# Replicant - Issue #1909

## Old French SFR SIM with PIN code setup card not detected

02/06/2019 07:54 PM - Denis 'GNUtoo' Carikli

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | 02/06/2019 |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Denis 'GNUtoo' Carikli | **% Done:** | 0% |
| **Category:** | SIM card not recognized | **Estimated time:** | 0.00 hour |
| **Target version:** | Replicant 6.0 | **Spent time:** | 26.35 hours |
| **Resolution:** | | **Device:** | |

**Description**

I've found a SIM card that isn't detected by libsamsung-ipc:

- On the Galaxy S 2 (i9100) with libsamsung-ril and libsamsung-ipc, the card "isn't detected"
- On an HTC Dream, it asks for a PIN.

As I don't know if the issue is the same than the other ones, and that I can easily reproduce it, I started looking at it from scratch and gathered logs on the Galaxy S2 both with this sim and another SIM that is working and that was setup to ask for a PIN code.

I've also removed the STK Android application that is part of stock Replicant installation to look if it would work without, but it didn't. All the test mentioned below were carried out with the STK Android application removed.

I still need to look deeper in the relevant source code, but I found that the main differences between the working and the non-working SIM card were:

- The non-working SIM card sends proactive Sim Toolkit commands, which were observed with simtrace and through libsamsung-ipc logging: libsamsung-ipc/samsung-ril receive a IPC_SAT_PROACTIVE_CMD command very early on and continue sending one or more of such commands.
- With the non-working SIM card, the Android framework didn't manage to retrieve the SIM ID, whereas that sim ID can clearly be found in the logs gathered with simtrace. After trying to get the SIM ID, with the working SIM card, we have "insertedSimCount = 1" in the logs whereas with the nonworking we have: "insertedSimCount = 0"

### History

#### #1 - 02/06/2019 08:00 PM - Denis 'GNUtoo' Carikli

*- File good-orange-sim-no-stk-app-with-simtrace_until-sim-pin.pcap.pcapng.gz added*

#### #2 - 02/06/2019 08:01 PM - Denis 'GNUtoo' Carikli

*- File good-orange-sim-no-stk-app-with-simtrace_until-sim-pin.log added*

#### #3 - 02/06/2019 08:01 PM - Denis 'GNUtoo' Carikli

*- File bad-sfr-sim-no-stk-app-with-simtrace.pcap.pcapng.gz added*

#### #4 - 02/06/2019 08:02 PM - Denis 'GNUtoo' Carikli

*- File bad-sfr-sim-no-stk-app-with-simtrace.log added*

#### #5 - 02/06/2019 08:08 PM - Denis 'GNUtoo' Carikli

Working Orange SIM:

```
02-06 13:47:56.903  2786  2786 D SubscriptionInfoUpdater: handleMessage : <EVENT_SIM_LOCKED_QUERY_ICCID_DONE>
SIM1
02-06 13:47:56.903  2786  2786 D SubscriptionInfoUpdater: sIccId[0] = 89330178269030170700
02-06 13:47:56.903  2786  2786 D SubscriptionInfoUpdater: All IccIds query complete
02-06 13:47:56.903  2786  2786 D SubscriptionInfoUpdater: updateSubscriptionInfoByIccId:+ Start
02-06 13:47:56.903  2786  2786 D SubscriptionInfoUpdater: insertedSimCount = 1
```

Whereas on the nonworking SFR SIM we have:

```
02-06 13:45:10.553  2773  2773 D SubscriptionInfoUpdater: handleMessage : <EVENT_SIM_LOCKED_QUERY_ICCID_DONE>
```

```
SIM1
02-06 13:45:10.553  2773  2773 D SubscriptionInfoUpdater: Query IccId fail: java.lang.NullPointerException: At
tempt to read from null array
02-06 13:45:10.553  2773  2773 D SubscriptionInfoUpdater: sIccId[0] =
02-06 13:45:10.553  2773  2773 D SubscriptionInfoUpdater: All IccIds query complete
02-06 13:45:10.553  2773  2773 D SubscriptionInfoUpdater: updateSubscriptionInfoByIccId:+ Start
02-06 13:45:10.553  2773  2773 D SubscriptionInfoUpdater: insertedSimCount = 0
```

And for the First STK message on the nonworking SFR SIM we have:

```
02-06 13:44:53.590  1941  2192 D use-Rlog/RLOG-RIL-IPC: xmm626_sec_modem_fmt_recv: Received FMT message
02-06 13:44:53.590  1941  2192 D use-Rlog/RLOG-RIL-IPC: xmm626_sec_modem_fmt_recv: Message: aseq=0xff, command
=IPC_SAT_PROACTIVE_CMD, type=IPC_TYPE_INDI, size=18
02-06 13:44:53.590  1941  2192 D use-Rlog/RLOG-RIL-IPC: ================================= IPC FMT data =======
==========================
02-06 13:44:53.590  1941  2192 D use-Rlog/RLOG-RIL-IPC: [0000]   10 00 D0 0E 81 03 01 05  00 82 02 81 82 99 03
 00   ........ ........
02-06 13:44:53.590  1941  2192 D use-Rlog/RLOG-RIL-IPC: [0010]   01 02
      ..
02-06 13:44:53.590  1941  2192 D use-Rlog/RLOG-RIL-IPC: ============================================================
==========================
02-06 13:44:53.590  1941  2192 D use-Rlog/RLOG-RIL-IPC: Unhandled IPC FMT message: IPC_SAT_PROACTIVE_CMD
```

#### #6 - 02/06/2019 08:11 PM - Denis 'GNUtoo' Carikli

I looked rapidely at the log of the Orange SIM, however I didn't find IccID (89330178269030170700) in the samsung-ipc messages (I tried to look for 33 and 9833 as this is how it appears in simtrace traces). I'll have to investigate more on this side. I also looked at the Java side of the RIL but didn't find enough information yet on how the SIM filesystem abstraction work.

#### #7 - 02/06/2019 09:02 PM - Denis 'GNUtoo' Carikli

Here we can find the ICCID (98:33:10:87:62:09:03:71:70:00):

```
02-06 13:47:56.902  1942  2203 D use-Rlog/RLOG-RIL-IPC: xmm626_sec_modem_fmt_recv: Received FMT message
02-06 13:47:56.902  1942  2203 D use-Rlog/RLOG-RIL-IPC: xmm626_sec_modem_fmt_recv: Message: aseq=0x26, command
=IPC_SEC_RSIM_ACCESS, type=IPC_TYPE_RESP, size=13
02-06 13:47:56.902  1942  2203 D use-Rlog/RLOG-RIL-IPC: ================================= IPC FMT data =======
==========================
02-06 13:47:56.902  1942  2203 D use-Rlog/RLOG-RIL-IPC: [0000]   90 00 0A 98 33 10 87 62  09 03 71 70 00
      ....3..b ..qp.
02-06 13:47:56.902  1942  2203 D use-Rlog/RLOG-RIL-IPC: ============================================================
==========================
```

#### #8 - 02/06/2019 09:05 PM - Denis 'GNUtoo' Carikli

Here's (still for the Orange SIM) the command to select the EF.ICCID file (0x2fe2):

```
02-06 13:47:56.895  1942  2207 D use-Rlog/RLOG-RIL-IPC: xmm626_sec_modem_fmt_send: Sent FMT message
02-06 13:47:56.895  1942  2207 D use-Rlog/RLOG-RIL-IPC: xmm626_sec_modem_fmt_send: Message: mseq=0x26, command
=IPC_SEC_RSIM_ACCESS, type=IPC_TYPE_GET, size=6
02-06 13:47:56.895  1942  2207 D use-Rlog/RLOG-RIL-IPC: ================================= IPC FMT data =======
==========================
02-06 13:47:56.895  1942  2207 D use-Rlog/RLOG-RIL-IPC: [0000]   B0 E2 2F 00 00 0A
      ../...
02-06 13:47:56.895  1942  2207 D use-Rlog/RLOG-RIL-IPC: ============================================================
==========================
```

#### #9 - 02/06/2019 09:21 PM - Denis 'GNUtoo' Carikli

I'm not totally sure about the last one. I'll look into the code to confirm. The ril_request_sim_io has some insights on how it works:

```
int ril_request_sim_io(void *data, size_t size, RIL_Token token)
{
[...]
    memset(&request_header, 0, sizeof(request_header));
    request_header.command = sim_io->command;
    request_header.file_id = sim_io->fileid;
    request_header.p1 = sim_io->p1;
    request_header.p2 = sim_io->p2;
    request_header.p3 = sim_io->p3;

    sim_io = NULL;
```

```
    request_size = ipc_sec_rsim_access_size_setup(&request_header, sim_io_data, sim_io_size);
    if (request_size == 0)
        goto error;

    request_data = ipc_sec_rsim_access_setup(&request_header, sim_io_data, sim_io_size);
    if (request_data == NULL)
        goto error;
[...]
    rc = ipc_fmt_send(ipc_fmt_request_seq(token), IPC_SEC_RSIM_ACCESS, IPC_TYPE_GET, request_data, request_siz
e);
    if (rc < 0)
        goto error;
}
```

#### #10 - 04/13/2019 07:49 PM - Denis 'GNUtoo' Carikli

After reading more about the protocol, I found out that the packets order didn't make any sense in the trace I got, because I didn't notice that wireshark didn't order them by time by default for some reason.

This is why I had a hard time understanding the link between them.

Once fixed, here's now what the part that the interesting part from the pcap trace:

```
No    Time        Protocol    Length    Info
13    25.070965442    GSM SIM        67    ISO/IEC 7816-4 SELECT File EF.ICCID : Response ready, Response leng
th is 27
14    25.071061474    GSM SIM        92    ISO/IEC 7816-4 GET RESPONSE
15    25.081092359    GSM SIM        75    ISO/IEC 7816-4 READ BINARY Offset=0
```

In the third packet/TPDU we can see in wireshark the following as APDU payload wihich is the SIM ID (ICCID): 98330163019072174071
which is transmited as 98 33 01 63 01 90 72 17 40 71 on the wire.

#### #11 - 04/13/2019 07:54 PM - Denis 'GNUtoo' Carikli

~~We also see more clearly that the SIM ID is queried right at the beginning, so it most probably cannot be related to Sim Toolkit(STK) as it appear way later in the conversation between the SIM and the modem.~~

The trace has precise timestamps in the Ethernet encapsulation. I'll use that to corelate it with the logcat -b radio.

Here's the trace from the very beginning:

```
No    Time        Protocol    Length    Info
8    23.809625449    GSM SIM    79    3 9e : 00f5
9    25.048405378    GSM SIM    67    ISO/IEC 7816-4 SELECT File MF
10    25.062601808    GSM SIM    67    ISO/IEC 7816-4 SELECT /EF.ELP : Response ready, Response length is 30
11    25.062696862    GSM SIM    95    ISO/IEC 7816-4 GET RESPONSE
12    25.062775503    GSM SIM    75    ISO/IEC 7816-4 READ BINARY Offset=0
13    25.070965442    GSM SIM    67    ISO/IEC 7816-4 SELECT File EF.ICCID : Response ready, Response length i
s 27
14    25.071061474    GSM SIM    92    ISO/IEC 7816-4 GET RESPONSE
15    25.081092359    GSM SIM    75    ISO/IEC 7816-4 READ BINARY Offset=0
[...]
```

#### #12 - 04/13/2019 07:57 PM - Denis 'GNUtoo' Carikli

*- Device deleted (Galaxy S 2 (I9100))*

The galaxy Nexus is also affected, and probably all other devices using libsamsung-ipc libsamsung-ril

#### #13 - 04/24/2019 03:39 AM - Denis 'GNUtoo' Carikli

In the last part of the trace it tries a second time to get the ICCID but fails:

```
No    Time        Protocol    Length    Info
732    42.471309922    GSM SIM        67    ISO/IEC 7816-4 SELECT /EF.ICCID
733    42.471375572    GSM SIM        92    ISO/IEC 7816-4 GET RESPONSE
```

Then nothing happens. To work a "READ BINARY Offset=0" would be required.

Comparing with other bug reports and traces:

- For the working Orange SIM, Reading a second time the ICCID works (it does a "READ BINARY Offset=0"). It is also the last SIM file read in the trace.
- The non-working trace in the bug #1868 exibit a similar behavior: the "READ BINARY Offset=0" is not present in the second read of the ICCID file. It's also the last file read in the trace.

#### #14 - 04/24/2019 03:53 AM - Denis 'GNUtoo' Carikli

The SIM tookkit messages names make me think that the card is probably waiting for an answer from the modem for one of the SIM initiated STK commands

#### #15 - 04/26/2019 03:04 PM - Denis 'GNUtoo' Carikli

*- File maguro-takju-imm76i.pcapng.gz added*

I wanted to use the stock LineageOS 13 images to do a comparison trace to save time.

However the images are not to be found even on archive.org.

So after trying to build one (which required an image for the proprietary components) I've settled on using the official googles images for maguro to do the diff. Here I used the boot, system and userdata from the takju-imm76i-factory-e8c33767.tar.xz.

For the Replicant I used a Replicant 6.0 0003 Release.

#### #16 - 04/26/2019 03:04 PM - Denis 'GNUtoo' Carikli

*- File maguro-takju-imm76i.log added*

#### #17 - 04/26/2019 03:05 PM - Denis 'GNUtoo' Carikli

*- File maguro-replicant.pcapng.gz added*

#### #18 - 04/26/2019 03:05 PM - Denis 'GNUtoo' Carikli

*- File maguro-replicant.log added*

#### #19 - 04/26/2019 03:22 PM - Denis 'GNUtoo' Carikli

How to diff the files easily:

```
$ tshark -r maguro-takju-imm76i.pcapng.gz | sed 's#.*GSM SIM ##' > maguro-takju-imm76i.pcapng.txt
$ tshark -r maguro-replicant.pcapng.gz | sed 's#.*GSM SIM ##' > maguro-replicant.pcapng.txt
$ meld maguro-takju-imm76i.pcapng.txt maguro-replicant.pcapng.txt
```

#### #20 - 04/26/2019 03:25 PM - Denis 'GNUtoo' Carikli

I've got some malformed packets in the stock capture, I'll re-do it. Normally the only diff that should show up would be that one which is at the very end as it was in previous capture (where I forgot to capture a logcat -b radio at the same time) :

For the stock image:

```
67 ISO/IEC 7816-4 SELECT File EF.ECC
98 ISO/IEC 7816-4 GET RESPONSE
85 ISO/IEC 7816-4 READ RECORD RecordNr=1
85 ISO/IEC 7816-4 READ RECORD RecordNr=2
```

For Replicant image:

```
67 ISO/IEC 7816-4 SELECT /EF.ICCID
92 ISO/IEC 7816-4 GET RESPONSE
```

#### #21 - 04/26/2019 03:34 PM - Denis 'GNUtoo' Carikli

*- File maguro-takju-imm76i.pcapng.gz added*

#### #22 - 04/26/2019 03:34 PM - Denis 'GNUtoo' Carikli

*- File maguro-takju-imm76i.log added*

#### #23 - 05/03/2019 03:06 PM - Denis 'GNUtoo' Carikli

I ended up using a grep -v to compare the two traces:

```
.PHONY: all

all: maguro-replicant.pcapng.txt maguro-takju-imm76i.pcapng.txt
    meld $^ &

%.txt: %.gz
    tshark -r $< -V \
           | grep -v "^ *Arrival Time: " \
           | grep -v "^ *Epoch Time: " \
           | grep -v "^ *Identification: " \
           | grep -v "^ *Header checksum: " \
           | grep -v "^User Datagram Protocol, Src Port: [0-9]\+, Dst Port: [0-9]\+" \
           | grep -v "^ *Source Port: [0-9]\+" \
           | grep -v "^ *\[Time delta from previous captured frame: .*\]" \
           | grep -v "^ *\[Time delta from previous displayed frame: .*\]" \
           | grep -v "^ *\[Time since reference or first frame: .*\]" \
           > $@
```

It has shown me that there is no difference at all in the traces before it asking again for the EF.ECCID when using Replicant (whereas with the stock OS, it asks for EF.ECC) as it was said before in the note 20 of this bugreport.

#### #24 - 07/31/2019 12:13 PM - Denis 'GNUtoo' Carikli

I tried with a stock Replicant 4.2, and the GUI was asking for a PIN.

=> The way to go is to try to bisect that.

#### #25 - 07/31/2019 12:24 PM - Denis 'GNUtoo' Carikli

*- File maguro-replicant-4.2.log added*

This is the log for Replicant 4.2

#### #26 - 09/23/2019 03:29 PM - Denis 'GNUtoo' Carikli

*- File radio_maguro_master_23_sept_2019.log added*

Weeks ago, I managed to build Replicant 4.2 from the replicant-4.2 branch of the manifest. As I forgot to check if the manifest had tags for the release too, I ended up having libsamsung-ipc and libsamsung-ril on the master revision. I didn't have any telephony working and I didn't send the logs here.

However when building again on the same branch, it now works, probably because some fixes were merged in libsamsung-ipc and libsamsung-ril in between.

Right now with the following revisions, under Replicant 4.2, it asks me for a PIN:
libsamsung-ril: 9a6fd5b Fix undefined references to MD5_{Init,Update,Final}
libsamsung-ipc: b1b1109 devices: fix strncmp use

This means that it will probably not be possible to bisect libsamsung-ril and libsamsung-ipc on Replicant 4.2 to find what broke such SIM cards.

However it's still possible to bisect other part of the code and also to do some bisecting and tests on Replicant 6.

Denis.

### Files

| | | | |
|---|---|---|---|
| good-orange-sim-no-stk-app-with-simtrace_until-sim-pin.pcap.pcapng.gz | 80.3 KB | 02/06/2019 | Denis 'GNUtoo' Carikli |
| good-orange-sim-no-stk-app-with-simtrace_until-sim-pin.log | 305 KB | 02/06/2019 | Denis 'GNUtoo' Carikli |
| bad-sfr-sim-no-stk-app-with-simtrace.pcap.pcapng.gz | 72.7 KB | 02/06/2019 | Denis 'GNUtoo' Carikli |
| bad-sfr-sim-no-stk-app-with-simtrace.log | 278 KB | 02/06/2019 | Denis 'GNUtoo' Carikli |
| maguro-takju-imm76i.pcapng.gz | 1.55 KB | 04/26/2019 | Denis 'GNUtoo' Carikli |
| maguro-takju-imm76i.log | 44 KB | 04/26/2019 | Denis 'GNUtoo' Carikli |
| maguro-replicant.pcapng.gz | 1.54 KB | 04/26/2019 | Denis 'GNUtoo' Carikli |
| maguro-replicant.log | 235 KB | 04/26/2019 | Denis 'GNUtoo' Carikli |
| maguro-takju-imm76i.pcapng.gz | 1.49 KB | 04/26/2019 | Denis 'GNUtoo' Carikli |
| maguro-takju-imm76i.log | 41.8 KB | 04/26/2019 | Denis 'GNUtoo' Carikli |
| maguro-replicant-4.2.log | 101 KB | 07/31/2019 | Denis 'GNUtoo' Carikli |
| radio_maguro_master_23_sept_2019.log | 129 KB | 09/23/2019 | Denis 'GNUtoo' Carikli |