



## CARTES CPS

### GUIDE DE REFERENCE DE LA CARTE CPS4

Identification du document	
Référence ANS	ANS_CPSV4_0011_Guide_De_Référence_v1_03.docx
Date de création	18/12/2024
Date de dernière mise à jour	04/02/2025
Etat	En cours / A vérifier / A valider / <u>Validé</u>
Rédaction (R)	ANS
Version	<b>V 1.03</b>
Vérification	
Validation finale (A)	
Classification	<u>Diffusion restreinte</u>

<b>Historique du document</b>			
<b>Version</b>	<b>Date</b>	<b>Auteur</b>	<b>Commentaires</b>
V1.0	18/12/2024	ANS/GIE	Création du document
V1.01	20/01/2025	ANS/GIE	Ajout fichiers EF_ACTUA / TMAJ_STATUS
V1.02	22/01/2025	ANS/GIE	Prise en compte remarques GIE/SV (orthographe)
V1.03	04/02/2025	ANS/GIE	Erreur sur graphes : 8 situations max au lieu de 16. Prise en compte des modifications apportées à partir du profil R4V3 (Fichiers PKCS#15).

# Sommaire

<b>1 INTRODUCTION .....</b>	<b>6</b>
1.1    OBJET DU DOCUMENT .....	6
1.2    DOCUMENTS DE REFERENCE .....	6
1.3    TERMINOLOGIE .....	7
1.4    NOTATION .....	8
1.5    CONVENTIONS .....	8
<b>2 DESCRIPTION GENERALE .....</b>	<b>9</b>
2.1    APPLICATIONS DE LA CARTE .....	9
2.2    ARBORESCENCE .....	10
2.3    CONDITIONS D'ACCES .....	10
2.3.1    Opérations vs Objets .....	11
2.3.2    Conditions .....	11
<b>3 ELEMENTS SOUS LE MF (MASTER FILE) .....</b>	<b>12</b>
3.1    FICHIERS .....	13
3.1.1    EF ATR [2F01] .....	13
3.1.2    EF SN [D003] .....	14
3.1.3    EF DIR [2F00] .....	15
3.1.4    EF TECH [2FFF] .....	16
3.2    OBJETS DE SECURITE .....	19
3.2.1    PIN .....	19
3.2.2    PUK .....	19
<b>4 ELEMENTS SOUS LE ADF CPS (0001) .....</b>	<b>20</b>
4.1    FICHIERS PKCS#15 .....	20
4.1.1    EF OD [5031] .....	20
4.1.2    EF CIAINFO [5032] .....	22
4.1.3    EF AOD [7001] .....	23
4.1.4    EF PrKD [7002] .....	25
4.1.5    EF PuKD [7004] .....	26
4.1.6    EF CD [7005] .....	27
4.1.7    EF DCOD [7006] .....	28
4.2    FICHIERS DE DONNEES METIERS .....	39
4.2.1    EF ID_CARTE [D101] .....	39
4.2.2    EF NOM [D102] .....	40
4.2.3    EF LANG [D103] .....	41
4.2.4    EF INFO_PS [D104] .....	42
4.2.5    EF APP_DATA [D107] .....	43
4.2.6    EF PS_SIT_XX [D120-D127] .....	44
4.2.7    DF SIG (0101) .....	45
4.2.8    EF cert-sig .....	45
4.2.9    DF AUTH (0102) .....	46
4.2.10    EF cert-auth .....	46
4.3    OBJETS DE SECURITE .....	47
4.3.1    Clé privée de signature Kpriv-sig .....	47
4.3.2    Clé publique de signature Kpub-sig .....	47
4.3.3    Clé privée d'authentification Kpriv-auth .....	48
4.3.4    Clé publique d'authentification Kpub-auth .....	48
<b>5 ELEMENTS SOUS LE DF CPS2TER (0003) .....</b>	<b>49</b>
5.1    DF CPS2TER (0003) .....	49
5.2    FICHIERS DE DONNÉES MÉTIERS .....	50

5.2.1	<i>EF DIR [2F00]</i> .....	50
5.2.2	<i>EF ATR [2F01]</i> .....	50
5.2.3	<i>EF IC [0005]</i> .....	51
5.2.4	<i>EF ICC [0002]</i> .....	52
5.2.5	<i>EF ID-CARTE [0003]</i> .....	54
5.2.6	<i>EF NAME [0004]</i> .....	55
5.2.7	<i>EF LANG [0006]</i> .....	56
5.2.8	<i>EF PS-IDNAT [4000]</i> .....	57
5.2.9	<i>EF PS-QUALIF [4001]</i> .....	58
5.2.10	<i>EF PS-SITxx [401X]</i> .....	59
5.2.11	<i>DF AMO [7F01]</i> .....	60
5.2.12	<i>EF DIRAMO [2F00]</i> .....	60
5.2.13	<i>EF sit-fact [4000]</i> .....	61
<b>6</b>	<b>APDU DE REFERENCE .....</b>	<b>62</b>
6.1	<i>INTRODUCTION</i> .....	62
6.2	<i>PROTOCOLE</i> .....	62
6.3	<i>GET RESPONSE</i> .....	63
6.4	<i>COMMANDES DE GESTION DU SYSTEME DE FICHIERS</i> .....	64
6.4.1	<i>SELECT</i> .....	64
6.4.2	<i>READ BINARY</i> .....	66
6.5	<i>COMMANDES D'AUTHENTIFICATION DE L'UTILISATEUR</i> .....	67
6.5.1	<i>VERIFY</i> .....	67
6.5.2	<i>RESET RETRY COUNTER</i> .....	68
6.6	<i>COMMANDES DE GESTION DE L'ENVIRONNEMENT DE SECURITE</i> .....	69
6.6.1	<i>MSE SET</i> .....	69
6.7	<i>COMMANDES CRYPTOGRAPHIQUES</i> .....	71
6.7.1	<i>PSO – COMPUTE DIGITAL SIGNATURE</i> .....	71
6.7.2	<i>PSO – DECRYPT</i> .....	72
<b>7</b>	<b>EXEMPLES D'USAGE.....</b>	<b>73</b>
7.1	<i>GESTION DES FICHIERS</i> .....	73
7.1.1	<i>Sélection/lecture de Fichiers sous le MF</i> .....	73
7.1.2	<i>Sélection/lecture de Fichiers sous ADF CPS '0001'</i> .....	75
7.1.3	<i>Sélection/lecture de Fichiers sous DF CPS2TER '0003'</i> .....	77
7.1.4	<i>Sélection/lecture de Certificat/Clé publique sous DF AUTH '0001/0102'</i> .....	81
7.2	<i>GESTION DU CODE PORTEUR</i> .....	84
7.2.1	<i>Blocage puis déblocage du PIN</i> .....	84
7.2.2	<i>Changement de la valeur du PIN avec PIN</i> .....	86
7.2.3	<i>Changement de la valeur du PIN avec PUK</i> .....	88
7.3	<i>OPERATIONS CRYPTOGRAPHIQUES</i> .....	90
7.3.1	<i>Signature d'un condensé/data</i> .....	90
7.3.2	<i>Authentification SSL / Déchiffrement</i> .....	92
<b>8</b>	<b>ANNEXES.....</b>	<b>95</b>
8.1	<i>ATR</i> .....	95
8.2	<i>STRUCTURE DES DONNEES</i> .....	97
8.3	<i>RECAPITULATIF DES IDENTIFIANTS</i> .....	98
8.4	<i>DIFFERENCES V3/V4</i> .....	99
<b>Tableau 1 : Liste des documents applicables .....</b>		<b>6</b>
<b>Tableau 2 : Liste des abréviations .....</b>		<b>7</b>
<b>Tableau 3 : CPLC .....</b>		<b>17</b>
<b>Tableau 4 : GET RESPONSE format de la commande .....</b>		<b>63</b>

Tableau 5 : GET RESPONSE format de la réponse .....	63
Tableau 4 : SELECT format de la commande .....	64
Tableau 7 : SELECT format de la réponse .....	64
Tableau 8 : – SELECT Valeurs possibles de P1, mode de sélection .....	65
Tableau 9 : – SELECT format du FCI .....	65
Tableau 10 : READ BINARY format de la commande .....	66
Tableau 11 : READ BINARY format de la réponse .....	66
Tableau 12 : VERIFY format de la commande .....	67
Tableau 13 : VERIFY format de la réponse .....	67
Tableau 14 : RESET RETRY COUNTER format de la commande .....	68
Tableau 15 : RESET RETRY COUNTER format de la réponse .....	68
Tableau 16 : MSE SET format de la commande .....	69
Tableau 17 : MSE SET format de la réponse .....	69
Tableau 18 : Champ DATA pour CRT DECHIFFREMENT – P1P2='41B8' .....	70
Tableau 19 : Champ DATA pour CRT SIGNATURE DST – P1P2='81B6' .....	70
Tableau 20 : Liste d'algorithmes .....	70
Tableau 21 : Référence des clés .....	70
Tableau 22 : PSO CDS, format de la commande .....	71
Tableau 23 : PSO CDS, format de la réponse .....	71
Tableau 24 : PSO DECIPHER, format de la commande .....	72
Tableau 25 : PSO DECIPHER, format de la réponse .....	72
Tableau 26 : ATR cartes CPS .....	95
Tableau 27 : Identifiants CPS V4 .....	98
Tableau 28 : Différences V3/V4 .....	99

## Figures

Figure 1 : Les applications de la carte CP4 .....	9
Figure 2 : Structure de fichiers/objets contenus dans l'application régalienne .....	10
Figure 3 : Structure de fichiers/objets contenus sous le MF .....	12
Figure 4 : Structure de fichiers/objets contenus sous ADF CPS (0001) .....	20
Figure 5 : Structure de fichiers/objets contenus sous DF CPS2TER (0003) .....	49

# 1 Introduction

## 1.1 Objet du document

Ce document est un guide de référence pour l'utilisation des cartes CPS4.

## 1.2 Documents de référence

N°	Date	Réf. Document	Document
[1]	2004	[ISO7816-3]	ISO/IEC 7816 - Part 3: Electronic signals and transmission protocols
[2]	2003	[ISO7816-4]	ISO/IEC 7816 - Part 4: Inter-industry commands for interchange
[3]	2004	[ISO7816-6]	ISO/IEC 7816 - Part 6: Inter-industry data elements.
[4]	2004	[ISO7816-15]	ISO/IEC 7816 - Part 15 : Cryptographic Information Application
[5]	2017	[ASIP]	ASIP-PUSC-PSCE_Guide_Référence_Carte_CPS3_v1.2.0.doc

Tableau 1 : Liste des documents applicables

## 1.3 Terminologie

Abréviation	Signification
<b>ADF</b>	Application Dedicated File
<b>AID</b>	Application Identifier
<b>AC</b>	Access Condition
<b>APDU</b>	Application Protocol Data Unit
<b>AT</b>	Authentication Template - Descripteur d'authentification
<b>BER</b>	Basic Encoding Rules
<b>CCT</b>	Cryptographic Checksum Template
<b>CLA</b>	Classe (premier) octet d'une commande carte
<b>CPLC</b>	Card Production Life Cycle
<b>CPS</b>	Carte de Professionnel de Santé
<b>CRT</b>	Control Reference Template
<b>CT</b>	Confidentiality Template
<b>EF</b>	Elementary File (fichier élémentaire)
<b>DOCP</b>	Data Object Control Parameters
<b>ES</b>	Etablissement de Santé
<b>FCI</b>	File Control Information
<b>FCP</b>	File Control Parameters
<b>HT</b>	Hash Template – Modèle hash
<b>CHIPDOC</b>	Application NXP conforme aux standards Européens SSCD 419212
<b>ID</b>	Identifiant
<b>INS</b>	Instruction (deuxième) octet d'une commande carte
<b>MAC</b>	Message Authentication Code
<b>MF</b>	Master File
<b>MSE</b>	Manage Security Environment
<b>P1</b>	Premier paramètre d'une commande carte
<b>P2</b>	Deuxième paramètre d'une commande carte
<b>PIN</b>	Personal Identification Number
<b>PUK</b>	Pin Unblocking Key (or PIN)
<b>PS</b>	Professionnel de Santé
<b>PSO</b>	Perform Security Operation - Exécuter l'opération cryptographique
<b>RFU</b>	Réservé pour Utilisation Future
<b>RSA</b>	Rivest, Shamir, Adleman
<b>SDO</b>	Security Data Object - Objet de sécurité
<b>SE</b>	Security Environment - Environnement de sécurité
<b>SEID</b>	Security Environment IDentifier byte - Octet Identificateur d'environnement de sécurité
<b>SM</b>	Secure Messaging - Échanges sécurisés
<b>SW1</b>	Premier octet du statut de retour carte
<b>SW2</b>	Second octet du statut de retour carte
<b>TLV</b>	Tag / Length / Value (Type / Longueur / Valeur)
<b>TMAJ</b>	Télé mise à jour
<b>UQB</b>	Usage Qualifier Byte

Tableau 2 : Liste des abréviations

## 1.4 Notation

0 à 9	Caractères décimaux
'0' à '9' et 'A' à 'F'	Caractères hexadécimaux
'xxxx'	Chaîne de caractères hexadécimaux
"ABC"	Chaîne de caractères alphanumériques
<Réf.>	Référence vers une table de codification ou l'origine de données
α-NUM	Format alphanumérique codé conforme à l'ISO 8859-1
ALPHA	Caractères alphabétiques contenus dans le jeu ASCII
ASCII	Caractères alphanumériques contenus dans le jeu ASCII
BCD	Format Binary Coded Decimal (décimal codé binaire)
	Chiffre de 0 à 9 codé sur un quartet (= demi-octet = 4 bits)
HEX	Format hexadécimal (valeurs de 0 à 255)

## 1.5 Conventions

Les conventions suivantes s'appliquent aux différents tableaux de description des objets de données :

- Le paramètre « Longueur » correspond toujours à un nombre d'octets. Cette longueur est celle du champ « Valeur » du TLV concerné.
- Lorsque la colonne « Valeur » contient la mention <TAB XXX>, cela signifie que la valeur du champ associé est fixée par le Centre de Gestion de l'ANS (SI-CPS). Cette valeur respecte les codes autorisés par la table XXX indiquée.
- La colonne « Présence » peut prendre les valeurs suivantes :
  - ✓ « **O** » : Le champ concerné est **Obligatoire**.
  - ✓ « **C** » : Le champ concerné est **Conditionnel**, la condition de présence est indiquée à la suite entre parenthèses.
  - ✓ « **F** » : Le champ concerné est **Facultatif**, il sera présent dans l'objet concerné s'il a été renseigné au niveau du Centre de Gestion de l'ASIP Santé
  - ✓ <C.G.> valeur définie, ou calculée, par le Centre de Gestion (ANS)

Des éléments généraux sont donnés sur la structuration des données en TLV, chapitre 8.2

## 2 Description générale

### 2.1 Applications de la carte

La carte CPS4 est la quatrième génération de carte CPS. Elle succède à la carte CPS3. C'est une carte multi-applicative contenant quatre applications :

- ✓ Une application dite régaliennes (CHIPDOC) conforme aux standards SSCD (Standards Européens EN419212), une application émulation CPS2TER et une application FIDO, toutes trois disponibles uniquement sur l'interface **contact**
- ✓ Une application DESFIRE EV3 disponible quant à elle uniquement sur l'interface **sans contact**.

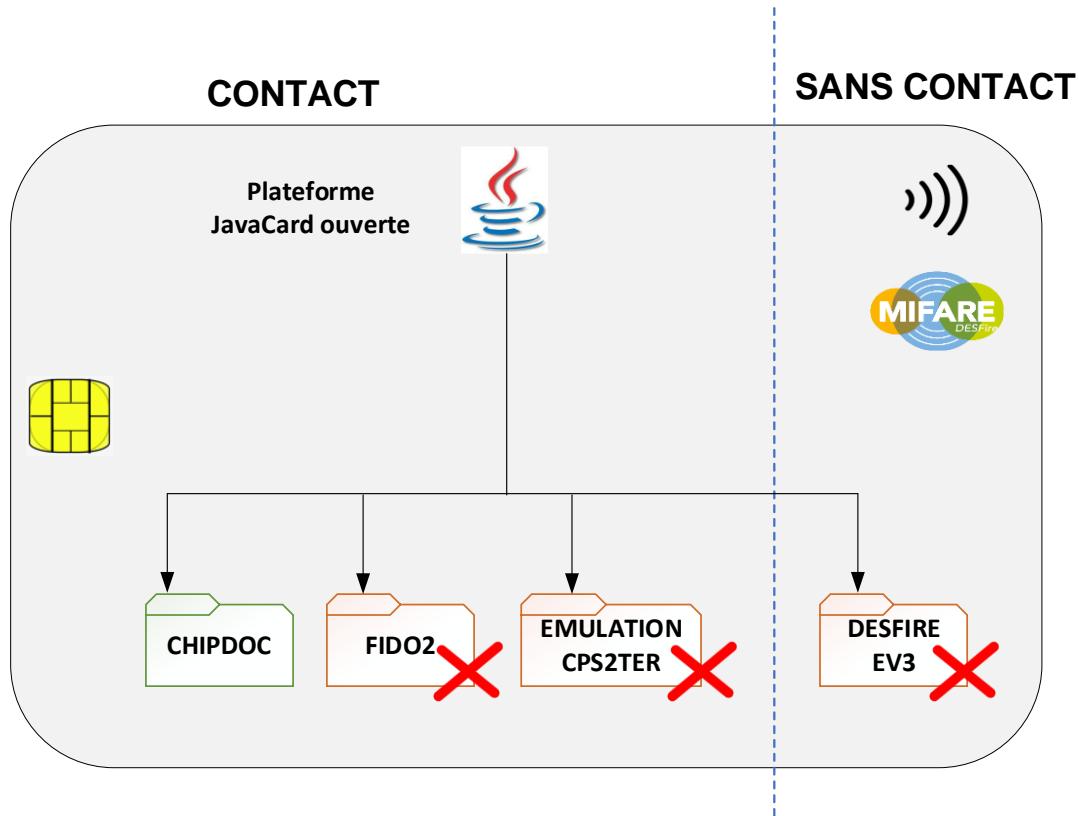


Figure 1 : Les applications de la carte CPS4

Ce document décrit uniquement les éléments techniques associés à **l'application régaliennes** de la carte CPS4 puisque c'est ce mode de fonctionnement qui est à terme ciblé (CHIPDOC).

L'application émulation CPS2ter est à usage exclusif de l'ANS/GIE dans le cadre d'une compatibilité ascendante des composants du GIE/SV avec la carte CPS3.

L'application FIDO, bien qu'offrant de nouvelles possibilités en termes d'authentification, n'est pas liée aux usages métiers.

## 2.2 Arborescence

L'application régaliennne contenue dans la carte CPS4 est initialisée avec une arborescence compatible au standard ISO/IEC 7816-15. Cette application n'est disponible que sur l'interface contact, l'interface sans contact est réservée quant à elle au DESFIRE.

L'application régaliennne utilise l'identifiant d'application (AID) suivant : **80 25 00 00 01 FF 01 00**

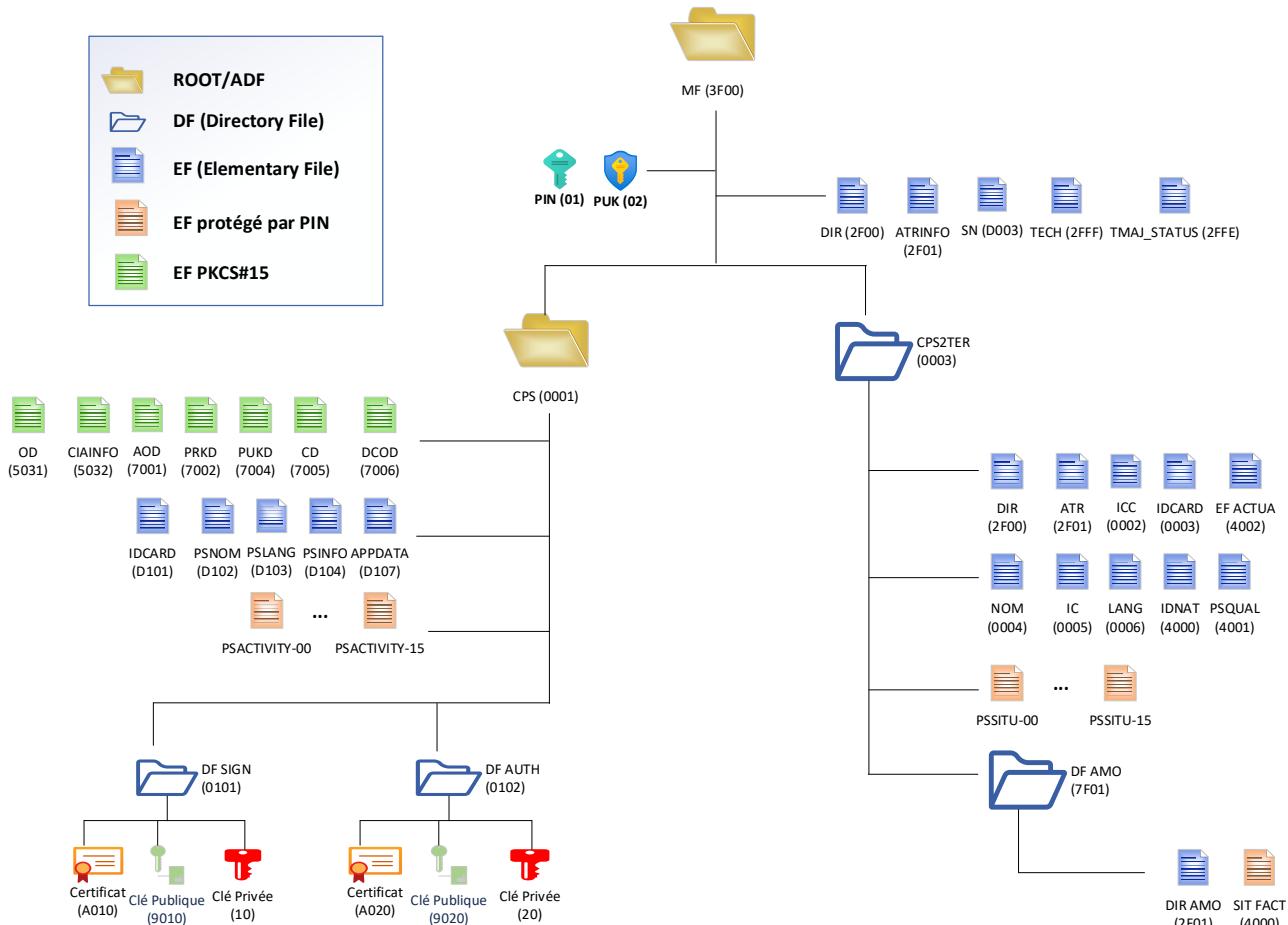


Figure 2 : Structure de fichiers/objets contenus dans l'application régaliennne



Deux remarques par rapport à l'arborescence ci-dessus :

- 1) Le PIN est en fait un PIN global à toutes les applications de la carte. Il est présenté ici (sous CHIPDOC) par souci de simplification.
- 2) Il existe d'autres objets (répertoires, fichiers et objets de sécurité) qui ne sont pas représentés ici car ne font pas l'objet de ce document (exemple : clés mises en œuvre dans le cadre de la Télé Mise à Jour ...).

## 2.3 Conditions d'accès

Pour chaque objet (répertoire, fichier, clé ...) les conditions d'accès définissent, par rapport à une **opération**, les **conditions** à mettre en œuvre pour pouvoir effectuer cette opération.

### 2.3.1 Opérations vs Objets

Sur la carte CPS V4, voici la liste des opérations selon le type d'objet considéré. Pour chaque type d'objet, trois opérations (ou plutôt ensemble d'opérations) sont autorisées : OP1, OP2 et OP3 :

MF	Opération	Description
OP1	Non utilisé	
OP2	CREATE	Création de EF, clé ...
OP3	DELETE	Suppression de toute l'arborescence sous le MF

ADF/DF	Opération	Description
OP1	ACTIVATE	Activation du DF
OP2	CREATE	Créations de EF ou Clé sous ce ADF/DF
OP3	DEACTIVATE	Désactivation du DF

EF	Opération	Description
OP1	READ	Lecture
OP2	WRITE	Ecriture
OP3	DELETE	Suppression

PIN/CLE SYMETRIQUE	Opération	Description
OP1	USE	Utilisation de la clé
OP2	CHANGE	Changer la valeur de la clé
OP3	UNLOCK	Débloquer la clé

CLE RSA	Opération	Description
OP1	USE	Utilisation de la clé
OP2	CHANGE	Changer la valeur de la clé
OP3	DELETE	Suppression de la clé

### 2.3.2 Conditions

Pour ce qui est des conditions à mettre en œuvre pour autoriser l'opération (il est possible de spécifier un OU/ET logique entre plusieurs opérations) :

	Description
ALWAYS	Opération toujours <b>autorisée</b>
NEVER	Opération toujours <b>interdite</b>
PIN	Opération autorisée si <b>PIN</b> présenté avec succès auparavant
PUK	Opération autorisée si <b>PUK</b> présenté avec succès auparavant
AD	Opération autorisée si authentification avec clé symétrique <b>ADMIN (AD)</b> effectuée auparavant
SA	Opération autorisée si authentification avec clé symétrique <b>SUPER ADMIN (SA)</b> effectuée auparavant

### 3 Eléments sous le MF (Master File)

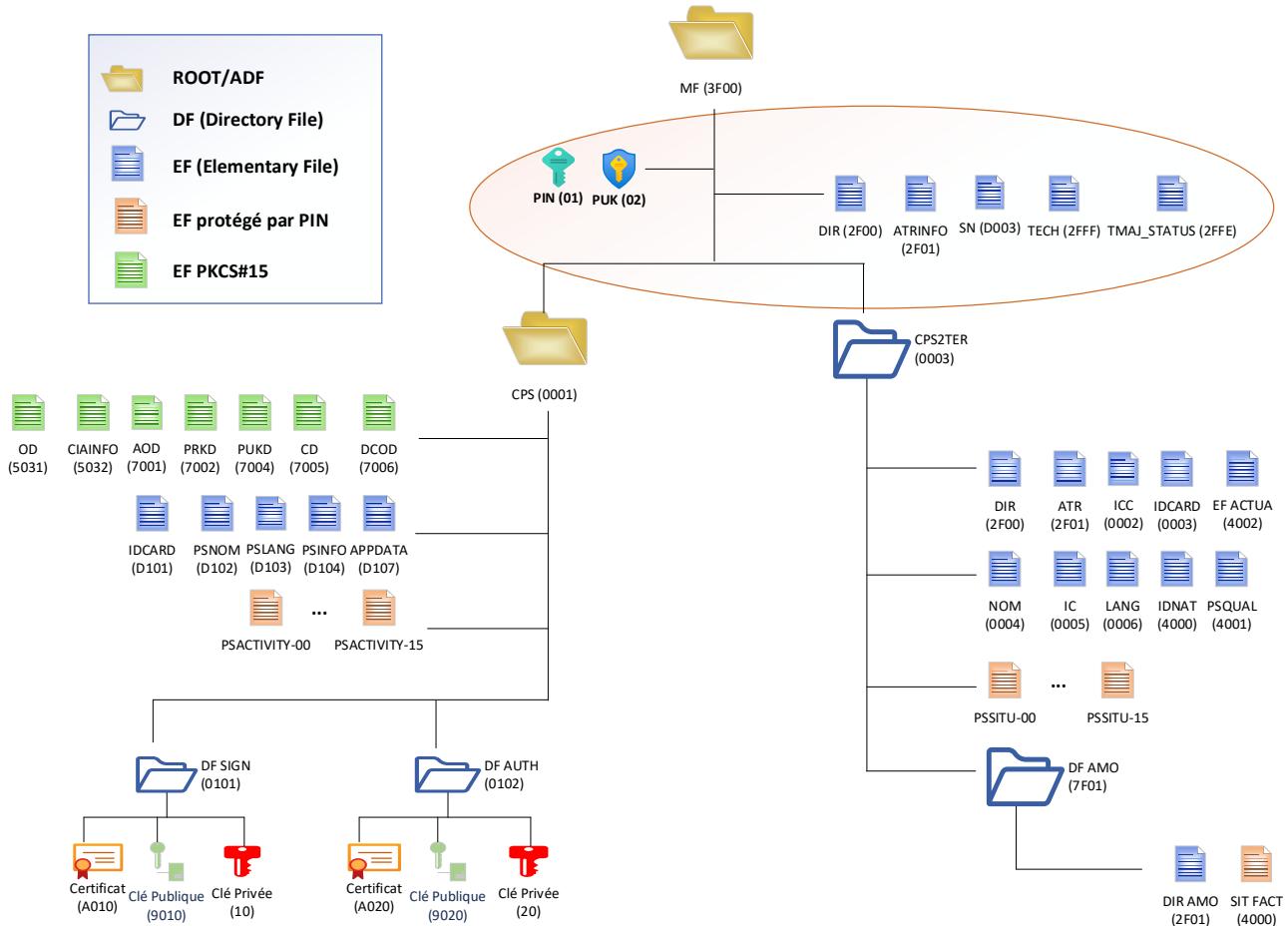


Figure 3 : Structure de fichiers/objets contenus sous le MF

## 3.1 Fichiers

### 3.1.1 EF ATR [2F01]

EF ATR est un fichier au format BER-TLV permettant d'obtenir des informations sur la carte CPS.

## Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF ATR
Identificateur de fichier (chemin complet)	'3F00/2F01'
Type de fichier	EF transparent
Taille du fichier	18
Référence PKCS#11	<a href="#">CPS_ATR</a>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

## Contenu du fichier

Tag	Length	Description	Value
'43'	'01'	Card service data tag	
		Card service data byte	<p>'B8'</p> <ul style="list-style-type: none"> <li>• <b>Application Selection:</b> by full DF Name</li> <li>• <b>DOs available:</b> in EF.DIR, in EF.ATR/INFO</li> <li>• Card with MF</li> </ul>
'47'	'03'	Card capabilities tag	
		Card capabilities data byte 1 → <i>Selection method</i>	<p>'38'</p> <ul style="list-style-type: none"> <li>• <b>DF selection:</b> by path, by file identifier</li> <li>• Implicit DF selection</li> </ul>
		Card capabilities data byte 2 → <i>Data coding byte</i>	<p>'21'</p> <ul style="list-style-type: none"> <li>• <b>Behaviour of write functions:</b> Proprietary</li> <li>• <b>Value 'FF' for the first byte of BER-TLV tag fields:</b> Invalid</li> <li>• Data unit size in quartets: 1</li> </ul>
		Card capabilities data byte 3 → <i>Miscellaneous</i>	<p>'92'</p> <ul style="list-style-type: none"> <li>• Command Chaining</li> <li>• <b>Logical channel number assignment:</b> by the card</li> <li>• <b>Maximum number of logical channels:</b> 2</li> </ul>
'4F'	'08'	Application Identifier (Application sélectionnée implicitement)	<p>'8025000001FF0002'</p> <p>AID of CPS2terLight applet</p>

### 3.1.2 EF SN [D003]

Le fichier EF SN est un fichier transparent contenant un numéro unique d'identification. Le numéro de série est encodé conformément à la spécification du PAN défini dans l'ISO/IEC 7812.

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF SN
Identificateur de fichier (chemin complet)	'3F00/ <b>D003</b> '
Type de fichier	EF transparent
Taille du fichier	'0C'
Référence PKCS#11	<b>CPS_ID_TECH</b>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

#### Contenu du fichier

Tag	Length	Description	Value
'5A'	'0A'	Serial Number tag	
		Identifiant ANS (5 octets)	'8025000001'
		Numéro unique (4 octets)	NNNNNNNN
		Clé de LUHN 4 bits + padding 'F'	LL

### 3.1.3 EF DIR [2F00]

Le fichier EF DIR est un fichier transparent qui contient la définition des applications pouvant être sélectionnées. Les informations contenues dans ce fichier sont décrites selon la syntaxe ASN.1 suivant la norme ISO/IEC 7816-15.

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF DIR
Identificateur de fichier (chemin complet)	'3F00/2F00'
Type de fichier	EF transparent
Taille du fichier	Variable
Référence PKCS#11	<a href="#">CPS_DIR</a>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

#### Contenu du fichier

Tag	Length	Description	Value
'61'	'15'	Entrée application CPS :	
		AID tag: '4F0D'	'E8 28 BD 08 0F 80 25 00 00 01 FF 00 10'
		PATH tag: '5104'	'3F00 0001'

### 3.1.4 EF TECH [2FFF]

Le fichier EF TECH est un fichier transparent qui contient des informations sur le profil électrique et graphique de la carte.

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF TECH
Identificateur de fichier (chemin complet)	'3F00/2FFF'
Type de fichier	EF transparent
Taille du fichier	100
Référence PKCS#11	<b>TECH</b>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

#### Contenu du fichier

Tag	Length	Description	Value
'F0'	'4B'	Tag - Len	
		'800C'	Numéro de support au format 'AAQQQLNNNNNN' où : <ul style="list-style-type: none"> <li>• AA = millésime (numérique)</li> <li>• QQ = numéro de jour dans l'année (numérique)</li> <li>• L = identifiant du numéroteur (alphanumérique)</li> <li>• NNNNNN = numéro de séquence dans la journée (numérique)</li> </ul>
		'8108'	Date d'émission au format ASCII 'AAAAMMJJ'
		'8204'	'30313031' : Version du profil au format ASCII (0101)
		'9F7F2A'	Valeur du CPLC en fin de perso

Structure du CPLC

	Ordre	L	Signification	Format
Fondeur	1	2	Fondeur	b16 BCD
	2	2	Type de composant (CI)	b16 BCD
	3	2	Identité de l'OS	b16
	4	2	Date de version de l'OS	b16 'YDDD'
	5	2	Version d'OS	b16
	6	2	Date de fabrication CI	b16 'YDDD'
	7	4	Numéro de série CI	b32
	8	2	Numéro de lot CI	b16
Encarteur (Pre-Personnalisateur)	9	2	Fabriquant de module	b16 BCD
	10	2	Date de fabrication module	b16 'YDDD'
	11	2	Encarteur	b16 BCD
	12	2	Date d'encartage	b16 'YDDD'
	13	2	Pré-personnalisateur	b16 BCD
	14	2	Date de pré-personnalisation	b16 'YDDD'
	15	4	Id de l'équipement de pré-personnalisation	b32
Personnalisateur	16	2	Personnalisateur	b16 BCD
	17	2	Date de personnalisation	b16 'YDDD'
	18	4	Id de l'équipement de personnalisation	b32

Tableau 3 : CPLC

### 3.1.5 EF TMAJ\_STATUS [2FFE]

Le fichier EF TMAJ\_STATUS est un fichier transparent qui contient l'état de la carte par rapport à une éventuelle TMAJ (Télé mise à jour).

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF TMAJ_STATUS
Identificateur de fichier (chemin complet)	'3F00/2FFE'
Type de fichier	EF transparent
Taille du fichier	8
Référence PKCS#11	<b>TMAJ_STATUS</b>
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: NEVER

#### Contenu du fichier

Tag	Length	Value
Statut	2	Statut de la TMAJ. Le premier octet est le complément (à 1) du second. Le second octet indique si une TMAJ est en cours. '00' indique pas de TMAJ en cours, tout autre valeur indique qu'une TMAJ est en cours. En sortie de personnalisation, on aura donc les 2 premiers octets à 'FF00' : pas de TMAJ en cours, carte fonctionnelle
Compteur	2	Compteur de TMAJ, indique le nombre de TMAJ réalisé. En sortie de personnalisation, ce compteur est à 1 ('0001').
Date	4	Date de la dernière TMAJ au format DCB : 'AAAAAMMJ'



**Les solutions qui n'utilisent pas la CRYPTOLIB ANS (Dispositifs intégrés ...) doivent considérer que la carte CPS V4 n'est exploitable que si une TMAJ n'est pas en cours, c'est-à-dire que les 2 premiers octets du fichier TMAJ STATUS sont à 'FF00'.**

## 3.2 Objets de sécurité

### 3.2.1 PIN

En fait le PIN est un PIN global au sens Global Plateforme et donc commun à toutes les applications de la carte.

Le PIN correspond au code porteur. Il est nécessaire pour authentifier le porteur de la carte. La présentation de ce PIN est notamment requise pour utiliser la clé privée de signature ou d'authentification ainsi que pour lire les fichiers métiers de situations.

#### Propriétés

Propriété	Valeur
Dénomination fonctionnelle	PIN
Identificateur	'01'
Maximum number of tries	3
Remaining tries counter	3
PIN / password maximum size (in bytes)	4
Attribut de sécurité en contact	Use: ALWAYS Change: PIN or PUK or AD or SA Unlock : PUK or AD or SA

### 3.2.2 PUK

Le PUK correspond au code de déblocage de la carte. Il est utilisé pour déverrouiller le PIN lorsque celui-ci est bloqué à la suite de 3 présentations erronées successives.

#### Propriétés

Propriété	Valeur
Dénomination fonctionnelle	PUK
Identificateur	'02'
Maximum number of tries	10
Remaining tries counter	10
PIN / password maximum size (in bytes)	8
Attribut de sécurité en contact	Use: ALWAYS Change: NEVER Unlock: NEVER

## 4 Eléments sous le ADF CPS (0001)

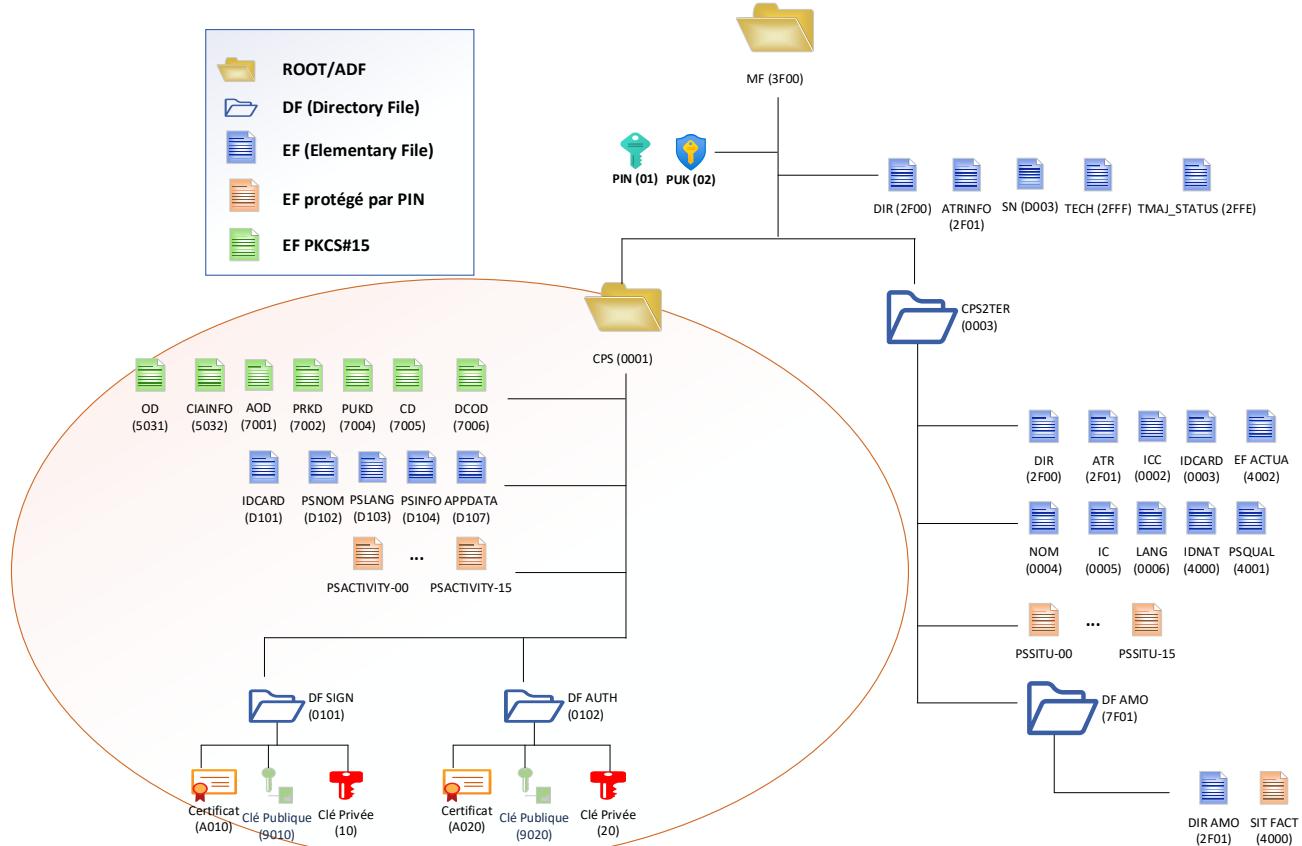


Figure 4 : Structure de fichiers/objets contenus sous ADF CPS (0001)

### Propriétés

Propriété	Valeur
Dénomination fonctionnelle	ADF CPS
Identificateur du DF (chemin complet)	'3F00/0001'
AID	'E828BD080F8025000001FF0010'
Attribut de sécurité en contact	ACTIVATE : SA CREATE : SA or AD DEACTIVATE : SA

## 4.1 Fichiers PKCS#15

### 4.1.1 EF OD [5031]

Le fichier EF OD est un fichier transparent qui contient des références vers les autres fichiers de l'application (PrKD, PuKD, CD, AOD ...). Les informations contenues dans ce fichier sont décrites selon la syntaxe ASN.1 suivant la norme ISO/IEC 7816-15.

Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF OD
Identificateur de fichier (chemin complet)	'3FOO/0001/ <b>5031</b> '
Type de fichier	EF transparent
Taille du fichier	Variable / Adaptée
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: SA or AD

Contenu du fichier

```

A0 06,  Private Keys
 30 04,  Path
    04 02,  EFidOrPath
      70 02
A1 06,  Public Keys
 30 04,  Path
    04 02,  EFidOrPath
      70 04
A4 06,  Certificates
 30 04,  Path
    04 02,  EFidOrPath
      70 05
A7 06,  Data Container Objects
 30 04,  Path
    04 02,  EFidOrPath
      70 06
A8 06,  Authentification Objects
 30 04,  Path
    04 02,  EFidOrPath
      70 01

```

#### 4.1.2 EF CIAINFO [5032]

Le fichier EF CIAINFO est un fichier transparent qui contient des informations sur la carte et l'application régalienne. Il contient en particulier l'étiquette de la carte, ses capacités et la liste des algorithmes supportés par l'application. Les informations contenues dans ce fichier sont décrites selon la syntaxe ASN.1 suivant la norme ISO/IEC 7816-15. Le contenu est au format « Hexadecimal DER-encoding, PKCS#15 notation ».

## Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF CIAINFO
Identificateur de fichier (chemin complet)	'3F00/0001/5032'
Type de fichier	EF transparent
Taille du fichier	Variable / Adaptée
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: SA or AD

## Contenu du fichier

```
30 2B,    CIAInfo
 02 01,    Version
 01,      <V2>
 04 0A,    Serial Number*
    xx xx
 0C 03,    Manufacturer ID
    41 4E 53 ,    <ANS>
 80 11,    Label*
    43 50 53 34 76 31 2D xx xx
 03 02,    CardFlags
    06 40 ,    <(AuthRequired)>
```

\*La donnée contenu dans le champs « Serial Number » est le contenu du fichier EF.SN

\*Le label est composé du préfixe « CPS4v1-» suivit de identifiant logique de la carte contenu dans le fichier EF.id-card, tag '81'



Le préfixe du label est amené à évoluer, en fonction des évolutions mineures de la carte CPS v4 : « CPS4v1- » deviendra « CPS4v2- », puis « CPS4v3- », etc...

#### 4.1.3 EF AOD [7001]

Le fichier EF AOD est un fichier transparent qui répertorie et décrit l'ensemble des objets d'authentification contenus dans la carte. Les informations contenues dans ce fichier sont décrites selon la syntaxe ASN.1 suivant la norme ISO/IEC 7816-15.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF AOD
Identificateur de fichier (chemin complet)	'3F00/0001/ <b>7001</b> '
Type de fichier	EF transparent
Taille du fichier	Variable / Adaptée
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: SA or AD

##### Contenu du fichier

```

30 4C,    CIOAuthenticationObject
 30 2D,    CommonObjectAttributes
    0C 03,    Label
      50 49 4E ,  <PIN>
    03 02,    CommonObjectFlags
      06 40 ,  <(Modifiable)>
    04 01,    AuthId
      C2
 30 1F,    AccessControlRules
    30 06,    AccessControlRule
      03 02,    AccessMode
        05 20 ,  <(Execute)>
      05 00,    SecurityCondition Always

    30 0C,    AccessControlRule
      03 02,    AccessMode
        06 40 ,  <(Update)>
    A2 06,    OrSecurityCondition
      04 01,    SecurityCondition AuthId
        C1
      04 01,    SecurityCondition AuthId
        C2
 30 07,    AccessControlRule
    03 02,    AccessMode
      03 08 ,  <(Attribute)>
    04 01,    SecurityCondition AuthId
      C2

```

```
30 03, CommonAuthenticationObjectAttributes
 04 01, AuthId
  C1
A1 16, PasswordAttributes
 30 14,
    03 03, PasswordFlags
    02 08 1C ,
<(Initialized|ExchangeRefData|ResetRetryCounter1|ResetRetryCounter2)>
  0A 01, PasswordType
    01, <ascii-numeric>
  02 01, MinLength
    04, <4>
  02 01, StoredLength
    04, <4>
  02 01, MaxLength
    04, <4>
  80 01, PasswordReference
    01, <1>

30 30, CIOAuthenticationObject
 30 12, CommonObjectAttributes
  0C 03, Label
    50 55 4B , <PUK>
  03 01, CommonObjectFlags
    00, <()>
  30 08, AccessControlRules
    30 06, AccessControlRule
      03 02, AccessMode
      05 20 , <(Execute)>
      05 00, SecurityCondition Always

 30 03, CommonAuthenticationObjectAttributes
  04 01, AuthId
  C2
A1 15, PasswordAttributes
 30 13,
    03 02, PasswordFlags
    00 19 , <(UnblockDisabled|Initialized|SoPassword)>
  0A 01, PasswordType
    01, <ascii-numeric>
  02 01, MinLength
    08, <8>
  02 01, StoredLength
    08, <8>
  02 01, MaxLength
    08, <8>
  80 01, PasswordReference
    02, <2>
```

#### 4.1.4 EF PrKD [7002]

Le fichier EF PrKD est un fichier transparent qui répertorie et décrit l'ensemble des clés privées contenues dans la carte. Il contient en particulier des références sur clés et, s'ils sont nécessaires, des pointeurs vers des objets d'identification utilisés pour protéger l'utilisation de ces clés. Les informations contenues dans ce fichier sont décrites selon la syntaxe ASN.1 suivant la norme ISO/IEC 7816-15.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF PrKD
Identificateur de fichier (chemin complet)	'3FO0/0001/7002'
Type de fichier	EF transparent
Taille du fichier	Variable / Adaptée
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: SA or AD

##### Contenu du fichier

```

30 59, CIORSAPrivateKey
30 20, CommonObjectAttributes
    0C 0C, Label
        43 50 53 5F 50 52 49 56 5F 53 49 47 , <CPS_PRIV_SIG>
    03 02, CommonObjectFlags
        07 80 , <(Private)>
    04 01, AuthId
        C1
    30 09, AccessControlRules
        30 07, AccessControlRule
            03 02, AccessMode
                05 20 , <(Execute)>
            04 01, SecurityCondition AuthId
                C1
    30 1F, CommonKeyAttributes
        04 0E, ID
            E8 28 BD 08 0F 80 25 00 00 01 FF 00 10 10
        03 03, KeyUsageFlags
            06 00 40 , <(NonRepudiation)>
        01 01, Native
            FF, <true>
        03 02, KeyAccessFlags
            04 B0 , <(Sensitive|AlwaysSensitive|NeverExtractable)>
        02 01, KeyReference
            10, <10>
    A0 02, CommonPrivateKeyAttributes
        30 00,
    A1 10, RSAPrivateKeyAttributes
        30 0E,
            30 08, Path
                04 06, EFidOrPath
                    3F 00 00 01 01 01
            02 02, Modulus Length
                08 00 , <2048>
    30 58, CIORSAPrivateKey
    30 20, CommonObjectAttributes
        0C 0C, Label
            43 50 53 5F 50 52 49 56 5F 41 55 54 , <CPS_PRIV_AUT>
        03 02, CommonObjectFlags
            07 80 , <(Private)>
        04 01, AuthId
            C1
        30 09, AccessControlRules
            30 07, AccessControlRule
                03 02, AccessMode
                    05 20 , <(Execute)>
                04 01, SecurityCondition AuthId

```

```

C1
30 1E, CommonKeyAttributes
 04 0E, id
    E8 28 BD 08 0F 80 25 00 00 01 FF 00 10 20
  03 02, KeyUsageFlags
    02 64, <(Decipher|Sign|KeyDecipher)>
  01 01, Native
    FF, <true>
  03 02, KeyAccessFlags
    04 B0, <(Sensitive|AlwaysSensitive|NeverExtractable)>
  02 01, KeyReference
    20, <20>
A0 02, CommonPrivateKeyAttributes
  30 00,
A1 10, RSAPrivateKeyAttributes
  30 0E,
    30 08, Path
      04 06, EFidOrPath
        3F 00 00 01 01 02
  02 02, Modulus Length
    08 00, <2048>

```



Les ID de clé mentionnés ci-dessus :

- 'E8 28 BD 08 0F 80 25 00 00 01 FF 00 10 10' ID pour la clé de signature
- 'E8 28 BD 08 0F 80 25 00 00 01 FF 00 10 20' ID pour la clé d'authentification

Sont remplacés par les ID suivants :

- 'E8 28 BD 08 0F 80 25 00 00 01 FF 00 10 01' ID pour la clé de signature
- 'E8 28 BD 08 0F 80 25 00 00 01 FF 00 10 02' ID pour la clé d'authentification

Afin d'avoir les mêmes ID que la CPS V3.

#### 4.1.5 EF PuKD [7004]

Le fichier EF PuKD est un fichier transparent qui répertorie et décrit l'ensemble des clés publiques contenu dans la carte. Il contient en particulier des références sur des clés. Les informations contenues dans ce fichier sont décrites selon la syntaxe ASN.1 suivant la norme ISO/IEC 7816-15.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF PuKD
Identificateur de fichier (chemin complet)	'3FO0/0001/7004'
Type de fichier	EF transparent
Taille du fichier	1
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: SA or AD

##### Contenu du fichier

Constitué d'un seul octet à '00', pas d'informations sur les clés publiques.

#### 4.1.6 EF CD [7005]

Le fichier EF CD est un fichier transparent qui répertorie et décrit l'ensemble des certificats contenus dans la carte. Les informations contenues dans ce fichier sont décrites selon la syntaxe ASN.1 suivant la norme ISO/IEC 7816-15.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF CD
Identificateur de fichier (chemin complet)	'3F00/0001/7005'
Type de fichier	EF transparent
Taille du fichier	Variable / Adaptée
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: SA or AD

##### Contenu du fichier

```

30 51, CIOAuthenticationObject
  30 2A, CommonObjectAttributes
    0C 1B, Label
      43 65 72 74 69 66 69 63 61 74 20 64 65 20 53 69 67 6E 61 74 75 72 65 20 43 50
  53, <Certificat de Signature CPS>
    03 01, CommonObjectFlags
      00, <()>
    30 08, AccessControlRules

      30 06, AccessControlRule
        03 02, AccessMode
          07 80, <(Read)>
        05 00, SecurityCondition Always

    30 13, CommonCertificateAttributes
      04 0E, ID
        E8 28 BD 08 0F 80 25 00 00 01 FF 00 10 10
      01 01, Authority
        00, <false>
    A1 0E, X509CertificateAttributes
      30 0C,
        30 0A, Path
          04 08, EFidOrPath
            3F 00 00 01 01 01 A0 10

  30 57, CIOAuthenticationObject
    30 30, CommonObjectAttributes
      0C 21, Label
        43 65 72 74 69 66 69 63 61 74 20 64 27 41 75 74 68 65 6E 74 69 66 69 63 61 74
    69 6F 6E 20 43 50 53, <Certificat d'Authentification CPS>
      03 01, CommonObjectFlags
        00, <()>
    30 08, AccessControlRules
      30 06, AccessControlRule
        03 02, AccessMode
          07 80, <(Read)>
        05 00, SecurityCondition Always

    30 13, CommonCertificateAttributes
      04 0E, ID
        E8 28 BD 08 0F 80 25 00 00 01 FF 00 10 20
      01 01, Authority
        00, <false>
    A1 0E, X509CertificateAttributes
      30 0C,
        30 0A, Path
          04 08, EFidOrPath
            3F 00 00 01 01 02 A0 20

```



Les ID de clé mentionnés ci-dessus :

- 'E8 28 BD 08 OF 80 25 00 00 01 FF 00 10 **10**' ID pour le certificat de signature
- 'E8 28 BD 08 OF 80 25 00 00 01 FF 00 10 **20**' ID pour le certificat d'authentification

Sont remplacés par les ID suivants :

- 'E8 28 BD 08 OF 80 25 00 00 01 FF 00 10 **01**' ID pour le certificat de signature
- 'E8 28 BD 08 OF 80 25 00 00 01 FF 00 10 **02**' ID pour le certificat d'authentification

#### 4.1.7 EF DCOD [7006]

Le fichier EF DCOD est un fichier transparent qui répertorie et décrit l'ensemble des objets contenus dans la carte. Les informations contenues dans ce fichier sont décrites selon la syntaxe ASN.1 suivant la norme ISO/IEC 7816-15.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF DCOD
Identificateur de fichier (chemin complet)	'3F00/0001/ <b>7006</b> '
Type de fichier	EF transparent
Taille du fichier	Variable / Adaptée
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: SA or AD

##### Contenu du fichier

```

30 29, OpaqueDO
30 16, CommonObjectAttributes
0C 07, Label
43 50 53 5F 44 49 52, <CPS_DIR>
03 01, CommonObjectFlags
00, <()>
30 08, AccessControlRules
30 06, AccessControlRule
03 02, AccessMode
07 80, <(Read)>
05 00, SecurityCondition Always
30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 08, OpaqueDO Attributes
30 06, Path
04 04, EFidOrPath
3F 00 2F 00

30 29, OpaqueDO
30 16, CommonObjectAttributes
0C 07, Label
43 50 53 5F 41 54 52, <CPS_ATR>
03 01, CommonObjectFlags
00, <()>
30 08, AccessControlRules
30 06, AccessControlRule
03 02, AccessMode
07 80, <(Read)>
05 00, SecurityCondition Always
30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 08, OpaqueDO Attributes
30 06, Path
04 04, EFidOrPath
3F 00 2F 01

30 2D, OpaqueDO
30 1A, CommonObjectAttributes
0C 0B, Label
43 50 53 5F 49 44 5F 54 45 43 48, <CPS_ID_TECH>
03 01, CommonObjectFlags
00, <()>
30 08, AccessControlRules
30 06, AccessControlRule
03 02, AccessMode
07 80, <(Read)>

```

```
        05 00, SecurityCondition Always
30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 08, OpaqueDO Attributes
30 06, Path
04 04, EFidOrPath
3F 00 D0 03

30 26, OpaqueDO
30 13, CommonObjectAttributes
0C 04, Label
54 45 43 48, <TECH>
03 01, CommonObjectFlags
00, <(>)
30 08, AccessControlRules
30 06, AccessControlRule
03 02, AccessMode
07 80, <(Read)>
05 00, SecurityCondition Always
30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 08, OpaqueDO Attributes
30 06, Path
04 04, EFidOrPath
3F 00 2F FF

30 2D, OpaqueDO
30 1A, CommonObjectAttributes
0C 0B, Label
54 4D 41 4A 5F 53 54 41 54 55 53, <TMAJ_STATUS>
03 01, CommonObjectFlags
00, <(>)
30 08, AccessControlRules
30 06, AccessControlRule
03 02, AccessMode
07 80, <(Read)>
05 00, SecurityCondition Always

30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 08, OpaqueDO Attributes
30 06, Path
04 04, EFidOrPath
3F 00 2F FE

30 2D, OpaqueDO
30 1A, CommonObjectAttributes
0C 0B, Label
43 50 53 32 54 45 52 5F 4D 41 50, <CPS2TER_MAP>
03 01, CommonObjectFlags
00, <(>)
30 08, AccessControlRules
30 06, AccessControlRule
```

```
03 02, AccessMode
07 80, <(Read)>
05 00, SecurityCondition Always

30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 08, OpaqueDO Attributes
30 06, Path
04 04, EFidOrPath
3F 00 2F FD

30 2B, OpaqueDO
30 18, CommonObjectAttributes
0C 09, Label
50 55 42 4B 45 59 5F 43 41, <PUBKEY_CA>
03 01, CommonObjectFlags
00, <()>
30 08, AccessControlRules
30 06, AccessControlRule
03 02, AccessMode
07 80, <(Read)>
05 00, SecurityCondition Always

30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 08, OpaqueDO Attributes
30 06, Path
04 04, EFidOrPath
3F 00 CA 01

30 2F, OpaqueDO
30 1A, CommonObjectAttributes
0C 0B, Label
43 50 53 5F 49 44 5F 43 41 52 44, <CPS_ID_CARD>
03 01, CommonObjectFlags
00, <()>
30 08, AccessControlRules
30 06, AccessControlRule
03 02, AccessMode
07 80, <(Read)>
05 00, SecurityCondition Always

30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 0A, OpaqueDO Attributes
30 08, Path
04 06, EFidOrPath
3F 00 00 01 D1 01

30 2F, OpaqueDO
30 1A, CommonObjectAttributes
0C 0B, Label
43 50 53 5F 4E 41 4D 45 5F 50 53, <CPS_NAME_PS>
03 01, CommonObjectFlags
00, <()>
```

```
30 08, AccessControlRules
  30 06, AccessControlRule
    03 02, AccessMode
      07 80, <(Read)>
      05 00, SecurityCondition Always
  30 05, CommonDataContainerObjectAttributes
    0C 03, Application Name
      43 50 53, <CPS>
  A1 0A, OpaqueDO Attributes
    30 08, Path
      04 06, EFidOrPath
        3F 00 00 01 D1 02

  30 2F, OpaqueDO
    30 1A, CommonObjectAttributes
      0C 0B, Label
        43 50 53 5F 4C 41 4E 47 5F 50 53, <CPS_LANG_PS>
      03 01, CommonObjectFlags
        00, <()>
    30 08, AccessControlRules
      30 06, AccessControlRule
        03 02, AccessMode
          07 80, <(Read)>
          05 00, SecurityCondition Always
    30 05, CommonDataContainerObjectAttributes
      0C 03, Application Name
        43 50 53, <CPS>
    A1 0A, OpaqueDO Attributes
      30 08, Path
        04 06, EFidOrPath
          3F 00 00 01 D1 03

  30 2F, OpaqueDO
    30 1A, CommonObjectAttributes
      0C 0B, Label
        43 50 53 5F 49 4E 46 4F 5F 50 53, <CPS_INFO_PS>
      03 01, CommonObjectFlags
        00, <()>
    30 08, AccessControlRules
      30 06, AccessControlRule
        03 02, AccessMode
          07 80, <(Read)>
          05 00, SecurityCondition Always
    30 05, CommonDataContainerObjectAttributes
      0C 03, Application Name
        43 50 53, <CPS>
    A1 0A, OpaqueDO Attributes
      30 08, Path
        04 06, EFidOrPath
          3F 00 00 01 D1 04

  30 36, OpaqueDO
    30 21, CommonObjectAttributes
      0C 08, Label
        43 50 53 5F 44 41 54 41, <CPS_DATA>
      03 02, CommonObjectFlags
```

```
06 40, <(Modifiable)>
30 11, AccessControlRules
  30 06, AccessControlRule
    03 02, AccessMode
      07 80, <(Read)>
    05 00, SecurityCondition Always

  30 07, AccessControlRule
    03 02, AccessMode
      06 40, <(Update)>
    04 01, SecurityCondition AuthId
      C1
  30 05, CommonDataContainerObjectAttributes
    0C 03, Application Name
      43 50 53, <CPS>
  A1 0A, OpaqueDO Attributes
    30 08, Path
      04 06, EFidOrPath
      3F 00 00 01 D1 07
  30 2F, OpaqueDO
    30 1A, CommonObjectAttributes
      0C 0B, Label
        43 50 53 32 54 45 52 5F 44 49 52, <CPS2TER_DIR>
    03 01, CommonObjectFlags
      00, <()>
    30 08, AccessControlRules
      30 06, AccessControlRule
        03 02, AccessMode
          07 80, <(Read)>
        05 00, SecurityCondition Always
    30 05, CommonDataContainerObjectAttributes
      0C 03, Application Name
        43 50 53, <CPS>
  A1 0A, OpaqueDO Attributes
    30 08, Path
      04 06, EFidOrPath
      3F 00 00 03 2F 00

  30 2F, OpaqueDO
    30 1A, CommonObjectAttributes
      0C 0B, Label
        43 50 53 32 54 45 52 5F 41 54 52, <CPS2TER_ATR>
    03 01, CommonObjectFlags
      00, <()>
    30 08, AccessControlRules
      30 06, AccessControlRule
        03 02, AccessMode
          07 80, <(Read)>
        05 00, SecurityCondition Always
    30 05, CommonDataContainerObjectAttributes
      0C 03, Application Name
        43 50 53, <CPS>
  A1 0A, OpaqueDO Attributes
    30 08, Path
      04 06, EFidOrPath
      3F 00 00 03 2F 01
```

```
30 2F, OpaqueDO
  30 1A, CommonObjectAttributes
    0C 0B, Label
      43 50 53 32 54 45 52 5F 49 43 43, <CPS2TER_ICC>
    03 01, CommonObjectFlags
      00, <()>
  30 08, AccessControlRules
    30 06, AccessControlRule
      03 02, AccessMode
        07 80, <(Read)>
      05 00, SecurityCondition Always
  30 05, CommonDataContainerObjectAttributes
    0C 03, Application Name
      43 50 53, <CPS>
  A1 0A, OpaqueDO Attributes
    30 08, Path
      04 06, EFidOrPath
        3F 00 00 03 00 02

30 2E, OpaqueDO
  30 19, CommonObjectAttributes
    0C 0A, Label
      43 50 53 32 54 45 52 5F 49 44, <CPS2TER_ID>
    03 01, CommonObjectFlags
      00, <()>
  30 08, AccessControlRules
    30 06, AccessControlRule
      03 02, AccessMode
        07 80, <(Read)>
      05 00, SecurityCondition Always
  30 05, CommonDataContainerObjectAttributes
    0C 03, Application Name
      43 50 53, <CPS>
  A1 0A, OpaqueDO Attributes
    30 08, Path
      04 06, EFidOrPath
        3F 00 00 03 00 03

30 30, OpaqueDO
  30 1B, CommonObjectAttributes
    0C 0C, Label
      43 50 53 32 54 45 52 5F 4E 41 4D 45, <CPS2TER_NAME>
    03 01, CommonObjectFlags
      00, <()>
  30 08, AccessControlRules
    30 06, AccessControlRule
      03 02, AccessMode
        07 80, <(Read)>
      05 00, SecurityCondition Always
  30 05, CommonDataContainerObjectAttributes
    0C 03, Application Name
      43 50 53, <CPS>
  A1 0A, OpaqueDO Attributes
    30 08, Path
      04 06, EFidOrPath
```

```

3F 00 00 03 00 04

30 2E, OpaqueDO
  30 19, CommonObjectAttributes
    0C 0A, Label
      43 50 53 32 54 45 52 5F 49 43, <CPS2TER_IC>
    03 01, CommonObjectFlags
      00, <()>
  30 08, AccessControlRules
    30 06, AccessControlRule
      03 02, AccessMode
        07 80, <(Read)>
      05 00, SecurityCondition Always
  30 05, CommonDataContainerObjectAttributes
    0C 03, Application Name
      43 50 53, <CPS>
A1 0A, OpaqueDO Attributes
  30 08, Path
    04 06, EFidOrPath
      3F 00 00 03 00 05

30 30, OpaqueDO
  30 1B, CommonObjectAttributes
    0C 0C, Label
      43 50 53 32 54 45 52 5F 4C 41 4E 47, <CPS2TER_LANG>
    03 01, CommonObjectFlags
      00, <()>
  30 08, AccessControlRules
    30 06, AccessControlRule
      03 02, AccessMode
        07 80, <(Read)>
      05 00, SecurityCondition Always
  30 05, CommonDataContainerObjectAttributes
    0C 03, Application Name
      43 50 53, <CPS>
A1 0A, OpaqueDO Attributes
  30 08, Path
    04 06, EFidOrPath
      3F 00 00 03 00 06

30 32, OpaqueDO
  30 1D, CommonObjectAttributes
    0C 0E, Label
      43 50 53 32 54 45 52 5F 50 53 49 4E 46 4F, <CPS2TER_PSINFO>
    03 01, CommonObjectFlags
      00, <()>
  30 08, AccessControlRules
    30 06, AccessControlRule
      03 02, AccessMode
        07 80, <(Read)>
      05 00, SecurityCondition Always
  30 05, CommonDataContainerObjectAttributes
    0C 03, Application Name
      43 50 53, <CPS>
A1 0A, OpaqueDO Attributes
  30 08, Path

```

```
04 06, EFidOrPath
3F 00 00 03 40 00

30 34, OpaqueDO
30 1F, CommonObjectAttributes
0C 10, Label
43 50 53 32 54 45 52 5F 50 53 51 55 41 4C 49 46, <CPS2TER_PSQUALIF>
03 01, CommonObjectFlags
00, <()>
30 08, AccessControlRules
30 06, AccessControlRule
03 02, AccessMode
07 80, <(Read)>
05 00, SecurityCondition Always
30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 0A, OpaqueDO Attributes
30 08, Path
04 06, EFidOrPath
3F 00 00 03 40 01

30 34, OpaqueDO
30 1F, CommonObjectAttributes
0C 10, Label
43 50 53 32 54 45 52 5F 44 49 52 5F 52 45 53 50, <CPS2TER_DIR_RESP>
03 01, CommonObjectFlags
00, <()>
30 08, AccessControlRules
30 06, AccessControlRule
03 02, AccessMode
07 80, <(Read)>
05 00, SecurityCondition Always

30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 0A, OpaqueDO Attributes
30 08, Path
04 06, EFidOrPath
3F 00 00 03 40 20

30 34, OpaqueDO
30 1D, CommonObjectAttributes
0C 0E, Label
43 50 53 32 54 45 52 5F 44 49 52 41 4D 4F, <CPS2TER_DIRAMO>
03 01, CommonObjectFlags
00, <()>
30 08, AccessControlRules
30 06, AccessControlRule
03 02, AccessMode
07 80, <(Read)>
05 00, SecurityCondition Always
30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
```

```
A1 0C, OpaqueDO Attributes
 30 0A, Path
 04 08, EFidOrPath
 3F 00 00 03 7F 01 2F 00

30 34, OpaqueDO
 30 1D, CommonObjectAttributes
 0C 0C, Label
 43 50 53 5F 53 49 54 5F 46 41 43 54, <CPS_SIT_FACT>
 03 02, CommonObjectFlags
 07 80, <(Private)>
 30 09, AccessControlRules
 30 07, AccessControlRule
 03 02, AccessMode
 07 80, <(Read)>
 04 01, SecurityCondition AuthId
 C1
 30 05, CommonDataContainerObjectAttributes
 0C 03, Application Name
 43 50 53, <CPS>
A1 0C, OpaqueDO Attributes
 30 0A, Path
 04 08, EFidOrPath
 3F 00 00 03 7F 01 40 00

30 3A, OpaqueDO
 30 23, CommonObjectAttributes
 0C 14, Label
 43 50 53 32 54 45 52 5F 44 49 52 5F 52 45 53 50 5F 41 4D 4F,
<CPS2TER_DIR_RESP_AMO>
 03 01, CommonObjectFlags
 00, <()>
 30 08, AccessControlRules
 30 06, AccessControlRule
 03 02, AccessMode
 07 80, <(Read)>
 05 00, SecurityCondition Always

30 05, CommonDataContainerObjectAttributes
 0C 03, Application Name
 43 50 53, <CPS>
A1 0C, OpaqueDO Attributes
 30 0A, Path
 04 08, EFidOrPath
 3F 00 00 03 7F 01 40 20

30 31, OpaqueDO
 30 1C, CommonObjectAttributes
 0C 0B, Label
 44 45 53 46 49 52 45 5F 4B 45 59, <DESFIRE_KEY>
 03 02, CommonObjectFlags
 07 80, <(Private)>
 30 09, AccessControlRules
 30 07, AccessControlRule
 03 02, AccessMode
 07 80, <(Read)>
```

```
04 01, SecurityCondition AuthId
C1
30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 0A, OpaqueDO Attributes
30 08, Path
04 06, EFidOrPath
3F 00 00 02 C0 01
```

Ensuite pour chaque fichier EF.ps-activity\_xx (0 à 15) sous l'ADF CPS il contient :

```
30 38, OpaqueDO
30 23, CommonObjectAttributes
0C 12, Label
43 50 53 5F 41 43 54 49 56 49 54 59 5F 78 78 5F 50 53, <CPS_ACTIVITY_xx_PS>
03 02, CommonObjectFlags
07 80, <(Private)>
30 09, AccessControlRules
30 07, AccessControlRule
03 02, AccessMode
07 80, <(Read)>
04 01, SecurityCondition AuthId
C1
30 05, CommonDataContainerObjectAttributes
0C 03, Application Name
43 50 53, <CPS>
A1 0A, OpaqueDO Attributes
30 08, Path
04 06, EFidOrPath
3F 00 00 01 D1 2X
```

Avec XX l'id de 0 à 15.

Ensuite pour chaque fichier EF.2Ter-ps-sit\_xx (0 à 15) sous le DF CPS2Ter il contient :

```
30 3C, OpaqueDO
  30 27, CommonObjectAttributes
    0C 16, Label
      43 50 53 32 54 45 52 5F 41 43 54 49 56 49 54 59 5F 78 78 5F 60 53,
<CPS2TER_ACTIVITY_XX_PS>
  03 02, CommonObjectFlags
    07 80, <(Private)>
  30 09, AccessControlRules
    30 07, AccessControlRule
      03 02, AccessMode
        07 80, <(Read)>
      04 01, SecurityCondition AuthId
        C1
  30 05, CommonDataContainerObjectAttributes
    0C 03, Application Name

      43 50 53, <CPS>
A1 0A, OpaqueDO Attributes
  30 08, Path
    04 06, EFidOrPath
      3F 00 00 03 40 1x
```

Avec XX l'id de 0 à 15.

## 4.2 Fichiers de données métiers

### 4.2.1 EF ID\_CARTE [D101]

Le fichier EF ID\_CARTE est un fichier transparent qui contient l'identifiant logique de la carte.

Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	Identification carte
Identificateur de fichier (chemin complet)	'3FOO/0001/ <b>D101</b> '
Type de fichier	EF transparent
Taille du fichier	31
Référence PKCS#11	<b>CPS_ID_CARD</b>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

Contenu du fichier

TAG	Longueur	Nom du champ	Description et codification	Valeur	Format	Présence
'E3'	29	Template	Champ construit contenant l'ensemble des champs de donnée de l'objet			0
'80'	5	Id_Emet	Identifiant de l'émetteur carte : - IIN = '8025000001'	'8025000001'	BCD	0
	5	Id_Carte_Log	Identification logique de la carte (10 caractères numériques)	<C.G.>	BCD	0
	1	Catégorie_Carte	Catégorie de carte	<TAB G01>	HEX	0
	4	Date_Début_Val	1er jour de validité de la carte	'AAAAMMMJJ'	BCD	0
	4	Date_Fin_Val	Dernier jour de validité de la carte	'AAAAMMMJJ'	BCD	0

#### 4.2.2 EF NOM [D102]

Le fichier EF NOM est un fichier transparent qui contient l'identité du porteur de la carte.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	Caractéristiques porteur
Identificateur de fichier (chemin complet)	'3F00/0001/ <a href="#">D102</a> '
Type de fichier	EF transparent
Taille du fichier	178 ('B2' hexadécimal)
Référence PKCS#11	<a href="#">CPS_NAME_PS</a>
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: NEVER

##### Contenu du fichier

TAG	Long	Nom du champ	Description et codification	Valeur	Format	Présence
'E4'		Var.	Template	Champ construit contenant l'ensemble des champs de donnée de l'objet		O
'80'	1	Code civilité	Code civilité (M., Mme, Mlle, . . .) du porteur	<TAB G06>	HEX	O
	2 à 27	Nom patronyme	Nom patronymique	<C.G.>	αNUM	O
	2 à 27	Nom Marital	Nom marital	<C.G.>	αNUM	F
	2 à 83	Prénoms	1 à 3 prénoms : - Prénom 1 (2 à 27 caractères) - Séparateur 'FF' si suite - Prénom 2 (2 à 27 caractères) - Séparateur 'FF' si suite - Prénom 3 (2 à 27 caractères)	<C.G.>	αNUM HEX αNUM HEX αNUM	O C (si suite) F C (si suite) F
	2 à 27	Prénom Usuel	Prénom usuel	<C.G.>	αNUM	O



Fichier créé avec sa taille maximale, soit 'B2' octets. Si moins de données on complète avec des '00'.

### 4.2.3 EF LANG [D103]

Le fichier EF LANG est un fichier transparent qui contient les langues parlées par le porteur de la carte.

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	Codes langues
Identificateur de fichier (chemin complet)	'3F00/0001/ <a href="#">D103</a> '
Type de fichier	EF transparent
Taille du fichier	'0C'
Référence PKCS#11	<a href="#">CPS_LANG_PS</a>
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: NEVER

#### Contenu du fichier

TAG	Long	Nom du champ	Description et codification	Valeur	Format	Présence
'E5'	10	Template	Champ construit contenant l'ensemble des champs de donnée de l'objet			0
'80'	8	Codes Langues	4 codes langues selon la norme ISO 639-1. Chaque code est sur 2 caractères alphabétiques (minuscules obligatoires). Au moins une langue est renseignée. Les langues non renseignées sont initialisées à blanc ('20 20').	<TAB G00>	ALPHA	0



- Les langues sont codées suivant la norme ISO 639 (2 caractères minuscules par langue).
- Au moins une langue est renseignée, la langue par défaut est le français (fr).
- Les langues non renseignées sont initialisées à '2020'.

#### 4.2.4 EF INFO\_PS [D104]

Le fichier EF INFO\_PS est un fichier transparent qui contient le type de carte et les caractéristiques du Professionnel ou personnel de Santé porteur de la carte.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	Caractéristiques professionnelles PS
Identificateur de fichier (chemin complet)	'3F00/0001/ <b>D104</b> '
Type de fichier	EF transparent
Taille du fichier	'32'
Référence PKCS#11	<b>CPS_INFO_PS</b>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

##### Contenu du fichier

TAG	Long	Nom du champ	Description et codification	Valeur	Format	Présence
'ED'		Var.	Template	Champ construit contenant l'ensemble des champs de donnée de l'objet		O
'80'	1	Type_Carte_PS	Type de carte professionnelle	<TAB G02>	HEX	O
'81'	10 à 31	Id_Nat_PS	Identification nationale du porteur de la carte	<C.G.>	αNUM	O
'82'	1	Code Profession PS	Code profession dans le cas d'un PS (uniquement et obligatoirement pour les PS)	<TAB G15>	HEX	C (pour les PS uniquement)
'83'	1	Code Profession PF	Code profession dans le cas d'un PF (uniquement et obligatoirement pour les PF)	<TAB G16>	HEX	C (pour les PF uniquement)
'84'	2 à 27	Nom_Exerc	Nom d'exercice	<C.G.>	ASCII	O
'85'	2 à 10	Spec_Ord_RPPS	Spécialité Ordinale (uniquement et obligatoirement pour les Médecins)	<TAB R01>	ASCII	C (pour les Médecins uniquement)

#### 4.2.5 EF APP\_DATA [D107]

Le fichier EF APP\_DATA est un fichier transparent.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF DATA
Identificateur de fichier (chemin complet)	'3F00/0001/ <a href="#">D107</a> '
Type de fichier	EF transparent
Taille du fichier	1024
Référence PKCS#11	<a href="#">CPS_DATA</a>
Attribut de sécurité en contact	Read: ALWAYS Write: PIN Delete: NEVER



- La taille de ce fichier était de 4096 octets (4K) sur la carte CPS V3.
- Ce fichier est initialisé avec des '00' et est laissé libre à discréction des utilisateurs.

#### 4.2.6 EF PS\_SIT\_XX [D120-D12F]

Les fichiers EF PS\_SIT sont des fichiers transparents qui contiennent les situations d'exercice du Professionnel de Santé.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	Activité / Situation d'exercice N°XY (XY=01 à 16) du PS
Identificateur de fichier (chemin complet)	'3F00/0001/ <b>D120</b> ' – '3F00/0001/ <b>D12F</b> '
Type de fichier	EF transparent
Taille du fichier	
Référence PKCS#11	<b>CPS_ACTIVITY_xx_PS</b> (xx=00 à 15)
Attribut de sécurité en contact	Read: PIN Write: SA or AD Delete: SA or AD

##### Contenu du fichier

TAG	Long	Nom du champ	Description et codification	Valeur	Format	Présence
'EE'		Var.	Template	Champ construit contenant l'ensemble des champs de donnée de l'objet		O
'80'	1	Mode_Exerc	Mode d'Exercice (Libéral, Salarié, ...)	<TAB G17>	HEX	O
	1	Statut_Exerc	Code Statut	<TAB G04>	HEX	O
	1 à 10	Tabl_Pharm	Tableau de Pharmaciens applicable à cette structure (uniquement et obligatoirement pour les Pharmaciens)	<TAB G05>	ALPHA	C (pharmacien uniquement)
	1	No_Log_Sit	Numéro logique de la situation	<C.G.>	HEX	O
	2 à 38	Struct_Rais_Soc	Raison Sociale de la Structure ou mnémonique d'un cabinet libéral	<C.G.>	αNUM	C (présent si le mode d'exercice est différent de « remplaçant »)
'85'	10 à 15	Struct_Id_Nat	Identification nationale de la structure	<C.G.>	ASCII	C (présent si le mode d'exercice est différent de « remplaçant »)
'86'	2 à 10	Struct_Sect_Act_RPPS	Secteur d'activité de la structure	<TAB R02>	ASCII	C (présent si le mode d'exercice est différent de « remplaçant »)



Peut être complété par des '00'



Chaque fichier de situation est optionnel et peut donc être absent.

#### 4.2.7 DF SIG (0101)

Ce répertoire contient tous les objets liés à la paire de bi-clé de signature : certificat, clé publique et clé privée.

##### Propriétés

Propriété	Valeur
Dénomination fonctionnelle	DF SIG
Identificateur (chemin complet)	'3F00/0001/ <b>0101</b> '
AID	
Attribut de sécurité en contact	Activate: SA or AD Create: SA Deactivate : SA or AD

#### 4.2.8 EF cert-sig

Ce fichier contient le certificat X509 dédié à la signature électronique et associé au bi-clé de signature.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF cert-sig
Identificateur de fichier (chemin complet)	'3F00/0001/0101/ <b>A010</b> '
Type de fichier	EF transparent
Taille du fichier	Variable / ajustée
Référence PKCS#11	
Attribut de sécurité en contact	Read: ALWAYS Write: SA Delete: SA

#### 4.2.9 DF AUTH (0102)

Ce répertoire contient tous les objets liés à la paire de bi-clé d'authentification : certificat, clé publique et clé privée.

##### Propriétés

Propriété	Valeur
Dénomination fonctionnelle	DF AUTH
Identificateur (chemin complet)	'3F00/0001/ <b>0102</b> '
AID	
Attribut de sécurité en contact	Activate: SA or AD Create: SA Deactivate : SA or AD

#### 4.2.10 EF cert-auth

Ce fichier contient le certificat X509 dédié à l'authentification et associé au bi-clé d'authentification.

##### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF cert-auth
Identificateur de fichier (chemin complet)	'3F00/0001/0102/ <b>A020</b> '
Type de fichier	EF transparent
Taille du fichier	Variable / ajustée
Référence PKCS#11	
Attribut de sécurité en contact	Read: ALWAYS Write: SA Delete: SA

## 4.3 Objets de sécurité

### 4.3.1 Clé privée de signature Kpriv-sig

Cette clé est la clé privée du bi-clé RSA de signature.

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	Kpriv-sig
Identificateur de fichier	'10'
Type de fichier	Clé
Taille de la clé	2048 bits
Attribut de sécurité en contact	Use : PIN Change : SA Delete : SA

### 4.3.2 Clé publique de signature Kpub-sig

Cette clé est la clé publique du bi-clé RSA de signature.

#### Propriétés

Propriété	Valeur
Dénomination fonctionnelle	Kpub-sig
Identificateur de fichier	'3F00/0001/0101/9010'
Type de fichier	EF transparent
Taille du fichier	Variable / Ajustée
Attribut de sécurité en contact	Read: ALWAYS Write: SA Delete: SA



La clé est générée par la plateforme de personnalisation.

Elle est au format asn.1:

```
RSAPUBLICKEY ::= SEQUENCE {
    MODULUS      INTEGER, -- N
    PUBLICEXPONENT INTEGER -- E
}
```

Soit :

'30'	'xx'			
	'02'	'xx'	'xxxxxxxx...xxxx'	modulus 2048 bits
	'02'	'03'	'010001'	public exponent

### 4.3.3 Clé privée d'authentification Kpriv-auth

Cette clé est la clé privée du bi-clé RSA d'authentification.

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	Kpriv-auth
Identificateur de fichier	'20'
Type de fichier	Clé
Taille de la clé	2048 bits
Attribut de sécurité en contact	Use : PIN Change : SA Delete : SA

### 4.3.4 Clé publique d'authentification Kpub-auth

Cette clé est la clé publique du bi-clé RSA d'authentification.

#### Propriétés

Propriété	Valeur
Dénomination fonctionnelle	Kpub-auth
Identificateur de fichier (chemin complet)	'3F00/0001/0102/9020'
Type de fichier	EF transparent
Taille du fichier	Variable / Ajustée
Attribut de sécurité en contact	Read: ALWAYS Write: SA Delete: SA



La clé est générée par la plateforme de personnalisation.

Elle est au format asn.1:

```
RSAPUBLICKEY ::= SEQUENCE {
    MODULUS      INTEGER, -- N
    PUBLICEXPONENT  INTEGER -- E
}
```

Soit :

'30'	'xx'				
		'02'	'xx'	'xxxxxxxx...xxxx'	modulus 2048 bits
		'02'	'03'	'010001'	public exponent

## 5 Eléments sous le DF CPS2TER (0003)

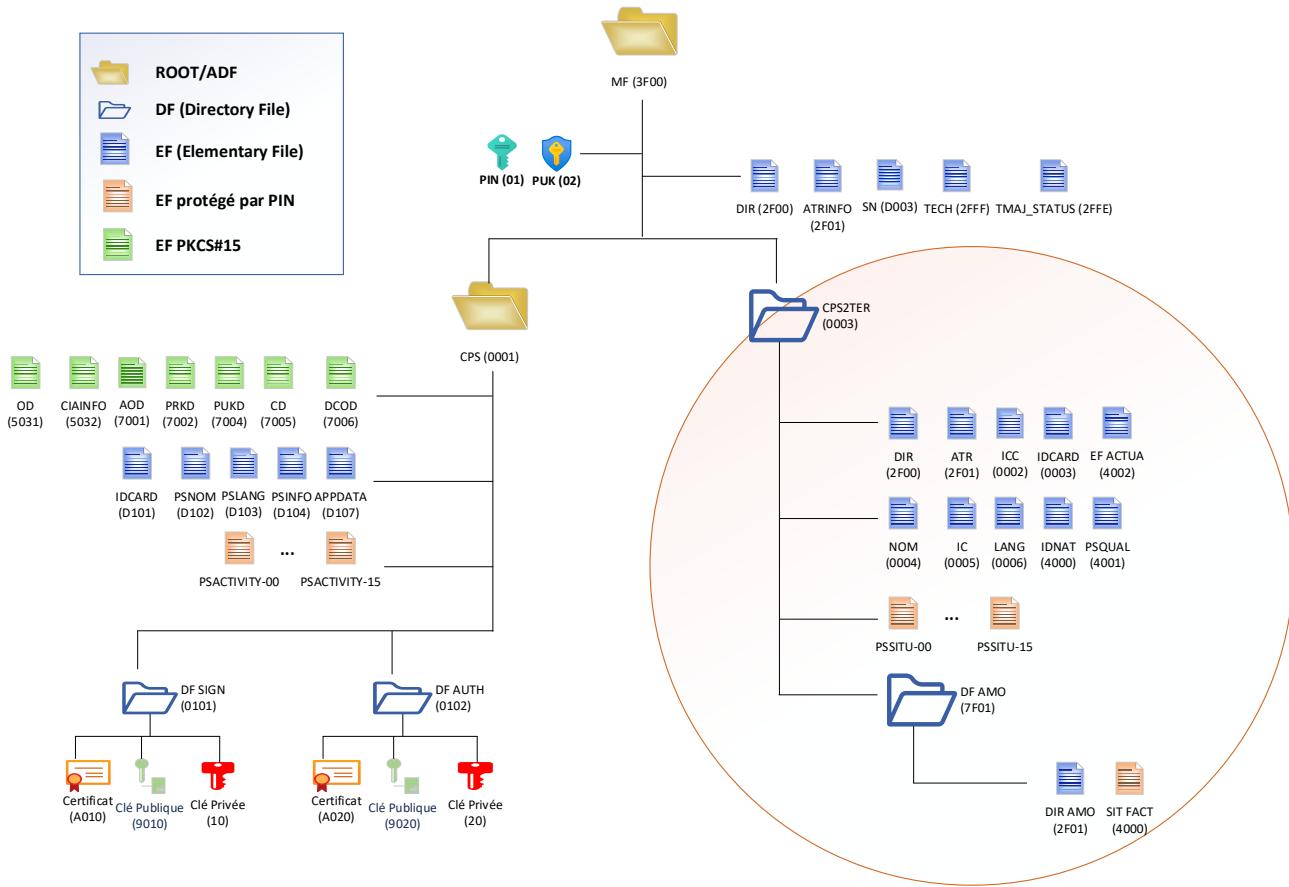


Figure 5 : Structure de fichiers/objets contenus sous DF CPS2TER (0003)

### 5.1 DF CPS2TER (0003)

Ce répertoire contient tous les fichiers de données de la CPS2TER (fichiers « miroir »).

#### Propriétés

Propriété	Valeur
Dénomination fonctionnelle	DF CPS2TER
Identificateur (chemin complet)	'3F00/0003'
AID	
Attribut de sécurité en contact	Activate: SA Create EF/KEYS: SA or AD Deactivate: SA

## 5.2 Fichiers de données métiers

### 5.2.1 EF DIR [2F00]

Le fichier EF DIR contient l'identification des applications disponibles et leur chemin d'accès.

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF DIR
Identificateur de fichier (chemin complet)	'3F00/0003/ <b>2F00</b> '
Type de fichier	EF transparent
Taille du fichier	'2B'
Référence PKCS#11	<b>CPS2TER_DIR</b>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

#### Contenu du fichier

TAG	Longueur	Nom du champ	Valeur	Format	Présence
'79'	10	Tag_Alloc_Auto	'4F 08 80 25 00 00 01 FF 00 02'	HEX	O
'61'	29	template Appli_AMO	'4F 08 A0 00 00 00 22 FF 10 00 50 0B 41 53 53 5F 4D 41 4C 5F 4F 42 4C 51 04 3F 00 7F 01'	HEX	O

### 5.2.2 EF ATR [2F01]

Le fichier EF ATR contient des informations supplémentaires qui feront partie des données ATR envoyées lors du RESET de la carte.

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF ATR
Identificateur de fichier (chemin complet)	'3F00/0003/ <b>2F01</b> '
Type de fichier	EF transparent
Taille du fichier	2
Référence PKCS#11	<b>CPS2TER_ATR</b>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

#### Contenu du fichier

Nom du champ	Description et codification	Valeur	Format	Présence
Code_Secteur_ATR	Code secteur (ISO 7812-2), indiquant le secteur auquel appartient la carte (ATR : octet T6). 80 ==> secteur Santé-Social	'80'	HEX	O
No_Vers_Cod_Opt	Non significatif	'00'	HEX	O

### 5.2.3 EF IC [0005]

Le fichier EF IC contient l'identification du composant.

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF IC
Identificateur de fichier (chemin complet)	'3F00/0003/ <b>0005</b> '
Type de fichier	EF transparent
Taille du fichier	18
Référence PKCS#11	<b>CPS2TER_IC</b>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

#### Contenu du fichier

TAG		Longueur	Description et codification	Valeur	Format	Présence
'E0'		16	Champ construit contenant l'ensemble des champs de donnée de l'objet			O
	'80'	7	<b>Id_Composant :</b> <ul style="list-style-type: none"> <li>• Type (1 octet) = '00'</li> <li>• Serial Number issu du CPLC (4 octets)</li> <li>• Lot Number issu du CPLC (2 octets)</li> </ul>	<TAB- Composants> <Fondeur> <Fondeur>	HEX	O
'81'	1		<b>Réf_Clé_Fab</b> = '00'	<Encarteur>	HEX	O
'82'	2		<b>CLCD Manufacturing date</b> = 'YDDD'	<Fondeur>	HEX	O

## 5.2.4 EF ICC [0002]

Le fichier EF ICC contient l'identification physique et les caractéristiques de la carte.

### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF ICC
Identificateur de fichier (chemin complet)	'3F00/0003/ <b>0002</b> '
Type de fichier	EF transparent
Taille du fichier	17
Référence PKCS#11	<b>CPS2TER_ICC</b>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

### Contenu du fichier

TAG		Longueur	Description et codification	Valeur	Format	Présence
'E1'		15				O
	'80'	5	<b>Id_Carte_Phys :</b> • id encarteur (1 octet) = '00' • N° Série issu du CPLC (4 octets)	<TAB-Sociétés> <Encarteur>	HEX	O
	'81'	3	<b>Réf_Logos</b>	<TAB-Logos>	HEX	O
	'82'	1	<b>Réf_Clé_Lot</b>	'00'	HEX	O

## 5.2.5 EF ACTUA [4002]

Le fichier EF DIR contient l'identification des applications disponibles et leur chemin d'accès.

### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF DACTUA
Identificateur de fichier (chemin complet)	'3F00/0003/ <b>4002</b> '
Type de fichier	EF transparent
Taille du fichier	'14'
Référence PKCS#11	
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: NEVER

### Contenu du fichier

TAG	Longueur	Description et codification	Valeur	Format	Présence
'EC'	'12'				O
'81'	7	Date/heure début d'actualisation	'AAAAMMJJHHMMSS'	DCB	O
'82'	7	Date/heure de fin d'actualisation	'AAAAMMJJHHMMSS'	DCB	O



Au début d'une procédure de TMAJ, le premier champ (tag '81') est renseigné avec la date/heure et le second champ (tag '82' est nul ('8200')). Ce second champ est renseigné à la fin de la procédure de TMAJ avec la date/heure de fin.

Le fichier EF ACTUA est créé lors de la première actualisation d'une carte (il n'est donc pas créé lors de la personnalisation).



Comme pour la CPS V3, les solutions qui n'utilisent pas la CRYPTOLIB ANS (Dispositifs intégrés ...) doivent considérer que la carte CPS V4 n'est exploitable que si une TMAJ n'est pas en cours, c'est-à-dire que le tag '82' n'est pas valorisé à '8200'.

## 5.2.6 EF ID-CARTE [0003]

Le fichier EF ID-CARTE contient l'identification logique de la carte.

### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF ID-CARTE
Identificateur de fichier (chemin complet)	'3FOO/0003/0003'
Type de fichier	EF transparent
Taille du fichier	32
Référence PKCS#11	<a href="#">CPS2TER_ID</a>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

### Contenu du fichier

TAG	Longueur	Description et codification	Valeur	Format	Présence
'E3'	30				O
'80'	6	<b>Identifiant de l'émetteur carte :</b> <ul style="list-style-type: none"><li>• IIN = '8025000001'</li><li>• Clé de luhn</li><li>• padding</li></ul>	'80250000017F'	BCD	O
'81'	5	<b>Identification logique de la carte (10 caractères numériques)</b>	<C.G.>	BCD	O
'82'	1	<b>Catégorie de carte</b>	<TAB G01>	HEX	O
'83'	4	1er jour de validité de la carte	'AAAAMMMJJ'	BCD	O
'84'	4	Dernier jour de validité de la carte	'AAAAMMMJJ'	BCD	O

### 5.2.7 EF NAME [0004]

Le fichier EF NAME contient l'identité du porteur.

#### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF NAME
Identificateur de fichier (chemin complet)	'3FOO/0003/0004'
Type de fichier	EF transparent
Taille du fichier	'B2'
Référence PKCS#11	<a href="#">CPS2TER_NAME</a>
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: NEVER

#### Contenu du fichier

TAG	Longueur	Description et codification	Valeur	Format	Présence
'E4'	Variable				O
'80'	1	<b>Code civilité</b> (M., Mme, Mlle, . . .) du porteur	<TAB G06>	HEX	O
'81'	2 à 27	<b>Nom patronymique</b>	<C.G.>	αNUM	O
'82'	2 à 27	<b>Nom marital</b>	<C.G.>	αNUM	F
'83'	2 à 83	1 à 3 <b>prénoms</b> : - Prénom 1 (2 à 27 caractères) - Séparateur 'FF' si suite - Prénom 2 (2 à 27 caractères) - Séparateur 'FF' si suite - Prénom 3 (2 à 27 caractères)	<C.G.>	αNUM HEX αNUM HEX αNUM	O C (si suite) F C (si suite) F
'84'	2 à 27	<b>Prénom usuel</b>	<C.G.>	αNUM	O



Taille maximale 'B2', complété par des '00' si moins de données utiles.

## 5.2.8 EF LANG [0006]

Le fichier EF LANG contient les langues parlées par le porteur de la carte.

### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF LANG
Identificateur de fichier (chemin complet)	'3F00/0003/0006'
Type de fichier	EF transparent
Taille du fichier	12
Référence PKCS#11	<a href="#">CPS2TER_LANG</a>
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: NEVER

### Contenu du fichier

TAG	Longueur	Description et codification	Valeur	Format	Présence
'E5'	10				O
'80'	5	Codes langues : 4 codes langues selon la norme ISO 639-1. Chaque code est sur 2 caractères alphabétiques (minuscules obligatoires). Au moins une langue est renseignée. Les langues non renseignées sont initialisées à blanc ('20 20').	<TAB G00>	ALPHA	O



- Les langues sont codées suivant la norme ISO 639 (2 caractères minuscules par langue).
- Au moins une langue est renseignée, la langue par défaut est le français (fr).
- Les langues non renseignées sont initialisées à '2020'.

## 5.2.9 EF PS-IDNAT [4000]

Le fichier EF PS-IDNAT contient le type de carte et les caractéristiques du Professionnel ou personnel de Santé porteur de la carte.

### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF PS-IDNAT
Identificateur de fichier (chemin complet)	'3F00/0003/ <b>4000</b> '
Type de fichier	EF transparent
Taille du fichier	'47'
Référence PKCS#11	<a href="#">CPS2TER_PSINFO</a>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

### Contenu du fichier

TAG	Longueur	Description et codification	Valeur	Format	Présence
'E6'		Variable			O
'80'	2	<b>Type de carte professionnelle</b> <ul style="list-style-type: none"> <li>• Type</li> <li>• Modèle :</li> <li>- Carte de remplacement</li> <li>- Carte de structure</li> <li>- 6 bits RFU</li> </ul>	<TAB G02>	HEX	O
		<b>IDNAT</b> Identification nationale du porteur de la carte			
		<b>Code profession PS</b> (Uniquement et obligatoirement pour les PS et PF)			
		<b>Nom d'exercice</b> du porteur.			



Taille maximale '47', complété par des '00'

## 5.2.10 EF PS-QUALIF [4001]

Le fichier EF PS-QUALIF contient des informations supplémentaires par rapport à la profession qu'exerce le Professionnels de Santé. Ce fichier est présent uniquement dans les cartes CPS.

### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF PS-QUALIF
Identificateur de fichier (chemin complet)	'3F00/0003/ <b>4001</b> '
Type de fichier	EF transparent
Taille du fichier	'1A'
Référence PKCS#11	<b>CPS2TER_PSQUALIF</b>
Attribut de sécurité en contact	Read: ALWAYS Write: SA or AD Delete: NEVER

### Contenu du fichier

TAG	Longueur	Description et codification	Valeur	Format	Présence
'E7'	Variable				O
'80'	1	<b>Spécialité de Qualification</b>	<TAB G12>	HEX	C
'84'	1	<b>Nature de Qualification</b>	<TAB G11>	HEX	C
'85'	2 ou 4	2 couples possibles de Compétence et Nature de Compétence Compétence 1 Nature 1 Compétence 2 Nature 2	<TAB G12> <TAB G11> <TAB G12> <TAB G11>	HEX	F



Taille maximale '1A', complété par des '00'



Fichier optionnel, peut être absent

## 5.2.11 EF PS-SITxx [401X]

Les fichiers EF PS-SITxx contiennent les situations d'exercice des Professionnels de Santé.

### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF PS-SITxx du PS (xx=00 à 15)
Identificateur de fichier (chemin complet)	'3F00/0003/ <b>4010</b> ' – '3F00/0003/ <b>401F</b> '
Type de fichier	EF transparent
Taille du fichier	Variable / Ajusté
Référence PKCS#11	<b>CPS2TER_ACTIVITY_xx_PS</b> (xx=00 à 15)
Attribut de sécurité en contact	Read: PIN Write: SA or AD Delete: SA or AD

### Contenu du fichier

TAG		Longueur	Description et codification	Valeur	Format	Présence
'E8'		Variable				
'80'	2	<b>Mode d'exercice :</b> Statut dans l'établissement	<TAB G17> <TAB G04>	HEX	O	
'81'	10	<b>Caractéristiques de la structure :</b> Secteur d'activité Valeur RFU Type identifiant Identifiant Padding	<TAB G19> 'FF' <TAB G07> <C.G.>	HEX	C (présent si le mode d'exercice est différent de « remplaçant »)	
'82'	2	<b>Spécialisation d'exercice :</b> Nature de qualification	<TAB G12> <TAB G11>	HEX	F	
'83'	1 à 2	<b>Code tableau des pharmaciens</b>	<C.G.>	ALPHA	C (pharmaciens uniquement)	
'84'	1	<b>Numéro logique de la situation</b>	<C.G.>	HEX	O	
'85'	6	Identification nationale du Professionnel Santé remplaçant ou remplacé	<C.G.>	HEX	F	
'88'	2 à 38	<b>Raison sociale de la structure</b>	<C.G.>	αNUM	C (présent si le mode d'exercice est différent de « remplaçant »)	



Peut être complété par des '00'



Chaque fichier de situation est optionnel et peut donc être absent.

## 5.2.12 DF AMO [7F01]

### Propriétés

Propriété	Valeur
Dénomination fonctionnelle	DF AMO
Identificateur (chemin complet)	'3F00/0003/ <b>7F01</b> '
AID	
Attribut de sécurité en contact	Activate: SA or AD Create EF/KEYS: SA or AD Deactivate: SA or AD

## 5.2.13 EF DIRAMO [2F00]

### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF DIRAMO
Identificateur de fichier (chemin complet)	'3F00/0003/7F01/ <b>2F00</b> '
Type de fichier	EF transparent
Taille du fichier	12
Référence PKCS#11	<b>CPS2TER_DIRAMO</b>
Attribut de sécurité en contact	Read: ALWAYS Write: NEVER Delete: NEVER

### Contenu du fichier

TAG	Longueur	Valeur	Format	Présence
'79'	'0A'	'4F08A000000022FF0001'	HEX	O

## 5.2.14 EF sit-fact [4000]

### Propriétés du fichier

Propriété	Valeur
Dénomination fonctionnelle	EF sit fact
Identificateur de fichier (chemin complet)	'3F00/0003/7F01/ <b>4000</b> '
Type de fichier	EF transparent
Taille du fichier	Variable / Ajustée
Référence PKCS#11	<b>CPS_SIT_FACT</b>
Attribut de sécurité en contact	Read: PIN Write: SA or AD Delete: SA or AD



Si taille des données utiles est < 203 octets, alors :

- La taille du fichier est de 203 octets.
- Les données du fichier correspondent aux données utiles, complétées par des zéros.

- Si taille des données utiles est  $\geq$  406 octets, alors :

- La taille du fichier correspond à celle des données utiles.
- Les données du fichier correspondent aux données utiles.

## 6 APDU de référence

### 6.1 Introduction

Les APDU traitées dans ce chapitre couvrent uniquement les APDU les plus fréquemment utilisées dans le cadre du profil CPS.



Ces APDU sont données à titre d'exemples et uniquement pour les utilisateurs **ne pouvant pas utiliser la CRYPTOLIB ANS**.



Contrairement à la carte CPS V3, l'application régaliennne n'est pas sélectionnée par défaut. Il faut donc avant de pouvoir l'utiliser, sélectionner cette application. L'AID de l'application régaliennne est '**8025000001FF0100**' (Par défaut, sur la CPS V4 c'est l'application émulation CPS2TER qui est sélectionnée). Attention, une seule instance de l'application régaliennne peut être sélectionnée sur la carte à un instant donné. Ce qui veut dire que si une autre application (CRYPTOLIB ANS par exemple ...) a sélectionné l'application régaliennne sur un canal autre que le canal sur lequel on veut travailler, le SELECT échouera (pour information la CRYPTOLIB ANS utilise le canal 1).



Dans la description des commandes, l'octet **CLA** est mentionné à '00', on peut aussi utiliser le canal 1, dans ce cas l'octet **CLA** est à renseigner à '01'

### 6.2 Protocole

Les composants métiers qui ne passent pas par l'abstraction ANS (CRYPTOLIB) doivent être indépendant du protocole utilisé par la carte. En clair, les protocoles T=0 et T=1 doivent être gérés.

La commande GET RESPONSE, décrite dans le chapitre suivant, n'est mise en œuvre que lorsque le protocole T=0 est utilisé.

Les retours (SW) suivants sont intrinsèques au protocole T=0 et peuvent donc être reçus lors de l'émission de toute commande (en T=0).

Paramètre	Valeur	Remarque
SW	'61XX'	En T=0, <b>SW2</b> indique le nombre d'octets disponibles, à récupérer par une commande GET RESPONSE.
	'6CXX'	En T=0, champ <b>Le</b> incorrect. <b>SW2</b> indique le nombre d'octets disponibles pour la commande utilisée (READ BINARY ...)

## 6.3 GET RESPONSE

Cette commande inhérente au protocole T = 0 restitue les données de réponse en fonction de la commande précédente.

Cette commande suit une commande dont le retour est un **SW** à '61XX'

Paramètre	Valeur	Remarque
CLA	'00'	
INS	'C0'	Commande GET RESPONSE
P1	'00'	
P2	'00'	
Lc		Absent
Data		Absent
Le		Longueur du champ Data attendu en réponse

Tableau 4 : GET RESPONSE format de la commande

Paramètre	Valeur	Remarque
Data		Champ de données, Le octets
SW	'61XX'	En T=0, <b>SW2</b> indique le nombre d'octets disponibles, à récupérer par une commande GET RESPONSE.
	'6CXX'	En T=0, champ <b>Le</b> incorrect. <b>SW2</b> indique le nombre d'octets disponibles pour la commande utilisée.
	'6D00'	SW_INS_NOT_SUPPORTED
	'9000'	Succès, pas d'erreur

Tableau 5 : GET RESPONSE format de la réponse

## 6.4 Commandes de gestion du système de fichiers

### 6.4.1 SELECT

Cette commande permet de sélectionner un répertoire (DF), le Master File (MF), une application (ADF) ou un fichier (EF). Elle affecte donc le fichier et/ou répertoire courant.

#### Format de la commande

Paramètre	Valeur	Remarque
CLA	'00'	
INS	'A4'	Commande SELECT
P1		Mode de sélection, voir ci-dessous
P2	'00' ou '0C'	'00' : retour du FCI dans la réponse '0C' : pas de FCI dans la réponse
Lc		Longueur du champ Data si présent
Data		Champ de données, Lc octets
Le		Absent ou Longueur du champ Data attendu (FCI)

Tableau 6 : SELECT format de la commande

Paramètre	Valeur	Remarque
Data		Champ de données, Le octets
SW	'6283'	SW_SELECTED_FILE_DEACTIVATED
	'6700'	SW_WRONG_LENGTH
	'6981'	SW_WRONG_FILE_TYPE
	'6982'	SW_SECURITY_STATUS_NOT_SATISFIED
	'6986'	SW_COMMAND_NOT_ALLOWED
	'6A82'	SW_FILE_NOT_FOUND
	'6A86'	SW_INCORRECT_P1P2
	'6A87'	SW_LC_INCONSISTENT_WITH_P1P2
	'9000'	Succès, pas d'erreur

Tableau 7 : SELECT format de la réponse

P1	Mode	Champ Data
'00'	Sélection EF/DF sous DF courant	'XXXX' EF/DF File ID
'00'	Sélection implicite du MF	Pas de champ Data
'00'	Sélection explicite du MF	'3F00' MF File ID
'01'	Sélection d'un DF	'XXXX' DF File ID
'02'	Sélection d'un EF	'XXXX' EF File ID
'03'	Sélection du DF père	Pas de champ Data
'04'	Sélection d'un ADF	Nom de l'ADF, 1 à 16 octets
'08'	Sélection d'un EF/DF par chemin absolu	Liste de File ID (sans inclure le MF)
'09'	Sélection d'un EF/DF par chemin relatif	Liste de File ID

Tableau 8 : – SELECT Valeurs possibles de P1, mode de sélection

Tag	Lg	Signification	Présence	
			EF	DF
'6F'	L		✓	✓
		<b>Tag</b>   <b>Lg</b>   <b>Signification</b>		
		'80'   '03'   Nombre d'octets du fichier	✓	✗
		'81'   '02'   Nombre de DF et EF sous ce DF	✗	✓
		'82'   '03' ou '01'   '38' pour DF et '010000' pour EF transparent.	✓	✓
		'83'   '02'   Identifiant du fichier (File ID)	✓	✓
		'86'   '03'   Attributs de sécurité	✓	✓

Tableau 9 : – SELECT format du FCI

### 6.4.2 READ BINARY

Cette commande permet de lire des données dans le fichier courant.

Paramètre	Valeur	Remarque
CLA	'00'	
INS	'B0'	Commande READ BINARY
P1 P2		0XXXXXXXXX : XXXXXXXX P1 : P2 = offset dans le fichier sur 15 bits ('0000' – '7FFF') <b>Les SFI (Short File Identifier) ne sont pas utilisés sur la CPS V4</b>
Lc		Absent
Data		Absent
Le		Longueur des données à lire

Tableau 10 : READ BINARY format de la commande

Paramètre	Valeur	Remarque
Data		Champ de données, Le octets
SW	'6700'	SW_WRONG_LENGTH
	'6982'	SW_SECURITY_STATUS_NOT_SATISFIED
	'6986'	SW_COMMAND_NOT_ALLOWED
	'6988'	SW_SM_OBJECT_ERROR
	'6A82'	SW_FILE_NOT_FOUND
	'6B00'	SW_EF_SCOPE_EXCEEDED
	'9000'	Succès, pas d'erreur

Tableau 11 : READ BINARY format de la réponse

## 6.5 Commandes d'authentification de l'utilisateur

### 6.5.1 VERIFY

Cette commande permet d'authentifier l'utilisateur par présentation du code PIN, ou de connaître le nombre d'essais restants.

Paramètre	Valeur	Remarque
CLA	'00'	
INS	'20'	Commande VERIFY
P1	'00'	
P2		'01' pour PIN et '02' pour PUK
Lc		Longueur du PIN (4) /PUK (8)
Data		PIN au format ASCII ( <b>pas de padding à 'FF'</b> )
Le		Absent

Tableau 12 : VERIFY format de la commande

Paramètre	Valeur	Remarque
Data		Absent
SW	'6300'	SW_VERIFICATION_FAILED
	'63Cx'	SW_ATTEMPTS_REMAINING (x)
	'6700'	SW_WRONG_LENGTH
	'6981'	SW_WRONG_FILE_TYPE
	'6982'	SW_SECURITY_STATUS_NOT_SATISFIED
	'6983'	SW_FILE_INVALID
	'6984'	SW_DATA_INVALID
	'6986'	SW_COMMAND_NOT_ALLOWED
	'6A82'	SW_FILE_NOT_FOUND
	'6A86'	SW_INCORRECT_P1P2
	'6A88'	SW_DATA_OBJECT_NOT_FOUND
	'9000'	Succès, pas d'erreur

Tableau 13 : VERIFY format de la réponse



La commande VERIFY avec **Lc='00'**, donne le status du PIN. '63CX' si il reste X essais (PTX = X). Si le PIN est bloqué, la carte retourne '63C0' ou '6983'.

## 6.5.2 RESET RETRY COUNTER

Cette commande est utilisée pour débloquer ou changer le code porteur.

Paramètre	Valeur	Remarque
CLA	'00'	
INS	'2C'	RESET RETRY COUNTER
P1		'02' : Débloquer et changer la valeur du PIN '03' : Débloquer le PIN
P2	'80'   ID	ID = '01' pour PIN et '02' pour PUK
Lc		'00' si P1 = '03' (déblocage uniquement) Longueur nouveau PIN/PUK si P1 = '02'
Data		
Le		Absent

Tableau 14 : RESET RETRY COUNTER format de la commande

Paramètre	Valeur	Remarque
Data		Absent
SW	'6700'	SW_WRONG_LENGTH
	'6981'	SW_WRONG_FILE_TYPE
	'6982'	SW_SECURITY_STATUS_NOT_SATISFIED
	'6983'	SW_FILE_INVALID
	'6985'	SW_CONDITIONS_NOT_SATISFIED
	'6986'	SW_COMMAND_NOT_ALLOWED
	'6A82'	SW_FILE_NOT_FOUND
	'6A86'	SW_INCORRECT_P1P2
	'9000'	Succès, pas d'erreur

Tableau 15 : RESET RETRY COUNTER format de la réponse



L'opération (CHANGE ou UNBLOCK) ne sera autorisée que si les conditions d'accès sont satisfaites : exemple pour débloquer le PIN il faut au préalable avoir présenté le PUK, pour changer le PIN il faut au préalable avoir présenté le PIN ou le PUK.



L'opération (CHANGE) ne sera autorisée que si le PIN/PUK n'est **pas bloqué**. Il n'est pas possible de changer le PUK (politique de sécurité ANS).

## 6.6 Commandes de gestion de l'environnement de sécurité

### 6.6.1 MSE SET

Cette commande permet de configurer, avant de réaliser des opérations cryptographiques, un environnement de sécurité. Elle va sélectionner une clé et/ou algorithme afin de pouvoir réaliser des opérations de déchiffrement ou de signature.

Paramètre	Valeur	Remarque
CLA	'00'	
INS	'22'	Commande MSE SET
P1 P2		CRT (Control Reference Template) sélectionné, voir ci-dessous : <ul style="list-style-type: none"> <li>• '41B8' déchiffrement</li> <li>• '81B6' signature</li> </ul>
Lc		Longueur des données en entrée
Data		Information sur algorithme/clé à utiliser
Le		Absent

Tableau 16 : MSE SET format de la commande

Paramètre	Valeur	Remarque
Data		Absent
SW	'6283'	SW_SELECTED_FILE_DEACTIVATED
	'63C0'	SW_ATTEMPTS_REMAINING_00
	'63Cx'	SW_ATTEMPTS_REMAINING (x)
	'6700'	SW_WRONG_LENGTH
	'6981'	SW_WRONG_FILE_TYPE
	'6982'	SW_SECURITY_STATUS_NOT_SATISFIED
	'6985'	SW_CONDITIONS_NOT_SATISFIED
	'6986'	SW_COMMAND_NOT_ALLOWED
	'6A80'	SW_WRONG_DATA
	'6A81'	SW_FUNC_NOT_SUPPORTED
	'6A82'	SW_FILE_NOT_FOUND
	'6A86'	SW_INCORRECT_P1P2
	'6A88'	SW_DATA_OBJECT_NOT_FOUND
	'9000'	Succès, pas d'erreur

Tableau 17 : MSE SET format de la réponse

Tag	Lg	Signification	Présence
'83'	'01'	Référence clé à utiliser	✓

**Tableau 18 : Champ DATA pour CRT DECHIFFREMENT – P1P2='41B8'**

Tag	Lg	Signification	Présence
'91'	'02'	Premier octet = Algorithme à utiliser Second octet = Référence clé à utiliser	✓

**Tableau 19 : Champ DATA pour CRT SIGNATURE DST – P1P2='81B6'**

Le second octet, est lui-même constitué de deux champs :

- Bit 8            bit qui indique si oui ou non le condensé est réalisé en dehors de la carte (1 : en dehors)
- Bits 7-1        référence de la clé à utiliser

ID	Algorithme	
'0A'	Signature SHA1 padding PKCS#1 v1.5	CKM_SHA1_RSA_PKCS
'28'	Signature SHA256 padding PKCS#1 v1.5	CKM_SHA256_RSA_PKCS
'15'	Signature SHA1 padding RSA PSS	CKM_SHA1_RSA_PKCS_PSS
'2A'	Signature SHA256 padding RSA PSS	CKM_SHA256_RSA_PKCS_PSS
	<b>Non géré, contournement : utiliser le déchiffrement</b>	CKM_RSA_PKCS

**Tableau 20 : Liste d'algorithmes**

ID	
'10'	Clé RSA de signature
'20'	Clé RSA d'authentification ou de Déchiffrement

**Tableau 21 : Référence des clés**

## 6.7 Commandes cryptographiques

### 6.7.1 PSO – COMPUTE DIGITAL SIGNATURE

Cette commande permet de mettre en œuvre une signature avec la clé et l'algorithme préalablement sectionné avec la commande MSE SET.

Pour ce qui est du condensé, deux cas sont à considérer :

- Bit 8 de l'octet « référence clé » est à 1. Dans ce cas, le **condensé** est réalisé **en dehors de la carte**. Par conséquent le champ **Data** correspond au condensé des données à signer.
- Bit 8 de l'octet « référence clé » est à 0. Dans ce cas, le **condensé** est réalisé **par la carte**. Par conséquent le champ **Data** correspond aux données à signer.

Paramètre	Valeur	Remarque
CLA	'00'	
INS	'2A'	Commande PSO Compute Digital Signature
P1 P2	'9E9A'	
Lc		Longueur des données à signer ou condensé
Data		Données à signer ou condensé
Le		Longueur du champ <b>Data</b> attendu

Tableau 22 : PSO CDS, format de la commande

Paramètre	Valeur	Remarque
Data		Absent
SW	'6700'	SW_WRONG_LENGTH
	'6981'	SW_WRONG_FILE_TYPE
	'6982'	SW_SECURITY_STATUS_NOT_SATISFIED
	'6985'	SW_CONDITIONS_NOT_SATISFIED
	'6986'	SW_COMMAND_NOT_ALLOWED
	'6A82'	SW_FILE_NOT_FOUND
	'6A86'	SW_INCORRECT_P1P2
	'9000'	Succès, pas d'erreur

Tableau 23 : PSO CDS, format de la réponse

### 6.7.2 PSO – DECIPHER

Cette commande permet de déchiffrer des données avec une clé privée asymétrique.

Les données retournées sont les données déchiffrées. La longueur des données en entrée et la réponse correspondent à la longueur de la clé (256 octets pour une clé RSA 2048 bits).

Paramètre	Valeur	Remarque
CLA	'00'	
INS	'2A'	Commande PSO Decipher
P1 P2	'8086'	
Lc		Longueur des données à déchiffrer
Data		Données à déchiffrer remplie selon PKCS#1 v1.5
Le		Longueur du champ Data attendu

Tableau 24 : PSO DECIPHER, format de la commande

Paramètre	Valeur	Remarque
Data		Absent ou Données déchiffrées (si SW = '9000')
SW	'6700'	SW_WRONG_LENGTH
	'6981'	SW_WRONG_FILE_TYPE
	'6982'	SW_SECURITY_STATUS_NOT_SATISFIED
	'6985'	SW_CONDITIONS_NOT_SATISFIED
	'6986'	SW_COMMAND_NOT_ALLOWED
	'6A80'	SW_WRONG_DATA
	'6A82'	SW_FILE_NOT_FOUND
	'6A86'	SW_INCORRECT_P1P2
	'9000'	Succès, pas d'erreur

Tableau 25 : PSO DECIPHER, format de la réponse

## 7 Exemples d'usage

Tous les exemples ont été réalisés après sélection de l'application CHIPDOC sur le canal 1, l'octet CLA est donc égal à XXXXXX01, le numéro de canal est codé sur les 2 bits de poids faible de l'octet CLA.

### 7.1 Gestion des fichiers

#### 7.1.1 Sélection/lecture de Fichiers sous le MF

```
>>> %Run cpsv4-read-mf-711.py
```

```
Readers: {"response": 0, "data": ["Alcorlink USB Smart Card Reader 0"]}
```

Use of T=0 protocol

```
[Event-INFO]Script: cpsv4-read-mf-711.py On reader: Alcorlink USB Smart Card Reader 0
```

```
[Event-INFO]Power-on
```

```
[Event-REC]State:The card has been reset and specific communication protocols have been established.
```

```
[Event-REC]Protocol:Use of T=0 protocol
```

```
[Event-REC]ATR:3BDC18FF00001225006480000401009000
```

```
==> SELECT CHIPDOC on canal 1
```

```
APDU: 01A4040C088025000001FF0100 (13 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
[Event-INFO]EF-DIR
```

```
APDU: 01A4080C023F00 (7 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
APDU: 01A40000022F0000 (8 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (21 bytes)
```

```
6F13800300001E820301000083022F00860300FFFF
```

```
APDU: 01B000001E (5 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (30 bytes)
```

```
61154F0DE828BD080F8025000001FF001051043F0000010000000000000000
```

```
[Event-INFO]EF-ATR
```

```
APDU: 01A4080C023F00 (7 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
APDU: 01A40000022F0100 (8 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (21 bytes)
```

```
6F138003000012820301000083022F01860300FFFF
```

APDU: 01B0000012 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (18 bytes)

4301B847033821924F088025000001FF0002

[Event-INFO]EF-SN

APDU: 01A4080C023F00 (7 bytes)

SW12: 9000 (Command successfully executed (OK))

APDU: 01A4000002D00300 (8 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (21 bytes)

6F13800300000C82030100008302D003860300FFFF

APDU: 01B000000C (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (12 bytes)

5A0A8025000001030953274F

## [Event-INFO]EF-TECH

APDU: 01A4080C023F00 (7 bytes)

SW12: 9000 (Command successfully executed (OK))

APDU: 01A40000022FFF00 (8 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (21 bytes)

6F138003000064820301000083022FFF860300FFFF

APDU: 01B0000064 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (100 bytes)

[Event-ENDS]No Error.

>>>

### 7.1.2 Sélection/lecture de Fichiers sous ADF CPS '0001'

```
>>> %Run cpsv4-read-cps-712.py
```

```
Readers: {"response": 0, "data": ["Alcorlink USB Smart Card Reader 0"]}
```

Use of T=0 protocol

```
[Event-INFO]Script: cpsv4-read-cps-712.py On reader: Alcorlink USB Smart Card Reader 0
```

```
[Event-INFO]Power-on
```

```
[Event-REC]State:The card has been reset and specific communication protocols have been established.
```

```
[Event-REC]Protocol:Use of T=0 protocol
```

```
[Event-REC]ATR:3BDC18FF00001225006480000401009000
```

```
==> SELECT CHIPDOC on canal 1
```

```
APDU: 01A4040C088025000001FF0100 (13 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
[Event-INFO]EF-IDCARD
```

```
APDU: 01A4080C043F000001 (9 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
APDU: 01A4000002D10100 (8 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (21 bytes)
```

```
6F13800300001F82030100008302D101860300FFFF
```

```
APDU: 01B000001F (5 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (31 bytes)
```

```
E31D8005802500000181053100603745820180830420240530840420270530
```

```
[Event-INFO]EF-PSNAME
```

```
APDU: 01A4080C043F000001 (9 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
APDU: 01A4000002D10200 (8 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (21 bytes)
```

```
6F13800300002782030100008302D10286030031FF
```

```
APDU: 01B0000027 (5 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (39 bytes)
```

```
E42580011F811452454D504C504841524D4554553030363232313383044855474F84044855474F
```

```
[Event-INFO]EF-PSLANG
```

```
APDU: 01A4080C043F000001 (9 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

APDU: 01A4000002D10300 (8 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (21 bytes)

6F13800300000C82030100008302D10386030031FF

APDU: 01B000000C (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (12 bytes)

E50A80086672202020202020

[Event-INFO]EF-PSINFO

APDU: 01A4080C043F000001 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

APDU: 01A4000002D10400 (8 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (21 bytes)

6F13800300002C82030100008302D104860300FFFF

APDU: 01B0000002C (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (44 bytes)

ED2A800101810C383939373030363232313339830115841452454D504C504841524D455455303036323231

33

## Authenticate PORTEUR

---

==> SELECT VERIFY

APDU: 012000010431323334 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

[Event-INFO]EF-ACTIVITY01

APDU: 01A4080C043F000001 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

APDU: 01A4000002D12000 (8 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (21 bytes)

6F13800300000B82030100008302D12086030131FF

APDU: 01B000000B (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (11 bytes)

EE09800104810101830101

[Event-ENDS]No Error.

>>>

### 7.1.3 Sélection/lecture de Fichiers sous DF CPS2TER '0003'

>>> %Run cpsv4-read-2ter-713.py

Readers: {"response": 0, "data": ["Alcorlink USB Smart Card Reader 0"]}

Use of T=0 protocol

[Event-INFO]Script: cpsv4-read-cps2ter-713.py On reader: Alcorlink USB Smart Card Reader 0

[Event-INFO]Power-on

[Event-REC]State:The card has been reset and specific communication protocols have been established.

[Event-REC]Protocol:Use of T=0 protocol

[Event-REC]ATR:3BDC18FF00001225006480000401009000

==> SELECT CHIPDOC on canal 1

APDU: 01A4040C088025000001FF0100 (13 bytes)

SW12: 9000 (Command successfully executed (OK))

[Event-INFO]EF2TER-DIR

APDU: 01A4080C043F000003 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

APDU: 01A40000022F0000 (8 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (21 bytes)

6F13800300002B820301000083022F00860300FFFF

APDU: 01B000002B (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (43 bytes)

790A4F088025000001FF0002611D4F08A000000022FF1000500B4153535F4D414C5F4F424C51043F007F01

[Event-INFO]EF2TER-ATR

APDU: 01A4080C043F000003 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

APDU: 01A40000022F0100 (8 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (21 bytes)

6F138003000002820301000083022F01860300FFFF

APDU: 01B0000002 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (2 bytes)

8000

[Event-INFO]EF2TER-ICC

APDU: 01A4080C043F000003 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

APDU: 01A4000002000200 (8 bytes)  
SW12: 9000 (Command successfully executed (OK))  
Data from card: (21 bytes)  
6F138003000011820301000083020002860300FFFF

APDU: 01B0000011 (5 bytes)  
SW12: 9000 (Command successfully executed (OK))  
Data from card: (17 bytes)  
E10F800500205384218103006C00820100

[Event-INFO]EF2TER-IDCARD  
APDU: 01A4080C043F000003 (9 bytes)  
SW12: 9000 (Command successfully executed (OK))

APDU: 01A4000002000300 (8 bytes)  
SW12: 9000 (Command successfully executed (OK))  
Data from card: (21 bytes)  
6F1C380030000020820301000083020003860300FFFF

APDU: 01B0000020 (5 bytes)  
SW12: 9000 (Command successfully executed (OK))  
Data from card: (32 bytes)  
E31E800680250000017F81053100603745820180830420240530840420270530

[Event-INFO]EF2TER-NAME  
APDU: 01A4080C043F000003 (9 bytes)  
SW12: 9000 (Command successfully executed (OK))

APDU: 01A4000002000400 (8 bytes)  
SW12: 9000 (Command successfully executed (OK))  
Data from card: (21 bytes)  
6F1C3800300004582030100008302000486030031FF

[Event-INFO]EF2TER-IC  
APDU: 01A4080C043F000003 (9 bytes)  
SW12: 9000 (Command successfully executed (OK))

APDU: 01A4000002000500 (8 bytes)  
SW12: 9000 (Command successfully executed (OK))  
Data from card: (21 bytes)  
6F138003000012820301000083020005860300FFFF



## [Event-INFO]EF2TER-AMO

APDU: 01A4080C063F0000037F01 (11 bytes)

SW12: 9000 (Command successfully executed (OK))

APDU: 01A4000002400000 (8 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (21 bytes)

6F13800300019482030100008302400086030131FF

APDU: 01B00000E7 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (231 bytes)

APDU: 01B000E7AD (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (173 bytes)

## [Event-INFO]EF2TER-SITUATION01

APDU: 01A4080C043F000003 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

APDU: 01A4000002401000 (8 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (21 bytes)

6F13800300003982030100008302401086030131FF

APDU: 01B0000039 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (57 bytes)

[Event-ENDS]No Error.

>>>

### 7.1.4 Sélection/lecture de Certificat/Clé publique sous DF AUTH '0001/0102'

```
>>> %Run cpsv4-read-auth-714.py
```

```
Readers: {"response": 0, "data": ["Alcorlink USB Smart Card Reader 0"]}
```

Use of T=0 protocol

```
[Event-INFO]Script: cpsv4-read-auth-714.py On reader: Alcorlink USB Smart Card Reader 0
```

```
[Event-INFO]Power-on
```

```
[Event-REC]State:The card has been reset and specific communication protocols have been established.
```

```
[Event-REC]Protocol:Use of T=0 protocol
```

```
[Event-REC]ATR:3BDC18FF00001225006480000401009000
```

```
==> SELECT CHIPDOC on canal 1
```

```
APDU: 01A4040C088025000001FF0100 (13 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
[Event-INFO]EF-AUTHPUB
```

```
APDU: 01A4080C063F0000010102 (11 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
APDU: 01A4000002902000 (8 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (21 bytes)
```

```
6F13800300010D8203010000830290208603000303
```

```
APDU: 01B00000E7 (5 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (231 bytes)
```

```
30820109028202100C928CD0348E077AB78E7D6A91F474613C4755AE30FBD7CF3D25745BA2DB0A8AED30FC  
2C9AF185948486EBC0B7F8F16DBA2D1C0705AF4CE9175EB2DD52B9798E74DAEAFEE4EA4976BA6D9F98932  
24D62B6A952DC5800329200CDFA4870113484DE12DF3ABD37F5B77CB02218AD4CE173994D00BDAAB25B1  
114186FD5FFDB82D6920B3A187A71705BADF805B53A78D0CADD24B0D651DAF057C59C901816DBA7E4696  
A54CB0620B714BC312A7C204E3555C7A1F81815FBE43D562ACC74D98DAA88671BBA92E19B524DD75C225  
4558577248330D4FD82CF23CFAD7C15851D83A28
```

```
APDU: 01B000E726 (5 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (38 bytes)
```

```
6E539F50C5BDC19DC094FE2A5A34E6E42B22F6EE8C8E92D5C4ED9970D054F524C10203010001
```

```
[Event-INFO]EF-AUTHCERT
```

```
APDU: 01A4080C063F0000010102 (11 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
APDU: 01A4000002A02000 (8 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
Data from card: (21 bytes)
```

```
6F1380030008D182030100008302A0208603000303
```

APDU: 01B00000E7 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (231 bytes)

308208CD308206B5A00302010202106D3C399C3485A9B1EB5A8A4E43E9E7BC300D06092A864886F70D010  
10B05003081A4310B300906035504061302465231133011060355040A0C0A415349502D53414E544531173  
015060355040B0C0E303030322031383735313237353131173015060355040B0C0E4947432D53414E544520  
5445535431233021060355040B0C1A504F555220494E544547524154494F4E205345554C454D454E5431293  
02706035504030C2054455354204143204947432D53414E544520464F525420504552534F4E4E4553301E17  
0D3234303533303039303934335A17

APDU: 01B000E7E7 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (231 bytes)

0D3237303533303039303934335A3074310B30090603550406130246523120301E060355040C0C17506861  
726D616369656E20656E20666F726D6174696F6E310D300B060355042A0C044855474F311D301B06035504  
040C1452454D504C504841524D455455303036323231333115301306035504030C0C383939373030363232  
31333930820122300D06092A864886F70D01010105000382010F003082010A0282010100C928CD0348E077  
AB78E7D6A91F474613C4755AE30FBD7CF3D25745BA2DB0A8AED30FC2C9AF185948486EBC0B7F8F16DBA2  
D1C0705AF4CE9175EB2DD52B9798E74DAE

APDU: 01B001CEE7 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (231 bytes)

AFEE4EA4976BA6D9F9893224D62B6A952DC5800329200CDFA4870113484DE12DF3ABD37F5B77CB02218A  
D4CE173994D00BDAAB25B1114186FD5FFDB82D6920B3A187A71705BADF805B53A78D0CADD24B0D651DA  
F057C59C901816DBA7E4696A54CB0620B714BC312A7C204E3555C7A1F81815FBE43D562ACC74D98DAA886  
71BBA92E19B524DD75C2254558577248330D4FD82CF23CFAD7C15851D83A286E539F50C5BDC19DC094FE2  
A5A34E6E42B22F6EE8C8E92D5C4ED9970D054F524C10203010001A382042830820424301F0603551D25041  
8301606082B06010505070302060A2B06010401

APDU: 01B002B5E7 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (231 bytes)

8237140202301F0603551D23041830168014DDA1F3CB438E109A30ECC43837F2C797F4C40390300F06082A  
817A0147010208040302011530818006082B0601050507010104743072302606082B06010505073001861A  
687474703A2F2F6F6373702E6573616E74652E676F75762E6672304806082B06010505073002863C6874747  
03A2F2F6967632D73616E74652E6573616E74652E676F75762E66722F4143253230544553542F4143492D46  
4F2D50502D544553542E636572300F06082A817A0147010202040302010130530603551D20044C304A3048  
060D2A817A01815501070201010101

APDU: 01B0039CE7 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (231 bytes)

3037303506082B060105050702011629687474703A2F2F6967632D73616E74652E6573616E74652E676F757  
62E66722F504325323054455354308201400603551D1F04820137308201333081F2A081EFA081EC8681E96C  
6461703A2F2F616E6E75616972652D6967632E6573616E74652E676F75762E66722F636E3D5445535425323  
041432532304947432D53414E5445253230464F5254253230504552534F4E4E45532C6F753D544553542532  
304143253230524143494E452532304947432D53414E5445253230464F52542C6F753D4947432D53414E544  
5253230544553542C6F753D3030

APDU: 01B00483E7 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (231 bytes)

30322532303138373531323735312C6F3D415349502D53414E54452C633D46523F63657274696669636174  
657265766F636174696F6E6C6973743B62696E6172793F626173653F6F626A656374436C6173733D706B694  
341303CA03AA0388636687474703A2F2F6967632D73616E74652E6573616E74652E676F75762E66722F4352  
4C2F4143492D464F2D50502D544553542E63726C301D0603551D0E04160414A3267CB2C11FEA1438B31444  
2B938E5EE83875B73081FA0603551D2E0481F23081EF3081ECA081E9A081E68681E36C6461703A2F2F616E6  
E75616972652D6967632E6573616E

APDU: 01B0056AE7 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (231 bytes)

74652E676F75762E66722F636E3D5445535425323041432532304947432D53414E5445253230464F5254253  
230504552534F4E4E45532C6F753D544553542532304143253230524143494E452532304947432D53414E54  
45253230464F52542C6F753D4947432D53414E5445253230544553542C6F753D3030303225323031383735  
31323735312C6F3D415349502D53414E54452C633D46523F64656C74617265766F636174696F6E6C6973743  
B62696E6172793F626173653F6F626A656374436C6173733D706B694341300E0603551D0F0101FF04040302  
0780300F06082A817A0147010205

APDU: 01B00651E7 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (231 bytes)

040304018030350603551D11042E302CA02A060A2B060104018237140203A01C0C1A382E39393730303632  
323133394063617274652D6370732E667230090603551D1304023000302306082A817A0147010203041713  
153830323530303030312F33313030363033373435300D06092A864886F70D01010B0500038202010048  
66301717F320EB3B0EA9A1B03ECAB030BBFD52321AE16D64596A7CAD95312663BBE7F3F9FF4CED7B7F5C2  
D86FE945C81DE4A5951F8B8775D0ADFBEBE9E91489BBE5EA871AF0940D46D84D02ECDBDFC97C6CE951F1  
82878E6650D0B527BBA3AD0BB4A6F6B11F

APDU: 01B00738E7 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (231 bytes)

41BB170D38ACF5B9576AF6CADB17F97826CBDCE058C9D36F6E444FC98DBD18EF975B9C3BCC70D307404B2  
37A04DE40F6F6F989013EBE4BF1A206878DE07023576882046F797FCA7B46C1F8FA94FAD41B101007F59E6  
ED41DA43EAEE7A3BB122BFA30C2ED121FC149ECAC307C34F926201091B7C196EA816BA1E1F4796A5C4049  
F4BD9A309A3D72DDFF91D95AB7A8858971C1AF1B637F24417FC40C31154CF1D2F298B9C5BFC8A59EE96D1  
2A29128F3A53AA1917DFA16844D0EA037EA071211674E4CF68ACB665B510DA3E6F09A48B7E8F38EC53AE2  
2121055E978C07621E9026F8D23E76A7577F

APDU: 01B0081FB2 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (178 bytes)

9CBC85F5D15CB1623C8ECF73E9AF8ADF9FCF1AEE0326EA0D6F1A39849598A93702DAB73DB25E055119C62  
DC87F88FE187BC71F423D56C5B2FDBC45D1602D29ABB6C238CB3599126696BF99C22E8D252E410E1F0FD8  
3E06E7F70D3108AAC056B10CDC2B096D492D6B601BA2B81D4DA5714FC9809D4A4C5385A483A782907AF7  
352826F4488F1528ED5755FAE51F5B1258BF03CF6A1C528E74A5B4C0A66E9462DE3BDD0740C3749FF7B592  
6D4F486185243064

[Event-ENDS]No Error.

>>>

## 7.2 Gestion du code porteur

### 7.2.1 Blocage puis déblocage du PIN

```
>>> %Run cpsv4-pin-unblock-721.py
```

```
Readers: {"response": 0, "data": ["Alcorlink USB Smart Card Reader 0"]}
```

Use of T=0 protocol

```
[Event-INFO]Script: cpsv4-read-auth-714.py On reader: Alcorlink USB Smart Card Reader 0
```

```
[Event-INFO]Power-on
```

```
[Event-REC]State:The card has been reset and specific communication protocols have been established.
```

```
[Event-REC]Protocol:Use of T=0 protocol
```

```
[Event-REC]ATR:3BDC18FF00001225006480000401009000
```

```
==> SELECT CHIPDOC on canal 1
```

```
APDU: 01A4040C088025000001FF0100 (13 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

Authenticate PORTEUR / GET PTC

```
=====
```

```
==> GET PTC
```

```
APDU: 0120000100 (5 bytes)
```

```
SW12: 63C3 (Verify fail, 3 tries left)
```

Authenticate PORTEUR / BAD PIN

```
=====
```

```
==> VERIFY BAD PIN
```

```
APDU: 012000010431313131 (9 bytes)
```

```
SW12: 63C2 (Verify fail, 2 tries left)
```

Authenticate PORTEUR / GET PTC

```
=====
```

```
==> GET PTC
```

```
APDU: 0120000100 (5 bytes)
```

```
SW12: 63C2 (Verify fail, 2 tries left)
```

Authenticate PORTEUR / BAD PIN

```
=====
```

```
==> VERIFY BAD PIN
```

```
APDU: 012000010431313131 (9 bytes)
```

```
SW12: 63C1 (Verify fail, 1 try left)
```

Authenticate PORTEUR / GET PTC

```
=====
```

```
==> GET PTC
```

```
APDU: 0120000100 (5 bytes)
```

```
SW12: 63C1 (Verify fail, 1 try left)
```

Authenticate PORTEUR / BAD PIN

=====  
==> VERIFY BAD PIN

APDU: 012000010431313131 (9 bytes)

SW12: 6300 (No information given (NV-Ram changed))

Authenticate PORTEUR / GET PTC

=====  
==> GET PTC

APDU: 0120000100 (5 bytes)

SW12: 6983 (Authentication method blocked)

Authenticate PUK

=====  
==> VERIFY PUK

APDU: 01200002083132333435363738 (13 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate CHANGE PIN not possible if PTC=0

=====  
==> PIN CHANGE

APDU: 012C02810434333231 (9 bytes)

SW12: 6983 (Authentication method blocked)

Authenticate PIN UNBLOCK

=====  
==> PIN UNBLOCK

APDU: 012C038100 (5 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate PORTEUR / GET PTC

=====  
==> GET PTC

APDU: 0120000100 (5 bytes)

SW12: 63C3 (Verify fail, 3 tries left)

Authenticate PORTEUR

=====  
==> VERIFY

APDU: 012000010431323334 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

[Event-ENDS]No Error.

>>>

## 7.2.2 Changement de la valeur du PIN avec PIN

>>> %Run cpsv4-pin-change-722.py

Readers: {"response": 0, "data": ["Alcorlink USB Smart Card Reader 0"]}

Use of T=0 protocol

[Event-INFO]Script: cpsv4-pin-change-722.py On reader: Alcorlink USB Smart Card Reader 0

[Event-INFO]Power-on

[Event-REC]State:The card has been reset and specific communication protocols have been established.

[Event-REC]Protocol:Use of T=0 protocol

[Event-REC]ATR:3BDC18FF00001225006480000401009000

==> SELECT CHIPDOC on canal 1

APDU: 01A4040C088025000001FF0100 (13 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate CHANGE PIN not possible if no PIN/PUK presented

=====

==> PIN CHANGE

APDU: 012C02810431313131 (9 bytes)

SW12: 6982 (Security condition not satisfied)

Authenticate PORTEUR

=====

==> VERIFY

APDU: 012000010431323334 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate CHANGE PIN possible with PIN presented

=====

==> PIN CHANGE

APDU: 012C02810431313131 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate PORTEUR

=====

==> VERIFY BAD PIN

APDU: 012000010431323334 (9 bytes)

SW12: 63C2 (Verify fail, 2 tries left)

Authenticate PORTEUR

=====

==> VERIFY PIN

APDU: 012000010431313131 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate CHANGE PIN possible with PIN presented

```
=====
```

==> PIN CHANGE

APDU: 012C02810431323334 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate PORTEUR / GET PTC

```
=====
```

==> GET PTC

APDU: 0120000100 (5 bytes)

SW12: 63C3 (Verify fail, 3 tries left)

Authenticate PORTEUR: verify PIN value restored

```
=====
```

==> VERIFY

APDU: 012000010431323334 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

[Event-ENDS]No Error.

>>>

### 7.2.3 Changement de la valeur du PIN avec PUK

>>> %Run cpsv4-pin-change-723.py

Readers: {"response": 0, "data": ["Alcorlink USB Smart Card Reader 0"]}

Use of T=0 protocol

[Event-INFO]Script: cpsv4-pin-change-723.py On reader: Alcorlink USB Smart Card Reader 0

[Event-INFO]Power-on

[Event-REC]State:The card has been reset and specific communication protocols have been established.

[Event-REC]Protocol:Use of T=0 protocol

[Event-REC]ATR:3BDC18FF00001225006480000401009000

==> SELECT CHIPDOC on canal 1

APDU: 01A4040C088025000001FF0100 (13 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate CHANGE PIN not possible if no PIN/PUK presented

=====

==> PIN CHANGE

APDU: 012C02810431313131 (9 bytes)

SW12: 6982 (Security condition not satisfied)

Authenticate PUK

=====

==> VERIFY PUK

APDU: 01200002083132333435363738 (13 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate CHANGE PIN possible with PUK presented

=====

==> PIN CHANGE

APDU: 012C02810431313131 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate PORTEUR

=====

==> VERIFY BAD PIN

APDU: 012000010431323334 (9 bytes)

SW12: 63C2 (Verify fail, 2 tries left)

Authenticate PORTEUR

=====

==> VERIFY PIN

APDU: 012000010431313131 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate CHANGE PIN possible with PIN presented

```
=====
```

==> PIN CHANGE

APDU: 012C02810431323334 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

Authenticate PORTEUR / GET PTC

```
=====
```

==> GET PTC

APDU: 0120000100 (5 bytes)

SW12: 63C3 (Verify fail, 3 tries left)

Authenticate PORTEUR: verify PIN value restored

```
=====
```

==> VERIFY

APDU: 012000010431323334 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

[Event-ENDS]No Error.

>>>

## 7.3 Opérations cryptographiques

### 7.3.1 Signature d'un condensé/data

L'exemple ci-dessous donne l'exemple de la signature d'un condensé (réalisé en dehors de la carte) ou de données avec condensé réalisé par la carte. On utilise les mêmes données, le résultat (signature) est donc identique dans les 2 commandes PSO CDS.

```
>>> %Run cpsv4-sign-sha-731.py
```

```
Readers: {"response": 0, "data": ["Alcorlink USB Smart Card Reader 0"]}
```

Use of T=0 protocol

```
[Event-INFO]Script: cpsv4-sign-sha-731.py On reader: Alcorlink USB Smart Card Reader 0
```

```
[Event-INFO]Power-on
```

```
[Event-REC]State:The card has been reset and specific communication protocols have been established.
```

```
[Event-REC]Protocol:Use of T=0 protocol
```

```
[Event-REC]ATR:3BDC18FF00001225006480000401009000
```

```
==> SELECT CHIPDOC on canal 1
```

```
APDU: 01A4040C088025000001FF0100 (13 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
==> SELECT DF Signature
```

```
APDU: 01A408000400010101 (9 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

Authenticate PORTEUR

```
=====
```

```
==> VERIFY
```

```
APDU: 012000010431323334 (9 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

Data	112233445566778899AABBCCDDEEFF4142434445464748495051525354555657
------	--

5859606162636465666768697071727374757677
--

Sha256	98dd57312c8cfe1a043a5c1cba0b0d7dd5d72e43b3af610013704a5bbe3c8707
--------	--

MSE SET CRT DST: CKM\_SHA256\_RSA\_PKCS / condensé en dehors de la carte

```
=====
```

```
==> MSE SET DST
```

```
APDU: 012281B60491022890 (9 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

PSO SIGN

=====

==> PSO SIGN

APDU: 012A9E9A2098dd57312c8fce1a043a5c1cba0b0d7dd5d72e43b3af610013704a5bbe3c8707 (37 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (256 bytes)

5F0921D806CA47C15490F28A692EEBDCC7CE0427C289035A483332AA9B3B945A3DC7AAF591049646AFA7D  
8E8E8007302D596513FDB66F5C940E01A8DE72374993BDF3702BD86F537AD0640539247273491D8FF073F0  
9399B59F1D39E18D2278FA30D314E9088AE6B4C79F6182B580E6D53197B7D214A544526AE07B48623EDB0  
7259E6A2E67659D769A930A41FD7123FB1D03D4C8AC3B984F66913FF45C6BFB3A7790E0A13B8EF86CF5E65  
AD396221379BD97AD0EB2BA557D991153F0264B7942483404516E90D0695A0B801283E5A5CBB6FF307106  
C2C96AC17BFC2C0C21E227CB129BDD03C1E7CF6D89D28FC78FA87A2A06E4EEC41C68D679B188B3D3BF869

MSE SET CRT DST: CKM\_SHA256\_RSA\_PKCS / condensé réalisé par la carte

=====

==> MSE SET DST

APDU: 012281B60491022810 (9 bytes)

SW12: 9000 (Command successfully executed (OK))

PSO SIGN

=====

==> PSO SIGN

APDU:

012A9E9A34112233445566778899AABBCCDDEFF4142434445464748495051525354555657585960616263  
6465666768697071727374757677 (57 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (256 bytes)

5F0921D806CA47C15490F28A692EEBDCC7CE0427C289035A483332AA9B3B945A3DC7AAF591049646AFA7D  
8E8E8007302D596513FDB66F5C940E01A8DE72374993BDF3702BD86F537AD0640539247273491D8FF073F0  
9399B59F1D39E18D2278FA30D314E9088AE6B4C79F6182B580E6D53197B7D214A544526AE07B48623EDB0  
7259E6A2E67659D769A930A41FD7123FB1D03D4C8AC3B984F66913FF45C6BFB3A7790E0A13B8EF86CF5E65  
AD396221379BD97AD0EB2BA557D991153F0264B7942483404516E90D0695A0B801283E5A5CBB6FF307106  
C2C96AC17BFC2C0C21E227CB129BDD03C1E7CF6D89D28FC78FA87A2A06E4EEC41C68D679B188B3D3BF869

[Event-ENDS]No Error.

>>>

### 7.3.2 Authentification SSL / Déchiffrement

Le masque CPS V4, contrairement au masque CPS V3, ne possède pas de signature sans condensé (équivalent à la commande INTERNAL AUTHENTICATE). Il existe un contournement qui consiste, si l'on veut signer N octets avec padding PKCS#1 v1.5 sans réaliser de condensé, à réaliser le padding puis à utiliser la commande de déchiffrement.

Cet exemple donne 2 exemples de DECIPHER, le premier est à utiliser en lieu et place d'un INTERNAL AUTH (utilisé dans le cadre des authentifications SSL), le second est un DECIPHER conforme PKCS#v1.5 (compliant avec l'API DECIPHER de la CRYPTOLIB ANS).

```
>>> %Run cpsv4-decipher-732.py
```

```
Readers: {"response": 0, "data": ["Alcorlink USB Smart Card Reader 0"]}
```

Use of T=0 protocol

```
[Event-INFO]Script: cpsv4-decipher-732.py On reader: Alcorlink USB Smart Card Reader 0
```

```
[Event-INFO]Power-on
```

```
[Event-REC]State:The card has been reset and specific communication protocols have been established.
```

```
[Event-REC]Protocol:Use of T=0 protocol
```

```
[Event-REC]ATR:3BDC18FF00001225006480000401009000
```

```
==> SELECT CHIPDOC on canal 1
```

```
APDU: 01A4040C088025000001FF0100 (13 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

```
==> SELECT DF Authentification
```

```
APDU: 01A408000400010102 (9 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

Authenticate PORTEUR

```
=====
```

```
==> VERIFY
```

```
APDU: 012000010431323334 (9 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

#### 1) DECIPHER for INTERNAL AUTH (SSL)

```
=====
```

Add padding for workaround to retrieve signature as INTERNAL AUTH of data

==> Padding has to be compliant with SIGN/PKCS#v1.5: 00 01 FF .... FF 00 DATA

MSE SET DECIPHER clé AUTH

```
=====
```

```
==> MSE SET DECIPHER
```

```
APDU: 012241B803830120 (8 bytes)
```

```
SW12: 9000 (Command successfully executed (OK))
```

## PSO DECIPHER (Envoyé avec le chainage des commandes en T=0)

==> PSO DECIPHER

## APDU:

112A808600001FFFFFFFFFFFFFFFFFFF (133 bytes)

SW12: 9000 (Command successfully executed (OK))

==> PSO DECIPHER

## APDU:

012A808680FFFFFFFFFFFFFFF00FFFFDCCBAA9988776655443322110001122334455667741 (133 bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (256 bytes)

29831A7868D9D25E62A1BD0BCC287E7CE7198654662074917150A7720267802C732CC45B45E9CFFBEB13  
63FAA59081D9F99A9D5E016415F8F5F67C5F60BB9FDB87FB5DC6902203EEC1AD6F603A412877D9B73988B  
306DD4FF8C70D963254BC1BEFB2F0FD0BCCDEB38FC963E559400DEFBC3EFF98B3B3174D582FEDA8CA10C2  
04E6340E5C807FFB57DCA8151317CB9DC3839281742D31D725AFACDEF6C843ED0FDF7139E6E32173A4B7B  
D25010F4B115953C61B41B4D8E8284314F748BC12501C7ED9DDE1D18C5EA15AEB6C4440A09F35B72BCBAE  
1E80AB79183C33195404C526F432219DCE686EB4943FFC4CA24A1231FD4CEF6BDAA3BF86991C5A2C0D6DA  
13

## 2) DECIPHER compliant PKCS#v1.5

==> Padding has to be compliant with DECRYPT/PKCS#v1.5: 00 02 RANDOM 00 DATA

==> len of RANDOM  $\geq 8$  and no byte equal to '00'

MSE SET DECRYPTER clé AUTH

-----

==> MSE SET DECIPHER

APDU: 012241B803830120 (8 bytes)

PSO DECIPHER (Envoyé avec le chainage des commandes en T=0)

=====

==> PS

APDU:  
112A80868000027c049873f872daa7cb354643d79889c6f089598d9f4742c3f939ef875cf839683bdbb5ce0a85  
278da254cbfb61b3c9417e0de04d323cc112188980eae6200551533558d4468c2f865762cb7cf1edef257108e4  
9486ff3e11139fa49a261ba779c41fbaba0070e799f24dcdd496dffbafb8e5b707bfae719a2cd2b629daf5 (133  
bytes)

SW12: 9000 [Command successfully executed (OK)]

==> PSO DECIPHER

APDU:

012A8086809021aea228491a1bf53bfdff8f9bcbdda40154079cf435a267f2bfaf10c725cb379d9aee8d39f9ac06  
29c05ad1f920b4faa01d3412cc420428dbce34a6d7736bc2a0f638aef65e3d8549ebd9a4675964ae2b817f090  
26f8cc55a2a1c0a310ef845108f0c39600FFEEEDCCBAA99887766554433221100001122334455667741 (133  
bytes)

SW12: 9000 (Command successfully executed (OK))

Data from card: (256 bytes)

65AB6896A62BC6157FDBEC2CE8A8DC0661C3134D04591CBA46E345B934C9F1FD8F8704B884674EC5246B1  
D4D625729E2365069AC622BAF545463713A9761E57247EA14C2CDDA0BB397AA30A3E75C820697A5DA61F  
D702E0DF26FF2DFBE0E9DEE6A16CB16687ABD1AD553A0D0B520D3431E81A84E27C4741C250B887CF65C6  
DF70FF2683D6A41EE3B1C697AEEE2EFC400B706CD582F8E1458CFA9116A9F0A1E51027D47B454AD8F0CA5  
8C00B0FF6B78507A5FCCC60EFB9F4E51A75A09B3889EBBC22B2C78E5F0D9ADED1679E0E5B7BCF6F1F8C207  
BFC2373840668B14894EF97E686F4FCDF6836FABE0D10ADA6AD1EFFFB2631A1936D188EF0C85280C00472A  
2B

[Event-ENDS]No Error.

>>>

## 8 ANNEXES

### 8.1 ATR

La reconnaissance d'une carte CPS (V3, V4 ...) peut être réalisée à l'aide de son ATR.



**Seuls les octets historiques doivent être testés pour la reconnaissance d'une carte CPS**

L'ATR des cartes CPS est compliant avec la norme ISO 7816-4.

Carte	ATR	Octets historiques
CPS V3	<b>'3B AC00402A 001225006480000310009000'</b>	'001225006480000310009000'
CPS V4 T=0 et T=1	<b>'3B 9B188001 0012250064800401000090 44'</b>	'0012250064800401000090'
CPS V4 T=0	<b>'3B DC18FF00 001225006480000401009000'</b>	'001225006480000401009000'

Tableau 26 : ATR cartes CPS

- Premier octet = octet TS      Toujours '3B' (convention directe)
- Octets **rouge**      Octets système ou Checksum, le premier octet est T0 = 'XY' avec Y = nombre d'octets historiques. Pour avoir les octets historiques, il faut donc prendre les 'Y' derniers octets de l'ATR. Attention dans le cas où il y a un autre protocole que le protocole T=0, ne pas prendre en compte le dernier octet qui est un CHECKSUM
- Octets **verts**      Octets historiques

Une fois les octets historiques extraits, se référer à la norme ISO 7816-4 pour avoir plus de précisions.  
Soient H1, H2 ... les octets historiques.

H1 = '00'	Category indicator byte: 0x00 (Norme ISO 7816), <b>compact TLV object</b>
H2 = '12'	Tag 1 = <b>Country Code</b> ISO 3166-1, Longueur = 2 octets suivants 'H3 H4' = '2500' ( <b>France</b> )
H5 = '64'	Tag 6 = Pre-issuing data, longueur sur 4 octets : 'H6 H7 H8 H9' = '80XXXXXX'
H11 H12 H13	Mandatory status indicator (3 last bytes) H11 = LCS (life card cycle): '00' è No information given 'H12 H13' = SW = '9000'



En conclusion, une fois les octets historiques de l'ATR extraits, la reconnaissance d'une carte CPS peut se faire sur les critères suivants :

- Nombre d'octets historiques au moins égal à 9
- H1H2H3H4H5H6 = '00122500 GIE est mi-septembre, leur ligne de PROD est 6480'
- H7H8H9 = '000310' → carte CPS V3
- H7H8H9 = '000401' → carte CPS V4 (T=0 uniquement)
- H7H8H9 = '040100' → carte CPS V4 (T=0 et T=1)



Pour la CPS V4, l'ATR (T=0 et T=1) n'est pas correct / norme ISO 7816-4 : par exemple H5 = '64' pour le Tag '6' alors qu'on n'a que 3 octets pour ce tag. Ces choix ont été effectués en fonction des contraintes terrain (compatibilité ascendante / librairies CPS V3).



**Bien que présenté, l'ATR (T=0 et T=1) n'est pas diffusé actuellement. Seul l'ATR T=0 seul est utilisé. Il faudrait cependant que les éditeurs qui testent l'ATR prennent en compte les 2 ATR afin de laisser la possibilité à l'ANS de faire évoluer la carte.**

## 8.2 Structure des données

Les données dans la CPS sont stockées sous forme d'objets au format TLV comme défini dans la norme ISO 7816-4. Chaque objet de données consiste en deux ou trois champs consécutifs :

- Un champ **tag** obligatoire,
- Un champ **longueur** obligatoire,
- Un champ **valeur** conditionnelle.

Le codage de ces champs se déroule comme suit :

- Le champ tag, composé d'un unique octet qui code un nombre tag compris entre 1 et 254. Les valeurs '00' et 'FF' sont invalides pour les champs de tags.
- Le champ de longueur est composé d'un, deux ou trois octets consécutifs.
  - Si le premier octet du champ longueur est inférieur 128 (bit 8 à 0), alors le champ de longueur est composé d'un seul octet qui code la longueur du champ valeur : nombre compris entre zéro et 127 et appelé N.
  - Si le premier octet du champ longueur est égale à '81', alors le champ longueur est constitué de deux octets '81XX' où 'XX' est la longueur du champ valeur : nombre compris entre zéro et 255 et appelé N
  - Si le premier octet du champ longueur est égale à '82', alors le champ longueur est constitué de trois octets '82XXYY' où 'XXYY' est la longueur du champ valeur : nombre compris entre zéro et 65535 et appelé N
- Si N vaut zéro, il n'y a aucun champ de valeur, c'est-à-dire que l'objet de données est vide. Sinon (N > 0), le champ de valeur est composé de N octets consécutifs.

La méthode de représentation de données primitive permet de représenter une unique information codée BER tel qu'un nombre entier, un offset, une longueur, etc. Quand plusieurs objets primitifs sont nécessaires à la représentation des données, on utilise une méthode de codage d'objets de données BER-TLV construite.

- Tout objet de données **BER-TLV** est noté **{T L V}** avec un champ tag suivi d'un champ de longueur de codage d'un nombre. Selon que le nombre est nul ou non, le champ de valeur est absent (objet de données vide) ou présent.
- Tout objet de données **BER-TLV construit** est noté **{TL {T1 L1 V1} ... - {Tn Ln Vn}}** avec un champ tag suivi d'un champ de longueur de codage d'un nombre. Si le nombre est différent de zéro, alors le champ de valeur de l'objet de données construit, c'est-à-dire, le modèle, est composé d'un ou de plusieurs objets de données **BER-TLV**, consistant respectivement d'un champ tag, d'un champ de longueur de codage d'un nombre et si le nombre est différent de zéro, d'un champ de valeur.



**Le tag '00' est réservé, il est utilisé pour indiquer la fin logique des objets TLV dans un EF ou dans un objet construit**

## 8.3 Récapitulatif des identifiants

Répertoires	Chemin	Identifiant
Application régaliennne	'3F00'	'80 25 00 00 01 FF 01 00'
ADF CPS	'3F00/0001'	'E8 28 BD 08 0F 80 25 00 00 01 FF 00 10'
DF SIGN	'3F00/0001/0101'	
DF AUTH	'3F00/0001/0102'	
DF CPS2TER	'3F00/0003'	
DF AMO	'3F00/0003/7F01'	
Fichiers/Objets de sécurité	Chemin	Identifiant
EF DIR	'3F00/2F00'	
EF ATR	'3F00/2F01'	
EF SN	'3F00/D003'	
EF TECH	'3F00/2FFF'	
EF ID CARTE	'3F00/0001/D101'	
EF NOM	'3F00/0001/D102'	
EF LANG	'3F00/0001/D103'	
EF INFO_PS	'3F00/0001/D104'	
EF APP_DATA	'3F00/0001/D107'	
EF PS_SIT_XX	'3F00/0001/D120-D12F'	
EF CERT SIG	'3F00/0001/0101/A010'	
EF PUB SIG	'3F00/0001/0101/9010'	
EF CERT AUTH	'3F00/0001/0102/A020'	
EF PUB AUTH	'3F00/0001/0102/9020'	
EF OD	'3F00/0001/5031'	
EF CIAINFO	'3F00/0001/5032'	
EF AOD	'3F00/0001/7001'	
EF PrKD	'3F00/0001/7002'	
EF PuKD	'3F00/0001/7004'	
EF CD	'3F00/0001/7005'	
EF DCOD	'3F00/0001/7006'	
2TER DIR	'3F00/0003/2F00'	
2TER ATR	'3F00/0003/2F01'	
2TER IC	'3F00/0003/0005'	
2TER ICC	'3F00/0003/0002'	
2TER ID CARD	'3F00/0003/0003'	
2TER NAME	'3F00/0003/0004'	
2TER LANG	'3F00/0003/0006'	
2TER PS_IDNAT	'3F00/0003/4000'	
2TER PS_QUALIF	'3F00/0003/4001'	
2TER EF PS_SIT_XX	'3F00/0003/4010-401F'	
AMO DIR	'3F00/0003/7F01/2F00'	
AMO EF	'3F00/0003/7F01/4000'	
Clé privée SIGN		'10'
Clé privée AUTH		'20'
PIN		'01'
PUK		'02'

**Tableau 27 : Identifiants CPS V4**

## 8.4 Différences V3/V4

Item	V3	V4
SELECTION APPLICATION	Par défaut Application régaliennne (IAS) sélectionnée	Par défaut, application émulation CPS2TER sélectionnée.  Applet régaliennne non sélectionnée, sélection nécessaire avant de pouvoir travailler avec cette application.
DONNEES CPS2TER	Présentes dans une autre application (CPS2TER) accessibles via des commandes propriétaires (classe 'A0')	Toutes les données sont accessibles dans une même application, l'application régaliennne (volet 2TER '0003')  <b>Ou bien (à éviter)</b>  <del>Au travers des commandes 'A0' et l'application émulation CPS2TER</del>
CONDENSE	Il existe une commande PSO HASH	Pas de commande PSO HASH
SIGNATURE SANS CONDENSE	Existe de manière « native » avec la commande INTERNAL AUTHENTICATE	N'existe pas mais il est possible de contourner avec la commande PSO DECIPHER
PADDING RSA PSS	N'existe pas	Supporté
MIDDLEWARE	Canal 0 utilisé pour toutes les applications	Canal 0 utilisé pour l'application émulation CPS2TER et Canal 1 utilisé pour l'application régaliennne
PIN	Padding du PIN (ASCII) par des 'FF' jusqu'à 8 octets	Pas de padding à faire

Tableau 28 : Différences V3/V4