Master SEMS

# Lab1 : Interaction with the smat card using ISO 7816-3 and ISO 7816-4

**Samia BOUZEFRANE**
**http://cedric.cnam.fr/~bouzefra/pfsem10-11.html**

**The example of the SIM card**

## Part 1: Basic features

### 1. The communication model of the smart cards (see Figure 1)

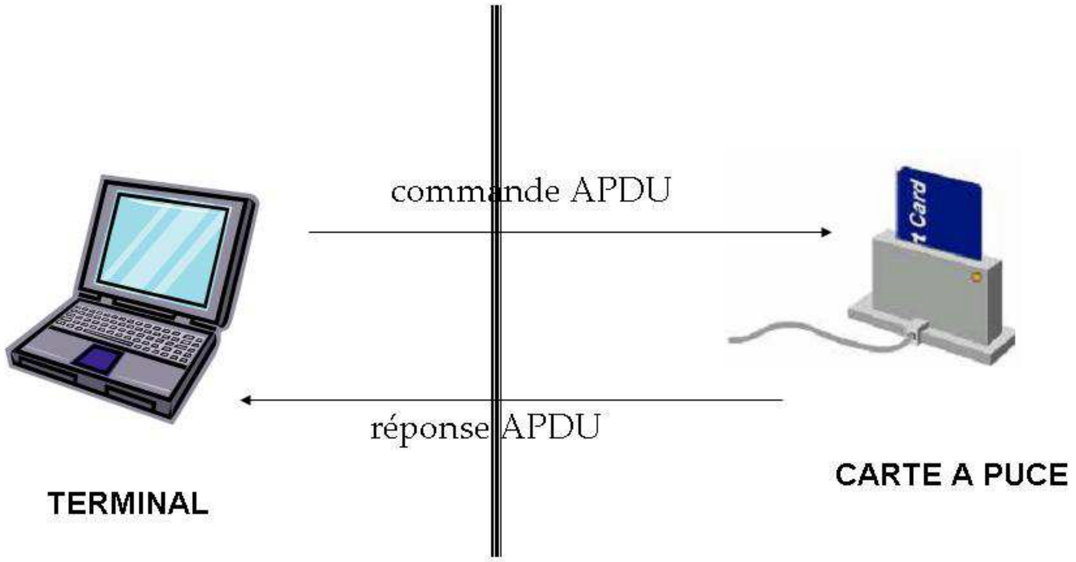

**Figure 1 : The communication model of the smart cards**

### 2. Format of the APDU commands

| APDU Command | | | | | | |
|---|---|---|---|---|---|---|
| Mandatory head | | | | Optional body | | |
| CLA | INS | P1 | P2 | Lc | Data field | Le |

- CLA (1 byte): instruction class --- dedicated for an application domain
- INS (1 byte): defines the instruction of the command
- P1 (1 byte) and P2 (1 byte): the parameters of the instruction
- Lc (1 byte): data length
- With Le=0, - if a writing command => no useful data
- - if reading command => the command must return 256 bytes of data
- Data field (bytes whose length is the Lc value): a sequence of bytes.

### 3. Format of the APDU responses

| APDU Response | | |
|---|---|---|
| Optional body | Mandatory part | |
| Data field | SW1 | SW2 |

- Data field (with a variable length): byte sequence
- SW1 (1 byte) and SW2 (1 byte): Status words sent by the card.

Status word values

| | |
|---|---|
| 0x6E 0x00 | CLA error |
| 0x6D 0x00 | INS error |
| 0x6B 0x00 | P1, P2 error |
| 0x67 0x00 | LEN error |
| 0x98 0x04 | Bad PIN |
| 0x98 0x08 | Unauthorized Access |
| 0x98 0x40 | Card blocked |

## 4. Examples of cards

| Fields of APDU command | Values |
|---|---|
| CLA | BC = french credit cards, vitale cards,<br>A0 = SIM cards |
| INS | 20 =PIN code verification,<br>B0 = Binary read<br>B2 = Read record<br>D0 = Binary write<br>DC = Write record<br>A4 = Directory selection<br>C0 = get an anwer |
| P1, P2 | parameters |
| LEN | Length of the data sent by the command |
| ARG | contains LC bytes (PIN code to check) |

# Part II: Interaction with the SIM card

## 1. The objective

The objective of this Lab is to explore the file system of a SIM card. This work is inspired from the article of Pascal Urien, « La carte SIM ou la sécurité du GSM par la pratique » published in magazine MISC, hors-série, Cartes à puce, nov/déc. 2008.

The Lab is achieved :

- First, using a script that interacts with the card by sending the APDU commands.
- Second, using a Java program that runs on the terminal and interacts with the card.

## 2. Development Environment:

### Under Windows:

- If you use Windows XP, install the driver of the SIM card reader (GENERIC2KXP USB Smart Card Reader) using the Cdrom. Under Windows 7, the reader is detected automatically, no need to install the driver explicitly.
- Download the script *gscriptor* from the following link:

*http://www.springcard.com/download/find.php?file=gscriptor*

gscriptor : is Perl script allowing to send commands via the graphical interface of the tool.

## 3. Exercice :

The mobile phone as soon as it is turned on, selects the GSM directory, detects whether the PIN code is required, and provides the PIN code value via the VERIFY command. Then, the cell phone reads the EF-Phase (FID = 6FAE) that contains the functional version number of the card. After that, the phone can read or write different files.

**Achieve the following commands using gscriptor and the table of commands given below.**

- Select GSM directory
- Provide the PIN code
- Read the IMSI
- Read TMSI and LAI
- Execute the authentification algorithm of GSM
- Update EF-Kc file

- Read the table of SIM services (EF-SIM-Service-Table)
- Read and write the SMS from the SIM card
- Read the agenda.

| **APDU COMMAND** | **INS** | **P1** | **P2** | **P3** |
|---|---|---|---|---|
| *SELECT* | A4 | 00 | 00 | 02 |
| *STATUS* | F2 | 00 | 00 | Length |
| *READ BINARY* | B0 | Offset high | Offset low | Length |
| *UPDATE BINARY* | D6 | Offset high | Offset low | Length |
| *READ RECORD* | B2 | Record number | Mode | Length |
| *UPDATE RECORD* | DC | Record number | Mode | Length |
| *SEEK* | A2 | 00 | Type/mode | Length |
| *INCREASE* | 32 | 00 | 00 | 03 |
| *VERIFY CHV* | 20 | 00 | CHV number | 08 |
| *CHANGE CHV* | 24 | 00 | CHV number | 10 |
| *DISABLE CHV* | 26 | 00 | 01 | 08 |
| *ENABLE CHV* | 28 | 00 | 01 | 08 |
| *UNBLOCK CHV* | 2C | 00 |  | 10 |
| *INVALIDATE* | 04 | 00 | 00 | 00 |
| *REHABILITATE* | 44 | 00 | 00 | 00 |
| *RUN GSM ALGORITHM* | 88 | 00 | 00 | 10 |
| *SLEEP* | FA | 00 | 00 | 00 |
| *GET RESPONSE* | C0 | 00 | 00 | Length |
| *TERMINAL PROFILE* | 10 | 00 | 00 | Length |
| *ENVELOPE* | C2 | 00 | 00 | Length |
| *FETCH* | 12 | 00 | 00 | Length |
| *TERMINAL RESPONSE* | 14 | 00 | 00 | Length |