

环境

1. 主机: win10
2. 手机: Pixel 4 , Android 10
3. APP版本: V5.54.0

工具

IDA、JADX、Frida、Charles

登录逆向分析

RegisterNatives

```
bool __fastcall register_kugou_player_mediautilsextra(_JNIEnv *a1)
{
    void *v2; // r6
    _BOOL4 v3; // r6
    void *v4; // r5
    int v5; // r10
    int v6; // r9
    int v7; // r8
    int v8; // r7
    jobject v9; // r0

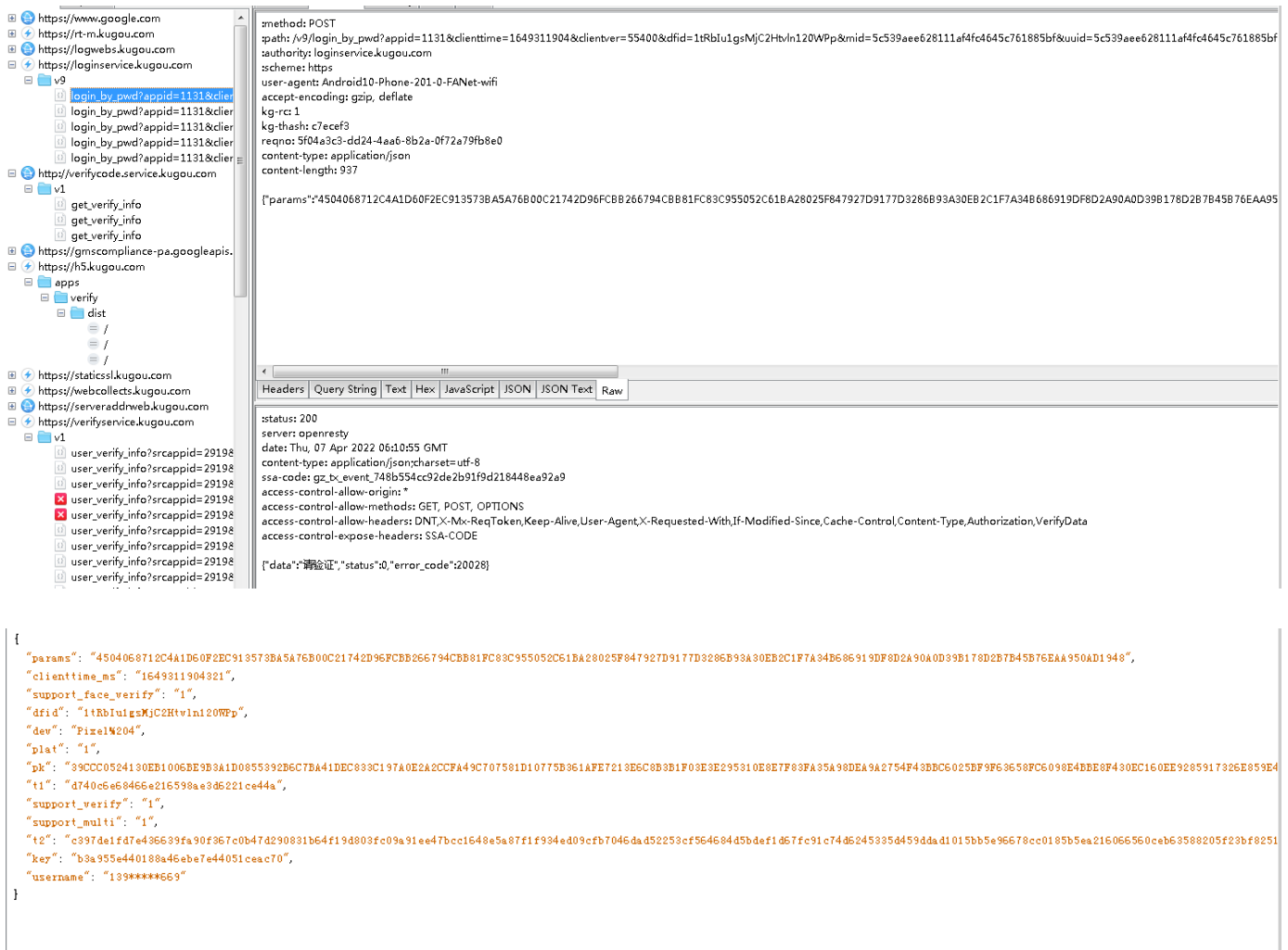
    v2 = (void *)_JNIEnv::FindClass(a1, "com/kugou/common/player/kugouplayer/j");
    if ( a1->functions->RegisterNatives((JNIEnv *)a1, v2, (const JNINativeMethod *)off_FA004, 21) < 0 )
        v3 = 0;
    else
        v3 = v2 != 0;
    v4 = (void *)_JNIEnv::FindClass(a1, "com/kugou/common/player/kugouplayer/j$A");
    if ( !_JNIEnv::ExceptionCheck(a1) )
        goto LABEL_5;
    if ( !v4 )
        return v3;
    v5 = _JNIEnv::GetMethodID(a1, v4, "<init>", "()V");
    if ( !_JNIEnv::ExceptionCheck(a1) )
    {
        || (v6 = _JNIEnv::GetFieldID(a1, v4, aEia, &aEia[1]), _JNIEnv::ExceptionCheck(a1))
        || (v7 = _JNIEnv::GetFieldID(a1, v4, &aEia[2], "[B]", _JNIEnv::ExceptionCheck(a1))
        || (v8 = _JNIEnv::GetFieldID(a1, v4, "r", "[B]", _JNIEnv::ExceptionCheck(a1)) )
    }
LABEL_5:
    _JNIEnv::ExceptionClear(a1);
}
else if ( v6 && v7 && v8 )
{
    v9 = a1->functions->NewGlobalRef(a1, v4);
    dword_FA744 = v5;
    dword_FA748 = v6;
    dword_FA74C = v7;
    dword_FA750 = v8;
    byte_FA754 = 1;
    dword_FA740 = (int)v9;
}
return v3;
}
```

动态注册的函数
偏移

.data:000FA008	7F 85 0D 00	DCD aLjavaLangObjec_1 ; "(Ljava/lang/Object;)I"
.data:000FA00C	29 77 03 00	DCD _ZN2cc1kEP7_JNIEnvP8_jobjectS3_+1 ; cc::k(_JNIEn
.data:000FA010	95 85 0D 00	DCD aC_1 ; "_c"
.data:000FA014	7F 85 0D 00	DCD aLjavaLangObjec_1 ; "(Ljava/lang/Object;)I"
.data:000FA018	5D 55 03 00	DCD _ZN2cc1cEP7_JNIEnvP8_jobjectS3_+1 ; cc::c(_JNIEn
.data:000FA01C	98 85 0D 00	DCD aD_1 ; "_d"
.data:000FA020	9B 85 0D 00	DCD aLjavaLangObjec_2 ; "(Ljava/lang/Object;)Ljava
.data:000FA024	A9 45 03 00	DCD _ZN2cc1dEP7_JNIEnvP8_jobjectS3_+1 ; cc::d(_JNIEn
.data:000FA028	C2 85 0D 00	DCD aE ; "_e"
.data:000FA02C	9B 85 0D 00	DCD aLjavaLangObjec_2 ; "(Ljava/lang/Object;)Ljava
.data:000FA030	C9 72 03 00	DCD _ZN2cc1eEP7_JNIEnvP8_jobjectS3_+1 ; cc::e(_JNIEn
.data:000FA034	C5 85 0D 00	DCD aF_0 ; "_f"
.data:000FA038	7F 85 0D 00	DCD aLjavaLangObjec_1 ; "(Ljava/lang/Object;)I"
.data:000FA03C	9D 42 03 00	DCD _ZN2cc1fEP7_JNIEnvP8_jobjectS3_+1 ; cc::f(_JNIEn
.data:000FA040	C8 85 0D 00	DCD aI_0 ; "_i"
.data:000FA044	9B 85 0D 00	DCD aLjavaLangObjec_2 ; "(Ljava/lang/Object;)Ljava
.data:000FA048	5D 60 03 00	DCD _ZN2cc1iEP7_JNIEnvP8_jobjectS3_+1 ; cc::i(_JNIEn
.data:000FA04C	CB 85 0D 00	DCD aI1 ; "_i1"
.data:000FA050	9B 85 0D 00	DCD aLjavaLangObjec_2 ; "(Ljava/lang/Object;)Ljava
.data:000FA054	F1 60 03 00	DCD _ZN2cc2i1EP7_JNIEnvP8_jobjectS3_+1 ; cc::i1(_JNI
.data:000FA058	CF 85 0D 00	DCD aH ; "_h"
.data:000FA05C	9B 85 0D 00	DCD aLjavaLangObjec_2 ; "(Ljava/lang/Object;)Ljava
.data:000FA060	FD 45 03 00	DCD _ZN2cc1hEP7_JNIEnvP8_jobjectS3_+1 ; cc::h(_JNIEn
.data:000FA064	D2 85 0D 00	DCD aL ; "_l"
.data:000FA068	D5 85 0D 00	DCD aLjavaLangObjec_3 ; "(Ljava/lang/Object;)V"
.data:000FA06C	C1 55 03 00	DCD _ZN2cc5func4EP7_JNIEnvP8_jobjectS3_+1 ; cc::func
.data:000FA070	EB 85 0D 00	DCD aM ; "_m"
.data:000FA074	9B 85 0D 00	DCD aLjavaLangObjec_2 ; "(Ljava/lang/Object;)Ljava
.data:000FA078	71 5C 03 00	DCD _ZN2cc5func5EP7_JNIEnvP8_jobjectS3_+1 ; cc::func
.data:000FA07C	EE 85 0D 00	DCD aN_0 ; "_n"
.data:000FA080	D5 85 0D 00	DCD aLjavaLangObjec_3 ; "(Ljava/lang/Object;)V"
.data:000FA084	39 56 03 00	DCD _ZN2cc5func6EP7_JNIEnvP8_jobjectS3_+1 ; cc::func
.data:000FA088	F1 85 0D 00	DCD aO ; "_o"
.data:000FA08C	F4 85 0D 00	DCD aLjavaLangObjec_4 ; "(Ljava/lang/Object;)[B"
.data:000FA090	41 51 03 00	DCD _ZN2cc5func7EP7_JNIEnvP8_jobjectS3_+1 ; cc::func
.data:000FA094	0B 86 0D 00	DCD aP ; "_p"
.data:000FA098	0E 86 0D 00	DCD aLjavaLangObjec_5 ; "(Ljava/lang/Object;)Z"
.data:000FA09C	C9 4A 03 00	DCD _ZN2cc5func8EP7_JNIEnvP8_jobjectS3_+1 ; cc::func
.data:000FA0A0	24 86 0D 00	DCD aQ ; "_q"
.data:000FA0A4	9B 85 0D 00	DCD aLjavaLangObjec_2 ; "(Ljava/lang/Object;)Ljava
.data:000FA0A8	4D 4D 03 00	DCD _ZN2cc5func9EP7_JNIEnvP8_jobjectS3_+1 ; cc::func
.data:000FA0AC	27 86 0D 00	DCD aR_1 ; "_r"

Request Params

首先抓包看一下登录包:



verifydata

```
{ "params":  
  "5F1D4B282F5E89548F98FC30853F4F9289D874AA82EDB0ED82330B07CF5D208ECA  
  AF446BAEA4D4848BB7515573123514EFDEA8E6C1000FEF8F30A9901382B7119822C  
  74B9C91394F9DDFF8B239E241F9", "clienttime_ms": "1647312358378",  
  "support_face_verify": "1", "dfid": "-", "dev": "Pixel%204", "plat": "1", "pk":  
  "7C9E722B7BCCE5B56BAB9D42902B2A4F238BAA40EACCCEECACE72C3EE46A85B1B  
  833F7800E6D2B968EE809303485376180BD8494EBA84F006DBD050E1F876B3713DE  
  64AFADDA4F51FCC522C6DAC0AFFC8A04AABAF47F95FE6D3F2FB6726A16FC4664AB  
  66CE065E3FCFD7D0FACD3DCA05AA96D74DD96EC74DBF54CA72D088E42C", "t1":  
  "bfaf46951c6a5c58d4bf618c517354de", "support_verify": "1", "support_multi": "1",  
  "t2":  
  "c397de1fd7e436639fa90f367c0b47d290831b64f19d803fc09a91ee47bcc1648e5a87f  
  1f934ed09cfb7046dad52253cf564684d5bdef1d67fc91c74d6245335237a78074f1576  
  41a1ddfb5ca4a66c5d72855feae18282b981d896c6f170cb54ca7c21aad3811d69c80a7  
  3ad3482c340", "key": "8d73d8b5da06bad3b466d864812d488b", "username":  
  "130*****696" }
```

params参数解析

通过JADX搜索“params”字符串定位到位置：

```
this.j.put("params", a.c(jsonObject.toString(), this.g));
```

进入c方法内可以看到：

```
public static String c(String str, String str2) throws Exception {  
    ...  
    return a(str, "utf-8", ao.a(str2).substring(0, 32), ao.a(str2).substring(16,  
32));  
}
```

通过objection hook该方法内的a方法，得到如下结果：

(agent) [302556] Called com.kugou.fanxing.allinone.common.utils.a.a(java.lang.String, java.lang.String, java.lang.String, java.lang.String)

(agent) [302556] Arguments

com.kugou.fanxing.allinone.common.utils.a.a({"username":"13062581696","clienttime_ms":"1647313862199","pwd":"557998555"}, utf-8, Odd6da40aea47d05b5f6612596fd2e7f, b5f6612596fd2e7f)

(agent) [793424] Backtrace: com.kugou.fanxing.allinone.common.utils.a.a(Native Method)
com.kugou.fanxing.allinone.common.utils.a.a(SourceFile:117)
com.kugou.fanxing.allinone.common.utils.a.a(Native Method)
com.kugou.fanxing.allinone.common.utils.a.c(SourceFile:106)
com.kugou.fanxing.core.protocol.g.e.c(SourceFile:85)
com.kugou.fanxing.core.protocol.g.e.c(Native Method)
com.kugou.fanxing.core.protocol.g.a.d(SourceFile:76)
com.kugou.fanxing.core.modul.user.login.c.a(SourceFile:50)
com.kugou.fanxing.core.modul.user.login.f.a(SourceFile:228)
com.kugou.fanxing.core.modul.user.ui.a.a(SourceFile:577)
com.kugou.fanxing.core.modul.user.ui.a.j(SourceFile:554)
com.kugou.fanxing.core.modul.user.ui.a.i(SourceFile:520)
com.kugou.fanxing.core.modul.user.ui.a.onClick(SourceFile:417)
android.view.View.performClick(View.java:7259)
android.view.View.performClickInternal(View.java:7236) android.view.View.access
3600(View.java : 801)android.view.ViewPerformClick.run(View.java:27892)
android.os.Handler.handleCallback(Handler.java:883)

```
android.os.Handler.dispatchMessage(Handler.java:100)
android.os.Looper.loop(Looper.java:214)
android.app.ActivityThread.main(ActivityThread.java:7356)
java.lang.reflect.Method.invoke(Native Method)
com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:492)
com.android.internal.os.ZygoteInit.main(ZygoteInit.java:930)
```

这样我们可以得到a.c(jsonObject.toString(), this.g)中两个参数的来源，其中第一个参数为JSON结构：

{"username":"13062581696","clienttime_ms":"1647313862199","pwd":"557998555"}，第二个参数为AES的KEY，最终将KEY拆分为(0,32)和(16,32)具体生成方式如下：

```
this.g =
com.kugou.fanxing.allinone.common.utils.a.a(com.kugou.fanxing.allinone.common.constant.c.h
? 128 : 64);
```

a方法如下：

```
public static String a(int i) {
    try {
        KeyGenerator instance = KeyGenerator.getInstance("AES");
        instance.init(i);
        return a(instance.generateKey().getEncoded());
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}

public static String a(byte[] bArr) {
    StringBuffer stringBuffer = new StringBuffer();
    for (byte b : bArr) {
        String hexString = Integer.toHexString(b & 255);
        if (hexString.length() == 1) {
            hexString = '0' + hexString;
        }
        stringBuffer.append(hexString.toUpperCase());
    }
    return stringBuffer.toString();
}

public static String c(String str, String str2) throws Exception {
    return a(str, "utf-8", ao.a(str2).substring(0, 32),
ao.a(str2).substring(16, 32));
}
```

```

public static String a(String str, String str2, String str3, String str4) throws
Exception {
    SecretKeySpec secretKeySpec = new SecretKeySpec(str3.getBytes(str2),
"AES");
    Cipher instance = Cipher.getInstance("AES/CBC/PKCS5Padding");
    instance.init(1, secretKeySpec, new IvParameterSpec(str4.getBytes()));
    return a(instance.doFinal(str.getBytes(str2)));
}

```

dfid参数解析

通过搜索字符串即可定位到如下函数：

```

public String e() {
    String a2 = com.kugou.fanxing.core.common.fingerprint.a.a();
    return TextUtils.isEmpty(a2) ? "-" : a2;
}

```

a2 -> share_data -> device_fingerprint

通过SharedPreferences读取了文件名为share_data里的device_fingerprint字段

pk参数解析

```

hashMap.put("pk",
com.kugou.fanxing.core.protocol.g.a.a(String.valueOf(currentTimeMillis), a2));

```

```

public static String a(String str, String str2) throws Exception {
    HashMap hashMap = new HashMap();
    hashMap.put("clienttime_ms", str);
    hashMap.put(ao.M, str2);
    return com.kugou.common.player.kugouplayer.a.d(hashMap);
}

```

str为时间戳，str2为AES随机密钥
组成形式如下：

```

{"clienttime_ms":"1647829236357","key":"E96E510C296711ECA6C85CAF6152F4D"}

```

最终调用native层的cc::加密方法

```

int __fastcall cc::i(_JNIEnv *a1, int a2, int a3)
{
    char *v4; // r1
    int v5; // r5
    int v7[3]; // [sp+0h] [bp-40h] BYREF
    char v8; // [sp+Ch] [bp-34h] BYREF
    char v9; // [sp+0h] [bp-33h] BYREF
    char *v10; // [sp+14h] [bp-2Ch]
    char v11[12]; // [sp+18h] [bp-28h] BYREF

    v7[0] = (int)v7;
    v7[1] = (int)v7;
    v7[2] = 0;
    cc::h2((int)a1, a3, (int)v7); // 取出hashmap里的值, Pair的first和second
    f10((int)v11, (int)v7); // 讲取出来的值进行格式化, 格式为{"clienttime_ms":"1647829236357", "key":"E96E510C296711ECA6C85CAF6152F4D"}
    f1((int)&v8, (int)v11); // 初始化RSA公钥并使用RSA_NO_PADDING方式加密, 加密结果唯一
    std::string::~string((int)v11);
    if ( (v8 & 1) != 0 )
        v4 = v10;
    else
        v4 = &v9;
    v5 = _JNIEnv::NewStringUTF(a1, v4);
    std::string::~string((int)&v8);
    std::_list_imp<std::pair<std::string, JsonObjectValue>>::clear(v7);
    return v5;
}

```

PUBKEY:

```

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQD2DT4odzkdDd7hMlZ7djdZQH1
2j38nKxriINW1MGjMry3tXheya113xwmbBOwN0GA4zTwKFauFJRzcsD0nDFq1eaatcFK
eDF25R4dnQRX+4BdTwFVS8lIb8nJMIuSBwK+i4Z3VF+gfZ0AqQOXda6lJ4jPBt9Ep7VX
EAHXUDn9JM8wIDAQAB

```

Python版RSA_NOPADDING加密方法实现:

```

import rsa
import base64
from Crypto.PublicKey import RSA

```

```

def zfillStrToBin(s):
    b = bytes(s.encode())
    for i in range(128 - len(b)):
        b += b'\0'
    print(len(b))
    return b

```

```

class RsaNopadding:

```

```

    def __init__(self, key):
        self.pubkey = RSA.importKey(base64.b64decode(key))

    def encrypt(self, message):
        kLen = rsa.common.byte_size(self.pubkey.n)
        msg = zfillStrToBin(message)
        _b = rsa.transform.bytes2int(msg)
        _i = rsa.core.encrypt_int(_b, self.pubkey.e, self.pubkey.n)
        result = rsa.transform.int2bytes(_i, kLen)
        return result.hex().upper()

```

```

message = '{"clienttime_ms":"1647829236357", "key":"E96E510C296711ECA6C85CAF6152F4D"}'

```

```
msg = RsaNopadding(

"MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQD2DT4odzkdDd7hMlZ7djdZQH12j38nKxriINW1MGjMry3tXheya1

print(msg.encrypt(message))
```

t1参数解析

```
String t1 = com.kugou.common.player.kugouplayer.a.b(null);
```

最终调用native层的cc::d方法

```
int __fastcall cc::d(_JNIEnv *a1, int a2, char *a3, int a4)
{
    char *v5; // r1
    int v6; // r6
    _DWORD v8[2]; // [sp+0h] [bp-20h] BYREF
    char *v9; // [sp+8h] [bp-18h]
    int v10; // [sp+Ch] [bp-14h]

    v8[0] = a1;
    v8[1] = a2;
    v9 = a3;
    v10 = a4;
    f4((int)v8); // 加密函数
    if ( (v8[0] & 1) != 0 )
        v5 = v9;
    else
        v5 = (char *)v8 + 1;
    v6 = _JNIEnv::NewStringUTF(a1, v5);
    std::string::~string((int)v8);
    return v6;
}
```

这里只分析f4函数内关键步骤部分：

```
int __fastcall f4(int a1)
{
    LABEL_44: // 走这
        v76[0] = (int)v76;
        v76[1] = (int)v76;
        v76[2] = 0;
        v77 = 0;
        v78 = 0;
        v79 = 0;
        std::string::__init((int)&v77, (int)&aEia[2], 1);
        v34 = v77;
        v35 = v78;
        v77 = 0;
        v78 = 0;
        v103 = v34;
```



```

v104 = v35;
v105 = v79;
v79 = 0;
std::string::basic_string((int)v106, (int)&v66);
std::list<std::pair<std::string, std::string>>::push_back((int)v76, (int)&v103);
std::string::~string((int)v106);
std::string::~string((int)&v103);
std::string::~string((int)&v77);
v80 = 0;
v81 = 0;
v82 = 0;
std::string::__init((int)&v80, (int)"b", 1);
f5(&v83); // daytime
v36 = v80;
v37 = v81;
v38 = v82;
v80 = 0;
v81 = 0;
v82 = 0;
v103 = v36;
v104 = v37;
v105 = v38;
v39 = v83;
v40 = v84;
v41 = v85;
v83 = 0;
v84 = 0;
v85 = 0;
v106[0] = v39;
v106[1] = v40;
v106[2] = v41;
std::list<std::pair<std::string, std::string>>::push_back((int)v76, (int)&v103);
std::string::~string((int)v106);
std::string::~string((int)&v103);
std::string::~string((int)&v83);
std::string::~string((int)&v80);
f9((int)v86, (int)v76);
v87 = 0;
v42 = v107;
v88 = 0;
v89 = 0;
v95[0] = 0;
v95[1] = 0;
v95[2] = 0;
v43 = aBdeaed243193ce; // v42=bdeaed243193ce1i#~DC<M[g
do
{
    v44 = *(_DWORD *)v43;
    v43 += 8;
    v45 = *((_DWORD *)v43 - 1);
    *(_DWORD *)v42 = v44; // 整个循环将bdeaed243193ce1i#~DC<M[g扩
    *((_DWORD *)v42 + 1) = v45;
}
展到bdeaed243193ce1i#~DC<M[g..=2..~4

```

```

    v42 += 8;
}
while ( v43 != (const char *)&unk_D90E4 );
v103 = 0;
v104 = 0;
v105 = 0;
std::string::__init__((int)&v103, (int)v107, 32);//
v105=v109=bdeaed243193ce1i#~DC<M[g..=2..~4
    ..... // 62 64 65 61 65 64 32 34 33 31 39 33
63 65 31 69
    ..... // 23 7E 44 43 3C 4D 5B 67 7F 0E 3D 32
01 2E 7E 34
v46 = (unsigned __int8)v103;
if ( (v103 & 1) == 0 )
    v46 = (int)(unsigned __int8)v103 >> 1;
if ( (v103 & 1) != 0 )
    v46 = v104;
v47 = v46 - 2;
do
{
    if ( v47 < 0 )
        break;
    v48 = (v103 & 1) != 0 ? v105 : (int *)((char *)&v103 + 1);
    v49 = (char *)v48 + v47; // 转换v105为
bdeaed243193ce11ac913bbd48d340a4
    v50 = (v103 & 1) != 0 ? v105 : (int *)((char *)&v103 + 1);
    v51 = *((_BYTE *)v50 + v47);
    v52 = (char *)&unk_D90E4 + v47 - v46;
    --v47;
    *v49 = v51 ^ v52[17];
}
while ( v47 != v46 - 18 );
std::string::basic_string__((int)&v100, (int)&v103);
std::string::~~string__((int)&v103);
v53 = v107;
v54 = aAc913bbd48d340; // v54 =
ac913bbd48d340a41234567890qwertyuiopasdfghjklzxcvbnm.
do
{
    v55 = *((_DWORD *)v54);
    v54 += 8;
    v56 = *((_DWORD *)v54 - 1);
    *((_DWORD *)v53 = v55; // 整个循环将
ac913bbd48d340a41234567890qwertyuiopasdfghjklzxcvbnm.转换为ac913bbd48d340a4
    *((_DWORD *)v53 + 1) = v56;
    v53 += 8;
}
while ( v54 != &aAc913bbd48d340[16] );
v103 = 0;
v104 = 0;
v105 = 0;
std::string::__init__((int)&v103, (int)v107, 16);// v103=v107=ac913bbd48d340a4
std::string::basic_string__((int)v99, (int)&v103);

```

```

std::string::~string((int)&v103);
sub_394FC(1, v86, v95, (unsigned __int8 *)&v100, v99); // AES加密
std::string::~string((int)v99);
std::string::~string((int)&v100);
f11((int)&v96, (int)v95); // HEX格式化
v58 = v87 & 1;
if ( (v87 & 1) != 0 )
{
    v57 = v89;
    v58 = 0;
}
else
{
    BYTE1(v87) = v87 & 1;
}
if ( (v87 & 1) != 0 )
{
    *v57 = v58;
    v88 = v58;
}
else
{
    LOBYTE(v87) = v58;
}
std::string::reserve((int)&v87, 0);
v59 = v96;
v60 = v97;
v61 = v98;
v96 = 0;
v97 = 0;
v98 = 0;
v87 = v59;
v88 = v60;
v89 = (_BYTE *)v61;
std::string::~string((int)&v96);
std::string::~string((int)v95);
if ( (*(_BYTE *)a1 & 1) != 0 )
{
    **(_BYTE **)(a1 + 8) = 0;
    *(_DWORD *)(a1 + 4) = 0;
}
else
{
    *(_BYTE *)(a1 + 1) = 0;
    *(_BYTE *)a1 = 0;
}
std::string::reserve(a1, 0);
v62 = v87;
v63 = v88;
v64 = (int)v89;
v87 = 0;
v88 = 0;
v89 = 0;

```

```

*(_DWORD *)a1 = v62;
*(_DWORD *)(a1 + 4) = v63;
*(_DWORD *)(a1 + 8) = v64;
std::string::~~string((__int64)v87);
std::string::~~string((__int64)v86);
std::__list_imp<std::pair<std::string, std::string>>::clear(v76);
std::string::~~string((__int64)v66);
return a1;
}

```

cc::d方法里为AES_256_CBC_ENCRYPT

encData: |1649905000102

key:bdeaed243193ce11ac913bbd48d340a4

iv:ac913bbd48d340a4

t2参数解析

```
String t2 = com.kugou.common.player.kugouplayer.a.c(null);
```

最终调用native层的cc::e方法

```

int __fastcall cc::e(_JNIEnv *a1)
{
    sub_346F4(&v33);
    v34[2] = 0;
    v34[0] = (int)v34;
    v34[1] = (int)v34;
    v35 = 0;
    v36 = 0;
    v37 = 0;
    std::string::__init((__int64)v35, (int)&aEia[2], 1);
    cc::h4((__int64)v38, a1); // 获取ANDROID_ID
    v2 = v35;
    v3 = v36;
    v35 = 0;
    v36 = 0;
    v66 = v2;
    v67 = v3;
    v68 = v37;
    v4 = v38;
    v5 = v39;
    v37 = 0;
    v38 = 0;
}

```

```

v39 = 0;
v69 = v4;
v70 = v5;
v71 = v40;
v40 = 0;
std::list<std::pair<std::string, std::string>>::push_back(v34, &v66);
std::pair<std::string, std::string>::~~pair(&v66);
std::string::~~string((int)&v38);
std::string::~~string((int)&v35);
v41 = 0;
v42 = 0;
v43 = 0;
std::string::__init((int)&v41, (int)"b", 1);
cc::h5((int)&v44, a1); // 获取DeviceId
v6 = v41;
v7 = v42;
v8 = v43;
v41 = 0;
v42 = 0;
v43 = 0;
v66 = v6;
v67 = v7;
v68 = v8;
v9 = v44;
v10 = v45;
v11 = v46;
v44 = 0;
v45 = 0;
v46 = 0;
v69 = v9;
v70 = v10;
v71 = v11;
std::list<std::pair<std::string, std::string>>::push_back(v34, &v66);
std::pair<std::string, std::string>::~~pair(&v66);
std::string::~~string((int)&v44);
std::string::~~string((int)&v41);
v47 = 0;
v48 = 0;
v49 = 0;
std::string::__init((int)&v47, (int)"c", 1);
cc::h6((cc *)&v50, a1); // 获取network HardwareAddress
v12 = v47;
v13 = v48;
v14 = v49;
v47 = 0;
v48 = 0;
v49 = 0;
v66 = v12;
v67 = v13;
v68 = v14;
v15 = v50;
v16 = v51;
v17 = v52;

```

```

v50 = 0;
v51 = 0;
v52 = 0;
v69 = v15;
v70 = v16;
v71 = v17;
std::list<std::pair<std::string, std::string>>::push_back(v34, &v66);
std::pair<std::string, std::string>::~~pair(&v66);
std::string::~~string((int)&v50);
std::string::~~string((int)&v47);
v53 = 0;
v54 = 0;
v55 = 0;
std::string::__init((int)&v53, (int)"d", 1);
cc::h8((cc *)&v56, a1); // 获取手机的型号设备名称
v18 = v53;
v19 = v54;
v20 = v55;
v53 = 0;
v54 = 0;
v55 = 0;
v66 = v18;
v67 = v19;
v68 = v20;
v21 = v56;
v22 = v57;
v23 = v58;
v56 = 0;
v57 = 0;
v58 = 0;
v69 = v21;
v70 = v22;
v71 = v23;
std::list<std::pair<std::string, std::string>>::push_back(v34, &v66);
std::pair<std::string, std::string>::~~pair(&v66);
std::string::~~string((int)&v56);
std::string::~~string((int)&v53);
v59 = 0;
v60 = 0;
v61 = 0;
std::string::__init((int)&v59, (int)aEia, 1);
f5(); // daytime
v24 = v59;
v25 = v60;
v26 = v61;
v59 = 0;
v60 = 0;
v61 = 0;
v66 = v24;
v67 = v25;
v68 = v26;
v27 = v62;
v28 = v63;

```

```

v29 = v64;
v62 = 0;
v63 = 0;
v64 = 0;
v69 = v27;
v70 = v28;
v71 = v29;
std::list<std::pair<std::string, std::string>>::push_back(v34, &v66);
std::pair<std::string, std::string>::~~pair(&v66);
std::string::~~string((int)&v62);
std::string::~~string((int)&v59);
f9(v65, v34); // 合并string
f6(&v66, v65); // aes
if ( (v66 & 1) != 0 )
    v30 = v68;
else
    v30 = (char *)&v66 + 1;
v31 = _JNIEnv::NewStringUTF(a1, v30);
std::string::~~string((int)&v66);
std::string::~~string((int)v65);
std::__list_imp<std::pair<std::string, std::string>>::clear(v34);
cc::sp<cc::RefJObject>::~~sp(&v33);
return v31;
}

```

f6函数内部逻辑与t1的f4函数一致

cc::e方法里为AES_256_CBC_ENCRYPT

encData: ||9eea2d301e53|Pixel 4|1649905000118

cc::d方法里为AES_256_CBC_ENCRYPT

key:dc8e123f07636a41361b62235fc313ac

iv:361b62235fc313ac

key参数解析

com.kugou.fanxing.core.protocol.g.e.c

```

map.put(ao.M, com.kugou.fanxing.allinone.common.utils.ao.a("" + this.b + this.e +
    getVersion() + String.valueOf(this.h).toLowerCase()));

```

ao.m = key

com.kugou.fanxing.allinone.common.d.a

```
public static int g() { return 1131; }
```

```
this.b = AppId
```

```
com.kugou.fanxing.allinone.common.d.a
```

```
public static String h() { return "4lu0l3cujt2KWljcM374F8oX5N2lGY59"; }
```

```
this.e = AppKey="4lu0l3cujt2KWljcM374F8oX5N2lGY59"
```

```
getVersion = APPVersion
```

```
public int getVersion() { return y.s(); }
```

```
55400
```

```
this.h = clienttime_ms
```

```
key = MD5("" + AppId + AppKey + APPVersion + String.valueOf(this.h).toLowerCase())
```

signature参数解析

```
com.kugou.fanxing.allinone.watch.common.protocol.user.a
```

```
public String a(String str, int i, String str2, String str3) {  
    StringBuilder sb = new StringBuilder(str);  
    StringBuilder sb2 = new StringBuilder();  
    sb.append("?");  
    sb.append("appid=");  
    sb.append(this.b);  
    sb2.append("appid=");  
    sb2.append(this.b);  
    sb.append("&clienttime=");  
    sb.append(i);  
    sb2.append("clienttime=");  
    sb2.append(i);  
    int version = getVersion();  
    sb.append("&clientver=");  
    sb.append(version);  
    sb2.append("clientver=");  
    sb2.append(version);  
    sb.append("&dfid=");  
    sb.append(str2);  
    sb2.append("dfid=");  
    sb2.append(str2);  
    String fxDeviceId = getFxDeviceId();  
    sb.append("&mid=");  
    sb.append(fxDeviceId);  
    sb2.append("mid=");  
    sb2.append(fxDeviceId);  
}
```



```

        String androidId = getAndroidId();
        sb.append("&uuid=");
        sb.append(androidId);
        sb2.append("uuid=");
        sb2.append(androidId);
        String a2 = com.kugou.fanxing.allinone.common.utils.ao.a(this.e +
sb2.toString() + str3 + this.e);
        sb.append("&signature=");
        sb.append(a2);
        return sb.toString();
    }

```

(agent) [013697] Arguments

com.kugou.fanxing.allinone.watch.common.protocol.user.a.a(https://loginservice.kugou.com/v9/login_by_pwd, 1650353888, 1tRblu1gsMjC2Htvln120WPp, {"params":"2CA44E5F5D8F189BC68662C0438844D7043484C37D09E7AD1A8A138D44CDCBB0465684EE3C6FD64263174944AD6F2027D0782D9BF64A904432F9FD764CF1C96FF5C90B16DB2932E15FDC56178940207F","clienttime_ms":"1650353888654","support_face_verify":"1","dfid":"1tRblu1gsMjC2Htvln120WPp","dev":"Pixel%204","plat":"1","pk":"5F03A05AFD4E10B7283A9F2634153FEFA0AF318DD781687BB907DA770614C021921AEB30AB7A808714C132ECA24285E43E6EA2134A91BB79893B10B4024040E32C8EA448463251173D9E412A5B640F240AA601690DDF9A180599F73AA5CF45F8570F6562ABEFA31196CBB0BB31BB6E781974C56A4640CB70D25955D04F3CC7EA","t1":"cdb7d88e278cab7c596af28cfcd873ba","support_verify":"1","support_multi":"1","t2":"c397de1fd7e436639fa90f367c0b47d290831b64f19d803fc09a91ee47bcc1648e5a87f1f934ed09cfb7046dad52253cf564684d5bdef1d67fc91c74d6245335d459ddad1015bb5e96678cc0185b5ea20cc2563d5fc8f040aa77b49c039e40c5e2ca000392eeebfce750e3612ff280f6","key":"ffa858faa437189eb024cdeeb37e372d","username":"130*****669"})

com.kugou.fanxing.allinone.common.utils.ao.a(

4lu0l3cujt2KWljcM374F8oX5N2lGY59

appid=1131clienttime=1650356391clientver=55400dfid=1tRblu1gsMjC2Htvln120WPp
Ppmid=5c539aee628111af4fc4645c761885bfuuid=5c539aee628111af4fc4645c761885bf

{"params":"3FF4D2AB2A549FDBD95EBA40F155635B1E9A5EB353EF147EC459EC47FE18EDE2ED465FAE1E974ABB8605499F1434228E6A30779E0E07DB12A79775A6073EE748709951692C97506CD6384513250870B3","clienttime_ms":"1650356391589","support_face_verify":"1","dfid":"1tRblu1gsMjC2Htvln120WPp","dev":"Pixel%204","plat":"1","pk":"62A6F1633AD5B9F85007AD597B90157C854A796FEA8D965E110C9D8FB152170E18CF6868F4497E1FAA62F13B257758DF803B2E4ABCF28980207534BE8B8728DFA5

```
0A4F00AD2626307B31C850E8ACED450393643C5CE8DD679E1F77186D3CED20DB72
E8003C4EF2048EC1942035159EDB2590095D09B6E7A706A72F5E96408212","t1":"512
640caae5a66bb98402eda86b5b026","support_verify":"1","support_multi":"1","t2":"c3
97de1fd7e436639fa90f367c0b47d290831b64f19d803fc09a91ee47bcc1648e5a87f1f9
34ed09cfb7046dad52253cf564684d5bdef1d67fc91c74d6245335d459ddad1015bb5e
96678cc0185b5ea2a4865a6be23bed4a31f7625c6927379348d014b77b354e00181c2e
0b1530397a","key":"7e8cfb92bb4be58f9a62e0af4c040783","username":"130*****669
"}

```

```
4lu0l3cujt2KWljcM374F8oX5N2lGY59)
```

signature=MD5(appkey+appid=1131clienttime=1650356391clientver=55400dfid=1tRblu
1gsMjC2Htvln120WPpmid=5c539aee628111af4fc4645c761885bfuuid=5c539aee628111af
4fc4645c761885bf+ post data + appkey)