Smart Social Network

Projet de Master 2 SSI

Zakaria Addi, Baptiste Dolbeau,
Yicheng Gao, Florian Guilbert,
Giovanni Huet, Emmanuel Mocquet,
Maxence Péchoux, Romain Pignard

Université de Rouen

3 mars 2013

Plan

Plan

Sujets

SmartCard

Étude et mise en œuvre de solutions d'authentification et de signature par cartes à puce. (Mme Bardet)

Secure Social Network

Solutions cryptographiques pour les réseaux sociaux. (M. Otmani)

Sujets

SmartCard

Étude et mise en œuvre de solutions d'authentification et de signature par cartes à puce. (Mme Bardet)

Secure Social Network

Solutions cryptographiques pour les réseaux sociaux. (M. Otmani)

Fusion des projets : Smart Social Network

Développer une solution cryptographique pour un réseau social en utilisant une base cryptographique sûre (carte à puce).

Gestion de projet

Organisation

Deux groupes :

- SmartCard (SC) :
  - 2 client-testeurs (Giovanni Huet et Romain Pignard ;
  - 2 développeurs (Yicheng Gao et Emmanuel Mocquet) ;
- Secure Social Network (SSN) :
  - 1 client-testeurs (Baptiste Dolbeau) ;
  - 2 développeurs (Zakaria Addi et Maxence Péchoux) ;

et un chef de projet (Florian Guilbert) ;

Déroulement

- quatre itérations.
- réunion hebdomadaire avec les clients ;
- documentations (STB, DAL, AdR, CdR, PdD, PdQ) ;

Plan

Introduction

Besoins

- Authentification forte
- Contenir des informations confidentielles

Technologies étudiées

- Génération de nombres aléatoires
- Chiffrement/Déchiffrement
- Signature/Vérification
- Code PIN/PUK
- SoftCard

Présentation

Rappel sur la carte à puce

- Dispose d'un processeur pour du traitement d'informations.
- Permet de stocker des données cachées.
- Assure l'authentification de l'utilisateur.

"Java Card" : qu'est-ce ?

- Désigne la technologie permettant de développer des applets ≪ sécurisées ≫ sur carte à puce.
- Mais c'est aussi une carte à puce :
  - programmable
  - multi-applications
  - interopérable

Abstraction

L'API Java Card

- Permet de s'abstraire de l'assembleur → Java
- Fournit un certain nombre d'objets : PIN, clefs RSA...

Exemple

```
// Nouveau PIN d'une taille de 2 octets,
// avec 3 tentatives.
pin = new OwnerPIN((byte) 3, (byte) 2 );
// Fixation d'un PIN aux octets 15 et 12 (i.e. 3852)
pin.update(new byte[]{15,12}, (short) 0, (byte) 2);
```

Dialogues

Les APDU

- Application Protocol Data Unit.
- Unité de communication entre le lecteur et la carte.

CLA	INS	P1	P2	Lc	Données	Le

Données	Status

TABLE : Structures d'une commande et d'une réponse

Exemple sans l'API Java Card

- Commande : 0xB0 0x00 0x00 0x00 0x01 0x05
- Réponse : 0x02 0xf2 0x23 0x42 0xcf 0x90 0x00

Exemple avec l'API Java Card

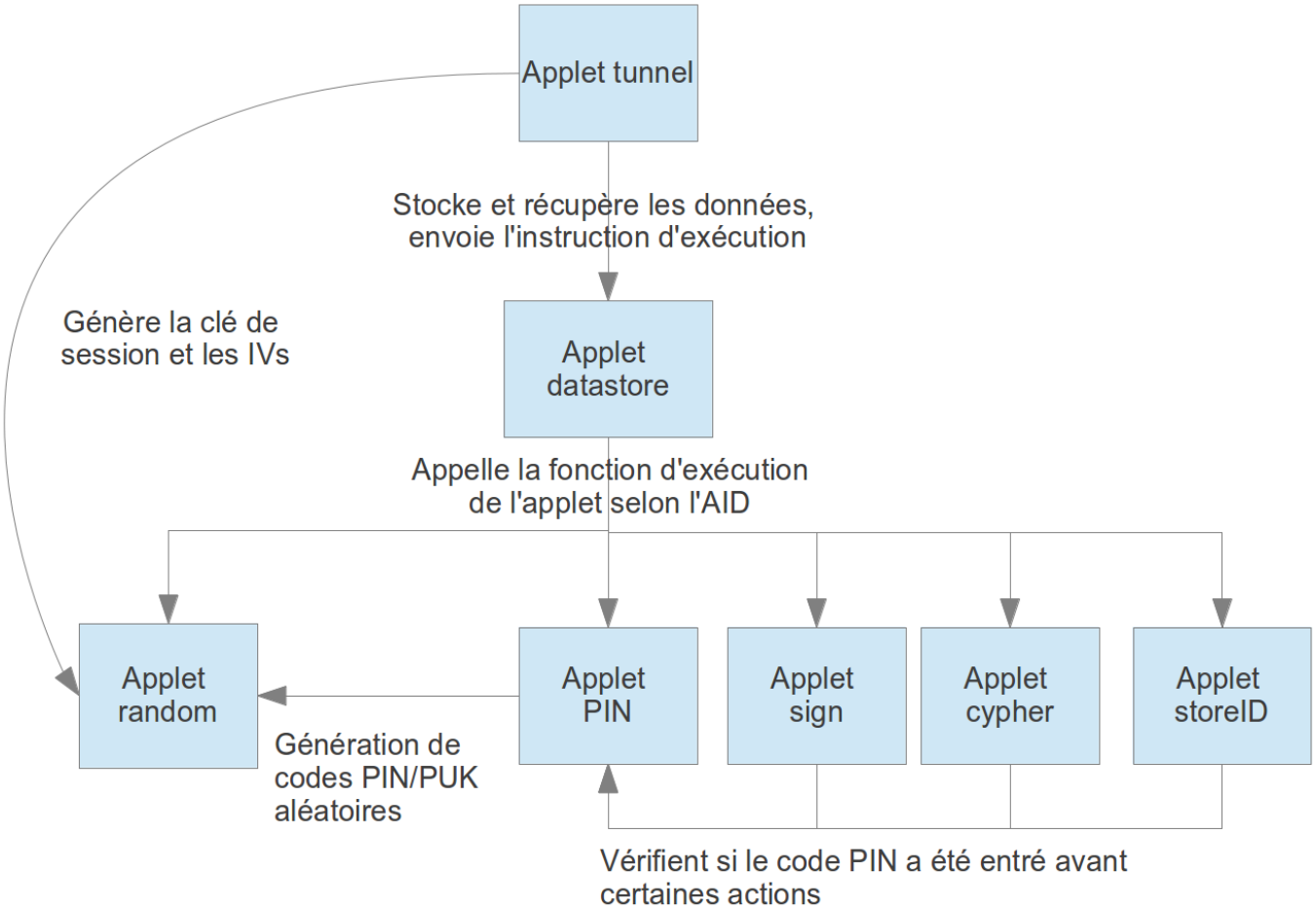
```
// Commande
ResponseAPDU r = channel.transmit(new CommandAPDU(
    (byte)0xB0, (byte) 0x00, (byte) 0x00, (byte) 0x00, 1));
// Partie donnees de la reponse 0x02 0xf2 0x23 0x42 0xcf
r.getData();
```

Principales contraintes

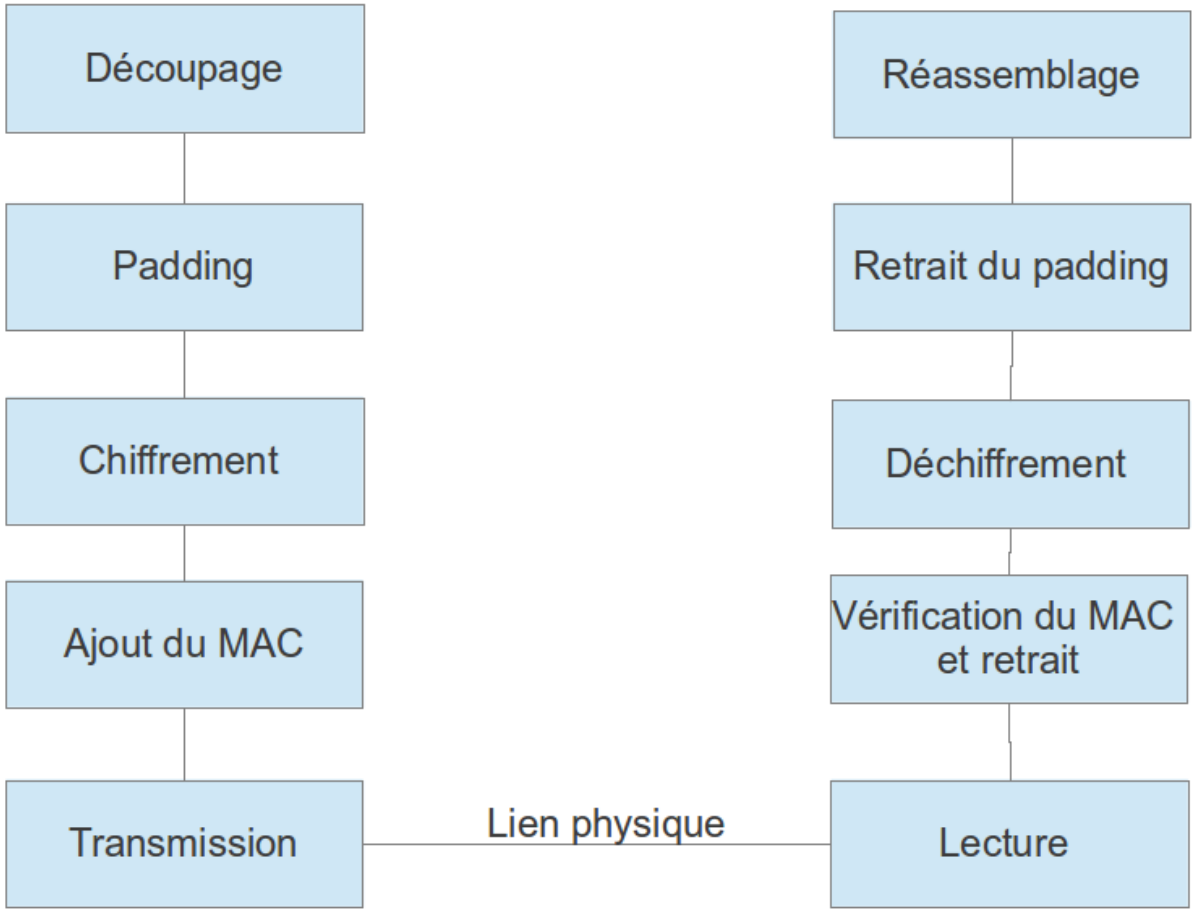
Les limitations de l'API Java Card

- types : boolean, byte, short, tableaux associés
- pas de ≪ garbage collector ≫

Les applications développées



L'aspect sécurité



Démonstration

L'interface entre SSN et la carte

Actuellement

- Applications de chiffrement, déchiffrement, signature, stockage...
- Client testant ces applications.

Mais par rapport à Facebook ?

L'interface entre SSN et la carte

Un serveur vis-à-vis de SSN

- Une entité (SoftCardServer) est instanciée et se met en attente de connexions.
- Pour chaque requête reçue, une action est transmise à une seconde entité : SoftCard.
- SoftCardServer renvoie le résultat de SoftCard au client.

Un client vis-à-vis de la carte

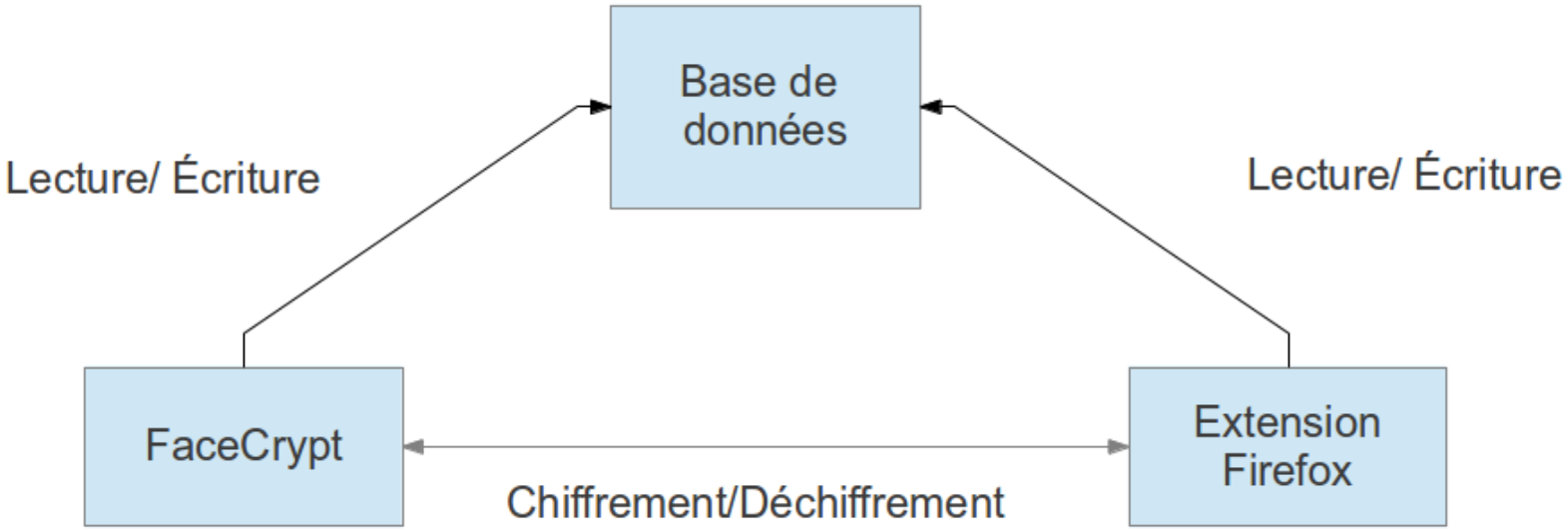
- Une unique instance de SoftCard se connecte au lecteur puis à la carte.
- Différentes méthodes permettent de déchiffrer, signer...
- Pour certaines, sensibles, la carte devra être déverrouillée.

Plan

Les besoins et exigences

Protection des données utilisateur vis-à-vis de tiers
Authentification forte par carte à puce

Présentation des composants



Base de données

Moteur SQLite

Base de données locale

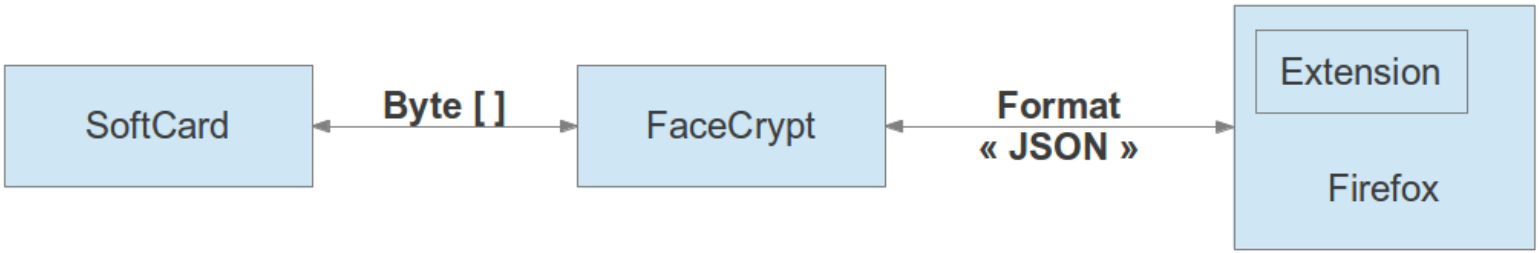
Accessible depuis Java et l'extension

Stockage des liens d'amitié dans la base

Listes d'amis

Clés publiques

La communication



Composition

Six classes java

- ASymCypher
- SymCypher
- ServerSSL
- Client
- Dataprocess
- CacheManager

Exemple de cycle

- Received from Facecrypt : {"action" :"getID"}
- Sent to Softcard : 47
- Received from Softcard : $\underbrace{666f6f2e6261722e333334343931 33}_{\text{login}}\ 20\ \underbrace{726f6f74726f6f74}_{\text{password}}$
- Sent to Facecrypt : {"action" :"getID" ,"login" :"foo.bar.3344913","firstConnection" :false, "pass" :"rootroot"}

Les extensions Firefox

Javascript

Langage de script orienté objet

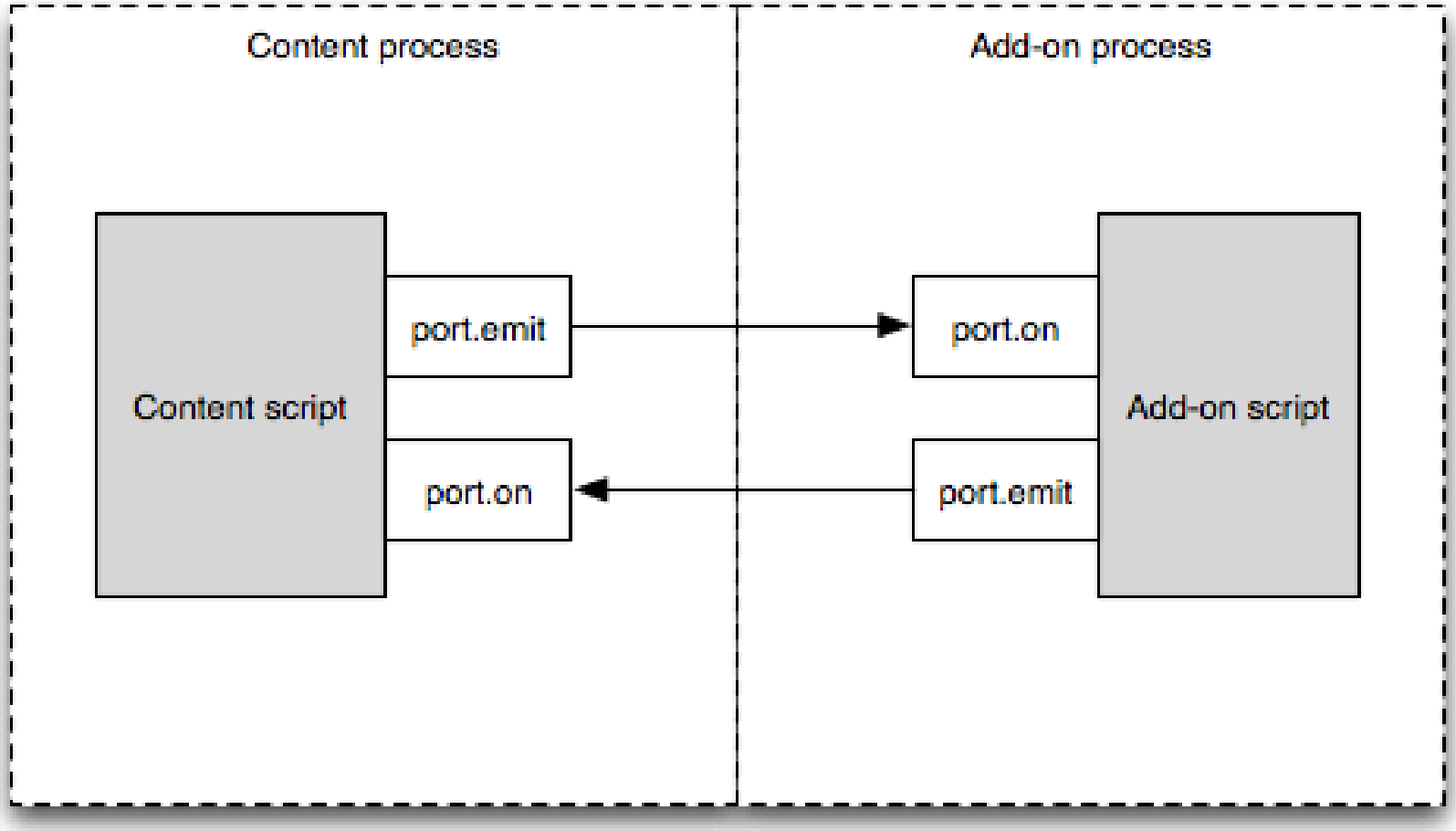
Add-on SDK Mozilla

Environnement de développement pour les extensions Firefox

Add-on Builder

Plateforme en ligne d'utilisation du SDK, permet le *versionning*

Schéma de fonctionnement



Modification du DOM

Apparence fidèle

Conservation du *Look and Feel*

Ajouts d'éléments

- Boutons de chiffrement, déchiffrement, gestion des clefs...
- Listes d'amis, liens de modifications...

Communication sécurisée

Tunnel SSL

Communication via sockets, utilisation PKCS#12 de Firefox, utilisation librairies NSS de Mozilla

Gestion des évènements

Plusieurs types d'évènements au niveau des librairies

Interaction avec la Base de Données

Manipulation des fichiers

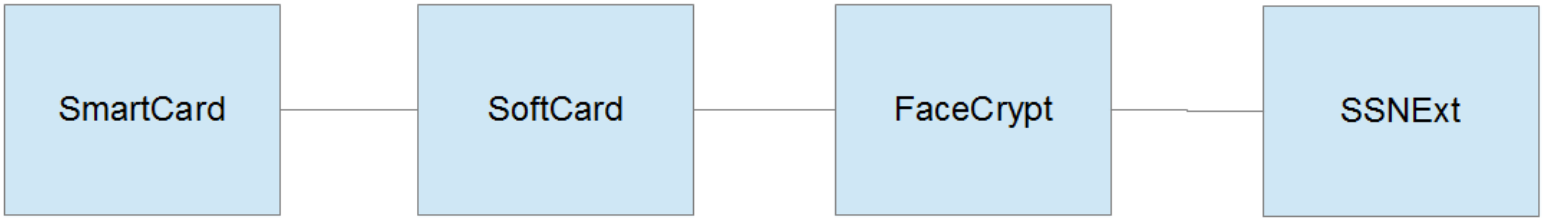
Création d'une instance de la Base de Données à la reception du pseudo, module FileUtils

Cas d'utilisations

Synchronisation clefs publiques, ajout/modification/suppression de listes d'amis

Plan

Démonstration



Plan

Conclusion

Difficultés rencontrées

- SmartCard :
  - taille des données ;
  - communications sécurisées entre la carte à puce et l'application cliente ;
  - installations des lecteurs ;
  - stockage « caché » ;
  - algorithmes implantés sur la carte ;
- Secure Social Network :
  - manipulation de la page Facebook ;
  - communications sécurisées entre SSNExt et FaceCrypt ;
  - fonctionnement d'une extension.

Conclusion

Améliorations possibles

- SmartCard :
  - IHM pour entrer le code PIN ;
  - gestion de l'arrachage de la carte ;
  - One Time Password ;
  - prendre en compte les attaques (canaux cachés) ;
- Secure Social Network :
  - finalisation pour mise en production ;
  - abstraction du réseau social pour l'extension ;
  - étudier le tatouage d'images.

Conclusion

Ce que cela nous a apporté

- SmartCard :
  - manipulation d'une carte à puce ;
- Secure Social Network :
  - gestion d'une communication sécurisée entre plusieurs composants ;
- utilisation concrète de la cryptographie ;
- travail en équipe.

Merci pour votre attention.

Des questions ?