# Introduction à la carte à puce

Pascal Urien

www.enst.fr/~urien/cours.html

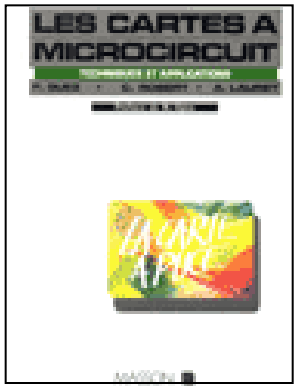
http://perso.telecom-paristech.fr/~urien/intro_carte_2018.pdf

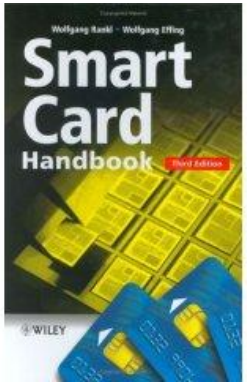
http://www.enst.fr/~urien/introcarte2016.pdf

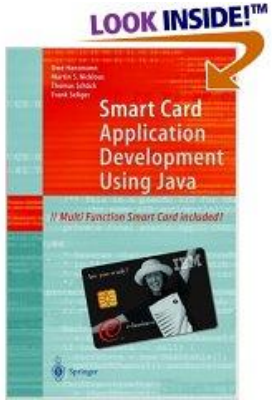
# Bibliographie

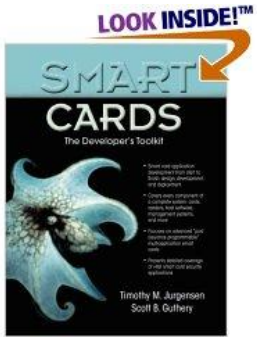

- *La carte à puce*. Jean Donio, Jean Leroux Les Jardins. Que sais-je? n°3492, Éditions PUF.
- *Les cartes à microcircuit (88)*. Éditions Éditions Masson. F.Guez, C.Robert, A.Lauret.
- *Smart Card handbook*. W. ERankl , W. Effing. Editions Willey.
- *Smart Card application Development Using Java*. Martin S. Nicklous, Thomas Schack, Frank Seliger, Uwe Hansmann, Martin Scott Nicklous, Thomas Schaeck. Editions Springer.
- *Smart Cards - The developer's Kit*. Timothy M. Jurgensen, Scott B. Guthery. Editions Prentice Hall.
- *Java Card™ Technology for Smart Cards*. Zhiqun Chen. Editions Addison Wesley.

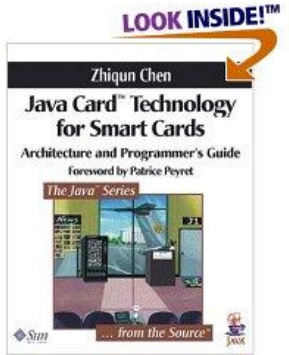




# La genèse

# La genèse

- La carte à puce est une technologie pluridisciplinaire qui s'appuie sur trois éléments
  - La microélectronique, le traitement de l'information, et la cryptographie
- René Barjavel «La Nuit des Temps» Éditions Denoël, 1968
  - « 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 »
- Les brevets
  - Etats Unis
    - Ellingboe (1970) propose un moyen de paiement électronique avec une carte de crédit à contacts;
    - Halpern (1972) introduit un stylo électronique sécurisé de paiement.


Japon
- Arimura (1970) décrit une méthode d'authentification dynamique réalisée à l'aide d'un dispositif d'identification.

En France,
- Roland Moreno (1974), Michel Ugon (1977) et Guillou (1979).

Carte

Lecteur

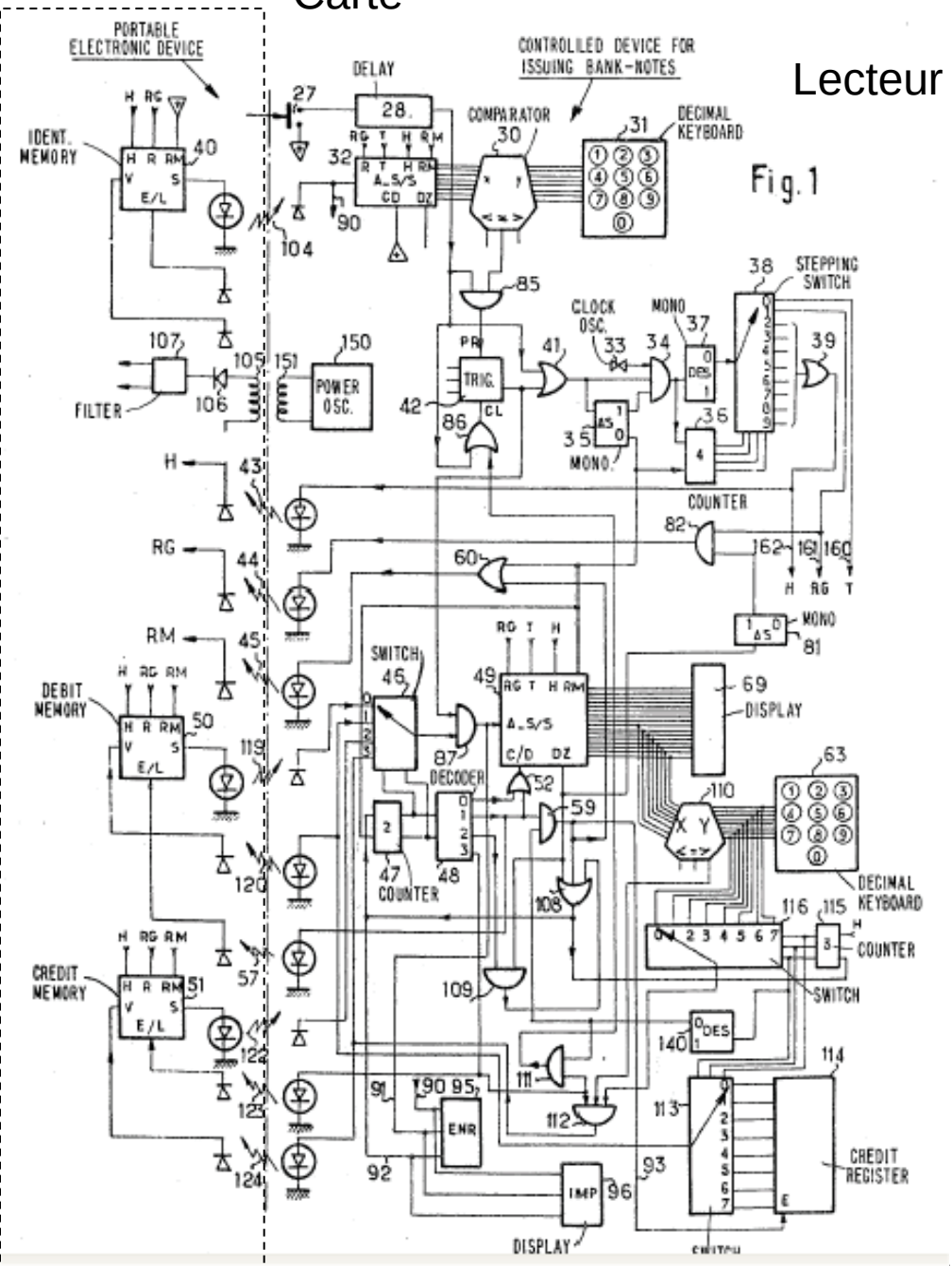


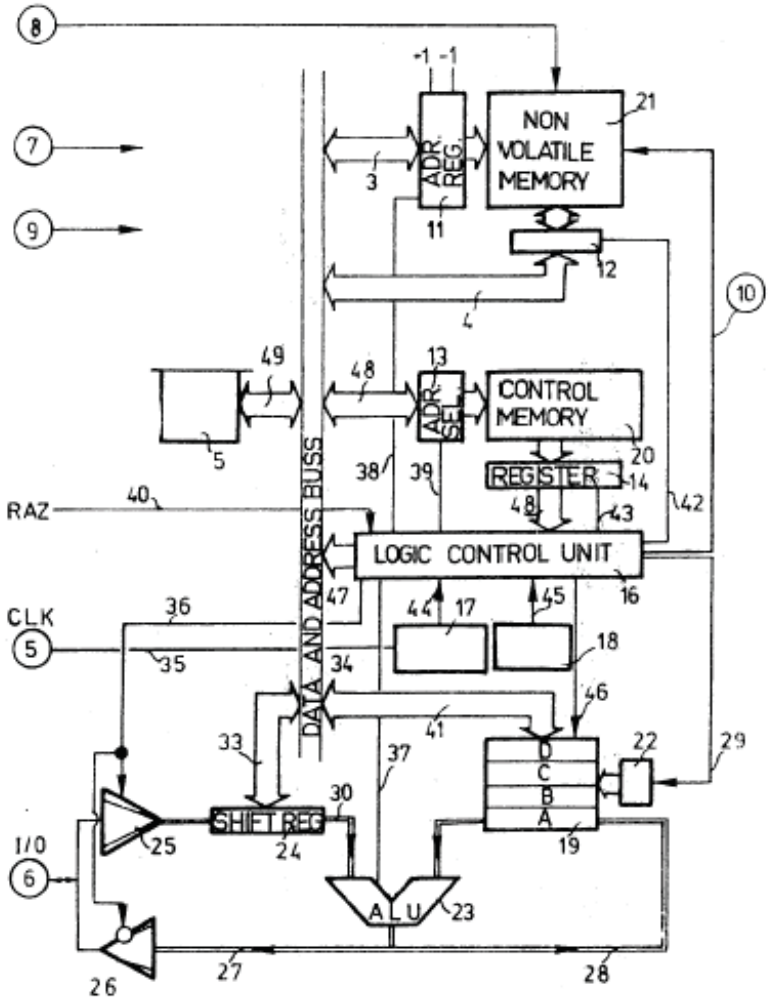

Fig. 1

# Roland Moreno

- 25 mars 1974, Brevet 74.10191
- 21 mars 1975, US 4,007,355


1978, une télécarte 1Kbit

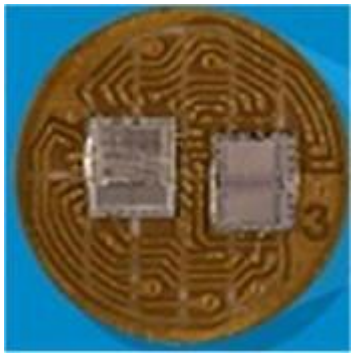
# Michel Ugon




- 26 août 1977, brevet 77.26107
- 25 août 1978, US 4,211,919

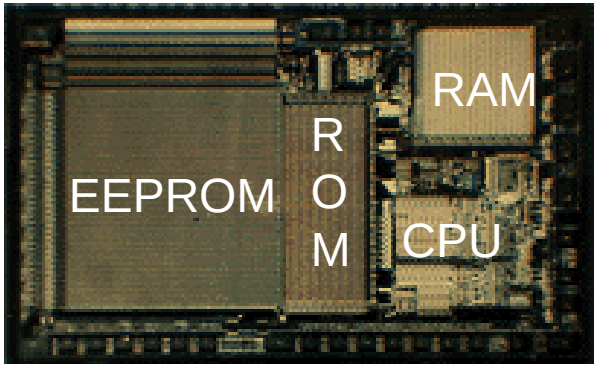
# Le SPOM

- 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 et Motorola
  - SPOM, Self Programmable One chip Microcomputer




1979, carte hybride
à deux puces

1981, chip SPOM1 en NMOS 3.5 µm
(42000 transistors sur 19.5mm²)



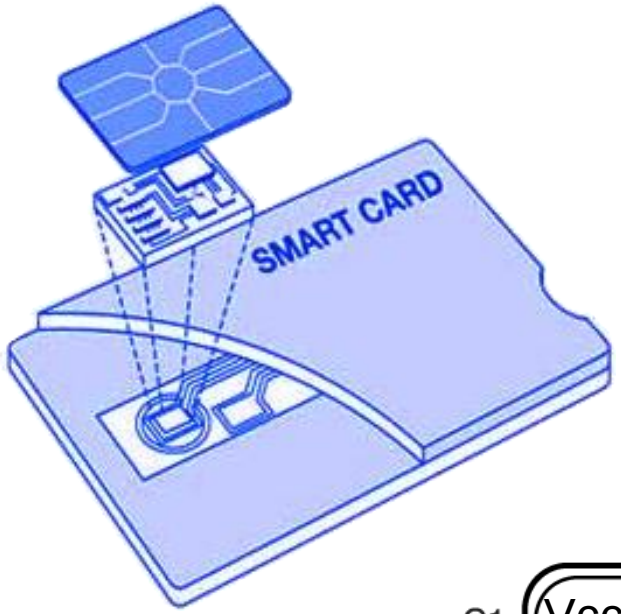

1988, le chip 21
avec une mémoire EEPROM

# Quelques dates

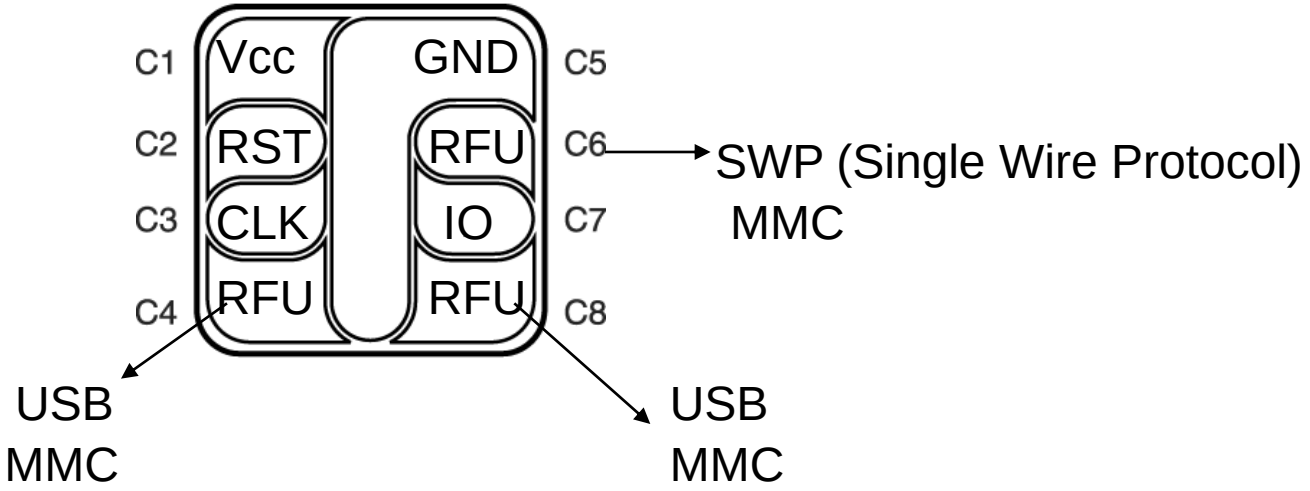
- 1974, Brevet de R.Moreno
- 1977, Brevet de M.Ugon
- 1987, Première norme ISO 7816
- 1988, Spécification de la carte SIM
- 1995, Attaque DPA Paul Kocher
- 1996, Première norme EMV
- 1997, Brevet Java Card, US 6,308,317
- 2002, dotnet smart card, Hiveminded

# La carte à puce, aperçu de la technologie

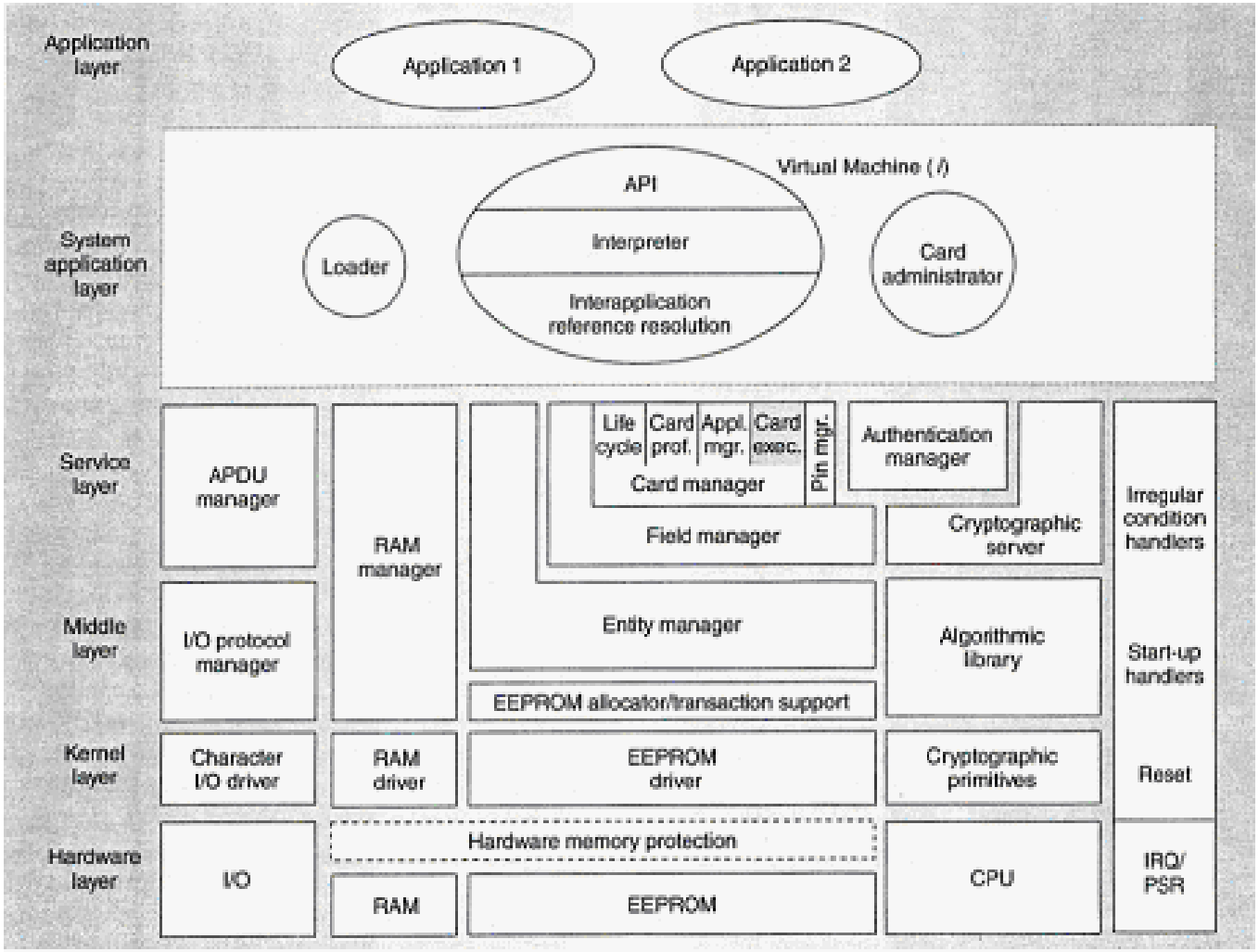
# Aperçu technologique



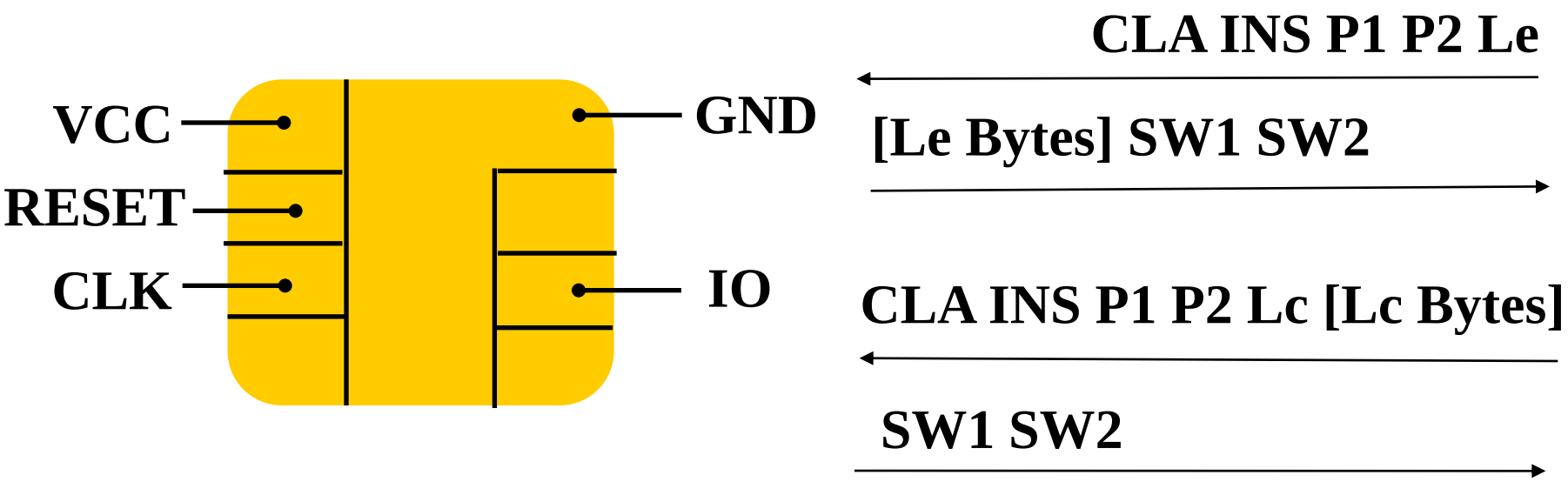




# Exemple de Système d'Exploitation



- ***Guillou,L.C, Ugon, M, Quisquater,J.J "Smartcard: a Standardized Security Device Dedicated to Public Cryptology", 1992.***
  - "What a smartcard does*. The five operations of a smartcard are 1- input data, 2- output data, 3- read data from non volatile memory (NVM), 4- write or erase data in NVM, 5- compute a cryptographic function."*



# Commandes de base ISO7816-4

**Y=F(x)**

**Lecture LE bytes**

**CLA INS P1 P2 Le**

**[Le Bytes] SW1 SW2**

**1- Ecriture xx bytes**

**CLA INS P1 P2 xx [xx Bytes]**

**SW1=61 SW2=yy**



**Ecriture Lc bytes**

**CLA INS P1 P2 Lc [Lc Bytes]**

**SW1 SW2**



**2- Lecture yy bytes**

**CLA INS=C0 P1=0 P2=0 P3=yy**

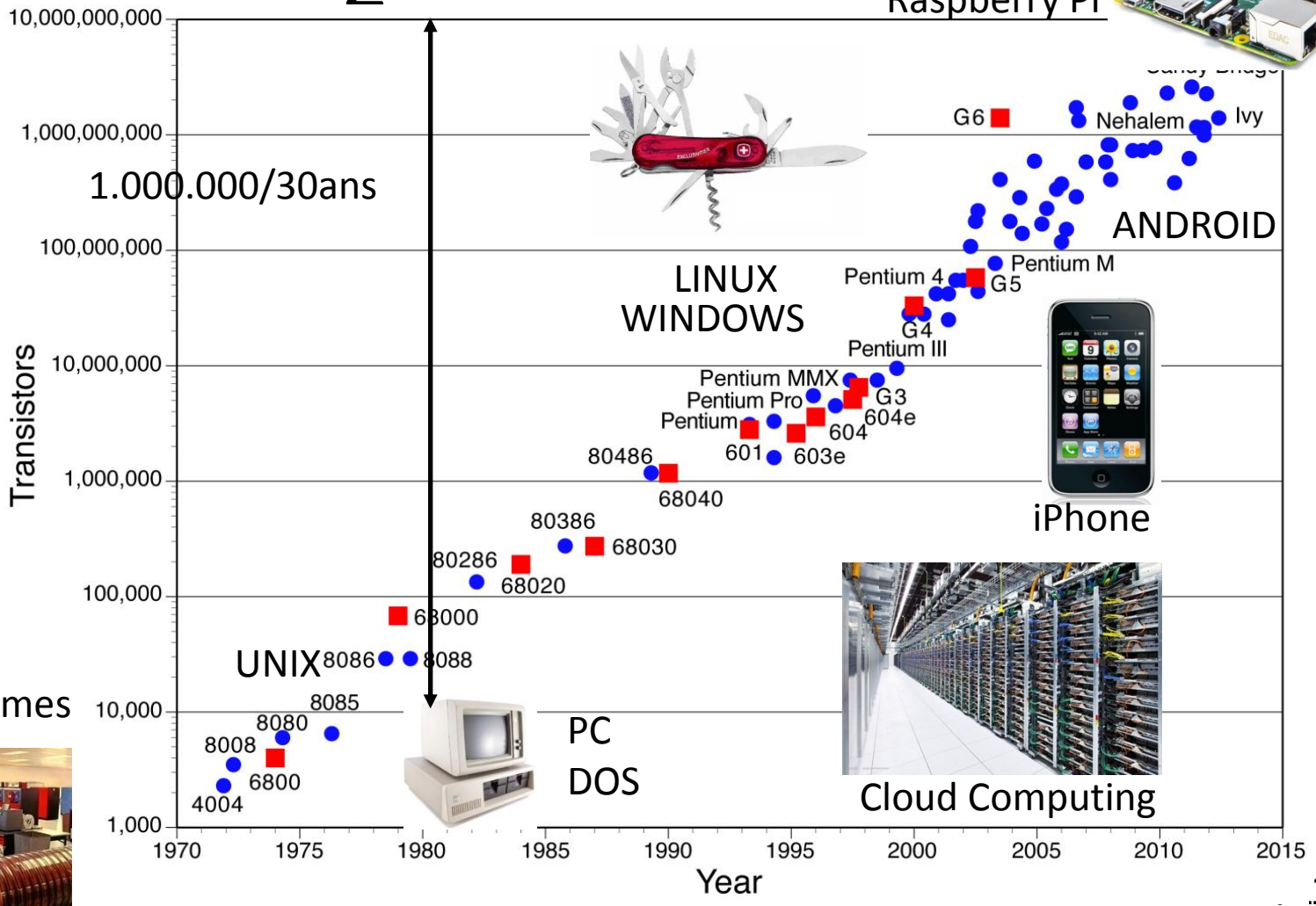
**[yy bytes] SW1 SW2**



**CLA INS P1 P2 Lc [Lc Bytes] Le [Expected bytes]**

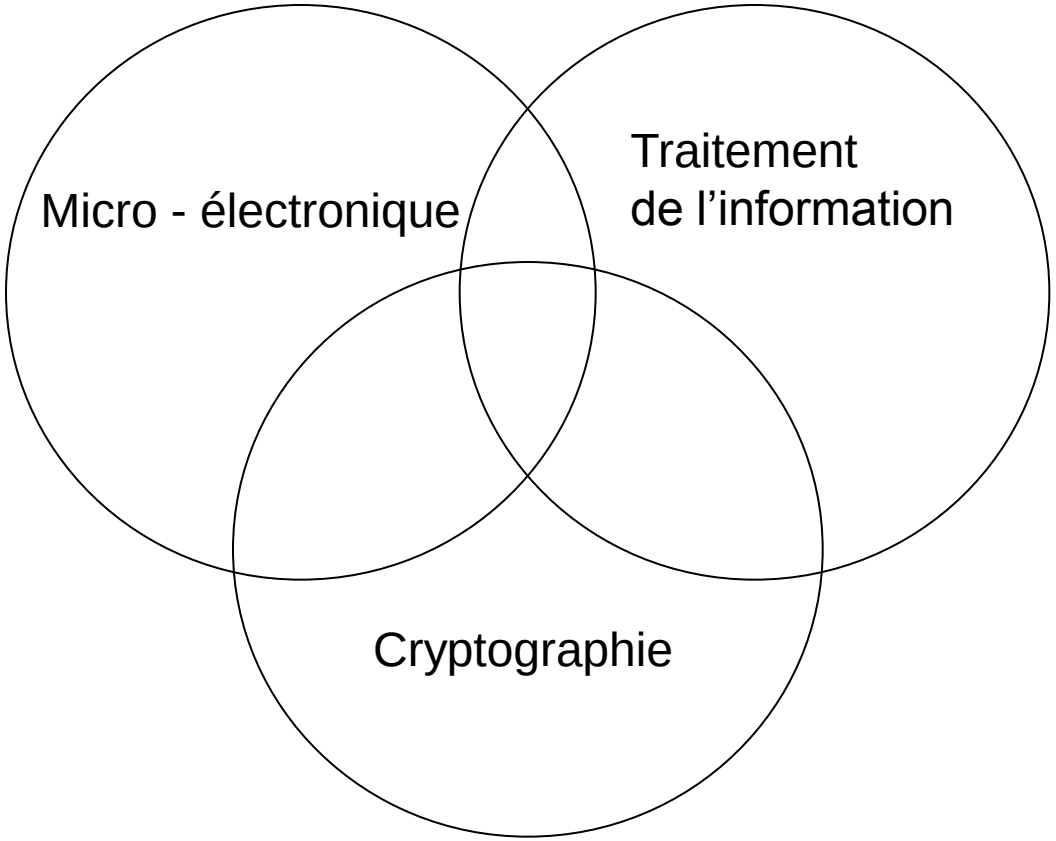
# 45 ans de loi de Moore

$2^{2N/3}$= 2.45/3=30



# L’écosystème

# L’écosystème de la *carte à puce*



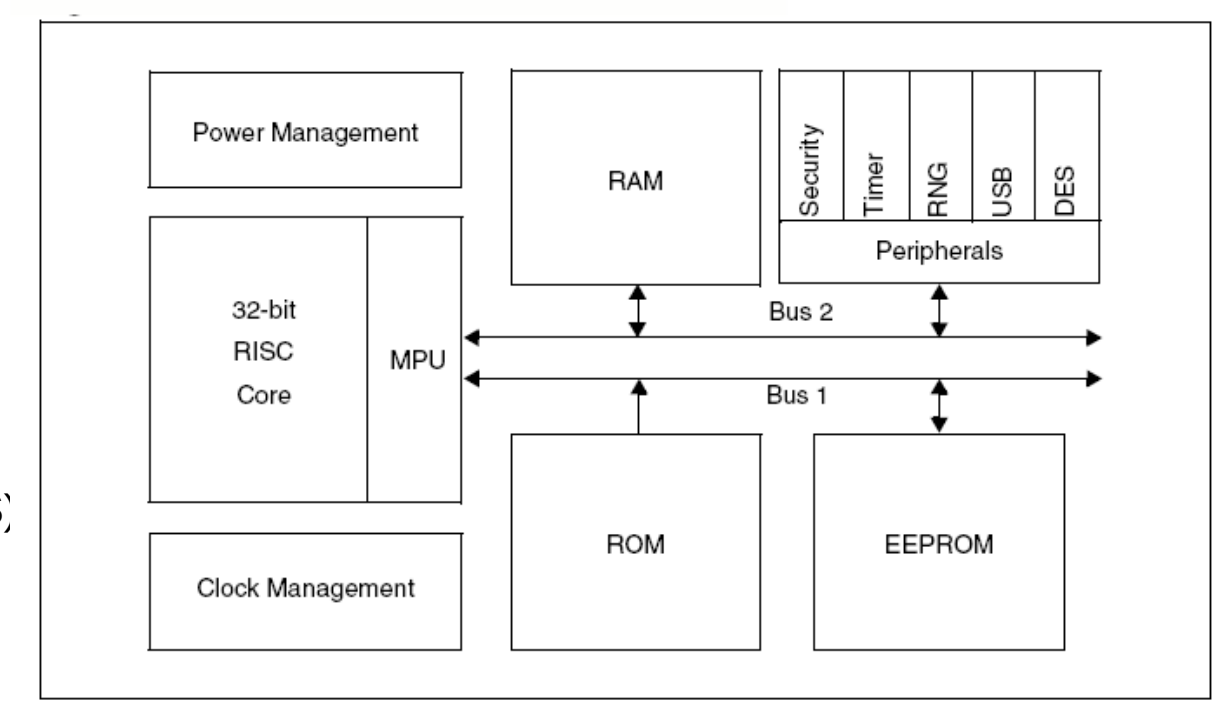
# Comment ça marche ?

- Du silicium sécurisé
  - Notion de *Tamper Resistant Device*
- Un système d'exploitation dédié
  - Gestion des contre-mesures
- Des implémentations d'algorithmes cryptographiques adaptées
  - Parades des attaques connues

# Des exemples de puces

# Le micro-contrôleur ST-22

- Non-Volatile Memory
- USB with Suspend mode
- Central Interrupt Controller
- Timer
- Random Number Generator
- Clock Manager
- Memory Protection Unit
- Sensors
- Encryption Coprocessor (DES)
- Security



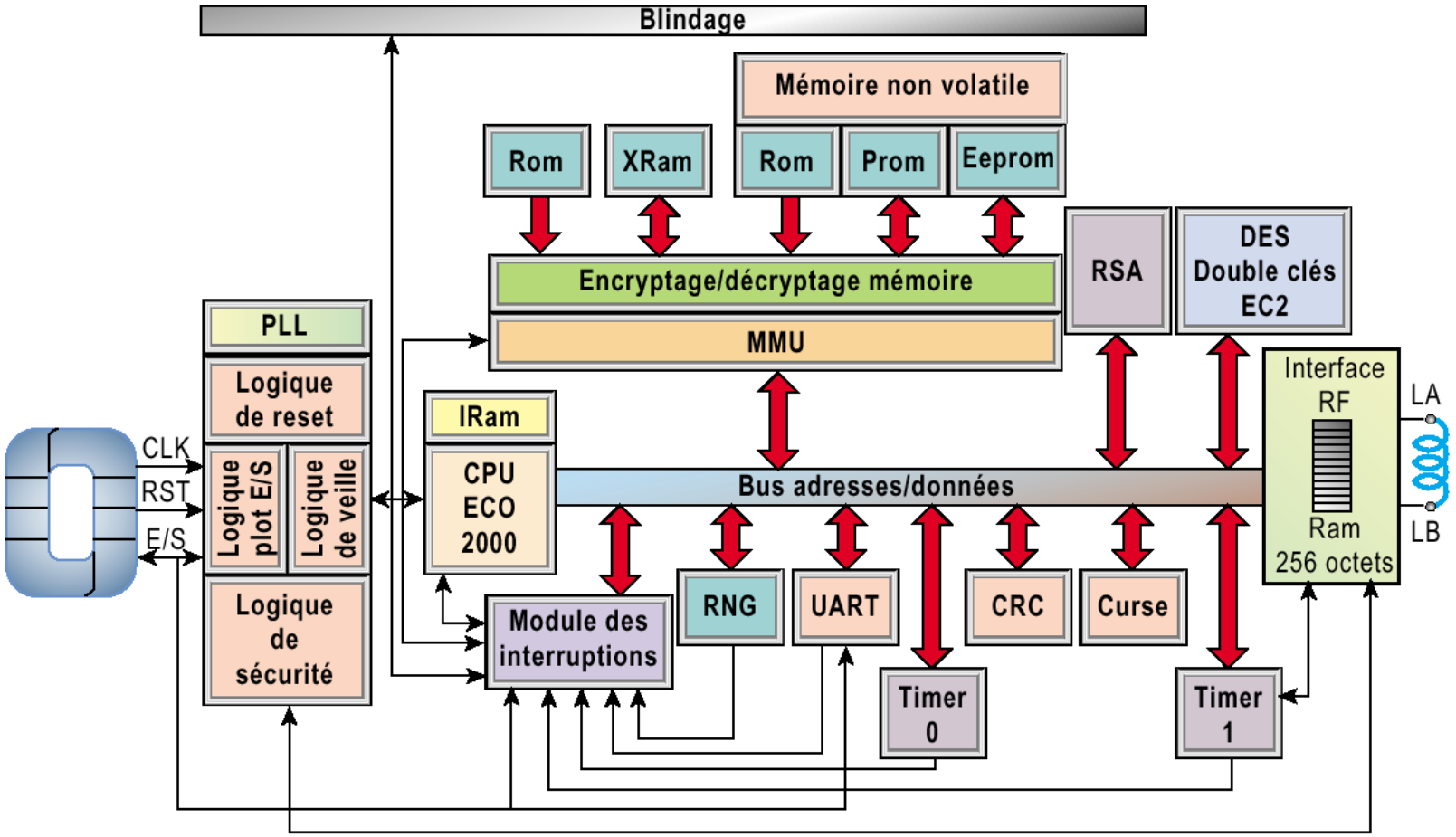
# Performances de la *CryptoLib* du microcontrôleur ST22

| Algorithm | Function | Time(1) |
|---|---|---|
| RSA 1024 bits | Signature with CRT | 79.0 ms |
| | Signature without CRT(2) | 242.0 ms |
| | Verification (e=0x10001) | 3.6 ms |
| RSA 2048 bits | Signature with CRT | 485.0 ms |
| | Signature without CRT | 1.7 s |
| | Verification (e=0x10001) | 11.0 ms |
| DES | Triple | 18 µs |
| | Single | 8 µs |
| TDES(3) | Triple (with keys loaded) | 1.8 us |
| SHA-1 | 512-bit Block | 194 µs |
| AES-128 | Encryption including subkey computation | 85 µs |
| Key generation | 1024 bits key<br>2048 bits key | 2.7 s<br>23.1 s |

1. Internal clock at 33 MHz

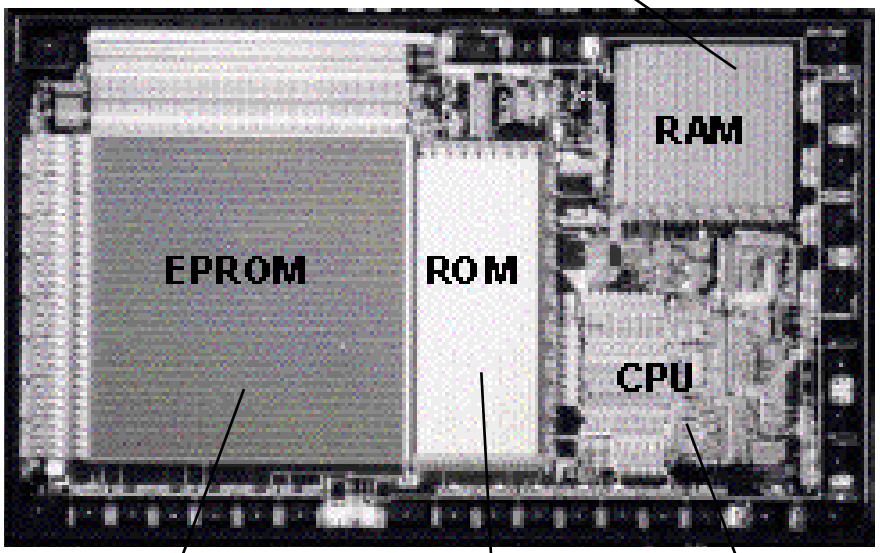
# SLE66CLX320P Infineon

*Le SLE66CLX320P d'Infineon réunit sur sa puce toutes les caractéristiques d'un microcontrôleur pour carte à puce: double interface avec contact et sans contact types A, B et FeliCa (premier produit compatible avec les trois types), CPU 16 bits, diverses mémoires, cryptoprocesseur et toutes les logiques de sécurité.*




*RNG : générateur de nombres aléatoires*

# La carte classique




52 B

2 KB

24 KB

8 bits
*6502 like*
Apple II µP

**80’**

16/32 bits

Crypto Processor

Random Number Generator


64 KB

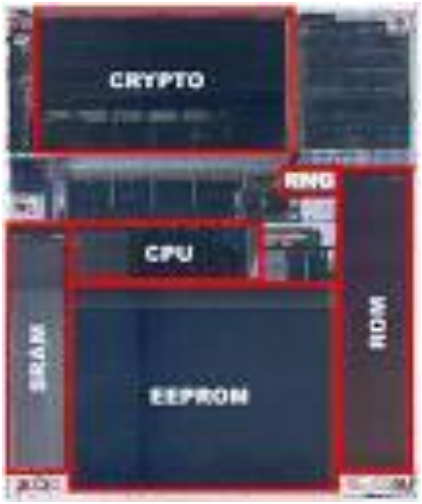
4 KB

**2000’**

**CPU**- 8bit data bus CPUs are dominating the microcontroller smart cards as in the industry globally.
Favorite 8bit CPUs are : 8051, 6805, HC05, AVR etc....
8bit CPU complexity is ranging from 1500 gates to 6000 gates.
Using state of the art 0.35 µm technology, 8bit CPU consumes 0.3-0.6 mm² of silicon.

32 bit CPU complexity is in the 100 000 gates range.

**EEPROM-** capacity is ranging today between 8kBytes and 32kBytes.
Using state of the art 0.35 µm technology, 32kByte EEPROM consumes 4-6 mm² of silicon.
EEPROM program/erase uses internally generated high voltage (15-20V) and low current (nA/cell) but takes about 2ms per cell.
To cope with the slow program/erase operation, 64Bytes are usually programmed at once in a Page mode mechanism. Main issue with the EEPROM functionality is its reliability expressed in data retention and program/erase cycling or endurance.
Access time to the data stored in the EEPROM is in the nanosecond range.




**SRAM** capacity ranges between 256 Bytes to 2k Bytes.
SRAM takes a lot of area on the IC since each memory cell consists of 6 transistors.
Using state of the art 0.35µm technology, 2kByte SRAM consumes 0.25-0.35mm² of silicon.

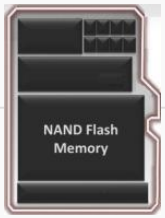
**ROM** capacity usually ranges between 8k Byte and 64k Bytes.
Since ROM unit memory cell is made of a single transistor, it is very dense.
Using state of the art 0.35 µm technology, a 64kByte ROM consumes 0.9-1.2mm² of silicon.
Access time to the Operating System microcode instruction is in the nanosecond range.
Multi application and needs for interoperability are requesting more complex operating system, and therefore larger ROM capacity (>64 k Bytes).
.

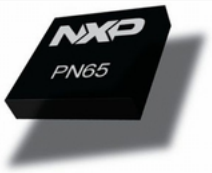
The current **Flash-EEPROM** memories are guaranteed for a data retention time of at least 10 years or at least 100.000 write/erase cycles. There is a considerable gain of writing time per memory access: to about 10µs with Flash, compare to 3-10 ms with normal EEPROM.

# Les marchés

# Les marchés


smart
VISA


NAND Flash
Memory


NXP
PN65


Secure elements shipments from 2010 to 2015
10k
8k
6k
4k
2k
0k
2010
2011
2012
2013
2014
2015
Others
Device Manufactures
Goverment
Payment Cards
Telecom

# Le Paiement Sécurisé

HSBC
MONEY GALLERY


*British Museum

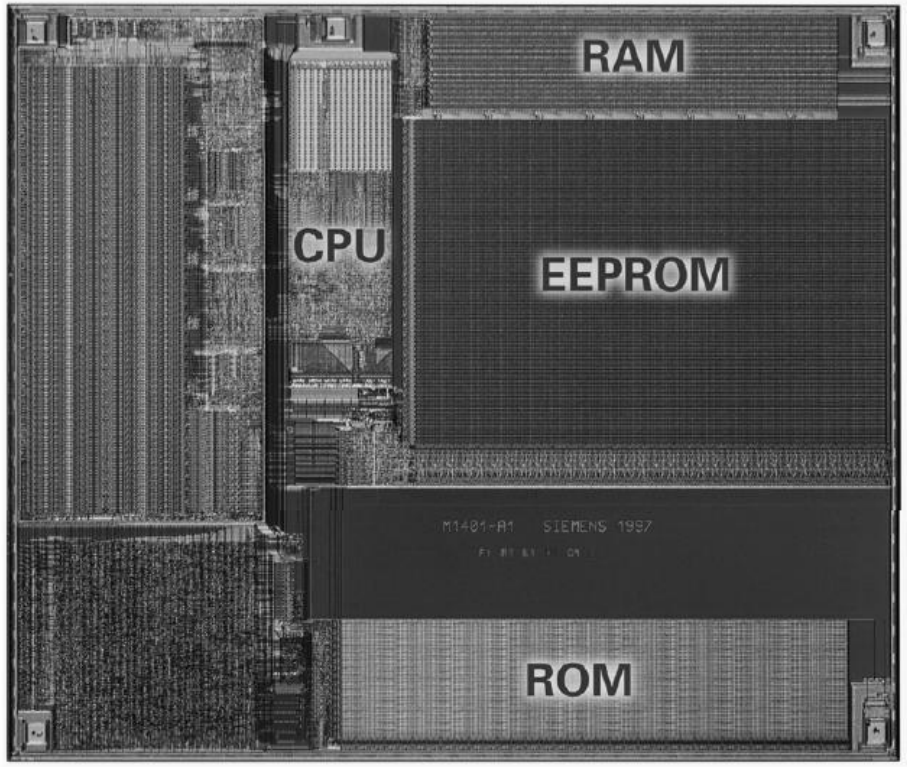

*Case 17, Section 4*
*Actual size 85 x 54mm*

## MONEY TODAY

The development of monetary technology continues today. A bank card permits the account holder to make payments by direct transfer and withdraw money from cash machines. Coins and notes now compete with a new generation of 'smart cards'. These contain microchips which store electronic cash to provide a fast, convenient way of paying.

# La carte SIM

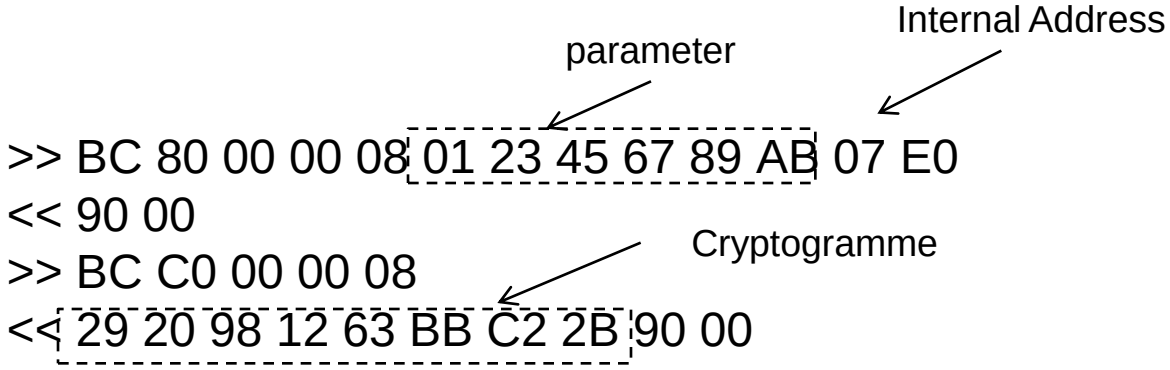
- **From the report of SIMEG#1 in January 1988**
  - **"A SIM is the physically secured module which contains the IMSI, an authentication algorithm, the authentication key and other (security related) information and functions. The basic function of the SIM is to authenticate the subscriber identity in order to prevent misuse of the MS (Mobile Station) and the network."**



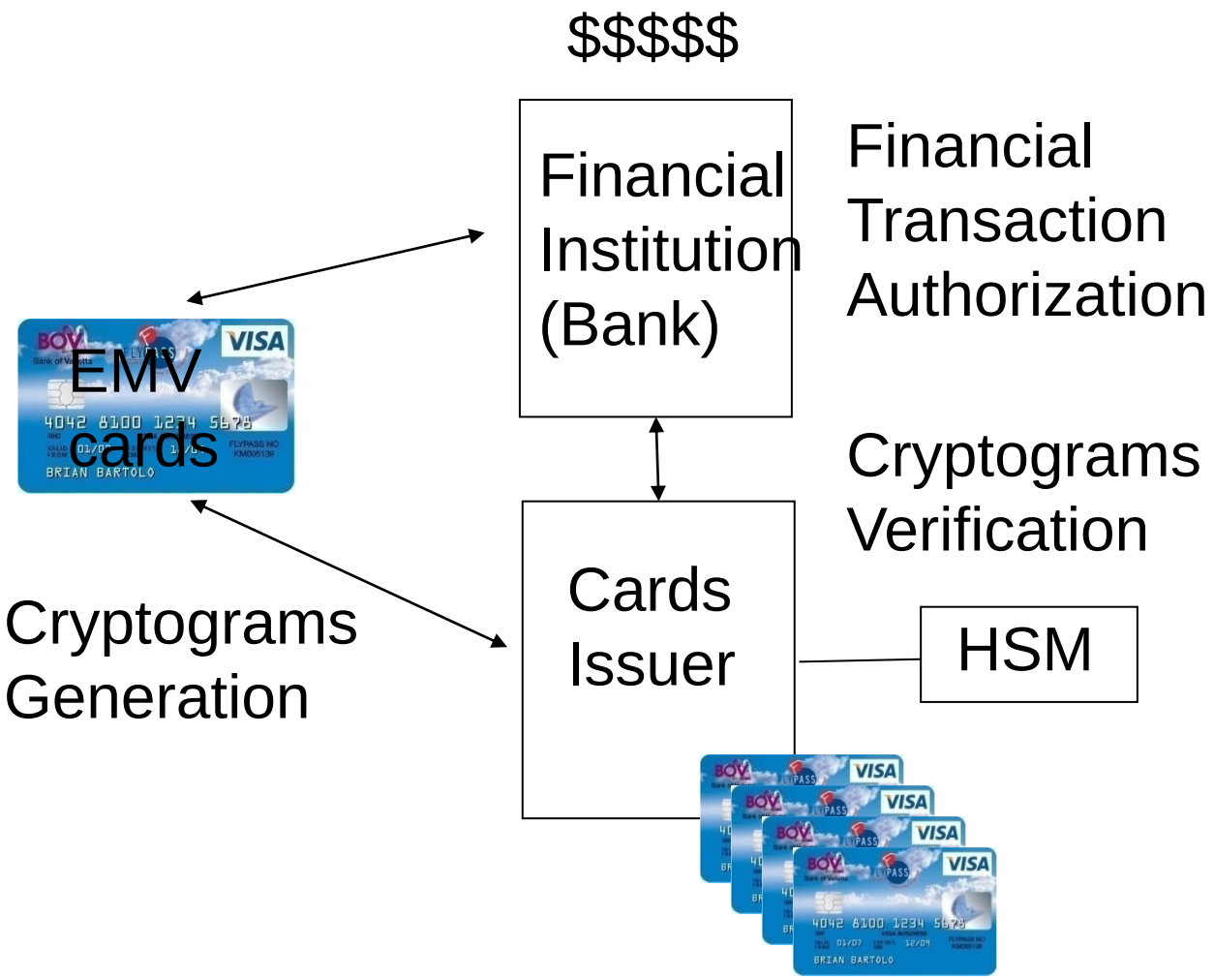

Siemens CHIP, 1997

# La Carte B0'

- Utilisée en France durant la période 1985-2000
- Une mémoire de 2 Ko divisée en plusieurs zones :
  - Accès non restreint  (lecture seulement)
  - Accès protégé par PIN code (lecture écriture)
  - Accès privé (administrateur seulement)
- Le contenu de la zone publique est signé avec une clé RSA (authentification statique)
- Les paramètres de transactions sont mémorisés dans la carte (date, montant)
- La fonction TELEPASS dont l'accès est protégé par le PIN code génère un cryptogramme basé sur l'algorithme 3xDES.



# Le modèle EMV : EuroCard, MasterCard, Visa



# Les Cartes EMV

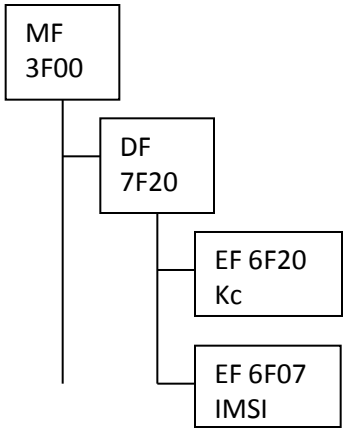
- Logent de multiples applications EMV
- Chaque application EMV fournit un ensemble de procédures et de données
- Les données sont organisées sous formes de fichiers comportant des listes d'enregistrements
  - Les enregistrements contiennent un ensemble d'objets ASN.1 (Data Object, DO)
- Quelques données (Data Objects)
  - Primary Account Number (PAN)
    - Le numéro de la carte
  - Signed Static Application Data (SSAD)
    - Une signature (à l'aide de la clé privée de l'ISSUER) des informations stockées dans la carte.
- Quelques procédures
  - DDA, Dynamic Data Authentication, chiffrement d'un nombre de 32 bits avec la clé privée de la carte EMV
  - Génération de Cryptogramme, basé sur l'algorithme 3xDES avec des paramètres d'entrée tels que montant de la transaction et date
    - ARQC, Authorization Request Cryptogram (ARQC), début d'une transaction EMV.
    - AAC, Application Authentication Cryptogram, fin d'une transaction EMV

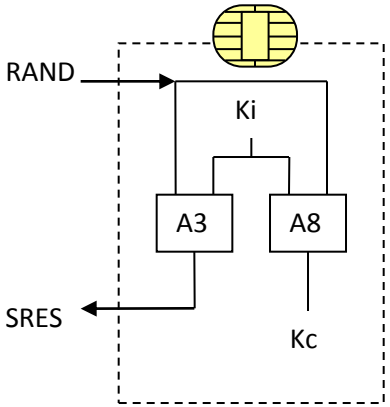
# Exemple: ARQC

- >> 80AE8000 1D
  - 00 00 00 00 00 00, the transaction amount
  - 00 00 00 00 00 00, the cash back
  - 00 00, the national code of the payment terminal
  - 80 00 00 00 00, the terminal verification result
  - 00 00, the transaction currency code
  - 01 01 01, the transaction date
  - 00, the type of transaction
  - 12 34 56 78, a four bytes random value
- << 77 1E
  - 9F 27 01 80, Cryptogram Information Data
  - 9F 36 02 00 18, Application Transaction Counter (ATC)
  - 9F 26 08 80 29 D3 A0 BB 2A 5E 60, Application Cryptogram
  - 9F 10 07 06 7B 0A 03 A4 A0 00, Issuer Application data

# La carte SIM

- L'information est organisée en répertoires et fichiers
- Quelques données
  - IMSI
  - Deux répertoires téléphoniques
  - Un fichier SMS
- Quelques procédures
  - RUN_GSM_ALGORITHM, calcule l'algorithme A3/A8



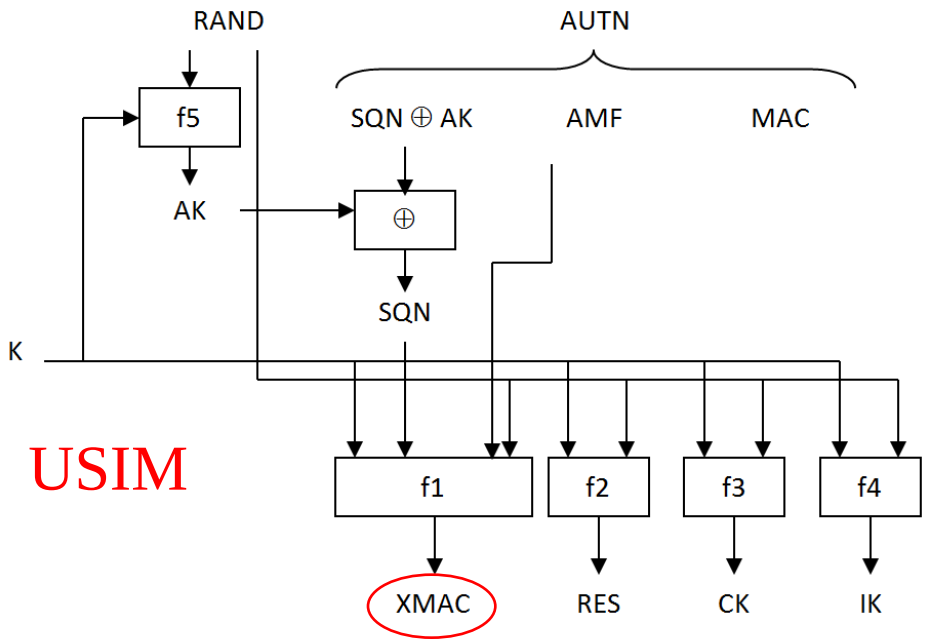



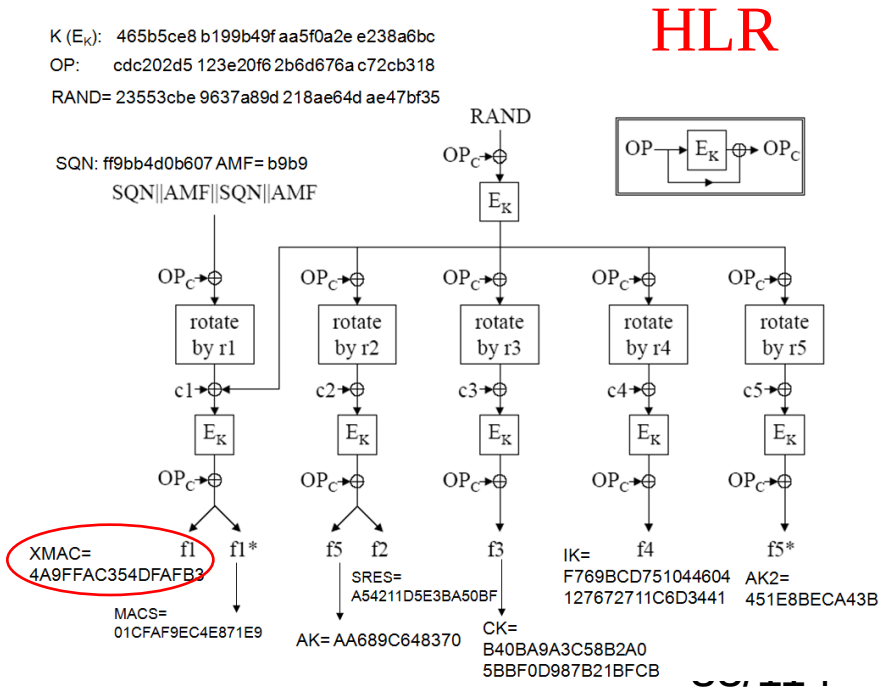

// Run_Gsm_Algorithm(RAND) **RAND**
>> A0 88 00 00 10 01 23 45 67 89 AB CD EF 01 23 45 67 89 AB CD EF
<< 9F 0C
>> A0 C0 00 00 0C
<< FE 67 7C 9D B8 DD F1 B1 DE 27 18 00 90 00

**SRES** **KC**

# La carte USIM

- Le module UICC stocke au moins une  application USIM
  - Le fichier EF_DIR contient la liste des applications USIM
- Au moins deux applications peuvent être activées simultanément (notion de canaux logiques
  - L'index de l'application est indiqué dans les deux derniers bits de l'octet CLA
- L'algorithme d'authentification AKA est réalisé par la commande AUTHENTICATE  (INS=88) command
- Exemple
- >> 00 88 00 00 20 RAND || AUTN
- <<  DB 28 SRES || CK || IK  9000
  - AUTN:= SQN ⊕ AK || AMF || XMAC



# Le Passeport Electronique

- Le passeport électronique est décrit par les normes ICAO 9303 (part 1,2,3)
- L'application passeport gère un ensemble de fichiers
  - EF.COM, la liste des fichiers stockés dans le passeport
  - EF.DG1, la copie des informations imprimées dans le MRZ (Machine Readable Zone)
  - EF.DG2, contient une photo biométrique du propriétaire du passeport
  - EF.DG3 contient les empreintes digitales. Le contenu est chiffré ou protégé par une procédure d'authentification nommée EAC (Extended Access Control)
  - EF.DG11 diverses informations additives sur le propriétaire du passeport
  - EF.DG12 diverses informations additives sur le passeport
  - EF.DG15 stocke la clé publique (RSA) optionnelle utilisée par le mode AA (*Active Authentication)*
  - EF.SOD contient la signature du contenu du passeport.

# Le Passeport Electronique

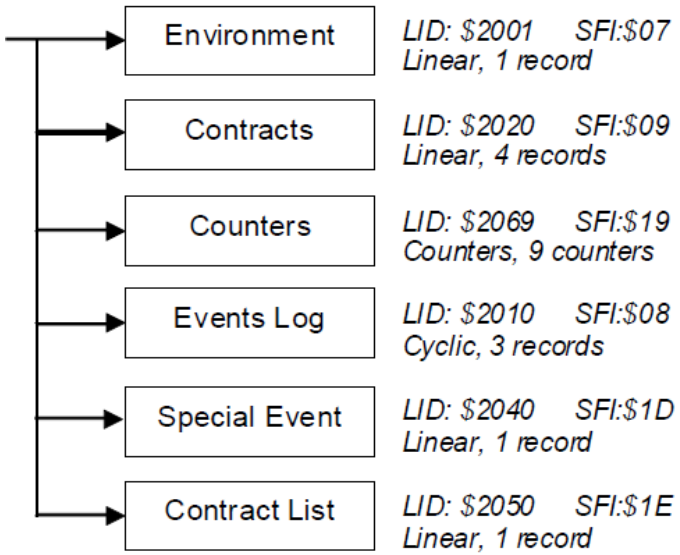
- L'accès aux données du passeport est protégé par trois types de procédures
  - **BAC**, *Basic Access Control*. Une clé maître (kseed) est déduite du contenu MRZ. Deux clés sont calculées à partir de kseed et de deux valeurs aléatoires. Elles sont utilisées pour le chiffrement et l'intégrité des données échangées entre le passeport et le lecteur sans contact.
  - **AA**, *Active Authentication*. Une clé RSA privée stockée dans le passeport prouve l'authenticité du passeport (mesure anti-clonage).
  - **EAC**, *Extended Authentication Access*. Une procédure réalisant une mutuelle authentification (Diffie-Hellman sur courbe elliptique) entre le passeport et le lecteur sans contact..

ZONE MRZ

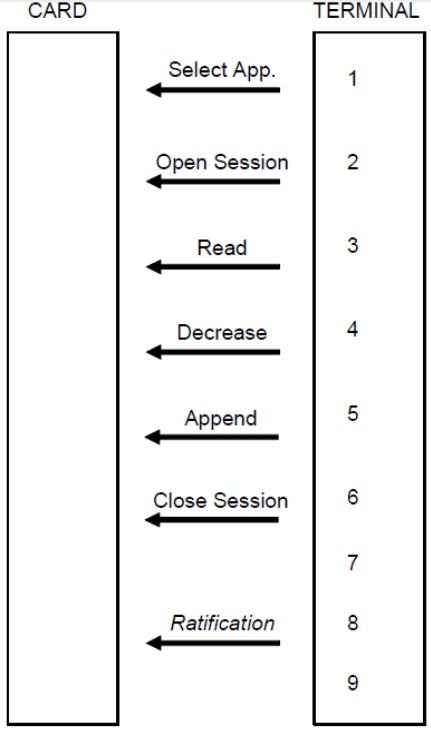
# La carte Navigo

- Conforme à la norme Calypso (www.calypsostandard.net ), semi propriétaire
  - Application AID:  "1TIC.ICA AID"
    - Select AID 00A40400 0C 315449432E49434120414944
- Un système de gestion de fichier dont l'accès en écriture est sécurisé par un protocole propriétaire
- L'information est enregistrée selon la norme expérimentale française Intercode, " Règle d'interopérabilité pour le codage des données billettiques (révision II, 30/09/2003) ".
- Au niveau du Master File (MF, 3F00), deux fichiers stockent des informations sur le type de puce (ICC) ou sur le porteur de la carte (ID)
- EF_ICC= 0002 (lecture), EF_ID=0003 (auth. Required)
  - Select  MF  94A40000 02 3F00
  - Select  EF_ICC  94A40000 02 0002
- Le répertoire Calypso (DF-Calypso =2000) loggent les principaux fichiers
  - EF_Env =2001, Environment (1 record), EF_Contracts =2020 (4 records) , EF_Events 2010 (3 records).

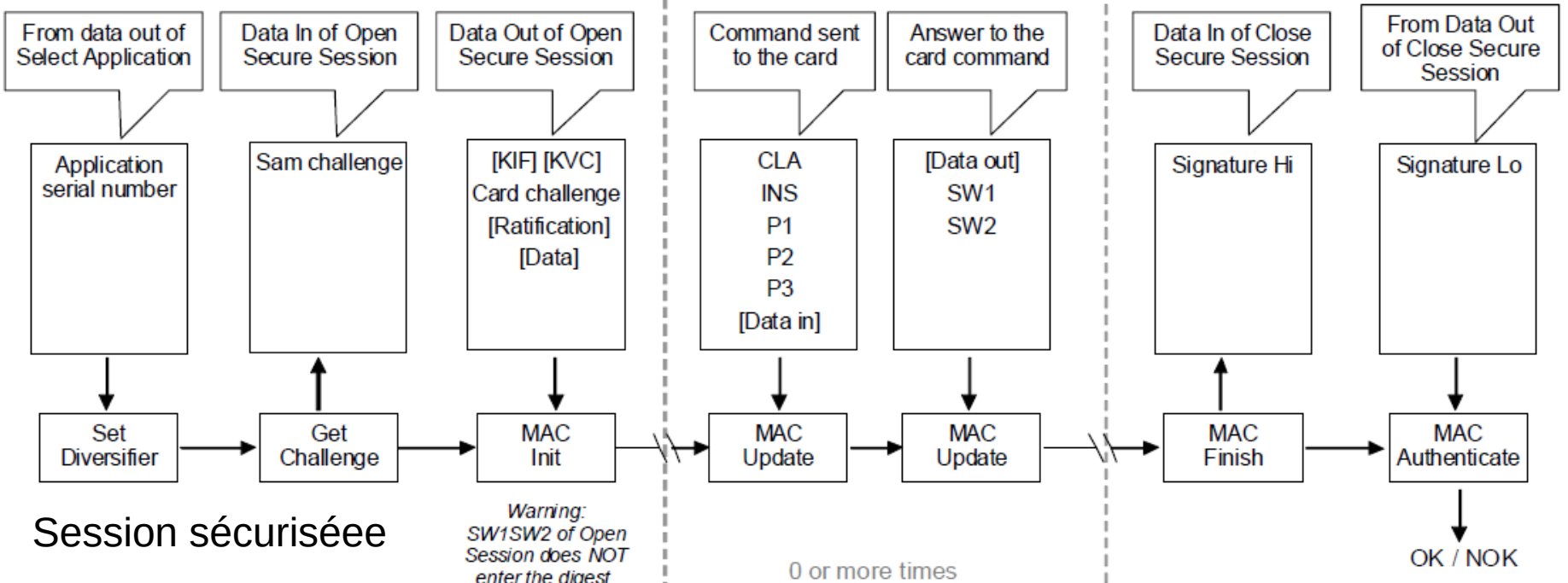
# Divers Calypso




Fichiers Calypso




Exemple de session en mode “débit”




Session sécuriséee

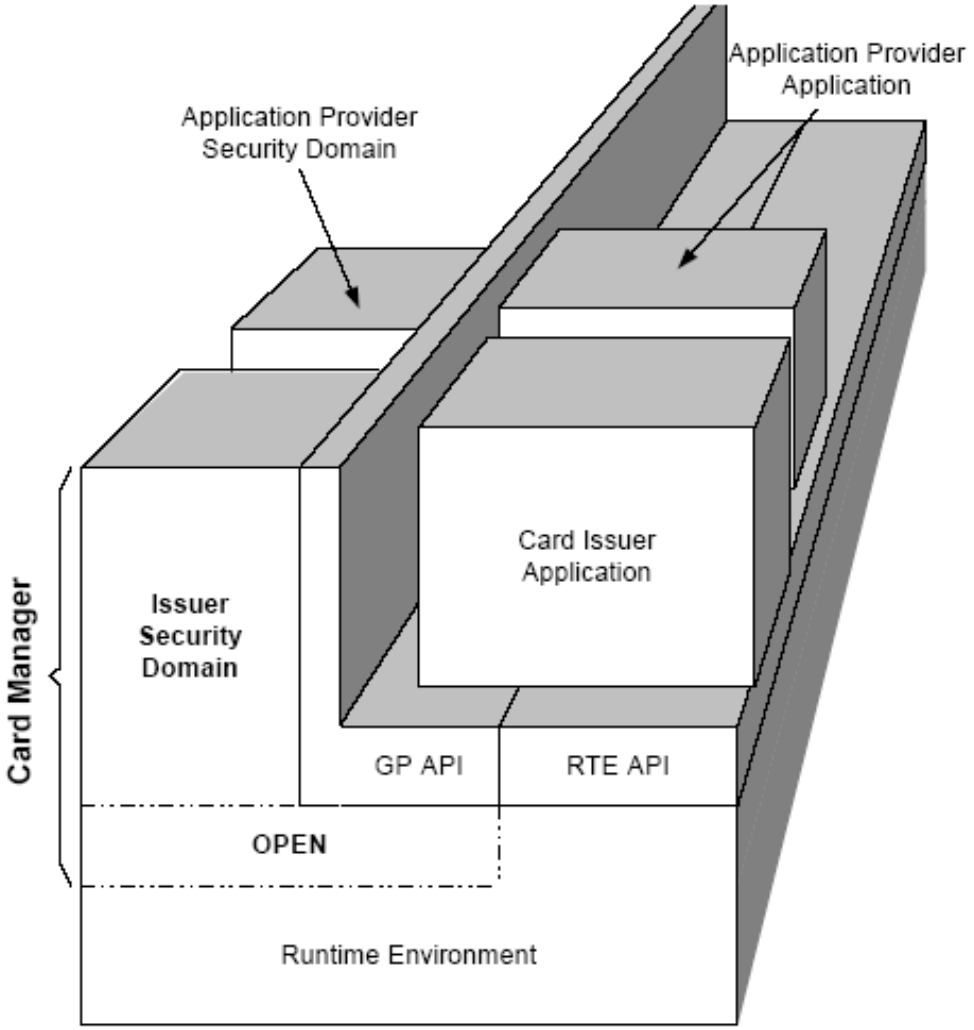
# Exemple de dialogue

- ATR: 3B8F8001805A0803040002002573BD1182900031
- Selection de l'application Calypso
  - Tx: 00A40400 0C315449432E49434120414944
  - Rx: 6F 22 84 08 31 54 49 43 2E 49 43 41 A5 16 BF 0C 13 C7 08 00 00 00 00 25 73 BD 11 53 07 03 08 03 04 00 02 00 90 00
- Select MF
  - Tx: 94A40000 02 3F00
  - Rx: 85 17 00 01 00 00 00 10 10 00 00 01 07 00 00 00 15 15 15 00 00 00 00 00 00 90 00
- Select EF_ICC
  - Tx: 94A40000 02 0002
  - Rx: 85 17 02 04 02 1D 01 1F 00 10 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 90 00
- Select DF_Calypso
  - Tx: 94A40000 02 2000
  - Rx: 85 17 00 02 00 00 00 10 10 00 00 01 07 00 00 00 15 15 15 00 00 00 00 00 00 90 00
- Select EF_Enr
  - Tx: 94A40000 02 2001
  - Rx: 85 17 07 04 02 20 01 1F 10 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 90 00
- Read Enr, Record#1
  - Tx: 94 B2 01 04 20
  - Rx: 24 B9 28 48 08 04 86 E1 6F B0 00 12 20 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 90 00

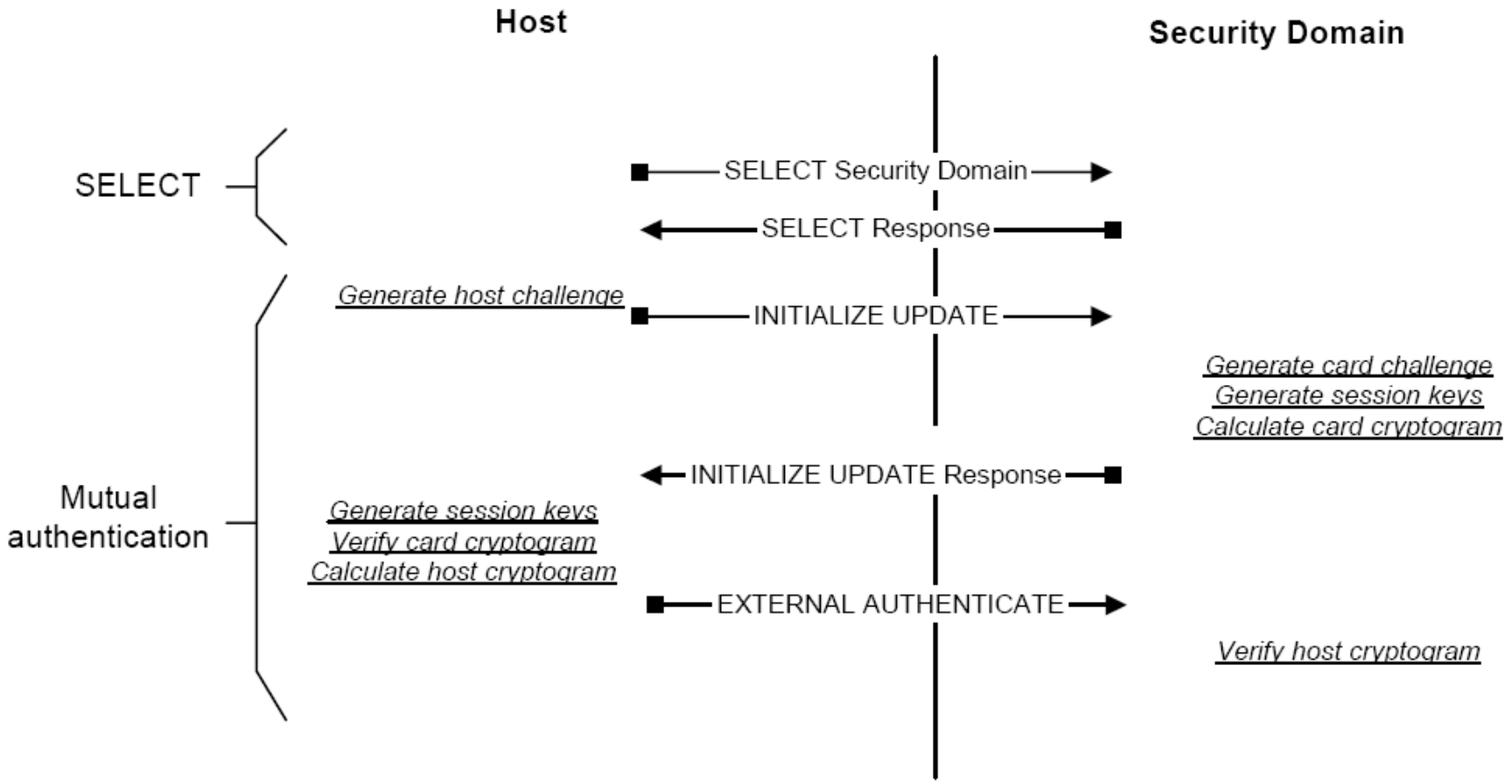
# Les normes Global Platform (GP)

# Issuer Security Domain (ISD)

- Une application qui gère le chargement et l'activation de logiciels embarqués
- Une application embarquées comporte trois états :
  - INSTALLED
  - SELECTABLE
  - LOCKED



# Mutuelle Authentification: clés de chargement



# SCP 01

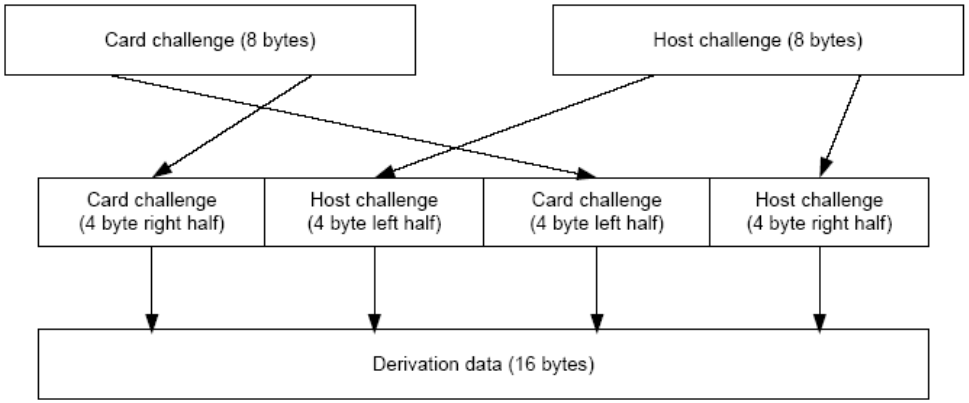



**Figure D-3: Session Key - Step 1 - Generate Derivation data**

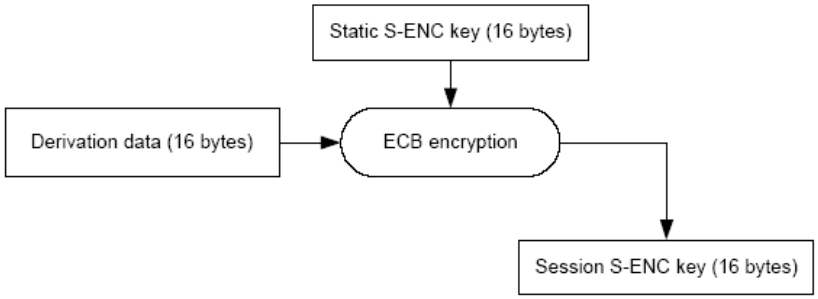



**Figure D-4: Session Key - Step 2 - Create S-ENC Session Key**

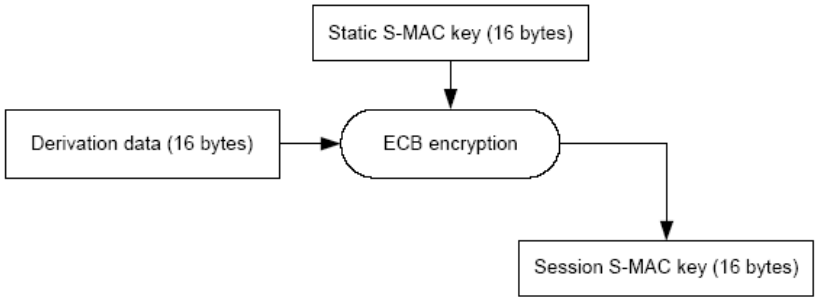



**Figure D-5: Session Key – Step 3 - Create S-MAC Session Key**

| Clé | Constante |
|---|---|
| C-MAC | 0101 |
| R-MAC | 0102 |
| S-ENC | 0182 |
| DEK | 0181 |

# SCP02

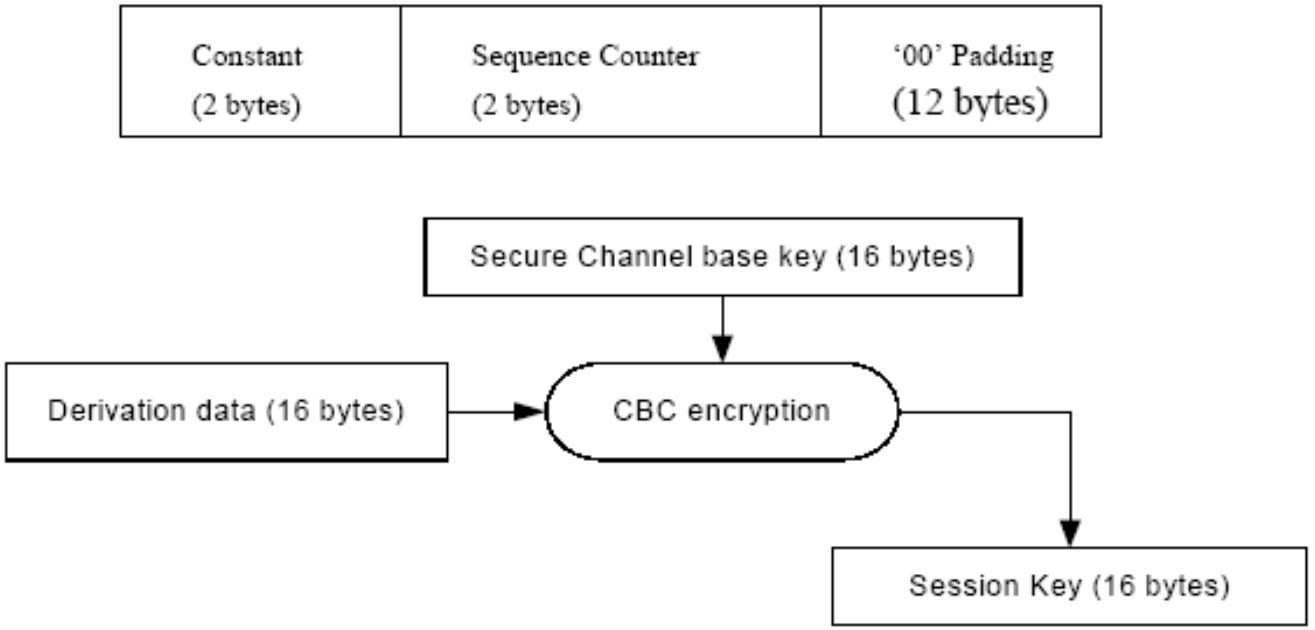



**Figure E-2: Create Secure Channel Session Key from the Base Key**

# Principales Commandes

- DELETE. Destruction d'un objet tel que application ou clé.
- GET DATA. Lecture d'une information identifiée par un TAG, plus particulièrement une clé.
- GET STATUS, Lecture d'informations telles que liste d'applications, liste de domaine de sécurité, ou état d'un cycle de vie géré par un domaine de sécurité.
- INSTALL. Commande adressée à un domaine de sécurité pour gérer les différentes étapes de l'installation d'une application
- LOAD. Chargement d'un fichier. Cette commande est généralement précédée de l'APDU INSTALL [for load] qui indique des options de chargement.
- PUT KEY. Création mise à jour ou destruction de clés
- SELECT. Sélection d'une application
- SET STATUS. Modification de l'état d'un cycle de vie
- STORE DATA. Transfert de données vers une application ou un domaine de sécurité

# Le modèle VISA*

KMC-ID (6B) CSN (Chip Serial Number, 4B)

KMC (DES Master Key for Personalization Session Keys)

*EMV Card Personalization Specification Version 1.1 July 2007

KEYDATA = $KMC_{ID}$ || CSN

KENC := DES3(KMC)[Six least significant bytes of the KEYDATA || F0 || 01 ] || DES3(KMC)[ Six least significant bytes of the KEYDATA || 0F || 01].

KMAC := DES3(KMC)[ Six least significant bytes of the KEYDATA || 'F0' || 02 ] || DES3(KMC)[ Six least significant bytes of the KEYDATA || '0F' || 02].

KDEK := DES3(KMC)[ Six least significant bytes of the KEYDATA || 'F0' || 03 ] || DES3(KMC)[ Six least significant bytes of the KEYDATA || '0F' || 03]

| Session Key | IC Card Key | Derivation Data |
|---|---|---|
| $SKU_{ENC}$ | $K_{ENC}$ | '0182' \|\| sequence counter \|\| '00000000000000000000' |
| $SKU_{MAC}$ | $K_{MAC}$ | '0101' \|\| sequence counter \|\| '00000000000000000000' |
| $SKU_{DEK}$ | $K_{DEK}$ | '0181' \|\| sequence counter \|\| '00000000000000000000' |

# GSM 3.48

# Au sujet de GSM 3.48

- Le standard 3GPP TS 03.48 "Digital cellular telecommunications system (Phase 2+);Security mechanisms for SIM application toolkit;Stage 2", définit un mécanisme de transport sécurisé pour des commandes APDU.
- Les messages TS 03.48 contiennent des requêtes et des réponses ISO7816, et sont transportés dans des SMS de type SMS-DELIVER dans le sens serveur vers carte SIM, et SMS-SUBMIT en sens inverse.
- Le format des SMS est détaillé par la norme GSM 03.40, "Technical realization of the Short Message Service (SMS) Point-to-Point (PP)".
- Un SMS comporte un entête et un contenu, ce dernier se divise un préfix 03.48 et une charge : une requête ou une réponse ISO 7816, avec ou sans chiffrement.

| **ISO7816-4** |
| --- |
| GSM 3.48 |
| SMS GSM 3.40 |

# Paquet de Commande 1/2

// Commande
80C20000 77 // ENVELOPE (INS=C2), length= 77
D1 75 // SMS-PP download TAG=D1
82 02 83 81 // Source ME(83) - Dest SIM(81)
86 02 80 F0
8B 6B // TAG= 8B, SMS-TPDU, length=6B
60 // TP-MTI=00, SMS DELIVER, TP-UDHI=1, TP-SRI=1
01 80 F0 // TP-OA (2-12 octets)
7F // TP-PID
F6 // TP-DCS
00 00 00 00 00 00 00 // TP-SCTS
5D // TP-UDL
02 70 00 // UDHL=02 IEI=70 IEIDL=00
0058 // CPL

# Paquet de commande 2/2

15 // CHL
06 // SL = Cryptographic Checksum (CC) , Ciphering
19 // RL= PoR ciphered, CC applied to PoR, PoR required
15 // KIC, KeyIndex=1, TripleDES, 2 keys
15 // KID, KeyIndex=1, TripleDES, 2 keys, CC= ISO 9797 padding method 1
00 00 00 // TAR
// Zone chiffrée
655C7380462D62252E87CB2F4A8FD407CED82C9BBA8945C23AD03A9CF90
A95FBAC8572A53E38A75594F89B5D8BE025938CC2270E186ECF53A772481
BABEA1687A111FDCBA047CD4D1F5C029D41E200
// Valeur déchiffrée
000000009C // CNTR
05 // PCNTR, 5 octets de bourrage
A7A4B0B5A8A273A5 // RC/CC/DS
80E6 0C00 38 // APDU ISO7816
05A00000003010A0000000300002FFFFFFFF89313233001 0A0000000300002
FFFFFFFF8931323300010 00CEF08C8020000C7020000C90000
0000000000 // 5 octets de bourrage

# Paquet de Réponse

6121
00C00000 21 // GSM 3.48
02 71 00 // UDHL=02 IEI=70 IEIDL=00
001C // RPL
12 // RHL
00 00 00 // TAR
// Zone chiffrée
6CC7F00F6632C2E31E40B12539FD8F6F65024D824C32F295

// Valeur déchiffrée
000000009C // CNTR
06 // PCNTR, 6 octets de bourrage
00 // Statut de la réponse = OK
BD1D9F0BC5499B64 // RC/CC/DS
016101 // Réponse ISO7816
000000000000 // 6 octets de bourrage

# Mode de calcul du CMAC

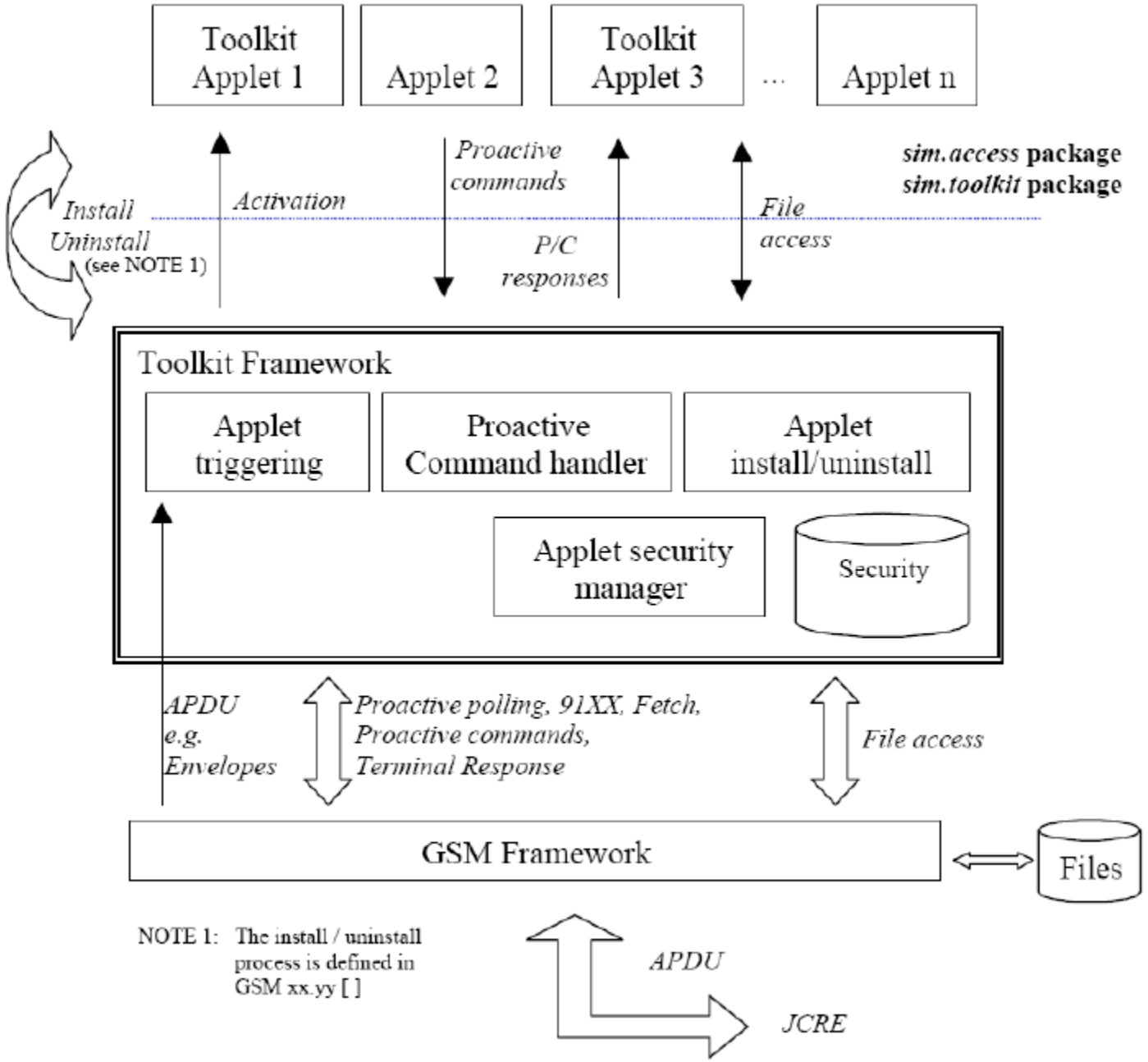
- Pour un paquet de commande le calcul s’applique sur
  - CPL || CHL|| SPI || KIC || KID || TAR || CNTR || PCNTR || SD,
- c'est-à-dire que l’élément RC/CC/DS n’est pas pris en compte.
- Pour un paquet de réponse le calcul s’applique sur
  - UDHL || IEI || IEIDL || RPL || RHL || TAR || CNTR || PCNTR || SD,
- c'est-à-dire que l’élément RC/CC/DS n’est pas pris en compte.

# SIM Tool Kit
# GSM 3.19

# GSM 3.19

- La norme GSM 3.19 introduit la notion de Toolkit Applet dont les services sont hérités des paquetages sim.toolkit et sim.access.
  - Le premier permet de s'enregistrer à des évènements particuliers et de produire des commandes proactives,
  - le deuxième offre des facilités pour accéder au système de fichier de la carte SIM depuis un applet embarqué JAVA.

# GSM 3.19


Toolkit Applet 1
Applet 2
Toolkit Applet 3
...
Applet n
Install
Uninstall
(see NOTE 1)
Activation
Proactive commands
P/C responses
File access
sim.access package
sim.toolkit package
Toolkit Framework
Applet triggering
Proactive Command handler
Applet install/uninstall
Applet security manager
Security
APDU e.g. Envelopes
Proactive polling, 91XX, Fetch, Proactive commands, Terminal Response
File access
GSM Framework
Files
NOTE 1: The install / uninstall process is defined in GSM xx.yy [ ]
APDU
JCRE

# SIM ACCESS

- L'accès aux services de sim.access s'effectue à l'aide d'un objet JAVA SIMView dont une instance est obtenue par le constructeur de l'applet embarqué
  - SIMView myView= SIMSystem.getTheSIMView();
- La classe SIMView réalise des opérations classiques (sélection, lecture, écriture) avec le système de fichier de la SIM, telles que par exemple
  - select(short, byte[], short, short), sélectionne au répertoire DF ou un fichier EF et retourne son entête FCI
  - readBinary(short, byte[], short, short), lecture des données d'un fichier linéaire
  - readRecord(short, byte, short, byte[], short, short) lecture d'un fichier à enregistrements
  - updateBinary(short, byte[], short, short) écriture dans un fichier linéaire
  - updateRecord(short, byte, short, byte[], short, short) écriture dans un fichier à enregistrement

# SIM Tool Kit

- Un applet SIM Tool Kit implémente une interface *ToolkitInterface*, c'est-à-dire qu'il contient obligatoirement une méthode
  - public void processToolkit(byte event)
- qui réalise le traitement des événements notifiés par le terminal à la SIM au moyen d'APDUs ENVELOPE.
- Le constructeur d'un tel applet s'enregistre pour les événements qu'il désire traiter à l'aide d'un objet *ToolkitRegistry*,
  - ToolkitRegistry reg = ToolkitRegistry.getEntry();
- // register to the EVENT_UNFORMATTED_SMS_PP_ENV reg.setEvent(EVENT_UNFORMATTED_SMS_PP_ENV);
- Dès lors les événements sollicités seront routés par le système d'exploitation de la SIM vers la méthode processToolkit(*byte event*).

# GSM 3.19

// Select DF GSM (7F20)
A0 A4 00 00 02 7F 20
// 9F 1A
A0 C0 00 00 1A
// 90 00

// Verify CHV1 "1111"
A0 20 00 01 08 31 31 31 31 FF FF FF FF
// 90 00

// Select EF_Phase
A0 A4 00 00 02 6F AE
// 9F 0F
A0 C0 00 00 0F
// 00 00 00 01 6F AE 04 00 04 FF 44 01 01 00 00
90 00

A0 B0 00 00 01
// 02 90 00  Phase_2  03<=> phase 2 and
PROFILE DOWNLOAD REQUIRED (2+)

// Terminal Profile
A0 10 00 00 04 EF FF FF FF

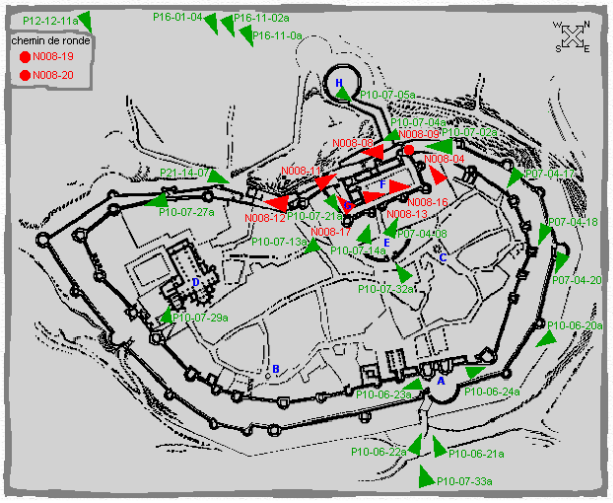
// UNFORMATTED
A0C20000 1F D1 1D 82 02 83 81 8B 17 04 00 A1 7F
F6 99 01 01 01 02 03 40 0A 01 14 00 06 0D 20 00 00
00 00
// 91 76
A0 12 00 00 76

// D0 68
// 81 03 01 13 00
// 82 02 81 83
// 0B 5D
// 01 A5 0B
// 91 21 43 65 87 90 F8 00 04 50 02 14 00 50 0D 80
// 00 00 00 46 16 03 01 00 41 01 00 00 3D 03 01 3F
// AA 2B 6A 08 BD D2 85 B4 3D 1F 3B C9 71 5F C9 F8
// 5F C4 53 FE 58 F3 A9 E0 7F F3 97 CD 65 39 22 00
// 00 16 00 04 00 05 00 0A 00 09 00 64 00 62 00 03
// 00 06 00 13 00 12 00 63 01 00 90 00

// terminal Response
A0 14 00 00 0C 81 03 01 13 01 82 02 82 81 83 01 00
// 9000

# Au sujet de la Sécurité

# La sécurité est une construction (design)

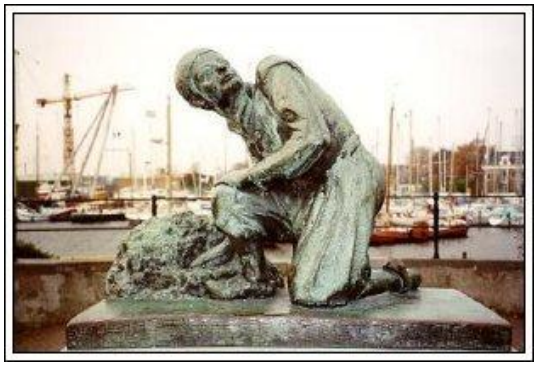



2 lignes de remparts
1 bastion

*La ville de Carcasonne

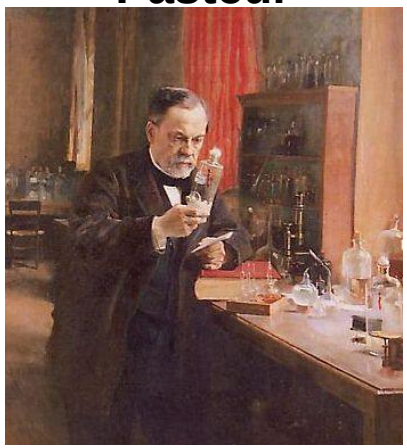
# Attaque et Défense

**Hans Brinker**



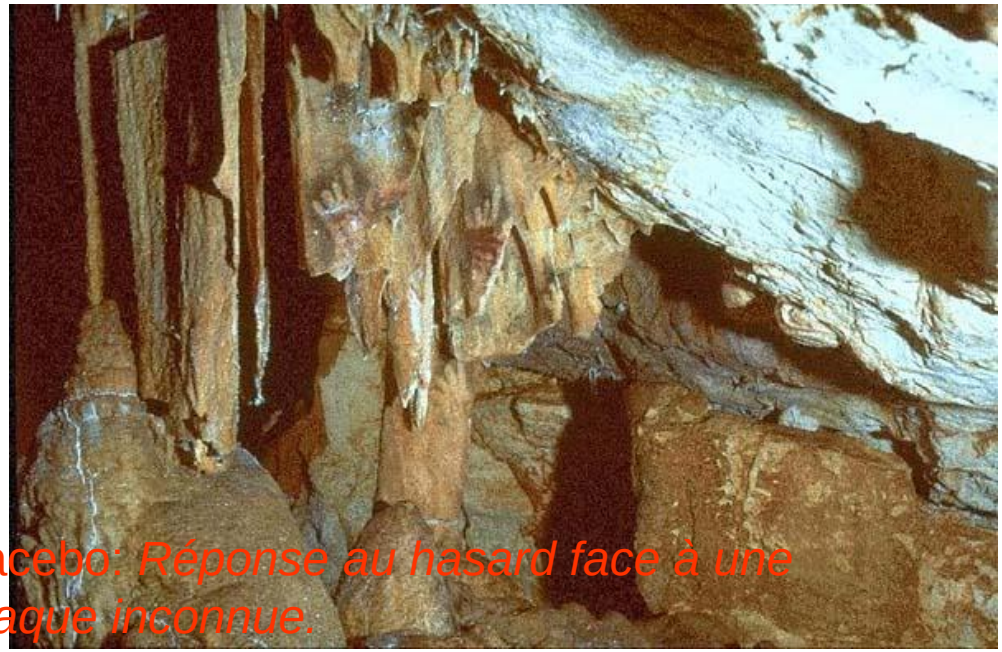
Défense immunitaire:
*Réponse efficace à une attaque inconnue*

**Pasteur**



Vaccin: *Réponse efficace à une attaque connue*

**Grotte de Cosquer**



Placebo: *Réponse au hasard face à une attaque inconnue.*

**Attaque** ⟺ **Contre-mesure**

**-Connue** **- Efficace**

**-Inconnue** **- Aléatoire**

# Les équations de Maxwell sont-elles sécurisées ?

| | |
|---|---|
| $\mathrm{div}\ \vec{B} = 0$ | $\mathrm{div}\ \vec{E} = \frac{\rho}{\varepsilon_0}$ |
| $\overrightarrow{\mathrm{rot}}\ \vec{B} = \mu_0 \vec{j} + \varepsilon_0 \mu_0 \frac{\partial \vec{E}}{\partial t}$ | $\overrightarrow{\mathrm{rot}}\ \vec{E} = -\frac{\partial \vec{B}}{\partial t}$ |

# Courants de Foucault (*Ellis current*)

$$J_x = \left(\frac{1}{e}\right)^{(x/\delta)}$$

$$\delta = \frac{1}{\sqrt{\pi f \mu \sigma}}$$

**Jx = Current Density ($A/m^2$)**

**e = Base Natural Log**

**x = Distance Below Surface**

**δ = Standard Depth of Penetration**

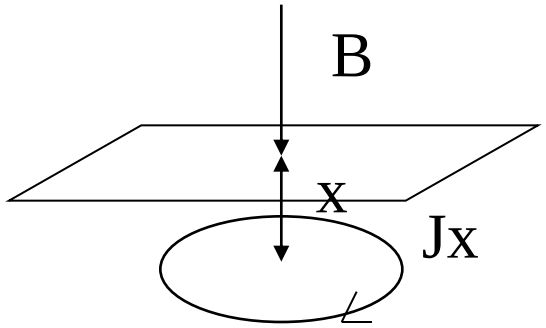
**δ = Standard Depth of Penetration (m)**

**π = 3.14**

**f = Test Frequency (Hz)**

**μ = Magnetic Permeability (Henry/m)**

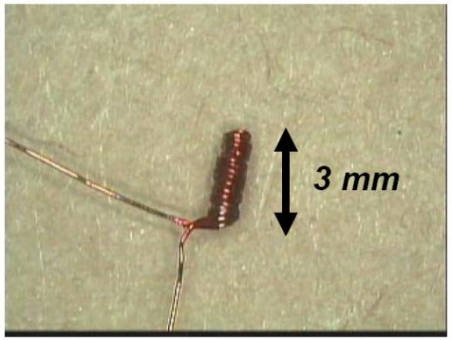
**σ = Electrical Conductivity (Siemens/m)**



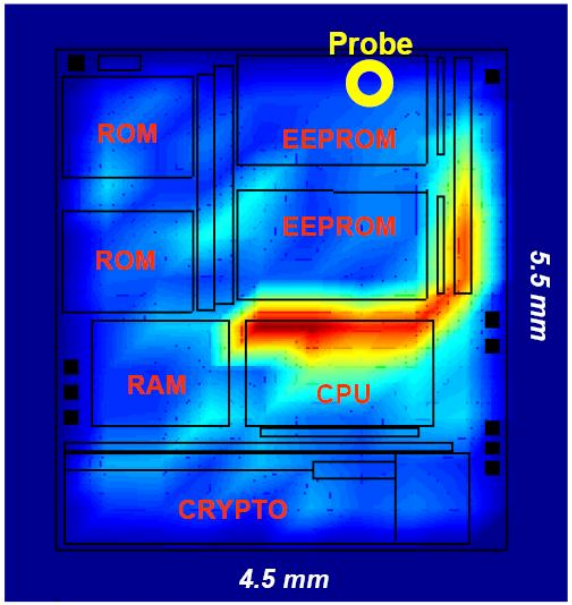
# Attaque par courant de Foucault d'une cellule SRAM

Il est possible de modifier l'état d'une cellule mémoire SRAM par courant de Foucault

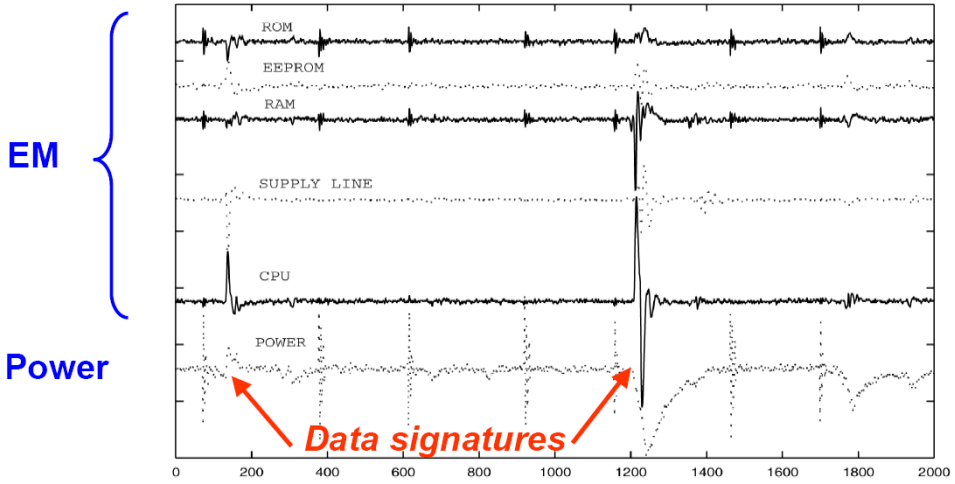
# Canaux cachés et équations de Maxwell




$$V = -\frac{d\phi}{dt}$$




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

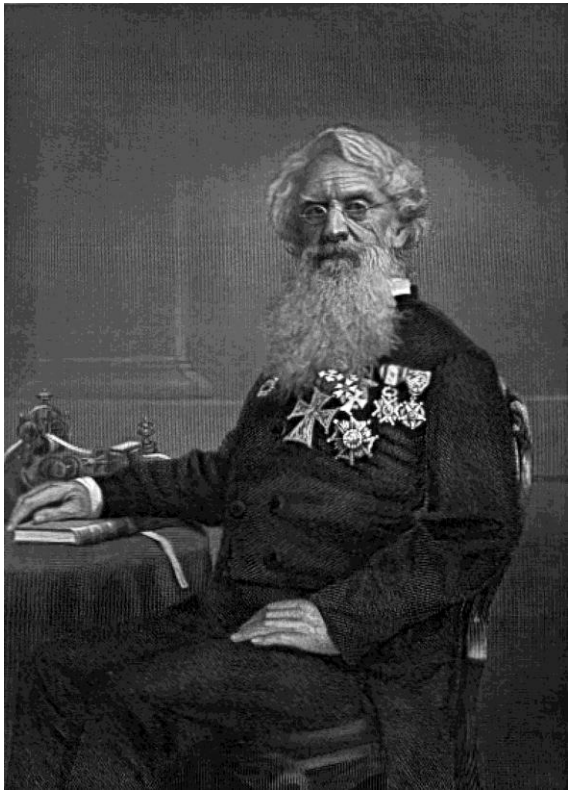


# RSA & *Morse Samuel F*

A • —
B — • • •
C — • — •
D — • •
E • (1 unit)
F • • — •
G — — •
H • • • •
I • •
J • — — —
K — • —
L • — • •
M — —

N — •
O — — —
P • — — •
Q — — • —
R • — •
S • • •
T — (3 units)
U • • —
V • • • —
W • — —
X — • • —
Y — • — —
Z — — • •

1 • — — — —
2 • • — — —
3 • • • — —
4 • • • • —
5 • • • • •
6 — • • • •
7 — — • • •
8 — — — • •
9 — — — — •
0 — — — — —



# $a^b$ modulus m

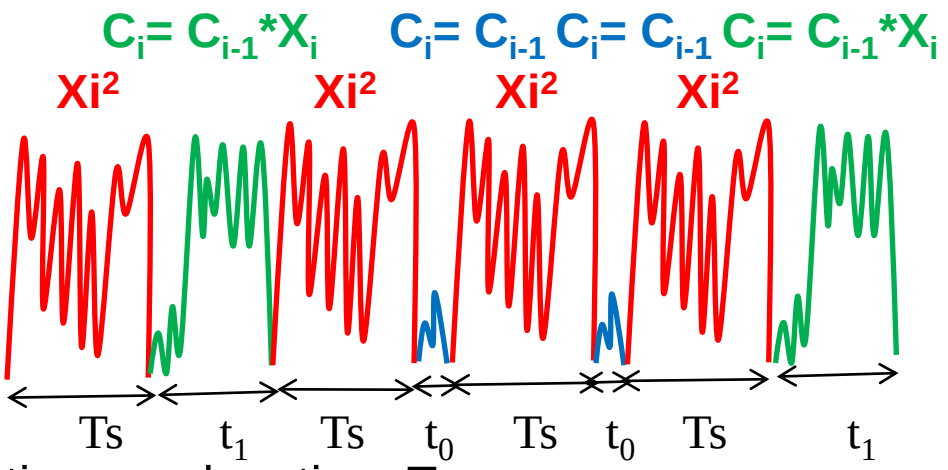
# Attaque d'un *exponentiator*

- C(forme chiffrée) = $M^d$ modulo m
- $d = d_0.2^0 + d_1.2^1 + d_2.2^2 + d_3.2^3 + d_4.2^4 + ... + d_i.2^i + ... + d_{p-1}.2^{p-1}$, ou di a pour valeur 0 ou 1.
- La forme chiffrée s'exprime sous forme d'un produit de p termes mi,
  - $C = m_0.m_1 m_2 ... m_i ... m_{p-1}$ modulo m, avec
    - mi= 1, si di=0 .
    - mi= $M^{2\wedge i}$ modulo m, si di=1
    - mi = $m_{i-1}^2$
- En constate que, dans cette implémentation de l'algorithme RSA (dite *exponentiation*), chaque bit (di) de la clé implique un temps calcul différent selon que sa valeur soit 0 (multiplication triviale par 1) ou 1 (multiplication par $M^{2\wedge i}$).

En fonction des différences de temps calculs observées on déduit la valeur de di (0 ou 1).

# Single Protocol Attack (SPA)

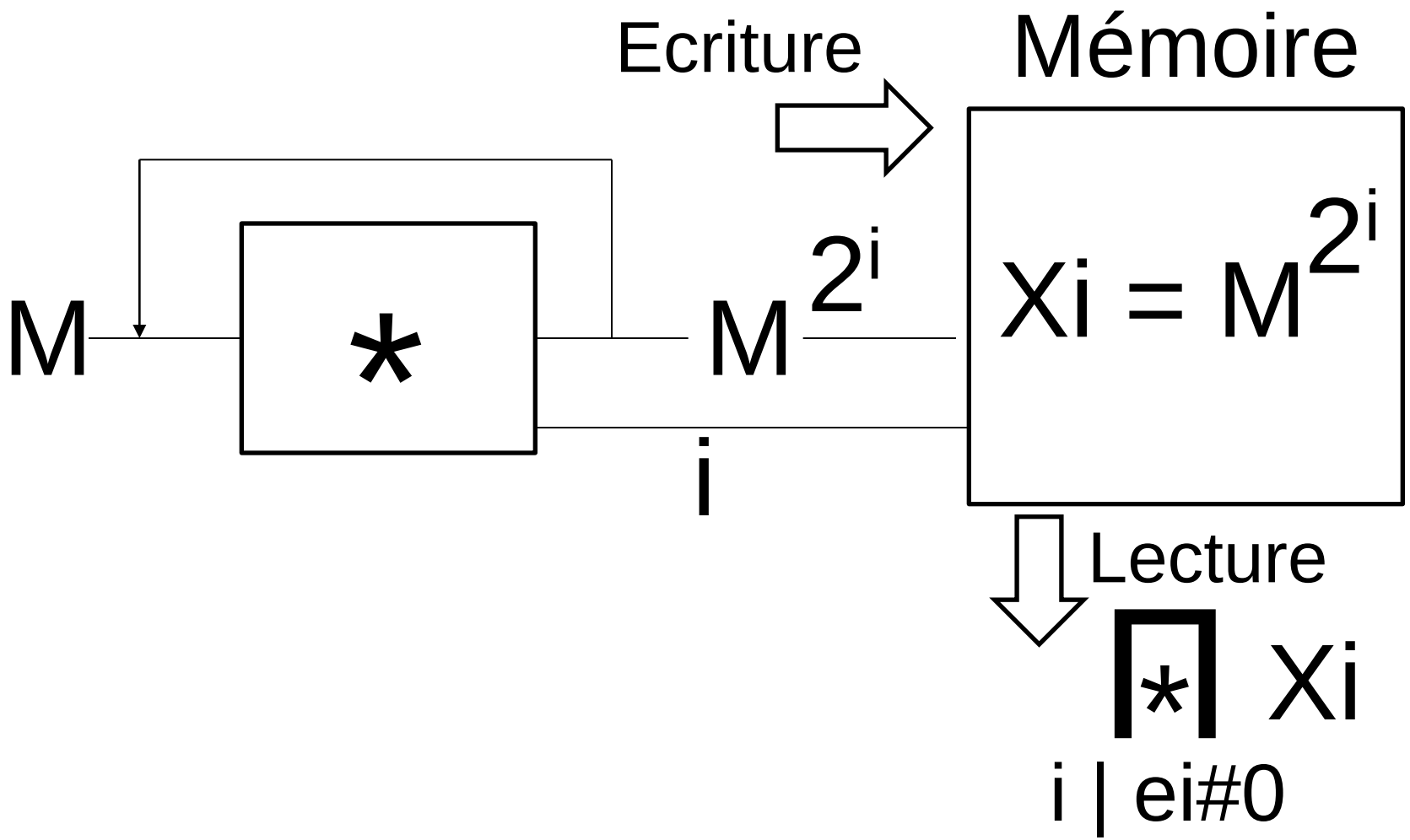
- $C = M^e = M * M * \ldots * M$ (e operations)
- $e = e_0\, 2^0 + e_1\, 2^1 + e_i\, 2^i + \ldots + e_{p-1}\, 2^{p-1}$ , ei = 0 or 1, $di = e_i\, 2^i$
- $C = M^{d0} * M^{d1} * M^{di} * \ldots * M^{dp-1}$
  - $C_0 = M_0 = M^{d0} =$ 1 or M
  - $C_i = C_{i-1} * M^{di}$
  - $C = C_{p-1}$




- Algorithm
  - Begin i=0
    - $X_0 = M$
    - $C_0 = X_0^{d0} = 1$ or M, this calculation needs a time $T_0$
  - Loop i<p
    - $X_i = X^2_{i-1} = X_{i-1} * X_{i-1}$ , this calculation needs a time Ts
    - If $e_i$=0 Then $C_i = C_{i-1}$, needs a "short" time $t_0$
    - If $e_i$=1 Then $C_i = C_{i-1} * X_i$ , this this calculation needs a "long" time $t_1$

- $T = T_0 + Ts + t_{e1} + Ts + t_{ei} + \ldots + Ts + t_{ep-1}$

# Stockage des termes M^2^i



# Attaque de Bellcore, D.Boneth 1997

$E1 = x^s \bmod p$, $E2 = x^s \bmod q$

$y = a.E1 + b.E2 \bmod pq$

$a = 1 \bmod p$, $a = 0 \bmod q$

$b = 1 \bmod q$, $b = 0 \bmod p$

$y = a.E1 + b.E2 \bmod pq$

$y' = a.E1' + b.E2$, faute de calcul sur $E1'$

$y - y' = a.(E1 - E1')$

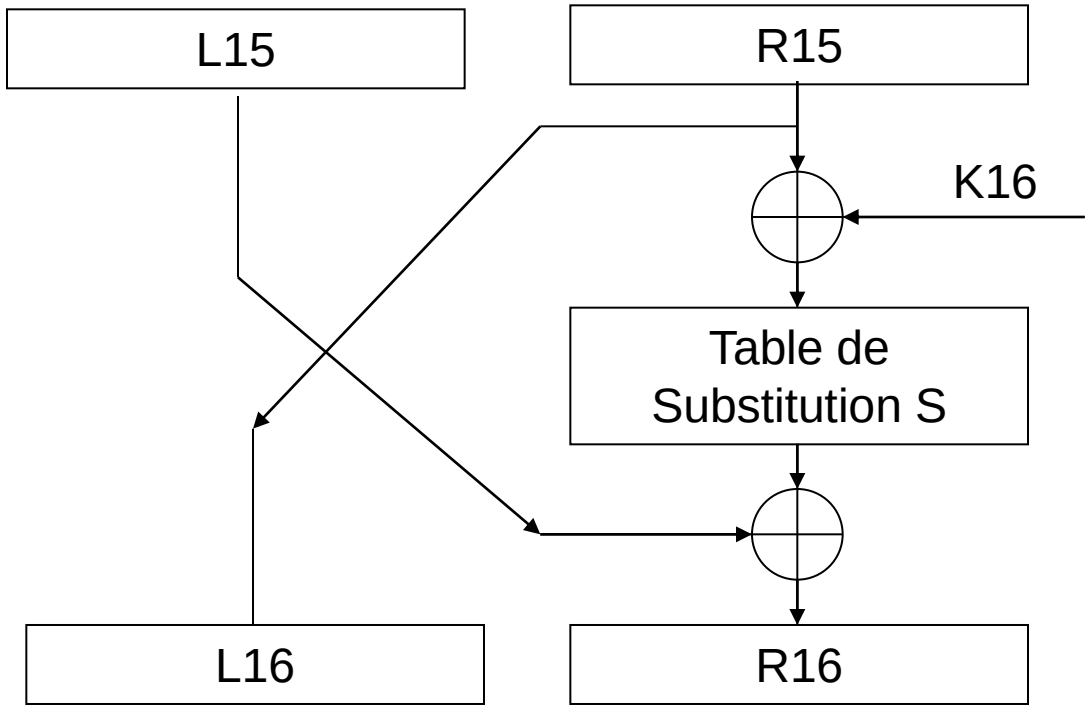
Si $E1 - E1'$ n'est pas divisible par p

$PGCD(y - y', n) = q$ $(n = p.q)$

# DPA

- Paul C. Kocher, Joshua Jaffe, Benjamin Jun: Differential Power Analysis. CRYPTO 1999: 388-397
  - Covariance, $cov(X,Y) = \sigma_{X,Y} = E(XY) - E(X)E(Y)$
  - Coefficient de corrélation, $\rho_{X,Y} = \frac{\sigma_{XY}}{\sqrt{V(X)V(Y)}}$, $\rho_{X,Y} \in [-1, 1]$
  - $E(XY) = E(X)E(Y) + \rho_{X,Y}\, \sigma(X)\, \sigma(Y)$
- Si l'on suppose :
  - Un domaine de clés (i) de $2^p$ valeurs, $i \in [0, 2^p - 1]$
  - Un effet physique associé à toutes les valeurs d'entrées (k) et des clés (i), $X_i(k,t)$, tel que la puissance électrique consommée.
  - Une fonction Y corrélée à la clé secrète j et définie pour toutes les valeurs d'entrée (k), et telle que pour chaque clé (i), $\langle Y_i(k) \rangle_k = 0$
  - Pour toute mauvaise clé (i)
    - $\rho_{X,Y} = 0$, $\langle X_i(k,t).Y_i(k) \rangle_k = \langle X_i(k,t) \rangle_k \langle Y_i(k) \rangle_k = 0$
  - Pour bonne clé (j), $\rho_{X,Y} \neq 0$
    - $\langle X_j(k,t).Y_j(k) \rangle_k = \rho_{X,Y}\, \sigma(X)\, \sigma(Y)$

# Injection de fautes DES - 1/2

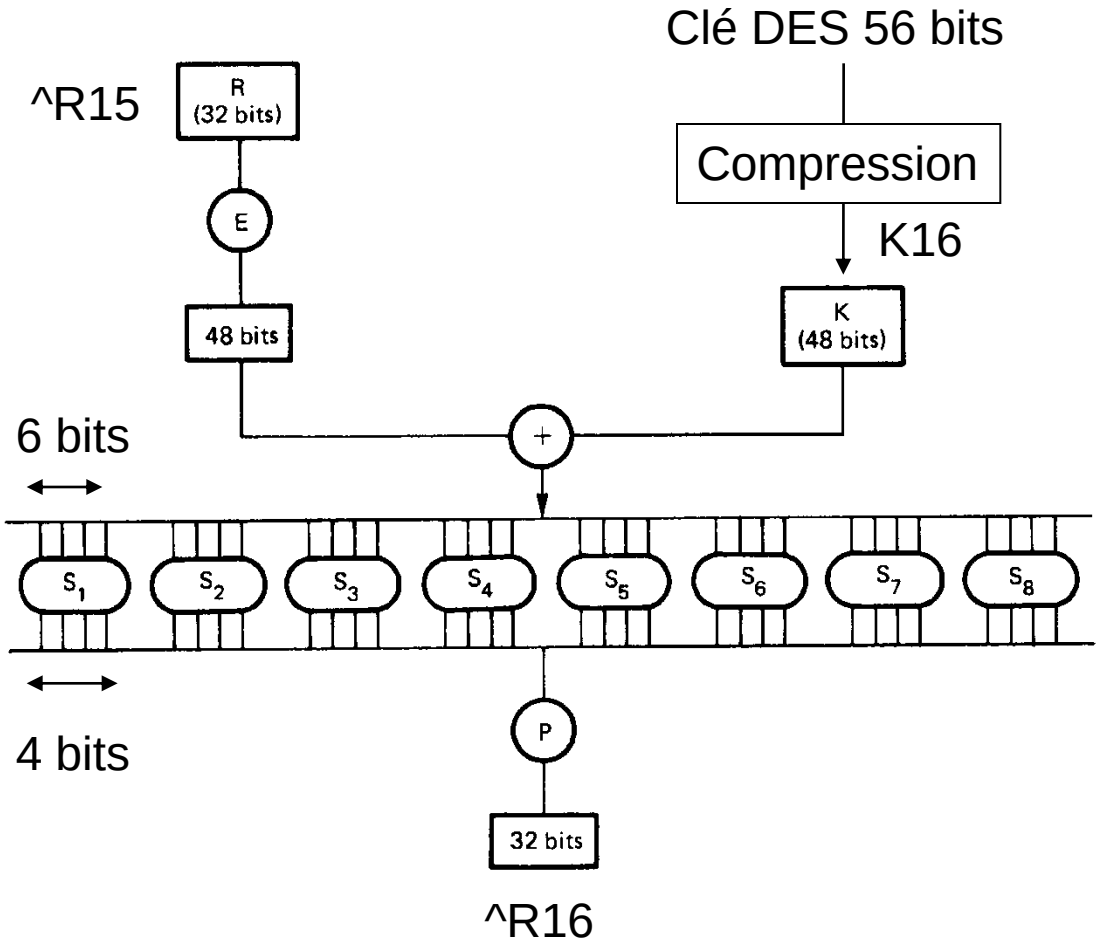



- L'attaquant connaît la *bonne* valeur R16
- Il crée une faute ^R15, qui implique la valeur ^R16

Contrainte (C)

$$R_{16} \oplus \hat{R}_{16} = S(L_{16} \oplus K_{16}) \oplus S(\hat{L}_{16} \oplus K_{16})$$

# Injection de fautes DES - 2/2

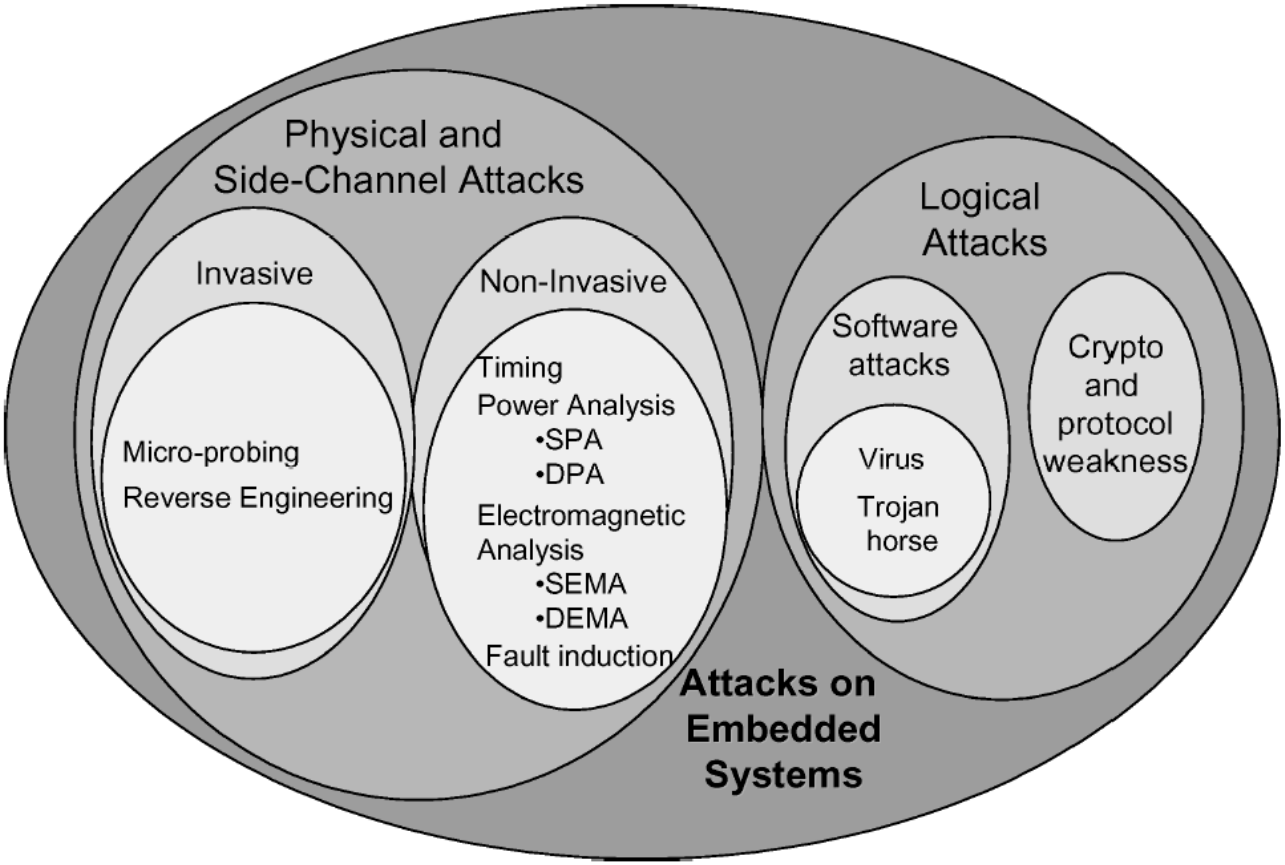



- Environ $2^{18}$ valeurs de K16 réalisent la contrainte (C)
- Environ $2^{24}$ clés DES réalisent la contrainte C

# Une tentative de taxonomie des attaques

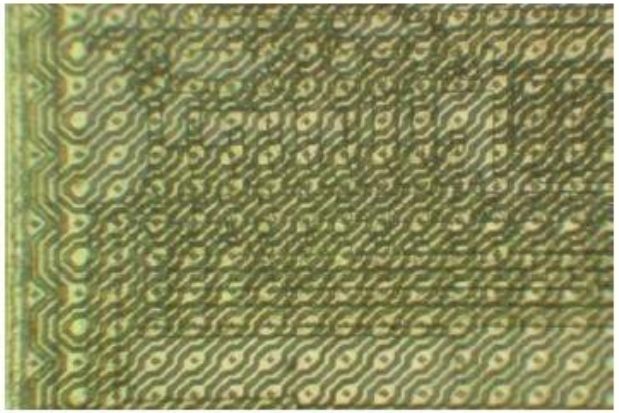
- Les attaques par canaux cachés (Side channels attacks)
  - Timing Attacks
  - Power Attacks
    - Simple Power Attack (SPA)
    - Differential Power Attack (DPA, Paul Kocher 1995)
  - Electromagnetic Attack (EMA)
    - Simple EMA (SEMA)
    - Differential EMA (DEMA)
- Les attaques par injection de fautes (fault injection)
  - Tension d'alimentation (power glitch)
  - Variation de l'horloge
  - Température
  - Lumière incohérente
  - Laser
  - Faisceau d'ion
  - Rayon X
  - Autre
- Les attaques par sondes (probing attack)
  - Acquisition d'information particulières durant l'exécution d'un algorithme

# Secure Embedded Systems, S.Ravi, 2004

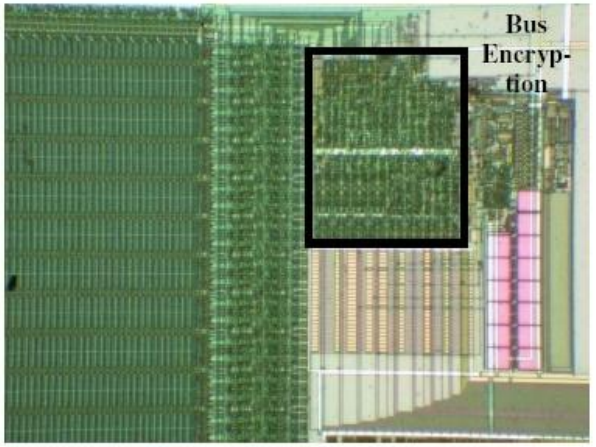



Examples of attack threats faced by embedded systems.

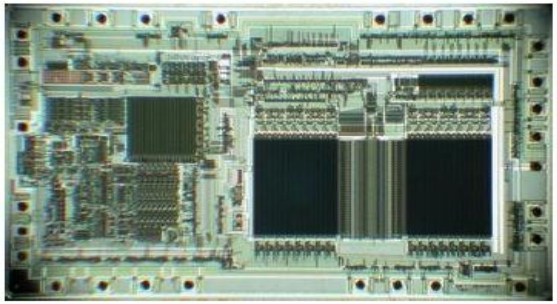
# Contre-mesures physiques



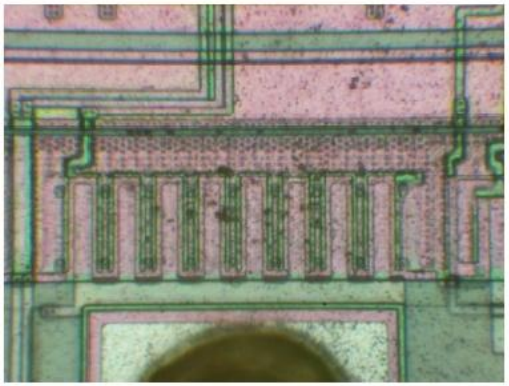
Top metal sensor on ST16 smartcard




Hardware bus encryption module in Infineon SLE66 family smartcard chip

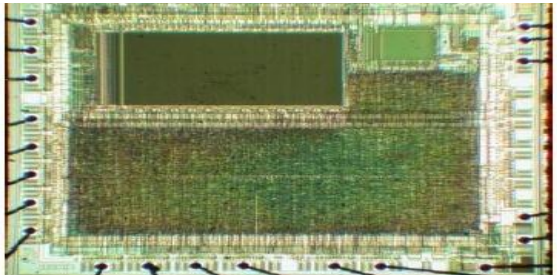


MC68HC705PA microcontroller with clearly distinguishable blocks



Second metal layer and polysilicon layer on microchip PIC16F877A microcontroller

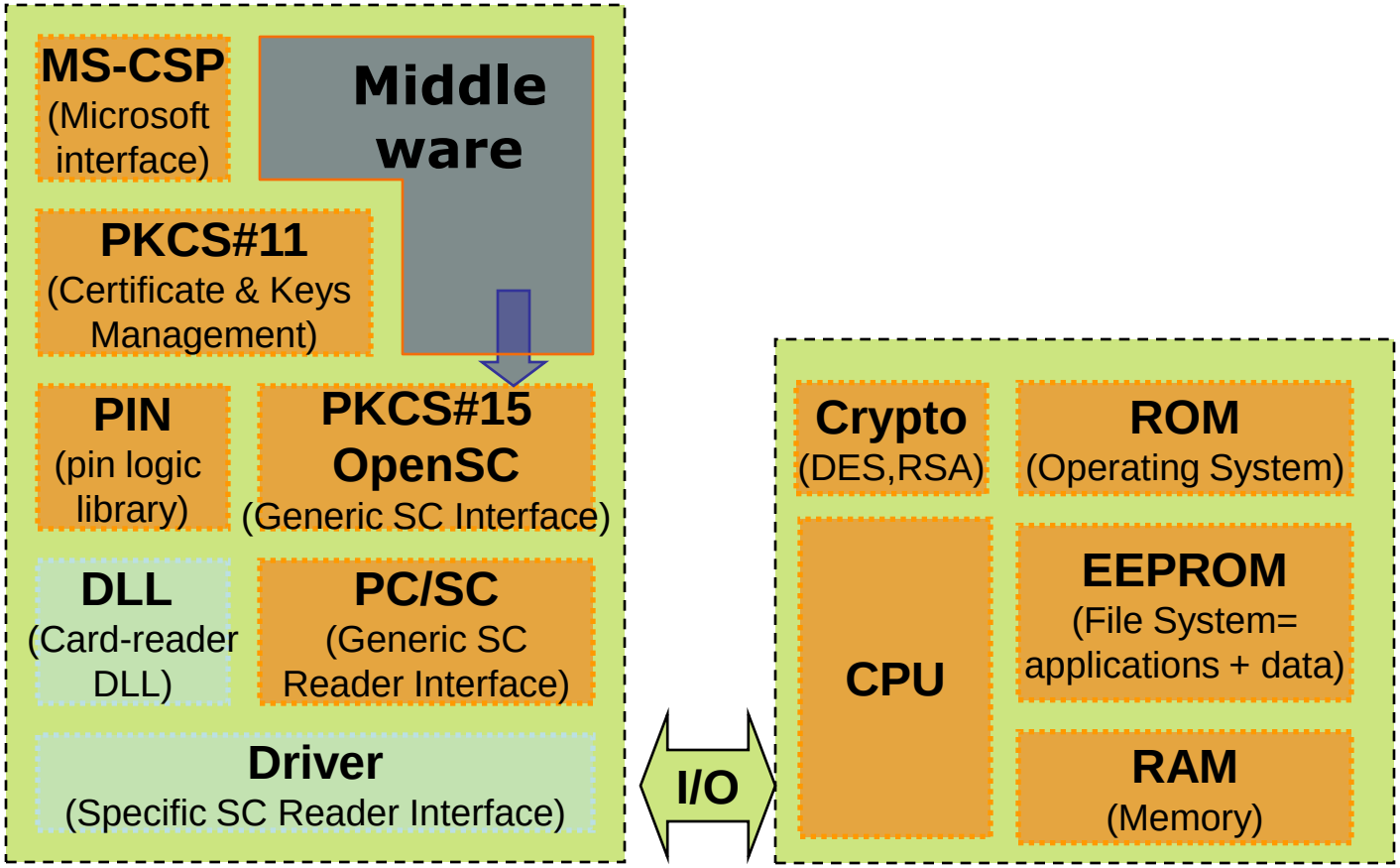
Top metal layer on microchip PIC16F877A microcontroller



SX28 microcontroller with ‘glue’ logic design

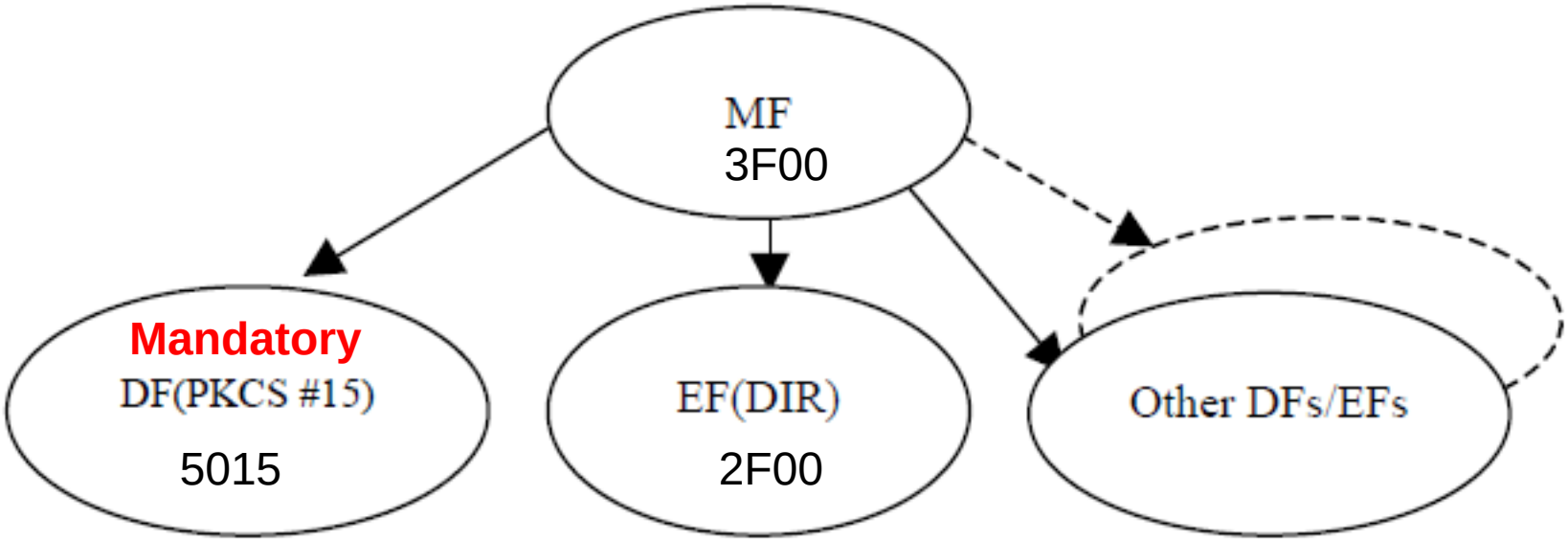
# APIs et Machines Virtuelles

# PKCS#11 et PKCS#15



# Structure de fichier PKCS#15

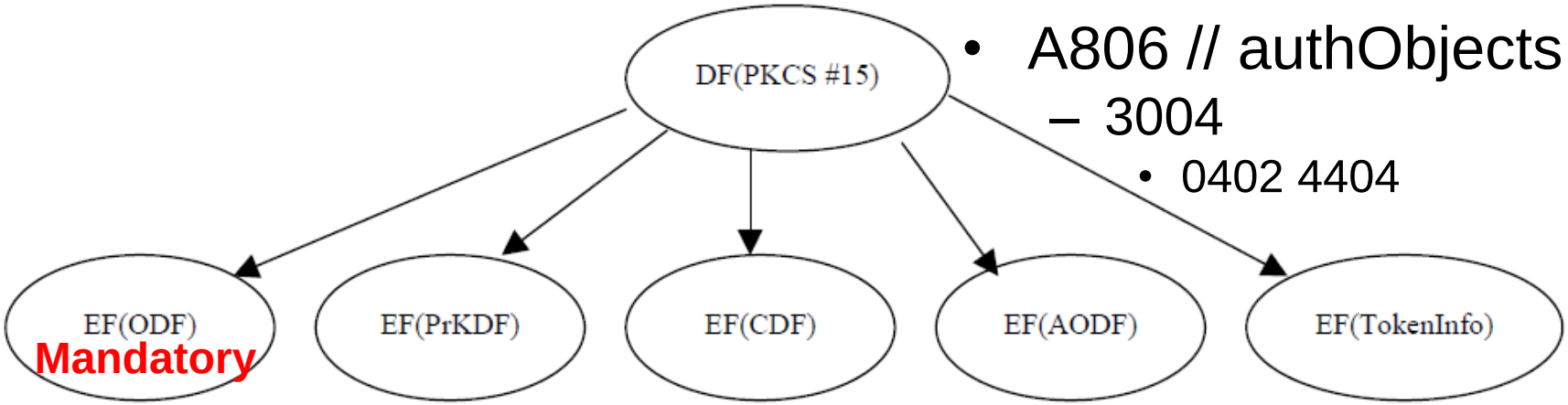
- AID= A000000063504B43532D3135



# DF(PKCS#15)

- ODF: object directory file
- PrKDF: private key directory file
- CDF: certificate directory file
- AODF: authentication object directory file (PINs)
- DODF: data object directory file

- A006 // privateKeys
  - 3004
    - 0402 4401
- A406 // certificates
  - 3004
    - 0402 4402
- A706 // dataObjects
  - 3004
    - 0402 4403
- A806 // authObjects
  - 3004
    - 0402 4404



# PKCS#15: Références Croisées

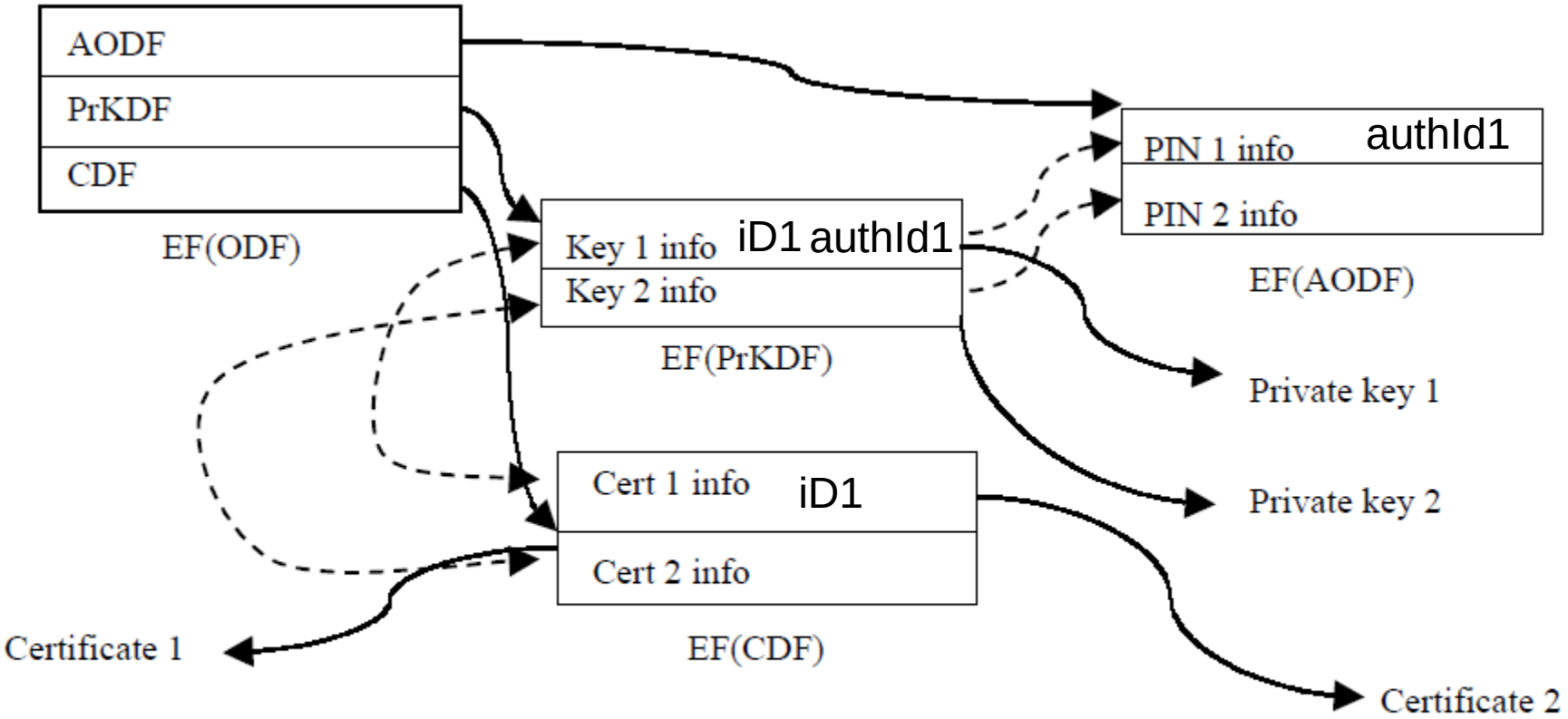
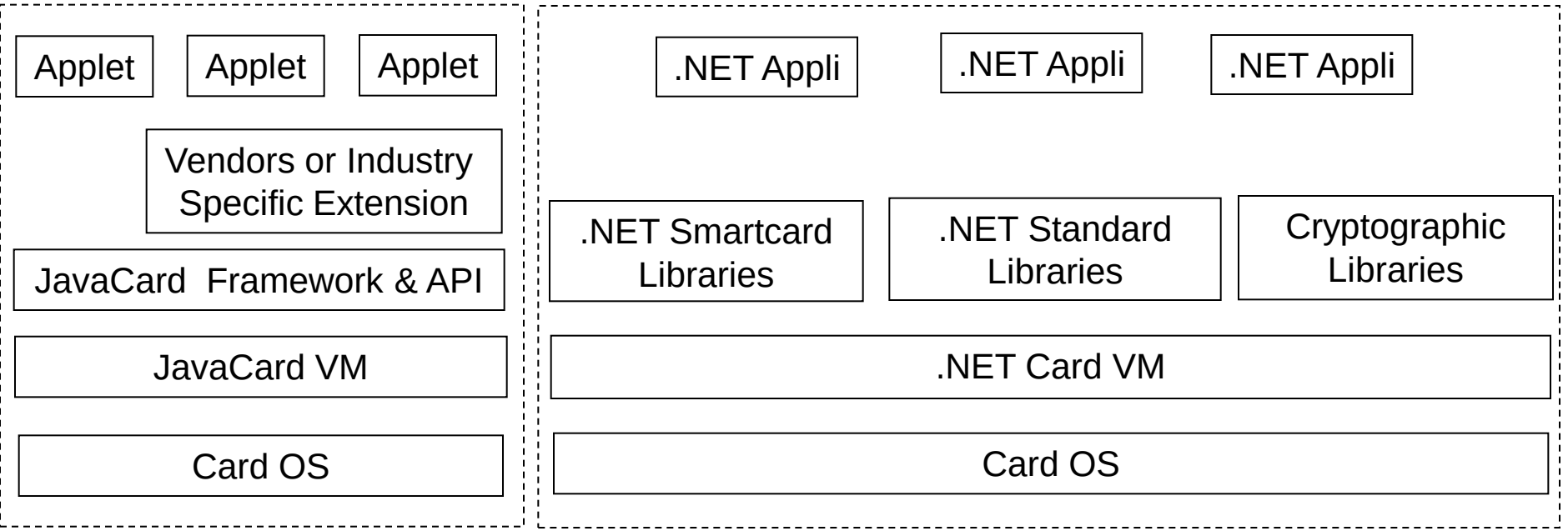



**Figure 16 – IC card file relationships in DF(PKCS15). Dashed arrows indicate cross-references.**

# Java VM and .NET VM

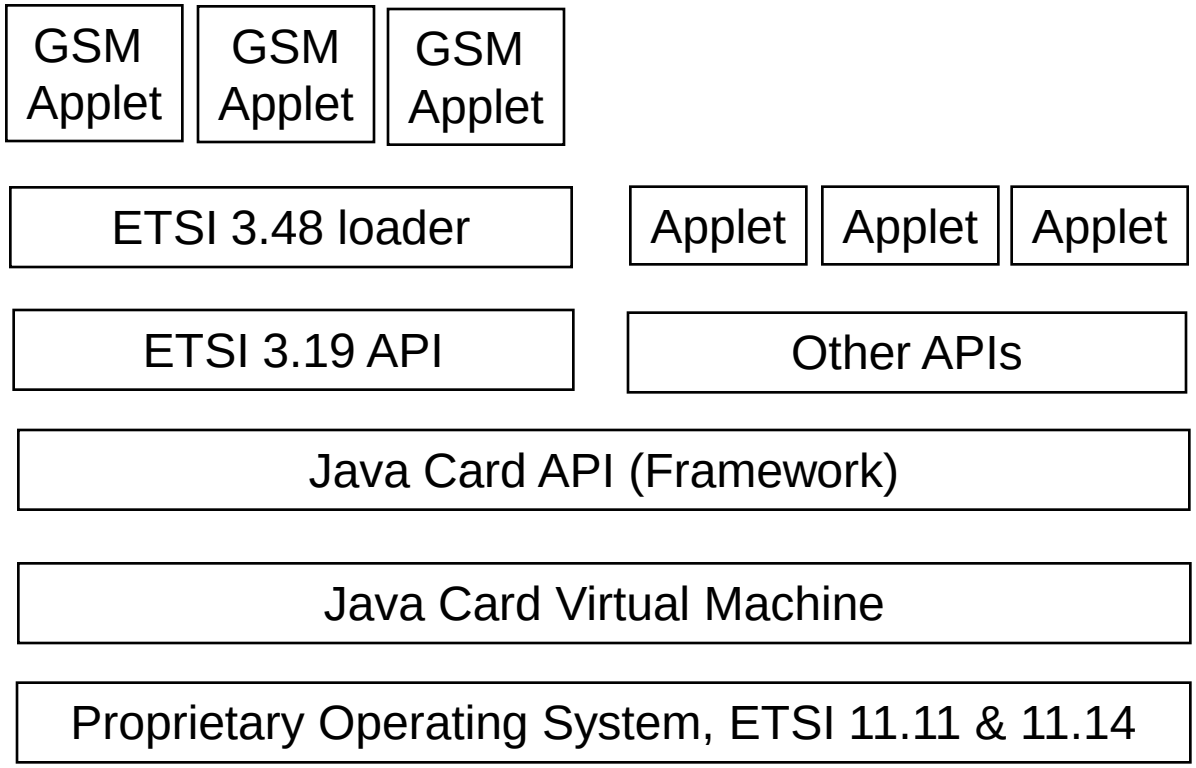




.JAVA


.CS

# Pile GSM


GSM Applet
GSM Applet
GSM Applet
ETSI 3.48 loader
Applet
Applet
Applet
ETSI 3.19 API
Other APIs
Java Card API (Framework)
Java Card Virtual Machine
Proprietary Operating System, ETSI 11.11 & 11.14

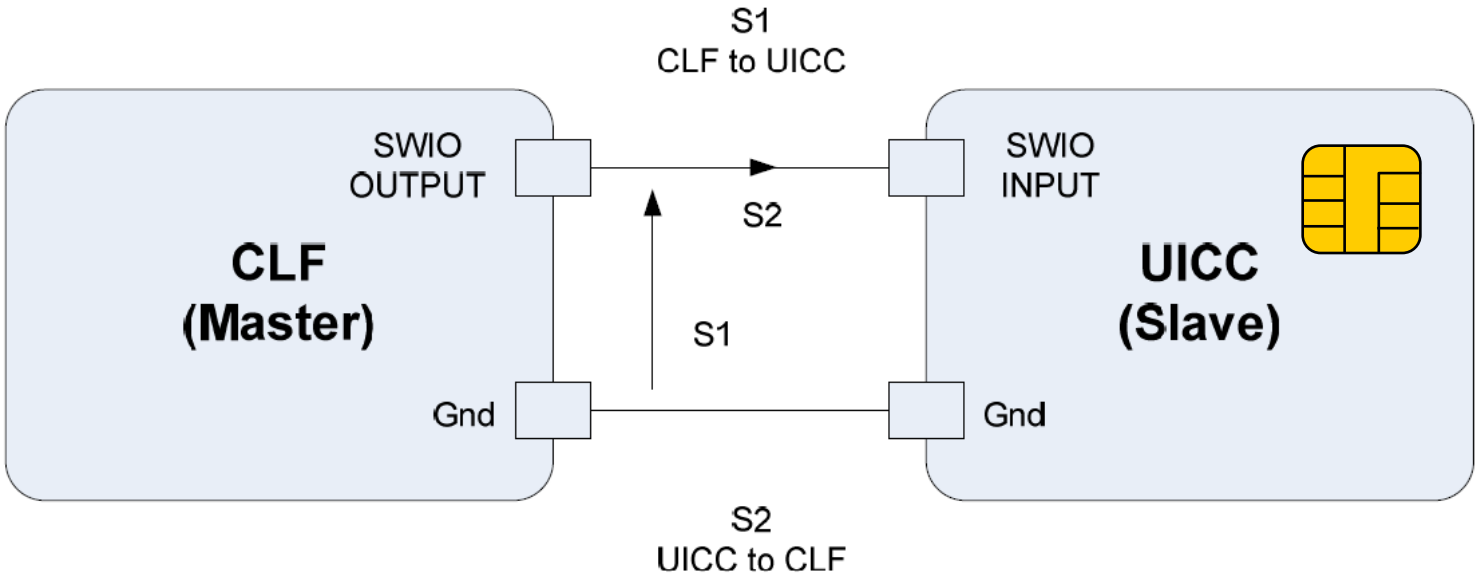
# Les normes NFC

# La genèse NFC

- 1994, Mifare 1K
  - En 2011 Mifare représente 70% du marché du transport
- 2001, Standards ISO 14443 (13,56 Mhz)
  - Type A (Mifare)
  - Type B
- 2004, NFC Forum
  - Mifare (NXP), ISO14443A, ISO14443B, Felica (Sony)
  - 3 modes fonctionnels
    - Reader/Writer, Card Emulation, Peer to Peer

# La carte SIM devient NFC: Le Contactless Front-end (CLF)

- ETSI TS 102 613



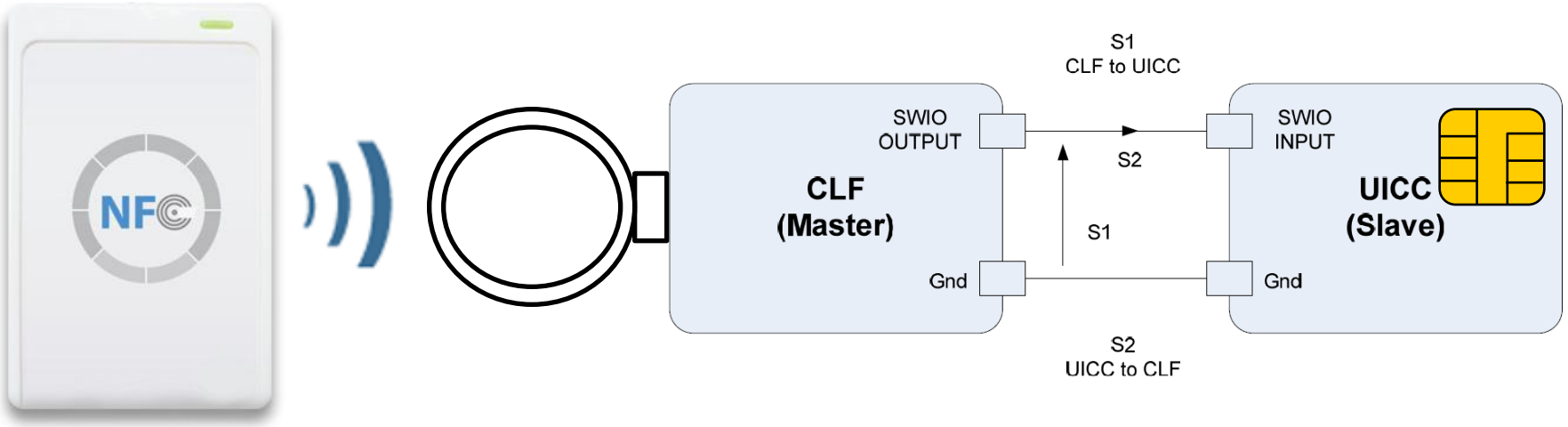

Un lien physique: le Single Wire Protocol (SWP)

Un protocole HDLC simplifié : SHDLC

# Reader/Writer – Card Emulation

Reader

Card Emulation



# Le Mode P2P

- Android NDEF Push Protocol Specification
  - Version 1, 2011-02-22

“The NDEF Push Protocol (NPP) is a simple protocol built on top of LLCP which is designed to push an NDEF message from one device to another.”

# Apercu des standards NFC

| Activity | Technology / Device Platform | | | | | | |
|---|---|---|---|---|---|---|---|
| Listen, RF Collision Avoidance, Technology Detection, Collision Resolution | NFC-A<br>ISO 14443-2A<br>ISO 14443-3A | | | | NFC-B<br>14443 -2B<br>14443 -3B | NFC-F<br>ISO 14443-2A<br>ISO 14443-3A<br>FELICA | |
| Device Activation | | Type 1 Tag Platform | Type 2 Tag Platform | Type 4A Tag Platform | Type 4B Tag Platform | Type 3 Tag Platform | |
| Data Exchange | NFC-DEP Protocol | Type 1, 2, and 3 Tag Half-duplex Protocol | | ISO-DEP Protocol | | Type 1, 2, and 3 Tag Half-duplex Protocols | NFC-DEP Protocol |
| Device Deactivation | NFCIP-1 | | | ISO 14443-4 | | | NFCIP-1 |

NDEF

SNEP

LLCP

NFC-SEC

DEP

Passive Mode
Active Mode
NFCIP-1

*ISO/IEC_18092 standard and NFCIP-1 standards are similar

DEP: Data Exchange Protocol (Supports Read/Write Operations for Tags)

# La Radio NFC

ISO 14443
106 kbps
212 kbps
424 kbps
848 kbps

| Standard | PCD to ICCC<br>Reader to Card | PICC to PCD<br>Card to Reader |
|---|---|---|
| ISO 14443-2A<br>NFC-A | ASK 100%<br>Modified Miller | Subcarrier fc/16<br>OOK Manchester |
| ISO 14443-2B<br>NFC-B | ASK 10%,<br>NRZ-L | Subcarrier fc/16<br>BPSK, NRZ-L |

NFCIP-1
Passive
Mode

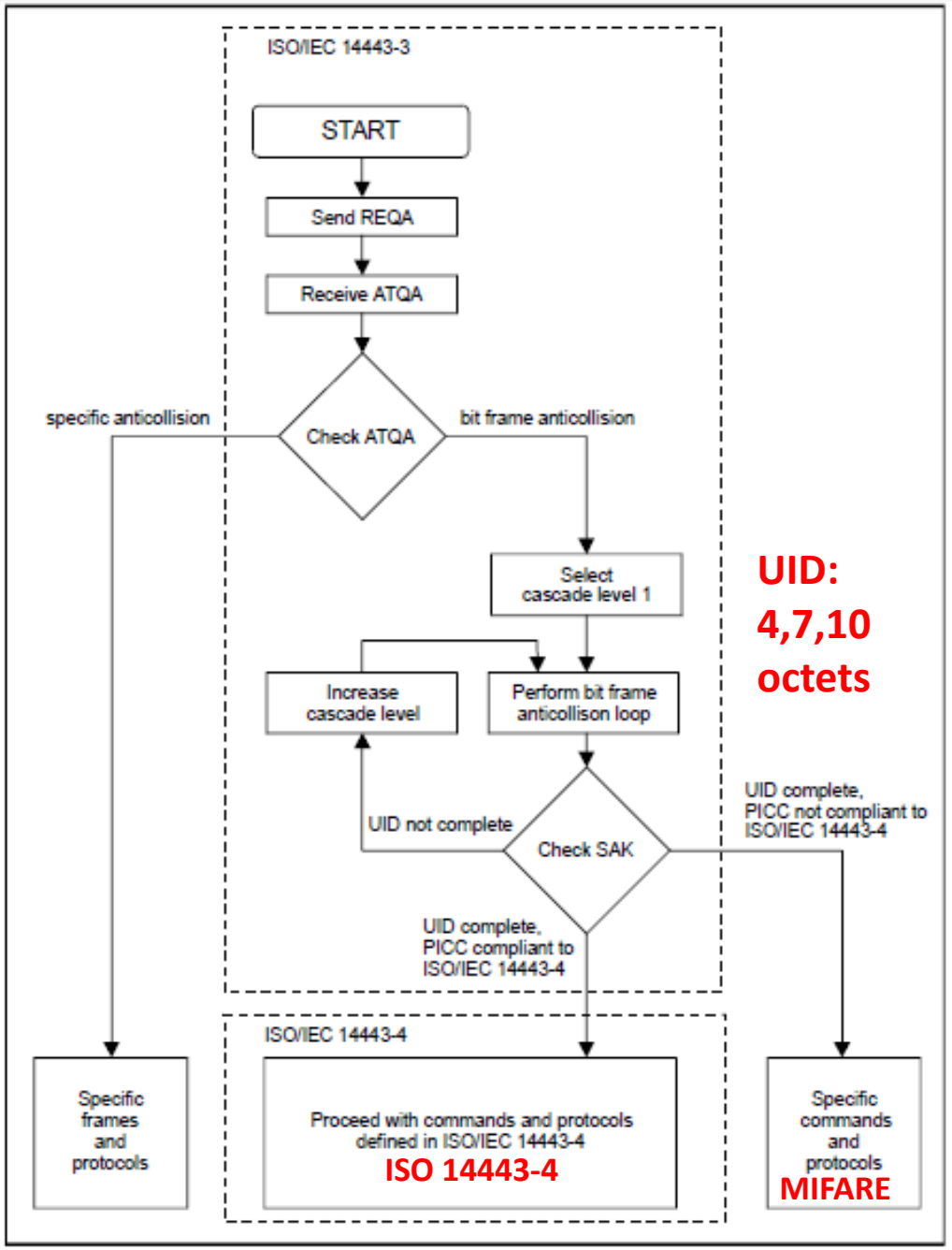
| Bit Rate | Initiator | Target |
|---|---|---|
| 106 kbps | ASK 100%<br>Modified Miller | Subcarrier fc/16<br>OOK Manchester |
| 212-424 kbps | ASK 8-30%<br>OOK Manchester | ASK 8-30%<br>OOK Manchester |

NFCIP-1
Active
Mode

| Bit Rate | Initiator | Target |
|---|---|---|
| 106 kbps | ASK 100%<br>Modified Miller | ASK 100%,<br>Modified Miller |
| 212-424 kbps | ASK 8-30 %<br>OOK Manchester | ASK 8-30%,<br>OOK Manchester |

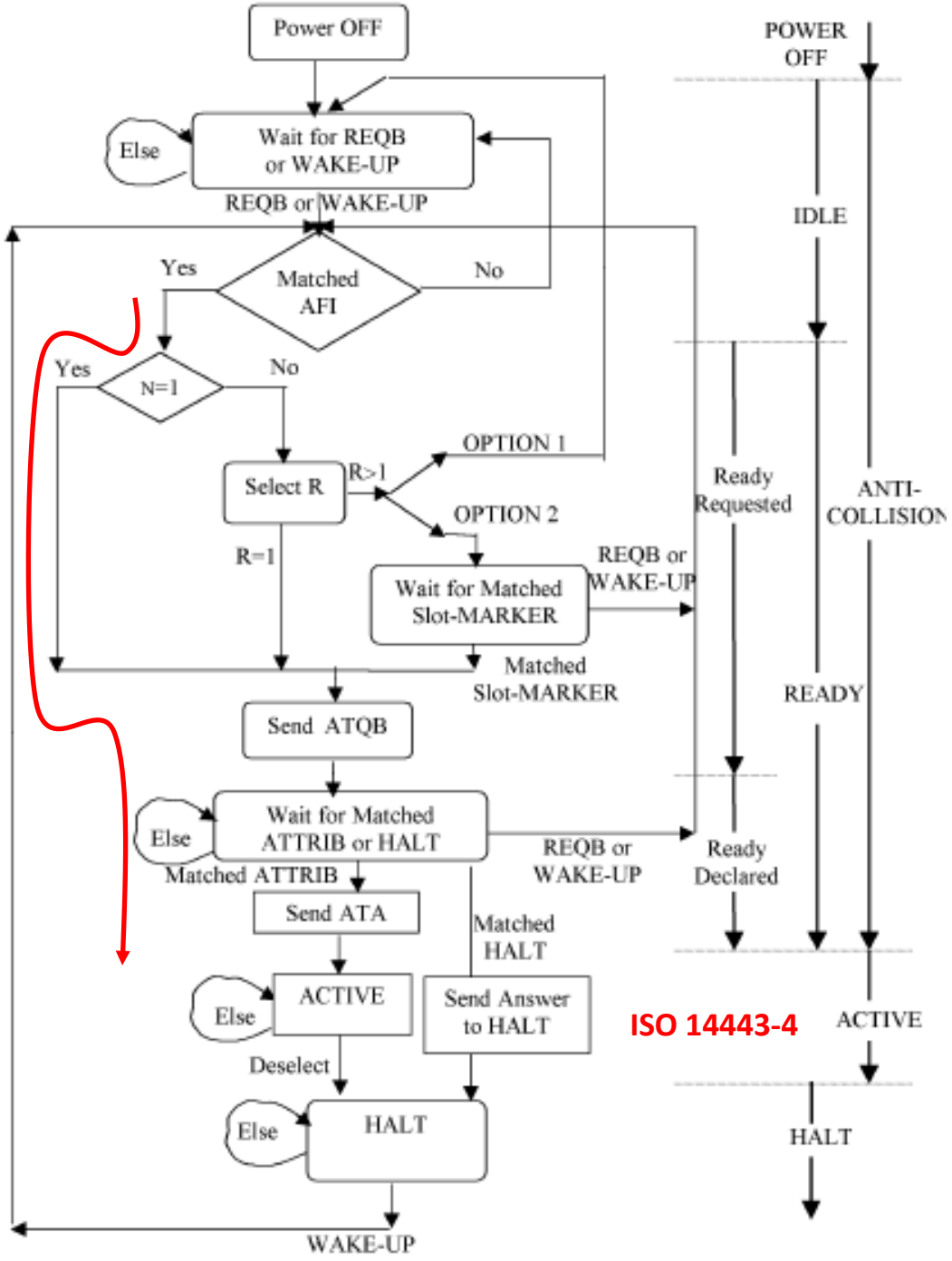
# ISO 14443-3A State Machine

- REQA: Request Command for Type A
- ATQA: Answer To Request of Type A
- SAK: Select AcKnowledge
- UID: Unique IDentifier



# ISO 14443-3B State Machine

- AFI: Application Family Identifier (4 bytes).
- REQB: Request of Type B
- ATQB: Answer To Request of Type B
- ATA Answer To ATTRIB



# ISO 14443-4 Frames (T=CL)

- Les trames ISO 14443-4 transportent des APDUs (ISO 7816-4)

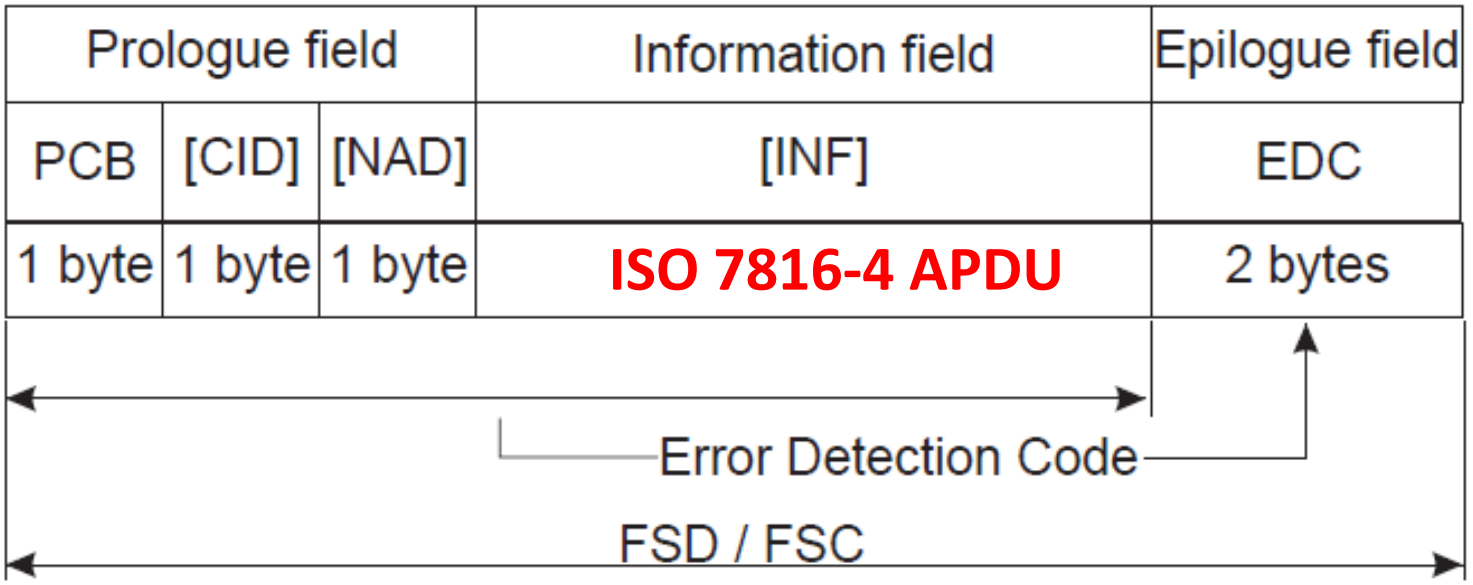



**Figure 14 — Block format**

# ISO 14443-4

- RATS: Request for Answer To Select
- ATS: Answer To Select

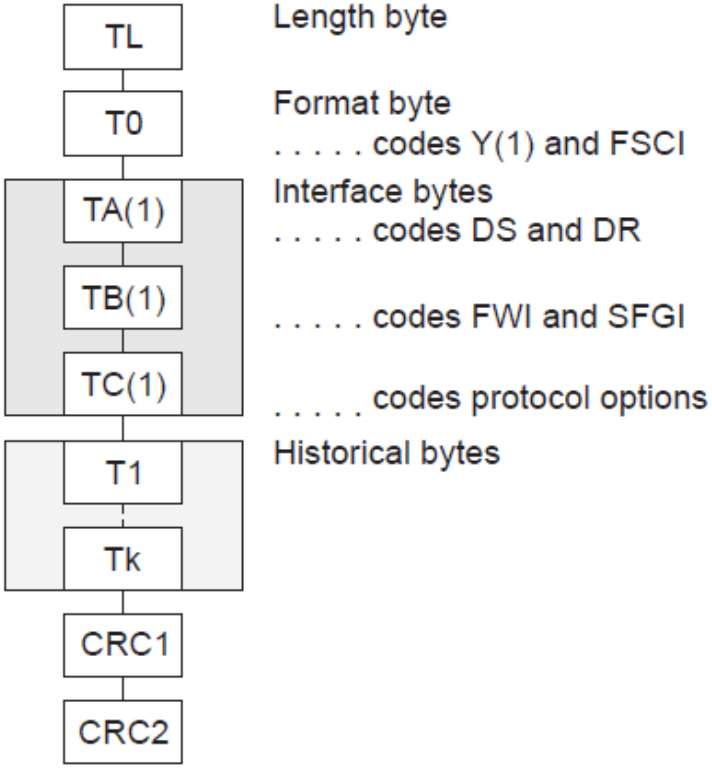



**Figure 4 — Structure of the ATS**

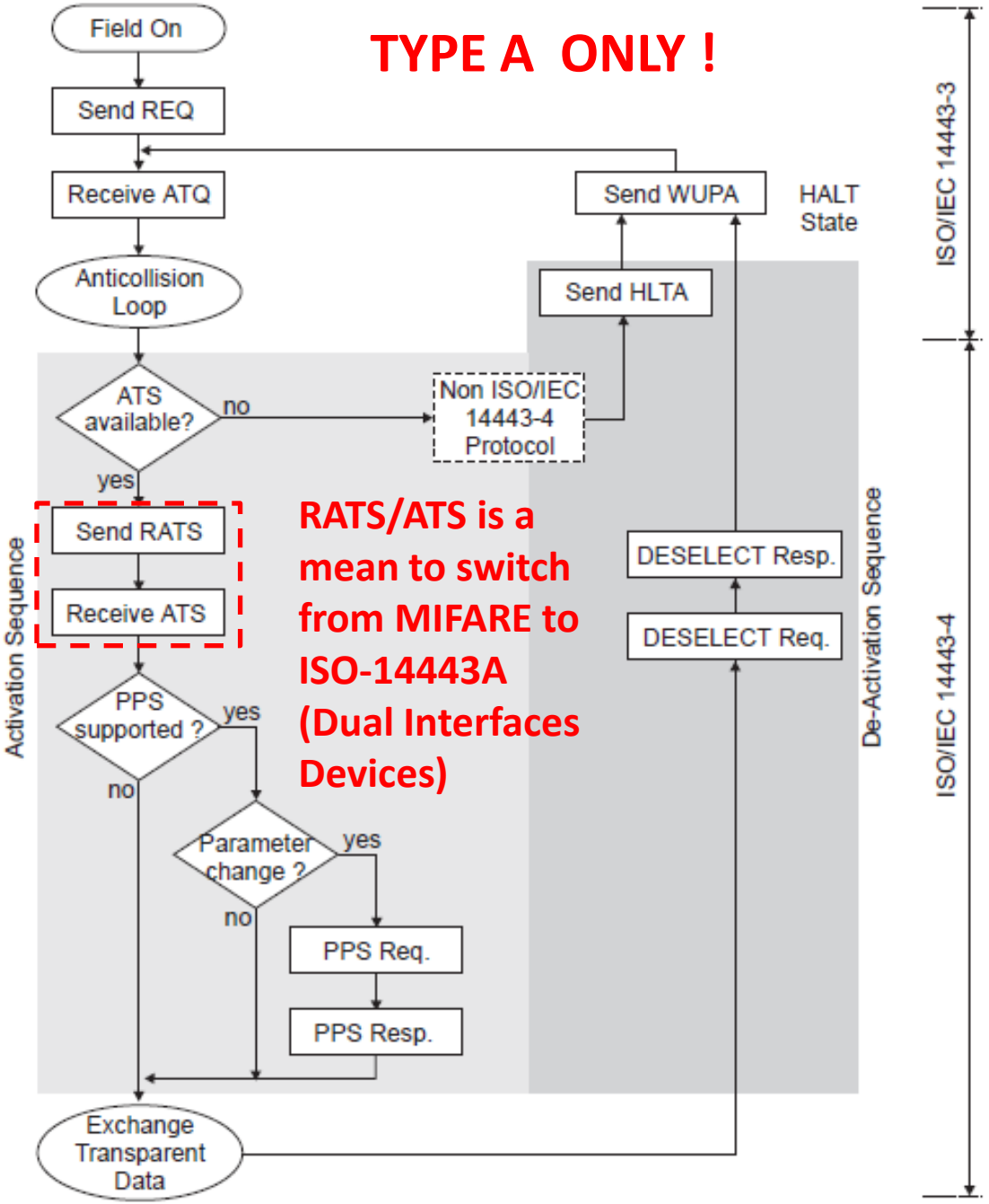
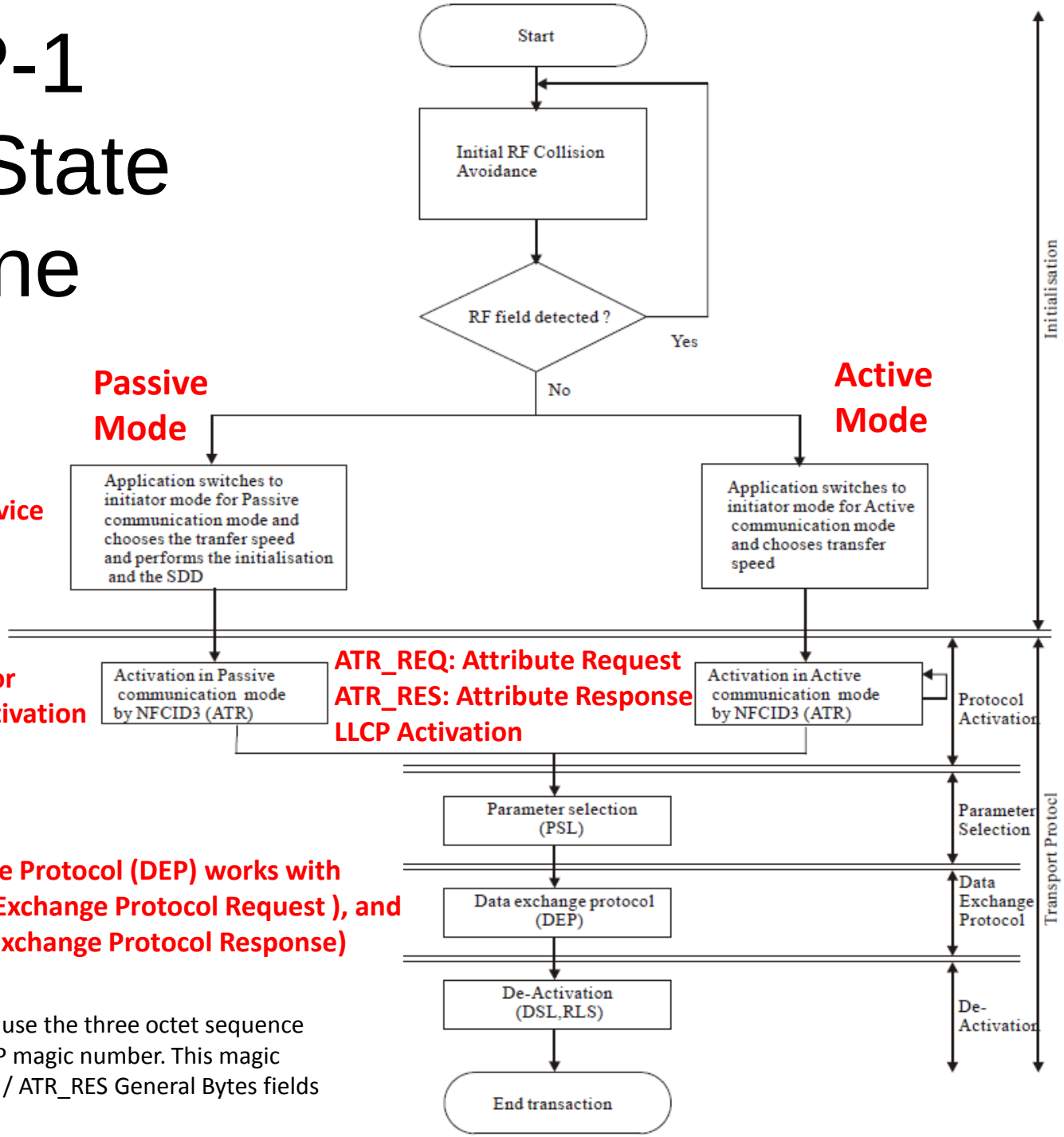



**Figure 1 — Activation of a PICC Type A by a PCD**

# NFCIP-1 Initiator State Machine




**SDD Single Device Detection**

**NFCID3: Random ID for transport protocol activation**

**ATR_REQ: Attribute Request**
**ATR_RES: Attribute Response**
**LLCP Activation**

**The Data Exchange Protocol (DEP) works with**
**- DEP_REQ (Data Exchange Protocol Request ), and**
**- DEP_RES (Data Exchange Protocol Response)**

The NFC-DEP MAC component SHALL use the three octet sequence “46h 66h 6Dh” as the NFC Forum LLCP magic number. This magic number is encoded into the ATR_REQ / ATR_RES General Bytes fields

# LLCP

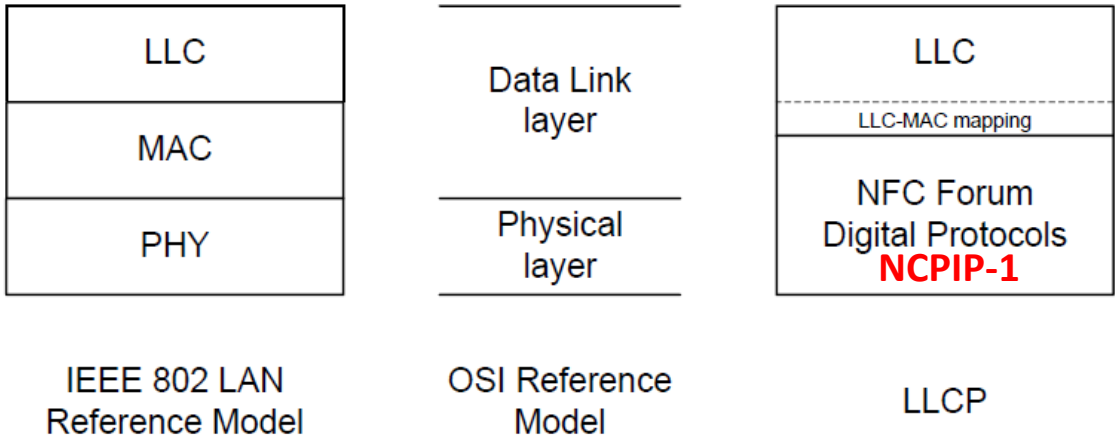



**Figure 1: Relationship to OSI Reference Model**

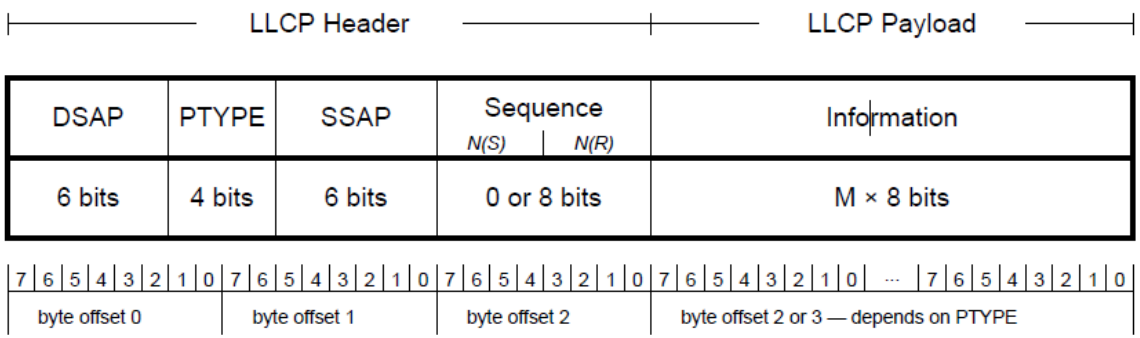



**Table 3: PDU Type Values**

| PDU Type | PTYPE | Link Service Class |
|---|---|---|
| SYMM | 0000 | 1, 2, 3 |
| PAX | 0001 | 1, 2, 3 |
| AGF | 0010 | 1, 2, 3 |
| UI | 0011 | 1, 3 |

| PDU Type | PTYPE | Link Service Class |
|---|---|---|
| CONNECT | 0100 | 2, 3 |
| DISC | 0101 | 1, 2, 3 |
| CC | 0110 | 2, 3 |
| DM | 0111 | 1, 2, 3 |
| FRMR | 1000 | 2, 3 |
| SNL | 1001 | 1, 2, 3 |
| reserved | 1010 | |
| reserved | 1011 | |
| I | 1100 | 2, 3 |
| RR | 1101 | 2, 3 |
| RNR | 1110 | 2, 3 |
| reserved | 1111 | |

# SNEP, Android 4.x

NFC Initiator

NFC Target

**ATR_REQ**, NFC-MAGIC VERSION WKS LTO →

← **ATR_RES**, NFC-MAGIC VERSION WKS LTO

**LLCP-SYMM** [0000] →

**LLCP-SYMM** [0000] ←

**LLCP-SYMM** [0000] →

**CONNECT** [1120], DSAP=4, SSAP=32 ←

**CC** [818402020078], DSAP=32, SSAP=4, MUI →

**Information , SNEP PUT**
[132000 10020000000F D1010B5402656E **6B657976616C7565**]
DSAP=4, SSAP=32, NS=0, NR=0, SNEP HEADER, NDEF RECORD, **keyvalue** ←

**RR**(1) [834401], SSAP=4, DSAP=32 →

**Information, SNEP Success** [830401 10810000000] ,
SSAP=4, DSAP=32, NS=0, NR=1 →

**RR**(1) [13600], DSAP=4, SSAP=32 ←

**DISCONNECT** [1160] DSAP=4, SSAP=32 ←

**DM** [1C00], DSAP=0, SSAP=0 →

**LLCP-SYMM** [0000] ←

# NDEF: NFC Data Exchange Format

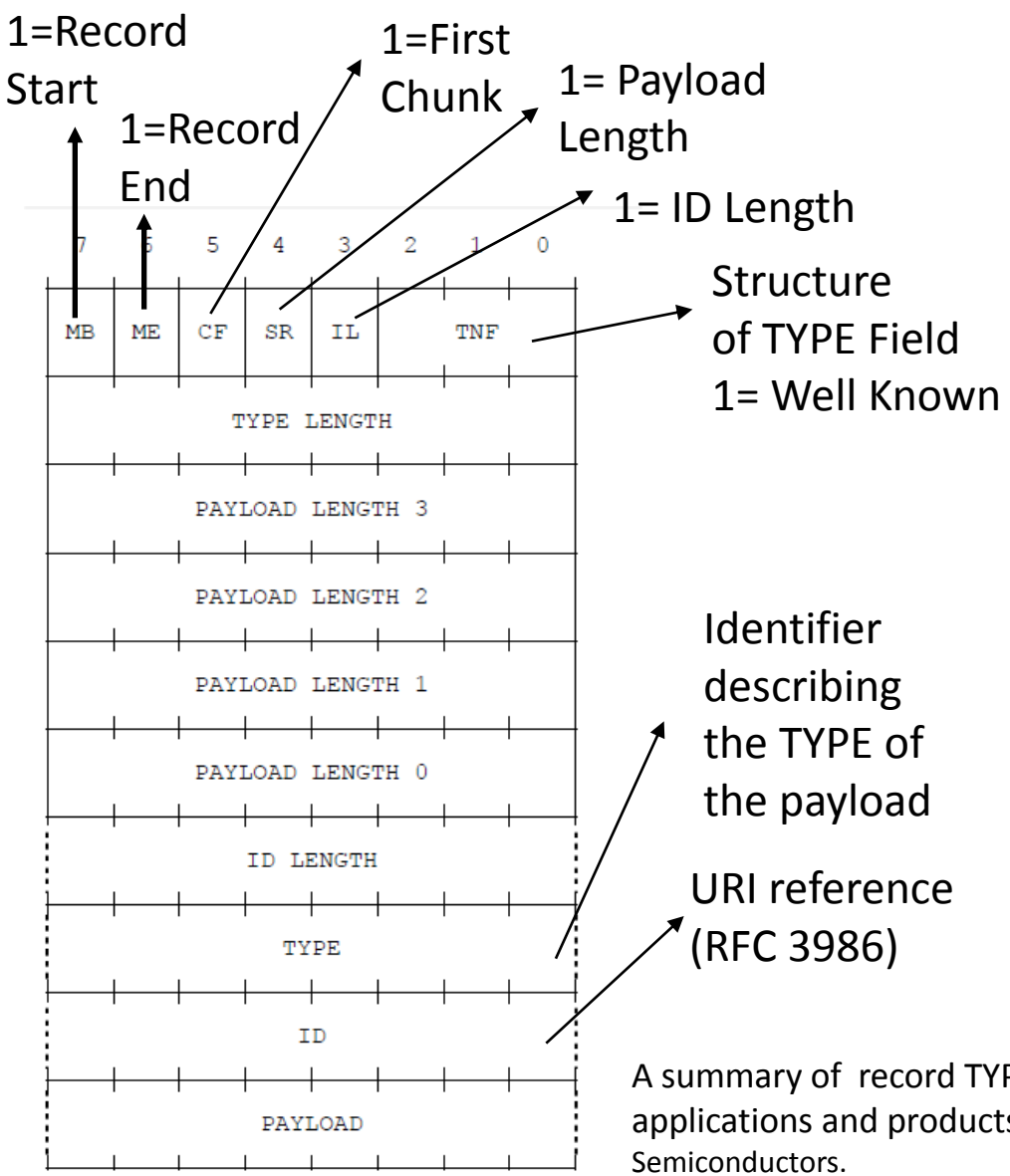



Figure 3. NDEF Record Layout

**NDEF Record Example:**
**(NFC Text Record Type Definition)**

D1: 1 1 0 1 0 001
01: Type Length
0A: Payload Length
54: Type= 'T', Text
02: ID= UTF8
65 6E: "EN"
53 61 6D 70 6C 65 20: "Sample "

A summary of record TYPE may be found in "NFC Tags A technical introduction, applications and products Rev. 1.3 - 1 December 2011 White paper", NXP Semiconductors.

# NFC TAGs

## Format NDEF pour les tags passifs

- **Type 1**
  - Basé sur la norme ISO 14443-A
  - Innovision Topaz, Broadcom BCM20203
- **Type 2**
  - Similaire au Type 1
  - Basé sur la norme ISO 144413-A
  - Compatible avec NXP MIFARE Ultralight.
- **Type 3**
  - Similaire au Type1
  - Basé sur le standard japonais (JIS) X 6319-4.
  - Compatible avec Sony Felica
- **Type 4**
  - Similaire au type 1
  - Basé sur la norme ISO 14443-A
  - Compatible avec le standard ISO 14433-4 (mode APDU)
- **NXP tag**
  - Mifare Classic

## Les services NDEF LLCP

- SNEP: Simple NDEF Exchange Protocol
- SNEP requêtes et réponses
- LLC service access point address 4
- Nom du service “urn:nfc:sn:snep”

# Exemple de tag type 2 Mifare

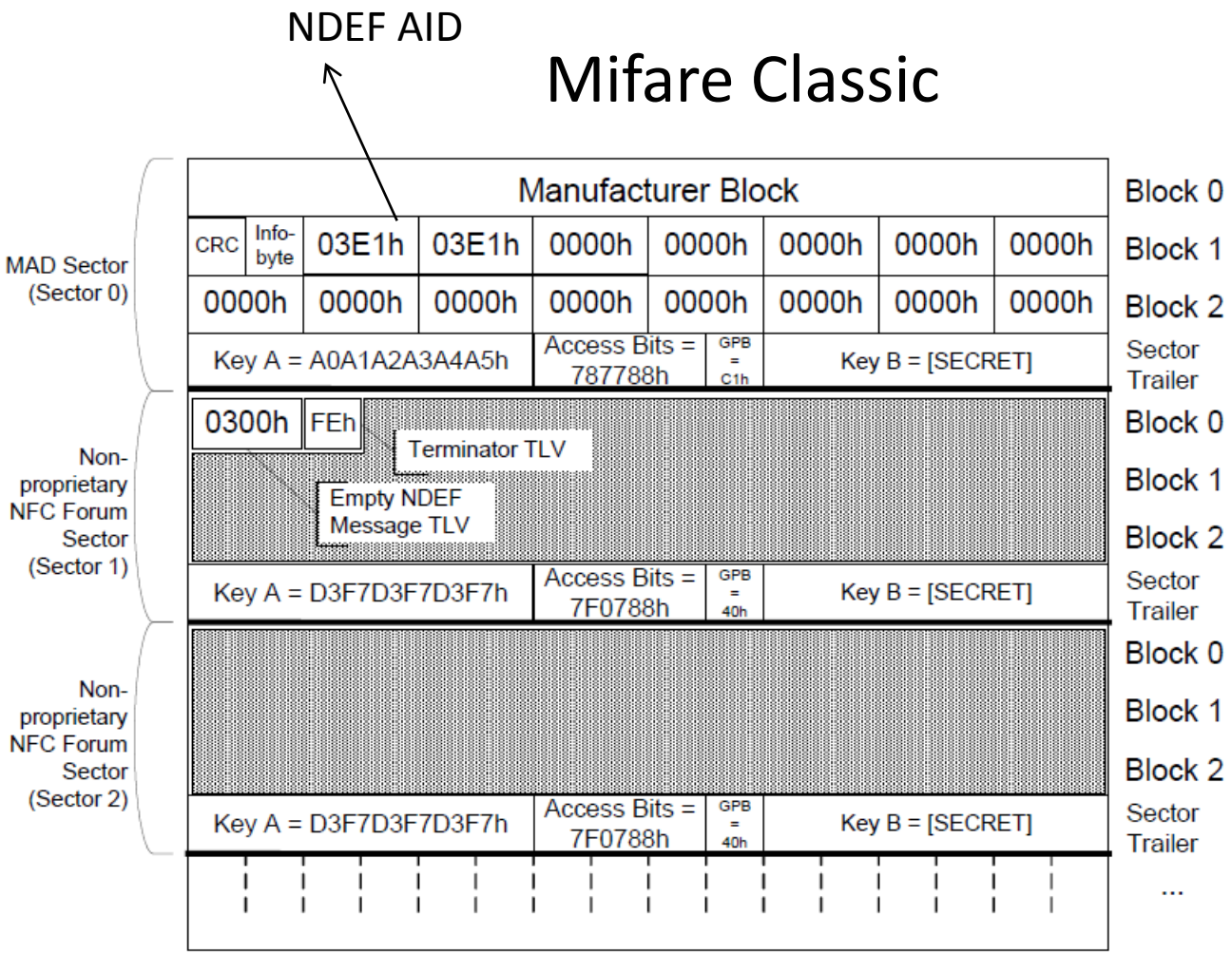
## Mifare Ultralight

| Block | Hex | | ASCII |
|---|---|---|---|
| 0 | 04D3 | E0BF | .Óà¿ |
| 1 | 0277 | 1E80 | .w.€ |
| 2 | EB48 | 0000 | ëH.. |
| 3 | E110 | 0600 | á... |
| 4 | 030E | D101 | ..Ñ. |
| 5 | 0B54 | 0265 | .T.e |
| 6 | 6E53 | 616D | nSam |
| 7 | 706C | 6520 | ple |
| 8 | FE00 | 0000 | þ... |
| 9 | 0000 | 0000 | .... |
| 10 | 0000 | 0000 | .... |
| 11 | 0000 | 0000 | .... |
| 12 | 0000 | 0000 | .... |
| 13 | 0000 | 0000 | .... |
| 14 | 0000 | 0000 | .... |
| 15 | 0000 | 0000 | .... |

Type2 Tag

Size 48 bytes

## Mifare Classic




Mifare Application Directory, MAD

# Mifare Classic 1K

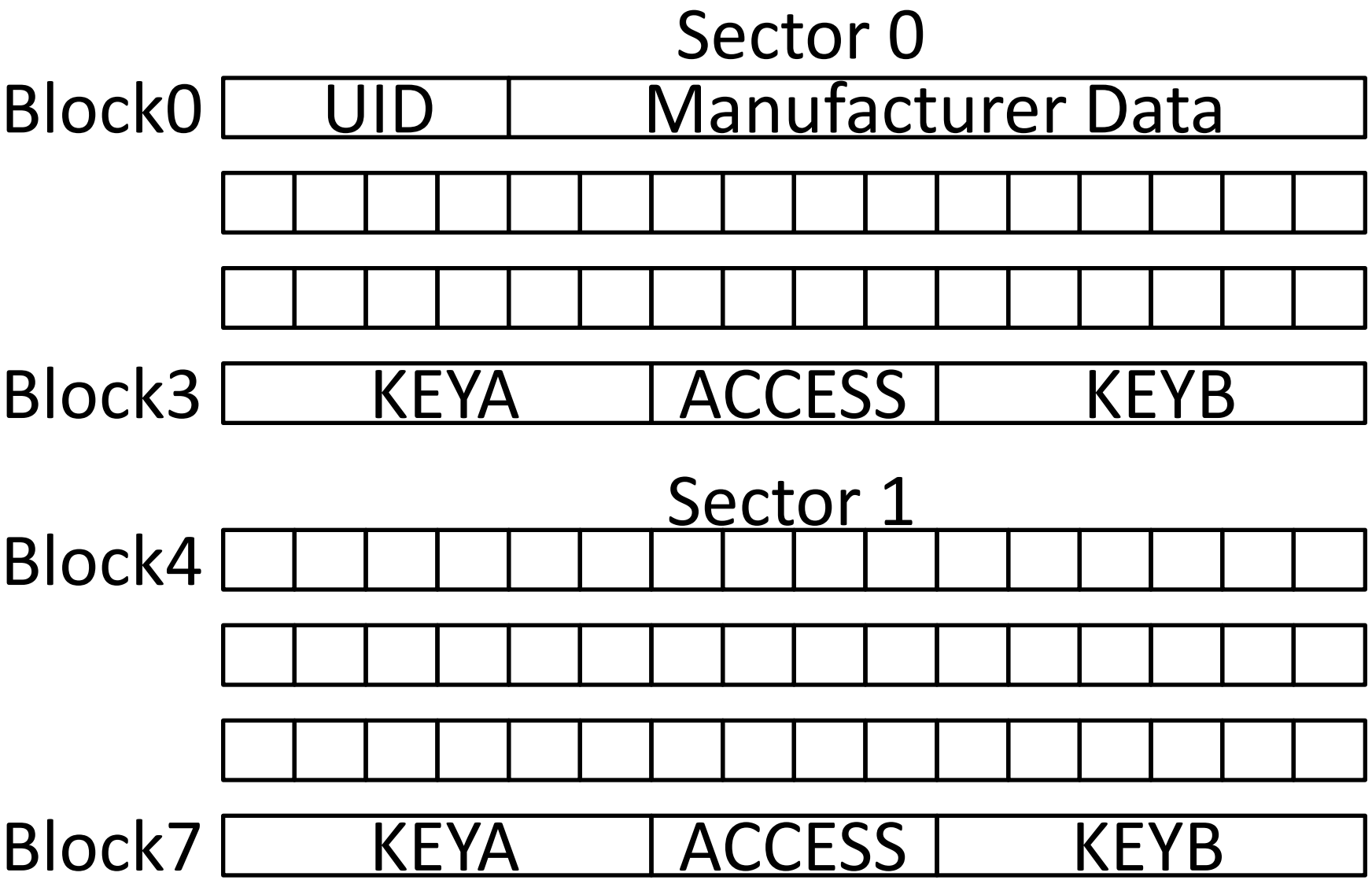


**Table 6. Access conditions**

| Access Bits | Valid Commands | | Block | Description |
|---|---|---|---|---|
| $C1_3$, $C2_3$, $C3_3$ | read, write | → | 3 | sector trailer |
| $C1_2$, $C2_2$, $C3_2$ | read, write, increment, decrement, transfer, restore | → | 2 | data block |
| $C1_1$, $C2_1$, $C3_1$ | read, write, increment, decrement, transfer, restore | → | 1 | data block |
| $C1_0$ ,$C2_0$, $C3_0$ | read, write, increment, decrement, transfer, restore | → | 0 | data block |

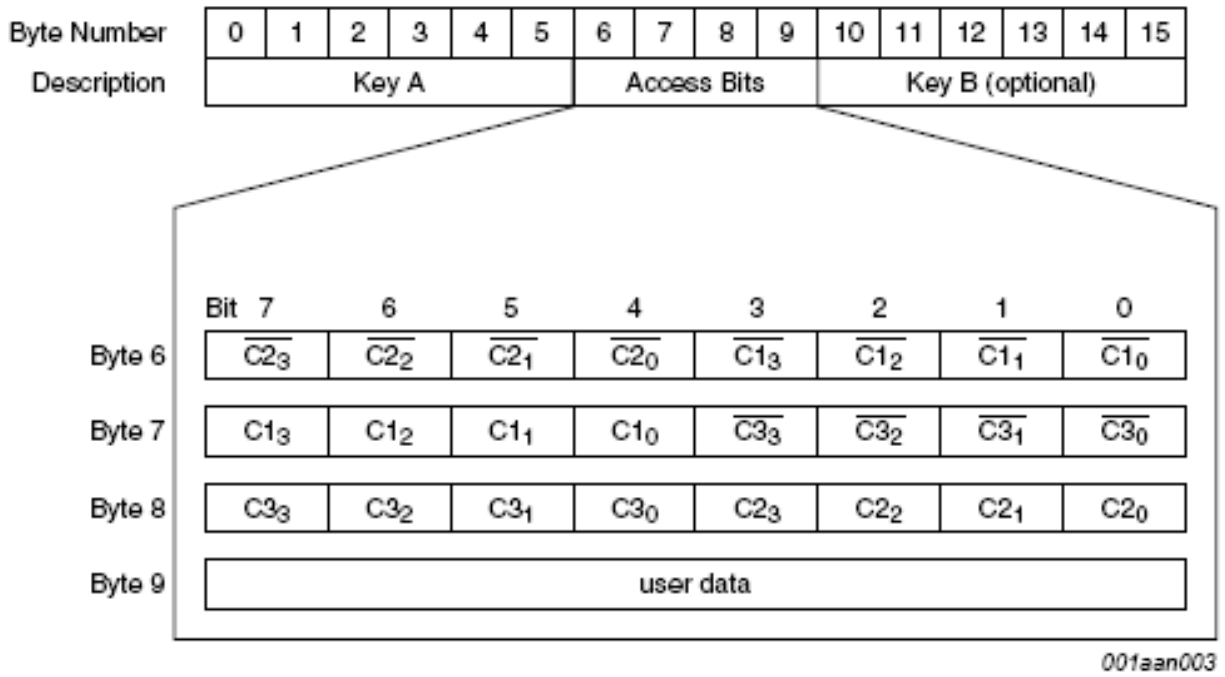


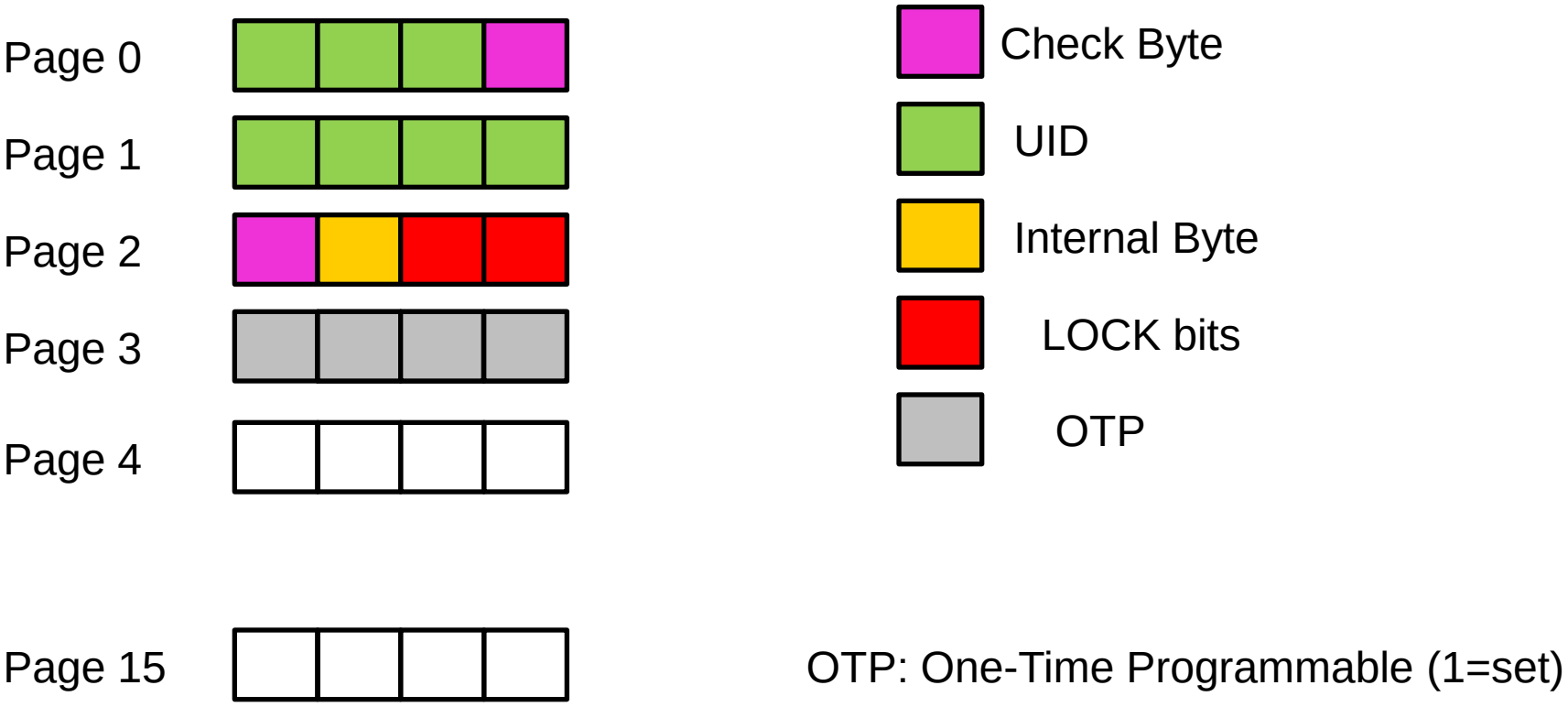

**Fig 10. Access conditions**

**Table 7. Access conditions for the sector trailer**

| Access bits | | | Access condition for | | | | | | Remark |
|---|---|---|---|---|---|---|---|---|---|
| | | | KEYA | | Access bits | | KEYB | | |
| C1 | C2 | C3 | read | write | read | write | read | write | |
| 0 | 0 | 0 | never | key A | key A | never | key A | key A | Key B may be read[1] |
| 0 | 1 | 0 | never | never | key A | never | key A | never | Key B may be read[1] |
| 1 | 0 | 0 | never | key B | key A\|B | never | never | key B | |
| 1 | 1 | 0 | never | never | key A\|B | never | never | never | |
| 0 | 0 | 1 | never | key A | key A | key A | key A | key A | Key B may be read, transport configuration[1] |
| 0 | 1 | 1 | never | key B | key A\|B | key B | never | key B | |
| 1 | 0 | 1 | never | never | key A\|B | key B | never | never | |
| 1 | 1 | 1 | never | never | key A\|B | never | never | never | |

[1] For this access condition key B is readable and may be used for data

# Mifare UltraLight




OTP: One-Time Programmable (1=set)

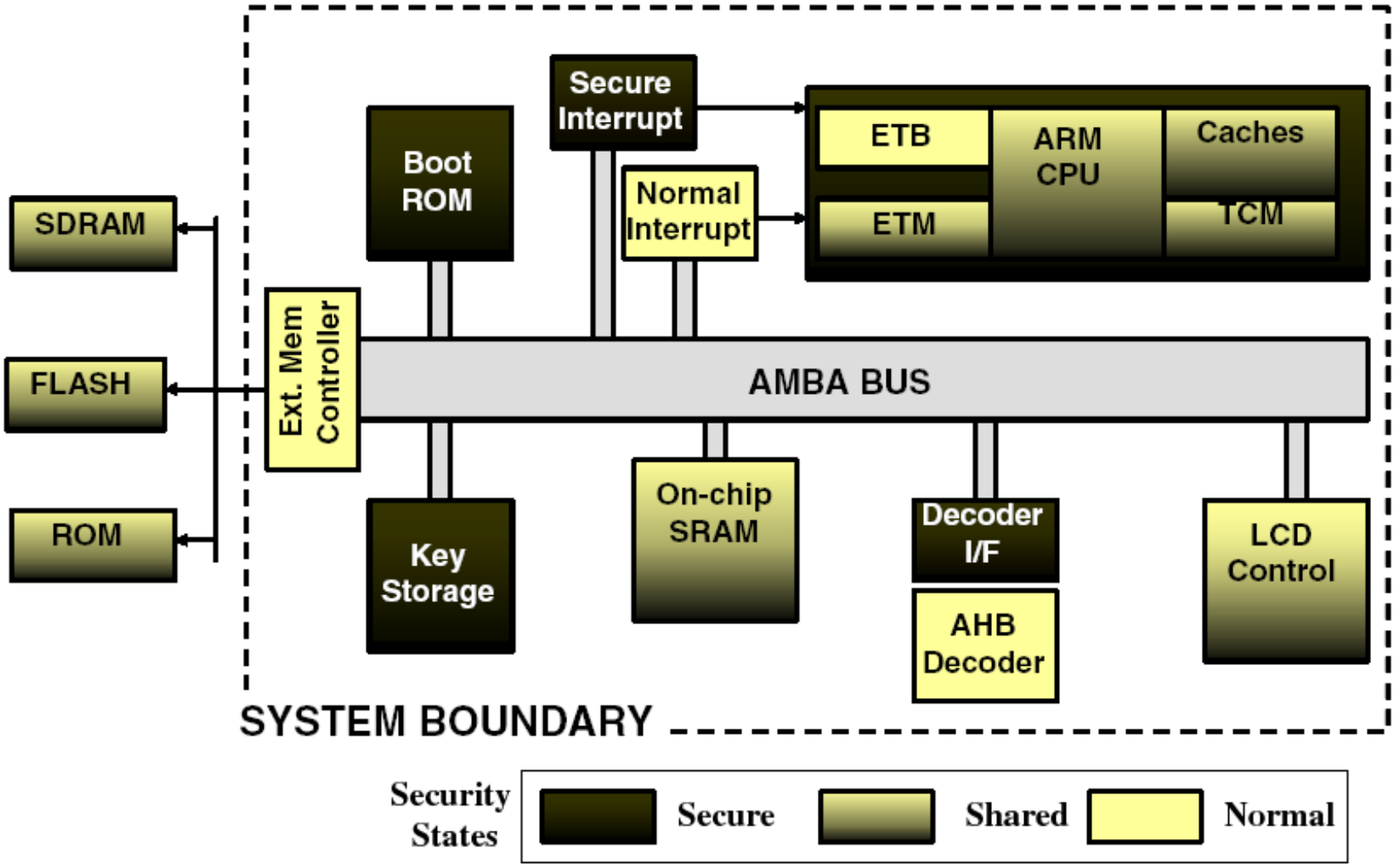
LOCK bits

| L7 | L6 | L5 | L4 | L OTP | BL 15-10 | BL 9-4 | BL OTP |
|---|---|---|---|---|---|---|---|

| L15 | L14 | L13 | L12 | L11 | L10 | L9 | L8 |
|---|---|---|---|---|---|---|---|

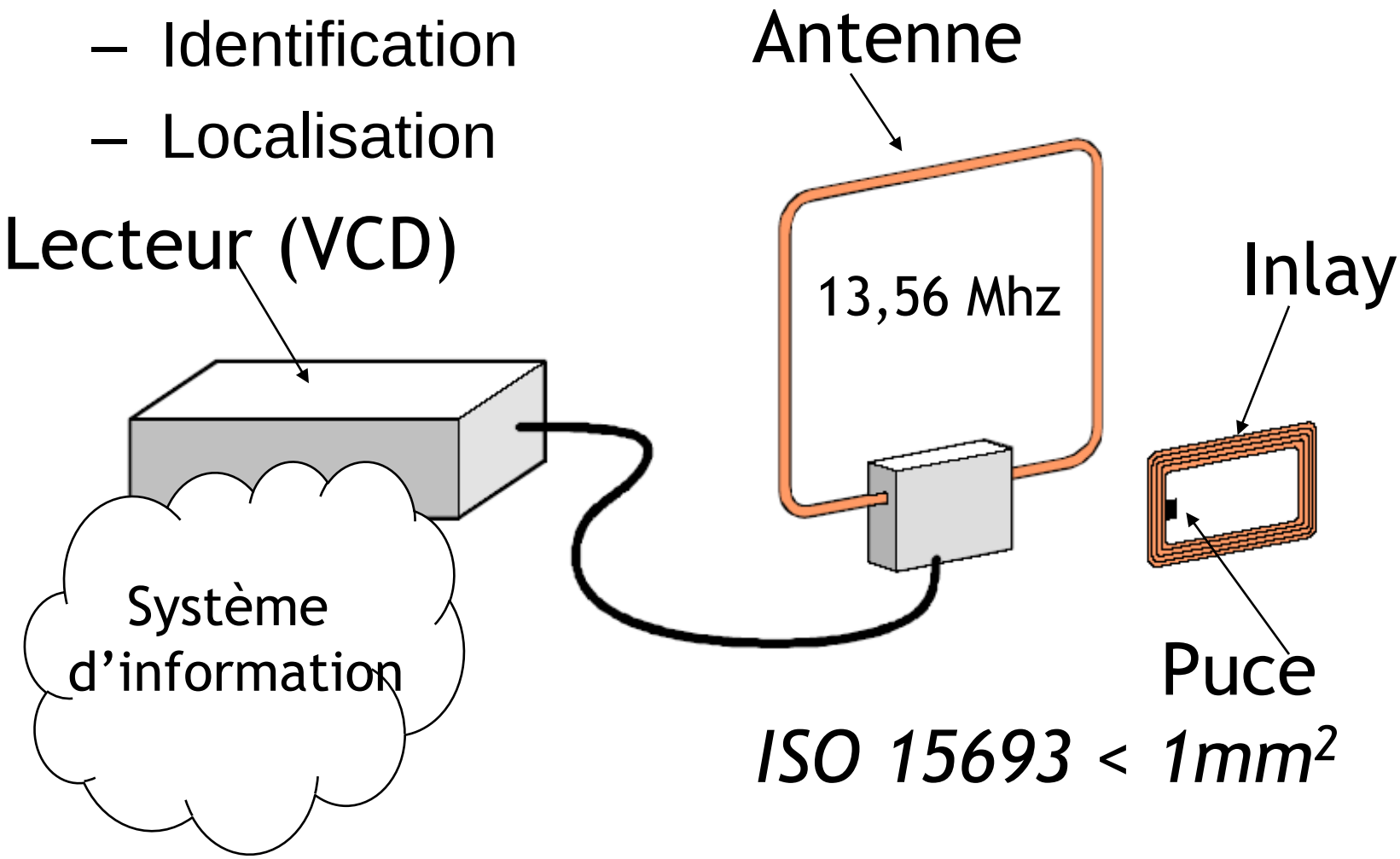
# De nouveaux circuits sécurisés

# ARM TrustZone

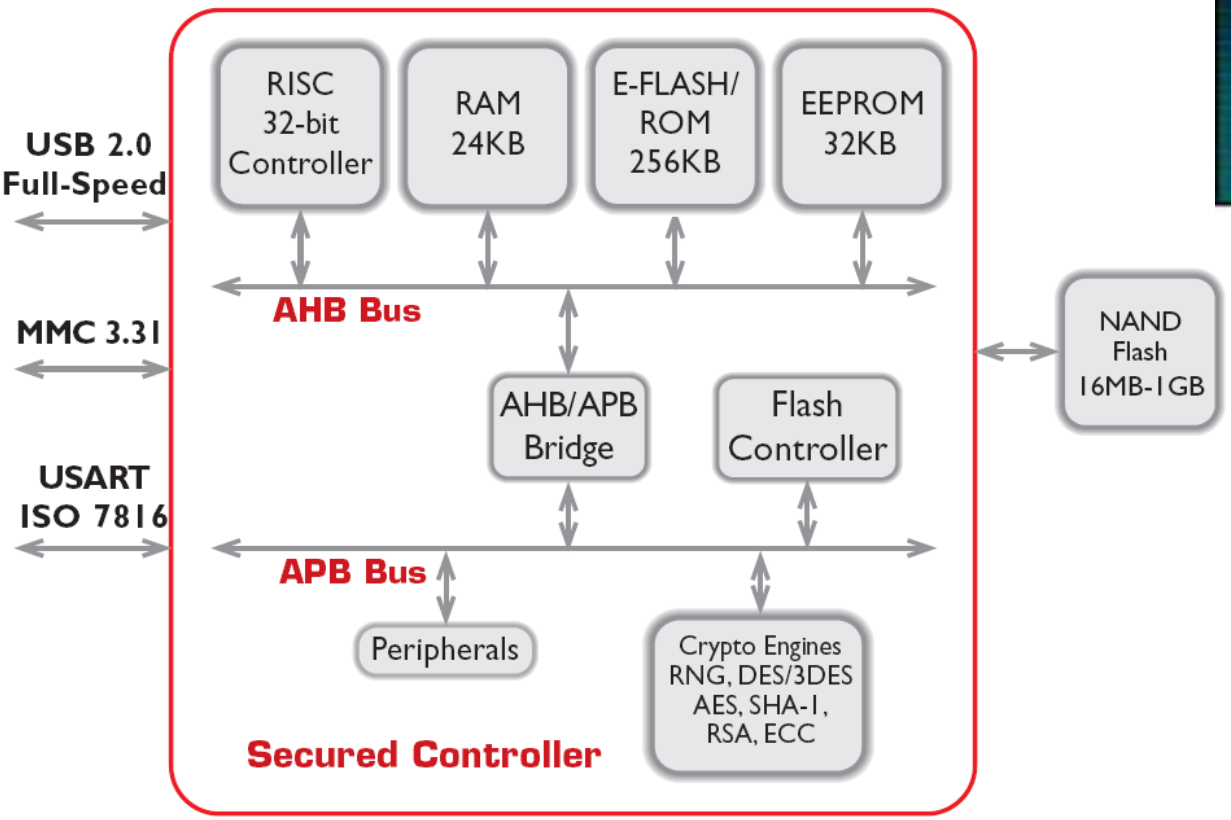


# RFID - ISO 15693

- Services
  - Identification
  - Localisation

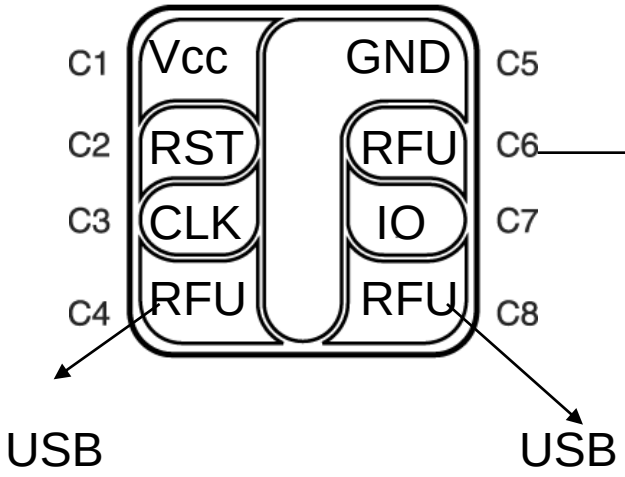


# MegaSIM

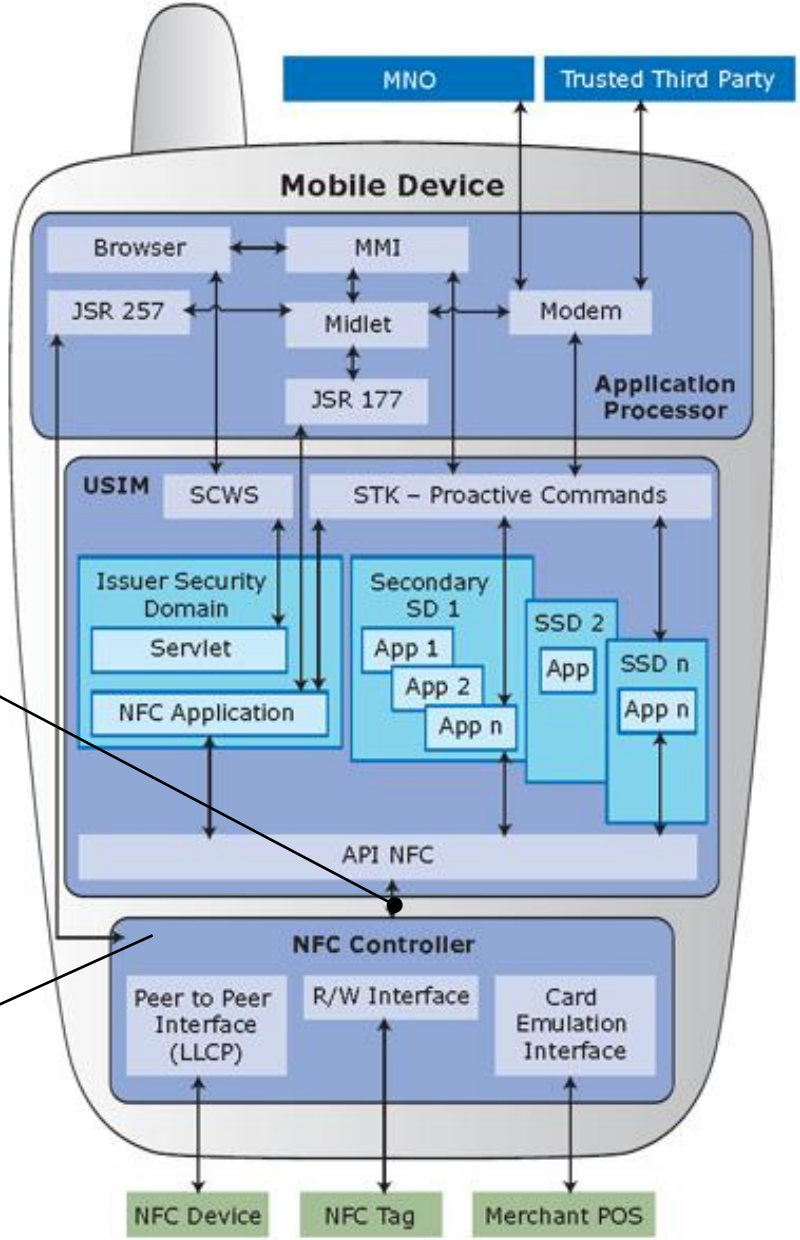

www.m-systems.com

# NFC




SWP Single Wire Protocol

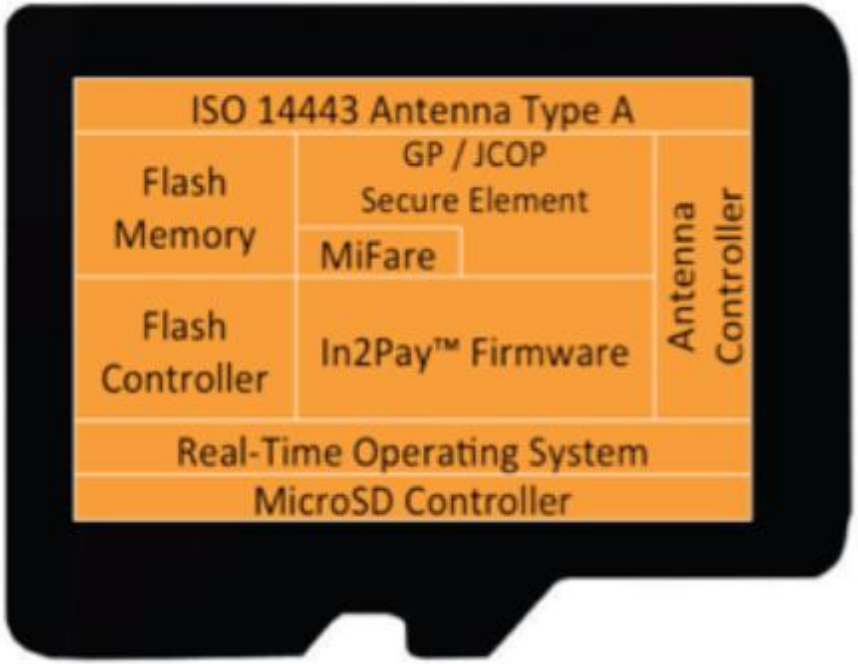
NFC Router



# SD Card with NFC Controller


http://www.tyfone.com/

EEPROM 72 Ko




http://www.devicefidelity.com/

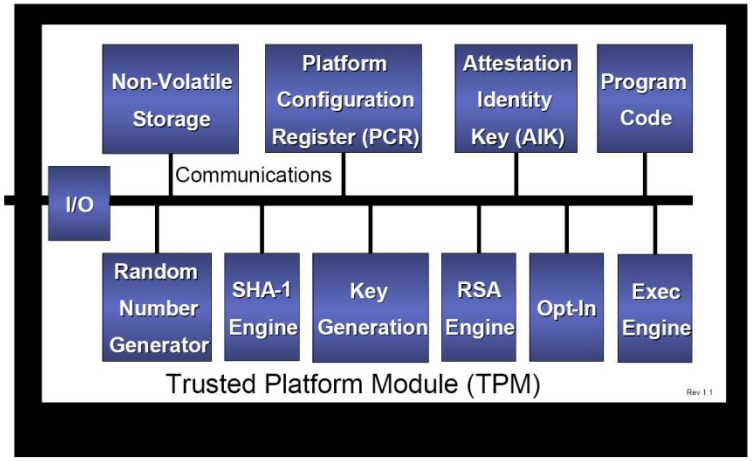
**Secure Element**

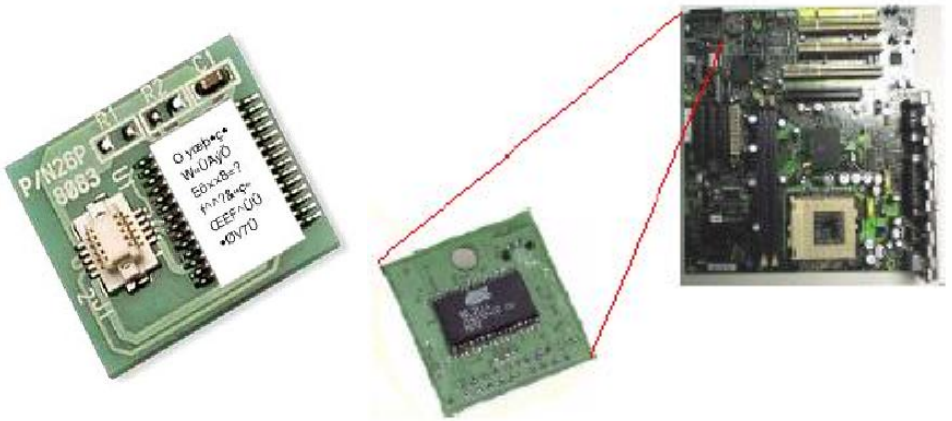
| | |
|---|---|
| ISO 7816, T=0, T=1 (communication speed, kbit/s | 223.1 |
| ISO 14443 T=CL (communication speed, kbit/s) | 424 |
| Available EEPROM Options kBytes | 72 |
| MiFare 1K | Yes |
| ROM (free for applets, up to kBytes) | 65 |
| APDU Buffer (RAM/bytes) | 261 |
| Java Card Version | 2.2.1 |
| Global Platform | 2.1.1 |
| SCP Secure Channel Protocol | SCP01, SCP02 |
| DES/TDES (bit) | 56/112/168 |
| RSA (bit) | 2432 |
| SHA | SHA-1 |
| Random Number Generator | Yes |
| RSA Key Generation | Yes |
| Logical Channels | 1 |
| VSDC 2.7.1 | Yes |
| VMPA 1.3.1 | Yes |
| MChip4 | Yes |
| Mobile PayPass 0.91 | Yes |
| Discover Zip | Yes |

# Trusted Platform Module - TPM

- Un module de sécurité lié à une carte mère (TPM, *Trusted Platform Module)*
  - Un composant proche d'une carte à puce (mêmes fondeurs)
- Un modèle de sécurité figé, TCG - *Trusted Computing Group*.

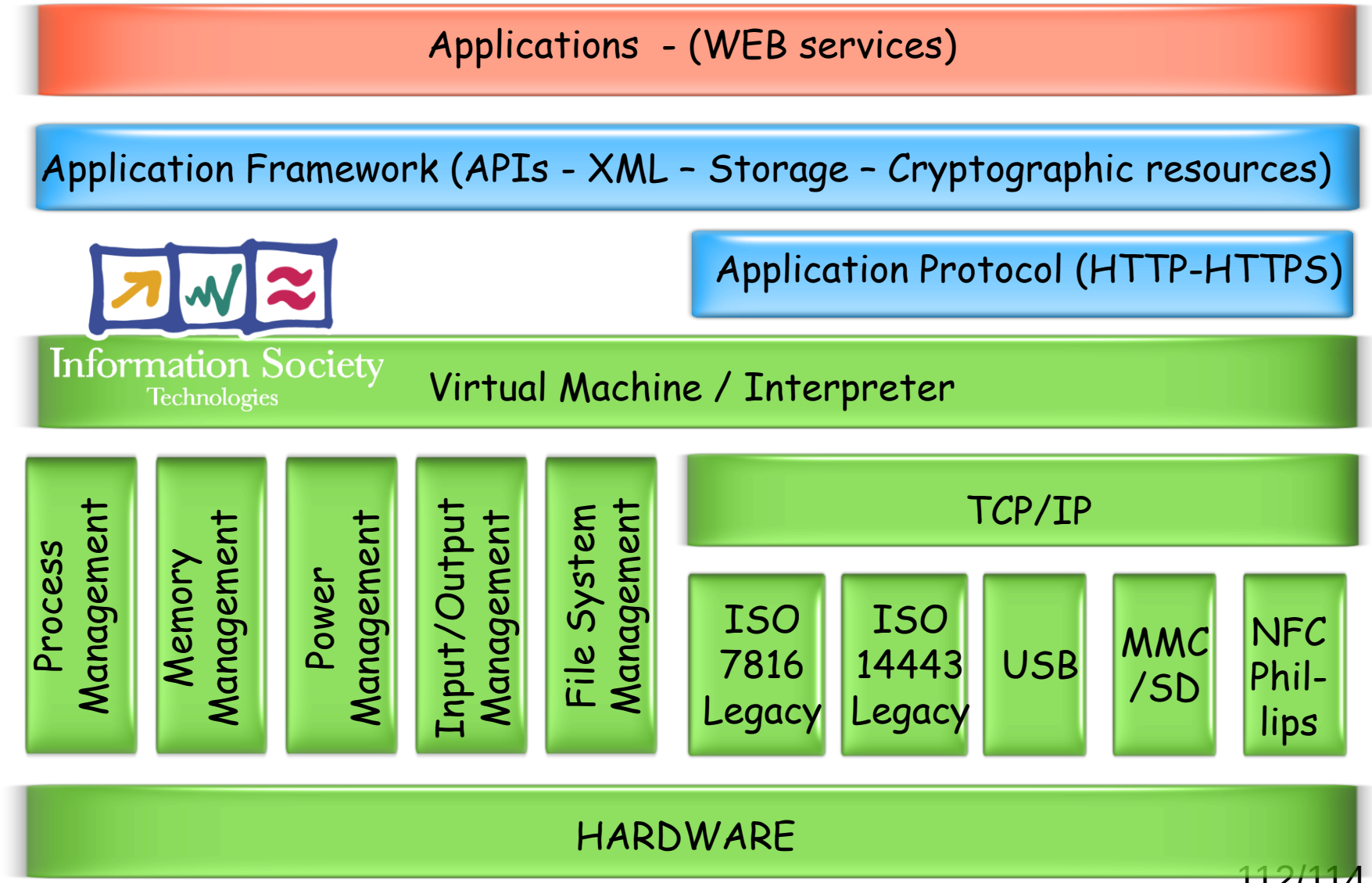
| Attaque | Solution courantes | Défauts | Apports TPM |
|---|---|---|---|
| Vol de données | Chiffrement, intégrité (VPN, …) | Stockage des clés dans des espaces non sures | Stockage sécurisé des données |
| Accès non autorisés à une plateforme | 1)Login, mot de passe<br>2) Biométrie<br>3)Token externes (cartes à puce…) | 1) Attaques par dictionnaire<br>2) Fiabilité des techniques biométriques<br>3) Crédits d'authentification indépendants de la plateforme | Protection des données d'authentification liée à la plateforme. |
| Accès au réseau non autorisé | Windows network logon, IEEE 802.1X | Données d'authentification stockées dans un espace non sure | Stockage sécurisé des données d'authentification. |



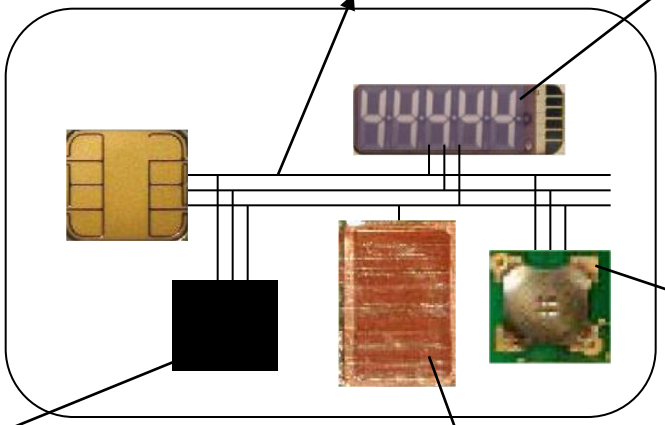



– Telecom ParisTech

# Next Smart Card Generation: the TPD - Trusted Personal Device



# System on Card (SoC)



**Bus**

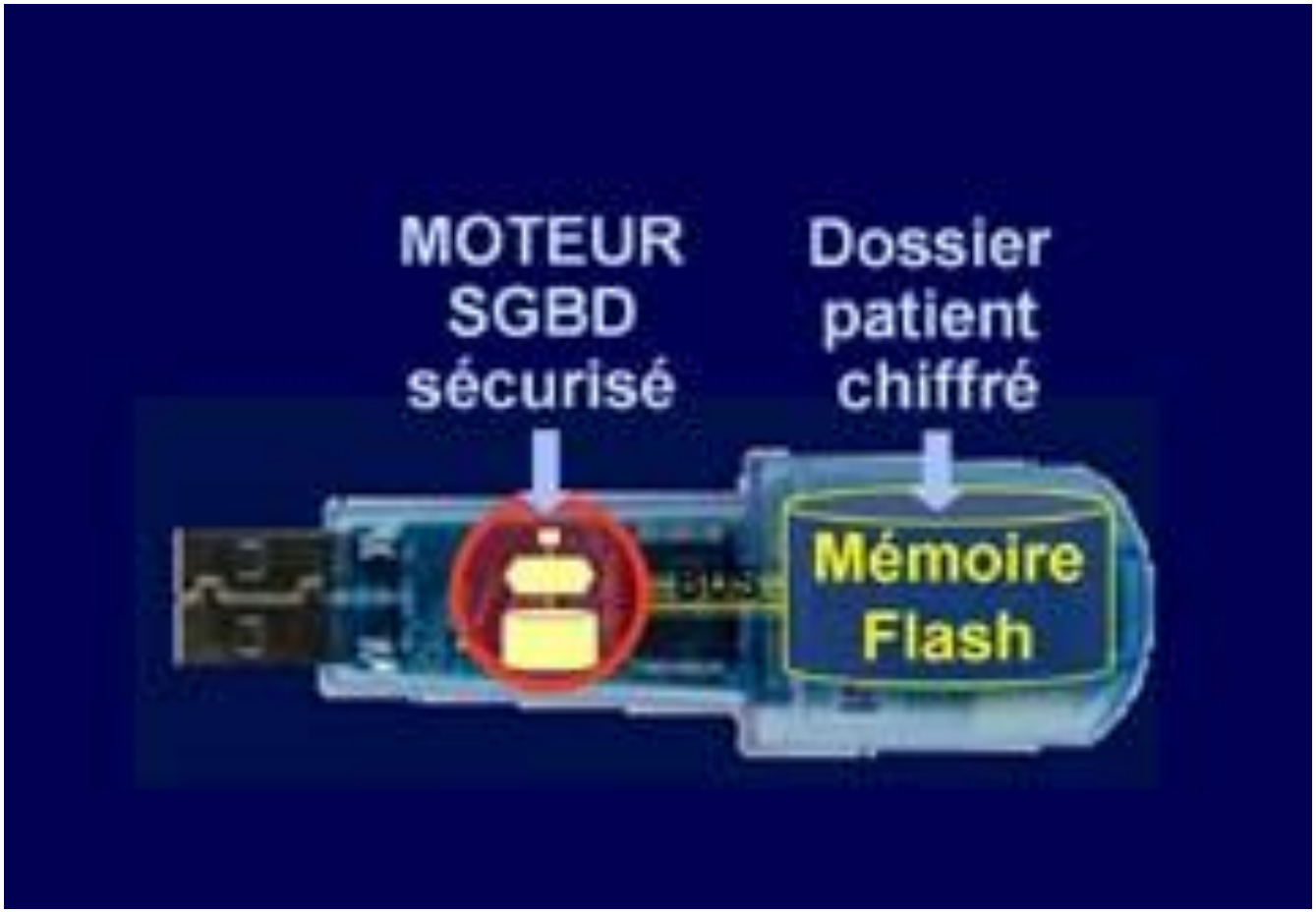
**Output**
- Displays
- LEDs
- Loudspeaker
- ...

**Input**
- Keypads/Knobs
- Sensors
- Loudspeaker
- ...

**Performance Enhancement**
- Microcontroller
- Memory
- ...

**Power Supply**
- Batteries
- Solar cells
- ...


# PicoDBMS




http://www.yvelines-competences.com/actualites/inria-cg-dossier-medico-social-200706.asp