CVE-2021-28136	ESP-WROVER-KIT	*Bounding*	LMP	Crash	[V.2] Part C, Sec. 4.2.7.1
8.4 **V4** Feature Resp. Flooding	CVE-2021-28135 CVE-2021-28155 CVE-2021-31717	ESP-WROVER-KIT JBL TUNE500BT Xiaomi MDZ-36-DB	***After Paging***	LMP	Crash	[V.1] Part E, Sec. 2.7
8.5 **V5** LMP Auto Rate Overflow	CVE-2021-31609 CVE-2021-31612	DKWT32I-A BT Audio Receiver	Data Rate Change	Baseband	Crash	[V.2] Part B, Sec. 6.6.2
8.6 **V6** LMP 2-DH1 Overflow	CVE-2021-35093	DVK-BT900-SA	After EDR Change	Baseband	Deadlock	[V.2] Part C, Sec. 2.3
8.7 **V7** LMP DM1 Overflow	CVE-2021-34150	AB32VG1	***Many***	Baseband	Deadlock	[V.2] Part B, Sec. 6.5.4.1
8.8 **V8** Truncated LMP Accepted	CVE-2021-31613	BT Audio Receiver XY-WRBT Module	***Many***	LMP	Crash	[V.2] Part C, Sec. 5.1
8.9 **V9** Invalid Setup Complete	CVE-2021-31611	BT Audio Receiver XY-WRBT Module	*Feature Exchange*	LMP	Deadlock	[V.1] Part E, Sec. 2.7
8.10 **V10** Host Conn. Flooding	CVE-2021-31785 CVE-2022-20021	Xiaomi MDZ-36-DB **Mediatek Undisclosed**	*Host Connection*	LMP	Deadlock	[V.1] Part E, Sec. 2.7
8.11 **V11** Same Host Connection	CVE-2021-31786 CVE-2022-20022	Xiaomi MDZ-36-DB **Mediatek Undisclosed**	*Host Connection*	LMP	Deadlock	[V.1] Part E, Sec. 2.7
8.12 **V12** AU Rand Flooding	CVE-2021-31610 CVE-2021-34149 CVE-2021-34146 CVE-2021-34143 CVE-2022-20023	AB32VG1 CC256XCQFN-EM CYW920735Q60EVB AC6366C_DEMO_V1.0 **Mediatek Undisclosed**	***After Paging***	LMP	Crash Deadlock	[V.1] Part E, Sec. 2.7
8.13 **V13** Invalid Max Slot Type	CVE-2021-34145	CYW920735Q60EVB	*After Setup Complete*	Baseband	Crash	[V.1] Part E, Sec. 2.7
8.14 **V14** Max Slot Length Overflow	CVE-2021-34148	CYW920735Q60EVB	*After Setup Complete*	Baseband	Crash	[V.1] Part E, Sec. 2.7
8.15 **V15** Invalid Timing Accuracy	CVE-2021-34147 CVE-2021-30348 Pending	CYW920735Q60EVB Pocophone F1 (WCN3990) Intel AX200	*Timing Accuracy*	LMP, Baseband	Crash	[V.1] Part E, Sec. 2.7
8.16 **V16** Paging Scan Deadlock	Pending	Intel AX200	*After Host Connection*	LMP, Baseband	Deadlock	[V.1] Part E, Sec. 2.7
A1 Accepts Lower LMP Length	N.A	**All tested, expect ESP32**	***Many***	Baseband	Non-Compliance	[V.2] Part C, Sec. 5.1
A2 Accepts Higher LMP Length	N.A	**All tested devices**	***Many***	Baseband	Non-Compliance	[V.2] Part C, Sec. 5.1
A3 Allows Multiple Encryption Start	N.A	Xiaomi MDZ-36-DB	*After Encryption Start*	LMP	Non-Compliance	[V.2] Part C, Sec. 4.2.5.3
A4 Ignored Role Switch Reject	N.A	Pocophone F1 (WCN3990)	*Role Switch*	LMP	Non-Compliance	[V.2] Part C, Sec. 4.4.2
A5 Invalid Response	N.A	Intel AX200 DVK-BT900-SA	*Feature Exchange*	LMP	Non-Compliance	[V.2] Part C, Sec. 4.3.4
A6 Unexpected Encryption Stop	N.A	CYW920735Q60EVB	*After Encryption Start*	LMP	Non-Compliance	[V.2] Part C, Sec. 4.2.5.4

Follow Up Study

Matheus E. Garbelini et al., **“BRAKTOOTH: Causing Havoc on Bluetooth Link Manager”** (White Paper 2021), https://asset-group.github.io/disclosures/braktooth/

Impact of BrakTooth

- Arbitrary Code Execution in IoTs
- DoS in Laptops & Smartphones
- Freezing Audio Products



Conclusion

We propose the **Key Negotiation Of Bluetooth (KNOB)** attack

- Reduces the entropy of any encryption key to 1 Byte, and brute forces the key
- Affects *any* standard compliant Bluetooth device (architectural attack)
- Allows to *decrypt all traffic* and *inject valid traffic*
- Runs in *parallel* (multiple links and piconets)

We implement and evaluate the KNOB attack

- 14 vulnerable chips (Intel, Broadcom, Apple, and Qualcomm)
- 21 vulnerable devices

Provide effective legacy and non legacy compliant countermeasures

For more information visit: `https://knobattack.com`

Q&A

안준호 (Best Question)

- Why Entropy Negotiation and LMP is neither integrity protected nor encrypted?

    - *Key negotiation → encryption and authentication*
    - *Export regulations (Maybe)*

Q&A

이용화 (Best Question)

- Not limited to the vulnerability in this paper(KNOB), we all know that in every security field including IoT, Wireless network, Embedded systems, etc, vulnerabilities found in firmwares are problematic because of the hardness of the post-handling process after the discovery. It is hard to patch, update, and fix the bug in those cases. I hope to know if there is a efficient way to solve the problem.

    - *Underexplored hardware vulnerability*
    - *No efficient way*

Q&A

김한나

- In this paper, it seems to they success to attack with N = 7 (W1 case). I wonder what is the minimum l
ength of entropy to defend the attack.

  - *A minimum encryption key length → 7 bytes by the Bluetooth SIG*
  - *However, 128bit recommend by NIST*