HITB SECCONF 2023 Phuket

The tragedy of Bluetooth Low Energy

Linfeng Xiao , Dalin Yang
Security Engineer, Xiaomi

Xiaomi ShadowBlade Security Lab

We are Xiaomi's security team. Our mission is to enhance the security of Xiaomi's products. We focus on:

- Researching advanced security technologies
- Developing automated security platforms
- Conducting security assessments and protection for smart devices

Our team members have won multiple top prizes in major security competitions such as Geekpwn, Tianfu Cup and Butian Cup.

Grand Theft Bicycle

Some proprietary protocols based on BLE implemented by certain manufacturers have vulnerabilities. Attackers can renew the timestamp of expired commands to keep them valid and easily unlock smart U-locks.

BLE Analysis Basics

- Analyzing BLE packets by sniffing BLE communications using nrf52840 dongle and Wireshark.
- A GATT Server includes multiple different Services, and each Service can contain multiple Characteristics.
- BLE basic operations: Read\Write\Notify\Indicate

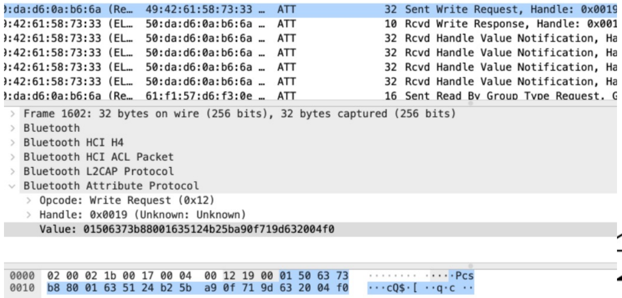
Command structure

Use HMAC-SH1 to hash the Command and TimeStamp.

KEY Hf********z

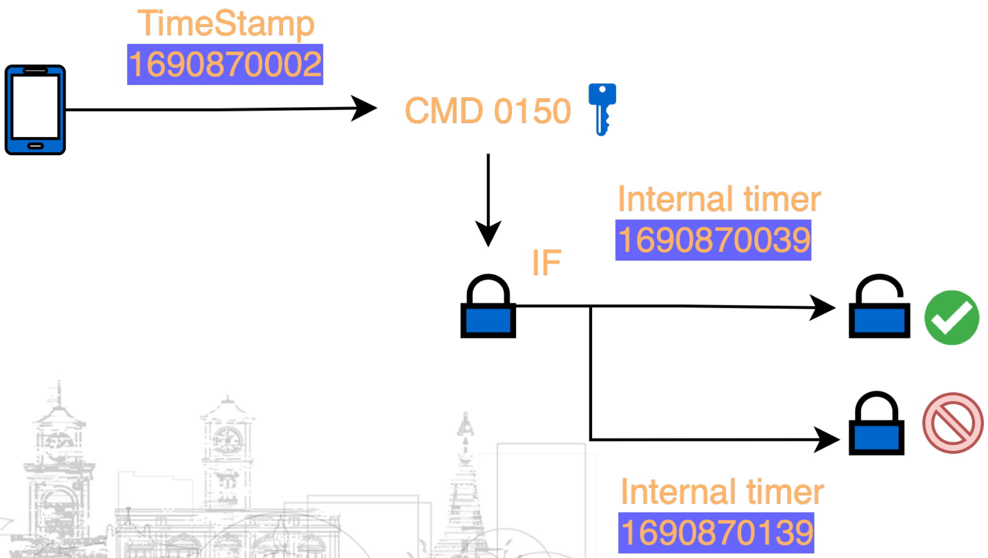
Command TimeStamp → HMAC-SH1 sign

Unlock 01506373b88001635124b25ba90f719d632004f0



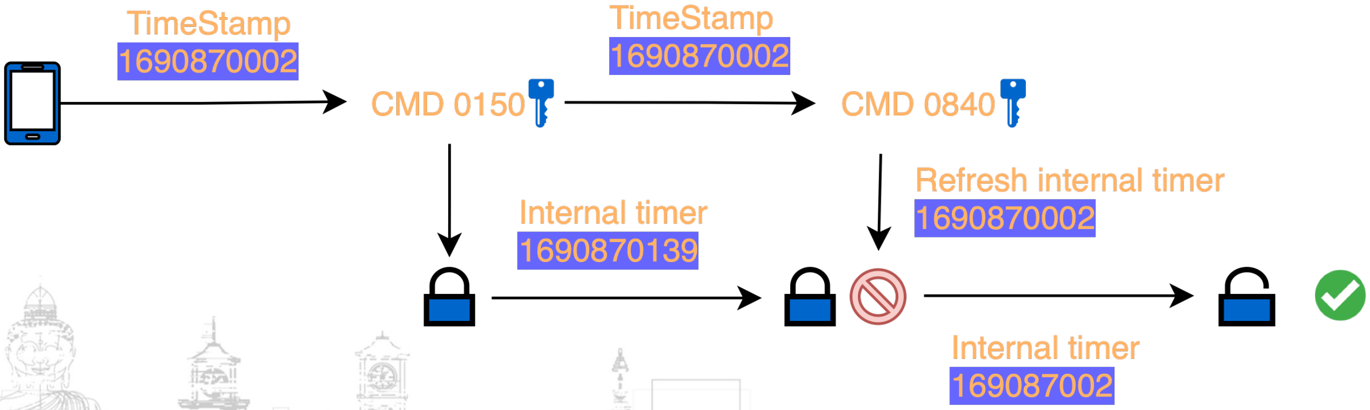
Normal unlock process

When the mobile phone's timestamp is synchronized with the U-lock's timestamp, the command can be executed normally.



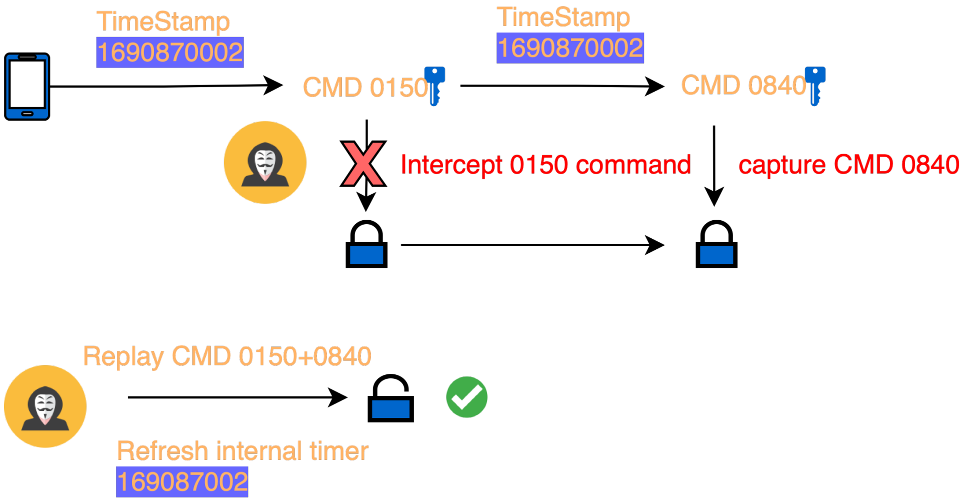
Timestamp rollback

To prevent timestamp desynchronization from causing unlock failure, this lock is designed with a timestamp rollback operation.

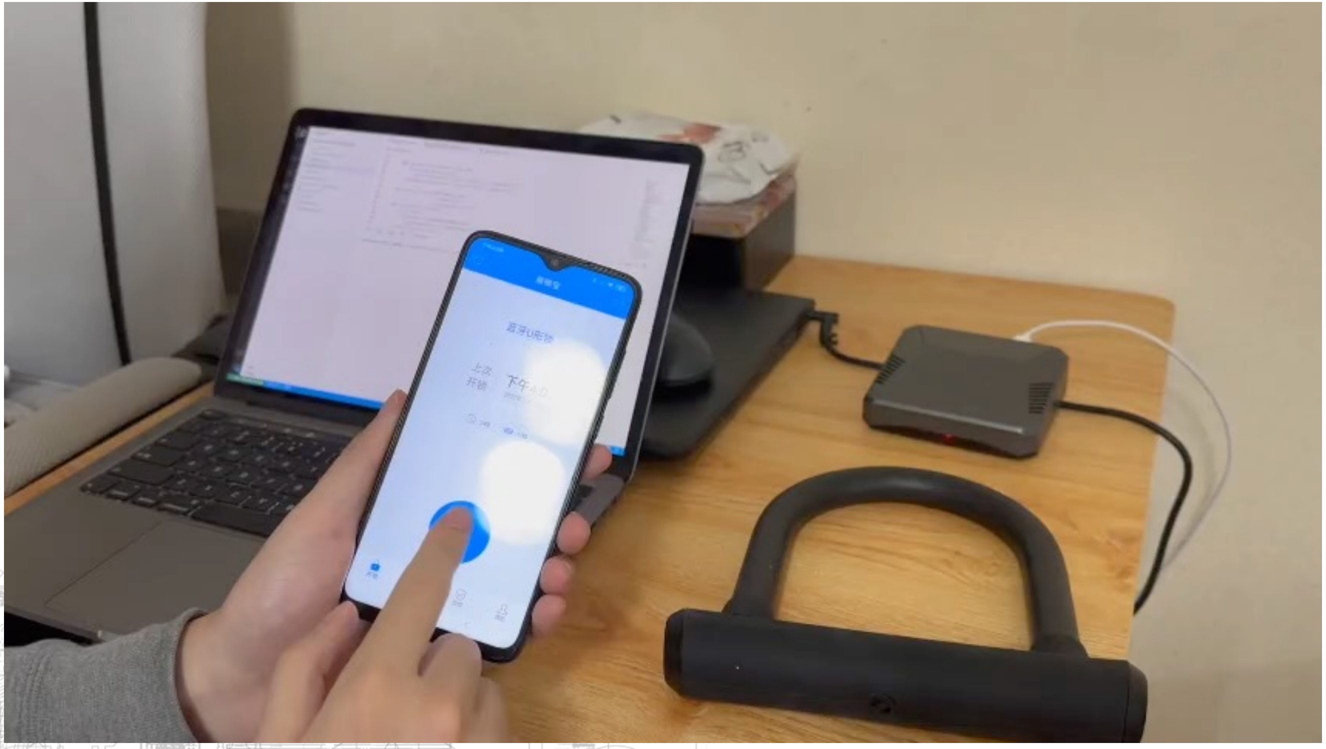


Replay attack again

- Hijack the 0150 command to make the APP think the timestamp is not synchronized.
- The APP sends the 0840 command to refresh the timestamp, capturing the 0840 command.
- Replay the 0150 command and 0840 command to unlock.



Video demonstration



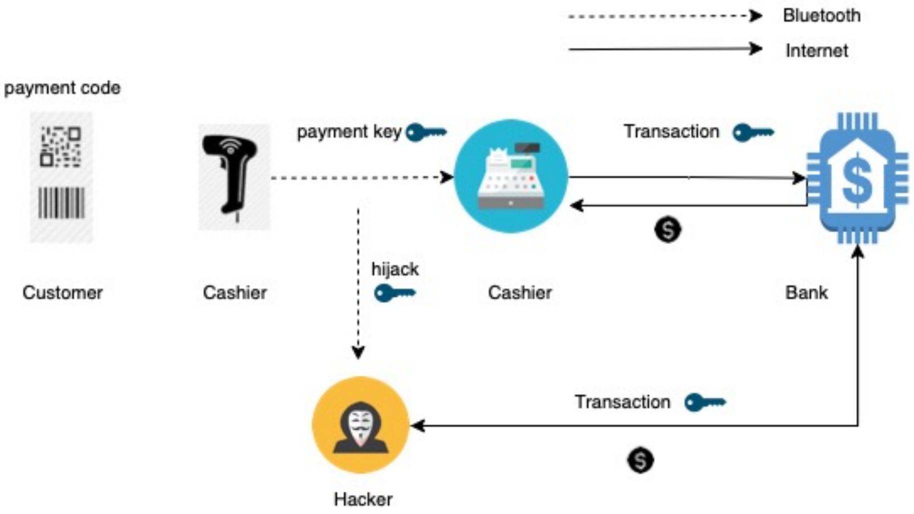
Robbery from afar

When shopping at convenience stores in China, the most common payment method is for the clerk to scan the payment code to receive payment. Attackers can hijack BLE to modify the payment amount and then take the money and run.

Attack Surface

Objective: Steal user payment credentials.

Means: Directly attack the BLE protocol.



Common wireless communication

- No frequency hopping or simple hopping logic
  - WIFI and radio control often work on fixed frequencies.
  - NRF24L01 hopping sequence is predictable.
- Lacks link layer design.
  - No connection or interaction is established.
  - The channel is insecure. Commands can be sent by seizing the channel. Whoever shouts the loudest is heard.

Low energy Bluetooth communication

- The BLE Controller implements "connections".
  - The link layer implements secure channel communication.”
  - Connection events" are implemented to ensure BLE devices can "interact".
- Frequency hopping complexity is high.
  - It is difficult to follow the hopping without capturing the connection packets. For example, when using an nrf52 dongle to capture packets, hopping can only be tracked normally after capturing the CONNECT_IND packet. It is very difficult to capture an established connection out of thin air!

Link layer packets

Header definition:

- ChSel: Channel selection algorithm, i.e. hopping algorithm #1 or #2
- TxAdd: Indicates whether the device address of the initiator in the InitA field is public (TxAdd = 0) or random (TxAdd = 1)
- RxAdd: Indicates whether the device address of the advertiser in the AdvA field is public (RxAdd = 0) or random (RxAdd = 1).

LSB → MSB

Preamble	Access-Address	PDU	CRC	Constant Tone Extension
(1 or 2 octets)	(4 octets)	(2-258 octets)	(3 octets)	(16 to 160 µs)

LSB → MSB

Header	Payload
(16 bits)	(1-255 octets)

Connection request

- Access Address: identifier for establishing link layer communication.
- CRCInit: CRC calculation initialization value.
- Interval: Connection interval connInterval = Interval * 1.25 ms.
- Timeout: Timeout tolerance connSupervisionTimeout = Timeout * 10 ms.
- Channel map: Contains channel map indicating used and unused data channels. Bit value 0 indicates channel is unused. Bit value 1 indicates channel is used.
- Hop: Hopping parameter, random value between 5 to 16 used in data channel selection algorithm.

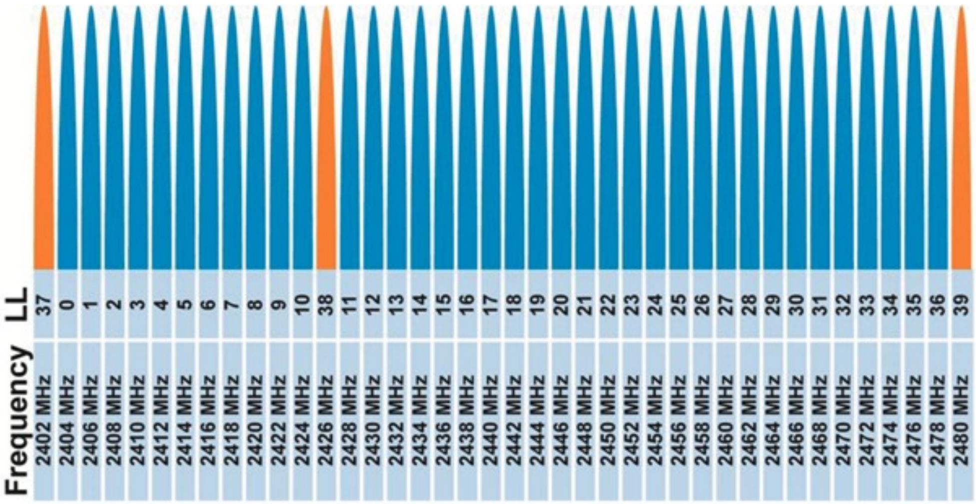
LLData									
AA (4 octets)	CRCInit (3 octets)	WinSize (1 octet)	WinOffset (2 octets)	Interval (2 octets)	Latency (2 octets)	Timeout (2 octets)	ChM (5 octets)	Hop (5 bits)	SCA (3 bits)

Obtain Access Address and CRCInit

- The Access Address can be obtained by sniffing over-the-air packets.
- CRCInit can be calculated through empty packets (crc24).

Obtain Channel Map

Listen on the same frequency band, stay on each channel for 4 seconds, total time cost 4*37 = 148s



Obtain Connection Interval

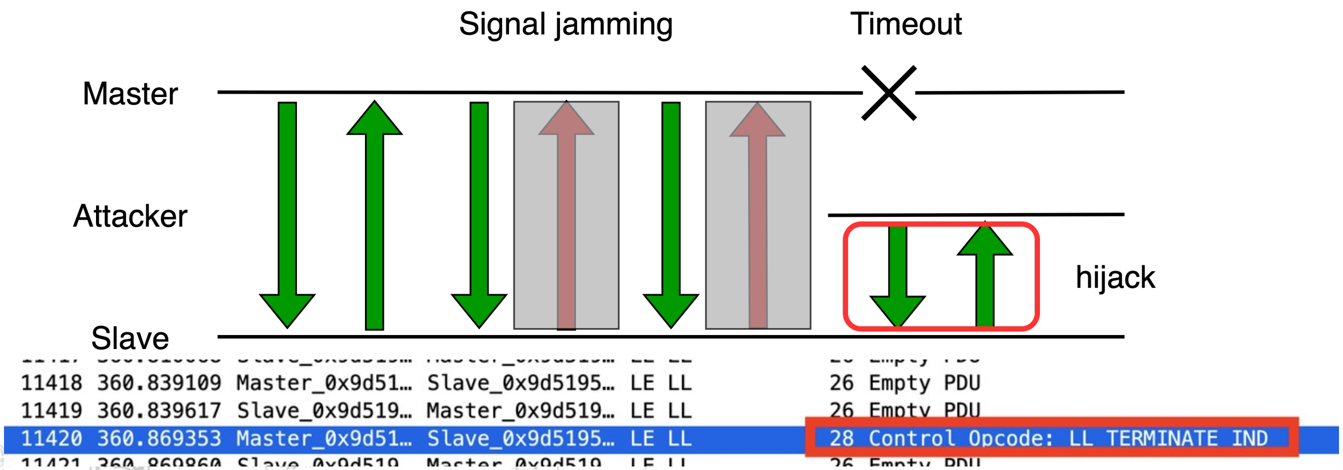
- Hopping algorithm #1: **Fn+1 = (Fn + hop) mod 37**.
    - Periodicity of modulo operation.
- Find a suitable channel, divide the time interval between two monitored packets by 37.

Obtain Hop Increment

- Hopping algorithm #1: **Fn+1 = (Fn + hop) mod 37**.
- It is a random value between 5-16, test all possible values

The key to hijacking

Utilize the Supervision Timeout mechanism. After timeout, the Master will actively send LL_TERMINATE_IND to disconnect the Bluetooth connection.



Imperceptible invasion

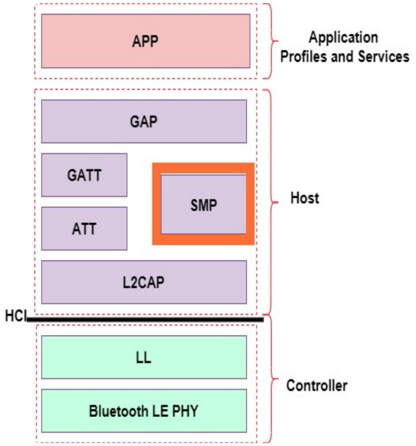
Attacks against BLE keyless entry, relaying from the Controller layer, able to bypass conventional detection methods. This could then be used to unlock and drive away any car that has BLE keyless entry enabled.

BLE Controller

Conventional BLE relay tools mainly perform relay attacks on the GAP layer in the host layer. Device manufacturers can effectively prevent relay attacks by using the built-in link layer encryption feature of BLE.

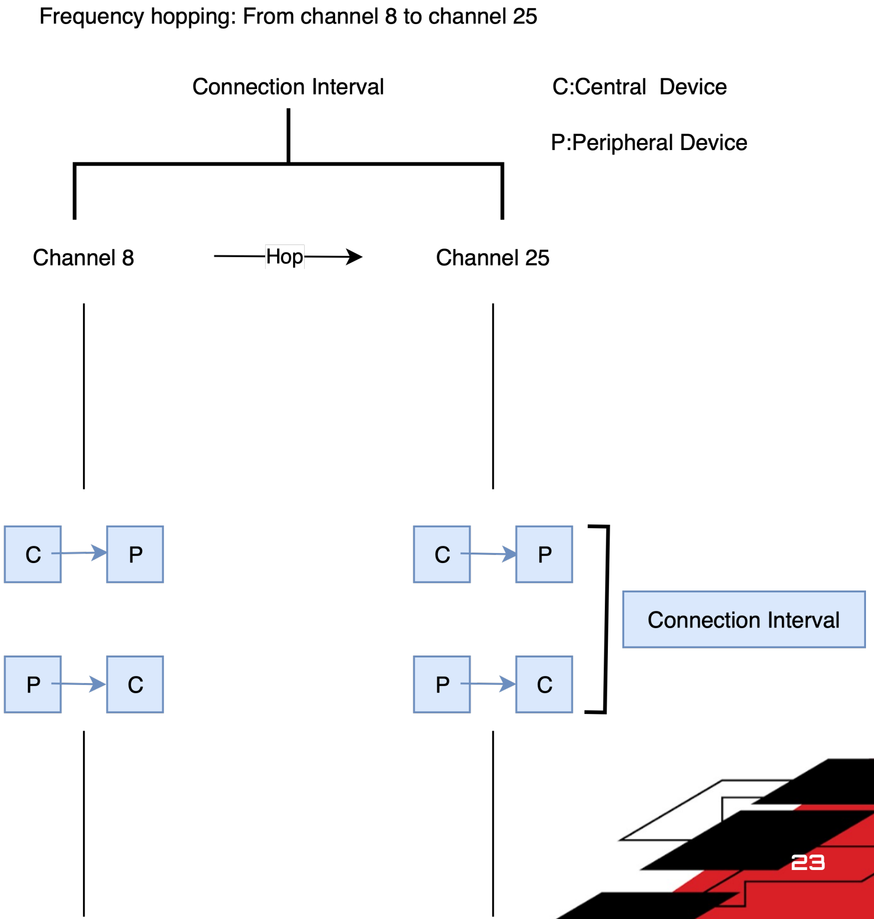
So our attacking focus naturally shifts from the Host to the Controller.

- Can we try relay attacks on the Controller?
- BLE is a kind of radio communication.
- Can we refer to RF relay solutions and directly amplify the signal?



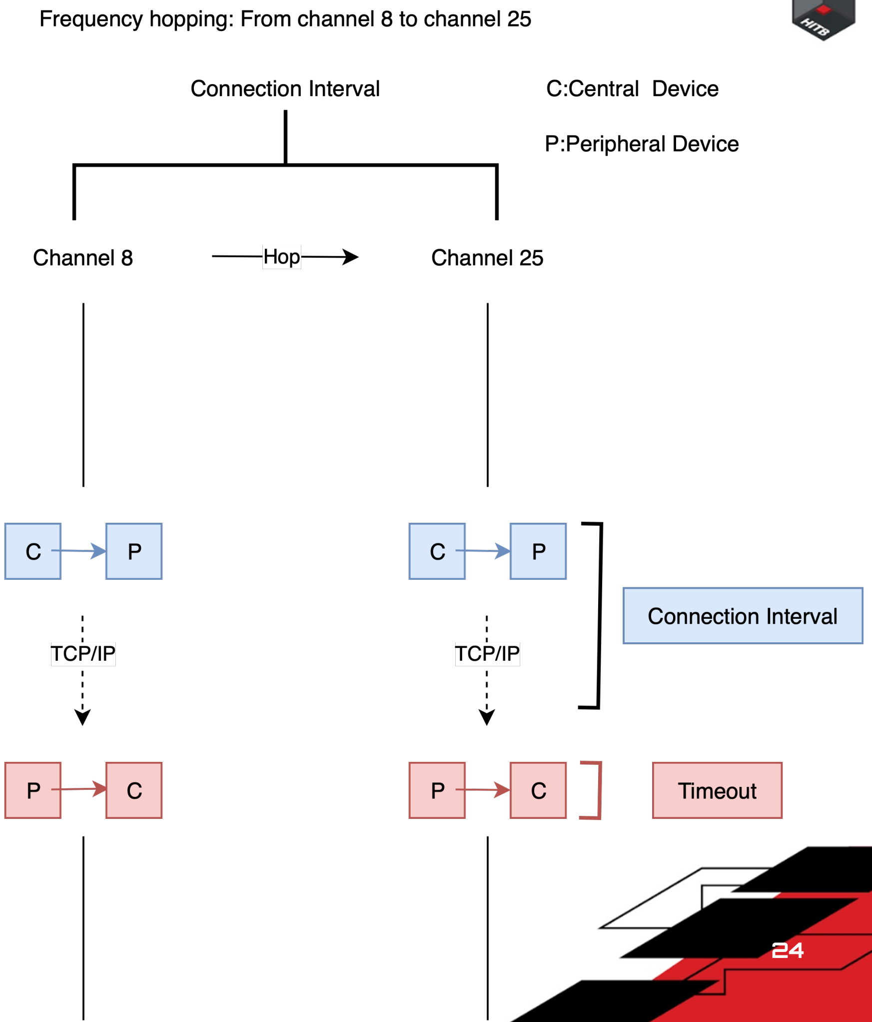
Frequency hopping

The right figure shows a connection event of a BLE device. In this event, the initial communication occurs on Channel 8. The Central device sends information to the Peripheral device and waits for a response within the ConnInterval time frame. After receiving the request, the Peripheral device returns a response packet through the same channel. After this communication is completed successfully, the communication hops to Channel 25 and continues the same operation, successfully completing the next communication.



Frequency hopping

The right shows another connection event of a BLE device. This time we relay the BLE signals over TCP/IP. The initial communication occurs on Channel 8, the Central device sends information to the Peripheral device and waits for a response. Due to the latency of TCP/IP, when the device response returns, the Central device has already hopped to Channel 25 waiting for the next communication. Since the timeout event exceeds the limit (connSupervisionTimeout = Timeout * 10 ms), the Central device thinks the Peripheral device is disconnected and sends a terminate command, causing the communication to fail.



The answer lies in the problem itself

The most important part is the Link Layer PDU, which carries all Host layer communication data.

The other parameters are for maintaining the link, so we only need to relay the Linker Layer PDU while discarding Access Address and CRCInit.

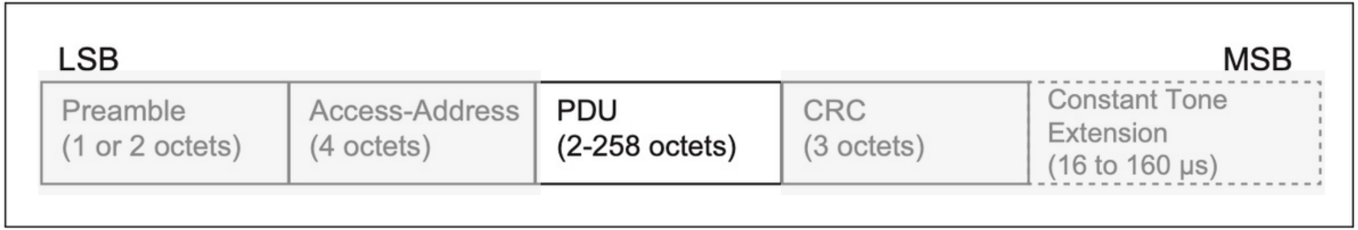
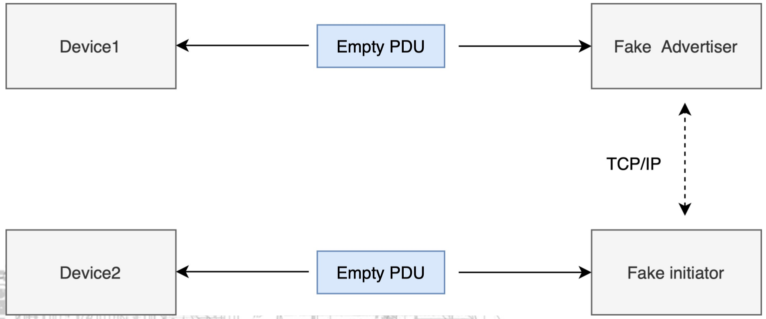


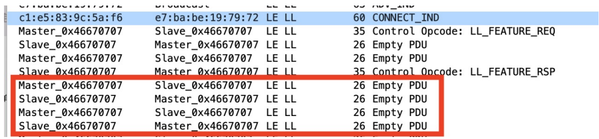
Figure 2.1: Link Layer packet format for the LE Uncoded PHYs

Bypassing the Maginot Line

- Discard the Access Address parameter in the data packets, making the devices on each side establish link connections separately. The two sets of devices negotiate connection parameters separately.

- Use empty packets to fill each Connection Interval (C->P and P->C), filling with data if there is interactive data. Each connection event (connInterval) is no longer affected by TCP/IP latency, avoiding connSupervisionTimeout caused by no response.



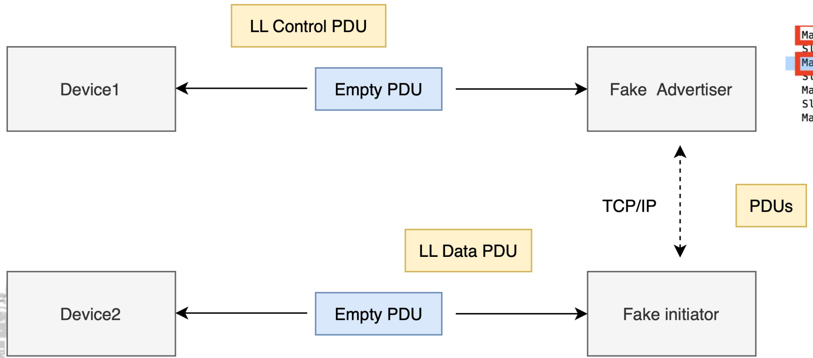


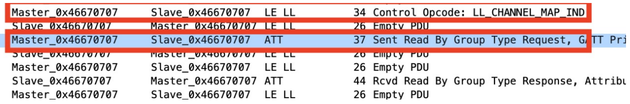
c1:e5:83:9c:5a:f6	e7:ba:be:19:79:72	LE LL	60	CONNECT_IND
Master_0x46670707	Slave_0x46670707	LE LL	35	Control Opcode: LL_FEATURE_REQ
Slave_0x46670707	Master_0x46670707	LE LL	26	Empty PDU
Master_0x46670707	Slave_0x46670707	LE LL	26	Empty PDU
Slave_0x46670707	Master_0x46670707	LE LL	35	Control Opcode: LL_FEATURE_RSP
Master_0x46670707	Slave_0x46670707	LE LL	26	Empty PDU
Slave_0x46670707	Master_0x46670707	LE LL	26	Empty PDU
Master_0x46670707	Slave_0x46670707	LE LL	26	Empty PDU
Slave_0x46670707	Master_0x46670707	LE LL	26	Empty PDU

Bypassing the Maginot Line

- After establishing connections with empty packets, let Device1 and Device2 chat with each other. The devices on both sides naturally send Link Layer Control PDUs or Link Layer Data PDUs to each other, and we just relay these Link Layer PDUs.

- In actual practice, you need to implement a Linker Layer code yourself. To establish the most basic Linker Layer link, it is necessary to implement CHM, hopping algorithm, etc. It is recommended to refer to the Bluetooth Core Specification Vol 6 Low Energy Controller and the open source project Sniffle.





Master_0x46670707	Slave_0x46670707	LE LL	34	Control Opcode: LL_CHANNEL_MAP_IND
Slave_0x46670707	Master_0x46670707	LE LL	26	Empty PDU
Master_0x46670707	Slave_0x46670707	ATT	37	Sent Read By Group Type Request, GATT Pri
Slave_0x46670707	Master_0x46670707	LE LL	26	Empty PDU
Master_0x46670707	Slave_0x46670707	LE LL	26	Empty PDU
Slave_0x46670707	Master_0x46670707	ATT	44	Rcvd Read By Group Type Response, Attribu
Master_0x46670707	Slave_0x46670707	LE LL	26	Empty PDU

Against encryption

- Some vendors switch to PHY 2M channels after link encryption to evade snooping or relaying. Specifically, once a channel is encrypted, the LL Control packets will be encrypted, causing us to be unable to parse their content. LL Control contains important data about channel state and hopping info, so we cannot accurately track channel changes.
- To prevent losing track due to hopping, we proactively switch to the PHY 2M channel before encryption. By adopting this strategy, the PHY 2M channel is fixed before link encryption, thus avoiding the impact of hopping on tracking.

Video demonstration

Research tools and BLE security summary

According to previous research, we engineered the results into test tools, and summarized some security defenses about the BLE link layer and GATT layer.

Relay Protective measures

Is this anti-relay measure effective?

WHY ANTI-RELAY WORKS - 1

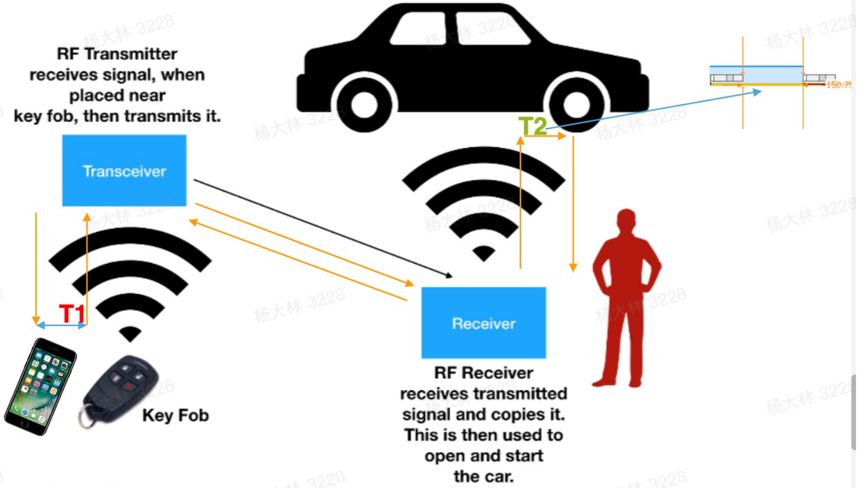
- $T2 >= 150us$
- $T1 = T2 + T_{others}$
- $T1\_max = (T_{IFS} + 2us) + 2*T_{range}$

  $= (150us + 2us) + 2*2D*4ns$

  $= 152us + 16*Dns$

  $= 152us + 16*100ns$

  $= 153.6us$ (max range, 100M)
- $T_{others} = T1 - T2 = 3.6us$



3.6us might be too big challenge to have info relayed

BLE Link Layer Relay Difficulties

The PDU data structure is as follows:

```python
send_dict = {
        #"pdu_data":data1[4:],
        "time":msg.ts_epoch,
        "pdu_data":data1,
        "channel": msg.chan,
        "accaddr": aa.upper(),
        #"type":pdu_type,
        "pdu_type":dpkt.pdutype,
        "phy":msg.phy
    }

if send_dict['pdu_type']== "LL DATA CONT" and send_dict['pdu_data'][4:] == "": #跳过空包
    continue
```

The Master Slave that established the connection maintains the connection by itself. Only introducing friends, not providing a way to maintain a romantic relationship.

BLE Link Layer Relay Difficulties

◉ BLE link layer encryption and switch link layer parameters, such as frequency hopping algorithm and PHY， BLE5.0 and its subsequent versions add a new physical layer rate of 2Mbps and support multiple communication rates of 125kbps/500kbps/1Mbps/2Mbps.

◉ apple MAC address random.

◉ BLE link layer encryption and configure moredata mode.

◉ others： auxiliary whitelist judgment and OOB authentication.

BLE Security Detection Tool

Link layer relay

The link layer relay mentioned above realizes portable and automatic detection.

Controller Hijack

Hijack and tamper with unencrypted link layer data.

Sniffer&Fake adv

Sniff Advertising packets, Fake advertising and active connection requests.

Identify UUIDs

Identify non-standard (private) UUID services.

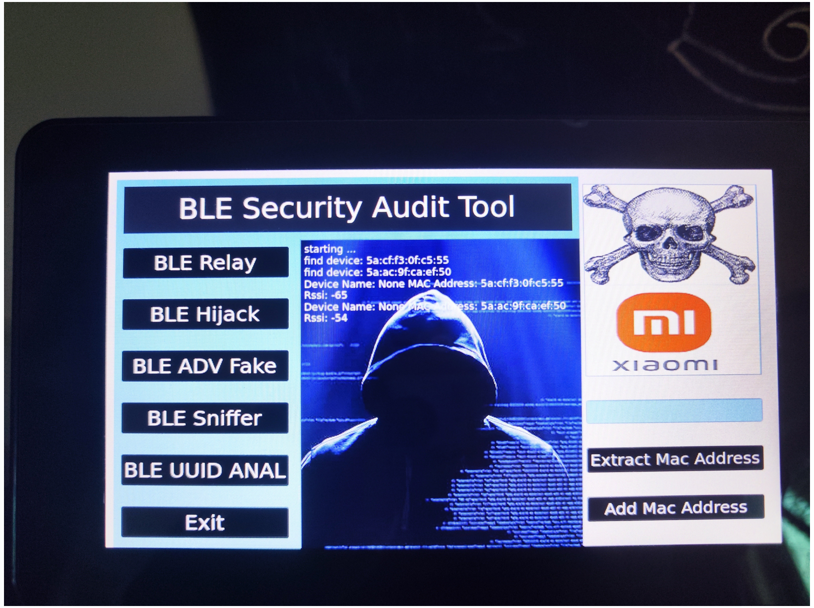
Advantage

Integrated BLE test cases

Integrates BLE sniffing, BLE relay, BLE hijacking, BLE adverting fake, UUID risk identification. Follow-up will continue to integrate packets capture, GATT layer fuzz, BLE historical CVEs test, etc.

Portable

It is more suitable for outdoor testing such as harsh network environment and no power supply. Such as underground garages, outdoor parking lots, etc.



Background

Hardware design

Due to the long hardware development cycle, we adopt modular stacking in the tool hardware architecture to realize lithium battery power supply, 4G networking, python development environment, and touch screen automation control.

01 raspberry pi 4B

02 NRF51822 dongle

03 TI CC2652 module

04 4G USB dongle

Background

Software design

Based on Bluetooth module firmware

and PC application development.

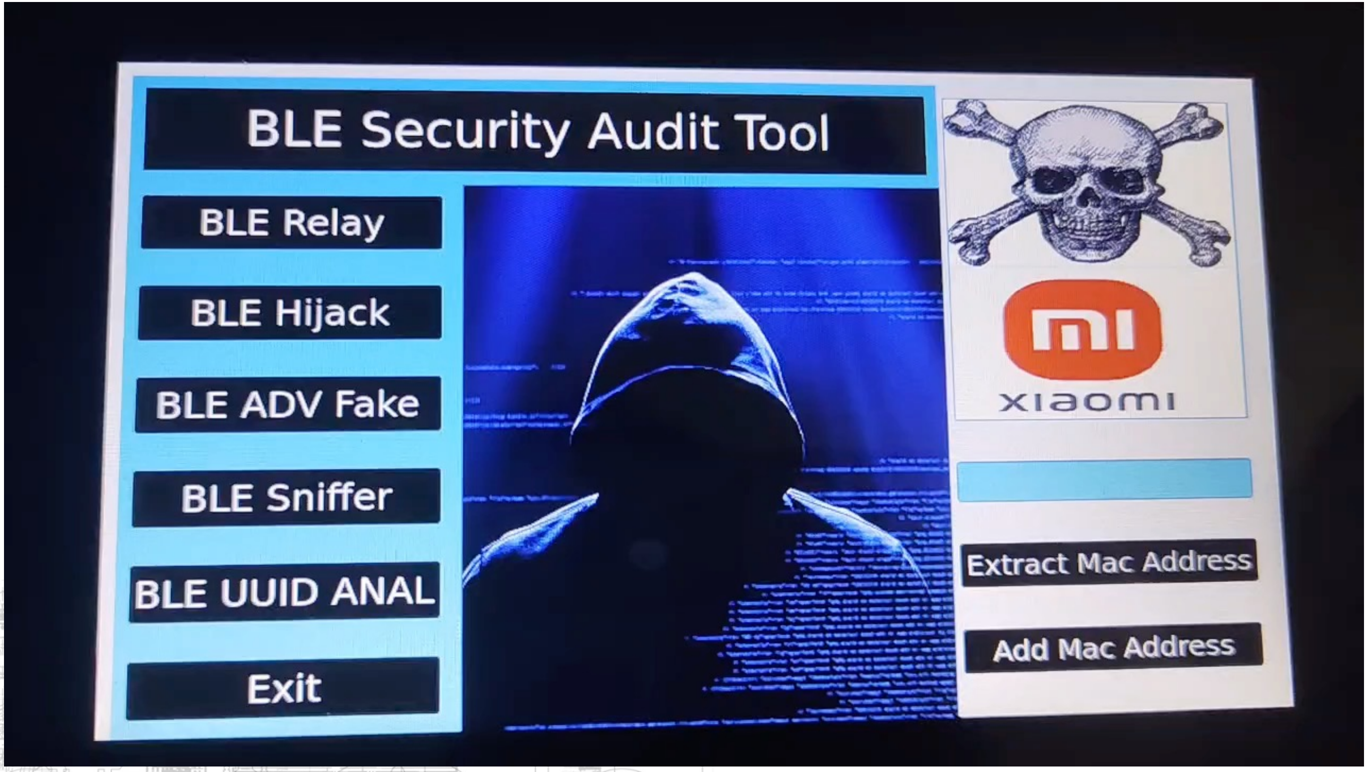
01 QT display

02 python3

03 NCC Sniffle

04 Controller Hijack

BLE relay video



Difficulties

Combining hardware platforms, the hardware development cycle is long, and different modules must be adapted to the system.

- A variety of hardware module tool combinations.
- System environment and hardware performance.
- Network environment configuration, network delay optimization.

Usage Scenario

Vehicles Device

vehicle BLE key，vehicle BLE equipment, and security threat test case analysis.

IoT devices

For security practitioners, IoT vendors plan BLE test cases and penetration tests.

To Do

Integration

Integrate the hardware and integrate multiple USB dongles into the Main-board.

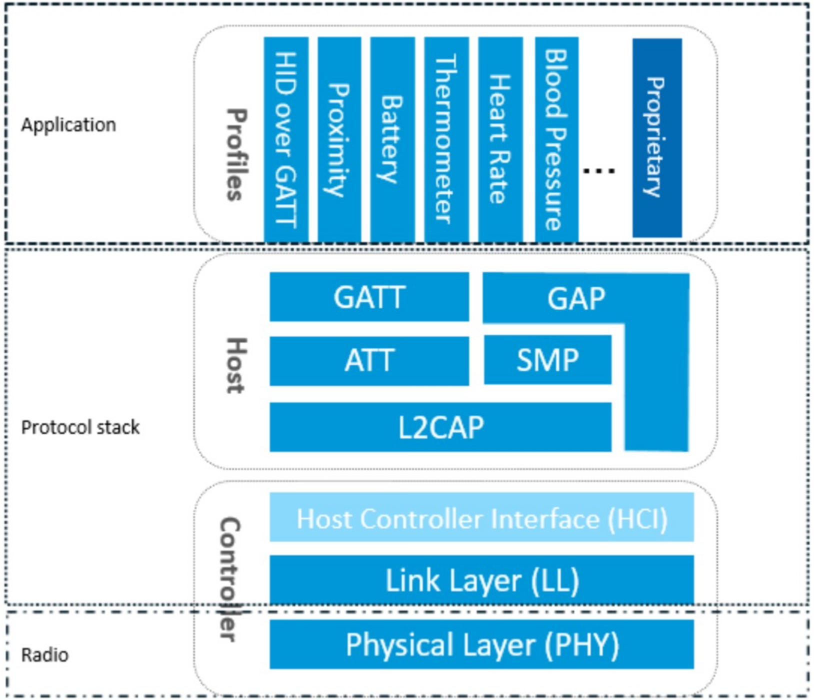
Extension

BLE historical CVEs detection, link layer fuzz, GATT layer fuzz .

Open Source

Open source software and hardware solutions to Github, everyone contributes this project.

BLE Security Defense



LL be encrypted

In the encrypted link, LL_CHANNEL_MAP_IND and other commands are used to update link parameters (such as ChM, WinSize, etc.), and the sniffer for the link of the relay cannot follow.

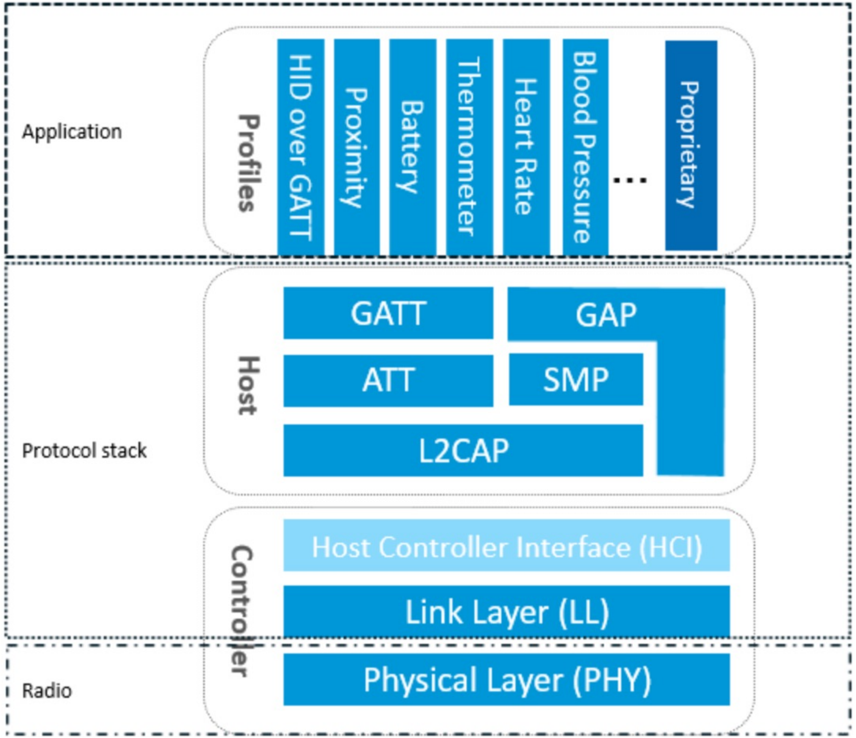
MD configure

more data is similar to TCP fragmentation, it tells you that there are still data packets to be transmitted, please continue to open the window. A connection event contains multiple data packet interactions.

multi-connection auth

In the field of car keys, multiple Bluetooth modules are used to assist in marking connected devices.

BLE Security Defense



SMP

SMP handles security between BLE devices, including authentication, encryption, and key management. The main goal of SMP is to secure communications between Bluetooth devices and provide protection against threats such as eavesdropping, tampering and counterfeiting.

Service encryption

Important service data is encrypted before encrypted transmission via Bluetooth. Strictly control the read and write permissions of UUID

THANK YOU!