Bluetooth (Low Energy) Security

Security of Wireless Networks – Fall 2022

Kari Kostiainen, Srdjan Capkun

ETH Zurich

Lecture outline

- **Part 1:** Bluetooth Low Energy (BLE) primer
  - Technology overview, physical layer, communication concepts
- **Part 2:** BLE Security and Privacy
  - Pairing attacks, data spoofing, user tracking
- **Part 3:** Example Application
  - Covid contact tracing using BLE beacons

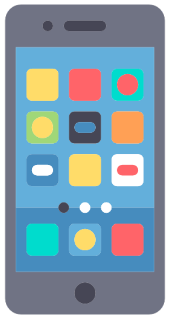
Part 1: BLE Primer

Technology overview, physical layer, communication concepts

Bluetooth Low Energy

- Two technologies
  - Bluetooth Classic (BTC) – example: music streaming
  - Bluetooth Low Energy (BLE) – example: fitness tracker, smart home sensor

- Communication: short occasional messages
- Range: long



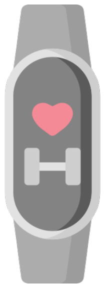
Central
Master

Client

smartphone

Peripheral
Slave

Server

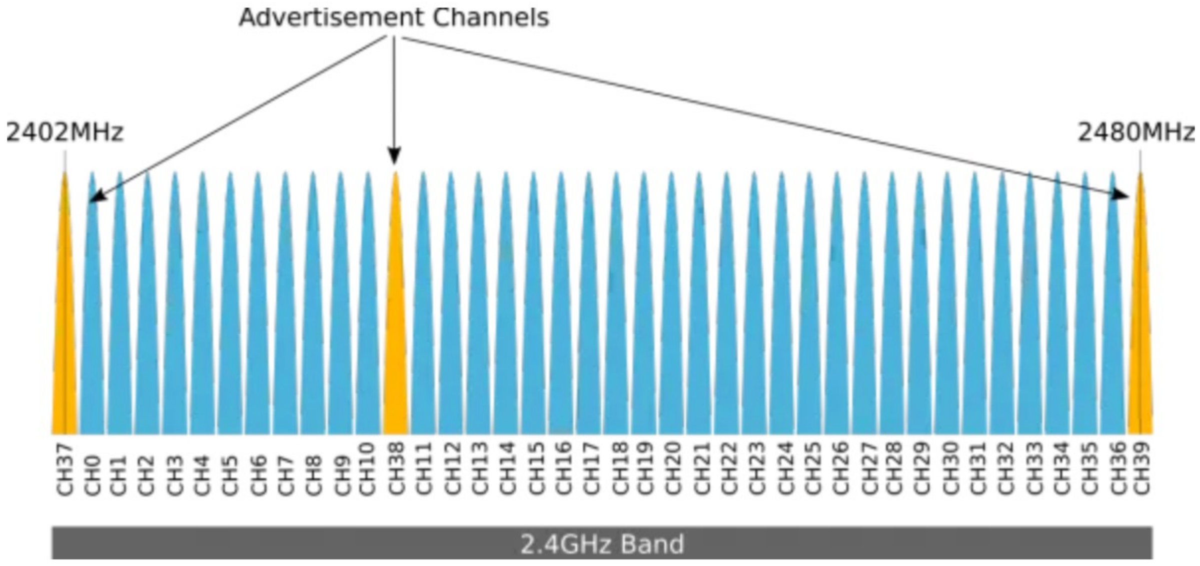


fitness tracker

Physical layer

- Spectrum: operates in 2.4 GHz band, spanning 80 MHz
- Modulation: Gaussian Frequency Shift Keying (GFSK)
- Bit rates: 125 kbps to 2 Mbps

- 40 channels with 2 MHz spacing
    - Advertising channels (37, 38, 39)
    - Data channels (0 … 36)



BLE Communication Band

Figure from: https://embeddedcentric.com/

Frequency hopping

- Schedule negotiated during connection establishment
- Devices use new channel for every packet
- Parameters
  - “hop increment” – defines next channel
  - “hop interval” – defines next sending time

Question: *What are the benefits?*

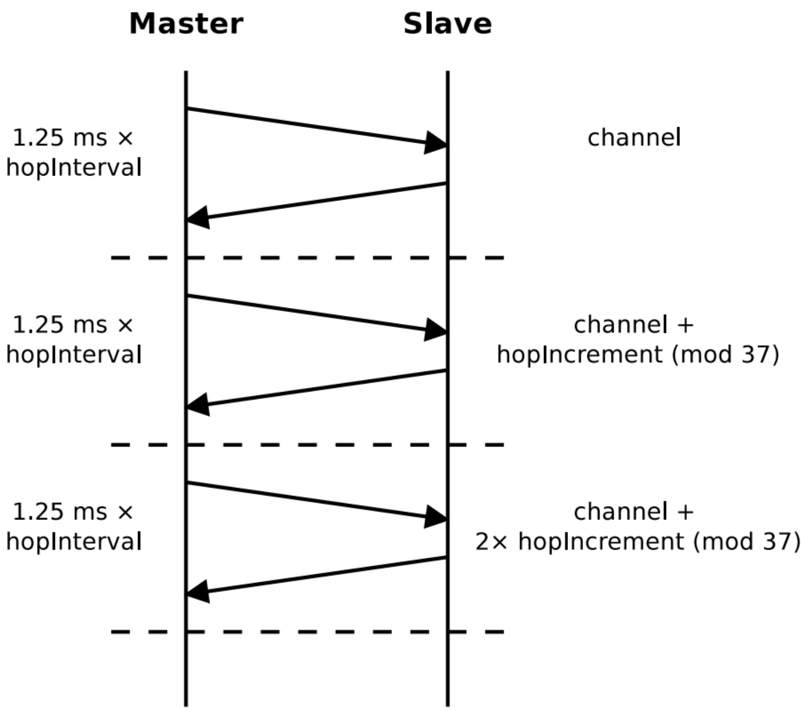
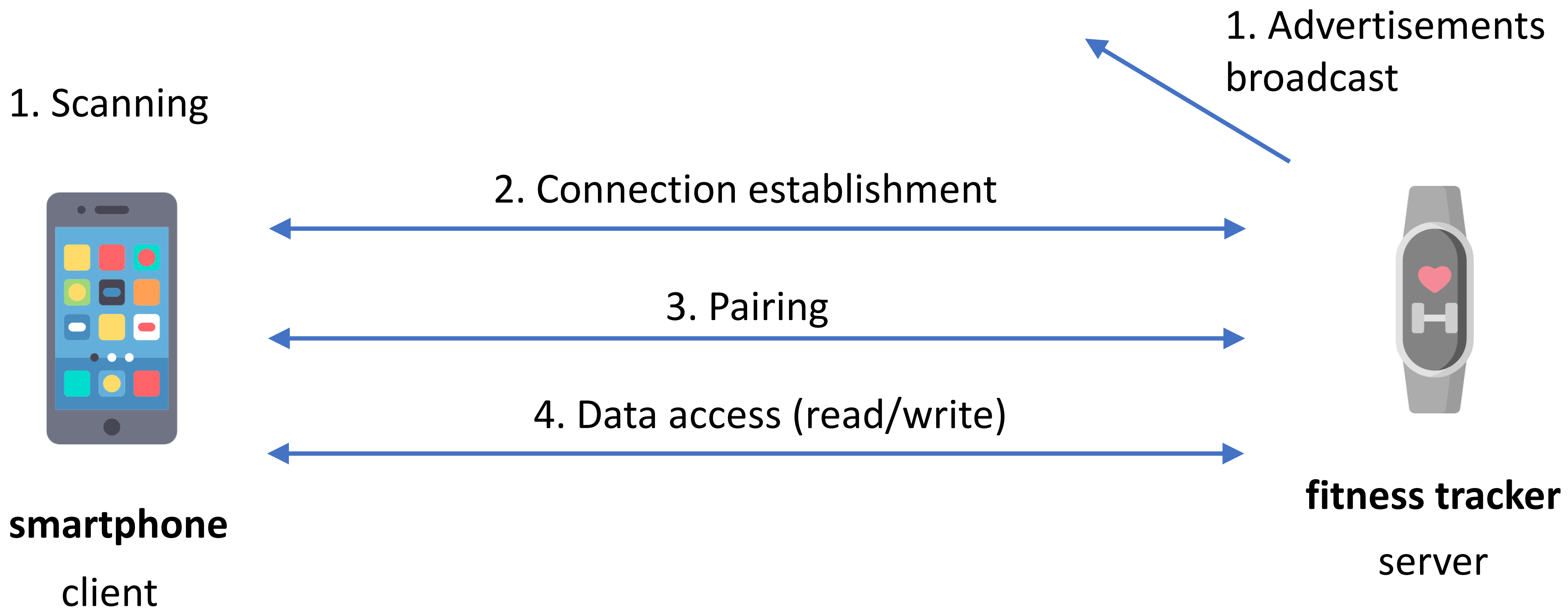


Figure from: Ryan, WOOT’13

Communication overview



1. Advertisements

- Advertising sesssion: server sends beacons to all 3 advertisement channels
- Advertising interval: 20ms to 10s

- Advertisement message:
  - Message type
  - Randomized MAC address
  - UUIDs of offered services

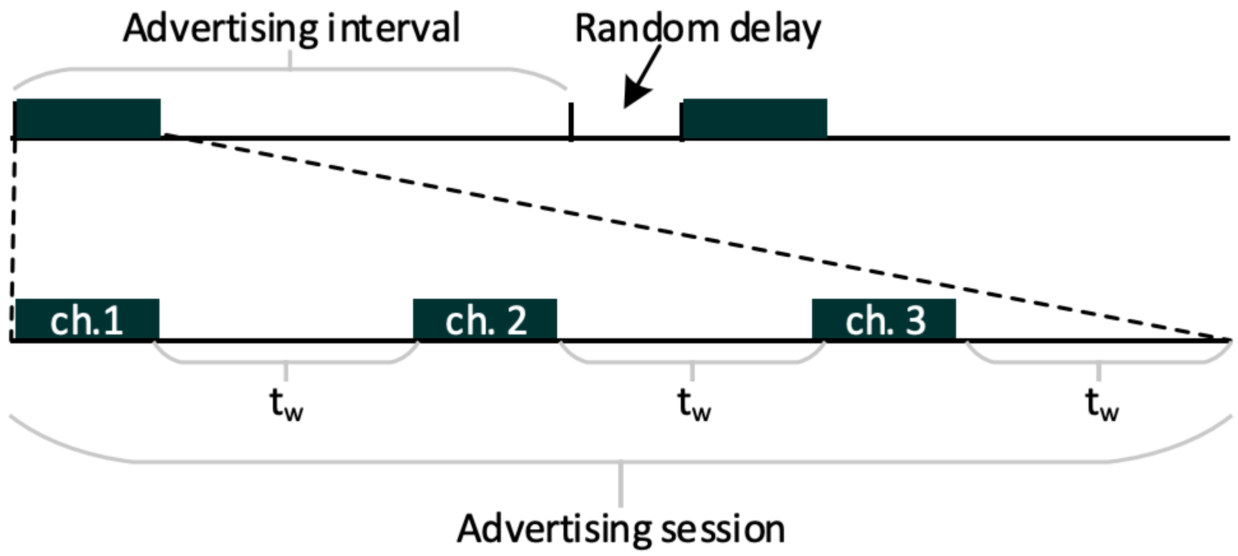
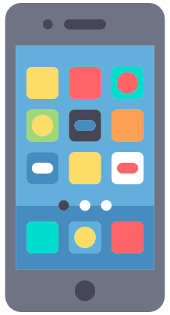


Figure from: Fawaz et al. Usenix Security'16

2. Connection establishment

smartphone

client



2. Connection request

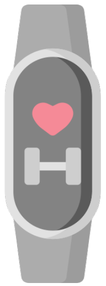
client and server address

connection parameters (timeouts, intervals…)

fitness tracker

server

1. Listen



3. Start frequency hopping

3. Start frequency hopping

3. Pairing

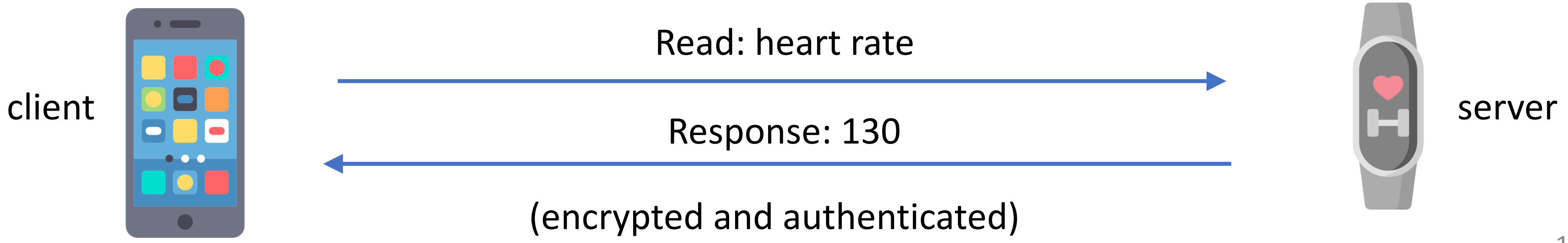
- Legacy pairing
  - Not secure (neither against passive or active adversary)

- Secure Connection pairing
  - 4 alternatives or “Association Methods” depending on I/O capabilities
  - Just Works (unauthenticated, passive adversary)
  - **Numeric Comparison** and **Passkey Entry** (authenticated, active adversary)
  - Out of Band (OOB)

- Diffie-Hellman key exchange → Long Term Key (LTK)

4. Data access

- Data on server stored in “attributes” (e.g., heart rate)
  - General Attribute Profile (GATT)
- Server maintains access control policy for each attribute
  - Access type: Read-only, write-only, read-and-write
  - Security level: no security, encryption, encryption and authentication
- Link layer: AES-CCM using session key (SK) derived from LTK



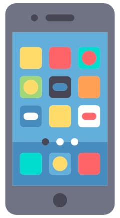
Privacy (tracking prevention)

- Fixed MAC address (in every beacon) would make tracking trivial
- **Address randomization**
  - Random static = may change during boot
  - Random non-resolvable = may change anytime
  - Random resolvable = peers can determine if known device

Identity Resolving Key

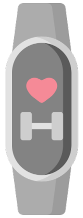
1. IRK ← LTK
2. Generate rand
3. Apply one-way function: hash ← f(IRK, rand)

Client



advertisement: address = 01 || rand || hash

Use for pre-defined period (e.g., **15 minutes**)

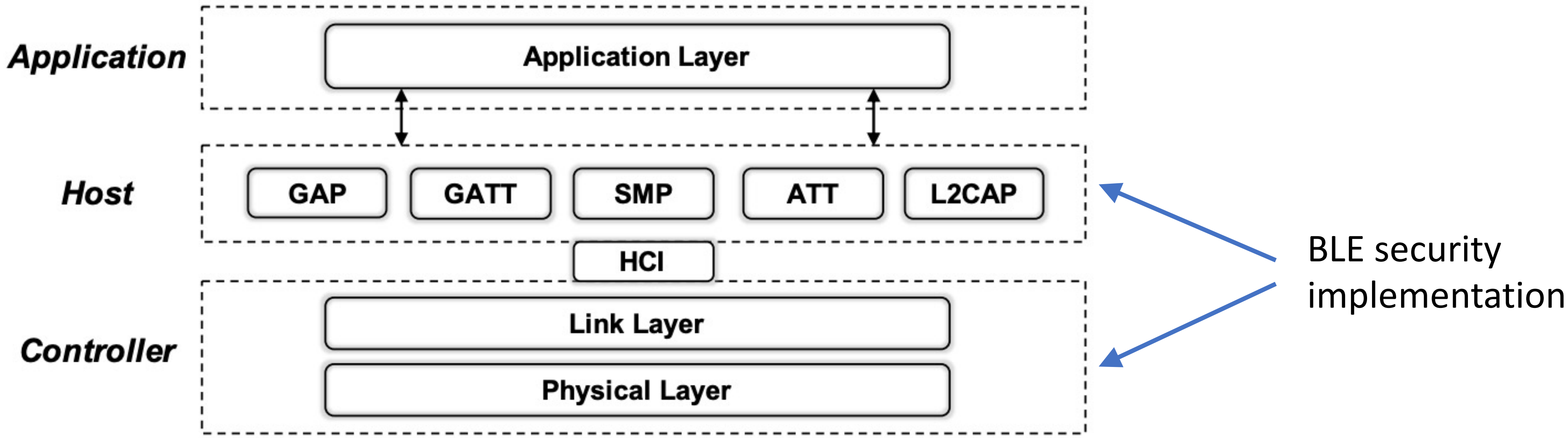


Server

1. IRK ← LTK
2. hash' ← f(IRK, rand)
3. Check if hash = hash'

BLE stack

- Bluetooth 5.2 specification more than 3000 pages
- Android Bluetooth stack 400k LoC



Question: *Which security mechanisms in hardware?*

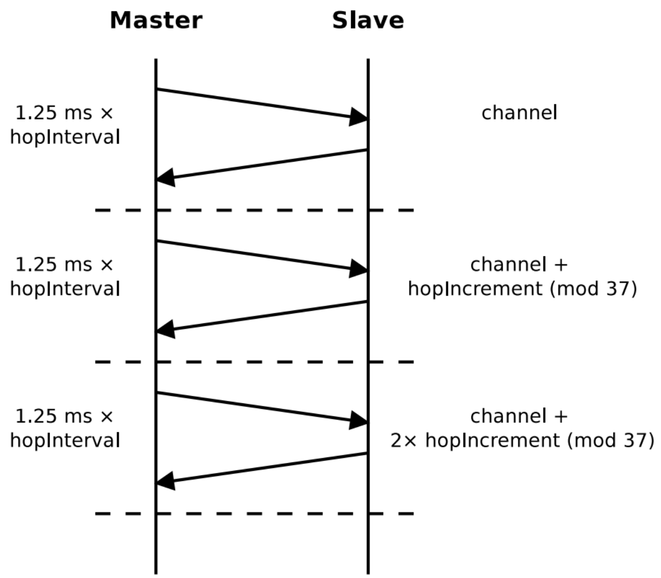
Figure from: Wen et al. CCS'20

Part 2: BLE Security and Privacy

Pairing attacks, data spoofing, user tracking

Recording communication

- **Recall:** BLE devices hop channels
- **Challenge:**
  - Assume adversary not present at initialization
  - Hopping pattern unknown



- Determining hopping sequence from on-air traffic (Ryan, WOOT’13)
  1. Measure time between two packets on the same channel
  2. Measure time between two packets on consequtive channels
  3. Solve a few modulo equations → hopping interval and increment

Legacy pairing

- Proprietary key exchange protocol by Bluetooth SIG
  - Authenticated using 6-digit PIN (PassKey) or Just Works
  - Bluetooth 4 spec (2009)

- Can be broken even by passive adversary (Ryan, WOOT'13)
  - **Secret TK derived from PIN**
  1. Try all PIN values (0 to 999,999)
  2. Check which gives correct "confirm"
  3. Then derive STK

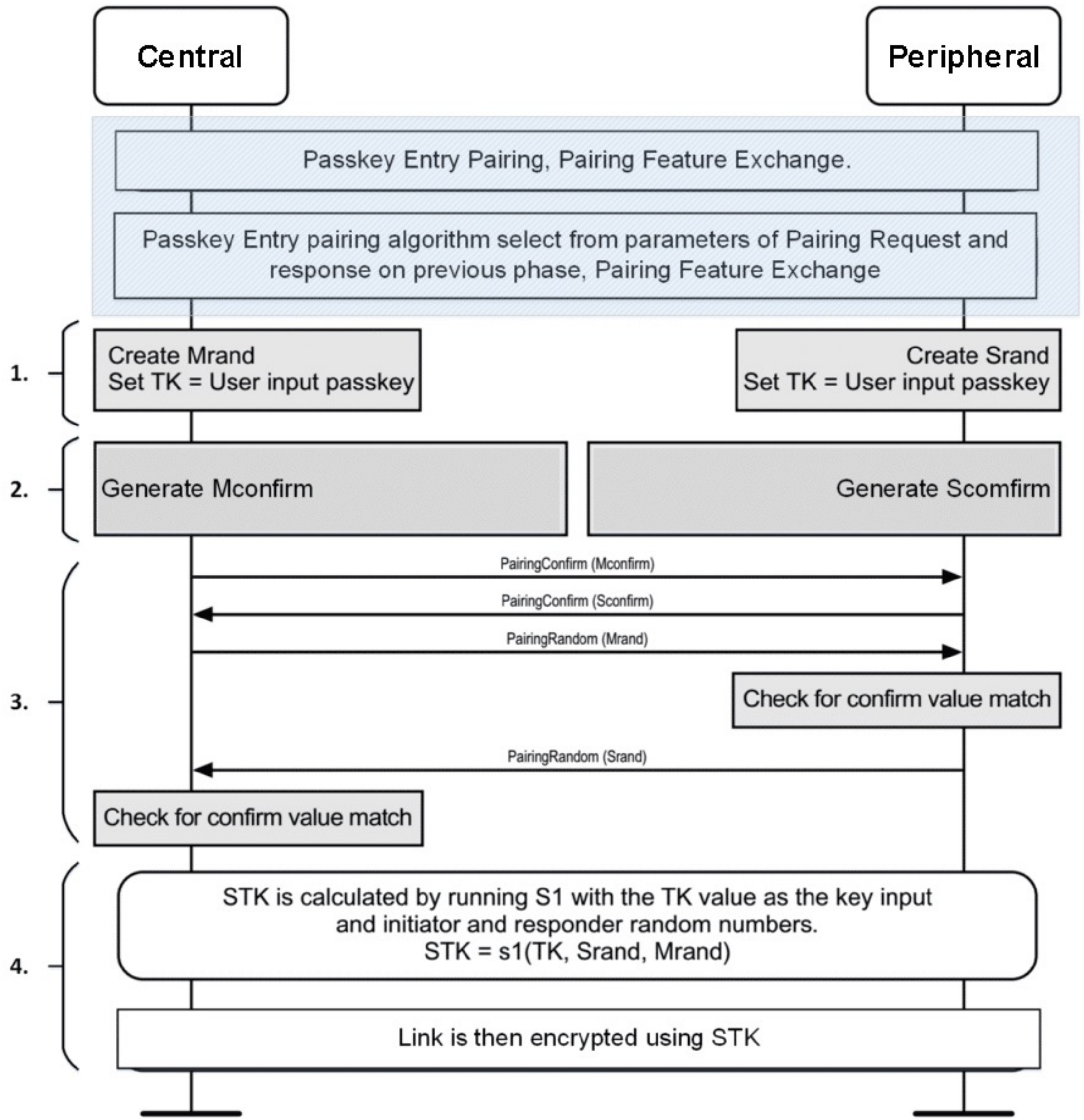


Figure from: bluetooth.com

Secure Connection pairing

- Authenticated Elliptic Curve Diffie-Hellman (ECDH) key exchange

- Association Methods
  - Just Works (passive)
  - <u>Numeric Comparison</u> (active)
  - <u>Passkey Entry</u> (active)
  - OOB (active)

- I/O capabilities
  - DisplayOnly
  - DisplayYesNo (can confirm)
  - NoInput NoOutput…

		Initiator				
		Display Only	Display YesNo	Keyboard Only	NoInput NoOutput	Keyboard Display
Responder	Display Only	Just Works	Just Works	Passkey Entry 🔴	Just Works	Passkey Entry 🔴
	Display YesNo	Just Works	Numeric Comparison	Passkey Entry 🔴	Just Works	Numeric Comparison
	Keyboard Only	Passkey Entry 🔵	Passkey Entry 🔵	Passkey Entry 🟡	Just Works	Passkey Entry 🔵
	NoInput NoOutput	Just Works	Just Works	Just Works	Just Works	Just Works
	Keyboard Display	Passkey Entry 🔵	Numeric Comparison	Passkey Entry 🔴	Just Works	Numeric Comparison

🔴 Responder displays, Initiator inputs
🔵 Initiator displays, Responder inputs
🟡 Initiator inputs and Responder inputs

Figure from: Tschirschnitz, S&P’21

Phases 1 and 2: Feature and key exchange

- Protocol phases
  1. Feature exchange
  2. Key exchange (DH)
  3. Authentication
  4. Validation

Phase 1: Feature exchange

Initiator IO caps (DisplayYesNo), …

Responder IO caps (Keyboard only), …

Phase 2: Key Exchange

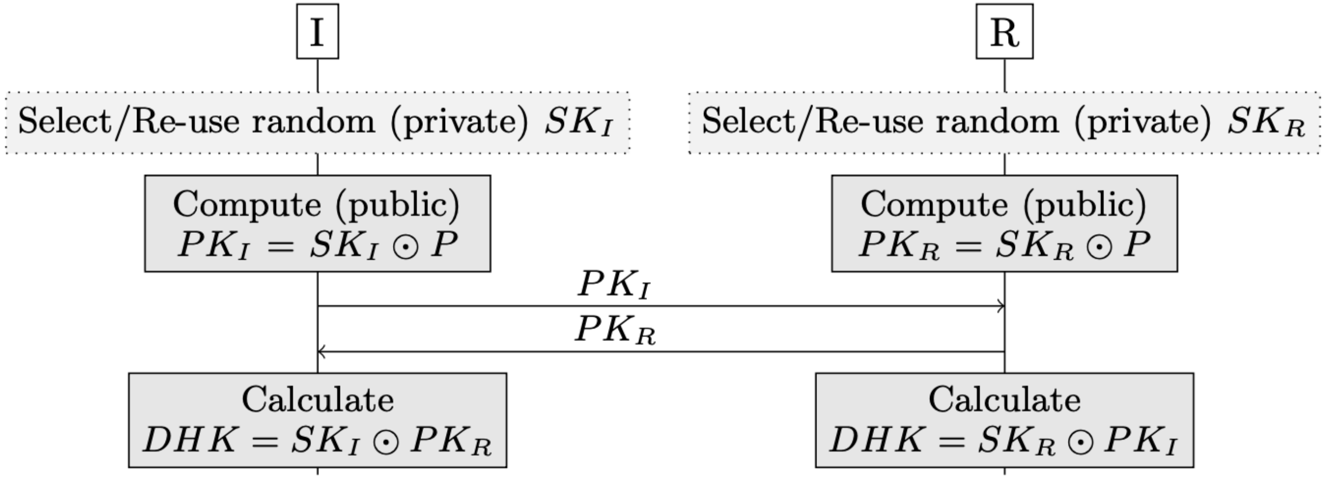


Figure from: Tschirschnitz, S&P’21

Phase 3: Authentication (Numeric comparison)

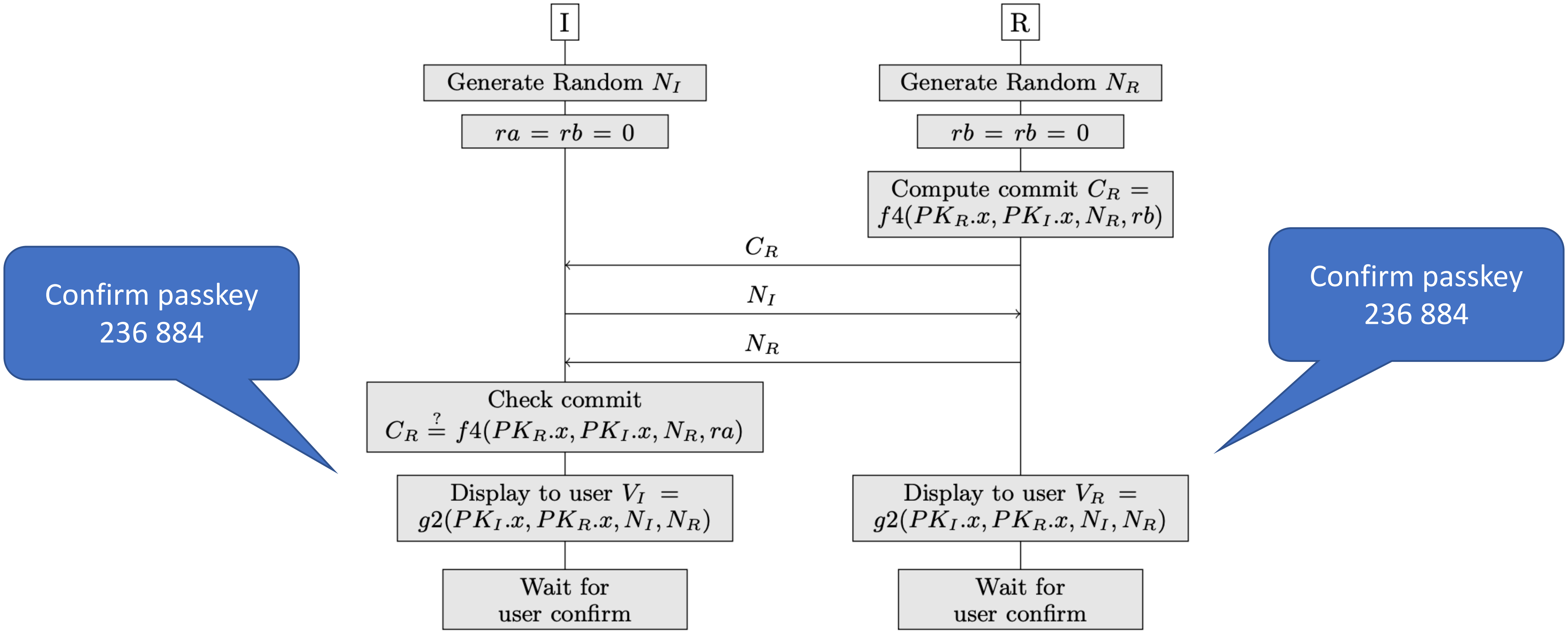
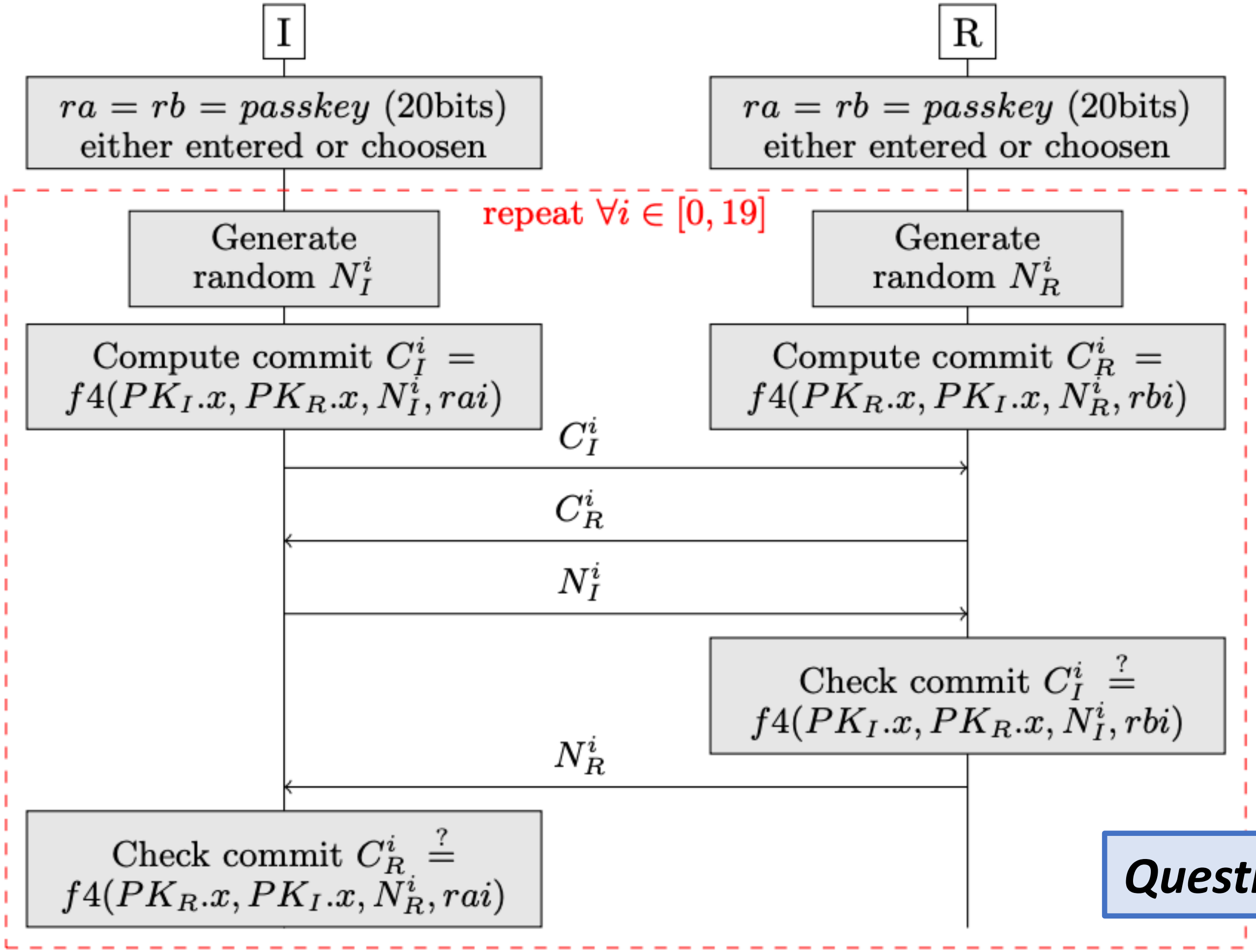


Figure from: Tschirschnitz, S&P'21

Phase 3: Authentication (Passkey entry)

Passkey:
236 884

Initiator device chooses passkey



Enter passkey:
****

User enters passkey to Responder device

Question: *Why repeat design?*

Figure from: Tschirschnitz, S&P'21

Phase 4: Pairing validation

- Check that everything done in previous phases went corrently
  - No man-in-the-middle manipulation
- If success
  - Derive LTK from agreed DHK

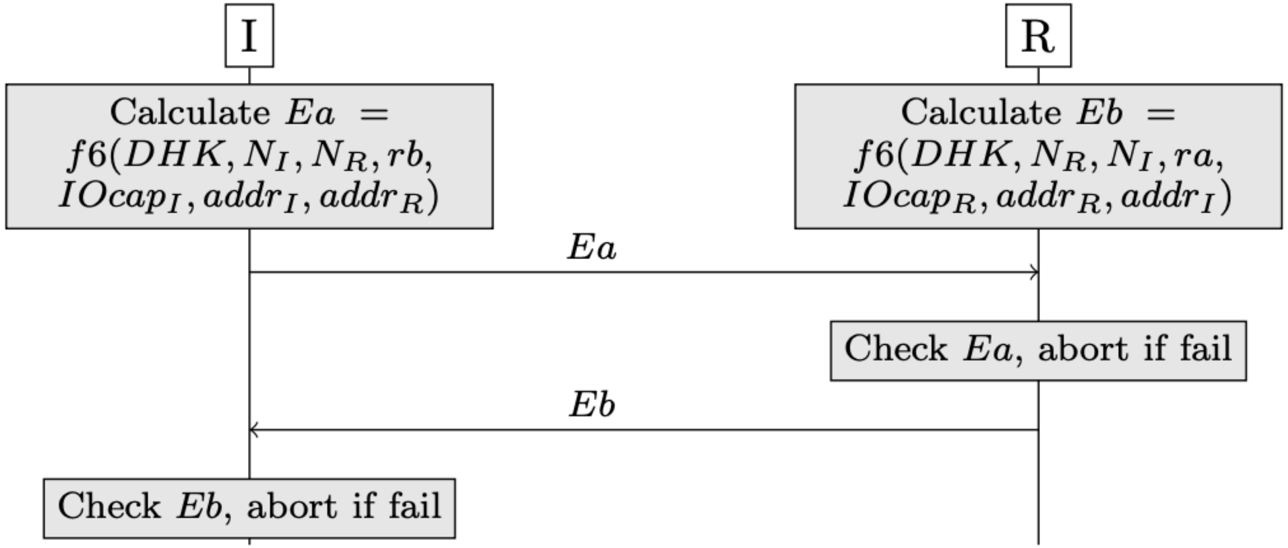


Figure from: Tschirschnitz, S&P'21

Method Confusion Attack

- Recent attack discovery (Tschirschnitz, S&P'21)

- Main idea:
  - Adversary plays man-in-the-middle
  - Use one Association method (e.g., Passkey entry) with Initiator
  - Use another Association method (e.g., Numeric comparison) with Responder
  - **Interleave both protocol runs**

Method Confusion Attack

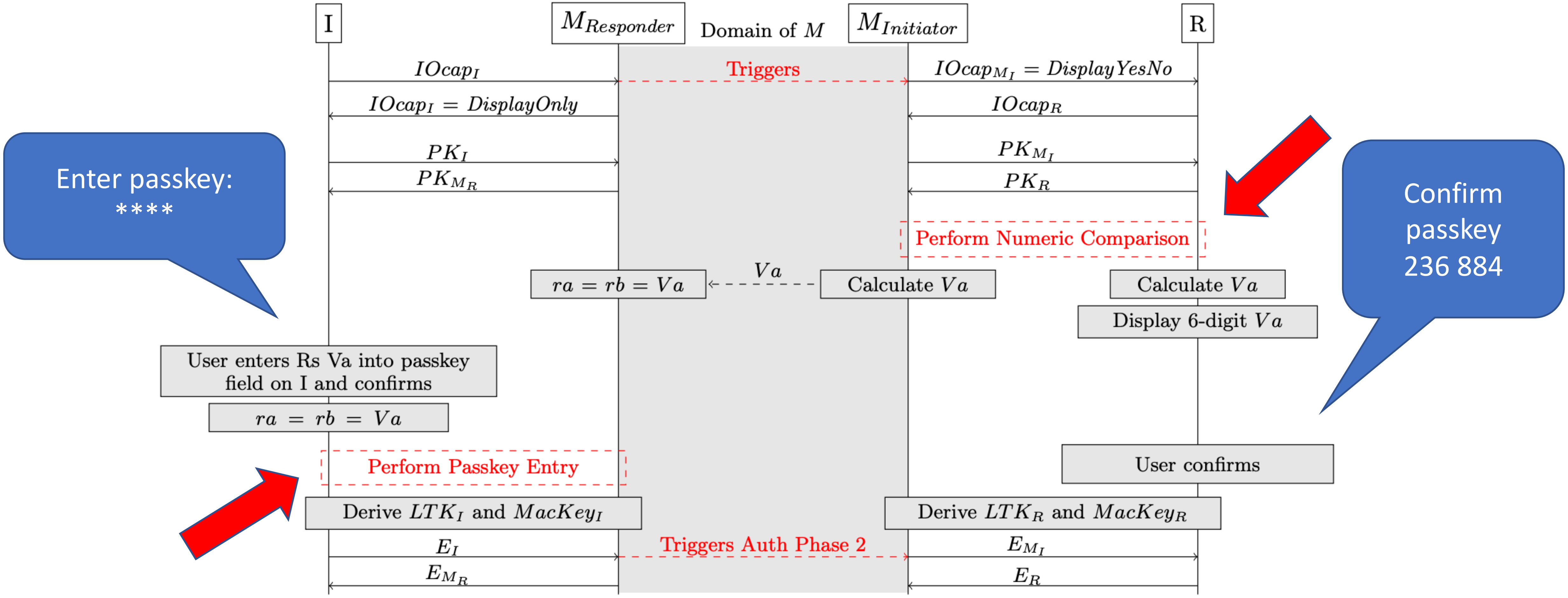
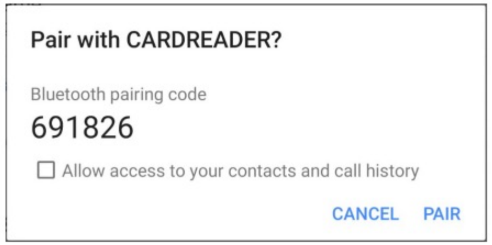


Figure from: Tschirschnitz, S&P'21

Attack discussion

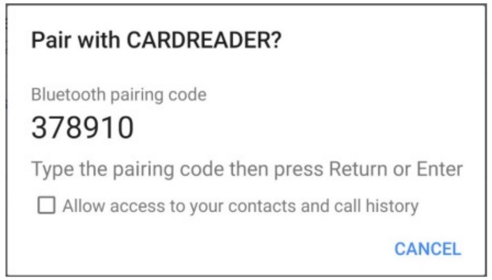
- Why does such attack work?
    - Different Association Models use similar “check value”
    - Specification is vague regarding wording
- Would users notice?
    - In user study, 92% fell for the attack
- **Realization:** MitM, selective jamming, low-latency implementation…
- **Fix:** make user-copied values “incompatible”

A



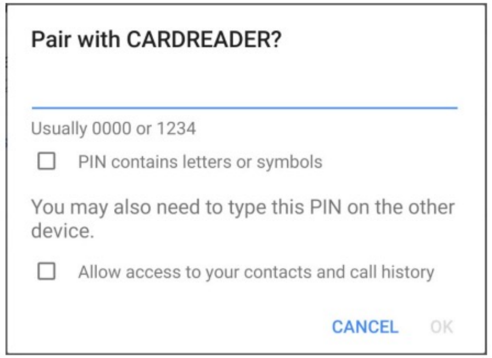
(a) Android 10.0 - Numeric Comparison.

B



(b) Android 10.0 - Passkey Display.

C



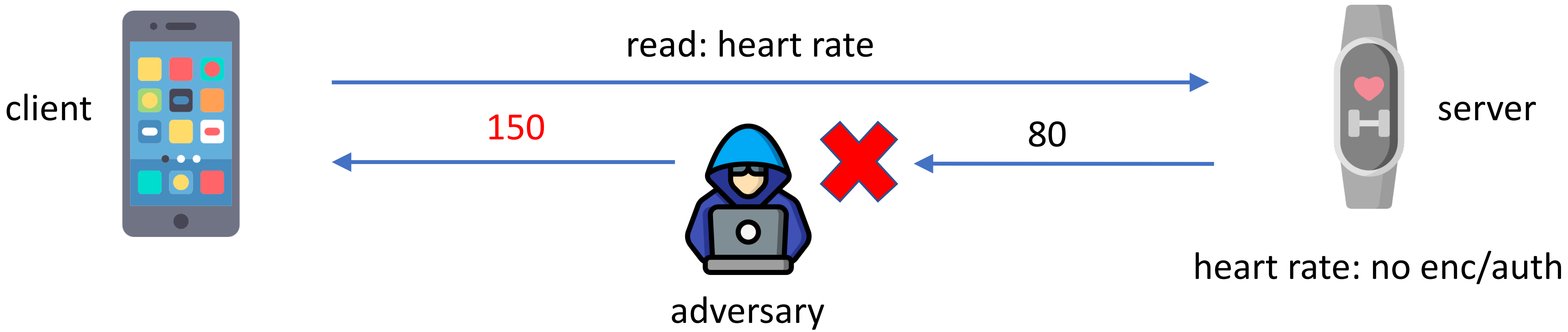
(c) Android 10.0 - Passkey Enter.

Figure from: Tschirschnitz, S&P’21

Data access and spoofing

- **Recall:** data on server stored as “attributes” (GATT)
  - Each attribute can have separate access control policy: read/write/enc/auth …

- **Obvious:** If attribute requires no protection, “spoofing” possible



Reactive authentication

- Scenario
  - Client and server already paired
  - New connection after disconnect
  - Attribute requires enc/auth

- Reactive authentication in BLE
  - Client sends plaintext request
  - Server asks to “turn on” auth/enc

Questions: *Why not always-on encryption? What might go wrong with this design?*

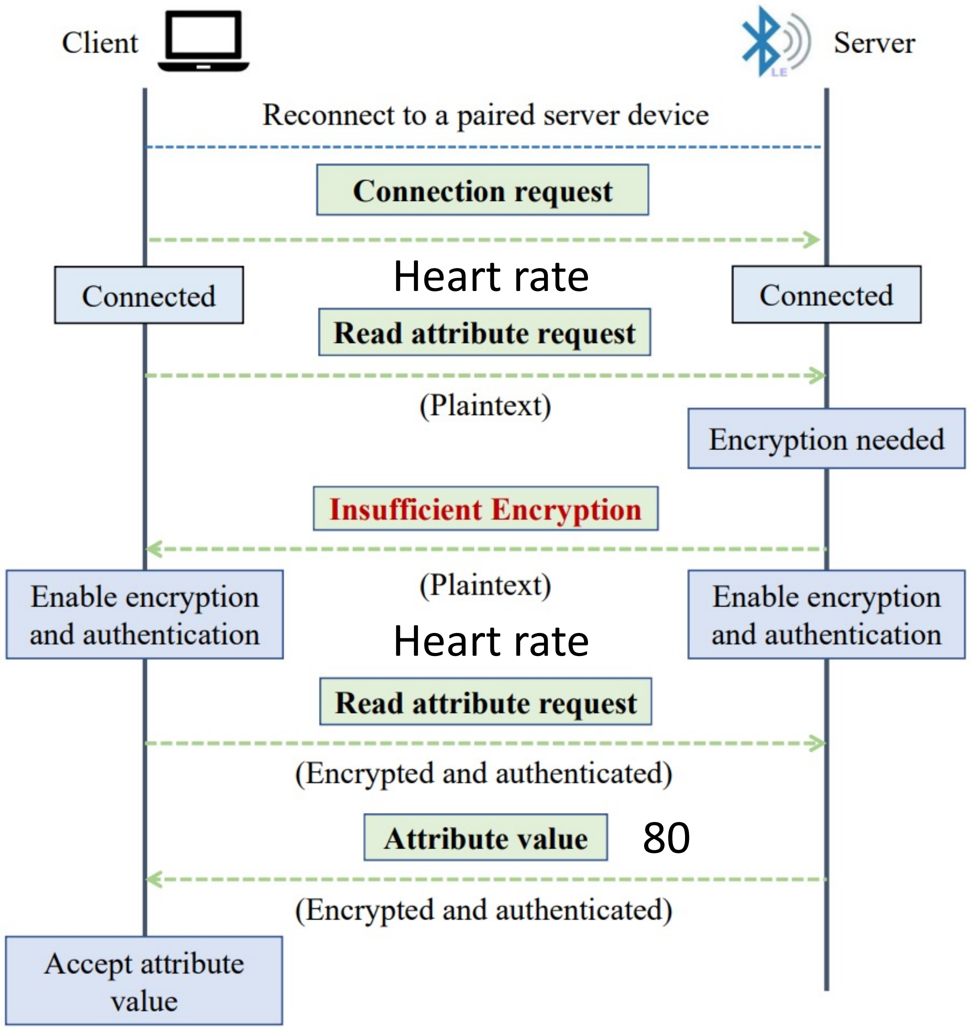


Figure from: Wu, WOOT’20

Spoofing data at reconnection

- Leverage reactive authentication for spoofing attack (Wu, WOOT'20)

- Adversary
    1. Advertise as honest server
    2. Capture connection request
    3. Provide spoofed plaintext

- Is this even an attack?
- **Fix:** use proactive authentication

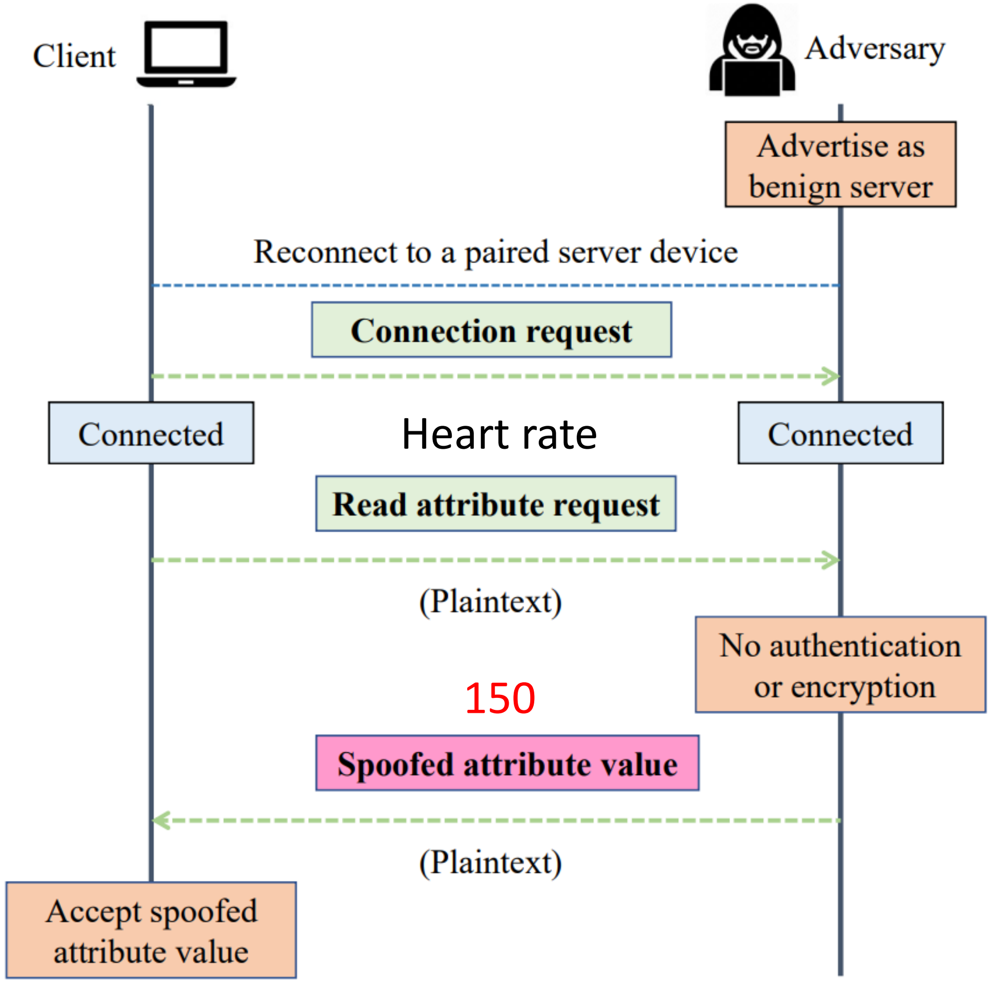


Figure from: Wu, WOOT'20

Privacy (tracking prevention)

- **Recall:** Fixed MAC address → simple tracking
- **Solution in BLE:** randomize MAC address periodically (e.g., every 15 min)
- Good practice but unfortunately not all manufacturers follow it

Name	Type	Days observed
One	activity tracker	37
Flex	activity tracker	37
Zip	activity tracker	37
Surge	activity tracker	36
Charge	activity tracker	36
Forerunner 920	smartwatch	36
Basis Peak	sleep tracker	25
MB Chronowing	smartwatch	15
dotti	pixel light	7
UP MOVE	fitness tracker	2
GKChain	laptop security	2
Gear S2 (0412)	smartwatch	2
Crazyflie	quadropter	1
Dropcam	camera	1

Table from: Fawaz, Usenix Security'16

Proprietary advertisements

- Many devices implement proprietary BLE advertisements

- Apple products support feature called “Continuity”
  - Universal clipboard: copy-paste across devices
  - Handoff: start email on one device, continue in another
  - Enabled by transmission of special BLE advertisements

- Windows 10 devices advertise “manufacturer specific data”
  - Also realized as BLE advertisements

Tracking anonymous devices

- **Observation 1:** parallel advertisements may enable long-term tracking if randomization not carefully synchronized (Becker, PETS’19)

Windows 10 device

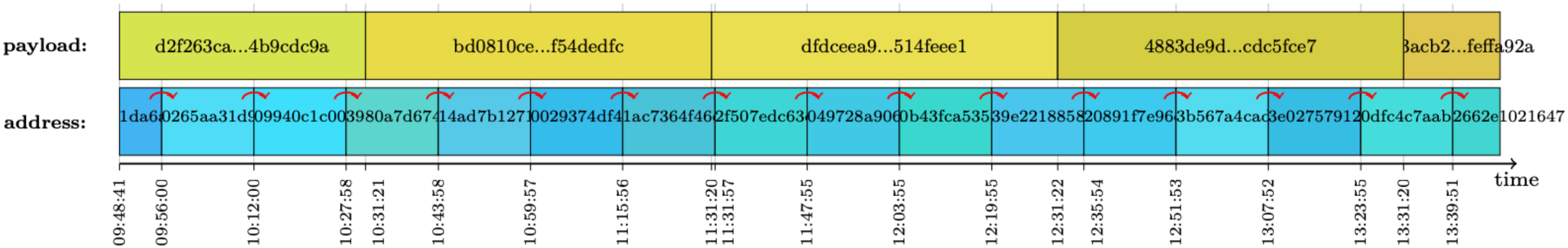
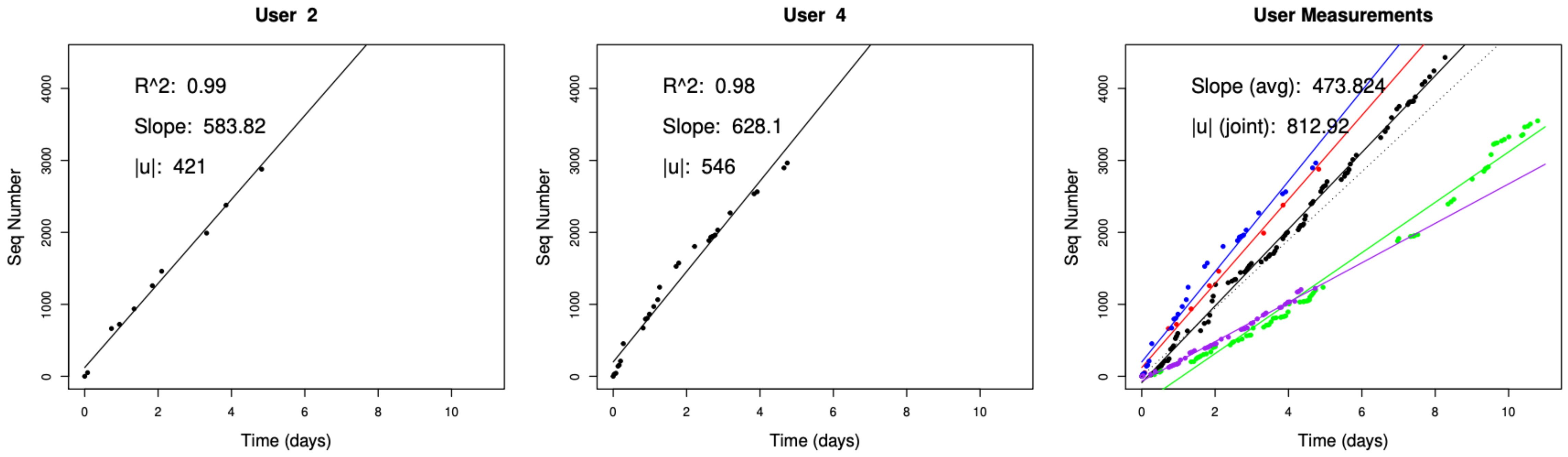


Figure from: Becker, PETS’19

Tracking anonymous devices

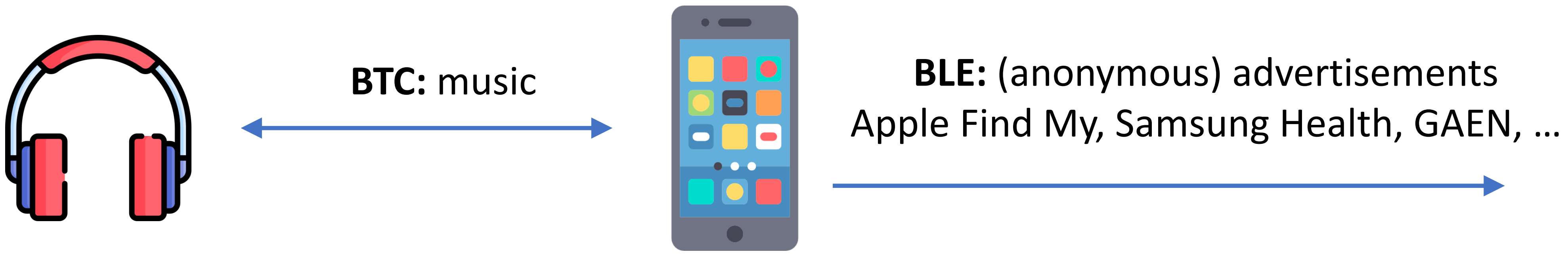
- **Observation 2:** proprietary advertisements may exhibit predictable patterns (below sequence numbers)



Figures from: Martin, PETS'19

BTC and BLE

- **Recall**
  - Bluetooth Classic (BTC) and Bluetooth Low Energy (BLE) separate technologies
  - But typically present in same device
- **Example scenario**
  - User listens music on his smartphone with BTC headphones
  - User's smartphone sends (anonymous) advertisements on the background



Linking BTC and BLE for tracking

- **Observations:**
  - BTC transmissions include a global identifier (BDADDR)
  - BTC and BLE modules combined to same chip with same

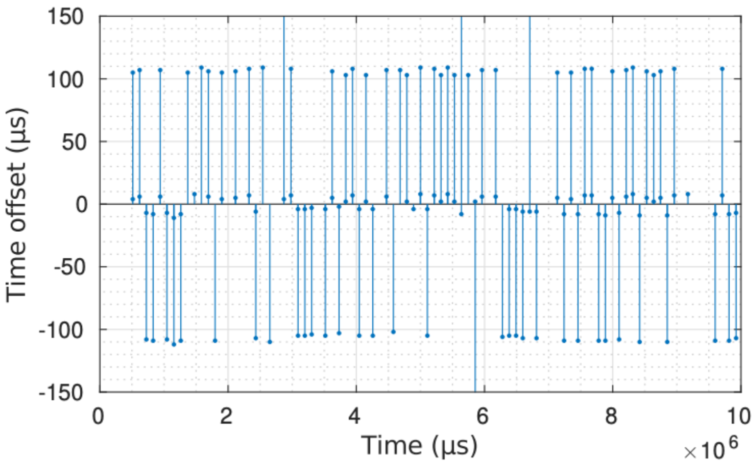
- **Attacker's goal:** Link anonymous BLE advertisements to BTC traffic?
  - Relevant when observes advertisements from multiple sources

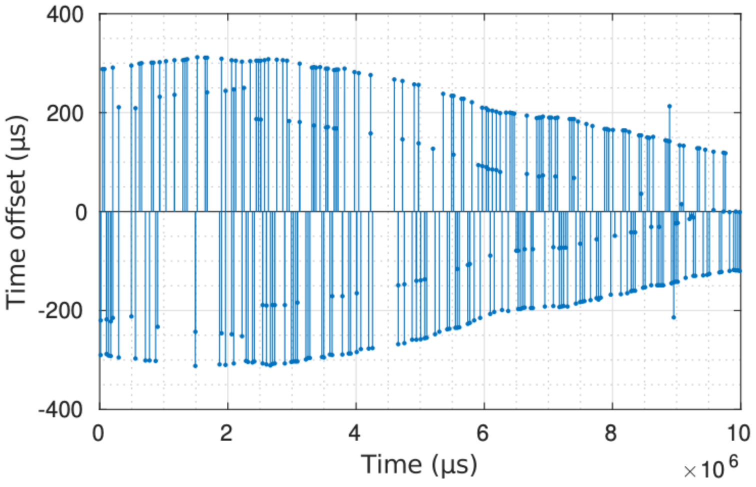
- **Attack idea:** (Ludant, S&P'21)
  - BTC and BLE modules on same chip with same clock source
  - Leverage timing-based side-channel to link BLE and BTC

Timing side-channel to link BTC and BLE

- Adversary's strategy
  1. Record BTC transmission and timestamps; extract global BDADDR
  2. Record BLE advertisements and timestamps
  3. Derive timing relation

Offset constant = same device





Offset non-constant = different device

Figures from: Ludant, S&P'19

Attack discussion

- In many apps BLE advertisements expected to be **unlinkable**
  - But that is not necessarily the case

- Such linking not severe privacy violation in itself
  - But **potential building block** for further attacks

- **Common pattern:**
  - Protocol may be private (unlinkable) in principle (or perfect isolation)
  - But the realization in practice is not private (unlinkable)

Recent research findings

- Ai et al. ”Blacktooth: Breaking through the Defense of Bluetooth in Silence.” CCS’22
  - Subtle vulnerabilities in BT
  - Allows adversary to **establish connection** with the victim **without any user involvement**
- Antonioli et al. “BLURtooth: Exploiting Cross-Transport Key Derivation in Bluetooth Classic and Bluetooth Low Energy.” AsiaCCS’22
  - **Cross-transport key derivation** (CTKD) functionality
  - Vulnerability enables adversary to **overwrite keys across BT and BLE**
- Wu et al. “Formal Model-Driven Discovery of Bluetooth Protocol Design Vulnerabilities.” S&P’22
  - **Extensive formal modeling** of BT and BLE
  - Found minor vulnerabilities such as the above CTKD issue

Part 3: Example Application

Covid contact tracing based on BLE beacons

Contact tracing

- Covid-19 pandemic triggered a new need
  - **Complement** traditional (manual) contact tracing with smartphone apps
- Contradicting requirements
  - Break chains of infection effectively
  - Do **not** create a tool of mass surveillance

Figure from: nzz.ch

- Many initiatives
  - **Our case study:** DP3T protocol (BLE advertisements)
  - Basis for Google/Apple Exposure Notification API (GAEN) and SwissCovid app

DP3T

- **Main idea:** smartphones broadcast and record BLE beacon
- Each beacon (BLE advertisement) contains randomized “EphID”
- **Goal:** user tracking difficult

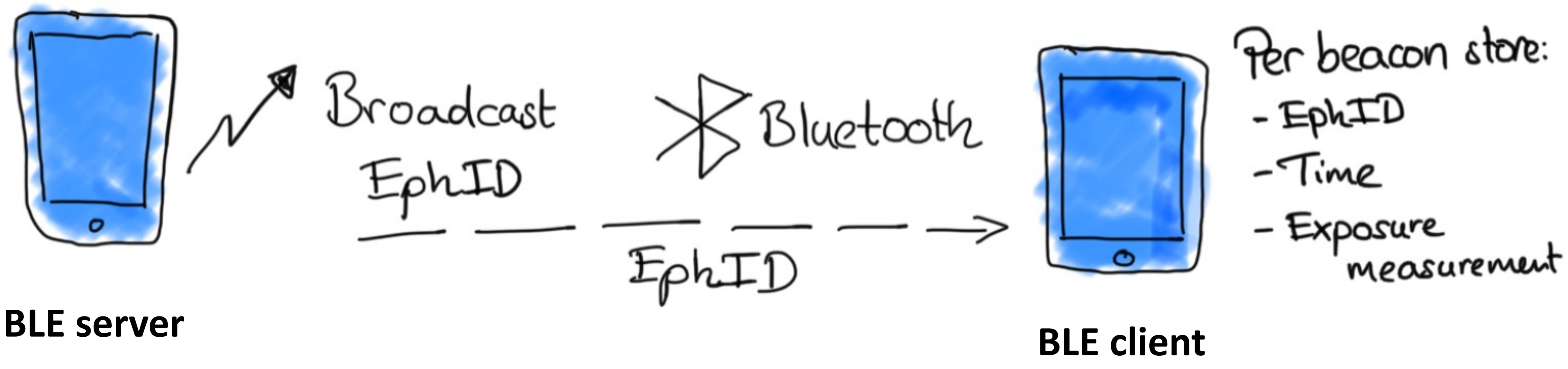


Figure from: Troncoso et al., White Paper 2020

3DPT protocol overview

- Daily operation
  - Picks seed *sk* and derive *EphIDs*
  - Broadcast and listen
  - Change *EphID* every 15 min
- Diagnosed patient (1)
  - Upload seed *sk* and date *t* to server (2)
  - Requires authorization
- Other devices
  - Download (*sk*, *t*) from server periodically (3)
  - Perform **local matching** by computing *EphIDs* (4)
  - Learn possible exposure date *t*

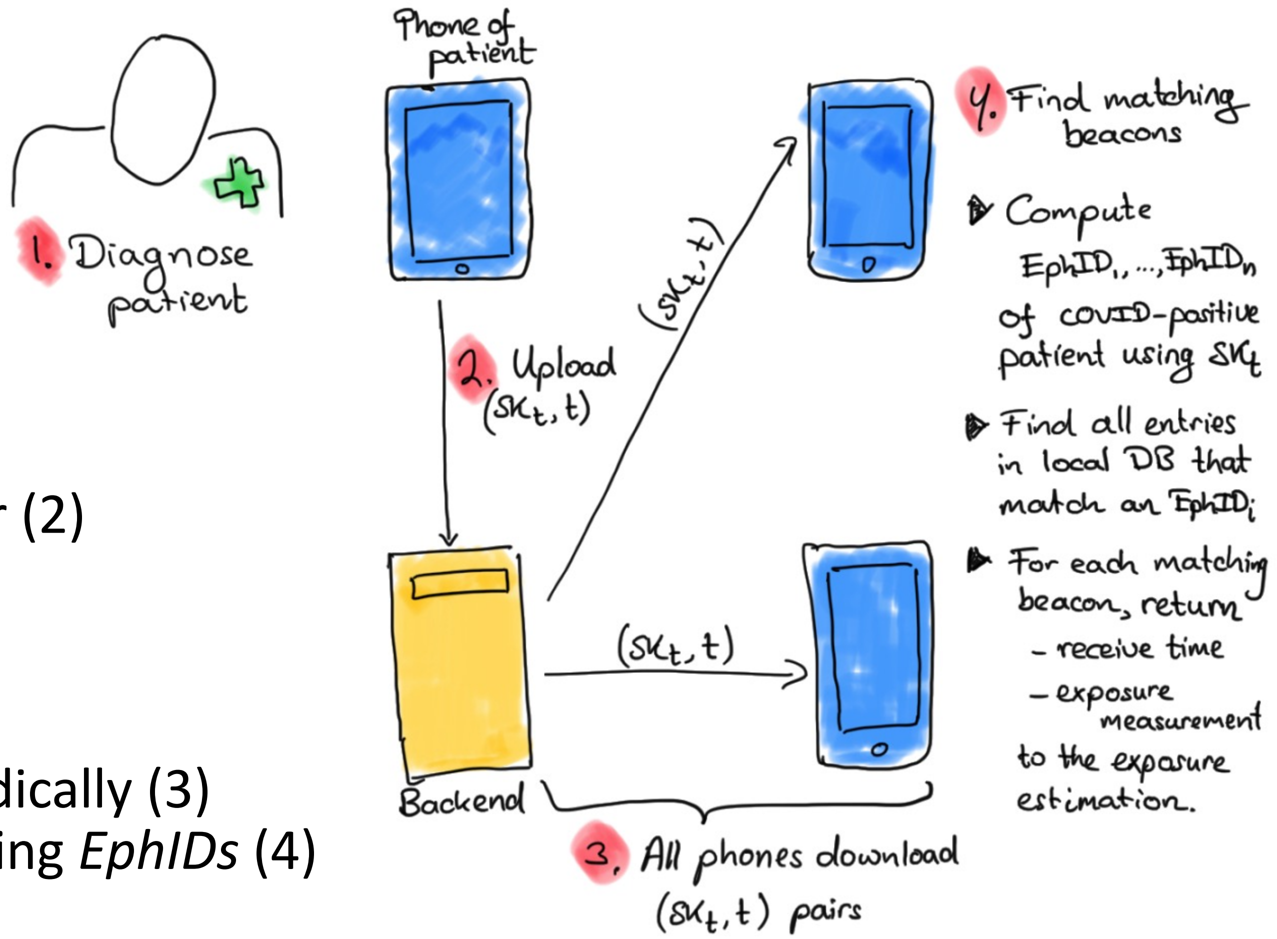


Figure from: Troncoso et al., White Paper 2020

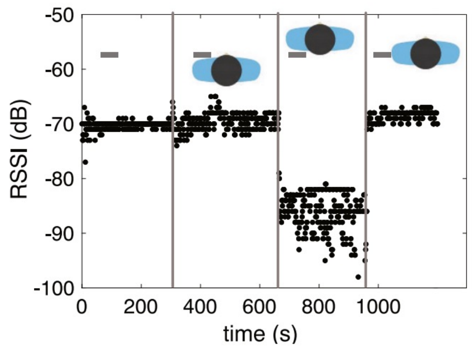
System realization

- Access to Bluetooth functionality controlled by Google/Apple
  - Google/Apple Exposure Notification API (GAEN)
  - National apps built on top of that

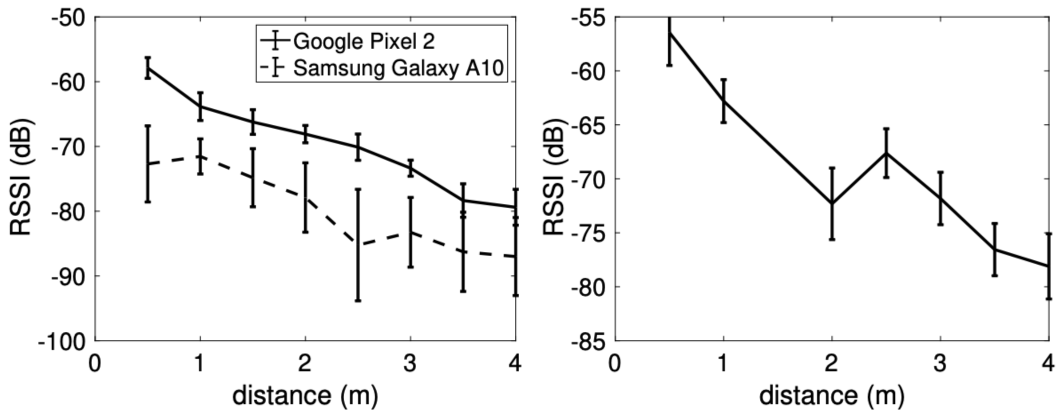
- What is “contact” or “exposure”?
  - **Example definition:** 15 min within 2 meters

- How to realize that?
  - Recall that BLE range up to hundreds of meters
  - **Duration:** control beacon sending and scanning schedule (easy)
  - **Distance:** approximate by measuring received beacon **signal strength** (tricky)

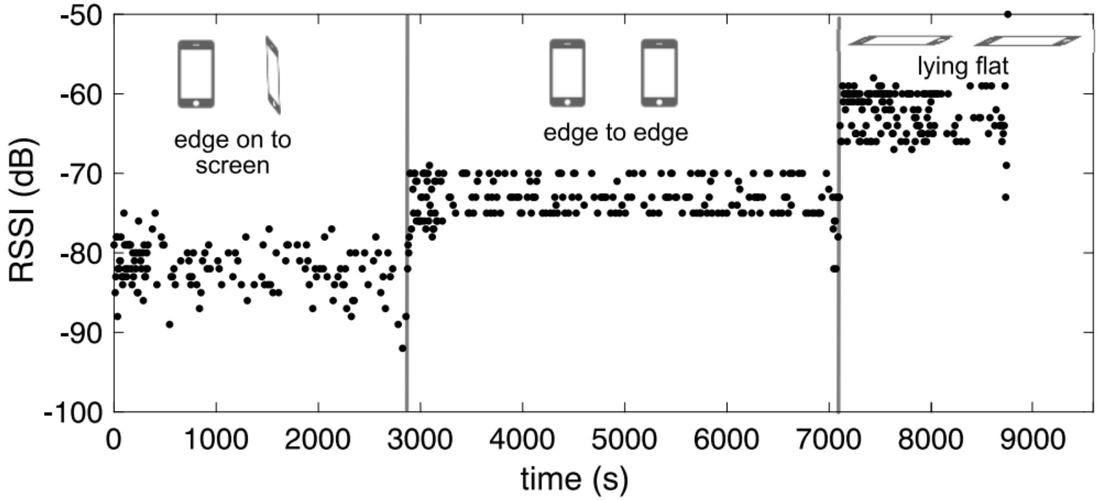
Distance approximation



- **Challenges:**
  - Propagation of radio signals is complex especially **indoors** (walls, furniture, …)
  - Also **person's body** or **device orientation** can influence measurements



(a) Outdoors (b) Indoors



Figures from: Leith, 2020

Effectiveness and critique

- Obviously not a “silver bullet” that ends pandemic

- UK case study (Wymant, Nature 2021)
  - 16M people installed the app
  - 1.7M notifications sent
  - Estimate that ~300K cases avoided (Wymant, Nature 2021)

- Tradeoffs (Cranor, 2020)
  - Privacy is good, but it can also cause uncomfortable uncertainties
  - User learns just the exposure date but not how or where?
  - Giving up some privacy might alleviate some of such concerns

Lecture summary

- BLE is widely-deployed technology
  - Smartphones, wearables, smart home sensors…
- BLE security and privacy is in general well designed
  - Modern pairing mechanism, strong link-layer encryption, privacy protections
- But systems can, and do, still fail in subtle ways
  - Example pattern: composition of multiple applications or protocols
  - Complete privacy (no tracking whatsoever) is really hard

Thank you!

Lecture end

References

- Antonioli et al. Key Negotiation Downgrade Attacks on Bluetooth and Bluetooth Low Energy. TOPS 2020.
- Becker et al. Tracking Anonymized Bluetooth Devices. PETS'19.
- Cranor. Digital Contact Tracing May Protect Privacy, But It Is Unlikely to Stop the Pandemic. Communication of the ACM, Nov 2020.
- Fawz et al. Protecting Privacy of BLE Device Users. Usenix Security'16.
- Landau. Digital exposure tools: Design for privacy, efficacy, and equity. Science, Sep 2021.
- Leith and Farrell. Coronavirus Contact Tracing: Evaluating The Potential Of Using Bluetooth Received Signal Strength For Proximity Detection. Computer Communication Review, 2020.
- Ludant et al. Linking Bluetooth LE & Classic and Implications for Privacy-Preserving Bluetooth-Based Protocols. S&P'21.

References

- Martin. Handoff All Your Privacy – A Review of Apple’s Bluetooth Low Energy Continuity Protocol. PETS’19.
- Ryan. Bluetooth: With Low Energy comes Low Security. WOOT’13.
- Troncoso et al. Decentralized Privacy-Preserving Proximity Tracing. White Paper, May 2020.
- Tschirschnitz et al. Method Confusion Attack on Bluetooth Pairing. S&P’21.
- Wen et al. FirmXRay: Detecting Bluetooth Link Layer Vulnerabilities From Bare-Metal Firmware. CCS’20.
- Wu et al. BLESA: Spoofing Attacks against Reconnections in Bluetooth Low Energy. WOOT’20.
- Wymant et al. The epidemiological impact of the NHS COVID-19 app. Nature, 20201.