Atelier Cartes à Puces

M2 SIC Pro
Guillaume Renier, Ghiles Mostafaoui

1 Objectifs de l'Atelier Cartes à Puces

L'objectif principal de l'atelier est de produire un système d'authentification fort (intégré ou non à un produit) multi-utilisateurs et multi-niveaux. La conception se fera à l'aide :

- D'une carte à puce.
- De données biométriques.
- D'algorithmes cryptographiques.
- D'une base de données.

Vous devrez : étudier les failles du système, connaître ses points faibles et décrire les éléments à surveiller lors d'une éventuelle mise en oeuvre. Par exemple lors de l'attaque du système carte bleue par Serge Humpich, c'est la possibilité d'étudier le fonctionnement d'un lecteur de CB qui a permis de casser le système (voir le livre : le cerveau bleu).
Si le système mis en place a une faiblesse de ce type, le vol d'un lecteur de carte compromet le système. Cela ne signifie pas que le système est mauvais. Mais il faut le savoir à l'avance et prévoir des parades.

1.1 Vue d'ensemble

Lors de cet atelier vous devrez :

- Produire l'assemblage du dispositif complet (Cartes à Puces, caméras, liaison avec l'ordinateur).
- Assurer la communication entre l'ordinateur et la carte à puce (lecture, écriture).
- Assurer l'acquisition du flux d'image (gestion de l'acquisition, de l'affichage, et traitements de base sur les images).
- Réaliser un algorithme de détection de formes afin de détecter puis caractériser l'IRIS.
- Déduire l'empreinte biométrique en rapport avec les capacités de la carte à puce.
- Mettre en oeuvre un prototype : préparation de la carte à puce, configuration de l'application.
- Authentifier l'utilisateur.
- Proposer une démonstration : utilisateur sincère, utilisateur qui oubli son mot de passe, tentatives de faudes (avec la carte, avec une copie de la carte, avec les bons identifiants...)
- Cryptanalyser le système d'authentification mis en place en fonction de la compromission des différents éléments mis en jeu : soit étude théorique, soit mise en pratique.
- Proposer des contres-mesures.

1.2 Constitution des équipes.

Nous disposons de 4 lecteurs de cartes et 20 cartes.
Vous vous regrouperez en 4 équipes comprenant au moins un étudiants de chaque parcours.
Les lecteurs de cartes étant reconnus automatiquement par les ordinateurs windows et MacOS il est possible de les déplacer.
Vous êtes libre d'utiliser les langages de programmation de votre choix mais nous recommandons l'utilisation de JAVA6 (les tests ont été effectués sur plateforme MacOS/Windows + JAVA6).

1.3 Déroulement et notation

L'atelier se déroule sur deux journées pleines pendant lesquelles les équipent développent à temps plein leurs dispositifs de stabilisation. Bien que le sujet soit présenté sous la forme séquentielle de deux parties distinctes (partie I et Partie II) il est recommandé aux équipes de veiller à un bon découpage des lots de travail de manière à atteindre leurs objectifs. Un encadrant de la partie I sera présent le premier jour, et un encadrant de la partie II le deuxième jour.

La notation se fera sur la base d'une démonstration de 20mn par équipe le dernier jour, ainsi que d'un rapport de 5 pages maximum (non compris page de couverture, index et sommaire, bibliographie et diagramme/photos/images...) décrivant le dispositif, ses fonctionnalités et l'organisation de l'équipe.

2 Partie I - Guillaume Renier

2.1 Caractéristiques des Cartes à Puces

Les cartes à puce mises à disposition sont des cartes de stockage :

- Avec une sécurité sur 3 niveaux (ISSUER mode, USER mode - PIN1 - PIN2/3)
- Qui ne sont pas protégées contre la copie de données après la saisie des codes PIN.
- Avec un système qui n'est pas protégé contre une attaque de l'homme du milieu.
  Il est inconcevable de baser la sécurité du système d'authentification sur le secret des données stockées sur la carte.
- Qui ne sont capable d'aucun traitement. Les cartes ne sont pas des JAVAcard ni des CRYPTOcard.
- Qui permettent de stocker à peu près 170 octets de données (vous avez bien lu !). Il faut donc prévoir des spécifications précises du produit.
- Utilisant le protocole de communication T=0, les lecteurs sont compatibles PC/SC.

Une base de données sera utilisée pour stocker les autres données qui peuvent être mises en relation avec les données stockées sur la carte.
Par exemple il est possible de stocker les données biométriques dans la base de donnée et une empreinte dans la carte.

2.2 Drivers

Pour les utilisateurs chanceux de MacOSX ou Windows : les drivers PC/SC sont présents en standard dans les installations des OS. Il est néanmoins possible de désinstaller ces driver et donc de les réinstaller (voir site gemalto ci-dessous).

Pour les utilisateurs de LINUX : il faut installer la librairie : libccid. Comme le dit Gemalto : "This USB CCID device is supported by the libccid library.This library provides a PC/SC IFD handler implementation for the USB smart card interface devices compliant to the CCID protocol.Gemalto is actively involved in the development and improvement of this library. This library is packaged and distributed by most of the Linux distributions.TIPS : Use the package manager from your specific Linux distribution to search for the libccid library and install it."[1]. Vous pourrez télécharger le code-source de la librairie : `http://pcsclite.alioth.debian.org/ccid.html`

[1] `http://support.gemalto.com/index.php?id=pc_usb_sl`

2.3 Documentation et liens divers

La documentation peut être téléchargée ici : `http://depinfo.u-cergy.fr/~renier/etudiants/Master/smartcards/gemaltoGemClubMemoDocs/`.

Vous pourrez aussi trouver d'autres informations sur les liens suivants :

- `http://boutique.gemalto.com/is-bin/INTERSHOP.enfinity/WFS/GEMALTO-B2CCORP-Site/en_US/-/EUR/ViewApplication-SwitchParam?LocaleId=fr_FR&currency_choice=EUR`
- `http://www.cryptoshop.com/`
- `http://docs.oracle.com/javase/6/docs/jre/api/security/smartcardio/spec/javax/smartcardio/package-summary.html`

2.4 Cryptographie

Vous devrez authentifier les utilisateurs à l'aide d'une authentification forte.
Pour authentifier un utilisateur on peut utiliser 3 possibilités :

- Ce qu'il connait (mot de passe).
- Ce qu'il détient (carte à puce).
- Ce qu'il est (biométrie).

Une authentification est dîtes forte lorsqu'elle met en oeuvre au moins 2 des 3 possibilités présentées ci-dessus.
En cas d'utilisation d'une carte à puce, on rajoute en général un code PIN.

Vous devrez donc stocker pour chaque utilisateur :

- Des données utilisateur (nom, prénom...) : au moins un identifiant unique.
- Des données d'authentification de type mot de passe, code PIN.
- Des données biométriques.

Vous utiliserez des schémas classiques d'authentification :

- Kerberos.
- CAS.
- Aucun mot de passe ou code PIN ne doit être stocké en clair dans la base de données (ou la carte en ce qui concerne un mot de passe).
  Aucune données biométriques ne devraient être stockées en clair.
- OneTimePassword.
- HMAC.

Et des algorithmes cryptographiques classiques et leurs implémentation dans JAVA :

- RSA (chiffrement, signature), RSA-OAEP.
- Diffie-Helmann
- El-Gamal, DSA.
- 3DES, AES-128/256
- SHA1/2/256/3 ; MD5/MD6
- PBDKF2 (dérivation de clés).

Voire des protocoles classiques comme SSH, SSL, IPsec...

2.5 Description minimale du processus d'authentification.

Dans un ordre quelconque, l'utilisateur :

- Insère sa carte dans le lecteur.
- Saisit son code PIN/mot de passe.
- Se fait photographier par la caméra.

Suite à cela le système authentifie l'utilisateur.
Le système doit être robuste et présenter les caractéristiques suivantes :

- Complétude : un utilisateur honnête est accepté.
- Solidité : un utilisateur fraudeur ne doit pas être accepté.

On peut imaginer la mise en oeuvre des procédés suivants :

- Un mot de passe remplace le code PIN. Le code PIN est donc calculé à partir du mot de passe (c'est là que peut intervenir PBDKF2).
  Attention : un mot de passe peut-être changé.
- Les données biométriques ne sont pas stockées mais servent à authentifier les communications client/serveurs.
- Le système compte le nombre d'utilisation, nombre qui peut-être remis à zéro dans une borne spéciale.
- Le système propose des authentifications différentes en fonction du lieu de connexion (utilisation du code PIN2).
- Les données biométriques peuvent permettre, seules d'identifier l'utilisateur.
  L'identification peut aussi se faire avec la carte à puce seule, avec un nom d'utilisateur, ou en utilisant toutes les données disponibles.

3 Partie II - Ghiles Mostafaoui

L'objectif est ici de vous initier, dans le cadre très appliqué de cet atelier, aux algorithmes de base de traitement d'images utilisés pour la biométrie. Vous pourrez ici non seulement utiliser les connaissances acquises en traitement d'images dans les ateliers précédents (indexation d'images et traitement d'images temps réel) mais aussi aller un peu plus loin sur les aspects reconnaissance de formes temps réel.

3.1 Reconnaissance de formes quelconques

Dans un premier temps il vous sera demandé de programmer une méthode "temps réel" de détection de formes quelconques. Pour cela, vous devrez implémenter une Transformée de Hough Généralisée (qui sera introduite dans la partie cours de cet atelier).

Voici le détail des étapes à réaliser :

- Créez artificiellement 2 images, la première représentera une ellipse noire sur fond blanc (prévoir un rapport 1/2 entre la largeur et la hauteur) et la seconde un cercle noir sur fond blanc
- Calculez pour chacune des 2 images la Look Up Table (LUT, à voir en cours) caractérisant les formes représentés
- Utilisez cette LUT pour détecter des ellipses et des cercles de même taille d'abord dans les 2 images artificielles puis dans des images réelles
- Adapter votre algorithme pour être plus invariant à l'échelle (variation du rayon) et testez

3.2 Détection de visages et Détection des yeux

Testez ce même algorithme sur image (fixe !!) représentant UN SEUL visage, cherchez 1 ellipse et 2 cercles afin de détecter le visage et les 2 yeux (IRIS).
Testez en condition réelle (visages devant votre caméra) et adaptez les rayons à votre application.

3.3 Signature biométrique

Une fois le visage et les yeux détectés, il faut maintenant caractériser la signature biométriques. Etant donné les "PEU" de places disponibles pour le stockage des données sur les cartes à puce, Il va falloir choisir avec parcimonie la caractéristique qui donne la signature biométrique de chacun. On vous propose deux approches simples en vous laissant libre de proposer toute autre alternative :

- Utiliser l'Histogramme couleur de l'IRIS
- Faire une segmentation en région de l'IRIS

Dans tous les cas il faudra penser à comparer (visualiser) les caractéristiques de différents IRIS pour en déduire un critère de sélection pertinent permettant de les identifier.