OACI

Doc 9303

Documents de voyage lisibles à la machine

Huitième édition, 2021

Partie 12 : Infrastructure à clés publiques 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 12 : Infrastructure à clés publiques 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/mrtd permet de télécharger les documents et d'obtenir des renseignements supplémentaires.

Doc 9303, *Documents de voyage lisibles à la machine*
Partie 12 — *Infrastructure à clés publiques pour les DVLM*

Commande n° : 9303P12
ISBN 978-92-9265-576-1 (version imprimée)

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	Inséré par

RECTIFICATIFS		
N°	Date	Inséré par

TABLE DES MATIÈRES

Page

1. PORTÉE

La Partie 12 du Doc 9303 définit l'infrastructure à clés publiques (ICP) pour l'application DVLM-e. Elle spécifie les exigences pour les États émetteurs et les organisations émettrices, notamment le fonctionnement d'une autorité de certification (AC), qui émet les certificats et les listes de certificats révoqués (CRL). Elle spécifie également les prescriptions applicables aux États récepteurs et à leurs systèmes d'inspection qui valident ces certificats et ces CRL.

La huitième édition du Doc 9303 intègre les spécifications relatives aux cachets numériques visibles (VDS) et aux applications facultatives dossiers de voyage, dossiers de visa et éléments biométriques supplémentaires (appelées applications SDL2) en tant qu'extension de l'application obligatoire DVLM-e (appelée SDL1).

Le Doc 9303-12 doit être lu en parallèle avec :

- Doc 9303-10 — *Structure de données logique (SDL) pour le stockage des données biométriques et d'autres données dans le circuit intégré (CI) sans contact* ;
- Doc 9303-11 — *Mécanismes de sécurité pour les DVLM* ;
- Doc 9303-13 — *Cachets numériques visibles.*

2. APERÇU DE L'INFRASTRUCTURE À CLÉS PUBLIQUES

L'infrastructure à clés publiques (ICP) des DVLM-e permet la création et la vérification ultérieure des signatures numériques contenues dans les objets DVLM-e, notamment l'objet de sécurité du document (SO_D), pour garantir que les données signées sont authentiques et n'ont pas été modifiées. La révocation d'un certificat, l'échec de la procédure de validation de l'itinéraire de certification ou l'échec de la vérification de la signature numérique ne sont pas par eux-mêmes suffisants pour considérer un DVLM-e comme non valide. Cet échec signifie que la vérification électronique de l'intégrité et de l'authenticité des données SDL a échoué, mais d'autres mécanismes non électroniques peuvent ensuite être employés dans le cadre de l'inspection globale du DVLM-e pour déterminer s'il est valide.

L'infrastructure ICP utilisée pour les DVLM-e est beaucoup plus simple que les ICP multi-applications plus génériques telles que l'ICP utilisée pour Internet, définie dans la norme RFC 5280. Dans l'ICP du DVLM-e, chaque État émetteur/autorité émettrice établit une seule autorité de certification (AC) qui émet tous les certificats directement à des entités finales, y compris les signataires de documents. Ces autorités de certification s'appellent des AC signataires nationales (ACSN). Il n'y a pas d'autres AC dans l'infrastructure. Les États récepteurs établissent la confiance directement dans les clés/certificats de l'ACSN de chaque État émetteur ou organisation émettrice.

L'ICP du DVLM-e est fondée sur des normes ICP génériques, notamment les normes X.509 et RFC 5280. Ces normes ICP de base définissent un grand ensemble d'éléments optionnels et de relations de confiance complexes entre les AC, qui ne concernent pas l'application DVLM-e. La présente partie du Doc 9303 donne un aperçu de ces normes, adaptées à l'application DVLM-e. Les éléments spécifiques de l'application DVLM-e sont notamment les suivants :

- il n'y a exactement qu'une ACSN par État émetteur ;
- les itinéraires de certification contiennent exactement un certificat (p. ex., le signataire du document) ;
- la vérification de la signature doit être possible de 5 à 10 ans après sa création ;

- la modification du nom de l'ACSN est prise en charge ;
- les certificats de liaison de l'ACSN ne sont pas traités comme des certificats intermédiaires dans un itinéraire de certification.

L'ICP du DVLM-e est, dans l'ensemble, conforme à la norme RFC 5280. Cependant, la possibilité qu'ont les ACSN de changer de nom impose à l'ICP du DVLM-e des exigences uniques qui sont incompatibles avec certaines procédures de validation des listes de certificats révoqués (CRL) définies dans la norme RFC 5280. Ces différences ont été réduites à un minimum et sont clairement identifiées.

Pour le VDS et l'application SDL2, l'ICP de signature numérique, qui assure l'intégrité et l'authenticité des objets de données, est une extension de l'ICP de la SDL1. Les signataires de VDS et de SDL2 sont émis par la même ACSN que celle qui émet les signataires de SDL1. Les modifications apportées aux profils de certificat pour ces nouvelles applications sont spécifiées dans le présent document. L'ensemble de cette infrastructure est appelé l'**ICP du DVLM-e**.

L'ICP de la signature numérique se compose des entités suivantes :

- l'autorité de certification signataire nationale (ACSN) ;
- les certificats de signataire de document (DSC), qui sont utilisés pour signer les objets de sécurité des documents (SO_D) ;
- les certificats de signataires de SDL2, qui se composent des éléments suivants :
  - le signataire de SDL2-TS – signe les tampons de voyage de la SDL2 ;
  - le signataire de SDL2-V – signe les visas électroniques de la SDL2 ;
  - le signataire de SDL2-B – signe les éléments biométriques supplémentaires de LDS2.
- les certificats de signataire de code à barres (BCSC), pour lesquels les deux types spécifiques suivants sont définis dans le présent document :
  - certificats de signataire de visa (VSC) ;
  - certificats de signataire de documents de voyage d'urgence (ESC).
- les certificats de signataire de liste de contrôle (MSC) utilisés pour signer les listes de contrôle ;
- les certificats de signataire de liste d'écarts (DLSC) sont utilisés pour signer les listes d'écarts ;
- la liste de certificats révoqués (CRL).

Tous les différents types de certificats sont signés par la même ACSN. L'ACSN signe aussi la CRL, qui contient tout certificat révoqué, quel que soit le type de certificat. Tous les certificats émis sous l'ACSN sont collectivement appelés **Signer Certificates (certificats de signataire)**.

Pour les applications SDL2, une **ICP d'autorisation** distincte est définie. L'ICP d'autorisation permet à l'État émetteur ou à l'organisation émettrice du DVLM-e de contrôler et de gérer les États étrangers qui sont autorisés à écrire des objets de données de SDL2 dans leur DVLM-e et à lire ces objets de données. Un État étranger qui a l'intention de lire ou d'écrire des données de SDL2 doit obtenir un certificat d'autorisation directement auprès de l'État émetteur ou de l'organisation émettrice du DVLM-e.

L'ICP d'autorisation utilise une structure de certificat différente (certificats vérifiables par carte ISO 7816) et nécessite donc des composants d'infrastructure supplémentaires.

La SDL2 exige que le terminal prouve au CI sans contact du DVLM-e qu'il est autorisé à écrire des objets de données de SDL2 dans le CI sans contact ou qu'il est autorisé à lire des objets de données de SDL2. Ce type de terminal est équipé d'au moins une clé privée et du certificat de terminal correspondant, codant la clé publique et les droits d'accès du terminal. Une fois que le terminal a prouvé qu'il connaît cette clé privée, la puce du DVLM accorde au terminal l'accès à la lecture ou à l'écriture des données de SDL2 comme indiqué dans le certificat du terminal.

L'ICP d'autorisation de SDL2 se compose des entités suivantes :

- les AC de vérification nationale (CVCA) ;
- les vérificateurs de documents (DV) ;
- les terminaux ;
- le point unique de contact (SPOC).

La distribution et la gestion des certificats d'autorisation entre les CVCA d'un État et les DV d'autres États sont assurées par un point unique de contact (SPOC) dans chaque État.

La Partie 12 du Doc 9303 spécifie le profil de l'ICP du DVLM-e, le profil de l'ICP d'autorisation et les objets correspondants, notamment :

La Partie 12 du Doc 9303 spécifie le profil de l'ICP du DVLM-e, notamment :

- les rôles et les responsabilités des entités de l'infrastructure ;
- les algorithmes de chiffrement et la gestion des clés ;
- le contenu des certificats et des CRL ;
- les mécanismes de distribution des certificats et des CRL ;
- la validation de l'itinéraire de certification.

3. RÔLES ET RESPONSABILITÉS

La présente section décrit en détail les entités et les rôles et responsabilités de l'ICP du DVLM-e et de l'ICP d'autorisation.

3.1 ICP du DVLM-e

L'authenticité et l'intégrité des données stockées dans les DVLM-e sont protégées par authentification passive. Ce mécanisme de sécurité est basé sur les signatures numériques et est constitué des entités pour l'ICP du DVLM-e suivantes :

- **AC signataire nationale (ACSN) :** Chaque État émetteur/autorité émettrice établit une ACSN unique comme point de confiance national dans le contexte des DVLM-e. L'ACSN émet des certificats de clés

publiques pour un ou plusieurs signataires (nationaux) de documents et, à titre facultatif, pour d'autres entités finales telles que les signataires de listes de contrôle et les signataires de listes d'écarts. L'ACSN émet aussi des CRL indiquant les certificats émis qui ont été révoqués.

- **Signataire de document (SD) :** Un signataire de document signe numériquement les données à stocker dans les DVLM-e ; cette signature est stockée dans un objet de sécurité du document sur le DVLM-e.

- **Signataires de SDL2 :** Un signataire de SDL2 signe numériquement les objets de données de SDL2 d'un ou de plusieurs types.

- **Signataire de code à barres (BCS) :** Un signataire de code à barres signe numériquement les données (en-tête et message) codées dans le code à barres. La signature est également enregistrée dans le code à barres. Le présent document spécifie deux cas d'utilisation pour le signataire de codes à barres, à savoir, visa et documents de voyage d'urgence.

- **Système d'inspection (IS) :** Un système d'inspection vérifie la signature numérique, y compris la validation de l'itinéraire de certification pour vérifier l'authenticité et l'intégrité des données électroniques stockées dans le DVLM-e dans le cadre de l'authentification passive.

- **Signataire de liste de contrôle :** Un signataire de liste de contrôle est une entité optionnelle qui signe numériquement une liste de certificats de l'ACSN (nationaux et étrangers) pour le mécanisme de distribution bilatérale des certificats de l'ACSN.

- **Signataires de liste d'écarts :** Les signataires de liste d'écarts sont utilisés pour signer les listes d'écarts ; les listes d'écarts sont définies dans le Doc 9303-3.

Les installations sécurisées pour générer des paires de clés DOIVENT être sous le contrôle de l'État émetteur ou de l'organisation émettrice. Chaque paire de clés comprend une clé « privée » et une clé « publique ». Les clés privées et les systèmes ou installations correspondants DOIVENT être bien protégés contre tout accès extérieur ou non autorisé, par leur conception intrinsèque et par des moyens de sécurisation du matériel.

Même si le certificat de l'ACSN demeure relativement statique, un grand nombre de certificats de signataire de document seront créés avec le temps.

L'ACSN de chaque État émetteur ou organisation émettrice sert de point de confiance pour l'État récepteur. L'État émetteur ou l'organisation émettrice distribue sa propre clé publique de l'ACSN aux États récepteurs sous forme de certificat. L'État récepteur établit la confiance dans ce certificat (et cette clé certifiée) par des moyens hors bande et enregistre une « ancre de confiance » pour cette clé et ce certificat de confiance. Ces certificats de l'ACSN DOIVENT être des certificats autosignés émis directement par l'ACSN. Les certificats de l'ACSN NE DOIVENT PAS être des certificats subordonnés ou croisés dans une infrastructure ICP plus large. Les certificats de liaison autoémis de l'ACSN peuvent aussi être émis pour aider l'État récepteur à établir la confiance dans une nouvelle clé ou un nouveau certificat de l'ACSN après un renouvellement de clé.

Note.— Certains États exigent que l'autorité suprême qui publie les certificats autosignés pour toutes les applications soit un contrôleur d'autorité de certification (CCA) centralisé. Dans ces cas, une solution possible est que l'ACSN crée un certificat autosigné (conforme aux prescriptions du Doc 9303) et que le CCA le contresigne (de manière conforme aux prescriptions nationales relatives au CCA). Cependant, ces certificats contresignés ne font pas partie de l'infrastructure ICP du DVLM-e et ne seraient pas distribués aux États récepteurs.

3.1.1 AC signataire nationale (ACSN)

Il est RECOMMANDÉ que les paires de clés de l'ACSN (KPu_{CSCA}, KPr_{CSCA}) soient générées et stockées dans une infrastructure d'AC hors ligne, hautement protégée.

La clé privée de l'ACSN (KPr_{CSCA}) est utilisée pour signer les certificats de signataire de document (C_{DS}), d'autres certificats et les CRL.

Les certificats de l'ACSN (C_{CSCA}) sont utilisés pour valider les certificats de signataire de document, les certificats de signataire de liste de contrôle, les certificats de signataire de liste d'écarts, les CRL et d'autres certificats émis par l'ACSN.

Tous les certificats et CRL DOIVENT être conformes aux profils spécifiés à la section 7 et DOIVENT être distribués au moyen des mécanismes de distribution spécifiés à la section 5.

Pour les participants au répertoire de clés publiques (RCP), chaque certificat de l'ACSN (C_{CSCA}) DOIT aussi être communiqué par l'émetteur du certificat au RCP [aux fins de validation des certificats de signataire de document (C_{DS})].

Les CRL DOIVENT être émises périodiquement, comme il est spécifié à la section 4.

3.1.2 Signataires de documents

Il est RECOMMANDÉ que les paires de clé de signataire de document (KPu_{DS}, KPr_{DS}) soient générées et stockées dans une infrastructure hautement protégée.

La clé privée de signataire de document (KPr_{DS}) est utilisée pour signer les objets de sécurité de document (SO_D).

Les certificats de signataire de document (C_{DS}) sont utilisés pour valider les objets de sécurité de document (SO_D).

Chaque certificat de signataire de document (C_{DS}) DOIT être conforme au profil de certificat défini à la section 7 et DOIT être stocké dans le CI sans contact de chaque DVLM-e qui a été signé avec la clé privée de signataire de document correspondante (voir le Doc 9303-10 pour plus de renseignements). L'État récepteur est ainsi assuré d'avoir accès au certificat de signataire de document applicable à chaque DVLM-e.

Les certificats de signataire de document des participants au répertoire de clés publiques (RCP) DEVRAIENT aussi être communiqués par l'émetteur du certificat à l'OACI en vue de leur publication dans le RCP de l'OACI.

3.1.3 Signataires de SDL2

Un signataire de SDL2 signe numériquement les objets de données de SDL2 d'un ou de plusieurs types.

Lorsqu'il est nécessaire de désigner un signataire de SDL2 comme étant celui qui signe un type d'objet de données de SDL2 particulier, il est désigné comme suit :

- le signataire de SDL2-TS – signe les tampons de voyage de la SDL2 ;
- le signataire de SDL2-V – signe les visas électroniques de la SDL2 ;
- le signataire de SDL2-B – signe les éléments biométriques supplémentaires de LDS2.

Il est RECOMMANDÉ que chaque État n'ait pas plus d'un signataire de SDL2-TS, un signataire de SDL2-V et un signataire de SDL2-B. Il est également possible pour un signataire de SDL2 de combiner certains ou tous ces rôles.

Si une différenciation supplémentaire est nécessaire, comme l'endroit où un tampon de voyage a été ajouté, l'agent individuel qui a autorisé un voyageur, l'agent qui a accordé un visa, ou l'endroit où des éléments biométriques supplémentaires ont été ajoutés, elle peut être incluse dans un champ propriétaire au sein de l'objet de données SDL2 respectif.

3.1.4 Signataires de code à barres

Il est RECOMMANDÉ que les paires de clé de signataire de code à barres (KPu_{BCS}, KPr_{BCS}) soient générées et stockées dans une infrastructure hautement protégée.

La clé privée du signataire de code à barres (KPr_{BCS}) est utilisée pour signer les données (en-tête et message) codées dans le code à barres. La signature est également enregistrée dans le code à barres.

Les certificats de signataire de code à barres (C_{BCS}) sont utilisés pour valider les données (en-tête et message) codées dans le code à barres.

Chaque certificat de signataire de code à barres (C_{BCS}) DOIT être conforme au profil de certificat défini dans la section 7. Les certificats de signataire de code à barres ne sont pas contenus dans le cachet numérique. Par conséquent, un pays qui émet des documents protégés par des cachets numériques DOIT publier tous ses certificats de signataire de code à barres. Le principal canal de distribution des certificats de signataire de code à barres est le RCP/bilatéral. D'autres mécanismes, par exemple la publication sur un site web, sont des canaux secondaires.

Les certificats de signataire de code à barres des participants au RCP DEVRAIENT aussi être communiqués par l'émetteur du certificat à l'OACI en vue de leur publication dans le RCP de l'OACI.

Le signataire de visa (VS) et le signataire de document de voyage d'urgence sont des cas particuliers de signataire de code à barres.

3.1.5 Système d'inspection

Les systèmes d'inspection effectuent une authentification passive pour s'assurer de l'intégrité et de l'authenticité des données stockées dans le CI sans contact du DVLM-e. Dans le cadre de ce processus, les systèmes d'inspection DOIVENT effectuer la validation d'itinéraire de certification indiquée à la section 6.

3.1.6 Signataire de liste de contrôle

La clé privée de signataire de liste de contrôle est employée pour signer les listes de contrôle de l'ACSN.

Les certificats de signataire de liste de contrôle sont employés pour valider les listes de contrôle de l'ACSN.

3.1.7 Signataire de liste d'écarts

La clé privée de signataire de liste d'écarts est employée pour signer les listes d'écarts.

Les certificats de signataire de liste d'écarts sont employés pour valider les listes d'écarts.

3.2 ICP d'autorisation

L'application SDL2 est écrite sur le CI sans contact d'un DVLM-e par l'État émetteur ou l'organisation émettrice au moment de la personnalisation.

Avant qu'un autre État puisse écrire des objets de SDL2 sur ce CI sans contact, il DOIT obtenir l'autorisation de l'État émetteur ou de l'organisation émettrice pour le faire. Chaque objet de données de SDL2 est signé numériquement par un signataire de SDL2 par l'État d'écriture et ensuite écrit sur le CI sans contact par un terminal autorisé dans cet État d'écriture. Le processus en deux étapes de signature par un signataire et d'écriture par un terminal autorisé est similaire au concept SDL1 dans lequel le signataire de document signe numériquement les objets de sécurité du document ; mais ces objets sont ensuite écrits sur le CI sans contact par le processus de personnalisation, comme illustré dans la Figure 1. La lecture ultérieure des objets de SDL2 à partir du CI sans contact se fait par l'intermédiaire de terminaux autorisés à lire les SDL2 du type d'objet de SDL2 en question.

L'ICP d'autorisation permet à l'État émetteur ou à l'organisation émettrice du DVLM-e de contrôler l'accès (lecture et écriture) aux données de SDL2 sur les CI sans contact dans les DVLM-e qu'il émet.

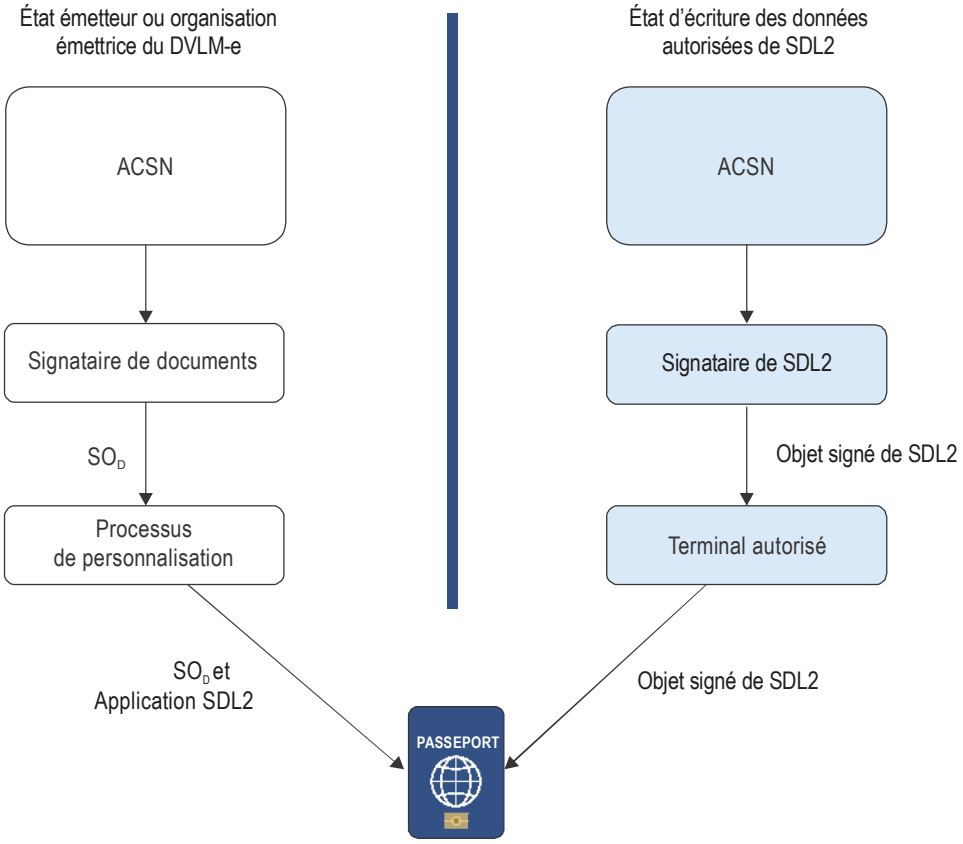

Figure 1. Modèle de confiance et architecture d'écriture SDL2

3.2.1 Autorité de certification de vérification nationale (CVCA)

Chaque État émetteur ou organisation émettrice qui autorise l'ajout de données de SDL2 à ses DVLM-e DOIT mettre en place une seule autorité de certification de vérification nationale (CVCA). Cette CVCA est une autorité de certification (AC) qui constitue l'ancre de confiance pour l'ICP d'autorisation de cet État ou de cette organisation et couvre toutes les applications SDL2. La CVCA peut être une entité autonome ou être intégrée à l'ACSN de ce même État ou organisation. Toutefois, même si elle est située au même endroit, la CVCA DOIT utiliser une paire de clés différente de celle de l'ACSN. La CVCA détermine les droits d'accès qui seront accordés à tous les vérificateurs de documents (DV), étrangers et nationaux, et émet des certificats contenant les autorisations individuelles à chacun de ces DV.

3.2.2 Vérificateur de documents (DV)

Un vérificateur de documents (DV) est une AC qui, dans le cadre d'une unité organisationnelle, gère un groupe de terminaux (p. ex. des terminaux exploités par la police des frontières d'un État) et émet des certificats d'autorisation à ces terminaux. Un DV DOIT avoir déjà reçu un certificat d'autorisation de la CVCA compétente avant de pouvoir émettre des certificats associés à ses terminaux. Les certificats émis par un DV à des terminaux PEUVENT contenir la même autorisation, ou un sous-ensemble, qui a été accordée au DV. Ils NE DOIVENT PAS contenir d'autorisation allant au-delà de celle accordée au DV.

3.2.3 Système d'inspection/terminaux

Dans le contexte de l'ICP d'autorisation, un terminal est l'entité qui accède au CI sans contact d'un DVLM-e et écrit un objet de données de SDL2 signé numériquement, ou lit un objet de données de SDL2. Le terminal DOIT disposer d'un certificat d'autorisation émis par son DV local, qui lui accorde l'autorisation requise. Le terminal est aussi appelé système d'inspection.

3.2.4 Point unique de contact (SPOC)

Chaque État qui participe à l'ICP d'autorisation du SDL2 DOIT créer un SPOC unique. Ce SPOC est l'interface utilisée pour toute communication entre la CVCA d'un État et les DV d'un autre État. Les demandes de certificat et les réponses sont communiquées entre les SPOC de chaque État au moyen du protocole SPOC défini à la section 8.

4. GESTION DES CLÉS

La gestion des clés est définie séparément pour les deux infrastructures à clés publiques.

4.1 ICP du DVLM-e

Les États émetteurs et les organisations émettrices DOIVENT avoir accès à au moins deux types de paires de clés :

- une paire de clés de l'ACSN ;
- une paire de clés de signataire de document.

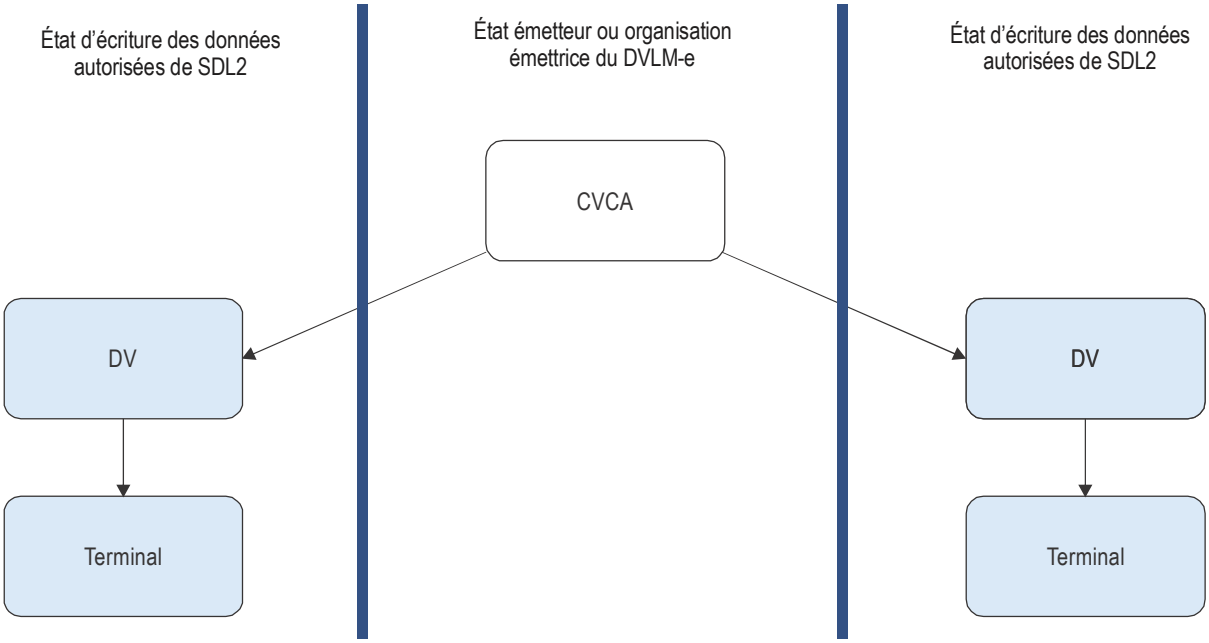

Figure 2. Modèle de confiance de l'ICP d'autorisation

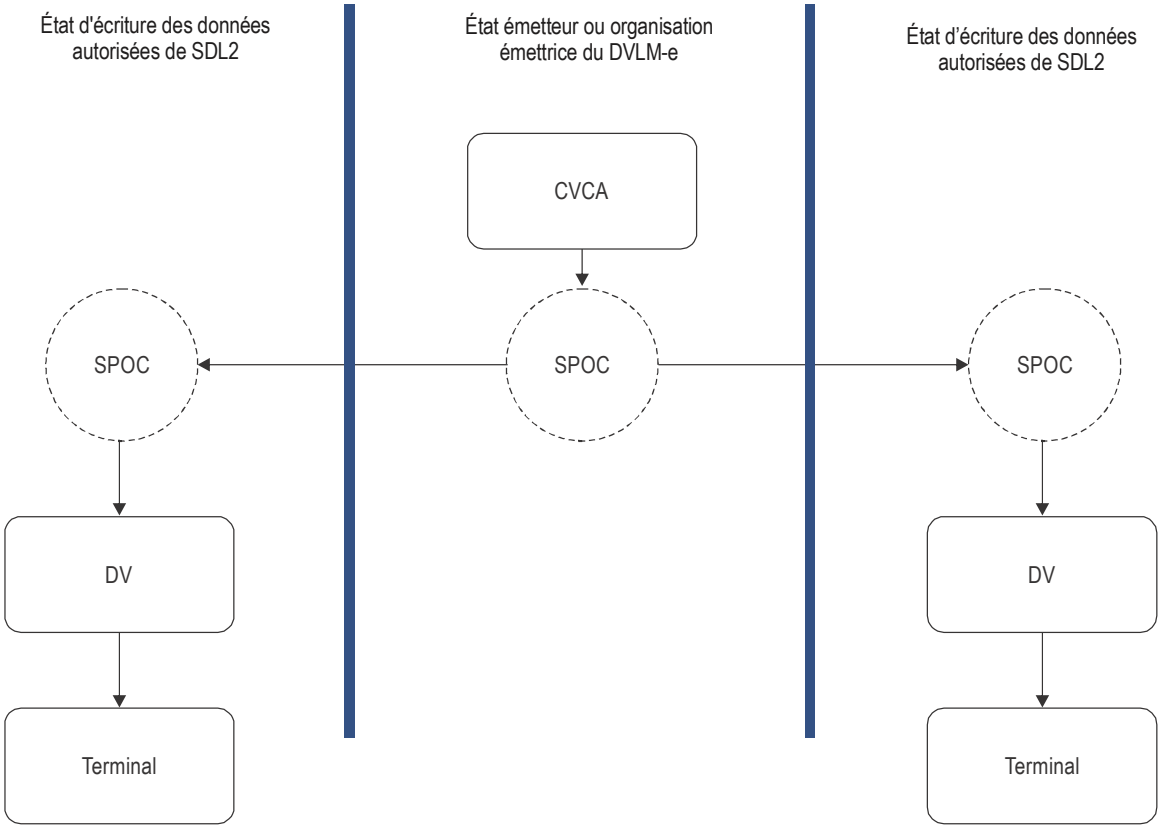

Figure 3. Rôle du SPOC

Les États émetteurs et les organisations émettrices PEUVENT avoir des types de paires de clés supplémentaires :

- une paire de clés de signataire de liste de contrôle ;
- une paire de clés de signataire de liste d'écarts ;
- une paire de clés de signataire de SDL2 ;
- une paire de clés client SPOC ;
- une paire de clés serveur SPOC ;
- une paire de clés de signataire de visa/signataire de document de voyage d'urgence (toutes deux sont des types de signataires de code à barres).

Les clés publiques de l'ACSN, de certificat de signataire et de certificat SPOC sont émises au moyen de certificats X.509. Les clés publiques contenues dans les certificats de l'ACSN sont utilisées pour vérifier la signature de l'ACSN sur les certificats de signataires, SPOC, ACSN émis et les CRL émises.

Dans le cas des clés et des certificats des signataires de listes de contrôle, des signataires de listes d'écarts et des communications, la durée de vie des clés privées et la période de validité des certificats sont laissées à la discrétion de l'État émetteur ou de l'organisation émettrice.

Les certificats de l'ACSN et les certificats de signataire de document sont associés à l'utilisation d'une clé privée et à une période de validité de clé publique, comme le montre le Tableau 1.

4.1.1 Clés et certificats des signataires de documents

La période d'utilisation d'une clé privée de signataire de document est de beaucoup inférieure à la période de validité du certificat de signataire de document pour la clé publique correspondante.

4.1.1.1 Validité des clés publiques des signataires de documents

La durée de vie, c'est-à-dire la période de validité du certificat, de la clé publique de signataire de document est déterminée par la combinaison des deux périodes suivantes :

- la période de temps pendant laquelle la clé privée correspondante sera utilisée pour l'émission de DVLM-e ;
- la période de validité la plus longue de tout DVLM-e émis avec cette clé[1].

1. Certains États émetteurs et certaines organisations émettrices peuvent émettre des DVLM-e avant que ceux-ci ne deviennent valides, par exemple, en cas de changement de nom lors d'un mariage. Dans ces cas, la période de validité la plus longue d'un DVLM-e comprend la période de validité réelle du DVLM-e (c'est-à-dire 10 ans) à laquelle s'ajoute le temps maximal entre le moment où le DVLM-e est émis et le moment où il devient valide.

Tableau 1. Utilisation et validité des clés

	Utilisation de la clé privée	***Validation de la clé publique (en supposant une durée de validité de 10 ans pour les passeports)***
ACSN	3-5 ans	13-15 ans
Signataire de document	Jusqu'à 3 mois[2]	Environ 10 ans
Signataire de SDL2-TS	1-2 ans	10 ans + 3 mois
Signataire de SDL2-V	1-2 ans	10 ans + 3 mois
Signataire de SDL2-B	1-2 ans	10 ans + 3 mois
Client SPOC	Non spécifié	6-18 mois
Serveur SPOC	Non spécifié	6-18 mois
Signataire de code à barres de visa	1-2 ans	Durée d'utilisation de la clé privée + validité du visa
Signataire de code à barres de document de voyage d'urgence	1 an + 2 mois (les 2 mois sont destinés à un renouvellement en douceur)	Durée d'utilisation de la clé privée + période de validité du document de voyage d'urgence
Signataire de liste de contrôle	À la discrétion de l'État émetteur ou de l'organisation émettrice	À la discrétion de l'État émetteur ou de l'organisation émettrice
Signataire de liste d'écarts	À la discrétion de l'État émetteur ou de l'organisation émettrice	À la discrétion de l'État émetteur ou de l'organisation émettrice
Communication	À la discrétion de l'État émetteur ou de l'organisation émettrice	À la discrétion de l'État émetteur ou de l'organisation émettrice

Le certificat de signataire de document (C_{DS}) DOIT être valide pendant la totalité de cette période pour que l'authenticité des DVLM-e puisse être vérifiée. Cependant, la clé privée correspondante ne DEVRAIT être utilisée que pour émettre des documents pendant une période limitée ; une fois que le dernier document pour l'émission duquel elle a été utilisée a expiré, la clé publique n'est plus requise.

4.1.1.2 Période d'émission des clés privées des signataires de documents

Dans le déploiement de leurs systèmes, les États émetteurs et les organisations émettrices trouveront peut-être utile de tenir compte du nombre de documents qui seront signés par chaque clé privée de signataire de document.

Un État émetteur ou une organisation émettrice peut déployer plus d'un signataire de document, chacun ayant sa propre paire de clés unique, qui sont actives à tout moment.

2. À noter que l'extension `privateKeyUsage` correspondante dans le certificat de signataire de document peut être légèrement plus longue pour satisfaire aux spécifications de chevauchement ou de production.

Afin de réduire au minimum les coûts de continuité des activités en cas de révocation d'un certificat de signataire de document, un État émetteur ou une organisation émettrice qui émet un grand nombre de DVLM-e par jour peut :

- employer une période très courte d'utilisation de clé privée ; et/ou
- déployer plusieurs signataires de documents qui sont actifs en même temps, chacun possédant sa propre clé privée unique et son propre certificat de clé publique.

Un État émetteur ou une organisation émettrice qui émet un petit nombre de DVLM-e par jour peut ne déployer qu'un seul signataire de document et peut aussi aisément appliquer une période d'utilisation de clé privée légèrement plus longue.

Quel que soit le nombre de DVLM-e émis par jour, ou le nombre de signataires de documents actifs en même temps, il est RECOMMANDÉ que la période maximale d'utilisation d'une clé privée de signataire de document pour signer un DVLM-e soit de trois mois.

Une fois que le dernier document signé à l'aide d'une clé privée donnée a été produit, il est RECOMMANDÉ que les États émetteurs ou les organisations émettrices effacent la clé privée d'une manière qui permette l'audit et la comptabilisation.

4.1.2 Clés et certificats des signataires de SDL2

Les paires de clés de signataire de SDL2 sont similaires aux paires de clés de signataire de document dans la mesure où la période d'utilisation de la clé privée est beaucoup plus courte que la période de validité du certificat correspondant. Les certificats DOIVENT rester valides pendant la durée de vie du DVLM-e ou de l'objet de SDL2 signé (la durée la plus longue étant retenue). Étant donné que des objets de données signés seront écrits sur les DVLM-e à partir de divers États, ces certificats DOIVENT être valides au moins pour la durée de vie du plus long des DVLM-e (c.-à-d. 10 ans).

4.1.2.1 Validité des clés publiques des signataires de SDL2

La durée de vie, c'est-à-dire la période de validité du certificat de la clé publique de signataire de SDL2 est déterminée par la concaténation des deux périodes suivantes :

- La période de temps pendant laquelle la clé privée correspondante sera utilisée pour signer des objets de SDL2 ;
- La période de validité la plus longue parmi les éléments suivants :
  - Tout DVLM-e qui stockera un objet de SDL2 signé avec cette clé ; ou
  - Tout objet de SDL2 signé avec cette clé. Notez que dans le cas du visa électronique de SDL2, il est possible que la période de validité d'un VISA électronique signé s'étende au-delà de la période de validité du DVLM-e incluant ce visa.

4.1.3 Clés et certificats de signataire de code à barres

Un signataire de code à barres est un type spécifique de serveur de signature utilisé pour signer une catégorie unique de type de document, par exemple un visa, un document de voyage d'urgence, etc. Pour suivre les meilleures pratiques dans le domaine, il est RECOMMANDÉ de n'utiliser qu'un nombre limité de clés de signature (un nombre à un chiffre,

vers le bas) en parallèle pour créer des signatures pour les cachets numériques, à moins que les exigences opérationnelles ne rendent un plus grand nombre de clés absolument nécessaire. Pour garantir la disponibilité du signataire de code à barres en cas d'incident de sécurité lié aux clés de signature, il est RECOMMANDÉ de mettre en place des mesures pour assurer la continuité des activités (p. ex. préparation de clés de secours, site de secours, etc.).

Afin de faciliter le traitement des certificats correspondants (voir la section 5), le nombre de clés de validation de signature publiées DOIT être limité à cinq clés de signature par an.

4.1.3.1 Validité de la clé publique de signataire de code à barres

La présente section s'applique à tous les signataires de code à barres, notamment le signataire de visa et le signataire de document de voyage d'urgence.

La durée de vie, c'est-à-dire la période de validité du certificat de la clé publique de signataire de code à barres, est déterminée par la concaténation des deux périodes suivantes :

- la période de temps pendant laquelle la clé privée correspondante sera utilisée pour l'émission d'un visa ou d'un document de voyage d'urgence ;
- la période de validité la plus longue de tout document émis avec cette clé[3].

Le certificat de signataire de code à barres DOIT être valide pendant la totalité de cette période pour que l'authenticité du document puisse être vérifiée. Cependant, la clé privée correspondante ne DEVRAIT être utilisée que pour émettre des documents pendant une période limitée ; une fois que le dernier document pour l'émission duquel elle a été utilisée a expiré, la clé publique n'est plus requise.

Temps d'utilisation de la clé privée :	selon le profil du document
Validité du certificat :	durée d'utilisation de la clé privée + période de validité du document

Exemple

Note.— Les périodes de validité réelles utilisées pour le calcul dans cet exemple n'impliquent aucune recommandation.

Supposons que des documents ayant une période de validité de cinq ans soient émis, et que la durée d'utilisation de la clé privée du certificat de signataire de code à barres soit d'un an. Alors, la validité du certificat de signataire de code à barres est de 1 + 5 = 6 ans. Si la durée d'utilisation de la clé privée du certificat de l'ACSN est de trois ans, alors la validité du certificat de l'ACSN est de 3 + 6 = 9 ans.

4.1.4 Clés et certificats de l'ACSN

La période d'utilisation d'une clé privée de l'ACSN est beaucoup plus courte que la période de validité du certificat de l'ACSN pour la clé publique correspondante.

3. Certains États émetteurs et certaines organisations émettrices peuvent émettre des DVLM-e avant que ceux-ci ne deviennent valides, par exemple, en cas de changement de nom lors d'un mariage. Dans ces situations, la « période de validité la plus longue d'un DVLM-e » inclut la validité réelle du DVLM-e (p. ex. 10 ans) plus le temps maximal entre l'émission du DVLM-e et la date de validité.

4.1.4.1 Validité des clés publiques de l'ACSN

La durée de vie, c'est-à-dire la validité du certificat, de la clé publique de l'ACSN est déterminée par la concaténation des périodes suivantes :

- la période de temps pendant laquelle la clé privée de l'ACSN correspondante sera utilisée pour signer tout certificat relevant de l'ACSN ;
- la durée de vie maximale de la clé de tout certificat émis relevant de l'ACSN.

4.1.4.2 Période d'émission de clé privée de l'ACSN

La période d'utilisation de la clé privée de l'ACSN pour signer les certificats et les CRL est un équilibre délicat entre les facteurs suivants :

- dans le cas improbable de compromission de la clé privée de l'ACSN d'un État émetteur ou d'une organisation émettrice, la validité de tous les DVLM-e émis en utilisant les clés de signataire de document dont les certificats ont été signés par la clé privée de l'ACSN compromise est mise en doute. Par conséquent, les États émetteurs et les organisations émettrices voudront PEUT-ÊTRE utiliser une période d'émission assez courte ;
- le fait de maintenir la période d'émission très courte conduit cependant à avoir un très grand nombre de clés publiques de l'ACSN valides à tout moment. Cela peut conduire à une gestion plus complexe des certificats au sein des systèmes de traitement des frontières.

Il est donc RECOMMANDÉ que la paire de clés de l'ACSN d'un État émetteur ou d'une organisation émettrice soit remplacée tous les trois à cinq ans.

4.1.4.3 Remplacement de clé de l'ACSN

Les clés de l'ACSN font office de points de confiance dans l'ensemble du système et sans elles le système s'effondrerait. Les États émetteurs et les organisations émettrices DEVRAIENT donc soigneusement planifier le remplacement de la paire de clés de leur ACSN. Une fois que la période d'émission de la clé privée de signature de l'ACSN initiale est écoulée, un État émetteur ou une organisation émettrice aura toujours au moins deux certificats ACSN (C_{ACSN}) valides à tout moment.

Les États émetteurs et les organisations émettrices DOIVENT notifier les États récepteurs qu'un renouvellement de clé de l'ACSN est prévu. Cette notification DOIT être communiquée 90 jours avant le renouvellement de la clé. Une fois la clé renouvelée, le nouveau certificat de l'ACSN (certifiant la nouvelle clé publique de l'ACSN) est distribué aux États récepteurs.

Si le certificat de l'ACSN est un nouveau certificat autosigné, il devrait être authentifié en utilisant une méthode hors bande.

Lors d'un renouvellement de clé de l'ACSN, un certificat reliant la nouvelle clé à l'ancienne clé DOIT être émis de manière à assurer une transition sûre pour les parties de confiance. Il s'agit généralement d'émettre un certificat autoémis (*self-issued-certificate*) ; les champs émetteur et sujet sont identiques mais la clé utilisée pour vérifier la signature représente l'ancienne paire de clés et la clé publique certifiée représente la nouvelle paire de clés. Il n'est pas nécessaire que ces certificats de liaison de l'ACSN soient vérifiés par une méthode hors bande vu que la signature du

certificat de liaison de l'ACSN est vérifiée au moyen d'une clé publique déjà fiable pour cet ACSN. Les listes de contrôle peuvent aussi être utilisées pour distribuer les certificats de liaison et les certificats racines autosignés de l'ACSN.

Les États émetteurs et les organisations émettrices devraient éviter d'utiliser leur nouvelle clé privée de l'ACSN durant les deux premiers jours après le renouvellement de la clé de l'ACSN afin de s'assurer que le nouveau certificat de clé publique de l'ACSN a effectivement été distribué.

Les États émetteurs et les organisations émettrices DOIVENT utiliser la clé privée de l'ACSN la plus récente pour signer tous les certificats, ainsi que pour signer les CRL.

4.1.5 Révocation de certificat

Il peut être nécessaire pour les États émetteurs ou les organisations émettrices de révoquer des certificats en cas d'incident (tel que la compromission d'une clé).

Toutes les ACSN DOIVENT produire périodiquement des informations de révocation sous forme de listes des certificats révoqués (CRL).

Les ACSN DOIVENT émettre au moins une CRL tous les 90 jours, même si aucun certificat n'a été révoqué depuis la dernière CRL émise. Les CRL PEUVENT être émises plus souvent que tous les 90 jours, mais pas plus fréquemment que toutes les 48 heures.

Si un certificat est révoqué, une CRL indiquant cette révocation doit être produite dans les 48 heures.

Seuls les certificats peuvent être révoqués, non les objets de sécurité du document. L'emploi de CRL se limite à des notifications de certificats révoqués qui ont été émis par l'ACSN qui a émis la CRL (y compris les notifications de révocation des certificats de l'ACSN, des certificats de signataire de document, des certificats de signataire de liste de contrôle, des certificats de signataire de liste d'écarts et de tout autre type de certificat émis par cette AC).

Les CRL subdivisées ne sont pas utilisées dans l'application DVLM-e. Tous les certificats révoqués par une ACSN, notamment les certificats de signataire de document, les certificats de l'ACSN, les certificats de signataire de liste de contrôle et les certificats de signataire de liste d'écarts, figurent sur la même CRL. Même si la CRL est toujours signée avec la clé privée de signature de l'ACSN la plus récente (en vigueur), la CRL comprend des notifications de révocation pour les certificats signés avec cette même clé privée ainsi que les certificats signés avec des clés privées de signature de l'ACSN précédentes.

4.1.5.1 Révocation de certificats de l'ACSN

La révocation d'un certificat de l'ACSN est une mesure à la fois extrême et difficile. Dès qu'un État récepteur est informé de la révocation d'un certificat de l'ACSN, tous les autres certificats signés à l'aide de la clé privée de l'ACSN correspondante sont effectivement révoqués.

Lorsqu'un certificat de liaison de l'ACSN a été signé en utilisant une ancienne clé privée de l'ACSN pour certifier une nouvelle clé publique de l'ACSN (voir Remplacement de clé de l'ACSN, § 4.1.4.3), la révocation de l'ancien certificat de l'ACSN DOIT également révoquer le nouveau certificat de l'ACSN.

Si un certificat de l'ACSN doit être révoqué, l'ACSN peut émettre une CRL signée avec la clé privée qui correspond à la clé publique en cours de révocation vu qu'il s'agit de la seule clé que les utilisateurs de la CRL pourront vérifier à ce moment. La clé publique de l'ACSN ne doit être considérée valide que pour la vérification de la signature de cette CRL.

Une fois que l'utilisateur de la CRL a vérifié la signature de la CRL, la clé privée de signature de l'ACSN est considérée comme compromise et le certificat est révoqué pour toutes les vérifications futures.

Pour émettre de nouveaux documents, l'État émetteur ou l'organisation émettrice DOIT réamorcer son processus d'authentification depuis le début, en émettant un nouveau certificat racine de l'ACSN, en distribuant ce certificat aux États récepteurs et en confirmant par un moyen hors bande que le certificat reçu par chaque État récepteur est en fait le certificat de l'ACSN authentique en vigueur.

4.1.5.2 Révocation d'autres certificats

Lorsqu'un État émetteur ou une organisation émettrice souhaite révoquer un certificat de signataire émis en vertu de l'ACSN, il n'est pas nécessaire d'attendre la prochaine période d'actualisation `nextUpdate` prévue dans la CRL en vigueur pour émettre une nouvelle CRL. Il est RECOMMANDÉ qu'une nouvelle CRL soit émise dans un délai de 48 heures après la notification de la révocation.

4.1.6 Algorithmes cryptographiques

Un État émetteur ou une organisation émettrice PEUT prendre en charge différents algorithmes à utiliser dans ses clés de l'ACSN et de certificat de signature. Par exemple, l'ACSN peut avoir été émise en utilisant RSA (Rivest, Shamir et Adleman), mais les certificats de signataires peuvent être des DSA à courbe elliptique (ECDSA) et vice versa.

Les États émetteurs et les organisations émettrices DOIVENT choisir des longueurs de clé appropriées, qui assurent une protection contre les attaques. Des catalogues de cryptographiques appropriés DEVRAIENT être pris en compte.

Les États récepteurs DOIVENT prendre en charge tous les algorithmes aux points où ils souhaitent valider la signature sur les DVLM-e.

Les États émetteurs ou les organisations émettrices DOIVENT prendre en charge un des algorithmes ci-dessous pour leur ACSN, leurs clés de signature et, s'il y a lieu, leurs objets de sécurité de document.

4.1.6.1 RSA

Les États émetteurs ou les organisations émettrices qui mettent en œuvre l'algorithme RSA pour la génération de signatures, la vérification des certificats et l'objet de sécurité de document (SO_D) DOIVENT utiliser la norme RFC 4055. Cette norme spécifie deux mécanismes de signature : RSASSA-PSS et RSASSA-PKCS1_v15. Il est RECOMMANDÉ que les États émetteurs ou les organisations émettrices utilisent RSASSA-PSS pour générer les signatures, mais les États récepteurs DOIVENT aussi être prêts à vérifier les signatures au moyen de RSASSA-PKCS1_v15.

4.1.6.2 Algorithme de signature numérique (DSA)

Les États émetteurs ou les organisations émettrices qui mettent en œuvre le DSA pour la génération et la vérification des signatures DOIVENT utiliser la norme FIPS 186-4.

4.1.6.3 DSA à courbe elliptique (ECDSA)

Les États émetteurs ou les organisations émettrices qui mettent en œuvre ECDSA pour la génération et la vérification des signatures DOIVENT utiliser la norme X9.62 ou la norme ISO/IEC 15946. Les paramètres de domaine de la courbe

elliptique utilisés pour générer la paire de clés ECDSA DOIVENT être explicitement décrits dans les paramètres de la clé publique, c'est-à-dire que les paramètres DOIVENT être du type ECParameters (pas de courbes nommées, pas de paramètres implicites) et DOIVENT inclure le cofacteur optionnel. Les points de courbe elliptique ECPoints DOIVENT être en format non compressé.

Il est RECOMMANDÉ de suivre la ligne directrice TR 03111.

4.1.6.4 Algorithmes de hachage

Les algorithmes de hachage SHA-224, SHA-256, SHA-384 et SHA-512 sont les seuls permis. Voir FIPS 180-2.

4.1.7 Algorithmes cryptographiques pour les certificats de signataires de SDL2

Les certificats de SDL2 et les objets signés étant stockés sur le CI sans contact, ils doivent être aussi compacts que possible. Par conséquent, les signataires de SDL2 DOIVENT utiliser ECDSA, quel que soit l'algorithme utilisé dans les clés de signature de l'ACSN et du document.

4.2 ICP d'autorisation

Les États émetteurs ou les organisations émettrices qui mettent en œuvre SDL2 DOIVENT disposer des types de paires de clés suivants :

- une paire de clés de l'autorité de certification de vérification nationale (CVCA) ;
- une paire de clés du vérificateur de documents (DV) ;
- une paire de clés du terminal.

Les clés publiques de la CVCA et du DV sont certifiées par la CVCA. Les clés publiques du terminal sont certifiées par le DV. Les certificats de clé publique de la CVCA, du DV et du terminal sont des certificats vérifiables par carte qui DOIVENT être conformes à leurs profils de certificat respectifs définis à la section 7. Il n'existe pas de mécanisme de révocation pour les certificats de la CVCA, du DV ou du terminal. Par conséquent, leurs périodes de validité sont beaucoup plus courtes que celles des types de certificats X.509.

La période d'utilisation de la clé privée n'est pas spécifiée et est laissée à la discrétion de l'État. Cependant, la période d'utilisation de la clé privée DOIT être au plus égale à la période de validité de la clé publique. La période de validité de la clé publique pour les paires de clés de la CVCA, du DV et du terminal est indiquée dans le Tableau 2.

Tableau 2. Utilisation des clés Validité du certificat vérifiable par la carte

	Validation de la clé publique
CVCA	de 6 mois à 3 ans
DV	de 2 semaines à 3 mois
Terminal	de 1 jour à 1 mois

4.2.1 Algorithmes cryptographiques pour l'authentification des terminaux

L'algorithme utilisé pour l'authentification du terminal dans l'ICP d'autorisation est déterminé par la CVCA de l'État émetteur du DVLM-e. Le même algorithme de signature, les mêmes paramètres de domaine et les mêmes tailles de clé DOIVENT être utilisés dans une chaîne de certificats (c.-à-d. les certificats de la CVCA, du DV et du terminal pour une autorisation donnée). Par conséquent, les vérificateurs de documents et les terminaux devront être munis de plusieurs paires de clés. Les certificats de liaison de la CVCA PEUVENT inclure une clé publique qui s'écarte des paramètres actuels, c'est-à-dire que la CVCA PEUT passer à un nouvel algorithme de signature, à de nouveaux paramètres de domaine ou à des tailles de clé.

Pour l'authentification du terminal, on PEUT utiliser soit RSA soit ECDSA. Voir les détails dans le Doc 9303-11.

4.2.2 Algorithmes cryptographiques pour SPOC

Les suites de chiffrement TLS à utiliser pour le protocole SPOC sont énumérées dans le Tableau 3.

Tableau 3. Suites de chiffrement TLS

Suite chiffrée	Algorithme d'échange de certificats et de clés
TLS_RSA_WITH_AES_128_CBC_SHA	RSA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE_RSA
TLS_RSA_WITH_AES_256_CBC_SHA	RSA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	DHE_RSA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	ECDHE_ECDSA
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	ECDHE_ECDSA

Dans le cadre de la négociation de la prise de contact TLS, le client DOIT prendre en charge toutes les suites de chiffrement TLS définies dans le Tableau 3. Le serveur et le client DOIVENT tous deux prendre en charge l'authentification basée sur RSA et ECDSA. Il est permis à un serveur de demander et au client d'envoyer un certificat client d'un type différent de celui du serveur.

L'utilisation de l'agrément de clé ECDHE_ECDSA dans la prise de contact TLS est conforme aux ajouts définis dans [TLSECC], [TLS1.2] et [TLSEXT]. Le client et le serveur DOIVENT tous deux prendre en charge les extensions de courbes elliptiques appropriées conformément à la spécification [TLSECC] dans le cadre de la prise de contact. Les courbes elliptiques et les formats de points EC pris en charge sont définis dans la section 5 de [TLSECC]. L'utilisation des suites de chiffrement TLS prises en charge, définies dans le Tableau 3, qui utilisent la norme de chiffrement avancée (AES) pour le chiffrement DOIT être conforme à la spécification [TLSAES].

5. MÉCANISMES DE DISTRIBUTION

Pour l'ICP du DVLM-e, les objets de l'ICP doivent être distribués aux États récepteurs. Plusieurs mécanismes de distribution sont employés, selon le type d'objet et les exigences opérationnelles. Il est important de noter que la distribution de ces objets N'ÉTABLIT PAS la confiance dans ces objets, ni dans les clés privées ou publiques qui leur sont associées. Les mécanismes d'établissement de la confiance sont spécifiés à la section 6.1.

Le mécanisme de distribution de l'ICP d'autorisation est traité à la section 8.

Les objets que les États émetteurs ou les organisations émettrices doivent distribuer aux États récepteurs sont notamment :

- les certificats de l'ACSN ;
- les certificats de liaison de l'ACSN ;
- les certificats de signataire de document ;
- les certificats de signataire de SLD2 ;
- les certificats initiaux de la CVCA ;
- les certificats de liaison de la CVCA ;
- les certificats de DV ;
- les certificats de signataire de codes à barres ;
- les CRL (nuls et non nuls) ;
- les certificats de signataire de liste de contrôle, les listes de contrôle ;
- les certificats de signataire de liste d'écarts, listes d'écarts.

Les mécanismes de distribution utilisés dans le DVLM-e et l'ICP d'autorisation incluent :

- le RCP ;
- l'échange bilatéral ;
- le SPOC ;
- les listes de contrôle ;
- les listes d'écarts ;
- le CI sans contact du DVLM-e.

Le Tableau 4 spécifie un mécanisme de distribution principal et un mécanisme de distribution secondaire (s'il y a lieu) pour chaque objet.

Tableau 4. Distribution des objets de l'ICP

	CI sans contact	SPOC	Bilatéral	RCP	Liste d'écarts	Liste de contrôle	Notes
Certificats de l'ACSN			Y (principal)			Y (secondaire)	
Certificats de signataire de document	Y (principal)			Y (secondaire)			Certificats écrits en même temps que le SOD est écrit
Certificats de signataire de SLD2	Y						Certificats écrits en même temps que l'objet signé est écrit
Certificats initiaux de la CVCA	Y						Certificat écrit au moment de la personnalisation du DVLM-e
Certificats de liaison de la CVCA	Y	Y					Certificats distribués aux DV via SPOC et CVCA ancre de confiance mise à jour sur le CI sans contact lors de la vérification suivante
DV Certificats		Y					Distribué uniquement au sujet DV
CRL (nul et non nul)			Y (secondaire)	Y (principal)			Les CRL émises par l'ACSN comprennent des informations de révocation relatives aux objets de l'ICP de SDL2
Certificats de signataire de liste de contrôle						Y	
Certificats de signataire de codes à barres			Y (secondaire)	Y (principal)			Les signataires du code à barres ne sont pas codés dans le code à barres et la distribution doit donc être assurée pour la validation du code à barres
Listes de contrôle			Y	Y			
Certificats de signataire de liste d'écarts					Y		

Du point de vue opérationnel, les États récepteurs ne sont pas tenus d'utiliser à la fois une source principale et une source secondaire. Dans le fonctionnement quotidien d'un système d'inspection, l'autorité d'inspection peut à son gré utiliser la source principale ou la source secondaire. Si l'autorité de l'État récepteur utilise la source secondaire pour un certificat ou une CRL dans ses opérations quotidiennes, il devrait être prêt à prendre en charge aussi la source principale.

Il est nécessaire que les États émetteurs et les organisations émettrices planifient leurs stratégies de renouvellement de paires de clés, tant pour les clés de l'ACSN que pour les clés de signataire, afin que les certificats et les CRL soient distribués en temps utile dans les systèmes de contrôle frontalier des États récepteurs. Idéalement, la distribution aura lieu dans les 48 heures, mais certains États récepteurs peuvent avoir des postes aux frontières avancés isolés et mal reliés pour lesquels la communication des certificats et des CRL demandera plus de temps. Les États récepteurs DEVRAIENT s'efforcer de distribuer ces certificats et ces CRL à tous les postes frontaliers dans les 48 heures.

Les États émetteurs et les organisations émettrices devraient s'attendre à ce que les certificats de l'ACSN (C_{CSCA}) soient distribués par les États récepteurs dans un délai de 48 heures.

Les États émetteurs et les organisations émettrices peuvent s'assurer que les certificats de signataire de document (C_{DS}) ont été communiqués en temps utile en incluant le certificat de signataire de document (C_{DS}) dans l'objet de sécurité du document (SO_D). Ils devraient aussi s'attendre à ce que les certificats de signataire de document (C_{DS}) publiés dans le RCP soient aussi communiqués aux postes frontaliers dans les 48 heures.

Les certificats de signataire de code à barres ne sont pas contenus dans le cachet numérique. Par conséquent, un pays qui émet des documents protégés par des cachets numériques DOIT publier tous ses certificats de signataire de code à barres. Le principal canal de distribution des certificats de signataire de code à barres est le RCP/bilatéral. D'autres mécanismes, par exemple la publication sur un site web, sont des canaux secondaires.

Pour les signataires de codes à barres, la publication DOIT respecter les principes suivants :

- dès qu'un nouveau certificat est créé, il DOIT être publié avec un délai ne dépassant pas 48 heures ;
- les certificats DOIVENT rester publiés jusqu'à leur expiration ou leur révocation.

Les États récepteurs DEVRAIENT tout mettre en œuvre, par voie électronique ou autrement, pour donner suite aux CRL, notamment les CRL émises dans des circonstances exceptionnelles.

On peut s'assurer que les certificats de signataire de liste de contrôle ont été distribués en les incluant dans chaque liste de contrôle.

5.1 Mécanisme de distribution du RCP

L'OACI assure un service de répertoire de clés publiques (RCP). Ce service DOIT accepter les objets d'ICP, notamment les certificats, les CRL et les listes de contrôle, provenant des participants au RCP, les enregistrer dans un répertoire et veiller à ce que tous les États récepteurs y aient accès.

Les certificats ACSN (C_{CSCA}) ne sont pas stockés individuellement dans le cadre du service RCP de l'OACI. Toutefois, ils peuvent être présents dans le RCP s'ils figurent sur les listes de contrôle.

Chaque certificat reste dans le RCP jusqu'à l'expiration de sa période de validité, que la clé privée correspondante soit encore utilisée ou non.

Les certificats, les CRL et les listes de contrôle enregistrés dans le RCP par tous les participants au RCP DOIVENT être mis à la disposition de toutes les parties (y compris les non-participants au RCP) qui ont besoin de ces informations

pour valider l'authenticité et l'intégrité des données de DVLM-e, des objets de SDL2 et des objets de VDS stockés sous forme numérique.

5.1.1 Téléversement dans le RCP

Seuls les participants au RCP PEUVENT téléverser des certificats, des CRL et des listes de contrôle dans le RCP. Tous les certificats et CRL DOIVENT être conformes aux profils décrits à la section 7. Toutes les listes de contrôle DOIVENT être conformes aux spécifications de la section 9.

Le RCP est constitué d'un « répertoire écriture » et d'un « répertoire lecture ». Les participants au RCP DOIVENT utiliser le protocole rapide d'accès à l'annuaire (LDAP) pour téléverser leurs objets dans le répertoire écriture. Une fois que la signature numérique d'un objet a été vérifiée, et que d'autres contrôles de diligence appropriée ont été effectués, l'objet est publié dans le répertoire lecture.

5.1.2 Téléchargement du RCP

Tous les participants et non-participants au RCP DOIVENT avoir accès en lecture à tous les certificats, à toutes les CRL et à toutes les listes de contrôle publiés dans le RCP. Le contrôle d'accès NE DOIT PAS être appliqué à l'accès en lecture au RCP.

Il incombe à l'État récepteur de distribuer à ses systèmes d'inspection les objets téléchargés du RCP et de maintenir un cache des CRL en vigueur avec les certificats nécessaires pour vérifier les signatures sur les données des DVLM-e.

5.2 Mécanisme de distribution par échange bilatéral

Pour les CRL et les certificats de l'ACSN (C_{CSCA}), le canal de distribution principal est l'échange bilatéral entre les États émetteurs ou les organisations émettrices et les États récepteurs. L'échange bilatéral peut aussi être employé pour distribuer les listes de contrôle.

La technologie spécifique employée pour l'échange bilatéral peut varier selon les politiques de chaque État émetteur ou de chaque organisation émettrice qui doit distribuer ses certificats, ses CRL et ses listes de contrôle, et selon les politiques de chaque État récepteur qui doit avoir accès à ces objets. Voici quelques exemples de technologies qui peuvent être utilisées dans les échanges bilatéraux :

- courrier/valise diplomatique ;
- échange de courriels ;
- téléchargement depuis un site web associé à l'ACSN émettrice ;
- téléchargement depuis un serveur LDAP associé à l'ACSN émettrice.

Cette liste n'est pas exhaustive et d'autres technologies peuvent également être employées.

5.3 Mécanisme de distribution des listes de contrôle

Les listes de contrôle sont une technologie de soutien du mécanisme de distribution bilatérale. La distribution des certificats de l'ACSN au moyen des listes de contrôle est donc un sous-ensemble du mécanisme de distribution bilatérale.

Une liste de contrôle est une liste signée numériquement des certificats de l'ACSN bénéficiant de la confiance de l'État récepteur ou de l'organisation réceptrice qui a émis la liste de contrôle. Les certificats racines autosignés de l'ACSN et les certificats de liaison de l'ACSN peuvent être inclus dans une liste de contrôle. La structure et le format de la liste de contrôle sont définis à la section 8. La publication d'une liste de contrôle permet à d'autres États récepteurs ou organisations réceptrices d'obtenir un ensemble de certificats de l'ACSN à partir d'une seule source (l'émetteur de la liste de contrôle) plutôt que d'établir directement un accord d'échange bilatéral avec chacune des autorités ou organisations émettrices représentées sur cette liste.

Un signataire de liste de contrôle est autorisé par une ACSN à grouper, à signer numériquement et à émettre des listes de contrôle. Les listes de contrôle NE DOIVENT PAS être signées et émises directement par l'ACSN elle-même. Les certificats de signataire de liste de contrôle DOIVENT être conformes au profil de certificat défini à la section 7.

Avant d'émettre une liste de contrôle, le signataire de liste de contrôle émetteur DEVRAIT amplement valider les certificats de l'ACSN qui doivent être contresignés, notamment s'assurer que les certificats appartiennent effectivement aux ACSN identifiées. Les procédures employées pour la validation hors bande DEVRAIENT être prises en compte dans les politiques de certification publiées de l'ACSN qui a émis le certificat de signataire de liste de contrôle.

Chaque liste de contrôle DOIT inclure le certificat de signataire de liste de contrôle qui sera utilisé pour vérifier la signature sur cette liste de contrôle ainsi que les certificats de l'ACSN qui a émis le certificat de signataire de liste de contrôle.

Si un État récepteur reçoit de nouveaux certificats de l'ACSN une fois que ses procédures de validation ont été exécutées, il est RECOMMANDÉ qu'une nouvelle liste de contrôle soit assemblée et émise.

L'emploi d'une liste de contrôle améliore l'efficacité de la distribution des certificats de l'ACSN pour les États récepteurs. Cependant, un État récepteur qui utilise des listes de contrôle DOIT quand même déterminer ses propres politiques d'établissement de la confiance dans les certificats figurant sur cette liste (voir la section 6).

6. CONFIANCE ET VALIDATION DE L'ICP

La confiance et la validation de l'ICP diffèrent entre l'ICP du DVLM-e et l'ICP d'autorisation.

6.1 ICP du DVLM-e

Dans l'environnement ICP des DVLM-e, les systèmes d'inspection des États récepteurs jouent le rôle de parties de confiance à l'infrastructure ICP. Le succès de la vérification de la signature numérique sur l'objet de sécurité du document d'un DVLM-e garantit l'authenticité et l'intégrité des données stockées dans le CI sans contact de ce DVLM-e. Ce processus de vérification de la signature exige que la partie de confiance établisse que la clé publique du signataire de document utilisée pour vérifier la signature est elle-même « fiable ».

Les divers mécanismes de distribution définis à la section 5 permettent aux États récepteurs d'avoir accès aux certificats et aux CRL dont ils doivent vérifier les signatures numériques en question. Cependant, ces mécanismes de distribution n'établissent pas la confiance dans les certificats, les CRL ou les clés publiques qui seront employés pour vérifier les signatures sur ces certificats et ces CRL.

Les clés publiques contenues dans les certificats de l'ACSN (C_{CSCA}) sont utilisées pour vérifier la signature sur les certificats et les CRL. Par conséquent, pour accepter un DVLM-e d'un autre État émetteur, l'État récepteur DOIT avoir

déjà placé dans une certaine forme de stockage de confiance, accessible à son système de contrôle frontalier, une copie fiable du certificat de l'ACSN (C_{CSCA}) de l'État émetteur ou de l'organisation émettrice, ou une autre forme d'information d'ancre de confiance pour cette clé publique de l'ACSN établie à partir du certificat.

Il incombe à l'État récepteur d'établir la confiance dans les certificats de l'ACSN (C_{CSCA}) et de stocker les certificats (ou les informations contenues dans les certificats) en tant qu'ancres de confiance de manière sécurisée pour être utilisés par leurs systèmes d'inspection aux frontières.

6.1.1 Gestion des ancres de confiance

Comme le spécifie la norme RFC 5280, il faut établir une ancre de confiance qui puisse être utilisée pour ancrer la procédure de validation d'un signataire de document, d'un signataire de liste de contrôle, d'un signataire de liste d'écarts ou de tout autre type de certificat.

Chaque ancre de confiance est constituée d'une clé publique fiable et des métadonnées correspondantes. Les ancres de confiance DOIVENT inclure, au minimum :

- la clé publique fiable et les paramètres de clé correspondants ;
- l'algorithme de clé publique ;
- le nom du propriétaire de la clé ;
- la valeur de l'extension `SubjectAltName` du certificat de l'ACSN contenant le code à trois lettres de l'autorité attribué par l'OACI de l'autorité ou organisation émettrice. Même si elle n'est pas utilisée dans les procédures de validation de l'itinéraire de certification ou des CRL, elle est utilisée dans l'authentification passive définie dans le Doc 9303-11.

Dans l'application DVLM-e, une ancre de confiance distincte est établie pour chaque clé publique d'une ACSN donnée. Pour la clé publique initiale obtenue d'une ACSN, la confiance DOIT être établie au moyen d'un mécanisme hors bande. Par exemple, si un certificat de l'ACSN a été téléchargé à partir d'un serveur associé à cette ACSN, une communication hors bande (p. ex., téléphone ou courriel) peut être employée pour vérifier si le certificat téléchargé est bien le certificat authentique de cette ACSN. En outre, la partie de confiance pourrait analyser les politiques, les procédures et les pratiques de l'ACSN émettrice pour déterminer si elles sont suffisamment sécurisées pour satisfaire aux spécifications locales d'utilisation des certificats. Une fois que l'ancre de confiance initiale est établie pour une ACSN donnée, le processus pourrait être simplifié pour les clés suivantes pour cette même ACSN. Si l'ACSN émet un certificat de liaison de l'ACSN, la communication hors bande avec l'ACSN pour vérifier l'authenticité du nouveau certificat peut être omise vu que la clé publique à laquelle il est déjà fait confiance pour cette même ACSN est utilisée pour vérifier la signature sur ce certificat de liaison de l'ACSN.

L'information relative à l'ancre de confiance peut être stockée sous forme de copie fiable du certificat de l'ACSN lui-même, ou sous un autre format fiable quelconque.

Vu que les signatures sur les certificats émis par les ACSN doivent pouvoir être vérifiées longtemps après que l'ACSN a actualisé sa paire de clés, un État récepteur aura normalement plus d'une ancre de confiance pour la même ACSN à un moment quelconque. Si une ACSN modifie son nom, certaines de ces ancres de confiance contiendront l'ancien nom de l'ACSN et d'autres contiendront le nouveau nom.

6.1.2 Validation des certificats/CRL et vérification des révocations

Dans le cadre du processus de vérification de l'authenticité et de l'intégrité des objets de données dans l'application DVLM-e (p. ex., objets de sécurité de documents, listes de contrôle, listes d'écarts, etc.), un État récepteur :

- valide le certificat utilisé pour vérifier la signature de l'objet de données (p. ex., certificat de signataire de document, certificat de signataire de la liste de contrôle, certificat de signataire de la liste d'écarts) ;
- valide la CRL utilisée pour vérifier le statut de révocation du certificat en question ;
- traite la CRL pour vérifier le statut de révocation du certificat en question.

Il existe des exemples d'algorithmes pour ces processus, comme ceux qui sont spécifiés dans la norme RFC 5280. Il n'est pas nécessaire que les États récepteurs mettent en œuvre l'algorithme spécifique défini dans la norme RFC 5280, mais ils DOIVENT assurer une fonctionnalité équivalente au comportement externe résultant de cette procédure. N'importe quel algorithme peut être utilisé par une implémentation donnée pourvu qu'elle permette d'obtenir le bon résultat.

L'Appendice D contient des éléments d'orientation pour les États récepteurs qui décident de baser leur algorithme sur celui qui est spécifié dans la norme RFC 5280.

6.1.3 Autorité de validation des codes à barres

L'autorité de validation des codes à barres valide un cachet numérique en appliquant une politique de validation. Le Doc 9303-13 spécifie en détail les critères de validation et les algorithmes permettant de générer un état de validation.

La Figure 4 illustre l'architecture fonctionnelle de l'autorité de validation des codes à barres. L'autorité de validation des codes à barres s'appuie sur un logiciel de validation qui peut être déployé sur tout ordinateur utilisé par les autorités de contrôle frontalier.

Le logiciel de validation est relié à un lecteur qui prend une image du code à barres pour récupérer le code à barres et la ZLA du document, ainsi qu'une image du document pour récupérer sa ZLA. Pour vérifier la validité de la signature du cachet numérique, le logiciel de validation DEVRAIT être synchronisé avec le point de publication de l'ICP au moins toutes les 24 heures pour récupérer les derniers certificats de signataires de codes à barres et les CRL.

Le logiciel de validation des codes à barres décode le cachet numérique et les ZLA de tout document associé (p. ex. un visa ou un passeport), valide la signature du cachet numérique et applique une politique de validation (voir le Doc 9303-13) pour générer un état de validation du document.

Dans les scénarios mobiles, le logiciel de validation peut également être exécuté directement sur un téléphone intelligent. Alors que la validité du cachet peut être vérifiée par le logiciel du téléphone intelligent, la comparaison entre les données (signées) contenues dans le cachet et les ZLA imprimées (par exemple du visa ou du passeport) DOIT être effectuée soit manuellement, soit par reconnaissance optique de caractères des ZLA à partir de l'image capturée, ce dernier point constituant souvent un problème difficile dans la pratique.

Les données suivantes sont traitées par le logiciel de validation des codes à barres :

- Données d'entrée fournies par les lecteurs, par exemple les images de visas ou de passeports ;
- Certificats et CRL.

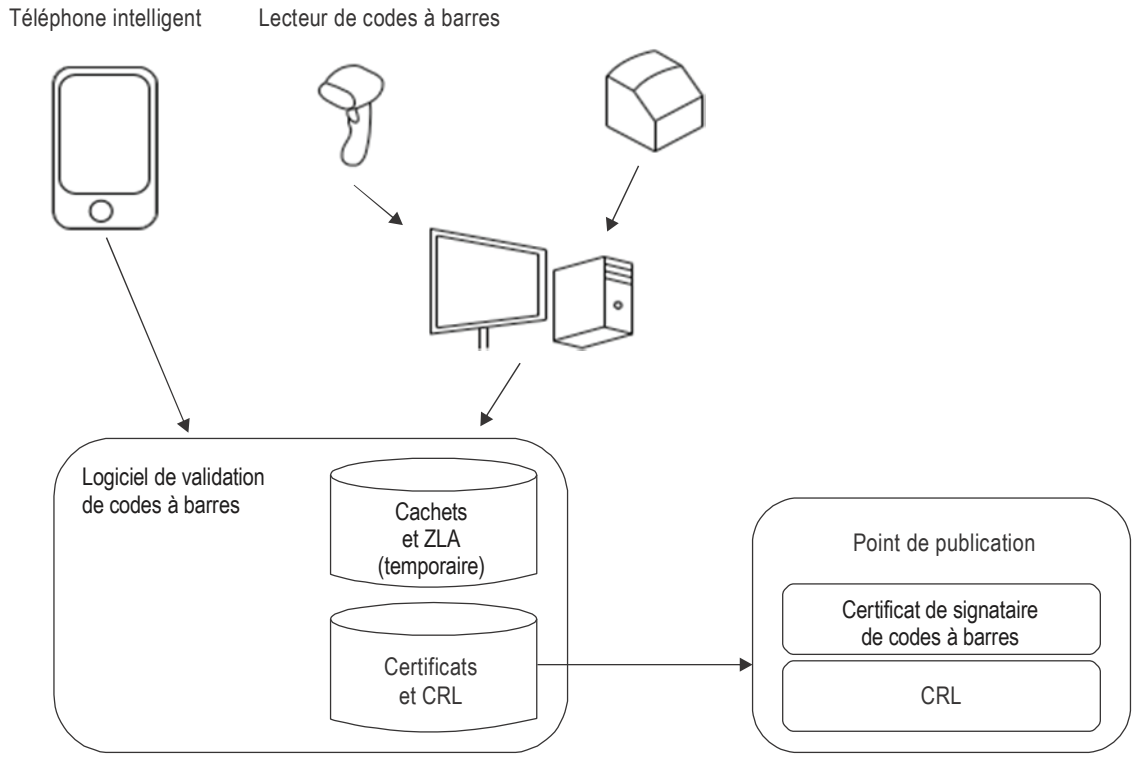

Figure 4. Validation des codes à barres

6.2 ICP d'autorisation

Pour l'ICP d'autorisation, l'ancre de confiance et la validation sont traitées différemment.

6.2.1 Validation des certificats vérifiables par carte

Pour les certificats DV et terminaux dans l'ICP d'autorisation, l'ancre de confiance est la clé publique la plus récente de la CVCA de l'État qui a émis le DVLM-e. L'ancre de confiance initiale DOIT être stockée de manière sécurisée dans le CI sans contact du DVLM-e lors de la phase de production ou de (pré-)personnalisation. Comme la paire de clés utilisée par la CVCA change au fil du temps, des certificats de liaison CVCA sont établis. Le CI sans contact du DVLM-e DOIT mettre à jour dans le programme son ou ses ancres de confiance en fonction des certificats de liaison valides reçus. En raison de l'ordonnancement des certificats de liaison CVCA, deux ancres de confiance CVCA au maximum seront stockées sur le CI sans contact à tout moment.

Pour valider un certificat de terminal, le CI sans contact du DVLM-e DOIT recevoir une chaîne de certificats commençant par une ancre de confiance stockée sur le CI sans contact du DVLM-e.

La procédure de validation des certificats DV et terminaux est spécifique au protocole d'authentification des terminaux SDL2 et est spécifiée dans le Doc 9303-11.

7. PROFILS DE CERTIFICAT ET DE CRL

Les profils de certificat sont définis à la fois pour l'ICP du DVLM-e et l'ICP d'autorisation.

7.1 ICP du DVLM-e

Les États émetteurs ou les organisations émettrices DOIVENT émettre des certificats et des CRL conformes aux profils spécifiés ci-dessous. Tous les certificats et toutes les CRL DOIVENT être produits dans le format des règles de codage distinctives (DER) pour préserver l'intégrité des signatures qu'ils contiennent. Il y a quelques différences entre les profils des certificats de l'ACSN et de signataire de document figurant dans la sixième édition de la présente spécification et les profils actuels. Les systèmes d'inspection DOIVENT être capables de traiter les certificats qui ont été émis conformément aux profils précédents (voir Appendice C) ainsi qu'aux profils actuels.

Ces profils partent du principe que chaque État émetteur ou chaque organisation ou entité émettrice DOIT créer une seule ACSN pour signer tous les DVLM-e conformes au Doc 9303.

Les profils de certificat des types de certificats suivants sont définis dans la présente section :

- AC du pays signataire ;
- Signataire de document ;
- Signataire de liste de contrôle de l'ACSN ;
- Signataire de liste d'écarts ;
- Communication — même si cette étape n'est pas strictement nécessaire actuellement. Il s'agit d'une future étape de vérification. Ces certificats peuvent être utilisés pour accéder au RCP ou pour les communications LDAP/EMAIL/HTTP entre États. Il est recommandé que ces certificats soient émis par l'ACSN.

Les objets de l'ACSN, de signataire de document, de signataire de liste d'écarts et de signataire de liste de contrôle de l'ACSN sont définis à la section 3.

Le profil de CRL est défini au § 7.1.4.

Les profils emploient la terminologie suivante pour la présence de chaque composant/extension :

m obligatoire (*mandatory*) — le champ DOIT être présent ;
x ne pas utiliser — le champ NE DOIT PAS être présent ;
o optionnel — le champ PEUT être présent.
C conditionnel — le champ DOIT être présent sous certaines conditions.

Les profils emploient la terminologie suivante pour la criticité des extensions qui peuvent ou doivent être incluses :

c critique — les applications réceptrices DOIVENT être capables de traiter cette extension ;
nc non critique — les applications réceptrices qui ne comprennent pas cette extension PEUVENT ne pas en tenir compte.

Certaines exigences de ces profils proviennent des profils de base mentionnés (p. ex., RFC 5280). Par commodité, le texte pertinent du profil de base qui s'applique à l'exigence spécifique est reproduit dans un tableau à l'Appendice B.

7.1.1 Profils de certificat

Le Tableau 5 définit les exigences de profil de certificat communes à l'ensemble des certificats des champs du corps du certificat. Le Tableau 6 définit les exigences pour les extensions des certificats.

Tableau 5. Profil des champs des certificats

Composant du certificat	*Présence*	*Observations*
`Certificate`	m	
`TBSCertificate`	m	Voir le Tableau 6.
`signatureAlgorithm`	m	La valeur insérée dépend de l'algorithme choisi.
`signatureValue`	m	La valeur insérée dépend de l'algorithme choisi.
`TBSCertificate`		
`version`	m	DOIT être`v3`.
`serialNumber`	m	DOIT être un entier positif et avoir un maximum de 20 octets. DOIT utiliser le codage en complément à 2 et être représenté par le plus petit nombre d'octets possible.
`signature`	m	La valeur insérée ici DOIT être la même que celle du composant `signatureAlgorithm` de la séquence `Certificate`.
`issuer`	m	S'ils sont présents, `countryName` et `serialNumber` DOIVENT être `PrintableString`. Les autres attributs qui ont la syntaxe `DirectoryString` DOIVENT être soit `PrintableString`, soit `UTF8String`. `countryName` DOIT être en haut de casse. Voir le § 7.1.1.1 pour les conventions de dénomination.
`validity`	m	DOIT se terminer par Zulu (Z). L'élément secondes DOIT être présent. Les dates jusqu'en 2049 DOIVENT être en `UTCTime`. `UTCTime` DOIT être représenté sous la forme AAMMJJHHMMSSZ. Les dates à partir de 2050 DOIVENT être en `GeneralizedTime`.

Composant du certificat	*Présence*	*Observations*
		`GeneralizedTime` NE DOIT PAS avoir de fractions de seconde. `GeneralizedTime` DOIT être représenté sous la forme AAAAMMJJHHMMSSZ.
`subject`	m	S'ils sont présents, `countryName` et `serialNumber` DOIVENT être `PrintableString`. Les autres attributs qui ont la syntaxe `DirectoryString` DOIVENT être soit `PrintableString`, soit `UTF8String`. `countryName` DOIT être en haut de casse. `countryName` dans les champs `issuer` et `subject` DOIVENT correspondre. Voir le § 7.1.1.1 pour les conventions de dénomination.
`subjectPublicKeyInfo`	m	
`issuerUniqueID`	x	
`subjectUniqueID`	x	
`extensions`	m	Voir le Tableau 6 pour les extensions qui devraient être présentes. Les valeurs par défaut des extensions NE DOIVENT PAS être codées.

Tableau 6. Profil des extensions des certificats

Nom de l'extension	*Racines autosignées de l'ACSN*		*Liaison de l'ACSN*		*Signataire de document*		*Signataire de liste de contrôle et signataire de liste d'écarts*		*Communication*		*Observations*
	Présence	*Criticité*	*Présence*	*Criticité*	*Présence*	*Criticité*	*Présence*	*Criticité*	*Présence*	*Criticité*	
`AuthorityKeyIdentifier`	**o**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	
`keyIdentifier`	m		m		m		m		m		
`authorityCertIssuer`	o		o		o		o		o		
`authorityCertSerialNumber`	o		o		o		o		o		
`SubjectKeyIdentifier`	**m**	**nc**	**m**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	

Nom de l'extension	*Racines autosignées de l'ACSN*		*Liaison de l'ACSN*		*Signataire de document*		*Signataire de liste de contrôle et signataire de liste d'écarts*		*Communication*		*Observations*
subjectKeyIdentifier	m		m		m		m		m		
KeyUsage	**m**	**c**	**m**	**c**	**m**	**c**	**m**	**c**	**m**	**c**	
digitalSignature	x		x		m		m		o		Certains certificats de communication (p. ex., certificats TLS) exigent que les bits de keyUsage soient positionnés conformément à la suite de chiffrement particulière utilisée. Certaines suites de chiffrement exigent de positionner le bit digitalSignature et d'autres non.
nonRepudiation	x		x		x		x		x		
keyEncipherment	x		x		x		x		o		
dataEncipherment	x		x		x		x		x		
keyAgreement	x		x		x		x		o		
keyCertSign	m		m		x		x		x		
cRLSign	m		m		x		x		x		
encipherOnly	x		x		x		x		x		
decipherOnly	x		x		x		x		x		
PrivateKeyUsagePeriod	**m**	**nc**	**m**	**nc**	**m**	**nc**	**o**	**nc**	**o**	**nc**	
notBefore	o		o		o		o		o		Au moins un notBefore ou notAfter DOIT être présent. DOIT être codé sous forme de generalizedTime.
notAfter	o		o		o		o		o		
CertificatePolicies	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	
PolicyInformation	m		m		m		m		m		
policyIdentifier	m		m		m		m		m		
policyQualifiers	o		o		o		o		o		
PolicyMappings	**x**		**x**		**x**		**x**		**x**		Voir la note 1.
SubjectAltName	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	Voir le § 7.1.2.
IssuerAltName	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	Voir le § 7.1.2.

Nom de l'extension	***Racines autosignées de l'ACSN***		***Liaison de l'ACSN***		***Signataire de document***		***Signataire de liste de contrôle et signataire de liste d'écarts***		***Communication***		***Observations***
SubjectDirectoryAttributes	**x**		**x**		**x**		**x**		**x**		
Basic Constraints	**m**	**c**	**m**	**c**	**x**		**x**		**x**		
cA	m		m		x		x		x		
PathLenConstraint	m		m		x		x		x		DOIT toujours être 0.
NameConstraints	**x**		**x**		**x**		**x**		**x**		Voir la note 1.
PolicyConstraints	**x**		**x**		**x**		**x**		**x**		Voir la note 1.
ExtKeyUsage	**x**		**x**		**x**		**m**	**c**	**m**	**c**	Voir le § 7.1.3.
CRLDistributionPoints	**m**	**nc**	**m**	**nc**	**m**	**nc**	**m**	**nc**	**o**	**nc**	
distributionPoint	m		m		m		m		m		DOIT être ldap, http ou https. Voir le § 7.1.1.4.
reasons	x		x		x		x		x		
cRLIssuer	x		x		x		x		x		
InhibitAnyPolicy	**x**		**x**		**x**		**x**		**x**		Voir la note 1.
FreshestCRL	**x**		**x**		**x**		**x**		**x**		Voir la note 2.
privateInternetExtensions	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	Voir la note 3.
NameChange	**o**	**nc**	**o**	**nc**	**x**		**x**		**x**		Voir le § 7.1.1.5.
DocumentType	**x**		**x**		**m**	**nc**	**x**		**x**		Voir le § 7.1.1.6.
Netscape Certificate Type	**x**		**x**		**x**		**x**		**x**		Voir la note 4.
other private extensions	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	**o**	**nc**	

Note 1.— Par définition, l'extension ne peut apparaître que dans les certificats d'AC intermédiaires (certificats émis par une AC à une autre AC). Les certificats d'AC intermédiaires ne sont pas employés dans l'infrastructure ICP des DVLM-e. Cette extension est par conséquent interdite dans les certificats des DVLM-e.

Note 2.— L'extension de CRL la plus récente est utilisée pour indiquer une CRL delta. Les CRL delta ne sont pas prises en charge dans l'ICP du DVLM-e. Cette extension est par conséquent interdite.

Note 3.— La norme RFC 5280 définit deux extensions Internet privées (accès aux informations d'autorité et accès aux informations du sujet) qui sont utilisées pour indiquer des informations sur l'émetteur ou le sujet d'un certificat. Ces extensions ne sont pas requises dans l'infrastructure ICP des DVLM-e. Cependant, comme elles n'ont pas d'incidence sur l'interopérabilité et ne sont pas critiques, elles peuvent être incluses à titre facultatif dans les certificats des DVLM-e.

Note 4.— L'extension de type de certificat Netscape peut être utilisée pour limiter les finalités pour lesquelles un certificat peut être utilisé. Les extensions `extKeyUsage` et `basicConstraints` sont désormais les extensions standard pour ces finalités et sont utilisées dans l'application DVLM-e. En raison du conflit potentiel entre les valeurs des extensions standard et de l'extension propriétaire Netscape, l'extension Netscape est interdite.

7.1.1.1 Exigences relatives aux champs émetteur et sujet

Les champs émetteur et sujet sont communs à tous les certificats, mais des restrictions spécifiques s'appliquent aux certificats de signataire SDL2.

7.1.1.1.1 Conditions générales

Les conventions suivantes relatives à la dénomination et à l'adressage des champs `Issuer` et `Subject` sont REQUISES :

- `countryName`. DOIT être présent. La valeur contient le code de pays qui DOIT suivre le format codes de pays à deux lettres, spécifié dans le Doc 9303-3 ;
- `commonName`. DOIT être présent.

D'autres attributs PEUVENT aussi être inclus à la discrétion de l'État émetteur ou de l'organisation émettrice.

7.1.1.1.2 Exigences relatives aux certificats de signataire SDL2

Les certificats de signataires SDL2 DOIVENT être conformes au profil de certificat de signataire de document défini ci-dessus, avec les exceptions définies au § 7.1.2.

7.1.1.2 Exigences relatives aux variantes des noms d'émetteur et de sujet

Vu que les fonctions desservies par les variantes de nom dans l'application DVLM-e sont propres à cette application, et différentes de celles qui sont définies pour l'infrastructure ICP d'Internet dans la norme RFC 5280, les valeurs dans l'extension variante de nom du sujet des certificats des DVLM-e n'identifient pas généralement le sujet du certificat de façon non ambiguë.

Dans l'application DVLM-e, les variantes de nom ont les deux fonctions suivantes.

La première fonction est de fournir les informations de contact avec le sujet et/ou l'émetteur du certificat. Il DEVRAIT donc inclure au moins un des éléments suivants :

- `rfc822Name` ;
- `dNSName` ; ou
- `uniformResourceIdentifier`.

La deuxième fonction est de fournir une chaîne d'annuaire composée des codes de pays assignés par l'OACI. À ces fins, les certificats émis au moyen de ce profil DOIVENT en outre inclure un nom d'annuaire, construit comme suit :

- `localityName` qui contient le code de pays OACI tel qu'il figure dans la ZLA ;
- si ce code de pays ne définit pas de façon univoque l'État émetteur ou l'organisation émettrice, l'attribut `stateOrProvinceName` DOIT être utilisé pour indiquer le code à trois lettres assigné par l'OACI à l'État émetteur ou à l'organisation émettrice ;
- aucun autre attribut n'est permis.

Dans les certificats racines autosignés de l'ACSN, les extensions `IssuerAltName` et `SubjectAltName` DOIVENT être identiques. Dans les certificats de liaison de l'ACSN, les valeurs PEUVENT être différentes. Par exemple, s'il s'est produit un changement dans le nom `rfc822Name` de l'ACSN immédiatement avant l'émission d'un certificat de liaison de l'ACSN, l'extension `IssuerAltName` contiendra l'ancien `rfc822Name` et l'extension `SubjectAltName` contiendra le nouveau `rfc822Name`. Les certificats de liaison de l'ACSN suivants contiendront le nouveau `rfc822Name` dans les deux extensions.

7.1.1.3 Exigences relatives à l'extension d'utilisation de clé étendue

L'identificateur d'objet (OID) qui doit être inclus dans l'extension `extendedKeyUsage` pour les certificats de signataire de liste de contrôle est `2.23.136.1.1.3`.

L'identificateur d'objet (OID) qui doit être inclus dans l'extension `extendedKeyUsage` pour les certificats de signataire de liste d'écarts est `2.23.136.1.1.8`.

Dans les certificats de communication, la valeur de cette extension dépend du protocole de communication utilisé (voir la norme RFC 5280, § 4.2.1.12).

7.1.1.4 Exigences relatives à l'extension de points de distribution de CRL

Les ACSN peuvent publier leur CRL à plusieurs endroits, notamment le RCP, leur propre site web, etc.

Pour les CRL qui sont publiées à des endroits autres que le RCP (p. ex., site web ou serveur LDAP local), les valeurs qui doivent être incluses dans cette extension sont contrôlées par l'ACSN qui émet les certificats et la CRL en question.

Pour les CRL publiées dans le RCP, les participants au RCP PEUVENT inclure deux valeurs URL pour leur CRL en utilisant le modèle suivant [remplacer « CountryCode » (code de pays) par le code à trois lettres assigné par l'OACI à l'État émetteur ou à l'organisation émettrice]. Si ce code de pays n'identifie pas sans ambiguïté l'État émetteur ou l'organisation émettrice, l'entrée est créée en ajoutant le symbole « _ » au code de pays à trois lettres dans la ZLA, puis le code à trois lettres assigné par l'OACI à l'État émetteur ou à l'organisation émettrice et qui identifie sans ambiguïté l'État émetteur ou l'organisation émettrice :

https://pkddownload1.icao.int/CRLs/CountryCode.crl
https://pkddownload2.icao.int/CRLs/CountryCode.crl

Cette extension est obligatoire et les vérifications du statut de révocation sont une partie obligatoire de la procédure de validation. Par conséquent, au moins une valeur DOIT être remplie :

- les valeurs de RCP peuvent être les seules valeurs dans l'extension ;
- il peut y avoir d'autres valeurs (p. ex., une ACSN peut décider de publier sa CRL sur un site web et inclure un pointeur vers cette source) ; ou
- une ACSN peut aussi décider de n'inclure qu'une seule valeur (p. ex., un pointeur vers son site web comme source) même si elle publie aussi sa CRL dans le RCP.

Les exemples suivants illustrent les valeurs de RCP qui seraient indiquées dans les certificats émis par l'autorité émettrice de Singapour et de Hong Kong :

Exemple de RCP pour Singapour :
https://pkddownload1.icao.int/CRLs/SGP.crl
https://pkddownload2.icao.int/CRLs/SGP.crl

Exemple pour Hong Kong :
https://pkddownload1.icao.int/CRLs/CHN_HKG.crl
https://pkddownload2.icao.int/CRLs/CHN_HKG.crl

7.1.1.5 Extension de changement de nom

Après un renouvellement de clé de l'ACSN, un certificat DOIT être émis pour relier l'ancienne clé publique à la nouvelle clé publique de manière à assurer une transition sûre pour les parties de confiance. Il s'agit généralement d'émettre un certificat autoémis ; les champs `issuer` et `subject` sont identiques mais la clé employée pour vérifier la signature représente l'ancienne paire de clés et la clé publique certifiée représente la nouvelle paire de clés.

Il est RECOMMANDÉ que les ACSN ne changent pas leur nom distinctif (DN) inutilement vu qu'il y a des incidences négatives sur les parties de confiance (ils doivent conserver l'ancien nom et le nouveau nom comme ACSN valides pour le même État émetteur ou la même organisation émettrice jusqu'à l'expiration de tous les PLM-e signés avec l'ancien nom). Cependant, si un changement de nom est nécessaire, il DOIT être communiqué aux parties de confiance au moyen de l'émission d'un certificat de liaison de l'ACSN, dans lequel le champ `issuer` contient l'ancien nom et le champ `subject` contient le nouveau nom. Ce certificat de liaison de l'ACSN véhicule aussi un renouvellement de clé où la clé utilisée pour vérifier la signature représente l'ancienne paire de clés et la clé publique certifiée représente la nouvelle paire de clés. Les certificats qui véhiculent à la fois un changement de nom de l'ACSN et un renouvellement de clé pour cette ACSN DOIVENT comprendre l'extension `NameChange` pour identifier le certificat comme tel. Cette exigence n'a pas d'incidence sur la contrainte `PathLengthConstraint` ; elle demeure « `0` ».

En outre, l'extension `NameChange` PEUT aussi être comprise dans le nouveau certificat autosigné de l'ACSN créé au moment du changement de nom distinctif (DN) de l'ACSN. Dans ce certificat racine autosigné de l'ACSN, les champs `issuer` et `subject` contiennent tous deux le nouveau DN. Contrairement au certificat de liaison autoémis de l'ACSN, qui contient à la fois l'ancien DN et le nouveau DN de l'ACSN, l'incorporation de l'extension `NameChange` dans un certificat racine autosigné de l'ACSN indique simplement qu'il s'est produit un changement de nom et ne relie pas l'ancien DN au nouveau DN.

Une ACSN NE DOIT PAS réutiliser les numéros de série de certificat. Chaque certificat émis par une ACSN, qu'elle ait ou non changé de nom, DOIT être unique.

Syntaxe ASN.1 pour l'extension de changement de nom :

```
nameChange EXTENSION ::= {
    SYNTAX NULL
    IDENTIFIED BY id-icao-mrtd-security-extensions-nameChange}

id-icao-mrtd-security-extensions OBJECT IDENTIFIER ::= {id-icao-
mrtd-security 6}
id-icao-mrtd-security-extensions-nameChange OBJECT IDENTIFIER ::=
{id-icao-
mrtd-security-extensions 1}
```

7.1.1.6 Extension de type de document

L'extension `DocumentType` DOIT être utilisée pour indiquer les types de documents, tels qu'ils figurent dans la ZLA, que le signataire de document correspondant est autorisé à produire. Cette extension DOIT toujours être mise à la valeur « non critique ».

Syntaxe ASN.1 pour l'extension de liste de type de document :

```
documentTypeList EXTENSION ::= {
    SYNTAX DocumentTypeListSyntax
    IDENTIFIED BY id-icao-mrtd-security-extensions-documentTypeList}

DocumentTypeListSyntax ::= SEQUENCE {
    version             DocumentTypeListVersion,
    docTypeList   SET OF DocumentType }

DocumentTypeListVersion ::= INTEGER {v0(0)}

-- Document Type as contained in MRZ, e.g. "P" or "ID" where a
-- single letter denotes all document types starting with that letter
DocumentType ::= PrintableString(SIZE(1..2))

id-icao-mrtd-security-extensions-documentTypeList OBJECT
IDENTIFIER ::= {id-icao-mrtd-security-extensions 2}
```

7.1.2 Profil de certificat de signataire SDL2

Les certificats de signataires SDL2 DOIVENT être conformes au profil de certificat de signataire de document défini au § 7.1.1, avec les exceptions suivantes :

Champ Subject :
Le champ « subject » des certificats de signataire SDL2 DOIT être rempli comme suit :

- `countryName` : DOIT être présent. La valeur contient le code de pays qui DOIT suivre le format des codes de pays à deux lettres, comme spécifié dans le Doc 9303-3.
- `commonName` : DOIT être présent. La valeur de cet attribut NE DOIT PAS dépasser 9 caractères.
- Les autres attributs NE DOIVENT PAS être inclus.

Extensions des certificats :
Les certificats de signataire SDL2 DOIVENT contenir les extensions de certificat définies dans le Tableau 7 ci-dessous. Toutes les autres extensions de certificat NE DOIVENT PAS être incluses.

Tableau 7. Extensions de certificat obligatoires pour SDL2

Nom de l'extension	Signataire de SDL2		Observations
	Présence	Criticité	
AuthorityKeyIdentifier	**m**	**nc**	
keyIdentifier	m		
authorityCertIssuer	o		
authorityCertSerialNumber	o		
ExtKeyUsage	**m**	**c**	Voir la note 1.

Note 1.— L'extension EKU pour chaque type de certificat de signataire SDL2 DOIT être remplie comme indiqué ci-dessous. Notez qu'un seul signataire DSL2 peut être autorisé à signer plusieurs types d'objets de données SDL2. Dans ce cas, l'extension EKU contiendrait tous les OID pertinents pour ce signataire :

```
id-icao-mrtd-security-lds2 OBJECT IDENTIFIER ::= {id-icao-mrtd-security 9}
id-icao-lds2Signer OBJECT IDENTIFIER ::= {id-icao-mrtd-
security-lds2 8}
```

- Certificats de signataire du tampon de voyage SDL2 (SDL2-TS)
  `id-icao-tsSigner OBJECT IDENTIFIER ::= { id-icao-lds2Signer 1}`

- Certificats de signataire de visa SDL2 (SDL2-V) :
  `id-icao-vSigner OBJECT IDENTIFIER ::= { id-icao-lds2Signer 2}`

- Certificats de signataire des éléments biométriques SDL2 (SDL2-B) :
  `id-icao-bSigner OBJECT IDENTIFIER ::= { id-icao-lds2Signer 3}`

Note 2.— Les certificats de signataire SDL2 doivent respecter les restrictions de taille imposées par EF.Certificates dans le Doc 9303-10.

Bien que l'extension des points de distribution de la CRL ne soit pas incluse dans ces certificats, il est obligatoire que l'état de révocation soit vérifié pour chaque certificat dans le cadre du processus de validation normal. La CRL émise par l'ACSN qui a délivré le certificat en question est la CRL utilisée pour vérifier son statut de révocation.

7.1.3 Profil de certificat de signataire de code à barres

Les certificats du signataire de code à barres DOIVENT être conformes au profil de certificat de signataire SDL2. Comme les certificats de signataire de code à barre jouent un rôle différent de celui des certificats SDL2, leur profil diffère à certains égards. En particulier, il existe des exigences spécifiques concernant le subjectDN du certificat du signataire du code à barres et le numéro de série (voir le Doc 9303-13).

Champ Subject :
Le champ « subject » des certificats de signataire de codes à barres DOIT être rempli comme suit :

- `commonName` : DOIT être présent. DOIT être constitué de deux caractères majuscules, au format printableString, qui définissent de manière unique le signataire du code à barres dans un pays, et DOIT correspondre aux lettres 3 et 4 de l'identificateur du signataire dans le code à barres, comme spécifié dans le Doc 9303-13.

- `countryName` : DOIT être constitué du code pays à deux lettres (voir le Doc 9303-3) du signataire du code à barres, en caractères majuscules, au format printableString, et DOIT correspondre aux lettres 1 et 2 de l'identificateur du signataire dans le code à barres, comme spécifié dans le Doc 9303-13.

- Les autres attributs NE DOIVENT PAS être inclus.

Extensions des certificats :
Les certificats de signataires de codes à barres DOIVENT contenir les extensions de certificat indiquées dans le Tableau 8 ci-dessous. Toutes les autres extensions de certificat NE DOIVENT PAS être incluses.

Tableau 8. Extensions autorisées pour certificats de signataire de codes à barres

Nom de l'extension	Signataire de SDL2		Observations
	Présence	**Criticité**	
AuthorityKeyIdentifier	**m**	**nc**	
keyIdentifier	m		
authorityCertIssuer	o		
authorityCertSerialNumber	o		
DocumentType	**o**		Cette extension indique le type de document que le signataire du code à barres est autorisé à produire
ExtKeyUsage	**m**	**c**	Voir note ci-dessous.

Note.— L'extension EKU pour chaque type de certificat de signataire de code à barres DOIT être remplie comme indiqué ci-dessous.

```
id-icao-mrtd-security-vds OBJECT IDENTIFIER ::= {id-icao-mrtd-security 11}
id-icao-vdsSigner OBJECT IDENTIFIER ::= {id-icao-mrtd-security-vds 1}
```

7.1.4 Profil de la CRL

Le Tableau 9 définit les exigences de profil de la CRL pour les champs du corps de la CRL. Le Tableau 10 définit les exigences de profil de la CRL pour les extensions CRL et CRL Entry.

Tableau 9. Profil des champs CRL

Composant de la liste de certificats	*CRL de l'ACSN*	*Observations*
`CertificateList`	m	
`tBSCertList`	m	Voir le Tableau 10.
`signatureAlgorithm`	m	La valeur insérée dépend de l'algorithme choisi.
`signatureValue`	m	La valeur insérée dépend de l'algorithme choisi.
`tBSCertList`		
`Version`	m	DOIT être `v2`.
`Signature`	m	La valeur insérée DOIT être la même que celle du composant `signatureAlgorithm` de la séquence `CertificateList`.
`Issuer`	m	S'ils sont présents, `countryName` et `serialNumber` DOIVENT être `PrintableString`. Les autres attributs qui ont la syntaxe `DirectoryString` DOIVENT être soit `PrintableString`, soit `UTF8String`. `countryName` DOIT être en haut de casse.
`thisUpdate`	m	DOIT se terminer par Zulu (Z). L'élément secondes DOIT être présent. Les dates jusqu'en 2049 DOIVENT être en `UTCTime`. `UTCTime` DOIT être représenté sous la forme AAMMJJHHMMSSZ. Les dates à partir de 2050 DOIVENT être en `GeneralizedTime`. `GeneralizedTime` NE DOIT PAS avoir de fractions de seconde. `GeneralizedTime` DOIT être représenté sous la forme AAAAMMJJHHMMSSZ.
`nextUpdate`	m	DOIT se terminer par Zulu (Z). L'élément secondes DOIT être présent. Les dates jusqu'en 2049 DOIVENT être en `UTCTime`. `UTCTime` DOIT être représenté sous la forme AAMMJJHHMMSSZ. Les dates à partir de 2050 DOIVENT être en `GeneralizedTime`. `GeneralizedTime` NE DOIT PAS avoir de fractions de seconde. `GeneralizedTime` DOIT être représenté sous la forme AAAAMMJJHHMMSSZ.

Composant de la liste de certificats	*CRL de l'ACSN*	*Observations*
`revokedCertificates`	c	DOIT être présent s'il existe des certificats révoqués. S'il n'y a pas de certificats révoqués, il NE DOIT PAS être présent. S'il est présent, ce composant NE DOIT PAS être vide.
`crlExtensions`	m	Voir le Tableau 10 pour les extensions qui devraient être présentes. Les valeurs par défaut des extensions NE DOIVENT PAS être codées.

Tableau 10. Profil des extensions de la CRL et CRL Entry

Nom de l'extension	*CRL de l'ACSN*	*Criticité*	*Observations*
Extensions de la CRL			
`authorityKeyIdentifier`	m	**nc**	DOIT être la même valeur que le champ `subjectKeyIdentifier` dans le certificat d'émetteur de la CRL.
`keyIdentifier`	m		
`authorityCertIssuer`	o		
`authorityCertSerialNumber`	o		
`issuerAlternativeName`	o	**nc**	Voir la note 1.
`cRLNumber`	m	**nc**	DOIT être un entier non négatif et avoir un maximum de 20 octets. DOIT utiliser un codage en complément à 2 et être représenté par le plus petit nombre d'octets possible.
`deltaCRLIndicator`	x		
`issuingDistributionPoint`	x		
`freshestCRL`	x		
Extensions CRL Entry			
`reasonCode`	x		
`holdInstructionCode`	x		
`invalidityDate`	x		
`certificateIssuer`	x		
`other private extensions`	o	**nc**	

Note 1.— Si une ACSN a changé de nom, cette extension PEUT être incluse dans les CRL émises après le changement de nom de l'ACSN. Si elles sont présentes, la valeur ou les valeurs de cette extension DOIVENT être identiques à celles du champ `issuer` *des certificats émis par l'ACSN sous son nom précédent. Après l'expiration de tous les certificats émis sous l'ancien nom de l'ACSN, ce nom de l'ACSN peut être exclu de toutes les autres CRL. Les systèmes d'inspection ne sont pas tenus de traiter cette extension. Vu que le Doc 9303 de l'OACI prescrit une seule ACSN par pays, le composant* `countryName` *du champ émetteur est suffisant pour identifier l'ACSN sans ambiguïté. La clé publique la plus récente de cette ACSN est utilisée pour vérifier la signature de la CRL. Vu que l'ACSN émet une seule CRL, celle-ci porte sur tous les certificats émis avec ce* `countryName`*. En plus de cette vérification obligatoire, il PEUT aussi y avoir une vérification optionnelle pour établir si le champ* `issuer` *du certificat est égal au champ* `issuer` *de la CRL ou à une des valeurs de l'extension* `issuerAltName` *de la CRL.*

Note 2.— Il est possible que la CRL contienne d'autres informations sur la révocation, par exemple sur l'opérateur du système ou les certificats d'autorités d'enregistrement.

7.2 ICP d'autorisation

L'ICP d'autorisation comprend des certificats X.509 pour le SPOC et des certificats vérifiables par carte pour le CVCA, le DV et les terminaux. La présente section spécifie les profils pour les certificats SPOC, CVCA, DV et IS. Une vue d'ensemble des objets de données contenus dans les certificats vérifiables par carte est fournie, et le codage de ces objets est également couvert.

7.2.1 Profil de certificat SPOC

Une configuration AC distincte peut être utilisée pour émettre directement des certificats SPOC avec les restrictions suivantes au profil de certificat AC auto-signé :

- Le certificat d'AC DOIT être conforme à la norme RFC 5280 ;
- Les algorithmes de hachage SHA-224, SHA-256, SHA-384 et SHA-512 sont les seuls permis ;
- `countryName` DOIT être présent dans le champ Subject.

Les certificats SPOC SDL2 (client et serveur) DOIVENT être conformes au profil de certificat de communication défini au § 7.1, avec les restrictions suivantes.

Champ Issuer :
Les certificats SPOC sont émis soit par l'ACSN, soit par une AC distincte mise en place spécifiquement pour émettre des certificats SPOC.

Champ Subject :
Pour les certificats SPOC SDL2, le champ sujet DOIT être renseigné comme suit :

- `countryName` : DOIT être présent. La valeur contient le code de pays qui DOIT suivre le format des codes de pays à deux lettres, comme spécifié dans le Doc 9303-3.
- `commonName` : DOIT être présent. Pour les certificats SPOC TLS client, la valeur DEVRAIT être « SPOC TLS client ». Pour les certificats de serveur SPOC TLS, la valeur DEVRAIT être « SPOC TLS server ».
- D'autres attributs PEUVENT aussi être inclus à la discrétion de l'État émetteur ou de l'organisation émettrice.

Extensions de l'utilisation des clés
Pour les certificats SPOC, la ou les valeurs dépendent de la suite de chiffrement utilisée.

Extensions de variantes de noms de sujets
En plus des valeurs indiquées dans le profil de certificat de communication, les certificats de serveur TLS SPOC DOIVENT également contenir une valeur dNSName qui est la partie hôte de l'URL SPOC.

Extensions de l'utilisation des clés étendues
Pour les certificats de client et de serveur SPOC, la valeur pertinente énumérée ci-dessous DOIT être incluse.

- Certificats de clients SPOC : L'OID est 2.23.136.1.1.10.1 ;
- Certificats de serveurs SPOC : L'OID est 2.23.136.1.1.10.2.

Extensions des points de distribution des CRL
Cette extension est obligatoire dans les certificats client et serveur SPOC.

7.2.2 Profils de certificat CVCA, DV et terminal

Les certificats de liaison CVCA, les certificats DV et les certificats de terminal doivent être validés par les CI. En raison des restrictions de calcul de ces puces, les certificats DOIVENT être dans un format vérifiable par carte (certificats CV).

Il FAUT utiliser le format et le profil de certificat spécifiés dans le Tableau 11. Le Doc 9303-11 donne des renseignements supplémentaires sur les valeurs de codage.

Tableau 11. Profil de certificat CV

Objet de données	Certificat Présence
Certificat CV	m
Corps du certificat	m
Identificateur de profil de certificat	m
Référence de l'autorité de certification	m
Clé publique	m
Référence du titulaire du certificat	m
Modèle d'autorisation du titulaire du certificat	m
Date de prise d'effet du certificat	m
Date d'expiration du certificat	m
Extensions des certificats	o
Signature	m

7.2.2.1 Identificateur du profil de certificat

La version du profil est indiquée par l'identificateur du profil de certificat. La version 1 DOIT être utilisée et est désignée par une valeur de 0.

7.2.2.2 Référence de l'autorité de certification et référence du titulaire de certificat

Chaque certificat CV DOIT contenir deux références de clés publiques (une référence du titulaire de certificat et une référence d'autorité de certification).

La référence de l'autorité de certification est une référence à la clé publique (externe) de l'autorité de certification (CVCA ou DV) qui DOIT être utilisée pour vérifier la signature du certificat.

La référence du titulaire du certificat est un identificateur pour la clé publique fournie dans le certificat qui DOIT être utilisé pour référencer cette clé publique.

Note.— En conséquence, la référence de l'autorité du certificat contenue dans un certificat DOIT être égale à la référence du titulaire du certificat dans le certificat correspondant de l'autorité de certification émettrice.

La référence du titulaire du certificat DOIT être constituée des éléments concaténés suivants : code pays, mnémonique du titulaire et numéro de séquence. Ces éléments DOIVENT être choisis conformément au Tableau 12 et aux règles suivantes :

a) Code pays :

- Le code pays DOIT être le code à deux lettres Doc 9303-3 du pays du titulaire du certificat.

b) Mnémonique du titulaire :

- La mnémonique du titulaire DOIT être attribuée comme identificateur unique, comme suit :
  - La mnémonique du titulaire d'une CVCA DOIT être attribuée par la CVCA elle-même ;
  - La mnémonique du titulaire d'un DV DOIT être attribuée par sa CVCA nationale ;
  - La mnémonique du titulaire d'un IS DOIT être attribuée par le DV superviseur.

c) Numéro de séquence :

- Le numéro de séquence DOIT être attribué par le titulaire du certificat ;
- Le numéro de séquence DOIT être numérique ou alphanumérique ;
  - Un numéro de séquence numérique DOIT être composé des caractères « 0...9 ».
  - Un numéro de séquence alphanumérique DOIT être composé des caractères « 0...9 » et « A...Z ».
- Le numéro de séquence DOIT commencer par le code pays à deux lettres de l'autorité de certification (Doc 9303-3), les trois caractères restants DOIVENT être attribués comme numéro de séquence alphanumérique ;
- Le numéro de séquence PEUT être réinitialisé si tous les numéros de séquence disponibles sont épuisés.

Tableau 12. Référence du titulaire du certificat

	Codage	Longueur
Code pays	Doc 9303-3.	2F
Mnémonique du titulaire	ISO/IEC 8859-1	9V
Numéro de séquence	ISO/IEC 8859-1	5F

7.2.2.3 Clé publique

Ce champ contient la clé publique à certifier.

Les certificats auto-signés de la CVCA DOIVENT contenir des paramètres de domaine. Les certificats de liaison CVCA PEUVENT contenir des paramètres de domaine, sauf dans le cas où les paramètres de domaine ont changé. Dans ce cas, les certificats de liaison DOIVENT contenir les nouveaux paramètres de domaine.

Les certificats DV et de terminal NE DOIVENT PAS contenir de paramètres de domaine. Les paramètres de domaine des clés publiques DV et de terminal DOIVENT être hérités de la clé publique CVCA correspondante.

7.2.2.4 Modèle d'autorisation du titulaire du certificat

Le rôle et l'autorisation du titulaire du certificat DOIVENT être codés dans le modèle d'autorisation du titulaire du certificat. Ce modèle est une séquence qui se compose des objets de données suivants :

a) un identificateur d'objet spécifiant le type de terminal et le format du modèle ;

b) un objet de données discrétionnaire qui code l'autorisation relative, c'est-à-dire le rôle et l'autorisation du titulaire du certificat par rapport à l'autorité de certification.

Les valeurs spécifiques sont définies dans le Doc 9303-10.

7.2.2.5 Date de prise d'effet et date d'expiration du certificat

La combinaison de ces deux dates indique la période de validité du certificat. La date de prise d'effet du certificat DOIT être la date de génération du certificat. La date d'expiration du certificat est la date après laquelle le certificat expire.

7.2.2.6 Extensions des certificats (extensions des autorisations)

Les extensions d'autorisation PEUVENT être incluses dans les certificats CVCA, DV et de terminal. Ces extensions transmettent des autorisations supplémentaires à celles du gabarit d'autorisation du titulaire de certificat dans le certificat.

Une extension d'autorisation est une séquence de gabarits de données discrétionnaires, où chaque gabarit de données discrétionnaire DOIT contenir une séquence des objets de données suivants, également indiqués dans le Tableau 13 :

a) un identificateur d'objet qui spécifie le contenu et le format de l'extension ;

b) un objet de données spécifique au contexte qui contient l'autorisation codée.

Tableau 13. Extensions des certificats

Objet de données
Extensions des certificats
Modèle de données discrétionnaire
Identificateur d'objet
Objet de données spécifique au contexte
Modèle de données discrétionnaire
Identificateur d'objet
Objet de données spécifique au contexte
…

Note.— La procédure de validation des certificats décrite dans le Doc 9303-11 ne prend pas en compte les extensions des certificats. Ainsi, les extensions sont des attributs non critiques et le CI NE DOIT PAS rejeter les certificats en raison d'extensions inconnues.

7.2.2.7 Signature

La signature du certificat DOIT être créée sur le corps codé du certificat (c.-à-d. incluant l'étiquette et la longueur). La référence de l'autorité de certification DOIT identifier la clé publique à utiliser pour vérifier la signature.

7.2.3 Objets de données

Le Tableau 14 donne un aperçu des étiquettes, des longueurs et des valeurs des objets de données utilisés dans les certificats CVCA, DV et de terminal.

7.2.3.1 Codage des valeurs

Les types de valeurs de base utilisés dans cette spécification sont les suivants : entiers (non signés), points de courbe elliptique, dates, chaînes de caractères, chaînes d'octets, identificateurs d'objets et séquences.

7.2.3.1.1 Entiers non signés

Tous les entiers utilisés dans cette spécification sont des entiers non signés. 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 minimum d'octets DOIT être utilisé, c'est-à-dire que les premiers octets de la valeur 0x00 NE DOIVENT PAS être utilisés.

Note.— En revanche, le type ASN.1 INTEGER est toujours un nombre entier signé.

7.2.3.1.2 Points de courbes elliptiques

La conversion des points de courbe elliptique en chaînes d'octets est spécifiée dans TR-03111. Le format non comprimé DOIT être utilisé.

Tableau 14. Aperçu des objets de données (triés par étiquette)

Nom	Étiquette	Longueur	Valeur	Observation
Identificateur d'objet	0x06	V	Identificateur d'objet	–
Référence de l'autorité de certification	0x42	16V	Chaîne de caractères	Identifie la clé publique de l'autorité de certification émettrice dans un certificat.
Données discrétionnaires	0x53	V	Chaîne d'octets	Contient des données arbitraires.
Référence du titulaire du certificat	0x5F20	16V	Chaîne de caractères	Associe la clé publique contenue dans un certificat à un identificateur.
Date d'expiration du certificat	0x5F24	6F	Date	La date après laquelle le certificat expire.
Date de prise d'effet du certificat	0x5F25	6F	Date	La date de génération du certificat.
Identificateur de profil de certificat	0x5F29	1F	Entier non signé	Version du certificat et format de la demande de certificat.
Signature	0x5F37	V	Chaîne d'octets	Signature numérique produite par un algorithme cryptographique asymétrique.
Extensions des certificats	0x65	V	Séquence	Emboîte les extensions de certificats.
Authentification	0x67	V	Séquence	Contient des objets de données liés à l'authentification.
Modèle de données discrétionnaire	0x73	V	Séquence	Emboîte des objets de données arbitraires.
Certificat CV	0x7F21	V	Séquence	Emboîte le corps du certificat et la signature.
Clé publique	0x7F49	V	Séquence	Emboîte la valeur de la clé publique et les paramètres de domaine.
Modèle d'autorisation du titulaire du certificat	0x7F4C	V	Séquence	Code le rôle du titulaire du certificat (c.-à-d. CVCA, DV, Terminal) et attribue des droits d'accès en lecture/écriture.
Corps du certificat	0x7F4E	V	Séquence	Emboîte les objets de données du corps du certificat.

F : longueur fixe (nombre exact d'octets), V : longueur variable (jusqu'à un nombre d'octets).

7.2.3.1.3 Dates

Une date est codée en 6 chiffres « d1...d6 » dans le format AAMMJJ en utilisant le fuseau horaire GMT. Il est converti en une chaîne d'octets « o1...o6 » en codant chaque chiffre dj sur un octet oj comme des BCD non regroupés ($1 \leq j \leq 6$).

L'année AA est codée en deux chiffres et doit être interprétée comme 20AA, c'est-à-dire que l'année est comprise entre 2000 et 2099.

7.2.3.1.4 Chaînes de caractères

Une chaîne de caractères « c1…cn » est une concaténation de n caractères cj avec $1 \leq j \leq n$. Elle DOIT être convertie en une chaîne d'octets « o1...on » en convertissant chaque caractère cj en un octet oj en utilisant le jeu de caractères ISO/IEC 8859-1.

Les codes de caractères 0x00-0x1F et 0x7F-0x9F ne sont pas attribués et NE DOIVENT PAS être utilisés. La conversion d'un octet en un caractère non attribué DOIT entraîner une erreur.

7.2.3.1.5 Chaînes d'octets

Une chaîne d'octets « o1...on » est une concaténation de n octets oj avec $1 \leq j \leq n$. Chaque octet oj est composé de 8 bits.

7.2.3.1.6 Identificateurs d'objets

Un identificateur d'objet « i1.i2...in » est codé comme une liste ordonnée de n entiers non signés ij avec $1 \leq j \leq n$. Elle DOIT être convertie en une chaîne d'octets « o1...on−1 » au moyen de la procédure suivante :

1) Les deux premiers entiers i1 et i2 sont regroupés en un seul entier i qui est ensuite converti en chaîne d'octets o1. La valeur i est calculée comme suit :

$$i=i1 \cdot 40+i2$$

2) Les autres entiers ij sont directement convertis en chaînes d'octets oj-1 avec $3 \leq j \leq n$.
[X.690] contient plus de renseignements sur le codage.

Note.— Les entiers non signés sont codés sous forme de chaînes d'octets en utilisant le format gros-boutiste décrit dans le Doc 9303-11, mais seuls les bits 1 à 7 de chaque octet sont utilisés. Le bit 8 (le bit le plus à gauche) mis à un est utilisé pour indiquer que cet octet n'est pas le dernier octet de la chaîne.

7.2.3.1.7 Séquences

Une séquence « D1...Dn » est une liste ordonnée de n objets de données Dj avec $1 \leq j \leq n$. La séquence DOIT être convertie en une liste concaténée de chaînes d'octets « O1...On » par codage DER de chaque objet de données Dj en une chaîne d'octets Oj.

7.2.3.2 Codage des objets de données de clé publique

Un objet de données de clé publique contient une séquence d'un identificateur d'objet et de plusieurs objets de données propres au contexte :

- l'identificateur d'objet est propre à l'application et se rapporte non seulement au format de la clé publique (c.-à-d. 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.

7.2.3.2.1 Clés publiques RSA

Les objets de données contenus dans une clé publique RSA sont indiqués dans le Tableau 15. L'ordre des objets de données est fixe.

7.2.3.2.2 Clés publiques à courbe elliptique

Les objets de données contenus dans une clé publique EC sont indiqués dans le Tableau 16. 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 :

- Les certificats CVCA auto-signés DOIVENT contenir des paramètres de domaine ;
- Les certificats de liaison CVCA PEUVENT contenir des paramètres de domaine ;
- Les certificats DV et de terminal NE DOIVENT PAS contenir de paramètres de domaine. Les paramètres de domaine des clés publiques DV et de terminal DOIVENT être hérités de la clé publique CVCA correspondante ;
- Les demandes de certificat DOIVENT toujours contenir des paramètres de domaine.

Tableau 15. Clé publique RSA

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

Tableau 16. Clé publique EC

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

8. PROTOCOLE SPOC

Le point unique de contact (SPOC) est la seule interface exposée par un État pour les opérations de gestion de clés avec des États étrangers pour l'ICP d'autorisation SDL2. Le protocole SPOC est le protocole de gestion des clés pour les opérations entre les CVCA et les DV dans différents États. Bien que le protocole SPOC puisse également être utilisé pour les communications nationales entre une CVCA et ses DV nationaux et entre un DV et l'ensemble des terminaux nationaux qu'il gère, cela n'est pas obligatoire. D'autres protocoles de gestion des clés peuvent être utilisés pour la gestion des clés domestiques.

Le protocole SPOC est utilisé pour échanger des clés et des certificats, afin :

- qu'un DV puisse envoyer une demande de certification à la CVCA étrangère ;
- qu'une CVCA puisse envoyer le certificat émis au DV qui le demande ;
- que les CVCA et les DV puissent demander l'ensemble des certificats valides à une CVCA étrangère ;
- que des messages généraux puissent être échangés entre les DV et les CVCA.

Au sein d'un État :

- La CVCA DOIT utiliser son SPOC national pour accepter les demandes de certification étrangères entrantes et pour envoyer les certificats ou les notifications d'échec au demandeur ;
- Les DV DOIVENT utiliser leur SPOC national pour envoyer des demandes de certification à des CVCA étrangers et pour recevoir les certificats ou les notifications d'échec qui en résultent ;
- Le SPOC DOIT collecter les demandes et les réponses de la CVCA nationale et des DV et les transmettre au SPOC de l'État destinataire ;

- Le SPOC DOIT recueillir les demandes et les réponses des SPOC des autres États et les transmettre à la CVCA/DV nationale concernée.

La communication du service web SPOC DOIT utiliser HTTPS avec authentification TLS du client et du serveur.

Note.— Les SPOC sont des nœuds de communication entre les entités de l'ICP d'autorisation. Ils devraient donc être disponibles 24 heures sur 24 et 7 jours sur 7 et être accessibles par les SPOC étrangers.

Chaque SPOC s'inscrit séparément auprès de tous les autres SPOC concernés, en fournissant au moins les informations suivantes :

- État SPOC – État pour lequel le SPOC fournit l'interface de communication ;
- URL du SPOC – URL du WSDL décrivant l'interface du SPOC et l'emplacement du service ;
- Certificat AC SPOC – certificat(s) utilisé(s) pour vérifier les certificats de communication SPOC.

8.1 Structures liées au SPOC

Les structures suivantes sont définies pour être utilisées dans les messages SPOC.

8.1.1 Structure de la demande de certificat

Les demandes de certificats sont des certificats réduits vérifiables par carte qui peuvent porter une signature supplémentaire. Le profil de demande de certificat spécifié dans le Tableau 17 DOIT être utilisé.

8.1.1.1 Identificateur de profil de certificat

La version est 1, identifiée par une valeur de 0.

Tableau 17. Profil de demande de certificat CV

Objet de données	Présence du certificat
Authentification	c
Certificat CV	m
Corps du certificat	m
Identificateur de profil de certificat	m
Référence de l'autorité de certification	r
Clé publique	m
Référence du titulaire du certificat	m
Signature	m
Référence de l'autorité de certification	c
Signature	c

8.1.1.2 Référence de l'autorité de certification

La référence de l'autorité de certification DEVRAIT être utilisée pour informer l'autorité de certification sur la clé privée dont le demandeur s'attend à ce qu'elle soit utilisée pour signer le certificat. Si la référence de l'autorité de certification contenue dans la demande diffère de la référence de l'autorité de certification contenue dans le certificat émis (c.-à-d. que le certificat émis est signé par une clé privée non prévue par le demandeur), le certificat correspondant de l'autorité de certification DEVRAIT également être fourni au demandeur en réponse.

8.1.1.3 Clé publique

Les demandes de certificat DOIVENT toujours contenir des paramètres de domaine.

8.1.1.4 Référence du titulaire du certificat

La référence du titulaire du certificat est utilisée pour identifier la clé publique contenue dans la demande et le certificat qui en résulte.

8.1.1.5 Signature(s)

Une demande de certificat peut comporter jusqu'à deux signatures ; une signature intérieure et une signature extérieure :

Signature intérieure (EXIGÉE)

Le corps du certificat est auto-signé, c'est-à-dire que la signature interne DOIT être vérifiable avec la clé publique contenue dans la demande de certificat. La signature DOIT être créée sur le corps du certificat codé (y compris l'étiquette et la longueur).

Signature extérieure (CONDITIONNEL)

- La signature est OPTIONNELLE si une entité demande le certificat initial. Dans ce cas, la demande PEUT être également signée par une autre entité à laquelle l'autorité de certification réceptrice fait confiance (par exemple, la CVCA nationale peut authentifier la demande d'un DV envoyée à une CVCA étrangère).

- La signature est EXIGÉE si une entité demande un certificat successif. Dans ce cas, la demande DOIT être également signée par le demandeur à l'aide d'une paire de clés récente préalablement enregistrée auprès de l'autorité de certification réceptrice.

Si la signature externe est utilisée, un objet de données d'authentification DOIT être utilisé pour intégrer le certificat CV (demande), la référence de l'autorité de certification et la signature supplémentaire. La référence de l'autorité de certification DOIT identifier la clé publique à utiliser pour vérifier la signature supplémentaire. La signature DOIT être créée sur la concaténation du certificat CV codé et de la référence de l'autorité de certification codée (c.-à-d. les deux incluant l'étiquette et la longueur).

8.2 Protocole de messages SPOC

La présente section détaille les messages utilisés dans le protocole SPOC.

8.2.1 Message de demande de certificat

Utilisation prévue :

Le message RequestCertificate est utilisé par un SPOC pour demander à une CVCA étrangère la génération d'un nouveau certificat pour l'un de ses DV.

Paramètres d'entrée :

callerID : (Obligatoire)
Ce paramètre contient l'identifiant de l'État d'origine de la demande. La valeur DOIT être le code pays à deux lettres conformément au Doc 9303-3. La valeur de callerID DOIT être vérifiée par le SPOC récepteur avec la valeur enregistrée par le SPOC d'origine lors de son enregistrement.

messageID : (Obligatoire)
Ce paramètre contient l'identificateur du message. Il DOIT identifier le message de manière unique parmi tous les messages de cet expéditeur. Si un message de réponse doit être envoyé à l'expéditeur à la suite de ce message, le message de réponse contiendra le même messageID. Ainsi, un message de réponse entrant peut être attribué au message original correct. La construction et l'attribution de l'identificateur du message peuvent être décidées par l'expéditeur et ne sont pas vérifiées par la partie destinataire.

certReq : (Obligatoire)
Ce paramètre contient la demande de certificat proprement dite. Il DOIT être construit conformément au § 8.1.1. Le codage DOIT suivre les spécifications du § 7.2.3.1.

Paramètres de sortie :

CertificateSeq : (Conditionnel)
Ce paramètre contiendra le résultat (un ou plusieurs certificats) après le traitement de ce message, si le message a été traité avec succès et de manière synchrone par le récepteur. Il est EXIGÉ si des certificats doivent être envoyés avec la réponse. Il DOIT être absent si aucun certificat ne sera envoyé avec le message.

Codes de retour :

- ok_cert_available : Le message a été traité avec succès et de manière synchrone. Le paramètre de sortie certificateSeq contient un ou plusieurs certificats.
- ok_reception_ack : La réception du message fait l'objet d'un accusé de réception. Aucune autre vérification du message n'a encore été effectuée. Le traitement du message se fera de manière asynchrone. Le résultat du traitement sera envoyé à l'URL enregistrée à l'aide du message SendCertificates.
- failure_inner_signature : La vérification de la signature interne de la demande de certificat proprement dite a échoué.
- failure_outer_signature : La vérification de la signature extérieure de la demande de certificat proprement dite a échoué.

- failure_syntax : Le message n'est pas syntaxiquement correct.
- failure_request_not_accepted : Le message a été traité correctement mais la demande n'a pas été acceptée.
- failure_request_syntax : La demande de certificat n'est pas correcte (p. ex., la syntaxe ou le format du fichier)
- failure_expired : Le certificat à utiliser pour vérifier la signature externe de la demande a expiré.
- failure_domain_parameters : Les paramètres de domaine contenus dans la demande ne correspondent pas aux paramètres de domaine du certificat CVCA destiné à signer le certificat DV demandé.
- failure_internal_error : Erreur autre que celles mentionnées ci-dessus.

Observations :

Le corps de la demande de certificat DEVRAIT contenir une référence de l'autorité de certification afin d'informer la CVCA de la clé privée qui, selon le demandeur, sera utilisée pour signer le certificat. Si la référence de l'autorité de certification de la demande diffère de celle du certificat délivré, le certificat correspondant de la CVCA DOIT également être fourni dans la réponse. Dans ce cas, et si le message est traité de manière synchrone, le certificat CVCA DOIT faire partie du paramètre de sortie certificateSeq. Le certificat DV DOIT être le premier certificat de la séquence. Les certificats CVCA (racine et/ou lien) DOIVENT être classés par date d'entrée en vigueur (ascendante) dans la séquence.

8.2.2 Message d'envoi de certificats

Utilisation prévue:

Le message SendCertificates est utilisé par un SPOC pour envoyer le nouveau certificat ou la nouvelle chaîne de certificats au SPOC demandeur. Ce message DOIT être généré en réponse à :

- RequestCertificate : lors du traitement réussi de la demande asynchrone après l'émission du certificat ;
- GetCACertificates.

En outre, le message DOIT être utilisé lorsqu'un nouveau certificat est créé (racine et lien CVCA) pour pousser les certificats vers les SPOC étrangers enregistrés.

Paramètres d'entrée :

callerID : (Obligatoire)

Ce paramètre contient l'identifiant de l'État d'origine. La valeur DOIT être le code pays à deux lettres conformément au Doc 9303-3. La valeur de callerID DOIT être vérifiée par le SPOC récepteur avec la valeur enregistrée par le SPOC d'origine lors de son enregistrement.

messageID : (Conditionnel)
Lorsque le message est généré en réponse à un message de demande, le paramètre DOIT contenir la même valeur que le paramètre messageID du message de demande. Lorsque la génération du message a été déclenchée sans intervention externe (génération d'une nouvelle clé du certificat CVCA), la valeur de statusInfo DOIT être new_cert_available_notification et le paramètre messageID PEUT être omis et DOIT être ignoré s'il est présent.

statusInfo : (Obligatoire)
Ce paramètre contient un code d'état concernant le résultat du traitement du message correspondant. Les états suivants sont possibles :

- new_cert_available_notification : Le SPOC d'origine veut notifier que le ou les nouveaux certificats CVCA sont disponibles sans être demandés.
- ok_cert_available : La demande a été traitée avec succès. Le paramètre d'entrée certificateSeq contient un ou plusieurs certificats.
- failure_inner_signature : La vérification de la signature interne de la demande de certificat proprement dite a échoué.
- failure_outer_signature : La vérification de la signature extérieure de la demande de certificat proprement dite a échoué.
- failure_syntax : Le message correspondant n'est pas syntaxiquement correct.
- failure_request_not_accepted : Le message correspondant a été traité correctement mais la demande n'a pas été acceptée.
- failure_certificate : Un ou plusieurs des certificats envoyés ne sont pas corrects (syntaxe ou signature).
- failure_internal_error : erreur autre que le certificateSeq ci-dessus (conditionnel).

Ce paramètre est EXIGÉ si des certificats doivent être envoyés avec le message. Il DOIT être absent si aucun certificat ne sera envoyé avec le message. Les certificats DOIVENT être des TLV binaires codés DER, comme défini au § 7.2.3.

Lorsque le message est généré en réponse à un message GetCACertificates, ou parce qu'il y a un nouveau certificat, la séquence DOIT contenir une liste de certificats d'AC. La liste DOIT être ordonnée. Les certificats CVCA (lien et/ou racine) DOIVENT être classés par date de prise d'effet dans l'ordre. Lorsque la séquence contient des certificats avec différents paramètres de domaine, au moins un certificat avec des paramètres de domaine inclus pour chaque variante de paramètres de domaine DOIT être présent. Tous les certificats AC actuels DOIVENT être inclus.

Lorsque le message est généré en réponse au message RequestCertificate, le contenu de la séquence est le même que celui décrit pour la réponse synchrone de RequestCertificate.

Paramètres de sortie :

Aucun

Codes de retour :

- ok_received_correctly : Le message a été traité avec succès.
- failure_syntax : Le message n'est pas syntaxiquement correct.
- failure_messageID_unknown : Le messageID contenu ne peut pas correspondre à un message précédemment envoyé.
- failure_internal_error : Erreur autre que celles mentionnées ci-dessus

8.2.3 Message « Get CA Certificates » (Obtenir des certificats d'autorité de certification)

Utilisation prévue :

Ce message est envoyé par un SPOC à un SPOC étranger afin d'obtenir tous les certificats CVCA valides (certificats de liaison et certificats auto-signés) de cet État.

Paramètres d'entrée :

callerID : (Obligatoire)
Ce paramètre contient l'identifiant de l'État d'origine. La valeur DOIT être le code pays à deux lettres conformément au Doc 9303-3. La valeur de callerID DOIT être vérifiée par le SPOC récepteur avec la valeur enregistrée par le SPOC d'origine lors de son enregistrement.

messageID : (Obligatoire)
Ce paramètre contient l'identificateur du message. Il DOIT identifier le message de manière unique parmi tous les messages de l'expéditeur. Si un message de réponse doit être envoyé à l'expéditeur à la suite de ce message, le message de réponse contiendra le même messageID. Ainsi, un message de réponse entrant peut être attribué au message original correct. La construction et l'attribution de l'identificateur du message peuvent être décidées par l'expéditeur.

Paramètres de sortie :

certificateSeq : (Conditionnel)
Ce paramètre contiendra le résultat (un ou plusieurs certificats) après le traitement de ce message, si le message a été traité avec succès et de manière synchrone par le récepteur. Il est EXIGÉ si des certificats doivent être envoyés avec la réponse. Il DOIT être absent si aucun certificat ne sera envoyé avec le message.

Codes de retour :

- ok_cert_available : Le message a été traité avec succès et de manière synchrone. Le paramètre de sortie certificateSeq contient un ou plusieurs certificats d'AC.
- ok_reception_ack : La réception du message fait l'objet d'un accusé de réception. Aucune autre vérification du message n'a encore été effectuée. Le traitement du message se fera de manière asynchrone. Le résultat du traitement sera envoyé à l'URL enregistrée à l'aide du message SendCertificates.
- failure_syntax : Le message n'est pas syntaxiquement correct.
- failure_internal_error : Erreur autre que celles mentionnées ci-dessus.

Observations :

Si le message est traité avec succès et accepté, la CVCA DOIT envoyer tous les certificats CVCA valides dans la réponse, soit dans le paramètre de sortie certificateSeq (traitement synchrone), soit dans le message de réponse correspondant SendCertificates (traitement asynchrone).

8.2.4 Messages généraux

Utilisation prévue :

Ce message est envoyé par un SPOC à un SPOC étranger afin d'envoyer une notification ou un autre message général en texte lisible par l'homme.

Paramètres d'entrée :

callerID : (Obligatoire)
Ce paramètre contient l'identifiant de l'État d'origine. La valeur DOIT être le code pays à deux lettres conformément au Doc 9303-3. La valeur de callerID DOIT être vérifiée par le SPOC récepteur avec la valeur enregistrée du SPOC d'origine lors de son enregistrement, y compris les caractéristiques de sécurité du message (le certificat de signature numérique/le certificat client TLS est enregistré pour l'État concerné).

messageID : (Obligatoire)
Ce paramètre contient l'identificateur du message. Il DOIT identifier le message de manière unique parmi tous les messages de l'expéditeur. Si un message de réponse doit être envoyé à l'expéditeur à la suite de ce message, le message de réponse contiendra le même messageID. Ainsi, un message de réponse entrant peut être attribué au message original correct. La construction et l'attribution de l'identificateur du message peuvent être décidées par l'expéditeur.

subject : (Obligatoire)
Ce paramètre contient l'objet du message. L'objet DEVRAIT décrire brièvement le contenu du corps du message. L'anglais DOIT être utilisé pour l'objet.

body : (Obligatoire)
Ce paramètre contient le corps du message. Le corps du message DOIT être un texte en clair lisible par l'homme et non destiné à un traitement automatisé direct. L'anglais DOIT être utilisé pour le corps du texte.

Codes de retour :

- ok : Le message a été accepté pour livraison.
- failure_syntax : Le message n'est pas syntaxiquement correct.
- failure_internal_error : Erreur autre que celles mentionnées ci-dessus.

8.3 Service web

L'interface de service web est l'interface pour l'échange de données filaires inter-SPOC de routine. L'interface DOIT utiliser le protocole [SOAP] sur [HTTPS]. L'interface du service web du SPOC DOIT être conforme au WSDL spécifié au § 8.3.3.

8.3.1 Utilisation de SOAP

Le [SOAP] pur sur [HTTPS] DOIT être utilisé pour mettre en œuvre les interfaces du service web. Toute autre extension SOAP (p. ex. WS-Adressing, WS-Security, WS-Secure Conversation, WS-Authorization, WS-Federation, WSAuthorization, WS-Policy, WS-Trust, WS-Privacy, WS-Test et autres extensions de WS) NE DOIT PAS être utilisée.

Le type de nœud SOAP intermédiaire NE DOIT PAS être utilisé. Seule une configuration directe entre SPOC client et SPOC serveur DOIT être utilisée.

L'élément SOAP fault DOIT être utilisé uniquement lorsqu'une erreur de traitement de la couche transport non couverte par cette spécification se produit. Les erreurs au niveau de l'application DOIVENT être communiquées sous forme de réponses SOAP normales à l'aide du mécanisme d'erreur décrit pour chaque message.

Il est RECOMMANDÉ que l'interface de service web soit mise en œuvre conformément à [WS-IBP] et [WSI-SSBP].

L'interface SOAP du SPOC DOIT être conforme aux définitions WSDL décrites au § 8.3.3.

8.3.2 Considérations relatives à la sécurité

La communication du service web du SPOC DOIT utiliser un canal sécurisé et authentifié. SOAP sur HTTPS DOIT être utilisé. TLS v1.2 DOIT être utilisé.

Le client TLS DOIT effectuer les vérifications suivantes :

- le certificat du serveur DOIT être entièrement validé conformément à [RFC 5280], y compris l'état de révocation ;
- l'extension ExtKeyUsage du certificat de serveur DOIT être présente et contenir les OID conformément au § 7.2.1 Certificat de serveur SPOC TLS ;
- le pays sujet du certificat serveur DOIT être égal à la valeur du paramètre callerID. En cas d'échec, le client TLS DOIT fermer la connexion.

Le serveur TLS DOIT effectuer les vérifications suivantes :

- le client DOIT être entièrement authentifié à l'aide d'un certificat ;
- le certificat du client DOIT être entièrement validé conformément à [RFC 5280], y compris l'état de révocation ;
- l'extension ExtKeyUsage du certificat du client DOIT être présente et DOIT contenir les OID conformément au § 7.2.1 Certificat du client SPOC TLS ;
- le pays du sujet du certificat du client DOIT correspondre à celui prévu.

Si certaines des vérifications échouent, la demande DOIT être rejetée au moyen du code de réponse HTTP 401 Unauthorized.

Dans le cadre de la négociation de la prise de contact TLS, le client DOIT prendre en charge toutes les suites de chiffrement TLS définies au § 4.2.2. Le côté serveur et le côté client DOIVENT prendre en charge l'authentification basée sur RSA et ECDSA. Il est permis à un serveur de demander et au client d'envoyer un certificat client d'un type différent de celui du serveur.

L'utilisation de l'agrément de clé ECDHE_ECDSA dans la prise de contact TLS est conforme aux ajouts définis dans [TLSECC], [TLS1.2] et [TLSEXT]. Le client et le serveur DOIVENT tous deux prendre en charge les extensions de courbes elliptiques appropriées conformément à la spécification [TLSECC] dans le cadre de la prise de contact TLS. Les courbes elliptiques et les formats de points EC pris en charge sont définis dans la section 5 de [TLSECC]. L'utilisation

des suites de chiffrement TLS prises en charge, définies au § 4.2.2, qui utilisent la norme de chiffrement avancée (AES) pour le chiffrement DOIT être conforme à la spécification [TLSAES].

8.3.3 WSDL pour l'interface de service web du SPOC

L'interface SOAP du SPOC DOIT être conforme aux définitions WSDL suivantes :

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/
  " xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:SPOC="http://namespaces.icao.int/lds2"
  targetNamespace="http://namespaces.
  icao.int/lds2">

  <wsdl:types>
    <xs:schema xmlns="http://namespaces.icao.int/lds2"
      targetNamespace="http://namespaces."
      elementFormDefault="qualified" attributeFormDefault="unqualified">
      <xs:element name="certificateSequence">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="certificate" type="xs:base64Binary" minOccurs="1"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
      </xs:element>
      <xs:element name="RequestCertificateRequest">
    <xs:complexType>
      <xs:sequence>
            <xs:element name="callerID" type="xs:string"/>
            <xs:element name="messageID" type="xs:string"/>
            <xs:element name="certificateRequest" type="xs:base64Binary"/>
      </xs:sequence>
    </xs:complexType>
      </xs:element>
      <xs:element name="RequestCertificateResponse">
    <xs:complexType>
      <xs:sequence>
            <xs:element ref="certificateSequence" minOccurs="0" maxOccurs="1"/>
            <xs:element name="result">
          <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="ok_cert_available"/>
                <xs:enumeration value="ok_reception_ack"/>
                <xs:enumeration value="failure_inner_signature"/>
                <xs:enumeration value="failure_outer_signature"/>
                <xs:enumeration value="failure_syntax"/>
                <xs:enumeration value="failure_request_not_accepted"/>
                <xs:enumeration value="failure_request_syntax"/>
```

```
            <xs:enumeration value="failure_expired"/>
            <xs:enumeration value="failure_domain_parameters"/>
            <xs:enumeration value="failure_internal_error"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
  </xs:element>
  <xs:element name="SendCertificatesRequest">
<xs:complexType>
  <xs:sequence>
      <xs:element name="callerID" type="xs:string"/>
      <xs:element name="messageID" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element ref="certificateSequence" minOccurs="0" maxOccurs="1"/>
      <xs:element name="statusInfo">
     <xs:simpleType>
         <xs:restriction base="xs:string">
           <xs:enumeration value="new_cert_available_notification"/>
           <xs:enumeration value="ok_cert_available"/>
           <xs:enumeration value="failure_inner_signature"/>
           <xs:enumeration value="failure_outer_signature"/>
           <xs:enumeration value="failure_syntax"/>
           <xs:enumeration value="failure_request_not_accepted"/>
           <xs:enumeration value="failure_certificate"/>
           <xs:enumeration value="failure_internal_error"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
  </xs:element>
  <xs:element name="SendCertificatesResponse">
<xs:complexType>
      <xs:sequence>
      <xs:element name="result">
     <xs:simpleType>
         <xs:restriction base="xs:string">
           <xs:enumeration value="ok_received_correctly"/>
           <xs:enumeration value="failure_syntax"/>
           <xs:enumeration value="failure_messageID_unknown"/>
           <xs:enumeration value="failure_internal_error"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
  </xs:element>
  <xs:element name="GetCACertificatesRequest">
<xs:complexType>
```

```
    <xs:sequence>
        <xs:element name="callerID" type="xs:string"/>
        <xs:element name="messageID" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
    </xs:element>
    <xs:element name="GetCACertificatesResponse">
  <xs:complexType>
    <xs:sequence>
        <xs:element ref="certificateSequence" minOccurs="0" maxOccurs="1"/>
        <xs:element name="result">
      <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="ok_cert_available"/>
            <xs:enumeration value="ok_reception_ack"/>
            <xs:enumeration value="failure_syntax"/>
            <xs:enumeration value="failure_internal_error"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    </xs:sequence>
  </xs:complexType>
    </xs:element>
    <xs:element name="GeneralMessageRequest">
  <xs:complexType>
    <xs:sequence>
        <xs:element name="callerID" type="xs:string"/>
        <xs:element name="messageID" type="xs:string"/>
        <xs:element name="subject" type="xs:string"/>
        <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
    </xs:element>
    <xs:element name="GeneralMessageResponse">
  <xs:complexType>
    <xs:sequence>
        <xs:element name="result">
      <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="ok"/>
            <xs:enumeration value="failure_syntax"/>
            <xs:enumeration value="failure_internal_error"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
  </xs:schema>
</wsdl:types>
```

```
<wsdl:message name="RequestCertificateRequest">
  <wsdl:part name="RequestCertificateRequest" element="SPOC:RequestCertificateRequest"/>
</wsdl:message>
<wsdl:message name="RequestCertificateResponse">
  <wsdl:part name="RequestCertificateResponse" element="SPOC:RequestCertificateResponse"/>
</wsdl:message>

<wsdl:message name="SendCertificatesRequest">
  <wsdl:part name="SendCertificatesRequest" element="SPOC:SendCertificatesRequest"/>
</wsdl:message>
<wsdl:message name="SendCertificatesResponse">
  <wsdl:part name="SendCertificatesResponse" element="SPOC:SendCertificatesResponse"/>
</wsdl:message>

<wsdl:message name="GetCACertificatesRequest">
  <wsdl:part name="GetCACertificatesRequest" element="SPOC:GetCACertificatesRequest"/>
</wsdl:message>
<wsdl:message name="GetCACertificatesResponse">
  <wsdl:part name="GetCACertificatesResponse" element="SPOC:GetCACertificatesResponse"/>
</wsdl:message>

<wsdl:message name="GeneralMessageRequest">
  <wsdl:part name="GeneralMessageRequest" element="SPOC:GeneralMessageRequest"/>
</wsdl:message>
<wsdl:message name="GeneralMessageResponse">
  <wsdl:part name="GeneralMessageResponse" element="SPOC:GeneralMessageResponse"/>
</wsdl:message>

<wsdl:portType name="SPOCPortType">
  <wsdl:operation name="RequestCertificate">
    <wsdl:input message="SPOC:RequestCertificateRequest"/>
    <wsdl:output message="SPOC:RequestCertificateResponse"/>
  </wsdl:operation>
  <wsdl:operation name="SendCertificates">
    <wsdl:input message="SPOC:SendCertificatesRequest"/>
    <wsdl:output message="SPOC:SendCertificatesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GetCACertificates">
    <wsdl:input message="SPOC:GetCACertificatesRequest"/>
    <wsdl:output message="SPOC:GetCACertificatesResponse"/>
  </wsdl:operation>
  <wsdl:operation name="GeneralMessage">
    <wsdl:input message="SPOC:GeneralMessageRequest"/>
    <wsdl:output message="SPOC:GeneralMessageResponse"/>
  </wsdl:operation>
</wsdl:portType>

<wsdl:binding name="SPOCSOAPBinding" type="SPOC:SPOCPortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <wsdl:operation name="RequestCertificate">
    <soap:operation soapAction="RequestCertificate"/>
```

```
      <wsdl:input>
          <soap:body parts="RequestCertificateRequest" use="literal"/>
        </wsdl:input>
        <wsdl:output>
          <soap:body parts="RequestCertificateResponse" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
      <wsdl:operation name="SendCertificates">
        <soap:operation soapAction="SendCertificates"/>
        <wsdl:input>
          <soap:body parts="SendCertificatesRequest" use="literal"/>
     </wsdl:input>
        <wsdl:output>
          <soap:body parts="SendCertificatesResponse" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
      <wsdl:operation name="GetCACertificates">
        <soap:operation soapAction="GetCACertificates"/>
        <wsdl:input>
          <soap:body parts="GetCACertificatesRequest" use="literal"/>
     </wsdl:input>
        <wsdl:output>
          <soap:body parts="GetCACertificatesResponse" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
      <wsdl:operation name="GeneralMessage">
        <soap:operation soapAction="GeneralMessage"/>
        <wsdl:input>
          <soap:body parts="GeneralMessageRequest" use="literal"/>
      </wsdl:input>
        <wsdl:output>
          <soap:body parts="GeneralMessageResponse" use="literal"/>
        </wsdl:output>
    </wsdl:operation>
     </wsdl:binding>

     <wsdl:service name="SPOC">
      <wsdl:port name="SPOCPort" binding="SPOC:SPOCSOAPBinding">
        <soap:address location="http://spoc-server/SPOC"/>
      </wsdl:port>
     </wsdl:service>
    </wsdl:definitions>
```

9. STRUCTURE DE LA LISTE DE CONTRÔLE DE L'ACSN

Les listes de contrôle sont implémentées comme des instances du type `ContentInfo`, comme le spécifie la norme RFC 5652. L'information `ContentInfo` DOIT contenir une seule instance du type `SignedData` comme l'indique le profil ci-dessous. Aucun autre type de données n'est inclus dans `ContentInfo`. Toutes les listes de contrôle DOIVENT être produites en format DER afin de préserver l'intégrité des signatures qu'elles contiennent.

9.1 Type SignedData

Les règles de traitement énoncées dans la norme RFC 5652 s'appliquent.

La spécification de la structure de la liste de contrôle emploie la terminologie suivante pour la présence de chaque champ :

- m obligatoire (*mandatory*) — le champ DOIT être présent ;
- r recommandé — le champ DEVRAIT être présent ;
- x ne pas utiliser — le champ NE DOIT PAS être présent ;
- o optionnel — le champ PEUT être présent.

Tableau 18. Liste de contrôle

Valeur		*Observations*
`SignedData`		
`Version`	m	Valeur = `v3`.
`digestAlgorithms`	m	
`encapContentInfo`	m	
`eContentType`	m	`id-icao-cscaMasterList`
`eContent`	m	Le contenu codé d'une `cscaMasterList`.
`Certificates`	m	Le certificat de signataire de liste de contrôle DOIT être inclus et le certificat de l'ACSN, qui peut être utilisé pour vérifier la signature du champ `signerInfos`, DEVRAIT être inclus.
`Crls`	x	
`signerInfos`	m	Il est RECOMMANDÉ que les États ne fournissent que 1 `signerinfo` dans ce champ.
`SignerInfo`	m	
`Version`	m	La valeur de ce champ est dictée par le champ `sid`. Voir les règles concernant ce champ dans RFC 5652.
`Sid`	m	

Valeur		*Observations*
`subjectKeyIdentifier`	r	Il est RECOMMANDÉ de prendre en charge ce champ plutôt que le champ `issuerandSerialNumber`.
`digestAlgorithm`	m	L'identificateur d'algorithme de l'algorithme utilisé pour produire la valeur de hachage sur `encapsulatedContent` et `SignedAttrs`. Voir la note ci-dessous.
`signedAttrs`	m	D'autres attributs peuvent figurer, mais les États récepteurs n'ont pas à les traiter sauf pour vérifier la valeur de la signature. `signedAttrs` DOIT inclure l'heure de signature (voir PKCS #9).
`signatureAlgorithm`	m	L'identificateur d'algorithme de l'algorithme utilisé pour produire la valeur de la signature et les paramètres qui pourraient y être associés. Voir la note ci-dessous.
`signature`	m	Le résultat du processus de génération de signature.
`unsignedAttrs`	o	Ce champ PEUT être inclus, mais les États récepteurs peuvent ne pas en tenir compte.

Note— `DigestAlgorithmIdentifiers` DOIT omettre les paramètres `NULL`, mais `SignatureAlgorithmIdentifier` (défini dans la norme RFC 3447) DOIT inclure `NULL` comme paramètre si aucun paramètre n'est présent, même lorsque les algorithmes SHA2 sont utilisés conformément à la norme RFC 5754. Les mises en œuvre DOIVENT accepter `DigestAlgorithmIdentifiers` avec les deux conditions, c'est-à-dire paramètres absents ou paramètres `NULL`.

9.2 Spécification ASN.1 de la liste de contrôle

```
CscaMasterList
{ iso-itu-t(2) international-organization(23) icao(136) mrtd(1)
security(1) masterlist(2)}

DEFINITIONS IMPLICIT TAGS ::=
BEGIN
```

```
IMPORTS

 -- Imports de RFC 5280 [PROFILE], Appendice A.1
     Certificate
       FROM PKIX1Explicit88
         { iso(1) identified-organization(3) dod(6)
           internet(1) security(5) mechanisms(5) pkix(7)
             mod(0) pkix1-explicit(18) };
-- Liste de contrôle de l'ACSN

CscaMasterListVersion ::= INTEGER {v0(0)}

CscaMasterList ::= SEQUENCE {
  version            CscaMasterListVersion,
  certList           SET OF Certificate }

-- Identificateurs d'objet

id-icao-cscaMasterList OBJECT IDENTIFIER ::=
                                 {id-icao-mrtd-security 2}
id-icao-cscaMasterListSigningKey OBJECT IDENTIFIER ::=
                                 {id-icao-mrtd-security 3}
END
```

10. STRUCTURE DE LA LISTE D'ÉCARTS

La liste d'écarts est mise en œuvre sous forme de type SignedData (données signées), comme il est spécifié dans la norme RFC 3852. Toutes les listes d'écarts DOIVENT être produites en utilisant les règles de codage distinctives (DER) pour préserver l'intégrité des signatures qu'elles contiennent.

La gamme d'écarts est limitée par :

- la plage de données (y compris la date de délivrance et la date d'expiration) ;
- le nom de l'émetteur et le numéro de série ;
- l'identificateur de clé du sujet de DSC ;
- la liste des numéros de DVLM-e.

Les combinaisons appropriées de ces valeurs seront employées pour limiter avec précision la gamme des DVLM touchés. Lorsque ces valeurs sont combinées, elles sont traitées comme si elles étaient liées par « ET ». Il n'y a aucune option de traitement des valeurs avec la liaison « OU ».

Tableau 19. Liste d'écarts

Valeur		Observations
SignedData		
version	m	Valeur = v3
digestAlgorithms	m	
encapContentInfo	m	
eContentType	m	id-icao-DeviationList
eContent	m	Contenu codé de DefectList
certificates	m	Les États DOIVENT inclure le certificat de signataire de la liste d'écarts et DEVRAIENT inclure le certificat ACSN, qui peut être utilisé pour vérifier la signature dans le champ signerInfos.
crls	x	
signerInfos	m	Il est RECOMMANDÉ que les États ne fournissent que 1 signerInfo dans ce champ.
SignerInfo	m	
version	m	La valeur de ce champ est dictée par le champ sid. Voir les règles concernant ce champ dans la norme RFC 3852, § 5.3.
sid	m	
subjectKeyIdentifier	r	Il est RECOMMANDÉ que les États prennent en charge ce champ plutôt que le champ issuerandSerialNumber.
digestAlgorithm	m	L'identificateur d'algorithme de l'algorithme utilisé pour produire la valeur de hachage sur encapsulatedContent et SignedAttrs.
signedAttrs	m	Les États producteurs voudront peut-être inclure des attributs supplémentaires à insérer dans la signature, mais ces attributs n'ont pas à être traités par les États récepteurs, sauf pour vérifier la valeur de la signature. signedAttrs DOIT inclure l'heure de signature (réf. PKCS#9).
signatureAlgorithm	m	L'identificateur d'algorithme de l'algorithme utilisé pour produire la valeur de la signature et les paramètres qui pourraient y être associés.
signature	m	Le résultat du processus de génération de signature.
unsignedAttrs	x	

10.1 Type SignedData

Les règles de traitement figurant dans la norme RFC 3852 s'appliquent :

- m obligatoire (mandatory) — le champ DOIT être présent ;
- r recommandé — le champ DEVRAIT être présent ;
- x ne pas utiliser — le champ NE DOIT PAS être rempli ;
- o optionnel — le champ PEUT être présent.

10.2 Spécification ASN.1

```
DeviationList
{ joint-iso-itu-t (2) international-organization(23) icao(136) mrtd(1) security(1)
deviationlist(7)}

DEFINITIONS IMPLICIT TAGS ::=
BEGIN

IMPORTS

  -- Imports from RFC 3280 [PROFILE], Appendix A.1
       AlgorithmIdentifier
          FROM PKIX1Explicit88
                 { iso(1) identified-organization(3) dod(6)
                   internet(1) security(5) mechanisms(5) pkix(7)
                   mod(0) pkix1-explicit(18) }

  -- Imports from RFC 3852
       SubjectKeyIdentifier, Digest, IssuerAndSerialNumber
          FROM CryptographicMessageSyntax2004
                 { iso(1) member-body(2) us(840) rsadsi(113549)
                   pkcs(1) pkcs-9(9) smime(16) modules(0)
                   cms-2004(24) };

DeviationListVersion ::= INTEGER {v0(0)}

DeviationList ::= SEQUENCE {
 version      DeviationListVersion,
 digestAlgorithm AlgorithmIdentifier OPTIONAL,
 deviations   SET OF Deviation
}

Deviation ::= SEQUENCE{
 documents       DeviationDocuments,
 descriptions SET OF DeviationDescription
}
```

```
DeviationDescription ::= SEQUENCE{
 description     PrintableString OPTIONAL,
 deviationType   OBJECT IDENTIFIER,
 parameters   [0] ANY DEFINED BY deviationType OPTIONAL,
 nationalUse     [1] ANY OPTIONAL

 -- The nationalUse field is for internal State use, and is not governed
 -- by an ICAO specification.
}

DeviationDocuments ::= SEQUENCE {
 documentType    [0] PrintableString (SIZE(2)) OPTIONAL,
          -- per MRZ, e.g. 'P'
 dscIdentifier   DocumentSignerIdentifier OPTIONAL,
 issuingDate     [4] IssuancePeriod OPTIONAL,
 documentNumbers    [5] SET OF PrintableString OPTIONAL
}

DocumentSignerIdentifier ::= CHOICE{
 issuerAndSerialNumber [1] IssuerAndSerialNumber,
 subjectKeyIdentifier [2] SubjectKeyIdentifier,
 certificateDigest  [3] Digest   --  if  used,  digestAlgorithm  must  be  present  in
DeviationList
}

IssuancePeriod ::= SEQUENCE {
 firstIssued   GeneralizedTime,
 lastIssued   GeneralizedTime
}

-- CertField is used to define which part of a certificate is
  affected by a coding error. Parts of the Body are identified by
  the corresponding value of CertificateBodyField, extensions
  by the corresponding OID identifying the extension.

CertField ::= CHOICE {
 body  CertificateBodyField,
 extension OBJECT IDENTIFIER
}
CertificateBodyField ::= INTEGER {
 generic(0), version(1), serialNumber(2), signature(3), issuer(4),
 validity(5), subject(6), subjectPublicKeyInfo(7),
 issuerUniqueID(8), subjectUniqueID(9)
}

Datagroup ::= INTEGER
              {dg1(1), dg2(2), dg3(3), dg4(4), dg5(5), dg6(6),
               dg7(7), dg8(8), dg9(9), dg10(10), dg11(11),
               dg12(12), dg13(13), dg14(14), dg15(15), dg16(16),
               sod(20), com(21)}
```

```
MRZField ::= INTEGER
             {generic(0), documentCode(1), issuingState(2), personName(3),
              documentNumber(4), nationality(5), dateOfBirth(6),
              sex(7), dateOfExpiry(8), optionalData(9)}

-- Base Object Identifiers

id-icao OBJECT IDENTIFIER ::= {2 23 136 }
id-icao-mrtd OBJECT IDENTIFIER ::= {id-icao 1}
id-icao-mrtd-security OBJECT IDENTIFIER ::= {id-icao-mrtd 1}
id-icao-DeviationList OBJECT IDENTIFIER ::= {id-icao-mrtd-security 7}
id-icao-DeviationListSigningKey OBJECT IDENTIFIER ::= {id-icao-mrtd-security 8}

-- Deviation Object Identifiers and Parameter Definitions

id-Deviation-CertOrKey OBJECT IDENTIFIER ::= {id-icao-DeviationList 1}
id-Deviation-CertOrKey-DSSignature OBJECT IDENTIFIER ::= {id-Deviation-CertOrKey 1}
id-Deviation-CertOrKey-DSEncoding OBJECT IDENTIFIER ::= {id-Deviation-CertOrKey 2}
id-Deviation-CertOrKey-CSCAEncoding OBJECT IDENTIFIER ::= {id-Deviation-CertOrKey 3}
id-Deviation-CertOrKey-AAKeyCompromised OBJECT IDENTIFIER ::= {id-Deviation-CertOrKey 4}
id-Deviation-LDS OBJECT IDENTIFIER ::= {id-icao-DeviationList 2}
id-Deviation-LDS-DGMalformed OBJECT IDENTIFIER ::= {id-Deviation-LDS 1}
id-Deviation-LDS-DGHashWrong OBJECT IDENTIFIER ::= {id-Deviation-LDS 2}
id-Deviation-LDS-SODSignatureWrong OBJECT IDENTIFIER ::= {id-Deviation-LDS 3}
id-Deviation-LDS-COMInconsistent OBJECT IDENTIFIER ::= {id-Deviation-LDS 4}

id-Deviation-MRZ OBJECT IDENTIFIER ::= {id-icao-DeviationList 3}
id-Deviation-MRZ-WrongData OBJECT IDENTIFIER ::= {id-Deviation-MRZ 1}
id-Deviation-MRZ-WrongCheckDigit OBJECT IDENTIFIER ::= {id-Deviation-MRZ 2}

id-Deviation-Chip OBJECT IDENTIFIER ::= {id-icao-DeviationList 4}

id-Deviation-NationalUse OBJECT IDENTIFIER ::= {id-icao-DeviationList 5}

END
```

11. RÉFÉRENCES (NORMATIVES)

FIPS 180-2	FIPS 180-2, Federal Information Processing Standards Publication (FIPS PUB) 180-2, *Secure Hash Standard*, août 2002.
FIPS 186-4	FIPS 186-4, Federal Information Processing Standards Publication (FIPS PUB) 186-4, *Digital Signature Standard (DSS)*, juillet 2013 (Remplace FIPS PUB 186-3 de juin 2009).
ISO 3166-1	ISO/IEC 3166-1: 2006, Codes for the representation of names of countries and their subdivisions — Part 1: Country Codes. (Codes pour la représentation des noms de pays et de leurs subdivisions — Partie 1 : Codes de pays).

ISO/IEC 15946	ISO/IEC 15946: 2002, Information technology — Security techniques — Cryptographic techniques based on elliptic curves. (Technologies de l'information — Techniques de sécurité — Techniques cryptographiques fondées sur les courbes elliptiques).
RFC 3280	RFC 3280, R. Housley, W. Polk, W. Ford, D. Solo, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, avril 2002.
RFC 4055	RFC 4055, J. Schaad, B. Kaliski, R. Housley, Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, juin 2005.
RFC 5652	RFC 5652, R. Housley, Cryptographic Message Syntax, septembre 2009.
RFC 5280	RFC 5280, D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, mai 2008.
TR 03111	BSI TR-03111: Elliptic Curve Cryptography v 2.0, 2012.
X9.62	X9.62, Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 7 janvier1999.
X.509	ITU-T X.509 | ISO/IEC 9594-8, 2008: Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks. (Technologies de l'information — Interconnexion des systèmes ouverts — L'annuaire : cadre général des certificats de clé publique et d'attribut).
X.690	ITU-T X.690 2008: Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER). (Technologies de l'information — Règles de codage ASN.1 : spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives).
RFC-RSA	Jonsson, Jakob and Kaliski, Burt RFC 3447, Public-key cryptography standards (PKCS)#1: RSA cryptography specifications version 2.1, 2003. (Normes de cryptographie à clé publique (PKCS) n° 1 : Spécifications de la cryptographie RSA version 2.1)
PKCS#1	RSA Laboratories RSA Laboratories Technical Note, PKCS#1 v2.2: RSA cryptography standard, 2012
TLSAES	Chown, P., "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)", RFC 3268, juin 2002
TLSECC	Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., and B. Moeller, "Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)", RFC 4492, mai 2006
TLS1.2	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, août 2008
TLSEXT	Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., ry T. Wright, « Transport Layer Security (TLS) Extensions », RFC 4366, avril 2006

SOAP	SOAP Version 1.2 Partie 1 : Messaging Framework (deuxième édition), W3C Recommendation 27 avril 2007
HTTPS	E. Rescorla., « HTTP Over TLS », RFC 2818, mai 2000
WSI-BP	WS-I Basic Profile, disponible à l'adresse : http://www.ws-i.org/Profiles/BasicProfile-1.1.html
WSI-SSBP	WS-I Basic Binding, disponible à l'adresse : http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0.html

— — — — — — — —

Appendice A à la Partie 12
(INFORMATIF)

DURÉES DE VIE

Les exemples suivants illustrent le calcul de la période d'utilisation d'une clé privée et de la période de validité d'un certificat de clé publique pour divers scénarios décrits à la section 4.

A.1 EXEMPLE 1

Dans le premier exemple, les DVLM-e ont une période de validité de cinq ans. Vu le nombre relativement élevé de DVLM-e qui sont émis chaque jour, la politique consiste à fixer à un minimum la période d'utilisation de la clé privée et la période de validité du certificat de clé publique. Dans cet exemple, la période minimale d'utilisation de la clé privée pour les certificats de signataire de documents est d'un mois.

Élément	***Période d'utilisation/de validité***
Validité du DVLM-e	5 ans
Période d'utilisation de la clé privée du signataire de document	1 mois
Période de validité du certificat de signataire de document	5 ans + 1 mois
Période d'utilisation de la clé privée de l'ACSN	3 ans
Période de validité du certificat de l'ACSN	8 ans + 1 mois

Cet exemple a les conséquences suivantes : à l'expiration de la période de validité du certificat de l'ACSN, au moins 36 certificats de signataire de document auront été émis (un pour chaque clé privée dont la période d'utilisation est d'un mois). Dans les derniers mois avant l'expiration de la validité du premier certificat de l'ACSN, au moins deux autres certificats de l'ACSN seront émis (un pour chaque clé privée dont la période d'utilisation est de trois ans).

A.2 EXEMPLE 2

Dans le deuxième exemple, les DVLM-e ont une période de validité de dix ans. La politique est d'attribuer une durée moyenne à la période d'utilisation de la clé privée et à la période de validité du certificat de clé publique.

Élément	***Période d'utilisation/de validité***
Validité du DVLM-e	10 ans
Période d'utilisation de la clé privée du signataire de document	2 mois
Période de validité du certificat de signataire de document	10 ans + 2 mois

Élément	*Période d'utilisation/de validité*
Période d'utilisation de la clé privée de l'ACSN	4 ans
Période de validité du certificat de l'ACSN	14 ans + 2 mois

Cet exemple a les conséquences suivantes : à l'expiration de la période de validité du certificat de l'ACSN, au moins 24 certificats de signataire de document auront été émis (un pour chaque clé privée dont la période d'utilisation est de deux mois). Dans les derniers mois avant l'expiration de la validité du premier certificat de l'ACSN, au moins trois autres certificats de l'ACSN seront émis (un pour chaque clé privée dont la période d'utilisation est de quatre ans).

A.3 EXEMPLE 3

Dans le dernier exemple, les DVLM-e ont une période de validité de 10 ans et la politique est d'utiliser la durée maximale de la période d'utilisation de la clé privée et de la période de validité du certificat de clé publique.

Élément	*Période d'utilisation/de validité*
Validité du DVLM-e	10 ans
Période d'utilisation de la clé privée du signataire de document	3 mois
Période de validité du certificat de signataire de document	10 ans + 3 mois
Période d'utilisation de la clé privée de l'ACSN	5 ans
Période de validité du certificat de l'ACSN	15 ans + 3 mois

Cet exemple a les conséquences suivantes : à l'expiration de la période de validité du certificat de l'ACSN, au moins 20 certificats de signataire de document auront été émis (un pour chaque clé privée dont la période d'utilisation est de trois mois). Dans les derniers mois avant l'expiration de la validité du premier certificat de l'ACSN, au moins trois autres certificats de l'ACSN seront émis (un pour chaque clé privée dont la période d'utilisation est de cinq ans).

— — — — — — — —

Appendice B à la Partie 12
(INFORMATIF)

TEXTE DE RÉFÉRENCE DES PROFILS DE CERTIFICAT ET DE CRL

Les profils de certificat et de CRL définis à la section 7 sont fondés sur des définitions et des spécifications de profil de base énoncées dans les documents de référence. Les tableaux suivants présentent de brefs extraits des sections pertinentes de ces documents sources (au moment de la rédaction du présent document). Ces extraits sont fournis pour aider le lecteur à comprendre le contexte de quelques-unes des exigences spécifiées dans les profils de certificat et de CRL des DVLM-e. Ils ne visent pas à remplacer les documents de référence. Pour obtenir la spécification complète du composant ou de l'extension mentionnés et pour obtenir la spécification la plus récente, il FAUT dans tous les cas se reporter aux documents de référence eux-mêmes.

Tableau B-1. Champs et extensions de certificat

Composant / extension	*Référence*	*Extraits applicables*
`Certificate`	RFC 5280 – § 4.1.1	
`TBSCertificate`	RFC 5280 – § 4.1.1.1	
`signatureAlgorithm`	RFC 5280 – § 4.1.1.2	
`signatureValue`	RFC 5280 – § 4.1.1.3	
`TBSCertificate`	RFC 5280 – § 4.1.2	
`version`	RFC 5280 – § 4.1.2.1	Lorsque les extensions sont utilisées, comme il est prévu dans ce profil, la version DOIT être 3 (la valeur est `2`).
`serialNumber`	RFC 5280 – § 4.1.2.2	Le numéro de série DOIT être un nombre entier positif attribué par l'AC à chaque certificat. Il DOIT être unique pour chaque certificat émis par une AC donnée (c.-à-d. le nom et le numéro de série de l'émetteur désignent un certificat unique). Les AC DOIVENT forcer le composant `serialNumber` à être un entier non négatif. Vu la nécessité d'avoir un numéro unique, on peut s'attendre à ce que les numéros de série soient des nombres entiers longs. Les utilisateurs de certificats DOIVENT être capables de traiter des valeurs allant jusqu'à 20 octets pour `serialNumber`.

Composant / extension	*Référence*	*Extraits applicables*
		Les AC conformes NE DOIVENT PAS utiliser des valeurs supérieures à 20 octets pour `serialNumber`.
	X.690 – § 8.3.2	Si le champ de contenu du codage d'un entier comporte plus d'un octet, les bits du premier octet et le bit 8 du deuxième octet : a) ne seront pas tous égaux à 1 ; et b) ne seront pas tous égaux à 0. *Note*.— Ces règles assurent le codage d'une valeur entière sur le plus petit nombre d'octets.
	X.690 – § 8.3.3	Le champ de contenu sera la représentation binaire en complément à deux de l'entier, et sera composé des bits 8 à 1 du premier octet, suivis des bits 8 à 1 du deuxième octet, et ainsi de suite jusques et y compris le dernier des octets du champ de contenu.
`signature`	RFC 5280 – § 4.1.1.2	Ce champ DOIT contenir le même identificateur d'algorithme que le champ `signatureAlgorithm` dans la séquence `Certificate`.
`issuer`	RFC 5280 – Appendice A.1	`X520countryName ::= PrintableString (SIZE (2))` `X520SerialNumber ::= PrintableString (SIZE (1..ub-serial-number))`
	RFC 5280 – § 4.1.2.4	Les AC conformes à ce profil DOIVENT utiliser soit le codage `PrintableString`, soit le codage `UTF8String` de `DirectoryString`.
	ISO 3166-1	
`validity`	RFC 5280 – § 4.1.2.5	Les composants `notBefore` et `notAfter` peuvent être codés en `UTCTime` ou en `GeneralizedTime`. Les AC conformes à ce profil DOIVENT toujours coder les dates de validité des certificats jusqu'en 2049 inclusivement en `UTCTime`. Les dates de validité commençant en 2050 DOIVENT être codées en `GeneralizedTime`.
`(if encoded as UTCTime)`	X.690 – § 11.8.1	Le codage se terminera par un « Z », comme indiqué dans le paragraphe consacré au temps `UTCTime` de la Recommandation UIT-T X.680 | ISO/CEI 8824-1.
	X.690 – § 11.8.2	L'élément « secondes » sera toujours présent.
`(if encoded as GeneralizedTime)`	X.690 – § 11.7.1	Le codage se terminera par un « Z », comme indiqué dans `GeneralizedTime` de la Recommandation UIT-T X.680 | ISO/CEI 8824-1.
	X.690 – § 11.7.2	L'élément « secondes » sera toujours présent.

Composant / extension	***Référence***	***Extraits applicables***
	RFC 5280 – § 4.1.2.5.2	Les valeurs de `GeneralizedTime` NE DOIVENT PAS inclure de fractions de seconde. Pour les fins de ce profil, les valeurs de `GeneralizedTime` DOIVENT être exprimées en temps moyen de Greenwich (Zulu) et DOIVENT inclure les secondes (c.-à-d., AAAAMMJJHHMMSSZ), même si le nombre de secondes est zéro.
`subject`	RFC 5280 – Appendice A.1	`X520countryName ::= PrintableString (SIZE (2))` `X520SerialNumber ::= PrintableString (SIZE (1..ub-serial-number))`
	RFC 5280 – § 4.1.2.6	Les AC conformes à ce profil DOIVENT utiliser soit le codage `PrintableString`, soit le codage `UTF8String` de `DirectoryString`.
`subjectPublicKeyInfo`	RFC 5280 – § 4.1.2.7	
`issuerUniqueID`	RFC 5280 – § 4.1.2.8	Les AC conformes à ce profil NE DOIVENT PAS générer des certificats ayant des identificateurs uniques.
`subjectUniqueID`	RFC 5280 – § 4.1.2.8	Les AC conformes à ce profil NE DOIVENT PAS générer des certificats ayant des identificateurs uniques.
`extensions`	X.690 – § 11.5	La valeur d'un composant d'un ensemble (`set`) ou d'une séquence (`sequence`) ne sera pas codée si elle est égale à sa valeur par défaut.
`AuthorityKeyIdentifier`	RFC 5280 – § 4.2.1.1	Le champ `keyIdentifier` de l'extension `authorityKeyIdentifier` DOIT être inclus dans tous les certificats générés par des AC conformes pour faciliter la construction de l'itinéraire de certification. Il y a une exception : lorsque l'AC distribue sa clé publique sous forme de certificat « autosigné », l'identificateur de clé de l'autorité PEUT être omis.
`keyIdentifier`		
`authorityCertIssuer`		
`authorityCertSerialNumber`		
`SubjectKeyIdentifier`	RFC 5280 – § 4.2.1.2	Pour faciliter la construction de l'itinéraire de certification, cette extension DOIT figurer dans tous les certificats d'AC conformes, c'est-à-dire tous les certificats comprenant l'extension contraintes de base (§ 4.2.1.9) où la valeur de `cA` est `TRUE`.

Composant / extension	*Référence*	*Extraits applicables*
subjectKeyIdentifier		
KeyUsage	RFC 5280 – § 4.2.1.3	La restriction d'utilisation pourrait être employée lorsqu'il faut restreindre une clé qui pourrait être utilisée pour plus d'une opération.
digitalSignature		Le bit digitalSignature est positionné lorsque la clé publique d'un sujet est utilisée avec un mécanisme de signature numérique pour prendre en charge des services de sécurité autres que la signature de certificat (bit 5) ou la signature de CRL (bit 6).
nonRepudiation		
keyEncipherment		
dataEncipherment		
keyAgreement		
keyCertSign		Le bit keyCertSign est positionné lorsque la clé publique d'un sujet est utilisée pour vérifier une signature sur des certificats de clé publique.
cRLSign		Le bit cRLSign est positionné lorsque la clé publique d'un sujet est utilisée pour vérifier une signature sur une liste de certificats révoqués (p. ex., une CRL, une CRL delta ou une ARL). Ce bit DOIT être positionné dans les certificats qui sont utilisés pour vérifier les signatures sur les CRL.
encipherOnly		
decipherOnly		
PrivateKeyUsagePeriod	RFC 3280 – § 4.2.1.4	Les AC conformes à ce profil NE DOIVENT PAS générer de certificats avec des extensions de période d'utilisation de clé privée sauf si au moins un des deux composants est présent et si l'extension est non critique.
notBefore		Lorsqu'ils sont utilisés, notBefore et notAfter sont représentés sous forme de GeneralizedTime et DOIVENT être spécifiés et interprétés conformément au § 4.1.2.5.2.
notAfter		
CertificatePolicies	RFC 5280 – § 4.2.1.4	Si cette extension est critique, le logiciel de validation d'itinéraire DOIT être capable d'interpréter cette extension (y compris le qualificatif optionnel) ou DOIT refuser le certificat.

Composant / extension	*Référence*	*Extraits applicables*
PolicyInformation		
policyIdentifier		
policyQualifiers		
PolicyMappings	RFC 5280 – § 4.2.1.5	
SubjectAltName	RFC 5280 – § 4.2.1.6	
IssuerAltName	RFC 5280 – § 4.2.1.7	
SubjectDirectoryAttributes	RFC 5280 – § 4.2.1.8	
Basic Constraints	RFC 5280 – § 4.2.1.9	L'extension contraintes de base indique si le sujet du certificat est une AC et la profondeur maximale des itinéraires de certification valides qui contiennent ce certificat. Les AC conformes DOIVENT inclure cette extension dans tous les certificats d'AC qui contiennent des clés publiques utilisées pour valider des signatures numériques sur des certificats et DOIVENT marquer l'extension comme critique.
cA		La valeur booléenne de cA indique si la clé publique certifiée appartient à une AC. Si la valeur booléenne de cA n'est pas déclarée, le bit keyCertSign de l'extension utilisation de clé NE DOIT PAS être positionné.
PathLenConstraint		
NameConstraints	RFC 5280 – § 4.2.1.10	
PolicyConstraints	RFC 5280 – § 4.2.1.11	
ExtKeyUsage	RFC 5280 – § 4.2.1.12	Cette extension indique un ou plusieurs buts possibles pour l'utilisation de la clé publique certifiée, en plus des buts de base indiqués dans le champ d'extension d'utilisation de clé.
CRLDistributionPoints	RFC 5280 – § 4.2.1.13	
distributionPoint		
reasons		
cRLIssuer		

Composant / extension	*Référence*	*Extraits applicables*
`InhibitAnyPolicy`	RFC 5280 – § 4.2.1.14	
`FreshestCRL`	RFC 5280 – § 4.2.1.15	
`privateInternetExtensions`	RFC 5280 – § 4.2.2	
`NameChange`		
`DocumentType`		
`Netscape Certificate Type`		
`other private extensions`		

Tableau B-2. Champs et extensions de CRL

Composant / extension	*Référence*	*Extraits applicables*
`CertificateList`	RFC 5280 – § 5.1.1	
`tBSCertList`	RFC 5280 – § 5.1.1.1	
`signatureAlgorithm`	RFC 5280 – § 5.1.1.2	
`signatureValue`	RFC 5280 – § 5.1.1.3	
	RFC 5280 – § 5.1.2	
`version`	RFC 5280 – § 5.1.2.1	Ce champ optionnel décrit la version de la CRL codée. Lorsque des extensions sont utilisées, comme le requiert ce profil, ce champ DOIT être présent et DOIT spécifier la version 2 (la valeur d'entier est `1`).
`signature`	RFC 5280 – § 5.1.2.2	Ce champ DOIT contenir le même identificateur d'algorithme que le champ signature de la séquence `CertificateList`.
`issuer`	RFC 5280 – Appendice A.1	`X520countryName ::= PrintableString (SIZE (2))` `X520SerialNumber ::= PrintableString (SIZE 1..ub-serial-number))`
	RFC 5280 – § 5.1.2.3 et 4.1.2.4	Les AC conformes à ce profil DOIVENT utiliser soit le codage `PrintableString`, soit le codage `UTF8String` de `DirectoryString`.
`thisUpdate`	RFC 5280 – § 5.1.2.4	Les émetteurs de CRL conformes à ce profil DOIVENT coder `thisUpdate` en temps `UTCTime` pour les dates allant jusqu'en 2049 inclusivement. Les émetteurs de CRL conformes à ce profil DOIVENT coder `thisUpdate` en `GeneralizedTime` pour les dates commençant en 2050.
`(if encoded as UTCTime)`	X.690 – § 11.8.1	Le codage se terminera par un « Z », comme indiqué dans le paragraphe consacré au temps `UTCTime` de la Recommandation UIT-T X.680 | ISO/CEI 8824-1.
	X.690 – § 11.8.2	L'élément « secondes » sera toujours présent.
`(if encoded as GeneralizedTime)`	X.690 – § 11.7.1	Le codage se terminera par un « Z », comme indiqué dans la disposition relative au temps généralisé `GeneralizedTime` de la Recommandation UIT-T X.680 | ISO/CEI 8824-1.
	X.690 – § 11.7.2	L'élément « secondes » sera toujours présent.
	RFC 5280 – § 4.1.2.5.2	Les valeurs de `GeneralizedTime` NE DOIVENT PAS inclure de fractions de seconde.

Composant / extension	*Référence*	*Extraits applicables*
		Pour les fins de ce profil, les valeurs de `GeneralizedTime` DOIVENT être exprimées en temps moyen de Greenwich (Zulu) et DOIVENT inclure les secondes (soit, AAAAMMJJHHMMSSZ), même si le nombre de secondes est zéro.
`nextUpdate`	§ 5.1.2.5	Les émetteurs de CRL conformes à ce profil DOIVENT coder les dates de `nextUpdate` allant jusqu'en 2049 inclusivement en `UTCTime`. Les émetteurs de CRL conformes à ce profil DOIVENT coder les dates de `nextUpdate` commençant en 2050 en `GeneralizedTime`.
`(if encoded at UTCTime)`	X.690 – § 11.8.1	Le codage se terminera par un « Z », comme indiqué dans le paragraphe consacré au temps `UTCTime` de la Recommandation UIT-T X.680 | ISO/CEI 8824-1.
	X.690 – § 11.8.2	L'élément « secondes » sera toujours présent.
`(if encoded at GeneralizedTime)`	X.690 – § 11.7.1	Le codage se terminera par un « Z », comme indiqué dans la disposition relative au temps généralisé `GeneralizedTime` de la Recommandation UIT-T X.680 | ISO/CEI 8824-1.
	X.690 – § 11.7.2	L'élément « secondes » sera toujours présent.
	RFC 5280 – § 4.1.2.5.2	Les valeurs `GeneralizedTime` NE DOIVENT PAS inclure de fractions de seconde. Pour les fins de ce profil, les valeurs de `GeneralizedTime` DOIVENT être exprimées en temps moyen de Greenwich (Zulu) et DOIVENT inclure les secondes (soit, AAAAMMJJHHMMSSZ), même si le nombre de secondes est zéro.
`revokedCertificates`	RFC 5280 – § 5.1.2.6	Lorsqu'il n'y a pas de certificats révoqués, la liste de certificats révoqués DOIT être absente. Autrement, les certificats révoqués sont énumérés par numéro de série.
`crlExtensions`	RFC 5280 – § 5.2	Les émetteurs de CRL conformes DOIVENT inclure les extensions identificateur de clé d'autorité (§ 5.2.1) et numéro de CRL (§ 5.2.3) dans toutes les CRL émises.
	X.690 – § 11.5	La valeur d'un composant d'un ensemble (`set`) ou d'une séquence (`sequence`) ne sera pas codée si elle est égale à sa valeur par défaut.

`authorityKeyIdentifier`	RFC 5280 – § 5.2.1	Les émetteurs de CRL conformes DOIVENT utiliser la méthode d'identificateur de clé et DOIVENT inclure cette extension dans toutes les CRL émises.
`issuerAlternativeName`	RFC 5280 – § 5.2.2	
`cRLNumber`	RFC 5280 – § 5.2.3	Les émetteurs de CRL conformes à ce profil DOIVENT inclure cette extension dans toutes les CRL et DOIVENT marquer cette extension comme non critique. `CRLNumber ::= INTEGER (0..MAX)` Compte tenu de ces exigences, on peut s'attendre à ce que les numéros de CRL soient des nombres entiers longs. Les vérificateurs de CRL DOIVENT être capables de traiter des valeurs allant jusqu'à 20 octets pour le composant `CRLNumber`. Les émetteurs de CRL conformes NE DOIVENT PAS utiliser des valeurs supérieures à 20 octets pour `CRLNumber`.
	X.690 – § 8.3.2	Si le champ de contenu du codage d'un entier comporte plus d'un octet, les bits du premier octet et le bit 8 du deuxième octet : a) ne seront pas tous égaux à 1 ; et b) ne seront pas tous égaux à 0. *Note*.— Ces règles assurent le codage d'une valeur entière sur le plus petit nombre d'octets.
	X.690 – § 8.3.3	Le contenu des octets doit être un nombre binaire à complément à deux égal à la valeur entière et doit être composé des bits 8 à 1 du premier octet, suivis des bits 8 à 1 du deuxième octet, suivis des bits 8 à 1 de chaque octet à tour de rôle jusqu'à et y compris le dernier octet du contenu des octets.
`deltaCRLIndicator`	RFC 5280 – § 5.2.4	
`issuingDistribution Point`	RFC 5280 – § 5.2.5	
`freshestCRL`	RFC 5280 – § 5.2.6	
`reasonCode`	RFC 5280 – § 5.3.1	
`holdInstructionCode`	RFC 5280 – § 5.3.2	
`invalidityDate`	RFC 5280 – § 5.3.3	
`certificateIssuer`	RFC 5280 – § 5.3.4	

— — — — — — — —

Appendice C à la Partie 12
(INFORMATIF)

PROFILS DE CERTIFICATS PRÉCÉDENTS

Les profils de certificat traités dans le présent appendice ont été spécifiés dans la sixième édition du Doc 9303 de l'OACI. Même si les ACSN DOIVENT émettre des certificats conformes aux profils actuels spécifiés dans la section 7, les profils précédents sont indiqués ci-après à titre d'information seulement, vu que les certificats émis conformément aux profils précédents seront en circulation et seront traités par les systèmes d'inspection pendant encore plusieurs années.

Tableau 10-1. Corps du certificat

Composant du certificat	*Section de RFC 3280*	*Certificat de l'ACSN*	*Certificat de signataire de document*	*Observations*
`Certificate`	4.1.1	m	m	
`TBSCertificate`	4.1.1.1	m	m	Voir le Tableau C-2.
`SignatureAlgorithm`	4.1.1.2	m	m	La valeur insérée dépend de l'algorithme choisi.
`SignatureValue`	4.1.1.3	m	m	La valeur insérée dépend de l'algorithme choisi.
`TBSCertificate`	4.1.2			
`version`	4.1.2.1	m	m	DOIT être `v3`.
`serialNumber`	4.1.2.2	m	m	
`signature`	4.1.2.3	m	m	La valeur insérée DOIT correspondre à l'OID dans `signatureAlgorithm`.
`issuer`	4.1.2.4	m	m	
`validity`	4.1.2.5	m	m	Les mises en œuvre DOIVENT spécifier que le temps UTC sera utilisé jusqu'à 2049, après quoi le temps généralisé `GeneralizedTime` sera utilisé.
`subject`	4.1.2.6	m	m	.

Composant du certificat	*Section de RFC 3280*	*Certificat de l'ACSN*	*Certificat de signataire de document*	*Observations*
subjectPublicKeyInfo	4.1.2.7	m	m	
issuerUniqueID	4.1.2.8	x	x	
subjectUniqueID	4.1.2.8	x	x	
extensions	4.1.2.9	m	m	Voir le Tableau C-2 suivant pour les extensions qui DEVRAIENT être présentes.

Tableau C-2. Extensions

Nom de l'extension	*Paragrahe dans RFC 3280*	*Certificat de l'ACSN*	*Certificat de signataire de document*	*Observations*
AuthorityKeyIdentifier	4.2.1.1	o	m	Obligatoire dans tous les certificats sauf dans les certificats de l'ACSN autosignés.
SubjectKeyIdentifier	4.2.1.2	m	o	
KeyUsage	4.2.1.3	mc	mc	Cette extension DOIT être marquée CRITIQUE.
PrivateKeyUsagePeriod	4.2.1.4	o	o	Il s'agit de la période d'émission de la clé privée.
CertificatePolicies	4.2.1.5	o	o	
PolicyMappings	4.2.1.6	x	x	
SubjectAltName	4.2.1.7	x	x	
IssuerAltName	4.2.1.8	x	x	
SubjectDirectoryAttributes	4.2.1.9	x	x	
BasicConstraints	4.2.1.10	mc	x	Cette extension DOIT être marquée CRITIQUE.
NameConstraints	4.2.1.11	x	x	
PolicyConstraints	4.2.1.12	x	x	
ExtKeyUsage	4.2.1.13	x	x	

Nom de l'extension	*Paragrahe dans RFC 3280*	*Certificat de l'ACSN*	*Certificat de signataire de document*	*Observations*
CRLDistributionPoints	4.2.1.14	o	o	Si les États émetteurs ou les organisations émettrices décident d'utiliser cette extension, ils DOIVENT inclure le RCP de l'OACI comme point de distribution. Les mises en œuvre peuvent également inclure des DP LCR relatifs à des fins locales ; celles-ci peuvent être ignorées par d'autres États récepteurs.
InhibitAnyPolicy	4.2.1.15	x	x	
FreshestCRL	4.2.1.16	x	x	
privateInternetExtensions	4.2.2	x	x	
other private extensions	s.o.	o	o	Si une extension privée est utilisée à des fins nationales, elle NE DOIT PAS être marquée. Il est déconseillé aux États émetteurs et aux organisations émettrices d'inclure des extensions privées.
AuthorityKeyIdentifier	4.2.1.1			
keyIdentifier		m	m	Si cette extension est utilisée, il FAUT au minimum que ce champ soit pris en charge.
authorityCertIssuer		o	o	
authorityCertSerialNumber		o	o	
SubjectKeyIdentifier	4.2.1.2			
subjectKeyIdentifier		m	m	
KeyUsage	4.2.1.3			
digitalSignature		x	m	
nonRepudiation		x	x	
keyEncipherment		x	x	
dataEncipherment		x	x	
keyAgreement		x	x	

Nom de l'extension	*Paragrahe dans RFC 3280*	*Certificat de l'ACSN*	*Certificat de signataire de document*	*Observations*
keyCertSign		m	x	
cRLSign		m	x	
encipherOnly		x	x	
decipherOnly		x	x	
BasicConstraints	4.2.1.10			
cA		m	x	TRUE pour les certificats d'AC.
PathLenConstraint		m	x	0 pour un nouveau certificat de l'ACSN ; 1 pour un certificat de liaison de l'ACSN.
CRLDistributionPoints	4.2.1.14			
distributionPoint		m	x	
reasons		m	x	
cRLIssuer		m	x	
CertificatePolicies	4.2.1.5			
PolicyInformation				
policyIdentifier		m	m	
policyQualifiers		o	o	

— — — — — — — —

Appendice D à la Partie 12
(INFORMATIF)

COMPATIBILITÉ AVEC LA VALIDATION RFC 5280

Le présent appendice contient des éléments d'orientation à l'intention des États récepteurs qui souhaitent utiliser des systèmes qui appliquent les algorithmes RFC 5280 pour la validation des itinéraires de certification et des CRL.

Le modèle de confiance de l'infrastructure ICP des DVLM-e est un sous-ensemble de l'ensemble des procédures de validation définies dans la norme RFC 5280. La section D.1 décrit le sous-ensemble d'étapes de la définition RFC 5280 requis pour l'application DVLM-e et indique les entrées et les valeurs et processus d'initialisation nécessaires pour la validation de l'itinéraire de certification, la validation des CRL et la vérification des révocations.

La section D.2 porte sur les étapes de la définition RFC 5280 qui ne concernent pas l'application DVLM-e. Elle indique les entrées et les valeurs d'initialisation pour la validation de l'itinéraire de certification et la validation des CRL. Les éléments d'orientation de cette section s'appliquent dans les cas où les outils mettent en œuvre les algorithmes RFC 5280 au complet plutôt que le seul sous-ensemble décrit à la section D.1.

La section D.3 contient des éléments d'orientation pour la prise en charge de l'extension du traitement de la CRL basé sur la norme RFC 5280 pour effectuer la vérification de la révocation après un changement de nom d'une ACSN.

D.1 ÉTAPES APPLICABLES AUX DVLM-e

La procédure de validation de l'itinéraire de certification des DVLM-e définie dans la présente section est basée sur la procédure décrite dans la norme RFC 5280 ainsi que sur la terminologie et les descriptions de processus utilisées dans la norme. Les profils de certificats de DVLM-e limitent les itinéraires de certification à un seul certificat et interdisent l'utilisation de nombreux éléments optionnels employés dans d'autres applications, telles que l'ICP d'Internet définie dans la norme RFC 5280. Les étapes de validation de l'itinéraire associées à ces éléments sont omises dans la procédure de validation de l'itinéraire de certification des DVLM-e.

D.1.1 Procédure de validation de l'itinéraire de certification

D.1.1.1 Entrées

La norme RFC 5280 définit un ensemble de neuf entrées pour l'algorithme de validation d'itinéraire. Seules les trois entrées suivantes concernent l'application DVLM-e :

- l'itinéraire de certification : un seul certificat (p. ex., le certificat de signataire de document) ;
- la date/l'heure actuelles ;

- l'information d'ancre de confiance, y compris :

    - o le nom de l'émetteur fiable. Si l'ancre de confiance est sous forme de certificat de l'ACSN, le nom de l'utilisateur fiable est la valeur du champ `subject` de ce certificat ;

    - o l'algorithme de clé publique fiable. Si l'ancre de confiance est sous forme de certificat de l'ACSN, l'algorithme de clé publique fiable est tiré du champ `SubjectPublicKeyInfo` de ce certificat ;

    - o la clé publique fiable. Si l'ancre de confiance est sous forme de certificat de l'ACSN, la clé publique fiable est tirée du champ `SubjectPublicKeyInfo` de ce certificat ;

    - o les paramètres de clé publique fiable. Il s'agit d'une entrée optionnelle qui n'est incluse que si l'algorithme de clé publique fiable exige des paramètres. Si l'ancre de confiance est sous forme de certificat de l'ACSN, ces paramètres sont tirés du champ `SubjectPublicKeyInfo` de ce certificat.

La section D.2 contient les recommandations pour les implémentations qui requièrent les six entrées additionnelles.

Il pourrait y avoir plusieurs ancres de confiance pour l'ACSN qui a émis le certificat faisant l'objet de la validation. L'ancre qui DOIT être utilisée est celle qui contient la clé publique correspondant à la valeur de l'extension d'identificateur de clé d'autorité du certificat faisant l'objet de la validation.

D.1.1.2 Initialisation

La norme RFC 5280 définit onze variables d'État. Seules les cinq variables suivantes concernent l'application DVLM-e :

- application: max_path_length : initialiser à « 0 » ;
- working_issuer_name : initialiser à la valeur du nom de l'émetteur fiable ;
- working_public_key_algorithm : initialiser à la valeur de l'algorithme de clé publique fiable ;
- working_public_key : initialiser à la valeur de la clé publique fiable ;
- working_public_key_parameters : initialiser à la valeur des paramètres de clé publique fiable.

La section D.2 contient les recommandations pour les implémentations qui requièrent l'initialisation des six variables additionnelles.

D.1.1.3 Traitement de certificat

Les étapes du traitement des certificats de DVLM-e sont un sous-ensemble des étapes définies dans la norme RFC 5280. Le résultat du traitement d'un certificat de DVLM-e utilisant ce processus simplifié sera compatible avec le résultat obtenu en utilisant l'algorithme RFC 5280 au complet. Si les entrées et les variables d'État supplémentaires sont configurées comme il est décrit à la section D.2 :

a) Vérifier l'information de certificat de base. Le certificat DOIT respecter chacun des points suivants :

    - la signature du certificat peut être vérifiée à l'aide de working_public_key_algorithm, de working_public_key et de working_public_key_parameters ;

- la période de validité du certificat comprend l'heure actuelle ;
- à l'heure actuelle, le certificat n'est pas révoqué (voir le § 6.3 pour des précisions) ;
- le nom de l'émetteur du certificat est le nom de l'émetteur working_issuer_name.

b) Attribuer la valeur de `subjectPublicKey` du certificat à la variable working_public_key.

c) Si le champ `subjectPublicKeyInfo` du certificat contient un champ algorithme avec des paramètres non nuls, attribuer les paramètres à la variable working_public_key_parameters. Si le champ `subjectPublicKeyInfo` du certificat contient un champ algorithme avec des paramètres nuls ou dans lequel les paramètres ont été omis, comparer l'algorithme `subjectPublicKey` du certificat à l'algorithme de working_public_key_algorithm. Si l'algorithme de `subjectPublicKey` du certificat et l'algorithme de working_public_key_algorithm sont différents, mettre la variable working_public_key_parameters à nul.

d) Attribuer l'algorithme de `subjectPublicKey` du certificat à la variable working_public_key_algorithm.

e) Reconnaître et traiter toute autre extension critique présente dans le certificat.

f) Traiter toute autre extension non critique reconnue présente dans le certificat.

En cas d'échec d'une des étapes indiquées à l'alinéa a) ou si le certificat contient des extensions critiques non reconnues qui ne peuvent pas être traitées, la procédure de validation de l'itinéraire échoue. Autrement, la procédure s'exécute avec succès.

D.1.1.4 Sorties

Si la validation de l'itinéraire réussit, la procédure prend fin et renvoie une indication de succès avec working_public_key, working_public_key_algorithm et working_public_key_parameters.

Si la validation de l'itinéraire échoue, la procédure prend fin et renvoie une indication d'échec et un motif approprié.

D.1.2 Validation de CRL et vérification de révocation

L'algorithme REC 5280 de validation de CRL porte sur plusieurs types de CRL, notamment les CRL delta, les CRL subdivisées, les CRL indirectes, etc. Le profil de CRL pour l'application DVLM-e est très restrictif et interdit toutes ces fonctions. L'emploi de l'extension `issuingDistributionPoint` ainsi que des extensions normalisées d'entrée de CRL est aussi interdit. La validation de CRL et la vérification de révocation dans l'application DVLM-e sont donc relativement simples.

D.1.2.1 Entrées

La norme RFC 5280 définit deux entrées pour l'algorithme de validation de CRL. Seule l'entrée suivante concerne l'application DVLM-e. La section D.2 contient les recommandations pour les implémentations qui requièrent l'entrée additionnelle.

- certificate : numéro de série du certificat et nom de l'émetteur.

D.1.2.2 Initialisation

La norme RFC 5280 définit trois variables d'État. Seule la variable suivante concerne l'application DVLM-e. La section D.2 contient les recommandations pour les implémentations qui requièrent l'initialisation des deux variables additionnelles.

- cert_status : initialiser à la valeur UNREVOKED *(non révoqué)*.

D.1.2.3 Traitement de la CRL

Toutes les CRL de l'application DVLM-e sont des CRL complètes qui s'appliquent à tous les certificats en vigueur de l'ACSN qui a émis la CRL. Il n'y a pas de CRL subdivisées, delta ou indirectes. Les étapes de l'algorithme de traitement des CRL pour l'application DVLM-e sont les suivantes :

a) Obtenir la CRL en vigueur pour l'ACSN qui a émis le certificat. S'il est impossible d'obtenir la CRL, la variable cert_status est mise à UNDETERMINED *(indéterminé)*, et le traitement prend fin.

b) Vérifier que l'émetteur de la CRL est la même ACSN qui a émis le certificat en question. Comme il n'y a qu'une seule ACSN dans chaque pays, que l'application DVLM-e est une application fermée et que les systèmes d'inspection ont un cache de CRL unique pour cette application, il suffit de vérifier que le nom du pays est le même dans le champ émetteur de la CRL et le champ émetteur du certificat :

   - si l'ACSN n'a pas changé de nom depuis l'émission du certificat, le champ émetteur de la CRL et le champ émetteur du certificat seront identiques ;

   - si l'ACSN a changé de nom depuis l'émission du certificat, l'attribut pays du nom dans le champ émetteur du certificat et dans le champ émetteur de la CRL sera le même, mais certains autres attributs peuvent être différents ;

   - si la partie de confiance souhaite vérifier qu'il n'y a pas eu substitution de certaines CRL non DVLM-e, elle peut à titre facultatif vérifier qu'elle a des ancres de confiance pour les deux noms de l'ACSN et que ces ancres de confiance s'appliquent à la même ACSN. Si l'ACSN a changé de nom et a inclus l'extension optionnelle `issuerAltName` dans la CRL, la partie de confiance PEUT à titre facultatif vérifier si le champ émetteur du certificat est identique à une des valeurs de cette extension.

Si l'émetteur de la CRL n'est pas l'ACSN qui a émis le certificat, la variable cert_status est mise à UNDETERMINED *(indéterminé)* et le traitement prend fin.

c) Valider l'itinéraire de certification pour l'émetteur de la CRL. À noter que dans l'application DVLM-e toutes les CRL sont émises par des ACSN qui sont les ancres de confiance pour leurs itinéraires respectifs. Contrairement à l'algorithme RFC 5280, l'application DVLM-e N'EXIGE PAS que l'ancre de confiance utilisée pour valider l'itinéraire de certification de la CRL soit la même ancre de confiance que celle qui a été utilisée pour valider le certificat visé. Cependant, si les ancres de confiance sont différentes, elles DOIVENT toutes deux être des ancres de confiance pour la même ACSN. Contrairement à la norme RFC 5280, l'application DVLM-e comporte plusieurs ancres de confiance valides en même temps pour une même ACSN. Si l'itinéraire de certification ne peut pas être effectivement validé, la variable cert_status est mise à UNDETERMINED *(indéterminé)* et le traitement prend fin.

d) Vérifier la signature dans la CRL. Si la signature ne peut pas être effectivement vérifiée, la variable cert_status est mise à UNDETERMINED *(indéterminé)* et le traitement prend fin.

e) Rechercher le certificat dans la CRL. Si une entrée correspondant à l'émetteur du certificat et au numéro de série est trouvée, la variable cert_status est mise à UNSPECIFIED *(non spécifié)*.

D.1.2.4 Sorties

Renvoyer le statut cert_status. Si les étapes a), b), c) ou d) échouent, le statut est UNDETERMINED *(indéterminé)*. Si la CRL indique que le certificat est révoqué, le statut est UNSPECIFIED *(non spécifié)*. Si la validation de la CRL réussit mais que le certificat n'est pas compris dans la CRL, le statut est UNREVOKED *(non révoqué)*.

D.2 ÉTAPES NON REQUISES PAR L'APPLICATION DVLM-e

D.2.1 Validation de l'itinéraire de certification

Les réglages des entrées additionnelles qui ne concernent pas la validation DVLM-e sont notamment :

- initial-policy-mapping-inhibit : mettre à interdiction de mappage de politique ;
- initial-any-policy-inhibit : mettre à interdiction de traitement pour toute valeur de politique ;
- initial-permitted-subtrees : mettre à autoriser tous les sous-arbres ;
- initial-excluded-subtrees : mettre à aucune interdiction de sous-arbres ;
- initial-explicit-policy : NE DEVRAIT PAS être activée ;
- user-initial-policy-set : mettre à la valeur spéciale « any-policy » *(toute politique)*.

L'initialisation des variables d'État qui ne concernent pas l'application DVLM-e comprend notamment :

- permitted_subtrees : initialiser à autoriser tous les sous-arbres ;
- excluded_subtrees : initialiser à aucune interdiction de sous-arbres ;
- inhibit_any_policy : si la variable initial-any-policy-inhibit est positionnée, initialiser à « 0 ». Autrement, mettre la valeur à 1 ou à une autre valeur supérieure à 1 ;
- policy_mapping : initialiser à « 0 » ;
- explicit_policy : initialiser à « 2 » ;
- valid_policy_tree : initialiser l'élément valid_policy à « anyPolicy » *(toute politique)*, l'élément qualifier_set à vide et l'élément expected_policy_set à « anyPolicy ».

D.2.2 Validation de CRL

Les réglages des entrées additionnelles qui ne concernent pas la validation DVLM-e sont notamment :

- use-deltas : mettre à interdiction d'utiliser les deltas.

L'initialisation des variables d'État qui ne concernent pas l'application DVLM-e comprend notamment :

- reasons_mask : initialiser à un ensemble vide ;
- Interim_reasons_mask : initialiser à la valeur spéciale « all-reasons » *(tous les motifs)*.

D.3 MODIFICATIONS REQUISES POUR TRAITER LES CRL

Les systèmes de validation de CRL conformes à la procédure de validation de CRL de la norme RFC 5280 ne prennent pas en charge les environnements où une AC change de nom, comme c'est le cas dans l'application DVLM-e. Pour traiter ce cas spécial, ces systèmes doivent donc être modifiés comme suit :

a) Au § 6.3.3, étape a), de la procédure de validation de CRL de la norme RFC 5280, le nom figurant dans le champ point de distribution de l'extension points de distribution de CRL du certificat en question est utilisé pour actualiser le cache local et indiquer la ou les CRL pertinentes. Cette étape devrait être modifiée pour l'application DVLM-e et seul l'attribut `countryName` du champ point de distribution devrait être utilisé pour identifier et obtenir la CRL appropriée.

b) Au § 6.3.3, étape f), de la procédure de validation de CRL de la norme RFC 5280, il est spécifié qu'il faut utiliser pour valider l'itinéraire de certification de l'émetteur de CRL la même ancre de confiance qui a été utilisée pour valider le certificat visé. Cette spécification N'EST PAS exigée pour l'application DVLM-e vu que des ancres de confiance indépendantes sont établies pour chaque clé publique de l'ACSN.

L'ancre de confiance utilisée pour valider l'émetteur de CRL est celle de la clé publique de l'ACSN qui correspond à la clé privée utilisée pour signer la CRL. L'ancre de confiance utilisée pour valider l'itinéraire de certification du certificat visé peut s'appliquer à une paire de clés de l'ACSN précédente.

— — — — — — — —

Appendice E à la Partie 12
(INFORMATIF)

EXEMPLE DE SDL2

L'exemple suivant illustre les interactions entre les différents éléments de l'ICP de signature SDL2 et de l'ICP d'autorisation SDL2.

Pour illustrer les interactions et les préliminaires nécessaires à un scénario commercial typique, considérons le scénario où le pays de Dystopie veut apposer des tampons de voyage sur les passeports des citoyens du pays d'Utopie. Plus tard, le pays Atlantis veut lire les tampons de voyage écrits par Dystopie sur les passeports d'Utopie.

Les préliminaires sont les suivants :

- Utopie a installé une application de tampon de voyage SDL2 sur ses passeports.
- Dystopie et Utopie ont mis en place leur ICP d'autorisation SDL2.
- Dystopie a mis en place son ICP de signature SDL1 pour émettre des certificats de signataire SDL2.
- Les certificats CVCA et les certificats client et serveur SPOC ont été échangés de manière fiable entre Utopie et Dystopie à un moment donné (par la suite, de nouveaux certificats CVCA et SPOC peuvent être échangés directement via le SPOC).
- Les certificats CVCA et les certificats client et serveur SPOC ont été échangés de manière fiable entre Utopie et Atlantis à un moment donné (par la suite, de nouveaux certificats CVCA et SPOC peuvent être échangés directement via le SPOC). Si l'application Travel Stamp SDL2 est ouverte à la lecture, c'est-à-dire que n'importe quel pays peut lire les tampons de voyage SDL2 (la permission n'est nécessaire que pour l'écriture), cette étape peut être omise.
- Les certificats ACSN ont été échangés de manière fiable entre Dystopie et Atlantis à un moment donné.

Le processus récurrent afin de permettre à Dystopie d'estampiller électroniquement les DVLM-e d'Utopie est le suivant :

- Dystopie demande un certificat DV à Utopie.
- Le SPOC de Dystopie utilise son certificat client SPOC et le certificat serveur SPOC d'Utopie pour initier une connexion SPOC. Ensuite, une requête est générée par un DV dystopien, et envoyée de SPOC à SPOC. Sur demande, Utopie génère un certificat DV étranger avec accès en lecture/écriture pour Dystopie, et le certificat est renvoyé par SPOC à SPOC.
- Après avoir reçu le certificat DV de son SPOC, le DV de Dystopie génère des certificats de terminal pour les terminaux de ses frontières. En se connectant au passeport, le CI sur les passeports utopiens vérifie le certificat terminal de Dystopie avec le certificat DV de Dystopie, et le certificat DV de Dystopie avec le certificat CVCA d'Utopie. Le CI accorde ensuite l'accès en lecture/écriture du terminal dystopien à l'application Travel Stamp SDL2.

Le processus d'estampillage électronique d'un DVLM-e est le suivant :

- Dystopie crée un tampon de voyage électronique et le signe avec la clé privée correspondant à la clé publique stockée dans un certificat de signataire SDL2 (tampon de voyage) de l'ICP de signature SDL2 de Dystopie. Le certificat de signataire SDL2 est stocké sur le CI sans contact du passeport utopien.

Face au passeport utopien à la frontière d'Atlantis :

- Si la lecture des tampons de voyage des passeports d'Utopie nécessite un certificat de terminal avec accès en lecture, une demande de certificat d'Atlantis est envoyée par SPOC à SPOC à Utopie. Sur demande, Utopie génère un certificat DV étranger avec accès en lecture pour Atlantis et envoie ce certificat à Atlantis par SPOC à SPOC. À partir de ce certificat DV, Atlantis génère des certificats de terminal avec accès en lecture pour les passeports utopiens pour les terminaux d'Atlantis. Si les tampons de voyage des passeports utopiens peuvent être lus par n'importe quel terminal, cette étape peut être omise.

- Pour vérifier un tampon de voyage du passeport écrit par Dystopie, Atlantis utilise l'ICP de signature SDL1 de Dystopie : Le certificat de signataire SDL2 dystopien stocké dans le passeport est utilisé pour vérifier le tampon de voyage. Ensuite, la chaîne est construite, c'est-à-dire que le certificat de signataire du SDL2 de Dystopie est vérifié avec le certificat de l'ACSC de Dystopie reçu au préalable.

— FIN —

ISBN 978-92-9265-576-1