The blackbox in your phone

Hunz
Zn000h at gmail.com

CCC Camp 2011
10.08.2011
Contents

- Smartcards in general
- The SIM
  - filesystem
  - commands
- SIM application toolkit (SAT)
- Tools
- Summary
Smartcards: physical connections

- Not just memory, but a microcontroller → card decides, what the user can do

- Connections:
  - RST: Reset input
  - CLK: Clock input
  - IO: Data in/out
  - Vcc: supply voltage (1.8V / 3V)
Smartcards: data transmissions

- Serial protocol like RS-232
- But: only one data line → half duplex
- Request/response with Terminal as Master
- Baudrate depends on input clock
  - Initial baudrate = clk / 372
A simple smartcard terminal

- Phoenix & Smart-/Dumbmouse Terminals
- RS-232 UART used for communication
  - Card clock = 9600 baud * 372 = 3.5712 MHz
  - IO: Open collector w/ pullup
- RTS used for card reset (polarity may vary)
- Or: use a PC/SC reader
Smartcards: Protocol setup

- Card reset
- Card sends Answer-to-Reset (ATR)
  - Supported parameters, protocols, etc.
  - ATR: 3B <more stuff>
  - Decode w/ `pcsc_tools`: ATR_analysis
- Protocol-Parameter-Selection (PPS)
  - protocol+baudrate selection
  - optional, but heavily used nowadays
Smartcards: T=0 Protocol

- Communication via Application Protocol Data Units (APDU)
  - **CLA**: Instruction Class
  - **INS**: Instruction (command)
  - **P1, P2**: Instruction-specific parameters
  - **Len**: Data length
  - **Data** (optional) either to or from card
  - **SW1, SW2**: Status (from card)
Smartcards: T=0 Example

1) ADPU Header (Terminal → Card)
2) ACK (Card → Terminal)
3) Data (Terminal → Card)
4) Status (Card → Terminal)

- Card sends ACK/INS (or error-status) after data length received
Smartcards: Further reading

- Smartcard handbook: http://www.wrankl.de/SCH/SCH.html
- Handbuch der Chipkarten (german): http://www.wrankl.de/HdC/HdC.html
- Phoenix reader – you can build your own
  - Several designs → use google
  - Replace MAX232 w/ FT232 or so for USB
  - Use 3.3V instead of 5V!
Purpose of the SIM

- User authentication
- Network authentication (3G)
- Data storage (phonebook, SMS, settings)
- Common platform for additional services
  → SIM Application Toolkit
SIM filesystem

- Access control
- Contains directories & files
  - identified by 16bit File-ID (FID)
  - MF (Master File) : root dir (FID: 3f00)
  - DF (Dedicated File) : directory
  - EF (Elementary File) : file
- Special EF types: record files
  - Fixed or variable length
  - Cyclic
    - Example: Phonebook
    - Example: Call History
SIM filesystem: important FIDs

- **DF_GSM**: Network related data  
  - **EF_IMSI**: IMSI  
  - **EF_Kc**: session key  
  - Etc.

- **DF_TELECOM**: Data for user  
  - **EF_SMS**: SMS storage  
  - **EF_ADN**: phonebook  
  - Etc.
SIM filesystem: a few notes

- SELECT instruction opens a file for access
- FIDs usually aren't unique across directories
  - Different EFs in different DFs may have same FID
    → SELECT needs to follow path of directories
  - Example: SELECT MF; SELECT DF_GSM; SELECT EF_IMSI

- There's no directory listing like "ls"
  - FIDs for GSM are published in the specs
  - Are there any hidden (non-specified) FIDs?
Tool: SIM_dump

- Phoenix only, no PC/SC yet
- Brute-force-approach on FIDs
  → find hidden files
- C-tool to dump files from SIMs - no USIMs yet
  - Quick, ugly hack. Stable?
  - But I tested it once!1
- Still want the code?
  → https://github.com/znuh/simdump
SIM instructions (1)

- 1 APDU can only transfer data to or from card
  - What if we need both?
  - GET_RESPONSE fetches the answer

<table>
<thead>
<tr>
<th>A0</th>
<th>C0</th>
<th>0</th>
<th>0</th>
<th>Len</th>
<th>Data</th>
<th>SW1</th>
<th>SW2</th>
</tr>
</thead>
</table>

- How to select a FID?
  - SELECT

<table>
<thead>
<tr>
<th>A0</th>
<th>A4</th>
<th>0</th>
<th>0</th>
<th>2</th>
<th>FID</th>
<th>SW1</th>
<th>SW2</th>
</tr>
</thead>
</table>

- Read/update/etc. File
  - Would bloat this talk too much
- **RUN GSM**
  - User authentication
  - Session key (Ciphering Key (Kc)) generation

  **Answer via GET RESPONSE:**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>Random value from net</th>
<th>SW1</th>
<th>SW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>88</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>SW1</td>
<td>SW2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>SRES(4) + Kc(8)</th>
<th>SW1</th>
<th>SW2</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0</td>
<td>C0</td>
<td>0</td>
<td>0</td>
<td>12</td>
<td>SW1</td>
<td>SW2</td>
<td></td>
</tr>
</tbody>
</table>

- SRES: Authentication response
- Kc: Ciphering key
USIMs

- Backwards compatible
- Multiple Applications on a single card
  - EF_DIR (2F00) has a list of installed applications
  - Application ID (AID) selection
- Other CLA for USIM – 00 instead of A0
- Mutual (network & user) authentication
  - AUTHENTICATE instruction
Tool: SIMtrace

- Hardware sniffer for phone ↔ SIM
- With inject support! → MITM
- Made by the osmocom guys

- Cheap AND open
- Get it here at the camp
- There's a workshop
- See RadioVillage
Example:

APDU: (22):
```
00 00 00 09 6f 38 04 00
15 00 55 01 02 00 00
91 78
```

APDU: (16):
```
ff 3f ff ff 00 00 3f 03
00
91 78
```
The SIM Application Toolkit (SAT)

- Normal way: phone sends commands to SIM
- SAT: Commands from SIM to phone

Why?
- Additional phone-independent services

How?
- Terminal is master → polling
- New instructions, status word (91xx instead of 9000)
- SAT Commands part of GSM/3G spec
- Most stuff is done in baseband!
- App-MCU mostly for user-interaction
SAT instructions

- Terminal profile (data: phone → SIM)
  - Notify SIM about SAT-features supported by phone

- Fetch (data: SIM → phone)
  - Fetch SAT commands from SIM
SAT instructions (2)

- Terminal response (data: phone → SIM)
  
<table>
<thead>
<tr>
<th>A0</th>
<th>14</th>
<th>0</th>
<th>0</th>
<th>Len</th>
<th>Data</th>
<th>SW1</th>
<th>SW2</th>
</tr>
</thead>
</table>

- Answer to SAT-commands from previous Fetch

- Envelope (data: phone → SIM)
  
<table>
<thead>
<tr>
<th>A0</th>
<th>C2</th>
<th>0</th>
<th>0</th>
<th>Len</th>
<th>Data</th>
<th>SW1</th>
<th>SW2</th>
</tr>
</thead>
</table>

- Notify SIM about some event
- Example: menu selection, SMS received, call setup
SAT commands

- Transmitted in data-part of Fetch-instruction
- Some interesting features:
  - Set up call & call control
  - Send short message
  - Run AT command
  - Data channel stuff
  - Provide local information (cell IDs, signal levels)
  - Geographical Location Request (yes, that's GPS)
SAT command encoding

Commands + parameters are TLV encoded:

- Proactive SIM tag
  - Command details tag
    - Actual command
  - Other Parameters ...
    - ...

- Mandatory and optional parameters
- **Alpha identifier** tag controls notification of user
### SAT example: send SMS

#### Fetch data:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>d0 1e</td>
<td>Proactive SIM Tag</td>
</tr>
<tr>
<td></td>
<td><strong>01 03</strong> Command details Tag</td>
</tr>
<tr>
<td></td>
<td><strong>01</strong> Command number</td>
</tr>
<tr>
<td></td>
<td><strong>13</strong> Type of command: Send short message</td>
</tr>
<tr>
<td></td>
<td><strong>01</strong> Command qualifier: packing required</td>
</tr>
<tr>
<td></td>
<td><strong>02 02</strong> Device identities Tag</td>
</tr>
<tr>
<td></td>
<td><strong>81</strong> Source device identity: SIM</td>
</tr>
<tr>
<td></td>
<td><strong>83</strong> Destination device id: Network</td>
</tr>
<tr>
<td></td>
<td><strong>05 00</strong> Alpha identifier Tag</td>
</tr>
<tr>
<td></td>
<td><strong>0b 11</strong> SMS TPDU Tag</td>
</tr>
<tr>
<td></td>
<td><strong>01</strong> SMS SUBMIT</td>
</tr>
<tr>
<td></td>
<td><strong>00</strong> Message reference</td>
</tr>
</tbody>
</table>

...
Over-the-air update

- SMS-PP download via Envelope instruction
- Like “silent SMS”, but sent to SIM card
- Usually, there's crypto (DES/RSA?) for this
- Haven't had a closer look at this
- A virgin SIM might be a good start for this
Further reading (SIM-related)

- **ETSI TS 102 221**: SIM instructions, etc.
- **3GPP TS 31.102**: SIM files, procedures
- **3GPP TS 31.111**: SIM application toolkit
- There's a lot more

- Useful tool for SMS de/encoding: **PDUspy**
- **Session-logs** from real (U)SIMs
Summary

- SIM features:
  - phone control via SAT (calls, SMS, data, etc.)
  - location tracking
  - remote updates
- You don't know what the SIM firmware does
- With most mobile phones you cannot
  - disable the SAT
  - or see what the SAT actually does
- 3GPP SAT spec is growing (new features!)
So what can be done?

- Watch 3GPP specs for new features
- Patches for phones (Problem: → baseband?)
  - SAT filter
  - SAT monitoring
- Which SAT-features do phones support? → SIMtrace
- Which SAT-features are actually used?
  - Operator specific
  - Needs long-term monitoring
•Thanks for your attention!