ARCHITECTURE DU LOGICIEL

Version:	1.0
Date:	01/03/2013
Rédigé par:	Zakaria Addi, Maxence Péchoux
Relu par:	Florian Guilbert

MISES A JOUR

Version	Date	Modifications réalisées
0.1	3/1/2013	Création et complétion
0.2	04/01/13	Ajout des schémas
1.0	27/02/13	Complétion des cas d'utilisation et corrections

1. Objet :

L'objectif de ce document est de présenter les moyens techniques mis en œuvre pour assurer les fonctionnalités spécifiées dans la STB.

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». C'est cette partie qui va être détaillée ici.

2. Documents applicables et de référence

Ce document s'appuie sur le sujet de la deuxième partie du projet : proxy-encryption.pdf ;
et sur la Spécification Technique des Besoins de cette même partie : SSN_STB.pdf

3. Terminologie et sigles utilisés

Addon SDK : Suite d'outils facilitant le développement d'extensions Firefox.
DOM : Document Object Model, décrit la structure d'un document HTML et permet un accès programmatique de celui-ci.
XPCOM : Composant logiciel utilisé par la fondation Mozilla pour l'ensemble de logiciels.
Javascript: Langage de programmation utilisé par la quasi totalité des navigateurs internet.
JAVA : Langage de programmation orienté objet. Les programmes sont exécutés sur une machine virtuelle, ce qui permet une grande portabilité du programme.
Jquery : Librairie rendant plus aisée la manipulation du DOM.
Navigateur: Logiciel permettant de consulter le Web.
Look and Feel : Ensemble des caractéristiques d'une interface utilisateur.
JSON : Javascript Object notation, format de données utilisé par le langage Javascript.

4. Architecture physique du matériel utilisé

Le projet pourra fonctionner sur un système d'exploitation qui supporte le navigateur Firefox ainsi que la machine virtuelle java. Au niveau des performances requises, seuls les opérations de chiffrement peuvent nécessiter des capacités de calcul relativement importantes.

Pour résumer le projet aura besoin de :
– Mozilla Firefox
– Java Virtual Machine (JVM)

5. Architecture statique du logiciel

5.1. Structure logique

Le système sera constitué de deux entités principales, une extension Firefox et une Application Java appelée Facecrypt. L'extension en constituera l'interface homme machine tandis que Facecrypt gérera les traitements cryptographiques. On utilisera aussi une base de données SQLite pour stocker les listes d'amis.

5.2. Constituants

SSNext

- Rôle: Interaction avec l'utilisateur.

- Propriétés et attributs de caractérisation: Elle sera simple d'utilisation et sera intégrée à Facebook. Le Look and Feel de Facebook sera maintenu.

– Services offerts: Elle assurera une utilisation transparente de Facebook, tout en gérant l'émission et la réception de messages chiffrés. Il permettra aussi à l'utilisateur d'envoyer à l 'application java toutes les informations nécessaires à son bon fonctionnement. La création et l'édition de listes d'amis sera aussi gérée via l'extension. Un service d'identification automatique sera aussi proposé.

- Dépendances : Toutes les opérations de chiffrement/déchiffrement seront laissées à l'application java.

- Langage de programmation : Javascript

- Procédé de développement : Extreme Programming

- Taille et complexité : petite taille, complexité importante

Facecrypt

- Rôle : Passerelle entre la carte à puce et l'extension.

- Propriétés et attributs de caractérisation : L'application une fois lancée devra être invisible pour l'utilisateur, sauf pour un paramétrage.

- Services offerts : Chiffrement, déchiffrement de contenu. Stockage du comportement à adopter pour les communications avec des amis.

- Dépendances : Le contenu à chiffrer/déchiffrer proviendra de l'extension Javascript. Certaines opérations de chiffrement/déchiffrement seront effectuées par la carte à puce.

- Langage de Programmation : Java

- Procédé : Extreme Programmming

- Taille et complexité : Taille moyenne et complexité moyenne

BDD Sqlite :

- Rôle : Stockage des listes d'amis

- Propriétés et attributs de caractérisation : la base de donnée sqlite sera un simple fichier accessible par SSNext et Facecrypt.

- Services offerts : Sauvegarde des listes d'amis et des clefs publiques.

- Langage de Programmation : la base est interrogée via des commandes SQL classiques

- Dépendances : les listes d'amis seront paramétrés via SSNext

- Taille et complexité : taille simple et complexité simple

5.3. Justifications techniques

Extension Firefox : Nous avons effectué ce choix pour assurer une meilleure expérience à l'utilisateur. On pourra ainsi, en manipulant directement le DOM de la page Facebook, préserver une utilisation normale de Facebook.

Traitement effectué sur Facebook : Bien qu'il commence à exister des librairies de chiffrement Javascript, les opérations de chiffrement vont être partagés entre Facecrypt et la carte à puce. De plus, le besoin d'une application java est évident du fait de l'utilisation d'un lecteur de carte à puce. Dans le futur, avec l'amélioration des machines virtuelles Javascript, l'amélioration d'outils tels qu'Emscripten qui permettent de traduire du code d'un langage X vers Javascript, et les futures possibilités offertes pour la communication avec des périphériques externe pour un navigateur, on peut supposer que l'ensemble de l'application pourra uniquement fonctionner du côté client.

6. Fonctionnement dynamique

Premier cas d'utilisation C1 :

Chiffrement d'un statut :

L'utilisateur publie un statut. Trois cas se présentent :

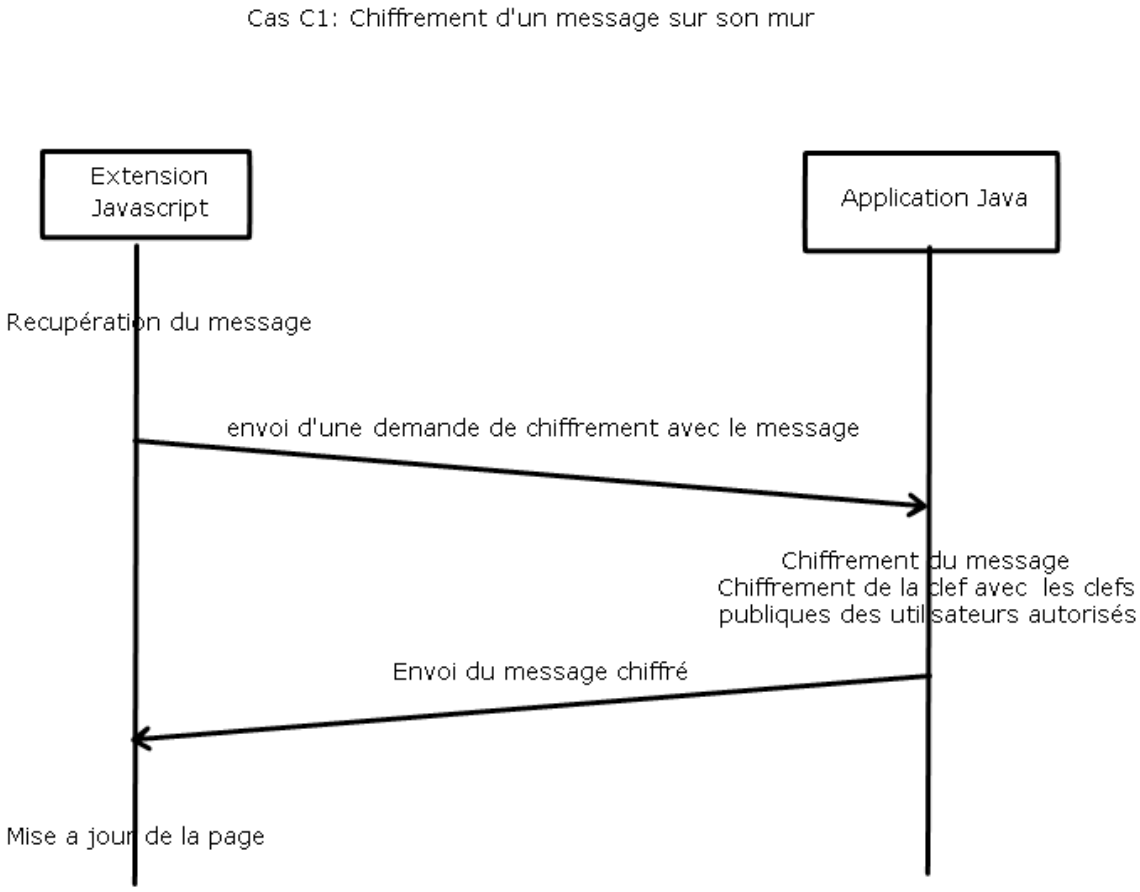
- L'utilisateur souhaite envoyer un message non chiffré. On laisse Facebook suivre son exécution normale.

- L'utilisateur souhaite envoyer un message chiffré anonyme. Tout d'abord, Notre extension récupère le texte entré par l'utilisateur ainsi que la liste des destinataires choisis puis l'envoie via un objet XPCOM à Facecrypt. Ensuite, Facecrypt chiffre le message reçu et le renvoie à notre extension ainsi qu'une liste constituée de la clé de session chiffrée par chacune des clés publiques des utilisateurs. Enfin, notre extension met à jour le champs d'entrée de la page facebook.

- L'utilisateur souhaite envoyer un message chiffré non-anonyme. Tout d'abord, Notre extension récupère le texte entré par l'utilisateur ainsi que la liste des destinataires choisis puis l'envoie via un objet XPCOM à Facecrypt. Ensuite, Facecrypt chiffre le message reçu et le renvoie à notre extension ainsi qu'une liste constituée de la clé de session chiffrée par chacune des clés publiques des utilisateur. A chacune des entrées on associera le nom du possesseur de la clé publique. Enfin, notre extension met à jour le champs d'entrée de la page facebook.

Lors de la publication d'un message on le préfixera d'une balise « SSNEncryptedA » pour

un chiffré anonyme ou « SSNEncryptedN » dans le cas contraire , ce qui permettra aux destinataires de traiter les bons messages.

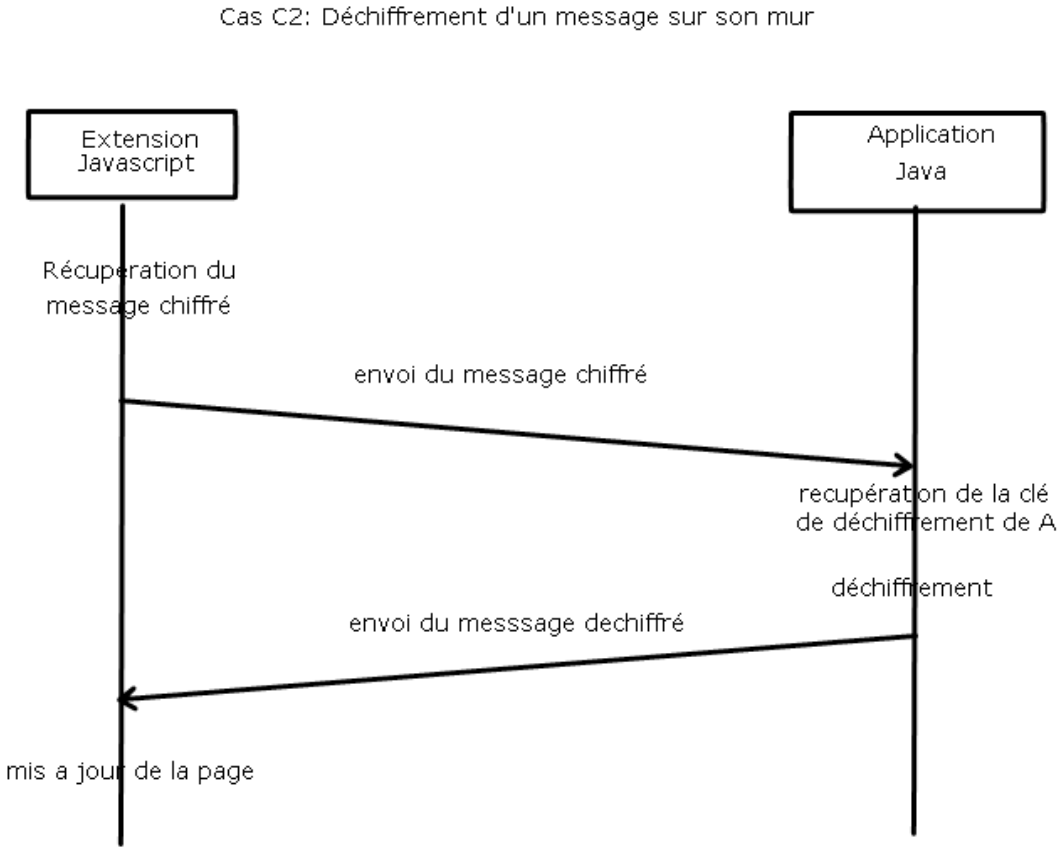


Deuxième cas d'utilisation C2 :

Déchiffrement d'un statut :

L'utilisateur a reçu un message chiffré de A sur son mur. L'extension Javascript récupère le message chiffré ainsi que la liste des chiffrés de la clé de message par les clefs publiques des destinataires et les envoie à l'application java. Celle-ci utilise alors la clef privée de l'utilisateur pour en déchiffrer la clef symétrique établit par A, puis déchiffre le message en lui-même grâce à cette dernière. Le message déchiffré est renvoyé à l'extension qui modifie le contenu du post. Si le message est anonyme, l'extension se contente de rajouter un bouton de déchiffrement afin de limiter les traitements effectués par la carte. L'envoi ci-dessus ne se fera alors qu'au clic du bouton, tandis que dans le cas d'un post non-anonyme, celui-ci se fera dès le chargement du post.

Les commentaires associés au statut seront envoyés avec le message chiffré pour être déchiffré.



Troisième cas d'utilisation C3 :

Chiffrement d'un document :

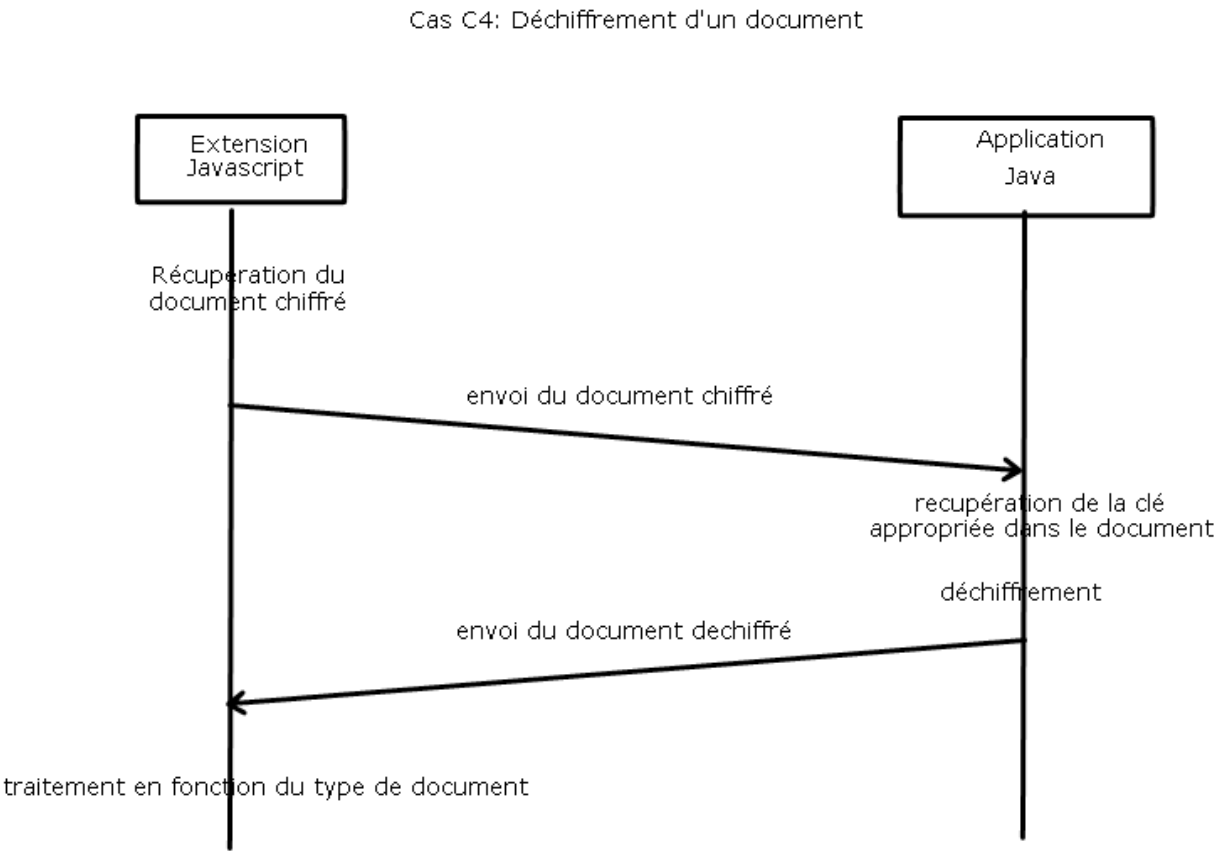
De façon analogue au post de statut, lorsqu'un document est « uploader », l'extension l'envoie à la machine Java qui se charge de le chiffrer en accord avec la liste des utilisateurs autorisés par l'auteur. Cependant, pour qu'il soit accepté par Facebook, le document doit être une image. Ainsi, une fois celui-ci chiffré, une en-tête d'image va être ajouté afin de faire croire à Facebook qu'il s'agit bien d'une image. Le document résultant est renvoyé à l'extension qui l'intègre à la page.

Quatrième cas d'utilisation C4 :

Déchiffrement d'un document :

L'extension détecte un document chiffré sur la page. Elle envoie alors celui-ci à l'application Java qui le traite de façon similaire aux statuts, si ce n'est qu'ici le document déchiffré n'est pas traité de la même façon si c'est une image ou non. Dans le cas d'une image, on l'ouvrira

via Facebook, sinon il sera sauvegardé. Cette fonctionnalité n'a pu être implémentée car Facebook effectue systématiquement des traitements sur les documents « uploadé ». Facebook utilisant un algorithme de compression à perte, les opérations de déchiffrement à perte sont impossibles.



Cinquième cas d'utilisation C5 :

Création d'une liste d'ami :

L'utilisateur clique sur un bouton préalablement inséré sur la page Facebook. Une fenêtre s'affiche et permet à l'utilisateur de créer une liste. Elle sera ensuite sauvegardée dans la BDD SQLite.

Sixième cas d'utilisation C6 :

Suppression d'une liste d'amis :

L'utilisateur clique sur un bouton préalablement inséré sur la page Facebook. Une fenêtre s'affiche et permet à l'utilisateur de supprimer une liste. Elle sera aussi supprimée de la BDD SQLite.

Septième cas d'utilisation C7 :

Ajout d'un ami :

L'utilisateur clique sur un bouton préalablement inséré sur la page Facebook. Une fenêtre s'affiche, l'utilisateur coche les cases des amis à ajouter dans la liste. La liste concernée sera mise à jour dans la BDD.

Huitième cas d'utilisation C8 :

Chiffrement des commentaires de statut :

On rajoute un bouton « Chiffrer » au formulaire d'entrée d'un nouveau commentaire. Les commentaires seront chiffrés avec la même clé que celle du statut associé.

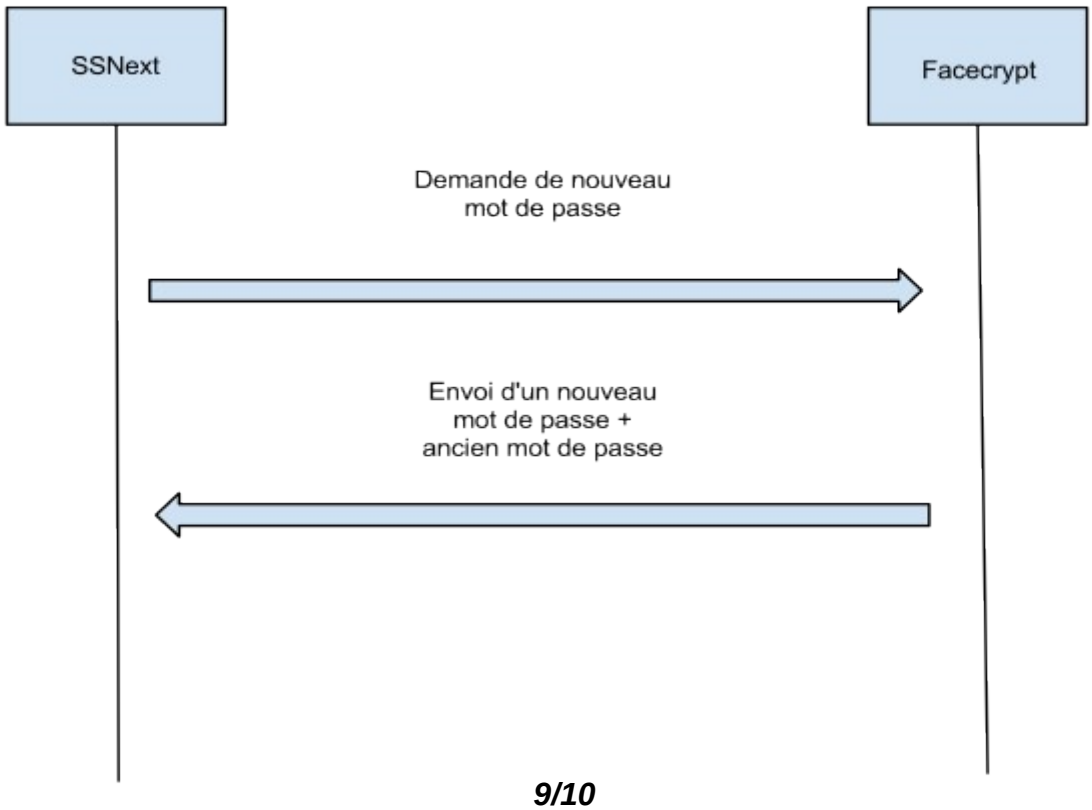
Neuvième cas d'utilisation C9 (STB-SmartCard C11):

Modification du mot de passe :

L'utilisateur suit la procédure normale pour la modification d'un mot de passe Facebook. Arrivé sur le formulaire de modification, l'extension demandera à Facecrypt de générer un mot de passe et de le lui renvoyer ainsi que l'ancien mot de passe. Les champs du formulaire seront automatiquement modifiés. L'utilisateur pourra ensuite :

– Accepter la modification du mot de passe, auquel cas un message sera renvoyé à Facecrypt qui mettra à jour le mot de passe.

– Annuler la modification au quel cas le précédent mot de passe sera conservé.



Dixième cas d'utilisation C10 (STB-SmartCard C10:

Lors d'une première utilisation le couple login/mot de passe sera demandé à l'utilisateur. Il sera enregistré par Facecrypt pour utilisations ultérieures.

7. Traçabilité

Exigences		F-F0-10	F-FI-10	F-FI-20	F-FI-30	F-FQ-10	F-FQ-20	F-FQ-30	F-FR-10
COMPOSANTS	SSNExt		X	X	X	X			X
	Facecrypt	X		X	X	X		X	X
	SQLite			X	X	X			