29TH USENIX
SECURITY SYMPOSIUM

Track 1: Wireless Security

AUGUST 12–14, 2020
VIRTUAL EVENT

Breaking Secure Pairing of Bluetooth Low Energy Using Downgrade Attacks

Yue Zhang, Jian Weng, Rajib Dey , Yier Jin,
Zhiqiang Lin, and Xinwen Fu

暨南大学 1906 JINAN UNIVERSITY JNU

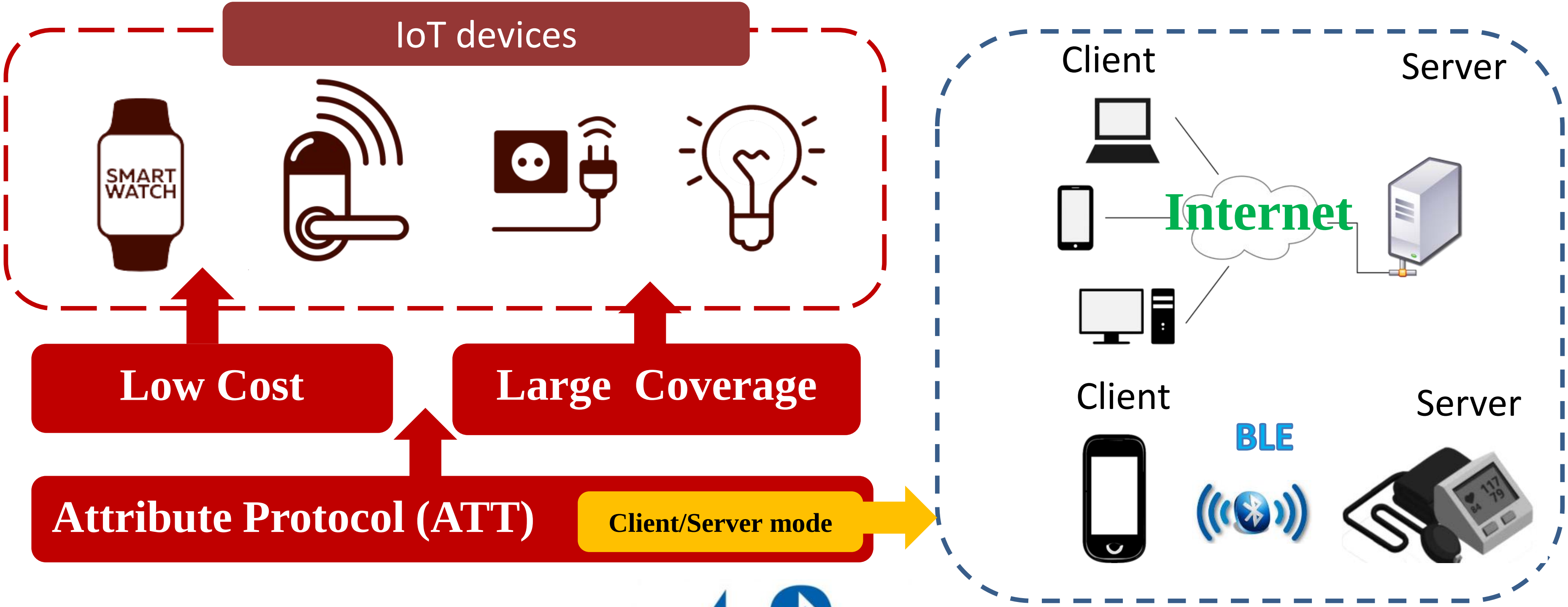
UNIVERSITY OF CENTRAL FLORIDA 1963

UMASS LOWELL

THE OHIO STATE UNIVERSITY

UF UNIVERSITY of FLORIDA

Bluetooth Low Energy (BLE) and IoT



IoT devices

Low Cost

Large Coverage

Attribute Protocol (ATT)

Client/Server mode

Client

Server

Internet

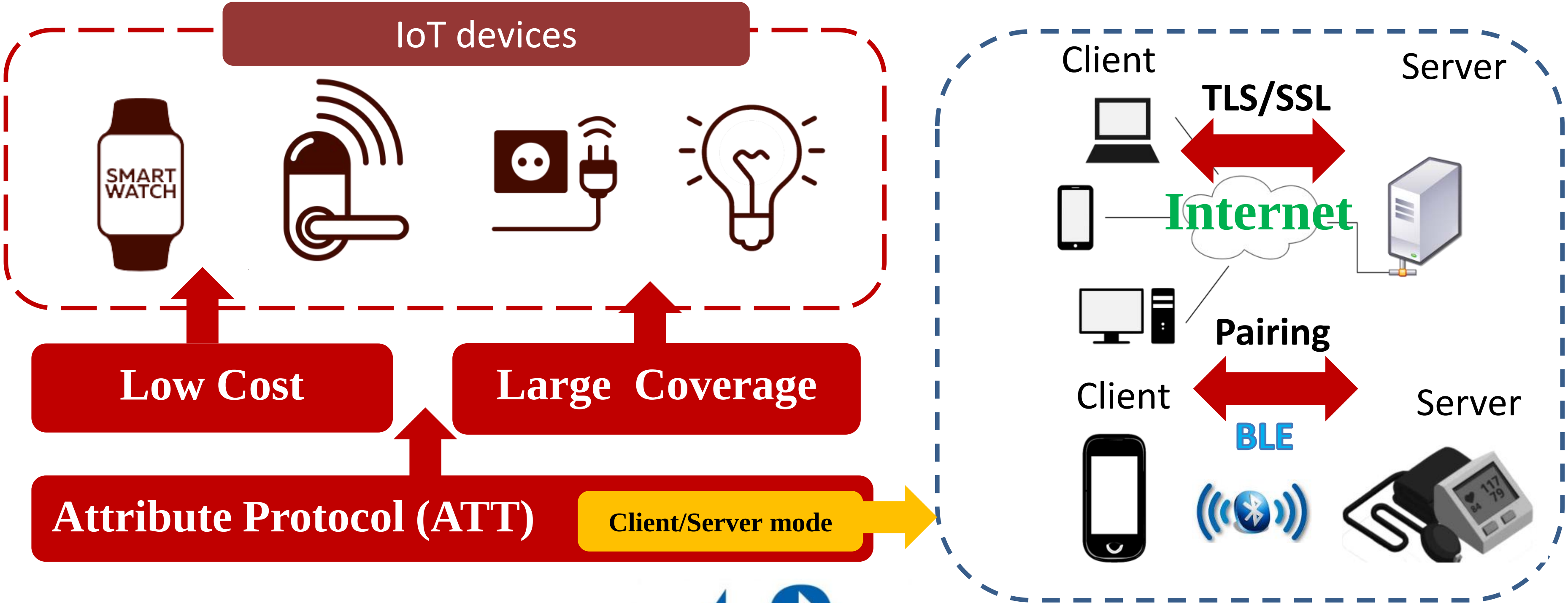
Client

Server

BLE

Bluetooth Low Energy 4.0

Bluetooth Low Energy (BLE) and IoT



IoT devices

Low Cost

Large Coverage

Attribute Protocol (ATT)

Client/Server mode

Client

Server

TLS/SSL

Internet

Pairing

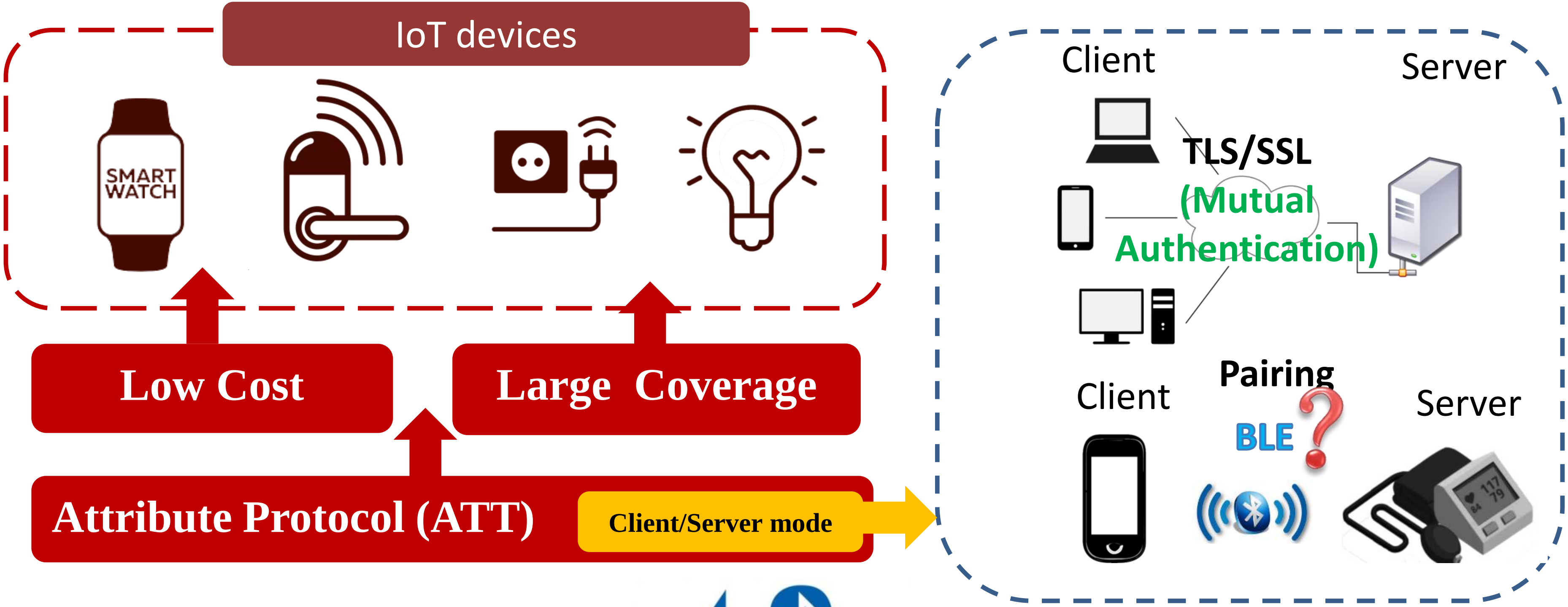
Client

Server

BLE

Bluetooth Low Energy 4.0

Bluetooth Low Energy (BLE) and IoT



IoT devices

Low Cost

Large Coverage

Attribute Protocol (ATT)

Client/Server mode

Client

Server

TLS/SSL

(Mutual Authentication)

Client

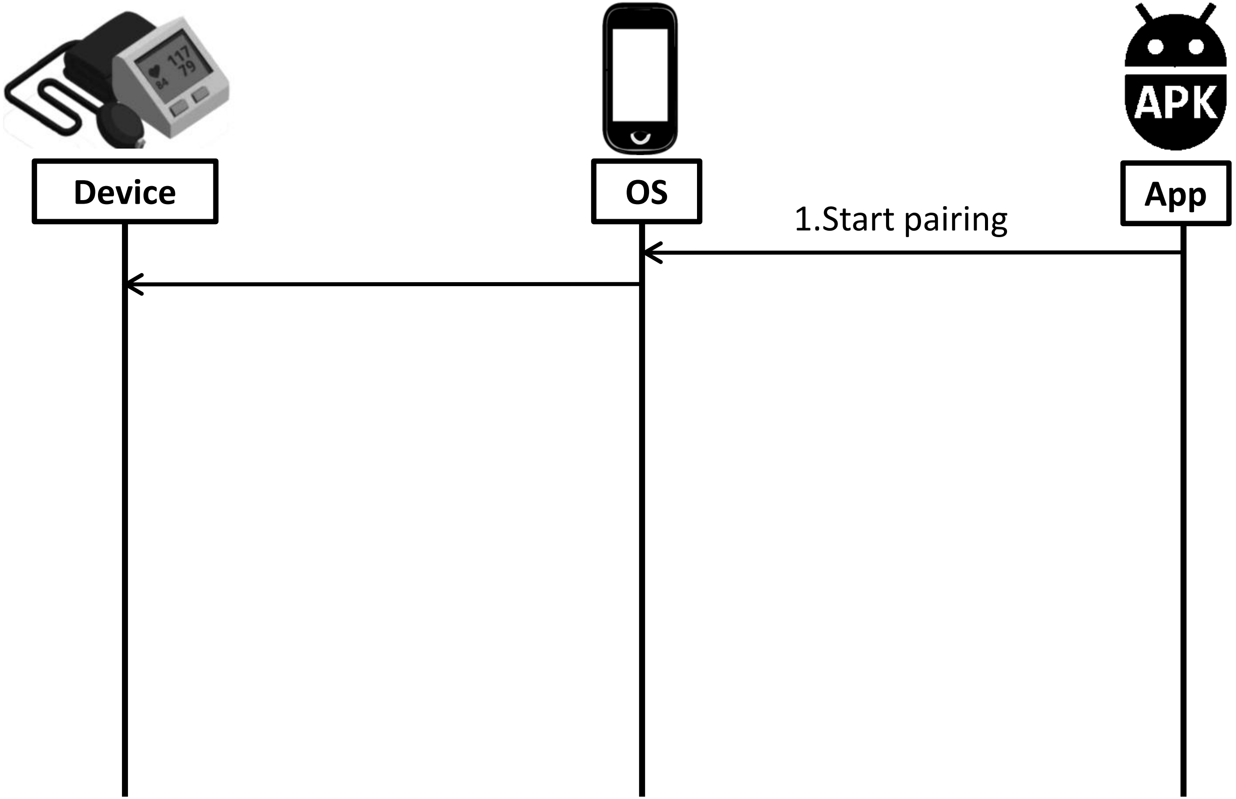
Pairing

BLE ?

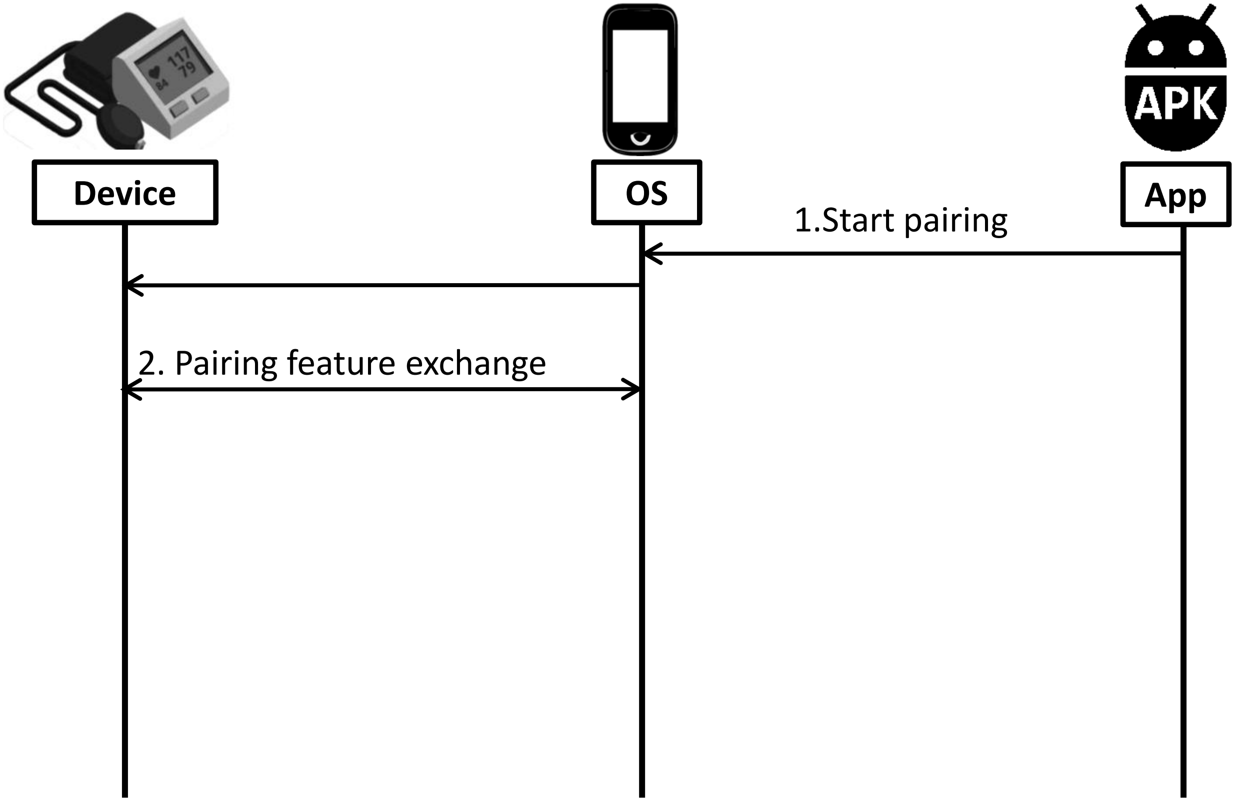
Server

Bluetooth Low Energy 4.0

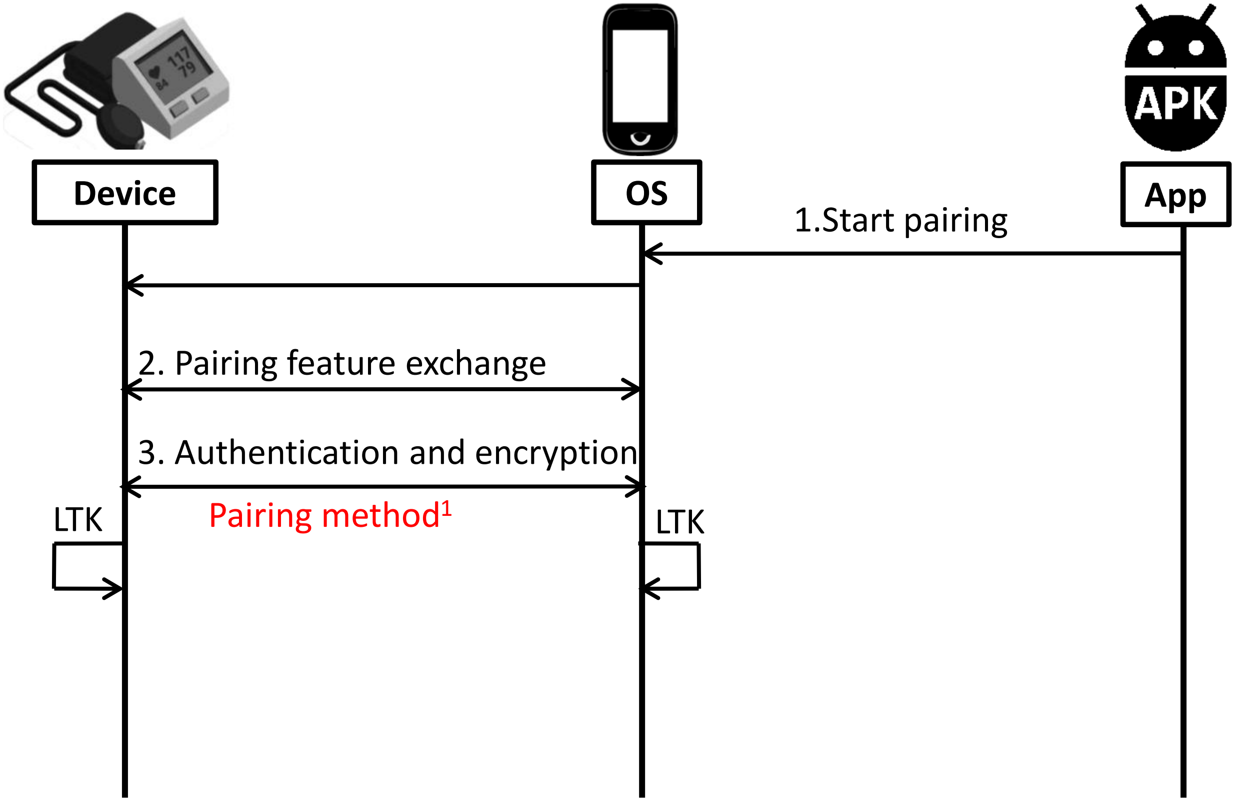
General Workflow of BLE Pairing



General Workflow of BLE Pairing

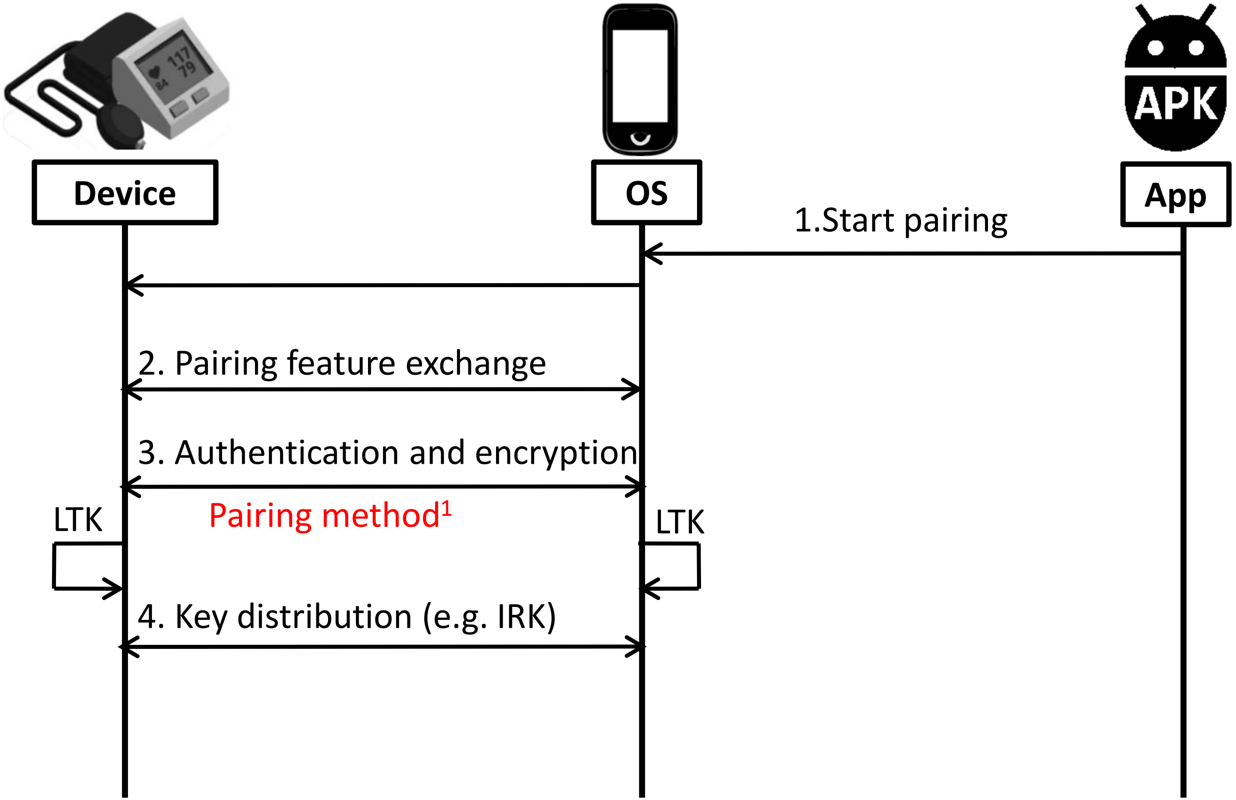


General Workflow of BLE Pairing



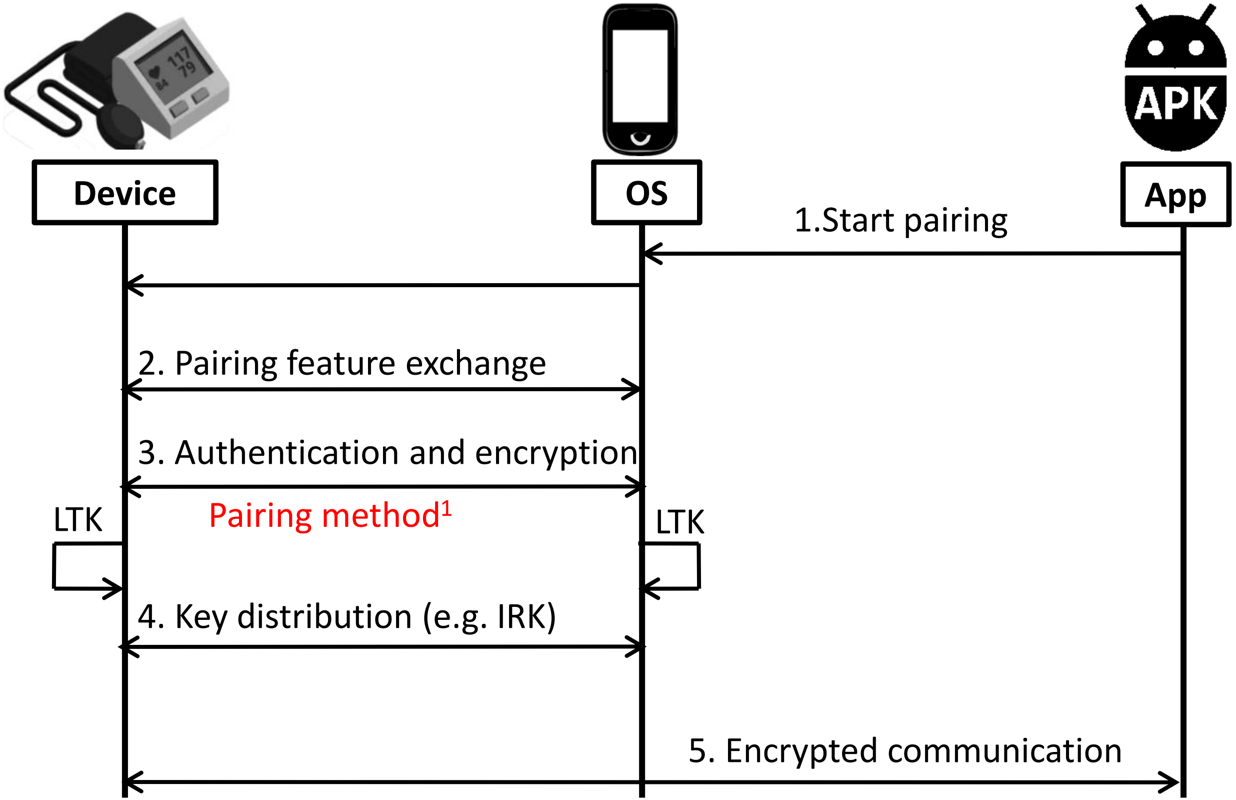
Pairing method:
Just Works, Passkey Entry
Numeric Comparison, Out of Band

General Workflow of BLE Pairing


117
79
84
APK
Device
OS
App
1.Start pairing
2. Pairing feature exchange
3. Authentication and encryption
Pairing method[1]
LTK
LTK
4. Key distribution (e.g. IRK)

Pairing method:
Just Works, Passkey Entry
Numeric Comparison, Out of Band

General Workflow of BLE Pairing


APK
Device
OS
App
1.Start pairing
2. Pairing feature exchange
3. Authentication and encryption
LTK
Pairing method[1]
LTK
4. Key distribution (e.g. IRK)
5. Encrypted communication

Pairing method:
Just Works, Passkey Entry
Numeric Comparison, Out of Band

Security Levels of BLE Pairing

Security Levels:

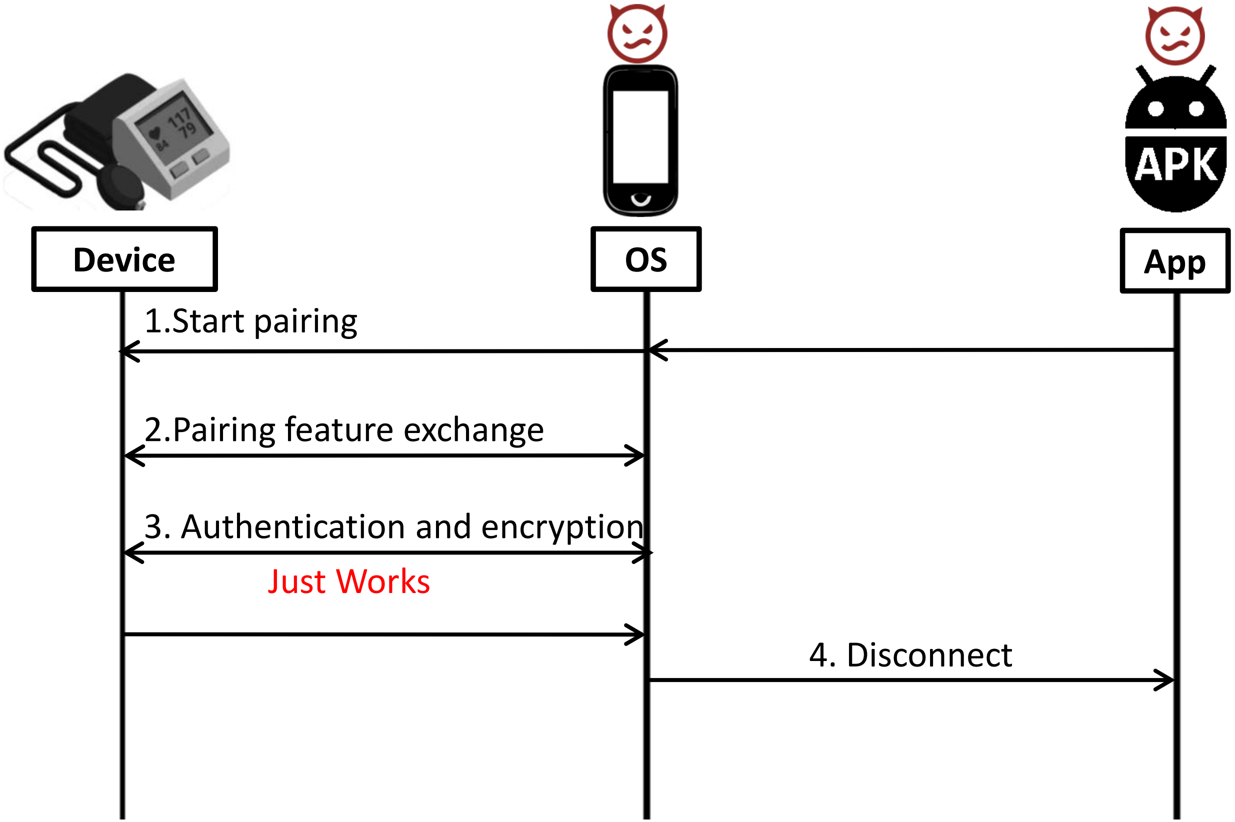
- None (Plaintext)
- Encrypted (Just Works)
- Authenticated (Passkey Entry or Numeric Comparison)
- Secure Connections Only (SCO) mode (Enforced Passkey Entry or Numeric Comparison)

Security Levels of BLE Pairing

Security Levels:

- None (Plaintext)
- Encrypted (Just Works)
- Authenticated (Passkey Entry or Numeric Comparison)
- **Secure Connections Only (SCO) mode (Enforced Passkey Entry or Numeric Comparison)**

Our Observation



Device

OS

App

1.Start pairing

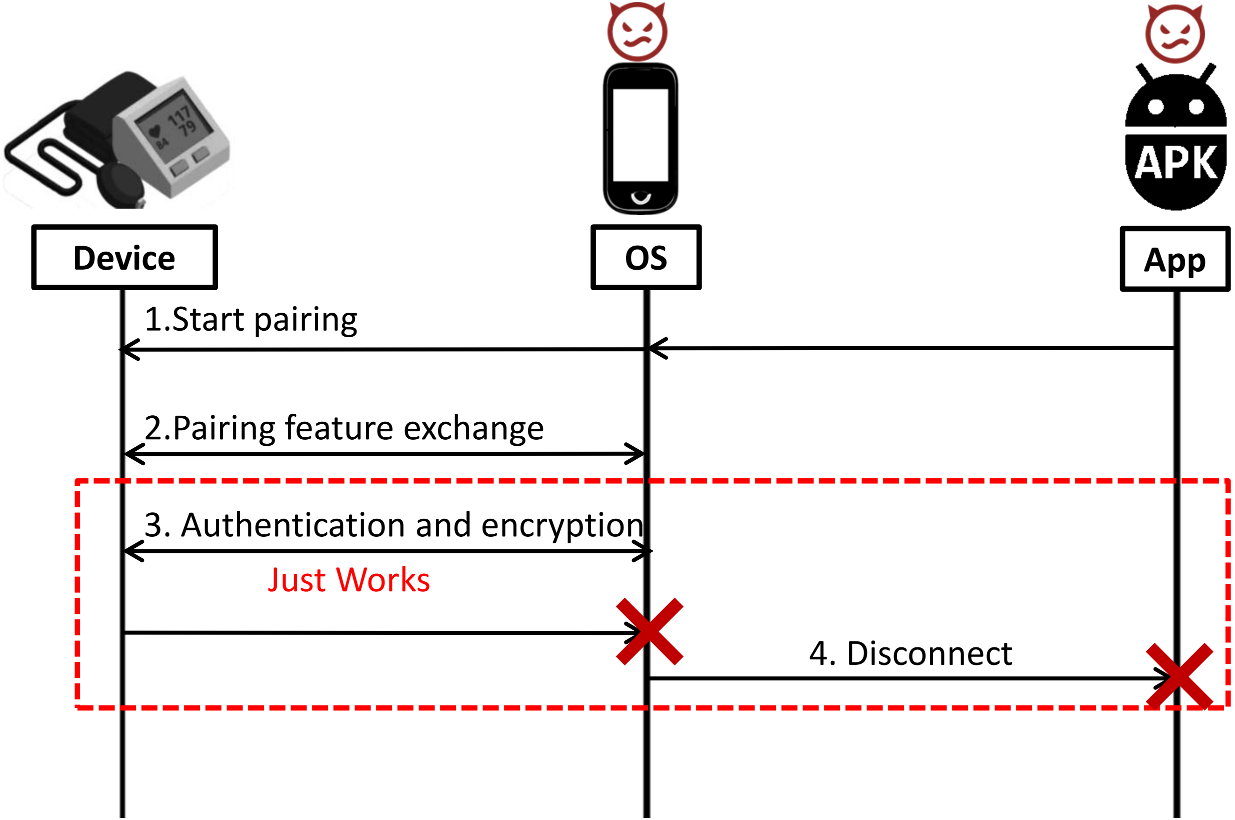
2.Pairing feature exchange

3. Authentication and encryption

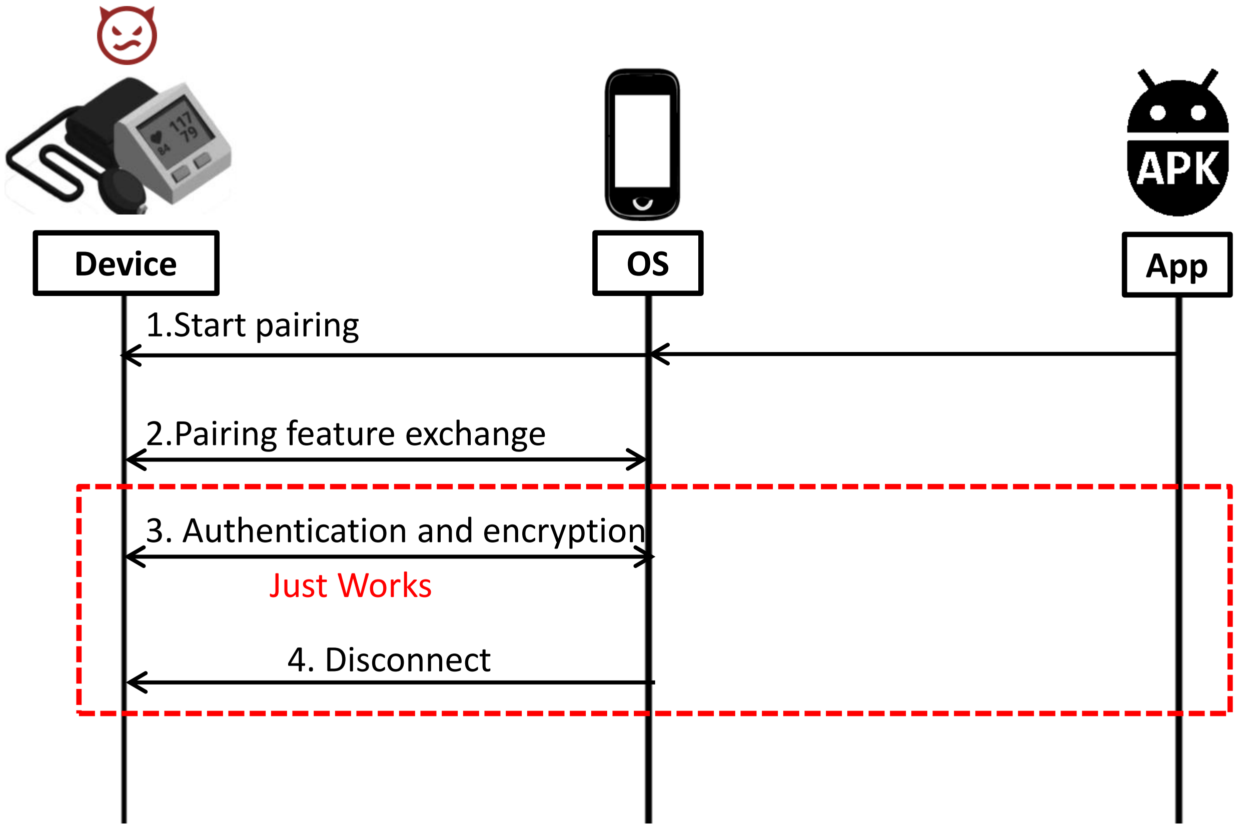
Just Works

4. Disconnect

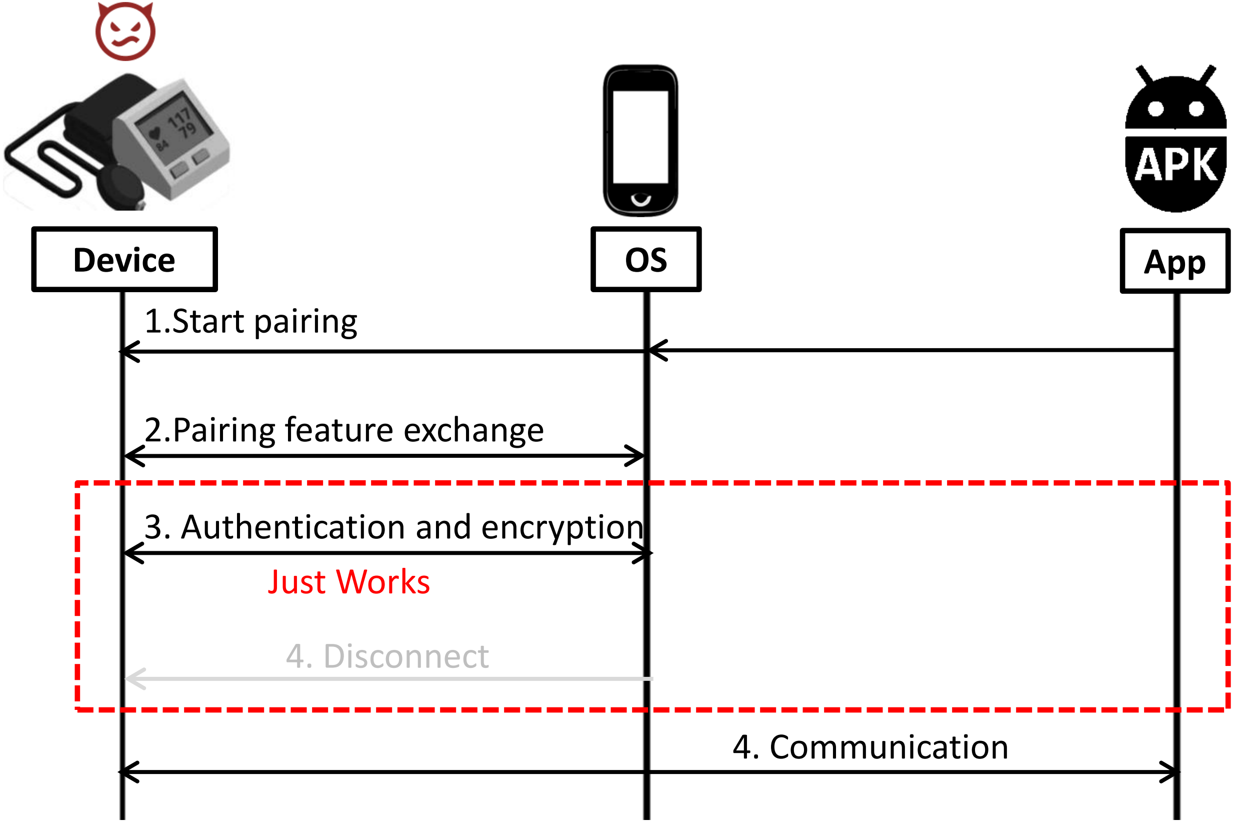
Our Observation



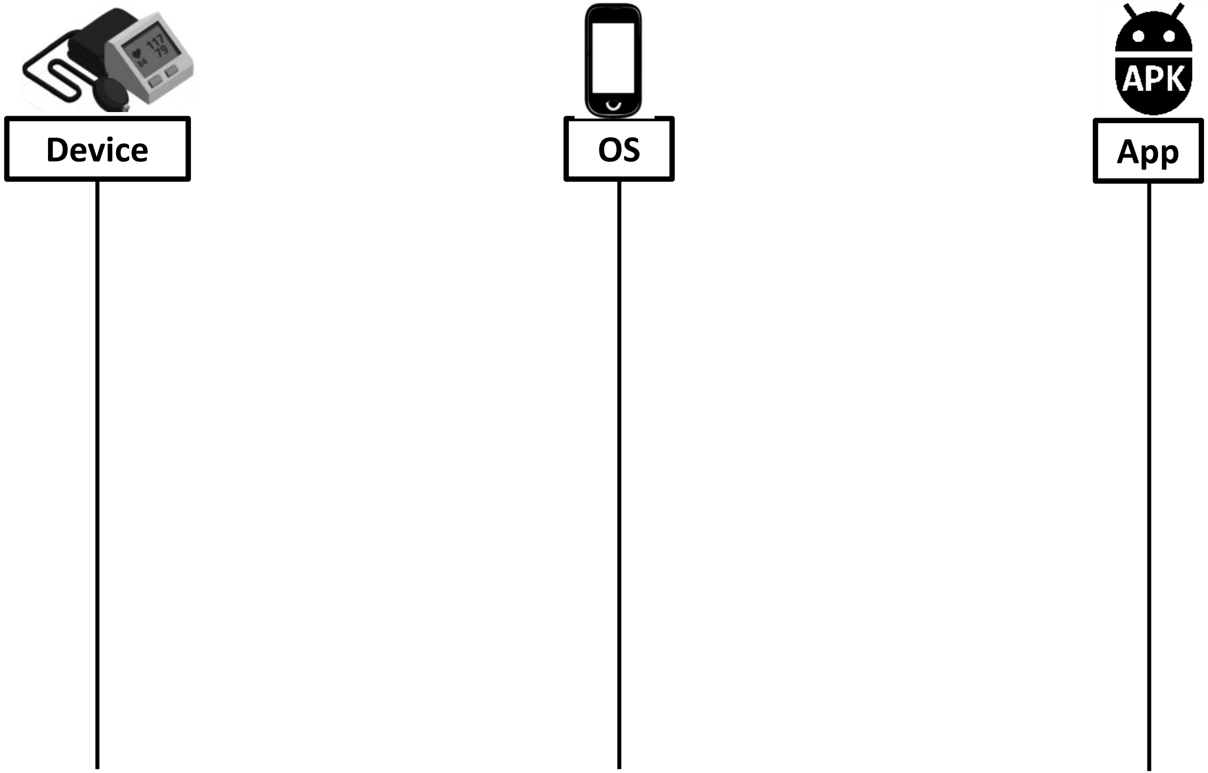
Our Observation



Our Observation



Required Four Capabilities at Initiator

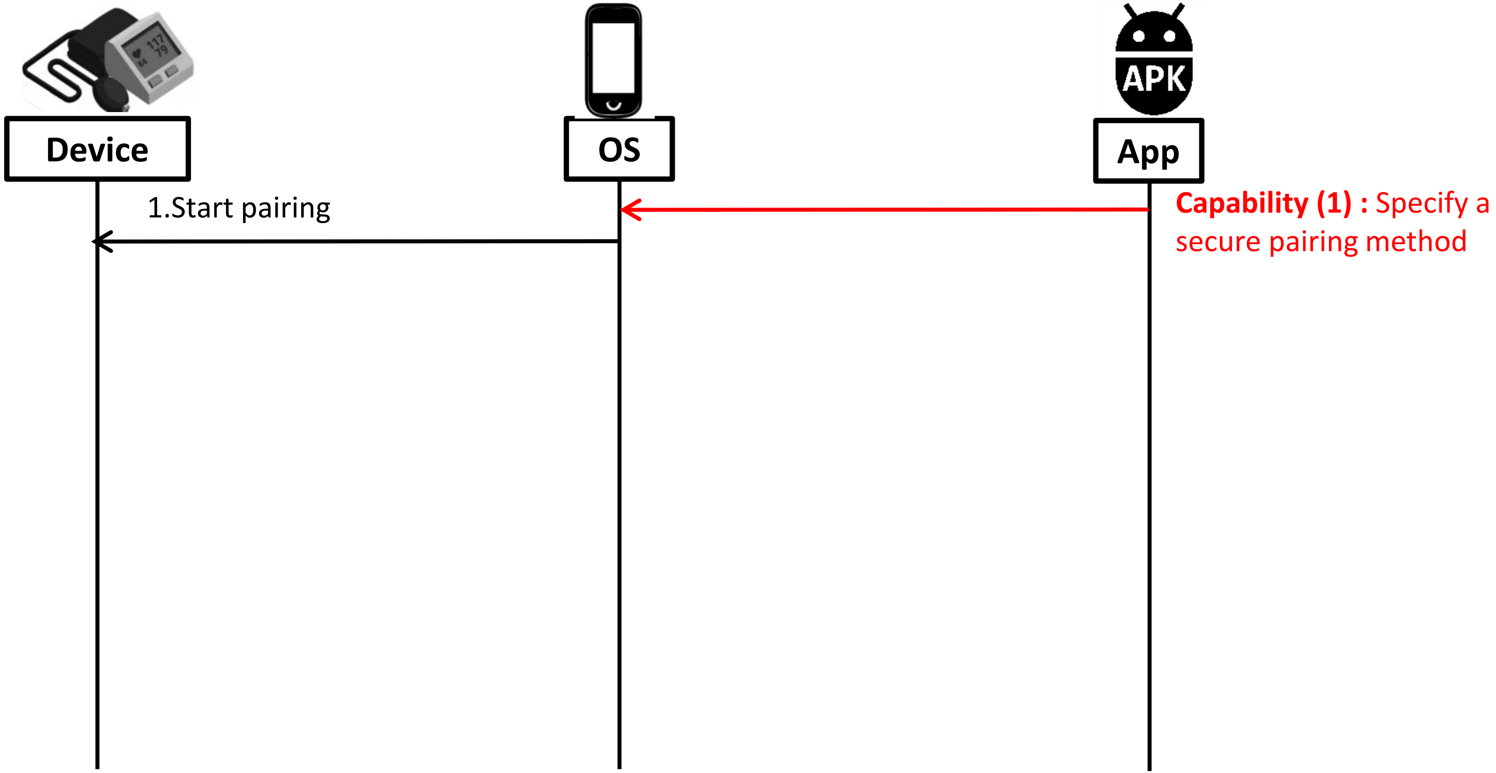


Device

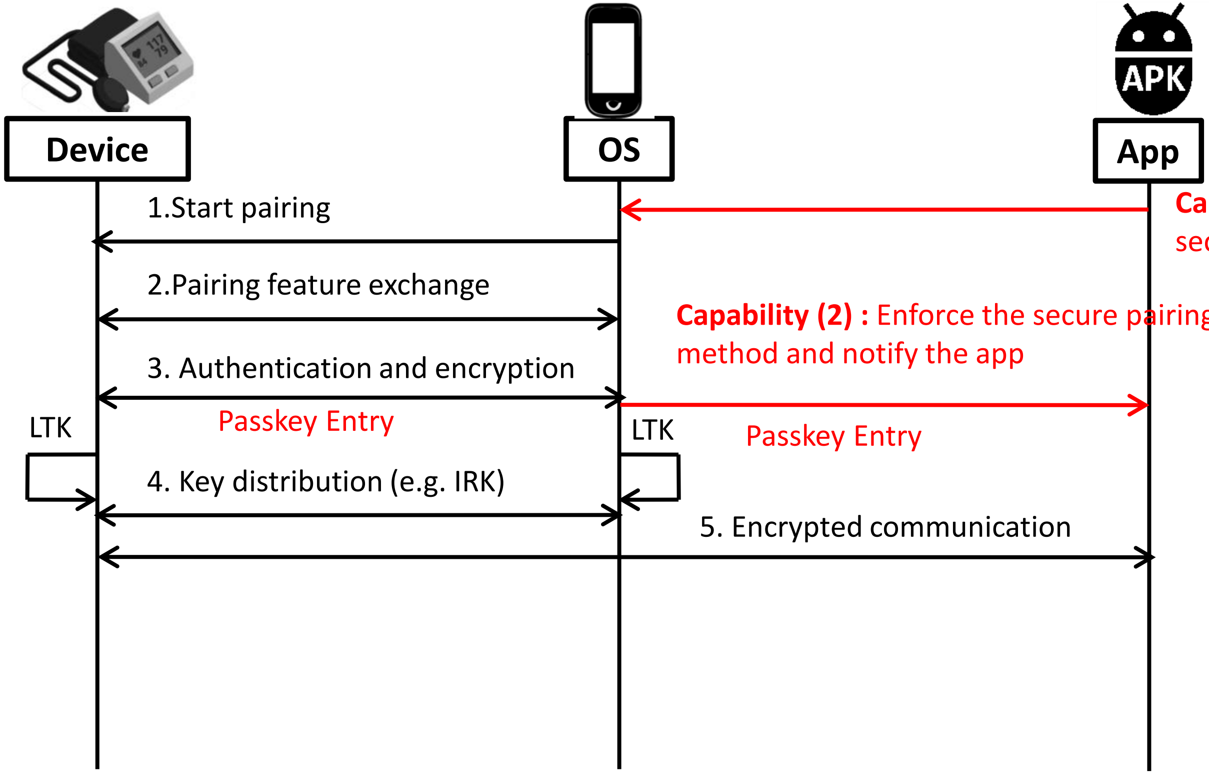
OS

App

Required Four Capabilities at Initiator - Initiation Stage



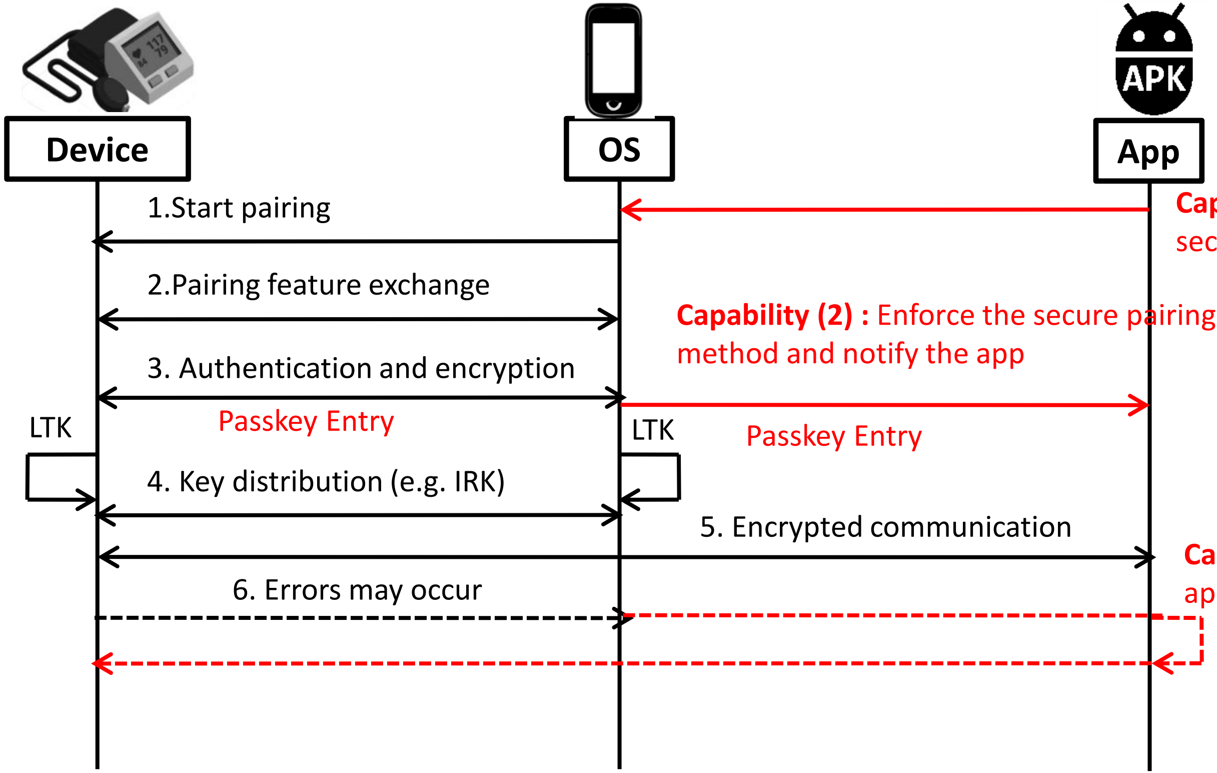
Required Four Capabilities at Initiator – Status management


Device
OS
APK
App
1.Start pairing
2.Pairing feature exchange
3. Authentication and encryption
Passkey Entry
LTK
LTK
Passkey Entry
4. Key distribution (e.g. IRK)
5. Encrypted communication

Capability (1) : Specify a secure pairing method

Capability (2) : Enforce the secure pairing method and notify the app

Required Four Capabilities at Initiator – Errors Handling

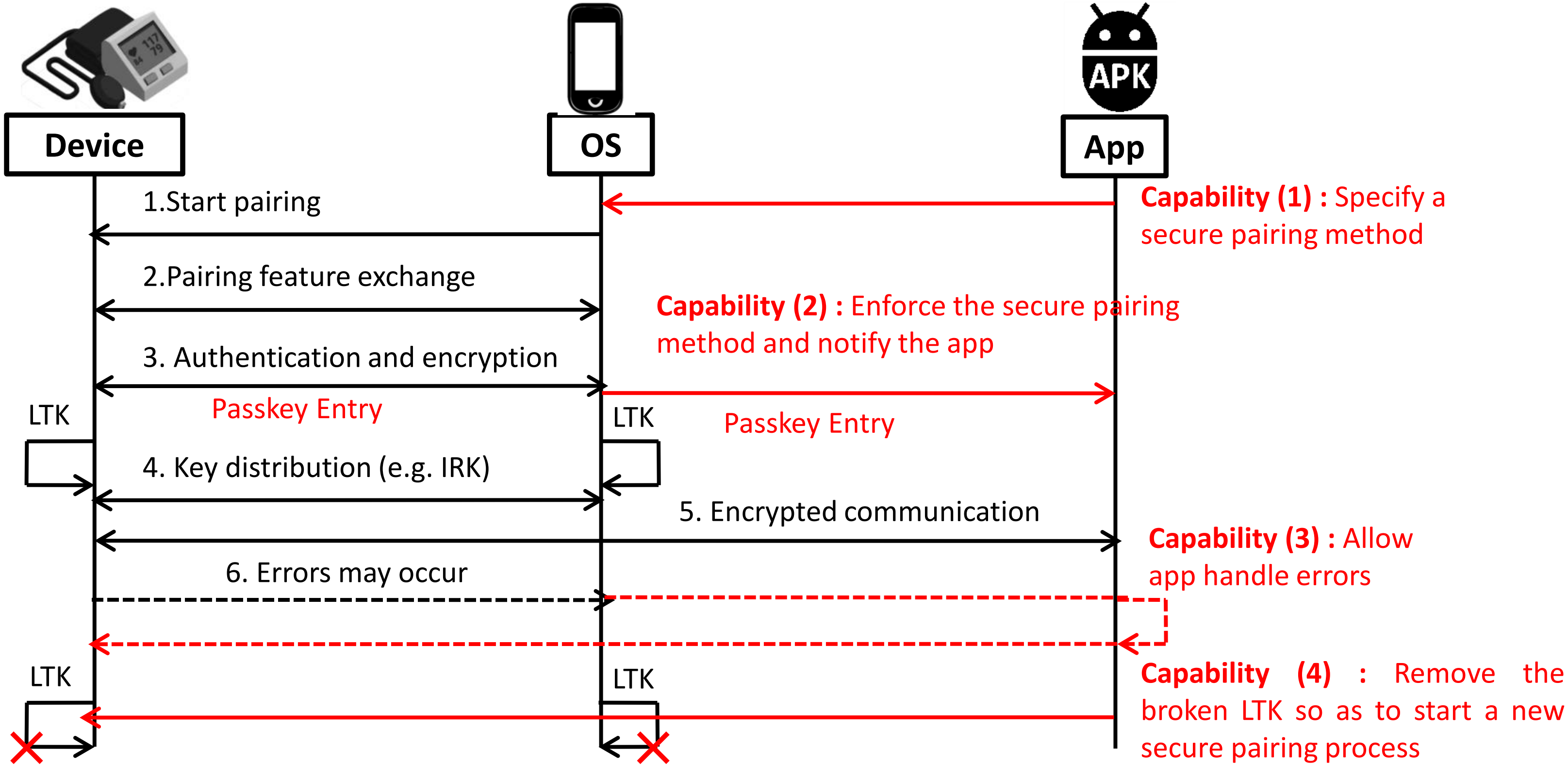

APK
Device
OS
App
1.Start pairing
2.Pairing feature exchange
3. Authentication and encryption
Passkey Entry
LTK
LTK
Passkey Entry
4. Key distribution (e.g. IRK)
5. Encrypted communication
6. Errors may occur

Capability (1) : Specify a secure pairing method

Capability (2) : Enforce the secure pairing method and notify the app

Capability (3) : Allow app handle errors

Required Four Capabilities at Initiator – Bond Management


Device
OS
App
APK
1.Start pairing
2.Pairing feature exchange
3. Authentication and encryption
Passkey Entry
LTK
LTK
Passkey Entry
4. Key distribution (e.g. IRK)
5. Encrypted communication
6. Errors may occur
LTK
LTK

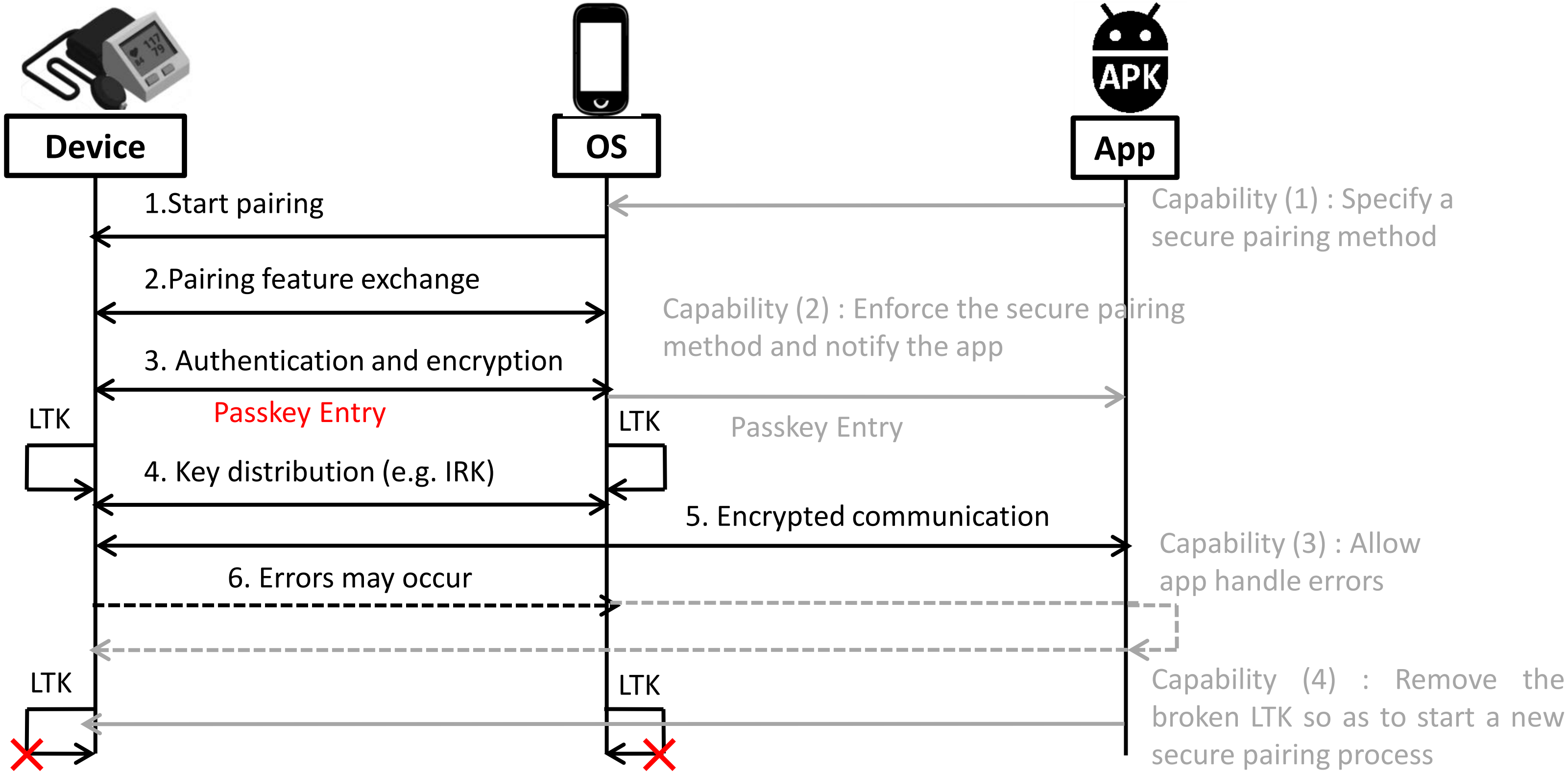
Capability (1) : Specify a secure pairing method

Capability (2) : Enforce the secure pairing method and notify the app

Capability (3) : Allow app handle errors

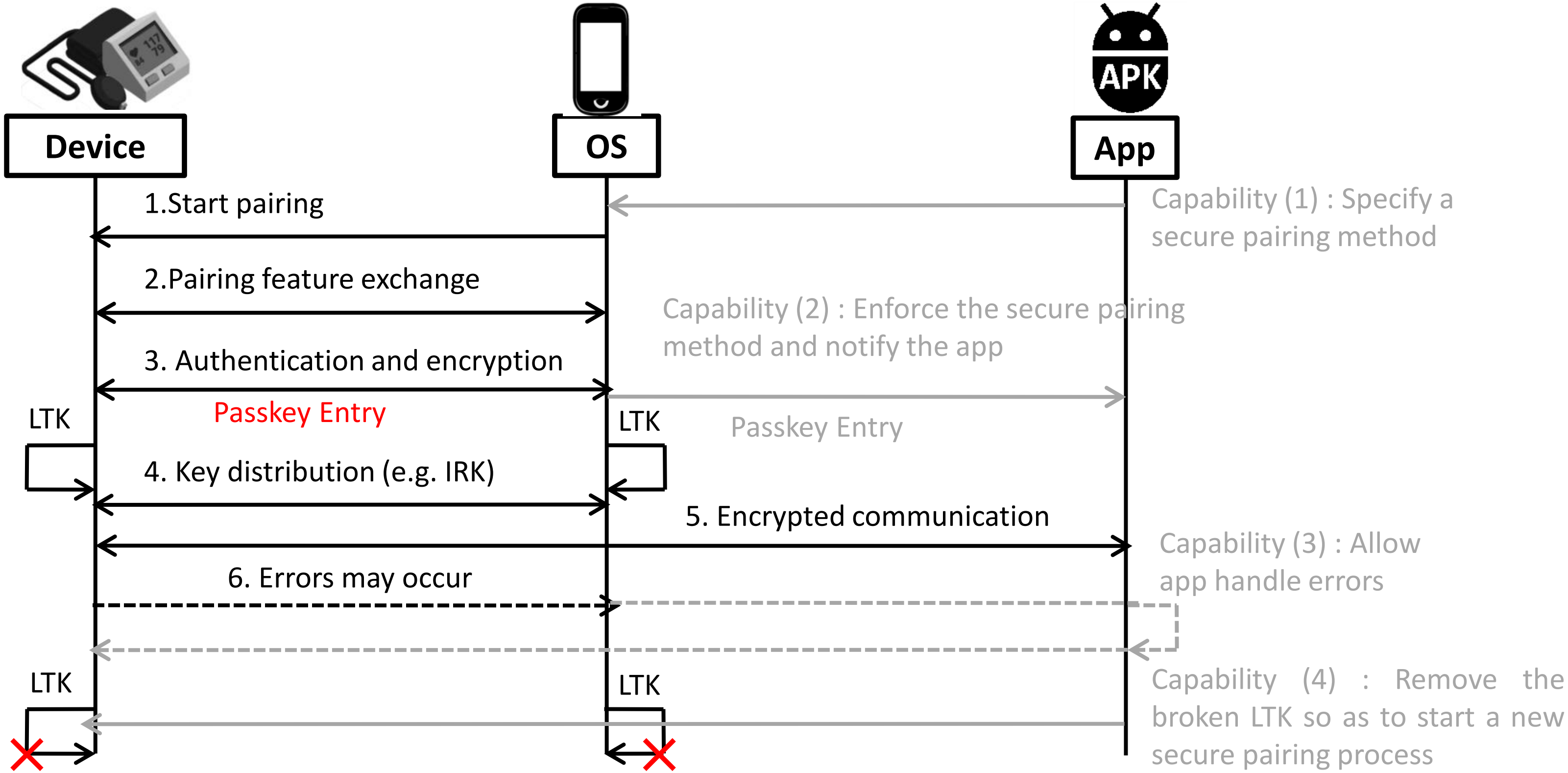
Capability (4) : Remove the broken LTK so as to start a new secure pairing process

No SCO mode at Initiator is Cause of Downgrade Attacks


Device
OS
APK
App
1.Start pairing
Capability (1) : Specify a secure pairing method
2.Pairing feature exchange
Capability (2) : Enforce the secure pairing method and notify the app
3. Authentication and encryption
LTK
Passkey Entry
LTK
Passkey Entry
4. Key distribution (e.g. IRK)
5. Encrypted communication
Capability (3) : Allow app handle errors
6. Errors may occur
LTK
LTK
Capability (4) : Remove the broken LTK so as to start a new secure pairing process

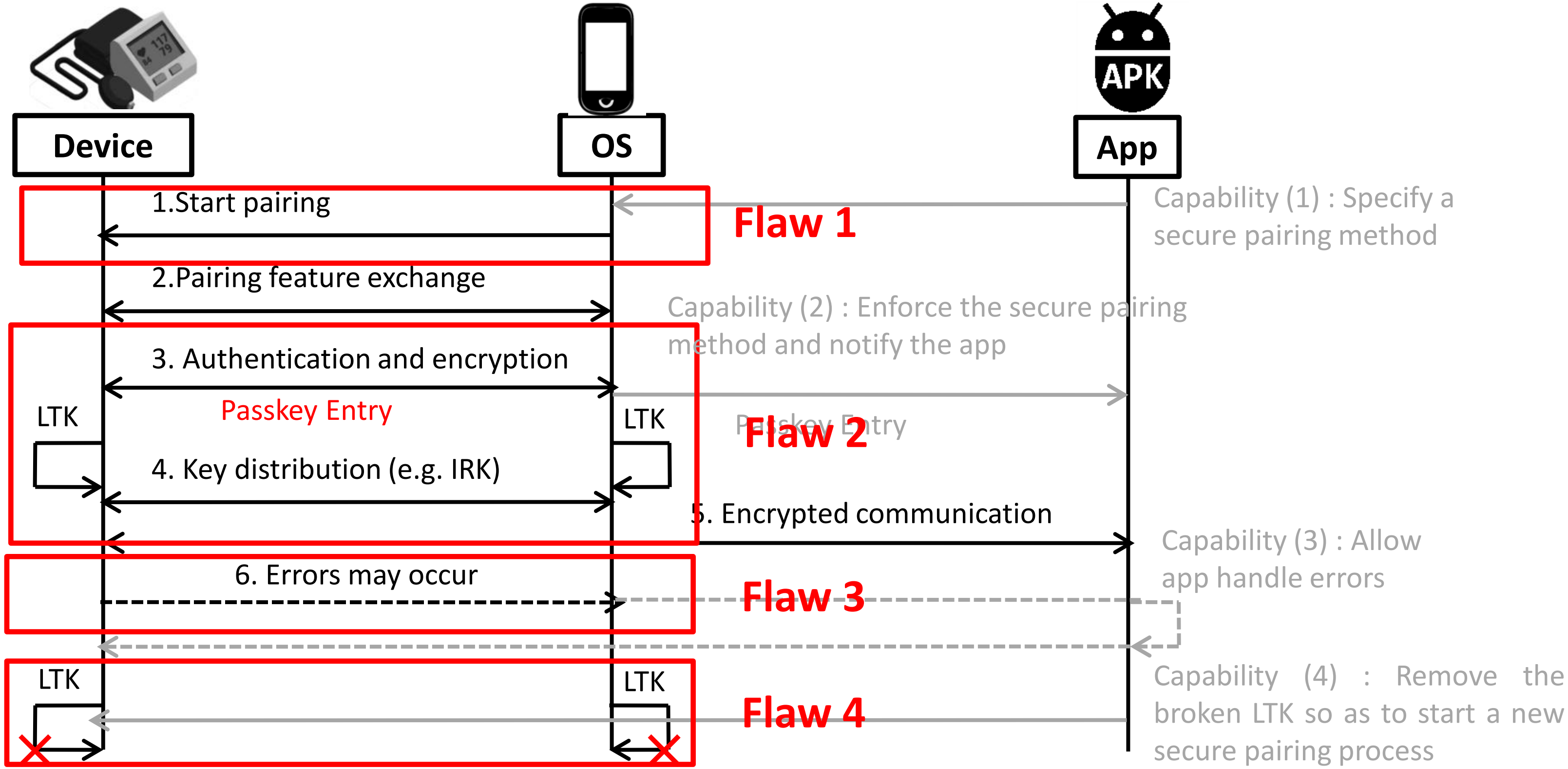
No SCO mode at Initiator is Cause of Downgrade Attacks

OS handles pairing events in a **compatible way without enforcing secure pairing**


Device
OS
APK
App
1.Start pairing
Capability (1) : Specify a secure pairing method
2.Pairing feature exchange
Capability (2) : Enforce the secure pairing method and notify the app
3. Authentication and encryption
Passkey Entry
LTK
LTK
Passkey Entry
4. Key distribution (e.g. IRK)
5. Encrypted communication
Capability (3) : Allow app handle errors
6. Errors may occur
LTK
LTK
Capability (4) : Remove the broken LTK so as to start a new secure pairing process

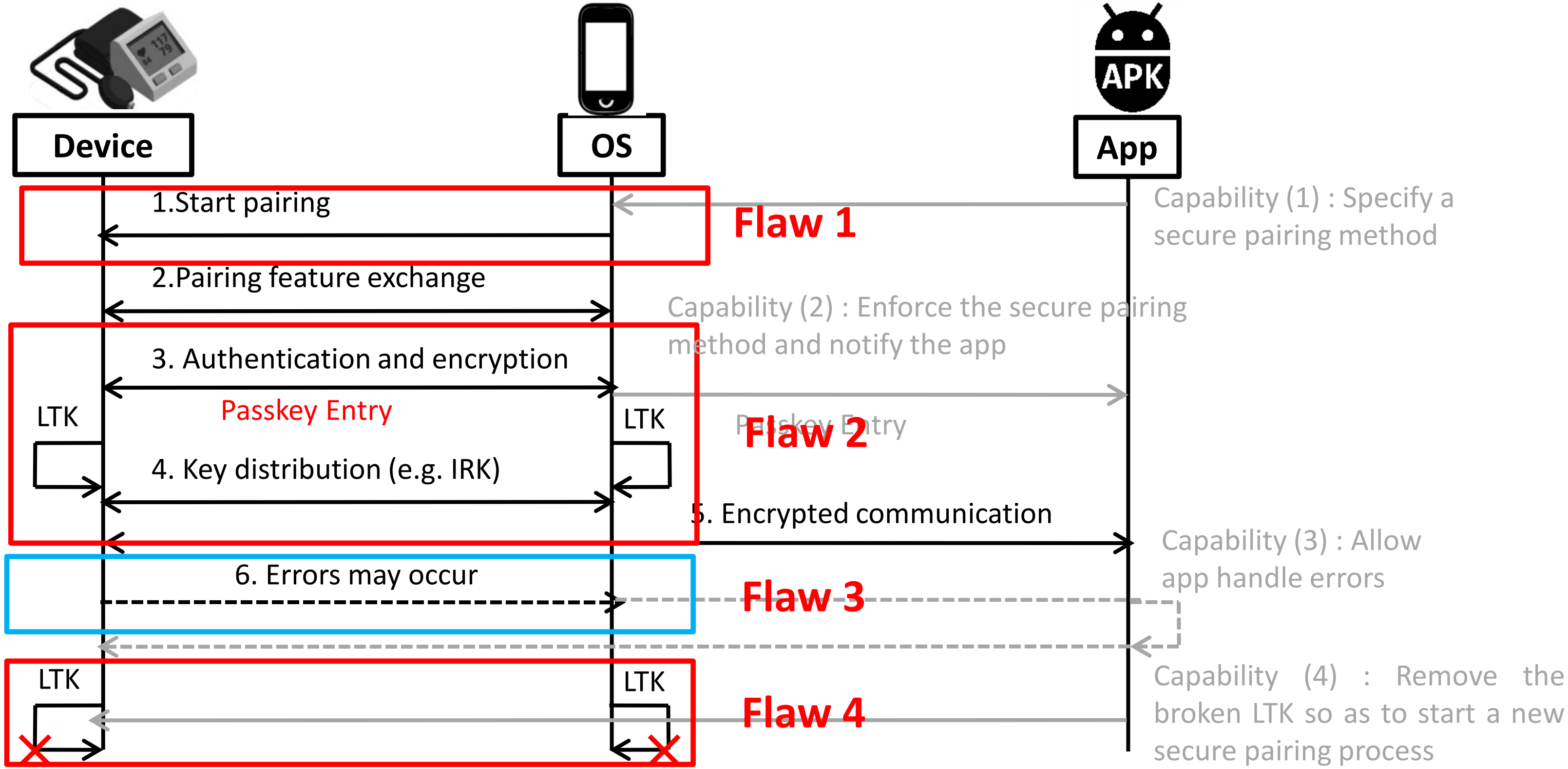
No SCO mode at Initiator is Cause of Downgrade Attacks

OS handles pairing events in a **compatible way without enforcing secure pairing**

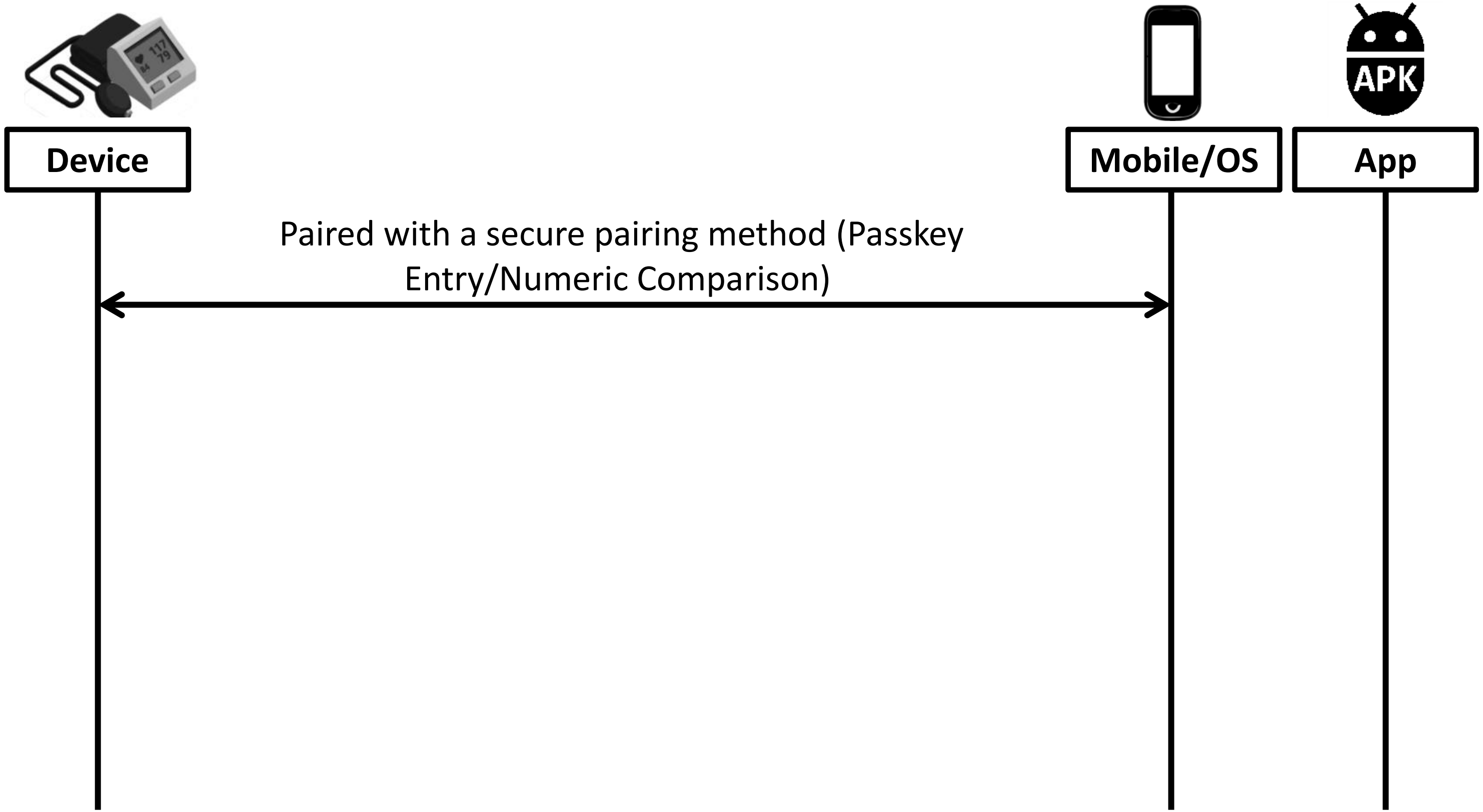

Device
OS
APK
App
1.Start pairing
Capability (1) : Specify a secure pairing method
Flaw 1
2.Pairing feature exchange
Capability (2) : Enforce the secure pairing method and notify the app
3. Authentication and encryption
LTK
Passkey Entry
LTK
Passkey Entry
Flaw 2
4. Key distribution (e.g. IRK)
5. Encrypted communication
Capability (3) : Allow app handle errors
6. Errors may occur
Flaw 3
LTK
LTK
Flaw 4
Capability (4) : Remove the broken LTK so as to start a new secure pairing process

No SCO mode at Initiator is Cause of Downgrade Attacks

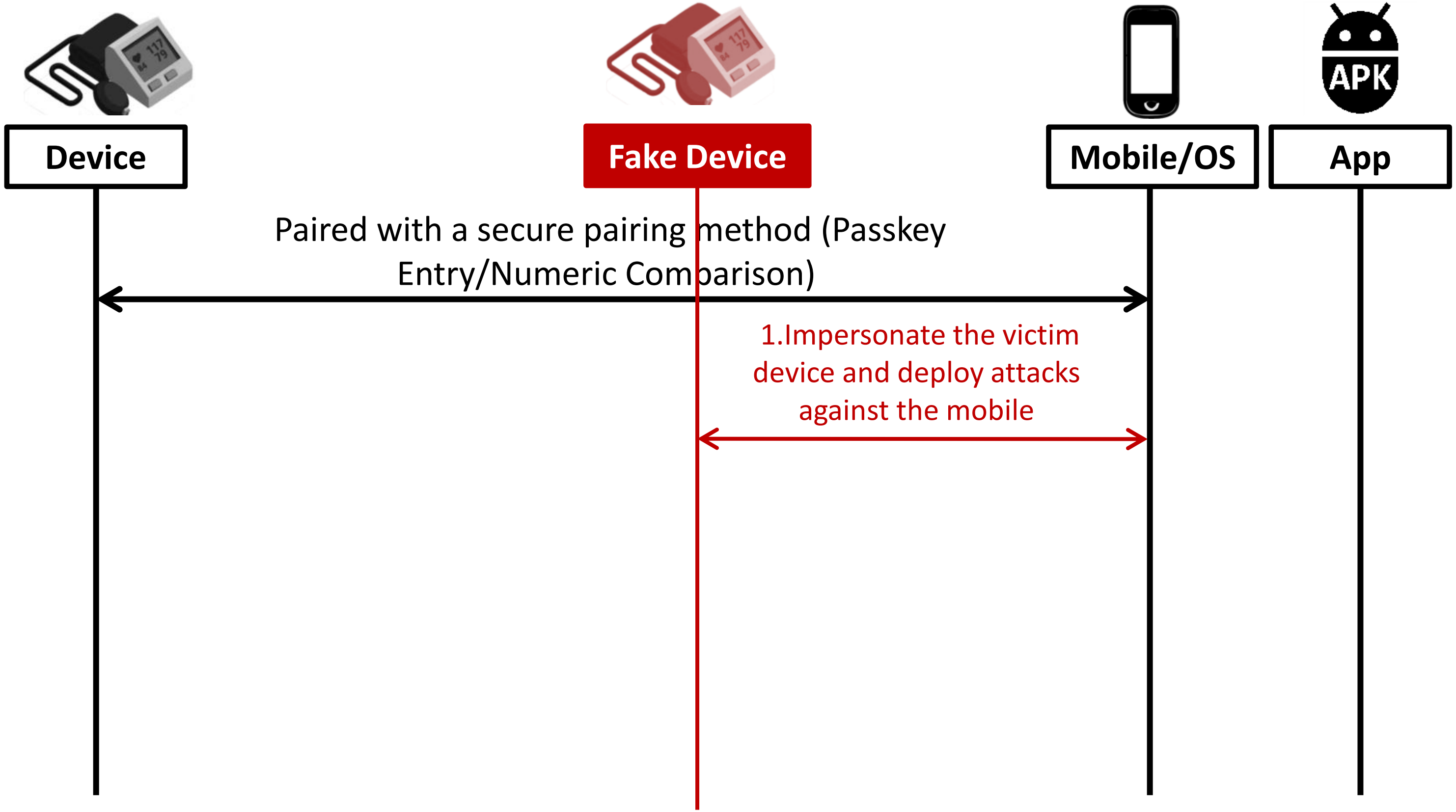
OS handle pairing events in a **compatible way without enforcing secure pairing**


APK
Device
OS
App
1.Start pairing
Flaw 1
Capability (1) : Specify a secure pairing method
2.Pairing feature exchange
Capability (2) : Enforce the secure pairing method and notify the app
3. Authentication and encryption
Passkey Entry
LTK
LTK
Passkey Entry
Flaw 2
4. Key distribution (e.g. IRK)
5. Encrypted communication
Capability (3) : Allow app handle errors
6. Errors may occur
Flaw 3
LTK
LTK
Flaw 4
Capability (4) : Remove the broken LTK so as to start a new secure pairing process

Threat model


Device
Mobile/OS
App
APK
Paired with a secure pairing method (Passkey Entry/Numeric Comparison)

Downgrade Attacks



Device

Fake Device

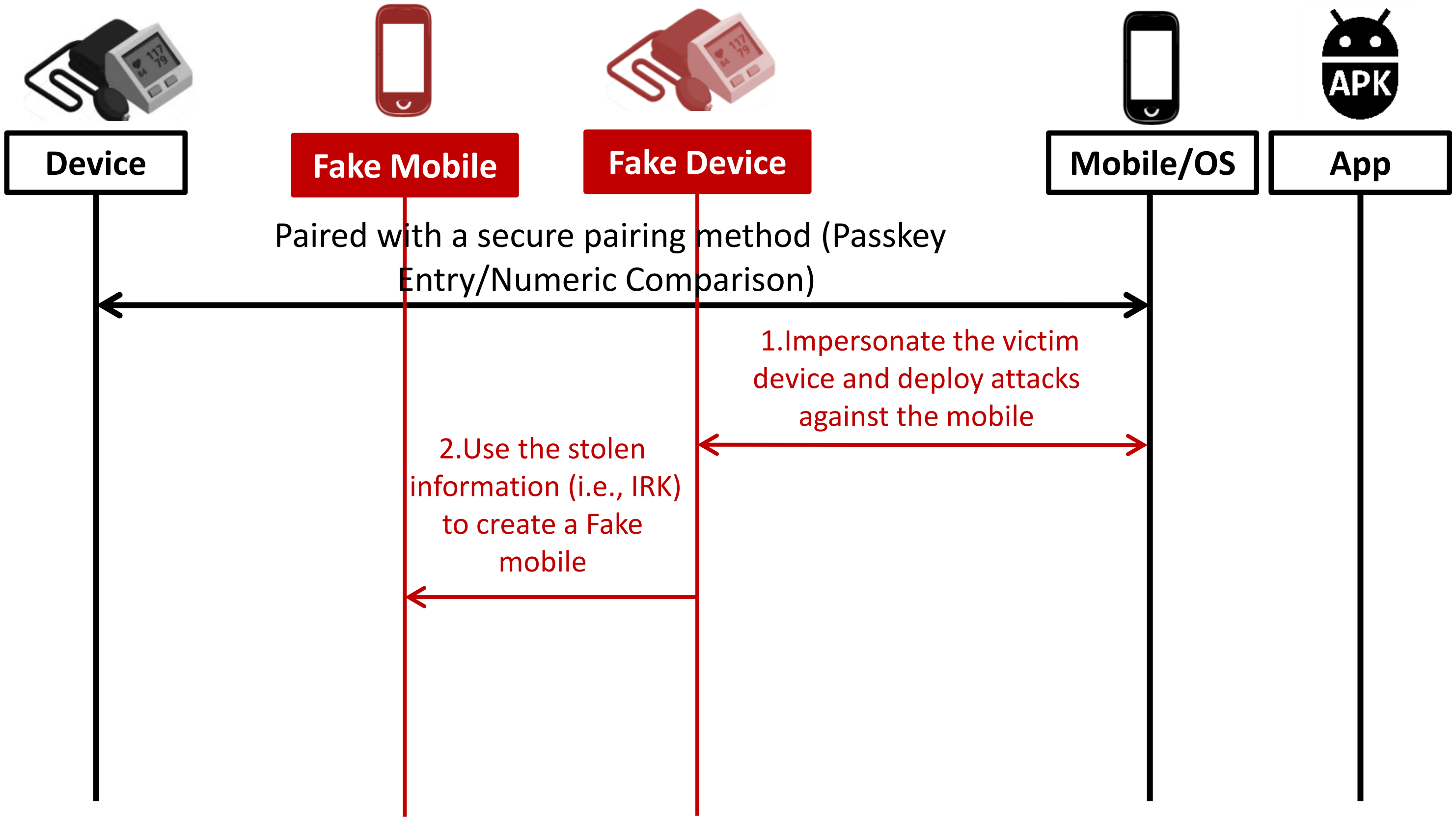
Mobile/OS

App

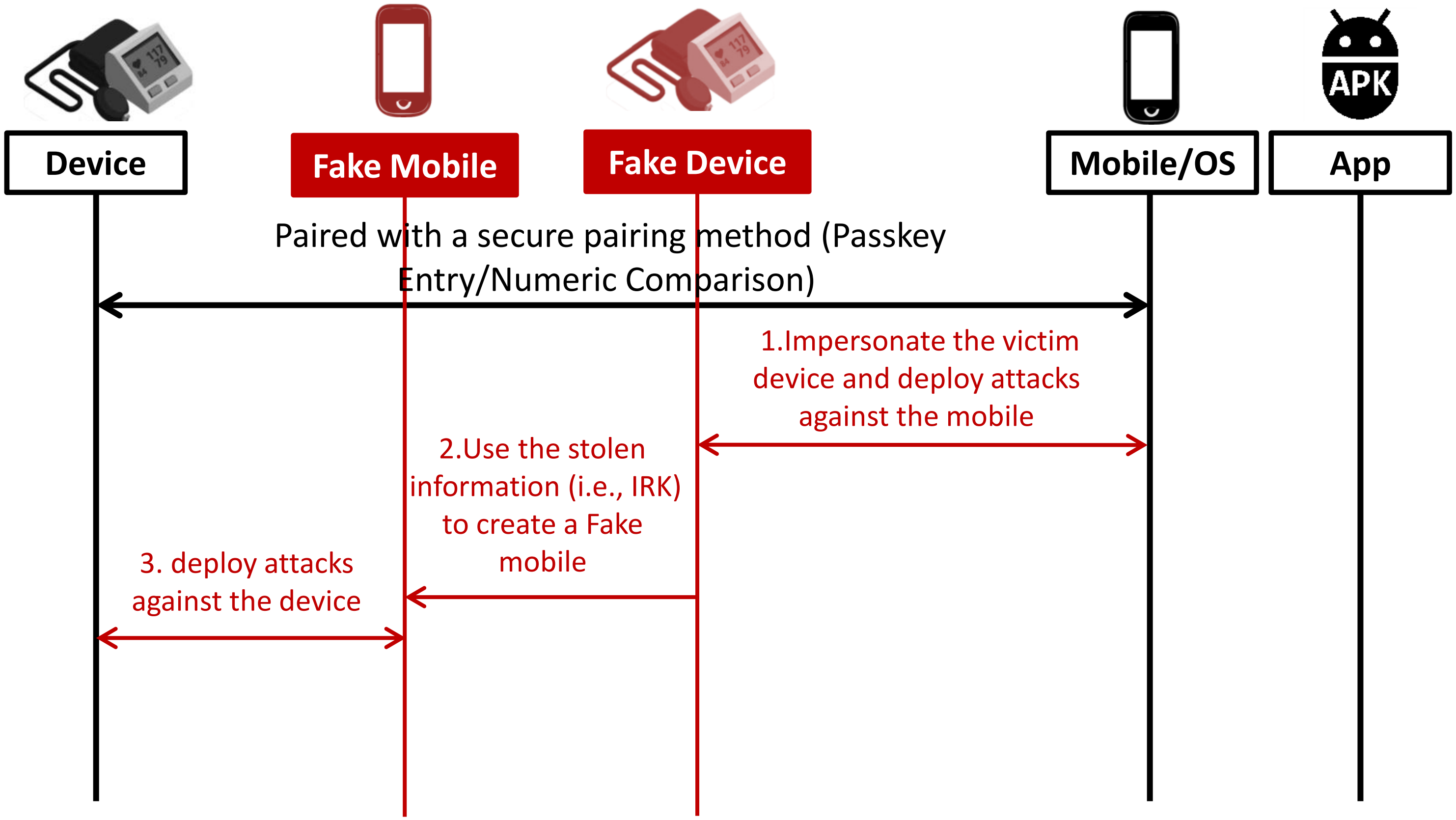
Paired with a secure pairing method (Passkey Entry/Numeric Comparison)

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

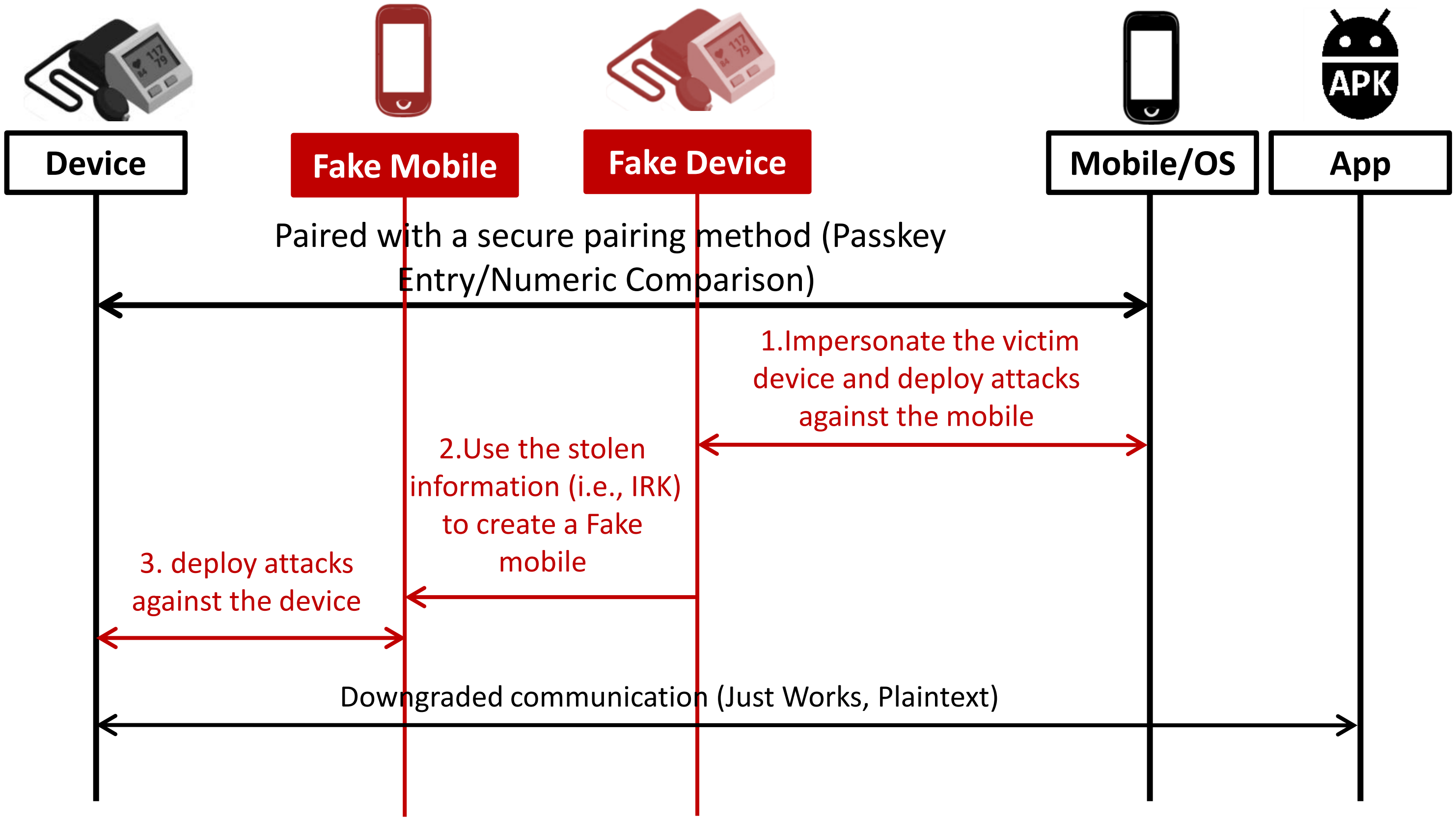
Downgrade Attacks


Device
Fake Mobile
Fake Device
Mobile/OS
App
APK
Paired with a secure pairing method (Passkey Entry/Numeric Comparison)
1.Impersonate the victim device and deploy attacks against the mobile
2.Use the stolen information (i.e., IRK) to create a Fake mobile

Downgrade Attacks


Device
Fake Mobile
Fake Device
Mobile/OS
APK
App
Paired with a secure pairing method (Passkey Entry/Numeric Comparison)
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

Downgrade Attacks



Device

Fake Mobile

Fake Device

Mobile/OS

App

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

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

2.Use the stolen information (i.e., IRK) to create a Fake mobile

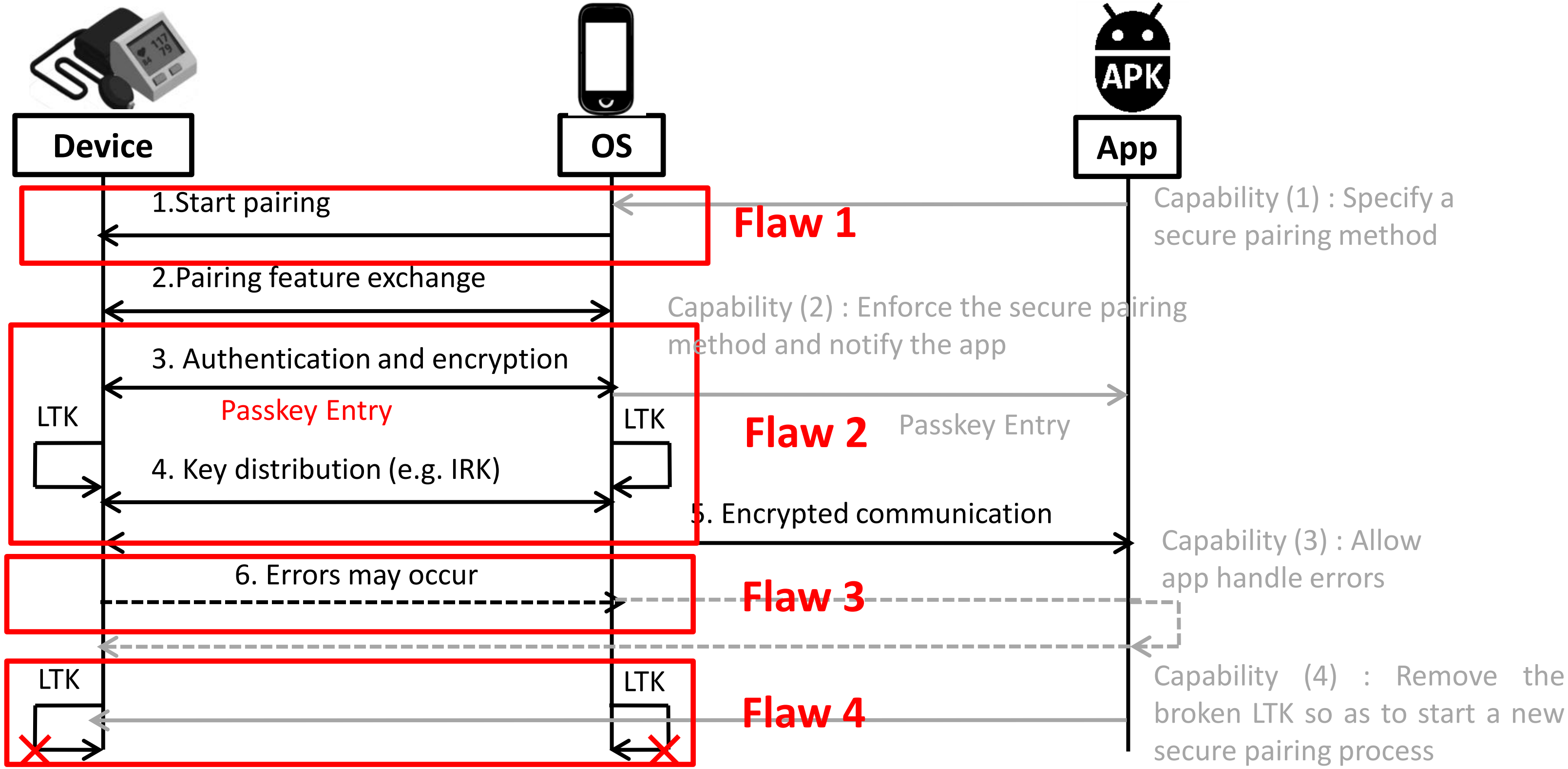
3. deploy attacks against the device

Downgraded communication (Just Works, Plaintext)

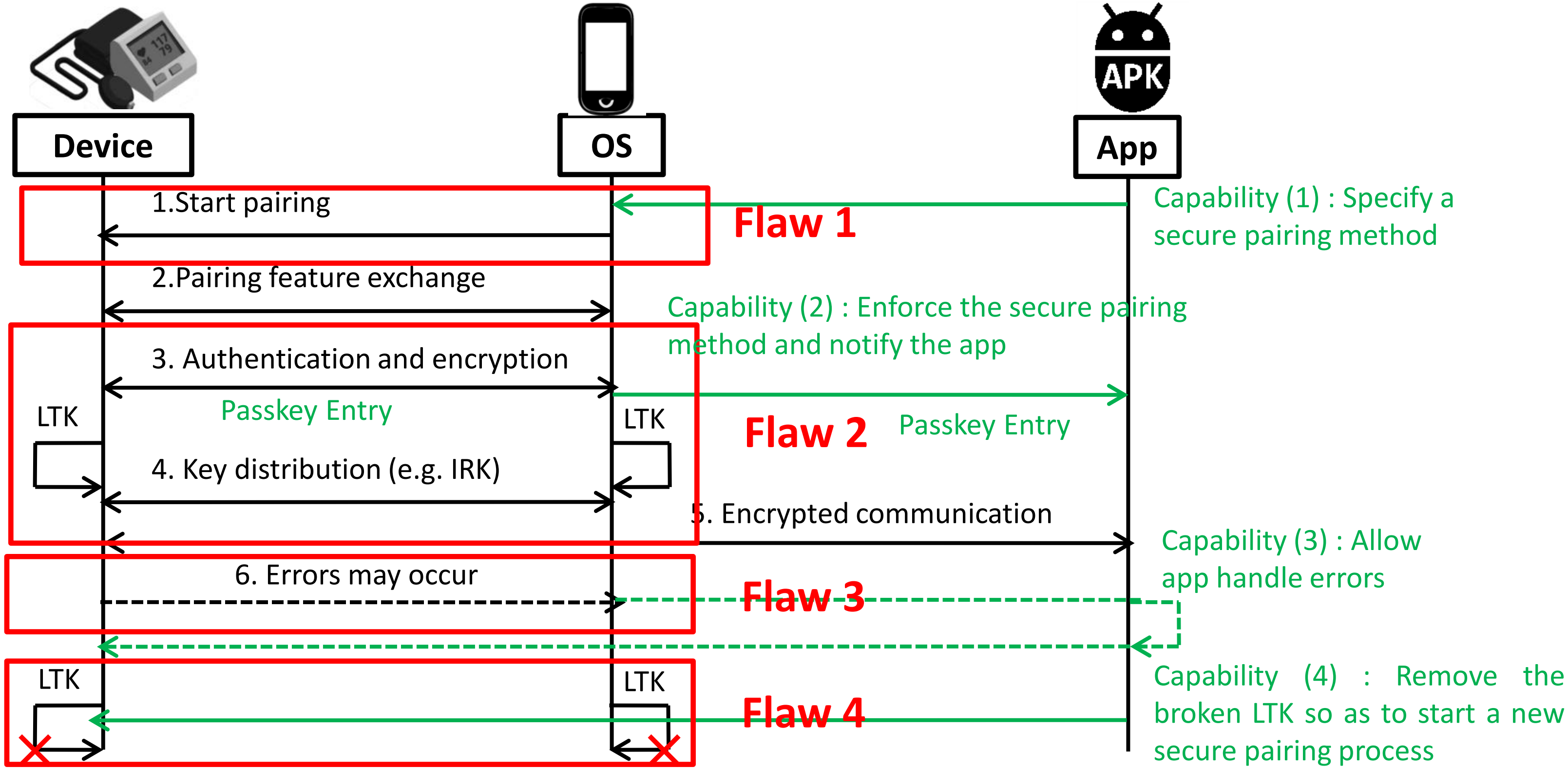
Downgrade Attacks

Attacks against Initiators	Attacks against Devices
Fake data injection	Passive eavesdropping
Sensitive data stealing	Whitelist bypassing
IRK stealing	Data manipulation
DoS attack	Man-in-the-Middle

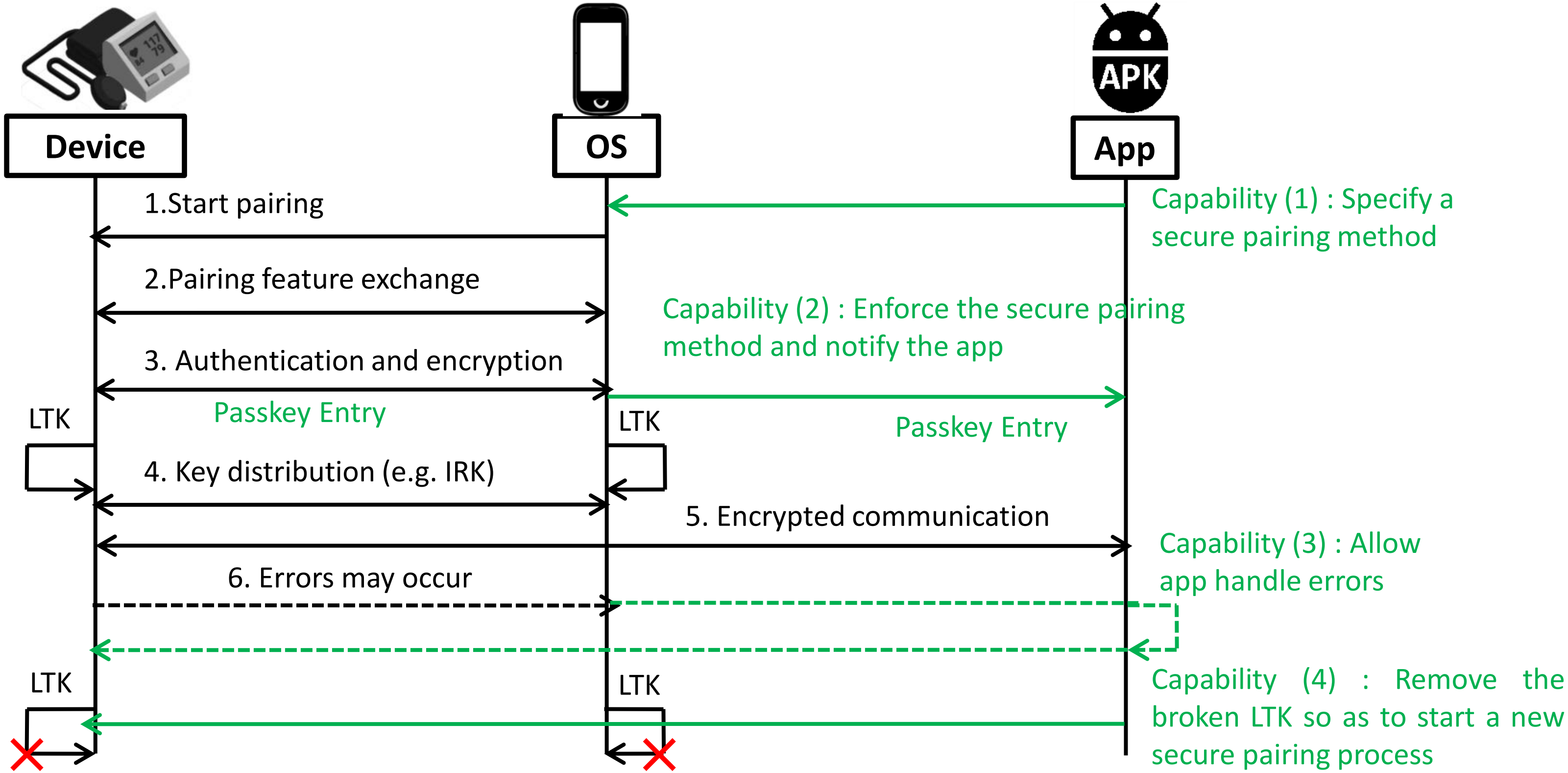
Enabling the SCO mode


Device
OS
APK
App
1.Start pairing
Flaw 1
Capability (1) : Specify a secure pairing method
2.Pairing feature exchange
Capability (2) : Enforce the secure pairing method and notify the app
3. Authentication and encryption
LTK
Passkey Entry
LTK
Flaw 2
Passkey Entry
4. Key distribution (e.g. IRK)
5. Encrypted communication
Capability (3) : Allow app handle errors
6. Errors may occur
Flaw 3
LTK
LTK
Flaw 4
Capability (4) : Remove the broken LTK so as to start a new secure pairing process

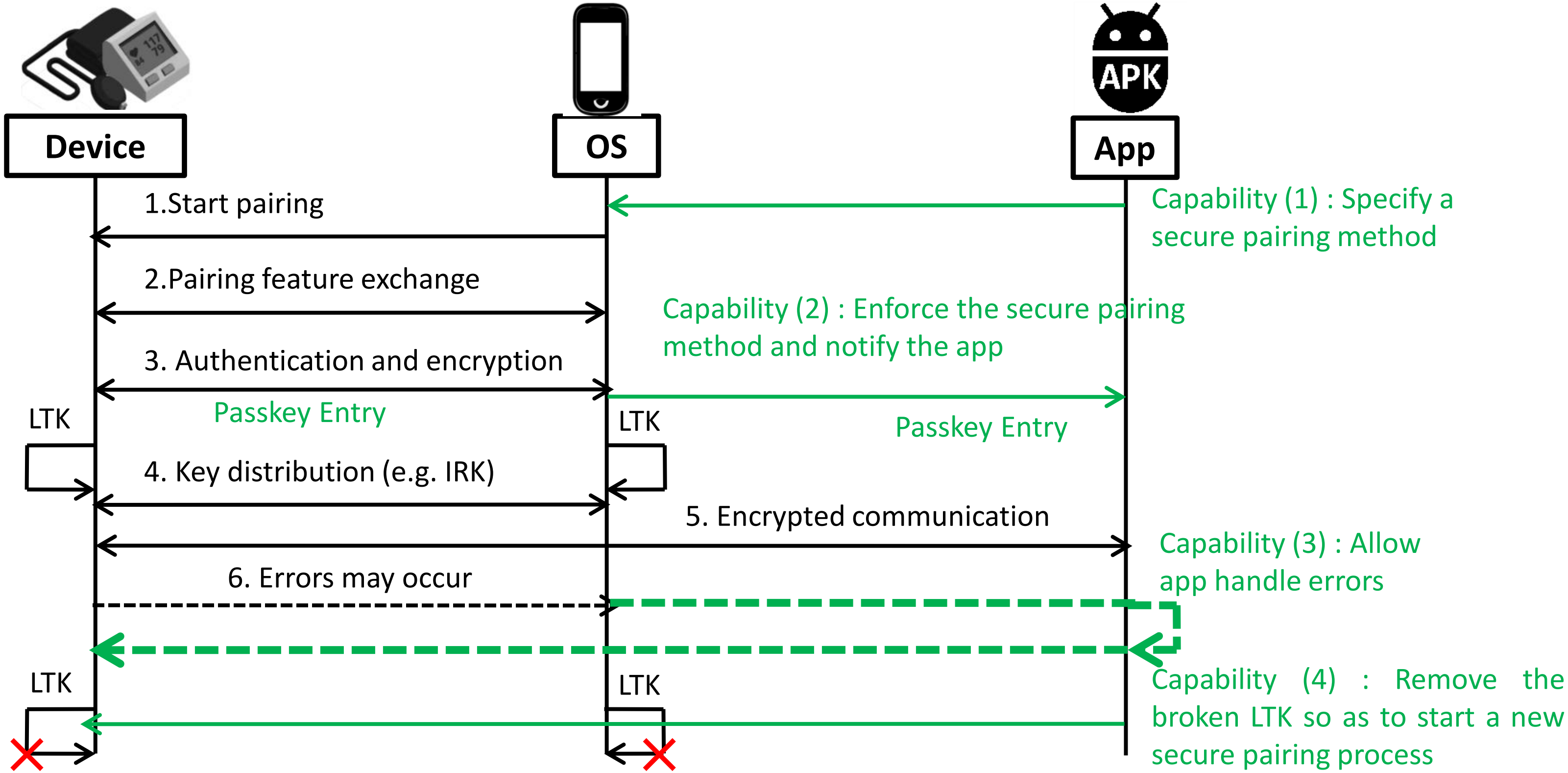
Enabling the SCO mode


Device
OS
APK
App
1.Start pairing
Flaw 1
Capability (1) : Specify a secure pairing method
2.Pairing feature exchange
Capability (2) : Enforce the secure pairing method and notify the app
3. Authentication and encryption
LTK
Passkey Entry
LTK
Flaw 2
Passkey Entry
4. Key distribution (e.g. IRK)
5. Encrypted communication
Capability (3) : Allow app handle errors
6. Errors may occur
Flaw 3
LTK
LTK
Flaw 4
Capability (4) : Remove the broken LTK so as to start a new secure pairing process

Enabling the SCO mode


Device
OS
App
APK
1.Start pairing
Capability (1) : Specify a secure pairing method
2.Pairing feature exchange
Capability (2) : Enforce the secure pairing method and notify the app
3. Authentication and encryption
LTK
Passkey Entry
LTK
Passkey Entry
4. Key distribution (e.g. IRK)
5. Encrypted communication
Capability (3) : Allow app handle errors
6. Errors may occur
LTK
LTK
Capability (4) : Remove the broken LTK so as to start a new secure pairing process

Enabling the SCO mode


Device
OS
APK
App
1.Start pairing
Capability (1) : Specify a secure pairing method
2.Pairing feature exchange
Capability (2) : Enforce the secure pairing method and notify the app
3. Authentication and encryption
LTK
Passkey Entry
LTK
Passkey Entry
4. Key distribution (e.g. IRK)
5. Encrypted communication
Capability (3) : Allow app handle errors
6. Errors may occur
LTK
LTK
Capability (4) : Remove the broken LTK so as to start a new secure pairing process

Attacks against Initiator

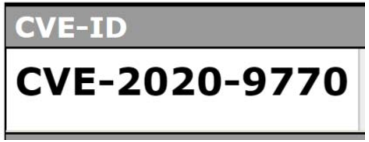
OS Name	Flaw 1	Flaw 2	Flaw 3	Flaw 4
Android	✓	✓	✓	✓
macOS	✓	✓	✓	✓
iOS	✓	✓	✓	✓
Windows	✓	✓	✗	✗
Linux	✓	✓	✗	✗

Flaws across OSes

Brand	Version
Samsung Galaxy S8+	Samsung Official Android 7.0
Google Pixel 2	AOSP Android 8.0
Samsung Tablet	Samsung Official Android 8.1
Samsung Note 8	Samsung Official Android 8.1
Google Pixel 2	AOSP Android 9.0

Tested Android mobiles

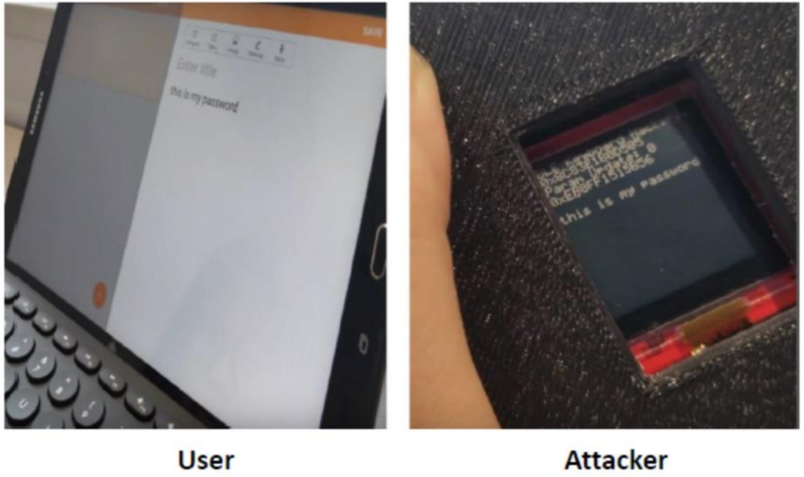
Common Vulnerabilities and Exposures


CVE-ID

CVE-2020-9770

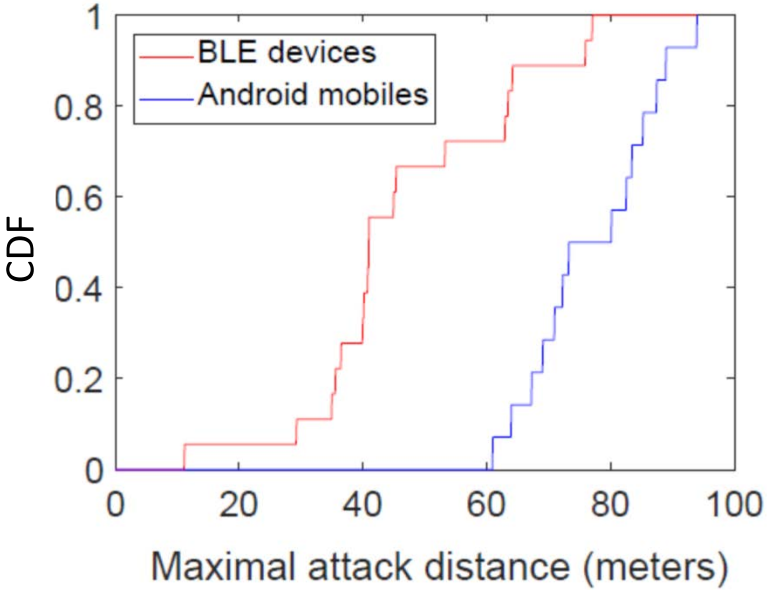
Attacks beyond Initiator

The Tested BLE devices

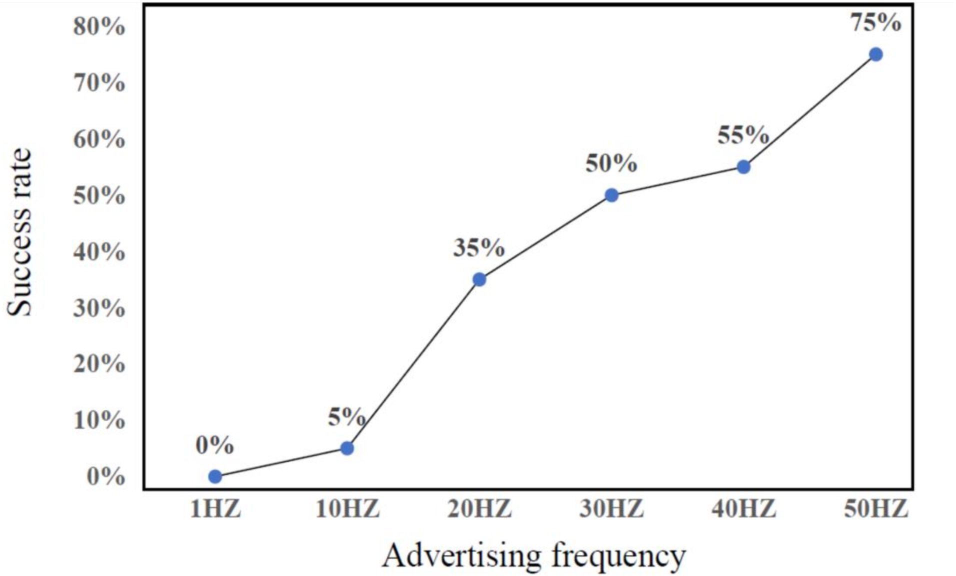


MITM attack against BLE keyboards

Attacks beyond Initiator (cont'd)



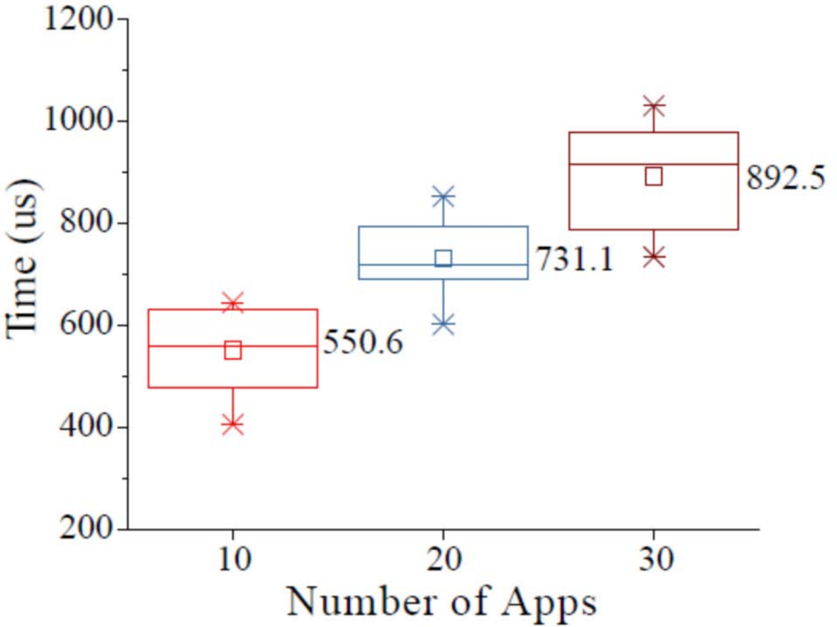
Maximal attack distance



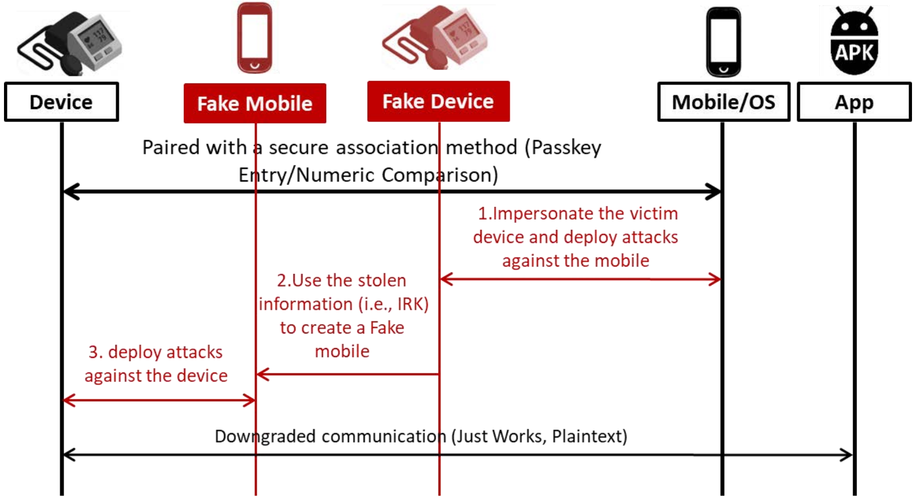
Success rate vs. advertising frequency

Countermeasures

Android 8.0



Summary


Device
Fake Mobile
Fake Device
Mobile/OS
APK
App
Paired with a secure association method (Passkey Entry/Numeric Comparison)
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
Downgraded communication (Just Works, Plaintext)

Downgrade Attacks

- **No mutual authentication:** SCO mode is not enforced for the pairing initiator, e.g., a mobile
- **Enabling SCO:** Four capabilities is required at initiator;
- **Mutual authentication:** SCO mode must be mutually enforced so as to achieve the strongest security

Impact of Downgrade Attacks

- **Initiators:** Android, iOS, macOS, Windows, Linux are subject to our attacks
- **Devices:** We analysed 18 BLE devices; none of them are secure;

THANK YOU !

Breaking Secure Pairing of Bluetooth Low Energy Using Downgrade Attacks

Yue Zhang
zyueinfosec@gmail.com
Joint work w/ Jian Weng, Rajib Dey , Yier Jin,
Zhiqiang Lin, and Xinwen Fu

暨南大学 JNU 1906 JINAN UNIVERSITY

UNIVERSITY OF CENTRAL FLORIDA 1963

UMASS LOWELL

THE OHIO STATE UNIVERSITY

UF UNIVERSITY of FLORIDA