Les Cartes à Puce

Objectifs : Acquérir des notions techniques suffisantes sur la technologie, le fonctionnement et l'utilisation des cartes à puces en vue de son implémentation dans des projets sur les systèmes électroniques embarqués.

Connaissances préalables : Architecture des systèmes à microcontrôleurs et électronique numérique.

Contenu de la matière : Les Chapitres	
1- *Généralités.*	**4-** *Construction..*
2- *Semiconducteurs pour cartes à puces*	**5-** *Systèmes d'exploitation*
3- *Cryptologie et sécurité*	**6-** *Les principaux Normes*

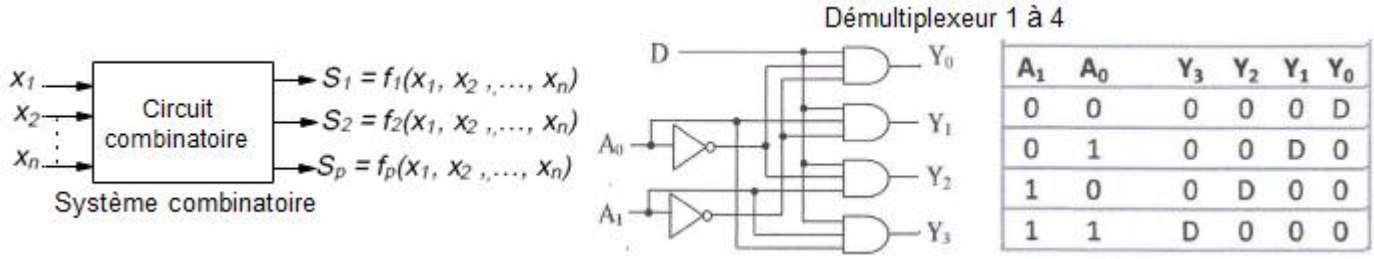
Références : **1** : W. Rankl, Smart Card Handbook, Wiley, 2010. **2** : C. Tavernier, « Les cartes à puce », Dunod, 2011. Jean-Pierre TUAL « Cartes à puces » techniques des ingénieurs E3440**.** Lilian BOSSUET « Sécurité des systèmes embarqués » techniques des ingénieurs -H8280.

1 Généralités

1.1 Rappel sur la logique classique et la logique programmable

Système combinatoire : C'est tout système numérique (logique) dont les sorties $(S_1, S_2, \cdots, S_p)$ sont exclusivement définies à partir des variables d'entrée$(X_1, X_2, \cdots, X_n)$. Les fonctions combinatoires sont généralement les opérateurs logiques suivants : NOT, AND, OR, NAND, NOR, XOR et NXOR

Exemples

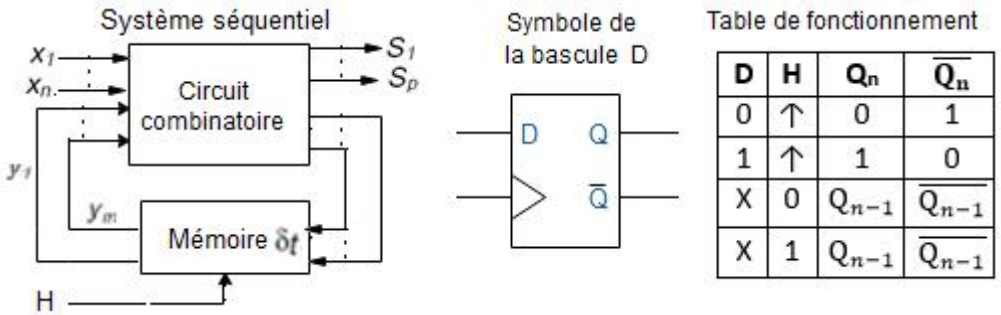


A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0

Figure 1.1 : Exemple de système combinatoire

Système séquentiel : C'est un système logique dont l'état des sorties $(S_1, S_2, \cdots, S_p)$ ne dépendent pas uniquement de l'état des entrées $(X_1, X_2, \cdots, X_n)$; mais également de l'état des variables internes $(Y_1, Y_2, \cdots, Y_m)$. Ce type de système nécessite pour fonctionner des mémoires. On utilise souvent une bascule qui est dite Asynchrone lorsque sa sortie change d'état uniquement fonction de des variables d'entrée. Elle est Synchrone lorsque sa sortie change après avoir eu l'autorisation d'un signal H de synchronisation appelé horloge.

Exemples



D	H	Q_n	$\overline{Q_n}$
0	↑	0	1
1	↑	1	0
X	0	Q_{n-1}	$\overline{Q_{n-1}}$
X	1	Q_{n-1}	$\overline{Q_{n-1}}$

Figure 1.2 : Schéma de principe d'un système séquentiel et la Bascule D (Mémoire)

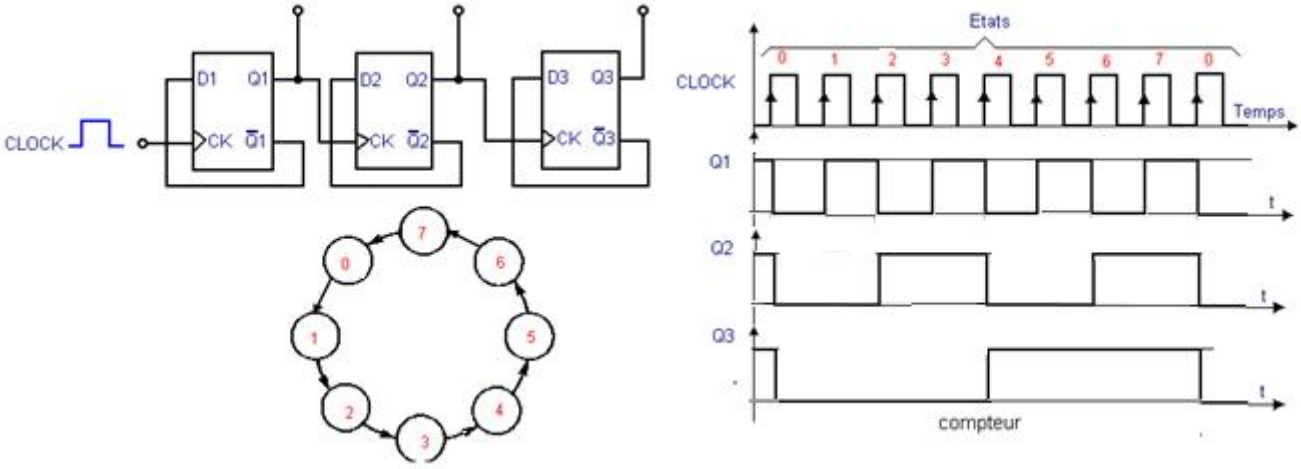


Figure 1.3 : Exemple de système séquentiel (Compteur)

La Mémoire : C'est un dispositif permettant de recueillir et de conserver les informations. Ses cellules mémoires se présentent avec une interface simple, limitée à quelques broches. Dans le cas général on peut grouper les broches d'une cellule mémoire (Figure 1.4) en trois types :

- Les **broches de données** : sur lesquelles on va lire ou écrire un bit (0 ou 1 logique).
- Les **broches d'adresse** : sur lesquelles on sélectionne la position de la donnée qu'on va lire ou écrire
- Les **broches de commande** : sur lesquelles on envoie des ordres tels que lecture/écriture, le signal d'horloge, ou les broches pour l'alimentation électrique et la masse.

Par convention une position contient un registre de 8 bits appelé Octet. La taille d'une mémoire est donnée en Octet ou Byte. Comme exemple 1 Ko(Kilo-Octet)=2^{10}=1024 positions, 1 Mo(Méga-Octet)=2^{20} positions.

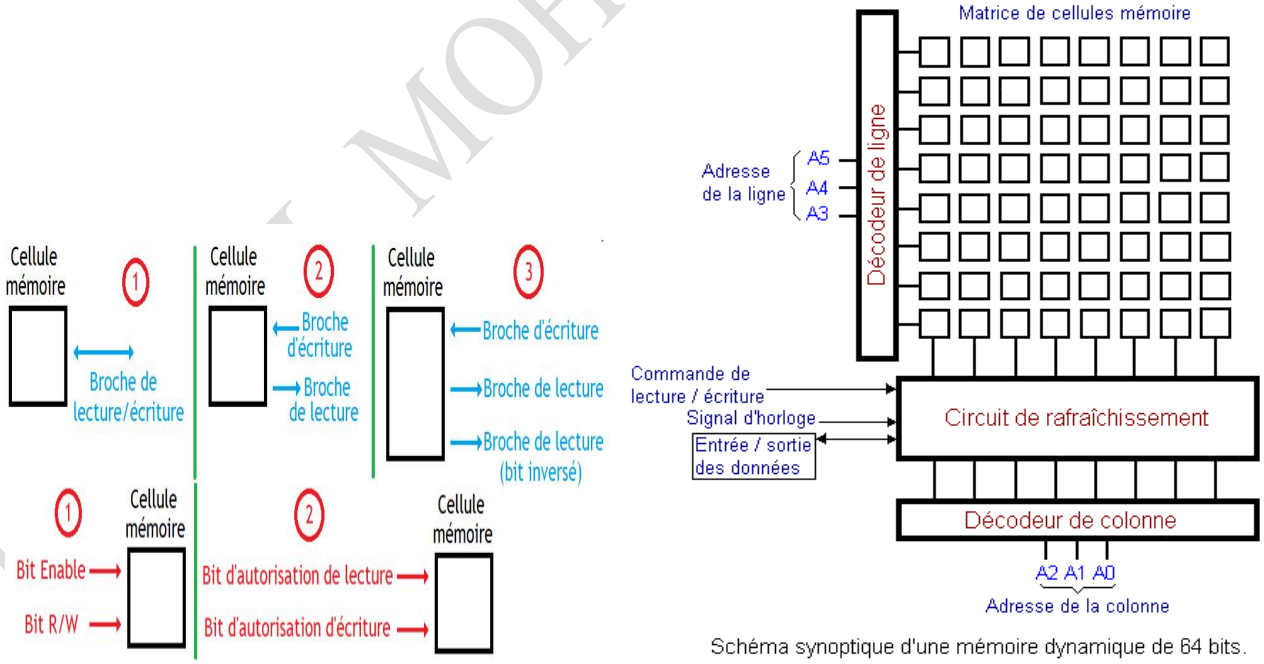
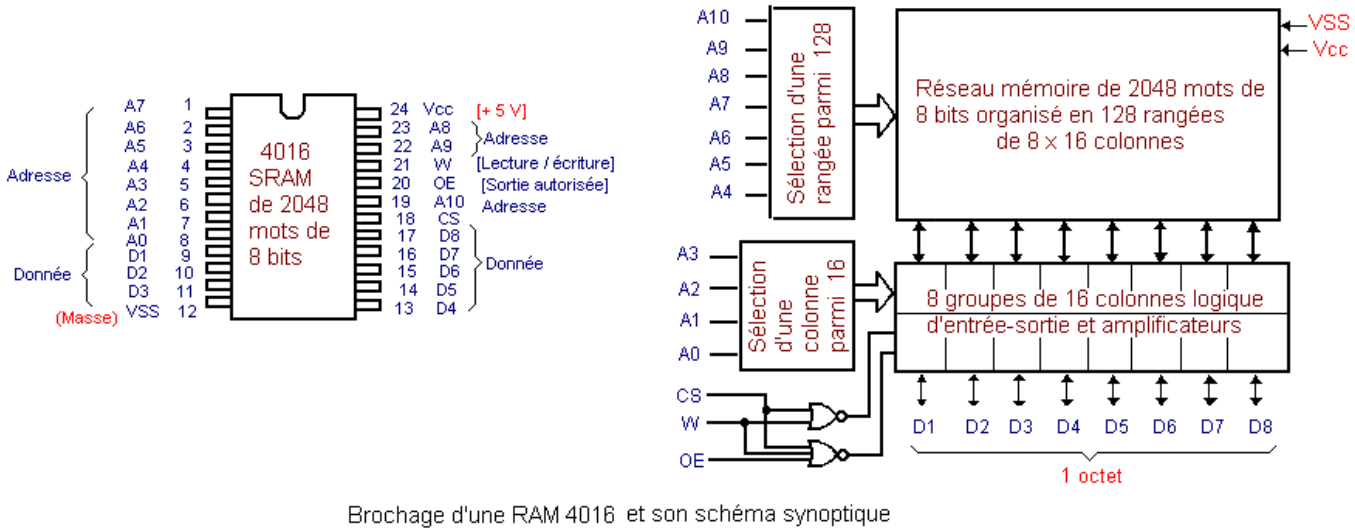


Figure1.4a : Schéma synoptique d'une mémoire dynamique



Brochage d'une RAM 4016 et son schéma synoptique

Figure 1.4b : Exemple d'une mémoire dynamique

Exemples d'applications

RAM (0000h à 1FFFh), ROM (F000h à FFFFh), I/O1 (8000h à 80FFh), I/O2 (B200h à B2FFh)

Utilisation de la logique combinatoire un Démultiplexeur 3 à 8 le 74LS138

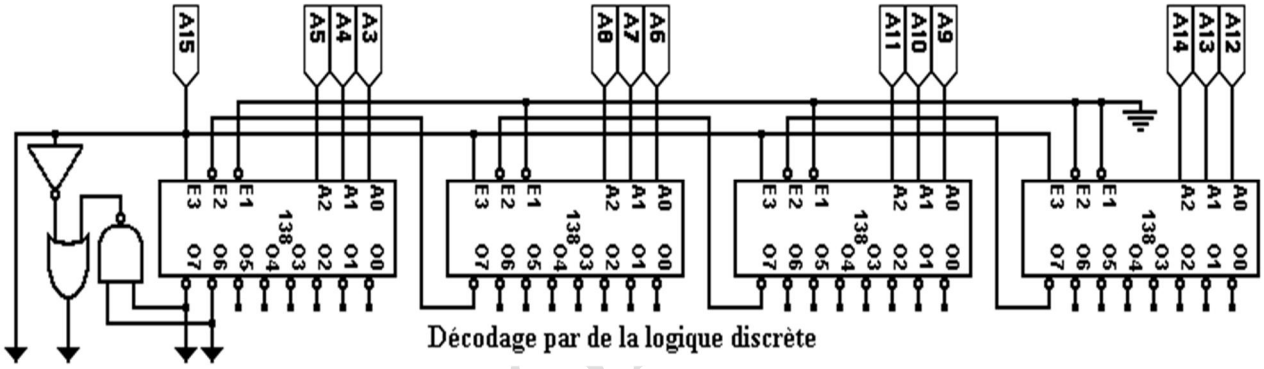
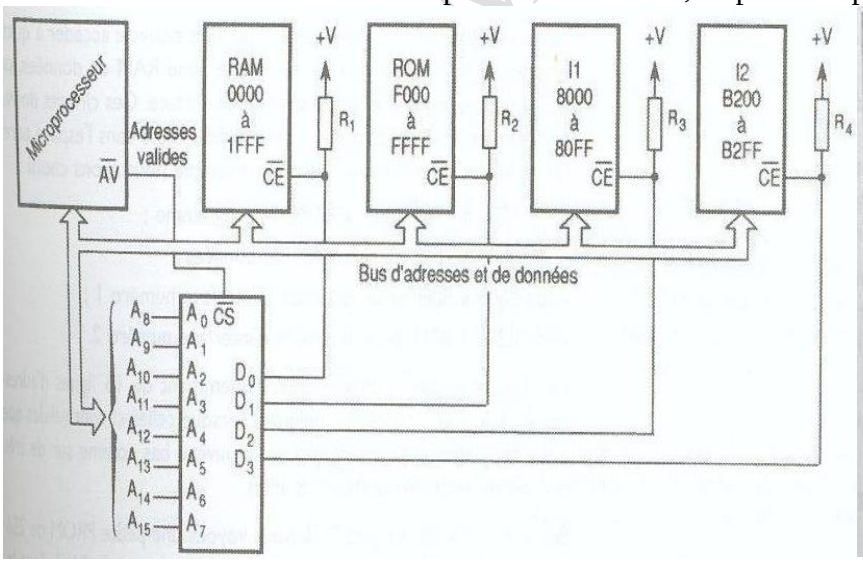


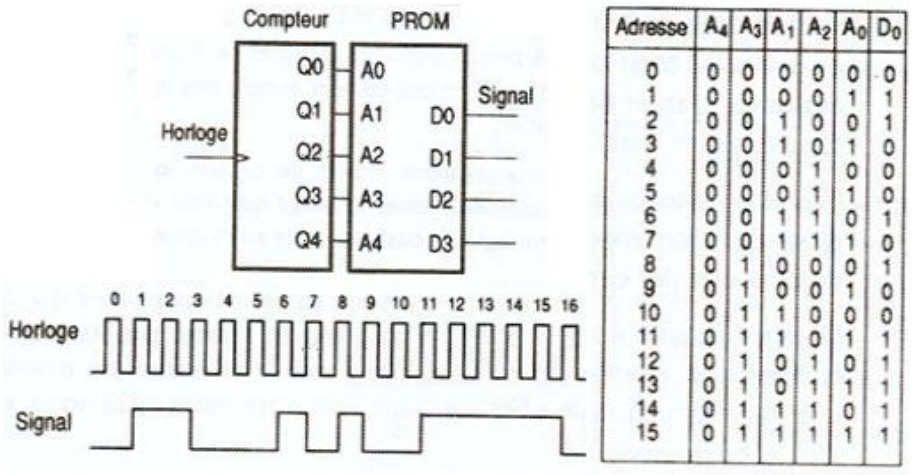
Figure 1.5 : Exemple décodage d'adresse (circuit logique)

Inconvénient : Consomme pas mal de courant, trop de composants….



Adresse micro	Adresse mémoire	D0	D1	D2	D3
0000 à 1FFF	00 à 1F	0	1	1	1
2000 à 7FFF	20 à 7F	1	1	1	1
0 à 80FF	80	1	1	0	1
8100 à B1FF	81 à B1	1	1	1	1
0 à B2FF	B2	1	1	1	0
B300 à EFFF	B3 à EF	1	1	1	1
F000 à FFFF	F0 à FF	1	0	1	1

Figure 1.6: Exemple décodage d'adresse avec mémoire



Adresse	A_4	A_3	A_1	A_2	A_0	D_0
0	0	0	0	0	0	0
1	0	0	0	0	1	1
2	0	0	1	0	0	1
3	0	0	1	0	1	0
4	0	0	0	1	0	0
5	0	0	0	1	1	0
6	0	0	1	1	0	1
7	0	0	1	1	1	0
8	0	1	0	0	0	1
9	0	1	0	0	1	0
10	0	1	1	0	0	0
11	0	1	1	0	1	1
12	0	1	0	1	0	1
13	0	1	0	1	1	1
14	0	1	1	1	0	1
15	0	1	1	1	1	1

Figure 1.7: Exemple d'un générateur de signaux complexes avec mémoire

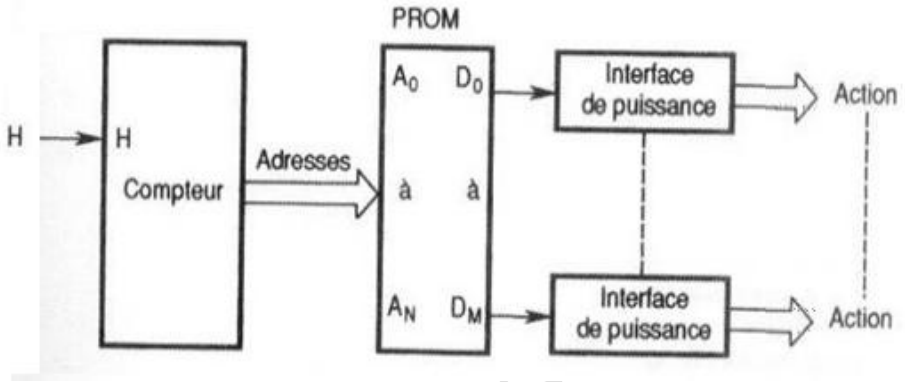
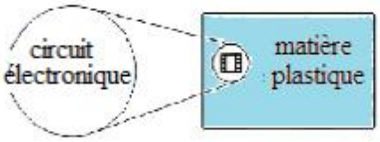


Figure 1.8 : Exemple d'une Automate programmé simplifié avec une mémoire

1.2 Introduction à la carte à puce



Le nom de carte à puces est souvent utilisé pour désigner des supports en matière plastique, papier ou carton, de quelques centimètres de côté et moins d'un millimètre d'épaisseur (carte) et qui contiennent un circuit électronique (puce) capable de mémoriser ou de traiter des informations.

Les cartes à puce sont présentes dans différents secteurs d'activités. Elles sont utilisées comme moyens d'identification personnelle (carte d'identité, badge d'accès aux bâtiments, carte d'assurance maladie, carte SIM) ou de paiement (carte bancaire, porte-monnaie électronique) ou preuve d'abonnement à des services prépayés (carte de téléphone, titre de transport,…). La carte à puce peut comporter un hologramme de sécurité pour éviter la contrefaçon. La lecture (ou l'écriture) des données est réalisée par des équipements spécialisés, certaines puces nécessitant un contact physique (liaisons électrique galvaniques), d'autres pouvant fonctionner à distance (communication par ondes radio).

L'évolution des cartes à puce on connue une utilisation croissante exponentielle depuis leurs première utilisation. La fabrication des cartes à puce fait appel à trois disciplines importantes la micro-électronique, l'informatique et la cryptographie. La puissance de calcul et les algorithmes de sécurité n'ont pas cessés d'évoluer en parallèle avec le perfectionnement du microcontrôleur et de l'informatique.

1.2 Fonctionnement et constitution de la carte

L'idée de la carte à puce est simple ; elle repose sur le principe qu'il faut stocker des données sécurisées (protéger ces données contre toute modification indésirable) sur une carte facile à utiliser et à transporter.

La puce comprend généralement :

- Une mémoire (mémoire vive et mémoire morte) permettant de stocker des informations
- Un microprocesseur ou logique combinatoire permettant de traiter ces informations (protection, lecture et modification),
- Une interface entre la carte et un appareil capable de lire ses données

Les premières cartes à puces (cartes de téléphone et de stationnement) étaient dites « passives » : toutes les opérations étaient effectuées par le lecteur extérieur. Le perfectionnement des microprocesseurs a permis aux cartes à puce de disposer d'avantage de capacités (Ex : fonctions de cryptage), les rendant « actives ».

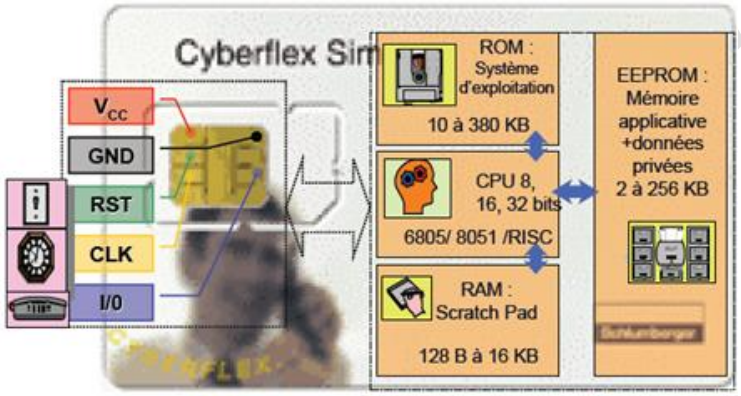


Figure 1.9 : Architecture d'une carte à puce

1.2.1 Historique

Dès 1967, l'utilisation d'un composant électronique doté d'une mémoire dans une carte à puce a fait l'objet de réflexions aux États Unis, au Japon et en Europe, comme en témoignent les très nombreux brevets.

Pendant longtemps, la carte à puce est restée un produit typiquement français boudée par les pays étrangers. Malgré cela, depuis 1985, le nombre de cartes n'a cessé d'augmenter. En effet en 2001 on compte : 41 millions de cartes bancaires ; 40 millions de cartes Vitale et surtout 1,5 milliard de cartes SIM (téléphonie mobile). Le Tableau suivant donne quelques années significatives dans l'évolution de la carte à puce

Tableau 1.1 - Quelques années significatives dans l'évolution de la carte à puce

Année	Événement
1967 à 1969	Deux ingénieurs allemands Jürgen Dethloff et Helmut Grotrupp inventèrent la carte à puce entre 1967 et 1968. Ils déposèrent d'ailleurs un brevet à ce sujet dès 1969
1970	Le Japonais Kunitaka Arimura de l'institut Arimura de Technology déposait au Japon un brevet relatif à la carte à puce.
1971	Paul Castrucci de chez IBM déposait un brevet intitulé *Information Card*
1974 à 1979	Roland Moreno utilise 47 brevets relatifs à la carte à puce dans 11 pays pour créer la société Innovatron développement et l'exploitation de ces brevets et applications

1979	Première carte à puce à base de microcontrôleur fabriquée par Motorola pour Bull CP8. Elle disposait d'une unité centrale de type 6805 (8bits) et d'une mémoire programmables de 1Ko. entreprise (CU-Honeywell-Bull). Cette carte à deux puces appelée CP8 fut essentielle pour prouver la faisabilité des concepts, convaincre les utilisateurs potentiels et lancer des expérimentations.
1980 à 1981	grâce aux efforts conjoints de Bull et de Philips la première expérimentation de télévision payante. Pas de grand débauché commercial
1981 à 1982	Premières expérimentations de paiement125 000 cartes bancaires (Bull/Philips/Schlumberger)
1983	Premières cartes à puce téléphoniques France Télécom. ces cartes n'exploitaient aucun microcontrôleur mais de simples mémoires comme. «Télécarte» (Schlumberger)
1984	Première version de la carte bleue (de payement) à puce à base de carte Bull CP8 Première «Telekarte» (G&D) Première expérience bancaire (Bull)
1987	Publication des normes ISO 7816 relatives à la carte à puce. Lancement à la mondialisation.
1988	Première carte multi-application (Bull) Première carte d'université (Bull) Première expérimentations bancaires
1989	Première expérimentations bancaires Première carte «club fidélité» Premières cartes GSM pour téléphones mobiles (Gemplus)
1998	Premières cartes à puce programmables en Java ou « Java Cards». Plusieurs puces par téléphone mobile. fin 2006 on compte au total de 3 milliards de cartes à puces (téléphonique).

1.3 Marchés de la carte à puce

Parmi les cartes à microcircuits, on peut distinguer deux familles de cartes :

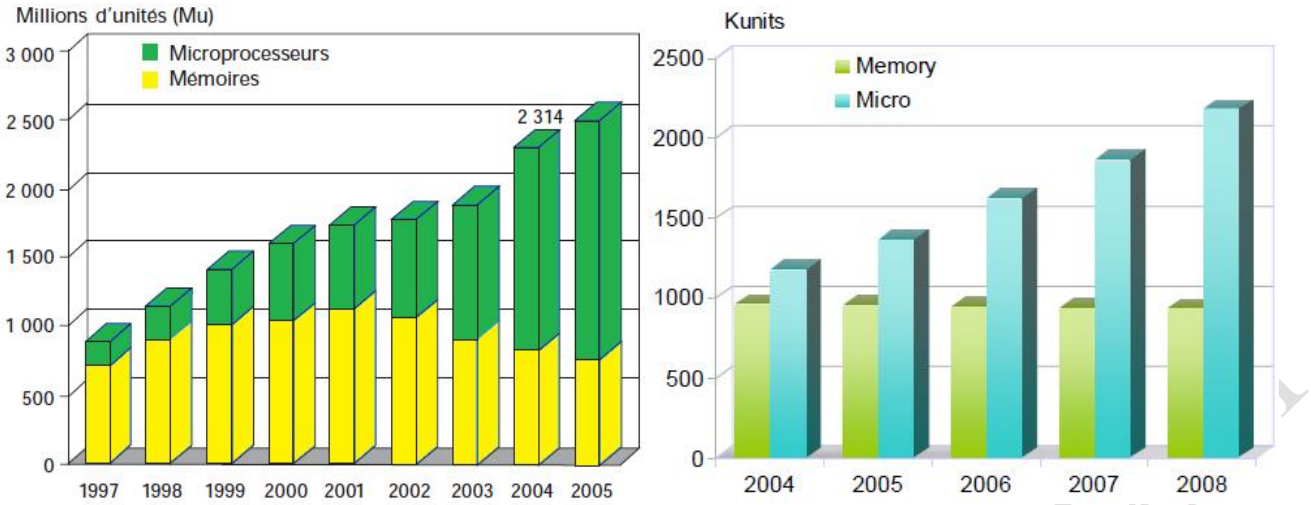
- *Les cartes à logique câblée et carte à mémoire* :

Carte à mémoire : C'est une mémoire simple (sans processeur) accessible en lecture sans protection, mais l'écriture peut être rendue impossible ; Programmation impossible ; c'est des Carte « porte-jetons » pour applications de prépaiement (carte téléphonique)

Carte à logique câblée : Sont des mémoires accessibles via des circuits préprogrammés et figés pour une application particulière. C'est une carte « sécuritaire » pouvant effectuer des calculs figés (accès à un local..).

- *Les cartes à microcalculateur (SmartCard) :* C'est une carte qui contient un microcontrôleur encarté (processeur + mémoires); Carte «programmable» pouvant effectuer tout type de traitements ; Interface électrique par contacts ou via signaux RF (Radio Fréquence).

La figure suivante illustre l quantité des cartes à logique câblée et à microcalculateur jusqu'à 2008



La figure 1.10 : La Quantités de cartes à logique câblée et à microcalculateur

1.3.1 Les cartes à logique câblée,

Quelques fonctions simples sont fixées par les circuits électroniques interposés entre la mémoire non volatile et l'interface extérieure. Dans le bas de gamme, il existe de multiples cartes à logique câblée centrées sur le prépaiement de services tels que le téléphone, le parking, le cinéma ou le lavage des voitures. Certaines cartes utilisent une simple mémoire non volatile à accès sérialisé et sans aucune protection sécuritaire.

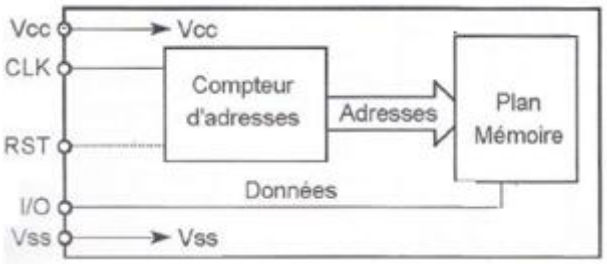


Figure 1.11 : Synoptique d'une carte à mémoire

Ces cartes supportent essentiellement des fonctions d'identification utilisées principalement par des applications implémentant des programmes de fidélisation, plus rarement par des services à haute niveau.

1.3.2 La carte à microcalculateur,

Elle, possède la même structure qu'un ordinateur. Elle permet non seulement de stocker des données mais aussi, et surtout, de traiter des informations de manière sécuritaire. En effet, ces deux fonctions sont réalisées par un programme qui est exécuté par un processeur central implanté sur un composant silicium. Ce dernier s'adapte ainsi à une large famille d'applications.

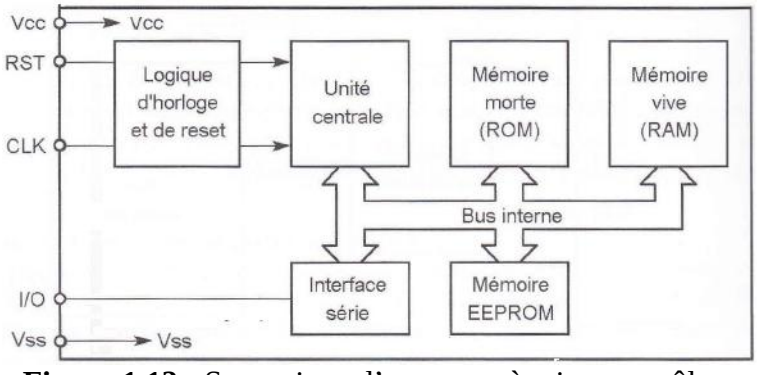


Figure 1.12 : Synoptique d'une carte à microcontrôleur

1.3.3 Domaine d'applications

Les domaines privilégiés de la carte à puces sont tous ceux où une très grande sécurité de traitement des informations est requise : le microcalculateur et son programme embarqué permettent, par exemple, d'authentifier la carte et son porteur, de chiffrer et de déchiffrer des messages, ou de calculer des signatures électroniques apportant la preuve de l'effective réalisation d'une opération licite. Les deux majeures applications actuelles de la carte à puces concernent d'une part l'authentification et l'identification sur les réseaux mobiles, et d'autre part le paiement électronique. La principale raison en est le compromis optimal offert par la technologie entre le coût de déploiement et le niveau de sécurité atteint pour une lutte efficace contre la fraude. Ces deux segments de marché (Téléphonie Mobile et Paiement) représentent aujourd'hui environ 85 % du marché global (en valeur) de la carte à puces.

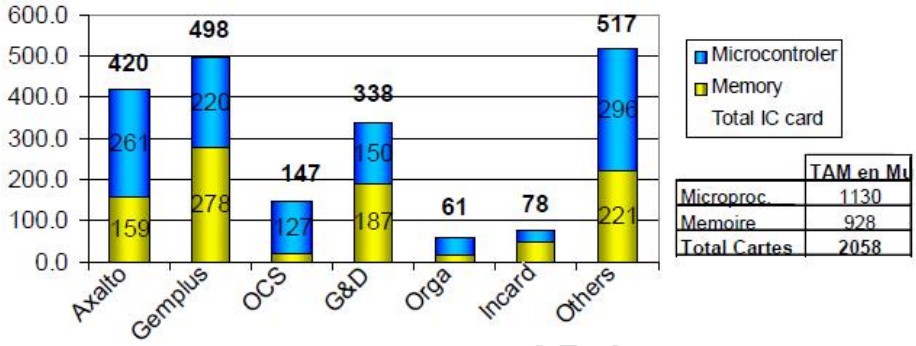
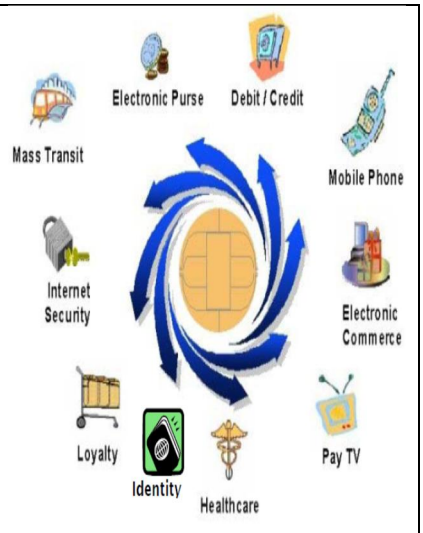


Figure 1.13 : Marché (des cartes à puce) et compétition en 2003 entre les acteurs majeurs de l'industrie

D'autres applications, actuellement en émergence, sont appelées à devenir à court ou moyen terme des relais de croissance pour le marché de la carte à puces : les applications gouvernementales (cartes d'identité, passeports électroniques, cartes de santé ...), la protection des ordinateurs et des réseaux d'entreprise, la télévision cryptée, la protection des droits d'auteur, les transports en commun. Signalons que, dans ce dernier cas, la puce électronique est le plus souvent pourvue d'une interface sans contact. Le tableau suivant donne les quantités de cartes à microcalculateur livrées dans le monde (sans introduire la chines) en 2005, telles qu'estimées par l'organisation EuroSmar.

Tableau 1.2 : Cartes à mémoire et à microcalculateurs par type d'application

Cartes à mémoire et à microcalculateurs par type d'application			
	Cartes (Millions d'unités-Mu)		
	Mémoires	**Microprocesseurs**	
			Croissance depuis 2004 (%)
Télécommunications	570	1310	25
Services financiers	24	330	18
Gouvernement-Santé	15	60	33
Transport	70	25	67
Télévision à la carte	–	60	9
Sécurité	20	15	25
Autres	10	12	–
TOTAL	709	1 812	23
TOTAL prévision pour l'année 2005	2 521		



1.4 Cartes à communication sans contact

Dans certaines applications, le circuit intégré possède une interface « sans contact », utilisant les mêmes principes que l'identification par radiofréquence (RFID : *Radio Frequency Identification Devices*). Dans ce cas, on utilise le principe de type transpondeur magnétique. Un lecteur envoie un signal radio à la carte qui elle-même contient une antenne déposée sur le substrat en plastique et connectée au circuit intégré. La puce est alors alimentée par le signal radio et va ainsi pouvoir communiquer avec le lecteur. Les intensités du champ magnétique ainsi créées sont comprises entre 1,5 et 7,5 A/m, la distance de fonctionnement entre le lecteur et la carte varie de 0 à 10 cm, la fréquence utilisée étant normalisée à 13,56 MHz. Cette technologie du « sans contact » trouve de nombreux champs d'applications tels que le paiement (y compris sur mobile), la billettique, les cartes d'identité, le contrôle d'accès...

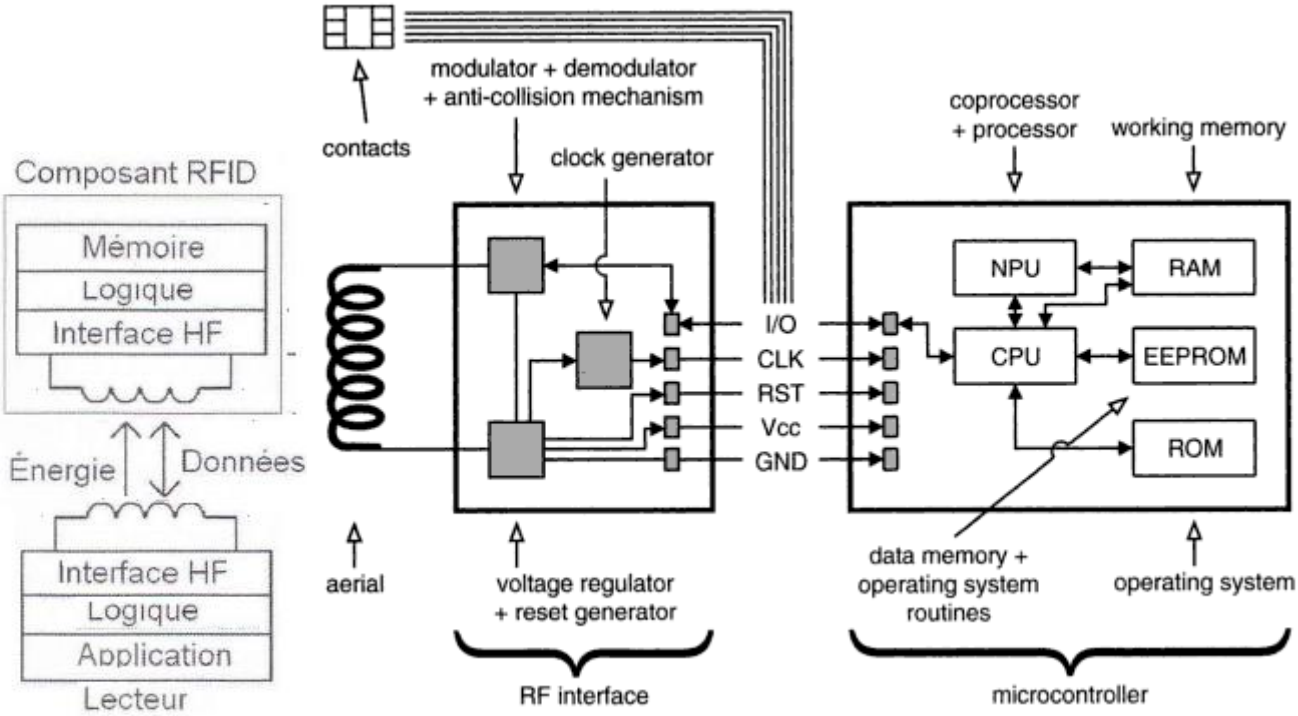


Figure 1.14 Principe général d'un composant de type RFID.

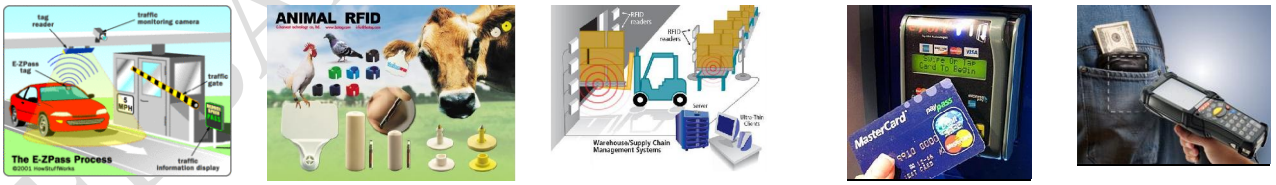


Figure 1.15 : Quelques applications des cartes à puce sans contacte

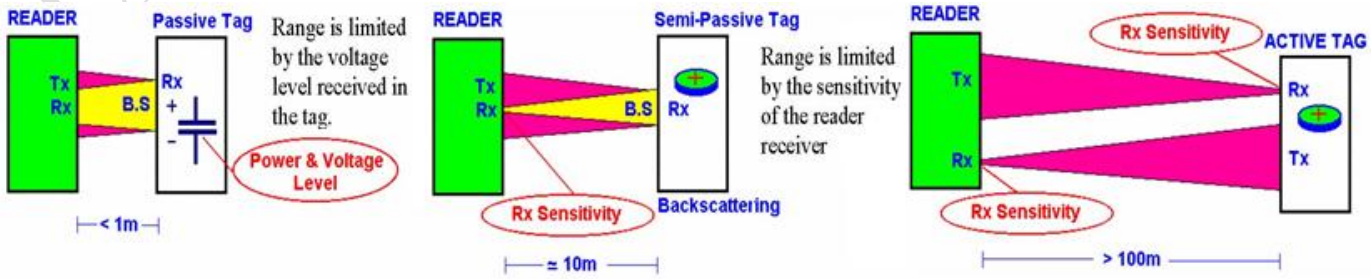


Figure 1.16 : Les distances maximales utilisées pour des cartes à puce sans contacte

2 Semi-conducteurs pour cartes à puces

Le cœur d'une carte à puces est constitué d'un composant électronique monolithique (un bloc rigide et homogène) en silicium introduit dans l'épaisseur d'une carte en plastique. Sur la figure suivante un aperçu rapide des technologies et les différentes étapes qui permettent de fabriquer les cartes à puce.

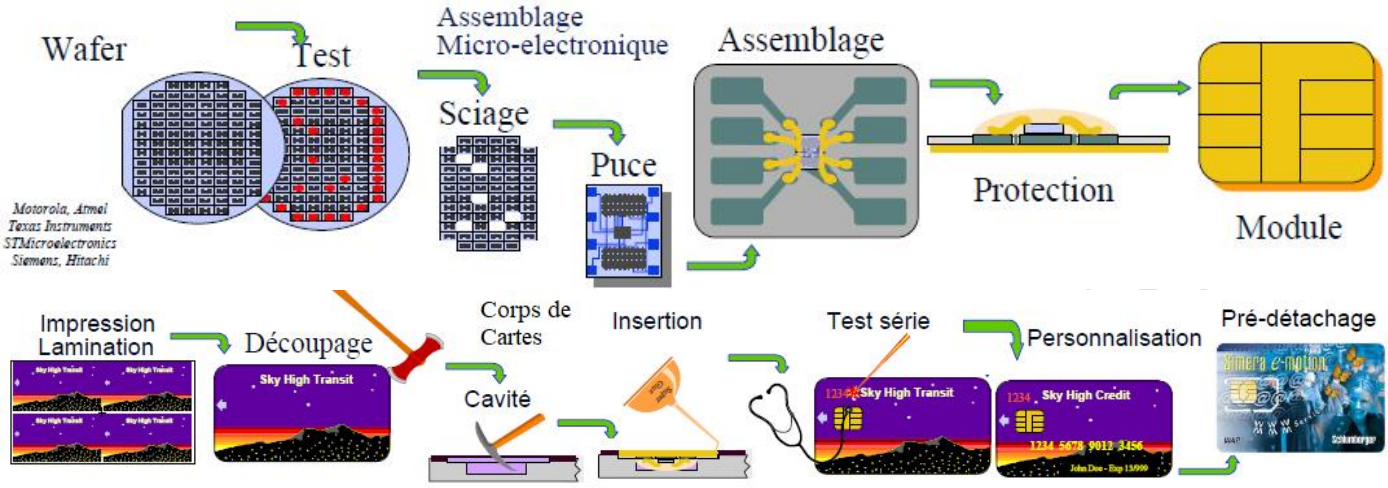


Figure 2.1 : Fabrication des cartes à puce

2.1 Technologies

Au début des cartes à puces, deux filières technologiques étaient en présence selon le type de transistor utilisé pour réaliser les circuits logiques. Technologie *bipolaire* (commande en courant) ou Technologie *unipolaires* MOS (*Metal Oxide Semiconductor*), (commande en tension). Cette dernière est utilisé en raison de la très faible consommation de puissance par rapport au bipolaire, et de la très grande capacité d'intégration. Ces deux dernières décennies, cette technologie a évoluée vers CMOS (*Complementary MOS*), qui se traduit par une très faible consommation et une bonne immunité au bruit.

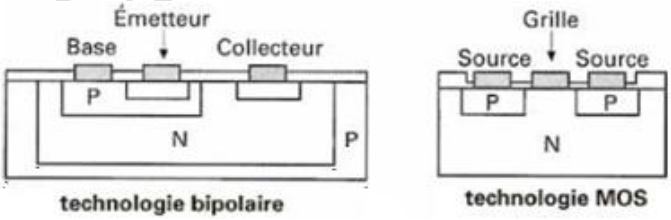


Figure 2.2 : Les filières technologiques utilisées dans les cartes à puce

2.1.1 Technologie des mémoires

Dans les cartes à puces, il faut pouvoir mémoriser des informations confidentielles, y compris en l'absence d'alimentation des circuits. Cette caractéristique constitue l'un des éléments fondamentaux des composants pour cartes à puces : en plus d'un microprocesseur spécialisé, les cartes doivent embarquer différents types de mémoires spécialisées. Celles-ci sont utilisées selon la nature de la mémorisation considérée, la vitesse de fonctionnement et la volatilité des informations stockées.

Les mémoires volatiles à accès aléatoire : appelées RAM (*Random Access Memory*), sont utilisées en tant que registres de travail temporaire et perdent leurs informations dès qu'elles ne sont plus alimentées.

Les RAM dynamiques (DRAM) : il faut renouveler leur contenu périodiquement (rafraichissement)

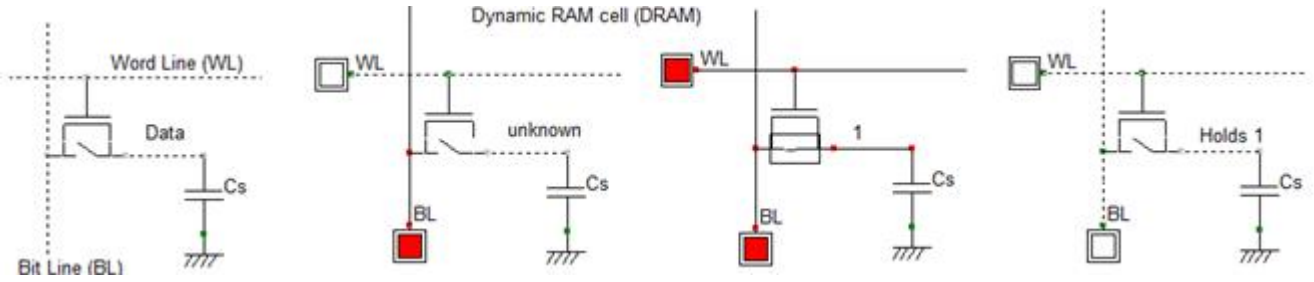


Figure 2.3 : Une cellule de mémoire DRAM

Les RAM statiques (SRAM), permettant de stocker un élément binaire sans rafraichissement.

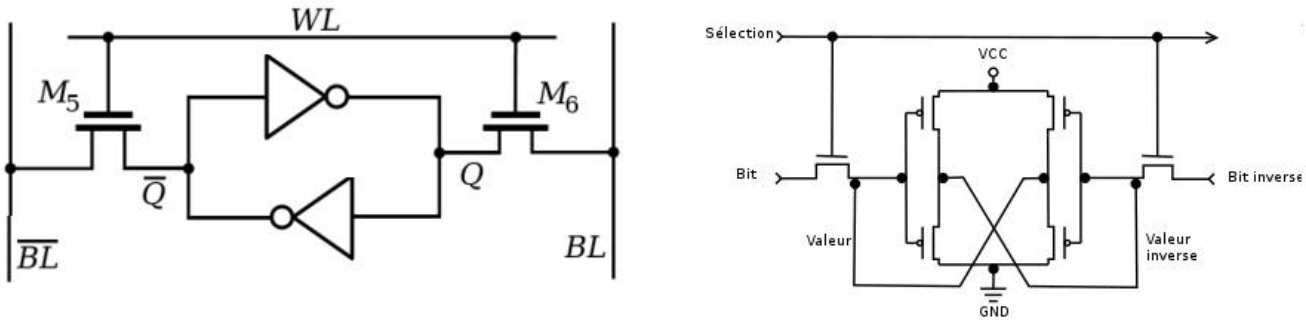


Figure 2.4 : Une cellule de mémoire SRAM

- **<u>Les mémoires non volatiles (NVM)</u>** : Elles gardent les informations en l'absence d'alimentation électrique. Il existe deux types principaux de NVM disponibles sur les composants pour carte à puce :

Les mémoires mortes (ROM (Read Only Memory)), qui sont inaltérables, contiennent des informations permanentes telles que les programmes. Elles ne sont accessibles qu'en lecture. L'inscription de la ROM est réalisée par masquage ou par implantation ionique dans le silicium pendant la fabrication du circuit intégré; les tailles des mémoires ROM embarquées dans le composant pour carte à puces varient en général entre quelques dizaines et quelques centaines de kilo-octets (kB).

Les mémoires mortes programmables (PROM) peuvent être programmées (ou écrites) par l'utilisateur. À l'origine effacer ce type de mémoire était réalisé à l'aide d'un rayonnement ionisant (EPROM, *Erasable Programmable ROM*) ne permettant qu'un petit nombre de reprogrammations des cellules mémoire.

Principe : Générer une haute tension pour sauter les fusibles (12 Volts)
Écrire un 0 $\Rightarrow$ faire sauter un fusible,
Écrire un 1 $\Rightarrow$ fusible intact

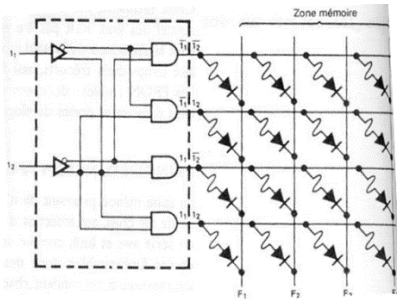


Figure 2.5 : Exemple d'une mémoire à fusible PROM

Les mémoires mortes programmables EEPROM, (*Electrically Erasable Programmable ROM*) Ce type de NVM constitue la quasi-totalité des composants utilisant des mémoires à effacement et reprogrammation par application d'une tension électrique V_{pp}. L'effacement est plus rapide et n'impose pas de retirer le circuit du système dans lequel il est installé.

Les mémoires SRAM occupe environ 20 fois plus de place qu'une cellule ROM (*Read Only Memory*) ce qui augmente leur prix. L'optimisation des coûts explique donc les tailles RAM relativement faibles que l'on trouve dans ces composants : les tailles typiques de mémoires RAM embarquées varient entre quelques kilooctets (kB) et quelques dizaines de kilooctets.

La différence entre les EEPROM et les mémoires vives réside essentiellement dans la vitesse d'écriture. Le cycle d'écriture d'une EEPROM est environ 1000 fois plus long que celui d'une RAM. Les mémoires EEPROM sont par ailleurs trois fois plus « encombrantes » que les mémoires SRAM. Les capacités des mémoires EEPROM embarquées dans les cartes à puces se situent, en 2006, de 2kB à 400kB, avec, pour des raisons essentiellement technologiques, une limite supérieure autour de 512 kB à 1MB.

Les limitations des mémoires EEPROM laissent envisager à court terme leur remplacement par des *mémoires Flash*. Celles-ci présentent l'avantage d'un faible encombrement (la densité est comparable à celle de la mémoire ROM) et des performances en écriture sensiblement meilleures que celles des mémoires EEPROM. L'inconvénient des mémoires flash est le problème d'une programmation complexe. L'introduction des mémoires Flash dans les cartes à puces est une réalité depuis 2005, avec une taille de l'ordre de 1 mégaoctet (1 MB).

À plus long terme, les industriels parient sur des technologies de type *MRAM* (*« Magnetoresistive RAM »*) ou *PC-RAM* (*Phase Change RAM*).

2.2 Puce en logique câblée

Ces composants, généralement très simples, sont des NVM à accès sériel synchrone, contrôlés par des circuits logiques interconnectés entre la mémoire et l'interface externe. Les commandes disponibles sur ces composants sont très limitées et figées par construction. Les cartes à logique câblée sont principalement utilisées comme jetons électroniques ou comme identifiants. L'exemple le plus répandu est la télécarte utilisée dans les téléphones. Dans ce type de carte, il n'y a malheureusement aucun véritable standard et les composants offerts sont très souvent incompatibles entre eux. La famille des jetons électroniques s'est développée en deux générations.

- **La première génération (1G)** : qui correspond au lancement de la télécarte en France et en Allemagne, se compose essentiellement de mémoires sérialisées possédant une zone protégée par un fusible.

Dans la première télécarte, encore utilisée aujourd'hui, la mémoire EPROM possède 256 bits, avec une zone protégée de 96 bits qui stocke une référence. Par contre, la carte allemande, développée plus tard, contenait 416 bits de mémoire EEPROM recyclable 64 fois, incluait 208 bits de mémoire de travail et un contrôle d'accès par un code. Ces deux réalisations ont engendré deux lignées d'interfaces électriques incompatibles entre elles : d'une part une interface nécessitant 8 contacts externes, utilisée par France Télécom, et d'autre part deux interfaces à 8 et 6 contacts utilisées par la Bundespost (DBP). La figure suivante illustre les trois types d'interfaces utilisées par France Télécom, Bundespost et recommandées par la norme ISO, ainsi que leurs commandes associées.

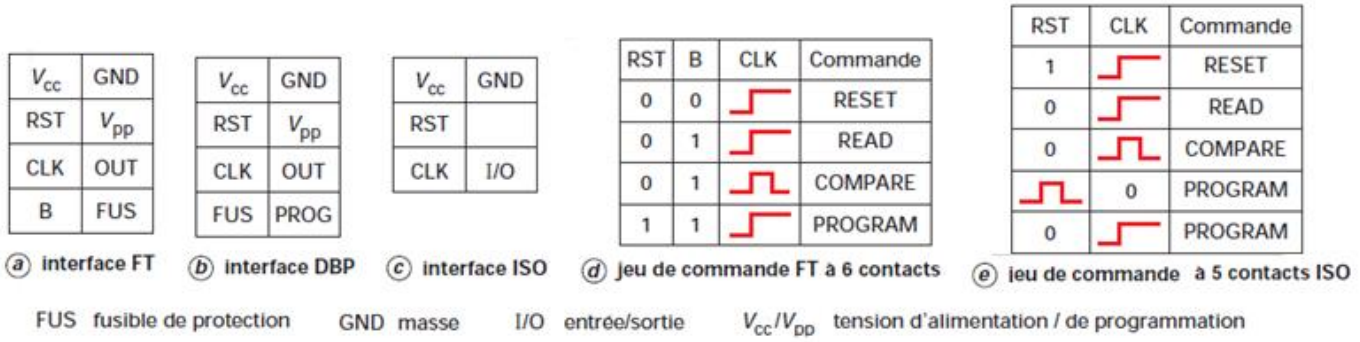


Figure 2.6 : Les trois types d'interfaces utilisées par France Télécom

La méthode d'accès à ces mémoires consiste à adresser bit à bit chaque cellule comme dans un registre à décalage. La remise à zéro du circuit permet de se positionner sur la première adresse mémoire et la lecture se fait sur le plot **OUT**, après la remontée du signal d'horloge.

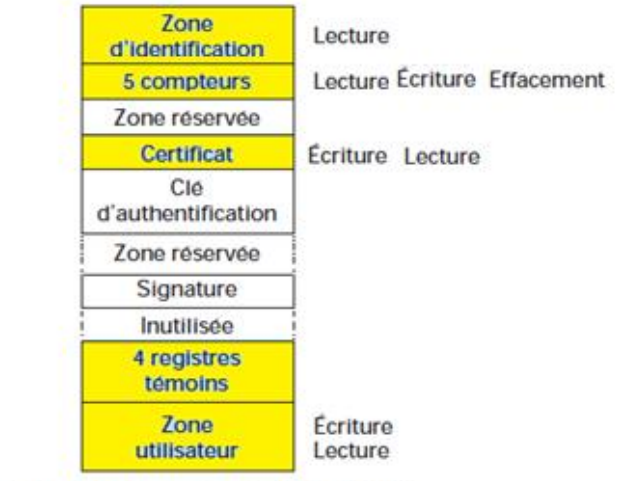
Lorsque le signal **RST** de remise à zéro n'est plus actif, une impulsion d'horloge sur **CLK** incrémente le compteur d'adresse du composant. Les commandes de lecture, d'écriture ou d'effacement correspondent aux combinaisons de signaux des figures 2.6 *d* et *e* sont prises en compte sur un front montant de l'horloge.

Dans le cas de l'interface de type **FT**, il faut utiliser une combinaison de deux signaux, **RST** et **B**, tandis que dans la première télécarte allemande **(DBP)**, il s'agit des signaux **RST** et **PROG**. Dans le cas de l'interface à **5** contacts, inspiré par le standard ISO, toutes les commandes sont assurées par le plot **RST**. Dans ce cas, une impulsion sur **RST** en maintenant **CLK** à zéro permet de passer en mode programmation, tout en restant sur l'adresse sélectionnée, et l'impulsion suivante sur **CLK** permet de programmer le bit correspondant.

- **La seconde génération (2G)** : Dans cette génération apparaissent deux caractéristiques nouvelles. D'une part, une fonction d'authentification dynamique, qui délivre un résultat sur quatre bits lorsque l'on fournit à la carte une donnée aléatoire de 64 bits et, d'autre part, une mémoire de travail constituée par des compteurs. La figure 2.7 donne l'organisation de la mémoire des composants ST 1335 et 1336 de STMicroelectronics, qui rassemblent pratiquement toutes les caractéristiques rencontrées dans les divers autres composants à logique câblée.

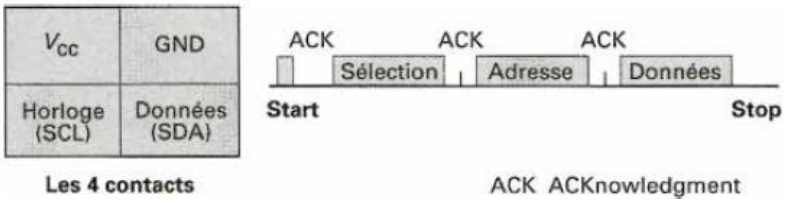
Une première zone protégée de 64 bits contient un code d'identification inscrit en usine et des informations sur l'application introduite par l'émetteur. Cinq registres de 8 bits constituent les compteurs et sont associés à une autre zone de quatre registres d'indicateurs témoins qui permettent les reprises, en cas de rupture de séquence à la suite d'une extraction intempestive.

La carte peut aussi contenir une signature électronique attestant les droits de la carte et une clé cryptographique pour l'authentification.



La figure 2.7 : L'organisation de la mémoire

La figure 2.8 montre, d'une part, l'interface électrique qui utilise seulement une horloge SCL de synchronisation et une ligne de données SDA et, d'autre part, un exemple de séquence d'écriture. Le premier octet sélectionne la mémoire, le second donne l'adresse et le troisième la donnée. Sur le bus I2C la réception d'un octet est acquittée par le forçage de la ligne à zéro pendant le neuvième bit.



La figure 2.8 : Interface I^2C (commande d'écriture)

Le tableau suivant donne une idée de la grande variété de composants à logique câblée (sans toutefois prétendre à l'exhaustivité). Lorsque les applications doivent supporter plusieurs types de cartes, la complexité est reportée du côté du terminal qui doit s'adapter électriquement et logiquement aux cartes supportées.

Tableau 2.1 Composants pour cartes à logique câblée

Fournisseur/type	**Capacité mémoire**	**Autres spécifications**
Infineon SLE 4404	ROM : 16 bit PROM : 144 bits EEPROM : 256 bits	*V*cc 5V Cycles E/L : 100 000
Infineon SLE 44 R35	EEPROM : 1 024 bits	*V*cc 5V : Cycles E/L : 100 000 Protection écriture par PIN
Infineon SLE 7736	ROM : 24 bits PROM : 177 bits EEPROM : 36 bits	*V*cc 3V-5V : Cycles E/L : 100 000 Compteur jusqu'à 20 000 unités
Philips MF1 S70	EEPROM : 40 966 bits	Cycles E/L : 100 000 Interface sans contact ISO 14443 A Authentification
STMicroelectronics M35101	EEPROM : 2 048 bits	Cycles E/L : 100 000 Interface sans contact ISO 14443B
STMicroelectronics 1335D	ROM : 16 bits EEPROM : 272 bits	*V*cc 5V Cycles E/L : 500 000 Compteur jusqu'à 32 567 unités Authentification

2.3 Puces à Microcalculateur

Les microcalculateurs pour cartes à puces sont de véritables ordinateurs intégrés sur un seul substrat de silicium : ce sont des machines à programme enregistré qui sont architecturées autour d'un (parfois deux) bus de données. Ils contiennent les différents types de mémoire, et les organes d'entrée-sortie, qui assurent les dialogues avec le monde extérieur. Le programme de fonctionnement est généralement contenu dans la ROM, tandis que la RAM contient les registres de travail nécessaires aux divers traitements internes.

Les mémoires non volatiles intégrées sur le composant permettent de reprogrammer la mémoire, à des fins de mise au point ou d'évolution, voire pour les architectures les plus modernes de rajouter des programmes après la délivrance de la carte (par exemple, dans le cas des architectures Javacard). Les microcalculateurs pour cartes à puces intègrent encore de nos jours le concept de SPOM (*Self Programmable One Chip Microcomputer*), architecture essentiellement sécuritaire, qui permet au microcalculateur de modifier lui-

même un programme contenu dans sa propre NVM sans intervention du monde extérieur (Figure 22). À cette caractéristique viennent s'ajouter divers dispositifs de sécurité qui empêchent les intrusions dans le composant. La figure suivante montre un exemple récent de composant : le SPOM ST 22FJ1M de STMicroelectronics réalisé en technologie CMOS 0,18 µm, qui intègre des mémoires Flash, en remplacement des traditionnelles mémoires EEPROM.

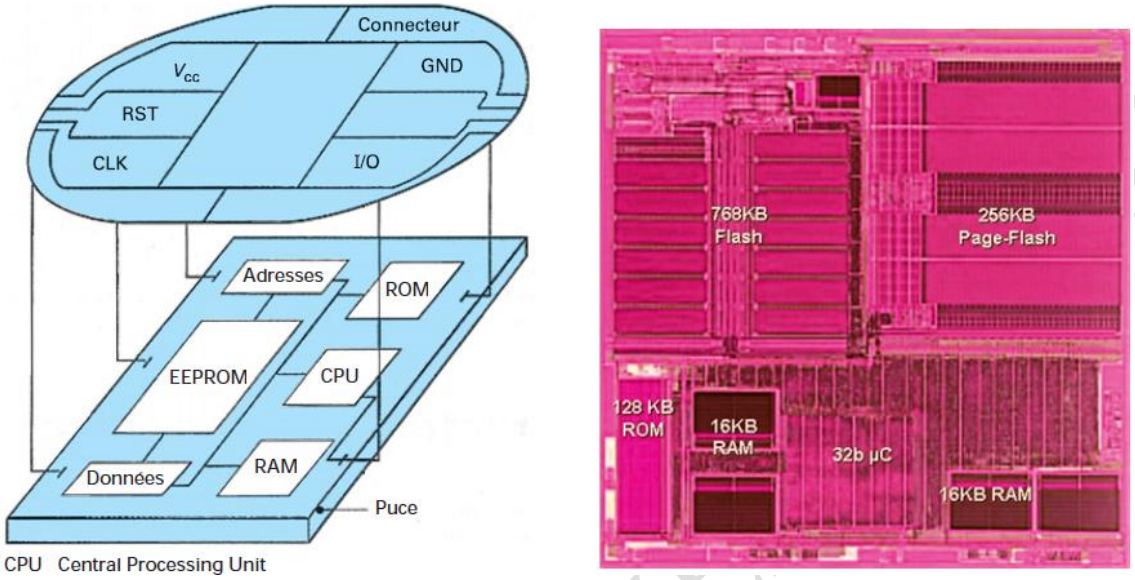


Figure 2.9 : Microcalculateur (SPOM) et ses interconnexions – SPOM ST 22FJ1M de ST Microelectronics

La capacité de traitement des microprocesseurs permet de rendre l'interface externe indépendante des applications, et donc normalisable. L'interface électrique, les protocoles d'échanges et le jeu de commandes de ces cartes font l'objet de la norme internationale ISO 7816.

3 Cryptologie et sécurité

La cryptologie peut se définir comme étant la science de la dissimulation de l'information. Elle constitue, avec la sécurité physique des composants et des systèmes d'exploitation, la dimension essentielle de la sécurité des cartes à puces. Elle englobe en premier temps la cryptographie, qui est l'art de chiffrer et de déchiffrer les messages, en deuxième temps la cryptanalyse, qui est l'art de casser les codes secrets.

3.1 Principes de la cryptographie

Le chiffrement des messages consiste à transformer une information à l'aide d'une convention secrète. La fonction de transformation constitue l'algorithme de cryptographique, dont le secret réside dans des paramètres appelés clés. Lorsque l'on déchiffre le message, on réalise l'opération inverse en connaissant ces clés. Dans les cartes à puces, la cryptographie met en œuvre divers mécanismes qui ont pour but d'assurer soit la confidentialité des informations, soit l'authentification des cartes ou des utilisateurs, soit encore la signature des messages. L'ensemble des moyens mettant en œuvre la cryptographie forme un cryptosystème.

Le schéma simplifié d'un cryptosystème est représenté figure 3.1. À l'émission, le message est transformé par l'algorithme A qui est fonction de la clé de chiffrement K_e . À la réception, les informations reçues sont déchiffrées à l'aide de l'algorithme inverse A^{-1} utilisant une clé K_r . Notons que le déchiffrement

implique un algorithme réversible, ce qui n'est pas le cas de l'authentification. En effet, certains mécanismes d'authentification utilisent un algorithme dans le même sens aux deux extrémités avec la même clé.

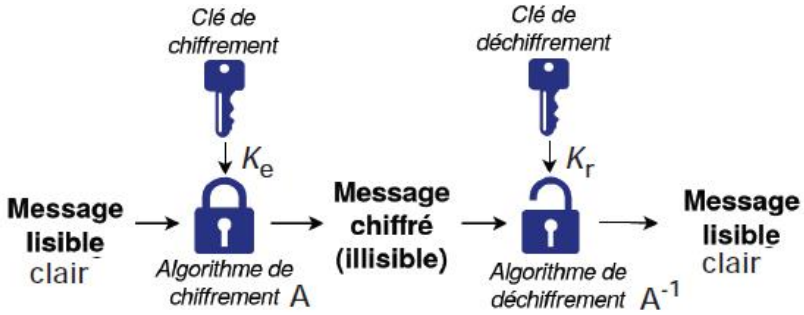


Figure 3.1 : Schéma simplifié d'un cryptosystème

Exemple : Cryptage du mot « bonjour» avec la Clé : Remplacer chaque lettre par la lettre qui la précède bonjour→ cpokpvs

II en existe trois catégories selon qu'ils sont symétriques, asymétriques ou « à apport nul de connaissance».

3.1.1 Cryptosystèmes symétriques

Le cryptosystème est dit symétrique lorsque $K_e = K_r$. Les algorithmes utilisés dans ce type de cryptosystèmes sont très nombreux. Ils peuvent être privés ou publics, mais le choix d'un algorithme pour carte à puces dépend essentiellement du service de sécurité attendu, de la performance, et surtout du coût des ressources nécessaires à son implantation (tailles de RAM et de ROM). En effet, l'utilisation d'algorithmes encombrants augmente très vite le prix des cartes. II est également important de tenir compte de la réglementation concernant l'usage de la cryptographie qui est variable d'un pays à l'autre.

Le plus connu des algorithmes symétriques est développé dans les années 1970 c'est le DES (Data Encryption Standard) normalisé par le NIST (National lnstitute of Standard and Technology) aux États-Unis. Il n'a pas été conçu pour une implémentation programmée et ne correspond probablement pas au meilleur compromis souhaitable entre performances et ressources utilisées. Cependant sa simplicité d'implémentation ainsi que sa sécurité intrinsèque l'ont rendu pratiquement incontournable dans les cartes à microcalculateurs. Utilisant une longueur de clés de 56 bits, et prenant en entrée des blocs de 64 bits. Le DES commence à devenir obsolète et tend à être remplacé par un nouvel algorithme, connu sous le nom d'AES ou « Advanced Encryption Standard » lancé en janvier 1997 par le NIST.

L'algorithme AES prend en entrée un bloc de 128 bits (16 octets), pour des longueurs de clés de 128, 192 ou 256 bits. Les algorithmes de chiffrement constituent en fait une « boîte noire » à trois canaux : deux d'entrée (la clé et l'information d'origine) et un de sortie, dont la largeur est généralement un multiple de 64 $bits$. On peut utiliser deux principaux modes de fonctionnement :

- **Le mode ECB (Electronic Code Book)**: Il s'agit du mode le plus simple. Le message à chiffrer est divisé en plusieurs blocs qui sont chiffrés indépendamment les uns après les autres avec la même clé secrète K (Figure 3.2).. Le gros défaut de cette méthode est que deux blocs avec le même contenu seront chiffrés de la même manière, on peut donc tirer des informations à partir du texte chiffré en cherchant les séquences

identiques. On obtient dès lors un « dictionnaire de codes » avec les correspondances entre le clair et le chiffré d'où le terme codebook.

Ce mode est pour ces raisons fortement déconseillé dans toute application cryptographique. Le seul avantage qu'il peut procurer est un accès rapide à une zone quelconque du texte chiffré et la possibilité de déchiffrer une partie seulement des données. Mais un mode bien plus sûr basé sur un compteur permet également ces accès aléatoires et les déchiffrements partiels.

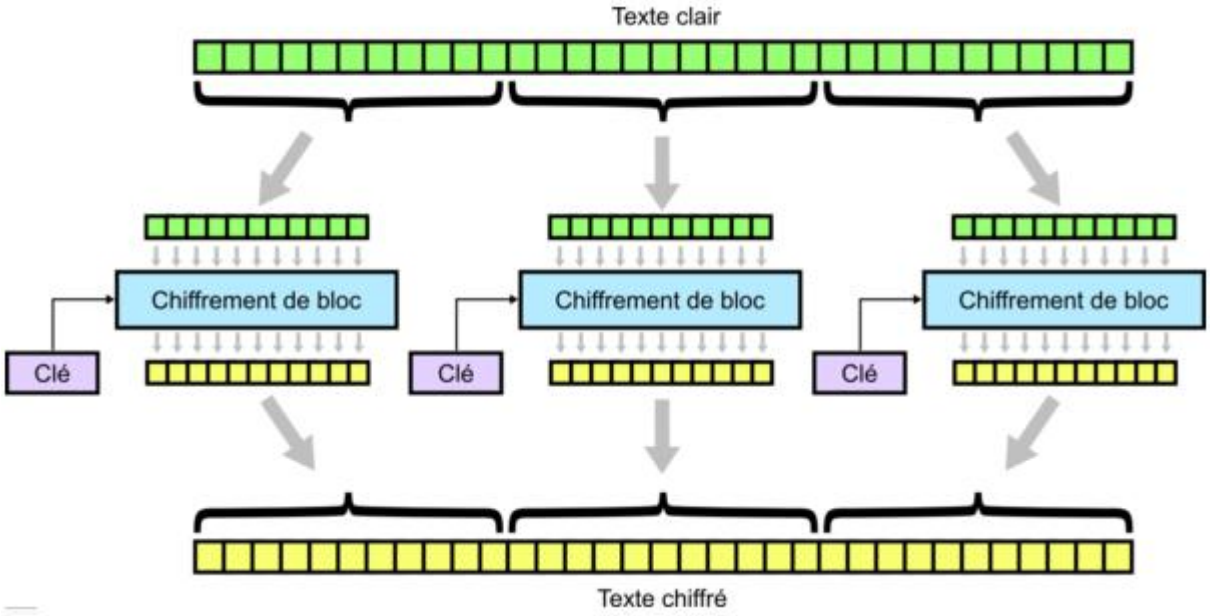


Figure 3.2 : Schéma simplifié d'un cryptosystème ECB

Pour l'AES on transforme simplement les $n \times 64$ bits d'entrée en un message de même longueur ;

- **Le mode CBC (Cipher Bloc Chaining)** : C'est un algorithme qui utilise un chiffrement par bloc pour assurer la sécurité de l'information. La taille de bloc est comprise entre 32 et 512 bits, le standard pour la AES est de 128 bits. Les blocs sont ensuite chiffrés les uns après les autres (Figure 3.3).

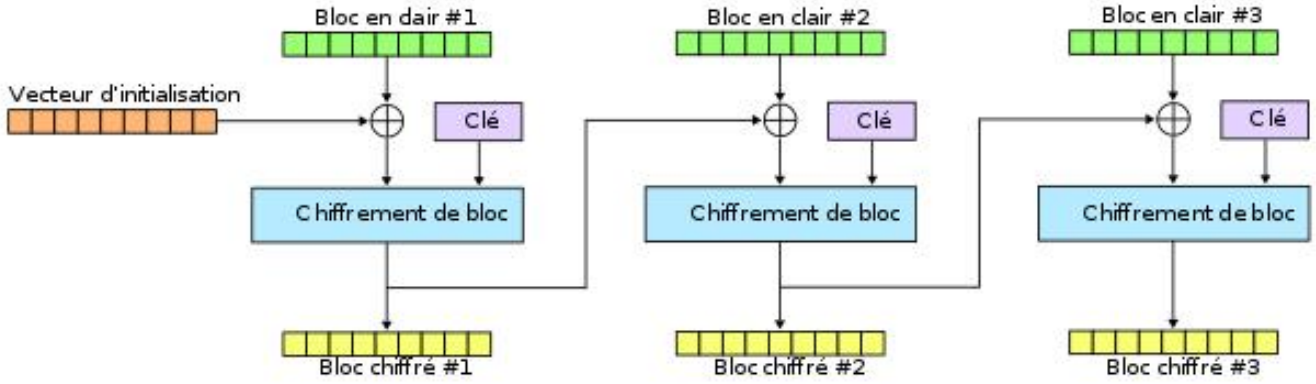


Figure 3.3 : Schéma simplifié d'un cryptosystème CBC

Ce mode consiste à chiffrer le $n^{ème}$ bloc (cryptogramme C_n d'un message M_n) préalablement combiné par un OU-Exclusif (XOR) avec le chiffré du bloc précédent C_{n-1} et d'un vecteur d'initialisation E (algorithme utilisé). Il s'agit d'un bloc de données aléatoires qui permet de commencer le chiffrement du premier bloc et

qui fournit ainsi une forme de hasard indépendant du document à chiffrer. Il n'a pas besoin d'être lui-même chiffré lors de la transmission, mais il ne doit jamais être réemployé avec la même clé K.

En d'autre terme, il combine les blocs de données en les chaînant : le $n^{ème}$ cryptogramme, C_n , d'un message M est en fait calculé à partir du $n^{ème}$ bloc de message, M_n , et du $(n-1)^{ème}$ cryptogramme, C_{n-1} , par la relation : $C_n = E\ K\ (C_{n-1} \oplus M_n)$ avec K_n clé d'encryptions, E algorithme utilisé, $\oplus$ représente le ou exclusif.

Les défauts de CBC étant qu'il ne peut pas être parallélisé étant donné que le bloc courant nécessite que le précédent soit chiffré. Il est donc séquentiel. Selon l'implémentation qui est faite, le mode CBC peut être vulnérable à la méthode "Padding Oracle" qui permet de retrouver les blocs en clair.

Un exemple d'authentification dynamique est donné figure 3.4. Le vérifieur choisit un nombre E au hasard, le chiffre avant de l'envoyer à la carte, et demande à cette dernière de déchiffrer le message M. Si la carte est authentique, elle est la seule à pouvoir retrouver le nombre E en utilisant sa clé secrète (ici l'algorithme est symétrique donc $K_1 = K_2$).

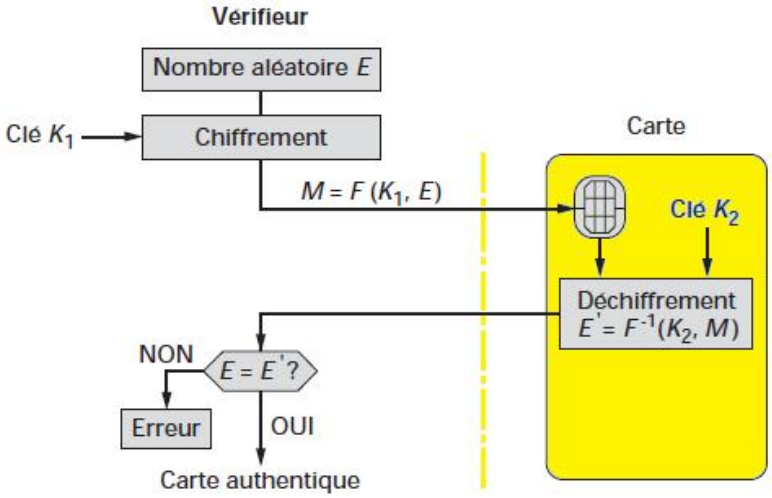


Figure 3.4 : Schéma simplifié d'un mécanisme d'authentification

Il faut attirer l'attention sur les problèmes de gestion de clés qu'ils posent. En effet, lorsqu'un grand nombre d'utilisateurs font partie d'un réseau, il faut que chacun d'entre eux possède une clé personnalisée, car une clé unique constituerait une menace pour tout le système dans le cas où elle serait compromise.

Comme il est peu pratique et risqué de stocker toutes les clés, la méthode consiste à les diversifier à partir d'une clé maître et d'un identifiant de chaque carte. Les clés maîtresses doivent être particulièrement protégées, et elles sont parfois contenues dans un module de sécurité ou une carte « mère » possédée par l'émetteur des cartes.

3.1.2 Crypto systèmes asymétriques

Lorsque $K_e \neq K_r$, le cryptosystème est dit asymétrique. L'idée de base date de 1976, suppose qu'il existe des algorithmes cryptographiques tels qu'il soit impossible, au sens de la complexité des calculs, de retrouver une des deux clés en connaissant l'autre.

Ce n'est qu'en 1978 qu'un tel algorithme a vu le jour : il s'agit du RSA, du nom de ses inventeurs Rivest, Shamir et Adleman. C'est d'ailleurs le seul système asymétrique connu qui ait résisté jusqu'à présent aux cryptanalyses, à condition de prendre des clés suffisamment longues. Tous les calculs de l'RSA se font modulo un nombre entier n qui est le produit de deux nombres premiers p et q. Le RSA est fondé sur la difficulté de factoriser des grands nombres. Les messages clairs et chiffrés sont des entiers inférieurs à l'entier n. Les opérations de chiffrement et de déchiffrement consistent à élever le message à une certaine puissance e et d modulo n (c'est l'opération d'exponentiation modulaire).

Le chiffrement RSA asymétrique : il utilise une paire de clés (des nombres entiers) composée d'une *clé publique* pour chiffrer et d'une *clé privée* pour déchiffrer des données confidentielles. Les deux clés sont créées par une personne, souvent nommée par convention *Alice*, qui souhaite que lui soient envoyées des données confidentielles. Alice rend la clé publique accessible. Cette clé est utilisée par ses correspondants (*Bob*, etc.) pour chiffrer les données qui lui sont envoyées. La clé privée est quant à elle réservée à Alice, et lui permet de déchiffrer ces données. La clé privée peut aussi être utilisée par Alice pour signer une donnée qu'elle envoie, la clé publique permettant à n'importe lequel de ses correspondants de vérifier la signature.

Le couple (n, e) ou (e, n) est la *clé publique* du chiffrement, alors que sa *clé privée* est le nombre d, sachant que l'opération de déchiffrement ne demande que la clef privée d et l'entier n, connu par la clé publique (la clé privée est parfois aussi définie comme le couple (d, n) ou le triplet (p, q, d)).

n étant un nombre composite, dont les facteurs premiers sont p et q, le chiffrement RSA du message M est donné par : $C = M^e mod\ n$. On montre que la fonction de déchiffrement est : $M = C^d mod\ n$ et que la relation entre les deux exposants e et d est la suivante : $ed = 1\ mod\ (p-1)\ (q-1)$

Exemple : avec de petits nombres premiers (en pratique il faut de très grands nombres premiers):

1. On choisit deux nombres premiers $p = 3, q = 11$;
2. Leur produit $n = 3 \times 11 = 33$ est le module de chiffrement ;
3. $\varphi(n) = (p-1)(q-1) = (3-1) \times (11-1) = 2 \times 10 = 20$
4. On choisit $e = 3$ (premier avec 20) comme exposant de chiffrement ;
5. L'exposant de déchiffrement est $d = 7$, l'inverse de 3 modulo 20 (en effet $ed = 3 \times 7 \equiv 1\ mod\ 20$).

La clé publique d'Alice est $(n, e) = (33,3)$, et sa clé privée est $(n, d) = (33, 7)$. Bob transmet un message à Alice.

- Chiffrement de $M = 4$ par Bob avec la *clé publique* d'Alice : $4^3 \equiv 31\ mod\ 33$, le chiffré est $C = 31$ que Bob transmet à Alice ;
- Déchiffrement de $C = 31$ par Alice avec sa *clé privée* : $31^7 \equiv 4\ mod\ 33$, Alice retrouve le message initial $M = 4$.

Le mécanisme de signature par Alice, à l'aide de sa clé privée, est analogue, en échangeant les clés.

Si *n* est suffisamment grand, on peut donc le publier ainsi que l'un des deux exposants e ou d, et garder p et q secrets. Dans ces conditions, quelqu'un ne connaissant pas p et q ne peut calculer e en fonction de d,

et inversement. Les propriétés asymétriques de ces algorithmes sont très intéressantes pour chiffrer, authentifier ou signer des messages. Ainsi :

- Pour assurer la confidentialité d'une information, toute personne connaissant n et e peut chiffrer un message. Seul le destinataire détenant la clé secrète pourra le déchiffrer ;
- N'importe qui connaissant la clé publique peut authentifier une carte grâce au schéma de la figure 26: le vérifieur choisit un nombre E au hasard, le chiffre à l'aide de la clé publique K_1 liée à la carte et demande à cette dernière de déchiffrer le message. Si la carte est authentique, elle est seule à pouvoir retrouver le nombre E en utilisant sa clé secrète K_2 . Si l'on veut utiliser cette méthode avec un temps de calcul raisonnable, il faut donc que le microcalculateur de la carte puisse calculer rapidement les multiplications modulaires.

Il est aujourd'hui admis que, pour la prochaine décennie, une application de sécurité moyenne basée sur le RSA ne devrait pas descendre au-dessous de 1024 bits, et que dès qu'il s'agit de haute sécurité, il faut passer à 1720 bits, voire 2048. Si le RSA constitue aujourd'hui le système à clés publiques le plus employé dans les cartes à microcalculateurs, d'autres systèmes commencent à devenir également populaires, tels les cryptosystèmes basés sur des courbes elliptiques sur des corps finis, qui présentent l'avantage d'une faible consommation mémoire sur la carte.

3.1.3 Crypto systèmes à apport nul de connaissance,ou preuves à divulgation nulle de connaissance

Les protocoles «à apport nul de connaissance » (*Zero Knowledge Proof*) (ZKP) reposent sur une idée introduite par Goldwasser, Micali et Rackoff en 1985. Elle a été concrétisée par A. Fiat et A. Shamir en 1986, et optimisée par L. Guillou et JJ. Quisquater en 1987 pour les cartes à microcalculateur. Dans les systèmes précédents, il est nécessaire qu'un secret de base soit utilisé par le vérifieur (cas des systèmes symétriques) ou par le prouveur (cas des systèmes asymétriques). Dans les systèmes « à apport nul de connaissance » est une méthode très puissante pour authentifier et signer des messages, sans donner la moindre information sur le secret utilisé, car une part est laissée au hasard. On demande au prouveur de résoudre un problème qu'il est seul capable de résoudre, le nombre de réponses possibles étant faible. Le protocole consiste à poser une série de questions pour augmenter la conviction du vérifieur avec le nombre de bonnes réponses données par le prouveur (Exemple sur la Figure 3.5).

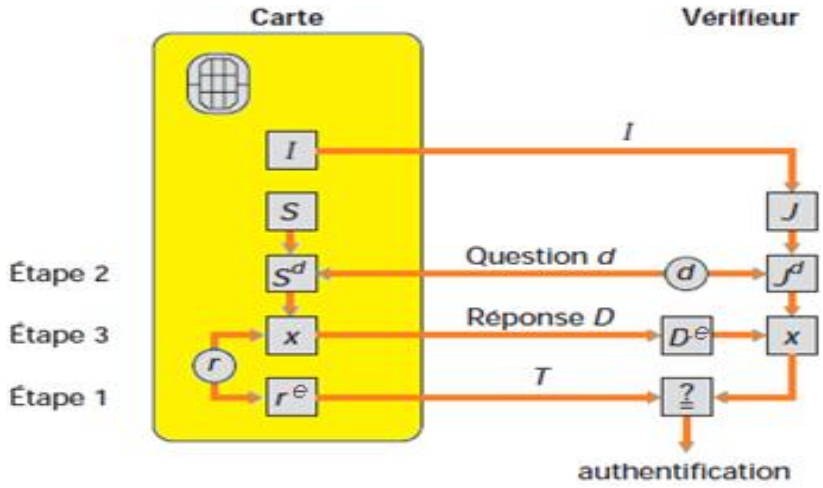


Figure 3.5 : Protocole Guillou-Quisquater authentification

La figure 27 illustre le protocole Guillou-Quisquater ou GQ qui réduit au minimum les échanges avec la carte.

Le protocole de Guillou-Quisquater est une preuve à divulgation nulle de connaissance proposée par Louis Guillou et Jean-Jacques Quisquater pour prouver la connaissance d'un message clair RSA pour un chiffré public. Cette preuve peut-être dérivée en protocole d'authentification par l'heuristique de Fiat-Shamir.

L'émetteur de la carte a émis un nombre composite n, dont la factorisation pq est gardée secrète, et un exposant de vérification e. Chaque carte contient un secret S tel que : $S^e J = 1 \ mod \ n$

J étant une identité redondante de la carte obtenue à partir d'informations I caractéristiques de cette carte.

L'opération de vérification se déroule de la façon suivante :

- 1 étape (L'engagement) : La carte tire au hasard un nombre r et délivre (envoie au vérifieur) : $T = r^e \ mod \ n$ et J
- 2 étape (Le défi) : on pose la question à la carte sous la forme d'un nombre d tiré au hasard par le vérifieur, avec $0 < d < e - 1$;
- 3 étape (La réponse) : la carte répond par la quantité : $D = r \times S^d \ mod \ n$
- 4 étape (La vérification): on vérifie (sur le vérifieur,) que :

$$T = D^e \cdot J^d = (rS^d)^e \cdot J^d = r^e \ mod \ n \text{ et que : } T \neq 0$$

Pour réussir l'authentification, il faut connaître le secret S et la décomposition de $n = pq$, ou encore deviner d par anticipation. Ce dernier cas correspond à une probabilité d'autant plus faible que e est grand. Un tel processus n'apporte aucune information sur le secret, mais permet d'acquérir la conviction de la preuve.

3.2 Sécurité physique et logique des cartes à puces

Les microcalculateurs pour cartes à puces assurent la confidentialité des informations sensibles, telles que les clés cryptographiques ou les mots de passe qui contrôlent l'accès à certaines fonctions. Comme on peut se l'imaginer, un certain nombre de dispositifs physiques sont nécessaires pour empêcher les intrusions dans les composants ou les modifications non désirées du contenu de la mémoire non volatile.

De plus, ces composants étant exposés à toutes les menaces, leur conception doit éviter qu'un comportement non maîtrisé (non standard ou piratage) soit effectué. Afin d'atteindre ce but les concepteurs s'appuient sur l'expérience et plusieurs itérations de tests sur les cartes pour développer des algorithmes de sécurité logique ou fabriquer des puces plus sécurisé physiquement.

Un bon niveau de sécurité physique des composants est en général suffisant si on respecte certaines règles dans les procédures de fabrication et de personnalisation des composants. Pour sécuriser un microprocesseur (et donc tout composant), il faut au minimum les quatre objectifs suivants :

- Proscrire (bannir) tout retour de fonctionnement en mode test ;
- Rendre toute investigation physique du composant impossible dans l'état de l'art;
- Détecter toute tentative d'intrusion ou d'altération dans les mémoires ;
- Interdire tout fonctionnement en dehors du domaine parfaitement contrôlé et spécifié.

3.2.1 Interdire le mode test

Pour parvenir au premier objectif, tous les chemins de test doivent être irréversiblement détruits, voire inexistants. Des détecteurs d'intrusion, des alarmes et des mécanismes de protection doivent être implémentés et exploités par le système d'exploitation. Le composant doit aussi être capable d'exécuter des tâches de contrôle, non seulement dans les conditions normales de fonctionnement, mais aussi, et surtout, dans les conditions anormales.

3.2.2 Résister aux attaques

Pour atteindre le deuxième objectif, le composant doit être exempt de vulnérabilité par rapport à toutes les techniques actives ou passives connues d'observation. Les cartes à puce sont des systèmes embarqués qui reposent sur une couche matérielle plus ou moins complexe qui les expose à de nombreuses d'attaques dites « physiques ». Ces dernières sont nombreuses et variées, leur faisabilité dépend principalement des moyens (équipements matériels, nombre de vecteurs de test disponibles, accès physique), des connaissances (accès aux documentations techniques détaillées, accès aux données, expérience) et du temps dont dispose l'attaquant. Il existe de nombreuses attaques dites physiques qui ciblent les systèmes électroniques embarqués. Nous pouvons les classer en deux groupes distincts : les attaques actives (ou invasive), et les attaques passives.

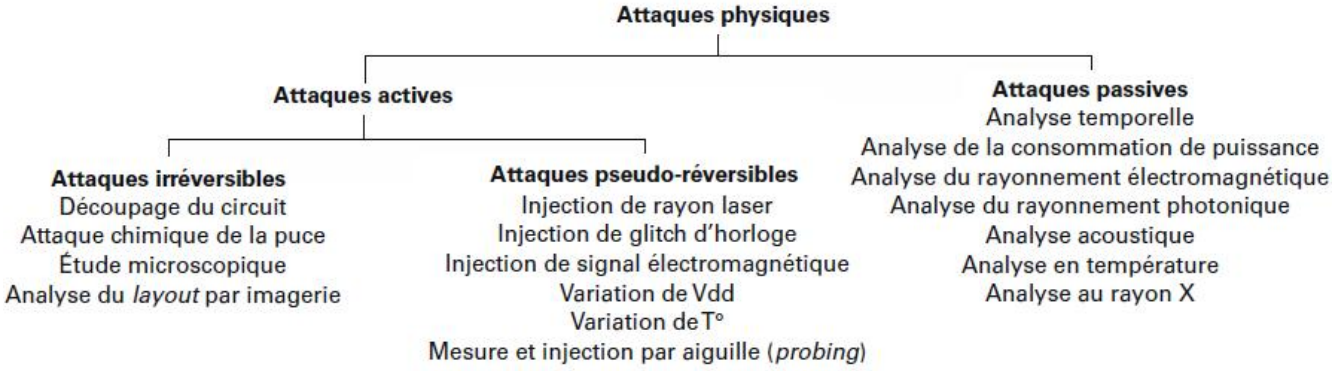


Figure 3.6 : Classification des attaques physiques

- Les attaques actives, comme leur nom l'indique, vont conduire l'attaquant à agir sur le système attaqué, à modifier son fonctionnement ou bien à attenter à son intégrité physique. Il existe deux sous-groupes d'attaques actives : les attaques irréversibles et les attaques pseudo-réversibles. Les attaques irréversibles conduisent à la destruction du système attaqué. Par exemple, l'abrasion chimique et le découpage laser d'un circuit intégré. Ce sont des attaques utilisées lors de la rétro-conception qui permettent alors à un attaquant d'obtenir des informations sur la conception d'un circuit. Les attaques actives pseudo-réversibles n'entraînent pas forcément la destruction du système attaqué, mais elles sont invasives. Elles nécessitent la préparation du circuit (exemple : découpe partielle du boîtier du circuit intégré) et elles entraînent parfois la dégradation du système attaqué. On trouve classiquement sous cette appellation les attaques par probing (utilisation d'aiguilles très fines pour la mesure ou l'injection de signal en un point très précis de la puce) et les attaques par injection de fautes (Padding oracle). Ces dernières consistent, à créer des fautes dans un circuit électronique via une perturbation externe. Pour cela, différents canaux peuvent être utilisés : le canal optique

principalement par l'utilisation de laser, le canal électromagnétique, le canal thermique, l'horloge d'un circuit synchrone (injection de glitchs (bug) d'horloge) ou encore par la tension d'alimentation. Les fautes ainsi créées peuvent entraîner le circuit dans des modes de fonctionnement conduisant à des erreurs dont l'exploitation peut révéler des données secrètes à l'attaquant. Les attaques actives (irréversibles ou non) nécessitent du temps, des moyens, des connaissances techniques et des informations sur les systèmes attaqués. Ce sont donc des solutions réservées aux experts du domaine.

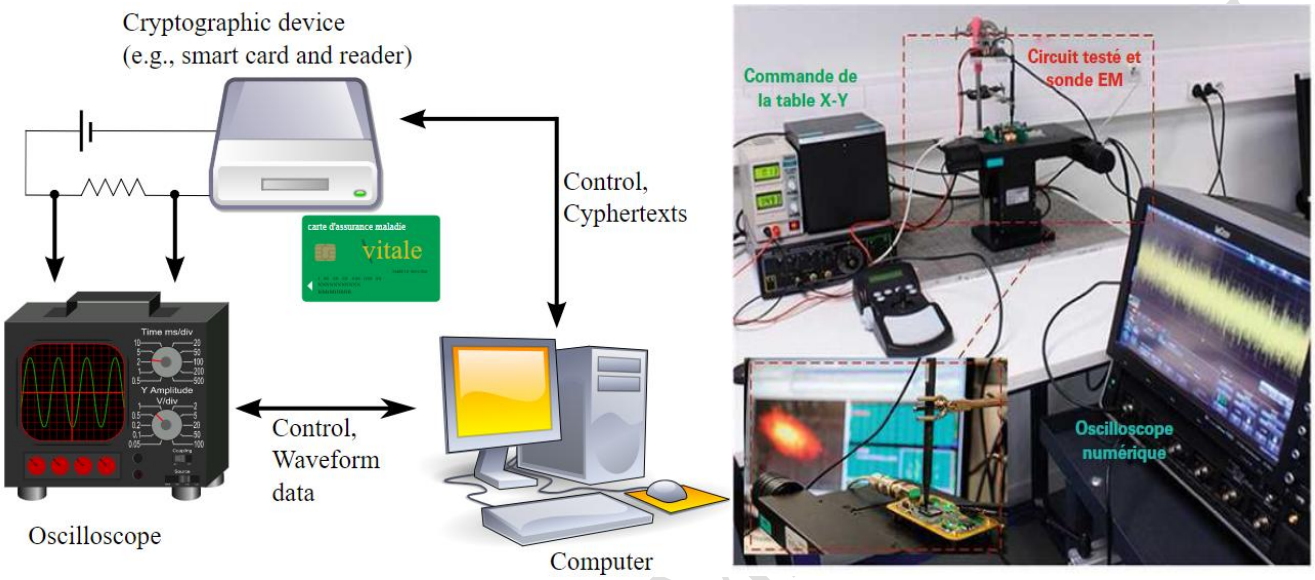


Figure 3.7 : Exemples des attaques physiques actives et passives

- Les attaques passives perturbant pas le fonctionnement du composant. Elles exploitent l'analyse des informations s'échappant de système électronique en surveillant en fonctionnement normal des canaux dits «cachés» émanant du circuit. Ces canaux sont cachés car ils n'apparaissent pas directement dans le fonctionnement d'un système électronique comme ses entrées et ses sorties. Les canaux cachés sont des canaux physiques existants naturellement dans les systèmes le long desquels de l'information s'échappe. Ces canaux sont par exemple, la consommation de puissance d'un système électronique intégré, son émanation électromagnétique, son rayonnement photonique, le temps nécessaire pour effectuer un calcul et tout ce qui peut être mesuré de l'extérieur et qui dépend de l'activité interne du circuit. Même si la quantité d'informations est très faible sur le canal, celle-ci peut être suffisante pour casser (c'est-à-dire trouver la clé secrète exploitée) un algorithme de chiffrement mathématiquement sûr comme l'algorithme AES (Advanced Encryption Standard). Effectivement les algorithmes de chiffrement modernes bien qu'extrêmement robustes théoriquement (c'est-à-dire mathématiquement) sont très sensibles aux fuites d'informations.

Un exemple de vulnérabilité est donné par les attaques de type « SPA » (*Single Power Analysis ou Analyse de Puissance Simple*) .Elles impliquent l'interprétation visuelle des traces de puissance ou des graphiques de l'activité électrique au fil du temps. Celles-ci sont fondées sur un espionnage des signaux électriques ou électromagnétiques émis par une machine. Selon la forme des signaux reçus, il est possible de déterminer l'activité de la machine, instruction par instruction, voire de « filtrer » certaines informations.

Il est également possible de connaître certaines données sensibles, comme des clés cryptographiques, en exploitant simplement le temps de traitement des différentes instructions du processeur (« *Timing attacks* »).

Les deux types d'attaques précédents ont fait l'objet de multiples variantes depuis plus de dix ans. Plus récemment, on a observé des attaques beaucoup plus sophistiquées comme la DPA (*Differential Power Analysis ou Analyse de Puissance Différentielle*) qui est connue depuis la fin des années 1990. Elle est une bonne illustration de la nécessité de considérer le matériel et le logiciel comme un couple indissociable. Les attaques (DPA) sont fondées sur une analyse statistique des variations de puissance consommée pendant l'exécution d'un algorithme cryptographique connu.

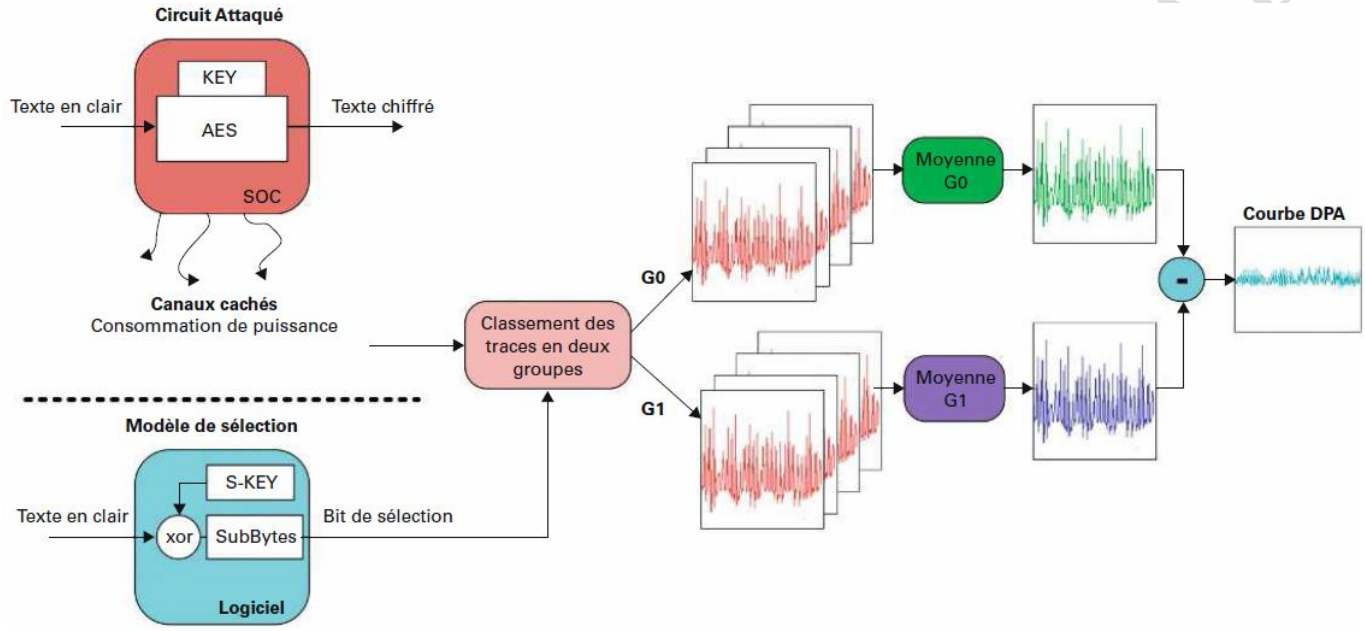


Figure 3.8 : Principe de l'attaque DPA proposée par Paul Kocher en 1999

3.2.3 Des mémoires intactes

Concernant le troisième point pour sécuriser un microprocesseur, on doit également contrôler l'intégrité des données et des programmes, ainsi que le déroulement et la reprise des diverses opérations en cas d'interruption brutale d'un processus pour une raison quelconque, comme la coupure de l'alimentation de la carte en cours de transaction (cas d'un arrachage de la carte, par exemple). Pour un système de sécurité, un état instable, voire « hors service », ne doit jamais exister. Nous n'en sommes pas là avec la plupart des OS (*Operating System Système d'Exploitation*) sur de nombreuses plates-formes informatiques traditionnelles.

3.2.4 Conserver l'union microprocesseur-logiciel

Concernant le quatrième objectif, une erreur classique consiste à considérer que la sécurité physique d'un microprocesseur est indépendante de la sécurité du logiciel qu'il supporte, ou réciproquement. En termes de sécurité, comme indiqué précédemment, le logiciel et le matériel sont inséparables, aussi doit-on consacrer des ressources supplémentaires pour réaliser des mécanismes sécuritaires. Ce point est très important, car cela nécessite l'implantation de fonctions dédiées au niveau du logiciel, dont le rôle est d'assurer l'interface avec le matériel, voire de compenser certaines de ses insuffisances.

Très schématiquement, les raisonnements appliqués partent du principe « diviser pour régner » : par exemple, pour l'accès aux ressources critiques du microprocesseur (mémoires, registres...), le système d'exploitation pourra être responsable de l'accès « logique », le matériel étant lui responsable de l'accès « physique ». Dans les composants actuels, des dispositifs très sophistiqués sont employés au niveau du matériel, de type ACL (« *Access Control List* ») ou MMU (« *Memory Management Unit* »), complétés par des mécanismes de cloisonnement des données et des applications supervisées par logiciel. Toute tentative d'accès frauduleux à une donnée ou à un programme doit ainsi être détectée et enregistrée, afin d'en tenir compte dans les étapes ultérieures. Dans certains cas, comme on ne peut discerner un mauvais fonctionnement d'une tentative frauduleuse, il est préférable d'assurer un mutisme complet du composant, afin de ne rien divulguer au fraudeur éventuel. Dans ce genre de composant, le comportement sécuritaire doit toujours prévaloir sur la divulgation d'un diagnostic, qui peut renseigner un pirate. En général, c'est le système d'exploitation des cartes qui supervise tous les cas litigieux et qui fixera la conduite à tenir en fonction du contexte.

4 Construction

4.1 Principes de construction

Une carte à puce est constituée de trois éléments : Une carte en matière plastique (souvent en PVC-chlorure de polyvinyle- ou ABS-acrylonitrile-budadiène-styrène), possédant ou non une piste magnétique ; Un module électronique, supportant les contacts électriques et Un circuit intégré en silicium (Puce).

Les puces sont réalisées par les fabricants de semi-conducteurs à partir de tranches circulaires de silicium en utilisant une série de masques pour réaliser la photolithogravure des circuits. Une même tranche (ou *wafer*) de huit pouces de diamètre peut ainsi contenir plusieurs milliers de puces rectangulaires, qu'il faut découper à l'aide de scies diamantées. Des machines automatiques viennent ensuite prélever ces puces pour les fixer au dos des modules, dans une cavité qui a été ménagée à cet effet. C'est l'opération dite «*die bonding*»

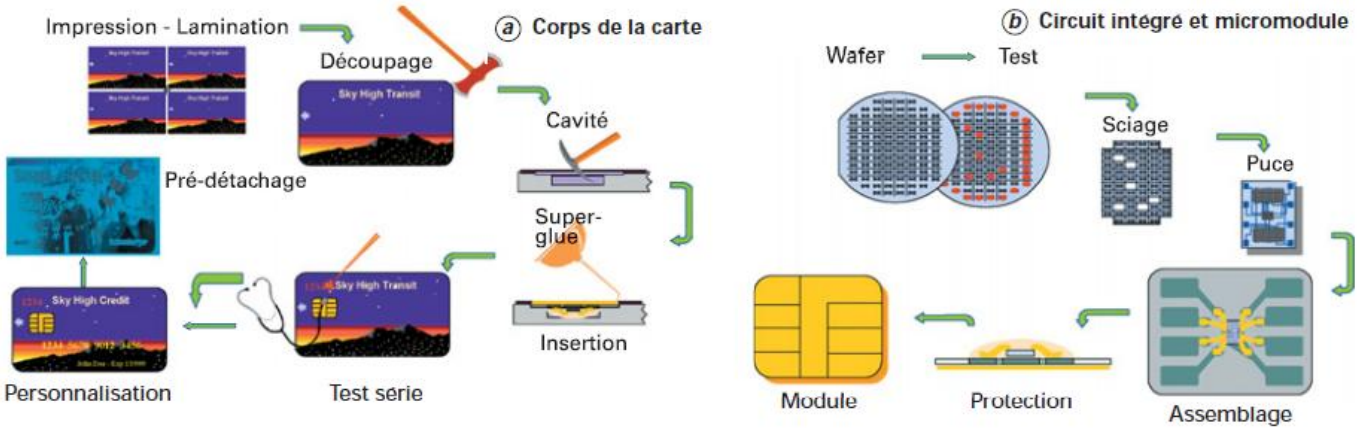


Figure 4.1 : Fabrication des cartes à puce

4.2 Interconnexion des composants

Cette opération consiste à relier électriquement les plots du composant aux contacts électriques du module. Ce câblage peut s'effectuer selon plusieurs techniques de base : le *wire bonding* se fait en utilisant du fil d'or ou d'aluminium, dont le diamètre est voisin de 10 µm à 20 µm et dont les soudures sur les plots des

composants sont réalisées par thermocompression ou ultrasons. Outre la fragilité relative des fils, cette technique, qui est la plus répandue, nécessite la répétition des soudures pour chaque extrémité des fils.

Une autre technique très fiable appelée TAB (*Tape Automated Bonding*) utilise un ruban continu découpé et soudé directement sur les composants en une seule opération par l'intermédiaire d'un métal d'apport. Elle est utilisée dans les premières cartes, mais elle a cédé la place à la précédente pour des raisons de coût.

4.3 Encartage

L'encartage consiste à fixer par collage les modules électroniques dans la carte, selon des procédés de fabrication permettant d'obtenir la qualité et la fiabilité requises. La carte terminée doit en effet résister aux agressions mécaniques et climatiques très sévères auxquelles elle est soumise dans les utilisations courantes.

Les modules sont encartés à l'aide de chaînes industrielles robotisées. Lorsque la carte comporte une piste magnétique, toutes ces opérations doivent être particulièrement maîtrisées afin de ne pas perturber le matériau et de conserver une parfaite planéité de la surface. La figure ci-dessous donne la coupe transversale d'une carte à puces montrant les différents éléments. Il faut noter que le module doit tenir dans une épaisseur inférieure à 0,6 millimètre, compte tenu de l'épaisseur normalisée des cartes qui est de 0,76 mm.

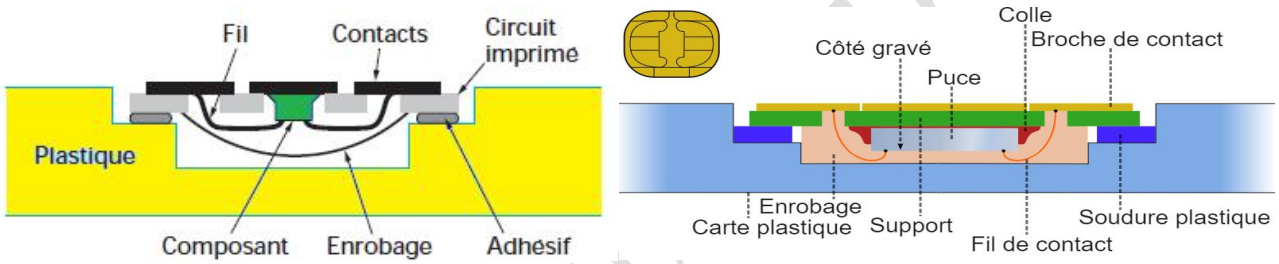


Figure 4.2 : Coupe transversale d'une carte à puce

Lorsque la carte est terminée, un test automatique vient vérifier que le composant fonctionne correctement avant d'être personnalisé. Cette dernière étape comprend deux phases distinctes :

- La personnalisation physique consiste généralement à estamper ou à graver des informations propres à l'utilisateur dans le plastique, à rajouter des dispositifs de sécurité contre la contrefaçon (hologrammes, guilloches, encres à effet optique variable, filigranes...) ;
- La personnalisation électrique consiste à encoder, s'il y a lieu, la piste magnétique et à inscrire les droits, les caractéristiques d'utilisation, les éléments de sécurité (clés, certificats...) dans la mémoire de la carte. Cette dernière opération devant s'effectuer de façon sécuritaire, elle est généralement réalisée après le contrôle de codes ou de clés de personnalisation, eux-mêmes calculés par des cartes mères qui sont capables de recalculer les clés diversifiées des cartes filles. En effet, chaque carte possède son propre jeu de clés pour protéger chaque composant pendant toutes les phases de la fabrication des cartes. Si un composant détecte une tentative d'intrusion non licite à une phase déterminée, il se bloque définitivement.

4.4 Connectique

Le module électronique supporte les contacts qui assurent les différentes liaisons électriques avec le monde extérieur. La métallurgie, l'état de surface et la topologie des contacts sont particulièrement délicats pour assurer une connectique fiable. Il est important de noter que la qualité des contacts est étroitement liée à

la conception du connecteur, utilisé dans les machines de traitement. Il existe maintenant sur le marché toute une gamme de connecteurs de technologie et de qualité très variables. Un connecteur à bas coût ne dépassera pas 10 000 manœuvres tandis que les meilleurs modèles peuvent dépasser plusieurs centaines de milliers de manœuvres dans des conditions difficiles. On aura soin de distinguer entre les connecteurs « frottants », qui peuvent entraîner des actions abrasives néfastes, et les connecteurs à contacts « atterrissants », qui ont une fiabilité nettement supérieure sans endommager les cartes.

5 Systèmes d'exploitation des cartes à puce

5.1 Généralités et mécanismes de base

Le terme système d'exploitation (*Operating System*, OS) désigne le logiciel de commande d'une carte à microcalculateur qui interprète et exécute les différents ordres élémentaires que cette carte peut réaliser. Situé dans la partie ROM du microcalculateur, il est implanté dans l'un des masques qui sert à la fabrication du circuit intégré. Ainsi par abus de langage, les notions d'OS et de « masque » sont couramment confondues.

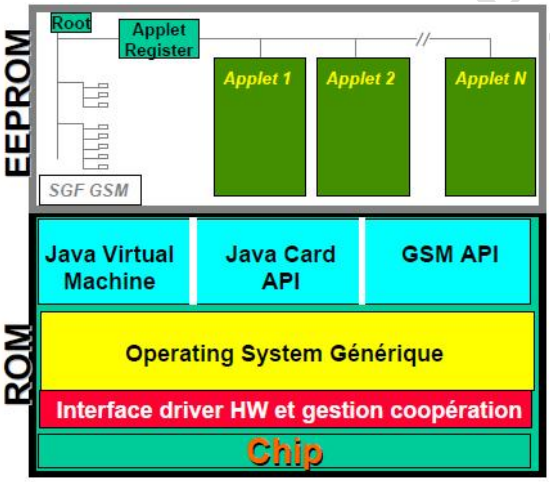


Figure 5.1 : Position du système d'exploitation

L'OS réalise principalement les fonctions suivantes :

- Gestion des échanges entre la carte et le monde extérieur, notamment le protocole d'échanges ;
- Gestion des différents fichiers et des données à l'intérieur de la mémoire ;
- Contrôle des accès aux informations et aux fonctions (par exemple sélection de fichier, lecture, écriture, modification de données) ;
- Gestion de la sécurité de la carte et de la mise en œuvre des algorithmes cryptographiques ;
- Fiabilité du fonctionnement, notamment cohérence et intégrité de certaines données, ruptures de séquences et reprise des erreurs ;
- Gestion du cycle de vie de la carte dans ses différentes phases (fabrication, personnalisation, utilisation, fin de vie).

L'implémentation d'un bon système d'exploitation doit tenir compte à la fois du faible coût, de la sécurité, de l'architecture de la machine, des performances, de la compacité, de la standardisation et de la

fiabilité. Le système d'exploitation et le composant ne font qu'un, ils constituent un couple indissociable, et cela est inhabituelle, pour qui développe du logiciel classique. Les OS de cartes se divisent en deux familles :

Les systèmes OS fermée, souvent liés à une application spécifique et qui possèdent des commandes adaptées à cette application, allant parfois jusqu'à la contenir elle-même ;

Les systèmes OS ouverts, autorisant le téléchargement (sécurisé) dans la carte de logiciels ou d'applications après sa délivrance à l'utilisateur.

Les OS pour cartes à puces sont généralement logés en mémoire ROM. Historiquement écrits en assembleur, optimisés pour une unique application. Les OS modernes sont maintenant principalement écrits en langage de haut niveau (C, voire C++), présentent un caractère « en couches », avec seulement les routines de bas niveau (celles liées à l'interface avec le Hardware) écrites en Assembleur. L'accroissement de la taille du code généré par la programmation de haut niveau est évidemment largement compensé par l'évolution des tailles de mémoire disponible dans les processeurs. De plus, l'écriture d'une large partie du code en langage de haut niveau est un gage important de portabilité du logiciel d'un composant à un autre (ce qui constitue, souvent, une exigence forte des utilisateurs). La figure ci-dessous présente l'architecture classique d'un OS moderne en couches pour cartes à puces. La couche de services génériques inclut en particulier un gestionnaire d'APDU (Application Programming Data Units), qui joue un rôle important et permet l'interfaçage avec toutes les applications (ce qui n'était pas le cas des OS spécifiques fermés sur un applicatif déterminé).

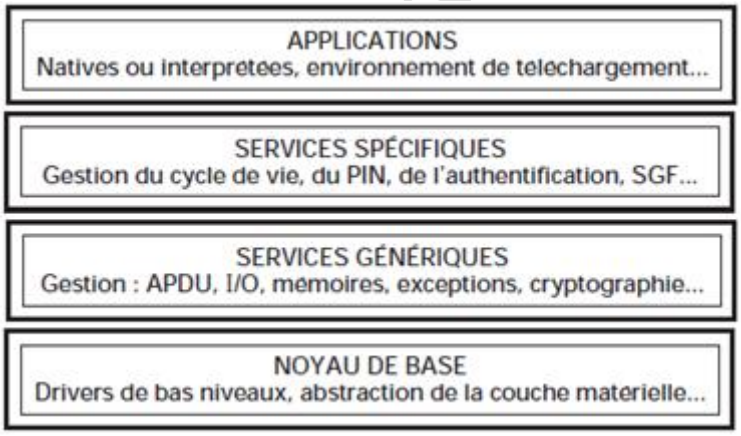


Figure 5.2 : Architecture en couches d'un OS moderne pour cartes à puces

Pour assurer la sécurité des accès et des fonctions, la couche générique fait appel à des algorithmes cryptographiques symétriques, mais aussi asymétriques, qui sont souvent associés à de nouveaux composants. Ces derniers intègrent des coprocesseurs, comportant à leur tour des opérateurs câblés, tels que des multiplications modulaires rapides ou encore à des microcalculateurs à architecture RISC (*Reduced Instruction Set Computer*), ou DSP (*Digital Signal Processor*).

Pour l'organisation des fichiers, la plupart des OS de cartes se rapprochent des recommandations de l'ISO qui a normalisé l'architecture arborescente. L'OS protège les différents fichiers en fonction du niveau où ils se trouvent. Il assure l'indépendance des applications et gère la transmission héréditaire des droits lorsqu'on accède aux fichiers à l'intérieur de la carte. Parmi les mécanismes de base d'un OS de la carte, trois

d'entre eux sont absolument fondamentaux : l'authentification dynamique de la carte, la signature électronique des messages et l'écriture sécurisée.

- L'authentification dynamique est réalisée à partir d'un dialogue aléatoire « Question-Réponse » avec le monde extérieur, de façon à permettre de prouver :
    a. Que la carte ou un fichier de cette carte ont bien été émis par un organisme habilité ;
    b. Que la carte n'a pas été falsifiée ou contrefaite.

Une carte peut authentifier un terminal par un processus identique dans l'autre sens ; on réalise ainsi une authentification mutuelle.

- La signature électronique, permet, suivant une norme ISO/IEC, d'authentifier des messages en vérifiant leur origine et leur intégrité. La certification des données de la carte est une extension de la notion de signature électronique, qui prouve qu'une information se trouve effectivement à l'intérieur d'une carte donnée. Les architectures correspondantes, appelées PKI (« Public Key Infrastructure ») ou IGC (« Infrastructure de gestion de clés ») s'appuient également sur un cryptosystème à clés publiques. Chaque utilisateur du système possède un certificat généré et géré par des autorités indépendantes (au moins une autorité d'enregistrement et une autorité émettrice). Le certificat original est souvent stocké dans une carte à puces, et permet de générer « à la volée », par la carte, des informations confidentielles (via sa clé privée) liées à une transaction. Ces données constituent en fait une signature numérique, et elles sont immergées de manière cryptographique avec d'autres données propres à la transaction et transmises au centre de traitement des transactions. Le serveur distant recalcule la signature à partir des données de transaction, et la compare avec celle qu'il peut construire à partir des données publiques de l'utilisateur. L'identité des deux signatures ainsi générées est une preuve d'opération licite, car seule la carte contenant le bon certificat aura pu engendrer un tel résultat.

- La troisième fonction fondamentale d'un OS de carte est : comment une carte est-elle née? Comment une application a-t-elle pu s'introduire dans une carte en toute sécurité ? Comment les clés cryptographiques peuvent-elles être créées ou modifiées dans une carte sans être compromises ?

L'écriture (ou la mise à jour) sécurisée répond à ce besoin essentiel, qui permet d'assurer le cycle de vie de la carte en toute sécurité. L'écriture sécurisée permet de transférer des données chiffrées à l'intérieur de la carte en authentifiant en même temps l'expéditeur. Ce processus est illustré sur la figure 5.3.

Cette fois, c'est la carte qui émet un nombre aléatoire N, sur la base duquel l'organisme habilité chiffre le message PT possédant une règle de syntaxe particulière. La carte déchiffre le message et vérifie la cohérence du résultat obtenu. Si le résultat est correct, la carte inscrit l'information dans sa mémoire. Une écriture sécurisée peut également se réaliser à l'aide d'une signature électronique vérifiée au préalable.

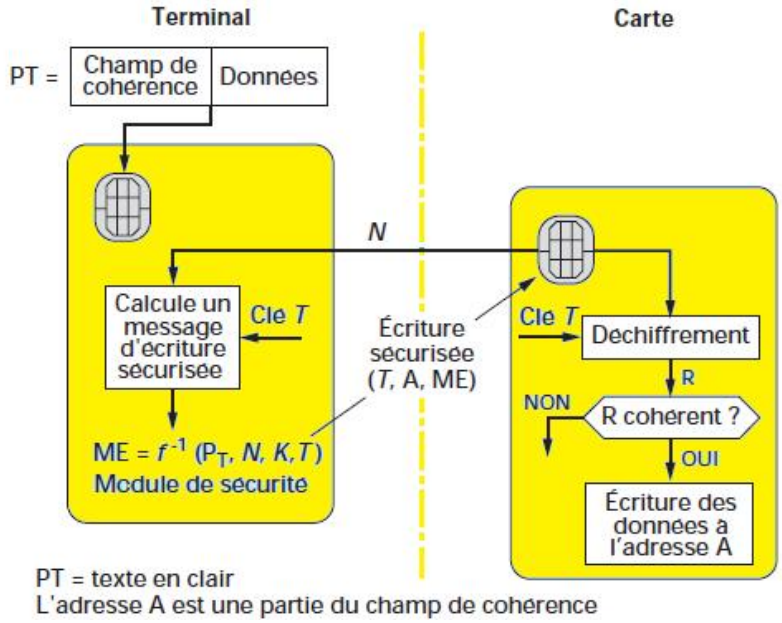


Figure 5.3: Ecriture sécurisée

5.2 Systèmes d'exploitation fermés

Ces OS, n'autorisent pas le chargement de code ou d'applications dans la carte après délivrance à l'utilisateur. Parmi eux, on distingue principalement trois catégories: Les OS monoprestataires à cryptographie symétrique; multiprestataires à cryptographie symétrique et multiprestataires à cryptographie asymétrique.

Un prestataire est l'entité responsable de la gestion des ressources de la carte et de la distribution des clés d'accès pour un ou plusieurs services. Il peut y avoir un ou plusieurs prestataires.

Un OS monoprestataire peut supporter en général plusieurs services, alors qu'un OS multiprestataire permet la cohabitation de plusieurs applications (chacune associée à un prestataire différent) dans la même carte. Chaque application peut à son tour abriter plusieurs services. Comme exemple de systèmes fermés on peut citer à titre d'exemple la version initiale de la carte SIM. De plus en plus, les OS pour cartes SIM sont en migration vers une architecture ouverte, construite autour du standard Java-Card, qui leur confère une structure de système multiprestataire ;

5.3 Systèmes d'exploitation ouverts.

Une carte à puces à OS ouvert est en quelque sorte une carte « programmable », c'est-à-dire une carte dans laquelle il est possible, en plus d'exécuter du code natif suivant le même principe que celui des systèmes fermés, soit de télécharger du code natif après délivrance de la carte, soit de programmer un interpréteur de commande particulier, et donc un jeu d'instructions complémentaire et des mécanismes de gestion de fichiers et/ou programmes additionnels. Cette définition est essentiellement théorique et se heurte immédiatement aux contraintes inhérentes à la sécurité et à l'espace mémoire disponible (i.e. la taille de la mémoire programmable, en l'occurrence l'EEPROM le plus souvent) dès que l'on veut la mettre en pratique.

5.3.1 Téléchargement de code natif

Le téléchargement de code natif ne peut être mis en œuvre de manière opérationnelle que si le composant de la carte dispose de tous les mécanismes de protection mémoire et de supervision adaptés. Dans

la négative, il est très difficile de restreindre a priori les fonctions d'accès aux mémoires du code embarqué. Lorsque le composant offre les garanties adaptées, il y a deux méthodes utilisables :

- Télécharger tout le programme dans un fichier élémentaire EF : Elementary File (Fichier de données), dont la structure est exécutable. Les paramètres d'exécution du code sont alors passés à la carte via la commande EXECUTE, éventuellement précédée d'étapes d'authentification ;
- Utiliser l'option « commande spécifique » loger toute l'application (incluant tous ses fichiers et commandes propriétaires) dans un unique fichier DF (Dedicated File- Répertoire + qq infos), dont l'espace mémoire est contrôlé par l'OS. La sélection du fichier en question et l'envoi à la carte d'une commande téléchargée (reconnue alors comme telle par la carte) permettent alors de déclencher l'exécution du code téléchargé. Le tous se construit à partir d'un fichier racine MF : Master File

5.3.2 Systèmes d'exploitation à interpréteurs

Le concept d'interpréteur est un environnement machine virtuelle embarqué dans une carte depuis 1996. Les principaux objectifs supportant les systèmes d'exploitation à interpréteurs sont les suivants :

- Permettre l'écriture d'un code unique, portable immédiatement sur tout nouvel environnement hardware (concept de « Write Once, Run Everywhere ») ;
- Permettre l'écriture de code applicatif téléchargeable par des tierces parties n'ayant pas nécessairement une connaissance approfondie de la technologie des cartes ;
- Garantir un haut niveau de sécurité, assurant une isolation parfaite entre applications.

À ce jour, trois principaux systèmes ouverts à interpréteur ont été développés pour la carte à puces :

- Multos, développé par MAOSCO pour le développement du porte-monnaie électronique MONDEX;
- Windows pour Smart Cards développé initialement par Microsoft ;
- JavaCard développé et soutenu par plus de 40 industriels réunis au sein du JavaCard forum.

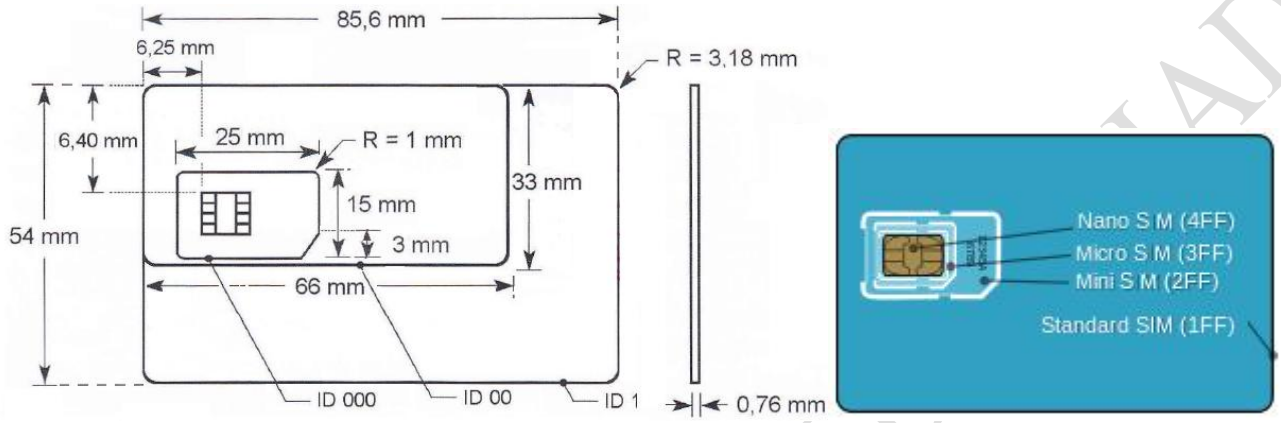
6 Normalisation

6.1 Normalisation des cartes à puce

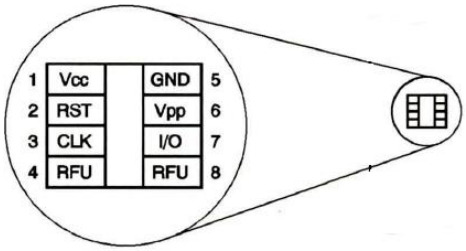
Jusqu'à ce jour il y'a déjà 13 standards internationaux génériques rassemblés au sein de la norme ISO tels que la Normes ISO/IEC 7816-x (x=1:16), Normes ISO/IEC 14443 (A,B,C) et 15693 pour cartes sans contact. Ces normes constituent le fondement normatif incontournable de la carte à puces.

- **7816/1 Caractéristiques physiques des cartes :**

  Définition du Format des cartes à puce et des contraintes physiques supportables (chaleur, humidité...)



- **7816/2 Dimensions et positions des contacts électriques de la puce**



□ Lecteur de cartes = CAD (Card Acceptance Device)

□ Surface ≤25 mm², □Épaisseur ≤0,3 mm

□ Composée de 8 contacts métalliques

□ VCC= 4,75 –5,25 V jusqu'à 200 mA

□ CLK: cap in/out < 30 pF, temps de transition < max(0,5 µs, 9%T)

□ I/O : Z état Haut-A état bas: quitte l'état haut seulement en transmission

□ Activation: RST bas, Vcc haut, VPP repos, I/O Z, CLK entre 1 et 5 MHz

□ Désactivation : RST bas, CLK bas, VPP inactif, I/O état A, VCC bas

□ RFU signifie tout simplement «Reserved for Future Use»·et indique que contacts sont réservés pour une utilisation future. Tels que pour 4 (MMC) USB (Multiple Media Card) USB Universal Seriai Bus ainsi que SWP(MMC) Single Wire Protocol

- **7816/3 Signaux électriques et les protocoles d'échanges**

□ Fréquence d'horloge 1 -5 Mhz, □Vitesse des communications < 115200 bauds :

□*Protocole de transmission*

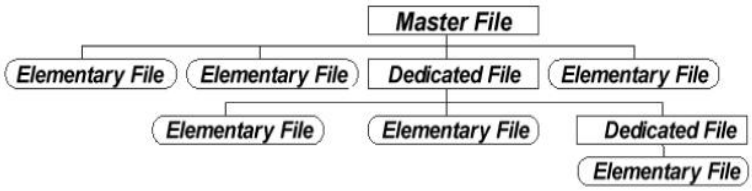
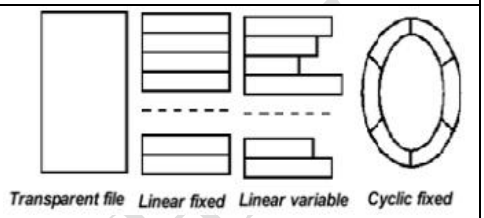
□TPDU (Transmission Protocol Data Unit), □T=0 Protocole orienté octet, □T=1 Protocole orienté paquet

□ Protocoles de communication asynchrones et half-duplex □T=2 Asynchrone, full duplex, orienté bloc

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

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

- **7816/4 commandes inter-industries**

Le système de 3 types de fichiers hiérarchique des cartes à puce. **MF :** Master File (Fichier racine), **DF :** Dedicated File (Répertoire+qq infos **EF :** Elementary File (Fichier de données)	
Protocole Asynchrone de type commande réponse *APDU (Application Programming Data Units)* READ BINARY, WRITE BINARY, UPDATE BINARY, ERASE BINARY, READ RECORD, WRITE RECORD, APPEND RECORD, UPDATE RECORD GET DATA, PUT DATA, SELECT_FILE VERIFY, NTERNAL_AUTHENTICATE, EXTERNAL_AUTHENTICATE, GET_CHALLENGE, GET_RESPONSE, ENVELOPE , MANAGE CHANNEL	

- **7816/5 : identifiants d'applications**

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

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

AID= chaîne de 16 octets

R premiers octets (RID) identifient le fournisseur d'application

Les 11 octets suivants représentent l'identifiant

Activation d'une application

Par exemple par SELECT_FILE (00 A4 04 00 10 [AID])

Exemple: ATR stocké dans ET_ATR en /3F00/2F01 est sélectionné par 00 A4 02 00 02 2F01

- **7816/6 éléments de données inter -industrie, formats de codage, et méthodes de récupération associées**

Spécifie les éléments de données inter-industrie

Spécifie les formats de codage, méthodes de récupération de ces éléments de données

Nom du porteur de la carte, Date d'expiration…

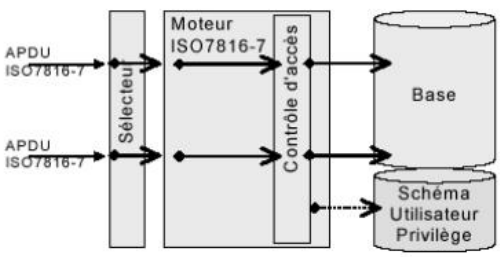
Etiquette	Longueur	valeur

- **7816/7 : commandes intersectorielle s pour langage d'interrogation de carte structurée (SCQL)**

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

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

2000 PicoDBMS : Un SGBD sur carte à puce



- **7816/8 : Sécurité de l'architecture et commandes pour les opérations de sécurité**
  - *Manage Security environment :* Passage d'un template à la carte
  - *Perform Security Operations*
    - Compute/Verify Cryptographic checksum
    - Encipher/Decipher/Hash
    - Compute/Verify Digital Signature
    - Generate Cryptographic key pairs
- **7816/9 : commandes pour la gestion administrative des cartes et de leurs fichiers**
  - Register File (DF or EF)
  - Create, Deactivate, Delete, Rehabilitate
  - Seulement si la carte est en environnement «sûr» !
  - Méthodes d'accès aux ressources Smart-Cards
- **7816/10 : La description des signaux électriques et ATR Spécifiques aux cartes synchrones**
- **7816/11 : vérification d'identité personnelle par moyens biométriques**
- **7816/12 : cartes à contact ; description de l'interface USB et procédures opératoires associées**
- **7816/15 : syntaxe et format des commandes et mécanismes cryptographiques.**

Les principales normes pour les carte à puce sans contacte

La norme générique pour les cartes sans contact *(distance carte lecteur de l'ordre de 10 cm)* :

- ISO/IEC 14443 partie 1 : caractéristiques physiques ;
- ISO/IEC 14443 partie 2 : interface radio fréquence et des signaux de communication ;
- ISO/IEC 14443 partie 3 : initialisation et anticollision ;
- ISO/IEC 14443 partie 4 : protocole de transmission.

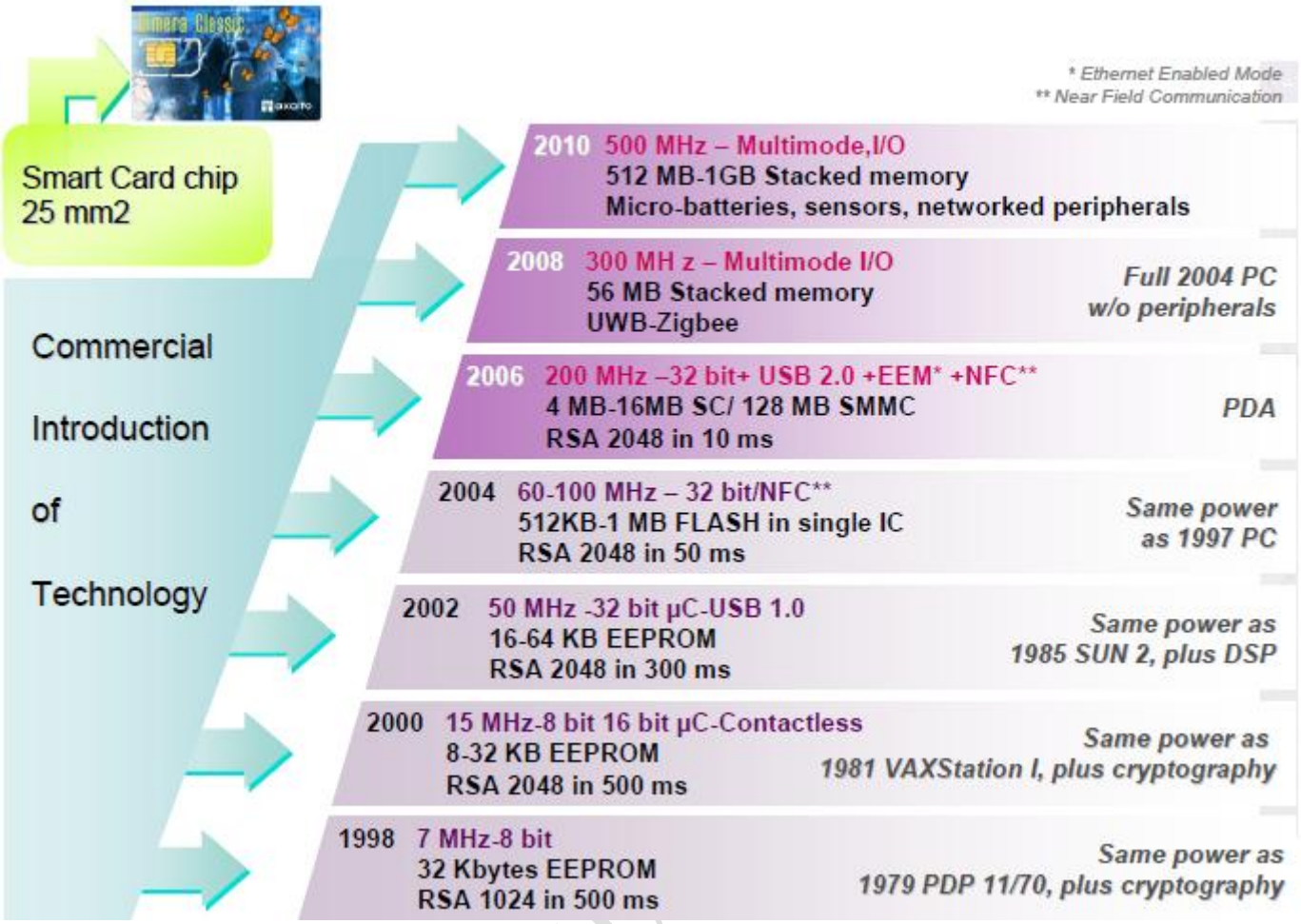
Les principales normes pour les cartes à puce dédiés à la Téléphonie mobile

Les standards clefs gouvernant le rôle de la carte SIM dans les réseaux GSM ont été définis à l'ETSI dans de nombreux documents. Parmi ces derniers, les cinq textes énumérés ci-après forment, à l'origine, une référence de base pour toute implémentation :

- GSM 11-11 : spécification de l'interface SIM-ME ;
- GSM 11-14 : spécification « SIM Application Toolkit » pour l'interface SIM-ME ;
- GSM 03.19 : API JavaCard de programmation pour la carte SIM Phase 2 ;
- GSM 03.40 : réalisation de la fonction « Short Message Service » (SMS) ; mode « Point to Point » (PP);
- GSM 03.48 : mécanismes de sécurité pour la carte « SIM application toolkit » Phase 2.

Ces standards ont été repris ou étendus dans le cadre des groupes de travail constitués par l'ITU pour la définition des systèmes de téléphonie mobile de 3e génération, notamment dans les documents répertoriés dans les familles TS 23.0x, TS.31.1xx et TS 102.2xx pour les systèmes W-CDMA (une vingtaine de documents au total).

Cartes à puce : Evolution du Hardware



Cartes à puce : Evolution du Software