Division R&D / MAPS / NSS

France Télécom
Division Recherche & Développement
Middleware and Advanced Platforms
Network and Service Security

	C2 **R&D / ANTD26 / 799AF**	**Diffusion contrôlée**

APPLATOO

CIBLE DE SECURITE

Livrable 1.2.a

Référence FTR&D : DTL/SSR/284.03/LF/Livrable 1.2.aa

Contact : support.applatoo@francetelecom.com

TABLES DES MATIERES

1. INTRODUCTION DE LA CIBLE DE SECURITE

1.1. IDENTIFICATION DE LA CIBLE DE SECURITE

Titre: Cible de sécurité APPLATOO.
Mots-clés : intégrité des données, signature sécurisée, vérification de signature, calcul d'un *condensat* (voir annexe), confidentialité des données, *chiffrement symétrique* (voir annexe), *déchiffrement symétrique* (voir annexe), génération de *clefs de session* (voir annexe), *chiffrement asymétrique* (voir annexe), *déchiffrement asymétrique* (voir annexe), vérification de la confiance, gestion de certificats, vérification de jeton d'horodatage, interface haute.
Référence Cible de Sécurité : C2-LV1.2.a-DCSSI APPLATOO-Cible de sécurité 1.12-20050214.doc.
Référence TOE : APPLATOO version 1.2.4
CESTI évaluateur : AQL.
Niveau d'évaluation visé : EAL2 augmenté de ADV_HLD.2, ADV_LLD.1*, ADV_IMP.1*, ALC_DVS.1, ALC_FLR.3, ALC_TAT.1*, AVA_MSU.1 et AVA_VLA.2. (* Pour les fonctions cryptographiques spécifiées par les composants de la classe FCS uniquement.)

Remarque : Les mots en italique font référence à des définitions se trouvant en annexe.

1.2. PRESENTATION GENERALE DE LA CIBLE DE SECURITE

Le besoin en services sécurisés est croissant, aussi bien au niveau des échanges dans l'entreprise, qu'au niveau des administrations (pour des échanges inter administrations, avec des entreprises, ou avec le grand public).

La solution APPLATOO a été développée pour répondre aux besoins d'entreprises et d'administrations qui développent des services sécurisés à base de signature, d'horodatage, de chiffrement/déchiffrement.

APPLATOO constitue une "interface haute" offrant des fonctions de base de sécurité : vérification de signature, fonctions relatives à la confiance, calcul de *condensats* (hachés), vérification de jetons d'horodatage, *chiffrement/déchiffrement symétrique*.
Ces fonctions sont les ressources mises à la disposition d'applications utilisatrices.
Par exemple on peut envisager la mise en œuvre d'une application basée sur APPLATOO pour réaliser de la signature électronique ou du chiffrement de document.

Cette "interface haute" permet de développer des services :
- ✓ Sur serveur via des servlets,
- ✓ Sur client dans un navigateur via des applets ou des appels JavaScript,
- ✓ Sur une machine, dans une application autonome.

Les données à protéger sont des informations qui peuvent avoir un caractère confidentiel et/ou d'intégrité et/ou d'authenticité :
- ✓ Documents RH à l'attention des employés de l'entreprise (notification d'augmentation…)
- ✓ Bons de commandes,
- ✓ Déclarations,
- ✓ Etc.

Il s'agit plus généralement d'informations dont la divulgation peut potentiellement créer un préjudice à l'une des parties concernées et/ou dont l'intégrité doit être garantie (et donc vérifiée).

APPLATOO vise à assurer un caractère de confidentialité et/ou d'intégrité à ces informations.

1.3. CONFORMITE AUX CC

La version des Critères Communs applicable est la version 2.2.

La fonctionnalité de sécurité de la cible d'évaluation est « Conforme à la partie 2 des Critères Communs »

Les mesures d'assurance sécurité mises en œuvre sur la cible d'évaluation sont « Partie 3 étendue ». Le paquet d'exigences d'assurance sécurité est celui du niveau standard défini par la DCSSI (cf. [QPS-std]).

Aucune conformité à un quelconque Profil de Protection n'est formulée.

Le niveau de résistance minimum demandé pour les fonctions de sécurité de la TOE et pour les exigences fonctionnelles de sécurité de la TOE est 'SOF-High'. Aucune annonce spécifique supplémentaire de résistance des fonctions n'est formulée.

2. DESCRIPTION DE LA CIBLE D'EVALUATION

2.1. TYPE DE LA CIBLE D'EVALUATION (TOE)

La TOE est un produit logiciel, développé en java, offrant des fonctionnalités de sécurité conçues pour être incorporées au sein de multiples systèmes et utilisés par des applications.

Le besoin de signature électronique correspond à un besoin de confiance en la validité de l'émetteur. Il permet aussi de s'assurer de la non-modification du document après sa signature. La signature se déroule en plusieurs phases : réception du document par APPLATOO, génération du *condensat* de ce document, puis signature de ce *condensat*.

La vérification de signature permet de valider le signataire et de s'assurer que le document est intègre. Les différentes phases de cette vérification sont : réception du document et du certificat signataire par APPLATOO, génération du *condensat* du document, vérification de la signature du document à l'aide du certificat et par comparaison avec le *condensat* généré.

Dans le cas d'un jeton d'horodatage, APPLATOO peut également être utilisée pour s'assurer d'une heure d'envoi ; pour cela le document est signé par un serveur d'horodatage considéré comme sûr qui associe à sa signature l'heure de celle-ci. APPLATOO permet de vérifier les jetons d'horodatage reçus mais aussi d'émettre des requêtes vers des serveurs d'horodatage (cependant dans cette dernière fonction APPLATOO n'intervient pas dans les processus de sécurisation). Dans ce cas, APPLATOO génère un jeton d'horodatage au bon format et l'envoi au serveur d'horodatage désigné et dans le cas de la vérification du jeton, APPLATOO effectue une vérification de signature sur le jeton et contrôle le format.

Le besoin de chiffrement est quant à lui différent, il permet de s'assurer de la non divulgation d'un document. Les phases du chiffrement sont les suivantes : réception du document par APPLATOO, génération d'une *clef de session*, *chiffrement symétrique* du document à l'aide de cette clef, *chiffrement asymétrique* de cette *clef de session* à l'aide d'une clef publique.

Le déchiffrement est quant à lui composé des phases suivantes : réception du document par APPLATOO, *déchiffrement asymétrique* de la *clef de session* par une clef privée, *déchiffrement symétrique* du document par la *clef de session* obtenue.

La TOE est un sous-ensemble de la plate-forme APPLATOO, il est décrit dans les paragraphes ci-dessous.

APPLATOO présente une interface haute offrant des fonctions de base de sécurité : signature et vérification de signature, vérification de jeton d'horodatage, chiffrement/déchiffrement fonctions de vérification de *chaîne de confiance*… Durant la réalisation de ces fonctions mises à disposition des applications utilisatrices, APPLATOO s'appuie sur des ressources extérieures.

APPLATOO permet l'utilisation des données des magasins de certificats et l'utilisation des ressources externes. Il reste cependant piloté par le service sécurisé et se contente de répondre à ses requêtes. Les données utilisées par APPLATOO sont juste consultées et en aucun cas modifiées.

Le tableau ci-dessous précise la contribution d'APPLATOO et celle de son environnement extérieur dans les tâches de chiffrement/déchiffrement et de signature/vérification de signature.

Opérations cryptographiques	**Chiffrement d'un document**	**Déchiffrement d'un document**	**Signature d'un document**	**Vérification de signature d'un document**
Parties effectuées par APPLATOO	Génération de la *clef de session* *Chiffrement symétrique* avec la *clef de session* *Chiffrement asymétrique* de la *clef de session* à l'aide de la clef publique	*Déchiffrement symétrique* du document à l'aide de la *clef de session*	Génération du *condensat* du document	Génération du *condensat* du document Déchiffrement asymétrique du *condensat* à l'aide du certificat public de l'émetteur Comparaison du *condensat* vérifié avec celui réalisé
Parties effectuées à l'extérieur d'APPLATOO		*Déchiffrement asymétrique* de la clef de session à l'aide de la clef privée	Signature du *condensat* à l'aide d'une clef privée	

Ces fonctions sont disponibles pour tous les services s'appuyant sur cette plate-forme.

2.2. PORTEE DE LA TOE ET FRONTIERES

La TOE contribue à la sécurisation des opérations de signature, de vérification de signature, de vérification d'horodatage, de chiffrement/déchiffrement d'un document :

- ✓ En offrant une interface unique aux applications utilisant ces fonctions (et donc aux développeurs de services sécurisés),
- ✓ En offrant une gestion des différents paramètres cryptographiques (algorithmes utilisés, longueurs de clés…)
- ✓ En offrant une interface d'accès uniformisée à des ressources cryptographiques bas niveau de différents environnements matériels.

L'ensemble des fonctions de signature et de *déchiffrement asymétrique* à l'aide de clefs privées sera fourni par les ressources sur lesquelles s'appuie APPLATOO.

APPLATOO peut également mettre en œuvre un procédé de ressources cryptographiques internes pour accéder à des certificats et clefs directement stockés sur le disque. Cette fonctionnalité ne fait pas partie du cadre de cette évaluation.

Formats supportés et connexions aux ressources cryptographiques :

Ce paragraphe permet de détailler techniquement les types de ressources auxquelles APPLATOO peut faire appel par l'intermédiaire de connecteurs faisant également l'objet de l'évaluation.

Pour permettre un accès aux différentes ressources du poste, un pont (entre le monde Java de APPLATOO et le monde d'OS particuliers) a été développé ainsi que des connecteurs vers des API spécifiques.

Ces connecteurs sont propres aux différentes ressources cryptographiques :

- ✓ Connecteur *CAPI* (voir annexe), qui permet d'accéder aux ressources accessibles depuis l'API CAPI (Microsoft). Via ce connecteur, le pont pourra accéder :
    - o Aux ressources cryptographiques des *CSP* internes de Microsoft : clefs et certificats associés, fonctions de signature et de déchiffrement utilisant les clefs privées, bases de confiance, éventuellement fonctions de génération de clefs secrètes
    - o Aux ressources des autres *CSP* gérés via CAPI : clefs et certificats associés stockés sur cartes à puces ou dongles *USB* par exemple, fonctions de signature et de déchiffrement utilisant les clefs privées, éventuellement fonctions de générations de clefs à bord des cartes
- ✓ Connecteur Confiance Netscape, qui permet d'accéder à la base de certificats de confiance de Netscape. On trouve dans cette base les certificats des Autorités de Certification et des Utilisateurs auxquels la confiance a été explicitement accordée.
- ✓ Connecteur fédérateur *PKCS#11* (voir annexe) permettant au pont de faire appel aux modules *PKCS#11* présents sur le poste via cette API *PKCS#11*. C'est à ce niveau que sont gérés les points suivants :
    - o persistance ou non des mots de passe (codes PIN) ;
    - o sessions *PKCS#11* : cette gestion doit permettre des performances en temps acceptables, c'est à dire de même durée à 20% près que le navigateur Netscape ; en particulier, il est préconisé de ne pas fermer les sessions *PKCS#11* après chaque opération, mais à la fin de l'utilisation de la plate-forme.

  Les modules *PKCS#11* auxquels il pourra être fait appel sont tous ceux décrits plus loin, en particulier :
    - o Un module *PKCS#11* permettant d'accéder aux clefs et certificats internes de Netscape et fondé sur les sources publiques du module *PKCS#11* interne de Netscape ;
    - o Divers modules *PKCS#11* offrant des accès à des supports cryptographiques (cartes à puces, dongles *USB*, etc.) du commerce (Gemplus, Oberthur, Schlumberger, Rainbow-iKey…) ;
    - o Eventuellement un module *PKCS#11* entièrement logiciel pouvant faire appel à un magasin de clefs et certificats logiciel (par exemple celui de MSI), et tout autre module *PKCS#11* identifiable.

Un des rôles de la TOE est donc la mise au format des différentes ressources pour pouvoir être utilisées par le middleware APPLATOO et mettre en œuvre des fonctions génériques sur ces objets.

2.2.1. LES SERVICES DE SECURITE DE LA TOE

Cette section précise les fonctions de sécurité rendues par la TOE et qui sont mises à disposition de services de sécurité ; ces différents services sont mis en œuvre par la TOE :

- ✓ Le *chiffrement/déchiffrement symétrique ;*
- ✓ Le *chiffrement asymétrique* à l'aide d'une clef publique dans le cas d'un chiffrement de données;
- ✓ Le *déchiffrement asymétrique* à l'aide d'une clef publique dans le cas d'une vérification de condensat;
- ✓ Génération de *clefs de session* ;
- ✓ Vérification de la signature numérique de messages ;
- ✓ Génération de *condensats* ;
- ✓ Vérification d'un jeton d'horodatage (basé sur la vérification de la signature du jeton et le contrôle de certains champs comme la politique de signature) ;
- ✓ Vérification de la confiance en un certificat donné :
    - o Vérification de *key usage* (l'utilisation prévue du certificat est bien celle qui en est faite) (optionnelle);
    - o Vérification que le certificat correspond à un utilisateur désigné par son adresse e-mail ou à un serveur donné par son URL (optionnelle);
    - o Etablissement d'une *chaîne de confiance* (voir annexe) commençant par le certificat à vérifier, se terminant par un certificat de confiance et constituée des certificats en lesquels on a confiance (certificats d'Autorité de Confiance, certificats auxquels on a explicitement accordé notre confiance).

      Pour tout certificat de cette chaîne, il est vérifié sa validité ;
    - o Vérification de la validité du certificat :
        - ✓ Vérification de la non révocation du certificat : non périmé, non prématuré, actif (optionnelle mais si demandée alors appliquée à toute la chaîne de confiance). Le contrôle de révocation est réalisé par le protocole OCSP ou par CRL.
        - ✓ Vérification de la signature du certificat.

2.2.2. PRESENTATION SOMMAIRE DE L'ARCHITECTURE

Afin de cerner précisément le périmètre de la TOE, nous présentons ci-dessous l'architecture du produit APPLATOO. La TOE est délimitée par le cadre noir sur fond grisé et comporte les blocs fonctions de sécurité, ressources, primitives et données ainsi que le pont JNI.

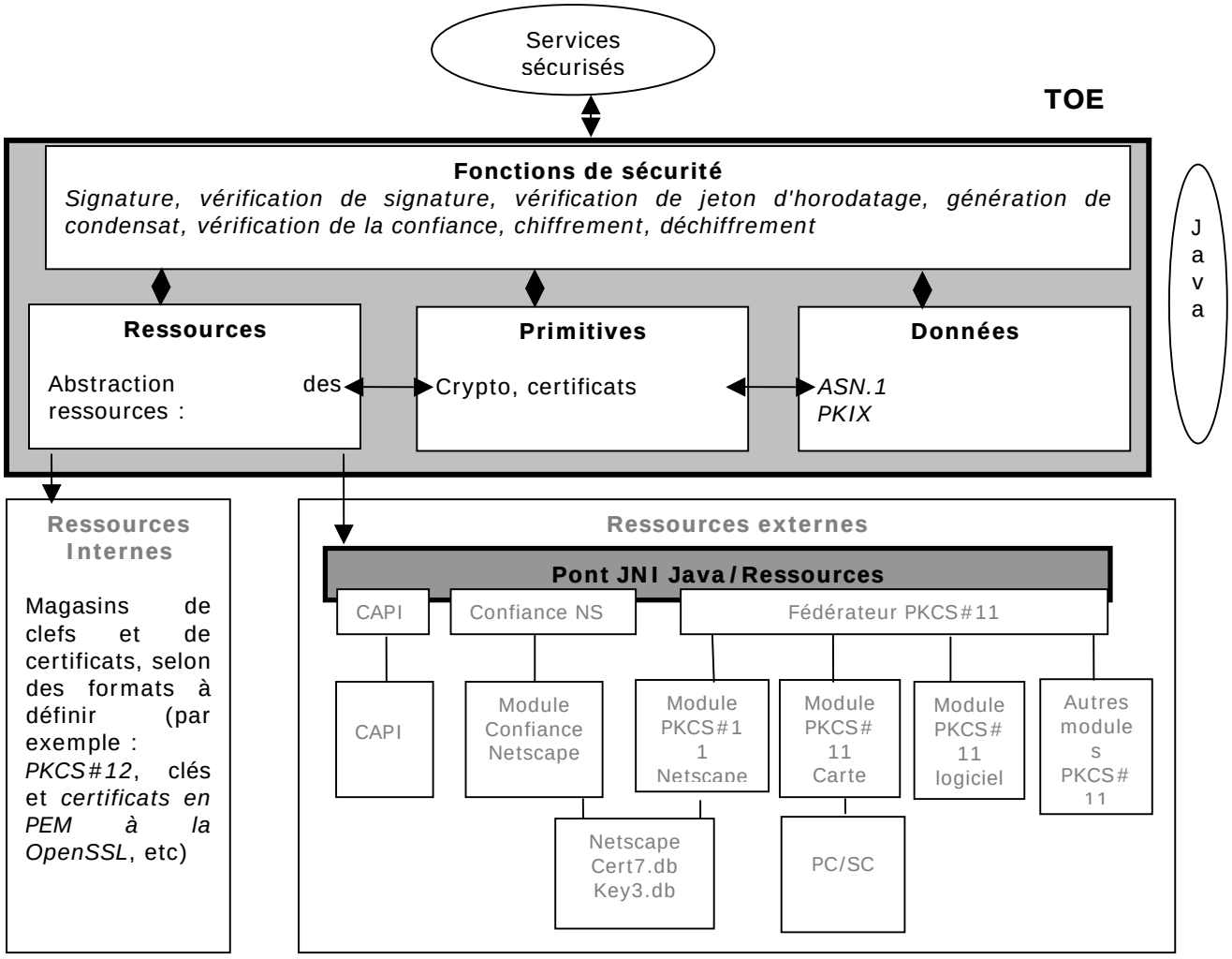

Figure 1 Architecture APPLATOO et périmètre de la TOE (Cas nominal pour le service)

Légende

Le Pont permet l'accès aux ressources externes à APPLATOO. Il doit donc être physiquement situé sur la plate-forme à laquelle les différentes ressources que l'on souhaite utilisée sont connectées.

L'exécution d'APPLATOO a lieu sur la plate-forme où se trouve les ressources pour permettre l'accès à ces dernières

Le module Fonctions offre aux applications une interface haut niveau pour développer des applications utilisatrices.

Le module Ressources permet de réaliser l'abstraction de ressources cryptographiques (clés et certificats). L'abstraction des ressources consiste en fait à les porter dans le monde Java puis à les mettre à un format commun pour qu'elles puissent être indifféremment traitées par APPLATOO. Dans le cas de ressources externes une dll du *pont JNI* et linkées puis abstraite dans le monde Java.

Dans le cas de ressources internes, le *PKCS#12* est lu sur le disque puis abstrait dans le monde Java. Certaines ne sont pas directement accessibles en Java standard (« Ressources externes »). Le module Ressources permet l'abstraction des Ressources pour produire des objets au format du module Primitives ;

Le module Primitives rassemble :

- des algorithmes cryptographiques (génération de condensats…),
- des algorithmes de gestion des certificats et de la révocation.

Ce module ne doit dépendre que du module Données.

Le module Données permet la manipulation de données et messages cryptographiques dans des formats standard (*PKCS#7*, etc.), en permettant notamment leur codage et leur décodage. Seuls les standards issus de *PKIX* sont envisagés d'être implémentés. L'architecture permet cependant de concevoir à l'avenir des services reposant sur d'autres standards. Le module Données doit être autonome. Il définit les différents formats utilisés au sein d'APPLATOO.

Un cinquième module rassemble des Utilitaires d'applet pour :

- gérer les écarts des plates-formes aux spécifications Java et les spécificités de l'OS ;
- le cas échéant, gérer l'auto installation et la mise à jour partielle.

Le pont permettant l'accès aux ressources externes fera partie du processus d'évaluation. Il permet l'accès à ces ressources pour les transporter dans l'environnement Java.

Remarque :

Nominalement, les applications utilisatrices accèdent au module Fonctions, et pas directement aux modules Ressources ou Primitives. Le module Fonction est à voir comme une boîte à outils cryptographique destinée au développement des fonctions haut niveau. Un service spécifique peut par contre avoir besoin d'un contrôle plus fin sur les manipulations cryptographiques (par exemple : production/vérification d'une signature sans emballage en *PKCS#7*).

L'architecture présentée ci-dessus montre le cas nominal d'utilisation, c'est pourquoi le cas particulier d'accès à d'autres modules n'est pas représenté. En effet, les applications ne feront pas alors appel aux Fonctions, mais directement aux Primitives. De multiples cas de ce type peuvent être envisagés, car le développeur de service peut avoir accès à toute méthode publique d'APPLATOO et donc si besoin est, redévelopper des fonctions de signatures spécifiques, par exemple. Le mode d'utilisation envisagé dans la suite de l'évaluation est le mode nominal en accédant au module Fonctions, les autres modes d'interaction étant des facilités pour des développements particuliers de services sécurisés.

2.2.3. LE CYCLE DE VIE DE LA TOE

En tant que brique logicielle, la TOE passe, entre la fin de son développement et le début de son exploitation, par une phase durant laquelle elle est sous le contrôle d'un développeur de services sécurisés, distinct du développeur de la TOE, mais qui ne relève pas de l'exploitation car les fonctions de sécurité ne sont pas actives ; ou du moins ne le sont pas en situation réelle, mais plutôt en situation de test et d'intégration.

Le modèle de cycle de vie standard des Critères Communs suppose une séparation entre le développement des fonctions de sécurité et l'exploitation, la transition étant assurée par les mesures de sécurité couvertes par la classe ADO : livraison, installation, génération et démarrage.

Dans le cadre de l'évaluation, le développement est interprété comme le développement de la brique logicielle APPLATOO, et l'exploitation englobe le développement de services sécurisés en plus de l'exécution de la TOE finalement intégrée aux services sécurisés. Le développeur de

services sécurisés est un utilisateur au sens des Critères Communs, et les guides de programmation d'APPLATOO constituent de la documentation d'exploitation.

2.2.4. ENVIRONNEMENTS D'EXECUTION DE LA TOE ET OPERATIONS CRYPTOGRAPHIQUES

Deux modes d'utilisation sont possibles suivant le service sécurisé développé à l'aide de la TOE : mode servlet (exécution sur un serveur) ou mode applet (execution à l'aide d'un navigateur sur un client).

Les environnements supportés en mode applet sont:

Windows 98	Windows Me	Windows NT4	Windows 2000	Windows XP	MACOS8/9	MacOSX	Linux
		SP3 à SP6	Sans SP à SP4	Sans SP, SP1			2.0, 2.2, 2.4
IE5.0 Machine virtuelle Microsoft	IE5.5 à 6 Machine virtuelle Microsoft	IE5.5 à 6 Machine virtuelle Microsoft	IE5.5 à 6 Machine virtuelle Microsoft	IE5.5 à 6 Machine virtuelle Microsoft			
NS4.7 à 4.78 Machine virtuelle NS	NS4.7 à 4.78 Machine virtuelle NS	NS4.7 à 4.78 Machine virtuelle NS	NS4.7 à 4.78 Machine virtuelle NS	NS4.7 à 4.78 Machine virtuelle NS	NS4.7 à 4.78 Machine virtuelle SUN (MRJ)	NS4.7 à 4.78 Machine virtuelle SUN (MRJ)	NS4.7 à 4.78 Machine virtuelle NS
NS 6.2 Machine virtuelle de SUN	NS 6.2 Machine virtuelle de SUN	NS 6.2 Machine virtuelle de SUN	NS 6.2 Machine virtuelle de SUN	NS 6.2 Machine virtuelle de SUN	NS 6.2 Machine virtuelle de SUN	NS 6.2 Machine virtuelle de SUN	NS 6.2 Machine virtuelle de SUN
Mozilla 1.0 à 1.5 Machine virtuelle de SUN	Mozilla 1.0 à 1.5 Machine virtuelle de SUN	Mozilla 1.0 à 1.5 Machine virtuelle de SUN	Mozilla 1.0 à 1.5 Machine virtuelle de SUN	Mozilla 1.0 à 1.5 Machine virtuelle de SUN	Mozilla 1.0 à 1.2 Machine virtuelle de SUN (MRJ)	Mozilla 1.0 à 1.5 (excepté 1.3) Machine virtuelle de SUN (MRJ)	Mozilla 1.0 à 1.5 Machine virtuelle SUN

NB : machine virtuelle de SUN : de la 1.3.1_02 jusqu'à la 1.4.2 (pour IE ou Netscape sous Windows). MAC OS 8/9 est exclus de l'évaluation ainsi que Netscape4.7 à 4.77 pour MacOS X.

S'il est utilisé en mode servlet, la configuration d'évaluation sera la suivante :

- ✓ Linux 2.0, 2.2, 2.4
- ✓ J2EE 1.4.2
- ✓ Tomcat 4.1.3

Les différents supports cryptographiques accessibles par le middleware sont :

Schlumberger (Cryptoflex eGate 32 K, Logiciel : SmartCard User Kit 4.3)

ActivCard (Schlumberger Cryptoflex 16 K, Schlumberger Cyberflex 16 K, Logiciel : ActivCard Gold v2.2)
Oberthur (Cartes à puce : Oberthur AuthentIC 16 K, Lecteur : OCR150)
Gemplus GCR410 et cartes GemSafe 8K
Rainbow-iKey : clé usb Rainbow ikey 2032 avec la librairie PKCS#11 dkck232.dll
PKCS#12 (Ressources internes)

Voici les opérations cryptographiques, les algorithmes et les caractéristiques disponibles :

- Algorithmes de hash (fournis par la TOE) :

SHA-1

- Cryptographie symétrique (fournie par la TOE) :

Pour les modes CBC, OFB, CFB une valeur initiale doit être imposée.

Pour les modes ECB les applications les utilisant doivent se limiter à des utilisations ne facilitant pas l'exploitation de leurs vulnérabilités.

L'utilisation de ces modes sera précisée dans les guides de développement.

des-ede3-cbc; tailles de clés: 128, 192[1].

id-aes128-ECB; tailles de clés: 128.

id-aes128-CBC; tailles de clés: 128.

id-aes128-OFB; tailles de clés: 128.

id-aes128-CFB; tailles de clés: 128.

id-aes192-ECB; tailles de clés: 192.

id-aes192-CBC; tailles de clés: 192.

id-aes192-OFB; tailles de clés: 192.

id-aes192-CFB; tailles de clés: 192.

id-aes256-ECB; tailles de clés: 256.

id-aes256-CBC; tailles de clés: 256.

id-aes256-OFB; tailles de clés: 256.

id-aes256-CFB; tailles de clés: 256.

id-rc6-ecb; tailles de clés: 128.

id-rc6-cbc; tailles de clés: 128.

id-rc6-ofb; tailles de clés: 128.

id-rc6-cfb; tailles de clés: 128.

idea-ecb; tailles de clés: 128.

[1] En réalité, le nombre de bits utiles pour les clés est respectivement de 112 et 168 bits. Ils sont complétés par des bits de remplissage pour obtenir des clés de taille 128 et 192 bits.

idea-cbc; tailles de clés: 128.

idea-ofb; tailles de clés: 128.

idea-cfb; tailles de clés: 128.

- Cryptographie asymétrique (sans restriction sur la taille des clés) :

RSA (avec padding)

Les applications utilisant le mécanisme de chiffrement "RSAES-PKCS1-v1_5" doivent se limiter à des utilisations ne facilitant pas l'exploitation de leurs vulnérabilités.

DSA (avec padding)

Les ressources externes fournissent les fonctions de cryptographie asymétrique dans le cas de l'utilisation de clefs privées (chiffrement pour création de signature, déchiffrement pour déchiffrement de clef de session).

D'autres algorithmes sont possibles lors de l'utilisation d'APPLATOO mais ils sont exclus de cette évaluation.

3. ENVIRONNEMENT DE SECURITE DE LA TOE

Ce chapitre constitue l'analyse de risques de l'environnement opérationnel cible de la TOE.

Il peut s'avérer nécessaire de faire des hypothèses pour assurer qu'une menace potentielle n'est pas appropriée à l'environnement de sécurité de la TOE (par exemple, on ne traitera pas les menaces relatives à l'environnement physique de la TOE).

C'est dans ce chapitre que l'on présente les biens qui doivent être protégés, les agents menaçants et les méthodes d'attaque contre lesquelles les biens doivent être protégés.

On y décrit également les contraintes techniques et organisationnelles qui ont un impact sur la TOE et son environnement.

3.1. MNEMONIQUES

H.identifiant : désigne une hypothèse d'utilisation sûre identifiée par identifiant.

HR.Identifiant : désigne une hypothèse d'utilisation sûre sur les ressources externes identifiée par identifiant.

P.identifiant : désigne un élément de politique de sécurité de l'organisation identifié par identifiant.

O.identifiant : désigne un objectif de sécurité identifié par identifiant.

OE.identifiant : désigne un objectif de sécurité sur l'environnement identifié par identifiant.

M.identifiant : désigne une menace identifiée par identifiant.

A.identifiant : désigne un acteur identifié par identifiant.

3.2. ACTEURS

Le développeur de briques de sécurité (A.DéveloppeurBrique)

Le développeur de la TOE est responsable du code des fonctions rendues par la TOE. Il est également en charge de la gestion en versions des éléments de la TOE, de la distribution des versions successives aux utilisateurs, de l'information, auprès des utilisateurs, lors de la découverte de problèmes (bugs, problèmes de sécurité).

Le développeur de service (A.DéveloppeurService)

Le développeur de service va utiliser les fonctions rendues par la TOE pour développer des services sécurisés.

L'usager final (A.UsagerFinal)

C'est un utilisateur qui a accès à la TOE de manière indirecte, i.e., à travers les applications utilisatrices auxquelles il accède. Il utilise la TOE en son nom ou au nom d'une personne/d'une entité qu'il représente. Il peut être signataire lorsqu'un des services qu'il utilise fait appel aux fonctions de signature de la TOE.

Le service sécurisé (A.Service)

C'est un utilisateur qui accède à la TOE de manière directe. Il invoque des fonctions mises à disposition par la TOE en ayant précisé des paramètres d'appel spécifiques.

L'administrateur de sécurité (A.Admin)

L'administrateur de sécurité de la TOE est responsable de la mise en œuvre de la politique de sécurité de la TOE. Il peut être confondu avec l'administrateur de sécurité du SI incluant la TOE.

L'administrateur du service (A.AdminServ)

L'administrateur de sécurité du service est responsable de la politique de sécurité du service qu'il fournit aux utilisateurs. Il peut être confondu avec l'administrateur de sécurité du SI de l'application. Il sera le contact direct des utilisateurs pour tout problème de sécurité. Il pourra faire appel, lorsqu'il le jugera nécessaire à l'administrateur de sécurité de la TOE.

On peut distinguer, parmi ces acteurs, les acteurs directs et les acteurs indirects :

- Acteurs directs : le développeur de briques de sécurité, le développeur de service, le service sécurisé.
- Acteurs indirects : l'administrateur de sécurité, l'administrateur de service sécurisé, l'usager final.

3.3. BIENS SENSIBLES A PROTEGER

La TOE se doit de protéger les informations sensibles appelées biens. On distingue les biens indirects et les biens directs.

Les **biens directs** sont les données sécurisées à l'aide de la TOE (message, messages signés, requêtes d'horodatage, jetons d'horodatage). Ils doivent être protégés en intégrité et/ou en confidentialité.

Les **biens indirects** sont les informations nécessaires à la protection des biens directs. Il s'agit de l'ensemble des variables cryptographiques : biens indirects secrets (clés privées) et les biens indirects publics (certificats et clés publiques).

- ✓ **SCD** - Données de création de signature : clés privées utilisées pour achever l'opération de signature (la confidentialité de cette donnée doit être maintenue).

- ✓ **SVD** - Données de vérification de signature : clé publique associée à la donnée de création de signature et utilisée pour vérifier une signature électronique ou un jeton d'horodatage (leur intégrité doit être garantie).

- ✓ **DTBS** - Données à signer (l'intégrité de cette donnée doit être garantie).

- ✓ **DTBSR** - Représentation des données à signer (créée par APPLATOO) (l'intégrité de cette donnée doit être garantie).

- ✓ **CD** - Données de chiffrement : clé publique permettant le chiffrement de la *clef de session*. (l'intégrité de cette donnée doit être garantie).

- ✓ **CSD** - Données de session : clé de session permettant le *chiffrement ou le déchiffrement symétrique* d'un document. (la confidentialité et l'intégrité de cette donnée doivent être maintenues)

- ✓ **DD** - Données de déchiffrement : clé privée permettant le déchiffrement de la *clef de session*. (la confidentialité de cette donnée doit être maintenue)

- ✓ **DS** – Données sensibles : Données à chiffrer ou données déchiffrées (la confidentialité et l'intégrité de ces données doivent être garanties).

- ✓ **ES** - Signature électronique (ou donnée signée) : c'est la donnée obtenue après signature. Son intégrité doit être garantie par la TOE.

Remarque : LA TOE s'appuie sur des biens protégés par son environnement : DD, SCD.

Les biens directs sont alors :

- DTBS, DTBSR, DS, ES, CSD.

Les biens indirects son les suivants :

- SCD, SVD, CD, DD.

Bilan des données manipulées par la TOE et son environnement :

Biens à manipuler	**Par la TOE**	**Par l'environnement**
SCD	Non	Oui (dans les ressources externes)
SVD	Oui	Oui
DTBS	Oui	Oui
DTBSR	Oui	Non
CD	Oui	Oui
CSD	Oui	Non
DD	Non	Oui (dans les ressources externes)
DS	Oui	Oui
ES	Oui	Oui

3.4. TYPOLOGIE DES ATTAQUANTS

Les acteurs pouvant nuire à la TOE sont les suivants (les catégories d'acteurs auxquelles ils appartiennent sont données entre parenthèses):

- ✓ Des utilisateurs autorisés mais pouvant commettre une erreur par inadvertance, (A.UsagerFinal)

- ✓ Des développeurs indélicats pouvant insérer sciemment des portes dérobées, des portions de codes malicieux dans le code qu'ils produisent (permettant par exemple la récupération de biens à protéger), (A.DéveloppeurBrique, A.DéveloppeurService)

- ✓ Des personnes ou des machines qui disposent de moyens d'investigation et d'action sur la TOE ou sur le service sécurisé ou sur toute composante de l'architecture (poste utilisateur, réseaux locaux, etc.).(A.Admin, A.AdminServ)

Les deux dernières catégories peuvent être considérées comme des attaquants. Leur objectif est d'accéder à l'information sensible, alors que la première catégorie n'a pas d'objectif précis. Dans le contexte d'évaluation, on ne considère que des attaquants ayant un potentiel élémentaire (connaissances techniques…).

3.5. CONTRIBUTION A UN SYSTEME DE SIGNATURE ELECTRONIQUE (ET DE VERIFICATION D'HORODATAGE)

3.5.1. PROBLEMATIQUE DE SECURITE PRINCIPALE

P.CréationSignature – La TOE doit offrir aux applications utilisatrices la capacité de créer une signature électronique comme garantie de la validité des données à signer.

P.VérificationSignature – La TOE doit offrir aux applications utilisatrices la capacité de vérifier une signature électronique par rapport à un document donné, à un certificat et à la chaîne de confiance de ce certificat.

Note : P.VérificationSignature couvre aussi la vérification d'horodatage qui n'est qu'un cas particulier de vérification de signature associée à une vérification de format.

3.5.2. MENACES PORTANT SUR LES FONCTIONS DE SIGNATURE

M.ModificationDTBS – Un attaquant modifie la donnée à signer entre la réception par la TOE et la création de sa représentation. Ainsi la donnée envoyée à signer ne correspond pas à celle que le signataire pensait signer. La vérification de signature devient impossible alors que la signature semble avoir été effectuée.

M.ModificationDTBSR – Un attaquant modifie la représentation de la donnée à signer entre sa génération et les ressources cryptographiques externes. La représentation de la donnée à signer est modifiée et la signature est alors non valide.

M.MauvaisAiguillageDTBSR – La représentation de la donnée à signer est aiguillée vers un mauvais dispositif de signature. La signature générée ne sera pas exploitable par la suite.

Ces Menaces remettent en cause la confiance même dans le processus de signature électronique.

3.6. CONTRIBUTION A UNE POLITIQUE DE CHIFFREMENT

3.6.1. PROBLEMATIQUE DE SECURITE PRINCIPALE

P.Chiffrement – La TOE doit offrir aux applications utilisatrices la capacité de chiffrer un document à partir du certificat du destinataire.

P.Déchiffrement – La TOE doit offrir aux applications utilisatrices la capacité de déchiffrer un document.

3.6.2. MENACES PORTANT SUR LES FONCTIONS DE CHIFFREMENT

M.ModificationDS – Un attaquant modifie le DS entre l'interface haute de la TOE et les ressources cryptographiques de *chiffrement symétrique* de la TOE. Ainsi la donnée envoyée à chiffrer ne correspond pas à celle que l'émetteur pensait chiffrer ou la donnée renvoyée au service n'est pas valide.

M.ModificationCSD – Un attaquant modifie la clef de session entre le *chiffrement symétrique* de la TOE et les ressources cryptographiques externes en vue du chiffrement asymétrique de cette dernière. Ainsi la qualité du chiffrement symétrique peut être altérée ou le document peut devenir indéchiffrable.

M.MauvaisAiguillageCSD – La CSD est aiguillée vers un mauvais dispositif de chiffrement (qui la divulgue par exemple).

Ces menaces remettent en cause le processus de chiffrement et la non divulgation des données à chiffrer.

3.7. HYPOTHESES

H.IntégritéCodeEnvironnement – L'intégrité du code d'APPLATOO est préservée dans l'environnement d'exploitation aussi bien du développeur d'applications utilisatrices que chez l'utilisateur final. Un mécanisme de signature d'archives garantit l'intégrité de son contenu.

Commentaire : Cette hypothèse permet de pérenniser la conformité des fonctions de sécurité de la TOE après sa distribution au développeur de services.

H.CertificatConfiance – Les certificats de confiance présents sur le poste (et qui permettent de remonter une chaîne de confiance) sont issus de chaînes de confiance sûres (voir annexe).

Commentaire : Cette hypothèse permet que tout certificat présent sur le poste soit l'aboutissement d'une chaîne sûre. Ainsi on peut limiter le parcours des chaînes de confiance et s'arrêter dès qu'un certificat d'un magasin du poste est trouvé ; on est alors sur que la chaîne de confiance est sûre. Ce cas précis correspond en fait à l'ajout de certificat dans les magasins manuellement par l'utilisateur afin de leur accorder explicitement sa confiance.

HR.ChiffrementAsymétrique – Les ressources cryptographiques utilisées permettent d'effectuer un *chiffrement asymétrique* dont la nature ne permet pas de révéler la clef privée de chiffrement.

Commentaire : Cette hypothèse permet de contrer les menaces sur un mauvais chiffrement asymétrique qui permettrait la récupération de la clef privée.

HR.DéchiffrementAsymétrique – Les ressources cryptographiques utilisées permettent d'effectuer un *déchiffrement asymétrique* dont la nature ne permet pas de révéler la clef privée de déchiffrement.

Commentaire : Cette hypothèse permet de contrer les menaces sur un mauvais déchiffrement asymétrique qui permettrait la récupération de la clef privée.

HR.ConfidentialitéIntégrité – Les ressources cryptographiques utilisées assurent l'intégrité et la confidentialité des clefs privées dans les ressources externes.

Commentaire : Cette hypothèse permet de contrer les menaces sur une divulgation de la clef privée car mal protégée par les ressources externes.

HR.LimitationAccés – Les mesures de sécurité des ressources cryptographiques utilisées et de leur environnement permettent de limiter l'accès aux services de chiffrement/déchiffrement asymétriques aux utilisateurs et entités TI (serveurs, clients…) autorisées.

Commentaire : Cette hypothèse permet de contrer les menaces sur une utilisation des ressources externes par une personne non autorisée.

HR.ConfianceRessources – La confiance dans les mesures de sécurité des ressources et de leur environnement est au moins équivalente à un niveau deux étoiles PRI. (Voir [PRI])

Commentaire : Cette hypothèse permet de contrer les menaces venant d'une mauvaise qualité des ressources externes.

H.ConfidentialitéExécution – Les services de la plate-forme sur laquelle la TOE s'exécute contrôlent l'accès en lecture et écriture en zone mémoire où sont stockés les DS et CSD, et ils contrôlent l'accès en écriture en zone mémoire où sont stockés les DTBS et DTBSR.

Commentaire : Cette hypothèse permet de contrer les menaces sur un accès à des zones mémoires sensibles durant l'exécution de la TOE.

H.EnvironnementUtilisation – Les mesures de sécurité de l'environnement d'utilisation garantissent l'intégrité et la confidentialité des données échangées entre la TOE et les ressources externes ainsi qu'entre la TOE et les différents magasins de certificats.

Commentaire : Cette hypothèse permet d'écarter les menaces sur une modification par l'environnement des données échangées avec la TOE en dehors du périmètre de la TOE.

4. OBJECTIFS DE SECURITE

Ce chapitre identifie les objectifs de sécurité de la TOE et de son environnement. Les objectifs de sécurité contrent les menaces identifiées et prennent en compte les politiques de sécurité organisationnelles et les hypothèses identifiées.

4.1. OBJECTIFS DE SECURITE POUR LA TOE

4.1.1. PROBLEMATIQUE DE SECURITE PRINCIPALE

O.ChiffrementSymétrique : La TOE doit pouvoir *chiffrer symétriquement* un document à partir d'une *clef de session* donnée.

Argumentaire : O.ChiffrementSymétrique vise à permettre le chiffrement d'un fichier à partir d'une clef de session d'APPLATOO. En ce sens, il participe à la réalisation de la politique P.Chiffrement.

O.DéchiffrementSymétrique : La TOE doit pouvoir déchiffrer symétriquement un document à partir d'une *clef de session* donnée.

Argumentaire : O.DéchiffrementSymétrique vise à permettre le déchiffrement d'un fichier à partir d'une clef de session à APPLATOO. En ce sens, il participe à la réalisation de la politique P.Déchiffrement.

O.VérificationSignature : La TOE doit permettre de vérifier que la signature électronique d'un document donné correspond à la clef publique du certificat donné.

Argumentaire : O.VérificationSignature vise à éliminer les erreurs d'accès et d'utilisation des différentes fonctions cryptographiques externe à APPLATOO. En ce sens, il est dédié à la réalisation de la politique P.VérificationSignature.

4.1.2. OBJECTIFS DE SOUTIEN

O.RoutagePilotageFonctionsCryptographiques : La TOE doit sélectionner les ressources adaptées à la création de signature des DTBSR et aux déchiffrements de *clefs de session* puis piloter l'enchaînement des opérations cryptographiques nécessaires.

Argumentaire : O.RoutagePilotageFonctionsCryptographiques vise à éliminer les erreurs d'accès et d'utilisation des différentes fonctions cryptographiques externes à APPLATOO. En ce sens, il participe à la réalisation de la politique P.Déchiffrement et à la politique P.CréationSignature. Il permet également de contrer en partie les menaces M.MauvaisAiguillageCSD, M.ModificationDS, M.ModificationCSD, M.MauvaisAiguillageDTBSR, M.ModificationDTBS et M.ModificationDTBSR. Il contre en fait ces menaces dans l'enceinte de la TOE en assurant le bon routage vers les fonctions cryptographiques choisies et l'enchaînement correct des opérations demandées. Des opérations mal effectuées pourraient amener à une modification des biens sensibles.

O.RoutagePilotageMagasinsCertificats : La TOE doit sélectionner les magasins de certificats utilisés pour la validation de la *chaîne de confiance* puis piloter l'enchaînement des opérations nécessaires aux différentes vérifications.

Argumentaire : O.RoutagePilotageMagasinsCertificats vise à éliminer les erreurs d'accès et d'utilisation des différents magasins de certificats externes à APPLATOO. En ce sens, il participe à la réalisation des politiques P.Déchiffrement, P.Chiffrement, P.VérificationSignature, P.CréationSignature.

O.ChiffrementAsymétrique : La TOE doit pouvoir chiffrer asymétriquement un objet à partir d'une clef donnée. Ce chiffrement s'effectue sur une *clef de session*.

Argumentaire : O.ChiffrementAsymétrique vise à permettre le chiffrement asymétrique d'un objet. En ce sens, il participe à la réalisation de P.Chiffrement (en permettant le chiffrement d'une clef de session).

O.DéchiffrementAsymétrique : La TOE doit pouvoir déchiffrer asymétriquement un objet à partir d'une clef donnée. Ce déchiffrement s'effectue sur un *condensat*.

Argumentaire : O.DéchiffrementAsymétrique vise à permettre le déchiffrement asymétrique d'un objet. En ce sens, il participe à la réalisation de P.VérificationSignature en permettant le déchiffrement du condensat.

O.ValiditéCertificat : La TOE doit permettre de vérifier qu'un certificat est valide (non révoqué, non périmé, non prématuré), sa conformité à une politique de signature ou de chiffrement (*key usage*, utilisateur…) et la validité de sa *chaîne de confiance*.

Argumentaire : O.ValiditéCertificat vise à permettre à APPLATOO de vérifier la validité et les paramètres d'un certificat. En ce sens, il participe à la réalisation des la politique P.Déchiffrement, P.Chiffrement, P.VérificationSignature, P.CréationSignature. Concernant la chaîne de confiance, plus précisément, O.ValiditéCertificat permet de vérifier que la chaîne se termine par un certificat présent dans le magasin de certificat de confiance. OE.CertificatConfiance permet d'assurer que cette condition est suffisante pour que la chaîne soit sûre (en supposant que cette chaîne se poursuit jusqu'à un certificat d'Autorité de Certification).

O.GénérationClefsSession : La TOE doit pouvoir générer des *clefs de session* en vue du *chiffrement symétrique* d'un document.

Argumentaire : O.GénérationClefsSession vise à permettre à APPLATOO de créer des clefs de sessions en vue de session de chiffrement symétrique. Il permet donc la réalisation de la politique P.Chiffrement.

O.GénérationCondensats : La TOE doit pouvoir générer un *condensat* d'un document en vue de la signature du document.

Argumentaire : O.GénérationCondensats vise à permettre à APPLATOO de créer des condensats de fichiers qui sont alors des images de ces derniers. Il permet donc la réalisation des politiques P.CréationSignature et P.VérificationSignature.

4.1.3. OBJECTIFS POUR LA TOE GARANTIS PAR SON ENVIRONNEMENT DE DEVELOPPEMENT

O.QualitéDocTOE : La qualité des documents de développement est contrôlée spécifiquement sous l'angle de la sécurité de l'installation et du démarrage, de la compréhension correcte par les développeurs de service sécurisé du comportement des fonctions de sécurité et de la prévention des risques d'utilisation impropre de la TOE.

Argumentaire : O.QualitéDocTOE vise à éliminer les cas d'erreurs de mise en œuvre de la TOE. Il soutient tous les objectifs pour garantir une bonne utilisation de la TOE. Il est couvert par les composants d'assurance sécurité AGD_ADM.1, ADO_IGS.1 et AVA_MSU.1.

O.ConformitéFonctions : Les procédures de l'environnement de développement garantissent que les fonctions de sécurité de la TOE sont développées conformément à leurs spécifications.

Argumentaire : O.ConformitéFonctions vise à éliminer les erreurs d'implémentation des fonctions de sécurité de la TOE vis-à-vis des objectifs de sécurité spécifiés pour ces fonctions. En ce sens, il soutient tous ces objectifs pour contrer les menaces et satisfaire les règles de la politique de sécurité.

O.EfficacitéFonctions : Les fonctions de sécurité de la TOE et leurs documents de développement sont contrôlées spécifiquement sous l'angle de l'existence de vulnérabilités permettant la réalisation de menaces et/ou d'infractions aux règles de la politique de sécurité.

Argumentaire : O.EfficacitéFonctions permet de confirmer, à travers un contrôle détaillé incluant des revues de documents de développement et des tests de la TOE, que les fonctions de sécurité spécifiées pour la TOE répondent bien à la problématique de sécurité énoncée. En ce sens, il soutient tous les objectifs de sécurité spécifiés pour ces fonctions pour contrer les menaces et satisfaire les règles de la politique de sécurité.

O.IntégritéTOE : Les procédures de l'environnement de développement préviennent le risque de modification accidentelle ou intentionnelle de la TOE au cours de son cycle de développement et jusqu'à sa livraison sur les sites d'exploitation.

Argumentaire : O.IntégritéTOE garantit que la TOE qui parvient aux sites d'exploitation fournit effectivement les fonctions de sécurité prévues. En ce sens, il soutient tous les objectifs de sécurité spécifiés pour ces fonctions pour contrer les menaces et satisfaire les règles de la politique de sécurité.

O.ConfidentialitéTOE : Les procédures de l'environnement de développement préviennent le risque de divulgation d'informations sensibles concernant le développement de la TOE.

Argumentaire : O.ConfidentialitéTOE rend plus difficile aux attaquants la mise en œuvre de vulnérabilités éventuelles de la TOE nécessitant la connaissance d'informations sensibles, dont on peut supposer qu'elles pourraient avoir un impact critique sur la sécurité. O.ConfidentialitéTOE soutient donc tous les objectifs de sécurité spécifiés pour les fonctions de sécurité pour contrer les menaces et satisfaire les règles de la politique de sécurité.

4.2. OBJECTIFS DE SECURITE POUR L'ENVIRONNEMENT DE LA TOE

OE.IntégritéCodeEnvironnement: L'environnement d'utilisation doit assurer l'intégrité du code de la TOE par un mécanisme technique.

Argumentaire : OE.IntégritéCodeEnvironnement couvre H.IntégritéCodeEnvironnement. OE.IntégritéCodeEnvironnement garantit que les fonctions de sécurité sont préservées dans l'environnement d'utilisation de la TOE. En ce sens, il soutient tous les objectifs de sécurité spécifiés pour ces fonctions (4.1.1 et 4.1.2) pour contrer les menaces et satisfaire les règles de la politique de sécurité dans l'environnement d'exploitation.

OE.CertificatConfiance – Les certificats de confiance présents sur le poste (et qui permettent de remonter une chaîne de confiance) doivent être issus de *chaînes sûres*.

Argumentaire : OE.CertificatConfiance couvre H.CertificatConfiance. OE.CertificatConfiance soutient O.ValiditéCertificat pour permettre la vérification de la chaîne de confiance de certificats.

OE.ChiffrementAsymétriqueEnvironnement – Les ressources cryptographiques utilisées doivent permettre d'effectuer un *chiffrement asymétrique* dont la nature ne permet pas de révéler la clef privée de chiffrement. Ce chiffrement s'effectue sur un *condensat*.

Argumentaire : OE.ChiffrementAsymétriqueEnvironnement couvre HR.ChiffrementAsymétrique. OE.ChiffrementAsymétrique participe à la réalisation des politiques P.CréationSignature. Il permet de créer la signature d'une donnée. La non-divulgation de la clé privée permet de garantir la validité de la signature.

OE.DéchiffrementAsymétriqueEnvironnement – Les ressources cryptographiques utilisées doivent permettre d'effectuer un *déchiffrement asymétrique* dont la nature ne permet pas de révéler la clef privée de déchiffrement. Ce déchiffrement s'effectue sur une *clef de session*.

Argumentaire : OE.DéchiffrementAsymétriqueEnvironnement couvre HR.DéchiffrementAsymétrique. OE.DéchiffrementAsymétrique participe à la réalisation des politiques P.Déchiffrement. Il permet de réaliser le déchiffrement symétrique du document à l'aide de la clé qu'il permet d'obtenir. La non-divulgation de la clé privée permet de garantir la non-divulgation du document chiffré.

OE.ConfidentialitéIntégrité – Les ressources cryptographiques utilisées doivent assurer l'intégrité et la confidentialité des clefs privées.

Argumentaire : OE.ConfidentialitéIntégrité couvre HR.ConfidentialitéIntégrité. OE.ConfidentialitéIntégrité participe à la réalisation de P.CréationSignature et P.Déchiffrement. En ce sens il empêche la divulgation des secrets et aide donc à la non-répudiation des signatures et à la non divulgation des documents chiffrés.

OE.LimitationAccés – Les mesures de sécurité des ressources cryptographiques utilisées et de son environnement doivent permettre de limiter l'accès aux services de *chiffrement asymétrique* aux utilisateurs et entités TI autorisées.

Argumentaire : OE.LimitationAccés couvre HR.LimitationAccés. OE.LimitationAccés participe à la réalisation de P.CréationSignature et P.Déchiffrement. En ce sens il empêche la divulgation des secrets et aide donc à la non répudiation des signatures et à la non divulgation des documents chiffrés.

OE.ConfianceRessources – La confiance dans les mesures de sécurité des ressources et de leur environnement doit être au moins équivalente à un niveau deux étoiles PRI. (Voir [PRI])

Argumentaire : OE.ConfianceRessources couvre HR.ConfianceRessources. OE.ConfianceRessources soutient l'ensemble des objectifs OE.ChiffrementAsymétrique, OE.ConfidentialitéIntégrité et OE.LimitationAccés.

OE.ConfidentialitéExécution – Les services de la plate-forme sur laquelle la TOE s'exécute contrôlent l'accès en lecture et écriture en zone mémoire où sont stockés les DS et CSD, et ils contrôlent l'accès en écriture en zone mémoire où sont stockés les DTBS et DTBSR.

Argumentaire : OE.ConfidentialitéExécution couvre H.ConfidentialitéExécution. OE.ConfidentialitéExécution participe à la réalisation des politiques P.Vérificationsignature, P.CréationSignature P.Chiffrement et P.Déchiffrement. En ce sens, il permet en particulier de préserver les données sensibles utilisées pendant les opérations menées par la TOE, il assure donc la validité des signatures effectuées (en protégeant l'accès au DTBS et DTBSR) la non-divulgation des CSD et DS. Il contre en partie les menaces M.ModificationDS, M.ModificationCSD, M.Modification DTBS et M.Modification DTBSR. En effet, ces menaces sont contrés par les services de la plate-forme qui n'autorisent pas l'accès à ces données sensibles aux autres services (autres processus aux niveau de l'OS et autres threads au niveau de la machine virtuelle). L'environnement assure le cloisonnement des différents services pour ces données.

OE.EnvironnementUtilisation – Les mesures de sécurité de l'environnement d'utilisation garantissent l'intégrité et la confidentialité des données échangées entre la TOE et les ressources externes ainsi qu'entre la TOE et les différents magasins de certificats.

Argumentaire : OE.EnvironnementUtilisation couvre H.EnvironnementUtilisation. OE.EnvironnementUtilisation soutient les objectifs de la TOE O.RoutagePilotageFonctionsCryptographiques et O.RoutagePilotageMagasinsCertificats. En ce sens, il participe à la réalisation des politiques P.Vérificationsignature, P.CréationSignature P.Chiffrement et P.Déchiffrement. Il contre en partie les menaces M.ModificationCSD et M.Modification DTBS. En

effet, cet objectif permet de contrer les menaces portant sur CSD et DTBS une fois que ces données sont sorties du périmètre de la TOE, c'est-à-dire une fois qu'elles ont dépassé le pont JNI d'APPLATOO.

4.3. BILAN DE L'ARGUMENTAIRE

	O.ChiffrementSymétrique	O.DéchiffrementSymétrique	O.VérificationSignature	O.RoutagePilotageFonctionsCryptogra	O.RoutagePilotageMagasinsCertificats	O.ChiffrementAsymétrique	O.DéchiffrementAsymétrique	O.ValiditéCertificat	O.GénérationClefsSession	O.GénérationCondensats	O.QualitéDocTOE	O.ConformitéFonctions	O.EfficacitéFonctions	O.IntégritéTOE	O.ConfidentialitéTOE	OE.IntégritéCodeEnvironnement	OE.CertificatConfiance	OE.ChiffrementAsymétrique	OE.DéchiffrementAsymétrique	OE.ConfidentialitéIntégrité	OE.LimitationAccés	OE.ConfianceRessources	OE.EnvironnementUtilisation	OE.ConfidentialitéExécution
P.CréationSignature				S	S			S		S	S	S	S	S	S	S	S	X		S	S	S	S	S
P.VérificationSignature			X		S		S	S			S	S	S	S	S	S	S						S	S
M.ModificationDTBS				X							S	S	S	S	S	S								S
M.ModificationDTBSR				X							S	S	S	S	S	S							S	S
M.MauvaisAiguillageDTBSR				X							S	S	S	S	S	S							S	
P.Chiffrement	X				S	S		S	S		S	S	S	S	S	S	S						S	S
P.Déchiffrement		X		S	S			S			S	S	S	S	S	S	S		X	S	S	S	S	S
M.ModificationDS				X							S	S	S	S	S	S								S
M.ModificationCSD				X							S	S	S	S	S	S							S	S
M.MauvaisAiguillageCSD				X							S	S	S	S	S	S							S	
H.IntégritéCodeEnvironnement																X								
H.CertificatConfiance																	X							
HR.Chiffrementasymétrique																		X						
HR.Déchiffrementasymétrique																			X					
HR.ConfidentialitéIntégrité																				X				
HR.LimitationAccés																					X			
HR.ConfianceRessources																						X		
H.EnvironnementUtilisation																							X	
H.ConfidentialitéExécution																								X

X : réalise/contre
S : soutient

5. EXIGENCES DE SECURITE

Ce chapitre précise les exigences fonctionnelles de la cible d'évaluation et les exigences d'assurance des TI de la cible d'évaluation.

5.1. EXIGENCES DE SECURITE DES TI DE LA TOE

Format des étiquettes des exigences de sécurité :

Les exigences fonctionnelles de sécurité ont des étiquettes au format suivant :

FCC_FFF.composan*t*.<*itération*.>*n*

- F*CC* est le trigramme de la classe ;
- *FFF* est le trigramme de la famille ;
- *composant* est l'identifiant du composant : soit un numéro pour les composants extraits de [CC-02], soit un trigramme pour les exigences de sécurité explicitement énoncées ;
- *itération* est une étiquette permettant d'identifier les différentes itérations d'un même composant à l'intérieur d'un même ensemble fonctionnel ;
- *n* est le numéro d'élément.

En plus des quatre types d'opérations définies dans les Critères Communs (cf. [CC-01], § 4.4.1.3, p. 26), deux types supplémentaires de modification du texte original des exigences de sécurité des TI ont été introduites :

- Le raffinement systématique : il s'agit d'un raffinement effectué de manière homogène sur tous les éléments d'un composant ;
- La mise en forme : il s'agit d'une transformation de la structure grammaticale d'un élément, de manière à le rendre plus facile à lire, ou à supprimer du texte inutile, mais qui ne change absolument pas le sens de l'élément. Cela correspond à la notion d'editorial refinement.

Les opérations ont été effectuées sur le texte anglais original des exigences de sécurité des TI, mais elles ont pour effet de remplacer ces termes anglais par des termes français, et/ou à ajouter des termes français à un patron original en anglais. Malgré leur difficulté d'emploi, ces exigences en « franglais » constituent en tout état de cause l'élément de preuve requis par l'élément ASE_REQ.1.1D, alors que les exigences énoncées à côté ne sont qu'une reformulation du contenu de cette section, fournie dans le but de faciliter la compréhension de l'énoncé des exigences de sécurité des TI.

Dans l'identification des opérations, les raffinements qui consistent à substituer un terme à un autre, les affectations et les sélections sont identifiés par le symbole « := ». Les raffinements qui consistent à rajouter du texte sont identifiés par le symbole « + ». Les mises en forme sont identifiées par le symbole « » pour les substitutions et « ⌫ » pour les suppressions.

Les itérations sont identifiables à l'aide des étiquettes, comme cela est expliqué au § 5.x.

Les exigences de sécurité des TI sont présentées sous la forme suivante :

- Pour chaque composant utilisé, les raffinements systématiques opérés sur les éléments de ce composant,
- Pour chaque élément du composant :
  - Le texte anglais original de l'élément, tel qu'extrait de [CC-02] ou [CC-03],
  - La liste des opérations effectuées sur l'élément.

Format des étiquettes des exigences d'assurance :

Les exigences d'assurance sécurité ont des étiquettes identiques à celles utilisées dans [CC-03].

5.1.1. EXIGENCES FONCTIONNELLES DE SECURITE DES TI DE LA TOE

FCS_COP.1.Asym.1 – Les fonctions de chiffrement/déchiffrement asymétrique de la TOE doivent effectuer **des opérations de chiffrement/déchiffrement asymétriques** conformément aux algorithmes cryptographiques **RSA et DSA** et avec des tailles de clefs **supérieures ou égales à 1024 bits** qui satisfont à ce qui suit **[RSA], [DSA]**.

Raffinement systématique	*The TSF :=* *Les fonctions de chiffrement/déchiffrement asymétrique de la TOE*

FCS_COP.1.1 *The TSF shall perform [assignment: list of cryptographic operations] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].*

Affectation	*list of cryptographic operations* := *des opérations de chiffrement/déchiffrement asymétriques*
Affectation	*cryptographic algorithm* := RSA et DSA
Affectation	*cryptographic key sizes :=* supérieures ou égales à 1024 bits
Affectation	*list of standards :=* [RSA], [DSA]

Note d'application :

[RSA] : La norme correspondante à RSA et aux divers modes de padding est définie par PKCS#1.

[DSA] : La norme est définie par FIPS PUB 186 : National Institute of Standards and Technology (NIST), FIPS Publication 186: Digital Signature Standard (DSS)*, 1994.*

Argumentaire : Cette exigence est dédiée au soutien des objectifs O.ChiffrementAsymétrique, O.DéchiffrementAsymétrique et O.VerificationSignature.

Les dépendances avec FDP_ITC.1, FCS_CKM.1 FCS_CKM.4 et FMT_MSA.2 ne sont pas applicables.

FCS_COP.1.Sym.1 – Les fonctions de chiffrement/déchiffrement symétrique de la TOE doivent effectuer **des opérations de chiffrement/déchiffrement symétrique** conformément

aux algorithmes cryptographiques **du tableau [Sym]** et avec des tailles de clefs **données dans le tableau [Sym]** qui satisfont aux standards **donnés dans le tableau [Sym]**.

Raffinement systématique	*The TSF :=* *Les fonctions de chiffrement/déchiffrement symétrique de la TOE*

FCS_COP.1.1 *The TSF shall perform [assignment: list of cryptographic operations] inaccordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].*

Affectation	*list of cryptographic operations* := des opérations de chiffrement/déchiffrement symétrique
Affectation	*cryptographic algorithm* := du tableau [Sym]
Affectation	*cryptographic key sizes :=* données dans le tableau [Sym]
Affectation	*list of standards :=* donnés dans le tableau [Sym]

[Sym] :

Référence Algorithme	**Taille de Clefs**	**Standard**
des128-ede3-CBC	128	FIPS 46-3
des192-ede3-CBC	192	FIPS 46-3
aes128-ECB	128	FIPS PUB 197
aes128-CBC	128	FIPS PUB 197
aes128-OFB	128	FIPS PUB 197
aes128-CFB	128	FIPS PUB 197
aes192-ECB	192	FIPS PUB 197
aes192-CBC	192	FIPS PUB 197
aes192-OFB	192	FIPS PUB 197
aes192-CFB	192	FIPS PUB 197
aes256-ECB	256	FIPS PUB 197
aes256-CBC	256	FIPS PUB 197
aes256-OFB	256	FIPS PUB 197
aes256-CFB	256	FIPS PUB 197
rc6128-ECB	128	RRSY98
rc6128-CBC	128	RRSY98
rc6128-OFB	128	RRSY98
rc6128-CFB	128	RRSY98

idea128-ECB	128	LMM92
idea128-CBC	128	LMM92
idea128-OFB	128	LMM92
idea128-CFB	128	LMM92

Note d'application :

Pour les modes CBC, une valeur initiale doit être imposée. Pour les modes ECB, les applications les utilisant doivent se limiter à des utilisations ne mettant pas à jour leurs vulnérabilités.

FIPS PUB 197 et FIPS 46-3 sont des publications du National Institute of Standards and Technology.

LMM92 : X. Lai, J.L. Massey and S. Murphy, Markov ciphers and differential cryptanalysis, Advances in Cryptology - Eurocrypt '91, Springer-Verlag (1992), 17-38.

RRSY98 : Ronald L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin, The RC6 Block Cipher (1998).

Argumentaire : Cette exigence est dédiée au soutien des objectifs O.ChiffrementSymétrique et O.DéchiffrementSymétrique.

Les dépendances avec FCS_CKM.4 et FMT_MSA.2 ne sont pas applicables.

FCS_CKM.1.Clefs.1 – Les fonctions de génération de clefs de session doivent générer des clefs cryptographiques conformément **à l'algorithme propriétaire spécifié dans [Clefs]** et à des tailles de clefs spécifiées **suivant l'algorithme de chiffrement symétrique utilisé**.

Raffinement systématique	*The TSF :* = *Les fonctions de génération de clefs de session*

FCS_CKM.1.1 *The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [assignment: cryptographic key generation algorithm] and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].*

Affectation	*cryptographic key generation algorithm* := *à l'algorithme propriétaire spécifié dans [Clefs]*
Affectation	*cryptographic key sizes*:= suivant l'algorithme de chiffrement symétrique utilisé
Affectation+mise en forme	*list of standards :*= aucun ⌫

[Clefs] : Les tailles des clés demandées sont vérifiées en fonction de l'algorithme.

La génération des clés est réalisée à partir d'un générateur de nombre pseudo-aléatoire configurable.

Un test sur la faiblesse de la clé est appliqué sur la valeur générée pour les algorithme DES et triple DES.

Argumentaire : Cette exigence est dédiée au soutien de l'objectif O.GénérationClefsSession.

Les dépendances avec FCS_CKM.4 et FMT_MSA.2 ne sont pas applicables.

FCS_COP.1.Condensats.1 – Les fonctions de génération de condensats doivent réaliser **la génération des condensats** de 160 bits conformément à l'algorithme **SHA-1** qui satisfait à **[SHA]**.

Raffinement systématique	*The TSF :=* Les fonctions de génération de condensats

FCS_COP.1.1 *The TSF shall perform [assignment: list of cryptographic operations] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].*

Affectation	*list of cryptographic operations* := la génération des condensats de 160 bits
Affectation	*cryptographic algorithm* := SHA-1
Affectation+mise en forme	*cryptographic key sizes :=* aucune (non applicable) ⌫
Affectation	*list of standards :=* [SHA]

Note d'application :

[SHA] : National Institute of Standards and Technology (NIST), Announcement of Weakness in the Secure Hash Standard, 1994.

Argumentaire : Cette exigence est dédiée à l'objectif O.GénérationCondensat.

Les dépendances avec FDP_ITC.1, FCS_CKM.1, FCS_CKM.4 et FMT_MSA.2 ne sont pas applicables.

FCS_CKM.3.RoutagePilotage.1 – Les fonctions cryptographiques doivent réaliser **les opérations de chiffrement déchiffrement externes et d'accès aux certificats** conformément à **la méthode d'accès spécifiée par l'application sécurisée**.

Raffinement systématique	*The TSF* := les fonctions cryptographiques

FCS_CKM.3.1 *The TSF shall perform [assignment: type of cryptographic key access] in accordance with a specified cryptographic key access method [assignment: cryptographic key access method] that meets the following: [assignment: list of standards].*

Affectation	*type of cryptographic key access* := les opérations de chiffrement déchiffrement externes et d'accès aux certificats
Affectation	*cryptographic key access method* := la méthode d'accès spécifiée par l'application sécurisée
Affectation + mise en forme	*list of standards :=* aucun ⌫

Argumentaire : Cette exigence est dédiée au soutien des objectifs O.RoutagePilotageFonctionsCryptographiques et O.RoutagePilotageMagasinsCertificats. Le pilotage des ressources externes est en pratique très proche de celui des magasins de certificats, les ressources externes sont en effet référencées par des certificats qui sont associées aux ressources privées qu'elles contiennent. Cette exigence couvre en fait l'accès et la manipulation des différents certificats (associés à des clefs privées ou publiques) par la TOE afin de réaliser les opérations cryptographiques demandées par le service.

Les dépendances avec FDP_ITC.1, FCS_CKM.1, FCS_CKM.4 et FMT_MSA.2 ne sont pas applicables.

FMT_MTD.3.ValiditéCertificat.1 – Les fonctions de vérification de la chaîne de confiance doivent garantir que seules des valeurs **valides** sont acceptées pour **les paramètres des certificats**. **Les conditions de validité des chaînes sont fixées par des filtres fixés par les fonctions**.

Raffinement systématique	*The TSF* := les fonctions de vérifications de la chaîne de confiance

FMT_MTD.3.1 The TSF shall ensure that only secure values are accepted for TSF data.

Raffinement	*secure* := valide
Raffinement	*TSF data* := les paramètres des certificats
Raffinement	+ Les conditions de validité des chaînes sont fixées par des filtres fixés par les fonctions.

Argumentaire : Cette exigence est dédiée au soutien de l'objectif O.ValiditéCertificat. Les filtres fixés par les fonctions permettent de sélectionner les certificats suivant leur paramètres, des contrôles de validité sont également possible sur les chaînes de confiance des certificats.

Les dépendances avec ADV_SPM.1 et FMT_MTD.1 ne sont pas applicables.

5.1.2. EXIGENCES D'ASSURANCE DES TI DE LA TOE

Le niveau d'assurance visé est EAL2 augmenté de plusieurs composantes d'assurance.
Le niveau EAL2+ correspond au niveau standard tel que défini par la DCSSI (cf. [QPS-std]).

Classe d'assurance	Composant(s) d'assurance
ASE	ASE_XXX.1 (cible de sécurité)
ACM	ACM_CAP.2
ADV	ADV_FSP.1, ADV_RCR.1, ADV_HLD.2, ADV_LLD.1, ADV_IMP.1
ADO	ADO_DEL.1, ADO_IGS.1
AGD	AGD_ADM.1, AGD_USR.1
ALC	ALC_DVS.1, ALC_FLR.3, ALC_TAT.1
ATE	ATE_COV.1, ATE_FUN.1, ATE_IND.2
AVA	AVA_MSU.1, AVA_SOF.1, AVA_VLA.2

5.1.2.1. EVALUATION D'UNE CIBLE DE SECURITE (ASE)

Fourniture de la cible de sécurité.

5.1.2.2. GESTION DE CONFIGURATION (ACM)

Eléments de configuration (ACM_CAP.2)

Cf. [CC-03], § 7.2, p. 70.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.IntégritéTOE.

5.1.2.3. DEVELOPPEMENT (ADV)

Spécifications fonctionnelles informelles (ADV_FSP.1)

Cf. [CC-03], § 9.1, p. 90.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.ConformitéFonctions.

Démonstration de correspondance informelle (ADV_RCR.1)

Cf. [CC-03], § 9.6, p. 112.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.ConformitéFonctions.

Conception de haut niveau – Identification des sous-systèmes dédiés à la sécurité (ADV_HLD.2)

Cf. [CC-03], § 9.2, p. 95.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.ConformitéFonctions.

Conception de bas niveau descriptive (ADV_LLD.1)

Pour le texte original, Cf. [CC-03], § 9.5, p. 108.

Tâches du développeur :

ADV_LLD.1.1D Le développeur doit fournir la conception de bas niveau **des fonctions cryptographiques**.

Contenu et présentation des éléments de preuve :

ADV_LLD.1.1C La présentation de la conception de bas niveau **des fonctions cryptographiques** doit être en style informel.

ADV_LLD.1.2C La conception de bas niveau **des fonctions cryptographiques** doit avoir une cohérence interne.

ADV_LLD.1.3C La conception de bas niveau **des fonctions cryptographiques** doit décrire **les fonctions cryptographiques** en termes de modules **cryptographiques**.

ADV_LLD.1.4C La conception de bas niveau **des fonctions cryptographiques** doit décrire le but de chaque module **cryptographique**.

ADV_LLD.1.5C La conception de bas niveau **des fonctions cryptographiques** doit définir les relations mutuelles entre les modules **cryptographiques** en termes de fonctionnalités de sécurité fournies et de dépendances vis-à-vis des autres modules.

ADV_LLD.1.6C La conception de bas niveau **des fonctions cryptographiques** doit décrire comment chaque **fonction cryptographique** est fournie.

ADV_LLD.1.7C La conception de bas niveau **des fonctions cryptographiques** doit identifier toutes les interfaces des modules **cryptographiques**.

ADV_LLD.1.8C La conception de bas niveau **des fonctions cryptographiques** doit identifier les interfaces des modules **cryptographiques** qui sont visibles de l'extérieur.

ADV_LLD.1.9C La conception de bas niveau **des fonctions cryptographiques** doit décrire le but et le mode d'utilisation de toutes les interfaces des modules **cryptographiques**, en fournissant, lorsque cela est approprié, les détails sur les effets, les exceptions et les messages d'erreur.

ADV_LLD.1.10C La conception de bas niveau des fonctions cryptographiques de soutien doit décrire la séparation de la TOE en modules **cryptographiques** et en autres modules.

Tâches de l'évaluateur :

ADV_LLD.1.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_LLD.1.2E L'évaluateur doit déterminer que la conception de bas niveau **des fonctions cryptographiques** est une instanciation correcte et complète des **exigences fonctionnelles de sécurité de la classe FCS** pour la TOE.

Raffinement systématique	*The TSF* := les fonctions cryptographiques de soutien

Raffinement systématique	*modules* + cryptographiques
Raffinement systématique	*TOE security functional requirements* := exigences fonctionnelles de sécurité de la classe FCS

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.ConformitéFonctions, en ce qui concerne spécifiquement les fonctions cryptographiques.

Sous-ensemble de l'implémentation de la TSF (ADV_IMP.1)

Pour le texte original, Cf. [CC-03], § 9.3, p. 100.

Tâches du développeur :

ADV_IMP.1.1D Le développeur doit fournir la représentation de l'implémentation **des fonctions cryptographiques**.

Contenu et présentation des éléments de preuve :

ADV_IMP.1.1C La représentation de l'implémentation **des fonctions cryptographiques** doit définir **les fonctions cryptographiques** d'une façon non ambiguë avec un niveau de détail suffisant pour qu'elle puisse être générée sans décision de conception supplémentaire.

ADV_IMP.1.2C La représentation de l'implémentation **des fonctions cryptographiques** doit avoir une cohérence interne.

Tâches de l'évaluateur :

ADV_IMP.1.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

ADV_IMP.1.2E L'évaluateur doit déterminer que le plus bas niveau de représentation de la TSF **des fonctions cryptographiques** fourni est une instantiation correcte et complète des **exigences fonctionnelles de sécurité de la classe FCS** pour la TOE.

Raffinement systématique	*The TSF* := les fonctions cryptographiques
Raffinement systématique	*TOE security functional requirements* := exigences fonctionnelles de sécurité de la classe FCS

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.ConformitéFonctions, en ce qui concerne spécifiquement les fonctions cryptographiques.

5.1.2.4. LIVRAISON ET EXPLOITATION (ADO)

Procédures de livraison (ADO_DEL.1)

Cf. [CC-03], § 8.1, p. 80.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.IntégritéTOE.

Procédures d'installation, de génération et de démarrage (ADO_IGS.1)

Cf. [CC-03], § 8.2, p. 82.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.QualitéDocTOE. Il couvre les aspects « sécurité de l'installation et du démarrage ».

5.1.2.5. GUIDES (AGD)

Guide de l'administrateur (AGD_ADM.1)

Cf. [CC-03], § 10.1, p. 119.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.QualitéDocTOE. Il couvre les aspects « compréhension correcte par les administrateurs du comportement des fonctions de sécurité ».

Guide de l'utilisateur (AGD_USR.1)

Cf. [CC-03], § 10.2, p. 120.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.QualitéDocTOE. Il couvre les aspects « compréhension correcte par les utilisateurs du comportement des fonctions de sécurité ».

5.1.2.6. SUPPORT AU CYCLE DE VIE (ALC)

Identification des mesures de sécurité (ALC_DVS.1)

Cf. [CC-03], § 11.1, p. 123.

Argumentaire : ce composant est dédié à la satisfaction des objectifs O.IntégritéTOE et O.ConfidentialitéTOE.

Evaluation de la correction d'anomalies (ALC_FLR.3)

Cf. [CC-03], Annexe A, § 11.2, p. 126.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.ConformitéFonctions.

Outils de développement bien définis (ALC_TAT.1)

Pour le texte original, Cf. [CC-03], § 11.4, p. 130.

Tâches du développeur :

ALC_TAT.1.1D Le développeur doit identifier les outils de développement utilisés pour **les fonctions cryptographiques** de la TOE.

ALC_TAT.1.2D Le développeur doit documenter les options dépendant de l'implémentation qui ont été choisies pour les outils de développement **des fonctions cryptographiques**.

Contenu et présentation des éléments de preuve :

ALC_TAT.1.1C Tous les outils de développement utilisés pour l'implémentation **des fonctions cryptographiques** doivent être bien définis.

ALC_TAT.1.2C La documentation relative aux outils de développement **des fonctions cryptographiques** doit définir sans ambiguïté la signification de toutes les instructions utilisées dans l'implémentation **des fonctions cryptographiques**.

ALC_TAT.1.3C La documentation relative aux outils de développement **des fonctions cryptographiques** doit définir sans ambiguïté la signification de toutes les options dépendant de l'implémentation.

Tâches de l'évaluateur :

ALC_TAT.1.1E L'évaluateur doit confirmer que les informations fournies satisfont à toutes les exigences relatives au contenu et à la présentation des éléments de preuve.

Raffinement systématique	*The TSF* := les fonctions cryptographiques

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.ConformitéFonctions, en ce qui concerne spécifiquement les fonctions cryptographiques.

5.1.2.7. TESTS (ATE)

Eléments de preuve de la couverture (ATE_COV.1)

Cf. [CC-03], § 12.1, p. 135.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.ConformitéFonctions.

Tests fonctionnels (ATE_FUN.1)

Cf. [CC-03], § 12.3, p. 143.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.ConformitéFonctions.

Tests indépendants – échantillonnage (ATE_IND.2)

Cf. [CC-03], § 12.4, p. 147.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.ConformitéFonctions.

5.1.2.8. ESTIMATION DES VULNERABILITES (AVA)

Examen des guides (AVA_MSU.1)

Cf. [CC-03], § 13.2, p. 155.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.QualitéDocTOE. Il couvre les aspects « prévention des risques d'utilisation impropre de la TOE ».

Evaluation de la résistance des fonctions de sécurité de la TOE (AVA_SOF.1)

Cf. [CC-03], § 13.3, p. 159.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.EfficacitéFonctions.

Analyse de vulnérabilités indépendantes (AVA_VLA.2)

Cf. [CC-03], § 13.4, p. 162.

Argumentaire : ce composant est dédié à la satisfaction de l'objectif O.EfficacitéFonctions.

5.1.3. BILAN DES EXIGENCES DE SECURITE DES TI DE LA TOE

Note : Les différentes clefs, certificats ou ressources externes utilisées par la TOE sont fournies par son environnement et ne sont pas modifiés par la TOE. Les différents paramètres d'un certificat (public ou liés à une ressource externe) sont, entre autres, le *key usage*, la date de validité, le possesseur, le type de la clef : algorithme associé, longueur… Ils sont utilisés par la TOE mais ne peuvent être modifiés.

5.1.3.1. TABLEAU DE SATISFACTION DES DEPENDANCES

Composant	*Dépendances*	*Satisfaction*
FCS_COP.1.Asym	FDP_ITC.1 | FCS_CKM.1	Non applicable, car la génération des clefs n'est pas à la charge de la TOE et la récupération des certificats est réalisée par la méthode d'accès aux certificats FCS_CKM.3.RoutagePilotage.

Composant	*Dépendances*	*Satisfaction*
	FCS_CKM.4	Non applicable, car la suppression d'un certificat d'un magasin ne nécessite pas de méthode particulière de destruction de son contenu.
	FMT_MSA.2	Non applicable, car les attributs de la clef (type de la clef, utilisateur, période de validité et *key usage*) ne peuvent pas être modifiés par la TOE.
FCS_COP.1.Sym	FDP_ITC.1 | FCS_CKM.1	Satisfait par FCS_CKM.1.Clefs
	FCS_CKM.4	Non applicable, la destruction des clefs est assurée par la machine virtuelle à l'aide du "Garbage Collector" qui détruit les objets qui ne sont plus référencés.
	FMT_MSA.2	Non applicable, car les attributs de la clef (type de la clef, utilisateur, période de validité et *key usage*) ne peuvent pas être modifiés par la TOE.
FCS_CKM.1.Clefs.1	FCS_CKM.2 | FCS_COP.1	Satisfait par FCS_COP.1.Sym
	FCS_CKM.4	Non applicable, la destruction des clefs est assurée par la machine virtuelle à l'aide du "Garbage Collector" qui détruit les objets qui ne sont plus référencés.
	FMT_MSA.2	Non applicable, car les attributs de la clef (type de la clef, utilisateur, période de validité et *key usage*) ne pas être modifiés par la TOE.
FCS_COP.1.Condensats	FDP_ITC.1 | FCS_CKM.1	Non applicable, car cet algorithme n'utilise pas de clef.
	FCS_CKM.4	Non applicable, car cet algorithme n'utilise pas de clef.
	FMT_MSA.2	Non applicable, car cet algorithme n'utilise pas de clef.

Composant	*Dépendances*	*Satisfaction*
FCS_CKM.3.RoutagePilotage	FDP_ITC.1 | FCS_CKM.1	Non applicable, car la génération des clefs n'est pas à la charge de la TOE et les données importées sont supposées sûres.
	FCS_CKM.4	Non applicable, car cette exigence ne spécifie pas l'accès à des clef mais à des fonctions de chiffrement.
	FMT_MSA.2	Non applicable, car la définition et le contrôle de ce qu'est une valeur sûre pour la méthode d'accès spécifiée par l'application sécurisée est en dehors de la portée de l'évaluation.
FMT_MTD.3.ValiditéCertificat	ADV_SPM.1	Non applicable, la politique de sécurité correspondant à cette fonction concerne en fait les filtres mis en place sur les certificats par le service sécurisé.
	FMT_MTD.1	Non applicable, les données ne sont pas administrées, car elles ne peuvent être modifiées. Elles sont juste consultées.

Composant	*Dépendances*	*Satisfaction*
ACM_CAP.2	Aucune	
ADO_DEL.1	Aucune	
ADO_IGS.1	AGD_ADM.1	Oui
AGD_ADM.1	ADV_FSP.1	Oui
AGD_USR.1	ADV_FSP.1	Oui
AVA_MSU.1	ADO_IGS.1, ADV_FSP.1, AGD_ADM.1, AGD_USR.1	Oui
AVA_SOF.1	ADV_FSP.1, ADV_HLD.1	Oui
AVA_VLA.2	ADV_FSP.1,ADV_HLD.2, ADV_IMP.1, ADV_LLD.1, AGD_ADM.1, AGD_USR.1	Oui
ADV_FSP.1	ADV_RCR.1	Oui
ADV_HLD.2	ADV_FSP.1, ADV_RCR.1	Oui
ADV_LLD.1	ADV_HLD.2, ADV_RCR.1	Oui
ADV_IMP.1	ADV_LLD.1, ADV_RCR.1, ALC_TAT.1	Oui
ADV_RCR.1	Aucune	
ALC_DVS.1	Aucune	
ALC_FLR.3	Aucune	
ALC_TAT.1	ADV_IMP.1	Oui
ATE_COV.1	ADV_FSP.1, ATE_FUN.1	Oui
ATE_FUN.1	Aucune	
ATE_IND.2	ADV_FSP.1, AGD_ADM.1, AGD_USR.1, ATE_FUN.1	Oui

5.1.3.2. SOUTIEN MUTUEL ET NON CONTRADICTION DES EXIGENCES

Toutes les dépendances sont satisfaites ou bien leur non-satisfaction a été justifiée. Les exigences de sécurité forment donc un ensemble qui se soutient mutuellement et ne présente pas de contradiction.

5.1.3.3. BILAN DE LA COUVERTURE OBJECTIF / EXIGENCES

	O.ChiffrementSymétrique	O.DéchiffrementSymétrique	O.VérificationSignature	O.RoutagePilotageFonctionsCryptographiques	O.RoutagePilotageMagasinsCertificats	O.ChiffrementAsymétrique	O.DéchiffrementAsymétrique	O.ValiditéCertificat	O.GénérationClefsSession	O.GénérationCondensats	O.QualitéDocTOE	O.ConformitéFonctions	O.EfficacitéFonctions	O.IntégritéTOE	O.ConfidentialitéTOE
FCS_COP.1.Asym			X			X	X								
FCS_COP.1.Sym	X	X													
FCS_CKM.1.Clefs									X						
FCS_COP.1.Condensats										X					
FCS_CKM.3.RoutagePilotage				X	X										
FMT_MTD.3.ValiditéCertificat								X							
ACM_CAP.2														X	
ADO_DEL.1														X	
ADO_IGS.1											X				
AGD_ADM.1											X				
AGD_USR.1											X				
AVA_MSU.1											X				
AVA_SOF.1													X		
AVA_VLA.2													X		
ADV_FSP.1												X			
ADV_HLD.2												X			
ADV_LLD.1												X			
ADV_IMP.1												X			
ADV_RCR.1												X			
ALC_DVS.1														X	X
ALC_FLR.3												X			
ALC_TAT.1												X			
ATE_COV.1												X			
ATE_FUN.1												X			
ATE_IND.2												X			

X : réalise

5.1.3.4. COHERENCE DE LA RESISTANCE DES FONCTIONS DE SECURITE AVEC LES OBJECTIFS DE SECURITE

Le niveau de résistance minimum des fonctions annoncé ('SOF-high') est supérieur au niveau maximum de moyens à la disposition des attaquants envisageable au cours de l'évaluation (potentiel d'attaque 'élémentaire'). Ce niveau est spécifié par le composant d'assurance sécurité AVA_VLA.2, qui soutient, à travers l'objectif O.EfficacitéFonctions, les objectifs de sécurité spécifiés pour les fonctions de sécurité de la TOE. Il n'y a donc pas d'incohérence entre le niveau de résistance minimum des fonctions annoncées et les objectifs de sécurité.

5.2. EXIGENCES DE SECURITE POUR L'ENVIRONNEMENT

Les formats utilisés sont les mêmes que ceux utilisés pour les exigences de sécurité des TI de la TOE au §5.1.

5.2.1. EXIGENCES FONCTIONNELLES DE SECURITE TI POUR L'ENVIRONNEMENT

FDP_ACF.1.ConfidentialitéExécution.1 – Les fonctions de sécurité de la machine virtuelle et du système d'exploitation doivent appliquer **la politique d'accès mémoire** aux **processus gérés par le système d'exploitation et les threads de la machine virtuelle** en se basant sur **l'identité du processus au niveau du système d'exploitation et l'identité du thread au niveau de la machine virtuelle**.

Raffinement systématique	*The TSF* := Les fonctions de sécurité de la machine virtuelle et du système d'exploitation

FDP_ACF.1.1 *The TSF shall enforce the [assignment: access control SFP] to objects based on [assignment: security attributes, named groups of security attributes].*

Affectation	*access control SFP* := la politique d'accès mémoire
Raffinement	*objects* processus gérés par le système d'exploitation et les threads de la machine virtuelle
Affectation	*security attributes, named groups of security attributes* := l'identité du processus au niveau du système d'exploitation et l'identité du thread au niveau de la machine virtuelle

Argumentaire : cette exigence est dédiée au soutien de l'objectif OE.ConfidentialitéExécution. Elle permet de contrôler l'origine des sujets manipulant les données à protéger.

FDP_ACF.1.ConfidentialitéExécution.2 – Les fonctions de sécurité de la machine virtuelle et du système d'exploitation doivent appliquer les règles suivantes pour déterminer si **un accès mémoire** entre **des processus du système d'exploitation et des threads de la machine virtuelle** est autorisé : **l'accès aux zones mémoires des DS et CSD en lecture et écriture doit être limité aux threads du service sécurisé, l'accès en lecture aux zones mémoires des DTBS et DTBSR doit être limité aux services sécurisés.**

Raffinement systématique	*The TSF* := Les fonctions de sécurité de la machine virtuelle et du système d'exploitation

FDP_ACF.1.2 *The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects].*

Raffinement	*operation* un accès mémoire
Raffinement	*subjects and controlled objects* processus du système d'exploitation et des threads de la machine virtuelle
Affectation	*rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects* := l'accès aux zones mémoires des DS et CSD en lecture et écriture doit être limité aux threads du service sécurisé, l'accès en lecture aux zones mémoires des DTBS et DTBSR doit être limité aux services sécurisés

Argumentaire : cette exigence est dédiée au soutien de l'objectif OE.ConfidentialitéExécution. Elle permet en effet de préciser les règles de contrôle d'accès aux données à protéger.

FDP_ACF.1.ConfidentialitéExécution.3 – Les fonctions de sécurité de la machine virtuelle et du système d'exploitation doivent autoriser explicitement l'accès de sujets à des objets en fonction des règles complémentaires suivantes.

Raffinement systématique	*The TSF* := Les fonctions de sécurité de la machine virtuelle et du système d'exploitation

FDP_ACF.1.2 *The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects].*

Affectation	*rules, based on security attributes, that explicitly authorise access of subjects to objects*°:= aucun ⌫

Argumentaire : cette exigence est sans objet.

FDP_ACF.1.ConfidentialitéExécution.4 – Les fonctions de sécurité de la machine virtuelle et du système d'exploitation doivent refuser explicitement l'accès de sujets à des objets**.**

Raffinement systématique	*The TSF* := Les fonctions de sécurité de la machine virtuelle et du système d'exploitation

FDP_ACF.1.2 *The TSF shall explicitly deny access of subjects to objects based on the [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects].*

Affectation	*assignment: rules, based on security attributes, that explicitly deny access of subjects to objects*:= aucun ⌫

Argumentaire : cette exigence est sans objet.

La dépendance avec FDP_ACC.1 est satisfaite par FDP_ACC.1.ConfidentialitéExécution.

La dépendance avec FMT_MSA.3 n'est pas applicable.

En effet la politique d'accès mémoire au niveau du service sécurisé pour tous les objets créés (DS, CSD, DTBS et DTBSR entre autres) dépend pour la machine virtuelle des attributs de ces objets qui sont définis par le code source du service sécurisé et de la TOE.

Le cloisonnement entre threads de la machine virtuelle est une propriété demandée à la machine virtuelle qui doit séparer ses différents threads et contrôler l'accès en fonction de leur identité. Dans un même service plusieurs threads peuvent cohabiter et communiquer entre eux. La machine virtuelle ne permet des échanges qu'entre les objets ayant des références entre eux (un objet ne peut accéder à un autre que s'il a une référence vers cet objet et le droit d'accès à cet objet); de même pour les threads. C'est pourquoi un autre service ne pourra pas accéder aux threads du service sécurisé. Au niveau des processus du système d'exploitation, la même propriété est demandée : un cloisonnement des différents espaces de chacun des processus et donc un contrôle en fonction de leur identité. Il n'y a donc pas d'initialisation statique d'attributs aussi bien à l'intérieur du service qu'au niveau des threads de la machine virtuelle ou des différents processus du système d'exploitation.

FDP_ACC.1.ConfidentialitéExécution.1 – Les fonctions de sécurité de la machine virtuelle doivent appliquer la **politique d'accès mémoire** aux **DS, CSD, DTBS et DTBSR du service sécurisé, aux threads de la machine virtuelle et aux processus du système d'exploitation au niveau des opérations de lecture et écriture.**

Raffinement systématique	*The TSF* := Les fonctions de sécurité de la machine virtuelle et du système d'exploitation

FDP_ACF.1.2 *The TSF shall enforce the [assignment: access control SFP] on [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP].*

Affectation	*assignment: access control SFP*:= politique d'accès mémoire
Affectation	*assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP*:= DS, CSD, DTBS et DTBSR du service sécurisé, aux threads de la machine virtuelle et aux processus du système d'exploitation au niveau des opérations de lecture et écriture

Argumentaire : cette exigence est dédiée au soutien de l'objectif OE.ConfidentialitéExécution. Elle définit les différents objets et opérations de la politique d'accès mémoire qui est dédiée à cet objectif.

FCS_COP.1.AsymRessources.1 – Les fonctions de chiffrement/déchiffrement asymétrique de l'environnement doivent s'exécuter conformément aux algorithmes cryptographiques **RSA et DSA** et avec des tailles de clefs **supérieures à 1024 bits** qui satisfont à ce qui suit **[RSA], [DSA]**.

Raffinement systématique	*The TSF := Les fonctions de chiffrement/déchiffrement asymétrique de l'environnement*

FCS_COP.1.1 *The TSF shall perform [assignment: list of cryptographic operations] in accordance with a specified cryptographic algorithm [assignment: cryptographic algorithm] and cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards].*

Affectation+ mise en forme	*list of cryptographic operations* := aucun ⌫
Affectation	*cryptographic algorithm* := RSA et DSA

Affectation	*cryptographic key sizes :*= supérieures à 1024 bits
Affectation	*list of standards :*= [RSA], [DSA]

Note d'application :

[RSA] : La norme correspondante à RSA et aux divers modes de padding est définie par PKCS#1.

[DSA] : La norme est définie par FIPS PUB 186 : National Institute of Standards and Technology (NIST), FIPS Publication 186: Digital Signature Standard (DSS)*, 1994.*

Argumentaire : Cette exigence est dédiée au soutien des objectifs OE.ChiffrementAsymétriqueEnvironnement et OE.DéchiffrementAsymétriqueEnvironnement.

Les dépendances avec FDP_ITC.1, FCS_CKM.1, FCS_CKM.4 et FMT_MSA.2 ne sont pas applicables.

La dépendance avec FDP_ITC.1 est non applicable car l'accès aux fonctions de chiffrement déchiffrement asymétrique et donc l'accès au certificat associé est réalisée par la méthode d'accès aux certificats FCS_CKM.3.RoutagePilotage.

La dépendance avec FCS_CKM.1 car la génération des clefs permettant les fonctions cryptographiques n'est pas à la charge de l'environnement.

La dépendance avec FCS_CKM.4 est non applicable car la suppression d'une fonction de chiffrement déchiffrement asymétrique d'un magasin et donc du certificat associé ne nécessite pas de méthode particulière de destruction de son contenu.

La dépendance avec FMT_MSA.2 est non applicable car les attributs de la clef associée aux fonctions de chiffrement déchiffrement asymétrique (type de la clef, utilisateur, période de validité et key usage) ne peuvent pas être modifiés.

6. SPECIFICATION DE LA TOE

6.1. FONCTIONS DE SECURITE DE LA TOE

6.1.1. CHIFFREMENT

La fonction chiffrement se déroule en plusieurs étapes :

- Choix des paramètres de chiffrement symétrique ; génération de la clef de session. Pour les algorithmes, voir [Sym].
- Choix du certificat public du ou des destinataires (voir obtention de certificat), possibilité d'itération de chiffrement
- Possibilité de vérification de la chaîne de confiance d'un certificat. (voir vérification de la chaîne de confiance d'un certificat)
- Chiffrement symétrique du document à l'aide de la clef de session.
- Chiffrement asymétrique de la clef de session à l'aide du certificat utilisateur. Pour l'algorithme voir [RSA].
- Mise au format demandé par le service. Si le message à chiffrer était signé, c'est-à-dire au format PKCS#7 *signed data*, alors le format retourné par la fonction est PKCS#7 *signed and enveloped data.* Dans un autre cas le format retourné est un PKCS#7 *enveloped data*. Il peut être encodé de diverses façons suivant la demande du service.

[Sym] :

Référence Algorithme	**Taille de Clefs**	**Standard**
des128-ede3-CBC	128	FIPS 46-3
des192-ede3-CBC	192	FIPS 46-3
aes128-ECB	128	FIPS PUB 197
aes128-CBC	128	FIPS PUB 197
aes128-OFB	128	FIPS PUB 197
aes128-CFB	128	FIPS PUB 197
aes192-ECB	192	FIPS PUB 197
aes192-CBC	192	FIPS PUB 197
aes192-OFB	192	FIPS PUB 197

aes192-CFB	192	FIPS PUB 197
aes256-ECB	256	FIPS PUB 197
aes256-CBC	256	FIPS PUB 197
aes256-OFB	256	FIPS PUB 197
aes256-CFB	256	FIPS PUB 197
rc6128-ECB	128	RRSY98
rc6128-CBC	128	RRSY98
rc6128-OFB	128	RRSY98
rc6128-CFB	128	RRSY98
idea128-ECB	128	LMM92
idea128-CBC	128	LMM92
idea128-OFB	128	LMM92
idea128-CFB	128	LMM92

Note d'application :

Pour les modes CBC, une valeur initiale doit être imposée. Pour les modes ECB, les applications les utilisant doivent se limiter à des utilisations ne mettant pas à jour leurs vulnérabilités.

FIPS PUB 197 et FIPS 46-3 sont des publications du National Institute of Standards and Technology.

LMM92 : X. Lai, J.L. Massey and S. Murphy, Markov ciphers and differential cryptanalysis, Advances in Cryptology - Eurocrypt '91, Springer-Verlag (1992), 17-38.

RRSY98 : Ronald L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin, The RC6 Block Cipher (1998).

[RSA] : La norme correspondante à RSA ("RSAES-PKCS1-v1_5" et "RSAEP-OAEP" pour le mécanisme de chiffrement asymétrique) et aux divers modes de padding est définie par PKCS#1, v2.1 : RSA Cryptography Standard- RSA Laboratories – 14/06/2002

Argumentaire : FCS_COP.1.Sym.1 permet la réalisation du chiffrement symétrique. FCS_COP.1.Asym.1 permet la réalisation du chiffrement asymétrique de la clef de session. FCS_CKM.3.RoutagePilotage.1 permet l'enchaînement correcte des opérations et l'accès aux ressources choisies. Elle permet en particulier l'accès aux différents magasins de certificats.

6.1.2. DECHIFFREMENT

La fonction déchiffrement permet de déchiffrer un message aux formats PKCS#7 *enveloped data* ou *signed and enveloped data*. Elle se déroule en plusieurs étapes :

- Choix du certificat associé à la ressource externe à partir des informations relatives aux destinataires du message. (voir obtention des certificats)

- Possibilité de vérification de la chaîne de confiance d'un certificat. (voir vérification de la chaîne de confiance d'un certificat)
- Déchiffrement asymétrique de la clef de session par la ressource externe.
- Déchiffrement symétrique du document à l'aide de la clef de session. Pour les algorithmes, voir [Sym].
- Mise au format demandé par le service. Si le message à déchiffrer était signé (PKCS#7 *signed and enveloped data)*, un format PKCS#7 *signed data* est retourné. Il peut alors être encodé de diverses façons.

[Sym] :

Référence Algorithme	**Taille de Clefs**	**Standard**
des128-ede3-CBC	128	FIPS 46-3
des192-ede3-CBC	192	FIPS 46-3
aes128-ECB	128	FIPS PUB 197
aes128-CBC	128	FIPS PUB 197
aes128-OFB	128	FIPS PUB 197
aes128-CFB	128	FIPS PUB 197
aes192-ECB	192	FIPS PUB 197
aes192-CBC	192	FIPS PUB 197
aes192-OFB	192	FIPS PUB 197
aes192-CFB	192	FIPS PUB 197
aes256-ECB	256	FIPS PUB 197
aes256-CBC	256	FIPS PUB 197
aes256-OFB	256	FIPS PUB 197
aes256-CFB	256	FIPS PUB 197
rc6128-ECB	128	RRSY98
rc6128-CBC	128	RRSY98
rc6128-OFB	128	RRSY98
rc6128-CFB	128	RRSY98
idea128-ECB	128	LMM92
idea128-CBC	128	LMM92
idea128-OFB	128	LMM92
idea128-CFB	128	LMM92

Note d'application :

Pour les modes CBC, une valeur initiale doit être imposée. Pour les modes ECB, les applications les utilisant doivent se limiter à des utilisations ne mettant pas à jour leurs vulnérabilités.

FIPS PUB 197 et FIPS 46-3 sont des publications du National Institute of Standards and Technology.

LMM92 : X. Lai, J.L. Massey and S. Murphy, Markov ciphers and differential cryptanalysis, Advances in Cryptology - Eurocrypt '91, Springer-Verlag (1992), 17-38.

RRSY98 : Ronald L. Rivest, M.J.B. Robshaw, R. Sidney, and Y.L. Yin, The RC6 Block Cipher (1998).

Argumentaire : FCS_COP.1.Sym.1 permet la réalisation du déchiffrement symétrique. FCS_CKM.3.RoutagePilotage.1 permet l'enchaînement correcte des opérations et l'accès aux ressources choisies. Elle permet en particulier le déchiffrement asymétrique de la clef de session.

6.1.3. SIGNATURE

La création de signature se déroule en plusieurs étapes :

- Initialisation :
  - choix des différents paramètres : sélection des certificats associés aux clefs privées (voir obtention d'un certificat), de l'algorithme de hachage (voir SHA)
  - mise au format PKCS#7 de l'objet à signer (format *signed data*)
- Itération de signatures : pour chaque itération on génère une signature électronique
  - La génération du condensat du document à signer
  - Le chiffrement asymétrique par la ressource externe associée au certificat choisi (voir obtention de certificat)
  - Possibilité de vérification de la chaîne de confiance d'un certificat. (voir vérification de la chaîne de confiance d'un certificat)

  Ces opérations sont renouvelées si plusieurs signataires sont sélectionnés, toutefois, s'ils utilisent le même algorithme de hachage, le condensat n'est généré qu'une fois.
- La finalisation de la signature : les données sont renvoyées au service au format de son choix. Le format renvoyé par la fonction est PKCS#7 (format *signed data*), il peut être encodé de diverses façons suivant la demande du service.

Note d'application :

Dans le cadre de l'évaluation, seul l'algorithme de hachage suivant est envisagé :

[SHA] : Federal Information Processing Standards Publication (FIPS-PUB) 180-2-Secure Hash Standard 01/08/2002 modifié le 25/02/2004.

Argumentaire : FCS_COP.1.Condensat.1 permet la génération du condensat du document à signer. FCS_CKM.3.RoutagePilotage.1 permet l'enchaînement correcte des opérations et l'accès aux ressources choisies. Elle permet en particulier le chiffrement asymétrique du condensat par la ressource externe.

6.1.4. VERIFICATION DE SIGNATURE

Cette fonction permet de vérifier la signature (éventuellement les signatures) d'un document au format PKCS#7.

Cette vérification comporte plusieurs étapes :

- Vérification cryptographique des signatures : déchiffrement asymétrique (pour les algorithmes voir [RSA], [DSA]) de la signature, génération du condensat du document (voir [SHA]) puis comparaison avec la signature déchiffrée.

- Possibilité de vérification de la chaîne de confiance d'un certificat. (voir vérification de la chaîne de confiance d'un certificat)

Note d'application :

Dans le cadre de l'évaluation, seul l'algorithme de hachage suivant est envisagé :

[SHA] : Federal Information Processing Standards Publication (FIPS-PUB) 180-2-Secure Hash Standard 01/08/2002 modifié le 25/02/2004.

Les algorithmes de chiffrement/déchiffrement asymétrique sont les suivants :

[RSA] : La norme correspondante à RSA ("RSASSA-PKCS1-v1_5" et "RSASSA-PSS" pour les mécanismes de signature) et aux divers modes de padding est définie par PKCS#1, v2.1 : RSA Cryptography Standard- RSA Laboratories – 14/06/2002

[DSA] : La norme est définie par FIPS PUB 186-2 : Federal Information Processing Standards Publication – Digital Signature Standard – 27/01/2000 modifié le 05/10/2001..

Argumentaire : FCS_COP.1.Condensat.1 permet la génération du condensat du document dont la signature doit être vérifiée. FCS_COP.1.Asym.1 permet le déchiffrement asymétrique du condensat chiffré par l'expéditeur. FCS_CKM.3.RoutagePilotage.1 permet l'enchaînement correcte des opérations et l'accès aux ressources choisies. Elle permet en particulier l'accès aux différents magasins de certificats.

6.2. FONCTIONS DE SECURITE DE SOUTIEN DE LA TOE

6.2.1. OBTENTION D'UN CERTIFICAT

6.2.1.1. CERTIFICATS LIES A UNE RESSOURCE EXTERNE

Le service doit pouvoir accéder aux certificats ressources et ainsi piloter les ressources externes.

La fonction permet la mise en place d'un filtre permettant la sélection de la ressource suivant le type du certificat avant utilisation pour un chiffrement ou un déchiffrement asymétrique éventuel.

Les filtres possibles sont des filtres de sélection selon le *key usage* du certificat, son numéro de série ou son possesseur ou des filtres de validité permettant de sélectionner uniquement les certificats en cours de validité (voir vérification de la chaîne de confiance).

Argumentaire : FCS_CKM.3.RoutagePilotage.1 permet l'enchaînement correcte des opérations et l'accès aux ressources choisies. Elle permet en particulier l'accès aux différents magasins de certificats et aux ressources externes.

6.2.1.2. AUTRES CERTIFICATS

Cette fonction doit permettre de positionner des filtres en vue de la sélection d'un certificat dans les divers magasins. Ce filtre est paramétrable suivant les besoins du service et permet d'obtenir le certificat public afin de l'utiliser après l'avoir sélectionné suivant les paramètres demandés par le service.

Les filtres qui peuvent être utilisés ici sont des filtres suivant le type de certificat : AC, serveur (qui peuvent correspondre à des magasins distincts) mais aussi suivant le *key usage*, le possesseur ou encore la validité.

Argumentaire : FCS_CKM.3.RoutagePilotage.1 permet l'enchaînement correcte des opérations et l'accès aux ressources choisies. Elle permet en particulier l'accès aux différents magasins de certificats et aux ressources externes.

6.2.2. VERIFICATION DE LA CHAINE DE CONFIANCE D'UN CERTIFICAT

Cette fonction permet de vérifier la confiance en un certificat donné, à une date donnée.

Pour cela, elle doit établir une « chaîne de confiance » partant du certificat à vérifier et aboutissant à un certificat déclaré de confiance sur le poste.

La première étape correspond à la vérification des champs du certificat : *key usage* correct, certificat correspondant à un utilisateur ou un serveur donné… Ces paramètres dépendent du filtre choisi par le service.

La vérification doit parcourir la chaîne et effectuer pour chacun des certificats un certain nombre de vérifications unitaires :

- le certificat est en cours de validité.
- le certificat a été signé par la clé se trouvant dans le certificat suivant, sauf pour le dernier certificat de la chaîne qui doit être auto signé ou déclaré de confiance par l'utilisateur ;
- le certificat n'a pas été révoqué ; cette fonction fait appel à un contrôleur de révocation au choix du service : OCS ou CRL. Par défaut pas de contrôle de révocation.

Argumentaire : FMT_MTD.3.ValiditéCertificat.1 permet la vérification des paramètres d'un certificat et la vérification de la chaîne de confiance. Elle permet en particulier le filtrage des certificats suivant des paramètres fixés par le service afin de les sélectionner dans les magasins. Les opérations de vérification de la chaîne de confiance s'appuient sur la vérification de signature et sur l'obtention des certificats. En ce sens les exigences citées pour la vérification de signature et l'obtention de certificat participe également à cette fonction.

6.2.3. GENERATION DE CLEFS DE SESSION

La génération des clés est réalisée à partir d'un générateur de nombre pseudo aléatoire (PRNG).

Le générateur pseudo aléatoire utilise un générateur de graine qui permet de construire 32 octets imprévisibles.

Pour cela on utilise plusieurs sources :

- des sources dites fixes qui dépendent de l'état de la machine :

  . Adresse IP

  . Mémoire utilisée et disponible

  . Paramètres système

  . Liste des fichiers temporaires avec taille et date de modification.

- et une source dite variable obtenue en faisant tourner 128 fois un compteur pendant 10 millisecondes. La source variable est constituée des 128 valeurs de ce compteur ainsi obtenues à l'issue des 10 millisecondes de comptage. On obtient ainsi 128 entiers.

Les sources sont condensées avec SHA1, ce qui permet d'obtenir 2 fois 20 octets qui constituent la graine originelle.

Le générateur pseudo aléatoire est constitué d'un algorithme de chiffrement par bloc en mode compteur. Précisément les 32 premiers octets de la graine sont utilisés comme clef pour chiffrer (algorithme AES) un compteur de 64 bits; les octets aléatoires produits sont pris séquentiellement dans chaque bloc de 16 octets ainsi produit, quand un bloc est épuisé on incrémente le compteur ce qui donne le bloc suivant. Ceci permet d'obtenir autant d'octets qu'on le souhaite.

Cet aléa est utilisé ensuite pour générer des clefs symétriques.

Dans le cas particulier des algorithmes DES et 3-DES, si la clé générée correspond à une des clés faibles connues (voir tableau ci-après), une nouvelle sera générée jusqu'à ce qu'elle ne soit plus identifiée comme étant faible.

Tableau des valeurs de clés faibles testées :

0x01,0x01,0x01,0x01, 0x01,0x01,0x01,0x01,
0x1f,0x1f,0x1f,0x1f, 0x0e,0x0e,0x0e,0x0e,
0xe0,0xe0,0xe0,0xe0, 0xf1,0xf1,0xf1,0xf1,
0xfe,0xfe,0xfe,0xfe, 0xfe,0xfe,0xfe,0xfe,
0x01,0xfe,0x01,0xfe, 0x01,0xfe,0x01,0xfe,
0x1f,0xe0,0x1f,0xe0, 0x0e,0xf1,0x0e,0xf1,
0x01,0xe0,0x01,0xe0, 0x01,0xf1,0x01,0xf1,
0x1f,0xfe,0x1f,0xfe, 0x0e,0xfe,0x0e,0xfe,
0x01,0x1f,0x01,0x1f, 0x01,0x0e,0x01,0x0e,
0xe0,0xfe,0xe0,0xfe, 0xf1,0xfe,0xf1,0xfe,
0xfe,0x01,0xfe,0x01, 0xfe,0x01,0xfe,0x01,
0xe0,0x1f,0xe0,0x1f, 0xf1,0x0e,0xf1,0x0e,
0xe0,0x01,0xe0,0x01, 0xf1,0x01,0xf1,0x01,
0xfe,0x1f,0xfe,0x1f, 0xfe,0x0e,0xfe,0x0e,
0x1f,0x01,0x1f,0x01, 0x0e,0x01,0x0e,0x01,
0xfe,0xe0,0xfe,0xe0, 0xfe,0xf1,0xfe,0xf1

Argumentaire : FCS_CKM.1.Clefs.1 permet la génération des clefs de session en vue d'un chiffrement symétrique.

6.3. MESURES D'ASSURANCE SECURITE

6.3.1. MESURES DE L'ENVIRONNEMENT DE DEVELOPPEMENT

6.3.1.1. METHODES ET OUTILS DE GESTION DE CONFIGURATION

Le système de gestion de configuration couvre la gestion et le contrôle du développement, de la production et de la maintenance de APPLATOO. Son application permet d'affecter un identifiant unique à chaque version de la TOE et d'établir une liste des versions des composants qui constituent une version donnée.

Le commanditaire documente les procédures du système de gestion de configuration et fournit une liste de configuration pour chaque version de la TOE présentée.

L'évaluateur évalue la documentation et contrôle, sur les versions de la TOE qui lui sont livrées par le commanditaire, que le système de gestion de configuration est bien appliqué tel que décrit dans la documentation (pas d'audit de l'environnement de développement sous ce critère).

Références des fournitures : "Procédures de l'environnement de développement"

Argumentaire : ces procédures satisfont à l'exigence ACM_CAP.2.

6.3.1.2. SECURITE DE L'ENVIRONNEMENT DE DEVELOPPEMENT

Les mesures de sécurité appliquées pour le développement et la maintenance de APPLATOO garantissent l'intégrité du code exécutable de la TOE et la confidentialité des documents de développement associés.

Le commanditaire documente les mesures de sécurité de l'environnement de développement en identifiant précisément le périmètre cet environnement, et fournit des traces de l'application de ces mesures.

L'évaluateur évalue la documentation et procède à un audit de l'environnement afin de vérifier et d'apprécier l'application des mesures, et d'interviewer les personnels concernés sur leur connaissance des mesures.

Références des fournitures : "Procédures de l'environnement de développement"

Argumentaire : ces procédures satisfont à l'exigence ALC_DVS.1.

6.3.1.3. PROCEDURES DE LIVRAISON

Les procédures et mesures mises en place pour transférer APPLATOO du développeur chez le développeur de service garantissent l'authenticité et l'intégrité de la TOE lors du transfert.

Le commanditaire documente les procédures de livraison.

L'évaluateur évalue la documentation et procède à un audit de l'environnement afin de vérifier et d'apprécier l'application des mesures, et d'interviewer les personnels concernés sur leur connaissance des mesures.

Références des fournitures : "Procédures de l'environnement de développement"

Argumentaire : ces procédures satisfont à l'exigence ADO_DEL.1.

6.3.1.4. PROCEDURES DE CORRECTION DES ANOMALIES

Des procédures de correction des anomalies sont mises en place au niveau de l'équipe de développement pour assurer une gestion et un contrôle des anomalies de sécurité découvertes en interne ou soumises par les exploitants (développeur de service ou utilisateur final), ainsi que la distribution des correctifs associés, une fois les anomalies résolues.

Le commanditaire documente les procédures visant à la correction des anomalies, et fournit les documents donnant des lignes directrices aux exploitants pour lui soumettre les anomalies.

L'évaluateur évalue la documentation (pas d'audit de l'environnement de développement sous ce critère).

Références des fournitures : "Procédures de l'environnement de développement"

Argumentaire : ces procédures satisfont à l'exigence ALC_FLR.3.

6.3.2. DOCUMENTATION ET OUTILS DE DEVELOPPEMENT DES FONCTIONS DE SECURITE

Le commanditaire fournit les documents permettant d'assurer un niveau de qualité compatible avec les exigences liées au paquet d'assurance sécurité : spécifications fonctionnelles, conception de haut niveau et, uniquement pour les fonctions cryptographiques, conception de bas niveau, documentation des outils et techniques de développement (compilateurs, makefiles…) et code source. Ces documents forment les niveaux successifs de représentation de la fonctionnalité de sécurité.

Des correspondances entre ces niveaux sont établies, en commençant par les fonctions de sécurité des TI spécifiées de manière abrégée dans ce document.

L'évaluateur évalue la documentation, et vérifie que les exigences fonctionnelles de sécurité se reflètent bien dans les différents niveaux de représentation de la fonctionnalité de sécurité.

Références des fournitures : "Architecture", "Spécifications détaillées", "Javadoc", "Code TOE", "Outils de développement"

Argumentaire : ces mesures satisfont aux exigences ADV_FSP.1, ADV_HLD.2, ADV_LLD.1, ALC_TAT.1, ADV_IMP.1 et ADV_RCR.1.

6.3.3. TEST DES FONCTIONS DE SECURITE

6.3.3.1. PROCEDURES DE TEST DU DEVELOPPEUR

Le commanditaire fournit les documents produits à l'occasion des tests qu'il a effectués sur la TOE. Ces documents doivent décrire le plan et les procédures de tests suivies et montrer le degré de couverture des spécifications fonctionnelles par les tests. Ils doivent inclure les résultats effectifs des tests et démontrer que les fonctions de sécurité se comportent bien de la manière spécifiée dans les spécifications fonctionnelles.

Le commanditaire met également à disposition de l'évaluateur une TOE se prêtant au repassage des tests qu'il a effectués sur la TOE.

L'évaluateur évalue la documentation et repasse une partie des tests du développeur.

Références des fournitures : "Cahier de Test", "Compte-rendu de Recette"

Argumentaire : ces procédures satisfont aux exigences ATE_FUN.1, ATE_COV.1, ATE_DPT.1 et à une partie d'ATE_IND.2 (repassage des tests).

6.3.3.2. TEST INDEPENDANT PAR L'EVALUATEUR

Le commanditaire met à disposition de l'évaluateur une TOE se prêtant à l'exécution de tests indépendants.

Sur la base des spécifications fonctionnelles et de la documentation de test, l'évaluateur conçoit des tests complémentaires des fonctions de sécurité, afin de valider des comportements de sécurité de la TOE que le commanditaire n'aurait pas testés.

Références des fournitures : sans objet.

Argumentaire : ces mesures satisfont à une partie de l'exigence ATE_IND.2 (test indépendant).

6.3.4. DOCUMENTATION D'EXPLOITATION

6.3.4.1. PROCEDURES D'INSTALLATION ET DE DEMARRAGE

Ces procédures permettent l'installation et le démarrage de la TOE dans des conditions qui garantissent une exécution satisfaisante de ses fonctions de sécurité.

Dans le contexte d'utilisation du produit, l'utilisateur final applique les procédures suivantes :

- Dans le cas où la TOE est utilisée côté serveur (mode « servlet »), les procédures préconisées par le développeur de service pour interfacer la servlet avec le serveur HTTP ;

- Dans le cas où la TOE est utilisée côté client (mode « applet »), la procédure implémentée par le navigateur pour l'installation de nouvelles applets.

Dans ces conditions, le commanditaire n'a pas à rédiger de procédure spécifique d'installation ou de démarrage à destination du développeur de services ou de l'utilisateur final. Il se contente de préciser quelles sont les procédures d'installation et de démarrage standard, en référence à des plate-formes du marché.

L'évaluateur évalue la sécurité des procédures standard dans les cas d'utilisation prévus, en les réappliquant toutes ou en partie.

Références des fournitures : "Howto", "FAQ", "Documentation Technique"

Argumentaire : ces procédures satisfont aux exigences ADO_IGS.1 et AVA_MSU.1 (concernant la documentation d'installation et de démarrage).

6.3.4.2. DOCUMENTATION D'ADMINISTRATION

La documentation d'exploitation à destination des développeurs de service oit décrire le comportement des fonctions de sécurité et refléter les hypothèses sur l'environnement d'exploitation, dans une optique de configuration, de maintenance et de maintien en condition opérationnelle corrects des fonctions de sécurité. Elle doit également décrire les différents types d'événements relatifs à la sécurité susceptibles de survenir, et fournir des lignes directrices sur la manière de les prendre en compte.

Des exigences spécifiques à la prévention de l'utilisation impropre, similaires à celles sur la documentation d'installation et de démarrage, pèsent également sur la documentation d'administration

Le commanditaire fournit la documentation d'administration.

L'évaluateur évalue la documentation d'administration.

Références des fournitures : "Howto", "FAQ", "Documentation Technique"

Argumentaire : ces mesures satisfont aux exigences AGD_ADM.1 et AVA_MSU.1 (concernant la documentation d'administration).

6.3.4.3. DOCUMENTATION UTILISATEUR

La documentation d'exploitation à destination des utilisateurs doit décrire le comportement des fonctions de sécurité qu'ils ont besoin de connaître, et refléter les hypothèses sur l'environnement d'exploitation et les responsabilités qui les concernent (et notamment les situations qui nécessitent d'en référer à l'administrateur).

Des exigences spécifiques à la prévention de l'utilisation impropre, similaires à celles sur la documentation d'installation et de démarrage et d'administration, peuvent également peser sur la documentation utilisateur, sous réserve de leur pertinence et de leur applicabilité aux utilisateurs.

Le commanditaire fournit la documentation utilisateur.

L'évaluateur évalue la documentation utilisateur.

Références des fournitures : "Howto", "FAQ", "Documentation Technique"

Argumentaire : ces mesures satisfont aux exigences AGD_USR.1 et AVA_MSU.1 (concernant la documentation utilisateur).

6.3.5. ESTIMATION DE LA VULNERABILITE

Cette tâche s'appuie sur les résultats de toutes les tâches précédentes et sur des sources publiques pour identifier les faits techniques (vulnérabilités) susceptibles de causer la réalisation de menaces identifiées dans la présente cible de sécurité ou des infractions aux règles de politiques de sécurité organisationnelles de la présente cible de sécurité. La résistance des mécanismes de sécurité de nature combinatoire ou probabiliste aux attaques directes est également estimée.

Le commanditaire doit fournir une analyse de vulnérabilités énonçant toutes les vulnérabilités qu'il a décelées au cours du développement, montrant qu'elles ne sont pas exploitables et justifiant que la cible d'évaluation résiste aux attaques de pénétration requérant une compréhension minimale de son fonctionnement.

Une analyse de la résistance des mécanismes pour lesquels une annonce de résistance des fonctions a été faite dans la présente cible de sécurité doit également être fournie par le commanditaire.

L'évaluateur rédige une analyse indépendante de vulnérabilités (à destination exclusive de l'organisme de certification), sur la base des autres fournitures de l'évaluation, afin de confirmer ou d'infirmer les résultats des analyses du commanditaire. Il procède d'autre part à des tests de pénétration dans le but d'estimer les moyens nécessaires à la mise en œuvre des vulnérabilités (compétence technique, temps, expertise, etc.). Cette tâche permet de confirmer ou d'infirmer que le niveau minimum de moyens nécessaires estimés, mesurés selon une métrique décrite dans les Critères Communs, est strictement supérieur à ceux correspondant à un potentiel d'attaque élémentaire.

Références des fournitures : "Analyse de la vulnérabilité"

Argumentaire : ces mesures satisfont aux exigences AVA_SOF.1 et AVA_VLA.2.

6.4. BILAN RECAPITULATIF

	ACM_CAP.2	ADO_DEL.1	ADO_IGS.1	ADV_FSP.1	ADV_HLD.2	ADV_IMP.1	ADV_LLD.1	ADV_RCR.1	AGD_ADM.1	AGD_USR.1	ALC_DVS.1	ALC_FLR.3	ALC_TAT.1	ATE_COV.1	ATE_DPT.1	ATE_FUN.1	ATE_IND.2	AVA_MSU.1	AVA_SOF.1	AVA_VLA.2
Méthodes et outils de gestion de configuration	X																			
Sécurité de l'environnement de développement											X									
Procédures de livraison		X																		
Procédures de correction des anomalies												X								
Documentation et outils de développement des fonctions de sécurité				X	X	X	X	X					X							
Procédures de test du développeur														X	X	X	X			

	ACM_CAP.2	ADO_DEL.1	ADO_IGS.1	ADV_FSP.1	ADV_HLD.2	ADV_IMP.1	ADV_LLD.1	ADV_RCR.1	AGD_ADM.1	AGD_USR.1	ALC_DVS.1	ALC_FLR.3	ALC_TAT.1	ATE_COV.1	ATE_DPT.1	ATE_FUN.1	ATE_IND.2	AVA_MSU.1	AVA_SOF.1	AVA_VLA.2
Test indépendant par l'évaluateur																	X			
Procédures d'installation et de démarrage			X															X		
Documentation d'administration									X									X		
Documentation utilisateur										X								X		
Estimation de la vulnérabilité																			X	X

7. ANNEXES

7.1. ACRONYMES

ASN.1	Norme de l'UIT-T. Définit une syntaxe générale d'expression de valeurs typées. L'ASN.1 est un format d'échange de données structurées (arbres, listes, etc.) entre plates-formes hétérogènes.
CAPI	Cryptographic API API générique de services cryptographique de Microsoft. Les services cryptographiques disponibles dans les programmes Windows (authentification, chiffrement, etc.) sont fournis au moyen de différentes techniques : cartes à puce, périphériques USB, logiciels, etc. Pour ajouter une nouvelle technique, on installe sur l'ordinateur un module répondant à la spécification CAPI. Lorsque Windows aura besoin d'accéder à une technique cryptographique précise, il passera par le l'implémentation de CAPI correspondante.
CSP	Cryptographic Service Provider
JNI	Java Native Interface
PKCS	Public Key Cryptography Standards Normes concernant des services cryptographiques
PKCS#1	Définition des standards de cryptographie RSA
PKCS#7	Format de message cryptographique : requête de certificat (ou CMS :Cryptographic Message Syntax)
PKCS#10	Format de requête de certificat (ou CSR : certificate Signing Request)
PKCS#11	Comme pour CAPI, il existe une implémentation de PKCS#11 pour chaque technique de cryptographie. Ces implémentations peuvent être appelées par un programmeur d'applications sans que celui-ci ait connaissance des détails bas-niveau de la technique cryptographique employée. Netscape fait appel, pour faire de la cryptographie, à des modules répondant à la spécification PKCS#11. Ou CryptoKi.
PKCS#12	Format de magasin de clés et certificats éventuellement protégés par mot de passe.
PKIX	Public Key Infrastructure - X.509 Groupe de travail de l'IETF (Internet Engineering Task Force) visant à faciliter la genèse d'Infrastructure à Clefs Publiques basées sur la norme X.509 pour des applications Internet. PKIX a produit des normes telles que les extensions de X.509 pour l'Internet, OCSP, etc.
ST	Security Target (Cible de Sécurité)
TOE	Target of Evaluation (Cible d'Evaluation)
USB.	Universal Serial Bus : port utilisé pour connecter en chaîne des périphériques (jusqu'a 127) sans conflit. Le périphérique fonctionne directement sans redémarrer (c'est que l'on appelle du "HotPlug", branchement à chaud en français). Son utilisation est très variée: Souris, claviers, joystick, enceinte, graveurs, cameras.
X509	Certification de clés publiques Norme de l'UIT-T (Union Internationale des Télécoms, section Normalisation des Télécoms) spécifiant un cadre de travail pour la certification de clés publiques. Cette norme définit entre autres un format de certificat qui sera notre référence. Nous entendons suivre plus précisément la norme RFC 2459 du groupe PKIX, qui adapte la norme X.509 pour les applications Internet.

7.2. DEFINITIONS

Certificat : Carte d'identité numérique. Il prend la forme d'un fichier contenant une clé publique et des informations sur son propriétaire. Ces informations sont certifiées (i.e. signées) par une Autorité de Confiance (AC).

Chaîne de confiance : Chaîne qui permet d'établir la validité d'un certificat. Lors de son émission un certificat est signé par un autre certificat. On peut ainsi remonter jusqu'à un certificat émis par une Autorité de Certification. Cette chaîne permet d'assurer la confiance dans un certificat en remontant jusqu'à un autre certificat considéré comme sûr (soit émis par une Autorité de Confiance, soit stocké dans un magasin de l'utilisateur et validé par ce dernier comme certificat de confiance).

Une chaîne est **valide** si elle aboutit à un certificat sûr.

Une chaîne de confiance est **sûre** si elle aboutit à un certificat d'Autorité de Certification.

Condensat : Image d'un fichier généré (sur un certain nombre d'octets) à partir d'une fonction (appelée fonction de hashage) qui donne une représentation du fichier et sur laquelle les opérations de signature sont effectuées.
Différents algorithmes peuvent être utilisé pour cette génération.

Key usage : champ du certificat précisant l'utilisation pour laquelle il est destiné : chiffrement, signature…

Chiffrement/Déchiffrement symétrique : la même clef est utilisé pour chiffrer et déchiffrer (ici elle est appelée clef de session).

Chiffrement/Déchiffrement asymétrique : deux clefs différentes sont utilisés pour chiffrer et déchiffrer : une clef publique (pour chiffrer) et une clef privée (pour déchiffrer). (ou inversement pour une signature)

Clef de Session : Clef générée à chaque transfert qui permet de réaliser le chiffrement/déchiffrement symétrique ; cette clef est par la suite elle-même chiffrée/déchiffrée de façon asymétrique.

7.3. REFERENCES

[QPS-std]	Qualification de Produits de Sécurité – niveau standard
[CC-01]	Critères Communs pour l'évaluation de la sécurité des technologies de l'information, Partie 1, Version 2.2
[CC-02]	Critères Communs pour l'évaluation de la sécurité des technologies de l'information, Partie 2, Version 2.2
[CC-03]	Critères Communs pour l'évaluation de la sécurité des technologies de l'information, Partie 3, Version 2.2
[ALC_FLR]	Evaluation Methodology, ALC_FLR - Flaw Remediation
[PRI]	Politique de Référencement Intersectorielle (voir www.adae.gouv.fr)