



Ministère du budget, des comptes publics et de la fonction publique

Direction Générale de la Modernisation de l'Etat

Middleware IAS

PKCS#11 - Crypto API Guide de programmation

AVEC VOUS l'administration
SE MODERNISE

Middleware IAS	
CAPI-PKCS#11	
Référence	Date
MDWIAS_PKCS11CAPI_v1 03.doc	27/09/07
Identification d'objet (OID)	Racine OID et gestionnaire
1.2.250.1.137.2.3.2.1.1	SDAE
Responsable	Version
DGME/SDAE	V1.03
Critère de diffusion	Nombre de pages
Public	19

HISTORIQUE DES VERSIONS			
DATE	VERSION	EVOLUTION DU DOCUMENT	REDACTEUR
11/10/06	0.01	Création	Dictao
13/10/06	0.02	Mise à jour	Dictao
13/10/06	1.0	Validation	Dictao
24/08/07	1.01	Mise à jour	Dictao
27/09/07	1.03	Support du PIN_SO et validation	Dictao

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	2/19

Content

1 - Introduction	4
1.1 - Références.....	4
2 - Contexte	5
3 - Impact d'IAS sur les fonctions P11 et CAPI supportées	6
3.1 - Incompatibilité avec un canal sécurisé	6
3.2 - Clés de signature nécessitant un condensé des données sur la carte	6
4 - PKCS #11	7
4.1 - Fonctions implémentées	7
4.2 - Informations du token.....	8
4.3 - Authentification de l'utilisateur	9
4.4 - Objets	9
4.4.1 - Objets clé secrète de session.....	9
4.4.2 - RSA private key objects.....	9
4.4.3 - Objet certificat X.509	11
4.5 - Mécanismes	11
4.5.1 - CKM_RSA_PKCS_KEY_PAIR_GEN	11
4.5.2 - CKM_RSA_PKCS.....	12
4.5.3 - CKM_SHA1_RSA_PKCS	13
4.5.4 - CKM_SHA256_RSA_PKCS	13
4.6 - Description des fonctions	14
5 - CryptoAPI	16
5.1 - Key containers	16
5.2 - Authentification de l'utilisateur	16
5.3 - Droit concernant les clés.....	16
5.4 - Clés de session et condensé	17
5.5 - Description des fonctions	17

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	3/19

1 - Introduction

Ce document spécifie l'ensemble des fonctions PKCS#11 et CryptoAPI qui seront supportées dans les modules correspondants et faisant partie du middleware IAS. Ce middleware IAS permet de piloter des cartes basées sur le socle commun IAS et référencées.

1.1 - Références

[PKCS #11]	PKCS #11 v2.01. http://www.rsasecurity.com/rsalabs
[CryptoAPI]	Cryptographic API in Windows. http://msdn2.microsoft.com/en-us/library/aa380252.aspx
[IAS]	“FR Common IAS Platform Specification 1.01 Premium.”,
[ISO 7816-4: 2005]	ISO/IEC 7816-4 Second Edition 2005-01-15
[ISO 7816-8: 2004]	ISO/IEC 7816-8 Second Edition 2004-06-01
[ISO 7816-15: 2004]	ISO/IEC 7816-15 First Edition 2004-01-15
[Profils Adèle]	« Profils de Personnalisation des cartes IAS pour le support de l'administration électronique »
[IAS API]	« Guide de programmation IAS API »

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	4/19

2 - Contexte

Le choix de la version 2.01 du standard PKCS#11 et la liste des fonctions PKCS#11 et de la CryptoAPI est lié à la définition des profils de carte pour l'administration électronique (cf. [Profils Adèle]). Ces profils ont été définis dans le cadre du projet IAS en accord avec les demandes spécifiques de la DGME. (cf. spécification requise par le client ou cas d'utilisation)

Parmi ces exigences, les suivantes ont un impact sur cette spécification :

- Trois niveaux spécifiques de sécurité, compatibles avec les niveaux de sécurité *, **, *** de la PRIS, ont été définis pour s'adresser aux objets de la carte (cf. [Profils Adèle]). Ces niveaux de sécurité nécessitent l'utilisation d'un canal sécurisé entre la carte et l'application côté serveur de confiance pour quelques fonctionnalités spécifiques. Comme il n'y a pas de moyen pour gérer un tel canal sécurisé avec l'interface PKCS#11 ou la CryptoAPI, l'ensemble des fonctionnalités supportées qui peuvent être utilisées au travers de cette interface dépend du niveau de sécurité requis. Pour les autres fonctionnalités voir la spécification de l'API IAS.
- L'utilisation et l'administration du PIN de signature doivent être effectuées via un canal sécurisé. C'est pour cette raison que le PIN de signature et tous les objets qui peuvent être utilisés pour gérer ce code PIN ne sont pas utilisables par l'interface Crypto API, ils sont gérés par l'API IAS.
- Le PIN global de la carte est le seul qui peut être utilisé sans canal sécurisé. De ce fait, on admettra qu'il sera le seul code PIN utilisé à la fois par PKCS#11 et par la CryptoAPI. Au niveau PKCS#11, il est représenté par CKU_USER PIN.
- Quel que soit le nombre d'applications cryptographiques dans la carte, il y a au moins l'application ADELE.
- Les mises à jour des applications ADELE et ADELE Administrateur 2 doivent être exécutées via un canal sécurisé. Par conséquent, l'application ADELE est considérée en lecture seule pour les interfaces PKCS#11 et CryptoAPI.
- Les applications de type Adèle Générique peuvent être mises à jour par les interfaces PKCS#11 et CryptoAPI.
- Les applications de type Emetteur contenant des objets non cryptographiques peuvent être lues par les interfaces PKCS#11 uniquement.

Les APIs PKCS11 et CSP permettent de tirer pleinement partie des cartes émises selon les profils basés sur les exemples de profils définis dans [ProfilsAdèle]. Certaines fonctions d'écriture (création ou mises à jours de certificats) peuvent ne pas être possibles si le profil émis ne le permet pas (typiquement si les conditions de mise à jour des objets nécessitent des clés de « secure messaging »)

Les fonctions de la carte qui nécessitent un canal sécurisé ne sont pas supportées par les interfaces PKCS#11 et CryptoAPI. Elles sont supportées par l'API IAS spécifique (cf. [Spécification IAS API])

L'implémentation du *Card Module* pour le provider *Microsoft Smart Card Base Cryptographic Provider* qui supporte la carte IAS n'est pas dans le cadre de cette spécification. Cela fera partie d'une autre spécification.

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	5/19

3 - Impact d'IAS sur les fonctions P11 et CAPI supportées

3.1 - Incompatibilité avec un canal sécurisé

Le standard IAS fournit le moyen d'établir un canal sécurisé authentifié. Un couple commande et réponse sécurisée est toujours accompagné par un MAC (Code d'authentification du message : *Message Authentication Code*) qui authentifie et préserve l'intégrité de la commande et de la réponse. Comme option, le canal sécurisé peut aussi chiffrer la commande et la réponse pour assurer la confidentialité.

Il y a deux façons d'établir un canal sécurisé :

1. Authentification symétrique où la paire de clés 3DES est partagée entre la carte et le monde extérieur.
2. Authentification asymétrique où la carte détient une clé privée RSA pour l'authentification interne et la clé publique de l'AC pour la vérification du ou des certificats vérifiables de la carte qui correspond à la clé privée RSA utilisée par le monde extérieur pour l'authentification externe. Cette authentification mutuelle se fait à l'intérieur d'une session chiffrée qui est fondée sur l'échange de clés de Diffie-Hellman.

Tout l'intérêt d'un canal sécurisé est d'établir un moyen sûr de communiquer entre une application de confiance et un environnement de confiance dans la carte. Comme le protocole de sécurité est fourni au niveau des commandes APDU et requiert d'avoir des clés d'authentification, ni la CryptoAPI ni PKCS#11 ne peuvent être utilisées comme des API entre la carte et les applications. Ainsi les fonctions de la carte exigeant un canal sécurisé ne sont pas exposées au travers des interfaces PKCS#11 et CryptoAPI. Se référer à [Matrice de cas d'usages IAS] pour plus d'information sur la visibilité des objets de la carte par les API en fonction des conditions d'accès.

3.2 - Clés de signature nécessitant un condensé des données sur la carte

Le standard IAS supporte deux commandes de la carte qui peuvent générer une signature numérique avec une clé privée RSA :

INTERNAL AUTHENTICATE : cette commande accepte n'importe quelles données en entrée sans contrainte de taille, elle ajoute un padding PKCS#1 type 1 avant d'exécuter l'exponentiation RSA.

PSO CDS¹ : cette commande pourra utiliser le résultat issu de la commande précédente PSO HASH, en lui ajoutant la structure de l'empreinte encode en BER, puis le padding PKCS#1 type 1 et pour finir effectué l'exponentiation RSA. La valeur de PSO HASH est créée en faisant le calcul intermédiaire du hash sur l'entité externe à la carte puis en exécutant le hachage du dernier block sur la carte. Ce qui signifie qu'effectivement les données en clair doivent être disponibles pour la fonction de signature, alors que la commande INTERNAL AUTHENTICATE n'a besoin que de la valeur du hash.

Une clé privée donnée aura l'ensemble des conditions d'accès pour activer / désactiver les commandes ci-dessus. Ce qui aboutit à certains comportements fonctionnels au niveau de la CryptoAPI et de PKCS#11 qui sont détaillés plus loin dans cette spécification.

¹ PSO CDS = PERFORM SECURITY OPERATION – CREATE DIGITAL SIGNATURE

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	6/19

4 - PKCS #11

Cette partie de la spécification définit quelles parties de PKCS#11 sont implémentées par une librairie dans le but de supporter une carte IAS conforme à la spécification de [Profil Adèle].

[PKCS #11], (aussi appelé Cryptoki) est une interface de programmation d'application (*API : Application Programming Interface*) pour, en général, les tokens cryptographiques. PKCS#11 fait parti de l'ensemble des standards PKCS (*Public Key Cryptographic Standards*) publiés par RSA Security et que l'on peut trouver sur <http://www.rsasecurity.com/rsalabs>.

Une librairie qui est conforme à cette spécification doit être conforme à la version 2.01 de PKCS#11.

4.1 - Fonctions implémentées

La spécification PKCS#11 définit un ensemble de fonctions appelées pour accéder au token et utiliser ses ressources. Comme cette spécification est construite pour supporter une grande variété de tokens, il n'est pas nécessaire d'implémenter toutes les fonctions, seulement celles qui sont applicables au token supporté.

Les fonctions requises par cette spécification sont limitées à celles qui sont applicables aux les applications de l'utilisateur final, comme un exploreur ou un client mail.

Le contenu ADELE des cartes est censé être personnalisé par l'émetteur des cartes, il n'est donc pas nécessaire de pouvoir modifier celui-ci via l'interface PKCS#11 après émission.

Néanmoins, si des modifications, après émission de la carte, d'une partie du contenu de ADELE sont en effet requises, ce sera géré par un autre module que PKCS#11 et qui est ici hors du cadre de cette spécification (cf. [Spécification IAS API]).

D'un point de vue PKCS#11, la partie ADELE de la carte n'apparaît qu'en lecture seule (ou en écriture protégée).

Seules les parties génériques de la carte peuvent être mises à jour par l'interface PKCS#11. Ainsi, d'un point de vue PKCS#11, ces parties seront en lecture et écriture.

De plus, les seules opérations cryptographiques exposées par l'interface PKCS#11 sont celles impliquant les clés privées RSA. On s'attend à ce que les opérations de clé publique et de clé symétrique, quand elles sont exigées, soient exécutées à l'extérieur de la carte.

Les fonctions PKCS#11 suivantes sont implémentées :

C_Initialize, C_Finalize, C_GetInfo, C_GetFunctionList, C_GetSlotList, C_GetSlotInfo, C_GetTokenInfo, C_GetMechanismList, C_GetMechanismInfo, C_SetPIN, C_OpenSession, C_CloseSession, C_CloseAllSessions, C_GetSessionInfo, C_Login, C_Logout, C_GetAttributeValue, C_FindObjectsInit, C_FindObjects, C_FindObjectsFinal, C_DecryptInit, C_Decrypt, C_SignInit, C_Sign, C_SignUpdate, C_SignFinal, C_UnwrapKey, C_WaitForSlotEvent, C_InitToken, C_InitPIN,

Les fonctions PKCS#11 suivantes doivent être implémentées avec la restriction suivante : elles renvoient le code CKR_TOKEN_WRITE_PROTECTED pour les objets de l'application ADELE :

C_CreateObject, C_DestroyObject, C_SetAttributeValue, C_GenerateKeyPair.

L'implémentation des fonctions restantes n'est pas exigée. Néanmoins la librairie supportera le fait que si elles sont appelées, elles devront renvoyer le code CKR_FUNCTION_NOT_SUPPORTED. Ces fonctions sont listées ci-dessous :

C_GetOperationState, C_SetOperationState, C_CopyObject, C_GetObjectSize, C_EncryptInit, C_Encrypt, C_EncryptUpdate, C_EncryptFinal, C_DecryptUpdate, C_DecryptFinal, C_DigestInit, C_Digest, C_DigestUpdate, C_DigestKey, C_DigestFinal, C_SignRecoverInit, C_SignRecover, C_VerifyInit, C_Verify, C_VerifyUpdate, C_VerifyFinal, C_VerifyRecoverInit, C_VerifyRecover, C_DigestEncryptUpdate, C_DecryptDigestUpdate, C_SignEncryptUpdate, C_DecryptVerifyUpdate, C_GenerateKey, C_WrapKey, C_DeriveKey, C_SeedRandom, C_GenerateRandom, C_GetFunctionStatus, C_CancelFunction.

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	7/19

4.2 - Informations du token

La plupart des informations essentielles à propos de la carte peuvent être retrouvées dans la structure CK_TOKEN_INFO par l'appel à la fonction C_GetTokenInfo. Cette structure est construite comme décrit dans le tableau suivant :

CK_TOKEN_INFO membre de la structure	Description du contenu
<i>label</i>	Champ fixé à 32 octets. Défini par l'émetteur.
<i>manufacturerID</i>	Champ fixé à 32 octets. Description du fabricant de la carte, défini par l'émetteur.
<i>model</i>	Champ fixé à 16 octets. Description du numéro d'identification (<i>Identification Number</i>) de l'émetteur, partie du numéro de série d'IAS. Il est encodé comme suit : " IAS #-###-##### ", où le premier groupe est l'identification de l'industriel (en accord avec la spécification IAS, il peut avoir la valeur '9'), le deuxième est le code ISO du pays et le troisième est l'identifiant de l'émetteur.
<i>serialNumber</i>	Champ fixé à 16 octets. Contient l'encodage en ASCII, pouvant aller jusqu'à 16 caractères décimaux, de la partie identifiant unique de la carte du numéro de série de IAS. S'il contient moins de 16 caractères, un <i>padding</i> sera fait en ajoutant des blancs en fin de champ (rien ne doit être ajoutant au début, ni zéro, ni blancs).
<i>flags</i>	
CKF_RNG	Optionnel (si TRUE les fonctions C_SeedRandom et C_GenerateRandom peuvent être implémentées)
CKF_WRITE_PROTECTED	TRUE s'il n'y a pas d'application générique PKCS#11 sur la carte, Sinon FALSE.
CKF_LOGIN_REQUIRED	TRUE
CKF_USER_PIN_INITIALIZED	TRUE
CKF_RESTORE_KEY_NOT_NEEDED	FALSE
CKF_CLOCK_ON_TOKEN	FALSE
CKF_PROTECTED_AUTHENTICATION_PATH	FALSE
CKF_DUAL_CRYPTO_OPERATIONS	FALSE
<i>ulMaxSessionCount, ulSessionCount, ulMaxRwSessionCount, ulRwSessionCount</i>	Obligatoire, comme défini dans la carte/librairie
<i>ulMaxPinLen, ulMinPinLen</i>	Obligatoire, comme défini par la carte et l'application CIA.
<i>ulTotalPublicMemory, ulFreePublicMemory, ulTotalPrivateMemory, ulFreePrivateMemory</i>	Optionnel. Si non utilisé, ces valeurs doivent être mises à CK_UNAVAILABLE_INFORMATION
<i>hardwareVersion, firmwareVersion</i>	Optionnel
<i>utcTime</i>	Vide

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	8/19

4.3 - Authentification de l'utilisateur

Les cartes IAS supportées par cette spécification possèdent un PIN global qui correspond à CKU_USER PIN dans la spécification PKCS#11.

4.4 - Objets

Une librairie PKCS#11 conforme avec cette spécification présentera à l'application appelante les clés privées RSA qui sont associées autant au détenteur de la carte qu'au certificat X509 correspondant. Les certificats des AC et les autres certificats exigés pour l'établissement d'une chaîne de confiance doivent être également présents.

4.4.1 - Objets clé secrète de session

La spécification IAS ne supporte pas les clés volatiles ou les opérations cryptographiques utilisant des clés symétriques (secrètes). Néanmoins, les librairies PKCS#11 implémentées en conformité avec cette spécification devront implémenter C_UnwrapKey pour l'échange des clés de session, où la clé déchiffré devient un objet de session dans la mémoire de la librairie. La valeur de la clé peut être retrouvée avec C_GetAttributeValue. D'un point de vue de la sécurité, cette étape revient à exécuter directement C_Decrypt, les deux se termineront en utilisant la commande PSO DECIPHER de la carte. C'est pourquoi, une librairie qui supporte C_UnwrapKey de cette manière devrait mettre CKA_DECRYPT à TRUE à chaque fois que CKA_UNWRAP est TRUE pour éviter de donner une fausse impression de sécurité.

4.4.2 - RSA private key objects

La table suivante donne les valeurs des attributs qui doivent être supportées pour les objets clé privée RSA. The table below shows attribute values that must be supported for RSA private key objects.

Attribut	Valeur
CKA_CLASS	CKO_PRIVATE_KEY
CKA_TOKEN	TRUE
CKA_PRIVATE	TRUE
CKA_MODIFIABLE	FALSE quand il appartient à l'application ADELE, TRUE quand il appartient à une application générique PKCS#11.
CKA_LABEL	Toujours défini.
CKA_KEY_TYPE	CKK_RSA
CKA_ID	Toujours défini. Il est unique dans la carte pour toute paire de clés. Il coïncide avec l'attribut CKA_ID du certificat correspondant.
CKA_START_DATE	Optionnel. Egal à la date de validité <i>notBefore</i> du certificat correspondant.
CKA_END_DATE	Optionnel. Egal à la date de validité <i>notAfter</i> du certificat correspondant.
CKA_DERIVE	FALSE
CKA_LOCAL	TRUE ou FALSE, dépend de l'endroit où la paire de clés est générée.
CKA_SUBJECT	Optionnel. Représente <i>subject</i> encodé en DER du certificat correspondant.
CKA_SENSITIVE	TRUE

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	9/19

Attribut	Valeur
CKA_DECRYPT	TRUE si le flag <i>dataEncipherment</i> du certificat correspondant est donné, sinon FALSE.
CKA_SIGN	TRUE si le flag <i>digitalSignature</i> ou <i>nonRepudiation</i> du certificat correspondant est donné, sinon FALSE.
CKA_SIGN_RECOVER	FALSE
CKA_UNWRAP	TRUE si le flag <i>keyEncipherment</i> du certificat correspondant est donné, sinon FALSE.
CKA_EXTRACTABLE	FALSE
CKA_ALWAYS_SENSITIVE	Dépend de l'historique de clé
CKA_NEVER_EXTRACTABLE	Dépend de l'historique de clé
CKA_MODULUS	Toujours présent
CKA_PUBLIC_EXPONENT	Toujours présent
CKA_PRIVATE_EXPONENT, CKA_PRIME_1, CKA_PRIME_2, CKA_EXPONENT_1, CKA_EXPONENT_2, CKA_COEFFICIENT	Jamais révélés

Comme expliquée dans la section 3.2 -, la présence ou l'absence de commande APDU de contrôle d'accès à la carte a un impact sur les utilisations des clés et les mécanismes permis, comme indiqué dans le tableau ci-dessous. Les permissions de clés doivent être cohérentes avec les flags d'utilisation des clés du certificat.

P S O C D S	I N T E R N A L A U T H	P S O D E C I P H E R	Mécanismes permis	CKA_S	CKA_D E C R Y P T C K A _ U N W R A P
				IGN	P
*	*	-	Pas de mécanisme de signature	FALSE	-
✓	*	-	CKM_SHA1_RSA_PKCS Optionnellement CKM_SHA256_RSA_PKCS	TRUE	-
*	✓	-	CKM_RSA_PKCS (Calcul du condensé et construction des blocs hors de la carte)	TRUE	-
✓	✓	-	CKM_RSA_PKCS en utilisant INTERNAL AUTHENTICATE CKM_SHA1_RSA_PKCS en utilisant PSO CDS CKM_SHA256_RSA_PKCS en utilisant PSO CDS	TRUE	-
-	-	*	Pas de mécanisme de déchiffrement	-	FALSE
-	-	✓	CKM_RSA_PKCS	-	TRUE

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	10/19

4.4.3 - Objet certificat X.509

Attribut	Valeur
CKA_CLASS	CKO_CERTIFICATE
CKA_TOKEN	TRUE
CKA_PRIVATE	FALSE
CKA_MODIFIABLE	FALSE
CKA_LABEL	Toujours fourni. Il est destiné à aider l'utilisateur pour choisir la clé.
CKA_CERTIFICATE_TYPE	CKC_X_509
CKA_START_DATE	Optionnel. Egal à la date de validité <i>notBefore</i> correspondant au certificat.
CKA_END_DATE	Optionnel. Egal à la date de validité <i>notAfter</i> correspondant au certificat.
CKA_SUBJECT	Toujours fourni. Encodage en DER du <i>subject name</i> du certificat
CKA_ID	Il est toujours fourni si la clé privée associée est enregistrée dans la carte. Dans ce cas, il correspond à l'attribut CKA_ID de la clé privée correspondante. Si la clé privée n'est pas présente, cet attribut est optionnel. Il est toujours unique dans la carte.
CKA_ISSUER	Toujours fourni. Encodage en DER du <i>issuer name</i> du certificat.
CKA_SERIAL_NUMBER	Toujours fourni. Encodage en DER du <i>serial number</i> du certificat
CKA_VALUE	Toujours fourni. Encodage BER du certificat

4.5 - Mécanismes

Une librairie PKCS#11 conforme à cette spécification expose au moins les mécanismes suivants :

- CKM_RSA_PKCS_KEY_PAIR_GEN
- CKM_RSA_PKCS
- CKM_SHA1_RSA_PKCS
- CKM_SHA256_RSA_PKCS

Il supporte les longueurs de clé RSA 1024, 1536 et 2048 bits.

Les sections suivantes spécifient ces mécanismes en termes de structure CK_MECHANISM_INFO retournée par la fonction C_GetMechanismInfo. Une librairie peut optionnellement supporter davantage d'opérations pour ces mécanismes, des longueurs de clé complémentaires ou d'autres mécanismes.

4.5.1 - CKM_RSA_PKCS_KEY_PAIR_GEN

Membre de la structure CK_MECHANISM_INFO	Description du contenu

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	11/19

Membre de la structure CK_MECHANISM_INFO	Description du contenu
<i>ulMinKeySize</i>	1024
<i>ulMaxKeySize</i>	2048
<i>Flags</i>	
CKF_HW	TRUE
CKF_ENCRYPT	FALSE
CKF_DECRYPT	FALSE
CKF_DIGEST	FALSE
CKF_SIGN	FALSE
CKF_SIGN_RECOVER	FALSE
CKF_VERIFY	FALSE
CKF_VERIFY_RECOVER	FALSE
CKF_GENERATE	FALSE
CKF_GENERATE_KEY_PAIR	TRUE
CKF_WRAP	FALSE
CKF_UNWRAP	FALSE
CKF_DERIVE	FALSE

4.5.2 - CKM_RSA_PKCS

Membre de la structure CK_MECHANISM_INFO	Description du contenu
<i>ulMinKeySize</i>	1024
<i>ulMaxKeySize</i>	2048
<i>Flags</i>	
CKF_HW	TRUE
CKF_ENCRYPT	FALSE
CKF_DECRYPT	TRUE
CKF_DIGEST	FALSE
CKF_SIGN	TRUE
CKF_SIGN_RECOVER	FALSE
CKF_VERIFY	FALSE
CKF_VERIFY_RECOVER	FALSE
CKF_GENERATE	FALSE
CKF_GENERATE_KEY_PAIR	FALSE
CKF_WRAP	FALSE
CKF_UNWRAP	TRUE
CKF_DERIVE	FALSE

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	12/19

4.5.3 - CKM_SHA1_RSA_PKCS

Membre de la structure CK_MECHANISM_INFO	Description du contenu
<i>ulMinKeySize</i>	1024
<i>ulMaxKeySize</i>	2048
<i>flags</i>	
CKF_HW	TRUE
CKF_ENCRYPT	FALSE
CKF_DECRYPT	FALSE
CKF_DIGEST	FALSE
CKF_SIGN	TRUE
CKF_SIGN_RECOVER	FALSE
CKF_VERIFY	FALSE
CKF_VERIFY_RECOVER	FALSE
CKF_GENERATE	FALSE
CKF_GENERATE_KEY_PAIR	FALSE
CKF_WRAP	FALSE
CKF_UNWRAP	FALSE
CKF_DERIVE	FALSE
CKF_EXTENSION	FALSE

4.5.4 - CKM_SHA256_RSA_PKCS

Note : S'il est présent, le mécanisme CKM_SHA256_RSA_PKCS doit avoir la même valeur que définie dans la spécification PKCS#11 v2.20 (i.e. 0x00000040).

Membre de la structure CK_MECHANISM_INFO	Description du contenu
<i>ulMinKeySize</i>	1024
<i>ulMaxKeySize</i>	2048
<i>flags</i>	
CKF_HW	TRUE
CKF_ENCRYPT	FALSE
CKF_DECRYPT	FALSE
CKF_DIGEST	FALSE
CKF_SIGN	TRUE
CKF_SIGN_RECOVER	FALSE
CKF_VERIFY	FALSE
CKF_VERIFY_RECOVER	FALSE
CKF_GENERATE	FALSE
CKF_GENERATE_KEY_PAIR	FALSE

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	13/19

Membre de la structure CK_MECHANISM_INFO	Description du contenu
CKF_WRAP	FALSE
CKF_UNWRAP	FALSE
CKF_DERIVE	FALSE
CKF_EXTENSION	FALSE

4.6 - Description des fonctions

Dans le tableau suivant, toutes les fonctions exigées sont listées (dans l'ordre alphabétique). Seuls les détails spécifiques à IAS sont décrits ici. Pour une description générale, se référer à la spécification PKCS #11 v2.01.

Fonction	Remarque
C_CloseSession	
C_CloseAllSessions	
C_CreateObject	Seuls les objets de session ou les objets d'une application PKCS#11 générique peuvent être créés.
C_Decrypt	
C_DecryptInit	Le mécanisme de déchiffrement doit faire référence à CKM_RSA_PKCS. La clé de déchiffrement doit être un objet clé privée RSA du token.
C_DestroyObject	Seuls les objets de session ou les objets d'une application PKCS#11 générique peuvent être détruits.
C_Finalize	
C_FindObjects	
C_FindObjectsFinal	
C_FindObjectsInit	
C_GenerateKeyPair	Seules les paires de clés d'une application PKCS#11 générique peuvent être générées. Voir mécanisme CKM_RSA_PKCS_KEY_PAIR_GEN.
C_GetAttributeValue	
C_GetFunctionList	
C_GetMechanismInfo	Le type du mécanisme doit être un des suivants : <ul style="list-style-type: none"> • CKM_RSA_PKCS • CKM_SHA1_RSA_PKCS • CKM_SHA256_RSA_PKCS Les membres de la structure CK_MECHANISM_INFO sont comme décrits dans la section 4.5 - .

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	14/19

Fonction	Remarque
C_GetMechanismList	Les types de mécanismes suivant doivent être supportés : <ul style="list-style-type: none"> CKM_RSA_PKCS CKM_SHA1_RSA_PKCS CKM_SHA256_RSA_PKCS
C_GetSessionInfo	
C_GetSlotInfo	
C_GetTokenInfo	Les membres de la structure CK_TOKEN_INFO sont comme décrits dans la section 4.2 -
C_Initialize	
C_InitToken	Le PUK sera passé en paramètre en lieu et place du Pin « SO »
C_InitPIN	Le PIN utilisateur peut être fixé dans le cas où une session est ouverte avec un user de type CKU_SO
C_Login	Le type d'utilisateur peut être CKU_USER pour ouvrir sa session avec le PIN utilisateur. (PIN global dans [ADELE profil]) Le type d'utilisateur peut être CKU_SO pour ouvrir sa session avec le PUK. (code de déblocage dans [ADELE profil])
C_Logout	
C_OpenSession	L'implémentation des notifications de <i>call backs</i> est optionnelle.
C_SetAttributeValue	Seuls les objets de session ou les objets d'une application PKCS#11 générique peuvent être mis à jour.
C_SetPIN	Disponible pour le CKU_USER PIN et CKU_SO (i.e. le PIN global et le PUK).
C_Sign	
C_SignFinal	
C_SignInit	Le mécanisme se réfère à l'un des suivants : <ul style="list-style-type: none"> CKM_RSA_PKCS CKM_SHA1_RSA_PKCS CKM_SHA256_RSA_PKCS
C_SignUpdate	
C_UnwrapKey	Le mécanisme se réfère à l'un des suivants : <ul style="list-style-type: none"> CKM_RSA_PKCS
C_WaitForSlotEvent	

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	15/19

5 - CryptoAPI

[CryptoAPI] est une interface de programmation d'application (API *Application Programming Interface*) de l'infrastructure cryptographique dans les systèmes d'exploitation Windows. C'est une architecture où les opérations cryptographiques sont effectuées par des *plug-ins* appelés *Cryptographic Service Provider* (CSP). Les cartes à puce peuvent être intégrées dans la CryptoAPI de deux façons :

- Le fournisseur de la carte peut écrire un plug-in qui communique avec la carte et qui est conforme avec la spécification du CSP.
- Microsoft a implémenté un CSP de base général pour les cartes à puce qui définit de nouveau une interface plus simple pour un Card Module. L'émetteur a alors la possibilité d'écrire un plug-in qui communique avec la carte et qui est conforme à la spécification du Card Module.

Cette partie du document décrit quelles parties de la CryptoAPI sont implémentées dans le but de supporter une carte IAS.

Une implémentation du *Card Module* pour le CSP *Microsoft Smart Card Base Cryptographic Provider* qui supporte la carte IAS n'est pas l'objet de ce document.

5.1 - Key containers

Les informations des objets cryptographiques sur la carte IAS sont rangés dans des fichiers définis par [ISO 7816-15: 2004]. Ce standard succède à PKCS#15, qui a de nouveau été développé comme équivalent de [PKCS #11]. Ainsi, même si les informations exportées au travers de PKCS#11 peuvent être lues directement à partir des fichiers ISO 7816-15, La CryptoAPI a le concept de *key containers* pour lesquels il n'y a pas de contrepartie dans ISO 7816-15.

De toute manière, les informations du *key container* sont dérivées des informations de CKA_ID PKCS #11.

Afin de gérer au mieux les *key containers*, il est nécessaire que le CKA_ID de tout ensemble de certificat relatif aux clés publique et privée partage le même CKA_ID non vide.

La notion de *Default Certificate* est spécifique à la Crypto API. Il y a un seul *Default Certificate* par carte. Il est choisi selon les priorités suivantes :

1. Le certificat utilisable pour un *Smart Card Logon* avec le dernier horodatage
2. Le premier certificat de *Smart Card Logon* valide
3. Le certificat de l'utilisateur avec le dernier horodatage

Le *key container* référençant le *Default Certificate* est dit le conteneur par *Default Container*.

5.2 - Authentification de l'utilisateur

Les cartes IAS supportées par cette spécification ont un PIN global et aussi potentiellement un PIN de signature. Seul le PIN global est approprié aux interfaces du CSP et du *Card Module*.

5.3 - Droit concernant les clés

Comme expliqué dans la section 3.2 -, les contrôles d'accès aux commandes de la carte associées à une clé privée proviennent de la disponibilité des fonctions de la CryptoAPI. Le tableau suivant décrit ces relations :

P	I	P	Fonctions autorisées
S	N	S	
O	T	O	
E	F		

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	16/19

			CSP
✗	✗	-	Pas de fonction de signature
✓	✗	-	CryptSignHash, CRYPT_NOHASHOID n'est pas disponible
✗	✓	-	CryptSignHash
✓	✓	-	CryptSignHash utilisant INTERNAL AUTHENTICATE si CRYPT_NOHASHOID est spécifié et en utilisant PSO CDS sinon
-	-	✗	Pas de fonction de déchiffrement
-	-	✓	CryptImportKey et CryptDecrypt

Les flags de permission d'une clé privée comme ils sont renvoyés par GetKeyParam avec le paramètre KP_PERMISSIONS sont comme suit :

Flag de permission	Description
CRYPT_DECRYPT	TRUE si CryptDecrypt ci-dessus est autorisé, sinon FALSE
CRYPT_EXPORT	FALSE
CRYPT_READ	TRUE
CRYPT_WRITE	FALSE
CRYPT_EXPORT_KEY	FALSE
CRYPT_IMPORT_KEY	TRUE si CryptImportKey est autorisé, sinon FALSE
CRYPT_ARCHIVE	FALSE

CryptoAPI n'a pas de flag de permission de clé pour contrôler l'usage de CryptSignHash.

5.4 - Clés de session et condensé

Un CSP doit implémenter un support pour les clés de session et le condensé afin d'être utile aux applications. Ce sera normalement mis en place comme un passage via les CSP qui font partie du système d'exploitation Windows.

Pour plus de compréhension, ces fonctions font néanmoins partie de cette spécification.

Il est à noter que chaque fois que la commande PSO CDS est utilisée pour créer la signature, l'objet condensé doit avoir été généré par CryptHashData uniquement, tandis que l'utilisation de CryptHashSessionKey retournera un condensé qui ne pourra pas être signé par PSO CDS.

5.5 - Description des fonctions

Dans ce qui suit, toutes les fonctions exigées par cette spécification sont décrites. Les détails spécifiques à IAS sont donnés. Pour une description générale, se référer à la documentation de la CryptoAPI sur *Microsoft Developer Network (MSDN)*.

Fonction	Remarques
CryptAcquireContext	Le type du <i>provider</i> doit être PROV_RSA_FULL. La valeur du flag peut être à zéro ou une combinaison des flags suivants : <ul style="list-style-type: none"> CRYPT_VERIFYCONTEXT CRYPT_SILENT CRYPT_NEWKEYSET

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	17/19

Fonction	Remarques				
	<ul style="list-style-type: none"> CRYPT_DELETEKEYSET 				
CryptGetProvParam	<p>Ce paramètre peut prendre une des valeurs suivantes :</p> <ul style="list-style-type: none"> PP_CONTAINER PP_UNIQUE_CONTAINER PP_ENUMALGS PP_ENUMALGS_EX PP_ENUMCONTAINERS PP_IMPTYPE PP_NAME PP_VERSION PP_PROVTYPE PP_KEYX_KEYSIZE_INC PP_SIG_KEYSIZE_INC PP_ENUMEX_SIGNING_PROT PP_KEYSPEC <p>La valeur du flag peut être une de ces valeurs :</p> <ul style="list-style-type: none"> CRYPT_FIRST CRYPT_MACHINE_KEYSET 				
CryptReleaseContext					
CryptSetProvParam	<p>Ce paramètre peut prendre une de ces valeurs :</p> <ul style="list-style-type: none"> PP_KEYEXCHANGE_PIN PP_SIGNATURE_PIN <p>Tous deux font référence au PIN Global. La valeur du PIN est donnée dans le paramètre pbData. Si on passe un pointeur NULL, le CSP demandera à l'utilisateur de la fixer au prochain accès aux données protégées.</p>				
CryptDeriveKey	S'applique seulement aux clés de session.				
CryptDestroyKey	S'applique seulement aux clés de session et aux clés RSA dans une application générique PKCS#11.				
CryptDuplicateKey	S'applique seulement aux clés de session.				
CryptExportKey	<p>Le type du blob à exporter contenant la clé peut être une de ces valeurs :</p> <ul style="list-style-type: none"> PUBLICKEYBLOB SIMPLEBLOB SYMMETRICWRAPKEYBLOB 				
CryptGenKey	S'applique seulement aux clés de session et aux clés RSA d'une application générique PKCS#11.				
CryptGenRandom					
CryptGetKeyParam	<p>Pour les clés de session, se référer à MSDN. Pour les clés RSA, la valeur du paramètre peut être une de celles-ci-dessous :</p> <table border="1"> <tbody> <tr> <td>KP_ALGID</td> <td>CALG_RSA_KEYX or CALG_RSA_SIGN.</td> </tr> <tr> <td>KP_BLOCKLEN</td> <td>2048</td> </tr> </tbody> </table>	KP_ALGID	CALG_RSA_KEYX or CALG_RSA_SIGN.	KP_BLOCKLEN	2048
KP_ALGID	CALG_RSA_KEYX or CALG_RSA_SIGN.				
KP_BLOCKLEN	2048				

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	18/19

Fonction	Remarques	
	KP_KEYLEN	Tailles de clé supportées : 1024, 1536 and 2048.
	KP_PERMISSIONS	Les flags retournés sont décrits dans la section 5.3 -.
	KP_CERTIFICATE	Renvoie les certificats associés au <i>key container</i>
CryptGetUserKey	AT_KEYEXCHANGE ou AT_SIGNATURE	
CryptImportKey	S'applique seulement aux clés de session et aux clés RSA d'une application générique PKCS#11. Les clés RSA de la carte servant à déchiffrer les clés de session doivent avoir l'autorisation en accord avec la section 5.3 -.	
CryptSetKeyParam	S'applique seulement aux clés de session.	
CryptCreateHash		
CryptDestroyHash		
CryptDuplicateHash		
CryptGetHashParam		
CryptHashData		
CryptHashSessionKey	Voir la note en section 5.4 -	
CryptSetHashParam		
CryptSignHash	Pour l'autorisation d'utiliser cette fonction, voir la section 5.3 -.	
CryptVerifySignature	Peut être exécuté hors carte.	
CryptDecrypt	Pour les clés de session, se référer à MSDN. Pour les clés RSA de la carte, voir la section 5.3 - quant à l'autorisation d'utiliser cette fonction.	
CryptEncrypt	S'applique seulement aux clés de session.	

Middleware IAS		CAPI-PKCS#11		
Identification du document (OID)	Version	Date	Critère de diffusion	Page
1.2.250.1.137.2.3.2.1.1	V1.03	27/09/07	Public	19/19