Rethinking the Security and Privacy of Bluetooth Low Energy

Dr. Zhiqiang Lin
Distinguished Professor of Engineering
zlin@cse.ohio-state.edu

10/14/2022

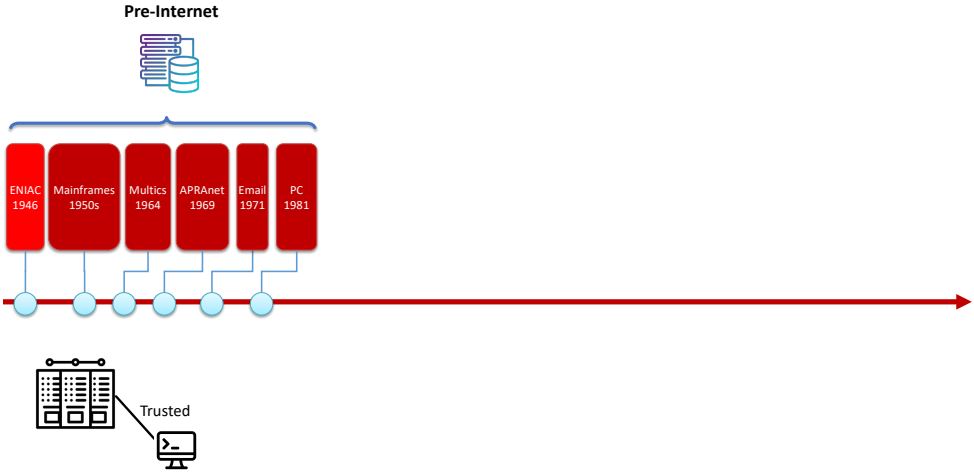
Outline

Outline

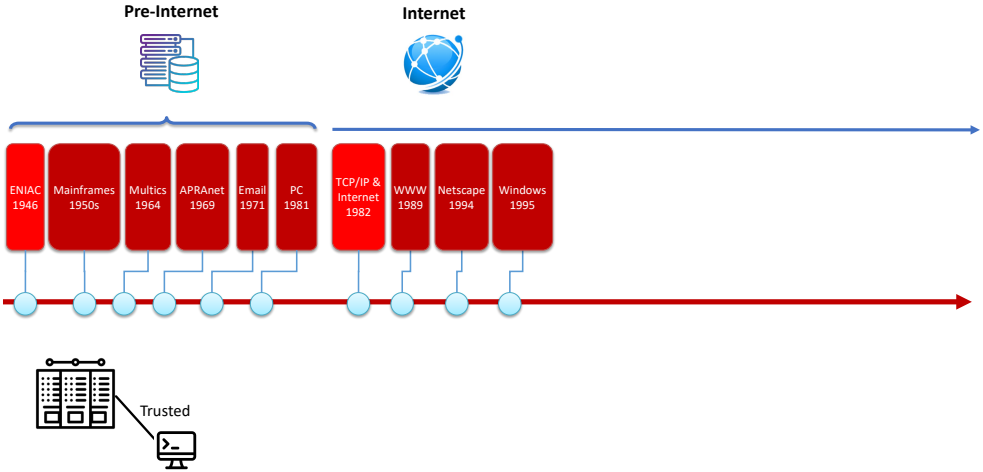
The evolution of our computing

The evolution of our computing

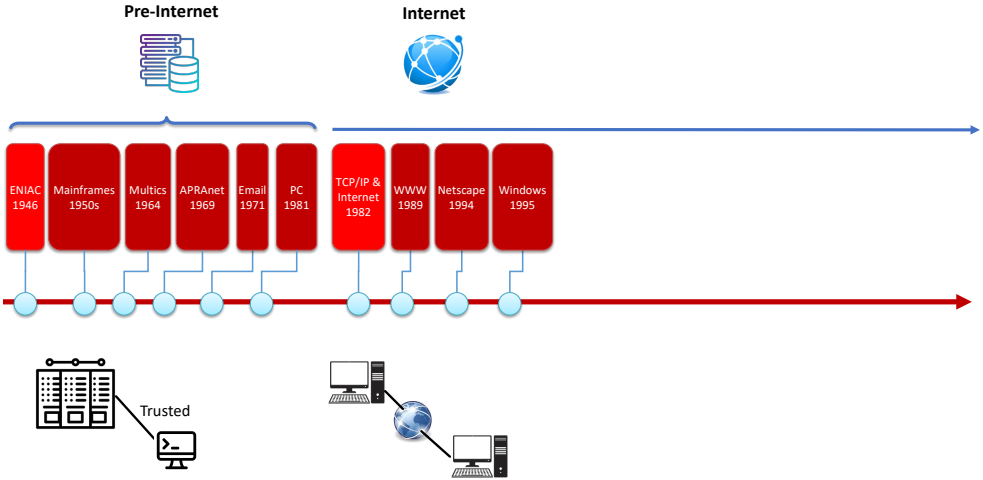
The evolution of our computing



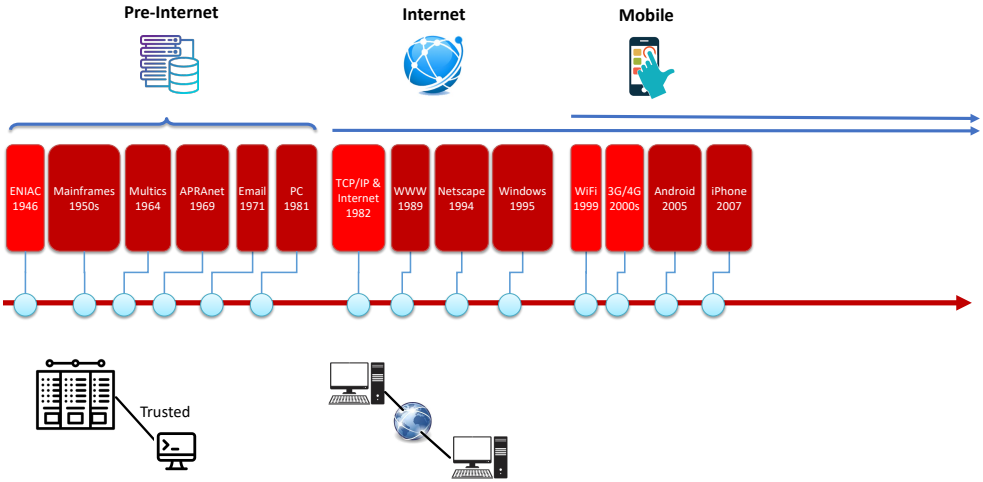
The evolution of our computing



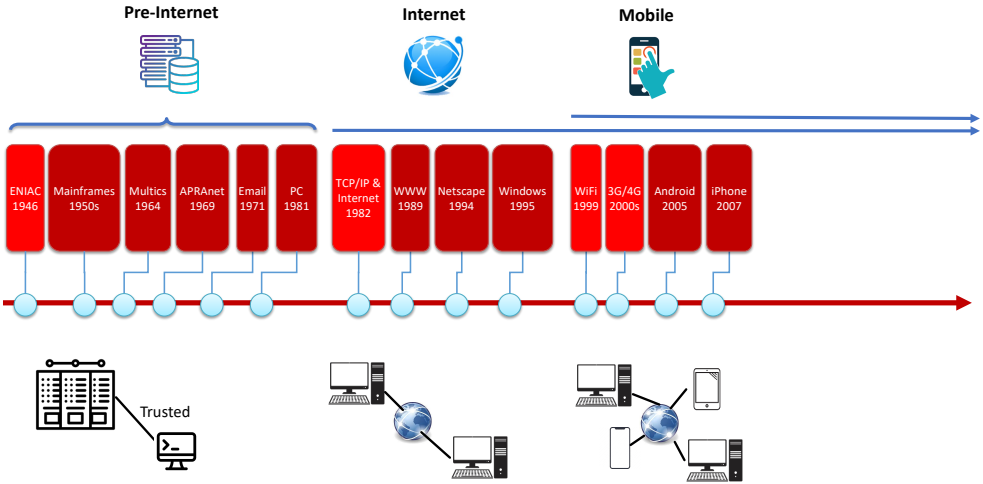
The evolution of our computing



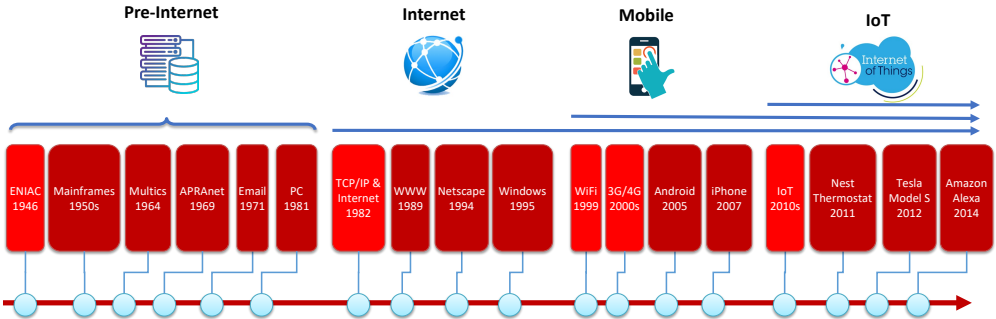
The evolution of our computing

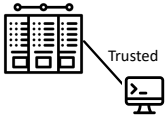


The evolution of our computing

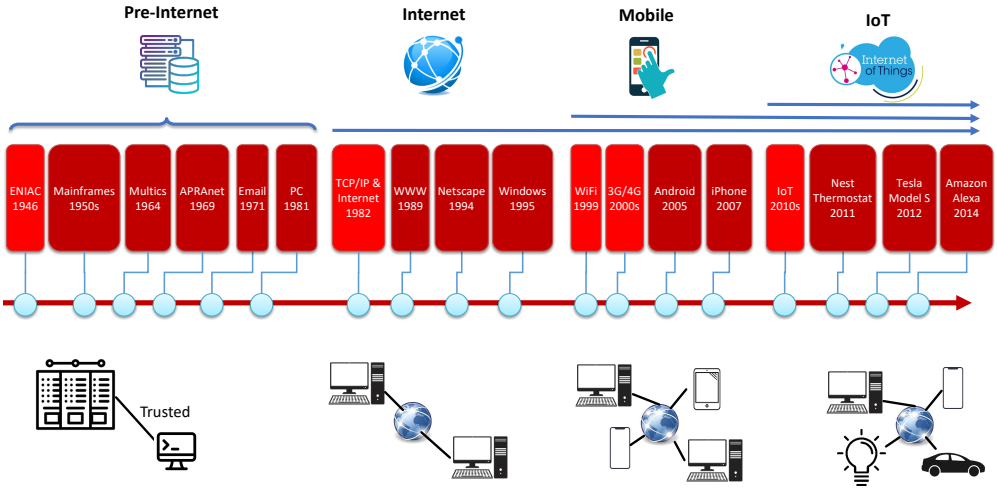


The evolution of our computing

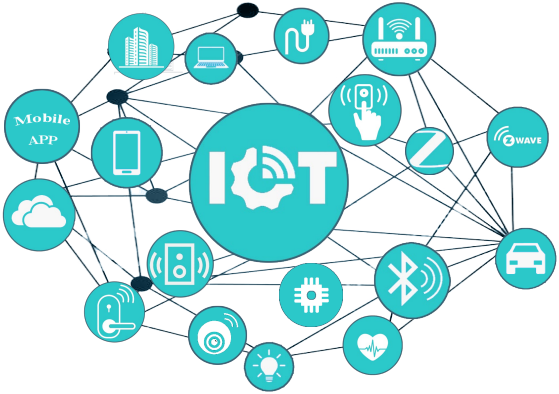




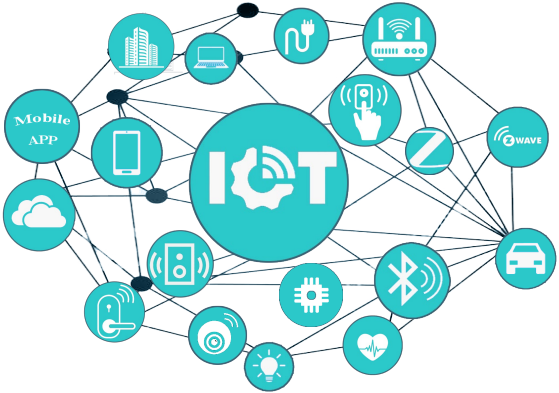
The evolution of our computing



Why IoT



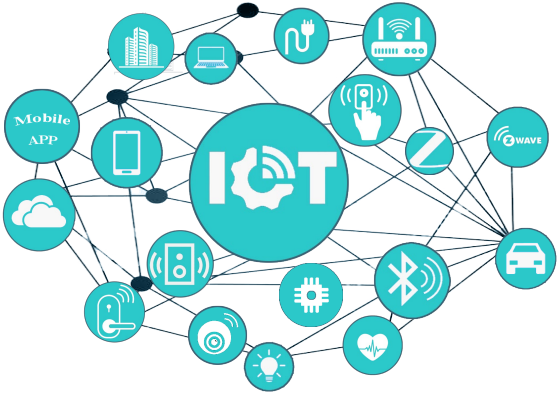
Why IoT



CIA:

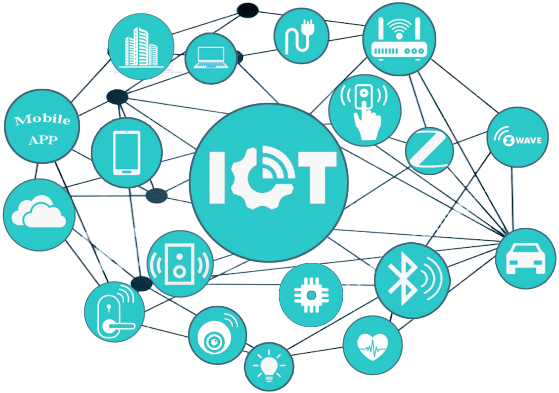
1. Convenience
2. Intelligence
3. Automation

Why IoT



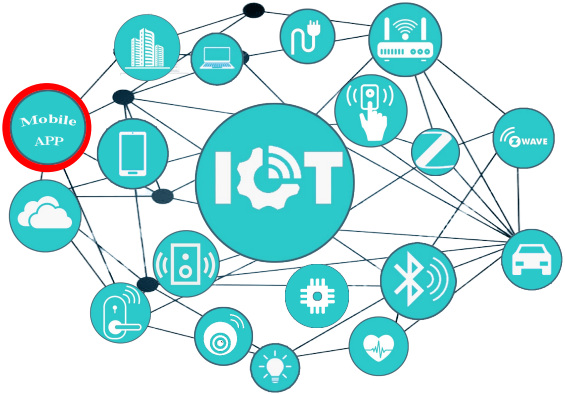
Security + Privacy

Our Recent Works in IoT Security and Privacy

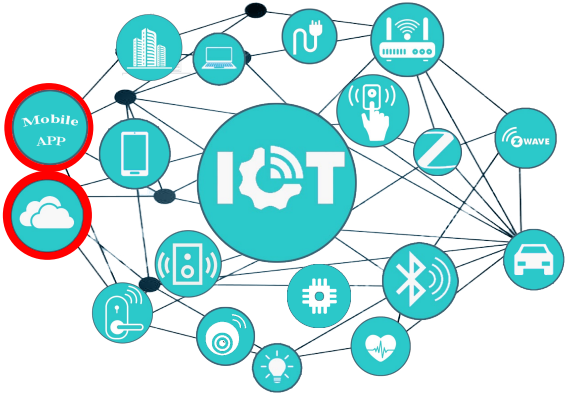


Our Recent Works in IoT Security and Privacy

1. Automatic Uncovering of Hidden Behaviors From Input Validation in Mobile Apps. In S&P 2020

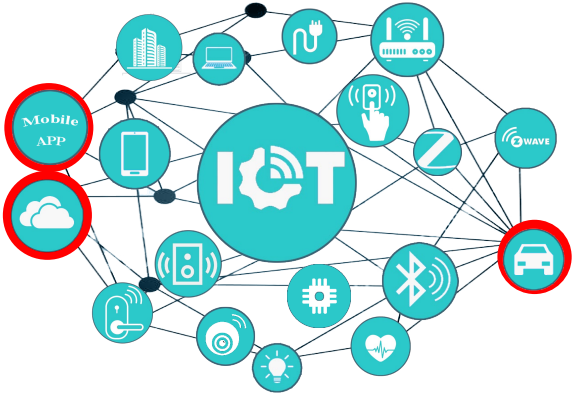


Our Recent Works in IoT Security and Privacy



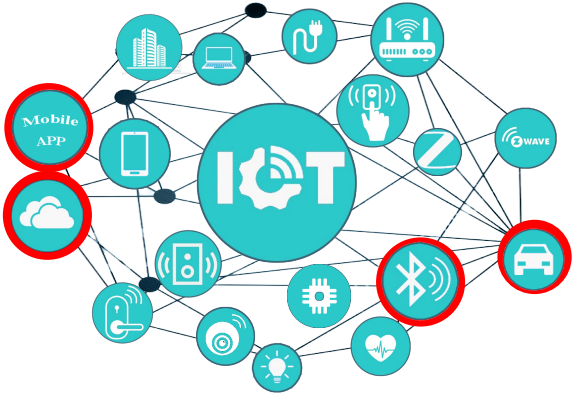
1. Automatic Uncovering of Hidden Behaviors From Input Validation in Mobile Apps. In S&P 2020
2. Why Does Your Data Leak? Uncovering the Data Leakage in Cloud From Mobile Apps. In S&P 2019
3. The Betrayal At Cloud City: An Empirical Analysis Of Cloud-Based Mobile Backends. In USENIX Security 2019

Our Recent Works in IoT Security and Privacy



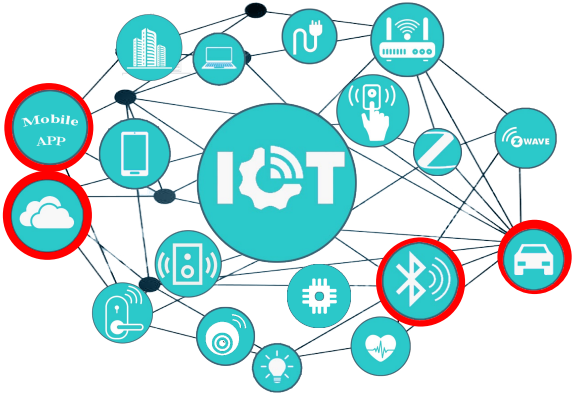
1. Automatic Uncovering of Hidden Behaviors From Input Validation in Mobile Apps. In S&P 2020
2. Why Does Your Data Leak? Uncovering the Data Leakage in Cloud From Mobile Apps. In S&P 2019
3. The Betrayal At Cloud City: An Empirical Analysis Of Cloud-Based Mobile Backends. In USENIX Security 2019
4. Plug-N-Pwned: Comprehensive Vulnerability Analysis of OBD-II Dongles as A New Over-the-Air Attack Surface in Automotive IoT. In USENIX Security 2020
5. Automated Cross-Platform Reverse Engineering of CAN Bus Commands from Mobile Apps. In NDSS 2020

Our Recent Works in IoT Security and Privacy



1. Automatic Uncovering of Hidden Behaviors From Input Validation in Mobile Apps. In S&P 2020
2. Why Does Your Data Leak? Uncovering the Data Leakage in Cloud From Mobile Apps. In S&P 2019
3. The Betrayal At Cloud City: An Empirical Analysis Of Cloud-Based Mobile Backends. In USENIX Security 2019
4. Plug-N-Pwned: Comprehensive Vulnerability Analysis of OBD-II Dongles as A New Over-the-Air Attack Surface in Automotive IoT. In USENIX Security 2020
5. Automated Cross-Platform Reverse Engineering of CAN Bus Commands from Mobile Apps. In NDSS 2020
6. **BLEScope: Automatic Fingerprinting of Vulnerable BLE IoT Devices with Static UUIDs from Mobile Apps**. In ACM CCS 2019
7. **FirmXRay: Detecting Bluetooth Link Layer Vulnerabilities From Bare-Metal Firmware**. In ACM CCS 2020.
8. **Breaking Secure Pairing of Bluetooth Low Energy in Mobile Devices Using Downgrade Attacks**. In USENIX Security 2020
9. **When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure"**. In ACM CCS 2022
10. **Extrapolating Formal Analysis to Uncover Attacks in Bluetooth Passkey Entry Pairing**. In NDSS 2023

Our Recent Works in IoT Security and Privacy



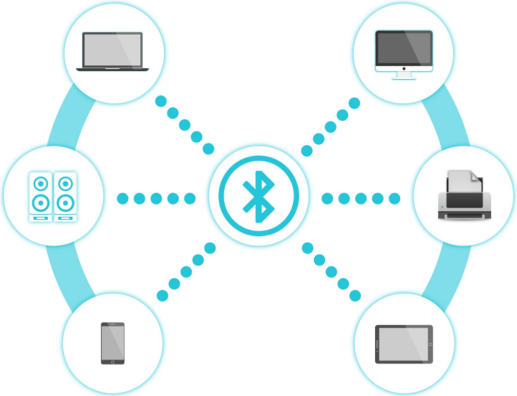
1. ~~Automatic Uncovering of Hidden Behaviors From Input Validation in Mobile Apps. In S&P 2020~~
2. ~~Why Does Your Data Leak? Uncovering the Data Leakage in Cloud From Mobile Apps. In S&P 2019~~
3. ~~The Betrayal At Cloud City: An Empirical Analysis Of Cloud-Based Mobile Backends. In USENIX Security 2019~~
4. ~~Plug-N-Pwned: Comprehensive Vulnerability Analysis of OBD-II Dongles as A New Over-the-Air Attack Surface in Automotive IoT. In USENIX Security 2020~~
5. ~~Automated Cross-Platform Reverse Engineering of CAN Bus Commands from Mobile Apps. In NDSS 2020~~
6. ~~BLEScope: Automatic Fingerprinting of Vulnerable BLE IoT Devices with Static UUIDs from Mobile Apps. In ACM CCS 2019~~
7. ~~FirmXRay: Detecting Bluetooth Link Layer Vulnerabilities From Bare-Metal Firmware. In ACM CCS 2020.~~
8. **Breaking Secure Pairing of Bluetooth Low Energy in Mobile Devices Using Downgrade Attacks**. In USENIX Security 2020
9. **When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure"**. In ACM CCS 2022.
10. ~~Extrapolating Formal Analysis to Uncover Attacks in Bluetooth Passkey Entry Pairing. In NDSS 2023~~

Outline

What is Bluetooth

Bluetooth wireless technology

- Low-cost, low-power
- Short-range radio
- For ad-hoc wireless communication
- For voice and data transmission



What is Bluetooth

Google

Contact tracing

Why Named Bluetooth

Harald "Bluetooth" Gormsson

- King of Denmark 940-981.
- He was also known for his bad **tooth**, which had a very dark **blue-grey** shade.
- He united the Tribes of Denmark.

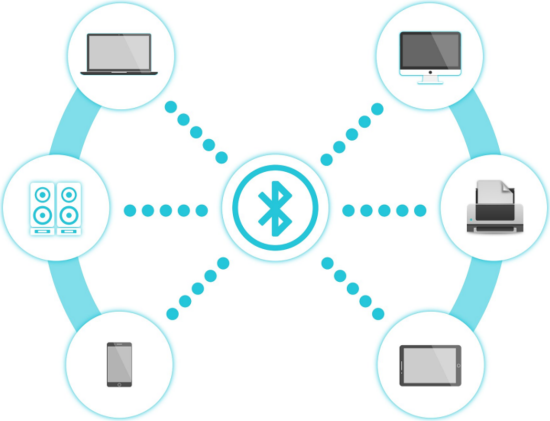
The Bluetooth wireless specification design was named after the king in 1997, based on an analogy **that the technology would unite devices the way Harald Bluetooth united the tribes of Denmark into a single kingdom**.

Why Named Bluetooth

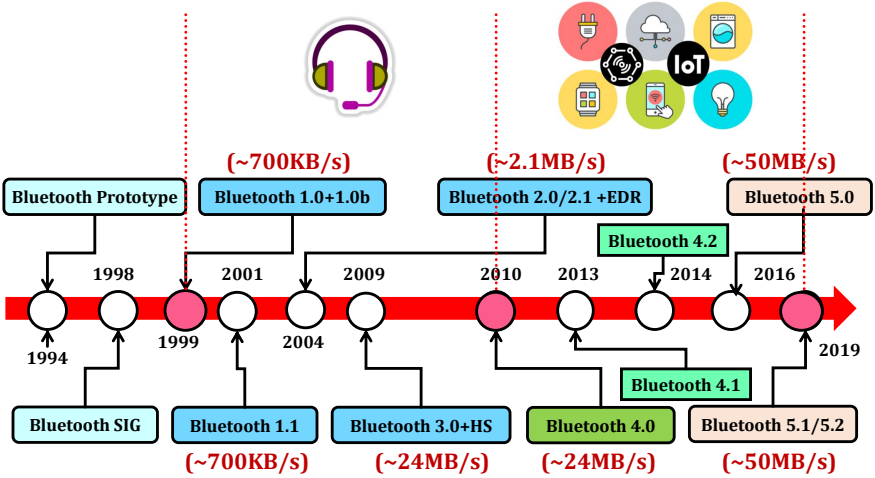
Harald "Bluetooth" Gormsson

- King of Denmark 940-981.
- He was also known for his bad **tooth**, which had a very dark **blue-grey** shade.
- He united the Tribes of Denmark.

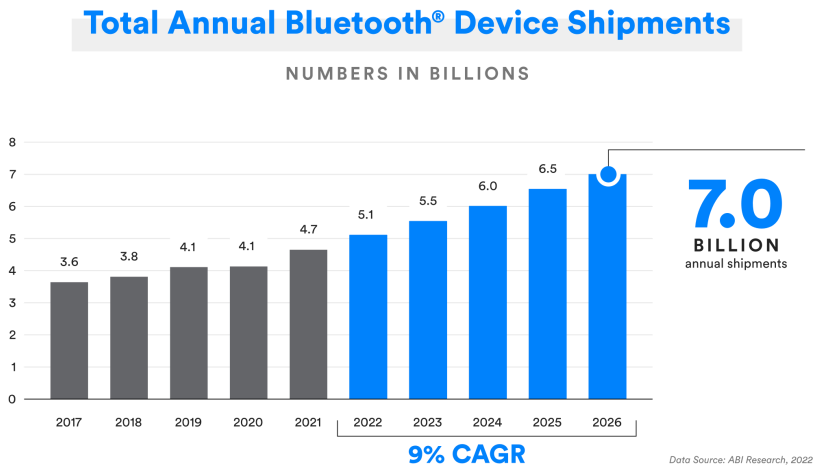
The Bluetooth wireless specification design was named after the king in 1997, based on an analogy **that the technology would unite devices the way Harald Bluetooth united the tribes of Denmark into a single kingdom**.



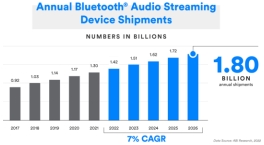
History of Bluetooth



Total Annual Bluetooth Device Shipments

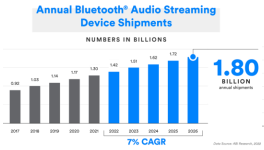


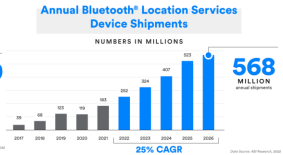
Total Annual Bluetooth Device Shipments



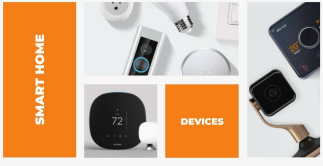
Total Annual Bluetooth Device Shipments

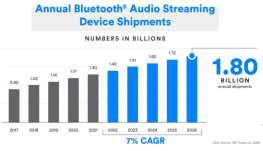
· AirTag ·

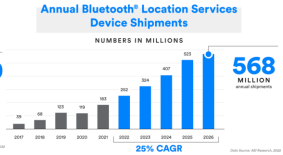


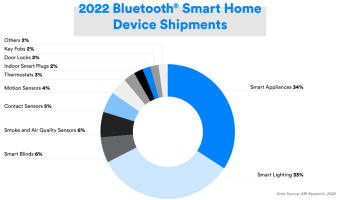


Total Annual Bluetooth Device Shipments

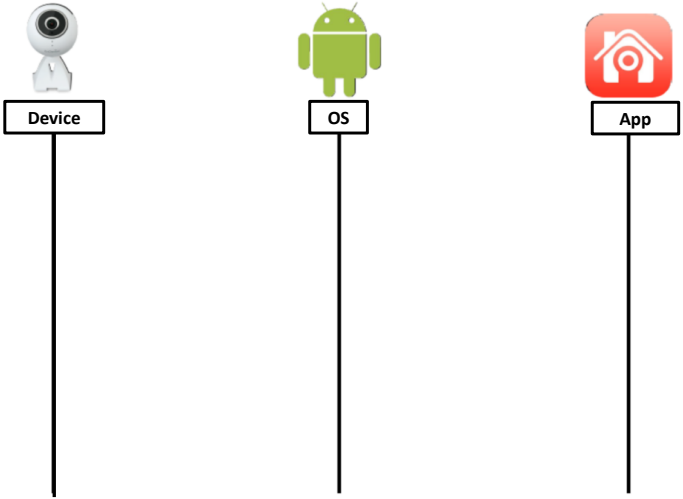




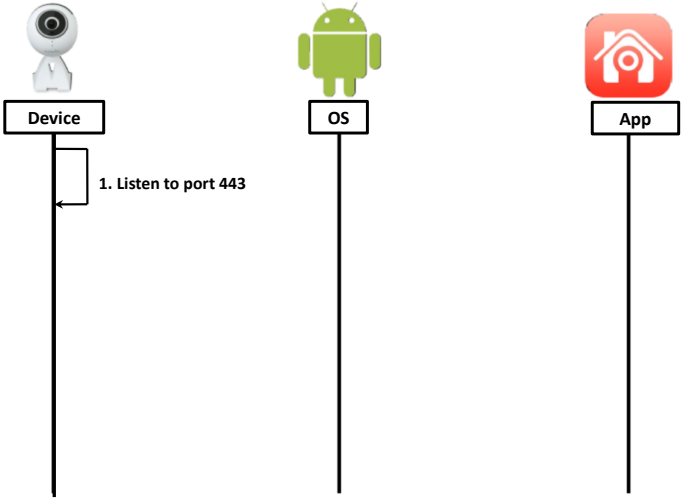




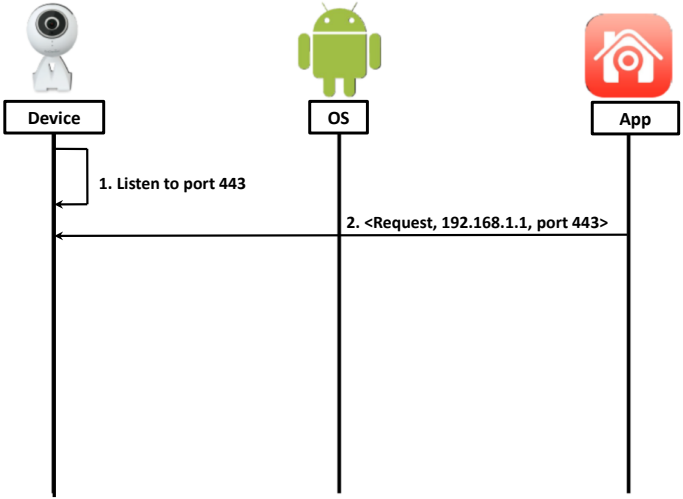
The General Workflow of Device Communication in TCP/IP Setting



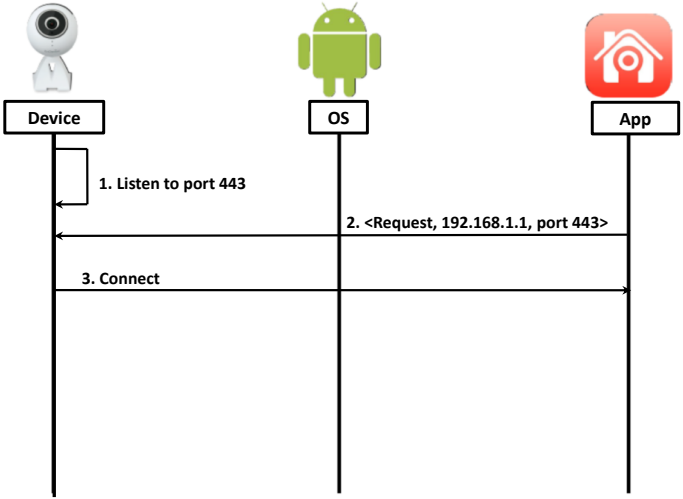
The General Workflow of Device Communication in TCP/IP Setting



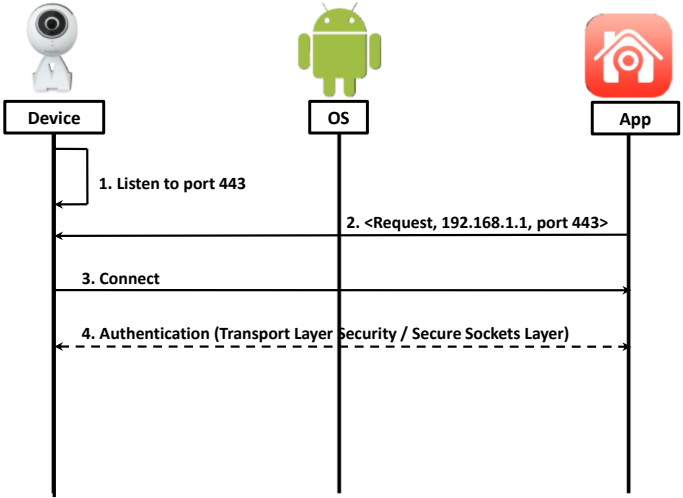
The General Workflow of Device Communication in TCP/IP Setting



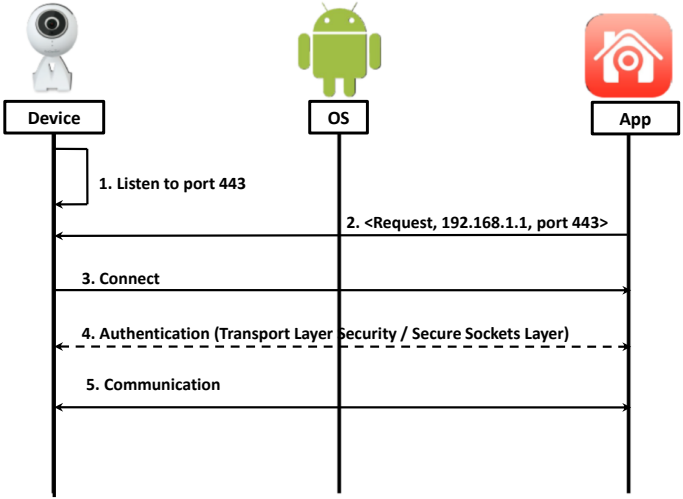
The General Workflow of Device Communication in TCP/IP Setting



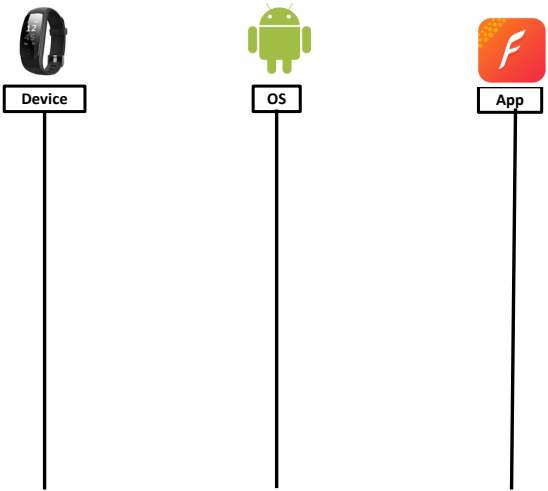
The General Workflow of Device Communication in TCP/IP Setting



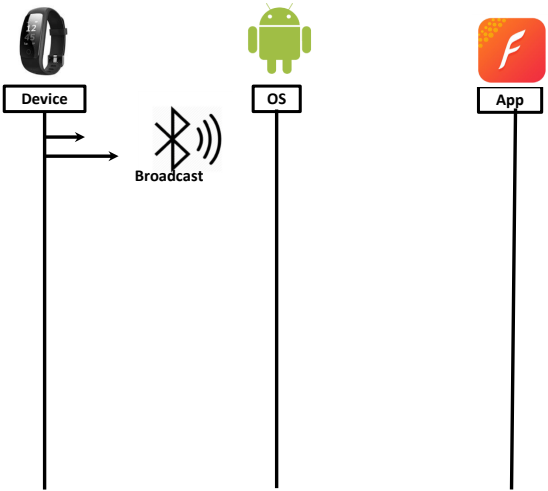
The General Workflow of Device Communication in TCP/IP Setting



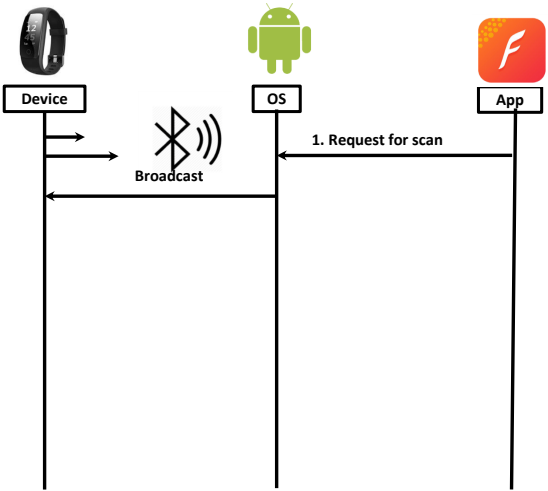
The General Workflow of Device Communication in TCP/IP Setting



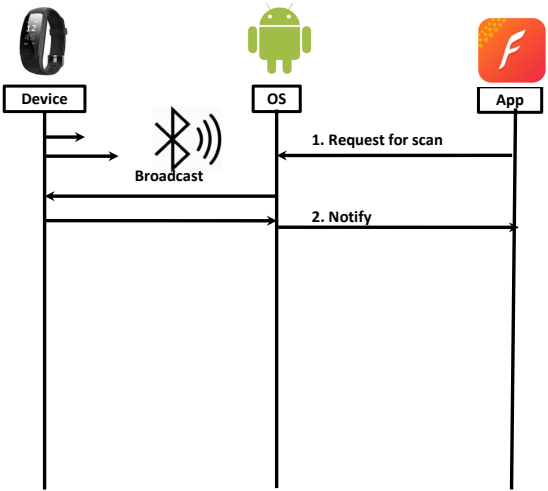
The General Workflow of BLE IoT Devices and Companion Apps



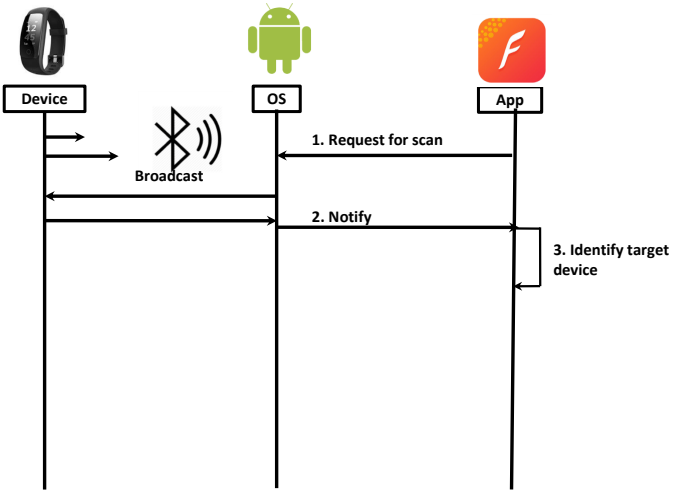
The General Workflow of BLE IoT Devices and Companion Apps



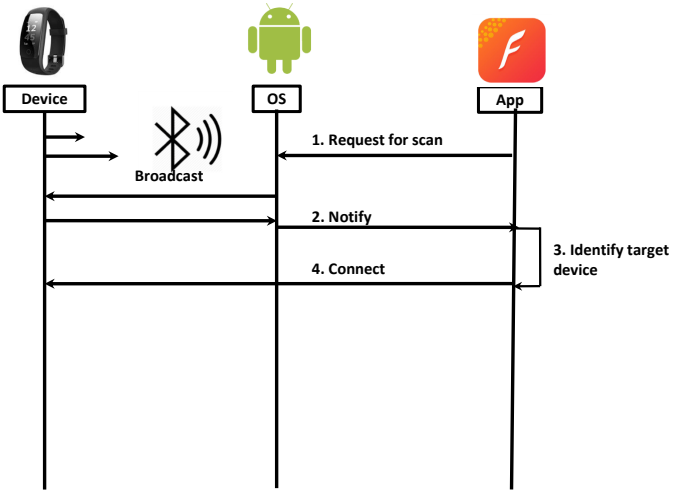
The General Workflow of BLE IoT Devices and Companion Apps



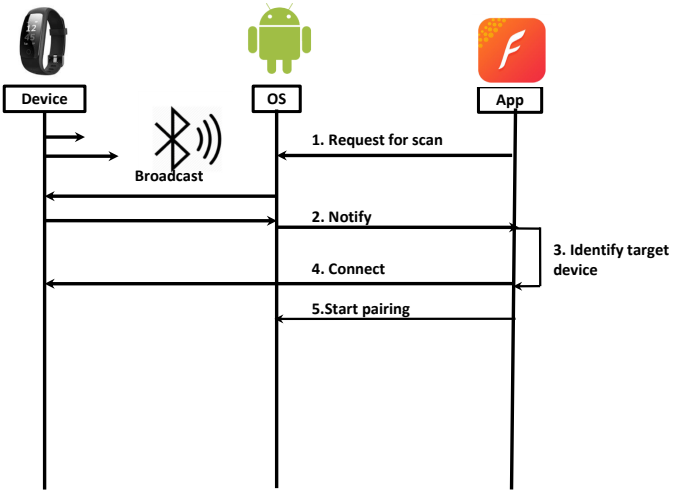
The General Workflow of BLE IoT Devices and Companion Apps



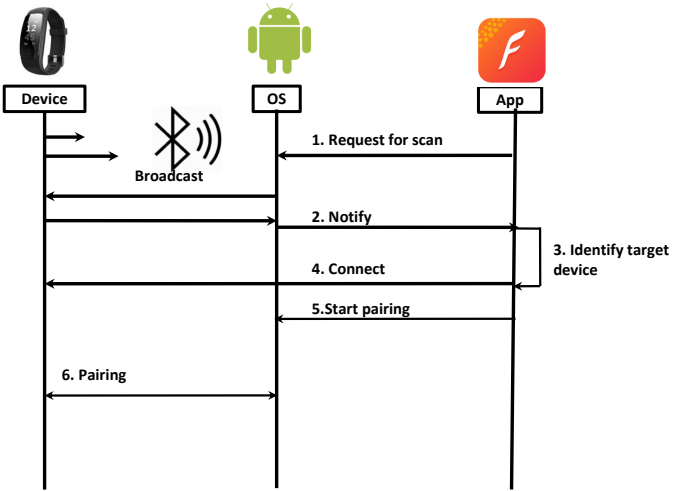
The General Workflow of BLE IoT Devices and Companion Apps



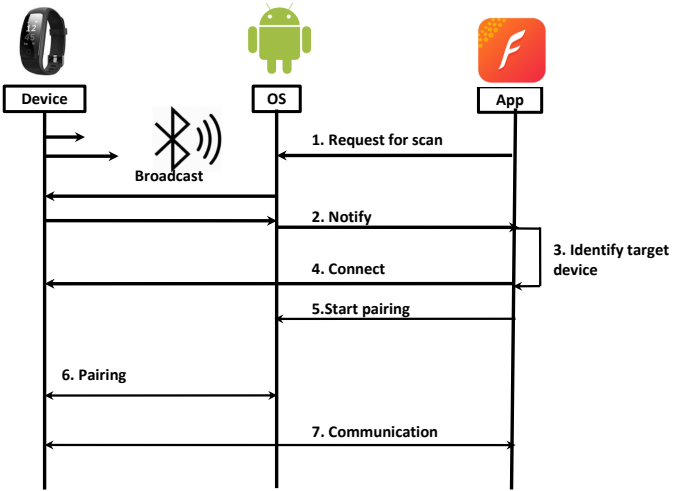
The General Workflow of BLE IoT Devices and Companion Apps



The General Workflow of BLE IoT Devices and Companion Apps

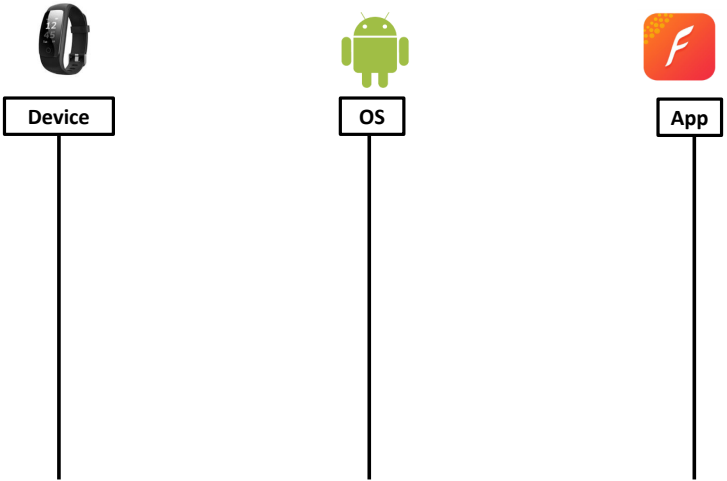


The General Workflow of BLE IoT Devices and Companion Apps

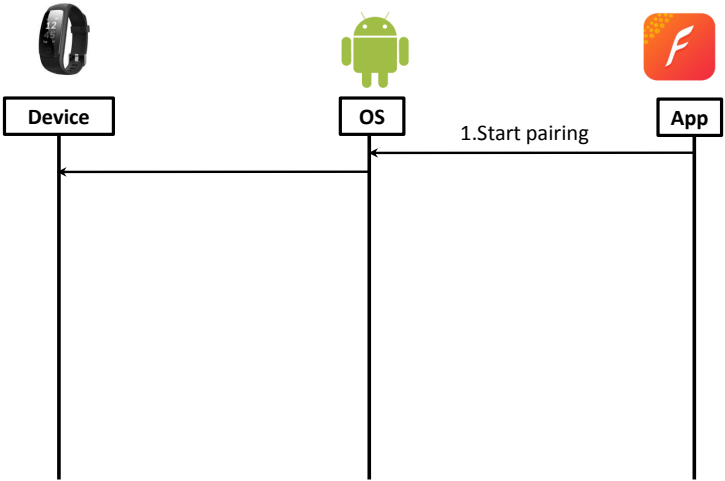


Outline

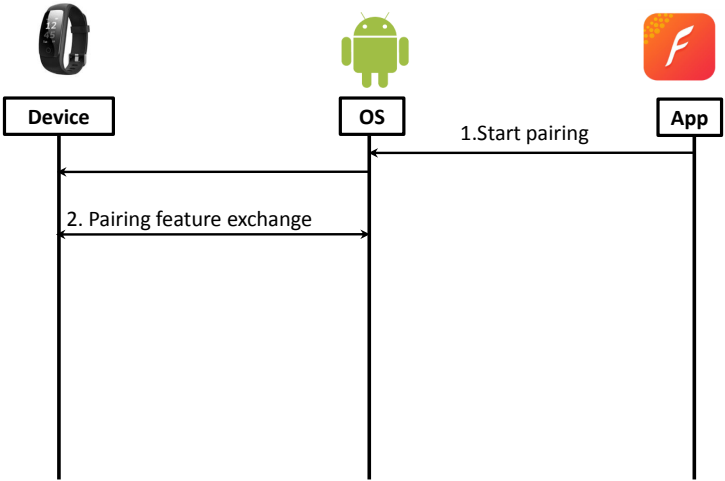
Pairing Workflow



Pairing Workflow



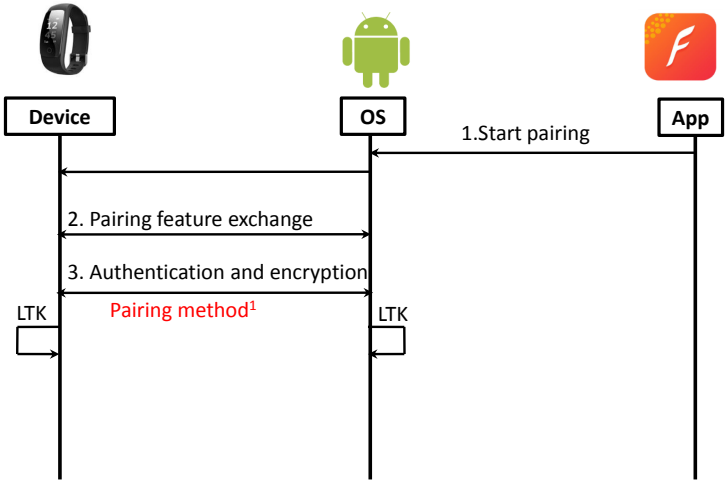
Pairing Workflow



I/O Features

- Keypad
- Screen
- Out of band Channel

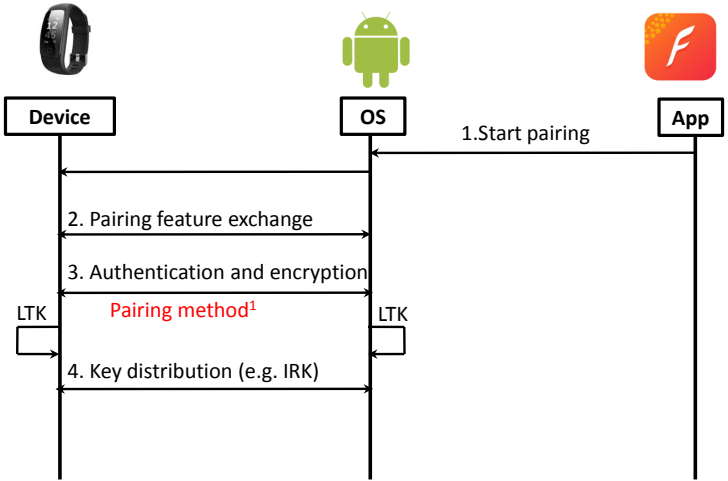
Pairing Workflow



Pairing Methods

- Just Works
- Passkey Entry
- Out of band
- Numeric Comparison

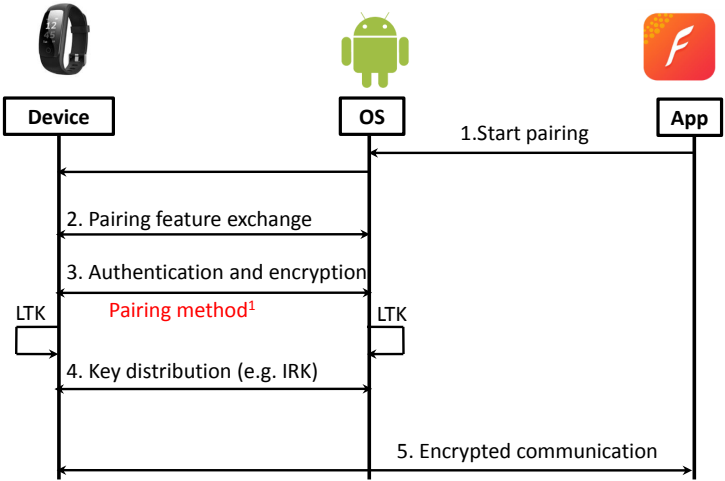
Pairing Workflow



Pairing Methods

- Just Works
- Passkey Entry
- Out of band
- Numeric Comparison

Pairing Workflow



Pairing Methods

- Just Works
- Passkey Entry
- Out of band
- Numeric Comparison

Workflow of Pairing: Elliptic Curve Diffie–Hellman (**ECDH**) Key Exchange

1. Alice generates a random ECC key pair: $\{Pri_A, PK_A = Pri_A * G\}$

Workflow of Pairing: Elliptic Curve Diffie–Hellman (**ECDH**) Key Exchange

1. Alice generates a random ECC key pair: $\{Pri_A, PK_A = Pri_A * \mathsf{G}\}$
2. Bob generates a random ECC key pair: $\{Pri_B, PK_B = Pri_B * \mathsf{G}\}$

Workflow of Pairing: Elliptic Curve Diffie–Hellman (**ECDH**) Key Exchange

1. Alice generates a random ECC key pair: $\{Pri_A, PK_A = Pri_A * \mathsf{G}\}$
2. Bob generates a random ECC key pair: $\{Pri_B, PK_B = Pri_B * \mathsf{G}\}$
3. Alice and Bob exchanges PK_A and PK_B

Workflow of Pairing: Elliptic Curve Diffie–Hellman (**ECDH**) Key Exchange

1. Alice generates a random ECC key pair: $\{Pri_A, PK_A = Pri_A * \mathrm{G}\}$
2. Bob generates a random ECC key pair: $\{Pri_B, PK_B = Pri_B * \mathrm{G}\}$
3. Alice and Bob exchanges PK_A and PK_B
4. Alice calculates sharedKey: $K_A = Pri_A * PK_B$

Workflow of Pairing: Elliptic Curve Diffie–Hellman (**ECDH**) Key Exchange

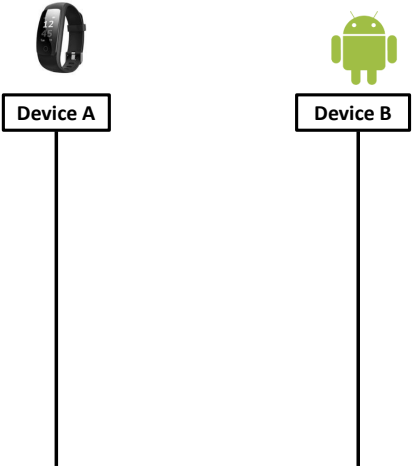
1. Alice generates a random ECC key pair: $\{Pri_A, PK_A = Pri_A$ * G$\}$
2. Bob generates a random ECC key pair: $\{Pri_B, PK_B = Pri_B$ * G$\}$
3. Alice and Bob exchanges PK_A and PK_B
4. Alice calculates sharedKey: $K_A = Pri_A * PK_B$
5. Bob calculates sharedKey: $K_B = Pri_B * PK_A$

Workflow of Pairing: Elliptic Curve Diffie–Hellman (**ECDH**) Key Exchange

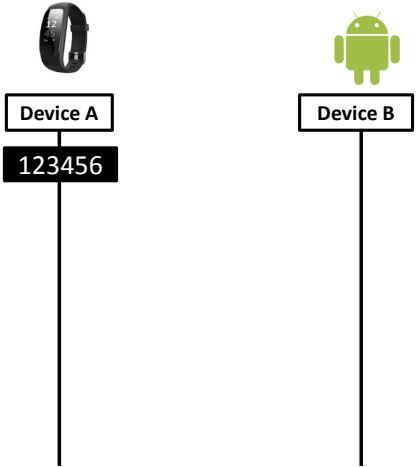
1. Alice generates a random ECC key pair: $\{Pri_A, PK_A = Pri_A$ * G$\}$
2. Bob generates a random ECC key pair: $\{Pri_B, PK_B = Pri_B$ * G$\}$
3. Alice and Bob exchanges PK_A and PK_B
4. Alice calculates sharedKey: $K_A = Pri_A * PK_B$
5. Bob calculates sharedKey: $K_B = Pri_B * PK_A$

$$Pri_A * (Pri_B * G) = Pri_B * (Pri_A * G)$$

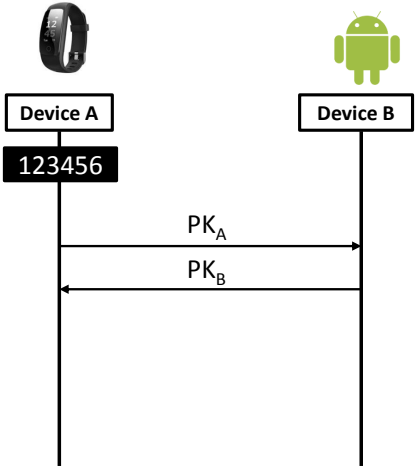
Workflow of Passkey Entry



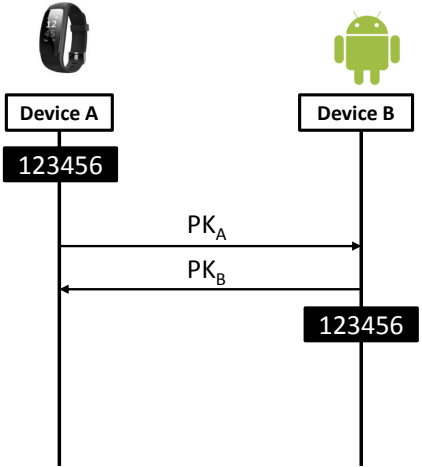
Workflow of Passkey Entry



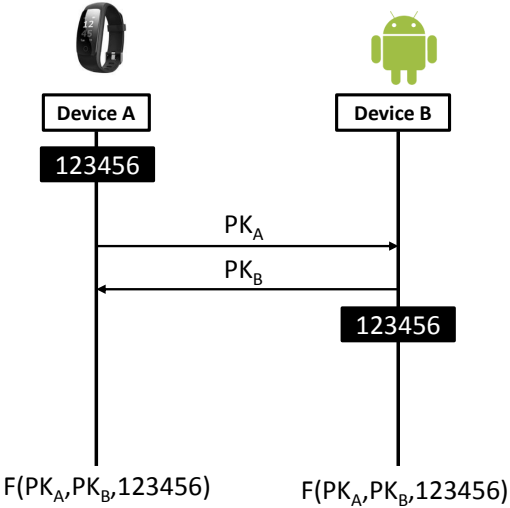
Workflow of Passkey Entry



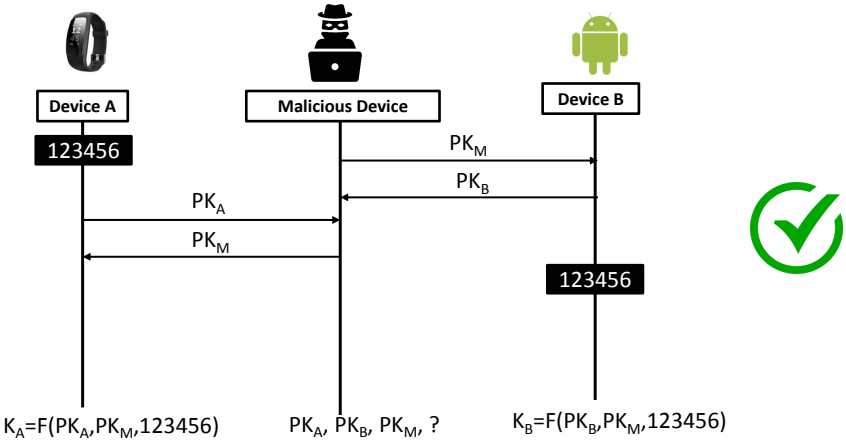
Workflow of Passkey Entry



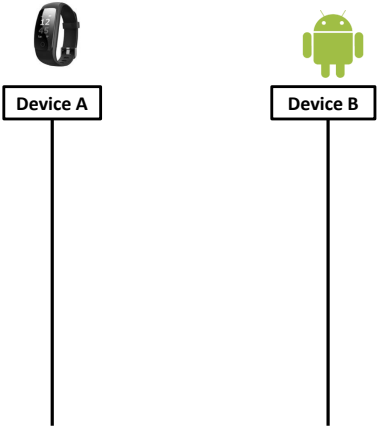
Workflow of Passkey Entry

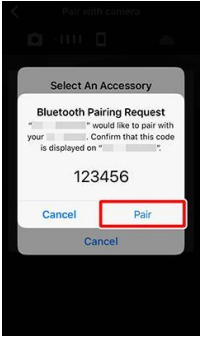


Workflow of Passkey Entry

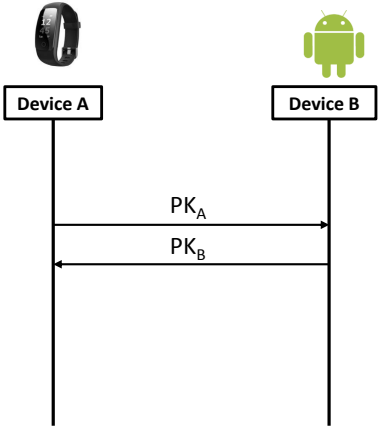


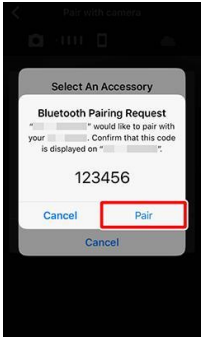
Workflow of Numeric Comparison

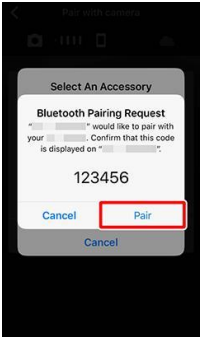




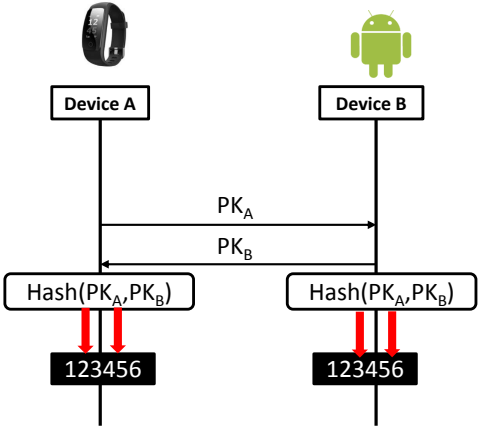
Workflow of Numeric Comparison

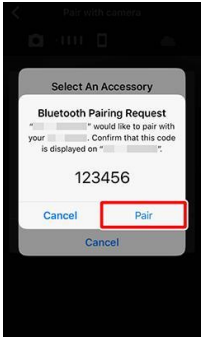


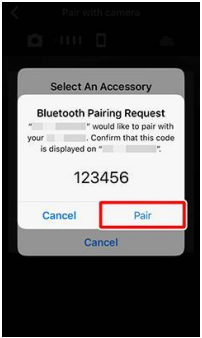




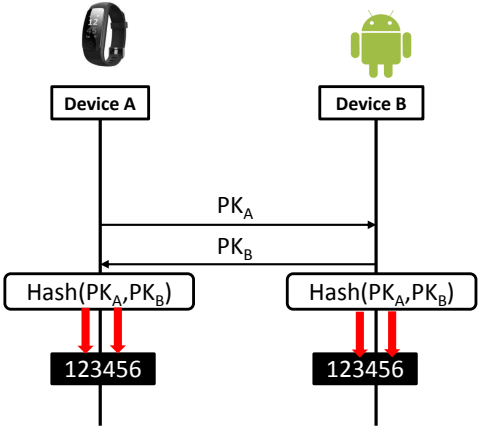
Workflow of Numeric Comparison

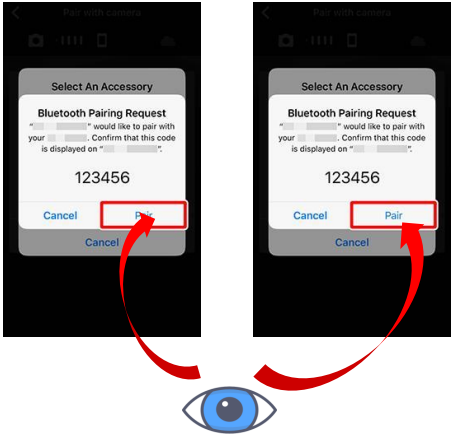




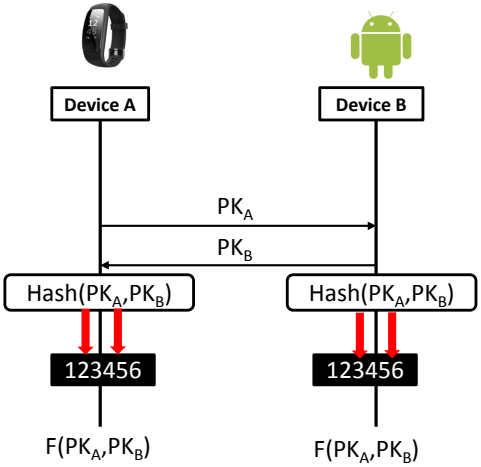


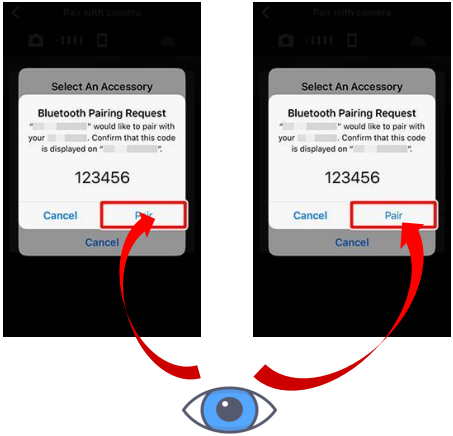
Workflow of Numeric Comparison



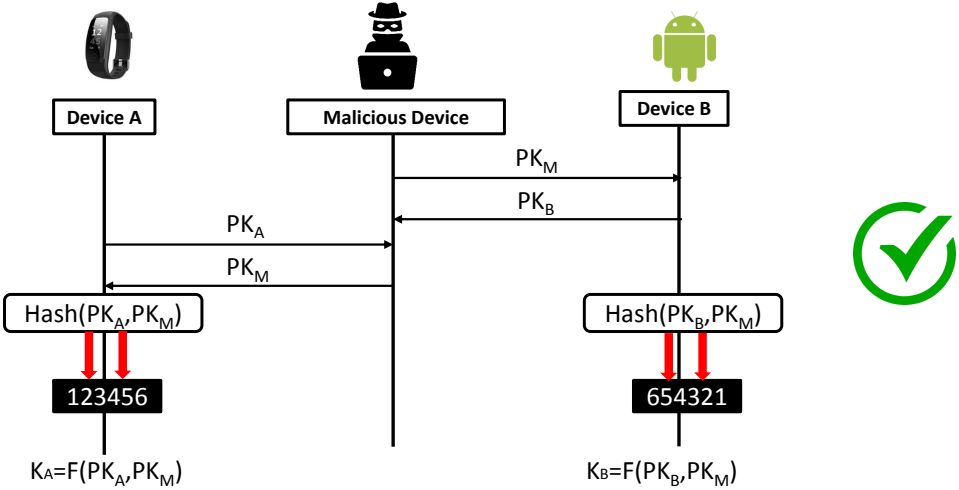


Workflow of Numeric Comparison

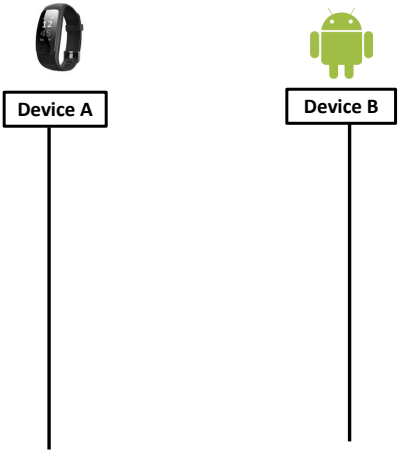




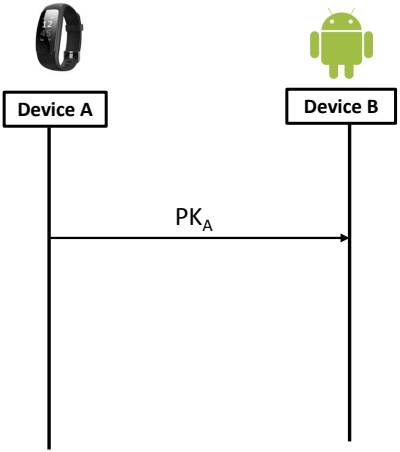
Workflow of Numeric Comparison



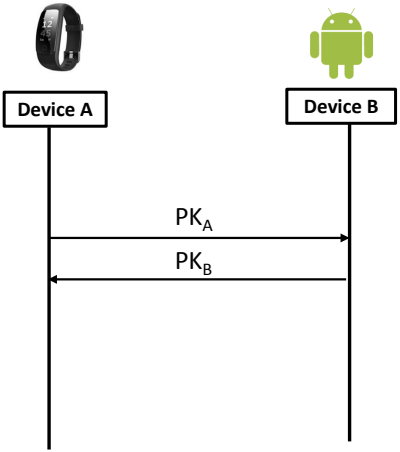
Workflow of Out of Band



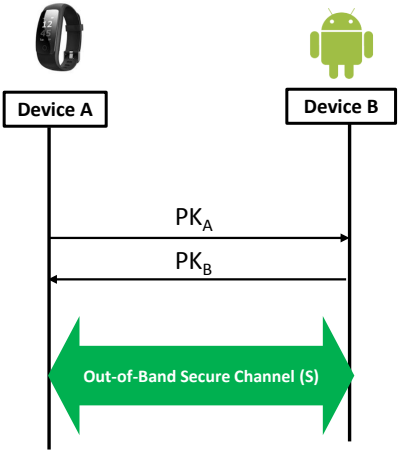
Workflow of Out of Band



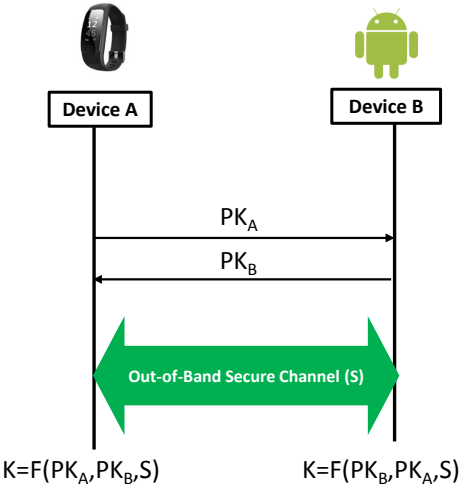
Workflow of Out of Band



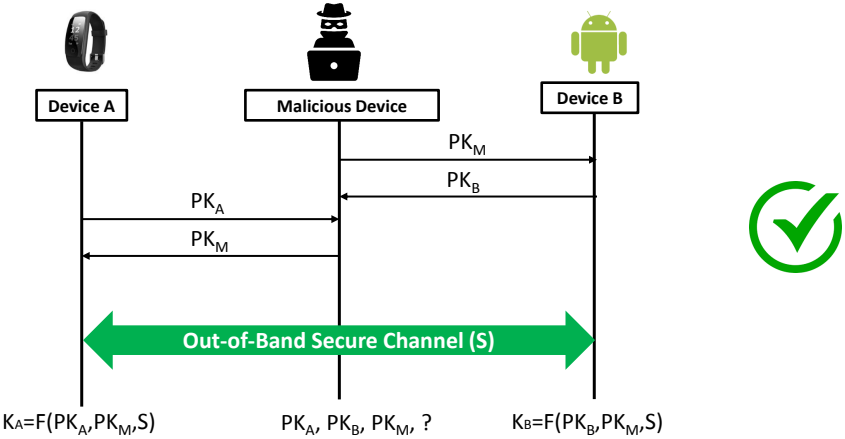
Workflow of Out of Band



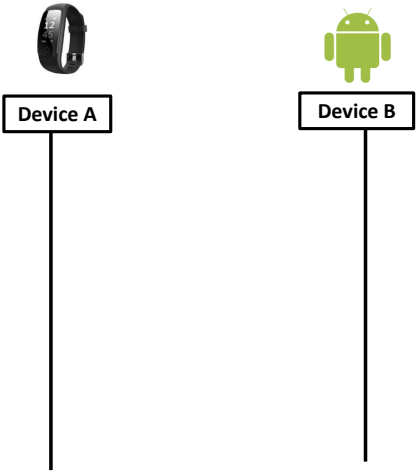
Workflow of Out of Band



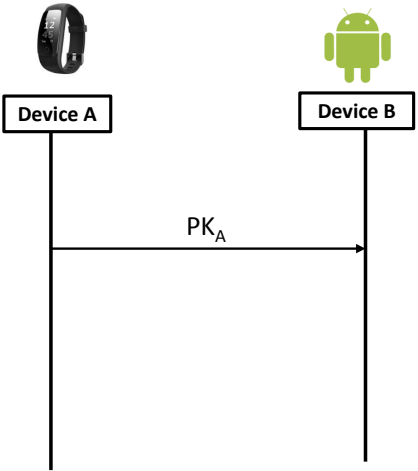
Workflow of Out of Band



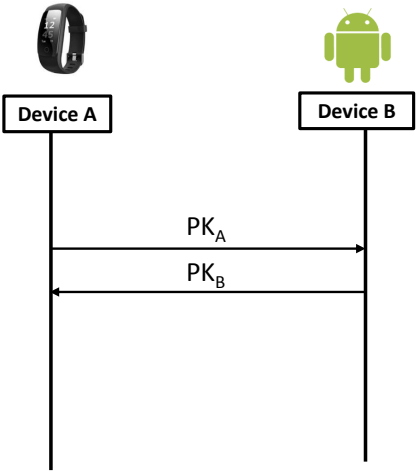
Workflow of Justworks



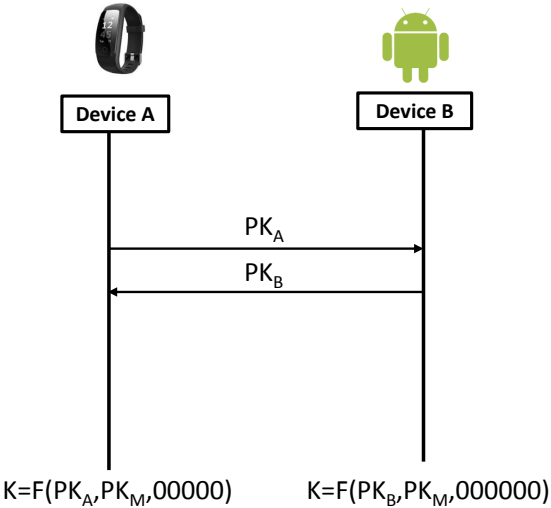
Workflow of Justworks



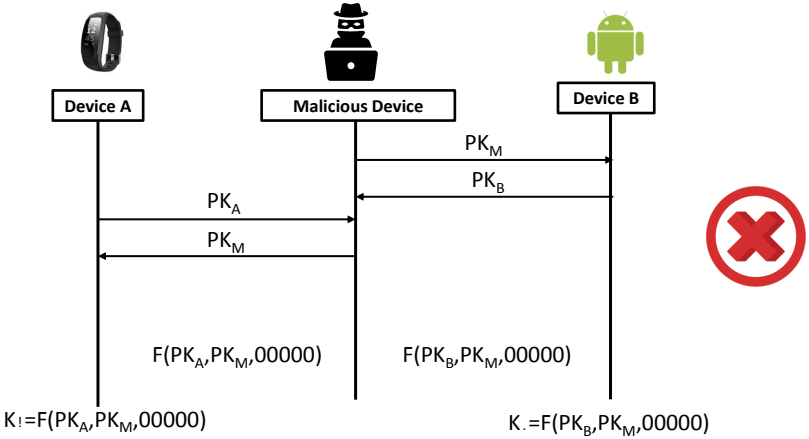
Workflow of Justworks



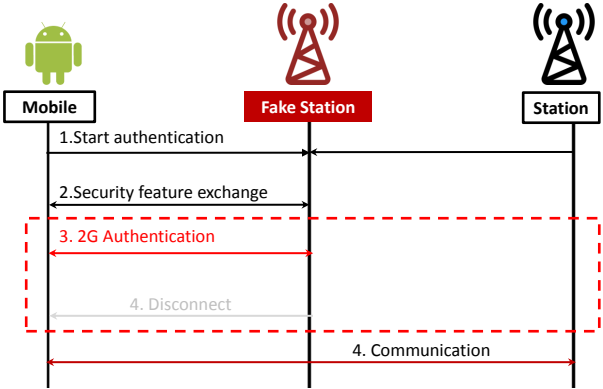
Workflow of Justworks



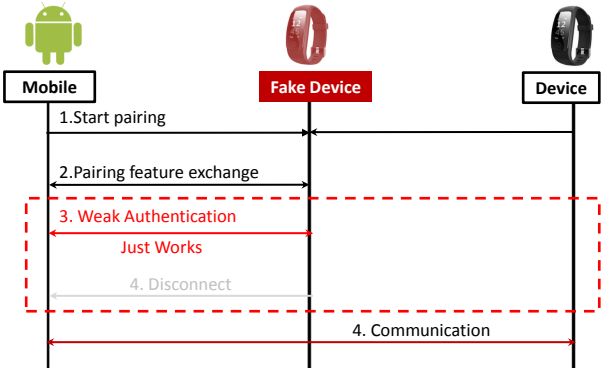
Workflow of Justworks



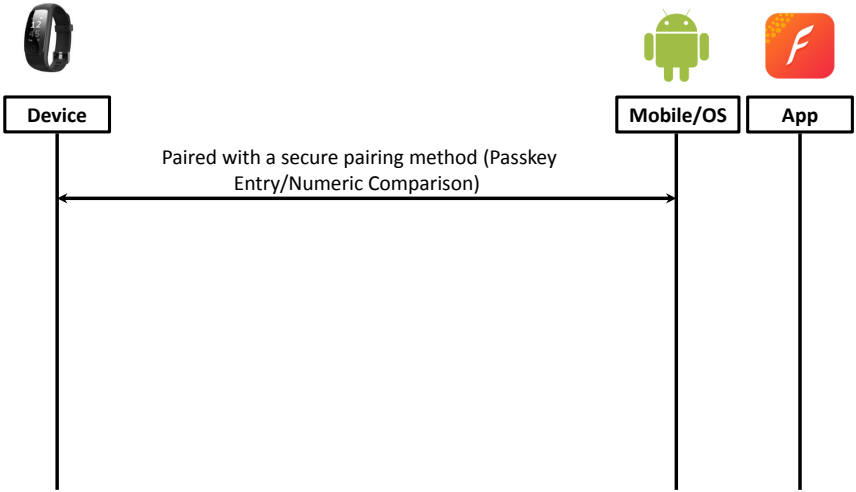
Our Downgrade Attacks against Bluetooth Low Energy



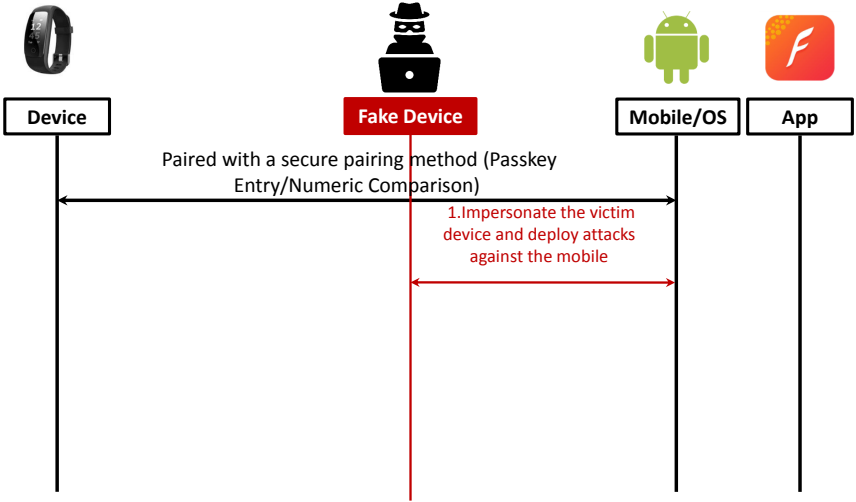
Our Downgrade Attacks against Bluetooth Low Energy



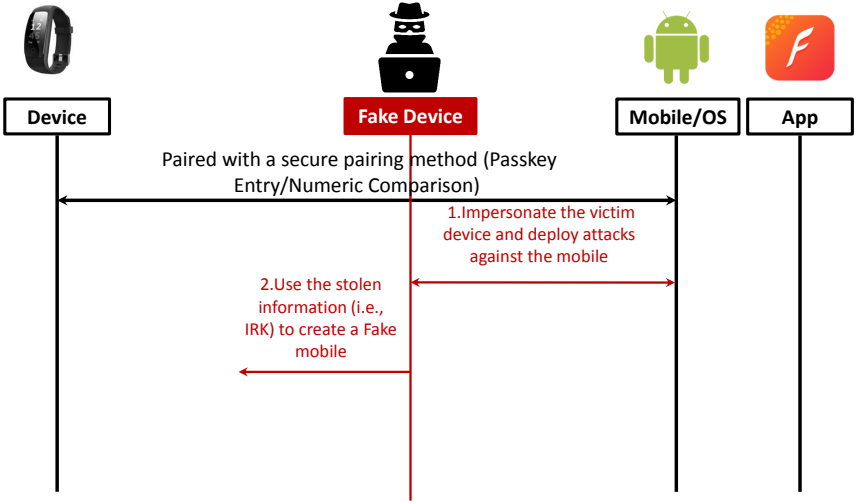
Our Downgrade Attacks against Bluetooth Low Energy



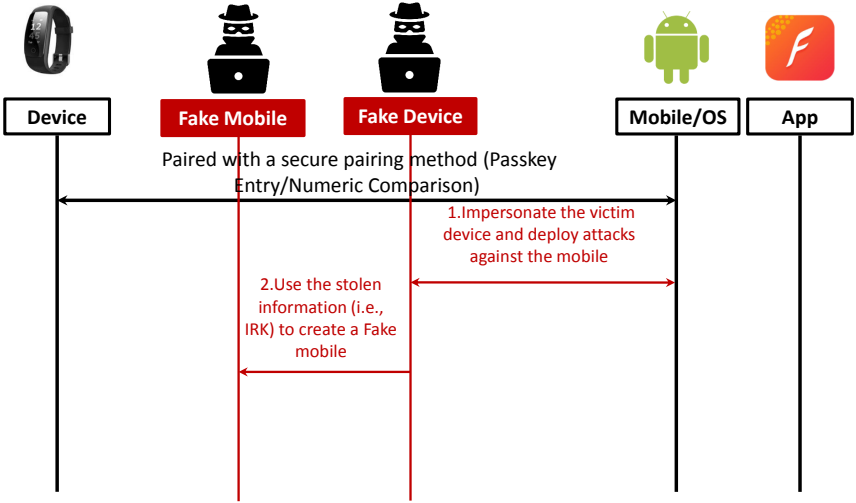
Our Downgrade Attacks against Bluetooth Low Energy



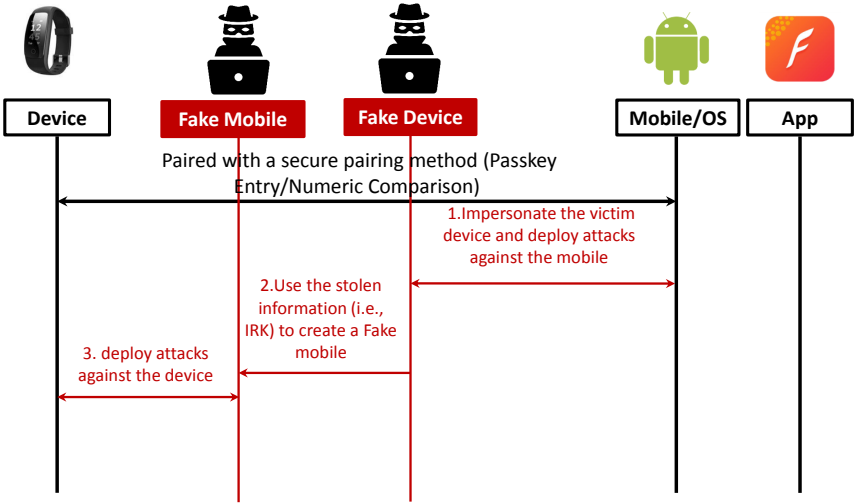
Our Downgrade Attacks against Bluetooth Low Energy



Our Downgrade Attacks against Bluetooth Low Energy

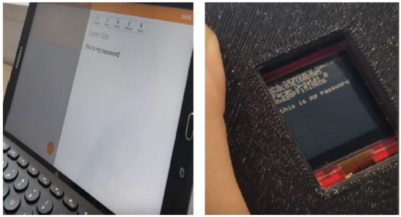


Our Downgrade Attacks against Bluetooth Low Energy



Our Downgrade Attacks against Bluetooth Low Energy

The Tested BLE devices

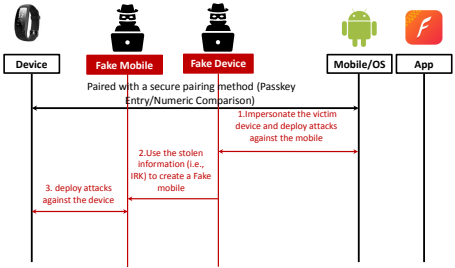


User Attacker

MITM attack against BLE keyboards

CVE-2020-9770

Our Downgrade Attacks against Bluetooth Low Energy



"Breaking Secure Pairing of Bluetooth Low Energy Using Downgrade Attacks", Yue Zhang, Jian Weng, Rajib Dey, Yier Jin, Zhiqiang Lin, and Xinwen Fu. *In Proceedings of the 29th USENIX Security Symposium*, Boston, MA. August 2020

Outline

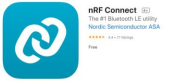
Bluetooth Sniffers

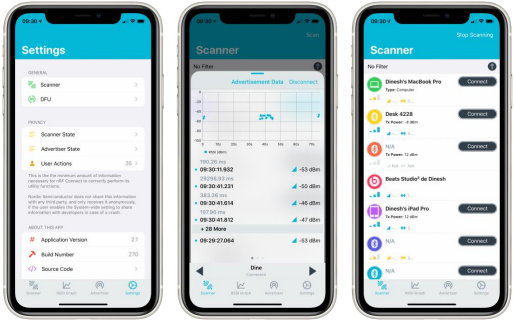
Ubertooth One Sniffer

125 USD

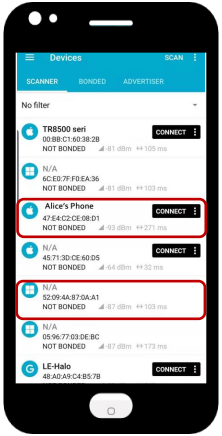
Adafruit LE sniffer

25 USD





Bluetooth Sniffers

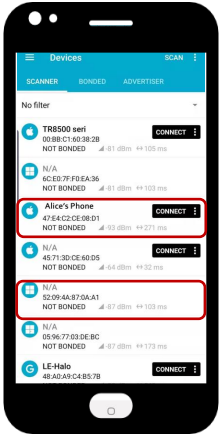


Alice's phone

Bob's phone

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

Bluetooth Sniffers



Alice's phone

Bob's phone

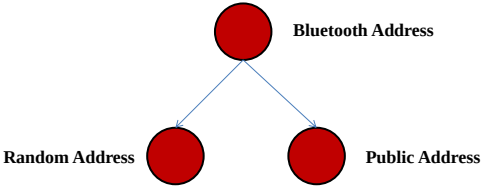
T1: 52:09:4A:87:0A:A1

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

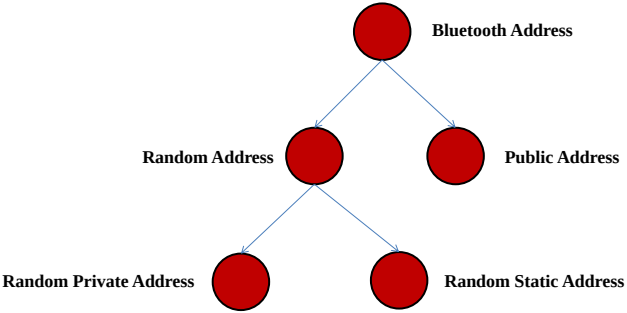
Bluetooth Address Types

Bluetooth Address

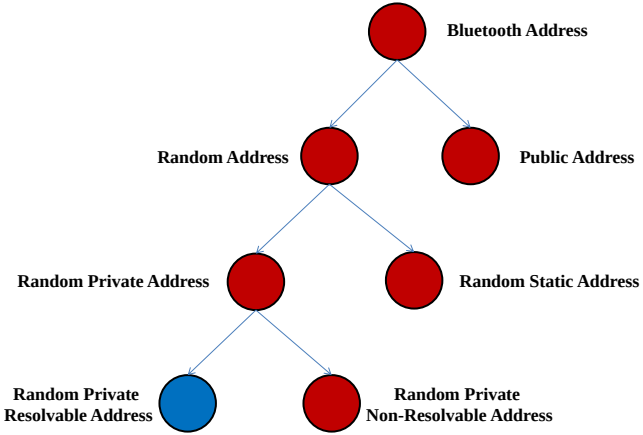
Bluetooth Address Types



Bluetooth Address Types



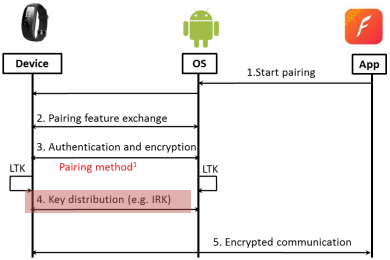
Bluetooth Address Types



How to Avoid Being Tracked: MAC Address Randomization

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)



How to Avoid Being Tracked: MAC Address Randomization

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

How to Avoid Being Tracked: MAC Address Randomization

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

(I) RPA Generation

How to Avoid Being Tracked: MAC Address Randomization

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

(I) RPA Generation

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

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)

How to Avoid Being Tracked: MAC Address Randomization

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

(I) RPA Generation

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

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

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

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)

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)

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

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)

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

$irk_p = irk_c$ ➔ $rpa_p = rpa_c$

Our Discovery I — 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 Discovery I — Allowlist-based Side Channel

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

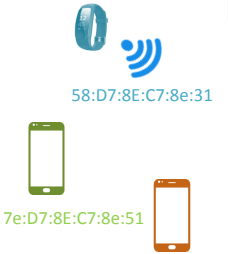
7e:D7:8E:C7:8e:51

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 Discovery I — Allowlist-based Side Channel

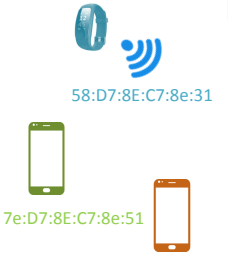
NO.	Time	Source	Destination	TYPE
1	00:00:04	58:D7:8E:C7:8e:31	Broadcast	ADV_IND
2	00:00:08	7e:D7:8E:C7:8e:51	58:D7:8E:C7:8e:31	SCAN_REQ
3	00:00:12	58:D7:8E:C7:8e:31	Broadcast	SCAN_RSP
4	00:00:16	4f:b7:8E:C7:8e:38	58:D7:8E:C7:8e:31	SCAN_REQ
5	00:00:24	58:D7:8E:C7:8e:31	Broadcast	ADV_IND

Our Discovery I — 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

Our Discovery I — Allowlist-based Side Channel



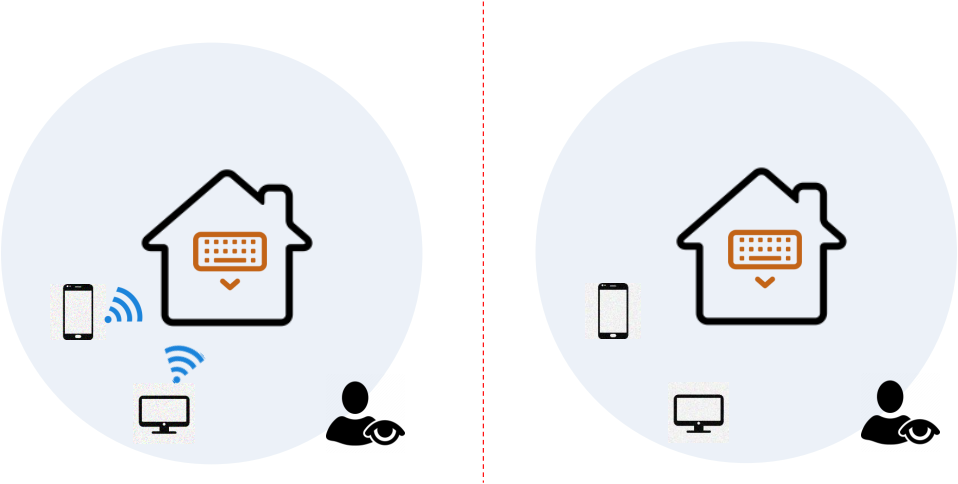
NO.	Time	Source	Destination	TYPE
1	00:00:04	58:D7:8E:C7:8e:31	Broadcast	ADV_IND
2	00:00:08	7e:D7:8E:C7:8e:51	58:D7:8E:C7:8e:31	SCAN_REQ
3	00:00:12	58:D7:8E:C7:8e:31	Broadcast	SCAN_RSP
4	00:00:16	4f:b7:8E:C7:8e:38	58:D7:8E:C7:8e:31	SCAN_REQ
5	00:00:24	58:D7:8E:C7:8e:31	Broadcast	ADV_IND
.....				
200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

Passive Bluetooth Address Tracking (BAT) Attacks

Attack I: Monitoring a Victim's Status

Passive Bluetooth Address Tracking (BAT) Attacks



Attack I: Monitoring a Victim's Status

Passive Bluetooth Address Tracking (BAT) Attacks

Attack I: Monitoring a Victim's Status

Passive Bluetooth Address Tracking (BAT) Attacks

Attack I: Monitoring a Victim's Status

Passive Bluetooth Address Tracking (BAT) Attacks

Attack I: Monitoring a Victim's Status

Our Discovery II — MAC Address Replay

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

Our Discovery II — 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 Discovery II — 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 Discovery II — 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 Discovery II — 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)

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

$irk_p = irk_c$ ➔ $rpa_p = rpa_c$

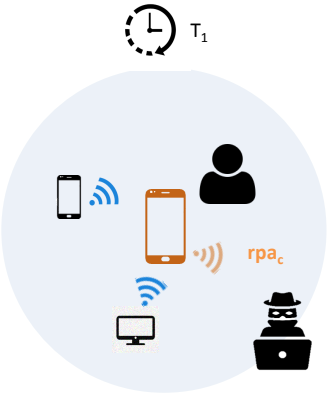
rpa_p

rpa'_p

Active BAT Attacks: Tracking a Victim's Past Trajetory

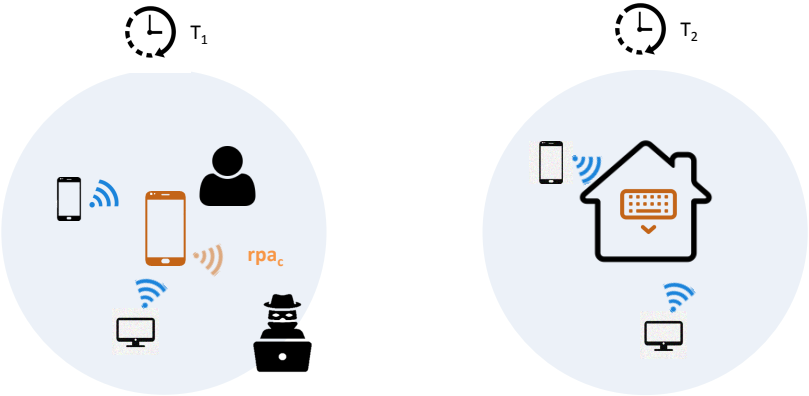
Attack II: Tracking a Victim's Past Trajectory

Active BAT Attacks: Tracking a Victim's Past Trajetory



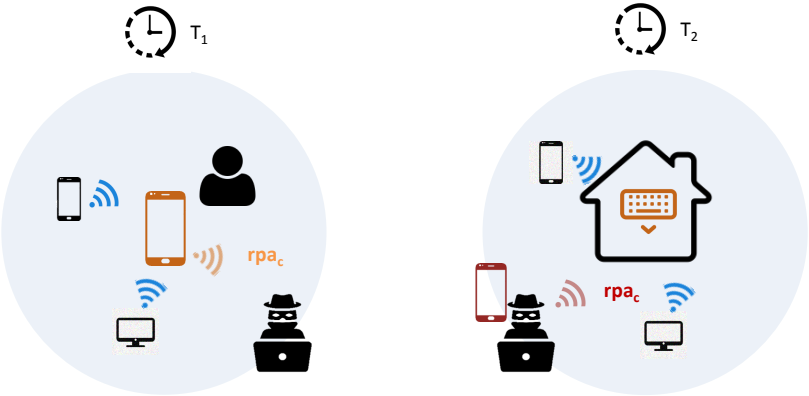
Attack II: Tracking a Victim's Past Trajectory

Active BAT Attacks: Tracking a Victim's Past Trajetory



Attack II: Tracking a Victim's Past Trajectory

Active BAT Attacks: Tracking a Victim's Past Trajetory

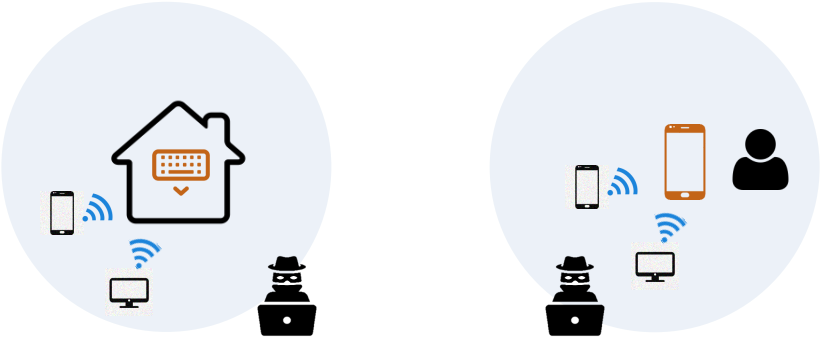


Attack II: Tracking a Victim's Past Trajectory

Active BAT Attacks: Tracking a Victim's Real-time Location

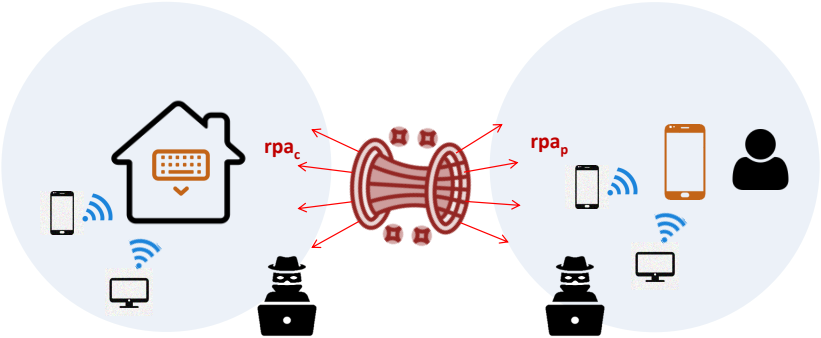
Attack III: Tracking a Victim's Real-time Location w/ Tunneling

Active BAT Attacks: Tracking a Victim's Real-time Location



Attack III: Tracking a Victim's Real-time Location w/ Tunneling

Active BAT Attacks: Tracking a Victim's Real-time Location



Attack III: Tracking a Victim's Real-time Location w/ Tunneling

Active BAT Attacks: Tracking a Victim's Real-time Location

Attack IV: Tracking a Victim's Real-time Location w/o Tunneling

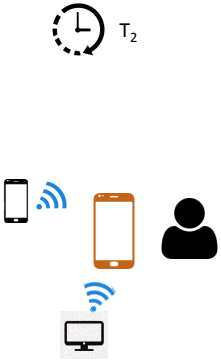
Active BAT Attacks: Tracking a Victim's Real-time Location

Attack IV: Tracking a Victim's Real-time Location w/o Tunneling

Active BAT Attacks: Tracking a Victim's Real-time Location

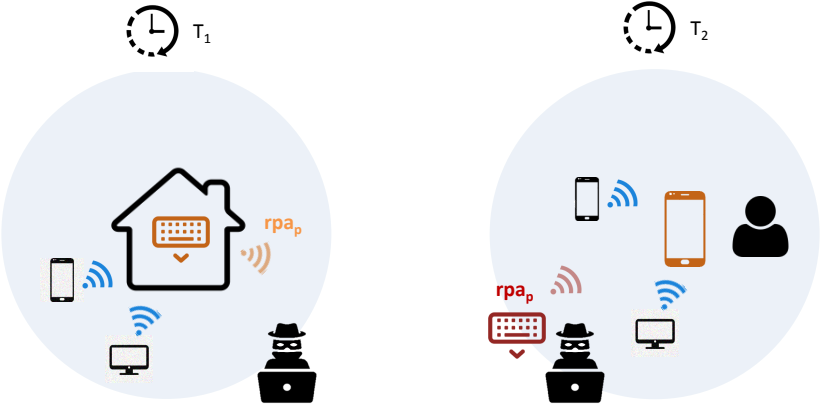
Attack IV: Tracking a Victim's Real-time Location w/o Tunneling

Active BAT Attacks: Tracking a Victim's Real-time Location



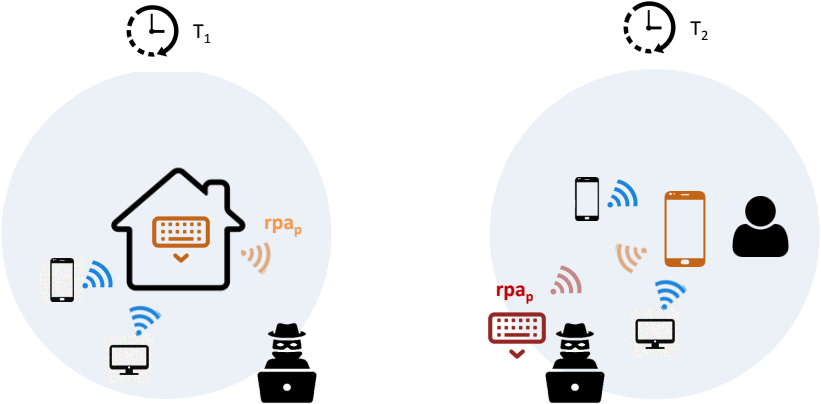
Attack IV: Tracking a Victim's Real-time Location w/o Tunneling

Active BAT Attacks: Tracking a Victim's Real-time Location



Attack IV: Tracking a Victim's Real-time Location w/o Tunneling

Active BAT Attacks: Tracking a Victim's Real-time Location



Attack IV: Tracking a Victim's Real-time Location w/o Tunneling

SABLE — Defense

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

SABLE — Defense

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

(I) RPA Generation

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

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

SABLE — Defense

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

(I) RPA Generation

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

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

SABLE — Defense

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

(I) RPA Generation

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

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)

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

$irk_p = irk_c$ ➔ $rpa_p = rpa_c$

Within a Threshold *Tx*

rpa_p

SABLE — Defense

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

(I) RPA Generation

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

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)

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

$irk_p = irk_c$ ➔ $rpa_p = rpa_c$

Within a Threshold *Tx*

rpa_p

SABLE — Defense

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

(I) RPA Generation

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

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)

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

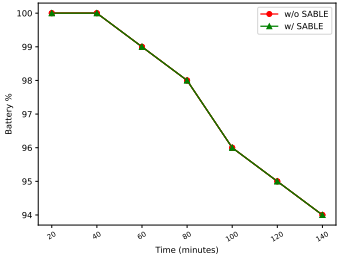
$irk_p = irk_c$ ➔ $rpa_p = rpa_c$

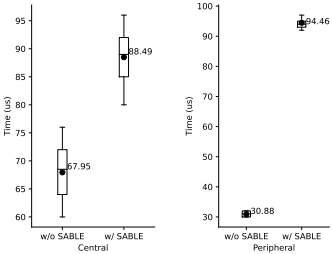
Within a Threshold *Tx*

rpa_p

rpa'_p

Performance of SABLE

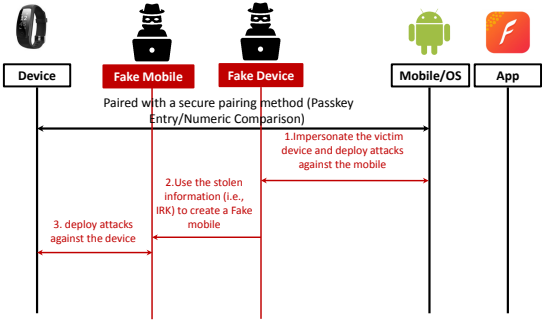




"When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure". Yue Zhang, and Zhiqiang Lin. *In Proceedings of the 29th ACM Conference on Computer and Communications Security (CCS 2022)*. November 2022

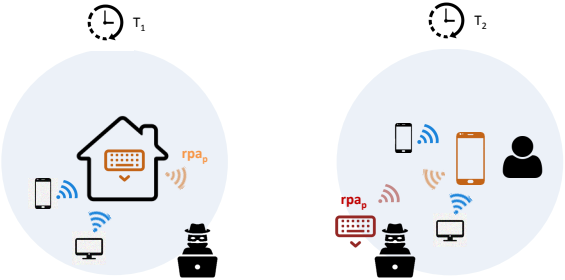
Outline

Lesson Learned (1/3): BLE Communication Can Be Downgraded



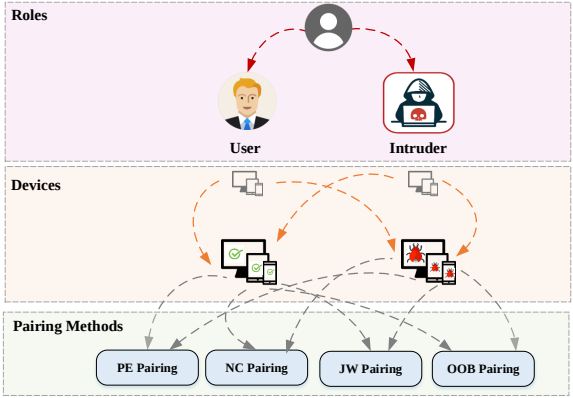
- Bluetooth low energy (BLE) pairing can be **downgraded**
- There are many stages that are not part of the pairing process, but they are, in fact, closely related to pairing security.
- A systematic analysis of the pairing process, including the **error handling** of BLE communication, is needed.

Lesson Learned (2/3): New Features Need Re-examinations



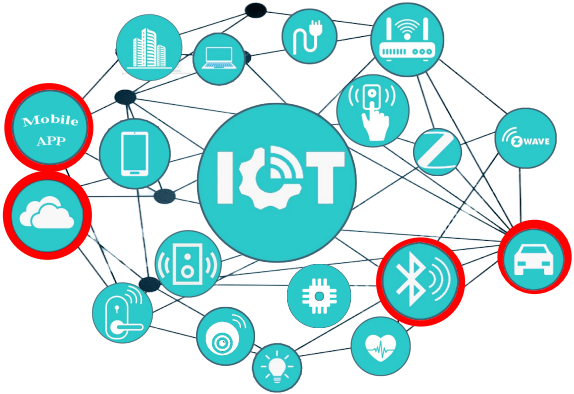
- BLE introduces multiple new features, some of which may **violate existing assumptions**
- Simliar to allowlist, those new features need to be **scrunitized**. For example, Cross-transport key derivation (CTKD); Authorization; The Connection Signature Resolving Key (CSRK).

Lesson Learned (3/3): Formal Method Can Help Improve BLE Security



- The specification (3,000+ pages) is often confusing and inconsistent across chapters.
- The confusion may lead to different vendors implement BLE protocols in quite different ways, for example, for error handling, and IRK use.
- Converting the Bluetooth specification to formal model (e.g., using NLP), and formally verify the entire protocol would help.
- See our NDSS'23 paper.

IoT Security and Privacy



1. ~~Automatic Uncovering of Hidden Behaviors From Input Validation in Mobile Apps. In S&P 2020~~
2. ~~Why Does Your Data Leak? Uncovering the Data Leakage in Cloud From Mobile Apps. In S&P 2019~~
3. ~~The Betrayal At Cloud City: An Empirical Analysis Of Cloud-Based Mobile Backends. In USENIX Security 2019~~
4. ~~Plug-N-Pwned: Comprehensive Vulnerability Analysis of OBD-II Dongles as A New Over-the-Air Attack Surface in Automotive IoT. In USENIX Security 2020~~
5. ~~Automated Cross-Platform Reverse Engineering of CAN Bus Commands from Mobile Apps. In NDSS 2020~~
6. ~~BLEScope: Automatic Fingerprinting of Vulnerable BLE IoT Devices with Static UUIDs from Mobile Apps. In ACM CCS 2019~~
7. ~~FirmXRay: Detecting Bluetooth Link Layer Vulnerabilities From Bare-Metal Firmware. In ACM CCS 2020.~~
8. **Breaking Secure Pairing of Bluetooth Low Energy in Mobile Devices Using Downgrade Attacks**. In USENIX Security 2020
9. **When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure"**. In ACM CCS 2022.
10. ~~Extrapolating Formal Analysis to Uncover Attacks in Bluetooth Passkey Entry Pairing. In NDSS 2023~~

Thank You

Rethinking the Security and Privacy of Bluetooth Low Energy

Dr. Zhiqiang Lin
Distinguished Professor of Engineering
zlin@cse.ohio-state.edu

10/14/2022