Rethinking the Security and Privacy of Bluetooth Low Energy

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

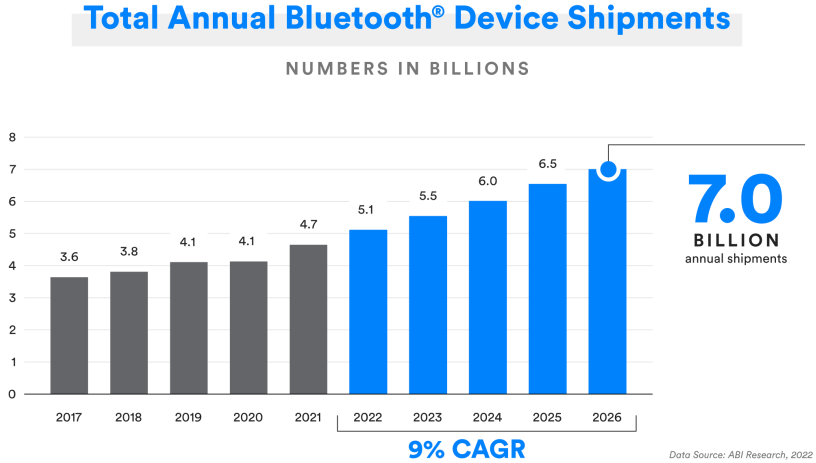
11/28/2022

Outline

Outline

What is Bluetooth

What is Bluetooth



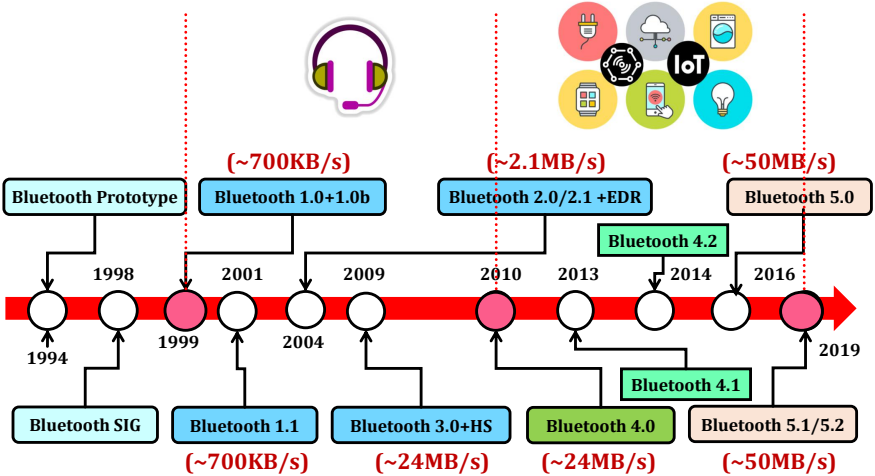
Why Named Bluetooth

Harald "Bluetooth" Gormsson

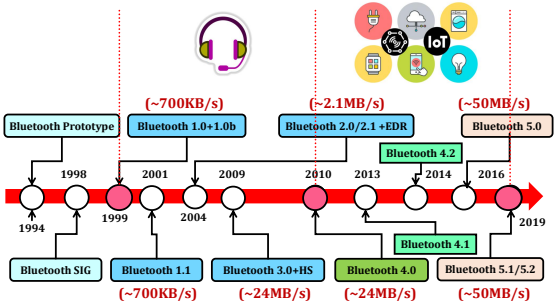
- King of Denmark 940-981.
- He was also known for his bad **tooth**, which had a very dark **blue-grey** shade.
- He united the Tribes of Denmark.

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**.

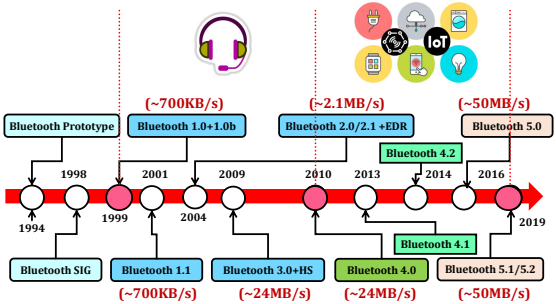
History of Bluetooth



Our Recent Works on Bluetooth Security and Privacy

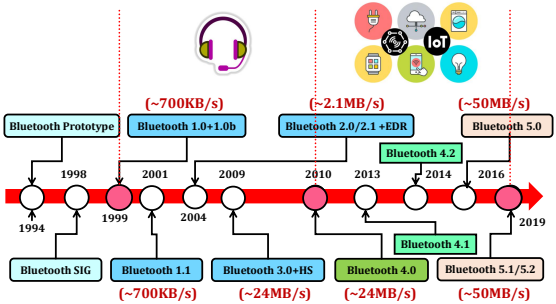


Our Recent Works on Bluetooth Security and Privacy



1. **BLEScope: Automatic Fingerprinting of Vulnerable BLE IoT Devices with Static UUIDs from Mobile Apps**. In ACM CCS 2019
2. **FirmXRay: Detecting Bluetooth Link Layer Vulnerabilities From Bare-Metal Firmware**. In ACM CCS 2020.
3. **Breaking Secure Pairing of Bluetooth Low Energy in Mobile Devices Using Downgrade Attacks**. In USENIX Security 2020
4. **When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure"**. In ACM CCS 2022 (Best paper award honorable mention)
5. **Extrapolating Formal Analysis to Uncover Attacks in Bluetooth Passkey Entry Pairing**. In NDSS 2023

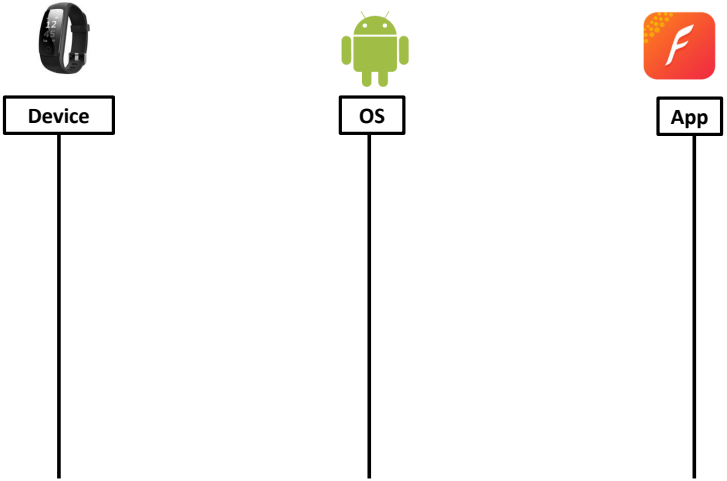
Our Recent Works on Bluetooth Security and Privacy



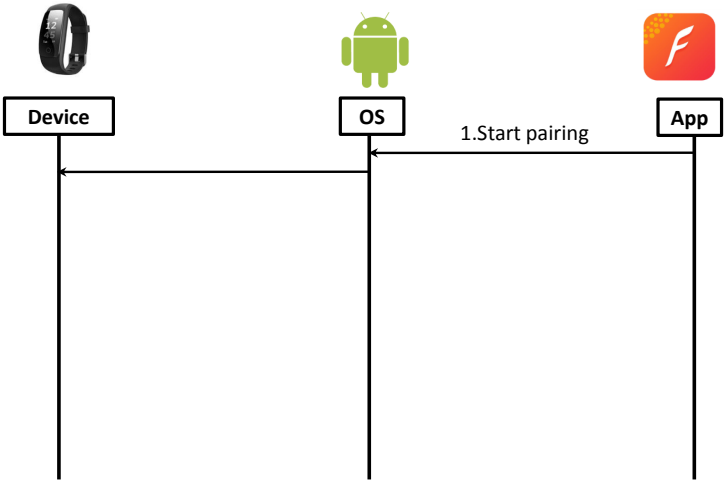
1. ~~BLEScope: Automatic Fingerprinting of Vulnerable BLE IoT Devices with Static UUIDs from Mobile Apps. In ACM CCS 2019~~
2. ~~FirmXRay: Detecting Bluetooth Link Layer Vulnerabilities From Bare-Metal Firmware. In ACM CCS 2020.~~
3. **Breaking Secure Pairing of Bluetooth Low Energy in Mobile Devices Using Downgrade Attacks**. In USENIX Security 2020
4. **When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure"**. In ACM CCS 2022 (Best paper award honorable mention)
5. ~~Extrapolating Formal Analysis to Uncover Attacks in Bluetooth Passkey Entry Pairing. In NDSS 2023~~

Outline

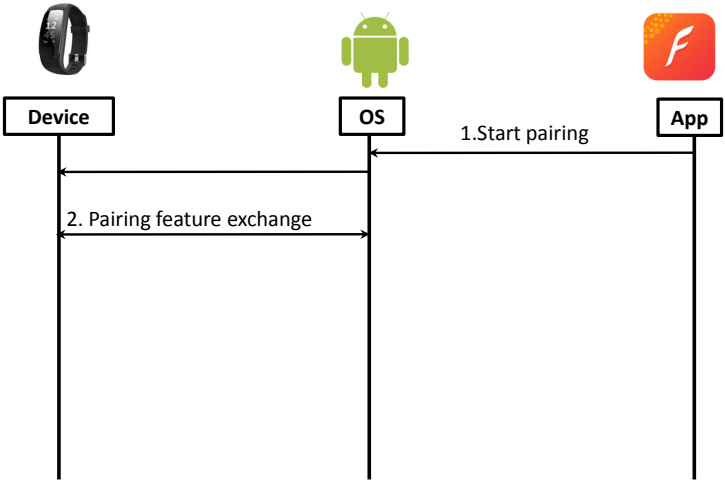
Pairing Workflow



Pairing Workflow



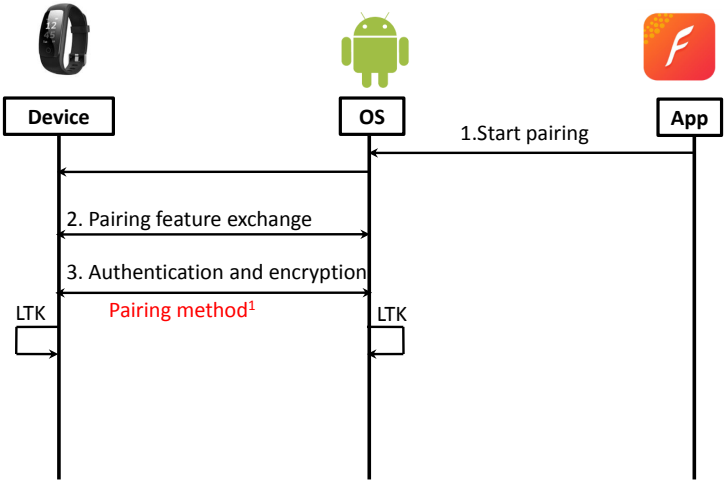
Pairing Workflow



I/O Features

- Keypad
- Screen
- Out of band Channel

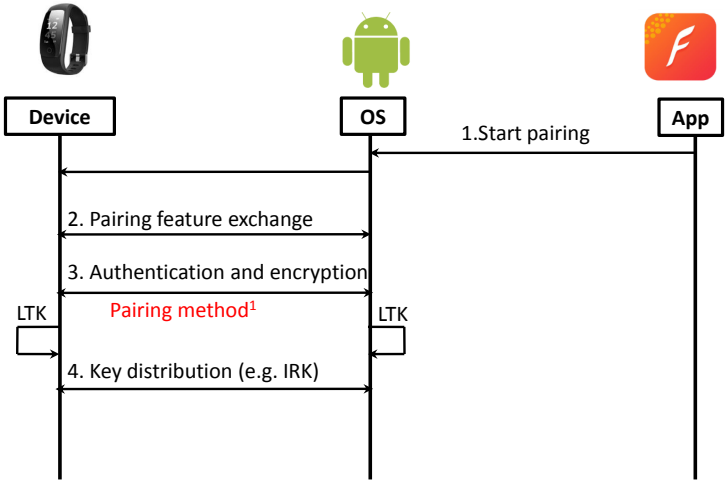
Pairing Workflow



Pairing Methods

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

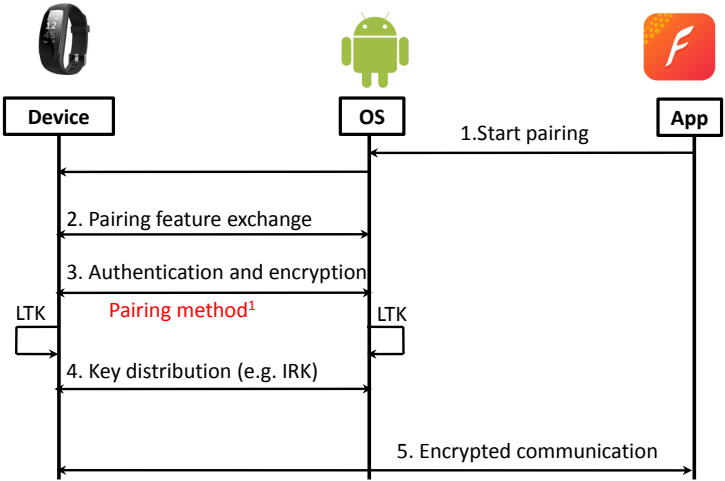
Pairing Workflow



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

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

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

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: $K_A = Pri_A * PK_B$

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

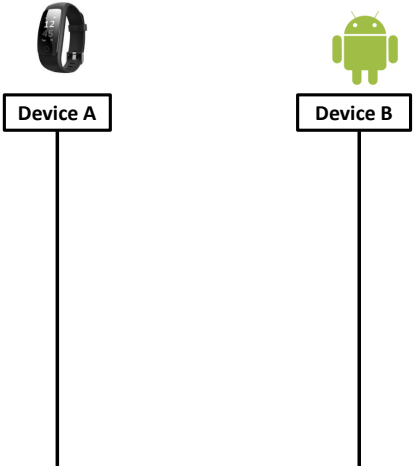
1. Alice generates a random ECC key pair: $\{Pri_A, PK_A = Pri_A$ * G$\}$
2. Bob generates a random ECC key pair: $\{Pri_B, PK_B = Pri_B$ * G$\}$
3. Alice and Bob exchanges PK_A and PK_B
4. Alice calculates sharedKey: $K_A = Pri_A * PK_B$
5. Bob calculates sharedKey: $K_B = Pri_B * PK_A$

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

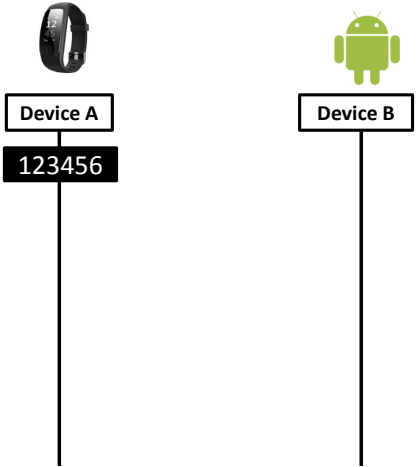
1. Alice generates a random ECC key pair: $\{Pri_A, PK_A = Pri_A * \mathrm{G}\}$
2. Bob generates a random ECC key pair: $\{Pri_B, PK_B = Pri_B * \mathrm{G}\}$
3. Alice and Bob exchanges PK_A and PK_B
4. Alice calculates sharedKey: $K_A = Pri_A * PK_B$
5. Bob calculates sharedKey: $K_B = Pri_B * PK_A$

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

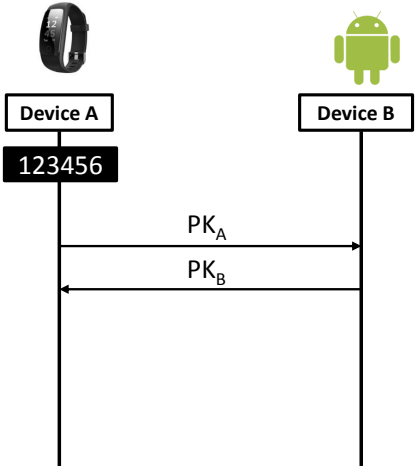
Workflow of Passkey Entry



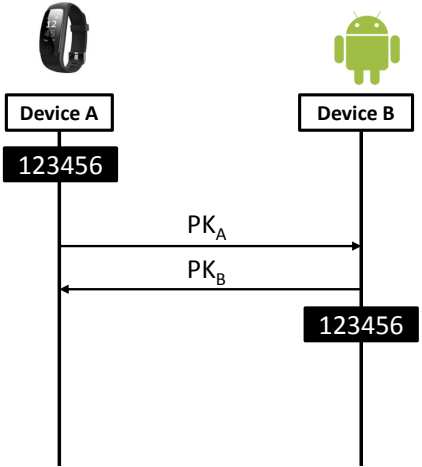
Workflow of Passkey Entry



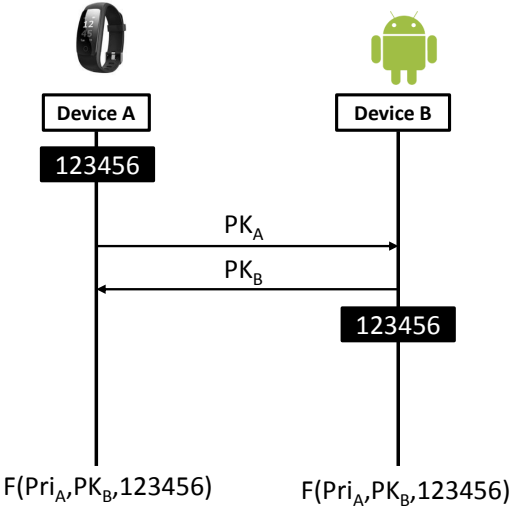
Workflow of Passkey Entry



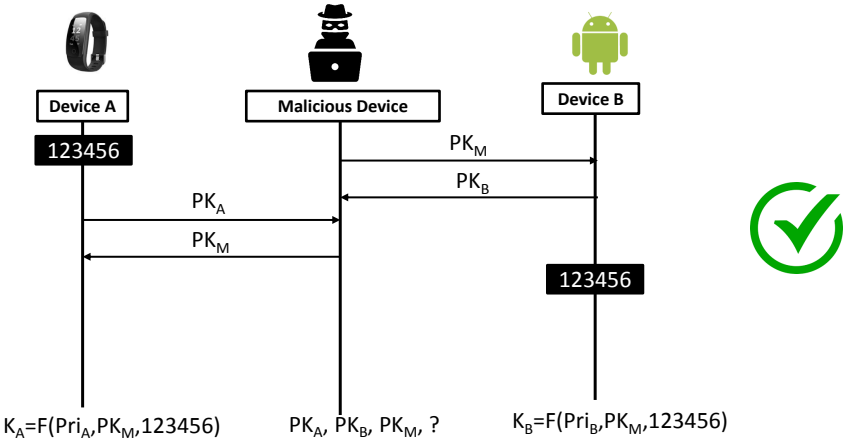
Workflow of Passkey Entry



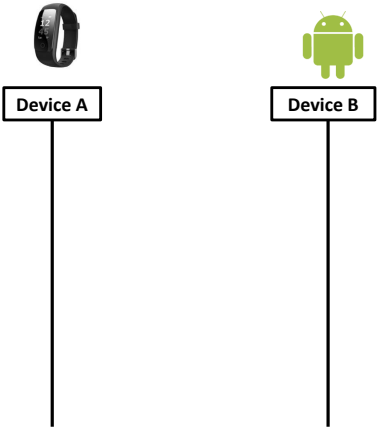
Workflow of Passkey Entry

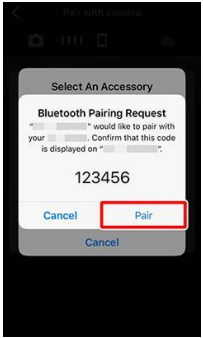


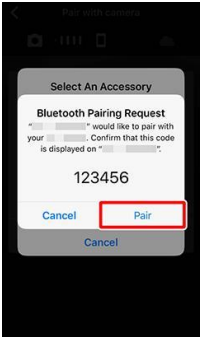
Workflow of Passkey Entry



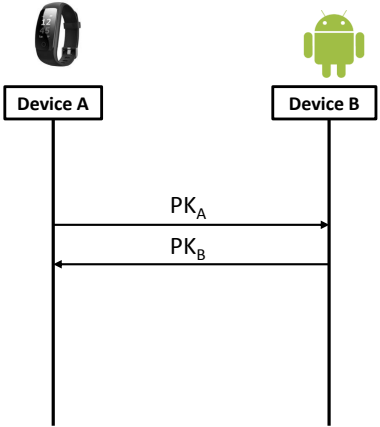
Workflow of Numeric Comparison

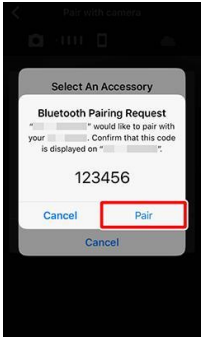


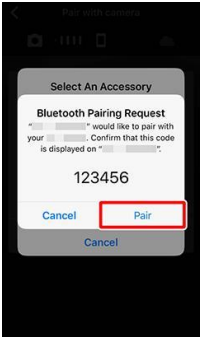




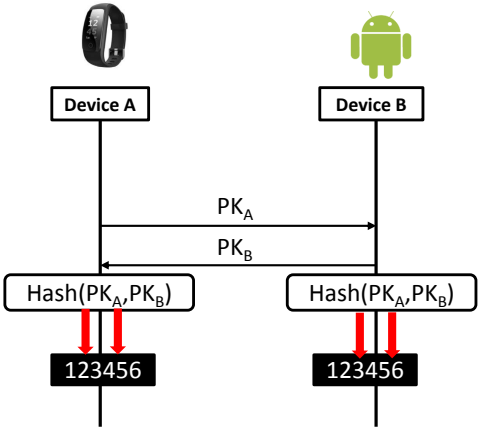
Workflow of Numeric Comparison

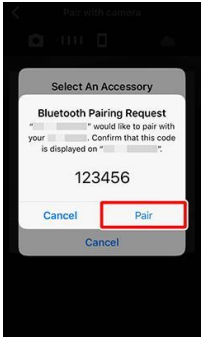


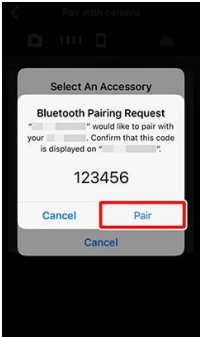




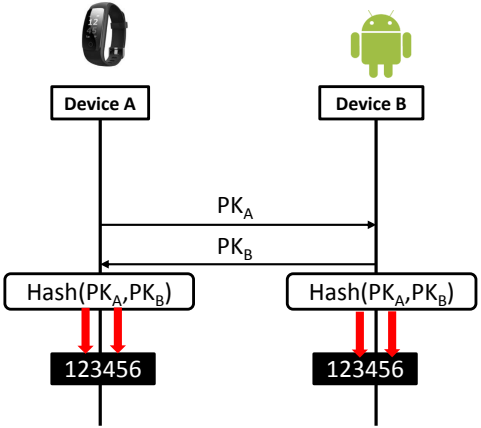
Workflow of Numeric Comparison

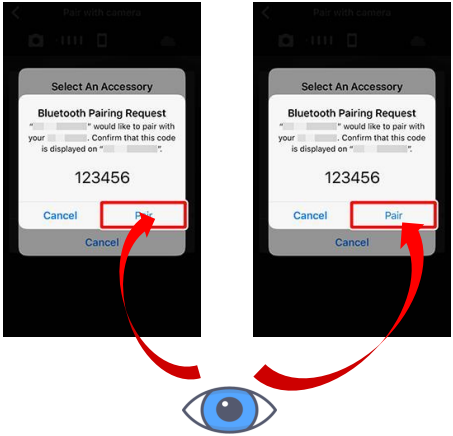




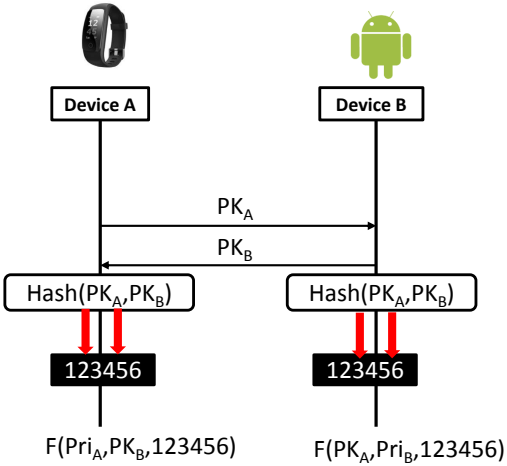


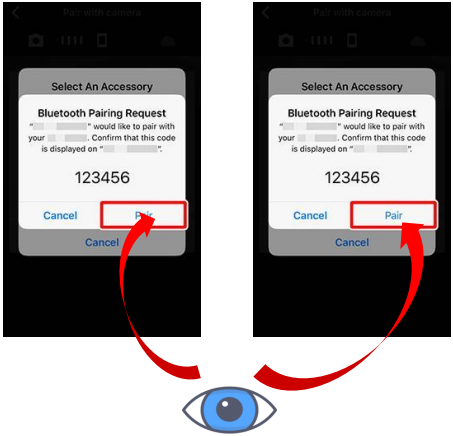
Workflow of Numeric Comparison



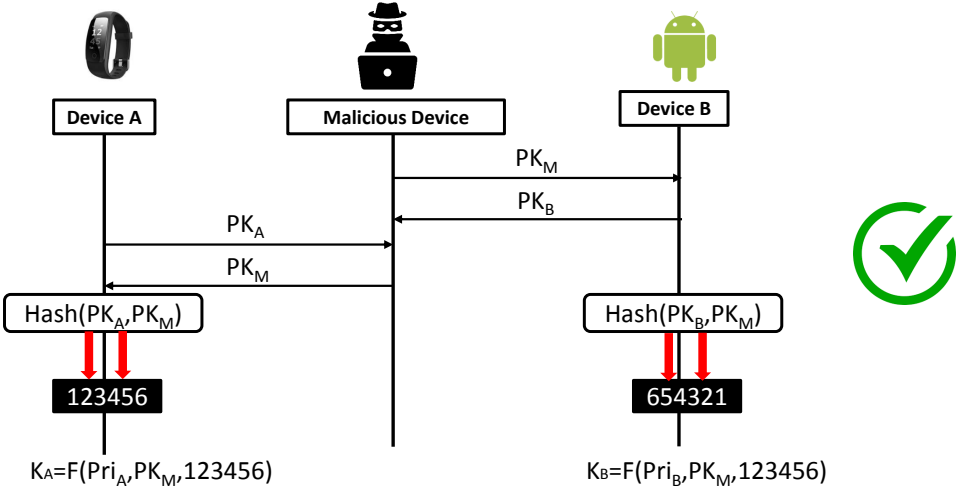


Workflow of Numeric Comparison

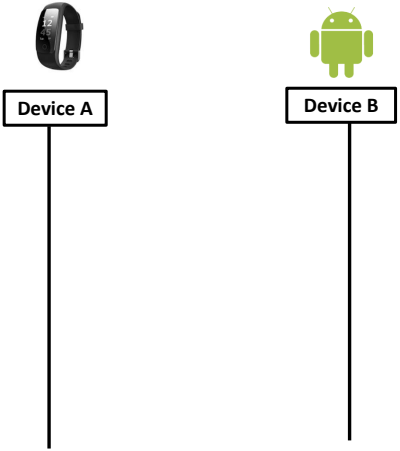




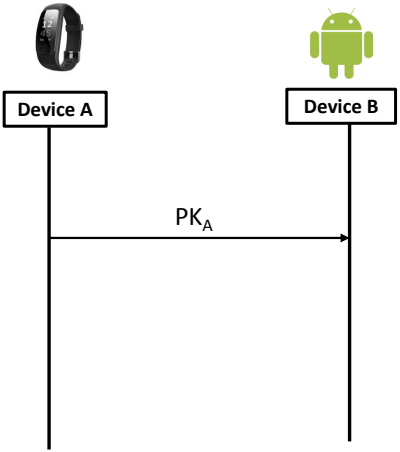
Workflow of Numeric Comparison



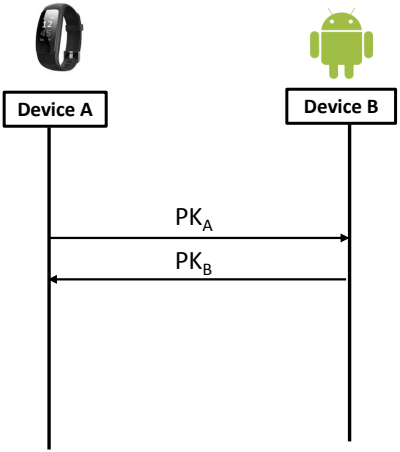
Workflow of Out of Band



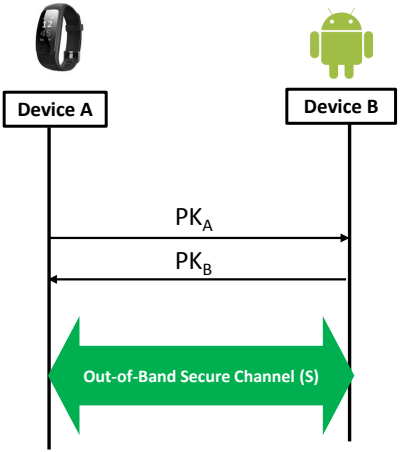
Workflow of Out of Band



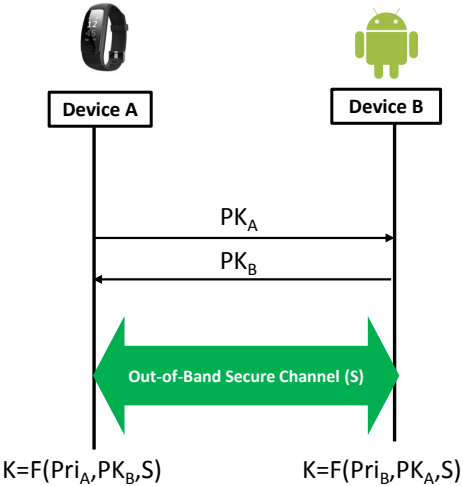
Workflow of Out of Band



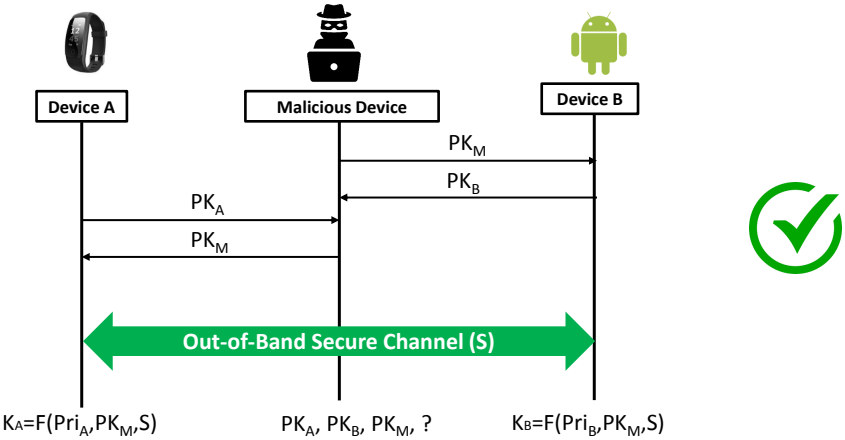
Workflow of Out of Band



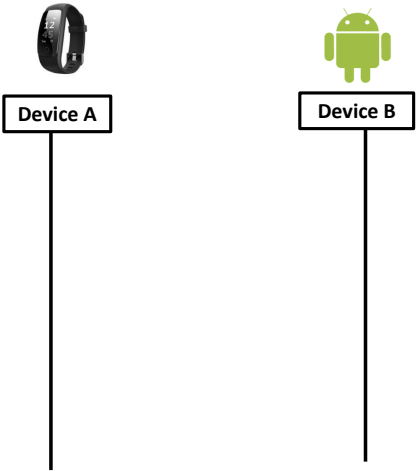
Workflow of Out of Band



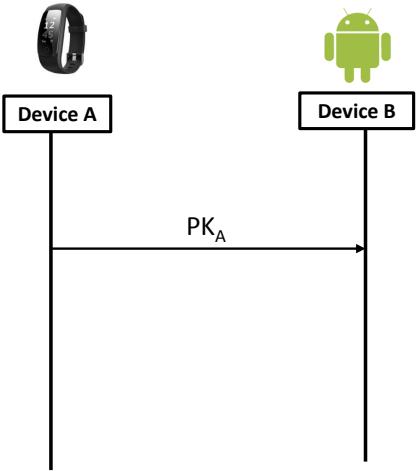
Workflow of Out of Band



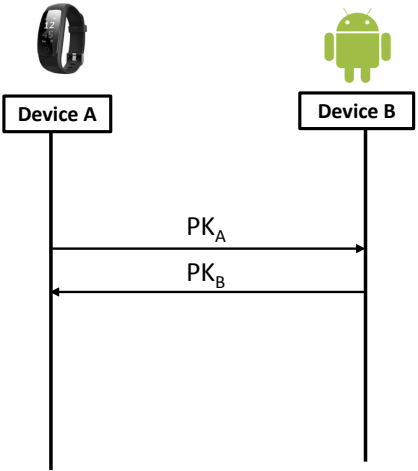
Workflow of Justworks



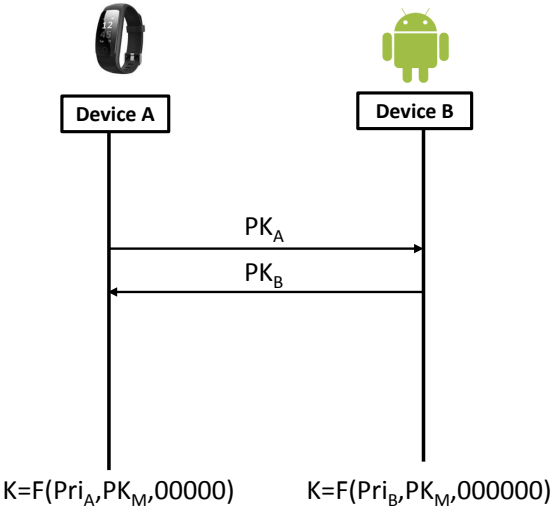
Workflow of Justworks



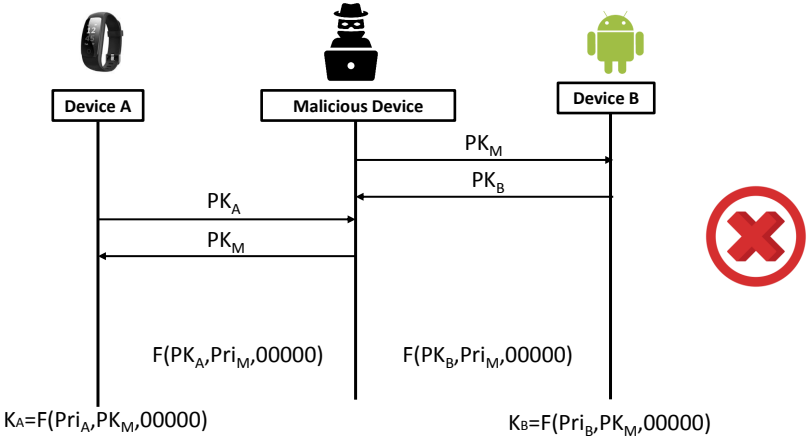
Workflow of Justworks



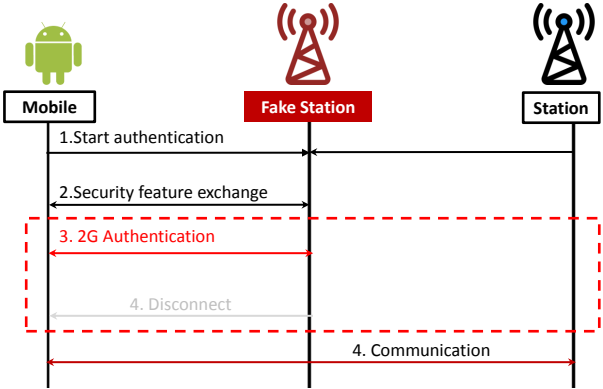
Workflow of Justworks



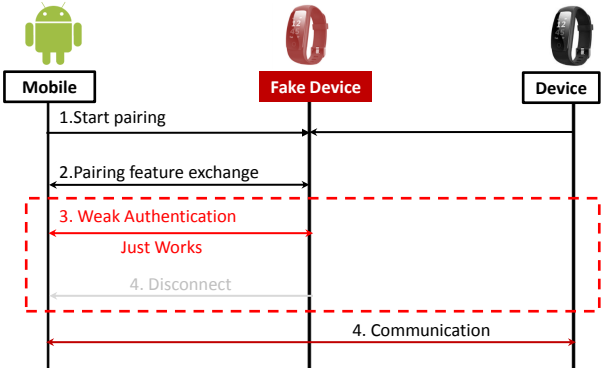
Workflow of Justworks



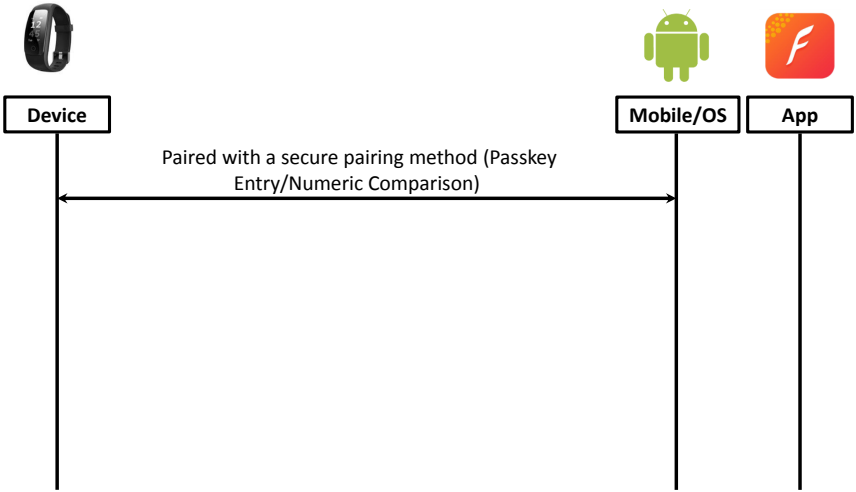
Our Downgrade Attacks against Bluetooth Low Energy



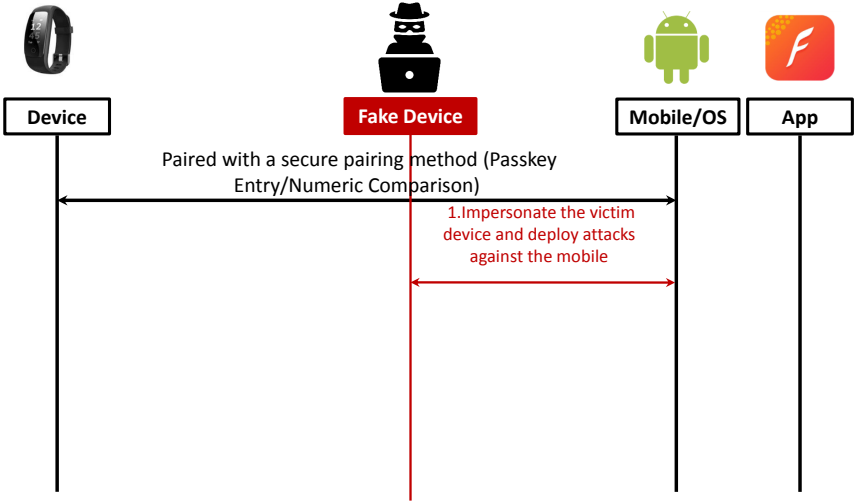
Our Downgrade Attacks against Bluetooth Low Energy



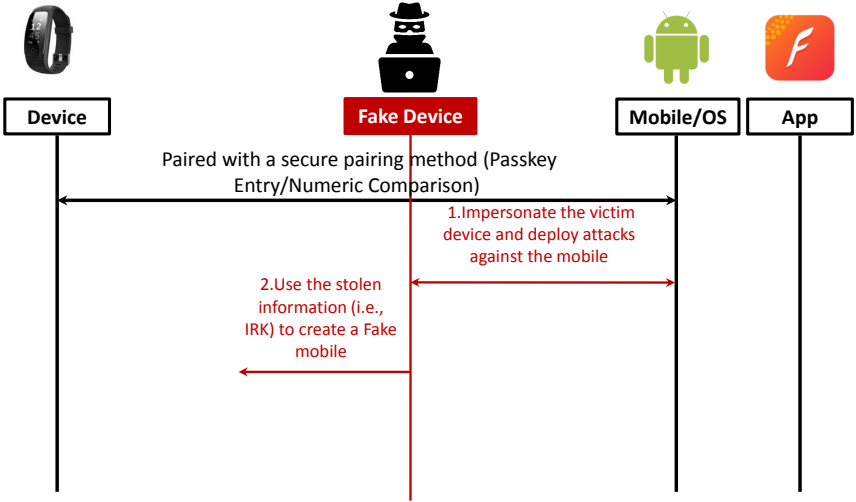
Our Downgrade Attacks against Bluetooth Low Energy



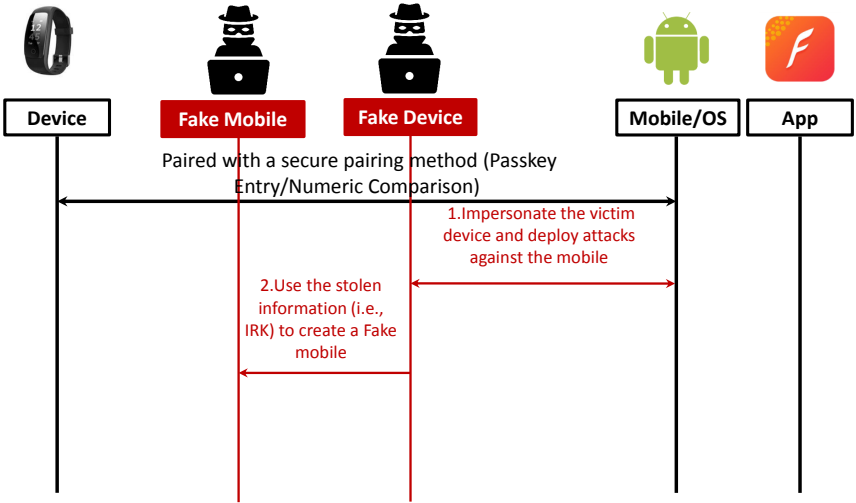
Our Downgrade Attacks against Bluetooth Low Energy



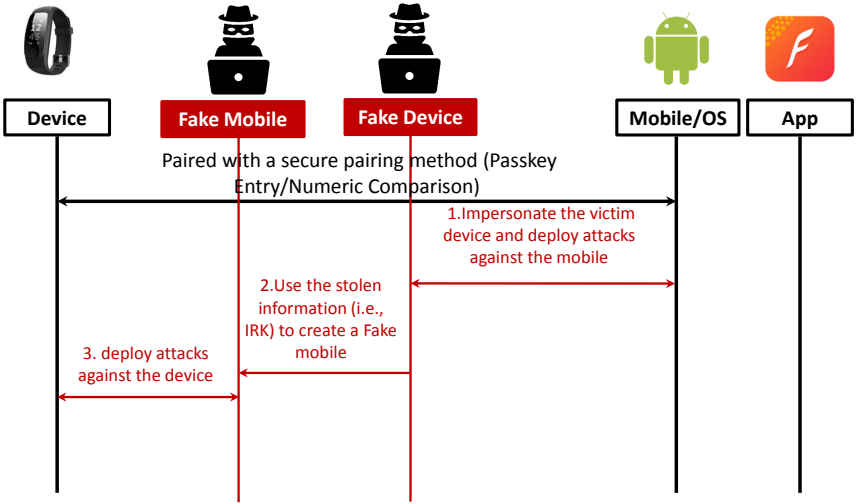
Our Downgrade Attacks against Bluetooth Low Energy



Our Downgrade Attacks against Bluetooth Low Energy

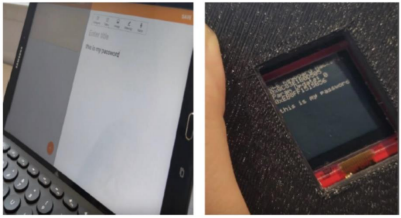


Our Downgrade Attacks against Bluetooth Low Energy



Our Downgrade Attacks against Bluetooth Low Energy

The Tested BLE devices

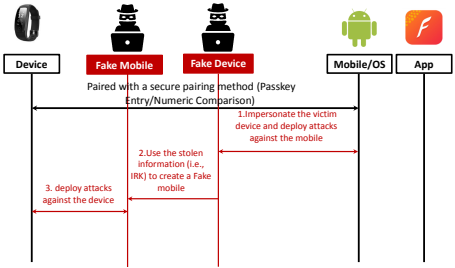


User Attacker

MITM attack against BLE keyboards

CVE-2020-9770

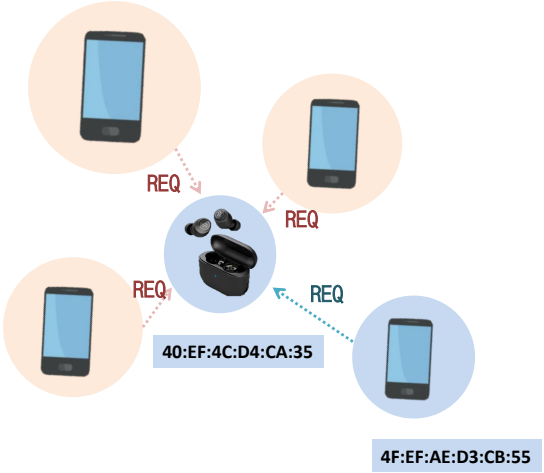
Our Downgrade Attacks against Bluetooth Low Energy



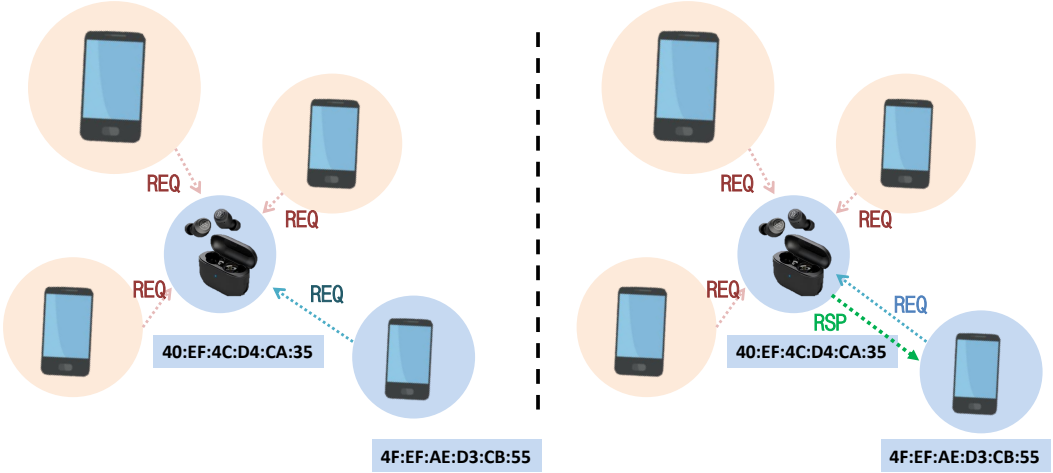
"Breaking Secure Pairing of Bluetooth Low Energy Using Downgrade Attacks", Yue Zhang, Jian Weng, Rajib Dey, Yier Jin, Zhiqiang Lin, and Xinwen Fu. *In Proceedings of the 29th USENIX Security Symposium*, Boston, MA. August 2020

Outline

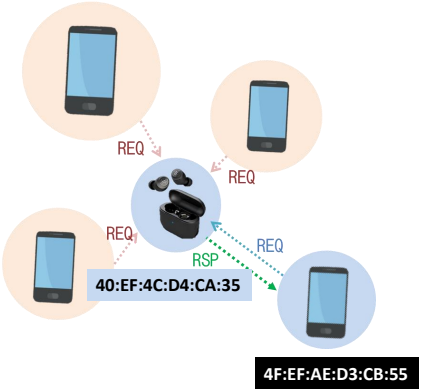
Our First Finding: Allowlist-based Side Channel



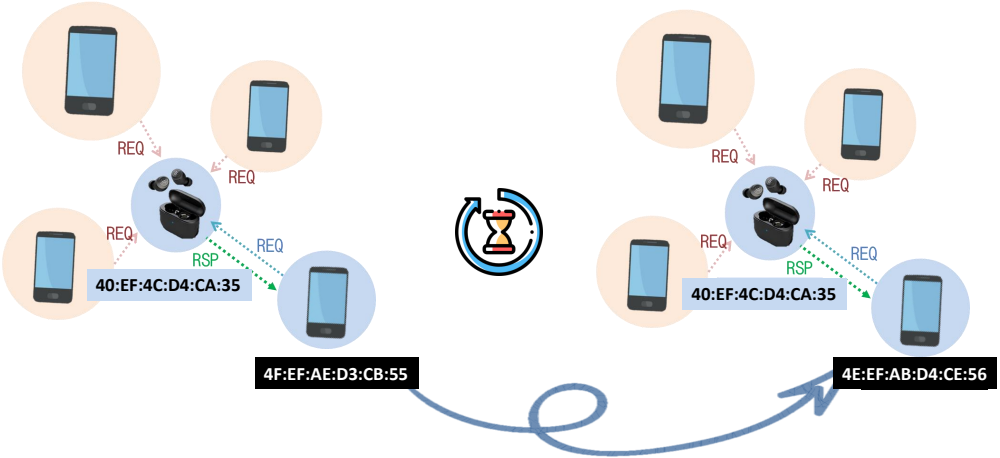
Our First Finding: Allowlist-based Side Channel



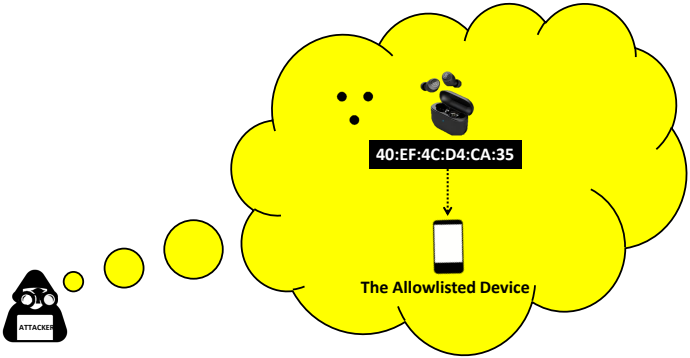
Our First Finding: Allowlist-based Side Channel



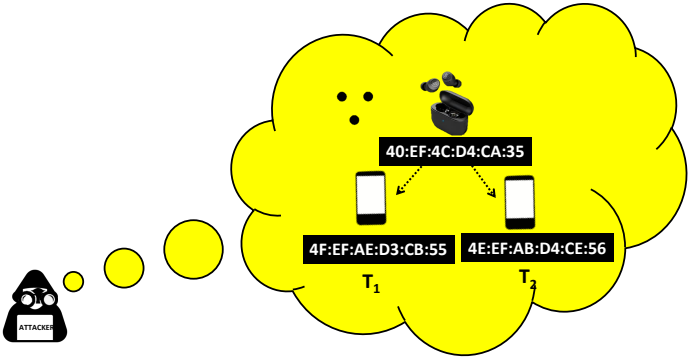
Our First Finding: Allowlist-based Side Channel



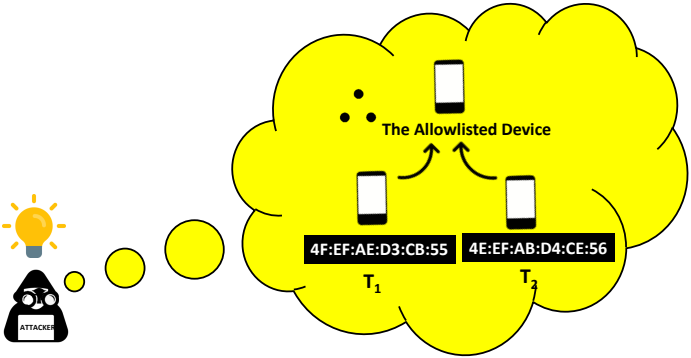
Our First Finding: Allowlist-based Side Channel



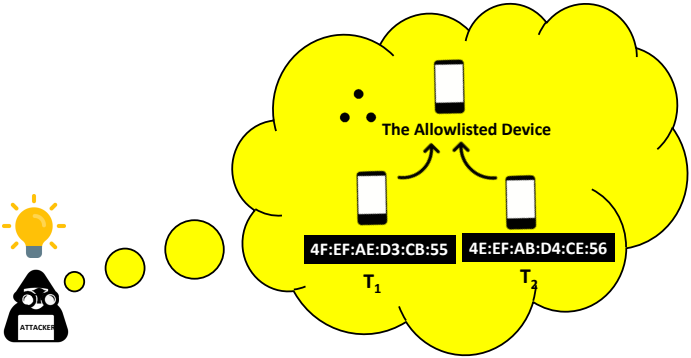
Our First Finding: Allowlist-based Side Channel



Our First Finding: Allowlist-based Side Channel

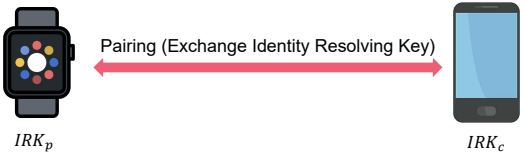


Our First Finding: Allowlist-based Side Channel

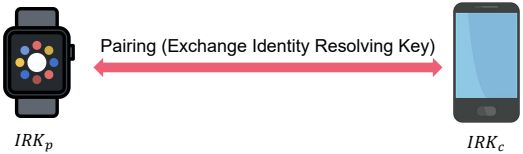


1. Cache
2. Timing
3. Power
4. Votage
5. Acoustic
6. **Allowlist**
7. ...

Our Second Finding: MAC Address Can be Replayed



Our Second Finding: MAC Address Can be Replayed



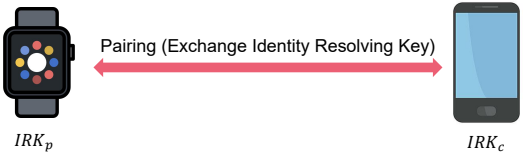
Random Address (RA) Generation

Random Address (RA) Resolution

$$RA_p = prand_{24} \,||\, H_{24}(prand_{24} \,||\, IRK_p)$$

47:2B:3C:6F:1C:DE

Our Second Finding: MAC Address Can be Replayed



Random Address (RA) Generation

$$RA_p = prand_{24} \,||\, H_{24}(prand_{24} \,||\, IRK_p)$$

47:2B:3C:6F:1C:DE

Random Address (RA) Resolution

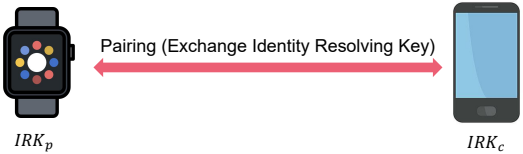
47:2B:3C:6F:1C:DE

$$RA_c = prand_{24} \,||\, H_{24}(prand_{24} \,||\, IRK_c)$$

Verification Results

RA_p ✓

Our Second Finding: MAC Address Can be Replayed



Random Address (RA) Generation

$$RA_p = prand_{24} \,||\, H_{24}(prand_{24} \,||\, IRK_p)$$

47:2B:3C:6F:1C:DE

Random Address (RA) Replay

$$RA_p^{'} = RA_p$$

47:2B:3C:6F:1C:DE

Random Address (RA) Resolution

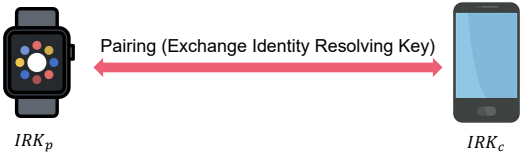
47:2B:3C:6F:1C:DE

$$RA_c = prand_{24} \,||\, H_{24}(prand_{24} \,||\, IRK_c)$$

Verification Results

RA_p ✓

Our Second Finding: MAC Address Can be Replayed



Random Address (RA) Generation

$$RA_p = prand_{24} \,||\, H_{24}(prand_{24} \,||\, IRK_p)$$

47:2B:3C:6F:1C:DE

Random Address (RA) Replay

$$RA_p^{'} = RA_p$$

47:2B:3C:6F:1C:DE

Random Address (RA) Resolution

47:2B:3C:6F:1C:DE

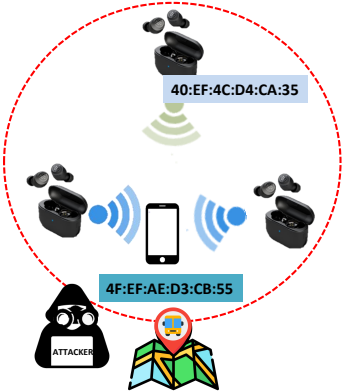
$$RA_c = prand_{24} \,||\, H_{24}(prand_{24} \,||\, IRK_c)$$

Verification Results

RA_p ✓

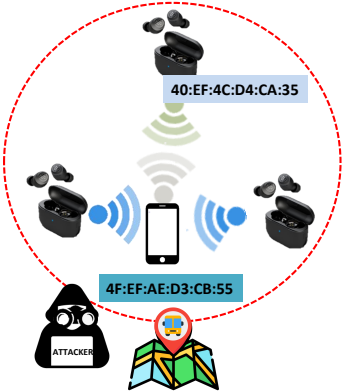
$RA_p^{'}$ ✓

Attack Example



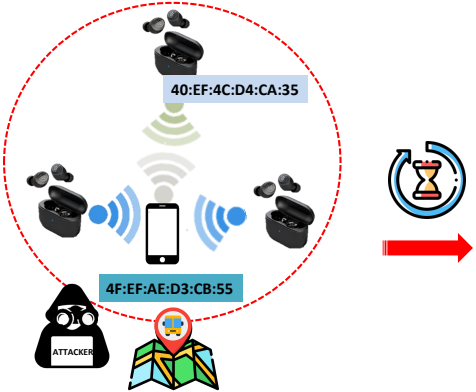
Tracking a Victim's Real-time Location

Attack Example



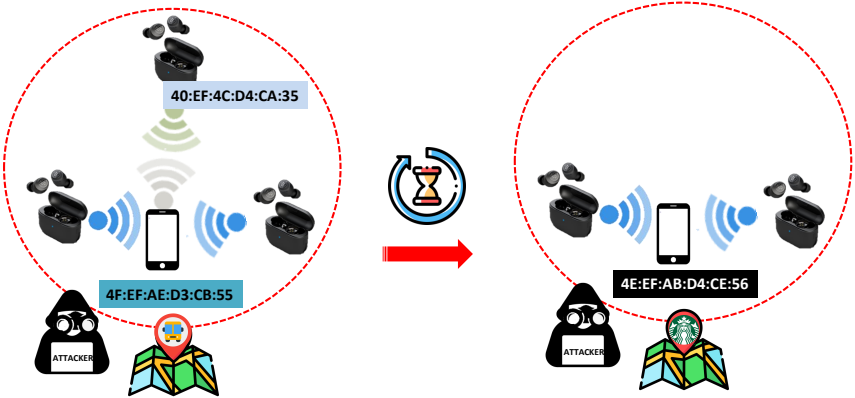
Tracking a Victim's Real-time Location

Attack Example



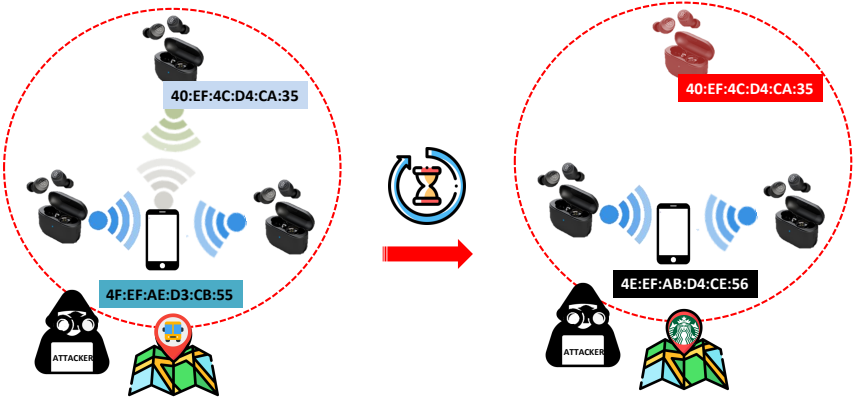
Tracking a Victim's Real-time Location

Attack Example



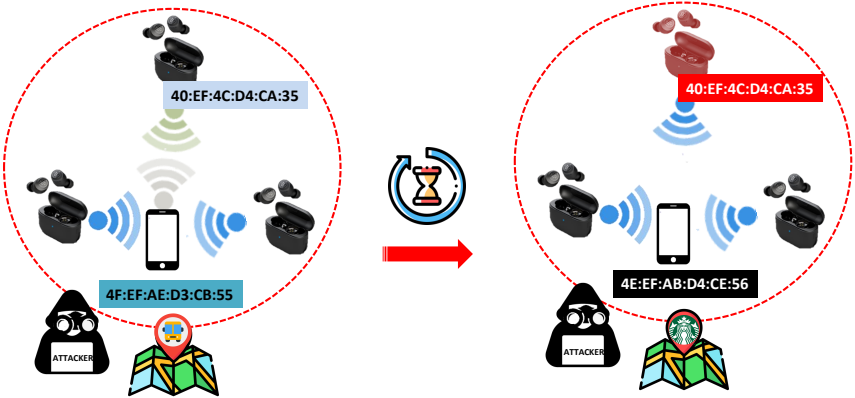
Tracking a Victim's Real-time Location

Attack Example



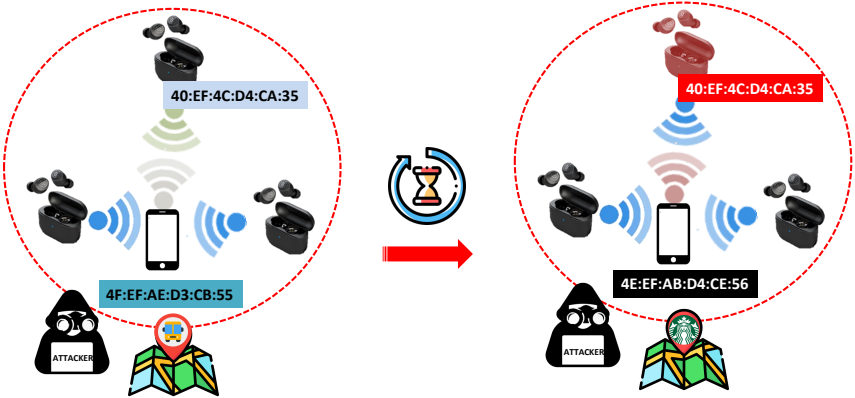
Tracking a Victim's Real-time Location

Attack Example



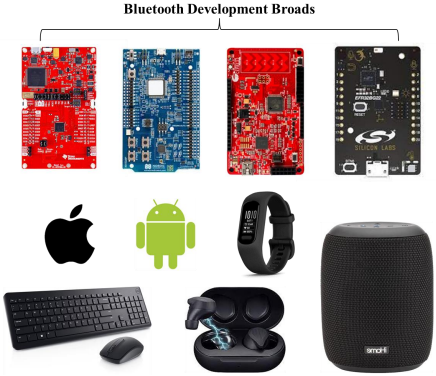
Tracking a Victim's Real-time Location

Attack Example



Tracking a Victim's Real-time Location

Devices That are Subject to BAT Attacks

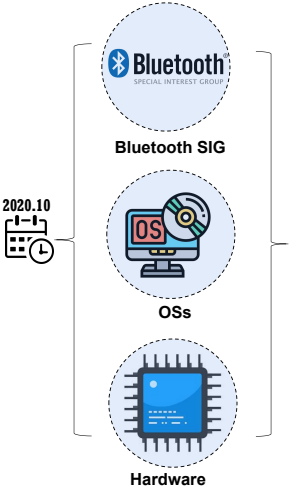


CVE-2020-35473

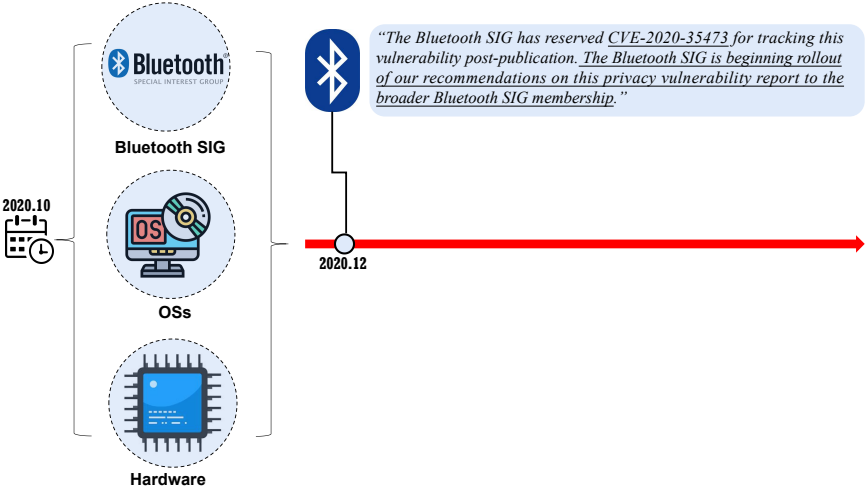
Brand & Model	Allowlist Enabled by P	Allowlist Used by C	Device Type	MAC Addr	Power Saving	Passive Attacks TC	Passive Attacks TP	Active Attacks From Malicious Central TC	Active Attacks From Malicious Central TP	Active Attacks From Malicious Peripheral TC	Active Attacks From Malicious Peripheral TP
Peripherals & Development Boards											
DRACONIC	✓	✓	Keyboard	SRA	✓	✓	✓	✓	✓	✓	✓
JellyComb	✓	✓	Keyboard	SRA	✓	✓	✓	✓	✓	✓	✓
iClever	✓	✓	Keyboard	SRA	✓	✓	✓	✓	✓	✓	✓
Microsoft (V1)	✓	✓	Keyboard	SRA	✓	✓	✓	✓	✓	✓	✓
Microsoft (V2)	✓	✓	Keyboard	SRA	✓	✓	✓	✓	✓	✓	✓
byteblue	✓	✓	Keyboard	SRA	✓	✓	✓	✓	✓	✓	✓
Logitech K780	✓	✓	Keyboard	SRA	✓	✓	✓	✓	✓	✓	✓
Logitech K830	✓	✓	Keyboard	SRA	✓	✓	✓	✓	✓	✓	✓
Logitech K380	✓	✓	Keyboard	SRA	✓	✓	✓	✓	✓	✓	✓
SXWL	✓	✓	Keyboard	SRA	✓	✓	✓	✓	✓	✓	✓
SXWL	✓	✓	Mouse	SRA	✓	✓	✓	✓	✓	✓	✓
Inphic	✓	✓	Mouse	SRA	✓	✓	✓	✓	✓	✓	✓
Vogek	✓	✓	Mouse	SRA	✓	✓	✓	✓	✓	✓	✓
JellyComb (V1)	✓	✓	Mouse	SRA	✓	✓	✓	✓	✓	✓	✓
JellyComb (V2)	✓	✓	Mouse	SRA	✓	✓	✓	✓	✓	✓	✓
SEENDA	✓	✓	Mouse	SRA	✓	✓	✓	✓	✓	✓	✓
MiBand 4C	✓	✗	Wristband	PA	✗	✓	✓	✓	✓	✗	✓
i-Home Alexa	✗	✓	Speaker	PA	✓	✗	✓	✓	✓	✓	✓
TEZO	✗	✓	Earbuds	PA	✓	✗	✓	✓	✓	✓	✓
Boltune	✗	✓	Earbuds	PA	✓	✗	✓	✓	✓	✓	✓
SoundBot	✗	✓	Earbuds	PA	✓	✗	✓	✓	✓	✓	✓
Rituk	✗	✓	Keyboard	PA	✓	✗	✓	✗	✓	✓	✓
Cimetech	✗	✓	Mouse	SRA	✓	✗	✓	✗	✓	✓	✓
Ergonomic	✗	✓	Mouse	SRA	✓	✗	✓	✗	✓	✓	✓
TI CC2640R2F	✓	✓	Dev Board	RPA	-	✓	✓	✓	✓	✓	✓
Nordic NRF52	✓	✓	Dev Board	RPA	-	✓	✓	✓	✓	✓	✓
Silicon Labs 6101D	✗	✓	Dev Board	RPA	-	-	-	✗	✗	✓	✓
Crypsis CYBiCIT	✗	✓	Dev Board	RPA	-	-	-	✗	✗	✓	✓

Brand & Model	Allowlist Enabled by C	Allowlist Used by P	Type & OS	MAC Addr	Random Interval	Passive Attacks TP	Passive Attacks TC	Active Attacks From Malicious Central TP	Active Attacks From Malicious Central TC	Active Attacks From Malicious Peripheral TP	Active Attacks From Malicious Peripheral TC
Centrals											
Google Pixel 4	✓	✓	Phone (Android 11)	RPA	5-15	✓	✓	✓	✓	✓	✓
Google Pixel 2	✓	✓	Phone (Android 10)	RPA	5-15	✓	✓	✓	✓	✓	✓
Samsung S10	✓	✓	Phone (Android 10)	RPA	5-15	✓	✓	✓	✓	✓	✓
Google Pixel 4	✓	✓	Phone (Android 10)	RPA	5-15	✓	✓	✓	✓	✓	✓
iPhone 8	✓	✓	Phone (iOS 13.2)	RPA	15	✓	✓	✓	✓	✓	✓
iPhone 11	✓	✓	Phone (iOS 13.2)	RPA	15	✓	✓	✓	✓	✓	✓
iPad	✓	✓	Tablet (iOS 13.2)	RPA	15	✓	✓	✓	✓	✓	✓
Dell GD1H4KU	✓	✓	Laptop (Windows 10)	PA	$+\infty$	✓	✓	✓	✓	✓	✓
Dell	✓	✓	Laptop (Ubuntu 20.02)	PA	$+\infty$	✓	✓	✓	✓	✓	✓
Thinkpad T450s	✓	✓	Laptop (Windows 8)	PA	$+\infty$	✓	✓	✓	✓	✓	✓
Surface Pro	✓	✓	Tablet (Windows 10)	PA	$+\infty$	✓	✓	✓	✓	✓	✓

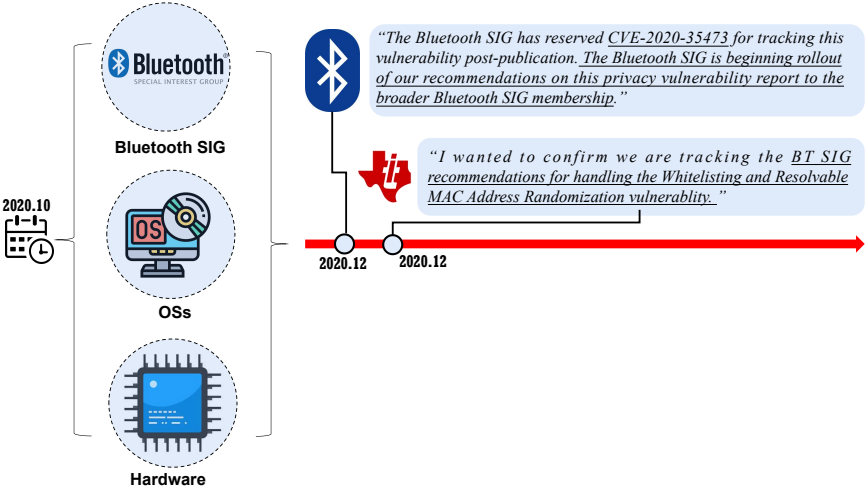
Responsible Disclosure



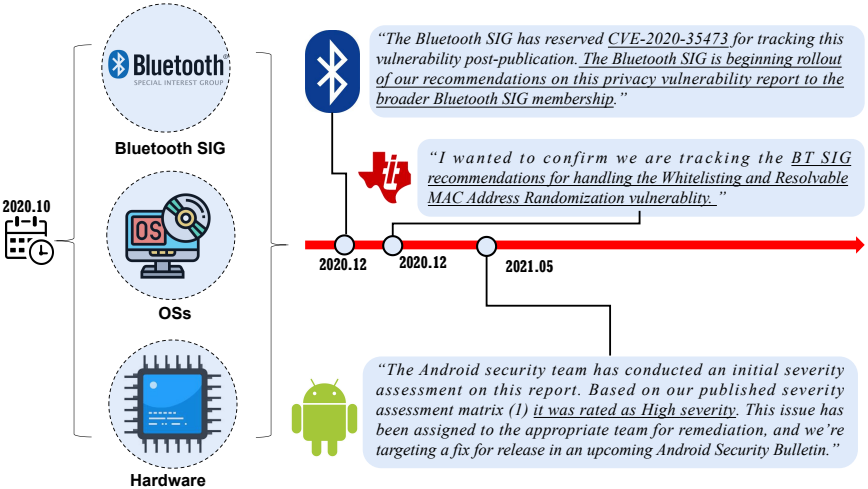
Responsible Disclosure



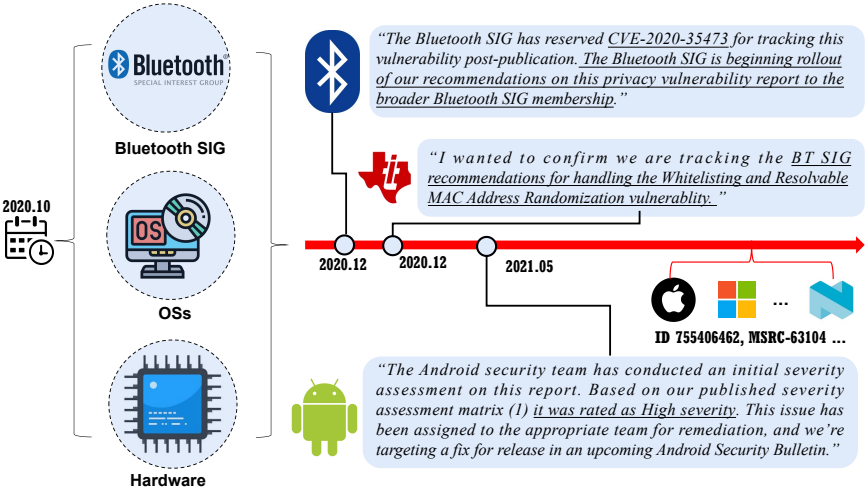
Responsible Disclosure



Responsible Disclosure



Responsible Disclosure



Our Countermeasure: Securing Address of BLE (SABLE)

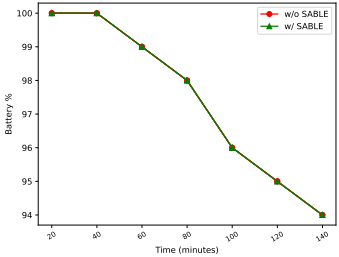
Allowlist Side Channel (Mitigation)

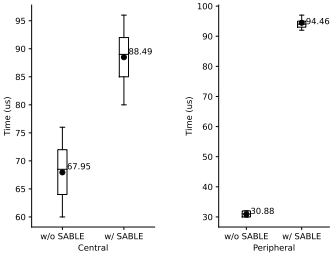
- We advocate the use of an interval unpredictable, central and peripheral synchronized RPA generation scheme to mitigate the side channel.

MAC Address Replay (Prevention)

- We propose adding a sequence number (which could be a timestamp) when generating the RPA to ensure that each MAC address can only be used once to prevent the replay attack.

Performance of SABLE

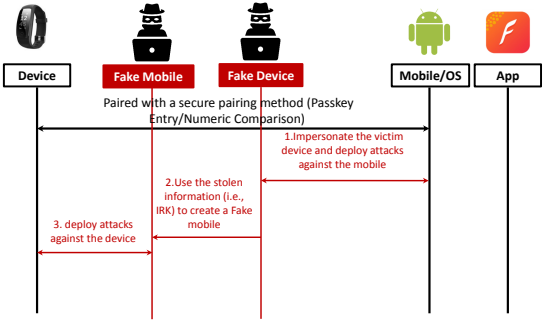




"When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure". Yue Zhang, and Zhiqiang Lin. *In Proceedings of the 29th ACM Conference on Computer and Communications Security (CCS 2022)*. November 2022

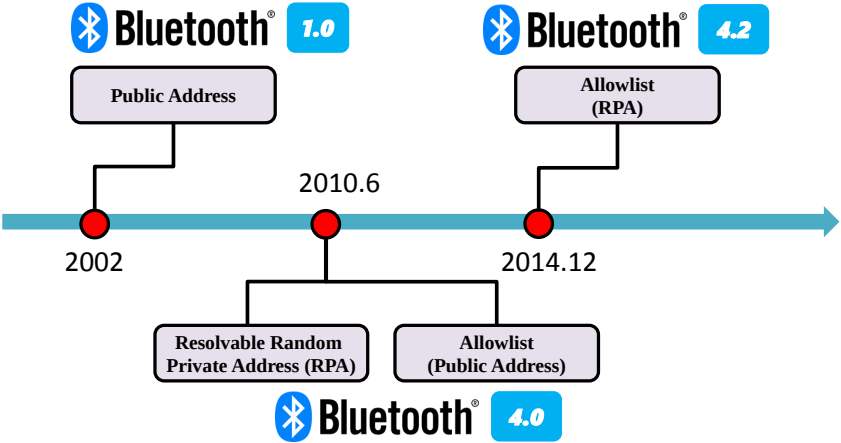
Outline

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

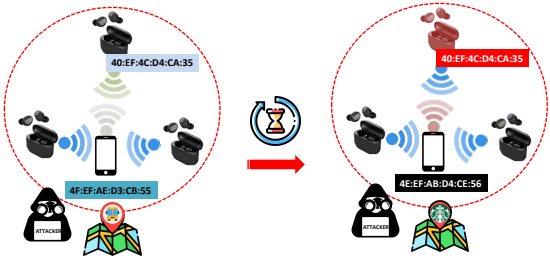


- Bluetooth low energy (BLE) pairing can be **downgraded**
- There are many stages that are not part of the pairing process, but they are, in fact, closely related to pairing security.
- A systematic analysis of the pairing process, including the **error handling** of BLE communication, is needed.

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

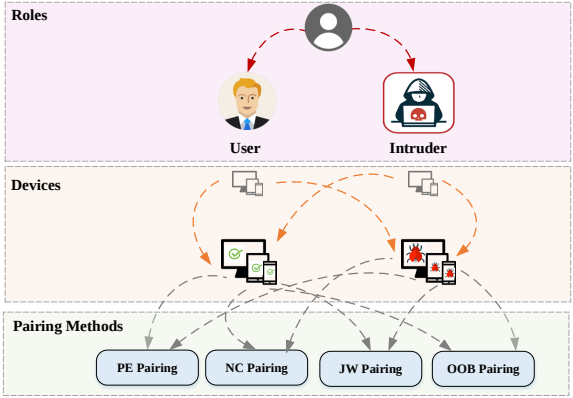


Lesson Learned (2/3): New Features Need Re-examinations



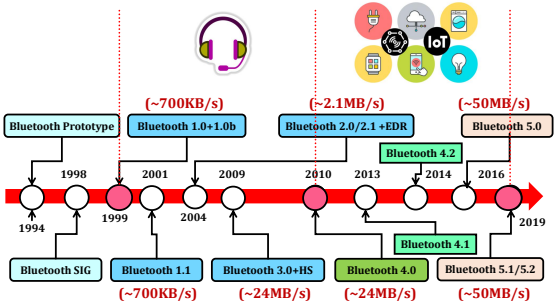
- BLE introduces multiple new features, some of which may **violate existing assumptions**
- Simliar to allowlist, those new features need to be **scrunitized**. For example, Cross-transport key derivation (CTKD); Authorization; The Connection Signature Resolving Key (CSRK).

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.

Our Recent Work on Bluetooth Security and Privacy



1. ~~BLEScope: Automatic Fingerprinting of Vulnerable BLE IoT Devices with Static UUIDs from Mobile Apps. In ACM CCS 2019~~
2. ~~FirmXRay: Detecting Bluetooth Link Layer Vulnerabilities From Bare-Metal Firmware. In ACM CCS 2020.~~
3. **Breaking Secure Pairing of Bluetooth Low Energy in Mobile Devices Using Downgrade Attacks**. In USENIX Security 2020
4. **When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure"**. In ACM CCS 2022 (Best paper award honorable mention)
5. ~~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

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

11/28/2022