

Spécification technique des besoins

Version	1.0
Date	12 décembre 2013
Rédigé par	Claire SMETS
Relu par	Edouard PASCAL
Approuvé par	Mr Otmani

MISES À JOUR

Version	Date	Modifications réalisées
0.1	02/12/2013	Création du Document
1.0	12/12/2013	Présentation de la première version

Table des matières

1	Objet	4
2	Documents applicables et de référence	4
3	Terminologie et sigles utilisés	4
4	Exigences fonctionnelles	5
4.1	Présentation de la mission du produit logiciel	5
4.2	UC 1 : Récupération des certificats	5
4.3	UC 2 : Factorisation des moduli des certificats	6
4.4	UC 3 : Présentation des résultats	6
4.5	UC 4 : Audit d'OpenSSL	6
4.6	UC 5 : Test d'un client	7
5	Diagrammes de cas d'utilisation	7
5.1	UC 1	7
5.2	UC 2	7
5.3	UC 3	8
5.4	UC 5	8
6	Exigences opérationnelles	8
7	Exigences d'interface	8
8	Exigences de qualité	8
9	Exigences de réalisation	8

1 Objet

Le projet s'inscrit dans un cadre relatif à un fait d'actualité récemment mis à jour, lié aux révélations de Snowden sur les pratiques de la NSA. Le monde de la cryptographie est en proie à de grandes incertitudes suite à ces révélations et remet en question tous les systèmes jusqu'alors développés. Notons que ce ne sont pas les algorithmes des systèmes cryptographiques qui sont remis en cause, mais leur développement machine qui n'est pas (ou plus) considéré comme nécessairement sûr. La NSA a par exemple très bien pu introduire des backdoors, ou faiblesses dans les logiciels, de telle sorte qu'ils puissent accéder aux données claires sans difficultés. On peut par exemple remettre en questions les outils développés par des laboratoires tels que RSA Labs, ou même encore des standards proposés par des agences comme le NIST (OpenSSL, AES, sont-ils vraiment sûrs ?)

Objectif technique :

- coder un outil permettant de récupérer un grand nombre de certificats
- coder un outil permettant de trouver d'éventuels facteurs communs aux certificats récupérés
- d'implémenter un système serveur permettant l'audit des clients. (éventuellement, selon le temps restant)

Résultat attendu :

Une étude sur les certificats utilisés sur internet, ainsi que l'implantation d'OpenSSL très utilisée.

2 Documents applicables et de référence

- Appel d'offre : Audit des implantations SSL/TLS
- les standards cryptographiques

3 Terminologie et sigles utilisés

- **Appli RC** : Application de Récupération des Certificats
- **Appli F** : Application de Factorisation
- **Machine D** : Machine Distant
- **Machine L** : Machine Locale
- **Certificat** : Un certificat électronique (aussi appelé certificat numérique ou certificat de clé publique) peut être vu comme une carte d'identité numérique. Il est utilisé principalement pour identifier une entité physique ou morale, mais aussi pour chiffrer des échanges.
Il est signé par un tiers de confiance qui atteste du lien entre l'identité physique et l'entité numérique (Virtuel).
- **Audit** : Procédure consistant à s'assurer du caractère complet, sincère et régulier des comptes d'une entreprise, à s'en porter garant auprès des divers partenaires intéressés de la firme et, plus généralement, à porter un jugement sur la qualité et la rigueur de sa gestion. Ici ce n'est pas une entreprise qui en sera la cible mais un programme : OpenSSL.
- **Modulus** : En mathématiques et plus précisément en théorie algébrique des nombres, l'arithmétique modulaire est un ensemble de méthodes permettant la résolution de problèmes sur les nombres entiers. Ces méthodes dérivent de l'étude du reste obtenu par une division euclidienne.

4 Exigences fonctionnelles

4.1 Présentation de la mission du produit logiciel

Id	Intitulé	Acteur(s)	Priorité
UC.1	Récupération des certificats	Appli RC, Machine D	Indispensable
UC.2	Factorisation des moduli des certificats	Appli F	Indispensable
UC.3	Présentation des résultats	Utilisateur	Indispensable
UC.4	Audit d'OpenSSL	Utilisateur	Indispensable
UC.5	Test d'un client	Client et Serveur	Optionnel

4.2 UC 1 : Récupération des certificats

Acteurs concernés	Appli RC, Machine D
Description	Récupérer une liste des certificats sur les adresses et les ports ouverts, scannés par ZMAP
Préconditions	Avoir une liste de machines distantes dont les ports sont ouverts
Événements déclenchants	L'administrateur lance l'application
Conditions d'arrêt	La liste est parcourue en entier
Description du flot d'événements principal	
Acteur(s)	Système
1. Lire la liste 2. Se connecter à la machine distante 3. Échanger les certificats 4. Stocker le certificat	3. Échanger les certificats
Flots d'exceptions	Coupure de connexion : retenter une deuxième fois si la Machine distante ne répond pas

4.3 UC 2 : Factorisation des moduli des certificats

Acteurs concernés	Appli F
Description	Pour chaque certificat obtenu, on essaie de lui trouver des facteurs communs dans son modulus
Préconditions	Avoir une liste de certificats et donc leurs moduli
Événements déclenchants	L'application est lancée
Conditions d'arrêt	La liste est parcourue en entier
Description du flot d'événements principal	
Acteur(s)	Système
1. Prendre les moduli 2. Construire l'arbre des produits 3. Construire l'arbre des restes 4. Sortir la liste des facteurs communs 5. Sortir la liste des moduli vulnérables trouvés	
Flots d'exceptions	Pas assez de ressources : Stocker l'arbre en dur au fur et à mesure.

4.4 UC 3 : Présentation des résultats

Acteurs concernés	Utilisateur
Description	Affiche les résultats produits par les précédentes fonctions
Préconditions	Des résultats ont été produits
Événements déclenchants	Demande d'affichage des ces résultats
Conditions d'arrêt	Demande d'arrêt d'affichage des résultats
Description du flot d'événements principal	
Acteur(s)	Système
1. Ouvrir la page web 2. Sélectionner et filtrer les données affichées 3. Exporter les données	
Flots d'exceptions	

4.5 UC 4 : Audit d'OpenSSL

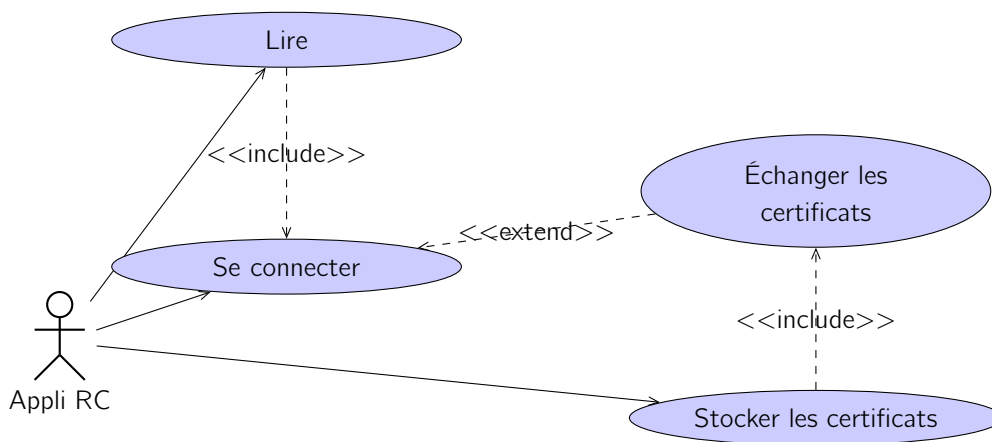
1. Nous étudierons tout d'abord les recommandations et les standards actuels (cf. PKCS) sous forme d'état de l'art, et nous en choisirons de façon argumentée un standard sur lequel nous baserons nos analyses.
2. Nous analyserons dans un deuxième temps le co de source d'Op enSSL, en particulier sa partie développant la génération d'éléments aléatoires et ses primitives cryptographiques. Nous en ferons de plus une description détaillée. Nous comparerons ensuite nos résultats avec les recommandations choisies, puis nous conclurons en déterminant si cette génération est efficace.

4.6 UC 5 : Test d'un client

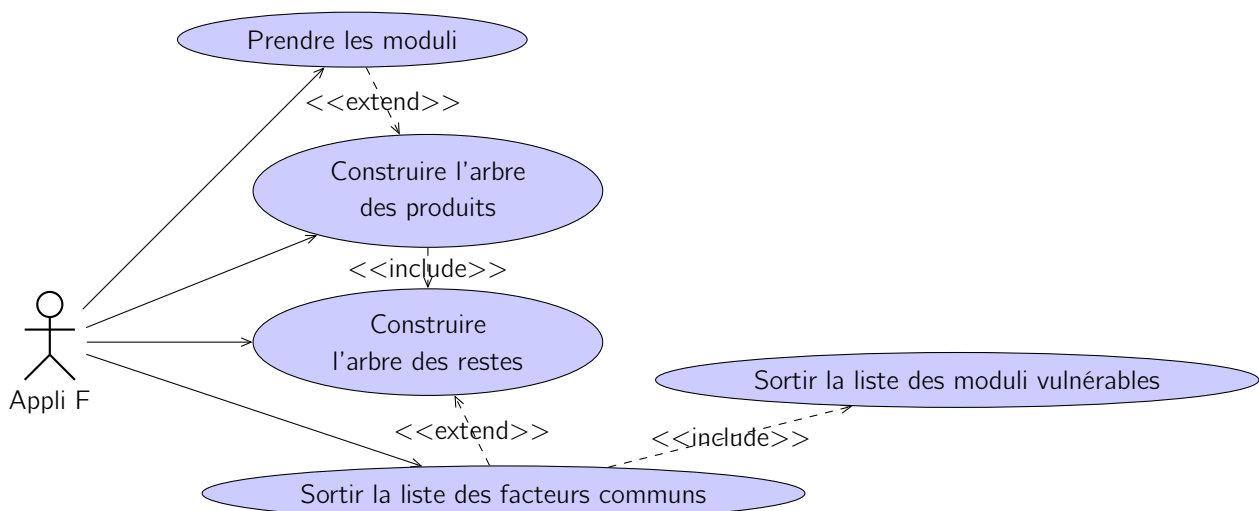
Suivant le temps restant, il nous est proposé d'implémenter un système serveur permettant l'audit des clients. Le logiciel sur le serveur doit évaluer la sécurité du client suivant un protocole établi, en analysant son système et/ou lui proposant des challenges spécifiques, en particulier autour de SSL. Le logiciel devra rendre un diagnostic et un rapport sur les différents éléments analysés, afin de permettre au client de constater des problèmes lorsqu'il y en a, et d'en connaître leur provenance.

5 Diagrammes de cas d'utilisation

5.1 UC 1



5.2 UC 2



```
graph LR; User((User)) --> A([Ouvrir la page web]); User --> B([Sélectionner et filtrer les données affichées]); B -.->|<<extend>>| C([Exporter les données]);
```

Diagramme UML d'un processus de sélection et d'exportation de données. Le diagramme implique un acteur nommé "User" et trois cas d'utilisation (Use Cases) représentés par des ovales :

- Ouvrir la page web** (Use Case 1)
- Sélectionner et filtrer les données affichées** (Use Case 2)
- Exporter les données** (Use Case 3)

Les relations entre les éléments sont les suivantes :

- Le **User** est associé à **Ouvrir la page web** et **Sélectionner et filtrer les données affichées** par des flèches solides.
- Il existe une relation d'extension (indiquée par une flèche en pointillés et le stéréotype **<<extend>>**) entre **Sélectionner et filtrer les données affichées** et **Exporter les données**.

