Rethinking the Security and Privacy of Bluetooth Low Energy

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

10/14/2022
The evolution of our computing
The evolution of our computing

Pre-Internet

- ENIAC (1946)
- Mainframes (1950s)
- Multics (1964)
- APRAnet (1969)
- Email (1971)
- PC (1981)
The evolution of our computing
The evolution of our computing

Pre-Internet
- ENIAC 1946
- Mainframes 1950s
- Multics 1964
- ARPAnet 1969
- Email 1971
- PC 1981
- TCP/IP & Internet 1982
- Netscape 1994
- Windows 1995

Internet
- ENIAC 1946
- Mainframes 1950s
- Multics 1964
- ARPAnet 1969
- Email 1971
- PC 1981
- TCP/IP & Internet 1982
- WWW 1989
- Netscape 1994
- Windows 1995

Trusted
The evolution of our computing
The evolution of our computing

**Pre-Internet**
- ENIAC 1946
- Mainframes 1950s
- Multics 1964
- APRAnet 1969
- Email 1971
- PC 1981
- TCP/IP & Internet 1982

**Internet**
- APRAnet 1969
- Email 1971
- PC 1981
- TCP/IP & Internet 1982
- WWW 1989
- Netscape 1994
- Windows 1995
- Netscape 1994
- Windows 1995
- WiFi 1999
- 3G/4G 2000s
- Android 2005
- iPhone 2007
- Mainframes 1950s
- Multics 1964
- Email 1971
- PC 1981
- TCP/IP & Internet 1982

**Mobile**
- ENIAC 1946
- Mainframes 1950s
- Multics 1964
- Email 1971
- PC 1981
- TCP/IP & Internet 1982
- APRAnet 1969
- Email 1971
- PC 1981
- WWW 1989
- Netscape 1994
- Windows 1995
- WiFi 1999
- 3G/4G 2000s
- Android 2005
- iPhone 2007

**Trusted**
The evolution of our computing

Pre-Internet
- ENIAC 1946
- Mainframes 1950s
- Multics 1964
- APRAnet 1969
- Email 1971
- PC 1981

Internet
- TCP/IP & Internet 1982
- WWW 1989
- Netscape 1994
- Windows 1995
- Wi-Fi 1999
- 3G/4G 2000s

Mobile
- Android 2005
- iPhone 2007

Trusted PC
- Windows 1995
- WiFi 1999
- 3G/4G 2000s
- Android 2005
- iPhone 2007
The evolution of our computing

Pre-Internet
- ENIAC 1946
- Mainframes 1950s
- Multics 1964
- ARPAnet 1969
- Email 1971
- PC 1981

Internet
- TCP/IP & Internet 1982
- WWW 1989
- Netscape 1994
- Windows 1995
- WiFi 1999
- 3G/4G 2000s
- Android 2005
- iPhone 2007

Mobile
- IoT 2010s
- Nest Thermostat 2011
- Tesla Model S 2012
- Amazon Alexa 2014

Trusted PC
1981
Windows
1995
WiFi
1999
iPhone
2007
Android
2005
Pre-Internet
 Internet Mobile IoT
ENIAC
1946
TCP/IP & Internet
1982
WWW
1989
Netscape
1994
Windows
1995
WiFi
1999
3G/4G
2000s
Android
2005
iPhone
2007
IoT
2010s
Nest Thermostat
2011
Tesla Model S
2012
Amazon Alexa
2014
The evolution of our computing

Pre-Internet
- ENIAC 1946
- Mainframes 1950s
- Multics 1964
- ARPAnet 1969
- Email 1971
- PC 1981

Internet
- TCP/IP & Internet 1982
- WWW 1989
- Netscape 1994
- Windows 1995
- WiFi 1999
- 3G/4G 2000s
- Android 2005
- iPhone 2007

Mobile
- IoT 2010s
- Nest Thermostat 2011
- Tesla Model S 2012
- Amazon Alexa 2014

IoT
- Tesla Model S 2012
- Amazon Alexa 2014
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
Our Recent Works in IoT Security and Privacy

1. Automatic Uncovering of Hidden Behaviors From Input Validation in Mobile Apps. In S&P 2020
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
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
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
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
10. Extrapolating Formal Analysis to Uncover Attacks in Bluetooth Passkey Entry Pairing. In NDSS 2023
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

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

1994: Bluetooth Prototype
1998: Bluetooth SIG
1999: Bluetooth 1.0+1.0b
2001: Bluetooth 1.1 (~700KB/s)
2004: Bluetooth 2.0/2.1 +EDR (~2.1MB/s)
2009: Bluetooth 3.0+HS (~24MB/s)
2010: Bluetooth 4.0 (~24MB/s)
2013: Bluetooth 4.1 (~24MB/s)
2014: Bluetooth 4.2
2016: Bluetooth 5.0 (~50MB/s)
2019: Bluetooth 5.1/5.2 (~50MB/s)
Total Annual Bluetooth Device Shipments

Total Annual Bluetooth® Device Shipments

NUMBERS IN BILLIONS

<table>
<thead>
<tr>
<th>Year</th>
<th>Shipments</th>
</tr>
</thead>
<tbody>
<tr>
<td>2017</td>
<td>3.6</td>
</tr>
<tr>
<td>2018</td>
<td>3.8</td>
</tr>
<tr>
<td>2019</td>
<td>4.1</td>
</tr>
<tr>
<td>2020</td>
<td>4.1</td>
</tr>
<tr>
<td>2021</td>
<td>4.7</td>
</tr>
<tr>
<td>2022</td>
<td>5.1</td>
</tr>
<tr>
<td>2023</td>
<td>5.5</td>
</tr>
<tr>
<td>2024</td>
<td>6.0</td>
</tr>
<tr>
<td>2025</td>
<td>6.5</td>
</tr>
<tr>
<td>2026</td>
<td>7.0</td>
</tr>
</tbody>
</table>

9% CAGR

Data Source: ABI Research, 2022
Total Annual Bluetooth Device Shipments
Total Annual Bluetooth Device Shipments
Total Annual Bluetooth Device Shipments

Annual Bluetooth® Audio Streaming Device Shipments
- Numbers in billions
- 7% CAGR

Annual Bluetooth® Location Services Device Shipments
- Numbers in millions
- 1.80 billion shipments
- 25% CAGR

2022 Bluetooth® Smart Home Device Shipments
- 568 million shipments

Smart Appliances 38%
Smart Lighting 38%
Smoke and Air Quality Sensors 4%
Smart Blinds 4%
Contactless Sensors 4%
Other 8%
The General Workflow of Device Communication in TCP/IP Setting

Device

OS

App
The General Workflow of Device Communication in TCP/IP Setting

1. Listen to port 443
The General Workflow of Device Communication in TCP/IP Setting

1. Listen to port 443
2. <Request, 192.168.1.1, port 443>
The General Workflow of Device Communication in TCP/IP Setting

1. Listen to port 443

2. <Request, 192.168.1.1, port 443>

3. Connect
The General Workflow of Device Communication in TCP/IP Setting

1. Listen to port 443
2. <Request, 192.168.1.1, port 443>
3. Connect
4. Authentication (Transport Layer Security / Secure Sockets Layer)
The General Workflow of Device Communication in TCP/IP Setting

1. Listen to port 443
2. <Request, 192.168.1.1, port 443>
3. Connect
4. Authentication (Transport Layer Security / Secure Sockets Layer)
5. Communication
The General Workflow of Device Communication in TCP/IP Setting
The General Workflow of BLE IoT Devices and Companion Apps

10 / 33
The General Workflow of BLE IoT Devices and Companion Apps

1. Request for scan
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

1. Request for scan
2. Notify
3. Identify target device
4. Connect
5. Start pairing
The General Workflow of BLE IoT Devices and Companion Apps

1. Request for scan
2. Notify
3. Identify target device
4. Connect
5. Start pairing
6. Pairing
The General Workflow of BLE IoT Devices and Companion Apps

1. Request for scan
2. Notify
3. Identify target device
4. Connect
5. Start pairing
6. Pairing
7. Communication
Outline

1. Introduction
2. Background
3. BLE Security
4. BLE Privacy
5. Takeaway
Pairing Workflow

- Device
- OS
- App
Pairing Workflow

1. Start pairing

Device  →  OS  ←  App
Pairing Workflow

1. Start pairing

2. Pairing feature exchange

I/O Features
- Keypad
- Screen
- Out of band Channel
Pairing Workflow

1. Start pairing

2. Pairing feature exchange

3. Authentication and encryption

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

1. Start pairing
2. Pairing feature exchange
3. Authentication and encryption
4. Key distribution (e.g., IRK)
5. Encrypted communication

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

Pairing method\(^1\)
Workflow of Pairing: Elliptic Curve Diffie–Hellman (ECDH) Key Exchange

1. Alice generates a random ECC key pair: \( \{Pri_A, PK_A = Pri_A \times G\} \)
Workflow of Pairing: Elliptic Curve Diffie–Hellman (ECDH) Key Exchange

1. Alice generates a random ECC key pair: \( \{Pri_A, PK_A = Pri_A * G\} \)
2. Bob generates a random ECC key pair: \( \{Pri_B, PK_B = Pri_B * G\} \)
Workflow of Pairing: Elliptic Curve Diffie–Hellman (ECDH) Key Exchange

1. Alice generates a random ECC key pair: \( \{ \text{Pri}_A, \text{PK}_A = \text{Pri}_A \times G \} \)
2. Bob generates a random ECC key pair: \( \{ \text{Pri}_B, \text{PK}_B = \text{Pri}_B \times G \} \)
3. Alice and Bob exchanges \( \text{PK}_A \) and \( \text{PK}_B \)
Workflow of Pairing: Elliptic Curve Diffie–Hellman (ECDH) Key Exchange

1. Alice generates a random ECC key pair: \( \{Pri_A, PK_A = Pri_A \ast G\} \)
2. Bob generates a random ECC key pair: \( \{Pri_B, PK_B = Pri_B \ast G\} \)
3. Alice and Bob exchanges \( PK_A \) and \( PK_B \)
4. Alice calculates sharedKey: \( K_A = Pri_A \ast PK_B \)
Workflow of Pairing: Elliptic Curve Diffie–Hellman (ECDH) Key Exchange

1. Alice generates a random ECC key pair: \( \{ \text{Pri}_A, \text{PK}_A = \text{Pri}_A \ast G \} \)
2. Bob generates a random ECC key pair: \( \{ \text{Pri}_B, \text{PK}_B = \text{Pri}_B \ast G \} \)
3. Alice and Bob exchanges \( \text{PK}_A \) and \( \text{PK}_B \)
4. Alice calculates sharedKey: \( K_A = \text{Pri}_A \ast \text{PK}_B \)
5. Bob calculates sharedKey: \( K_B = \text{Pri}_B \ast \text{PK}_A \)
Workflow of Pairing: Elliptic Curve Diffie–Hellman (ECDH) Key Exchange

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

\[
Pri_A \times (Pri_B \times G) = Pri_B \times (Pri_A \times G)
\]
Workflow of Passkey Entry
Workflow of Passkey Entry
Workflow of Passkey Entry
Workflow of Passkey Entry
Workflow of Passkey Entry

Device A
123456
PK_A

Device B
PK_B
123456

F(PK_A, PK_B, 123456) F(PK_A, PK_B, 123456)
Workflow of Passkey Entry

Device A

- $PK_A$
- $123456$

$PK_A = F(PK_A, PK_M, 123456)$

Malicious Device

- $PK_M$
- $PK_B$

Device B

- $123456$

$K_B = F(PK_B, PK_M, 123456)$
Workflow of Numeric Comparison

Device A

Device B

Select An Accessory

Bluetooth Pairing Request

123456

Cancel  Pair

Select An Accessory

Bluetooth Pairing Request

123456

Cancel  Pair

Cancel
Workflow of Numeric Comparison

Device A

PK_A

PK_B

Device B

Pairing request on Device A:

Bluetooth Pairing Request

"[Device B]" would like to pair with your device. Confirm that this code is displayed on "[Device B]."

123456

[Cancel] [Pair]

Pairing request on Device B:

Bluetooth Pairing Request

"[Device B]" would like to pair with your device. Confirm that this code is displayed on "[Device A]."

123456

[Cancel] [Pair]
Workflow of Numeric Comparison

Device A

PK_A

Hash(PK_A, PK_B)

123456

Device B

PK_B

Hash(PK_A, PK_B)

123456
Workflow of Numeric Comparison

Device A

PK_A

PK_B

Hash(PK_A, PK_B)

123456

Device B

PK_B

PK_A

Hash(PK_A, PK_B)

123456

123456

123456
Workflow of Numeric Comparison

Device A

PK_A

PK_B

Hash(PK_A,PK_B)

123456

F(PK_A,PK_B)

Device B

PK_B

PK_A

Hash(PK_A,PK_B)

123456

F(PK_A,PK_B)
Workflow of Numeric Comparison

**Device A**
- PK\(_A\)
- PK\(_M\)
- Hash(PK\(_A\), PK\(_M\))
- 123456
- KA = F(PK\(_A\), PK\(_M\))

**Malicious Device**
- PK\(_B\)
- PK\(_M\)
- PK\(_A\)
- PK\(_M\)

**Device B**
- PK\(_B\)
- PK\(_M\)
- Hash(PK\(_B\), PK\(_M\))
- 654321
- KB = F(PK\(_B\), PK\(_M\))

**Equations**

\[ K_A = F(PK_A, PK_M) \]
\[ K_B = F(PK_B, PK_M) \]
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

Device A

PK_A

PK_B

Device B

Out-of-Band Secure Channel (S)

K=F(PK_A,PK_B,S)

K=F(PK_B,PK_A,S)
Workflow of Out of Band

\[ K_A = F(\text{PK}_A, \text{PK}_M, S) \]

\[ K_B = F(\text{PK}_B, \text{PK}_M, S) \]
Workflow of Justworks

Device A

Device B
Workflow of Justworks

- Device A
- Device B
- \( P_{KA} \)
Workflow of Justworks

Device A

Device B

PK_A

PK_B
Workflow of Justworks

Device A

PK_A

PK_B

Device B

K = F(PK_A, PK_M, 00000)

K = F(PK_B, PK_M, 000000)
Workflow of Justworks

\[ K_A = F(PK_A, PK_M, 00000) \]

\[ K_I = F(PK_B, PK_M, 00000) \]
Our Downgrade Attacks against Bluetooth Low Energy
Our Downgrade Attacks against Bluetooth Low Energy

1. Start pairing
2. Pairing feature exchange
3. Weak Authentication
   - Just Works
4. Disconnect
4. Communication
Our Downgrade Attacks against Bluetooth Low Energy

Paired with a secure pairing method (Passkey Entry/Numeric Comparison)
Our Downgrade Attacks against Bluetooth Low Energy

1. Impersonate the victim device and deploy attacks against the mobile

Paired with a secure pairing method (Passkey Entry/Numeric Comparison)
Our Downgrade Attacks against Bluetooth Low Energy

1. Impersonate the victim device and deploy attacks against the mobile.
2. Use the stolen information (i.e., IRK) to create a Fake mobile.

Paired with a secure pairing method (Passkey Entry/Numeric Comparison)
Our Downgrade Attacks against Bluetooth Low Energy

1. Impersonate the victim device and deploy attacks against the mobile
2. Use the stolen information (i.e., IRK) to create a Fake mobile

Paired with a secure pairing method (Passkey Entry/Numeric Comparison)
Our Downgrade Attacks against Bluetooth Low Energy

1. Impersonate the victim device and deploy attacks against the mobile
2. Use the stolen information (i.e., IRK) to create a Fake mobile
3. Deploy attacks against the device

Paired with a secure pairing method (Passkey Entry/Numeric Comparison)
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

Outline

1. Introduction
2. Background
3. BLE Security
4. BLE Privacy
5. Takeaway
# Bluetooth Sniffers

<table>
<thead>
<tr>
<th>Bluetooth Sniffer</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ubertooth One Sniffer</td>
<td>125 USD</td>
</tr>
<tr>
<td>Adafruit LE sniffer</td>
<td>25 USD</td>
</tr>
</tbody>
</table>
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 Types

Bluetooth Address

Random Address

Public Address
Bluetooth Address Types

- Bluetooth Address
  - Random Address
    - Random Private Address
    - Random Static Address
  - Public Address
Bluetooth Address Types

- Bluetooth Address
  - Public Address
  - Random Address
  - Random Static Address
  - Random Private Address
  - Random Private Non-Resolvable Address
  - Random Private Resolvable Address
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 ($irk_p$)

(I) RPA Generation

Identity Resolving Key ($irk_c$)
How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key** \((irk_p)\)

\[(I) \text{ RPA Generation} \]

\[rpa_p = prand_{24} \parallel H_{24}(Prand_{24} | irk_p)\]

**Identity Resolving Key** \((irk_c)\)
How to Avoid Being Tracked: MAC Address Randomization

_identity_resolving_key_{irk_p}_

(I) RPA Generation

\[ rpa_p = \text{Prand}_{24} \oplus H_{24}(\text{Prand}_{24} || irk_p) \]

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2 bits)</td>
<td>0x00...3 (22 bits)</td>
<td>0x00...04 (24 bits)</td>
</tr>
</tbody>
</table>
How to Avoid Being Tracked: MAC Address Randomization

Identity Resolving Key \((irk_p)\)

\[
\text{RPA Generation} \\
\text{Type rand Hash} \\
01 (2\text{bits}) 0x00...3 (22\text{bits}) 0x00...04 (24\text{bits})
\]

Identity Resolving Key \((irk_c)\)
How to Avoid Being Tracked: MAC Address Randomization

Identity Resolving Key ($irk_p$)

(I) RPA Generation

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

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00…3 (22bits)</td>
<td>0x00…04 (24bits)</td>
</tr>
</tbody>
</table>

Identity Resolving Key ($irk_c$)

(II) RPA Resolution
How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key (irkₚ)**

(I) RPA Generation

\[ rpaₚ = \text{prand}_{24} || H_{24}(\text{prand}_{24}||\text{irkₚ}) \]

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>

(II) RPA Resolution

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>
How to Avoid Being Tracked: MAC Address Randomization

**Identity Resolving Key** \((irkp)\)

(\textbf{I}) \textbf{RPA Generation}

\[
\text{rpa}_p = \text{prand}_{24}||H_{24}(\text{prand}_{24}||irkp)
\]

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>

(\textbf{II}) \textbf{RPA Resolution}

\[
\text{rpa}_c = \text{prand}_{24}||H_{24}(\text{prand}_{24}||irkc)
\]

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>
How to Avoid Being Tracked: MAC Address Randomization

Identity Resolving Key ($irk_p$)

(I) RPA Generation

$$rpa_p = \text{rand}_{24} \| H_{24}(\text{Prand}_{24}||irk_p)$$

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0x00...3</td>
<td>0x00...04</td>
</tr>
</tbody>
</table>

Identity Resolving Key ($irk_c$)

(II) RPA Resolution

$$rpa_c = \text{rand}_{24} \| H_{24}(\text{Prand}_{24}||irk_c)$$

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0x00...3</td>
<td>0x00...04</td>
</tr>
</tbody>
</table>

$irk_p = irk_c \rightarrow rpa_p = rpa_c$
Our Discovery I — Allowlist-based Side Channel

<table>
<thead>
<tr>
<th>NO.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00:00:04</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>ADV_IND</td>
</tr>
</tbody>
</table>
## Our Discovery I — Allowlist-based Side Channel

<table>
<thead>
<tr>
<th>NO.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00:00:04</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>ADV_IND</td>
</tr>
<tr>
<td>2</td>
<td>00:00:08</td>
<td>7e:D7:8E:C7:8e:51</td>
<td>58:D7:8E:C7:8e:31</td>
<td>SCAN_REQ</td>
</tr>
<tr>
<td>3</td>
<td>00:00:12</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>SCAN_RSP</td>
</tr>
</tbody>
</table>

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

7e:D7:8E:C7:8e:51
Our Discovery I — Allowlist-based Side Channel

<table>
<thead>
<tr>
<th>NO.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00:00:04</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>ADV_IND</td>
</tr>
<tr>
<td>2</td>
<td>00:00:08</td>
<td>7e:D7:8E:C7:8e:51</td>
<td>58:D7:8E:C7:8e:31</td>
<td>SCAN_REQ</td>
</tr>
<tr>
<td>3</td>
<td>00:00:12</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>SCAN_RSP</td>
</tr>
<tr>
<td>4</td>
<td>00:00:16</td>
<td>4f:b7:8E:C7:8e:38</td>
<td>58:D7:8E:C7:8e:31</td>
<td>SCAN_REQ</td>
</tr>
<tr>
<td>5</td>
<td>00:00:24</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>ADV_IND</td>
</tr>
</tbody>
</table>
## Our Discovery I — Allowlist-based Side Channel

<table>
<thead>
<tr>
<th>NO.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00:00:04</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>ADV_IND</td>
</tr>
<tr>
<td>2</td>
<td>00:00:08</td>
<td>7e:D7:8E:C7:8e:51</td>
<td>58:D7:8E:C7:8e:31</td>
<td>SCAN_REQ</td>
</tr>
<tr>
<td>3</td>
<td>00:00:12</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>SCAN_RSP</td>
</tr>
<tr>
<td>4</td>
<td>00:00:16</td>
<td>4f:b7:8E:C7:8e:38</td>
<td>58:D7:8E:C7:8e:31</td>
<td>SCAN_REQ</td>
</tr>
<tr>
<td>5</td>
<td>00:00:24</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>ADV_IND</td>
</tr>
</tbody>
</table>

.....

<table>
<thead>
<tr>
<th>NO.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>00:15:08</td>
<td>73:D7:8E:C7:8e:45</td>
<td>58:D7:8E:C7:8e:31</td>
<td>SCAN_REQ</td>
</tr>
<tr>
<td>201</td>
<td>00:15:12</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>SCAN_RSP</td>
</tr>
</tbody>
</table>
Our Discovery I — Allowlist-based Side Channel

<table>
<thead>
<tr>
<th>NO.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>00:00:04</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>ADV_IND</td>
</tr>
<tr>
<td>2</td>
<td>00:00:08</td>
<td>7e:D7:8E:C7:8e:51</td>
<td>58:D7:8E:C7:8e:31</td>
<td>SCAN_REQ</td>
</tr>
<tr>
<td>3</td>
<td>00:00:12</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>SCAN_RSP</td>
</tr>
<tr>
<td>4</td>
<td>00:00:16</td>
<td>4f:b7:8E:C7:8e:38</td>
<td>58:D7:8E:C7:8e:31</td>
<td>SCAN_REQ</td>
</tr>
<tr>
<td>5</td>
<td>00:00:24</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>ADV_IND</td>
</tr>
<tr>
<td>200</td>
<td>00:15:08</td>
<td>73:D7:8E:C7:8e:45</td>
<td>58:D7:8E:C7:8e:31</td>
<td>SCAN_REQ</td>
</tr>
<tr>
<td>201</td>
<td>00:15:12</td>
<td>58:D7:8E:C7:8e:31</td>
<td>Broadcast</td>
<td>SCAN_RSP</td>
</tr>
</tbody>
</table>

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

---

Introduction
Background
BLE Security
BLE Privacy
Takeaway
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 \((irk_p)\)

\[
(I) \text{ RPA Generation} \\
\text{rpa}_p = prand_{24}||H_{24}(Prand_{24}||irk_p)
\]

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...03 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>

Identity Resolving Key \((irk_c)\)
Our Discovery II — MAC Address Replay

Identity Resolving Key ($irk_p$)

(I) RPA Generation

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

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>

Identity Resolving Key ($irk_c$)

(II) RPA Resolution

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

$$irk_p = irk_c \Rightarrow rpa_p = rpa_c$$

$rpa_p$
Our Discovery II — MAC Address Replay

Identity Resolving Key ($irk_p$)

(I) RPA Generation

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

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>

No Identity Resolving Key

RPA Replay ($rpa'_p$)

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>

Identity Resolving Key ($irk_c$)

(II) RPA Resolution

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

$irk_p = irk_c \Rightarrow rpa_p = rpa_c$

$rpa_p$
**Our Discovery II — MAC Address Replay**

### Identity Resolving Key \((\text{irk}_p)\)

**Type**

```
01 (2 bits) 0x00...3 (22 bits) 0x00...04 (24 bits)
```

#### (I) RPA Generation

\[
\text{rpa}_p = \text{prand}_{24}||H_{24}(\text{Prand}_{24}||\text{irk}_p)
\]

#### (II) RPA Resolution

\[
\text{rpa}_c = \text{prand}_{24}||H_{24}(\text{Prand}_{24}||\text{irk}_c)
\]

### No Identity Resolving Key

#### RPA Replay \((\text{rpa'}_p)\)

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2 bits)</td>
<td>0x00...3 (22 bits)</td>
<td>0x00...04 (24 bits)</td>
</tr>
</tbody>
</table>

\[ \text{irk}_p = \text{irk}_c \rightarrow \text{rpa}_p = \text{rpa}_c \]

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2 bits)</td>
<td>0x00...3 (22 bits)</td>
<td>0x00...04 (24 bits)</td>
</tr>
</tbody>
</table>
Active BAT Attacks: Tracking a Victim’s Past Trajectory

Attack II: Tracking a Victim’s Past Trajectory
Active BAT Attacks: Tracking a Victim’s Past Trajectory

Attack II: Tracking a Victim’s Past Trajectory
Active BAT Attacks: Tracking a Victim’s Past Trajectory

Attack II: Tracking a Victim’s Past Trajectory
Active BAT Attacks: Tracking a Victim’s Past Trajectory

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
Identity Resolving Key ($irk_p$)  
Identity Resolving Key ($irk_c$)
SABLE — Defense

Identity Resolving Key \((irk_p)\)

(I) RPA Generation

\[ rpa_p = \text{prand}_{24}||H_{24}(\text{Prand}_{24}||T||irk_p) \]

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>
SABLE — Defense

Identity Resolving Key ($irk_p$)

(I) RPA Generation

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

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00…3 (22bits)</td>
<td>0x00…04 (24bits)</td>
</tr>
</tbody>
</table>

Identity Resolving Key ($irk_c$)
SABLE — Defense

Identity Resolving Key \((irkp)\)

(I) RPA Generation

\[ rpa_p = prand_{24} \parallel H_{24}(Prand_{24} \parallel T \parallel irkp) \]

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>

Identity Resolving Key \((irk_c)\)

(II) RPA Resolution

\[ rpa_c = prand_{24} \parallel H_{24}(Prand_{24} \parallel T \parallel irkc) \]

\[ irkp = irkc \Rightarrow rpa_p = rpa_c \]

Within a Threshold \(Tx\)

rpa_p

\[ 01 \ (2bits) \quad 0x00...3 \ (22bits) \quad 0x00...04 \ (24bits) \]
SABLE — Defense

Identity Resolving Key ($irk_p$)

(I) RPA Generation

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

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0x00...3</td>
<td>0x00...04</td>
</tr>
</tbody>
</table>

No Identity Resolving Key

RPA Replay ($rpa'_p$)

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0x00...3</td>
<td>0x00...04</td>
</tr>
</tbody>
</table>

Identity Resolving Key ($irk_c$)

(II) RPA Resolution

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

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>0x00...3</td>
<td>0x00...04</td>
</tr>
</tbody>
</table>

Within a Threshold $Tx$

$irk_p = irk_c \rightarrow rpa_p = rpa_c$

$rpa_p$
SABLE — Defense

Identity Resolving Key \( (irk_p) \)

(I) RPA Generation
\[
\text{rpa}_p = \text{prand}_{24} \| H_{24}(\text{Prand}_{24} \| T \| irk_p)
\]

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>

No Identity Resolving Key

RPA Replay \( (rpa'_p) \)

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>

Identity Resolving Key \( (irk_c) \)

(II) RPA Resolution
\[
\text{rpa}_c = \text{prand}_{24} \| H_{24}(\text{Prand}_{24} \| T \| irk_c)
\]

<table>
<thead>
<tr>
<th>Type</th>
<th>rand</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>01 (2bits)</td>
<td>0x00...3 (22bits)</td>
<td>0x00...04 (24bits)</td>
</tr>
</tbody>
</table>

\( irk_p = irk_c \Rightarrow rpa_p = rpa_c \)

Within a Threshold \( Tx \)

\[
\text{rpa}_p \quad \text{OK}
\]

\[
\text{rpa'}_p \quad \text{NG}
\]
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. Similar to allowlist, those new features need to be scrutinized. 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
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
10. Extrapolating Formal Analysis to Uncover Attacks in Bluetooth Passkey Entry Pairing. In NDSS 2023
Rethinking the Security and Privacy of Bluetooth Low Energy

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

10/14/2022