HardaBLE:

Hardening BLE Against Software Compromise

Tommaso Sacchetti, Daniele Antonioli, Norrathep Rattanavipanon

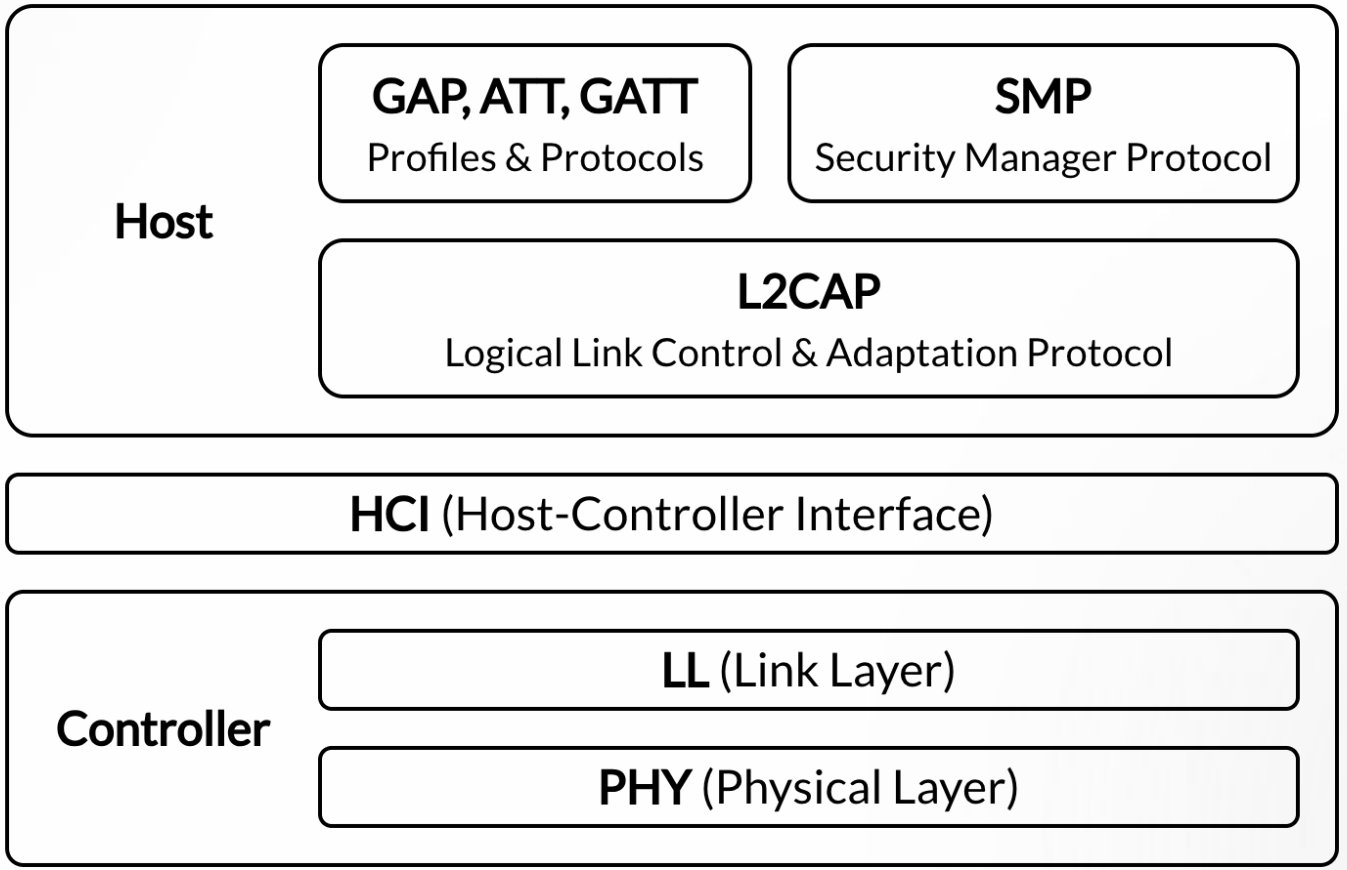
ACM WiSec 2026, Saarbrucken, Germany

Bluetooth Low Energy (BLE)

Ubiquitous wireless technology

- Used by IoT devices, wearables, and smartphones
- Low-power alternative to Bluetooth Classic
- Specified in an open standard (v6.3)
- Connection-oriented and connectionless (e.g., tracking)

BLE Stack


Host
GAP, ATT, GATT
Profiles & Protocols
SMP
Security Manager Protocol
L2CAP
Logical Link Control & Adaptation Protocol
HCI (Host-Controller Interface)
Controller
LL (Link Layer)
PHY (Physical Layer)

BLE Security Protocols

Pairing

- Establish Long-Term Key (LTK)
- Implemented in the Host (SMP)

Session Establishment

- Derives Session Key (SK) for encryption
- Implemented in the Controller (Link Layer)

Motivation

BLE security focuses on protocol-level issues

- Assumes uncompromised software

Devices have implementation-level flaws

- Enabling software compromise (e.g., code execution)

Attacker with arbitrary code execution

- Can extract secrets (e.g., LTK) and use them

Motivation (2)

BLE cannot protect secrets if software is compromised

- Hardware isolation and secure storage protect keys at rest
- **Not** during protool execution

A New Threat Model

Attacker achieved software compromise

- Arbitrary code execution
- Read and write access

Device has

- Secure Processing Environment (SPE)
- Non-Secure Processing Environment (NSPE)

Security Goals

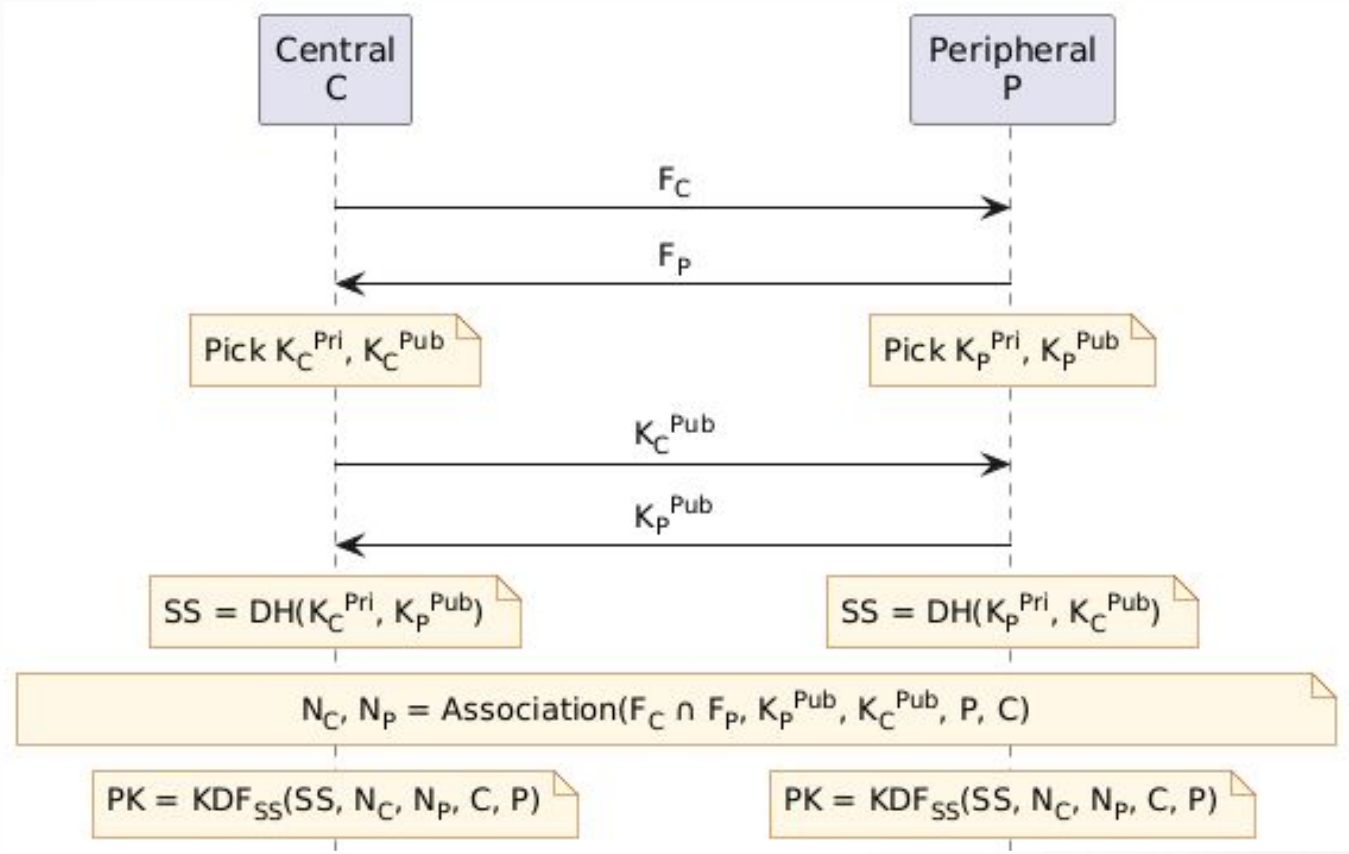
1. **LTK Confidentiality**
   - Attackers cannot obtain LTK
2. **Session Key (SK) Confidentiality**
   - Attackers cannot obtain SK
3. **Integrity-bound Authorization**
   - Block LTK / SK usage if compromised

HardaBLE

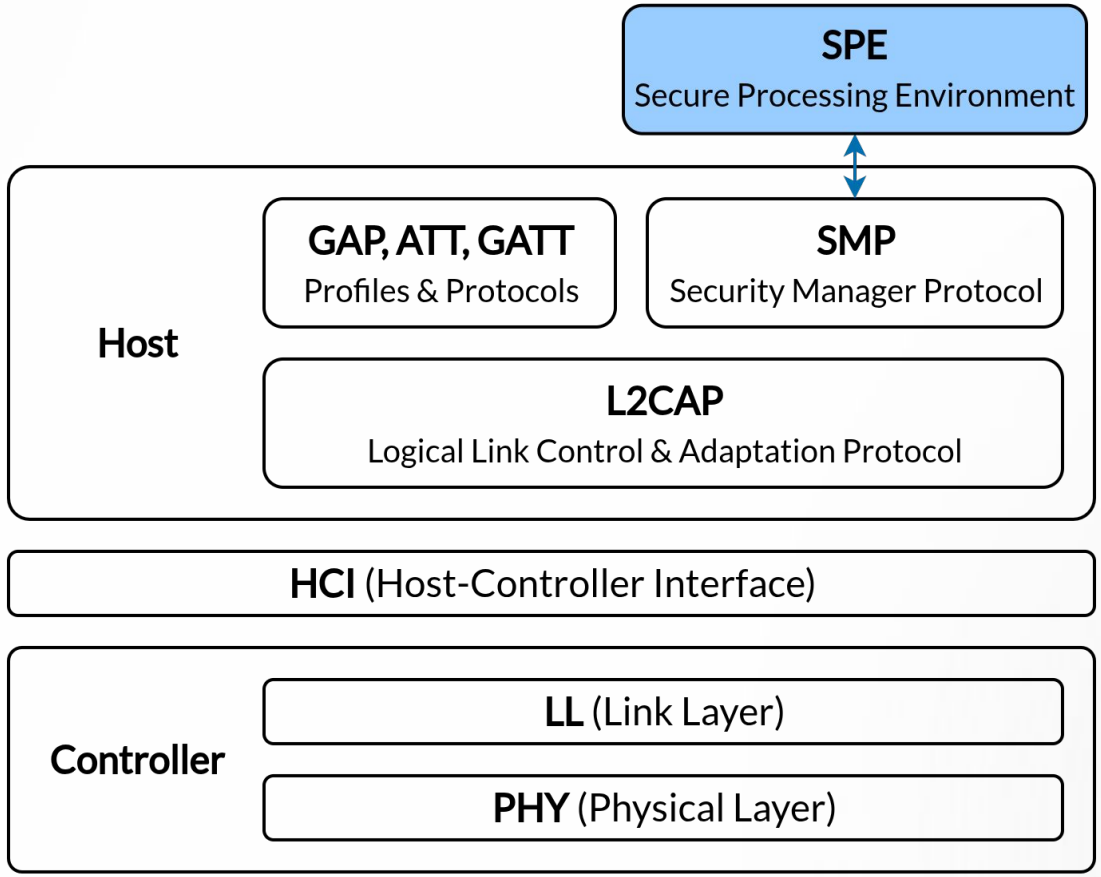
A New Hardened BLE Stack Design

- Pairing and Session Establishment leverage the SPE
- Mitigates software compromise
- Preventing key extraction and usage under compromise

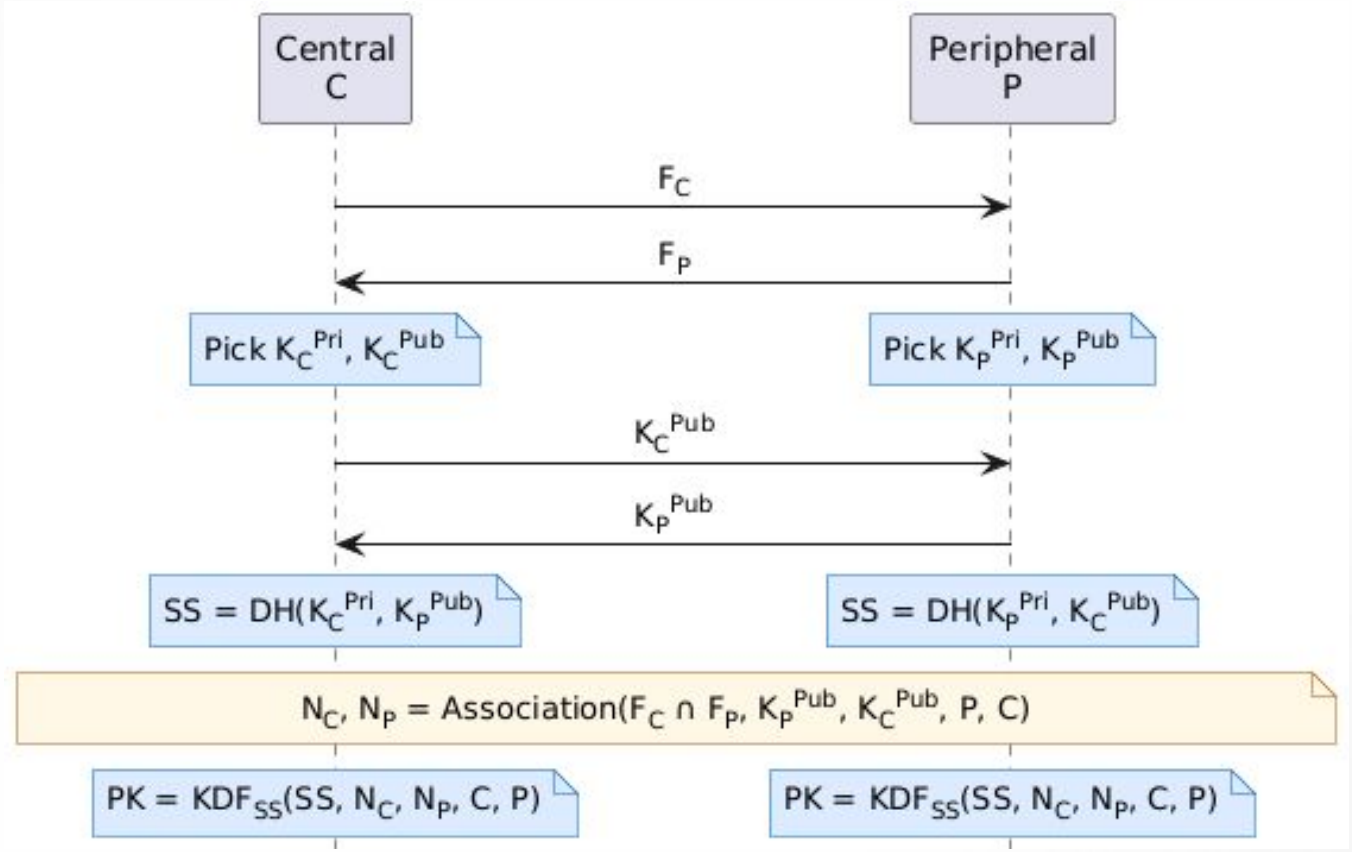
BLE Pairing


Central
C
Peripheral
P
F_C
F_P
Pick K_C^{Pri}, K_C^{Pub}
Pick K_P^{Pri}, K_P^{Pub}
K_C^{Pub}
K_P^{Pub}
$SS = DH(K_C^{Pri}, K_P^{Pub})$
$SS = DH(K_P^{Pri}, K_C^{Pub})$
$N_C, N_P = Association(F_C \cap F_P, K_P^{Pub}, K_C^{Pub}, P, C)$
$PK = KDF_{SS}(SS, N_C, N_P, C, P)$
$PK = KDF_{SS}(SS, N_C, N_P, C, P)$

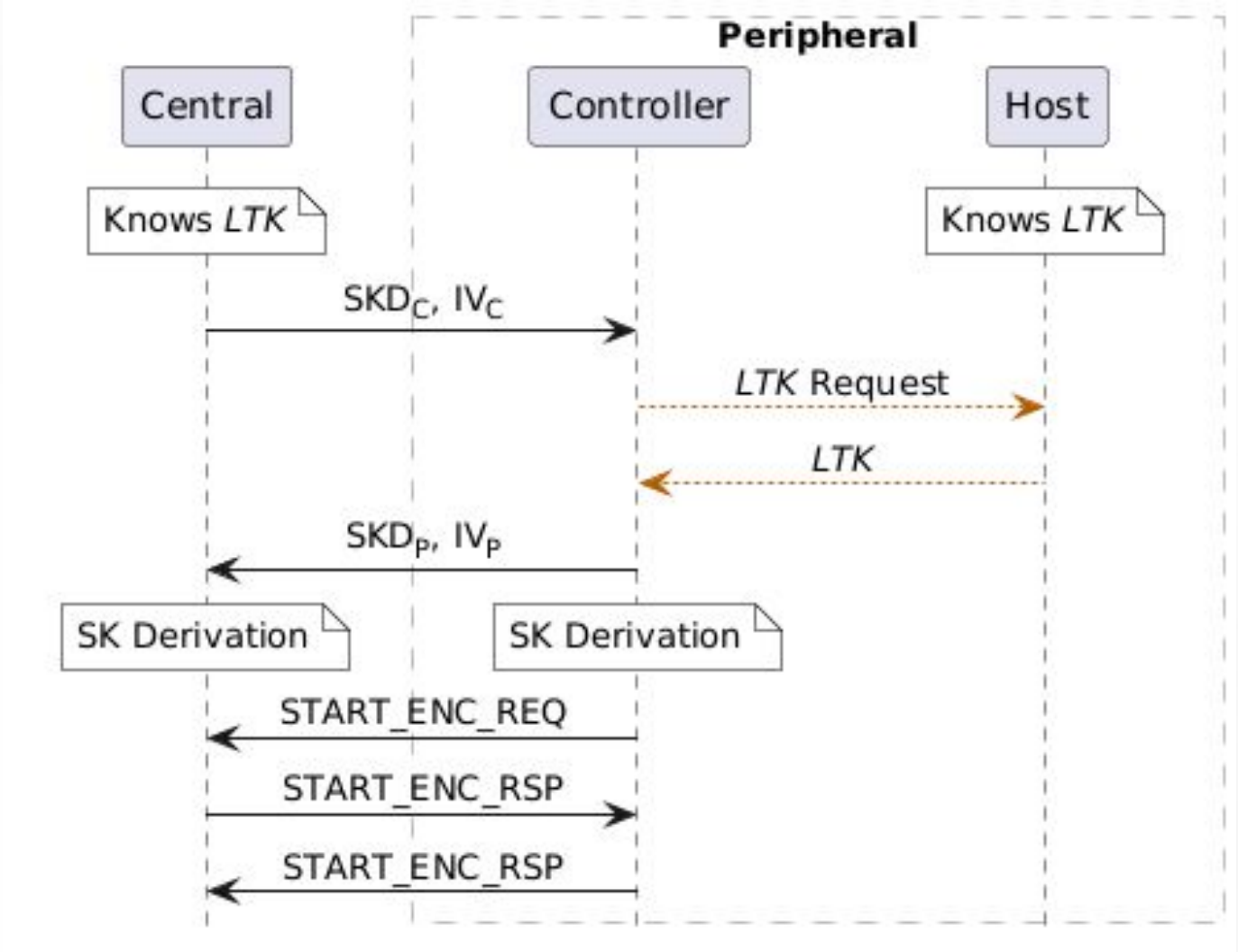
HardaBLE: Pairing



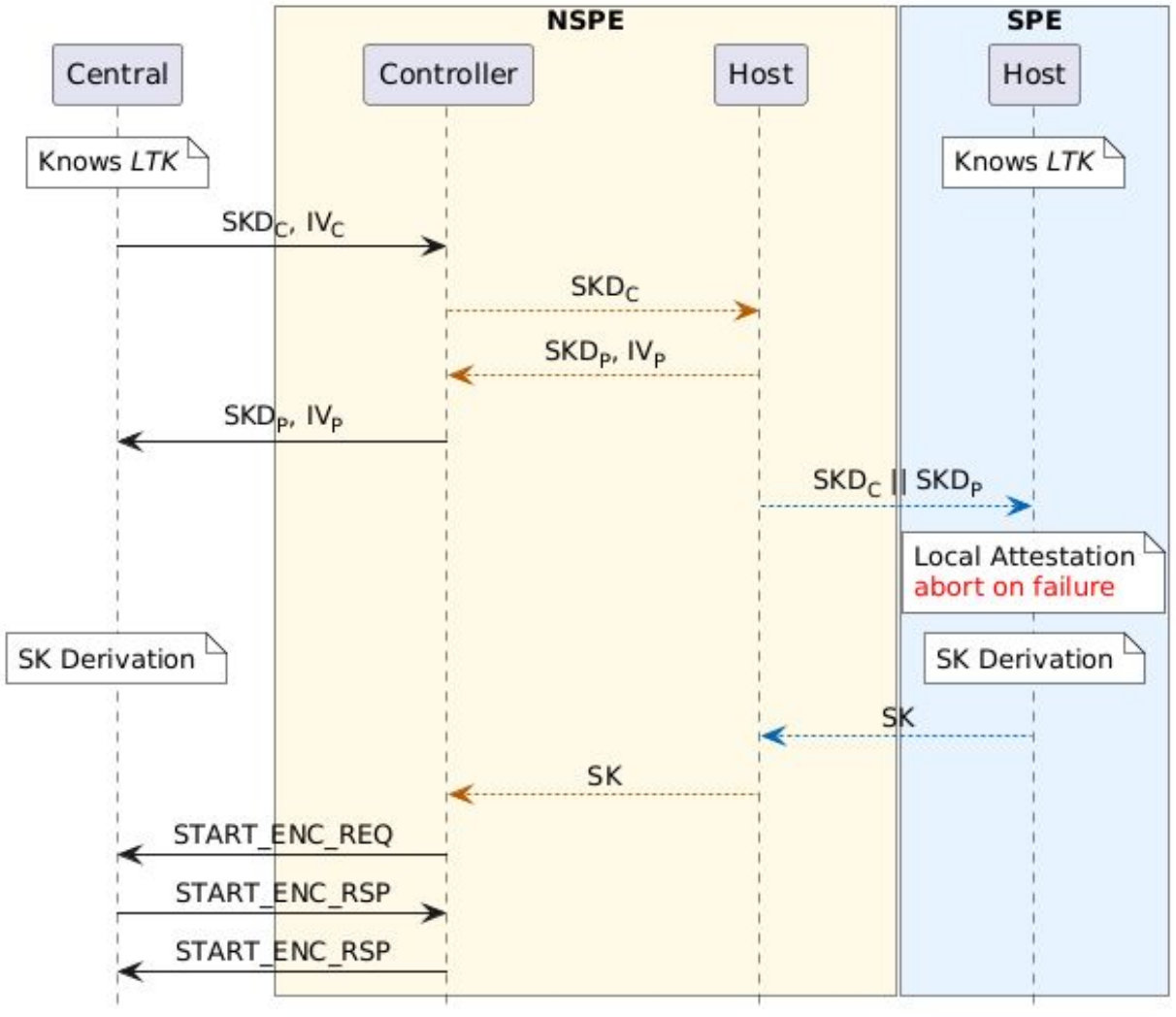
HardaBLE: Pairing



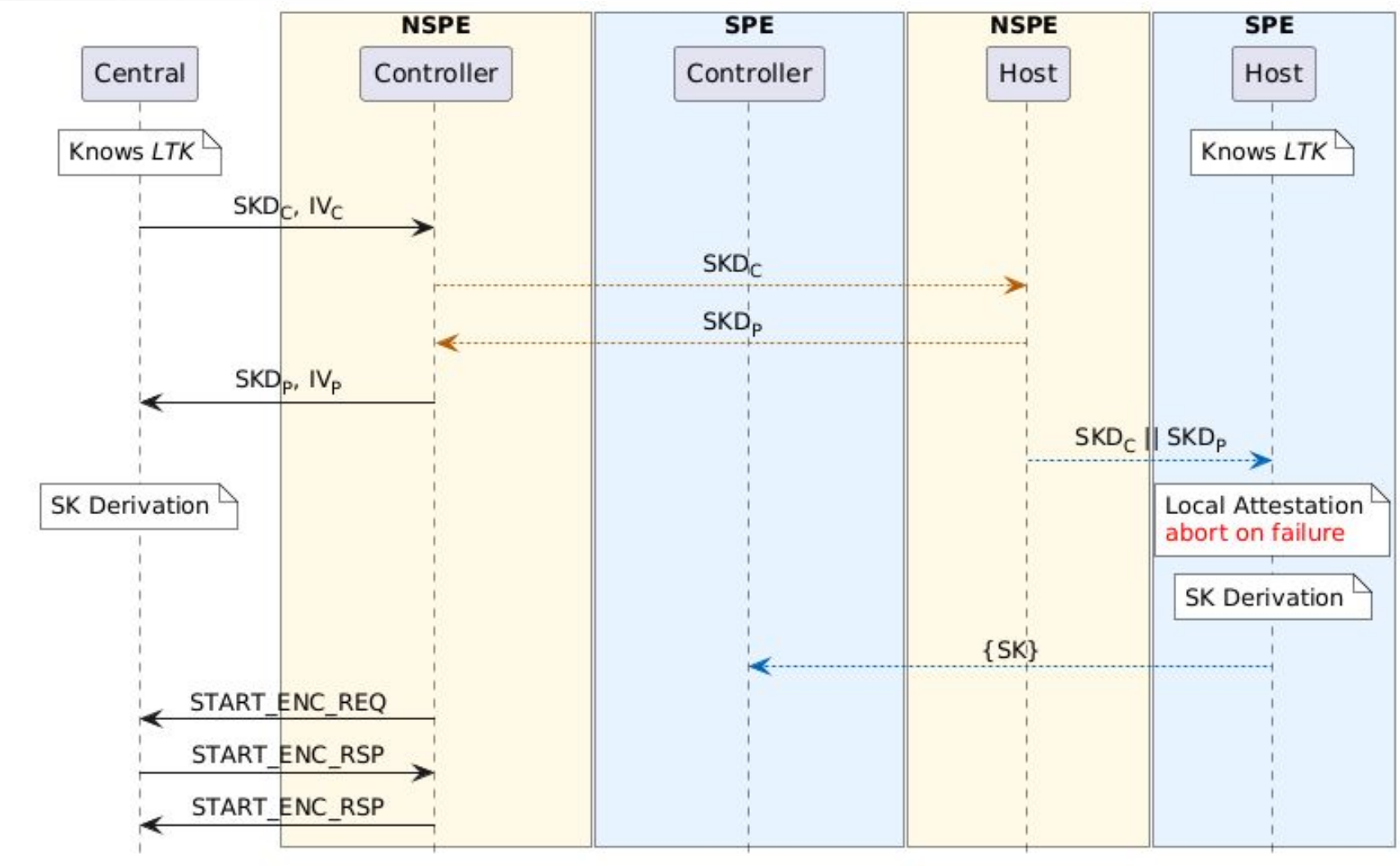
BLE Session Establishment



HardaBLE: Session Establishment



HardaBLE: Session Establishment (2)



Leveraging the SPE

1. MCUboot metadata locates NSPE
2. Compute SHA-256 memory hash
3. Compare **hash** with provisioned **golden hash**
    a. Stored in SPE
4. Mismatch = Compromise → SPE denies operations

Security Evaluation

Formal verification with TLA+

- Verified architecture-level trust boundaries
- Proved keys are never exposed to NSPE

Empirical implementation testing

- Simulated code execution in NSPE
- Successfully blocked key exfiltration and usage

Implementation

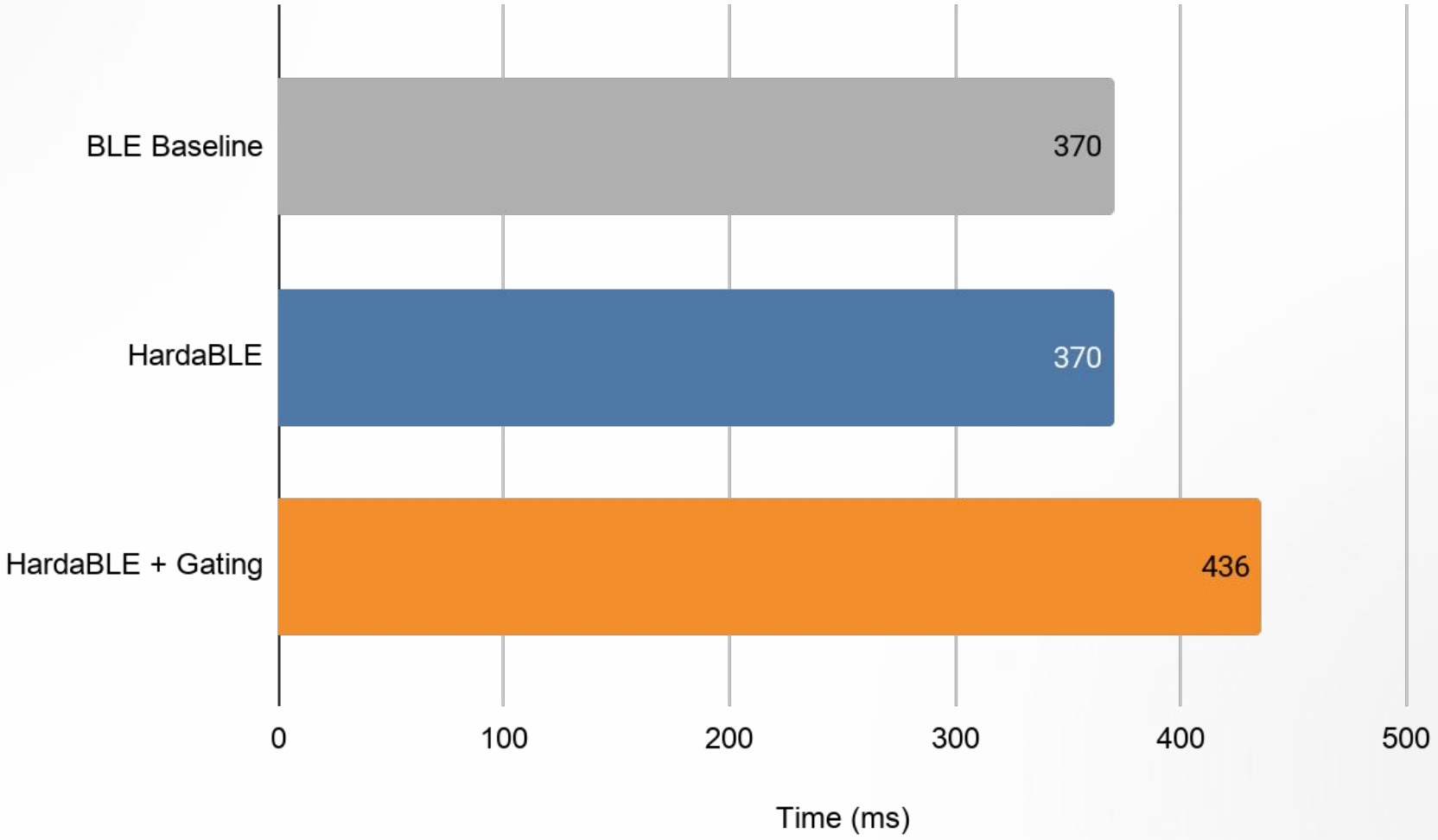
- **Hardware:**
  - Nordic nRF5340 SoC (TrustZone-M)
- **Operating System:**
  - Zephyr RTOS and BLE stack
- **Secure World:**
  - TrustZone-M with Trusted Firmware-M (TF-M)

Performance Evaluation

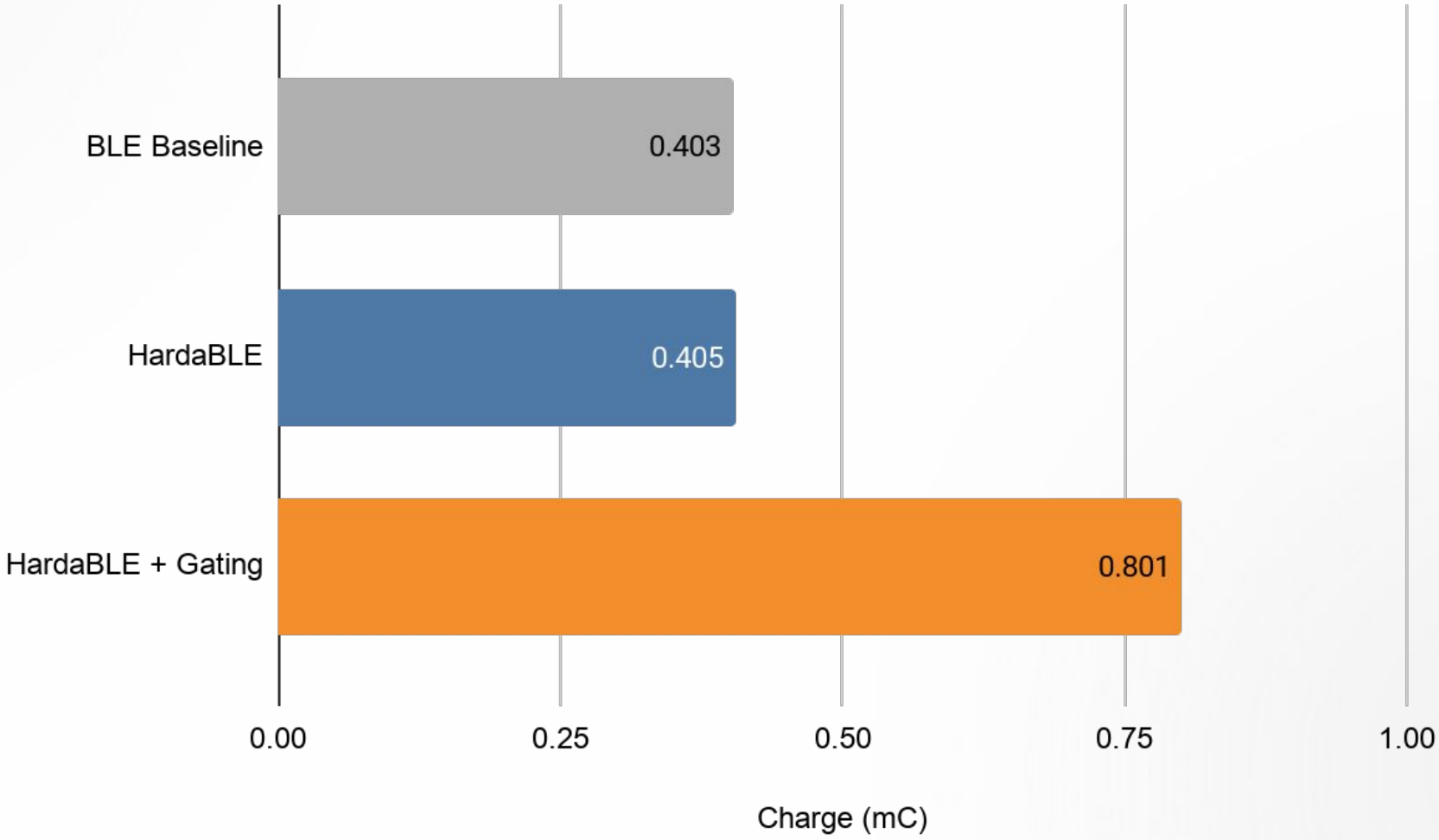
Session Establishment

- 1 device running HardaBLE
- 1 device running standard BLE

Evaluation Results: Latency



Evaluation Results: Energy



Conclusion

HardaBLE: a New Hardened BLE stack design

- Leverages SPE to mitigate software compromise
  - Prevents key extraction and usage
- Proven secure
- Working implementation on **ZephyrOS** and **nRF53**