

Sécurité et vérification d'applications embarquées en environnement Java CardTM

Damien SAUVERON

sauveron@labri.u-bordeaux.fr

<http://dept-info.labri.u-bordeaux.fr/~sauveron>

Plan

Les différentes cartes à puce

La technologie Java Card

Développement logiciel

Perspectives

Objectifs

Objectif principal : mettre en place les bases de travail nécessaires au commencement d'une thèse CIFRE

- ☞ compréhension et maîtrise des spécifications Java Card
- ☞ mise en place de documentations sur la technologie Java Card
⇒ *un rapport interne du LaBRI RR-1259-01*

Autres objectifs :

- ☞ pré-étude des pistes à explorer
- ☞ développement logiciel

Les différentes cartes à puce

Il existe plusieurs sortes de cartes à puce.

Plusieurs classements possibles :

les cartes à mémoire



versus

les cartes à microprocesseur

les cartes à contact



versus

les cartes sans contact

La carte à mémoire

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

Elle possède :

- ➡ *une puce mémoire* de 1 à 4 Ko
- ➡ *une logique câblée non programmable*

Avantages :

- sa technologie simple
- son faible coût (1\$)

Inconvénients :

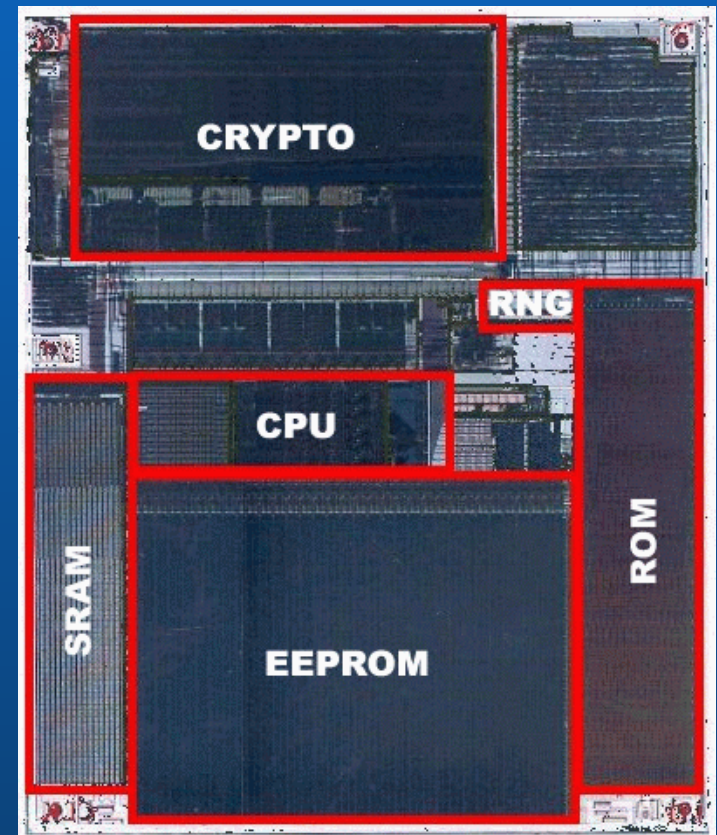
- sa dépendance vis-à-vis du lecteur de carte
- “assez” facile à dupliquer

Les différentes cartes à puce

La carte à microprocesseur

Taille de la puce : $25mm^2$

- ➡ *Microprocesseur* (CPU) : 8, 16 ou 32 bits (à architecture RISC ou pas)
- ➡ *ROM* : 16 à 24 Ko
- ➡ *EEPROM* : 8 à 64 Ko
- ➡ *RAM* : 256 octets à 1Ko
- ➡ Coprocesseur cryptographique
- ➡ Générateur de nombres aléatoires (RNG)



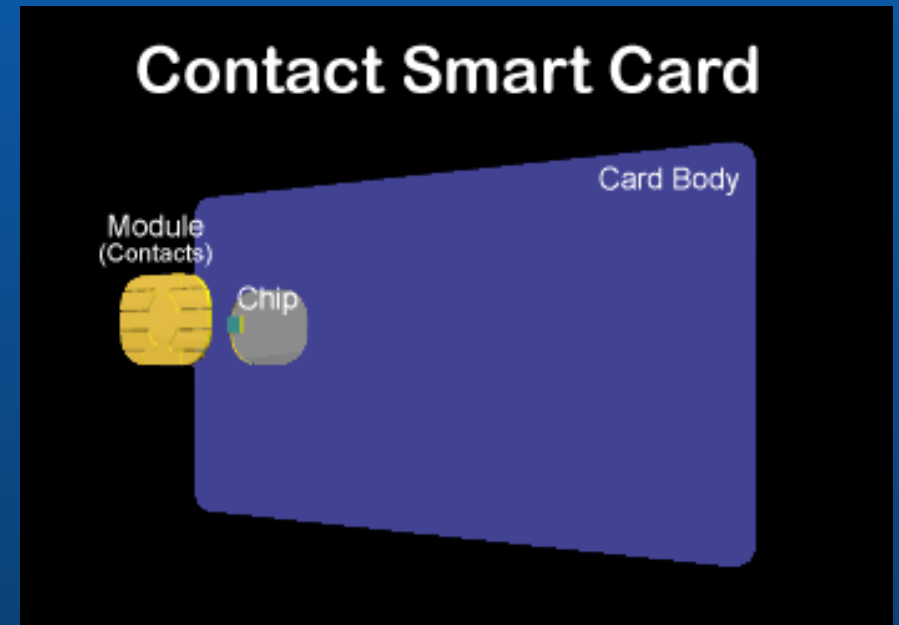
Avantage : le coût acceptable pour tant de sécurité (entre 1\$ et 20\$).

La carte à contact

- ➡ Suit le standard *ISO 7816*
- ➡ *Communication série via huit contacts* $\implies$ insertion dans un lecteur de carte

Problèmes :

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

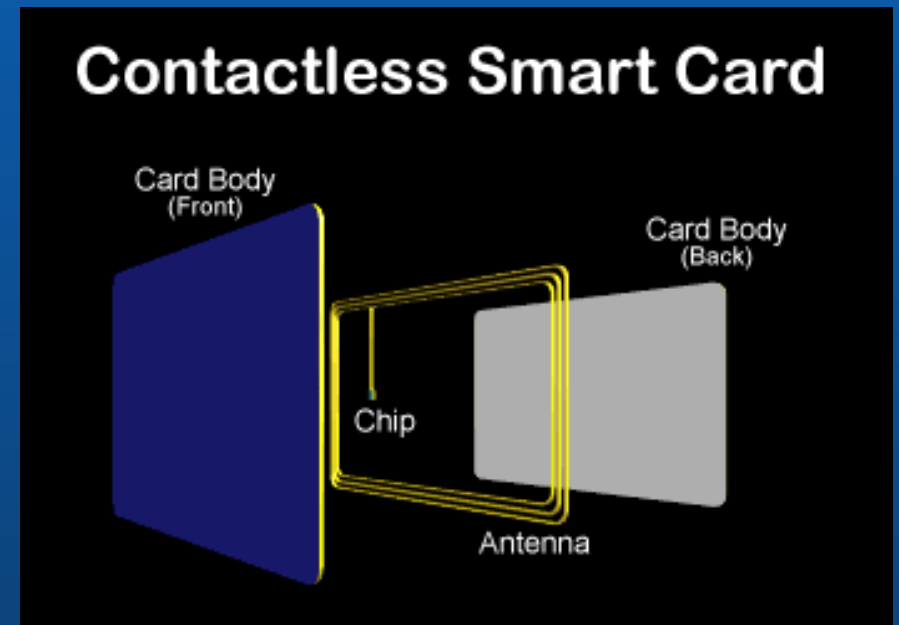


La carte sans contact

- ➡ *Communication via une antenne* dans la carte
- ➡ Récupère son énergie d'un couplage capacitif ou d'un couplage inductif

Problèmes :

- distance de communication limitée (environ 10 cm)
- temps de transaction est de l'ordre de 200 ms $\Rightarrow$ limite les données à échanger
- le coût élevé



La carte combi

C'est une combinaison entre :

- ➡ la carte à contact
- ➡ et la carte sans contact

Ces deux possibilités de communication en font une carte “idéale”.

La technologie Java Card

Technologie permettant de faire fonctionner des applications écrites en langage Java pour :

- ➡ les cartes à puce
- ➡ d'autres périphériques à mémoire limitée

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

Avantages de la technologie Java Card

La facilité de développement des applications grâce :

- ➡ à la **programmation orientée objet** offerte par Java
- ➡ à l'utilisation des **environnements de développement** existants pour Java
- ➡ à une **plateforme ouverte** qui définit des APIs et un environnement d'exécution standard
- ➡ à l'**encapsulation de la complexité** fondamentale du système des cartes à puce

Avantages de la technologie Java Card

La sécurité grâce :

- ➡ à **plusieurs niveaux de contrôle d'accès** aux méthodes et aux variables (public, protected, private)
- ➡ à un **langage fortement typé**
- ➡ à l'**impossibilité de construire des pointeurs**
- ➡ à un **“firewall”**

Avantages de la technologie Java Card

L'indépendance au hardware réalisée grâce au langage Java

⇒ “Write Once, Run Anywhere”

La capacité de stockage et de gestion de multiples applications.

⇒ possibilité de mise à jour des applications de la Java Card sans avoir besoin de changer de cartes

La *compatibilité* avec les standards existants sur les cartes à puce.

Présentation de son architecture

Problème : contraintes mémoires

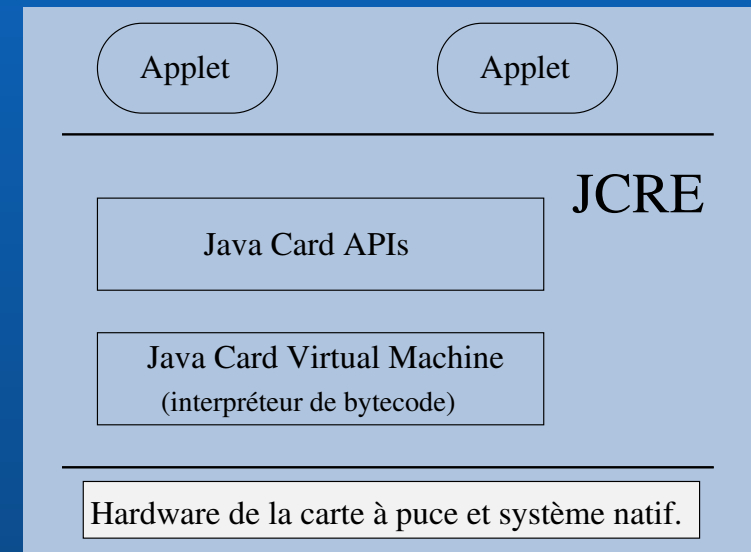
Solutions :

- un sous-ensemble des caractéristiques du langage Java
- découper la machine virtuelle Java en deux parties

Problème : pas de vérificateur embarqué

Solution :

- fournir des mécanismes sécuritaires avec l'environnement d'exécution



Le langage Java Card

Caractéristiques Java non supportées

- ✗ Type simple de donnée de grosse taille : long, double, float
- ✗ Tableau plusieurs dimensions
- ✗ Caractères et chaînes
- ✗ Chargement dynamique des classes
- ✗ Security Manager
- ✗ Ramasse-miettes et finalisation
- ✗ Threads
- ✗ Serialisation d'objet
- ✗ Clonage d'objet

Le langage Java Card

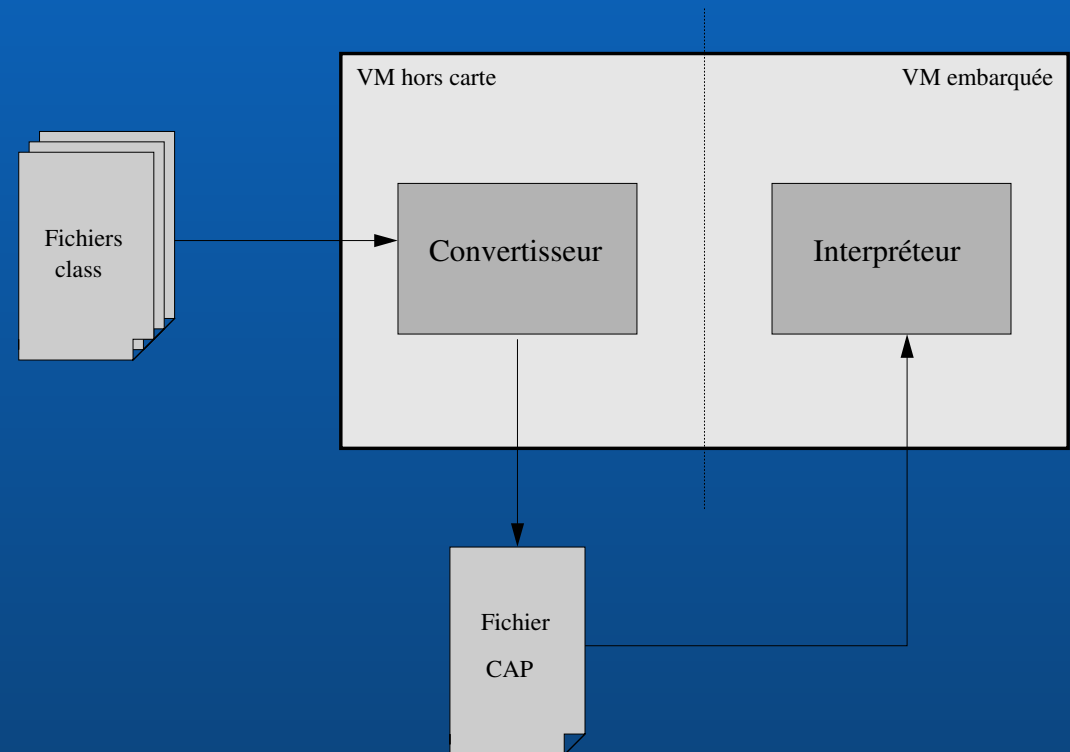
Caractéristiques Java supportées

- ✓ Type simple de donnée de petite taille : boolean, byte, short
- ✓ Tableau à une dimension
- ✓ Paquetage Java, classes, interfaces et exceptions
- ✓ Caractéristique orientée objet : héritage, méthodes virtuelles, surcharge et création dynamique d'objet, contrôle d'accès
- ✓ Le mot clé `int` et le support des entiers sur 32 bits sont optionnels

La machine virtuelle Java Card : JCVM

Les deux parties implémentent toutes les fonctions d'une machine virtuelle.

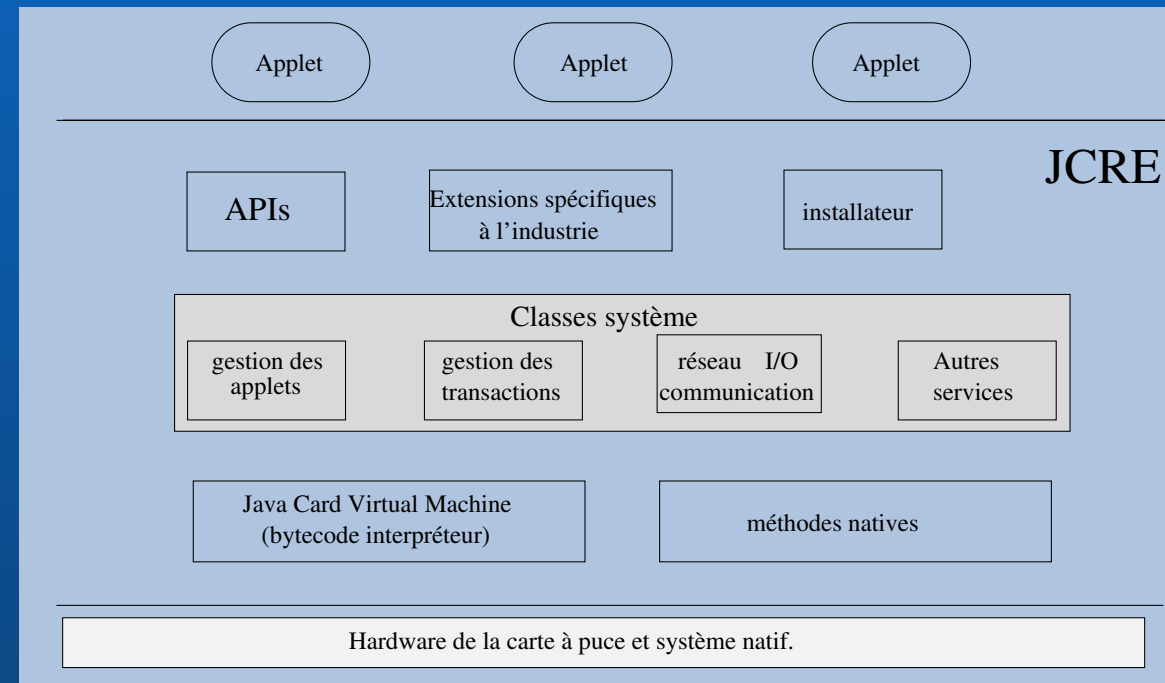
A cause du découpage de la JCVM, *la plateforme est distribuée dans le temps et dans l'espace.*



L'environnement d'exécution Java Card

Responsable :

- ➡ de la gestion des ressources de la carte
- ➡ de la communication réseau
- ➡ de l'exécution des applets
- ➡ du système de la carte
- ➡ de la sécurité des applets



L'environnement d'exécution Java Card

Les caractéristiques du JCRE

- ➡ *Objets persistants* existant au travers des sessions avec le lecteur.
- ➡ *Objets temporaires* dont les données ne persistent pas au travers des sessions avec le lecteur.
- ➡ Chaque opération d'écriture de la JCVm est *atomique*.
- ➡ *Une transaction* est un bloc d'opérations atomiques.
- ➡ Le “*firewall*” isole les applets à l'intérieur de leur espace (contexte).
- ➡ Pour mettre en commun des données, il existe des *mécanismes sécurisés de partage*.

Les APIs Java Card

Ensemble de paquetages optimisés pour la programmation des cartes à puce en accord avec le modèle ISO 7816.

`java.lang` : un sous ensemble strict de son équivalent sur la plateforme Java

`javacard.framework` : classes et interfaces pour le noyau fonctionnel des applets Java Card

`javacard.security` : modèle pour les fonctions cryptographiques supportées sur la plateforme Java Card

`javacardx.crypto` : un paquetage d'extension

Développement logiciel

Réalisation d'un outil de visualisation et de modification d'applications ou de paquetages Java Card.

Une application ou un paquetage Java Card est représenté par un fichier CAP.

⇒ *Éditeur de fichiers CAP.*

☞ Réalisé en Java

☞ Interface utilisant la bibliothèque graphique Swing

Objectifs :

➤ Créer des attaques sur la carte en modifiant le fichier CAP

Fichier CAP

Le fichier CAP est le format standard de fichier pour la compatibilité binaire de la plateforme Java Card.

- ➡ Représentation binaire exécutable des classes d'un paquetage Java Card
- ➡ Fichier JAR qui contient un ensemble de composants

Chaque composant décrit un aspect du contenu d'un fichier CAP :

- ➡ les informations sur les classes
- ➡ les bytecodes exécutables
- ➡ les informations de "linkage"
- ➡ les informations de vérifications
- ➡ etc.

Étiquette	Taille	Données
-----------	--------	---------

TAB. 1 – Format des composants

Présentation de l'interface

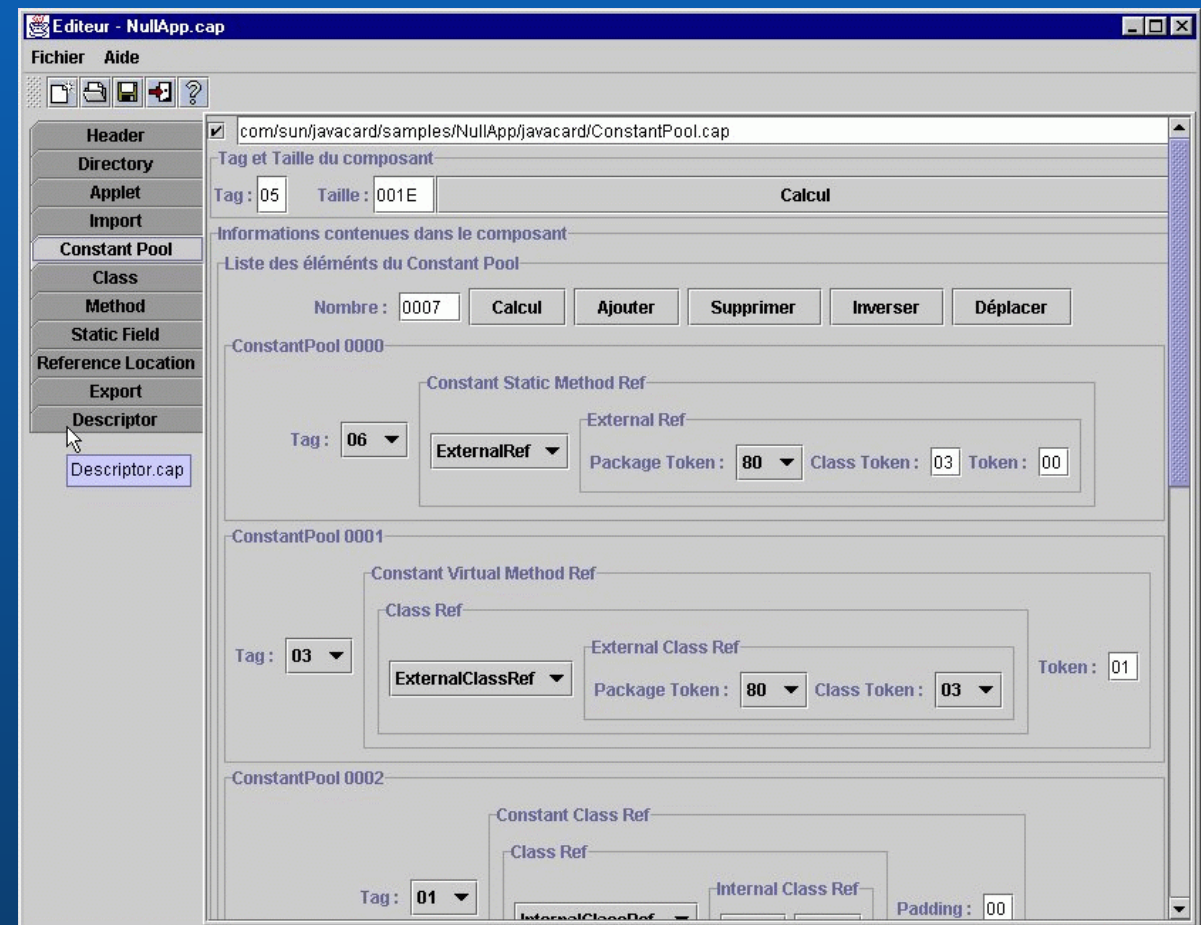
Composant Constant Pool

Avantages :

- Interactif
- Simple

Opérations :

- ☞ Insertion
- ☞ Suppression
- ☞ Inversion
- ☞ Déplacement



Perspectives

Le projet entre le LaBRI et SERMA Technologies

- ➡ Développement d'une machine virtuelle Java Card
- ➡ Développement de logiciels afin de créer des attaques
- ➡ Modélisation des spécifications de la technologie Java Card

La thèse CIFRE

- ➡ Modélisation formelle de la plateforme Java Card :
 - “firewall” entre les applets
 - mécanismes de partage
 - etc.
- ➡ Modéliser des attaques
- ➡ Identifier leurs origines dans les spécifications
- ➡ Proposer des corrections

Conclusion

- ➡ En 30 ans la carte à puce s'est imposée en Europe.
- ➡ *Il reste un marché énorme* (États-Unis, etc.).

La technologie Java Card saura-t-elle le conquérir ?

Pour cela il lui faudra prouver qu'elle est parfaitement *sûre*.