# Advancing Security and Privacy of Bluetooth IoTs via Formal Protocol Analysis

Zhiqiang Lin
zlin@cse.ohio-state.edu

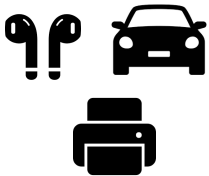
01/11/2024

# Outline

# Outline

# What is Bluetooth Low Energy

**Power Consumption:** High
**Communication Distance:** Short (10+ m)



**Power Consumption:** Low
**Communication Distance:** Long (100+ m)

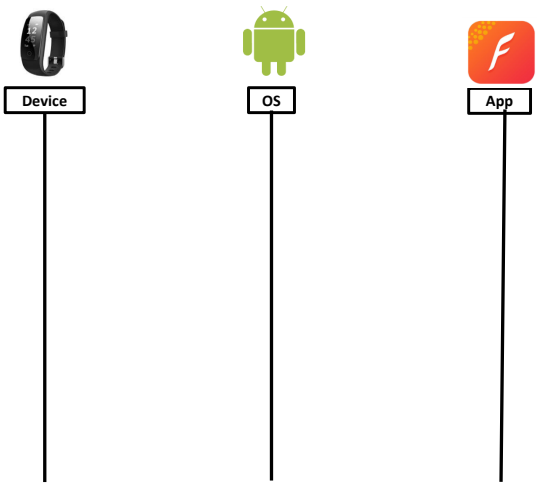
# Bluetooth Low Energy Applications



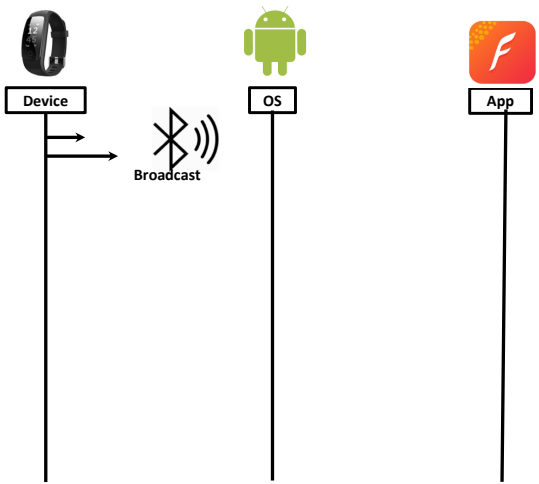




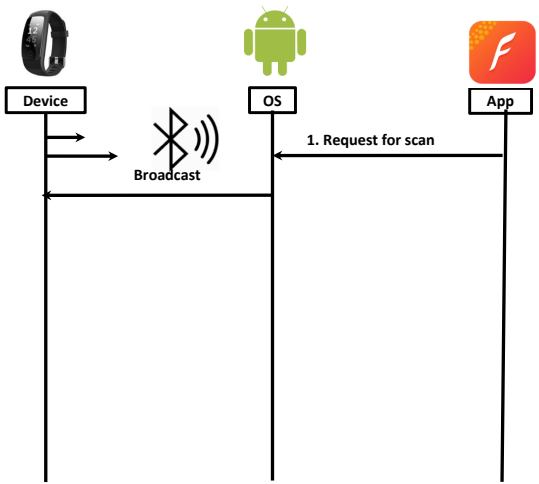
# Bluetooth Low Energy Communication



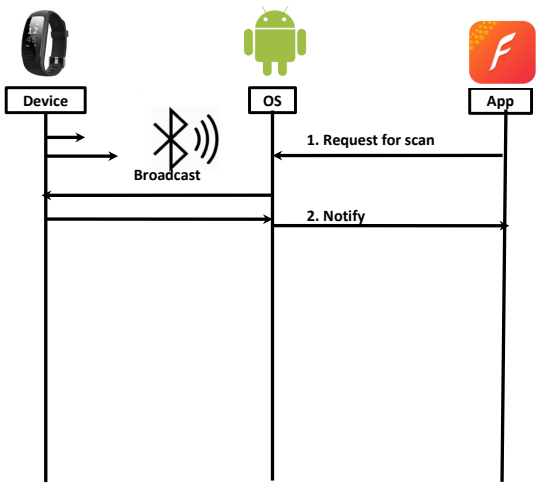
# Bluetooth Low Energy Communication



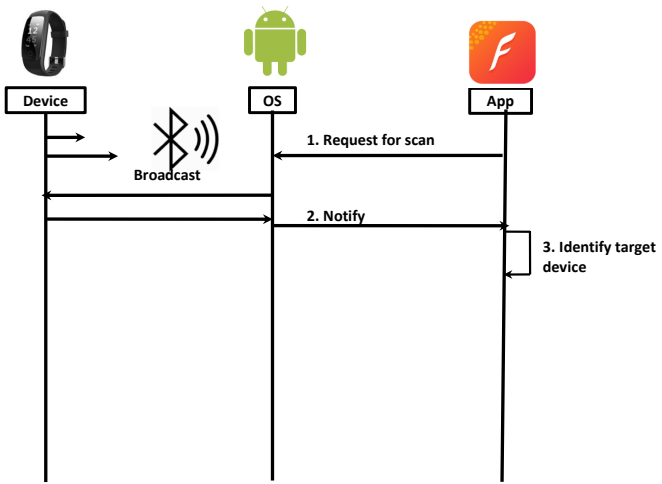
# Bluetooth Low Energy Communication



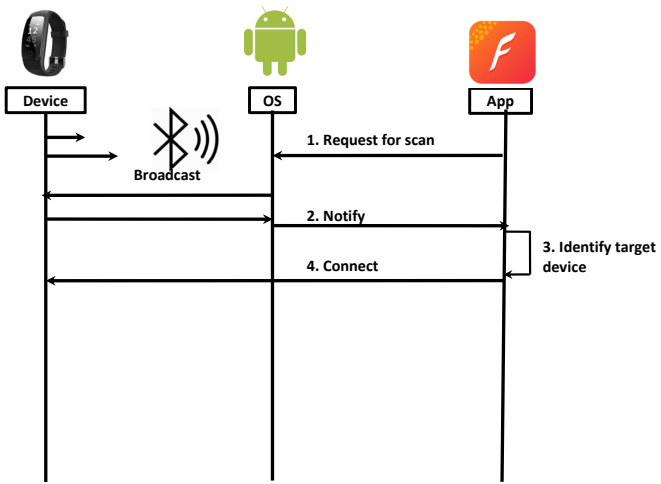
# Bluetooth Low Energy Communication



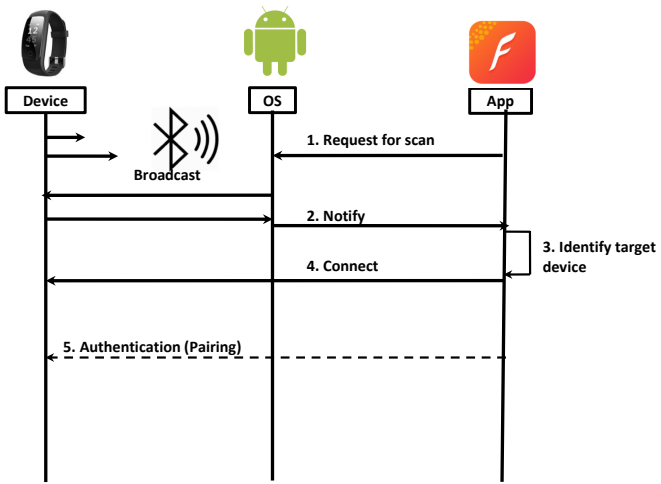
# Bluetooth Low Energy Communication



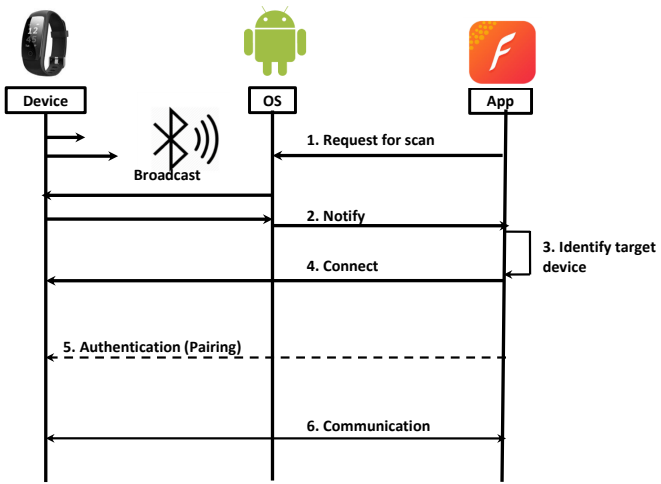
# Bluetooth Low Energy Communication



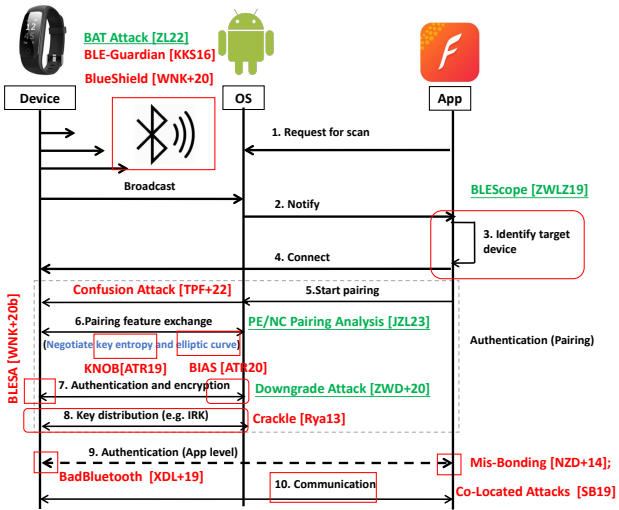
# Bluetooth Low Energy Communication



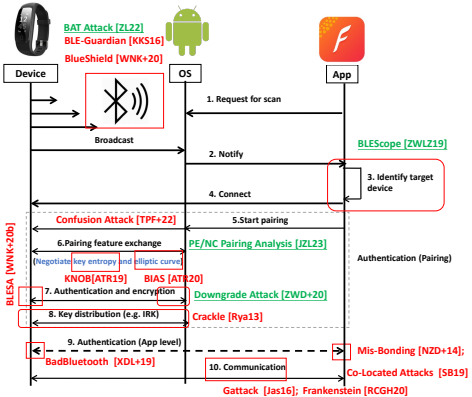
# Bluetooth Low Energy Communication



# Bluetooth Low Energy Communication



# Bluetooth Low Energy Communication




The goal of this project is to systematically uncover the attacks via formal methods

# Outline

# Mathematical Proof: A Simple Example

Assume a set of even integers
$E = \{..., -2, 0, 2, 4, ...\}$

And a set of odd integers
$O = \{..., -1, 1, 3, 5, ...\}$

**Property (Lemma)**

P = "Elements of E and O are distinct"

# Mathematical Proof: A Simple Example

## Math Algebra

$E = 2x \; for \; x \in Integer$
$O = 2y + 1 \; for \; y \in Integer$

## Proof by Contradiction

not P = "There exist some common elements among E and O"

# Mathematical Proof: A Simple Example

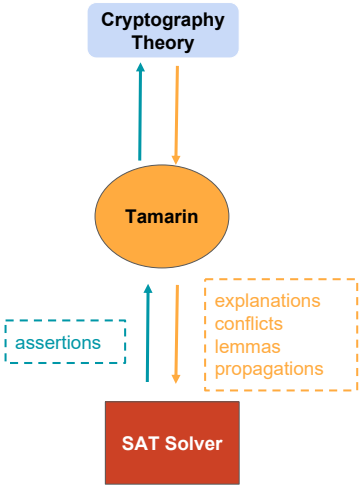
**Proof**

Start with assuming for some x and y
$2x = 2y + 1$ holds true
$\Rightarrow 2(x - y) = 1$

# Mathematical Proof: A Simple Example

| **Possible Sub-Cases of X-Y** | **Generally known axioms (Generalized Constraint Resolution Methods)** | **applying axioms to 2(x-y) = 1** | **Conclusion (Contradicting all not P system states)** |
|---|---|---|---|
| 0 | 0 is the only neutral difference of Integers | 2*0=1 | Contradict |
| negatives | all negatives differences of integers are -1 or less {..., -3, -2, -1} | 2*(-1 or less)= 1 | Contradict |
| positive | all positive differences of Integers are 1 or more {1, 2, 3, ...} | 2*(1 or more)= 1 | Contradict |

# How does Protocol Verification (Tamarin) Work?




- **Dec**(**Enc**(msg, key)) = msg
- **Sign**(msg, privKey) = **Verify**(msg, pubKey)
- Adversary Replay

…

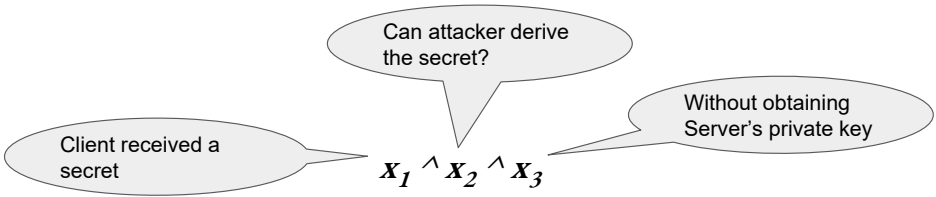
- Find next proof requirements
- Choose which SAT problems to solve first
- Convert problem algebra theory into SAT problems

…

$$x_1 \wedge x_2 \wedge x_3$$

- Does SAT problem have a solution?

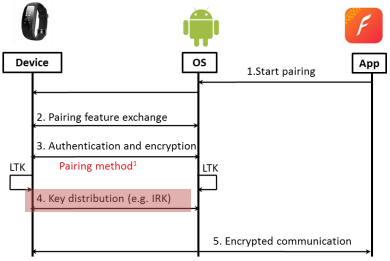
# How does Protocol Verification (Tamarin) Work?



# (Privacy) How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**



# (Privacy) How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

# (Privacy) How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

**(I) RPA Generation**

# (Privacy) How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

**(I) RPA Generation**

$$rpa_p = prand_{24} || H_{24}(Prand_{24} || irk_p)$$

# (Privacy) How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

**(I) RPA Generation**

$$rpa_p = prand_{24} : H_{24}(Prand_{24}||irk_p)$$

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

# (Privacy) How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

**(I) RPA Generation**

$$rpa_p = prand_{24} : H_{24}(Prand_{24}||irk_p)$$

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

# (Privacy) How to Avoid Being Tracked: MAC Address Randomization

Identity Resolving Key ($irk_p$)

Identity Resolving Key ($irk_c$)

**(I) RPA Generation**

$$rpa_p = prand_{24} : H_{24}(Prand_{24}||irk_p)$$

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

**(II) RPA Resolution**

# (Privacy) How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

## (I) RPA Generation

$$rpa_p = prand_{24} \| H_{24}(Prand_{24} || irk_p)$$

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

## (II) RPA Resolution

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

# (Privacy) How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

## (I) RPA Generation

$$rpa_p = prand_{24} \,||\, H_{24}(Prand_{24}||irk_p)$$

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

## (II) RPA Resolution

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

$$rpa_c = prand_{24} \,||\, H_{24}(Prand_{24}||irk_c)$$

# (Privacy) How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

## (I) RPA Generation

$$rpa_p = prand_{24} \,|\, H_{24}(Prand_{24}||irk_p)$$

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

## (II) RPA Resolution

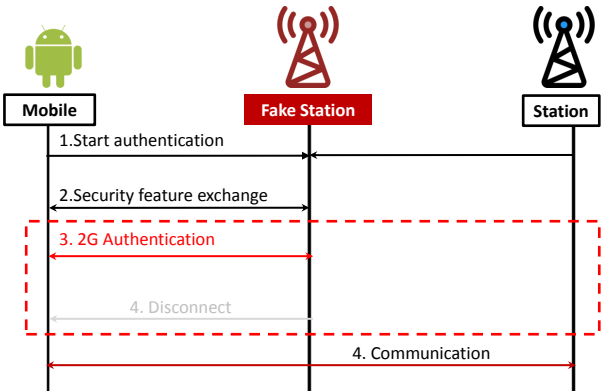
| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

$$rpa_c = prand_{24} \,|\, H_{24}(Prand_{24}||irk_c)$$

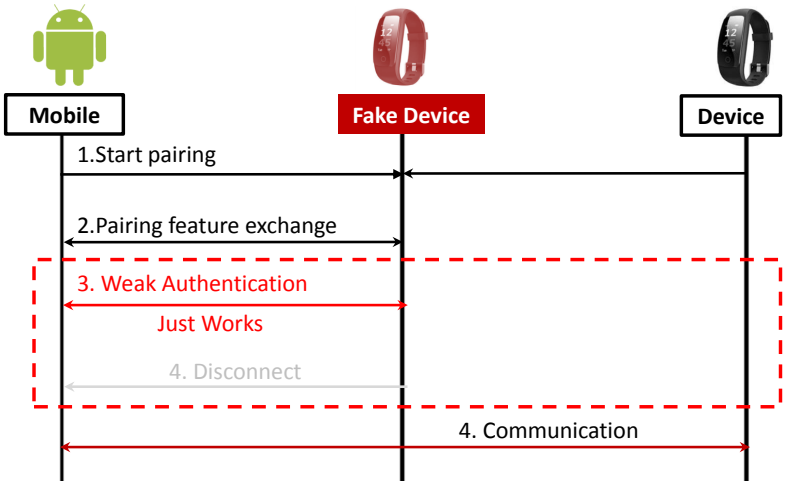
$irk_p = irk_c$ ➔ $rpa_p = rpa_c$

# Outline

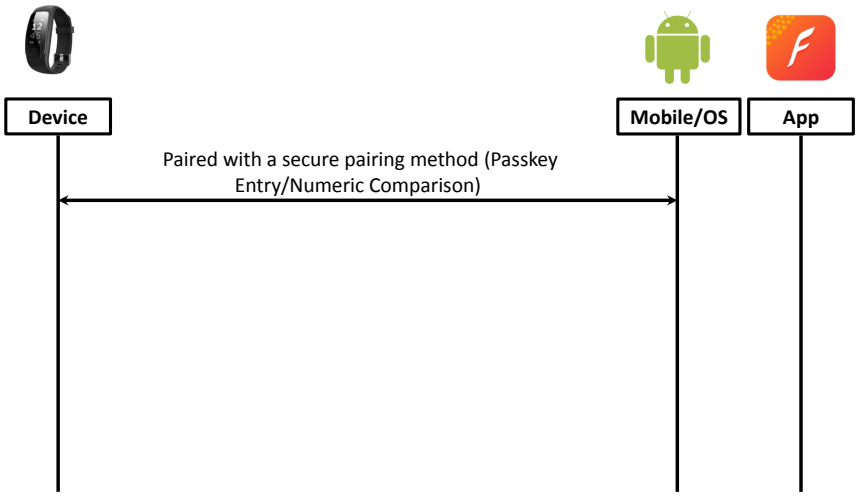
# Our Downgrade Attacks against Bluetooth Low Energy [USENIX'20]



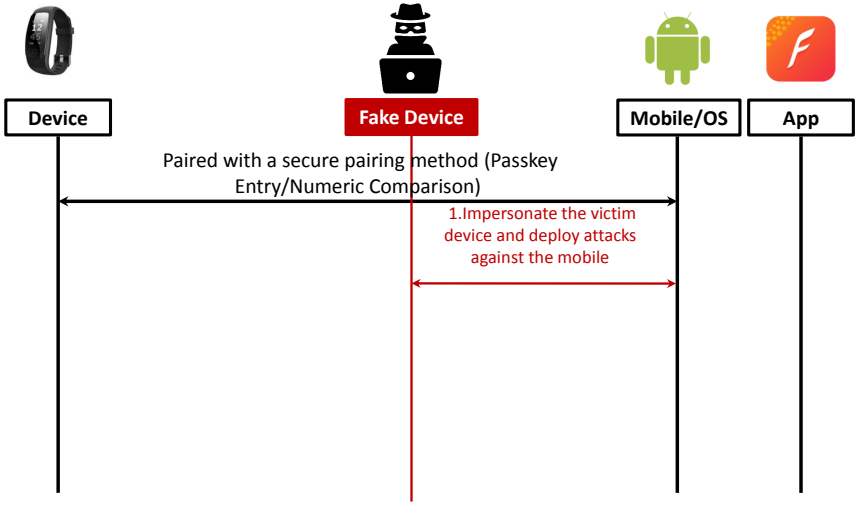
# Our Downgrade Attacks against Bluetooth Low Energy [USENIX'20]



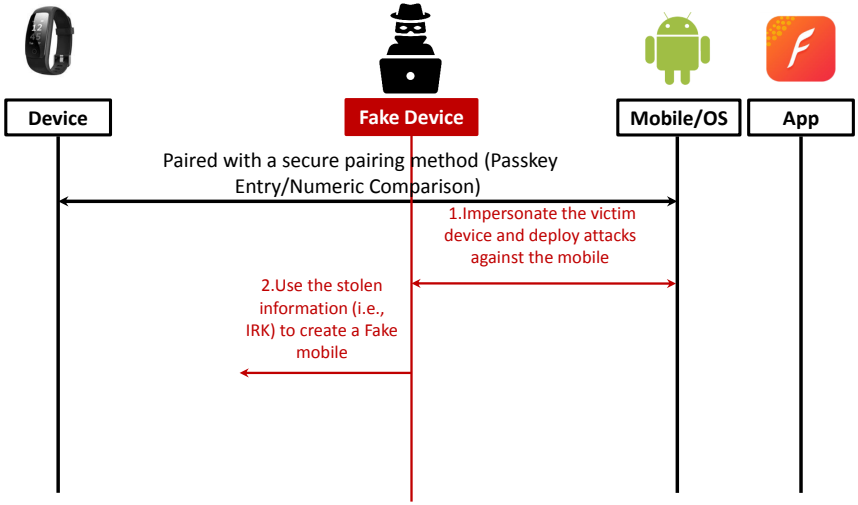
# Our Downgrade Attacks against Bluetooth Low Energy [USENIX'20]



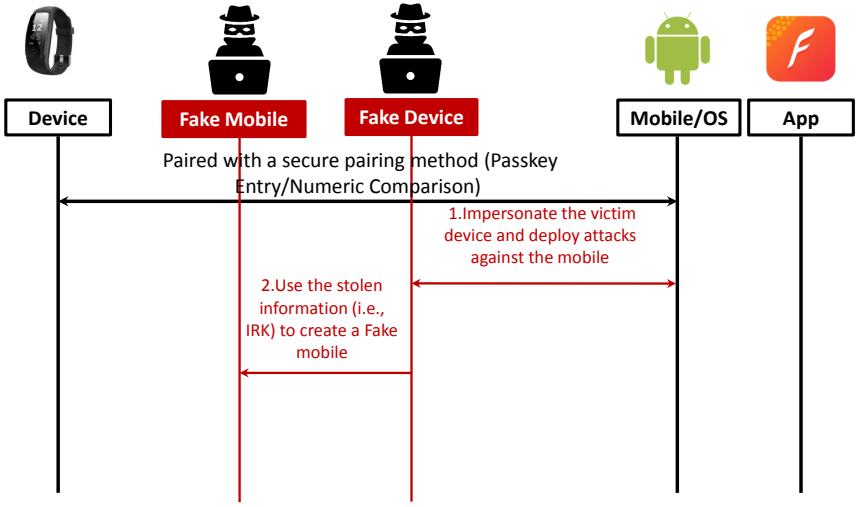
# Our Downgrade Attacks against Bluetooth Low Energy [USENIX'20]



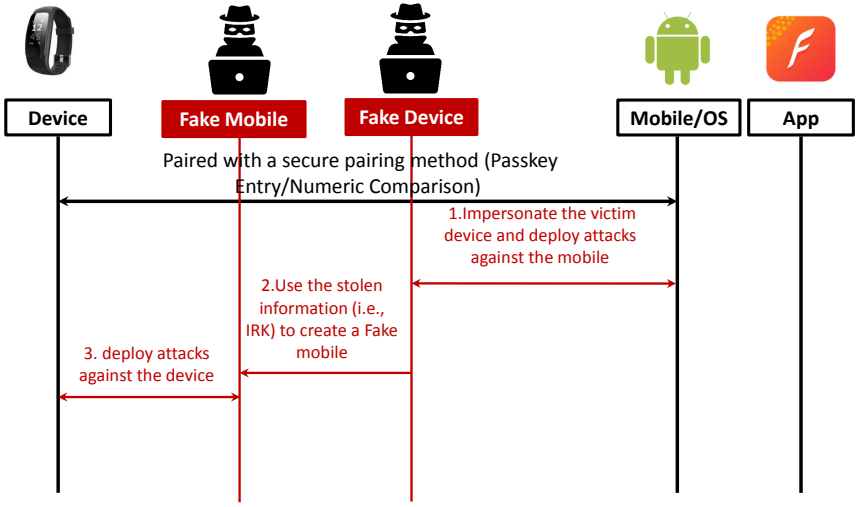
# Our Downgrade Attacks against Bluetooth Low Energy [USENIX'20]



# Our Downgrade Attacks against Bluetooth Low Energy [USENIX'20]

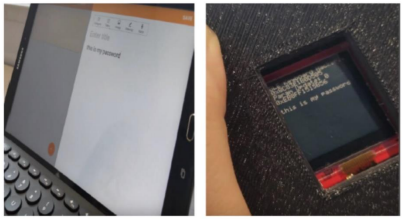


# Our Downgrade Attacks against Bluetooth Low Energy [USENIX'20]



# Our Downgrade Attacks against Bluetooth Low Energy [USENIX'20]


The Tested BLE devices




MITM attack against BLE keyboards

**CVE-2020-9770**

# Our BAT Privacy Attacks [CCS'22]: Allowlist-based Side Channel

58:D7:8E:C7:8e:31

| NO. | Time | Source | Destination | TYPE |
|---|---|---|---|---|
| 1 | 00:00:04 | 58:D7:8E:C7:8e:31 | Broadcast | ADV_IND |

# Our BAT Privacy Attacks [CCS'22]: Allowlist-based Side Channel



| NO. | Time | Source | Destination | TYPE |
|---|---|---|---|---|
| 1 | 00:00:04 | 58:D7:8E:C7:8e:31 | Broadcast | ADV_IND |
| 2 | 00:00:08 | 7e:D7:8E:C7:8e:51 | 58:D7:8E:C7:8e:31 | SCAN_REQ |
| 3 | 00:00:12 | 58:D7:8E:C7:8e:31 | Broadcast | SCAN_RSP |

# Our BAT Privacy Attacks [CCS'22]: Allowlist-based Side Channel


| NO. | Time | Source | Destination | TYPE |
|---|---|---|---|---|
| 1 | 00:00:04 | 58:D7:8E:C7:8e:31 | Broadcast | ADV_IND |
| 2 | 00:00:08 | 7e:D7:8E:C7:8e:51 | 58:D7:8E:C7:8e:31 | SCAN_REQ |
| 3 | 00:00:12 | 58:D7:8E:C7:8e:31 | Broadcast | SCAN_RSP |
| 4 | 00:00:16 | 4f:b7:8E:C7:8e:38 | 58:D7:8E:C7:8e:31 | SCAN_REQ |
| 5 | 00:00:24 | 58:D7:8E:C7:8e:31 | Broadcast | ADV_IND |

# Our BAT Privacy Attacks [CCS'22]: Allowlist-based Side Channel

58:D7:8E:C7:8e:31

7e:D7:8E:C7:8e:51

4f:b7:8E:C7:8e:38

| NO. | Time | Source | Destination | TYPE |
|---|---|---|---|---|
| 1 | 00:00:04 | 58:D7:8E:C7:8e:31 | Broadcast | ADV_IND |
| 2 | 00:00:08 | 7e:D7:8E:C7:8e:51 | 58:D7:8E:C7:8e:31 | SCAN_REQ |
| 3 | 00:00:12 | 58:D7:8E:C7:8e:31 | Broadcast | SCAN_RSP |
| 4 | 00:00:16 | 4f:b7:8E:C7:8e:38 | 58:D7:8E:C7:8e:31 | SCAN_REQ |
| 5 | 00:00:24 | 58:D7:8E:C7:8e:31 | Broadcast | ADV_IND |

.....

| | | | | |
|---|---|---|---|---|
| 200 | 00:15:08 | 73:D7:8E:C7:8e:45 | 58:D7:8E:C7:8e:31 | SCAN_REQ |
| 201 | 00:15:12 | 58:D7:8E:C7:8e:31 | Broadcast | SCAN_RSP |

# Our BAT Privacy Attacks [CCS'22]: Allowlist-based Side Channel


| NO. | Time | Source | Destination | TYPE |
|---|---|---|---|---|
| 1 | 00:00:04 | 58:D7:8E:C7:8e:31 | Broadcast | ADV_IND |
| 2 | 00:00:08 | 7e:D7:8E:C7:8e:51 | 58:D7:8E:C7:8e:31 | SCAN_REQ |
| 3 | 00:00:12 | 58:D7:8E:C7:8e:31 | Broadcast | SCAN_RSP |
| 4 | 00:00:16 | 4f:b7:8E:C7:8e:38 | 58:D7:8E:C7:8e:31 | SCAN_REQ |
| 5 | 00:00:24 | 58:D7:8E:C7:8e:31 | Broadcast | ADV_IND |
| ..... | | | | |
| 200 | 00:15:08 | 73:D7:8E:C7:8e:45 | 58:D7:8E:C7:8e:31 | SCAN_REQ |
| 201 | 00:15:12 | 58:D7:8E:C7:8e:31 | Broadcast | SCAN_RSP |

1. Cache
2. Timing
3. Power
4. Votage
5. Electromagnetic
6. Acoustic
7. Allow-list
8. ...

# Our BAT Privacy Attacks [CCS'22]: MAC Address Replay

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

# Our BAT Privacy Attacks [CCS'22]: MAC Address Replay

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

**(I) RPA Generation**

$$rpa_p = prand_{24} || H_{24}(Prand_{24} || irk_p)$$

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

# Our BAT Privacy Attacks [CCS'22]: MAC Address Replay

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

## (I) RPA Generation

$$rpa_p = prand_{24} || H_{24}(Prand_{24} || irk_p)$$

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

## (II) RPA Resolution

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

$$rpa_c = prand_{24} || H_{24}(Prand_{24} || irk_c)$$

$irk_p = irk_c$ ➔ $rpa_p = rpa_c$

**$rpa_p$**

# Our BAT Privacy Attacks [CCS'22]: MAC Address Replay

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

**(I) RPA Generation**

$$rpa_p = prand_{24} || H_{24}(Prand_{24} || irk_p)$$

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

**No Identity Resolving Key**

**RPA Replay ($rpa'_p$)**

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

**(II) RPA Resolution**

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

$$rpa_c = prand_{24} || H_{24}(Prand_{24} || irk_c)$$

$irk_p = irk_c$ ➔ $rpa_p = rpa_c$

**$rpa_p$**

# Our BAT Privacy Attacks [CCS'22]: MAC Address Replay

**Identity Resolving Key ($irk_p$)**

**Identity Resolving Key ($irk_c$)**

**(I) RPA Generation**

$$rpa_p = prand_{24} || H_{24}(Prand_{24} || irk_p)$$

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

**No Identity Resolving Key**

**RPA Replay ($rpa'_p$)**

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

**(II) RPA Resolution**

| Type | rand | Hash |
|---|---|---|
| 01 ( 2bits ) | 0x00...3 ( 22bits ) | 0x00...04 ( 24bits ) |

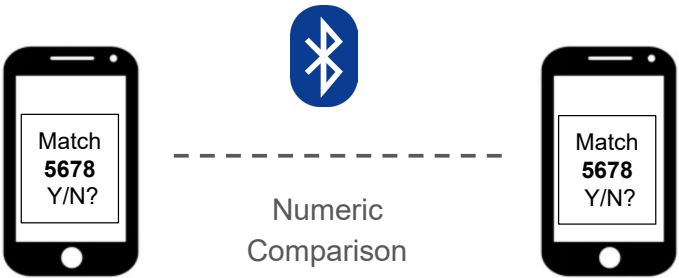
$$rpa_c = prand_{24} || H_{24}(Prand_{24} || irk_c)$$

$irk_p = irk_c \rightarrow rpa_p = rpa_c$

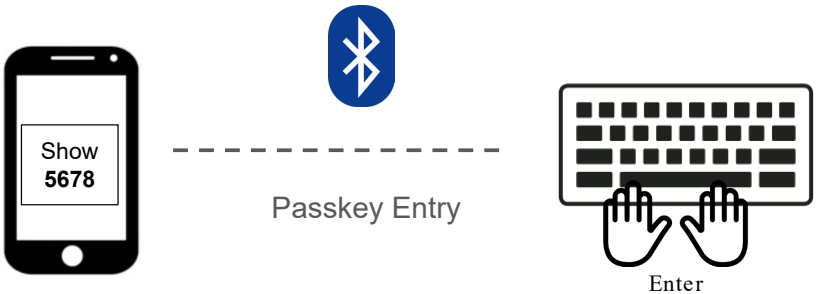
$rpa_p$

$rpa'_p$

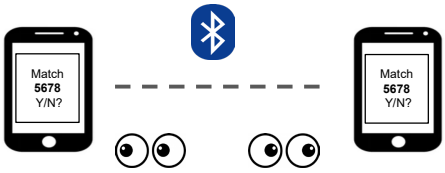
# Using Protocol Verification to Identify Confusion Attacks [NDSS'23]



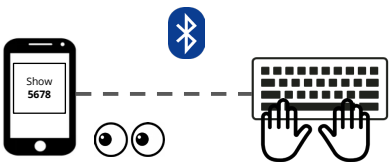
# Using Protocol Verification to Identify Confusion Attacks [NDSS'23]



# Using Protocol Verification to Identify Confusion Attacks [NDSS'23]

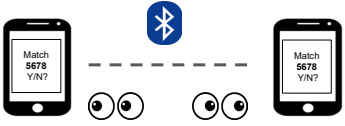


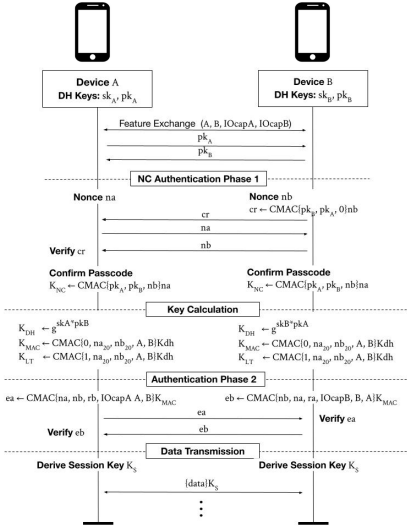

Numeric Comparison



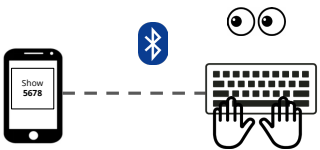

Passkey Entry

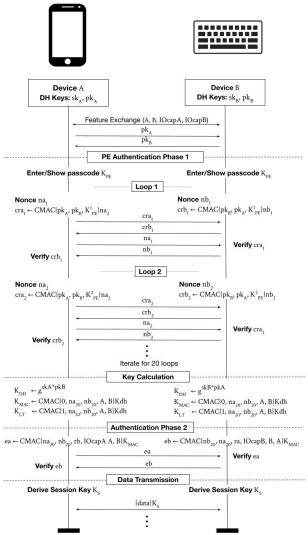
# Using Protocol Verification to Identify Confusion Attacks [NDSS'23]



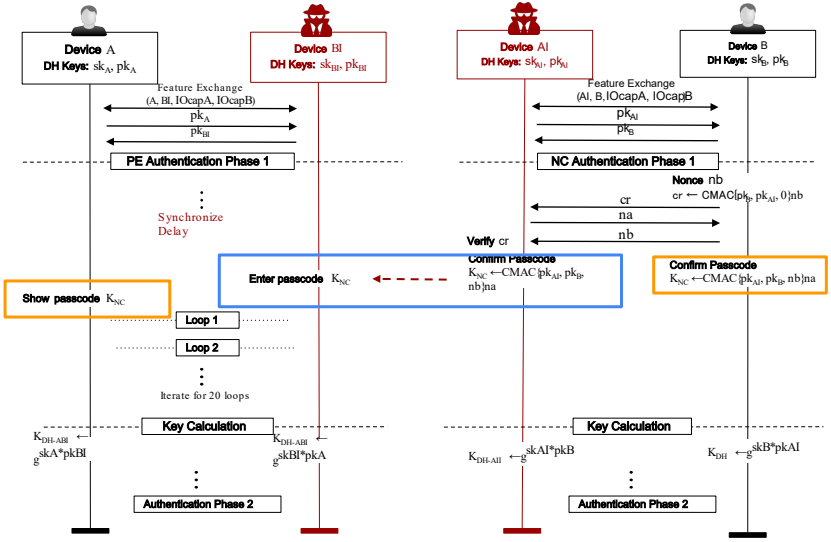



# Using Protocol Verification to Identify Confusion Attacks [NDSS'23]

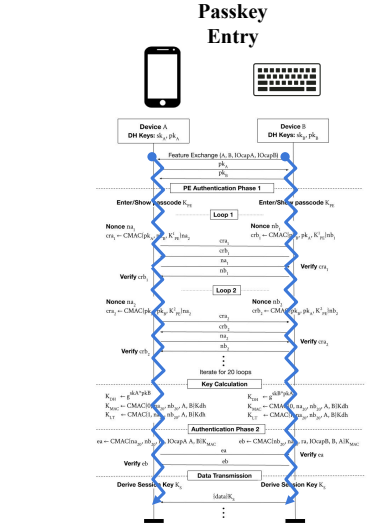


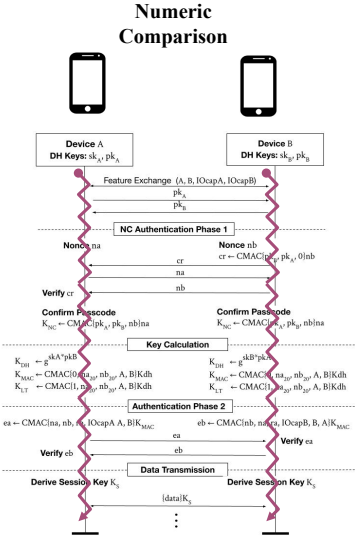



# Using Protocol Verification to Identify Confusion Attacks [NDSS'23]

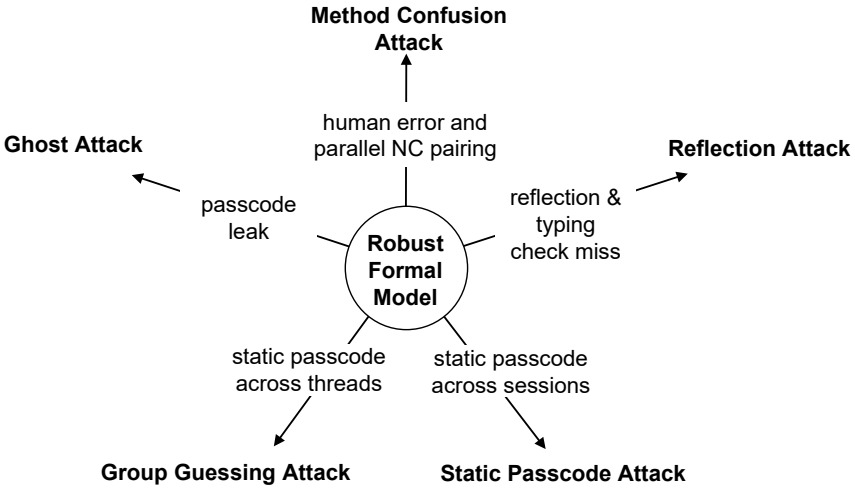


# Using Protocol Verification to Identify Confusion Attacks [NDSS'23]



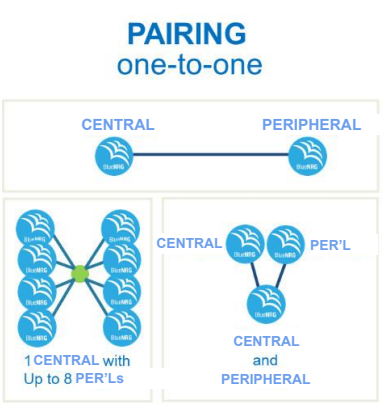



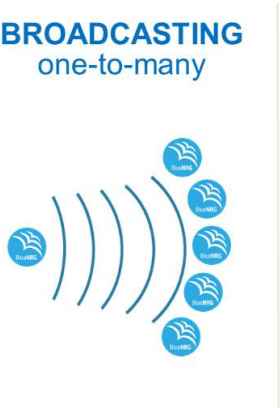
# Using Protocol Verification to Identify Confusion Attacks [NDSS'23]

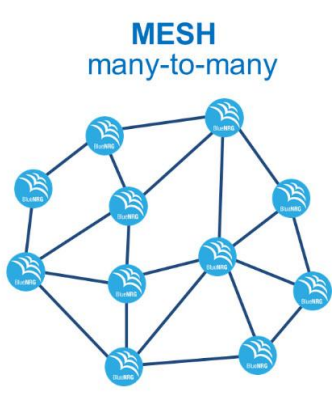


# Outline

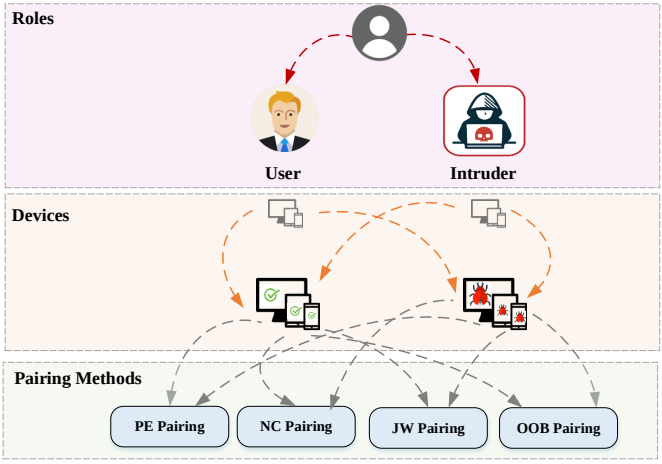
# Task 1: Developing a Formal Model for Full Spectrum of the Protocols



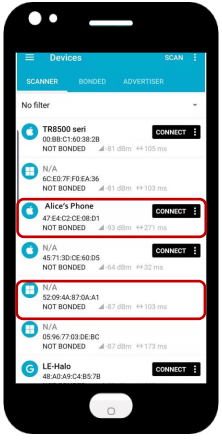



# Task 2: Developing a Formal Model for All Pairing Methods (Security)



# Task 3: Modelling Linkability of BLE Devices for **Privacy**



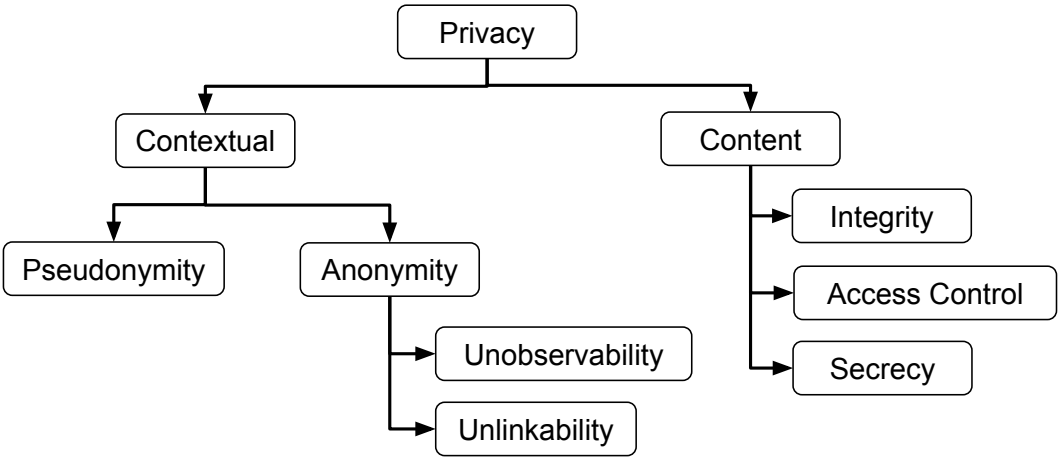

Alice's phone

Bob's phone

**T1: 52:09:4A:87:0A:A1**

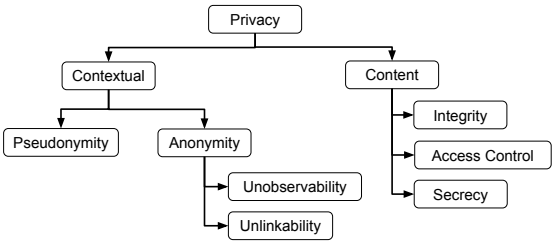
**T2: 52:09:4A:87:0A:A1**

# Task 3: Modelling Linkability of BLE Devices for **Privacy**

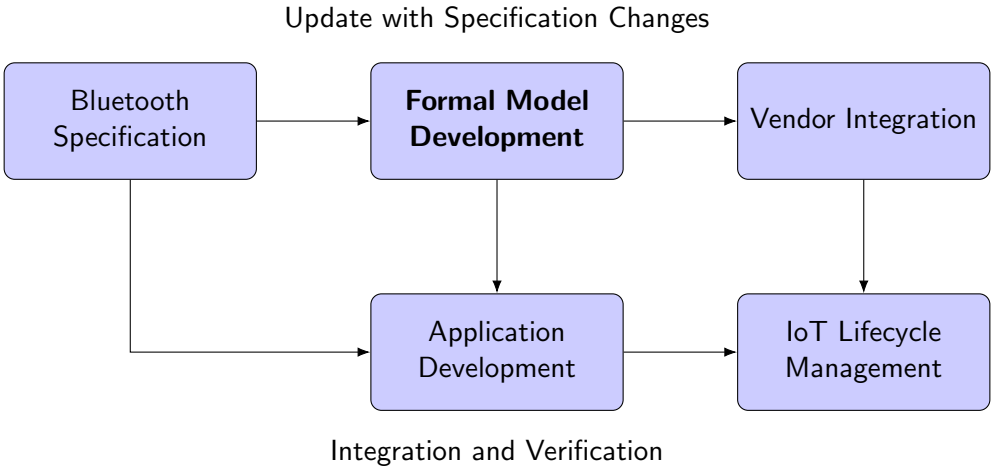


# Task 3: Modelling Linkability of BLE Devices for **Privacy**

- **Unlinkability** implies that an attacker cannot relate multiple observations of user actions.
- Proposed Solution: Using Observational Equivalence
  - Finding differences between all possible execution traces of two annotated systems: left and right.



# Task 4: Integrating Formal Verification into the Supply-chain



# Deliverables

1. **Formal models of the Bluetooth protocol:** Complete formal models for the Bluetooth Low Energy protocol, covering its various aspects, including device pairing, authentication, and communication.
2. **Analysis of the discovered vulnerabilities:** A report detailing the identified vulnerabilities in Bluetooth, based on the formal models developed.
3. **Open-source implementation:** A prototype implementation of the proposed security enhancements for the Bluetooth protocol, released as an open-source project for the community.
4. **Research publications:** Publish findings in peer-reviewed venues to contribute to the global knowledge base on Bluetooth IoT security and privacy.