Rethinking the Security and Privacy of Bluetooth Low Energy

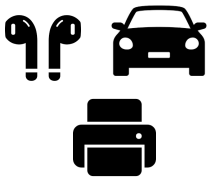
Zhiqiang Lin
Distinguished Professor of Engineering
zlin@cse.ohio-state.edu

05/02/2024

What is Bluetooth (Low Energy)

Bluetooth®

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



Bluetooth SMART

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

Applications of Bluetooth





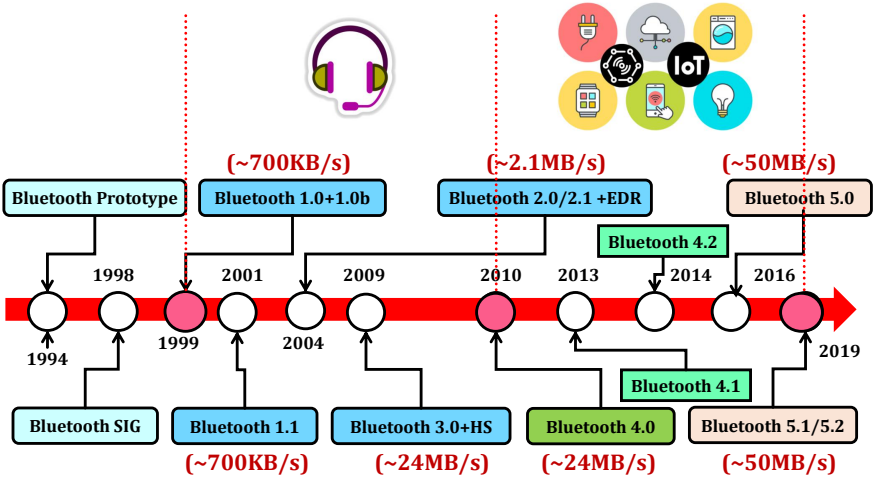


Why Named Bluetooth

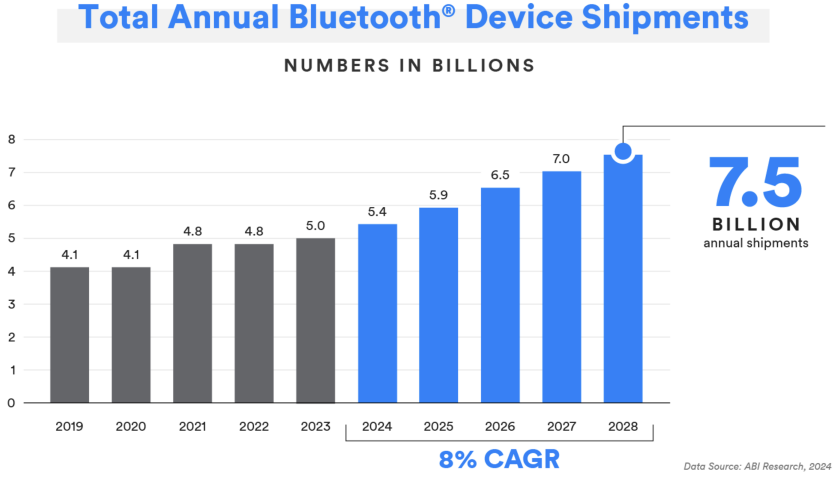
Harald "Bluetooth" Gormsson

- 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.

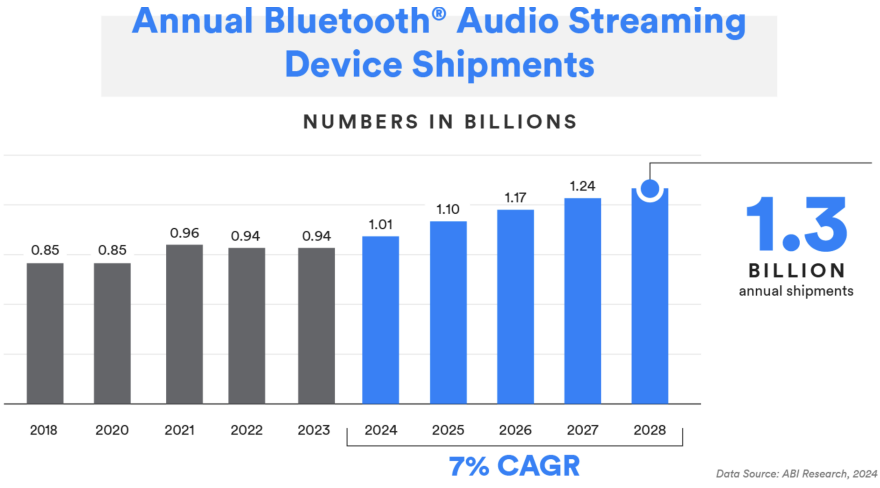
History of Bluetooth



Total Annual Bluetooth Device Shipments



Total Annual Bluetooth Device Shipments



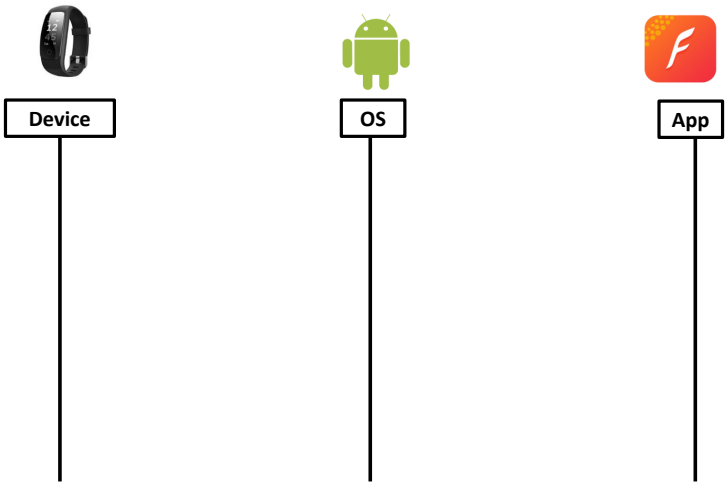
Our Recent Works on Bluetooth Security and Privacy

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. On the Accuracy of Measured Proximity of **Bluetooth**-based Contact Tracing Apps. In SECURECOMM 2020
5. Replay (Far) Away: Exploiting and Fixing Google/Apple Exposure Notification Contact Tracing. In PETS 2022.
6. 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)
7. Uncovering Vulnerabilities of **Bluetooth Low Energy** IoT from Companion Mobile Apps with Ble-Guuide. In ASIACCS 2023
8. Extrapolating Formal Analysis to Uncover Attacks in **Bluetooth** Passkey Entry Pairing. In NDSS 2023

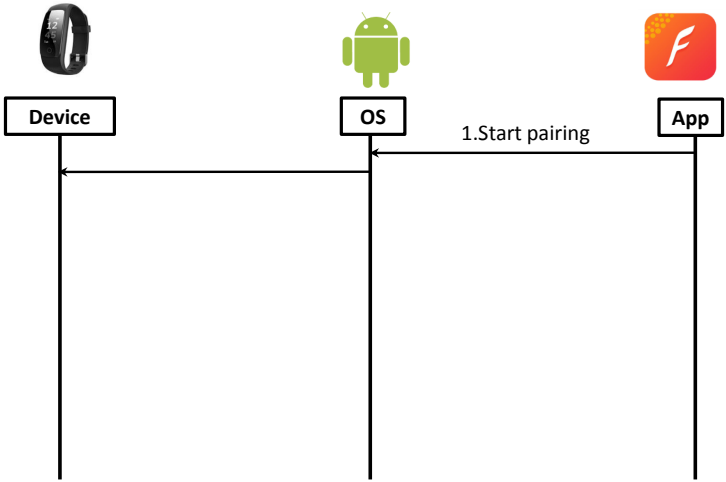
Our Recent Works on Bluetooth Security and Privacy

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. ~~On the Accuracy of Measured Proximity of Bluetooth-based Contact Tracing Apps. In SECURECOMM 20. October 2020~~
5. ~~Replay (Far) Away: Exploiting and Fixing Google/Apple Exposure Notification Contact Tracing. In PETS, July 2022.~~
6. 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)
7. ~~Uncovering Vulnerabilities of **Bluetooth Low Energy** IoT from Companion Mobile Apps with Ble-Guuide. In ASIACCS 2023~~
8. Extrapolating Formal Analysis to Uncover Attacks in **Bluetooth** Passkey Entry Pairing. In NDSS 2023

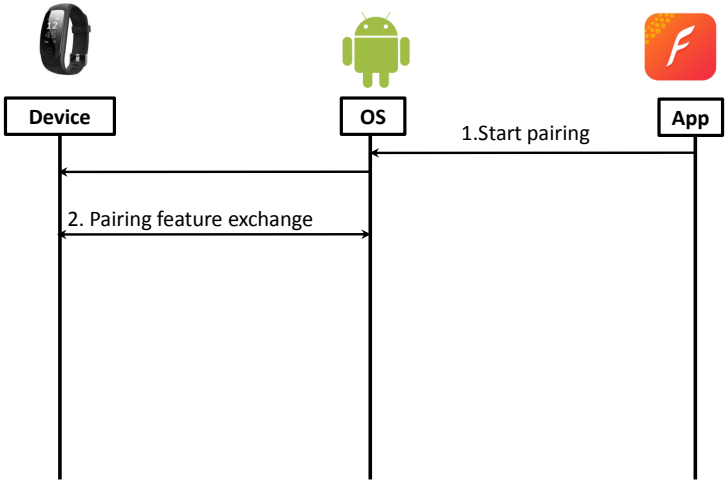
Pairing Workflow



Pairing Workflow



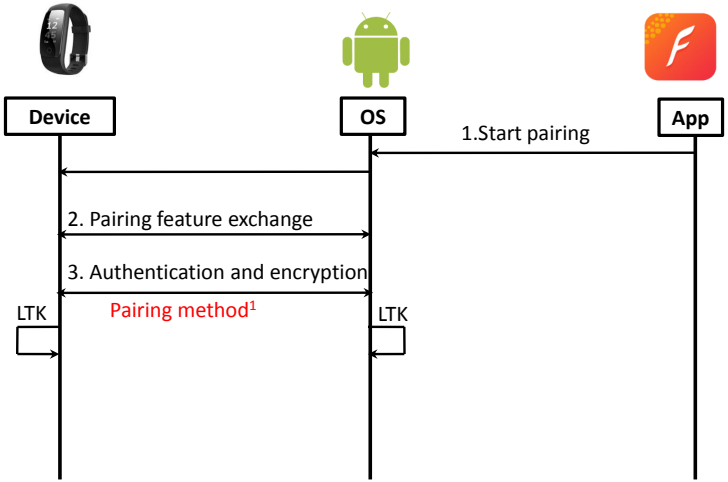
Pairing Workflow



I/O Features

- Keypad
- Screen
- Out of band Channel

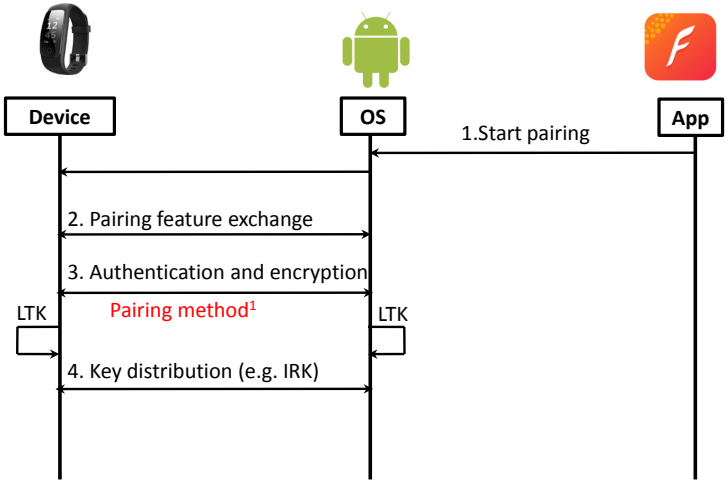
Pairing Workflow



Pairing Methods

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

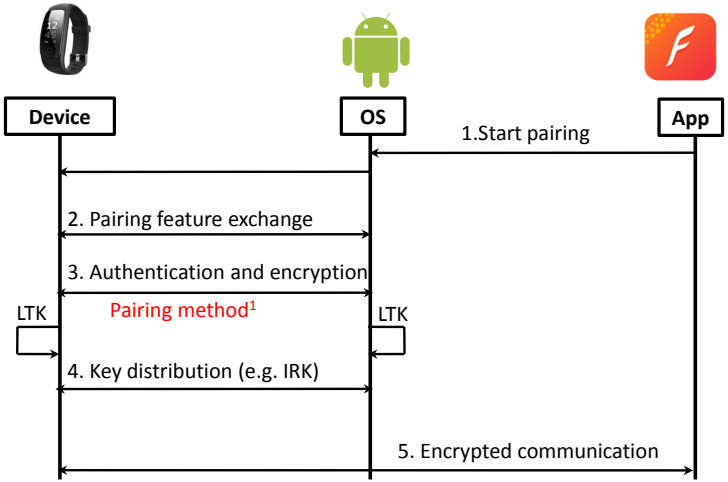
Pairing Workflow



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: $\{Pri_A, PK_A = Pri_A * \mathrm{G}\}$

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: $\{Pri_B, PK_B = Pri_B * \mathrm{G}\}$

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

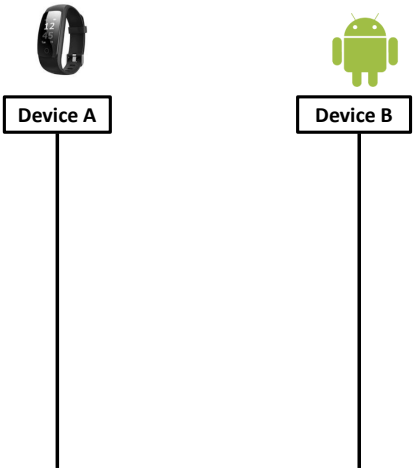
1. Alice generates a random ECC key pair: $\{Pri_A, PK_A = Pri_A$ * G$\}$
2. Bob generates a random ECC key pair: $\{Pri_B, PK_B = Pri_B$ * G$\}$
3. Alice and Bob exchanges PK_A and PK_B
4. Alice calculates sharedKey: $K_A = Pri_A * PK_B$
5. Bob calculates sharedKey: $K_B = Pri_B * PK_A$

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

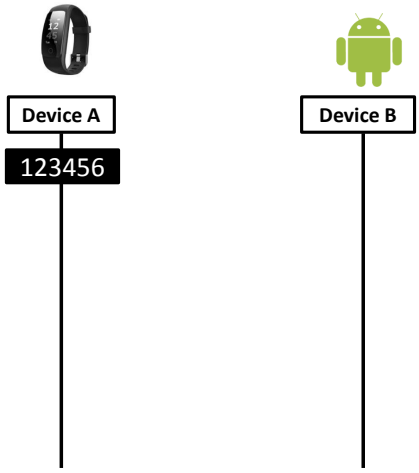
1. Alice generates a random ECC key pair: $\{Pri_A, PK_A = Pri_A$ * G$\}$
2. Bob generates a random ECC key pair: $\{Pri_B, PK_B = Pri_B$ * G$\}$
3. Alice and Bob exchanges PK_A and PK_B
4. Alice calculates sharedKey: $K_A = Pri_A * PK_B$
5. Bob calculates sharedKey: $K_B = Pri_B * PK_A$

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

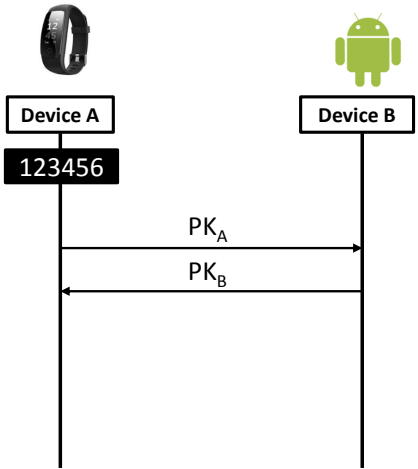
Workflow of Passkey Entry



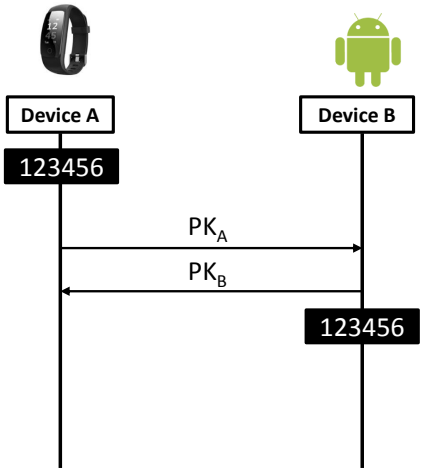
Workflow of Passkey Entry



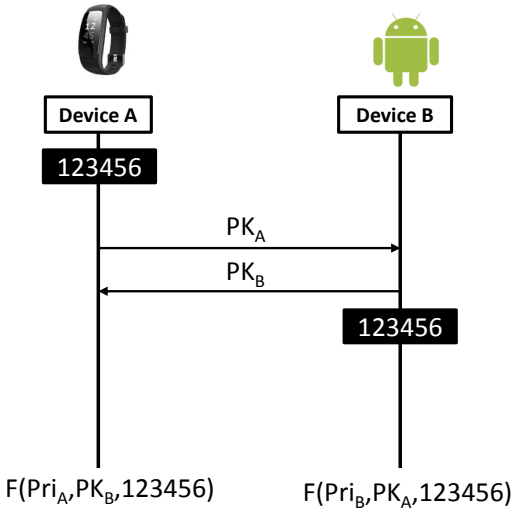
Workflow of Passkey Entry



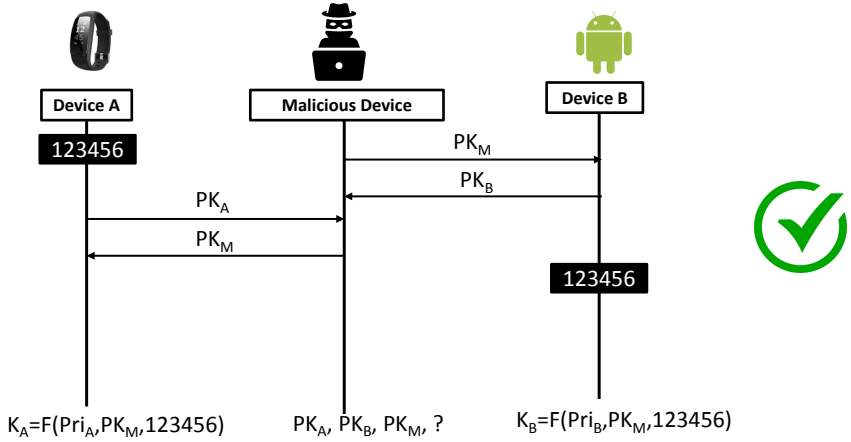
Workflow of Passkey Entry



Workflow of Passkey Entry



Workflow of Passkey Entry

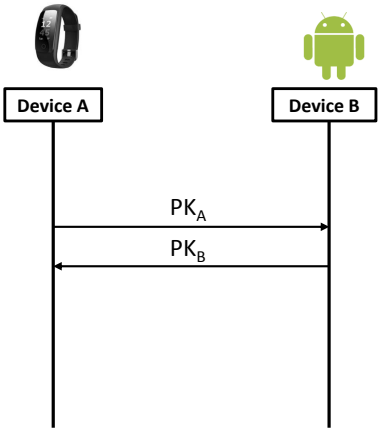


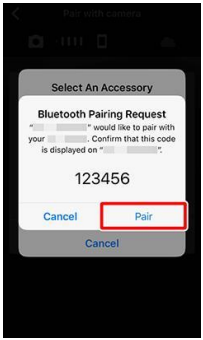
Workflow of Numeric Comparison

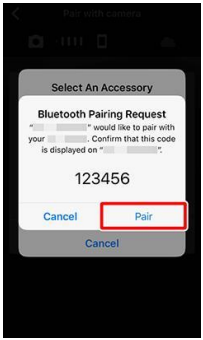
Device A

Device B

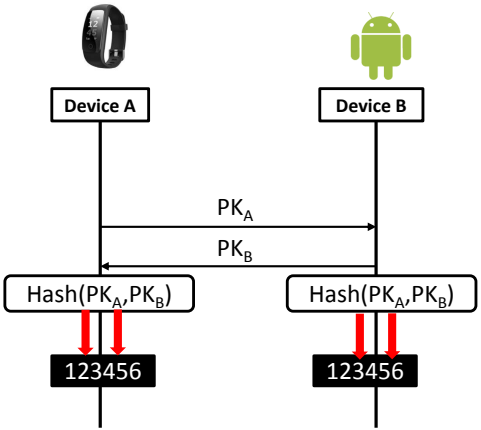
Workflow of Numeric Comparison

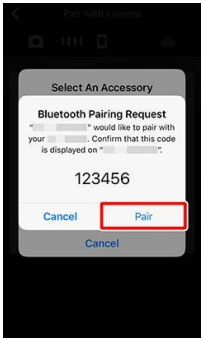


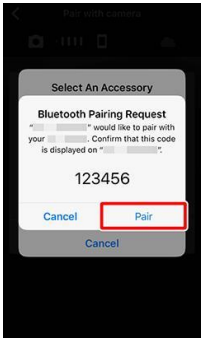




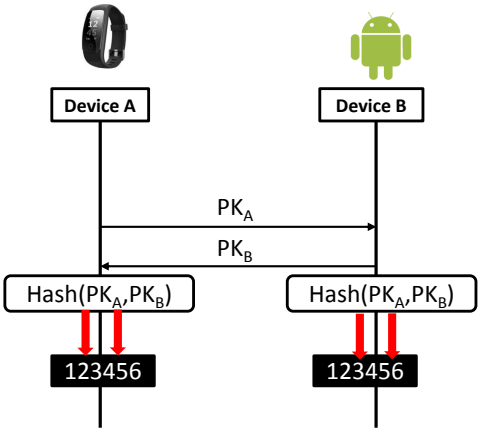
Workflow of Numeric Comparison

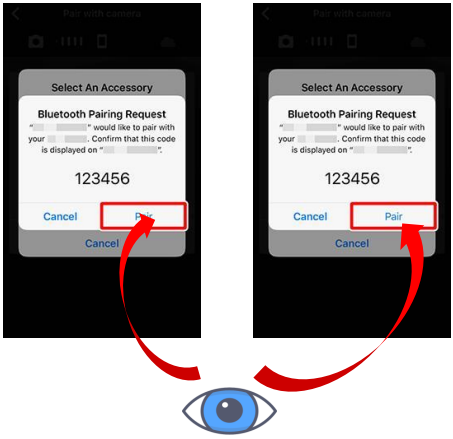




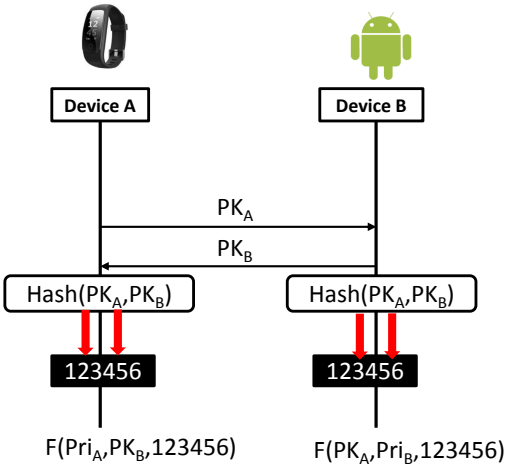


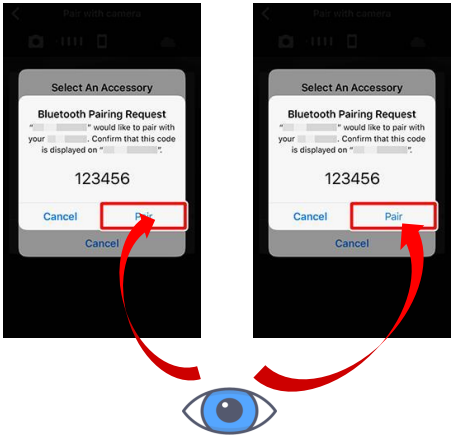
Workflow of Numeric Comparison



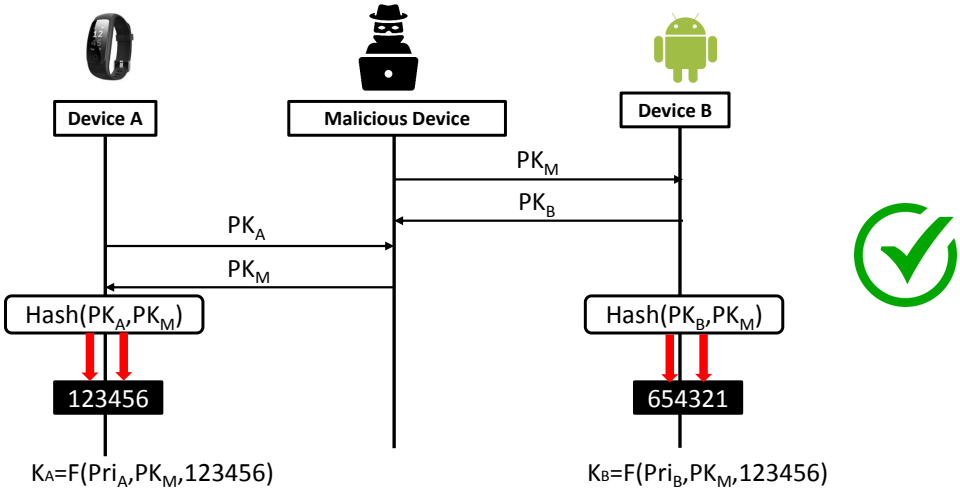


Workflow of Numeric Comparison

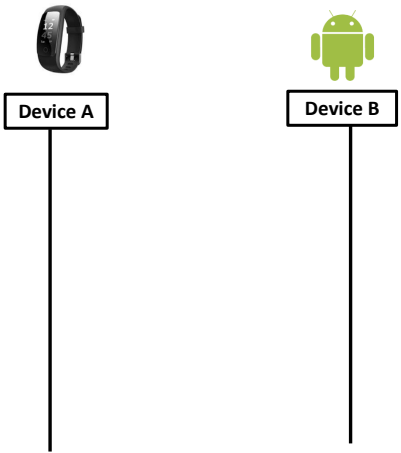




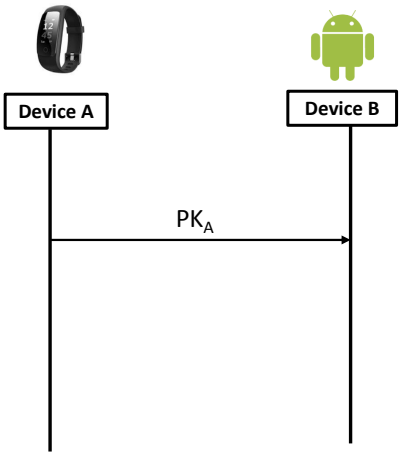
Workflow of Numeric Comparison



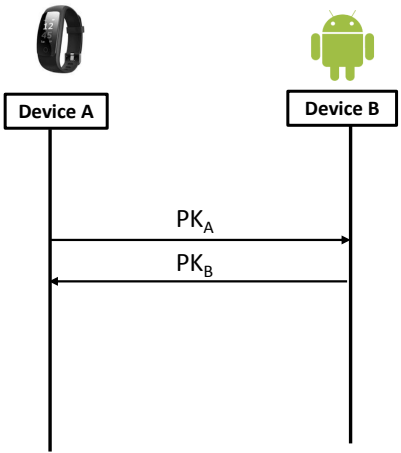
Workflow of Out of Band



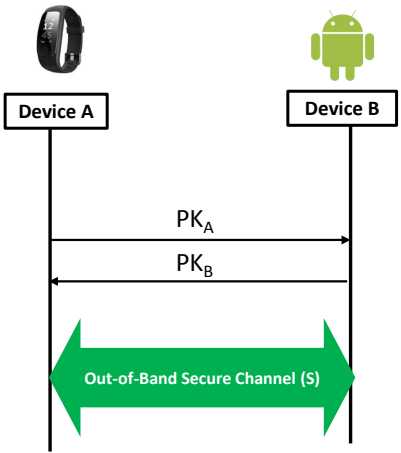
Workflow of Out of Band



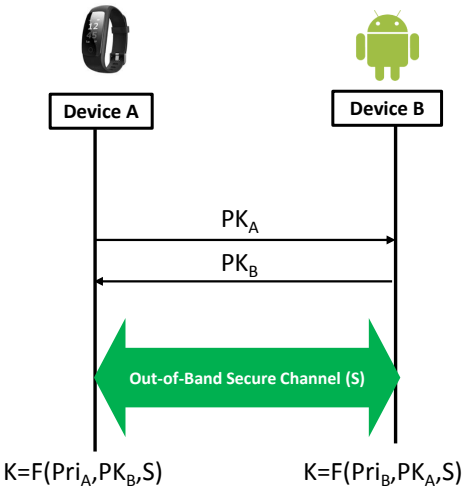
Workflow of Out of Band



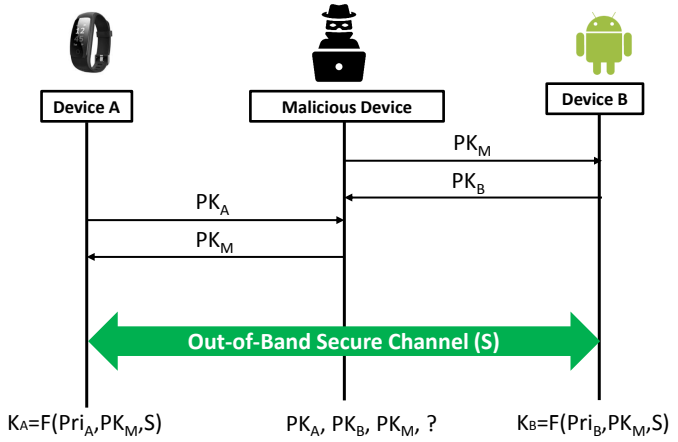
Workflow of Out of Band



Workflow of Out of Band



Workflow of Out of Band

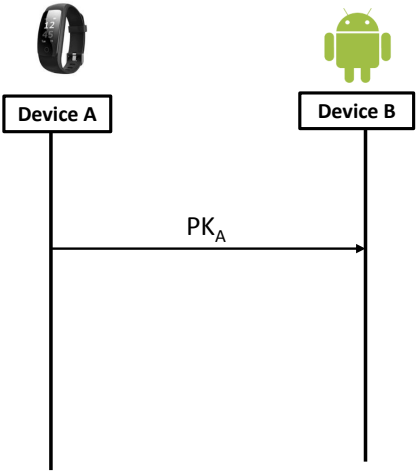


Workflow of Justworks

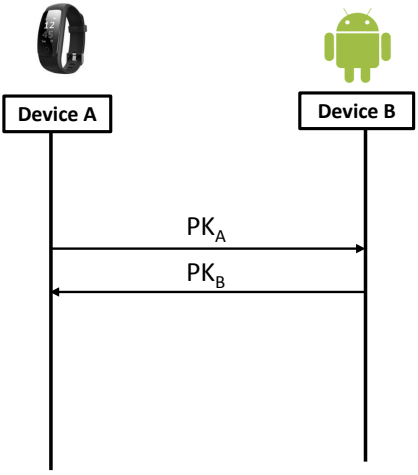
Device A

Device B

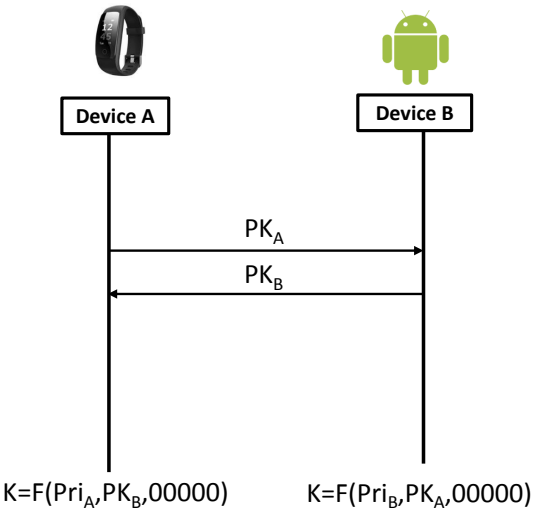
Workflow of Justworks



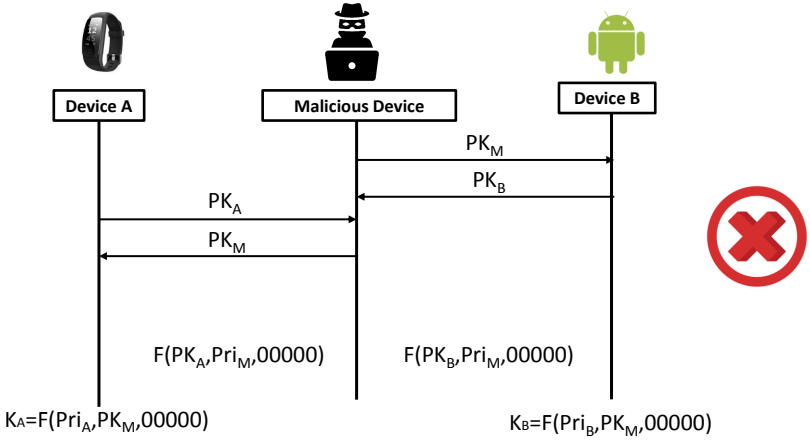
Workflow of Justworks



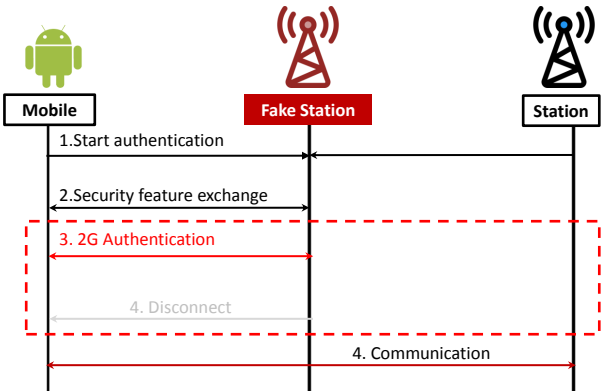
Workflow of Justworks



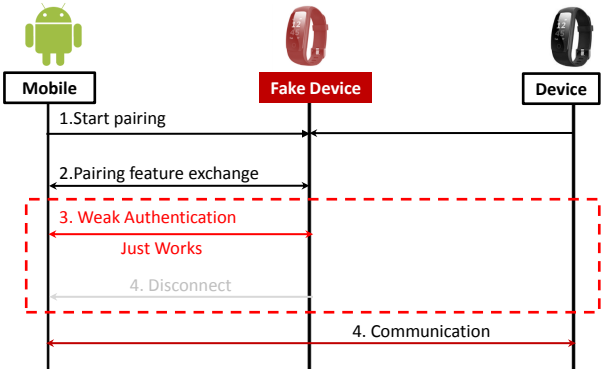
Workflow of Justworks



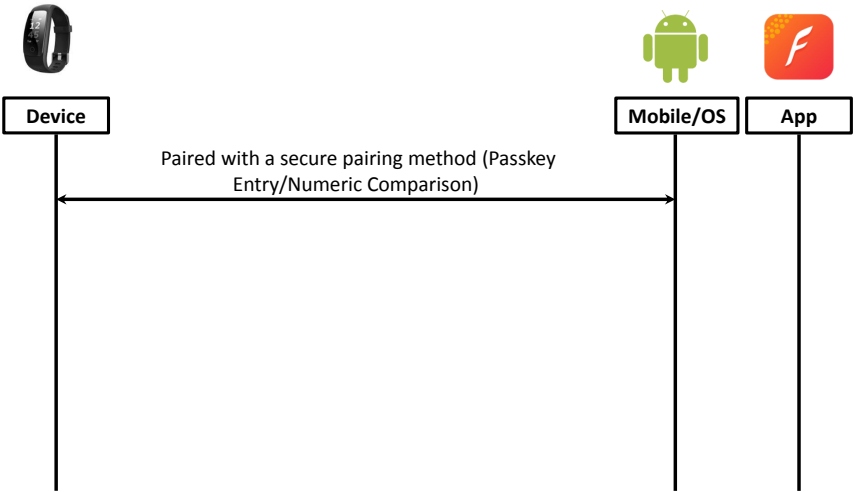
Our Downgrade Attacks against Bluetooth Low Energy



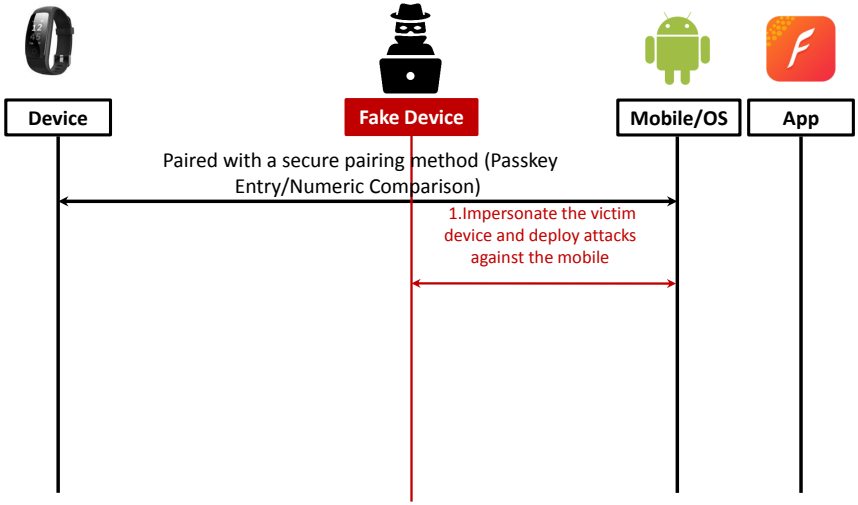
Our Downgrade Attacks against Bluetooth Low Energy



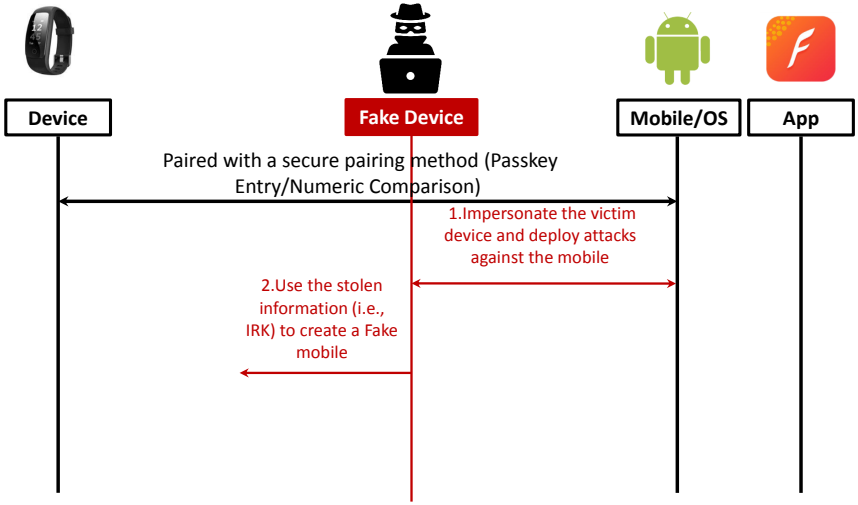
Our Downgrade Attacks against Bluetooth Low Energy



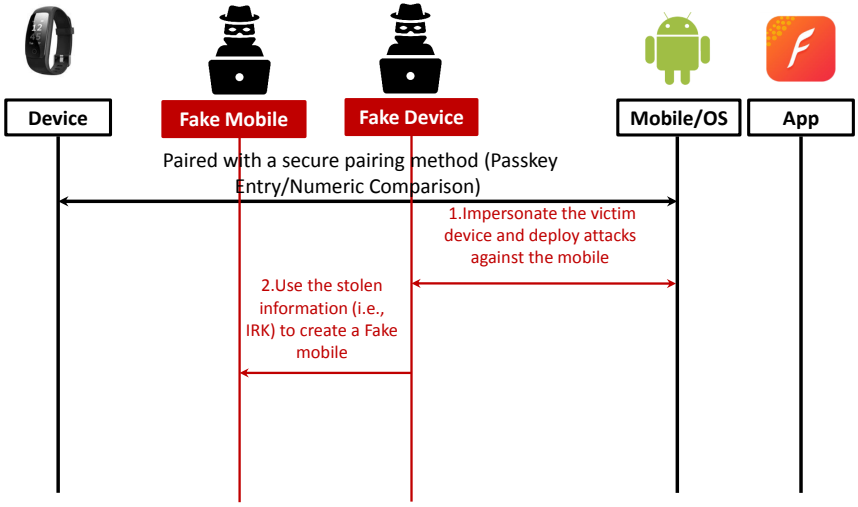
Our Downgrade Attacks against Bluetooth Low Energy



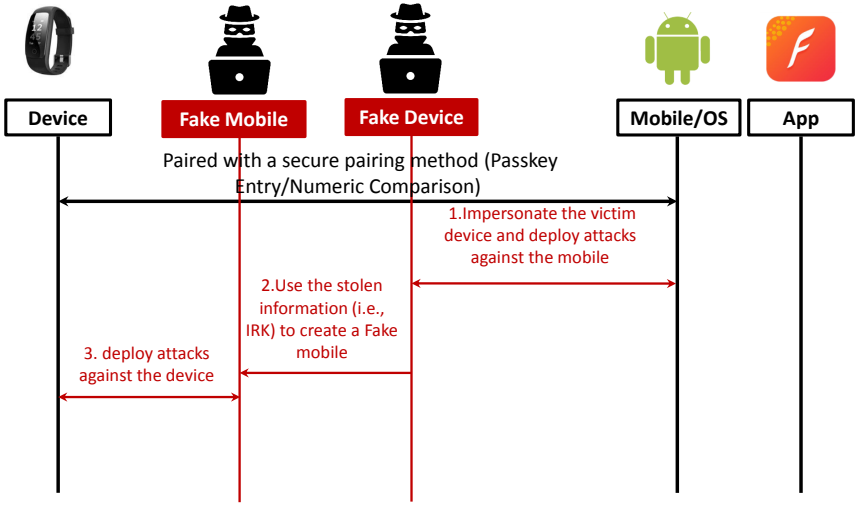
Our Downgrade Attacks against Bluetooth Low Energy



Our Downgrade Attacks against Bluetooth Low Energy

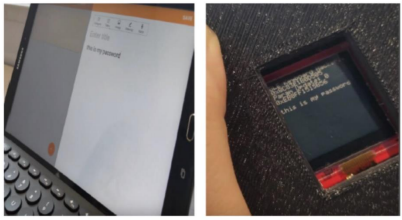


Our Downgrade Attacks against Bluetooth Low Energy



Our Downgrade Attacks against Bluetooth Low Energy

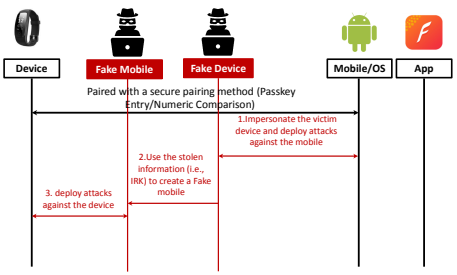
The Tested BLE devices



MITM attack against BLE keyboards

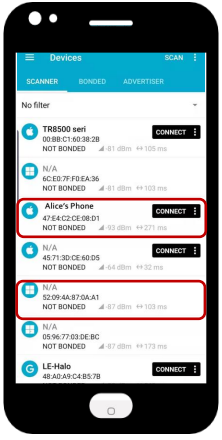
CVE-2020-9770

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

Bluetooth Sniffers



Alice's phone

Bob's phone

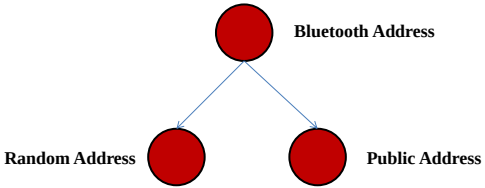
T1: 52:09:4A:87:0A:A1

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

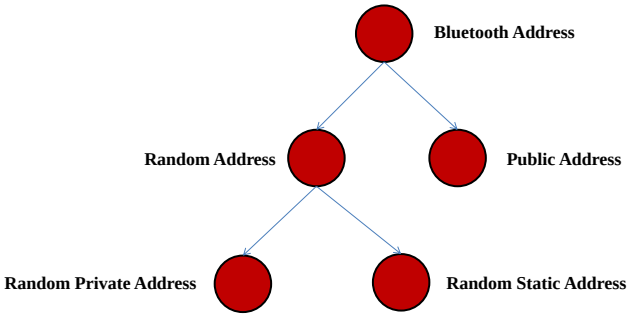
Bluetooth Address Types

Bluetooth Address

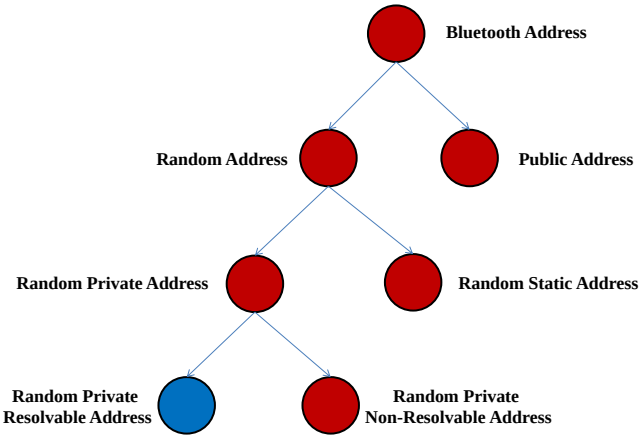
Bluetooth Address Types



Bluetooth Address Types



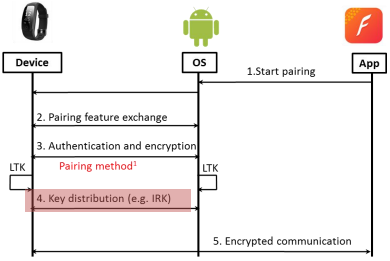
Bluetooth Address Types



(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)

(I) RPA Generation

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

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

Identity Resolving Key (irk_c)

(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)

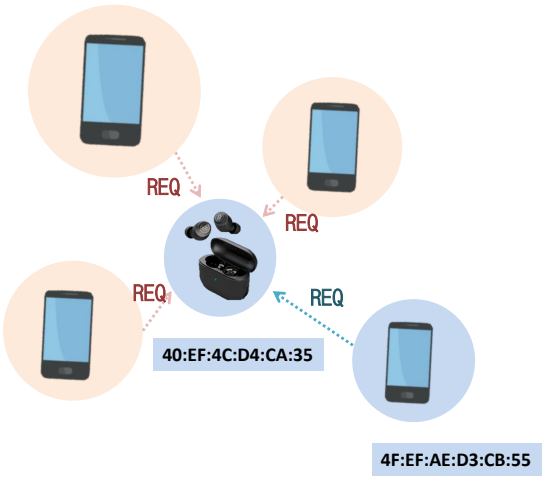
(II) RPA Resolution

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

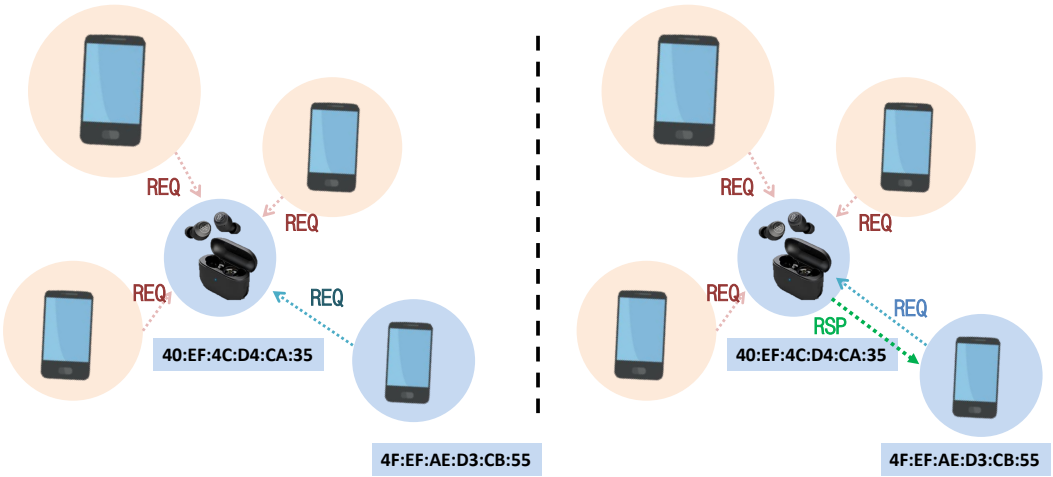
$$rpa_c = prand_{24} \,||\, H_{24}(Prand_{24}||irk_c)$$

$irk_p = irk_c \rightarrow rpa_p = rpa_c$

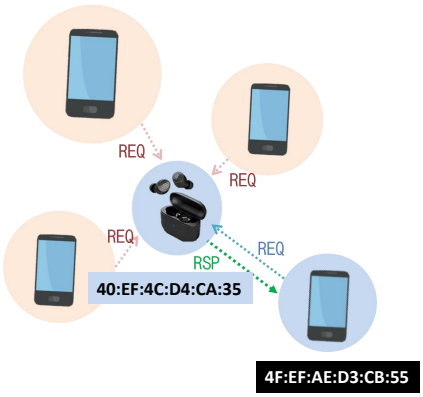
Our First Finding: Allowlist-based Side Channel



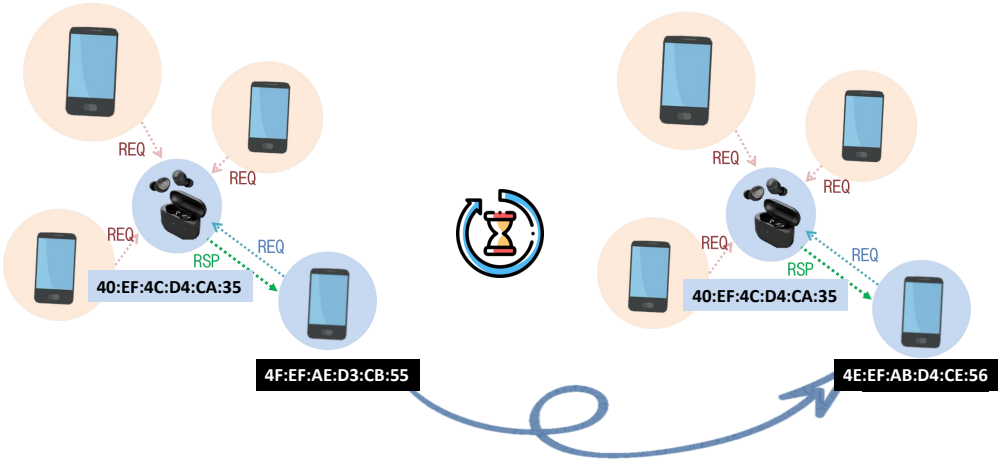
Our First Finding: Allowlist-based Side Channel



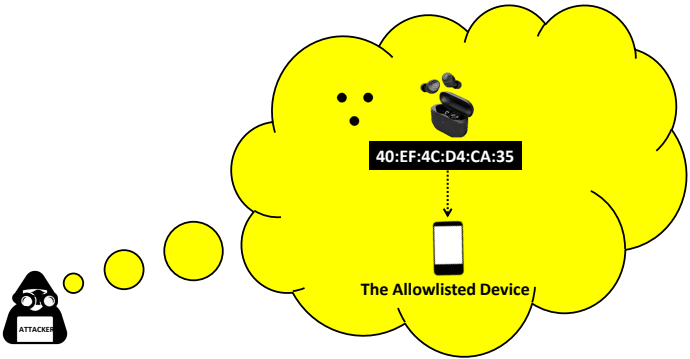
Our First Finding: Allowlist-based Side Channel



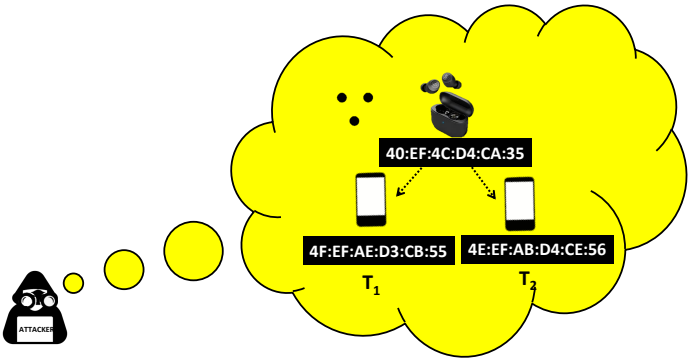
Our First Finding: Allowlist-based Side Channel



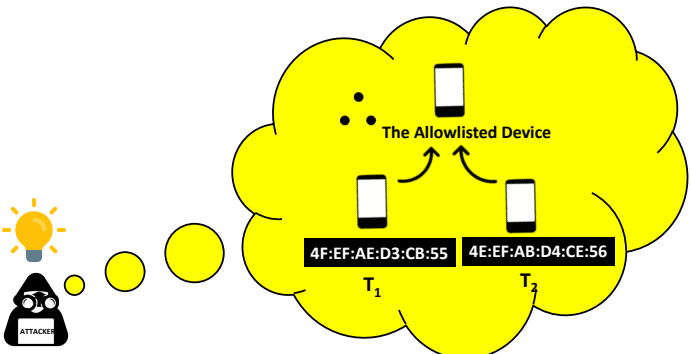
Our First Finding: Allowlist-based Side Channel



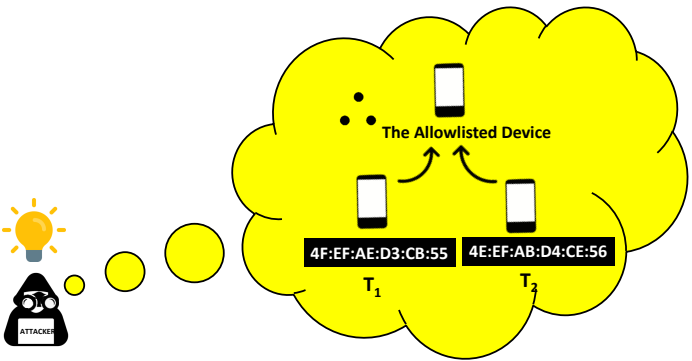
Our First Finding: Allowlist-based Side Channel



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

Identity Resolving Key (irk_p)

Identity Resolving Key (irk_c)

Our Second Finding: MAC Address Can be Replayed

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 Second Finding: MAC Address Can be Replayed

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 Second Finding: MAC Address Can be Replayed

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 Second Finding: MAC Address Can be Replayed

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)

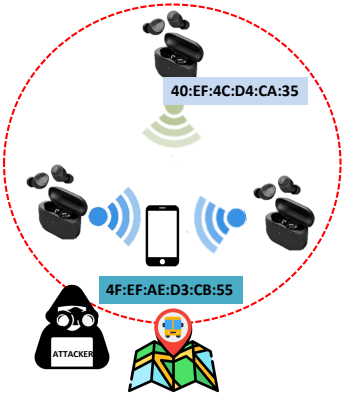
$$rpa_c = prand_{24} || H_{24}(Prand_{24} || irk_c)$$

$irk_p = irk_c$ ➔ $rpa_p = rpa_c$

rpa_p

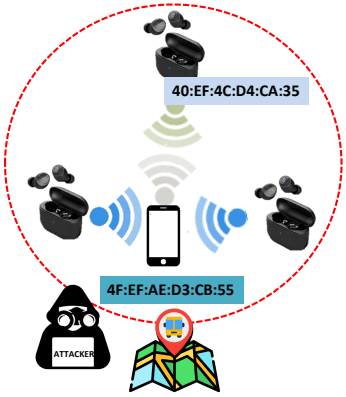
rpa'_p

Attack Example



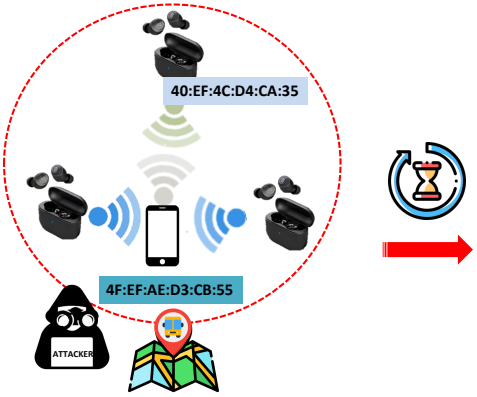
Tracking a Victim's Real-time Location

Attack Example



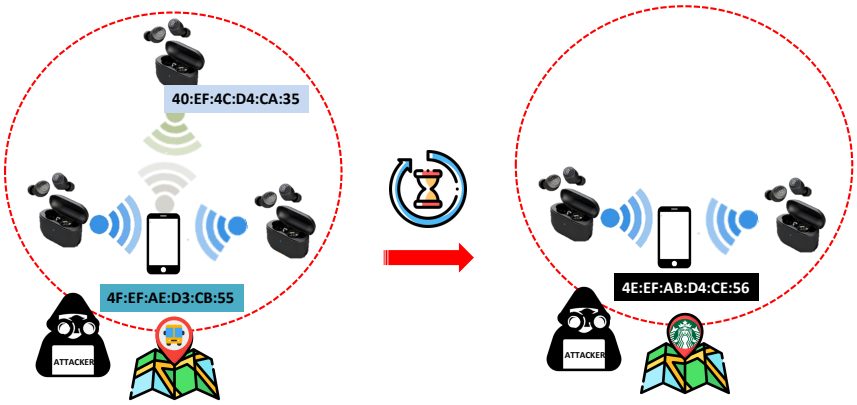
Tracking a Victim's Real-time Location

Attack Example



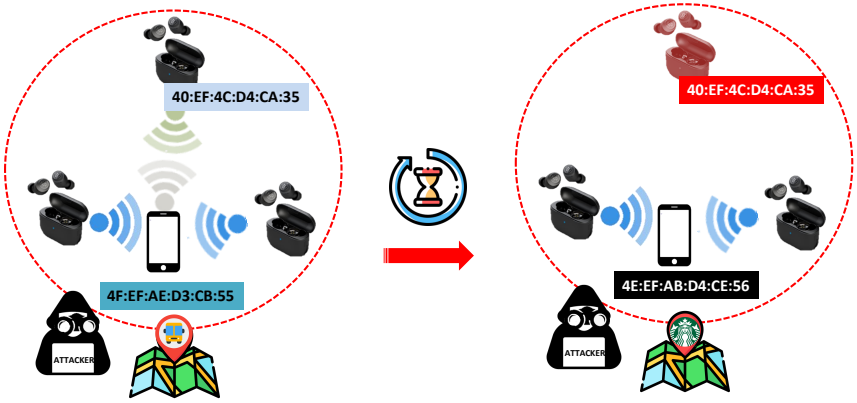
Tracking a Victim's Real-time Location

Attack Example



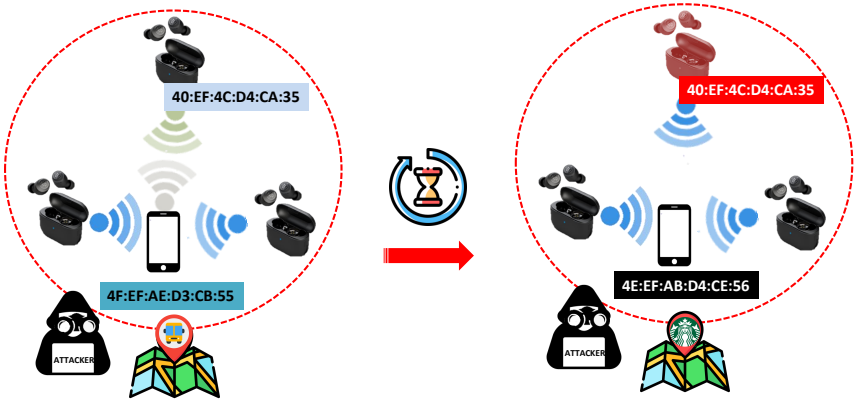
Tracking a Victim's Real-time Location

Attack Example



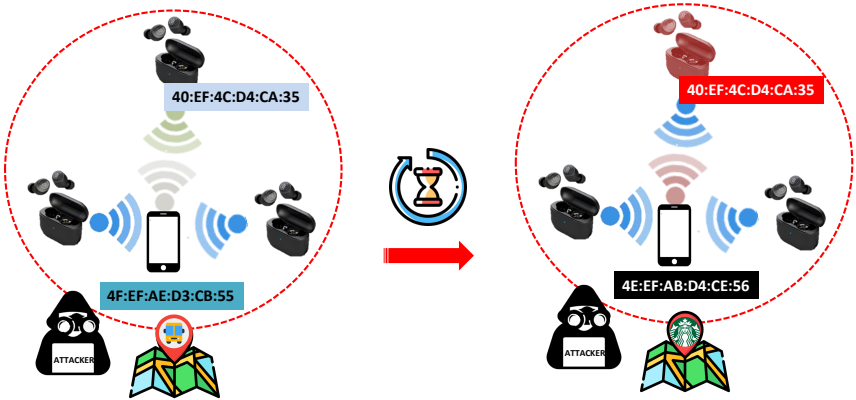
Tracking a Victim's Real-time Location

Attack Example



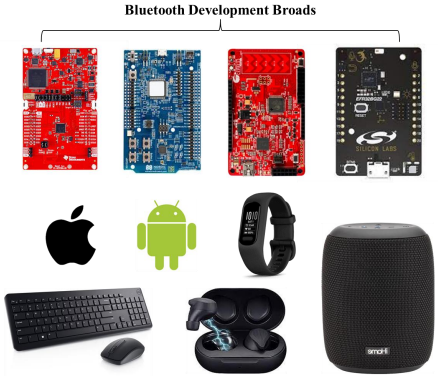
Tracking a Victim's Real-time Location

Attack Example



Tracking a Victim's Real-time Location

Devices That are Subject to BAT Attacks



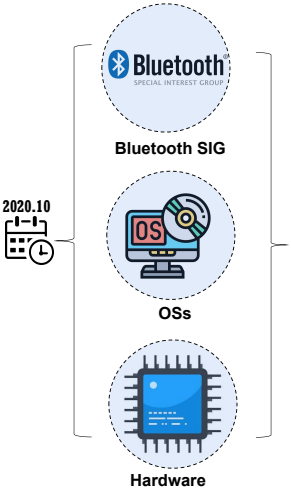
Peripherals & Development Boards

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
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	✓	✓	✓	✓	✓	✓	✓
Viqok	✓	✓	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	✓	✗	✓	✓	✓	✓	✓
Riitek	✗	✓	Keyboard	PA	✓	✗	✓	✗	✗	✓	✓
Cimetech	✗	✓	Mouse	SRA	✓	✗	✓	✗	✓	✓	✓
Ergonomic	✗	✓	Mouse	SRA	✓	✗	✓	✗	✓	✓	✓
TI CC2640R2F	✓	✓	Dev Board	RPA	-	✓	✓	✓	✓	✓	✓
Nordic NRF52	✓	✓	Dev Board	RPA	-	✓	✓	✓	✓	✓	✓
Silicon Labs 6101D	✗	✓	Dev Board	RPA	-	-	-	✗	✗	✓	✓
Cryptea CYBiCIT	✗	✓	Dev Board	RPA	-	-	-	✗	✗	✓	✓

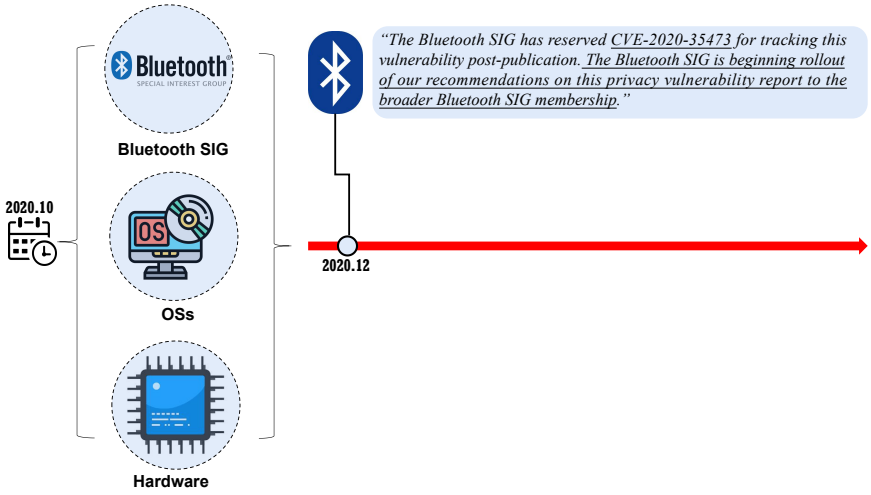
Centrals

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
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	$+\infty$	✓	✓	✓	✓	✓	✓

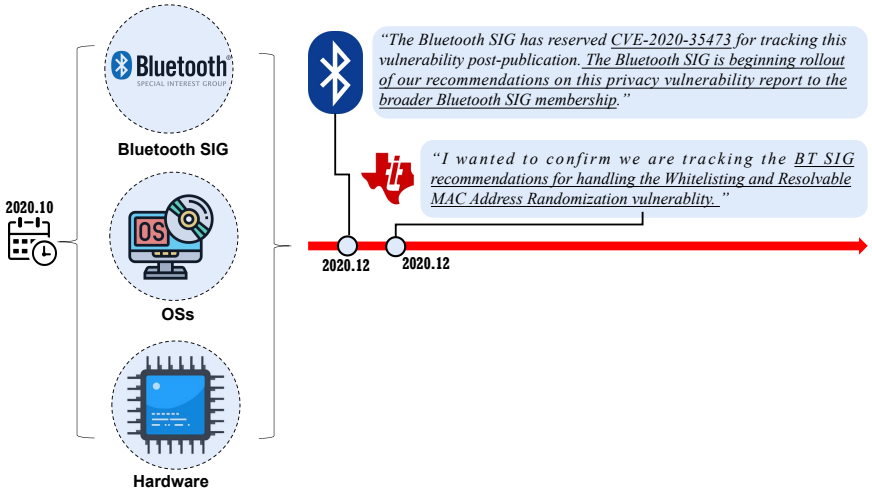
Responsible Disclosure



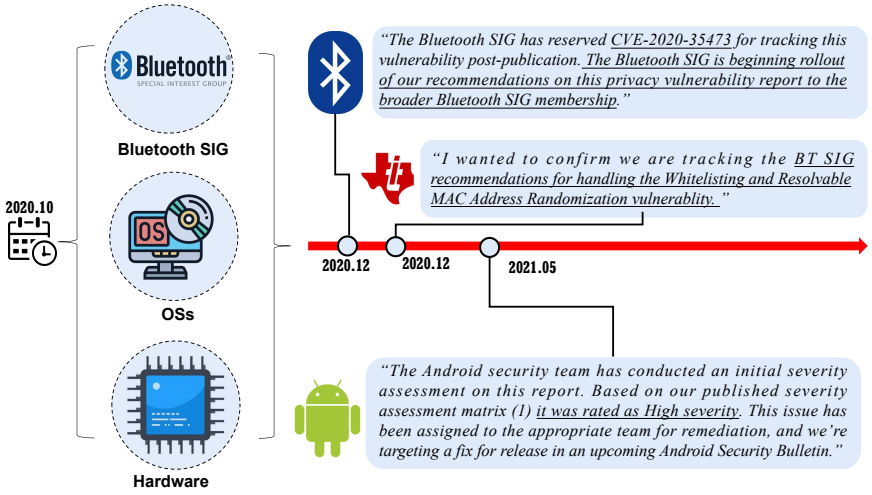
Responsible Disclosure



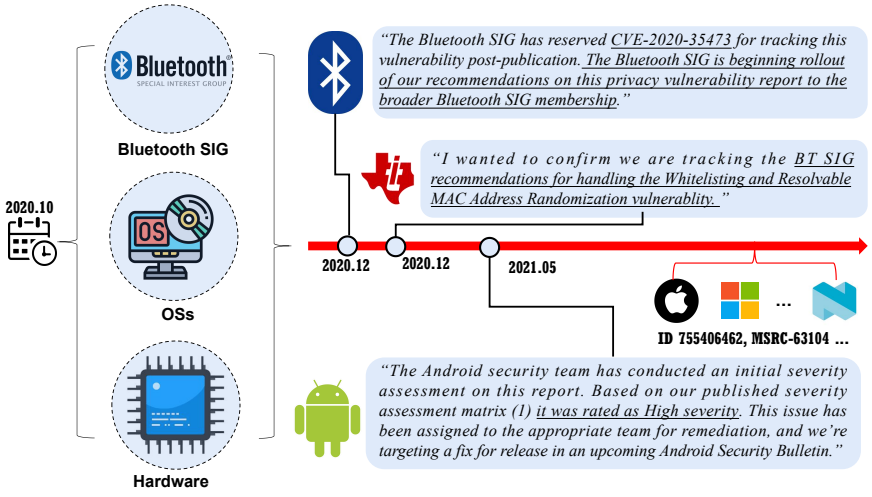
Responsible Disclosure



Responsible Disclosure



Responsible Disclosure



Our Countermeasure: Securing Address of BLE (SABLE)

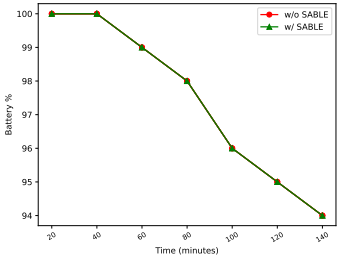
Allowlist Side Channel (Mitigation)

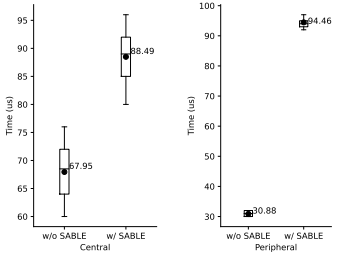
- We advocate the use of an interval unpredictable, central and peripheral synchronized RPA generation scheme to mitigate the side channel.

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.

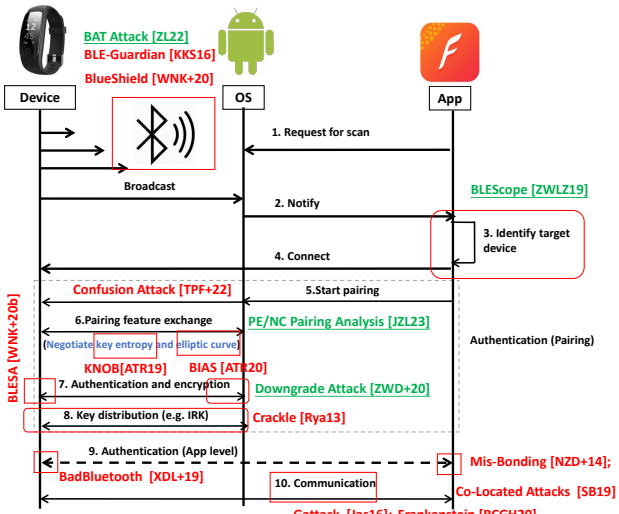
Our Countermeasure: Securing Address of BLE (SABLE)



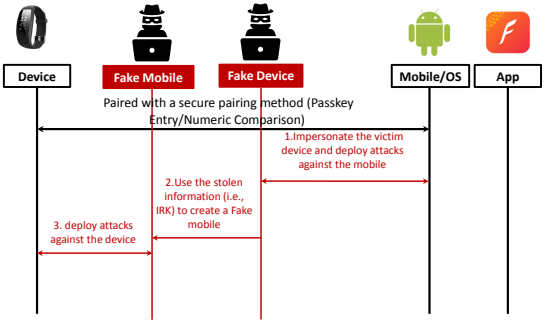


"When Good Becomes Evil: Tracking Bluetooth Low Energy Devices via Allowlist-based Side Channel and Its Countermeasure". Yue Zhang, and Zhiqiang Lin. *In Proceedings of the 29th ACM Conference on Computer and Communications Security (CCS 2022)*. November 2022 (**Best Paper Award Honorable Mention**)

Attacks and Defenses in Bluetooth Security and Privacy

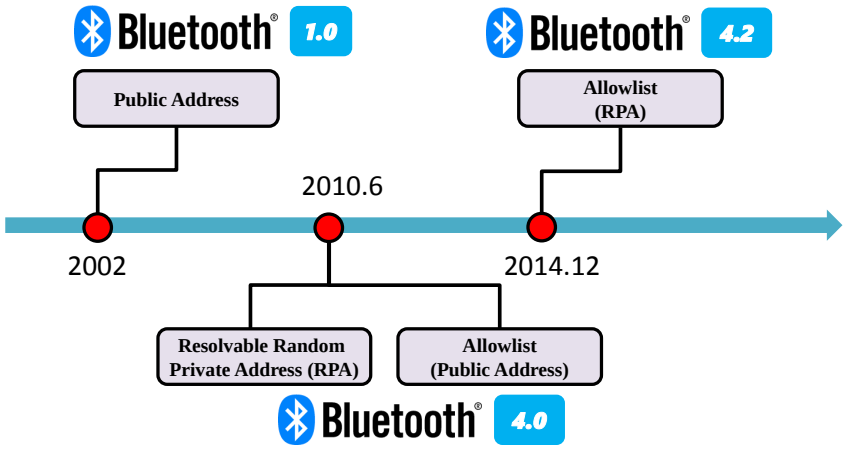


Lesson Learned (1/3): BLE Communication Can Be Downgraded

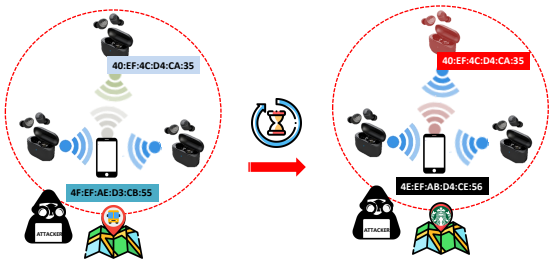


- Bluetooth low energy (BLE) pairing can be **downgraded**
- There are many stages that are not part of the pairing process, but they are, in fact, closely related to pairing security.
- A systematic analysis of the pairing process, including the **error handling** of BLE communication, is needed.

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

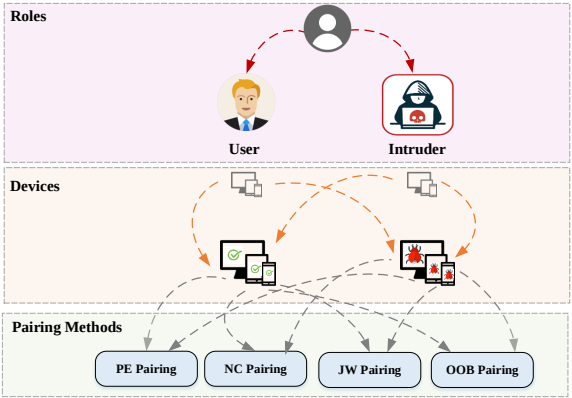


Lesson Learned (2/3): New Features Need Re-examinations



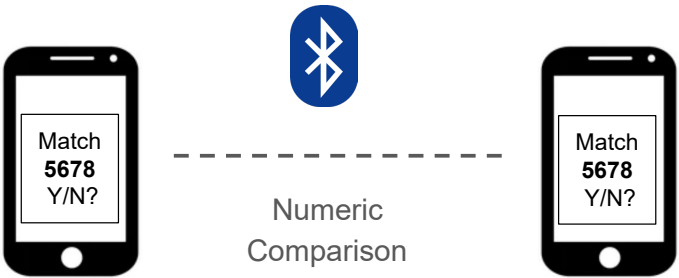
- BLE introduces multiple new features, some of which may **violate existing assumptions**
- Simliar to allowlist, those new features need to be **scrunitized**. For example, Cross-transport key derivation (CTKD); Authorization; The Connection Signature Resolving Key (CSRK).

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

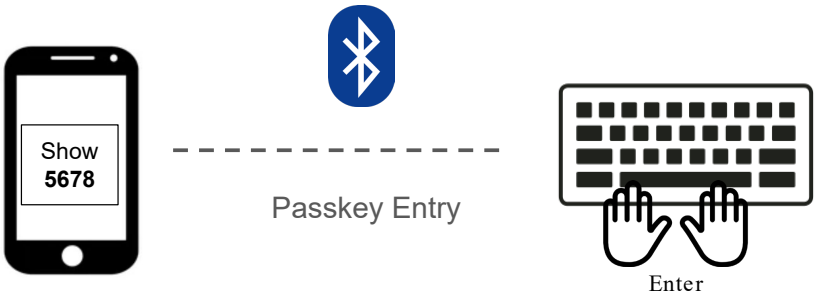


- The specification (3,000+ pages) is often confusing and inconsistent across chapters.
- The confusion may lead to different vendors implement BLE protocols in quite different ways, for example, for error handling, and IRK use.
- Converting the Bluetooth specification to formal model, and formally verify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

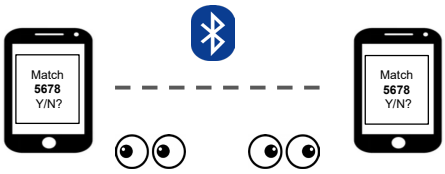
Using Protocol Verification to Identify Confusion Attacks [NDSS'23]



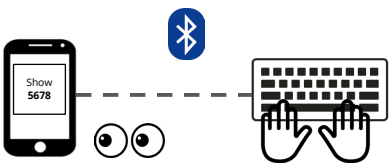
Using Protocol Verification to Identify Confusion Attacks [NDSS'23]



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

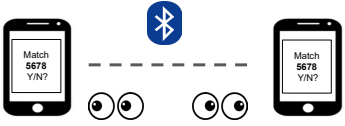


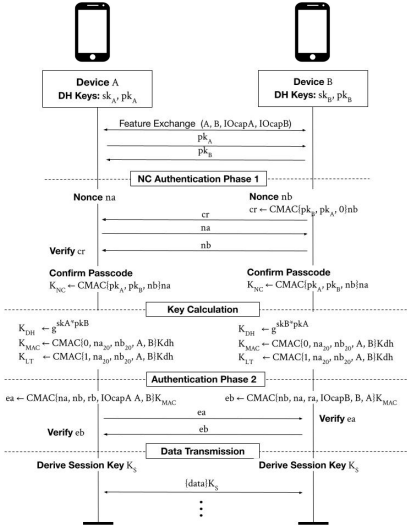
Numeric Comparison



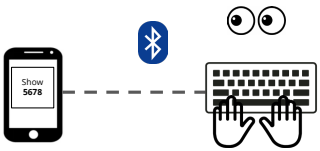
Passkey Entry

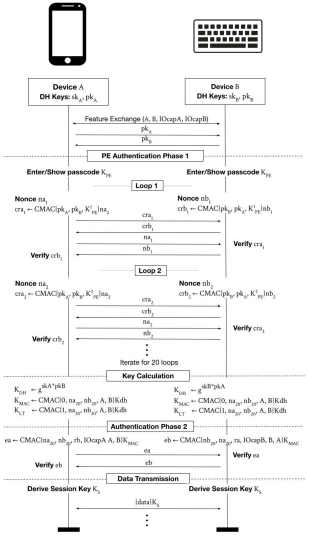
Using Protocol Verification to Identify Confusion Attacks [NDSS'23]



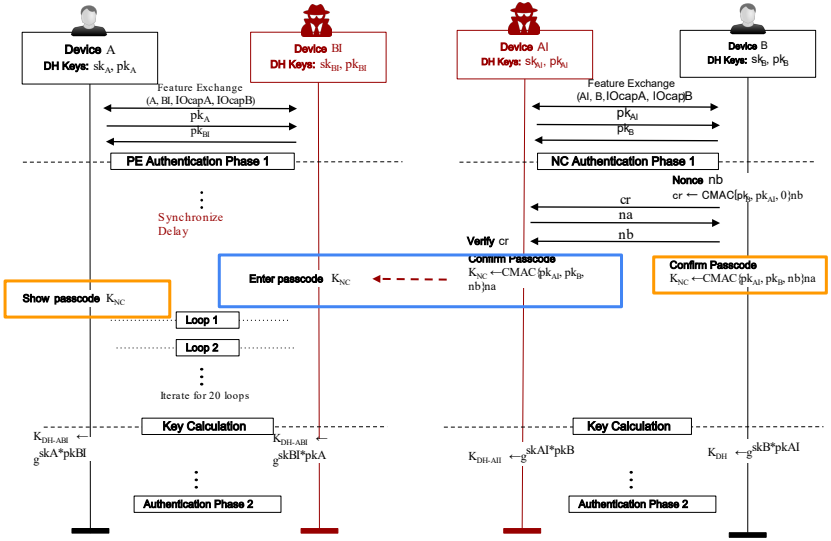


Using Protocol Verification to Identify Confusion Attacks [NDSS'23]

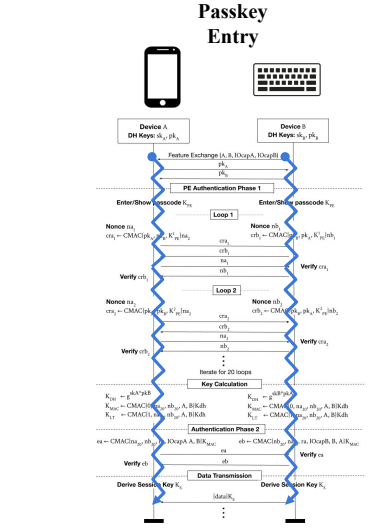


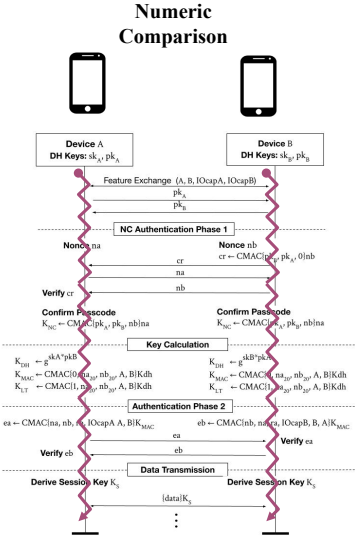


Using Protocol Verification to Identify Confusion Attacks [NDSS'23]

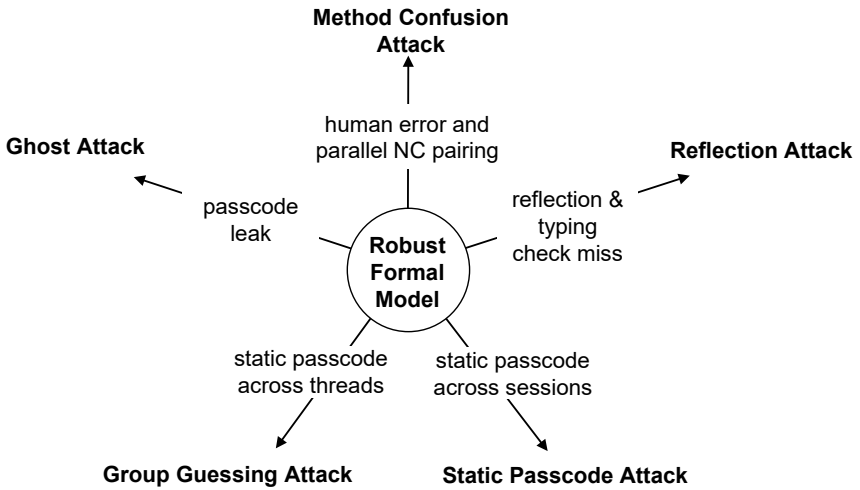


Using Protocol Verification to Identify Confusion Attacks [NDSS'23]





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



Acknowledgement

1. "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 (USENIX'20)*, Boston, MA. August 2020
2. "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'22)*. November 2022
3. "Extrapolating Formal Analysis to Uncover Attacks in Bluetooth Passkey Entry Pairing". **Mohit K. Jangid**, **Yue Zhang** and Zhiqiang Lin. *In Proceedings of the 30th ISOC Network and Distributed System Security Symposium (NDSS'23)*, San Diego, CA, April 2023.

Thank You

Rethinking the Security and Privacy of Bluetooth Low Energy

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

05/02/2024