PySIM Osmoscom + OpenJCVM

Haddad Rafik LCS - lcs-ota-profiles-issues-3

link : https://github.com/osmocom/pysim

pySim is a suite of programs (develped in python) for interfacing with SIM/UICC/USIM/ISIM cards.

The most recommended configuration is to use a Smart Card Interface device that complies with the USB CCID specification, utilizing the libccid/pcsc-lite driver stack along with the `pyscard` library.
Other project like sniffer-apdu and blank SIM need to get link :
https://github.com/LudovicRousseau/pyscard

python smart card library - is a python module adding smart cards support.

It supports common smart card operations such as reading, writing, and authentication. This includes sending and receiving APDU (Application Protocol Data Unit) commands to and from the smart card.

Connect SIM card reader. And Insert programmable SIM card .

PySim-prog

Here we find documentation: link

https://osmocom.org/projects/pysim/wiki/PySim-prog

parameters:

```
 > SMSP    : e1fffffffffffffffffffffffff0581005155f5ffffffffffff000000
 > ICCID   : 89882110000000110000
 > MCC/MNC : 801/71
 > IMSI    : 901990000000018
 > Ki      : 1D8B2562B992549F20D0F42113EAA6FA
 > OPC     : 398153093661279FB1FC74BE07059FEF
 > ACC     : None
```

./pySim-prog.py -help

```
Usage: pySim-prog.py [options]
Options:
  -h, --help            show this help message and exit
  -d DEV, --device=DEV  Serial Device for SIM access [default: /dev/ttyUSB0]
  -b BAUD, --baud=BAUD  Baudrate used for SIM access [default: 9600]
  -p PCSC, --pcsc-device=PCSC
                        Which PC/SC reader number for SIM access
  -t TYPE, --type=TYPE  Card type (user -t list to view) [default: auto]
  -a PIN_ADM, --pin-adm=PIN_ADM
                        ADM PIN used for provisioning (overwrites default)
  -e, --erase           Erase beforehand [default: False]
  -S SOURCE, --source=SOURCE
```

```
                        Data Source[default: cmdline]
  -n NAME, --name=NAME  Operator name [default: Magic]
  -c CC, --country=CC   Country code [default: 1]
  -x MCC, --mcc=MCC     Mobile Country Code [default: 901]
  -y MNC, --mnc=MNC     Mobile Network Code [default: 55]
  -m SMSC, --smsc=SMSC  SMSP [default: '00 + country code + 5555']
  -M SMSP, --smsp=SMSP  Raw SMSP content in hex [default: auto from SMSC]
  -s ID, --iccid=ID     Integrated Circuit Card ID
  -i IMSI, --imsi=IMSI  International Mobile Subscriber Identity
  -k KI, --ki=KI        Ki (default is to randomize)
  -o OPC, --opc=OPC     OPC (default is to randomize)
  --op=OP               Set OP to derive OPC from OP and KI
  --acc=ACC             Set ACC bits (Access Control Code). not all card types
                        are supported
  -z STR, --secret=STR  Secret used for ICCID/IMSI autogen
  -j NUM, --num=NUM     Card # used for ICCID/IMSI autogen
  --batch               Enable batch mode [default: False]
  --batch-state=FILE    Optional batch state file
  --read-csv=FILE       Read parameters from CSV file rather than command line
  --write-csv=FILE      Append generated parameters in CSV file
  --write-hlr=FILE      Append generated parameters to OpenBSC HLR sqlite3
  --dry-run             Perform a 'dry run', don't actually program the card
```

All Subscriber Identity Module (SIM) and USIM cards adhere to the ISO 7816 standard. Additionally, they conform to ETSI and 3GPP standards. Both ETSI standards (available at ETSI) and 3GPP standards (available at 3GPP) are publicly accessible.

The *ISO7816* class within the *ICC* module provides basic methods for many of the ISO7816 commands,Link

https://github.com/mitshell/card