The Zen of Bluetooth Security

Daniele Antonioli
EURECOM (FR)

ACM WiSec 2026 Keynote
Saarbrücken, Germany

Talk Outline

1. Why the Zen of Bluetooth Security (ZOBS)
2. Bluetooth Introduction
3. 12 ZOBS principles (protocol edition)
4. 2 Bluetooth Research Directions
5. Acks and Conclusion

Why the Zen of Bluetooth Security (ZOBS)

The Zen of Python (PEP 20, wiki, song)

19 guiding principles for the **design of Python** by Tim Peters!


```
m4dc0d3r@xubuntu:~$ python
Python 3.6.9 (default, Nov  7 2019, 10:44:02)
[GCC 8.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import this
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
>>>
```

The Zen of Bluetooth Security (ZOBS)

- 12 principles on Bluetooth Security Protocols
  - Simple, intuitive, complementary, and clear
  - **Attack** and **Defense**
  - Bluetooth Classic and Bluetooth Low Energy
- Covering 6 papers from last 7 years (2019--2026)
  - KNOB, KNOB-BLE, BIAS, BLUR, BLUFFS, BLERP
  - Average 2 principles per paper

Why ZOBS?

- New content for WiSec keynote!
- A way to revisit and crystallize related papers
- Create reusable and concise knowledge
- Stimulate more Bluetooth security research!

Do it for your papers!

Five ZOBS Types

1. Key Entropy
2. Device Authentication
3. Cross-Transport
4. Forward and Future Secrecy
5. Re-pairing

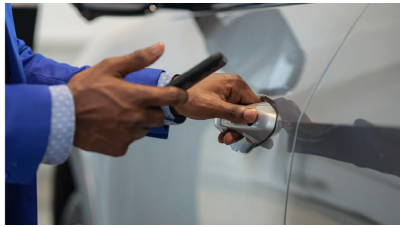
More to come!

We Need a Better Zen-friendly Logo!

Bluetooth Introduction

Bluetooth Overview [ref]

- 2.4 GHz (ISM) band
- Bluetooth Classic (BC)
- Bluetooth Low Energy (BLE)
- Short and long range (30m to 1Km)
- Connected (audio, data, I/O, access, distance, …)
- Connectionless (presence, findmy, …)
- Billions of Bluetooth devices



Bluetooth Specification [ref]

- Core Specification v6.3 (html, pdf)
- BT SIG: advance, protect, promote
- Layers (PHY, link, logical, …)
- Security protocols (pairing, session)
- Components (Host and Controller)

Bluetooth Core Specification

***Bluetooth*® Specification**

- **Version:** v6.3
- **Version Date:** 2026-05-05
- **Prepared By:** Core Specification Working Group

Abstract

This specification defines the technologies required to create interoperable Bluetooth devices.

1 vulns in the spec → billions of exploitable device

Bluetooth Security Model is Unique

Negotiable **security modes** (SC, LSC), **authentication** (MitM protection), and **key strength (entropy)**.

No certificates and PKI. **Pair once** (PK) with **user assisted auth. Cross-transport** pairing. **Re-pairing**.

Automatic secure sessions (fresh SKs) with explicit or implicit authentication using PK.

ZOBS principles are covering these Bluetooth quirks!

Bluetooth Protocol-Level Threat Model

Victims: Alice and Bob (any BC or BLE device)

Roles: Central, Peripheral,

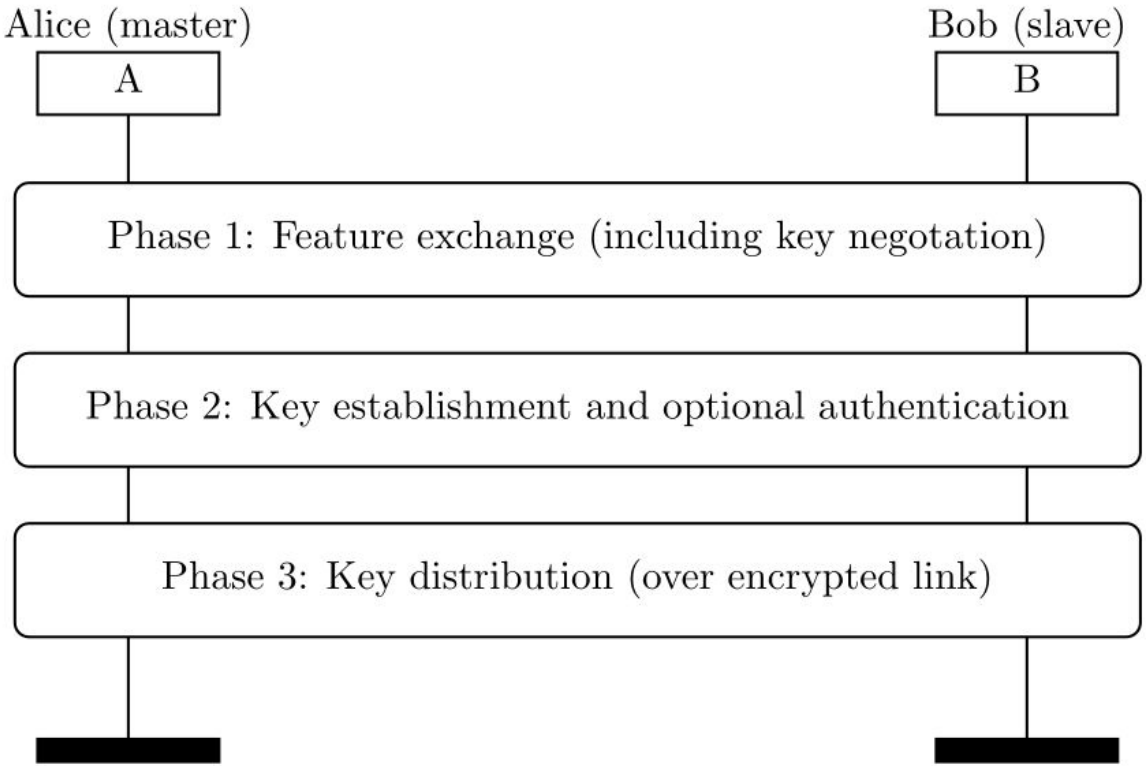
Charlie: Protocol-level Dolev-Yao attacker

Goals: Central and Peripheral Impersonation (CI, PI), MitM, Eavesdropping

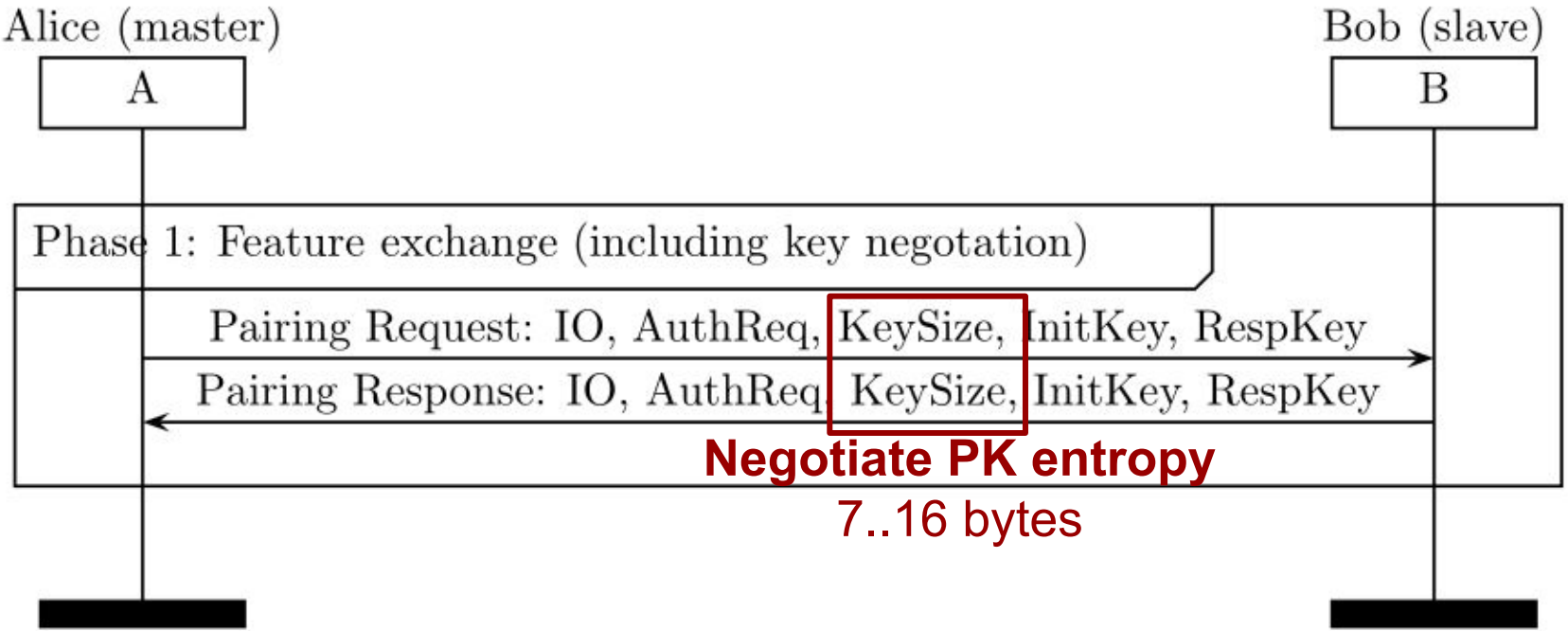
Huge impact: Exploit device regardless of HW, SW, BT version, security mode, …

ZOBS: Key Entropy

BLE Pairing Phases



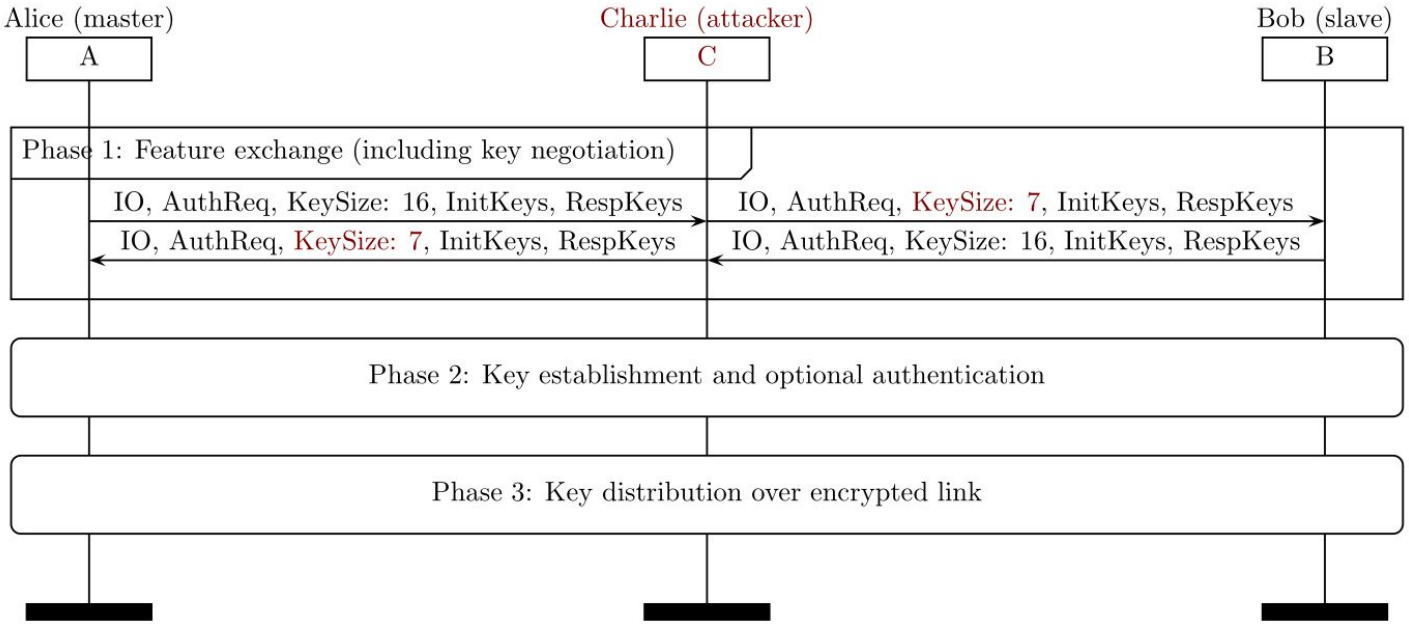
BLE Pairing Feature Exchange



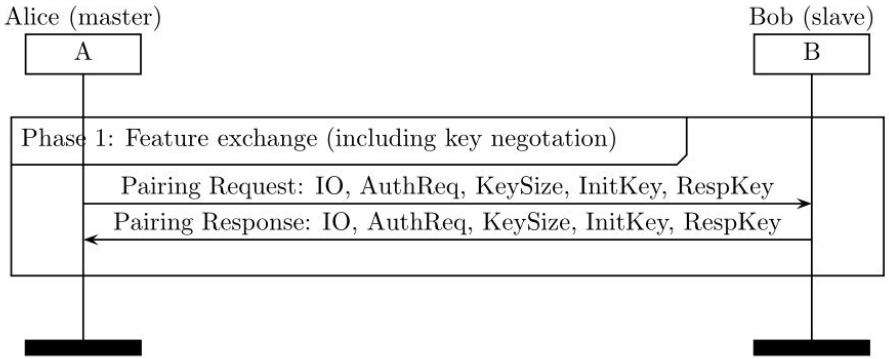
$\mathbf{ZOBS}_1$: Remove Key Entropy Reduction

KNOB MitM on BLE Pairing [TOPS'20]

SK entropy is **7 Byte**! Brute-force a DES key



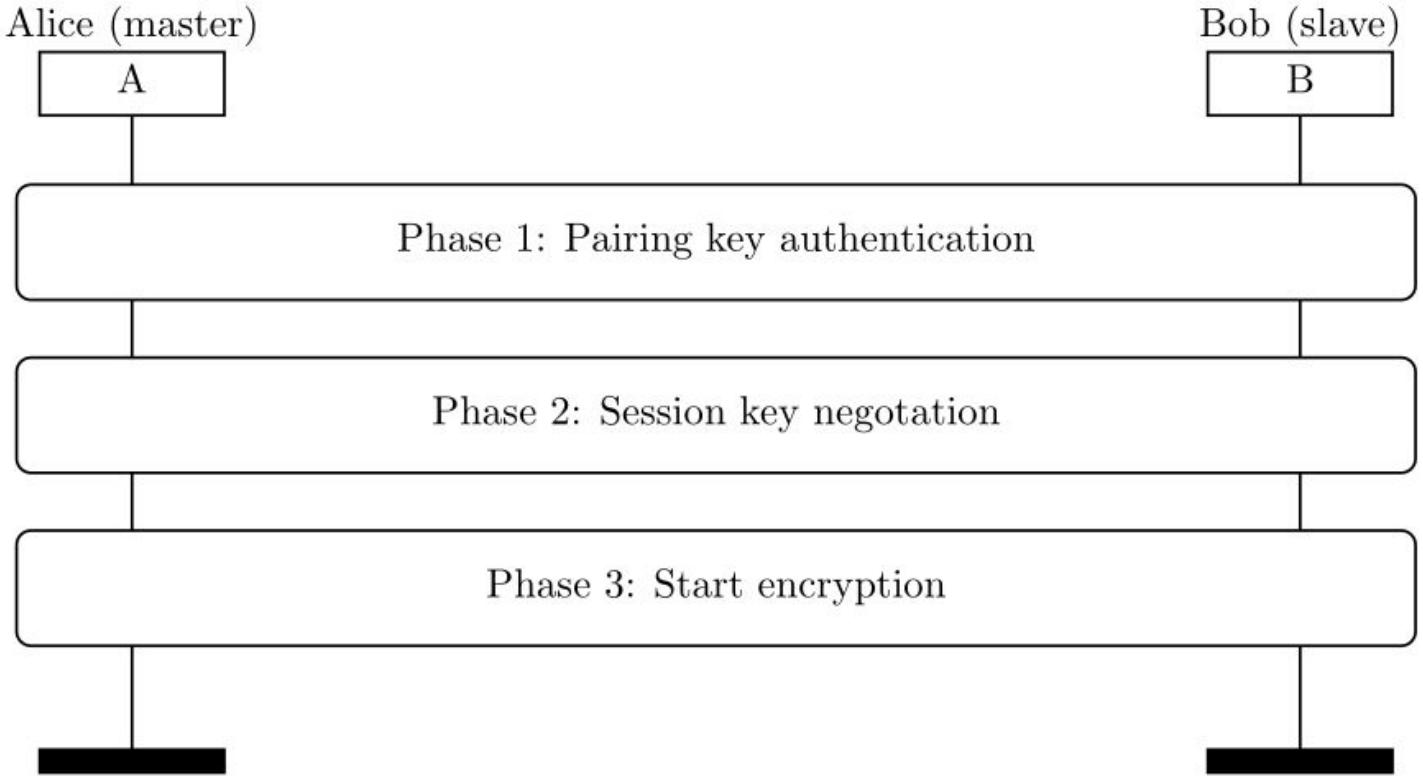
ZOBS$_1$: Remove Key Entropy Reduction



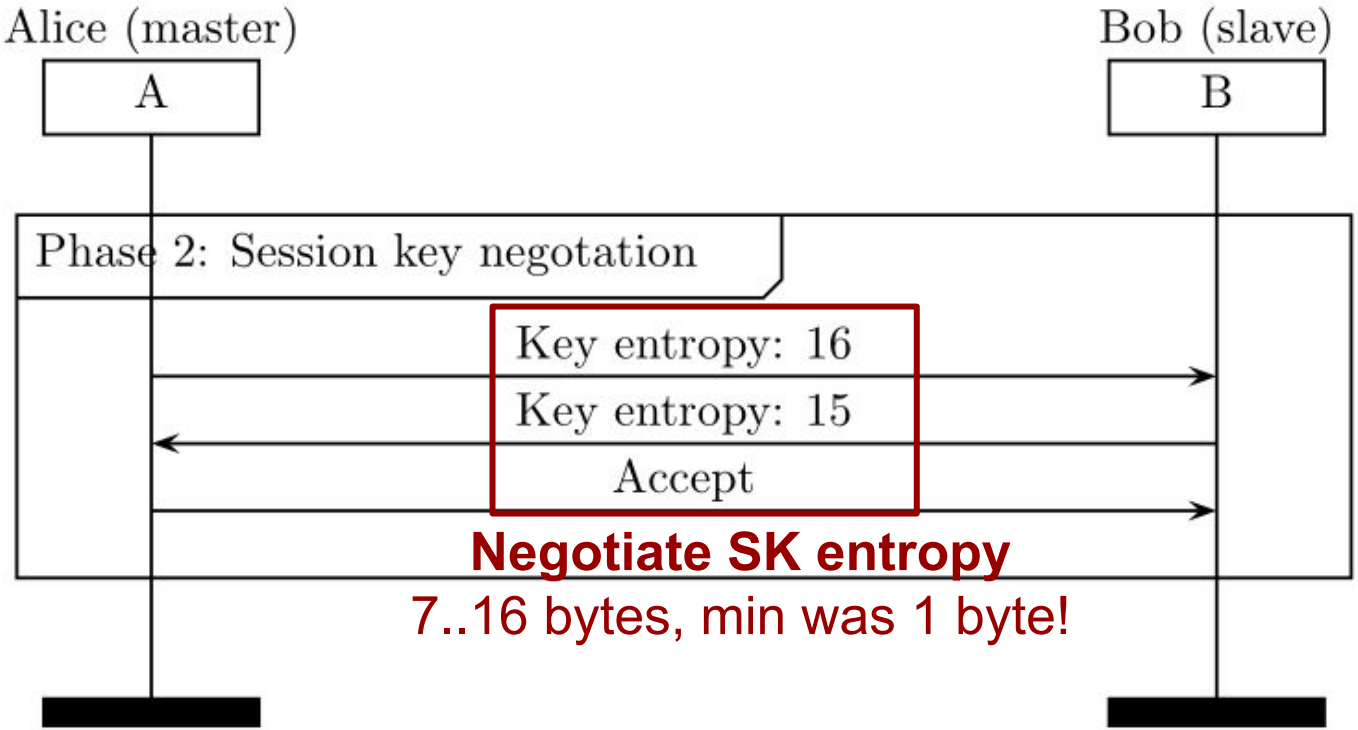
N = 7 0x123456789ABCDEF0123456789ABCDEF0
is **zeroed (weakened)** to
0x00000000000000003456789ABCDEF0

Key length still 16 B: no speedup, no energy saving! Entropy reduction is an attacker backdoor!

BC Session Phases



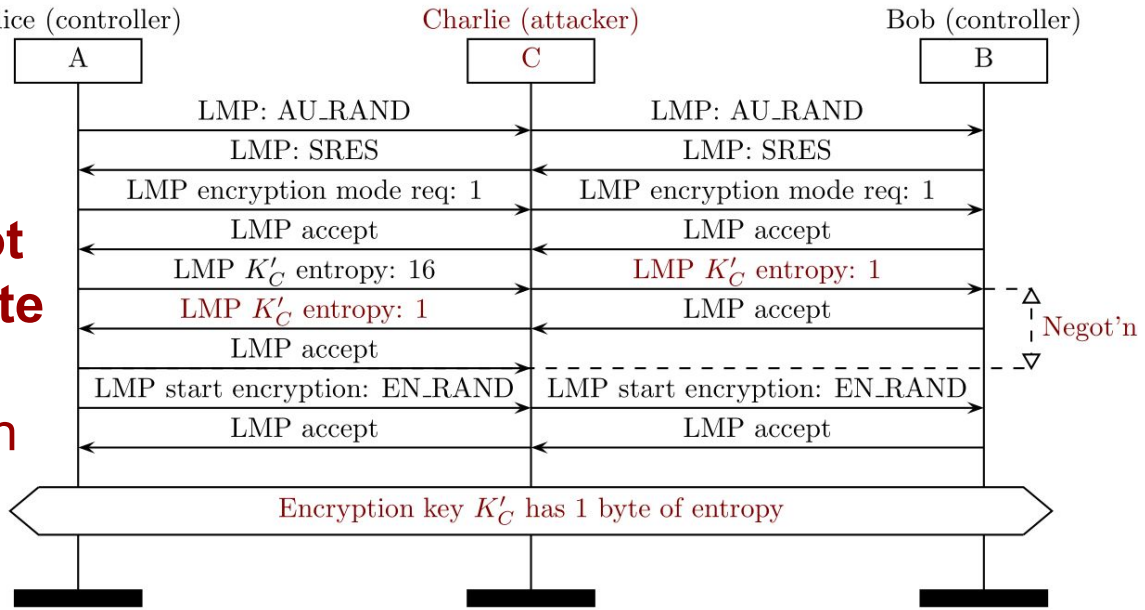
BC Session Key Negotiation



ZOBS$_2$: or Authenticate Entropy Negotiation

KNOB MitM on BC Session [SEC'19]

Alice and Bob **do not authenticate** entropy negotiation with PK



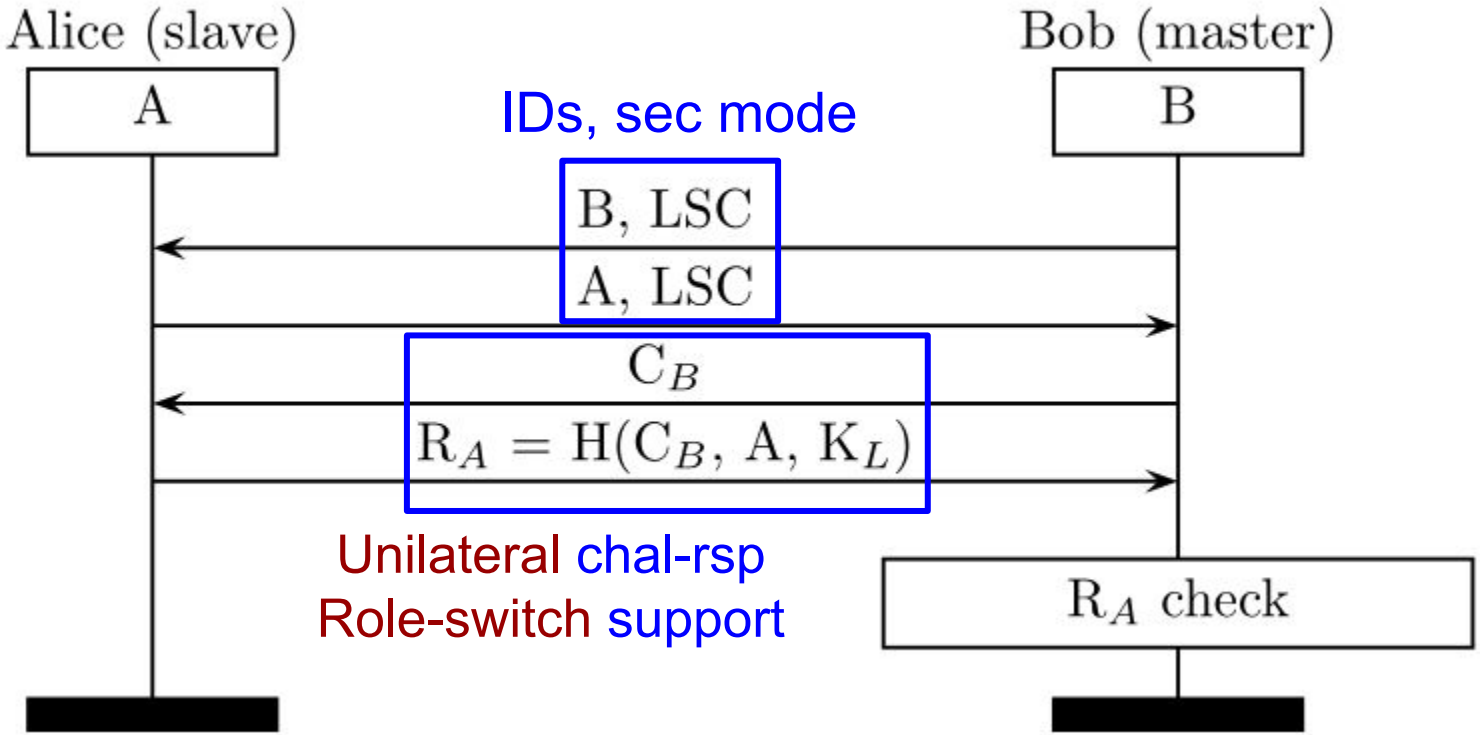
Key Entropy ZOBS

1 Remove Key Entropy Reduction

2 or Authenticate Entropy Values

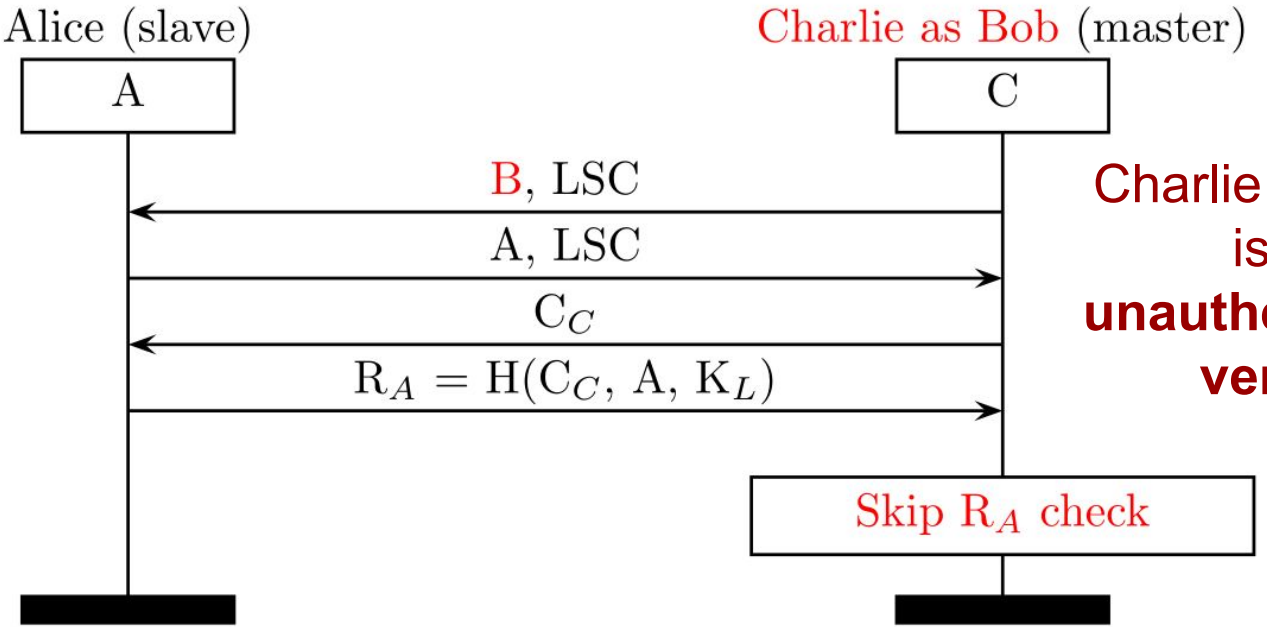
ZOBS: Device Authentication

BC Session LSC PK Authentication



$\mathbf{ZOBS_3}$: Mutually Authenticate Sessions

BIAS Central Imp. on LSC BC Session [SP'20]

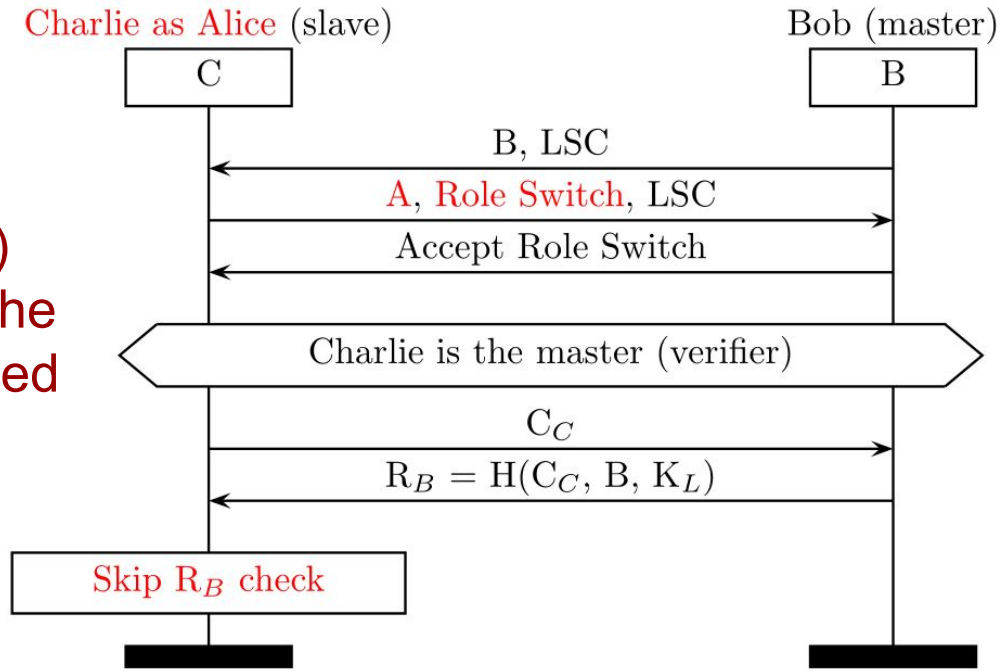


Charlie (Central) is an **unauthenticated verifier**

$\mathbf{ZOBS_4}$: No Authentication Role Switch

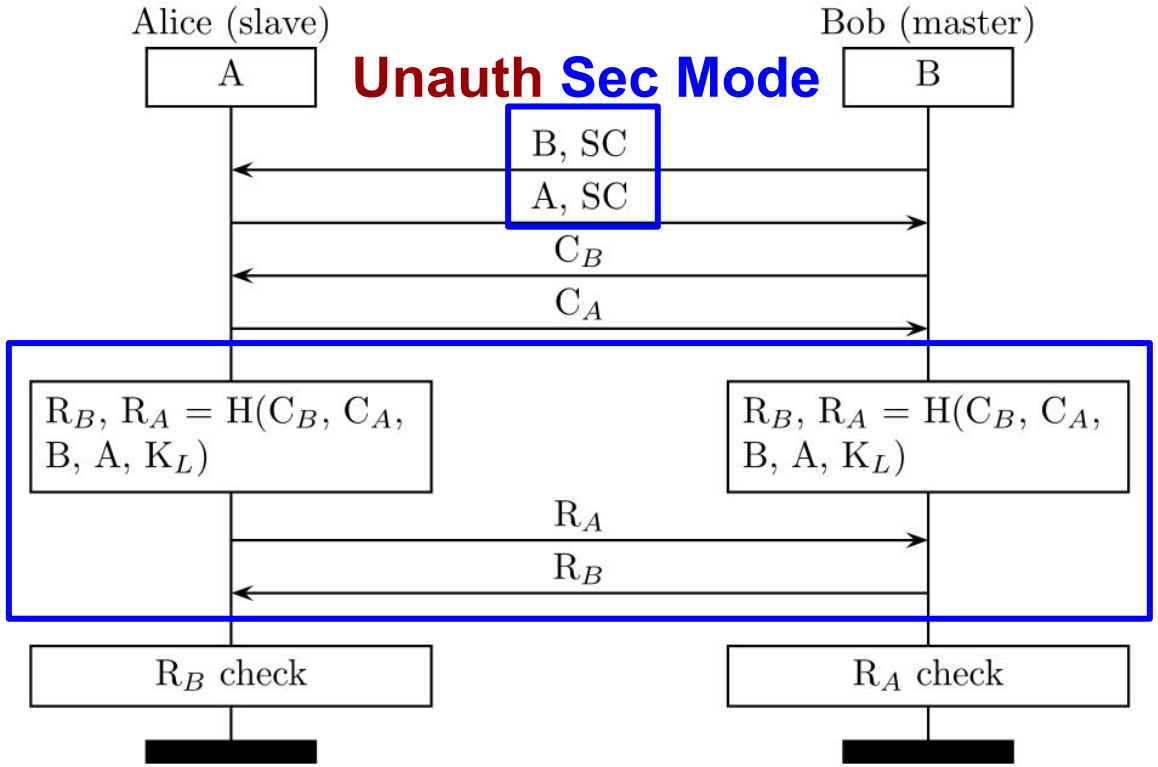
BIAS Peripheral Imp on LSC BC Session [SP'20]

Charlie (Peripheral) **switches** to the unauthenticated verifier role



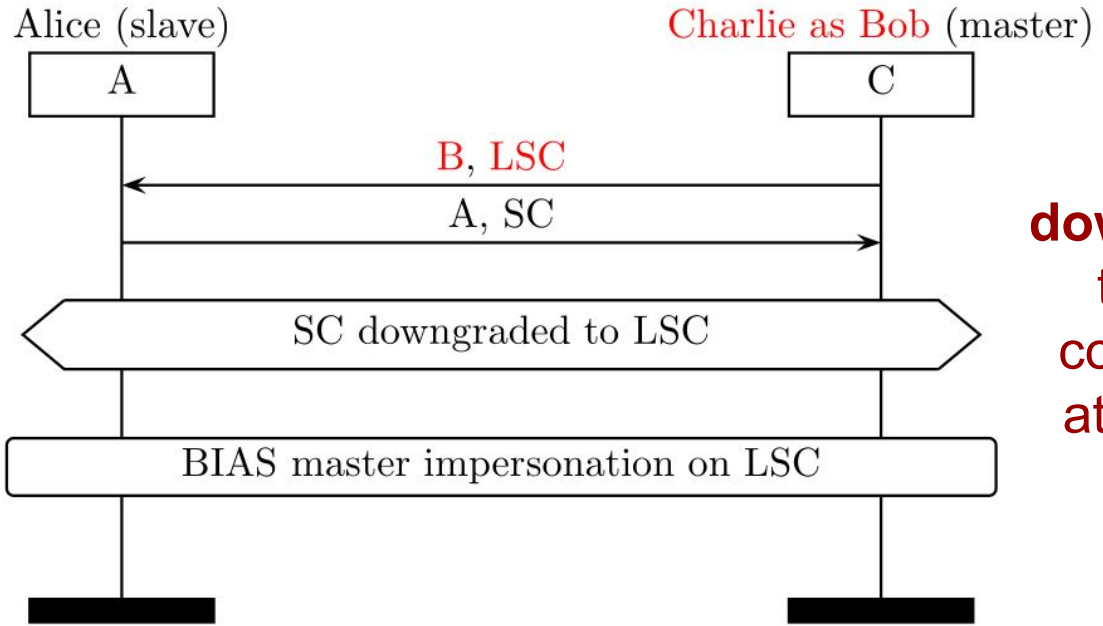
BC SC Session PK Authentication

Mutual
Chal-Rsp
Role-switch
support



ZOBS$_5$: No Session Security Mode Downgrade

BIAS Central Imp on SC BC Session [SP'20]

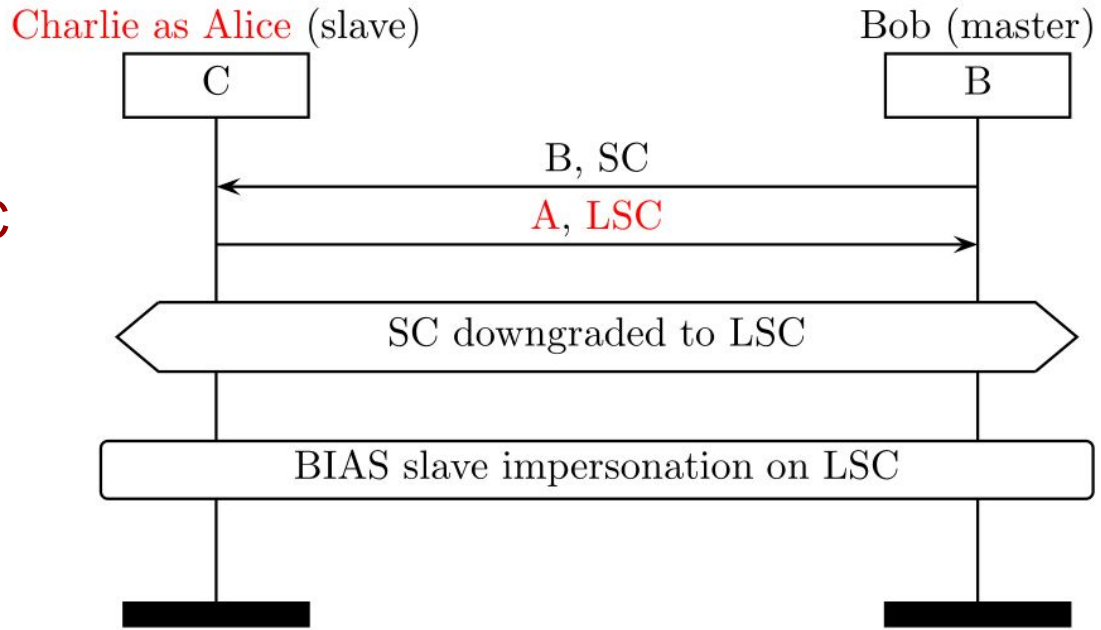


Charlie **downgrades** SC to LSC and conduct the CI attack on LSC

$\mathbf{ZOBS_5}$: No Session Security Mode Downgrade

BIAS Peripheral Imp. on SC BC Session [SP'20]

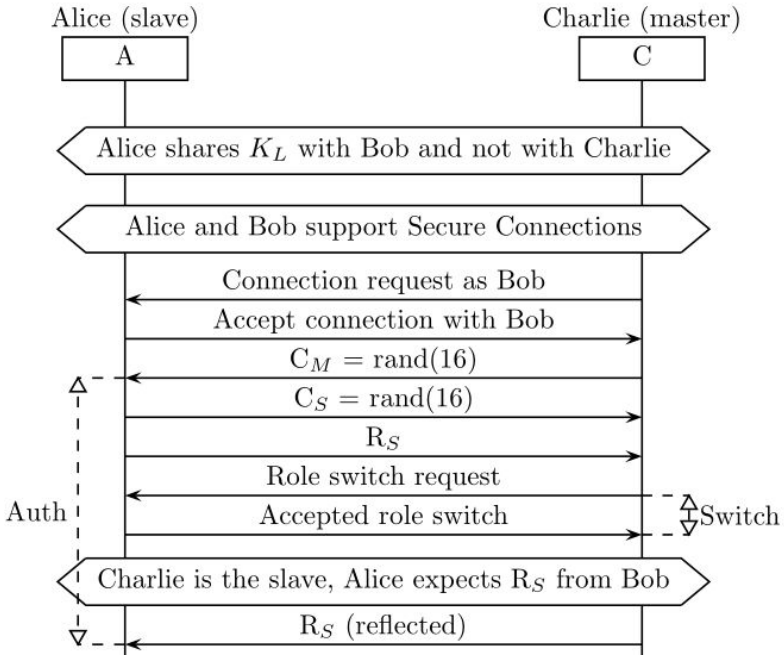
Charlie **downgrades** SC to LSC and conduct the PI attack on LSC



ZOBS$_6$: No Authentication Reflections

BIAS Central Imp. on SC BC Session [SP'20]

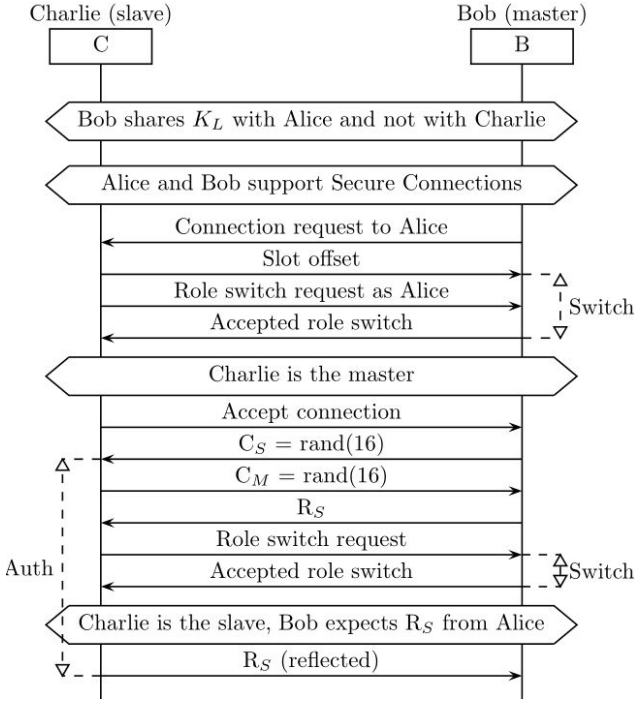
Charlie **reflects** **R_S** after a role switch



ZOBS$_6$: No Authentication Reflections

BIAS Peripheral Imp on SC BC Session [SP'20]

Charlie **reflects** **R$_s$** after two role switches



Device Authentication ZOBS

1 Remove Key Entropy Reduction

2 or Authenticate Entropy Values

3 Mutually Authenticate Sessions

4 No Authentication Role Switch

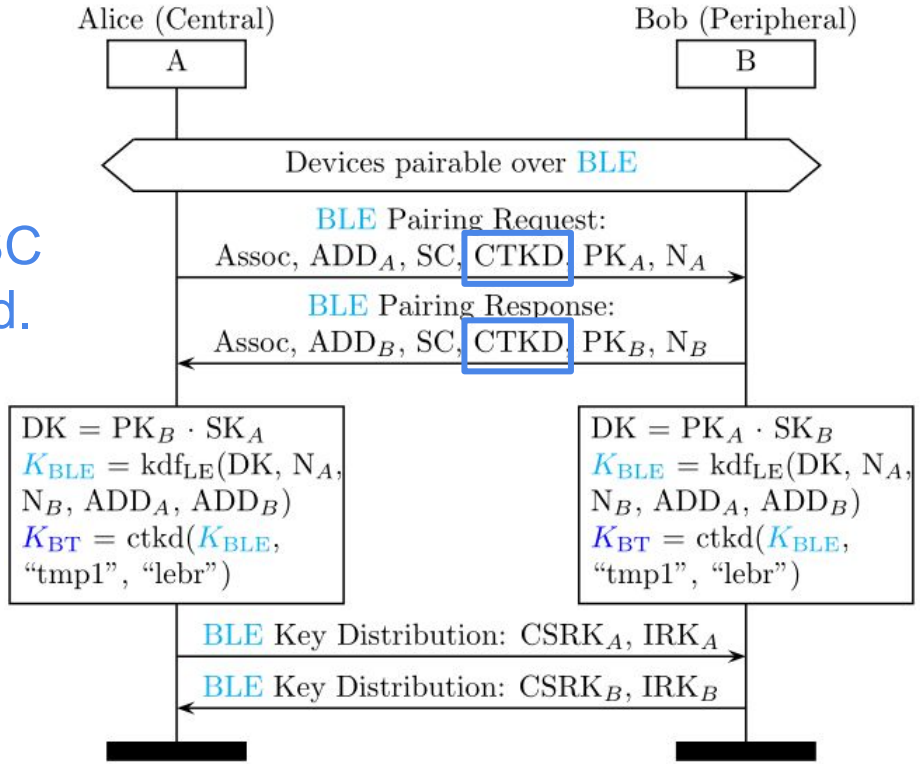
5 No Session Security Mode Downgrade

6 No Authentication Reflections

ZOBS: Cross-Transport

BLE Cross-Transport Key Derivation

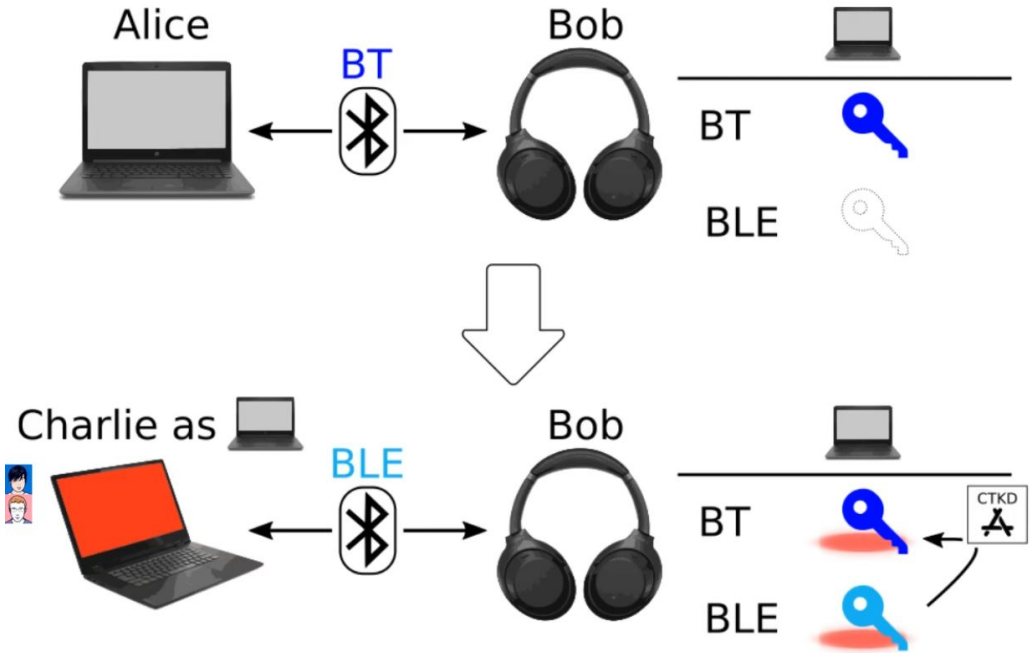
CTKD requires SC and is negotiated.



$\mathbf{ZOBS_7}$: No Cross-Transport Key Downgrade

BLUR Central Imp via BLE Pairing [AsiaCCS'22]

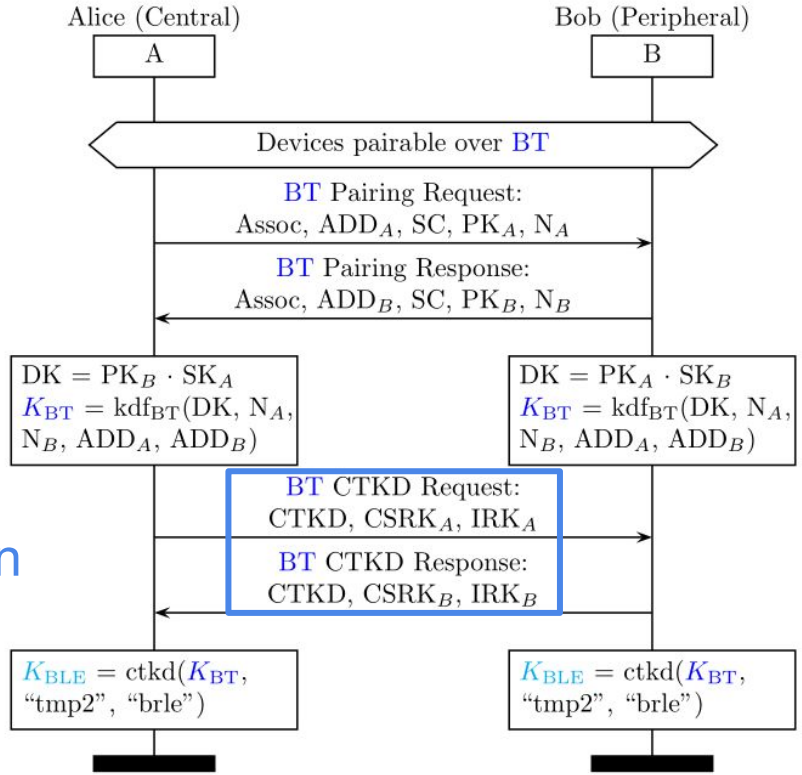
Charlie as Alice can **(over)write** and **downgrade** K_{BT}



BC Cross-Transport Key Derivation

CTKD is not negotiated!

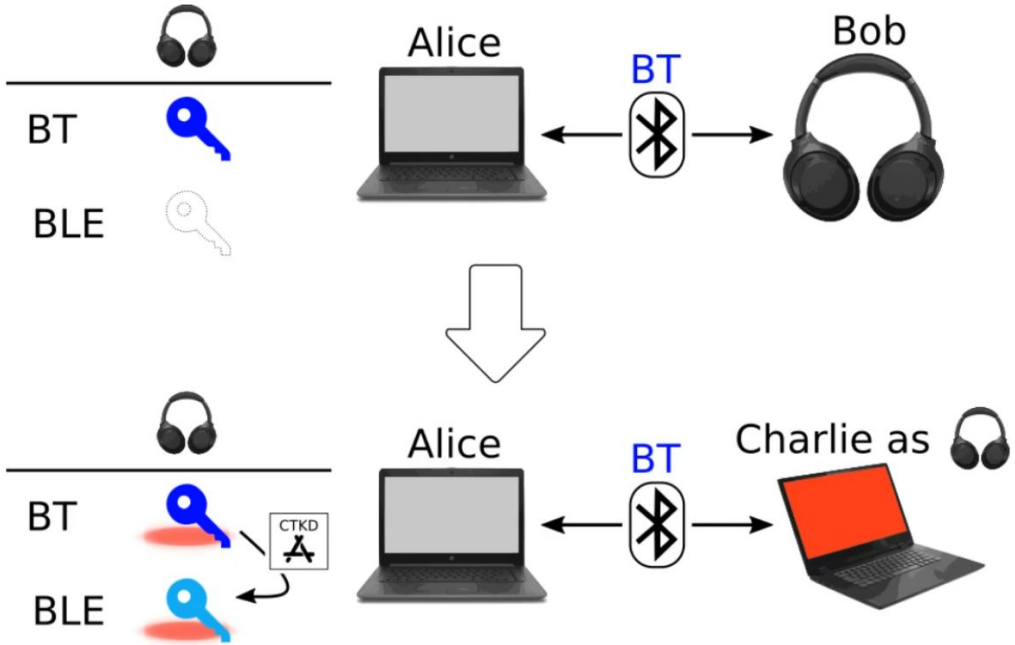
Optional BLE key distribution over BLE



ZOBS$_7$: No Cross-Transport Key Downgrade

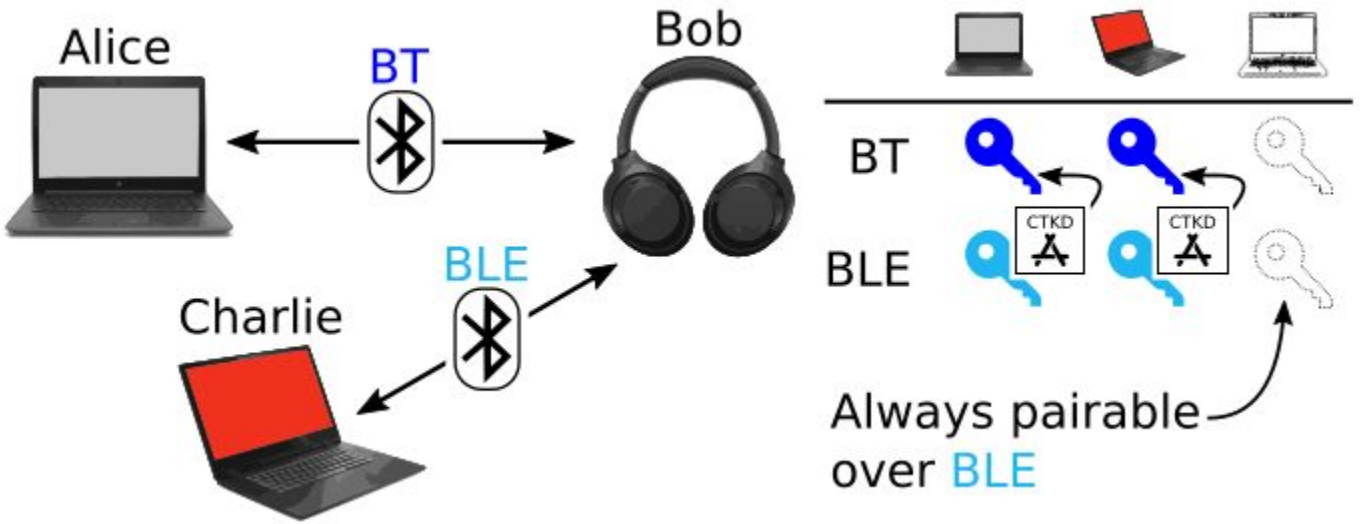
BLUR Peripheral Imp via BC Pairing [AsiaCCS'22]

Charlie as Bob can **(over)write** and **downgrade** K_{BLE}



$\mathbf{ZOBS_8}$: Non Discoverable can be Pairable

BLUR Unintended Sessions [AsiaCCS'22]



Cross-Transport ZOBS

1 Remove Key Entropy Reduction

2 or Authenticate Entropy Values

3 Mutually Authenticate Sessions

4 No Authentication Role Switch

5 No Session Security Mode Downgrade

6 No Authentication Reflections

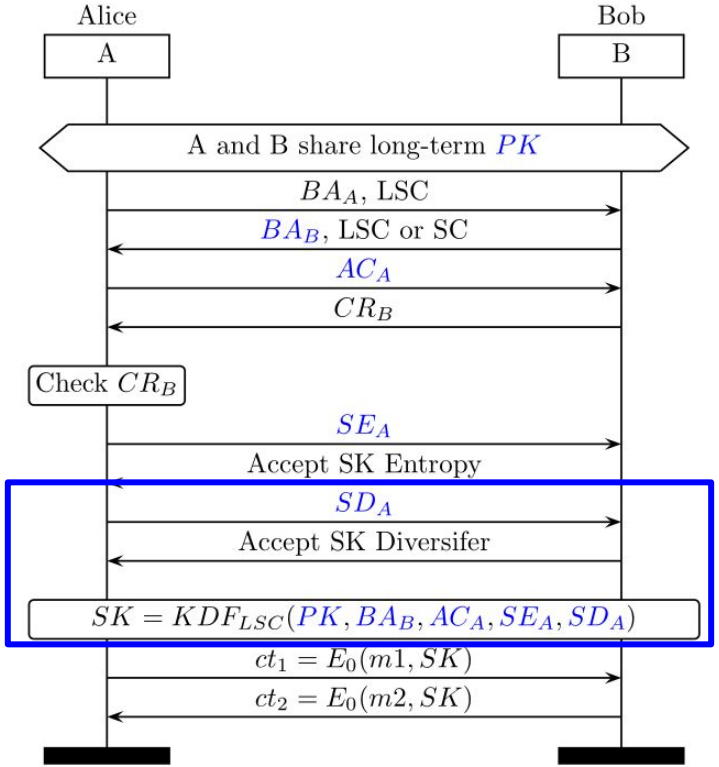
7 No Cross-Transport Key Downgrade

8 Non Discoverable can be Pairable

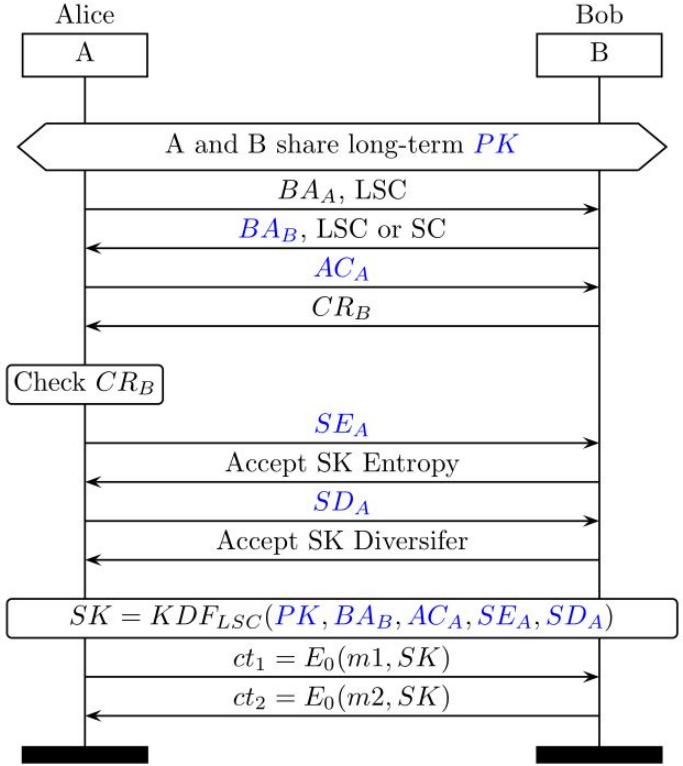
ZOBS: Forward and Future Secrecy

BC Session Key Derivation

Unilateral SK diversifier



BC Session Forward and Future Secrecy



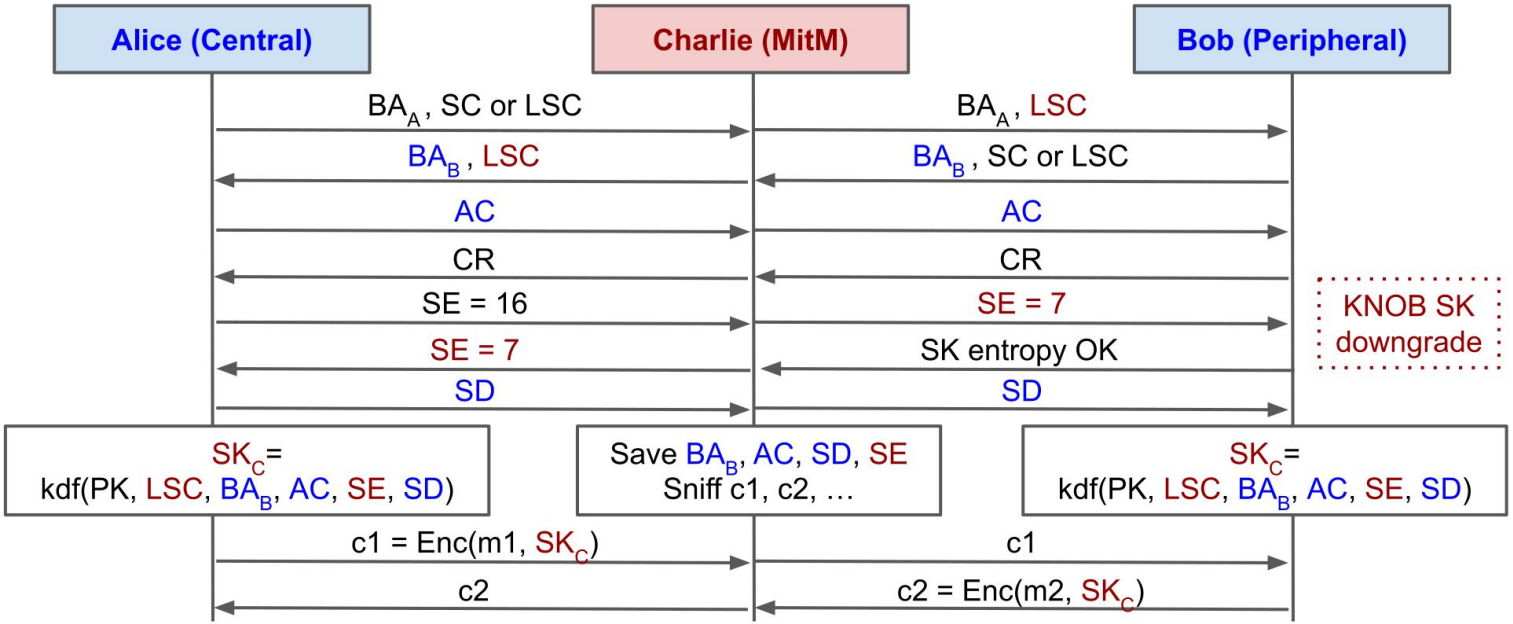
Assuming PK is not compromised.

Forward secrecy protects **past sessions** if current SK is compromised

Future secrecy protects **future sessions** if current SK is compromised

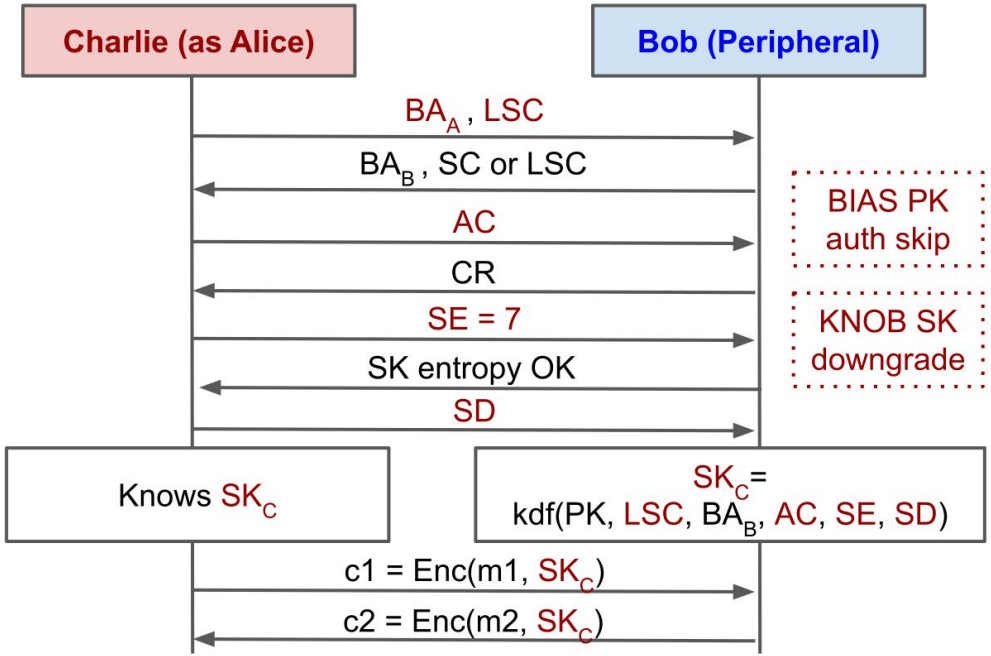
$\mathbf{ZOBS_9}$: Nonce as Session Key Diversifier

BLUFFS MitM breaking Forward Secrecy [CCS'23]



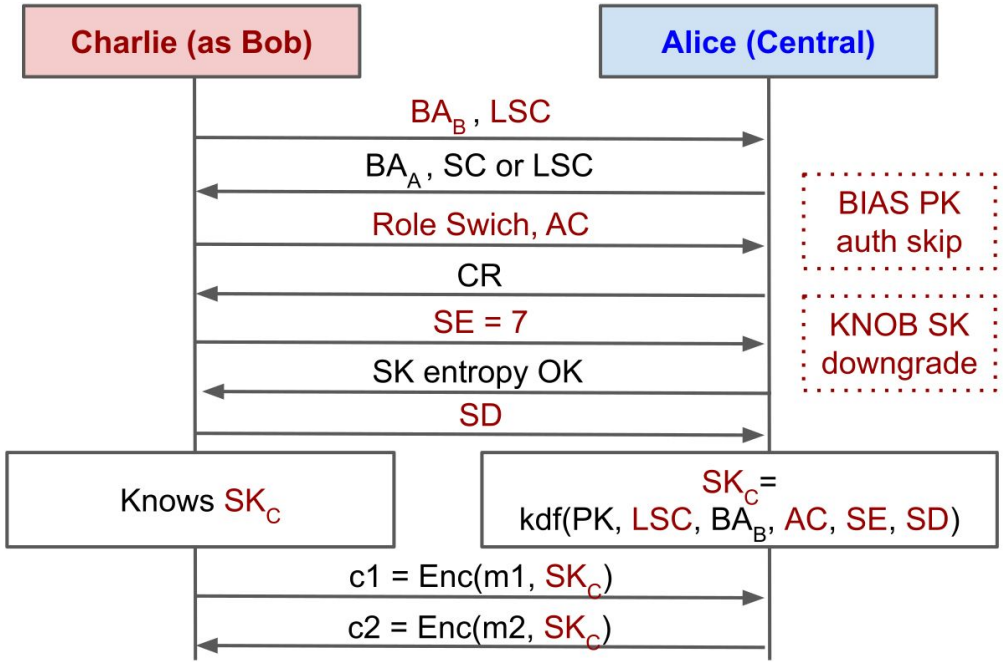
$\mathbf{ZOBS_{10}}$: Mutual Session Key Diversification

BLUFFS Central Imp breaking Future Secrecy [CCS'23]



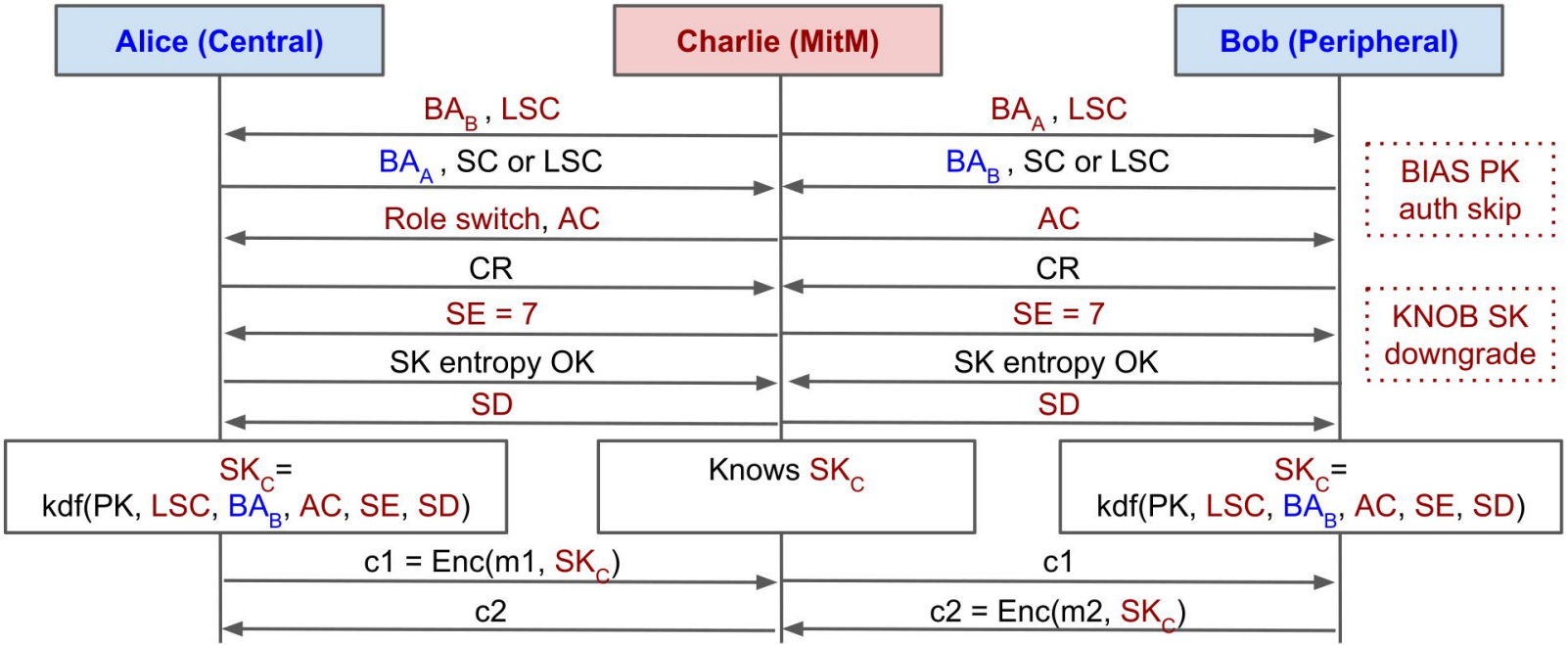
$\mathbf{ZOBS_{10}}$: Mutual Session Key Diversification

BLUFFS Periph Imp breaking Future Secrecy [CCS'23]



$ZOBS_{10}$: Mutual Session Key Diversification

BLUFFS MitM breaking Future Secrecy [CCS'23]



Forward and Future Secrecy ZOBS

1 Remove Key Entropy Reduction

2 or Authenticate Entropy Values

7 No Cross-Transport Key Downgrade

8 Non Discoverable can be Pairable

3 Mutually Authenticate Sessions

4 No Authentication Role Switch

5 No Session Security Mode Downgrade

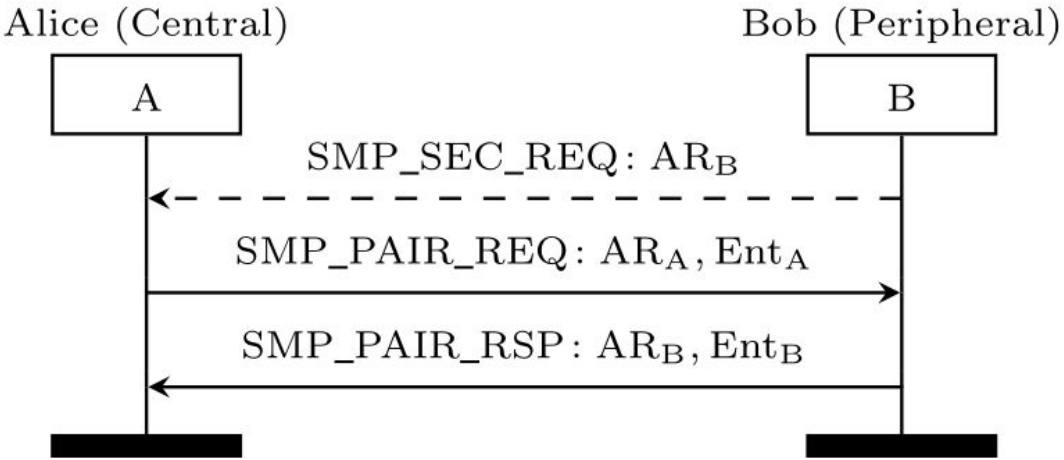
6 No Authentication Reflections

9 Nonce as Session Key Diversifier

10 Mutual Session Key Diversification

ZOBS: Re-pairing

BLE Re-Pairing

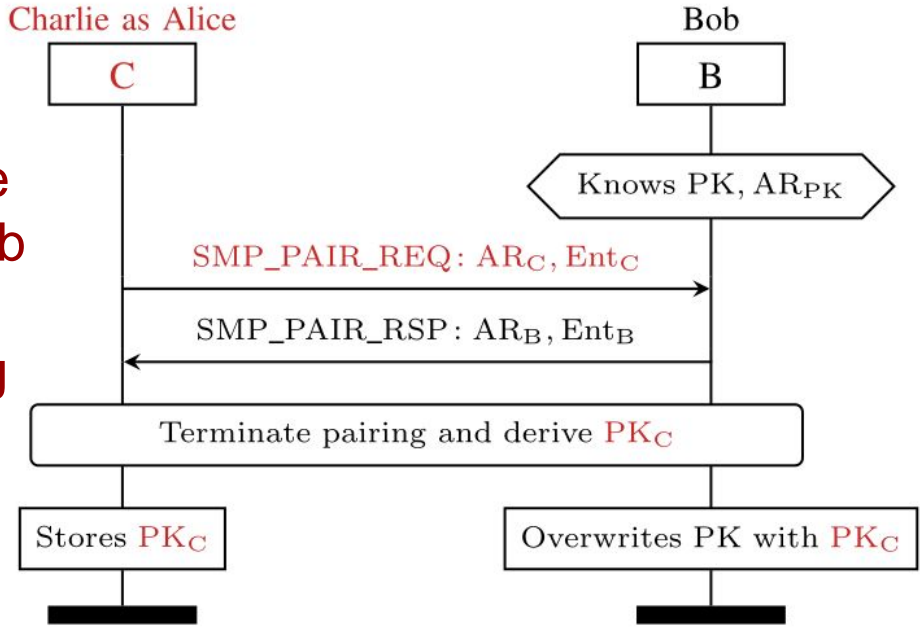


Devices share **PK** and $\mathbf{AR_{PK}}$
If SMP_SEC_REQ then $\mathbf{AR_B > AR_{PK}}$
Re-pair generates $\mathbf{PK_2}$ and $\mathbf{AR_{PK2}}$

$\mathbf{ZOBS}_{11}$: Authenticate Re-pairing

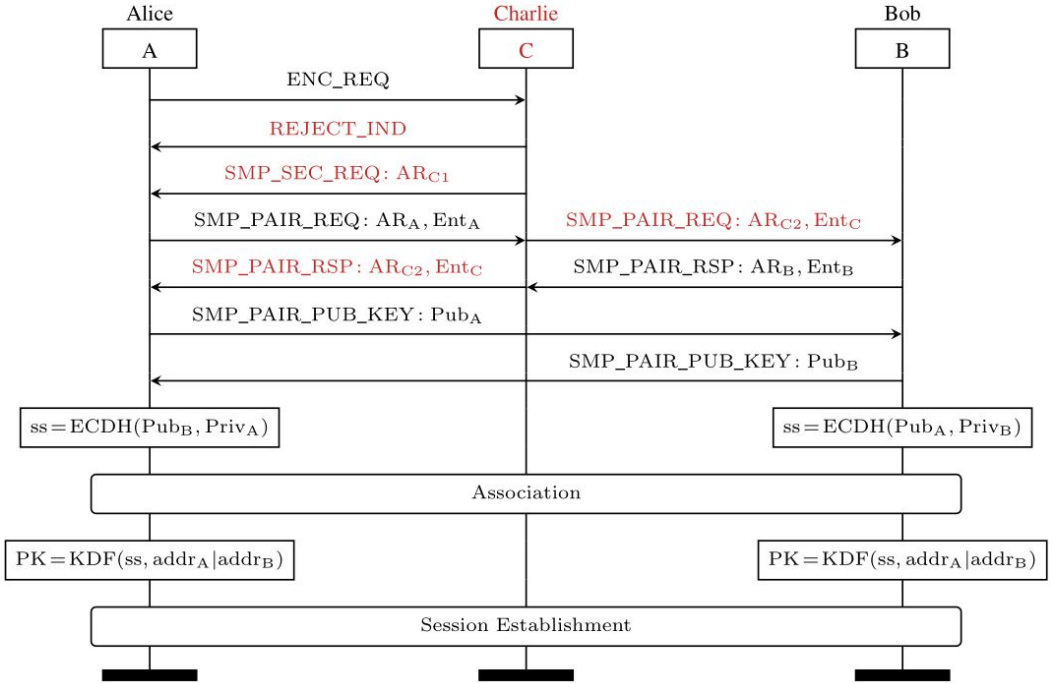
BLERP Central Imp via BLE re-pairing [NDSS'26]

Charlie as Alice re-pairs with Bob **without authenticating**



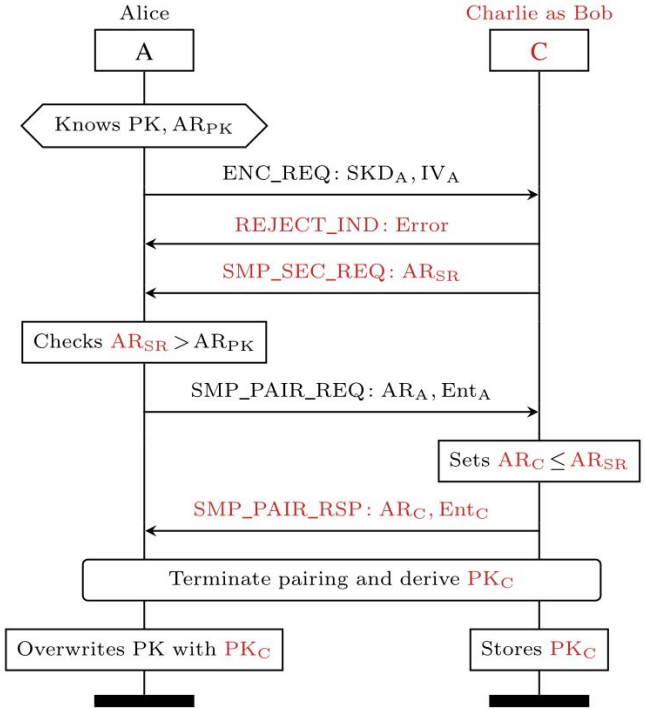
$\mathbf{ZOBS}_{11}$: Authenticate Re-pairing

BLERP SC MitM via BLE re-pairing [NDSS'26]



$\mathbf{ZOBS}_{12}$: No Re-pairing Key Downgrade

BLERP Peripheral Imp via BLE re-pairing [NDSS'26]



Charlie can trigger re-pairing and then **downgrade** PK via AR_C

Re-Pairing ZOBS

1 Remove Key Entropy Reduction

2 or Authenticate Entropy Values

3 Mutually Authenticate Sessions

4 No Authentication Role Switch

5 No Session Security Mode Downgrade

6 No Authentication Reflections

7 No Cross-Transport Key Downgrade

8 Non Discoverable can be Pairable

9 Nonce as Session Key Diversifier

10 Mutual Session Key Diversification

11 Authenticate Re-pairing

12 No Re-pairing Key Downgrade

Future Research Directions

Fix or **Replace** Bluetooth Security Protocols?!

Still pairing and session vulns!

Type	Attack	Year	Mode	Vulns.	Protocol
BLE	Pairing confusion [14]	2023	SC	C1	Pairing
BC	Pairing confusion [14]	2023	SC	C1	Pairing
BLE	Method confusion 2 [14]	2023	SC	C1	Pairing
BC	Method confusion 2 [14]	2023	SC	C1	Pairing
BC	BLUR [4]	2022	SC	C1, C2	Pairing
BLE	BLUR [4]	2022	SC	C1, C2	Pairing
BC	Method confusion [41]	2021	SC	C1	Pairing
BLE	Method confusion [41]	2021	SC	C1	Pairing
BC	BlueMirror A [15]	2021	LSC	C1, C2	Pairing
BLE	BlueMirror A [15]	2021	LSC	C1, C2	Pairing
BLE	BlueMirror PE-A1 [15]	2021	SC	C1, C2	Pairing
BLE	BlueMirror PE-A2 [15]	2021	SC	C1, C2	Pairing
BLE	KNOB [3]	2020	SC	C1	Pairing
BC	BLUFFS [1]	2023	SC	C3, C4	Session Est.
BC	BIAS [2]	2020	SC	C2, C4	Session Est.
BLE	BLESA [45]	2020	SC	C4	Session Est.
BC	KNOB [5]	2019	SC	C2, C4	Session Est.

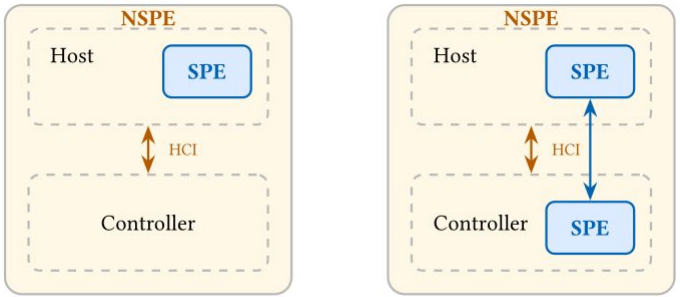
BlueBrothers, 3 new protocols to secure BC and BLE [WiSec'26]

Bluetooth Protocols vs **Software Compromise**

- Pairing and session **assume no SW compromise**
  - But Bluetooth stacks are riddled with SW vulnerabilities!
  - Enabling to leak PK, SK, …

Hardable: Secure Processing Env (SPE) for BLE [WiSec'26]

Protect Host and Controller even if the attacker compromises the NSPE



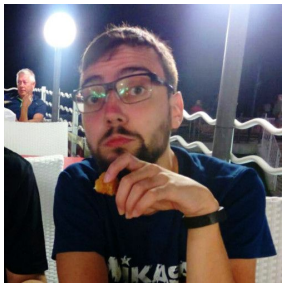
Single SPE Dual SPE

Conclusions

Ack to **co-authors**, **students**, and **funding**






ORSHIN

FRANCE 2030 PROGRAMME DE RECHERCHE RÉSEAUX DU FUTUR

European Commission Horizon Europe

anr agence nationale de la recherche

Shout-out: Zen of Research 2026 (ref)

The Zen of Research

Tips for New Researchers

Prof. Todd Austin
University of Michigan
austin@umich.edu

References (more)

- https://francozappa.github.io/publication/2019/knob/
- https://francozappa.github.io/publication/2020/knob-ble/
- https://francozappa.github.io/publication/2020/bias/
- https://francozappa.github.io/publication/2022/blur/
- https://francozappa.github.io/publication/2023/bluffs/
- https://francozappa.github.io/publication/2026/blerp/
- https://francozappa.github.io/publication/2026/bbro/
- https://francozappa.github.io/publication/2026/hardable/

Thanks! **12 ZOBS protosec**. AI. Q&A

1 Remove Key Entropy Reduction

2 or Authenticate Entropy Values

3 Mutually Authenticate Sessions

4 No Authentication Role Switch

5 No Session Security Mode Downgrade

6 No Authentication Reflections

7 No Cross-Transport Key Downgrade

8 Non Discoverable can be Pairable

9 Nonce as Session Key Diversifier

10 Mutual Session Key Diversification

11 Authenticate Re-pairing

12 No Re-pairing Key Downgrade