black hat ASIA 2026

Exploiting BLE Re-Pairing with the BLERP Attacks

Tommaso Sacchetti & Daniele Antonioli

Who Are We?

- Daniele Antonioli
  - Asst. Prof at EURECOM
  - System Security and Privacy
  - francozappa.github.io

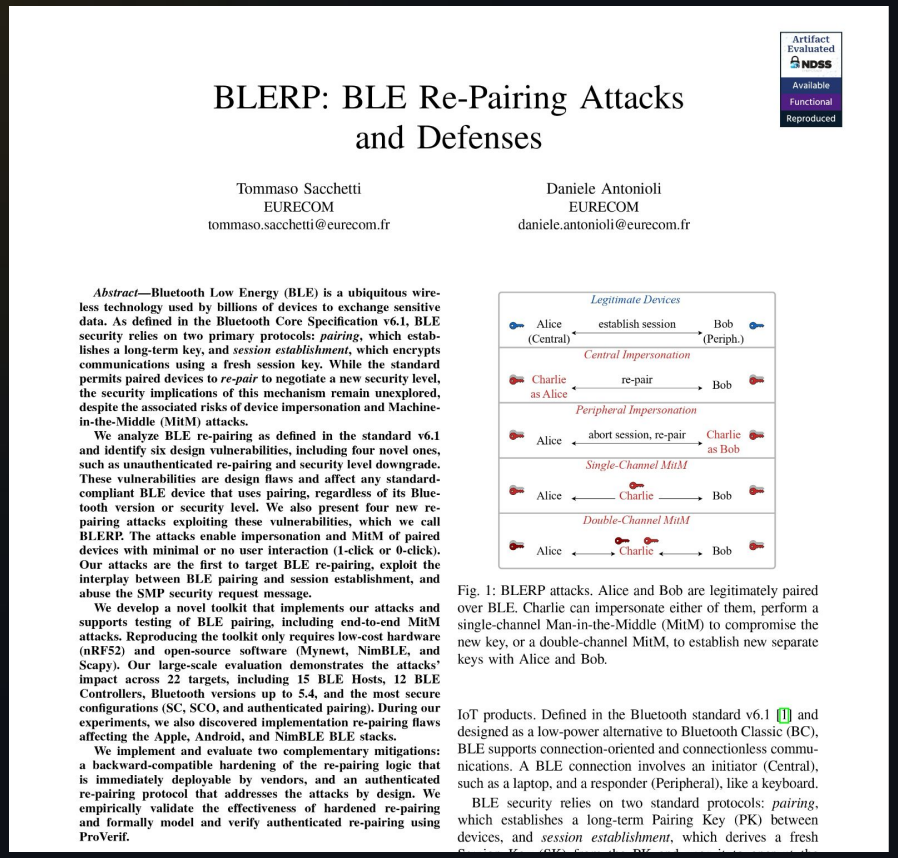
- Tommaso Sacchetti
  - PhD at EURECOM
  - Wireless and Protocol Security

EURECOM
Sophia Antipolis

Talk Outline

1. BLE Introduction
2. BLERP Vulnerabilities and Attacks
3. BLERP Attacks Demo
4. Toolkit and Evaluation
5. Mitigations and Fixes
6. Disclosure and Takeaways

BLERP NDSS'26 Paper and Artifact



Artifact Evaluated NDSS
Available
Functional
Reproduced

BLERP: BLE Re-Pairing Attacks and Defenses

Tommaso Sacchetti
EURECOM
tommaso.sacchetti@eurecom.fr

Daniele Antonioli
EURECOM
daniele.antonioli@eurecom.fr

***Abstract*—Bluetooth Low Energy (BLE) is a ubiquitous wireless technology used by billions of devices to exchange sensitive data. As defined in the Bluetooth Core Specification v6.1, BLE security relies on two primary protocols: *pairing*, which establishes a long-term key, and *session establishment*, which encrypts communications using a fresh session key. While the standard permits paired devices to *re-pair* to negotiate a new security level, the security implications of this mechanism remain unexplored, despite the associated risks of device impersonation and Machine-in-the-Middle (MitM) attacks.**

We analyze BLE re-pairing as defined in the standard v6.1 and identify six design vulnerabilities, including four novel ones, such as unauthenticated re-pairing and security level downgrade. These vulnerabilities are design flaws and affect any standard-compliant BLE device that uses pairing, regardless of its Bluetooth version or security level. We also present four new re-pairing attacks exploiting these vulnerabilities, which we call BLERP. The attacks enable impersonation and MitM of paired devices with minimal or no user interaction (1-click or 0-click). Our attacks are the first to target BLE re-pairing, exploit the interplay between BLE pairing and session establishment, and abuse the SMP security request message.

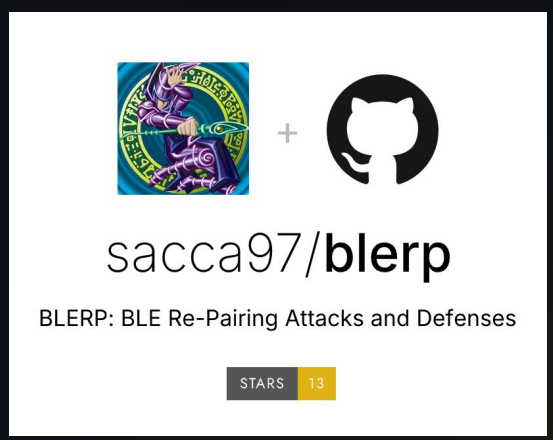
We develop a novel toolkit that implements our attacks and supports testing of BLE pairing, including end-to-end MitM attacks. Reproducing the toolkit only requires low-cost hardware (nRF52) and open-source software (Mynewt, NimBLE, and Scapy). Our large-scale evaluation demonstrates the attacks' impact across 22 targets, including 15 BLE Hosts, 12 BLE Controllers, Bluetooth versions up to 5.4, and the most secure configurations (SC, SCO, and authenticated pairing). During our experiments, we also discovered implementation re-pairing flaws affecting the Apple, Android, and NimBLE BLE stacks.

We implement and evaluate two complementary mitigations: a backward-compatible hardening of the re-pairing logic that is immediately deployable by vendors, and an authenticated re-pairing protocol that addresses the attacks by design. We empirically validate the effectiveness of hardened re-pairing and formally model and verify authenticated re-pairing using ProVerif.

Fig. 1: BLERP attacks. Alice and Bob are legitimately paired over BLE. Charlie can impersonate either of them, perform a single-channel Man-in-the-Middle (MitM) to compromise the new key, or a double-channel MitM, to establish new separate keys with Alice and Bob.

IoT products. Defined in the Bluetooth standard v6.1 [1] and designed as a low-power alternative to Bluetooth Classic (BC), BLE supports connection-oriented and connectionless communications. A BLE connection involves an initiator (Central), such as a laptop, and a responder (Peripheral), like a keyboard.

BLE security relies on two standard protocols: *pairing*, which establishes a long-term Pairing Key (PK) between devices, and *session establishment*, which derives a fresh



francozappa.github.io/publication/2026/blerp/

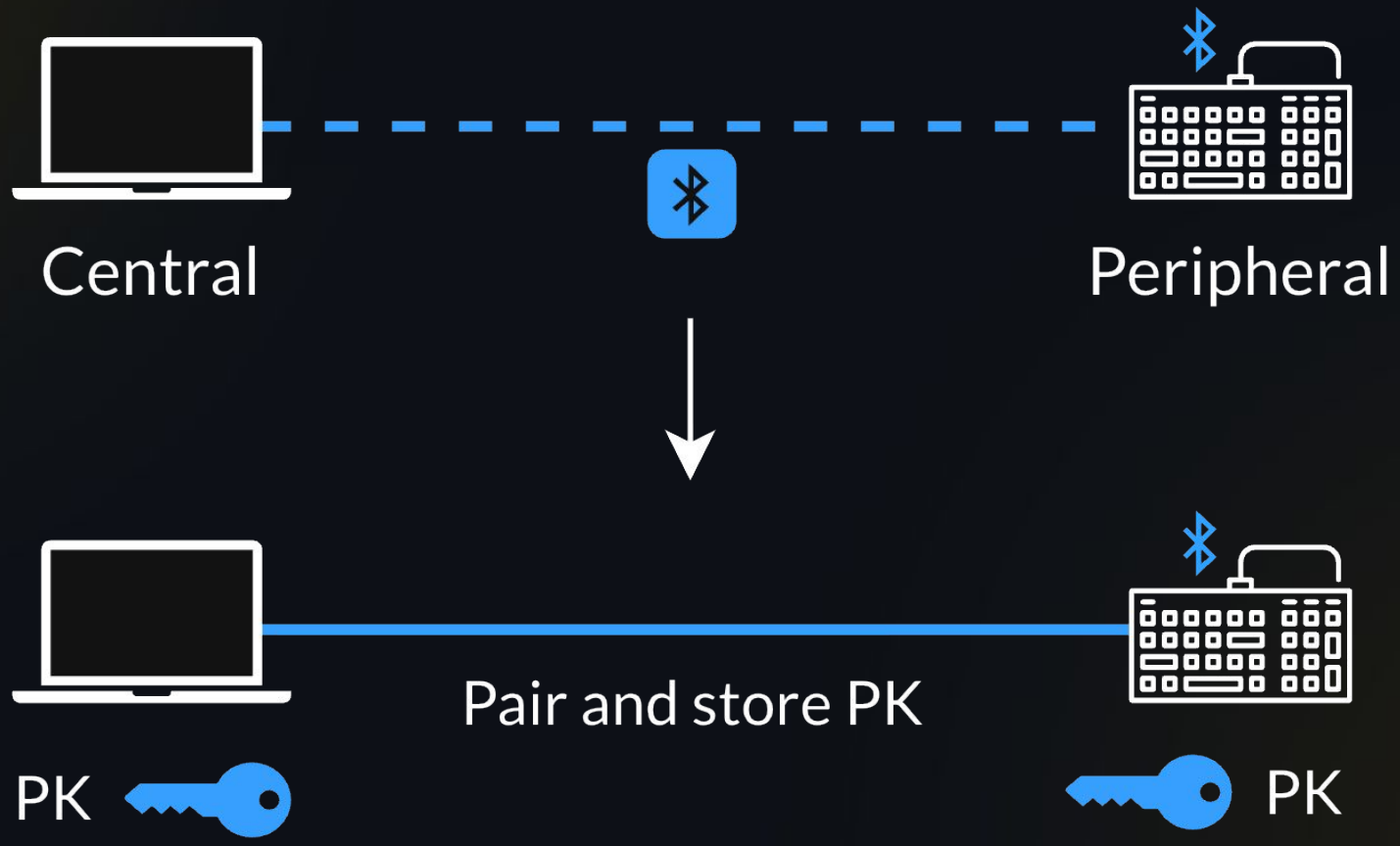
BLE Introduction

Bluetooth Low Energy (BLE)

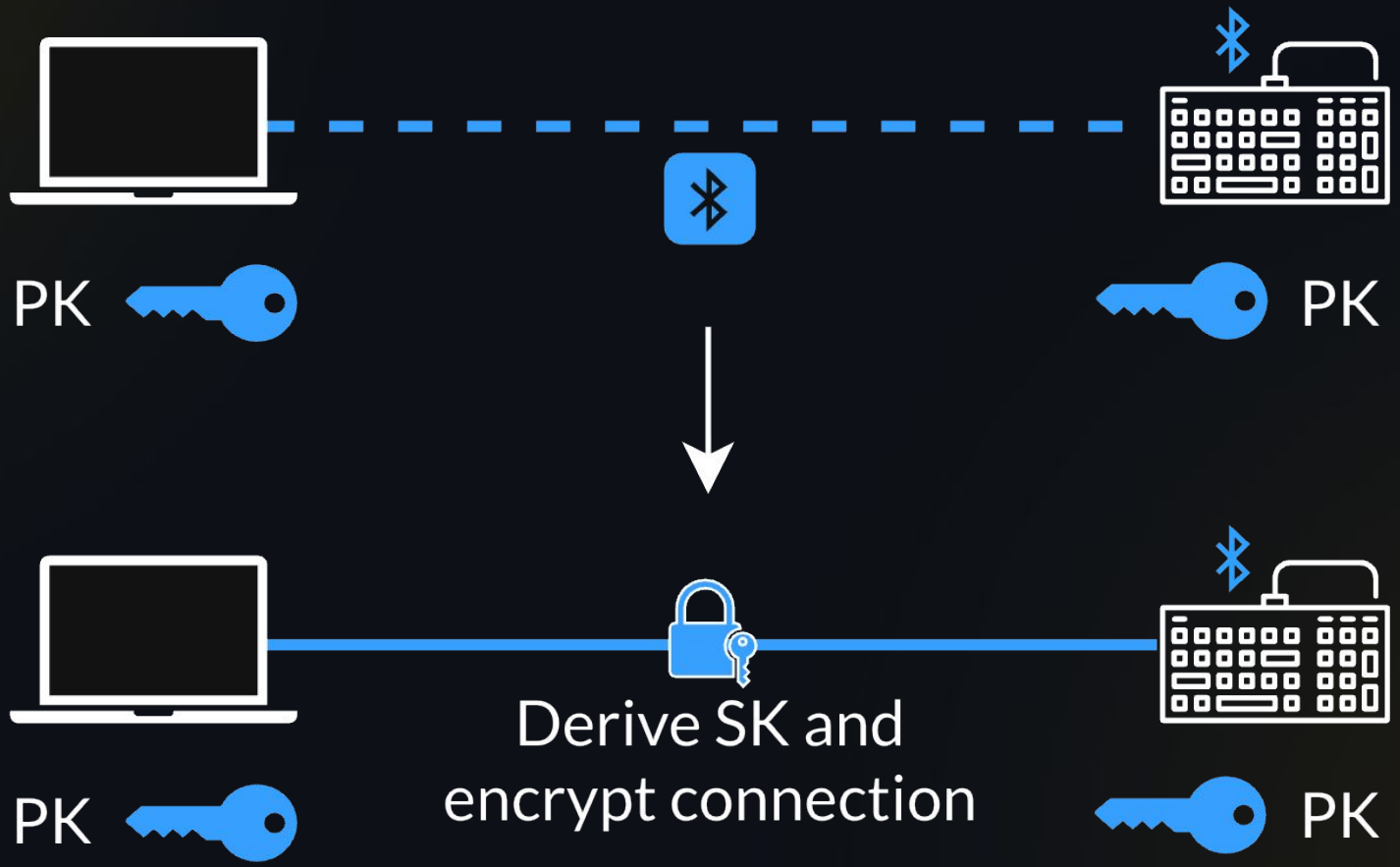
Low-power wireless technology

- 2.4 GHz (ISM), alternative to Bluetooth Classic (BC)
- Short and long range (30m to 1km LoS)
- Connection-oriented and connectionless
- Billions of devices (mobile, IoT, cars ...)
- Core specification v6.2

BLE Discovery and Pairing



BLE Session Establishment



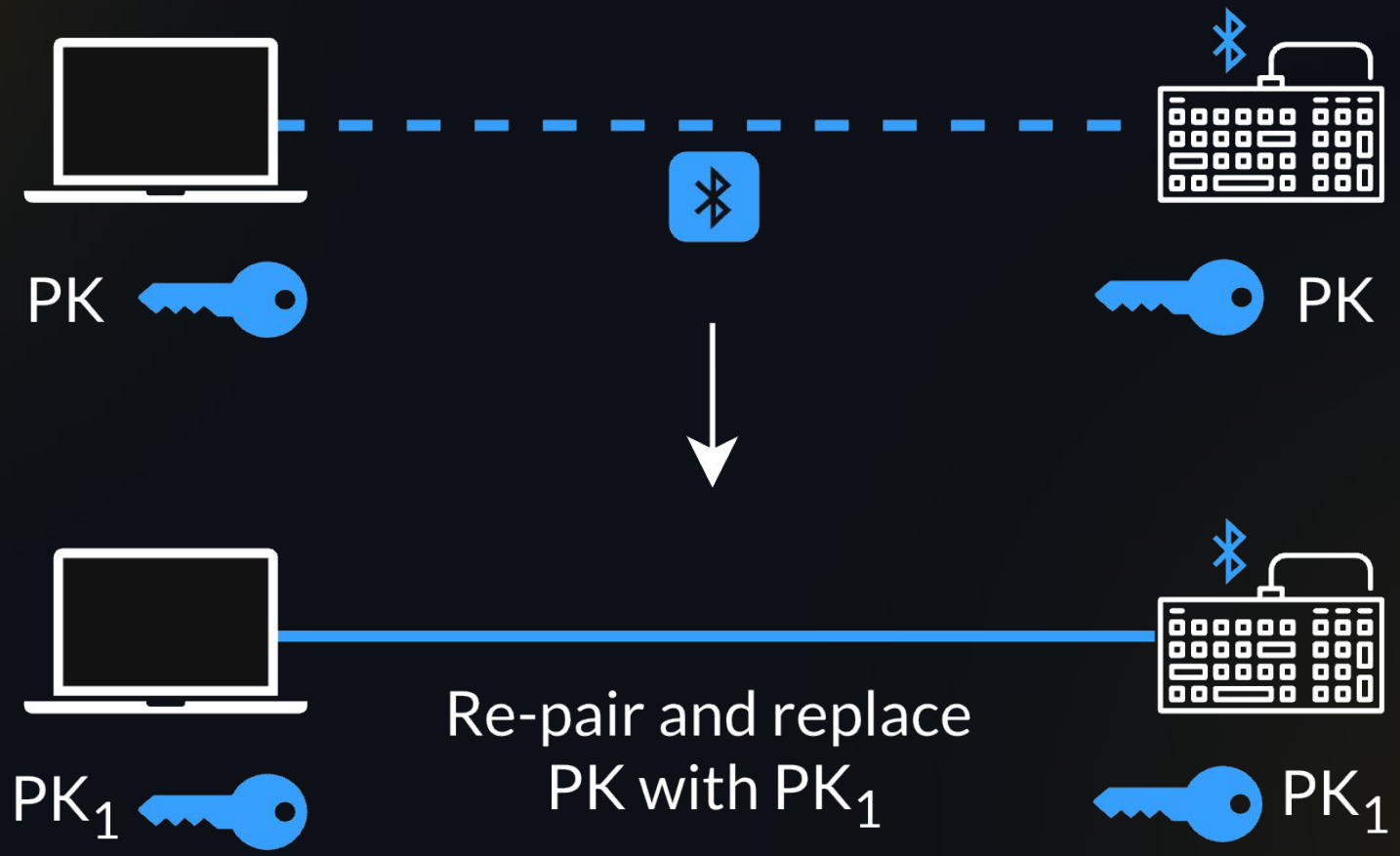
BLE Protocol Security

One **protocol vulnerability** on pairing or session establishment → Billions of **exploitable** BLE devices!

E.g., BLE Pairing:

- Crackle on legacy pairing
- KNOB PK entropy downgrade
- Invalid ECDH coordinate attack

Focus on BLE RE-Pairing



Focus on BLE RE-Pairing (2)

Specification (Core v6.2)

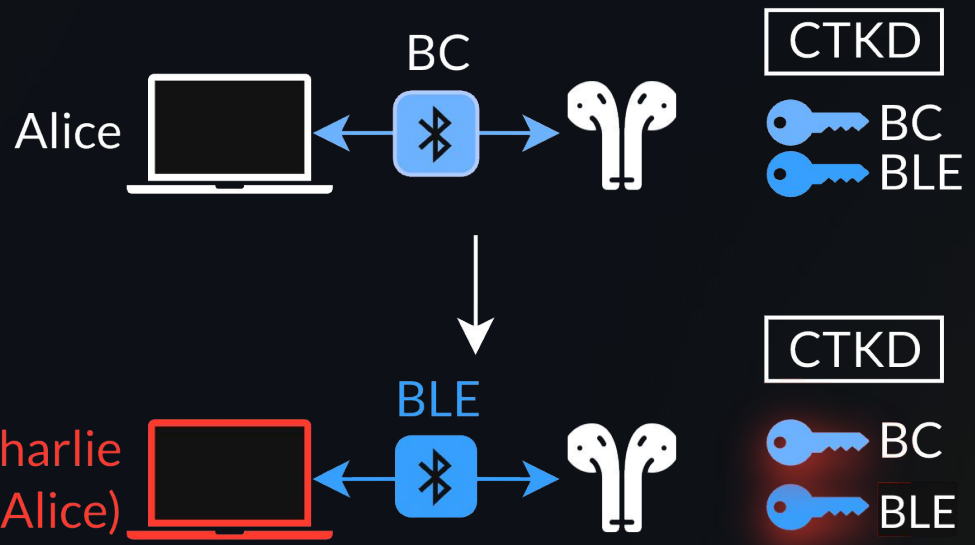
- Short and scattered
- *grep* “re-pair” yields 4 results in 3800 pages
- Not in BLE threat model

Critical attack surface

- Victim pairs with attacker instead of re-pairing
- Chain with pairing attacks (Crackle, KNOB, …)

BLE Re-Pairing Research?!

- BLURtooth attacks on BC-BLE re-pairing (CTKD)



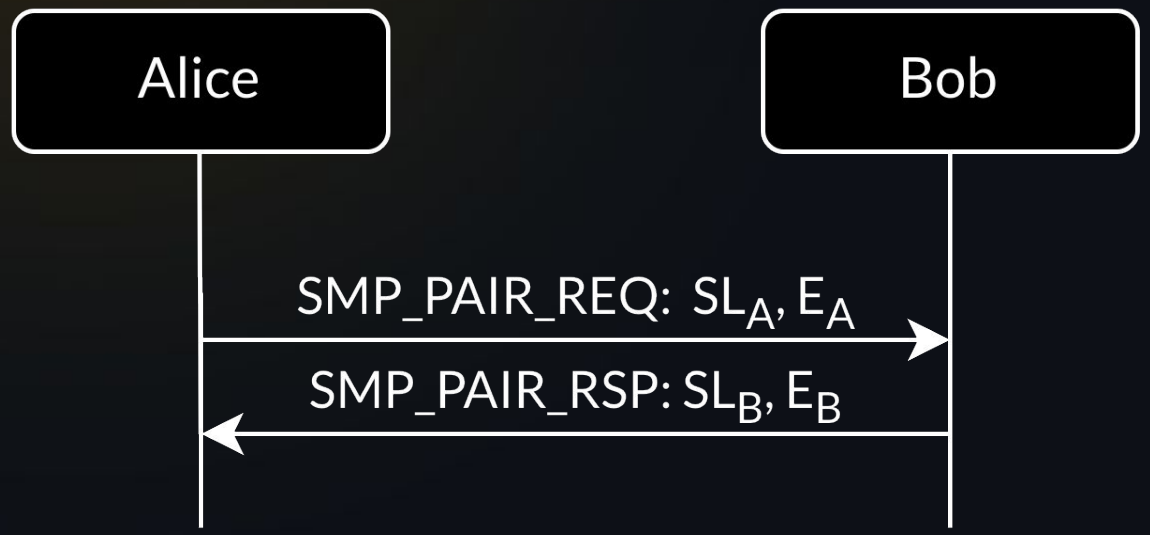
No security research on **BLE re-pairing**!

BLERP Contributions

- First evaluation of **BLE re-pairing**
- Six vulnerabilities (downgrade, no auth, ...)
- Four attacks (impersonation, MitM, 0-click, ...)
- Open-source BLERP toolkit and demos
- Large-scale evaluation on **22 devices** (SC, MitM, ...)
- Build mitigations and fixes
- Responsible disclosure (CVE-2025-62235)

BLERP Vulnerabilities and Attacks

BLE Pairing (Central)



SL =

Bonding Flag (2 bits)	MitM (1 bit)	SC (1 bit)	Keypress (1 bit)	CT2 (1 bit)	RFU (2 bits)

$E = 7 < n < 16$

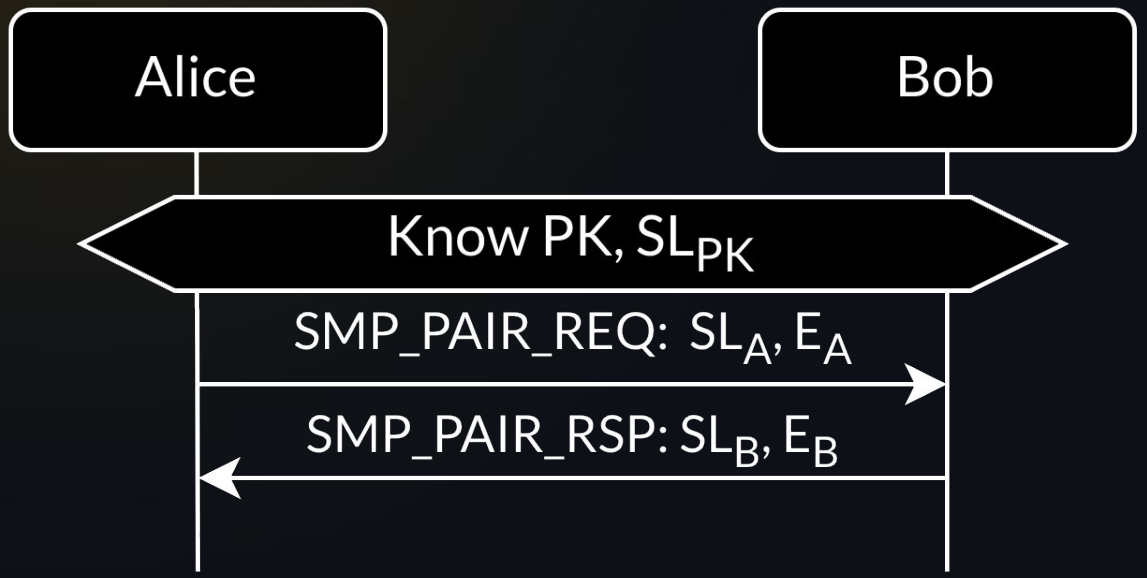
BLE Pairing (Peripheral)

SL =

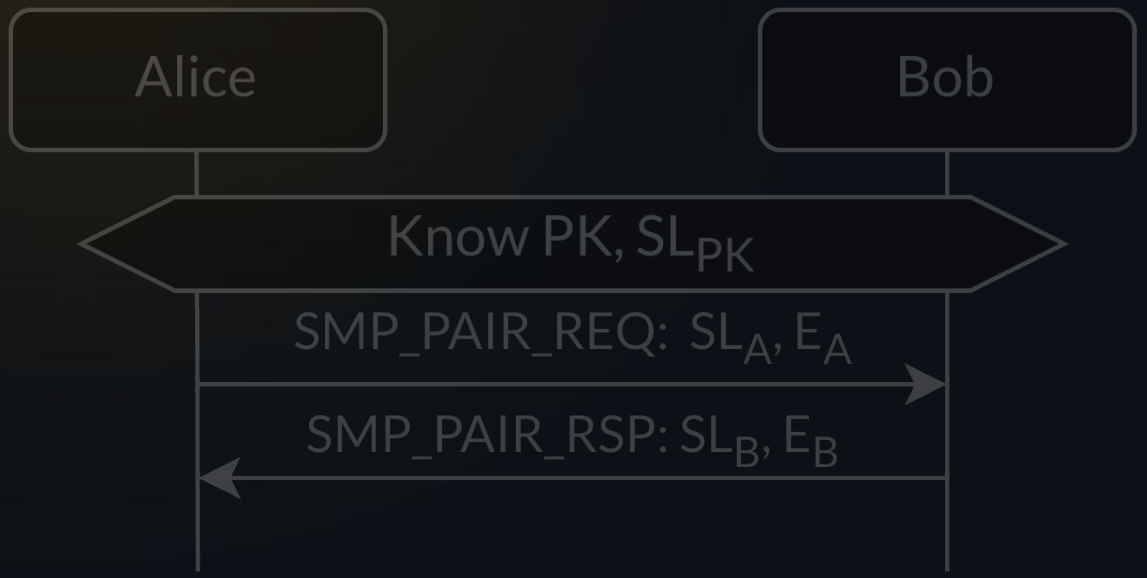
Bonding Flag (2 bits)	MitM (1 bit)	SC (1 bit)	Keypress (1 bit)	CT2 (1 bit)	RFU (2 bits)

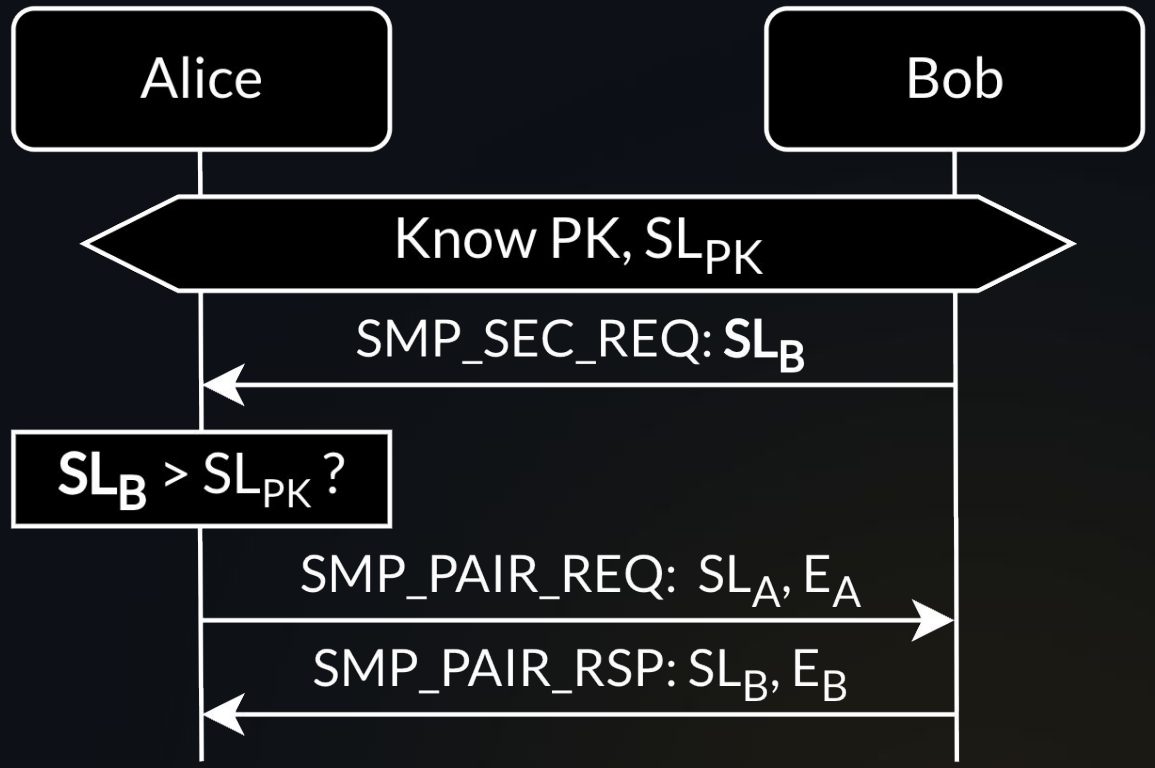
$E = 7 < n < 16$

BLE Re-Pairing (Central)

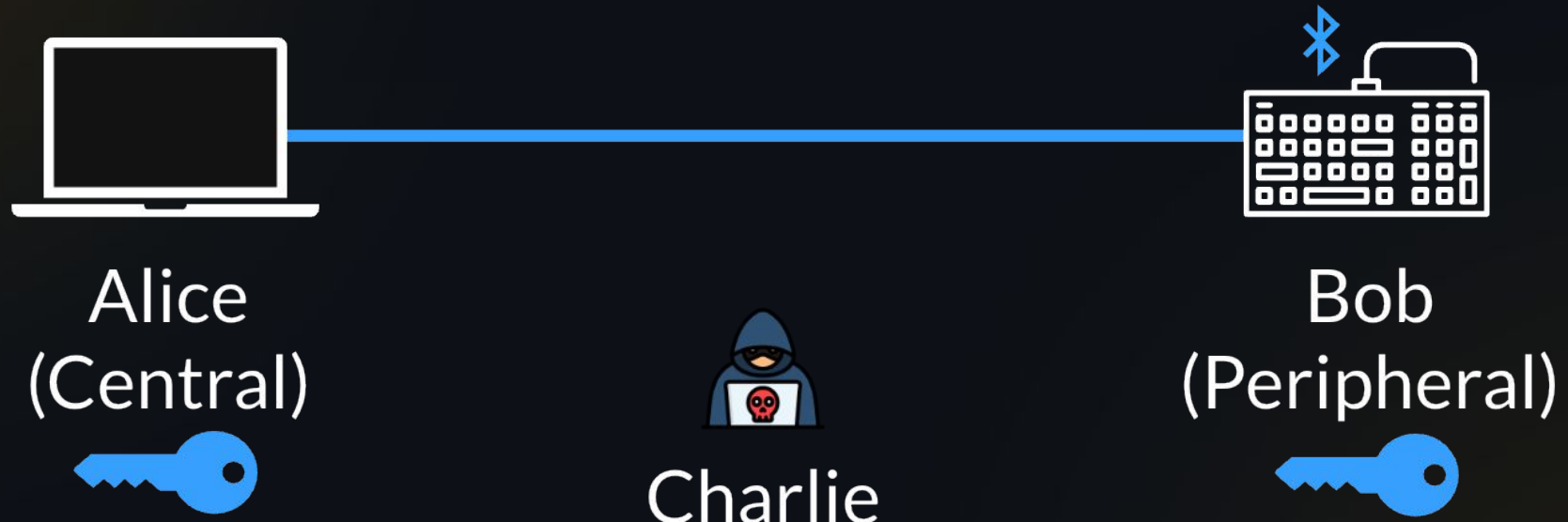


BLE Re-Pairing (Peripheral)



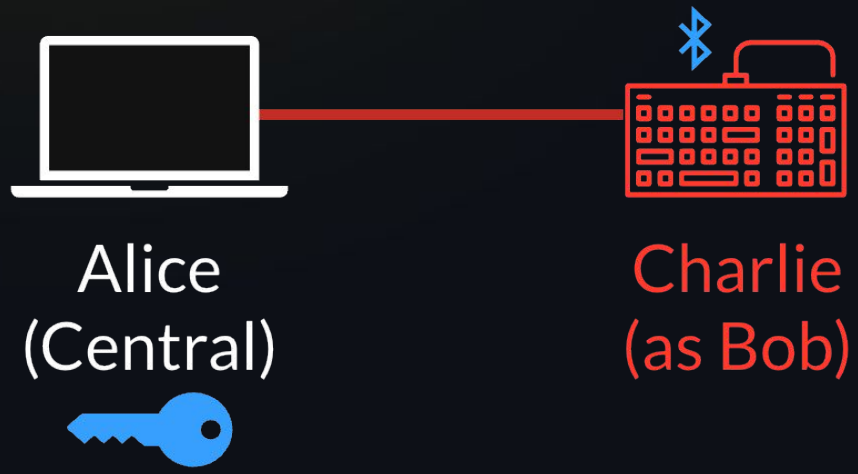


Re-Pairing Threat Model



Re-Pairing Threat Model (2)

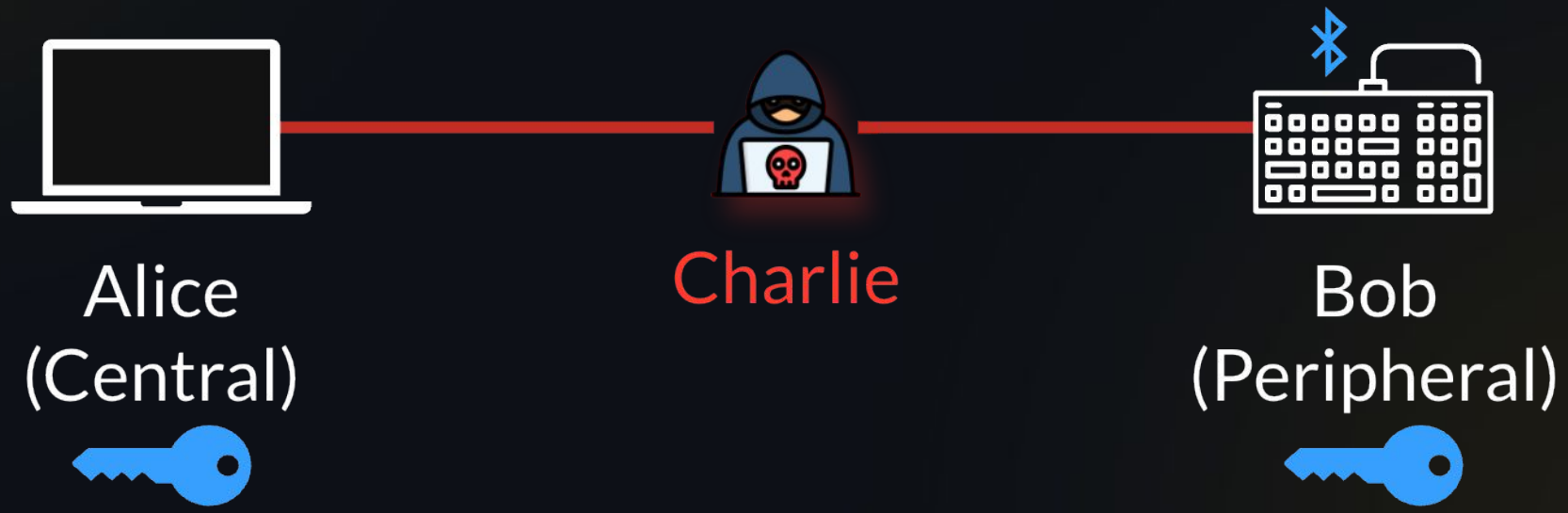
Impersonate Peripheral



Impersonate Central

Re-Pairing Threat Model (3)

MitM Central and Peripheral



Four BLERP Attacks

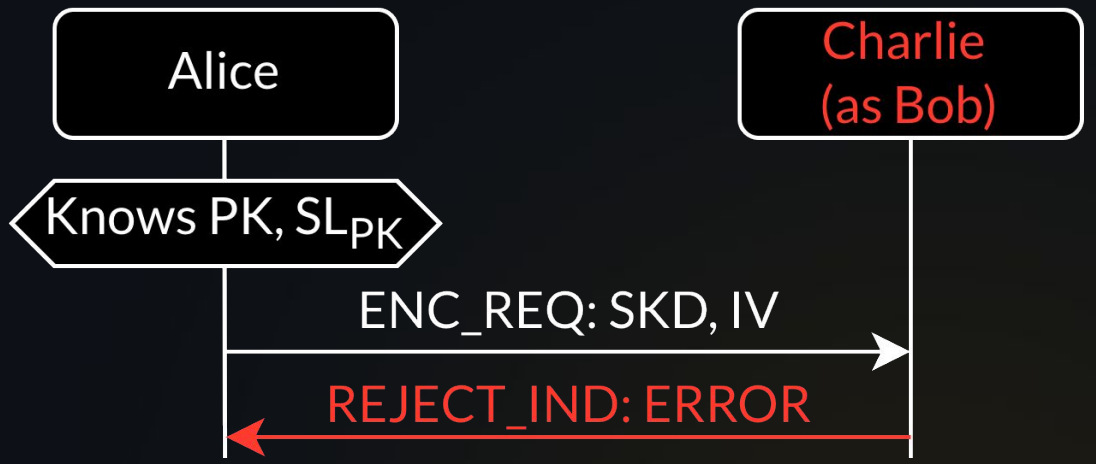
1. Peripheral Impersonation
2. Central Impersonation
3. Single-Channel MitM
4. Double-Channel MitM

BLERP Peripheral Impersonation

Charlie impersonates **Bob**, **Alice** connects with him and starts session establishment.

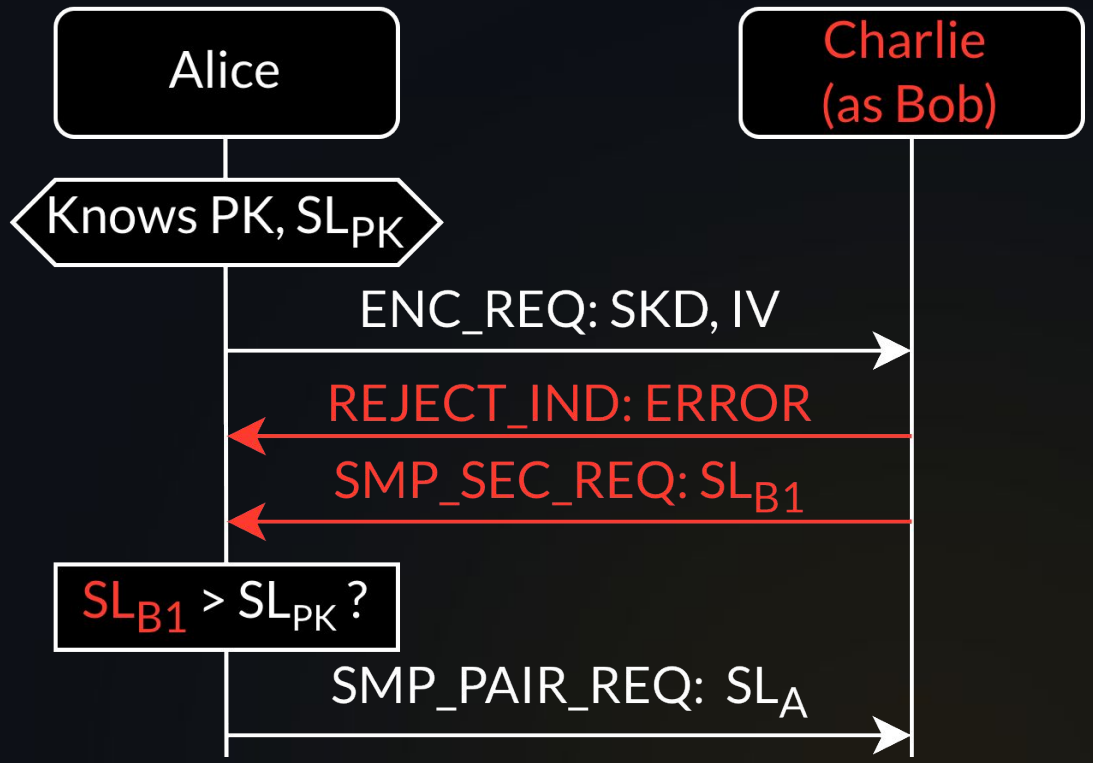
BLERP Peripheral Impersonation (2)

Charlie aborts session establishment and remains connected to **Alice (V5)**



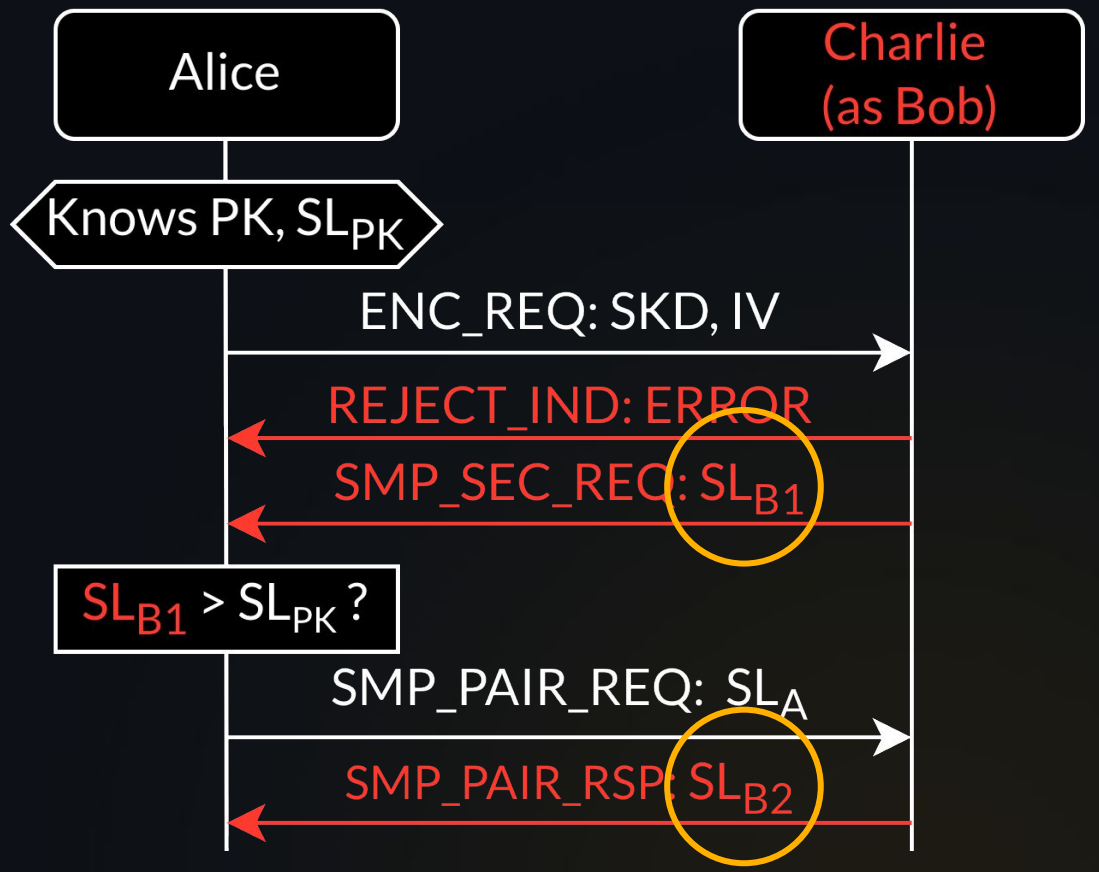
BLERP Peripheral Impersonation (3)

Charlie asks **Alice** to re-pair as **Bob** without authenticating **(V2)**



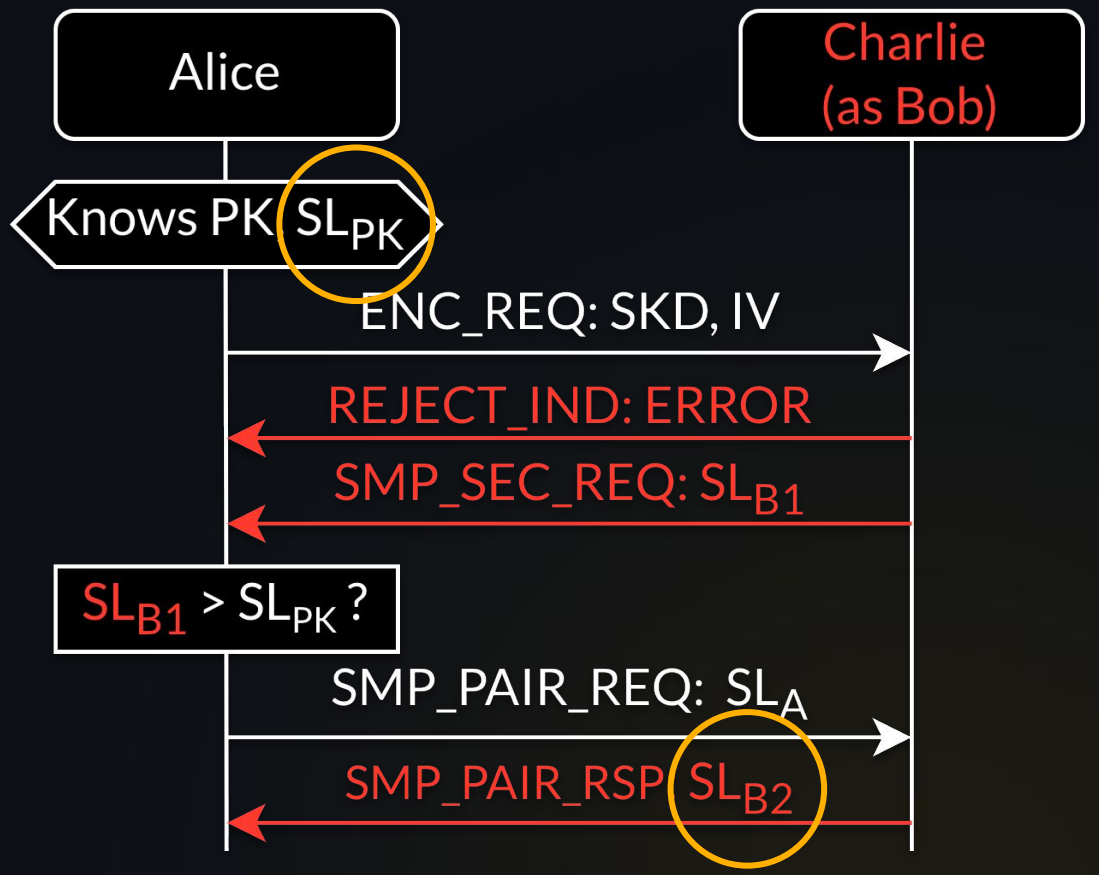
BLERP Peripheral Impersonation (4)

Charlie can downgrade the pairing response security level: $SL_{B2} < SL_{B1}$ **(V3)**



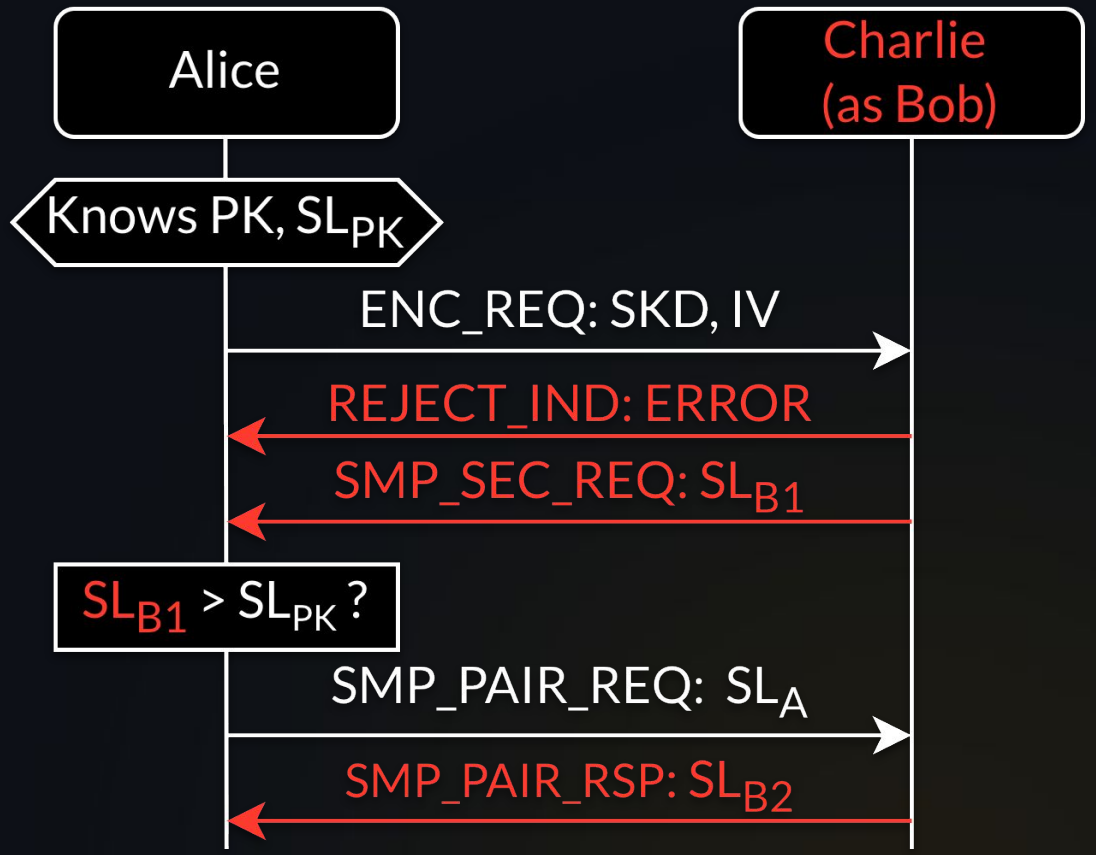
BLERP Peripheral Impersonation (5)

Charlie can also downgrade the previous pairing security level: $SL_{B2} < SL_{PK}$ **(V4)**



BLERP Peripheral Impersonation (6)

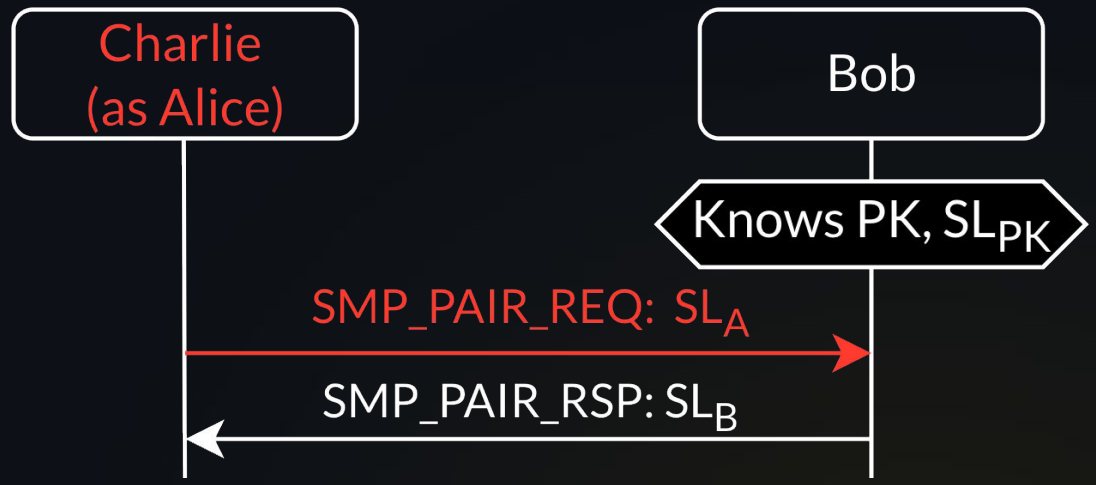
Alice pairs with **Charlie** but thinks she re-paired with **Bob**!



Demo: BLERP Peripheral Impersonation

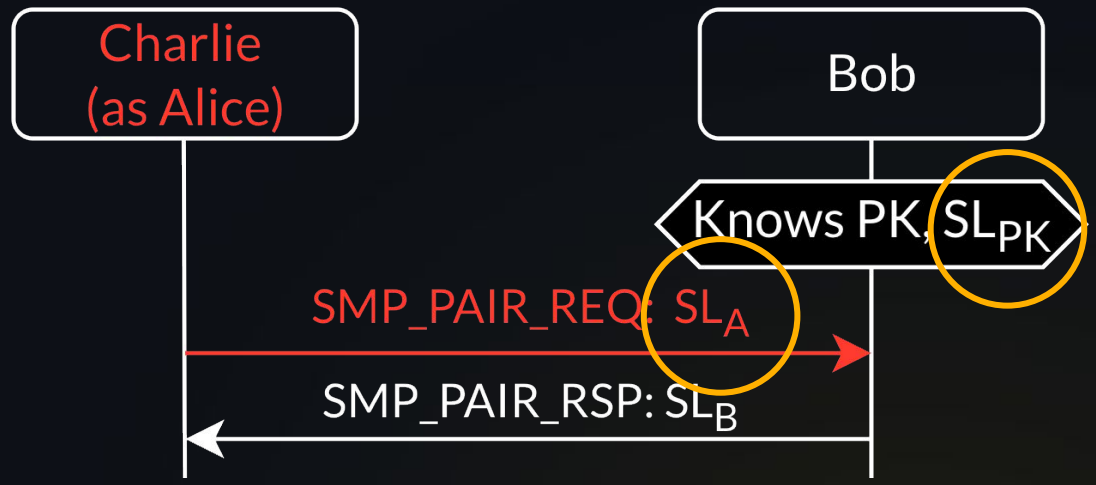
BLERP Central Impersonation

Charlie start re-pairing with **Bob** as **Alice** without authenticating **(V1)**



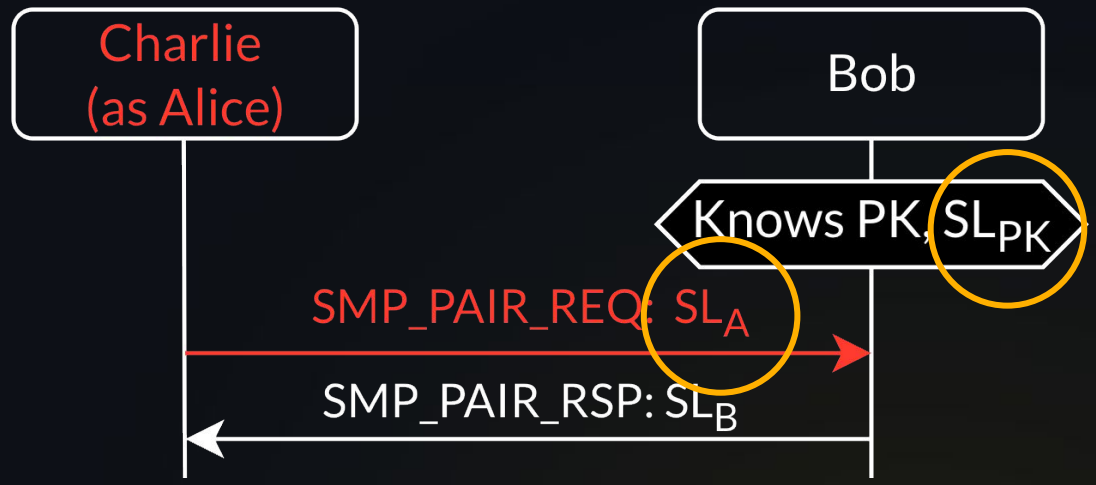
BLERP Central Impersonation (2)

Charlie can downgrade the previous pairing security level $\mathbf{SL_A} < SL_{PK}$

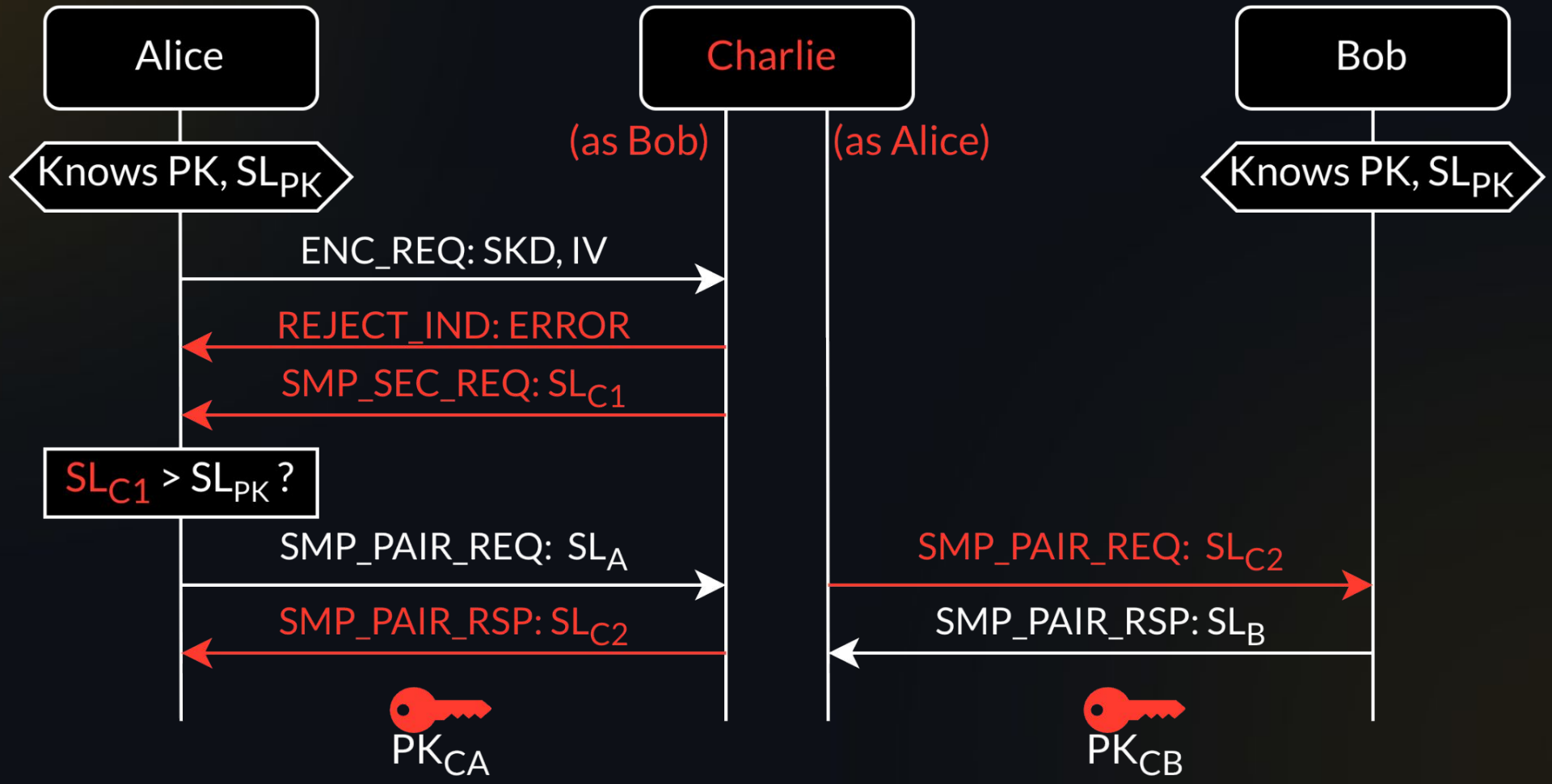


BLERP Central Impersonation (3)

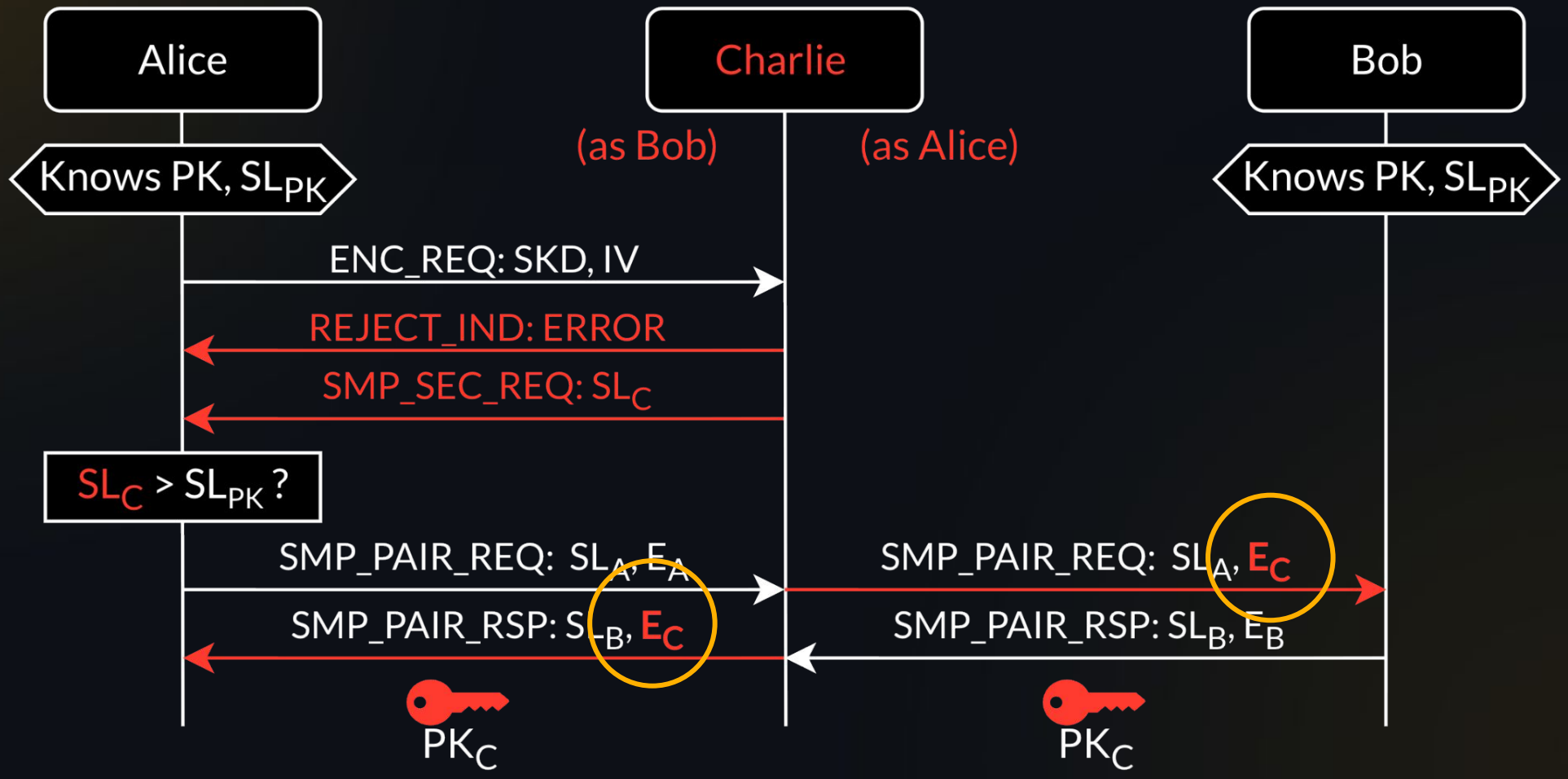
Bob pairs with **Charlie** but thinks he re-paired with **Alice**!



BLERP Double-Channel MitM



BLERP Single-Channel MitM



BLERP Single-Channel MitM (2)

Charlie reduces the re-pairing PK entropy **(V6)**

- 7 bytes → bruteforcing possible
- PK knowledge → **No security** anymore

Impact of the BLERP Attacks

- **Zero-click** for devices **w/o display**
  - **One-click** otherwise
- Impersonation
  - Central (smartphone) → unlock smart-lock
  - Peripheral (smartwatch) → access sensitive data (health, …)
- MitM
  - Full connection compromise (eavesdrop, tamper, …)

Six BLERP Vulnerabilities

V1 No Central re-pairing authentication
V2 No Peripheral re-pairing authentication
V3 Pairing response SL downgrade
V4 Re-pairing SL downgrade
V5 Re-pairing after session establishment error
V6 Re-pairing PK entropy downgrade

Toolkit and Evaluation

BLERP Toolkit

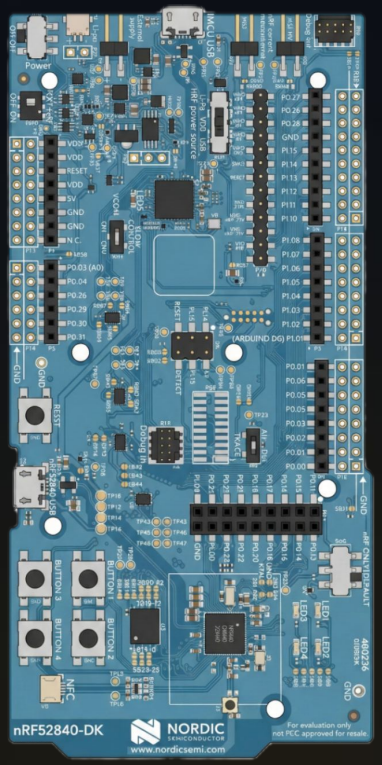
Testing the BLERP attacks

- nRF52 hardware
- NimBLE stack w/ mods
- CLI application
- Python Host
- https://github.com/sacca97/blerp

BLERP Toolkit (2)

CLI application

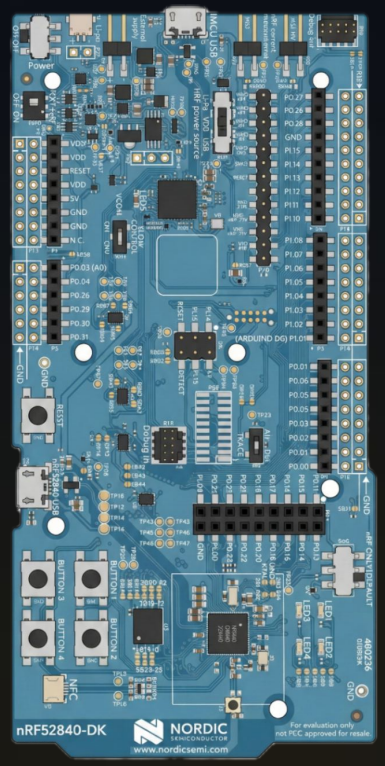
- Self-contained (runs on nRF5x)
- Interaction via CLI (over UART/USB)
- Central / Peripheral Impersonation



BLERP Toolkit (3)

Python Host

- Runs on Linux + nRF5x
- Double-Channel MitM
- Advertisements cloning



Evaluation (2024)

Exploited 23 BLE devices

- 13 vendors (Apple, Google, Xiaomi, Samsung, ...)
- 16 Centrals, 9 Peripherals
- Bluetooth v4.2 to v5.3 (most popular)
- All security modes (LSC, SC, SCO, MitM, ...)

Centrals Evaluation

	BLEv	V1	V3	V4	V5	V6	PI
MacBook Air	5.0	❌	❌	❌	❌	❌	❌
Windows 11	5.2	❌	❌	❌		❌	
Linux 6.10.9	5.2	❌	❌	❌		❌	
Oculus Quest	5.0	❌	❌	❌	❌	❌	❌
iPhone 15	5.3	❌	❌	❌	❌	❌	❌
iPad 2022	5.2	❌	❌	❌	❌	❌	❌
Google Pixel 8 *	5.3	❌	❌	❌	❌	❌	❌
Realme X2 Pro	5.0	❌	❌	❌	❌	❌	❌
Xiaomi Mi 11 Lite	5.1	❌	❌	❌	❌	❌	❌
Samsung Galaxy A15	5.3	❌	❌	❌	❌	❌	❌
TCL 43P638	5.0	❌	❌	❌	❌	❌	❌

* = Tested on Android 15, 16, and Graphene OS

❌ = vulnerable

Peripherals Evaluation

	BLEv	V2	V4	V6	CI
Xbox Controller	5.0	❌	❌	❌	❌
Logitech MX Anywhere 3S	5.1	❌	❌	❌	❌
Logitech MX Keys S	5.1	❌	❌	❌	❌
Logitech MX Master 3	4.2	❌	❌	❌	❌
Garmin Vivoactive 5	5.0	❌	❌	❌	❌

Centrals and Peripherals Evaluation

	BLEv	V1	V2	V3	V4	V5	V6	PI	CI
NimBLE	5.4	❌	❌	❌	❌	❌	❌	❌	❌
NimBLE SC-Only	5.4	❌	❌	❌	❌	❌		❌	❌
Zephyr	5.4	❌	❌		❌	❌		❌	❌
BTstack	5.2		❌	❌	❌	❌	❌		❌
ESP32-C3	5.0	❌	❌	❌	❌		❌		❌

Mitigations and Countermeasures

BLERP Countermeasures

Authenticate re-pairing (implicit auth.)

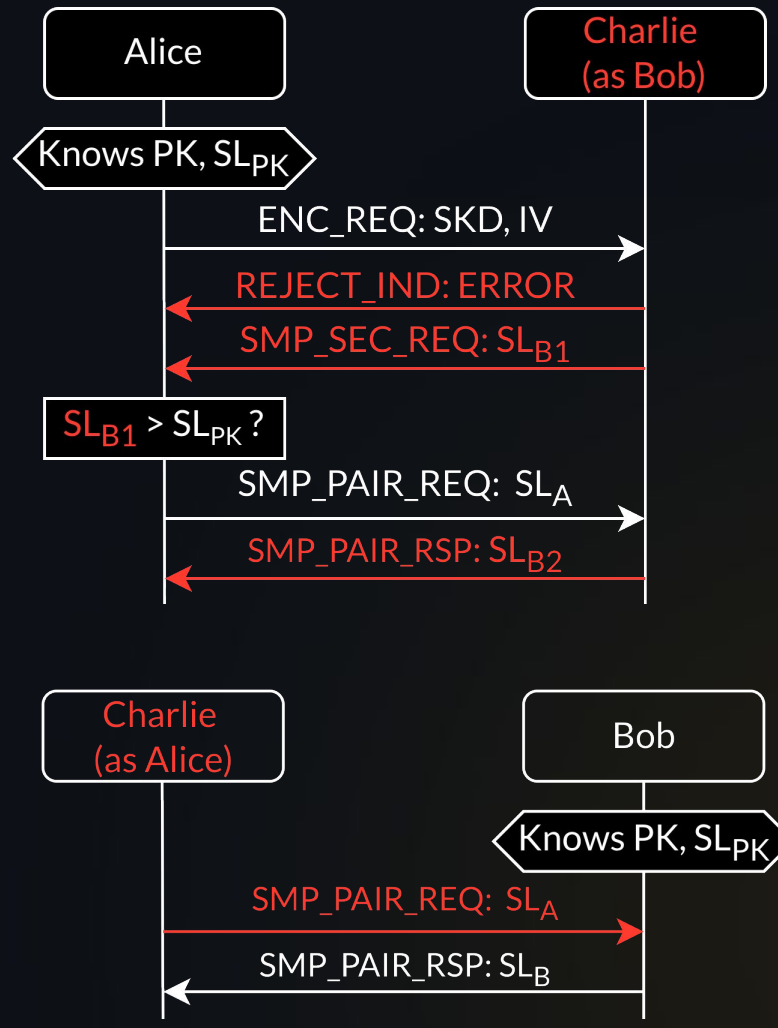
- $PK_{NEW} = f(PK_{OLD})$
- Attacker doesn't have PK_{OLD}
- Blocks PI, CI, Double-Channel MitM

Add integrity protection

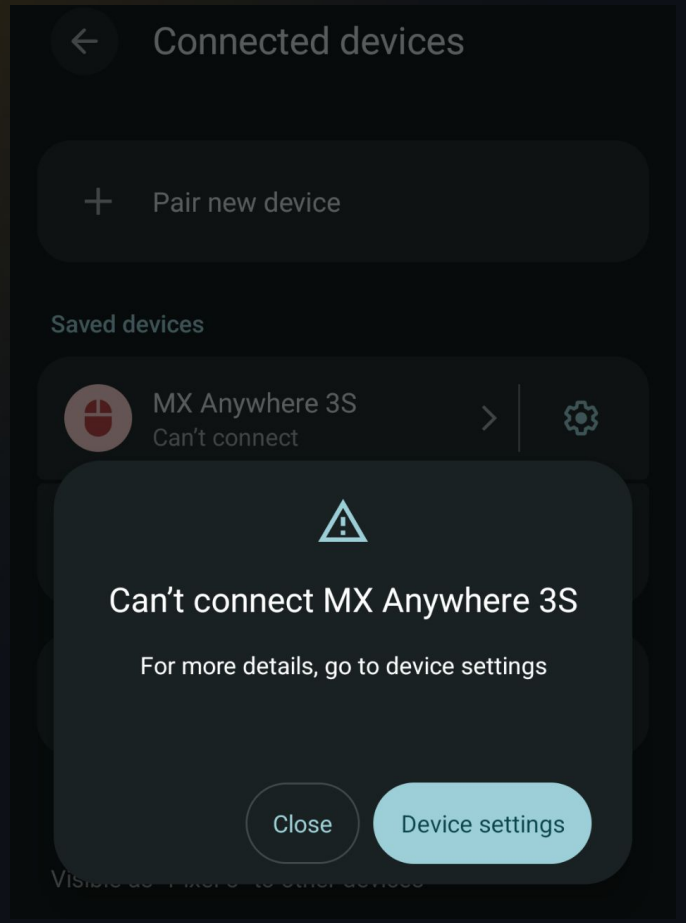
- Protocol hash transcript (H)
- $PK_{NEW} = f(PK_{OLD}, H)$
- Blocks tampering in Single-Channel MitM

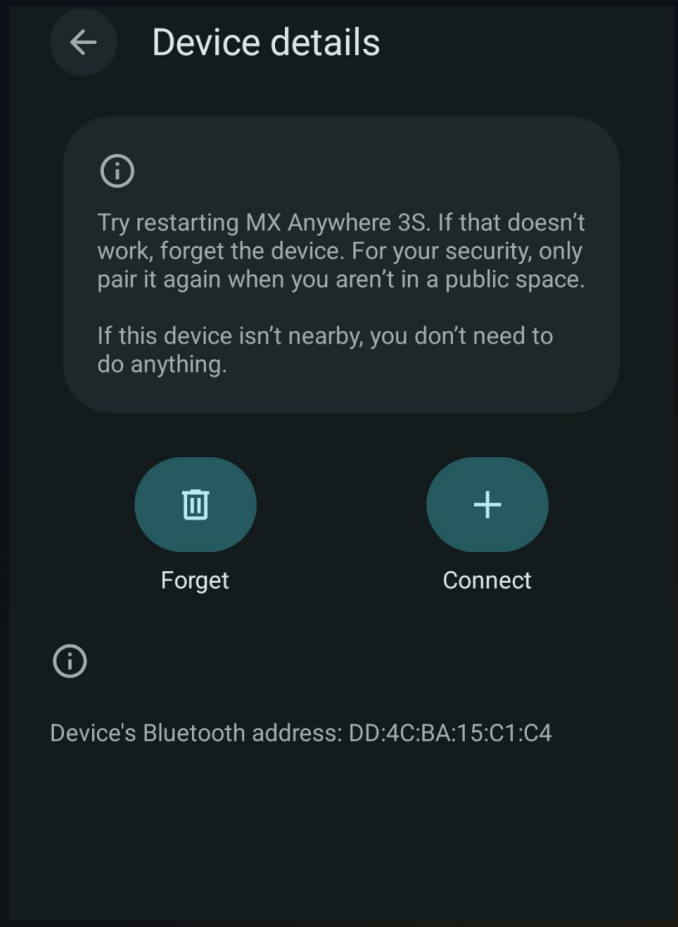
BLERP Countermeasures (2)

- Disconnect after error
- Enforce SL_{B1} == SL_{B2}
- Enforce SL_{B2} > SL_{PK}
- Optional: warn user



BLERP Countermeasures (3)





Disclosure and Takeaways

Bluetooth SIG Disclosure

Bluetooth SIG (August 2024)

- “known attack techniques used in novel ways…”
- “Re-pairing is a new pairing, there’s no downgrade, but could enforce security level”

No updates, standard still vulnerable to BLERP (2026)

Vendor Disclosure

Google

- Acknowledged and fixed

Logitech

- “Intended behavior”

Microsoft

- “Not exploitable, won’t fix”

Vendor Disclosure (2)

Apple

- Acknowledged and fixed

Apache (NimBLE)

- Acknowledged and fixed
- CVE-2025-62235, CVSS 8.1

Takeaways

1. BLE **is** vulnerable to the re-pairing attacks
   - Standard has not been fixed
   - Many vendors may still be vulnerable

2. BLE threat model must account for re-pairing
   - Attackers can conduct BLERP at will
   - And chain other pairing attacks (KNOB, ...)

Conclusion and Q&A

- First security evaluation of **BLE re-pairing**
  - Six **vulnerabilities** (no auth, …)
  - Four **attacks** (Impersonation, MitM, 0-click, …)
  - Evaluated on 23 devices (SC, MitM, …)
- Open-source BLERP **toolkit** and **demos**
- Responsible disclosure (CVE-2025-62235)