SIM Cards

Smart Card Basics

Terminology

**SIM**  Subscriber Identity Module

**USIM**  Universal Subscriber Identity Module

**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 (European Telecommunications 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

- **MF** ('3F00')
  - **DF$_{GSM}$** ('7F20')
  - **DF$_{GSLECTCOM}$** ('7F10')
  - **DF$_{IS-41}$** ('7F22')
  - **DF$_{FP-CTS}$** ('7F23')
    - **EF$_{ADN}$** ('6F3A')
    - **EF$_{FDN}$** ('6F3B')
    - **EF$_{SMS}$** ('6F3C')
    - **EF$_{CCP}$** ('6F3D')
    - **EF$_{MSCSidan}$** ('6F40')
  - **EF$_{IME}$** ('4F20')
  - **DF$_{GRAPHICS}$** ('5F50')
    - **DF$_{GLOBIST}$** ('5F31')
    - **DF$_{IHO}$** ('5F32')
    - **DF$_{ACCSD}$** ('5F33')
  - **EF$_{BCP}$** ('6F4F')
  - **EF$_{CM1}$** ('6F58')
  - **EF$_{BDM}$** ('6F4D')
  - **EF$_{EXT2}$** ('6F4B')
  - **EF$_{EXT3}$** ('6F4C')
  - **EF$_{EXT4}$** ('6F4E')
  - **EF$_{MSCSIDN}$** ('6F49')
  - **EF$_{L24D}$** ('6F44')
  - **EF$_{SMSS}$** ('6F43')
  - **EF$_{SMSP}$** ('5F42')
  - **EF$_{IP}$** ('4F20')
  - **EF$_{ICCID}$** ('2F2E')
  - **EF$_{ELP}$** ('2F05')

See GSM 11.19
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)
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
SIM Application Toolkit (SAT)

- Ability for card to run applications that have UI on the phone
  - Display menu items on-screen
  - Get user input from keypad/touch-screen
- Original Version Described in TS 11.14 and 11.11
SAT – Proactive SIM

The *Proactive SIM* features

- Sending a short message
- Setting up a voice call
- Playback of a tone in earpiece
- Providing location information from ME to SIM
- Have ME execute timers on behalf of SIM
- Sending DTMF to network
- Running an AT command received from SIM, sending result back to SIM
- Ask ME to launch browser to SIM-provided URL
SAT – Call and SMS Control

- ME passes MO call setup attempts to SIM for approval
- SIM can then
  - approve or decline the MO call
  - modify the call details such as phone number
  - replace the call with USSD message
- ME passes USSD requests similar to Call Control
- Similar mechanism exists for all MO SMS
The SIM can inquire the ME about:

- MCC / MNC / LAC / Cell ID
- IMEI of ME
- Network Measurement Results
- BCCH channel list
- Date, Time, Timezone
- ME language setting
- Timing Advance
SAT – Event download

The SIM is notified by ME about certain events such as:
- Call Connected / Disconnected
- Location Status (Location Area change)
- User activity (keyboard input)
- Idle screen available
- Browser termination
SAT - Data download

- Enables Operator to exchange arbitrary data with the SIM
- Could be RFM (Remote File Management)
  - Read or modify phone book entries
  - Even change the IMSI of the SIM (!)
- In case of Java Card, can be download of card applets
  - Applets are stored permanently on SIM
  - Can later use SAT procedures to interact with ME
  - TS 03.19 specifies Java API to access SAT from Java RE
SAT - Data download

SAT Data Download can happen via
- via SMS or Cell Broadcast
  - Uses TS 03.40 TP-PID *SIM DATA Download*
  - ME forwards such SMS to the SIM in **ENVELOPE APDU**
  - Response from SIM is sent back as MO-SMS or DELIVERY REPORT
- via BIP (Bearer Independent Protocol)
  - Dedicated CSD call between network and SIM
  - GPRS session between network and SIM
SAT - Data download
Data download security

- GSM TS 03.48 specifies secure messaging for data download
- Includes replay protection
- Supports DES and 3DES
- SMS chaining for long commands / large data
SIM card abuse by hostile operator

- Even if the phone might be considered trusted, the SIM card is owned and controlled by the operator.
- Using SAT features, the operator can control many aspects of the phone.
- Examples:
  - Remotely reading address book / stored SMS
  - Monitor user behavior (browser termination, idle screen, ...)
  - Ask phone to establish packet data session
SIM card re-programming by attacker

- If the SIM is not properly secured (auth + encryption keys, ...) a third party attacker can send SAT envelope SMS to the card and install resident Java applets.
- The attacker can then:
  - Obtain detailed location information and send it via SMS.
  - Intercept/log outgoing calls.
  - Sending copies of incoming + outgoing SMS elsewhere.
- Even using SIM card channel to exploit baseband stack is feasible.
SIM card proxy / MITM by attacker

As soon as an attacker has temporary physical access to a phone, he can

- Insert a proxy-SIM between real SIM and phone
- Do everything a Java applet could do, but even with a securely configured SIM as he does not modify the existing SIM
- Sniff current Kc and send it out e.g. via SMS or even UDP/TCP packets over GPRS
- ... by only using standard interfaces that are common among all phones (as opposed to baseband software hacking which is very model-specific)

Most users would never notice this as they rarely check their SIM slot
Defending against SIM based attacks

- SIM cards are Operator issued, Ki is on the SIM
  - SIM card can thus not be replaced, but original SIM must be used
- Configure telephone to not store contacts or SMS on SIM
- Communication between SIM and ME is not encrypted/authenticated
- Solution: Proxy SIM between SIM and ME to break STK / OTA
  - Filter all STK/OTA/Proactive commands like ENVELOPE
  - Indicate lack of STK support to ME (EF.Phase)
Proxy SIM with firewall

- There are no known commercial products that implement STK/OTA filtering
- But there are a number of shim SIM cards that are plugged between SIM and SIM slot
- Most of them are used for SIM unlocking modern phones
- Some vendors produce freely (re)programmable proxy SIMs:

**Figure:** Bladox TurboSIM (AVR) and RebelSIM II (8051)
Analyzing SIM toolkit applications is hard

- 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 smard card interfaces is actually very simple
Introducing Osmocom SIMtrace

- Osmocom SIMtrace is a passive (U)SIM-ME communication sniffer
- Insert SIM adapter cable into actual phone
- Insert (U)SIM into SIMtrace hardware
- SIMtrace hardware provides USB interface to host PC
- simtrace host PC program encapsulates APDU in GSMTAP
- GSMTAP is sent via UDP to localhost
- wireshark dissector for GSM TS 11.11 decodes APDUs
Osmocom SIMtrace Principle

Phone

Flexi-PCB cable

SIMtrace hardware

USB cable

SIM

PC
Osmocom SIMtrace Hardware
Osmocom SIMtrace Hardware

- Hardware is based around AT91SAM7S controller
- SAM7S Offers two ISO 7816-3 compatible USARTs
- USARTs can be clock master (SIM reader) or slave (SIM card)
- Open Source Firmware on SAM7S implementing APDU sniffing
- Auto-bauding depending CLK signal, PPS supported
- Schematics / layout is open source (CC-BY-SA)
- Assembled + tested kits can be bought from [http://shop.sysmocom.de/](http://shop.sysmocom.de/)
### wireshark decoding

#### User Datagram Protocol, Src Port: 52294 (52294), Dst Port: gsmtap (4729)

<table>
<thead>
<tr>
<th>No.</th>
<th>Time</th>
<th>Source</th>
<th>Destination</th>
<th>Protocol</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1.788053</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>GSMTAP</td>
<td>GSM SELECT EF.IMSI</td>
</tr>
<tr>
<td>13</td>
<td>1.788078</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>GSMTAP</td>
<td>GSM GET RESPONSE</td>
</tr>
<tr>
<td>14</td>
<td>1.788099</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>GSMTAP</td>
<td>GSM SELECT EF.SST</td>
</tr>
<tr>
<td>15</td>
<td>2.063939</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>GSMTAP</td>
<td>GSM GET RESPONSE</td>
</tr>
<tr>
<td>16</td>
<td>2.063982</td>
<td>127.0.0.1</td>
<td>127.0.0.1</td>
<td>GSMTAP</td>
<td>GSM READ BINARY Offset=0</td>
</tr>
</tbody>
</table>

#### GSM SIM 11.11

- **Class:** GSM (0xc0)
- **Instruction:** GET RESPONSE (0xc0)
- **Parameter 1:** 0x00
- **Parameter 2:** 0x00
- **Length (Parameter 3):** 0x00

**APDU Payload:** 000000000f0704000150015010200000

**Status Word:** Normal ending of command with info from proactive SIM

#### ISO 7816-4 APDU Data Payload (iso...)

- Packets: 445 Displayed: 445 Marked: 0 Loa... Profile: Default
SIMtrace TODO

SIMtrace hardware is capable, but no software yet for:

- perform MITM (APDU filtering)
- full software SIM card emulation
- PC/SC compatible smart card reader
- 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!