SIM card technology from A(PDU) to X(RES)

Harald Welte

osmocom.org

Chaos Communication Congress 2019

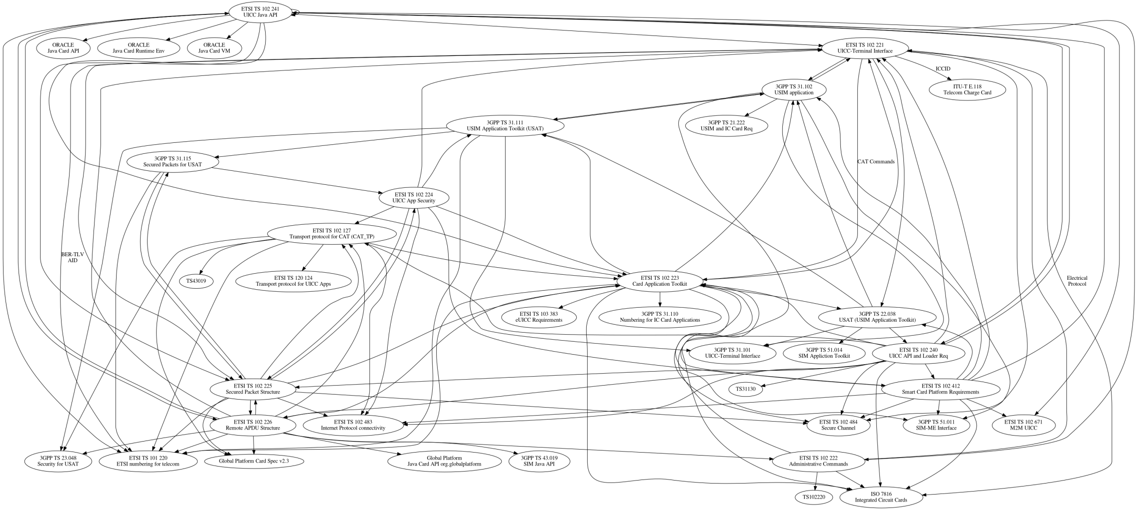
Outline

- Relevant Specs + Spec Bodies
- Card Interfaces, Protocols
- Card File System
- SIM Evolution from 2G to 5G
- SIM Toolkit
- OTA (Over The Air)

About the speaker

- Free Software + OSHW developer for more than 20 years
- Used to work on the Linux kernel from 1999-2009
- working with contact chip cards since 1999, contactless since 2006
- developing FOSS in cellular communications (Osmocom) since 2008
  - developed various SIM card related tools in software an hardware
- Living and working in Berlin, Germany.

Relation of SIM card specifications



Relevant specification bodies/sources

- ISO (Integrated Circuit[s] Card)
- ITU (Telecom Charge Cards)
- ETSI (where GSM was originally specified)
- 3GPP (where 3G to 5G was specified)
- GlobalPlatform Card Specification
- Sun/Oracle JavaCard API, Runtime, VM
- GSMA

The SIM: Subscriber Identity Module

- probably anyone in the audience has at least one, likely more
- ubiquitous; every device withe cellular connectivity has at least one
- not many people outside the telecom industry ever look at them in much detail
- SIM card hacking (in the security sense) has a tradition at CCC since at least 1998
  - Vodafone Germany SIM card cloning: https://ftp.ccc.de/software/gsm/gsm_hack.tar.gz
  - SIM card simulator in Turbo C (1998): https://ftp.ccc.de/software/gsm/SIM_sim.zip
- meanwhile: SIM technology stack is getting more complex and deep
- let's recap what SIM cards actually are, and what they do

Classic SIM in early GSM

- Idea of storing subscriber identity predates GSM (e.g. C-Netz since 1988)
- GSM from the very beginning introduces concept of SIM card
- store subscriber identity outside of the phone
- store some network related parameters
  - static (like access control class)
  - dynamic (like TMSI, Kc, ...)
- full credit card size so it can be used in radios installed in (rented, shared company) cars.

DIN EN ISO/IEC 7816

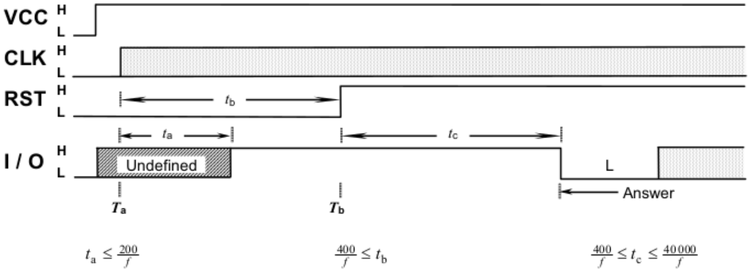
- the *mother of all smart card* spec
- "Integrated circuit(s) cards with contacts"
- 15 parts, most relevant are below:
  - Part 1: Physical characteristics
  - Part 2: Dimensions and location of the contacts
  - Part 3: Electronic signals and transmission protocols
  - Part 4: Interindustry commands for interchange
    - Why not international inter-industry commands for interworking information interchange? Anyone?

GSM TS 11.11

- *Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) Interface*
- repeats (and some times amends) large portions of 7816-1/2/3/4
  - Section 4: physical characteristics
  - Section 5: electronic signals, transmission protocol
- but also specifies what makes the SIM a SIM: Information model, file system, commands
- last, but not least how to execute authentication: RUN GSM ALGORITHM

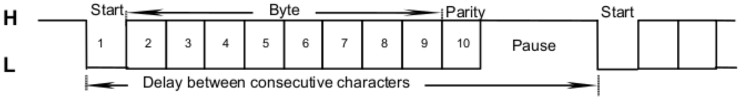
Physical Smart Card Interface

- Relevant pins:
  - VCC: Provides supply voltage (5V, 3V or 1.8V)
  - CLK: Provides a clock signal (1 .. 5 MHz default)
  - RST: To reset the card
  - IO: bidirectional serial communications
- Activation sequence triggers card to send ATR (Answer To Reset)



Bit transmission level

- despite the clock, communication is asynchronous!
- baud rate derived from divided clock
- no defined phase relationship between clock and data
- serial data is just like UART/RS232, ... but:
  - one line for both Rx and Tx
  - direction changes once after every byte (ACK in T=0)
  - direction changes every few bytes (TPDU state machine)
  - timings are actually not very well specified



Smart Card Communication

- based on APDU (Application Protocol Data Unit) as per ISO 7816-4
  - CLA (class byte)
  - INS (instruction byte)
  - P1, P2 (parameter bytes)
  - Lc (command length)
  - Command data
  - Le (expected response length)
  - Response data
  - SW (status word)

Smart Card Transmission Protocol

- different protocols transceive APDUs from/to cards
- T=0 most commonly used with SIM cards
- T=1 also possible but rarely used in SIM
  - specs require phones to implement both T=0 and T=1
  - SIM card can be either T=0 or T=1
  - T=1 more used in banking / crypto smart card world
- APDU gets mapped to protocol-specific TPDU (Transmission Protocol Data Unit)
  - : Example Command TPDU: A0 A4 00 00 02 3F 00
  - : Example Response TPDU: 90 00 (just status word)

Card Filesystem

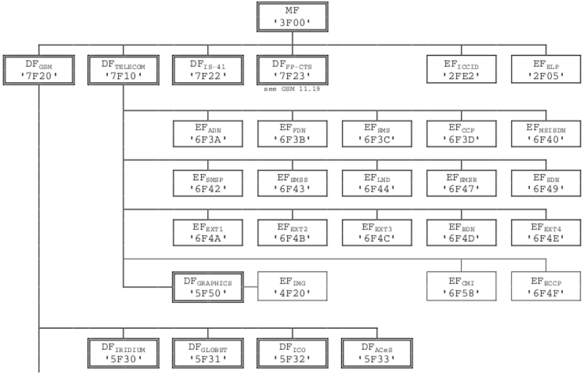
- most smart cards contain file system abstraction
- cannot be mounted (not exposed like a block device / USB drive!)
- access based on file-level commands (analogy: more like MTP/PTP)
- some similarities to general-purpose (computer) OS file systems:
  - MF: Master File (root directory)
  - DF: Dedicated File (subdirectory)
  - EF: Elementary File (data file)
- However, much more comprehensive than computer OS file systems, e.g.
  - *transparent EF*: opaque stream of data, like PC
  - *linear fixed EF*: fixed-size records, seekable
  - *cyclic fixed EF*: ring buffer of records, seekable
  - *incrementable*: for monotonically incrementing counters
- Each file has Access Control Conditions (ACC)
  - read/write/update only after PIN1/PIN2/ADM-PIN

Key SIM Card Commands

- SELECT (file)
- READ RECORD / UPDATE RECORD
  - for record-oriented EF
- READ BINARY / UPDATE BINARY
  - for transparent EF
- CHANGE CHV / DISABLE CHV / ENABLE CHV
  - CHV: Card Holder Verification (PIN)
- RUN GSM ALGORITHM
  - ask SIM to execute authentication algorithm in card

SIM card filesystem hierarchy

- MF (3F00)
  - DF_TELECOM (7F10)
    - EF_SMS
    - EF_MSISDN
    - ...
  - DF_GSM (7F20)
    - EF_IMSI
    - EF_Kc
    - ...
  - EF_ICCID
  - ...



3G: ETSI UICC and the 3GPP USIM

- The GSM SIM was fully specified by ETSI in TS 11.11
- As GSM specs moved from ETSI to 3GPP, card specs were split:
  - ETSI UICC (Universal Integrated Circuit Card)
    - like a *base class* abstracting out those parts that are not cellular related, or at very least not 3GPP network related
  - 3GPP USIM Application on top of UICC
    - specifies those parts specifically relevant to 3GPP networks
    - implemented in ADF_USIM (Application Dedicated File)
    - ADF can be entered via SELECT, similar to classic DF

4G: The ISIM

- 3G/LTE reuses the existing 3G Authentication (UMTS AKA)
- 4G/LTE simply reuses existing USIM
- some new optional files were introduced in ADF_USIM
- IMS (IP Multimedia System used for not only VoLTE) specifies ISIM application
  - stores additional IMS related parameters like SIP server / user identity
  - presence of ISIM not required for IMS to work
  - if present, ISIM application present next to USIM (and possibly SIM)

SIM cards in 5G

- 5G reuses existing 3G/4G USIM
- some new optional files were introduced in ADF_USIM
- SUCI (Subscriber Concealed Identifier) can optionally be computed by SIM
    - this is the only feature requiring different card / apps on card

SIM card hardware

- processor core
  - many different vendors and architectures, from 8-bit 8051 to 32bit ARM
  - today quite often ARM SCxxx "Secure Core" family
  - documentation on hardware, often even simple one-page data sheets not public
- built-in RAM
- built-in ROM (at least boot loader, possibly also OS)
- built-in flash (file system storage, possibly also OS, applications)
- contrary to expensive crypto smart cards, SIM card chip mostly selected purely by low cost
  - blame pre-paid cards for that

SIM card software

- Every Smart Card has a Card Operating System (COS)
- Cards without COS are simple memory cards (like I2C EEPROM), insufficient for SIM
- Card OS for Crypto Smart Cards (banking, access control) often publicly known
- SIM Card OS are rarely known / publicly documented or even named
- Example: ARM not only offers SIM card CPU core designs, but also OS (Kigen OS)
- SIM Card OS is *implementation detail*, almost everything relevant is standardized across OS vendors

SIM card software modularity

- Early SIM cards were (likely) monolithic,
  - no separation between OS and SIM application
- Today, SIM cards software is modular
  - Core OS
  - Applications (SIM, USIM, ISIM, ...)
- traditionally, OS very chip/hardware dependent, non-portable
- traditionally, applications very OS dependent, non-portable

Java Cards

- independent of SIM cards, Java Smart Cards have been developed
- based on Java Card Platform specifications by Sun (now Oracle)
- first cards in 1996 by Schlumberger (now Gemalto)
- independent of SIM cards, Java Smart Cards have been developed in 1996 by Schlumberger
- most cards implement GlobalPlatform specifications for vendor-independent management
  - super constrained, weird subset of Java
  - special on-card VM (not normal JVM)
  - special CAP format (not normal JAR)
  - Idea: Portability of Cardlets (card applications)

Java SIM Cards

- There is no functional requirement for a SIM/USIM/ISIM to be a java card
- In reality, most SIM cards probably are Java Cards these days
- Portability is the main driver here
- Operators want to share same applications over multiple vendors/generations of cards
- 3GPP and ETSI specify Java APIs / packages available specifically on Java SIM cards

SIM Toolkit (STK) / Card Application Toolkit (CAT)

- Ability by card to offer applications with UI/menu on the phone
- New APDUs/Instructions
  - TERMINAL PROFILE
  - ENVELOPE
  - FETCH
  - TERMINAL RESPONSE

Proactive SIM

- SIM cards are "slave" in the 7816 interface
- All actions are triggered by the phone, card can only respond
- Proactive SIM works around this restriction
- Piggy-backs proactive commands to card responses
- Phone can be requested to poll the SIM if it has some proactive commands pending
- Phone can be requested to provide event notifications

OTA (Over-The-Air)

- Ability for operator to transparently communicate with SIM card in the field
- Based on Proactive SIM
- Can use different transport channels, such as
  - SMS-PP (normal SMS as you know it)
  - SMS-CB (bulk update of cards via cell broadcast)
  - USSD (Release 7)
  - BIP (via CSD, GPRS): ETSI TS 102 223 / TS 102 127
  - now also HTTPS (Release 9)
- Cryptographic security mechanisms specified, but detailed use up to operator
  - Message Authentication (optional)
  - Message Encryption (optional)
  - Replay Protection (optional)

Remote File Management (RFM)

- Introduced in Release 6
- Common use case of OTA
- Allows remote read / update of files in file system
- Example: Change of preferred/forbidden roaming operator list
- Example (ancient): Backup of phonebook at operator

Remote Application Management (RAM)

- Introduced in Release 6
- Common use case of OTA
- Allows remote installation / removal of applications on card
- Example: New multi-IMSI application (MVNOs)
- Example: New STK applications

OTA over HTTPs

- 4G and beyond don't natively support SMS-PP, USSD, ...
- In Release 9, OTA over HTTPs is first introduced
- References to GlobalPlatform 2.2 Amd B + ETSI TS 102 226
- Uses HTTP as per RFC 2616
- Uses PSK-TLS as per RFC4279, RFC4785, RFC5487
  - TLS 1.0 / 1.1: TLS_PSK_WITH_3DES_EDE_CBC_SHA
  - TLS 1.0 / 1.1: TLS_PSK_WITH_AES_128_CBC_SHA
  - TLS 1.0 / 1.1: TLS_PSK_WITH_NULL_SHA (RFC4785)
  - TLS 1.2: TLS_PSK_WITH_AES_128_CBC_SHA256 (RFC5487)
  - TLS 1.2: TLS_PSK_WITH_NULL_SHA256 (RFC5487)
- IP and TCP socket terminated in phone, only TCP payload handled by card

OTA over HTTPs

- Card acts as HTTP client performing HTTP POST
- TLS payload is remote APDU format of ETSI TS 102 226
- additional HTTP headers
  - X-Admin-Targeted-Application
  - X-Admin-Next-URI
  - X-Admin-Protocol: globalplatform-remote-admin/1.0
  - X-Admin-From
  - X-Admin-Script-Status
  - X-Admin-Resume

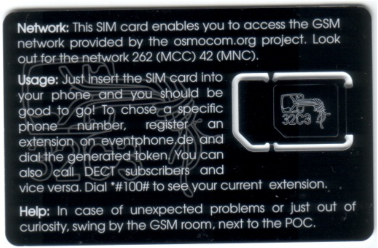
S@T

- a strange beast specified outside of ETSI/3GPP
- allows SIM toolkit applications without writing Java or native applications
- special byte code format interpreted by S@T browser
- to me, one of those WTF? technologies

GSMA eSIM

- system for remote provisioning of *profiles* to SIM
- allows change of operator / identity without replacement of physical card
- main use case is non-removable / soldered SIM chip (MFF2)
- also available from some operators in classic smart card size
- main relevant spec is GSMA SGP.22
- based around PKI between operators, all parties approved by GSMA

The CCC event SIM cards



- are Java SIM + USIM cards
- support OTA, RAM, RFM (via SMS-PP and maybe BIP, not HTTPS)
- you can get the ADM PIN and OTA keys from the event GSM team
- a "hello world" Java applet and tools for installation are provided (thanks to shadytel + Dieter Spaar)
- identities and key data can be modified using Osmocom pySim software

Further Reading (hyperlinked)

- SIM alliance stepping stones
- SIMtrace2 wiki
- Simjacker vulnerability
- SRLabs SIMtester
- for historians
  - CCC SIM simulator in Turbo C
  - CCC sim clone / D2 Pirat

Thanks

Thanks for your attention.
You have a General Public License to ask questions now :)