OACI

Doc 9303

Documents de voyage lisibles à la machine

Huitième édition, 2021

Partie 11 : Mécanismes de sécurité pour les DVLM

Approuvé par la Secrétaire générale et publié sous son autorité

ORGANISATION DE L'AVIATION CIVILE INTERNATIONALE

OACI

Doc 9303

Documents de voyage lisibles à la machine

Huitième édition, 2021

Partie 11 : Mécanismes de sécurité pour les DVLM

Approuvé par la Secrétaire générale et publié sous son autorité

ORGANISATION DE L'AVIATION CIVILE INTERNATIONALE

Publié séparément en français, en anglais, en arabe, en chinois, en espagnol et en russe par l'ORGANISATION DE L'AVIATION CIVILE INTERNATIONALE
999, boul. Robert-Bourassa, Montréal (Québec) H3C 5H7 Canada

Le site www.icao.int/Security/FAL/TRIP permet de télécharger les documents et d'obtenir des renseignements supplémentaires.

Doc 9303, *Documents de voyage lisibles à la machine*
Partie 11 — *Mécanismes de sécurité pour les DVLM*

Commande nº : 9303P11
ISBN 978-92-9265-569-3 (version imprimée)
ISBN 978-92-9275-554-6 (version électronique)

AMENDEMENTS

La parution des amendements est annoncée dans les suppléments au *Catalogue des produits et services*. Le Catalogue et ses suppléments sont disponibles sur le site web de l'Organisation (www.icao.int). Le tableau ci-dessous est destiné à rappeler les divers amendements.

RELEVÉ DES AMENDEMENTS ET DES RECTIFICATIFS

AMENDEMENTS		
N°	Date	Inscrit par
1	14/6/24	OACI

RECTIFICATIFS		
N°	Date	Inscrit par

Les appellations employées dans cette publication et la présentation des éléments qui y figurent n'impliquent de la part de l'OACI aucune prise de position quant au statut juridique des pays, territoires, villes ou zones, ou de leurs autorités, ni quant au tracé de leurs frontières ou limites.

TABLE DES MATIÈRES

Page

1. PORTÉE

La Partie 11 du Doc 9303 présente des spécifications destinées à permettre aux États et aux fournisseurs de mettre en œuvre des éléments de sécurité cryptographiques pour les DVLM électroniques (DVLM-e) avec accès à un circuit intégré (CI) sans contact. Les protocoles cryptographiques spécifiés visent à :

- empêcher l'écrémage des données contenues dans le CI sans contact ;
- empêcher l'interception illicite des communications entre le CI sans contact et le lecteur ;
- assurer l'authentification des données stockées dans le CI sans contact sur la base de l'infrastructure à clés publiques (ICP) décrite à la Partie 12 ;
- assurer l'authentification du CI sans contact lui-même.

La huitième édition du Doc 9303 intègre les spécifications des applications optionnelles de dossiers de voyage, de dossiers de visa et d'éléments biométriques supplémentaires (appelées applications SDL2) en tant qu'extension optionnelle du DVLM-e. La présente partie du Doc 9303 inclut les protocoles de contrôle d'accès étendus nécessaires pour protéger l'écriture et la lecture des données des applications SDL2 respectives. Ces protocoles de contrôle d'accès peuvent également être utilisés pour la protection des données biométriques secondaires dans l'application DVLM-e.

L'authentification des données stockées dans le CI sans contact est l'élément de sécurité de base qui permet l'emploi des CI pour l'inspection manuelle et/ou automatisée. Cette fonction est donc REQUISE.

La mise en œuvre d'un protocole pour empêcher l'écrémage des données stockées dans le CI sans contact et prévenir l'interception illicite des communications entre le CI et le terminal est REQUISE.

La mise en œuvre d'autres protocoles est OPTIONNELLE ; l'État émetteur ou l'organisation émettrice peut ainsi choisir l'ensemble d'éléments de sécurité nécessaire en fonction de la réglementation et des exigences nationales.

La présente partie doit être lue en parallèle avec les parties suivantes du Doc 9303 :

- Partie 1 — *Introduction* ;
- Partie 10 — *Structure de données logique (SDL) pour le stockage des données biométriques et d'autres données dans le circuit intégré (CI) sans contact ;*
- Partie 12 — *Infrastructure à clés publiques pour les DVLM*.

2. PRÉSUPPOSÉS ET NOTATIONS

Il est supposé que le lecteur du présent document est familiarisé avec les concepts et les mécanismes offerts par la cryptographie à clé publique et les infrastructures à clés publiques.

L'usage des techniques de cryptographie à clé publique ajoute une certaine complexité à la mise en œuvre des DVLM-e, mais ces techniques ajoutent de la valeur en ce sens qu'elles procurent aux postes de contrôle frontalier de première ligne une mesure supplémentaire de détermination de l'authenticité des DVLM-e. Cette technique n'est pas censée

être la seule mesure de détermination de l'authenticité et elle NE DEVRAIT PAS être employée comme unique facteur déterminant.

L'impossibilité d'utiliser les données du CI sans contact, en raison d'une révocation du certificat ou d'une vérification de signature non valide, ou parce que le CI sans contact a été intentionnellement laissé sans données (voir section 4.5.4 du Doc 9303-10), ne signifie pas nécessairement que le DVLM-e n'est pas valide. En pareil cas, un État récepteur PEUT se fier, aux fins de validation, à d'autres éléments de sécurité du document.

2.1 Spécifications relatives aux puces et aux terminaux de DVLM-e

La présente partie du Doc 9303 présente les spécifications applicables aux mises en œuvre des puces (ou CI équivalent) et des terminaux (ou systèmes d'inspection) des DVLM-e. Les puces des DVLM-e doivent être conformes à ces spécifications selon la terminologie indiquée dans le Doc 9303-1, mais les spécifications relatives aux terminaux doivent être interprétées comme des éléments indicatifs, c'est-à-dire que l'interopérabilité de la puce et du terminal du DVLM-e n'est garantie que si le terminal est conforme à ces spécifications ; s'il n'est pas conforme, le terminal ne pourra pas interagir avec la puce du DVLM-e ou la puce aura un comportement indéterminé. En général, il n'est pas nécessaire que la puce du DVLM-e applique les spécifications relatives aux terminaux à moins que la sécurité de la puce du DVLM-e ne soit directement en cause.

2.2 Notations

Les notations suivantes sont employées pour désigner les primitives cryptographiques indépendamment des algorithmes :

- chiffrement de texte en clair S avec une clé symétrique K : **E**(K, S).
- déchiffrement de texte chiffré C avec une clé symétrique K : **D**(K, C).
- opération de calcul d'un hachage sur un message m : **H**(m).
- calcul d'un code d'authentification de message avec une clé symétrique K sur un message M : **MAC**(K,M).
- agrément de clé basé sur des paires de clés asymétriques (SK, PK) et (SK', PK') et des paramètres de domaine D : **KA**(SK,PK',D) / **KA**(SK',PK,D).
- calcul de clé à partir d'un secret partagé S : **KDF** (S).
- signature d'un message *m* avec la clé privée SK_{IFD} est désignée par *s* = **Sign**(SK_{IFD}, m) ;
- vérification de la signature résultante *s* avec la clé publique PK_{IFD} et le message *m* : **Verify**(PK_{IFD}, s, m) ;
- calcul d'une représentation comprimée d'une clé publique *PK* : **Comp***(PK)*.

3. SÉCURISATION DES DONNÉES ÉLECTRONIQUES

En plus de l'authentification passive par signatures numériques et du contrôle d'accès à la puce, les États émetteurs ou les organisations émettrices PEUVENT assurer une sécurisation supplémentaire en employant des moyens plus complexes de sécurisation du CI sans contact et de ses données.

L'accès à un DVLM-e comprend les étapes suivantes :

1. Accès au CI sans contact du DVLM-e (section 4)
2. Authentification des données (section 5)
3. Authentification de la puce (section 6)
4. Mécanismes de contrôle d'accès supplémentaires (section 7)
5. Lecture des données (voir le Doc 9303-10)

Il existe différents protocoles pour les différentes étapes. La configuration exacte d'un DVLM-e est choisie par l'État émetteur ou l'organisation émettrice. Les options présentées dans le Tableau 1 peuvent être convenablement combinées pour apporter une sécurisation supplémentaire selon les besoins des émetteurs.

Les procédures d'inspection pour des configurations différentes des DVLM-e sont décrites à l'Appendice J.

Tableau 1. Sécurisation des données électroniques (sommaire)

Méthode	*CI sans contact*	*Système d'inspection*	*Avantages*	*Note*
MÉTHODE DE SÉCURISATION DE BASE				
Authentification passive (§ 5.1)	m	m	Prouve que le contenu du SO_D et celui de la SDL sont authentiques et n'ont pas été modifiés.	N'empêche ni la copie exacte, ni la substitution du CI. N'empêche pas l'accès non autorisé. N'empêche pas l'écrémage.
MÉTHODES DE SÉCURISATION AVANCÉES				
Comparaison de la ZLA conventionnelle (ROC-B) et de la ZLA sur le CI (SDL)	s.o.	o	Prouve que le contenu du CI sans contact et le DVLM-e physique vont ensemble.	Ajoute de la complexité (mineure). N'empêche pas une copie exacte du CI sans contact ou du document conventionnel.
Authentification active (§ 6.1)	o	o	Empêche de copier le SO_D et prouve qu'il a été lu à partir du CI sans contact authentique. Prouve que le CI sans contact n'a pas été remplacé.	N'empêche pas l'accès non autorisé. Ajoute de la complexité. L'authentification de la puce est REQUISE pour SDL2.
Authentification de la puce (§ 6.2)	o/c	o		

Méthode	*CI sans contact*	*Système d'inspection*	*Avantages*	*Note*
Contrôle d'accès de base (BAC) (§ 4.3)	c (voir aussi § 4.1)	M (voir aussi § 4.1)	Empêche l'écrémage et l'utilisation abusive. Empêche l'interception illicite des communications entre le DVLM-e et le système d'inspection (lorsqu'il est employé pour établir un canal de session chiffré).	N'empêche ni la copie exacte, ni la substitution du CI (exige aussi de copier le document conventionnel). Ajoute de la complexité. Le DVLM-e DOIT prendre en charge au moins BAC ou PACE. PACE est REQUIS pour SDL2. PACE offre une meilleure protection contre l'interception illicite que BAC. Voir aussi l'Appendice A.
Établissement de connexion avec authentification par mot de passe (PACE) (§ 4.4)	r/c (voir aussi § 4.1)	m (voir aussi § 4.1)		
Authentification du terminal (§ 7.1)	o/c	o	Empêche l'accès non autorisé aux données sensibles. Empêche l'écrémage des données sensibles.	Exige la gestion d'une clé supplémentaire. N'empêche ni la copie exacte ni la substitution du CI (exige aussi de copier le document conventionnel). Ajoute de la complexité. L'authentification du terminal est REQUISE pour SDL2.
Chiffrement des données (§ 7.2)	o	o	Sécurise les éléments biométriques additionnels. N'exige pas de CI processeur.	Exige une gestion complexe de clés de déchiffrement. N'empêche ni la copie exacte, ni la substitution du CI. Ajoute de la complexité.
m = REQUIS, r = RECOMMANDÉ, o = OPTIONNEL, c = CONDITIONNEL, s.o. = sans objet.				

Note.— Voir la section 4 pour les renseignements détaillés sur les configurations conformes des CI sans contact en ce qui concerne le contrôle d'accès de base (BAC) et l'établissement de connexion avec authentification par mot de passe (PACE).

La mise en œuvre des méthodes de sécurisation avancées indiquées dans le Tableau 1 n'a pas d'incidence sur la conformité avec l'OACI.

4. ACCÈS AU CI SANS CONTACT

L'ajout d'un CI sans contact qui ne comporte pas de contrôle d'accès au DVLM-e présente deux nouvelles possibilités d'attaque :

- les données stockées dans le CI sans contact peuvent être lues électroniquement sans autorisation de lecture du document (écrémage) ;

- une interception illicite des communications non chiffrées entre un CI sans contact et un lecteur est possible à plusieurs mètres de distance.

Des dispositions matérielles peuvent être prises contre l'écrémage (p. ex., le blindage à l'aide d'une trame métallique dans la couverture du passeport en livret), mais elles n'empêchent pas l'interception illicite. Les États émetteurs et les organisations émettrices DOIVENT donc mettre en œuvre un mécanisme de contrôle d'accès à la puce, c'est-à-dire un mécanisme de contrôle d'accès qui exige de fait que le détenteur du DVLM-e sache que les données stockées dans le CI sans contact sont en train d'être lues d'une façon sécurisée. Ce mécanisme de contrôle d'accès à la puce empêche l'écrémage aussi bien que l'interception illicite.

Un CI sans contact protégé par un mécanisme de contrôle d'accès à la puce refuse l'accès à son contenu sauf si le système d'inspection peut prouver qu'il est autorisé à accéder au CI sans contact. Cette preuve est donnée dans un protocole cryptographique, où le système d'inspection prouve la connaissance des informations provenant du document physique.

Le système d'inspection DOIT obtenir cette information avant d'être capable de lire le CI sans contact. L'information doit être extraite optiquement/visuellement du DVLM-e (p. ex., de la ZLA). Il DOIT aussi être possible à un inspecteur d'introduire manuellement cette information dans le système d'inspection si la lecture automatique de l'information n'est pas possible.

En partant du principe que les informations du document physique ne peuvent pas être obtenues à partir d'un document qui n'est pas vu (étant donné qu'elles sont tirées de la ZLA à lecture optique), on peut établir que le DVLM-e est sciemment remis pour inspection. Du fait du chiffrement du canal, l'interception illicite de la communication demanderait un effort considérable.

La présente section définit deux mécanismes de contrôle d'accès à la puce :

- le contrôle d'accès de base (BAC, § 4.3), purement fondé sur la cryptographie symétrique ;
- l'établissement de connexion avec authentification par mot de passe (PACE, § 4.4), qui emploie la cryptographie asymétrique pour fournir des clés de session à plus grande entropie.

Voir aussi l'Appendice A pour plus de renseignements sur la force des clés de session.

4.1 Configurations conformes

Les configurations suivantes sont conformes à cette spécification :

- puces de DVLM-e avec BAC seulement ;
- puces de DVLM-e avec PACE *et* BAC ;
- puces de DVLM-e avec PACE seulement.

La sécurité fournie par le contrôle d'accès de base est limitée par la conception du protocole, comme il est expliqué dans l'Aappendice A. En raison de l'augmentation prévue de la puissance de calcul des ordinateurs au fil des ans, il sera possible de mener avec succès des attaques contre le BAC avec des moyens financiers modestes et dans un délai réduit. Par conséquent, il est convenu de procéder à un passage progressif de BAC à PACE.

La période de transition suivante a été établie :

• À partir du 1er janvier 2027, les puces de DVLM-e doivent utiliser PACE et les puces de DVLM-e. avec BAC seulement deviennent obsolètes. Toutes les puces de DVLM-e avec BAC seulement émises avant le 1er janvier 2027 demeurent conformes durant toute leur période de validité.

• À partir du 1er janvier 2028, BAC devient obsolète et les puces de DVLM-e DOIVENT utiliser PACE seulement. Tous les DVLM-e utilisant PACE et BAC émis avant le 1er janvier 2028 demeurent conformes durant toute leur période de validité.

Les systèmes d'inspection conformes DOIVENT prendre en charge toutes les configurations de DVLM-e conformes. Si un DVLM-e prend en charge à la fois PACE et BAC, le système d'inspection DOIT utiliser soit BAC, soit PACE, mais non les deux durant une même session.

Note 1.— Les versions précédentes du Doc 9303 autorisaient les puces de DVLM-e sans contrôle d'accès à la puce (DVLM-e ordinaires). Ceci est obsolète dans la huitième édition. Néanmoins, les systèmes d'inspection conformes DOIVENT prendre en charge les DVLM-e sans contrôle d'accès à la puce.

Note 2.— Pour l'accès aux applications SDL2, le CI DOIT exiger l'exécution de PACE.

4.2 Procédure d'accès à la puce

La procédure d'accès à la puce pour authentifier le système d'inspection comprend les étapes indiquées ci-après :

1. **Lecture EF.CardAccess** **(REQUIS)**

   Si le DVLM-e prend en charge PACE, la puce du DVLM-e DOIT fournir les paramètres à employer pour PACE dans le fichier EF.CardAccess.

   Si le fichier EF.CardAccess est disponible, le système d'inspection DOIT le lire (voir § 9.2.11) pour déterminer les paramètres (c'est-à-dire algorithmes cryptographiques symétriques, algorithmes d'agrément de clé, paramètres de domaine et mappages) pris en charge par la puce du DVLM-e. Le système d'inspection peut choisir n'importe lequel de ces paramètres.

   Si le fichier EF.CardAccess n'est pas disponible ou ne contient pas les paramètres pour PACE, le système d'inspection DEVRAIT essayer de lire le DVLM-e avec le BAC (passer à l'étape 4).

2. **Lecture d'EF.DIR** **(OPTIONNEL)**

   Le système d'inspection PEUT lire EF.DIR (s'il est présent) pour récupérer une liste des applications présentes sur la puce du DVLM-e.

3. **PACE** **(CONDITIONNEL)**

   La présente étape est RECOMMANDÉE si la puce du DVLM-e prend en charge PACE. Elle est REQUISE si l'accès aux applications SDL2 est prévu.

   • Le système d'inspection DEVRAIT calculer la clé K_{π} à partir de la ZLA. Il PEUT employer le code d'accès à la carte (CAN) au lieu de la ZLA si le système d'inspection connaît le CAN.

   • La puce du DVLM-e DOIT accepter la ZLA comme mots de passe pour PACE. Elle PEUT aussi accepter le CAN au lieu de la ZLA.

   • Le système d'inspection et la puce du DVLM-e s'authentifient mutuellement au moyen de K_{π} et calculent les clés de session KS_{Enc} et KS_{MAC}. Le protocole PACE décrit au § 4.4 DOIT être utilisé.

Si cette procédure réussit, la puce du DVLM-e exécute les étapes suivantes :

- Elle DOIT démarrer la messagerie sécurisée.
- Elle DOIT accorder l'accès aux données moins sensibles (p. ex., EF.DG1, EF.DG2, EF.DG14, EF.DG15, etc., de l'application DVLM-e, et l'objet de sécurité du document. Voir la définition de « données sensibles » dans le Doc 9303-1).
- Elle DOIT restreindre les droits d'accès pour exiger la messagerie sécurisée.

Le système d'inspection DOIT vérifier l'authenticité du contenu du fichier EF.CardAccess en utilisant EF.DG14 ou EF.CardSecurity, et du fichier EF.DIR (s'il est présent et lu) en utilisant EF.CardSecurity.

Note.— Si aucune application SDL2 n'est présente sur la puce du DVLM-e, il se peut qu'EF.CardSecurity ne contienne pas une copie sécurisée du fichier EF.DIR.

4. **Contrôle d'accès de base** **(CONDITIONNEL)**

La présente étape est REQUISE si le contrôle d'accès à la puce est appliqué par la puce du DVLM-e et que PACE n'a pas été utilisé. Si PACE a été exécuté avec succès ou si le DVLM-e n'applique pas le contrôle d'accès à la puce, cette étape est omise.

L'application DVLM-e DOIT être sélectionnée avant que le contrôle d'accès de base ne soit effectué.

- Le système d'inspection DEVRAIT calculer, à partir de la ZLA, les clés d'accès de base au document (K_{Enc} et K_{MAC}).
- Le système d'inspection et la puce du DVLM-e s'authentifient mutuellement au moyen des clés d'accès de base au document et calculent les clés de session KS_{Enc} et KS_{MAC}.

Si cette procédure réussit, la puce du DVLM-e :

- DOIT démarrer la messagerie sécurisée ;
- DOIT accorder l'accès aux données moins sensibles (p. ex., EF.DG1, EF.DG2, EF.DG14, EF.DG15, etc., de l'application DVLM-e, et l'objet de sécurité du document) ;
- DOIT restreindre les droits d'accès pour exiger la messagerie sécurisée.

Note.— En raison de la procédure d'accès à la puce, le DF actuel peut être soit le fichier principal (si PACE a été utilisé) ou l'application du DVLM-e (si BAC a été utilisé).

4.3 Contrôle d'accès de base

4.3.1 Spécification du protocole

Pour l'authentification et l'établissement de clés, on utilise un protocole d'authentification par question-réponse à trois passages, conformément au mécanisme d'établissement de clés 6 de l'ISO/IEC 11770-2, en utilisant la norme de chiffrement de données 3DES [FIPS 46-3] pour le chiffrement par blocs. Une somme de contrôle cryptographique selon l'algorithme MAC 3 de l'ISO/IEC 9797-1 est calculée sur les cryptogrammes et y est jointe. Les modes opératoires décrits au § 4.3.3 DOIVENT être utilisés. Les mots de circonstance (nonces) échangés DOIVENT avoir une taille de 8 octets ; les données de clé échangées DOIVENT avoir une taille de 16 octets. Le dispositif d'interface (IFD) (c.-à-d. système d'inspection) et le CI sans contact NE DOIVENT PAS utiliser des identificateurs distinctifs comme nonces.

De façon plus détaillée, l'IFD et le CI DOIVENT exécuter les étapes suivantes :

1) L'IFD demande une question RND.IC en envoyant la commande GET CHALLENGE (acquérir question). Le CI génère un nonce RND.IC et répond en utilisant ce nonce.

2) L'IFD effectue les opérations suivantes :

   a) générer un nonce RND.IFD et les données de clé K.IFD ;

   b) générer la concaténation S = RND.IFD || RND.IC || K.IFD ;

   c) calculer le cryptogramme E_{IFD} = **E**($K_{Enc,}$ S) ;

   d) calculer la somme de contrôle M_{IFD} = **MAC**($K_{MAC,}$ E_{IFD}) ;

   e) envoyer la commande EXTERNAL AUTHENTICATE (authentification externe) avec la fonction d'authentification mutuelle en utilisant les données E_{IFD} || M_{IFD}.

3) Le CI effectue les opérations suivantes :

   a) vérifier la somme de contrôle M_{IFD} du cryptogramme E_{IFD} ;

   b) décrypter le cryptogramme E_{IFD} ;

   c) extraire RND.IC de S et vérifier si IFD a retourné la bonne valeur ;

   d) générer les données de clé K.IC ;

   e) générer la concaténation R = RND.IC || RND.IFD || K.IC ;

   f) calculer le cryptogramme E_{IC} = **E**(K_{Enc}, R) ;

   g) calculer la somme de contrôle M_{IC} = **MAC**(K_{MAC}, E_{IC}) ;

   h) envoyer la réponse en utilisant les données E_{IC} || M_{IC}.

4) L'IFD effectue les opérations suivantes :

   a) vérifier la somme de contrôle M_{IC} du cryptogramme E_{IC} ;

   b) déchiffrer le cryptogramme E_{IC} ;

   c) extraire RND.IFD de R et vérifier si le CI a retourné la bonne valeur.

5) L'IFD et le CI calculent les clés de session KS_{Enc} et KS_{MAC} au moyen du mécanisme de calcul de clé décrit aux § 9.7.1 et § 9.7.4 avec (K.IC xou K.IFD) comme secret partagé.

4.3.2 Processus d'inspection

Lorsqu'un DVLM-e avec contrôle d'accès de base (BAC) est présenté au système d'inspection, l'information lue optiquement ou visuellement est utilisée pour calculer les clés d'accès de base au document (K_{Enc} et K_{MAC}) afin d'obtenir l'accès au CI sans contact et d'établir un canal sécurisé pour les communications entre le CI sans contact du DVLM-e et le système d'inspection.

Le CI sans contact d'un DVLM-e qui prend en charge le contrôle d'accès de base DOIT réagir aux tentatives de lecture non authentifiées, c'est-à-dire des tentatives de lecture envoyées sans messagerie sécurisée [y compris la sélection de fichiers (protégés) dans la SDL], par « état de sécurité non satisfait » (0x6982) une fois que le canal sécurisé est établi. Si le CI reçoit un SELECT en clair (c'est-à-dire sans que la messagerie sécurisée ait été appliquée) dans le canal sécurisé, le CI DOIT interrompre le canal sécurisé. Si un SELECT en clair est envoyé avant que le canal sécurisé soit établi ou s'il y a eu interruption du canal sécurisé, le CI PEUT renvoyer tant 0x6982 que 0x9000, qui sont des réponses conformes aux normes de l'OACI.

Pour authentifier le système d'inspection, les étapes suivantes DOIVENT être exécutées :

1) Le système d'inspection lit l'information_ZLA, constituée de la concaténation du numéro de document, de la date de naissance et de la date d'expiration, y compris leurs chiffres de contrôle respectifs, comme il est décrit dans les Doc 9303-4, 9303-5 ou 9303-6 pour les formats de document TD3, TD1 et TD2 respectivement, à partir de la ZLA en utilisant un lecteur ROC-B. L'information requise peut aussi être entrée manuellement ; dans ce cas, elle DOIT être entrée telle qu'elle figure dans la ZLA. Les 16 octets les plus significatifs de la valeur de hachage SHA-1 de cette information_ZLA sont utilisés comme germe de clés pour calculer les clés d'accès de base au document, en utilisant le mécanisme de calcul de clé décrit au § 9.7.2.

2) Le système d'inspection et le CI sans contact du DVLM-e s'authentifient mutuellement et calculent les clés de session. Le protocole d'authentification et d'établissement de clés décrit plus haut DOIT être utilisé.

3) Après l'exécution réussie du protocole d'authentification, l'IFD et le CI calculent tous deux les clés de session KS_{Enc} et KS_{MAC} en utilisant le mécanisme de calcul de clé décrit aux § 9.7.1 et § 9.7.4, avec (K.IC xou K.IFD) comme secret partagé. Toutes les communications ultérieures DOIVENT être protégées par messagerie sécurisée, comme il est décrit au § 9.8.

4.3.3 Spécifications cryptographiques

4.3.3.1 Chiffrement de la question et de la réponse

Le chiffrement triple DES (3DES) à deux clés en mode CBC avec zéro IV (c.-à-d. 0x00 00 00 00 00 00 00 00) selon la norme ISO/IEC 11568-2 DOIT être utilisé pour le calcul de E_{IFD} et E_{IC}. AUCUN remplissage NE DOIT être utilisé pour les données d'entrée lors de l'exécution de la commande EXTERNAL AUTHENTICATE (authentification externe).

4.3.3.2 Authentification de la question et de la réponse

Les sommes de contrôle cryptographiques M_{IFD} et M_{IC} DOIVENT être calculées à l'aide de l'algorithme MAC 3 de la norme ISO/IEC 9797-1 avec le chiffrement par bloc DES, zéro IV (8 octets) et la méthode de remplissage 2 de l'ISO/IEC 9797-1. La longueur de MAC DOIT être de 8 octets.

4.3.4 Unités de données du protocole d'application

Le contrôle d'accès de base est exécuté au moyen des commandes GET CHALLENGE (acquérir question) et EXTERNAL AUTHENTICATE (authentification externe) et de la fonction d'authentification mutuelle. Les commandes DOIVENT être codées conformément à la norme ISO/IEC 7816-4.

4.3.4.1 GET CHALLENGE (acquérir question)

Commande		
CLA		Propre au contexte
INS	0x84	GET CHALLENGE
P1/P2	0x0000	—
Données		Absentes
Réponse		
Données	Nonce aléatoire	
Octets d'état	0x9000	*Traitement normal* Nonce aléatoire généré et transmis avec succès.
	Autre	*Erreur dépendant du système d'exploitation* Le nonce aléatoire n'a pas pu être transmis.

4.3.4.2 EXTERNAL AUTHENTICATE (authentification externe)

Commande			
CLA		Propre au contexte	
INS	0x82	EXTERNAL AUTHENTICATE	
P1/P2	0x0000	—	
Données		Données de commande E_{IFD} || M_{IFD}	REQUIS
Réponse			
Données		Données de réponse E_{IC} || M_{IC}	REQUIS
Octets d'état	0x9000	*Traitement normal* Protocole exécuté avec succès.	
	Autre	*Erreur dépendant du système d'exploitation* Échec du protocole.	

4.4 Établissement de connexion avec authentification par mot de passe (PACE)

PACE est un protocole d'agrément de clés Diffie-Hellman avec authentification par mot de passe qui assure des communications sécurisées et une authentification par mot de passe de la puce du DVLM-e et du système d'inspection (c'est-à-dire que la puce du DVLM-e et le système d'inspection utilisent le même mot de passe π).

PACE établit la messagerie sécurisée entre la puce du DVLM-e et un système d'inspection au moyen de mots de passe faibles (courts). Le contexte de sécurité est établi dans le fichier principal. Le protocole permet à la puce du DVLM-e de vérifier si le système d'inspection est autorisé à accéder aux données stockées et présente les caractéristiques suivantes :

- Les clés de session fournies sont fortes quelle que soit la force du mot de passe.
- L'entropie du ou des mots de passe utilisés pour authentifier le système d'inspection peut être très faible (p. ex., 6 chiffres suffisent en général).

PACE utilise des clés K_π calculées à partir de mots de passe avec une fonction de calcul de clé $\mathbf{KDF_\pi}$ (voir § 9.7.3). Les deux mots de passe suivants et les clés correspondantes sont disponibles pour les DVLM interopérables à l'échelle mondiale :

- ZLA : la clé K_π définie par $K_\pi = \mathbf{KDF_\pi}$ (ZLA) est REQUISE. Elle est calculée à partir de la zone de lecture automatique (ZLA) d'une manière semblable au contrôle d'accès de base, c'est-à-dire calculée à partir du numéro du document, de la date de naissance et de la date d'expiration.
- CAN : la clé K_π définie par $K_\pi = \mathbf{KDF_\pi}$(CAN) est OPTIONNELLE. Elle est calculée à partir du numéro d'accès à la carte (CAN). Le CAN est un numéro imprimé sur le document et DOIT être choisi de façon aléatoire ou pseudo-aléatoire (p. ex. en utilisant une fonction pseudo-aléatoire cryptographique forte). Le Doc 9303, parties 4, 5 et 6 spécifie le champ du CAN.

Note.— Le CAN a l'avantage de pouvoir être facilement entré manuellement, contrairement à la ZLA (numéro de document, date de naissance, date d'expiration).

PACE prend en charge différents mappages dans le cadre de l'exécution du protocole :

- le *mappage générique* basé sur l'agrément de clé Diffie-Hellman ;
- le *mappage intégré* basé sur un mappage direct d'un élément de champ avec le groupe cryptographique ;
- le *mappage d'authentification de puce* élargit le mappage générique et intègre l'authentification de la puce dans le protocole PACE.

Si la puce prend en charge le mappage d'authentification de puce, elle DOIT aussi prendre en charge au moins le mappage générique ou le mappage intégré et l'authentification de la puce, ce qui signifie que pour les systèmes d'inspection avec PACE, seul le mappage générique ou le mappage intégré est REQUIS. La prise en charge du mappage d'authentification de puce est OPTIONNELLE.

4.4.1 Spécification du protocole

Le système d'inspection lit les paramètres de PACE pris en charge par la puce du DVLM-e dans le fichier EF.CardAccess (voir § 9.2.11) et sélectionne les paramètres à employer, puis le protocole est exécuté.

Les commandes suivantes DOIVENT être utilisées :

- READ BINARY (lire binaire), spécifiée dans le Doc 9303-10.
- MSE:Set AT [commande MANAGE SECURITY ENVIRONMENT (gestion de l'environnement de sécurité) avec la fonction de gabarit d'authentification Set], comme il est spécifié au § 4.4.4.1.

- Le système d'inspection et la puce du DVLM-e DOIVENT exécuter les étapes suivantes en utilisant une chaîne de commandes d'AUTHENTIFICATION GÉNÉRALE, comme il est spécifié au § 4.4.4.2 :

  1) La puce du DVLM-e choisit un nonce ***s*** de manière aléatoire et uniforme, chiffre le nonce en utilisant $z = \mathbf{E}(K_\pi, s)$, où $K_\pi = \mathbf{KDF}_\pi$ (π) est calculé à partir du mot de passe partagé π, et envoie le texte chiffré ***z*** au système d'inspection.

  2) Le système d'inspection retrouve le texte en clair $s = \mathbf{D}(K_\pi, z)$ à l'aide du mot de passe partagé π.

  3) La puce du DVLM-e et le système d'inspection exécutent les étapes suivantes :

     a) Ils échangent les données supplémentaires requises pour le mappage du nonce :

        i) pour le mappage générique, la puce du DVLM-e et le système d'inspection échangent des clés publiques éphémères ;

        ii) pour le mappage intégré, le système d'inspection envoie un nonce supplémentaire à la puce du DVLM-e.

     b) Ils calculent les paramètres de domaine éphémères $D = \mathbf{Map}(D_{IC}, s, \ldots)$ décrits au § 4.4.3.3.

     c) Ils exécutent l'agrément de clé Diffie-Hellman anonyme (voir § 9.6) sur la base des paramètres de domaine éphémères et génèrent le secret partagé $K = \mathbf{KA}(SK_{DH,IC}, PK_{DH,IFD}, D) = \mathbf{KA}(SK_{DH,IFD}, PK_{DH,IC}, D)$.

     d) Durant l'agrément de clé Diffie-Hellman, le CI et le système d'inspection DEVRAIENT vérifier que les deux clés publiques $PK_{DH,IC}$ et $PK_{DH,IFD}$ diffèrent.

     e) Ils calculent les clés de session $KS_{MAC} = \mathbf{KDF}_{\mathrm{MAC}}(K)$ et $KS_{Enc} = \mathbf{KDF}_{\mathrm{Enc}}(K)$, comme il est décrit au § 9.7.1.

     f) Ils échangent et vérifient le jeton d'authentification $T_{IFD} = \mathbf{MAC}(KS_{MAC}, PK_{DH,IC})$ et $T_{IC} = \mathbf{MAC}(KS_{MAC}, PK_{DH,IFD})$, comme il est décrit au § 4.4.3.4.

  4) À titre conditionnel, la puce du DVLM-e calcule les données d'authentification de puce CA_{IC}, les chiffre en $A_{IC} = \mathbf{E}(KS_{Enc}, CA_{IC})$ et les envoie au terminal (voir § 4.4.3.5.1). Le terminal déchiffre A_{IC} et vérifie l'authenticité de la puce en utilisant les données d'authentification de puce CA_{IC} obtenues (voir § 4.4.3.5.2).

La Figure 1 montre aussi une version simplifiée du protocole.

4.4.2 État de sécurité

Une puce de DVLM-e qui prend en charge PACE DOIT répondre aux tentatives de lecture non authentifiées [y compris la sélection de fichiers (protégés) dans la SDL] par « état de sécurité non satisfait » (0x6982).

Note.— Cette spécification est plus restrictive que la spécification correspondante pour les DVLM-e avec BAC seulement.

Si PACE est exécuté avec succès, la puce du DVLM-e a vérifié le mot de passe utilisé. La messagerie sécurisée est démarrée au moyen des clés de session calculées KS_{MAC} et KS_{Enc}.

Puce désignée par CI		***Système d'inspection désigné par IFD***
Paramètres de domaine statiques D_{IC} Choisir nonce aléatoire s Calculer z = **E**(K_π,**s**)		
	— z →	
		Calculer *s* = **D**(K_π,*z*)
	← données supplémentaires pour **Map** →	
D = **Map**(D_{IC},**s**,…) Choisir une paire de clés éphémère aléatoire ($SK_{DH,IC}$, $PK_{DH,IC}$,*D*)		*D* = **Map**(D_{IC},**s**,…) Choisir une paire de clés éphémère aléatoire ($SK_{DH,IFD}$,$PK_{DH,IFD}$,*D*)
	← $PK_{DH,IC}$, $PK_{DH,IFD}$ →	
Vérifier que $PK_{DH,IC} \neq PK_{DH,IFD}$ K = **KA**($SK_{DH,IC}$, $PK_{DH,IFD}$,*D*) Calculer T_{IC} = **MAC**(KS_{MAC},$PK_{DH,IFD}$) [calculer CA_{IC} et chiffrer en A_{IC} = **E**(KS_{Enc}, CA_{IC})]		Vérifier que $PK_{DH,IC} \neq PK_{DH,IFD}$ K = **KA**($SK_{DH,IFD}$,$PK_{DH,IC}$,*D*) Calculer T_{IFD} = **MAC**(KS_{MAC},$PK_{DH,IC}$)
	← T_{IC}, T_{IFD} → [— A_{IC} →]	
Vérifier T_{IFD}		Vérifier T_{IC} [déchiffrer A_{IC} et authentifier la puce]

Figure 1. Établissement de connexion avec authentification par mot de passe

4.4.3 Spécifications cryptographiques

La présente section décrit la spécification cryptographique en détail.

Les algorithmes particuliers sont sélectionnés par l'État émetteur ou l'organisation émettrice. Le système d'inspection DOIT prendre en charge toutes les combinaisons décrites dans les paragraphes suivants, à l'exception du mappage d'authentification de puce, qui est OPTIONNEL. La puce du DVLM-e PEUT prendre en charge plus d'une combinaison d'algorithmes.

Note.— Certains algorithmes ne sont pas disponibles pour le mappage d'authentification de puce. Pour des raisons de sécurité, l'emploi de 3DES n'est plus recommandé. Il n'y a pas de variantes DH pour réduire le nombre de variantes à appliquer par les terminaux.

4.4.3.1 DH

Pour PACE avec DH, il FAUT utiliser les algorithmes et formats indiqués au § 9.6 et au Tableau 2.

4.4.3.2 ECDH

Pour PACE avec ECDH, il FAUT utiliser les algorithmes et formats indiqués au § 9.6 et au Tableau 3.

Seules les courbes avec points non compressés DOIVENT être employées. Les paramètres de domaine normalisés décrits au § 9.5.1 DEVRAIENT être utilisés.

Tableau 2. Algorithmes et formats pour DH

OID	*Mappage*	*Chiffrement symétrique*	*Longueur de clé*	*Messagerie sécurisée*	*Jeton d'auth.*
id-PACE-DH-GM-3DES-CBC-CBC	Générique	3DES	112	CBC / CBC	CBC
id-PACE-DH-GM-AES-CBC-CMAC-128	Générique	AES	128	CBC / CMAC	CMAC
id-PACE-DH-GM-AES-CBC-CMAC-192	Générique	AES	192	CBC / CMAC	CMAC
id-PACE-DH-GM-AES-CBC-CMAC-256	Générique	AES	256	CBC / CMAC	CMAC
id-PACE-DH-IM-3DES-CBC-CBC	Intégré	3DES	112	CBC / CBC	CBC
id-PACE-DH-IM-AES-CBC-CMAC-128	Intégré	AES	128	CBC / CMAC	CMAC
id-PACE-DH-IM-AES-CBC-CMAC-192	Intégré	AES	192	CBC / CMAC	CMAC
id-PACE-DH-IM-AES-CBC-CMAC-256	Intégré	AES	256	CBC / CMAC	CMAC

Tableau 3. Algorithmes et formats pour ECDH

OID	*Mappage*	*Chiffrement symétrique*	*Longueur de clé*	*Messagerie sécurisée*	*Jeton d'auth.*
id-PACE-ECDH-GM-3DES-CBC-CBC	Générique	3DES	112	CBC / CBC	CBC
id-PACE-ECDH-GM-AES-CBC-CMAC-128	Générique	AES	128	CBC / CMAC	CMAC
id-PACE-ECDH-GM-AES-CBC-CMAC-192	Générique	AES	192	CBC / CMAC	CMAC
id-PACE-ECDH-GM-AES-CBC-CMAC-256	Générique	AES	256	CBC / CMAC	CMAC
id-PACE-ECDH-IM-3DES-CBC-CBC	Intégré	3DES	112	CBC / CBC	CBC
id-PACE-ECDH-IM-AES-CBC-CMAC-128	Intégré	AES	128	CBC / CMAC	CMAC
id-PACE-ECDH-IM-AES-CBC-CMAC-192	Intégré	AES	192	CBC / CMAC	CMAC
id-PACE-ECDH-IM-AES-CBC-CMAC-256	Intégré	AES	256	CBC / CMAC	CMAC
id-PACE-ECDH-CAM-AES-CBC-CMAC-128	Authenti-fication de la puce	AES	128	CBC / CMAC	CMAC
id-PACE-ECDH-CAM-AES-CBC-CMAC-192		AES	192	CBC / CMAC	CMAC
id-PACE-ECDH-CAM-AES-CBC-CMAC-256		AES	256	CBC / CMAC	CMAC

4.4.3.3 Chiffrement et mappage des nonces

La puce du DVLM-e DOIT sélectionner de manière aléatoire et uniforme le nonce *s* comme chaîne binaire de longueur *l*, où *l* est un multiple de la taille en bits des blocs du chiffrement par blocs **E**() respectif choisi par la puce du DVLM-e.

- Le nonce s DOIT être chiffré en mode CBC conformément à l'ISO/IEC 10116 en utilisant la clé $K_\pi = KDF_\pi(\pi)$ calculée à partir du mot de passe π et IV mis à la chaîne toute à 0.

- Le nonce s DOIT être converti en générateur aléatoire en utilisant une fonction de mappage **Map** propre à l'algorithme.

- Dans le cas du mappage intégré, le nonce supplémentaire *t* DOIT être sélectionné de manière aléatoire et uniforme comme chaîne binaire de longueur *k* et transmis en clair. Dans ce cas *k* est la taille en bits de la clé du chiffrement par blocs **E**() respectif et *l* DOIT être le plus petit multiple de la taille des blocs de **E**() de manière que $l \geq k$.

Pour mapper le nonce *s* ou les nonces *s*, *t* dans le groupe cryptographique, il FAUT employer un des mappages suivants :

- *mappage générique* (§ 4.4.3.3.1) ;

- *mappage intégré* (§ 4.4.3.3.2) ;

- *mappage d'authentification de puce* (§ 4.4.3.3.3).

4.4.3.3.1 Mappage générique

ECDH

La fonction **Map**:G → Ĝ est définie par Ĝ = s×G+H, où H en <G> est choisi de façon que $\log_G H$ est inconnu. Le point H DOIT être calculé par un agrément de clé Diffie-Hellman anonyme [TR-03111] comme suit : H = **KA**$(SK_{Map,IC}, PK_{Map,IFD}, D_{IC})$ = **KA**$(SK_{Map,IFD}, PK_{Map,IC}, D_{IC})$.

Note.— L'algorithme d'agrément de clé ECKA empêche les petites attaques de sous-groupes au moyen d'une multiplication de cofacteurs compatibles.

DH

La fonction **Map**:g → ĝ est définie par ĝ =g^s×h, où h dans <g> est choisi de façon que $\log_g h$ est inconnu. L'élément de groupe h DOIT être calculé par un agrément de clé Diffie-Hellman anonyme comme suit : h = **KA**$(SK_{Map,IC}, PK_{Map,IFD}, D_{IC})$ = **KA**$(SK_{Map,IFD}, PK_{Map,IC}, D_{IC})$.

Note.— La méthode de validation de clé publique décrite dans la norme RFC 2631 DOIT être utilisée pour empêcher de petites attaques de sous-groupes.

4.4.3.3.2 Mappage intégré

ECDH

La fonction **Map**:G → Ĝ est définie par Ĝ = f_G(**R**$_p$(s,t)), où **R**$_p$() est une fonction pseudo-aléatoire qui mappe les chaînes d'octets sur les éléments de GF(p), et f_G() est une fonction qui mappe les éléments de GF(p) sur <G>. Le nonce aléatoire t DOIT être choisi aléatoirement par le système d'inspection et transmis à la puce du DVLM-e. La fonction pseudo-aléatoire **R**$_p$() est décrite ci-dessous. La fonction f_G() est définie dans BCIMRT2010. L'Appendice B en donne une description à titre informatif.

DH

La fonction **Map**:g → ĝ est définie par ĝ = f_g(**R**$_p$(s,t)), où **R**$_p$() est une fonction pseudo-aléatoire qui mappe les chaînes d'octets sur les éléments de GF(p), et f_g() est une fonction qui mappe les éléments de GF(p) sur <g>. Le nonce aléatoire t DOIT être choisi aléatoirement par le système d'inspection et transmis à la puce du DVLM-e. La fonction pseudo-aléatoire **R**$_p$() est décrite ci-dessous. La fonction f_g() est définie par $f_g(x)=x^a \bmod p$, et a = (p-1)/q est le cofacteur. Les mises en œuvre DOIVENT vérifier que ĝ ≠ 1.

Mappage de nombres pseudo-aléatoires

La fonction **R**$_p$(s,t) est une fonction qui mappe les chaînes d'octets s (de longueur de l bits) et t (de longueur de k bits) sur un élément int($x_1||x_2||...||x_n$) mod p de GF(p). La fonction **R**$_p$(s,t) est spécifiée à la Figure 2.

La construction est fondée sur le chiffrement par blocs **E**() respectif en mode CBC conformément à l'ISO/IEC 10116 avec IV=0, où k est la taille de la clé (en bits) de **E**(). S'il y a lieu, la sortie k_i DOIT être tronquée jusqu'à la taille de clé *k*. La valeur *n* choisie DOIT être le plus petit nombre, de manière que $n*l \geq \log_2 p + 64$.

Note.— La troncature n'est nécessaire que pour AES-192 : utiliser les octets 1 à 24 de k_i ; les octets additionnels ne sont pas utilisés. Dans le cas de DES, k est considéré comme étant égal à 128 bits et la sortie de R(s,t) doit être de 128 bits.

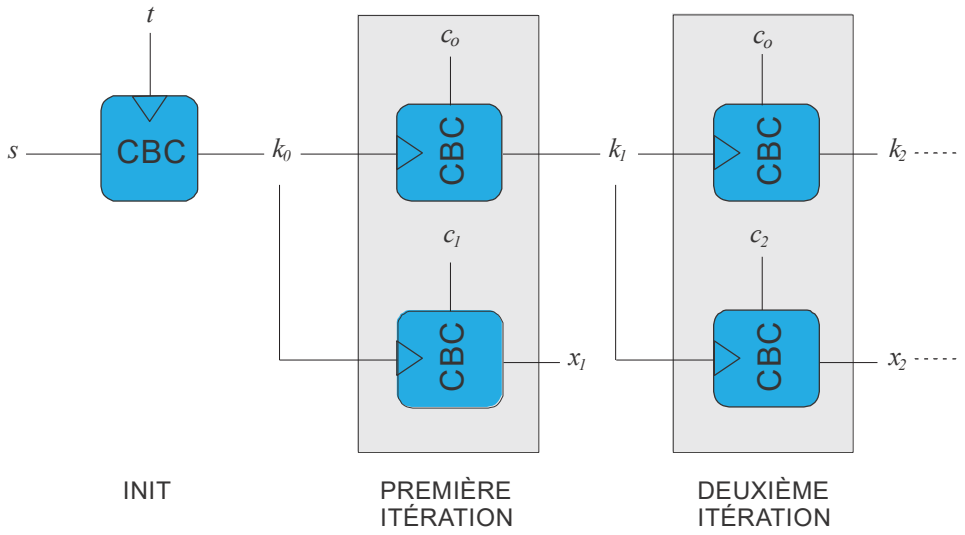

Figure 2. Mappage de nombres pseudo-aléatoires

Les constantes c_0 et c_1 sont définies comme suit :

- Pour 3DES et AES-128 (l=128) :
  - `c0=0xa668892a7c41e3ca739f40b057d85904`
  - `c1=0xa4e136ac725f738b01c1f60217c188ad`
- Pour AES-192 et AES-256 (l=256) :
  - `c0=`
    `0xd463d65234124ef7897054986dca0a174e28df758cbaa03f240616414d5a1676`
  - `c1=`
    `0x54bd7255f0aaf831bec3423fcf39d69b6cbf066677d0faae5aadd99df8e53517`

4.4.3.3.3 Mappage d'authentification de puce

La phase de mappage de PACE-CAM est identique à la phase de mappage de PACE-GM (voir § 4.4.3.3.1).

4.4.3.4 Jeton d'authentification

Le jeton d'authentification DOIT être calculé sur un objet de données à clé publique (voir § 9.4) contenant l'identificateur d'objet comme indiqué dans MSE:Set AT (voir § 4.4.4.1), et la clé publique éphémère reçue (c.-à-d. paramètres de domaine non compris — voir § 9.4.5) en utilisant le code d'authentification et la clé KS_{MAC} obtenue de l'agrément de clé.

Note.— Le remplissage est effectué intérieurement par le code d'authentification de message, c'est-à-dire qu'aucun remplissage propre à l'application n'est effectué.

3DES

Le 3DES [FIPS 46-3] DOIT être utilisé en mode « retail » (de détail) conformément à l'algorithme MAC 3 / méthode de remplissage 2 de l'ISO/IEC 9797-1 avec chiffrement par blocs DES et IV=0.

AES

L'AES [FIPS 197] DOIT être utilisé en mode CMAC [SP 800-38B] avec une longueur de MAC de 8 octets.

4.4.3.5 Données d'authentification de puce chiffrées

La puce du DVLM-e doit fournir une paire ou des paires de clés statiques SK_{IC}, PK_{IC} comme il est décrit au § 6.2. Les données d'authentification de puce chiffrées sont REQUISES pour PACE avec mappage d'authentification de puce.

4.4.3.5.1 Génération par la puce du DVLM-e

Les données d'authentification de puce DOIVENT être calculées par $CA_{IC} = (SK_{IC})^{-1} * SK_{Map,IC} \bmod p$, où SK_{IC} est la clé privée statique de la puce, $SK_{Map,\ IC}$ est la clé privée éphémère utilisée par la puce pour calculer H dans la phase de mappage de PACE (voir § 4.4.3.3.1) et p est l'ordre du groupe cryptographique utilisé. Les données d'authentification de

puce DOIVENT être chiffrées en utilisant la clé KS_{Enc} dérivée de l'agrément de clé comme A_{IC} = **E***(KS_{Enc}, CA_{IC})* pour donner les données d'authentification de puce chiffrées.

Note.— $(SK_{IC})^{-1}$ *peut être précalculé durant la personnalisation de la puce du DVLM-e et stocké de façon sécurisée dans la puce, évitant ainsi l'inverse modulaire durant l'exécution.*

4.4.3.5.2 Vérification par le terminal

Le terminal DOIT déchiffrer A_{IC} pour retrouver CA_{IC} et vérifier $PK_{Map,IC}$ = **KA***(CA_{IC}, PK_{IC}, D_{IC}),* où PK_{IC} est la clé publique statique de la puce du DVLM-e.

Note.— L'authentification passive DOIT être exécutée en combinaison avec le mappage d'authentification de puce. La puce du DVLM-e ne peut être considérée comme authentique qu'après la validation réussie de l'objet de sécurité respectif.

4.4.3.5.3 Remplissage

Le remplissage des données à chiffrer DOIT être effectué conformément à la méthode de remplissage 2 de l'ISO/IEC 9797-1.

4.4.3.5.4 AES

L'AES [19] DOIT être utilisé en mode CBC conformément à l'ISO/IEC 10116 avec $IV=E(KS_{Enc},-1)$, où *-1* est la chaîne de bits d'une longueur de 128 bits, tous les bits étant mis à 1.

4.4.4 Unités de données du protocole d'application

La suite de commandes suivantes DOIT être utilisée pour appliquer le protocole PACE :

1. MSE:Set AT
2. GENERAL AUTHENTICATE (AUTHENTIFICATION GÉNÉRALE)

4.4.4.1 MSE:Set AT

La commande MSE:Set AT est utilisée pour sélectionner et initialiser le protocole PACE. L'utilisation de MSE:Set AT pour PACE est indiquée par un identificateur d'objet PACE (voir § 4.4.3 et § 9.2.3) contenu comme référence de mécanisme cryptographique avec l'étiquette 0x80, voir le tableau ci-dessous.

Commande		
CLA		Propre au contexte
INS	0x22	Gestion de l'environnement de sécurité
P1/P2	0xC1A4	Mettre le gabarit d'authentification à authentification mutuelle.

Commande			
Données	0x80	*Référence du mécanisme cryptographique* L'identificateur d'objet du protocole à sélectionner (valeur seulement ; l'étiquette 0x06 est omise).	REQUIS
	0x83	*Référence de clé publique / clé secrète* Le mot de passe à utiliser est indiqué par les valeurs suivantes dans le présent objet de données : 0x01: Information_ZLA 0x02: CAN	REQUIS
	0x84	*Référence d'une clé privée / Référence pour le calcul d'une clé de session* Cet objet de données est REQUIS pour indiquer l'identificateur des paramètres de domaine à utiliser si les paramètres de domaine sont ambigus, c'est-à-dire s'il y a plus d'un ensemble de paramètres de domaine disponibles pour PACE.	CONDITIONNEL
	0x7F4C	*Modèle d'autorisation du titulaire de certificat* Cet objet de données (défini dans le Doc 9303-12) DOIT être présent si le terminal demande que la ou les références de l'autorité de certification à utiliser pour l'authentification du terminal soient renvoyées dans le cadre de PACE (voir § 4.4.5). L'identificateur d'objet contenu dans cet objet de données DOIT être réglé sur id-IS (voir Doc 9303-10). Les bits d'accès du modèle de données discrétionnaires DOIVENT tous être mis à 1 par le terminal.	CONDITIONNEL
Réponse			
Données	–	Absentes	
Octets d'état	0x9000	*Traitement normal* Le protocole a été sélectionné et initialisé.	
	0x6A80	*Mauvais paramètres dans le champ données de la commande* L'algorithme n'est pas pris en charge ou l'initialisation a échoué.	
	0x6A88	*Données de référence non trouvées* Les données de référence (mot de passe ou paramètre de domaine) ne sont pas disponibles.	
	Autre	*Erreur dépendant du système d'exploitation* Échec de l'initialisation du protocole.	

Note 1.— Certains systèmes d'exploitation acceptent la sélection d'une clé non disponible et n'envoient une erreur que lorsque la clé est utilisée dans le but choisi.

Note 2.— Pour la commande MSE:Set, le CI DEVRAIT ignorer les objets de données dont les étiquettes ne sont pas spécifiées pour cette commande. Pour être comprises par le CI, le terminal NE DEVRAIT PAS inclure d'objets de données dont les étiquettes ne sont pas connues.

4.4.4.2 AUTHENTIFICATION GÉNÉRALE

Une chaîne de commandes d'AUTHENTIFICATION GÉNÉRALE est utilisée pour exécuter le protocole PACE.

Commande			
CLA		Propre au contexte	
INS	0x86	AUTHENTIFICATION GÉNÉRALE	
P1/P2	0x0000	Clés et protocole connus implicitement	
Données	0x7C	*Données d'authentification dynamique* Objets de données propres au protocole	REQUIS
Réponse			
Données	0x7C	*Données d'authentification dynamique* Objets de données propres au protocole décrits au § 4.4.5.	REQUIS
Octets d'état	0x9000	*Traitement normal* Protocole (étape) exécuté(e) avec succès.	
	0x6300	*Échec de l'authentification* Échec du protocole (de l'étape).	
	0x6A80	*Mauvais paramètres dans le champ données de la commande* Les données fournies ne sont pas valides.	
	Autre	*Erreur dépendant du système d'exploitation* Échec du protocole (de l'étape).	

4.4.4.3 Chaînage des commandes

Le chaînage des commandes DOIT être utilisé pour la commande GENERAL AUTHENTICATE (authentification générale) afin de relier la séquence de commandes à l'exécution du protocole. Le chaînage des commandes NE DOIT PAS être utilisé à d'autres fins à moins que la puce ne l'indique clairement. Pour plus de renseignements sur le chaînage des commandes, voir l'ISO/IEC 7816-4.

4.4.5 Données échangées

Les objets de données propres au protocole DOIVENT être échangés dans une chaîne de commandes GENERAL AUTHENTICATE (authentification générale), la commande propre au protocole et les données de réponse étant encapsulées dans l'objet de données d'authentification dynamique (voir § 4.4.4.2) avec des étiquettes propres au contexte, comme il est indiqué au Tableau 4.

La ou les références de l'autorité de certification DOIVENT être présentes si un objet de données 0x7F4C a été transmis au CI pendant la configuration de PACE (voir § 4.4.4.1) et l'authentification du terminal est prise en charge par le CI. Dans ce cas, l'objet de données 0x87 DOIT contenir la référence la plus récente de l'autorité de certification. L'objet de données 0x88 PEUT contenir la référence de l'autorité de certification précédente.

Tableau 4. Données échangées pour PACE

Étape	***Description***	***Données de commande du protocole***		***Données de réponse du protocole***	
1.	Nonce chiffré	–	Absent[1]	0x80	Nonce chiffré
2.	Mappage de nonce	0x81	Données de mappage	0x82	Données de mappage
3.	Exécution de l'agrément de clé	0x83	Clé publique éphémère	0x84	Clé publique éphémère
4.	Authentification mutuelle	0x85	Jeton d'authentification	0x86	Jeton d'authentification
				0x87	Référence de l'autorité de certification (CONDITIONNEL)
				0x88	Référence de l'autorité de certification (CONDITIONNEL)
				0x8A	Données d'authentification de puce chiffrées (CONDITIONNEL)

Les données d'authentification de puce chiffrées (voir § 4.4.3.5) DOIVENT être présentes si le mappage d'authentification de puce est utilisé ; autrement, elles NE DOIVENT PAS être présentes.

4.4.5.1 Nonce chiffré

Le nonce chiffré (voir § 4.4.3.3) DOIT être codé sous forme d'une chaîne d'octets.

4.4.5.2 Données de mappage

Les données échangées sont propres au mappage utilisé :

4.4.5.2.1 Mappage générique

Les clés publiques éphémères (voir § 4.4.3.3 et 9.4.5) DOIVENT être codées sous forme de point de courbe elliptique (ECDH) ou d'entier non signé (DH).

4.4.5.2.2 Mappage intégré

Le nonce *t* DOIT être codé sous forme d'une chaîne d'octets.

Note.— L'objet de données propre au contexte 0x82 DOIT être vide pour le mappage intégré.

1. Ce qui signifie que l'objet de données d'authentification dynamique est vide.

4.4.5.2.3 Mappage d'authentification de puce

Le codage des données de mappage est identique à celui du mappage générique (voir § 4.4.5.2.1).

4.4.5.3 Clés publiques

Les clés publiques DOIVENT être codées comme il est décrit au § 9.4.5.

4.4.5.4 Jeton d'authentification

Le jeton d'authentification (voir § 4.4.3.4) DOIT être codé sous forme d'une chaîne d'octets.

4.4.5.5 Référence de l'autorité de certification

Les objets de données de la référence de l'autorité de certification (CAR) DOIVENT être codés comme indiqué dans le Doc 9303-12.

4.4.5.6 Données d'authentification de puce chiffrées

Les données d'authentification de puce DOIVENT être codées sous forme d'une chaîne d'octets en utilisant la fonction FE2OS() spécifiée dans TR-03111, avant le chiffrement. À noter que la fonction FE2OS() exige de coder le même nombre d'octets que l'ordre premier du groupe, c'est-à-dire en incluant éventuellement des 0x00 de tête. Les données d'authentification de puce chiffrées DOIVENT être codées sous forme d'une chaîne d'octets.

5. AUTHENTIFICATION DES DONNÉES

En plus des groupes de données de la SDL, le CI sans contact contient aussi un objet de sécurité du document (SO_D). Cet objet est signé numériquement par l'État émetteur ou l'organisation émettrice et contient des représentations de hachage du contenu de la SDL (voir le Doc 9303-10).

Un système d'inspection, contenant la clé publique de signataire de document de chaque État, ou ayant lu le certificat de signataire de document (C_{DS}) provenant du DVLM-e, sera capable de vérifier l'objet de sécurité de document (SO_D). Le contenu de la SDL est ainsi authentifié au moyen du contenu de l'objet de sécurité de document (SO_D).

Ce mécanisme de vérification n'exige pas de capacités de traitement dans le CI sans contact du DVLM-e. C'est pourquoi on parle d'une « authentification passive » du contenu du CI sans contact.

L'authentification passive prouve que le contenu de l'objet de sécurité de document (SO_D) et de la SDL est authentique et n'a pas été modifié. Elle n'empêche pas la réalisation d'une copie exacte du CI sans contact, ni la substitution du CI sans contact.

L'authentification passive DEVRAIT donc être accompagnée d'une inspection physique additionnelle du DVLM-e.

5.1 Authentification passive

5.1.1 Processus d'inspection

Le système d'inspection exécute les étapes suivantes :

1. Le système d'inspection DOIT lire l'objet de sécurité du document (SO_D) [qui DOIT contenir le certificat de signataire de document (C_{DS}) — voir aussi le Doc 9303-10] contenu dans le CI sans contact.

2. Le système d'inspection DOIT construire et valider un itinéraire de certification depuis une ancre de confiance jusqu'au certificat de signataire de document utilisé pour signer l'objet de sécurité du document (SO_D), conformément au Doc 9303-12.

3. Le système d'inspection DOIT utiliser la clé publique vérifiée de signataire de document pour vérifier la signature de l'objet de sécurité du document (SO_D).

4. Le système d'inspection PEUT lire les groupes de données pertinents dans le CI sans contact.

5. Le système d'inspection DOIT s'assurer que le contenu du groupe de données est authentique et inchangé en hachant le contenu et en comparant le résultat avec la valeur de hachage correspondante dans l'objet de sécurité de document (SO_D).

Les vérifications supplémentaires suivantes sont considérées comme une meilleure pratique :

1. Le système d'inspection ou l'agent d'inspection DEVRAIT vérifier la présence d'une extension de type de document DocumentTypeExtension dans le certificat du signataire du document :

   - si l'extension est présente, le système d'inspection DEVRAIT vérifier la cohérence entre l'extension DocumentTypeExtension, le type de document dans le groupe de données 1 et le type de document dans la ZLA visuelle (voir les Doc 9303-12, 9303-10 et 9303-3, respectivement) ;

   - si l'extension est absente, le système d'inspection DEVRAIT vérifier si KeyUsage (utilisation de la clé) du certificat du signataire du document est mis à digitalSignature (signature numérique) et que le certificat du signataire du document ne contient pas l'extension d'utilisation de clé étendue ExtendedKeyUsageExtension (voir le Doc 9303-12).

2. Le système d'inspection ou l'agent d'inspection DEVRAIT vérifier la cohérence des codes de pays dans :

   - le champ de sujet Subject-field et, si elle est présente, dans l'extension SubjectAltName du certificat du signataire du document ;

   - le champ Subject-field et, si elle est présente, dans l'extension SubjectAltName de l'ancre de confiance (certificat ACSN) ;

   - le groupe de données 1 lu dans le CI sans contact ;

   - la ZLA visuelle.

   En outre, le système d'inspection ou l'agent d'inspection PEUT comparer le contenu du groupe de données 1 avec la ZLA visuelle (voir les Doc 9303-12, 9303-10 et 9303-3, respectivement).

3. Le système d'inspection DEVRAIT vérifier que la date de délivrance du DVLM-e est incluse dans la durée d'utilisation de clé privée contenue dans le certificat du signataire du document (voir Doc 9303-12).

Les informations biométriques peuvent maintenant être utilisées pour effectuer la vérification des éléments biométriques avec la personne qui présente le DVLM-e.

5.1.2 Processus d'inspection supplémentaire pour les applications SDL2

Les données écrites après l'émission du DVLM-e ne sont pas protégées par l'objet de sécurité du document, qui est signé par l'émetteur du document. Dans le but de vérifier l'authenticité des données écrites après la délivrance, les étapes suivantes DOIVENT être effectuées par le système d'inspection pour chaque objet de données écrit :

1. Le système d'inspection DOIT construire et valider un itinéraire de certification depuis une ancre de confiance jusqu'au certificat de signataire utilisé pour signer l'objet de données, conformément au Doc 9303-12. Le système d'inspection PEUT utiliser à la fois les certificats connus à l'avance et les certificats récupérés sur la puce pour construire l'itinéraire (voir Doc 9303-10).

2. Le système d'inspection DOIT utiliser la clé publique vérifiée de signataire de document pour vérifier la signature de l'objet de sécurité.

Note.— Cette procédure peut être omise pour les objets de données dont l'authenticité n'est pas jugée pertinente pour le processus d'inspection par l'État récepteur ou l'organisation réceptrice.

6. AUTHENTIFICATION DU CI SANS CONTACT

Un État émetteur ou une organisation émettrice PEUT opter pour protéger ses DVLM-e contre la substitution de la puce.

Les mécanismes suivants peuvent être employés pour vérifier l'authenticité de la puce.

1. *Authentification active*, définie au § 6.1. La prise en charge de l'authentification active est indiquée par la présence du fichier EF.DG15. Si ce groupe est présent, le terminal PEUT lire et vérifier EF.DG15 et effectuer l'authentification active.

2. *Authentification de la puce*, définie au § 6.2. La prise en charge de l'authentification de la puce est indiquée par la présence des informations `SecurityInfos` correspondantes dans les fichiers EF.DG14/EF.CardSecurity. Si ce groupe est présent, le terminal PEUT lire et vérifier les fichiers EF.DG14/EF.CardSecurity et effectuer l'authentification de la puce.

3. *PACE avec mappage d'authentification de puce* (PACE-CAM), défini au § 4.4. La prise en charge de cet élément est indiquée par la présence de la structure `PACEInfo` correspondante dans le fichier EF.CardAccess. Si PACE-CAM est exécuté avec succès dans la procédure d'accès à la puce, le terminal PEUT procéder comme suit pour authentifier la puce :

   - lire et vérifier EF.CardSecurity ;

   - utiliser la clé publique du fichier EF.CardSecurity avec les données de mappage et les données d'authentification de puce reçues dans le cadre de PACE-CAM pour authentifier la puce (§ 4.4.3.5.2).

6.1 Authentification active

L'authentification active authentifie le CI sans contact en signant une question envoyée par l'IFD (système d'inspection) avec une clé privée connue seulement du CI.

À cette fin, le CI sans contact contient sa propre paire de clés d'authentification active (KPr_{AA} et KPu_{AA}). Une représentation hachée du groupe de données 15 [information de clé publique (KPu_{AA})] est stockée dans l'objet de sécurité de document (SO_D) et est donc authentifiée par la signature numérique de l'émetteur. La clé privée (KPr_{AA}) correspondante est stockée dans la mémoire sécurisée du CI sans contact.

En authentifiant la ZLA visuelle [par le haché de la ZLA dans l'objet de sécurité de document (SO_D)] en combinaison avec la réponse à la question, au moyen de la paire de clés d'authentification active (KPr_{AA} et KPu_{AA}) du DVLM-e, le système d'inspection vérifie que l'objet de sécurité de document (SO_D) a été lu à partir du CI sans contact authentique, stocké dans le DVLM-e authentique.

L'authentification active exige des capacités de traitement dans le CI sans contact du DVLM-e.

6.1.1 Spécification du protocole

L'authentification active est exécutée à l'aide de la commande INTERNAL AUTHENTICATE (authentification interne) de l'ISO/IEC 7816-4.

Si l'authentification active est exécutée après l'établissement de la messagerie sécurisée, toutes les commandes et réponses DOIVENT être transmises sous forme d'APDU de messagerie sécurisée, conformément au § 9.8.

De façon plus détaillée, l'IFD (système d'inspection) et le CI (CI sans contact du DVLM-e) exécutent les étapes suivantes :

1. L'IFD génère un nonce RND.IFD et l'envoie au CI en utilisant la commande INTERNAL AUTHENTICATE.

2. Le CI effectue les opérations suivantes :

   a) générer le message M ;

   b) calculer h(M) ;

   c) calculer la signature σ et envoyer la réponse à l'IFD.

3. L'IFD vérifie la réponse à la commande INTERNAL AUTHENTICATE envoyée et vérifie si le CI a envoyé la valeur correcte.

6.1.2 Spécifications cryptographiques

6.1.2.1 Nonce

L'entrée est un nonce (RND.IFD) qui DOIT être de 8 octets.

Note.— Les nonces NE DOIVENT PAS être réutilisés ; par exemple, le nonce utilisé pour BAC/PACE NE DOIT PAS être réutilisé pour l'authentification active.

6.1.2.2 RSA

Lorsqu'un mécanisme basé sur la factorisation d'entiers est utilisé, le CI DOIT calculer une signature conformément au procédé de signature numérique 1 de la norme ISO/IEC 9796-2.

Dans les paragraphes qui suivent, k désigne la longueur de la clé pour la génération de la signature et L_h la longueur de la sortie de la fonction de hachage H utilisée durant la génération de la signature. L'option 1 du champ de fin DOIT être utilisée (et t doit être mis à 1) si SHA-1 est employé durant la génération de la signature ; autrement, l'option 2 du champ de fin DOIT être utilisé (et t mis à 2).

Les valeurs suivantes pour le champ de fin DOIVENT être utilisées pour l'option 2 :

Fonction de hachage	SHA-224	SHA-256	SHA-384	SHA-512
Champ de fin	0x38CC	0x34CC	0x36CC	0x35CC

Pour des raisons d'interopérabilité, seules les fonctions de hachage SHA-1, SHA-224, SHA-256, SHA-384 et SHA-512 sont prises en charge pour l'authentification active avec RSA.

Le message M à signer DOIT être la concaténation de M1 et de M2 : M1 DOIT être un nonce de longueur c − 4 bits (RND.IC) généré par le DVLM-e, c (la *capacité de la signature*) étant donné par $c = k - L_h - (8 \times t) - 4$, et M2 est le RND.IFD généré par le système d'inspection.

Le résultat du calcul de la signature DOIT être une signature σ sans la partie non récupérable du message M2.

Les DVLM-e DEVRAIENT appliquer le mécanisme de génération de signature spécifié dans l'ISO/IEC 9796-2, § B.6, et NE DEVRAIENT PAS utiliser le mécanisme de génération de signature spécifié dans l'ISO/IEC 9796-2, § B.4. Les DVLM-e NE DOIVENT PAS mettre en œuvre d'autres mécanismes de génération de signature.

Les systèmes d'inspection DOIVENT mettre en œuvre le mécanisme de génération de signature spécifié dans l'ISO/IEC 9796-2, § B.6, et DEVRAIENT mettre en œuvre le mécanisme de génération de signature spécifié dans l'ISO/IEC 9796-2, § B.4.

6.1.2.3 ECDSA

Le format de signature en clair conforme à la norme TR-03111 DOIT être utilisé pour l'ECDSA. NE DOIVENT être employées que des courbes primaires avec des points non compressés. Un algorithme de hachage, dont la longueur de sortie est égale ou inférieure à la longueur de la clé ECDSA utilisée DOIT être employé. Seules les fonctions de hachage SHA-224, SHA-256, SHA-384 ou SHA-512 sont prises en charge. RIPEMD-160 et SHA-1 NE DOIVENT PAS être utilisés.

Le message M à signer est le nonce RND.IFD fourni par le système d'inspection.

6.1.3 Unités de données du protocole d'application

L'authentification active est exécutée par une seule invocation de la commande INTERNAL AUTHENTICATE (AUTHENTIFICATION INTERNE) comme le spécifie l'ISO/IEC 7816-4.

Commande			
CLA		Propre au contexte	
INS	0x88	INTERNAL AUTHENTICATE (AUTHENTIFICATION INTERNE)	
P1/P2	0x0000	—	
Données		*RND.IFD*	REQUIS
Réponse			
Données		Signature σ générée par le CI	REQUIS
Octets d'état	0x9000	*Traitement normal* Protocole exécuté avec succès.	
	Autre	*Erreur dépendant du système d'exploitation* Échec du protocole.	

6.1.4 Clés d'authentification active

Les paires de clés d'authentification active (KPr_{AA} et KPu_{AA}) DOIVENT être générées de façon sécurisée.

La clé publique d'authentification active (KPu_{AA}) et la clé privée d'authentification active (KPr_{AA}) sont toutes deux stockées dans le CI sans contact du DVLM-e. Après cela, aucune gestion de clé n'est possible pour ces clés.

Note.— Il convient de noter que lorsque des longueurs de clé supérieures à 1 848 bits (si la messagerie sécurisée avec 3DES est employée) / 1 792 bits (si la messagerie sécurisée avec AES est employée) sont utilisées dans l'authentification active avec messagerie sécurisée, les APDU de longueur étendue DOIVENT être prises en charge par la puce du DVLM-e et le système d'inspection.

Les États émetteurs et les organisations émettrices DOIVENT choisir des longueurs de clé appropriées, qui offrent une protection contre les attaques pendant toute la durée de vie du DVLM-e. Des catalogues cryptographiques appropriés DEVRAIENT être pris en compte.

6.1.5 Information de clé publique d'authentification active

La clé publique d'authentification active est stockée dans le groupe de données 15 de la SDL. Le format de la structure (`SubjectPublicKeyInfo`) est spécifié dans RFC 5280 (voir § 9.1). Tous les objets de sécurité DOIVENT être produits dans le format des règles de codage distinctives (DER) pour préserver l'intégrité des signatures qu'ils contiennent.

```
ActiveAuthenticationPublicKeyInfo ::= SubjectPublicKeyInfo
```

6.1.6 Processus d'inspection

Lorsqu'un DVLM-e avec le groupe de données 15 est présenté au système d'inspection, le mécanisme d'authentification active PEUT être utilisé pour s'assurer que les données sont lues à partir du CI sans contact authentique et que le CI sans contact et le document physique vont ensemble.

Le système d'inspection et le CI sans contact effectuent les étapes suivantes :

1. La ZLA est lue visuellement en entier dans le DVLM-e (si elle n'a pas déjà été lue dans le cadre de la procédure de contrôle d'accès de base) et elle est comparée à la valeur de la ZLA dans le groupe de données 1. Étant donné que l'authenticité et l'intégrité du groupe de données 1 ont été vérifiées par l'authentification passive, la similarité assure que la ZLA visuelle est authentique et non modifiée.

2. L'authentification passive a aussi prouvé l'authenticité et l'intégrité du groupe de données 15. Ce qui assure que la clé publique d'authentification active (KPu_{AA}) est authentique et non modifiée.

3. Pour s'assurer que l'objet de sécurité de document (SO_D) n'est pas une copie, le système d'inspection utilise la paire de clés d'authentification active (KPr_{AA} et KPu_{AA}) du DVLM-e dans un protocole de question-réponse avec le CI sans contact du DVLM-e, comme il est décrit plus haut.

L'exécution réussie du protocole question-réponse prouve que l'objet de sécurité de document (SO_D) appartient au document physique, que le CI sans contact est authentique et que le CI sans contact et la page de renseignements vont ensemble.

6.2 Authentification de la puce

Le protocole d'authentification de la puce est un protocole d'agrément de clé Diffie-Hellman éphémère-statique qui assure une communication sécurisée et l'authentification unilatérale de la puce du DVLM-e.

Les principales différences par rapport à l'authentification active sont :

- la sémantique de la question n'est pas autorisée parce que les transcriptions produites par ce protocole ne sont pas transférables ;

- en plus de l'authentification de la puce du DVLM-e ce protocole fournit des clés de session fortes.

La sémantique des questions est décrite en détail à l'Appendice C.

La paire ou les paires de clés d'authentification statiques de la puce DOIVENT être stockées dans la puce du DVLM-e.

- La clé privée DOIT être stockée de manière sécurisée dans la mémoire de la puce du DVLM-e.

- La clé publique DOIT être fournie comme `SubjectPublicKeyInfo` dans la structure `ChipAuthenticationPublicKeyInfo` (voir § 9.2.6).

Le protocole assure l'authentification implicite de la puce du DVLM-e lui-même et des données stockées en exécutant la messagerie sécurisée à l'aide des nouvelles clés de session.

Si le CI prend en charge l'authentification de la puce, il PEUT prendre en charge l'authentification de la puce dans le fichier principal et/ou PEUT prendre en charge l'authentification de la puce dans l'application DVLM-e. Si

l'authentification de la puce est utilisée en conjonction avec l'accès aux groupes de données dans les applications SDL2, le CI DOIT prendre en charge l'authentification de la puce dans le fichier principal.

Note.— Si la compatibilité avec le contrôle d'accès étendu de l'Union européenne TR-03110 est requise, le CI DOIT prendre en charge l'authentification de la puce dans l'application DVLM-e.

6.2.1 Spécification du protocole

Le terminal et la puce du DVLM-e exécutent les étapes suivantes :

1. La puce du DVLM-e envoie sa clé publique Diffie-Hellman statique PK_{IC} et les paramètres de domaine D_{IC} au terminal.

2. Le terminal génère une paire de clés Diffie-Hellman éphémère $(SK_{DH,IFD}, PK_{DH,IFD}, D_{IC})$ et envoie la clé publique éphémère $PK_{DH,IFD}$ à la puce du DVLM-e.

3. La puce du DVLM-e et le terminal calculent les éléments suivants :

   a) le secret partagé K = **KA**$(SK_{IC}, PK_{DH,IFD}, D_{IC})$ = **KA**$(SK_{DH,IFD}, PK_{IC}, D_{IC})$;

   b) les clés de session KS_{MAC} = **KDF**$_{MAC}(K)$ et KS_{Enc} = **KDF**$_{Enc}(K)$ dérivées de K pour la messagerie sécurisée.

La Figure 3 montre une version simplifiée.

Pour vérifier l'authenticité de PK_{IC}, le terminal DOIT exécuter l'authentification passive.

6.2.2 État de sécurité

Si l'authentification de la puce est exécutée avec succès, la messagerie sécurisée est redémarrée au moyen des clés de session KS_{MAC} et KS_{Enc} calculées. Autrement, la messagerie sécurisée se poursuit en utilisant les clés de session établies auparavant (PACE ou BAC).

Note.— L'authentification passive DOIT être exécutée en combinaison avec l'authentification de la puce. La puce du DVLM-e ne peut être considérée comme authentique qu'après la validation réussie de l'objet de sécurité correspondant.

CI (puce)		***IFD (système d'inspection)***
Paire de clés statique $(SK_{IC}, PK_{IC}, D_{IC})$		
	— PK_{IC}, D_{IC} →	
		Choisir une paire de clés éphémère aléatoire $(SK_{DH,IFD}, PK_{DH,IFD}, D_{IC})$
	← $PK_{DH,IFD}$ —	
K = **KA**$(SK_{IC}, PK_{DH,IFD}, D_{IC})$		K = **KA**$(SK_{DH,IFD}, PK_{IC}, D_{IC})$

Figure 3. Authentification de la puce

6.2.3 Spécifications cryptographiques

Les algorithmes particuliers sont sélectionnés par l'État émetteur ou l'organisation émettrice. Le système d'inspection DOIT prendre en charge toutes les combinaisons décrites dans les paragraphes suivants. La puce du DVLM-e PEUT prendre en charge plus d'une combinaison d'algorithmes.

6.2.3.1 Authentification de la puce avec DH

Pour l'authentification de la puce avec DH, les algorithmes et formats indiqués respectivement au § 9.6 et au Tableau 5 DOIVENT être employés. Pour les clés publiques, il FAUT utiliser PKCS#3 [PKCS#3] au lieu de X9.42 [X9.42].

6.2.3.2 Authentification de la puce avec ECDH

Pour l'authentification de la puce avec ECDH, les algorithmes et formats indiqués respectivement au § 9.6 et au Tableau 6 DOIVENT être employés.

Tableau 5. Identificateurs d'objet pour l'authentification de la puce avec DH

OID	*Chiffrement symétrique*	*Longueur de clé*	*Messagerie sécurisée*
id-CA-DH-3DES-CBC-CBC	3DES	112	CBC / CBC
id-CA-DH-AES-CBC-CMAC-128	AES	128	CBC / CMAC
id-CA-DH-AES-CBC-CMAC-192	AES	192	CBC / CMAC
id-CA-DH-AES-CBC-CMAC-256	AES	256	CBC / CMAC

Tableau 6. Identificateurs d'objet pour l'authentification de la puce avec ECDH

OID	*Chiffrement symétrique*	*Longueur de clé*	*Messagerie sécurisée*
id-CA-ECDH-3DES-CBC-CBC	3DES	112	CBC / CBC
id-CA-ECDH-AES-CBC-CMAC-128	AES	128	CBC / CMAC
id-CA-ECDH-AES-CBC-CMAC-192	AES	192	CBC / CMAC
id-CA-ECDH-AES-CBC-CMAC-256	AES	256	CBC / CMAC

6.2.4 Unités de données du protocole d'application

Deux mises en œuvre de l'authentification de puce sont disponibles selon l'algorithme symétrique utilisé.

- La commande suivante DOIT être utilisée pour mettre en œuvre l'authentification de la puce avec messagerie sécurisée 3DES :

  1. MSE:Set KAT

- La suite de commandes suivantes DOIT être utilisée pour mettre en œuvre l'authentification de la puce avec messagerie sécurisée AES et PEUT être employée pour mettre en œuvre l'authentification de la puce avec messagerie sécurisée 3DES :

  1. MSE:Set AT
  2. GENERAL AUTHENTICATE (AUTHENTIFICATION GÉNÉRALE)

6.2.4.1 Mise en œuvre en utilisant MSE:Set KAT

Note.— MSE:Set KAT ne peut être utilisée que pour `id-CA-DH-3DES-CBC-CBC` *et* `id-CA-ECDH-3DES-CBC-CBC`*; la messagerie sécurisée est limitée à 3DES.*

Commande			
CLA		Propre au contexte	
INS	0x22	Gestion de l'environnement de sécurité	
P1/P2	0x41A6	Régler le gabarit d'agrément de clé à calculer.	
Données	0x91	*Clé publique éphémère* Clé publique éphémère $PK_{DH,IFD}$ (voir § 9.4.5) codée à la valeur de clé publique en clair.	REQUIS
	0x84	*Référence d'une clé privée* Cet objet de données est REQUIS si la clé privée est ambiguë, c'est-à-dire s'il y a plus d'une paire de clés pour l'authentification de la puce (voir § 6.2 et 9.2.6).	CONDITIONNEL
Réponse			
Données	–	Absentes	
Octets d'état	0x9000	*Traitement normal* L'opération d'agrément de clé a été exécutée avec succès. De nouvelles clés de session ont été calculées.	
	0x6A80	*Mauvais paramètres dans le champ données de la commande* Échec de la validation de la clé publique éphémère.	
	Autre	*Erreur dépendant du système d'exploitation* Les clés de session établies auparavant demeurent valides.	

6.2.4.2 Mise en œuvre en utilisant MSE:Set AT et GENERAL AUTHENTICATE (AUTHENTIFICATION GÉNÉRALE)

1. MSE:Set AT : La commande MSE:Set AT est utilisée pour sélectionner et initialiser le protocole. L'utilisation de MSE:Set AT pour authentification de la puce est indiquée par un identificateur d'objet d'authentification de la puce (voir § 6.2.3 et § 9.2.7) contenu comme référence de mécanisme cryptographique avec l'étiquette 0x80, voir le tableau ci-dessous.

Commande			
CLA		Propre au contexte	
INS	0x22	Gestion de l'environnement de sécurité	
P1/P2	0x41A4	*Authentification de la puce* Mettre le gabarit d'authentification à authentification interne.	
Données	0x80	*Référence du mécanisme cryptographique* L'identificateur d'objet du protocole à sélectionner (valeur seulement, l'étiquette 0x06 est omise).	REQUIS
	0x84	*Référence d'une clé privée* Cet objet de données est REQUIS pour indiquer l'identificateur de clé privée à utiliser si la clé privée est ambiguë, c'est-à-dire s'il y a plus d'une clé privée pour l'authentification de la puce.	CONDITIONNEL
Réponse			
Données	–	Absentes	
Octets d'état	0x9000	*Traitement normal* Le protocole a été sélectionné et initialisé.	
	0x6A80	*Mauvais paramètres dans le champ données de la commande* L'algorithme n'est pas pris en charge ou l'initialisation a échoué.	
	0x6A88	*Données de référence non trouvées* Les données de référence (clé privée) ne sont pas disponibles.	
	Autre	*Erreur dépendant du système d'exploitation* Échec de l'initialisation du protocole.	

Note.— Certains systèmes d'exploitation acceptent la sélection d'une clé non disponible et n'envoient une erreur que lorsque la clé est utilisée dans le but choisi.

2. AUTHENTIFICATION GÉNÉRALE : La commande GENERAL AUTHENTICATE (AUTHENTIFICATION GÉNÉRALE) est employée pour exécuter l'authentification de la puce.

Commande				
CLA		Propre au contexte		
INS	0x86	AUTHENTIFICATION GÉNÉRALE		
P1/P2	0x0000	Clés et protocole connus implicitement		
Données	0x7C	*Données d'authentification dynamique* Objets de données propres au protocole		REQUIS
		0x80	Clé publique éphémère	
Réponse				
Données	0x7C	*Données d'authentification dynamique* Objets de données propres au protocole		REQUIS
Octets d'état	0x9000	*Traitement normal* Protocole (étape) exécuté(e) avec succès.		
	0x6300	*Échec de l'authentification* Échec du protocole (de l'étape).		
	0x6A80	*Mauvais paramètres dans le champ de données* Les données fournies ne sont pas valides.		
	0x6A88	*Données de référence non trouvées* Les données de référence (clé privée) ne sont pas disponibles.		
	Autre	*Erreur dépendant du système d'exploitation* Échec du protocole (de l'étape).		

Note.— Les clés publiques pour l'authentification de la puce prises en charge par la puce sont disponibles dans l'objet de sécurité (voir § 9.2.11). S'il prend en charge plus d'une clé publique, le terminal DOIT sélectionner la clé privée correspondante de la puce à utiliser dans MSE:Set AT.

6.2.4.3 Clé publique éphémère

Les clés publiques éphémères (voir § 9.4.5) DOIVENT être codées sous forme de point de courbe elliptique (ECDH) ou d'entier non signé (DH).

7. MÉCANISMES DE CONTRÔLE D'ACCÈS SUPPLÉMENTAIRES

Les données personnelles stockées dans le CI sans contact qui, par définition, constituent le minimum obligatoire pour l'interopérabilité mondiale sont la ZLA et l'image stockée numériquement du visage du titulaire. L'une et l'autre peuvent aussi être vues (lues) visuellement après l'ouverture du DVLM-e et sa présentation pour inspection.

Outre l'image du visage stockée numériquement comme élément biométrique principal pour l'interopérabilité mondiale, l'OACI a entériné l'emploi d'images stockées numériquement des doigts et/ou des iris, en plus du visage. Pour un usage

national ou bilatéral, les États PEUVENT choisir de stocker des gabarits et/ou PEUVENT choisir de limiter l'accès ou de chiffrer ces données, comme ils le décideront eux-mêmes.

L'accès à ces données personnelles plus sensibles DEVRAIT être plus restreint. Ce qui peut être accompli de deux façons : contrôle d'accès étendu ou chiffrement des données. Le § 7.1 spécifie l'authentification du terminal comme un mécanisme interopérable pour le contrôle d'accès étendu. Si aucune interopérabilité n'est requise, d'autres mécanismes peuvent être utilisés.

7.1 Authentification du terminal

Le mécanisme d'authentification du terminal est CONDITIONNEL. La mise en œuvre est REQUISE pour les applications SDL2. L'authentification du terminal PEUT être utilisée pour protéger les données biométriques secondaires dans l'application DVLM-e.

Le protocole d'authentification du terminal est un protocole question-réponse à deux mouvements qui permet une authentification unilatérale explicite du terminal. Le protocole est basé sur le contrôle d'accès étendu spécifié dans [TR-03110]. Si ce protocole est pris en charge par le CI, il DOIT prendre en charge l'authentification de la puce ou PACE avec mappage d'authentification de puce.

Ce protocole permet au CI de vérifier que le terminal est autorisé à accéder aux données sensibles. Comme le terminal peut accéder à des données sensibles par la suite, toute communication ultérieure DOIT être protégée de manière appropriée. L'authentification du terminal permet donc également d'authentifier une clé publique éphémère choisie par le terminal qui a été utilisée pour configurer la messagerie sécurisée avec authentification de la puce ou PACE avec authentification de la puce. Le CI DOIT lier les droits d'accès au terminal à la messagerie sécurisée établie par la clé publique éphémère authentifiée du terminal.

Le CI PEUT prendre en charge l'authentification du terminal dans le fichier principal et/ou l'application DVLM-e. Si l'authentification du terminal est utilisée pour protéger des groupes de données dans d'autres applications que l'application DVLM-e, le CI DOIT prendre en charge l'authentification du terminal dans le fichier principal.

Note.— Si la compatibilité avec le contrôle d'accès étendu de l'Union européenne [TR-03110] est requise, le CI DOIT prendre en charge l'authentification du terminal dans l'application DVLM-e.

7.1.2 Spécification du protocole

Le terminal et le CI exécutent les étapes suivantes :

1. Le terminal envoie une chaîne de certificats au CI. La chaîne commence par un certificat vérifiable avec la clé publique CVCA stockée sur la puce et se termine par le certificat du terminal.

2. Le CI vérifie les certificats et extrait la clé publique du terminal PK_{IFD}.

3. Le CI choisit aléatoirement une question r_{IC} et l'envoie au terminal.

4. Le terminal répond avec la signature s_{IFD} = **Sign***(SK_{IFD}, $ID_{IC}||r_{IC}||$***Comp***$[PK_{DH,IFD})]$.*

5. Le CI vérifie que **Verify***(PK_{IFD} , s_{IFD}, ID_{IC} $||r_{IC}||$***Comp***(PKDH,IFD)) = true*.

Note.— La clé $PK_{DH,IFD}$ est générée pendant l'authentification de la puce ou le mappage PACE avec authentification de puce. Si plus d'une clé est générée (par ex. l'authentification de la puce est effectuée après PACE avec mappage de l'authentification de puce), la clé la plus récente DOIT être utilisée.

Dans ce protocole, ID_{IC} est un identificateur du CI :

- Si le BAC est utilisé, ID_{IC} est le numéro de document du DVLM-e contenu dans la ZLA, y compris le chiffre de contrôle.
- Si PACE est utilisé, ID_{IC} est calculé à l'aide de la clé publique PACE éphémère du CI, c'est à dire ID_{IC} = **Comp***($PK_{DH,IC}$)*.

Note.— Une exécution réussie du protocole PACE est REQUISE avant que l'authentification du terminal puisse être effectuée dans le fichier principal.

Une version simplifiée est présentée ci-dessous à la Figure 4.

7.1.3 État de sécurité

Si l'authentification du terminal a été effectuée avec succès, le CI DOIT accorder l'accès aux données sensibles stockées en fonction de l'autorisation effective du terminal authentifié. Si l'autorisation effective n'accorde de droits d'accès à aucune donnée dans une application SDL2, la sélection de cette application DOIT être rejetée par le CI.

Le CI DOIT cependant restreindre les droits d'accès du terminal à la messagerie sécurisée établie par la clé publique éphémère authentifiée, c'est-à-dire la clé publique éphémère fournie par le terminal dans le cadre de l'authentification de la puce ou de PACE avec mappage d'authentification de puce. Le CI NE DOIT PAS accepter plus d'une exécution de l'authentification du terminal au cours de la même session (voir § 9.8.1 et § 9.8.3 sur la définition de « session »).

Note 1.— Les droits d'accès sont valables tant que la messagerie sécurisée établie par les clés publiques éphémères authentifiées est active ; par conséquent, l'état de sécurité n'est pas affecté par la sélection ou la désélection d'applications.

Note 2.— La messagerie sécurisée n'est pas affectée par l'authentification du terminal. La puce DVLM-e DOIT conserver la messagerie sécurisée même si l'authentification du terminal échoue (sauf si une erreur de messagerie sécurisée se produit).

CI (puce)		**IFD (système d'inspection)**
Choisir r_{IC} aléatoirement		
	— r_{IC} →	
		s_{IFD} = **Sign**(SK_{IFD},ID_{IC}||r_{IC}||**Comp***($PK_{DH,IFD}$)*)
	← s_{IFD} —-	
Verify*(PK_{IFD} , s_{IFD}, ID_{IC}||r_{IC}||**Comp***($PK_{DH,IFD}$)) = true*		

Figure 4. Authentification du terminal

Tableau 7. Identificateurs d'objets pour l'authentification des terminaux avec RSA

OID	Signature	Hachage	Paramètres
id-TA-RSA-PSS-SHA-256	RSASSA-PSS	SHA-256	par défaut
id-TA-RSA-PSS-SHA-512	RSASSA-PSS	SHA-512	par défaut

7.1.4 Spécifications cryptographiques

7.1.4.1 Authentification du terminal avec RSA

Pour l'authentification du terminal avec RSA, les algorithmes et formats suivants DOIVENT être utilisés.

7.1.4.1.1 Algorithme de signature

RSA [RFC-3447], [PKCS#1] comme spécifié dans le Tableau 7 DOIT être utilisé.

Les paramètres par défaut à utiliser avec RSA-PSS sont définis comme suit :

- Algorithme de hachage : L'algorithme de hachage est sélectionné selon le Tableau 7.
- Algorithme de génération de masques : MGF1 [RFC-3447], [PKCS#1] en utilisant l'algorithme de hachage sélectionné.
- Longueur du sel : Longueur en octets de la sortie de l'algorithme de hachage sélectionné.
- Champ de fin : 0xBC

7.1.4.1.2 Format de clé publique

Le format TLV [ISO/IEC 7816-8] décrit dans le Doc 9303-12 DOIT être utilisé.

- L'identificateur d'objet DOIT être tiré du Tableau 7.
- La longueur binaire du module DOIT être de 2 048 ou de 3 072.
- La longueur binaire de l'exposant DOIT être de 32 au maximum.

7.1.4.1.3 Compression des clés publiques

La clé publique éphémère comprimée du terminal **Comp**($PK_{DH,IFD}$) est définie comme le hachage SHA-1 de la valeur publique DH, c'est-à-dire une chaîne d'octets de longueur fixe 20.

7.1.4.2 Authentification du terminal avec ECDSA

Pour l'authentification du terminal avec ECDSA, les algorithmes et formats suivants DOIVENT être utilisés.

Tableau 8. Identificateurs d'objets pour l'authentification des terminaux avec ECDSA

OID	Signature	Hachage
id-TA-ECDSA-SHA-224	ECDSA	SHA-224
id-TA-ECDSA-SHA-256	ECDSA	SHA-256
id-TA-ECDSA-SHA-384	ECDSA	SHA-384
id-TA-ECDSA-SHA-512	ECDSA	SHA-512

7.1.4.2.1 Algorithme de signature

ECDSA avec le format de signature en clair [TR-03111] comme spécifié dans le Tableau 8 DOIT être utilisé.

7.1.4.2.2 Format de clé publique

Le format TLV [ISO/IEC 7816-8] décrit dans le Doc 9303-12 DOIT être utilisé.

- L'identificateur d'objet DOIT être tiré du Tableau 8.
- La longueur binaire de la courbe DOIT être de 224, 256, 320, 384 ou 512.
- Les paramètres de domaine DOIVENT être conformes à [TR-03111].

7.1.4.2.3 Compression des clés publiques

La clé publique éphémère comprimée du terminal **Comp**($PK_{DH,IFD}$) est définie comme la coordonnée x du point public ECDH, c'est-à-dire une chaîne d'octets de longueur fixe [log256p].

7.1.4.3 Validation du certificat

Pour valider un certificat de terminal, le CI DOIT recevoir une chaîne de certificats commençant à un point de confiance stocké sur le CI. Ces points de confiance sont des clés publiques plus ou moins récentes de la CVCA du CI.

7.1.4.3.1 État initial du ou des points de confiance du CI

Le ou les points de confiance initiaux DOIVENT être stockés de manière sécurisée dans la mémoire du CI lors de la phase de production ou de (pré-)personnalisation.

L'agent de (pré-)personnalisation DOIT :

- fixer la date actuelle du CI à la date de la (pré-)personnalisation ;
- personnaliser la clé CVCA avec la date de prise d'effet la plus récente comme point de confiance.

L'agent de (pré-)personnalisation PEUT en outre personnaliser la clé CVCA précédente comme point de confiance.

7.1.4.3.2 Certificats de liaison

Comme la paire de clés utilisée par la CVCA change au fil du temps, des certificats de liaison CVCA doivent être établis. Les certificats de liaison CVCA DOIVENT être signés avec la clé CVCA précédente, c'est-à-dire la clé CVCA avec la date de prise d'effet la plus récente. Le CI est TENU de mettre à jour en interne son ou ses points de confiance en fonction des certificats de liaison valides reçus.

Le CI DOIT être capable de stocker jusqu'à deux points de confiance.

Note.— En raison de l'ordonnancement des certificats de liaison CVCA (voir Doc 9303-12), il faut stocker au maximum deux points de confiance sur le CI.

7.1.4.3.3 Date actuelle

Le CI DOIT accepter les certificats de liaison CVCA expirés mais NE DOIT PAS accepter les certificats DV et de terminaux expirés. Pour déterminer si un certificat est expiré, le CI DOIT utiliser sa date actuelle.

Date actuelle : Si le CI n'a pas d'horloge interne, la date actuelle du CI DOIT être approximée comme décrit ci-après. La date est approximée de manière autonome par le CI en utilisant la date de prise d'effet du certificat le plus récent contenu dans un certificat de liaison CVCA valide, un certificat DV ou un certificat de terminal exact.

Certificat terminal exact : un certificat de terminal est exact si le vérificateur de documents (DV) émetteur a la confiance du CI pour établir des certificats de terminal avec la date correcte de prise d'effet du certificat. Les certificats de liaison CVCA, les certificats DV et les certificats de terminal délivrés par un DV national DOIVENT être considérés comme exacts par le CI. Les autres certificats NE DOIVENT PAS être considérés comme exacts.

Un terminal PEUT envoyer des certificats de liaison CVCA, des certificats DV et des certificats de terminal à un CI pour mettre à jour la date actuelle et le point de confiance stocké sur le CI même si le terminal n'a pas l'intention ou n'est pas en mesure de poursuivre l'authentification du terminal.

Note.— Le CI vérifie uniquement qu'un certificat est apparemment récent (c.-à-d. par rapport à la date actuelle approximative), sauf si le CI contient une horloge interne.

7.1.4.3.4 Procédure de validation générale

La procédure de validation du certificat se déroule en trois étapes :

1. **Vérification du certificat :** La signature DOIT être valide et, sauf si le certificat est un certificat de liaison CVCA, le certificat NE DOIT PAS être expiré. Si la vérification échoue, la procédure DOIT être interrompue.

   Note.— Le cas d'un certificat de liaison CVCA expiré ne peut se produire que si le CI a une source de temps au-delà de la date actuelle approximative décrite ci-dessus.

2. **Mise à jour du statut interne :** La date actuelle DOIT être *mise à jour*, la clé publique et les attributs (y compris les extensions de certificat pertinentes) DOIVENT être importés, les nouveaux points de confiance DOIVENT être *activés*, les points de confiance expirés DOIVENT être *désactivés* pour la vérification des certificats DV.

3. **Nettoyage :** La puce DOIT fournir au maximum deux points de confiance activés par application. Si plus de deux points de confiance pour une application restent activés après la mise à jour du statut interne, le point de confiance dont la date de prise d'effet est la moins récente DOIT être *désactivé*.

L'opération de *mise à jour* de la date actuelle et les opérations d'*activation* et de *désactivation* d'un point de confiance DOIVENT être mises en œuvre en tant qu'opération atomique.

Activation d'un point de confiance : le nouveau point de confiance DOIT être ajouté à la liste des points de confiance.

Désactivation d'un point de confiance : les points de confiance expirés NE DOIVENT PAS être utilisés pour la vérification des certificats DV. Dans le cas de CI dont la date actuelle peut être avancée au-delà de la date d'expiration d'un point de confiance, par exemple les CI utilisant une horloge interne, les points de confiance expirés DOIVENT rester utilisables pour la vérification des certificats de liaison CVCA. Les points de confiance désactivés PEUVENT être supprimés après l'importation réussie du certificat de liaison successif.

7.1.4.3.5 Exemple de procédure de validation

La procédure de validation suivante, fournie à titre d'exemple, PEUT être utilisée pour valider une chaîne de certificats. Pour chaque certificat reçu, le CI effectue les étapes suivantes :

1. Le CI vérifie la signature du certificat. Si la signature est incorrecte, la vérification échoue.

2. Si le certificat n'est pas un certificat de liaison CVCA, la date d'expiration du certificat est comparée à la date actuelle du CI. Si la date d'expiration est antérieure à la date actuelle, la vérification échoue.

3. Le certificat est accepté comme valide, et la clé publique et les attributs (y compris les extensions de certificat pertinentes) contenus dans le certificat sont importés.

    - Pour les certificats de terminal CVCA, DV et exacts : La date de prise d'effet du certificat est comparée à la date actuelle du CI. Si la date actuelle est antérieure à la date de prise d'effet, la date actuelle est mise à jour à la date de prise d'effet.

    - Pour les certificats de liaison CVCA : La nouvelle clé publique CVCA est ajoutée à la liste des points de confiance stockés de manière sécurisée dans la mémoire du CI. Le nouveau point de confiance est alors activé.

    - Pour les certificats de terminal et DV : La nouvelle clé publique du DV ou du terminal est importée temporairement pour la vérification ultérieure du certificat ou l'authentification du terminal, respectivement.

4. Les points de confiance expirés stockés en toute sécurité dans la mémoire du CI sont désactivés pour la vérification des certificats DV et peuvent être supprimés de la liste des points de confiance.

7.1.4.3.6 Autorisation effective

Chaque certificat DOIT contenir un gabarit d'autorisation du titulaire du certificat (voir Doc 9303-12) et PEUT contenir des extensions d'autorisation (voir Doc 9303-12, § 7.2.2.6).

- Le gabarit d'autorisation du titulaire du certificat identifie le type de terminal (cette spécification ne considère que les systèmes d'inspection, mais d'autres spécifications peuvent utiliser des types de terminaux différents).

- Le gabarit d'autorisation du titulaire de certificat et les extensions d'autorisation déterminent l'*autorisation relative* du titulaire de certificat attribuée par l'autorité de certification émettrice.

Pour déterminer l'*autorisation effective* d'un titulaire de certificat, le CI DOIT calculer un « et » booléen binaire de l'autorisation relative contenue dans le certificat du terminal, le certificat DV de référence et le certificat CVCA de référence.

L'autorisation effective DOIT être interprétée par le CI comme suit :

- Le rôle effectif est un CVCA :
  - Ce certificat de liaison a été délivré par la CVCA nationale.
  - Le CI DOIT mettre à jour son point de confiance interne, c'est-à-dire la clé publique et l'autorisation effective.
  - L'émetteur du certificat est une source de temps fiable et le CI DOIT mettre à jour sa date actuelle en utilisant la date effective du certificat.
  - Le CI NE DOIT PAS accorder à la CVCA l'accès à des données sensibles (c'est-à-dire que l'autorisation effective DEVRAIT être ignorée).
- Le rôle effectif est un DV :
  - Le certificat a été délivré par la CVCA nationale pour un DV autorisé.
  - L'émetteur du certificat est une source de temps fiable et le CI DOIT mettre à jour sa date actuelle en utilisant la date effective du certificat.
  - Le CI NE DOIT PAS accorder à un DV l'accès à des données sensibles (c'est-à-dire que l'autorisation effective DEVRAIT être ignorée).
- Le rôle effectif est un terminal :
  - Le certificat a été délivré par un DV national ou étranger.
  - Si le certificat est un certificat de terminal exact (voir § 7.1.4.3.3), l'émetteur du certificat est une source de temps fiable et le CI DOIT mettre à jour sa date actuelle en utilisant la date effective du certificat.
  - Le CI DOIT accorder au terminal authentifié l'accès aux données sensibles conformément à l'autorisation effective.

Note.— Le modèle d'autorisation du titulaire de certificat et les extensions d'autorisation peuvent contenir des bits non attribués à un droit d'accès (bits RFU). Le CI DOIT ignorer ces bits lors de l'évaluation des droits d'accès.

7.1.4.3.7 Importation de clés publiques

Les clés publiques importées par la procédure de validation des certificats sont soit *stockées de manière permanente, soit temporairement* sur le CI.

Le CI DEVRAIT refuser d'importer une clé publique si la référence du titulaire du certificat est déjà connue du CI.

Importation permanente : les clés publiques contenues dans les certificats de liaison CVCA DOIVENT être importées de manière permanente par le CI et DOIVENT être stockées de manière sécurisée dans la mémoire du CI. Une clé publique importée de façon permanente et ses métadonnées DOIVENT remplir les conditions suivantes :

- Elle PEUT être écrasée après expiration par une clé publique ultérieure importée de façon permanente.
- Elle DOIT être écrasée par une clé publique importée de façon permanente avec la même référence de titulaire de certificat ou l'importation DOIT être rejetée.
- Elle NE DOIT PAS être écrasée par une clé publique importée temporairement.

L'activation et la désactivation d'une clé publique importée de façon permanente DOIT être une opération atomique.

Importation temporaire : les clés publiques contenues dans les certificats DV et de terminal DOIVENT être importées temporairement par le CI. Une clé publique importée temporairement et ses métadonnées DOIVENT remplir les conditions suivantes :

- Elle NE DOIT PAS être sélectionnable ou utilisable après une mise hors tension du CI.
- Elle DOIT rester utilisable jusqu'à ce que l'opération cryptographique suivante soit menée à bien (c.-à-d. PSO:Verify Certificate ou External Authenticate).
- Elle PEUT être écrasée par une clé publique ultérieure importée de façon temporaire.

Un terminal NE DOIT PAS utiliser d'autre clé publique importée temporairement que la plus récente.

Métadonnées importées : Pour chaque clé publique importée de façon permanente ou temporaire, les données supplémentaires suivantes contenues dans le certificat (voir le Doc 9303-12) DOIVENT être stockées :

- Référence du titulaire du certificat
- Autorisation du titulaire du certificat (rôle effectif et autorisation effective)
- Date d'entrée en vigueur du certificat
- Date d'expiration du certificat
- Extensions de certificat (le cas échéant)

Le calcul du rôle effectif (CVCA, DV ou terminal) et de l'autorisation effective du titulaire du certificat est décrit au § 7.1.4.3.6.

Note.— Le format des données stockées dépend du système d'exploitation et n'entre pas dans le cadre de cette spécification.

7.1.5 Unités de données du protocole d'application

La séquence de commandes suivante DOIT être utilisée avec la messagerie sécurisée pour mettre en œuvre l'authentification du terminal :

- MSE:Set DST
- PSO:Verify Certificate
- MSE:Set AT
- Get Challenge
- External Authenticate

Les étapes 1 et 2 sont répétées pour chaque certificat CV à vérifier (certificats de liaison CVCA, certificat DV, certificat de terminal).

7.1.5.1 MSE:Set DST

La commande MSE:Set DST est utilisée pour configurer la vérification des certificats.

Commande			
CLA		Propre au contexte	
INS	0x22	Gestion de l'environnement de sécurité	
P1/P2	0x81B6	Définir le gabarit de signature numérique pour la vérification.	
Données	0x83	*Référence d'une clé publique* Nom codé ISO 8859-1 de la clé publique à définir	REQUIS
Réponse			
Données	–	Absentes	
Octets d'état	0x9000 0x6A88 Autre	*Opération normale* La clé a été choisie en fonction de l'objectif visé. *Données de référence non trouvées* La sélection a échoué car la clé publique n'est pas disponible. *Erreur dépendant du système d'exploitation* La clé n'a pas été sélectionnée.	

Note.— Certains systèmes d'exploitation acceptent la sélection d'une clé non disponible et n'envoient une erreur que lorsque la clé est utilisée dans le but choisi.

7.1.5.2 PSO:Verify Certificate

La commande PSO:Verify Certificate est utilisée pour vérifier et importer des certificats.

Commande			
CLA		Propre au contexte	
INS	0x2A	Effectuer une opération de sécurité	
P1/P2	0x00BE	Vérifier le certificat auto-descriptif.	
Données	0x7F4E 0x5F37	*Corps du certificat* *Le corps du certificat à vérifier.* *Signature* *La signature du certificat à vérifier.*	REQUIS REQUIS
Réponse			
Données	–	Absentes	
Octets d'état	0x9000 Autre	*Traitement normal* Le certificat a été validé avec succès et la clé publique a été importée. *Erreur dépendant du système d'exploitation* La clé publique n'a pas pu être importée (par exemple, le certificat n'a pas été accepté).	

7.1.5.3 MSE:Set AT

L'utilisation de MSE:Set AT pour l'authentification du terminal est indiquée par la valeur 0x81A4 de P1/P2, voir le tableau ci-dessous.

Commande			
CLA		Propre au contexte	
INS	0x22	Gestion de l'environnement de sécurité	
P1/P2	0x81A4	Authentification du terminal :	
Données	0x83	*Référence de clé publique / clé secrète* Cet objet de données est utilisé pour sélectionner la clé publique du terminal par son nom codé ISO 8859-1.	REQUIS
Réponse			
Données	–	Absentes	
Octets d'état	0x9000 0x6A80 0x6A88	*Traitement normal* *Le protocole a été sélectionné et initialisé.* *Mauvais paramètres dans le champ données de la commande* L'algorithme n'est pas pris en charge ou l'initialisation a échoué. *Données de référence non trouvées* Les données de référence ne sont pas disponibles.	

	Autre	*Erreur dépendant du système d'exploitation* Échec de l'initialisation du protocole.

Note.— Certains systèmes d'exploitation acceptent la sélection d'une clé non disponible et n'envoient une erreur que lorsque la clé est utilisée dans le but choisi.

7.1.5.4 Get Challenge

Commande			
CLA		Propre au contexte	
INS	0x84	Get Challenge	
P1/P2	0x0000		
Données	–	Absentes	
Le	0x08		REQUIS
Réponse			
Données	r_{IC}	8 octets d'aléatoire.	
Octets d'état	0x9000 Autre	*Traitement normal* *Erreur dépendant du système d'exploitation*	

7.1.5.5 External Authenticate

Commande			
CLA		Propre au contexte	
INS	0x82	External Authenticate	
P1/P2	0x0000	Clés et algorithmes connus implicitement	
Données		Signature générée par le terminal.	REQUIS
Réponse			
Données	–	Absentes	
Octets d'état	0x9000 0x6300 0x6982 Autre	*Traitement normal* L'authentification a réussi. L'accès aux groupes de données sera accordé en fonction de l'autorisation effective du certificat vérifié correspondant. *Avertissement* La vérification de la signature a échoué. *Statut de sécurité non satisfait* L'authentification a échoué car le niveau d'authentification actuel du terminal ne permet pas d'utiliser l'authentification du terminal (par ex. l'authentification du terminal a déjà été effectuée, etc.). *Erreur dépendant du système d'exploitation* L'authentification a échoué.	

7.2 Chiffrement des éléments biométriques supplémentaires

On PEUT aussi restreindre l'accès aux éléments biométriques supplémentaires en les chiffrant. Pour que les données chiffrées puissent être déchiffrées, le système d'inspection DOIT être doté d'une clé de déchiffrement. La définition de l'algorithme de chiffrement/déchiffrement et des clés à utiliser relève de l'État de mise en œuvre et n'entre pas dans le cadre du présent document.

La mise en œuvre de la protection des éléments biométriques supplémentaires dépend des spécifications internes de l'État ou des spécifications convenues bilatéralement entre États qui partagent cette information.

8. SYSTÈME D'INSPECTION

Pour qu'il puisse prendre en charge la fonctionnalité requise et les options définies qui peuvent être implémentées dans les DVLM-e qui lui seront présentés, le système d'inspection doit répondre à certaines conditions préalables.

8.1 Contrôle d'accès de base

Les systèmes d'inspection qui prennent en charge le contrôle d'accès de base DOIVENT remplir les conditions préalables suivantes :

1. Le système d'inspection est équipé d'un moyen d'acquérir la ZLA à partir du document physique pour calculer les clés d'accès de base au document (K_{Enc} et K_{MAC}) à partir du DVLM-e.

2. Le logiciel du système d'inspection prend en charge le protocole décrit au § 4.3, si un DVLM-e avec contrôle d'accès de base lui est présenté, y compris le chiffrement du canal de communication avec la messagerie sécurisée.

8.2 Établissement de connexion avec authentification par mot de passe (PACE)

Les systèmes d'inspection qui prennent en charge le protocole PACE DOIVENT remplir les conditions préalables suivantes :

1. Le système d'inspection est équipé d'un moyen d'acquérir la ZLA et/ou le CAN à partir du document physique.

2. Le logiciel du système d'inspection prend en charge le protocole décrit au § 4.4, si un DVLM-e avec protocole PACE lui est présenté, y compris le chiffrement du canal de communication avec la messagerie sécurisée.

8.3 Authentification passive

Pour pouvoir réaliser une authentification passive des données stockées dans le CI sans contact du DVLM-e, le système d'inspection doit avoir connaissance d'informations clés des États émetteurs ou des organisations émettrices :

1. Pour chaque État émetteur ou organisation émettrice, le certificat d'AC signataire nationale ou les informations pertinentes extraites du certificat DOIVENT être stockés de manière sécurisée dans le système d'inspection.

2. Autrement, pour chaque État émetteur ou organisation émettrice, les certificats de signataires de documents (C_{DS}) ou les informations pertinentes extraites des certificats DOIVENT être stockés de manière sécurisée dans le système d'inspection.

Avant d'utiliser une clé publique d'AC signataire nationale d'un État émetteur ou d'une organisation émettrice, l'État récepteur ou l'organisation réceptrice DOIT établir la confiance en cette clé.

Avant d'utiliser un certificat de signataire de document (C_{DS}) pour la vérification d'un SO_D, le système d'inspection DOIT vérifier sa signature numérique en utilisant la clé publique d'AC signataire nationale.

En outre, les systèmes d'inspection DOIVENT avoir accès aux informations de révocation vérifiées.

8.4 Authentification active

La prise en charge de l'authentification active par les systèmes d'inspection est OPTIONNELLE.

Si le système d'inspection prend en charge l'authentification active, il est OBLIGATOIRE que le système d'inspection ait la capacité de lire la ZLA visuelle.

Si le système d'inspection prend en charge l'authentification active, son logiciel DOIT prendre en charge le protocole d'authentification active décrit au § 6.1.

8.5 Authentification de la puce

La prise en charge de l'authentification de la puce par les systèmes d'inspection est OPTIONNELLE.

Si le système d'inspection prend en charge l'authentification de la puce, il DOIT pouvoir lire la ZLA visuelle.

Si le système d'inspection prend en charge l'authentification de la puce, son logiciel DOIT prendre en charge le protocole d'authentification de la puce décrit au § 6.2.

8.6 Authentification du terminal

La prise en charge de l'authentification du terminal par les systèmes d'inspection est OPTIONNELLE.

Si le système d'inspection prend en charge l'authentification du terminal, il est OBLIGATOIRE que le système d'inspection ait la capacité de stocker de manière sécurisée la clé privée du système d'inspection. Le système d'inspection DOIT avoir accès à son DV à intervalles réguliers pour renouveler le certificat du terminal.

Si le système d'inspection prend en charge l'authentification du terminal, le logiciel du système d'inspection DOIT prendre en charge le protocole d'authentification du terminal tel que décrit dans le § 7.1.

8.7 Déchiffrement des éléments biométriques supplémentaires

La mise en œuvre de la protection des éléments biométriques supplémentaires optionnels dépend des spécifications internes de l'État ou des spécifications convenues bilatéralement entre États qui partagent cette information.

9. SPÉCIFICATIONS COMMUNES

9.1 Structures ASN.1

Les structures de données `SubjectPublicKeyInfo` et `AlgorithmIdentifier` se définissent comme suit :

```
SubjectPublicKeyInfo ::= SEQUENCE {
  algorithm  AlgorithmIdentifier,
  subjectPublicKey    BIT STRING
}

AlgorithmIdentifier ::= SEQUENCE {
  algorithm OBJECT IDENTIFIER,
  parameters ANY DEFINED BY algorithm OPTIONAL
}
```

On trouvera des renseignements détaillés sur les `paramètres` dans X9.42 et TR-03111.

9.2 Informations sur les protocoles et les applications pris en charge

La structure des données ASN.1 `SecurityInfos` DOIT être fournie par la puce du DVLM-e pour indiquer les protocoles de sécurité pris en charge. La structure des données est spécifiée comme suit :

```
SecurityInfos  ::= SET OF SecurityInfo

  SecurityInfo ::= SEQUENCE {
  protocol  OBJECT IDENTIFIER,
  requiredData ANY DEFINED BY protocol,
  optionalData ANY DEFINED BY protocol OPTIONAL
}
```

Les éléments contenus dans une structure de données `SecurityInfo` ont la signification suivante :

- l'identificateur d'objet `protocol` indique le protocole pris en charge ;
- la structure `requiredData` de type ouvert contient les données obligatoires propres au protocole ;
- la structure `optionalData` de type ouvert contient les données optionnelles propres au protocole.

Informations de sécurité pour PACE

Pour indiquer la prise en charge de PACE `SecurityInfos` peut contenir les éléments suivants :

- au moins un élément `PACEInfo` utilisant un paramètre de domaine normalisé DOIT être présent ;
- pour chaque ensemble de paramètres de domaine explicites pris en charge, un élément `PACEDomainParameterInfo` DOIT être présent.

Informations de sécurité pour l'authentification active

Si un algorithme de signature ECDSA est utilisé pour l'authentification active par la puce du DVLM-e, `SecurityInfos` DOIT contenir l'élément `SecurityInfo` suivant :

- `ActiveAuthenticationInfo`

Informations de sécurité pour l'authentification de la puce

Pour indiquer la prise en charge de l'authentification de la puce `SecurityInfos` peut contenir les entrées suivantes :

- au moins un élément `ChipAuthenticationInfo` et l'élément `ChipAuthenticationPublicKeyInfo` correspondant utilisant les paramètres de domaine explicites DOIVENT être présents.

Informations de sécurité pour l'authentification du terminal

Pour indiquer la prise en charge de l'authentification du terminal, `SecurityInfos` peut contenir l'entrée suivante :

- au moins un `TerminalAuthenticationInfo` DOIT être présent.

Informations de sécurité pour les applications présentes

Le § 3.11.2 du Doc 9303-10 recommande la présence d'un fichier élémentaire transparent EF.DIR pour indiquer les applications prises en charge. Ce fichier est obligatoire si une application SDL2 est présente. Étant donné que le fichier EF.DIR n'est pas signé et qu'il peut donc être manipulé, par exemple pour cacher des applications existantes à l'IFD, une copie sécurisée du fichier EF.DIR est fournie comme `SecurityInfo` si une application SDL2 est présente.

Informations de sécurité pour d'autres protocoles

`SecurityInfos` PEUT contenir des entrées additionnelles indiquant la prise en charge d'autres protocoles ou fournissant d'autres informations. Le système d'inspection PEUT rejeter toute entrée inconnue.

9.2.1 PACEInfo

Cette structure de données fournit des informations détaillées sur la mise en œuvre de PACE.

- L'identificateur d'objet `protocol` DOIT identifier les algorithmes à utiliser (c'est-à-dire agrément de clé, chiffrement symétrique et MAC).

- L'entier `version` DOIT indiquer la version du protocole. Seule la version 2 est prise en charge par la présente spécification.

- L'entier `parameterId` est utilisé pour indiquer l'identificateur de paramètre de domaine. Il DOIT être employé si la puce du DVLM-e utilise des paramètres de domaine normalisés (voir § 9.5.1), si elle fournit plusieurs paramètres de domaine explicites pour PACE ou si `protocol` est un des OID `*-CAM-*`. Dans le cas de PACE avec mappage d'authentification de puce, `parameterID` indique aussi l'ID de la clé d'authentification de la puce utilisée, c'est-à-dire que la puce DOIT fournir un élément `ChipAuthenticationPublicKeyInfo` avec l'ID `keyID` égal à l'ID `parameterID` de cette structure de données.

```
PACEInfo ::= SEQUENCE {
    protocol    OBJECT IDENTIFIER(
                id-PACE-DH-GM-3DES-CBC-CBC |
                id-PACE-DH-GM-AES-CBC-CMAC-128 |
                id-PACE-DH-GM-AES-CBC-CMAC-192 |
                id-PACE-DH-GM-AES-CBC-CMAC-256 |
                id-PACE-ECDH-GM-3DES-CBC-CBC |
                id-PACE-ECDH-GM-AES-CBC-CMAC-128 |
                id-PACE-ECDH-GM-AES-CBC-CMAC-192 |
                id-PACE-ECDH-GM-AES-CBC-CMAC-256 |
                id-PACE-DH-IM-3DES-CBC-CBC |
                id-PACE-DH-IM-AES-CBC-CMAC-128 |
                id-PACE-DH-IM-AES-CBC-CMAC-192 |
                id-PACE-DH-IM-AES-CBC-CMAC-256 |
                id-PACE-ECDH-IM-3DES-CBC-CBC |
                id-PACE-ECDH-IM-AES-CBC-CMAC-128 |
                id-PACE-ECDH-IM-AES-CBC-CMAC-192 |
                id-PACE-ECDH-IM-AES-CBC-CMAC-256 |
                id-PACE-ECDH-CAM-AES-CBC-CMAC-128 |
                id-PACE-ECDH-CAM-AES-CBC-CMAC-192 |
                id-PACE-ECDH-CAM-AES-CBC-CMAC-256),
    version     INTEGER, -- MUST be 2
    parameterId INTEGER OPTIONAL
}
```

9.2.2 PACEDomainParameterInfo

Cette structure de données est REQUISE si la puce du DVLM-e fournit des paramètres de domaine explicites pour PACE ; autrement, elle DOIT être omise.

- L'identificateur d'objet `protocol` DOIT identifier le type de paramètres de domaine (DH ou ECDH).

- La séquence `domainParameter` DOIT contenir les paramètres de domaine.
- L'entier `parameterId` PEUT être utilisé pour indiquer l'identificateur de paramètre de domaine local. Il DOIT être utilisé si la puce du DVLM-e fournit plusieurs paramètres de domaine explicites pour PACE.

```
PACEDomainParameterInfo ::= SEQUENCE {
  protocol          OBJECT IDENTIFIER(
                    id-PACE-DH-GM |
                    id-PACE-ECDH-GM |
                    id-PACE-DH-IM |
                    id-PACE-ECDH-IM |
                    id-PACE-ECDH-CAM),
  domainParameter   AlgorithmIdentifier,
  parameterId       INTEGER OPTIONAL
}
```

Note.— La puce du DVLM-e PEUT prendre en charge plus d'un ensemble de paramètres de domaine explicites (c'est-à-dire que la puce peut prendre en charge des algorithmes et/ou des longueurs de clés différents). Dans ce cas l'identificateur DOIT être divulgué dans l'élément `PACEDomainParameterInfo` *correspondant.*

Les paramètres de domaine contenus dans `PACEDomainParameterInfo` ne sont pas protégés et peuvent ne pas être sécurisés. L'emploi de paramètres de domaine non sécurisés pour PACE révélera le mot de passe utilisé. Les puces des DVLM-e DOIVENT prendre en charge au moins un ensemble de paramètres de domaine normalisés, comme il est spécifié au § 9.5.1. Les systèmes d'inspection NE DOIVENT PAS utiliser les paramètres de domaine explicites fournis par la puce du DVLM-e à moins qu'ils ne sachent explicitement que ces paramètres de domaine sont sécurisés.

Les clés publiques éphémères DOIVENT être échangées comme des valeurs de clés publiques en clair. § 9.4.5. contient plus de renseignements sur le codage.

9.2.3 Identificateur d'objet de PACE

Les identificateurs d'objet utilisés pour PACE sont contenus dans le sous-arbre de `bsi-de` :

```
bsi-de OBJECT IDENTIFIER ::= {
    itu-t(0) identified-organization(4) etsi(0)
    reserved(127) etsi-identified-organization(0) 7
}
```

L'identificateur d'objet suivant DOIT être utilisé :

```
id-PACE OBJECT IDENTIFIER ::= {
    bsi-de protocols(2) smartcard(2) 4
}

id-PACE-DH-GM                    OBJECT IDENTIFIER ::= {id-PACE 1}
id-PACE-DH-GM-3DES-CBC-CBC       OBJECT IDENTIFIER ::= {id-PACE-DH-GM 1}
id-PACE-DH-GM-AES-CBC-CMAC-128   OBJECT IDENTIFIER ::= {id-PACE-DH-GM 2}
id-PACE-DH-GM-AES-CBC-CMAC-192   OBJECT IDENTIFIER ::= {id-PACE-DH-GM 3}
id-PACE-DH-GM-AES-CBC-CMAC-256   OBJECT IDENTIFIER ::= {id-PACE-DH-GM 4}



id-PACE-ECDH-GM                  OBJECT IDENTIFIER ::= {id-PACE 2}
id-PACE-ECDH-GM-3DES-CBC-CBC     OBJECT IDENTIFIER ::= {id-PACE-ECDH-GM 1}
id-PACE-ECDH-GM-AES-CBC-CMAC-128 OBJECT IDENTIFIER ::= {id-PACE-ECDH-GM 2}
id-PACE-ECDH-GM-AES-CBC-CMAC-192 OBJECT IDENTIFIER ::= {id-PACE-ECDH-GM 3}
id-PACE-ECDH-GM-AES-CBC-CMAC-256 OBJECT IDENTIFIER ::= {id-PACE-ECDH-GM 4}

id-PACE-DH-IM                    OBJECT IDENTIFIER ::= {id-PACE 3}
id-PACE-DH-IM-3DES-CBC-CBC       OBJECT IDENTIFIER ::= {id-PACE-DH-IM 1}
id-PACE-DH-IM-AES-CBC-CMAC-128   OBJECT IDENTIFIER ::= {id-PACE-DH-IM 2}
id-PACE-DH-IM-AES-CBC-CMAC-192   OBJECT IDENTIFIER ::= {id-PACE-DH-IM 3}
id-PACE-DH-IM-AES-CBC-CMAC-256   OBJECT IDENTIFIER ::= {id-PACE-DH-IM 4}

id-PACE-ECDH-IM                  OBJECT IDENTIFIER ::= {id-PACE 4}
id-PACE-ECDH-IM-3DES-CBC-CBC     OBJECT IDENTIFIER ::= {id-PACE-ECDH-IM 1}
id-PACE-ECDH-IM-AES-CBC-CMAC-128 OBJECT IDENTIFIER ::= {id-PACE-ECDH-IM 2}
id-PACE-ECDH-IM-AES-CBC-CMAC-192 OBJECT IDENTIFIER ::= {id-PACE-ECDH-IM 3}
id-PACE-ECDH-IM-AES-CBC-CMAC-256 OBJECT IDENTIFIER ::= {id-PACE-ECDH-IM 4}
```

```
id-PACE-ECDH-CAM                   OBJECT IDENTIFIER ::= {id-PACE 6}
id-PACE-ECDH-CAM-AES-CBC-CMAC-128 OBJECT IDENTIFIER ::= {id-PACE-ECDH-CAM 2}
id-PACE-ECDH-CAM-AES-CBC-CMAC-192 OBJECT IDENTIFIER ::= {id-PACE-ECDH-CAM 3}
id-PACE-ECDH-CAM-AES-CBC-CMAC-256 OBJECT IDENTIFIER ::= {id-PACE-ECDH-CAM 4}
```

9.2.4 ActiveAuthenticationInfo

Si la puce du DVLM-e utilise l'algorithme de signature ECDSA pour l'authentification active, l'élément `SecurityInfos` du groupe de données 14 de la SDL de la puce du DVLM-e DOIT contenir l'entrée `SecurityInfo` suivante :

```
ActiveAuthenticationInfo ::= SEQUENCE {
  protocol           OBJECT IDENTIFIER(id-icao-mrtd-security-aaProtocolObject),
  version            INTEGER, -- MUST be 1
  signatureAlgorithm OBJECT IDENTIFIER
}

id-icao-mrtd-security-aaProtocolObject OBJECT IDENTIFIER ::=
                                   { id-icao-mrtd-security 5 }
```

Pour `signatureAlgorithm`, les identificateurs d'objet définis dans TR-03111 DOIVENT être employés.

Note.— L'identificateur d'objet `id-icao-mrtd-security` est défini dans le Doc 9303-10.

9.2.5 ChipAuthenticationInfo

Cette structure de données donne des informations détaillées sur l'implémentation de l'authentification de la puce.

- L'identificateur d'objet `protocol` DOIT identifier les algorithmes à utiliser (c'est-à-dire agrément de clé, chiffrement symétrique et MAC).
- L'entier version DOIT indiquer la version du protocole. Seule la version 1 est prise en charge par la présente spécification.
- L'entier `keyId` PEUT être utilisé pour indiquer l'identificateur de clé locale. Il DOIT être utilisé si la puce du DVLM-e fournit plusieurs clés publiques pour l'authentification de la puce.

```
ChipAuthenticationInfo ::= SEQUENCE {
  protocol   OBJECT IDENTIFIER(
             id-CA-DH-3DES-CBC-CBC |
             id-CA-DH-AES-CBC-CMAC-128 |
             id-CA-DH-AES-CBC-CMAC-192 |
             id-CA-DH-AES-CBC-CMAC-256 |
             id-CA-ECDH-3DES-CBC-CBC |
             id-CA-ECDH-AES-CBC-CMAC-128 |
             id-CA-ECDH-AES-CBC-CMAC-192 |
             id-CA-ECDH-AES-CBC-CMAC-256),
  version    INTEGER, -- MUST be 1
  keyId      INTEGER OPTIONAL
}
```

9.2.6 ChipAuthenticationPublicKeyInfo

Cette structure de données fournit une clé publique pour l'authentification de la puce ou PACE avec mappage d'authentification de puce du DVLM-e.

- L'identificateur d'objet `protocol` DOIT identifier le type de clé publique (c'est-à-dire DH ou ECDH).
- La séquence `chipAuthenticationPublicKey` DOIT contenir la clé publique sous forme codée.
- L'entier `keyId` PEUT être utilisé pour indiquer l'identificateur de clé locale. Il DOIT être employé si la puce du DVLM-e fournit plusieurs clés publiques pour l'authentification de la puce ou si cette clé est utilisée pour PACE avec mappage d'authentification de puce.

```
ChipAuthenticationPublicKeyInfo ::= SEQUENCE {
    protocol                    OBJECT IDENTIFIER(id-PK-DH | id-PK-ECDH),
    chipAuthenticationPublicKey SubjectPublicKeyInfo,
    keyId                       INTEGER OPTIONAL
}
```

Note.— La puce du DVLM-e PEUT prendre en charge plus d'une paire de clés d'authentification de la puce (c'est-à-dire que la puce peut prendre en charge plusieurs algorithmes et/ou longueurs de clé). Dans ce cas, l'identificateur de clé locale DOIT être divulgué dans l'élément `ChipAuthenticationInfo` et l'élément `ChipAuthenticationPublicKeyInfo` correspondants.

9.2.7 Identificateur d'objet d'authentification de la puce

L'identificateur d'objet suivant DOIT être utilisé :

```
id-PK OBJECT IDENTIFIER ::= {
  bsi-de protocols(2) smartcard(2) 1
}

id-PK-DH                    OBJECT IDENTIFIER ::= {id-PK 1}
id-PK-ECDH                  OBJECT IDENTIFIER ::= {id-PK 2}

id-CA OBJECT IDENTIFIER ::= {
  bsi-de protocols(2) smartcard(2) 3
}

id-CA-DH                    OBJECT IDENTIFIER ::= {id-CA 1}
id-CA-DH-3DES-CBC-CBC       OBJECT IDENTIFIER ::= {id-CA-DH 1}
id-CA-DH-AES-CBC-CMAC-128   OBJECT IDENTIFIER ::= {id-CA-DH 2}
id-CA-DH-AES-CBC-CMAC-192   OBJECT IDENTIFIER ::= {id-CA-DH 3}
id-CA-DH-AES-CBC-CMAC-256   OBJECT IDENTIFIER ::= {id-CA-DH 4}

id-CA-ECDH                  OBJECT IDENTIFIER ::= {id-CA 2}
id-CA-ECDH-3DES-CBC-CBC     OBJECT IDENTIFIER ::= {id-CA-ECDH 1}
id-CA-ECDH-AES-CBC-CMAC-128 OBJECT IDENTIFIER ::= {id-CA-ECDH 2}
id-CA-ECDH-AES-CBC-CMAC-192 OBJECT IDENTIFIER ::= {id-CA-ECDH 3}
id-CA-ECDH-AES-CBC-CMAC-256 OBJECT IDENTIFIER ::= {id-CA-ECDH 4}
```

9.2.8 TerminalAuthenticationInfo

Cette structure de données fournit des informations détaillées sur la mise en œuvre de l'authentification du terminal.

- L'identificateur d'objet `protocol` DOIT identifier le protocole *general* d'authentification du terminal car le protocole spécifique peut changer au fil du temps.
- L'entier `version` DOIT indiquer la version du protocole. Seule la version 1 est prise en charge par la présente spécification. Notez que les versions ultérieures de TR-03110 définissent la version 2 de ce protocole, qui n'entre pas dans le cadre de la présente spécification.

```
TerminalAuthenticationInfo ::= SEQUENCE {
   protocol OBJECT IDENTIFIER(id-TA),
   version  INTEGER        -- MUST be 1
  }
```

9.2.9 Identificateur d'objet d'authentification du terminal

L'identificateur d'objet suivant DOIT être utilisé :

```
id-TA OBJECT IDENTIFIER ::= {
  bsi-de protocols(2) smartcard(2) 2
}

id-TA-RSA               OBJECT IDENTIFIER ::= {id-TA 1}
id-TA-RSA-PSS-SHA-256   OBJECT IDENTIFIER ::= {id-TA-RSA 4}
id-TA-RSA-PSS-SHA-512   OBJECT IDENTIFIER ::= {id-TA-RSA 6}

id-TA-RSA               OBJECT IDENTIFIER ::= {id-TA 2}
id-TA-ECDSA-SHA-224     OBJECT IDENTIFIER ::= {id-TA-ECDSA 2}
id-TA-ECDSA-SHA-256     OBJECT IDENTIFIER ::= {id-TA-ECDSA 3}
id-TA-ECDSA-SHA-384     OBJECT IDENTIFIER ::= {id-TA-ECDSA 4}
id-TA-ECDSA-SHA-512     OBJECT IDENTIFIER ::= {id-TA-ECDSA 5}
```

9.2.10 EFDIRInfo

Cette structure de données encapsule une copie complète du contenu du fichier élémentaire transparent EF.DIR contenu dans le fichier principal.

```
  EFDIRInfo ::= SEQUENCE {
       protocol                        OBJECT IDENTIFIER(id-EFDIR),
       eFDIR                           OCTET STRING
  }

id-EFDIR OBJECT IDENTIFIER ::={
  id-icao-mrtd-security 13
}
```

9.2.11 Stockage dans la puce

Pour indiquer la prise en charge des protocoles et des paramètres admis, la puce du DVLM-e DOIT fournir les informations `SecurityInfos` dans des fichiers élémentaires transparents (on trouvera la structure générique de ces fichiers dans le Doc 9303-10) :

- Le fichier EF.CardAccess contenu dans le fichier principal est REQUIS si la puce du DVLM-e prend en charge PACE et il DOIT contenir les informations `SecurityInfos` requises pour PACE :
  - `PACEInfo`
  - `PACEDomainParameterInfo`
- Le fichier EF.CardSecurity contenu dans le fichier principal est REQUIS si :
  - la puce du DVLM-e prend en charge PACE avec mappage d'authentification de puce, ou
  - la puce du DVLM-e prend en charge l'authentification du terminal dans le fichier principal, ou
  - l'authentification de la puce dans le fichier principal est prise en charge par le DVLM-e

  et DOIT contenir les informations `SecurityInfos` suivantes :
  - `ChipAuthenticationInfo` requise pour l'authentification de la puce
  - `ChipAuthenticationPublicKeyInfo` requise pour PACE-CAM/authentification de la puce
  - `TermninalAuthenticationInfo` requise pour l'authentification du terminal
  - `EFDIRInfo` si plus que l'application DVLM-e est présente sur la puce
  - les informations `SecurityInfos` contenues dans EF.CardAccess.
- Le fichier EF.DG14 contenu dans l'application DVLM-e est REQUIS si :
- PACE avec mappage générique/intégré est supporté par la puce du DVLM-e
- L'authentification du terminal dans l'application DVLM-e est prise en charge par la puce du DVLM-e, ou
- L'authentification de la puce dans l'application DVLM-e est prise en charge par la puce du DVLM-e.

et DOIT contenir les informations `SecurityInfos` suivantes :

- `ChipAuthenticationInfo` requise pour l'authentification de la puce
- `ChipAuthenticationPublicKeyInfo` requise pour l'authentification de la puce
- `TermninalAuthenticationInfo` requise pour l'authentification du terminal
- les informations `SecurityInfos` contenues dans EF.CardAccess

- L'ensemble complet des informations `SecurityInfos` (y compris les informations `SecurityInfos` du fichier EF.CardAccess non spécifiées dans le Doc 9303) DOIVENT aussi être stockées dans le fichier EF.DG14 de l'application DVLM-e (voir le Doc 9303-10).

Les fichiers PEUVENT contenir des informations `SecurityInfos` additionnelles qui n'entrent pas dans le cadre de la présente spécification.

Note.— Même si l'authenticité des informations `SecurityInfos` *contenues dans les fichiers EF.DG14 et EF.CardSecurity est protégée par authentification passive, le fichier EF.CardAccess n'est pas protégé.*

9.3 APDU

9.3.1 Longueur étendue

Selon la taille des objets cryptographiques (p. ex. clés publiques, signatures), les APDU avec des champs de longueur étendue DOIVENT être employées pour envoyer ces données à la puce du DVLM-e. Pour plus de renseignements sur la longueur étendue, voir la norme ISO/IEC 7816-4.

9.3.1.1 Puces de DVLM-e

Pour les puces de DVLM-e, la prise en charge de la longueur étendue est CONDITIONNELLE. Si les algorithmes cryptographiques et les tailles de clés choisies par l'État émetteur exigent l'emploi de la longueur étendue, les puces de DVLM-e DOIVENT prendre en charge la longueur étendue. Si la puce du DVLM-e prend en charge la longueur étendue, cette prise en charge doit être indiquée dans l'ATR/ATS ou dans le fichier EF.ATR/INFO, comme il est spécifié dans l'ISO/IEC 7816-4.

9.3.1.2 Terminaux

Dans le cas des terminaux, la prise en charge de la longueur étendue est REQUISE. Un terminal DEVRAIT examiner si la prise en charge de la longueur étendue est indiquée dans l'ATR/ATS ou dans le fichier EF.ATR/INFO de la puce du DVLM-e avant d'utiliser cette option. Le terminal NE DOIT PAS utiliser la longueur étendue pour des APDU autres que les commandes suivantes à moins que la taille exacte des tampons d'entrée et de sortie de la puce du DVLM-e ne soit explicitement indiquée dans l'ATR/ATS ou l'EF.ATR/INFO :

- MSE:Set KAT
- GENERAL AUTHENTICATE

9.3.2 Chaînage des commandes

Le chaînage des commandes DOIT être utilisé pour la commande GENERAL AUTHENTICATE (AUTHENTIFICATION GÉNÉRALE) afin de relier la séquence de commandes à l'exécution du protocole PACE. Le chaînage des commandes NE DOIT PAS être employé à d'autres fins à moins que la puce ne l'indique clairement. Pour plus de renseignements sur le chaînage des commandes, voir l'ISO/IEC 7816-4.

9.3.3 Objets de données

L'expéditeur d'une commande ou d'une réponse APDU DOIT transmettre les objets de données dans le champ données dans l'ordre défini dans les descriptions des APDU.

Note.— Accepter les objets de données dans n'importe quel ordre n'est pas nécessaire, mais améliore l'interopérabilité pour certaines commandes, par exemple pour MSE:Set AT/GENERAL AUTHENTICATE. Cependant, il faut faire attention dans le cas de commandes telles que PSO:Verify Certificate, où l'ordre est fixé pour des raisons cryptographiques.

9.4 Objets de données de clé publique

Un objet de données de clé publique est une structure TLV BER de type constructeur contenant un identificateur d'objet et plusieurs objets de données propres au contexte imbriqués dans le gabarit 0x7F49 de clé publique du titulaire de la carte.

- L'identificateur d'objet est propre à l'application et se rapporte non seulement au format de la clé publique (c'est-à-dire les objets de données propres au contexte), mais aussi à son utilisation.

- Les objets de données propres au contexte sont définis par l'identificateur d'objet et contiennent la valeur de la clé publique et les paramètres de domaine.

Le format des objets de données des clés publiques employé dans la présente spécification est décrit plus bas.

9.4.1 Codage de l'objet de données

Un entier non signé DOIT être converti en chaîne d'octets en utilisant la représentation binaire de l'entier en format gros-boutiste. Le nombre minimal d'octets DOIT être utilisé, c'est-à-dire que les octets de tête de valeur 0x00 NE DOIVENT PAS être utilisés.

Pour coder les points de courbe elliptique, il FAUT utiliser un codage sans compression conforme à TR-03111.

9.4.2 Clés publiques RSA

Les objets de données contenus dans une clé publique RSA sont indiqués dans le Tableau 9. L'ordre des objets de données est fixe.

Tableau 9. Clé publique RSA

Objet de données	Notation	Étiquette	Type	CV Certificate
Identificateur d'objet		0x06	Identificateur d'objet	m
Module composite	*n*	0x81	Entier non signé	m
Exposant public	*e*	0x82	Entier non signé	m

9.4.3 Clés publiques Diffie-Hellman

Les objets de données contenus dans une clé publique DH sont indiqués dans le Tableau 10. L'ordre des objets de données est fixe.

Tableau 10. Objets de données des clés publiques DH

Objet de données	*Notation*	*Étiquette*	*Type*
Identificateur d'objet		0x06	Identificateur d'objet
Module principal	p	0x81	Entier non signé
Ordre du sous-groupe	q	0x82	Entier non signé
Générateur	g	0x83	Entier non signé
Valeur publique	y	0x84	Entier non signé

Note.— Le codage des composants de clés sous forme d'entiers non signés signifie que chacun d'eux est codé sur le plus petit nombre d'octets possible, c'est-à-dire sans qu'ils soient précédés par des octets mis à 0x00. En particulier, une clé publique DH peut être codée sur un nombre d'octets inférieur au nombre d'octets de l'entier.

9.4.4 Clés publiques à courbe elliptique

Les objets de données contenus dans une clé publique EC sont indiqués dans le Tableau 11. L'ordre des objets de données est fixe ; les paramètres de domaine CONDITIONNELS DOIVENT tous être présents, sauf le cofacteur, ou tous être absents, comme suit :

Tableau 11. Objets de données des clés publiques ECDH

Objet de données	*Notation*	*Étiquette*	*Type*
Identificateur d'objet		0x06	Identificateur d'objet
Module principal	p	0x81	Entier non signé
Premier coefficient	a	0x82	Entier non signé
Deuxième coefficient	b	0x83	Entier non signé
Point de base	G	0x84	Point de courbe elliptique
Ordre du point de base	r	0x85	Entier non signé
Point public	Y	0x86	Point de courbe elliptique
Cofacteur	f	0x87	Entier non signé

9.4.5 Clés publiques éphémères

Dans le cas des clés publiques éphémères, le format et les paramètres de domaine sont déjà connus. Par conséquent, seule la valeur de clé publique en clair, c'est-à-dire la valeur publique y pour les clés publiques Diffie-Hellman et le point public Y pour les clés publiques à courbe elliptique, est utilisée pour véhiculer la clé publique éphémère dans un objet de données propre au contexte.

Note.— La validation des clés publiques éphémères est RECOMMANDÉE. Dans le cas de DH, l'algorithme de validation exige que la puce du DVLM-e ait une connaissance plus détaillée des paramètres de domaine (c'est-à-dire l'ordre du sous-groupe utilisé) que ce que fournit habituellement PKCS#3.

9.5 Paramètres de domaine

À l'exception des paramètres de domaine contenus dans `PACEInfo`, tous les paramètres de domaine DOIVENT être fournis par `AlgorithmIdentifier` (voir § 9.1).

Dans `PACEInfo`, l'ID des paramètres de domaine normalisés décrits au Tableau 12 DOIT être désigné directement. Les paramètres de domaine explicites fournis par `PACEDomainParameterInfo` NE DOIVENT PAS utiliser les ID réservés aux paramètres de domaine normalisés.

9.5.1 Paramètres de domaine normalisés

Les ID des paramètres de domaine normalisés décrits dans le tableau ci-dessous DEVRAIENT être utilisés. Les paramètres de domaine explicites NE DOIVENT PAS utiliser les ID réservés aux paramètres de domaine normalisés.

L'identificateur d'objet suivant DEVRAIT être utilisé pour désigner les paramètres de domaine normalisés dans un `AlgorithmIdentifier` (voir § 9.1) :

```
standardizedDomainParameters OBJECT IDENTIFIER ::= {
  bsi-de algorithms(1) 2
}
```

Dans un `AlgorithmIdentifier`, cet identificateur d'objet DOIT désigner l'ID du paramètre de domaine normalisé indiqué dans le Tableau 12 comme entier `INTEGER`, figurant comme `parameters` dans `AlgorithmIdentifier`.

Tableau 12. Paramètres de domaine normalisés

ID	*Nom*	*Taille (bits)*	*Type*	*Référence*
0	Groupe MODP de 1 024 bits avec un sous-groupe d'ordre premier de 160 bits	1 024/160	GFP	[RFC 5114]
1	Groupe MODP de 2 048 bits avec un sous-groupe d'ordre premier de 224 bits	2 048/224	GFP	[RFC 5114]
2	Groupe MODP de 2 048 bits avec un sous-groupe d'ordre premier de 256 bits	2 048/256	GFP	[RFC 5114]
3-7	RFU			

ID	*Nom*	*Taille (bits)*	*Type*	*Référence*
8	NIST P-192 (secp192r1)	192	ECP	[RFC 5114], [FIPS 186-4]
9	BrainpoolP192r1	192	ECP	[RFC 5639]
10	NIST P-224 (secp224r1)*	224	ECP	[RFC 5114], [FIPS 186-4]
11	BrainpoolP224r1	224	ECP	[RFC 5639]
12	NIST P-256 (secp256r1)	256	ECP	[RFC 5114], [FIPS 186-4]
13	BrainpoolP256r1	256	ECP	[RFC 5639]
14	BrainpoolP320r1	320	ECP	[RFC 5639]
15	NIST P-384 (secp384r1)	384	ECP	[RFC 5114], [FIPS 186-4]
16	BrainpoolP384r1	384	ECP	[RFC 5639]
17	BrainpoolP512r1	512	ECP	[RFC 5639]
18	NIST P-521 (secp521r1)	521	ECP	[RFC 5114], [FIPS 186-4]
19-31	RFU			

* Cette courbe ne peut pas être utilisée avec le mappage intégré.

9.5.2 Paramètres de domaine explicites

L'identificateur d'objet `dhpublicnumber` ou `ecPublicKey` pour DH ou ECDH, respectivement, DOIT être utilisé pour désigner les paramètres de domaine explicites dans un `AlgorithmIdentifier` (voir § 9.1) :

```
dhpublicnumber OBJECT IDENTIFIER ::= {
  iso(1) member-body(2) us(840) ansi-x942(10046) number-type(2) 1
}

ecPublicKey OBJECT IDENTIFIER ::= {
  iso(1) member-body(2) us(840) ansi-x962(10045) keyType(2) 1
}
```

Dans le cas des courbes elliptiques, les paramètres de domaine DOIVENT être décrits explicitement dans la structure `ECParameters` figurant comme `parameters` dans `AlgorithmIdentifier`, c'est-à-dire que les courbes nommées et les paramètres de domaine implicites NE DOIVENT PAS être utilisés.

9.6 Algorithmes d'agrément de clé

La présente spécification prend en charge l'agrément de clé Diffie-Hellman et Diffie-Hellman à courbe elliptique, comme il est indiqué dans le tableau suivant :

Tableau 13. Algorithmes d'agrément de clé

Algorithme / Format	***DH***	***ECDH***
Algorithme d'agrément de clé	[PKCS#3]	ECKA [TR-03111]
Format de clé publique X.509	[X9.42]	[TR-03111]
Format de clé publique TLV	TLV, voir § 9.4.3	TLV, voir § 9.4.4
Validation de clé publique éphémère	[RFC 2631]	[TR-03111]

9.7 Mécanisme de calcul de clé

9.7.1 Fonction de calcul de clé

La fonction de calcul de clé **KDF**(K,c) se définit comme suit :

Entrée : Les entrées suivantes sont requises :

- la valeur du secret partagé K (REQUIS) ;
- un compteur d'entier gros-boutiste c de 32 bits (REQUIS).

Sortie : Données de clé (keydata) en chaîne d'octets.

Actions : Les actions suivantes sont exécutées :

- keydata = **H**(K || c) ;
- keydata de sortie en chaîne d'octets.

La fonction de calcul de clé **KDF**(K,c) exige une fonction de hachage appropriée désignée par **H**(), c'est-à-dire que la longueur en bits de la fonction de hachage DOIT être supérieure ou égale à la longueur en bits de la clé calculée. La valeur de hachage DOIT être interprétée comme sortie en octets gros-boutiste.

Note.— Le secret partagé K est défini comme une chaîne d'octets. Si le secret partagé est généré avec ECKA [TR-03111], la coordonnée x du point généré DOIT être utilisée.

9.7.1.1 3DES

Pour calculer des clés 3DES [FIPS 46-3] de 128 bits (112 bits en excluant les bits de parité), la fonction de hachage SHA-1 [FIPS 180-4] DOIT être utilisée et les étapes supplémentaires suivantes DOIVENT être exécutées :

- utiliser les octets 1 à 8 de keydata pour former keydataA et les octets 9 à 16 de keydata pour former keydataB ; les octets additionnels ne sont pas utilisés ;

- ajuster les bits de parité de keydataA et de keydataB pour former des clés DES correctes (OPTIONNEL).

9.7.1.2 AES

Pour calculer des clés AES [FIPS 197] de 128 bits, la fonction de hachage SHA-1 [FIPS 180-4] DOIT être utilisée et l'étape supplémentaire suivante DOIT être exécutée :

- utiliser les octets 1 à 16 de keydata ; les octets additionnels ne sont pas utilisés.

Pour calculer des clés AES [FIPS 197] de 192 bits et de 256 bits, la fonction SHA-256 [FIPS 180-4] DOIT être utilisée. Pour les clés AES de 192 bits, l'étape supplémentaire suivante DOIT être exécutée :

- utiliser les octets 1 à 24 de keydata ; les octets additionnels ne sont pas utilisés.

9.7.2 Clés d'accès de base au document

Le calcul de clés 3DES à deux clés à partir d'un germe de clé (K) est utilisé pour établir les clés d'accès de base au document K_{Enc} = **KDF**(K,1) et K_{MAC} = **KDF**(K,2).

9.7.3 PACE

Prenons $\mathbf{KDF}_{\pi}(\pi)$ = **KDF**(f(π),3) comme fonction de calcul de clé pour calculer les clés de chiffrement à partir d'un mot de passe π. Le codage des mots de passe, soit K = f(π), est spécifié au Tableau 14 :

Tableau 14. Codage des mots de passe

Mot de passe	Codage
ZLA	SHA-1 (Numéro de document || Date de naissance || Date d'expiration)
CAN	Chaîne de caractères codée selon l'ISO/IEC 8859-1

Note.— Le numéro du document à utiliser en entrée est toujours le numéro complet du document. Dans le cas de documents TD1 dont le numéro de document est supérieur à neuf caractères, le numéro de document doit être concaténé à partir du champ de numéro de document et du champ de données optionnelles de la ZLA, à l'exclusion du caractère de remplissage. Voir aussi Note j) au § 4.2.2 dans le Doc 9303-5.

9.7.4 Clés de messagerie sécurisée

Les clés de chiffrement et d'authentification sont calculées à partir d'un secret partagé K au moyen de $\mathbf{KDF}_{Enc}(K)$ = **KDF**(K,1) et de $\mathbf{KDF}_{MAC}(K)$ = **KDF**(K,2), respectivement.

9.8 Messagerie sécurisée

9.8.1 Initiation de session

Une session démarre à l'établissement de la messagerie sécurisée. À l'intérieur d'une session, les clés de messagerie sécurisée (établies par BAC, PACE ou l'authentification de la puce) peuvent être changées.

La messagerie sécurisée est basée soit sur 3DES [FIPS 46-3], soit sur AES [FIPS 197] en mode encrypt-then-authenticate, c'est-à-dire que les données sont complétées par remplissage et chiffrées, et les données chiffrées formatées constituent ensuite l'entrée pour le calcul d'authentification. Les clés de session DOIVENT être calculées en utilisant la fonction de calcul de clé décrite au § 9.7.1.

Note.— Le remplissage est toujours effectué par la couche de messagerie sécurisée ; le code d'authentification de message sous-jacent n'a donc pas besoin d'effectuer le remplissage interne.

9.8.2 Compteur de séquence d'envoi

Un entier non signé DOIT être utilisé comme compteur de séquence d'envoi (SSC). La taille en bits du SSC DOIT être égale à la taille en blocs du chiffrement par blocs utilisé pour la messagerie sécurisée, c'est-à-dire 64 bits pour 3DES et 128 bits pour AES.

La valeur du SSC DOIT être augmentée avant chaque génération d'une APDU de commande ou d'une APDU de réponse, c'est-à-dire que, si la valeur de départ est x, la valeur du SSC est x+1 pour la première commande. La valeur du SSC pour la première réponse est x+2.

Si la messagerie sécurisée est redémarrée, le SSC est utilisé comme suit :

- les commandes utilisées pour l'agrément de clé sont protégées avec des clés de session antérieures et des SSC antérieurs. Ce procédé s'applique particulièrement pour la réponse de la dernière commande employée pour l'agrément des clés de session ;
- le SSC est mis à sa nouvelle valeur de départ (voir § 9.8.6.3 pour 3DES et § 9.8.7.3 pour AES) ;
- les nouvelles clés de session et le nouveau SSC sont utilisés pour protéger les commandes et les réponses suivantes.

9.8.3 Fin de session

La puce du DVLM-e DOIT interrompre la messagerie sécurisée si, et seulement si, une erreur de messagerie sécurisée se produit ou si une APDU en clair est reçue.

Si la messagerie sécurisée est interrompue, la puce du DVLM-e DOIT effacer les clés de session stockées et réinitialiser les droits d'accès du terminal.

Note.— La puce du DVLM-e PEUT implicitement choisir le fichier principal lorsqu'il est mis fin à une session.

9.8.4 Structure de message des APDU SM

Les objets de données de la messagerie sécurisée (SM) (voir l'ISO/IEC 7816-4) DOIVENT être utilisés dans l'ordre suivant :

- APDU commande : [DO'85' ou DO'87'] [DO'97'] DO'8E'
- APDU réponse : [DO'85' ou DO'87'] [DO'99'] DO'8E'

Si INS est pair, DO'87' EST utilisé, et si INS est impair, DO'85' EST utilisé.

Tous les objets de données SM DOIVENT être codés en TLV BER comme il est spécifié dans l'ISO/IEC 7816-4. L'en-tête de commande DOIT être inclus dans le calcul de MAC ; l'octet de classe CLA = 0x0c DOIT donc être utilisé.

La valeur réelle de Lc sera modifiée en Lc' après application de la messagerie sécurisée. Au besoin, un objet de données approprié peut facultativement être inclus dans la partie données de l'APDU pour communiquer la valeur d'origine de Lc.

La Figure 5 montre la transformation d'une APDU commande non protégée en APDU commande protégée dans le cas où *Données* et *Le* sont disponibles. Si *Données* n'est pas disponible, omettre la construction de DO '87'. Si *Le* n'est pas disponible, omettre la construction de DO '97'. Pour éviter toute ambiguïté, il est RECOMMANDÉ de ne pas utiliser un champ de valeur vide de l'objet de données Le (voir aussi § 10.4 de l'ISO/IEC 7816-4).

La Figure 6 montre la transformation d'une APDU réponse non protégée en APDU réponse protégée dans le cas où *Données* est disponible. Si *Données* n'est pas disponible, omettre la construction de DO '87'.

9.8.5 Erreurs de SM

Le canal sécurisé pour l'application DVLM-e est interrompu lorsque :

- le CI sans contact est désactivé ;
- le CI sans contact reconnaît une erreur de SM pendant l'interprétation d'une commande. Dans ce cas, les octets d'état doivent être retournés sans SM.

Si la messagerie sécurisée est interrompue, la puce du DVLM-e DOIT effacer les clés de session stockées et réinitialiser les droits d'accès du terminal.

Note.— Il PEUT y avoir d'autres circonstances où le CI sans contact abandonne une session. Il est cependant impossible de donner une liste complète de ces circonstances.

9.8.6 Modes opératoires 3DES

9.8.6.1 Chiffrement

Le chiffrement triple DES (3DES) à deux clés en mode CBC avec zéro IV (c.-à-d. 0x00 00 00 00 00 00 00 00) selon la norme ISO 11568-2 est utilisé. Un remplissage conforme à la méthode de remplissage 2 de la norme ISO/IEC 9797-1 est utilisé.

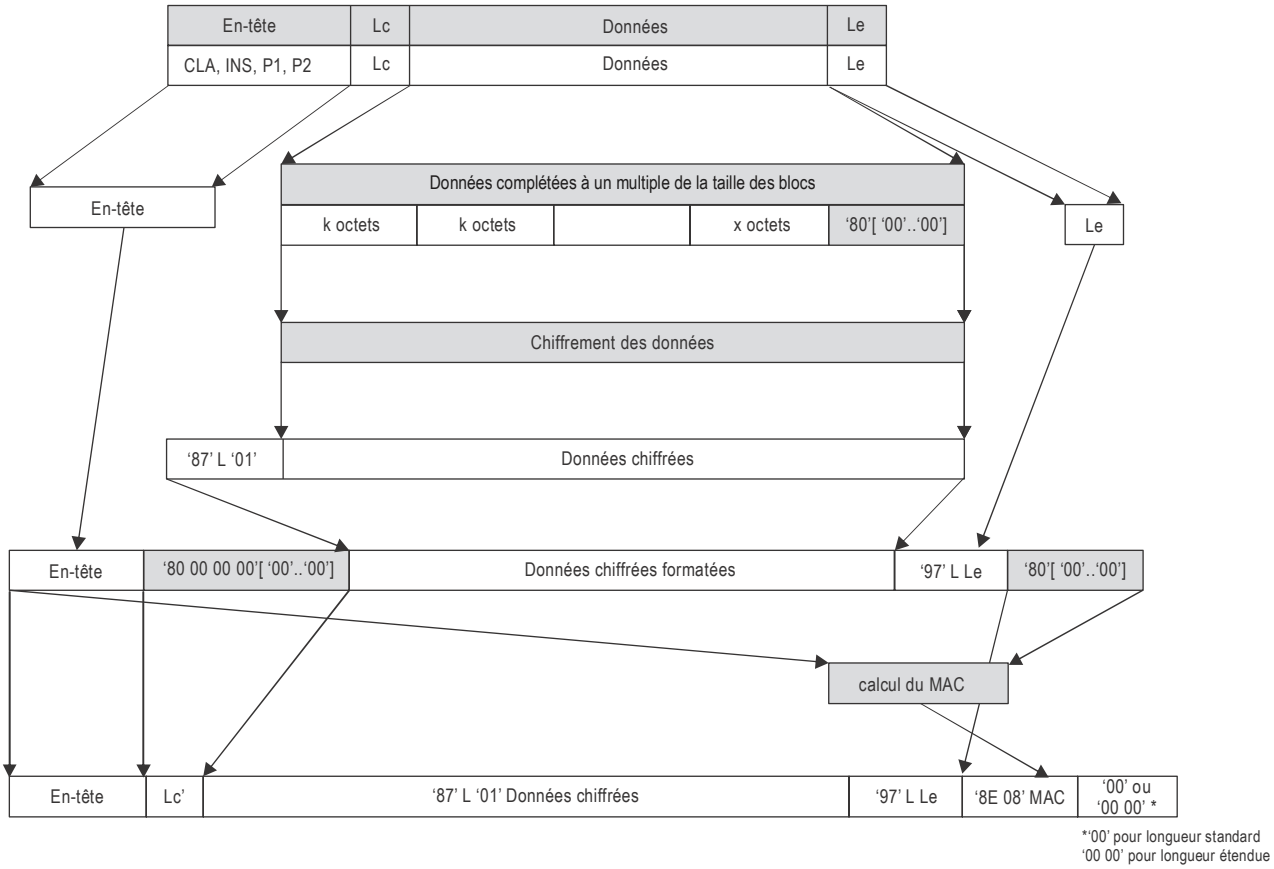

Figure 5. Calcul d'une APDU commande de SM pour un octet INS pair

9.8.6.2 Authentification de message

Les sommes de contrôle cryptographiques sont calculées en utilisant l'algorithme MAC 3 avec chiffrement par blocs DES, zéro IV (8 octets) de l'ISO/IEC 9797-1 et la méthode de remplissage 2 de l'ISO/IEC 9797-1. La longueur de MAC DOIT être de 8 octets.

Après une authentification réussie, le datagramme pour lequel il faut obtenir le MAC DOIT être préfixé du compteur de séquence d'envoi.

9.8.6.3 Compteur de séquence d'envoi

Pour la messagerie sécurisée après le BAC, le compteur de séquence d'envoi (SSC) DOIT être initialisé en concaténant les quatre octets de plus faible poids de RND.IC et de RND.IFD, respectivement :

SSC = RND.IC (4 octets de plus faible poids) || RND.IFD (4 octets de plus faible poids).

Dans tous les autres cas, le SSC DOIT être initialisé à zéro (c.-à-d. 0x00 00 00 00 00 00 00 00).

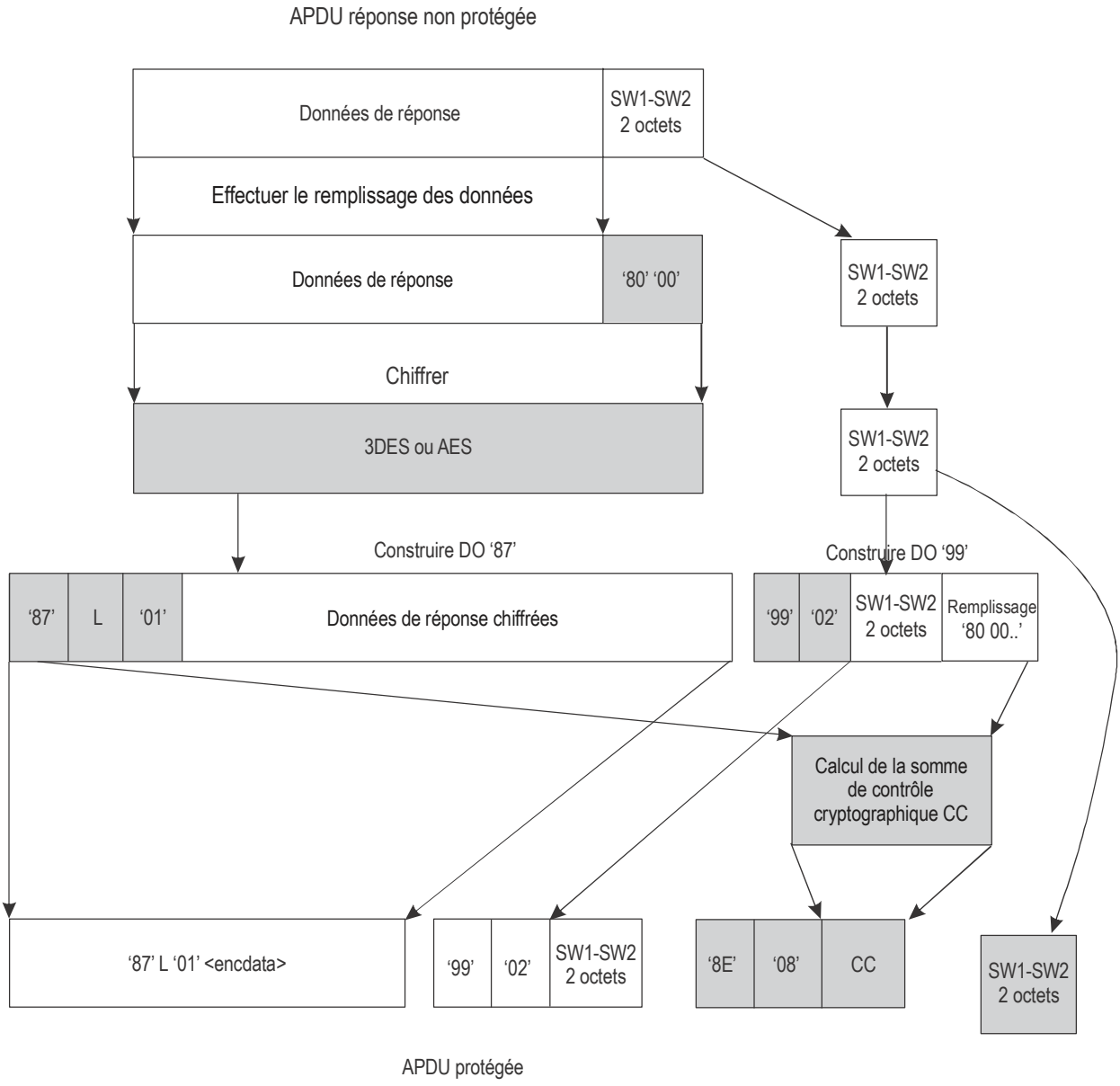

Figure 6. Calcul d'une APDU réponse de SM pour un octet INS pair

9.8.7 Modes opératoires AES

9.8.7.1 Chiffrement

Pour le chiffrement de messages, l'AES [FIPS 197] DOIT être utilisée en mode CBC conformément à l'ISO/IEC 10116, avec clé KS_{Enc} et IV = **E**(KS_{Enc}, SSC).

9.8.7.2 Authentification de message

Pour l'authentification de message, l'AES DOIT être utilisée en mode CMAC [SP 800-38B] avec KS_{MAC} avec une longueur de MAC de 8 octets. Le datagramme à authentifier DOIT être préfixé du compteur de séquence d'envoi.

9.8.7.3 Compteur de séquence d'envoi

Le compteur de séquence d'envoi DOIT être initialisé à zéro (c.-à-d. 0x00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00).

10. RÉFÉRENCES (NORMATIVES)

X9.42 ANSI: X9.42, *Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography,* 1999.

ISO/IEC 7816-4 ISO/IEC 7816-4:2013, *Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange* (Cartes d'identification — Cartes à circuit intégré — Partie 4 : Organisation, sécurité et commandes pour les échanges).

ISO/IEC 7816-8 ISO/IEC 7816-8:2019, *Identification cards — Integrated circuit cards — Part 8: Commands and mechanisms for security operations* (Cartes d'identification — Cartes à circuit intégré — Partie 8 : Commandes et mécanismes pour opérations de sécurité).

ISO/IEC 8859-1 ISO/IEC 8859-1:1998, *Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1* (Technologies de l'information — Jeux de caractères graphiques codés sur un seul octet — Partie 1 : Alphabet latin n° 1).

ISO/IEC 9796-2 ISO/IEC 9796-2:2010, *Information technology — Security techniques — Digital signature schemes giving message recovery — Part 2: Integer factorization based mechanisms* (Technologies de l'information — Techniques de sécurité — Schémas de signature numérique rétablissant le message — Partie 2 : Mécanismes basés sur une factorisation entière).

ISO/IEC 9797-1 ISO/IEC 9797-1:2011, *Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher* (Technologies de l'information — Techniques de sécurité — Codes d'authentification de message (MAC) — Partie 1 : Mécanismes utilisant un chiffrement par blocs).

ISO/IEC 10116 ISO/IEC 10116:2017, *Information technology — Security techniques — Modes of operation for an n-bit block cipher* (Technologies de l'information — Techniques de sécurité — Modes opératoires pour un chiffrement par blocs de n bits).

ISO/IEC 11568-2 ISO/IEC 11568-2:2012, *Financial services — Key management (retail) — Part 2: Symmetric ciphers, their key management and life cycle* [Services financiers — Gestion de clés (services aux particuliers) — Partie 2 : Algorithmes cryptographiques symétriques, leur gestion de clés et leur cycle de vie].

ISO/IEC 11770-2	ISO/IEC 11770-2:2018 IT Security techniques — *Key management — Part 2: Mechanisms using symmetric techniques* (Techniques de sécurité IT — Gestion de clés — Partie 2 : Mécanismes utilisant des techniques symétriques).
FIPS 46-3	NIST FIPS PUB 46-3, *Data Encryption Standard (DES*), 1999.
FIPS 180-4	NIST FIPS PUB 180-4, *Secure hash standard*, 2015.
FIPS 186-4	NIST FIPS PUB 186-4, *Digital Signature Standard (DSS),* 2013.
FIPS 197	NIST FIPS PUB 197, *Specification for the Advanced Encryption Standard (AES),* 2001.
SP 800-38B	NIST Special Publication 800-38B, *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication,* 2005.
RFC 2631	Rescorla, Eric: *RFC 2631 Diffie-Hellman key agreement method*, 1999.
RFC 3447	Jonsson, Jakob and Kaliski, Burt: RFC 3447, Public-key cryptography standards (PKCS) #1: RSA cryptography specifications version 2.1, 2003
RFC 5114	Lepinski, Matt; Kent, Stephen: RFC 5114 *Additional Diffie-Hellman Groups for Use with IETF Standards*, 2008.
RFC 5280	D. Cooper, S. Santesson, S. Farrell, S. Boyen, R. Housley, W. Polk: *RFC 5280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, 2008.
RFC 5639	Lochter, Manfred; Merkle, Johannes: RFC 5639 *Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation*, 2010.
TR-03110	BSI: Technical Guideline TR-03110: Advanced Security Mechanisms for Machine Readable Travel Documents.
TR-03111	BSI: *Technical Guideline TR-03111: Elliptic Curve Cryptography*, Version 2.0, 2012.
PKCS#1	RSA Laboratories, PKCS#1 v2.2: RSA Cryptography Standard, 2012.
PKCS#3	RSA Laboratories, PKCS#3: *Diffie-Hellman key-agreement standard*, 1993.
Keesing2009	J. Bender, D. Kügler: *Introducing the PACE solution*, dans: Keesing Journal of Documents & Identity, nº 30, Keesing, 2009.
BFK2009	J. Bender, M. Fischlin, D. Kügler: *Security Analysis of the PACE Key-Agreement Protocol*, in: Proceedings ISC 2009, LNCS volume 5735, Springer, 2009.
BCIMRT2010	Brier, Eric; Coron, Jean-Sébastien; Icart, Thomas; Madore, David; Randriam, Hugues; and Tibouch, Mehdi: *Efficient Indifferentiable Hashing into Ordinary Elliptic Curves, Advances in Cryptology — CRYPTO 2010*, Springer-Verlag, 2010.

— — — — — — — —

Appendice A à la Partie 11
(INFORMATIF)

ENTROPIE DES CLÉS D'ACCÈS CALCULÉES À PARTIR DE LA ZLA

En raison de sa simplicité, le contrôle d'accès de base (BAC) a eu beaucoup de succès et il est implémenté dans presque tous les PLM-e.

La sécurité assurée par le contrôle d'accès de base est limitée par la conception du protocole. Les clés d'accès de base au document (K_{Enc} et K_{MAC}) sont produites à partir de données imprimées avec très peu d'éléments aléatoires. Les données utilisées pour la génération des clés sont le numéro de document, la date de naissance et la date d'expiration. En conséquence, les clés qui en résultent ont une entropie relativement faible et sont cryptographiquement faibles. L'entropie réelle dépend principalement du type du numéro de document. Pour un document de voyage d'une durée de validité de 10 ans, la force maximale des clés est d'environ :

- 56 bits pour un numéro de document numérique ($365^2 * 10^{12}$ possibilités)
- 73 bits pour un numéro de document alphanumérique ($365^2*36^9*10^3$ possibilités).

Dans le second cas particulièrement, cette estimation exige que le numéro de document soit choisi de manière aléatoire et uniforme, ce qui n'est habituellement pas le cas. Selon les connaissances de l'attaquant, l'entropie réelle d'une clé d'accès de base au document peut être inférieure, par exemple, si l'attaquant connaît tous les numéros de document en vigueur ou s'il est capable de corréler les numéros de document et les dates d'expiration.

Il n'existe aucun moyen simple de renforcer le contrôle d'accès de base étant donné que ses limites sont inhérentes à la conception du protocole, qui est basée sur un chiffrement symétrique (« clé secrète »). Un mécanisme de contrôle d'accès cryptographiquement fort doit (en plus) utiliser un chiffrement asymétrique (« clé publique »).

L'établissement de connexion avec authentification par mot de passe (PACE) a été conçu pour résoudre ce problème. Il utilise un chiffrement asymétrique pour établir les clés de session, dont la force ne dépend pas de l'entropie du mot de passe employé. Si PACE est mis en œuvre avec un chiffrement à courbe elliptique avec des courbes 256 bits et l'AES-128 (un choix courant), les clés de session ont une entropie de 128 bits.

Il convient de faire la différence entre deux types d'attaque :

- L'écrémage : il s'agit d'une attaque en ligne, c'est-à-dire que l'attaquant tente d'accéder au CI sans contact en temps réel, par exemple, en devinant le mot de passe. Si le protocole utilisé pour protéger le CI sans contact n'a pas de faiblesses cryptographiques, la probabilité de succès de l'attaquant est donnée par la durée d'accès au CI par l'attaquant, la durée d'une seule tentative pour deviner le mot de passe et l'entropie du passeport.
- L'interception illicite : il s'agit d'une attaque hors ligne, c'est-à-dire que l'attaquant essaie de déchiffrer la communication interceptée sans accéder au CI sans contact. Si le protocole employé pour établir les clés de session n'a pas de faiblesses cryptographiques, la probabilité de succès est donnée par la force des clés de session et la capacité de calcul dont dispose l'attaquant.

Pour plus de renseignements, voir Keesing2009 pour une analyse générale de l'entropie des clés de session et une comparaison entre BAC et PACE, et BFK2009 pour une analyse cryptographique de PACE.

— — — — — — — —

Appendice B à la Partie 11
(INFORMATIF)

CODAGE DES POINTS POUR LE MAPPAGE INTÉGRÉ — ECDH

B.1 DESCRIPTION DE HAUT NIVEAU DE LA MÉTHODE DE CODAGE DES POINTS

L'algorithme prend comme entrées les paramètres de la courbe *(a, b, p, f)*, où *(a, b)* sont les coefficients de la courbe et p est la caractéristique du corps premier sur lequel la courbe

$$E : y^2 \equiv x^3 + ax + b \ (mod\ p)$$

est définie. L'ordre de E est toujours de forme fq pour un nombre premier q et f est appelé le cofacteur. PACE v2 exige la génération d'un point appartenant au sous-groupe q de E, désigné par $E[q]$. Le codage de points prend aussi comme entrée un nombre t tel que

$$0 < t < p$$

et renvoie, en temps constant, un point appartenant à $E[q]$. Comme il est décrit dans BCIMRT2010, le codage de points emploie deux modes, selon le système de coordonnées utilisé :

- une première mise en œuvre, décrite dans § B.2, donne le point de courbe elliptique en coordonnées affines *(x, y)* ;
- une autre mise en œuvre, décrite dans § B.3, donne le même point en coordonnées jacobiennes *(X, Y, Z).*

Quelle que soit l'option choisie, le point généré est identique en ce sens que

$$x = XZ^{-2} \ mod\ p \ et\ y = YZ^{-3} \ mod\ p$$

et la mise en œuvre de la phase suivante de PACE v2 (la phase d'échange de clés Diffie-Hellman à courbe elliptique) peut donc utiliser l'option qui correspond le mieux à l'interface de l'API cryptographique qui exécute les opérations de courbe elliptique.

Comme il est noté plus bas, le codage de points pour les coordonnées affines exige environ deux exponentiations modulaires modulo p tandis que les coordonnées jacobiennes n'en exigent qu'une seule.

À noter que dans les deux options disponibles, le codage de points exige explicitement que $p \equiv 3 \ mod\ 4$.

B.2 OPTION COORDONNÉES AFFINES

L'algorithme se présente comme suit :

Entrées : paramètres de la courbe *(a, b, p, f)* et t tel que $0 < t < p$

Sortie : un point (x, y) dans le sous-groupe d'ordre premier $E[q]$ de E

1. Calculer $\alpha = -t^2 \bmod p$
2. Calculer $X_2 = -ba^{-1}(1+(\alpha+\alpha^2)^{-1}) \bmod p$
3. Calculer $X_3 = \alpha X_2 \bmod p$
4. Calculer $h_2 = (X_2)^3 + a X_2 + b \bmod p$
5. Calculer $h_3 = (X_3)^3 + a X_3 + b \bmod p$
6. Calculer $U = t^3 h_2 \bmod p$
7. Calculer $A = (h_2)^{p-1-(p+1)/4} \bmod p$
8. Si $A^2 h_2 = 1 \bmod p$ définir $(x, y) = (X_2, A h_2 \bmod p)$
9. Sinon définir $(x, y) = (X_3, A U \bmod p)$
10. Sortie $(x, y) = [f] (x, y)$

Notes

Compte non tenu des multiplications et additions modulaires, le temps d'exécution de cette option est dominé par deux exponentiations modulaires :

- l'étape 2 peut être réécrite :

$$X_2 = -ba^{-1}(1+(\alpha+\alpha^2)^{-1}) = -b(1+\alpha+\alpha^2)\,(a(\alpha+\alpha^2))^{p-2} \bmod p$$

  ce qui correspond essentiellement à une exponentiation modulaire d'exposant $p-2$;

- l'étape 7 est une exponentiation modulaire d'exposant $p-1-(p+1)/4$.

Note.— L'étape 10 requiert une multiplication scalaire par le cofacteur f. *Pour de nombreuses courbes, le cofacteur est égal à 1 de manière à éviter cette multiplication scalaire.*

B.3 OPTION COORDONNÉES JACOBIENNES

L'algorithme se présente comme suit :

Entrées : paramètres de la courbe (a, b, p, f) et t tel que $0 < t < p$
Sortie : un point (X, Y, Z) dans le sous-groupe d'ordre premier $E[q]$ de E

1. Calculer $\alpha = -t^2 \bmod p$
2. Calculer $Z = a(\alpha+\alpha^2) \bmod p$
3. Calculer $X_2 = -bZ\,(1+\alpha+\alpha^2) \bmod p$
4. Calculer $X_3 = \alpha X_2 \bmod p$
5. Calculer $h_2 = (X_2)^3 + a X_2 Z^4 + b Z^6 \bmod p$
6. Calculer $h_3 = (X_3)^3 + a X_3 Z^4 + b Z^6 \bmod p$
7. Calculer $U = -\alpha\, t\, h_2 \bmod p$
8. Calculer $A = (h_2)^{p-1-(p+1)/4} \bmod p$
9. Si $A^2 h_2 = 1 \bmod p$ définir $(X, Y, Z) = (X_2, A h_2 \bmod p, Z)$
10. Sinon définir $(X, Y, Z) = (X_3, A U \bmod p, Z)$
11 Sortie $(X, Y, Z) = [f](X, Y, Z)$

Notes

Compte non tenu des multiplications et additions modulaires, le temps d'exécution de cette option est dominé par une seule exponentiation modulaire (étape 7). Cette option devrait donc être environ deux fois plus rapide que celle des coordonnées affines.

Note.— La multiplication scalaire de l'étape 10 peut être complètement évitée si le cofacteur f *est égal à 1.*

— — — — — — — —

Appendice C à la Partie 11
(INFORMATIF)

SÉMANTIQUE DES QUESTIONS

Prenons comme exemple une signature basée sur un protocole question-réponse entre une puce (CI) de DVLM-e et un terminal (IFD), la puce du DVLM-e voulant démontrer qu'elle connaît sa clé privée SK_{IC} :

- le terminal envoie une question *c* choisie aléatoirement à la puce du DVLM-e ;
- la puce du DVLM-e répond par la signature *s*=**Sign**(SK_{IC},c).

Même si ce protocole est très simple et efficace, la puce du DVLM-e signe en fait le message *c* sans connaître la sémantique du message. Vu que les signatures fournissent une preuve d'authenticité transférable, n'importe quelle tierce partie peut (en principe) être persuadée que la puce du DVLM-e a bien signé ce message.

Même si le message *c* devrait être une chaîne de bits aléatoire, le terminal peut tout aussi bien générer cette chaîne de bits d'une façon imprévisible mais (publiquement) vérifiable, par exemple : SK_{IFD} est la clé privée du terminal et

$$c = \textbf{Sign}(SK_{IFD}, ID_{IC}||Date||Heure||Lieu)$$

est la question générée en utilisant un schéma de signature avec rétablissement du message. La signature garantit que le terminal a effectivement généré cette question. Vu la transférabilité de la signature du terminal, n'importe quel tiers faisant confiance au terminal et connaissant la clé publique PK_{IFD} correspondante peut vérifier que la question a été créée correctement en vérifiant cette signature. En outre, vu la transférabilité de la signature de la puce du DVLM-e sur la question, le tiers peut conclure que l'affirmation est vraie : la puce du DVLM-e se trouvait effectivement à une certaine date et à une certaine heure à un certain endroit.

L'aspect positif est que les États peuvent utiliser la sémantique des questions pour leur usage national, par exemple, pour prouver qu'une certaine personne a effectivement immigré. L'aspect négatif est qu'il peut être fait mauvais usage de ces preuves pour poursuivre des personnes. En particulier, vu que l'authentification active n'est pas limitée aux terminaux autorisés, l'usage abusif est possible. Le pire scénario serait des puces de DVLM-e fournissant une authentification active sans contrôle d'accès de base. Dans ce cas, un système de poursuite très puissant peut être mis en place en installant des modules matériels sécurisés à des endroits de choix. Les consignations qui en résultent ne peuvent pas être falsifiées en raison des signatures. Le contrôle d'accès de base réduit ce problème dans une certaine mesure vu qu'il est nécessaire d'avoir une interaction avec le détenteur du document. Le problème demeure néanmoins mais il se limite aux endroits où le document de voyage du détenteur est lu, par exemple, par les transporteurs aériens ou les hôtels.

On pourrait objecter que, tout particulièrement dans un scénario sans contact, les questions peuvent être interceptées illicitement et réutilisées à une date, une heure ou un endroit différents et rendre la preuve du moins non fiable. Bien que l'interception illicite des questions soit techniquement possible, l'argument n'est pas valide. Par hypothèse, on considère que le terminal produit les questions correctement et on peut supposer qu'il a vérifié l'identité de la puce du DVLM-e avant de commencer l'authentification active. La question interceptée contiendra donc une identité différente de celle du démonstrateur qui signe la question.

— — — — — — — —

Appendice D à la Partie 11
(INFORMATIF)

EXEMPLE DÉTAILLÉ : CONTRÔLE D'ACCÈS DE BASE

D.1 CALCUL DES CLÉS À PARTIR DU GERME DE CLÉ (K_{SEED})

La présente section donne un exemple de calcul de clés 3DES à partir d'une valeur germe K_{seed}. Cette procédure est utilisée comme « sous-programme » dans les exemples du contrôle d'accès de base.

Entrée :

K_{seed} = '239AB9CB282DAF66231DC5A4DF6BFBAE'

Calculer la clé de chiffrement (c = '00000001') :

1. Concaténer K_{seed} et c :
   D = '239AB9CB282DAF66231DC5A4DF6BFBAE00000001'

2. Calculer le hachage SHA-1 de D :
   H_{SHA-1}(D) = 'AB94FCEDF2664EDFB9B291F85D7F77F27F2F4A9D'

3. Former des clés DES K_a et K_b, destinées à être utilisées comme première et deuxième clé 3DES (c'est-à-dire la clé 3DES dans la concaténation de K_a et de K_b) :
   $K_{a'}$ = 'AB94FCEDF2664EDF'
   $K_{b'}$ = 'B9B291F85D7F77F2'

4. Ajuster les bits de parité :
   K_a = 'AB94FDECF2674FDF'
   K_b = 'B9B391F85D7F76F2'

Calculer la clé de calcul de MAC (c = '00000002') :

1. Concaténer K_{seed} et c :
   D = '239AB9CB282DAF66231DC5A4DF6BFBAE00000002'

2. Calculer le hachage SHA-1 de D :
   H_{SHA-1}(D) = '7862D9ECE03C1BCD4D77089DCF131442814EA70A'

3. Former des clés $K_{a'}$ et $K_{b'}$:
   $K_{a'}$ = '7862D9ECE03C1BCD'
   $K_{b'}$ = '4D77089DCF131442'

4. Ajuster les bits de parité :
   K_a = '7962D9ECE03D1ACD'
   K_b = '4C76089DCE131543'

D.2 CALCUL DES CLÉS D'ACCÈS DE BASE AU DOCUMENT (K_{ENC} ET K_{MAC})

La présente section donne des exemples de calcul des clés d'accès de base à partir de la ZLA.

ZLA d'un TD2, numéro de document dépassant 9 caractères

1. Lire la ZLA :

   ZLA = `I<UTOSTEVENSON<<PETER<JOHN<<<<<<<<<<`
   `D23145890<UTO3407127M95071227349<<<8`

2. Construire l'information_ZLA à partir de la ZLA :

   Numéro de document = `D23145890734` Chiffre de contrôle = `9`
   Date de naissance = `340712` Chiffre de contrôle = `7`
   Date d'expiration = `950712` Chiffre de contrôle = `2`
   Information_ZLA = `D231458907349340712795071 22`

Passer à l'étape 3.

ZLA d'un TD2, numéro de document ayant 9 caractères

1. Lire la ZLA :

   ZLA = `I<UTOERIKSSON<<ANNA<MARIA<<<<<<<<<<`
   `L898902C<3UTO6908061F9406236<<<<<<<8`

2. Construire l'information_ZLA à partir de la ZLA :

   Numéro de document = `L898902C<` Chiffre de contrôle = `3`
   Date de naissance = `690806` Chiffre de contrôle = `1`
   Date d'expiration = `940623` Chiffre de contrôle = `6`
   Information_ZLA = `L898902C<3690806194062 36`

Passer à l'étape 3.

ZLA d'un TD1, numéro de document dépassant 9 caractères

1. Lire la ZLA :

   ZLA = `I<UTOD23145890<7349<<<<<<<<<<`
   `3407127M9507122UTO<<<<<<<<<<<2`
   `STEVENSON<<PETER<JOHN<<<<<<<<<`

2. Construire l'information_ZLA à partir de la ZLA :

   Numéro de document = `D23145890734` Chiffre de contrôle = `9`
   Date de naissance = `340712` Chiffre de contrôle = `7`
   Date d'expiration = `950712` Chiffre de contrôle = `2`
   Information_ZLA = `D231458907349340712795071 22`

Passer à l'étape 3.

ZLA d'un TD1, numéro de document ayant 9 caractères

1. Lire la ZLA :

   ZLA = `I<UTOL898902C<3<<<<<<<<<<<<<<`
   `6908061F9406236UTO<<<<<<<<<<<1`
   `ERIKSSON<<ANNA<MARIA<<<<<<<<<<`

2. Construire l'information_ZLA à partir de la ZLA :
   Numéro de document = L898902C< Chiffre de contrôle = 3
   Date de naissance = 690806 Chiffre de contrôle = 1
   Date d'expiration = 940623 Chiffre de contrôle = 6
   Information_ZLA = L898902C<3690806194062 36

3. Calculer le hachage SHA-1 de l'information_ZLA :
   H_{SHA-1}(information_ZLA) = '239AB9CB282DAF66231DC5A4DF6BFBAEDF477565'

4. Prendre les 16 octets les plus significatifs pour former K_{seed} :
   K_{seed} = '239AB9CB282DAF66231DC5A4DF6BFBAE'

5. Calculer les clés d'accès de base (K_{Enc} et K_{MAC}) conformément aux indications du § 9.7.1/Appendice D.1 :
   K_{Enc} = 'AB94FDECF2674FDFB9B391F85D7F76F2'
   K_{MAC} = '7962D9ECE03D1ACD4C76089DCE131543'

D.3 AUTHENTIFICATION ET ÉTABLISSEMENT DE CLÉS DE SESSION

La présente section donne un exemple d'exécution du contrôle d'accès de base.

Système d'inspection :

1. Demander au CI sans contact du DVLM-e un nombre aléatoire de 8 octets :

APDU commande :				
CLA	**INS**	**P1**	**P2**	**Le**
00	84	00	00	08

APDU réponse :	
Champ données de la réponse	**SW1-SW2**
RND.IC	9000

   RND.IC = '4608F91988702212'

2. Générer un élément aléatoire de 8 octets et un élément aléatoire de 16 octets :
   RND.IFD = '781723860C06C226'
   K_{IFD} = '0B795240CB7049B01C19B33E32804F0B'

3. Concaténer RND.IFD, RND.IC et K_{IFD} :
   S = '781723860C06C2264608F91988702212
   0B795240CB7049B01C19B33E32804F0B'

4. Chiffrer S avec la clé 3DES K_{Enc} :
   E_{IFD} = '72C29C2371CC9BDB65B779B8E8D37B29
   ECC154AA56A8799FAE2F498F76ED92F2'

5. Calculer MAC sur E_{IFD} avec la clé 3DES K_{MAC} :
   M_{IFD} = '5F1448EEA8AD90A7'

6. Construire les données de commande pour EXTERNAL AUTHENTICATE (authentification externe) et envoyer l'APDU commande au CI sans contact du DVLM-e :

   cmd_data = '72C29C2371CC9BDB65B779B8E8D37B29ECC154AA
   56A8799FAE2F498F76ED92F25F1448EEA8AD90A7'

APDU commande :						
CLA	**INS**	**P1**	**P2**	**Lc**	**Champ données de la commande**	**Le**
00	82	00	00	28	cmd_data	28

CI sans contact du DVLM-e :

1. Déchiffrer et vérifier les données reçues et comparer RND.IC avec la réponse à GET CHALLENGE (acquérir question).

2. Générer un élément aléatoire de 16 octets :
   K_{IC} = '0B4F80323EB3191CB04970CB4052790B'

3. Calculer XOR de K_{IFD} et K_{IC} :
   K_{seed} = '0036D272F5C350ACAC50C3F572D23600'

4. Calculer les clés de session (KS_{Enc} et KS_{MAC}) conformément aux indications du § 9.7.1/Appendice D.1 :
   KS_{Enc} = '979EC13B1CBFE9DCD01AB0FED307EAE5'
   KS_{MAC} = 'F1CB1F1FB5ADF208806B89DC579DC1F8'

5. Calculer le compteur séquence d'envoi :
   SSC = '887022120C06C226'

6. Concaténer RND.IC, RND.IFD et K_{IC} :
   R = '4608F91988702212781723860C06C226
   0B4F80323EB3191CB04970CB4052790B'

7. Chiffrer R avec la clé 3DES K_{Enc} :
   E_{IC} = '46B9342A41396CD7386BF5803104D7CE
   DC122B9132139BAF2EEDC94EE178534F'

8. Calculer MAC sur E_{IC} avec la clé 3DES K_{MAC} :
   M_{IC} = '2F2D235D074D7449'

9. Construire les données de réponse pour EXTERNAL AUTHENTICATE (authentification externe) et envoyer l'APDU réponse au système d'inspection :

   resp_data = '46B9342A41396CD7386BF5803104D7CEDC122B91
   32139BAF2EEDC94EE178534F2F2D235D074D7449'

APDU réponse :	
Champ données de la réponse	**SW1-SW2**
resp_data	9000

Système d'inspection :

1. Déchiffrer et vérifier les données reçues et comparer le RND.IFD reçu avec le RND.IFD généré.

2. Calculer XOR de K_{IFD} et K_{IC} :
   K_{seed} = '0036D272F5C350ACAC50C3F572D23600'

3. Calculer les clés de session (KS_{Enc} et KS_{MAC}) conformément aux indications du § 9.7.1/D.1 :
   KS_{Enc} = '979EC13B1CBFE9DCD01AB0FED307EAE5'
   KS_{MAC} = 'F1CB1F1FB5ADF208806B89DC579DC1F8'

4. Calculer le compteur de séquence d'envoi :
   SSC = '887022120C06C226'

D.4 MESSAGERIE SÉCURISÉE

Après authentification et établissement des clés de session, le système d'inspection sélectionne le fichier EF.COM (ID de fichier = '011E') et lit les données en utilisant la messagerie sécurisée. Les KS_{Enc}, KS_{MAC} et SSC calculés (précédentes étapes 3 et 4 du système d'inspection) sont utilisés.

Le fichier EF.COM est sélectionné en premier ; les quatre premiers octets de ce fichier sont lus en premier de sorte que la longueur de la structure dans le fichier puisse être déterminée ; les octets restants sont lus ensuite.

1. Sélectionner EF.COM :

   APDU commande non protégée :

CLA	INS	P1	P2	Lc	Champ données de la commande
00	A4	02	0C	02	01 1E

   a) Masquer l'octet de classe et ajouter le remplissage à l'en-tête de la commande :
      CmdHeader = '0CA4020C80000000'

   b) Effectuer le remplissage des données :
      Données = '011E800000000000'

   c) Chiffrer les données avec KS_{Enc} :
      Données chiffrées = '6375432908C044F6'

   d) Construire DO'87' :
      DO87 = '8709016375432908C044F6'

e) Concaténer CmdHeader et DO'87' :
M = '0CA4020C8000000087090163754329 08C044F6'

f) Calculer le MAC de M :

1) Incrémenter SSC de 1 :
SSC = '887022120C06C227'

2) Concaténer SSC et M et ajouter le remplissage :
N = '887022120C06C2270CA4020C80000000
8709016375432908C044F68000000000'

3) Calculer le MAC sur N avec KS_{MAC} :
CC = 'BF8B92D635FF24F8'

g) Construire DO'8E' :
DO8E = '8E08BF8B92D635FF24F8'

h) Construire et envoyer l'APDU protégée :
ProtectedAPDU = '0CA4020C158709016375432908C0
44F68E08BF8B92D635FF24F800'

i) Recevoir l'APDU réponse du CI sans contact du DVLM-e :
RAPDU = '990290008E08FA855A5D4C50A8ED9000'

j) Vérifier RAPDU CC en calculant le MAC de DO'99' :

1) Incrémenter SSC de 1 :
SSC = '887022120C06C228'

2) Concaténer SSC et DO'99' et ajouter le remplissage :
K = '887022120C06C2289902900080000000'

3) Calculer MAC avec KS_{MAC} :
CC' = 'FA855A5D4C50A8ED'

4) Comparer CC' avec les données de DO'8E' de RAPDU :
'FA855A5D4C50A8ED' == 'FA855A5D4C50A8ED' ? OUI.

2. Lire les éléments binaires des quatre premiers octets :

APDU commande non protégée :

CLA	INS	P1	P2	Le
00	B0	00	00	04

a) Masquer l'octet de classe et ajouter le remplissage à l'en-tête de la commande :
CmdHeader = '0CB0000080000000'

b) Construire DO'97' :
DO97 = '970104'

c) Concaténer CmdHeader et DO'97' :
M = '0CB0000080000000970104'

d) Calculer le MAC de M :

1) Incrémenter SSC de 1 :
SSC = '887022120C06C229'

2) Concaténer SSC et M et ajouter le remplissage :
N = '887022120C06C2290CB00000
800000009701048000000000'

3) Calculer le MAC sur N avec KS_{MAC} :
CC = 'ED6705417E96BA55'

e) Construire DO'8E' :
DO8E = '8E08ED6705417E96BA55'

f) Construire et envoyer l'APDU protégée :
ProtectedAPDU = '0CB000000D9701048E08ED6705417E96BA5500'

g) Recevoir l'APDU réponse du CI sans contact du DVLM-e :
RAPDU = '8709019FF0EC34F992265199029000
8E08AD55CC17140B2DED9000'

h) Vérifier RAPDU CC en calculant le MAC de la concaténation de DO'87' et de DO'99' :

1) Incrémenter SSC de 1 :
SSC = '887022120C06C22A'

2) Concaténer SSC, DO'87' et DO'99' et ajouter le remplissage :
K = '887022120C06C22A8709019F
F0EC34F99226519902900080'

3) Calculer MAC avec KS_{MAC} :
CC' = 'AD55CC17140B2DED'

4) Comparer CC' avec les données de DO'8E' de RAPDU :
'AD55CC17140B2DED' == 'AD55CC17140B2DED' ? OUI.

i) Déchiffrer les données de DO'87' avec KS_{Enc} :
DecryptedData = '60145F01'

j) Déterminer la longueur de la structure :
L = '14' + 2 = 22 octets

3. Lire les éléments binaires des 18 octets restants à partir du décalage 4 :

APDU commande non protégée :

CLA	INS	P1	P2	Le
00	B0	00	04	12

a) Masquer l'octet de classe et ajouter le remplissage à l'en-tête de la commande :
CmdHeader = '0CB0000480000000'

b) Construire DO'97' :
DO97 = '970112'

c) Concaténer CmdHeader et DO'97' :
M = '0CB000048000000970112'

d) Calculer le MAC de M :

1) Incrémenter SSC de 1 :
SSC = '887022120C06C22B'

2) Concaténer SSC et M et ajouter le remplissage :
N = '887022120C06C22B0CB00004
8000000097011280000000000'

3 Calculer le MAC sur N avec KS_{MAC} :
CC = '2EA28A70F3C7B535'

e) Construire DO'8E' :
DO8E = '8E082EA28A70F3C7B535'

f) Construire et envoyer l'APDU protégée :
ProtectedAPDU = '0CB000040D9701128E082EA28A70F3C7B53500'

g) Recevoir l'APDU réponse du CI sans contact du DVLM-e :
RAPDU = '871901FB9235F4E4037F2327DCC8964F1F9B8C30F42
C8E2FFF224A990290008E08C8B2787EAEA07D749000'

h) Vérifier RAPDU CC en calculant le MAC de la concaténation de DO'87' et de DO'99' :

1) Incrémenter SSC de 1 :
SSC = '887022120C06C22C'

2) Concaténer SSC, DO'87' et DO'99' et ajouter le remplissage :
K = '887022120C06C22C871901FB9235F4E4037F232
7DCC8964F1F9B8C30F42C8E2FFF224A99029000'

3) Calculer le MAC avec KS_{MAC} :
CC' = 'C8B2787EAEA07D74'

4) Comparer CC' avec les données de DO'8E' de RAPDU :
'C8B2787EAEA07D74' == 'C8B2787EAEA07D74' ? OUI.

i) Déchiffrer les données de DO'87' avec KS_{Enc} :
DecryptedData = '04303130365F36063034303030305C026175'

RÉSULTAT :

Données EF.COM = '60145F0104303130365F36063034303030305C026175'

— — — — — — — —

Appendice E à la Partie 11
(INFORMATIF)

EXEMPLE DÉTAILLÉ : AUTHENTIFICATION PASSIVE

Étape 1. Lire l'objet de sécurité du document (SO_D) [contenant facultativement le certificat de signataire de document (C_{DS})] à partir du CI sans contact.

Étape 2. Lire le signataire de document (SD) à partir de l'objet de sécurité du document (SO_D).

Étape 3. Le système d'inspection vérifie SO_D en utilisant la clé publique de signataire de document.

Étape 4. Le système d'inspection vérifie C_{DS} en utilisant la clé publique d'AC signataire nationale.

Si les vérifications des étapes 3 et 4 sont correctes, cela garantit qu'il peut être fait confiance au contenu de SO_D et qu'il peut être utilisé dans le processus d'inspection.

Étape 5. Lire les groupes de données pertinents à partir de la structure de données logique (SDL).

Étape 6. Calculer les hachages des groupes de données pertinents.

Étape 7. Comparer les hachages calculés avec les valeurs de hachage correspondantes dans le SO_D.

Si les valeurs de hachage de l'étape 7 sont identiques, cela garantit que le contenu du groupe de données est authentique et inchangé.

— — — — — — — —

Appendice F à la Partie 11
(INFORMATIF)

EXEMPLE DÉTAILLÉ : AUTHENTIFICATION ACTIVE

Le présent exemple emploie les valeurs suivantes :

1. Mécanisme basé sur la factorisation d'entiers : RSA

2. Longueur de module (k) : 1 024 bits (128 octets)

3. Algorithme de hachage : SHA-1

Système d'inspection :

Étape 1. Générer un élément aléatoire de 8 octets :
RND.IFD = 'F173589974BF40C6'

Étape 2. Construire la commande pour l'authentification interne et envoyer l'APDU commande au CI sans contact du DVLM-e :

APDU commande

CLA	INS	P1	P2	Lc	Champ données de la commande	Le
00	88	00	00	08	RND.IFD	00

CI sans contact du DVLM-e :

Étape 3. Déterminer M_2 à partir de l'APDU entrante :
M_2 = 'F173589974BF40C6'

Étape 4. Créer l'indicateur de fin :
T = 'BC' (c.-à-d. SHA-1)
t (longueur de T en octets) = 1

Étape 5. Déterminer les longueurs :
a. $c = k - L_h - 8t - 4 = 1\,024 - 160 - 8 - 4 = 852$ bits
b. $L_{M1} = c - 4 = 848$ bits

Étape 6. Générer le nonce M_1 de longueur L_{M1} :
M_1 = '9D2784A67F8E7C659973EA1AEA25D95B
6C8F91E5002F369F0FBDCE8A3CEC1991
B543F1696546C5524CF23A5303CD6C98
599F40B79F377B5F3A1406B3B4D8F967
84D23AA88DB7E1032A405E69325FA91A
6E86F5C71AEA978264C4A207446DAD4E
7292E2DCDA3024B47DA8'

Étape 7. Créer M :

M = M_1 | M_2 = ‘9D2784A67F8E7C659973EA1AEA25D95B
6C8F91E5002F369F0FBDCE8A3CEC1991
B543F1696546C5524CF23A5303CD6C98
599F40B79F377B5F3A1406B3B4D8F967
84D23AA88DB7E1032A405E69325FA91A
6E86F5C71AEA978264C4A207446DAD4E
7292E2DCDA3024B47DA8F173589974BF
40C6’

Étape 8. Calculer le condensé SHA-1 de M :

H = SHA-1(M) = ‘C063AA1E6D22FBD976AB0FE73D94D2D9
C6D88127’

Étape 9[2]. Construire la représentation du message :

F = ‘6A’ | M_1 | H | T =
‘6A9D2784A67F8E7C659973EA1AEA25D9
5B6C8F91E5002F369F0FBDCE8A3CEC19
91B543F1696546C5524CF23A5303CD6C
98599F40B79F377B5F3A1406B3B4D8F9
6784D23AA88DB7E1032A405E69325FA9
1A6E86F5C71AEA978264C4A207446DAD
4E7292E2DCDA3024B47DA8C063AA1E6D
22FBD976AB0FE73D94D2D9C6D88127BC’

Étape 10. Chiffrer F avec la clé privée d’authentification active pour former la signature :

S = ‘756B683B036A6368F4A2EB29EA700F96
E26100AFC0809F60A91733BA29CAB362
8CB1A017190A85DADE83F0B977BB513F
C9C672E5C93EFEBBE250FE1B722C7CEE
F35D26FC8F19219C92D362758FA8CB0F
F68CEF320A8753913ED25F69F7CEE772
6923B2C43437800BBC9BC028C49806CF
2E47D16AE2B2CC1678F2A4456EF98FC9’

Étape 11. Construire les données de réponse pour INTERNAL AUTHENTICATE (authentification interne) et envoyer l’APDU réponse au système d’inspection :

APDU réponse

Champ données de la réponse	**SW1-SW2**
S	9000

2. Puisque la partie connue (RND.IFD) n’est pas retournée, mais doit être adjointe par l’IFD lui-même, le rétablissement partiel s’applique (‘6A’).

Système d'inspection :

Étape 12. Déchiffrer la signature avec la clé publique :

F = '6A9D2784A67F8E7C659973EA1AEA25D9
5B6C8F91E5002F369F0FBDCE8A3CEC19
91B543F1696546C5524CF23A5303CD6C
98599F40B79F377B5F3A1406B3B4D8F9
6784D23AA88DB7E1032A405E69325FA9
1A6E86F5C71AEA978264C4A207446DAD
4E7292E2DCDA3024B47DA8C063AA1E6D
22FBD976AB0FE73D94D2D9C6D88127BC'

Étape 13. Déterminer l'algorithme de hachage par l'élément de fin T* :

T = 'BC' (c.-à-d. SHA-1)

Étape 14. Extraire le condensé :

D = 'C063AA1E6D22FBD976AB0FE73D94D2D9
C6D88127'

Étape 15. Extraire M_1 :

M_1 = '9D2784A67F8E7C659973EA1AEA25D95B
6C8F91E5002F369F0FBDCE8A3CEC1991
B543F1696546C5524CF23A5303CD6C98
599F40B79F377B5F3A1406B3B4D8F967
84D23AA88DB7E1032A405E69325FA91A
6E86F5C71AEA978264C4A207446DAD4E
7292E2DCDA3024B47DA8'

Étape 16. L'en-tête indique un rétablissement partiel mais la signature a la longueur du module, donc concaténer M_1 avec M_2 connu (c'est-à-dire RND.IFD) :

M* = '9D2784A67F8E7C659973EA1AEA25D95B
6C8F91E5002F369F0FBDCE8A3CEC1991
B543F1696546C5524CF23A5303CD6C98
599F40B79F377B5F3A1406B3B4D8F967
84D23AA88DB7E1032A405E69325FA91A
6E86F5C71AEA978264C4A207446DAD4E
7292E2DCDA3024B47DA8F173589974BF
40C6'

Étape 17. Calculer le condensé SHA-1 de M* :

D* = 'C063AA1E6D22FBD976AB0FE73D94D2D9
C6D88127'

Étape 18. Comparer D et D* :

D est égal à D*, donc la vérification est réussie.

— — — — — — — —

Appendice G à la Partie 11
(INFORMATIF)

EXEMPLE DÉTAILLÉ : PACE — MAPPAGE GÉNÉRIQUE

Le présent appendice donne deux exemples détaillés du protocole PACE défini au § 4.4 en utilisant le mappage générique. Le premier exemple est basé sur ECDH et le second utilise DH. Tous les nombres figurant dans les tableaux sont notés en hexadécimal.

Dans les deux exemples, la ZLA est utilisée comme mot de passe, ce qui conduit à la même clé symétrique K_π. Les champs de données pertinents de la ZLA, y compris les chiffres de contrôle, sont :

- Numéro de document : T220001293 ;
- Date de naissance : 6408125 ;
- Date d'expiration : 1010318.

Le codage de K de la ZLA et la clé de chiffrement calculée K_π sont donc :

K	`7E2D2A41 C74EA0B3 8CD36F86 3939BFA8 E9032AAD`
K_π	`89DED1B2 6624EC1E 634C1989 302849DD`

G.1 EXEMPLE BASÉ SUR ECDH

Cet exemple est basé sur ECDH en appliquant les paramètres de domaine normalisés BrainpoolP256r1 (voir RFC 5639).

La première section présente la structure de données `PACEInfo` correspondante. Les APDU échangées, y compris tous les nonces et toutes les clés éphémères générés, sont ensuite indiquées et examinées.

Paramètres de courbe elliptique

En utilisant les paramètres de domaine normalisés, toutes les informations requises pour exécuter PACE sont données par la structure de données `PACEInfo`. Notamment, aucune structure `PACEDomainParameterInfo` n'est nécessaire.

PACEInfo	`3012060A 04007F00 07020204 02020201 0202010D`

La structure de `PACEInfo` est présentée en détail dans le tableau suivant :

Étiquette	*Longueur*	*Valeur*	*Type ASN.1*	*Observation*
30	12		SEQUENCE	PACEInfo
06	0A	04 00 7F 00 07 02 02 04 02 02	OBJECT IDENTIFIER	PACE avec ECDH, mappage générique et clés de session AES 128
02	01	02	INTEGER	Version 2
02	01	0D	INTEGER	Paramètres de domaine normalisés Brainpool P256r1

Pour faciliter la présentation de l'exemple, un codage ASN.1 des paramètres de domaine BrainpoolP256r1 est indiqué ci-dessous.

Étiquette	*Longueur*	*Valeur*	*Type ASN.1*	*Observation*
30	81 EC		SEQUENCE	Paramètre de domaine
06	07	2A 86 48 CE 3D 02 01	OBJECT IDENTIFIER	Algorithme id-ecPublicKey
30	81 E0		SEQUENCE	Paramètre de domaine
02	01	01	INTEGER	Version
30	2C		SEQUENCE	Champ sous-jacent
06	07	2A 86 48 CE 3D 01 01	OBJECT IDENTIFIER	Champ principal
02	21	00 A9 FB 57 DB A1 EE A9 BC 3E 66 0A 90 9D 83 8D 72 6E 3B F6 23 D5 26 20 28 20 13 48 1D 1F 6E 53 77	INTEGER	Nombre premier p
30	44		SEQUENCE	Équation de la courbe
04	20	7D 5A 09 75 FC 2C 30 57 EE F6 75 30 41 7A FF E7 FB 80 55 C1	OCTET STRING (chaîne d'octets)	Paramètre a
04	20	26 DC 5C 6C E9 4A 4B 44 F3 30 B5 D9 BB D7 7C BF 95 84 16 29 5C F7 E1 CE 6B CC DC 18 FF 8C 07 B6	OCTET STRING	Paramètre b

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***
04	41		OCTET STRING (chaîne d'octets)	Générateur de groupe G
		04	–	Point non compressé
		8B D2 AE B9 CB 7E 57 CB 2C 4B 48 2F FC 81 B7 AF B9 DE 27 E1 E3 BD 23 C2 3A 44 53 BD 9A CE 32 62	–	Coordonnée x
		54 7E F8 35 C3 DA C4 FD 97 F8 46 1A 14 61 1D C9 C2 77 45 13 2D ED 8E 54 5C 1D 54 C7 2F 04 69 97	–	Coordonnée y
02	21	00 A9 FB 57 DB A1 EE A9 BC 3E 66 0A 90 9D 83 8D 71 8C 39 7A A3 B5 61 A6 F7 90 1E 0E 82 97 48 56 A7	INTEGER	Ordre de groupe n
02	01	01	INTEGER	Cofacteur f

Flux d'application de l'exemple basé sur ECDH

Pour initialiser PACE, le terminal envoie la commande MSE:Set AT à la puce.

T>C :	`00 22 C1 A4 0F 80 0A 04 00 7F 00 07 02 02 04 02 02 83 01 01`
C>T :	`90 00`

Dans ce cas-ci, T>C désigne une APDU envoyée du terminal à la puce, et C>T désigne la réponse correspondante envoyée par la puce au terminal. Le tableau ci-après explique le codage de la commande.

Commande				
CLA	00		En clair	
INS	22		Gestion de l'environnement de sécurité	
P1/P2	C1 A4		Mettre le gabarit d'authentification à authentification mutuelle.	
Lc	0F		Longueur du champ de données	
Données	Étiquette	Longueur	Valeur	Observation
	80	0A	04 00 7F 00 07 02 02 04 02 02	Mécanisme cryptographique : PACE avec ECDH, mappage générique et clés de session AES 128
	83	01	01	Mot de passe : ZLA
Réponse				
Octets d'état	90 00		Traitement normal	

Nonce chiffré

La puce génère ensuite aléatoirement le nonce s et le chiffre au moyen de K_{π}.

Nonce déchiffré s	3F00C4D3 9D153F2B 2A214A07 8D899B22
Nonce chiffré z	95A3A016 522EE98D 01E76CB6 B98B42C3

Le terminal demande le nonce chiffré.

T>C :	10 86 00 00 02 7C 00 00
C>T :	7C 12 80 10 95 A3 A0 16 52 2E E9 8D 01 E7 6C B6 B9 8B 42 C3 90 00

Le codage de l'APDU commande et de la réponse correspondante est décrit dans le tableau suivant :

Commande				
CLA	10		Chaînage des commandes	
INS	86		AUTHENTIFICATION GÉNÉRALE	
P1/P2	00 00		Clés et protocole connus implicitement	
Lc	02		Longueur des données	
Données	Étiquette	Longueur	Valeur	Observation
	7C	00	–	Absentes
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.	
Réponse				
Données	Étiquette	Longueur	Valeur	Observation
	7C	12		Données d'authentification dynamique
	80	10	95 A3 A0 16 52 2E E9 8D 01 E7 6C B6 B9 8B 42 C3	Nonce chiffré
Octets d'état	90 00		Traitement normal	

Mappage du nonce

Le nonce est mappé sur un générateur de groupe éphémère en utilisant le mappage générique. Les clés éphémères requises choisies aléatoirement sont aussi recueillies dans le tableau suivant :

Clé privée du terminal	7F4EF07B 9EA82FD7 8AD689B3 8D0BC78C F21F249D 953BC46F 4C6E1925 9C010F99
Clé publique du terminal	7ACF3EFC 982EC455 65A4B155 129EFBC7 4650DCBF A6362D89 6FC70262 E0C2CC5E, 544552DC B6725218 799115B5 5C9BAA6D 9F6BC3A9 618E70C2 5AF71777 A9C4922D
Clé privée de la puce	498FF497 56F2DC15 87840041 839A8598 2BE7761D 14715FB0 91EFA7BC E9058560
Clé publique de la puce	824FBA91 C9CBE26B EF53A0EB E7342A3B F178CEA9 F45DE0B7 0AA60165 1FBA3F57, 30D8C879 AAA9C9F7 3991E61B 58F4D52E B87A0A0C 709A49DC 63719363 CCD13C54
Secret partagé H	60332EF2 450B5D24 7EF6D386 8397D398 852ED6E8 CAF6FFEE F6BF85CA 57057FD5, 0840CA74 15BAF3E4 3BD414D3 5AA4608B 93A2CAF3 A4E3EA4E 82C9C13D 03EB7181
Générateur Ĝ mappé	8CED63C9 1426D4F0 EB1435E7 CB1D74A4 6723A0AF 21C89634 F65A9AE8 7A9265E2, 8C879506 743F8611 AC33645C 5B985C80 B5F09A0B 83407C1B 6A4D857A E76FE522

Le terminal et la puce échangent les APDU suivantes pour mapper le nonce.

T>C :	10 86 00 00 45 7C 43 81 41 04 7A CF 3E FC 98 2E C4 55 65 A4 B1 55 12 9E FB C7 46 50 DC BF A6 36 2D 89 6F C7 02 62 E0 C2 CC 5E 54 45 52 DC B6 72 52 18 79 91 15 B5 5C 9B AA 6D 9F 6B C3 A9 61 8E 70 C2 5A F7 17 77 A9 C4 92 2D 00
C>T :	7C 43 82 41 04 82 4F BA 91 C9 CB E2 6B EF 53 A0 EB E7 34 2A 3B F1 78 CE A9 F4 5D E0 B7 0A A6 01 65 1F BA 3F 57 30 D8 C8 79 AA A9 C9 F7 39 91 E6 1B 58 F4 D5 2E B8 7A 0A 0C 70 9A 49 DC 63 71 93 63 CC D1 3C 54 90 00

La structure des APDU peut être décrite comme suit :

Commande					
CLA	10		Chaînage des commandes		
INS	86		AUTHENTIFICATION GÉNÉRALE		
P1/P2	00 00		Clés et protocole connus implicitement		
Lc	45		Longueur des données		
Données	Étiquette	Longueur	Valeur	Observation	
	7C	43	–	Données d'authentification dynamique	
	81	41		Données de mappage	
			04		Point non compressé
			7A CF 3E FC 98 2E ... C2 CC 5E		Coordonnée x
			54 45 52 DC B6 72 ... C4 92 2D		Coordonnée y
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.		
Réponse					
Données	Étiquette	Longueur	Valeur	Observation	
	7C	43		Données d'authentification dynamique	
	82	41		Données de mappage	
			04		Point non compressé
			82 4F BA 91 C9 CB ... BA 3F 57		Coordonnée x
			30 D8 C8 79 AA A9 ... D1 3C 54		Coordonnée y
Octets d'état	90 00		Traitement normal		

Exécution de l'agrément de clé

Durant la troisième étape, la puce et le terminal exécutent un agrément de clé ECDH anonyme en utilisant les nouveaux paramètres de domaine déterminés par le générateur de groupe éphémère de l'étape précédente. Seule la coordonnée x est requise comme secret partagé vu que KDF n'utilise que la première coordonnée pour calculer les clés de session.

Clé privée du terminal	A73FB703 AC1436A1 8E0CFA5A BB3F7BEC 7A070E7A 6788486B EE230C4A 22762595
Clé publique du terminal	2DB7A64C 0355044E C9DF1905 14C625CB A2CEA487 54887122 F3A5EF0D 5EDD301C, 3556F3B3 B186DF10 B857B58F 6A7EB80F 20BA5DC7 BE1D43D9 BF850149 FBB36462
Clé privée de la puce	107CF586 96EF6155 053340FD 633392BA 81909DF7 B9706F22 6F32086C 7AFF974A
Clé publique de la puce	9E880F84 2905B8B3 181F7AF7 CAA9F0EF B743847F 44A306D2 D28C1D9E C65DF6DB, 7764B222 77A2EDDC 3C265A9F 018F9CB8 52E111B7 68B32690 4B59A019 3776F094
Secret partagé	28768D20 701247DA E81804C9 E780EDE5 82A9996D B4A31502 0B273319 7DB84925

L'agrément de clé est exécuté comme suit :

T>C :	10 86 00 00 45 7C 43 83 41 04 2D B7 A6 4C 03 55 04 4E C9 DF 19 05 14 C6 25 CB A2 CE A4 87 54 88 71 22 F3 A5 EF 0D 5E DD 30 1C 35 56 F3 B3 B1 86 DF 10 B8 57 B5 8F 6A 7E B8 0F 20 BA 5D C7 BE 1D 43 D9 BF 85 01 49 FB B3 64 62 00
C>T :	7C 43 84 41 04 9E 88 0F 84 29 05 B8 B3 18 1F 7A F7 CA A9 F0 EF B7 43 84 7F 44 A3 06 D2 D2 8C 1D 9E C6 5D F6 DB 77 64 B2 22 77 A2 ED DC 3C 26 5A 9F 01 8F 9C B8 52 E1 11 B7 68 B3 26 90 4B 59 A0 19 37 76 F0 94 90 00

Le codage de l'agrément de clé est analysé dans le tableau ci-après :

Commande		
CLA	10	Chaînage des commandes
INS	86	AUTHENTIFICATION GÉNÉRALE
P1/P2	00 00	Clés et protocole connus implicitement
Lc	45	Longueur des données

Commande					
Données	Étiquette	Longueur	Valeur	Observation	
	7C	43	–	Données d'authentification dynamique	
	83	41		Clé publique éphémère du terminal	
			04		Point non compressé
			2D B7 A6 4C 03 55 ... DD 30 1C		Coordonnée x
			35 56 F3 B3 B1 86 ... B3 64 62		Coordonnée y
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.		
Réponse					
Données	Étiquette	Longueur	Valeur	Observation	
	7C	43		Données d'authentification dynamique	
	84	41		Clé publique éphémère de la puce	
			04		Point non compressé
			9E 88 0F 84 29 05 ... 5D F6 DB		Coordonnée x
			77 64 B2 22 77 A2 ... 76 F0 94		Coordonnée y
Octets d'état	90 00		Traitement normal		

Les clés de session AES 128 KS_{Enc} et KS_{MAC} suivantes sont calculées à partir du secret partagé en utilisant KDF. Notamment :

KS_{Enc}	F5F0E35C 0D7161EE 6724EE51 3A0D9A7F
KS_{MAC}	FE251C78 58B356B2 4514B3BD 5F4297D1

Authentification mutuelle

Les jetons d'authentification sont calculés au moyen de KS_{MAC} en utilisant :

Données d'entrée pour T_{IFD}	7F494F06 0A04007F 00070202 04020286 41049E88 0F842905 B8B3181F 7AF7CAA9 F0EFB743 847F44A3 06D2D28C 1D9EC65D F6DB7764 B22277A2 EDDC3C26 5A9F018F 9CB852E1 11B768B3 26904B59 A0193776 F094

Données d'entrée pour T_{IC}	7F494F06 0A04007F 00070202 04020286 41042DB7 A64C0355 044EC9DF 190514C6 25CBA2CE A4875488 7122F3A5 EF0D5EDD 301C3556 F3B3B186 DF10B857 B58F6A7E B80F20BA 5DC7BE1D 43D9BF85 0149FBB3 6462

comme entrée. Le codage des données d'entrée est indiqué ci-dessous :

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***
7F49	4F		PUBLIC KEY	Données d'entrée pour T_{IFD}
06	0A	04 00 7F 00 07 02 02 04 02 02	OBJECT IDENTIFIER	PACE avec ECDH, mappage générique et clés de session AES 128
86	41		ELLIPTIC CURVE POINT (point de courbe elliptique)	Point public éphémère de la puce
		04		Point non compressé
		9E 88 0F 84 29 ... 5D F6 DB		Coordonnée x
		77 64 B2 22 77 ... 76 F0 94		Coordonnée y

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***
7F49	4F		PUBLIC KEY	Données d'entrée pour T_{IC}
06	0A	04 00 7F 00 07 02 02 04 02 02	OBJECT IDENTIFIER	PACE avec ECDH, mappage générique et clés de session AES 128
86	41		ELLIPTIC CURVE POINT	Point public éphémère du terminal
		04		Point non compressé
		2D B7 A6 4C 03 ... DD 30 1C		Coordonnée x
		35 56 F3 B3 B1 ... B3 64 62		Coordonnée y

Les jetons d'authentification calculés sont :

T_{IFD}	C2B0BD78 D94BA866
T_{IC}	3ABB9674 BCE93C08

En dernier lieu, ces jetons sont échangés et vérifiés.

T>C :	00 86 00 00 0C 7C 0A 85 08 C2 B0 BD 78 D9 4B A8 66 00
C>T :	7C 0A 86 08 3A BB 96 74 BC E9 3C 08 90 00

G.2 EXEMPLE BASÉ SUR DH

Le second exemple est basé sur DH en utilisant le groupe MODP 1 024 bits avec un sous-groupe d'ordre premier de 160 bits spécifié par RFC 5114. Les paramètres du groupe sont :

Nombre premier *p*	B10B8F96 A080E01D DE92DE5E AE5D54EC 52C99FBC FB06A3C6 9A6A9DCA 52D23B61 6073E286 75A23D18 9838EF1E 2EE652C0 13ECB4AE A9061123 24975C3C D49B83BF ACCBDD7D 90C4BD70 98488E9C 219A7372 4EFFD6FA E5644738 FAA31A4F F55BCCC0 A151AF5F 0DC8B4BD 45BF37DF 365C1A65 E68CFDA7 6D4DA708 DF1FB2BC 2E4A4371
Générateur de sous-groupe *g*	A4D1CBD5 C3FD3412 6765A442 EFB99905 F8104DD2 58AC507F D6406CFF 14266D31 266FEA1E 5C41564B 777E690F 5504F213 160217B4 B01B886A 5E91547F 9E2749F4 D7FBD7D3 B9A92EE1 909D0D22 63F80A76 A6A24C08 7A091F53 1DBF0A01 69B6A28A D662A4D1 8E73AFA3 2D779D59 18D08BC8 858F4DCE F97C2A24 855E6EEB 22B3B2E5
Ordre premier *q* de *g*	F518AA87 81A8DF27 8ABA4E7D 64B7CB9D 49462353

La première section introduit la structure `PACEInfo`. Les APDU échangées, y compris tous les nonces et toutes les clés éphémères générés, sont énumérées et examinées.

Paramètres Diffie-Hellman

Les informations applicables à PACE sont données par la structure `PACEInfo`.

PACEInfo	3012060A 04007F00 07020204 01020201 02020100

La structure détaillée de `PACEInfo` est :

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***	
30	12		SEQUENCE	PACEInfo	
06	0A	04 00 7F 00 07 02 02 04 01 02	OBJECT IDENTIFIER		OID : PACE avec DH, mappage générique et clés de session AES 128
02	01	02	INTEGER		Version 2
02	01	00	INTEGER		Groupe de 1 024 bits normalisé spécifié par RFC 5114

Flux d'application de l'exemple basé sur DH

Pour initialiser PACE, le terminal envoie la commande MSE:AT à la puce.

T>C :	`00 22 C1 A4 0F 80 0A 04 00 7F 00 07 02 02 04 01 02 83 01 01`
C>T :	`90 00`

Le codage de la commande est décrit dans le tableau suivant :

Commande				
CLA	00		En clair	
INS	22		Gestion de l'environnement de sécurité	
P1/P2	C1 A4		Mettre le gabarit d'authentification à authentification mutuelle.	
Lc	0F		Longueur du champ de données	
Données	Étiquette	Longueur	Valeur	Observation
	80	0A	04 00 7F 00 07 02 02 04 01 02	OID : mécanisme cryptographique : PACE avec DH, mappage générique et AES 128
	83	01	01	Mot de passe : ZLA
Réponse				
Octets d'état	90 00		Traitement normal	

Nonce chiffré

Le terminal demande ensuite un nonce de la puce.

Nonce déchiffré s	`FA5B7E3E 49753A0D B9178B7B 9BD898C8`
Nonce chiffré z	`854D8DF5 827FA685 2D1A4FA7 01CDDDCA`

La communication se présente comme suit :

T>C :	10 86 00 00 02 7C 00 00
C>T :	7C 12 80 10 85 4D 8D F5 82 7F A6 85 2D 1A 4F A7 01 CD DD CA 90 00

Le codage de l'APDU commande et de la réponse correspondante est décrit dans le tableau suivant :

Commande				
CLA	10		Chaînage des commandes	
INS	86		AUTHENTIFICATION GÉNÉRALE	
P1/P2	00 00		Clés et protocole connus implicitement	
Lc	02		Longueur des données	
Données	Étiquette	Longueur	Valeur	Observation
	7C	00	–	Absentes
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.	
Réponse				
Données	Étiquette	Longueur	Valeur	Observation
	7C	12		Données d'authentification dynamique
	80	10	85 4D 8D F5 82 7F A6 85 2D 1A 4F A7 01 CD DD CA	Nonce chiffré
Octets d'état	90 00		Traitement normal	

Mappage du nonce

Le nonce est mappé sur un générateur de groupe éphémère au moyen du mappage générique. Les clés éphémères suivantes sont générées aléatoirement par le terminal et la puce dans ce but.

Clé privée du terminal	5265030F 751F4AD1 8B08AC56 5FC7AC95 2E41618D
Clé publique du terminal	23FB3749 EA030D2A 25B278D2 A562047A DE3F01B7 4F17A154 02CB7352 CA7D2B3E B71C343D B13D1DEB CE9A3666 DBCFC920 B49174A6 02CB4796 5CAA73DC 702489A4 4D41DB91 4DE9613D C5E98C94 160551C0 DF86274B 9359BC04 90D01B03 AD54022D CB4F57FA D6322497 D7A1E28D 46710F46 1AFE710F BBBC5F8B A166F431 1975EC6C
Clé privée de la puce	66DDAFEA C1609CB5 B963BB0C B3FF8B3E 047F336C
Clé publique de la puce	78879F57 225AA808 0D52ED0F C890A4B2 5336F699 AA89A2D3 A189654A F70729E6 23EA5738 B26381E4 DA19E004 706FACE7 B235C2DB F2F38748 312F3C98 C2DD4882 A41947B3 24AA1259 AC22579D B93F7085 655AF308 89DBB845 D9E6783F E42C9F24 49400306 254C8AE8 EE9DD812 A804C0B6 6E8CAFC1 4F84D825 8950A91B 44126EE6
Secret partagé H	5BABEBEF 5B74E5BA 94B5C063 FDA15F1F 1CDE9487 3EE0A5D3 A2FCAB49 F258D07F 544F13CB 66658C3A FEE9E727 389BE3F6 CBBBD321 28A8C21D D6EEA3CF 7091CDDF B08B8D00 7D40318D CCA4FFBF 51208790 FB4BD111 E5A968ED 6B6F08B2 6CA87C41 0B3CE0C3 10CE104E ABD16629 AA48620C 1279270C B0750C0D 37C57FFF E302AE7F
Générateur Ĝ mappé	7C9CBFE9 8F9FBDDA 8D143506 FA7D9306 F4CB17E3 C71707AF F5E1C1A1 23702496 84D64EE3 7AF44B8D BD9D45BF 6023919C BAA027AB 97ACC771 666C8E98 FF483301 BFA4872D EDE9034E DFACB708 14166B7F 36067682 9B826BEA 57291B5A D69FBC84 EF1E7790 32A30580 3F743417 93E86974 2D401325 B37EE856 5FFCDEE6 18342DC5

Le terminal et la puce échangent les APDU suivantes pour mapper le nonce.

T>C :	10 86 00 00 86 7C 81 83 81 81 80 23 FB 37 49 EA 03 0D 2A 25 B2 78 D2 A5 62 04 7A DE 3F 01 B7 4F 17 A1 54 02 CB 73 52 CA 7D 2B 3E B7 1C 34 3D B1 3D 1D EB CE 9A 36 66 DB CF C9 20 B4 91 74 A6 02 CB 47 96 5C AA 73 DC 70 24 89 A4 4D 41 DB 91 4D E9 61 3D C5 E9 8C 94 16 05 51 C0 DF 86 27 4B 93 59 BC 04 90 D0 1B 03 AD 54 02 2D CB 4F 57 FA D6 32 24 97 D7 A1 E2 8D 46 71 0F 46 1A FE 71 0F BB BC 5F 8B A1 66 F4 31 19 75 EC 6C 00
C>T :	7C 81 83 82 81 80 78 87 9F 57 22 5A A8 08 0D 52 ED 0F C8 90 A4 B2 53 36 F6 99 AA 89 A2 D3 A1 89 65 4A F7 07 29 E6 23 EA 57 38 B2 63 81 E4 DA 1 9E0 04 70 6F AC E7 B2 35 C2 DB F2 F3 87 48 31 2F 3C 98 C2 DD 48 82 A4 19 47 B3 24 AA 12 59 AC 22 57 9D B9 3F 70 85 65 5A F3 08 89 DB B8 45 D9 E6 78 3F E4 2C 9F 24 49 40 03 06 25 4C 8A E8 EE 9D D8 12 A8 04 C0 B6 6E 8C AF C1 4F 84 D8 25 89 50 A9 1B 44 12 6E E6 90 00

La structure des APDU peut être décrite comme suit :

Commande				
CLA	10		Chaînage des commandes	
INS	86		AUTHENTIFICATION GÉNÉRALE	
P1/P2	00 00		Clés et protocole connus implicitement	
Lc	86		Longueur des données	
Données	Étiquette	Longueur	Valeur	Observation
	7C	81 83	–	Données d'authentification dynamique
	81	81 80	23 FB 37 49 EA 03 ... 75 EC 6C	Données de mappage
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.	
Réponse				
Données	Étiquette	Longueur	Valeur	Observation
	7C	81 83		Données d'authentification dynamique
	82	81 80	ED 0F C8 90 A4 B2 ... 12 6E E6	Données de mappage
Octets d'état	90 00		Traitement normal	

Exécution de l'agrément de clé

La puce et le terminal exécutent ensuite un agrément de clé DH anonyme en utilisant les nouveaux paramètres de domaine déterminés par le générateur de groupe éphémère de l'étape précédente.

Clé privée du terminal	89CCD99B 0E8D3B1F 11E1296D CA68EC53 411CF2CA
Clé publique du terminal	00907D89 E2D425A1 78AA81AF 4A7774EC 8E388C11 5CAE6703 1E85EECE 520BD911 551B9AE4 D04369F2 9A02626C 86FBC674 7CC7BC35 2645B616 1A2A42D4 4EDA80A0 8FA8D61B 76D3A154 AD8A5A51 786B0BC0 71470578 71A92221 2C5F67F4 31731722 36B7747D 1671E6D6 92A3C7D4 0A0C3C5C E397545D 015C175E B5130551 EDBC2EE5 D4
Clé privée de la puce	A5B78012 6B7C980E 9FCEA1D4 539DA1D2 7C342DFA
Clé publique de la puce	075693D9 AE941877 573E634B 6E644F8E 60AF17A0 076B8B12 3D920107 4D36152B D8B3A213 F53820C4 2ADC79AB 5D0AEEC3 AEFB9139 4DA476BD 97B9B14D 0A65C1FC 71A0E019 CB08AF55 E1F72900 5FBA7E3F A5DC4189 9238A250 767A6D46 DB974064 386CD456 743585F8 E5D90CC8 B4004B1F 6D866C79 CE0584E4 9687FF61 BC29AEA1
Secret partagé	6BABC7B3 A72BCD7E A385E4C6 2DB2625B D8613B24 149E146A 629311C4 CA6698E3 8B834B6A 9E9CD718 4BA8834A FF5043D4 36950C4C 1E783236 7C10CB8C 314D40E5 990B0DF7 013E64B4 549E2270 923D06F0 8CFF6BD3 E977DDE6 ABE4C31D 55C0FA2E 465E553E 77BDF75E 3193D383 4FC26E8E B1EE2FA1 E4FC97C1 8C3F6CFF FE2607FD

L'agrément de clé est exécuté comme suit :

T>C :	10 86 00 00 86 7C 81 83 83 81 80 90 7D 89 E2 D4 25 A1 78 AA 81 AF 4A 77 74 EC 8E 38 8C 11 5C AE 67 03 1E 85 EE CE 52 0B D9 11 55 1B 9A E4 D0 43 69 F2 9A 02 62 6C 86 FB C6 74 7C C7 BC 35 26 45 B6 16 1A 2A 42 D4 4E DA 80 A0 8F A8 D6 1B 76 D3 A1 54 AD 8A 5A 51 78 6B 0B C0 71 47 05 78 71 A9 22 21 2C 5F 67 F4 31 73 17 22 36 B7 74 7D 16 71 E6 D6 92 A3 C7 D4 0A 0C 3C 5C E3 97 54 5D 01 5C 17 5E B5 13 05 51 ED BC 2E E5 D4 00
C>T :	7C 81 83 84 81 80 07 56 93 D9 AE 94 18 77 57 3E 63 4B 6E 64 4F 8E 60 AF 17 A0 07 6B 8B 12 3D 92 01 07 4D 36 15 2B D8 B3 A2 13 F5 38 20 C4 2A DC 79 AB 5D 0A EE C3 AE FB 91 39 4D A4 76 BD 97 B9 B1 4D 0A 65 C1 FC 71 A0 E0 19 CB 08 AF 55 E1 F7 29 00 5F BA 7E 3F A5 DC 41 89 92 38 A2 50 76 7A 6D 46 DB 97 40 64 38 6C D4 56 74 35 85 F8 E5 D9 0C C8 B4 00 4B 1F 6D 86 6C 79 CE 05 84 E4 96 87 FF 61 BC 29 AE A1 90 00

Commande				
CLA	10		Chaînage des commandes	
INS	86		AUTHENTIFICATION GÉNÉRALE	
P1/P2	00 00		Clés et protocole connus implicitement	
Lc	86		Longueur des données	
Données	Étiquette	Longueur	Valeur	Observation
	7C	81 83	–	Données d'authentification dynamique
	83	81 80	90 7D 89 E2 D4 25 ... 2E E5 D4	Clé publique éphémère du terminal
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.	
Réponse				
Données	Étiquette	Longueur	Valeur	Observation
	7C	81 83		Données d'authentification dynamique
	84	81 80	07 56 93 D9 AE 94 ... 29 AE A1	Clé publique éphémère de la puce
Octets d'état	90 00		Traitement normal	

Les clés de session AES 128 KS_{Enc} et KS_{MAC} sont calculées à partir du secret partagé en utilisant KDF.

KS_{Enc}	2F7F46AD CC9E7E52 1B45D192 FAFA9126
KS_{MAC}	805A1D27 D45A5116 F73C5446 9462B7D8

Authentification mutuelle

Les jetons d'authentification sont construits à partir des données d'entrée suivantes :

Données d'entrée pour T_{IFD}	7F49818F 060A0400 7F000702 02040102 84818007 5693D9AE 94187757 3E634B6E 644F8E60 AF17A007 6B8B123D 9201074D 36152BD8 B3A213F5 3820C42A DC79AB5D 0AEEC3AE FB91394D A476BD97 B9B14D0A 65C1FC71 A0E019CB 08AF55E1 F729005F BA7E3FA5 DC418992 38A25076 7A6D46DB 97406438 6CD45674 3585F8E5 D90CC8B4 004B1F6D 866C79CE 0584E496 87FF61BC 29AEA1

Données d'entrée pour T_{IC}	7F49818F 060A0400 7F000702 02040102 84818090 7D89E2D4 25A178AA 81AF4A77 74EC8E38 8C115CAE 67031E85 EECE520B D911551B 9AE4D043 69F29A02 626C86FB C6747CC7 BC352645 B6161A2A 42D44EDA 80A08FA8 D61B76D3 A154AD8A 5A51786B 0BC07147 057871A9 22212C5F 67F43173 172236B7 747D1671 E6D692A3 C7D40A0C 3C5CE397 545D015C 175EB513 0551EDBC 2EE5D4

Le codage des données d'entrée est indiqué ci-dessous :

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***	
7F49	81 8F		PUBLIC KEY	Données d'entrée pour T_{IFD}	
06	0A	04 00 7F 00 07 02 02 04 01 02	OBJECT IDENTIFIER		PACE avec DH, mappage générique et clés de session AES 128
84	81 80	07 56 93 D9 AE ... 29 AE A1	UNSIGNED INTEGER (entier non signé)		Clé publique éphémère de la puce

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***	
7F49	81 8F		PUBLIC KEY	Données d'entrée pour T_{IC}	
06	0A	04 00 7F 00 07 02 02 04 01 02	OBJECT IDENTIFIER		PACE avec DH, mappage générique et clés de session AES 128
84	81 80	90 7D 89 E2 D4 ... 2E E5 D4	UNSIGNED INTEGER		Clé publique éphémère du terminal

Les jetons d'authentification calculés sont :

T_{IFD}	B46DD9BD 4D98381F
T_{IC}	917F37B5 C0E6D8D1

Ces jetons sont ensuite échangés et vérifiés.

T>C :	00 86 00 00 0C 7C 0A 85 08 B4 6D D9 BD 4D 98 38 1F 00
C>T :	7C 1B 86 08 91 7F 37 B5 C0 E6 D8 D1 87 0F 44 45 54 45 53 54 43 56 43 41 30 30 30 30 33

Commande				
CLA	00		En clair	
INS	86		AUTHENTIFICATION GÉNÉRALE	
P1/P2	00 00		Clés et protocole connus implicitement	
Lc	0C		Longueur des données	
Données	Étiquette	Longueur	Valeur	Observation
	7C	0A	–	Données d'authentification dynamique
	85	08	B4 6D D9 BD 4D 98 38 1F	Jeton d'authentification du terminal
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.	
Réponse				
Données	Étiquette	Longueur	Valeur	Observation
	7C	0A		Données d'authentification dynamique
	86	08	91 7F 37 B5 C0 E6 D8 D1	Jeton d'authentification de la puce
Octets d'état	90 00		Traitement normal	

— — — — — — — —

Appendice H à la Partie 11
(INFORMATIF)

EXEMPLE DÉTAILLÉ : PACE — MAPPAGE INTÉGRÉ

Le présent appendice donne deux exemples du protocole PACE avec mappage intégré. Le premier est basé sur ECDH et le second sur DH. La clé *K* calculée à partir de la ZLA de l'exemple précédent est utilisée.

H.1 EXEMPLE BASÉ SUR ECDH

Cet exemple est basé sur la courbe elliptique BrainpoolP256r1. Le chiffrement par blocs utilisé dans le présent exemple est AES-128. Les paramètres de la courbe sont les suivants :

Nombre premier p	A9FB57DB A1EEA9BC 3E660A90 9D838D72 6E3BF623 D5262028 2013481D 1F6E5377
Paramètre a	7D5A0975 FC2C3057 EEF67530 417AFFE7 FB8055C1 26DC5C6C E94A4B44 F330B5D9
Paramètre b	26DC5C6C E94A4B44 F330B5D9 BBD77CBF 95841629 5CF7E1CE 6BCCDC18 FF8C07B6
Coordonnée x du générateur de groupe G	8BD2AEB9 CB7E57CB 2C4B482F FC81B7AF B9DE27E1 E3BD23C2 3A4453BD 9ACE3262
Coordonnée y du générateur de groupe G	547EF835 C3DAC4FD 97F8461A 14611DC9 C2774513 2DED8E54 5C1D54C7 2F046997
Ordre de groupe n	A9FB57DB A1EEA9BC 3E660A90 9D838D71 8C397AA3 B561A6F7 901E0E82 974856A7
Cofacteur f	01

La clé de chiffrement est la suivante :

K_{π}	591468CD A83D6521 9CCCB856 0233600F

Nonce chiffré

Un nonce s est choisi aléatoirement par la puce et chiffré en utilisant K_{TT}. Le nonce chiffré z est ensuite envoyé au terminal.

Nonce déchiffré s	2923BE84 E16CD6AE 529049F1 F1BBE9EB
Nonce chiffré z	143DC40C 08C8E891 FBED7DED B92B64AD

Mappage du nonce

Un nonce t est choisi aléatoirement et envoyé en clair. Les nonces t et s sont ensuite utilisés pour calculer le mappage intégré. D'abord, la fonction pseudo-aléatoire R_p, obtenue à partir d'AES, est appliquée à s et à t. Le codage de points f_G est ensuite appliqué au résultat pour calculer le générateur mappé $\hat{G}=f_G[R_p(s,t)]$.

Nonce t	5DD4CBFC 96F5453B 130D890A 1CDBAE32
R(s,t) pseudo-aléatoire	E4447E2D FB3586BA C05DDB00 156B57FB B2179A39 49294C97 25418980 0C517BAA 8DA0FF39 7ED8C445 D3E421E4 FEB57322
$R_p(s,t)$	A2F8FF2D F50E52C6 599F386A DCB595D2 29F6A167 ADE2BE5F 2C3296AD D5B7430E
Coordonnée x du générateur mappé $\hat{G}$	8E82D315 59ED0FDE 92A4D049 8ADD3C23 BABA94FB 77691E31 E90AEA77 FB17D427
Coordonnée y du générateur mappé $\hat{G}$	4C1AE14B D0C3DBAC 0C871B7F 36081693 64437CA3 0AC243A0 89D3F266 C1E60FAD

Exécution de l'agrément de clé

La puce et le terminal exécutent un agrément de clé Diffie-Hellman anonyme en utilisant leurs clés secrètes et le générateur mappé $\hat{G}$. Le secret partagé K est la coordonnée x de l'agrément.

Clé privée de la puce SK_{IC}	107CF586 96EF6155 053340FD 633392BA 81909DF7 B9706F22 6F32086C 7AFF974A
Clé publique de la puce PK_{IC}	67F78E5F 7F768608 2B293E8D 087E0569 16D0F74B C01A5F89 57D0DE45 691E51E8 932B69A9 62B52A09 85AD2C0A 271EE6A1 3A8ADDDC D1A3A994 B9DED257 F4D22753
Clé privée du terminal SK_{IFD}	A73FB703 AC1436A1 8E0CFA5A BB3F7BEC 7A070E7A 6788486B EE230C4A 22762595

Clé publique du terminal PK_{IFD}	89CBA23F FE96AA18 D824627C 3E934E54 A9FD0B87 A95D1471 DC1C0ABF DCD640D4 6755DE9B 7B778280 B6BEBD57 439ADFEB 0E21FD4E D6DF4257 8C13418A 59B34C37
Secret partagé K	4F150FDE 1D4F0E38 E95017B8 91BAE171 33A0DF45 B0D3E18B 60BA7BEA FDC2C713

En utilisant les spécifications de [1], les clés de session K_{Enc} et K_{MAC} sont calculées à partir de K en utilisant la fonction de hachage SHA-1 : K_{Enc}=SHA-1(K||0x00000001) et K_{MAC}=SHA-1(K||0x00000002). Ensuite, seuls les 16 premiers octets du condensé sont utilisés avec le résultat suivant :

K_{Enc}	0D3FEB33 251A6370 893D62AE 8DAAF51B
K_{MAC}	B01E89E3 D9E8719E 586B50B4 A7506E0B

Authentification mutuelle

Les jetons d'authentification sont calculés en appliquant un CMAC aux entrées suivantes avec la clé K_{MAC}.

Données d'entrée pour T_{IC}	7F494F06 0A04007F 00070202 04040286 410489CB A23FFE96 AA18D824 627C3E93 4E54A9FD 0B87A95D 1471DC1C 0ABFDCD6 40D46755 DE9B7B77 8280B6BE BD57439A DFEB0E21 FD4ED6DF 42578C13 418A59B3 4C37
Données d'entrée pour T_{IFD}	7F494F06 0A04007F 00070202 04040286 410467F7 8E5F7F76 86082B29 3E8D087E 056916D0 F74BC01A 5F8957D0 DE45691E 51E8932B 69A962B5 2A0985AD 2C0A271E E6A13A8A DDDCD1A3 A994B9DE D257F4D2 2753

Les jetons d'authentification correspondants sont :

T_{IC}	75D4D96E 8D5B0308
T_{IFD}	450F02B8 6F6A0909

H.2 EXEMPLE BASÉ SUR DH

Le présent exemple est basé sur le groupe MODP de 1 024 bits avec un sous-groupe d'ordre premier de 160 bits. Le chiffrement par blocs utilisé dans cet exemple est AES 128.

Les paramètres du groupe sont :

Nombre premier p	B10B8F96 A080E01D DE92DE5E AE5D54EC 52C99FBC FB06A3C6 9A6A9DCA 52D23B61 6073E286 75A23D18 9838EF1E 2EE652C0 13ECB4AE A9061123 24975C3C D49B83BF ACCBDD7D 90C4BD70 98488E9C 219A7372 4EFFD6FA E5644738 FAA31A4F F55BCCC0 A151AF5F 0DC8B4BD 45BF37DF 365C1A65 E68CFDA7 6D4DA708 DF1FB2BC 2E4A4371
Générateur de sous-groupe g	A4D1CBD5 C3FD3412 6765A442 EFB99905 F8104DD2 58AC507F D6406CFF 14266D31 266FEA1E 5C41564B 777E690F 5504F213 160217B4 B01B886A 5E91547F 9E2749F4 D7FBD7D3 B9A92EE1 909D0D22 63F80A76 A6A24C08 7A091F53 1DBF0A01 69B6A28A D662A4D1 8E73AFA3 2D779D59 18D08BC8 858F4DCE F97C2A24 855E6EEB 22B3B2E5
Ordre premier q de g	F518AA87 81A8DF27 8ABA4E7D 64B7CB9D 49462353

La clé de chiffrement suivante est utilisée :

K_π	591468CD A83D6521 9CCCB856 0233600F

Nonce chiffré

Un nonce s est choisi aléatoirement par la puce et chiffré à l'aide de K_π. Le nonce chiffré z est ensuite envoyé au terminal.

Nonce déchiffré s	FA5B7E3E 49753A0D B9178B7B 9BD898C8
Nonce chiffré z	9ABB8864 CA0FF155 1E620D1E F4E13510

Mappage du nonce

Un nonce t est choisi aléatoirement et envoyé en clair. Les nonces t et s sont ensuite utilisés pour calculer le mappage intégré. D'abord, la fonction pseudo-aléatoire R_p, obtenue à partir d'AES, est appliquée à s et à t. Le codage de points f_g est ensuite appliqué au résultat.

Nonce t	B3A6DB3C 870C3E99 245E0D1C 06B747DE
R(s,t) pseudo-aléatoire	EAB98D13 E0905295 2AA72990 7C3C9461 84DEA0FE 74AD2B3A F506F0A8 3018459C 38099CD1 F7FF4EA0 A078DB1F AC136550 5E3DC855 00EF95E2 0B4EEF2E 88489233 BEE0546B 472F994B 618D1687 02406791 DEEF3CB4 810932EC 278F3533 FDB860EB 4835C36F A4F1BF3F A0B828A7 18C96BDE 88FBA38A 3E6C35AA A1095925 1EB5FC71 0FC18725 8995944C 0F926E24 9373F485
$R_p(s,t)$	A0C7C50C 002061A5 1CC87D25 4EF38068 607417B6 EE1B3647 3CFB800D 2D2E5FA2 B6980F01 105D24FA B22ACD1B FA5C8A4C 093ECDFA FE6D7125 D42A843E 33860383 5CF19AFA FF75EFE2 1DC5F6AA 1F9AE46C 25087E73 68166FB0 8C1E4627 AFED7D93 570417B7 90FF7F74 7E57F432 B04E1236 819E0DFE F5B6E77C A4999925 328182D2
Générateur mappé $\hat{g}= f_g[R_p(s,t)]$	1D7D767F 11E333BC D6DBAEF4 0E799E7A 926B9697 3550656F F3C83072 6D118D61 C276CDCC 61D475CF 03A98E0C 0E79CAEB A5BE2557 8BD4551D 0B109032 36F0B0F9 76852FA7 8EEA14EA 0ACA87D1 E91F688F E0DFF897 BBE35A47 2621D343 564B262F 34223AE8 FC59B664 BFEDFA2B FE7516CA 5510A6BB B633D517 EC25D4E0 BBAA16C2

Exécution de l'agrément de clé

La puce et le terminal exécutent un agrément de clé Diffie-Hellman anonyme en utilisant leurs clés secrètes et le générateur mappé $\hat{g}$.

Clé privée de la puce SK_{IC}	020F018C 7284B047 FA7721A3 37EFB7AC B1440BB3 0C5252BD 41C97C30 C994BB78 E9F0C5B3 2744D840 17D21FFA 6878396A 6469CA28 3EF5C000 DAF7D261 A39AB886 0ED4610A B5343390 897AAB5A 7787E4FA EFA0649C 6A94FDF8 2D991E8E 3FC332F5 142729E7 040A3F7D 5A4D3CD7 5CBEE1F0 43C1CAD2 DD484FEB 4ED22B59 7D36688E

Clé publique de la puce PK_{IC}	928D9A0F 9DBA450F 13FC859C 6F290D1D 36E42431 138A4378 500BEB4E 0401854C FF111F71 CB6DC1D0 335807A1 1388CC8E AA87B079 07AAD9FB A6B169AF 6D8C26AF 8DDDC39A DC3AD2E3 FF882B84 D23E9768 E95A80E4 746FB07A 9767679F E92133B4 D379935C 771BD7FB ED6C7BB4 B1708B27 5EA75679 524CDC9C 6A91370C C662A2F3
Clé privée du terminal SK_{IFD}	4BD0E547 40F9A028 E6A515BF DAF96784 8C4F5F5F FF65AA09 15947FFD 1A0DF2FA 6981271B C905F355 1457B7E0 3AC3B806 6DE4AA40 6C1171FB 43DD939C 4BA16175 103BA3DE E16419AA 248118F9 0CC36A3D 6F4C3736 52E0C3CC E7F0F1D0 C5425B36 00F0F0D6 A67F004C 8BBA33F2 B4733C72 52445C1D FC4F1107 203F71D2 EFB28161
Clé publique du terminal PK_{IFD}	0F0CC629 45A80292 51FB7EF3 C094E12E C68E4EF0 7F27CB9D 9CD04C5C 4250FAE0 E4F8A951 557E929A EB48E5C6 DD47F2F5 CD7C351A 9BD2CD72 2C07EDE1 66770F08 FFCB3702 62CF308D D7B07F2E 0DA9CAAA 1492344C 85290691 9538C98A 4BA4187E 76CE9D87 832386D3 19CE2E04 3C3343AE AE6EDBA1 A9894DC5 094D22F7 FE1351D5
Secret partagé K	419410D6 C0A17A4C 07C54872 CE1CBCEB 0A2705C1 A434C8A8 9A4CFE41 F1D78124 CA7EC52B DE7615E5 345E48AB 1ABB6E7D 1D59A57F 3174084D 3CA45703 97C1F622 28BDFDB2 DA191EA2 239E2C06 0DBE3BBC 23C2FCD0 AF12E0F9 E0B99FCF 91FF1959 011D5798 B2FCBC1F 14FCC24E 441F4C8F 9B08D977 E9498560 E63E7FFA B3134EA7

Les clés de session K_{Enc} et K_{MAC} sont calculées à partir de K en utilisant la fonction de hachage SHA-1 : K_{Enc}= SHA-1(K||0x00000001) et K_{MAC}=SHA-1(K||0x00000002). Ensuite, seuls les 16 premiers octets du condensé sont utilisés avec le résultat suivant :

K_{Enc}	01AFC10C F87BE36D 8179E873 70171F07
K_{MAC}	23F0FBD0 5FD6C7B8 B88F4C83 09669061

Authentification mutuelle

Les jetons d'authentification sont calculés en appliquant un CMAC aux entrées suivantes avec la clé K_{MAC}.

Données d'entrée pour T_{IC}	7F49818F 060A0400 7F000702 02040302 8481800F 0CC62945 A8029251 FB7EF3C0 94E12EC6 8E4EF07F 27CB9D9C D04C5C42 50FAE0E4 F8A95155 7E929AEB 48E5C6DD 47F2F5CD 7C351A9B D2CD722C 07EDE166 770F08FF CB370262 CF308DD7 B07F2E0D A9CAAA14 92344C85 29069195 38C98A4B A4187E76 CE9D8783 2386D319 CE2E043C 3343AEAE 6EDBA1A9 894DC509 4D22F7FE 1351D5
Données d'entrée pour T_{IFD}	7F49818F 060A0400 7F000702 02040302 84818092 8D9A0F9D BA450F13 FC859C6F 290D1D36 E4243113 8A437850 0BEB4E04 01854CFF 111F71CB 6DC1D033 5807A113 88CC8EAA 87B07907 AAD9FBA6 B169AF6D 8C26AF8D DDC39ADC 3AD2E3FF 882B84D2 3E9768E9 5A80E474 6FB07A97 67679FE9 2133B4D3 79935C77 1BD7FBED 6C7BB4B1 708B275E A7567952 4CDC9C6A 91370CC6 62A2F3

Les jetons d'authentification correspondants sont :

T_{IC}	C2F04230 187E1525
T_{IFD}	55D61977 CBF5307E

— — — — — — — —

Appendice I à la Partie 11
(INFORMATIF)

EXEMPLE DÉTAILLÉ : PACE — MAPPAGE D'AUTHENTIFICATION DE PUCE

Le présent appendice présente un exemple de protocole PACE avec mappage d'authentification de puce basé sur Diffie-Hellman à courbe elliptique (ECDH). Tous les nombres figurant dans les tableaux sont notés en hexadécimal.

La ZLA est utilisée comme mot de passe. Les champs de données pertinents de la ZLA, y compris les chiffres de contrôle, sont :

- Numéro de document : C11T002JM4 ;
- Date de naissance : 9608122 ;
- Date d'expiration : 2310314.

Le codage de K de la ZLA et la clé de chiffrement calculée K_π sont donc :

K	894D03F1 48C6265E 89845B21 8856EA34 D00EF8E8
K_π	4E6F6FBF 7BE748B9 32C7B741 61BBA9DF

I.1 EXEMPLE BASÉ SUR ECDH

Cet exemple est basé sur ECDH en appliquant les paramètres de domaine normalisés BrainpoolP256r1 (voir RFC 5639).

La première section présente la structure de données `PACEInfo` correspondante. Les APDU échangées, y compris tous les nonces et toutes les clés éphémères générés, sont indiquées et examinées.

Paramètres de courbe elliptique

En utilisant les paramètres de domaine normalisés, toutes les informations requises pour exécuter PACE sont données par la structure de données `PACEInfo`. Notamment, aucune structure `PACEDomainParameterInfo` n'est nécessaire.

PACEInfo	3012060A 04007F00 07020204 06020201 0202010D

La structure de `PACEInfo` est présentée en détail dans le tableau suivant :

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***
30	12		SEQUENCE	PACEInfo
06	0A	04 00 7F 00 07 02 02 04 06 02	OBJECT IDENTIFIER	PACE avec ECDH, mappage d'authentification de puce et clés de session AES 128
02	01	02	INTEGER	Version 2
02	01	0D	INTEGER	Paramètres de domaine normalisés Brainpool P256r1

Pour faciliter la présentation de l'exemple, un codage ASN.1 des paramètres de domaine BrainpoolP256r1 est indiqué ci-dessous.

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***
30	81 EC		SEQUENCE	Paramètre de domaine
06	07	2A 86 48 CE 3D 02 01	OBJECT IDENTIFIER	Algorithme id-ecPublicKey
30	81 E0		SEQUENCE	Paramètre de domaine
02	01	01	INTEGER	Version
30	2C		SEQUENCE	Champ sous-jacent
06	07	2A 86 48 CE 3D 01 01	OBJECT IDENTIFIER	Champ principal
02	21	00 A9 FB 57 DB A1 EE A9 BC 3E 66 0A 90 9D 83 8D 72 6E 3B F6 23 D5 26 20 28 20 13 48 1D 1F 6E 53 77	INTEGER	Nombre premier p
30	44		SEQUENCE	Équation de la courbe
04	20	7D 5A 09 75 FC 2C 30 57 EE F6 75 30 41 7A FF E7 FB 80 55 C1 26 DC 5C 6C E9 4A 4B 44 F3 30 B5 D9	OCTET STRING (chaîne d'octets)	Paramètre a
04	20	26 DC 5C 6C E9 4A 4B 44 F3 30 B5 D9 BB D7 7C BF 95 84 16 29 5C F7 E1 CE 6B CC DC 18 FF 8C 07 B6	OCTET STRING (chaîne d'octets)	Paramètre b

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***
04	41		OCTET STRING (chaîne d'octets)	Générateur de groupe G
		04	–	Point non compressé
		8B D2 AE B9 CB 7E 57 CB 2C 4B 48 2F FC 81 B7 AF B9 DE 27 E1 E3 BD 23 C2 3A 44 53 BD 9A CE 32 62	–	Coordonnée x
		54 7E F8 35 C3 DA C4 FD 97 F8 46 1A 14 61 1D C9 C2 77 45 13 2D ED 8E 54 5C 1D 54 C7 2F 04 69 97	–	Coordonnée y
02	21	00 A9 FB 57 DB A1 EE A9 BC 3E 66 0A 90 9D 83 8D 71 8C 39 7A A3 B5 61 A6 F7 90 1E 0E 82 97 48 56 A7	INTEGER	Ordre de groupe n
02	01	01	INTEGER	Cofacteur f

Flux d'application de l'exemple basé sur ECDH

Pour initialiser PACE, le terminal envoie la commande MSE:AT à la puce.

T>C :	`00 22 C1 A4 0F 80 0A 04 00 7F 00 07 02 02 04 06 02 83 01 01`
C>T :	`90 00`

Dans ce cas-ci, T>C désigne une APDU envoyée du terminal à la puce, et C>T désigne la réponse correspondante envoyée par la puce au terminal. Le tableau ci-après explique le codage de la commande.

Commande				
CLA	00		En clair	
INS	22		Gestion de l'environnement de sécurité	
P1/P2	C1 A4		Mettre le gabarit d'authentification à authentification mutuelle	
Lc	0F		Longueur du champ de données	
Données	Étiquette	Longueur	Valeur	Observation
	80	0A	04 00 7F 00 07 02 02 04 06 02	Mécanisme cryptographique : PACE avec ECDH, mappage d'authentification de puce et clés de session AES 128
	83	01	01	Mot de passe : ZLA

Réponse		
Octets d'état	90 00	Traitement normal

Nonce chiffré

La puce génère ensuite aléatoirement le nonce s et le chiffre au moyen de K_{π}.

Nonce déchiffré s	658B860B C94DF6F0 44FCE6D5 C82CF8E5
Nonce chiffré z	CB60E8E0 D85B76A9 BD304747 C2AD42E2

Le terminal demande le nonce chiffré.

T>C :	10 86 00 00 02 7C 00 00
C>T :	7C 12 80 10 CB 60 E8 E0 D8 5B 76 A9 BD 30 47 47 C2 AD 42 E2 90 00

Le codage de l'APDU commande et de la réponse correspondante est décrit dans le tableau suivant :

Commande				
CLA	10		Chaînage des commandes	
INS	86		AUTHENTIFICATION GÉNÉRALE	
P1/P2	00 00		Clés et protocole connus implicitement	
Lc	02		Longueur des données	
Données	Étiquette	Longueur	Valeur	Observation
	7C	00	–	Absentes
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.	
Réponse				
Données	Étiquette	Longueur	Valeur	Observation
	7C	12		Données d'authentification dynamique
	80	10	CB60E8E0 D85B76A9 BD304747 C2AD42E2	Nonce chiffré
Octets d'état	90 00		Traitement normal	

Mappage du nonce

Le nonce est mappé sur un générateur de groupe éphémère en utilisant le mappage générique. Les clés éphémères requises choisies aléatoirement sont aussi recueillies dans le tableau suivant :

Clé privée du terminal	5D8BB87B D74D985A 4B7D4325 B9F7B976 FE835122 77340079 8914AA22 738135CC
Clé publique du terminal	7F1D410A DB7DDB3B 84BF1030 800981A9 105D7457 B4A3ADE0 02384F30 86C67EDE 1AB88910 4A27DB6D 842B0190 20FBF3CE ACB0DC62 7F7BDCAC 29969E19 D0E553C1
Clé privée de la puce	9E56A6B5 9C95D06E CE5CD10F 983BB2F4 F1943528 E577F238 81D89D8C 3BBEE0AA
Clé publique de la puce	A234236A A9B9621E 8EFB73B5 245C0E09 D2576E52 77183C12 08BDD552 80CAE8B3 04F36571 3A356E65 A451E165 ECC9AC0A C46E3771 342C8FE5 AEDD0926 85338E23
Secret partagé H	2C1DCC17 73346492 C6636A36 EE4B965E 292E9AAE 7EE37736 EF58B9D0 A043F348 403A8CF3 3CA7DC0D 9DF61D08 89CE2442 4FF97C1A AD48A5CA 2A554B07 1EF7638D
Générateur Ĝ mappé	89F0B5EA BF3BE293 C75903A3 98613192 5C9F5B51 5CA95AF4 85DC7E88 6F03245D 44BEFB2D D3A0DBD7 1CB5E618 971CF474 7F12B79E 548379A4 0E45963B AAF3E829

Le terminal et la puce échangent les APDU suivantes pour mapper le nonce.

T>C :	10 86 00 00 45 7C 43 81 41 04 7F 1D 41 0A DB 7D DB 3B 84 BF 10 30 80 09 81 A9 10 5D 74 57 B4 A3 AD E0 02 38 4F 30 86 C6 7E DE 1A B8 89 10 4A 27 DB 6D 84 2B 01 90 20 FB F3 CE AC B0 DC 62 7F 7B DC AC 29 96 9E 19 D0 E5 53 C1 00
C>T :	7C 43 82 41 04 A2 34 23 6A A9 B9 62 1E 8E FB 73 B5 24 5C 0E 09 D2 57 6E 52 77 18 3C 12 08 BD D5 52 80 CA E8 B3 04 F3 65 71 3A 35 6E 65 A4 51 E1 65 EC C9 AC 0A C4 6E 37 71 34 2C 8F E5 AE DD 09 26 85 33 8E 23 90 00

La structure des APDU peut être décrite comme suit :

Commande					
CLA	10		Chaînage des commandes		
INS	86		AUTHENTIFICATION GÉNÉRALE		
P1/P2	00 00		Clés et protocole connus implicitement		
Lc	45		Longueur des données		
Données	Étiquette	Longueur	Valeur	Observation	
	7C	43	–	Données d'authentification dynamique	
	81	41		Données de mappage	
			04		Point non compressé
			7F 1D 41 0A ... 86 C6 7E DE		Coordonnée x
			1A B8 89 10 ... D0 E5 53 C1		Coordonnée y
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.		
Réponse					
Données	Étiquette	Longueur	Valeur	Observation	
	7C	43		Données d'authentification dynamique	
	82	41		Données de mappage	
			04		Point non compressé
			A2 34 23 6A ... 80 CA E8 B3		Coordonnée x
			04 F3 65 71 ... 85 33 8E 23		Coordonnée y
Octets d'état	90 00		Traitement normal		

Exécution de l'agrément de clé

Durant la troisième étape, la puce et le terminal exécutent un agrément de clé ECDH anonyme en utilisant les nouveaux paramètres de domaine déterminés par le générateur de groupe éphémère de l'étape précédente. Seule la coordonnée x est requise comme secret partagé vu que KDF n'utilise que la première coordonnée pour calculer les clés de session.

Clé privée du terminal	76ECFDAA 9841C323 A3F5FC5E 88B88DB3 EFF7E35E BF57A7E6 946CB630 006C2120
Clé publique du terminal	446C9340 84D9DAB8 63944F21 9520076C 29EE3F7A E6722B11 FF319EC1 C7728F95 5483400B FF60BF0C 59292700 09277DC2 A515E125 75010AD9 BA916CF1 BF86FEFC
Clé privée de la puce	CD626EF3 C256E235 FE8912CA C28279E6 26008EDA 6B3A05C4 CF862A3B DAB79E78
Clé publique de la puce	02AD566F 3C6EC7F9 324509AD 50A51FA5 2030782A 4968FCFE DF737DAE A9933331 11C3B9B4 C2287789 BD137E7F 8AA882E2 A3C633CC D6ECC2C6 3C57AD40 1A09C2E1
Secret partagé	67950559 D0C06B4D 4B86972D 14460837 461087F8 419FDBC3 6AAF6CEA AC462832

L’agrément de clé est exécuté comme suit :

T>C :	10 86 00 00 45 7C 43 83 41 04 44 6C 93 40 84 D9 DA B8 63 94 4F 21 95 20 07 6C 29 EE 3F 7A E6 72 2B 11 FF 31 9E C1 C7 72 8F 95 54 83 40 0B FF 60 BF 0C 59 29 27 00 09 27 7D C2 A5 15 E1 25 75 01 0A D9 BA 91 6C F1 BF 86 FE FC 00
C>T :	7C 43 84 41 04 02 AD 56 6F 3C 6E C7 F9 32 45 09 AD 50 A5 1F A5 20 30 78 2A 49 68 FC FE DF 73 7D AE A9 93 33 31 11 C3 B9 B4 C2 28 77 89 BD 13 7E 7F 8A A8 82 E2 A3 C6 33 CC D6 EC C2 C6 3C 57 AD 40 1A 09 C2 E1 90 00

Le codage de l’agrément de clé est analysé dans le tableau ci-après :

Commande				
CLA	10		Chaînage des commandes	
INS	86		AUTHENTIFICATION GÉNÉRALE	
P1/P2	00 00		Clés et protocole connus implicitement	
Lc	45		Longueur des données	
Données	Étiquette	Longueur	Valeur	Observation
	7C	43	–	Données d’authentification dynamique
	83	41		Clé publique éphémère du terminal
			04	Point non compressé

Commande					
			44 6C 93 40 ... C7 72 8F 95		Coordonnée x
			54 83 40 0B ... BF 86 FE FC		Coordonnée y
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.		
Réponse					
Données	Étiquette	Longueur	Valeur	Observations	
	7C	43		Données d'authentification dynamique	
	84	41		Clé publique éphémère de la puce	
			04		Point non compressé
			02 AD 56 6F ... A9 93 33 31		Coordonnée x
			11 C3 B9 B4 ... 1A 09 C2 E1		Coordonnée y
Octets d'état	90 00		Traitement normal		

Les clés de session AES 128 KS_{Enc} et KS_{MAC} suivantes sont calculées à partir du secret partagé en utilisant KDF :

KS_{Enc}	0A9DA4DB 03BDDE39 FC5202BC 44B2E89E
KS_{MAC}	4B1C0649 1ED5140C A2B537D3 44C6C0B1

Authentification mutuelle

Les jetons d'authentification sont calculés au moyen de KS_{MAC} en utilisant :

Données d'entrée pour T_{IFD}	7F494F06 0A04007F 00070202 04060286 410402AD 566F3C6E C7F93245 09AD50A5 1FA52030 782A4968 FCFEDF73 7DAEA993 333111C3 B9B4C228 7789BD13 7E7F8AA8 82E2A3C6 33CCD6EC C2C63C57 AD401A09 C2E1
Données d'entrée pour T_{IC}	7F494F06 0A04007F 00070202 04060286 4104446C 934084D9 DAB86394 4F219520 076C29EE 3F7AE672 2B11FF31 9EC1C772 8F955483 400BFF60 BF0C5929 27000927 7DC2A515 E1257501 0AD9BA91 6CF1BF86 FEFC

comme entrée. Le codage des données d'entrée est indiqué ci-dessous :

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***
7F49	4F		PUBLIC KEY	Données d'entrée pour T_{IFD}
06	0A	04 00 7F 00 07 02 02 04 06 02	OBJECT IDENTIFIER	PACE avec ECDH, mappage d'authentification de puce et clés de session AES 128
86	41		ELLIPTIC CURVE POINT (point de courbe elliptique)	Point public éphémère de la puce
		04		Point non compressé
		02 AD 56 6F ... A9 93 33 31		Coordonnée x
		11 C3 B9 B4 ... 1A 09 C2 E1		Coordonnée y

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***
7F49	4F		PUBLIC KEY	Données d'entrée pour T_{IC}
06	0A	04 00 7F 00 07 02 02 04 06 02	OBJECT IDENTIFIER	PACE avec ECDH, mappage d'authentification de puce et clés de session AES 128
86	41		ELLIPTIC CURVE POINT (point de courbe elliptique)	Point public éphémère du terminal
		04		Point non compressé
		44 6C 93 40 ... C7 72 8F 95		Coordonnée x
		54 83 40 0B ... BF 86 FE FC		Coordonnée y

Les jetons d'authentification calculés sont :

T_{IFD}	E86BD060 18A1CD3B
T_{IC}	8596CF05 5C67C1A3

En dernier lieu, ces jetons sont échangés et vérifiés.

T>C :	00 86 00 00 0C 7C 0A 85 08 E8 6B D0 60 18 A1 CD 3B 00
C>T :	7C 3C 86 08 85 96 CF 05 5C 67 C1 A3 8A 30 1E EA 96 4D AA E3 72 AC 99 0E 3E FD E6 33 33 53 BF C8 9A 67 04 D9 3D A8 79 8C F7 7F 5B 7A 54 BD 10 CB A3 72 B4 2B E0 B9 B5 F2 8A A8 DE 2F 4F 92 90 00

Le codage de l'authentification mutuelle est analysé dans le tableau ci-après :

Commande					
CLA	00		Pas de chaînage des commandes (dernière commande en chaîne)		
INS	86		AUTHENTIFICATION GÉNÉRALE		
P1/P2	00 00		Clés et protocole connus implicitement		
Lc	0C		Longueur des données		
Données	Étiquette	Longueur	Valeur	Observation	
	7C	0A	–	Données d'authentification dynamique	
	85	08		Jeton d'authentification du terminal	
			E8 6B D0 60 18 A1 CD 3B		T_{IFD}
Le	00		La longueur maximale en octets attendue du champ données de la réponse est de 256.		
Réponse					
Données	Étiquette	Longueur	Valeur	Observation	
	7C	3C		Données d'authentification dynamique	
	86	08		Jeton d'authentification de la puce	
			85 96 CF 05 5C 67 C1 A3		T_{IC}
	8A	30			Coordonnée x
			1E EA 96 4D … DE 2F 4F 92		Données d'authentification de puce chiffrées
Octets d'état	90 00		Traitement normal		

Authentification de la puce

Obtenir `ChipAuthenticationPublicKeyInfo` de EF.CardSecurity

ChipAuthenticationPublicKeyInfo	30620609 04007F00 07020201 02305230 0C060704 007F0007 01020201 0D034200 04187270 9494399E 7470A643 1BE25E83 EEE24FEA 568C2ED2 8DB48E05 DB3A610D C884D256 A40E35EF CB59BF67 53D3A489 D28C7A4D 973C2DA1 38A6E7A4 A08F68E1 6F02010D

La structure de `ChipAuthenticationPublicKeyInfo` est présentée en détail dans le tableau suivant :

Étiquette	***Longueur***	***Valeur***	***Type ASN.1***	***Observation***	
30	62		SEQUENCE	ChipAuthenticationPublicKeyInfo	
06	09	04 00 7F 00 07 02 02 01 02	OBJECT IDENTIFIER		id-PK-ECDH
30	52		SEQUENCE		SubjectPublicKeyInfo
30	0C		SEQUENCE		Paramètres de domaine normalisés Brainpool P256r1
06	07	04 00 7F 00 07 01 02	OBJECT IDENTIFIER		standardizedDomainParameters
02	01	0D	INTEGER		Brainpool256r1
03	42	00 04 18 72 70… 8F 68 E1 6F	BIT STRING		CA Public Key
02	01	0D	INTEGER		keyID 13

Les données suivantes sont utilisées pour l'authentification de la puce :

Données d'authentification de puce chiffrées	1EEA964D AAE372AC 990E3EFD E6333353 BFC89A67 04D93DA8 798CF77F 5B7A54BD 10CBA372 B42BE0B9 B5F28AA8 DE2F4F92
Données d'authentification de puce déchiffrées	85DC3FA9 3D0952BF A82F5FD1 89EE75BD 82F11D1F 0B8ED4BF 5319AC9B 53C426B3
IV pour (dé)chiffrement des données CA IV = E(KS_{ENC}, -1)	F6A3B75A1 E933941 DD7A13E2 520779DF

Clé publique de la puce du nonce de mappage de l'AUTHENTIFICATION GÉNÉRALE $PK_{MAP,IC}$	A234236A A9B9621E 8EFB73B5 245C0E09 D2576E52 77183C12 08BDD552 80CAE8B3 04F36571 3A356E65 A451E165 ECC9AC0A C46E3771 342C8FE5 AEDD0926 85338E23
Clé publique CA de la puce de ChipAuthenticationPublicKeyInfo PK_{IC}	18727094 94399E74 70A6431B E25E83EE E24FEA56 8C2ED28D B48E05DB 3A610DC8 84D256A4 0E35EFCB 59BF6753 D3A489D2 8C7A4D97 3C2DA138 A6E7A4A0 8F68E16F

Le terminal vérifie que $PK_{MAP,IC} = KA(CA_{IC}, PK_{IC}, D_{IC})$.

— — — — — — — —

Appendice J à la Partie 11
(INFORMATIF)

PROCÉDURES D'INSPECTION

J.1 PROCÉDURE D'INSPECTION POUR APPLICATION DVLM-e

La présente section décrit une procédure d'inspection qui ne contient qu'une application DVLM-e (« documents-SDL1 »).

1. Accès au circuit intégré sans contact (voir § 4.2)

   - Si l'accès au CI est protégé, PACE ou BAC peuvent être utilisés dans cette étape, bien qu'il soit recommandé d'utiliser PACE pour des raisons de sécurité. Depuis le 1/1/2018, les DVLM-e ne peuvent prendre en charge que PACE.

   - Si le CI et le terminal le permettent, PACE-CAM devrait être utilisé pour des raisons de performance.

   - L'IC accorde l'accès à des données moins sensibles dans l'application DVLM-e et à EF.CardSecurity dans le fichier principal, s'il est présent.

2. Démarrage de l'authentification des données

   - Lire l'objet de sécurité du document et vérifier la signature, y compris la vérification en chaîne du certificat de signataire de document.

3. Authentification de la puce

   - En fonction de la prise en charge par le CI, effectuer une authentification de la puce ou une authentification active. La prise en charge de l'authentification active est indiquée par la présence de EF.DG15 dans l'application DVLM-e ; la prise en charge de l'authentification de la puce est indiquée par la présence des `SecurityInfos` correspondantes dans EF.DG14.

   - Cette étape peut également être exécutée dans le cadre de l'étape 1 si PACE avec mappage d'authentification de puce est utilisé.

   - L'authentification n'est complète qu'en combinaison avec l'authentification du fichier contenant la clé publique (EF.CardSecurity, EF.DG14 ou EF.DG15) utilisée pour cette étape.

4. Contrôle d'accès supplémentaire

   - L'authentification du terminal est nécessaire si le DVLM-e est configuré de façon à l'exiger pour l'accès à des données sensibles, c'est-à-dire EF.DG3 et/ou EF.DG4.

5. Lecture des données

   - La lecture des données peut commencer dès que les droits d'accès nécessaires sont accordés, par exemple, les données moins sensibles peuvent être lues après l'étape 1.

   - Les données ne doivent pas être considérées comme authentiques sans authentification des données lues (étape 2).

J.2 PROCÉDURE D'INSPECTION POUR DVLM-e À APPLICATIONS MULTIPLES

La présente section décrit une procédure d'inspection conçue pour les DVLM-e contenant une ou plusieurs applications en plus de l'application DVLM-e (« documents-SDL2 »). Cette procédure peut également être utilisée pour n'accéder qu'à l'application DVLM-e.

1. Accès au circuit intégré sans contact (voir § 4.2)

   - Dans cette configuration, seul PACE est disponible pour accéder au CI.

   - Si le CI et le terminal le permettent, PACE-CAM devrait être utilisé pour des raisons de performance.

   - Le CI accorde l'accès aux données moins sensibles dans l'application DVLM-e et à EF.CardSecurity dans le fichier principal.

2. Vérification de la présence d'EF.CardSecurity

   - Si EF.CardSecurity n'est pas présent, le DVLM-e ne prend pas en charge l'authentification dans le fichier principal (ce qui implique que le CI ne contient qu'une application DVLM-e). Dans ce cas, sélectionner l'application DVLM-e et poursuivez avec l'étape 2 de la procédure de la section J.1 du présent appendice.

3. Démarrage de l'authentification des données

   - Lire EF.CardSecurity et vérifier la signature, y compris la vérification en chaîne du certificat de signataire de document.

   - Les données de l'application DVLM-e sont protégées par l'objet de sécurité du document, qui doit être vérifié lorsque les données de cette application sont lues. Les données provenant d'autres applications sont protégées par les signatures des données, qui doivent également être vérifiées lors de la lecture de ces données.

4. Authentification de la puce

   - Effectuer l'authentification de la puce dans le fichier principal. Si les informations nécessaires ne figurent pas dans les `SecurityInfos` dans EF.CardSecurity, le CI ne prend pas en charge l'authentification dans le fichier principal. Dans ce cas, sélectionner l'application DVLM-e et poursuivre avec l'étape 2 de la procédure de la section J.1 du présent appendice.

   - Cette étape peut également être exécutée dans le cadre de l'étape 1 si PACE avec mappage d'authentification de puce est utilisé.

- L’authentification n’est complète qu’en combinaison avec l’authentification du fichier contenant la clé publique (EF.CardSecurity) utilisée pour cette étape.

5. Contrôle d’accès supplémentaire

   - Effectuer l’authentification du terminal.

   - Si seul un accès en lecture aux données moins sensibles de l’application DVLM-e est requis, cette étape peut être ignorée.

6. Lecture/écriture de données

   - La lecture/écriture de données inclut la sélection des applications contenant les fichiers.

   - La lecture des données peut commencer dès que les droits d’accès nécessaires sont accordés, par exemple, les données moins sensibles de l’application DVLM-e peuvent être lues après l’étape 1.

   - Les données ne doivent pas être considérées comme authentiques sans authentification des données lues (étape 3).

— — — — — — — —

Appendice K à la Partie 11
(INFORMATIF)

CONTRÔLE D'ACCÈS ÉTENDU EUROPÉEN

L'authentification du terminal définie dans le présent document se fonde sur le contrôle d'accès étendu utilisé dans l'Union européenne (voir TR-03110) pour protéger l'accès aux empreintes digitales archivées dans l'application SDL1. Le présent appendice souligne les différences entre TR-03110 et les protocoles définis dans ce document.

La procédure d'inspection avancée utilisée pour accéder aux DVLM-e équipés d'un EAC selon TR-03110 comprend les étapes suivantes :

1. Effectuer la procédure d'accès à la puce (voir § 4.2) et sélectionner l'application DVLM-e ;

2. Effectuer l'authentification de la puce dans l'application DVLM-e (voir § 6.2) et commencer l'authentification passive (voir § 5.1) ;

3. Effectuer l'authentification du terminal (voir ci-dessous) dans l'application DVLM-e (voir § 7.1).

Note.— L'authentification de la puce et du terminal est effectuée dans l'application DVLM-e dans le cadre du contrôle d'accès étendu européen. Les spécifications du présent document permettent à ces protocoles, selon le contexte, d'être exécutés soit dans l'application DVLM-e, soit dans le fichier principal.

K.1 DROITS D'ACCÈS

Tableau K-1. Autorisation des systèmes d'inspection

7	6	5	4	3	2	1	0	Description
x	x	-	-	-	-	-	-	Rôle (voir le Doc 9303-12)
-	-	x	x	x	x	x	x	Droits d'accès
-	-	x	x	x	x	-	-	RFU
-	-	-	-	-	-	1	-	Accès en lecture à l'application DVLM-e : DG4 (Iris)
-	-	-	-	-	-	-	1	Accès en lecture à l'application DVLM-e : DG3 (empreinte digitale)

Les droits d'accès aux groupes de données dans les applications autres que l'application DVLM-e sont transmis par le biais d'extensions d'autorisation telles qu'elles sont définies dans les parties 12 et 10 du Doc 9303. Les droits d'accès aux empreintes digitales (et à l'iris) sont transmis par le biais du modèle d'autorisation du détenteur de certificat :

Pour le calcul des droits d'accès effectifs, voir § 7.1.4.3.6.

K.2 EF.CVCA

Selon la spécification, les points de confiance (références de l'autorité de certification) connus du CI pour la vérification du certificat dans le cadre de l'authentification du terminal sont transmis à l'IFD dans le cadre du protocole PACE (voir § 4.4.3.5).

Au lieu de cela, le contrôle d'accès étendu européen définit un fichier transparent EF.CVCA dans l'application DVLM-e. La spécification est présentée ci-dessous :

Tableau K-2. Fichier élémentaire EF.CVCA

Nom de fichier	EF.CVCA
ID de fichier	0x011C (par défaut)
ID de fichier court	0x1C (par défaut)
Accès en lecture	PACE
Accès en écriture	JAMAIS (mise à jour interne uniquement)
Taille	36 octets (fixes) complétés par des octets de valeur 0x00
Contenu	[CARi][||CARi-1][||0x00..00]

Si le CI prend en charge l'authentification du terminal dans l'application DVLM-e, il DOIT mettre à disposition les références des clés publiques CVCA convenant aux systèmes d'inspection dans un fichier élémentaire transparent EF.CVCA dans l'application DVLM-e, comme indiqué dans le Tableau K-2.

Ce fichier DOIT contenir une séquence d'objets de données de référence de l'autorité de certification (CAR) (voir le Doc 9303-12) adaptée à l'authentification du terminal.

- Il DOIT contenir au maximum deux objets de données de référence de l'autorité de certification.
- La référence de l'autorité de certification la plus récente DOIT être le premier objet de données de cette liste.
- Le fichier DOIT être complété par l'ajout d'octets de valeur 0x00.

Le fichier EF.CVCA comporte un identificateur EF par défaut et un identificateur EF court. Si les valeurs par défaut ne peuvent pas être utilisées, l'identificateur EF (court) DOIT être spécifié dans le paramètre OPTIONNEL `efCVCA` de `TerminalAuthenticationInfo`. Si `efCVCA` est utilisé pour indiquer l'identificateur EF à utiliser, l'identificateur EF par défaut est remplacé. Si aucun identificateur EF court n'est donné dans `efCVCA`, le fichier EF.CVCA DOIT être explicitement sélectionné à l'aide de l'identificateur EF donné.

```
TerminalAuthenticationInfo ::= SEQUENCE {
  protocol OBJECT IDENTIFIER(id-TA),
  version  INTEGER, -- MUST be 1
  efCVCA   FileID OPTIONAL
}

FileID ::= SEQUENCE {
  fid  OCTET STRING (SIZE(2)),
  sfid OCTET STRING (SIZE(1)) OPTIONAL
}
```

— FIN —

ISBN 978-92-9275-554-6