La carte à puce (suite)

Samia Bouzefrane

Maître de Conférences

CEDRIC –CNAM

samia.bouzefrane@cnam.fr
http://cedric.cnam.fr/~bouzefra

Gestion des fichiers : Personnalisation de la carte

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Personnalisation

- Définir la structure des fichiers, conditions d'accès, mots de passe, etc.

- Écrire dans des registres ou fichiers de configuration de la carte des informations relatives :
  - aux fichiers utilisateur
  - aux fichiers de l'application

- Commandes utilisées :
  - WRITE BINARY, UPDATE BINARY
  - WRITE RECORD, UPDATE RECORD

- Une fois la carte personnalisée (positionnement d'un bit à 1), elle est prête à l'utilisation (opération irréversible)

- Exemple d'outil de personnalisation gratuit : CardEasy

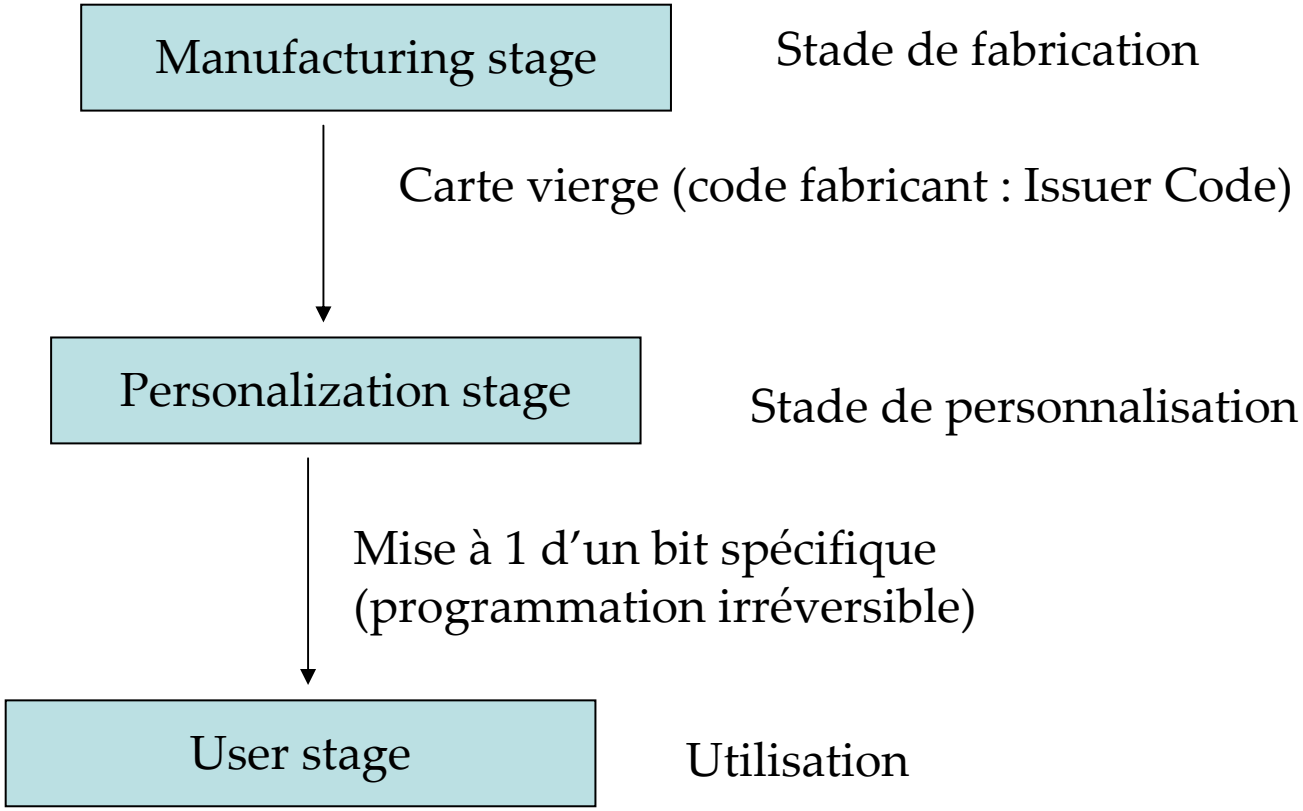
samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemple de la carte ACOS1

- ACOS1 produit de ACS (Advanced Card Systems, Hong Kong)
- EEPROM: 8 Ko (pour le stockage de fichiers)
- Protocole T =0
- ISO 7816-1-2-3
- Fichiers linéaires fixes
- Mots de passe: 5
- Cryptographie : DES, 3DES
- Dédiée à l'usage du porte-monnaie électronique

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemple de la carte ACOS1



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

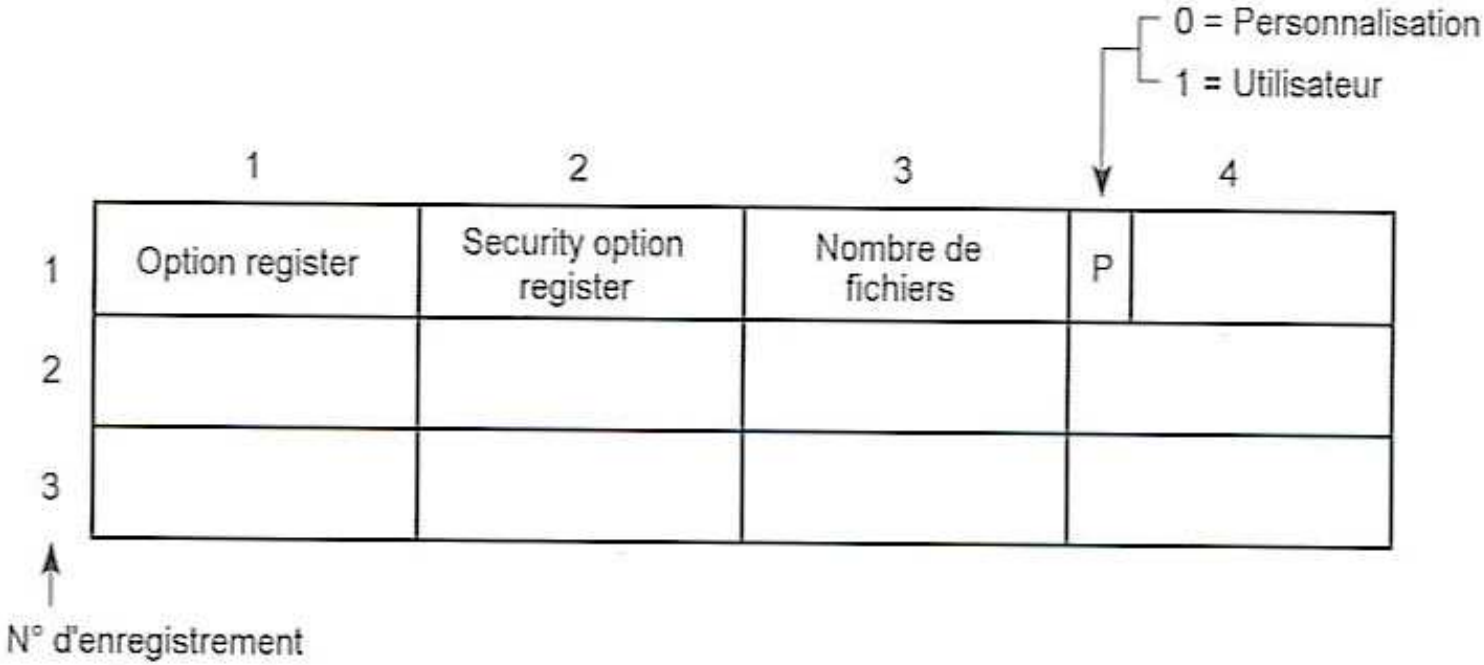
Les registres de configuration d'une carte ACOS1

Fichier		État de la carte	
Nom	FID	Personnalisation	Utilisateur
MCU-ID File	FF 00	L : libre E : non	L : libre E : non
Manufacturer File	FF 01	L : libre E : non	L : libre E : non
Personalization File	FF 02	L : libre E : code IC	L : libre E : non
Security File	FF 03	L : code IC E : code IC	L : non E : code IC
User File Management File	FF 04	L : libre E : code IC	L : libre E : code IC
Account File	FF 05	L : libre E : code IC	L : code IC E : code IC
Account Security File	FF 06	L : libre E : code IC	L : non E : code IC

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Personalization File

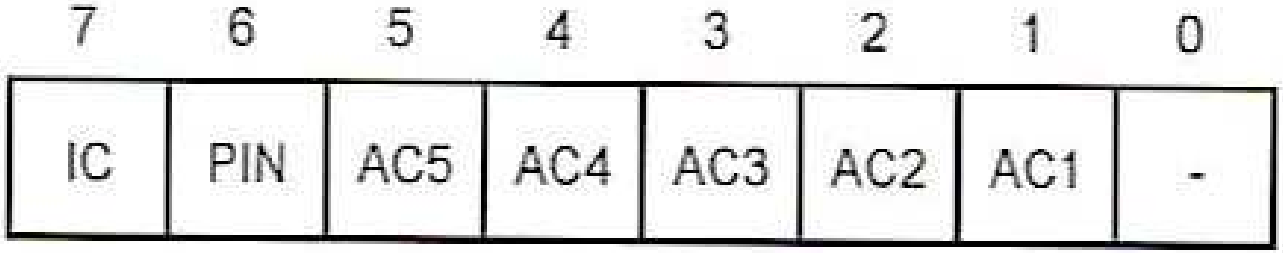
- **FID=FF 02**
- **3 enregistrements de 4 octets**



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Personalization File

- **4ème octet : bit de poids fort = 1 si fin de personnalisation et début d'utilisation**
- **3ème octet : nb de fichiers utilisateur que doit contenir la carte (fichiers accessibles par l'application)**
- **2ème octet: Security Option Register (correspond aux codes secrets de la carte). Si bit=1 le mot de passe doit être échangé de manière cryptée entre la carte et le lecteur, bit=0 échange en clair.**
- **1er octet : Option Register**



Security Option Register : Octet 2

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Option Register

7	6	5	4	3	2	1	0
INQ AUT	TRNS AUT	REV DEB	DEB PIN	DEB MAC	PIN ALT	3-DES	ACCNT

ACCNT	Validation de la structure de porte-monnaie électronique
3-DES	Validation du mode triple DES
PIN-ALT	Validation de la modification du code PIN par la commande CHANGE PIN
DEB_MAC	Validation de l'authentification par somme cryptographique MAC d'une opération de débit
DEB_PIN	Validation de la présentation du code PIN pour autoriser un débit
REV_DEB	Validation de la commande REVOKE DEBIT
TRNS_AUT	Validation de la nécessité d'une authentification mutuelle avant l'exécution de transactions de porte-monnaie électronique
INQ_AUT	Validation de la nécessité d'une authentification mutuelle avant l'exécution d'une commande INQUIRE ACCOUNT.

La fonction est validée lorsque le bit correspondant est mis à 1

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

User File Management File

- **FID=FF 04**
- **Gestion des fichiers utilisateur**
- **Actualisé automatiquement au reset de la carte**
- **Contient *n* enregistrements de 6 octets si *n* est le nb de fichiers utilisateur défini dans le fichier FF 02**

1	2	3	4	5	6
Longueur des enregistrements	Nombre d'enregistrements	Attributs de sécurité en lecture	Attributs de sécurité en écriture	FID poids forts	FID poids faibles

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

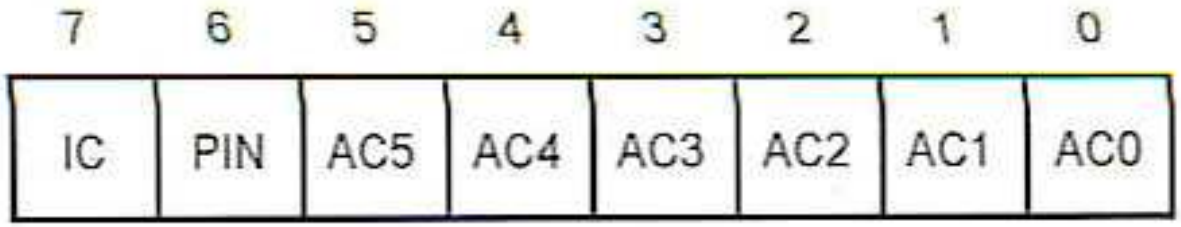
User File Management File

- 1er octet : contient la longueur des enregistrements du fichier concerné (taille maximale : 255 (FF))
- 2ème octet : nb d'enregistrements (au maximum 255)
- 3ème octet: attributs de sécurité en lecture.
- 4ème octet: attributs de sécurité en écriture.
- 5ème et 6ème octets : FID du fichier

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Attributs de sécurité en lect/écrit

- Codes secrets : IC, PIN, AC_0 à AC_5
- Bit PIN=0 => code PIN est requis pour l'accès au fichier
- Par ex., $octet_3$=01000010 (b_6=1, PIN) b_1=1, AC_1)
- Un AC_i par application et code PIN associé à toutes les applications



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Règles logiques de présentation des codes secrets

Attribut de sécurité	Règle d'accès correspondante
–	Accès libre
ACx	Présentation du code ACx correspondant
ACx, ACy, ACz	Présentation du code ACx OU du code ACy OU du code ACz
IC	Présentation du code IC
PIN	Présentation du code PIN
PIN, IC	Présentation du code IC ET du code PIN
ACx, IC	Présentation du code IC ET du code ACx
ACx, PIN, IC	Présentation du code IC ET du code PIN ET du code ACx
ACx, ACy, PIN	Présentation du code PIN ET du code ACx OU du code ACy
AC0	Aucun accès autorisé

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

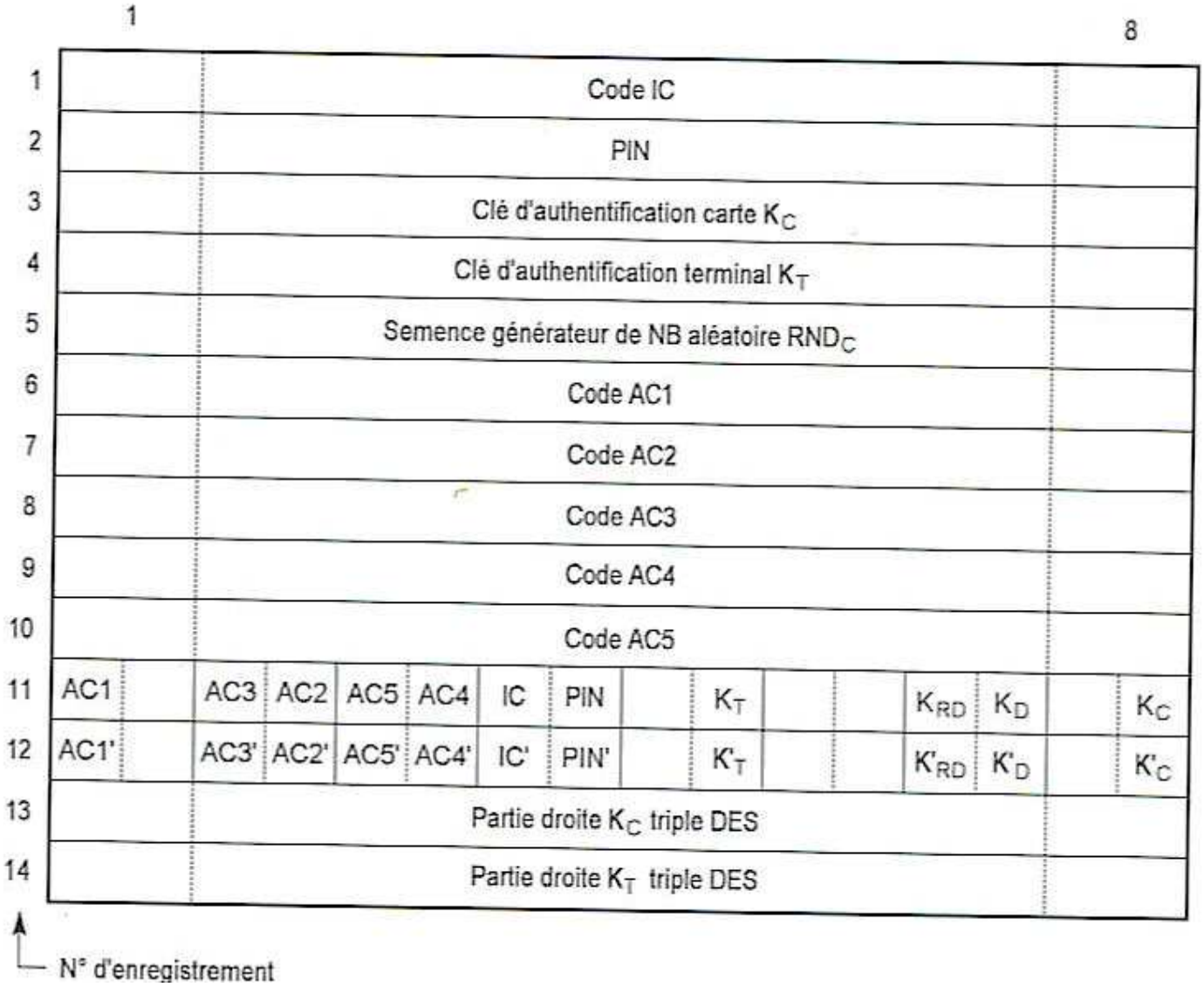
Security File

- FID=FF 03
- 14 enregistrements de 8 octets chacun
- Contient des informations secrètes
- Lecture du fichier impossible après la phase de personnalisation
- Reste accessible en écriture en vue de modifier les codes secrets (PIN)

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Security File

K_T et K_C clés du DES



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Security File

- **Les enregistrements 11 et 12 contiennent les compteurs associés à chaque code secret.**

- **Un compteur s'incrémente automatiquement de 1 à chaque présentation erronée du mot de passe**

  -Blocage de la carte lorsque la valeur = 8

  -Remise à 0 à chaque présentation de mot de passe valide

  -Compteurs non modifiables une fois la phase de personnalisation terminée

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Quelques notions sur les algorithmes cryptographiques

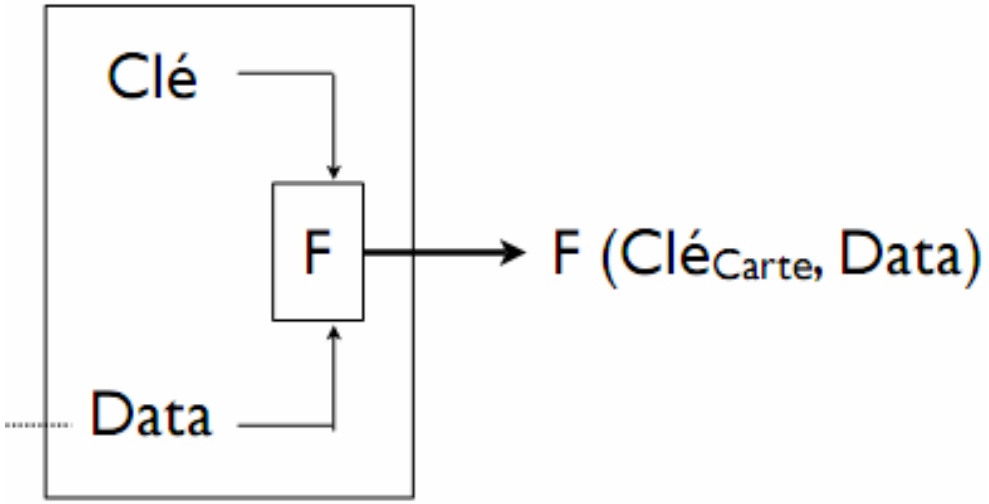
Propriétés des cartes

- Les cartes sont potentiellement de l'argent, un moyen d'accès à un service
- Besoin de sécuriser l'accès à un local, à un compte bancaire, etc.
- Besoin de sécuriser les informations se trouvant sur la carte

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Propriétés des cartes (suite)

- Dispositif de protection de clé(s) cryptographiques
- Des mots (zones) mémoires peuvent être utilisés pour mémoriser des clés cryptographiques :
- Confidentialité de la clé (personne ne peut la lire)
- Intégrité de la clé et du calcul de la fonction F (personne ne peut les modifier)



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

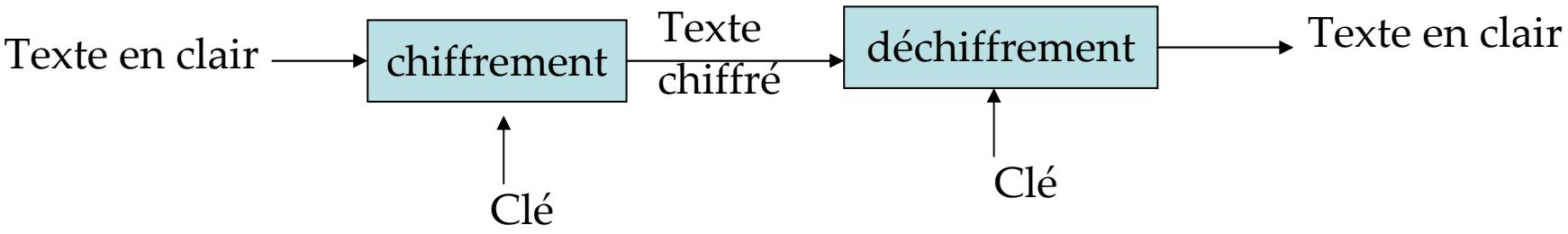
Algorithmes à clé secrète/publique

➢**Algorithmes restreints (secrets) connus de l'expéditeur et du destinataire**
Si fuite de l'algorithme => la sécurité est cassée

➢**Algorithmes publics (connus) mais utilisent des informations secrètes**

- Algorithmes à clé secrète

- Algorithmes à clé publique



Algorithme à clé secrète

- L'expéditeur et le destinataire se mettent d'accord sur une clé (qui servira pour le chiffrement/déchiffrement)
- Exemple : le DES, le triple DES ou l'AES
- Inconvénient : La clé étant secrète, toute interception de cette clé casse la sécurité du système

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Algorithmes à clé publique

- Le destinataire diffuse sa clé publique
- Le texte en clair est chiffré avec la clé publique
- La clé publique étant échangée, elle peut être interceptée
- Mais le déchiffrage du texte reçu par le destinataire nécessite la clé publique associée à une clé privée (qui est connue uniquement du destinataire et n'est donc pas échangée avec les autres correspondants)

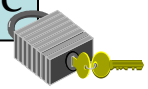
samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Exemple

Alice **Bob**

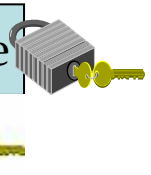
Clé du cadenas1 → Ne peut pas lire le message

Message



Enlève son cadenas car elle a la clé ← Met un cadenas2

Message



Enlève le cadenas2 →

Message

Message

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

DES : Algorithme cryptographique à clé secrète

- **Proposé initialement par Horst Feistel (les années 70) sous le nom de Lucifer**
- **Normalisé en 1976 sous le nom de DES (Data Encryption Standard)**
- **Standard utilisé depuis 30 ans malgré quelques faiblesses dues à la puissance des moyens de calcul d'aujourd'hui**

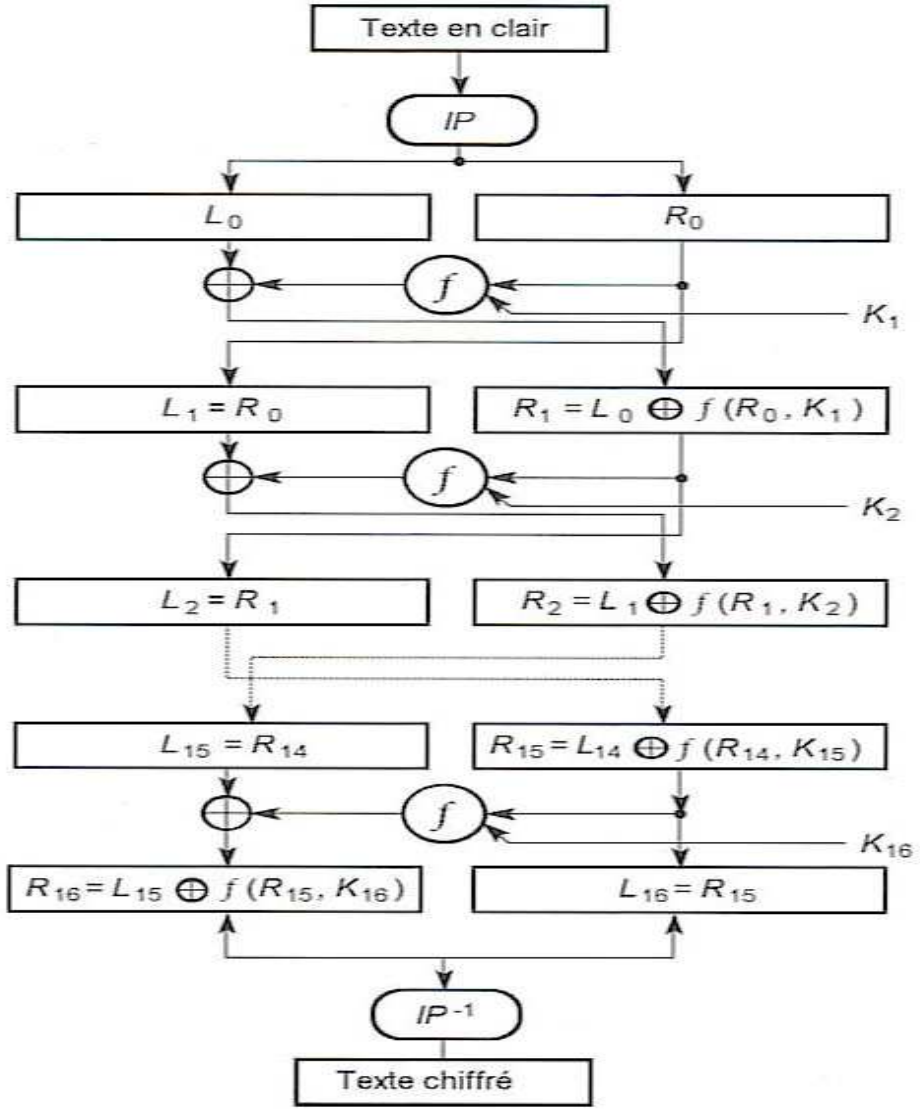
samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Principe du DES

- Algo de chiffrement par blocs
- Blocs d'entrée de 64 bits
- Blocs de sortie de 64 bits
- Bourrage des données si elles sont inférieures à la taille d'un bloc
- Algo à clé secrète de 56 bits (8 mots de 8 bits avec le 8ème bit de chaque octet inutilisé ou bit de parité)

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Principe du DES (suite)



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Principe du DES (suite)

- ***f***: fonction de permutation utilise des tables de permutation définies dans la norme
- + : ou exclusif
- K_i: obtenu par décalage des bits de la clé initiale en utilisant une table
- L, R: partie gauche (32 bits) partie droite (32 bits)
- Le déchiffrage fait les opérations inverses

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Algorithme DES

➢Si une carte utilise un DES (voir INTERNAL/EXTERNAL AUTHENTICATE), le terminal doit aussi implanter cet algorithme.

➢ Des versions programmées du DES existent (gratuites ou commerciales)
➢ Des sources de cryptage/décryptage en C++ :

http://www.cppfrance.com/codes/CLASSE-DATA-ENCRYPTION-STANDARD_31759.aspx

➢Des exécutables sous DOS et des sources en Basic :
http://www.planet.nl/error_404.html

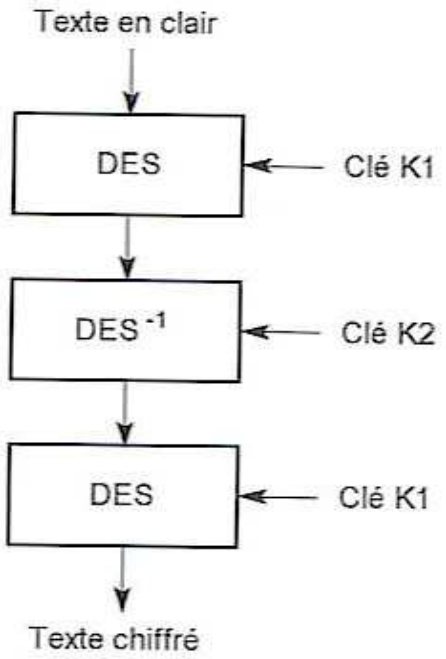
➢ CryptoTools 3.0 (DES, triple DES, RSA), version gratuite :
http://www.cryptotools.com/index.aspx

➢ chargement gratuit de plusieurs algorithmes :
http://us.cryptosoft.de/html/home.htm

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Triple DES

- Nécessite deux clés
- DES^{-1} décryptage avec une $2^{ème}$ clé
- Pour casser un triple DES, il faut : 2^{56} DES successifs
- Si on peut casser un DES en 1s alors il faudra 2^{56} s=2,2 millions d'années pour casser un triple DES
- Inc. Plus lent que le DES



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

L'AES

- **AES : Advanced Encryption Standard**
  résultat d'un appel d'offres déposé en Janvier 1977 par le NIST (National Institute od Standards & Technologies)

- **Cahier des charges**
  - **Grande sécurité**
  - **Portabilité maximale d'un petit processeur 8 bits aux cryptoprocesseurs de chiffrement**
  - **Rapidité**
  - **Lecture facile de l'algorithme (public)**
  - **utilisation de blocs de 128bits et d'une clé sur 128, 192 ou 256 bits**

- **21 candidatures déposées auprès du NIST**

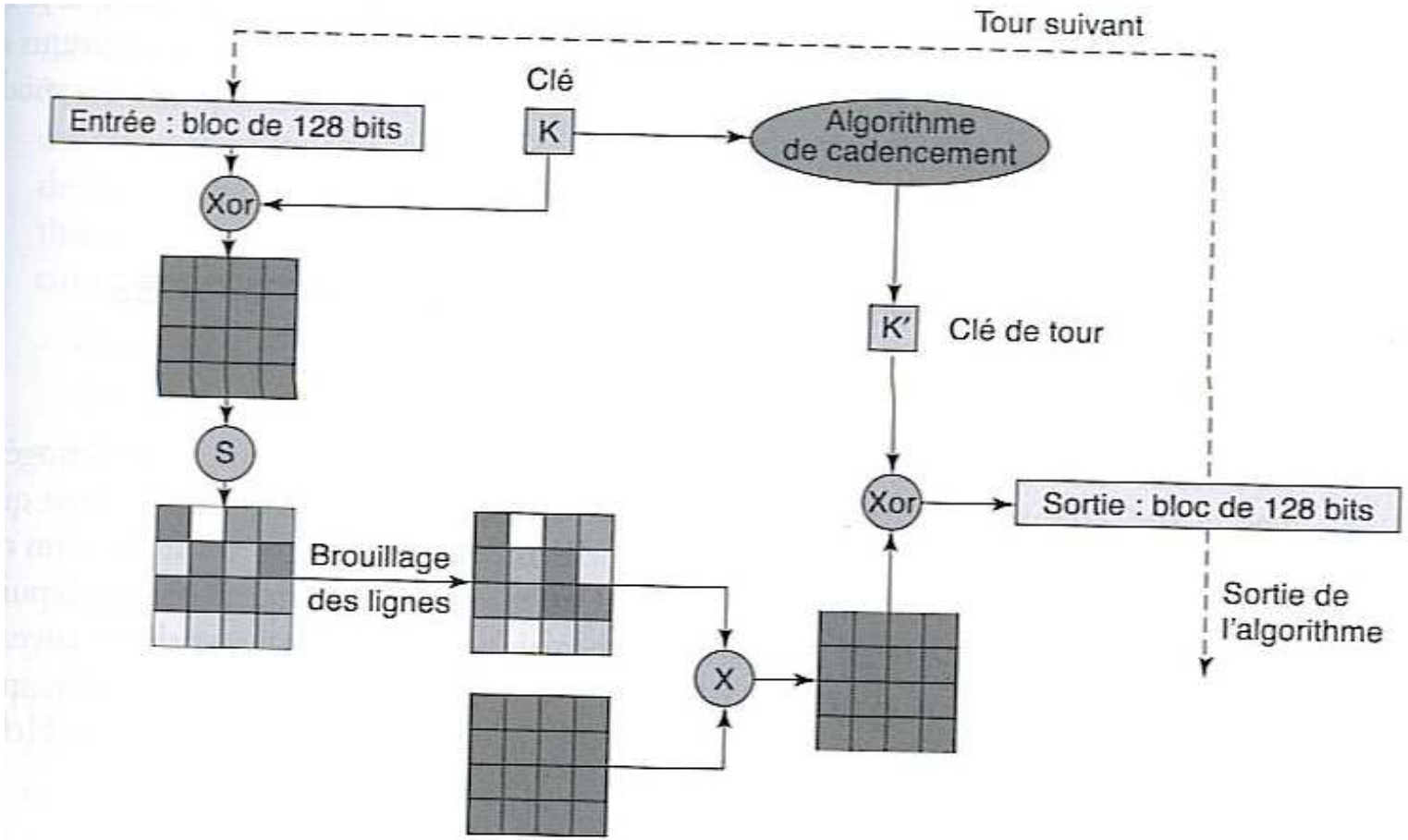
samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

AES

- Algorithme retenu en octobre 2000 de Joan Daemen et Vincent Rijmen
- Travaille sur des blocs de 128 bits avec les options suivantes :
  - 128-128 (option utilisée par les cartes à puce)
  - 128-192
  - 128-256
- L'algorithme comporte 10 tours
- 1 tour = 5 transformations, S une fonction non linéaire

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

AES



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

AES

- **Facile à implémenter sur le plan matériel ou logiciel**
- **Algo rapide (6 fois plus vite qu'un DES) malgré une clé 2 fois plus longue**
- **Il faut 2^{128} secondes pour casser un AES = 150 000 milliards d'années**

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

RSA : Algorithme à clé publique

- **RSA (Ron Rivest, Adi Shamir & Léonard Adleman)**
- **Basé sur le principe des algorithmes de chiffrement asymétriques découvert en 1975 par Diffie, Hellman et Merkle**
- **Le texte est chiffré avec une clé publique**
- **La clé publique est associée à une clé privée qui permet de déchiffrer le message**
- **Algorithme réversible : on peut chiffrer avec la clé privée et déchiffrer avec la clé publique**

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

RSA : principe

- **Repose sur les propriétés des nombres premiers**

- **Fabrication de clés :**
  - choisir 2 grands nombres premiers *p* et *q* et faire le produit : *n*=*p*q***
  - choisir une clé de chiffrement aléatoire *e* tel que *e* et (*p*-1)*(*q*-1) soient premiers entre eux.
  - utiliser l'algorithme d'Euclide pour déterminer une clé de chiffrement tel que : *d*= (1/*e*) modulo (*p*-1)*(*q*-1)

- ***n* et *e* = clé publique (*p* et *q* doivent être détruits après l'opération)**

- ***d* : la clé privée (ne doit pas être révélée)**

- **Chiffrement : $C=M^e$ modulo *n***

- **Déchiffrement : $M=C^d$ modulo *n***

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Algorithmes de cryptage proposés dans Java Card 2.2.2

- AES: Advanced Encryption Standard (FIPS-197)
- SEED Algorithm Specification : KISA - Korea Information Security Agency Standard Names for Security and Crypto Packages
- SHA (SHA-1): Secure Hash Algorithm, as defined in Secure Hash Standard, NIST FIPS 180-1
- SHA-256,SHA-384,SHA-512: Secure Hash Algorithm,as defined in Secure Hash Standard,NIST FIPS 180-2
- MD5: The Message Digest algorithm RSA-MD5, as defined by RSA DSI in RFC 1321
- RIPEMD-160: as defined in ISO/IEC 10118-3:1998 Information technology – Security techniques - Hash-functions - Part 3: Dedicated hash-functions
- DSA: Digital Signature Algorithm, as defined in Digital Signature Standard, NIST FIPS 186
- DES: The Data Encryption Standard, as defined by NIST in FIPS 46-1 and 46-2
- RSA: The Rivest, Shamir and Adleman Asymmetric Cipher algorithm
- ECDSA: Elliptic Curve Digital Signature Algorithm
- ECDH: Elliptic Curve Diffie-Hellman algorithm
- AES: Advanced Encryption Standard (AES), as defined by NIST in FIPS 197
- HMAC: Keyed-Hashing for Message Authentication, as defined in RFC-2104

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Commandes relatives à la sécurité

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Architecture de sécurité

➢**Statut de sécurité : État atteint après l'achèvement de fonction de sécurité comme :**

- Présentation d'un code secret (VERIFY Command),

- Connaissance d'une clé (GET CHALLENGE puis EXTERNAL AUTHENTICATE ou d'un SECURE MESSAGING).

➢**Les statuts :**

- **Global (sur le MF),**
- **Relatif à une application (DF), lors d'une (dé)sélection le statut sera modifié en fonction des règles de l'application,**
- **Spécifique à un fichier,**
- **Spécifique à une commande.**

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les fonctions de sécurité

- **Authentification par mot de passe**

- **Authentification par clé à l'aide des fonctions (GET CHALLENGE suivi de EXTERNAL AUTHENTICATE, regroupées aussi dans GENERAL AUTHENTICATE)**

- **Authentification des données :**
  - **la carte compare des données reçues à celles dont elle dispose en utilisant des données qui lui sont propres (clé secrète ou clé publique),**

- **Chiffrement de données.**

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Commande VERIFY

➢ **Compare les données de la cde avec celles de la carte (ex. mot de passe)**
Ex: C0 20 00 00 05 70 61 73 73 65 (passwd='passe')

Commande APDU						
Entête obligatoire				Corps optionnel		
CLA	**INS**	**P1**	**P2**	**Lc**	**Data field**	**Le**

- CLA: dépend de l'application
- INS: "20"
- P1 = 0
- P2 :

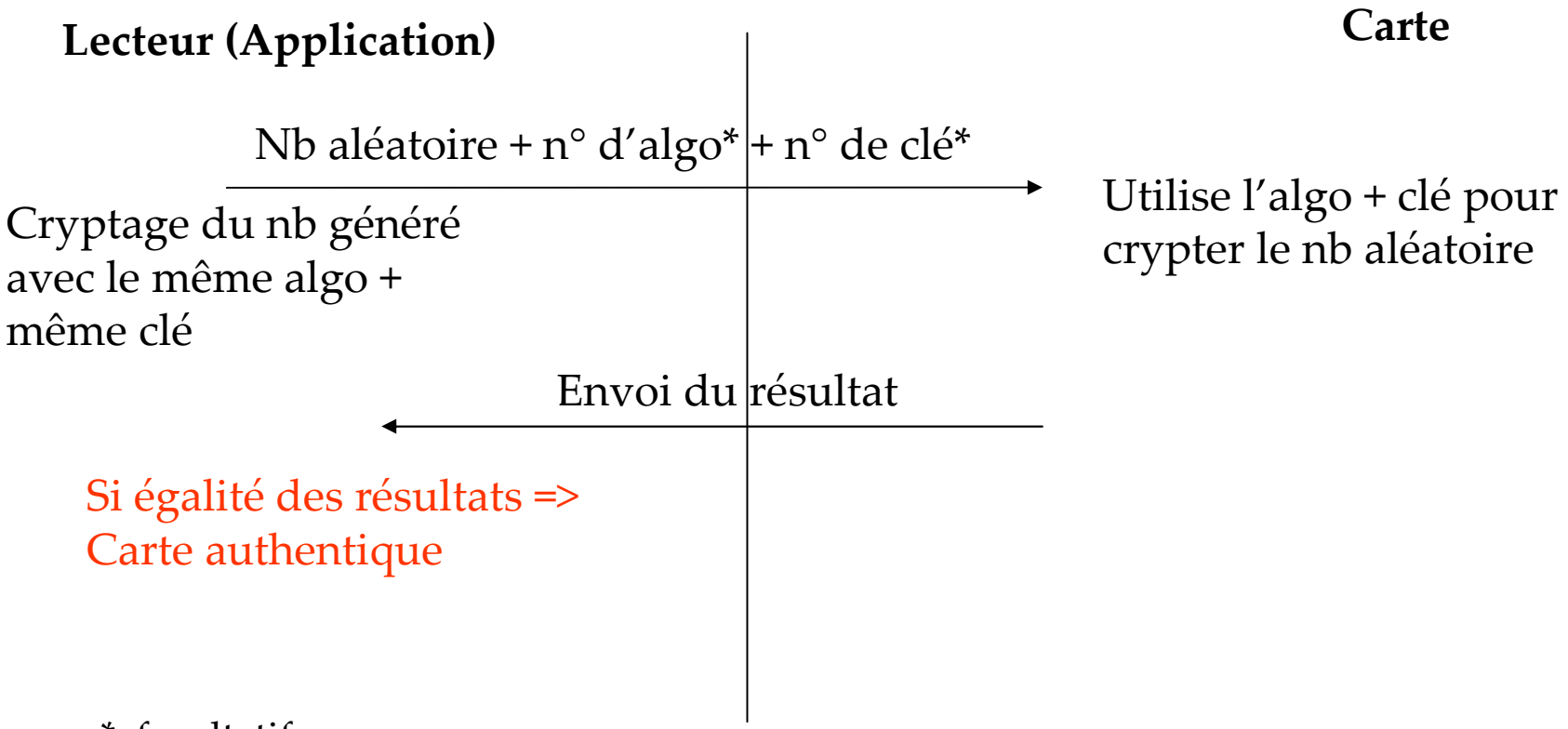
Si b_7=0, mot de passe au niveau carte, b_7=1 mot de passe au niveau répertoire, etc.

- Data field : contient le mot de passe envoyé

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Commande INTERNAL AUTHENTICATE

- **Permet l'authentification de la carte vis-à-vis de l'application**



*: facultatif

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Commande INTERNAL AUTHENTICATE

➢ **Permet l'authentification de la carte vis-à-vis de l'application**
Ex: C0 88 00 00 03 03 02 01 03

Commande APDU						
Entête obligatoire				Corps optionnel		
CLA	INS	P1	P2	Lc	Data field	Le

- CLA: dépend de l'application
- INS: "88"
- P1 =0 : algo par défaut, sinon contient le numéro de l'algorithme
- P2 : (b_7=0) authentification au niveau carte, (b_7=1) authentification au niveau répertoire. Il contient aussi la clé en clair ou le FID du fichier qui contient la clé. Si le contenu de P2=0 alors la clé est connue par la carte.
- Data field : contient le nb aléatoire

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Commande GET CHALLENGE

➢ **Demande à la carte de générer un *challenge* (nb aléatoire) utilisé par la cde EXTERNAL AUTHENTICATE**
Ex: C0 84 00 00 10

Commande APDU						
Entête obligatoire				Corps optionnel		
CLA	INS	P1	P2	Lc	Data field	Le

- CLA: dépend de l'application
- INS: "88"
- P1 = 0
- P2 = 0
- Le : taille du nb aléatoire attendu

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Commande EXTERNAL AUTHENTICATE

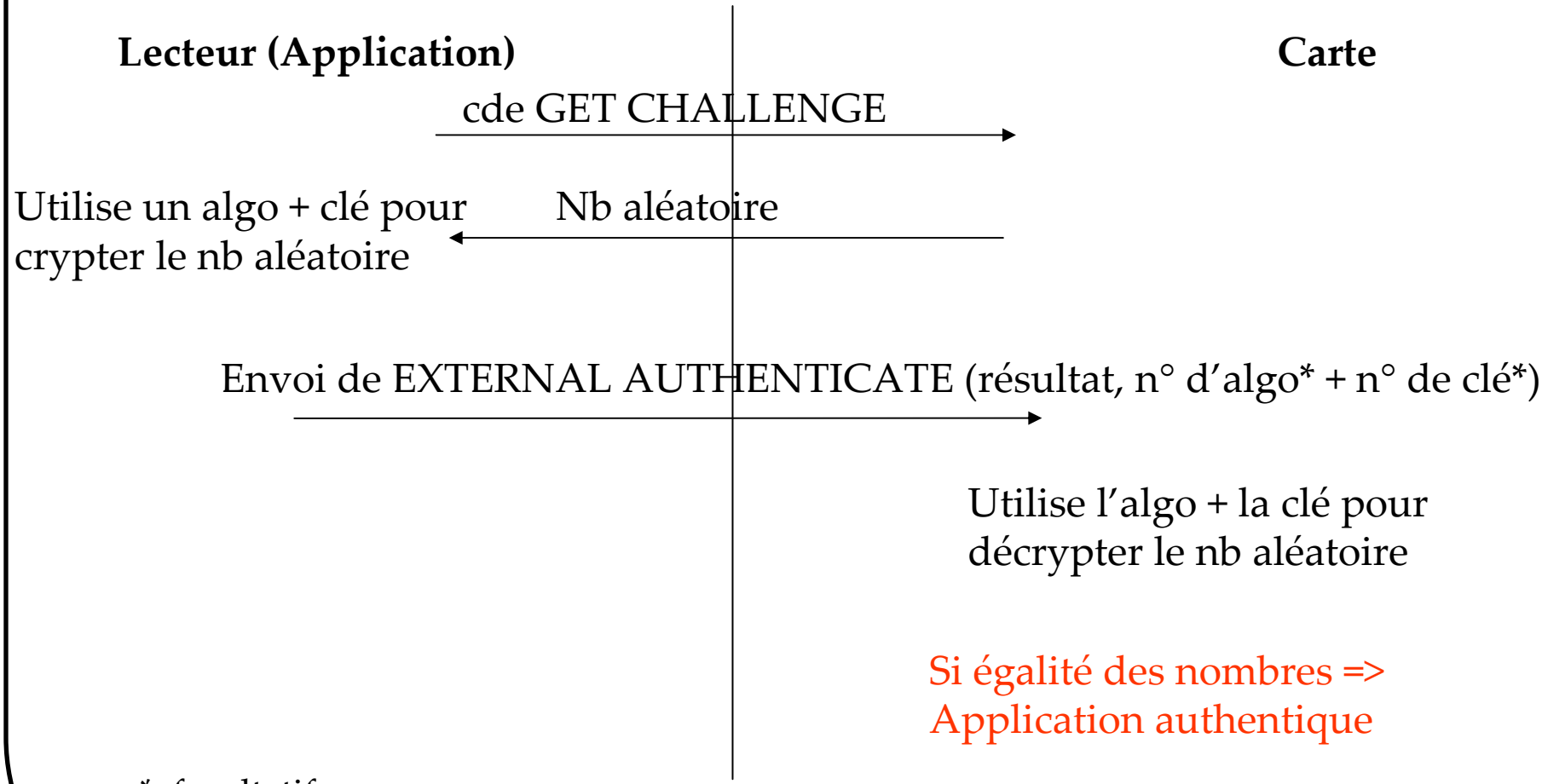
- **est la réciproque de la cde INTERNAL AUTHENTICATE**

- **permet d'authentifier l'application pour que le terminal puisse accéder à des données sensibles de la carte**

- **comme la carte ne génère pas spontanément de nb aléaoire avant la cde EXTERNAL AUTHENTICATE, le terminal envoie la cde GET CHALLENGE à la carte afin que celle-ci génère un nb aléatoire**

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Commande EXTERNAL AUTHENTICATE



*: facultatif

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Commande EXTERNAL AUTHENTICATE

➢ **Permet d'authentifier l'application pour que le terminal puisse accéder à des données sensibles de la carte**
Ex: C0 82 00 00 03 03 02 01

Commande APDU						
Entête obligatoire				Corps optionnel		
CLA	INS	P1	P2	Lc	Data field	Le

- CLA: dépend de l'application
- INS: "82"
- P1 : contient le numéro de l'algo, Si P1=0, l'algo est connu de la carte
- P2 : (b_7=0) authentification au niveau carte, (b_7=1) authentification au niveau répertoire. Il contient aussi la clé en clair ou le FID du fichier qui contient la clé. Si le contenu de P2=0 alors la clé est connue par la carte.
- Data field : contient le résultat du cryptage

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Commande ENVELOPE

➢ **Permet d'envelopper des données (APDU ou morceau d'APDU cryptés)**
Ex: C0 C2 00 00 07 C0 A4 00 00 02 3F 00

Commande APDU						
Entête obligatoire				Corps optionnel		
CLA	INS	P1	P2	Lc	Data field	Le

- CLA: dépend de l'application
- INS: "C2"
- P1 = 0
- P2 = 0
- Data field : contient les données enveloppées

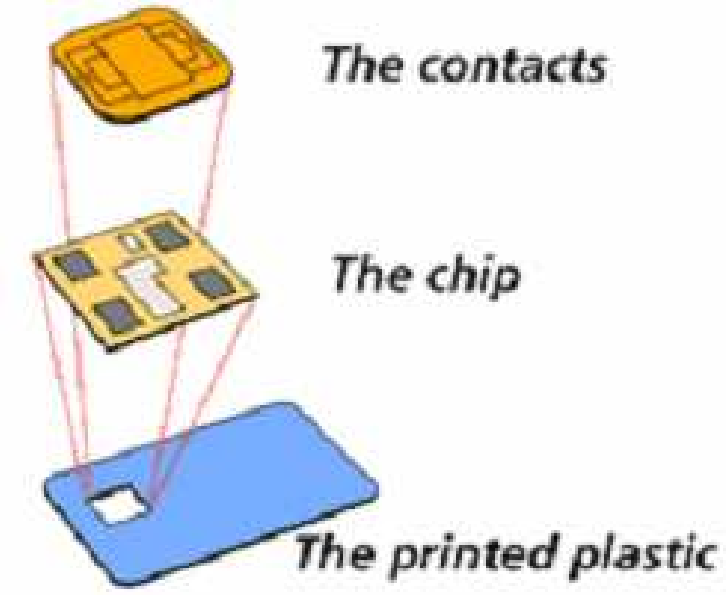
samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Le logiciel de la carte

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Cycle de vie d'une carte

- **Fabrication du support ou corps de la carte**
- **Plastique laminé, ABS coulé**
- **Fabrication du composant (galette de silicium ou « wafer »)**
- **Fabrication du module :**
  - **Découpage/Sciage,**
  - **Contact/Binding,**
  - **Protection dans le module,**
  - **Collage.**



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Cycle de vie d'une carte (suite)

- **Personnalisation : la carte comporte généralement le SE (il est inscrit en ROM)**

- **La personnalisation consiste :**

  - **À finir de paramétrer le SE**
    - **taille des mémoires utilisateur !**

  - **Renseigner les informations de l'application par rapport aux :**
    - **émetteur et ses services**
    - **porteur et ses services**

- **Distribution**

- **Vie de la carte :**
  - **Utilisation,**
  - **Perte, fin de vie.**

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Écriture du logiciel pour la carte

- **Les langages et méthodes :**
  - ✓ **Assembleur au début**

  - ✓ **Début des années 1990 arrivée de C/Pascal interprété (gain de place au niveau du code, prototypage facile) mais manque d'efficacité**

  - ✓ **Puis langage C, processeurs plus gros et plus adaptés**
    Av. : proche des machines, efficace, disponible,
    Inc. : un peu « bidouille ».

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Écriture du logiciel pour la carte (suite)

- Puis "Java Card Technology", la raison du passage à Java n'est pas liée au langage et à ses bonnes propriétés...
  - ✓C'est lié au besoin d'adaptabilité des services des applications.
- Les challenges et enjeux aujourd'hui sont liés à :
  - ✓L'augmentation de la complexité du code (plusieurs 100 ko), même si celui-ci reste « petit » au regard d'autres industries, les contraintes sont fortes :
    - taille/efficacité/sécurité, QUALITE,
    - Arrivée de nouveaux concepts.

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les concepts émergents

- La carte va vivre dans les 3 à 5 ans une nouvelle vague d'innovations dans le domaine du logiciel :
  - ✓Nouveau composant (aujourd'hui sous utilisé) : MMU/MPU, mode user/super user,
  - ✓Multi-tâche,
  - ✓Nouvelle pile de protocole (IP, TCP-IP,...)
  - ✓Les contraintes industrielles vont aussi pousser à une meilleure modularité et ré-utilisabilité des codes (.Net).
- Ceci va générer une réelle mise en place d'un système d'exploitation pour carte. (www.inspired.org, Java Card 3.0 pour fin 2008).

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Qualité du logiciel

- Par évitement des fautes à la construction,
- Par la qualité du développement,
- et par le Test...

- Quelle est la part du test dans le coût de développement ?
- Étapes :

Conception	développement	Test

- Coût/temps :

Conception	développement	Test

- Temps : peut être le frein au lancement du produit.

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Étapes du développement

- La carte à microprocesseur et les grandes étapes de développement (marché et technique) :

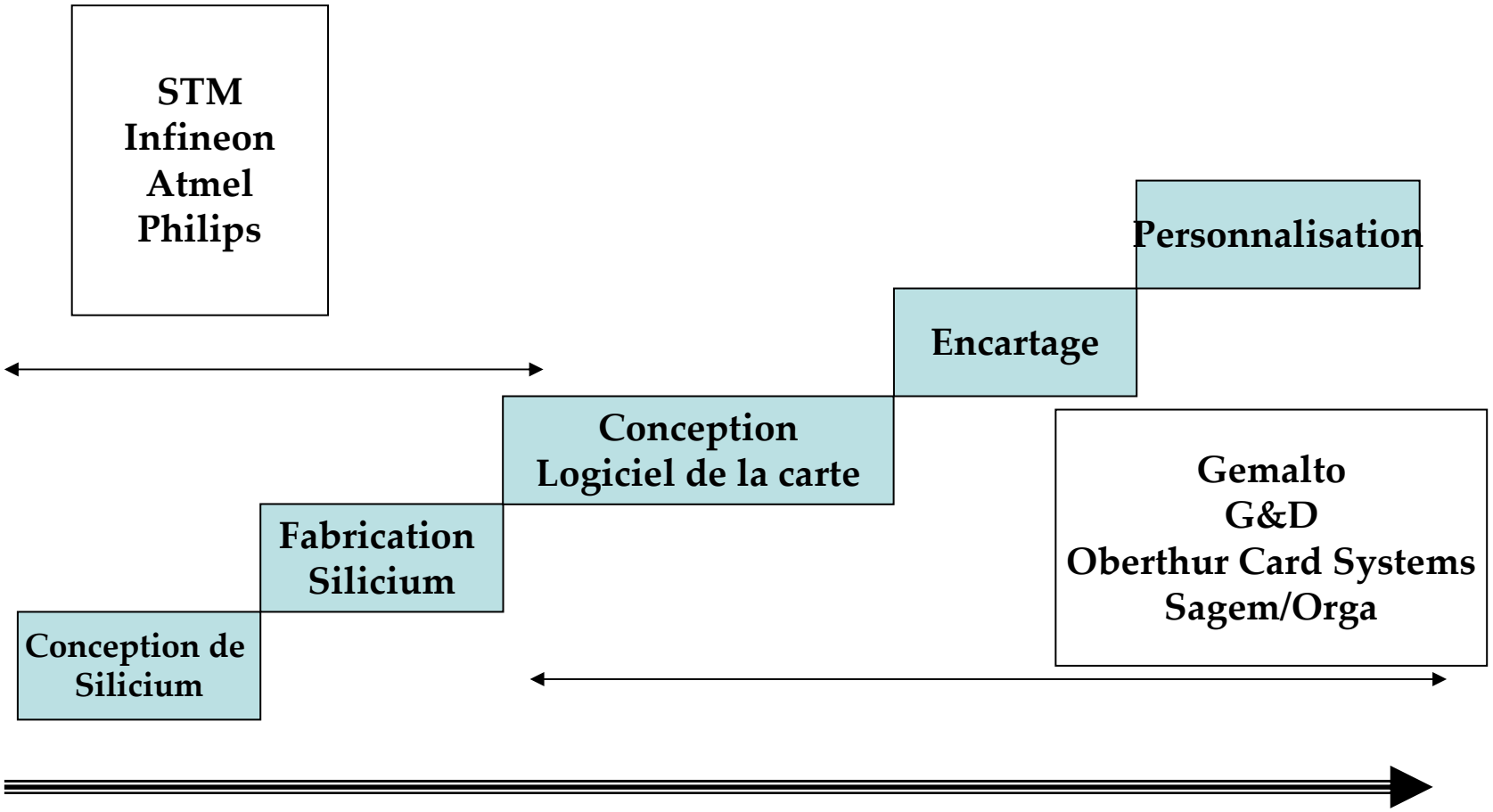
  - ✓Les pionniers (1975-1985) : de(s) idées aux premiers produits, les bases technologiques sont établies,

  - ✓1985-1995 : les marchés et les déploiements importants sont là, la technologie du départ est améliorée et renforcée, des limites apparaissent (besoin de flexibilité qui annonce Java Card)

  - ✓1995-2005 : explosion du marché, qui s'accompagne d'un nouveau paradigme les cartes évolutives

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

De nouveaux acteurs



samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Les volumes d'après http://www.eurosmart.com/index.htm

Eurosmart figures - Cards (Millions of Units- Mu)

Segments	Smart card shipments worldwide 2006		Smart card shipments worldwide forecast 2007 (March 2007)		
	Memory	Microprocessor	Memory	Microprocessor	
					Growth Vs. 2006
Telecom	480	2040	440	2400	18%
Financial services- Retail- Loyalty	30	410	30	490	20%
Government- Healthcare	250[1]	90	350[2]	140	56%
Transport	140	20	160	30	-
Pay TV	-	65	-	70	-
Corporate Security*	15	15	20	20	-
Others	10	15	10	15	-
Total	925	2655	1010	3165	20%
TOTAL	3580		4175		

[1] Including Chinese ID at 200 Mu

[2] Including 300 Mu forecasted for Chinese ID Card

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -

Bibliographie

1. Technology for smart cards: architecture and programmer's guide, Zhiqun Chen, Addison Wesley, sept. 2000
2. Les Cartes à puce: théorie et mise en œuvre, Christian Tavernier, 2ème édition, Ed. Dunod, 2007.
3. Understanding Java Card 2.0, Zhiqun Chen & Rinaldo Di Giorgio
4. http://www.javaworld.com/javaworld/jw-03-1998/jw-03-javadev.html
5. http://javacardforum.org
6. Zhiqun Chen, "**How to write a Java Card applet: A developer's guide**", http://www.javaworld.com/javaworld/jw-07-1999/jw-07-javacard.html.
7. Pierre Paradinas, Support de cours sur « **la Carte à puce** » et « **Java Card** », UE de Systèmes Enfouis et Embarqués, Valeur C, Laboratoire CEDRIC, CNAM. http://deptinfo.cnam.fr/~paradinas/cours/
8. **Global Platform, Card Specification :** http://www.globalplatform.org/specificationform2.asp?id=archived
9. **API Java Card** : http://java.sun.com/products/javacard/htmldoc
10. Eric Vétillard : http://javacard.vetilles.com/2006/09/17/hello-world-smart-card/

Webographie

http://java.sun.com/products/javacard/
http://www.gemalto.com
http://www.oberthur.com
http://www.globalplatform.org
http://www.javacardforum.org
http://www.opencard.org
http://www.linuxnet.com/
http://www.iso.org
http://www.ttfn.net/techno/smartcards/
http://eurekaweb.free.fr/ih1-carte_a_puce.htm
http://membres.lycos.fr/dbon/historique.htm
http://apte.net/info-e/pubs.htm
http://www.eurosmart.com/index.htm

samia.bouzefrane@cnam.fr - CEDRIC (CNAM) -