pySim - Feature #4532

fix writing the various PLMN files

05/05/2020 03:14 PM - laforge

Status:	Stalled	**Start date:**	05/05/2020
Priority:	High	**Due date:**	
Assignee:	dexter	**% Done:**	70%
Category:			
Target version:			
Spec Reference:			

Description

Changing the MCC/MNC prefix of the IMSI is not sufficient anymore for making phones assume that the new MCC/MNC is their home network. In addition, a variety of files with "PLMN" in their name may need updating:

- PLMNsel
- PLMNwAcT
- OPLMNwAcT
- HPLMNAcT

However, related updates don't always happen on all cards, as they should.

History

#1 - 05/05/2020 03:39 PM - laforge

- Checklist item [ ] verify sysmoISIM-SJA2 fix added
Checklist item [ ] ensure we have a test covering all PLMN files are written correctly added
Checklist item [ ] port fix to WavemobileSIM (if needed, maybe it doesn't haven have that EF?) added

- Subject changed from support writing the various PLMN files to fix writing the various PLMN files

- Assignee set to dexter

- Priority changed from Low to High

Contrary to my original assumption, we actually do have code to write them. This code is used in both SysmoUSIMSJS1 and FairwavesSIM. However, the code is missing from the sysmoISIMSJA2 (and also from WavemobileSIM).

dexter, I created a trivial fix for the sysmoISIMSJA2, see https://gerrit.osmocom.org/c/pysim/+/18051

Please verify this patch, make sure we have test coverage (our tests didn't discover this bug, they should!) and also consider expanding the fix to WavemobileSIM

#2 - 05/05/2020 03:39 PM - laforge

- Description updated

#3 - 05/11/2020 11:45 AM - dexter

- Checklist item [x] verify sysmoISIM-SJA2 fix set to Done

- File mksim_sja2.sh added

- File sysmo_usim_sja2_after_write.txt added

- File sysmo_usim_sja2_before_write.txt added

- Status changed from New to In Progress

- % Done changed from 0 to 20

I have checked the problem back and from what I can see PLMNsel, PLMNwAcT, OPLMNwAcT, HPLMNAcT are updated. I have attached the output of pySim-Read.py before and after the write. So from what I can see the patch is working fine.

#4 - 05/12/2020 12:11 PM - laforge

There are some open questions on the correctness of the encoding of the files (see related sysmocom customer support inquiry)

#5 - 05/12/2020 04:44 PM - dexter

The implementation of enc_plmn() in utils.py seems have a problem with padding unused digits correctly.

See 3GPP TS 51.011, 10.3.4 gives an example, lets try encoding it manually and see if the result matches up

```
246 81 MCC/MNC
246F81 Padd unused digit with 0xF
42F618 Swap nibbles (matches also output of enc_plmn()
```

And now lets try the PLMN we try to use

```
262 02
262F02
62F220
```

When calling enc_plmn() manually with 262 02, we get: 62f22f

I think the way enc_plmn handles leading zeros is wrong.

#6 - 05/12/2020 05:00 PM - dexter

- File mobilcom_sim.txt added

I had a look at a simcard from a productive network. There the leading zeros of the MNC are also preserved but what confuses me is that the PLMNsel file does not start with 62f210, which would be the home network (262 01).

Here is what what the beginning of my PLMNsel looks like:

```
32f23012f47022f21002f8100

32f230 => 232f03 => 232 03
12f470 => 214f07 => 214 07
```

Attached one can find the dump from the simcard I used for this test.

#7 - 05/12/2020 06:31 PM - laforge

In case this is not obvious: MNCs come as 2-digit MNC or as 3-digit MNC.

Hence, 262-02 and 262-002 are different networks.

The encoding as per TS 04.08 is implemented in libosmocore for more than a decade and we use it pretty much anywhre on the radio interface, whether in RR or MM or other sub-layers such as BSSMAP. They python code should just simply replicate what the C code is doing.

The encoding of the above two examples should be:

- 262-02 -> 62F220
- 262-002 -> 620220

Furthermore, it is not surprising that EF_PLMNsel doesn't list the home operator (derived from the IMSI prefix). In classic 2G SIM cards, the home network is always derived from the IMSI prefix, and it cannot be overridden. Only in USIM, it is possible that the HPLMN MCC/MNC is explicitly set different (e.g. like the EF_HPLMNwAcT

#8 - 05/12/2020 07:15 PM - laforge

It seems like pySim unfortunately treats mnc as integer. So there is no way to know if the user entererd 02 or 002 (referring to the above example). I suggest to

- treat MNC as a string
- verify that it only contains decimal digits
- pad it to three characters length (if it is 2 only) using a leading F
- then concatenate + nibble-swap with the MCC in enc_plmn()

I hacked this up in https://gerrit.osmocom.org/c/pysim/+/18225 and it seemed to work locally with both 2-digit and 3-digit MNC as well as 1/2/3 digit MCC.

#9 - 05/14/2020 07:29 PM - laforge

- % Done changed from 20 to 70

patch has been merged. dexter, please check the two missing checklist items in this ticket and take any action, if required.

#10 - 01/20/2021 08:56 PM - laforge

- Status changed from In Progress to Stalled

Files

mksim_sja2.sh	326 Bytes	05/11/2020	dexter
sysmo_usim_sja2_after_write.txt	5.65 KB	05/11/2020	dexter
sysmo_usim_sja2_before_write.txt	5.62 KB	05/11/2020	dexter
mobilcom_sim.txt	4.8 KB	05/12/2020	dexter