AGENCE DES SYSTÈMES D'INFORMATION PARTAGÉS DE SANTÉ

CryptoLib CPS3

Spécifications externes du module PKCS#11

« Pôle Technique et Sécurité »

Identification du document	
Référence	SpécificationsExternes_PKCS11_CryptoLib_CPS3.docx
Date de création	04/08/10
Date de dernière mise à jour	10/10/12
Etat	En cours / A vérifier / A valider / **Validé**
Rédaction (R)	PTS/PSCE
Version	**V 1.0.1**
Vérification	PTS/PSCE
Validation finale (A)	PTS/PSCE
Classification[1]	**Public**
Nombre de pages	**12**

Agence des systèmes d'information partagés de santé
9, rue Georges Pitard - 75015 Paris
T. + 33 (0)1 58 45 32 50
www.asipsante.fr

Historique du document			
Version	Date	Auteur	Commentaires
V 1.0.0	04/08/10	PSCC	Création
V 1.0.1	10/10/12	PTS/PSCE	Modifications de forme

Sommaire

1 Préambule

1.1 Documents de références

Appellation	Type du document	Nom du document
[R1]	Spécifications	PKCS #11 v2.20: Cryptographic Token Interface Standard
[R2]	Spécifications	ASIP_CPS3_Données-métier_v1.0.2

1.2 Terminologies et abréviations

Abréviation	Nom du document
CPS	Carte de Professionnel de Santé
GALSS	Gestionnaire d'Accès au Lecteur Santé Social
PSS	Protocole Santé Social
Cryptolib	Ensemble des composants librairie PKCS#11, CSP, CCM, TokenD
PC/SC	Personal Computer/Smart Card
CSP	Cryptographic Service Provider
PKCS#11	Standard définissant une interface générique d'accès aux périphériques cryptographiques. (Public Key Cryptographic Standards)
CAPI	Microsoft Cryptography API
CCM	CAPI Certificate Manager
PKCS#15	Cryptographic Token Information Format Standard
IAS ECC	Identification, Authentification, Signature, Carte Européenne du Citoyen
PIN	Personal Identification Number

2 Introduction

Le présent document constitue les spécifications externes de la librairie PKCS#11 de la Cryptolib supportant les cartes CPS3 au format IAS ECC ainsi que les cartes CPS2ter.

3 Support

3.1 CPS3

Une carte CPS3 comporte deux modes possibles pour l'accès aux données et aux fonctions cryptographiques :

- Le mode CPS2ter qui assure la compatibilité ascendante pour les applications CPS2ter et les APIs CPS.
- Le mode IAS ECC.

La librairie PKCS#11 assure uniquement le support de la carte CPS3 au travers du mode IAS ECC en utilisant un driver carte IAS.

Afin de se conformer au standard IAS, la Cryptolib doit implémenter les contraintes de ce dernier. Par exemple lors de signature avec hashing, le standard IAS impose que la dernière étape de hashing soit réalisée par la carte à puce.

3.2 CPS2ter

La carte CPS2ter n'ayant pas une structure PKCS#15, la librairie PKCS#11 assure son support au travers de la structure spécifique de la CPS2ter.

4 Traitements spécifiques de la Cryptolib

4.1 Fonctions

4.1.1 Vue d'ensemble

La Cryptolib CPS3 implémente un sous ensemble des fonctions présentées dans les spécifications PKCS#11 [R1]. De plus certaines fonctions bénéficient d'une implémentation spécifique. Le tableau ci-dessous, représente les fonctions implémentées conformément aux spécifications PKCS#11 avec la mention « Implémentation » à « standard »; les fonctions implémentées de façon spécifique avec la mention « Implémentation» à « spécifique »; et enfin les fonctions non implémentées avec la mention « Implémentation » à « non ».

Categorie	Fonction	Implémentation
Général	C_Initialize	standard
	C_Finalize	standard
	C_GetInfo	standard
	C_GetFunctionList	standard
Slots & Cartes	C_GetSlotList	standard
	C_GetSlotInfo	standard
	C_GetTokenInfo	standard
	C_WaitForSlotEvent	standard
	C_GetMechanismList	standard
	C_GetMechanismInfo	standard
	C_InitToken	non
	C_InitPIN	**spécifique**
	C_SetPIN	standard
Sessions	C_OpenSession	standard
	C_CloseSession	standard
	C_CloseAllSessions	standard
	C_GetSessionInfo	standard
	C_GetOperationState	non
	C_SetOperationState	non
	C_Login	**spécifique**
	C_Logout	standard
Objets	C_CreateObject	non
	C_CopyObject	non
	C_DestroyObject	non
	C_GetObjectSize	non
	C_GetAttributeValue	oui
	C_SetAttributeValue	**spécifique**
	C_FindObjectsInit	standard
	C_FindObjects	standard
	C_FindObjectsFinal	standard
Chiffrement	C_EncryptInit	non
	C_Encrypt	non
	C_EncryptUpdate	non
	C_EncryptFinal	non

Categorie	Fonction	Implémentation
Déchiffrement	C_DecryptInit	non
	C_Decrypt	non
	C_DecryptUpdate	non
	C_DecryptFinal	non
Empreintes	C_DigestInit	standard
	C_Digest	standard
	C_DigestUpdate	standard
	C_DigestKey	non
	C_DigestFinal	standard
Signature	C_SignInit	standard
	C_Sign	standard
	C_SignUpdate	standard
	C_SignFinal	standard
	C_SignRecoverInit	non
	C_SignRecover	non
Vérification de signature	C_VerifyInit	standard
	C_Verify	standard
	C_VerifyUpdate	standard
	C_VerifyFinal	standard
	C_VerifyRecoverInit	non
	C_VerifyRecover	non
Fonctions dual	C_DigestEncryptUpdate	non
	C_DecryptDigestUpdate	non
	C_SignEncryptUpdate	non
	C_DecryptVerifyUpdate	non
Clés	C_GenerateKey	non
	C_GenerateKeyPair	non
	C_WrapKey	non
	C_UnwrapKey	non
	C_DeriveKey	non
Génération aléatoire	C_SeedRandom	non
	C_GenerateRandom	standard
Fonctions parallèles	C_GetFunctionStatus	non
	C_CancelFunction	non

Tableau 1 : Tableaux des fonctions PKCS#11 (implémentées conformément au standard, non-implémentées, et implémentées spécifiquement)

Dans la suite de ce document, seront décrites plus en détail, uniquement les fonctions présentées en gras dans le tableau ci-dessus, c'est-à-dire, les fonctions dont l'implémentation est spécifique. Pour plus de détails sur les autres fonctions implémentées, le lecteur est invité à consulter le document de spécifications de la norme PKCS#11 [R1].

4.1.2 Implémentations spécifiques

4.1.2.1 C_InitPin

La nouvelle CryptoLib PKCS#11, dédiée aux cartes CPS2Ter et CPS3, implémente un traitement standard du recyclage du code PIN.

Le traitement standard se fait par l'enchainement des fonctions suivantes :

- Login avec le PIN SO : présente le code PUK à la carte
- InitPIN en passant le nouveau code PIN : initialise le code PIN de l'utilisateur
- Logout : met fin à la session loguée SO

Pour la carte CPS3 IAS, le processus est conforme au standard.

Cependant pour la carte CPS2ter, le recyclage du code PIN nécessite de passer dans le même ordre carte le code PUK et le nouveau code PIN. Pour contourner cette contrainte, il faut :

- Lors du login SO, faire appel à l'instruction carte de recyclage du code PIN afin de tester la validité du code PUK. Le code PIN passé lors de cette commande est un code PIN aléatoire.
- Lors de l'initPIN, faire appel à l'instruction carte de modification du code PIN pour remplacer le code PIN aléatoire par celui fourni par l'utilisateur.

4.1.2.2 C_Login

Afin de répondre à la contrainte introduite par le recyclage du code PIN des cartes CPS2Ter, la fonction de Login, pour le cas Security Officer(SO), intègre le traitement spécifique décrit dans le paragraphe précédent.

4.1.2.3 C_SetAttributeValue

La Cryptolib PKCS#11 permet uniquement la modification de l'attribut CKA_VALUE, et cette dernière n'est réalisable que si le champ CKA_MODIFIABLE de l'objet est positionné à CK_TRUE.

Remarque : Fonction implémentée uniquement pour pouvoir modifier l'objet « CPS_DATA » qui est le seul objet modifiable sur une carte CPS.

4.2 Algorithmes

Parmi les algorithmes présentés dans les spécifications PKCS#11 [R1], la Cryptolib CPS3 supporte uniquement les algorithmes suivants :

Algorithmes supportés
CKM_RSA_PKCS
CKM_RSA_X_509
CKM_SHA1_RSA_PKCS
CKM_SHA256_RSA_PKCS
CKM_SHA_1
CKM_SHA256

Tableau 2 : Algorithmes supportés par la Cryptolib CPS3

4.3 Gestion des objets métier CPS

4.3.1 Accès

Pour accéder aux objets métier de la carte CPS, l'application doit disposer des labels associés. Ces labels diffèrent en fonction qu'il s'agisse d'une carte CPS2Ter ou d'une carte CPS3. Ils sont référencés, avec leur dénomination, dans le paragraphe suivant.

L'accès à certains de ces objets est protégé par code porteur. Il nécessite donc le login utilisateur. Ces objets sont représentés avec la mention « Accès » à « Protégé » dans le tableau du paragraphe suivant.

4.3.2 Labels

Label (CKA_LABEL) CPS3	Label (CKA_LABEL) CPS2Ter	Dénomination	Accès
CPS_DATA		Objet de données applicative	Libre
CPS_ID_CARD	CPS2TER_ID_CARD	Identification carte	Libre
CPS_NAME_PS	CPS2TER_NAME_PS	Caractéristiques porteur	Libre
CPS_LANG_PS	CPS2TER _LANG_PS	Codes langues	Libre
	CPS2TER_QUALIF_PS	Qualification	Libre
CPS_ACTIVITY_01_PS	CPS2TER_ACTIVITY_01_PS	Activité / Situation d'exercice 1	Protégé
CPS_ACTIVITY_02_PS	CPS2TER_ACTIVITY_02_PS	Activité / Situation d'exercice 2	Protégé
CPS_ACTIVITY_03_PS	CPS2TER_ACTIVITY_03_PS	Activité / Situation d'exercice 3	Protégé
CPS_ACTIVITY_04_PS	CPS2TER_ACTIVITY_04_PS	Activité / Situation d'exercice 4	Protégé
CPS_ACTIVITY_05_PS	CPS2TER_ACTIVITY_05_PS	Activité / Situation d'exercice 5	Protégé
CPS_ACTIVITY_06_PS	CPS2TER_ACTIVITY_06_PS	Activité / Situation d'exercice 6	Protégé
CPS_ACTIVITY_07_PS	CPS2TER_ACTIVITY_07_PS	Activité / Situation d'exercice 7	Protégé
CPS_ACTIVITY_08_PS	CPS2TER_ACTIVITY_08_PS	Activité / Situation d'exercice 8	Protégé
CPS_ACTIVITY_09_PS	CPS2TER_ACTIVITY_09_PS	Activité / Situation d'exercice 9	Protégé
CPS_ACTIVITY_10_PS	CPS2TER_ACTIVITY_10_PS	Activité / Situation d'exercice 10	Protégé
CPS_ACTIVITY_11_PS	CPS2TER_ACTIVITY_11_PS	Activité / Situation d'exercice 11	Protégé
CPS_ACTIVITY_12_PS	CPS2TER_ACTIVITY_12_PS	Activité / Situation d'exercice 12	Protégé
CPS_ACTIVITY_13_PS	CPS2TER_ACTIVITY_13_PS	Activité / Situation d'exercice 13	Protégé
CPS_ACTIVITY_14_PS	CPS2TER_ACTIVITY_14_PS	Activité / Situation d'exercice 14	Protégé
CPS_ACTIVITY_15_PS	CPS2TER_ACTIVITY_15_PS	Activité / Situation d'exercice 15	Protégé
CPS_ACTIVITY_16_PS	CPS2TER_ACTIVITY_16_PS	Activité / Situation d'exercice 16	Protégé
CPS_INFO_PS	CPS2TER_INFO_PS	Caractéristiques professionnelles PS	Libre
	CPS2TER_SIT_FACT	Situation / Facturation	Libre

Tableau 3 : Tableau représentant les libellés des objets métiers pour les cartes CPS3 et CPS2Ter

4.4 Format

Dans la précédente Cryptolib PKCS#11 de la carte CPS, les objets spécifiques CPS étaient lus sous forme de structure C. Ces objets étaient remontés au travers de l'interface PKCS#11 sous forme d'objets de type VENDOR _DEFINED.

Dans la nouvelle Cryptolib PKCS#11 de la carte CPS, les objets spécifiques CPS stockés sur la CPS3 sont remontés tels que lus sur la carte, c'est-à-dire au format ASN.1.

Afin d'obtenir un comportement homogène pour les applications, la Cryptolib CPS3 remonte également les objets spécifiques CPS lus sur une carte CPS2Ter dans les objets PKCS#11 CKO_DATA au format ASN.1.

C'est le rôle de l'application d'adapter le décodage ASN.1 des données métier en fonction du type de carte détectée.

Le format des différents objets de données est décrit dans le document référencé [R2].

5 Traces

La Cryptolib CPS3 peut gérer 2 types de traces :

- Les traces purement PKCS#11

- Les traces internes à l'implémentation (plus fines).

Remarque : Les données sensibles, telles que le code porteur, ou le code de déblocage, sont masqués.

5.1 Configuration

La Cryptolib détermine les éléments à tracer en fonction de variables d'environnement présentes dans son contexte d'exécution.

Il est à la charge de l'utilisateur de la librairie de positionner ces variables.

Pour les traces purement PKCS#11 il faut positionner la variable à « true ».

Pour les traces internes à l'implémentation, il faut positionner un niveau pouvant aller de 0 à 10. Plus ce niveau est élevé, plus les traces sont détaillées. Il est recommandé d'utiliser le niveau 10 qui trace les données échangées avec la carte.

Sous Microsoft Windows il existe deux manières de positionner les variables nécessaires :

- En positionnant la clé de registre en exécutant le fichier .reg suivant :

```
[HKEY_CURRENT_USER\Software\ASIP Sante\PKCS11]
"Traces"=dword:00000001
"Debug"="10"
```

Remarque : Pour désactiver ces traces, il faut exécuter le fichier .reg suivant :

```
[HKEY_CURRENT_USER\Software\ASIP Sante\PKCS11]
"Traces"=-
"Debug"=-
```

- En positionnant les variables d'environnement comme suit :
    - set CPS_PKCS11_TRACES=true pour les traces purement PKCS#11.
    - set CPS3_DEBUG=0 à 10 pour les traces internes à l'implémentation.

  Remarque : Pour désactiver ces traces, il faut écraser les variables d'environnement comme suit :
    - CPS_PKCS11_TRACES=
    - CPS3_DEBUG=

Sous Linux il existe une manière de positionner les variables nécessaires:

- En positionnant les variables d'environnement comme suit :
    - export CPS3_PKCS11_TRACES=true pour les traces purement PKCS#11.
    - export CPS3_DEBUG=0 à 10 pour les traces internes à l'implémentation.

  Remarque : Pour désactiver ces traces, il faut écraser les variables d'environnement comme suit :
    - unset CPS_PKCS11_TRACES
    - unset CPS3_DEBUG

Sous MAC OS X, il faut créer un fichier dans /Library/Preferences/santesocial/CPS/Coffre/ avec le contenu suivant :

```
traces
{
        active = true;
        debug = 10;
}
```

5.2 Emplacement

Les traces sont sauvegardées :

- Sous Microsoft Windows, dans le répertoire :

  C:\Documents and Settings\All Users\Application Data\santesocial\cps\log
- Sous Linux, dans le répertoire :

  /var/opt/santesocial/CPS/log/
- Sous MAC OS X, dans le répertoire :

  /Library/Logs/santesocial/CPS