LISV Laboratoire d'ingénierie des systèmes de Versailles
UVSQ université PARIS-SACLAY

Cartes à puce
Plan de cours

Halim Djerroud

révision 1.0

Résumé :

De nos jours, les cartes à puce sont omniprésentes dans nos vies. En 2020 plus de 12 milliards de cartes à puce seront livrées dans le monde, un chiffre en constante augmentation. Il y a donc un travail important de cryptologie à effectuer autour de cette technologie. Ce cours à pour objectif d'introduire la norme ISO 7816 ainsi que la programmation des cartes à puce et les différents mécanismes liés à leur sécurité.

Cours 1 :

Généralités sur le contexte historique de la carte à puce et ce qu'elle représente aujourd'hui. Nous étudierons brièvement l'architecture d'une carte et son cycle de vie, et nous verrons comment mettre en œuvre ce cycle dans ce cours en prenant connaissance des rudiments de la norme ISO 7816. Nous présenterons les outils qui seront utilisés et nous commenterons le premier programme `HelloSmartCard`.

1. Histoire des cartes à puce
2. Description de l'architecture des cartes ATMega Card
3. prise en main des outils
4. programmation et test de la première carte
5. TP :
   — Hello world

Cours 2 :

Se familiariser avec l'environnement de développement et d'apprendre à accéder aux mémoires non volatiles : on va vouloir écrire dans la mémoire programme, et lire et écrire dans la mémoire EEPROM. Mise en oeuvre à travers une application de porte-monnaie électronique.

Cours 3 :

Mise en place des transactions, Lorsque plusieurs opérations sont liées entre elles, on voudrait qu'elles s'effectuent toutes ou bien qu'aucune ne s'effectue.

Cours 4 :

Le but de la séance est d'intégrer le programme en mode console aes-128.c, disponible sur le site du cours, dans la carte à puce et de sécuriser le porte-monnaie électronique.

Cours 5 :

Développer un projet complet de carte à puce industrielle. On va simuler le fonctionnement d'une carte SIM.

Prérequis

— Savoir écrire des programmes en langage C
— Savoir utiliser de ligne de commande (bash)
— Savoir utiliser GCC en ligne de commande
— Savoir utiliser d'un éditeur de texte pour écrire du code source (ex : emacs, vim ou autre)
— Savoir utiliser la commande `man`, et identifier les sections
— Savoir différencier un appel système d'une fonction de la bibliothèque standard
— Savoir lire une documentation en anglais