Python libraries (mostly) for telecommunication systems:

card, *CryptoMobile*, *libmich* & *corenet*

https://github.com/mitshell/

card

- Smartcard communication library
- Runs on top of *pyscard* library
- Mostly dedicated to SIM / USIM cards
  - select(), reads directory and file metadata, and file content if permission granted
  - get_ICCID(), get_imsi()
  - run_gsm_alg(), authenticate()
  - read_services(), reads and prints (U)SIM service table
- Bruteforces and graphs (U)SIM's filesystem
- Includes scripts to personalize sysmo-SIM and USIM

CryptoMobile

- Cryptographic algorithms for mobile networks
- C implementation and Python bindings:
  - Kasumi (3G algo)
  - SNOW 3G (3G and LTE algo)
  - ZUC (LTE algo)
- CMAC mode of operation for AES (LTE algo), on top of *pycrypto*
- Milenage, for authenticating with USIM cards, using AES from *pycrypto*

libmich

- Library for easy format encoding / decoding
  - Inspired from *scapy*
- Supports (sometimes not that well) many formats:
  - Ethernet, VLAN, IPv4, IPv6, UDP, TCP, PPP, SCTP, SIGTRAN, BGPv4, TLS, RTP, EAP, EAP-SIM, EAP-AKA, IKEv2
  - GTPv1, GTPv2, L1CTL, LAPDm, L3GSM_RR (including CSN.1), L3Mobile (MM, CC, SMS, SS, GMM, SM, EMM, ESM), UMA, UICC Secure Channel
  - IEEE 802.11, IEEE 802.15.4
  - BMP, PNG, JPEG, pcap, MPEG2 (transport stream), MPEG4 (container), ELF header

libmich (continued)

- Runs an ASN.1 compiler, BER and PER (unaligned / aligned) codecs:
  - RRC (3G and LTE)
  - RANAP (3G), S1AP and X2AP (LTE)
  - PCAP (3G), LPP (LTE)
  - MAP, SS, TCAP (ongoing)
- And contains other various useful routines
  - PRF 186.2, Diffie-Hellman, CRC
  - shtr (to shift *str* …), and more !

```
>>> a = shtr('hackito ergo sum 2015')
>>> a << 15

'\xb1\xb5\xb4\xba7\x902\xb93\xb7\x909\xba\xb6\x90\x19\x18\x18\x9a\x80'
```

libmich (finally)

- Supports advanced functions to run an LTE mobile core network
  - MME
  - Authentication vector generator (HSS-like)
  - GTP tunnel handler (PGW-like)
  - SMS handler (ongoing)
- Powers the newly published *corenet* application: an LTE core network within a single Python script

corenet

LTE-Uu S1 SGi

UE <-------Radio/UP------> eNodeB <--GTPU/UP---> corenet <---UP---> LAN

UE <--Radio/RRC/NAS--> eNodeB <--S1AP/NAS--> corenet

eNodeB <------S1AP-----> corenet

- Requires eNodeB(s) (we uses the Amarisoft one), LTE terminals and customizable USIM
- Everything runs within an IPython interpreter
  - access dynamically to all classes and instances settings
  - run network-initiated procedures interactively

Thank you

- Thanks to ANSSI for allowing me to open-source this
- Do not hesitate to
  - Use
  - Report experience
  - Modify, extend, contribute
  - Provide captures / pcap

https://github.com/mitshell/