USENIX 2019 @ Santa Clara, US

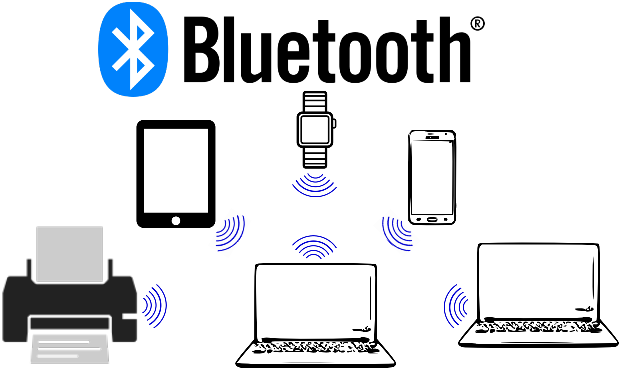
The KNOB is Broken: Exploiting Low Entropy in the Encryption Key Negotiation Of Bluetooth BR/EDR

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

Bluetooth

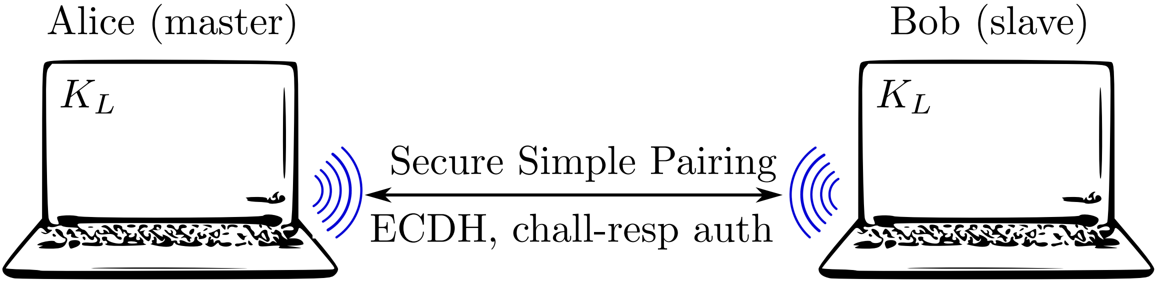
- Bluetooth (BR/EDR or Classic)
  - Pervasive wireless technology for personal area networks
  - E.g., mobile, automotive, medical, and industrial devices
- Bluetooth uses custom security mechanisms (at the link layer)
  - Open but complex specification
  - No public reference implementation



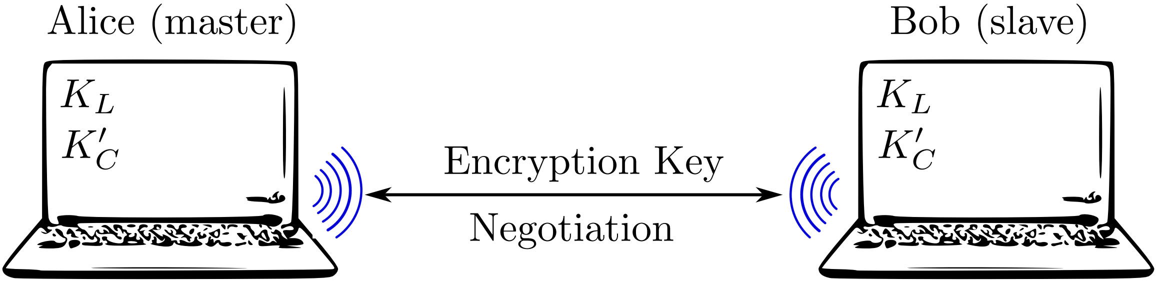
Bluetooth Security Mechanisms

Bluetooth Security Mechanisms

Bluetooth®

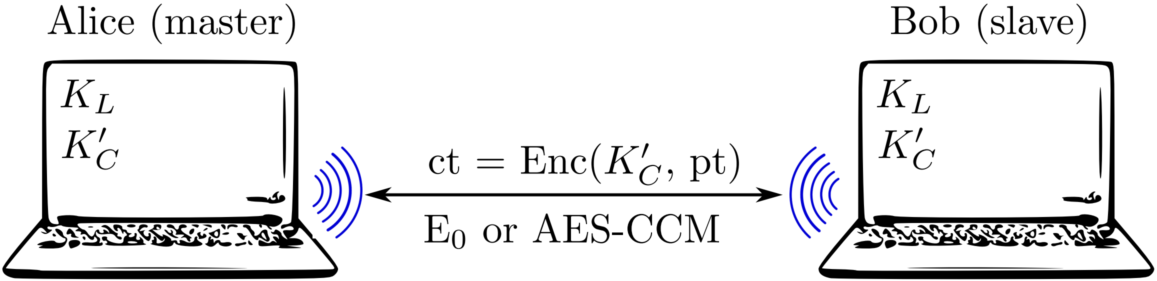


Bluetooth Security Mechanisms



Bluetooth Security Mechanisms

Bluetooth®



Alice (master)

K_L

K'_C

Bob (slave)

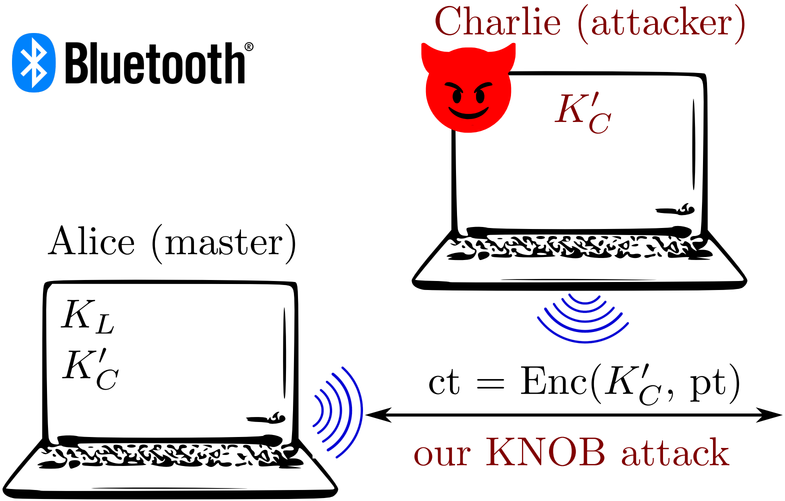
K_L

K'_C

$\mathrm{ct} = \mathrm{Enc}(K'_C, \mathrm{pt})$

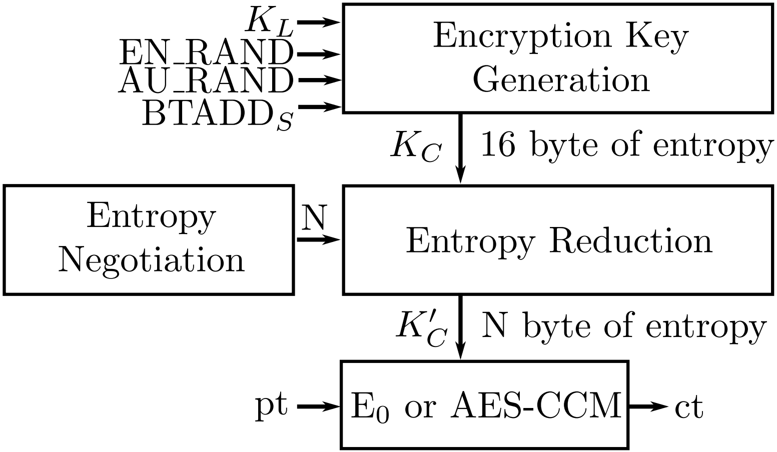
E_0 or AES-CCM

Bluetooth Security Mechanisms



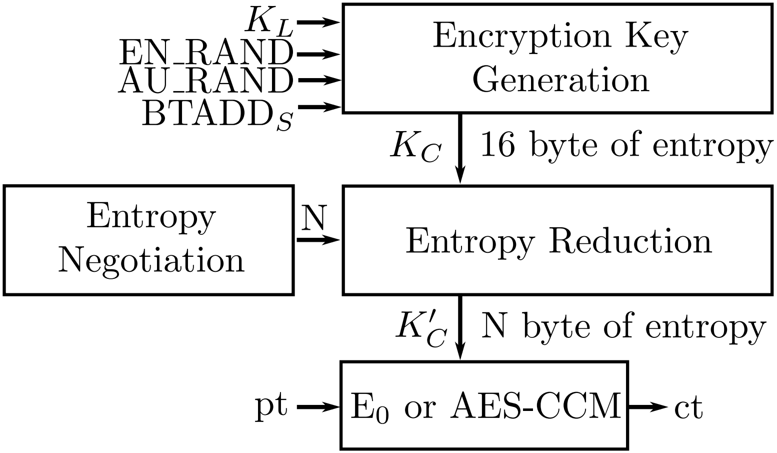
Encryption Key Negotiation Of Bluetooth (KNOB)

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



Encryption Key Negotiation Of Bluetooth (KNOB)

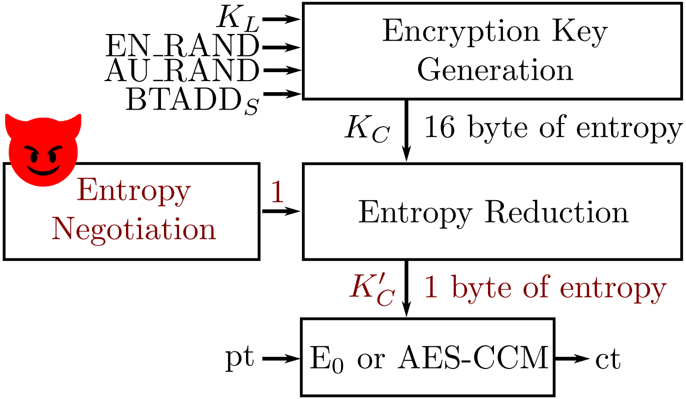
- Paired devices negotiate an encryption key (K'_C) upon connection



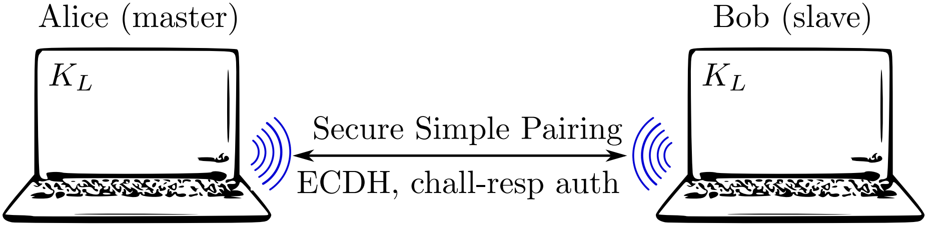
Bluetooth allows K'_C with 1 byte of entropy and does not authenticate Entropy Negotiation

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)

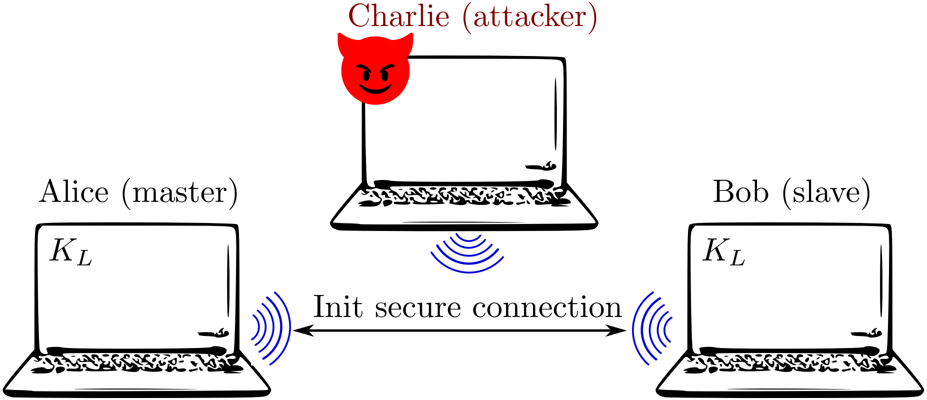


KNOB Attack Stages



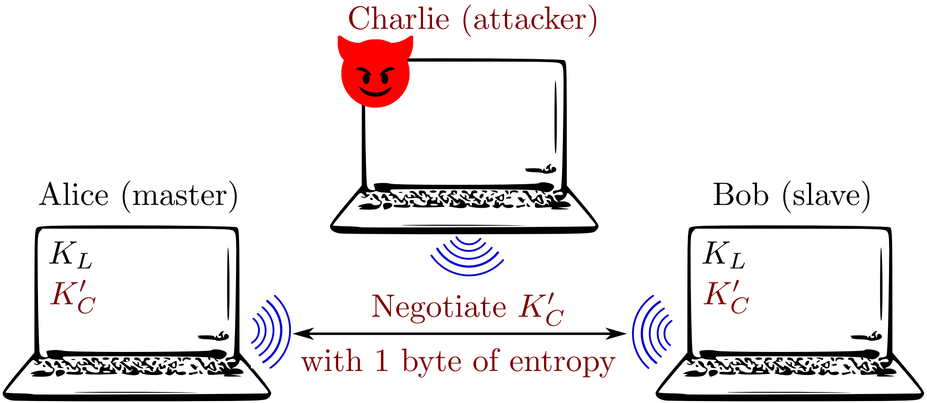
1 Alice and Bob securely pair in absence of Eve

KNOB Attack Stages



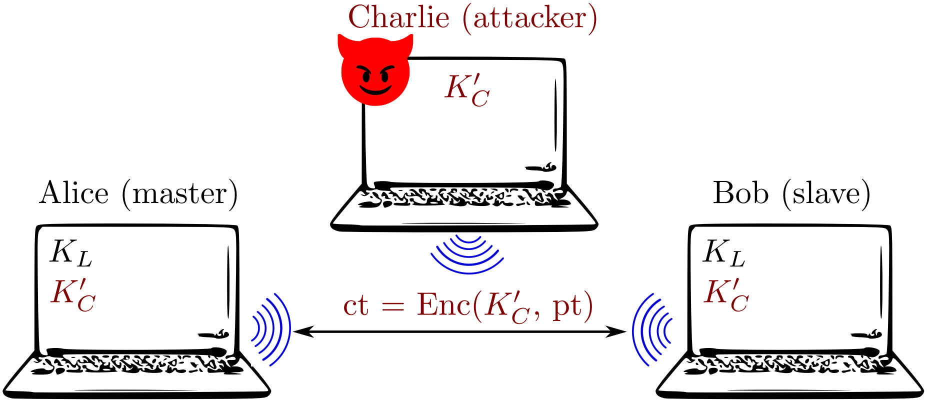
1. Alice and Bob securely pair in absence of Eve
2. Alice and Bob initiate a secure connection

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

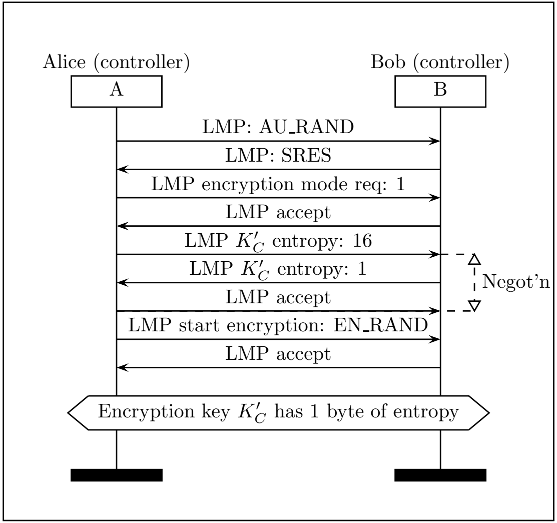
KNOB Attack Stages



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

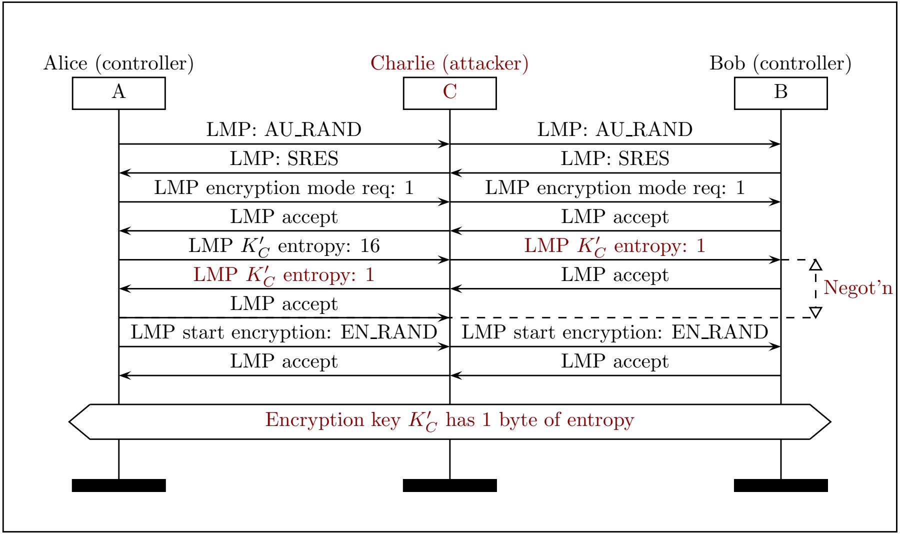
Bluetooth Entropy Negotiation

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

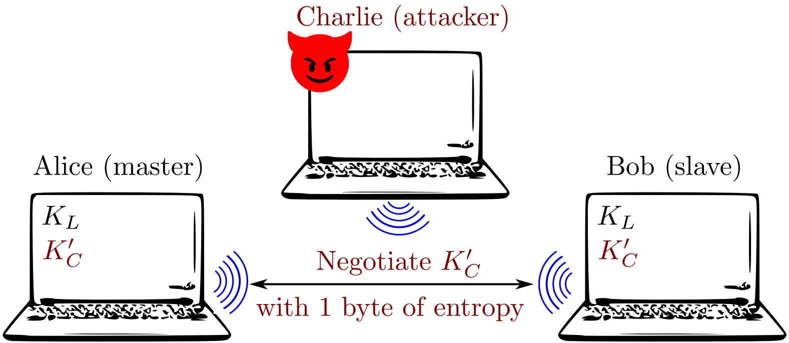


Adversarial Bluetooth Entropy Negotiation

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

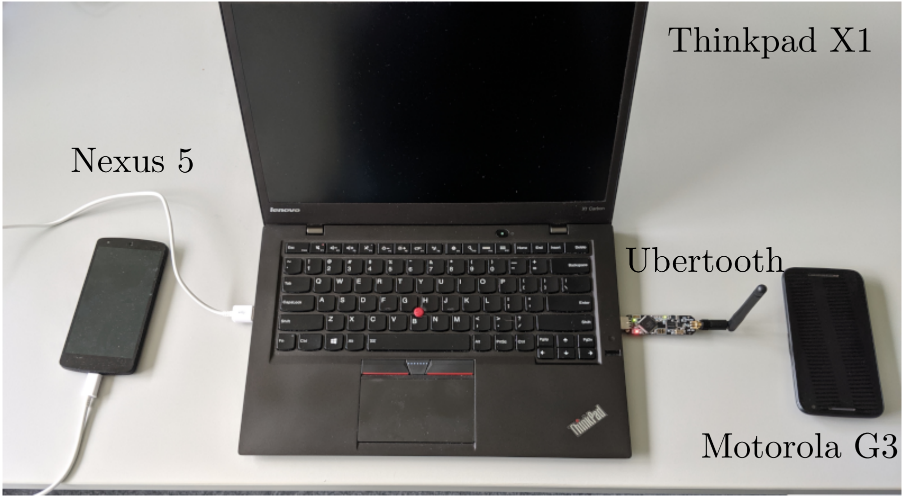


Brute Forcing the Encryption Key (K'_C) in Real Time



- 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 $\leq$ *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

- We did 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

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`

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`

Thanks for your time! Questions?