

# Document d'architecture logicielle

<b>Version</b>	0.1
<b>Date</b>	12/12/2013
<b>Rédigé par</b>	Julien Legras
<b>Relu par</b>	X
<b>Approuvé par</b>	X

## MISES À JOUR

Version	Date	Modifications réalisées
0.1	12/12/2013	Création du document

## Table des matières

<b>1</b>	<b>Objet</b>	<b>4</b>
<b>2</b>	<b>Documents applicables et de référence</b>	<b>4</b>
<b>3</b>	<b>Terminologie et sigles utilisés</b>	<b>4</b>
<b>4</b>	<b>Configuration requise</b>	<b>4</b>
4.1	Machine de récupération de certificats . . . . .	4
4.2	Machine de factorisation . . . . .	4
<b>5</b>	<b>Architecture statique</b>	<b>5</b>
5.1	Structure . . . . .	5
5.2	Description des constituants . . . . .	5
<b>6</b>	<b>Fonctionnement dynamique</b>	<b>7</b>
6.1	UC.1 : Récupération des certificats . . . . .	7
6.1.1	UC.1.2 : Récupération des certificats SSH . . . . .	7
6.1.2	UC.1.2 : Récupération des certificats SSL/TLS . . . . .	8
6.2	UC.2 : Factorisation des moduli des certificats . . . . .	9
6.3	UC.3 : Présentation des résultats . . . . .	10
6.4	UC.4 : Audit d'OpenSSL . . . . .	11

## 1 Objet

Ce document présente l'architecture utilisée pour réaliser les outils nécessaires à notre audit SSL :

- Récupération des certificats : Application RC ;
- Factorisation : Application F.

## 2 Documents applicables et de référence

- STB (Spécification Technique des Besoins) ;
- N. Heninger, Z. Durumeric, E. Wustrow, J. A. Halderman. Mining Your Ps and Qs : Detection of Widespread Weak Keys in Network Devices. Proceedings of the 21st USENIX Security Symposium. 2012.

## 3 Terminologie et sigles utilisés

- **Appli RC** : Application de Récupération des Certificats
- **Appli F** : Application de Factorisation
- **Machine D** : Machine Distante
- **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 Configuration requise

### 4.1 Machine de récupération de certificats

- accès aux ports 22 et 443 en destination
- bande passante importante
- espace disque pour stockage des certificats : 20 Go
- système : Ubuntu Server 13.10

### 4.2 Machine de factorisation

- au moins 8 Go de RAM
- processeur quatre cœurs
- espace disque pour stockage : 20 Go

- système : Ubuntu Server 13.10

## 5 Architecture statique

### 5.1 Structure

Les principales parties à développer sont :

- Application récupération de certificats
- Application de factorisation de grands entiers RSA
- Données :
  - adresses IP fournies par ZMap
  - clefs publiques RSA récupérées par l'application RC

### 5.2 Description des constituants

Constituant	
Rôle	
Propriétés et attributs de caractérisation	
Dépendances avec d'autres constituants	
Langages de programmation	
Procédé de développement	
Taille complexité	

Application RC SSH	
Rôle	Récupération et stockage des certificats SSH
Propriétés et attributs de caractérisation	Permet d'obtenir des certificats SSH en tentant une connexion
Dépendances avec d'autres constituants	ZMap : liste des adresses IP ayant le port 22 ouvert
Langages de programmation	C
Procédé de développement	<ol style="list-style-type: none"><li>1. client SSH simple</li><li>2. itérer les connexions pour toutes les adresses IP données par ZMap</li></ol>
Taille complexité	20% du projet

Application RC SSL/TLS	
Rôle	Récupération et stockage des certificats SSL/TLS
Propriétés et attributs de caractérisation	Permet d'obtenir des certificats SSL/TLS en tentant une connexion
Dépendances avec d'autres constituants	ZMap : liste des adresses IP ayant le port 443 ouvert
Langages de programmation	C
Procédé de développement	<ol style="list-style-type: none"> <li>1. client SSL/TLS simple</li> <li>2. itérer les connexions pour toutes les adresses IP données par ZMap</li> </ol>
Taille complexité	20% du projet

Application F	
Rôle	Trouver des facteurs communs des clefs récupérées
Propriétés et attributs de caractérisation	???
Dépendances avec d'autres constituants	Application RC SSH et SSL/TLS
Langages de programmation	C
Procédé de développement	<ol style="list-style-type: none"> <li>1. Arbre des produits</li> <li>2. Arbre des restes</li> <li>3. Exploitation des résultats</li> </ol>
Taille complexité	10%

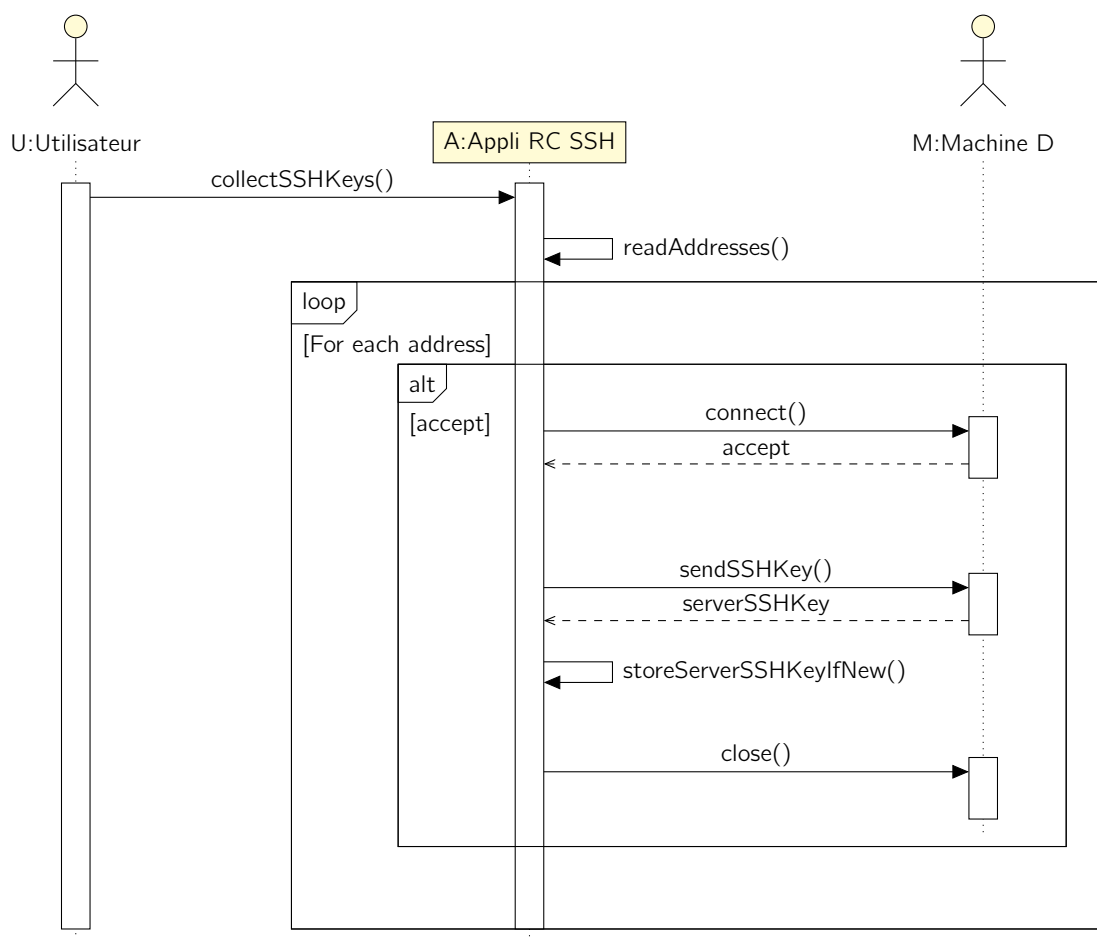
Serveur web	
Rôle	Affiche les résultats de l'audit
Propriétés et attributs de caractérisation	Pages statiques et dynamiques qui affichent sous forme graphique les résultats de l'audit
Dépendances avec d'autres constituants	Appli RC, F
Langages de programmation	HTML, PHP, JS
Procédé de développement	<ol style="list-style-type: none"> <li>1. Affichage clefs récupérées</li> <li>2. Affichage des facteurs communs</li> <li>3. Documentation audit OpenSSL</li> </ol>
Taille complexité	10%

## 6 Fonctionnement dynamique

### 6.1 UC.1 : Récupération des certificats

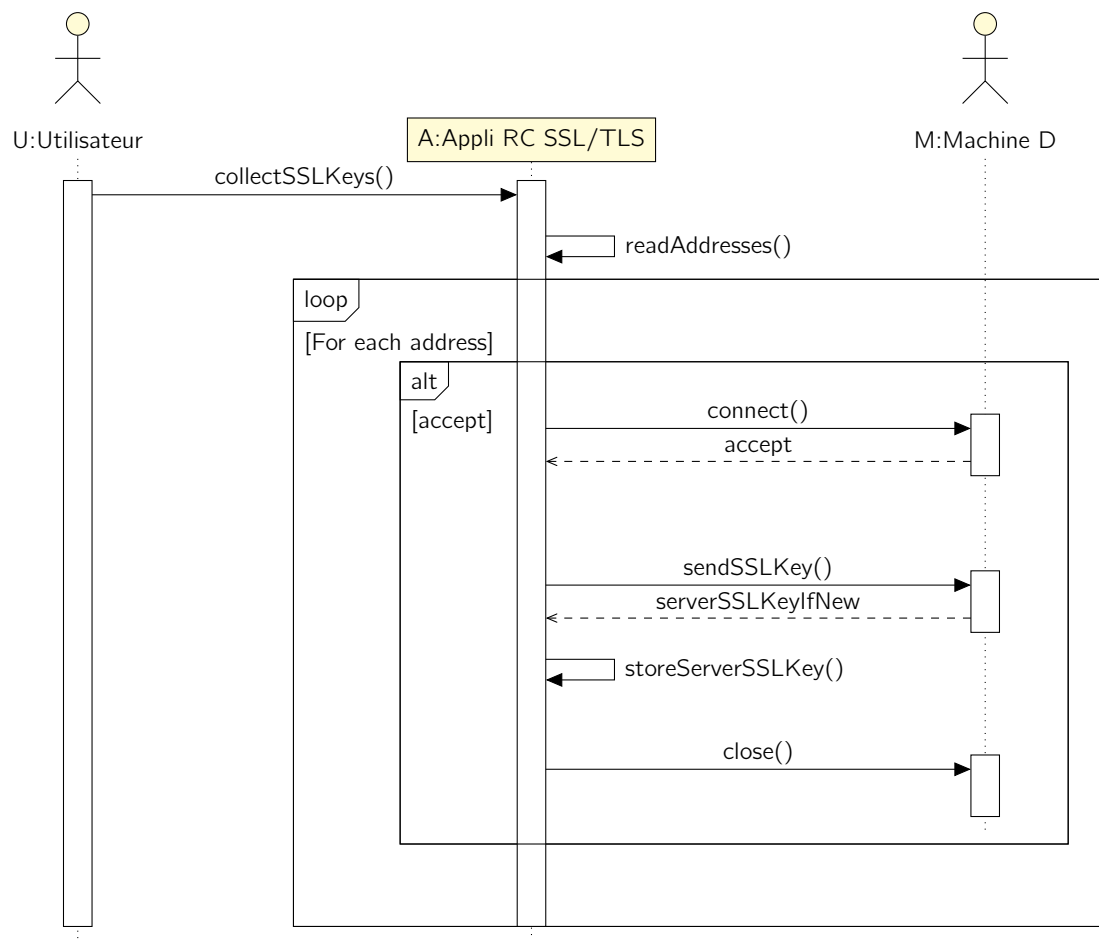
#### 6.1.1 UC.1.2 : Récupération des certificats SSH

UC.1.2 : Récupération des certificats SSH	
<b>Composants mis en jeu</b>	Appli RC SSH
<b>Intervenants</b>	Utilisateur, Machine D
<b>Processus de mise en œuvre</b>	
Pour toutes les adresses IP ayant un port 22 ouvert : <ol style="list-style-type: none"> <li>Établissement de la connexion TCP entre Appli RC SSH et Machine D</li> <li>Échange des clefs/certificats entre Appli RC SSH et Machine D</li> <li>Appli RC SSH stocke la clef/le certificat de Machine D dans un répertoire <code>keys_ssh</code> sous forme de fichier dont le nom est l'adresse IP de Machine D (si la clef n'est pas déjà présente)</li> <li>Fermeture de la connexion</li> </ol>	



### 6.1.2 UC.1.2 : Récupération des certificats SSL/TLS

UC.1.2 : Récupération des certificats SSL/TLS	
<b>Composants mis en jeu</b>	Appli RC SSL/TLS
<b>Intervenants</b>	Utilisateur, Machine D
<b>Processus de mise en œuvre</b>	
<p>Pour toutes les adresses IP ayant un port 443 ouvert :</p> <ol style="list-style-type: none"> <li>1. Établissement de la connexion TCP entre Appli RC SSL/TLS et Machine D</li> <li>2. Échange des clefs/certificats entre Appli RC SSL/TLS et Machine D</li> <li>3. Appli RC SSL/TLS stocke la clef/le certificat de Machine D dans un répertoire keys_ssl sous forme de fichier dont le nom est l'adresse IP de Machine D (si la clef n'est pas déjà présente)</li> <li>4. Fermeture de la connexion</li> </ol>	





## 6.2 UC.2 : Factorisation des moduli des certificats

UC.2 : Factorisation des moduli des certificats	
Composants mis en jeu	Appli F
Intervenants	Utilisateur
Processus de mise en œuvre	
<ol style="list-style-type: none"> <li>1. Récupérer la liste des clefs</li> <li>2. Construire l'arbre des produits en stockant chaque niveau dans un fichier a</li> <li>3. Construire l'arbre des restes</li> <li>4. Stocker les facteurs communs dans un fichier</li> </ol>	

**Entrées** : Tableau des moduli des clefs publiques :  $T$

**Sorties** : Hauteur de l'arbre, produits des moduli des clefs publiques

**Données** : Tableaux  $v$ ,  $tmp$ ; Entier  $i$ ,  $level$

$v \leftarrow T$ ;

$level \leftarrow 0$ ;

**tant que**  $|v| > 1$  **faire**

$tmp \leftarrow \emptyset$ ;

**pour chaque**  $i \in \{0, \dots, |v|/2\}$  **faire**

$tmp[i] \leftarrow v[i \times 2] \times v[i \times 2 + 1]$ ;

**fin**

$storeProductLevel(v, level)$ ;

$v \leftarrow tmp$ ;

$level \leftarrow level + 1$ ;

**fin**

**retourner**  $level$

**Algorithme 1:** Construction de l'arbre des produits

**Entrées** : Hauteur de l'arbre des produits : level

**Sorties** : PGCDs des moduli des clefs publiques

**Données** : Tableaux P, v, w; Entier i

$P \leftarrow getProductLevel(level);$

**tant que** level > 0 **faire**

$v \leftarrow getProductLevel(level - 1);$

**pour chaque**  $i \in \{0, \dots, |v|\}$  **faire**

$v[i] \leftarrow P[i/2] \pmod{v[i]^2};$

**fin**

    level  $\leftarrow 1;$

    storeRemainderLevel(v, level);

$v \leftarrow tmp;$

    level  $\leftarrow level + 1;$

**fin**

$w \leftarrow \emptyset;$

**pour chaque**  $i \in \{0, \dots, |v|\}$  **faire**

$w[i] \leftarrow P[i/2] \pmod{v[i]^2};$

$w[i] \leftarrow w[i]/v[i];$

$w[i] \leftarrow pgcd(w[i], v[i]);$

**fin**

**retourner** w

**Algorithme 2:** Construction de l'arbre des restes

### 6.3 UC.3 : Présentation des résultats

UC.3 : Présentation des résultats	
<b>Composants mis en jeu</b>	Serveur web
<b>Intervenants</b>	Utilisateur
<b>Processus de mise en œuvre</b>	
<ol style="list-style-type: none"> <li>1. Développement partie clefs publiques récupérées (tri par critères : taille, issuer si connu etc.)</li> <li>2. Développement partie facteurs communs avec graphiques HighCharts</li> <li>3. Documentation avec doxygen de l'audit d'OpenSSL</li> </ol>	

## 6.4 UC.4 : Audit d'OpenSSL

UC.4 : Audit d'OpenSSL	
Composants mis en jeu	code source d'OpenSSL
Intervenants	
Processus de mise en œuvre	
<ol style="list-style-type: none"><li>1. Sélectionner les fonctions à auditer</li><li>2. Développer les codes de tests sur ces fonctions</li><li>3. Générer une documentation doxygen associée à l'audit</li></ol>	