

Carte à puce et développement Java Card

Damien Sauveron

XLIM (UMR CNRS 7252 – Université de Limoges)

Équipe CRYPTIS

<http://damien.sauveron.fr/>



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

1

1

Damien Sauveron

- Maître de conférences Habilité à Diriger des Recherches
 - Thèse au LaBRI/SERMA Technologies
 - Post doctorat Information Security Group du Royal Holloway University of London
- Membre de l'équipe CRYPTIS du laboratoire XLIM
 - 14 permanents + 14 doctorants
 - 1 salle dédiée à la sécurisation des objets connectés
- Enseignant dans les deux Masters CRYPTIS
 - Première formation BAC+5 en France en sécurité et cryptologie créée en 1986
 - Plus de 750 diplômés
- Président du WG11.2 de l'IFIP « Pervasive Systems Security »
- Membre du DC de l'IFIP « IoT » en charge des aspects sécurité



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

2

2

Plan de l'UE

- La carte à puce
 - Concepts de base
 - Applications
 - SIM/USIM
 - EMV
 - Prépayés
 - Les cartes du futur
 - Sa sécurité
 - Les attaques
- Java Card
 - Architecture générale
 - Développement Java Card
 - Mise en œuvre d'une application malveillante
- Multos
 - IoT Trust Anchor
- IoT
 - Quelques attaques
 - Quelques protocoles / procédures

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

3

3

La carte à puce

Un vrai périphérique sécurisé

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

4

4

Sous plan

- La carte à puce et les concepts de base
 - Qu'est-ce qu'une carte ?
 - Historique
 - Acteurs
 - Quelques chiffres
 - Deux classifications:
 - Par technologie pour la puce
 - Par technologie de communication
 - Chaîne de fabrication
 - Le système d'exploitation
 - Standard ISO 7816
 - Protocole de communication
 - Applications: SIM, bancaire, prépayé
- Les sécurités
 - Au niveau physique
 - Au niveau logicielle
 - Au niveau de l'environnement de production
 - Méthodes formelles & Test
 - Évaluation sécuritaire et certification
- Les attaques
 - Non invasive
 - Invasive
- Les cartes du futur

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

5

5

Et si on se commençait par une vidéo ?

On ne rit pas !!!!

Roland Moreno La carte a puce (prod Le sabre).flv

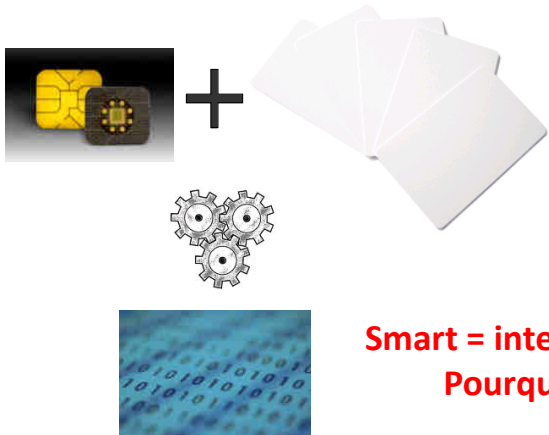
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

6

6

Qu'est ce qu'une carte à puce

- Un morceau de plastique de la taille d'une carte de crédit
- Un circuit électronique capable de manipuler (stocker, calculer, ...) des informations



**Smart = intelligente !
Pourquoi ?**



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

7

7

Historique



La carte à puce,
une innovation française ?

En France : **OUI**

Dans le reste du monde : **NON**

- En 1968, deux Allemands **Jürgen Dethloff** et **Helmut Grötrupp** introduisent un circuit intégré dans une carte plastique
 - déposent un brevet en 1969
- En 1970, **K. Arimura** au Japon dépose un brevet sur la carte à puce.
- En 1971, **P. Castrucci** de IBM dépose aux USA un brevet intitulé Information Card



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

8

8

Historique

- Entre 1974 et 1978, le français **Roland Moreno**, le père de la carte à puce dépose 47 brevets dans 11 pays.

- (crée ensuite la société Innovatron)
- Implication industrielle de Bull et Schlumberger



11/06/1945 – 29/04/2012



Prototype de puce portable imaginé par Roland Moreno

- En 1968, René Barjavel dans la « La nuit des temps »
 - « Chaque fois qu'un Gonda désirait quelque chose de nouveau, des vêtements, un voyage, des objets, il payait avec sa clé. Il pliait le majeur, enfonçait sa clé dans un emplacement prévu à cet effet et son compte, à l'ordinateur central, était aussitôt diminué de la valeur de la marchandise ou du service demandés. »
- Il met au point **les moyens inhibiteurs logés sur la puce** (PIN, compteur, zones, etc.)
 - comparaison interne du code confidentiel ;
 - compteur d'erreurs, qui provoque l'autodestruction de la puce en cas de soumission répétée d'un code faux ;
 - moyens de traitement ;
 - lecture impossible de zones prédéterminées, notamment code confidentiel, clés, etc. ;
 - écriture, modification, effacement impossibles de zones prédéterminées de la mémoire.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

9

9

Historique

- En 1977, le Français Michel Ugon (de la compagnie Bull) propose de mettre les moyens inhibiteurs sur un micro-processeur.



- Mars 1979, CII Honeywell Bull et Motorola :
 - Deux puces: une mémoire 2716 EPROM et un microprocesseur 8 bits 3870.
- Octobre 1981, puce monolithique CII-Honeywell Bull
 - (CP8 : Circuit Portatif des années 80)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

10

10

Historique

- En 1979, la première carte est créée par Bull CP8 (1 ko de mémoire programmable et cœur à base de microprocesseur 6805).



- En 1983, apparition des premières cartes téléphoniques à mémoire
 - commercialisation des premières télécartes par France Télécom fabriquées par Schlumberger



- En 1984, adoption par le G.I.E carte bancaire de la « carte bleue »

- <https://www.planet.fr/banque-carte-bancaire-savez-vous-pourquoi-on-parle-de-carte-bleue.1605398.687347.html>

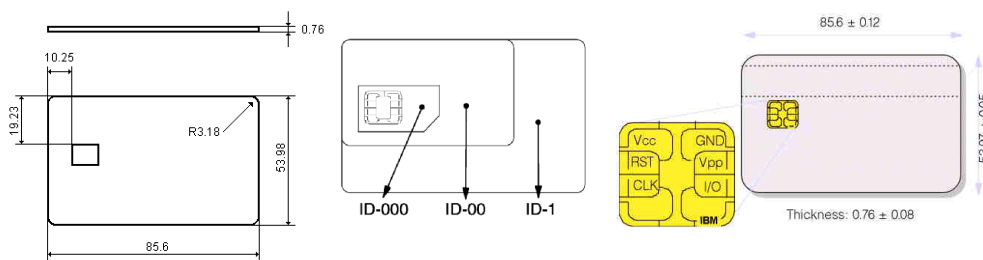


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

11

Historique

- Entre 1984 et 1987, normes ISO 7816 (carte à puce à contact)
 - Objectif : Permettre aux cartes de fonctionner partout dans le monde !**



- En 1989 : Premières cartes GSM pour téléphones mobiles
 - 1992 : Le GSM devient un produit commercial**
- En 1992 : Toutes les cartes CB sont dotées d'un micro-circuit
- En 1994 : Développement de la carte de Sécurité Sociale en Allemagne (près de 70 millions en 1997)
- En 1995 : Début du projet « SESAM Vitale »
 - 1999 : Généralisation de la carte vitale (près de 40 millions)**

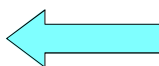
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

12

12

Historique

- En 1996 : Développement important du porte-monnaie électronique
- En 1996 : Premières normes EMV
- En 1997, **apparition des premières Java Cards**
 - 30 ans après des nouveautés dans le monde de la carte. Enfin !!!



- En 2004 : Le système CB passe au standard international EMV
- En 2004 : Premières CB sans contact

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

13

13

La carte à puce aujourd'hui

- Aujourd'hui : **plus de 10 milliards de cartes seront émises en 2018 !**
- **Monétique :**
 - Carte bancaire : Groupement Cartes Bancaires, nouvelles cartes EMV, etc.
 - Porte-monnaie : Octopus, Monéo en France, Proton en Belgique, Geldkarte en Allemagne
- **Identification :**
 - Cartes d'identité nationales (eID en Belgique), E-passeports (août 2006 en France), Passeport biométrique (depuis la fin Juin 2009)
- **Enseignement** (carte d'étudiant et/ou de restauration)
- **Téléphonie mobile** (carte SIM)
- **Secteur médical** (carte Vitale en France, carte SIS en Belgique).
- **Titre de transport** (Passe Navigo à Paris, Oyster à Londres)
- **Sécurité informatique** (authentification forte et signature électronique): carte doté d'un cryptoprocresseur pour la génération des clés et le stockage de la clé privée).

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

14

14

Exemples des Passeports

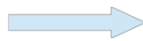
- Depuis 2006 en France :
 - Passeport électronique comporte une puce électronique qui stocke les données personnelles du détenteur : (son nom, sa date de naissance, sa nationalité, son numéro de passeport et la photo numérisée du titulaire).
- Depuis le 15 Juin 2009 :
 - Passeport biométrique : sur une puce RFID, qui permet de lire les informations à courte distance, sont enregistrés - outre les informations personnelles classiques et la photo numérisée - deux empreintes digitalisées des doigts du détenteur (à partir de l'âge de 6 ans).

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

15

15

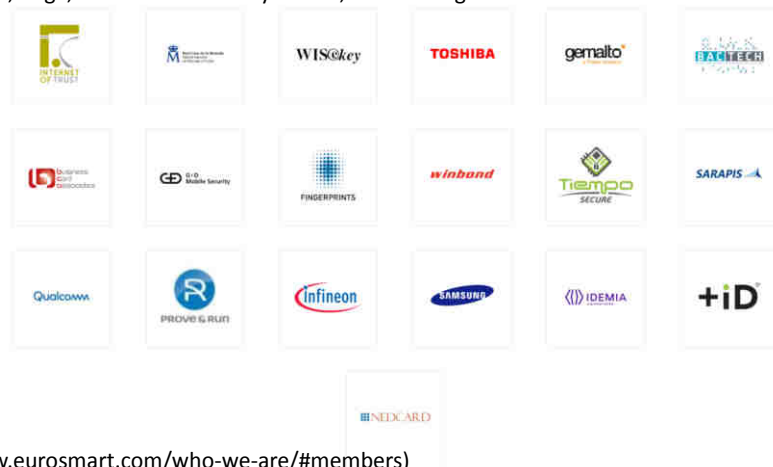
EUROSMART
The Voice of the Smart Card Industry



EUROSMART
The Voice of the Smart Security Industry

- Association fondée en 1994 pour promouvoir la carte à puce
- Membres fondateurs : Bull CP8, Dassault AT, FNMT, Gemplus, Giesecke & Devrient, Oberthur Smart Cards, ODS Oldenburg Datensysteme, Orga, Delarue Cartes & Systèmes, Schlumberger.
- En 2019, members :

2018 members



- Other members (<https://www.eurosmart.com/who-we-are/#members>)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

16

16

Une partie des acteurs de l'écosystème

- Fondateur : Atmel, Infineon, NXP, Samsung, ST Microelectronics, ...
- Encarteur : Feitian Technology, Thales-Gemalto, Giesecke&Devrient, IDEMIA (Oberthur Technology, Safran Morpho), ...
- Bancaire : Carte Bancaire, Carte Bleue, EMV, MasterCard, ~~Monéo~~, VISA, ...
- Téléphonie : Bouygues Télécom, ~~France Telecom~~, Free, Orange, SFR, Vodafone, ...
- TV : Canal+, CanalSat, ~~TPS~~, ...
- Santé : Sécurité sociale, ...
- Évaluation/Certification : CEA Leti, SERMA Safety & Security (ex-SERMA Technologies), SGDN -- ANSSI, Silicomp, Thales, ...
- Laboratoire : CNAM, INRIA, LaBRI, XLIM, ...

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

17

17

Un peu de géopolitique ?

- La société Innovatron est créée par R. Moreno pour exploiter ses brevets.
- En 1979, le géant des services pétroliers Schlumberger entre au capital d'Innovatron pour 23 %, ensuite pour 34 %.
- En 1997, Schlumberger acquiert le concurrent français : SOLAIC, ensuite Bull CP8 en 2001.
- En 2001, Schlumberger externalise l'activité « Cartes à puce » => Axalto
- En 1988, Marc Lassus crée Gemplus en France.
- En 2000, Texas Pacific Group (TPG) entre au capital de Gemplus
- 09/2002-06/2006, Alex J. Mandl est directeur général
 - administrateur de In-Q-Tel (le fonds de capital-risque créé par la CIA)
- Gemplus était numéro 1 mondial jusqu'à sa fusion avec Axalto en 2006.
 - Naissance de Gemalto

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

18

18

Un peu de géopolitique ?

- En 2009, l'État français rachète 8 % de Gemalto via le Fonds stratégique d'investissement
 - Trop tard les américains ont obtenu les technologies de cryptologie de la carte à puce
 - TPG revend ses parts
- Fin 2016, Gemalto était le leader mondial, suivi par Oberthur Technologies et Giesecke & Devrient.
- Il restait aussi Safran – Morpho (résultant de la fusion Sagem-Orga)
- En 2017 : Oberthur Technologies rachète Morpho et devient IDEMIA
- En décembre 2017, Thales rachète Gemalto.

Plus d'info : <http://blogs.univ-poitiers.fr/n-moinet/2015/02/23/de-gemplus-a-gemalto-la-guerre-secrete-continue/>

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

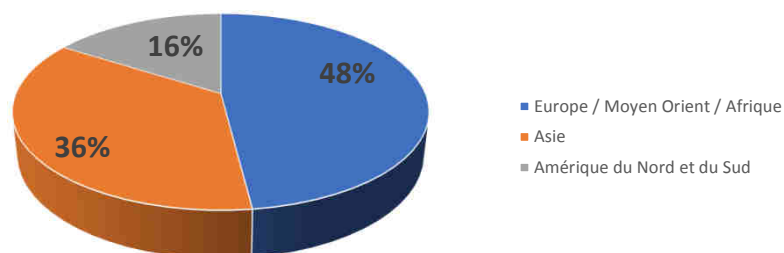
19

19

Répartition géographique du marché

- Chiffres de 2005

Zones Géographiques	à Microprocesseur
Europe / Moyen Orient / Afrique	48,00%
Asie	36,00%
Amérique du Nord et du Sud	16,00%
Total 2005	100,00%



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

20

20

2014

Cartes (Millions d'unités - Mu)	
Secteur	à Microprocesseur (contactless)
Télécommunications	5100
Services financiers / Fidélité	1800 (740)
Gouvernement / Santé	400 (240)
Autres (Transport, TV Payante, accès logique et physique)	410 (70) + (180 pour le transport)
Total prévision 2014	7710 (1230)

Cartes (Millions d'unités - Mu)	
Secteur	à Microprocesseur
Télécommunications	5200
Services financiers / Fidélité	2050 (880)
Gouvernement / Santé	380 (230)
Fabricant d'appareil	190
Autres (Transport, TV Payante, accès logique et physique)	400 (70) + (180 pour le transport)
Total réalisé 2014	8220 (1360)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

21

21

Prévisions 2015

Worldwide Secure Elements shipment – 2013 & 2014 shipments and 2015 forecasts

(Million of units)

Source: Eurosmart, April 2015

Smart Secure Device (Mu)

WW shipments forecast	2013	2014	2015f	2015f vs 2014 % growth
Telecom	4,850*	5,200*	5400	3,8%
Banking	1,550	2050	2450	19,5%
Government	350	380	420	10,5%
Device Manufacturers	190	190	310	63,2%
Others**	390	400	430	7,5%
Total	7,140	8,220	9,010	9,6%

* Source SIMAlliance

**Others include Transport, PayTV and physical and logical access cards.

Worldwide Smart Secure Contactless market figures – 2013 & 2014 shipments and 2015 forecasts

(Million of units)

Source: Eurosmart, April 2015

Smart Secure Contactless (Mu)

Of which contactless	2013	2014	2015f	2015f vs 2014 % growth
Financial services	590	880	1050	19,3%
Government – Healthcare	200	230	260	13,2%
Transport	160	180	210	16,7%
Others*	70	70	70	0%
Total	1,020	1,360	1,590	17%

*Others include Transport, PayTV and physical and logical access cards.

NFC Secure Elements (Mu)

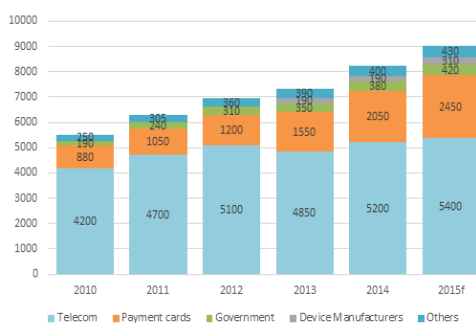
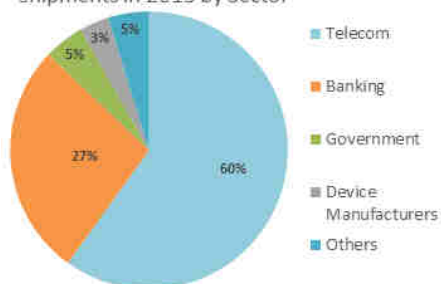
WW shipment forecast	2013	2014	2015f	2015f vs 2014 % growth
NFC Secure Elements	270	350	550	57,1%

22

22

Répartition par secteur et évolution

Worldwide Secure Elements shipments in 2015 by Sector



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

23

23

Quelques chiffres

- Prévision de livraison de cartes à microprocesseur en 2018 (en Millions d'Unités – Mu)

Secteur	2018 (prévision)
Télécommunications	5600
Services financiers	3150 (1500)
Gouvernement / Santé	510 (300)
Fabricant de dispositifs communicants	470
Transport	300 (300)
TV payante	95
Autres (Contrôle d'accès physique et logique)	90
Total	10215 (2100)

() volume en sans contact

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

24

24

Plusieurs classements possibles

les cartes à mémoire
versus
les cartes à microprocesseur

les cartes à contact
versus
les cartes sans contact
versus
les cartes dual-interface

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

25

25

La carte à mémoire

- Premier modèle de cartes à puce
 - Majorité des cartes vendues dans le monde en 1999

1999 Card Shipments figures per sector

- Strong increase in microprocessor cards (+60%)
- Slight progression of memory cards (+14%)

- Elle possède :
 - une puce mémoire de 1 à 16 ko
 - une logique câblée non programmable
- Avantages :
 - sa technologie simple
 - son faible coût (<1€)
- Inconvénients :
 - sa dépendance vis-à-vis du lecteur de carte
 - « assez » facile à dupliquer

	Memory (MU)	Microprocessor (MU)
Banking		108
Healthcare	27.5	30
Telecom	913	200
Transport	40.5	3
Pay TV / IT	1	29
Others	49	28
Total 1999	1031	398
	1429	

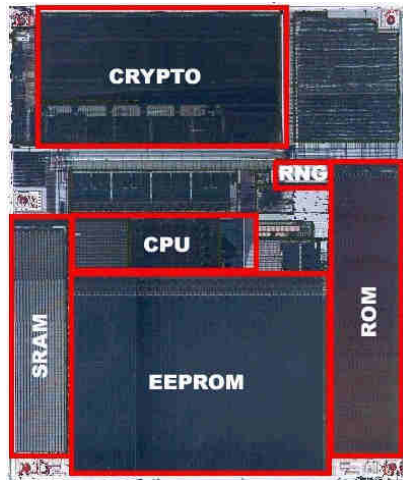
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

26

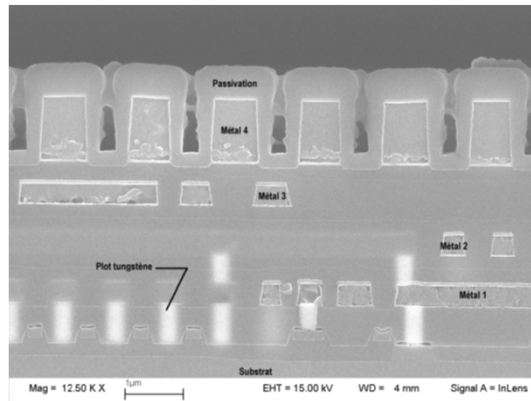
26

La carte à microprocesseur

- Taille de la puce : $25\text{mm}^2 * 200\mu\text{m}$ ($6\mu\text{m}$)



Vue de dessus



Vue en coupe

- Coût : entre 1€ et 20€ (acceptable pour tant de sécurité).

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

27

27

La carte à microprocesseur

- **Microprocesseur** : 8, 16 ou 32 bits (à architecture CISC ou RISC)

- **ROM** : 32 à 256 ko

- Stocke le système d'exploitation et des données permanentes
- Figée en usine

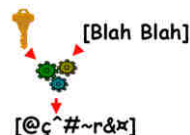
- **EEPROM** : 32 à 256 ko

- Mémoire persistante => stocke les données applicatives
- Problèmes : durée de vie limitée et temps d'accès lent

- **RAM** : 1 à 4 ko

- Mémoire de travail
- Avantages : durée de vie illimitée et temps d'accès rapide

- **Coprocasseur cryptographique**



- **Générateur de nombres aléatoires (RNG)**

Vers de nouvelles mémoires (+ rapide, + grosse) : FeRAM, Flash, MRAM

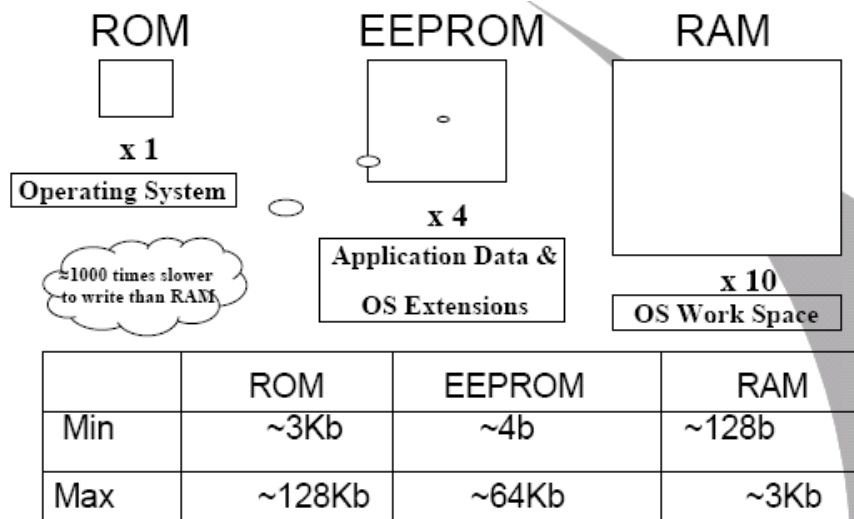
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

28



28

Les différents types de mémoires



EPROM : Write once, read FOR EVER !

– Utilisé pour les zones d'initialisation (e.g. Lock bytes)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

29

29

Les différents types de mémoires

	RAM Statique	EEPROM	Flash-RAM	Fé-RAM (expérimentale)
Persistence	Volatile	Non Volatile	Non Volatile	Non Volatile
Tps Accès Read	0.1 μ s	0.15 μ s	0.15 μ s	0.1 μ s
Tps Accès Write	0.1 μ s	10 μ s	10 μ s	0.4 μ s
Tps Effacement (Reset)	Sans	5 000 μ s	100 000 μ s	Sans
Granularité Effacement	Sans	4 Octets	64 Octets	Sans
Nombre de Cycles garanti	Très grand	10 ⁵ en Ecriture	10 ⁵ en Ecriture	10 ¹⁰ en Ecriture/Lecture
Taille du point mémoire	> 100 μ m ²	> 30 μ m ²	< 10 μ m ²	< 10 μ m ²

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

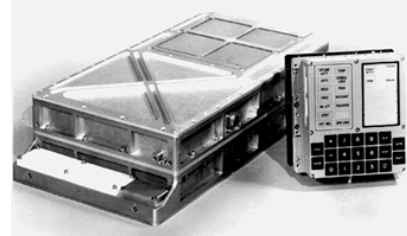
30

30

Comparatif entre la carte et ...

- L'ordinateur de pilotage d'Apollo 11 (1969)

- 2 MHz CPU
- 16 bits
- 2 ko de RAM
- 36 ko de ROM



- Un PC de la fin des années 90

	Carte à puce	PC	Ratio
RAM	1 ko	128 Mo	130000
Stockage	64 ko	6 Go	100000
Connectivité	192 kbits	100 Mbits	500
Microprocesseur	20 Mips	500 Mips	25

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

31

31

Les technologies de communication

les cartes à contact
versus
 les cartes sans contact
versus
 les cartes dual-interface

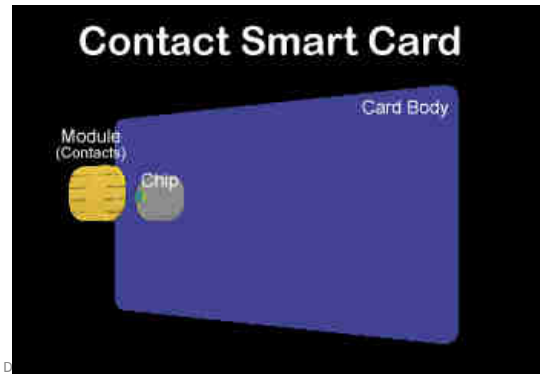
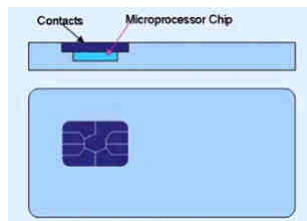
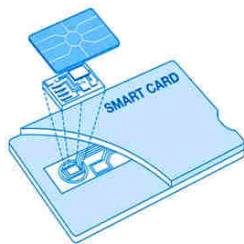
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

32

32

La carte à contact

- Communication série via huit contacts
- Énergie fournie par le lecteur de carte
- Suit le standard ISO 7816
- Problèmes :
 - l'insertion et le retrait sont des facteurs d'usure de la carte
 - l'orientation de la carte dans le lecteur

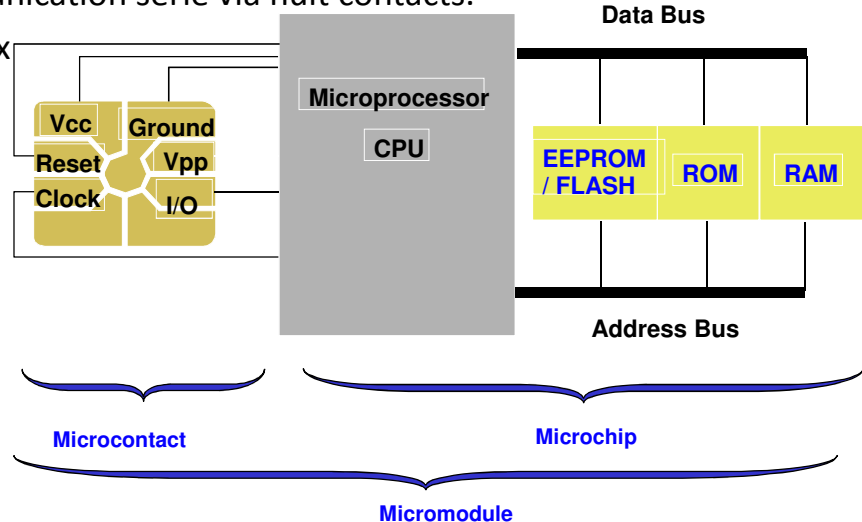


CaP et Dev JC - Limoges - 2019-2020 - D

33

La carte à contact

- Son fonctionnement, l'oblige à être insérée dans un lecteur de carte.
- Elle utilise une communication série via huit contacts.
- 1 ligne I/O semi duplex



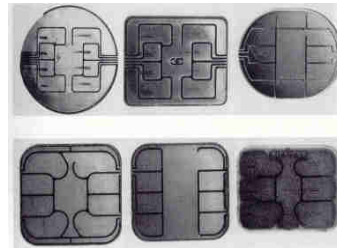
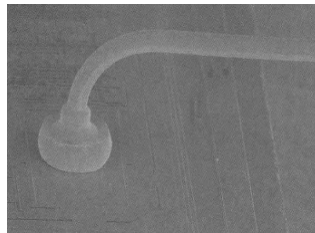
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

34

34

La carte à contact

- Electrical connections between the chip and the module (wire bonding process)
- 8 contacts (C1-C8) but only 6 used
- C6 used as Vpp while EEPROM where not embedding charge pump,
- Supply voltage 2,7v (SIM) to 5,5v (standard TTL) and clock provided by the reader.



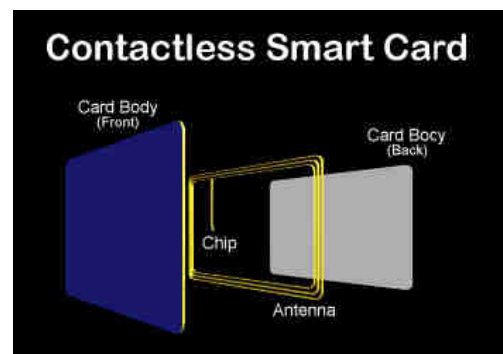
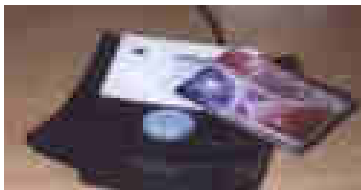
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

35

35

La carte sans contact

- Communication via une antenne dans la carte
- Récupère son énergie d'un couplage capacitif ou d'un couplage inductif
- Suit le standard ISO 14443
- Problèmes :
 - la distance de communication limitée (environ 10 cm)
 - temps de transaction est de l'ordre de 200 ms
 - => limite les données à échanger
 - le coût élevé



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

36

36

Carte sans contact

- Need: modulator, demodulator, anti-collision mechanism, voltage regulator, reset generator and an antenna.
- For data transfer all known digital modulation techniques can be used (ASK, FSK and PSK).
- Standards : close coupling ISO/IEC 10536 (3-5MHz), proximity cards ISO/IEC 14443 (13,56MHz) and Hand Free Cards ISO/IEC 15693,
- Used for public transportation, ski pass, access control, payment with GSM (NFC)...

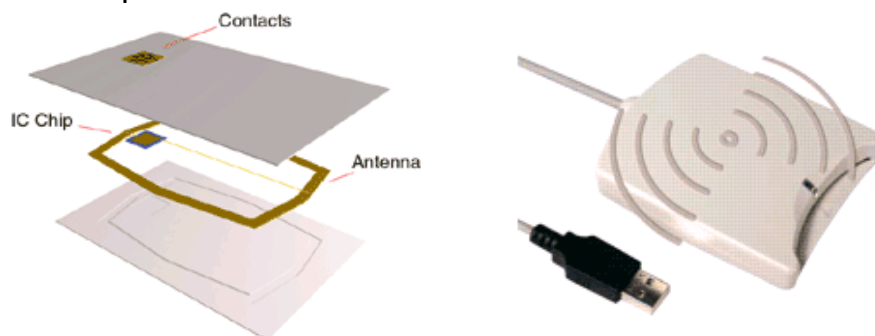
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

37

37

La carte combi

- C'est une combinaison entre :
 - la carte à contact
 - et la carte sans contact
- Ces deux possibilités de communication en font une carte « idéale ».



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

38

38

Quelques spécimens de mon Zoo



Carte à Microprocesseur, 1987-2007

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

39

39

Les différentes formes et interfaces de communication

- Cartes avec contact (Module SIM GSM)
 - ISO 7810, 7816-1, 7816-2
 - USB ou OTG (USB pour mobile)
- Cartes sans contact
 - Plusieurs normes
- Cartes Hybrides
 - combinaison contact et sans contact
- Boutons
 - Produits iButton (bus DS 1-Wire)
 - JavaRing (iButton monté sur bague)
- Dongle
 - Série, parallèle, USB



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

40

40

Les terminaux cartes

- Contact / sans contact
- Simple / complexe (clavier, LCD, multi-emplacement, biométrie, ...)



41

41

Quelques périphériques pouvant intégrer une puce sécurisée

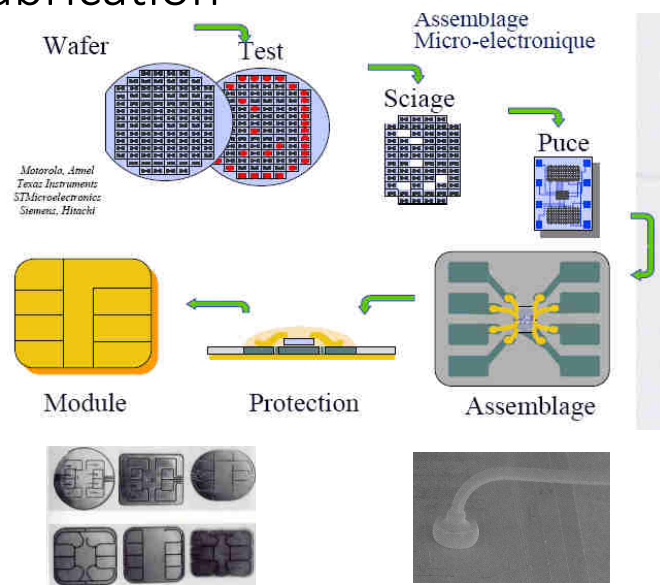


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

42

42

Cycle de fabrication



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

43

43

Cycle de fabrication

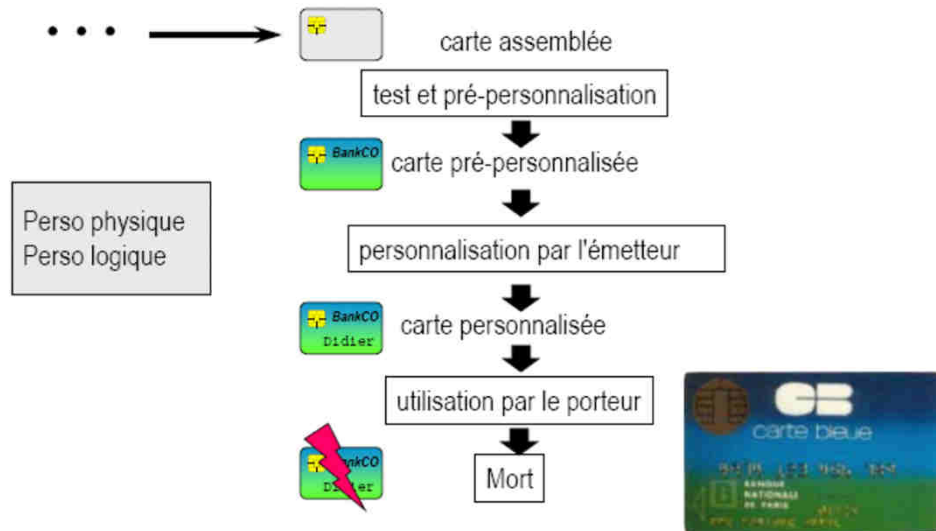


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

44

44

Personnalisation et cycle de vie



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

45

45

Applications

- L'industrie des télécommunications
- L'industrie bancaire et monétaire (BO', EMV)
- Le porte-monnaie électronique (Monéo, Proton, CEPS)
- Le secteur de la santé
- L'industrie audiovisuelle avec la télévision à péage, ...
- Les transports en commun
- Le contrôle d'accès physique de personnes à des locaux, ...
- L'identification : à des sites sur l'Internet, ...
- Les « e-services »
- L'identification gouvernementale (carte d'identité, ...)
- Les applications de fidélité



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

46

46

Le système d'exploitation (OS)

- Gestion des échanges entre la carte et le monde extérieur via un protocole d'échange
- Gestion des accès aux informations et aux fonctions (sélection de fichier, lecture, écriture, etc.)
- Gestion des fichiers et des données à l'intérieur de la mémoire
- Gestion de la sécurité (authentification, signature, confidentialité) de la carte et de la mise en œuvre des algorithmes cryptographiques.
- Gestion des erreurs
- Gestion du cycle de vie de la carte (fabrication, personnalisation, utilisation, fin de vie).

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

47

47

Quelques remarques sur la programmation d'un système d'exploitation pour carte

- No user interface or access to external memory,
- Program written in ROM code (no self modifying techniques allowed),
- No change are possible once the chip is manufactured => quick and dirty programming IS NOT AN OPTION !!!
- Smart Card OS is reliable and robust,
- Design consideration:
 - Persistence...
 - Closely coupled with the hardware

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

48

48

Types de système d'exploitation

- **OS mono-application** (exemple de la carte bancaire B0) : se base sur un fichier maître contenant un certain nombre de fichiers élémentaires. Ces fichiers possèdent des droits d'accès pour la lecture, l'écriture et l'effacement. L'application est associée au système de fichiers.
- **OS multi-application** : présente plusieurs niveaux de répertoires et de fichiers. Chaque répertoire est associé à une application particulière.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

49

49

Quelques exemples de SE pour carte

- Masque dédié : une seule application figée
 - B0' (CB, Sésame Vitale, CPS)
 - Monéo
- Mono Application
 - Cartes systèmes de fichiers (ISO 7816-4) avec authentification du porteur et contrôle d'accès
 - Moteur de base de données (ISO 7816-7)
 - BasicCard
- Multi Applications
 - ~~Wfsc (Windows for smart card)~~
 - Multos
 - Java Card
 - .NET SmartCard

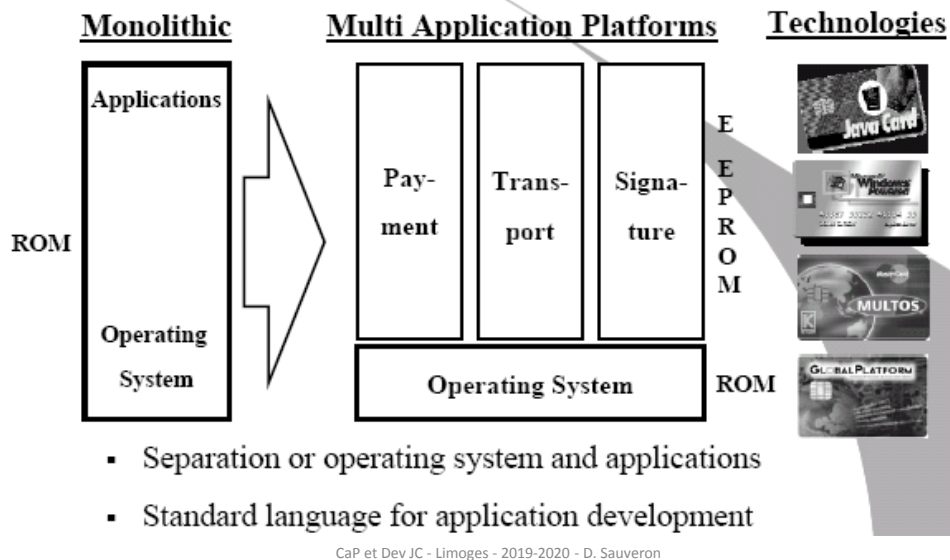


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

50

50

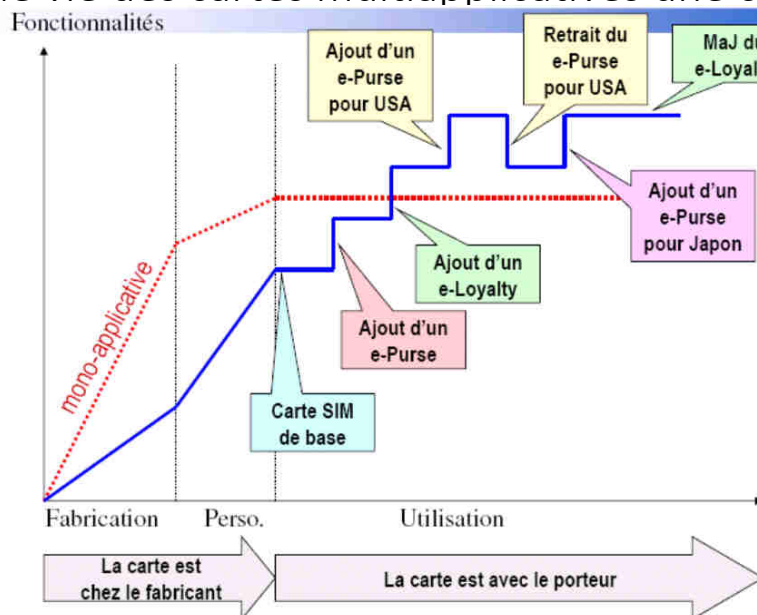
Évolutions



51

51

Le cycle de vie des cartes multiapplicatives une carte dynamique



52

52

Ses avantages ...

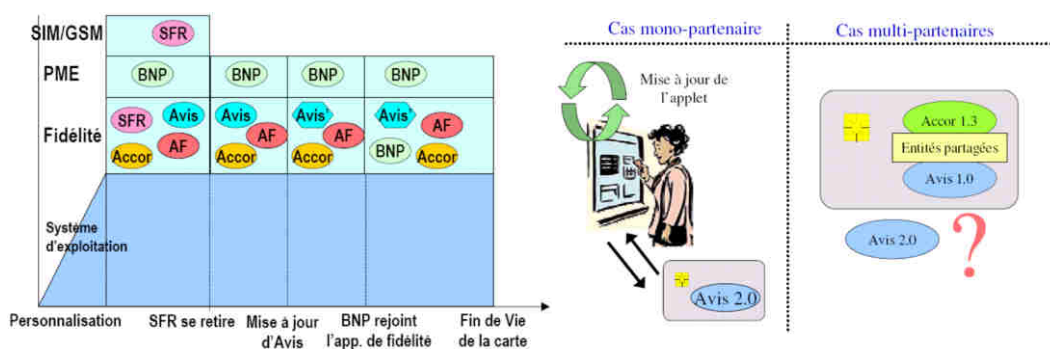
- Le time2market
- Besoins des développeurs
- Sortir la carte d'une programmation « élitiste » :
 - Seuls les programmeurs de longue date savent correctement écrire une application carte
 - Solution : utiliser des langages de programmation courants en informatique (C, Java, Visual Basic)
- Permettre de tester des solutions :
 - portage sur carte plus rapide => possibilité de tests
- Besoins des utilisateurs
 - Se simplifier la vie :
 - utiliser la même carte pour toutes les applications
 - Plusieurs applications sur une même carte
- Pouvoir changer de prestataire sans avoir à recommander une carte
 - Chargement / déchargement d'application

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

53

53

... et ses problèmes lors de la mise à jour



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

54

54

ISO 7816 : le standard !

- Partie 1 : Caractéristiques physiques
- Partie 2 : Dimensions et position des contacts
- Partie 3 : Signaux électroniques et protocoles de transmission
- Partie 4 : Commandes inter-industrie pour l'échange
- Partie 5 : Identificateur d'application
- Partie 6 : Éléments de données inter-industrie
- Partie 7, 8, 9, ...

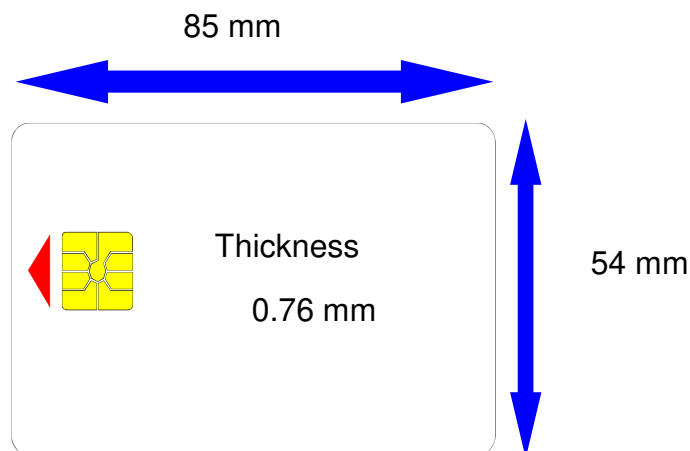
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

55

55

ISO 7816-1

- Définit les caractéristiques physiques : dimensions, flexibilité, résistivité.
- Définit aussi les formats plus petits (SIM)

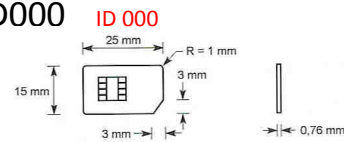
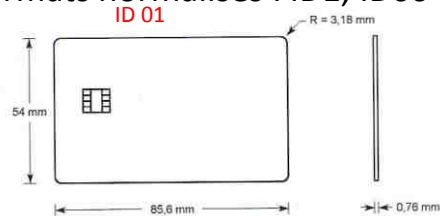


56

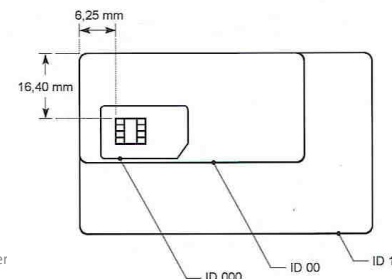
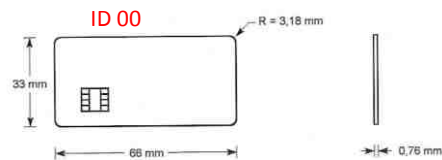
56

ISO 7816-1

- Même si on connaît en général deux formats de la carte à puce
 - Celui de la carte bancaire
 - Celui de la carte SIM
- 3 formats normalisés : ID1, ID00 et ID000



Le fabricant produit une seule taille (ID1), le client final pourra réduire ses dimensions au format ID00 ou ID000 (ex. carte SIM)

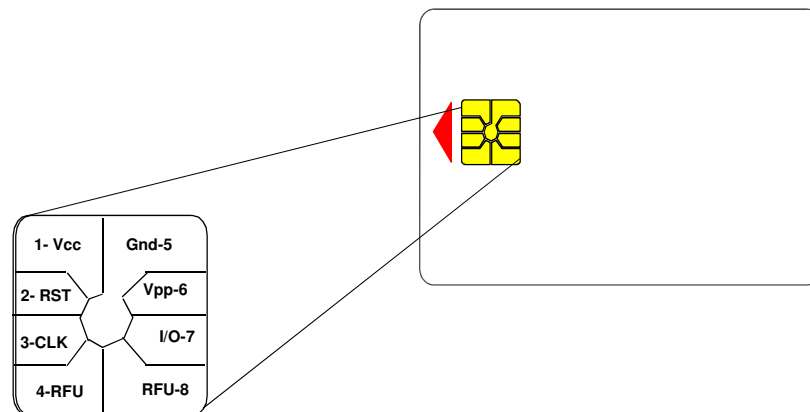


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

57

ISO 7816-2

- Définit les aspects électriques et la position des contacts sur la carte.



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

58

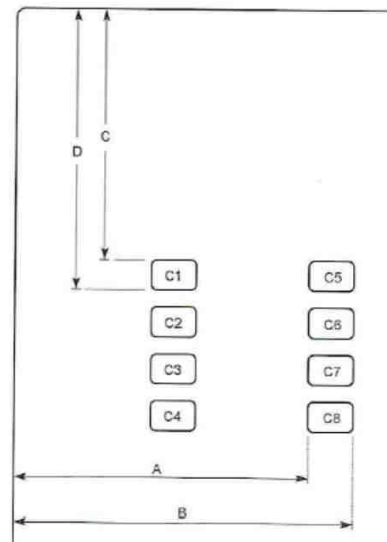
58

Position des contacts sur le support plastique

	A	B	C	D
C1	10,25	12,25	19,23	20,93
C2	10,25	12,25	21,77	23,47
C3	10,25	12,25	24,31	26,01
C4	10,25	12,25	26,85	28,55
C5	17,87	19,87	19,23	20,93
C6	17,87	19,87	21,77	23,47
C7	17,87	19,87	24,31	26,01
C8	17,87	19,87	26,85	28,55

Position ISO 7816

Valeurs en mm



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

59

59

Normalisation AFNOR / ISO

- La position des contacts : position AFNOR et position ISO



Carte ISO

Carte AFNOR

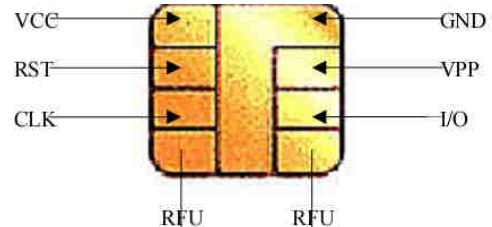
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

60

60

ISO 7816-2

- Vcc: tension d'alimentation positive de la carte fournie par le lecteur
 - (4.75V > Vcc > 5.25V) Vcc=3.3V pour une carte SIM
- RST: c'est le « reset », initialise le microprocesseur (warm reset)
 - cold reset = coupure et rétablissement de l'alimentation
- CLK: signal d'horloge fourni à la carte par le lecteur
 - rythme les échanges de données entre la carte et le lecteur
- RFU (Reserved for Future Use) non utilisés
- GND: masse électrique de la carte



- I/O: utilisé pour le transfert des données et des commandes entre la carte et le lecteur. La communication est half-duplex.
- Vpp: destiné à l'origine à recevoir la tension de programmation des mémoires non-volatiles EPROM. Cette tension était de 25 Volts à l'origine, puis 21 Volts, puis devenue inutile avec l'utilisation d'EEPROM qui possèdent leur propre générateur de tension de programmation sur la puce à partir du 5 Volts (Vcc). **Utilisé pour SWP (full duplex)**

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

61

61

ISO 7816-3

- Cette partie normalise :
 - les protocoles de transmission (TPDU : Transmission Protocol Data Unit) :
 - T=0, protocole orienté octet
 - T=1, protocole orienté paquet
 - T=14, réservé pour les protocoles propriétaires
 - la sélection du type de protocole (PTS : Protocol Type Selection)
 - la négociation des paramètres du protocole (PPS : Protocol Parameter Selection)
- la réponse au reset (ATR : Answer To Reset) qui correspond à la mise en route du prog ROM de la carte
 - Min. 2 et Max. 33 caractères et 5 champs
 - Permet de fixer :
 - Les conventions de codage des octets
 - Le temps de transmission d'un bit
 - La valeur de la tension de programmation
 - Le protocole de communication
 - Un historique qui s'affichera à la mise sous tension de la carte (ex: version de l'OS)
- les caractéristiques électriques comme :
 - la fréquence d'horloge (entre 1MHz et 5MHz)
 - la vitesse des communications (jusqu'à 115200 bauds)

La carte n'est jamais l'initiateur de la communication



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

62

62

Exemples d'ATR

- Carte Santé Vitale
 - ATR = 3F 65 25 00 2C 09 69 90 00
- Carte Bancaire CB
 - ATR = 3F 65 25 08 36 04 6C 90 00
- Carte Verte Monéo
 - ATR= 3B E6 00 FF 81 31 42 45 19 16 01 01 27 B1 37
- Carte Cinema (perimé)
 - ATR = 3B 23 00 35 13 96
- Carte GSM Itineris
 - ATR=3F 2F 00 30 AF 59 02 01 02 80 00 17 0A 0E 83 1E 9F 16
- GXP211_PKIS
 - ATR = 3F 6D 00 00 80 31 80 65 B0 05 01 02 5E 83 00 90 00
- GemSafe
 - ATR= 3B A7 00 40 18 80 65 A2 08 01 01 52
- Schlumberger Palmera
 - ATR= 3B 65 00 00 9C 02 02 06 01
- Cyberflex Access e-gate 32K
 - ATR=3B 75 94 00 00 62 02 02 00 80

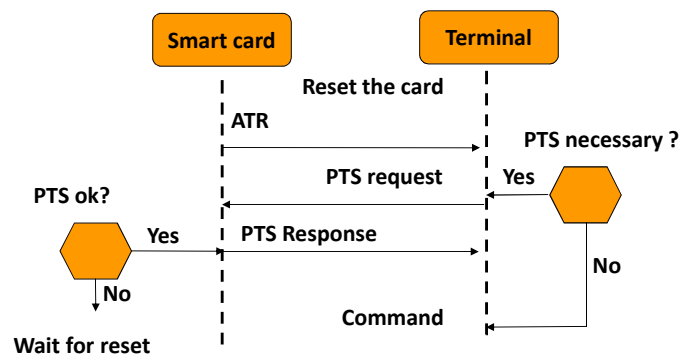
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

63

63

Protocol Type Selection

- Needed only if the terminal wants to modify parameters,
- If the card agrees, it sends the PTS back to the terminal
- Otherwise the terminal execute a reset (warm => protocol change),
- Only one PTS after the ATR.

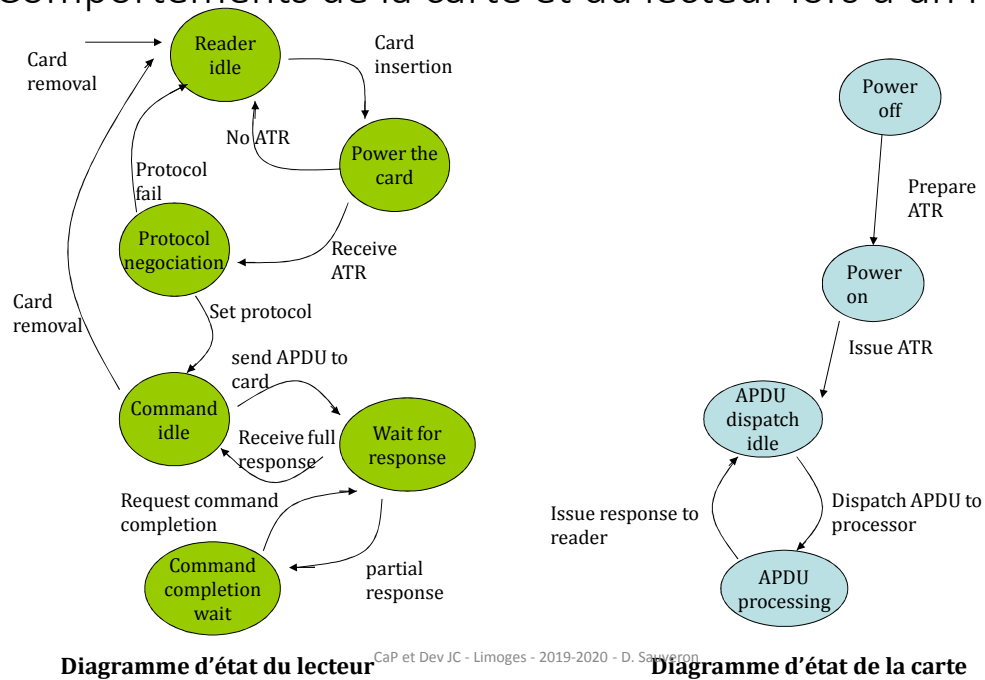


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

64

64

Comportements de la carte et du lecteur lors d'un Reset



65

Transmission protocols

- T=0 most widely used (1989), T=1 block oriented
- T=14 Japan and Germany

Transmission protocol	Meaning	ISO
T=0	Asynchronous, half duplex, byte oriented	7816-3
T=1	Asynchronous, half duplex, block oriented	7816-3
T=2	Asynchronous, full duplex, block oriented, tbs	10536-4
T=14	National functions	No ISO

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

66

66

Transport protocols

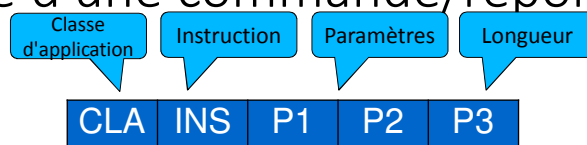
- T=0
 - Byte oriented, Serial transmission (1 start bit, 8 bits data, 1 parity bit, 2 stop bits)
 - Transmission error (parity only) 2 etu mute ("0")
- T=1
 - Block oriented, Header : NAD, PCB, LEN; data : INF, CRC.
 - NAD 3 bits destination address, 3 bits source address
 - PCB define the kind of block
 - I (#block, more) numbered mod 2, more = 1, another block follow
 - R(#block, error) numbered mod 2, next expected block,
 - S specific command (RESYNC, IFS, ABORT, WTX)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

67

67

T=0 Structure d'une commande/réponse



- Commande
 - CLA est normalisé (par exemple FF est utilisé pour le PPS)
 - INS ne doit pas être 9x ou 6x et devrait être paire avant l'édition 3 de l'ISO7816-3 (2006)



- Réponse
 - Les valeurs sont normalisées
 - 90 00 : succès
 - 6E xx : classe inconnue
 - 6D xx : instruction inconnue
 - ...

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

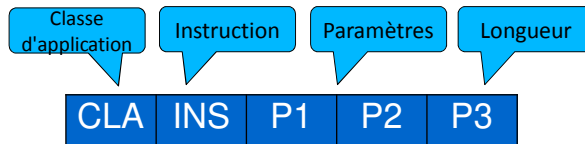
68

68

Protocole T=0 (Transmission semi-duplex de caractères asynchrones)

- **Commande entrante** (envoi des données à la carte)
- **Commande sortante** (récupère des données de la carte)

- Structure de l'ordre



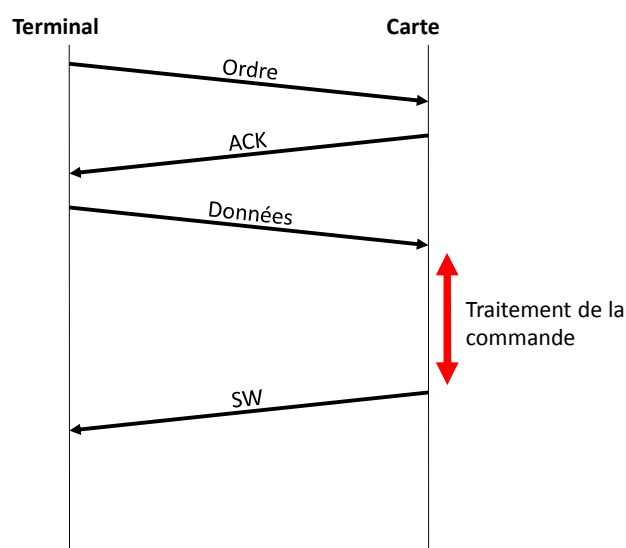
- Octets de procédure

Byte	Value	Action on data transfer	Then reception of
NULL	'60'	No action	A procedure byte
SW1	'6X' (≠'60'), '9X'	No action	A SW2 byte
ACK	INS	All remaining data bytes (if any bytes remain)	A procedure byte
	$\text{INS} \oplus \text{'FF'}$	The next data byte (if it exists)	A procedure byte

The first two editions of ISO/IEC 7816-3 specified the use of two values of ACK (namely, the exclusive-or of the value of INS with '01' and 'FE') to control the deprecated use of contact C6 (see 5.1.1). These two values are deprecated.

69

T=0 : Commande entrante

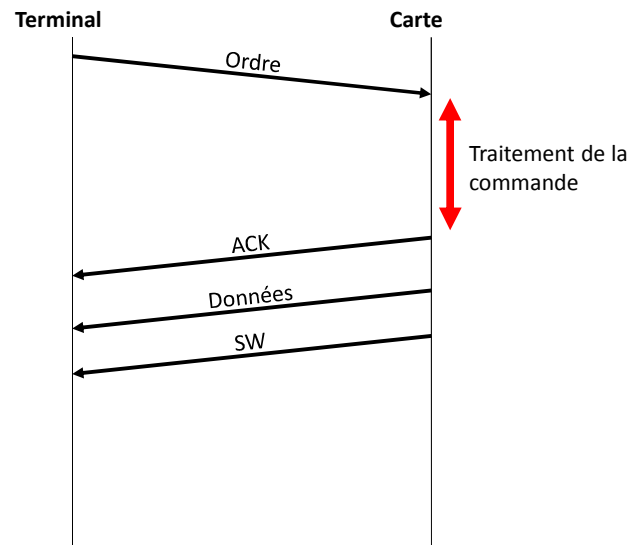


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

70

70

T=0 : Commande sortante



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

71

71

ISO 7816-4

- ISO 7816-4 vise à assurer une interopérabilité.
 - But : indépendance des applications par rapport aux couches physique et liaison
- Il spécifie :
 - le contenu des messages entre la carte et le lecteur
 - les commandes
 - les réponses
 - les structures des fichiers et de données :
 - l'accès à ces données
 - l'architecture de sécurité
 - la sécurisation des communications
- Le protocole APDU
 - Protocole de niveau application.
 - La commande APDU (C-APDU) : émise par le CAD vers la carte

Entête obligatoire				Corps optionnel		
CLA	INS	P1	P2	Lc (1-3 octets)	Champs de Données (Nc octets)	Le (1-3 octets)

- La réponse APDU (R-APDU) : transite de la carte vers le CAD

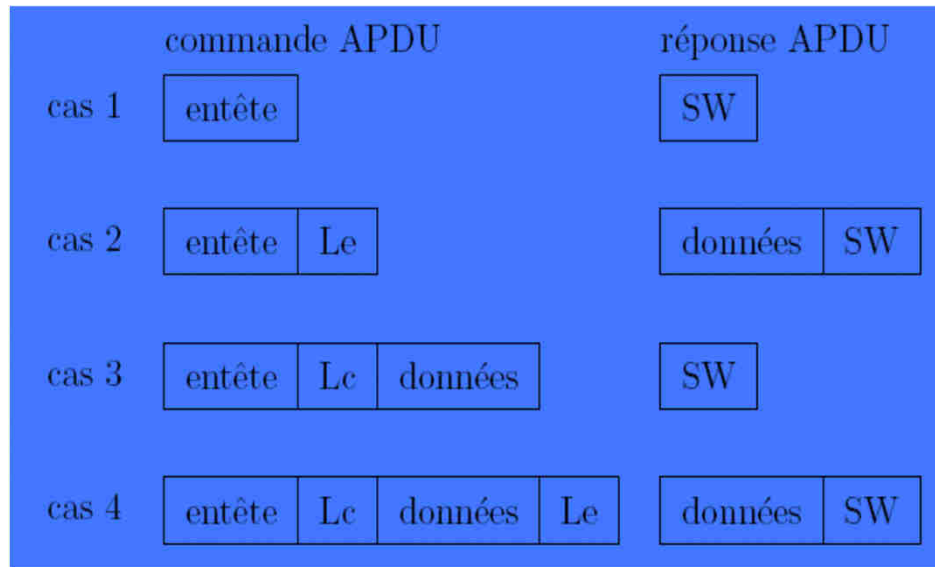
Corps optionnel		État obligatoire	
Champs de Données (Nr octets <= Ne)		SW1	SW2

72

72

ISO 7816-4

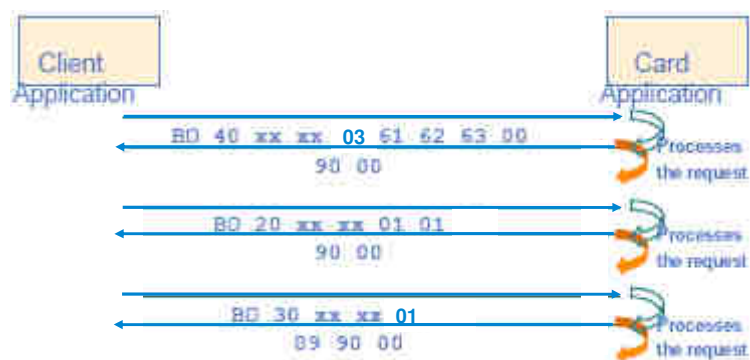
- Les différents cas d'échanges APDU :



73

73

Exemple de communication APDU



- La transmission d'APDU en T=0 ou T=1 est détaillée dans les annexes de l'ISO7816-3

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

74

74

Complexité

- Afin de pouvoir rendre transparent la communication APDU vis à vis des différents protocoles sous-jacents, on verra en Java Card qu'il sera nécessaire d'appeler les méthodes de communication dans un certain ordre.
- En effet, **T=0 par exemple** ne permet pas d'avoir des commandes entrantes **et** sortantes ...

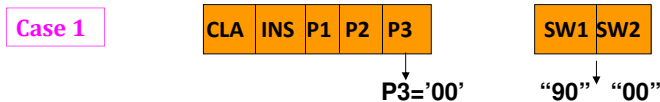
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

75

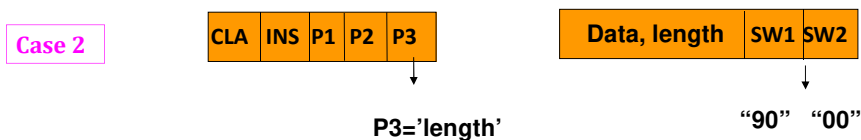
75

APDU Command-Response

- Different configurations **in T=0**
 - Command without data, response without data



- Command without data, response with data length known



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

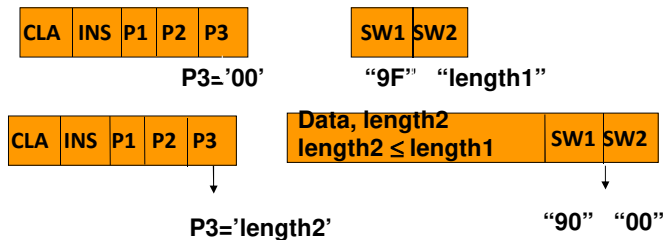
76

76

APDU Command-Response

- Command without data, response with data, length unknown

Case 2b



- Command with data, response without data

Case 3



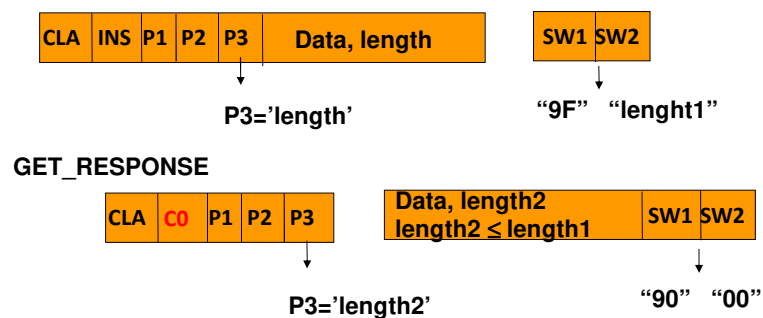
77

77

APDU Command-Response

- Command with data, response with data length known or unknown

Case 4



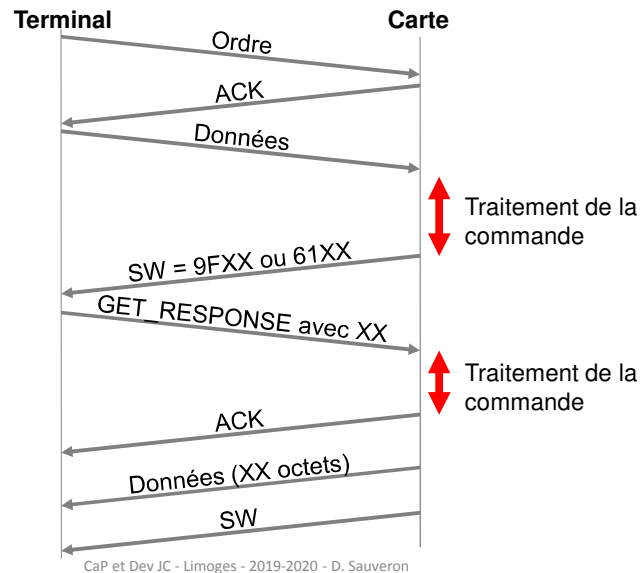
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

78

78

Comment est traité le **cas 4** APDU (données entrantes et sortantes) ?

- Utilisation de la commande spéciale GET_RESPONSE



79

79

CLA Class byte

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	x	-	-	-	-	Command chaining control
0	0	0	0	-	-	-	-	-- The command is the last or only command of a chain
0	0	0	1	-	-	-	-	-- The command is not the last command of a chain
0	0	0	-	x	x	-	-	Secure messaging indication
0	0	0	-	0	0	-	-	-- No SM or no indication
0	0	0	-	0	1	-	-	-- Proprietary SM format
0	0	0	-	1	0	-	-	-- SM with command header not processed
0	0	0	-	1	1	-	-	-- SM with command header authenticated
0	0	0	-	-	-	x	x	Logical channel number from zero to three
0	0	1	x	x	x	x	x	Reserved for future use by ISO/IEC JTC 1/SC 17
0	1	x	-	-	-	-	-	Secure messaging indication
0	1	0	-	-	-	-	-	-- No SM or no indication
0	1	1	-	-	-	-	-	-- SM with command header not processed
0	1	-	x	-	-	-	-	Command chaining control
0	1	-	0	-	-	-	-	-- The command is the last or only command of a chain
0	1	-	1	-	-	-	-	-- The command is not the last command of a chain
0	1	-	-	x	x	x	x	Logical channel number from four to nineteen
1	x	x	x	x	x	x	x	Proprietary class. The application-context defines the other bits.
1	1	1	1	1	1	1	1	Invalid (used by PPS in ISO 7816-3)

interindustry class

80

80

CLA Class byte

Class	Application
'80'	Electronic purse compliant with EN 1546-3
'8x'	Credit card compliant with EMV-2
'A0'	GSM compliant with prETS 300 608/GSM 11.11

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

81

81

INS Instruction byte

Table 4.1 — Commands in the alphabetic order

Command name	INS	See
ACTIVATE FILE	'44'	Part 9
APPEND RECORD	'E2'	7.3.7
CHANGE REFERENCE DATA	'24'	7.5.7
CREATE FILE	'E0'	Part 9
DEACTIVATE FILE	'04'	Part 9
DELETE FILE	'E4'	Part 9
DISABLE VERIFICATION REQUIREMENT	'26'	7.5.9
ENABLE VERIFICATION REQUIREMENT	'28'	7.5.8
ENVELOPE	'C2', 'C3'	7.6.2
ERASE BINARY	'0E', '0F'	7.2.7
ERASE RECORD (s)	'0C'	7.3.8
EXTERNAL (/ MUTUAL) AUTHENTICATE	'82'	7.5.4
GENERAL AUTHENTICATE	'86', '87'	7.5.5
GENERATE ASYMMETRIC KEY PAIR	'46'	Part 8
GET CHALLENGE	'84'	7.5.3
GET DATA	'CA', 'CB'	7.4.2
GET RESPONSE	'C0'	7.6.1
INTERNAL AUTHENTICATE	'88'	7.5.2
MANAGE CHANNEL	'70'	7.1.2
MANAGE SECURITY ENVIRONMENT	'22'	7.5.11
PERFORM SCQL OPERATION	'10'	Part 7
PERFORM SECURITY OPERATION	'2A'	Part 8
PERFORM TRANSACTION OPERATION	'12'	Part 7
PERFORM USER OPERATION	'14'	Part 7
PUT DATA	'DA', 'DB'	7.4.3
READ BINARY	'B0', 'B1'	7.2.3
READ RECORD (s)	'B2', 'B3'	7.3.3
RESET RETRY COUNTER	'2C'	7.5.10
SEARCH BINARY	'A0', 'A1'	7.2.6
SEARCH RECORD	'A2'	7.3.7
SELECT	'A4'	7.1.1
TERMINATE CARD USAGE	'FE'	Part 9
TERMINATE DF	'E6'	Part 9
TERMINATE EF	'E8'	Part 9
UPDATE BINARY	'D6', 'D7'	7.2.5
UPDATE RECORD	'DC', 'DD'	7.3.5
VERIFY	'20', '21'	7.5.6
WRITE BINARY	'D0', 'D1'	7.2.4
WRITE RECORD	'D2'	7.3.4

Table 4.2 — Commands in the numeric order

INS	Command name	See
'04'	DEACTIVATE FILE	Part 9
'0C'	ERASE RECORD (s)	7.3.8
'0E', '0F'	ERASE BINARY	7.2.7
'10'	PERFORM SCQL OPERATION	Part 7
'12'	PERFORM TRANSACTION OPERATION	Part 7
'14'	PERFORM USER OPERATION	Part 7
'20', '21'	VERIFY	7.5.6
'22'	MANAGE SECURITY ENVIRONMENT	7.5.11
'24'	CHANGE REFERENCE DATA	7.5.7
'26'	DISABLE VERIFICATION REQUIREMENT	7.5.9
'28'	ENABLE VERIFICATION REQUIREMENT	7.5.8
'2A'	PERFORM SECURITY OPERATION	Part 8
'2C'	RESET RETRY COUNTER	7.5.10
'44'	ACTIVATE FILE	Part 9
'46'	GENERATE ASYMMETRIC KEY PAIR	Part 8
'70'	MANAGE CHANNEL	7.1.2
'82'	EXTERNAL (/ MUTUAL) AUTHENTICATE	7.5.4
'84'	GET CHALLENGE	7.5.3
'86', '87'	GENERAL AUTHENTICATE	7.5.5
'88'	INTERNAL AUTHENTICATE	7.5.2
'A0', 'A1'	SEARCH BINARY	7.2.6
'A2'	SEARCH RECORD	7.3.7
'A4'	SELECT	7.1.1
'B0', 'B1'	READ BINARY	7.2.3
'B2', 'B3'	READ RECORD (s)	7.3.3
'C0'	GET RESPONSE	7.6.1
'C2', 'C3'	ENVELOPE	7.6.2
'CA', 'CB'	GET DATA	7.4.2
'D0', 'D1'	WRITE BINARY	7.2.4
'D2'	WRITE RECORD	7.3.4
'D6', 'D7'	UPDATE BINARY	7.2.5
'DA', 'DB'	PUT DATA	7.4.3
'DC', 'DD'	UPDATE RECORD	7.3.5
'E0'	CREATE FILE	Part 9
'E2'	APPEND RECORD	7.3.6
'E4'	DELETE FILE	Part 9
'E6'	TERMINATE DF	Part 9
'E8'	TERMINATE EF	Part 9
'FE'	TERMINATE CARD USAGE	Part 9

— In the Interindustry class, any valid INS code not defined in ISO/IEC 7816 is reserved for future use by ISO/IEC JTC 1/SC 17.

82

82

Exemples de commandes APDU

Champ de la commande APDU	Valeurs
CLA	BC = cartes de crédit françaises, cartes vitales françaises, A0 = cartes SIM (téléphonie) 00 = cartes Monéo (porte-monnaie en France), Mastercard, Visa
INS	20 = vérification du PIN, B0 = Lecture B2 = Lecture de record D0 = Écriture DC = Écriture de record A4 = Sélection du répertoire (directory) C0 = Demander une réponse (get response)
P1, P2	paramètres contenant des adresses à lire
LEN	longueur prévue pour la réponse ou bien longueur de l'argument de l'instruction
ARG	contient LEN octets (octets à écrire, PIN à vérifier, etc.)

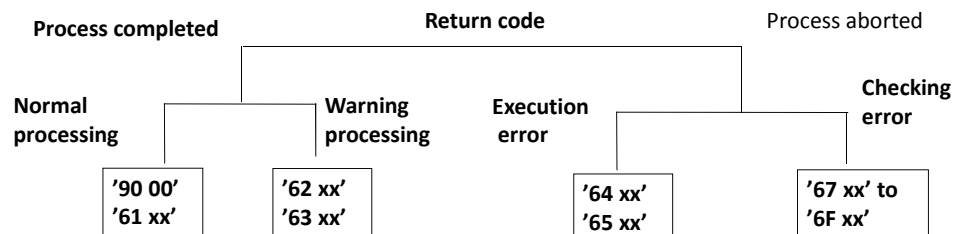
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

83

83

Return Codes

- SW1, SW2 = '90 00' command successful, '63xx' or '65xx' means EEprom has been modified,
- More than 50 different return codes defined by standard,
- Often not respected...

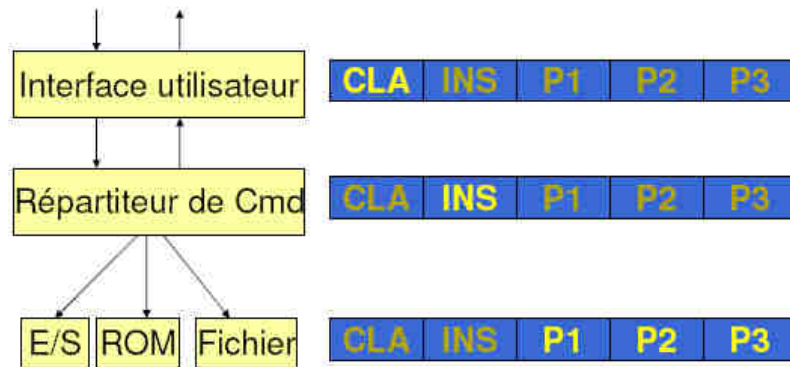


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

84

84

Un dispatcheur

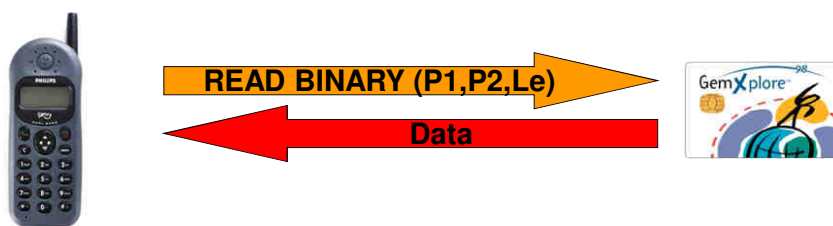


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

85

85

Exemple



- P1=Offset High,
- P2=Offset low.

Syntax :						P1, P2 : specify the data to be retrieved Le : length of data to retrieve
CLA	INS	P1	P2	Le		
A0	B0	xx	yy	Le		

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

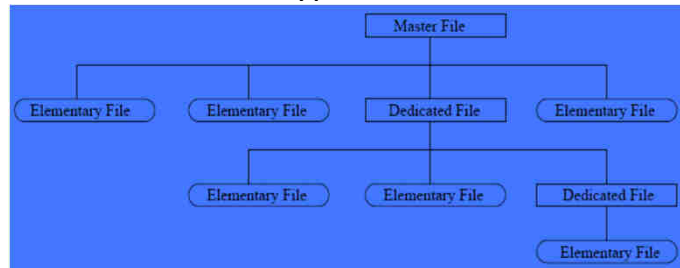
86

86

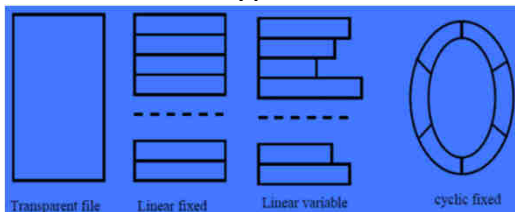
ISO 7816-4

- Le système de fichiers des cartes à puce supporte un système de fichiers hiérarchique qui peut contenir trois types de fichiers :

- "Master File" (MF)
- "Dedicated File" (DF)
- "Elementary File" (EF)



- Les différents types d' « Elementary File » (EF)



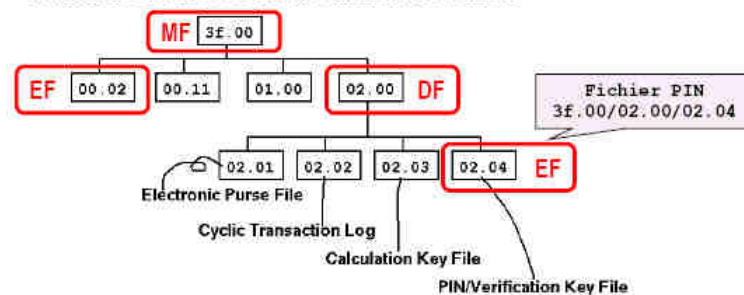
© 2019-2020 - D. Sauveron

87

87

Système de Fichier ISO 7816-4

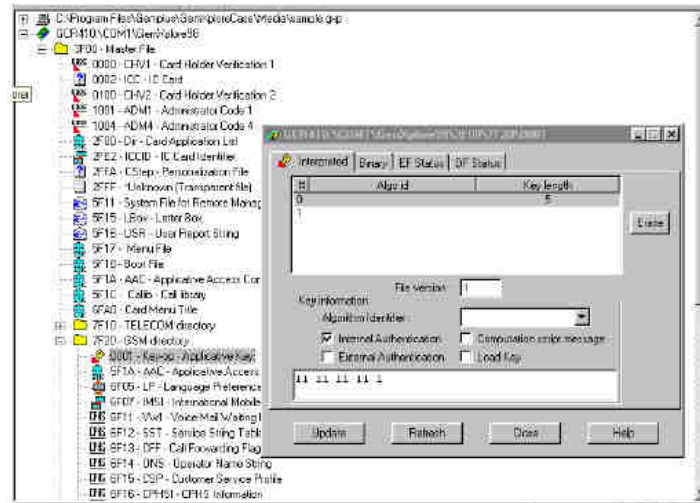
- Structure hiérarchique (MS-Dos, Unix)
 - MF : Fichier Maître : Obligatoire (Racine) 3F.00
 - Contient un fichier de contrôle d'information et de mémoire allouée
 - DF : Fichier Dédié :
 - contient des mots de passe pour accéder aux EF
 - EF Fichier Élémentaire :
 - contient des données ou des informations de contrôle



88

88

Exemple de système de fichier d'un module SIM GSM

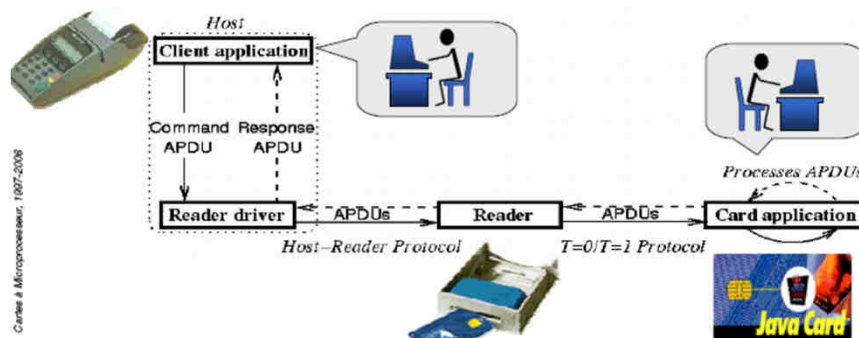


89

89

Architecture d'application carte

- Schéma général
 - Le terminal contrôle, la carte est passive
 - Dialogue terminal-carte de type requête/réponse
 - Format de messages standard : APDUs (ISO 7816-3 & 4)

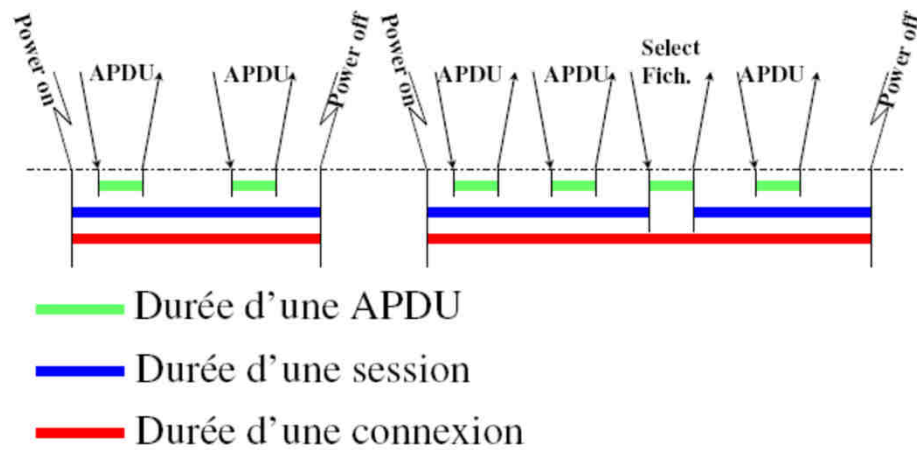


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

90

90

Déroulement d'une session du point de vue de l'application hôte



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

91

91

ISO 7816-5

- Spécifie le système de numérotation et les procédures d'enregistrement des identifiants d'applications (AID pour Application Identifier).
- Un AID = identification unique d'une application de la carte et de certains types de fichiers
- Un AID se compose de deux parties :

RID (5 octets)

PIX (0-11 octets)

- Ressource Identifier
 - RID is assigned by a national or international authority

RID			Meaning
D1	D2-D4	D5-D10	
X	-	-	Registration category (A international, D national)
	X		Country code according ISO 3166
		X	Application vendor number assigned by national or international body

- Proprietary Identifier extension
 - PIX is used to identify applications

En Java Card : le RID doit être le même pour le
paquetage et l'applet.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

92

92

ISO 7816-6

- Spécifie les éléments de données inter-industrie tels que le nom du porteur de carte, de la date d'expiration, etc.

Étiquette	Longueur	Valeur
-----------	----------	--------

- Exemple :

Name	Description	Source	Format	Template	Tag	Length
Application Expiration Date	Date after which application expires	ICC	n 6 YYMMDD	'70' or '77'	'5F24'	3
Application File Locator (AFL)	Indicates the location (SFI, range of records) of the AEFs related to a given application	ICC	var.	'77' or '80'	'94'	var. up to 252
Application Dedicated File (ADF) Name	Identifies the application as described in ISO/IEC 7816-5	ICC	b	'61'	'4F'	5-16
Application Identifier (AID) – terminal	Identifies the application as described in ISO/IEC 7816-5	Terminal	b	—	'9F06'	5-16
Application Interchange Profile	Indicates the capabilities of the card to support specific functions in the application	ICC	b	'77' or '80'	'82'	2
Application Label	Mnemonic associated with the AID according to ISO/IEC 7816-5	ICC	ans with the special character limited to space	'61' or 'A5'	'50'	1-16
Application Preferred Name	Preferred mnemonic associated with the AID	ICC	ans (see section 4.3)	'61' or 'A5'	'9F12'	1-16
Application Primary Account Number (PAN)	Valid cardholder account number	ICC	cn var. up to 19	'70' or '77'	'5A'	var. up to 10

(from « EMV Integrated Circuit Card Specifications for Payment Systems », Book 3 Application Specification, Version 4.3)

93

93

D'autres standards du monde de la carte

- ISO 14443, ...
- ETSI, (télécommunications, GSM)
- EMV, (cartes de paiement)
- ICAO (International Civil Aviation Organization, agence de l'ONU, biométrie, passeport)
- Santé,
- ...

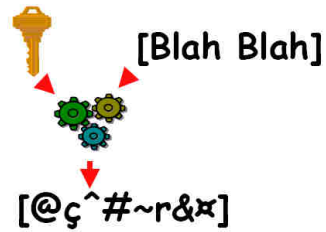
94

Pourquoi utiliser la carte à puce ?

La sécurité du stockage



La sécurité du calcul



La portabilité

La facilité d'utilisation



La personnalisation

Le coût

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

95

95

Quelques applications

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

96

96

La carte SIM/USIM

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

97

97

Les réseaux cellulaires

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

98

98

Il y a eu le GSM/1

- Dans les années 80, plusieurs réseaux cellulaires ont vu le jour en Europe
 - Les systèmes sont incompatibles d'un pays à un autre
 - Conséquences :
 - équipements mobiles limités aux frontières du pays
 - marché limité
- Création du « Groupe Spécial Mobile » pour :
 - Améliorer la qualité de la transmission
 - support international : roaming
 - rajout de nouvelles fonctionnalités
 - offrir des terminaux et des services à coûts accessibles

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

99

99

Il y a eu le GSM/2

- Normalisation 1982 : Baptisé « Groupe Spécial Mobile »
- Depuis 1989, l'**ETSI** (European Telecommunications Standard Institute) édite les spécifications du GSM et de l'UMTS (**Universal Mobile Telecommunications System**, réseau de 3^{ème} génération).
 - Siège de l'ETSI à Sophia Antipolis.
- 1991 : devenu une norme internationale nommée « Global System for Mobile communications »
- En Europe, le standard GSM utilise les bandes de fréquences 900 MHz et 1800 MHz.
- Aux États-Unis, la bande de fréquence utilisée est la bande 1900 MHz.
- **Tri-bande** : les téléphones portables pouvant fonctionner en Europe et aux États-Unis
- **Bi-bande** : les téléphones fonctionnant uniquement en Europe.
- La norme GSM autorise un débit maximal de 9,6 kbps
 - => transmission de la voix, des données numériques de faible volume,
 - des messages textes (**SMS**, pour Short Message Service) ou
 - des messages multimédias (**MMS**, pour Multimedia Message Service).

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

100

100

Évolutions des communications réseaux cellulaires

- 1G : reposait sur un système de communications mobiles
- 2G : repose sur une technologie numérique. Utilise essentiellement le standard GSM (9,6 kbps)
 - Pour la voix d'abord + données numériques de faible volume (SMS)
- Deux autres normes GSM :
 - GPRS (2.5G) : pour l'échange des données (21,4-171,2 kbps)
 - EDGE (2.75G) : pour des échanges de données plus performants (43,2-345,6 kbps)
- 3G : communications numériques plus rapides. (0,144-2 Mbps)
 - UMTS (3G)
 - HSDPA (3.5G)
- 4G : idem mais utilise LTE (10-300 Mbps)

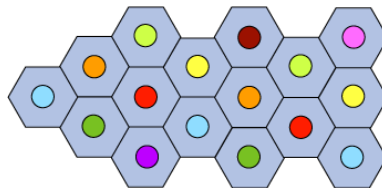
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

101

101

Notion de réseau cellulaire

- Un réseau de téléphonie mobile est basé sur la notion de **cellules**,
- Une cellule : est une zone circulaire qui couvre une zone géographique.
- Une cellule : centaine de mètres (zone urbaine), une trentaine de kms (zone rurale).



- Chaque cellule dispose d'un émetteur-récepteur central appelé « **station de base** » (en anglais Base Transceiver Station, **BTS**).
 - Plus le rayon d'une cellule est petit, plus la bande passante disponible est élevée (pour la même densité surfacique d'utilisateurs)
- Chaque cellule est entourée de 6 cellules voisines.
- Les cellules adjacentes **ne peuvent pas** utiliser la même fréquence.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

102

102

Éléments du réseau cellulaire GSM

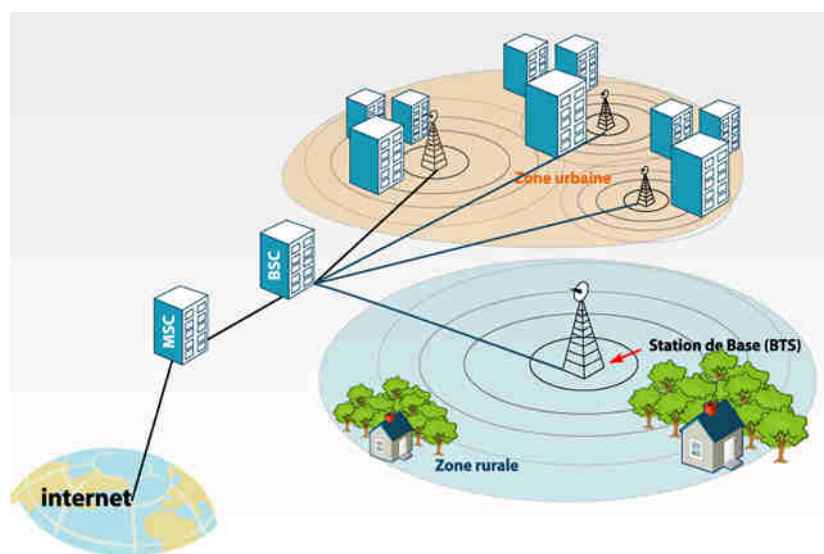
- **Un contrôleur de stations** (BSC, Base Station Controller) qui relie toutes les stations de base, chargé de gérer la répartition des ressources.
- **Sous-système radio** (en anglais **BSS** pour Base Station Subsystem) = contrôleur de stations + les stations de base.
- **Centre de commutation du service mobile** (MSC, Mobile Switching Center), géré par l'opérateur téléphonique, relie les contrôleurs de stations au réseau téléphonique public et à internet.
- **Sous-système réseau** (NSS, Network Station Subsystem) auquel appartient le MSC, chargé de gérer les identités des utilisateurs, leur localisation et l'établissement de la communication avec les autres abonnés.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

103

103

Architecture du réseau GSM



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

104

104

Bases de données manipulées

- **Le registre des abonnés locaux (HLR, Home Location Register)**: base de données contenant des informations (position géographique, informations administratives, etc.) sur les abonnés inscrits dans la zone du commutateur (MSC).
- **Le registre des abonnés visiteurs (VLR, Visitor Location Register)**: base de données contenant des informations sur les autres utilisateurs que les abonnés locaux. Le VLR rapatrie les données sur un nouvel utilisateur à partir du HLR correspondant à sa zone d'abonnement. Les données sont conservées pendant tout le temps de sa présence dans la zone et sont supprimées lorsqu'il la quitte ou après une longue période d'inactivité (terminal éteint).
- **Le registre des terminaux (EIR, Equipment Identity Register)** : base de données répertoriant les terminaux mobiles.
- **Le centre d'authentification (AuC, Authentication Center)** : élément chargé de vérifier l'identité des utilisateurs.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

105

105

Mobilité

- Le réseau cellulaire supporte la mobilité grâce à la gestion du handover
 - c-à-d le passage d'une cellule à une autre.
- Les réseaux GSM supportent aussi la notion d'itinérance (roaming)
 - c-à-d le passage du réseau d'un opérateur à un autre.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

106

106

Les stations mobiles

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

107

107

Station mobile

- **Station mobile** : terminal de l'utilisateur
- **Station mobile** composée de :
 - Une carte **SIM** (Subscriber Identity Module), pour identifier l'utilisateur de façon unique.
 - Un équipement mobile identifié par un numéro d'identification unique de 15 chiffres appelé **IMEI** (International Mobile Equipment Identity).
- Chaque carte SIM possède un numéro d'identification unique (et secret) : **IMSI** (International Mobile Subscriber Identity), qui peut être protégé à l'aide d'une clé de 4 chiffres appelés code PIN.
- La communication entre une station mobile et la station de base se fait par l'intermédiaire d'un lien radio, généralement appelé **interface air**.



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

108

108

Carte SIM

- Notion introduite en 1988
- Plus de 5,6 milliards de cartes SIM fabriquées en 2018
- Rôle fonctionnel dans le réseau :
 - Contient les détails concernant l'abonnement de l'utilisateur de téléphone mobile
 - Détient les secrets nécessaires pour prouver l'authenticité du mobile et pour chiffrer les échanges
 - Chargement de nouveaux services

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

109

109

Carte SIM : Mobilité

- **Détails d'abonnement mémorisés sur la carte :**
 - Identité unique de l'abonné (IMSI)
 - Numéro de téléphone de l'abonné (MSISDN)
 - Code de service (opérateur)
 - etc.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

110

110

Carte SIM : Services sécuritaires

- La carte SIM stocke des informations sensibles :

- Codes secrets :
 - Authentification de l'utilisateur : Code PIN (Personal Identification Code)
 - Authentification de l'opérateur : Code PUK (Personal Unlock Code)
- Clés secrètes :
 - Pour l'authentification de la carte SIM par le réseau : Ki
 - Pour la communication chiffrée : Kc

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

111

111

Carte SIM : Services téléchargeables

- La carte SIM est un environnement d'exécution pour les applications de confiance

- Capables d'interagir avec le mobile
 - Affichage d'infos sur l'écran du mobile
 - Récupérer les infos de l'utilisateur
 - etc.
- Capables d'interagir avec le réseau :
 - envoyer et recevoir des messages (SMS, GPRS, etc.)
 - géolocalisation
- Capables d'interagir avec le système fichiers de la carte SIM
 - écrire/lire des fichiers de la SIM

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

112

112

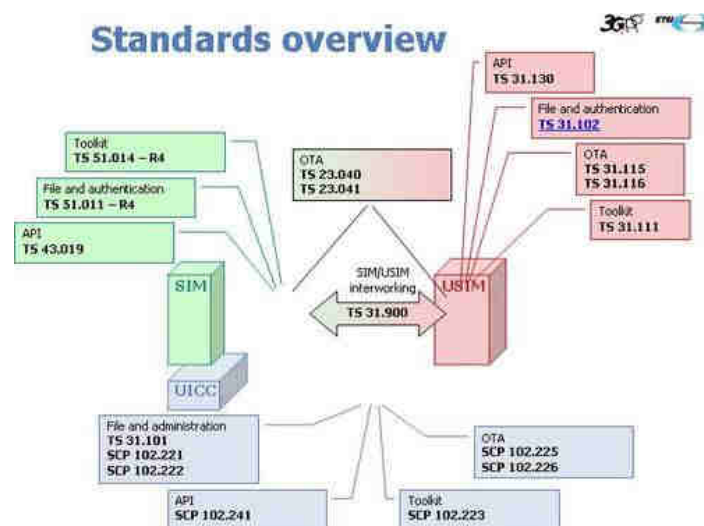
La normalisation et la sécurité

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

113

113

Les standards



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

114

114

Les standards ETSI

- SIM

- Gestion des Fichiers et Authentification : 3 GPP TS 51.011 (ETSI GSM 11.11)
- SIM Toolkit Applet Management : 3 GPP TS 51.014 (ETSI GSM 11.14)
- SIM API for Java Card : 3 GPP TS 43.019

- USIM

- Gestion des Fichiers et Authentification : 3 GPP TS 31.102
- USIM Toolkit Applet Management : 3 GPP TS 31.111
- USIM API for Java Card : 3 GPP TS 31.130

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

115

115

Méthodes de protection proposées dans GSM 02.09/1

- **La protection de l'identité d'un abonné :**

- L'abonné possède un identifiant (IMSI : International Mobile Subscriber Identity) permettant de retrouver les paramètres d'abonnement dans le HLR (Host Location Register) : base de données des comptes client. Le réseau délivre un TMSI (Temporary Mobile Subscriber Identity) une identité temporaire qui change à chaque appel pour interdire la traçabilité des communications.

- **L'authentification d'un abonné :**

- Une authentification forte est réalisée à l'aide de l'algorithme A3 associé à une clé Ki de 128 bits.

GSM 02.09: "Digital cellular telecommunications system (Phase 2+); Security Aspects".

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

116

116

Méthodes de protection proposées dans GSM 02.09/2

- **La confidentialité des données utilisateur :**

- Dans un réseau cellulaire radio, l'information est transmise par des ondes électromagnétiques (Over The Air) entre le téléphone mobile et la station de base. Les échanges entre mobile et station de base sont chiffrés à l'aide de l'algorithme A5 qui utilise une clé de chiffrement Kc. Kc est mise à jour à chaque appel (authentification) avec l'algorithme A8 de génération de clés. A3 et A8 sont souvent confondus (nommés A38 ou A3A8).

- **La protection de certaines informations telles que :**

- IMSI, numéros appelés ou appelants, le numéro de série du téléphone
- (IMEI : International Mobile Equipment Identity).

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

117

117

Infrastructures d'authentification du GSM

- Il existe cinq entités :

- La carte SIM
- Le mobile
- VLR (Visitor Location Register) : entité associée à plusieurs entités de base
- HLR (Host Location Register) : base de données clients
- Le centre d'authentification (AuC, Authentication Center).

- La norme 3GPP TS 43.020 identifie une cellule ou un ensemble de cellules à l'aide de l'étiquette LAI (Location Area Identity).

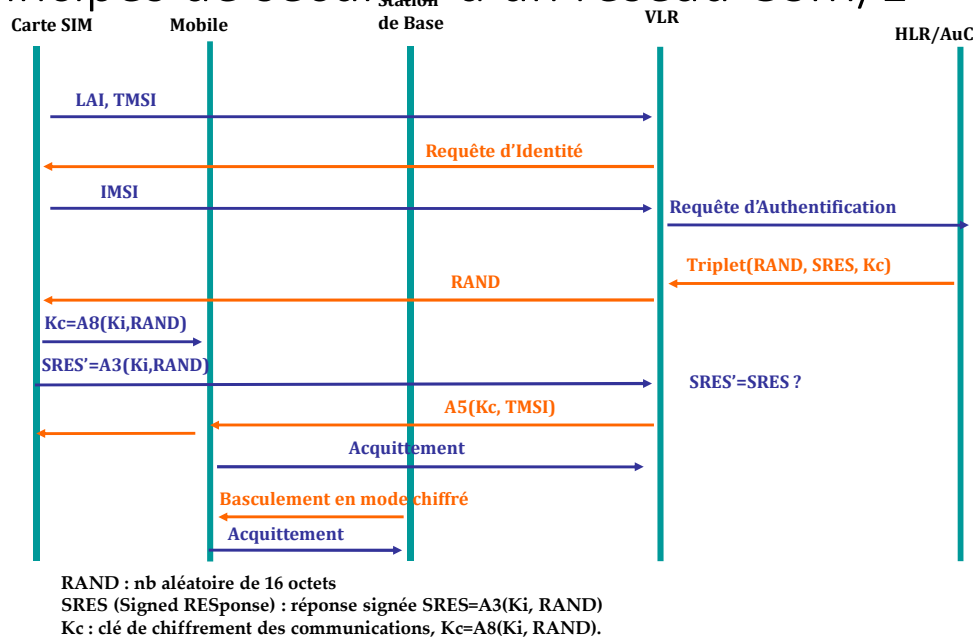
3GPP TS 43.020 – Technical Specification Group Services & System Aspects; Security Related Network Functions (Release 5, 2002).

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

118

118

Principes de sécurité d'un réseau GSM/1



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

119

119

Principes de sécurité d'un réseau GSM/2

- 1. L'abonné dispose des valeurs (LAI, TMSI) stockées dans le module SIM, suite à un appel précédent.
 - 2. Le mobile transmet au VLR les valeurs (LAI, TMSI).
 - 3. Si le VLR échoue pour retrouver l'IMSI, il envoie une requête d'identification au mobile.
 - 4. Le VLR récupère l'IMSI mémorisé dans la carte SIM.
 - 5. Le VLR envoie au HLR/AuC une demande d'authentification.
 - 6. AuC produit un triplet GSM (RAND, SRES, Kc).
 - 7. À la réception du triplet, le VLR transmet au mobile RAND.
 - 8. La carte SIM calcule $SRES' = A3(Ki, RAND)$ qui est envoyé au HLR.
 - 9. Le HLR vérifie l'égalité entre SRES et $SRES' \Rightarrow$ authentification de l'abonné en cas de succès.
 - 10. Le VLR choisit un nouveau TMSI, le chiffre avec l'algorithme A5 et la clé Kc et l'envoie au mobile qui le déchiffre.
- Les opérations de chiffrement et de déchiffrement appliqués aux signaux radio sont réalisées par le mobile (et non la carte SIM). Au-delà des stations de base, dans le réseau câblé de l'opérateur, il n'y a aucune garantie de confidentialité.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

120

120

Algorithmes cryptographiques

- La carte SIM réalise le calcul A3A8 dans un espace sûr.
- En 1998, Mark Briceno, Ian Goldberg et David Wagner (chercheurs à l'université de Berkeley) ont cassé l'algorithme A3A8.
- Même si GSM ne recommande aucun algorithme, les opérateurs utilisent la procédure secrète COMP128-1.
- Ces chercheurs ont aussi cassé cet algorithme en retrouvant la clé Ki en 2^{19} calculs (environ 500 000 essais). Pour cette raison, les composants qui intègrent COMP128-1 sont munis d'un compteur limitant le nombre d'appels à 100 000.
- Les modules SIM sont aujourd'hui basés sur l'algorithme COMP128-2 dont l'algorithme est pour le moment secret.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

121

121

Spoofing et phishing

- Une petite vidéo : <https://p-fb.net/gsm-demo.html>

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

122

122

Le système de fichiers

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

123

123

Caractéristiques physiques d'une carte SIM

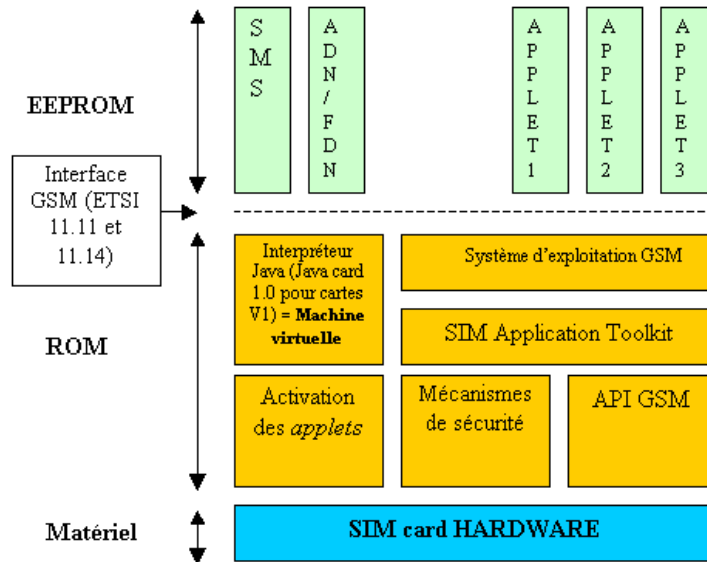
- Début des années 90:
 - Une carte SIM : un CPU (8 bits), RAM (128 octets), ROM (7 ko), EEPROM (3 ko).
- Actuellement :
 - Une carte SIM : un CPU (32 bits), RAM (16 ko), ROM (512 ko), EEPROM/FLASH (512 ko), processeur dédié au calcul cryptographique.
 - La ROM (Read Only Memory) contient le système d'exploitation de la carte, les mécanismes de sécurité (algorithmes spécifiques (API GSM)).
 - L'EEPROM (Electrically Erasable Programmable Read Only Memory) contient des répertoires définis par la norme GSM (tels que les numéros de téléphones l'abonné...) et des données liées aux applets (service de messages courts et applications spécifiques).
 - la RAM (Random Access Memory) permet d'effectuer des calculs ou de charger des instructions et les exécuter.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

124

124

Structure d'une carte SIM

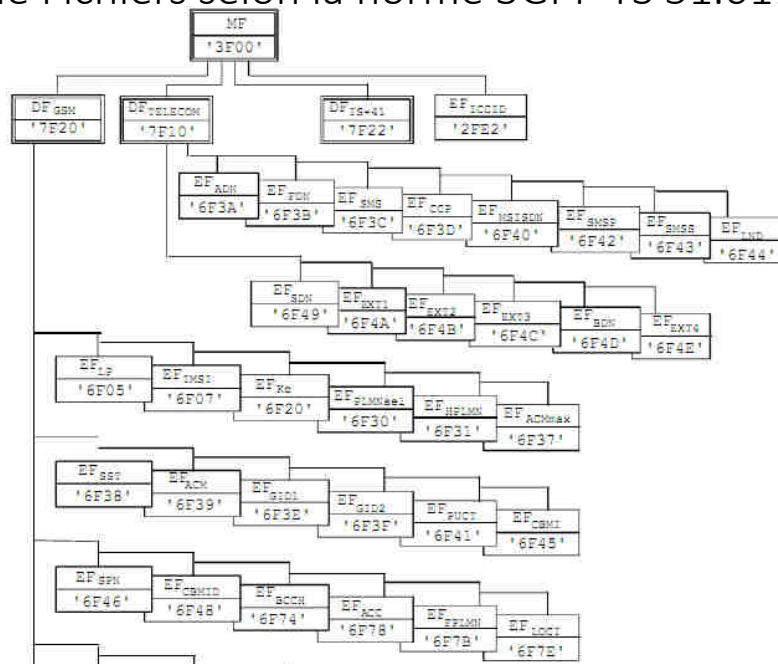


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

125

125

Système de Fichiers selon la norme 3GPP TS 51.011



126

126

La commande SELECT

- A0 A4 00 00 02 XX XX (XX XX : FID du fichier/répertoire à sélectionner).
- La sélection d'un répertoire entraîne une réponse qui peut inclure des informations telles que :
 - la taille mémoire non utilisée
 - le nom du répertoire sélectionné
 - le type du répertoire (MF ou non)
 - présentation du code PIN
 - nombre de sous répertoires
 - nécessité éventuelle de présentation du code PIN, avec le nombre d'essais possibles.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

127

127

Lectures de Fichiers

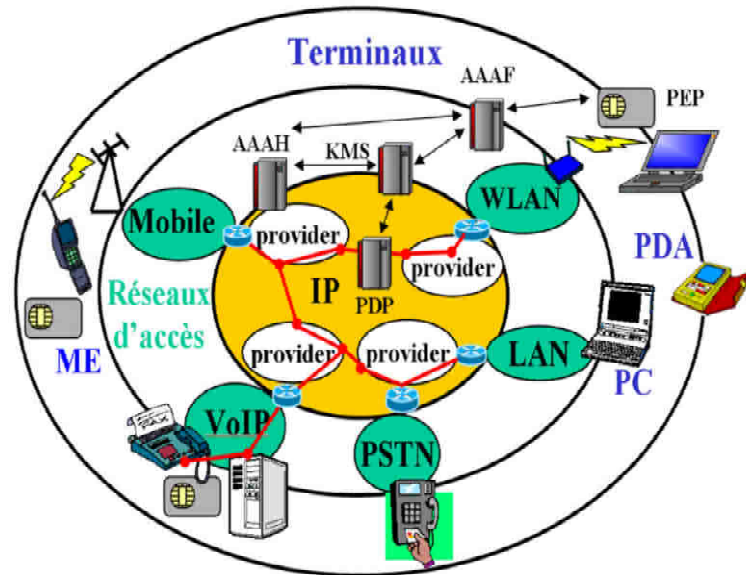
- Lecture de l'IMSI
 - Le fichier EF_{IMSI} (6F07) du répertoire GSM est de type transparent, il contient l'IMSI.
 - La sélection du fichier retourne la taille du fichier.
 - A0 B0 00 00 09 (READ BINARY 9 octets, taille de l'IMSI).
- Lecture de TMSI et LAI
 - Ces paramètres sont lus à partir du fichier EF_{LOCI} (6F7E)
 - A0 B0 00 00 0B (READ BINARY 11 octets, 4 octets pour TMSI suivis de 5 octets pour LAI, ..)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

128

128

La carte : omniprésente !



129

129



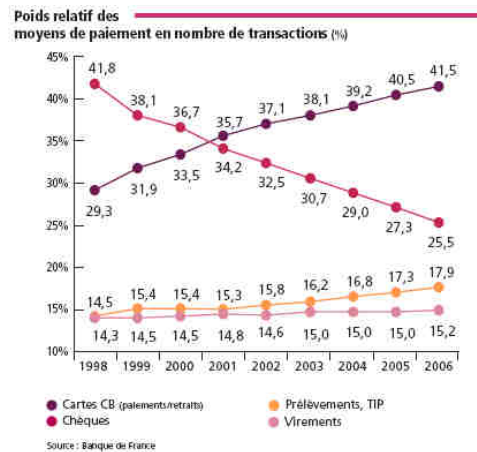
La carte bancaire

130

130

Applications de paiement

- Cartes de Crédit : Somme débité sur le compte du titulaire avec un taux d'intérêt fixé
- Cartes de Débit : Compte débité quelques jours après l'achat

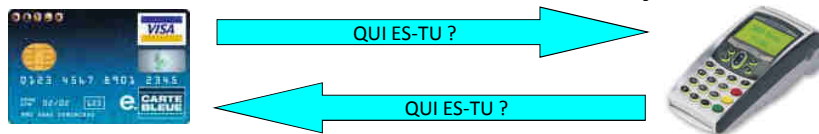


131

131

Applications de paiement : Techniques

- Authentification mutuelle carte et terminal commerçant



- Signature Électronique



- Contrôle du PIN



- Contrôle (régulier ou systématique) du solde bancaire (en ligne)

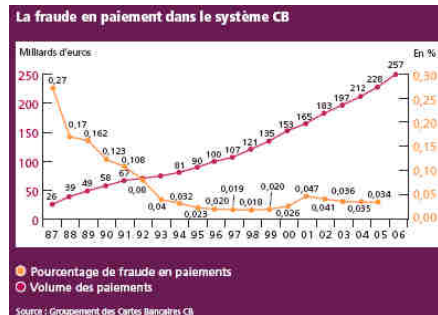
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

132

132

Applications de paiement (Exemple : GIE Cartes Bancaires)

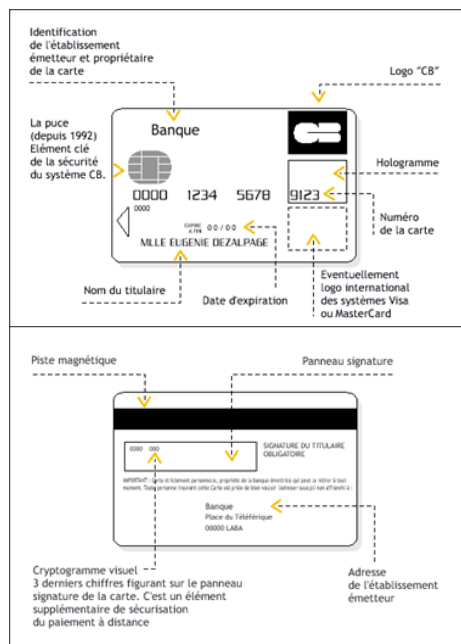
- 53,6 millions de cartes
- 1,2 million de points d'acceptation :
 - 950 000 terminaux de paiement dans les commerces
 - 140 000 automates de vente
 - 50 000 distributeurs de billets
 - 75 000 commerçants à distance dont 15 000 sites Internet
- 1 Distributeur Automatique de Billets pour 1200 habitant en France
- 349,5 Mds€ de transactions en 2006 (257,3 Mds€ de paiements et 92,2 Mds€ de retraits)
- 260 M€ de fraudes du à la piste magnétique contrefaite et utilisée depuis l'étranger (en 2003)



133

133

La carte bancaire



Limo...



134

134

La carte à puce et sa sécurité



- La Yescard : une carte « pirate » qui dit OUI à toutes les transactions !
- Une remise en cause la sécurité de la carte à puce ?



NON

Il s'agit d'une mauvaise utilisation de la carte dans un système plus large, ici, le système bancaire

Exemple : Utilisation d'un seul des différents verrous du coffre-fort

- Aujourd'hui on utilise « tous les » verrous et le système est à nouveau sécurisé.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

135

135

Plan de Migration de VISA vers les SC

Feb 1998	Launch of Chip Migration program
Jan 1999	New Visa and Electron chip cards to be EMV and VIS compliant
Jan 2001	All new Visa Cash programmes to be CEPS compliant
Oct 2001	All Acquirer hosts certified to EMV, All new Terminals EMV+VIS compliant (POS, EPOS, ATM)
Jan 2002	Regional review to assess EMV and PIN readiness
July 2002	Visa, Electron and Visa Cash to be EMV, VIS and CEPS compliant
Jan 2005	All Visa Electron terminals to be EMV and PIN enabled, Proposed Intra regional liability shift

Royaume Uni: PIN obligatoire depuis le 14/02/2006

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

136

136

Jouons un peu

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

137

137

Smart Card Fraud

- Scenario 1:
 - Although PIN protected stolen magnetic stripe credit cards were successfully used to withdraw money.
 - Audit of the ATM's log file showed that although the thief presented three false PIN code he could somehow get the card back and try again.
 - The correct PIN was found by exhaustive search after approximately 5000 attempts.
- What happened ?

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

138

138

Smart Card Fraud

- Scenario 1:
 - Although PIN protected stolen magnetic stripe credit cards were successfully used to withdraw money.
 - Audit of the ATM's log file showed that although the thief presented three false PIN code he could somehow get the card back and try again.
 - The correct PIN was found by exhaustive search after approximately 5000 attempts.
- What happened ?
 - After stealing the card, the thief made a small hole in it, attached a wire to pull out the card after three false presentation

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

139

139

Smart Card Fraud

- Scenario 2:
 - Users insert their cards to ATMs enter their PINs but get no money.
 - The ATM swallows the card and display the message "Invalid card contact your bank".
 - Money was however withdrawn with the card later.
- What happened ?

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

140

140

Smart Card Fraud

- Scenario 2:
 - Users insert their cards to ATMs enter their PINs but get no money.
 - The ATM swallows the card and display the message “Invalid card contact your bank”.
 - Money was however withdrawn with the card later.
- What happened ?
 - A false ATM...

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

141

141

Smart Card Fraud

- Scenario 3:
 - Same as the previous one but using smart card with an EEPROM having a retry counter limited to 3. The card is always returned to the user but its EEPROM retry counter never decrease.
 - Audit of the ATM's log file showed that although the thief presented three false PIN code he could somehow get the card back and try again.
 - The correct PIN was found by exhaustive search after approximately 5000 attempts.
- What happened ?

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

142

142

Smart Card Fraud

- Scenario 3:
 - Same as the previous one but using smart card with an EEPROM having a retry counter limited to 3. The card is always returned to the user but its EEPROM retry counter never decrease.
 - Audit of the ATM's log file showed that although the thief presented three false PIN code he could somehow get the card back and try again.
 - The correct PIN was found by exhaustive search after approximately 5000 attempts.
- What happened ?
 - In old cards EEPROM programming voltage was done using an external programming voltage (Vpp) supplied through a specific ISO contact. The thief had covered this contact with a sticker.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

143

143

Smart Card Fraud

- Scenario 4:
 - The ATM's log file and cash do not match, money is missing,
 - Audit of the ATM's log file showed that the same user withdrew money several times. He always forgot his banknotes that were swallowed back by the ATM after a short time-out (security features)
- What happened ?

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

144

144

Smart Card Fraud

- Scenario 4:
 - The ATM's log file and cash do not match, money is missing,
 - Audit of the ATM's log file showed that the same user withdrew money several times. He always forgot his banknotes that were swallowed back by the ATM after a short time-out (security features)
- What happened ?
 - The thief would withdraw three banknotes but take only two of them. The remaining banknote was detected by the paper sensor and swallowed back by the ATM which automatically cancelled the transaction. The sensor could not distinguish between one, two or three banknotes.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

145

145

Smart Card Fraud

- Scenario 5:
 - Users complain that when attempting to withdraw money they get nothing, money is however debited from their accounts.
 - An audit of the ATM's log file shows nothing abnormal
- What happened ?

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

146

146

Smart Card Fraud

- Scenario 5:
 - Users complain that when attempting to withdraw money they get nothing, money is however debited from their accounts.
 - An audit of the ATM's log file shows nothing abnormal
- What happened ?
 - The hole through which money was delivered was covered by a fake hole (piece of metal).
 - The back of the fake hole was covered with glue to prevent the machine from swallowing back forgotten banknotes.
 - After each victim's withdrawal, the thief would come to the ATM, remove the piece of metal and collect the banknote.

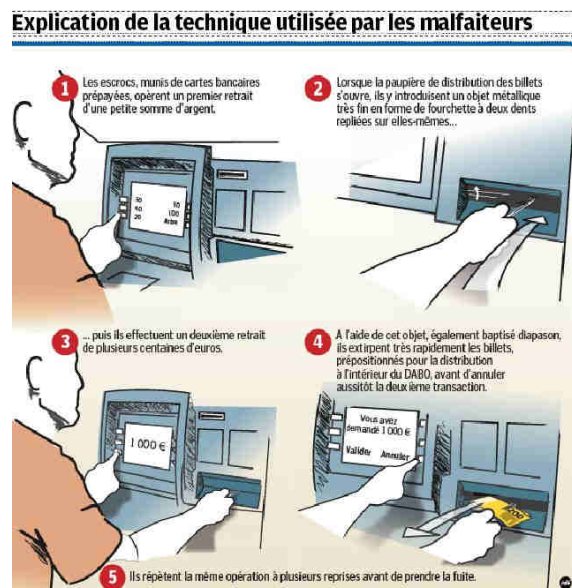
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

147

147

Un bon coup de fourchette

- <http://www.leparisien.fr/faits-divers/le-coup-de-la-fourchette-leurrapporte-un-million-d-euros-16-08-2012-2126839.php>



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

148

148

Skimming

- vendredi 07 novembre 2008 : un million retiré sur des comptes de l'Ouest (Ouest France)



- Cout 8k\$
 - Wifi + sms
 - Hong kong, malaisie, USA, France

- <http://www.sudouest.fr/2013/06/24/de-bordeaux-a-la-rochelle-l-escroquerieau-terminal-de-paiement-fait-des-ravages-1094946-7.php>

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

149

149

Smart Card Fraud

- Scenario 6:
 - Although PIN protected stolen smart card were successfully used to withdraw money.
 - An audit of the ATM's log file shows that the correct PIN was used in the withdrawal operation.
- What happened ?

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

150

150

Smart Card Fraud

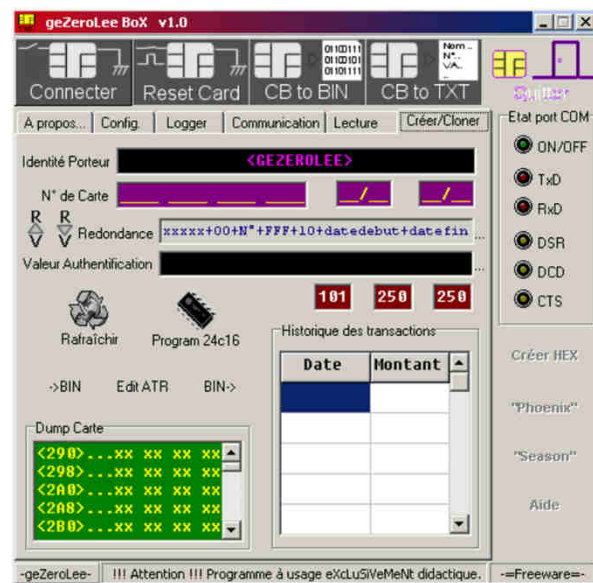
- Scenario 6:
 - Although PIN protected stolen smart card were successfully used to withdraw money.
 - An audit of the ATM's log file shows that the correct PIN was used in the withdrawal operation.
- What happened ?
 - The fraud was technical: the smart card's software was programmed to compare the presented PIN and if incorrect to increase the EEPROM counter.
 - EEPROM programming is characterized by an increased power consumption and requires 5ms.
 - The thief used a board that presented automatically all the PIN value (0000 to 9999) but detected the current consumption increase and powered off the card before the EEPROM retry counter could be updated.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

151

151

Yescard



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

152

152

Yescard

- Some details

- Below a given threshold, card and holder authentication are done locally,
- Only some terminals (gasoline, transport ticket, video rental and so on...) are concerned
 - ATM need always an on-line authorization
 - Merchant terminal will be detected by the merchant or he/she is himself part of the attack.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

153

153

Yescard

- Context

- Weakness known by industrial experts
 - Off line authentication : public key
 - On line authentication : secret key
 - Ks is not stored into the card but easy to retrieve (key was only 320 bits)
 - Cloning a card with forged Vs compatible with Id
- Ended with the court case "Serge Humpich vs GIE-CB."
- Keys have been broadcasted thanks to Usenet.
- Card have migrated to EMV 5.1 and 5.2

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

154

154

Yescard

- Consequence
 - Media focused too much on cloning Banking Card
 - Moved from a technical risk to industrial image
 - Problem related with knowledge broadcast
 - Know-how used in a fraud context
 - Do we need an internet based *full disclosure*...

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

155

155

Fraud conclusion

- All this flaws described here are at least 15 years old,
- All of them of course have been corrected,
- Security is a permanent race...

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

156

156

La norme EMV : introduction

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

157

157

Contexte de la norme

- EMVCo manages, maintains and enhances the EMV® Integrated Circuit Card **Specifications** for chip-based payment cards and acceptance devices, including point of sale (POS) terminals and ATMs. EMVCo also establishes and administers **testing** and approval processes to evaluate **compliance** with the EMV Specifications. EMVCo is currently owned by American Express, JCB (Japan Credit Bureau), MasterCard and Visa.
- A primary goal of EMVCo and the EMV Specifications is to help facilitate global **interoperability** and **compatibility** of chip-based payment cards and acceptance devices. This objective extends to new types of payment devices as well, including **contactless** payment and **mobile** payment.
- Source : <http://www.emvco.com> (2018)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

158

158

EMV

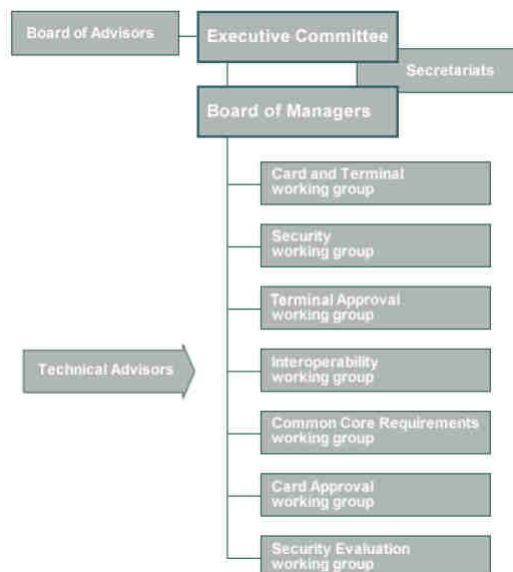
- Standard des cartes de paiement depuis 1995
- Organismes fondateurs (déc. 1993):
 - Europay International (racheté par Mastercard en 2002) ;
 - MasterCard International ;
 - Visa International ;
- Rejoint par :
 - le japonais JCB International (depuis Déc. 2004)
 - l'américain American Express (depuis Fév. 2009)
- Première version des spécifications en 1996.
- En France, depuis fin 2006 les cartes bancaires et les terminaux de paiement électroniques (TPE) respectent le standard EMV.



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

159

159



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

160

160

Spécifications EMV

- Spécifications disponibles sur <http://www.emvco.com>
- Longue spécification
 - de l'ordre de 747 pages

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

161

161

Plusieurs parties

- Integrated Circuit Card (ICC) Specifications for Payment Systems
- Version 4.3, November 2011
- Book 1
 - Application Independent ICC to terminal
- Book 2
 - Security and Key Management
- Book 3
 - Application Specification
- Book 4
 - Cardholder, Attendant, and Acquirer

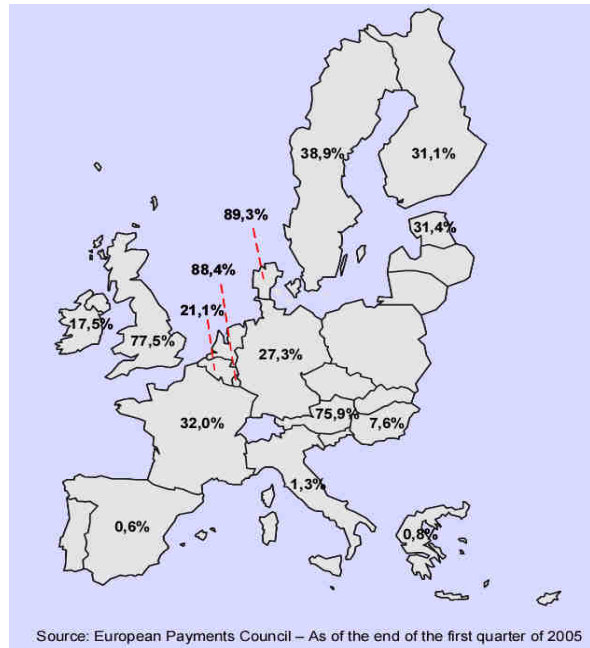
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

162

162

Déploiement EMV

- En 2005, il y avait :
 - 176 millions cartes EMV sur 550 millions de cartes circulant en Europe

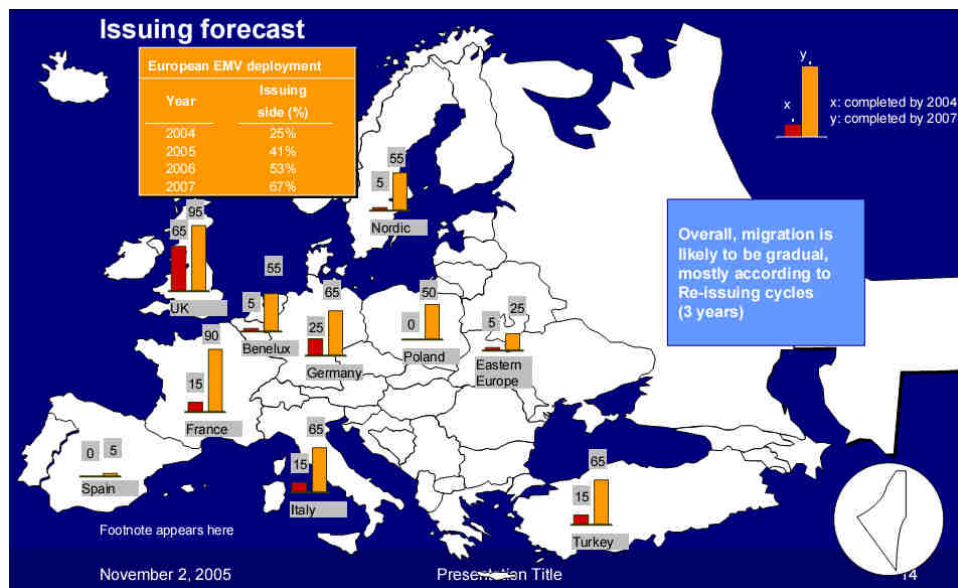


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

163

163

Déploiement de cartes EMV



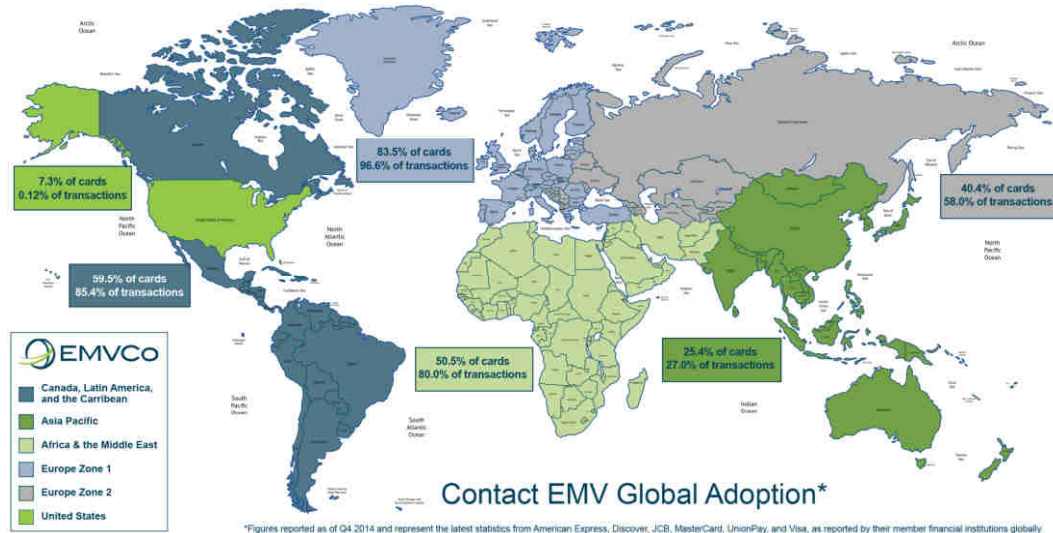
Source : International Master Card

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

164

164

Déploiement de cartes EMV

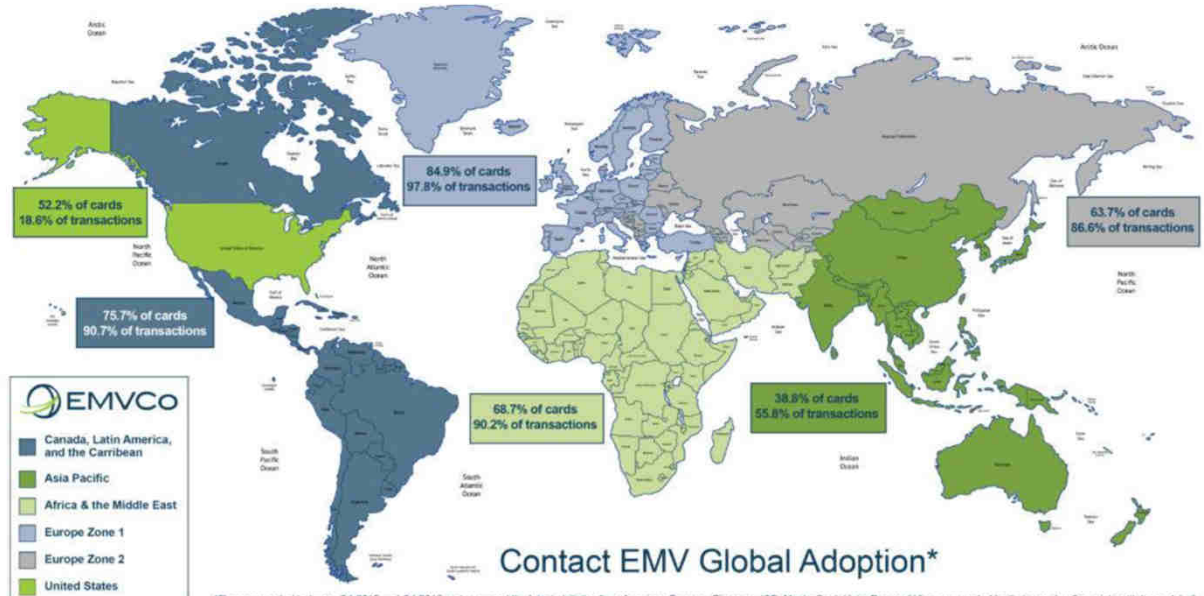


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

165

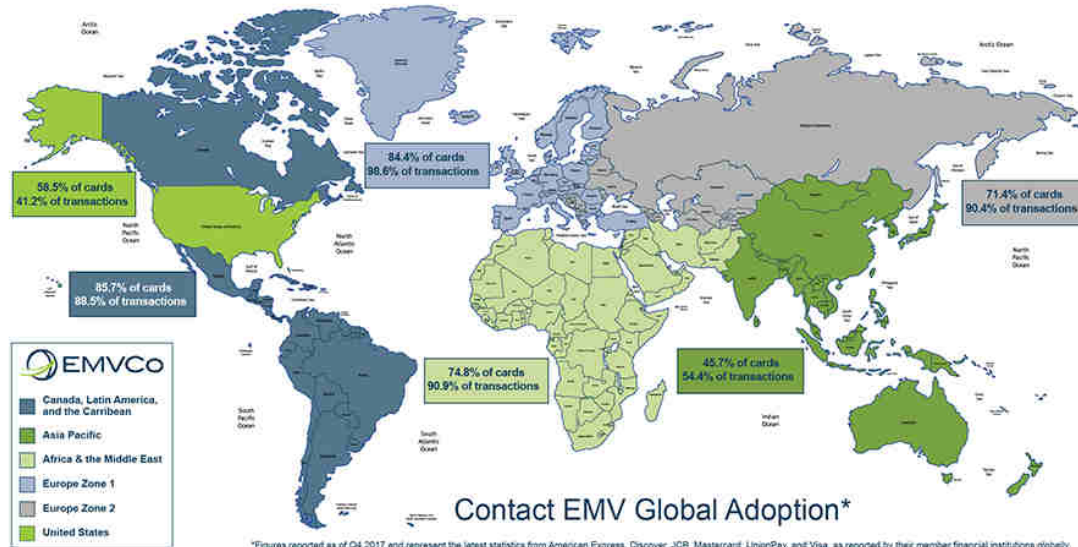
165

Déploiement de cartes EMV



166

Déploiement de cartes EMV



167

Déploiement de cartes EMV

Worldwide EMV Chip Card Deployment and Adoption*

Region	2013		2014	
	EMV Cards	Adoption Rate	EMV Cards	Adoption Rate
Canada, Latin America, and the Caribbean	471M	54.2%	544M	59.5%
Asia Pacific	942M	17.4%	1,676M	25.4%
Africa & the Middle East	77M	38.9%	116M	50.5%
Europe Zone 1	794M	81.6%	833M	83.5%
Europe Zone 2	84M	24.4%	153M	40.4%
United States	-	-	101M	7.3%

*Figures reported in Q4 2013 and Q4 2014, respectively, and represent the latest statistics from American Express, Discover, JCB, MasterCard, UnionPay, and Visa, as reported by their member institutions globally.

Worldwide EMV Chip Card Deployment and Adoption*

Region	2014		2015		2016	
	EMV Cards	Adoption Rate	EMV Cards	Adoption Rate	EMV Cards	Adoption Rate
Canada, Latin America, and the Caribbean	544M	75.7%	680M	71.7%	717M	75.7%
Asia Pacific	1,676M	25.4%	2,459M	32.7%	3,331M	38.8%
Africa & the Middle East	116M	50.5%	160M	61.2%	184M	68.7%
Europe Zone 1	833M	83.5%	881M	84.3%	921M	84.9%
Europe Zone 2	153M	40.4%	200M	52.3%	243M	63.7%
United States	101M	7.3%	394M	26.4%	675M	52.2%

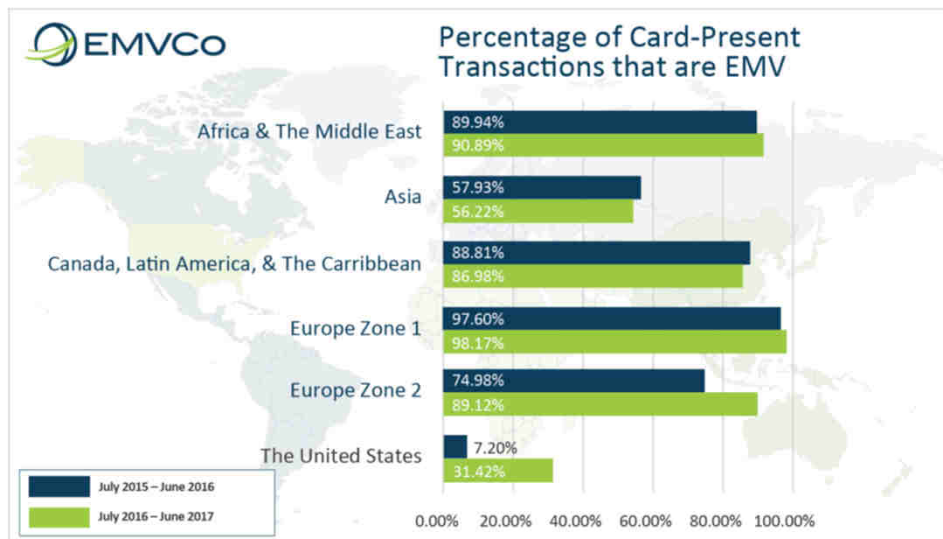
*Figures reported in Q4 of 2014, 2015, and 2016, respectively, and represent the latest statistics from American Express, Discover, JCB, MasterCard, UnionPay, and Visa, as reported by their member institutions globally.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

168

168

Card-Present Transaction

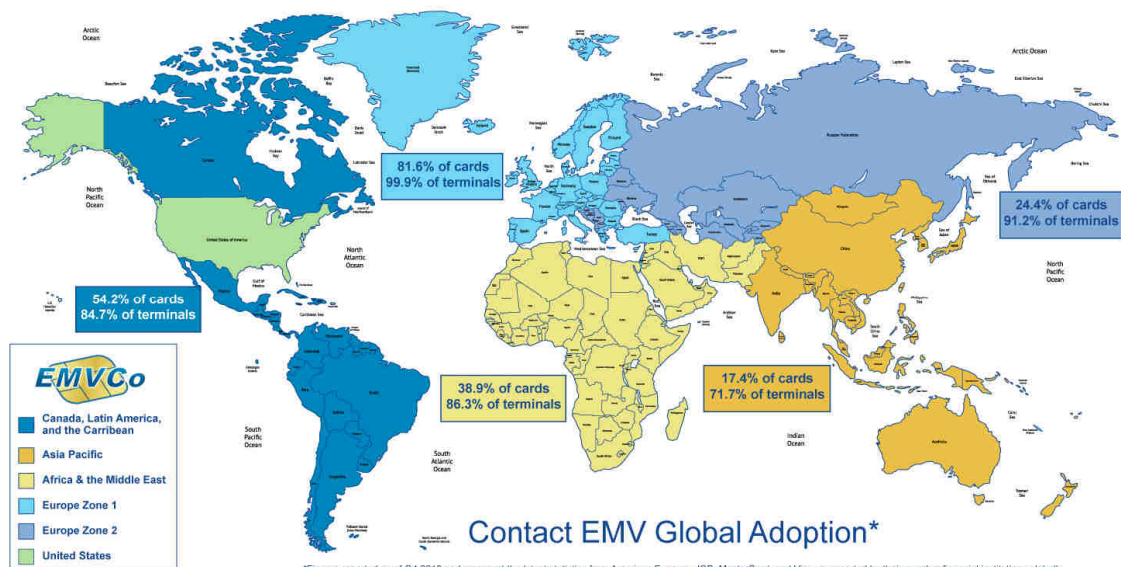


*Figures represent the percentage of all card-present transactions processed by each member institution that are EMV transactions (Contact or Contactless). The reported data is from the twelve months of July 2015 through June 2016 (blue bar) and July 2016 through June 2017 (green bar) and represents the most accurate possible data that could be obtained by American Express, Discover, JCB, MasterCard, UnionPay, and Visa during this period. To qualify as an "EMV transaction" for the purpose of this methodology, both the card and terminal used during a transaction must be EMV-enabled. Data is reported from the acquirer perspective. These figures do not include offline transactions, "on us" transactions (defined as a transaction handled exclusively by another processor), and/or transactions processed by non-EMVCo-member institutions, such as local schemes.

169

169

Déploiement de terminaux EMV



*Figures reported as of Q4 2013 and represent the latest statistics from American Express, JCB, MasterCard, and Visa, as reported by their member financial institutions globally. Figures do not include data from the United States. Figures are reported by region and do not imply country-by-country statistics.

Card EMV Co - 2013-2020 - 10.000000000000000

170

170

Déploiement de l'EMV aujourd'hui

- Europe :
 - Zone SEPA (Single Euro Payment Area)
 - Migration vers l'EMV de Janvier 2008 au 31 Décembre 2010.
 - En 2010: 100% des cartes doivent être conformes à l'EMV
 - En 2010, 590 millions de cartes bancaires EMV seront en circulation

- Dans le monde (Europe, Asie et Amérique Latine)
 - En 2010, 830 millions de cartes EMV en circulation.

- USA : pas de cartes conformes à l'EMV, idem pour les cartes e-ID.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

171

171

Raisons de la migration

- Liability shift
 - les coûts induits par une fraude lecteur ou terminal seront pris en charge par les banques (émetteurs de la carte) ou les commerçants dont les matériels ne seraient pas conformes à EMV

- Fraude
 - différente selon les pays et les systèmes de paiement

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

172

172

Spécifications EMV

- Spécifications EMV:
 - basées sur la norme ISO/IEC 7816
 - doivent être lues conjointement avec la norme ISO
- Si des définitions fournies dans EMV sont différentes de la norme ISO alors les définitions de la norme EMV remplacent celles de l'ISO
- Ces spécifications doivent être utilisées par :
 - Les fabricants de ICC et de terminaux
 - Les concepteurs de systèmes de paiement
 - Les institutions financières qui implantent des applications financières sur ICC

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

173

173

Book 2 : Security and Key Management

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

174

174

Book 2

- Static Data Authentication
- Dynamic Data Authentication
- Cryptage du code PIN hors ligne
- Intégrité et confidentialité
- Mécanismes de sécurité : chiffrement symétrique, asymétrique, signature numérique.
- Algorithmes cryptographiques : RSA, DES, SHA-1

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

175

175

Acteurs du protocole EMV

- La banque du client : émetteur de la carte
- Le client : Carte bancaire
- Le TPE (Terminal de Paiement Électronique ou le DAB (Distributeur Automatique de Billets) : marchand
- Une autorité de certification : CA

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

176

176

Mécanismes d'authentification

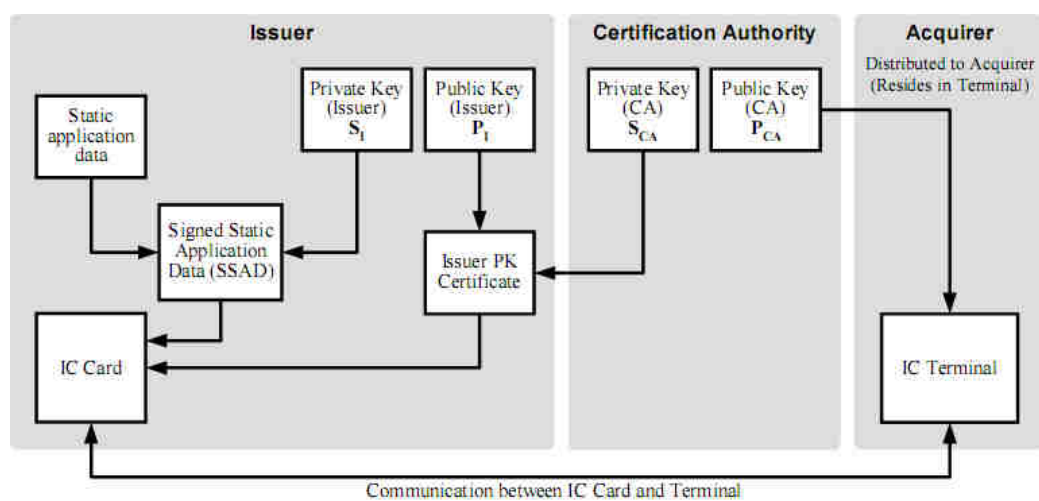
- **Processus SDA : Static Data Authentication**
 - consiste pour le terminal à vérifier une donnée signée mise dans la carte durant sa personnalisation
- **Processus DDA : Dynamic Data Authentication**
 - en plus d'une authentification statique, vérifie si la carte possède un secret délivré par l'émetteur de la carte

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

177

177

SDA : Static Data Authentication

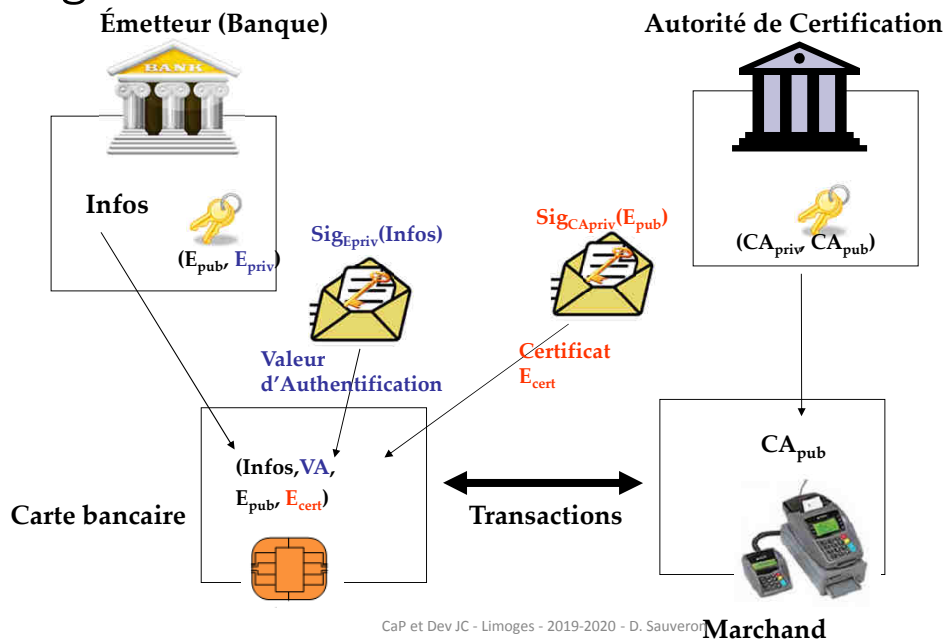


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

178

178

Organisation des acteurs dans SDA



179

179

SDA : à la personnalisation

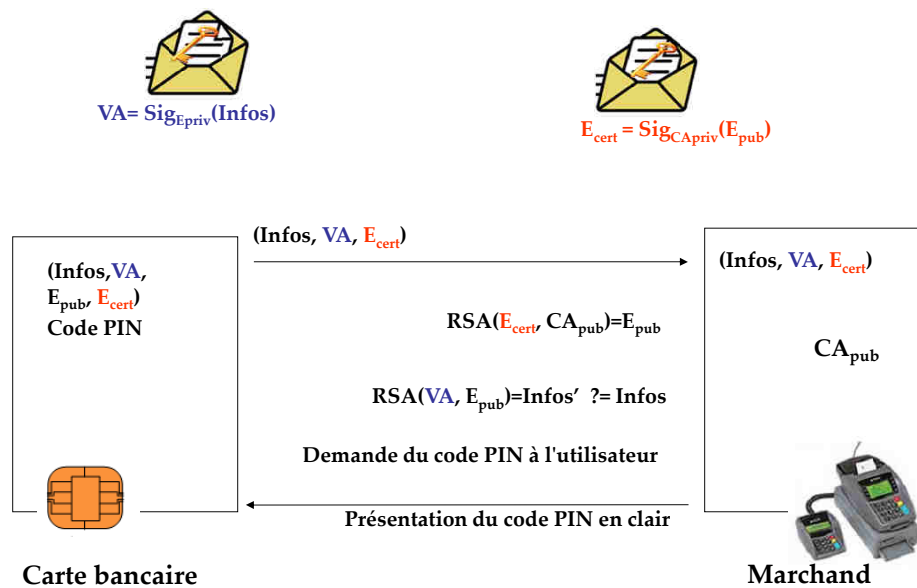
- Pendant la phase de personnalisation, la carte reçoit les informations suivantes :
 - Le nom du porteur, le numéro de la carte ou encore la date limite de validité de celle-ci (notés *Information*).
 - une valeur d'authentification (noté VA), signature RSA d'Information générée avec la partie privée de la clé de l'émetteur : $VA = \text{Sig}_{E_{priv}}(\text{Information})$
 - le certificat de l'émetteur (E_{cert}) contenant sa clé publique signée par une autorité de certification
 - le code PIN transmis au porteur de cette carte.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

180

180

Lors de l'utilisation



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

181

181

SDA : à la l'utilisation

• Avant toute transaction :

- la carte fournit au terminal *Information*, le certificat E_{cert} de la banque émettrice, ainsi que la valeur d'authentification VA
- le terminal vérifie E_{cert} avec la clé publique de l'autorité de certification (CA_{pub}) et vérifie VA avec la clé publique de la banque émettrice
- le terminal demande à l'utilisateur le code PIN et le transmet (**en clair**) à la carte pour qu'elle le vérifie.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

182

182

La Yes card est possible

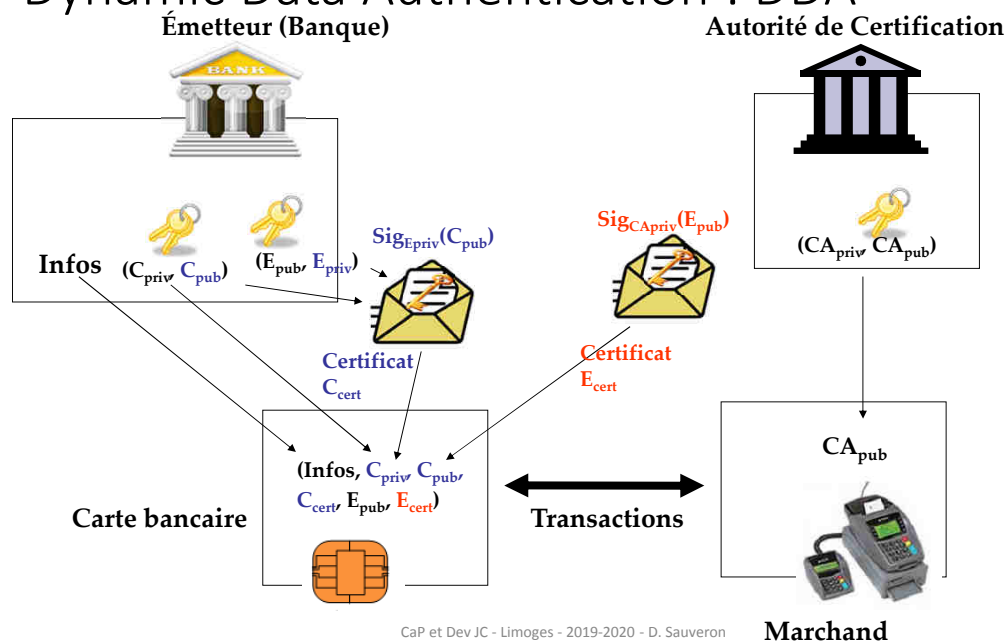
- Lors de transactions offline (par exemple le terminal n'a pas de connexion), il est possible de renvoyer les données statiques copiées d'une autre carte.
- Mais la plupart des terminaux étant en ligne de façon permanente, la fraude a diminuée car les yes cards peuvent être détectées
- Puis la DDA arrive afin de permettre à la carte de signer un aléa produit par le terminal (et donc de prouver qu'elle est une vraie carte)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

183

183

Dynamic Data Authentication : DDA



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

Marchand

184

184

DDA : à la personnalisation

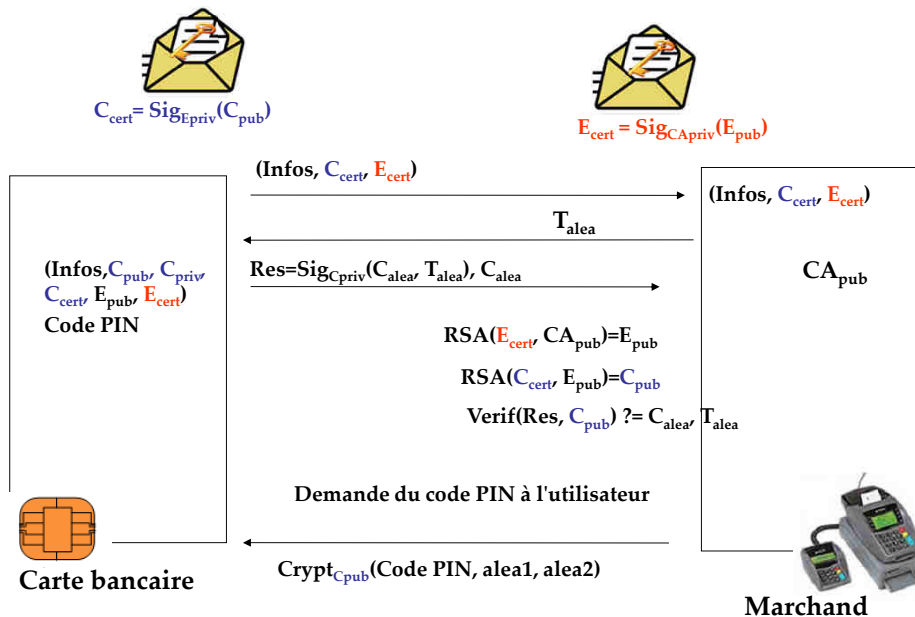
- Pendant la phase de personnalisation, la carte reçoit les informations suivantes :
 - Le nom du porteur, le numéro de la carte ou encore la date limite de validité de celle-ci (notés *Information*).
 - une paire de clés RSA (C_{pub} , C_{priv})
 - un certificat (C_{cert}) contenant C_{pub} signée par l'émetteur
 - le certificat de l'émetteur (E_{cert}) contenant sa clé publique E_{pub} signée par une autorité de certification
 - le code PIN transmis au porteur de cette carte.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

185

185

Lors de l'utilisation



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

186

186

DDA : à l'utilisation

- Avant toute transaction :
 - la carte fournit au terminal Informations, le certificat E_{cert} de la banque émettrice, et son certificat C_{cert}
 - le terminal génère une valeur aléatoire T_{alea} et l'envoie à la carte
 - la carte génère une valeur aléatoire C_{alea} . Puis, elle signe T_{alea} et C_{alea} avec sa clé privée C_{priv} . Elle envoie le résultat de la signature et C_{alea} au terminal.
 - le terminal vérifie E_{cert} avec CA_{pub} et vérifie C_{cert} avec E_{pub} . Puis, il vérifie la signature des aléas avec C_{pub} .
 - le terminal demande à l'utilisateur le code PIN et le transmet (**chiffré par C_{pub}**) à la carte pour qu'elle le vérifie. Le code PIN est d'abord concaténé avec deux nouvelles valeurs aléatoires fournies par la carte et le terminal, afin d'éviter les attaques par rejeu.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

187

187

Mode de gestion des transactions

- La transaction est finalisée en ligne ou hors ligne, choix fait par la carte ou le terminal selon une politique de gestion de risques :
 - sélection aléatoire
 - validation en ligne pour n validations hors ligne
 - en fonction du montant de la transaction
 - en fonction du montant cumulé des transactions déjà effectuées hors ligne ou d'un plancher fixé par le marchand.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

188

188

Fonctionnement en ligne et hors ligne

- Une clé secrète (clé 3DES de 112bits) est utilisée :
 - unique par transaction
 - notée C_{trans} calculée à partir de C_{maitre} et d'un compteur de transactions (ATC: Application Transaction Counter)
 - C_{maitre} est une clé de la carte générée par la banque émettrice à partir d'une clé maître de la banque E_{maitre} et des informations bancaires
 - C_{maitre} est mise dans la carte lors de la personnalisation
 - ATC est un compteur sur deux octets géré par la carte et incrémenté à chaque transaction.

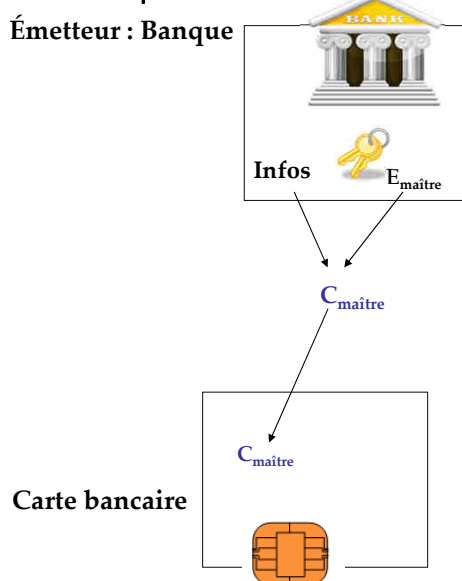
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

189

189

Transaction: personnalisation

Émetteur : Banque



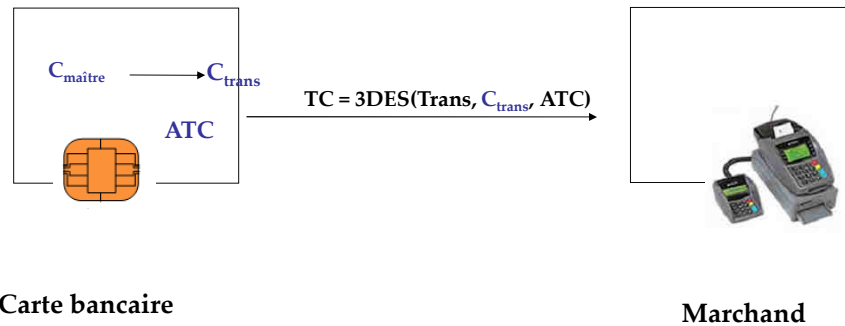
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

190

190

Transaction: utilisation

Dérivation d'une clé unique par transaction:



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

191

191

Fonctionnement hors ligne

- Le terminal envoie à la carte les détails de la transaction
- La carte produit alors un **certificat de transaction TC** en signant ces données (algorithme DES CBC-MAC) à l'aide de C_{trans} .
- Le terminal ne peut pas vérifier TC mais le garde pour validation ultérieure auprès de sa banque.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

192

192

Fonctionnement en ligne

- Le terminal envoie à la banque émettrice **le cryptogramme généré par la carte** (ARQC Authorization ReQuest Cryptogram).
- La banque le vérifie et génère un cryptogramme réponse (ARPC Authorization ResPonse Cryptogramm) envoyé à la carte via le terminal
- Le terminal redemande alors à la carte de lui générer un **certificat de transaction TC** qui inclut l'autorisation de la banque.

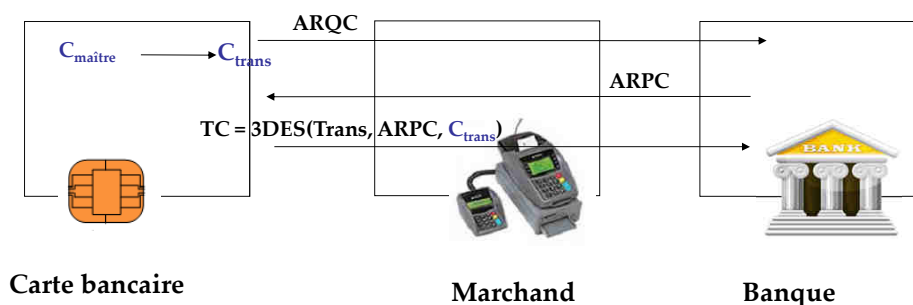
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

193

193

Transaction en ligne

Dérivation d'une clé unique par transaction:

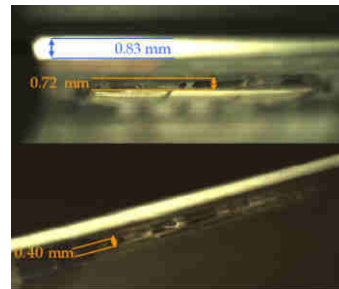
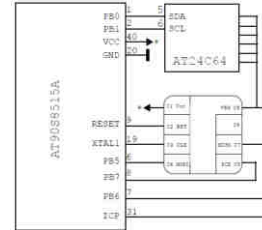
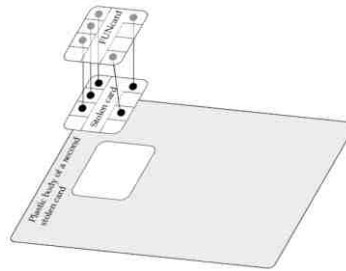


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

194

194

La Yes card est toujours possible



- Voir :
 - <https://eprint.iacr.org/2015/963.pdf>
 - <http://sec.cs.ucl.ac.uk/users/smurdoch/papers/oakland14chipandskim.pdf>
 - <https://www.cl.cam.ac.uk/research/security/banking/nopin/oakland10chipbroken.pdf>

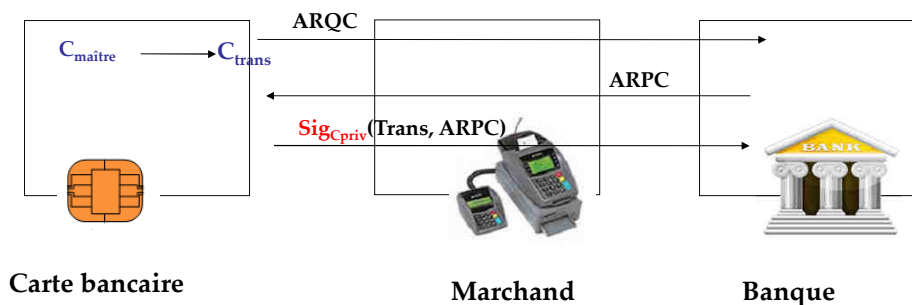
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

197

197

Authentification CDA

- CDA : Combined Data Authentication, variante de DDA
- utilise TC
- inclut TC dans le bloc de données signé par la carte avec sa clé privée



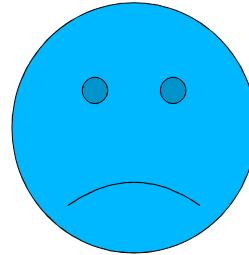
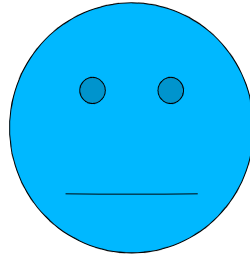
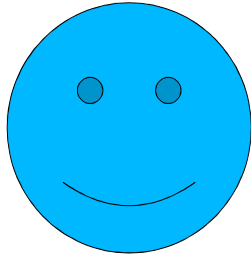
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

198

198

La Yes card est toujours possible

- Sauf si :
 - Le terminal vérifie la signature puis les champs devant contenir si la carte a oui ou non fait la vérification de PIN.



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

199

199

Book 3 : Application specifications

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

200

200

Book III : Spécification de l'application

- Partie 1: Les données et les commandes
- Partie 2: Flux transactionnel

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

201

201

Structure des fichiers

- Les fichiers d'un ICC sont vus du terminal comme une structure arborescente. Chaque branche de l'arbre est :
 - Un ADF (Application Definition File)
 - Un ADF est un point d'entrée pour un ou plusieurs AEF
 - AEF (Application Elementary Files) : fichiers élémentaires
 - ou un DDF (Directory Definition File)
 - Un DDF est un point d'entrée à d'autres ADF ou DDF
- Un ADF est associé à chaque application

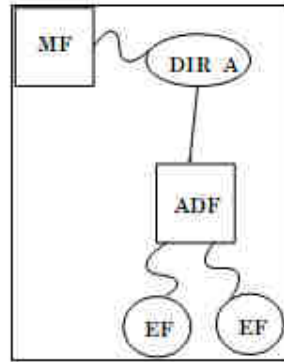
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

202

202

Exemple simple d'arborescence

- MF (Master File) : fichier maître, c'est le répertoire racine identifié par '3F 00'



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

203

203

Identification et nommage des fichiers

- Identification par nom
 - un DDF ou un ADF peut être référencé à l'aide d'un nom.
 - Le nom d'un ADF correspond à l'AID ou bien contient l'AID comme début du nom
 - Chaque nom doit être unique au sein d'une carte
- Identification par SFI (Short File Identifier)
 - Les SFI sont utilisés pour sélectionner des AEF.
 - Chaque AEF d'une application est identifié par un SFI à 5 bits ayant des valeurs entre 1 et 30.
 - Un SFI doit être unique dans une application

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

204

204

READ RECORD

- Lit les enregistrements d'un fichier à structure linéaire (basé sur les enregistrements)
- L'ICC retourne l'enregistrement

Code	Value
CLA	'00'
INS	'B2'
P1	Record number
P2	Reference control parameter (see Table 39)
Lc	Not present
Data	Not present
Le	'00'

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
x	x	x	x	x				SFI
					1	0	0	P1 is a record number

Table 39: READ RECORD Command Reference Control Parameter

205

205

PSE (Payment System Environment)

- Un système de paiement comprend :
 - Un ensemble de fichiers dans l'ICC
 - Des données au niveau du terminal fournies par le marchand
 - Un protocole d'application compris par l'ICC et le terminal
 - Les applications sont identifiées par AIDs de manière unique conformément à l'ISO7816-5
 - Pour les cartes à contact : 1PAY.SYS.DDF01
 - Pour les cartes sans contact : 2PAY.SYS.DDF01 (PPSE pour Proximity PSE)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

206

206

Identification d'applications de paiement

- Elle définit la procédure d'enregistrement et d'attribution des identifiants des applications (AID, ou Application IDentifier).
- Un unique AID est associé à chaque application = {RID, PIX}
- RID : numéro unique par fournisseur d'application
- Regardez vos facturettes !

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

207

207

Interopérabilité

- Le terminal doit être capable de travailler avec tous les ICC supportant des PSE
- Conformité avec la norme ISO
- Les ICCs doivent être capables de supporter des applications multiples, pas uniquement des applications de paiement

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

208

208

Commandes spécifiques à l'EMV

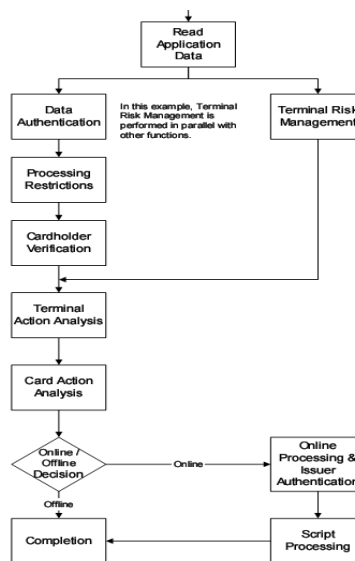
- GET PROCESSING OPTION : initie la transaction dans la carte
- GENERATE AC : produit un cryptogramme
- APPLICATION BLOCK : bloque l'application courante
- APPLICATION UNBLOCK : débloquent une application bloquée
- CARD BLOCK : bloque la carte
- PIN CHANGE/UNBLOCK: change ou débloquent le code PIN

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

209

209

Flux d'une transaction



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

210

210

Application Selection

- Le terminal choisit l'application en sélectionnant l'AID de l'appli (MasterCard, Visa, etc.)
- L'ADF est sélectionné
- La carte retourne le FCI (File Control Information) de l'ADF

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

211

211

Initiate Application Processing

- Informe l'ICC que le traitement d'une nouvelle transaction va commencer
- Opération effectuée après la sélection de l'application
- Obtient de la carte le profile d'échange (AIP: Application Interchange Profile) et la liste des fichiers à utiliser durant la transaction : SDA ou DDA supporté, vérification du détenteur de la carte, authentification de l'émetteur, etc.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

212

212

Read Application Data

- Le terminal lit les données contenues dans les fichiers de la carte et utilise les données de la carte pour vérifier l'authentification SDA ou DDA
- Le terminal utilise Read Record
- Opération effectuée après l'opération Initiate Application Processing

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

213

213

Processing restrictions

- Vérification :
 - du numéro de version de l'application
 - de l'Application Usage Control
 - des dates d'expiration de l'application

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

214

214

Application Usage Control

If:	and if Issuer Country Code:	then the following bit must be set to 1 in Application Usage Control:
Transaction Type indicates cash transaction	matches Terminal Country Code	'Valid for domestic cash transactions'
	does not match Terminal Country Code	'Valid for international cash transactions'
Transaction Type indicates purchase of goods	matches Terminal Country Code	'Valid for domestic goods'
	does not match Terminal Country Code	'Valid for international goods'
Transaction Type indicates purchase of services	matches Terminal Country Code	'Valid for domestic services'
	does not match Terminal Country Code	'Valid for international services'
transaction has a cashback amount	matches Terminal Country Code	'Domestic cashback allowed'
	does not match Terminal Country Code	'International cashback allowed'

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

215

215

Cardholder Verification

- Vérification du détenteur de la carte
- La carte doit être capable de supporter au moins une méthode de vérification de détenteur de carte (CVM: Cardholder Verification Method). Ceci doit être spécifié dans le profile d'échange envoyé par la carte. Le terminal doit utiliser cette méthode.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

216

216

Codes CVM

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							Fail cardholder verification if this CVM is unsuccessful
	1							Apply succeeding CV Rule if this CVM is unsuccessful
		0	0	0	0	0	0	Fail CVM processing
		0	0	0	0	0	1	Plaintext PIN verification performed by ICC
		0	0	0	0	1	0	Enciphered PIN verified online
		0	0	0	0	1	1	Plaintext PIN verification performed by ICC and signature (paper)
		0	0	0	1	0	0	Enciphered PIN verification performed by ICC
		0	0	0	1	0	1	Enciphered PIN verification performed by ICC and signature (paper)
		0	x	x	x	x	x	Values in the range 000110-011101 reserved for future use by this specification
		0	1	1	1	1	0	Signature (paper)
		0	1	1	1	1	1	No CVM required
		1	0	x	x	x	x	Values in the range 100000-101111 reserved for use by the individual payment systems
		1	1	x	x	x	x	Values in the range 110000-111110 reserved for use by the issuer
		1	1	1	1	1	1	This value is not available for use

217

217

PIN Verification

- La vérification du PIN
 - peut se faire en ligne ou hors ligne selon les valeurs de CVM
 - peut nécessiter le chiffrement du PIN
 - peut être complétée par une signature manuelle
- VERIFY PIN : vérification du code PIN (crypté ou non) hors ligne
- GET CHALLENGE: le terminal demande un nombre aléatoire (nb) à la carte

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

218

218

Terminal Risk Management

- Mécanismes utilisées pour éviter la fraude :
 - floor limit checking : vérification d'informations relatives à la transaction sur le terminal (numéro de compte, montant de la transaction, date de la transaction, etc.)
 - velocity checking : vérifier que les transactions effectuées hors ligne seront complétées en ligne.
 - algorithmes de gestion des risques propres à l'émetteur de la carte

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

219

219

Card Action Analysis

- Comme conséquence à la gestion des risques :
 - la carte peut accepter une transaction hors ligne en générant un certificat de transaction TC.
 - la carte peut demander une autorisation en ligne en générant ARQC.
 - la carte peut décider de rejeter la transaction.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

220

220

Online Processing

- Online Processing : sert à s'assurer que l'émetteur de la carte peut réviser, autoriser ou rejeter les transactions.
- EXTERNAL AUTHENTICATE : le terminal envoie les données d'authentification de la banque
- La carte vérifie les données d'authentification et génère un cryptogramme.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

221

221

Script Processing

- Script Processing : ne fait pas partie de l'application
- Ensemble de commandes adressées à la carte en mode administrateur :
 - PIN CHANGE/ UNBLOCK
 - APPLICATION BLOCK/UNBLOCK
 - CARD BLOCK.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

222

222

Book 4

- Besoins fonctionnels et caractéristiques physiques
- Gestion des données et du logiciel
- Interfaces utilisées

Sur Internet

- L'utilisation des 16 chiffres visibles de la carte n'est pas un protocole de paiement sûr.
- Ce numéro à 16 chiffres n'est plus imprimé sur les factures
- 9 chiffres sont encore inscrits sur les factures et correspondent à peu près à l'aléa choisi par la banque. => possibilité de reconstituer le numéro entier
- En 2001, un cryptogramme a été rajouté et imprimé uniquement sur la carte, appelé CVV (Card Verification Value) chez Visa et CVC (Card Validation Code) chez MasterCard
- Ce code est généré par la banque à partir des informations bancaires du client et de données secrètes de la banque
- Ce code ne peut être reconstruit car l'algorithme est secret
- Depuis 2004, le cryptogramme doit être demandé par tout site marchand en plus du numéro de la carte afin de valider toute transaction à distance auprès de la banque émettrice.
- Ce numéro n'est écrit nulle part sauf sur la carte.

Conclusion

- L'authentification DDA n'est pas obligatoire et
- DDA est plus robuste mais plus chère
- En France, la majorité des cartes aujourd'hui supportent DDA
- L'authentification SDA :
 - Code PIN envoyé en clair
 - Clé RSA de 1984 bits mais la donnée d'authentification peut être lue sans présentation de code
- Aujourd'hui, il manque des liens entre l'authentification, la vérification du code PIN et la génération de TC : les requêtes envoyées par le terminal peuvent être interceptées en forgeant des réponses à envoyer
- Nouvelle tendance de paiement : paiement sans contact (ISO 14443) => nouveaux systèmes, nouveaux protocoles et nouvelles failles ...

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

225

225

Les cartes prépayées

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

226

226

- Carte Prépayée

- Cas particulier du PME : la carte contient des jetons (et non de l'argent)
- Ex : Abonnement autoroute (Marseille), Places de cinéma, ...
- Une réussite : la Télécarte

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

227

227

Analyse du succès de la télécarte

- C'est une carte prépayée.
- Elle contient des jetons (et non de l'argent)
- Les raisons de sa réussite :
 - Fin du vandalisme des cabines téléphoniques
 - Temps de communication +50%



- Support publicitaire



- Avance de trésorerie



eron

228

228

Bilan

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

229

229

Applications Actuelles des Cartes

- Différents secteurs porteurs
 - Les Télécommunications : Cartes Pré-payées, carte SIM
 - Recherche tj + de services, $\forall$ le coût
 - Les Banques : Porte-monnaie Électronique, Carte Bancaire
 - Recherche de coût de fonctionnement le plus faible
 - La Santé : Carte patient, carte professionnels de santé
 - Recherche la sécurité d'informations sensibles
- Sous l'impulsion du GSM,
la carte doit offrir de + en + de services

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

230

230

Le futur de la carte à puce

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

231

231

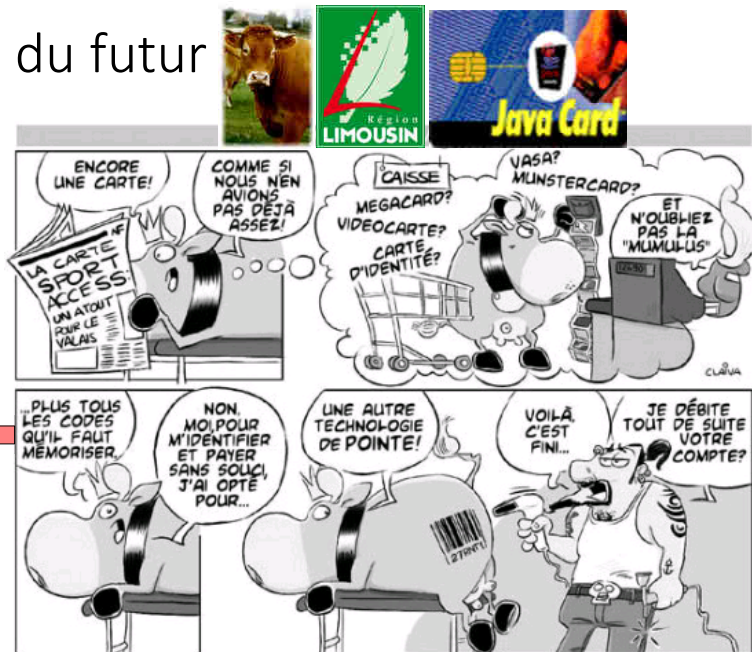
Les cartes à puce du futur

Des cartes ... encore ... toujours ...
... toujours plus ...
... partout ...

Comment résoudre
ce problème ?



La carte multi-applicative



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

232

232

Oyster Card et Barclays

- Oyster

- Facile et pratique pour voyager pas cher dans Londres.
- Chargement de crédits ou carte d'abonnement.
- C'est une fonctionnalité totalement séparé du compte associé à la carte de crédit



- Cashless

- Technologie Visa "wave and pay" pour des petits achats (inférieurs à £10) sans PIN.
- Les transactions apparaissent sur le relevé de compte.



- Crédit - Paiement

- Application Visa classique avec PIN pour les paiements supérieurs à £10.



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

233

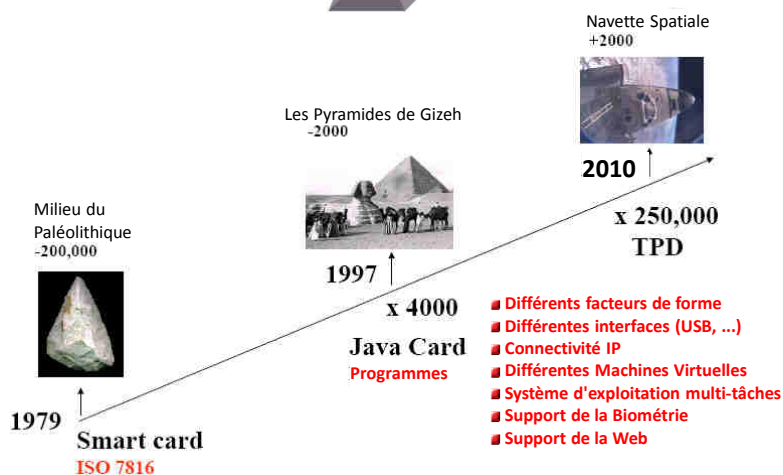
233

Le difficile équilibre

Sécurité

Fonctionnalités

LOI DE MOORE

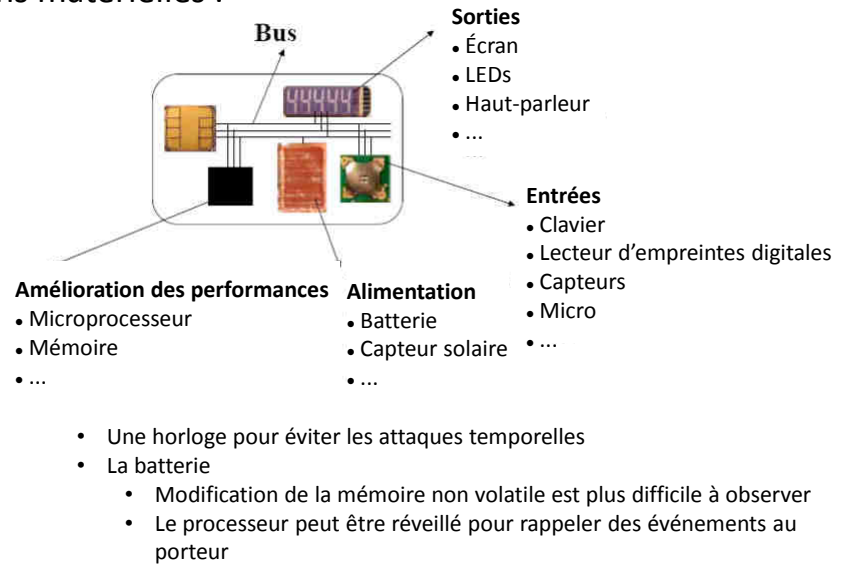


234

234

Les cartes du futur

• Quelques évolutions matérielles :



235

La sécurité

Le haut niveau de sécurité est assuré par la triple alliance de l'électronique, de l'informatique et de la cryptographie



236

La sécurité physique

- techniques d'impression sophistiquées
 - différentes couches d'impression
 - hologramme
 - embossage
- procédés d'encollage
- matériaux plastiques
- design du micro-contact

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

237

237

La sécurité hardware

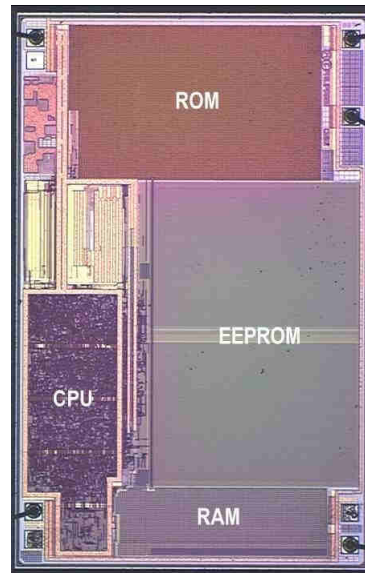
- Le micromodule est monolithique => difficile de distinguer ses différents composants
- un numéro de série unique
- l'utilisation de mémoire de type PROM
- blindage physique du composant (grille, grille active, ...)
- des détecteurs de conditions anormales (tension, fréquence, température, lumière, ...)
- brouillage des informations dans le composant (mémoire, bus, ...)
- co-processeurs cryptographiques
- pompe de charge pour le lissage de la consommation
- dual logic, circuit asynchrone

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

238

238

Une puce

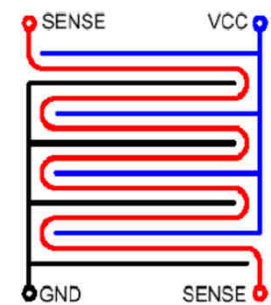
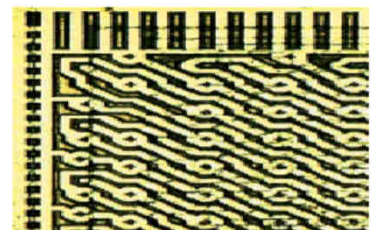
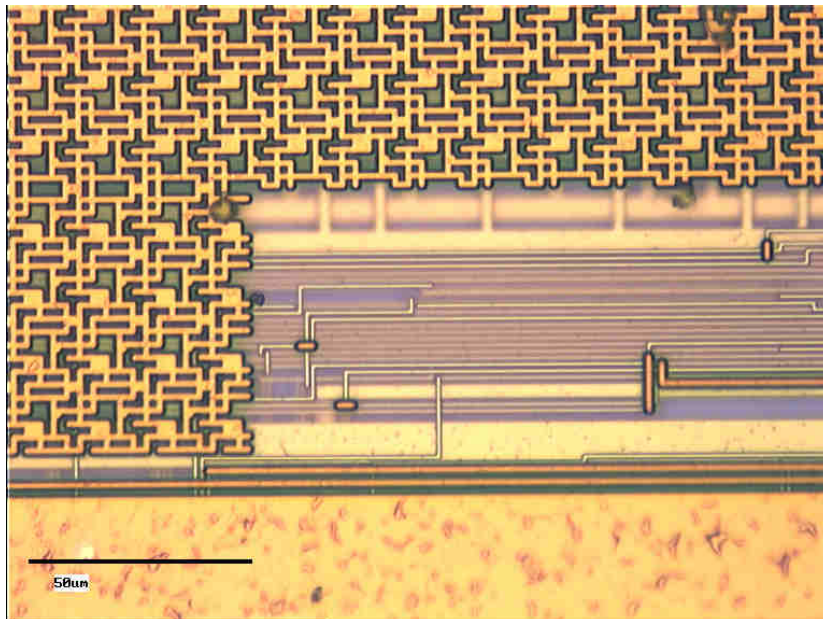


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

239

239

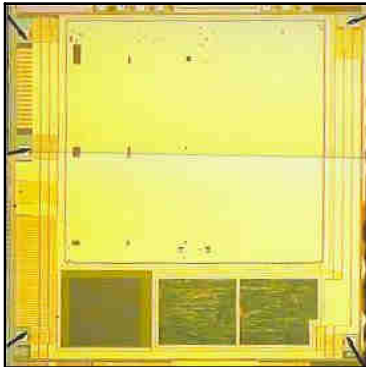
La grille de protection (active ou non)



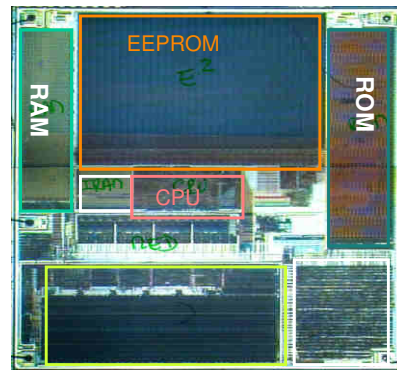
240

240

Protégeons la puce



- Shield
- Scrambled glue logic
- No Buses visible



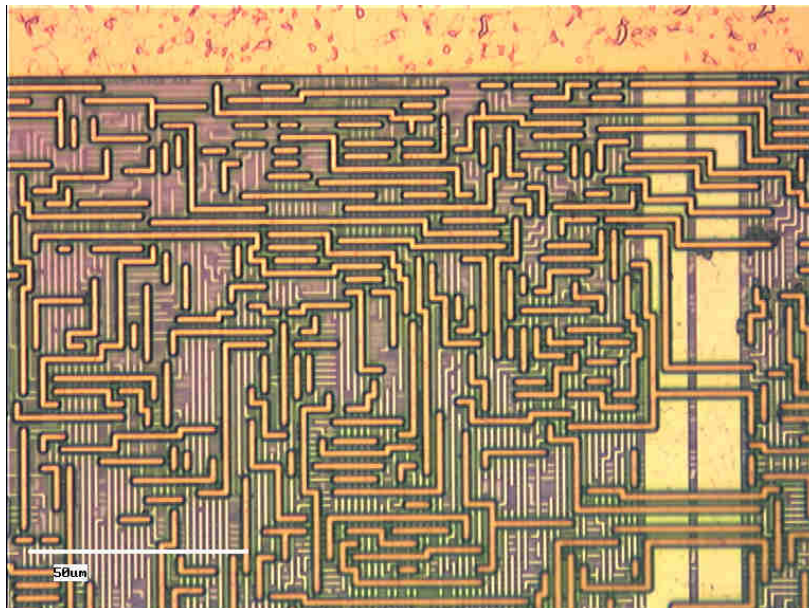
- Blocks easily identified
- No shield
- No glue logic
- Buses clearly visible

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

241

241

Le CPU

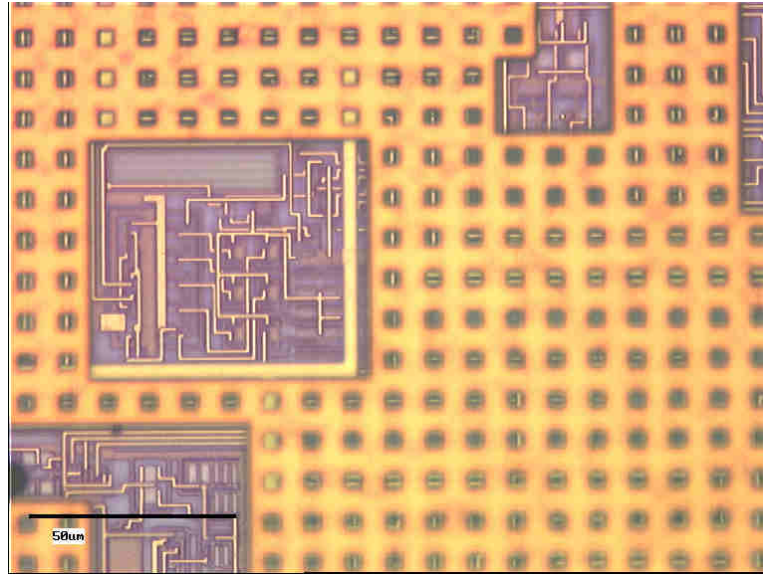


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

242

242

La partie analogique

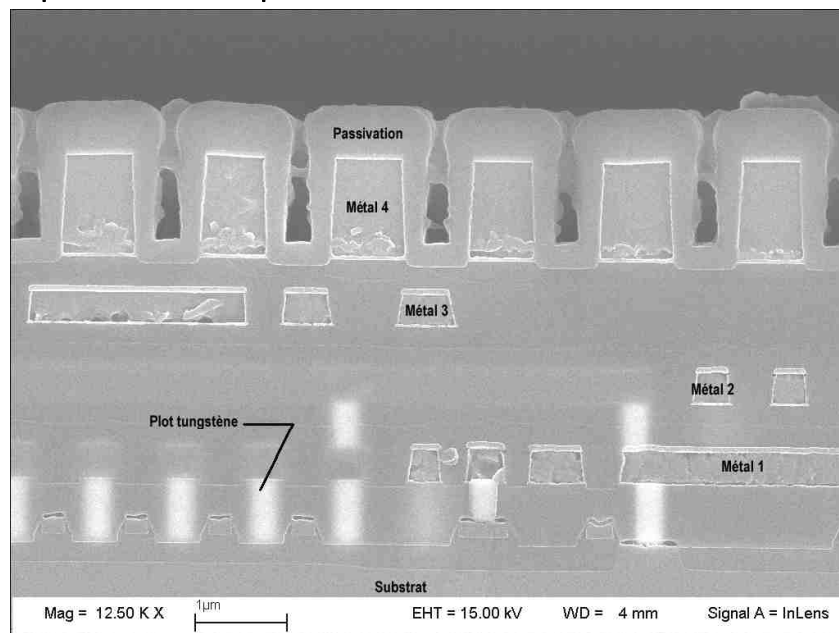


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

243

243

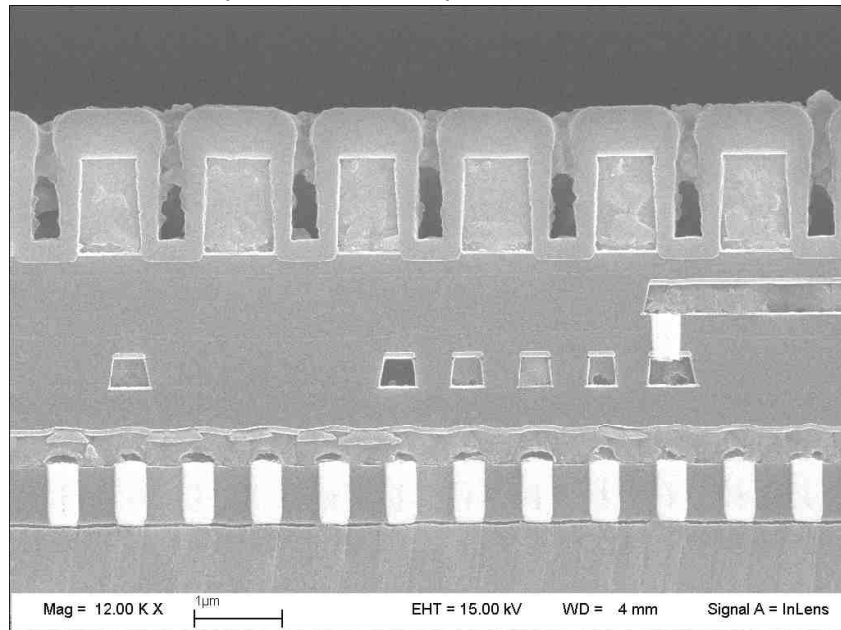
La coupe d'une puce



244

244

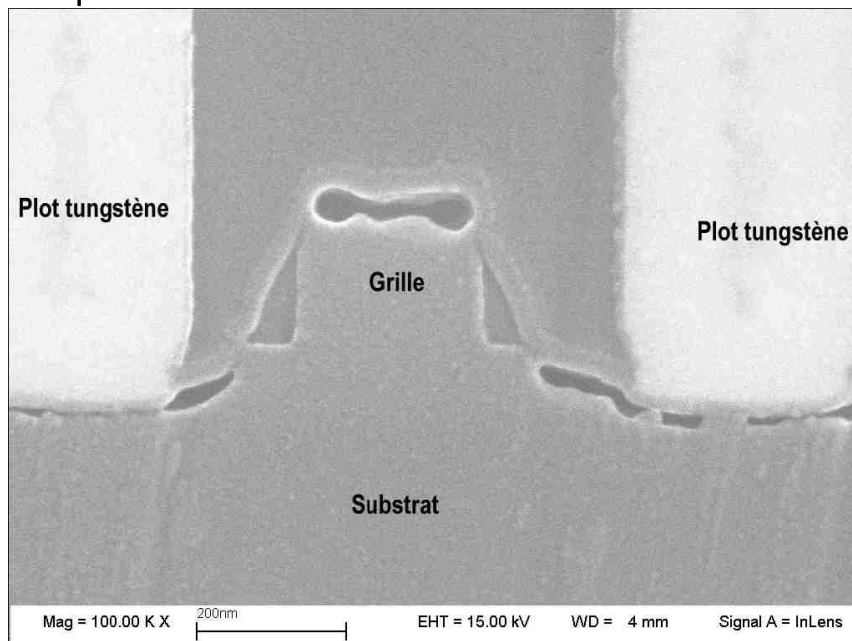
Une autre coupe d'une puce



245

245

La coupe d'un transistor



246

246

La sécurité software

- contrôles d'accès aux données
- maintien de l'intégrité des données
- entrées/sorties sécurisées
- migration du code

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

247

247

Et pour encore plus de sécurité et de confiance

- La sécurité de l'environnement de production
 - physiquement sécurisé
 - régulièrement audité
- Les méthodes formelles et le test
 - Conception formelle de certains partie de l'OS (voire de l'OS complet)
 - Preuve formelle sur des aspects sécuritaires
 - Génération de jeux de tests associés aux spécifications
 - ...
- L'évaluation sécuritaire et la certification

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

248

248

L'évaluation sécuritaire et la certification

• La certification

• Objectifs :

- Garantir le niveau de sécurité d'un produit TI (confidentialité, intégrité, disponibilité).
- Bénéficier des accords de reconnaissance mutuelle.

• Avantages pour :

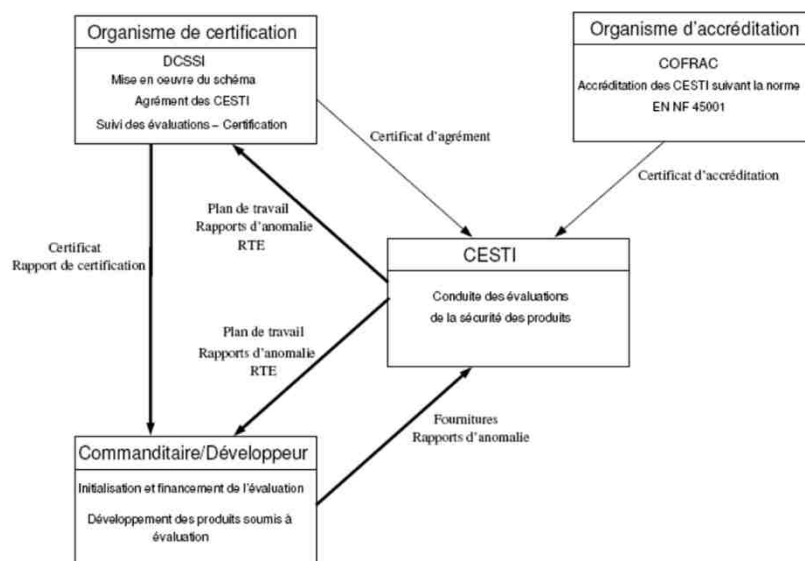
- les utilisateurs : comparer sur une base objectif ;
- les industriels : prouver leur compétence et d'étendre leurs marchés ;
- les autorités d'homologation : s'assurer que les objectifs de sécurité sont satisfaits et que leur pays ne court pas de risque majeur à utiliser des produits TI !

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

249

249

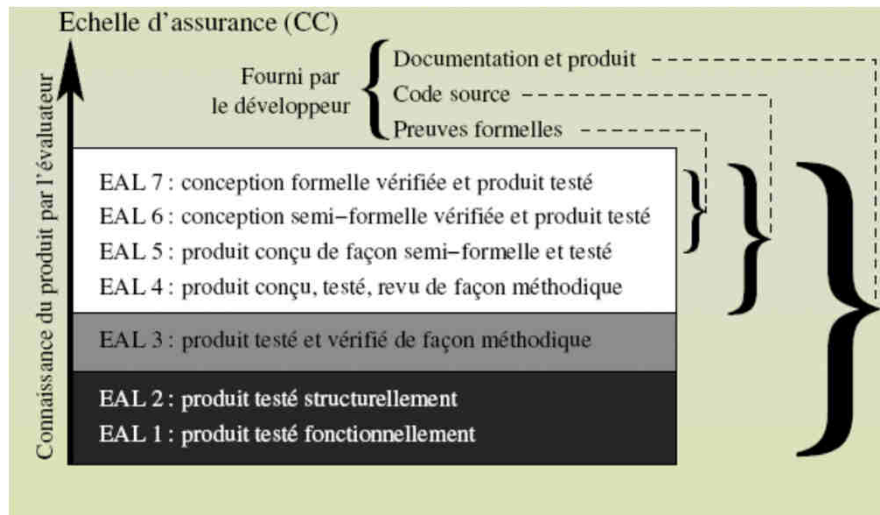
Le schéma Français



250

250

L'échelle d'assurance



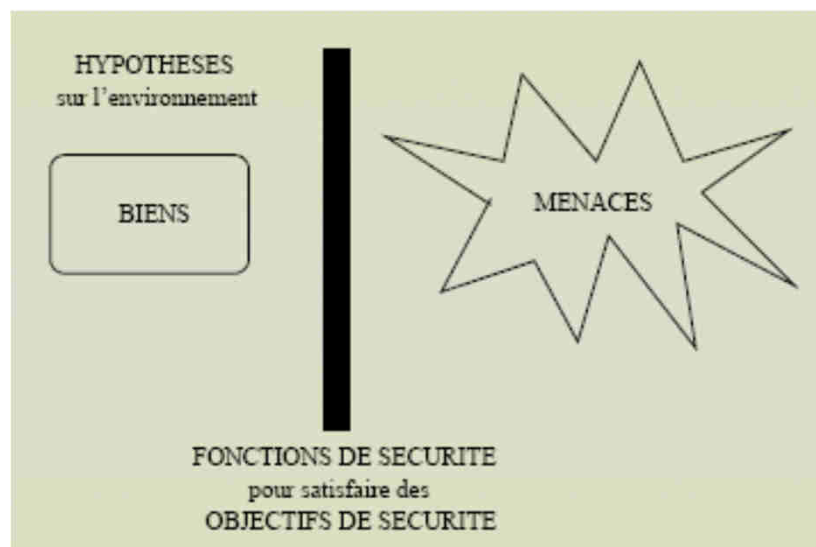
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

251

251

Les bases des critères communs

- Critères Communs (ISO15408)



252

252

Les attaques

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

253

253

Deux grandes catégories

- Non invasives
 - Le produit reste fonctionnel
- Invasives
 - Le produit est partiellement endommagé

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

254

254

Quelques attaques non invasives

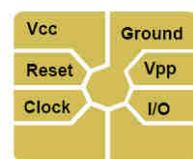
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

255

255

Des attaques non invasives

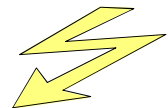
- modification des conditions opérationnelles (Vcc, F)



- Attack on VPP
- Using nail polish
- Card not debited...

- modification de la température
- modification des rayonnements lumineux (UV, rayon X, lumière blanche, IR, ...)

- injection de fautes (glitches, rayonnements lumineux)
 - (injection de fautes sur la JVM : <http://www.cs.princeton.edu/~sudhakar/papers/>)



- attaques sur les **canaux cachés** :
 - Temps
 - Consommation électrique
 - Émissions électromagnétiques

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

256

256

Injection de fautes

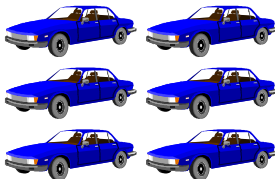
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

257

257

Êtes vous prêt ?

- Politique : Les jouets cassés ne sont pas payés



voiture = \$3



avion = \$5

Jack



Dino



Dino achète des jouets à Jack

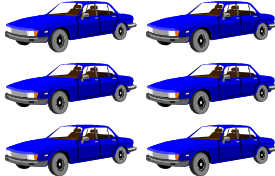
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

258

258

Êtes vous prêt ?

- Politique : Les jouets cassés ne sont pas payés



voiture = \$3



avion = \$5

Jack



Je voudrais acheter
3 avions

Dino



Dino achète des jouets à Jack

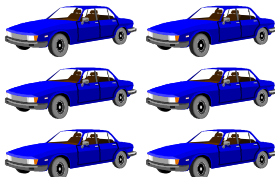
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

259

259

Êtes vous prêt ?

- Politique : Les jouets cassés ne sont pas payés

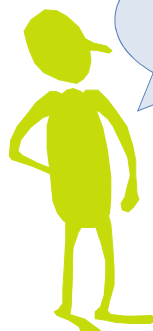


voiture = \$3



avion = \$5

Jack



Ça fera \$15

Je voudrais acheter
3 avions

Dino



Dino achète des jouets à Jack

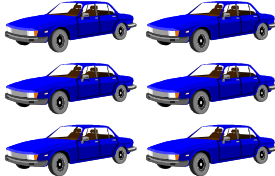
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

260

260

Êtes vous prêt ?

- Politique : Les jouets cassés ne sont pas payés



voiture = \$3



avion = \$5

Jack

Ça fera \$15

OK, s'il vous plaît
envoyez le par DHL

Dino

Dino achète des jouets à Jack

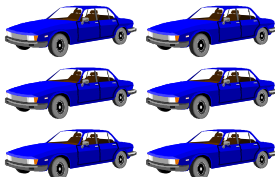
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

261

261

Êtes vous prêt ?

- Politique : Les jouets cassés ne sont pas payés



voiture = \$3



avion = \$5

Jack

Comment payez vous ?

OK, s'il vous plaît
envoyez le par DHL

Dino

Dino achète des jouets à Jack

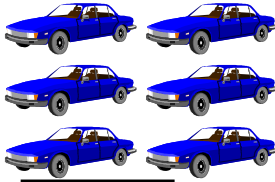
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

262

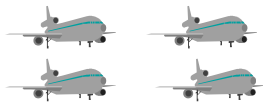
262

Êtes vous prêt ?

- Politique : Les jouets cassés ne sont pas payés



voiture = \$3



avion = \$5

Jack

Comment payez vous ?

J'enverrai \$15
en mandat postal

Dino

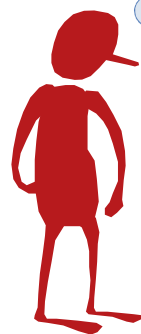
Dino achète des jouets à Jack

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

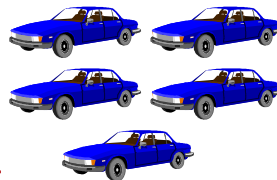
263

263

Le facteur veut savoir ce que Dino a acheté pour \$15

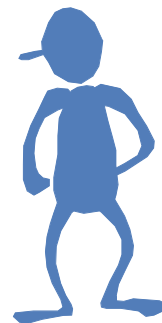


Qu'est ce qu'il a
acheté pour \$15 ?



Méchant facteur

Dino

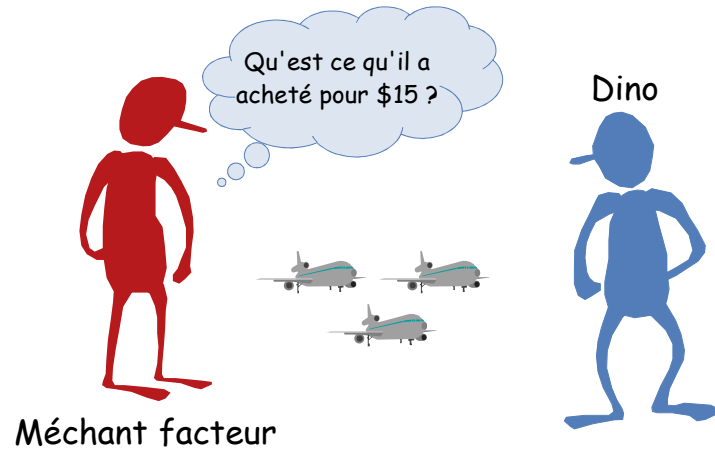


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

264

264

Le facteur veut savoir ce que Dino a acheté pour \$15

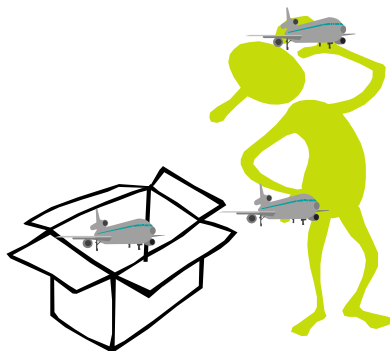


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

265

265

Pendant ce temps, Jack prépare le colis DHL

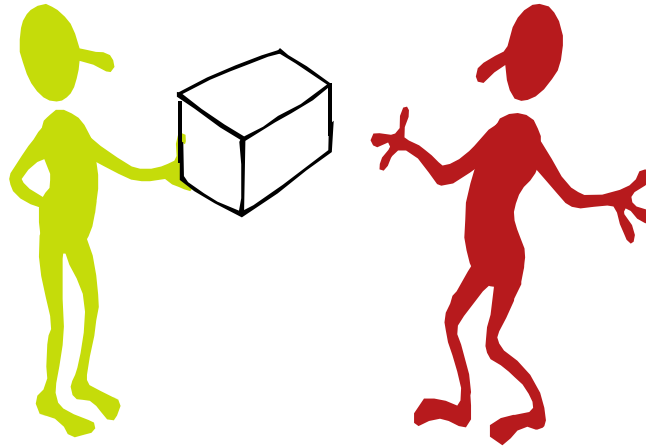


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

266

266

Et le donne au facteur



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

267

267

Qui tape dedans assez fort pour casser **un** jouet

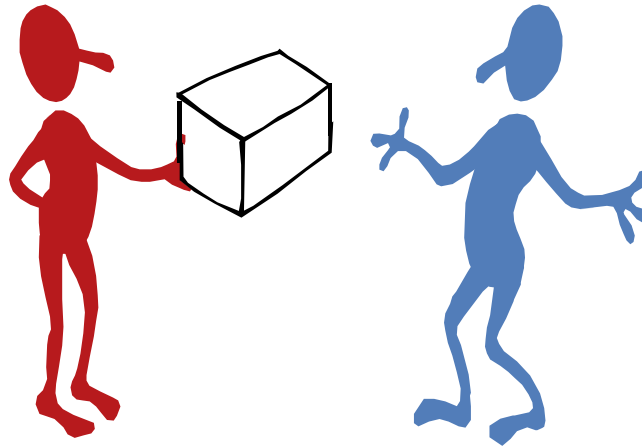


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

268

268

Et le donne à Dino



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

269

269

Une semaine plus tard, il surveille le mandat postal de Dino...



- La leçon apprise : Les attaques en faute peuvent aussi permettre d'extraire des secrets de tokens!

- Les fautes matérielles peuvent être venir de diverses sources :
 - Glitches de tension, faisceau lumineux, faisceau laser ...
- Utilisable sur la signature RSA, le chiffrement DES, ...
- Il y a des détecteurs lasers embarqués sur les puces récentes !
 - (voir thèse d'Alexandre SARAFIANOS)

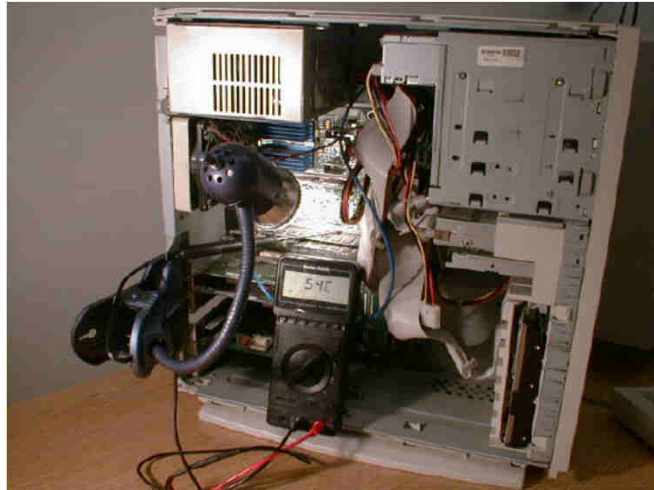
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

270

270

Attaques combinées (software/hardware)

- Pour forger des pointeurs (réaliser une confusion de types) et ensuite ...



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

271

271

Les canaux cachés

On parle aussi de canaux auxiliaires

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

272

272

Canaux cachés

- le temps d'exécution
 - nombre de cycles d'une instruction ou d'un algorithme
- la consommation de courant
 - Les modifications rapides de la tension et de l'intensité du courant au sein du même composant sont à la base des émissions du circuit car ils conduisent des courants RF à l'intérieur et à l'extérieur du chip.
- les émissions électromagnétiques
 - Les courants RF entraînent un rayonnement électromagnétique.
- le son
- les émissions de photons

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

273

273

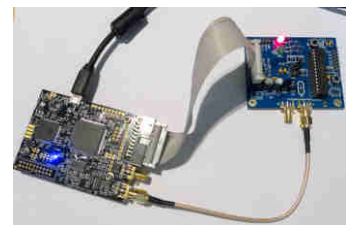
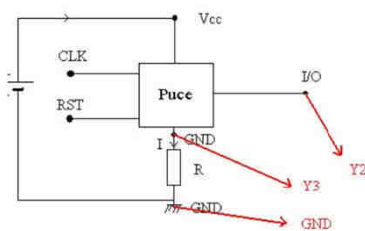
Le matériel pour les attaques par canaux cachés

• Mesure :

- du temps d'exécution
- de la consommation électrique
- des émissions électromagnétiques
- du son

• Équipements

- un oscilloscope numérique,
- un pc équipé de cartes d'acquisition et de logiciels mathématiques pour le traitement des données



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

274

274

La timing attack

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

275

275

La timing attack

- Cette attaque consiste à mesurer le temps d'exécution d'un algorithme.
 - Elle révèle des informations sur les opérations et/ou les opérandes.
 - Elle nécessite souvent :
 - un grand nombre d'exécution à messages choisis,
 - un traitement statistique des résultats obtenus.

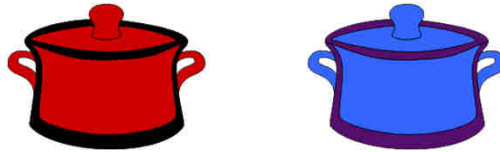
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

276

276

Un jeu

- Mettez 28 dans un des pots et 10 dans l'autre :



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

277

277

Un jeu

- Mettez 28 dans un des pots et 10 dans l'autre :



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

278

278

Un jeu

- Mettez 28 dans un des pots et 10 dans l'autre :



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

279

279

Un jeu

- Mettez 28 dans un des pots et 10 dans l'autre :



- Je vous demande de :
 - multiplier le contenu du pot rouge par 7
 - multiplier le contenu du pot bleu par 10
 - d'additionner les deux résultats
 - de me dire si la somme est paire ou impaire.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

280

280

Un jeu

- Mettez 28 dans un des pots et 10 dans l'autre :



- Je vous demande de :
 - multiplier le contenu du pot rouge par 7
 - multiplier le contenu du pot bleu par 10
 - d'additionner les deux résultats
 - de me dire si la somme est paire ou impaire.
- Est-ce que votre réponse est suffisante pour révéler le contenu de chaque pot ?

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

281

281

Est-ce que ce jeu peut m'aider?

- Et bien, normalement non :

$$28 \times 7 + 10 \times 10 = 296$$

est un nombre pair

et

$$10 \times 7 + 28 \times 10 = 350$$

est aussi un nombre pair...

- Pourtant, juste en observant le temps pris pour donner la réponse (le calcul mental conduisant à 296 est plus compliqué que celui conduisant à 350), on peut dire quelles valeurs étaient dans quels pots !

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

282

282

Conclusion

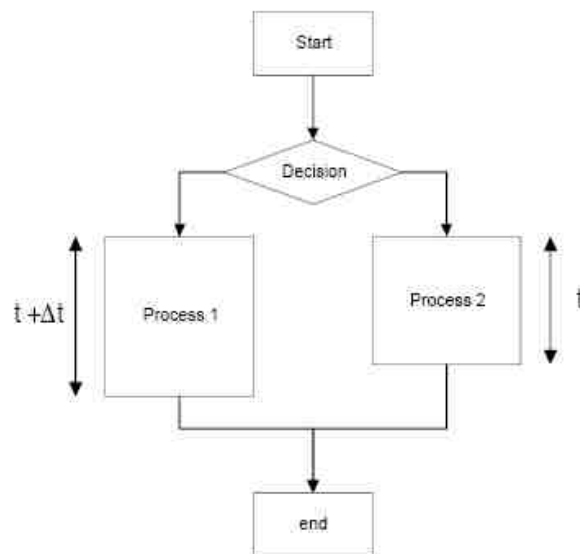
- Une observation externe du temps de calcul d'une carte peut conduire à la fuite d'informations secrètes vers l'extérieur (par exemple des clés, des PINs, etc.).
- Les **timing attacks** sont apparues au début des années 1990.
- La leçon apprise : **Les logiciels actuels pour cartes s'exécute en temps constant.**

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

283

283

La « timing attack »



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

284

284

La "timing attack" sur le PIN (1/2)

```
for ( i = 0 ; i <= 7; i++)
    if ( pinCarte [ i ] != pinPresente [ i ] )
        return false ;
return true ;
```

 **pinCarte**
1,3,2,4,250,7,9,134

- Présentons un PIN quelconque et déroulons le programme :

- cas du premier octet FAUX

```

i = 0
i <= 7 ? OUI
( pinCarte [ i ] != pinPresente [ i ] ) ? OUI
return false
```

pinPresente
0,0,0,0,0,0,0,0

- Cas du premier octet VRAI

```

i = 0
i <= 7 ? OUI
( pinCarte [ i ] != pinPresente [ i ] ) ? NON
i++
i <= 7 ? OUI
( pinCarte [ i ] != pinPresente [ i ] ) ? OUI
return false
```

Δt (time difference between the two cases)

pinPresente
1,0,0,0,0,0,0,0

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

285

285

La "timing attack" sur le PIN (2/2)

```
for ( i = 0 ; i <= 7; i++)
    if ( pinCarte [ i ] != pinPresente [ i ] )
        return false ;
return true ;
```

 **pinCarte**
1,3,2,4,250,7,9,134

- Présenter les n valeurs possibles de `pinPresente[0]` (256 valeurs) (n,0,0,0,0,0,0,0).
- Mesurer la durée d'exécution de la commande T pour les n valeurs.
 - Calculer $T[n0]$ le maximum des T
soit $T[n0] = \max(T[n]); n = 0; \dots; 255$
n0 est la solution pour `pinCarte[0]`
- Itérer sur tous les `pinPresente[i]`

pinPresente
0,0,0,0,0,0,0,0
1,0,0,0,0,0,0,0
...
255,0,0,0,0,0,0,0

- Nombre d'essais : $8 * 256 = 2048$ (contre 256^8 en force brute)
- On peut même s'arrêter avant $n=255$!

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

286

286

Exemples de protection (1/2)

```
alea = random ( 0 , 7 ) ; // alea situe dans l'intervalle [0 ; 7]
for ( i = 0 ; i <= 7; i++) {
    octet = ( alea + i ) mod 8 ;
    if ( pinCarte [ octet ] != pinPresente [ octet ] )
        return false ;
}
return true ;
```

- Nombre d'essais **en moyenne** : $8 * 2048$ (contre 256^8 en force brute)
- Possibilité d'attaques de type MasterMind.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

287

287

Exemples de protection (2/2)

```
boolean test = true ;
for ( i = 0 ; i <= 7; i++) {
    if ( pinCarte [ i ] != pinPresente [ i ] )
        test = test && false ;
    else
        test = test && true ;
}
return test ;
```

- Équivaut à :

```
boolean test = true ;
for ( i = 0 ; i <= 7; i++) {
    test = ( pinCarte [ i ] == pinPresente [ i ] ) && test ;
}
return test ;
```

- Et pas :

```
boolean test = true ;
for ( i = 0 ; i <= 7; i++) {
    test = test && ( pinCarte [ i ] == pinPresente [ i ] ) ;
}
return test ;
```

Des attaques en temps existent sur d'autres supports (e.g. PC).

Voir par exemple : *Cache-timing attacks on AES* ou *CACHE MISSING FOR FUN AND PROFIT*

288

288

Security Rules

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

289

289

And what about security ?

- Write **securely** a function that checks that an array is equal to another with less than 3 trials.

```
boolean verify (byte[] buffer, short ofs, byte len)
{...}
```

- Need a constant `maxTries` initialized at 3,
- A field which memorize the trial number, says `triesLeft`

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

290

290

```

boolean verify (byte[] buffer, short ofs, byte len)
{
    // No comparison if PIN is blocked
    if (triesLeft <= 0)
        return false ;

    // Main comparison
    for(short i=0; i < len; i++)
        if (buffer[ofs+i] != pin[i])
        {
            triesLeft-- ;
            authenticated[0] = false ;
            return false ;
        }

    // Comparison is successful
    triesLeft = maxTries ;
    authenticated[0] = true ;
    return true ;
}

```

Called by a friendly
programmer who has
checked len before ;-)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

291

291

```

boolean verify (byte[] buffer, short ofs, byte len)
{
    // No comparison if PIN is blocked
    if (triesLeft <= 0)
        return false ;

    // First decrements the number of remaining tries
    triesLeft-- ;

    // Main comparison
    boolean equal = true ;
    for(short i=0; i < len; i++)
        equal = (equal && (buffer[ofs+i] != pin[i])) ; // cst time

    if (!equal) {
        // Comparison failed
        authenticated[0] = false ;
        return false ;
    }
    else {
        // Comparison is successful
        triesLeft = maxTries ;
        authenticated[0] = true ;
        return true ;
    }
}

```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

292

292

```

boolean verify (byte[] buffer, short ofs, byte len)
{
    // No comparison if PIN is blocked
    if (triesLeft <= 0)
        return false ;

    // First decrements the number of remaining tries
    triesLeft-- ;

    // Main comparison
    boolean equal = true ;
    for(short i=0; i < len; i++)
        equal = (equal && (buffer[ofs+i] != pin[i])) ;// cst time

    if (!equal) {
        // Comparison failed
        authenticated[0] = false ;
        return false ;
    }
    else {
        // Comparison is successful
        triesLeft = maxTries ;
        authenticated[0] = true ;
        return true ;
    }
}

```

No!

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

293

293

```

boolean verify (byte[] buffer, short ofs, byte len)
{
    // No comparison if PIN is blocked
    if (triesLeft <= 0)
        return false ;

    // First decrements the number of remaining tries
    triesLeft-- ;

    // Main comparison
    boolean equal = true ;
    for(short i=0; i < len; i++)
        equal = ((buffer[ofs+i] != pin[i]) && equal) ;// cst time

    if (!equal) {
        // Comparison failed
        authenticated[0] = false ;
        return false ;
    }
    else {
        // Comparison is successful
        triesLeft = maxTries ;
        authenticated[0] = true ;
        return true ;
    }
}

```

Yes!

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

294

294

```

public final static short BOOL_TRUE = (short)0x5a5a ;
public final static short BOOL_FALSE = (short)0xa5a5 ;
short equal = BOOL_TRUE ;

...

// Main comparison
for(short i=0; i < len; i++)
    equal = (short)(equal &
        ((buffer[ofs+i] != pin[i])? BOOL_FALSE:  BOOL_TRUE));

if (equal == BOOL_TRUE) {
    // Comparison is successful
    triesLeft = maxTries ;
    authenticated[0] = true ;
    return true ;
}
else {
    ...

```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

295

295

```

// First checks the integrity of the variable
if (triesLeft != triesLeftBackup)
    takeCountermeasure() ;

// No comparison if PIN is blocked
if (triesLeft < 0)
    return false ;

```

Correct?

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

296

296

```
// First checks the integrity of the variable
if (triesLeft != triesLeftBackup)
    takeCountermeasure() ;

// No comparison if PIN is blocked
if (triesLeft < 0)
    return false ;
```

- It protects only against a writing between the last and the current one. If the attack is during the evaluation...
- RAM is safer than EEPROM

```
// Transfer in a local and check the integrity of the variable
byte t1 = triesLeft ;

if (t1 != triesLeftBackup)
    takeCountermeasure() ;

// No comparison if PIN is blocked
if (t1 < 0)
    return false ;
```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

297

297

```
boolean verify (byte[] buffer, short ofs, byte len) {
    byte t1 = triesLeft ;
    if (t1 != (short)(~triesLeftBackup)) takeCountermeasure() ;
    if (t1 < 0) return false;
    JCSystem.beginTransaction() ;
    triesLeft = --t1 ;
    triesLeftBackup++ ;
    JCSystem.commitTransaction() ;
    if (triesLeft != (short)(~triesLeftBackup)) takeCountermeasure() ;
    short equal = BOOL_TRUE ;
    for(short i=0; i < len; i++)
        equal = (short)(equal &
            ((buffer[ofs+i] != pin[i]) ? BOOL_FALSE : BOOL_TRUE));

    if (equal == BOOL_TRUE) {
        JCSystem.beginTransaction() ;
        triesLeft = maxTries ;
        triesLeftBackup = (byte)(~maxTries) ;
        JCSystem.commitTransaction() ;
        // Verifies the new value
        if (triesLeft != (short)(~triesLeftBackup))
            takeCountermeasure() ;
        authenticated[0] = true ;
        return true ;
    } else {
        authenticated[0] = false ;
        return false ;
    }
}
```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

298

298

```

boolean verify(byte[] buffer, short ofs, byte len)
{
    // Initializes the step counter
    short stepCounter = INITIAL_COUNTER ;

    // First checks the integrity of the variable
    byte tl = triesLeft ;
    stepCounter++ ;
    if (tl != (short)(~triesLeftBackup)) takeCountermeasure() ;
    stepCounter++ ;
    // No comparison if PIN is blocked
    if (tl < 0) return false ;
    stepCounter++ ;
    // First decrements the number of remaining tries
    JCSystem.beginTransaction() ;
    triesLeft = --tl ;
    stepCounter++ ;
    triesLeftBackup++ ;
    JCSystem.commitTransaction() ;
    stepCounter++ ;
    // Verifies the new value
    if (triesLeft != (short)(~triesLeftBackup)) takeCountermeasure() ;
}

```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

299

299

```

    stepCounter++ ;
    short equal = BOOL_TRUE ; // Main comparison
    stepCounter++ ;
    for(short i=0;i<len;i++)
        equal = (short)(equal &
            ((buffer[ofs+i]!=pin[i])?BOOL_FALSE:BOOL_TRUE));
    stepCounter++ ;
    if (equal == BOOL_TRUE) { // Comparison is successful
        //Reset the tries to the max
        stepCounter++ ;
        JCSystem.beginTransaction() ;
        triesLeft = maxTries ;
        triesLeftBackup = (byte)(~maxTries);
        JCSystem.commitTransaction() ;
        stepCounter++ ;
        if (triesLeft!=(short)(~triesLeftBackup))
            takeCountermeasure() ;
        stepCounter++ ;
        authenticated[0] = true ;
        if (stepCounter == (short)(INITIAL_VALUE+11) )
            return true ;
    } else { // Comparison failed
        stepCounter++ ;
        authenticated[0] = false ;
        if (stepCounter == (short)(INITIAL_VALUE+9))
            return false ;
    }
    takeCountermeasure() ; // Should have returned at this point
}

```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

300

300

Other well known rules ...

```
if (pin.isValidated() == true)
{
    // sensitive operations
}
else
{
    ISOException.throwIt((short) (0x6300+pin.getTriesRemaining()));
}
```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

301

301

Other well known rules ...

```
if (pin.isValidated() == true)
{
    if (pin.isValidated() == false)
    {
        ISOException.throwIt((short) (0x6300+pin.getTriesRemaining()));
    }
    else
    {
        // sensitive operations
    }
}
else
{
    ISOException.throwIt((short) (0x6300+pin.getTriesRemaining()));
}
```

- Références :

<http://javacard.vetilles.com/2008/05/15/jc101-12c-defending-against-attacks/>

Ronald De Keulenaer, Jonas Maebe, Koen De Bosschere, Bjorn De Sutter. Link-time smart card code hardening. International Journal of Information Security, 2015

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

302

302

La consommation de courant

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

303

303

La consommation de courant

- Elle est surtout utilisée dans le domaine de la cryptographie.
- Il existe différentes attaques :
 - la SPA (Simple Power Analysis)
 - la DPA (Differential Power Analysis)
 - la HODPA (High Order Differential Power Analysis)

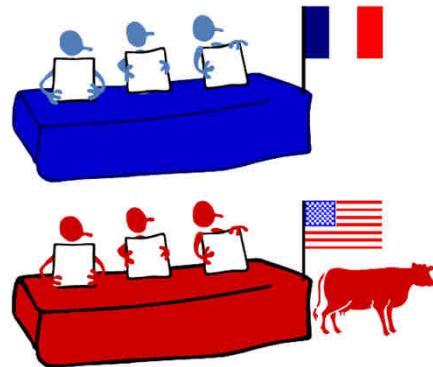
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

304

304

IMPORTATION DU BOEUF ?

- Seattle, 1999.
- Les représentants Français et Américains négocient sous quelles conditions le bœuf pourrait être importé en France.



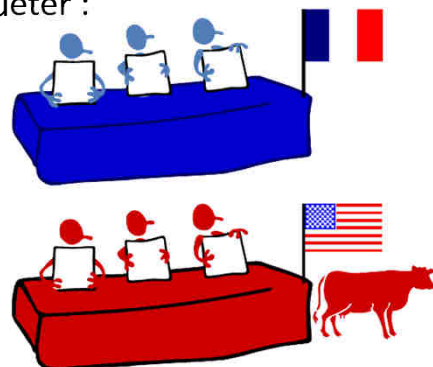
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

305

305

IMPORTATION DU BOEUF ?

- Seattle, 1999.
- Les représentants Français et Américains négocient sous quelles conditions le bœuf pourrait être importé en France.
- «The Sun » envoie un journaliste pour enquêter :



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

306

306

IMPORTATION DU BOEUF ?

- Mais il y a un problème technique : les négociations se déroulent dans un hôtel aux vitres opaques



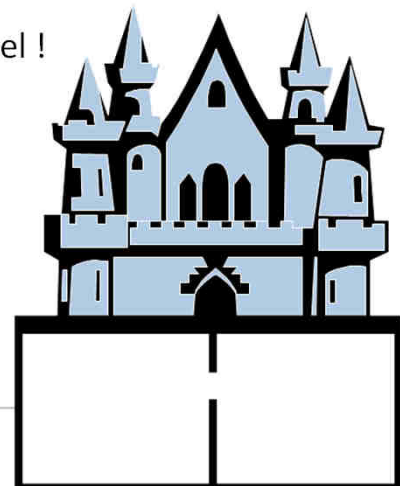
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

307

307

Attaque sur la consommation

- Idée : Regarder le compteur électrique de l'hôtel !



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

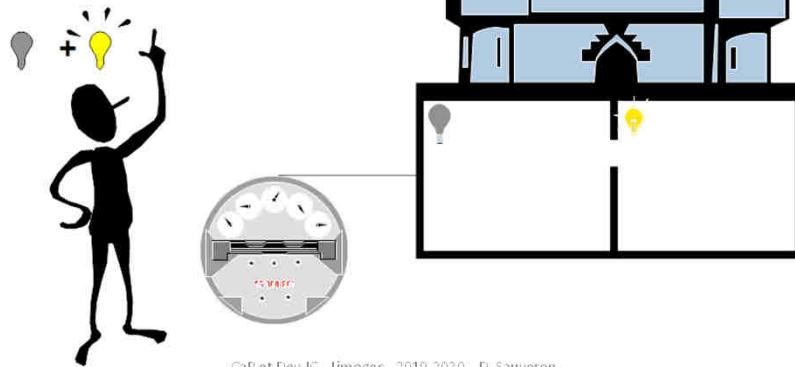
308

308

Attaque sur la consommation

- Le disque tourne lentement:

DEAL CONCLUDED



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

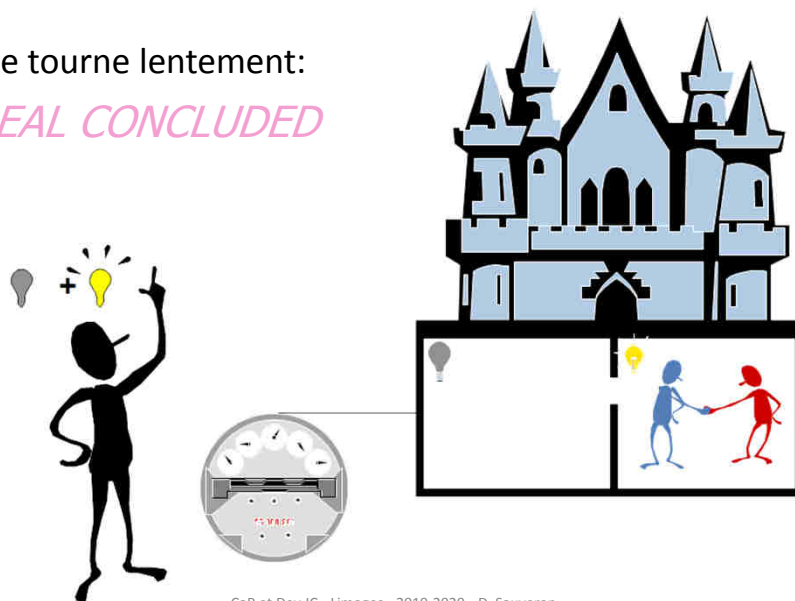
309

309

Attaque sur la consommation

- Le disque tourne lentement:

DEAL CONCLUDED



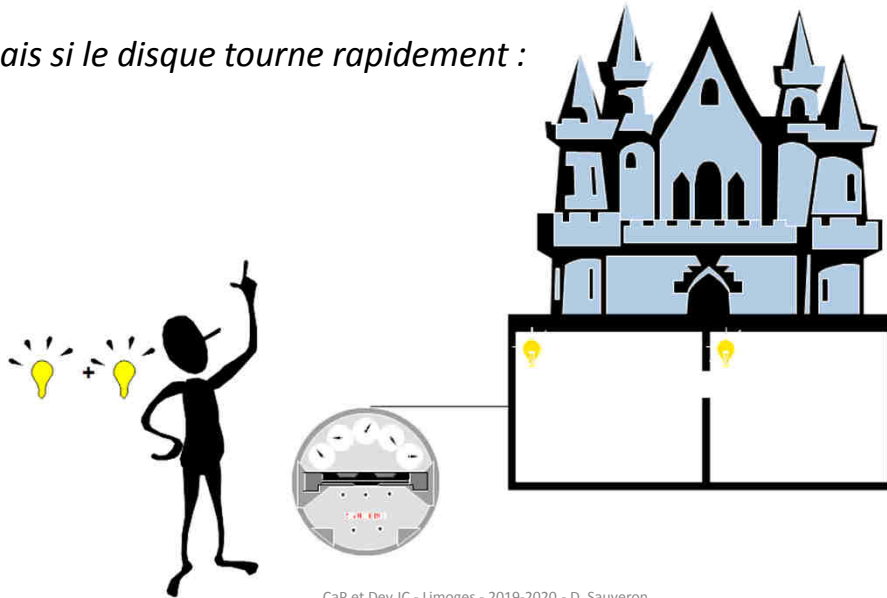
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

310

310

Attaque sur la consommation

- *Mais si le disque tourne rapidement :*



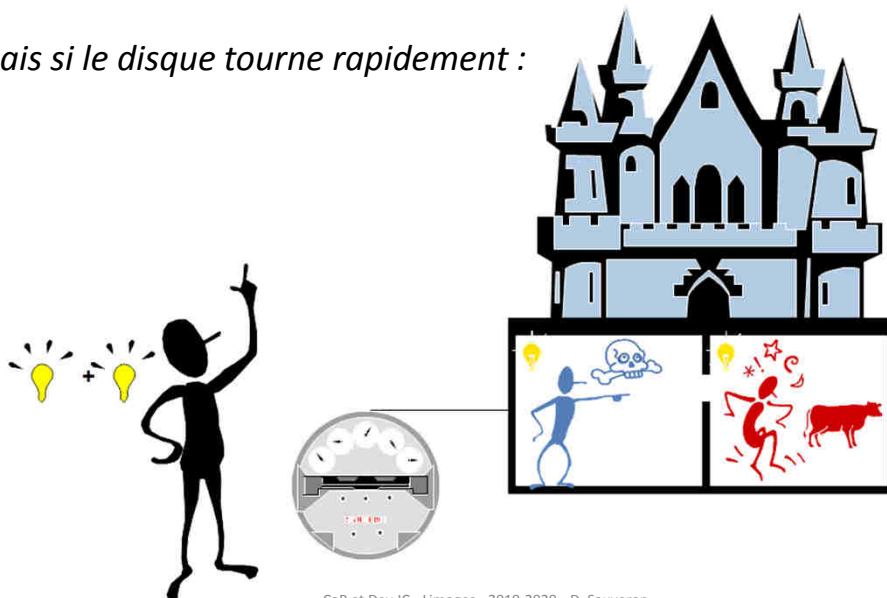
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

311

311

Attaque sur la consommation

- *Mais si le disque tourne rapidement :*



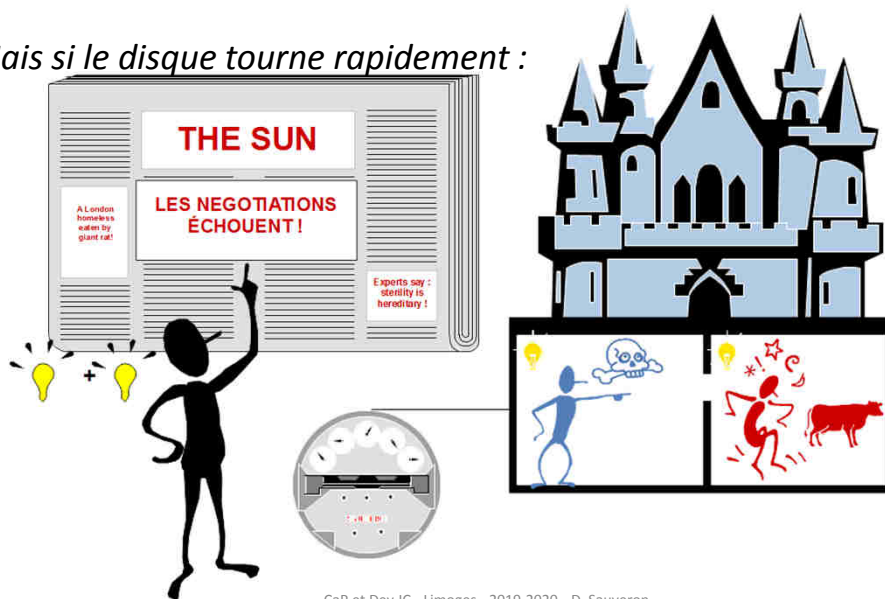
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

312

312

Attaque sur la consommation

- *Mais si le disque tourne rapidement :*



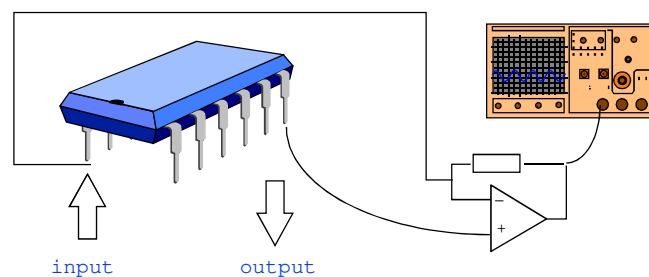
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

313

313

Conclusion

- The card's current consumption **may reveal secret information.**



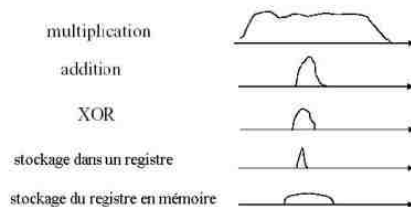
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

314

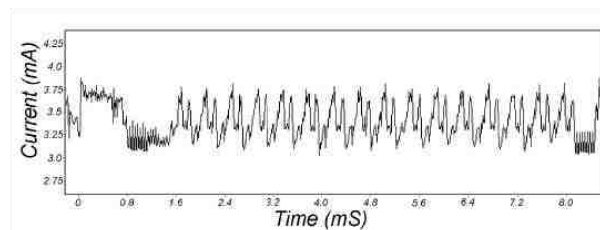
314

La SPA (Simple Power Analysis)

- Principe : Des instructions différentes ont une trace différente.



Consommation en courant d'un DES. On peut voir la permutation initiale, suivie des 16 tours.



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

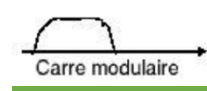
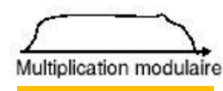
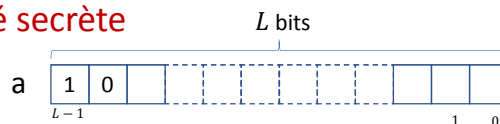
315

315

SPA sur la signature RSA (théorie)

- Signature RSA : $y^a \bmod n$
- y est le message à signer,
- n est public,
- a , l'exposant peut être considéré comme la clé secrète

```
s = 1 ;
for ( i = L - 1 ; i >= 0 ; i-- )
{
    s = s*s mod n ;
    if ( a [ i ] == 1 )
        s = s*y mod n ;
}
```

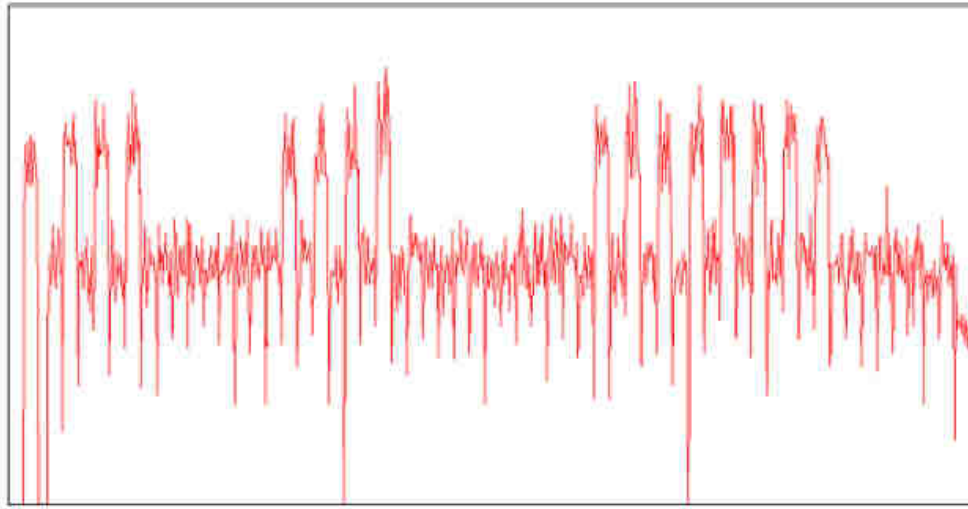


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

316

316

Un exemple réel

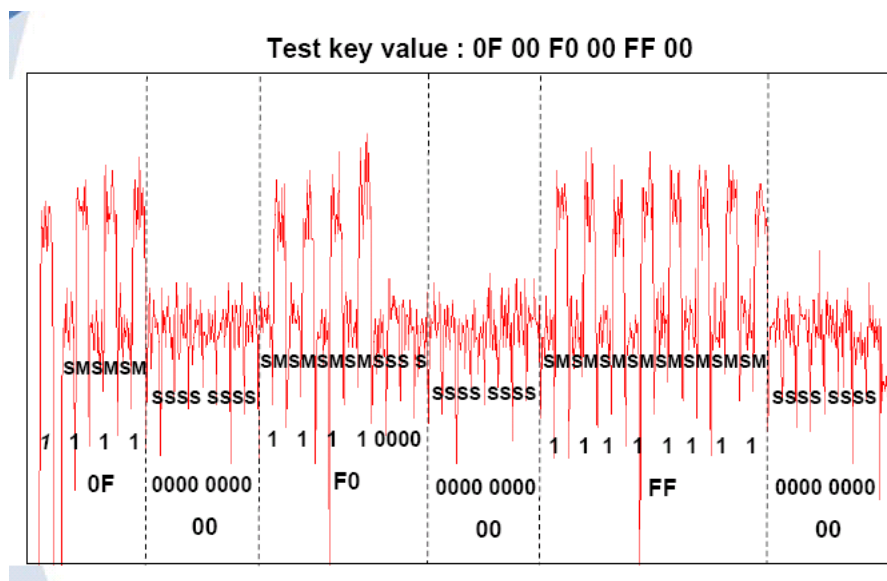


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

317

317

Un exemple réel

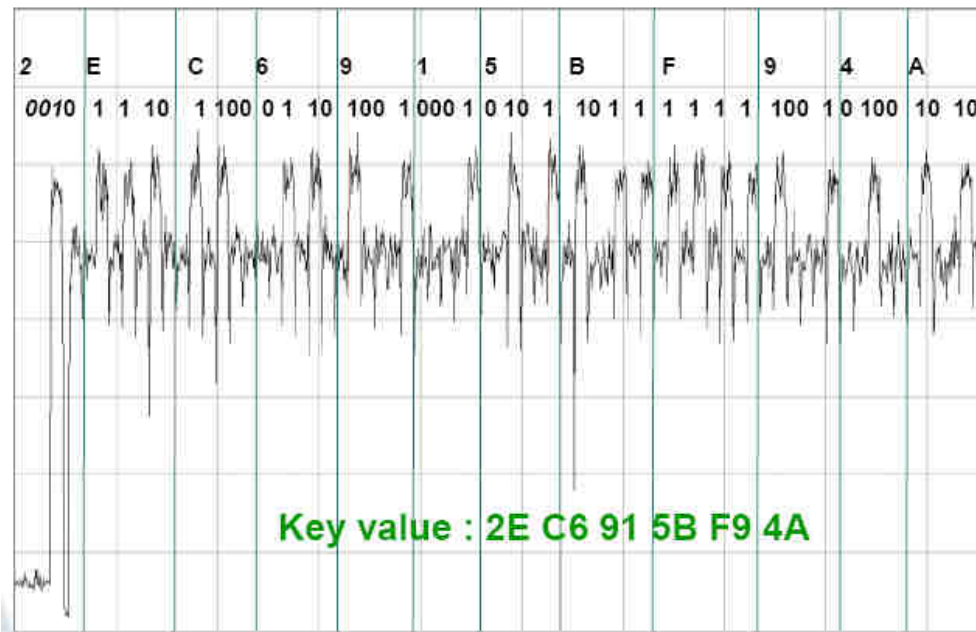


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

318

318

Un exemple réel



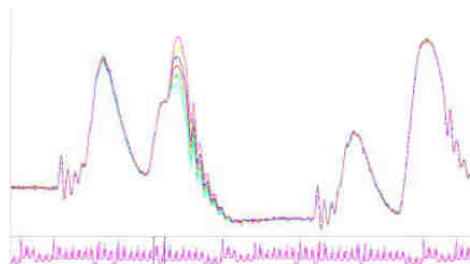
319

319

La DPA (Differential Power Analysis)

- Principe : La consommation dépend des opérations effectuées (les instructions) mais aussi des opérandes.
- Elle utilise des fonctions statistiques adaptées à l'algorithme visé qui font ressortir des corrélations entre un bit intermédiaire a (ne dépendant que d'un fragment Kr de r bits de la clé et du message d'entrée M) et la consommation de courant.

- Based on SPA
Adding the power of statistics to separate signal from noise



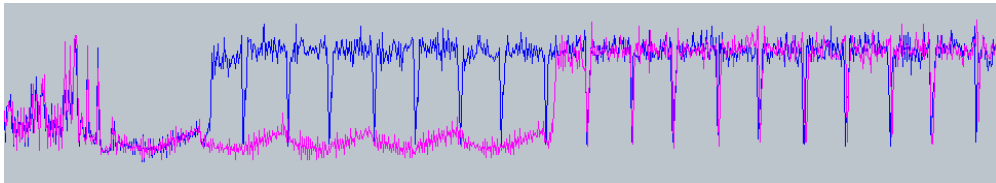
320

320

Concrete attack on concrete keys

- Comparing:

- decryption with key = 0000 1111 ...
- decryption with key = 1111 1111 ...



token designed and manufactured in 1998...

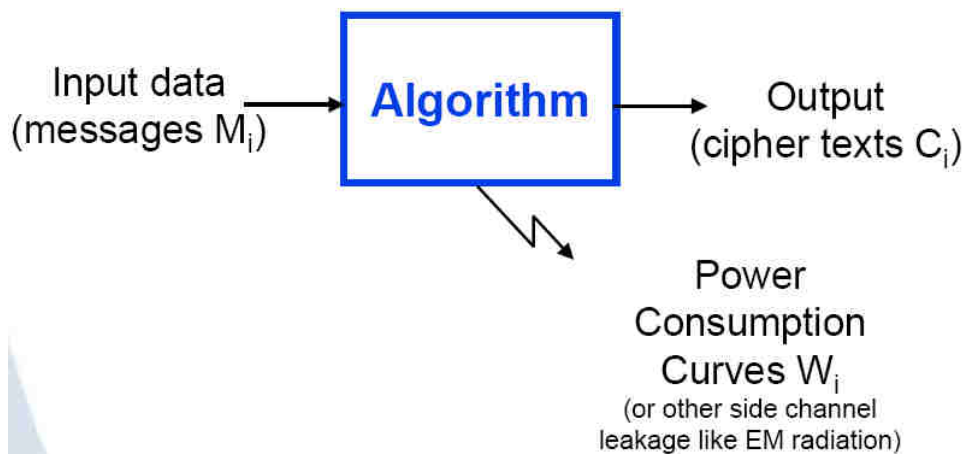
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

321

321

DPA Hypothesis

Play the algorithm N times
($100 < N < 100000$)



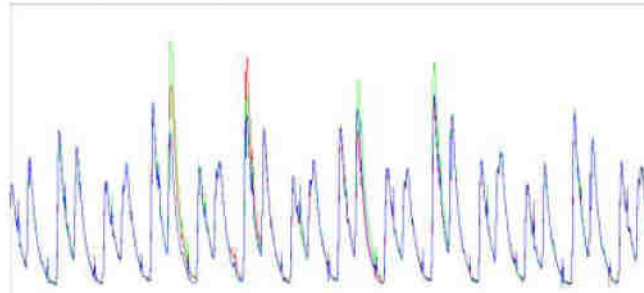
322

322

Acquisition procedure

- After data collection, what is available ?
 - N plain and/or cipher random texts

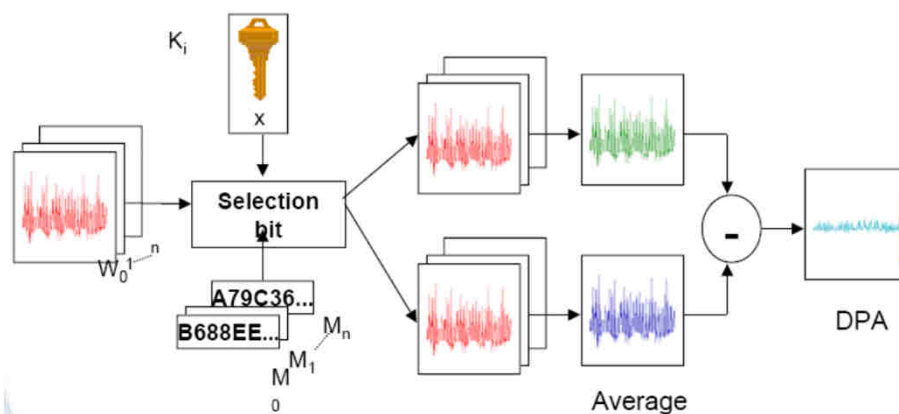
00	B688EE57BB63E03E
01	185D04D77509F36F
02	C031A0392DC881E6 ...
 - N corresponding power consumption waveforms



323

323

Hypothèse et test

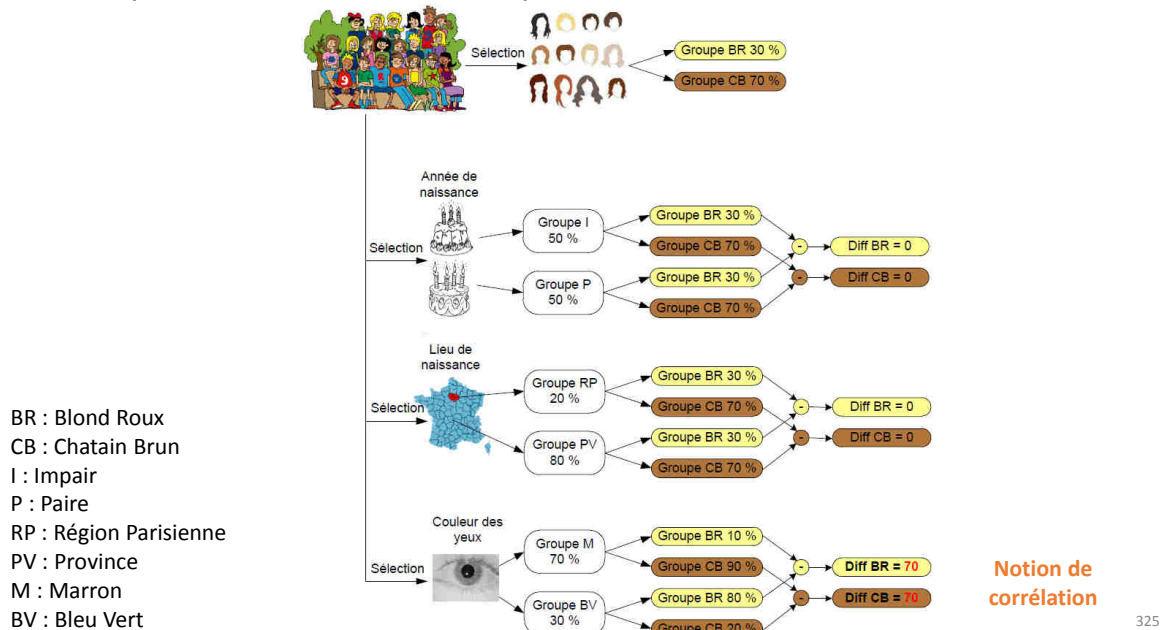


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

324

324

Explication didactique



325

La HODPA (High Order DPA)

- La HODPA ou DPA d'ordre n est aussi basée sur une étude statistique de la consommation de courant de la carte.
- Différence : elle utilise des corrélations entre la consommation de courant et n variables intermédiaires ne dépendant que d'un fragment de la clé et du message d'entrée.
- Elle est beaucoup plus difficile à réaliser, mais beaucoup plus puissante.

326

Exercice

- Une pièce sans fenêtre dispose d'une lumière (une ampoule).
- Trois interrupteurs () sont placés à l'extérieur de cette pièce.
- Sachant que les trois interrupteurs sont en position OFF et **qu'on a le droit d'entrer** dans la pièce **qu'une seule fois**, comment savoir quel est l'interrupteur qui allume cette lumière ?
- *Attention : On ne peut pas laisser la porte ouverte, on ne peut pas se faire aider de quelqu'un et la porte ne contient pas de vitre et ne comporte pas de trou de serrure. Il n'y a pas de jour autour de la porte. Les câbles partant des interrupteurs sont blindés et ne sont pas accessibles physiquement. En gros la pièce ne laisse fuir aucune information d'aucun nature que ce soit vers l'extérieur.*

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

327

327

COUNTER-COUNTER MEASURE

- Against adding noise to the power consumption signal.
- Capture electromagnetic radiation at various chip locations!
- Equipment:



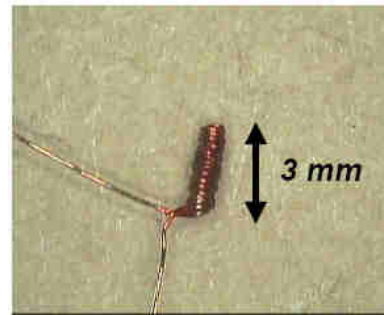
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

328

328

Les émissions EM

- **Principe** : Les courants qui circulent dans la puce induisent des champs électromagnétiques qui sont susceptibles de donner le même type d'information que le courant.
- **Différence** : L'information est plus locale. On peut déplacer la micro-sonde électromagnétique au dessus de la zone qui nous donnera le plus d'informations (exemple : co-processeur cryptographique).
- **Avantage** : Insensible aux contre-mesures physiques tels que l'ajout de bruit en sortie ou le lissage de la consommation globale de courant.
- **Inconvénient** : La reproductibilité des mesures est difficile.



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

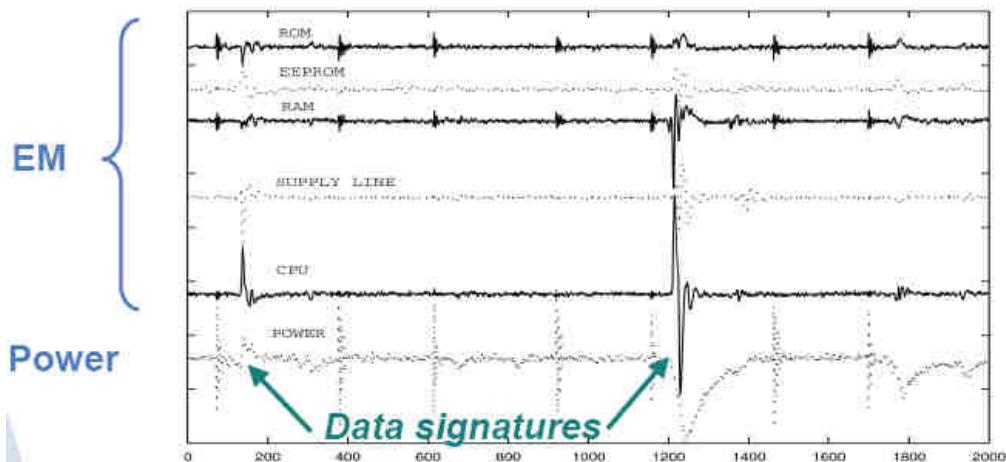
329

329

Différence EM/Power

- **EM signals versus XY probe position**

Differential traces between (00h ⊕ 00h) and (FFh ⊕ 00h) picked up at different locations



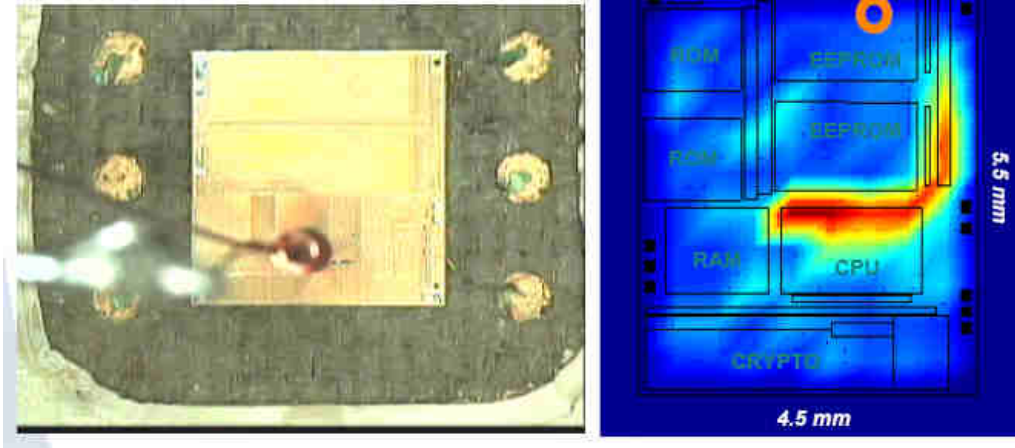
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

330

330

Le dispositif de cartographie EM

- Horizontal cartography (XY plane)
 - to pinpoint instruction related areas
 - better if automated

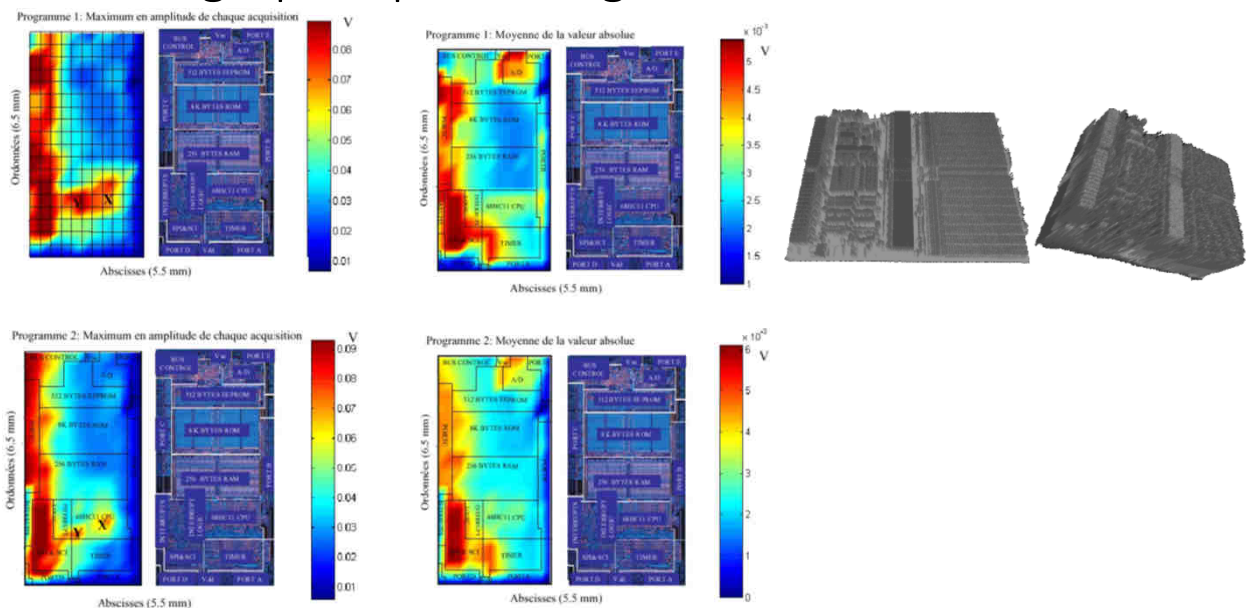


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

331

331

Cartographie pour 2 algo / Vue 3D



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

332

332

Attaques EM

- la SEMA (Simple EM Analysis)
- la DEMA (Differential EM Analysis)
 - => nécessite moins d'acquisitions

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

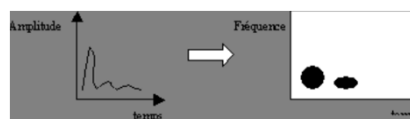
333

333

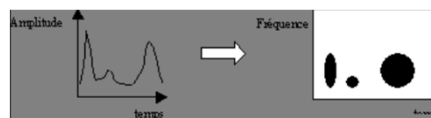
La rétro-conception logicielle

- méthode du « dictionnaire » :

- Une instruction :



- Une autre instruction :



- La séquence des deux :



- Utilisation possible de la consommation en courant ou des émissions EM.
- Statut : recherches en cours mais de beaux résultats à WISTP 2007,

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

334

334

Quelques attaques invasives

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

335

335

Des attaques invasives

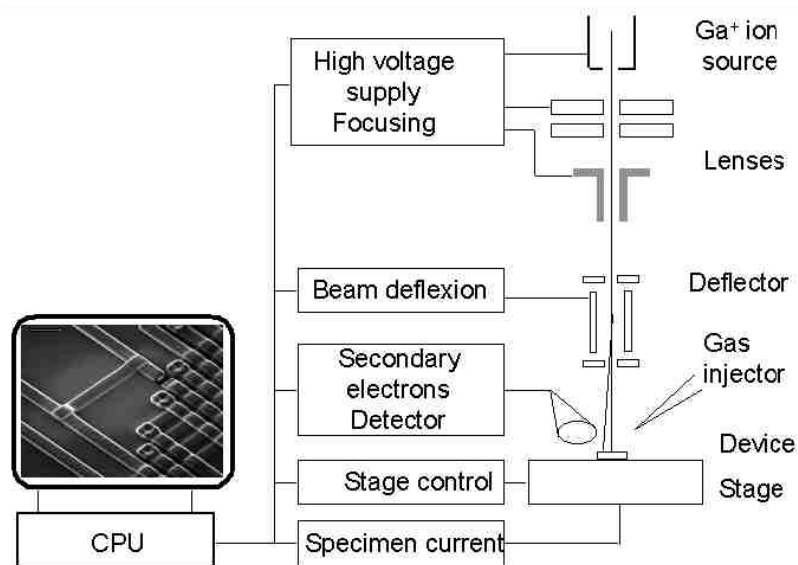
- micro-probing
- modification de circuit (FIB)
- rétro-conception du circuit

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

336

336

Le FIB (Focused Ion Beam)

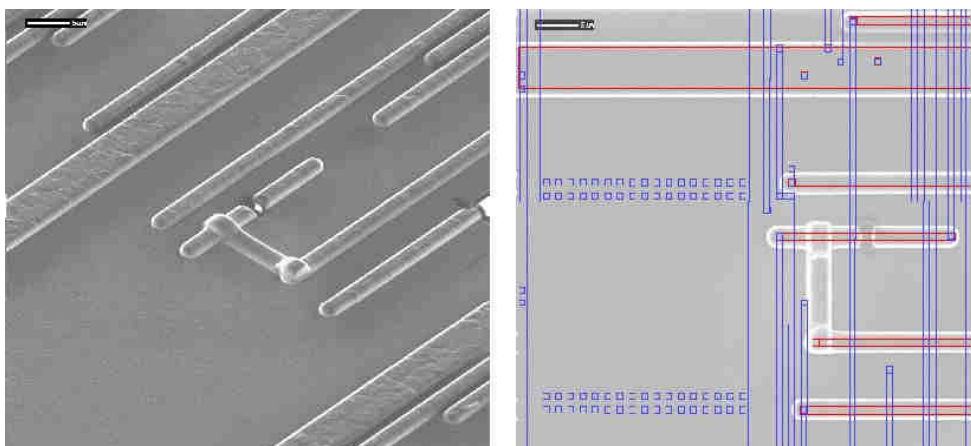


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

337

337

Modifications de circuits (1/2)

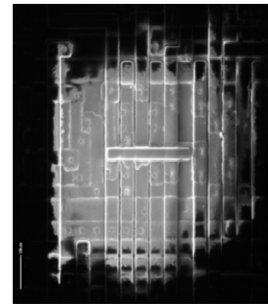
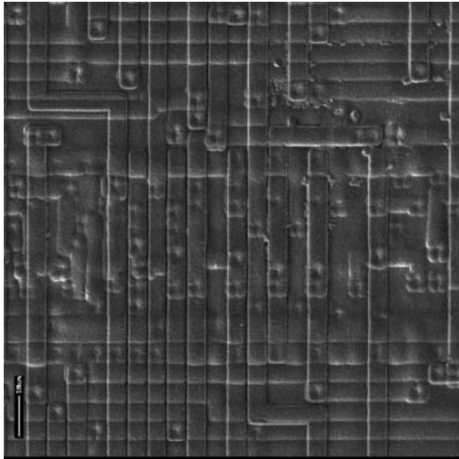


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

338

338

Chip Re-Wiring / Addition of a Track



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

339

339

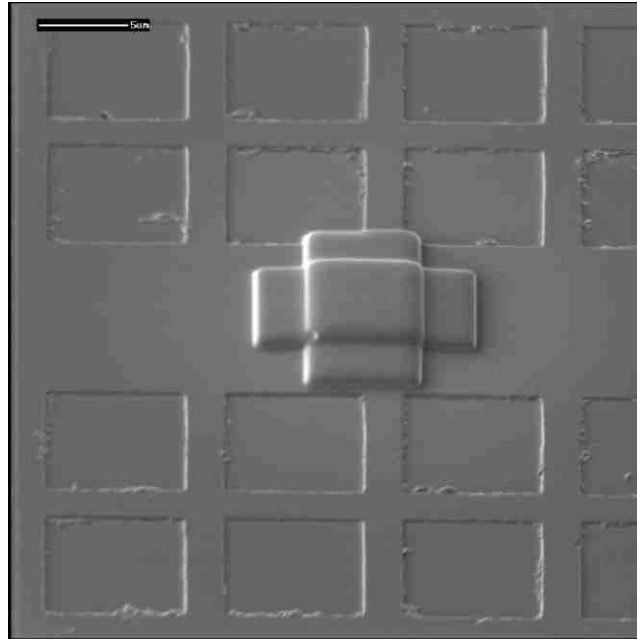
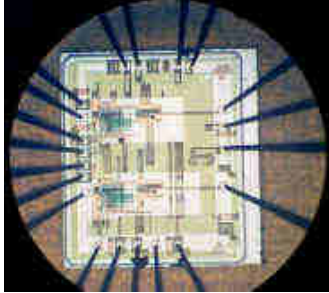
Modifications de circuits (2/2)



340

340

Micro-probing



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

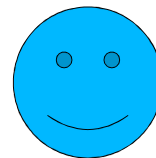
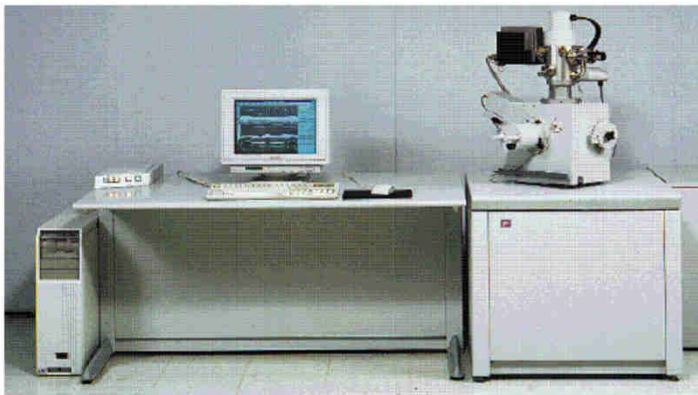
341

341

Coût

Très cher !

- If you have more money or if you are a student.



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

342

342

Cotation d'une attaque

- Niveau d'expertise requise
 - Expert, ..., homme de la rue
- Connaissance du produit
 - Information publique, diffusion restreinte, sensible, critique
- Accès au produit
 - Nombre d'échantillon nécessaire : <10 , <100, <1000, non praticable
- Temps d'identification
 - <1 heure, 1 jour, 1 semaine, 1 mois, ...
- Temps d'exploitation
 - <1 heure, 1 jour, 1 semaine, 1 mois, ...
- Équipement
 - Aucun, standard, spécialisé, ...
- ...

=> La sécurité et le business s'opposent. Il faut donc trouver un juste compromis !

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

343

343

Le piratage facile en vidéo



How to Reverse-Engineer a Satellite TV Smart Card.mp4

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

344

344

Mifare



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

345

345

JSR-268 Java Smart Card I/O API

- APIs for APDU-based communication with Smart Cards
 - in Java Platform 6.0
- package `javax.smartcardio`
 - `TerminalFactory`, `TerminalFactorySpi`, `CardTerminal`
 - `Card`, `CardChannel`
 - `CardException`, `CardPermission`
 - `ATR`, `CommandAPDU`, `ResponseAPDU`
- Comments
 - JC-RMI proxy generators based on it

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

346

346

Example

```
// show the list of available terminals
TerminalFactory factory = TerminalFactory.getDefault();
List<CardTerminal> terminals = factory.terminals().list();
System.out.println("Terminals: " + terminals);

// get the first terminal
CardTerminal terminal = terminals.get(0);

// establish a connection with the card
Card card = terminal.connect("T=0");
System.out.println("card: " + card);

CardChannel channel = card.getBasicChannel();
ResponseAPDU r = channel.transmit(new CommandAPDU(c1));
System.out.println("response: " + toString(r.getBytes()));

// disconnect
card.disconnect(false);
```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

347

347



Java Card

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

348

348

Qu'est ce que Java Card ?

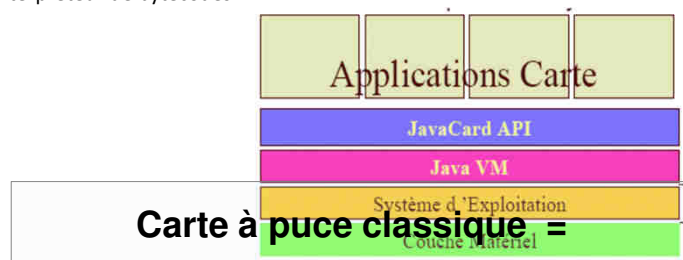
- Technologie permettant de faire fonctionner des applications écrites en langage Java pour :
 - carte à puce



- autres périphériques à mémoire limitée (par exemple iButton)



- De manière intuitive :
 - Il s'agit d'une carte basée sur un interpréteur de bytecodes.



- La technologie Java Card définit une plate-forme pour cartes à puce sécurisée, portable et multi-applicative qui incorpore beaucoup des avantages du langage Java.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

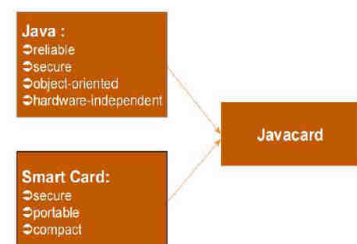
349

349

Les avantages de la technologie Java Card

- La facilité de développement des applications grâce :
 - à la programmation orientée objet offerte par Java
 - à l'utilisation des environnements de développement existants pour Java.
 - à une plate-forme ouverte qui définit des APIs et un environnement d'exécution standard.
 - à l'encapsulation de la complexité sous-jacente du système des cartes à puce.
- L'indépendance au hardware réalisée grâce au langage Java
 - => « Write Once, Run Anywhere »
- La gestion dynamique de multiples applications
 - => possibilité de mises à jour des applications de la Java Card sans avoir besoin de changer de cartes
 - => possibilité d'offrir des nouveaux services
- La compatibilité avec les standards existants pour les cartes à puce
 - ISO7816 :
 - Protocole de communication
 - Protocole de haut niveau pour la gestion de fichiers, etc
- On verra les inconvénients plus tard.

Java Card = Java + smart Card



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

350

La plate forme Java

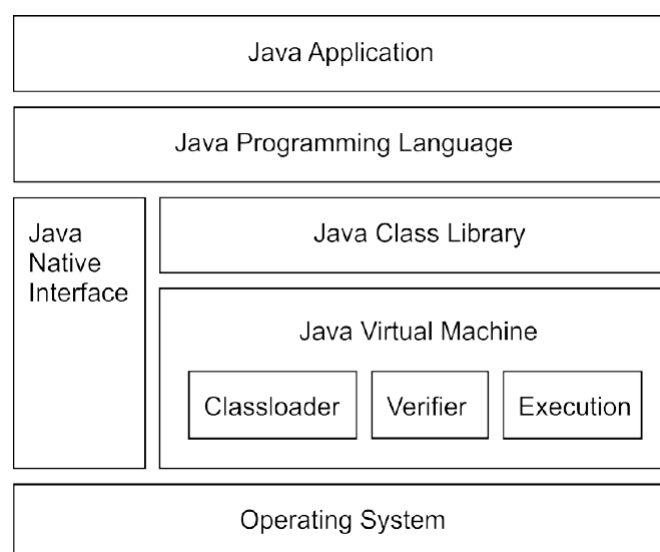
- Java est un langage de programmation
 - Voir le « white paper » de J. Gosling
 - Un programme Java est compilé et interprété
- Java est une plate-forme
 - La plate-forme Java, uniquement software, est exécutée sur la plate-forme du système d'exploitation,
 - La « Java Platform » est constituée de :
 - La « Java Virtual Machine » (JVM)
 - Des interfaces de programmation d'application (Java API),
 - Une interface pour les méthodes natives (JNI)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

351

351

Architecture de la plate-forme

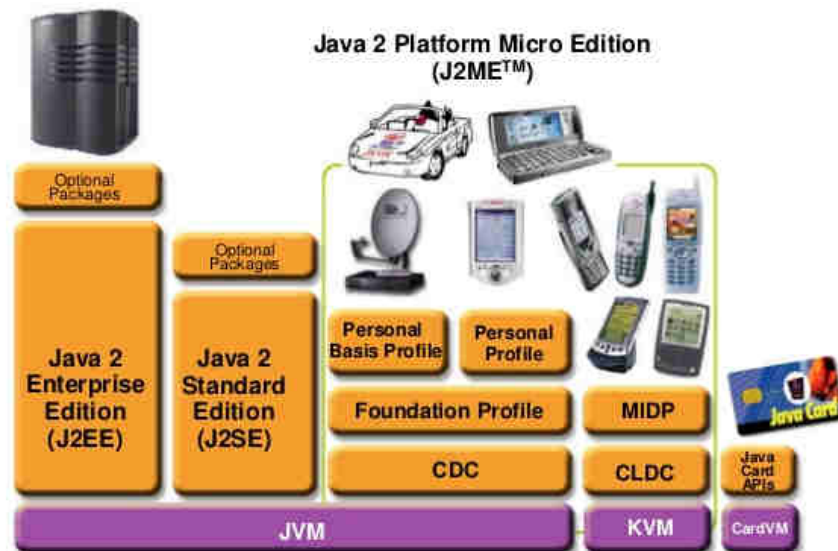


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

352

352

Les différentes plates-formes



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

353

353

Les composants

- Le langage de programmation,
- La librairie,
- La machine virtuelle
 - Un jeu d'instruction interprétable (*byte codes*)
 - Un format de chargement, pas forcément équivalent au format d'exécution le *class file*
 - Un algorithme de vérification des règles de typage Java ramenées au byte code, obligatoire pour les applets,
 - Un compilateur à la volée.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

354

354

Java langage de programmation

- Java est un langage de programmation particulier qui possède des caractéristiques avantageuses :
 - Simplicité et productivité :
 - Intégration complète de l'OO
 - Gestion mémoire (« Garbage collector »)
 - Robustesse, fiabilité et sécurité (langage fortement typé)
 - Indépendance par rapport aux plates-formes
 - Distribution et aspects dynamiques
 - Performance

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

355

355

Machine virtuelle Java

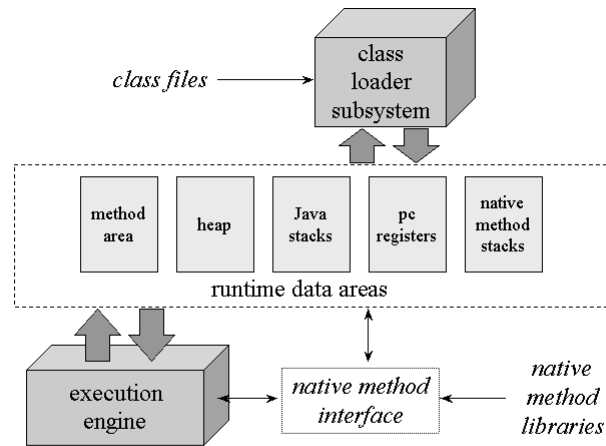
- La JVM définit:
 - Les instructions du CPU
 - Les différents registres (PC, SP,...)
 - Le format des fichiers .class
 - La pile Java et la pile C, en RAM,
 - Le tas (Heap) des objets « garbage-collectés » en RAM aussi,
 - L'espace mémoire partagé :
 - Stockage des classes et des données statiques,
 - Stockage des objets (les instances) dans le tas.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

356

356

Machine virtuelle Java



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

357

357

Machine virtuelle Java

- Trois tâches principales:
 - Charger le code (class loader)
 - Vérifier le code (byte code verifier)
 - Exécuter le code (runtime interpreter)
- D'autres threads s'exécutent:
 - Garbage collector
 - (JIT compiler)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

358

358

Sécurité par le typage

- En Java les pointeurs existent mais:
 - la manipulation explicite d'un pointeur en tant que tel est impossible en Java **le compilateur vérifie** cette règle ainsi que la VM. Ainsi le code suivant n'est pas compilable:


```
...
p=new Object();
p++;
```
 - tous les objets Java sont en fait des pointeurs (que l'on nomme références).
- Les variables locales (variables de types primitifs et références) sont allouées sur la pile (ou dans les registres) et les objets dans le tas.
- On ne contrôle pas la zone d'allocation ni quand sera effectuée la désallocation.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

359

359

Sécurité par le typage

- Lors du chargement **la VM vérifie** si, au niveau byte code, il n'y a pas de transtypage ne correspondant pas aux règles de Java : **c'est le rôle du vérifieur de type**.
- Après, une référence n'a PLUS besoin de vérifier si le typage est correct.
- Donc la sécurité par le typage repose sur :
 - Un algorithme dans le compilateur,
 - Un autre dans la VM.

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

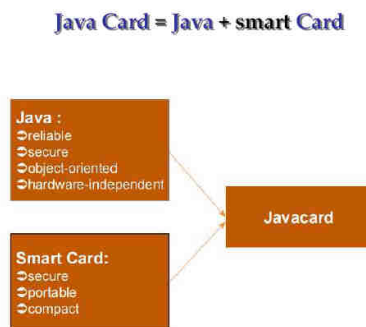
360

360

Java Card

• Attention

- Les informations concernent Java Card 3.0 Classic Edition (pas Connected Edition) et Java Card 2.X



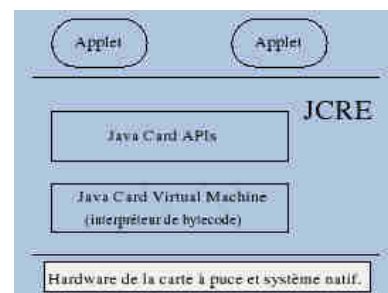
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

361

361

Présentation générale

- Un ensemble de spécifications
 - Publiées par Oracle
 - Basées sur la plateforme Java
 - <http://www.oracle.com/technetwork/java/embedded/javacard/index.html>
- Découpées en trois parties
 - Application programming interfaces (APIs)
 - Execution environment (JCRE)
 - Virtual machine (VM)



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

362

362

Agenda

- Introduction
- JCRE (Runtime Environment)
- JCVM (Virtual Machine)
- APIs

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

363

363

Un autre point de vue

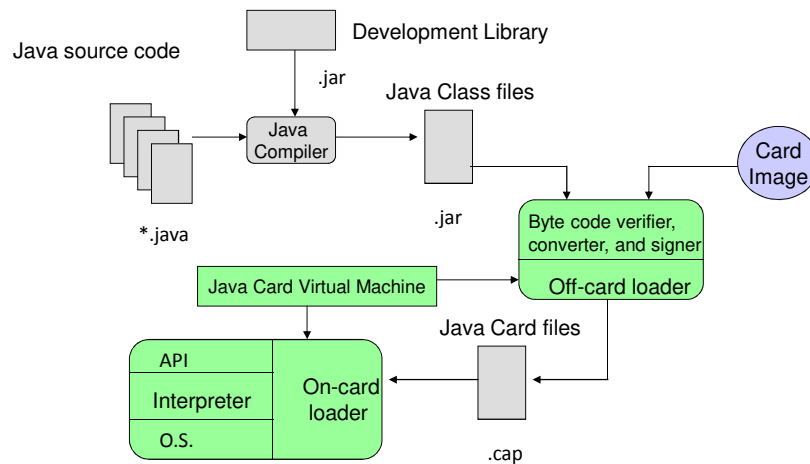
- Architecture
- Mémoire
- Les différentes versions

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

364

364

L'architecture Java Card



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

365

365

Architecture

- Converter (part of the JCDK):
 - Class file format takes too much space on Java Card
 - Produces a format that fits the Smart Card constraints,
- Tokenization of the format
 - Need a representation of the content for pre-linking
 - The converter uses all class files of a package and all export files of ALL imported packages,
 - Output an Export file and a Cap file
- Conversion process
 - Verifies the Java Card language restriction
 - Optimize byte code
 - Invokes the off card verifier

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

366

366

Two specific file formats

- The CAP (Converted Applet) file format
 - Contains all the classes from one package
 - Semantically, is equivalent to a set of class (.class) files
 - Syntactically, differs a lot from class (.class) files
 - All “string names” are replaced by “token identifiers”
- The EXP (Export) file format
 - Maintains the consistency between the originated class (.class) files and the resulting CAP file
 - Only for public (exported) data
 - Contains API information for a package of classes (access scope, class name, method signature, ...)
 - Can be freely distributed, used during pre-linking phase
 - Not loaded into the card

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

367

367

The CAP file

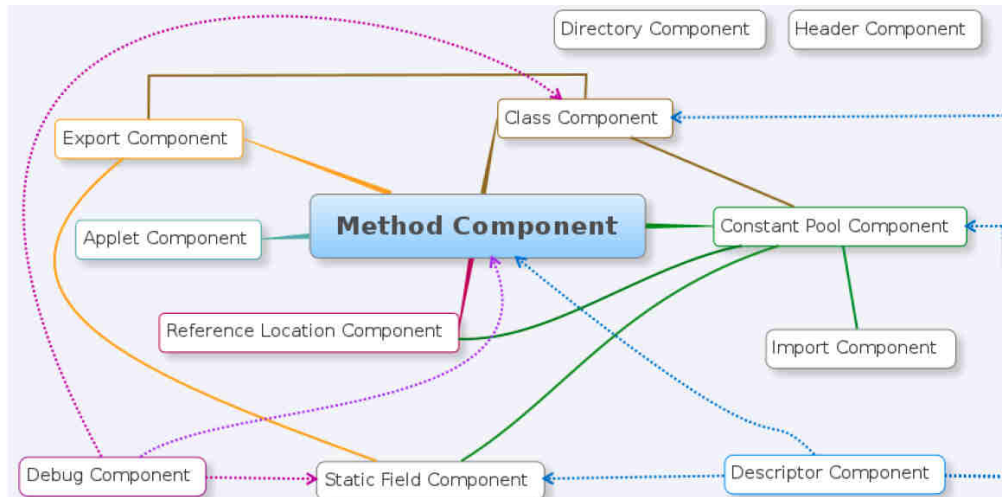
- Contains an executable representation of package classes
- Contains a set of components (11)
- Each component describes an aspect of CAP file
 - Class info
 - Executable byte code
 - Linking info,...
- Optimized for small footprint by compact data structure
- Loaded on card

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

368

368

Interdependences

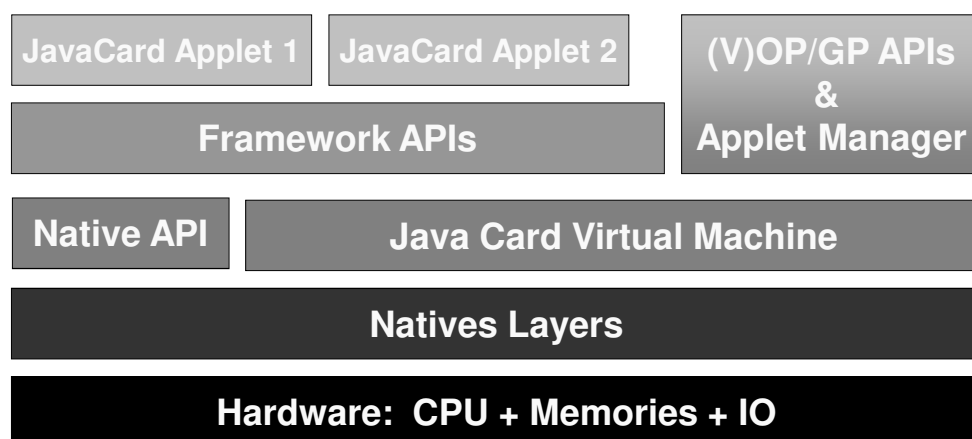


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

369

369

Java Card architecture



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

370

370

Java Card memory model

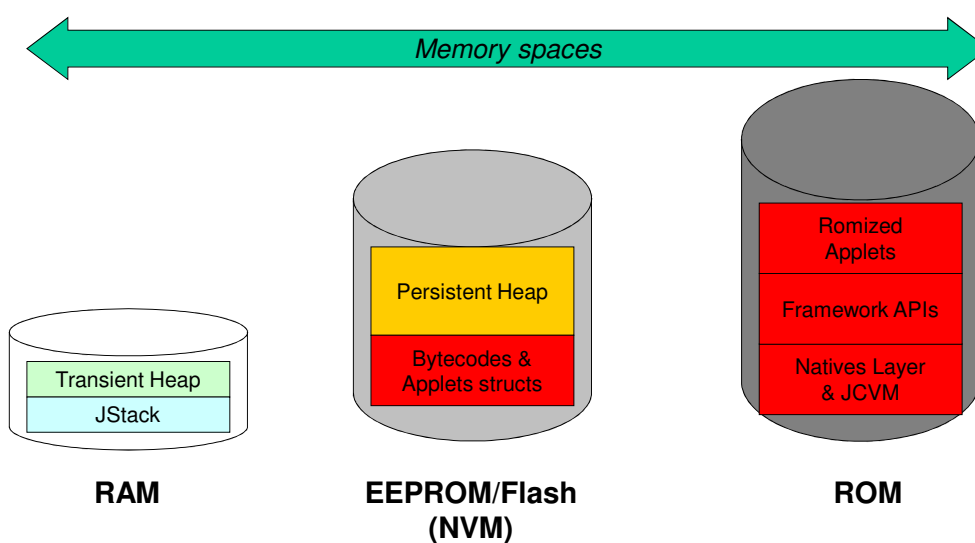
- By default, all objects are *implicitly* persistent
 - Because we have few RAM
 - Objects must survive between two sessions
- Some arrays can be transient
 - For efficiency and security reasons
- Transactional mechanisms are provided
 - All write operations on persistent memory are atomic
 - At the programming level a mechanism to handle transactions is also available

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

371

371

Java Card memory



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

372

372

Java Card 2.1

- File format of applet:
 - Standardized
 - file .cap identical to .class excepted:
 - One .cap file per Java package
- « Firewall » between applets
 - The virtual machine must ensure that the code doesn't run out of its execution space (context)
 - Means to switch the execution context
 - Entry point object and global array can be accessed by applets (e.g., APDU)
 - JCRE can access each object
 - Interaction between applet through a shareable interface
 - System.share (Object ...) method suppressed

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

373

373

Java Card 2.2

- Load process of the .cap file standardized
 - Interoperability of JC 2.1 stops at the smart card loading,
- Object, applet and package deletion
- On card verifier (optional),
- Logical channels,
- Optional Garbage collector on demand,
- Support elliptic curves and AES algorithm,
- JC-RMI
 - Skeleton/stub generator 'a la RMI' hide the APDU encoding-decoding
 - Rapid development and integration of SC applets

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

374

374

Agenda

- Introduction
- JCRE (Runtime Environment)
- JCVM (Virtual Machine)
- APIs

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

375

375

Execution environment (JCRE)

- Define how a Java Card manages its resources
- Define constraints on the Java Card operating system
 - Applet lifetime (installation, register and deletion)
 - Logical channels and applet selection,
 - Transient objects,
 - Applet isolation (firewall) and sharing,
 - Transaction and atomicity,
- RMI
- The JCRE is at the heart of a Java Card

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

376

376

APDU commands

- 2 types of APDU can be sent to the card:

- **OS/Administrative** commands

- OS commands available in JCRE and CM
 - Select, Load, Install ...
 - Administrative commands specified by SC manufacturer
 - Get Info, ...



- **Applicative** commands

- specific to the JC applets loaded in the card
 - eg: debit, credit, getbalance for an e-purse applet
 - eg: create file, update file for the GSM applet

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

377

377

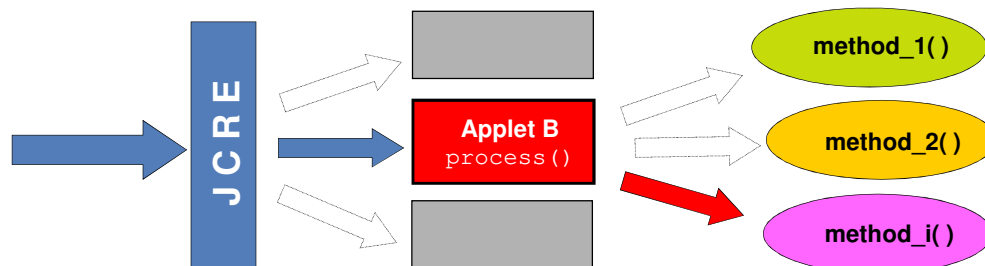
2 command dispatchers

- **JCRE's task:** main dispatcher

- Route the incoming commands to the JCVM and the selected applet

- **Applet designer's task:** second dispatcher

- Implement the applet's command dispatcher (extraction of the header information and call of the associated method)



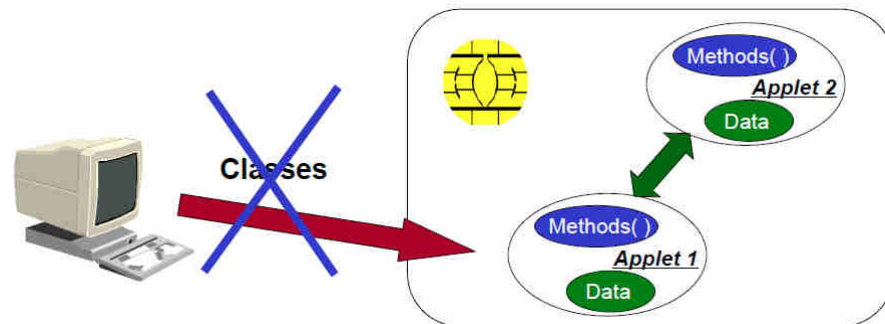
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

378

378

Unsupported features Dynamic class loading

- Classes are **statically linked** before being downloaded
- No way to download classes on the fly as needed...
- Applets only refer to classes which already exist on the card

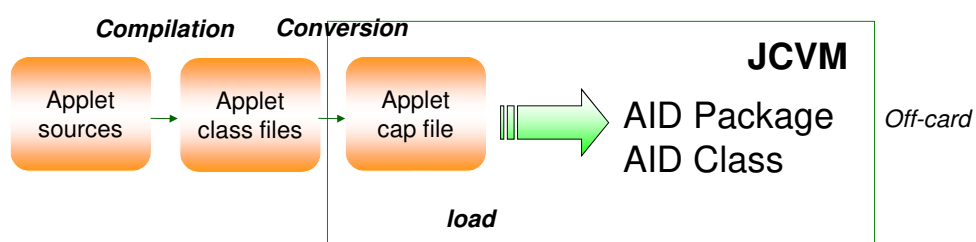


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

379

379

Java Card applet life cycle



Un AID

RID (5 octets)

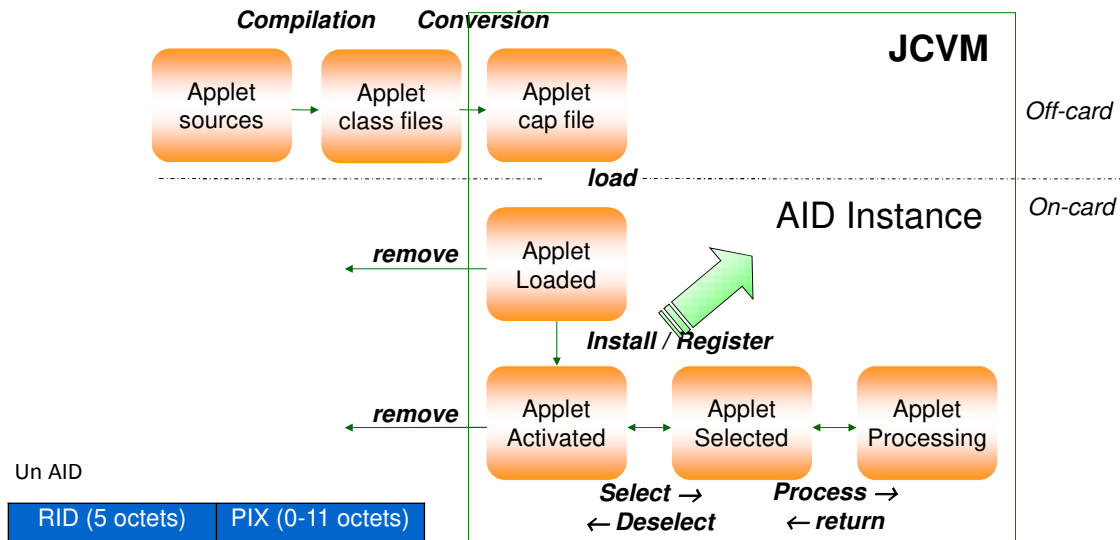
PIX (0-11 octets)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

380

380

Java Card applet life cycle



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

381

381

Methods of the applet

- An applet extends `javacard.framework.Applet`,
- It **MUST** implement the method `install`
- It **MUST** call the method `register`
- It **MUST** implement the method `process`
- It **SHOULD** implement `select`, `deselect`
- The `process` method can only be called if the applet has been selected

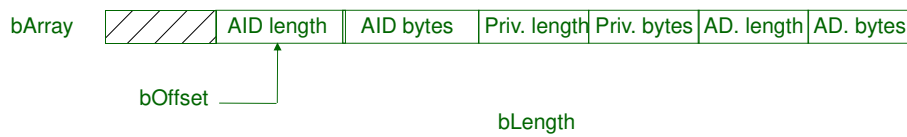
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

382

382

install method

- JCRE calls this static method first and gives
 - Applet instance AID
 - Applet privilege
 - Applet parameters
- `install` method creates an instance of an `Applet` subclass
 - Performs any necessary initializations,
 - Must call once one of the `register` methods
 - If no parameter is provided, only one installation



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

383

383

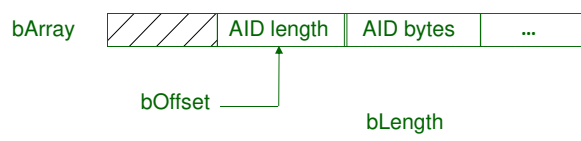
register method

- Used by the applet to register this applet instance
- Interacts with the Java Card Runtime Environment
 - `final void register()`
 - Assign the applet instance AID with default AID bytes defined in CAP file
 - `final void register(byte [] bArray, short bOffset, byte bLength)`
 - Assign applet instance AID with the specified AID bytes
- Warning: when receiving the byte array as parameters of the `install` command, the length is sent before the AID

```
public static void install(byte[] bArray, short bOffset, byte bLength) throws
IOException {
```

```
=>
```

```
register(bArray, (short)(bOffset + 1), (byte)bArray[bOffset]);
```



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

384

384

process method

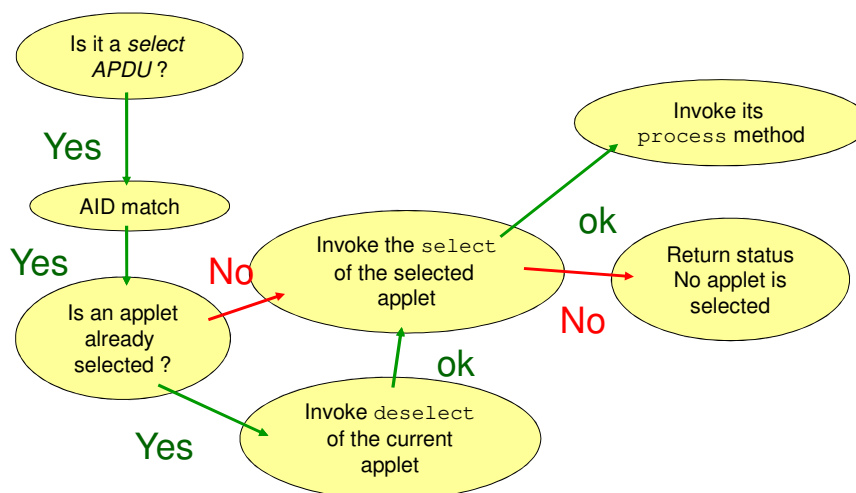
- Contains the core application code of the applet
- Handles all the incoming APDU messages for the applet selected
- Called by the JCRE
- Upon normal return from this method the Java Card runtime environment sends the ISO 7816-4 success status word 90 00
- If it throws an exception the JCRE sends the associated reason code as the response status

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

385

385

Selection protocol

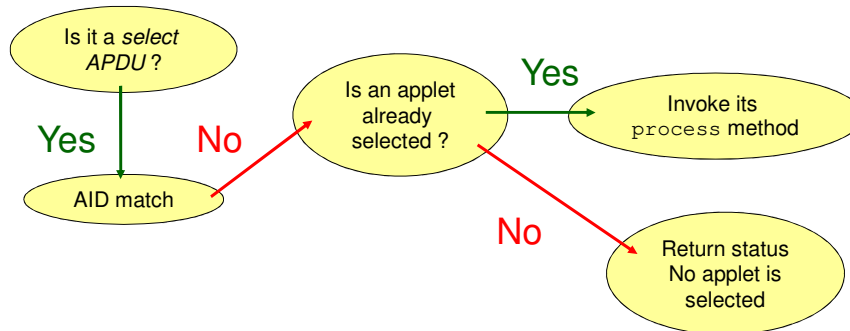


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

386

386

Selection protocol

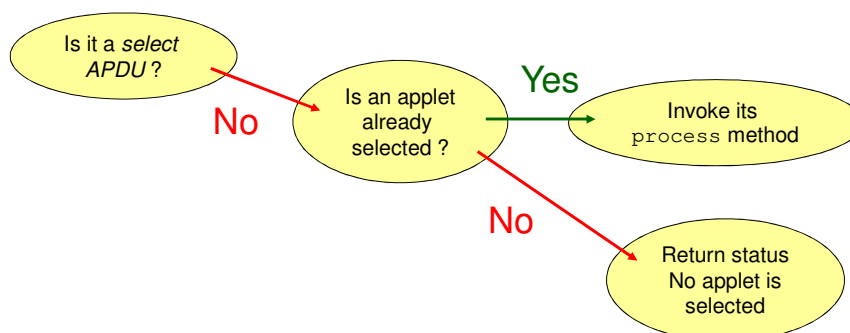


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

387

387

Selection protocol

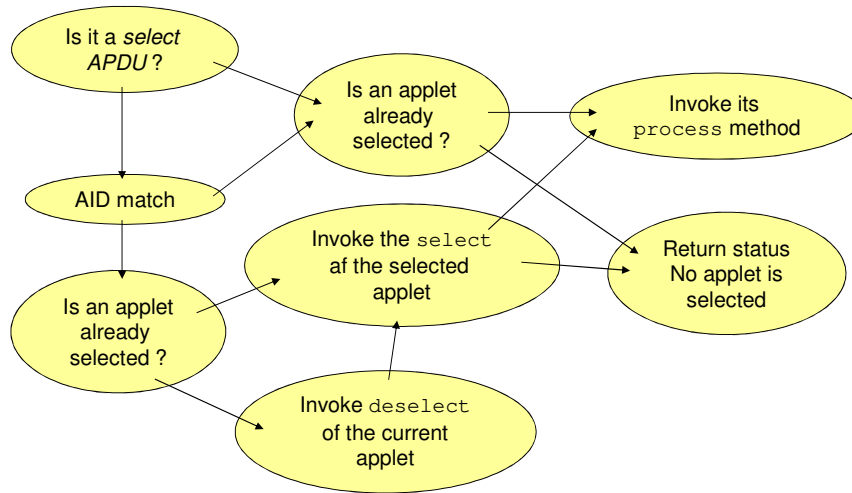


CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

388

388

Selection protocol



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

389

389

Agenda

- Introduction
- JCRE (Runtime Environment)
- **JCVM (Virtual Machine)**
- APIs

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

390

390

Virtual machine (VM)

- This specification is made of several parts
 - The definition of the Java language subset that is supported
 - The definition of 2 specific file formats
 - The definition of a specific instructions set

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

391

391

Java language subset

- | | |
|---|--|
| <ul style="list-style-type: none"> • Supported <ul style="list-style-type: none"> • boolean, byte, short • int (optional) • Objects • Arrays • Virtual methods • Dynamic allocation • Packages • Exceptions • Interfaces | <ul style="list-style-type: none"> • Not supported <ul style="list-style-type: none"> • float, double, long • char, String • Multi-dimensional arrays • Garbage collector • Finalization • Threads • Dynamic loading of classes • Security manager |
|---|--|

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

392

392

BC interpretation

- It is the execution engine for the byte code loaded into the card,
- It controls byte code execution, memory allocation and participate to the security through the firewall,
- Often it includes more tests than the firewall due to the absence of BC verifier...

Note that hardware processors executing Java Card bytecode exist (e.g. ST22).

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

393

393

Agenda

- Introduction
- JCRE (Runtime Environment)
- JCVM (Virtual Machine)
- APIs

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

394

394

APIs (1/2)

- Define a set of Java classes
 - Used to develop Java Card services (applets)
 - Dedicated for a smart card environment
- Those classes comply with current standard
 - Business Java (`java.lang.Object`)
 - ISO-7816 (e.g., APDUs)
 - Cryptographic
- Those classes do not define by themselves services

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

395

395

APIs (2/2)

- Contains 3 mandatory packages
 - `java.lang`
 - Basic classes of the language
 - `javacard.framework`
 - Framework of classes and interfaces for the core functionality of an applet (AID, APDU, Applet, ISO7816, PIN, OwnerPIN, JCSYSTEM, Util, and exceptions classes)
 - `javacard.security`
 - Core classes dedicated to cryptographic services (public/private key, random number generator,...)
- And one optional package
 - `javacardx.crypto`
 - Implementation classes for ciphering/deciphering (strong cipher)

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

396

396

java.lang

- Object { public Object();
public boolean equals(Object obj); }
- Throwable { public Throwable(); }
 - Exception
 - RuntimeException
 - ArithmeticException
 - ClassCastException
 - NullPointerException
 - SecurityException
 - ArrayStoreException
 - NegativeArraySizeException
 - IndexOutOfBoundsException
 - ArrayIndexOutOfBoundsException

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

397

397

javacard.framework

- class java.lang.Object
 - class javacard.framework.AID
 - class javacard.framework.APDU
 - class javacard.framework.Applet
 - class javacard.framework.JCSystem
 - class javacard.framework.OwnerPIN (implements javacard.framework.PIN)
 - class java.lang.Throwable
 - class java.lang.Exception
 - class javacard.framework.CardException
 - class javacard.framework.UserException
 - class java.lang.RuntimeException
 - class javacard.framework.CardRuntimeException
 - class javacard.framework.APDUException
 - class javacard.framework.ISOException
 - class javacard.framework.PINException
 - class javacard.framework.SystemException
 - class javacard.framework.TransactionException
 - class javacard.framework.Util
 - interface javacard.framework.ISO7816
 - interface javacard.framework.PIN
 - interface javacard.framework.Shareable
- } Optionally used by the developer
- class javacard.framework.ISOException used very often in applet development

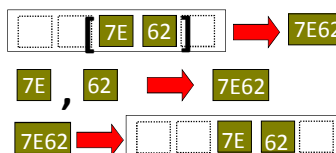
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

398

398

javacard.framework

- `public class AID`
 - Encapsulates constants related to ISO7816-5
 - Mainly used at beginning of “communication” between applets
- `public class Util`
 - Functions for byte array manipulation and comparison
 - `arrayCompare()` compares 2 arrays (*in constant time*)
 - `arrayCopy()` copies 1 array into another *atomically*
 - `arrayCopyNonAtomic()` copies 1 array into another *non-atomically*
 - `arrayFillNonAtomic()` fills an array with a byte value *non-atomically*
 - Functions for type conversion (short/byte)
 - `short getShort ()`
 - `short makeShort ()`
 - `void setShort ()`



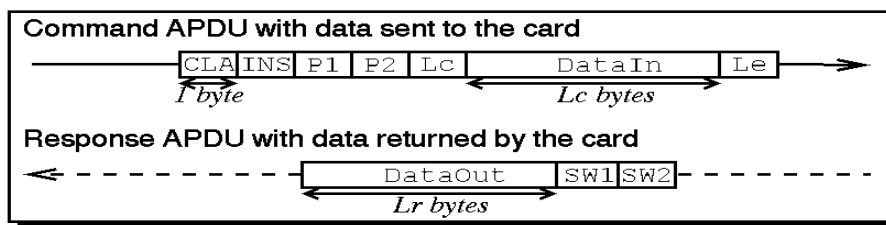
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

399

399

javacard.framework

- `public final class APDU`
 - Encapsulates 7816-4 APDU elements in an I/O buffer
 - APDUs (Application Programming Data Units) define the format of the «data packets» exchanged between a reader and a card:
 - APDUs are defined for both commands and responses to hide the low level details of the protocol used (T=0 or T=1)
 - Status words SW1 et SW2 are standardized (OK = 0x9000)



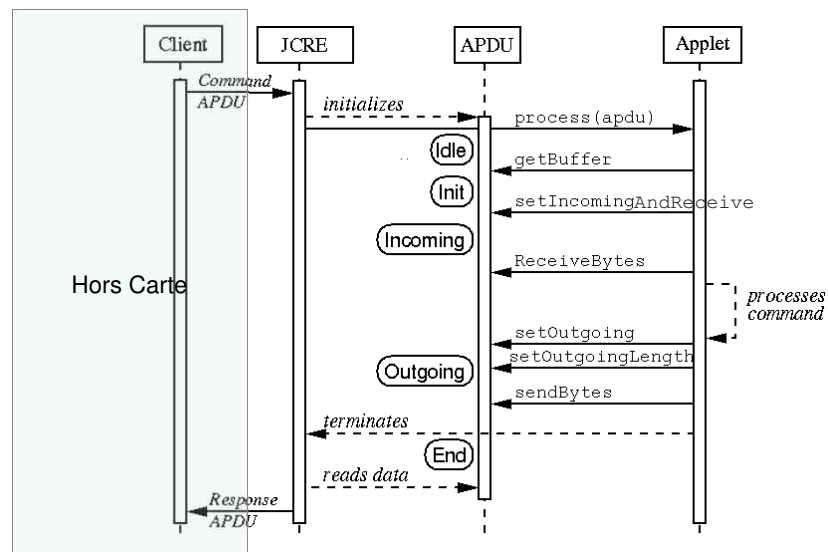
- Provides methods for **receiving** data and **sending** data
- Designed to be protocol independent
- The incoming / outgoing APDU data size may be bigger than the APDU buffer size and need to be read/write in portions by the applet.
- *APDU objects are owned by the JCRE (Temporary JCRE entry point objects)*

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

400

400

Handling the APDU class



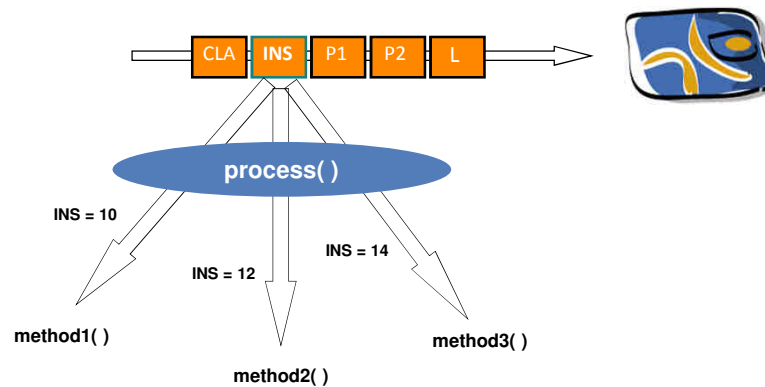
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

401

401

Handling the APDU class

- `process (APDU apdu)` method is called
- `apdu.getBuffer()`
- CLA is checked and then INS is examined



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

402

402

javacard.framework

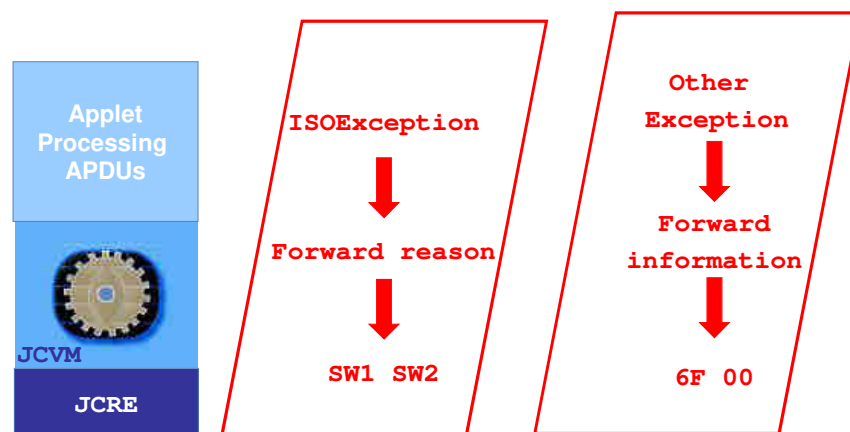
- `public interface ISO7816`
 - Contains only static constants
 - `public final static` field with **SW_** prefixes define status words (standardized status response of ISO7816-4)
 - e.g.: `SW_WRONG_P1P2` // Incorrect parameters (P1,P2) = 0x6B00
 - Fields with **OFFSET_** prefixes define constants/offset to use as index in the APDU buffer (byte array)
 - e.g.: `OFFSET_CLA` // APDU header offset: CLA = 0
 - Fields with “_00” suffix require the low order byte to be customized appropriately
 - e.g.: `SW_BYTES_REMAINING_00` // Response bytes remaining = 0x6100
- `public class ISOException`
 - Encapsulates an ISO 7816-4 response status word as its reason code
 - `throwIt(short reason)` allows to output a status word error

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

403

403

Java Card Exceptions



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

404

404

javacard.framework

- `public interface PIN`
 - Define a PIN behavior and a PIN value
 - Try limit maximum trials of an incorrect PIN before being blocked,
 - Max PIN size, the maximum length of PIN,
 - Try counter, the remaining number of trial,
 - Validated flag => true if a valid PIN has been presented. Flag is reset on every card reset.
 - If a transaction is in progress **update of the try counter shall not participate to the transaction,**

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

405

405

javacard.framework

- `public class OwnerPIN`
 - Implements interface `PIN`
 - Full rights for PIN code management
 - Update / Read capabilities -> for the code "owner" only
 - Features:
 - direct modification of the validation flag
 - know whether or not the code has been validated
 - securely implemented against attacks

Methods available:

- | | |
|-----------------------------------|------------------------------------|
| • <code>check()</code> | • <code>reset()</code> |
| • <code>isValidated()</code> | • <code>getTriesRemaining()</code> |
| • <code>resetAndUnblock()</code> | • <code>getValidatedFlag()</code> |
| • <code>setValidatedFlag()</code> | • <code>update()</code> |

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

406

406

javacard.framework

- `public final class JCSystem`
 - Static methods (natives) for interaction with the JCRE
 - Transactions
 - Only one level of transaction allowed
 - `beginTransaction()`, `commitTransaction()`, `abortTransaction()`
 - Limited by the RAM and EEPROM capacity
 - Transient array
 - `makeTransientXXXArray(length, event)`
 - Build a transient array reinitialized at reset or deselection
 - **XXX** being Boolean, Byte, Short or Object
 - event is `CLEAR_ON_DESELECT` or `CLEAR_ON_RESET`
 - Object sharing
 - Build an object that inherit from `Shareable`
 - Provide a reference with `getAppletShareableInterfaceObject(AID, Parameter)`

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

407

407

javacard.framework

- `public abstract class Applet`
 - Super class of all Java Card Applet, user applets must subclass `Applet` class.
 - Methods called by the JCRE
 - User applets **MUST** implements
 - `static void install(byte[] buf, short off, byte len)`
 - Create a new applet instance
 - `abstract void process(APDU apdu)`
 - Called for requesting the applet instance to execute a received command
 - User applets **MAY** overrides
 - `boolean select()`
 - Called when the applet instance is selected
 - `void deselect()`
 - Called when the applet instance is selected
 - Note that:
 - `protected void final register()`
 - Called during installation for registering the newly created instance

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

408

408

Développement Java Card

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

409

409

Construction d'applications carte

- Une application carte
 - Code de la carte (application serveur = applet Java Card)
 - Code dans le terminal (application cliente)
- Construction d'une application Java Card
 - Construction de l'application serveur (applet)
 - Implémentation des services
 - Installation de l'applet dans les cartes
 - Initialisation des services
 - Construction de l'application cliente
 - Invocation des services
- Définir le but de l'application puis spécifier le protocole de communication entre l'application cliente et l'applet Java Card.
 - Commandes à traiter
 - Réponses possibles
 - Cas d'erreurs

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

410

410

Construction d'une applet Java Card

- Rôle d'une applet
 - Maintenir son propre état : gestion des champs de l'applet, créer des objets et les référencer pour travailler
 - Répondre à des APDUs de commande (méthode `process()`) et retourner des APDUs de réponse
- Conception
 - Définir les structures de données nécessaires
 - Créer les objets de base à l'installation, initialiser les champs
 - Implémentation de la méthode `install()`
 - Définir les APDUs traités par la méthode `process()`
 - Implémentation d'un analyseur de commandes
 - Utilisation des champs et objets de l'applet
 - Utilisation des APIs
 - Définir les traitements à la sélection et à la désélection

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

411

411

Construction d'une applet Java Card

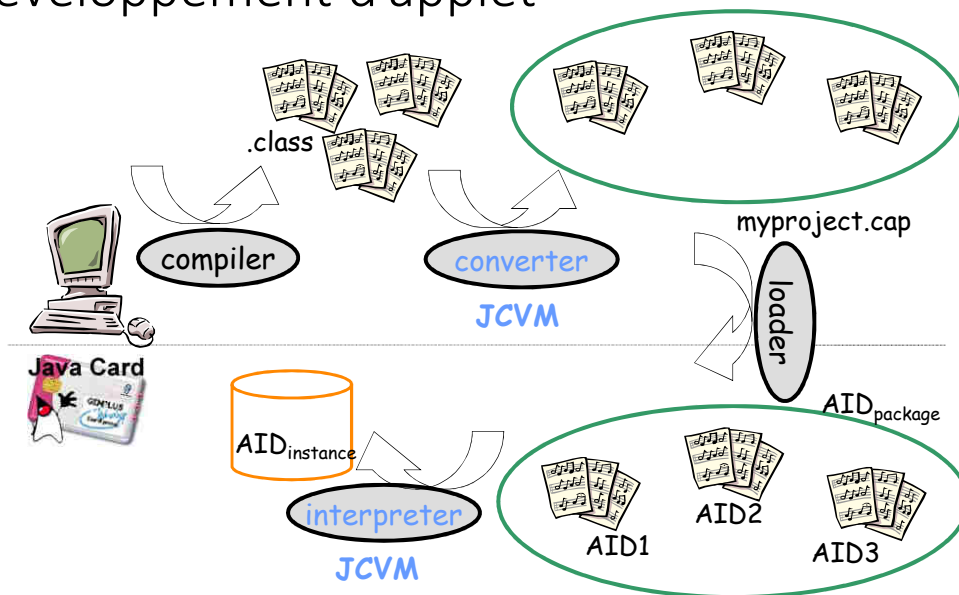
- Fabrication (avec un IDE)
 - Compilation
 - Production de fichier class
 - Conversion
 - Production du fichier CAP
- Initialisation
 - Chargement du fichier CAP
 - Installation de l'applet (instanciation) via un appel à la méthode `install()` en lui donnant un AID
- Utilisation
 - Via l'AID défini pour cette applet

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

412

412

Développement d'applet



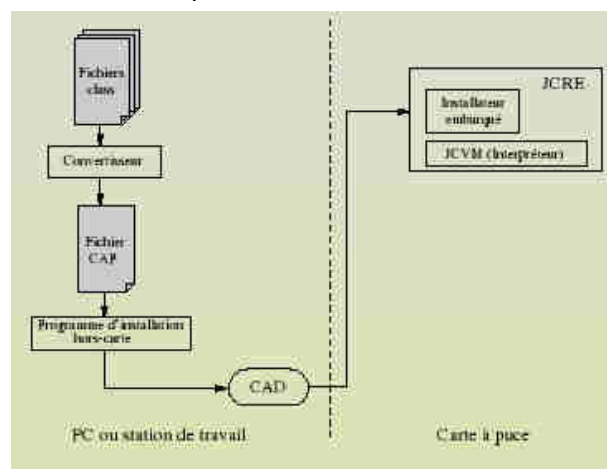
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

413

413

Installateur Java Card

- L'installation se fait grâce à deux entités :
 - l'installateur Java Card et un programme d'installation hors carte
 - Suit quasiment tout le temps le standard GlobalPlatform



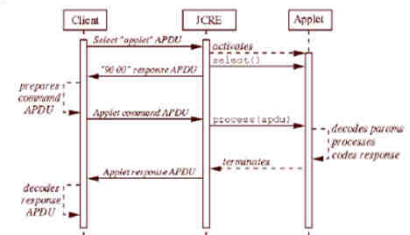
CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

414

414

Construction de l'application cliente

- Rôle de l'application cliente
 - Stimuler l'application serveur et utiliser les résultats pour faire son travail (exemple informer le commerçant que oui l'ordre a bien été accepté)
- Conception
 - Définir les structures de données propre à l'application
 - Choisir son langage, un compilateur, éventuellement un framework développement (.NET, Java, etc.)
 - Choisir un framework de communication : PC/SC, JPC/SC, OCF, javax.smartcardio, etc.
 - Implémentation en utilisant ce framework du protocole de communication avec le serveur (carte)
 - Établissement de la liaison avec le serveur : envoi d'un APDU de sélection avec l'AID de l'applet
 - Invocation des services de l'applet :
 - Codage et envoi d'APDUs de commande conformes à ceux traités par l'applet
 - Réception et décodage des APDUs de réponse retournés par l'applet
- Fabrication
 - Avec l'IDE choisi
- Déploiement
 - sur les machines où le service sera utilisé
- Utilisation



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

415

415

Résumé construction d'applications

- Développement de l'applet carte
 - Utilisation de l'API Java Card
 - Essentiellement pour traiter des APDUs
 - Conversion et installation dans la carte
 - Format du bytecode et procédure d'installation dans la carte non «encore» définis par Java Card
 - Exécution par le JCRE
 - Sélection d'applet définie par Java Card
 - Réception et transmission d'APDUs par l'applet définies par Java Card
- Développement de l'application cliente
 - Utilisation d'une API de communication carte permettant de transporter des APDUs
 - Exemple : API OCF pour Java standard
 - Sélection de l'applet par son AID
 - Commande définie par Java Card
 - Échanges d'APDUs avec l'applet carte
 - Codage d'APDUs de commande et décodage d'APDUs de réponse conformes aux APDUs traités par l'applet

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

416

416

Les difficultés du développement

- Client et applet communiquent par APDUs
 - Structure de données pauvre (tableau d'octets), peu explicite (pas de typage), et difficile à manipuler
 - Oblige à définir le contenu et la sémantique des messages échangés : décodage et encodage d'APDUs par le client et l'applet
- La spécification des échanges entre client et applet correspond à un protocole
 - Le format des messages APDUs échangés est utilisé comme spécification commune
 - Travail sur un protocole plutôt que sur des fonctionnalités
 - Nécessite une formation carte
- L'applet et le client implémentent un protocole
 - Code «duplicé» dans les programmes clients et applets
 - Dans la carte : toutes les applets décodent des APDUs
 - Code «sensible»
 - Difficile à programmer : prévoir tous les cas, pas d'outils
 - Source d'erreurs

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

417

417

Exemple «Compteur» : Conception

- État de l'applet :
 - Maintient une valeur entière positive ou nulle (pas d'objets)
- Installation : création de l'objet Applet
 - Initialisation de la valeur du compteur
 - Enregistrement auprès du JCRE
- Opérations :
 - Lecture : retourne la valeur du compteur
 - Incrémentation/décrémentation : ajoute/soustrait un montant au compteur et retourne la nouvelle valeur du compteur
- Sélection et désélection :
 - Aucun traitement

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

418

418

Exemple «Compteur» : Conception

- APDUs traités par l'applet :

- int lire()

- Commande : A0 10 XX XX 00 04

- Réponse : RV3 RV2 RV1 RV0 90 00

- int incrementer(int)

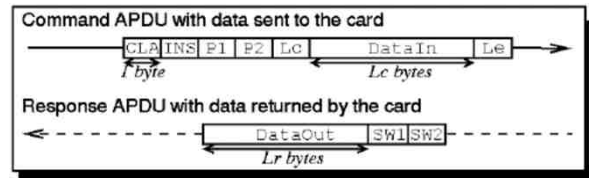
- Commande : A0 12 XX XX 04 AM3 AM2 AM1 AM0 04

- Réponse : RV3 RV2 RV1 RV0 90 00

- int decrements(int) :

- Commande : A0 14 XX XX 04 AM3 AM2 AM1 AM0 04

- Réponse : RV3 RV2 RV1 RV0 90 00



CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

419

419

Applet «Compteur» : Classe Applet

```
package fr.cryptis.compteur ;
import javacard.framework.* ;

public class Compteur extends Applet {

    private int valeur;

    public Compteur() { valeur = 0; }

    public static void install( byte[] bArray, short bOffset, byte bLength)
    { new Compteur().register(); }

    public void process( APDU apdu ) {
        byte[] buffer = apdu.getBuffer();
        if ( buffer[ISO7816.OFFSET_CLA] != (byte)0xA0 )
            ISOException.throwIt(ISO7816.SW_CLA_NOT_SUPPORTED);
        switch ( buffer[ISO7816.OFFSET_INS] ) {
            case (byte)0x10: ... // Opération de lecture
            case (byte)0x12: ... // Opération d'incrément
            case (byte)0x14: ... // Opération de décrémentation
            default: ISOException.throwIt(ISO7816.SW_INS_NOT_SUPPORTED);
        }
    }
}
```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

420

420

Un point sur les types

- Java Card uses 16 bit arithmetic, not 32 bit
 - Java Card code contains many `(short)` & `(byte)` casts: all intermediate results (which are of type `int`) must be cast to `short` or `(byte)` so that results are the same on a 16 bits Java Card VM as on a normal 32 or 64 bits Java VM

```
short s; byte b;
s = b+s+1;
// not ok, compiler complains; there is an
// implicit cast from 32 to 16 bit
s = (short) (b+s+1);
// not ok, converter complains; a 16 bit CPU will
// implicitly cast intermediate result to 16 bit
s = (short) (b+(short) (s+1)) // ok
```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

421

421

Un point sur les types

- (un)signed bytes
 - bytes in Java and Java Card are signed
 - i.e. for any byte `b`, `b ∈ {-128, ..., 127}`
 - To interpret byte as unsigned , write
 - `b & 0x00FF ∈ {0, ..., 255 }`

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

422

422

Applet «Compteur» : décrémentation

```
case 0x14: // Opération de décrémentation
{
    // Réception des données
    byte octetsLus = apdu.setIncomingAndReceive();
    if ( octetsLus != 4 )
        ISOException.throwIt(ISO7816.SW_WRONG_LENGTH);

    int montant = (buffer[ISO7816.OFFSET_CDATA]<<24) |
        (buffer[ISO7816.OFFSET_CDATA+1]<<16) |
        (buffer[ISO7816.OFFSET_CDATA+2]<<8) |
        buffer[ISO7816.OFFSET_CDATA+3];

    // Traitement
    if ( montant<0 || valeur-montant<0 )
        ISOException.throwIt((short)0x6910);
    valeur = valeur - montant;

    // Envoie de la réponse
    buffer[0] = (byte)(valeur>>24);
    buffer[1] = (byte)(valeur>>16);
    buffer[2] = (byte)(valeur>>8);
    buffer[3] = (byte)(valeur);
    apdu.setOutgoingAndSend((short)0, (short)4);
    return;
}
```

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

423

423

Rappel : Applet Java Card

- Une applet carte est un programme serveur de la Java Card
 - APDU de sélection depuis le terminal
 - Sélection par AID (chaque applet doit avoir un AID unique)
- Une fois installée dans la carte, l'applet est toujours disponible
 - Classe qui hérite de `javacard.framework.Applet`
 - Doit implémenter les méthodes qui interagissent avec le JCRE :
 - `public static void install( byte[] bArray, short bOffset, byte bLength )`
 - Appelée (une fois) par le JCRE quand l'applet est chargée dans la carte
 - Doit s'enregistrer auprès du JCRE (méthode `register()`)
 - `public boolean select()`
 - Appelée par le JCRE quand un APDU de sélection est reçu et désigne cette applet
 - Rend l'applet active
 - `public abstract void process( APDU apdu )`
 - Appelée par le JCRE quand un APDU de commande est reçu pour cette applet (doit être active)
 - `public void deselect()`
 - Appelée par le JCRE pour désélectionner l'applet courante

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

424

424

Réalisation d'applets Java Card malveillantes

CaP et Dev JC - Limoges - 2019-2020 - D. Sauveron

425