# Osmocom SIMtrace2 Tutorial

## SIM card protocol tracing - why and how

Harald Welte

sysmocom - s.f.m.c. GmbH

OsmoDevCall 2022-10-19

# Terminology

| | |
|---|---|
| SIM | Subscriber Identity Module |
| USIM | Universal Subscriber Identity Mdoule |
| UICC | Universal Integrated Chip Card |
| MS | GSM Mobile Station (phone, modem) |
| UE | UMTS User Equipment |
| ME | GSM Mobile Equipment (MS + SIM) |
| OTA | Over The Air |
| SAT | SIM Application Toolkit |
| CAT | Card (UICC) Application Toolkit |
| USAT | USIM Application Toolkit |
| TAR | Toolkit Application Reference |

# Relevant Specification Bodies

- ISO (ISO 7816) smart cards
- ETSI (Eurpoean Telecomms Standardisation Institute)
  - Classic GSM SIM
  - UICC card as basis for various telecom ID purposes
  - Card Application Toolkit (CAT)
- 3GPP (3rd Generation Partnership Project)
  - USIM Application
  - USIM Application Toolkit (USAT)
  - API based applet interworking
- Global Platform
  - Overall spec for SIM/USIM with Java
- Sun Microsystems (now Oracle)
  - Java Card Virtual Machine
  - Java Card Runtime Environment

# The Subscriber Identity Module (SIM)

- Basic idea was to store cryptographic identity of subscriber inside smart card
- User can thus migrate identity from one device to another
- User can furthermore use different SIM in same device (e.g. local prepaid SIM while travelling)
- Original SIM card design mostly ISO 7816-4 filesystem and single command to execute A3/A8 algorithm inside card
  - This could even be done in logic, no processor required

# The modern SIM

The modern SIM is an entirely different beast

- Cryptographic processor smart card
  - Symmetric cryptography such as DES, 3DES, AES
  - Public key cryptography such as RSA, ECC
- Java Card including a small Java VM and Java RE
- Multiple application support
- Ability to download applications (Applets) into card

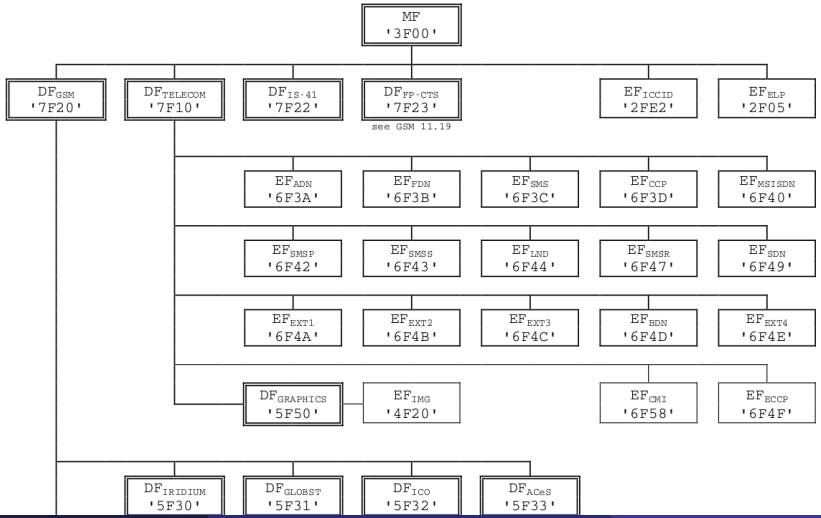
# Smart Card Basics

- microprocessor with RAM, Flash and Operating System
- Interface: Electrical + Logical Protocol (ISO7816-3, ISO7816-4)
- File System based representation of information
- Protocol describes remote operations on the file system
- Few non-filesystem related commands for e.g. authentication

# Smart Card Filesystem

- Hierarchical file system like on PC
  - MF (master file): root directory
  - DF (dedicated file): subdirectory
  - EF (entry file): actual file
    - transparent or record oriented
    - record linear fixed/variable or record cyclic
- File names don't exist on card. 16bit FID (File ID) or 8bit SFID used instead

# Smart Card Filesystem Hierarchy



# Smart Card Filesystem Permissions

- similar to 'permission bits' on Linux or other PC OS
- each file can define separate read/write permissions
- some cards are permanently read-only
- other files can be written to after regular PIN verification
- yet another set of files e.g. needs one of the ADM PINs

# SIM Card APDU Commands

Classic SIM card commands include the following

- SELECT (change directory / open file)
- READ BINARY, UPDATE BINARY (read/write transparent EF)
- READ RECORD, UPDATE RECORD (read/write record EF)
- ENABLE CHV, DISABLE CHV, CHANGE CHV (enable, disable or change PIN)
- VERIFY CHV, UNBLOCK CHV (verify or unblock PIN)
- RUN GSM ALGORITHM (A3/A8 authentication)

# Smart Card Filesystem

Typical operations of the phone include

- navigating inside filesystem by SELECT on DF/EF
- authenticating the user PIN
- reading/updating files
  - reading IMSI
  - old-school SMS and contact storage
  - storing session keys (Kc/KcGPRS, ...)
  - storing last cell on power-off

# Smart Card PINs

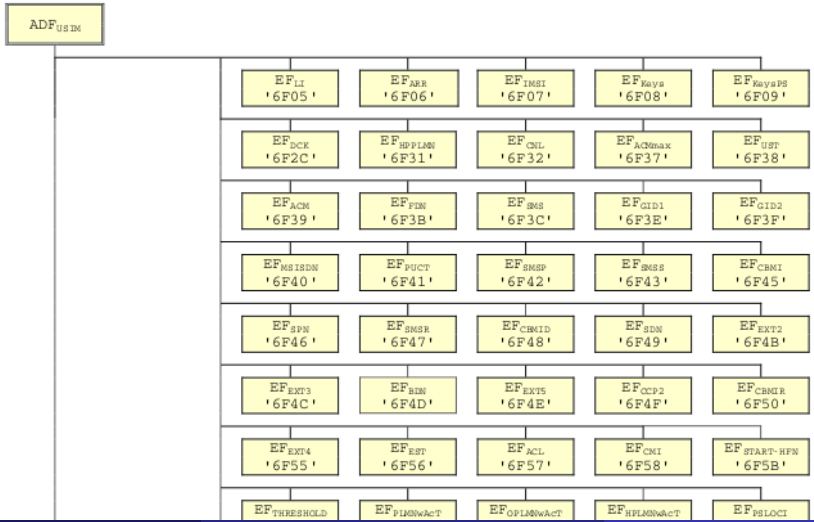
The level of access to the filesystem and other card features is determined by authentication using a shared secret, called 'PIN'.

- Regular PIN for normal use of the card by the end user
- PUK for resetting the pin after too many retries
- ADM1..n PIN for access by the operator only

# Multi-Application Smart Cards

- Classic SIM cards are single application, accessing the GSM related files works by entering the known DF.GSM directory with its well-known FID
- Later the idea of multi-application smart cards entered the market
- A multi-application smart card contains an EF.DIR in the MF
- EF.DIR contains records with the AIDs of all applications on the card.
- AID prefix is well-known to the application, AID suffix is manufacturer specific. Applications use prefix-match
- application specific directory can be entered by SELECT on the AID

# USIM Application Dedicated File (ADF.USIM)



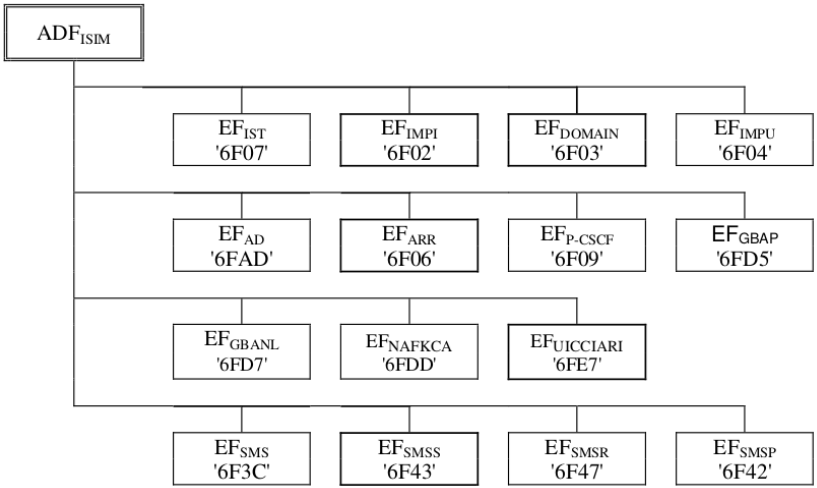
# Evolution of the SIM

- Classic GSM SIM cards
  - initial GSM / ETSI TS 11.11 for classic GSM SIM, based on ISO 7816-2/3/4
  - small changes for GPRS support by introducing a few new optional files
  - Class byte 0xA0 used in GSM SIM
- USIM cards
  - Completely new approach based on ETSI UICC spec, multi-application capable
  - Selection of ADF.USIM by AID
  - Many new files
  - backwards compatibility achieved by placing DF.GSM in MF and linking (think of symlink/hardlink) of relevant files
  - Authentication for GSM and UMTS can be completely different (algorithm, secret key used, ...)
- Additional application profiles exist for GSM-R, TETRA and other ETSI related communications systems.

# Evolution of Specifications

- Classic SIM: ETSI TS 11.11 / 3GPP TS 51.011
- UICC Card: 3GPP TS 31.101, 31.900, ETSI TS 102 221, 102 222
- USIM application: 3GPP TS 31.102
- ISIM application for IMS (VoIP for LTE): 3GPP TS 31.103

# ISIM Application Dedicated File (ADF.ISIM)



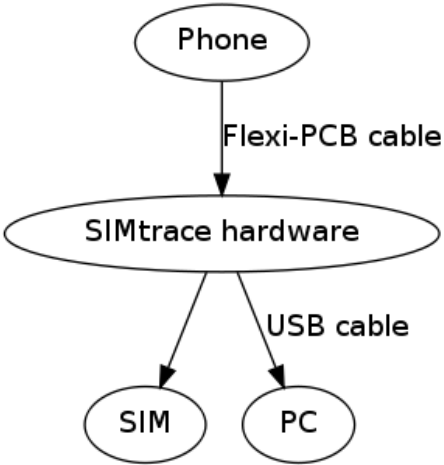
# Analyzing SIM problems

- Regular end-user phone does not give much debugging
- SIM card itself has no debug interface for printing error messages, warnings, etc.
- However, as SIM-ME interface is unencrypted, sniffing / tracing is possible
- Commercial / proprietary solutions exist, but are expensive (USD 5,000 and up)
- Technically, sniffing smart card interfaces is actually very simple

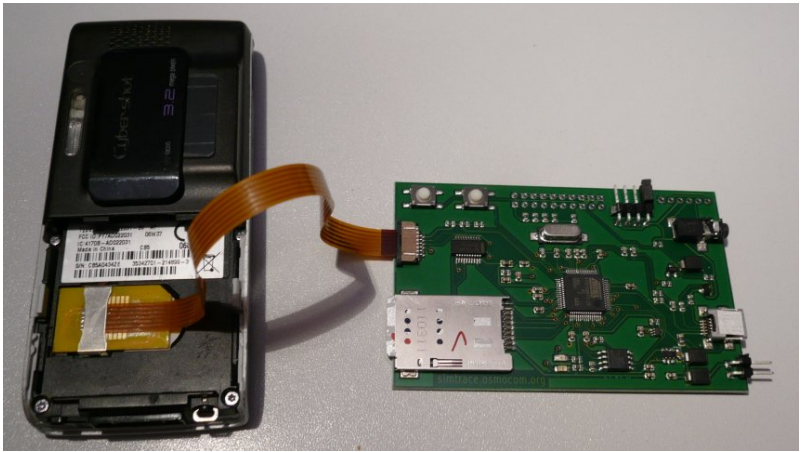
# Introducing Osmocom SIMtrace2

- Osmocom SIMtrace2 is *primarily* a passive (U)SIM-ME communication sniffer
- Insert SIM adapter cable into actual phone
- Insert (U)SIM into SIMtrace2 hardware
- SIMtrace2 hardware provides USB interface to host PC
- `simtrace2-sniff` host PC program encapsulates APDU in GSMTAP
- GSMTAP is sent via UDP to localhost
- wireshark dissector for GSM TS 11.11 decodes APDUs
- NEW: pySim-trace for higher-level decoding

# Osmocom SIMtrace2 Principle



# Osmocom SIMtrace2 Hardware



# History: Osmocom SIMtrace1 Hardware

- before 2015, there was a SIMtrace hardware, now called SIMtrace1
- based on much older AT91SAM7S controller (ARM7TDMI)
- firmware was a crude extension of an earlier RFID project (OpenPCD)
- SIMtrace1 is unsupported; it uses different firmware and host tools
- if somebody really cared, SIMtrace2 firmware *could* could in theory be ported to support SIMtrace1 hardware
- in case of doubt, check marking of TQFP chip on the device. If it's `SAM3S....` you're good.

# Osmocom SIMtrace2 Hardware

- Hardware is based around AT91SAM3S controller
- SAM3S Offers two ISO 7816-3 compatible USARTs
- USARTs can be clock master (SIM reader) or slave (SIM card)
- Open Source Firmware available
- Auto-bauding depending CLK signal, PPS supported
- Schematics / layout is open source (CC-BY-SA)
- Source at `https://gitea.osmocom.org/sim-card/simtrace2` in the `hardware` directory
- Assembled + tested kits can be bought from `https://shop.sysmocom.de/`

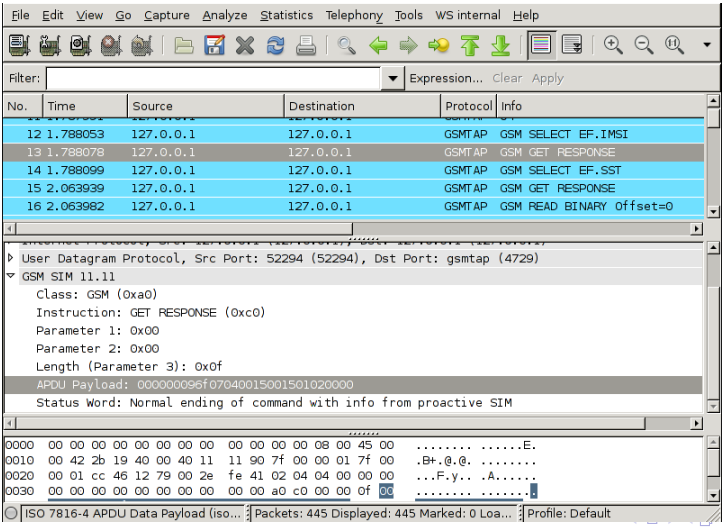
# Osmocom SIMtrace2 Firmware

- Open Source (GPLv2) Firmware on SAM3S implementing
  - `dfu` DFU bootloader for easy (and standardized) firmware flashing
  - `cardem` card physical layer emulation / remote SIM
  - `trace` passive protocol tracing
- Source at `https://gitea.osmocom.org/sim-card/simtrace2` in the `firmware` directory
- Binaries at `https://downloads.osmocom.org/binaries/simtrace2/firmware/`
- not only for SIMtrace2, but other boards like ngff-cardem, sysmoQMOD

# Osmocom SIMtrace2 Host Software

- Open Source (GPLv2) Host Software (for Linux), implementing
- implementing the following parts:
  - `libosmo-simtrace2` library encapsulating bulk of the functionality
  - `simtrace2-sniff` for protocol tracing with `trace` firmware
  - `simtrace2-list` to list all compatible devices connected via USB
  - `simtrace2-tool` for some miscellaneous features (ngff-cardem/QMOD)
- Source at `https://gitea.osmocom.org/sim-card/simtrace2` in the `host` directory
- Packages (dpkg, rpm) at `https://osmocom.org/projects/cellular-infrastructure/wiki/Binary_Packages`
- not only for SIMtrace2, but other boards like ngff-cardem, sysmoQMOD

# wireshark decoding



# wireshark decoding - DEMO

DEMO

# New in 2022: `pySim-trace` decoding

- basic APDU level decode in wireshark is all fine, but rather limited
- interesting bits are actually happening at application layer above
- every file has different content/format/encoding
- if we have code to decode the file contents, we can provide higher-level decode
- this led to `pySim-trace`
- `pySim` is the Osmocom *swiss army knife* for SIM/USIM/ISIM card reading/writing
  - It already has encoders/decoders for most of the files
  - `pySim-trace` consumes GSMTAP and maintains state (which file is currently selected, ...) to then use those decoders

# New in 2022: `pySim-trace` decoding - DEMO

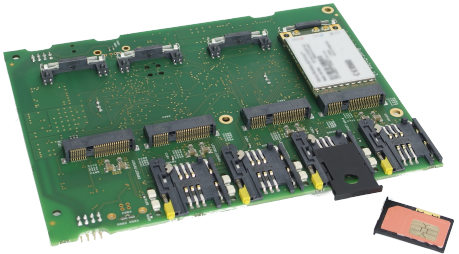
DEMO

# SIMtrace2 *card emulation* / *remote SIM*

- The SIMtrace2 hardware can emulate the physical SIM card interface
- This means that SIMtrace2 is connected to SIM *instead* of a SIM
- The communication is picked up and passed via USB to the host
- Host can now, for example, forward this communication to a (remote) smart card reader with the actual SIM
  - `simtrace2-cardem-pcsc` is a simplistic implementation of that: Pass communication to a locally connected PC/SC compatible reader
  - The `osmo-remsim` software suite is a comprehensive software package for managing a fleet of phones/modems and SIM cards, allowing dynamic assignment of remote SIMS to phones/modems.
  - See a previous OsmoDevCall (`https://media.ccc.de/v/osmodevcall-20210827-laforge-osmo-remsim`) for a talk on that

# sysmoQMOD board

- a proprietary board hosting two SAM3S with SIMtrace2 *cardem* firmware
- each SAM3S serves two cellular modems in mPCIe form-factor
- can pick up SIM signalling of four modems and pass it to remote SIMs
- Product page:
  `https://sysmocom.de/products/lab/sysmoqmod/index.html`



# ngff-cardem board

- a NGFF (M.2) cellular modem carrier board with on-board SIMtrace2
- allows SIM tracing and card emulation/forwarding without any flex cables
- an open source hardware project, just like SIMtrace2
- Homepage: `https://osmocom.org/projects/ngff-cardem/wiki`

# SIMtrace2 TODO

SIMtrace2 hardware is capable, but no software yet for:

- Use board as CCID / PC/SC compatible smart card reader
- perform MITM (APDU filtering)
- T=1 protocol support (tracing of crypto smart cards, banking cards)
- autonomous tracing operation (No PC / USB), store APDU logs *in the field* on integrated SPI flash

Firmware and host software all FOSS, anyone can extend and innovate!