Cahier de recettes

Version	1.0
Date	1er mars 2013
Rédigé par	Giovanni Huet
Relu par	Baptiste Dolbeau

MISES À JOUR

Version	Date	Modifications réalisées
0.1	05/02/2013	Création
1.0	24/02/2013	Vérification

Table des matières

1 Introduction

Ce document est un support pour la validation du logiciel lors de la recette auprès du client. Il est consacré à la définition des moyens et des procédures utilisés pour assurer la recette du produit développé. La recette est un procédé permettant d'assurer la conformité du logiciel à la spécification déjà définie. Nous allons recenser dans ce document les objectifs de tests de validation et les moyens nécessaires pour les atteindre en précisant :

- Les pré-conditions à satisfaire ;
- Les moyens matériels requis (plate-forme de tests) ;
- La logique de leur déroulement (étapes successives).

Notre logiciel peut être divisé en une liste de constituants qui seront testés à tour de rôle. L'ensemble des opérations devra être transparent vis à vis de l'utilisateur. Les différents cas d'utilisation prélevés de la spécification technique de besoin sont les suivants :

Génération de nombres aléatoires

La carte à puce doit pouvoir générer des nombres aléatoires de manière sécurisée : c'est à dire non prévisible.

Déblocage de la carte (via authentification par code PIN ou via PUK)

Afin d'utiliser la carte, il nous faut nous authentifier à l'aide d'un code PIN. Si l'utilisateur échoue à l'authentification par code PIN suite à un certain nombre de tentatives, la carte sera verrouillée. Pour la déverrouiller nous utiliserons le code PUK.

Transmission de données

La carte doit pouvoir transmettre des données stockées à SoftCard telles que le login, le mot de passe, la clef publique, etc...

Chiffrement/Déchiffrement de données

Sur la carte sont stockées la clef publique et la clef privée préalablement générées (Crypto-système assymétrique de type RSA). Ces clefs nous permettront de chiffrer et de déchiffrer des données reçues ou à envoyer.

Signature/Vérification de données

Par le biais de la carte, nous serons en mesure de signer des données avec notre clef privée afin d'assurer la non répudiation. Nous devons également pouvoir vérifier l'auteur des données. Nous utiliserons pour cela la clef publique.

Administration des cartes

On devra également pouvoir administrer les cartes : effectuer des opérations telles que la réinitialisation du code PIN, son attribution, etc...

2 Documents applicables et de références

- SC_STB : Le document renfermant les spécifications techniques de Besoin ;
- SC_DaL : Le document contenant l'architecture logicielle ;
- Les comptes rendu de réunion du projet ;
- Le sujet du projet : "cartes-a-puce.pdf".

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 Besoin ;

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

SSN : *Secure Social Network*, relatif au sous-projet sur les réseaux sociaux ;

FaceCrypt : Application Java gérant les traitements lourds (chiffrement/déchiffrement) de l'extension et relais entre l'Extension et *SmartCard* ;

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

Utilisateur : Entité (humain ou programme) intéragissant avec ce sous-projet ;

Système : Ce sous-projet ;

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

Extension : Programme incorporé dans le navigateur ;

Aléatoire : Indistinguable en temps polynomial, distribution de probabilité uniforme ;

PRNG : *Pseudo Random Number Generator* - Générateur de nombres pseudo-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.

4 Environnement de tests

L'ensemble des tests se sont effectués sur des machines ayant les caractéristiques suivantes :
- Système d'exploitation : Ubuntu 12.04 ;
- Processeur : Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz ;
- Mémoire : 2Go RAM ;
- Logiciel : Eclipse Platform Version : 3.8.0, Java 1.6 (Client) et Java 1.5 (Java card).

Nous utilisons également des cartes Java Card J3A (marque NXP) avec 40K d'EEPROM et des lecteurs Omnikey 3121. Les cartes sont conformes aux standards Java Card 2.2.2 et Global Platform 2.1.1.

5 Responsabilité

Afin de mener les tests dans les meilleurs conditions, une organisation au sein du groupe a été mise en place :

La conception et la définition des données de tests a été réalisée par Giovanni Huet et Romain Pignard. Après avoir exécuté les différents tests, les responsables de ce module transmettront aux développeurs un compte rendu contenant les résultats afin d'améliorer la version actuelle du logiciel et d'en fournir une nouvelle à tester. Chaque version fournie doit être testée et validée.

6 Stratégie de tests

La démarche utilisée pour effectuer les tests est la suivante :
- Mettre à la disposition de l'équipe testeur les modules développés.
- Réalisation des tests à travers une procédure, celle ci comportera un jeu de tests ainsi que la modalité de leur exécution.
- Élaboration d'un compte rendu des résultats des tests qui sera transmis aux développeurs.
- Correction des anomalies par l'équipe développeur.
- Des tests secondaires seront effectués pour s'assurer que toutes les anomalies ont été corrigées.

Les tests seront réalisés par ordre de priorité. Les modules ayant une priorité indispensable seront pris en compte dès que possible. La condition d'arrêt des tests sera le succès de ces derniers après correction des anomalies.

7 Gestion des anomalies

A chaque modification apportée (correction), nous devrons réaliser un nombre de tests permettant de détecter les anomalies persistantes. Toute anomalie détectée sera notée dans un rapport et ce dernier sera envoyé aux développeurs afin qu'ils apportent les modifications nécessaires.

8 Procédures de tests

Pour chaque cas d'utilisation, nous décrivons une procédure de test détaillée. Chaque procédure dispose d'un jeu de tests basé sur des données réelles.

Objet testé : Génération de nombres aléatoires			Version : 1.0	
Objectif de test : Vérifier le comportement du générateur aléatoire de la carte à puce.				
Procédure n°1 : Générer un nombre aléatoire.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Générer un nombre aléatoire à l'aide des fonctions javacard disponibles.	Obtention d'un nombre aléatoire.	F-Gl-10	OK

Procédure n°2 : Evaluer le niveau de l'aléatoire.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Générer plusieurs (millions) nombres aléatoires afin d'établir des stastitiques et vérifier le niveau de l'aléatoire.	Générateur non prévisible (Probabilité uniforme)	F-Gl-10	NC

Procédure n°3 : Evaluer le temps d'exécution.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Chronométrer la génération de plusieurs (milliers) nombres aléatoires pour en évaluer la moyenne.	Nous souhaitons que le temps de génération d'un nombre aléatoire soit < 300ms pour que cela soit invisible à l'utilisateur.	F-Gl-10	OK

Objet testé : Déblocage de la carte (via authentification par code PIN et PUK)			Version : 1.0	
Objectif de test : Vérifier le comportement de la carte lors de plusieurs tentatives d'authentifications.				
Procédure n°4 : Déverrouillage de la carte (via code PIN).				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Entrer le code PIN valide.	Authentification de l'utilisateur rendant la carte utilisable.	F-Gl-20	OK

Procédure n°5 : Verrouillage de la carte.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Effectuer trois authentifications erronées à la suite.	Verrouillage de la carte.	F-Gl-20	OK

Procédure n°6 : Déverrouillage de la carte (via code PUK).				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Entrer le code PUK valide.	La carte est de nouveau opérationnelle.	F-Gl-20	OK

Objet testé : Transmission de données			Version : 1.0	
Objectif de test : Vérifier si les données contenues dans la carte peuvent être transmises.				
Procédure n°7 : Transmettre des données.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Envoyer des données de la carte à SoftCard.	Réception intégrale des données par SoftCard.	F-Gl-30	OK

Objet testé : Communication sécurisée lecteur/carte (Tunnel)			Version : 1.0	
Objectif de test : Vérifier les fonctions de chiffrement, d'authentification et d'intégrité.				
Procédure n°8 : Envoi de données chiffrées.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Envoyer des données chiffrées à l'aide d'un chiffrement symétrique (AES-128)	Données non compréhensibles sans la clef de déchiffrement.	F-FI-30	OK

Procédure n°9 : Envoi de données authentifiées sans altération.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Envoyer des données authentifiées à l'aide de CBC-MAC sans altérer le contenu.	Validation de la non modification des données reçues.	F-FI-30	OK

Procédure n°10 : Envoi de données authentifiées avec altération.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Envoyer des données authentifiées à l'aide de CBC-MAC en altérant leur contenu.	Détection de la modification des données reçues.	F-FI-30	OK

Procédure n°11 : Déchiffrement de données reçues.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Déchiffrement des données issues d'un cryptosystème symétrique à l'aide de la clef secrète partagée.	Récupération en clair des données transitant dans le tunnel.	F-FI-30	OK

Objet testé : Déchiffrement de données issues d'un cryptosystème asymétrique (RSA)			Version : 1.0	
Objectif de test : Vérifier si les données reçues peuvent être déchiffrées.				
Procédure n°12 : Déchiffrer avec la clef privée.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Déchiffrer des données à partir de la carte avec la clef privée.	Données déchiffrées.	F-Gl-40	OK

Procédure n°13 : Déchiffrer avec une clef invalide.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Déchiffrer des données à partir de la carte avec une clef privée invalide.	Données non déchiffrées et une erreur est détectée.	F-Gl-40	OK

Objet testé : Signature/Vérification de données.			Version : 1.0	
Objectif de test : Signer des données et vérifier la signature.				
Procédure n°14 : Signer des données.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Signer des données à partir de la carte avec la clef privée.	Données signées.	F-Gl-50	OK

Procédure n°15 : Vérification avec la clef publique associée.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Vérifier des données signées à partir de la carte avec la clef publique.	Données vérifiées.	F-Gl-50	OK

Procédure n°16 : Vérification des données invalides.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Vérifier des données signées d'une autre personne à partir de notre carte avec notre clef publique.	Données non vérifiées.	F-Gl-50	OK

Objet testé : Administration des cartes.			Version : 1.0	
Objectif de test : Vérifier si l'administration des cartes est possible.				
Procédure n°17 : Attribuer un code PIN.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Attribution d'un code PIN à un utilisateur.	L'utilisateur possède un code PIN qui lui est propre.	F-Gl-60	OK

Procédure n°18 : Insertion de données sur la carte.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Stocker des données qui soient propres à l'utilisateur : login, mot de passe, etc...	La carte contient bien les données.	F-Gl-60	OK

Procédure n°19 : Modification des données sur la carte.				
N°	Action(s)	Résultats attendus	Exigence	OK/NOK
1	Modification des données préalablement insérées : login, mot de passe, code PIN, etc...	Les données sont bien modifiées.	F-Gl-60	OK

9 Couverture de tests

Ce tableau reprend les exigences de la STB et précise, pour chacune d'entre elles, la méthode de vérification (démonstration / tests) et une description de celle ci.

Exigence	Méthode de vérification	Procédure utilisée	Commentaire
F-Gl-10	Démonstration	Procédure 1	Ce test consiste à utiliser une fonction disponible via la librairie javacard pour générer un nombre aléatoire.
F-Gl-10	Test	Procédure 2	Le test consiste à générer plusieurs nombres aléatoires (millions) et les soumettre à un test stastistique pour évaluer le niveau de l'aléatoire.
F-Gl-10	Test	Procédure 3	Le test consiste à chronométrer la génération de plusieurs nombres aléatoires (milliers) et d'en faire la moyenne afin de connaitre le temps d'exécution moyen pour la génération d'un nombre aléatoire, que l'on considérera valide s'il est inférieur à 300 ms.
F-Gl-20	Test	Procédure 4	Le test consiste à débloquer la carte en s'authentifiant auprès de celle ci via le code PIN.
F-Gl-20	Test	Procédure 5	Le test consiste à entrer 3 codes PIN erronés afin de verrouiller la carte, et d'entrer ensuite le bon code PIN pour vérifier le verrouillage.
F-Gl-20	Test	Procédure 6	Le test consiste à déverrouiller la carte après avoir entré 3 codes PIN erronés en entrant le code PUK valide.
F-Gl-30	Test	Procédure 7	Le test consiste à envoyer des données contenues sur la carte.
F-FI-30	Test	Procédure 8	Le test consiste à envoyer des données chiffrées via une communication sécurisée par un chiffrement symétrique.
F-FI-30	Test	Procédure 9	Le test consiste à envoyer des données authentifiées sans altérer leur contenu. Nous devons ensuite vérifier que le contenu est bien intègre.
F-FI-30	Test	Procédure 10	Le test consiste à envoyer des données authentifiées en altérant leur contenu. Nous devons ensuite pouvoir détecter la modification du contenu.
F-FI-30	Test	Procédure 11	Le test consiste à recevoir des données au préalable chiffrées, et vérifier si elles ont bien été déchiffrées.
F-Gl-40	Test	Procédure 12	Le test consiste à déchiffrer des données avec la clef privée, nous devrions alors obtenir les données en clair.
F-Gl-40	Test	Procédure 13	Le test consiste à déchiffrer des données avec une clef non valide, nous devrions alors obtenir une erreur.
F-Gl-50	Test	Procédure 14	Le test consiste à signer des données à partir de la clef privée stockée sur la carte.
F-Gl-50	Test	Procédure 15	Le test consiste à vérifier avec la clef publique des données au préalable signées. Les données doivent alors être vérifiées.
F-Gl-50	Test	Procédure 16	Le test consiste à vérifier avec une clef publique non correspondante des données au préalable signées. Les données ne doivent alors pas être vérifiées.
F-Gl-60	Démonstration	Procédure 17	Dans cette procédure, nous affectons à un utilisateur, donc à la carte lui appartenant, un code PIN afin de pouvoir débloquer la carte et l'utiliser.
F-Gl-60	Démonstration	Procédure 18	Dans cette procédure, nous insérons des données dans une carte telles qu'un mot de passe, un login, etc...
F-Gl-60	Test	Procédure 19	Ce test consiste à modifier les données précédemment insérées sur la carte.