Rapport

Version	1.0
Date	22 février 2013
Rédigé par	Giovanni HUET
Relu par	Florian GUILBERT, Emmanuel MOCQUET

1 Introduction

De nos jours, les réseaux sociaux ont pris une grande ampleur sur internet et possédent chaque jour de plus en plus d'adhérents. Le principe consiste à créer un profil sur celui ci, à y insérer des données voulant être partagées telles que des photos, des vidéos, des messages,... Sur ce réseau social nous y ajouterons des amis qui seront les personnes pouvant accéder à ces informations. La problèmatique soulevée étant de limiter la diffusion des informations à certaines personnes dans nos amis, mais surtout limiter la diffusion d'information vis à vis du réseau social lui même. En effet, lorsque nous partageons une données, le réseau social posséde cette information qu'elle soit privée ou non. L'idée de ce projet est alors de parait à cette fuite d'information, afin de garantir la confidentialité des données des utilisateurs, le tout renforcer par une authentification forte. Nous nous concentrerons sur le réseau social facebook puisqu'il est le premier sur le marché.

L'objectif du projet était donc de développer une extension pour le logiciel Mozilla Firefox permettant à l'utilisateur de gérer le chiffrement/déchiffrement de ses données sur le réseau social Facebook, les traitements lourd étant confiés à une application Java. Pour ce qui est de l'authentification forte, nous avons utilisé des cartes à puce de type Java Card J3A (marque NXP) avec 40K d'EEPROM via des lecteurs Omnikey 3121. L'intérêt du projet était également d'analyser la sécurité de ces cartes à puce, à savoir la génération de nombres aléatoires, de clefs (sysmétriques et asymétriques), chiffrement, déchiffrement, signature. C'est cette même carte à puce qui contiendra les données sensibles de l'utilisateur (login/mot de passe), clef privée,... Le dialogue avec cette carte à puce se fera par l'intermédiaire d'un client Java qu'on appellera SoftCard.

On distinguera alors ce projet en une composition de deux sous-projets :

- Étude et mise en œuvre de solutions d'authentifications et de signatures par cartes à puce, proposé par Magali Bardet ;
- Solutions cryptographiques pour les réseaux sociaux, proposé par Ayoub Otmani ;

2 SmartCard

Aujourd'hui nous sommes tous menés à utiliser les cartes à puce (carte bleue, carte vitale,...). Elle est utilisé notamment pour de l'authentification forte et contient des informations confidentielles. D'où l'importance de la sécurité concernant les cartes à puce. Dans cette partie de ce projet, on se propose d'étudier les solutions cryptographiques pouvant permettre l'authentification ou la signature, puis de mettre à profit ces caractéristiques à la confidentialité lié à Facebook.

2.1 Génération de nombres aléatoires

La carte à puce nous permet de générer des nombres aléatoires qui nous a été utile dans la création d'IV (Initialization Vector), de mots de passe, de clefs,... Ce qui fait du générateur de nombres aléatoires un point crucial dans la sécurité de l'application. En effet il faut que celui ci ne soit pas prévisible. C'est pour cela que nous avons réalisé une étude statistique sur ce générateur. La librairie javacard met à notre disposition deux algorithmes de générateur.

- ALG_PSEUDO_RANDOM : algorithme pseudo aléatoire.
- ALG_SECURE_RANDOM : algorithme cryptographiquement sûr.

C'est ainsi que pour notre projet nous avons utilisé l'algorithme sécurisé cryptographiquement. Cependant par acquis de conscience, nous avons voulu vérifier le niveau de l'aléatoire du générateur

dit sûr à l'aide d'un outils permettant de réaliser une analyse stastistique dont le compte rendu apparait ci-dessous :

```
************************************************
*                                              *
*                  CR de Yicheng               *
*                                              *
************************************************
```

A noté que le temps moyen pour la génération d'un nombre aléatoire est de 28.1ms.

2.2 Chiffrement/Déchiffrement

Dans l'étude des cartes à puce nous avons également utilisé des méthodes de chiffrement et de déchiffrement sysmétrique et asymétrique. Pour notre cas d'utilisation, nous avons choisi de prendre RSA (1024 bits) comme algorithme de chiffrement asymétrique et AES (128 bits) comme algorithme de chiffrement symétrique car ceux ci sont jugés comme sûr.

- Asymétrique : ALG_RSA_PKCS1
- Sysmétrique : ALG_AES_BLOCK_128_CBC_NOPAD

Nous avons testé ces algorithmes en chiffrant et en déchiffrant, à l'aide de clefs préalablement générées par la carte elle même. Pour ce projet nous utilisons ces méthodes dans différents cas. Nous utilisons le crypto système asymétrique sur la carte seulement pour le cas du déchiffrement à l'aide de la clef privée. En effet pour ce qui est du chiffrement des données concernant facebook, c'est Facecrypt qui s'en chargera à l'aide de notre clef publique fournie au préalable.

Quant au crypto système sysmétrique, nous l'utilisons lors de la communication avec SoftCard via un "tunnel" sécurisé par AES-128. Ce tunnel permet de chiffrer les données transmis par la carte à SoftCard et inversement afin d'éviter la fuite d'informations qui est confidentiel (login, mot de passe, code PIN,...).

2.2.1 Signature/Vérification

Nous avons étudié un autre cas où le crypto système asymétrique peut être utilisé via la carte à puce, qui la signature et la vérification de données. Grâce à la signature nous pouvons authentifier des données, ce qui applique la non répudiation, car la signature se fait via la clef privée de l'utilisateur. Une méthode de vérification existe également, qui permet de vérifier l'authenticité des données après signature d'un utilisateur via la clef publique de ce dernier. Un booléen nous est alors retourné selon la réponse.

2.3 Code PIN/PUK

Une carte à puce étant associée à un utilisateur, il est logique que ce dernier dispose d'un secret pour pouvoir utiliser la carte. Il existe pour cela un code PIN qui est affecté chaque carte et dont seul l'utilisateur principal de la carte en a connaissance. Le code PIN est défini sur 2 octets, ce qui représente $2^1 = 65536$ solutions. Ce qui est faible pour contrer une attaque de type "brute force". Cependant l'attaque est contrée via à un nombre d'essais limité avant le succès de l'authentification. Dans notre cas nous limitons le nombre d'essais à 3 (tout comme les cartes bleue). En cas de blocage

de la carte du à un nombre de tentive trop élevé, seul le code PUK peut débloquer la carte qui est connu seulement de l'administrateur. Tout comme le code PIN, le nombre d'essais pour entrer le code PUK est de 3. Si ce nombre est dépassé la carte devient inutilisable, la réinstallation de l'applet est alors la seule option pour la rendre de nouveau opérationnelle.

3 Terminologie et sigles utilisés

CdR : Cahier de Recettes ;

AdR : Analyse des Risques ;

DAL : Document d'Architecture Logicielle ;

PdD : Plan de développement ;

STB : Spécification Technique de Besoins ;

SC : *SmartCard*, relatif au sous-projet sur les cartes à puce ;

SSN : *Secure Social Network*, relatif au sous-projet sur Facebook ;

FaceCrypt : Application Java gérant les traitements lourds (chiffrement/déchiffrement) de l'extension et étant en relation avec la carte à puce ;

IHM : Interface Homme-Machine, (interface graphique) ;

Utilisateur : entité (humain ou programme) interagissant avec ce sous-projet ;

Système : ce sous-projet ;

Sécurisé : ce terme sous-entend un chiffrement des données et une vérification de leur intégrité ;

SoftCard : application effectuant le relais entre la carte et FaceCrypt ;

Extension : programme incorporé dans le navigateur ;

Aléatoire : les résultats suivent une distribution de probabilité uniforme ;

Pseudo-aléatoire : les résultats sont indistingables en temps polynomial d'une distribution de probabilité uniforme ;

PRNG : (Pseudo Random Number Generator) générateur de nombres pseudo-aléatoires ;

RNG : (Random Number Generator) générateur de nombres aléatoires ;

PIN : (Personal Identification Number) code servant à authentifier l'utilisateur ;

PUK : (Personal Unlock Key) code servant à débloquer la carte quand trop de codes PIN erronés ont été entrés.