

La carte à puce

Damien Sauveron

damien.sauveron@unilim.fr

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

(et tous les auteurs cités en bibliographie)

1

Intervenants dans le module

Damien Sauveron (Cours, TD et TP)

- Membre de l'équipe « Cryptis » en charge du projet « Sécurité des Systèmes et Réseaux » (P.F. Bonnefoi, E. Conchon, K. Damme)
- Thème de recherche : sécurité des systèmes embarqués et des réseaux mobiles

Pour me contacter :



Damien Sauveron

XIM UMR 7252 CNRS -- Université de Limoges

123 avenue Albert Thomas

87060 Limoges Cedex, FRANCE

Email: Damien.Sauveron@unilim.fr

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

Phone: +33 (0) 5 87 50 67 93

MCF à l'Université de Limoges depuis septembre 2006 – HDR (2014)

Visite postdoctorale au Smart Card Centre de l'Information Security Group du Royal Holloway University of London

Thèse au LaBRI – Université Bordeaux 1 sur la Sécurité de la Technologie Java Card

Ingénieur R&D dans le CESTI de SERMA Technologies (Pessac)

Organisateur de WISTP2007 (Héraklion) et WISTP2008 (Séville)

Vice président du groupe de travail de l'IFIP : Pervasive System Security

2

Plan

La carte à puce et les concepts de base

Qu'est-ce qu'une carte ?

Historique

Deux classifications:

- Par technologie pour la puce
- Par technologie de communication

Standard ISO 7816

- Protocole de communication

Chaîne de fabrication

Acteurs

Applications

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

Bibliographie

Rendez vous

Java Card Grid Projet

3

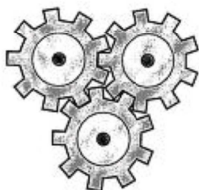
Qu'est ce qu'une carte à puce

un morceau de plastique de la taille d'une carte de crédit

un circuit électronique capable de manipuler (stocker, calculer, etc) des informations



+



NE PAS DIFFUSER



**Smart = intelligente !
Pourquoi ?**

5

Historique



La carte à puce, une innovation française ?

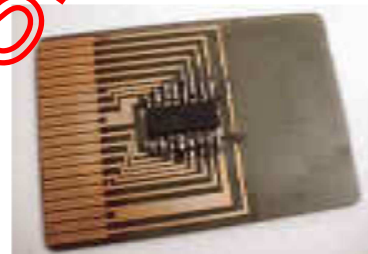
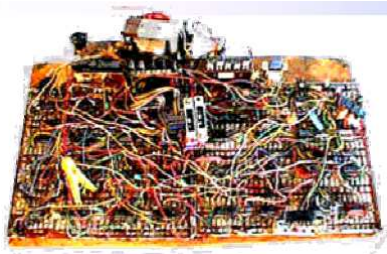
En France : OUI

Dans le reste du monde : NON

En 1967-1968, deux ingénieurs Allemands **Jürgen Dethloff** et **Helmut Grötrupp** introduisent un circuit intégré dans une carte plastique (déposent un brevet en 1969)

En mars 1970, Kunitaka Arimura au Japon dépose un brevet sur la carte à puce.

En 1971, Paul Castrucci de IBM dépose aux USA un brevet intitulé *Information Card*



6

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

Prototype de puce portable imaginé par Moreno :



11/06/1945 - 29/04/2012

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. »

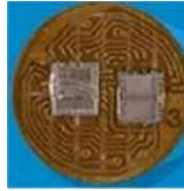
Mise au point de moyens inhibiteurs (par Moreno) logés sur la puce :

- 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 ;

En 1997, Roland Moreno est entré au National Museum of American History.

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)

8

Historique

En 1979, la première carte est créée par Bull CP8 (1Ko 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 » (carte à microprocesseur)

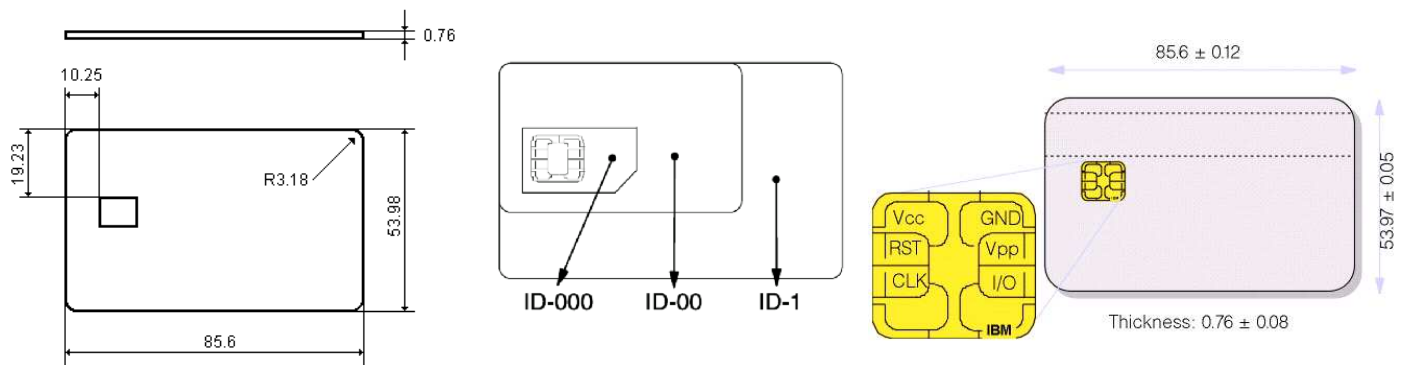


9

Historique

Entre 1984 et 1987, normes ISO 7816 (carte à puce à contact)

- **Objectif : Permettre aux cartes de fonctionner partout dans le monde !**



En 1997, apparition des premières Java Cards

- 30 ans après des nouveautés dans le monde de la carte. Enfin !!



10

Premières entreprises de la carte

- La société Innovatron créée par Moreno pour exploiter ses brevets.
- En 1979, Schlumberger (géant des services pétroliers) entre au capital d'Innovatron pour 34 %.
- Schlumberger devient le numéro 1 mondial de la carte à puce (cartes, lecteurs, cabines téléphoniques, systèmes),
- En 1997, Schlumberger acquiert le concurrent français : SOLAIC.
- En 2001, Schlumberger acquiert le concurrent français Bull CP8.
- En 1981, le GIE « Carte à Mémoire » lance trois expérimentations de la carte à puce avec: Bull, Philips, et Schlumberger.
- En 1984, le GIE devient « Cartes Bancaires ».
- Fin des années 80, GIE « CB » commande 16 millions de cartes CP8, généralisant l'utilisation de la carte à puce en France en 1992.

Vers les géants de la carte à puce

- En 1988, Marc Lassus crée Gemplus en France
- En 2001, l'activité « Cartes à puce » est cédée à Schlumberger qui externalise cette activité à Axalto
- En 2004, Schlumberger qui veut se recentrer sur les métiers du secteur pétrolier, se sépare d'Axalto en l'introduisant à la bourse de Paris.
- En 2006, fusion de Axalto et de Gemplus =>
Gemalto (numéro 1 mondial de la carte à puce)
- Exemples d'entreprises concurrentes aujourd'hui : Oberthur Technologies, Sagem, Orga

12

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

15

La carte à mémoire

Premier modèle de cartes à puce

Majorité des cartes vendues dans le monde en 1999 (on va voir quelques chiffres plus récents)

Elle possède :

- une puce mémoire de 1 à 16 Ko
- une logique câblée non programmable

Avantages :

- sa technologie simple
- son faible coût (1€)

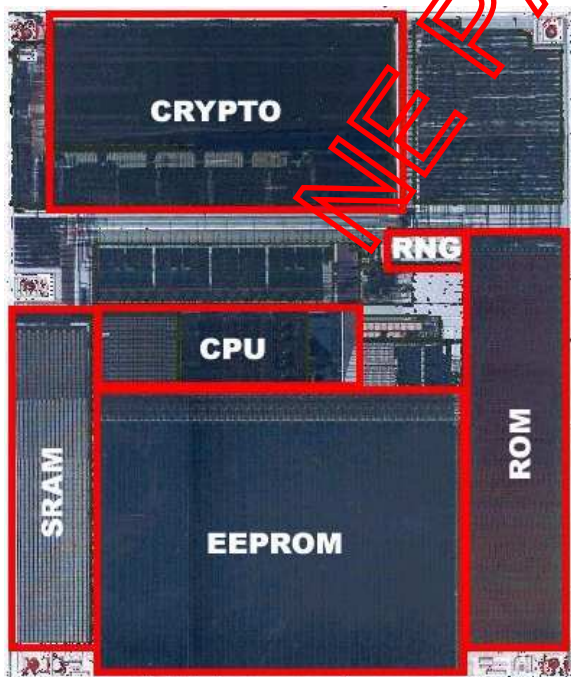
Inconvénients :

- sa dépendance vis-à-vis du lecteur de carte
- « assez » facile à dupliquer

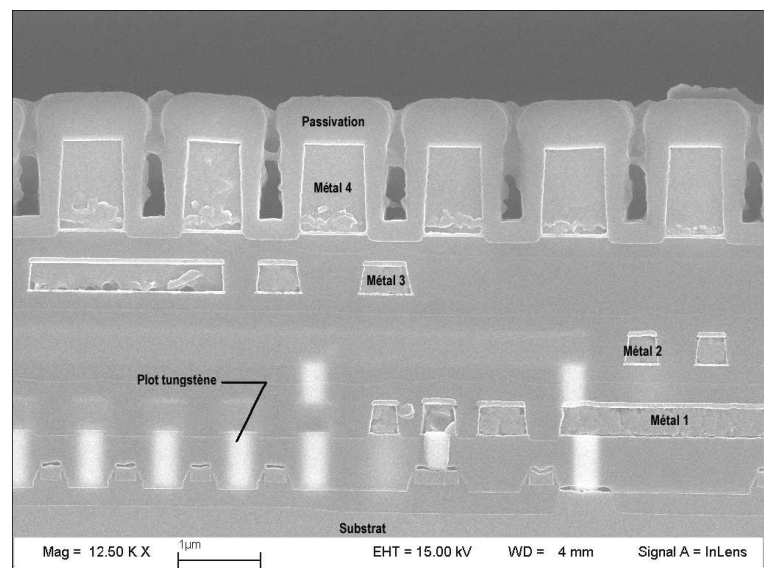
17

La carte à microprocesseur

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



Vue de dessus



Vue en coupe

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

18

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

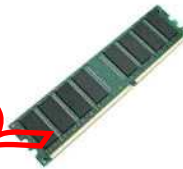
Mémoire persistante => stocke les données applicatives
Problèmes : durée de vie limitée et temps d'accès lent



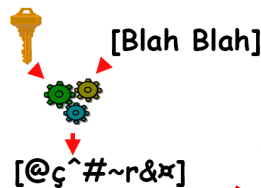
RAM : 1 à 4 Ko

Mémoire de travail

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



Coprocésseur cryptographique



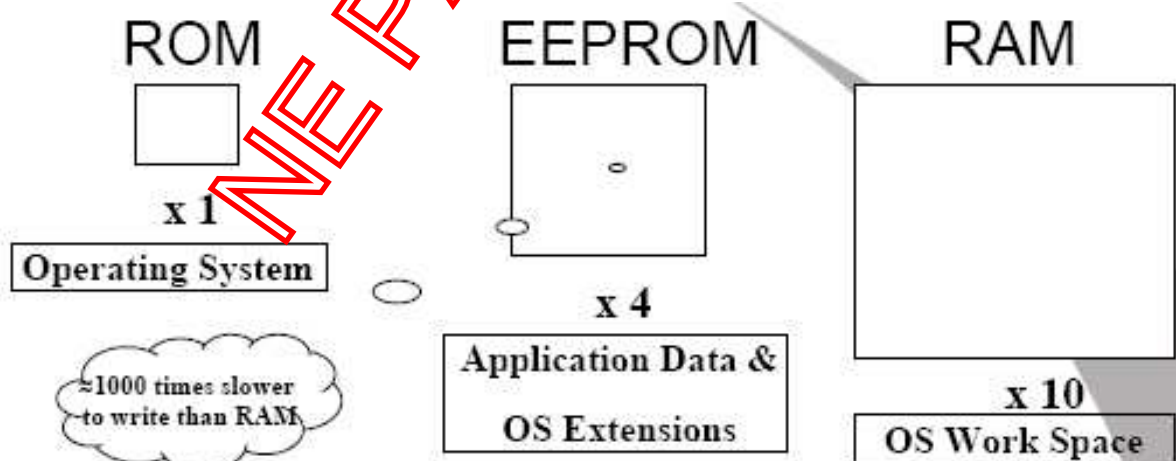
Générateur de nombres aléatoires (RNG)



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

19

Les différents types de mémoires



	ROM	EEPROM	RAM
Min	~3Kb	~4b	~128b
Max	~128Kb	~64Kb	~3Kb

EPROM : Write once, read FOR EVER !
- Used for initialization area (eg. Lock bytes)

20

Les différents types de mémoires

	RAM Statique	EEPROM	Flash-RAM	Fé-RAM (expérimentale)
Persistance	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

21

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

22

Carte à contact

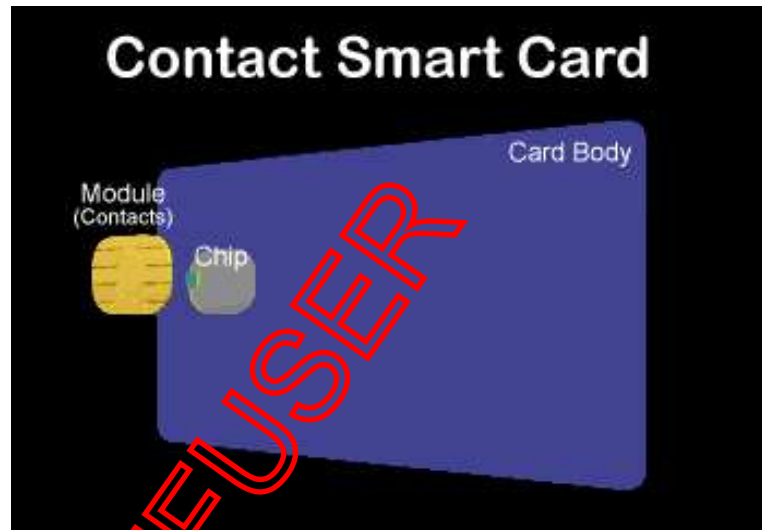
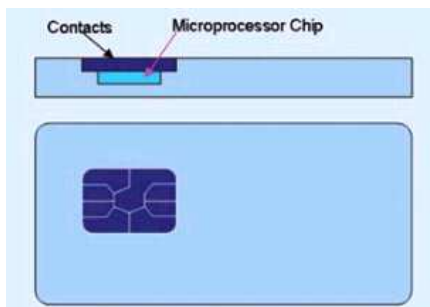
Suit le standard ISO 7816

Communication série via huit contacts => insertion dans un le



Problèmes :

l'insertion et le retrait sont des facteurs d'usure de la carte
orientation de la carte dans le lecteur

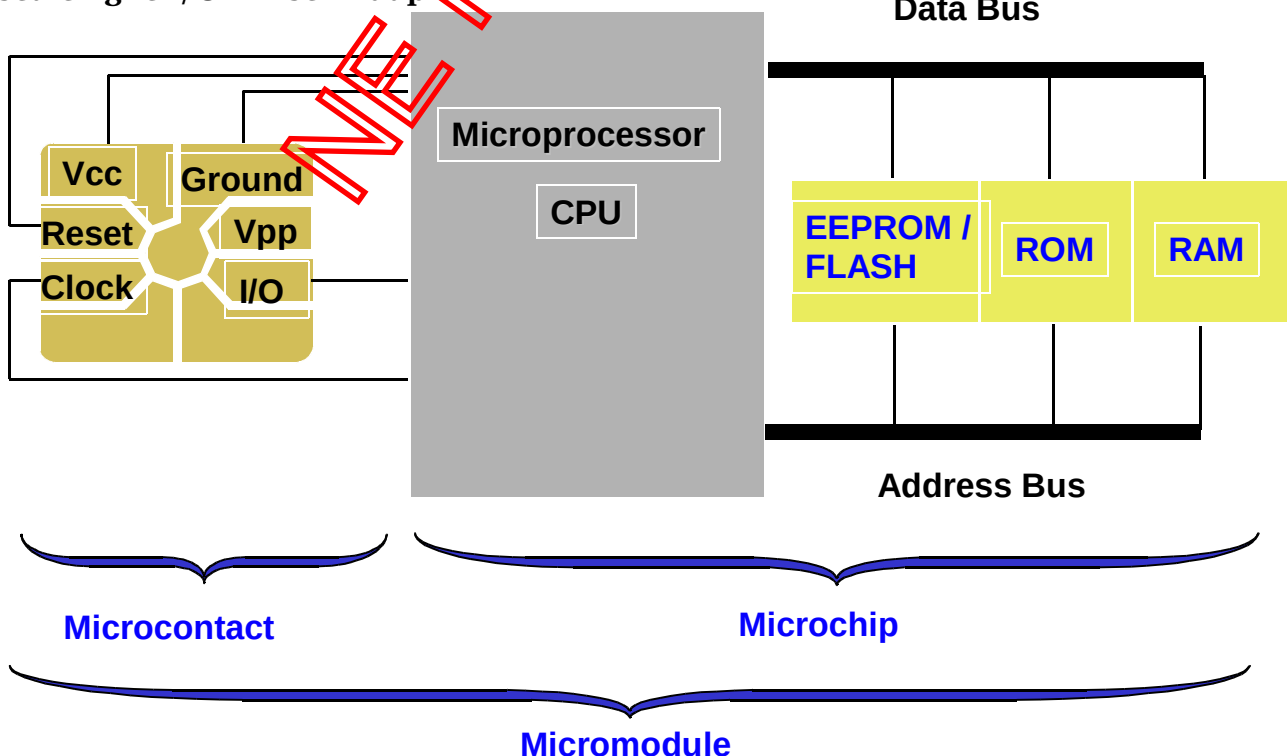


23

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 seule ligne I/O => semi duplex



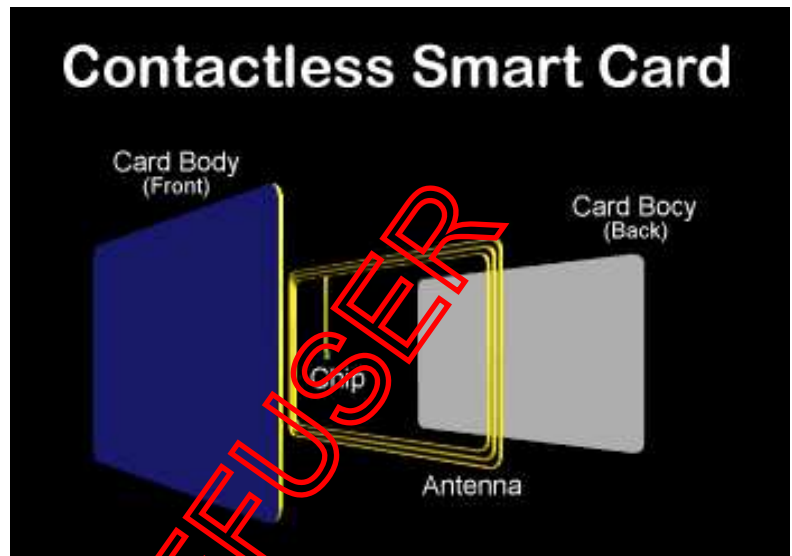
24

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 :

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

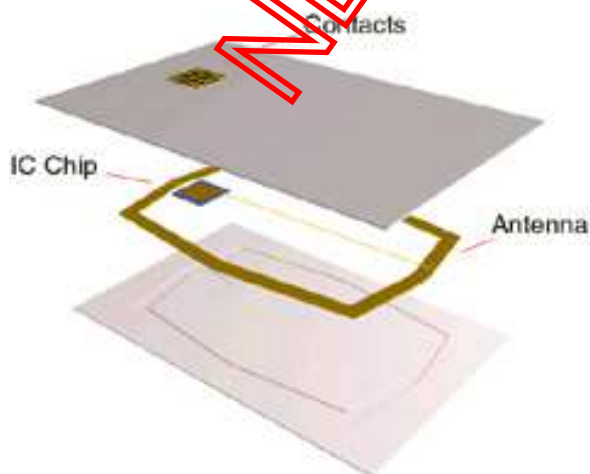


26

La carte combi

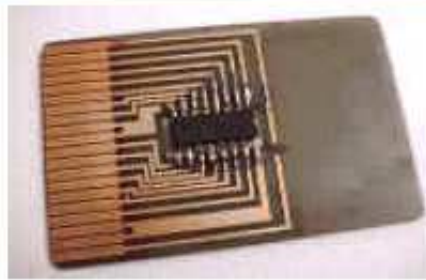
C'est une combinaison entre :
la carte à contact
et la carte sans contact

Ces deux possibilités de communication en font une carte « idéale »



28

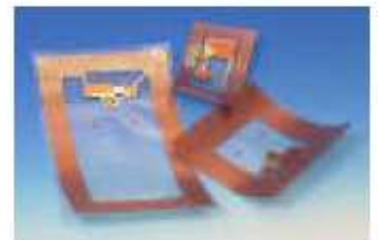
Quelques spécimens de mon Zoo



Cartes à Microprocesseur, 1987-2007

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



Les terminaux cartes

- Contact / sans contact
- Simple / complexe (keypad, LCD, multi-slot, biometrie ...)

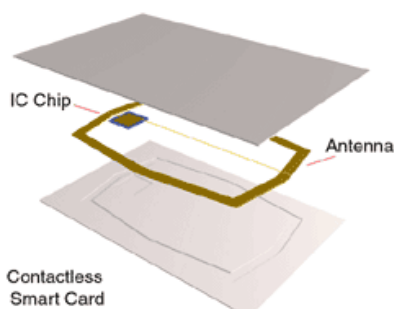


Quelques périphériques pouvant intégrer une puce sécurisée



Anneau Java

IButton



Dongles : Série, parallèle, USB

1999 Card Shipments figures per sector

Quelques chiffres

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

	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	

Worldwide Market 2004

Cards (Millions of units - Mu)			
By segments	Memory (MU)	Microprocessor	
			Growth % From 2003
Telecom	710	1050	57%
Financial services / Retail / Loyalty	35	280	37%
Government/ Healthcare	20	45	12%
Transport	60	15	25%
Pay TV	-	55	57%
Corporate Security	10	12	20%
Others	10	12	20%
Total H1 2004	845	1469	54%
Total 2004	2314		

33

Quelques chiffres

Worldwide Market 2005

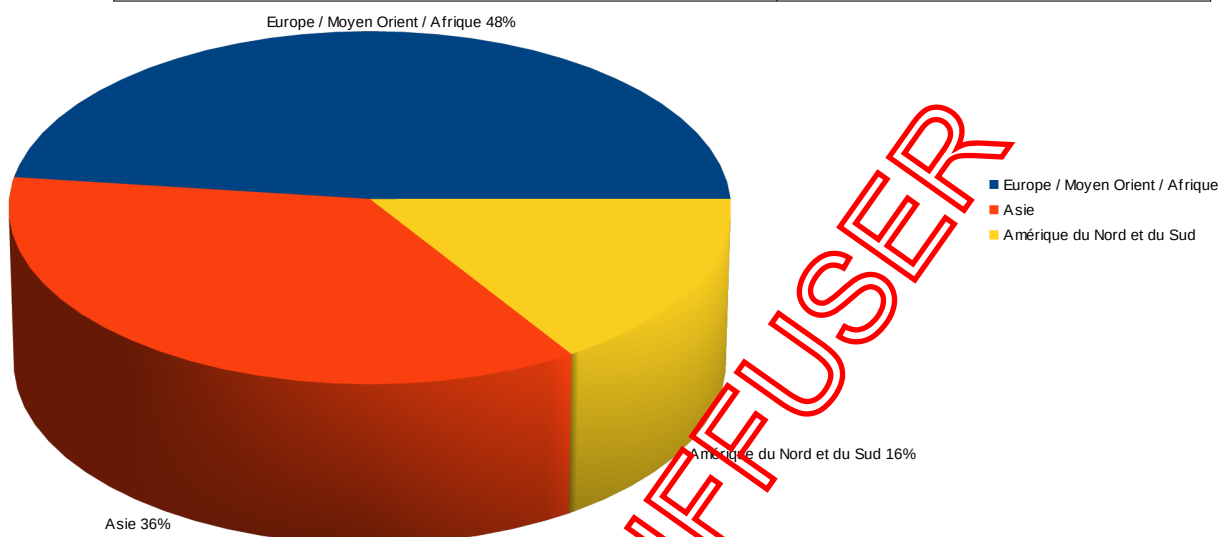
Cards (Millions of units - Mu)			
Sectors	Memory (MU)	Microprocessor	
			Growth % From 2004
Telecom	580	1390	32,38%
Financial services / Retail / Loyalty	30	336	20%
Government/ Healthcare	25	60*	33,33%
Transport	73	20	-
Pay TV	-	55	-
Corporate Security	20	15	-
Others	10	12	-
Total	738	1 888	28,52%
Total 2005	2626		

34

Répartition géographique du marché

Chiffres de 2005

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



35

Prévisions 2015

Worldwide Secure Elements shipments 2013 & 2014 shipments and 2015 forecasts
(Million of units)

Source: Eurosmart, April 2015

WW shipment 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

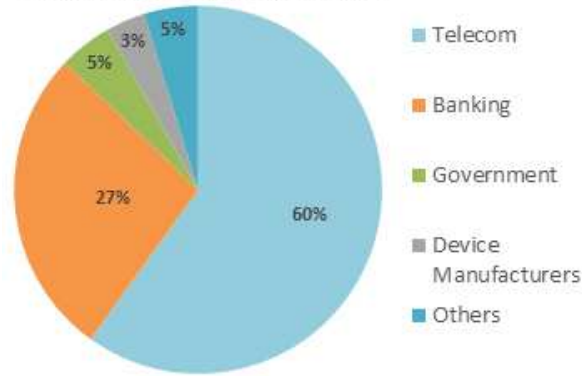
Of which contactless	2013	2014	2015f	2015f vs 2014 % growth
Financial services	590	880	1050	19,3%
Government – Healthcare	200	230	260	13,2%
Transport	160	180	210	16,7%
Others*	70	70	70	0%
Total	1,020	1,360	1,590	17%

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

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

41

Worldwide Secure Elements shipments in 2015 by Sector



42

Few attacks

- Smart card is a tamper resistant token not a tamper proof one.
 - Attacks to the system can come from
 - Human error (e.g. entering wrong data)
 - Unintentional fraud (e.g. equipment failure)
 - Intentional fraud
 - Misuse of equipment
 - Passive attacks (e.g. listening without modifying)
 - Difficult to detect
 - Preventable
 - Active attacks (e.g. generation, modification of messages)
 - Generally detectable
 - Prevention difficult
- etc.

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 show 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

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...

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.

The audit of the ATM's log file showed that although the thief presented three false PIN codes he could somehow try again and 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 (V_{pp}) supplied through a specific ISO contact. The thief had covered this contact with a sticker.

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.

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.

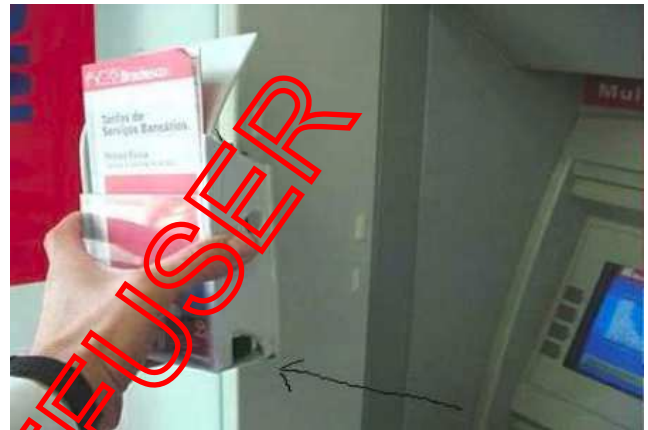
Skimming

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



Skimming

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



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.

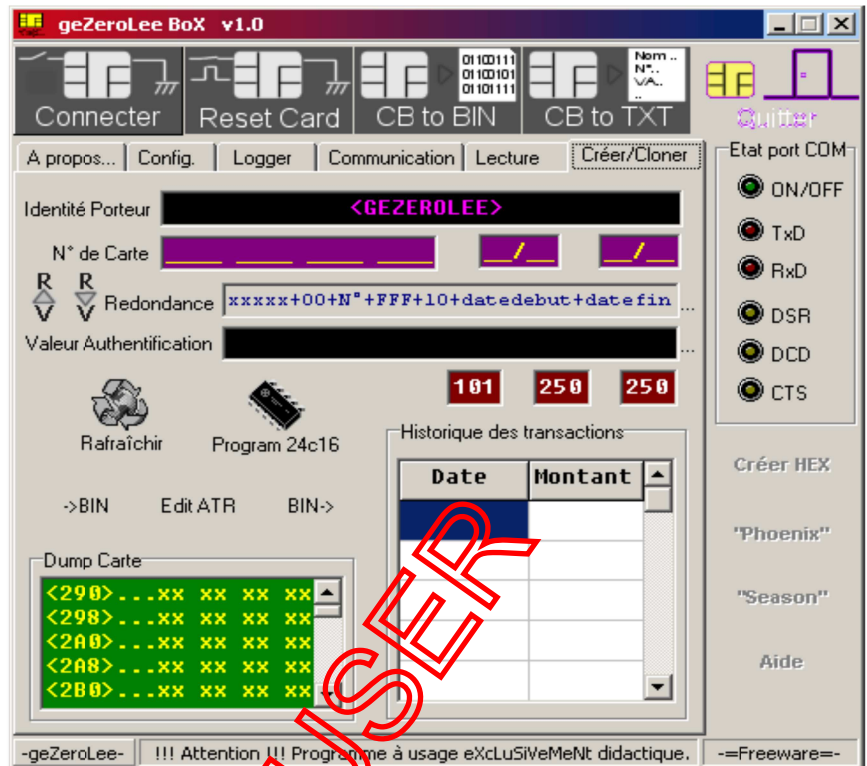
- What happened ?

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.

Yescard



Le logiciel G0lee pour la fabrication de Yescard

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 authorisation
- Merchant terminal will be detected by the merchant or he/she is himself part of the attack.

Yescard

- Context

Weakness known by industrial experts

- Off line authentication : public key
- On line authentication : secret key
- Ks 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

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*...

Fraud conclusion

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

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

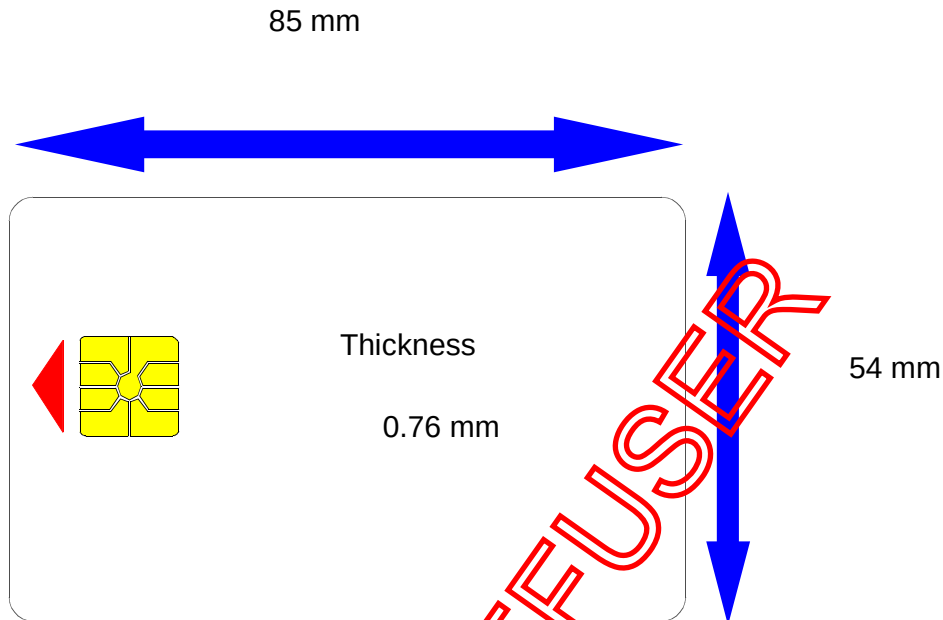
Partie 7 : Commandes inter-industrie pour SCQL (Structured Card Query Language)

Partie 8 : Sécurité de l'architecture et des commandes inter-industrie

ISO 7816-1

Définit les caractéristiques physiques : dimensions, flexibilité, résistivité.

Définit aussi les formats plus petits (SIM)



58

ISO 7816-1

Formats normalisés des cartes à puce

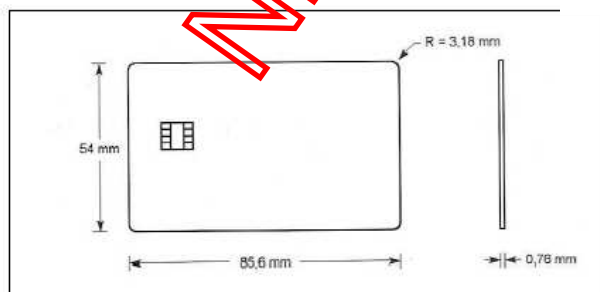
➤ Même si on connaît en général deux formats de la carte à puce

✓ Celui de la carte bancaire

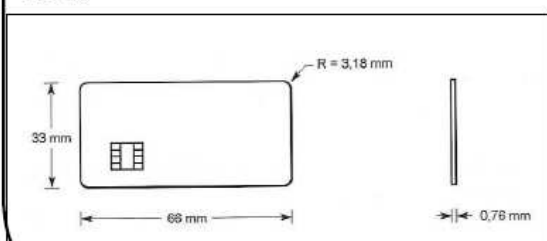
✓ Celui de la carte SIM

➤ 3 formats normalisés : ID1, ID00 et ID000

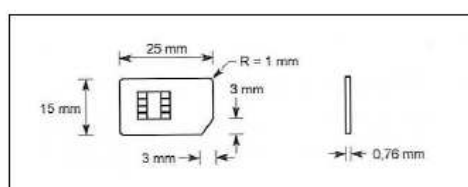
Les 3 formats



ID 00



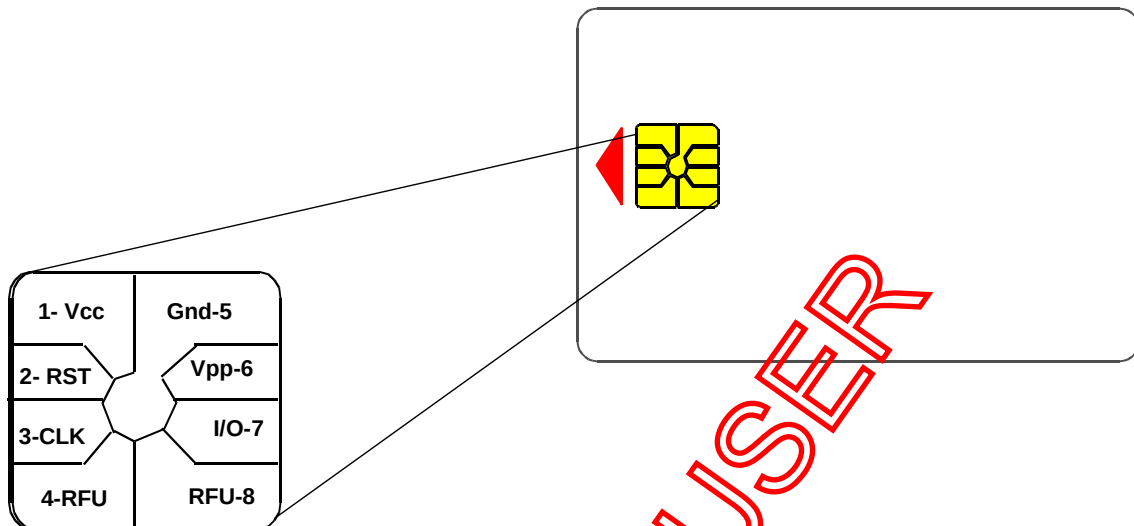
ID 000



59

ISO 7816-2

Définit les aspects électriques et la situation des contacts sur la carte.



60

ISO7816-2

Normalisation AFNOR / ISO

La position des contacts : position AFNOR et position ISO



Carte ISO

Carte AFNOR

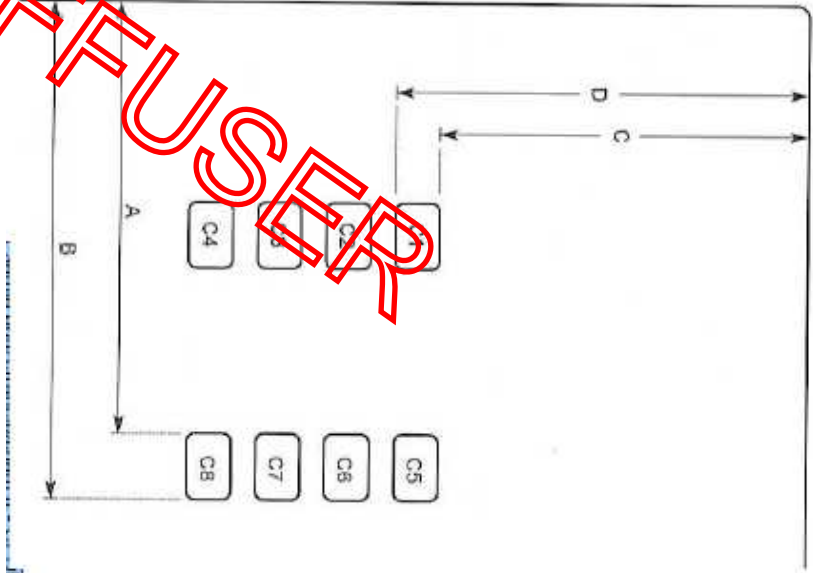
61

Position de la puce % au 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

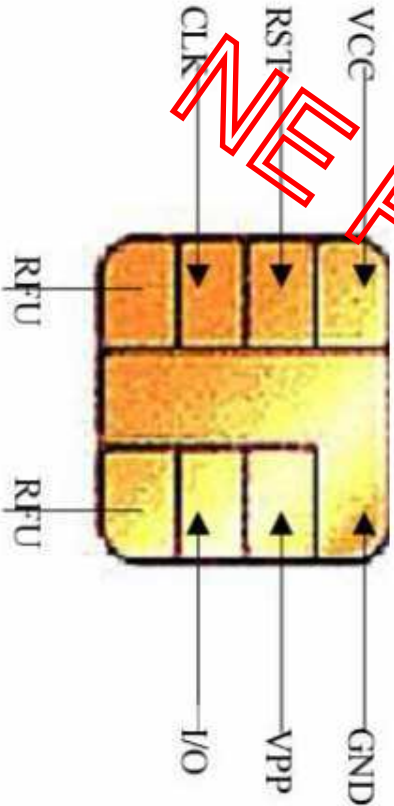
Valeurs en mm



ISO 7816-2



Les différents contacts



Vcc: tension électrique (3 à 5 V)

RST: c'est le « reset », initialise le microprocesseur (warm reset)
cold reset = coupure et rétablissement de l'alimentation

CLK: signal d'horloge car pas d'horloge sur la carte

GND: masse

Vpp: utilisé dans les anciens modèles pour avoir une autre source d'alimentation
I/O: utilisé pour le transfert des données et des commandes entre la carte et le lecteur.

ISO 7816-3

Cette partie normalise :

les caractéristiques électriques comme :

- la fréquence d'horloge (entre 1Mhz et 5Mhz)
- la vitesse des communications (jusqu'à 115200 bauds)

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)

Correspond à la mise en route du prog ROM de la carte

- Max. 33 caractères et 5 champs

- Permet de fixer :

Les conventions de codage des octets

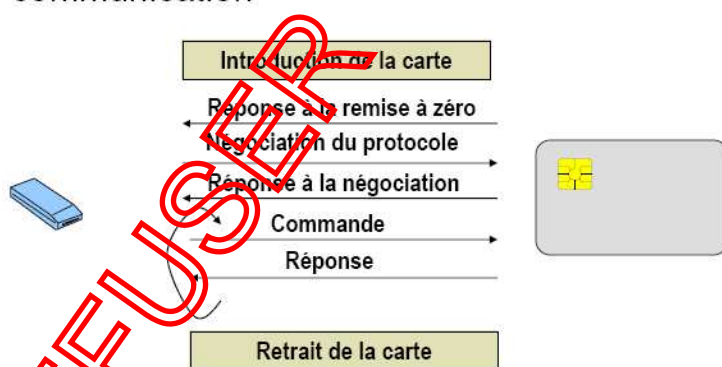
Le temps de transmission d'un bit

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)

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



04

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

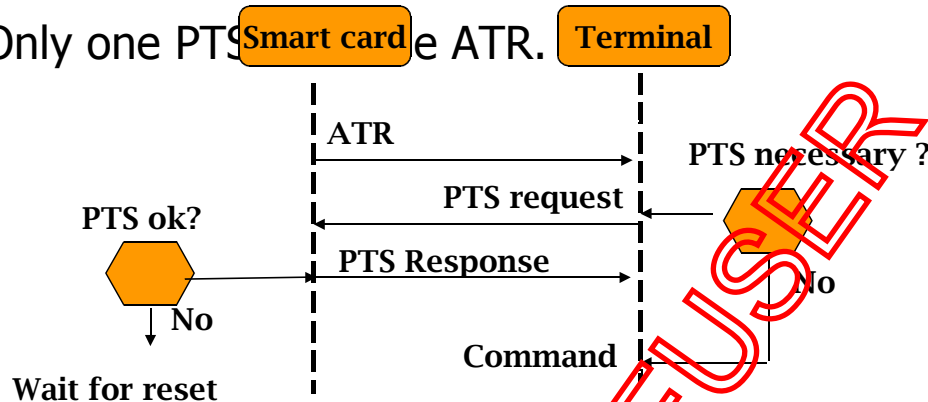
Cyberflex Access e-gate 32K

ATR = 3B 75 94 00 00 62 02 02 00 80

65

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



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

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

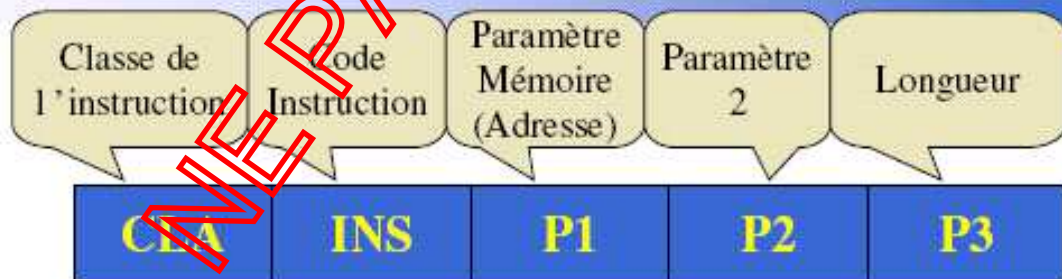
Block oriented, Header : NAD, PCB, LEN; data : INF, CRC.

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 bloc,
- S specific command (RESYNC, IFS, ABORT, WTX)

T=0 : Structure de l'ordre



- Commande

- CLA 00 pour ISO, AA pour GSM, ...
- INS pair pour l'ISO

- Réponse

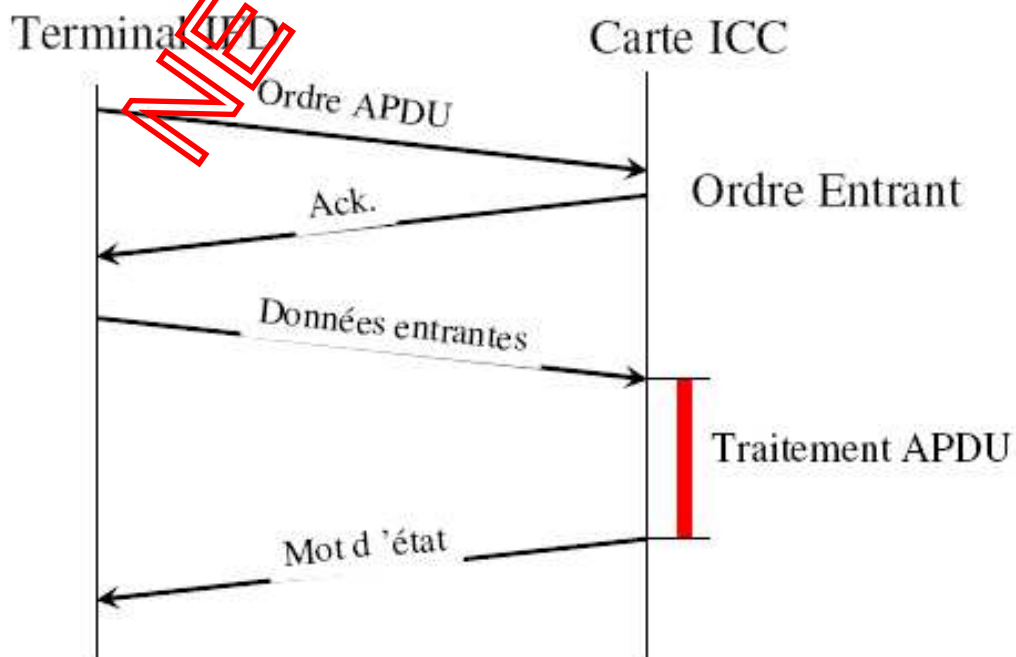
- SW1/SW2
 - 90 00
 - 6E XX: Classe inconnue
 - 6D XX: instruction inconnue
 - 9X XX: Erreur applicative

CLA Class byte

b7 to b4	b3	b2	b1	b0	Meaning
			X	X	Logical channel number
	0	0			No secure messaging
	1	0			Secure messaging header not authentic
	1	1			Secure messaging header authentic
'0'					Structure and coding compliant with 7816-4
'8','9'					User specific codes
'A'					Structure and code defined in additional
Class		Application			
'80'		Electronic purse compliant with EN 1546-3			
'8x'		Credit card compliant with EMV-2			
'A0'		GSM compliant with GPRS 300 608/GSM 11.11			

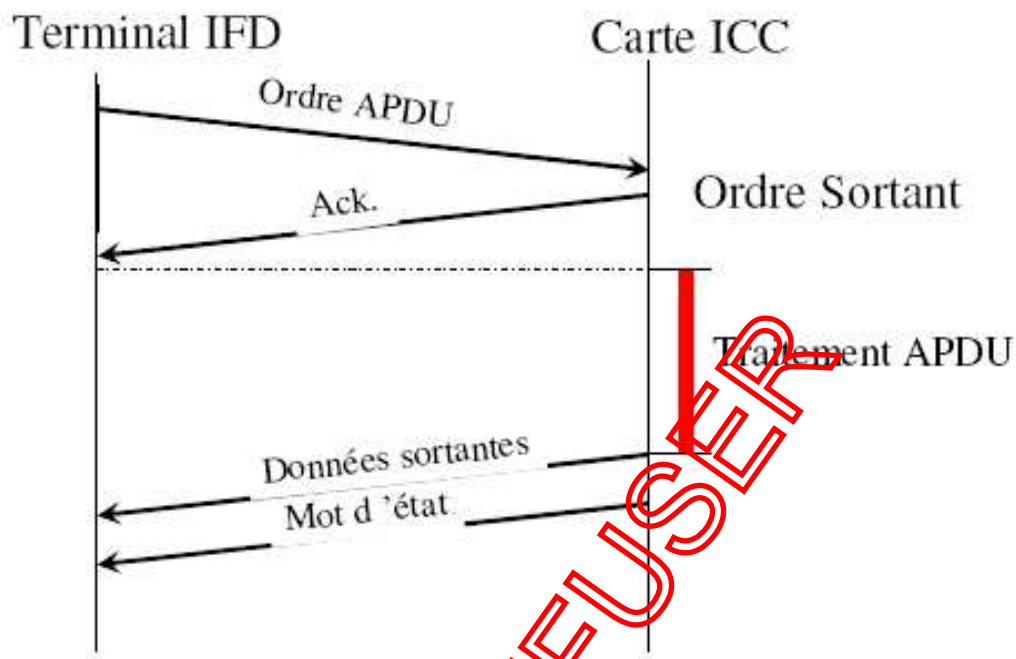
Echange de Données : T=0

Transmission de caractères asynchrones



Echange de Données : T=0

Transmission de caractères asynchrones



72

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

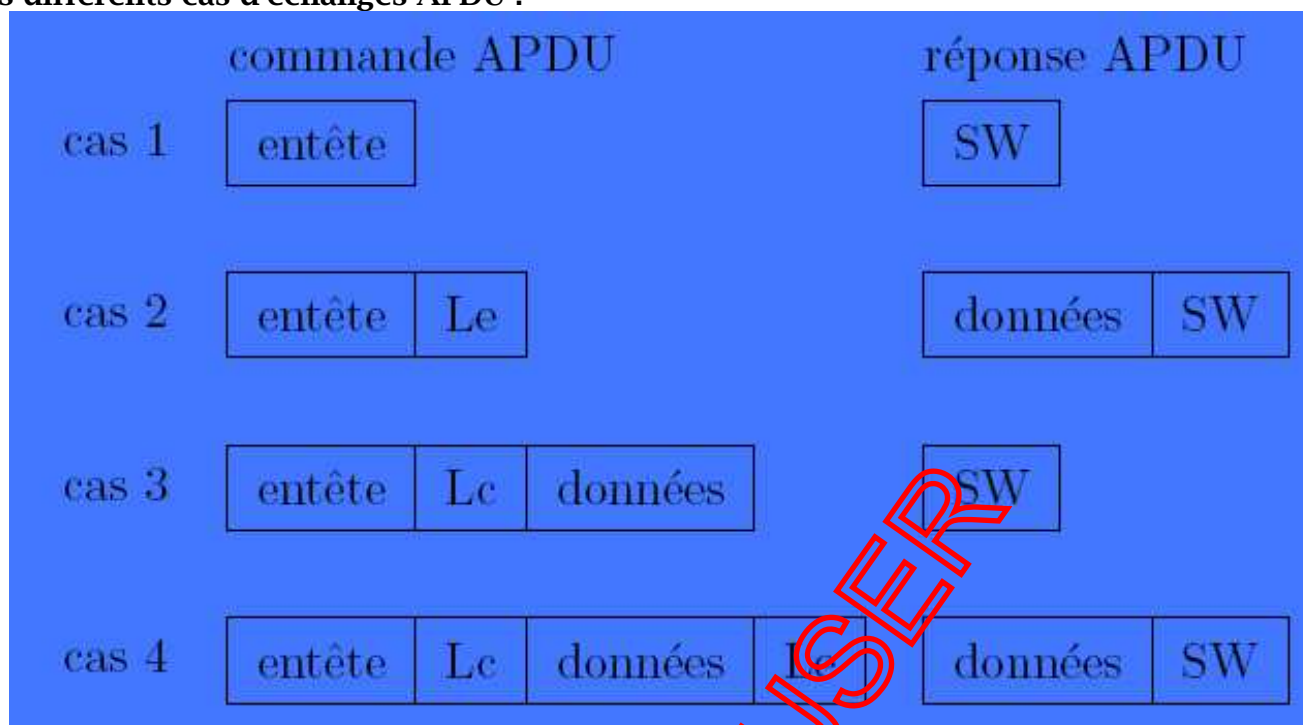
Entête obligatoire				Corps optionnel		
CLA	INS	P1	P2	Lc	Champ de données	Le

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

Corps optionnel		Enqueue obligatoire	
Champ de données		SW1	SW2

73

Les différents cas d'échanges APDU :



Standardise différents codes de réponse SW1/SW2

90 00

6E XX: Classe inconnue

6D XX: instruction inconnue

9X XX: Erreur applicative

74

APDU Command-Response

- Different configurations

Command without data, response without data

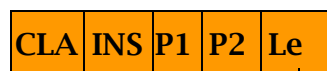


Le='00'



"90" "00"

– Command without data, response with data length known



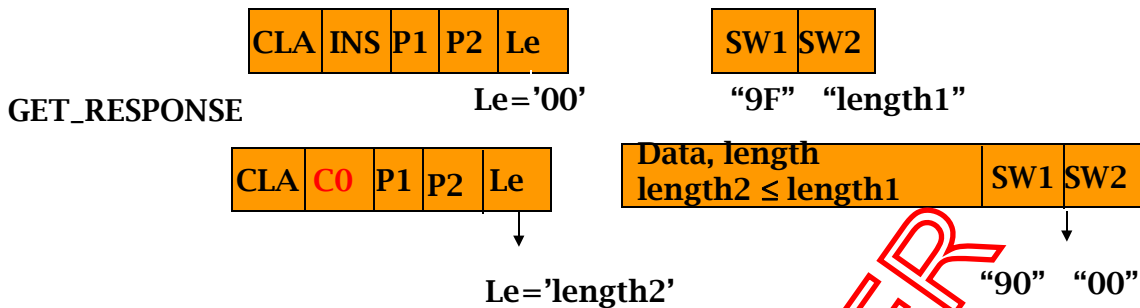
Le='length'



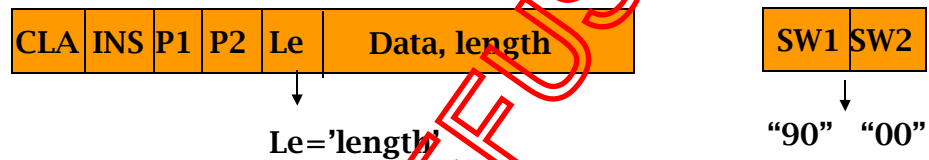
"90" "00"

APDU Command-Response

Command without data, response with data, length unknown

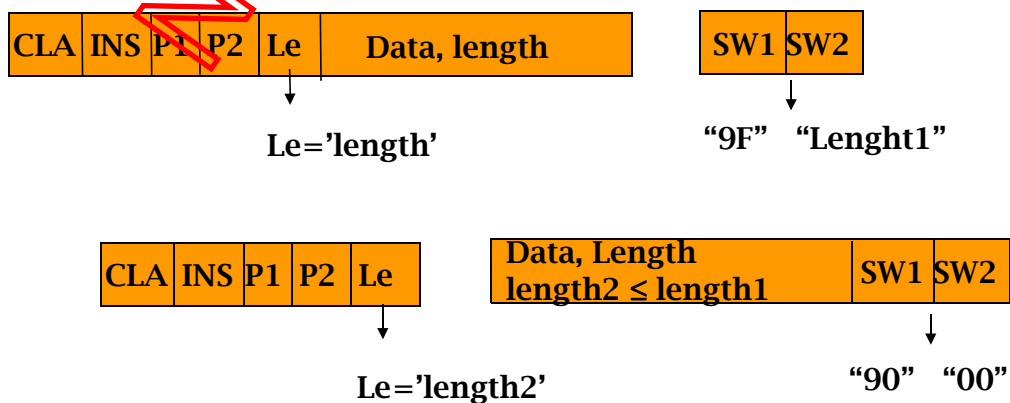


– Command with data, response without data



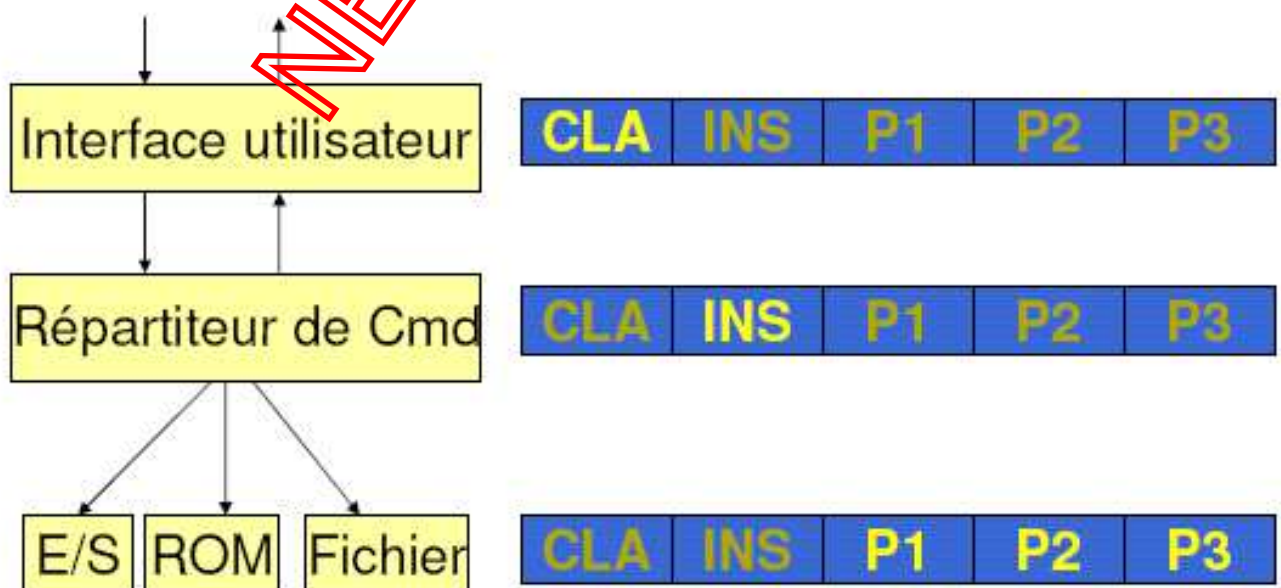
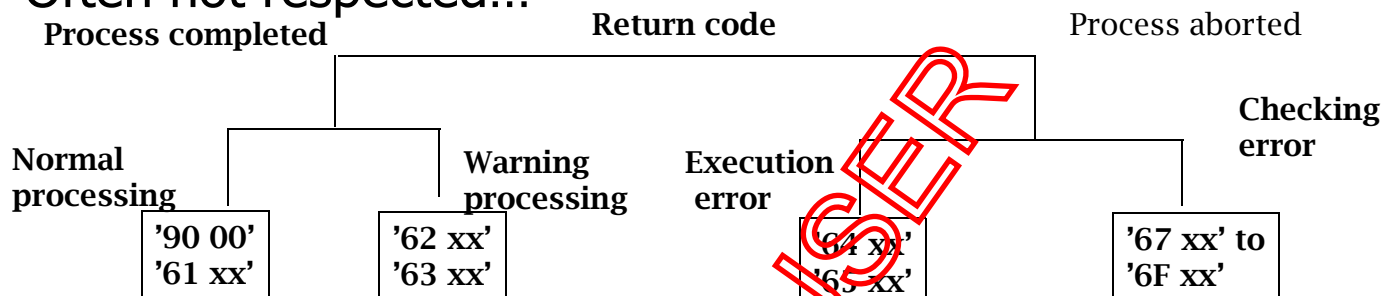
APDU Command-Response

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

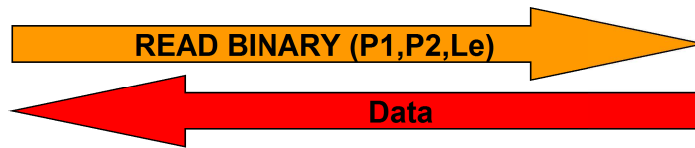


Return Codes

- 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...



Example



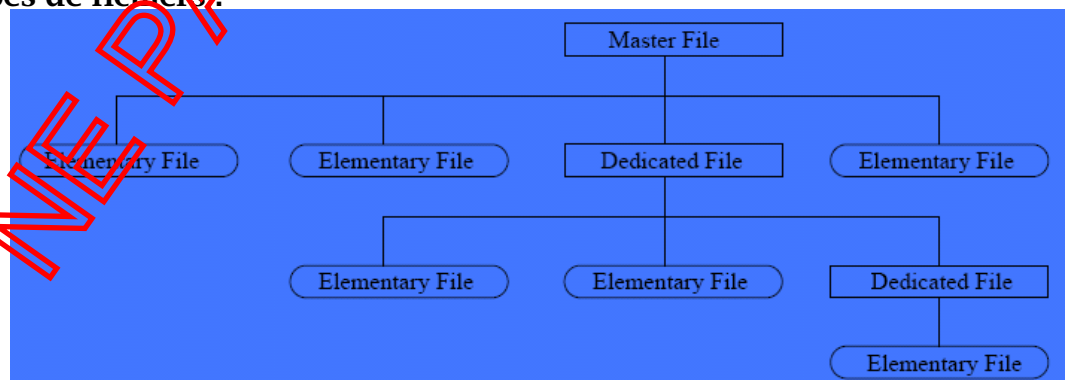
- P1=Offset High,
- P2=Offset low.

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

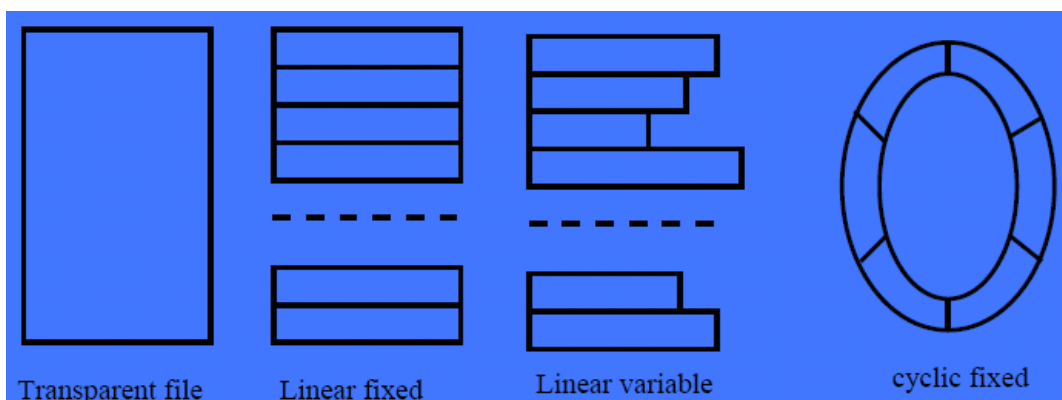
ISO 7816-4

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)

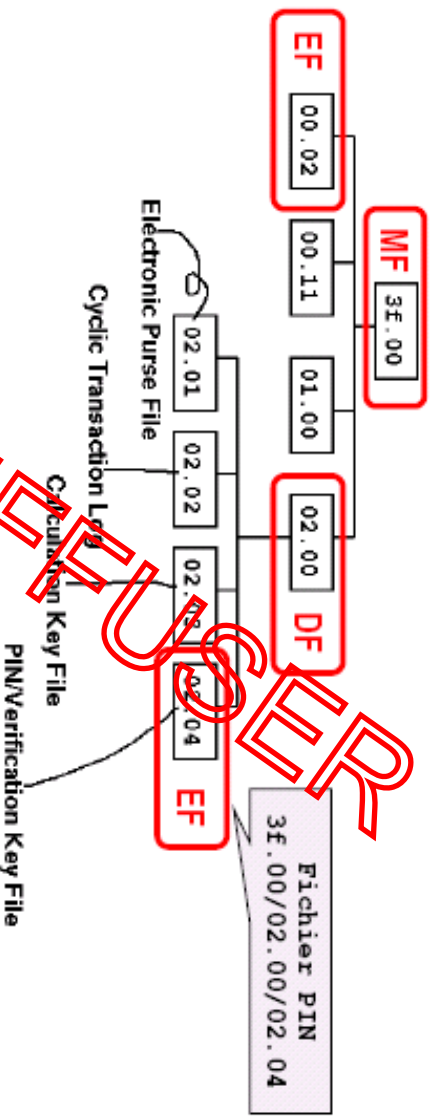


Les différents types de « Elementary File » (EF)

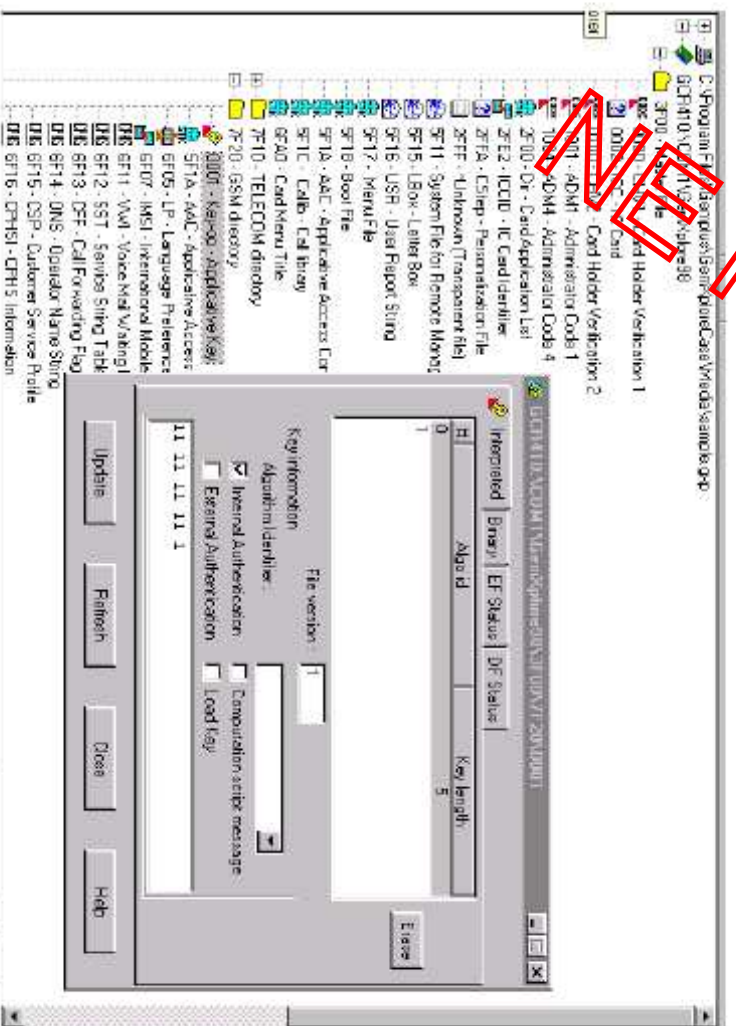


Système de Fichier ISO 7816-4

- 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



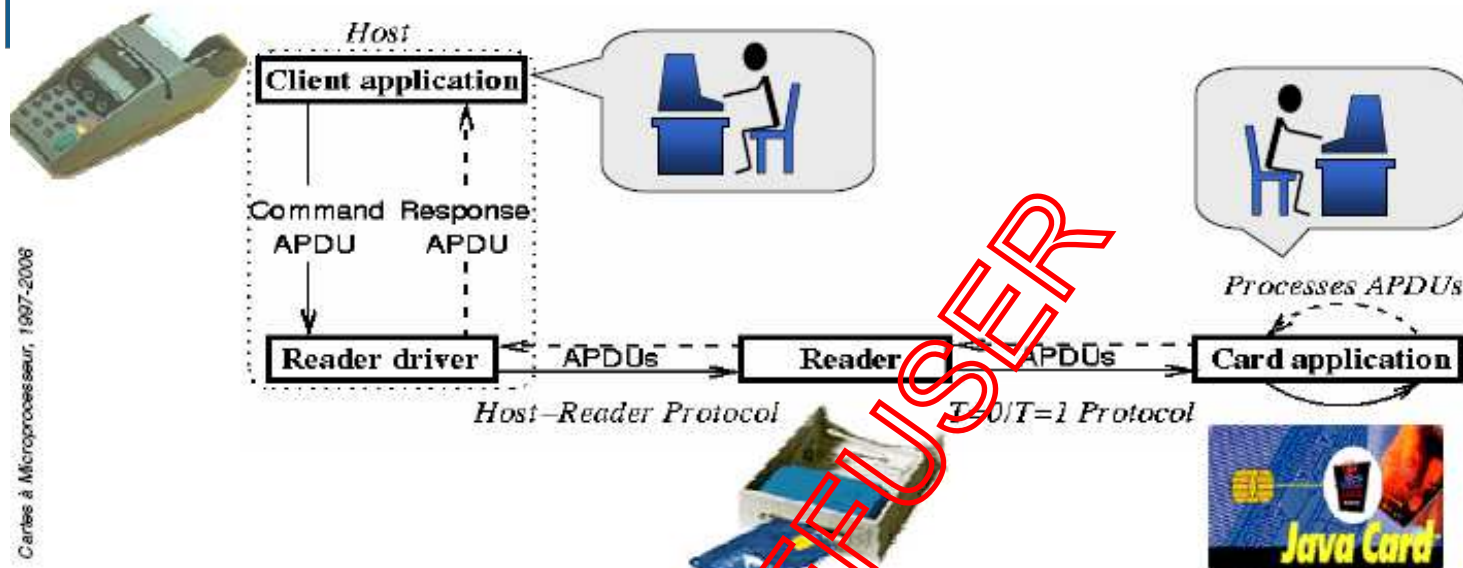
Exemple de système de fichier d'un module **SM/GSM**



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)



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 :



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

ISO 7816-6

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

Etiquette	Longueur	valeur
-----------	----------	--------

Exemple :

Tag	L	Value	Applies to
'80'	2	Number of data bytes in the file, excluding structural information.	Transparent EFs
'81'	2	Number of data bytes in the file, including structural information if any	Any file
'82'	1	File descriptor byte (see table 3)	Any file
'82'	2	File descriptor byte followed by data coding byte (see table 86)	Any file
'82'	3 or 4	File descriptor byte followed by data coding byte and maximum record length.	EFs with record structure
'83'	2	File identifier	Any file
'84'	1 to 16	DF name	EFs
'85'	var.	Proprietary information	Any file
'86'	var.	Security attributes (coding outside the scope of this part of ISO/IEC 7816)	Any file
'87'	2	Identifier of an EF containing an extension of the FCI	Any file
'88' to '9E'		RFU	
'9FXY'		RFU	

87

ISO 7816-7

Propose des commandes inter-industrie pour SCQL.

Create table	Nom de la table: Etudiants
Create view	Nom
Select	Age
Read	faculté
Write	Année
Update	
	Pierre 19 Sciences 1 ère
	Marie 23 Medecine 5 ème
	Alain 21 Droit 3 ème
	Henri 21 Histoire 3 ème
	Jean 20 Economie 2 ème
	Marc 22 Droit 4 ème

88

ISO 7816-8

Concerne la sécurité de l'architecture et des commandes inter-industrie.

89

ISO 7816-9, ISO 7816-10, ...

A vous de chercher !

NE PAS DIFFUSER

90

Des normes ... encore des normes !

Norme Europ Telecom Standard Institut:

GSM 11.11 : définition de la carte SIM

TE 9 : spécification de cartes et terminaux

EMV, (cartes de paiement)

ICAO, (agence de l'ONU, biométrie, passeport)

Santé (ISO 21 549)

...

91

Pourquoi utiliser la carte à puce ?

Avantages :

la sécurité
la portabilité
la facilité d'utilisation
la personnalisation

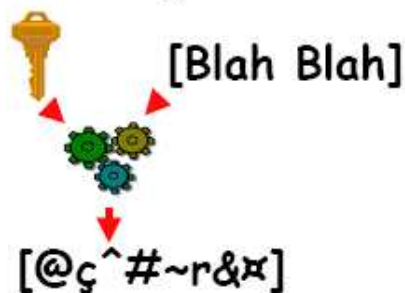
- Tamper resistance
 - Storage



- Portability



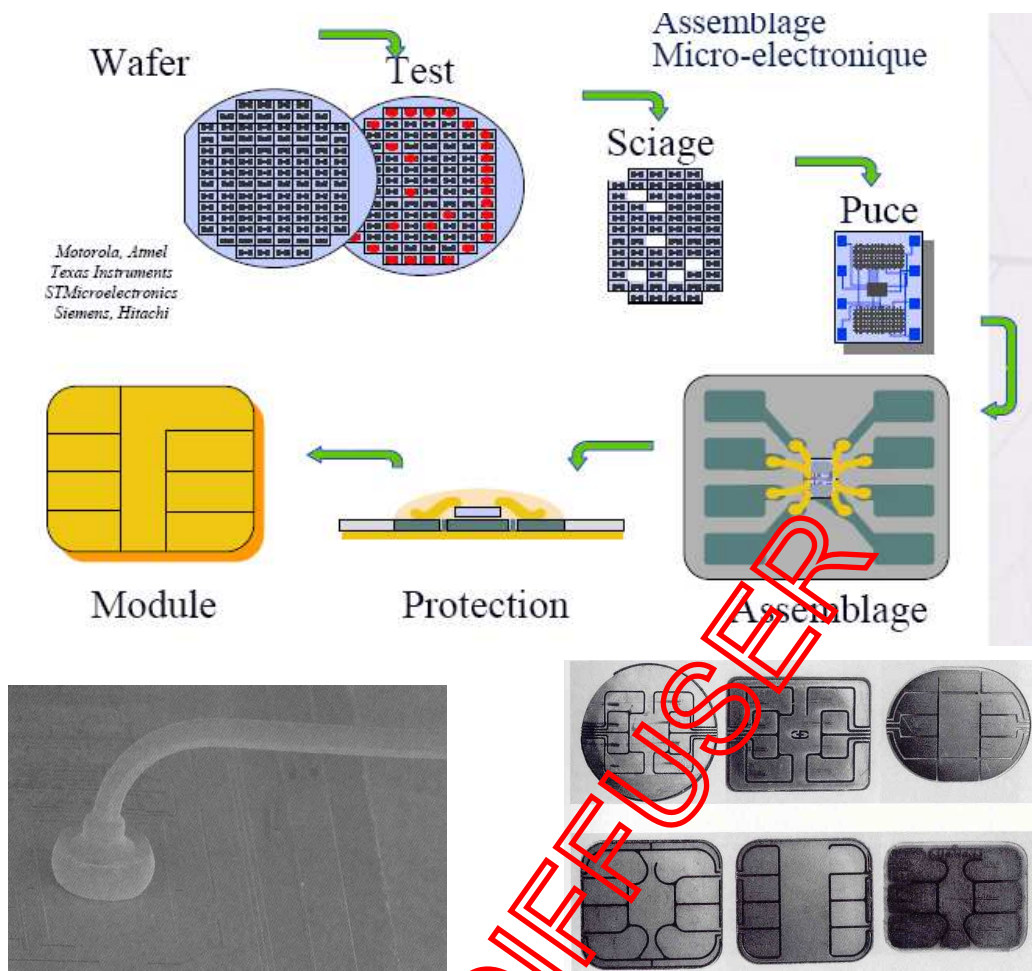
- Tamper resistance
 - Processing



- Ease of use
- Onboard key generation
- Cost

92

Cycle de fabrication



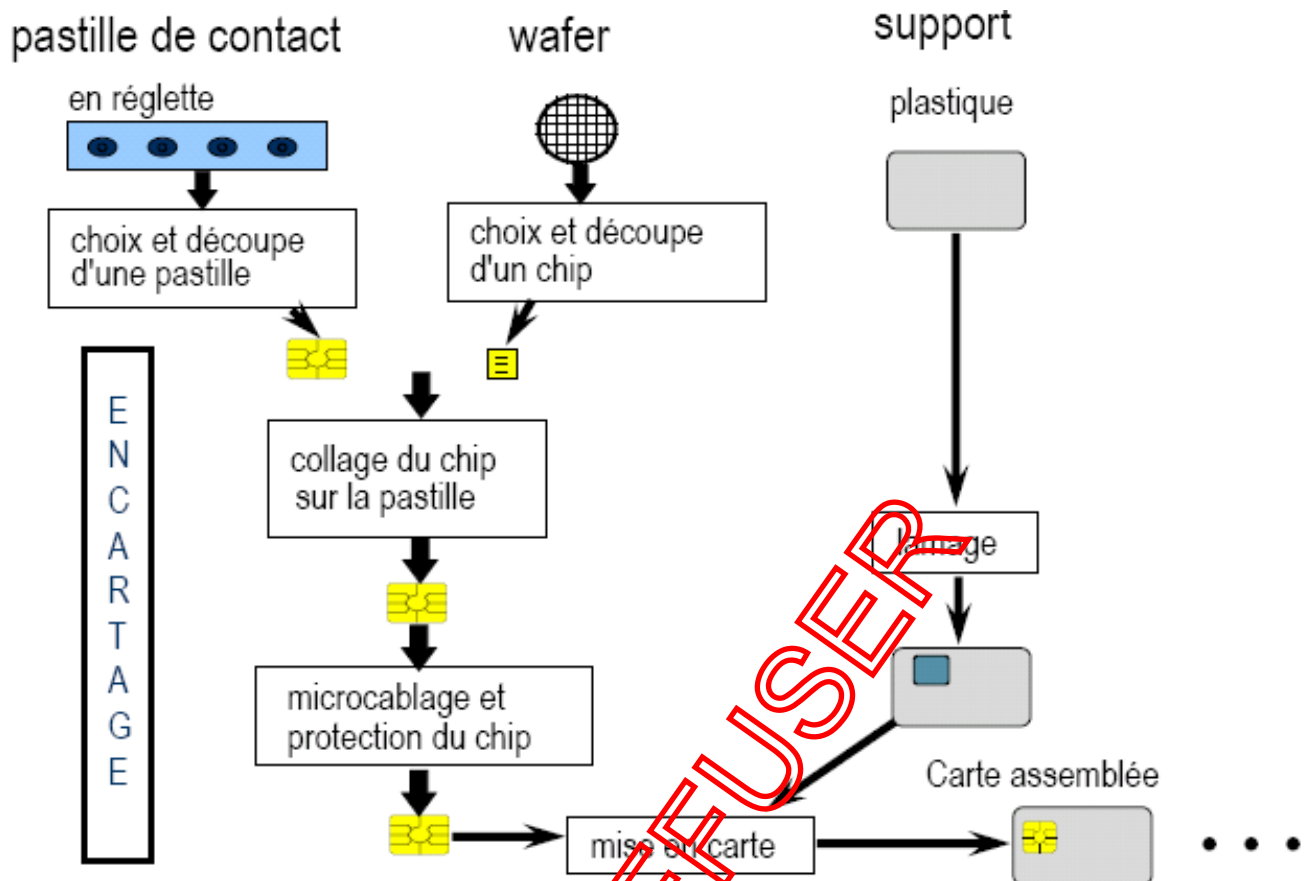
93

Cycle de fabrication

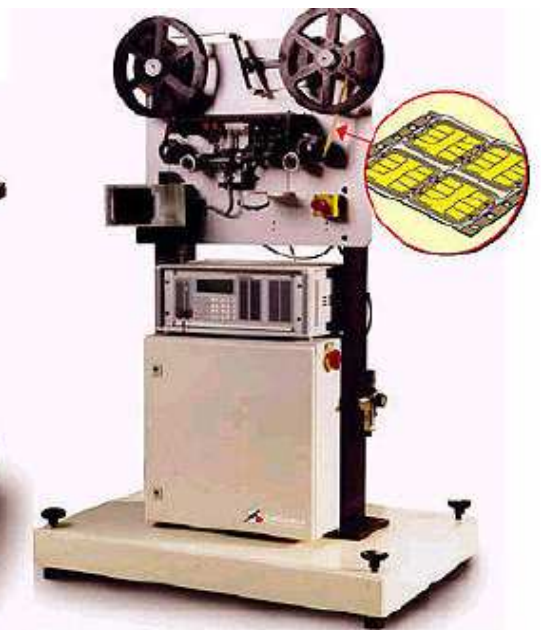
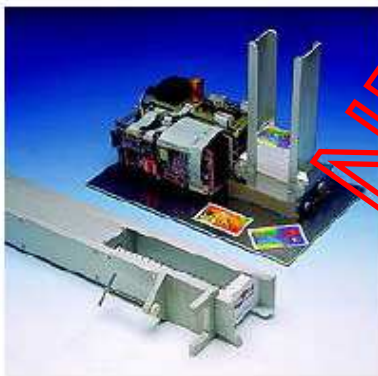


94

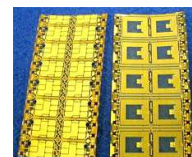
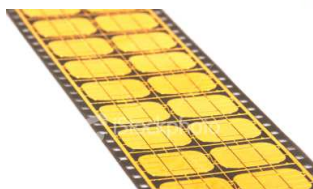
Cycle de fabrication

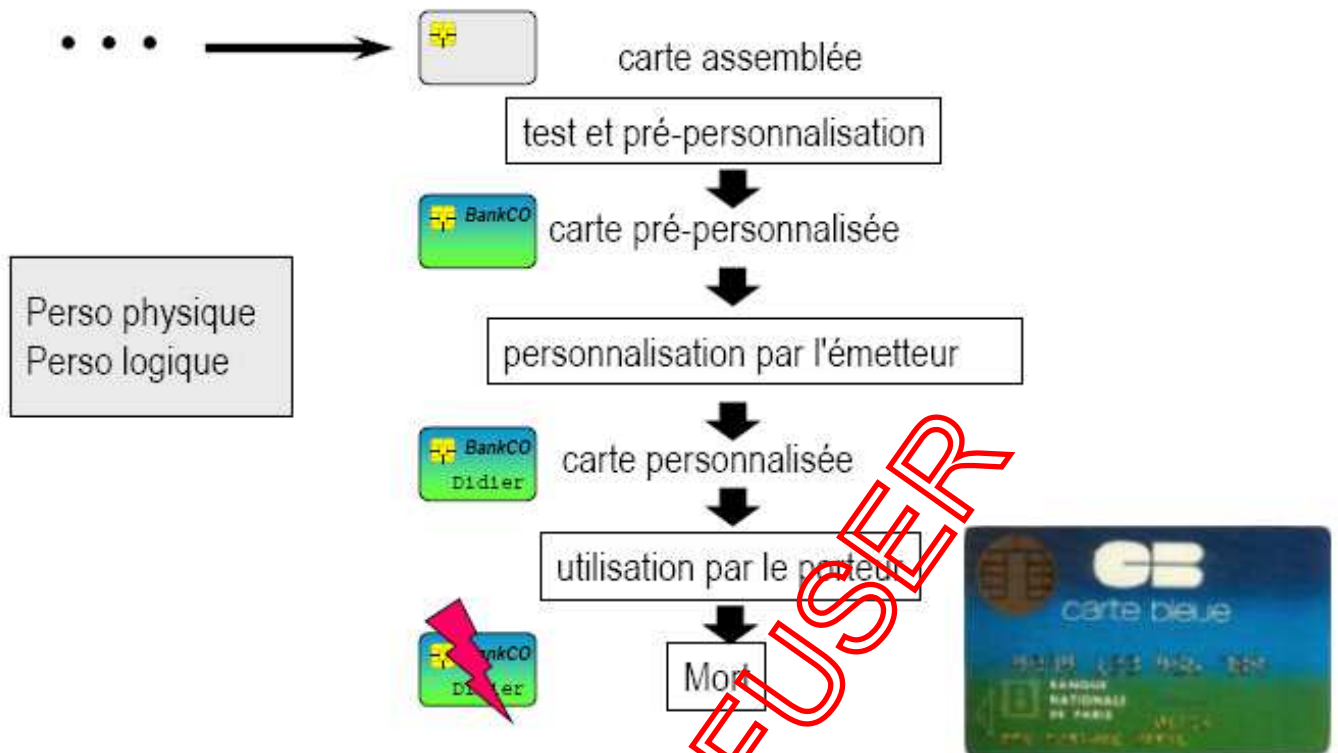


Machine pour la fabrication



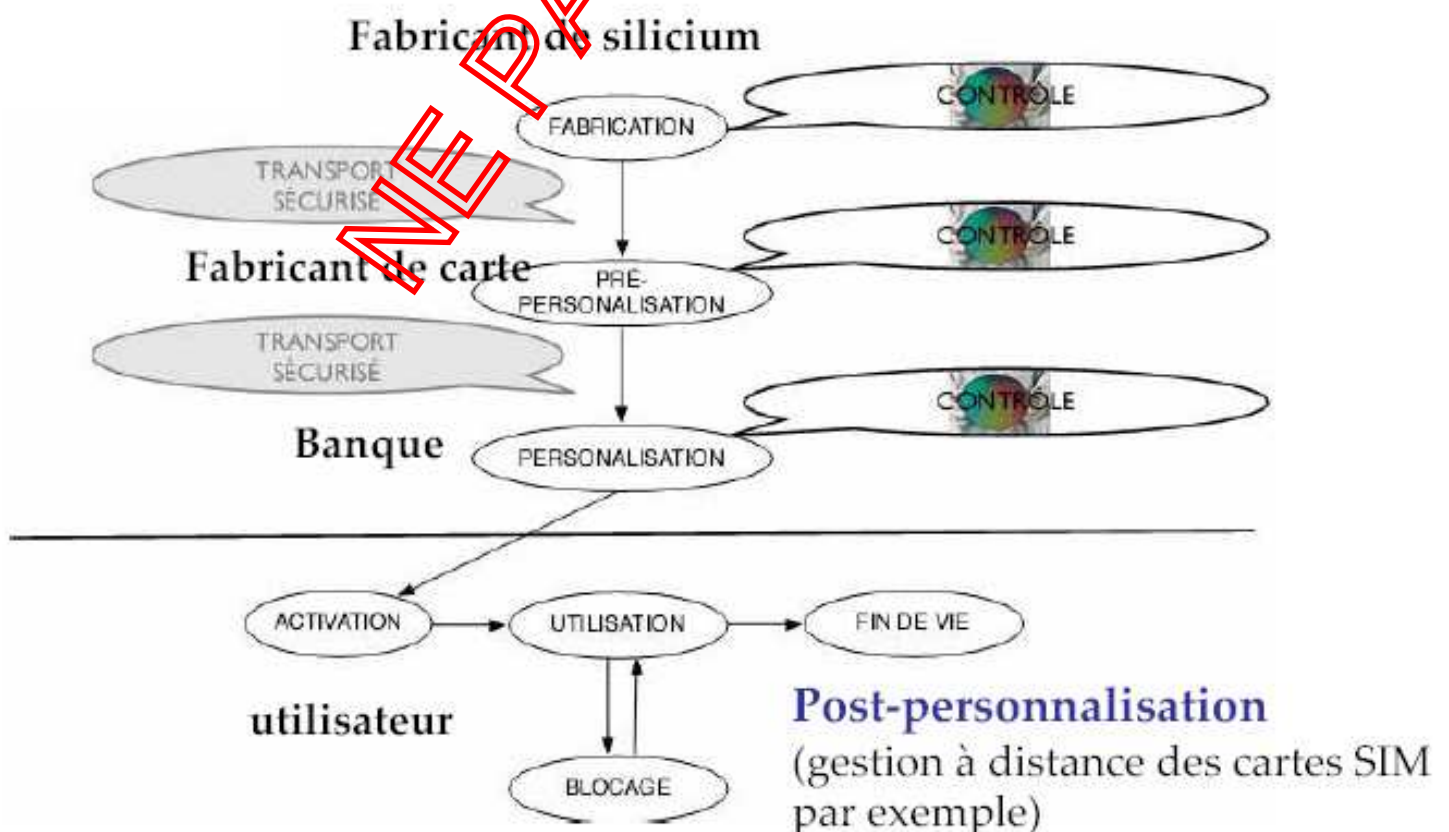
Cartes à Microprocesseur, 1997-2008





97

Cycle de vie (exemple de la carte bancaire)



Phases de Dev. D'une appli carte

- **Choix de la carte :**
 - Puissance du processeur
 - Types et tailles des mémoires
 - Respect (ou non) des normes
- **Choix du masque**
 - Caractéristiques des « ROM » de la carte
 - OS + fonctions applicatives
 - Ecriture du masque
 - directement en assembleur
 - Dev. D'une application carte spécifique
 - Utilisation d'un masque « générique » + *firmwares*
- **Initialisation+personnalisation de la carte**
 - Chargement des fonctions spécifiques de l'application, de l'émetteur, des utilisateurs et du porteur
- **Environnement de l'application**

9

Systèmes d'exploitation carte

- **Masque dédié : une seule application figée**
 - B0 (CB, Seagat Vitale, CPS)
 - Monéo
- **Mono Application**
 - ISO 7816-4

- **Multi Applications**
 - *BasicCard*
 - Motos
 - JavaCard
 - .NET SmartCard
- **Masque dédié : une seule application figée**
 - B0 (CB, Seagat Vitale, CPS)
 - Monéo
- **Mono Application**
 - ISO 7816-4
- **Multi Applications**
 - *BasicCard*
 - Motos
 - JavaCard
 - .NET SmartCard

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



Quelques acteurs

Fondeur: Atmel, Infineon, NXP, Samsung, ST Microelectronics,

Encateur: Gemalto, Giesecke&Devrient, Oberthur Card Systems, Orga

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

Evaluation/Certification: Silicomp, Thales, CEA Leti, SERMA Technologies, SGDN -- ANSSI

Laboratoire: LaBRI, XLIM, INRIA, CNRS, CNAM, ...

Les Acteurs (2006)

- Bull CP8 (racheté par Schlumberger)
- Schlumberger
- Axalto (filiale mise sur le marché par Schlumberger) (#2)
- Gemplus (#1)
- Gemalto (Mariage Gemplus+Axalto) 7 Décembre 2005
 - 1,8 G€ de CA, 11000 employés, ~50 pays, 22 usines
 - 55% du marché mondial des cartes SIM avec plus de 600 millions de cartes livrées sur un total estimé de 1,22 milliards de SIM pour 2005. La nouvelle entité sera aussi un poids lourd sur le marché des services financiers et sur celui des lecteurs. Elle ne devrait toutefois contrôler "que" 16% du marché des cartes de sécurité
- Oberthur Card Systems
- Delarue
- Giesecke & Devrient (#3)
- Sagem Orga (#4)
 - 1,800 employés
- ...

NE PAS DIFFUSER

La carte à puce

—
Sa sécurité

La sécurité physique

techniques d'impression sophistiquées

- différentes couches d'impression
- hologramme
- Embossage

procédés d'encollage

matériaux plastiques

design du micro-contact

125

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

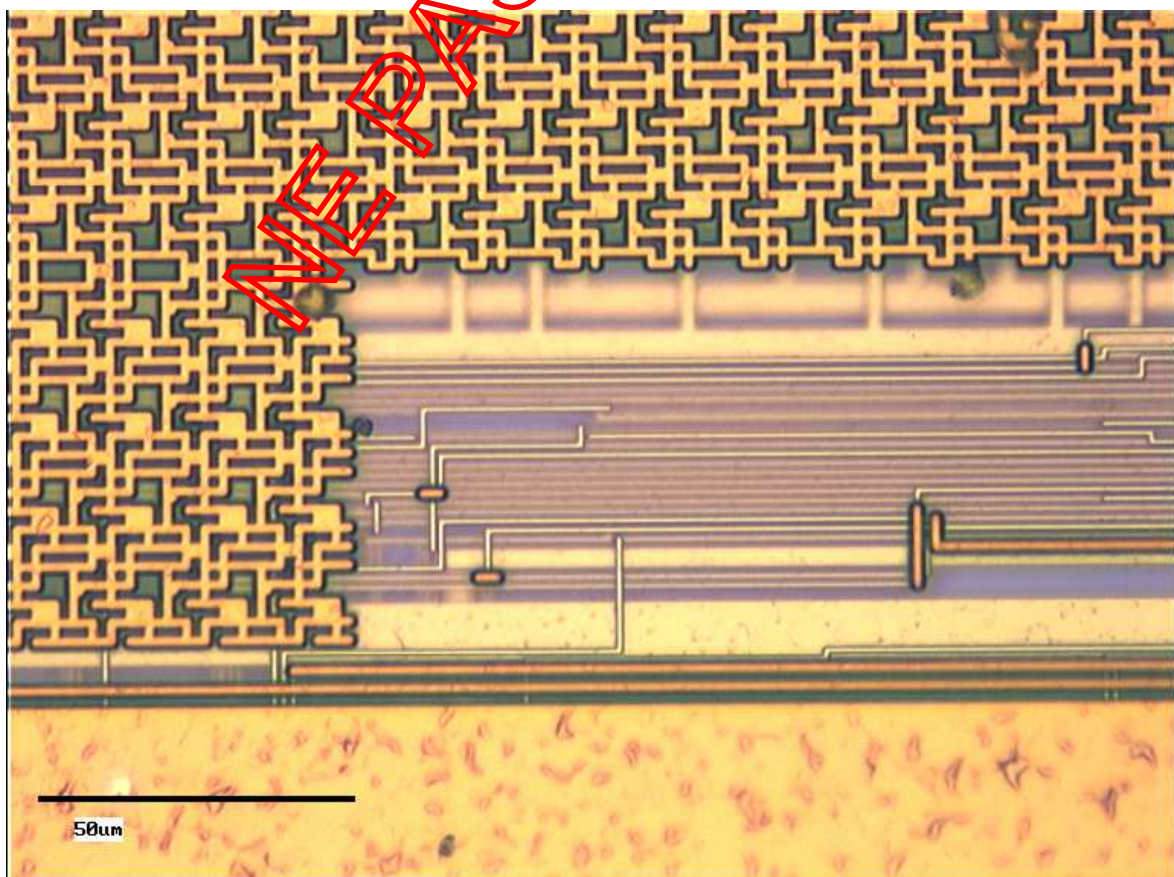
126

Une puce



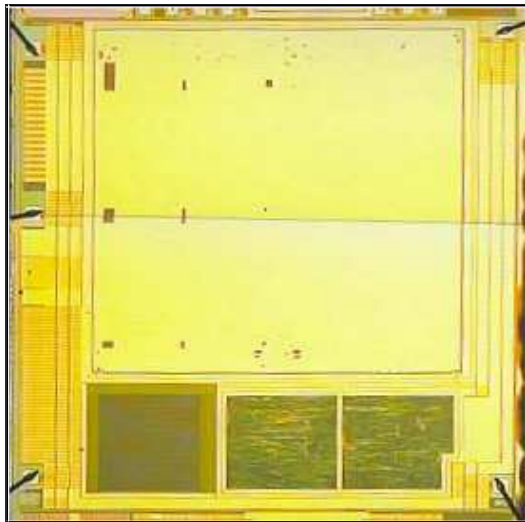
127

La grille de protection

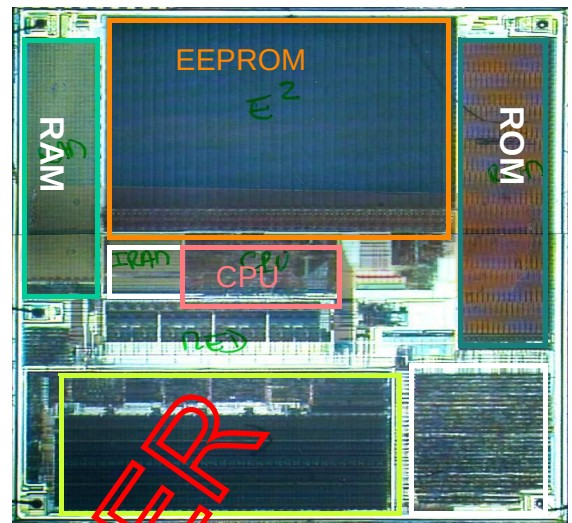


128

LESSON LEARNED: SHIELD IT!

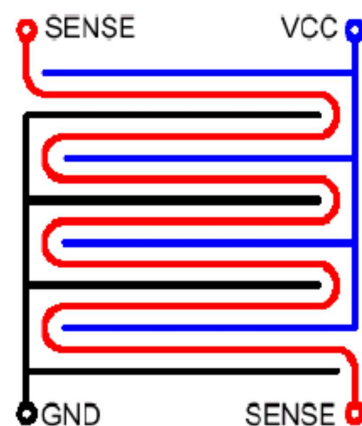


- Shield
- Scrambled glue logic
- No Buses visible

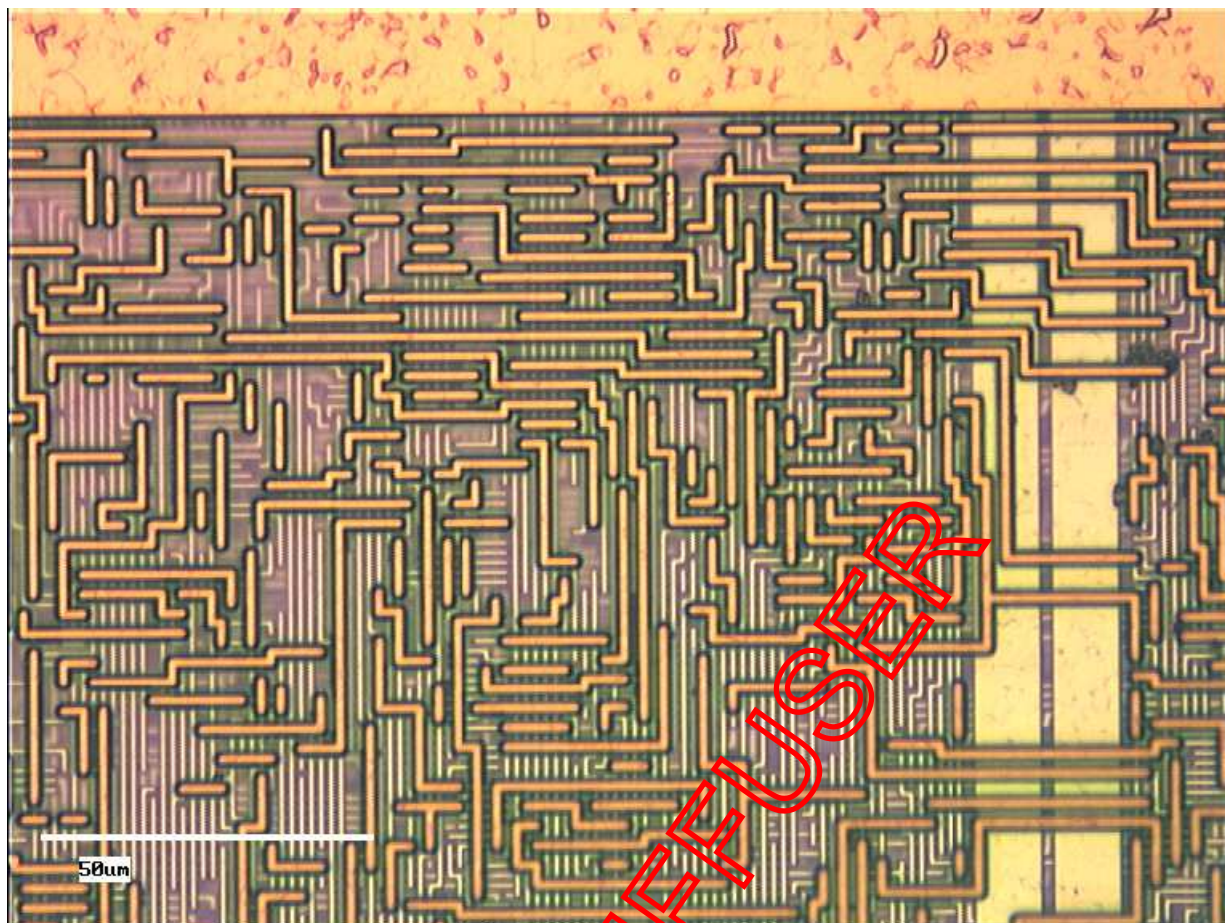


- Blocks easily identified
- No shield
- No glue logic
- Buses clearly visible

Grille active

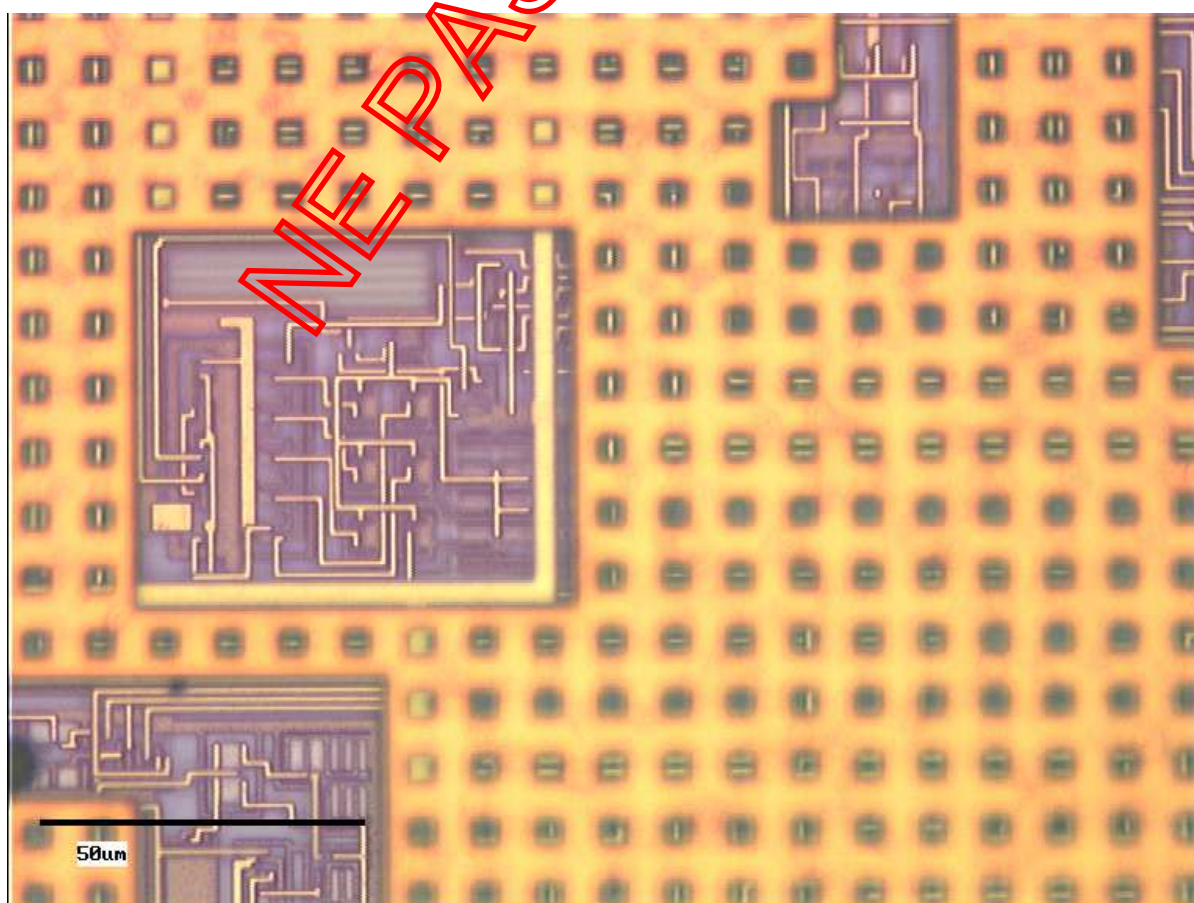


Le CPU



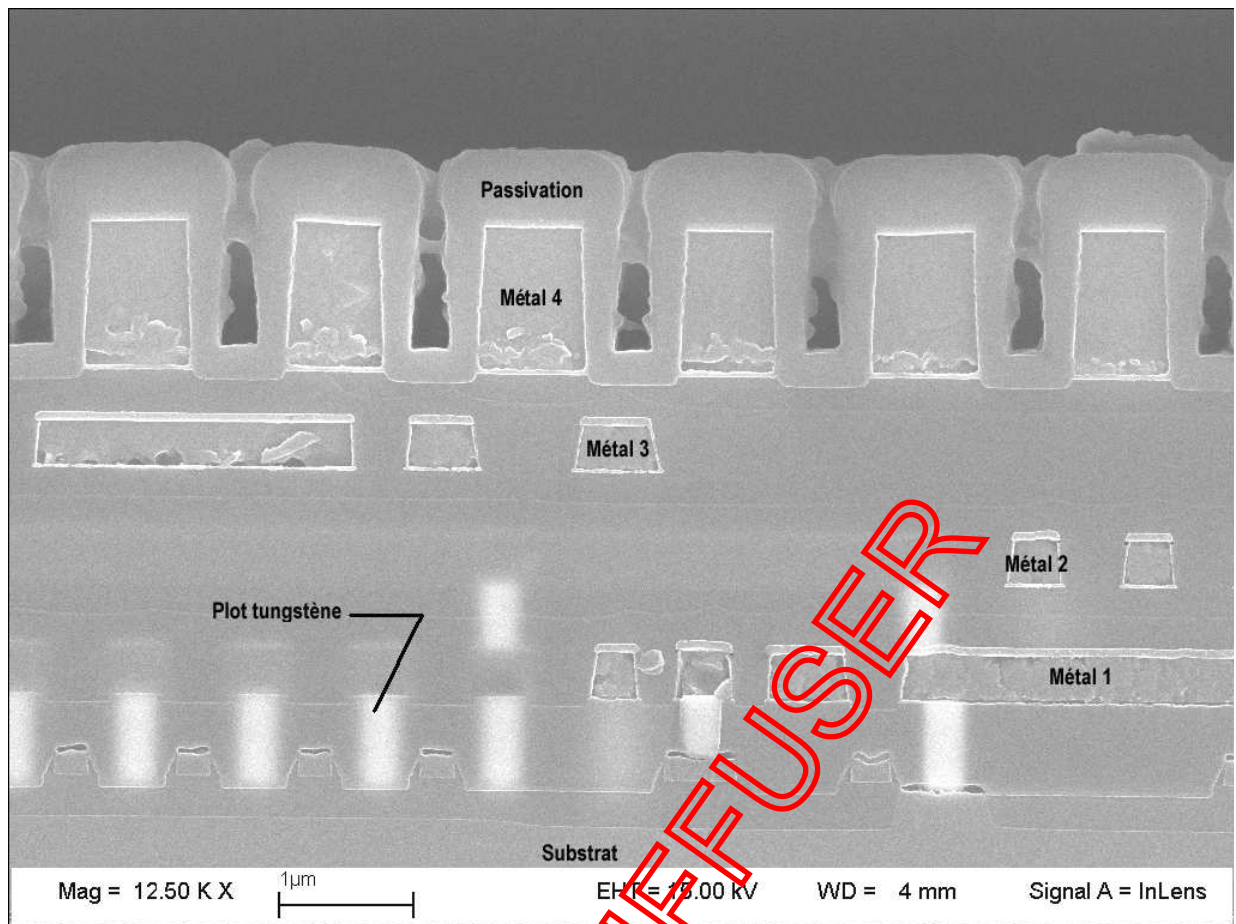
131

La partie analogique



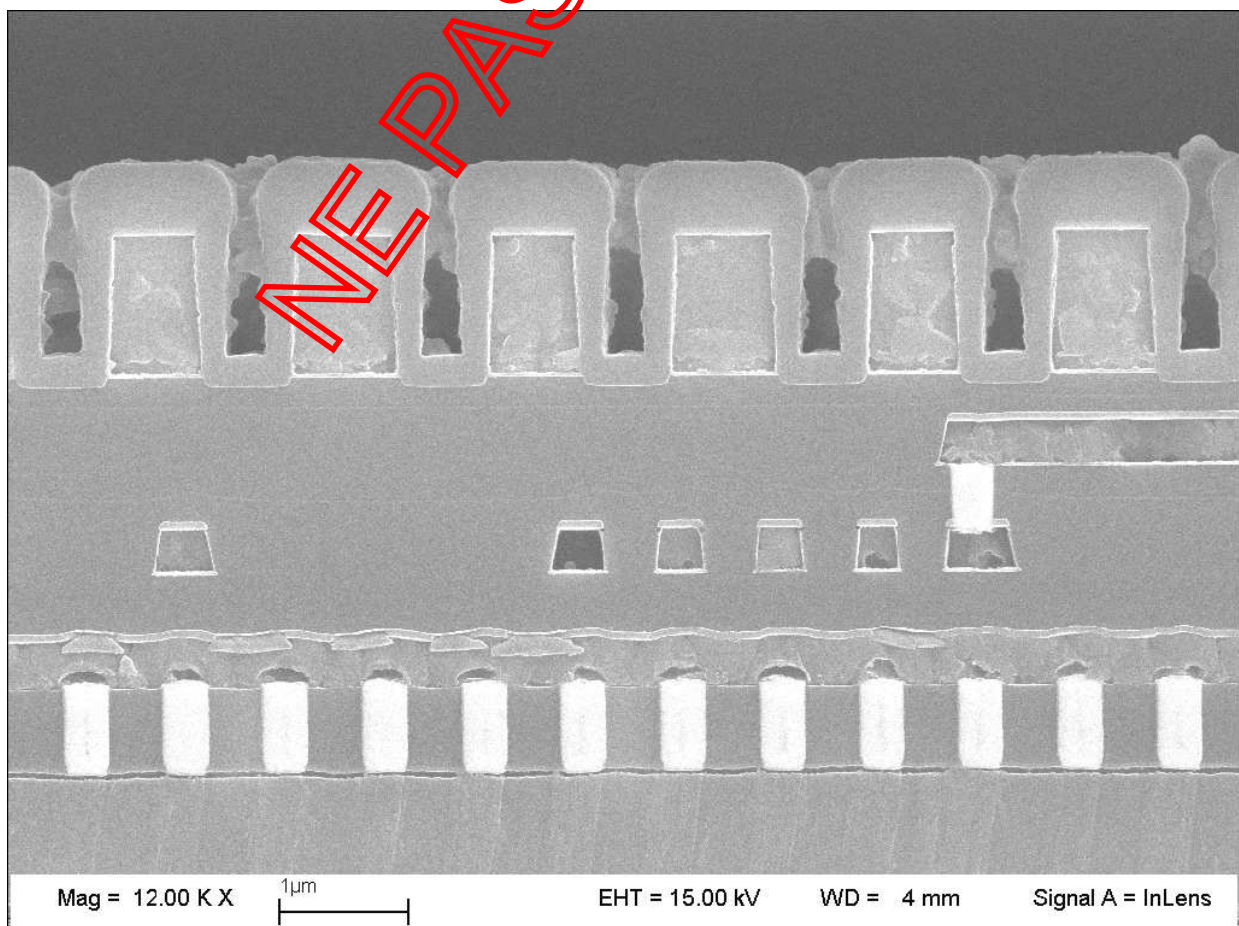
132

La coupe d'une puce



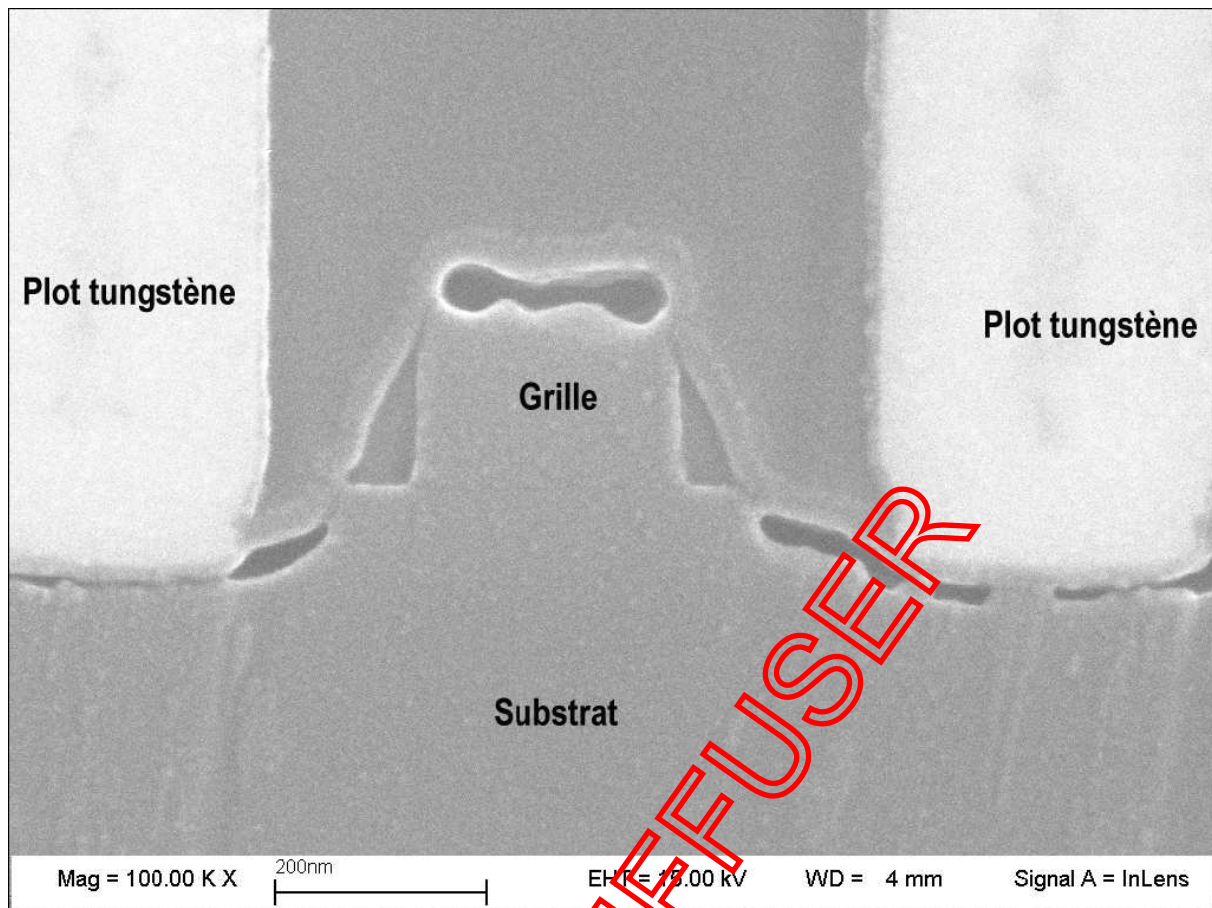
133

La coupe d'une puce



134

La coupe d'un transistor



135

La sécurité software

contrôles d'accès aux données

maintien de l'intégrité des données

entrées/sorties sécurisées

migration du code

136

La sécurité de l'env. de production

physiquement sécurisé

régulièrement audité

137

La sécurité

Pour résumer :

NE PAS DIFFUSER

LE PÉRIPHÉRIQUE
LE PLUS SÉCURISÉ
AU MONDE



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

138

Les méthodes formelles et le test

Conception formelle de certains parties de l'OS (voire de l'OS complet)

Preuve formelle sur des aspects sécuritaire

Génération de jeux de tests associés aux spécifications

...

139

L'évaluation sécuritaire et la certification

La certification

Objectifs :

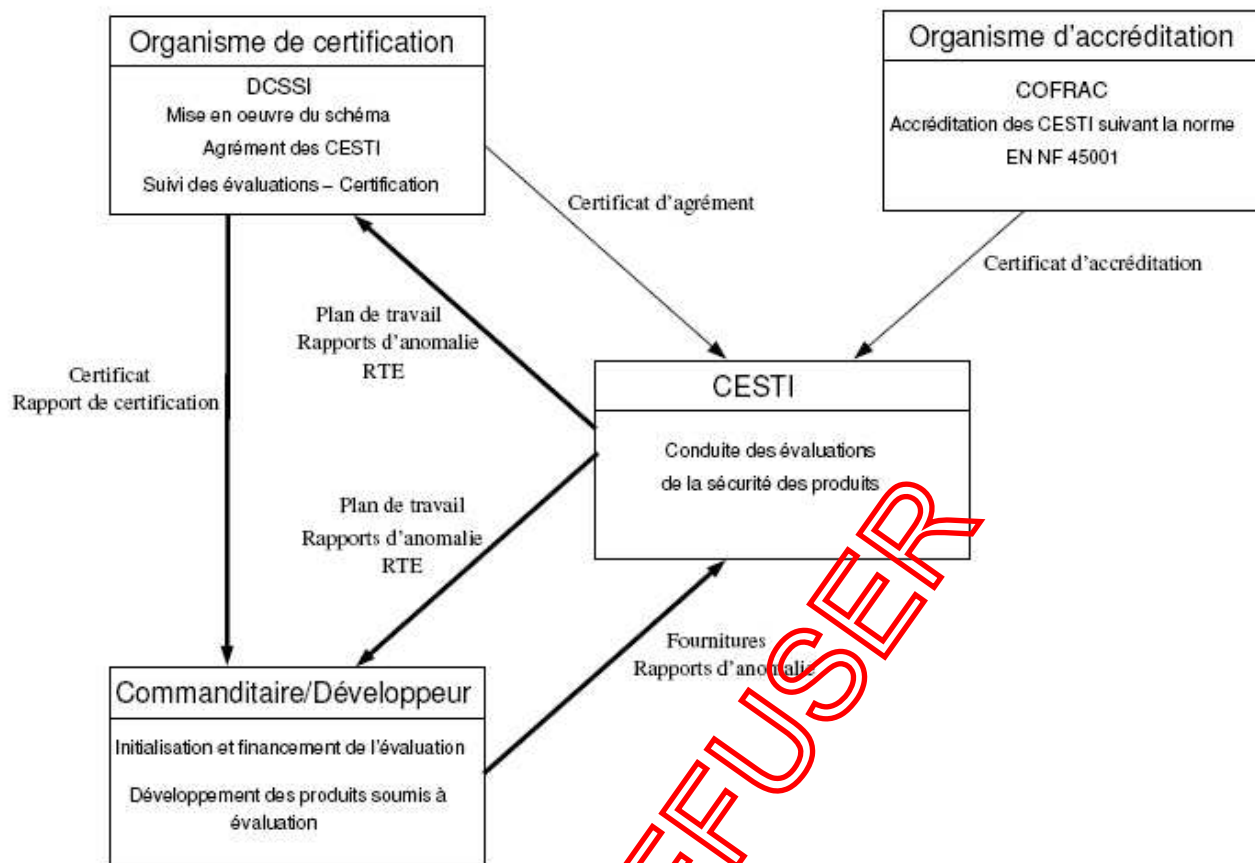
- Garantir le niveau de sécurité d'un produit TI (confidentialité, intégrité, disponibilité).
- Bénéficier des accords de reconnaissance mutuelle.

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 !

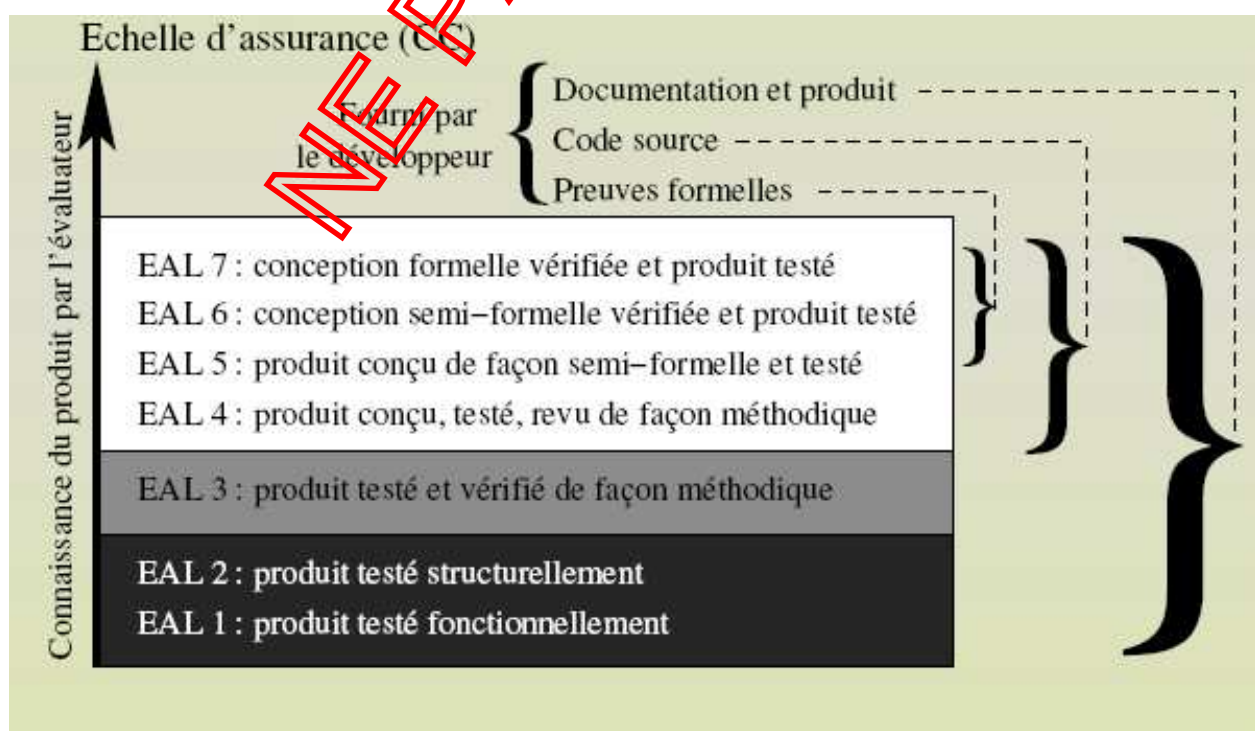
140

Le schéma Français



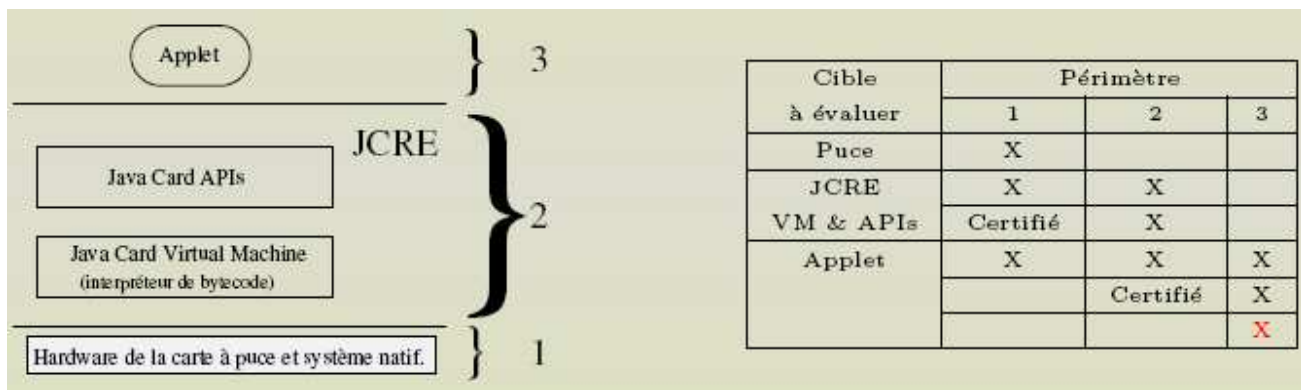
1

L'échelle d'assurance



Le périmètre d'évaluation

Défini dans la cible de sécurité dans la description de la cible d'évaluation.



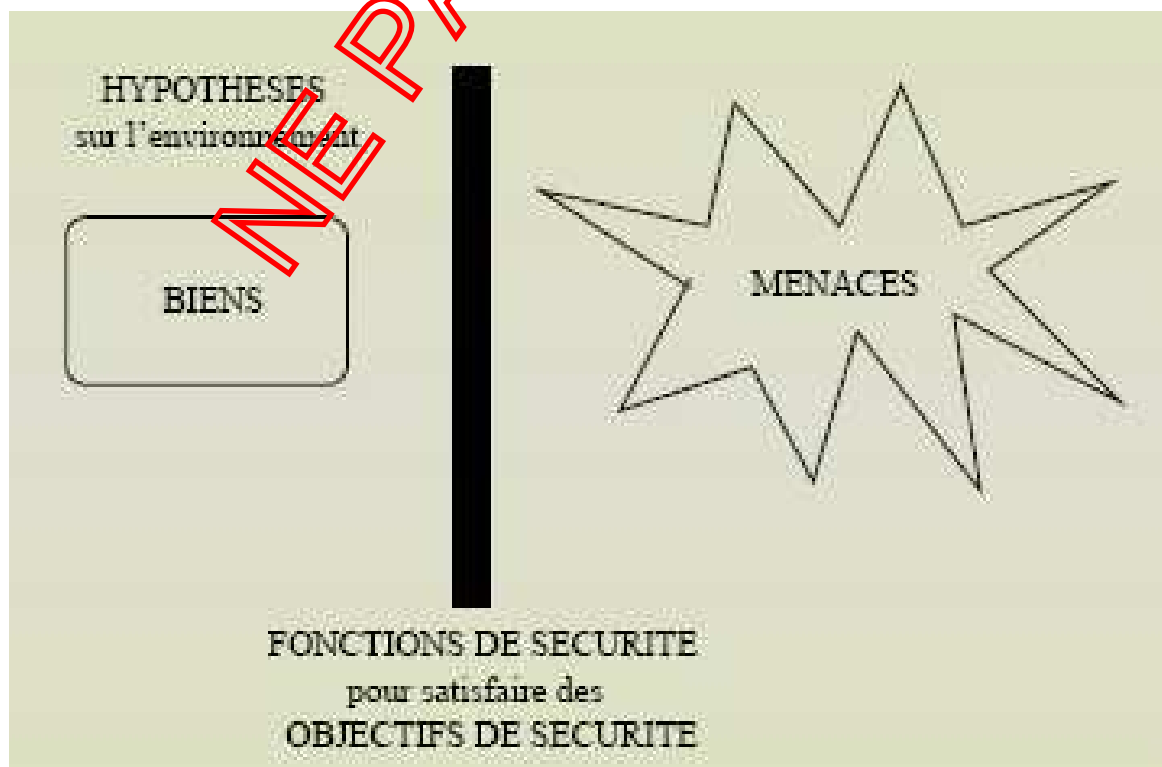
Nouveaux problèmes apportés par les applets Java Card :

- pas de définition précise du périmètre d'évaluation ;
- pas de méthodologie d'évaluation d'une applet ;
- la multi-application.

143

Les bases des critères communs

Critères Communs (ISO15408)



144

La carte à puce

Les attaques

145

Différentes catégories

Non invasives

Le produit reste fonctionnel

Invasives

Le produit est partiellement endommagé

146

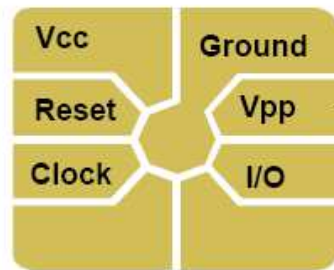
Non invasives

modification des conditions opérationnelles (Vcc, F)

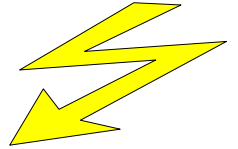
modification de la température

modification des rayonnements lumineux (UV, rayon X, lumière blanche, IR, ...)

injection de fautes (glitches, rayonnements lumineux)



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



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

attaques sur les canaux cachés

Temps

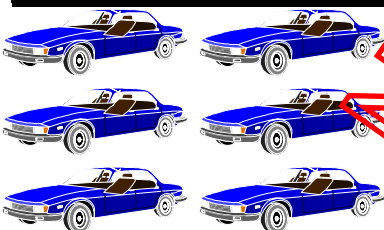
Consommation électrique

Émissions électromagnétiques

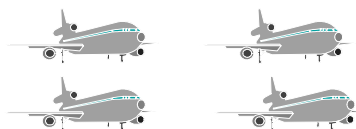
147

Êtes vous prêt ?

Politique : Les jouets cassés ne sont pas payés

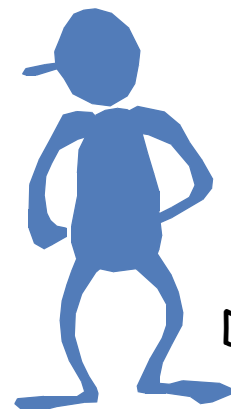
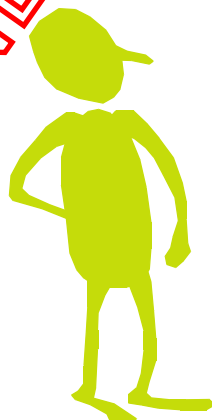


voiture = \$3



avion = \$5

Jack

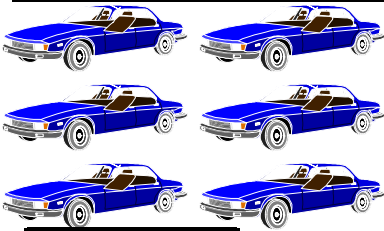


Dino

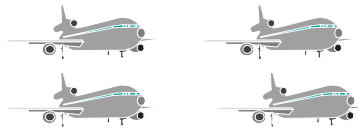
Dino achète des jouets à Jack

Êtes vous prêt ?

Politique : Les jouets cassés ne sont pas payés

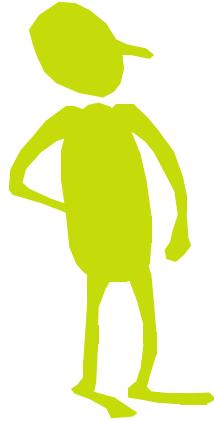


voiture = \$3

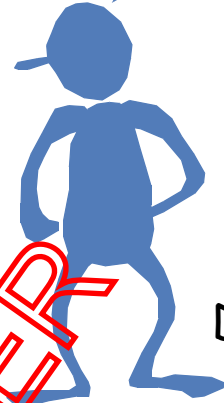


avion = \$5

Jack



Je voudrais acheter
3 avions

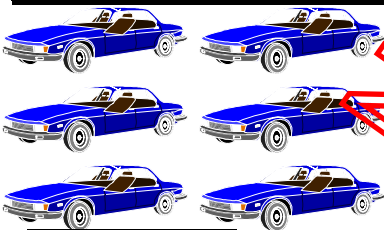


Dino

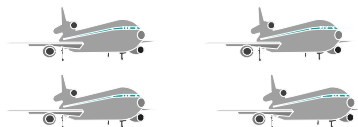
Dino achète des jouets à Jack

Êtes vous prêt ?

Politique : Les jouets cassés ne sont pas payés

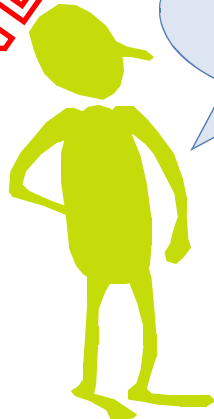


voiture = \$3



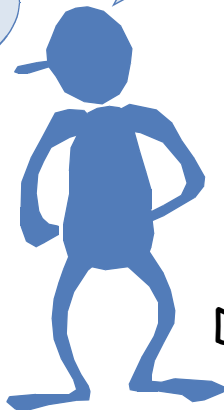
avion = \$5

Jack



Ça fera \$15

Je voudrais acheter
3 avions

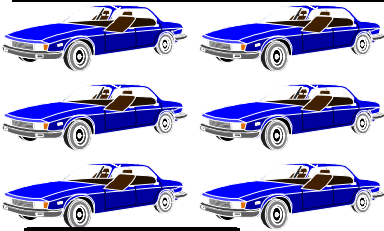


Dino

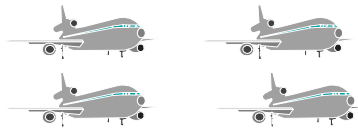
Dino achète des jouets à Jack

Êtes vous prêt ?

Politique : Les jouets cassés ne sont pas payés



voiture = \$3



avion = \$5

Jack

Ça fera \$15

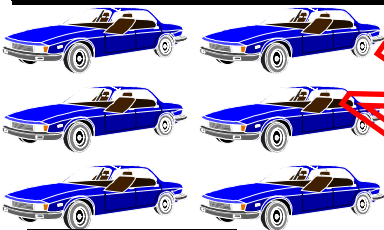
OK, s'il vous plaît envoyez le par DHL

Dino

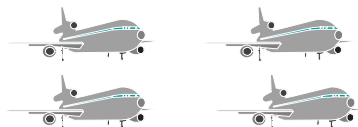
Dino achète des jouets à Jack

Êtes vous prêt ?

Politique : Les jouets cassés ne sont pas payés



voiture = \$3



avion = \$5

Jack

Comment payez vous ?

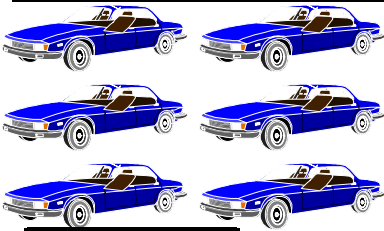
OK, s'il vous plaît envoyez le par DHL

Dino

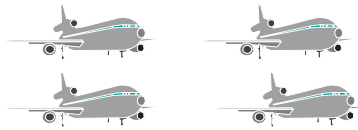
Dino achète des jouets à Jack

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

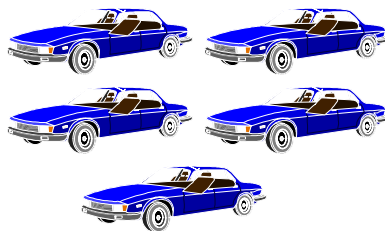
Dino

Dino achète des jouets à Jack

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

Qu'est ce qu'il a
acheté pour \$15 ?

Dino

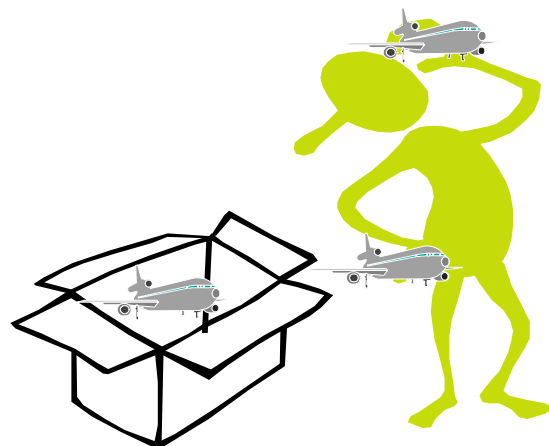


Méchant facteur

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

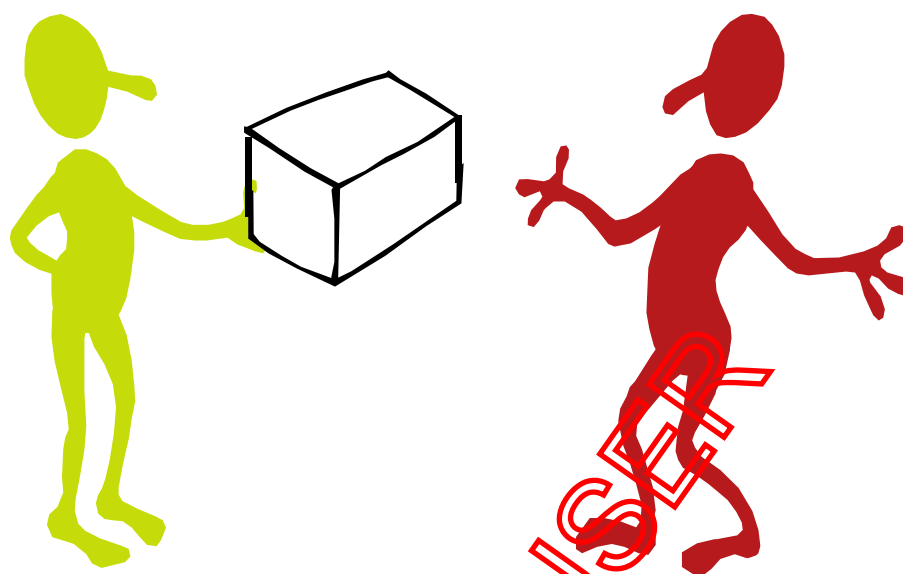


*Pendant ce temps, Jack prépare le
colis DHL*



NE PAS DIFFUSER

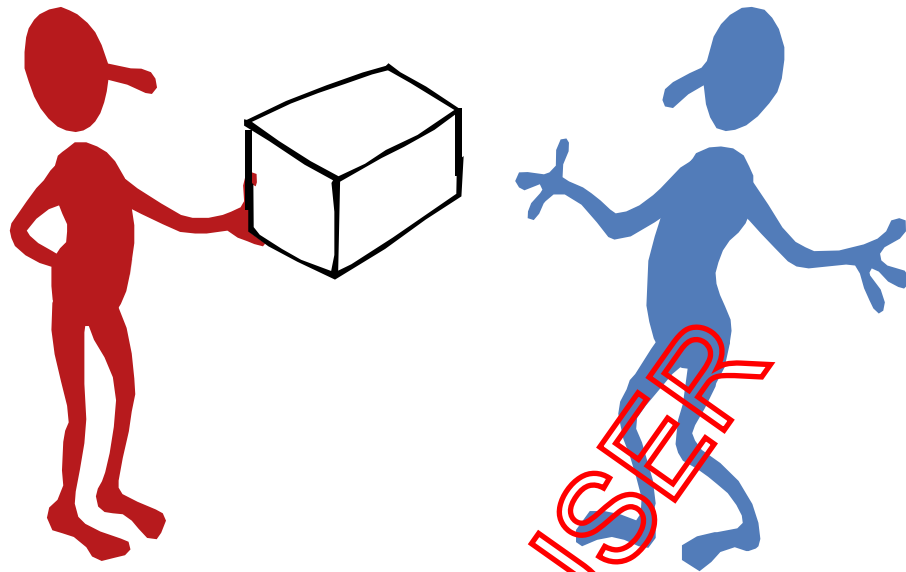
Et le donne au facteur



*Qui tape dedans assez fort pour
casser un jouet*



Et le donne à Dino



*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)

Attaques software/hardware

Pour forger des pointeurs et ensuite ...



<http://www.cs.princeton.edu/~sudhakar/papers/memerr-slashdot-commentary.html>

161

Les canaux cachés

le temps d'exécution

=> nombre de cycle 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 matériel

un oscilloscope numérique,

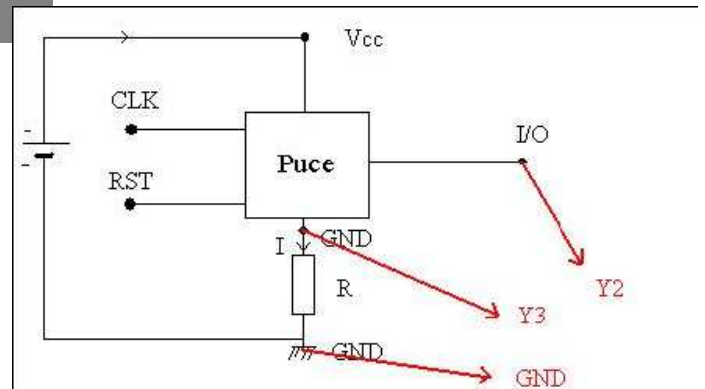
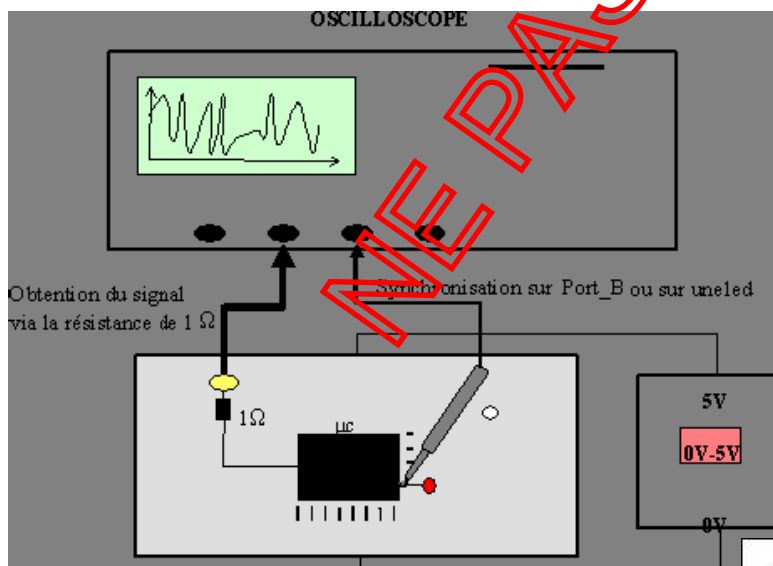
un lecteur de carte à puce,

un pc équipé de cartes d'acquisition et de logiciels mathématique pour le traitement des données,

une sonde CEM si on veut étudier les émissions électromagnétiques.



Le montage pour suivre la consommation en courant



La “timing attack”

Cette attaque consiste à mesurer le temps d'exécution d'un algorithme.

=> Révèle des informations sur les opérations et/ou les opérandes.

Nécessite souvent :

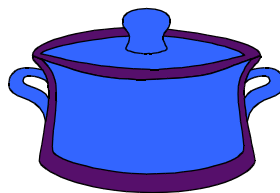
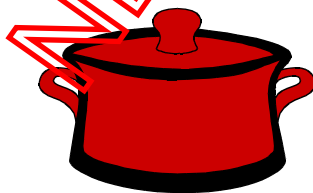
un grand nombre d'exécution à messages choisis,
un traitement statistique des résultats obtenus.

Exemple:

165

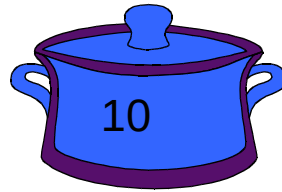
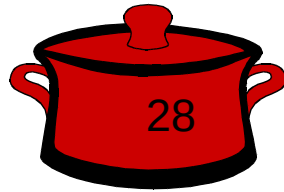
Un jeu

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



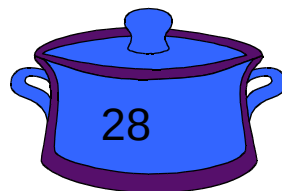
Un jeu

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



Un jeu

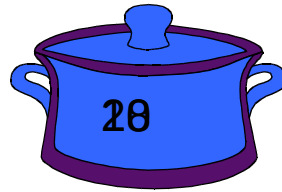
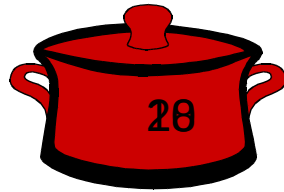
- Mettez 28 dans un des pots et 10 dans l'autre :



NE PAS DIFFUSER

Un jeu

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



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

Est-ce que ce jeu à un sens ?

NE PAS DIFFUSER

Est-ce que ce jeu à un sens ?

- Et bien, normalement non :

$$28 \times 7 + 10 \times 10 = 296 \quad \text{est un nombre pair}$$

et

$$10 \times 7 + 28 \times 10 = 350 \quad \text{est aussi un nombre pair...}$$

- 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 quelle valeur était dans quel pot !

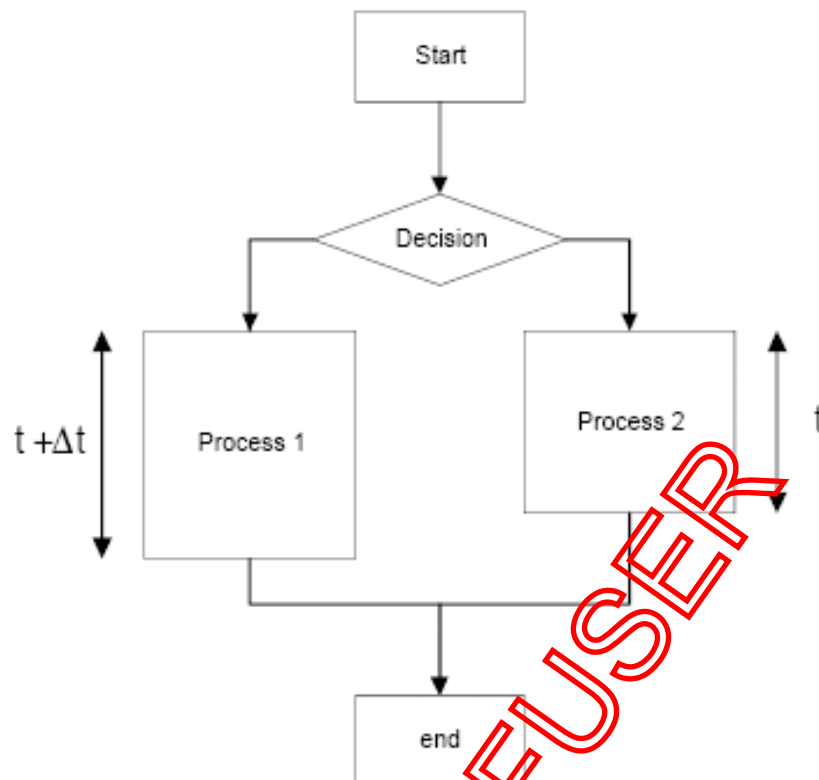
Conclusion

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.**

La "timing attack"

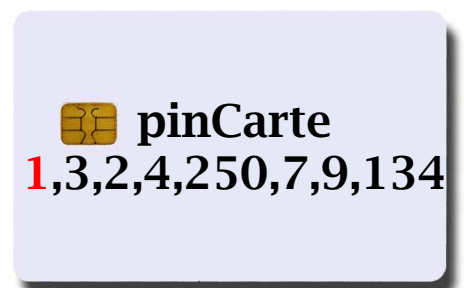


173

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

```

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



Présentons un PIN quelconque et déroulons le programme :

t cas du premier octet FAUX

```

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

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

174

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. pinPresente

Calculer T[n0] le maximum des T

– T[n0] = max(T[n]); n = 0; ... ; 255

n0 est la solution pour pinCarte[0]

Itérer sur tous les pinPresente[i]

Nombre d'essais : $8 * 256 = 2048$ (contre 256^8 en force brute)

On peut même s'arrêter avant $n=255$!

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.

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 ;
```

équivalent à :

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

Et pas test = test && (pinCarte [i] == pinPresente [i]) ;

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

177

Et sinon vous savez codé un vérification
de PIN sécurisé maintenant ?

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

```
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 ;
}
```

```

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 ;
    }
}

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 {

...

```

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

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

•Correct?

```
// 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 ;
```

```

boolean verify (byte[] buffer, short ofs, byte len) {
    byte tl = triesLeft ;
    if (tl != (short)(~triesLeftBackup)) takeCountermeasure() ;
    if (tl < 0) return false;
    JCSystemSystem.beginTransaction() ;
        triesLeft = --tl ;
        triesLeftBackup++ ;
    JCSystemSystem.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) {
        JCSystemSystem.beginTransaction() ;
            triesLeft = maxTries ;
            triesLeftBackup = (byte)(~maxTries) ;
        JCSystemSystem.commitTransaction() ;
        // Verifies the new value
        if (triesLeft!=(short)(~triesLeftBackup))
            takeCountermeasure() ;
        authenticated[0] = true ;
        return true ;
    } else {
        authenticated[0] = false ;
        return false ;
    }
}

```

```

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
    JCSystemSystem.beginTransaction() ;
        triesLeft = --tl ;
        stepCounter++ ;
        triesLeftBackup++ ;
    JCSystemSystem.commitTransaction() ;
    stepCounter++ ;
    // Verifies the new value
    if (triesLeft != (short)(~triesLeftBackup)) takeCountermeasure() ;
}

```

```

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
}

```

Other well known rules ...

```

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

```

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 :**

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

La consommation de courant

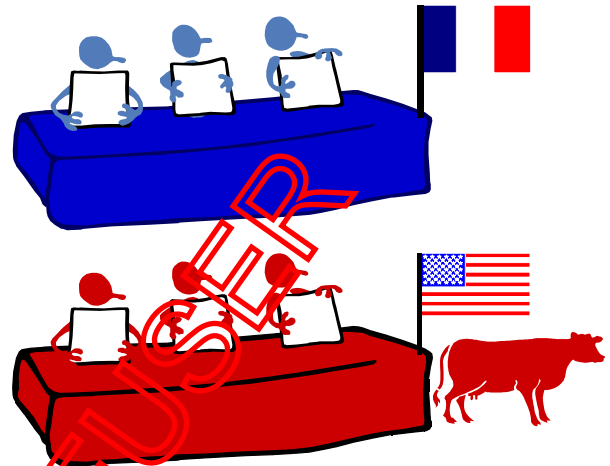
Elle est surtout utilisée dans le domaine de la cryptographie.

Il existe différentes attaques :

- la SPA (Simple Power Analysis)
- la DPA (Differential Power Analysis)
- la HODPA (High Order Differential Power Analysis)

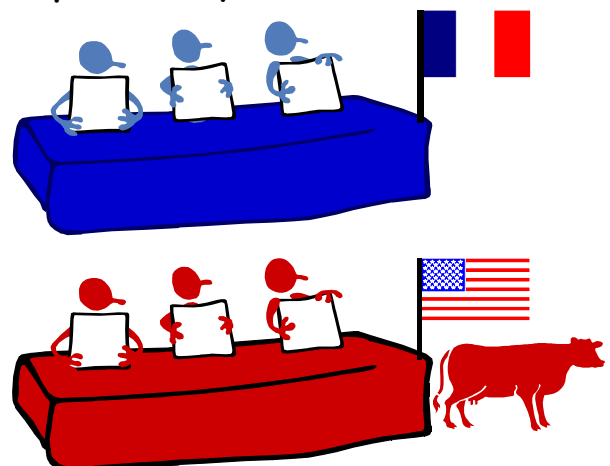
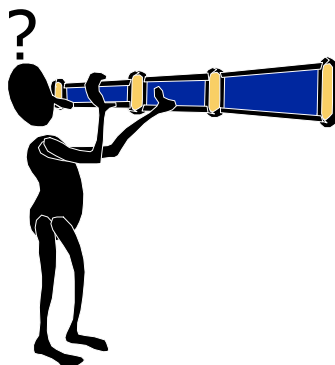
IMPORTATION DU BOEUF ?

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



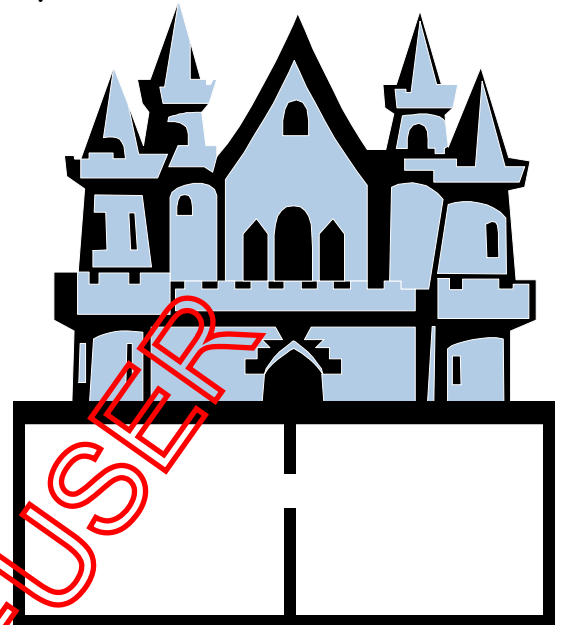
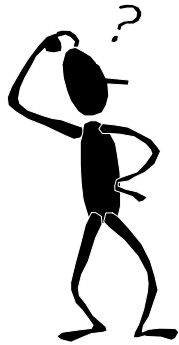
IMPORTATION DU BOEUF ?

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



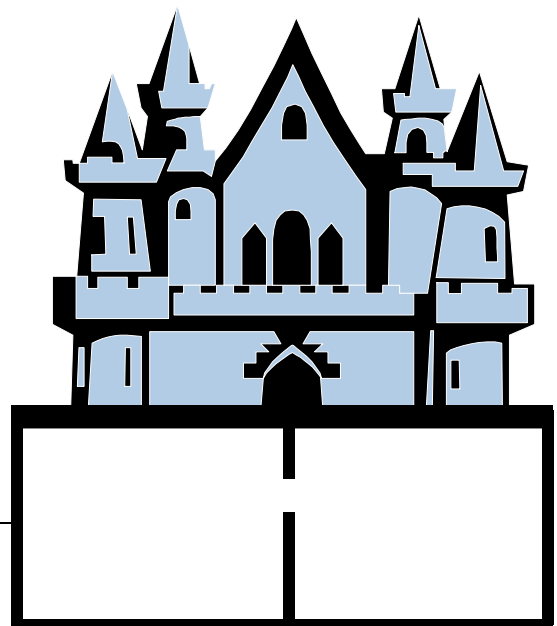
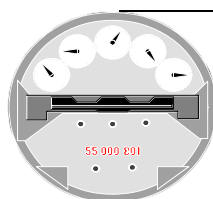
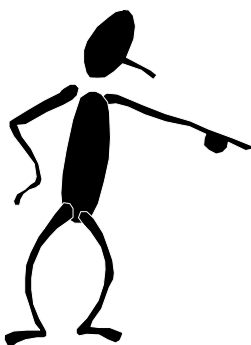
IMPORTATION DU BOEUF ?

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



POWER ATTACKS

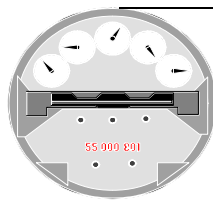
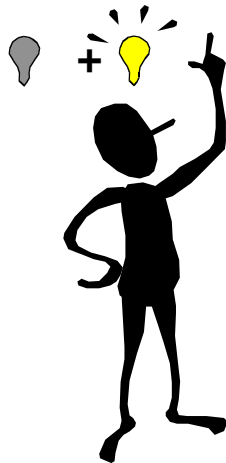
- Idée : Regarder le compteur électrique de l'hôtel!



POWER ATTACKS

- Le disque tourne lentement:

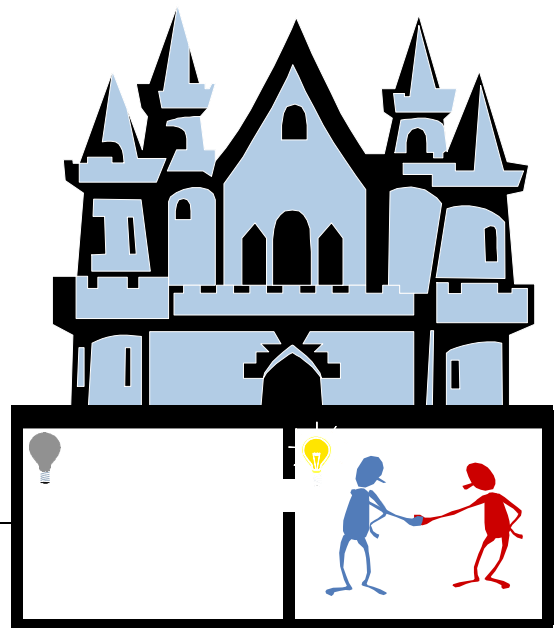
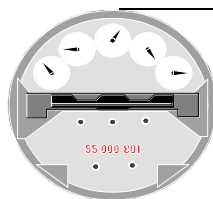
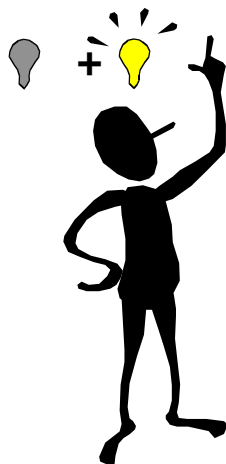
DEAL CONCLUDED



POWER ATTACKS

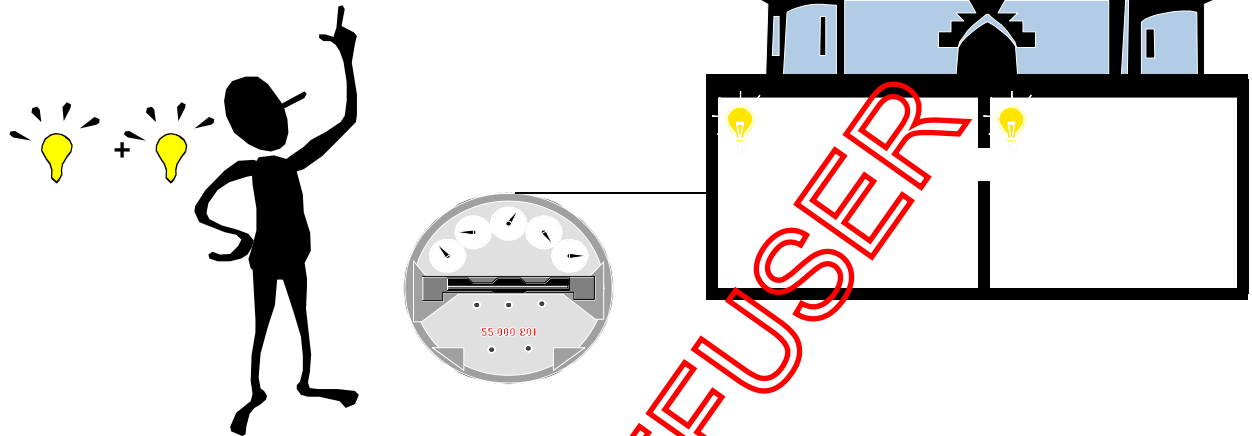
- Le disque tourne lentement:

DEAL CONCLUDED



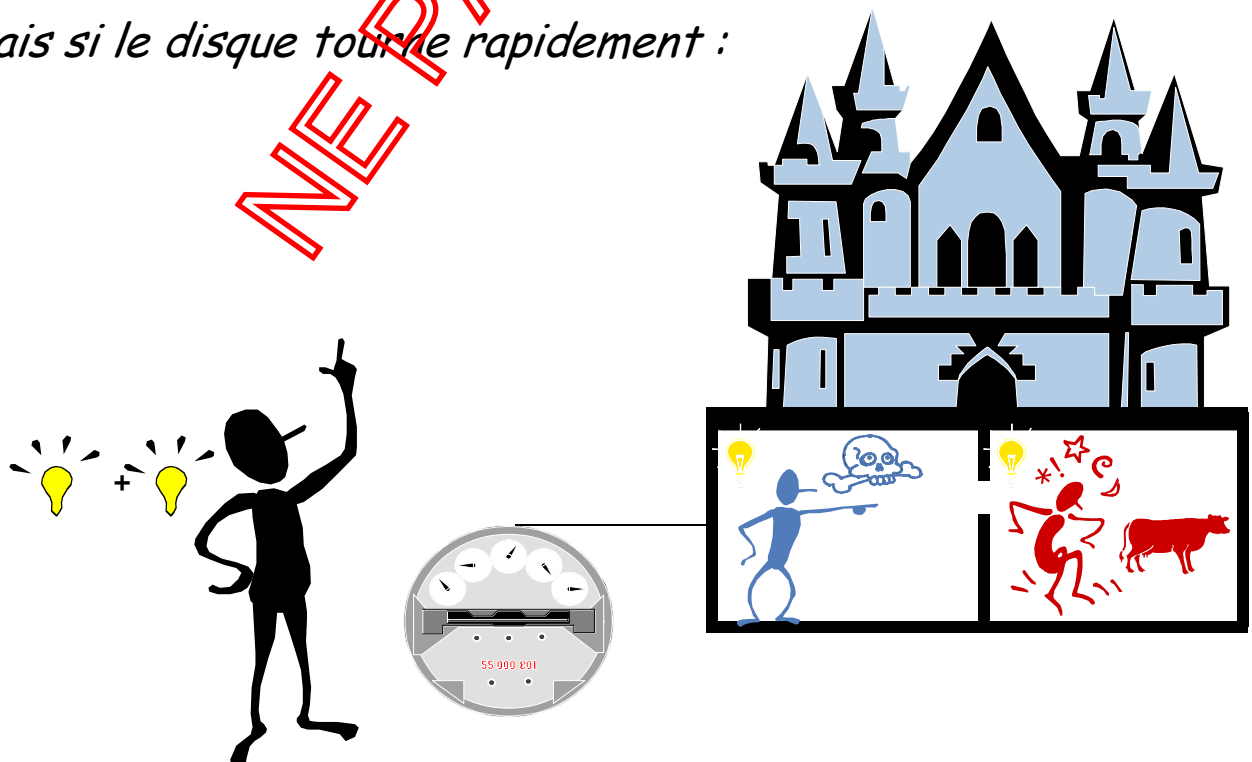
POWER ATTACKS

- Mais si le disque tourne rapidement :



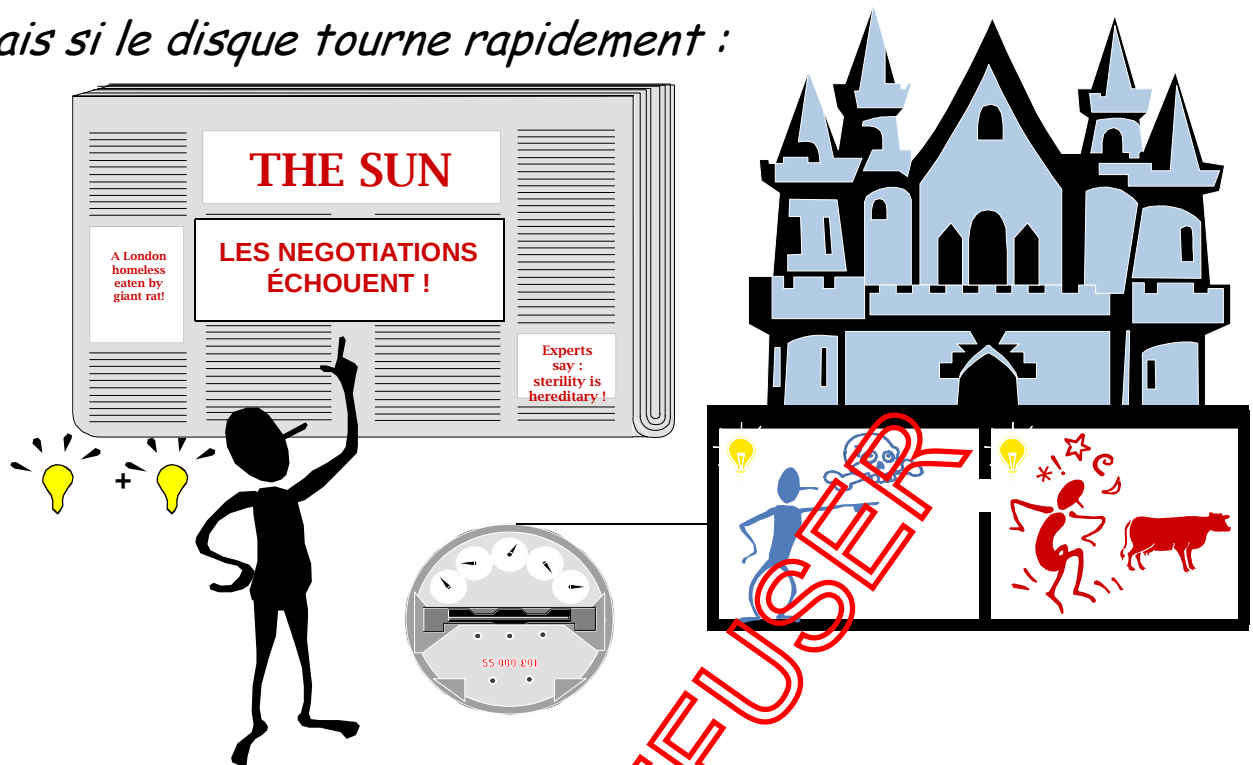
POWER ATTACKS

- Mais si le disque tourne rapidement :



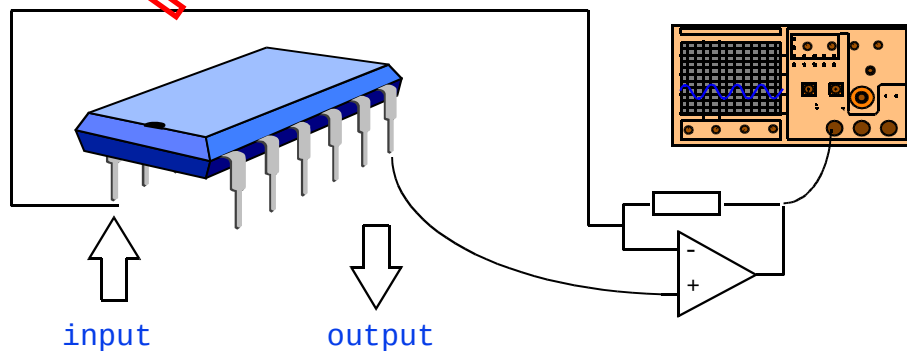
POWER ATTACKS

- Mais si le disque tourne rapidement :



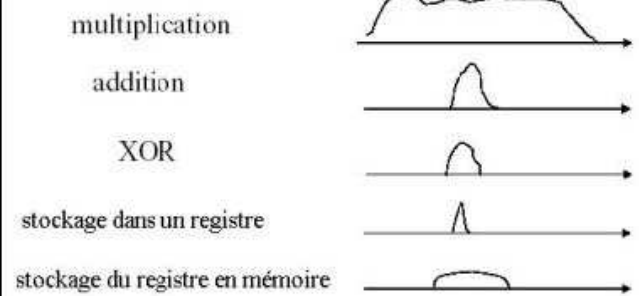
CONCLUSION

The card's current consumption may reveal secret information.

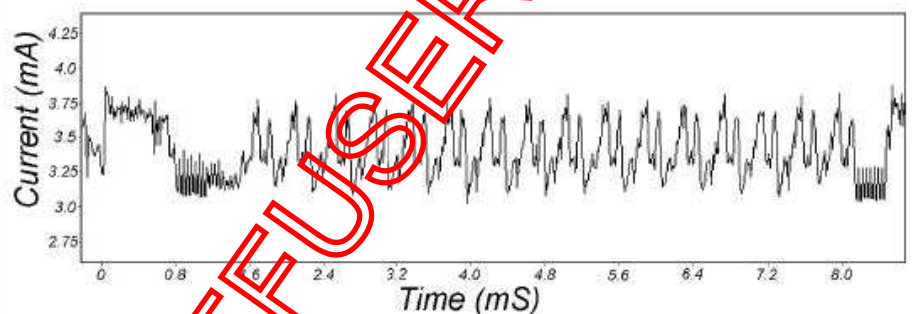


La SPA (Simple Power Analysis)

Principe : Des instructions différentes ont une trace différente.



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

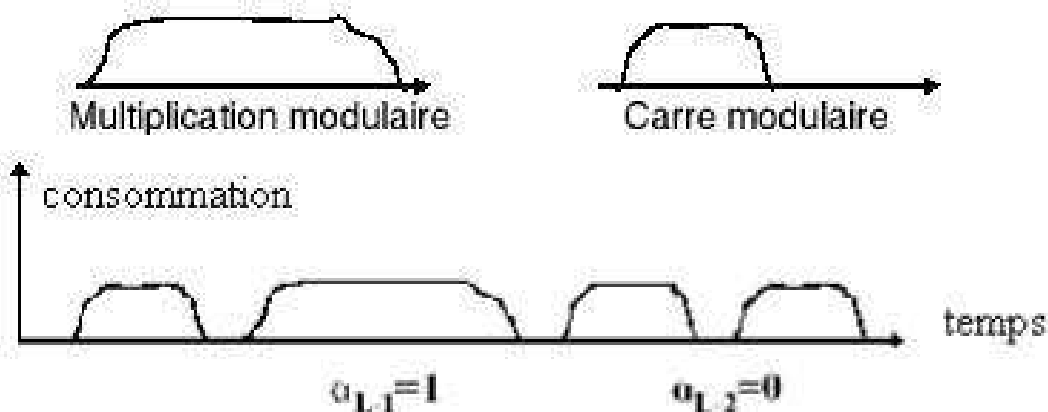


201

La SPA sur la signature RSA

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

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



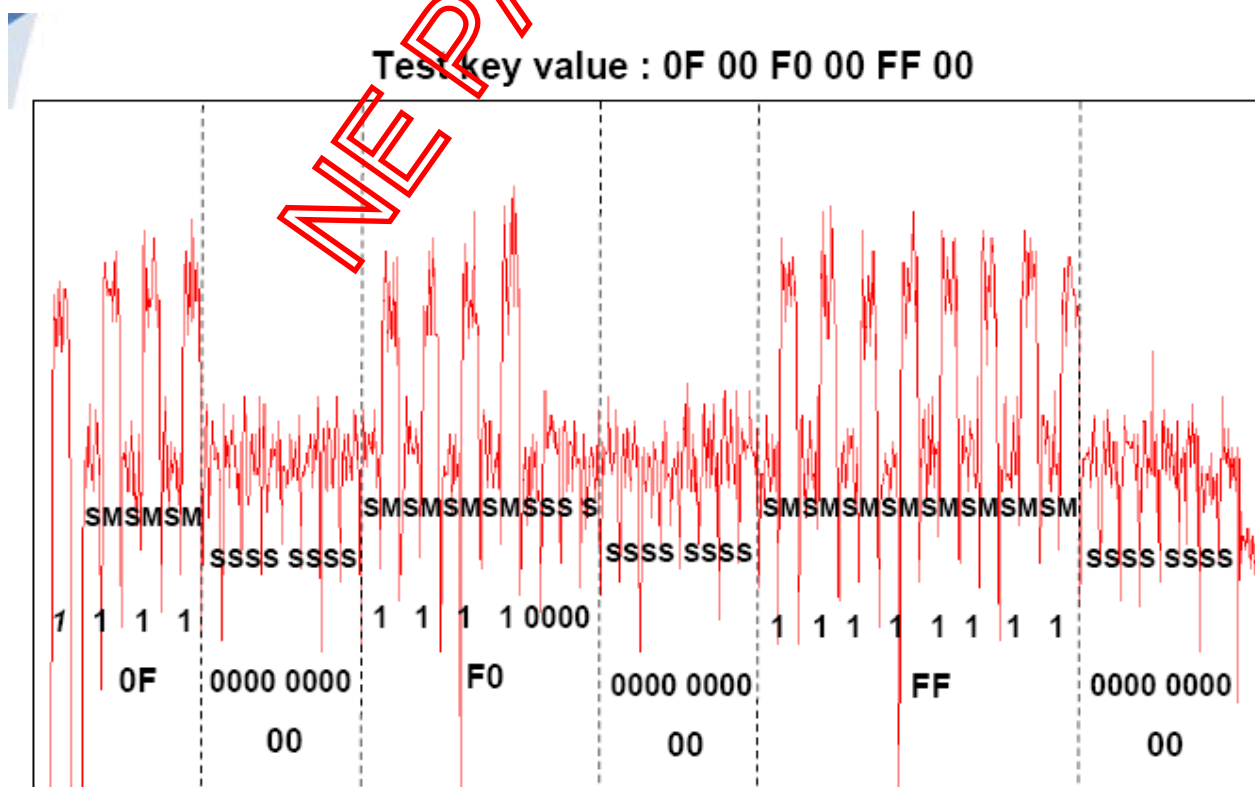
202

Exemple « réel »



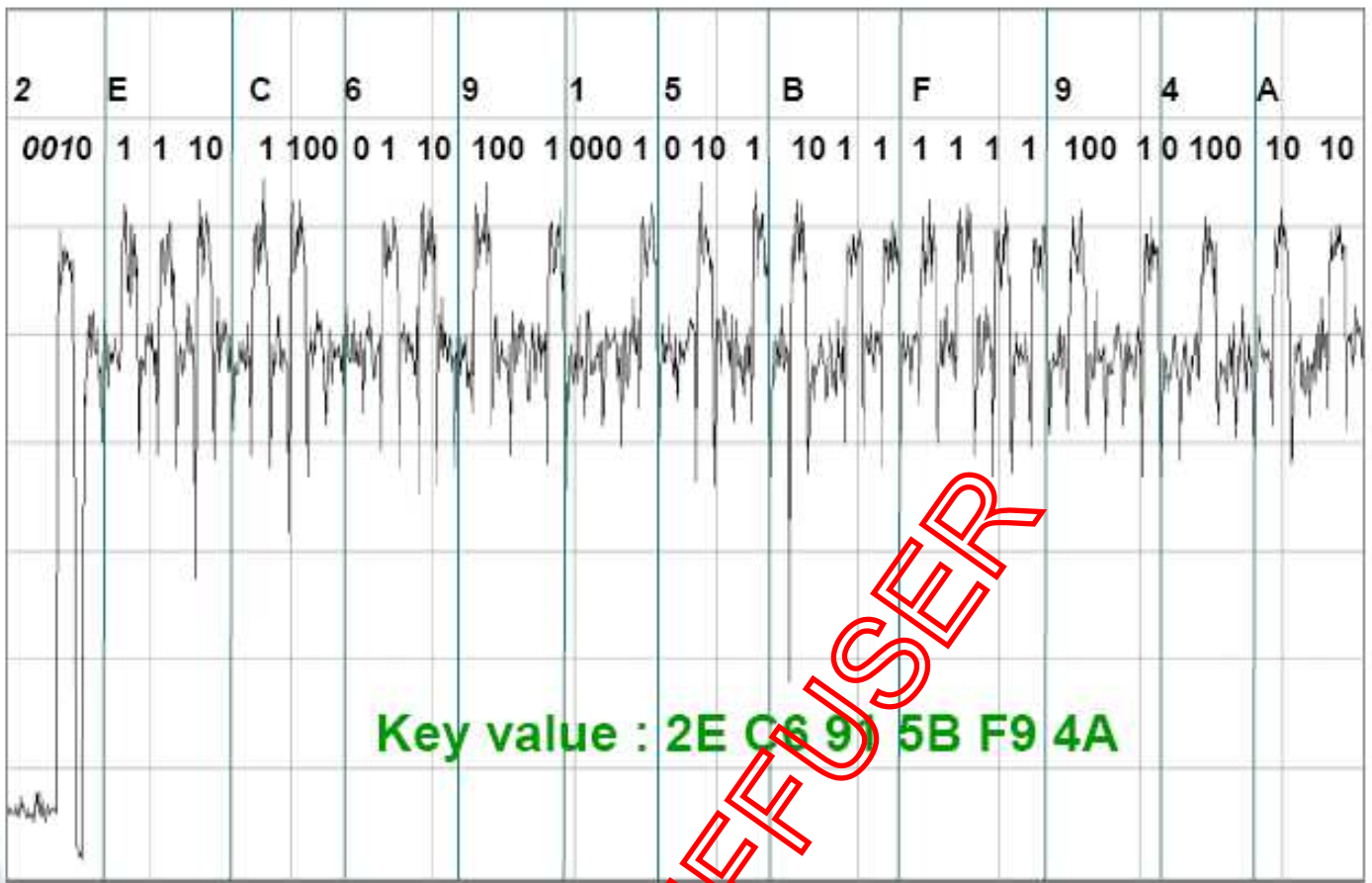
203

Exemple « réel »



204

Exemple « réel »

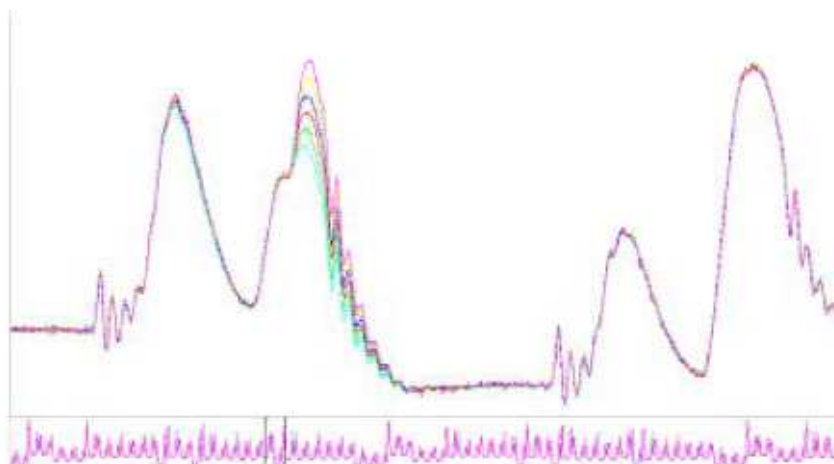


La DPA (Differential Power Analysis)

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.

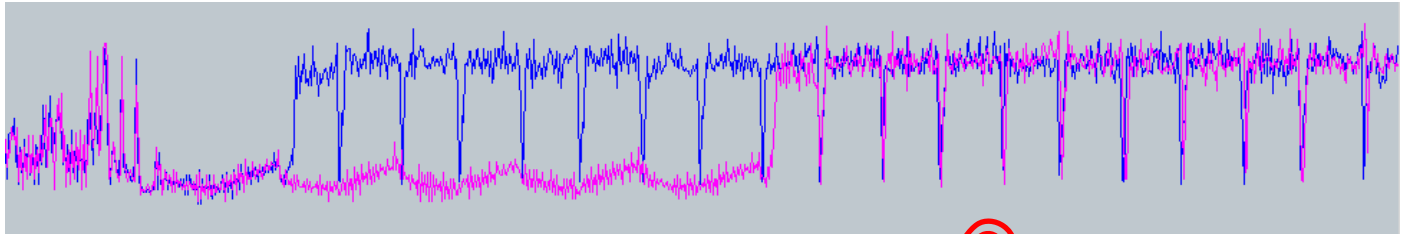
- Based on SPA
Adding the power of statistics to separate signal from noise



CONCRETE ATTACK ON CONCRETE KEYS

- Comparing:

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



token designed and manufactured in 1998...

DPA Hypothesis

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

Input data
(messages M_i)

Algorithm

Output
(cipher texts C_i)

Power
Consumption
Curves W_i
(or other side channel
leakage like EM radiation)

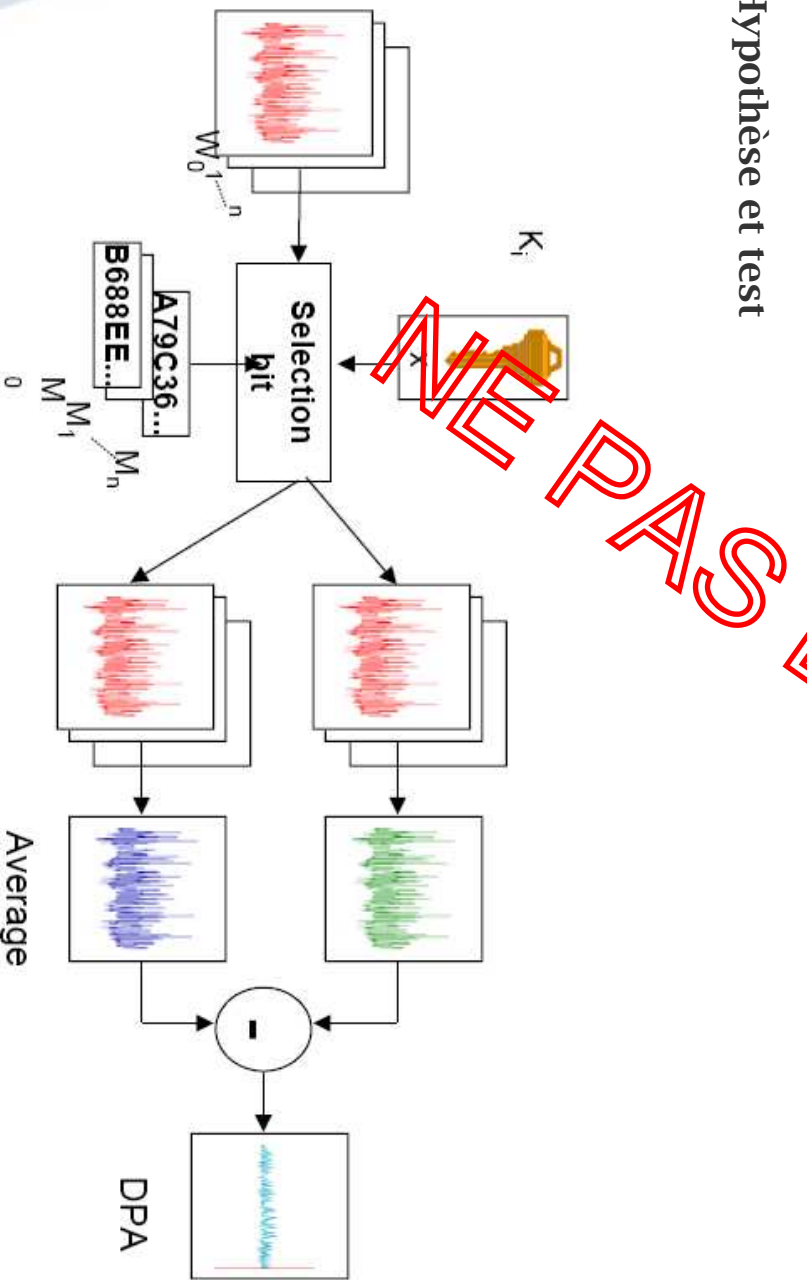
Acquisition procedure

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



209

Hypothèse et test



210

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.

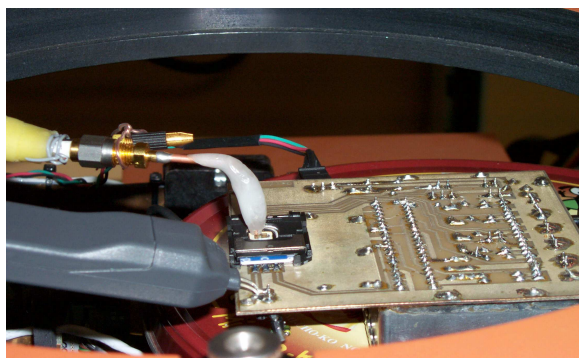
Différence : elle utilise des corrélations entre la consommation de courant et n variables intermédiaires ne dépendant que d'un fragment de la clé et du message d'entrée.

Elle est beaucoup plus difficile à réaliser, mais beaucoup plus puissante.

211

COUNTER-COUNTER MEASURE

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



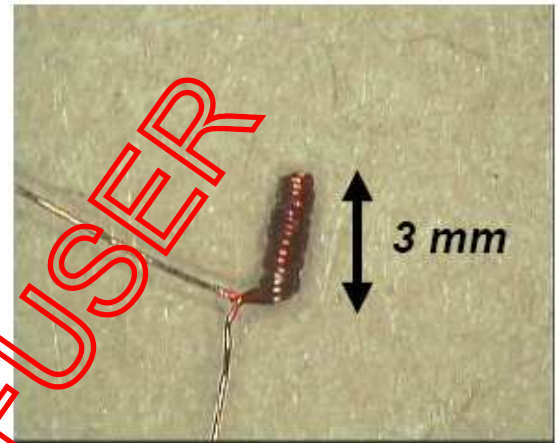
Les émissions EM

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).

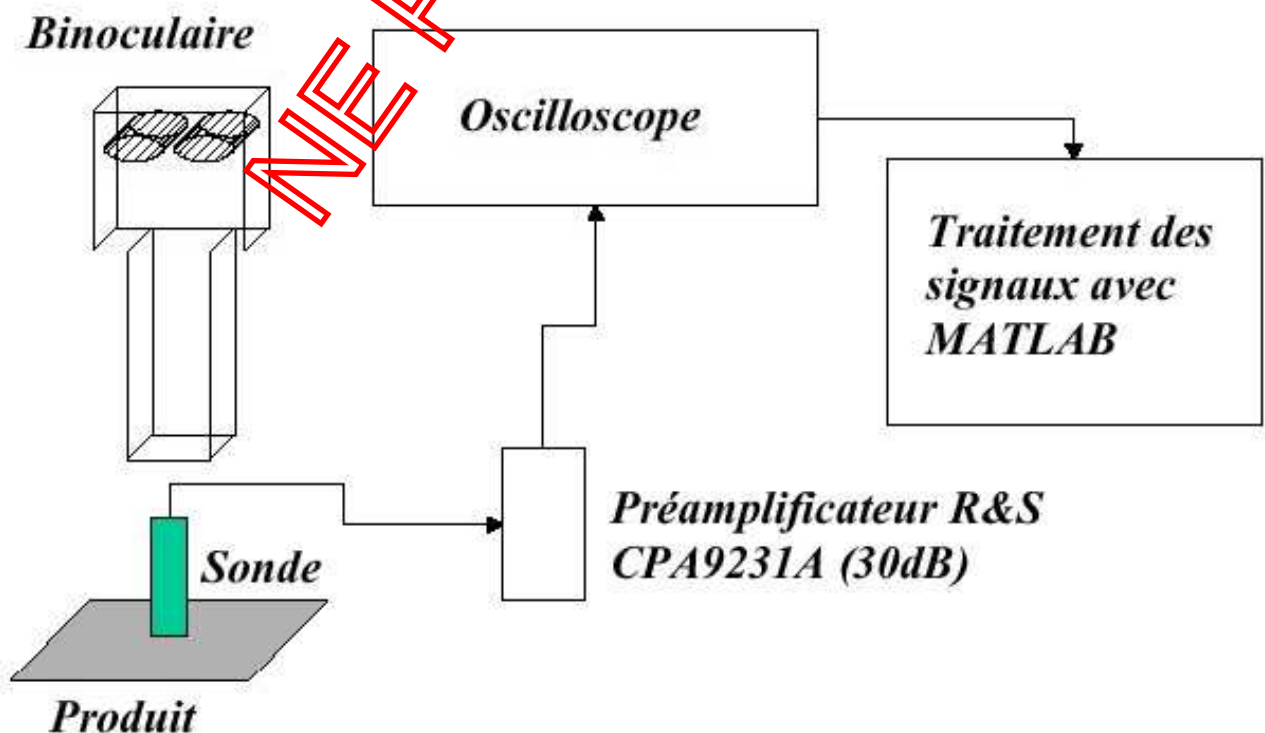
Avantage : Insensible aux contre-mesures physiques tels que l'ajout de bruit en sortie ou le lissage de la consommation globale de courant.

Inconvénient : La reproductibilité des mesures est difficile.



213

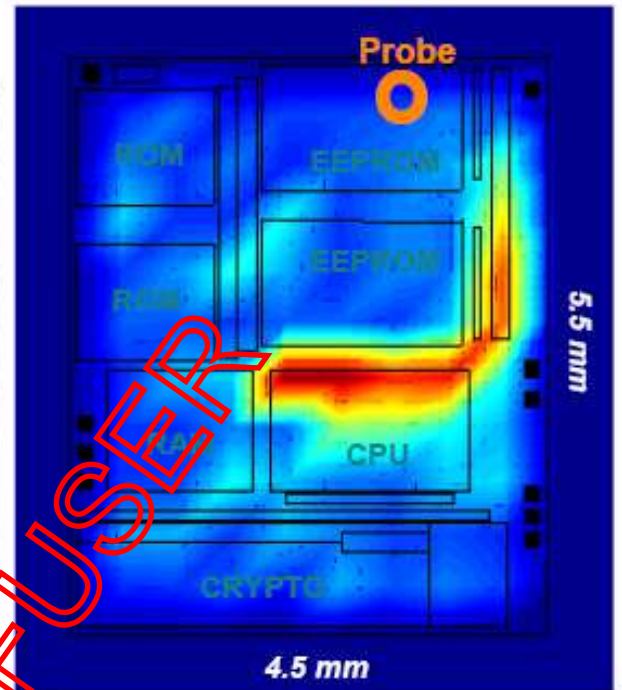
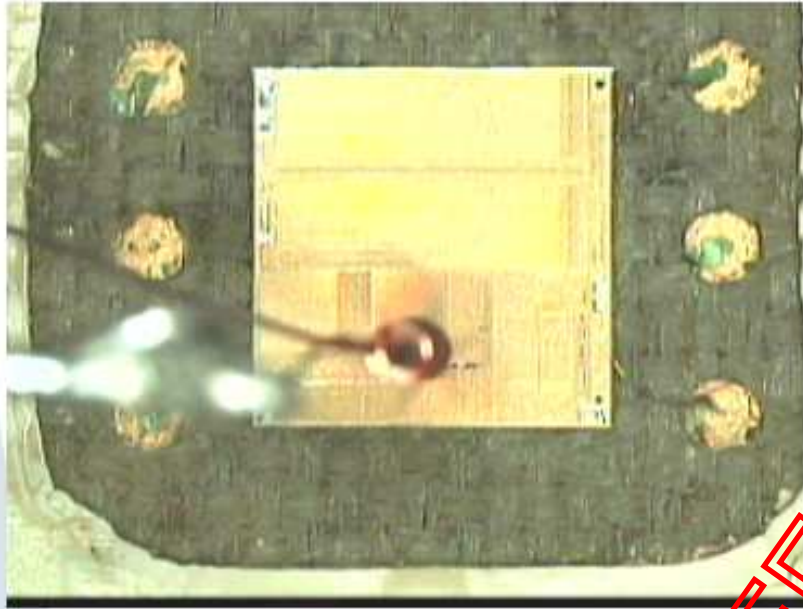
Le dispositif de cartographie EM



214

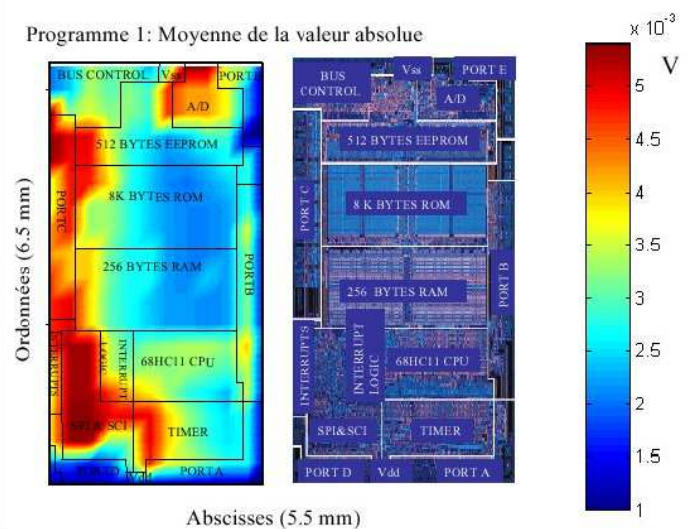
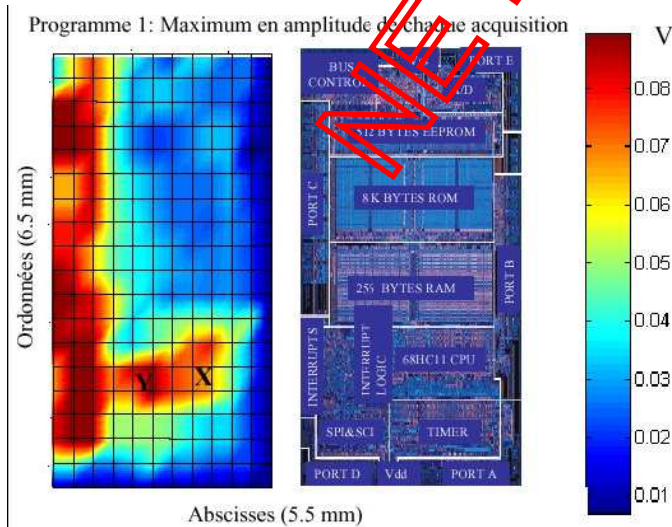
Le dispositif de cartographie EM

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



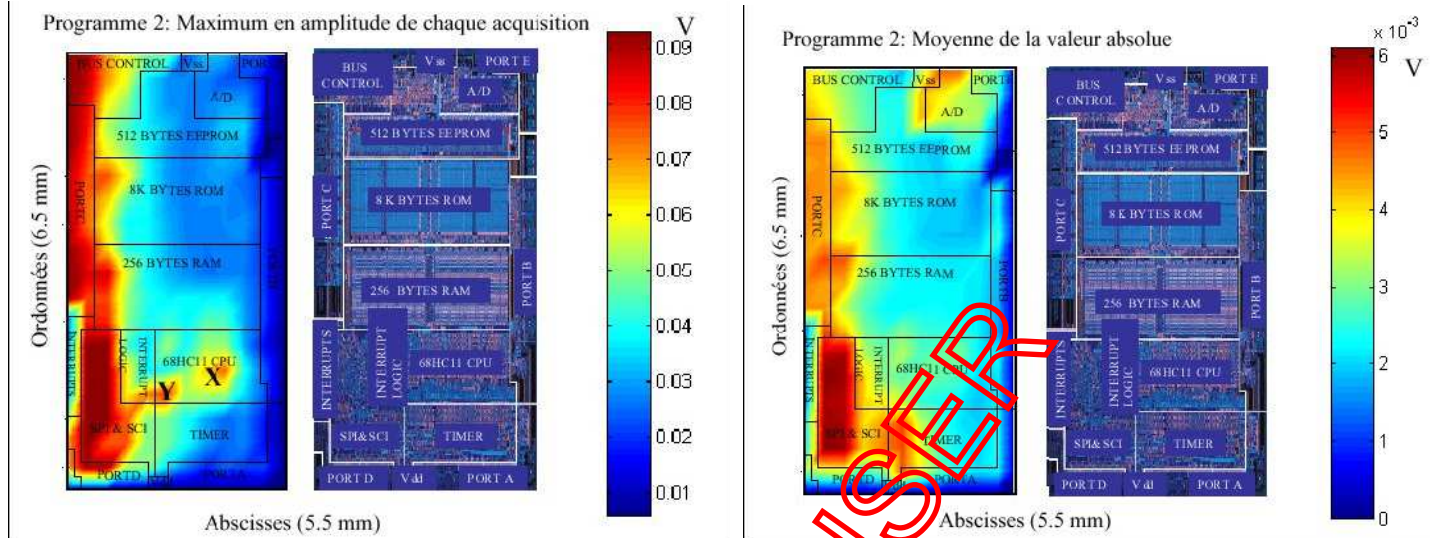
215

Cartographie pour l'algo 1



216

Cartographie pour l'algo 2

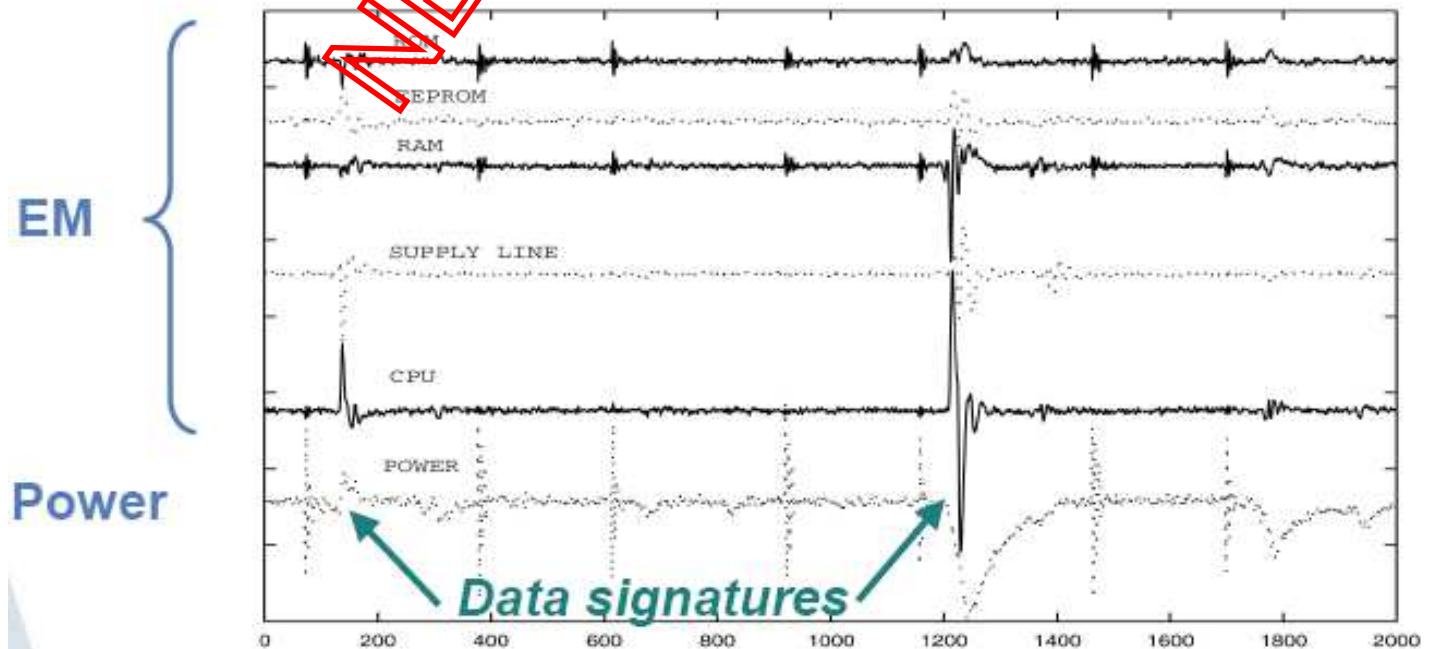


217

Différence EM/Power

• EM signals versus XY probe position

Differential traces between (00h $\oplus$ 00h) and (FFh $\oplus$ 00h) picked up at different locations



218

Attaques EM

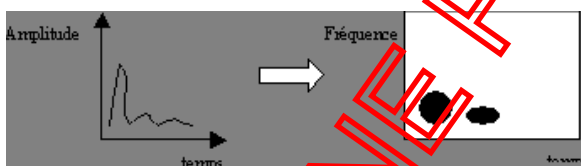
la SEMA (Simple EM Analysis)

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

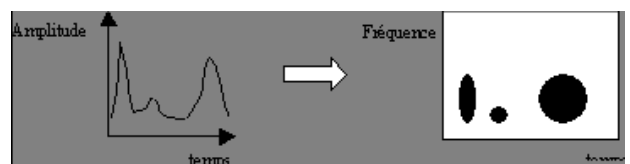
219

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.

220

Invasives

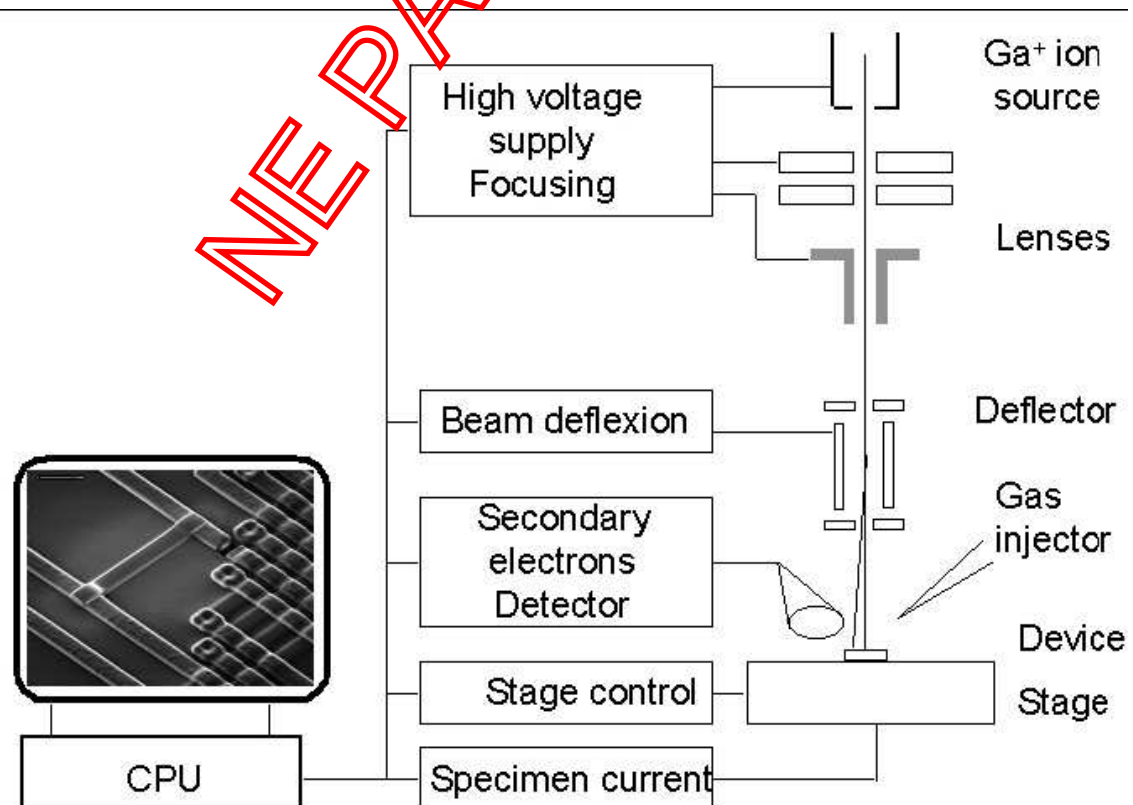
micro-probing

modification de circuit (FIB)

réetro-conception du circuit

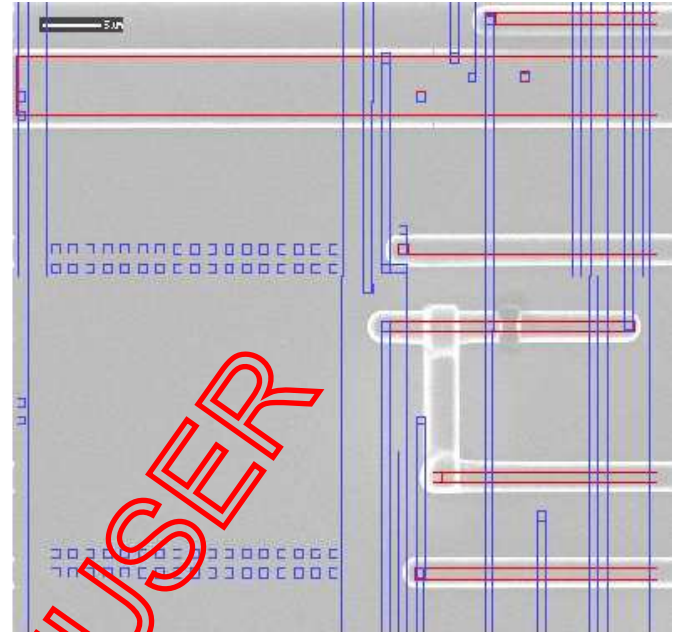
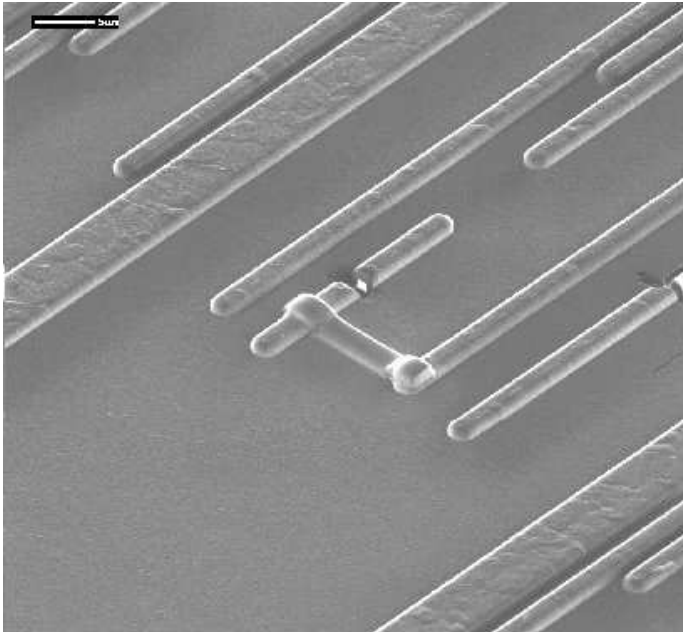
221

Le FIB (Focused Ion Beam)



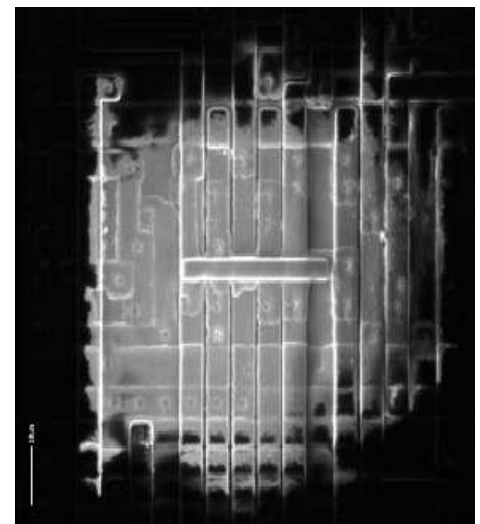
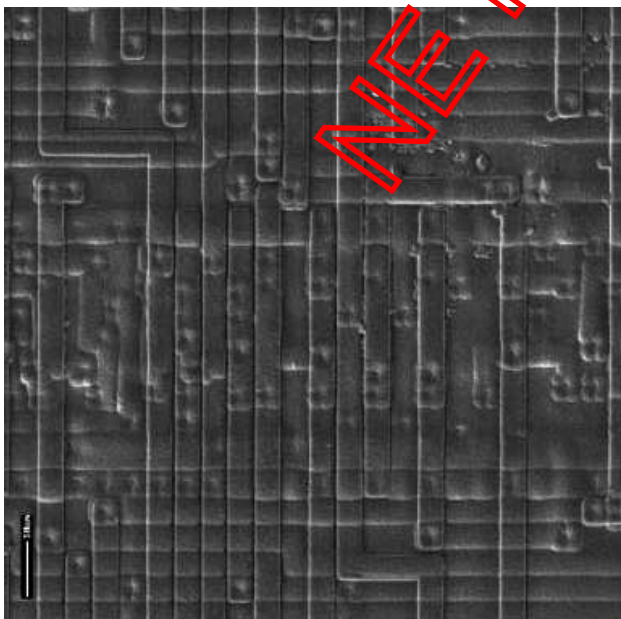
222

Modifications de circuits (1/2)



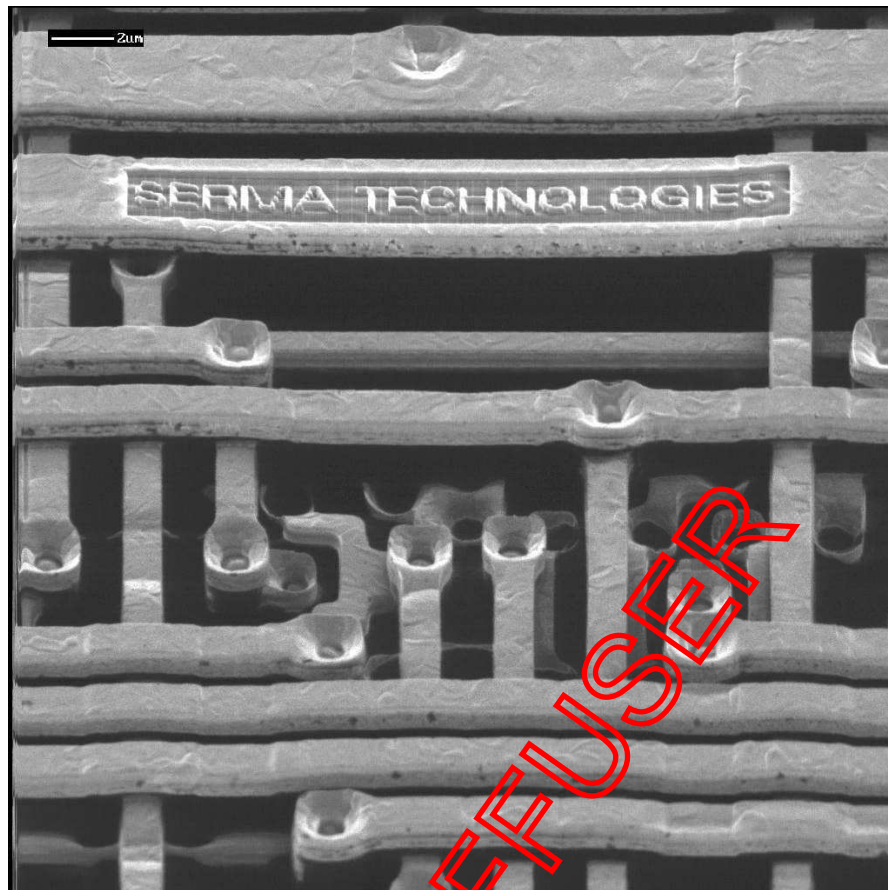
223

Chip Re-Wiring / Addition of a Track



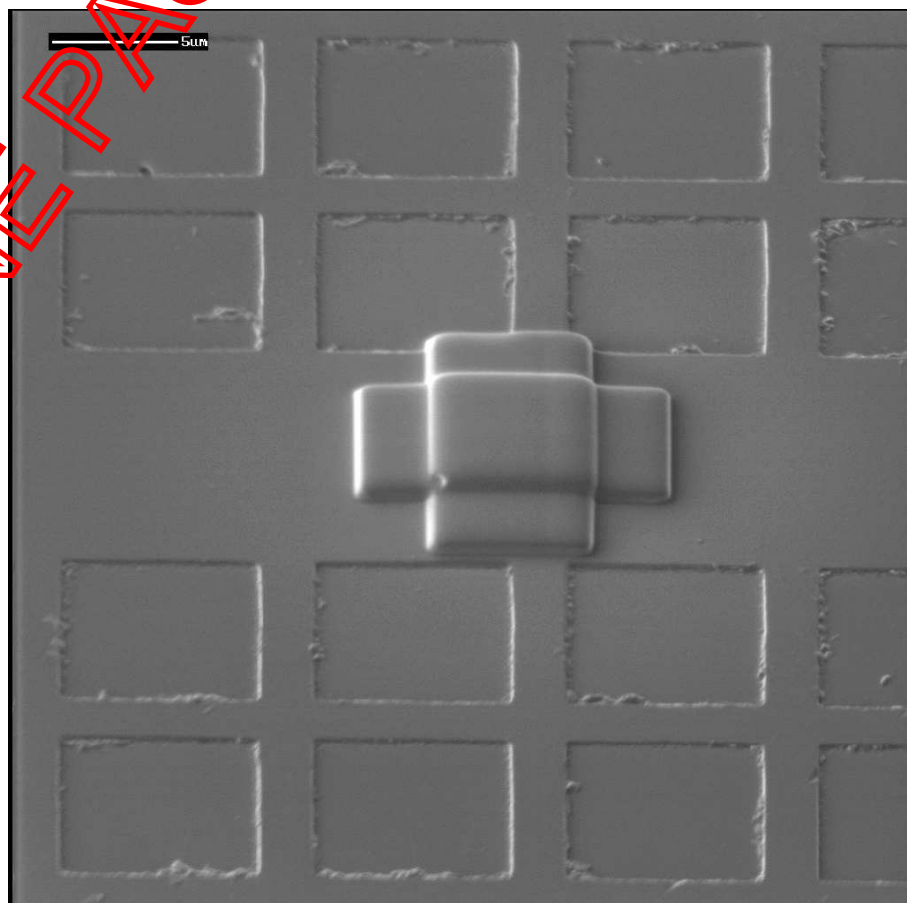
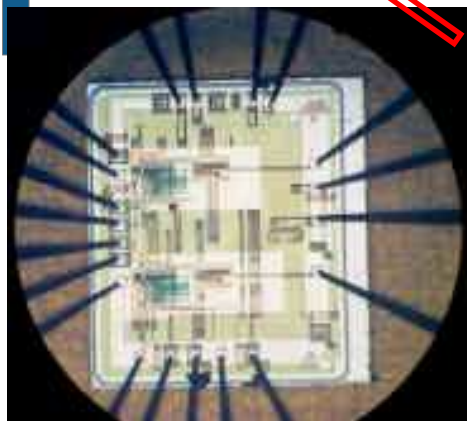
224

Modifications de circuits (2/2)



225

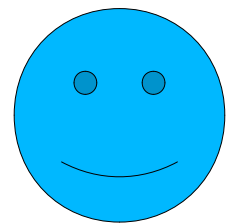
Micro-probing



226

Très cher !

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



227

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 !

Le piratage facile en vidéos

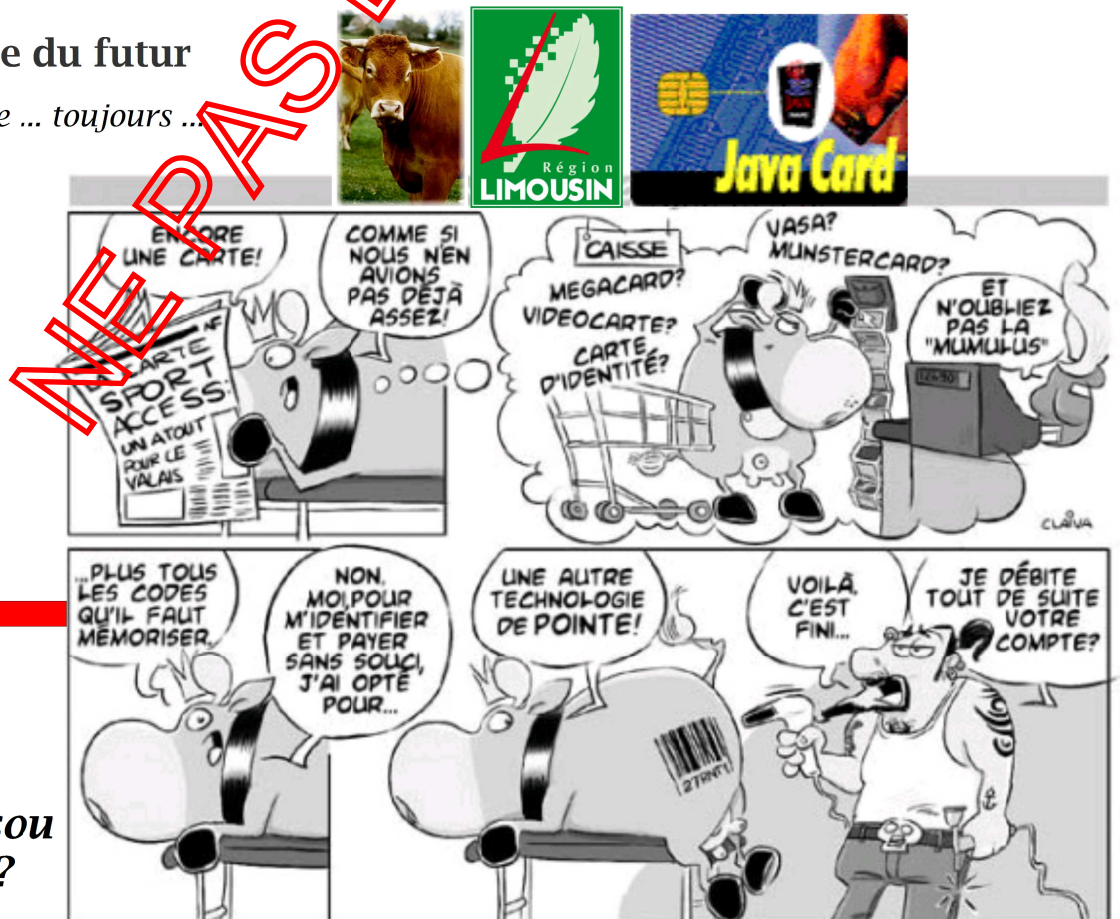
How to Reverse-Engineer a Satellite TV Smart Card.mp4

Mifare Hack.mp4

229

Les cartes à puce du futur

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



Comment résoudre ce problème ?

La carte multi-applicative

240

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.

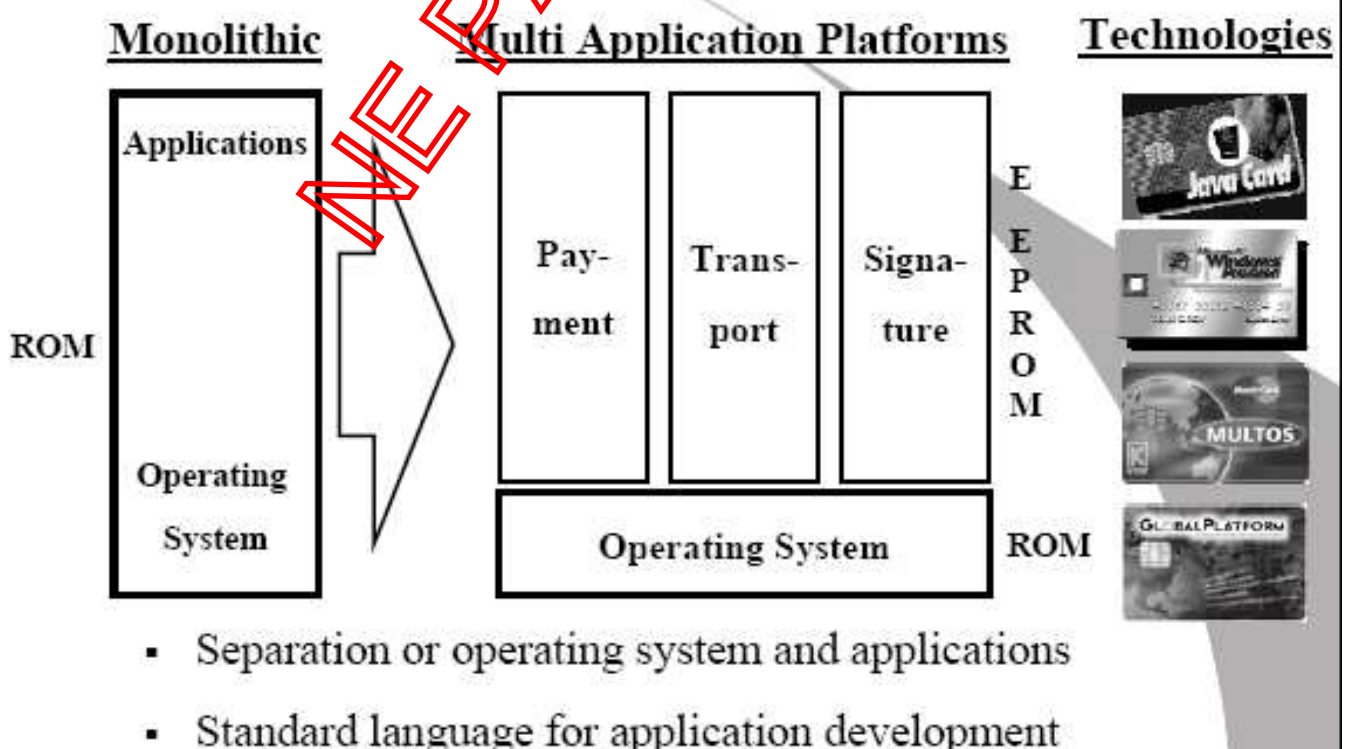


Credit – Application Visa classique avec PIN pour les paiements supérieurs à £10.



241

Évolution

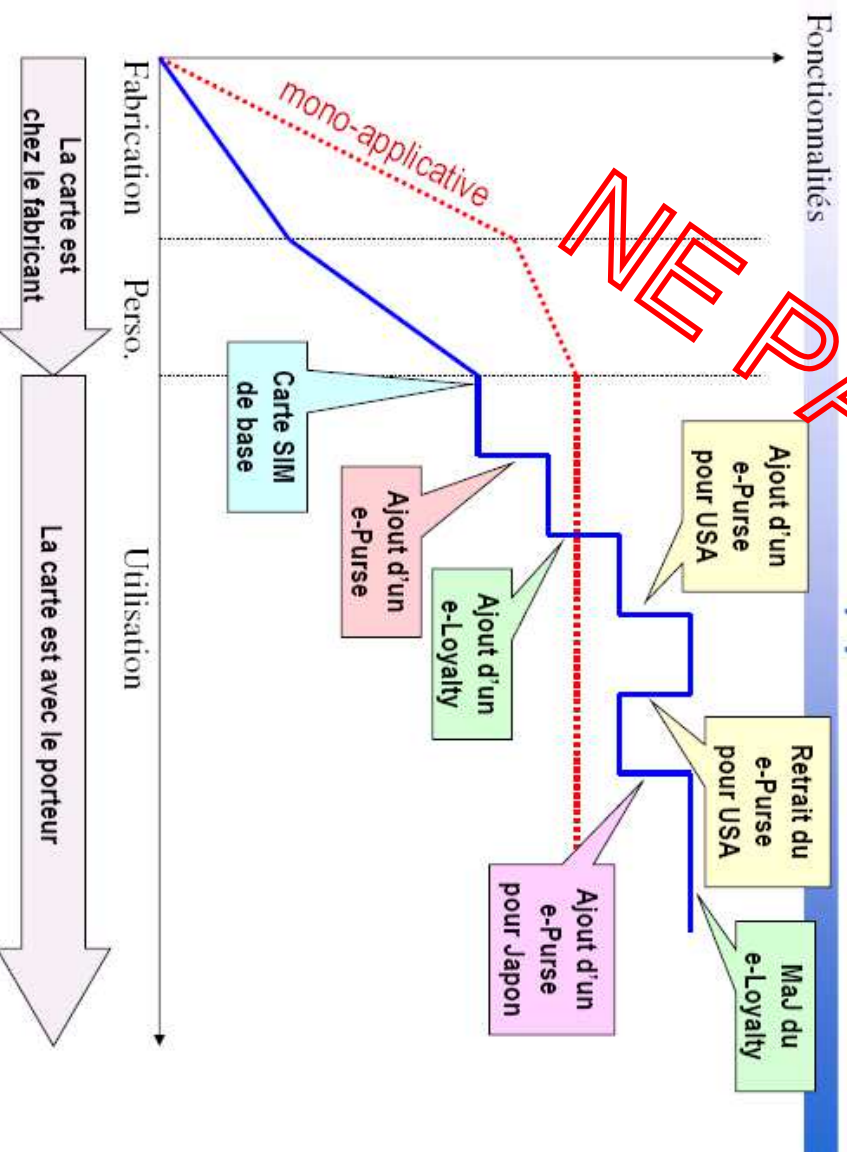


242

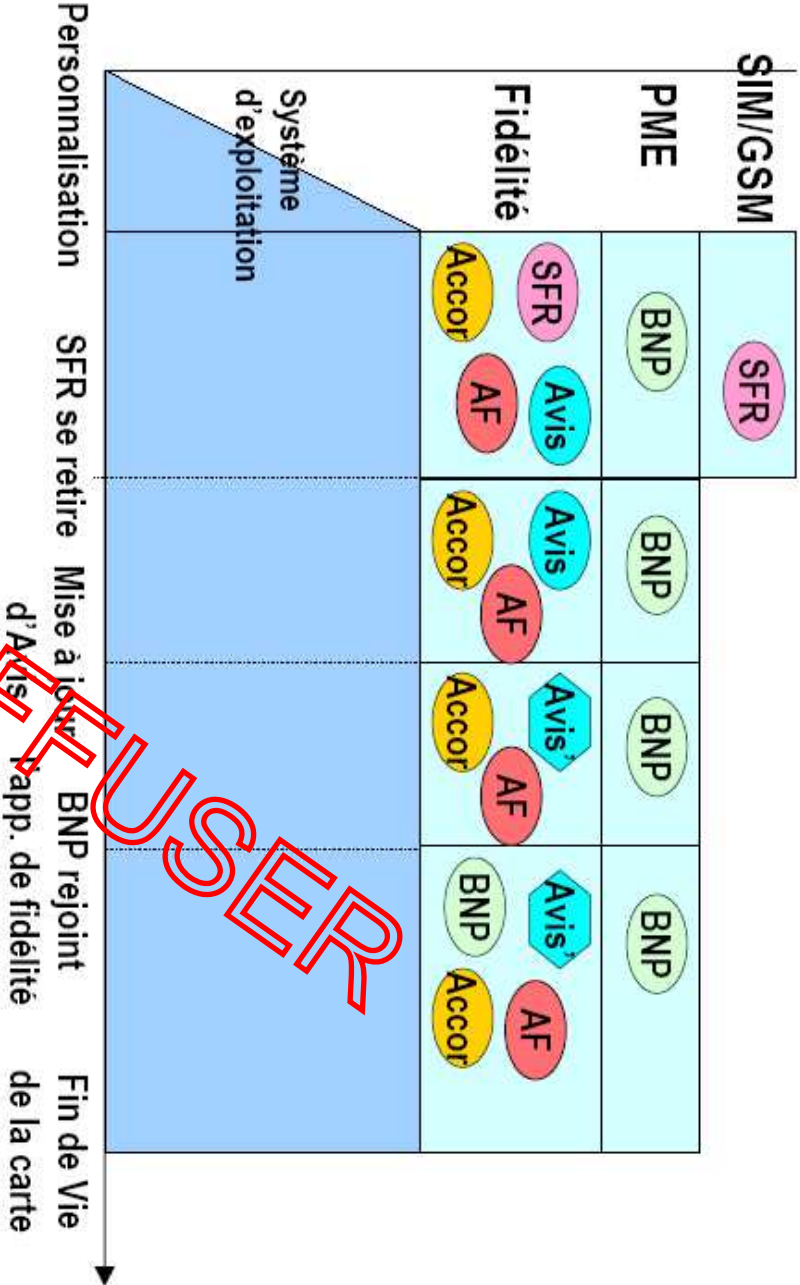
Une nouvelle génération : Les cartes multi-services

- But :
 - Permettre à plusieurs applications de coexister
 - Partager des données entre plusieurs applications (non redondance d'information)
 - Possibilité de charger/décharger des applications en cours de vie de la carte
- Avantages
 - une seule carte pour l'utilisateur
 - Possibilité d'évolution des cartes
- Inconvénients
 - Peur des partages
 - Tailles réduite de la place / application

Le cycle de vie des cartes multi applicative



Cycle de vie d'une carte multi-partenaires

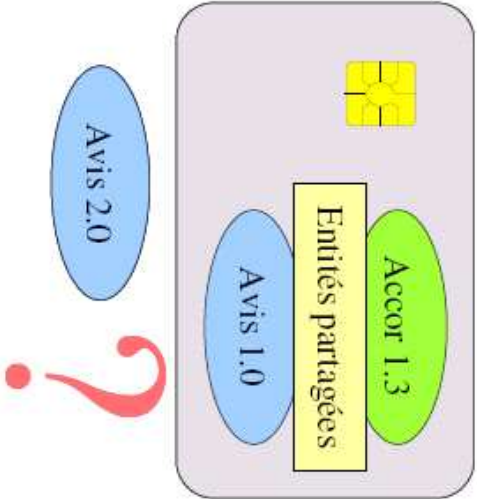
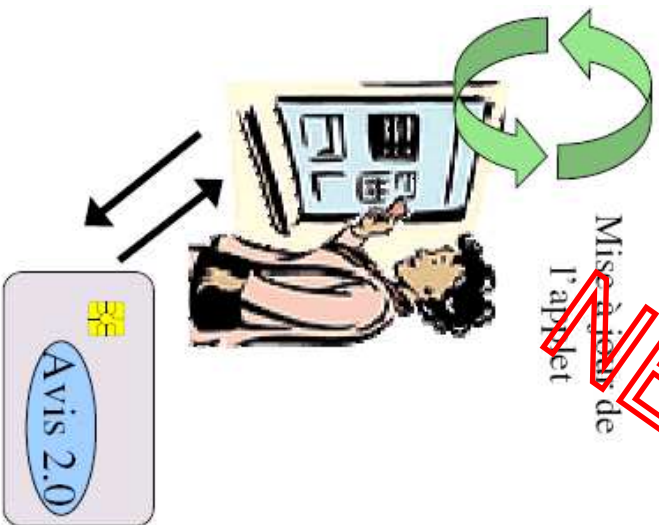


2007

Évolution d'applications Cartes multi-applications

Cas mono-partenaire

Cas multi-partenaires



Besoins des développeurs

- Sortir la carte d' une programmation « élitiste » :
 - Seul 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 test

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

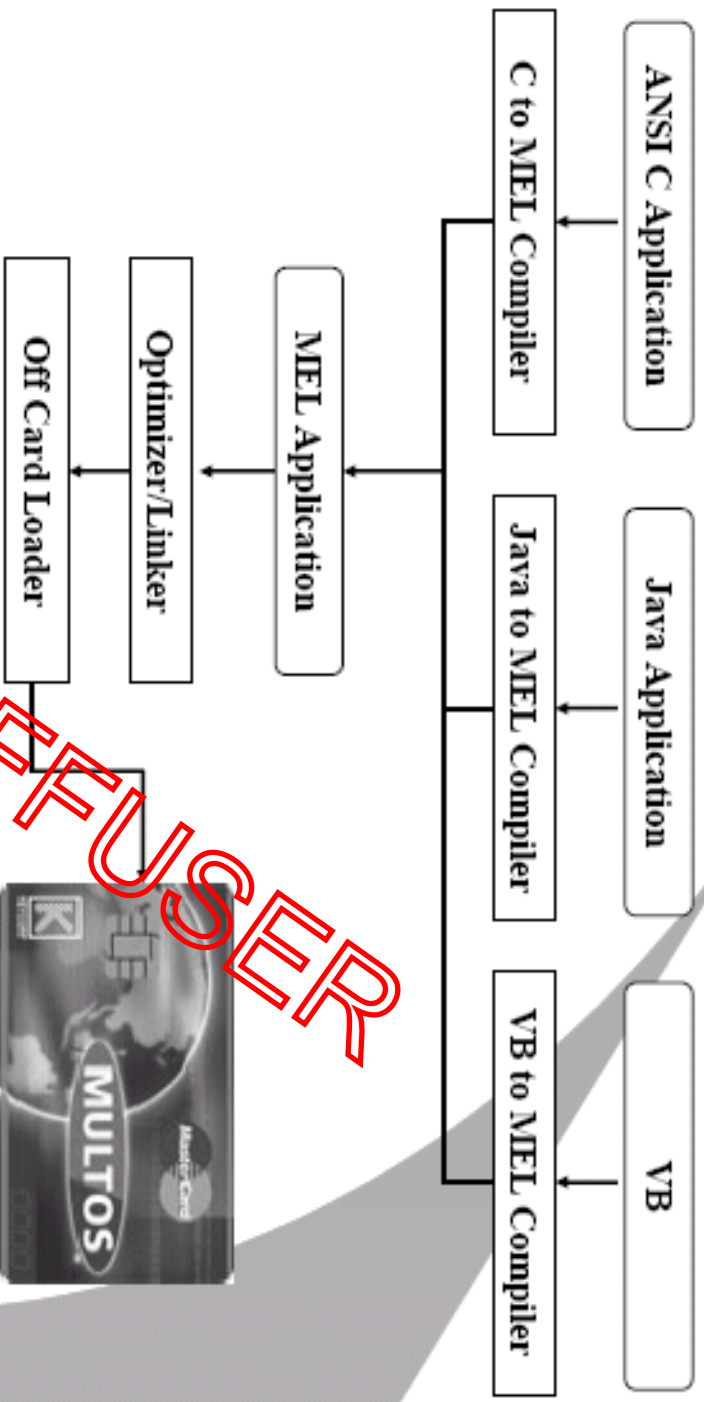
247

NE PAS DÉTESTER

Multos

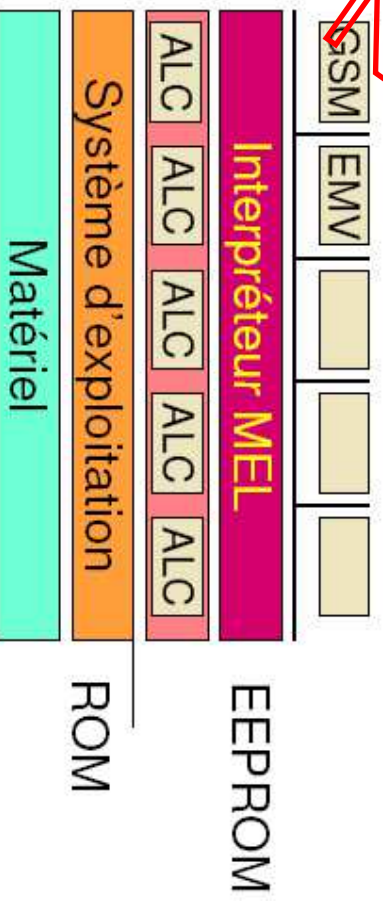


Multos Application Development



Multos

- Carte basée sur un interpréteur MEL (Multos Executable Language)



- Carte limitée par la norme 7816-4
- Plus d'info : <http://www.multos.com>

BasicCard

- Développement d'applications carte en langage Basic
 - Coté Carte
 - interpréteur de « bytecode » P-Code
 - Système de fichier DOS-like
 - Coté Terminal
 - Interpréteur de « bytecode » P-Code
 - Remarque : l'application terminal peut être écrite aussi en C/C++/C# avec un driver PS/SC ou MUSCLE en Java avec OCF
- Avantages
 - Aux applications en très petites séries (Club de tennis, ...)
 - Pour une première approche pédagogique de la carte
- Inconvénients
 - Propriétaire (<http://www.zeitcompol.de/>)
 - Pas de soutien des *Majors*

251

.NET SmartCard

<http://www.hivemindtech.com/>

- Récent (2003)
 - Effet d'annonce !! Souvenez vous de Windows SmartCard !!
- Caractéristiques
 - Implémentation de la CLI adapté à la carte
 - Multi-applications
 - Développement Multi-langage : C#, J#, VB, Jscript, Perl, ...
 - Isolation
 - Application Domain de .NET
 - Transactions
 - Multi-niveaux ?
 - Garbage Collector
 - Mark and Sweep (sans marquage en EEPROM)
 - Communication
 - Inter-applications
 - Channel : flux d'octets bidirectionnel
 - Terminal-Application : APDU, .NET Remoting, Javacard 2.2 RMI



.NET SmartCard

<http://www.hiveminded.com/>

- Implémentation de la CLI adapté à la carte
- Multi-applications
- Développement Multi-langage : C#, J#, VB, Jscript, Perl, ...
- Caractéristiques
 - Isolation
 - Application Domain de .NET
- Transactions
 - Multi-niveaux ?
- Garbage Collector
 - Mark and Sweep (sans marquage en EEPROM)
- Communication
 - Inter-applications
 - Channel : flux d'octets bidirectionnel
- Terminal-Application
 - APDU, .NET Remoting, Javacard 2.2 RMI

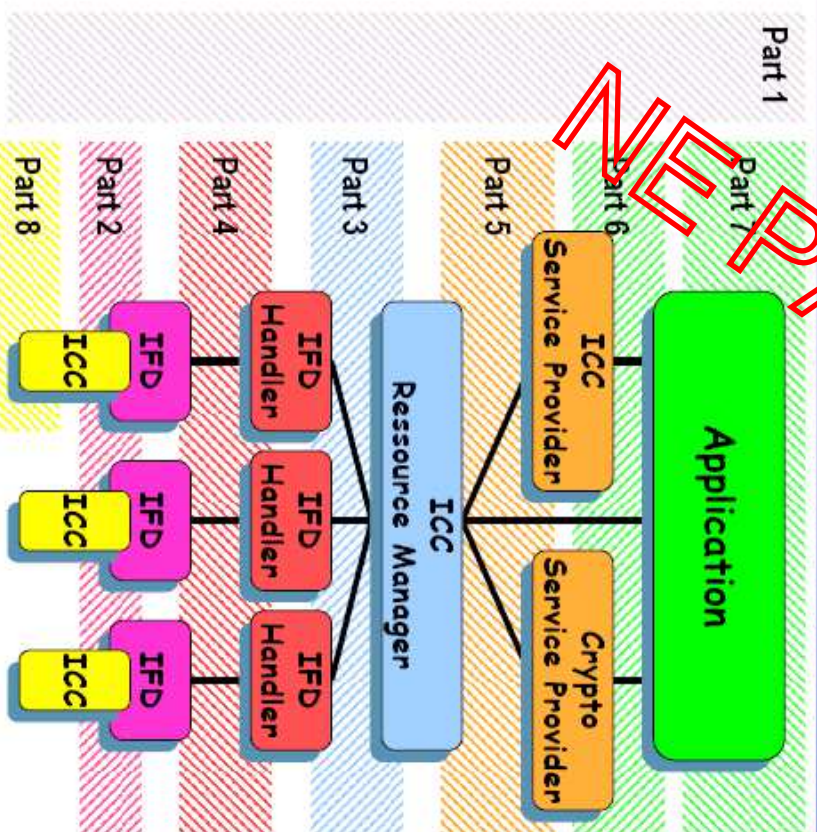


PC/SC

- Define Specifications
 - Platform and OS independent
 - Windows
 - Linux, Apple (MUSCLE)
 - Compliant with ISO7816, support and endorse industry specifications (EMV, GSM...)

260

Architecture et spécifications PCSC



261

Bibliographie

Cours de Gemplus

Cours de Cartes à Microprocesseur de :

Didier Donsez (Univ. Joseph Fourier, Grenoble 1)
Gilles Grimaud (Univ. Lille 1)
Sébastien Jean (Univ Pierre Mendès France, Grenoble 2)
Sylvain Lecomte (Univ. Valenciennes)
Jean-Jacques Vandewalle (Gemalto R&D)

Cours du CNAM de Samia Bouzefrane

« Smart Card Handbook » de Rankl & Effing

Courses of the RHUL : Markantonakis et Mayes

- Javacard
 - <http://java.sun.com/products/javacard/specs.html>
- Multos
 - www.multos.com
- Windows for Smart cards
 - www.microsoft.com/smartcards
- GlobalPlatform
 - <http://www.globalplatform.org/>
- PC Client APIs
 - www.pcscworkgroup.com
 - <http://www.opencard.org/>
- Smart card & Security
 - www.gemplus.com/publications
 - www.iaik.tugraz.at

Webographie

- Une page pleine de références
- <http://paroissien.free.fr/>

262

Bibliographie

• Livres carte

- La Carte à Puce, Coll Que Sais Je ?, n°3492, Ed PUF, 1999
- Henry Dreifus & J. Thomas Monk, *smartcards -- A guide to building and managing smart card applications*, ISBN: 0-471-15748-1, New York: John Wiley & Sons, 1998.
- Scott B. Guthery & Timothy M. Jurgensen, *SmartCard Developer's Kit*, ISBN: 1-57870-027-2, Indianapolis, Indiana: Macmillan Technical Publishing, 1998.
 - <http://www.scdk.comm>

• Ressources

- Ressources du site CITI
 - <http://www.citi.umich.edu/projects/smartcard/>
- FAQ
 - <http://www.scdk.com/atsfaq.htm>
- Google
 - <http://directory.google.com/Top/Computers/Hardware/Systems/Smartcards/>

Bibliographie

- Articles introductifs
 - "The smart card primer," Rinaldo Di Giorgio (JavaWorld, December 1997)
 - <http://www.javaworld.com/jw-12-1997/jw-12-javadev.html>
 - How to write a Java Card applet: A developer's guide
 - http://www.javaworld.com/javaworld/jw-07-1999/jw-07-javacard_p.html
 - Sun's Java Card page contains the specifications of the Java Card APIs, the Java Card Virtual Machine, and the Java Card Runtime Environment:
 - <http://java.sun.com/javacard>
 - "Get a jumpstart on the Java Card," Rinaldo Di Giorgio (JavaWorld, February 1998):
 - <http://www.javaworld.com/javaworld/jw-02-1998/jw-02-javadev.html>
 - "Understanding Java Card 2.0," Zhiguo Chen with Rinaldo Di Giorgio (JavaWorld, March 1998):
 - <http://www.javaworld.com/javaworld/jw-03-1998/jw-03-javadev.html>

4

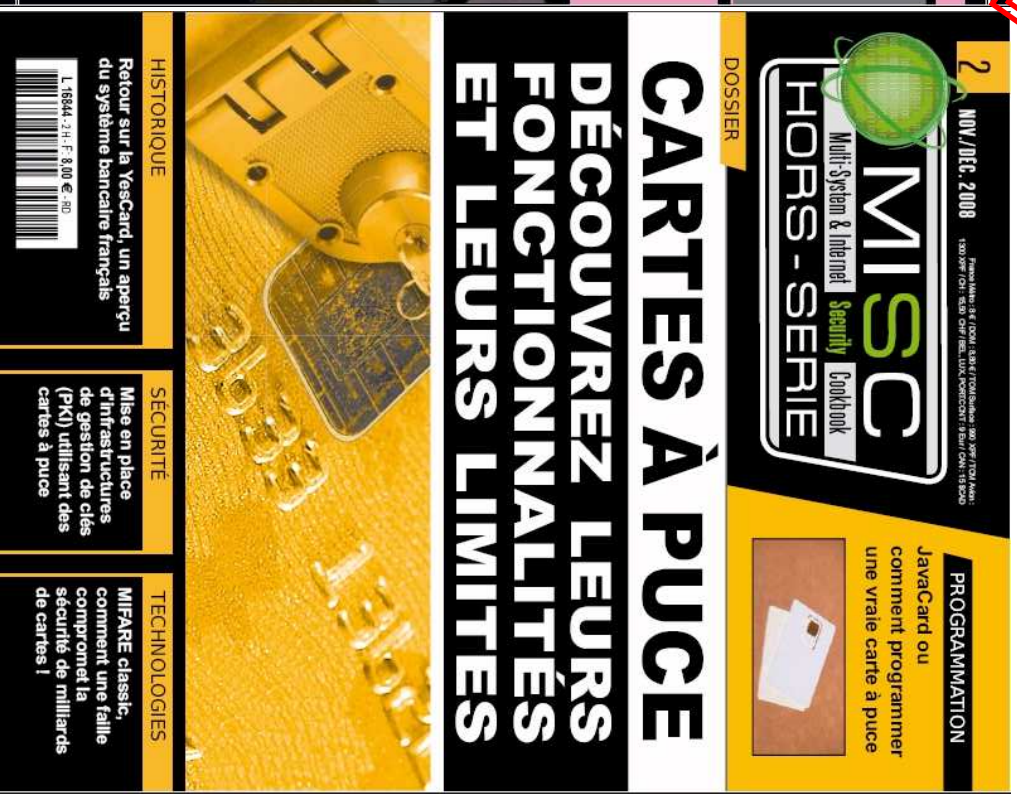
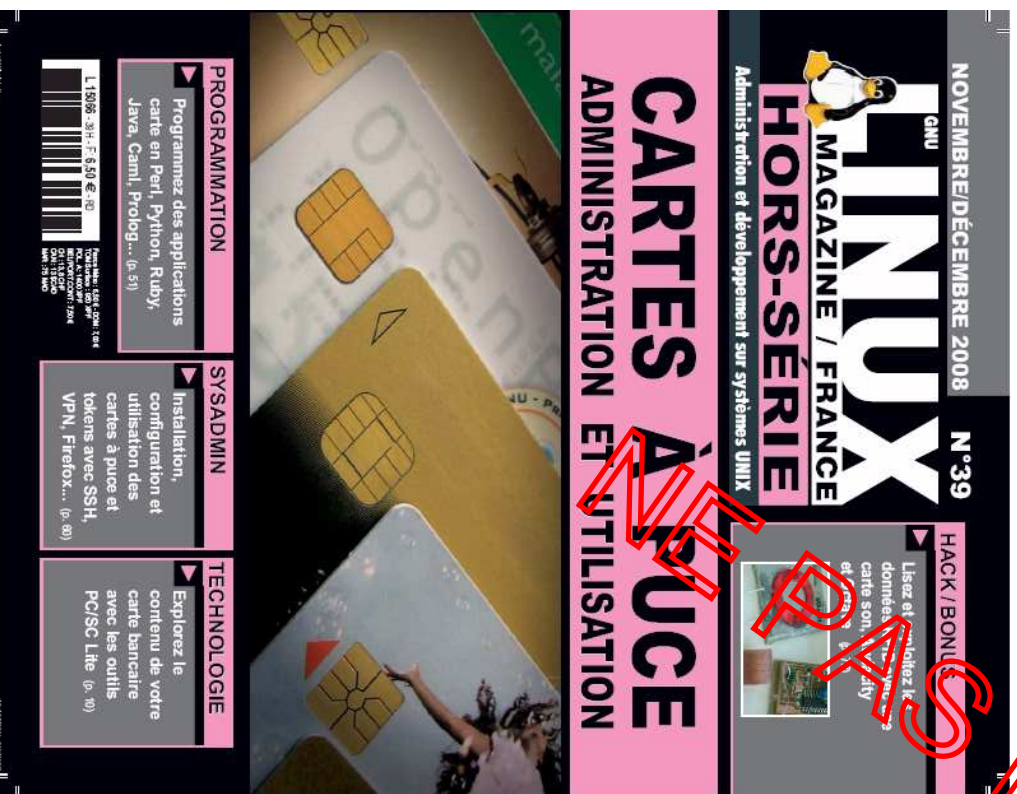
Bibliographie

- API d'accès aux cartes
 - Uwe Hansmann, Martin S. Nicklous, Thomas Schäck, Frank Seliger, Smart Card Application Development Using Java, Ed Springer, 2000, ISBN: 3-540-68829-7,
 - <http://www.opencard.org/SCJavaBook>
 - Orienté OCF et livré avec une carte pour les tests
 - Open Card Framework (OCF)
 - <http://www.opencard.org/>
 - <http://www.gemplus.fr/developers/technologies/opencard/index.htm>
 - PC/SC
 - <http://www.smartcardsys.com/>
 - <http://www.microsoft.com/smartcard/>
 - <http://www.pcseworkgroup.com/>
 - MUSCLE pour Linux
 - <http://www.linuxnet.com/>

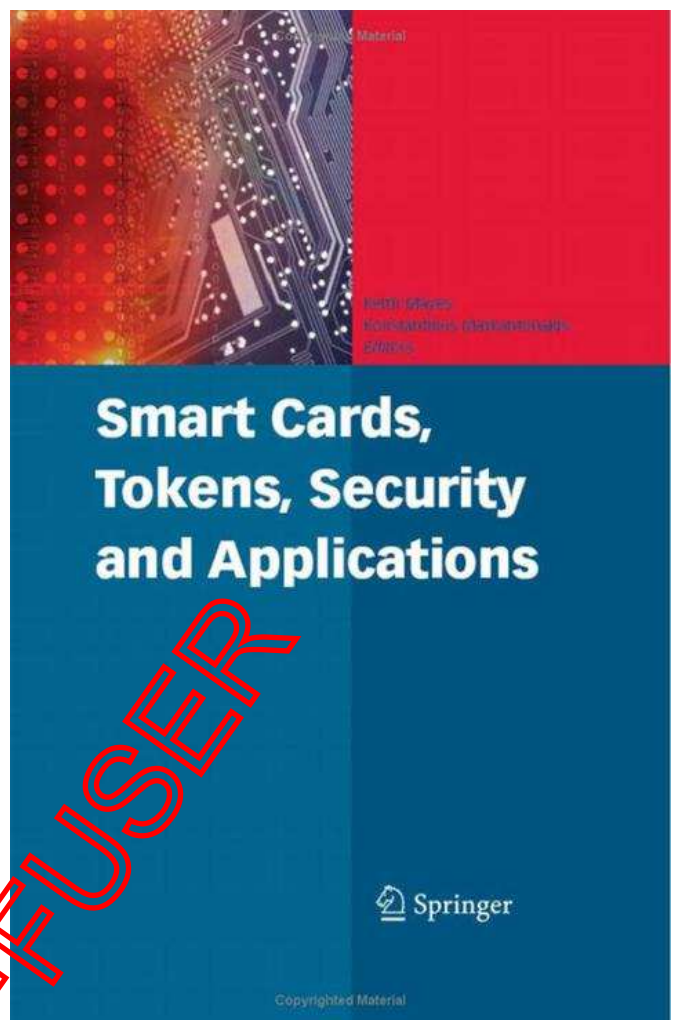
Bibliographie

- Jack M. Kaplan, *SmartCards -- The Global Information Passport*, ISBN: 1-850-32212-0, Boston, Massachusetts: Thomson Computer Press, 1996.
- Mike Hendry, *Smart Card Security and Applications*, ISBN: 0-89006-953-0, Norwood, Massachusetts: ARTECH House, Inc., 1997.
- Présentation des concepts de la sécurité et des principales applications de la carte
 - W. Rankl & W. Effing, *Smart Card Handbook*, ISBN: 0-47196-720-3, New York: John Wiley & Sons, 1997.
 - Chuck Wilson, *Get Smart*, 338 pages (Juin 1, 2001), Mullaney Corporation; ISBN: 0967446058
 - Orienté décideurs (pas de programmation)

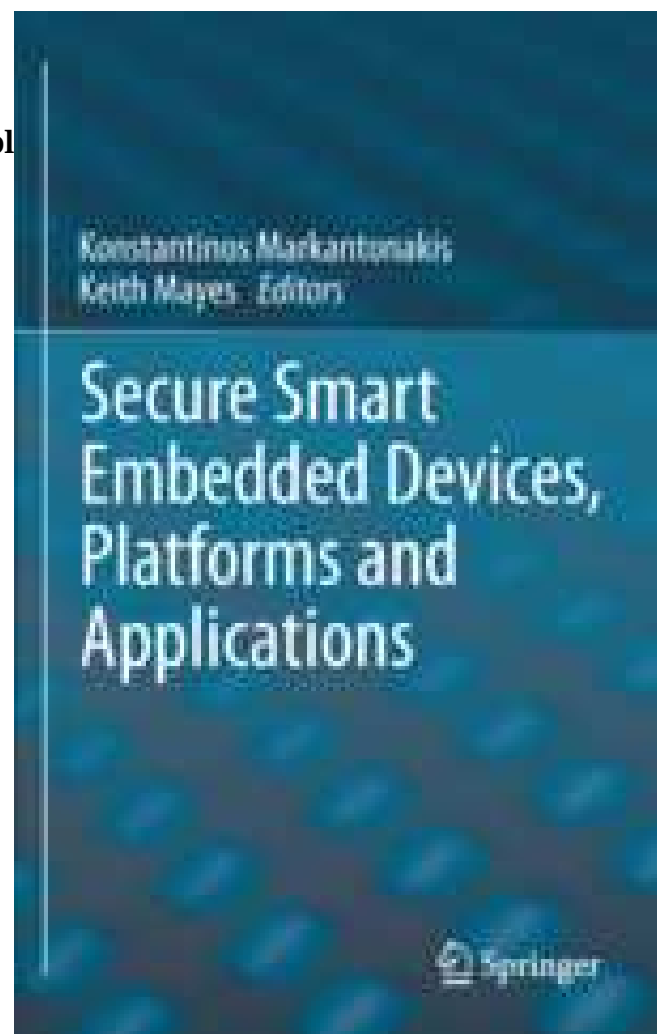
DIFFUSION



Smart Cards, Tokens, Security and Applications
Mayes, Keith; Markantonakis, Konstantinos (Eds.)
2008, XXXVIII, 392 p. 128 illus., Hardcover
ISBN: 978-0-387-72197-2



Secure Smart Embedded Devices, Platforms and Applications
Konstantinos Markantonakis, Keith Mayes
2014, XLI, 568 p. 135 illus., Hardcover
ISBN: 978-1-4614-7914-7



Les cartes à puce
BOUZEFRANE Samia, PARADINAS Pierre
Ouvrage 326 p. 15.6x23.4 cm Relié



Manifestations

Cartes (Paris en novembre) www.cartes.com

Conférence e-Smart (Nice, septembre)

Conférence Cardis (tous les deux ans)

Conférence CHES (tous les ans)

3GPP

...

Questions ?

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com



"My smart-card doesn't
understand me."

**Si votre carte ne vous comprend pas,
la comprenez vous mieux ?**

NE PAS DIFFUSER

