ARCHITECTURE DU LOGICIEL

Version :	1
Date :	14/02/13
Rédigé par :	Emmanuel Mocquet
Relu par :	
Approuvé par :	

MISES A JOUR

Version	Date	Modifications réalisées
0.1	26/11/12	Création
0.2	02/01/13	Complétion
1	14/02/13	Corrections vis à vis de l'avancement du projet et de la STB

1. Objet :

Ce document a pour but de décrire les solutions techniques conçues pour répondre aux exigences définies dans la spécification technique de besoin. Il doit identifier et définir les différents modules ou constituants du logiciel ainsi que leur interfaces de telle sorte que chacun d'entre eux puisse être développé de façon autonome par un ou plusieurs membre de l'équipe, avant d'être intégré.

Ce projet se découpe en deux parties distinctes. La première consiste à développer une ou plusieurs applications établissant la communication entre ce support et un PC. Il est attendu qu'au final, il puisse y avoir chiffrement, authentification et/ou stockage de données.

La seconde partie consiste à permettre à un utilisateur de contrôler les données qu'il dépose sur « Facebook ». Les détails consacrés à cette partie sont disponibles dans le document associé.

Nous décrirons ici la partie architecturale du projet « SmartCards ».

2. Documents applicables et de référence

- Le sujet de la première partie projet et sa description sont disponibles dans le document « sujet-cartes-a-puce.pdf » ;
- Le document de spécification technique de besoin s'intitule « Smartcards STB.pdf » ;

Le sujet propose plusieurs documents de référence :

- *Cartes à puce. Administration et utilisation,* LINUX Magazine Hors-Série no 39. Diamond Editions, Nov./Déc. 2008 ;
- *Cartes à puce. Découvrez leurs fonctionnalités et leurs limites*, MISC Hors-Série no 2. Diamond Editions, Nov./Déc. 2008.

3. Terminologie et sigles utilisés

- SDK: un Software Development Kit est un ensemble d'outils, de documents et des exemples permettant ou facilitant le développement d'applications.
- API : une Application programming Interface fournit un certain nombre de structures (fonctions, objets...) et/ou spécifie la façon dont doit être exécutée une certaine action.
- APDU : « un Application Protocol Unit est un message échangé entre une carte à puce et un lecteur de carte à puce. Il est normalisé et décrit dans l'ISO 7816 partie 4 ». (Wikipedia)

4. Architecture physique du matériel utilisé

Par la nature du projet, des cartes à puces et leurs lecteurs sont nécessaires. Ces cartes seront des Java Card avec 40K de mémoire EEPROM et les lecteurs seront des Omnikey 3121.

Afin d'établir un dialogue entre le lecteur et le système, nous utiliserons la bibliothèque PC/SC-Lite.

Le développement d'applications se fera sur des machines de type GNU/Linux. Pour cette raison, le SDK JavaCard utilisé sera au maximum en version 2.2, la dernière disponible (3) étant uniquement compatible avec Windows. Aussi, pour des raisons de compatibilité, le SDK Java deva être en version 1.5.

Nous utiliserons ainsi l'API fournie par ce SDK et le plugin d'Eclipse JCDE pour automatiser certaines étapes durant la conception des applications.

Leur gestion se fera grâce à l'outil GlobalPlatform, suivant lui-même le standard globalplatform.

5. Architecture des logiciels

5.1. Structure logique

Concernant le fonctionnement final, c'est-à-dire l'intégration avec l'application pour Facebook (qu'on appelera ici «SSN»), les composants seront les suivants :

5.2. Description du constituant «SSN»

SSN se servira de la carte à puce pourobtenir les identifiants de l'utilisateur, obtenir sa clef publique, etc. La description complète de ce composant est disponible dans le document associé à la partie Facebook.

5.3. Description du constituant « SoftCard »

Le composant SoftCard fera office de passerelle entre le SSN en traduisant les requêtes émises par celui-ici et en les transmettant à la carte. Afin de s'assurer que la communication entre SSN et lui-même soit sécurisée et que ces deux entités s'authentifient mutuellement, il possèdera deux certificats : l'un identifiant SSN, l'autre lui-même.

Il devra être suffisamment générique ou modulable pour pouvoir être utilisé sans le composant SSN, c'est-à-dire dans une situation d'authentification par exemple.

Il sera écrit en Java et possèdera un secret partagé avec la carte qui permettra d'ouvrir un tunnel sécurisé (garantissant authentification, confidentialité et intégrité) avec la carte, cette opération étant indispensable avant toute autre communication.

Enfin, le nombre d'actions possibles à gérer étant important, sa complexité sera grande.

5.4. Description du constituant «SmartCard»

La carte à puce contiendra différentes informations de l'utilisateur de SSN. Seront ainsi stockés, sa biclef – la partie publique étant transmise à l'utilisateur –, son identifiant Facebook ainsi que son mot de passe. Afin que l'utilisateur s'authentifie auprès de la carte, il devra ainsi posséder un code PIN et PUK, stockés sur la partie cachée de la carte.

Le programme installé sur la carte et permettant la manipulation de ces données sera écrit en Java et installé sur la carte grâce au SDK JavaCard 2.2.

Enfin, étant donné les opérations possibles (entrée de code PIN, envoi de clef publique, génération de nombre aléatoire, déchiffrement...), sa complexité sera grande.

5.5. Justifications techniques

Java est actuellement le langage dans lequel la documentation est la plus complète, en ce qui concerne le développement d'applications pour Javacard.

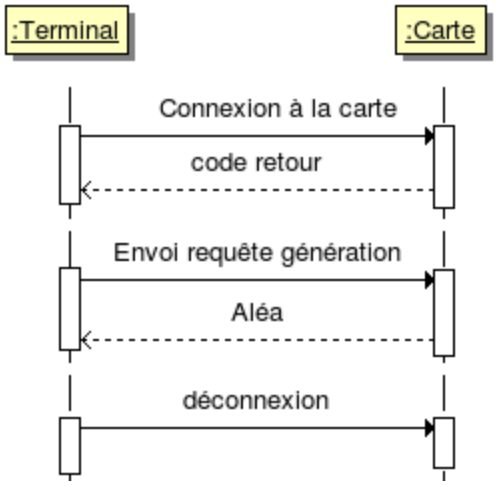
Ensuite, le fait d'utiliser deux certificats (l'un pour SSN, l'autre pour Softcard) permettra de garantir une communication chiffrée, authentifiée et intègre entre ces deux entités. Un dialogue pourra ensuite se faire avec SSN, via le canal sécurisé.

6. Fonctionnement dynamique

6.1. Génération d'un nombre aléatoire

La génération d'un nombre aléatoire sera simple, puisqu'il s'agira de « sélectionner » l'applet via l'envoi d'un APDU. A la réception de celle-ci, une méthode sera appelée (« process ») qui se chargera de générer puis renvoyer un nombre aléatoire.

Dans le cas où l'utilisateur souhaite obtenir un nouveau mot de passe, cette méthode sera appelée par SoftCard. Il sera ainsi possible de fournir la taille désirée du mot de passe.

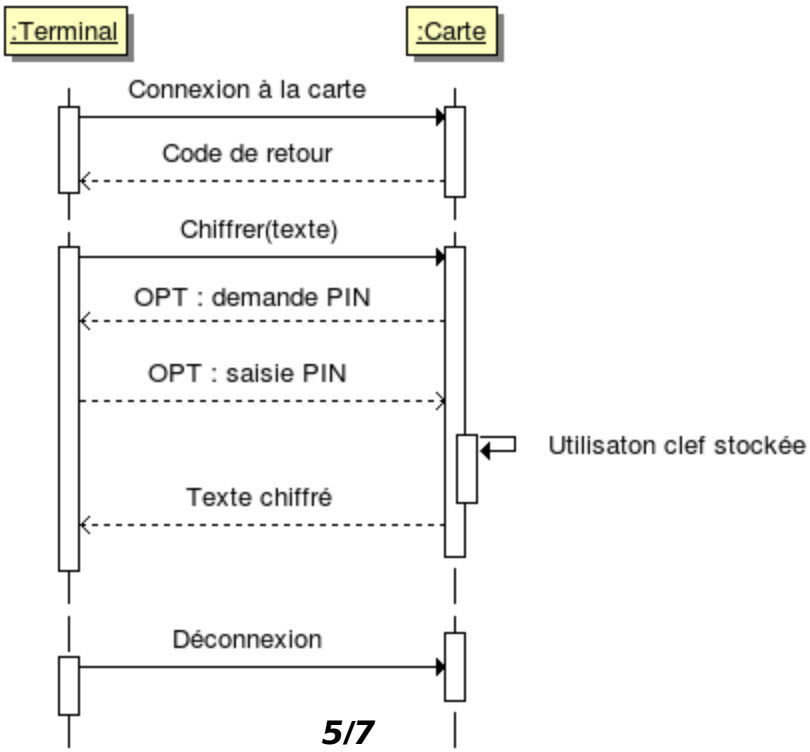


6.2. Déchiffrement de données

Le schéma suivant représente le scénario durant lequel aucune erreur n'est produite. Sont mis en jeu le terminal (en l'occurence un PC) et la carte à puce.

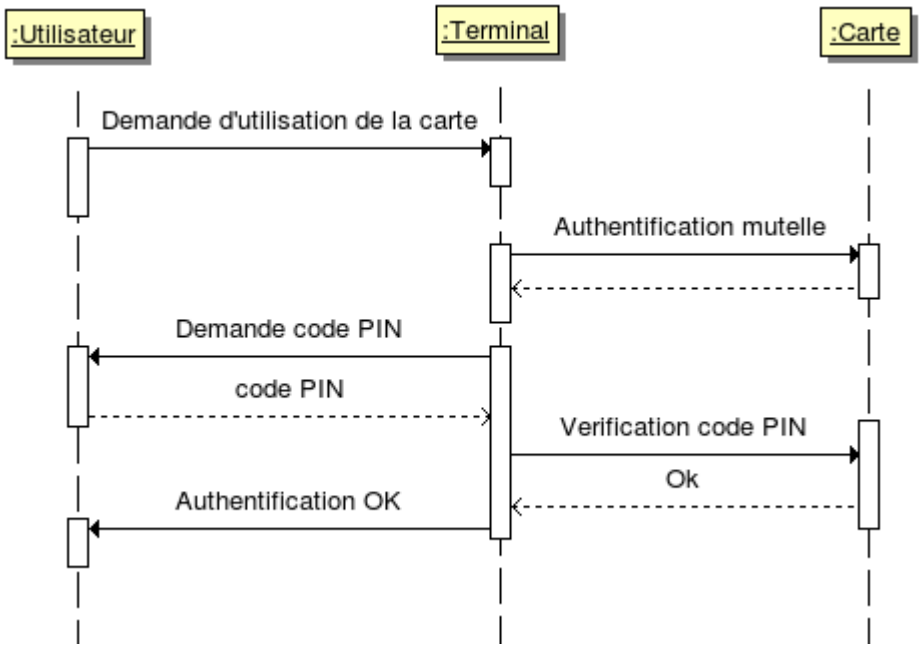
En cas de signature, le protocole sera sensiblement le même.

« OPT » signifie que l'utilisateur devra se ré-authentifier si la carte s'est bloquée.



6.3. Authentification de l'utilisateur

Lorsque l'utilisateur insère la carte et qu'il cherche à s'y connecter, le scénario suivant s'applique, en supposant qu'aucune erreur ne survient.



6.4. Modification du mot de passe de l'utilisateur

Dès lors que l'utilisateur désirera changer de mot de passe, et comme nous l'avons précisé dans la partie 6.1, une série d'octets aléatoires sera générée par la carte, puis renvoyée en base 64 au module appelant (SoftCard ou SSN). Ce nouveau mot de passe sera stocké temporairement et l'ancien conservé.

SSN pourra alors modifier le mot de passe sur Facebook. Si ceci s'effectue correctement, FaceCrypt de SSN enverra une validation à SoftCard, indiquant ainsi que le mot de passe pourra remplacer l'ancien.

(schéma à venir)

6.5. Récupération d'identifiants

SSN et son extension auront besoin d'authentifier l'utilisateur auprès de Facebook. Pour ce faire, il leur faudra envoyer une requête de récupération d'identifiants auprès de SoftCard. Celui-ci la transmettra à SmartCard qui les renverra, si l'utilisateur est authentifié (i.e. si la carte est débloquée).

(schéma à venir)

7. Traçabilité

Exigences		F-FO-10	F-FO-20	F-FI-10	F-FI-20	F-FI-30	F-FQ-10	F-FQ-20	F-FQ-30	F-FQ-40	F-FQ-50	F-FR-10
Composants	*SSN*	X				X	X				X	
	SoftCard	X		X	X	X	X	X	X		X	X
	SmartCard	X	X	X			X	X	X	X	X	X