le cnam

ED 1 : Interaction avec la carte à puce selon les protocoles ISO 7816-3 et ISO 7816-4

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

Exemple de la carte SIM

Partie I : Introduction

1. Introduction

Les normes ISO 7816 « Idenfication cards – Integrated circuit cards with contacts » ont été publiées par l'organisation internationale de normalisation (ISO, International Organisation for Standardisation). C'est le plus important standard définissant les caractéristiques des cartes à puce qui fonctionnent avec un contact électrique. Sachant que 15 normes sont proposées pour les cartes à contact, nous décrivons brièvement ici uniquement les 7 premières normes.

1.1 ISO 7816-1

Cette norme définit les caractéristiques physiques des cartes à puce à contact : la géométrie, la résistance, les contacts, etc.

1.2 ISO 7816-2

Cette norme spécifie le dimensionnement physique (extérieur) des contacts de la puce. Deux des huit contacts réservés à une utilisation future (RFU) sont redéfinis pour l'utilisation USB dans la norme ISO 7816-12.

1.3 ISO 7816-3

Cette norme définit l'interface électrique et les protocoles de transmission :
- les protocoles de transmission (TPDU, Transmission Protocol Data Unit) : T=0 : protocole orienté octet, T1 : protocole orienté paquet, T=14 : réservé pour les protocoles propriétaires.
- la sélection d'un type de protocole.
- la réponse à un reset (ATR, ou Answer To Reset en anglais) qui correspond aux données envoyées par la carte immédiatement après la mise sous tension.
- les signaux électriques, tels que le voltage, la fréquence d'horloge et la vitesse de communication.

1.4 ISO 7816-4

Cette norme vise à assurer l'interopérabilité des échanges. Elle définit les messages APDU (Application Protocol Data Units), par lesquels les cartes à puce communiquent avec le lecteur. Les échanges s'effectuent en mode client-serveur, le terminal ayant toujours l'initiative de communication.

1.5 ISO 7816-5

Cette norme définit le système de numérotation et les procédures d'enregistrement et d'attribution des identifiants des applications (AID, ou Application IDentifier). Un unique AID est associé à chaque application et à certains fichiers sur la carte. Ils sont représentés par des tableaux d'octets de taille allant de 5 à 16. Les cinq premiers octets représentent le numéro d'enregistrement du fournisseur d'application (RID, Registered Application Provider IDentifier en anglais) qui est

attribué par la Copenhagen Telephone Company Ltd ou l'ISO. Ils sont suivis par l'identifiant optionnel PIX (Proprietary Application Identifier eXtension) d'une longueur allant jusqu'à 11 octets.
L'identifiant RID est le même pour le paquetage et l'applet, mais le PIX doit être différent.

1.6 ISO 7816-6

Cette norme spécifie des éléments de données inter-industrie pour les échanges, tels que le numéro du porteur de carte, sa photo, sa langue, la date d'expiration, etc.

1.7 ISO 7816-7

Cette norme définit les commandes inter-industrie pour langage d'interrogation de carte structurée (SCQL).

2. ATR (Answer To Reset) défini dans l'ISO 7816-3

Dès que la carte est mise sous tension, elle envoie un message de réponse d'initialisation appelé ATR, il peut atteindre une taille maximale de 33 octets. Il indique à l'application cliente les paramètres nécessaires pour établir une communication avec elle. Il fournit un nombre varié de paramètres liés à la carte et au protocole de transmission utilisé :
– Le protocole de transport ;
– Taux de transmission des données ;
– Numéro de série de la puce …
Le premier octet noté TS ="3F" pour convention indirecte ou "3B" pour convention directe.

3. Echange de commandes avec le lecteur de carte à puce tel que défini dans l' ISO 7816-4

La communication entre l'hôte et la carte est half-duplex. Elle se fait à l'aide de paquets appelés APDU (Application Protocol Data Units) en respectant le protocole de l'ISO 7816-4. Un APDU contient une commande ou une réponse. Le mode Maître/Esclave est utilisé. Ainsi la carte joue un rôle passif et attend une commande APDU à partir de l'hôte. Elle exécute l'instruction spécifiée dans la commande et retourne une réponse APDU.

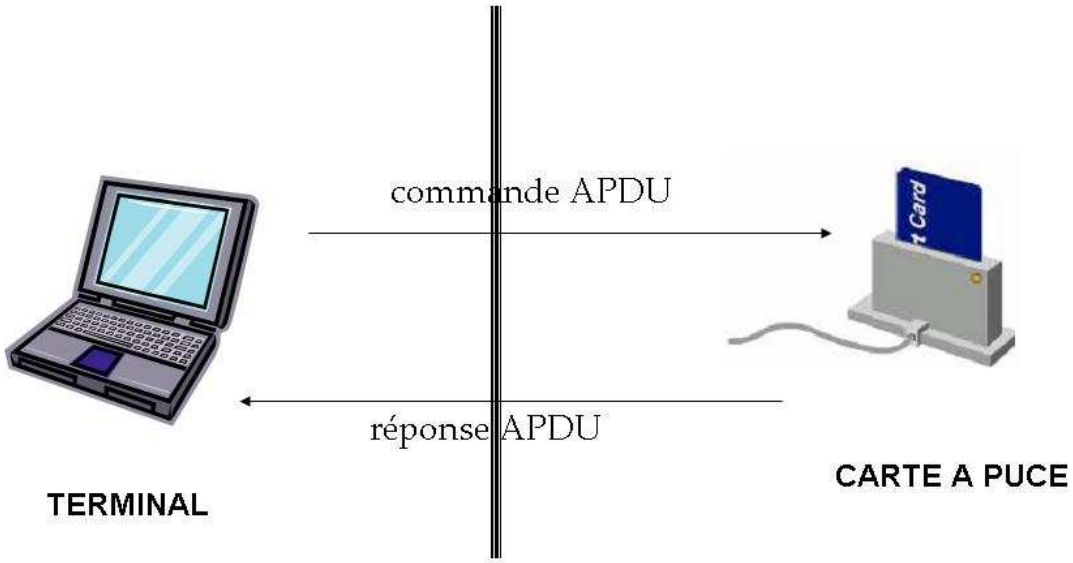

Figure 1 : Le modèle de communication de la carte à puce

3.1. Format des commandes APDU

Commande APDU						
Entête obligatoire				Corps optionnel		
CLA	INS	P1	P2	Lc	Data field	Le

- CLA (1 octet): Classe d'instructions : indique la structure et le format pour une catégorie de commandes et de réponses APDU.
- INS (1 octet): code d'instruction: spécifie l'instruction de la commande.
- P1 (1 octet) et P2 (1 octet): paramètres de l'instruction.
- Lc (1 octet): nombre d'octets présents dans le champ données de la commande.
- Data field (octets dont le nombre est égal à la valeur de Lc): une séquence d'octets dans le champ données de la commande.

3.2. Format des réponses APDU

Réponse APDU		
Corps optionnel	Partie obligatoire	
Data field	SW1	SW2

- Data field (longueur variable): une séquence d'octets reçus dans le champ données de la réponse.
- SW1 (1 octet) et SW2 (1 octet): Status words (Mots d'état) état de traitement par la carte.

4. Exemples de cartes

Le tableau suivant donne des exemples de commandes APDU utilisées dans le monde de la carte.

Champ de la commande	Valeurs
CLA	BC = cartes de crédit françaises, cartes vitales françaises, A0 = cartes SIM (téléphonie)
INS	20 = présentation du PIN, 40 = validation (ratification du code PIN) B0 = Lecture B2 = Lecture de record D0 = Ecriture DC = Ecriture de record A4 = Sélection du répertoire (directory) C0 = Demander une réponse (get an anwer)
P1, P2	paramètres contenant des adresses à lire
LEN	longueur prévue pour la réponse ou bien longueur de l'argument del'instruction.
ARG	contient LEN octets (octets à écrire, PIN à vérifier, etc.)

La réponse APDU sert à accuser réception la commande APDU envoyée par le terminal. Ainsi, la carte répond en envoyant le code instruction INS, suivi de données de longueur LEN en terminant par SW1 et SW2 (0x90 0x00 lorsque la commande s'est déroulée avec succès). En cas d'échec, seuls les champs SW1 et SW2 seront envoyés au terminal avec les codes d'erreur suivants :

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
...

Partie II: Interaction avec la carte SIM
Les premières commandes APDUs en utilisant un outil de script PC/SC

Exploration de la carte SIM

L'objectif de ce TP est d'explorer l'arborescence de fichiers d'une carte à puce. Nous avons choisi d'utiliser une carte SIM car son arborescence de fichiers est très riche. D'autre part, les informations à lire ne sont pas toutes confidentielles.

Le TP proposé ici s'inspire de l'article Pascal Urien, « La carte SIM ou la sécurité du GSM par la pratique » paru dans le magazine MISC, hors-série, Cartes à puce, nov/déc. 2008.

Ce TP sera réalisé :

- Dans un premier temps, à l'aide d'un script qui interroge la carte en envoyant des commandes APDUs.
- Dans un deuxième temps, à l'aide d'un programme Java qui tourne sur le terminal et qui interroge la carte.

Sous l'environnement Windows :

- Installer le driver du lecteur de carte SIM (GENERIC2KXP USB Smart Card Reader) à l'aide du CD d'installation.
- Télécharger le script à partir de l'adresse suivante :

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

gscriptor : script Perl servant à envoyer des commandes en utilisant une interface graphique.

Exercice :

Le téléphone portable, dès qu'il est mis sous tension, sélectionne le répertoire GSM, détecte si la présentation d'un code PIN est nécessaire, et présente cette valeur via la commande VERIFY. Ensuite, le portable lit le fichier EF-Phase (FID= 6FAE) qui contient le numéro de version fonctionnel de la carte. Après cela, le téléphone peut écrire ou lire différents fichiers. C'est ce que nous allons faire dans cet exercice.

Réaliser la suite des commandes suivantes en utilisant le tableau des commandes APDU donné ci-dessous.

- Sélection du répertoire GSM

- Présentation du code PIN
- Lecture de l'IMSI
- Lecture des paramètres TMSI et LAI
- Exécution de l'algorithme d'authentification du GSM
- Mise à jour du fichier EF-Kc
- Lecture de la table des services SIM (EF-SIM-Service-Table)
- Lecture et écriture des SMS dans la SIM
- Lecture de l'annuaire des numéros, ADN

COMMANDE	**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

Bibliographie :

- Pascal Urien, « *La carte SIM ou la sécurité du GSM par la pratique* », Magazine MISC hors-série carte à puce, découvrez leurs fonctionnalités et leurs limites, novembre/décembre 2008, pages 26-37.
- Technology for smart cards: architecture and programmer's guide, Zhiqun Chen, Addison Wesley, sept. 2000
- Les Cartes à puce: théorie et mise en œuvre, Christian Tavernier, 2ème édition, Ed. Dunod, 2007.
- Pierre Paradinas, Support de cours sur « Java Card », UV de Systèmes Enfouis et Embarqués,
- Samia Bouzefrane, cours cartes à puce, http://cedric.cnam.fr
- Wolfgang Rankl and Wolfgang Effing, « Smart Card Handbook», 3rd Edition, John Wiley & Sons Ed., 2003, ISBN 0-470-85668-8.