# BlueBrothers:

## Three New Protocols to Secure Bluetooth

**Tommaso Sacchetti**, Kasper Rasmussen, Daniele Antonioli

ACM WiSec 2026, Saarbrucken, Germany

# Bluetooth

## Ubiquitous wireless technology

- Bluetooth Low Energy (BLE)
- Bluetooth Classic (BC)
- Specified in an open standard (v6.3)
- Connection-oriented and connectionless
  - Tracking
  - Audio streaming

# Bluetooth Security

## Pairing

- Establishes Long-Term Key (LTK)
- Optionally authenticated (e.g., via user interaction)

## Session Establishment

- Short-term Session Key (SK)
- Used for encryption

# Bluetooth Security Problems

## Reviewed literature and latest specification

- 17 known **working** attacks

## Stemming from four root causes

- No Message Integrity (C1)
- No Replay / reflection resistance (C2)
- No Forward or backward secrecy (C3)
- Weak authentication (C4)

# Bluetooth Security Problems (2)

| 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. |

# Bluetooth Security Problems (3)

**Informal, complex, and fragmented specification**

- 3800+ pages

**No reference implementation**

- No fully open-source BC stack

**Legacy functionalities**

- Key entropy negotiation
- 7-byte key → easy to brute-force

# Why BlueBrothers?

**Bluetooth needs new security protocols**

- Address design (C1 - C4) and complexity issues
- Introduce missing security properties
- Deprecate legacy functionalities (e.g., entropy negotiation)

# BlueBrothers: Three New Protocols

**BB-Pairing**

- Establishes root of trust w/ public keys

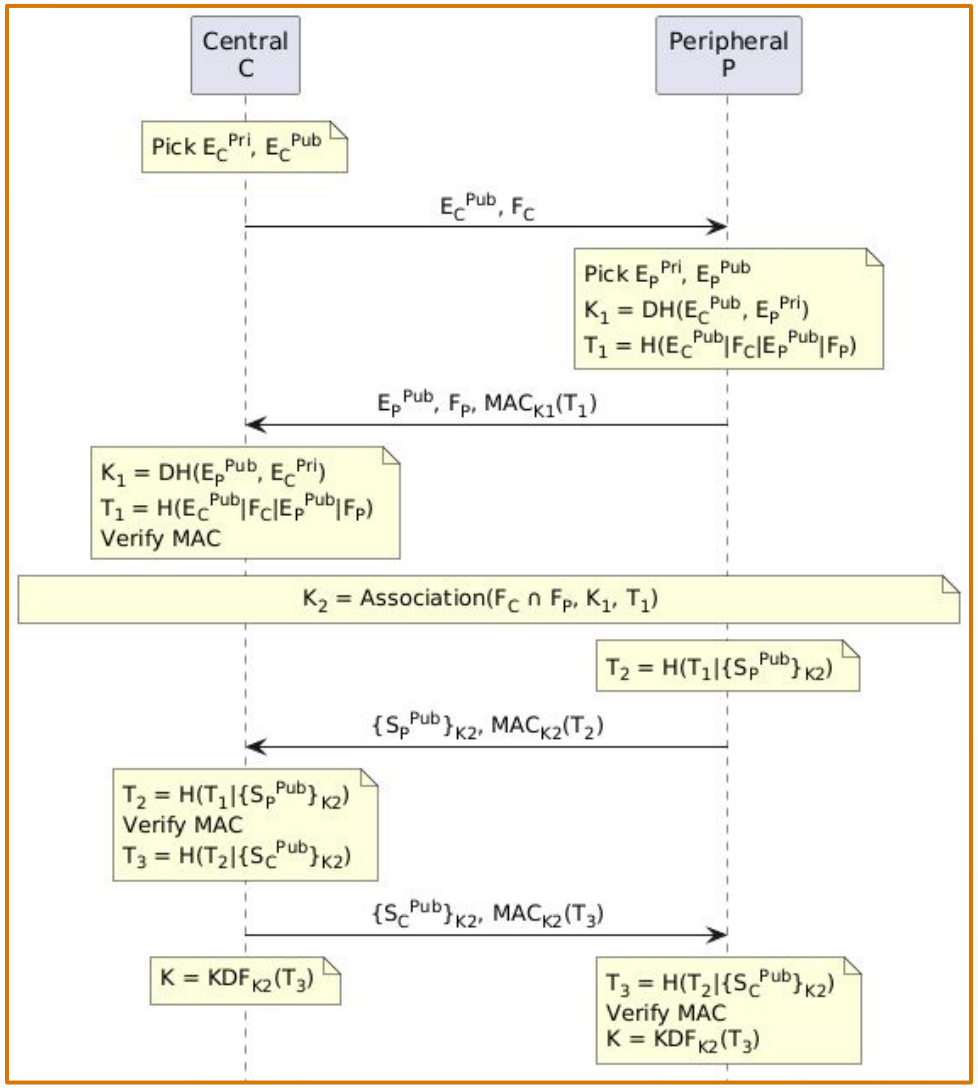
**BB-Session**

- Authenticated session establishment

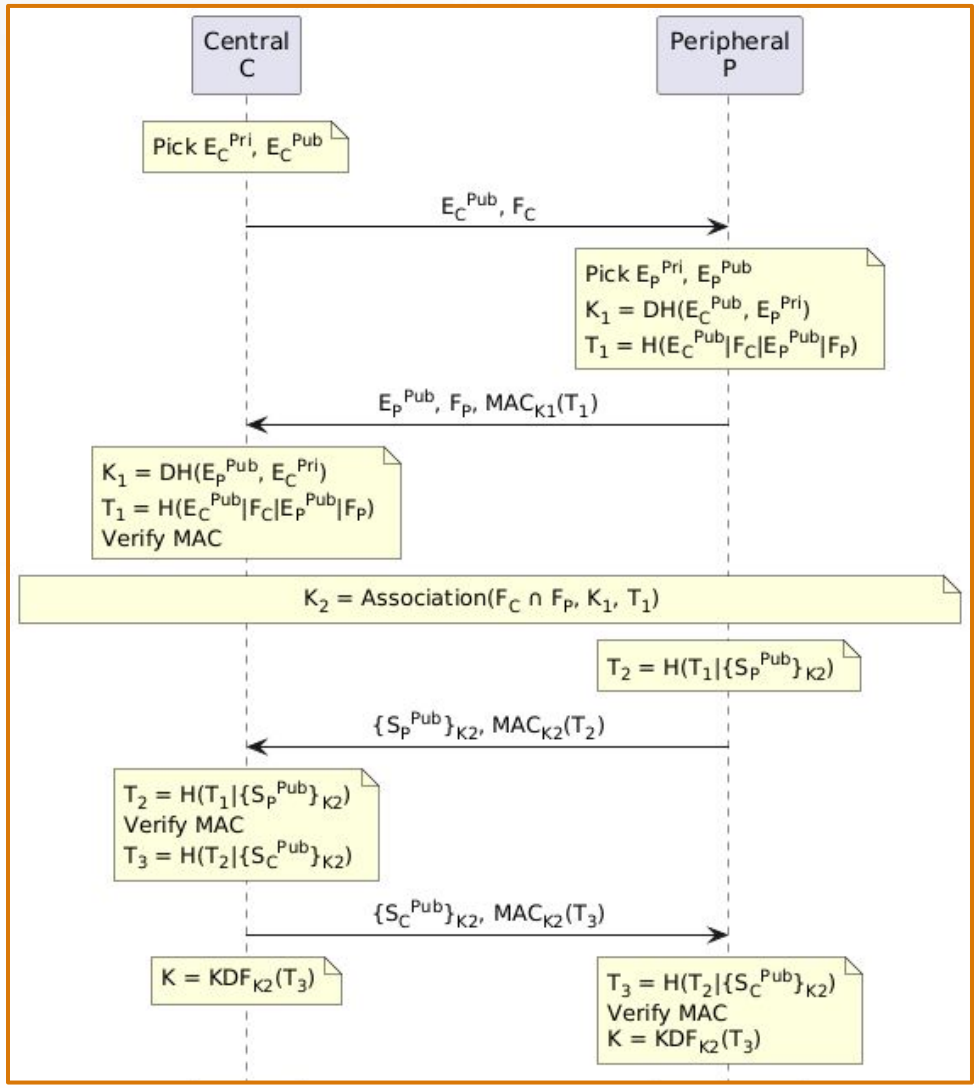
**BB-Rekey**

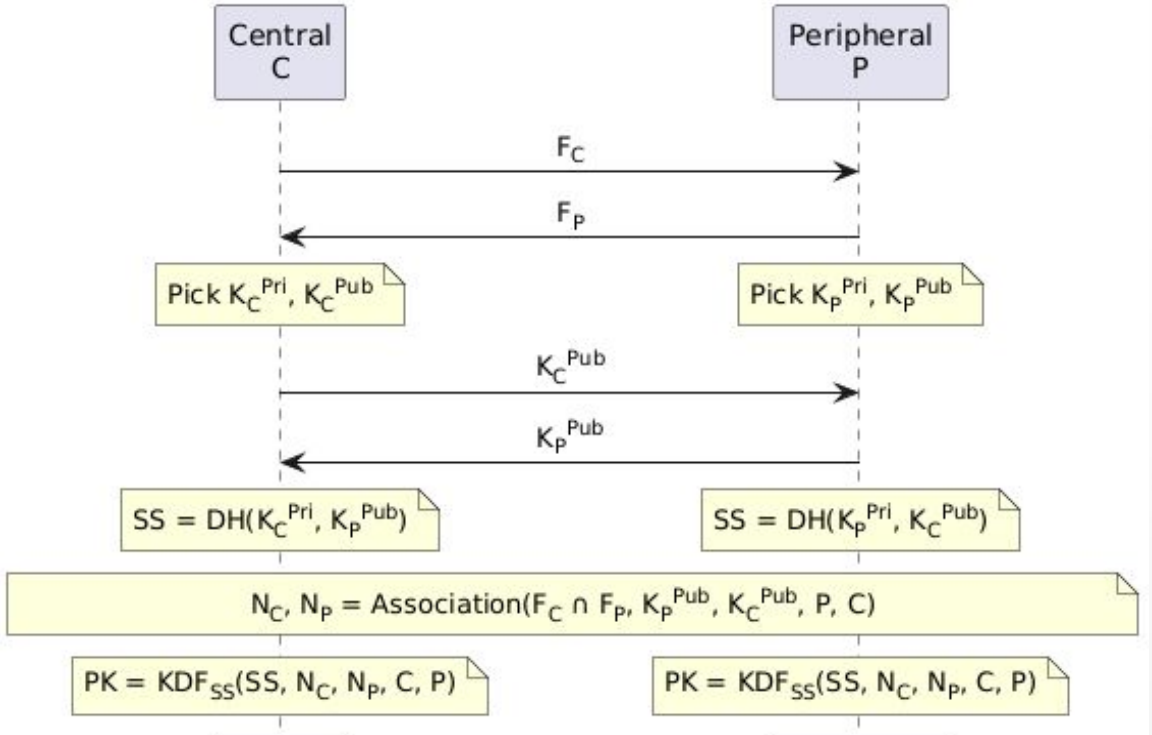
- Intra-session forward and backward secrecy

# BB-Pairing

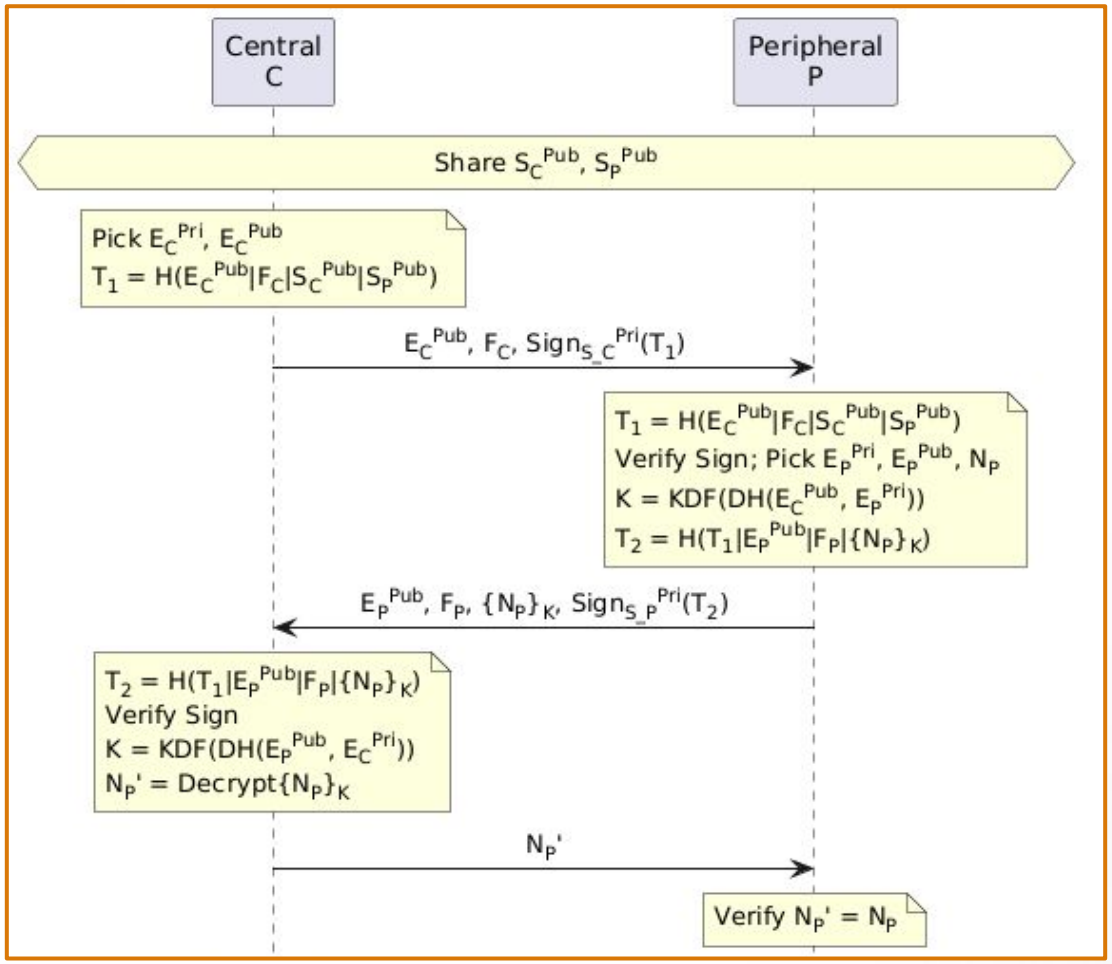

Central
C
Peripheral
P
Pick $E_C^{Pri}$, $E_C^{Pub}$
$E_C^{Pub}$, $F_C$
Pick $E_P^{Pri}$, $E_P^{Pub}$
$K_1 = DH(E_C^{Pub}, E_P^{Pri})$
$T_1 = H(E_C^{Pub}|F_C|E_P^{Pub}|F_P)$
$E_P^{Pub}$, $F_P$, $MAC_{K1}(T_1)$
$K_1 = DH(E_P^{Pub}, E_C^{Pri})$
$T_1 = H(E_C^{Pub}|F_C|E_P^{Pub}|F_P)$
Verify MAC
$K_2 = Association(F_C \cap F_P, K_1, T_1)$
$T_2 = H(T_1|\{S_P^{Pub}\}_{K2})$
$\{S_P^{Pub}\}_{K2}$, $MAC_{K2}(T_2)$
$T_2 = H(T_1|\{S_P^{Pub}\}_{K2})$
Verify MAC
$T_3 = H(T_2|\{S_C^{Pub}\}_{K2})$
$\{S_C^{Pub}\}_{K2}$, $MAC_{K2}(T_3)$
$K = KDF_{K2}(T_3)$
$T_3 = H(T_2|\{S_C^{Pub}\}_{K2})$
Verify MAC
$K = KDF_{K2}(T_3)$

# BB-Pairing (2)

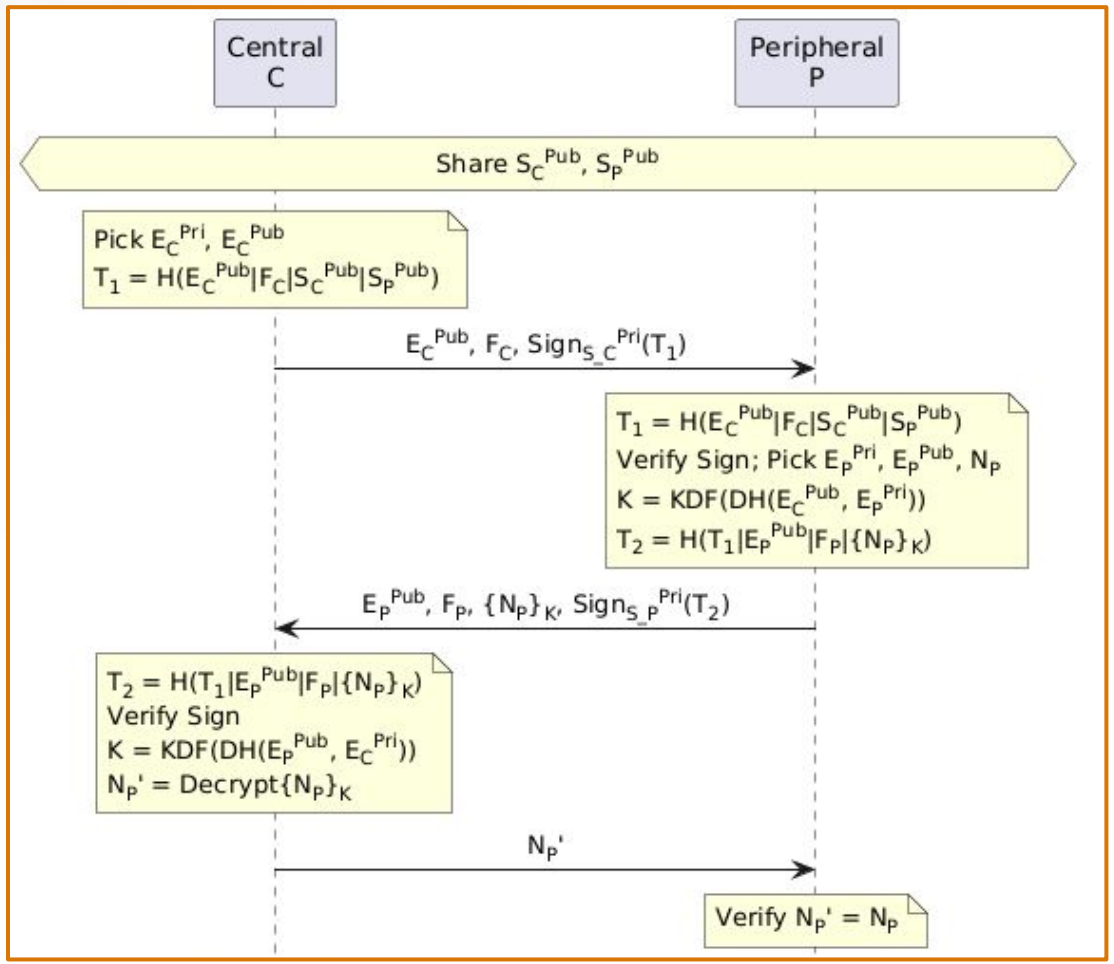

Central
C
Peripheral
P
Pick $E_C^{Pri}$, $E_C^{Pub}$
$E_C^{Pub}$, $F_C$
Pick $E_P^{Pri}$, $E_P^{Pub}$
$K_1 = DH(E_C^{Pub}, E_P^{Pri})$
$T_1 = H(E_C^{Pub}|F_C|E_P^{Pub}|F_P)$
$E_P^{Pub}$, $F_P$, $MAC_{K1}(T_1)$
$K_1 = DH(E_P^{Pub}, E_C^{Pri})$
$T_1 = H(E_C^{Pub}|F_C|E_P^{Pub}|F_P)$
Verify MAC
$K_2$ = Association($F_C \cap F_P$, $K_1$, $T_1$)
$T_2 = H(T_1|\{S_P^{Pub}\}_{K2})$
$\{S_P^{Pub}\}_{K2}$, $MAC_{K2}(T_2)$
$T_2 = H(T_1|\{S_P^{Pub}\}_{K2})$
Verify MAC
$T_3 = H(T_2|\{S_C^{Pub}\}_{K2})$
$\{S_C^{Pub}\}_{K2}$, $MAC_{K2}(T_3)$
$K = KDF_{K2}(T_3)$
$T_3 = H(T_2|\{S_C^{Pub}\}_{K2})$
Verify MAC
$K = KDF_{K2}(T_3)$

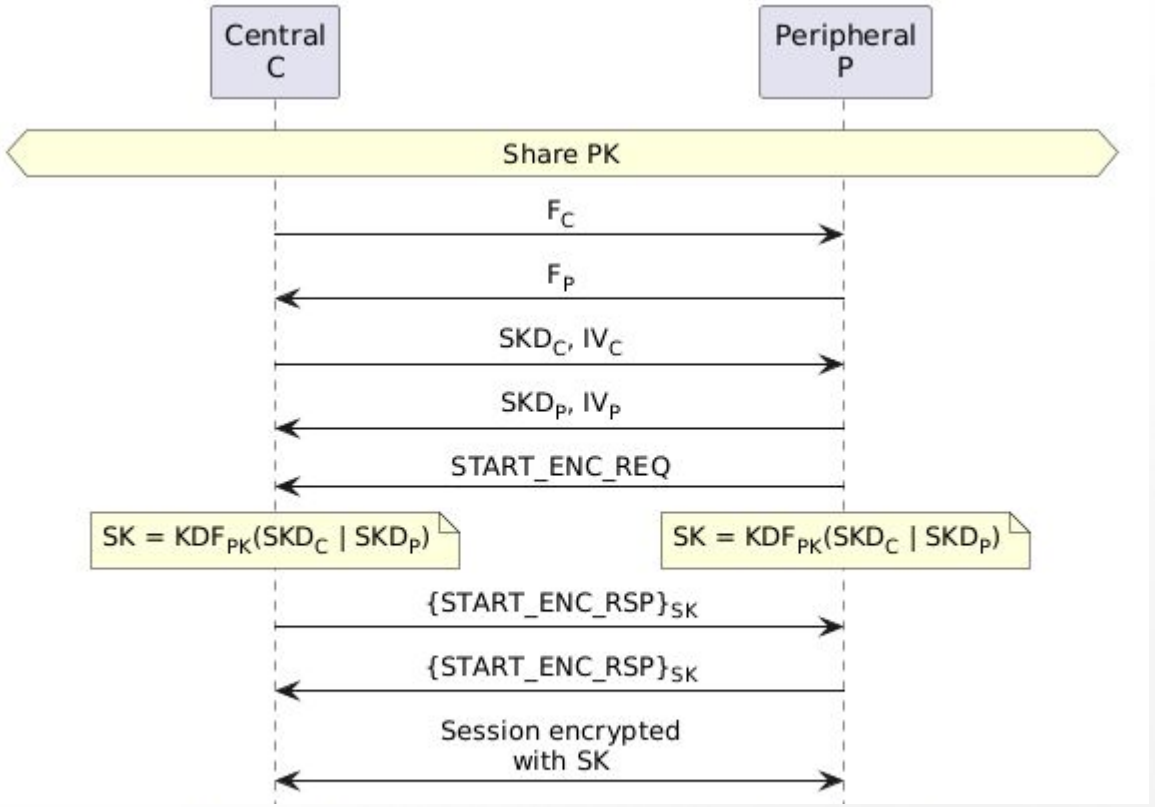

Central
C
Peripheral
P
$F_C$
$F_P$
Pick $K_C^{Pri}$, $K_C^{Pub}$
Pick $K_P^{Pri}$, $K_P^{Pub}$
$K_C^{Pub}$
$K_P^{Pub}$
$SS = DH(K_C^{Pri}, K_P^{Pub})$
$SS = DH(K_P^{Pri}, K_C^{Pub})$
$N_C$, $N_P$ = Association($F_C \cap F_P$, $K_P^{Pub}$, $K_C^{Pub}$, P, C)
$PK = KDF_{SS}(SS, N_C, N_P, C, P)$
$PK = KDF_{SS}(SS,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

# BB-Session

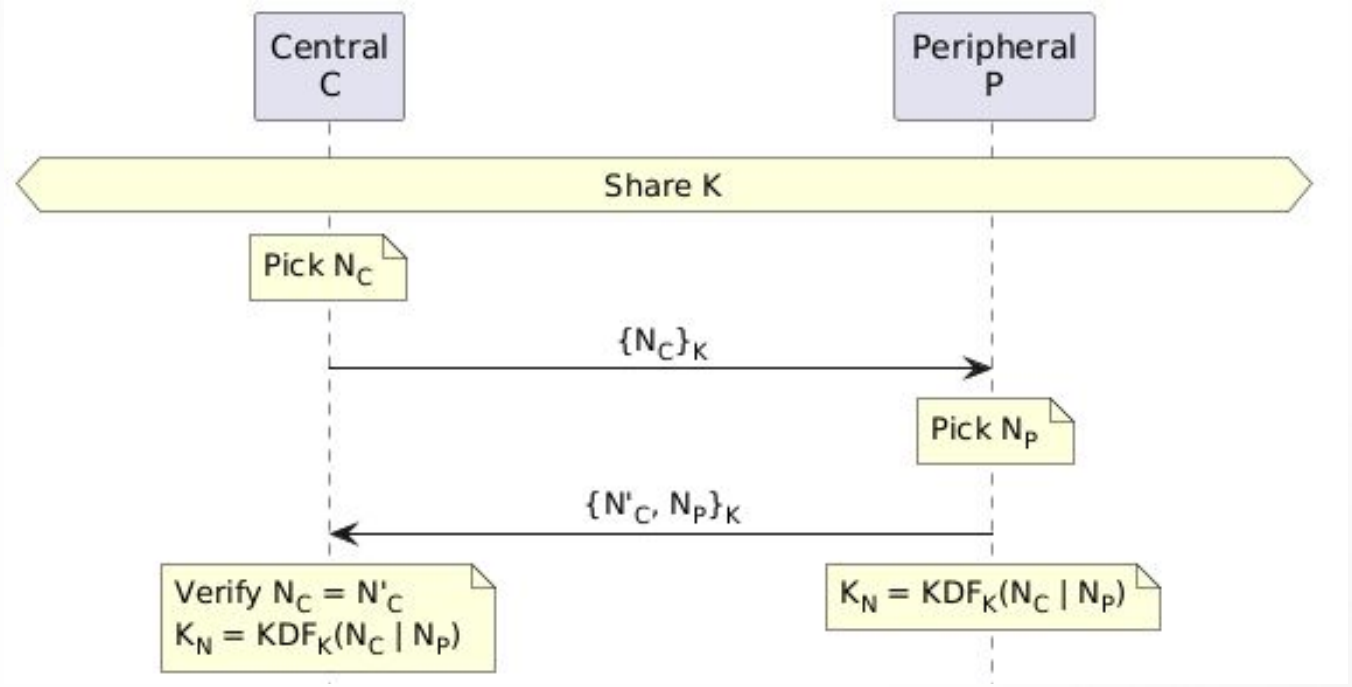

Central
C
Peripheral
P
Share $S_C^{Pub}$, $S_P^{Pub}$
Pick $E_C^{Pri}$, $E_C^{Pub}$
$T_1 = H(E_C^{Pub}|F_C|S_C^{Pub}|S_P^{Pub})$
$E_C^{Pub}$, $F_C$, $Sign_{S\_C}^{Pri}(T_1)$
$T_1 = H(E_C^{Pub}|F_C|S_C^{Pub}|S_P^{Pub})$
Verify Sign; Pick $E_P^{Pri}$, $E_P^{Pub}$, $N_P$
$K = KDF(DH(E_C^{Pub}, E_P^{Pri}))$
$T_2 = H(T_1|E_P^{Pub}|F_P|\{N_P\}_K)$
$E_P^{Pub}$, $F_P$, $\{N_P\}_K$, $Sign_{S\_P}^{Pri}(T_2)$
$T_2 = H(T_1|E_P^{Pub}|F_P|\{N_P\}_K)$
Verify Sign
$K = KDF(DH(E_P^{Pub}, E_C^{Pri}))$
$N_P' = Decrypt\{N_P\}_K$
$N_P'$
Verify $N_P' = N_P$

# BB-Session (2)

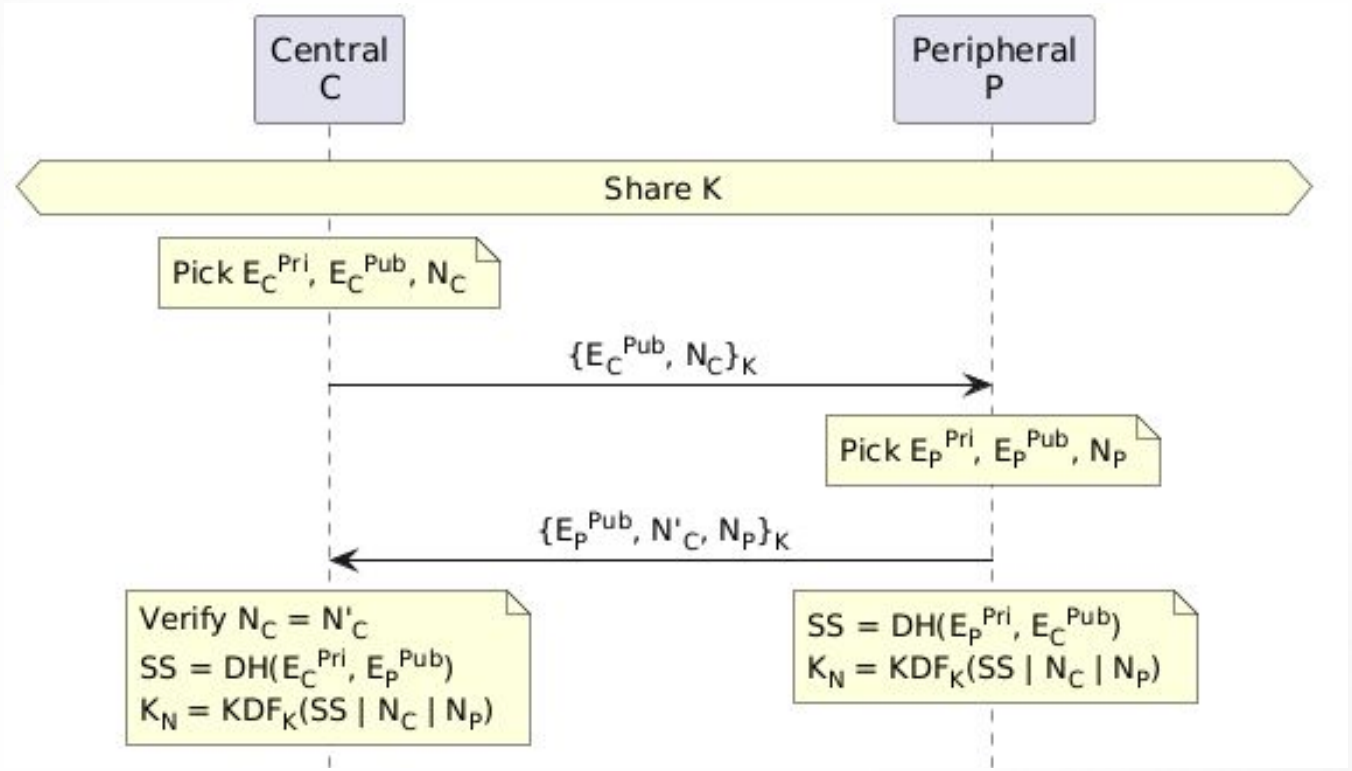

Central C
Peripheral P
Share $S_C^{Pub}$, $S_P^{Pub}$
Pick $E_C^{Pri}$, $E_C^{Pub}$
$T_1 = H(E_C^{Pub}|F_C|S_C^{Pub}|S_P^{Pub})$
$E_C^{Pub}$, $F_C$, $Sign_{S\_C}^{Pri}(T_1)$
$T_1 = H(E_C^{Pub}|F_C|S_C^{Pub}|S_P^{Pub})$
Verify Sign; Pick $E_P^{Pri}$, $E_P^{Pub}$, $N_P$
$K = KDF(DH(E_C^{Pub}, E_P^{Pri}))$
$T_2 = H(T_1|E_P^{Pub}|F_P|\{N_P\}_K)$
$E_P^{Pub}$, $F_P$, $\{N_P\}_K$, $Sign_{S\_P}^{Pri}(T_2)$
$T_2 = H(T_1|E_P^{Pub}|F_P|\{N_P\}_K)$
Verify Sign
$K = KDF(DH(E_P^{Pub}, E_C^{Pri}))$
$N_P' = Decrypt\{N_P\}_K$
$N_P'$
Verify $N_P' = N_P$


Central C
Peripheral P
Share PK
$F_C$
$F_P$
$SKD_C$, $IV_C$
$SKD_P$, $IV_P$
START_ENC_REQ
$SK = KDF_{PK}(SKD_C \mid SKD_P)$
$SK = KDF_{PK}(SKD_C \mid SKD_P)$
$\{START\_ENC\_RSP\}_{SK}$
$\{START\_ENC\_RSP\}_{SK}$
Session encrypted with SK

# BB-Rekey Symmetric



# BB-Rekey Asymmetric



# Provable Security

## Formal Analysis with ProVerif

- Key confidentiality and agreement ✅
- Mutual authentication ✅
- Forward and backward secrecy (Intra/Inter-Session) ✅
- Key Confusion Impersonation (KCI) resistance ✅
- Method confusion resistance (BB-Pairing) ✅

# Implementation

**Implemented on real-world hardware and software**

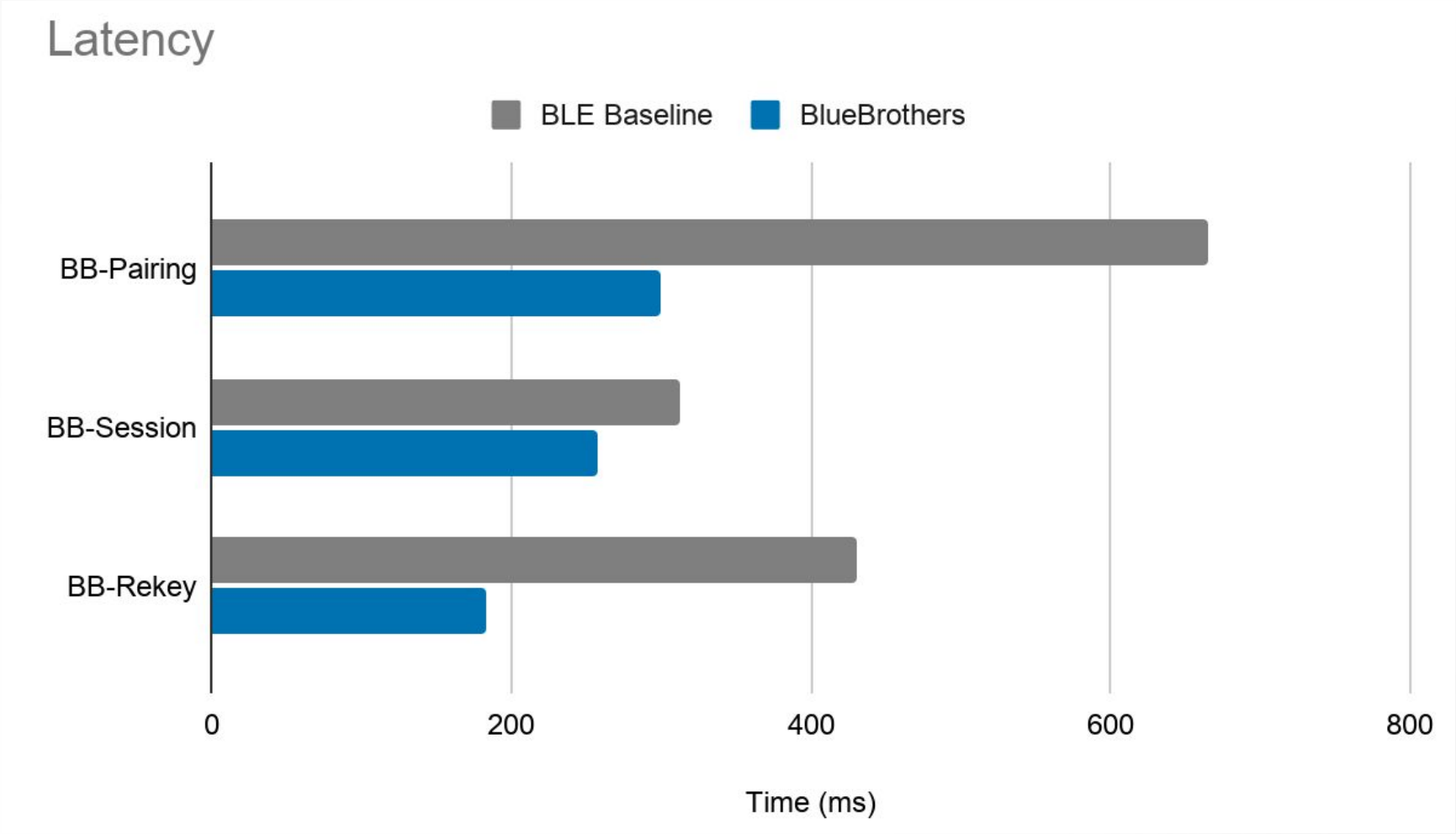
- For BLE modifying NimBLE open-source stack
  - Runs on nRF52x, nRF53x
- For BC via a C library using BlueZ APIs (Linux)
  - Runs on any Linux machine

# Evaluation

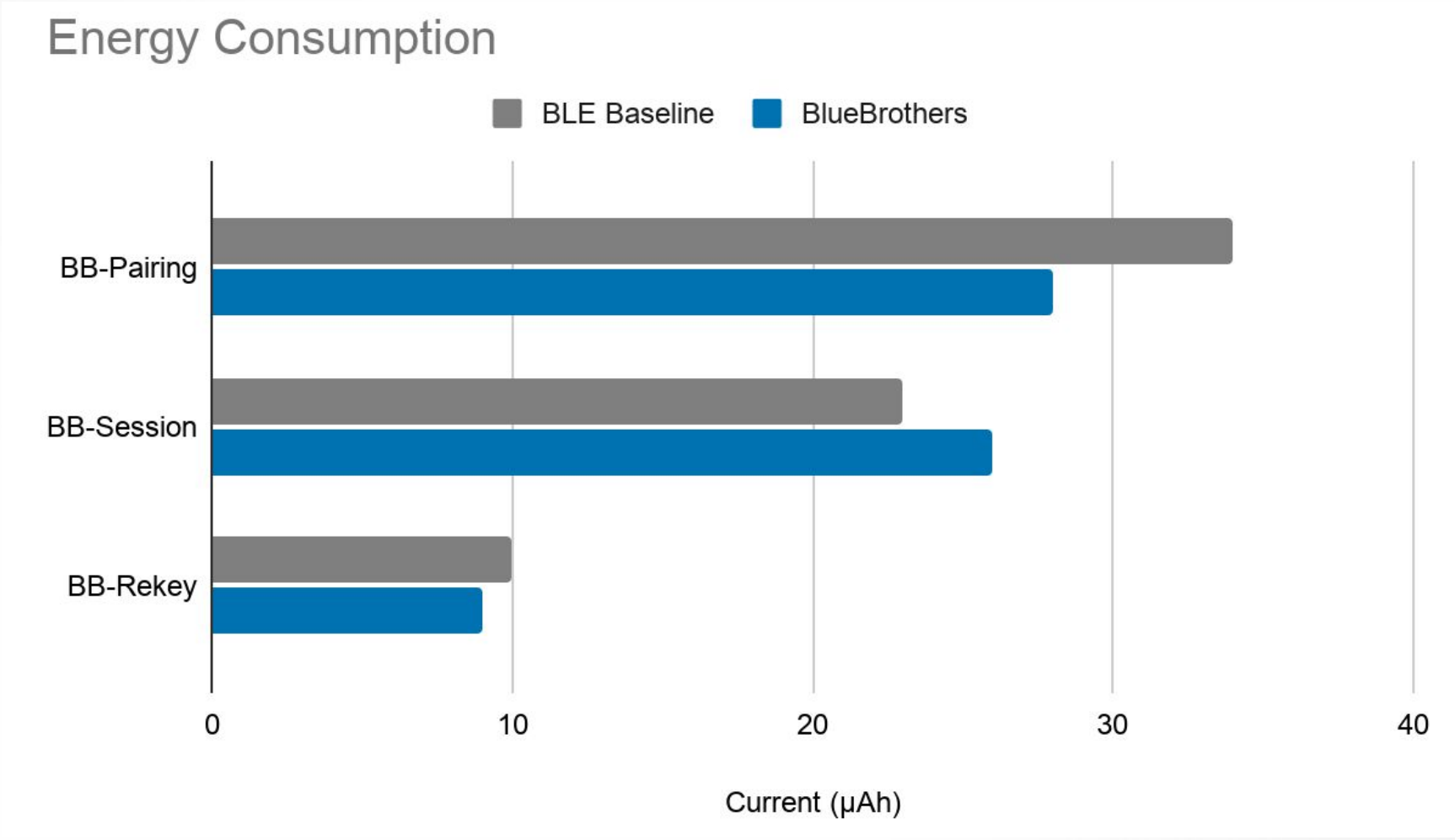
## Tested on constrained BLE devices

- Nordic nRF52 (used by mice, airtags, ...)
  - ARM Cortex-M4
  - 1 MB flash, 256 KB RAM
- Representative of low-power devices
- Latency and Energy overhead

# Evaluation Results: Latency



# Evaluation Results: Energy

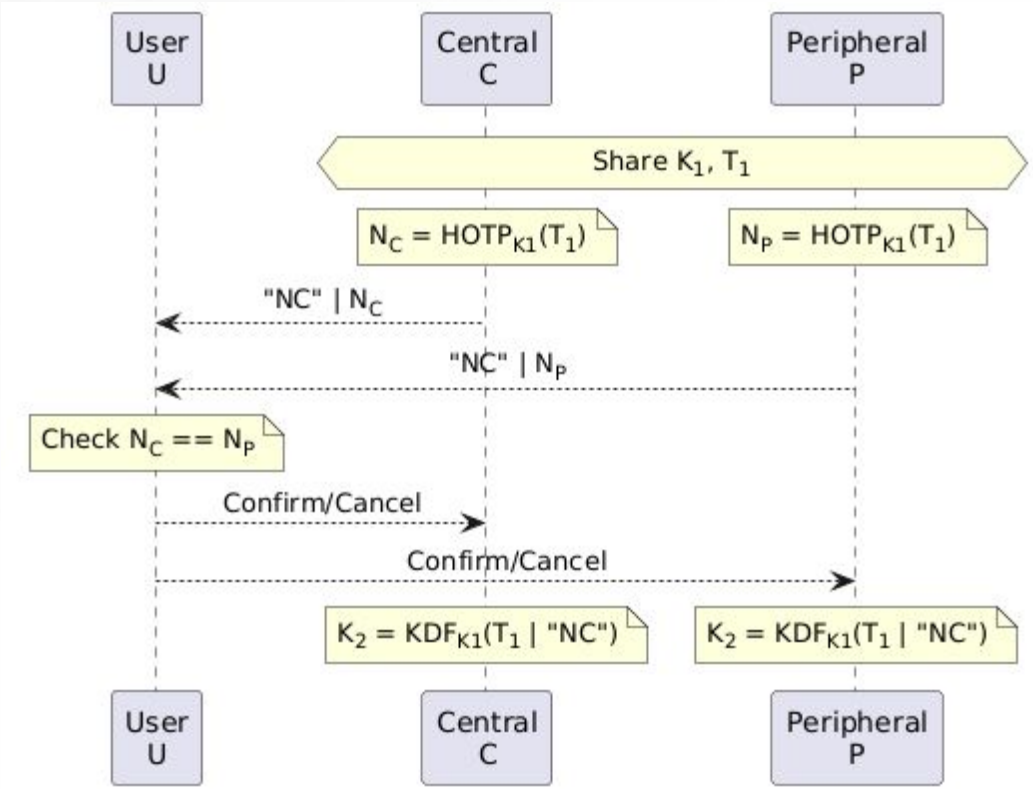


# Conclusion

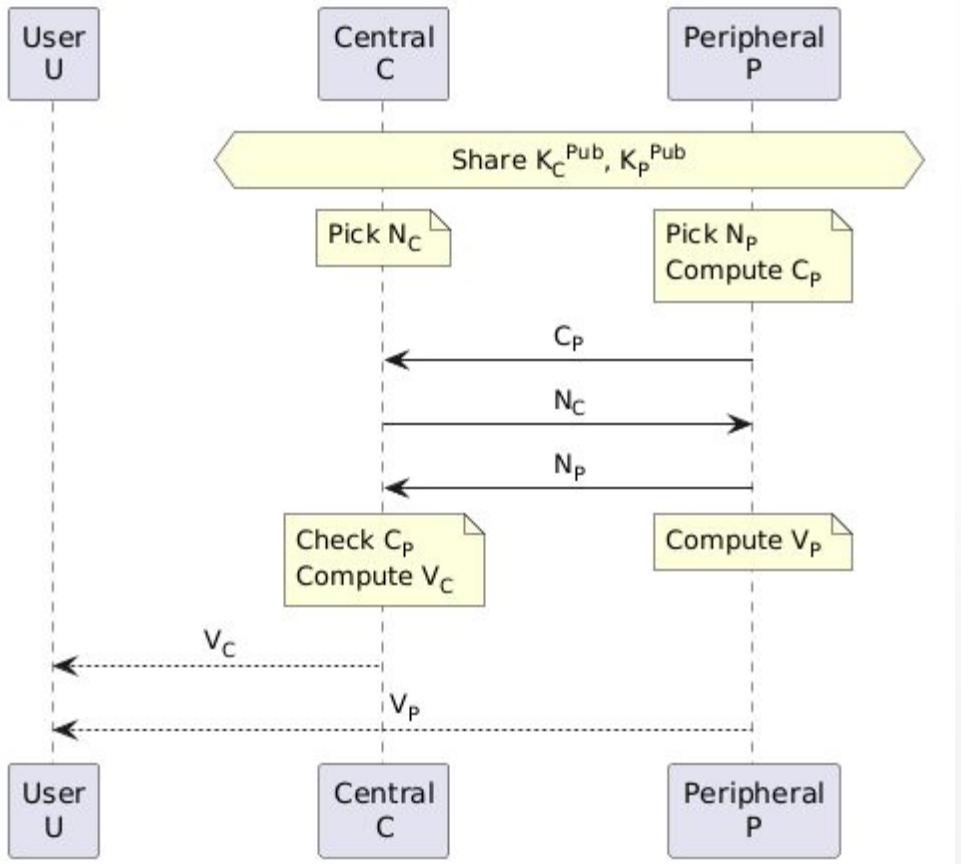
**BlueBrothers** can replace flawed Bluetooth protocols

- Address known issues (C1 - C4)
- Add new security properties (e.g., forward secrecy)
- Equal or better performances
- Simple specification
  - https://github.com/sacca97/bb-protocols
- Complementary with HardaBLE

# BB-Association: Numeric Comparison

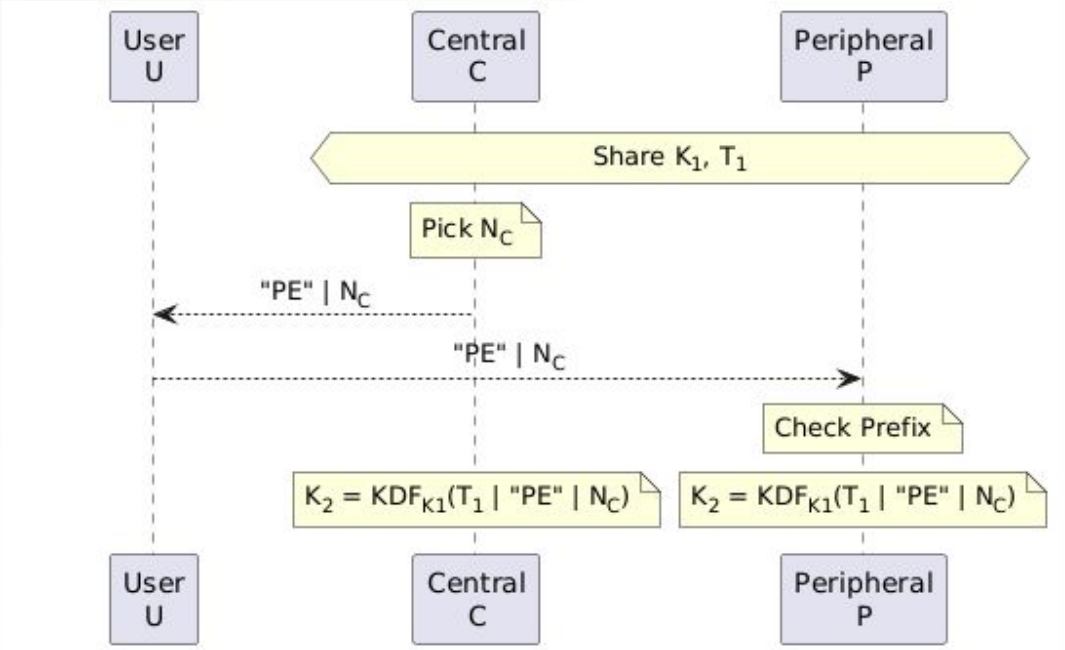



BB-NC

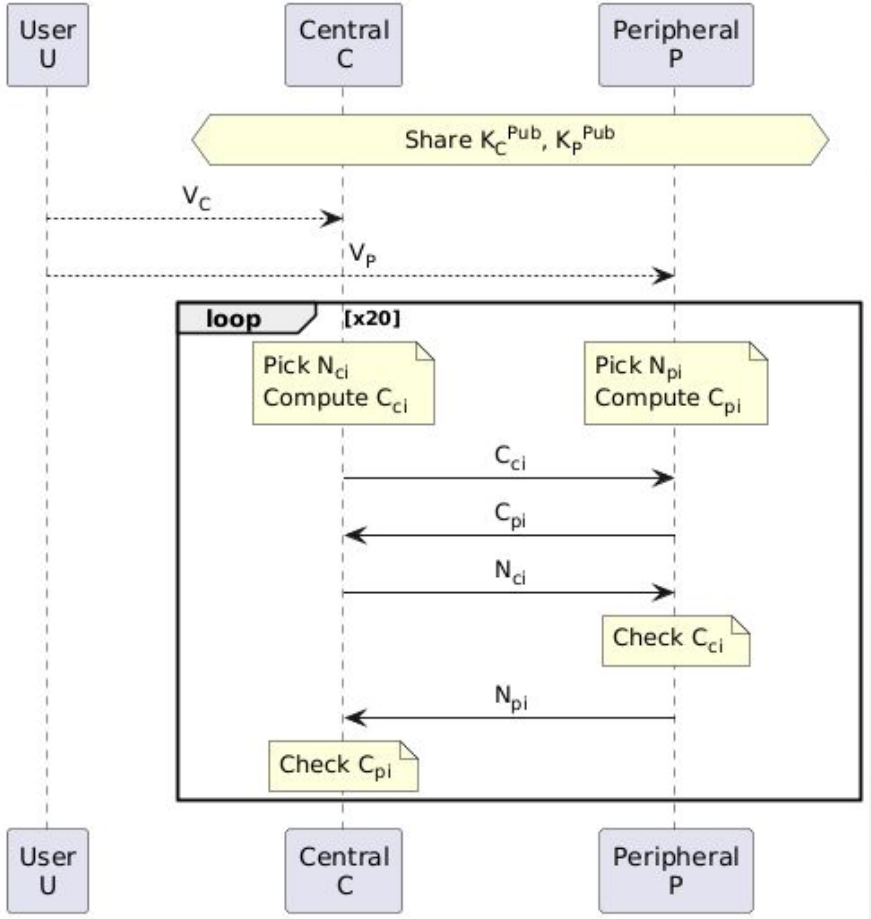



Standard NC

# BB-Association: Passkey Entry




BB-PE




Standard PE