Introduction à la carte à puce

Jean-Pierre Tual

21/06/05

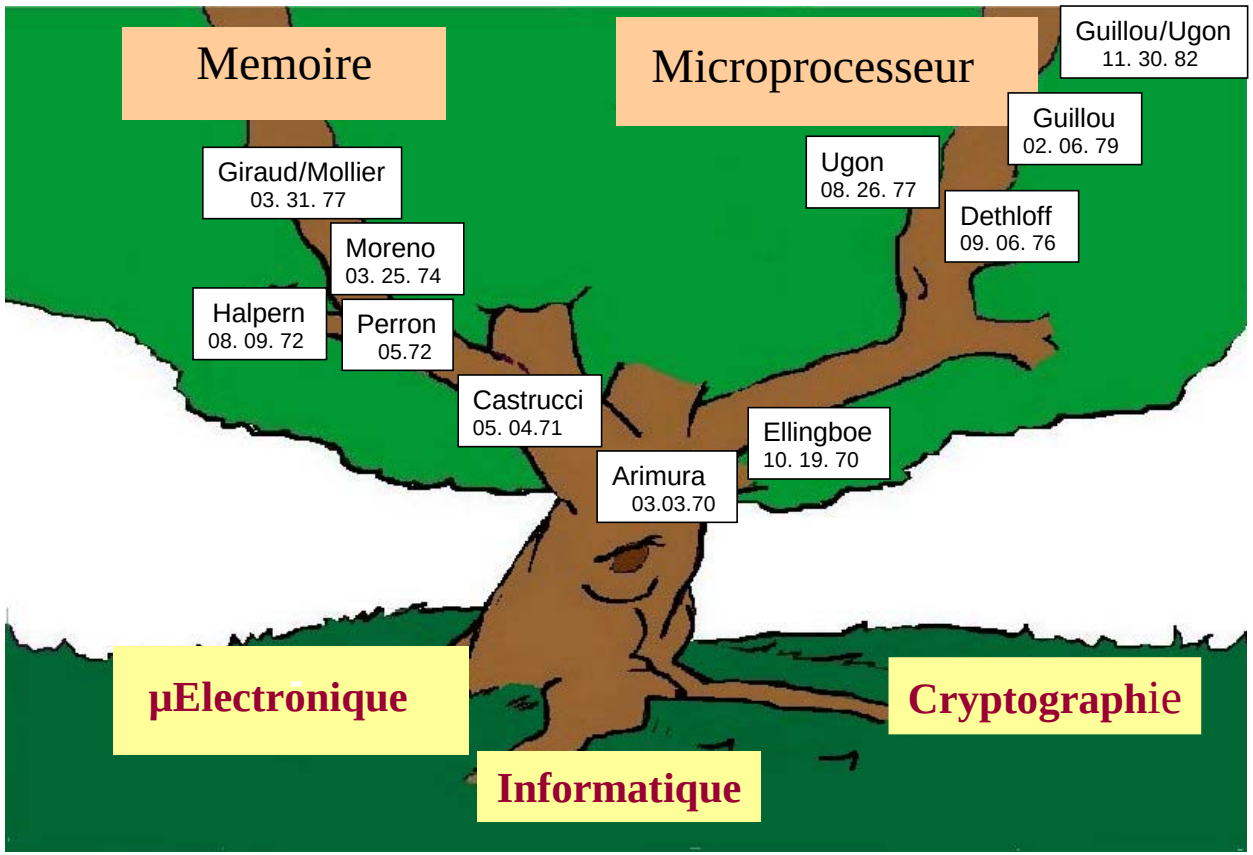
axalto
A Schlumberger company

Plan

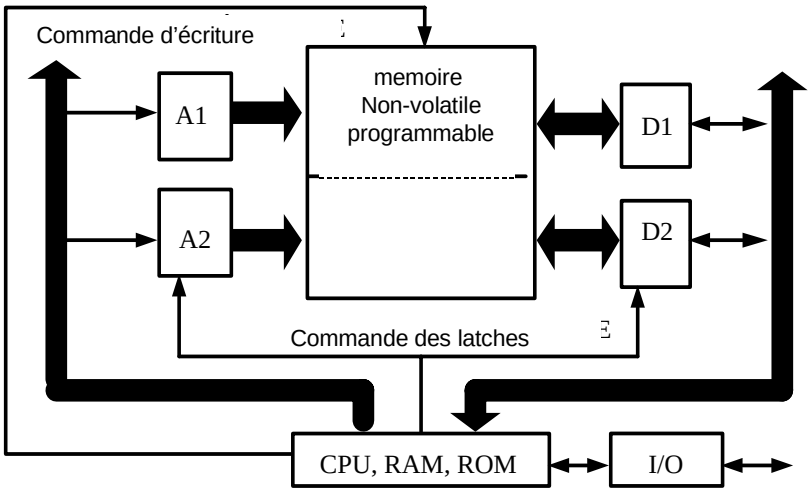
- Un peu d'Histoire
- Panorama du marché et grandes classes d'applications
- Technologies de base
- Architecture des cartes à puce
- Les standards fondamentaux
- Les normes ISO 7816
- La technologie JavaCard
- La sécurité des cartes à puce
- Exemples fondamentaux
  - Carte GSM/SIM
  - Systèmes de paiement EMV
- Futures tendances de la carte à puce

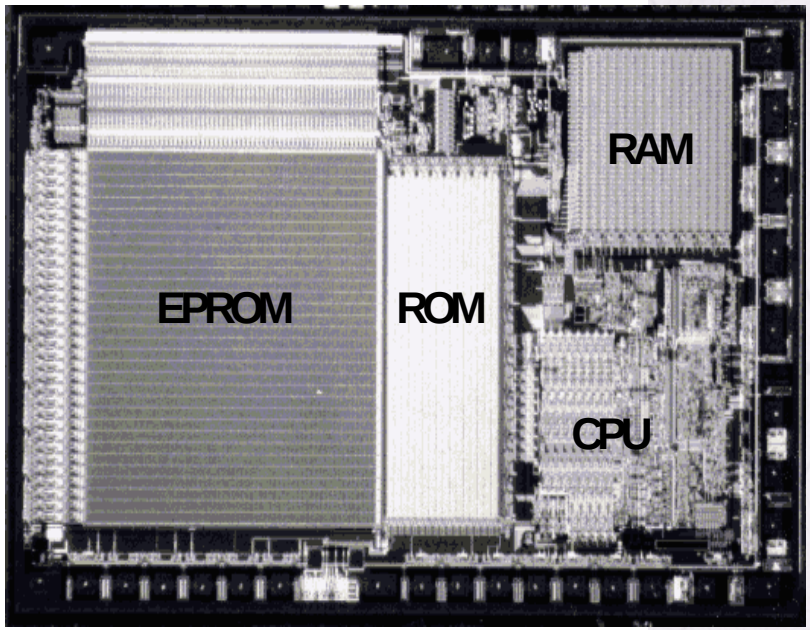
Cartes à puce: les pionniers

Brevets historiques



La base des cartes à puce: le SPOM





Octobre 1981: Premier SPOM industriel
RAM: 36B, ROM: 1,6 KB, EPROM: 1 KB
NMOS 3,5 µ- 42 KTransistors

Premières expérimentations

1979	Première carte à microprocesseur (Bull)
1980-1981	Première carte pour la PayTV (Bull/Philips)
1981-1982	Premières expérimentations de paiement 125 000 cartes bancaires (Bull/Philips/Schlumberger)
1983	Première « Télécarte » (Schlumberger)
1984	Première « Telekarte » (G&D)
1984	Première expérience bancaire (Bull)
1988	Première carte multi-application **(Bull)**
1988	Première carte d ’université (Bull)
1988	Première expérimentations bancaires
1989	Première carte « club fidélité »
1989	Première expérimentation bancaires
1989	Première carte GSM (Gemplus)

axalto

Quelques réalisations industrielles majeures

1985-2005

GIE-CB

Carte bancaire Française

49 M cartes en circulation en 2004

Pour 305 Beuros de transactions

1996-2005

GIE SESAM-Vitale

Carte Nationale de Santé

45 M cartes en circulation en 2004

1996-2005

Porte Monnaie Electronique

100 M de cartes Proton livrées dans 18 pays:

Suisse, Belgique,Pays-Bas, Suède, Malaisie

80 M Geldkarte en Allemagne, Luxembourg, Islande…

1992- 2005

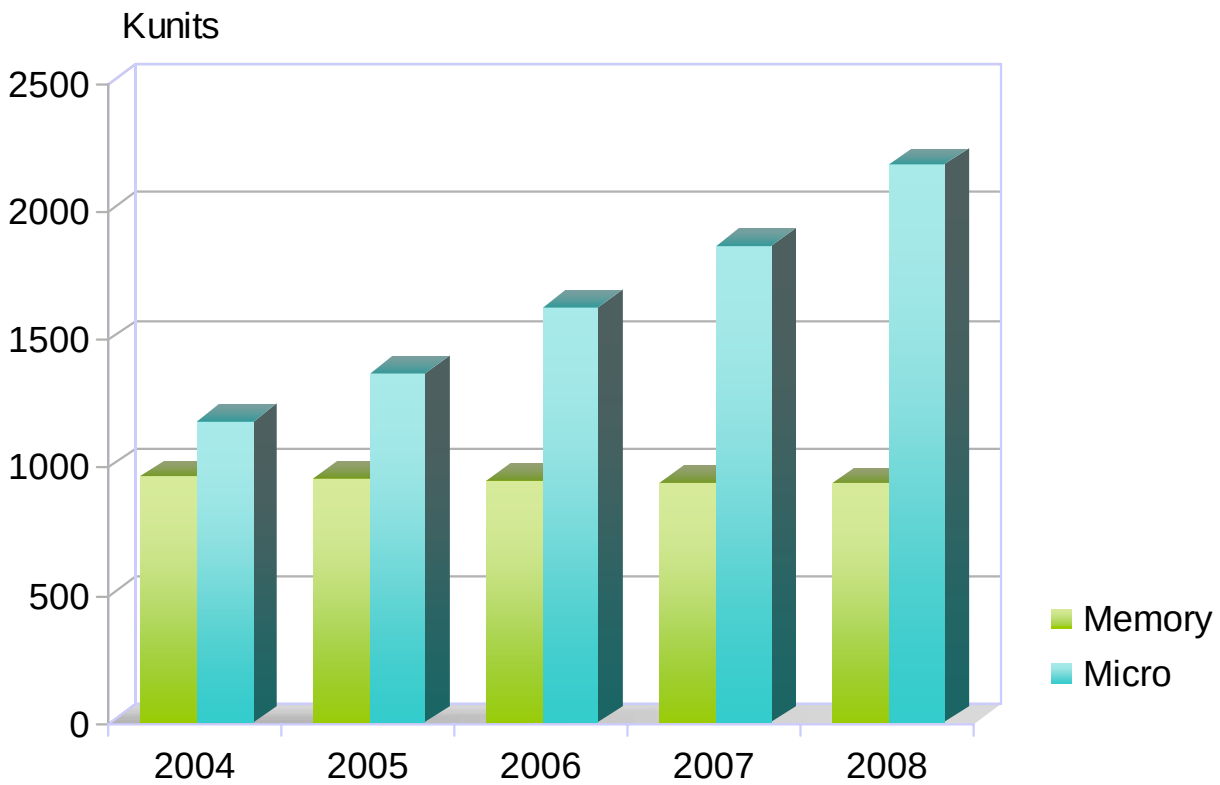
Téléphonie Mobile

Pénétration sans précédent

> 5 Md de cartes SIM vendues

axalto

Evolution du marché de la carte à puce



* Frost & Sullivan, 2004

Evolution du marché de la carte à puce

- En volume (Mu)

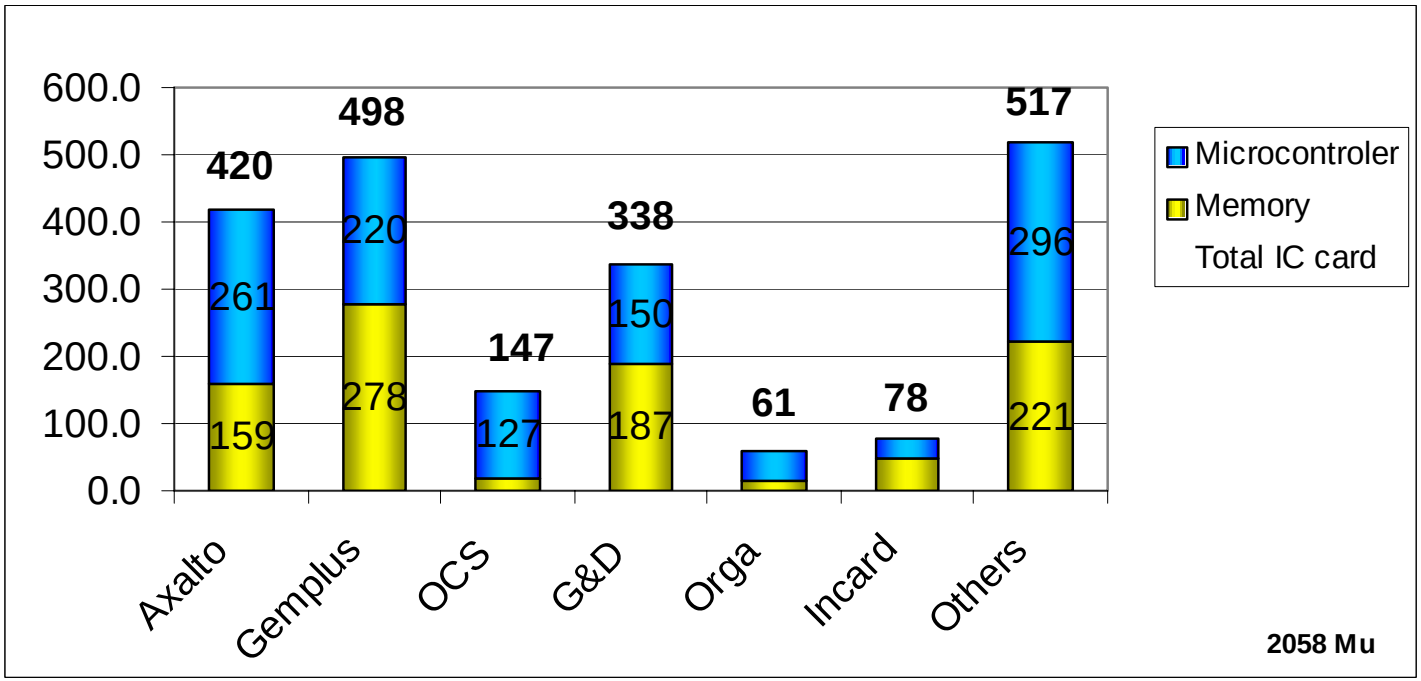
Year	2005	2006	2007	2008	2010	2012	CAGR 05/12
Bank	350	400	480	600	810	980	16%
Telco	1100	1250	1400	1600	1700	1800	7%
Pay TV	50	55	80	100	150	180	21%
Government	85	140	230	280	450	800	41%
Internet + IT security cards	15	20	40	80	140	160	46%
Transport	85	120	150	180	250	300	21%
Total	**1 685**	**1 985**	**2 380**	**2 840**	**3 500**	**4 220**	**14%**

- En valeur (Me)

Year	2005	2006	2007	2008	2010	2012	CAGR 05/12
Bank	850	1000	1050	1270	1550	1600	10%
Telco	2300	2400	2300	2450	2600	2700	2%
Pay TV	165	180	240	280	375	430	15%
Government	194	320	535	580	885	1420	36%
Internet + IT security cards	65	80	80	160	270	290	28%
Transport	125	155	155	175	230	245	11%
Total Value	**3 699**	**4 135**	**4 360**	**4 915**	**5 910**	**6 685**	**9%**

axalto

Marché et compétition en 2003



	TAM en Mu
Microproc.	1130
Memoire	928
Total Cartes	**2058**

Types de cartes (1/2)

- Carte à mémoire
  - Mémoire simple (sans processeur) accessible en lecture sans protection, mais l'écriture peut être rendue impossible
  - Programmation impossible
  - Carte « porte-jetons » pour applications de prépaiement (carte téléphonique)
- Carte à logique câblée
  - Mémoire accessible via des circuits préprogrammés et figés pour une application particulière
  - Carte « sécuritaire » pouvant effectuer des calculs figés (accès à un local …)

axalto

Types de cartes (2/2)

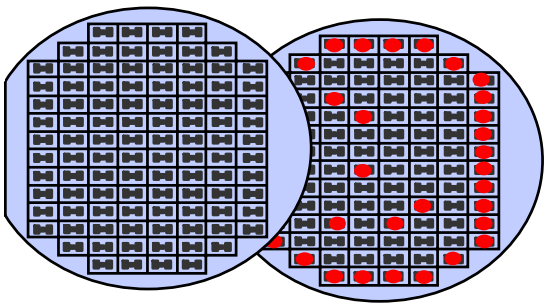
- Carte à puce ou SmartCard
  - Microcontrôleur encarté (processeur + mémoires)
  - Carte « programmable » pouvant effectuer tout type de traitements
  - Interface électrique par contacts ou via signaux RF

Fabrication des cartes à puce (1/2)

Assemblage
Micro-electronique

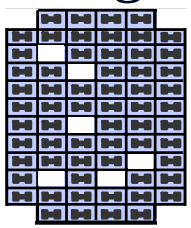
Wafer

Test

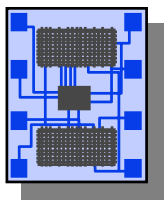


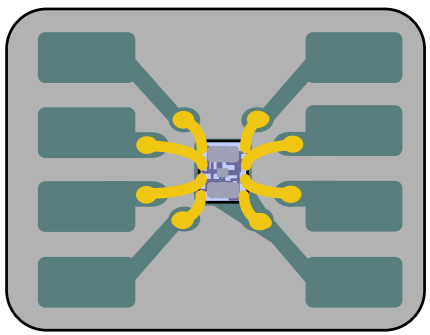
Motorola, Atmel
Texas Instruments
STMicroelectronics
Siemens, Hitachi

Sciage

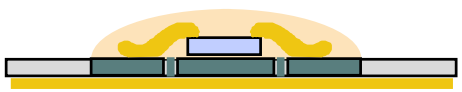


Puce

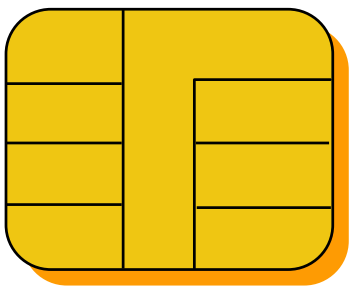




Assemblage



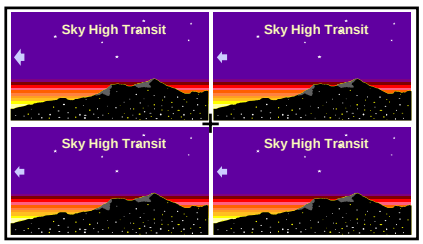
Protection



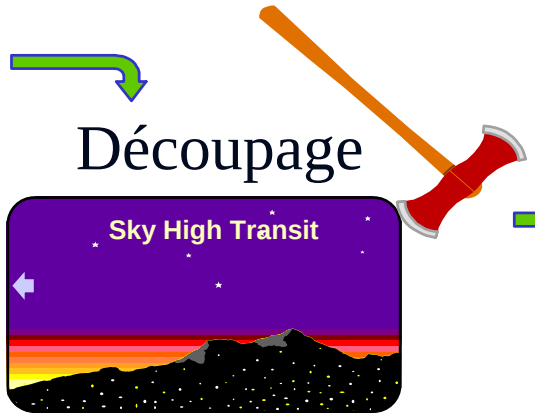
Module

Fabrication des cartes à puce (2/2)

Impression - Lamination

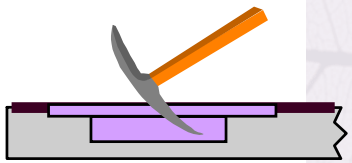


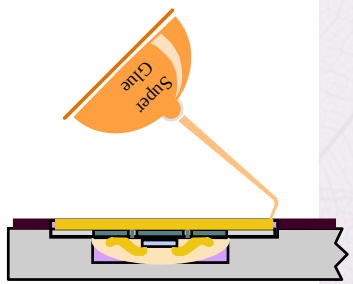
Découpage



Corps de Cartes

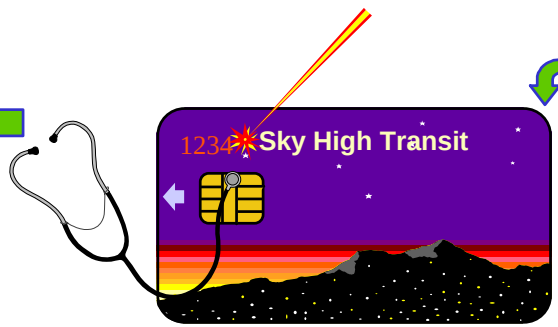
Cavité



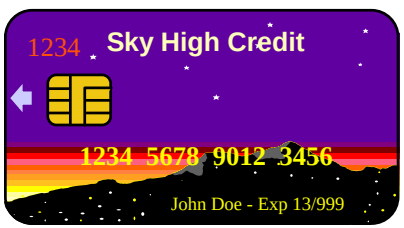


Insertion

Pré-détachage



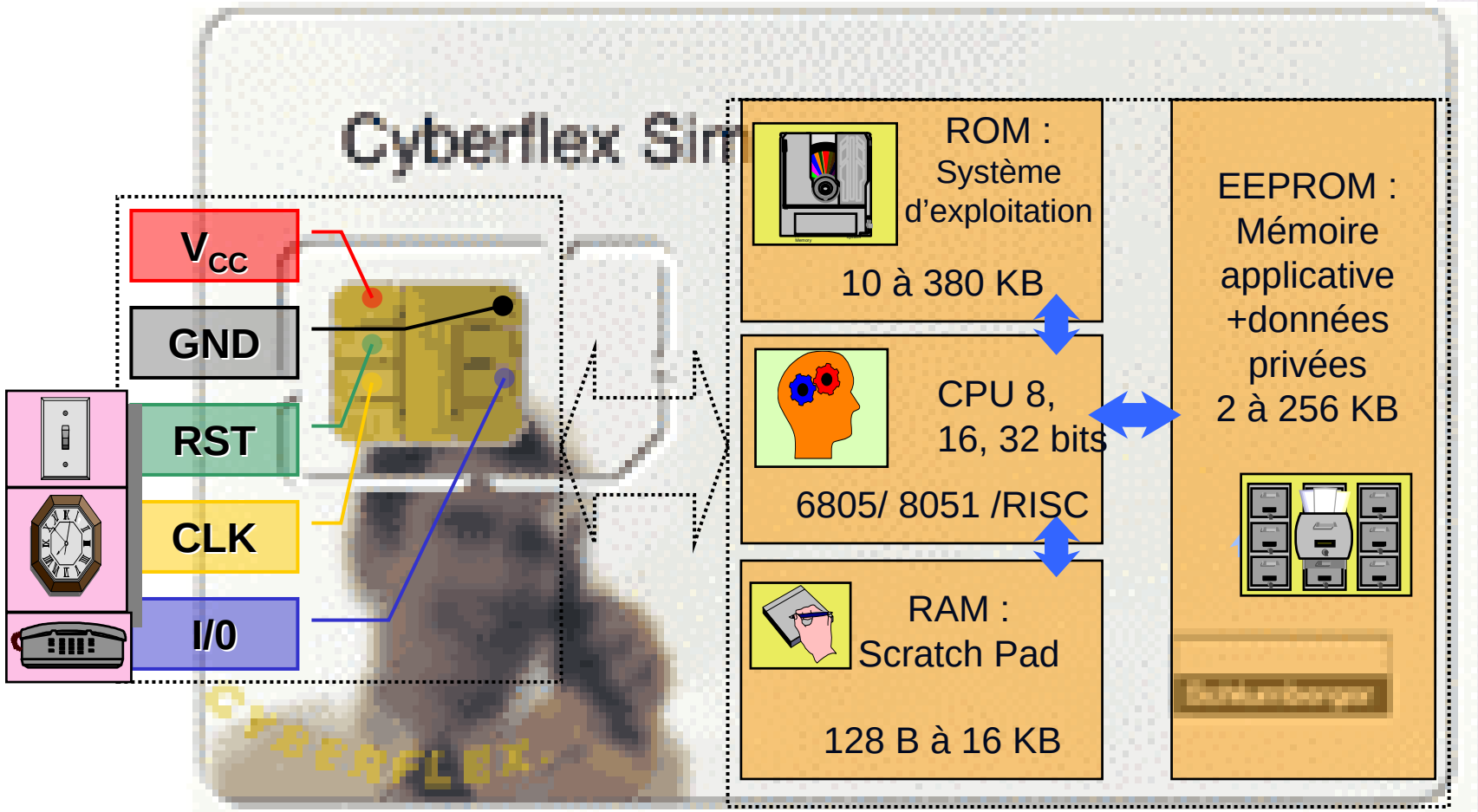
Test série



Personnalisation

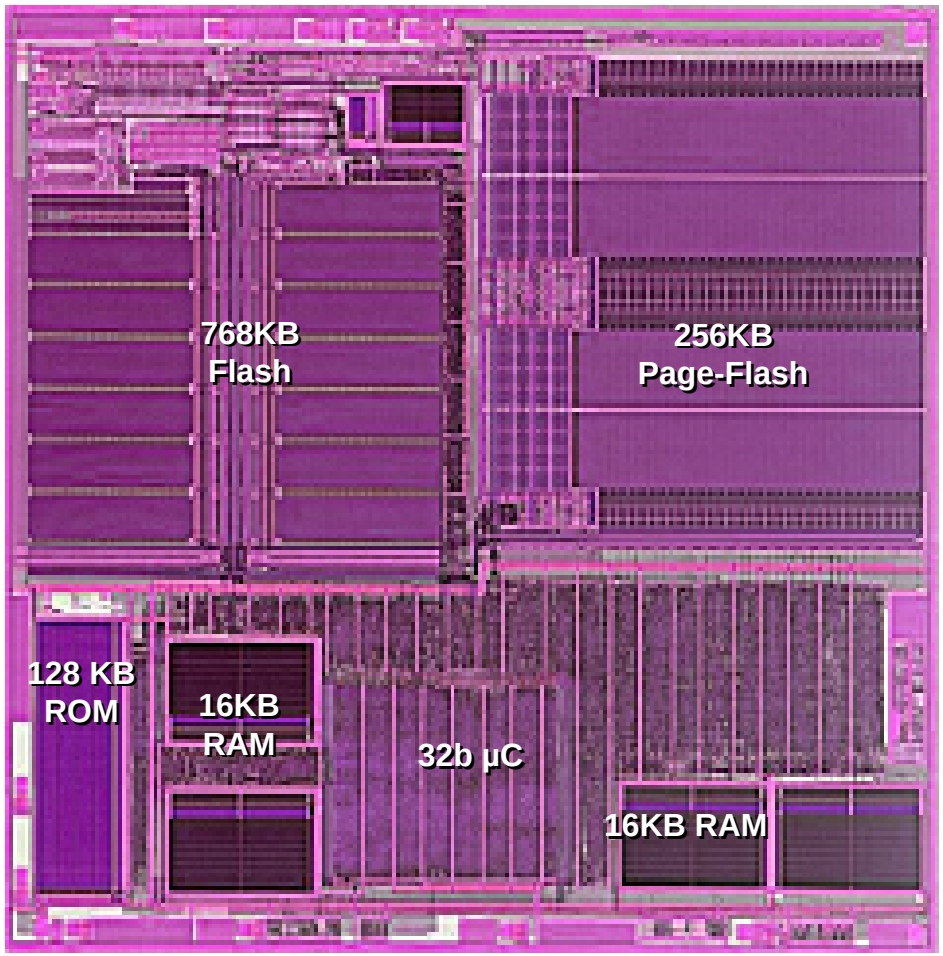
axalto

Architecture d'une carte à puce



Exemple de composant « haut de gamme »

- ST22 Flash de STMicroelectronics



L'OS de base ou « Masque »

Le Masque ("Hard Mask") est le système d'exploitation de la carte.

Il est généralement écrit en C ou en langage d'assemblage.

Il est stocké en ROM et ne peut donc être modifié durant la vie de la carte

Que fait le masque ?

- Gère les communications avec le monde extérieur
- Execute les commandes reçues via l'interface I/O
- Supervise l'exécution des programmes exécutables stockés dans la carte
- Gère le SGF et assure un accès sécurisé à l'ensemble des fichiers
- Assure les fonctions de cryptographie (DES, RSA, SHA, ECC,…)
- Sur les cartes les plus modernes, integre une JVM (Java Virtual Machine) pour exécuter des applets

La fonction principale de l' OS est une boucle qui attend l'arrivée de commandes externes.

A l'arrivée d'une commande, elle est exécutée, puis une réponse est émise vers l'extérieur et la boucle redémarre..

Les « Softmasks »

Un “Softt Mask” est une extension du masque.

Il est écrit en général en C, compilé, et lié aux librairies du Masque.

Il peut être chargé en EEPROM tant que la carte n’est pas bloquée.

Quand a t’on besoin d’un “Soft Mask” ?

- Quand une nouvelle fonctionnalité doit être ajoutée à une carte pour une application spécifique
- Pour les besoins de “bug fixing” au niveau du masque (cela arrive!)
- Quand l’exécution d’une commande du masque ne satisfait pas les besoins d’un client particulier
- Quand certaines spécifications de clients ne peuvent pas être implémentées en Java
- Quand une applet est trop lente => réécriture!

axalto

Principales normes applicables (1/2)

- **Normes de base**
  - Normes ISO/IEC 7816-x (x=1:16)
  - Normes ISO/IEC 14443 (A,B,C) et 15693 pour cartes sans contact
- **Normes génériques inter-domaines**
  - PC/SC: API's d'intégration de cartes à puce en environnement Windows
  - OCF: Environnement Java et API pour applications smart-cards
  - JavaCard (2:1 et 2:2) pour programmation des smart-cards
  - PKCS #15: Stockage de clefs cryptographiques
  - ISO/IEC 15408: dérivée des Critères Communs
- **Normes spécifiques du domaine bancaire**
  - EMV (96 et 2000): spécifications cartes/terminaux multi-applicatifs
  - Visa Open Platform: gestion (sécuritaire) de cartes multiapplicatives
  - CEPS: spécification d'interopérabilité pour les porte-monnaie électroniques
  - EN 1546: Spécifications génériques de porte-monnaie électroniques

axalto

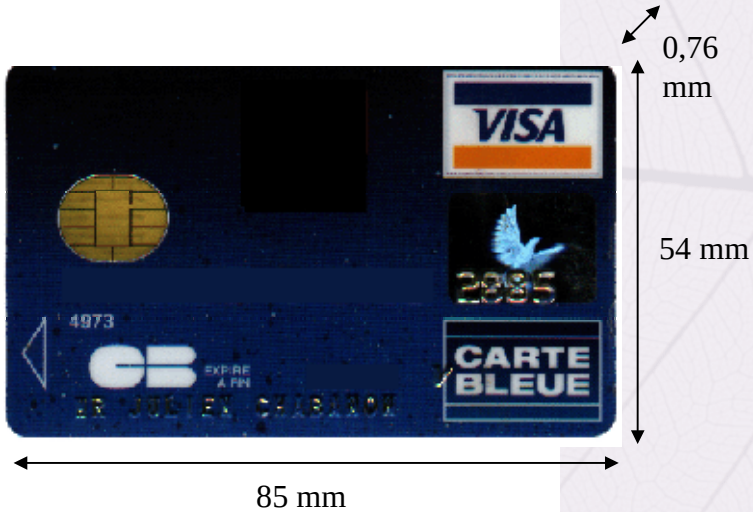
Principales normes applicables (2/2)

- **Téléphonie mobile**
  - **GSM 11-11: Spécification de l'interface SIM-ME (3GPP: TS 51.011)**
  - **GSM 11-14: Spécification « SIM Application Toolkit » pour l'interface SIM-ME**
  - **GSM 03.19: API JavaCard™ de programmation pour la carte SIM Phase 2**
  - **GSM 03.40: Réalisation de la fonction Short Message Service (SMS); mode Point to Point (PP)**
  - **GSM 03.48: Mécanismes de sécurité pour la carte SIM application toolkit Stage 2**
  - **ETSI TS 102 221: Spécifications de la carte UICC et de l'interface UICC terminal**
  - **3GPP: 31.101 V4.0.0, 31.102 V4.0.0 (Release 99)- cartes 3G (W-CDMA)**
  - **3GPP2-C00-1999-1206-1208: Spécification du module RUIM pout systèmes large-bande (systèmes CDMA 2000)**
- **Signature électronique**
  - **ETSI TS 101 333: formats de signature életronique**
  - **ETSI TS 101 808: spécification des politiques de gestion des "CA"**
  - **CEN/ISSS: directive européenne pour la signature électronique**

axalto

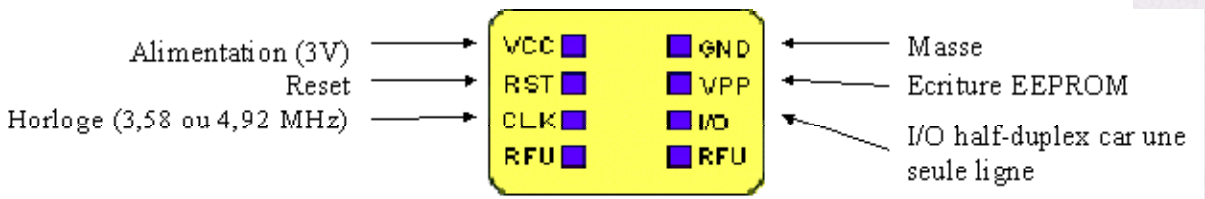
Norme ISO 7816-1

- Format carte de crédit
- Définition des contraintes physiques supportables (chaleur, humidité...)



Norme ISO 7816-2

- La puce
  - Seule interface de communication avec l'extérieur
  - Lecteur de cartes = CAD (Card Acceptance Device)
  - Surface ≤ 25 mm²
  - Épaisseur ≤ 0,3 mm
  - Composée de 8 contacts métalliques
  - I/O: Z état Haut- A état bas: quitte l'état haut seulement en transmission
  - VCC= 4,75 – 5,25 V jusqu'à 200 mA
  - CLK: cap in/out < 30 pF, temps de transition < max(0,5 µs, 9%T)
  - Activation: RST bas, Vcc haut, VPP repos, I/O Z, CLK entre 1 et 5 MHz
  - Désactivation: RST bas, CLK bas, VPP inactif, I/O état A, VCC bas



Norme ISO 7816-3

- Caractéristiques électriques
  - Fréquence d'horloge 1 - 5 Mhz
  - Vitesse des communications < 115200 bauds:

- Protocole de transmission
  - TPDU (Transmission Protocol Data Unit)
  - T=0 Protocole orienté octet
  - T=1 Protocole orienté paquet
  - **Protocoles de communication asynchrones et half-duplex**
  - T=2 Asynchrone, full duplex, orienté bloc => en cours de spécification

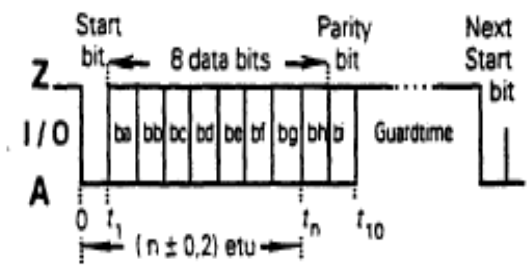
- Sélection du type de protocole
  - **PTS (Protocol Type Selection Réponse) au reset** :

- Réponse au reset :
  - **ATR (Answer To Reset)**

axalto

Normalisation et protocole de transport

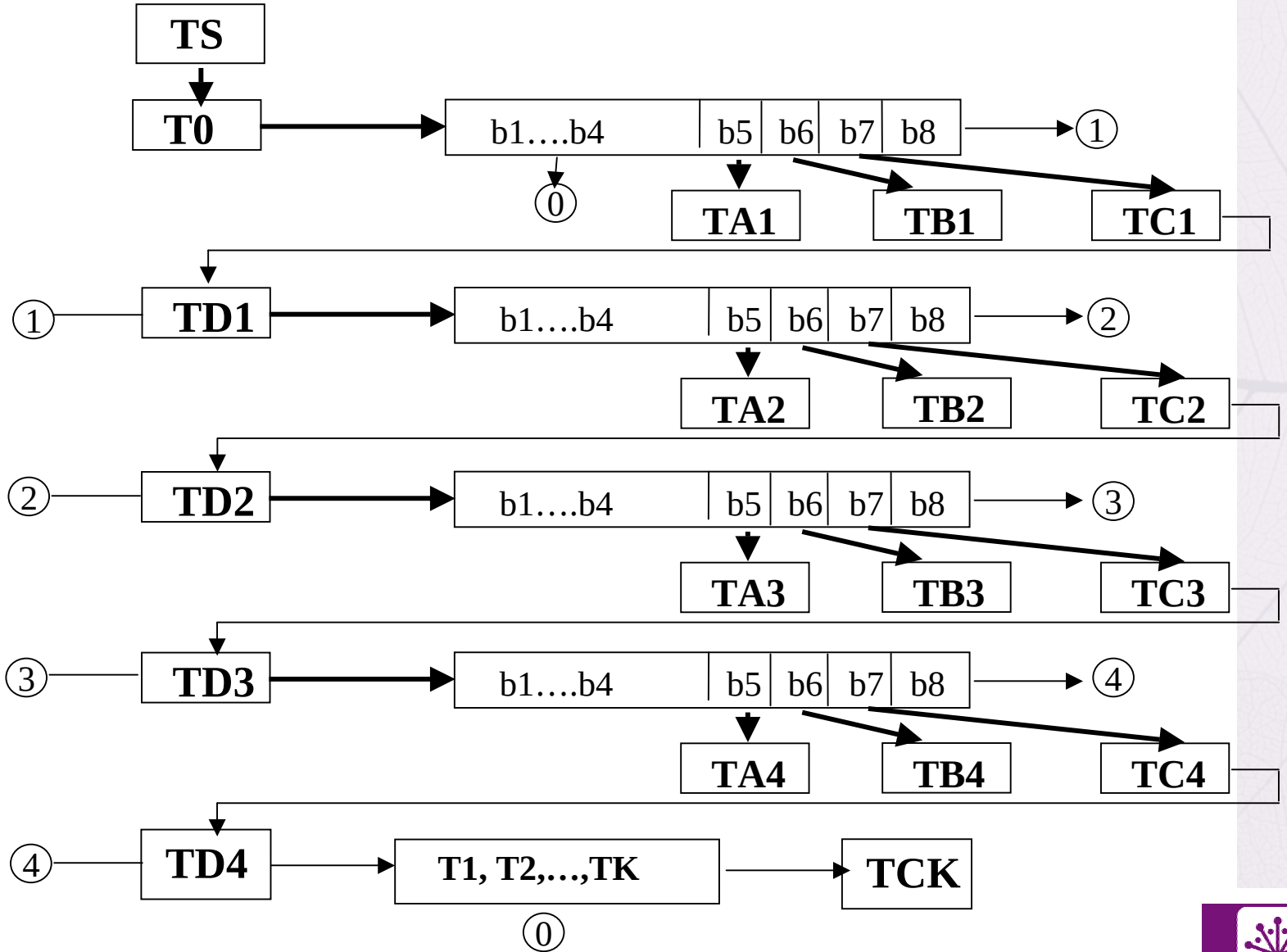
- Horloge
  - Freq: 3,579,545 Hz (valeur par défaut des lecteurs de cartes)
  - Débit par défaut: 9622 bauds
  - Durée d'un bit par défaut: 1 etu= 372 = F/D périodes d'Horloge
  - Horloge fournie par le lecteur compris entre 1 et 5 Mhz: F,D négociables
- Format des caractères



Un caractère comprend 1 bit start, 8 bits de données, un bit de parité (nombre de 1 pair parmi 8+1 bits). Le nombre de stop bit est égal à 2+N (N= ***Guardtime*** = 0 lors de la mise en tension).

- Conventions
  - Directes: A=0: 1er caractère= '3B'; b1:b8= A(ZZAZZZAA)Z
  - Inverses: A=1: 1er caractère= '3F'; b8:b1= A(ZZAAAAAA)Z

Structure générale d'une ATR



ATR: réponse carte à la RAZ (1/2)

- Doit intervenir entre 400 et 40,000 cycles d'horloge
- Série d'octets (b8,b7,b6,b5,b4,b3,b2,b1) dont les deux premiers, TS et T0 sont obligatoires
  - **TS:** Transmission. "3B" pour logique directe (positive), "3F" pour une logique inverse (négative)
  - **T0:** Présence d'octets d'interface (TAi:TDi; I=1:4) et historiques (T1:T15)
    - B8=1 présence TD1 B7=1 présence TC1
    - B6=1 présence TB1 B5=1 présence TA1
    - B4 B3 B2 B1, nombre d'octets historiques (0…15)
  - **TA1** – Valeur des paramètres d'ajustement d'etu F et D
  - **TB1** – Paramètres de programmation de l'EPROM (obsolète). "25"=> (Vpp =Vcc)
  - **TC1** – Nombre de bits stop excédentaires (N). La valeur par défaut est "00". "FF" fixe la valeur de N à 0 pour T=0 (2 stop bits) et –1 (1 stop bit) pour T=1.
  - **TD1** – Indique le type de protocole de transport mis en œuvre
    - B4 B3 B2 B1, numéro du protocole i= 0…15
    - B8 B7 B6 B5 indiquent respectivement la présence d'octets TDi+1, TCi+1, TBi+1, TAi+1 fournissant des informations complémentaire sur le protocole i.

ATR: réponse carte à la RAZ (2/2)

- **TA2**, indique la possibilité de négocier les paramètres de transfert (PTS)
  - B8=1 spécifie l'absence de cette option**.**

    Le numéro du protocole de négociation est renseigné par les bits B4 B3 B2 B1,
  - B5=1 notifie l'usage de paramètres implicites dans les Tai
  - B5=0 signifie que les paramètres sont explicites
- **TB2**, code la valeur de Vpp en dixième de volts.
- **TC2**, valeur d'un paramètre WI (0…255) permettant de calculer le temps d'attente maximum d'une réponse de la carte par le lecteur**;**
  - WT = WI . 960 . F/f secondes, soit 1 s pour f=3,58 Mz, F=372, $WI_{défaut}$=10 ("0A").
- **TAi+1, (i>2) indique la longueur max. du champ d'information reçu par la carte (IFSC)**
  - Défaut 32 B- Plage 1=> 254
- **TBi (i>2), fournit la valeur (0,15) des paramètres CWI et BWI utilisé dans leprotocole T=1 pour calculer:**
  - Le délai max entre deux caractères d'un même bloc:

    $CWT = (2^{CWI} + 11)$ etu (0,85 s avec CWI=13)
  - Le délai max de réponse de la carte

    $BWT =(2^{BWI} *960* 372 / f) + 11$ etu (1,6s BWI=4))
- **Tci (I>2)** définit le type de méthode pour la correction

  b1= 0 => LRC- b1=1 => CRC
- **TCK = XOR de l'ensemble des bytes de l'ATR jusqu'à TCK exclus**

Exemple d’ATR

- Cas du GSM
  - TS= ‘3B’ => Convention directe
  - T0= ’89’= ‘1000’ll’1001’ => TD1 suit + 9 caractères historiques
  - TD1= ’40’= ‘0010’ll’0000’ => TC2 suit et protocole T=0
  - TC2= ’14’= ‘0001’ll’1110’ => Waiting time 14 (1,4s)
  - T1…T9: ’47’ll’47’ll’32’ll’34’ll’4D’ll’35’ll’32’ll’38’ll’30’ => “GG24M5520” (donnée constructeur)

Protocoles

- **PPS**
  - Négociation de la vitesse de transfert conditionnée par absence de l'octet TA2 (Pas de TA2 => PPS avec F= 372 et D=1)
  - Echange d'au plus deux séries de 5 octets
    - Lecteur => carte: PPSS (FF), PPS0 (~TDi), PPS1 ~(~TA1), PPS2 (00), PPS3 (00), PCK= PPSS+PPS0+PPS1+PPS2
    - Carte => lecteur: répétition des 5 octets précédents si acceptation
- **Protocoles de transport**
  - T=0
    - Transmission série des octets (1 start, 8 bits, 1 parité, 2+N bits stop)
    - Erreur de parité: 0 logique sur la ligne de transmission durant 1 ou 2 etu.
  - T=1
    - Orienté bloc: (NAD,PCB, LEN), INF (0:254 octets), LRC (1B) ou CRC (2B)

      NAD: 3 bit adresse source, 3 bit adresse destination

      PCB:

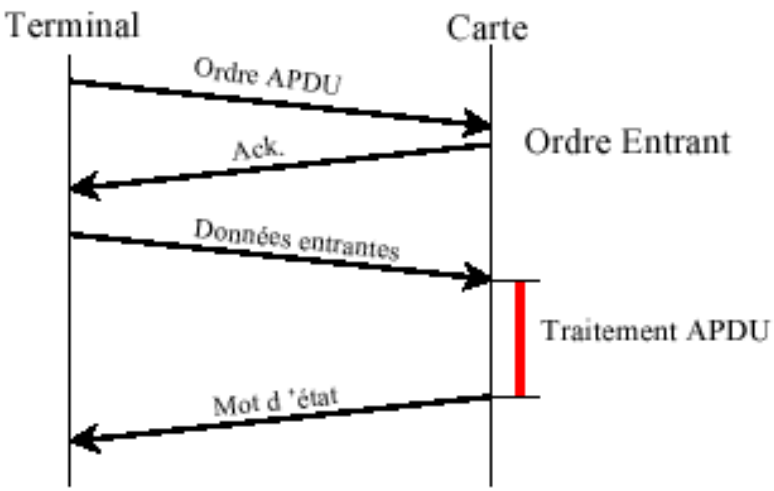
      I(#bloc, more): blocs numérotés modulo 2, more=1 => pas dernier bloc

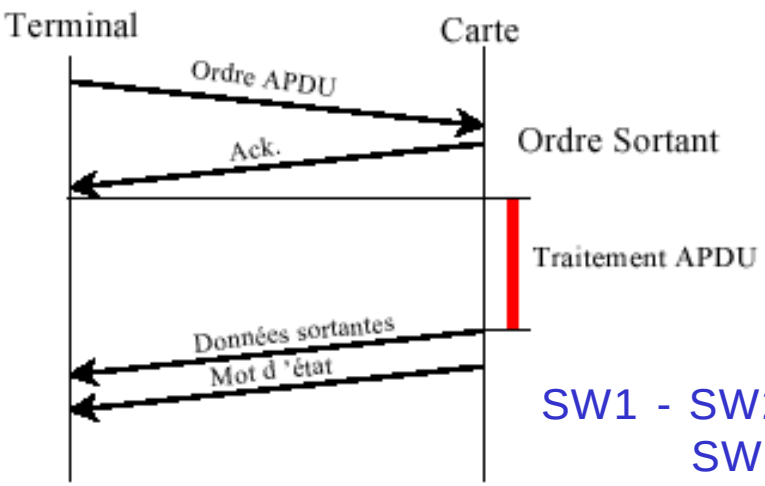
      R(#bloc, erreur): numérotation modulo 2, n° du prochain bloc attendu (erreur = 0)

      S notification de commandes diverses (RESYNC, IFS, ABORT,WTX)

Norme ISO 7816-4

- Protocole Asynchrone de type commande réponse
- APDU (Application Programming Data Units)





Command APDU						
Mandatory Header				Conditional Body		
CLA	INS	P1	P2	Lc	Data Field	Le

CLA : 1 octet pour identifier l'application
INS : 1 octet pour le code de l'instruction
P1 - P2 : Paramètres de l'instruction
Lc : Longueur du champ de données
Le : Longueur maxi du champ de données de la réponse

Response APDU		
Conditional Body	Mandatory Trailer	
Data Field	SW1	SW2

SW1 - SW2 : Code d'exécution 90 00 → OK
SW2 : 61 OK- 62,63 Warning
SW2: 64,65-erreur d'exécution
SW2: 67 commande incorrecte

Types de commandes (1/2)

Cas 1: Pas d'entrée / Pas de sortie

CLA	INS	P1	P2	P3
				lgth (= ' 00 ')

SW1	SW2
' 90 '	' 00 '

Cas 2: Pas d'entrée / Sortie de longueur connue

CLA	INS	P1	P2	P3
				lgth

Data de longueur lgth	SW1	SW2
	' 90 '	' 00 '

Cas 3: Pas d'entrée / Sortie de longueur inconnue

CLA	INS	P1	P2	P3
				lgth (= ' 00 ')

SW1	SW2
' 9F '	$lgth_1$

GET RESPONSE

CLA	INS	P1	P2	P3
				$lgth_2$

Data de longueur $lgth_2 < lgth_1$	SW1	SW2
	' 90 '	' 00 '

Note: lgth=' 00 ' cause un transfert de données de 256 bytes

Types de commandes (2/2)

Cas 4: Entrée / Pas de sortie

CLA	INS	P1	P2	P3	Data de longueur lgth
				lgth	

SW1	SW2
‘ 90 ’	‘ 00 ’

Cas 5: Entrée / Sortie de longueur connue ou inconnue

CLA	INS	P1	P2	P3	Data de longueur lgth
				Lgth	

SW1	SW2
‘ 9F ’	‘ lgth1’

GET RESPONSE

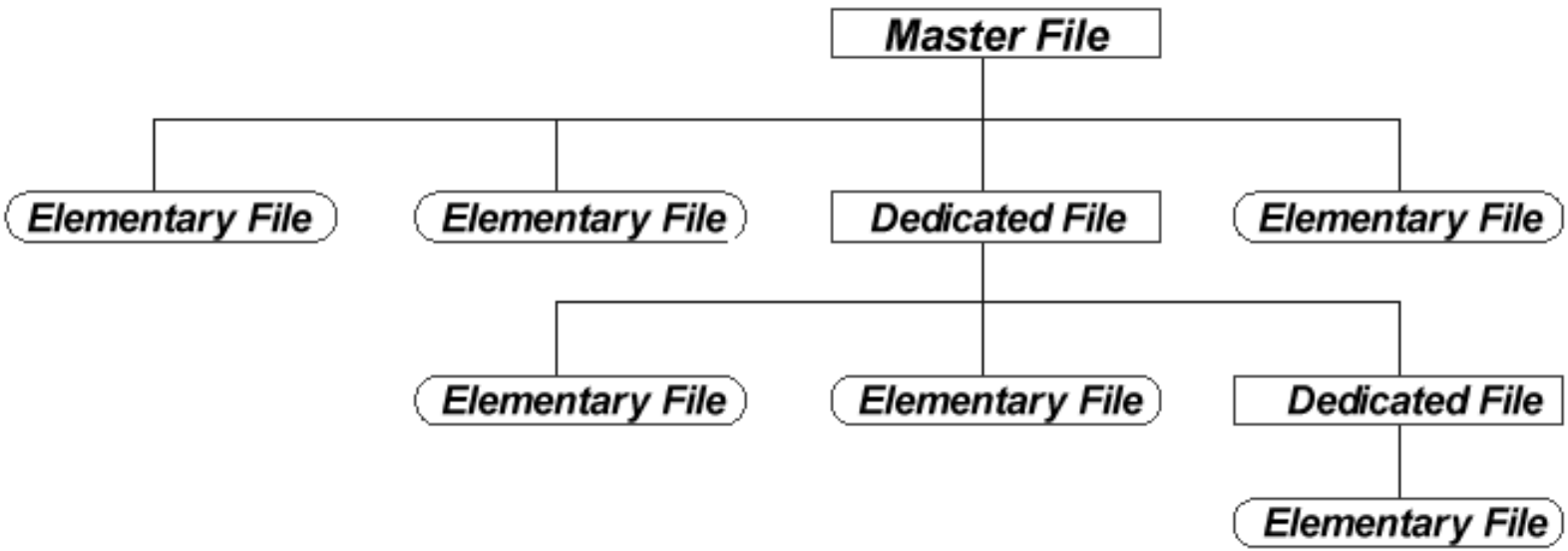
CLA	INS	P1	P2	P3
				$Lgth_2$

Data de longueur $lgth_2 < lgth_1$	SW1	SW2
	‘ 90 ’	‘ 00 ’

ISO 7816-4

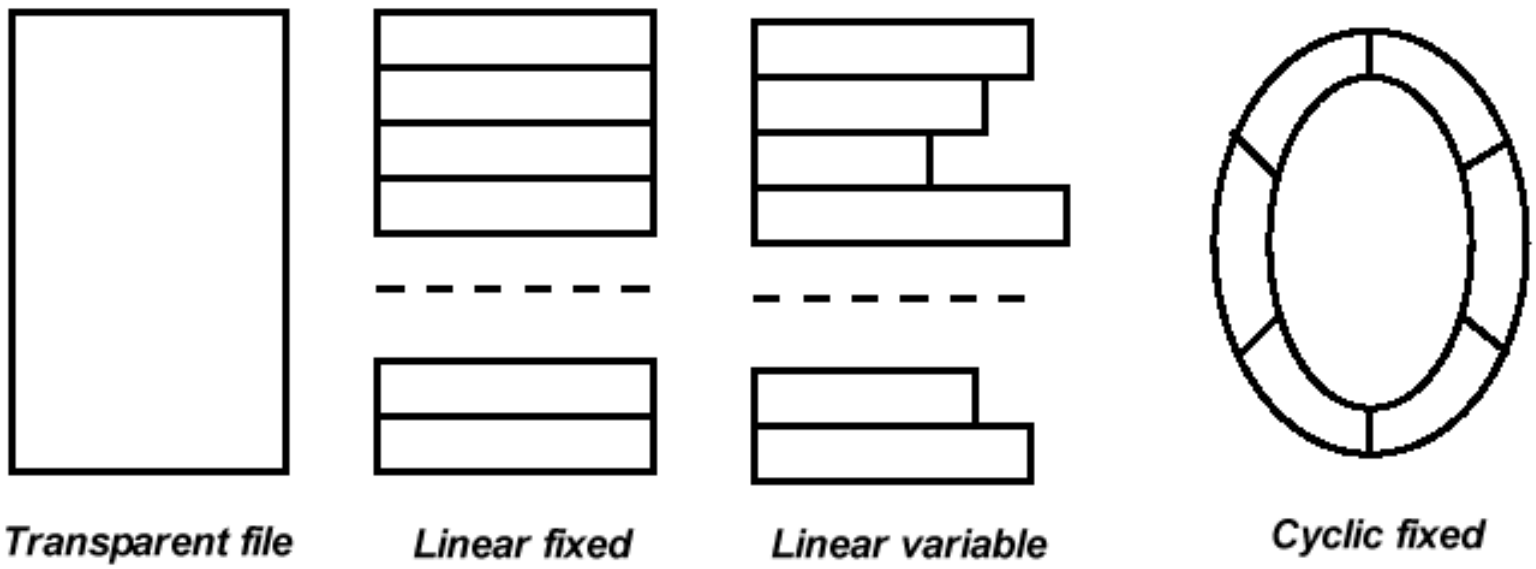
- Le système de fichiers des cartes à puce.
  Système de fichiers hiérarchique qui peut contenir 3 types de fichiers :

"Master File" (Fichier racine)
"Dedicated File" (Répertoire + qq infos)
"Elementary File" (Fichier de données)



ISO 7816-4

- 4 structures de données :



Exemples (1/3)

1. READ_BINARY.

CLA B0 P1 P2 Le.

Si P1(b8) = 1, EF est désigné par P1 (b5,b4,b3,b2,b1) et P2 représente l'offset. Sinon l'offset est égal à (256*P1) +P2

Lecture de Le octets à partir de offset dans un fichier transparent.

2. WRITE_BINARY.

CLA D0 P1 P2 Lc [Lc octets]

Si P1(b8) = 1, EF est désigné par P1 (b5,b4,b3,b2,b1) et P2 représente l'offset. Sinon l'offset est égal à (256*P1) +P2

Ecriture de Le octets à partir de offset dans un fichier transparent.

3. UPDATE_BINARY.

CLA D6 P1 P2 Lc [Lc octets].

Si P1(b8) = 1, EF est désigné par P1 (b5,b4,b3,b2,b1) et P2 représente l'offset. Sinon l'offset est égal à (256*P1) +P2

Ecriture de Le octets à partir de offset dans un fichier transparent.

Exemples (2/3)

5. READ_RECORD

CLA B2 P1 P2 Le

Lit un enregistrement dans un fichier

P1, numéro d’enregistrement ou premier enregistrement à lire. P1= 00 indique l’enregistrement courant

P2= 04 lecture de l’enregistrement P1, P2= 05 lecture des enregistrements à partir de P1 jusqu'à la fin du fichier.

6. WRITE_RECORD

CLA D2 P1 P2 Lc [lc octets]

Ecriture d’un enregistrement.

P1 numéro d’enregistrement

P2= 04 enregistrement P1

Exemples (3/3)

13. INTERNAL_AUTHENTICATE

CLA 88 P1 P2 Lc [Lc octets] Le

Cette commande réalise un calcul d'authentification relativement à une clé interne en transférant un nombre aléatoire (challenge) délivré par le lecteur.

P1 représente la référence d'un algorithme. P2 est égal à zéro par défaut.
Le challenge est contenu dans les Lc octets sortants.

14. EXTERNAL_AUTHENTICATE

CLA 88 P1 P2 Lc[Lc octets] Le

Cette commande met à jour l'état d'une carte en fonction du résultat d'un calcul réalisé par le lecteur à partir d'un nombre aléatoire délivré par la carte (CHALLENGE).

P1, référence d'un algorithme. P2 est égal à zéro par défaut.

15. GET_CHALLENGE

CLA 84 P1 P2 Le

Cette commande produit un nombre aléatoire

Commandes ISO 7816-4 inter-industries

- READ BINARY
- WRITE BINARY
- UPDATE BINARY
- ERASE BINARY
- READ RECORD
- WRITE RECORD
- APPEND RECORD
- UPDATE RECORD

- GET DATA
- PUT DATA
- SELECT_FILE
- VERIFY
- INTERNAL_AUTHENTICATE
- EXTERNAL_AUTHENTICATE
- GET_CHALLENGE
- GET_RESPONSE
- ENVELOPE
- MANAGE CHANNEL

ISO 7816-5

- Spécifie des identifiants d'applications (AID ou Application IDentifier)

- Un AID = identification unique d'une application de la carte et de certains types de fichiers.

- AID= chaîne de 16 octets
  - R premiers octets (RID) identifient le fournisseur d'application
  - Les 11 octets suivants représentent l'identifiant

- Activation d'une application
  - Par exemple par SELECT_FILE (00 A4 04 00 10 [AID])
  - Exemple: ATR stocké dans ET_ATR en /3F00/2F01 est sélectionné par 00 A4 02 00 02 2F01

ISO 7816-6

- Spécifie les éléments de données inter-industrie
- Spécifie les formats de codage, méthodes de récupération de ces éléments de données

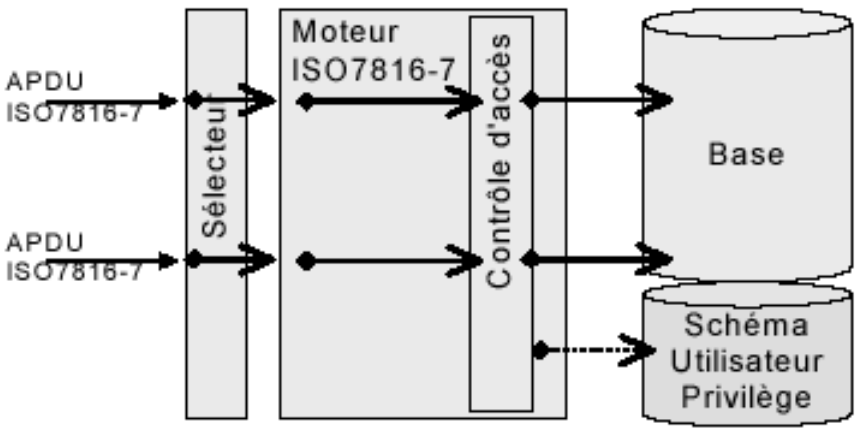
Nom du porteur de la carte
Date d'expiration
...

Etiquette	Longueur	valeur

ISO 7816-7

- Données organisées en tables, avec des colonnes, lignes, … (Similarité aux bases de données)

- Langage spécifique de requêtes : SCQL (Smart Card Query Language)



2000 PicoDBMS : Un SGBD sur carte à puce

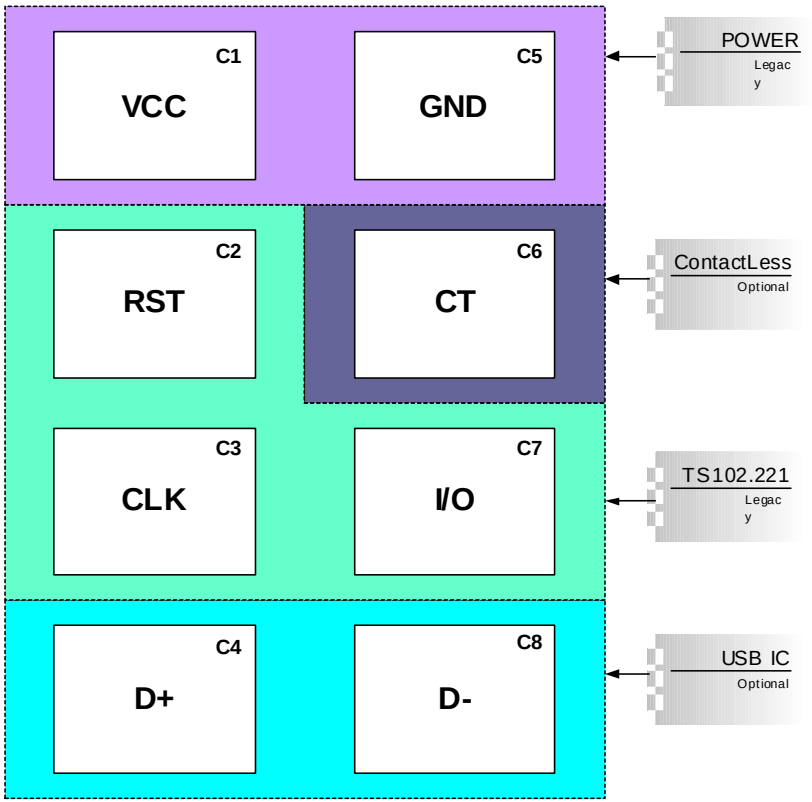
ISO 7816-8 à 10

- ISO 7816-8 : Sécurité de l'architecture et des commandes inter-industrie. 2 commandes à options
  - Manage Security environment
    - Passage d'un template à la carte
  - Perform Security Operations
    - Compute/Verify Cryptographic checksum
    - Encipher/Decipher/Hash
    - Compute/Verify Digital Signature
    - Generate Cryptographic key pairs
- ISO 7816-9 : Commandes inter-industries améliorées pour la gestion des cartes et de leurs fichiers
  - Register File (DF or EF)
  - Create, Deactivate, Delete, Rehabilitate
  - Seulement si la carte est en environnement « sûr » !
  - Méthodes d'accès aux ressources Smart-Cards
- ISO 7816-10 : Spécifiques aux cartes synchrones

axalto

ISO 7816-12

- Cartes à contact supportant le mode USB et procédures associées



Cycle de vie de la carte

- Fabrication
  - Inscription d'un programme en mémoire ROM définissant les fonctionnalités de base de la carte : "masque" figé traitant quelques commandes
- Initialisation
  - Inscription en EEPROM/NVM des données de l'application
- Personnalisation
  - Inscription en EEPROM/NVM des données relatives à chaque porteur
- Utilisation
  - Echange d'APDU
- Mort
  - Invalidation logique

Développement d'applications

- Le code applicatif de la carte est gravé en ROM au moment de la fabrication
  - carte figée
  - développeurs spécialisés
  - pas d'évolution possible : pas de chargement dynamique de nouveaux programmes en EEPROM

Implication

Si une application requiert de nouvelles fonctions carte => nécessité de re-développer un nouveau masque

axalto

Vers des cartes plus ouvertes

- **Problèmes à résoudre et/ou besoins à satisfaire**
  - permettre le développement de programmes pour la carte sans avoir besoin de graver un nouveau masque
  - faire de la carte un environnement d'exécution de programmes ouvert (chargement dynamique de code)
  - faciliter l'intégration des cartes dans les applications

Elément de solution :
Java Card

Utiliser le langage orienté objet : Java

Utiliser la plate-forme Java pour charger et exécuter des applications dynamiquement

axalto

Qu'est ce que la Java Card ?

- Une carte à puce qui exécute des programmes Java

- Java Card définit un sous-ensemble de Java (2.1 puis 2.2)
  dédié pour la carte à puce :
  - Sous-ensemble du langage de programmation Java
  - Sous-ensemble du paquetage `java.lang`
  - Découpage de la machine virtuelle Java
  - Modèle mémoire adapté à la carte
  - APIs spécifiques à la carte

axalto

Java Card par rapport à Java (1/4)

- Pas de chargement dynamique de classes
- Objets : Allocation dynamique d'objets supportée (**new**)

Mais

- Pas de ramasse-miettes (gc)
- Pas de désallocation explicite non plus
- ==> mémoire allouée ne peut pas être récupérée
- Pas de méthode **finalize()**

axalto

Java Card par rapport à Java (2/4)

- Types de base (nombres signés, complément à 2) : `byte`, `short`, `boolean` (8 bits), `int` (16 bits)
  - Pas de types **`char`** (pas de classe **`String`**), **`double`**, **`float`** et **`long`**
  - Pas de classes **`Boolean`**, **`Byte`**, **`Class`**, *etc.*

- Tableaux à une dimension :
  - Eléments : des types de base

Java Card par rapport à Java (3/4)

- Pas de threads
  - Pas de classe `Thread`, pas de mots-clé `synchronized`
- Mécanisme d'héritage identique à Java
  - Surcharge de méthodes, méthodes abstraites et interfaces
  - Invocation de méthodes virtuelles
  - Mots-clés `instanceof`, `super` et `this`
- Sécurité
  - Notion de paquetage et modifieurs `public`, `protected` et `private` identiques à Java
  - Pas de classe `SecurityManager` : politique de sécurité implémentée dans la machine virtuelle
- Méthodes natives (`native`)
- Atomicité
  - Mise à jour de champs d'objets doit être atomique
  - Modèle transactionnel : `beginTransaction()`, `commitTransaction()` et `abortTransaction()`

axalto

Java Card par rapport à Java (4/4)

- Mécanismes d'exception supportés
  - Peuvent être définis (`extends Throwable`), propagés (`throws`) et interceptés (`catch`)
  - Classes `Throwable`, `Exception` et `Error` supportées et certaines de leurs sous-classes (dans `java.lang`)
  - Throwable { public Throwable(); }

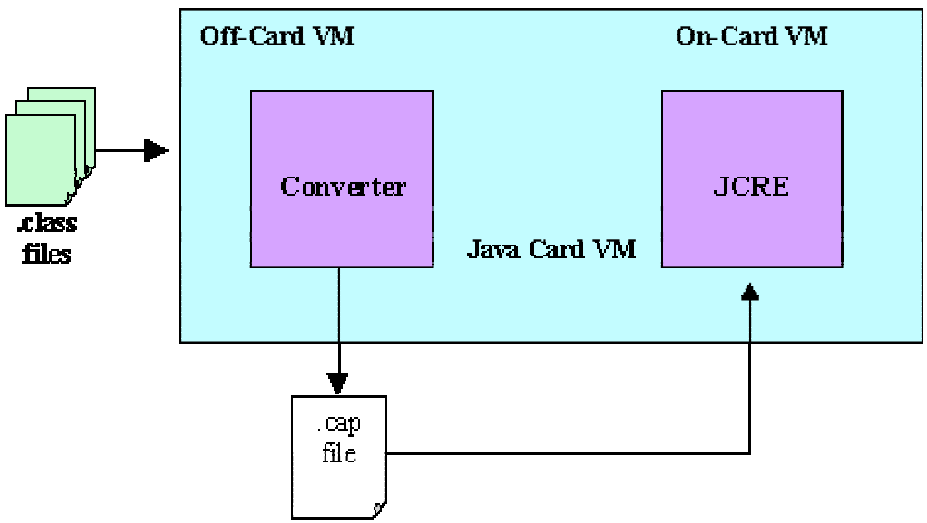
```
-- Exception
  -- RuntimeException
        -- ArithmeticException
        -- ClassCastException
        -- NullPointerException
        -- SecurityException
        -- ArrayStoreException
        -- NegativeArraySizeException
        -- IndexOutOfBoundsException
-- ArrayIndexOutOfBoundsException
```

Java Card 2.2: principales extensions

- Gestion des canaux logiques
- Destruction d’ Applet
- RMI JavaCard
- Ramasse-miettes
- Extension des classes APDU
- API d’accès aux objets transients (tbc)
- API de « Card Management »

Machine virtuelle

- Implémentation en deux parties :
  - La partie on-card (SmartCard)
  - La partie off-card (JavaCard)



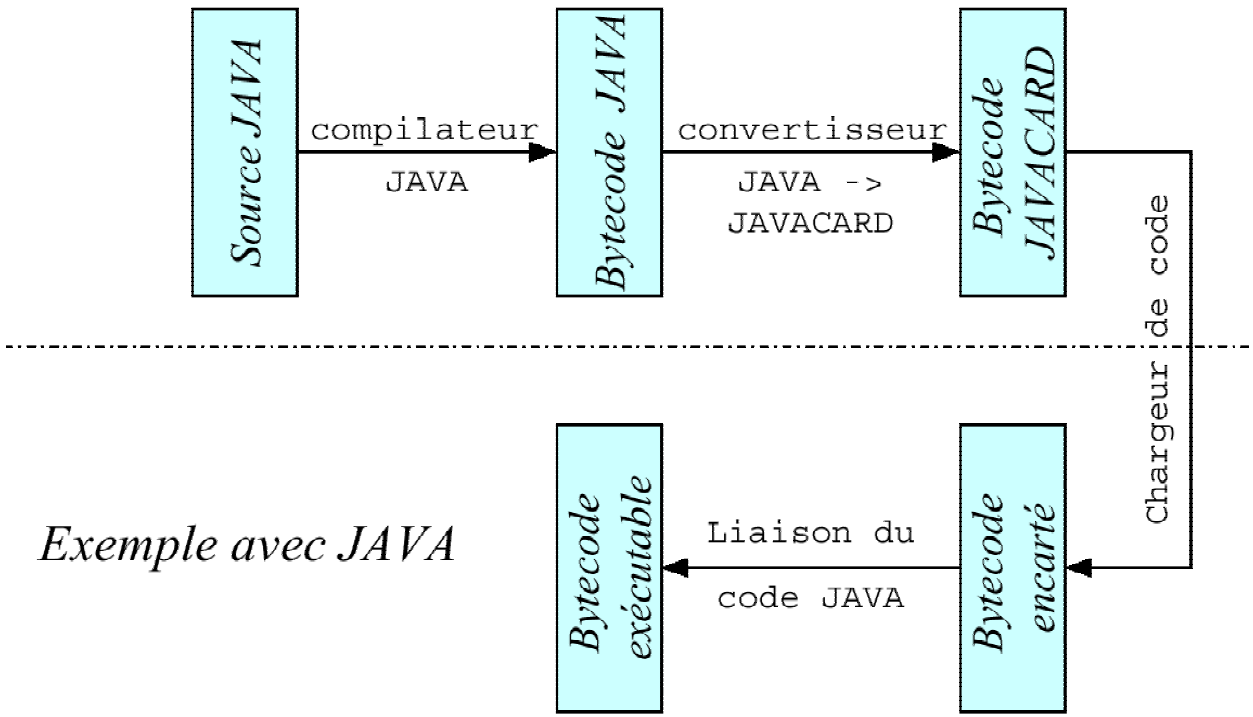
Librairies standard

- JavaCard.lang
  - Classes fondamentales (object, throwable) pour le langage JavaCard
- JavaCard.framework
  - Classes et interfaces pour les fonctionnalités de base des applets JavaCard (ISO7816, PIN,AID, APDU,JCSystem, Exceptions…)
- JavaCard.security
  - Classes et interfaces pour l'environnement sécuritaire (DesKey,DASPrivateKey,SecretKey,RandomData,Signature, MessageDigest…)
- JavaCardx.crypto
  - Classes de sécurité et interfaces pour les fonctionnalités soumises à contrôle d'export (KeyEncryption, Cipher)

axalto

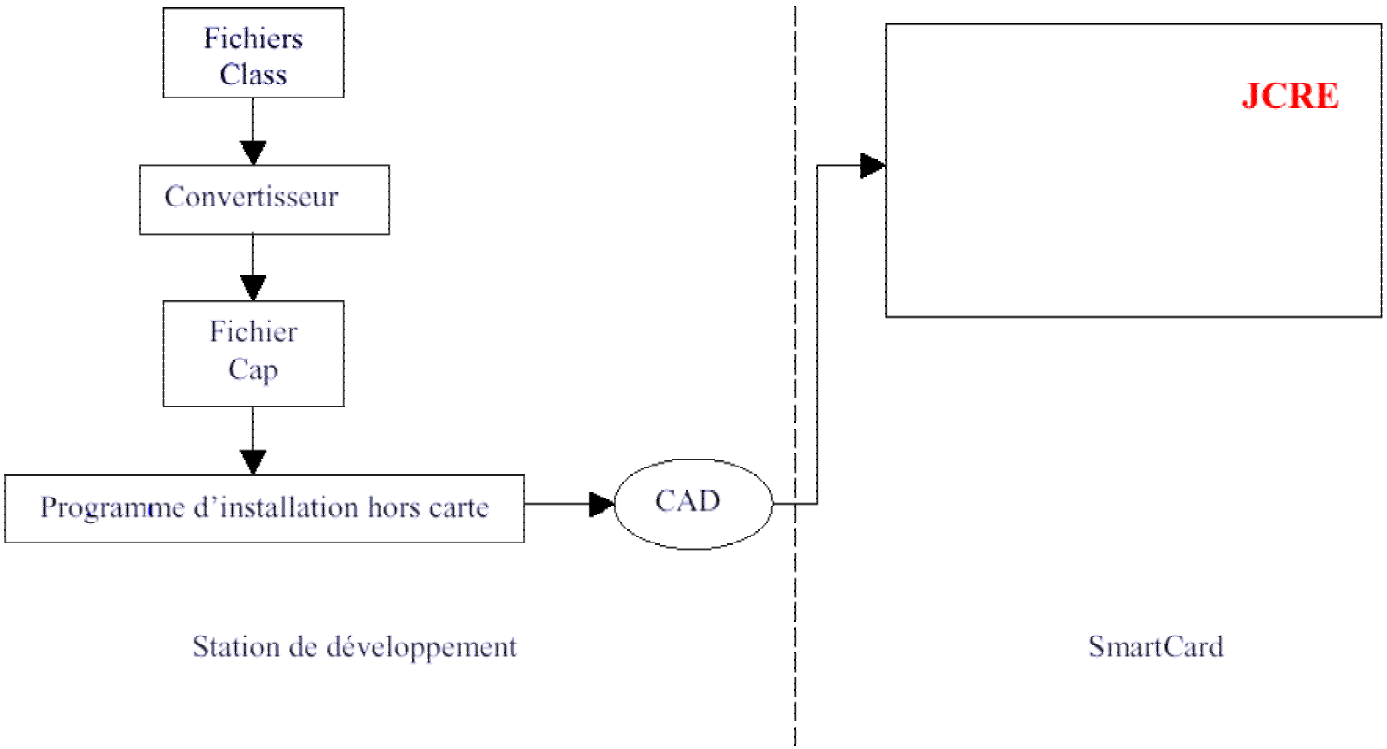
Compiler en Java (1/2)

- Obtention d'un code JavaCard

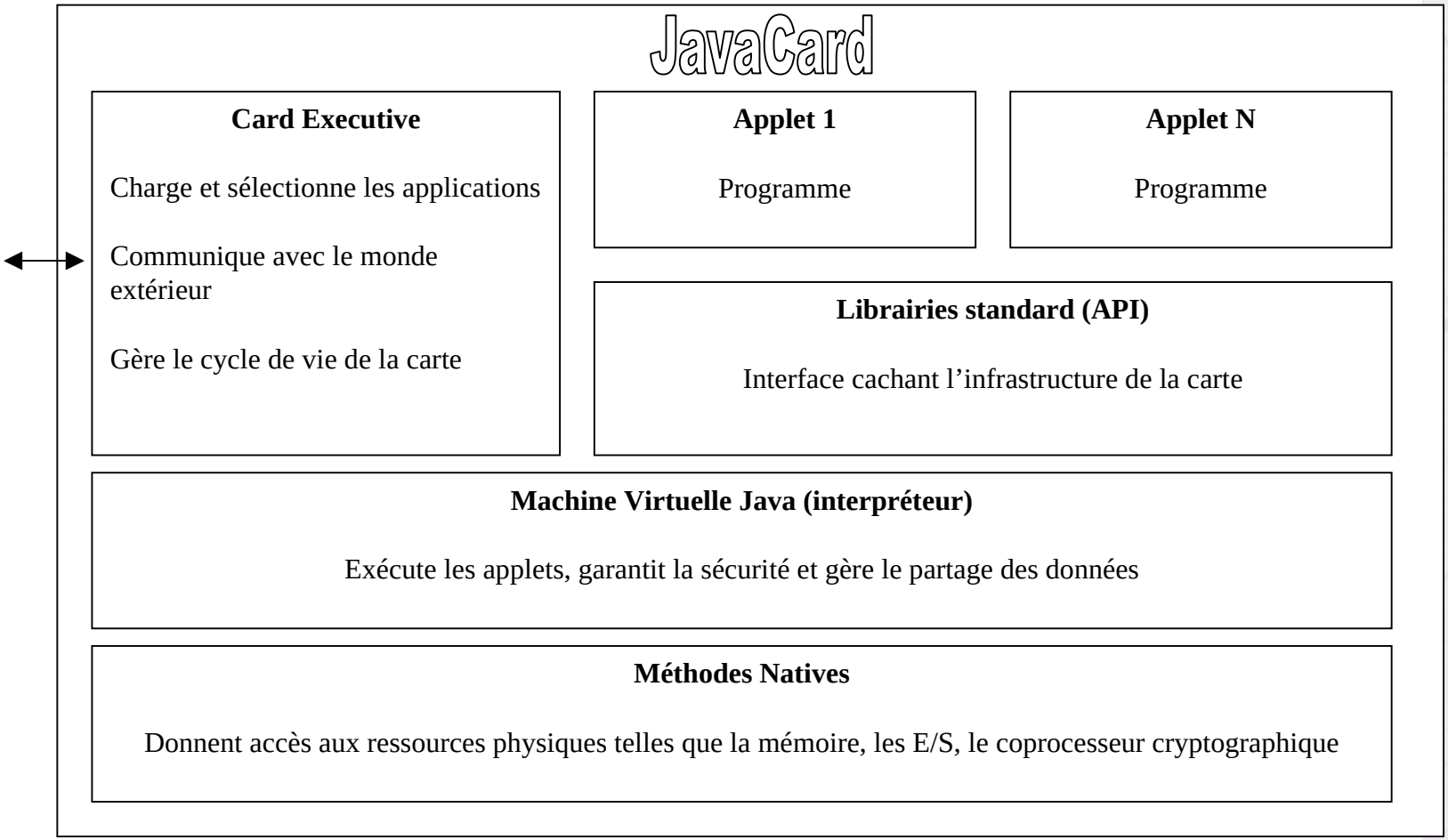


Compiler en Java (2/2)

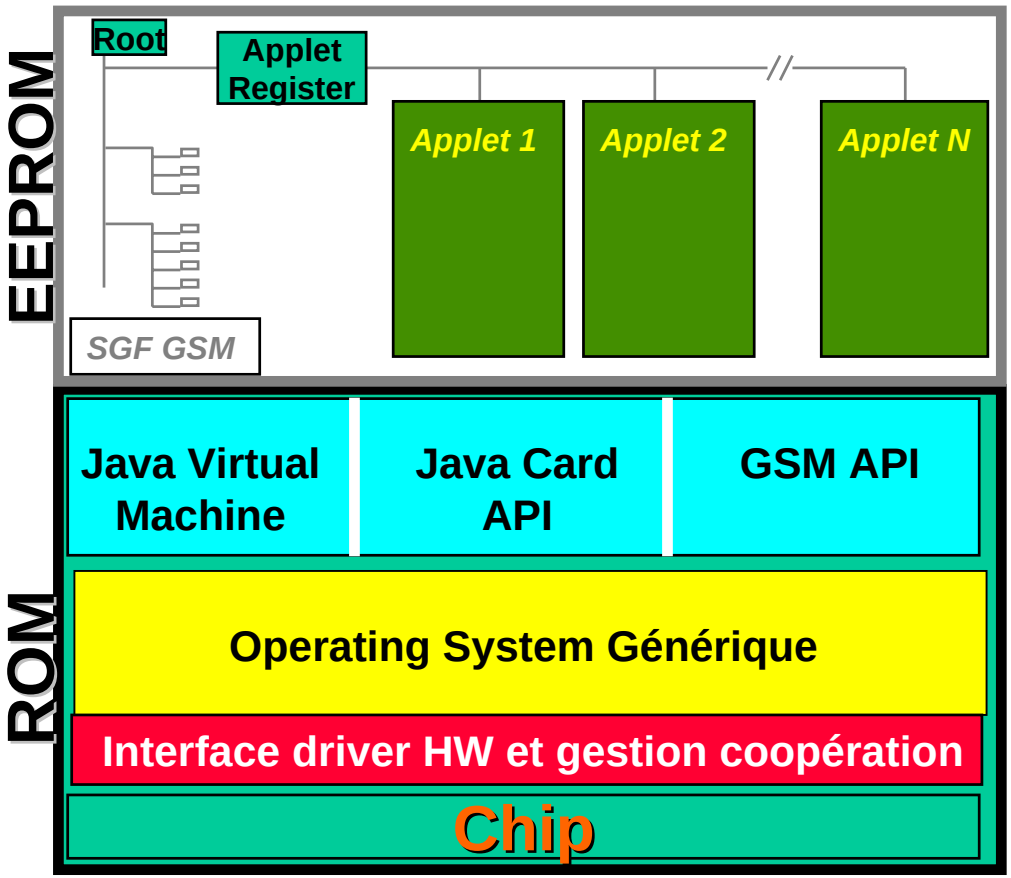
- Récapitulatif des opérations



Architecture type Java Card



Cartes à puce: architecture moderne



- + Portabilité
- + Rapidité de development
- + Plate-forme ouverte
- + Multi-application

Sécurité des cartes à puce

6/21/2005

axalto
A Schlumberger company

Types d'attaques sur les cartes à puce (1)

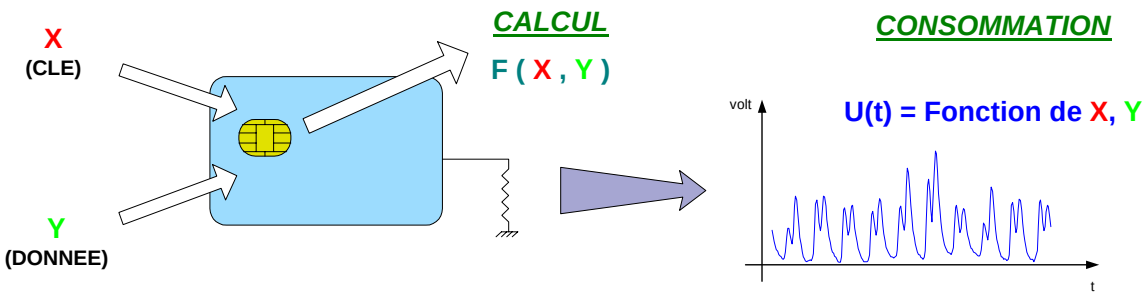
- Information transmises
  - Canaux de fuite (courant, paramètres d'algorithmes cryptographiques…)
- Information stockées
  - Données (clés) ou code exécutable
- Mise en défaut du Hardware
  - Tension, horloge, mise hors specs de manière générale
- Défauts du Software
  - Chainage de commandes ou erreurs protocoles cryptographiques
- Attaques temporelles
  - Détection de changement de temps d'exécution
- Attaques sur les consommations
  - photographie temps réel des instructions en cours d'exécution
  - Post-traitement statistique des informations

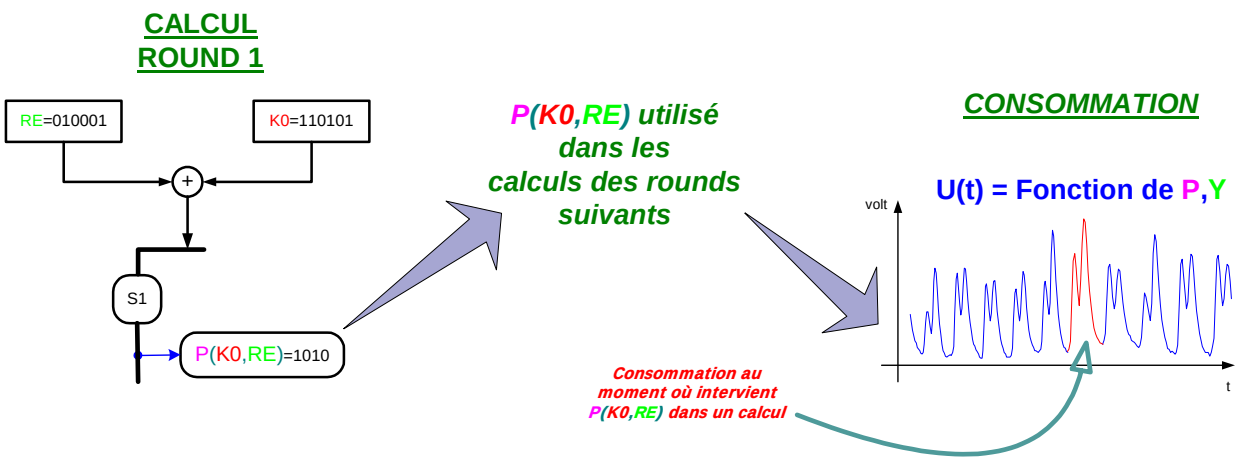
Attaques sur les cartes à puce (2)

- Attaques sur le chemin de test
  - Mise en mode test, sondes, reconstruction du design logique
- Ingéniérie inverse
  - Reconstruction du Layout, dump du code ROM, révélation chimique
  - Utilisation de MEB, FIB ou autre pour révélation du contenu EEPROM
- Glissements d'alimentation ou interruption d'horloge
  - Corruption de données sur le (les) bus
  - Peuvent affecter un tout petit nombre de cycles CPU
- Attaques Lumière (Laser)
- Mesures de radiations électromagnetique
- Imagerie par scan Laser

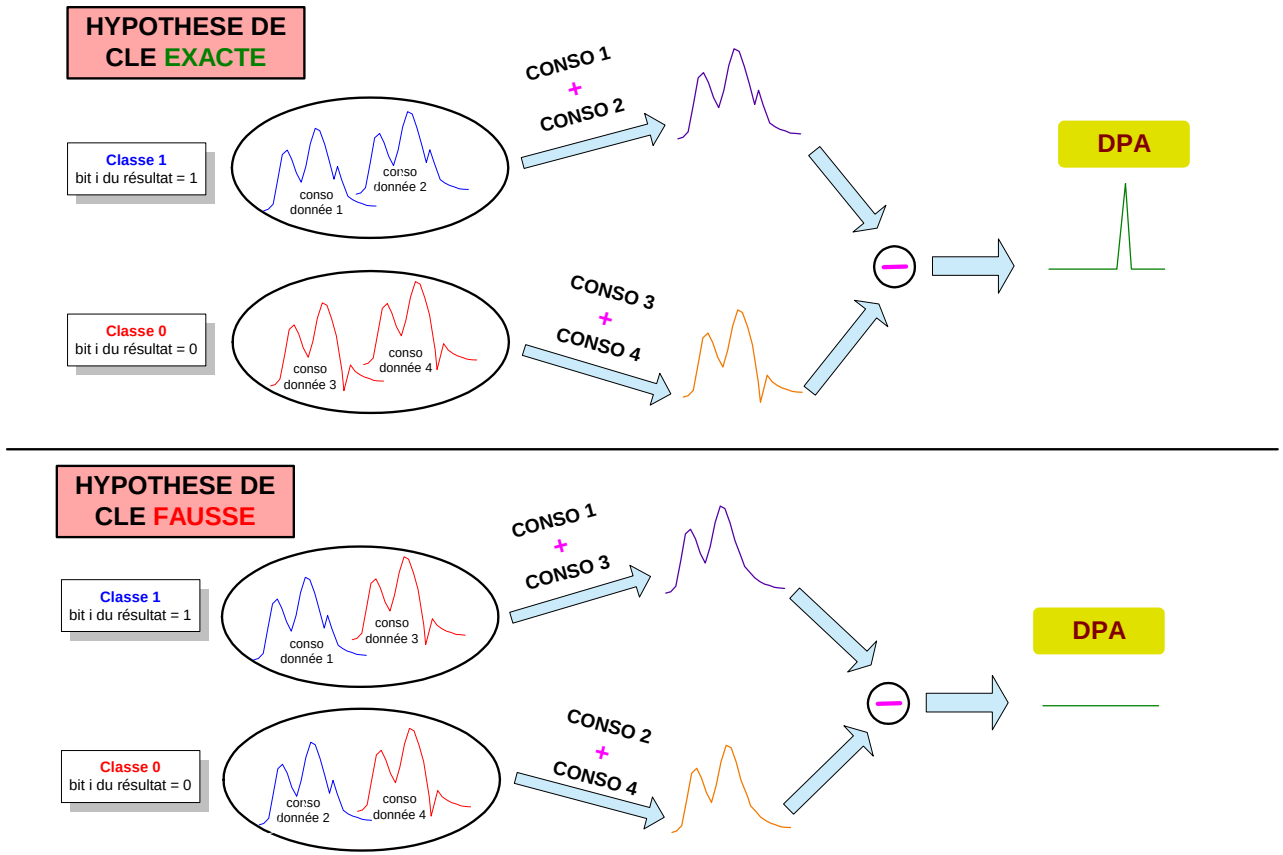
axalto

Exemple: attaque DPA (1/3)

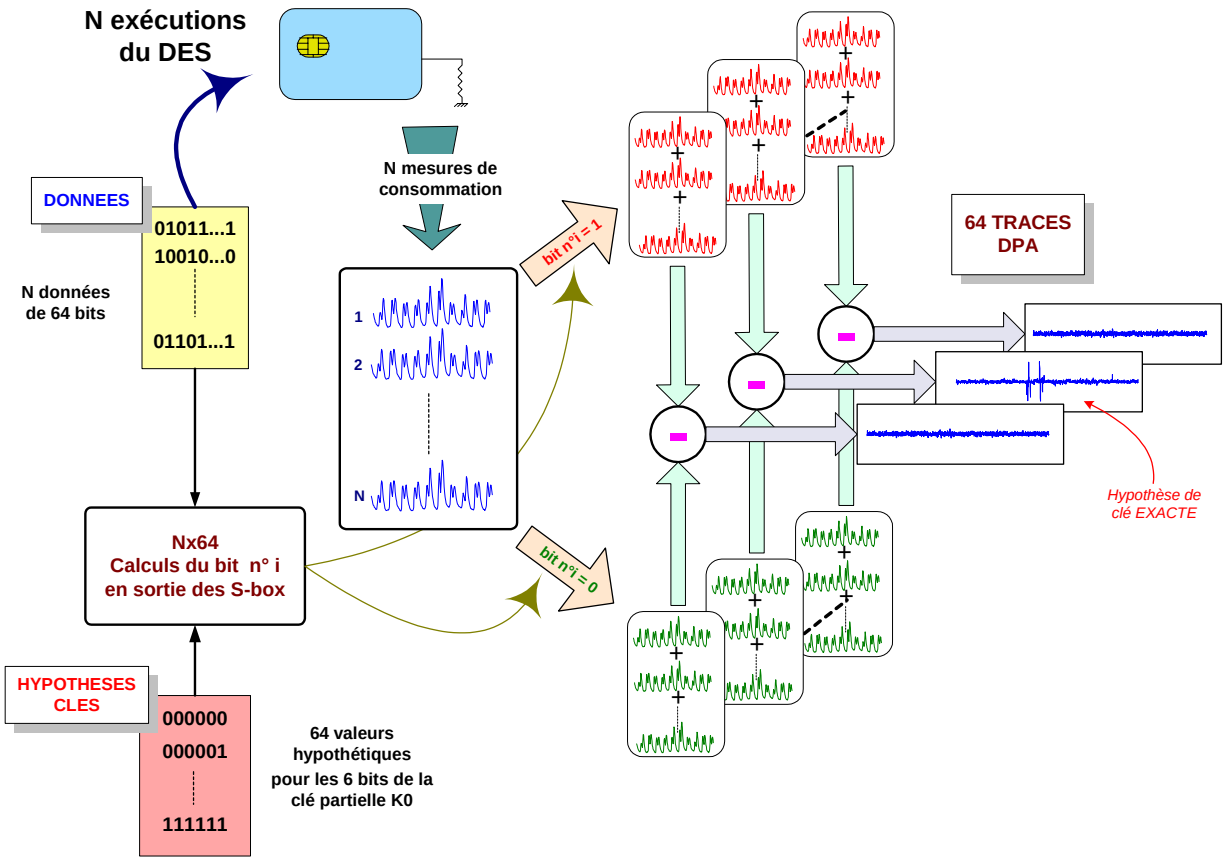




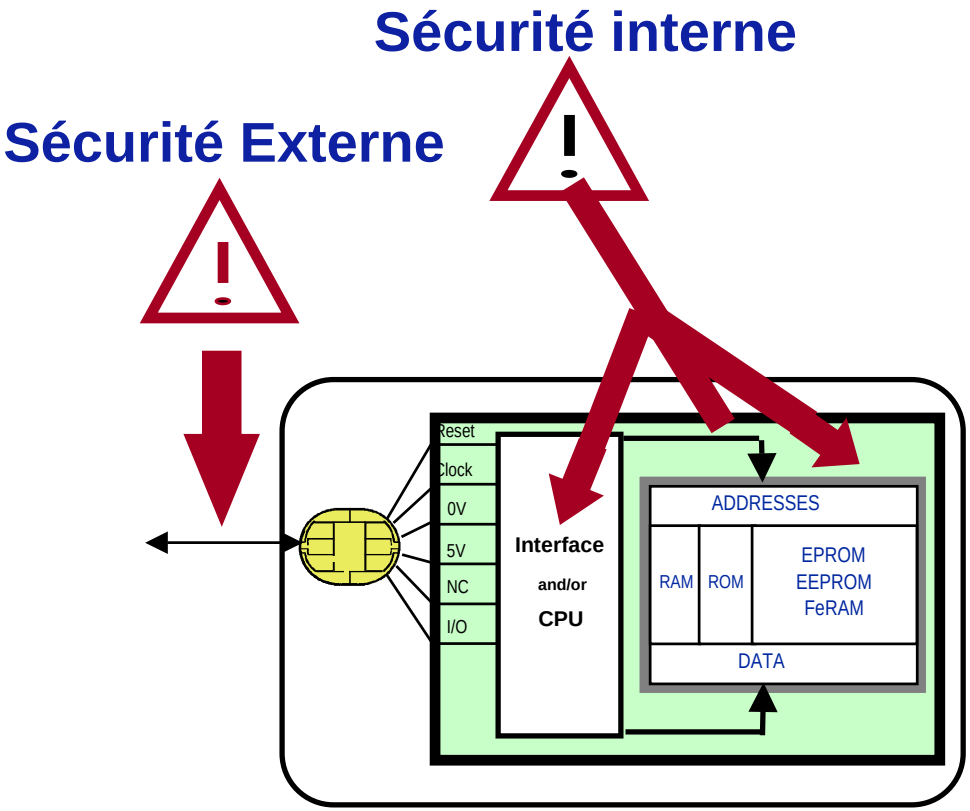
Exemple: attaque DPA (2/3)



Exemple: attaque DPA (3/3)



Cartes à puce: Sécurité Hardware



- Couches actives anti-intrusion
- EEProm non révélable
- Sécurité physique
  - **pas de retour mode test**
- Détecteurs de rayonnements
  - **UV, ionisation, etc...**
- Pas de comportement statique
  - **fréquence d'horloge minimale**
- Contrôle des modes hors specs
- Protection DPA/SPA
- Protections auto-programmées
- Pas d'interférence OS/Interface
  - **contrôle du PIN**
- Logique asynchrone, faible conso**.**

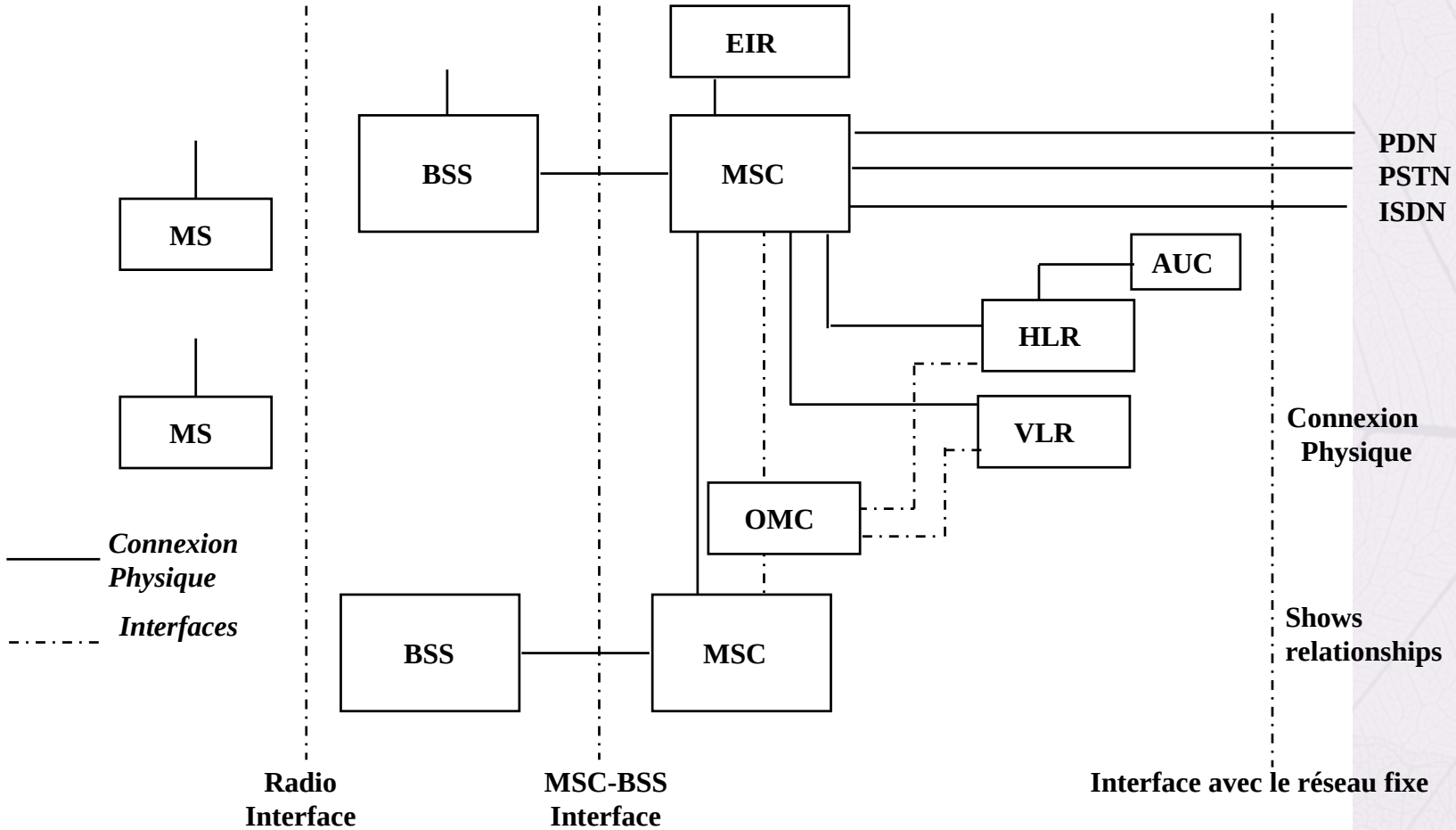
Cartes à puce: Sécurité Software

- Masquage des points de synchronisation
  - Timing aléatoires, opérations aléatoires
- Masquage des actions cruciales
  - Duplication, fausses pistes, symétrisation,..
- Cryptographie
- Contrôle des droits d 'accès
- Collaboration avec le MMU HW
- Certification commune HW/SW (« Critères Communs »)

Exemple fondamental: La carte GSM/SIM

axalto
A Schlumberger company

Rappel: architecture Globale GSM



MS:	Mobile Station	VLR:	Visited Location Register
BSS:	Base Station System	OMC:	Operation and Maintenance Centre
MSC:	Mobile Services Switching Centre	EIR:	Equipment Identity Register
HLR:	Home Location Register	AUC:	Authentication Centre

Fonctions de sécurité offertes au niveau d'un réseau GSM PLMN

- Confidentialité de l'identité du souscripteur (IMSI)
- Authentification de l'identité du souscripteur (IMSI)
- Confidentialité des données utilisateur lors des connexions physiques
- Confidentialité des données utilisateur en mode « Connectionless »
- Confidentialité des éléments d'information de signalisation

GSM 11.11

- Définit l' interface entre le « Subscriber Identity Module » (SIM) et le terminal mobile (« Mobile Equipment ou ME ») pendant les phases d'opération du réseau GSM
- Definit l'organisation interne de la carte SIM.
- Assure l'interoperabilité entre SIM et ME independemment des différents fabricants et des opérateurs

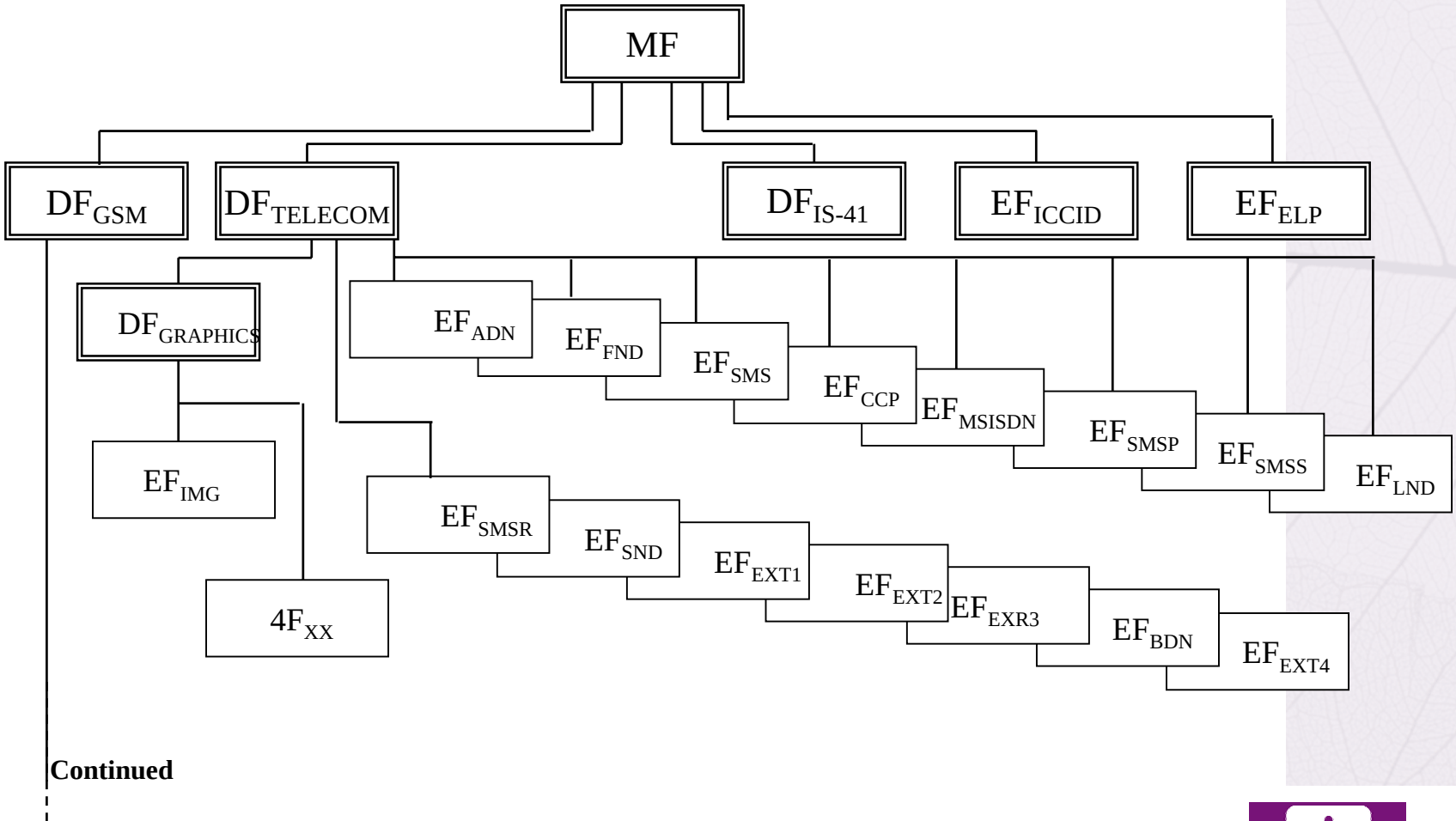
Partage de la Station Mobile (MS) → SIM
Partage de la Station Mobile (MS) → ME

Partage des fonctions → GSM
Partage des fonctions → Gestion de l'Administration

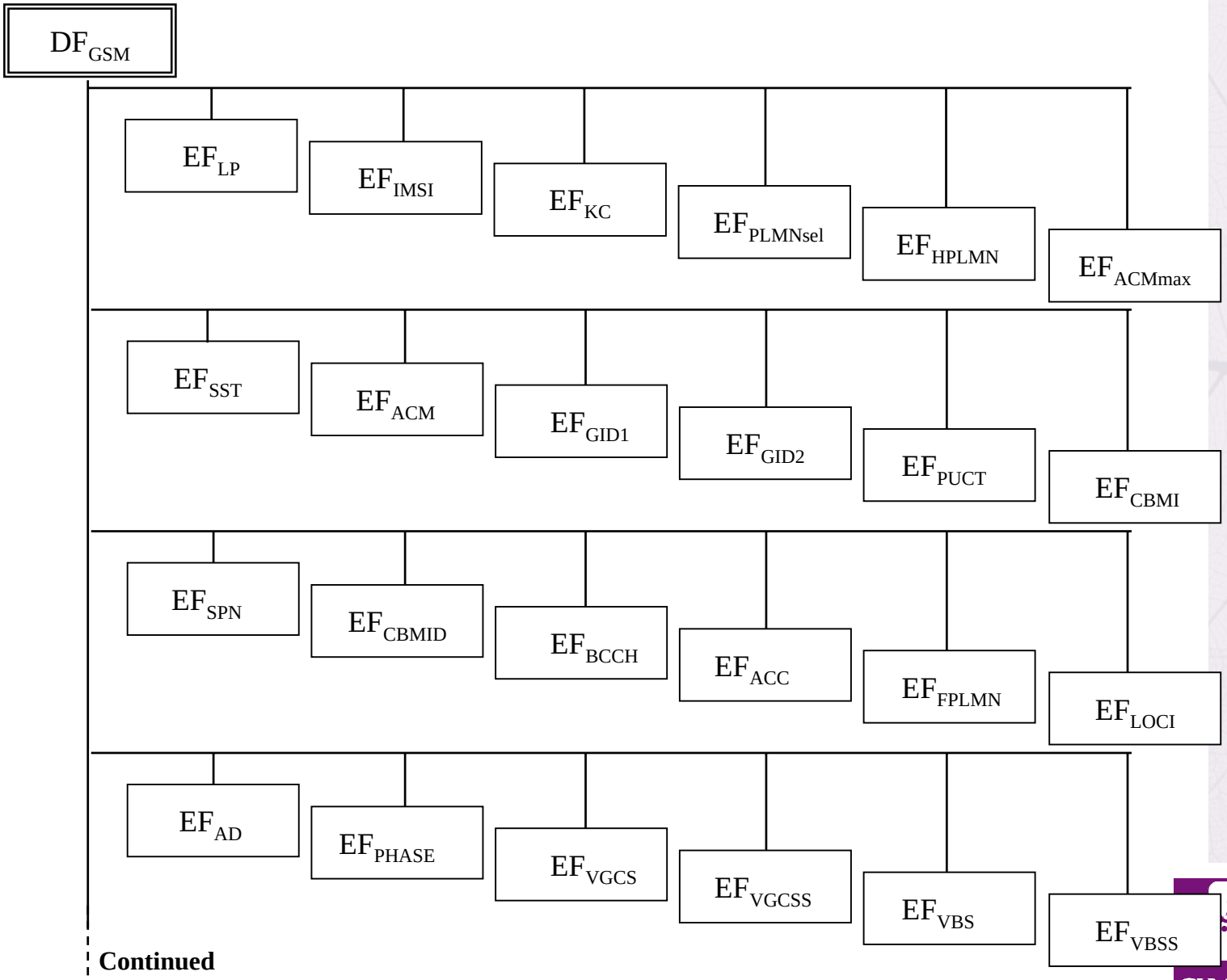
Items couverts par la norme

- Caractéristiques physiques, signaux électriques protocoles de transmission
- Modèle pour la conception de la structure logique de la SIM
- Caractéristiques sécuritaires
- Fonctions d’interface
- Description des commandes
- Contenu des fichiers requis pour l’application GSM
- Protocole applicatif

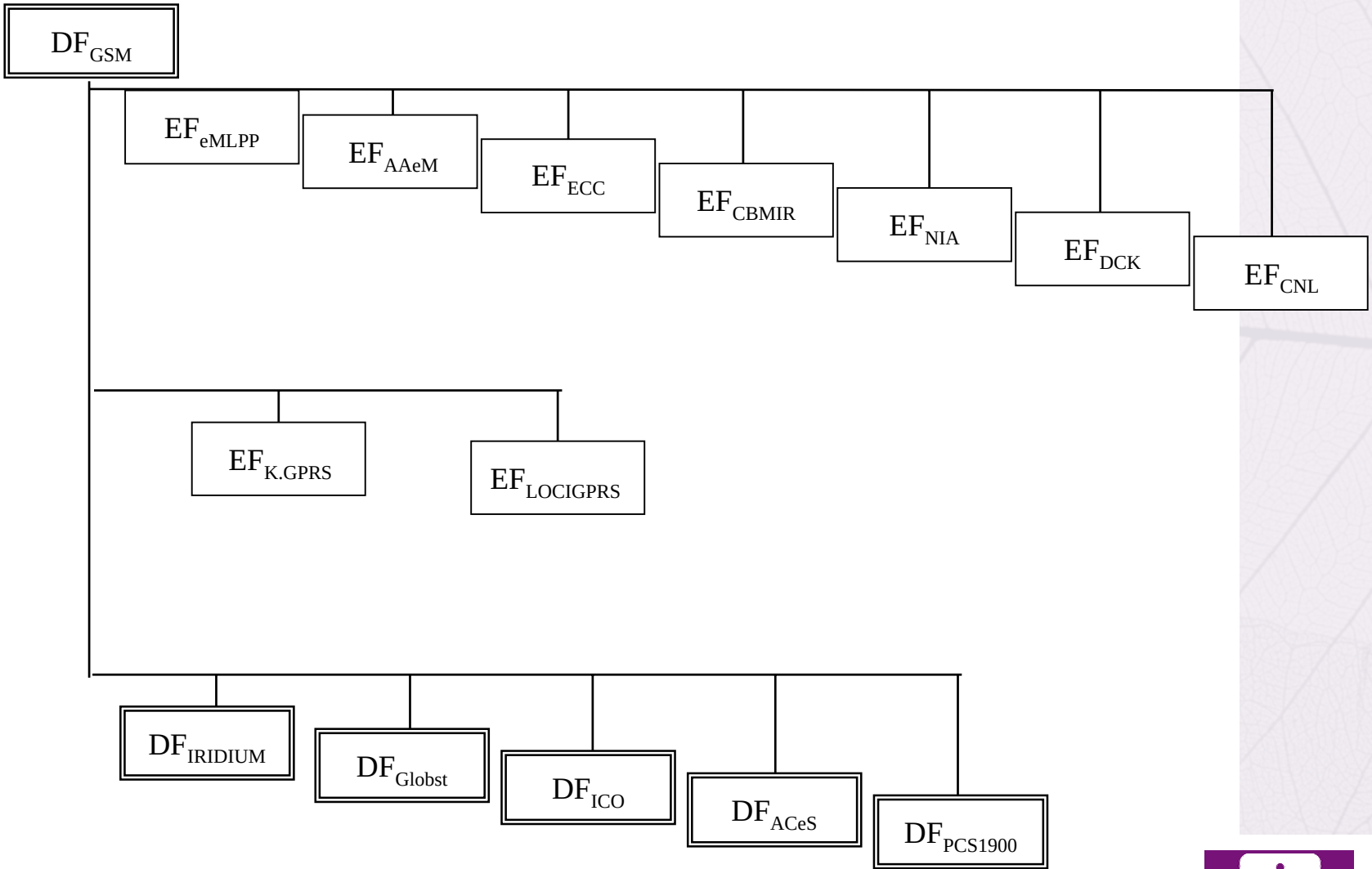
Carte SIM: organisation interne (1/3)



Carte SIM: organisation interne (2/3)



Carte SIM: organisation interne (3/3)



Protocole Applicatif GSM

- Message: commande ou réponse.
- Couple commande/réponse GSM : suite de messages consistant en une commande et la réponse associée.
- Procédure GSM: suite de un ou plusieurs couples commande/réponse GSM
- Session GSM: intervalle de temps entre la fin de la procédure d'initialisation et se terminant:
  - Soit au démarrage de procédure de fin de session GSM
  - Soit à l'instant d'interruption du lien SIM-ME

Procedures (1)

- Procédures générales:
  - Lecture d'un EF ME
  - Mise à jour d'un EF ME
  - Extension d'un EF ME

- Procédures de gestion SIM :
  - Initialisation SIM ME
  - Fin de session GSM ME
  - Requêtes de codes d'appel d' urgence ME
  - Requêtes d'extension de langage préférentiel ME
  - Requêtes de langage préférentiel ME
  - Requêtes d'information administrative ME
  - Requêtes sur la table des services SIM ME
  - Requêtes de phase SIM ME

axalto

Procedures (2)

- Procédures liées à la sécurité GSM :
  - Calcul d'algorithme GSM — NET
  - Requête d'IMSI — NET
  - Requête d'information type contrôle d'accès — NET
  - Requête intervalle de recherche HPLMN — NET
  - Informations de localisation — NET
  - Clé de Chiffrement — NET
  - Information canal BCCH — NET
  - Information sur PLMN interdits — NET
- Procédures liées au code CHV:
  - Vérification CHV — MMI
  - Substitution de valeur CHV — MMI
  - Déconnexion CHV — MMI
  - Mise en route CHV — MMI
  - Déblocage CHV — MMI

Procédures (3)

Procédures de souscription:

Numéros d'appel (ADN, FDN, MSISDN, LND, SDN, BDN)	MMI/ME
Short messages (SMS)	MMI
Indications de facturation (AoC)	MMI
Paramètres sur Capacités de Configuration (CCP)	MMI
Sélection PLMN	MMI
Cell Broadcast Message Identifier (CBMI)	MMI
Group Identifier Level 1 (GID1)	MMI/ME
Group Identifier Level 2 (GID2)	MMI/ME

Procedures (4)

- Procédures de souscription (suite.):
  - Nom du fournisseur de services (SPN) — ME
  - Voice Group Call Service (VGCS) — MMI/ME
  - Voice Broadcast Service (VBS) — MMI/ME
  - Pré-emption et Priorité Multi-niveau étendue (eMLPP) — MMI/ME
  - Dé-personnalisation des clés de contrôle — ME
  - Status report sur les SMS (SMSR) — MMI
  - Indicateurs d'alerte réseau — ME

Commandes SIM

1 SELECT
2 STATUS
3 READ BINARY
4 UPDATE BINARY
5 READ RECORD
6 UPDATE RECORD
7 SEEK
8 INCREASE
9 VERIFY CHV
10 CHANGE CHV
11 DISABLE CHV
12 ENABLE CHV
13 UNBLOCK CHV
14 INVALIDATE
15 REHABILITATE
16 RUN GSM ALGORITHM
17 SLEEP (Obsolete: put SIM in Low Power)
18 TERMINAL PROFILE
19 ENVELOPPE
20 FETCH
21 TERMINAL RESPONSE
22 GET RESPONSE

	File				
Function	**MF**	**DF**	**EF transparent**	**EF linear fixed**	**EF cyclic**
SELECT	*	*	*	*	*
STATUS	*	*	*	*	*
READ BINARY			*		
UPDATE BINARY			*		
READ RECORD				*	*
UPDATE RECORD				*	*
SEEK				*	
INCREASE					*
INVALIDATE			*	*	*
REHABILITATE			*	*	*

Codage des commandes GSM

COMMAND	INS	P1	P2	P3	S/R
SELECT	'A4'	'00'	'00'	'02'	S/R
STATUS	'F2'	'00'	'00'	lgth	R
READ BINARY	'B0'	offset high	offset low	lgth	R
UPDATE BINARY	'D6'	offset high	offset low	lgth	S
READ RECORD	'B2'	rec No.	mode	lgth	R
UPDATE RECORD	'DC'	rec No.	mode	lgth	S
SEEK	'A2'	'00'	type/mode	lgth	S/R
INCREASE	'32'	'00'	'00'	'03'	S/R
VERIFY CHV	'20'	'00'	CHV No.	'08'	S
CHANGE CHV	'24'	'00'	CHV No.	'10'	S
DISABLE CHV	'26'	'00'	'01'	'08'	S
ENABLE CHV	'28'	'00'	'01'	'08'	S
UNBLOCK CHV	'2C'	'00'	see note	'10'	S
INVALIDATE	'04'	'00'	'00'	'00'	-
REHABILITATE	'44'	'00'	'00'	'00'	-
RUN GSM ALGORITHM	'88'	'00'	'00'	'10'	S/R
SLEEP	'FA'	'00'	'00'	'00'	-
GET RESPONSE	'C0'	'00'	'00'	lgth	R
TERMINAL PROFILE	'10'	'00'	'00'	lgth	S
ENVELOPE	'C2'	'00'	'00'	lgth	S/R
FETCH	'12'	'00'	'00'	lgth	R
TERMINAL RESPONSE	'14'	'00'	'00'	lgth	S

CLA= ‘A0’ pour l’application GSM

axalto

GSM 11.14: vue d'ensemble de l'environnement SIM Application Toolkit

- L'environnement SIM Application Toolkit fournit des mécanismes permettant aux applications présentes dans la carte SIM, d'interagir et d'inter-opérer avec tout terminal mobile (ME) supportant les mécanismes spécifiques requis par ces applications.

- Si $ (Multiple Card) $ est supporté une carte SIM supportant les mécanismes SAT doit être capable de communiquer avec des cartes additionnelles et de recevoir de informations des lecteurs additionnels via le ME.

- Les mécanismes SAT sont dépendants des commandes et protocoles relevant de la norme GSM 11.11.
  - Découverts par une fin de procédure en '91 XX' (nécessite Fetch Data de XX)
  - SAT identifié dans le EF_{SST}
  - Capacités du ME identifiées dans la commande terminal profile

Procédures SAT

- Procédures SIM Application Toolkit:
  - **Téléchargement de données via SMS-CB (CBMID) NET**
  - **Téléchargement de données via SMS-PP NET**
  - **Sélection de menu MMI**
  - **Contrôle d'appel MMI/ME/NET**
  - **SIM Proactive MMI/ME/NET**
  - **Contrôle SIM de SMS généré par le ME MMI/ME/NET**
  - **Requête d'image (si §(Image)§ est supportée) MMI/ME**

GSM 11.14: Vue d’ensemble de la structure SIM Application Toolkit

- Mécanismes de base
  - Profile Download
  - Proactive SIM
  - Data download to SIM
  - Menu Selection
  - Call control by SIM
  - MO Short Message control by SIM
  - Event download
  - Security
  - Multiple card
  - Timer Expiration

Commandes et procédures SIM proactives

DISPLAY TEXT
GET INKEY
GET INPUT
LANGUAGE NOTIF
PLAY TONE
SET UP IDLE MODE TEXT
SELECT ITEM
SET UP MENU
CLOSE CHANNEL
GET CHANNEL STATUS
OPEN CHANNEL
SEND SHORT MESSAGE
SEND USSD
SET UP CALL
SEND DTMF

RUN AT COMMAND
SEND DATA
RECEIVE DATA
SEND SS
POWER OFF CARD
POWER ON CARD
GET READER STATUS
PERFORM CARD APDU
POLL INTERVAL
REFRESH
POLLING OFF
LAUNCH BROWSER
PROVIDE LOCAL INFORMATION
SET UP EVENT LIST
TIMER MANAGEMENT
MORE TIME

GSM 11.14: vue d'ensemble de l'environnement SIM Application Toolkit

Support du SIM Application Toolkit par les Equipements Mobiles

	Classe		
Command description	**1**	**2**	**3**
CALL CONTROL		X	X
CELL BROADCAST DOWNLOAD		X	X
DISPLAY TEXT		X	X
EVENT DOWNLOAD			
. MT call			X
. Call connected			X
. Call disconnected			X
. Location status			X
. User activity			X
. Idle screen available			X
GET INKEY		X	X
GET INPUT		X	X
GET READER STATUS (if $(MultipleCard)$ is supported)			Lc
MENU SELECTION		X	X

GSM 11.14: vue d'ensemble de l'environnement SAT

Support du SIM Application Toolkit par les Equipements Mobiles

	Classe		
Command description	**1**	**2**	**3**
MO SHORT MESSAGE CONTROL			X
MORE TIME		X	X
PERFORM CARD APDU (if $(MultipleCard)$ is supported)			Lc
PLAY TONE		X	X
POLLING OFF		X	X
POLL INTERVAL		X	X
POWER ON CARD (if $(MultipleCard)$ is supported)			Lc
POWER OFF CARD (if $(MultipleCard)$ is supported)			Lc
PROVIDE LOCAL INFORMATION		X	X
REFRESH	X	X	X
SELECT ITEM		X	X
SEND SHORT MESSAGE		X	X
SEND SS		X	X
SEND USSD			X
SET UP CALL		X	X
SET UP EVENT LIST			X
SET UP MENU		X	X
SMS-PP DOWNLOAD	X	X	X
TIMER MANAGEMENT (if $(Timer)$ is supported)			Lc
TIMER EXPIRATION (if $(Timer)$ is supported)			Lc

axalto

SIM et ME en action (1)

- Receive ATR
- Execute PPS

Initialisation

- SELECT DF_{GSM}
- GET RESPONSE
- SELECT EF_{Phase}
- READ BINARY
- SELECT E_{LP}
- GET RESPONSE
- READ BINARY

Basic Infos

- VERIFY CHV

User Authentification

- STATUS
- SELECT EF_{SST}

- GET RESPONSE
- READ BINARY

Services disponibles

SIM et ME en action (2)

- TERMINAL PROFILE
- SELECT MF
- SELECT EF $_{ICCD}$
- GET RESPONSE
- READ BINARY
- SELECT DF_{GSM}
- SELECT EF_{IMSI}
- GET RESPONSE
- READ BINARY
- SELECT EF_{AD}
- GET RESPONSE
- READ BINARY

- SELECT EF_{LOCI}
- READ BINARY

Identif+ Paramètres com

SIM et ME en action (3)

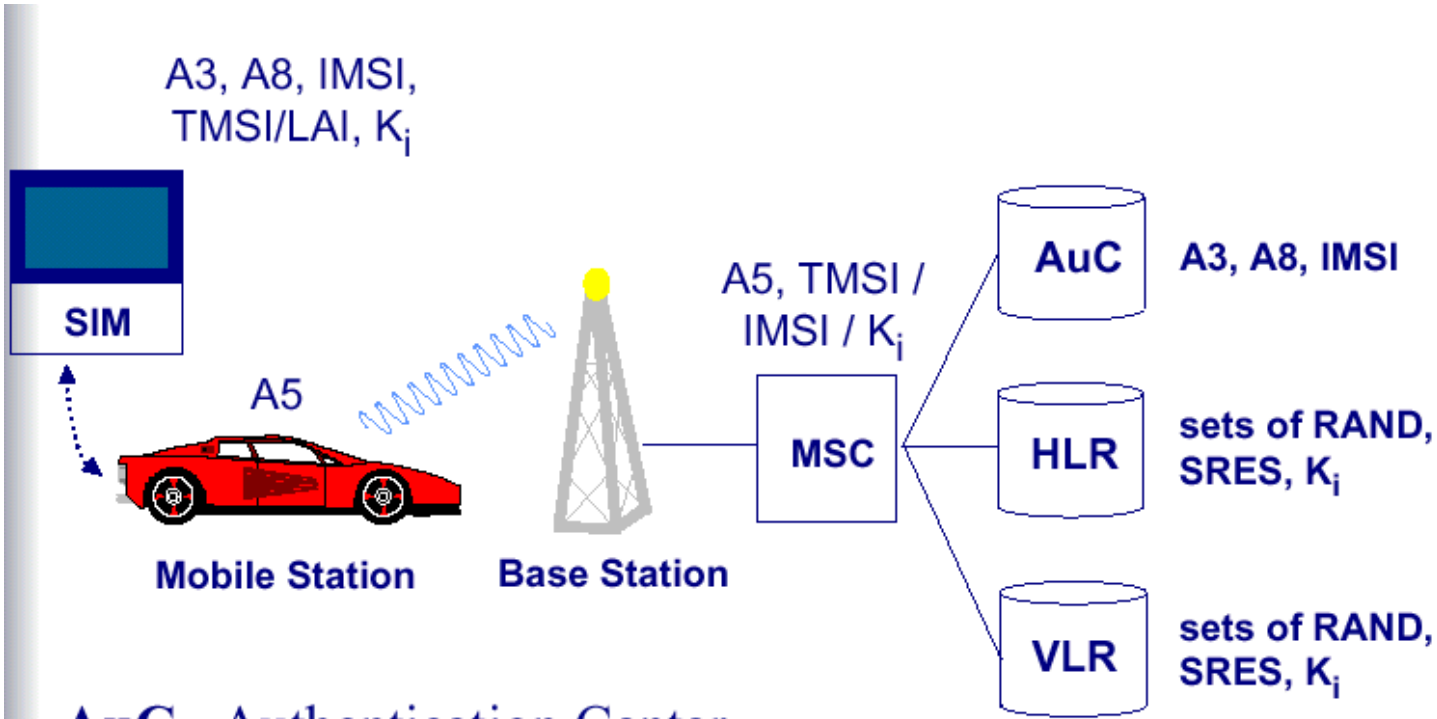
- SELECT EF_{KC}
- READ BINARY
- SELECT EF_{BCCH}
- READ BINARY
- SELECT EF_{FPLMN}
- READ BINARY
- SELECT EF_{HPLMN}
- READ BINARY
- SELECT $DF_{TELECOM}$
- SELECT EF_{SMSS}
- GET RESPONSE
- READ BINARY
- SELECT EF_{SMSp}
- GET RESPONSE
- READ BINARY
- SELECT EF_{SMS}
- GET RESPONSE
- Nx READ RECORD

SELECT DF GSM
RUN GSM ALGORITHM
GET RESPONSE COM
SELECT EF_{KC}
UPDATE BINARY
SELECT EF_{LOCI}
UPDATE BINARY
SELECT EF_{BCCH}

UPDATE BINARY
SELECT EF_{LOCI}
UPDATE BINARY Off
SELECT EF_{BCCH}
UPDATE BINARY

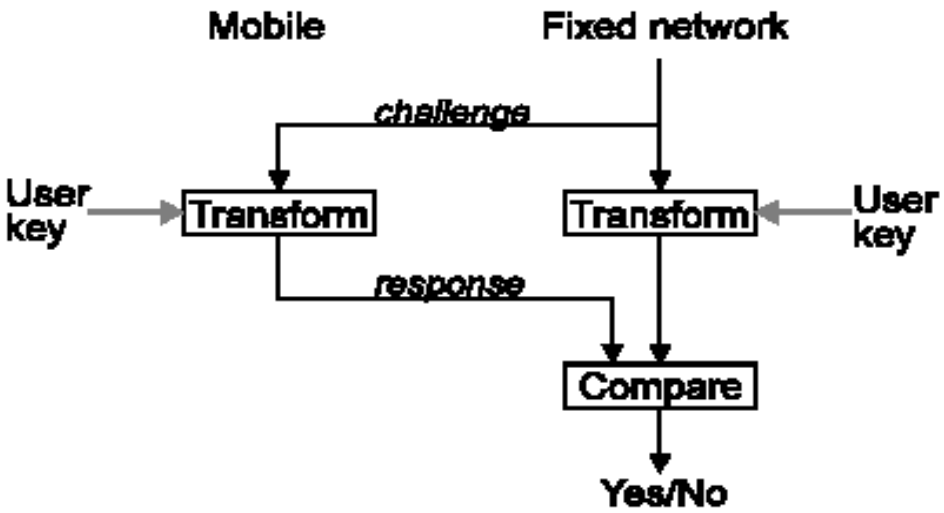
axalto

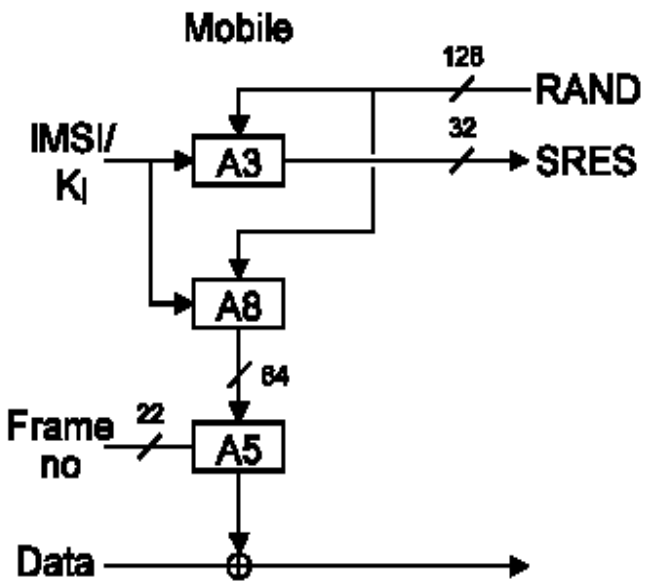
Distribution de la Sécurité dans le réseau GSM



AuC Authentication Center
HLR Home Location Register
VLR Visitor Location Register
MSC Mobile Switching Center

Principe d'authentification dans le réseau GSM





- BS transmet un challenge de 128-bit RAND
- Le MEs retourne une réponse signée de 32-bit SRES via A3
- RAND et K_i sont combinées via A8 pour donner une clé de 64-bit pour l'algorithme
- Les trames de 114-bit sont chiffrées en utilisant la clé et le numéro de trame comme entrée de A5

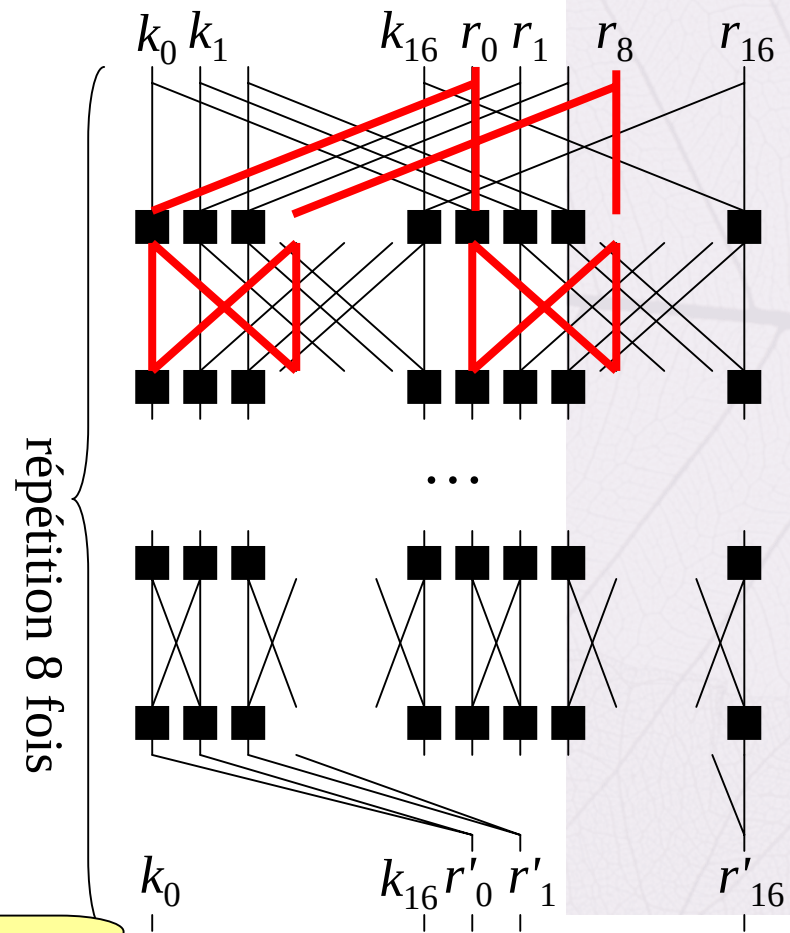
Sécurité du réseau GSM

- La sécurité du réseau GSM a été cassée en Avril 1998
  - A3/A8= COMP128 V1 est faible, permet d'extraire IMSI et K_i
    - Accès direct au SIM (clonage du téléphone mobile)
    - Requêtes OTA au ME
  - Certains types de cartes ont été modifiées depuis pour limiter le nombre de requêtes COMP 128
- De nombreux pays ont été pourvus d'une version affailblie de l'algorithme A5, dite A5/2:
  - Sécurité du A5/1 : Brisable en temps réel avec 2^{40} précalculs
  - Sécurité du A5/2: Aucune (cassable en 5 cycles d'horloge);

Principe de Cryptanalyse du COMP128

- Principes
  - Nombreuses itérations (5*8)
  - L'application de génération des clefs

    $f_k : r \mapsto r'$ est appliquée 8 fois
  - Chaque round est peu efficace:
  - Les bytes i,i+8,i+16,i+24 à la sortie du second round dependent seulement des bytes i,i+8,i+16,i+24 de l'entrée de COMP128
- Génération de collisions!
  - Tentative: Modification simultanée de r_0 and r_8, et recherche de collisions internes [BGW98]



Ca marche!

Deuxième exemple: la norme EMV pour les systèmes de paiement

axalto
A Schlumberger company

Agenda

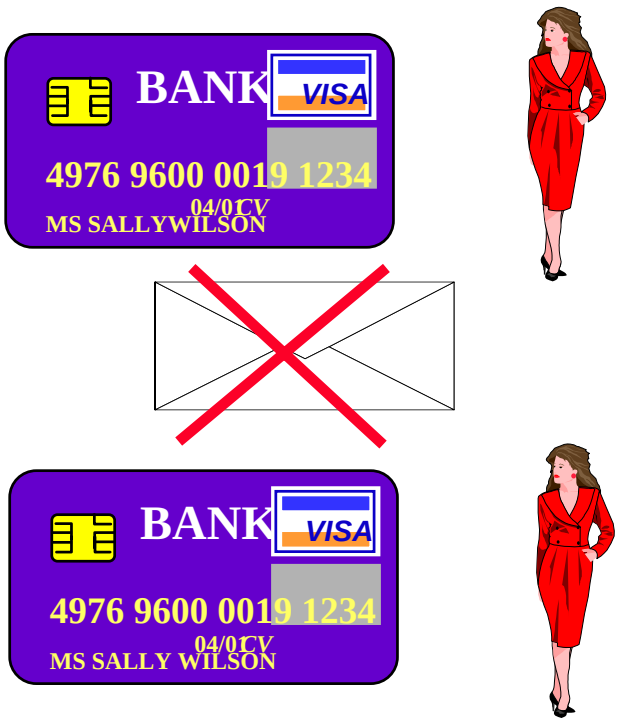
- Pourquoi EMV?
- Un bref rappel sur EMV
- Les apports sécuritaires
- Traitement des données sécuritaires par les Réseaux Cartes VISA/MCI

Pourquoi EMV - Le “Business Case”

- Les objectifs fondamentaux
  - Fraude galopante
  - Interopérabilité
  - Coût des télécoms
  - Dématérialisation des transactions
- Répondre à de nouveaux contextes et créer de la valeur
  - E-commerce
  - Banque en Ligne
  - Services à valeur ajoutée
- Etat de la Technologie:
  - Obsolescence de la technologie piste magnétique (35 ans d’age)
  - Maturité de la technologie puce (25 ans d’existence)

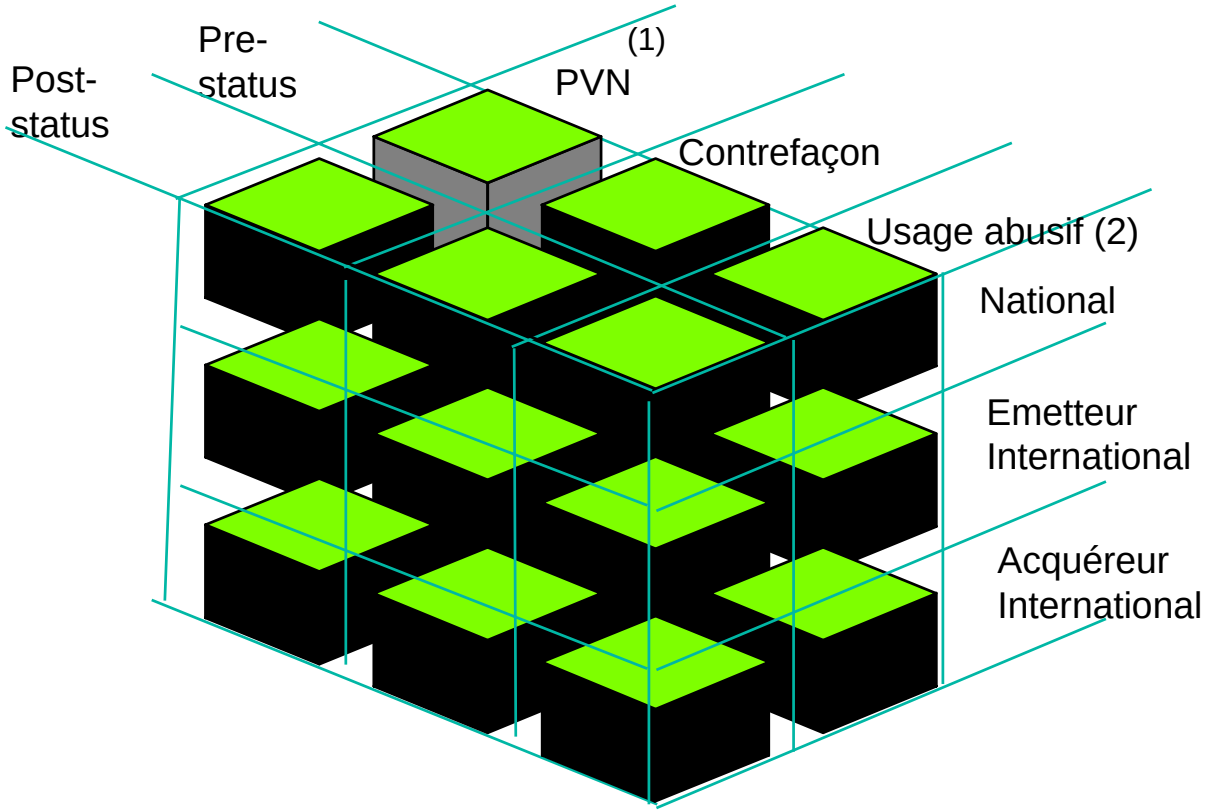
axalto

Types de fraude



- Perdue
- Volée
- Non reçue
- Contrefaçon
- Usage abusif
- Carte-non-présente

Les dimensions de la fraude



(1) Perdue, Volée, Non Reçue

(2) Risque Crédit

Une brève introduction à EMV

- EMV – specifications globales pour les cartes, terminaux, & applications, developpées par Europay, MasterCard & Visa, pour assurer le bon fonctionnement et l'interopérabilité des transactions puce pour les applications de débit et de crédit.

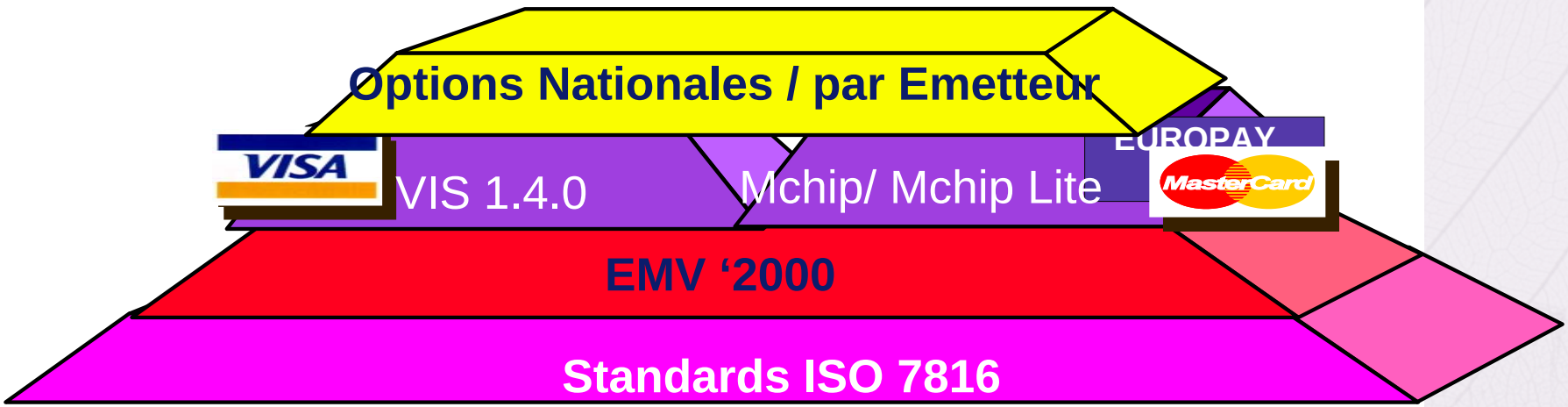
- Interopérabilité
  - N'importe quel terminal de paiement peut accepter des cartes originaires de n'importe quel Réseau de cartes.
  - Tout carte émise dans un pays peut être utilisée dans d'autres pays.
- Déployée en UK
- Pilotes dans une vingtaine de pays
- Base du prochain système bancaire français

axalto

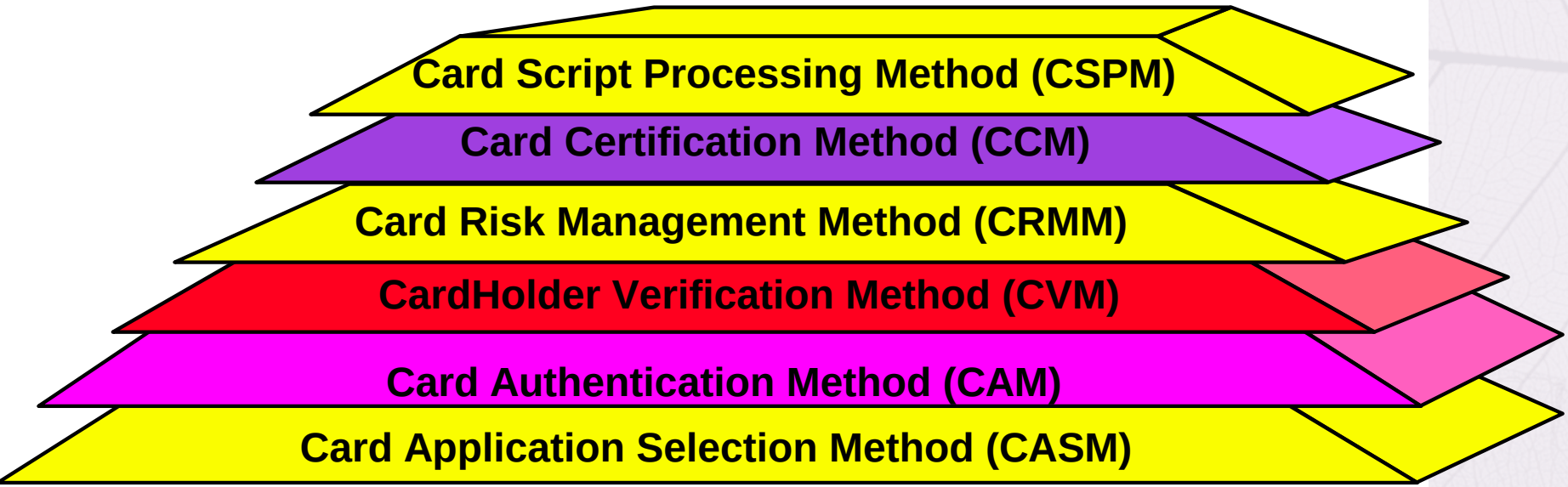
Une brève introduction à EMV (suite)

Ex: France: Droit au Rejet

Ex: Israel : Le terminal est programmé pour éviter le blocage de la carte après 3 essais alors que 5 essais sont possibles



Une brève introduction à EMV (suite)



Card Authentication Method

- Objectif
  - Eliminer la contrefaçon carte
  - Rendre la duplication carte difficile voire impossible
- La Méthode
  - Le terminal vérifie les cryptogrammes SDA / DDA / CDA

CAM :
3 niveaux: SDA / DDA / CDA

- 1ier niveau : SDA (EMV 96 et EMV 2000): le terminal contrôle l'authenticité des paramètres de la carte PAN, Date de validité, ... Comme pour B0' (VA / VS).

- 2ième niveau : DDA(EMV 96 and EMV 2000): le terminal contrôle à la fois l'authenticité des paramètres de la carte (PAN, Date de validité, ...) et de la carte elle-même.

- 3ième niveau : CDA (EMV 2000) le terminal contrôle à la fois l'authenticité des paramètres de la carte (PAN, Date de validité, ...) et de la carte elle-même. De plus il contrôle l'authenticité de la décision carte (accord de transaction offline, décision d'aller online)

L’authentification en SDA

La CARTE contient :

Card Issuer Public Key Certificate
PI : Public information
S=F(PI, Issuer Private Key)

Le TERMINAL contient :

Scheme Provider Public Key

- Le terminal lit les différentes informations dans la carte
- Le terminal recouvre la clé publique de l’émetteur
- Le terminal vérifie la signature

L’authentification en DDA

La CARTE contient :

Card Issuer Public Key Certificate
Card Public Key Certificate
Card Private Key

Le TERMINAL contient :

Scheme Provider Public Key

- Le terminal lit les différentes informations dans la carte
- Le terminal recouvre la clé publique de l’émetteur
- Le terminal recouvre la clé publique de la carte
- Le terminal envoie un challenge à la carte
- La carte calcule un cryptogramme d ’authentification à l ’aide de sa clé privée
- Le terminal vérifie la signature

L’authentification en CDA

La CARTE contient :

Card Issuer Public Key Certificate
Card Public Key Certificate
Card Private Key

Le TERMINAL contient :

Scheme Provider Public Key

- Le terminal lit les différentes informations dans la carte
- Le terminal recouvre la clé publique de l’émetteur
- Le terminal recouvre la clé publique de la carte
- Le terminal commence la transaction de paiement et envoie un challenge à la carte
- La carte calcule un cryptogramme (MAC Triple DES) sur les données de la transaction à l ’aide de la clé de transaction et calcule un cryptogramme d ’authentification à l ’aide de sa clé privée
- Le terminal vérifie la signature et recouvre le cryptogramme de transaction

Cardholder Verification Method

- Objectif
  - Eliminer la fraude liée aux cartes perdues, volées, non reçues

- La Méthode
  - Contrôle local du code secret en clair ou chiffré
  - Méthodes en vigueur : signature, contrôle distant du code secret
  - Techniques biométriques

Card Risk Management

- Objectif
  - Limiter l’usage abusif de la carte, l’usage frauduleux des cartes perdues, volées, non reçues
- La Méthode
  - Limiter la fraude par l’analyse du risque par la carte; contrôle de flux en montant et en nombre de transactions, mémorisation de l’activité transactionnelle de la carte pour aide à la décision, etc.
  - Transaction acceptée / refusée off-line
  - Décision de demande d’autorisation

Card Certification Method

- Objectif
  - Avoir une preuve de la transaction
  - Disposer d’une méthode d’authentification carte / émetteur forte
  - Assurer la confidentialité et l’intégrité des commandes de script
- La Méthode
  - Calculer un crypto-certificat vérifiable par l ’Emetteur
  - Disposer d’un mécanisme cryptographique d’authentification forte de type « Challenge / Réponse »
  - Supporter le mécanisme de Secure Messaging (Chiffrement + Intégrité) pour les commandes de Post-modification

Card Script Processing Method

- Objectif
  - Pouvoir modifier le comportement de la carte sur le terrain
  - Post modification des paramètres de la carte
- La Méthode
  - Commandes de Post-modification émetteur vers la carte

Flot de transaction EMV

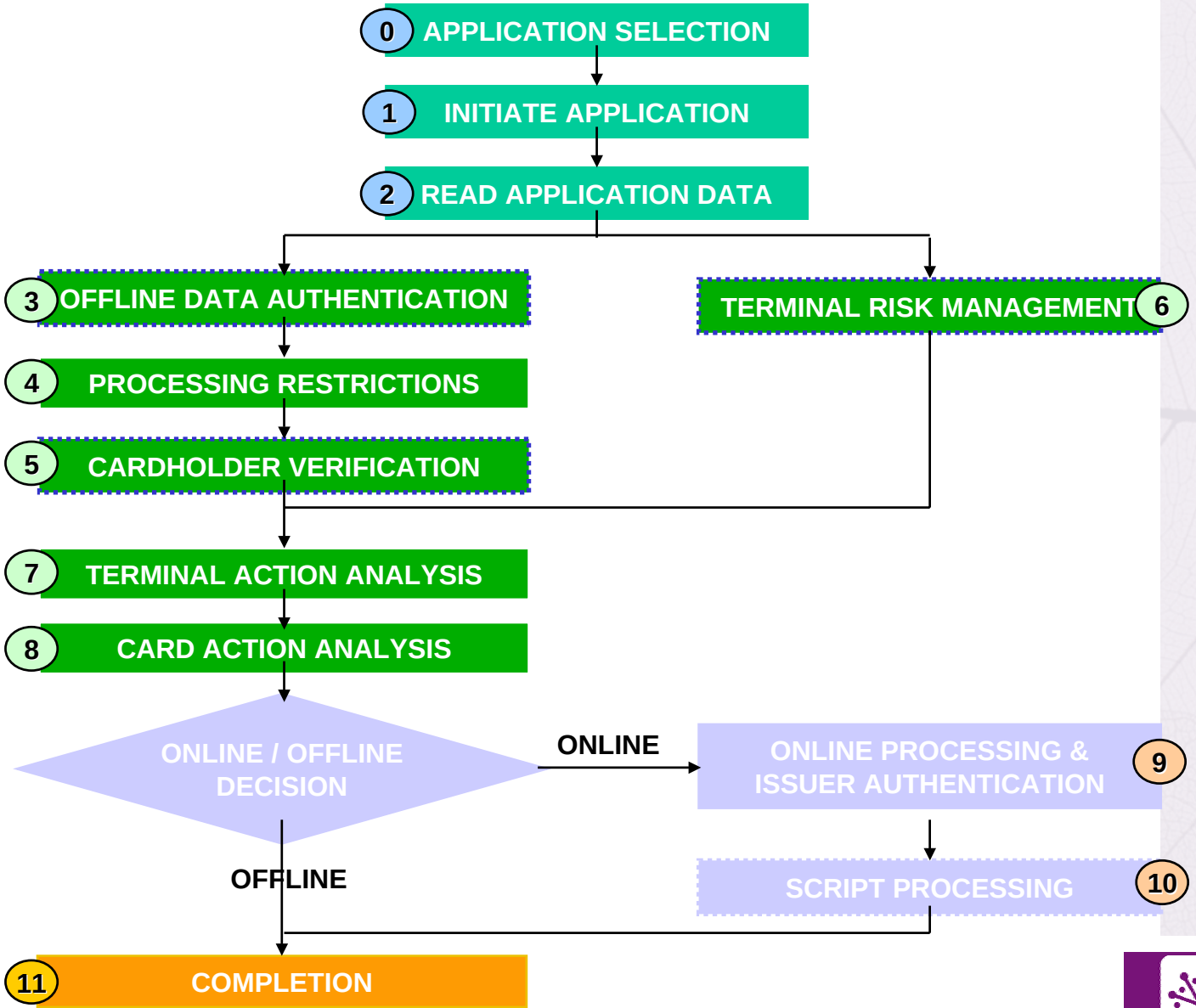
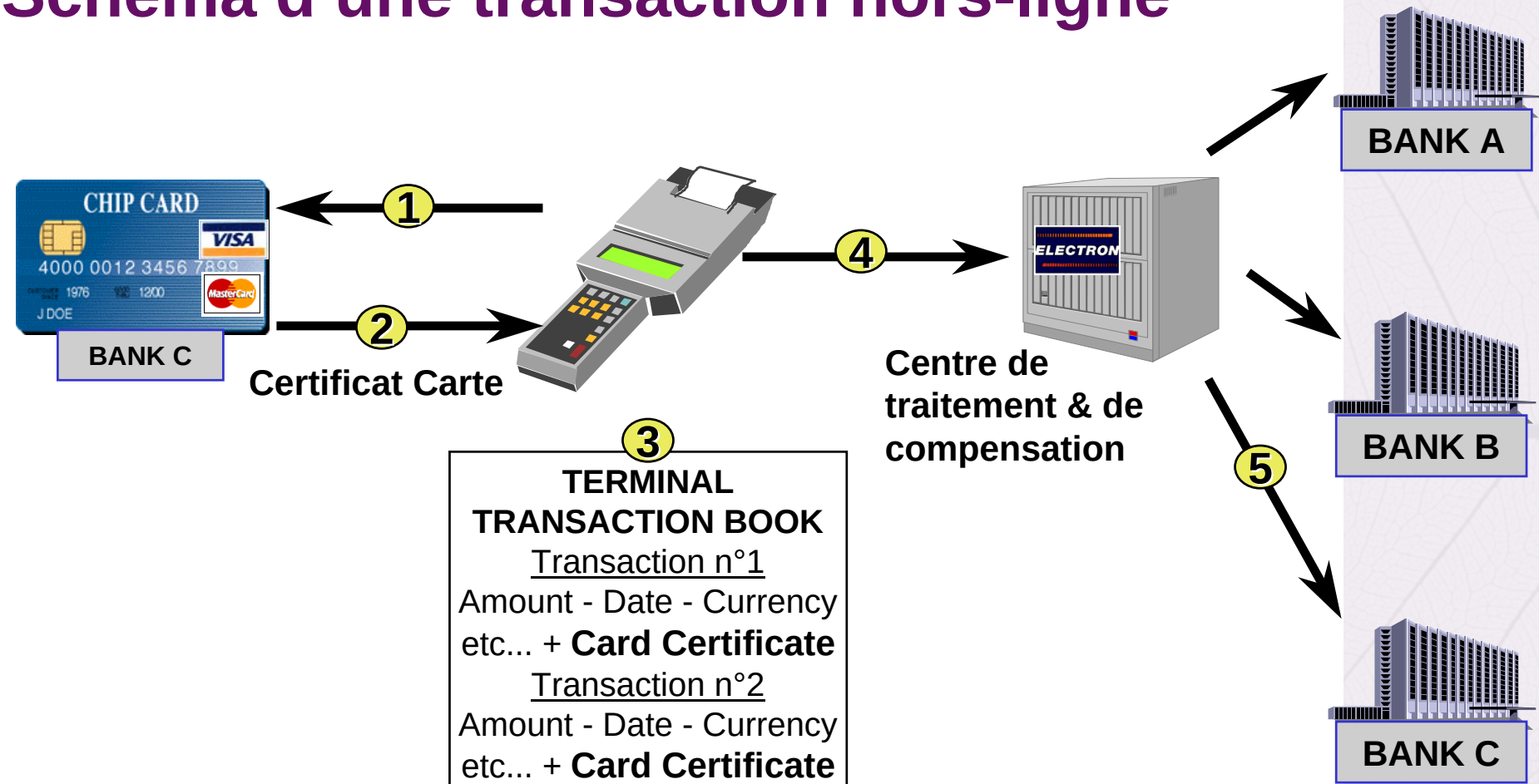


Schéma d'une transaction hors-ligne



3

TERMINAL TRANSACTION BOOK

Transaction n°1
Amount - Date - Currency etc... + **Card Certificate**

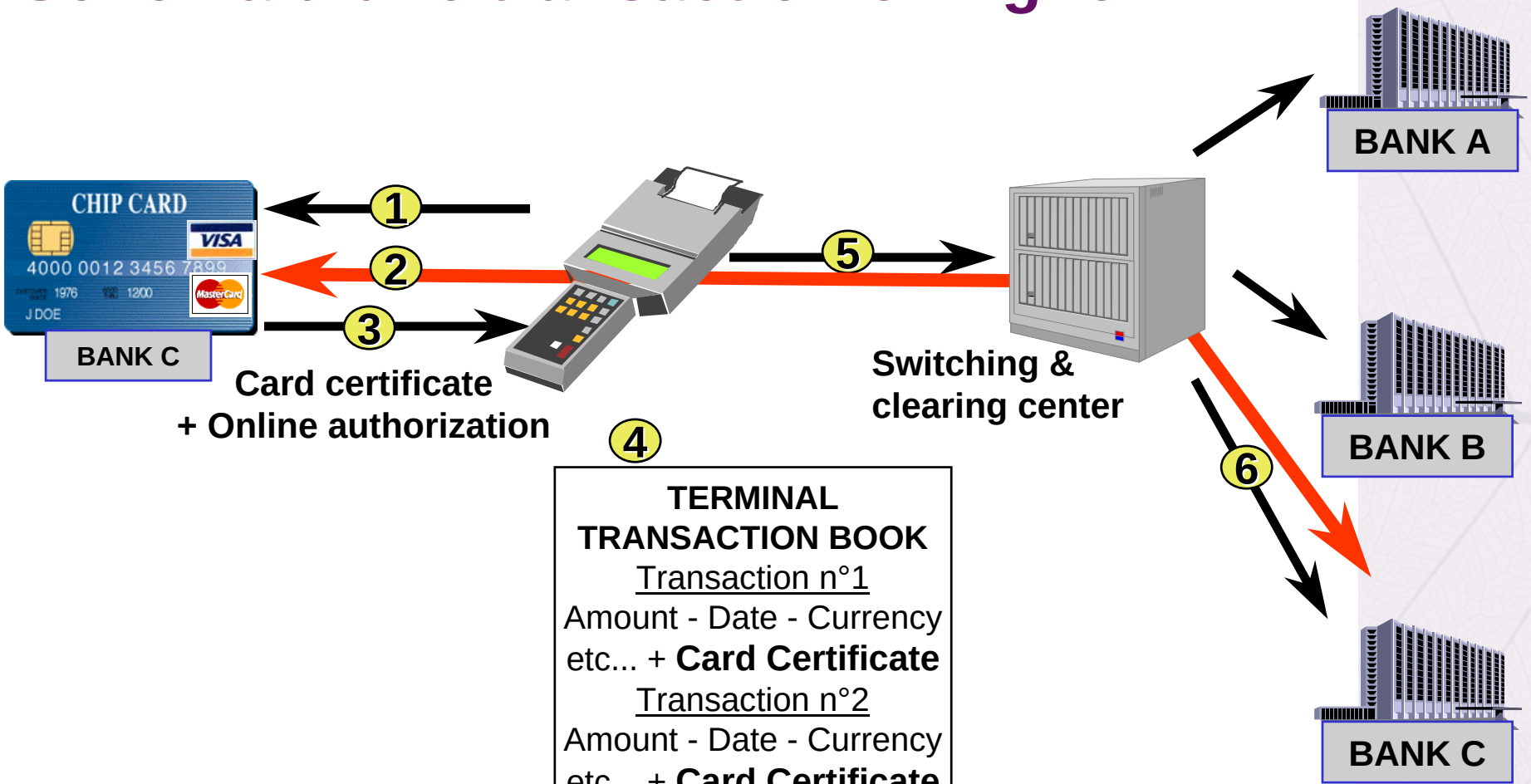
Transaction n°2
Amount - Date - Currency etc... + **Card Certificate**

....

Transaction n°N
Amount - Date - Currency etc... + **Card Certificate**

axalto

Schéma d'une transaction en-ligne



4

TERMINAL
TRANSACTION BOOK

Transaction n°1
Amount - Date - Currency
etc... + **Card Certificate**

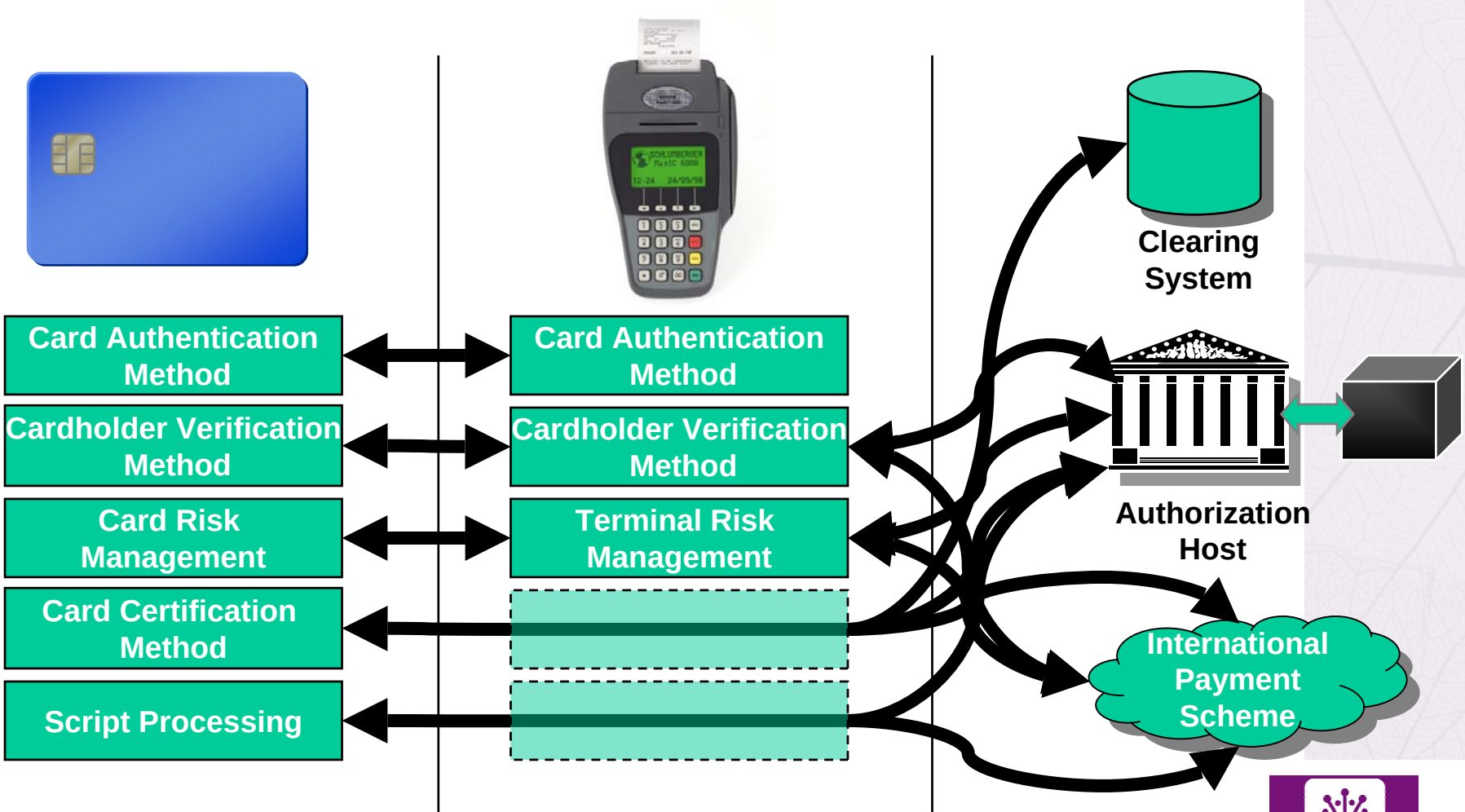
Transaction n°2
Amount - Date - Currency
etc... + **Card Certificate**

....

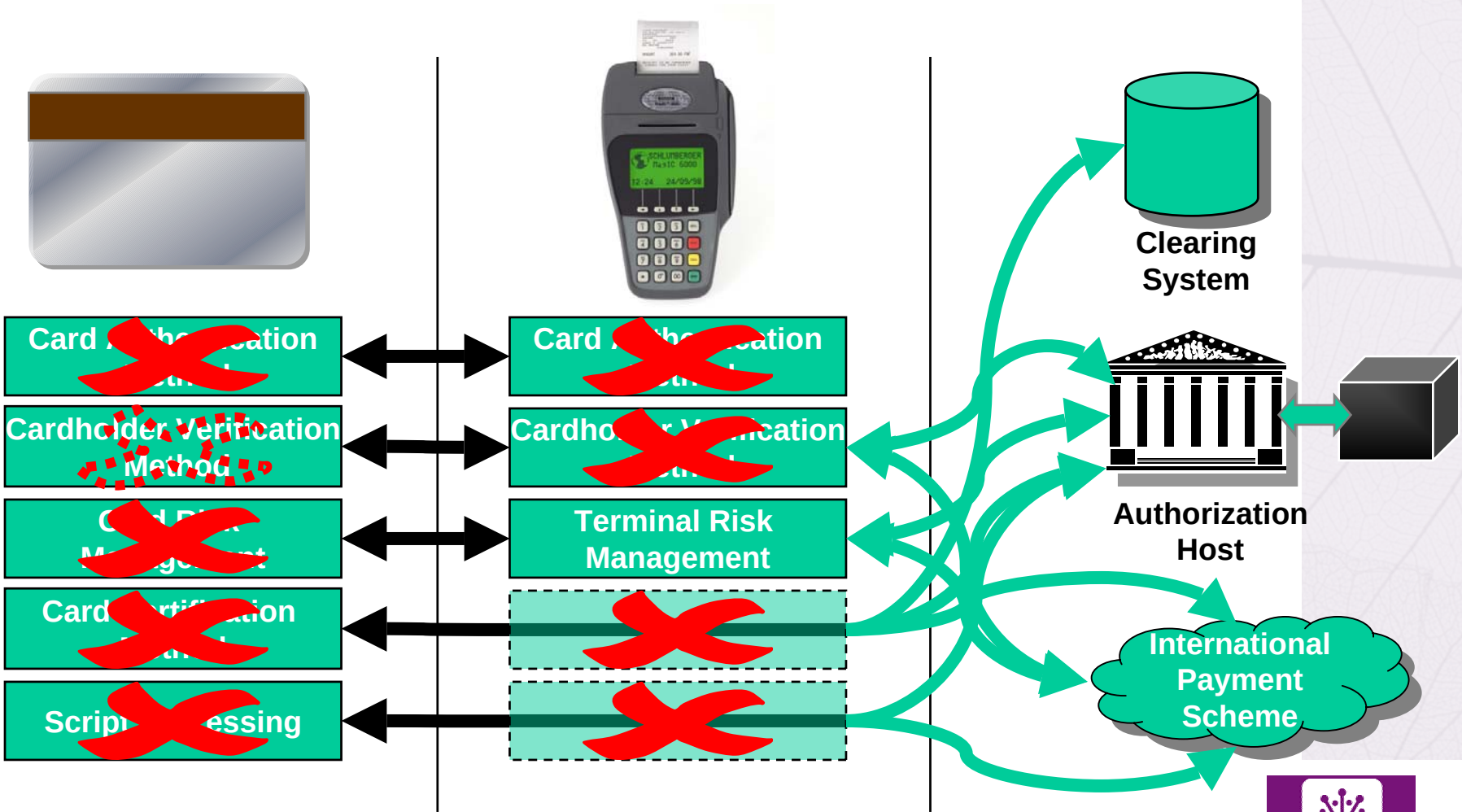
Transaction n°N
Amount - Date - Currency
etc... + **Card Certificate**
+ Online authorization

axalto

Transaction EMV VS ...



Transaction piste



Les types d'attaques

- Pour l'authentification carte
  - Copie d'une carte
    - Facile pour une authentification SDA : Copie de données accessible en lecture
    - Très difficile en DDA, CDA : Nécessite une attaque sur le composant
  - Fausse carte (générer des cartes associées à différents PAN)
    - Très difficile en SDA, DDA, CDA : Nécessite une crypto-analyse de la clé secrète RSA de l'émetteur
- Pour la génération de certificats de transaction
  - Copie d'une carte :
    - Très difficile : Nécessite une attaque sur le composant (clef de certificats émetteur)
  - Fausse carte :
    - Très, très difficile : Nécessite la connaissance de la clé maître de génération des certificats

Les attaques et les réponses

Mode d'authentification	Attaques	Niveau de l'attaque	Réponses
SDA	Copie de la carte	Très facile	Surveillance par l'émetteur du comportement du porteur Vérification des certificats de transaction pour mise à jour de Black List
	Fausse carte	Très difficile	Allongement de la taille de la clé émetteur
DDA	Copie de la carte	Très difficile	Amélioration de la sécurité du chip et de l'OS
	Fausse carte	Très difficile	Allongement de la taille de la clé émetteur
	Substitution de cartes	"Facile" selon conditions	Réponse : CDA
CDA	Copie de la carte	Très difficile	Amélioration de la sécurité du chip et de l'OS
	Fausse carte	Très difficile	Allongement de la taille de la clé émetteur
	Substitution de cartes	Idem copie carte	

La gestion de la sécurité

- Composant sûr et OS évalué
- Gestion des clés secrétes émetteur (renouvellement, revocation, stockage...)
- Contrôle des certificats de transaction
- Gestion de liste noire

Analyse par type de transaction

Type de transaction	Type de terminal	Niveau de risque
Retrait d'espèces	ATM	Très Faible car transaction On-Line
Paiement de proximité On-Line	POS	Très Faible car transaction On-Line
Paiement de proximité Off-Line	POS	Faible. Le marchand peut être acteur dans le processus de vérification.
Automate (ticket, vidéo, essence) On-Line	POS dédié	Très Faible car transaction On-Line
Automate (ticket, vidéo, essence) Off-Line	POS dédié	Fort
Web On-line	Server	Très Faible car transaction On-Line
Web Off-line	Server	Très fort

Carte à puce
Evolution de la technologie

6/21/2005

axalto
A Schlumberger company

Cartes à puce: nouvelles technologies (1/2)

- Hardware
  - Migration vers le 32 bit vers 2003-2004
    - MMU, Cryptographie intégrée, accélérateurs HW (Java)
    - Horizon 2003: CMOS 0.18µ, 8K RAM, 512 KB ROM, 128-256KB EEPROM
  - Support intégré du mode sans-contact
  - Nouvelles interfaces I/O: USB,…
  - Nouvelles technologies de mémoires: Flash, FERAM, Flex
- Architecture
  - Cartes ouvertes multi-applications avec pare-feux HW et SW
  - Prise en compte directe au niveau de la carte de nouveaux protocoles : IEEE 802.11, TCP/IP, Bluetooth
  - Multi-thread voire multitâche

axalto

Cartes à puce: nouvelles technologies (2/2)

- Logiciel
  - Architectures modulaires de type « PC »
  - Evolutions de JavaCard: Sécurité, RMI, Convergence vers Java
  - Téléchargement sécurisé d ’applications (modes interprété ET natif)
- Sécurité
  - Différentiateur essentiel
  - Importance de la méthodologie Critères Communs
  - Approche sécuritaire préventive supportée par des techniques de preuve et modélisation abstraites
- Importance des standards mondiaux
  - ETSI/3GPP, EMV/Global Platform, ISO/IEC 15408,…

Cartes à puce: Evolution du Hardware

Smart Card chip
25 mm2

Commercial Introduction of Technology

** Ethernet Enabled Mode*
*** Near Field Communication*

2010 **500 MHz – Multimode,I/O**
512 MB-1GB Stacked memory
Micro-batteries, sensors, networked peripherals

2008 **300 MH z – Multimode I/O**
56 MB Stacked memory
UWB-Zigbee
Full 2004 PC w/o peripherals

2006 **200 MHz –32 bit+ USB 2.0 +EEM* +NFC****
4 MB-16MB SC/ 128 MB SMMC
RSA 2048 in 10 ms
PDA

2004 **60-100 MHz – 32 bit/NFC****
512KB-1 MB FLASH in single IC
RSA 2048 in 50 ms
Same power as 1997 PC

2002 **50 MHz -32 bit µC-USB 1.0**
16-64 KB EEPROM
RSA 2048 in 300 ms
Same power as 1985 SUN 2, plus DSP

2000 **15 MHz-8 bit 16 bit µC-Contactless**
8-32 KB EEPROM
RSA 2048 in 500 ms
Same power as 1981 VAXStation I, plus cryptography

1998 **7 MHz-8 bit**
32 Kbytes EEPROM
RSA 1024 in 500 ms
Same power as 1979 PDP 11/70, plus cryptography

axalto

Cartes à puce: Evolution du Software

Commercial Introduction of Technology

- 2010 Spontaneous networking
- 2008 Direct IPV6 addresability
- 2007 MPEG2-4 streaming, DRM
  Digital right management
- 2006 Advanced multi-modal biometry
- 2005 Web-service card, ID federation
- 2004 Real Time, multi-thread OS
- 2003 XML/TCP-IP on Card
- 2002 On-Card RMI, single-mode biometry
- 2000 WAP/Internet Microbrowsing (SIM Alliance)
- 1999 Java Card combined with RSA
- 1998 Java Card +GSM Subscriber Identity Module (SIM)

axalto