UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T X.509

(03/2000)

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

SÉRIE X: RÉSEAUX DE DONNÉES ET
COMMUNICATION ENTRE SYSTÈMES OUVERTS

Annuaire

Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: Cadre général des certificats de clé publique et d'attribut

Recommandation UIT-T X.509

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE X

RÉSEAUX DE DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

RÉSEAUX PUBLICS DE DONNÉES	
Services et fonctionnalités	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalisation et commutation	X.50–X.89
Aspects réseau	X.90–X.149
Maintenance	X.150–X.179
Dispositions administratives	X.180–X.199
INTERCONNEXION DES SYSTÈMES OUVERTS	
Modèle et notation	X.200–X.209
Définitions des services	X.210–X.219
Spécifications des protocoles en mode connexion	X.220–X.229
Spécifications des protocoles en mode sans connexion	X.230–X.239
Formulaires PICS	X.240–X.259
Identification des protocoles	X.260–X.269
Protocoles de sécurité	X.270–X.279
Objets gérés des couches	X.280–X.289
Tests de conformité	X.290–X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Généralités	X.300–X.349
Systèmes de transmission de données par satellite	X.350–X.369
Réseaux à protocole Internet	X.370–X.399
SYSTÈMES DE MESSAGERIE	X.400–X.499
ANNUAIRE	**X.500–X.599**
RÉSEAUTAGE OSI ET ASPECTS SYSTÈMES	
Réseautage	X.600–X.629
Efficacité	X.630–X.639
Qualité de service	X.640–X.649
Dénomination, adressage et enregistrement	X.650–X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680–X.699
GESTION OSI	
Cadre général et architecture de la gestion-systèmes	X.700–X.709
Service et protocole de communication de gestion	X.710–X.719
Structure de l'information de gestion	X.720–X.729
Fonctions de gestion et fonctions ODMA	X.730–X.799
SÉCURITÉ	X.800–X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850–X.859
Traitement transactionnel	X.860–X.879
Opérations distantes	X.880–X.899
TRAITEMENT RÉPARTI OUVERT	X.900–X.999

Pour plus de détails, voir la Liste des Recommandations de l'UIT-T.

NORME INTERNATIONALE 9594-8
RECOMMANDATION UIT-T X.509

TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES OUVERTS – L'ANNUAIRE: CADRE GÉNÉRAL DES CERTIFICATS DE CLÉ PUBLIQUE ET D'ATTRIBUT

Résumé

La présente Recommandation | Norme internationale définit un cadre général des certificats de clé publique et d'attribut. Ces cadres peuvent être utilisés par d'autres organismes de normalisation afin de définir leurs profils de candidature pour des infrastructures de clé publique (PKI) et des infrastructures de gestion de privilège (PMI). La présente Recommandation | Norme internationale définit également un cadre pour la fourniture de services d'authentification de l'annuaire au bénéfice de ses utilisateurs. Elle décrit deux niveaux d'authentification, l'authentification simple utilisant un mot de passe pour vérifier l'identité déclarée et l'authentification forte nécessitant des justificatifs créés au moyen de méthodes de chiffrement. L'authentification simple fournit une certaine protection contre les accès non autorisés, mais seule l'authentification forte devrait être utilisée pour fournir la base de services fiables.

Source

La Recommandation X.509 de l'UIT-T, a été approuvée le 31 mars 2000. Un texte identique est publié comme Norme internationale ISO/CEI 9594-8.

Notes

Les réalisateurs et les utilisateurs sont priés de noter qu'il existe un processus de résolution des erreurs et que des corrections peuvent être apportées à la présente Recomandation | Norme internationale sous la forme de corrigenda techniques. Des corrections identiques peuvent aussi être apportées à la présente Recommandation sous la forme d'un Guide d'implémentation. Une liste des corrigenda techniques qui ont été approuvés pour la présente Norme internationale figure sur le site web de l'ISO. Pour obtenir les corrigenda techniques qui ont été publiés, veuillez vous adresser à l'organisme de normalisation de votre pays. Les corrigenda techniques et le Guide d'implémentation pour la présente Recommandation figurent sur le site web de l'UIT.

AVANT-PROPOS

L'UIT (Union internationale des télécommunications) est une institution spécialisée des Nations Unies dans le domaine des télécommunications. L'UIT-T (Secteur de la normalisation des télécommunications) est un organe permanent de l'UIT. Il est chargé de l'étude des questions techniques, d'exploitation et de tarification, et émet à ce sujet des Recommandations en vue de la normalisation des télécommunications à l'échelle mondiale.

La Conférence mondiale de normalisation des télécommunications (CMNT), qui se réunit tous les quatre ans, détermine les thèmes d'étude à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution 1 de la CMNT.

Dans certains secteurs des technologies de l'information qui correspondent à la sphère de compétence de l'UIT-T, les normes nécessaires se préparent en collaboration avec l'ISO et la CEI.

NOTE

Dans la présente Recommandation, l'expression "Administration" est utilisée pour désigner de façon abrégée aussi bien une administration de télécommunications qu'une exploitation reconnue.

DROITS DE PROPRIÉTÉ INTELLECTUELLE

L'UIT attire l'attention sur la possibilité que l'application ou la mise en œuvre de la présente Recommandation puisse donner lieu à l'utilisation d'un droit de propriété intellectuelle. L'UIT ne prend pas position en ce qui concerne l'existence, la validité ou l'applicabilité des droits de propriété intellectuelle, qu'ils soient revendiqués par un Membre de l'UIT ou par une tierce partie étrangère à la procédure d'élaboration des Recommandations.

A la date d'approbation de la présente Recommandation, l'UIT avait été avisée de l'existence d'une propriété intellectuelle protégée par des brevets à acquérir pour mettre en œuvre la présente Recommandation. Toutefois, comme il ne s'agit peut-être pas de renseignements les plus récents, il est vivement recommandé aux responsables de la mise en œuvre de consulter la base de données des brevets du TSB.

TABLE DES MATIÈRES

Introduction

La présente Recommandation | Norme internationale, associée à d'autres Recommandations | Normes internationales, a été produite en vue de faciliter l'interconnexion de systèmes de traitement de l'information pour la fourniture de services d'annuaire. Un ensemble de tels services, associés aux informations qu'ils détiennent, peut être considéré comme une entité intégrée, appelée *annuaire*. Les informations détenues par l'annuaire, appelées collectivement base d'information d'annuaire (DIB) sont utilisées en général pour faciliter les communications s'effectuant entre, ou concernant, des objets, tels que des entités d'application, des individus, des terminaux et des listes de répartition.

L'annuaire joue un rôle important dans l'interconnexion des systèmes ouverts; moyennant un minimum d'accords techniques en dehors des normes d'interconnexion proprement dites, il a pour but de permettre l'interconnexion de systèmes de traitement de l'information:

- de fournisseurs divers;
- sous des responsabilités de gestion diverses;
- de niveaux de complexité divers;
- d'âges divers.

De nombreuses applications ont des besoins de sécurité pour se protéger contre des menaces portant sur la communication des informations. Pratiquement tous les services de sécurité font appel à la connaissance fiable des identités des participants de la communication, c'est-à-dire à leur authentification.

La présente Recommandation | Norme internationale définit un cadre pour des certificats de clé publique. Ce cadre comprend la spécification des objets de données utilisés pour représenter les certificats proprement dits ainsi que les notifications de révocation de certificats émis et auxquels il ne doit plus être fait confiance. Le cadre de certificat de clé publique décrit dans la présente Spécification définit certains composants critiques d'une infrastructure de clé publique (PKI), mais pas la totalité d'une telle infrastructure. La présente Spécification fournit toutefois une base permettant d'édifier des infrastructures PKI complètes et leurs spécifications.

La présente Recommandation | Norme internationale définit de même un cadre pour des certificats d'attribut. Ce cadre contient la spécification des objets de données utilisés pour représenter les certificats proprement dits, ainsi que les notifications de révocation de certificat émis auxquels il ne doit plus être fait confiance. Le cadre de certificat d'attribut décrit dans la présente Spécification définit certains composants critiques d'une infrastructure de gestion de privilège (PMI), mais pas la totalité d'une telle infrastructure. La présente Spécification fournit toutefois une base permettant d'édifier des infrastructures PMI complètes et leurs spécifications.

Sont définis également les objets d'informations permettant de stocker les objets d'infrastructure PKI et PMI dans l'annuaire et de comparer des valeurs présentées avec les valeurs stockées.

La présente Recommandation | Norme internationale définit également un cadre pour la fourniture de services d'authentification par l'annuaire au bénéfice de ses utilisateurs.

La présente Recommandation | Norme internationale fournit les cadres de base permettant la définition de profils industriels par d'autres organismes de normalisation et par des forums industriels. L'utilisation d'un grand nombre des fonctionnalités optionnelles figurant dans ces cadres peut être rendue obligatoire dans certains environnements au moyen de profils. Cette quatrième édition révise et étend sur le plan technique la troisième édition de la présente Recommandation | Norme internationale mais ne la remplace pas. Des implémentations peuvent continuer à déclarer la conformité avec la troisième édition. Cette dernière ne sera toutefois plus prise en charge à partir d'une certaine date (c'est-à-dire que les comptes rendus de faute ne seront plus traités). Il est recommandé que les implémentations se conforment à la présente quatrième édition, et ce dès que possible.

La présente quatrième édition spécifie la version 1 et la version 2 des protocoles d'annuaire.

Les première et deuxième éditions ne spécifiaient que la version 1. La plupart des services et des protocoles spécifiés dans la présente édition sont conçus pour fonctionner dans le cadre de la version 1. Toutefois, certains services et protocoles améliorés – les erreurs signées, par exemple – ne fonctionneront que si toutes les entités de l'annuaire qui participent à l'opération ont négocié la version 2. Quelle que soit la version qui a été négociée, les différences entre les services et entre les protocoles définis dans les quatre éditions, à l'exception de celles qui s'appliquent expressément à la version 2, sont prises en compte selon les règles d'extensibilité définies dans la présente édition de la Recommandation UIT-T X.519 | ISO/CEI 9594-5.

L'Annexe A, qui fait partie intégrante de la présente Recommandation | Norme internationale, fournit le module ASN.1 contenant toutes les définitions associées au cadre d'authentification.

L'Annexe B, qui fait partie intégrante de la présente Recommandation | Norme internationale, fournit des règles de génération et de traitement des listes de révocation de certificat.

L'Annexe C, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, fournit des exemples d'émission de liste CRL delta.

L'Annexe D, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, fournit des exemples de syntaxe de politiques de privilège et des exemples d'attribut de privilèges.

L'Annexe E, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, constitue une introduction au chiffrement avec clé publique.

L'Annexe F, qui fait partie intégrante de la présente Recommandation | Norme internationale, définit les identificateurs d'objets attribués aux algorithmes d'authentification et de chiffrement, en l'absence d'un enregistrement formel.

L'Annexe G, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, contient des exemples d'utilisation de contraintes de certification d'itinéraire.

L'Annexe H, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, contient les définitions des éléments d'information par ordre alphabétique.

L'Annexe I, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, fournit la liste des amendements et des comptes rendus d'erreur qui ont été incorporés dans cette édition de la présente Recommandation | Norme internationale.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES OUVERTS – L'ANNUAIRE: CADRE GÉNÉRAL DES CERTIFICATS DE CLÉ PUBLIQUE ET D'ATTRIBUT

SECTION 1 – GÉNÉRALITÉS

1 Domaine d'application

La présente Recommandation | Norme internationale traite de certains besoins de sécurité dans les domaines de l'authentification et d'autres services de sécurité, en fournissant un ensemble de cadres sur la base desquels il est possible d'édifier des services complets. La présente Recommandation | Norme internationale définit de manière plus spécifique les cadres suivants:

- certificats de clé publique;
- certificats d'attribut;
- services d'authentification.

Le cadre de certificat de clé publique défini dans la présente Recommandation | Norme internationale englobe la définition des objets d'information pour une infrastructure de clé publique (PKI, *public key infrastructure*), incluant les certificats de clé publique et les listes de révocation de certificat (CRL, *certificate revocation list*). Le cadre de certificat d'attribut englobe la définition des objets d'information pour une infrastructure de gestion de privilège (PMI, *privilege management infrastructure*), incluant les certificats d'attribut et la liste de révocation de certificat d'attribut (ACRL, *attribute certificate revocation list*). La présente Spécification fournit également le cadre pour l'émission, la gestion, l'utilisation et la révocation de certificats. Les formats définis pour les deux types de certificats et pour tous les types de liste de révocation prévoient un procédé d'extension. La présente Recommandation | Norme internationale contient également un ensemble d'extensions normalisées pour chaque type; il est prévu que cet ensemble sera d'une utilité générale pour un certain nombre d'infrastructures PKI et PMI. Les composants du schéma, englobant les classes d'objets, les types d'attribut et les règles de concordance pour le stockage des objets PKI et PMI dans l'annuaire font partie de la présente Recommandation | Norme internationale. Il est prévu que d'autres organismes de normalisation (par exemple le comité TC 68 de l'ISO, l'IETF, etc.) définiront des éléments d'infrastructure PKI et PMI supplémentaires qui sortent de ces cadres, tels que les protocoles de gestion de clé et de certificat, les protocoles opérationnels ou d'autres certificats et extensions de liste CRL.

Le procédé d'authentification défini dans la présente Recommandation | Norme internationale possède un caractère générique et peut s'appliquer à une variété d'applications et d'environnements.

L'annuaire utilise les certificats de clé publique et les certificats d'attribut; le cadre d'utilisation de ces fonctionnalités par l'annuaire est également défini dans la présente Recommandation | Norme internationale. L'annuaire utilise une technologie de clé publique avec certificats pour fournir une authentification forte et des opérations avec signature et/ou chiffrement, ainsi que pour stocker des données signées et/ou chiffrées. Il peut utiliser des certificats d'attribut pour fournir un contrôle d'accès basé sur des règles. Bien que le cadre correspondant soit fourni dans la présente Spécification, la définition complète de l'utilisation de l'annuaire, des services associés qu'il fournit et de ses composants font l'objet d'une définition dans un ensemble complet de spécifications de l'annuaire.

La présente Recommandation | Norme internationale précise également les points suivants dans le cadre des services d'authentification:

- spécification du format des informations d'authentification contenues dans l'annuaire;
- description de la manière dont les informations d'authentification peuvent être obtenues à partir de l'annuaire;
- énoncé des hypothèses faites sur la manière dont les informations d'authentification sont créées et placées dans l'annuaire;
- définition de trois modes d'utilisation possibles des informations d'authentification par des applications en vue d'effectuer l'authentification et la description de la manière dont d'autres services de sécurité peuvent être pris en charge par une authentification.

La présente Recommandation | Norme internationale décrit deux niveaux d'authentification: l'authentification simple utilisant un mot de passe pour vérifier l'identité déclarée et l'authentification forte nécessitant des justificatifs créés au moyen de méthodes de chiffrement. L'authentification simple fournit une certaine protection contre les accès non autorisés, mais seule l'authentification forte devrait être utilisée pour fournir la base de services fiables. Elle n'est pas conçue pour établir de ce fait un cadre d'authentification, mais peut être utilisée d'une manière générale pour des applications qui considèrent ces procédés comme adéquats.

L'authentification (comme d'autres services de sécurité) peut uniquement être fournie dans le contexte de la définition d'une politique de sécurité. Les utilisateurs d'une application ont la charge de définir leur propre politique de sécurité, pouvant être soumise aux contraintes des services fournis dans le cadre d'une norme.

Les normes de définition d'applications utilisant le cadre d'authentification ont la charge de spécifier les échanges de protocole nécessaires pour réaliser une authentification basée sur les informations d'authentification obtenues à partir de l'annuaire. Le protocole d'accès à l'annuaire (DAP, *directory access protocol*) utilisé par les applications pour obtenir des justificatifs à partir de l'annuaire est spécifié dans la Rec. UIT-T X.519 | ISO/CEI 9594-5.

2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui est faite, constituent des dispositions valables pour la présente Recommandation | Norme internationale. Au moment de la publication, les éditions indiquées étaient en vigueur. Toutes Recommandations et Normes sont sujettes à révision et les parties prenantes aux accords fondés sur la présente Recommandation sont invitées à rechercher la possibilité d'appliquer les éditions les plus récentes des Recommandations et Normes indiquées ci-après. Les membres de la CEI et de l'ISO possèdent le registre des Normes internationales en vigueur. Le Bureau de la normalisation des télécommunications de l'UIT-T tient à jour une liste des Recommandations de l'UIT-T en vigueur.

2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.411 (1999) | ISO/CEI 10021-4:1999, *Technologies de l'information – Systèmes de messagerie: système de transfert de messages: définition et procédures du service abstrait*.
- Recommandation UIT-T X.500 (2001) | ISO/CEI 9594-1:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: aperçu général des concepts, modèles et services*[1)].
- Recommandation UIT-T X.501 (2001) | ISO/CEI 9594-2:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: les modèles*.
- Recommandation UIT-T X.511 (2001) | ISO/CEI 9594-3:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: définition du service abstrait*.
- Recommandation UIT-T X.518 (2001) | ISO/CEI 9594-4:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: procédures pour le fonctionnement réparti*.
- Recommandation UIT-T X.519 (2001) | ISO/CEI 9594-5:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: spécification du protocole*.
- Recommandation UIT-T X.520 (2001) | ISO/CEI 9594-6:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: types d'attributs sélectionnés*.
- Recommandation UIT-T X.521 (2001) | ISO/CEI 9594-7:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: classes d'objets sélectionnées*.
- Recommandation UIT-T X.525 (2001) | ISO/CEI 9594-9:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: duplication*.
- Recommandation UIT-T X.530 (2001) | ISO/CEI 9594-10:2001, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: utilisation de la gestion-systèmes pour l'administration de l'annuaire*.
- Recommandation CCITT X.660 (1992) | ISO/CEI 9834-1:1993, *Technologies de l'information – Interconnexion des systèmes ouverts – procédures pour le fonctionnement des autorités d'enregistrement OSI: Procédures générales*.

---

1) Pour chacune des Recommandations de la série X.500 | parties 9594 référencées dans cet article, il convient d'utiliser la quatrième édition de ces spécifications dès qu'elle sera publiée.

- Recommandation UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.
- Recommandation UIT-T X.681 (1997) | ISO/CEI 8824-2:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels*.
- Recommandation UIT-T X.682 (1997) | ISO/CEI 8824-3:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes*.
- Recommandation UIT-T X.683 (1997) | ISO/CEI 8824-4:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: paramétrage des spécifications de la notation de syntaxe abstraite numéro un*.
- Recommandation UIT-T X.690 (1997) | ISO/CEI 8825-1:1998, *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*.
- Recommandation UIT-T X.691 (1997) ISO/CEI 8825-2:1998, *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage compact*.
- Recommandation UIT-T X.812 (1995) ISO/CEI 10181-3:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Cadres de sécurité pour les systèmes ouverts: cadre de contrôle d'accès*.
- Recommandation UIT-T X.813 (1996) | ISO/CEI 10181-4:1997, *Technologies de l'information – Interconnexion des systèmes ouverts – Cadres de sécurité pour les systèmes ouverts: non-répudiation*.
- Recommandation UIT-T X.880 (1994) | ISO/CEI 13712-1:1995, *Technologies de l'information – Opérations distantes: concepts, modèle et notation*.
- Recommandation UIT-T X.881 (1994) | ISO/CEI 13712-2:1995, *Technologies de l'information – Opérations distantes: réalisations OSI – Définition du service de l'élément de service d'opérations distantes*.

2.2 Paires de Recommandations | Normes internationales équivalentes par leur contenu technique

- Recommandation CCITT X.800 (1991), *Architecture de sécurité pour l'interconnexion en systèmes ouverts d'applications du CCITT*.

  ISO 7498-2:1989, *Systèmes de traitement de l'information – Interconnexion de systèmes ouverts – Modèle de référence de base – Partie 2: Architecture de sécurité*.

3 Définitions

Pour les besoins de la présente Recommandation UIT-T | Norme internationale, les définitions suivantes s'appliquent.

3.1 Définitions relatives à l'architecture de sécurité du modèle de référence OSI

Les termes suivants sont définis dans la Rec. CCITT X.800 | ISO 7498-2:

a) asymétrique (chiffrement);

b) échange d'authentifications;

c) information d'authentification;

d) confidentialité;

e) justificatifs (ou habilitation);

f) cryptographie;

g) authentification de l'origine des données;

h) déchiffrement;

i) chiffrement;

j) clé;

k) mot de passe;

l) authentification de l'entité homologue;

m) symétrique (chiffrement).

3.2 Définitions relatives au modèle d'annuaire

Les termes suivants sont définis dans la Rec. UIT-T X.501 | ISO/CEI 9594-2:

a) attribut;
b) base d'informations d'annuaire;
c) arbre d'informations d'annuaire;
d) agent de système d'annuaire;
e) agent d'utilisateur d'annuaire;
f) nom distinctif;
g) entrée;
h) objet;
i) racine.

3.3 Définitions

Pour les besoins de la présente Recommandation | Norme internationale les termes suivants sont définis:

3.3.1 certificat d'attribut: structure de donnée, portant la signature numérique d'une autorité d'attribut, qui lie certaines valeurs d'attribut à des informations d'identification concernant son détenteur.

3.3.2 autorité d'attribut (AA): autorité qui attribue des privilèges par l'émission de certificats d'attribut.

3.3.3 liste de révocation d'autorité d'attribut (AARL, *attribute authority revocation list*): liste de révocation contenant une liste de références de certificats d'attribut concernant des autorités d'attribut qui ne sont plus considérées comme valides par l'autorité émettrice.

3.3.4 liste de révocation de certificat d'attribut (ACRL, *attribute certificate revocation list*): liste de révocation contenant une liste de références de certificats d'attribut qui ne sont plus considérés comme valides par l'autorité émettrice.

3.3.5 jeton d'authentification; jeton: information véhiculée pendant un échange d'authentification forte et pouvant être utilisée pour authentifier son émetteur.

3.3.6 autorité: entité responsable de l'émission de certificats. La présente Spécification définit les deux types suivants: les autorités de certification émettant des certificats de clé publique et les autorités d'attribut émettant des certificats d'attribut.

3.3.7 certificat d'autorité: certificat émis à destination d'une autorité (par exemple, une autorité de certification ou une autorité d'attribut).

3.3.8 liste CRL de base: liste CRL utilisée comme base pour la création d'une liste dCRL.

3.3.9 certificat d'autorité de certification: certificat émis par une autorité de certification pour une autre autorité de certification.

3.3.10 politique de certificat: ensemble nommé de règles indiquant la possibilité d'appliquer un certificat pour une communauté particulière et/ou une classe d'applications particulière avec des besoins de sécurité communs. Une politique de certificat particulière peut, par exemple, indiquer la possibilité d'application d'un certificat pour des transactions avec échange de données électroniques pour le commerce de biens dans une fourchette de prix donnée.

3.3.11 liste de révocation de certificat (CRL, *certificate revocation list*): liste signée indiquant un ensemble de certificats qui ne sont plus considérés comme valides par leur émetteur. Certains types de listes CRL spécifiques sont définis en plus du type générique de liste CRL, pour couvrir des domaines particuliers.

3.3.12 utilisateur de certificat: entité qui a besoin de connaître avec certitude la clé publique d'une autre entité.

3.3.13 numéro de série de certificat: valeur entière, non ambiguë pour l'autorité émettrice, qui est associée de manière biunivoque à un certificat émis par cette autorité de certification.

3.3.14 système utilisant des certificats: implémentation de celles des fonctions définies dans la présente Spécification d'annuaire qui sont mises en œuvre par un utilisateur de certificat.

3.3.15 validation de certificat: processus consistant à s'assurer qu'un certificat était valide à un instant donné, impliquant éventuellement la construction et le traitement d'un itinéraire de certification avec la garantie que tous les certificats de l'itinéraire étaient valides (c'est-à-dire, non caducs ou révoqués) à l'instant donné.

3.3.16 autorité de certification (CA, *certification authority*): autorité jouissant de la confiance d'un ou de plusieurs utilisateurs pour la création et l'attribution de certificats. L'autorité de certification peut, de manière optionnelle, créer les clés des utilisateurs.

3.3.17 liste de révocation d'autorité de certification (CARL, *certification authority revocation list*): liste de révocation contenant une liste de certificats de clé publique émise pour des autorités de certification qui ne sont plus considérées comme valides par l'émetteur du certificat.

3.3.18 itinéraire de certification: séquence ordonnée de certificats concernant des objets contenus dans l'arbre DIT et qui peuvent être traités à partir de la clé publique de l'objet initial de l'itinéraire pour obtenir l'objet final de cet itinéraire.

3.3.19 point de répartition de liste CRL: élément de dictionnaire ou autre source de distribution de listes CRL; une telle liste distribuée par le biais d'un point de répartition de liste CRL peut contenir des éléments révoquant uniquement un sous-ensemble de la totalité des certificats émis par une autorité de certification ou peut contenir des éléments révoquant plusieurs autorités de certification.

3.3.20 système de chiffrement: ensemble de transformations d'un texte en clair pour obtenir un texte chiffré et réciproquement, le choix de la ou des transformations particulières à utiliser se faisant au moyen de clés. Les transformations sont définies en général par un algorithme mathématique.

3.3.21 confidentialité des données: ce service peut être utilisé pour protéger des données contre une divulgation non autorisée. Le service de confidentialité des données est pris en charge par le cadre d'authentification. Il peut être utilisé pour protéger des données contre les interceptions.

3.3.22 délégation: transfert d'un privilège d'une entité détentrice vers une autre entité.

3.3.23 itinéraire de délégation: séquence ordonnée de certificats qui peuvent, conjointement à l'authentification de l'identité du déclarant, être traités pour vérifier l'authenticité d'un privilège de ce déclarant.

3.3.24 liste CRL delta (liste dCRL): liste de révocation partielle contenant uniquement des éléments pour des certificats dont le statut de révocation a été modifié depuis la publication de la liste CRL de base référencée.

3.3.25 entité finale: sujet d'un certificat qui utilise sa clé privée à d'autres fins que la signature de certificats ou entité qui est un participant faisant confiance.

3.3.26 liste de révocation de certificat d'attribut d'entité finale (EARL, *end-entity attribute certificate revocation list*): liste de révocation contenant une liste de certificats d'attribut émis à destination de détenteurs, qui ne sont pas également des autorités d'attribut et qui ne sont plus considérés comme valides par l'émetteur du certificat.

3.3.27 liste de révocation de certificat de clé publique (EPRL, *end-entity public-key certificate revocation list*): liste de révocation contenant une liste de certificats de clé publique, émise à destination de sujets qui ne sont pas également des autorités de certification, et qui ne sont plus considérés comme valides par l'émetteur du certificat.

3.3.28 variables d'environnement: caractéristiques d'une politique nécessaires pour une décision d'autorisation, qui ne sont pas contenues dans des structures statiques mais qui sont accessibles localement par un vérificateur de privilège (par exemple, le jour et l'heure ou le solde actuel d'un compte).

3.3.29 liste CRL complète: liste de révocation complète contenant des éléments pour tous les certificats qui ont été révoqués pour le domaine d'application donné.

3.3.30 fonction de hachage: fonction (mathématique) qui fait correspondre un argument pris dans un domaine étendu (éventuellement très étendu) à une valeur appartenant à un domaine plus réduit. Une "bonne" fonction de hachage est telle que l'application de la fonction à un ensemble (étendu) d'arguments du premier domaine fournira des valeurs réparties de manière égale (apparemment aléatoire) dans le second domaine.

3.3.31 détenteur: entité qui a reçu la délégation d'un privilège, soit directement de la source d'autorité, soit indirectement par le biais d'une autre autorité d'attribut.

3.3.32 liste CRL indirecte (iCRL, *indirect CRL*): liste de révocation qui contient au moins une information de révocation concernant des certificats émis par des autorités autres que l'émetteur de cette liste.

3.3.33 agrément de clé: méthode de négociation en ligne de la valeur d'une clé sans transfert de cette dernière, même sous forme chiffrée, par exemple en utilisant la méthode Diffie-Hellman (se référer à ISO/CEI 11770-1 pour plus d'informations concernant les procédés d'agrément de clé).

3.3.34 méthode d'objet: action pouvant être invoquée pour une ressource (par exemple, un système de fichier peut disposer de méthodes objet de lecture, d'écriture et d'exécution).

3.3.35 fonction non réversible: fonction mathématique facile à calculer, mais qui, pour une valeur quelconque y du domaine image, il est difficile de trouver une valeur x du domaine source telle que $f(x) = y$. Il peut exister un nombre réduit de valeurs de y pour lesquelles le calcul de x est trivial.

3.3.36 mappage de politique: reconnaissance du fait que, lorsqu'une autorité de certification d'un domaine certifie une autorité de certification d'un autre domaine, une politique de certificat propre au deuxième domaine peut être considérée par l'autorité du premier domaine comme équivalente (mais pas nécessairement comme identique sous tous ses aspects) à une politique de certificat dans le premier domaine.

3.3.37 clé privée; clé secrète (déconseillé): (dans un système de chiffrement avec clé publique) celle des clés d'une paire de clés d'un utilisateur qui est connue uniquement par l'utilisateur concerné.

3.3.38 privilège: attribut ou propriété attribué par une autorité à un utilisateur.

3.3.39 déclarant de privilège: détenteur de privilège utilisant son certificat d'attribut de clé publique pour déclarer un privilège.

3.3.40 infrastructure de gestion de privilège (PMI, *privilege management infrastructure*): infrastructure qui peut prendre en charge la gestion des privilèges correspondant à un service complet d'autorisation et en relation avec une infrastructure de clé publique.

3.3.41 politique de privilège: politique qui définit dans ses grandes lignes les conditions sous lesquelles les vérificateurs de privilège peuvent fournir ou effectuer des services sensibles au profit ou pour le compte de déclarants de privilège qualifiés. La politique de privilège est liée à des attributs associés au service, ainsi qu'à des attributs associés aux déclarants de privilège.

3.3.42 vérificateur de privilège: entité effectuant la vérification de certificats conformément à une politique de privilège.

3.3.43 clé publique: (dans un système de chiffrement avec clé publique) celle des clés d'une paire de clés d'un utilisateur qui est connue de manière publique.

3.3.44 certificat de clé publique: clé publique d'un utilisateur, associée à certaines autres informations qui sont rendues non falsifiables par chiffrement en utilisant la clé privée de l'autorité de certification émettrice.

3.3.45 infrastructure de clé publique (PKI, *public key infrastructure*): infrastructure pouvant prendre en charge la gestion de clés publiques afin de fournir des services d'authentification, de chiffrement, d'intégrité et de non répudiation.

3.3.46 participant faisant confiance: utilisateur ou agent qui fait confiance aux données contenues dans un certificat pour prendre des décisions.

3.3.47 certificat d'attribution de rôle: certificat contenant l'attribut de rôle qui assigne un ou plusieurs rôles au sujet/au détenteur du certificat.

3.3.48 certificat de spécification de rôle: certificat contenant l'attribution de privilèges à un rôle.

3.3.49 sensibilité: caractéristique d'une ressource liée à sa valeur ou à son importance.

3.3.50 authentification simple: authentification utilisant de simples accords de mot de passe.

3.3.51 politique de sécurité: ensemble de règles fixées par l'autorité de sécurité qui régit l'utilisation et la fourniture de services et de fonctionnalités de sécurité.

3.3.52 source d'autorité (SOA, *source of authority*): autorité d'attribut auquel peut faire confiance un vérificateur de privilège pour une ressource donnée, en tant qu'autorité ultime pour l'attribution d'un ensemble de privilèges.

3.3.53 authentification forte: authentification utilisant des justificatifs obtenus par des moyens de chiffrement.

3.3.54 confiance: on peut dire d'une manière générale qu'une entité "fait confiance" à une autre entité si la première fait l'hypothèse que la deuxième se comportera exactement comme attendu (par la première). Il se peut que cette confiance s'applique uniquement pour une fonction donnée. Le rôle clé de la confiance dans ce cadre décrit la relation entre une entité effectuant l'authentification et une autorité; une entité sera certaine qu'elle peut faire confiance à l'autorité pour ne créer que des certificats valides et fiables.

4 Abréviations

Pour les besoins de la présente Recommandation UIT-T | Norme internationale, les abréviations suivantes sont utilisées.

AA Autorité d'attribut

AARL Liste de révocation d'autorité d'attribut *(attribute authority revocation list)*

ACRL Liste de révocation de certificat d'attribut *(attribute certificate revocation list)*

CA Autorité de certification *(certification authority)*

CARL Liste de révocation d'autorité de certification *(certification authority revocation list)*

CRL Liste de révocation de certificat *(certificate revocation list)*

dCRL Liste delta de révocation de certificat *(delta certificate revocation list)*

DIB Base d'informations d'annuaire *(directory information base)*

DIT Arbre d'informations d'annuaire *(directory information tree)*

DSA Agent de système d'annuaire *(directory system agent)*

DUA Agent utilisateur d'annuaire *(directory user agent)*

EARL Liste de révocation de certificat d'attribut d'entité finale *(end-entity attribute certificate revocation list)*

EPRL Liste de révocation de certificat de clé publique d'entité finale *(end-entity public-key certificate revocation list)*

iCRL Liste indirecte de révocation de certificat *(indirect certificate revocation list)*

PKCS Système de chiffrement avec clé publique *(public-key cryptosystem)*

PKI Infrastructure de clé publique *(public-key infrastructure)*

PMI Infrastructure de gestion de privilège *(privilege management infrastructure)*

SOA Source d'autorité *(source of authority)*

5 Conventions

La présente Spécification d'annuaire a été préparée, avec des exceptions mineures, conformément aux directives "Présentation de texte commun UIT-T/ISO/CEI" figurant dans le guide de coopération entre l'UIT-T et le comité JTC 1 de l'ISO/CEI en date de mars 1996.

Le terme "Spécification d'annuaire" (comme dans "la présente Spécification d'annuaire") doit être considéré comme indiquant la Rec. UIT-T X.509 | ISO/CEI 9594-8. Le terme "Spécifications d'annuaire" doit être considéré comme indiquant les Recommandations de la série X.500 et la totalité des parties de l'ISO/CEI 9594.

La présente Spécification d'annuaire utilise le terme "systèmes de l'édition 1988" pour faire référence à la première édition des Spécifications d'annuaire, c'est-à-dire l'édition 1988 des Recommandations X.500 du CCITT et de l'ISO/CEI 9594:1990. La présente Spécification d'annuaire utilise le terme "systèmes de l'édition 1993" pour faire référence à la deuxième édition des Spécifications d'annuaire, c'est-à-dire l'édition 1993 des Recommandations UIT-T X.500 et de l'ISO/CEI 9594:1995. La présente Spécification d'annuaire utilise le terme "systèmes de l'édition 1997" pour faire référence à la troisième édition des Spécifications d'annuaire, c'est-à-dire l'édition 1997 des Recommandations UIT-T X.500 et de l'ISO/CEI 9594:1998. La présente Spécification d'annuaire utilise le terme "systèmes de la quatrième édition des Spécifications d'annuaire, c'est-à-dire l'édition 2001 des Recommandations UIT-T X.500, K.501, X.511, X.518, X.519, X.520, X.521, X.525 et X.530, l'édition 2000 de la Recommandation UIT-T X.509 et les parties 1 à 10 de l'ISO/CEI 9594:2001.

La présente Spécification d'annuaire représente la notation ASN.1 en utilisant des caractères Helvetica gras. Lorsque des types et des valeurs ASN.1 font l'objet d'une référence dans un texte normal, ils sont mis en relief par l'utilisation de caractères Helvetica gras. Les noms de procédures, qui font en général l'objet d'une référence lors de la spécification de la sémantique de traitement se distinguent du texte normal par leur présentation en caractères Times gras. Les permissions de commande d'accès sont représentées en caractères Times italique.

Si les éléments de la liste sont numérotés (au lieu de l'utilisation du caractère "–" ou de lettres), ils seront alors considérés comme des étapes d'une procédure.

Le Tableau 1 ci-dessous définit la notation utilisée dans la présente Spécification d'annuaire.

Tableau 1 – Notation

Notation	Signification
Xp	Clé publique d'un utilisateur X.
Xs	Clé privée de X.
Xp[I]	Chiffrement des informations I avec utilisation de la clé publique de X.
Xs[I]	Chiffrement des informations I avec utilisation de la clé privée de X.
X{I}	Signature des informations I par l'utilisateur X. Elle se constitue de I avec l'ajout d'un résumé chiffré.
CA(X)	Autorité de certification de l'utilisateur X.
$CA^n(X)$	(Avec n>1): CA(CA(...n fois...(X))).
$X_1«X_2»$	Certificat de l'utilisateur X_2 émis par l'autorité de certification X_1.
$X_1«X_2» X_2«X_3»$	Chaîne de certificats (de longueur quelconque) dont chaque élément est le certificat pour l'autorité de certification qui a produit le suivant. Elle est fonctionnellement équivalente au certificat suivant $X_1«X_{n+1}»$. La possession de la chaîne de certificats A«B»B«C» fournit les mêmes capacités que A«C», à savoir la capacité de trouver Cp connaissant Ap.
$X_1p \circ X_1«X_2»$	Opération d'ouverture d'un certificat (ou d'une chaîne de certificats) pour l'extraction d'une clé publique. Il s'agit d'un opérateur infixe dont l'opérande de droite est la clé publique d'une autorité de certification et l'opérande de gauche un certificat émis par cette autorité. Le résultat fournit la clé publique de l'utilisateur dont le certificat est l'opérande de droite. Par exemple: Ap ∘ A«B» B«C» indique l'opération d'utilisation de la clé publique de A permettant d'obtenir la clé publique Bp de B à partir de son certificat, suivie de l'utilisation de Bp pour ouvrir le certificat de C. Le résultat de cette opération fournit la clé publique Cp de l'utilisateur C.
A→B	Itinéraire de certification de A vers B, formé d'une chaîne de certificats débutant avec CA(A) «CA^2(A)» et se terminant par CA(B) «B».
NOTE – Les symboles X, X_1, X_2, etc. du tableau représentent les noms des utilisateurs et le symbole I représente une information quelconque.	

6 Apercu général des cadres

La présente Spécification définit un cadre permettant d'obtenir la clé publique d'une entité et de lui faire confiance pour le déchiffrement d'informations chiffrées par cette entité ou pour la vérification de la signature numérique de cette entité. Le cadre englobe l'émission d'un certificat de clé publique par une autorité de certification (CA) et la validation de ce certificat par l'utilisateur de certificat. La validation se constitue des opérations suivantes:

- établissement d'un itinéraire fiable constitué de certificats entre l'utilisateur de certificat et le sujet du certificat;
- vérification des signatures numériques de chaque certificat de l'itinéraire;
- validation de tous les certificats de l'itinéraire (c'est-à-dire, vérifier s'ils ne sont pas caducs ou n'ont pas été révoqués à une date donnée).

La présente Spécification définit un cadre permettant d'obtenir et de faire confiance à des attributs de privilège d'une entité afin de déterminer si l'accès de cette dernière à une ressource donnée est autorisé. Le cadre englobe l'émission d'un certificat par une autorité d'attribut (AA) et la validation de ce certificat par un vérificateur de privilège. La validation se constitue des opérations suivantes:

- s'assurer que les privilèges du certificat sont suffisants, compte tenu de la politique de privilège;
- établissement, si nécessaire, d'un itinéraire fiable de délégation de certificats;
- vérification de la signature numérique de chaque certificat de l'itinéraire;
- s'assurer que chaque émetteur était autorisé à déléguer des privilèges;
- vérifier que les certificats ne sont pas caducs ou n'ont pas été révoqués par leur émetteur.

Bien que les infrastructures PKI et PMI soient distinctes et puissent être mises en place de manière indépendante, elles sont toutefois en relation. La présente Spécification recommande que les détenteurs et émetteurs de certificats d'attribut soient identifiés au sein des certificats d'attribut par des pointeurs vers leurs certificats de clé publique adéquats. L'authentification des émetteurs et détenteurs de certificat d'attribut, nécessaire pour garantir que les entités qui déclarent un privilège et émettent un privilège sont bien celles qu'elles affirment être, est faite au moyen du processus normal de l'infrastructure PKI permettant d'authentifier des identités. Ce processus d'authentification n'est pas dupliqué dans le cadre du certificat d'attribut.

6.1 Signatures numériques

Les signatures numériques sont utilisées dans les infrastructures PKI et PMI par l'autorité émettrice d'un certificat comme méthode de certification de la liaison figurant dans le certificat. Dans le cas de l'infrastructure PKI, la signature numérique de l'autorité de certification émettrice sur un certificat de clé publique certifie la liaison entre les informations de clé publique et le sujet du certificat. Dans le cas de l'infrastructure PMI la signature numérique de l'autorité d'attribut émettrice certifie la liaison entre les attributs (privilèges) et le détenteur du certificat. Ce paragraphe décrit les signatures numériques d'une manière générale. Les sections 2 et 3 de la présente Spécification analysent d'une manière spécifique l'utilisation de signatures numériques dans les infrastructures PKI et PMI.

Ce paragraphe n'a pas l'intention de spécifier d'une manière générale une norme pour les signatures numériques, mais plutôt de spécifier les moyens permettant de signer les jetons dans les infrastructures PKI et PMI ainsi que dans l'annuaire.

Les informations (info) sont signées par l'ajout d'un résumé chiffré des informations. Le résumé est produit par une fonction de hachage non réversible et le chiffrement est effectué au moyen de la clé privée du signataire voir Figure 1). Il en résulte que:

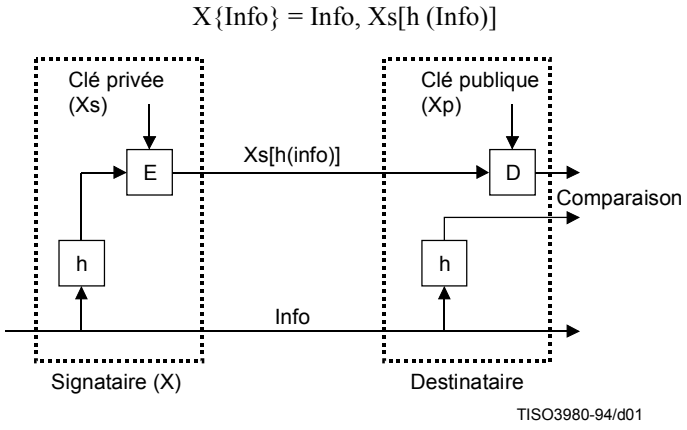


Figure 1 – Signatures numériques

NOTE 1 – Le chiffrement avec la clé privée garantit que la signature ne peut pas être falsifiée. La nature non réversible de la fonction de hachage garantit qu'il n'est pas possible de substituer une information falsifiée générée dans le but de fournir un résultat de hachage identique (et donc, de contrefaire la signature).

Le destinataire des informations signées vérifie la signature de la manière suivante:

- application de la fonction de hachage non réversible aux informations;
- comparaison du résultat avec celui qui est obtenu par déchiffrement de la signature au moyen de la clé publique du signataire.

La présente Spécification n'impose pas l'utilisation d'une seule fonction de hachage non réversible pour la signature. Elle est conçue de manière à ce que le cadre s'applique à toute fonction de hachage convenable et prendra de ce fait en charge les modifications apportées aux méthodes utilisées en fonction des progrès futurs du chiffrement, des techniques mathématiques et des capacités de calcul. Toutefois, deux utilisateurs souhaitant s'authentifier prendront en charge la même fonction de hachage pour effectuer correctement l'authentification. Il s'ensuit que, dans le contexte d'un ensemble d'applications en relation, le choix d'une fonction unique permettra de maximiser la taille de la communauté des utilisateurs susceptibles de s'authentifier et de communiquer d'une manière fiable.

Les informations signées contiennent des indicateurs identifiant l'algorithme de hachage et l'algorithme de chiffrement utilisés pour le calcul de la signature numérique.

Le chiffrement d'un certain élément de données peut être décrit par le code ASN.1 suivant:

```
ENCRYPTED { ToBeEnciphered } ::=   BIT STRING ( CONSTRAINED BY {
-- doit être le résultat de l'application d'une procédure de chiffrement --
-- aux octets du codage BER d'une valeur de -- ToBeEnciphered } )
```

La valeur de la chaîne de bits est générée en prenant les octets qui forment le codage complet (utilisant les règles de codage ASN.1 de base de la Rec. UIT-T X.690 (1997) | ISO/CEI 8825-1:1998) de la valeur du type **ToBeEnciphered** (*à chiffrer*) et en leur appliquant la procédure de chiffrement.

NOTE 2 – La procédure de chiffrement nécessite un accord au sujet de l'algorithme à appliquer, y compris pour tous les paramètres de l'algorithme, tels que l'ensemble des clés nécessaires, les valeurs d'initialisation et les instructions de remplissage. Les procédures de chiffrement sont responsables de la spécification des moyens utilisés pour la synchronisation entre l'émetteur et le récepteur des données, pouvant inclure des informations contenues dans les bits à transmettre.

NOTE 3 – La procédure de chiffrement doit accepter en entrée une chaîne d'octets et générer comme résultat une seule chaîne de bits.

NOTE 4 – Les méthodes utilisées pour obtenir un accord fiable entre l'émetteur et le récepteur des données au sujet de l'algorithme de chiffrement et de ses paramètres sont en dehors du domaine d'application de la présente Spécification d'annuaire.

La signature d'un certain élément de données s'effectue par le chiffrement d'une transformée raccourcie ou "hachée" de cet élément et peut être décrite par le code ASN.1 suivant:

```
HASH {ToBeHashed}          ::=   SEQUENCE {
    algorithmIdentifier          AlgorithmIdentifier,
    hashValue                    BIT STRING ( CONSTRAINED BY {
        -- doit être le résultat de l'application d'une procédure de hachage aux octets du codage DER --
        -- d'une valeur de -- ToBeHashed } ) }

ENCRYPTED-HASH { ToBeSigned }    ::=          BIT STRING ( CONSTRAINED BY {
    -- doit être le résultat de l'application d'une procédure de hachage aux octets du codage DER --
    -- d'une valeur de -- ToBeSigned -- et en appliquant ensuite une procédure de chiffrement à ces octets -- })

SIGNATURE { ToBeSigned }         ::=          SEQUENCE {
    algorithmIdentifier          AlgorithmIdentifier,
    encrypted                    ENCRYPTED-HASH { ToBeSigned }}
```

NOTE 5 – La procédure de chiffrement nécessite les accords indiqués dans la Note 2 ainsi qu'un accord sur le fait que les octets hachés sont chiffrés directement ou après leur codage sous la forme d'une chaîne de bits au moyen des règles de codage ASN.1 de base.

Le code ASN.1 suivant peut être utilisé pour définir le type des données résultant de l'application d'une signature au type de données en question, dans le cas où une signature doit être ajoutée à un type de données.

```
SIGNED { ToBeSigned }                    ::=          SEQUENCE {
    toBeSigned                   ToBeSigned,
    COMPONENTS OF                SIGNATURE { ToBeSigned }}
```

Un codage distinctif est nécessaire pour la validation des types **SIGNED** et **SIGNATURE** (*signé et signature*) dans un environnement réparti. Un codage distinctif des valeurs de données de type **SIGNED** ou **SIGNATURE** sera obtenu en appliquant les règles de codage de base définies dans la Rec. UIT X.690 (1997) | ISO/CEI 8825:1998, avec les limitations suivantes:

a) la forme de codage précise de la longueur sera utilisée, codée dans le nombre minimum d'octets;

b) la forme de codage interprétée ne sera pas utilisée pour les types de chaîne;

c) la valeur d'un type sera absente s'il s'agit de la valeur par défaut;

d) les composants d'un type Set (*ensemble*) seront codés dans l'ordre ascendant de leurs valeurs d'étiquette;

e) les composants d'un type Set-of (*ensemble de*) seront codés dans l'ordre ascendant de leurs valeurs d'octet;

f) si la valeur d'un type booléen est égale à "Vrai", la valeur du codage contiendra un octet positionné sur "FF"16;

g) tous les bits éventuellement non utilisés dans l'octet final du codage d'une valeur de chaîne de bits seront positionnés sur zéro;

h) le codage d'un type réel n'utilisera pas les bases 8, 10 et 16 et le facteur de normalisation binaire sera égal à zéro;

i) le codage d'un temps UTC se fera comme spécifié dans la Rec. UIT-T X.690 (1997) | ISO/CEI 8825-1:1998;

j) le codage d'un temps normalisé se fera comme spécifié dans la Rec. UIT-T X.690 (1997) | ISO/CEI 8825-1:1998.

La génération d'un codage distinctif nécessite que la syntaxe abstraite des données devant être codées soit totalement comprise. L'utilisation de l'annuaire peut être nécessaire pour signer des données ou vérifier la signature de données qui contiennent des extensions de protocole ou des syntaxes d'attribut non connues. L'annuaire utilisera les règles suivantes:

– il préservera le codage dont il ne reconnaît pas complètement la syntaxe abstraite et dont il prévoit qu'il doit les signer ultérieurement;

– lorsqu'il signe des données devant être émises, il émettra avec un codage distinctif les données dont il reconnaît entièrement la syntaxe et préservera le codage des autres données; la signature portera sur les codages effectifs émis;

– lors de la vérification des signatures dans les données reçues, il vérifiera la signature par rapport aux données effectivement reçues avant de faire une conversion vers un codage distinctif.

SECTION 2 – CADRE DE CERTIFICAT DE CLÉ PUBLIQUE

L'utilisation du cadre de certificat de clé publique défini dans la présente Spécification est destinée à des applications qui ont des besoins d'authentification, d'intégrité, de confidentialité et de non répudiation.

La liaison d'une clé publique avec une entité est fournie par une autorité au moyen d'une structure de données avec signature numérique appelée "certificat de clé publique". Le format des certificats de clé publique défini dans la présente Spécification comprend un procédé d'extension et un ensemble d'extensions de certificat spécifiques. Si une autorité révoque pour une raison quelconque un certificat de clé publique émis précédemment, il est alors nécessaire de permettre aux utilisateurs de prendre connaissance de cette révocation, de sorte qu'ils n'utilisent pas un certificat non digne de confiance. Les listes de révocation sont un moyen pouvant être mis en œuvre pour notifier des révocations aux utilisateurs. Le format des listes de révocation est défini dans la présente Spécification; il comprend un procédé d'extension et un ensemble d'extensions de liste de révocation. D'autres organismes peuvent également définir des extensions supplémentaires pour les cas de certificat et de liste de révocation utiles dans leurs propres environnements.

Un système de clé publique utilisant des certificats doit pouvoir valider un certificat avant de l'utiliser pour une application. Des procédures de validation sont définies dans la présente Spécification, englobant la vérification de l'intégrité du certificat proprement dit et sa validité pour son utilisation prévue.

L'annuaire utilise des certificats de clé publique pour la fourniture des services de sécurité suivants:

– authentification forte entre et parmi des composants d'annuaire;

– authentification, intégrité et confidentialité des opérations d'annuaire;

– intégrité et authentification des données stockées.

7 Clés publiques et certificats de clé publique

La clé publique doit être obtenue à partir d'une source fiable si un utilisateur doit pouvoir faire confiance à la clé publique d'un autre utilisateur, par exemple pour authentifier son identité. Une telle source, appelée autorité de certification (CA), certifie une clé publique en émettant un certificat de clé publique qui lie la clé publique à l'entité détentrice de la clé privée correspondante. Les procédures utilisées par une autorité de certification pour garantir qu'une entité est effectivement en possession de la clé privée et les autres procédures liées à l'émission de certificats de clé publique sont en dehors du domaine d'application de la présente Spécification. Le certificat, dont le format est spécifié plus loin dans cet article, possède les propriétés suivantes:

– tout utilisateur qui a accès à la clé publique de l'autorité de certification peut récupérer la clé publique certifiée;

– aucun participant autre que l'autorité de certification ne peut modifier le certificat sans être détecté (les certificats ne peuvent pas être falsifiés).

Comme les certificats ne peuvent pas être falsifiés, ils peuvent être publiés dans l'annuaire sans qu'il soit nécessaire de les protéger de manière particulière.

NOTE 1 – Bien que les autorités de certification soient définies sans ambiguïté par un nom distinctif dans l'arbre DIT, ceci n'implique aucune relation entre les organismes des autorités de certification et l'arbre DIT.

Une autorité de certification produit le certificat d'un utilisateur en signant (voir 6.1) un ensemble d'informations contenant le nom distinctif de l'utilisateur, une clé publique ainsi qu'un *identificateur unique* optionnel contenant des informations supplémentaires concernant l'utilisateur. La forme exacte du contenu de l'identificateur unique n'est pas spécifiée dans la présente Spécification et laissée au choix de l'autorité de certification; il peut s'agir, par exemple, d'un identificateur d'objet, d'un certificat, d'une date ou d'une autre forme de certification de la validité du nom distinctif. D'une manière spécifique, le certificat d'un utilisateur avec le nom distinctif A et l'identificateur unique UA, produit par l'autorité de certification avec le nom CA et l'identificateur unique UCA se présentera sous la forme suivante:

$$CA\langle\langle A\rangle\rangle = CA\{V,SN,AI,CA,UCA,A,UA,Ap,T^A\}$$

dans laquelle V indique la version du certificat, SN le numéro de série du certificat, AI l'identificateur de l'algorithme utilisé pour signer le certificat, UCA l'identificateur unique optionnel de l'autorité de certification, UA l'identificateur optionnel unique de l'utilisateur A et T^A la durée de validité du certificat, constituée des dates de début et de fin de validité du certificat. La durée de validité du certificat correspond à l'intervalle de temps pendant lequel l'autorité de certification garantit qu'elle maintiendra des informations concernant le statut de certification, c'est-à-dire la publication de données de révocation. Si l'on admet que la durée T^A n'est pas modifiée avec une périodicité inférieure à 24 heures, il est prévu que les systèmes utiliseront le temps universel coordonné comme base de référence de temps. Tout utilisateur connaissant la clé publique CAp de l'autorité peut vérifier la validité de la signature du certificat. Le type de données ASN.1 suivant est utilisable pour la représentation des certificats:

```
Certificate                     ::=     SIGNED { SEQUENCE {
    version                 [0] Version DEFAULT v1,
    serialNumber                CertificateSerialNumber,
    signature                   AlgorithmIdentifier,
    issuer                      Name,
    validity                    Validity,
    subject                     Name,
    subjectPublicKeyInfo        SubjectPublicKeyInfo,
    issuerUniqueIdentifier  [1] IMPLICIT UniqueIdentifier OPTIONAL,
                                -- si ce composant est présent, la version doit être v2 ou v3
    subjectUniqueIdentifier [2] IMPLICIT UniqueIdentifier OPTIONAL,
                                -- si ce composant est présent, la version doit être v2 ou v3
    extensions              [3] Extensions OPTIONAL
                                -- si ce composant est présent, la version doit être v3 -- } }

Version                         ::=     INTEGER { v1(0), v2(1), v3(2) }

CertificateSerialNumber         ::=     INTEGER

AlgorithmIdentifier             ::=     SEQUENCE {
algorithm           ALGORITHM.&id ({SupportedAlgorithms}),
parameters              ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }
-- La définition des informations suivantes est suspendue, éventuellement jusqu'à la disponibilité de
-- profils normalisés ou de déclarations de conformité d'implémentation de protocole. L'ensemble est
-- nécessaire pour la spécification de contraintes pour le composant parameters de l'identificateur.
-- SupportedAlgorithms          ALGORITHM       ::=   { ... }

Validity                ::=     SEQUENCE {
    notBefore   Time,
    notAfter    Time }

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

Time  ::=  CHOICE {
    utcTime             UTCTime,
    generalizedTime     GeneralizedTime }

Extensions ::= SEQUENCE OF Extension

Extension ::=   SEQUENCE {
    extnId      EXTENSION.&id ({ExtensionSet}),
    critical    BOOLEAN DEFAULT FALSE,
    extnValue   OCTET STRING
                -- contient un codage DER d'une valeur du type &ExtnType
                -- pour l'objet extension identifié par extnId -- }

ExtensionSet   EXTENSION        ::=     { ... }
```

Avant l'utilisation d'une valeur du type **Time** dans toute opération de comparaison, par exemple dans une règle de concordance lors d'une recherche, et si la syntaxe choisie pour **Time** (*temps*) est celle du type **UTCTime** (*temps universel coordonné*) la valeur du champ "année" à deux chiffres sera normalisée de la manière suivante sous la forme d'une valeur d'année à quatre chiffres:

- ajouter 2000 si la valeur à deux chiffres est comprise entre 00 et 49, bornes incluses;
- ajouter 1900 si la valeur à deux chiffres est comprise entre 50 et 99, bornes incluses.

NOTE 2 – L'utilisation du type **GeneralizedTime** (*temps généralisé*) peut faire obstacle à l'interfonctionnement avec des implémentations qui n'ont pas connaissance de la possibilité d'un choix entre les types **UTCTime** et **GeneralizedTime**. Il est de la responsabilité des personnes qui spécifient les domaines d'utilisation des certificats définis dans la présente Spécification d'annuaire, par exemple des groupes de définition de profil, de tenir compte de l'utilisation possible du type **GeneralizedTime**. Le type **UTCTime** ne sera utilisé en aucun cas pour des dates supérieures à 2049.

version représente la version du certificat codé. La version du certificat sera égale à v3 si le composant **extensions** figure dans le certificat. Elle doit être égale à v2 ou v3 si le composant **issuerUniqueIdentifier** (*identificateur unique de l'émetteur*) ou **subjectUniqueIdentifier** (*identificateur unique du sujet*) est présent.

Si des éléments inconnus figurent dans les extensions et si l'extension n'est pas marquée comme étant critique, ces éléments inconnus seront alors ignorés conformément aux règles d'extensibilité décrites au 7.5.2.2 de la Rec. UIT-T X.519 | ISO/CEI 9594-5.

Le composant **serialNumber** (*numéro de série*) est un nombre entier attribué par l'autorité de certification à tout certificat. La valeur **serialNumber** doit être unique pour tout certificat émis par une autorité de certification donnée (c'est-à-dire que le nom de l'émetteur et le numéro de série identifient un certificat unique).

Le composant **signature** contient l'identificateur d'algorithme de l'algorithme et de la fonction de hachage utilisés par l'autorité de certification pour la signature du certificat (par exemple les chiffrements md5WithRSAEncryption, sha-1WithRSAEncryption, id-dsa-with-sha1, etc.).

Le composant **issuer** (*émetteur*) indique l'entité qui a signé et émis le certificat.

Le composant **validity** (*validité*) est l'intervalle de temps pendant lequel l'autorité de certification garantit qu'elle maintiendra des informations concernant le statut du certificat.

Le composant **subject** (*sujet*) indique l'entité associée à la clé publique qui se trouve dans le champ "clé publique du sujet".

Le composant **subjectPublicKeyInfo** (*informations de clé publique du sujet*) est utilisé pour véhiculer la clé publique en cours de certification et indiquer l'algorithme dont cette clé publique constitue une instance (par exemple, le chiffrement rsaEncryption, dhpublicnumber, id-dsa, etc.).

Le composant **issuerUniqueIdentifier** est utilisé pour identifier sans ambiguïté un émetteur en cas de réutilisation d'un nom.

Le composant **subjectUniqueIdentifier** est utilisé pour identifier sans ambiguïté un sujet en cas de réutilisation d'un nom.

NOTE 3 – Dans des situations où un nom distinctif peut être à nouveau attribué à un utilisateur différent par une autorité de dénomination, les autorités de certification peuvent utiliser l'identificateur unique pour faire la distinction entre des instances réutilisées. Si toutefois un même utilisateur reçoit des certificats en provenance de plusieurs autorités de certification, il est recommandé que ces dernières coordonnent l'attribution d'identificateurs unique dans le cadre de leurs procédures d'enregistrement d'utilisateur.

Le champ **extensions** permet l'ajout de nouveaux champs à la structure sans en modifier la définition ASN.1. Un champ d'extension se constitue d'un identificateur d'extension, d'un fanion critique et d'un codage d'une valeur de données de type ASN.1 associée à l'extension concernée. Lorsque l'ordre des extensions individuelles au sein d'une expression **SEQUENCE** est significatif, leur spécification contiendra les règles de signification de l'ordre utilisé. Si une implémentation ne reconnaît pas une extension lors de son traitement et si le fanion critique est positionné sur "Faux", elle peut alors ignorer cette extension. Si par contre le fanion critique est positionné sur "Vrai", les extensions non reconnues auront alors pour effet de faire considérer la structure comme non valide, ce qui signifie qu'une extension critique non reconnue dans un certificat conduira au rejet d'une signature utilisant ce certificat. Lorsqu'elle reconnaît une extension et qu'elle est capable de la traiter, une implémentation utilisant des certificats doit traiter ladite extension quelle que soit la valeur du marquage de criticité. Il convient de noter que toute extension marquée comme étant non critique entraînera deux types de comportement contradictoires parmi les systèmes utilisant des certificats, avec d'une part les systèmes qui traiteront l'extension et, d'autre part, les systèmes qui ne reconnaîtront pas l'extension et l'ignoreront.

Si des éléments inconnus figurent dans une extension qui n'est pas marquée comme étant critique, ces éléments inconnus seront alors ignorés conformément aux règles d'extensibilité décrites au 7.5.2.2 de la Rec. UIT-T X.519 | ISO/CEI 9594-5.

Une autorité de certification a le choix entre trois solutions en présence d'une extension:

i) elle peut supprimer l'extension du certificat;

ii) elle peut incorporer dans celui-ci l'extension et la marquer comme étant non critique;

iii) elle peut incorporer l'extension dans le certificat et la marquer comme étant critique.

Un mécanisme de validation peut opérer de deux manières en présence d'une extension:

i) il peut ignorer l'extension et accepter le certificat (la situation à tous autres égards demeurant inchangée);

ii) il peut traiter l'extension et accepter ou refuser le certificat en fonction du contenu de l'extension et des conditions dans lesquelles le traitement est effectué (compte tenu, par exemple, des valeurs retenues pour les variables de traitement d'itinéraire).

Certaines extensions peuvent uniquement être marquées comme étant critiques. En pareils cas, un mécanisme de validation qui comprend l'extension en assure le traitement, et l'acceptation ou le refus du certificat dépend (du moins en partie) du contenu de l'extension. Un mécanisme de validation qui ne comprend pas l'extension refuse le certificat.

Certaines extensions peuvent uniquement être marquées comme étant non critiques. En pareils cas, un mécanisme de validation qui comprend l'extension en assure le traitement, et l'acceptation ou le rejet du certificat dépend (du moins en partie) du contenu de l'extension. Un mécanisme de validation qui ne comprend pas l'extension accepte le certificat (sauf si des facteurs autres que l'extension considérée conduisent à le refuser).

Certaines extensions peuvent être marquées comme étant critiques ou non critiques. En pareils cas, un mécanisme de validation qui comprend l'extension en assure le traitement, et l'acceptation ou le refus du certificat dépend (du moins en partie) du contenu de l'extension, quel que soit le marquage de criticité. Un mécanisme de validation qui ne comprend pas l'extension accepte le certificat si l'extension est marquée comme étant non critique (sauf si des facteurs autres que l'extension considérée conduisent à le refuser) et refuse le certificat si l'extension est marquée comme étant critique.

Lorsqu'une autorité de certification envisage d'incorporer une extension dans un certificat, elle espère toujours que le but de l'incorporation de cette extension sera perçu le plus largement possible. S'il est nécessaire d'examiner le contenu de l'extension avant de pouvoir utiliser le certificat, une autorité de certification marquera l'extension comme étant critique, en étant bien consciente que tout mécanisme de validation qui ne traite pas l'extension conduira au refus du certificat, (probablement en limitant la série d'applications que le certificat peut vérifier). L'autorité de certification peut marquer certaines extensions comme étant non critiques afin d'assurer la compatibilité vers l'arrière pour des applications de validation qui ne peuvent pas traiter les extensions. Lorsque la nécessité d'assurer la compatibilité vers l'arrière et l'interfonctionnement pour des applications de validation incapables de traiter les extensions l'emporte sur la capacité de l'autorité de certification à appliquer les extensions, les éventuelles extensions marquées comme étant critiques seront marquées comme étant non critiques. Il est très probable que les autorités de certification marqueront comme étant non critiques les éventuelles extensions marquées comme étant critiques pendant une période de transition durant laquelle il sera procédé à l'amélioration des applications de traitement des certificats des vérificateurs pour en faire des applications capables de traiter les extensions.

Des extensions spécifiques peuvent être définies par des Recommandations UIT-T | Normes internationales ou par tout organisme qui en éprouve le besoin. L'identificateur d'objet qui désigne une extension sera défini conformément à la Rec. UIT-T X.660 | ISO/CEI 9834-1. L'article 8 de la présente Spécification d'annuaire définit des extensions de certificat normalisées.

La classe d'objet suivante sert à définir des extensions spécifiques.

```
EXTENSION ::= CLASS {
    &id     OBJECT IDENTIFIER UNIQUE,
    &ExtnType }
WITH SYNTAX {
    SYNTAX              &ExtnType
    IDENTIFIED BY       &id }
```

Il existe deux types primaires de certificats de clé publique: les certificats d'entité finale et les certificats d'autorité de certification.

Un certificat d'entité finale est un certificat émis par une autorité de certification à destination d'un sujet qui n'est pas lui-même émetteur d'autres certificats de clé publique.

Un certificat d'autorité de certification est un certificat émis par une autorité de certification à destination d'un sujet qui est également une autorité de certification et qui est de ce fait en mesure d'émettre des certificats de clé publique. Les certificats d'autorité de certification peuvent être classés selon les types suivants:

- certificat auto-émis – Il s'agit d'un certificat dont l'émetteur et le sujet sont la même autorité de certification. Une autorité de certification peut utiliser des certificats émis à l'ordre d'elle-même, par exemple lors d'une opération de renouvellement de clé pour transférer la confiance de l'ancienne clé vers la nouvelle;
- certificat auto-signé – Il s'agit d'un cas particulier de certificats auto-émis pour lequel la clé privée utilisée par l'autorité de certification pour la signature du certificat correspond à la clé publique qui est certifiée au sein du certificat. Une autorité de certification peut, par exemple, utiliser un certificat signé par elle-même pour publier sa clé publique ou d'autres informations concernant son fonctionnement;
- certificat croisé – Il s'agit d'un certificat dont l'émetteur et le sujet sont des autorités de certification différentes. Des autorités de certification émettent des certificats destinés à d'autres autorités de certification, soit comme procédé d'autorisation de l'existence de l'autorité de certification sujette (par exemple, au sein d'une hiérarchie stricte), soit pour reconnaître l'existence de l'autorité de certification sujette (par exemple dans un modèle de confiance réparti). La structure de certificat croisé est utilisée dans les deux cas.

L'entrée d'annuaire de tout utilisateur A qui participe à une authentification forte contient le ou les certificats de cet utilisateur. Un tel certificat est généré par une autorité de certification de A qui est une entité appartenant à l'arbre DIT. Une autorité de certification de A, qui peut ne pas être unique, est indiquée par la notation CA(A) ou plus simplement CA si A est implicite. La clé publique de A peut être retrouvée par tout utilisateur qui connaît la clé publique de CA. Il s'ensuit que la recherche des clés publiques se fait de manière récursive.

Le processus prend fin lorsqu'un utilisateur A qui tente d'obtenir la clé publique d'un autre utilisateur B a déjà obtenu la clé publique de CA(B). L'entrée d'annuaire de chaque autorité de certification X contient un certain nombre de certificats afin de permettre à l'utilisateur A d'obtenir la clé publique de CA(B). Ces certificats sont de deux types, d'une part les certificats directs de X générés par d'autres autorités de certification et d'autre part les certificats en retour générés par l'autorité X elle-même, qui sont les clés publiques certifiées concernant d'autres autorités de certification. L'existence de ces certificats permet aux utilisateurs de construire des itinéraires de certification d'un point vers un autre.

Une liste de certificats permettant à un utilisateur donné d'obtenir la clé publique d'un autre utilisateur est appelée *itinéraire de certification*. Chaque élément de la liste se constitue d'un certificat de l'autorité de certification de l'élément suivant de la liste. Un itinéraire de certification de A vers B (représenté par la notation A→B):

- débute par un certificat produit par l'autorité de certification (A), à savoir CA(A) «X1» pour une certaine entité X1;
- se poursuit par d'autres certificats Xi"Xi+1";
- se termine par le certificat de B.

Un itinéraire de certification constitue une chaîne logique ininterrompue de points fiables dans l'arbre d'informations d'annuaire entre les deux utilisateurs souhaitant s'authentifier. Les détails de la méthode précise employée par les utilisateurs A et B pour obtenir les itinéraires de certification A→B et B→A peuvent différer. Une façon de faciliter cette recherche consiste à mettre en place une hiérarchie d'autorités de certification, qui peut différer en tout ou partie de la hiérarchie de l'arbre DIT. Cette solution présente l'avantage que des utilisateurs qui ont des autorités de certification appartenant à la hiérarchie peuvent établir un itinéraire de certification entre eux en utilisant l'annuaire sans autres informations préalables. Chaque autorité de certification peut, pour ce faire, stocker un certificat et un certificat en retour permettant de désigner son autorité de certification supérieure.

Un utilisateur peut obtenir un ou plusieurs certificats émis par une ou plusieurs autorités de certification. Chaque certificat porte le nom de l'autorité émettrice. Les types de données ASN.1 suivants peuvent être utilisés pour représenter les certificats et un itinéraire de certification:

```
Certificates                ::=  SEQUENCE {
    userCertificate         Certificate,
    certificationPath       CertPath OPTIONAL }

CertificationPath           ::=  SEQUENCE {
    userCertificate         Certificate,
    theCACertificates       SEQUENCE OF CertificatePair OPTIONAL }
```

Les types de données ASN.1 suivants peuvent en outre être utilisés pour représenter les itinéraires de certification directs. Ce composant contient l'itinéraire de certification qui peut pointer en retour vers l'initiateur.

```
CertPath  ::=  SEQUENCE OF CrossCertificates

CrossCertificates           ::=       SET OF Certificate
```

Chaque certificat dans un itinéraire de certification doit être unique. Aucun certificat ne peut figurer plus d'une fois dans une valeur du composant **theCACertificates** de l'itinéraire **CertificationPath**, ou dans une valeur du certificat **Certificate** dans le composant **CrossCertificates** de l'itinéraire **CertPath**.

7.1 Génération de paires de clés

La politique de gestion globale de la sécurité d'une implémentation définira le cycle de vie des paires de clés et, de ce fait, ne fait pas partie du domaine d'application de ce cadre. Il est toutefois d'une importance vitale pour la sécurité globale que toutes les clés privées restent connues uniquement des utilisateurs auxquels elles appartiennent.

Il est difficile pour un utilisateur humain de mémoriser des données de clé, de sorte qu'une méthode de transport convenable sera utilisée. Un moyen possible consiste à utiliser une "carte intelligente", contenant la clé privée et (de manière optionnelle) les clés publiques de l'utilisateur ainsi que le certificat de l'utilisateur et une copie de la clé publique de l'autorité de certification. L'utilisation de cette carte sera en outre protégée, par exemple au moins à l'aide d'un numéro d'identification personnel (PIN, *personal identification number*), ce qui augmente la sécurité du système en imposant que l'utilisateur soit en possession de la carte et sache comment y accéder. La méthode exacte utilisée pour le stockage de ces données est toutefois en dehors du domaine d'application de la présente Spécification.

Une paire de clés utilisateur peut être produite de l'une des trois manières suivantes:

a) l'utilisateur génère sa propre paire de clés. Cette méthode présente l'avantage qu'une clé privée d'utilisateur n'est jamais confiée à une autre entité, mais nécessite un certain niveau de compétence de l'utilisateur;

b) la paire de clés est générée par un tiers. Ce dernier livrera à l'utilisateur sa clé privée d'une manière physiquement fiable, puis détruira de manière active toutes les informations concernant la création de la paire de clés ainsi que les clés proprement dites. Des mesures de sécurité physique convenables seront utilisées pour garantir que le tiers et les opérations sur les données ne peuvent pas être manipulés;

c) la paire de clés est générée par l'autorité de certification. Il s'agit d'un cas particulier de l'alinéa b) auquel s'appliquent les mêmes considérations.

NOTE – L'autorité de certification présente déjà des fonctionnalités fiables vis-à-vis de l'utilisateur et sera soumise aux mesures de sécurité physique nécessaires. Cette méthode présente l'avantage de ne pas nécessiter de transfert de données sécurisé vers l'autorité de certification à des fins de certification.

Le système de chiffrement utilisé impose des contraintes (techniques) particulières pour la génération des clés.

7.2 Création d'un certificat de clé publique

Un certificat de clé publique associe la clé publique et un nom distinctif unique de l'utilisateur concerné. De ce fait:

a) une autorité de certification doit avoir accepté l'identité fournie par un utilisateur avant de créer un certificat qui lui est destiné;

b) une autorité de certification n'émettra pas de certificats pour deux utilisateurs portant le même nom.

Il est important que le transfert d'informations vers l'autorité de certification ne soit pas mis en danger et des mesures de sécurité physique convenables doivent être prises. De ce fait:

a) une violation grave de la sécurité se produirait si l'autorité de certification émettait un certificat pour un utilisateur avec une clé publique manipulée;

b) si la génération des paires de clés se fait conformément aux alinéas 7.1 b) ou 7.1 c), la clé privée de l'utilisateur doit alors lui être remise de manière fiable;

c) l'utilisateur peut mettre en œuvre diverses méthodes (en ligne ou en temps différé) pour communiquer sa clé publique de manière fiable à l'autorité de certification si la génération des paires de clés se fait conformément aux alinéas 7.1 a) ou 7.1 b). Les méthodes en ligne peuvent fournir une plus grande souplesse pour les opérations distantes entre l'utilisateur et l'autorité de certification.

Un certificat de clé publique est un ensemble d'informations disponible de manière publique qui ne nécessite aucune mesure spéciale de protection pour son transport vers l'annuaire. Comme il est produit en temps différé par une autorité de certification pour le compte d'un utilisateur qui en recevra une copie, l'utilisateur a uniquement besoin de stocker ces informations dans son entrée d'annuaire lors d'un accès ultérieur à l'annuaire. L'autorité de certification peut, en variante, déposer le certificat pour le compte de l'utilisateur, auquel cas cet agent recevra les droits d'accès nécessaires.

7.3 Validité des certificats

L'autorité qui émet des certificats (de clé publique ou d'attribut) est également responsable de l'indication de leur validité. Les certificats peuvent en général être révoqués ultérieurement. Cette révocation et sa notification peuvent être effectuées directement par l'autorité émettrice du certificat ou de manière indirecte par une autre autorité dûment mandatée par la première. Une autorité qui émet des certificats a l'obligation d'indiquer, si possible dans une déclaration publique de ses pratiques figurant dans les certificats proprement dits, ou par d'autres moyens identifiés:

- si les certificats ne peuvent pas être révoqués;
- si les certificats peuvent être révoqués directement par l'autorité qui a émis le certificat;
- si l'autorité émettrice du certificat permet à une autre autorité d'effectuer la révocation.

Les autorités qui révoquent des certificats doivent déclarer par des moyens similaires quelles sont les méthodes pouvant être utilisées, par des participants qui lui font confiance, pour obtenir des informations de révocation de statut concernant des certificats qu'elles ont émis. La présente Spécification définit un procédé de liste de révocation de certificat (CRL) mais n'interdit pas d'en utiliser d'autres. Lorsqu'ils appliquent les procédures de traitement d'itinéraire décrites dans l'article 10 et la procédure d'itinéraire de délégation décrite dans l'article 16, les participants faisant confiance valident un certificat en vérifiant de manière adéquate les informations de statut de révocation pour tous les certificats concernés.

Les certificats, comprenant les certificats de clé publique et les certificats d'attribut, posséderont une durée de vie au bout de laquelle ils expirent. L'autorité assurera le remplacement en temps opportun des certificats caducs ou en cours d'expiration, afin d'assurer la continuité du service. La date de notification de la révocation correspond à la date et à l'heure de la première publication d'une notification de révocation dans une liste CRL, pouvant être une liste de base ou une liste dCRL. La date de notification de révocation figure dans le champ **thisUpdate** (*cette mise à jour*) de la liste CRL. La date de révocation correspond à la date et l'heure de la révocation effective du certificat par l'autorité de certification, qui peut différer de la date de première publication dans une liste CRL. La date de révocation figure, pour une liste CRL, dans le composant **revocationDate** (*date de révocation*).

Les deux points suivants sont liés à la durée de vie des certificats:

- la validité des certificats peut être déterminée de manière à ce que chacun d'eux prenne sa validité au moment de l'expiration de son prédécesseur ou peut permettre une durée de chevauchement. Le dernier cas peut permettre à l'autorité d'éviter l'installation et la distribution d'un grand nombre de certificats qui peuvent expirer simultanément;
- les certificats caducs sont normalement retirés de l'annuaire. L'autorité a la charge et la responsabilité de conserver les anciens certificats pendant une certaine durée en cas de fourniture d'un service de non-répudiation des données.

Les certificats peuvent être révoqués avant leur date d'expiration, par exemple si l'on présume que la clé privée de l'utilisateur risque d'avoir été mise en danger, si l'utilisateur ne doit plus être certifié par l'autorité ou si l'on présume que le certificat de l'autorité a été mis en danger. L'autorité fera connaître la révocation d'un certificat d'utilisateur ou d'un certificat d'autorité et fournira le cas échéant un nouveau certificat. L'autorité peut utiliser ensuite une procédure en temps différé pour indiquer la révocation au détenteur du certificat.

Une autorité qui émet des certificats et les révoque par la suite:

a) peut avoir l'obligation de maintenir un enregistrement d'audit de ses événements de révocation pour tous les types de certificat qu'elle a émis (par exemple, des certificats de clé publique, des certificats d'attribut pour des entités finales ainsi que pour d'autres autorités);

b) fournira aux participants qui lui font confiance des informations de statut de révocation par le biais de listes CRL, d'un protocole de statut en ligne ou par d'autres procédés de publication d'informations de statut de révocation;

c) maintiendra et publiera les listes CRL qu'elle utilise éventuellement, même si ces dernières sont vides.

Les participants faisant confiance peuvent utiliser un certain nombre de procédés pour localiser les informations de statut de révocation fournies pas une autorité. Il peut s'agir, par exemple, d'un pointeur figurant dans le certificat lui-même qui indique au participant faisant confiance un emplacement où se trouvent les informations de révocation. Il peut également s'agir d'un pointeur dans une liste de révocation qui renvoie le participant faisant confiance vers un autre emplacement. Le participant faisant confiance peut localiser les informations de révocation au sein d'un référentiel (par exemple, un annuaire) ou par d'autres moyens en dehors du domaine d'application de la présente Spécification (gérés, par exemple, de manière locale).

La maintenance des entrées de l'annuaire affectées par les listes de révocation de l'autorité est de la responsabilité de l'annuaire et de ses utilisateurs, agissant dans le cadre de la politique de sécurité. L'utilisateur peut, par exemple, modifier son entrée d'objet en remplaçant un ancien certificat par un nouveau. Ce dernier sera utilisé ensuite pour authentifier l'utilisateur vis-à-vis de l'annuaire.

Si des listes de révocation sont publiées dans l'annuaire, elles figureront alors dans des entrées sous la forme d'attributs des types suivants:

- liste de révocation de certificat;
- liste de révocation d'autorité;
- liste delta de révocation;
- Liste de révocation de certificat d'attribut;
- liste de révocation d'autorité d'attribut.

```
CertificateList          ::=   SIGNED { SEQUENCE {
    version                        Version OPTIONAL,
                                   -- si elle est présente, la version doit être v2
    signature                      AlgorithmIdentifier,
    issuer                         Name,
    thisUpdate                     Time,
    nextUpdate                     Time OPTIONAL,
    revokedCertificates            SEQUENCE OF SEQUENCE {
        serialNumber                       CertificateSerialNumber,
        revocationDate                     Time,
        crlEntryExtensions             Extensions OPTIONAL } OPTIONAL,
    crlExtensions      [0]         Extensions OPTIONAL }}
```

Le composant **version** représente la version de la liste de révocation codée. La version sera égale à v2 si le composant **extensions** présent dans la liste de révocation est marqué comme étant critique. La version peut être égale à v2 ou absente si aucun composant **extensions** n'est présent et marqué comme critique dans la liste de révocation.

Le composant **signature** contient l'identificateur de l'algorithme utilisé par l'autorité pour signer la liste de révocation.

Le composant **issuer** identifie l'entité qui a signé et émis la liste de révocation.

Le composant **thisUpdate** contient la date et l'heure d'émission de cette liste de révocation.

Le composant **nextUpdate** (*mise à jour suivante*) indique, s'il est présent, la date et l'heure auxquelles sera émise la prochaine liste de révocation de cette série. La prochaine liste de révocation peut être émise avant la date indiquée, mais en aucun cas après.

Le composant **revokedCertificates** (*certificats révoqués*) identifie des certificats qui ont été révoqués. Les certificats révoqués sont identifiés par leur numéro de série. Si aucun des certificats visés par cette liste CRL n'a été révoqué, il est fortement conseillé d'omettre le paramètre **revokedCertificates** de la liste CRL plutôt que de l'inclure avec une **SEQUENCE** vide.

Le composant **crlExtensions** (*extensions de liste CRL*) contient une ou plusieurs extensions de liste CRL, s'il est présent.

NOTE 1 – La vérification de la totalité de la liste de certificats est un problème local. La liste sera considérée comme non ordonnée, à moins que des règles d'ordre particulières n'aient été indiquées par l'autorité émettrice, par exemple dans la politique de cette dernière.

NOTE 2 – Si un service de non-répudiation des données dépend de clés fournies par l'autorité, ce service doit alors s'assurer que toutes les clés pertinentes de l'autorité (révoquées ou caduques) et les listes de révocation horodatées sont archivées et certifiées par une autorité actuelle.

NOTE 3 – L'élément de version de la liste **CertificateList** (*liste de certificats*) sera présent si une extension quelconque contenue dans une liste **CertificateList** est définie comme étant critique. L'élément **version** sera absent si aucune extension critique n'est présente, ce qui peut permettre à une implémentation prenant uniquement en charge la version 1 de liste CRL de continuer à utiliser cette dernière si l'examen de la séquence **revokedCertificates** dans la liste CRL ne révèle aucune extension. Une implémentation qui prend en charge la version 2 (ou supérieure) de liste CRL peut, en l'absence d'indication de la version, être en mesure d'optimiser son traitement si elle peut déterminer dans une phase précoce du traitement qu'aucune extension critique ne figure dans la liste CRL.

NOTE 4 – Si une implémentation qui traite une liste de révocation de certificat ne reconnaît pas une extension critique figurant dans le champ **crlEntryExtensions** (*extensions de liste CRL*) elle fera alors l'hypothèse, au minimum, que le certificat identifié a été révoqué et n'est plus valide et procédera aux autres actions concernant le certificat révoqué, telles qu'elles sont imposées par la politique locale. Si une implémentation ne reconnaît pas une extension critique dans le champ **crlExtensions**, elle fera alors l'hypothèse que le certificat en question a été révoqué et n'est plus valide. Dans ce dernier cas toutefois, étant donné que la liste peut être incomplète, les certificats qui n'ont pas été identifiés comme étant révoqués ne peuvent pas être considérés comme valides. La politique locale peut imposer dans tous les cas d'autres actions et/ou imposer des actions plus strictes que celles énoncées dans la présente Spécification.

NOTE 5 – Si une extension affecte le traitement de la liste (des listes CRL multiples doivent, par exemple, être examinées en totalité pour déterminer les certificats révoqués, ou un élément peut représenter un domaine de certificats), cette extension sera alors marquée comme critique dans le champ **crlExtensions** quel que soit l'endroit où elle figure dans la liste CRL. Une extension indiquée dans le champ **crlEntryExtensions** d'un élément sera insérée dans cet élément et affectera uniquement le ou les certificats spécifiés par ce dernier.

NOTE 6 – L'article 8 de la présente Spécification d'annuaire définit des extensions normalisées pour les listes CRL.

8 Certificat de clé publique et extensions de liste CRL

Les extensions de certificat définies dans cet article sont utilisables avec des certificats de clé publique, sauf indication contraire. L'article 15 définit les extensions utilisables avec des certificats d'attribut. Les extensions de liste CRL définies dans cet article peuvent être utilisées pour des listes CRL et des listes CARL, ainsi que pour des listes ACRL et AARL définies dans l'article 17.

Cet article spécifie des extensions dans les domaines suivants:

a) *informations de clé et de politique*: ces extensions de certificat et de liste CRL véhiculent des informations supplémentaires pour les clés invoquées, contenant des identificateurs de clé pour le sujet et l'émetteur de la clé, des indicateurs d'utilisation prévue ou restreinte de la clé, ainsi que des indicateurs de politique de certificat;

b) *attributs de sujet et d'émetteur*: ces extensions de certificat et de liste CRL prennent en charge des variantes de nom sous des formats divers pour un sujet de certificat, un émetteur de certificat ou un émetteur de liste CRL. Ces extensions peuvent également véhiculer des informations d'attribut supplémentaires concernant le sujet du certificat, de manière à aider un utilisateur à établir avec certitude que le sujet du certificat est une personne ou une entité donnée;

c) *contraintes d'itinéraire de certification*: ces extensions de certificat permettent d'inclure des spécifications de contrainte dans des certificats CA, c'est-à-dire des certificats destinés à des autorités de certification et émis par d'autres autorités de certification, ce qui facilite le traitement automatisé des itinéraires de certification lorsque plusieurs politiques de certification sont impliquées. Le cas de politiques de certification multiples se présente lorsque ces dernières varient en fonction des diverses applications au sein d'un même environnement ou dans le cas d'interfonctionnement avec des environnements externes. Les contraintes peuvent limiter les types de certificat pouvant être émis par l'autorité de certification sujette ou se présenter ultérieurement sur un itinéraire de certification;

d) *extensions de liste CRL de base*: ces extensions de liste CRL permettent à cette dernière d'inclure des indications de motif de révocation, de suspendre un certificat de manière temporaire et d'inclure des numéros de séquence d'émission de liste CRL permettant aux utilisateurs de certificats de détecter les listes CRL absentes dans une séquence de listes en provenance d'un émetteur de liste CRL;

e) *points de répartition de liste CRL et listes CRL delta*: ces extensions de certificat et de liste CRL permettent de subdiviser l'ensemble total des informations de révocation fournies par une autorité de certification en plusieurs listes CRL ou de combiner des informations de révocation issues de plusieurs autorités de certification en une liste CRL unique. Ces extensions prennent également en charge l'utilisation de listes CRL partielles qui indiquent uniquement les modifications par rapport à une liste CRL émise précédemment.

La présence de toute extension dans un certificat ou une liste CRL est une option de l'autorité émettrice de ce certificat ou de cette liste CRL.

Une extension d'un certificat ou d'une liste CRL est marquée comme étant critique ou non critique. Si un système utilisant des certificats ne reconnaît pas le type de champ extension ou n'implémente pas la syntaxe d'une extension critique, il considérera alors le certificat comme non valide. Si un système utilisant des certificats ne reconnaît pas le type de champ extension ou n'implémente pas la syntaxe d'une extension non critique, il peut alors traiter le reste du certificat en ignorant l'extension. Si un système utilisant des certificats reconnaît une extension marquée comme étant non critique, il doit traiter cette extension. Les définitions de type d'extension dans la présente Spécification d'annuaire indiquent si l'extension est toujours critique, toujours non critique ou si son caractère critique peut faire l'objet d'une décision de l'émetteur du certificat ou de la liste CRL. La raison de l'exigence que certaines extensions soient toujours non critiques est de permettre à des implémentations utilisant des certificats et qui n'ont pas besoin de telles extensions, d'omettre leur prise en charge sans compromettre leur capacité d'interfonctionnement avec toutes les autorités de certification.

NOTE – Un système utilisant des certificats peut exiger la présence de certaines extensions non critiques dans un certificat pour accepter ce dernier. La présence obligatoire de telles extensions peut dépendre des règles de politique locale de l'utilisateur de certificat ou d'une règle indiquée par une autorité de certification au système utilisant des certificats par la présence d'un identificateur de politique particulier dans une extension de politiques de certificat marquée comme étant critique.

Une seule instance de chaque type d'extension, au plus, figurera respectivement dans tout type de certificat, de liste CRL ou d'entrée de liste CRL pour toutes les extensions de certificat, de liste CRL et d'entrée de liste CRL définies dans la présente Spécification d'annuaire.

8.1 Traitement de la politique

8.1.1 Politique de certificat

Ce cadre contient trois types d'entités: l'utilisateur de certificat, l'autorité de certification et le sujet du certificat (ou entité finale). Chacune d'elles intervient dans le cadre d'obligations imposées par les deux autres et bénéficie en retour des garanties limitées qu'elles offrent. Ces obligations et garanties sont définies dans une politique de certificat. Une politique de certificat est un document (rédigé généralement dans un langage naturel). Elle peut faire l'objet d'une référence au moyen d'un identificateur unique, qui peut figurer dans l'extension de politiques de certificat du certificat émis vers l'entité finale par l'autorité de certification, et auquel l'utilisateur de certificat fait confiance. Un certificat peut être émis dans le cadre d'une ou de plusieurs politiques. La définition de la politique et l'attribution de l'identificateur sont faites par une autorité de politique. L'ensemble des politiques administrées par une autorité de politique est appelé "domaine de politique". Tous les certificats sont émis conformément à une politique, même si cette dernière ne figure pas dans le certificat ou n'est pas référencée par ce dernier. La présente Spécification ne prescrit ni le style ni le contenu de la politique de certificat.

L'utilisateur de certificat peut être lié à ses obligations résultant de la politique de certificat par le fait d'importer une clé publique d'autorité et de l'utiliser comme ancre de confiance, ou en faisant confiance à un certificat qui contient l'identificateur de politique associé. L'autorité de certification peut être liée à ses propres obligations résultant de la politique par le fait d'émettre un certificat qui contient l'identificateur de politique associé. L'entité finale peut être liée à ses propres obligations résultant de la politique par le fait de demander et d'accepter un certificat qui contient l'identificateur de politique associé et par l'utilisation de la clé privée correspondante. Les implémentations qui n'utilisent pas l'extension de politique de certificat doivent établir les liaisons correspondantes par d'autres moyens.

Le fait qu'une entité déclare simplement la conformité à une politique ne satisfait pas en général les besoins de garanties des autres entités appartenant au cadre. Ces dernières ont besoin d'une raison pour admettre que les autres participants utilisent une implémentation fiable de la politique. Toutefois, si cela est énoncé explicitement dans la politique, les utilisateurs de certificat peuvent accepter les garanties de l'autorité de certification indiquant que ses entités finales sont d'accord pour être liées par leurs obligations résultant de la politique, ce qui évite d'effectuer une confirmation directe avec ces entité finales. Cet aspect de la politique de certificat n'entre pas dans le cadre du domaine d'application de la présente Spécification.

Une autorité de certification peut imposer des limitations à l'utilisation de ses certificats afin de rester maîtresse des risques qu'elle assume par l'émission de certificats. Elle peut, par exemple, restreindre la communauté des utilisateurs de certificat, les buts pour lesquels ces derniers utilisent les certificats et/ou le type de dommages qu'elle est prête à assumer en cas d'une défaillance de sa part ou de ses entités finales. Ces points doivent être définis dans la politique de certificat.

D'autres informations peuvent figurer dans l'extension de politiques de certificat sous la forme de qualificatifs de politique afin d'aider les entités impliquées à comprendre les dispositions de la politique.

8.1.2 Certification croisée

Une autorité de certification peut être le sujet d'un certificat émis par une autre autorité de certification. Le certificat est appelé dans ce cas un certificat croisé, l'autorité de certification constituant le sujet du certificat est appelée autorité de certification sujette et l'autorité de certification qui émet le certificat croisé autorité de certification intermédiaire (voir Figure 2). Le certificat croisé et le certificat de l'entité finale peuvent contenir tous deux une extension de politiques de certificat.

Les garanties et obligations partagées par l'autorité de certification sujette, l'autorité de certification intermédiaire et l'utilisateur de certificat sont définies par la politique de certificat identifiée dans le certificat croisé, en accord avec lequel l'autorité de certification sujette peut agir comme, ou pour le compte d'une entité finale. Les garanties et obligations partagées par le sujet du certificat, l'autorité de certification sujette et l'autorité de certification intermédiaire sont définies par la politique de certificat identifiée dans le certificat de l'entité finale, en accord avec lequel l'autorité de certification intermédiare peut agir comme un utilisateur de certificat ou pour le compte de ce dernier.

Un itinéraire de certification est considéré comme valide sous l'ensemble des politiques communes à tous les certificats de l'itinéraire.

Une autorité de certification intermédiaire peut être à son tour le sujet d'un certificat émis par une autre autorité de certification, ce qui conduit à la création d'itinéraires de certification d'une longueur supérieure à deux certificats. Etant donné que la confiance est affectée par le niveau de diffusion en fonction de l'augmentation de la longueur des itinéraires de certificat, des mesures de contrôle sont nécessaires pour garantir que des certificats d'entité finale avec un niveau de confiance trop faible pour être acceptés seront rejetés par l'utilisateur de certificat. Cette fonction fait partie de la procédure de traitement de l'itinéraire de certification.

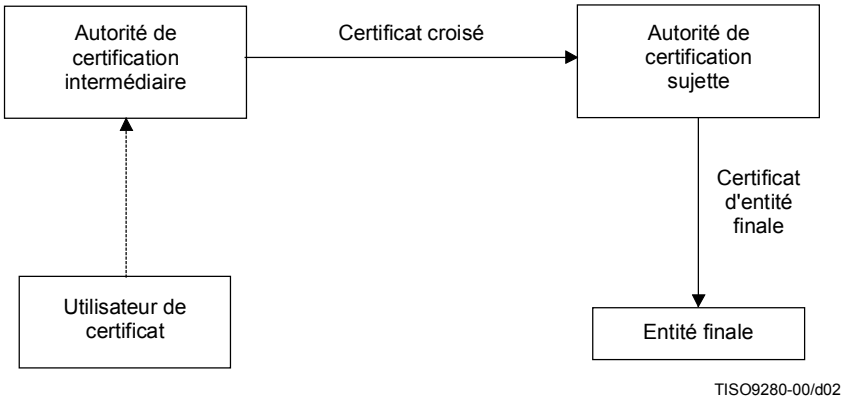


Figure 2 – Certification croisée

Les deux cas particuliers suivants doivent être pris en considération en plus de la situation décrite précédemment:

a) l'autorité de certification n'utilise pas d'extension de politiques de certificat pour véhiculer ses prescriptions de politique à destination des utilisateurs de certificat;

b) l'utilisateur de certificat ou l'autorité de certification intermédiaire délègue les tâches de vérification de politique à l'autorité suivante de l'itinéraire.

Dans le premier cas, le certificat ne doit contenir aucune extension de politiques de certificat et il s'ensuit que l'ensemble des politiques sous lequel l'itinéraire est valide sera vide, l'itinéraire pouvant toutefois être valide. Les utilisateurs de certificat doivent toujours s'assurer qu'ils utilisent le certificat en conformité avec les politiques des autorités de l'itinéraire.

Dans le deuxième cas, l'utilisateur de certificat ou l'autorité de certification doit fournir la valeur spéciale *any-policy* (*toute politique*) dans l'ensemble *initial-policy-set* (*ensemble de politiques initiales*) ou dans le certificat croisé. Lorsqu'un certificat contient la valeur spéciale *any-policy*, il ne peut contenir aucun autre identificateur de politique de certificat. Les identificateurs *any-policy* ne doivent posséder aucun qualificatif de politique associé.

L'utilisateur de certificat peut s'assurer que toutes ses obligations sont véhiculées conformément à la norme en positionnant l'indicateur *initial-explicit-policy* (*politique initiale explicite*). De cette manière, seules des autorités qui utilisent l'extension de politiques de certificat normalisée pour réaliser des liaisons sont acceptées sur l'itinéraire et les utilisateurs de certificats ne sont soumis à aucune obligation supplémentaire. Etant donné que les autorités contractent des obligations lorsqu'elles agissent comme un utilisateur de certificat ou pour son compte, elles peuvent s'assurer que toutes leurs obligations sont véhiculées conformément à la norme en positionnant le composant **requireExplicitPolicy** (*exigence de politique explicite*) dans le certificat croisé.

8.1.3 Mappage de politique

Certains itinéraires de certification peuvent franchir des frontières entre domaines de politique. Les garanties et obligations selon lesquelles est émis le certificat peuvent être matériellement équivalentes à tout ou partie des garanties et obligations selon lesquelles l'autorité de certification sujette émet des certificats destinés à des entités finales, même si les autorités de politique sous lesquelles agissent les deux autorités de certification peuvent avoir choisi des identificateurs uniques différents pour ces politiques matériellement équivalentes. L'autorité de certification intermédiaire peut, dans ce cas, faire figurer dans le certificat croisé une extension de mappage de politique. Dans une telle extension, l'autorité de certification intermédiaire garantit à l'utilisateur de certificat qu'il continuera à bénéficier des garanties habituelles et qu'il doit continuer à remplir ses obligations habituelles, même si des entités suivantes de l'itinéraire de certification agissent dans un autre domaine de politique. L'autorité de certification intermédiaire doit indiquer un ou plusieurs mappages pour chacun des sous-ensembles de politiques sous lesquelles est émis le certificat croisé; elle ne doit pas indiquer de mappage pour toute autre politique. Si un ou plusieurs certificats de politique sous lesquels intervient l'autorité de certification sujette sont identiques à ceux sous lesquels intervient l'autorité de certification intermédiaire (c'est-à-dire s'ils possèdent le même identificateur unique), alors ces identificateurs ne doivent alors pas figurer dans l'extension de mise en correspondance de politique mais doivent être présents dans l'extension de politiques de certificat.

Le mappage de politique a pour effet, pour tous les certificats sur la suite de l'itinéraire de certification, de convertir tous les identificateurs de politique vers un identificateur de la politique équivalente, telle qu'il est reconnu par l'utilisateur de certificat.

Les politiques ne seront pas mappées, dans un sens ou dans l'autre, avec la valeur spéciale *any-policy*.

Les utilisateurs de certificat peuvent établir s'ils peuvent ou non faire confiance à des certificats émis dans un domaine de politique autre que le leur, en dépit du fait qu'une autorité de certification intermédiaire fiable peut décider que sa politique est matériellement équivalente à leur propre politique. Ceci peut se faire en positionnant la valeur spéciale *initial-policy-mapping-inhibit* (*interdiction de mappage de politique initiale*) sur la procédure de validation d'itinéraire. Une autorité de certification intermédiaire peut en outre agir de même pour le compte de ses utilisateurs de certificat. Elle peut positionner la valeur du composant **inhibitPolicyMapping** (*interdiction de mappage de politique*) dans une extension de contraintes de politique pour s'assurer que les utilisateurs de certificat appliquent correctement cette prescription.

8.1.4 Traitement de l'itinéraire de certification

L'utilisateur de certificat a le choix entre deux stratégies:

a) il peut exiger que l'itinéraire de certification soit valide conformément à l'un au moins des ensembles de politiques qu'il a déterminé à l'avance; ou

b) il peut demander au module de validation d'itinéraire de lui rendre compte de l'ensemble des politiques pour lequel l'itinéraire de certification est valide.

La première stratégie peut être préférable lorsque l'utilisateur de certificat connaît a priori l'ensemble des politiques acceptables pour l'utilisation prévue.

La deuxième stratégie peut être préférable lorsque l'utilisateur de certificat ne connaît pas a priori l'ensemble des politiques acceptables pour l'utilisation prévue.

Dans le premier cas, la procédure de validation de l'itinéraire de certification indiquera que l'itinéraire est valide uniquement s'il est valide conformément à une ou plusieurs des politiques spécifiées dans l'ensemble *initial-policy-set* et renverra le sous-ensemble de l'ensemble *initial-policy-set* pour lequel l'itinéraire est valide. Dans le deuxième cas, la procédure de validation de l'itinéraire de certification peut indiquer que l'itinéraire n'est pas valide conformément à l'ensemble *initial-policy-set*, mais qu'il est valide pour un ensemble disjoint: l'ensemble *authorities-constrained-policy-set* (*ensemble de politiques imposé par des autorités*). L'utilisateur de certificat doit alors déterminer si l'emploi qu'il souhaite faire du certificat est en accord avec une ou plusieurs politiques de certificat pour lesquelles l'itinéraire est effectivement valide. L'utilisateur de certificat peut forcer la procédure à renvoyer un résultat valide conformément à toute politique (non spécifiée) en positionnant la valeur de l'ensemble *initial-policy-set* sur *any-policy*.

8.1.5 Certificats auto-émis

Une autorité de certification peut émettre un certificat à sa propre intention dans les trois cas suivants:

a) comme procédé commode pour le codage de sa clé publique à des fins de communication à ses utilisateurs de certificat et de stockage de cette clé par ces derniers;

b) pour certifier des utilisations de clés autres que la signature de certificat et de liste (par exemple, pour un horodatage);

c) pour remplacer ses certificats après expiration.

Ces types de certificat sont appelés certificats auto-émis; ils peuvent être reconnus par le fait qu'ils contiennent des noms d'émetteur et de sujet identiques. Les certificats auto-émis de type a) peuvent être vérifiés, à des fins de validation d'itinéraire, au moyen de la clé publique qu'ils contiennent et seront ignorés s'ils sont rencontrés sur l'itinéraire.

Les certificats auto-émis de type b) apparaissent exclusivement sous la forme de certificats en fin d'un itinéraire et seront traités en conséquence.

Les certificats auto-émis de type c) (appelés également certificats intermédiaires auto-émis) peuvent apparaître comme certificats intermédiaires sur un itinéraire. La procédure correcte pour une autorité de certification qui remplace une clé au moment de son expiration consiste à demander l'émission de tous les certificats croisés, engagés dans des liaisons, dont elle a besoin pour remplacer sa clé publique avant d'utiliser la nouvelle clé. Si toutefois des certificats auto-émis sont rencontrés sur l'itinéraire, ils seront traités comme des certificats intermédiaires avec l'exception suivante: ils ne contribuent pas au comptage de la longueur de l'itinéraire dans le traitement du composant **pathLenConstraint** (*contrainte de longueur d'itinéraire*) de l'extension **basicConstraints** (*contraintes de base*) et des valeurs de *skip-certificates* (*certificat ignorés*) associées aux indicateurs *policy-mapping-inhibit-pending* (*attente de mappage de politique*) et *explicit-policy-pending* (*attente de politique explicite*).

8.2 Extensions d'informations de clé et de politique

8.2.1 Expression des besoins

Les besoins suivants sont liés aux informations de clé et de politique:

a) la mise à jour d'une paire de clés d'autorité de certification peut s'effectuer à des intervalles réguliers ou dans des circonstances spéciales. Il en résulte le besoin d'un champ de certificat destiné à véhiculer un identificateur de la clé publique utilisée pour vérifier la signature du certificat. Un système utilisant des certificats peut employer de tels identificateurs pour trouver le certificat d'autorité de certification correct à des fins de validation de la clé publique de l'émetteur du certificat;

b) un sujet de certificat possède en général diverses clés publiques et de ce fait divers certificats utilisés à des fins diverses, par exemple pour les signatures numériques et le chiffrement d'un agrément de clé. Un champ de certificat est nécessaire pour permettre à un utilisateur de certificat de choisir le certificat correct pour un sujet donné et une utilisation particulière, ou pour permettre à une autorité de certification de stipuler qu'une clé certifiée peut uniquement être utilisée dans un but particulier;

c) la mise à jour d'une paire de clés de sujet peut s'effectuer à des intervalles réguliers ou dans des circonstances spéciales. Il en résulte le besoin d'un champ de certificat destiné à véhiculer un identificateur de clé publique permettant de faire la distinction entre diverses clés publiques d'un même sujet utilisées à des instants différents. Un système utilisant des certificats peut utiliser de tels identificateurs pour trouver le certificat correct;

d) la clé privée correspondant à une clé publique certifiée est utilisée en général pendant un laps de temps différent de la durée de validité de la clé publique. Dans le cas de clés de signature numérique, la durée d'utilisation de la clé privée de signature est en général plus courte que celle de la clé publique de vérification. La durée de validité du certificat indique un laps de temps pendant lequel la clé publique peut être utilisée, qui n'est pas nécessairement identique à la durée d'utilisation de la clé privée. Dans le cas où une clé privée est mise en danger, la durée de sa divulgation peut être limitée si les vérificateurs de la signature connaissent la durée d'utilisation légitime de la clé privée. Il en résulte le besoin d'une indication de la durée d'utilisation de la clé privée dans un certificat;

e) comme des certificats peuvent être utilisés dans des environnements où s'appliquent plusieurs politiques de certificat, il est nécessaire de pouvoir indiquer dans les certificats des informations de politique de certificat;

f) dans le cas d'une certification croisée entre deux organismes, on peut parfois convenir que certaines des politiques des deux organismes peuvent être considérées comme équivalentes. Un certificat d'autorité de certification doit permettre à l'émetteur du certificat d'indiquer que l'une de ces politiques de certificat est équivalente à une politique de certificat du domaine de l'autorité de certification sujette. Ce procédé est appelé mappage de politique;

g) l'utilisateur d'un système de chiffrement ou de signature numérique qui emploie des certificats définis dans la présente Spécification d'annuaire doit pouvoir déterminer à l'avance les algorithmes pris en charge par d'autres utilisateurs.

8.2.2 Champs d'extension de clé publique et de liste CRL

Les champs d'extension suivants sont définis:

a) *identificateur de clé d'autorité*;

b) *identificateur de clé de sujet*;

c) *extension d'utilisation de clé*;

d) *utilisation de clé étendue*;

e) *durée d'utilisation de clé privée*;

f) *politiques de certificat*;

g) *mappages de politique*.

Ces champs d'extension seront utilisés exclusivement comme extensions de certificat, à l'exception de l'identificateur de clé d'autorité qui peut également être utilisé comme extension de liste CRL. Ces extensions peuvent être utilisées, sauf indication contraire, pour des certificats d'autorité de certification et des certificats d'entité finale.

8.2.2.1 Extension d'identificateur de clé d'autorité

Ce champ, pouvant être utilisé comme extension de certificat ou comme extension de liste CRL, identifie la clé publique devant être utilisée pour vérifier la signature de ce certificat ou de cette liste CRL. Il permet de faire la distinction entre des clés utilisées par une même autorité de certification (par exemple lors de la mise à jour des clés). Il est défini comme suit:

```
authorityKeyIdentifier EXTENSION ::= {
    SYNTAX              AuthorityKeyIdentifier
    IDENTIFIED BY       id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier               [0] KeyIdentifier               OPTIONAL,
    authorityCertIssuer         [1] GeneralNames                OPTIONAL,
    authorityCertSerialNumber   [2] CertificateSerialNumber     OPTIONAL }
    ( WITH COMPONENTS           {..., authorityCertIssuer PRESENT,
                                authorityCertSerialNumber PRESENT} |
     WITH COMPONENTS            {..., authorityCertIssuer ABSENT,
                                authorityCertSerialNumber ABSENT} )

KeyIdentifier ::= OCTET STRING
```

La clé peut être indiquée par un identificateur de clé explicite dans le composant **keyIdentifier** (*identificateur de clé*) par l'identification d'un certificat pour la clé (indiquant l'émetteur du certificat dans le composant **authorityCertIssuer** (*autorité émettrice de certificat*) et le numéro de série de certificat dans le composant **authorityCertSerialNumber** (*numéro de série de l'autorité de certificat*) ou à la fois par un identificateur de clé explicite et l'identification d'un certificat pour la clé. L'émetteur du certificat ou de la liste CRL s'assurera de la cohérence lorsque les deux formes d'identification sont utilisées. Un identificateur de clé sera unique par rapport à tous les identificateurs de clé de l'autorité émettrice du certificat ou de la liste CRL contenant l'extension. Une implémentation qui prend en charge cette extension n'a pas l'obligation de traiter toutes les formes de nom figurant dans le composant **authorityCertIssuer** [(voir 8.3.2.1 pour les détails du type **GeneralNames** (*noms généraux*)].

Les autorités de certification attribueront des numéros de série de certificat de manière à ce que chaque couple (émetteur, numéro de série de certificat) identifie sans ambiguïté un certificat unique. La forme **keyIdentifier** peut être utilisée pour sélectionner des certificats d'autorité de certification au cours de la construction de l'itinéraire. La paire **authorityCertIssuer**, **authoritySerialNumber** ne peut être utilisée que pour donner la préférence à un certificat par rapport aux autres au moment de la construction de l'itinéraire.

Cette extension est toujours non critique.

8.2.2.2 Extension d'identificateur de clé de sujet

Ce champ indique la clé publique certifiée. Il permet de faire la distinction entre des clés utilisées par le même sujet (par exemple, lors de la mise à jour des clés). Il est défini comme suit:

```
subjectKeyIdentifier EXTENSION ::= {
    SYNTAX              SubjectKeyIdentifier
    IDENTIFIED BY       id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier
```

Un identificateur de clé sera unique par rapport à tous les identificateurs de clé du sujet pour lequel il est utilisé. Cette extension est toujours non critique.

8.2.2.3 Extension d'utilisation de clé

Ce champ indique le but de l'utilisation de la clé publique certifiée. Il est défini comme suit:

```
keyUsage EXTENSION ::= {
    SYNTAX              KeyUsage
    IDENTIFIED BY       id-ce-keyUsage }

KeyUsage ::= BIT STRING {
    digitalSignature        (0),
    nonRepudiation          (1),
    keyEncipherment         (2),
    dataEncipherment        (3),
    keyAgreement            (4),
    keyCertSign             (5),
    cRLSign                 (6),
    encipherOnly            (7),
    decipherOnly            (8) }
```

L'utilisation des bits du type **KeyUsage** (*utilisation de la clé*) est la suivante:

a) **digitalSignature** (*signature numérique*): vérification des signatures numériques pour un but autre que ceux indiqués dans les alinéas b), f) ou g) ci-dessous;

b) **nonRepudiation** (*non répudiation*): vérification des signatures numériques utilisées pour la fourniture d'un service de non-répudiation qui protège contre tout déni frauduleux d'une action quelconque effectuée par l'entité signataire (à l'exception d'une signature de certificat ou de liste CRL, comme dans les alinéas f) ou g) ci-dessous);

c) **keyEncipherment** (*chiffrement de clé*): chiffrement de clés ou d'autres informations de sécurité, par exemple pour le transport de clé;

d) **dataEncipherment** (*chiffrement de données*): chiffrement de données utilisateur, autres que des clés ou des informations de sécurité comme dans l'alinéa c) ci-dessus;

e) **keyAgreement** (*accord de clé*): utilisé comme clé d'agrément d'une clé publique;

f) **keyCertSign** (*signature de certificat de clé*): vérification de la signature d'une autorité de certification pour des certificats;

g) **cRLSign** (*signature de liste CRL*): vérification de la signature d'une autorité pour des listes CRL;

h) **encipherOnly** (*chiffrement seulement*): agrément de clé pour une clé publique utilisée exclusivement pour le chiffrement de données lorsque le bit **keyAgreement** est également positionné (la signification n'est pas définie lorsque l'autre bit d'utilisation de clé est positionné);

i) **decipherOnly** (*déchiffrement seulement*): agrément de clé pour une clé publique utilisée exclusivement pour le déchiffrement de données lorsque le bit **keyAgreement** est également positionné (la signification n'est pas définie lorsque l'autre bit d'utilisation de clé est positionné).

Le bit **keyCertSign** est utilisable exclusivement dans les certificats d'autorité de certification. Si le champ **KeyUsage** est positionné sur **keyCertSign** et si l'extension de contraintes de base est présente dans le même certificat, la valeur du composant **cA** (*autorité de certification*) de cette extension doit alors être positionnée sur "Vrai". Les autorités de certification peuvent également employer les autres utilisations définies pour les bits dans **KeyUsage**, par exemple **digitalSignature** pour fournir l'authentification et l'intégrité de transactions d'administration en ligne.

Cette extension peut être critique ou non, au choix de l'émetteur du certificat.

Si l'extension est marquée comme critique, le certificat sera alors uniquement utilisé dans un but pour lequel le bit d'utilisation de clé est positionné sur un.

Si l'extension est marquée comme non critique, elle indique alors le ou les buts prévus pour la clé et peut être utilisée pour retrouver la clé ou le certificat correct d'une entité qui possède plusieurs clés ou certificats. Si cette extension est présente et s'il reconnaît et traite le type d'extension **KeyUsage**, le système utilisant des certificats doit veiller à ce que le certificat ne soit utilisé que pour l'une des fins pour lesquelles le bit d'utilisation de clé correspondant est mis à un.. Un bit positionné sur zéro indique que la clé n'est pas prévue pour ce but. Si tous les bits sont positionnés sur zéro, ceci indique que la clé est prévue pour un autre but que ceux énumérés.

8.2.2.4 Extension d'utilisation de clé étendue

Ce champ 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. Il est défini comme suit:

```
extKeyUsage EXTENSION ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF KeyPurposeId
    IDENTIFIED BY       id-ce-extKeyUsage }

KeyPurposeId ::= OBJECT IDENTIFIER
```

Des buts d'utilisation de clé peuvent être définis par tout organisme qui en éprouve le besoin. Les identificateurs d'objet utilisés pour indiquer ces buts seront attribués conformément à la Rec. UIT-T X.660 | ISO/CEI 9834-1.

Cette extension peut être critique ou non, au choix de l'émetteur du certificat.

Si l'extension est marquée comme critique, le certificat sera alors utilisé exclusivement pour l'un des buts indiqués.

Si l'extension est marquée comme non critique, elle indique alors le ou les buts prévus pour la clé et peut être utilisée pour retrouver la clé ou le certificat correct d'une entité qui possède plusieurs clés ou certificats. Si cette extension est présente et s'il reconnaît et traite le type d'extension **extendedKeyUsage**, le système utilisateur de certificat doit veiller à ce que le certificat ne soit utilisé que pour l'une des fins (les applications utilisatrices peuvent néanmoins exiger l'indication d'un but particulier pour accepter le certificat).

Si un certificat contient à la fois un champ d'utilisation de clé critique et un champ d'extension d'utilisation de clé critique, les deux champs sont alors traités de manière indépendante et le certificat sera utilisé exclusivement dans un but cohérent avec la signification des deux champs. Le certificat ne sera pas utilisé s'il n'existe pas de but cohérent.

8.2.2.5 Extension de durée d'utilisation de clé privée

Ce champ indique la durée d'utilisation de la clé privée correspondant à la clé publique certifiée. Il s'applique exclusivement pour des clés de signature numérique. Il est défini comme suit:

```
privateKeyUsagePeriod EXTENSION ::= {
    SYNTAX              PrivateKeyUsagePeriod
    IDENTIFIED BY       id-ce-privateKeyUsagePeriod }

PrivateKeyUsagePeriod ::= SEQUENCE {
    notBefore   [0]   GeneralizedTime OPTIONAL,
    notAfter    [1]   GeneralizedTime OPTIONAL }
    ( WITH COMPONENTS {..., notBefore PRESENT} |
    WITH COMPONENTS       {..., notAfter PRESENT} )
```

Le composant **notBefore** (*pas avant*) indique la date et l'heure au plus tôt à partir desquelles la clé privée peut être utilisée pour une signature. Si le composant **notBefore** n'est pas présent, aucune information n'est alors fournie en ce qui concerne le début de la durée de validité de l'utilisation de la clé privée. Le composant **notAfter** (*pas après*) indique la date et l'heure au plus tard jusqu'auxquelles la clé privée peut être utilisée pour une signature. Si le composant **notAfter** n'est pas présent, aucune information n'est alors fournie en ce qui concerne la fin de la durée de validité de l'utilisation de la clé privée.

Cette extension est toujours non critique.

NOTE 1 – La durée de validité d'utilisation de la clé privée peut différer de celle du certificat de la clé publique telle qu'elle est indiquée par la durée de validité du certificat. Dans le cas de clés de signature numérique, la durée d'utilisation des clés privées de signature est en général plus courte que celle de la clé publique de vérification.

NOTE 2 – Un certificat valide doit encore exister pour la clé à l'instant de la vérification si le vérificateur d'une signature numérique veut vérifier que la clé n'a pas été révoquée avant l'instant de la vérification, par exemple en raison d'une mise en danger de la clé. Un vérificateur ne peut plus faire confiance aux listes CRL pour la notification d'une mise en danger après l'expiration du, ou des certificats pour une clé publique.

8.2.2.6 Extension de politiques de certificat

Ce champ donne la liste des politiques de certificat reconnues par l'autorité de certification émettrice qui s'appliquent au certificat, ainsi que les informations du qualificatif optionnel concernant ces politiques de certificat. La liste des politiques de certificat est utilisée pour déterminer la validité d'un itinéraire de certification, comme décrit à l'article 10. Les qualificatifs optionnels ne sont pas utilisés dans la procédure de traitement de l'itinéraire de certification, mais des qualificatifs pertinents sont fournis, en sortie de ce processus, à l'application utilisant le certificat de manière à faciliter la décision concernant la validité d'un itinéraire pour une transaction donnée. Diverses politiques de certificat concernent en général des applications diverses pouvant utiliser la clé certifiée. La présence de cette extension dans un certificat d'entité finale indique les politiques de certificat pour lesquelles ce certificat est valide. La présence de cette extension dans un certificat émis par une autorité de certification destiné à une autre autorité de certification indique les politiques de certificat pour lesquelles des itinéraires contenant ce certificat peuvent être valides. Ce champ est défini comme suit:

```
certificatePolicies EXTENSION ::= {
    SYNTAX              CertificatePoliciesSyntax
    IDENTIFIED BY       id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier    CertPolicyId,
    policyQualifiers    SEQUENCE SIZE (1..MAX) OF
                            PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId   CERT-POLICY-QUALIFIER.&id
                        ({SupportedPolicyQualifiers}),
    qualifier           CERT-POLICY-QUALIFIER.&Qualifier
                        ({SupportedPolicyQualifiers}{@policyQualifierId})
                        OPTIONAL }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }
```

Une valeur du type **PolicyInformation** (*informations de politique*) identifie et véhicule des informations de qualificatif pour une seule politique de certificat. Le composant **policyIdentifier** (*identificateur de politique*) contient l'identificateur d'une politique de certificat et le composant **policyQualifiers** (*qualificatifs de politique*) des valeurs de qualificatifs pour cet élément.

Cette extension peut être critique ou non, au choix de l'émetteur du certificat.

Si l'extension est marquée comme critique, ceci indique alors que le certificat sera utilisé uniquement pour les buts et conformément aux règles impliquées par l'une des politiques de certificat indiquées. Les règles d'une politique particulière peuvent exiger que le système utilisant des certificats traite d'une manière particulière la valeur du qualificatif.

Si l'extension est marquée comme non critique, son utilisation ne limite pas nécessairement l'utilisation du certificat aux politiques figurant dans la liste. Un utilisateur de certificat peut toutefois exiger la présence d'une politique particulière pour utiliser le certificat (voir article 10). Les qualificatifs de politique peuvent être traités ou ignorés au choix de l'utilisateur de certificat.

Des types de politique de certificat et de qualificatif de politique de certificat peuvent être définis par tout organisme qui en éprouve le besoin. Les identificateurs d'objet utilisés pour indiquer ces types de politique de certificat et de qualificatif de politique de certificat seront attribués conformément à la Rec. CCITT X.660 | ISO/CEI 9834-1. Une autorité de certification peut faire confiance à un certificat pour toutes les politiques possibles en utilisant l'identificateur **anyPolicy** (*toute politique*). Cet identificateur d'objet est attribué dans la présente Spécification, du fait qu'il est nécessaire d'indiquer l'utilisation de cette valeur spéciale indépendamment de l'application ou de l'environnement. Aucun identificateur d'objet ne sera attribué dans la présente Spécification pour des politiques de certificat particulières. Cette attribution est de la responsabilité de l'entité qui définit la politique de certificat.

```
anyPolicy       OBJECT IDENTIFIER  ::=          { 2 5 29 32 0 }
```

Aucun qualificatif de politique associé ne doit exister pour l'identificateur **anyPolicy**.

La classe d'objets ASN.1 suivante sert à définir les types de qualificatif de certificat de politique:

```
CERT-POLICY-QUALIFIER ::= CLASS {
    &id                 OBJECT IDENTIFIER UNIQUE,
    &Qualifier          OPTIONAL }
WITH SYNTAX {
    POLICY-QUALIFIER-ID     &id
    [QUALIFIER-TYPE     &Qualifier] }
```

La définition d'un type de qualificatif de politique aura le contenu suivant:

- une déclaration de la sémantique des valeurs possibles;
- une indication de la présence possible de l'identificateur de qualificatif dans une extension de politiques de certificat sans valeur associée et de la sémantique impliquée dans un tel cas.

NOTE – Un qualificatif peut être spécifié avec un type ASN.1 quelconque. La spécification du type **OCTET STRING** (*chaîne d'octets*) est recommandée lorsque son utilisation principale est prévue pour des applications qui ne possèdent pas de fonctions de décodage ASN.1. La valeur ASN.1 **OCTET STRING** peut alors véhiculer une valeur de qualificatif codée conformément à toute convention spécifiée par l'organisme qui définit l'élément de politique.

8.2.2.7 Extensions de mappages de politique

Ce champ sera utilisé exclusivement dans des certificats d'autorité de certification; il permet à un émetteur de certificat d'indiquer que, pour les besoins de l'utilisateur d'un itinéraire de certification contenant ce certificat, l'une des politiques de certificat de l'émetteur peut être considérée comme équivalente à une autre politique de certificat utilisée dans le domaine de l'autorité de certification sujette. Ce champ est défini comme suit:

```
policyMappings EXTENSION ::= {
    SYNTAX              PolicyMappingsSyntax
    IDENTIFIED BY       id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    issuerDomainPolicy      CertPolicyId,
    subjectDomainPolicy     CertPolicyId }
```

Le composant **issuerDomainPolicy** (*politique du domaine de l'émetteur*) indique une politique de certificat reconnue dans le domaine de l'autorité de certification et pouvant être considérée comme équivalente à la politique de certificat indiquée dans le composant **subjectDomainPolicy** (*politique du domaine du sujet*) et reconnue dans le domaine de l'autorité de certification sujette.

Les politiques ne seront pas mappées, dans un sens ou dans l'autre, avec la valeur spéciale **anyPolicy**.

Cette extension peut être critique ou non, au choix de l'émetteur du certificat. Il est recommandé qu'elle soit critique, faute de quoi il est possible qu'un utilisateur de certificat n'interprète pas correctement la stipulation de l'autorité de certification émettrice.

NOTE 1 – L'exemple suivant décrit un mappage de politique. Le gouvernement des Etats-Unis peut avoir une politique appelée "Commerce avec le Canada" et le gouvernement canadien une politique appelée "Commerce avec les Etats-Unis". Ces deux politiques sont identifiées et définies de manière distincte, mais il peut exister un accord entre les deux gouvernements pour accepter, à des fins adéquates, des itinéraires de certification s'étendant au-delà de la frontière pour des règles impliquées par ces politiques.

NOTE 2 – Le mappage de politique entraîne une charge administrative significative et l'intervention d'un personnel actif et autorisé pour les prises de décision connexes. Il est en général préférable de convenir d'une utilisation plus globale de politiques communes que d'appliquer un mappage de politique. Dans l'exemple précédent, il est préférable pour les Etats-Unis, le Canada et le Mexique de convenir d'une politique commune de commerce pour l'Amérique du Nord.

NOTE 3 – Il est prévu que le mappage de politique sera uniquement praticable dans des environnements limités utilisant des déclarations de politique très simples.

8.3 Extensions d'information de sujet et d'émetteur

8.3.1 Expression des besoins

Les besoins suivants sont liés aux attributs de sujet et d'émetteur de certificat:

a) Les certificats doivent être utilisables par des applications qui emploient diverses formes de nom, à savoir les noms de messagerie électronique Internet, les noms de domaines Internet, les adresses d'origine et de réception X.400 et les noms de participants d'échange EDI. Il est de ce fait nécessaire d'associer de manière fiable un sujet de certificat ou un émetteur de certificat ou de liste CRL avec des noms multiples de formes diverses;

b) Un utilisateur de certificat peut avoir besoin de connaître de manière fiable certaines informations d'identification d'un sujet pour avoir la certitude que le sujet est effectivement la personne ou l'entité prévue. Il est possible d'exiger, par exemple, des informations telles que l'adresse postale, la fonction dans l'entreprise ou une photographie. De telles informations peuvent être représentées de manière commode par des attributs qui ne font pas nécessairement partie du nom distinctif. Il s'ensuit qu'il est nécessaire de disposer d'un champ de certificat permettant de véhiculer des attributs d'annuaire autres que ceux figurant dans le nom distinctif.

8.3.2 Champs d'extension de certificat et de liste CRL

Les champs d'extension suivants sont définis:

a) *Autre nom de sujet*;

b) *Autre nom d'émetteur*;

c) *Attributs d'annuaire du sujet*.

Ces champs seront utilisés exclusivement comme extensions de certificat, à l'exception de l'autre nom d'émetteur qui peut également être utilisé comme extension de liste CRL. Ils peuvent être présents comme extensions de certificat dans des certificats d'autorité de certification ou des certificats d'entité finale.

8.3.2.1 Extension d'autre nom de sujet

Ce champ contient une ou plusieurs variantes de nom, utilisant diverses formes de nom, pour l'entité qui est liée à la clé publique certifiée par l'autorité de certification. Il est défini comme suit:

```
subjectAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-subjectAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName                   [0]     INSTANCE OF OTHER-NAME,
    rfc822Name                  [1]     IA5String,
    dNSName                     [2]     IA5String,
    x400Address                 [3]     ORAddress,
    directoryName               [4]     Name,
    ediPartyName                [5]     EDIPartyName,
    uniformResourceIdentifier   [6]     IA5String,
    iPAddress                   [7]     OCTET STRING,
    registeredID                [8]     OBJECT IDENTIFIER }
```

```
OTHER-NAME ::= TYPE-IDENTIFIER

EDIPartyName ::= SEQUENCE {
    nameAssigner        [0]     DirectoryString {ub-name} OPTIONAL,
    partyName           [1]     DirectoryString {ub-name} }
```

Les valeurs des variantes dans le type **GeneralName** (*nom général*) sont des noms avec les diverses formes suivantes:

- **otherName** (*autre nom*) est un nom de forme quelconque défini comme une instance de la classe d'objets d'information **OTHER-NAME**;
- **rfc822Name** est une adresse de messagerie électronique Internet définie conformément au document Internet RFC 822;
- **dNSName** est un nom de domaine Internet défini conformément au document Internet RFC 1035;
- **x400Address** est une adresse O/R définie conformément à la Rec. UIT-T X.411 | ISO/CEI 10021-4;
- **directoryName** est un nom d'annuaire défini conformément à la Rec. UIT-T X.501 | ISO/CEI 9594-2;
- **ediPartyName** est un nom dont la forme a fait l'objet d'un accord entre des partenaires d'échange de données EDI qui souhaitent communiquer; le composant **nameAssigner** indique une autorité qui attribue des valeurs de nom uniques dans le composant **partyName**;
- **uniformResourceIdentifier** est un identificateur de ressource uniforme pour le réseau web mondial, défini conformément au document Internet RFC 1630;
- **iPAddress** est une adresse de protocole Internet définie conformément au document Internet RFC 791, se présentant sous la forme d'une chaîne de bits;
- **registeredID** est un identificateur de tout objet enregistré, attribué conformément à la Rec. CCITT X.660 | ISO/CEI 9834-1.

Il existera, pour chaque forme de nom utilisée dans le type **GeneralName**, un système d'enregistrement de nom garantissant que tout nom utilisé désigne de manière non ambiguë une entité unique pour l'émetteur et les utilisateurs du certificat.

Cette extension peut être critique ou non, au choix de l'émetteur du certificat. Une implémentation qui prend en charge cette extension n'a pas l'obligation de traiter toutes les formes de nom. Si l'extension est marquée comme critique, l'une au moins des formes de nom présentes sera reconnue et traitée, faute de quoi le certificat sera considéré comme non valide. A l'exception précédente près, un système utilisant des certificats peut ignorer tout nom dont la forme n'est pas reconnue ou prise en charge. Il est recommandé que ce champ soit marqué comme non critique, dans la mesure où le champ de sujet du certificat contient un nom d'annuaire qui identifie le sujet de manière non ambiguë.

NOTE 1 – L'utilisation de la classe **TYPE-IDENTIFIER** (*identificateur de type*) est décrite dans les Annexes A et C de la Rec. UIT-T X.681 | ISO/CEI 8824-2.

NOTE 2 – Si ce champ d'extension est présent et marqué comme critique, le champ **subject** du certificat peut contenir un nom vide (par exemple une succession de longueur nulle de noms distinctifs relatifs), auquel cas le sujet est identifié uniquement par le ou les noms de cette extension.

8.3.2.2 Extension d'autre nom d'émetteur

Ce champ contient un ou plusieurs autres noms de l'émetteur du certificat ou de la liste CRL, utilisant divers formes de nom. Il est défini comme suit:

```
issuerAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-issuerAltName }
```

Cette extension peut être critique ou non, au choix de l'émetteur du certificat ou de la liste CRL. Une implémentation qui prend en charge cette extension n'a pas l'obligation de traiter toutes les formes de nom. Si l'extension est marquée comme critique, l'une au moins des formes de nom présents sera reconnue et traitée, faute de quoi le certificat ou la liste CRL sera considéré comme non valide. A part l'exception précédente près, un système utilisant des certificats peut ignorer tout nom dont la forme n'est pas reconnue ou prise en charge. Il est recommandé que ce champ soit marqué comme non critique, dans la mesure où le champ d'émetteur du certificat ou de la liste CRL contient un nom d'annuaire qui identifie le sujet de manière non ambiguë.

NOTE – Si ce champ d'extension est présent et marqué comme critique, le champ **issuer** du certificat ou de la liste CRL peut contenir un nom vide (par exemple une succession de longueur nulle de noms distinctifs relatifs), auquel cas le sujet est identifié uniquement par le ou les noms de cette extension.

8.3.2.3 Extension d'attributs d'annuaire du sujet

Ce champ véhicule toute valeur d'attribut d'annuaire souhaitée pour le sujet du certificat. Il est défini comme suit:

```
subjectDirectoryAttributes EXTENSION ::= {
    SYNTAX              AttributesSyntax
    IDENTIFIED BY       id-ce-subjectDirectoryAttributes }

AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute
```

Cette extension est toujours non critique.

Si cette extension est présente dans un certificat de clé publique, certaines des extensions définies à l'article 15 peuvent également être présentes.

8.4 Extensions de contrainte d'itinéraire de certification

8.4.1 Expression des besoins

Les besoins suivants sont liés au traitement de l'itinéraire de certification:

a) les certificats d'entité finale doivent pouvoir être distingués des certificats d'autorité de certification pour interdire que des entités finales s'établissent elles-mêmes comme autorité de certification d'une manière non autorisée. Une autorité de certification doit également avoir la possibilité d'imposer une limite à la taille de la suite d'une chaîne en provenance d'une autorité de certification sujette certifiée, par exemple pas plus d'un certificat ou pas plus de deux certificats;

b) une autorité de certification doit avoir la possibilité de spécifier des contraintes permettant à un utilisateur de certificat de vérifier que les autorités de certification moins fiables dans un itinéraire de certification (c'est-à-dire les autorités de certification se trouvant en aval sur l'itinéraire de certification partant de l'autorité de certification qui emploie la clé publique de l'utilisateur de certificat) ne mettent pas en danger la sécurité en émettant des certificats pour des sujets appartenant à un espace de noms qui n'est pas autorisé. Le respect de ces contraintes doit pouvoir être vérifié de manière automatique par l'utilisateur de certificat;

c) le traitement de l'itinéraire de certification doit pouvoir être implémenté dans un module automatisé autonome. Ceci est nécessaire pour permettre l'implémentation de modules matériels ou logiciels fiables fournissant des fonctions de traitement de l'itinéraire de certification;

d) il doit être possible d'implémenter le traitement de l'itinéraire de certification indépendamment de toute interaction en temps réel avec l'utilisateur local;

e) il doit être possible d'implémenter le traitement de l'itinéraire de certification sans devoir faire confiance à l'utilisation d'informations de bases de données locales de description de politique (certaines informations locales fiables, par exemple une clé publique initiale, sont nécessaires au minimum pour le traitement de l'itinéraire de certification, mais la quantité de ces informations doit être réduite à un minimum);

f) les itinéraires de certification doivent pouvoir fonctionner dans des environnements dans lesquels plusieurs politiques de certificat sont reconnues. Une autorité de certification doit pouvoir stipuler quelles sont les autorités de certification, situées dans d'autres domaines, auxquelles elle accorde sa confiance et pour quels buts. Il doit être possible de concaténer des domaines de politique;

g) une flexibilité totale est requise pour les modèles de confiance. Un modèle hiérarchique strict, convenant à une organisation donnée, ne convient pas aux besoins d'entreprises multiples interconnectées. La flexibilité est également nécessaire dans le choix de la première autorité de certification fiable d'un itinéraire de certification. Il doit en particulier être possible de prescrire que l'itinéraire de certification parte du domaine de sécurité local du système utilisateur de clé publique;

h) les structures de dénomination ne doivent pas être soumises à la contrainte d'utilisation de noms dans les certificats. En d'autres termes, les structures de nom d'annuaire considérées comme naturelles pour certaines organisations ou zones géographiques ne doivent pas nécessiter d'adaptation pour tenir compte des prescriptions édictées par les autorités de certification;

i) les champs d'extension des certificats n'ont pas besoin d'être compatibles en amont avec la démarche d'itinéraire de certification sans contrainte telle qu'elle est spécifiée dans des éditions antérieures de la Rec. UIT-T X.509 | ISO/CEI 9594-8;

j) une autorité de certification doit pouvoir interdire l'utilisation du mappage de politique et doit pouvoir prescrire la présence d'identificateurs explicites de politique de certificat pour les certificats suivants sur un itinéraire de certification.

NOTE – Le traitement d'un itinéraire de certification dans un système utilisant des certificats nécessite un niveau de confiance adéquat. La présente Spécification d'annuaire définit des fonctions utilisables dans des implémentations dont on exige la conformité à des déclarations d'assurance spécifiques. Une prescription d'assurance peut, par exemple, déclarer que l'itinéraire de certification doit toujours être protégé contre une manipulation du processus (telle que l'intrusion logicielle ou la modification de données). Le niveau d'assurance doit être compatible avec le risque commercial. Par exemple:

– un traitement interne par un module de chiffrement adéquat peut être requis pour des clés publiques utilisées afin de valider des transferts de fonds de montant élevé;

– alors qu'un traitement logiciel peut convenir pour des demandes d'état de compte bancaire à domicile.

Il en résulte que les fonctions de traitement du chemin de certification doivent pouvoir être implémentées dans des modules de chiffrement matériels ou dans des jetons chiffrés, pour ne citer que cette option.

k) une autorité de certification doit pouvoir empêcher que la valeur spéciale "any-policy" (toute politique) soit considérée comme une politique valide dans les certificats subséquents d'un chemin de certification.

8.4.2 Champs d'extension de certificat

Les champs d'extension suivants sont définis:

a) *contraintes de base*;

b) *contraintes de nom*;

c) *contraintes de politique*;

d) *interdiction de toute politique*.

Ces champs d'extension seront utilisés exclusivement comme extension de certificat. Les contraintes de nom et de politique ne doivent être utilisées que dans des certificats d'autorité de certification. Les contraintes de base peuvent également être utilisées dans des certificats d'entité finale. L'Annexe G fournit des exemples d'utilisation de ces contraintes.

8.4.2.1 Extension de contraintes de base

Ce champ indique si le sujet peut jouer le rôle d'une autorité de certification, la clé publique certifiée étant utilisée pour vérifier des signatures de certificat. Dans ce cas, il est également possible de spécifier une contrainte de longueur d'itinéraire de certification. Ce champ est défini comme suit:

```
basicConstraints EXTENSION ::= {
    SYNTAX              BasicConstraintsSyntax
    IDENTIFIED BY       id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
    cA                      BOOLEAN DEFAULT FALSE,
    pathLenConstraint       INTEGER (0..MAX) OPTIONAL }
```

Le composant **cA** indique si la clé publique certifiée peut être utilisée pour vérifier des signatures de certificat.

Le composant **pathLenConstraint** sera présent uniquement si le composant **cA** est positionné sur "Vrai". Il indique le nombre maximal de certificats d'autorité de certification qui peuvent venir à la suite de ce certificat dans un itinéraire de certification. La valeur 0 indique que l'entité titulaire de ce certificat peut uniquement émettre des certificats pour des entités finales et non pour d'autres autorités de certification. Si aucun champ **pathLenConstraint** ne figure dans aucun des certificats d'un itinéraire de certification, aucune limite n'est alors imposée à la longueur autorisée du chemin de certification.

Cette extension peut être critique ou non, au choix de l'émetteur du certificat. Il est recommandé qu'elle soit critique, faute de quoi une entité non autorisée comme autorité de certification pourrait émettre des certificats et un système utilisateur de certificat pourrait faire usage d'un tel certificat par inadvertance.

Si cette extension est présente et étiquetée comme critique, ou si elle est marquée comme étant non critique mais est reconnue par le système utilisateur de certificat:

– si la valeur du composant **cA** n'est pas positionnée sur "Vrai", la clé publique certifiée ne sera alors pas utilisée pour vérifier une signature de certificat;

– si la valeur du composant **cA** est positionnée sur "Vrai" et si le composant **pathLenConstraint** est présent, le système utilisateur du certificat doit alors vérifier que l'itinéraire de certification en cours de traitement est compatible avec la valeur du composant **pathLenConstraint**.

NOTE 1 – Si cette extension n'est pas présente ou si elle est marquée comme non critique et n'est pas reconnue par un système utilisateur de certificat, le certificat doit alors être considéré comme un certificat d'entité finale et ne peut pas être utilisé pour vérifier des signatures de certificat.

NOTE 2 – L'émetteur peut inclure ce champ d'extension avec une valeur de **SEQUENCE** vide pour limiter un sujet de certificat à un rôle d'entité finale, c'est-à-dire lui interdire le rôle d'autorité de certification.

8.4.2.2 Extension de contraintes de nom

Ce champ, qui sera utilisé uniquement dans un certificat CA, indique un espace de noms auquel doivent appartenir tous les noms de sujet figurant dans les certificats suivants d'un itinéraire de certification. Ce champ est défini comme suit:

```
nameConstraints EXTENSION ::= {
    SYNTAX              NameConstraintsSyntax
    IDENTIFIED BY       id-ce-nameConstraints }

NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees   [0]     GeneralSubtrees OPTIONAL,
    excludedSubtrees    [1]     GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
    base                GeneralName,
    minimum     [0]     BaseDistance DEFAULT 0,
    maximum     [1]     BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)
```

Les composants **permittedSubtrees** (*sous-arbres autorisés*) et **excludedSubtrees** (*sous-arbres interdits*) éventuellement présents spécifient chacun un ou plusieurs sous-arbres de dénomination, définis par le nom de leur racine et, de manière optionnelle, par un domaine indiqué par les niveaux supérieurs ou inférieurs dans chaque sous-arbre. Si le composant **permittedSubtrees** est présent, alors parmi tous les certificats émis par l'autorité de certification sujette et par les autorités de certification suivantes sur l'itinéraire de certification, seuls sont acceptables ceux dont les noms d'entité appartiennent à ces sous-arbres. Si le composant **excludedSubtrees** est présent, alors tout certificat émis par l'autorité de certification sujette et les autorités de certification suivantes sur l'itinéraire de certification n'est pas acceptable si le nom d'entité appartient à l'un de ces sous-arbres. La déclaration d'exclusion a priorité si les deux composants **permittedSubtrees** et **excludedSubtrees** sont présents et que les espaces de nom se chevauchent.

Parmi les formes de nom disponibles dans le type **GeneralName**, seules celles qui possèdent une structure hiérarchique bien définie sont utilisables dans ces champs. La forme de nom **directoryName** satisfait à cette prescription. Lorsqu'elle est utilisée, un sous-arbre de dénomination correspond à un sous-arbre d'un arbre DIT. Les implémentations conformes n'ont pas l'obligation de reconnaître toutes les formes de nom possibles. Si l'extension est marquée comme critique et si une implémentation utilisant des certificats ne reconnaît pas une forme de nom utilisée dans un composant **base** quel qu'il soit, le certificat sera alors traité comme si une extension critique non reconnue avait été rencontrée. Si l'extension est marquée comme non critique et qu'une implémentation utilisant des certificats ne reconnaît pas une forme de nom utilisée dans un composant **base** quel qu'il soit, cette spécification de sous-arbre peut alors être ignorée. Lorsqu'un sujet de certificat possède plusieurs noms avec la même forme (y compris, dans le cas de la forme **directoryName**, le nom figurant éventuellement dans le champ de sujet du certificat), tous les noms répondant à ces conditions feront alors l'objet d'une vérification de cohérence avec une contrainte de nom de cette forme de nom.

NOTE – Lors de la vérification des noms de sujet de certificat en vue de leur cohérence avec une contrainte de nom, les noms figurant dans des extensions non critiques d'autres noms de sujet seront traités et non ignorés.

Le champ **minimum** spécifie la borne supérieure du domaine contenu dans le sous-arbre. Tous les noms dont le composant de nom final est supérieur au niveau spécifié n'appartiennent pas au domaine. Une valeur de **minimum** égale à zéro (valeur par défaut) correspond à la base, c'est-à-dire au nœud sommital du sous-arbre. Si, par exemple, la valeur de **minimum** est positionnée sur un, le sous-arbre de dénomination ne contient pas le nœud de base mais contient ses nœuds subordonnés.

Le champ **maximum** spécifie la borne inférieure du domaine contenu dans le sous-arbre. Tous les noms dont le dernier composant est inférieur au niveau spécifié n'appartiennent pas au domaine. Une valeur de **maximum** égale à zéro correspond à la base, c'est-à-dire au sommet du sous-arbre. L'absence du composant **maximum** indique qu'aucune limite inférieure n'est imposée au domaine dans le sous-arbre. Si, par exemple, la valeur de **maximum** est positionnée sur un, le sous-arbre de dénomination ne contient aucun nœud à l'exception de la base du sous-arbre et de ses subordonnés immédiats.

Cette extension peut être critique ou non, au choix de l'émetteur du certificat. Il est recommandé qu'elle soit critique, faute de quoi un utilisateur de certificat peut omettre de vérifier que les certificats suivants sur un itinéraire de certification appartiennent à l'espace de noms prévu par l'autorité de certification émettrice.

Si cette extension est présente et marquée comme critique, ou si elle est marquée étant non critique mais est reconnue par le système utilisateur de certificat, un système utilisant des certificats vérifiera alors que l'itinéraire de certification en cours de traitement est cohérent avec la valeur figurant dans cette extension.

8.4.2.3 Extension de contraintes de politique

Ce champ spécifie des contraintes qui peuvent exiger une identification explicite de certificat de politique ou interdire le mappage de politique pour la suite de l'itinéraire de

```
policyConstraints EXTENSION ::= {
    SYNTAX              PolicyConstraintsSyntax
    IDENTIFIED BY       id-ce-policyConstraints }

PolicyConstraintsSyntax ::= SEQUENCE {
    requireExplicitPolicy       [0]   SkipCerts OPTIONAL,
    inhibitPolicyMapping        [1]   SkipCerts OPTIONAL }

SkipCerts ::= INTEGER (0..MAX)
```

Si le composant **requireExplicitPolicy** est présent et si l'itinéraire de certification contient un certificat émis par une autorité nommée, il est nécessaire que tous les certificats de l'itinéraire contiennent, dans l'extension de politiques de certificat, un identificateur de politique acceptable. Un identificateur de politique acceptable est celui qui est exigé par l'utilisateur de l'itinéraire de certification, ou l'identificateur d'une politique qui a été déclarée comme équivalente à l'une de ces politiques par un mappage de politique, ou la valeur spéciale *any-policy*. L'autorité de certification nommée est, soit l'autorité de certification qui a émis le certificat contenant cette extension (si la valeur du composant **requireExplicitPolicy** est nulle), soit une autorité de certification qui est le sujet d'un certificat suivant sur l'itinéraire de certification (indiqué par une valeur non nulle).

Si le composant **inhibitPolicyMapping** (*interdiction de mappage de politique*) est présent, il indique alors que le mappage de politique est interdit pour tous les certificats à partir d'une autorité de certification nommée dans l'itinéraire de certification jusqu'à la fin de cet itinéraire. L'autorité de certification nommée est, soit l'autorité de certification sujette du certificat contenant cette extension (si la valeur de **inhibitPolicyMapping** est nulle), soit une autorité de certification qui est le sujet d'un certificat suivant sur l'itinéraire de certification (indiqué par une valeur non nulle).

Une valeur du type **SkipCerts** (*certificats ignorés*) indique le nombre de certificats à ignorer sur l'itinéraire de certification avant qu'une contrainte prenne effet.

Cette extension peut être critique ou non, au choix de l'émetteur du certificat. Il est recommandé qu'elle soit critique, faute de quoi un utilisateur de certificat peut interpréter de manière incorrecte la stipulation de l'autorité de certification émettrice.

8.4.2.4 Extension d'inhibition de la valeur spéciale "toute politique"

Ce champ spécifie une contrainte indiquant que la valeur spéciale *any-policy* n'est pas prise en considération pour une correspondance explicite avec d'autres politiques de certificat pour tous les certificats sur l'itinéraire de certification à partir d'une autorité de certification nommée. L'autorité de certification nommée est, soit l'autorité de certification sujette du certificat contenant cette extension [si la valeur du composant **inhibitAnyPolicy** (*interdiction de toute politique*) est nulle], soit une autorité de certification qui est le sujet d'un certificat suivant dans l'itinéraire de certification (indiqué par une valeur non nulle).

```
inhibitAnyPolicy        EXTENSION   ::= {
    SYNTAX              SkipCerts
IDENTIFIED BY           id-ce-inhibitAnyPolicy }
```

Cette extension peut être critique ou non, au choix de l'émetteur du certificat. Il est recommandé qu'elle soit critique, faute de quoi un utilisateur de certificat peut interpréter de manière incorrecte la stipulation de l'autorité de certification émettrice.

8.5 Extensions de liste CRL de base

8.5.1 Expression des besoins

Les besoins suivants sont liés aux listes CRL:

a) les utilisateurs de certificat doivent pouvoir retrouver la trace de toutes les listes CRL issues d'un émetteur de liste CRL ou d'un point de répartition de listes CRL (voir 8.6) et détecter l'absence d'une liste CRL dans la séquence. Il est donc nécessaire d'attribuer des numéros de séquence aux listes CRL;

b) certains utilisateurs de liste CRL peuvent souhaiter réagir de manière différente à une révocation en fonction de son motif. Il est donc nécessaire qu'un élément de liste CRL puisse indiquer le motif de la révocation;

c) une autorité doit pouvoir suspendre de manière temporaire la validité d'un certificat et le révoquer ou le réinstaller ultérieurement. Les raisons possibles d'une telle action sont les suivantes:

   – souhait de limiter les responsabilités en cas de révocation erronée non authentifiée pour laquelle il n'existe pas d'informations adéquates permettant d'en déterminer la validité;

   – autres besoins de l'entreprise, tels que l'inactivation temporaire d'un certificat d'une entité en cours d'audit ou d'investigation.

d) une liste CRL contient, pour chaque certificat révoqué, la date d'expédition de la révocation par l'autorité. D'autres informations éventuellement connues, telles que l'instant de la mise en danger réelle ou présumée de la clé peuvent être importantes pour un utilisateur de certificat. La date de révocation ne suffit pas pour la résolution de certains litiges parce que, dans le cas le plus défavorable, toutes les signatures émises pendant la durée de validité du certificat doivent être considérées comme non valides. Il est toutefois important, pour un utilisateur, qu'un document signé soit considéré comme valide même si la clé utilisée pour la signature du message a été mise en danger après la production de la signature. Une liste CRL peut contribuer à la solution de ce problème si elle contient une deuxième date indiquant à partir de quel instant il a été établi ou présumé que la clé privée a été mise en danger;

e) les utilisateurs de certificat doivent pouvoir retrouver, à partir de la liste CRL proprement dite, des informations supplémentaires incluant le domaine d'application des certificats de cette liste, l'ordre des notifications de révocation et les ensembles de listes au sein desquels le numéro de liste CRL est unique;

f) les utilisateurs doivent pouvoir modifier de manière dynamique le découpage de listes CRL et fournir aux utilisateurs de certificat la référence des listes CRL concernées lorsque le découpage a été modifié;

g) des listes CRL delta peuvent également être disponibles pour la mise à jour d'une liste CRL de base. Les utilisateurs de certificat doivent pouvoir déterminer, à partir d'une liste CRL donnée, l'existence éventuelle de listes CRL delta, l'endroit où elles se trouvent et la date d'émission de la prochaine liste CRL delta.

8.5.2 Champs d'extension de liste CRL et d'élément de liste

Les champs d'extension suivants sont définis:

a) *numéro de liste CRL*;

b) *code motif*;

c) *code d'instruction de maintien*;

d) *date de non validité*;

e) *domaine d'application de la liste CRL*;

f) *référence de statut*;

g) *identificateur de flux de liste CRL*;

h) *liste ordonnée*;

i) *informations de delta*.

Les champs numéro de liste CRL, référence de statut, identificateur de flux de liste CRL, liste ordonnée et informations de delta seront utilisés comme champs d'extension de liste CRL; les autres champs seront utilisés exclusivement comme champs d'extension d'élément de liste CRL.

8.5.2.1 Extension de numéro de liste CRL

Ce champ d'extension de liste CRL véhicule un numéro de séquence uniformément croissant pour chaque liste CRL émise par un émetteur de liste CRL donné utilisant un attribut d'annuaire d'autorité donné ou un point de répartition de liste CRL donné. Il permet à un utilisateur de liste CRL de détecter s'il a également reçu et traité des listes CRL émises avant celle qui est en cours de traitement. Ce champ est défini comme suit:

```
cRLNumber EXTENSION ::= {
    SYNTAX              CRLNumber
    IDENTIFIED BY       id-ce-cRLNumber }

CRLNumber ::= INTEGER (0..MAX)
```

Cette extension est toujours non critique.

8.5.2.2 Extension de code motif

Ce champ d'extension d'élément de liste CRL indique un motif de révocation de certificat. Le code motif peut être utilisé par les applications pour décider comment réagir à l'expédition de révocations, en fonction de leur politique locale. Ce champ est défini comme suit:

```
reasonCode EXTENSION ::= {
    SYNTAX              CRLReason
    IDENTIFIED BY       id-ce-reasonCode }

CRLReason ::= ENUMERATED {
    unspecified             (0),
    keyCompromise           (1),
    cACompromise            (2),
    affiliationChanged      (3),
    superseded              (4),
    cessationOfOperation    (5),
    certificateHold         (6),
    removeFromCRL           (8),
    privilegeWithdrawn      (9),
    aaCompromise            (10) }
```

Les codes motif suivants indiquent la raison de la révocation d'un certificat:

- le code **keyCompromise** (*clé mise en danger*) est utilisé pour révoquer un certificat d'entité finale; il indique qu'il est établi ou présumé que la clé privée du sujet, ou d'autres caractéristiques du sujet validées dans le certificat ont été mises en danger;
- le code **cACompromise** (*autorité de certification mise en danger*) est utilisé pour révoquer un certificat d'autorité de certification; il indique qu'il est établi ou présumé que la clé privée du sujet, ou d'autres caractéristiques du sujet validées dans le certificat ont été mises en danger;
- le code **affiliationChanged** (*changement d'affiliation*) indique que le nom du sujet ou d'autres informations figurant dans le certificat ont été modifiées, mais qu'il n'y a aucune raison de présumer que la clé privée ait été mise en danger;
- le code **superseded** (*remplacé*) indique que le certificat a été remplacé, mais qu'il n'y a aucune raison de présumer que la clé privée ait été mise en danger;
- le code **cessationOfOperation** (*fin d'opération*) indique que le certificat n'est plus nécessaire dans le but pour lequel il a été émis, mais qu'il n'y a aucune raison de présumer que la clé privée a été mise en danger;
- le code **privilegeWithdrawn** (*privilège retiré*) indique qu'un certificat (clé publique ou certificat d'attribut) a été révoqué parce qu'un privilège qu'il contenait a été retiré;
- le code **aACompromise** (*autorité d'attribut mise en danger*) indique qu'il est établi ou présumé que des caractéristiques de l'autorité d'attribut validées dans le certificat d'attribut ont été mises en danger.

Un certificat peut être mis en attente par l'émission d'un élément de liste CRL contenant un code motif **certificateHold** (*certificat mis en attente*). La notification de mise en attente du certificat peut contenir un code d'instruction de mise en attente optionnel véhiculant des informations supplémentaires à l'intention des utilisateurs de certificat (voir 8.5.2.3). Un certificat peut être traité de l'une des trois manières suivantes une fois qu'il a été mis en attente:

a) il peut être conservé dans la liste CRL sans autre action, ce qui conduit les utilisateurs à rejeter des transactions émises pendant la durée de la mise en attente; ou

b) il peut être remplacé par une révocation (définitive) du même certificat, auquel cas le motif sera l'un des motifs normalisés de révocation; la date sera celle à laquelle le certificat a été mis en attente et le champ d'extension de code d'instruction optionnel ne sera pas présent;

c) il peut être annulé de manière explicite avec suppression de l'élément de la liste CRL.

Le code motif **removeFromCRL** (*retirer de la liste CRL*) est utilisé uniquement avec des listes CRL delta (voir 8.6) et indique qu'un élément existant d'une liste CRL doit maintenant être supprimé en raison de l'expiration du certificat ou de la suppression de sa mise en attente. Un élément contenant ce code motif sera utilisé dans des listes CRL delta pour lesquelles la liste CRL de base correspondante et toute liste CRL ultérieure (delta ou complète pour le domaine d'application) contient un élément concernant le même certificat avec un code motif **certificateHold**.

Cette extension est toujours non critique.

8.5.2.3 Extension de code d'instruction de mise en attente

Ce champ d'extension d'élément de liste CRL fournit un identificateur d'instruction enregistré pouvant être présent pour indiquer la mesure à prendre en présence d'un certificat mis en attente. Il s'applique uniquement pour un élément qui contient un code motif **certificateHold**. Ce champ est défini comme suit:

```
holdInstructionCode EXTENSION ::= {
    SYNTAX              HoldInstruction
    IDENTIFIED BY       id-ce-instructionCode }

HoldInstruction ::= OBJECT IDENTIFIER
```

Cette extension est toujours non critique. La présente Spécification d'annuaire ne définit aucun code normalisé d'instruction de mise en attente.

NOTE – Exemples d'instructions de mise en attente: "veuillez vous mettre en rapport avec l'autorité de certification" ou "récupération du jeton d'utilisateur".

8.5.2.4 Extension de date de non validité

Ce champ d'extension d'élément de liste CRL indique la date à laquelle il a été établi ou présumé que la clé privée a été mise en danger ou à laquelle le certificat doit être considéré comme non valide pour une autre raison. Cette date peut être antérieure à la date de révocation figurant dans l'élément de la liste CRL, qui correspond à la date de traitement de la révocation par l'autorité. Ce champ est défini comme suit:

```
invalidityDate EXTENSION ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-invalidityDate }
```

Cette extension est toujours non critique.

NOTE 1 – La date figurant dans cette extension n'est pas suffisante, en elle-même, à des fins de non-répudiation. Il peut s'agir, par exemple, d'une date indiquée par le détenteur de la clé privée et il est possible que ce dernier revendique à tort que cette clé a été mise en danger à une date antérieure, afin de répudier une signature générée de manière valide.

NOTE 2 – Lorsqu'une répudiation est publiée pour la première fois par une autorité dans une liste CRL, la date de non validité peut être antérieure à la date d'émission de listes CRL antérieures. La date de révocation ne doit pas précéder la date d'émission de listes CRL antérieures.

8.5.2.5 Extension de domaine d'application de liste CRL

Le domaine d'application d'une liste CRL est indiqué dans cette dernière afin de prévenir une attaque par substitution de liste CRL contre une application qui ne prend pas en charge l'extension de domaine d'application; cette extension doit être marquée comme critique si elle est présente.

Cette extension peut être utilisée pour la fourniture de déclarations de domaine d'application pour divers types de liste CRL suivants:

- listes CRL simples fournissant des informations de révocation concernant des certificats émis par une autorité unique;
- listes CRL indirectes fournissant des informations de révocation concernant des certificats émis par plusieurs autorités;
- listes CRL delta mettant à jour des informations de révocation émises précédemment;
- listes CRL delta fournissant des informations de révocation mettant à jour plusieurs listes CRL de base émises par une ou plusieurs autorités.

```
crlScope EXTENSION ::= {
    SYNTAX                  CRLScopeSyntax
    IDENTIFIED BY           id-ce-cRLScope }

CRLScopeSyntax ::=          SEQUENCE SIZE (1..MAX) OF PerAuthorityScope

PerAuthorityScope ::= SEQUENCE {
    authorityName               [0]     GeneralName OPTIONAL,
    distributionPoint           [1]     DistributionPointName OPTIONAL,
    onlyContains                [2]     OnlyCertificateTypes OPTIONAL,
    onlySomeReasons             [4]     ReasonFlags OPTIONAL,
    serialNumberRange           [5]     NumberRange OPTIONAL,
    subjectKeyIdRange           [6]     NumberRange OPTIONAL,
    nameSubtrees                [7]     GeneralNames OPTIONAL,
    baseRevocationInfo          [9]     BaseRevocationInfo OPTIONAL
    }

OnlyCertificateTypes        ::= BIT STRING {
    user            (0),
    authority       (1),
    attribute       (2) }

NumberRange ::= SEQUENCE {
    startingNumber      [0]     INTEGER OPTIONAL,
    endingNumber        [1]     INTEGER OPTIONAL,
    modulus                     INTEGER OPTIONAL }

BaseRevocationInfo ::= SEQUENCE {
    cRLStreamIdentifier     [0]     CRLStreamIdentifier     OPTIONAL,
    cRLNumber               [1]     CRLNumber,
    baseThisUpdate          [2]     GeneralizedTime }
```

Si la liste CRL est une liste CRL indirecte qui fournit des informations de statut de révocation pour plusieurs autorités, l'extension contiendra plusieurs structures **PerAuthorityScope** (*domaine d'application de l'autorité*), à savoir une ou plusieurs pour chacune des autorités pour lesquelles des informations de révocation sont fournies. Chacune des instances de **PerAuthorityScope** concernant une autorité autre que celle qui émet cette liste CRL contiendra le composant **authorityName** (*nom d'autorité*).

Si la liste CRL est une liste dCRL qui fournit des informations de delta de statut de révocation pour plusieurs listes CRL de base émises par une seule autorité, l'extension contiendra plusieurs structures **PerAuthorityScope**, à savoir une pour chacune des listes CRL de base pour lesquelles cette liste dCRL fournit des mises à jour. La valeur du composant **authorityName** éventuellement présent sera la même pour toutes les instances multiples de la structure **PerAuthorityScope**.

Si la liste CRL est une liste dCRL indirecte qui fournit des informations de delta de statut de révocation pour plusieurs listes CRL de base émises par plusieurs autorités, l'extension contiendra plusieurs structures **PerAuthorityScope**, à savoir une pour chacune des listes CRL de base pour lesquelles cette liste dCRL fournit des mises à jour. Chacune des instances de **PerAuthorityScope** concernant une autorité autre que celle qui émet cette liste dCRL indirecte contiendra le composant **authorityName.**

Les champs de chacune des instances de la structure **PerAuthorityScope** présents dans l'extension seront utilisés comme suit. Il convient de noter que dans le cas de listes CRL indirectes et de listes dCRL indirectes, chaque instance de **PerAuthorityScope** peut contenir des combinaisons différentes de ces champs et avec des valeurs différentes.

Le champ **authorityName** indique, s'il est présent, l'autorité qui a émis les certificats pour lesquels des informations de révocation sont fournies. La valeur par défaut de **authorityName**, s'il est absent, est le nom de l'émetteur de la liste CRL.

Le champ **distributionPoint** (*point de répartition*), s'il est présent, est utilisé comme décrit dans l'extension **issuingDistributionPoint** (*point de répartition émetteur*).

Le champ **onlyContains** (*contient uniquement*), s'il est présent, indique le ou les types de certificat concernés par les informations de statut contenues dans la liste CRL. Si ce champ est absent, la liste CRL contient alors des informations concernant tous les types de certificats.

Le champ **onlySomeReasons** (*uniquement certains motifs*) s'il est présent, est utilisé comme décrit dans l'extension **issuingDistributionPoint**.

L'élément **serialNumberRange** (*domaine du numéro de série*) est utilisé de la manière suivante s'il est présent. Lorsqu'une valeur **modulus** (*module*) est présente, le numéro de série est réduit en prenant le reste de la division de la valeur en question par le module avant de vérifier l'appartenance au domaine. Un certificat avec un numéro de série (réduit) est considéré comme appartenant au domaine d'application de la liste CRL si son numéro de série est:

- supérieur ou égal à **startingNumber** (*numéro de départ*) et inférieur à **endingNumber** (*numéro de fin*) lorsque ces deux éléments sont présents;
- supérieur ou égal à **startingNumber** lorsque **endingNumber** n'est pas présent;
- inférieur à **endingNumber** lorsque **startingNumber** n'est pas présent.

L'élément **subjectKeyIdRange** (*domaine d'identificateur de clé du sujet*), s'il est présent, est interprété comme un domaine **serialNumberRange**, avec l'exception que le nombre utilisé est la valeur figurant dans l'extension **subjectKeyIdentifier** (*identificateur de clé du sujet*) du certificat. Le codage DER de l'élément **BIT STRING** (en omettant l'étiquette, la longueur et l'octet des bits non utilisés) doit être considéré comme la valeur du codage DER d'un élément **INTEGER**. Si le bit0 de l'élément **BIT STRING** est défini, un octet nul additionnel doit être ajouté de façon à garantir que le codage résultant représente un entier (**INTEGER**) positif, par exemple:

03 02 01 f7 (représente l'élément bits 0-6 défini)

correspond à

02 02 00 f7 (c'est-à-dire valeur décimale 247)

Le champ **nameSubtrees** (*sous-arbres de nom*), s'il est présent, utilise les mêmes conventions de formes de nom que celles spécifiées dans l'extension **nameConstraints** (*contraintes de nom*).

Le champ **baseRevocationInfo** (*informations de révocation de base*), s'il est présent, indique que la liste CRL est une liste dCRL concernant les certificats couverts par la structure **PerAuthorityScope**. L'utilisation de l'extension **crlScope** (*domaine d'application de liste CRL*) pour indiquer qu'une liste CRL est une liste dCRL diffère de la manière suivante de l'utilisation de l'extension **deltaCRLIdentifier** (*identificateur de liste CRL delta*). Dans le cas **crlScope**, les informations du composant **baseRevocationInfo** indiquent l'instant à partir duquel la liste CRL contenant cette extension fournit des mises à jour. Bien que ceci fasse référence à une liste CRL, cette dernière peut ou non s'appliquer pour l'ensemble du domaine, alors que l'extension **deltaCRLIdentifier** fait référence à une liste CRL complète émise pour le domaine d'application. Cependant, l'information mise à jour fournie dans une liste dCRL contenant l'extension **crlScope** est une mise à jour pour l'information de révocation complète pour le domaine d'application, indépendamment du fait que la référence faite par le composant **baseRevocationInfo** était effectivement émise de manière complète pour le même domaine d'application. Ce procédé est plus souple que l'extension **deltaCRLIndicator** (*indicateur de liste CRL delta*) parce que les utilisateurs peuvent construire localement des listes CRL complètes basées sur la date et l'heure plutôt que d'émettre des listes CRL de base complètes pour le domaine d'application. Une liste dCRL fournit toujours, dans les deux cas, une mise à jour du statut de révocation de certificat valable au sein d'un domaine d'application donné et à un instant donné. Toutefois, dans le cas **deltaCRLIndicator**, cet instant doit être tel qu'une liste CRL complète pour le domaine a été émise et a fait l'objet d'une référence. Dans le cas **crlScope**, la liste CRL faisant l'objet d'une référence n'est pas nécessairement complète pour ce domaine d'application.

Plusieurs listes dCRL peuvent être publiées avant la publication d'une nouvelle liste CRL de base, en fonction de la politique de l'autorité responsable. Les listes dCRL contenant l'extension **crlScope** pour faire référence à leur instant de construction ne font pas nécessairement référence au numéro **cRLNumber** (*numéro de liste CRL*) de la liste CRL de base émise le plus récemment dans le champ **BaseRevocationInfo** (*informations de révocation de base*). Le numéro **cRLNumber** auquel fait référence le champ **BaseRevocationInfo** d'une liste dCRL sera toutefois inférieur ou égal au numéro **cRLNumber** de la liste CRL émise la plus récente et qui est complète pour le domaine d'application.

Il convient de noter que les extensions **issuingDistributionPoint** et **crlScope** peuvent entrer en conflit et que leur utilisation simultanée n'est pas prévue. Si la liste CRL contient toutefois une extension **issuingDistributionPoint** et une extension **crlScope**, le certificat concerne alors le domaine d'application de la liste CRL si, et seulement s'il est conforme aux critères des deux extensions. Si la liste CRL ne contient aucune des extensions **issuingDistributionPoint** et **crlScope**, le domaine d'application est alors celui de l'autorité dans son ensemble et la liste CRL peut être utilisée pour tout certificat émis par cette autorité.

Lorsqu'un système utilisant des certificats emploie une liste CRL qui contient une extension **crlScope** pour la vérification du statut d'un certificat, il doit vérifier de la manière suivante que le certificat et les codes motif en question appartiennent au domaine d'application de la liste CRL telle qu'elle est définie par l'extension **crlScope**:

a) le système utilisant des certificats doit vérifier que le certificat appartient au domaine d'application indiqué par l'intersection des domaines **serialNumberRange**, **subjectKeyIdRange** et **nameSubtrees** et qu'il est en cohérence avec les champs **distributionPoint** et **onlyContains** éventuellement présents dans la structure **PerAuthorityScope** concernée;

b) si la liste CRL contient un composant **onlySomeReasons** dans l'extension **crlScope**, le système utilisant des certificats doit alors vérifier que les codes motif couverts par cette liste CRL sont pertinents aux fins de l'application. Dans le cas contraire, il est possible qu'aucune liste CRL supplémentaire ne soit requise. Il convient de noter que si la liste CRL contient simultanément les extensions **crlScope** et **issuingDistributionPoint** et que ces dernières contiennent un composant **onlySomeReasons**, cette liste CRL couvre alors uniquement ceux des codes motif qui sont contenus dans les deux composants **onlySomeReasons** des extensions couvertes par cette liste CRL.

8.5.2.6 Extension de référence de statut

Cette extension de liste CRL est utilisable pour véhiculer des informations concernant des notifications de révocation vers des utilisateurs de certificat. Elle figure de ce fait dans une liste CRL qui ne contient pas de notifications de révocation de certificat. Une liste CRL qui contient cette extension ne sera pas employée par des utilisateurs de certificat ou des participants comme source fiable de notification de révocation, mais comme un outil permettant de s'assurer que les informations de révocation adéquates sont utilisées.

Cette extension concerne les deux fonctions primaires suivantes:

- elle fournit un procédé permettant de publier une liste fiable de listes CRL contenant toutes les informations pertinentes qui aident les participants faisant confiance à déterminer s'ils disposent ou non des informations de révocation suffisantes pour leurs besoins. Une autorité peut, par exemple, émettre de manière périodique une nouvelle liste CRL authentifiée, en général avec une fréquence de publication relativement élevée (en comparaison avec les fréquences d'autres listes CRL). La liste peut contenir une date et une heure de mise à jour pour chaque référence de liste CRL. Lorsqu'il a obtenu cette liste, un utilisateur de certificat peut alors déterminer rapidement si les copies des listes CRL présentes dans son cache sont encore à jour. Ceci peut éliminer une grande partie des extractions inutiles de listes CRL. Ce procédé permet en outre aux utilisateurs de certificat de prendre connaissance de listes CRL émises par l'autorité en-dehors de son cycle normal de mise à jour, ce qui améliore la pertinence du système de liste CRL;

- cette extension fournit également un procédé permettant de transférer un participant faisant confiance depuis un emplacement préliminaire (indiqué par exemple par un point de répartition d'extension de liste CRL ou par l'entrée d'annuaire de l'autorité émettrice) vers un emplacement différent contenant des informations de révocation. Cette fonctionnalité permet aux autorités de modifier le mode de découpage qu'elles utilisent pour la liste CRL sans impact sur les certificats existants ou les utilisateurs de certificat. L'autorité indique pour ce faire tout nouvel emplacement ainsi que le domaine d'application de la liste CRL qui s'y trouve. Le participant faisant confiance peut comparer le certificat qui l'intéresse avec la déclaration de domaine d'application et utiliser le pointeur pour accéder au nouvel emplacement des informations concernant le certificat qu'elles valident.

L'extension est elle-même extensible et peut également être utilisée par d'autres systèmes de révocation qui ne se basent pas sur des listes CRL.

```
statusReferrals EXTENSION ::= {
    SYNTAX              StatusReferrals
    IDENTIFIED BY       id-ce-statusReferrals }

StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral

StatusReferral ::= CHOICE {
    cRLReferral         [0]     CRLReferral,
    otherReferral       [1]     INSTANCE OF OTHER-REFERRAL}

CRLReferral ::= SEQUENCE {
    issuer              [0]     GeneralName OPTIONAL,
    location            [1]     GeneralName OPTIONAL,
    deltaRefInfo        [2]     DeltaRefInfo OPTIONAL,
    cRLScope                    CRLScopeSyntax,
    lastUpdate          [3]     GeneralizedTime OPTIONAL,
    lastChangedCRL      [4]     GeneralizedTime OPTIONAL}

DeltaRefInfo ::= SEQUENCE {
    deltaLocation       GeneralName,
    lastDelta           GeneralizedTime OPTIONAL }

OTHER-REFERRAL ::= TYPE-IDENTIFIER
```

Le champ **issuer** indique l'entité qui signe la liste CRL; la valeur par défaut est le nom de l'émetteur de la liste CRL dans laquelle il figure.

Le champ **location** fournit l'emplacement vers lequel doit être dirigé le demandeur de référence et possède la même valeur par défaut que le nom **issuer**.

Le champ **deltaRefInfo** (*informations de référence de delta*) fournit un emplacement optionnel de remplacement à partir duquel une liste dCRL peut être obtenue, ainsi qu'une date optionnelle du delta précédent.

Le champ **crlScope** fournit le domaine d'application de la liste CRL qui se trouve à l'emplacement de référence.

Le champ **lastUpdate** (*dernière mise à jour*) contient la valeur du champ **thisUpdate** de la liste CRL ayant fait l'objet de la référence la plus récente.

Le champ **lastChangedCRL** (*dernière liste CRL modifiée*) contient la valeur du champ **thisUpdate** de la liste CRL avec un contenu modifié ayant fait l'objet de la référence la plus récente.

L'identificateur **OTHER-REFERRAL** (*autre référence*) fournit l'extensibilité permettant l'utilisation future d'autres systèmes de révocation non basés sur des listes CRL.

Cette extension est toujours marquée comme critique de manière à garantir que la liste CRL qui la contient n'est pas utilisée par inadvertance par un système utilisant des certificats et qui lui fait confiance comme source d'informations pour le statut de révocation de certificat.

Si cette extension est présente et reconnue par un système utilisant des certificats, ce dernier n'utilisera pas la liste CRL comme source d'informations de statut de révocation. Il doit dans ce cas localiser les informations adéquates de statut de révocation en utilisant, soit les informations contenues dans cette extension, soit d'autres moyens en dehors du domaine d'application de la présente Spécification.

Si cette extension est présente mais non reconnue par un système utilisant des certificats, ce dernier n'utilisera pas la liste CRL comme source d'informations de statut de révocation. Il doit dans ce cas localiser les informations adéquates de statut de révocation en utilisant d'autres moyens en dehors du domaine d'application de la présente Spécification.

8.5.2.7 Extension d'identificateur de flux de liste CRL

Le champ d'identificateur de flux de liste CRL est utilisé pour indiquer le contexte au sein duquel le numéro de liste CRL est unique.

```
cRLStreamIdentifier EXTENSION      ::= {
    SYNTAX                 CRLStreamIdentifier
    IDENTIFIED BY          id-ce-cRLStreamIdentifier }

CRLStreamIdentifier ::= INTEGER (0..MAX)
```

Cette extension est toujours non critique.

La valeur de cette extension doit être unique pour chaque autorité. L'identificateur de flux de liste CRL associé à un numéro de liste CRL permet d'identifier de manière unique toute liste CRL d'un type quelconque émise par une autorité donnée.

8.5.2.8 Extension de liste ordonnée

L'extension de liste ordonnée indique que la séquence de certificats révoqués figurant dans le champ **revokedCertificates** d'une liste CRL est classée dans l'ordre ascendant des numéros ou des dates de révocation. Ce champ est défini comme suit:

```
orderedList EXTENSION      ::= {
    SYNTAX                 OrderedListSyntax
    IDENTIFIED BY          id-ce-orderedList }

OrderedListSyntax ::= ENUMERATED {
ascSerialNum       (0),
ascRevDate         (1) }
```

Cette extension est toujours non critique.

- **ascSerialNum** (*numéro de série ascendant*) indique que la séquence des certificats révoqués dans une liste CRL est classée par ordre ascendant de numéro de série de certificat, sur la base de la valeur du composant **serialNumber** de chaque élément de la liste.
- **ascRevDate** (*date de révocation ascendante*) indique que la séquence des certificats révoqués dans une liste CRL est classée par ordre ascendant de date, sur la base de la valeur du composant **revocationDate** de chaque élément de la liste.

Aucune information n'est fournie au sujet de l'ordre des certificats révoqués dans la liste CRL si le composant **orderedList** (*liste ordonnée*) n'est pas présent.

8.5.2.9 Extensions d'informations delta

Cette extension de liste CRL est utilisable dans des listes CRL qui ne sont pas des listes dCRL; elle indique à des participants faisant confiance que des listes dCRL sont également disponibles pour la liste CRL qui contient cette extension. L'extension fournit l'emplacement où il est possible de trouver les listes connexes et, de manière optionnelle, la date à laquelle la prochaine liste dCRL sera émise.

```
deltaInfo EXTENSION        ::=  {
    SYNTAX                 DeltaInformation
    IDENTIFIED BY          id-ce-deltaInfo }

DeltaInformation           ::=        SEQUENCE {
    deltaLocation          GeneralName,
    nextDelta              GeneralizedTime OPTIONAL }
```

Cette extension est toujours non critique.

8.6 Points de répartition de liste CRL et extensions delta de liste CRL

8.6.1 Expression des besoins

Les listes de révocation peuvent devenir longues et encombrantes, de sorte qu'il est nécessaire de pouvoir représenter des listes partielles. Diverses solutions existent pour deux implémentations possibles du traitement des listes CRL.

Le premier type d'implémentation fait appel à des stations de travail personnelles, utilisant éventuellement un jeton de chiffrement attaché. Ces implémentations disposeront probablement au plus d'une capacité de stockage sécurisée limitée. Il s'ensuit qu'il est nécessaire d'examiner la totalité de la liste CRL pour vérifier sa validité puis celle du certificat. Ce traitement peut être relativement long si la liste CRL est volumineuse. Il devient alors nécessaire de fractionner les listes CRL pour éliminer ce problème dans ce type d'implémentation.

Le deuxième type d'implémentation fait appel à des serveurs à hautes performances qui traitent un volume important de messages, par exemple des serveurs de traitement de transactions. Les listes CRL sont en général traitées dans un tel environnement par une tâche d'arrière-plan; une fois que la liste CRL est validée, son contenu est stocké localement sous une forme qui accélère son examen, par exemple en utilisant un bit pour chaque certificat révoqué. Cette représentation est stockée dans une mémoire sécurisée. Ce type de serveur nécessite en général des listes CRL à jour pour un grand nombre d'autorités. Comme il dispose déjà d'une liste de certificats révoqués précédemment, il a besoin d'extraire uniquement une liste des certificats nouvellement révoqués. Cette liste, appelée liste dCRL, sera d'une taille plus réduite et nécessitera moins de ressources d'extraction et de traitement qu'une liste CRL complète.

Les besoins suivants sont liés de ce fait aux points de répartition de liste CRL et aux listes dCRL:

a) il doit être possible, pour gérer les tailles de liste CRL, d'attribuer à des listes CRL différentes des sous-ensembles de tous les certificats émis par une autorité. Ceci peut se faire en associant chaque certificat à un point de répartition de liste CRL qui est:
   - soit une entrée d'annuaire dont l'attribut de liste CRL contiendra le cas échéant un élément de révocation; ou
   - un emplacement tel qu'une adresse de messagerie électronique ou un identificateur uniforme de ressource Internet capable de fournir la liste CRL pertinente,

b) il est souhaitable, pour des raisons de performance, de limiter le nombre de listes CRL devant être vérifiées lors de la validation de certificats multiples, par exemple pour un itinéraire de certification. Ceci peut se faire en faisant émettre et signer par un émetteur des listes CRL contenant des révocations en provenance de plusieurs autorités;

c) il est nécessaire de disposer de listes CRL distinctes pour les révocations de certificats d'autorité et de certificats d'entité finale. Ceci facilite le traitement des itinéraires de certification puisque la liste CRL de révocation des certificats d'autorité est probablement très courte (en général vide). Les attributs **authorityRevocationList** (*liste de révocation d'autorité*) et **certificateRevocationList** (*liste de révocation de certificat*) ont été spécifiés à cet effet. La présence d'un indicateur dans une liste CRL est toutefois nécessaire pour identifier cette dernière. Cet indicateur est nécessaire pour permettre la détection de la substitution illégitime d'une liste par une autre;

d) il est nécessaire de prévoir une autre liste CRL pour des situations dangereuses éventuelles (en cas de risque significatif d'utilisation irrégulière de clé privée) à la place de la liste contenant toutes les terminaisons de rattachement habituelles (lorsqu'il n'existe pas de risque significatif d'utilisation irrégulière de la clé privée);

e) il est également nécessaire de disposer de listes CRL partielles (appelées listes dCRL ou listes CRL delta) contenant uniquement des éléments qui ont été révoqués depuis l'émission d'une liste CRL de base;

f) il est nécessaire de pouvoir indiquer, pour les listes CRL delta, la date et l'heure à partir desquelles s'appliquent les mises à jour qu'elles contiennent;

g) il est nécessaire de pouvoir indiquer, au sein d'un certificat, l'endroit où il est possible de trouver la liste CRL la plus récente (par exemple, le delta le plus récent).

8.6.2 Point de répartition de liste CRL et champs d'extension de liste CRL delta

Les champs d'extension suivants sont définis:

a) *points de répartition de liste CRL*;

b) *point de répartition émetteur*;

c) *emetteur de certificat*;

d) *indicateur de liste CRL delta*;

e) *mise à jour de base*;

f) *liste CRL la plus récente*.

Les points de répartition de liste CRL et la liste CRL la plus récente seront utilisés exclusivement comme extension de certificat. Le point de répartition émetteur, l'indicateur de liste CRL delta et la mise à jour de base seront utilisés exclusivement comme extensions de liste CRL. L'émetteur de certificat sera utilisé exclusivement comme une extension d'élément de liste CRL.

8.6.2.1 Extension de point de répartition de liste CRL

Cette extension sera utilisée exclusivement comme extension de certificat et peut être utilisée dans des certificats de clé publique d'entité finale émis par des autorités et dans des certificats d'attribut. Ce champ indique le point de répartition de liste CRL ou pointe vers une entité à laquelle doit se référer un utilisateur de certificat pour établir si le certificat a été révoqué. Un utilisateur de certificat peut obtenir une liste CRL à partir d'un point de répartition pertinent ou peut être en mesure d'obtenir une liste CRL actuelle complète à partir de l'élément d'annuaire de l'autorité.

Ce champ est défini comme suit:

```
cRLDistributionPoints EXTENSION ::= {
    SYNTAX              CRLDistPointsSyntax
    IDENTIFIED BY       id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
    distributionPoint   [0]     DistributionPointName OPTIONAL,
    reasons             [1]     ReasonFlags OPTIONAL,
    cRLIssuer           [2]     GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
    fullName                        [0]     GeneralNames,
    nameRelativeToCRLIssuer         [1]     RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {
    unused                  (0),
    keyCompromise           (1),
    cACompromise            (2),
    affiliationChanged      (3),
    superseded              (4),
    cessationOfOperation    (5),
    certificateHold         (6),
    privilegeWithdrawn      (7),
    aACompromise            (8) }
```

Le composant **distributionPoint** indique l'emplacement à partir duquel la liste CRL peut être obtenue. La valeur par défaut du point de répartition est le nom de l'émetteur de la liste CRL si ce composant est absent.

Le nom du point de répartition peut prendre plusieurs formes de nom lorsque la variante **fullName** [*nom complet*] est utilisée ou lorsque l'option par défaut s'applique. Le même nom sera présent, au moins sous l'une de ses formes, dans le champ **distributionPoint** l'extension de point de répartition émetteur de la liste CRL. Un système utilisant des certificats n'a pas l'obligation d'être en mesure de traiter toutes les formes de nom. Il peut utiliser un point de répartition dans la

mesure où une forme de nom au moins peut être traitée. Si aucune forme de nom ne peut être traitée par un point de répartition, un système utilisant des certificats peut alors continuer à utiliser le certificat dans la mesure où les informations de révocation peuvent être obtenues à partir d'une autre source, par exemple un autre point de répartition ou l'entrée d'annuaire de l'autorité.

Le composant **nameRelativeToCRLIssuer** [*nom relatif par rapport à l'émetteur de liste CRL*] peut être utilisé uniquement si le point de répartition de liste CRL a reçu l'attribution d'un nom d'annuaire qui est subordonné directement au nom d'annuaire de l'émetteur de la liste CRL. Le composant **nameRelativeToCRLIssuer** véhicule dans un tel cas le nom distinctif relatif par rapport au nom d'annuaire de l'émetteur de la liste CRL.

Le composant **reasons** [*motifs*] indique les motifs de révocation couverts par cette liste CRL. Si le composant **reasons** est absent, le point de répartition de liste CRL correspondant distribue alors une liste CRL qui contiendra un élément pour un certificat révoqué, quel qu'en soit le motif. Dans le cas contraire, la valeur du composant **reasons** indique quels sont les motifs de révocation qui sont couverts par le point de répartition de liste CRL correspondant.

Le composant **cRLIssuer** [*émetteur de liste CRL*] identifie l'autorité qui émet et signe la liste CRL. La valeur par défaut de l'émetteur de liste CRL est le nom de l'émetteur du certificat si ce composant est absent.

Cette extension peut être critique ou non, au choix de l'émetteur du certificat. Il est recommandé qu'elle soit marquée comme non critique pour favoriser l'interfonctionnement.

Si cette extension est marquée comme critique, un système utilisant des certificats n'emploiera alors pas le certificat avant d'avoir extrait et vérifié, à partir d'un des points de répartition nommés, une liste CRL qui couvre les codes motif concernés. Lorsque les points de répartition sont utilisés pour distribuer des informations de liste CRL pour tous les codes motif de révocation et que tous les certificats émis par l'autorité de certification contiennent le composant **cRLDistributionPoints** [*point de répartition de liste CRL*] comme extension critique, l'autorité de certification n'a pas l'obligation de publier également une liste CRL complète dans l'entrée d'autorité de certification.

Si cette extension est marquée comme non critique et si un système utilisant des certificats ne reconnaît pas le type du champ d'extension, celui-ci doit alors utiliser le certificat uniquement dans l'un des cas suivants:

- il peut acquérir auprès de l'autorité et vérifier la liste CRL complète (l'indication que cette liste est complète est fournie par l'absence d'un champ d'extension d'émetteur de point de répartition dans la liste CRL);
- la vérification de la révocation n'est pas exigée par la politique locale;
- la vérification de la révocation est effectuée par d'autres moyens.

NOTE 1 – Il est possible que des listes soient émises, pour un même certificat, par plusieurs émetteurs de liste CRL. La coordination entre ces émetteurs de liste CRL et l'autorité émettrice est du ressort de la politique de l'autorité.

NOTE 2 – La signification de chaque code motif est définie de la même manière que pour le champ motif du 8.5.2.2 de la présente Spécification.

8.6.2.2 Extension de point de répartition émetteur

Cette extension identifie le point de répartition de liste CRL de cette liste particulière et indique si cette dernière contient uniquement des révocations de certificats d'entité finale, des révocations de certificats d'autorité ou des révocations concernant un certain ensemble de motifs. La liste CRL est signée au moyen de la clé de l'émetteur de liste CRL, les points de répartition de liste CRL ne possédant pas de paire de clés. Toutefois, une liste CRL répartie par le biais de l'annuaire est stockée dans l'entrée du point de répartition de liste CRL, qui peut ne pas être l'entrée d'annuaire de l'émetteur de la liste CRL. La liste CRL contiendra des éléments pour tous les certificats révoqués non caducs en provenance de l'émetteur de la liste CRL si ce champ est absent.

Ce champ est défini comme suit:

```
issuingDistributionPoint EXTENSION ::= {
    SYNTAX              IssuingDistPointSyntax
    IDENTIFIED BY  id-ce-issuingDistributionPoint }

IssuingDistPointSyntax ::= SEQUENCE {
    distributionPoint           [0]  DistributionPointName OPTIONAL,
    onlyContainsUserCerts       [1]  BOOLEAN DEFAULT FALSE,
    onlyContainsAuthorityCerts  [2]  BOOLEAN DEFAULT FALSE,
    onlySomeReasons             [3]  ReasonFlags OPTIONAL,
    indirectCRL                 [4]  BOOLEAN DEFAULT FALSE,
    onlyContainsAttributeCerts  [5]  BOOLEAN DEFAULT FALSE }
```

Le composant **distributionPoint** contient le nom du point de répartition sous une ou plusieurs formes de nom. La liste CRL contiendra des éléments pour tous les certificats révoqués en provenance de l'émetteur de la liste CRL si ce champ est absent. Une fois qu'un certificat est paru dans une liste CRL, il ne figurera plus dans toutes les listes CRL suivantes après son expiration.

Si la valeur du composant **onlyContainsUserCerts** [*contient uniquement des certificats d'utilisateur*] est égale à "Vrai", la liste CRL contient alors uniquement des révocations pour des certificats d'entité finale. Si la valeur du composant **onlyContainsAuthorityCerts** [*contient uniquement des certificats d'autorité*] est égale à "Vrai", la liste CRL contient alors uniquement des révocations pour des certificats d'autorité. Si le composant **onlySomeReasons** est présent, la liste CRL contient alors uniquement des révocations pour le ou les motifs indiqués; dans le cas contraire, la liste CRL contient des révocations pour tous les motifs.

Si la valeur du composant **indirectCRL** [*liste CRL indirecte*] est égale à "Vrai", la liste CRL peut alors contenir des notifications de révocation provenant d'autorités autres que l'émetteur de la liste CRL. L'autorité particulière responsable pour chaque entrée est indiquée par l'extension d'entrée d'émetteur de liste CRL dans cette entrée ou conformément aux règles par défaut décrites au 8.6.2.3. Il est de la responsabilité de l'émetteur d'une telle liste CRL de s'assurer qu'elle est complète, c'est-à-dire qu'elle contient toutes les entrées de révocation, d'une manière cohérente avec les indicateurs **onlyContainsUserCerts**, **onlyContainsAuthorityCerts** et **onlySomeReasons** en provenance de toutes les autorités qui indiquent cet émetteur de liste CRL dans leurs certificats.

Si la valeur du composant **onlyContainsAttributeCerts** [*contient uniquement des certificats d'attribut*] est égale à "Vrai", la liste CRL contient alors uniquement des révocations de certificats d'attribut.

Les règles suivantes d'utilisation des attributs s'appliquent pour des listes CRL réparties par le biais de l'annuaire. Une liste CRL dont le composant **onlyContainsAuthorityCerts** est positionné sera répartie par le biais de l'attribut **authorityRevocationList** du point de répartition associé, ou, si aucun point de répartition n'est indiqué, par le biais de l'attribut **authorityRevocationList** de l'entrée de l'émetteur de la liste CRL. Dans le cas contraire, la liste CRL sera répartie par le biais de l'attribut **certificateRevocationList** du point de répartition associé, ou si aucun point de répartition n'est indiqué, par le biais de l'attribut **certificateRevocationList** de l'entrée d'autorité.

Cette extension est toujours critique. Un utilisateur de certificat qui ne comprend pas cette extension ne peut pas faire l'hypothèse que la liste CRL contient une liste complète de certificats révoqués de l'autorité indiquée. Les listes CRL qui ne contiennent pas d'extension critique doivent contenir tous les éléments actuels de liste CRL pour l'autorité émettrice, y compris les éléments pour tous les certificats d'utilisateur et certificats d'autorité révoqués.

NOTE 1 – Les moyens utilisés par les autorités pour communiquer les informations de révocation aux émetteurs de liste CRL sont en dehors du domaine d'application de la présente Recommandation | Norme internationale.

NOTE 2 – Si une autorité émet, à partir de sa propre entrée d'annuaire (et non à partir d'un point de répartition de liste CRL avec un nom différent), une liste CRL avec un composant **onlyContainsUserCerts** ou **onlyContainsAuthorityCerts** positionné, elle doit alors s'assurer que tous les certificats couverts par cette liste CRL contiennent l'extension **basicConstraints**.

8.6.2.3 Extension d'émetteur de certificat

Cette extension indique l'émetteur de certificat associé à un élément d'une liste CRL indirecte, c'est-à-dire une liste CRL dont l'indicateur **indirectCRL** est positionné dans son extension de point de répartition émetteur. Si cette extension ne figure pas dans le premier élément d'une liste CRL indirecte, l'émetteur de certificat est alors par défaut l'émetteur de la liste CRL. Si cette extension n'est pas présente dans un autre élément de cette liste, l'émetteur de certificat pour cet élément est alors le même que celui de l'élément précédent.

Ce champ est défini comme suit:

```
certificateIssuer EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-certificateIssuer }
```

Cette extension est toujours critique. Si une implémentation ignore cette extension, elle ne peut alors pas attribuer correctement des éléments de liste CRL à des certificats.

8.6.2.4 Extension d'indicateur de liste CRL delta

L'extension d'indicateur de liste CRL delta signale qu'une liste CRL est une liste CRL delta (dCRL) qui fournit des mises à jour pour une liste CRL de base faisant l'objet d'une référence. La liste CRL de base est une liste CRL qui a été émise de manière explicite sous la forme d'une liste CRL complète pour un domaine d'utilisation donné. La liste CRL contenant l'extension d'indicateur de liste CRL delta met à jour le statut de révocation de certificat pour le même domaine d'application. Ce domaine n'inclut pas nécessairement tous les motifs de révocation ou tous les certificats émis par une autorité de certification, en particulier dans le cas où la liste CRL est un point de répartition de liste CRL. Toutefois, la combinaison d'une liste CRL qui contient à la fois l'extension d'indicateur de liste CRL delta et la liste CRL faisant l'objet d'une référence dans le composant **BaseCRLNumber** [*numéro de liste CRL de base*] de cette extension est équivalente à une liste CRL complète, pour le domaine d'application et au moment de la publication de la liste dCRL.

Ce champ est défini comme suit:

```
deltaCRLIndicator EXTENSION ::= {
    SYNTAX              BaseCRLNumber
    IDENTIFIED BY       id-ce-deltaCRLIndicator }

BaseCRLNumber ::= CRLNumber
```

La valeur du type **BaseCRLNumber** indique le numéro de la liste CRL de base qui a été utilisée comme base pour la génération de cette liste dCRL. La liste CRL de référence sera une liste CRL complète pour le domaine d'application.

Cette extension est toujours critique. Un utilisateur de certificat qui ne comprend pas l'utilisation des listes dCRL ne doit pas utiliser une liste CRL avec cette extension, étant donné que cette liste peut ne pas être complète comme attendu par l'utilisateur.

8.6.2.5 Extension de mise à jour de base

Cette extension est utilisable dans des listes dCRL pour indiquer la date et l'heure après lesquelles ce delta fournit des mises à jour du statut de révocation. Cette extension doit être utilisée exclusivement dans des listes dCRL qui contiennent l'extension **deltaCRLIndicator**. Une liste dCRL qui contient par contre l'extension **crlScope** ne nécessite pas cette extension, étant donné que le champ **baseThisUpdate** [*base de cette mise à jour*] de l'extension **crlScope** peut être utilisé dans le même but.

```
baseUpdateTime EXTENSION        ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-baseUpdateTime }
```

Cette extension est toujours non critique.

8.6.2.6 Extension de liste CRL la plus récente

Cette extension sera utilisée exclusivement comme extension de certificat; elle peut être utilisée dans des certificats émis à l'intention d'autorités ou d'utilisateurs. Ce champ indique la liste CRL à laquelle doit se référer un utilisateur de certificat pour obtenir les informations de révocation les plus récentes (par exemple, la dernière liste dCRL). Ce champ est défini comme suit:

```
freshestCRL EXTENSION     ::= {
    SYNTAX              CRLDistPointsSyntax
    IDENTIFIED BY       id-ce-freshestCRL }
```

Cette extension peut être critique ou non, au choix de l'émetteur du certificat. Si l'extension de liste CRL la plus récente est marquée comme critique, un système utilisant des certificats n'utilisera pas de certificat avant d'avoir extrait et vérifié la liste CRL la plus récente. Si l'extension est marquée comme non critique, le système utilisant des certificats peut alors employer des moyens locaux pour déterminer s'il est nécessaire ou non de vérifier la liste CRL la plus récente.

9 Relations entre la liste CRL delta et la liste de base

Une liste dCRL contient une extension **deltaCRLIndicator** ou une extension **crlScope** identifiant les informations de révocation de base qu'elle met à jour.

Si l'extension **deltaCRLIndicator** figure dans une liste dCRL, les informations de révocation de base mises à jour se constituent alors de la liste CRL de base à laquelle fait référence cette extension. La liste CRL de base référencée par une extension **deltaCRLIndicator** sera une liste CRL émise d'une manière complète pour son domaine d'application (c'est-à-dire, qui n'est pas elle-même une liste dCRL).

Si l'extension **crlScope** est présente et contient le composant **baseRevocationInfo** faisant référence aux informations de révocation de base mises à jour, il s'agit alors de la référence à un instant particulier à partir duquel cette liste dCRL fournit des mises à jour. Le composant **baseRevocationInfo** fait référence à une liste CRL qui peut avoir été émise sous forme complète ou non pour ce domaine d'application (c'est-à-dire qu'il se peut que la liste CRL référencée ait été émise sous la forme d'une liste dCRL). La liste dCRL qui contient le composant **baseRevocationInfo** met toutefois à jour les informations de révocation complètes pour le domaine d'application de la liste CRL référencée au moment de l'émission de cette dernière. L'utilisateur de certificat peut appliquer la liste dCRL pour une liste CRL complète pour le domaine d'application donné et qui a été émise au même instant ou après la liste CRL référencée dans la liste dCRL contenant le composant **baseRevocationInfo**.

Compte tenu de la possibilité de conflit entre leurs informations, une liste CRL ne contiendra pas simultanément l'extension **deltaCRLIndicator** et une extension **crlScope** avec le composant **baseRevocationInfo**. Une liste CRL peut contenir à la fois les extensions **deltaCRLIndicator** et **crlScope** uniquement si le composant **baseRevocationInfo** n'est pas présent dans l'extension **crlScope**.

Une liste dCRL peut également être une liste CRL indirecte dans la mesure où elle peut contenir des informations de révocation mises à jour concernant des listes CRL de base émises par une ou plusieurs autorités. L'extension **crlScope** sera utilisée pour indiquer qu'une liste CRL est une liste dCRL indirecte. L'extension **crlScope** contiendra une instance du composant **PerAuthorityScope** pour chaque liste CRL de base pour laquelle la liste dCRL indirecte fournit des informations mises à jour.

L'application d'une liste dCRL aux informations de révocation de base référencées doit tenir compte de manière précise du statut de révocation actuel.

- Une notification de révocation de certificat avec le motif de révocation **certificateHold** peut figurer, soit dans une liste dCRL, soit dans une liste CRL qui est complète pour un domaine donné. Ce code motif est prévu pour indiquer une révocation temporaire du certificat en attente d'une décision future de révocation permanente ou d'une réinstallation comme s'il n'avait pas été révoqué.

  - Si un certificat figurait dans une liste CRL comme révoqué avec le motif **certificateHold** (soit dans une liste dCRL, soit dans une liste CRL qui est complète pour un domaine d'application donné) avec un numéro **cRLNumber** égal à *n* et que la mise en attente est annulée par la suite, ce certificat doit alors figurer dans toutes les listes dCRL émises après l'annulation de la mise en attente et dont le numéro **cRLNumber** de la liste CRL de base référencée est inférieur ou égal à *n*. Le numéro de liste CRL d'une liste CRL de base référencée est égal, soit à la valeur du composant **BaseCRLNumber** de l'extension **deltaCRLIndicator**, soit à l'élément **cRLNumber** du composant **BaseRevocationInfo** de l'extension **cRLScope**, en fonction de l'extension utilisée pour indiquer que cette liste CRL est une liste dCRL. Le certificat doit figurer dans la liste avec un motif de révocation égal à **removeFromCRL**, à moins qu'il ne soit révoqué de nouveau par la suite avec l'un des motifs de révocation couverts par la liste dCRL, auquel cas il doit figurer dans la liste avec le motif de révocation pertinent pour la révocation ultérieure.

  - Si la mise en attente du certificat n'est pas annulée mais que ce dernier est révoqué de manière permanente, il doit alors figurer dans toutes les listes dCRL ultérieures pour lesquelles le numéro **cRLNumber** de la liste CRL de base référencée est inférieur au numéro **cRLNumber** de la liste CRL (soit une liste dCRL, soit une liste CRL qui est complète pour le domaine d'application donné) dans laquelle la notification de révocation permanente est apparue pour la première fois. Selon l'extension utilisée pour indiquer que cette liste CRL est une liste dCRL, le numéro de liste CRL d'une liste CRL de base référencée est égal, soit à la valeur du composant **BaseCRLNumber** de l'extension **deltaCRLIndicator**, soit à l'élément **cRLNumber** du composant **BaseRevocationInfo** de l'extension **cRLScope**.

- Une notification de révocation de certificat peut apparaître pour la première fois dans une liste dCRL et il est possible que la durée de validité du certificat expire avant l'émission de la prochaine liste CRL complète pour le domaine d'application considéré. La notification de révocation doit figurer dans un tel cas dans toutes les listes dCRL ultérieures, jusqu'à ce que la notification de révocation figure dans au moins une liste CRL émise qui est complète pour le domaine d'application de ce certificat.

Une liste CRL complète à l'instant actuel pour un domaine d'application donné peut être construite localement de l'une des manières suivantes:

- par extraction de la liste dCRL actuelle pour ce domaine d'application et en la combinant avec une liste CRL complète émise pour ce domaine d'application et dont le numéro **cRLNumber** est supérieur ou égal à celui qui figure dans la liste CRL de base référencée par la liste dCRL; ou

- par extraction de la liste dCRL actuelle pour ce domaine d'application et en la combinant avec une liste CRL complète pour ce domaine d'application qui a été construite localement à partir d'une liste dCRL dont le numéro **cRLNumber** est supérieur ou égal à celui qui figure dans la liste CRL de base référencée par la liste dCRL actuelle.

10 Procédure de traitement de l'itinéraire de certification

Le traitement de l'itinéraire de certification s'effectue dans un système qui a besoin d'utiliser la clé publique d'une entité finale distante, par exemple pour vérifier une signature numérique générée par une telle entité. Les politiques de certificat, les contraintes de base, les contraintes de nom et les extensions de contraintes de politique ont été conçues pour faciliter une implémentation automatisée et autonome de la logique de traitement de l'itinéraire de certification.

L'exposé sommaire qui suit présente une procédure de validation des itinéraires de certification. Une implémentation sera fonctionnellement équivalente au comportement externe résultant de cette procédure. L'algorithme utilisé par une implémentation particulière pour fournir les sorties correctes à partir des entrées données n'est pas normalisé.

10.1 Informations d'entrée du traitement d'itinéraire

Les informations d'entrée de la procédure de traitement de l'itinéraire de certification sont les suivantes:

a) un ensemble de certificats constituant un itinéraire de certification;

NOTE – Chaque certificat dans un ininéraire de certification est unique. Un itinéraire qui contient le même certificat deux fois ou plus n'est pas un itinéraire de certification valable.

b) une valeur fiable de clé publique ou d'identificateur de clé (si la clé est stockée de manière interne par le module de traitement de l'itinéraire de certification) utilisée pour vérifier le premier certificat de l'itinéraire de certification;

c) un ensemble *initial-policy-set* constitué d'un ou de plusieurs identificateurs de certificat de politique indiquant qu'une ou plusieurs politiques sont acceptables par l'utilisateur de certificat aux fins de traitement de l'itinéraire de certification; cet ensemble peut également prendre la valeur *any-policy*;

d) une valeur d'indicateur *initial-explicit-policy*, spécifiant si un identificateur de politique acceptable doit figurer de manière explicite dans le champ d'extension de politiques de certificat pour tous les certificats de l'itinéraire;

e) une valeur d'indicateur *initial-policy-mapping-inhibit* spécifiant si le mappage de politique est interdit sur l'itinéraire de certification;

f) une valeur d'indicateur *initial-inhibit-policy*, qui indique si la valeur spéciale **anyPolicy**, lorsqu'elle est présente dans l'extension des politiques de certification, est considérée correspondre à une valeur politique quelconque de certification spécifique dans un ensemble avec contraintes;

g) la date et l'heure actuelle (si ces dernières ne sont pas disponibles de manière interne dans le module de traitement de l'itinéraire de certification).

Les valeurs de c), d), e) et f) dépendront des prescriptions de politique du couple utilisateur-application qui utilise la clé publique certifiée de l'entité finale.

Il convient de noter que ces informations d'entrées individuelles du processus de validation d'itinéraire peuvent servir à un utilisateur de certificat pour limiter la confiance qu'il accorde à toute clé publique fiable pour un ensemble donné de politiques de certificat. Ceci peut se faire en s'assurant qu'une clé publique donnée constitue les informations d'entrée du processus uniquement lorsque l'ensemble initial de politiques contient des politiques dont l'utilisateur considère la clé publique comme fiable. Comme l'itinéraire de certification lui-même constitue une autre entrée du processus, cette vérification peut s'effectuer transaction par transaction.

10.2 Informations de sortie du traitement d'itinéraire

Les informations de sortie de la procédure sont les suivantes:

a) indication de réussite ou d'échec de la validation de l'itinéraire de certification;

b) en cas d'échec de la validation, un code diagnostic indiquant le motif de la défaillance;

c) l'ensemble de politiques imposées par l'autorité et leurs qualificatifs associés sous lesquelles l'itinéraire de certification est valide ou la valeur spéciale *any-policy*.

d) l'ensemble de politiques imposées par l'utilisateur constitué de l'intersection de l'ensemble *authorities-constrained-policy-set* et de l'ensemble *initial-policy-set*;

e) l'indicateur *explicit-policy-indicator* signalant si l'utilisateur du certificat ou une autorité située sur l'itinéraire exige qu'une politique acceptable soit identifiée dans chaque certificat de l'itinéraire;

f) les détails de tout mappage de politique qui a été rencontré lors du traitement de l'itinéraire de certification.

NOTE – En cas de réussite de la validation, le système utilisant des certificats peut toutefois décider de ne pas employer le certificat en fonction des valeurs de qualificatif de politique ou d'autres informations figurant dans le certificat.

10.3 Variables de traitement d'itinéraire

La procédure utilise les variables d'état suivantes:

a) *authorities-constrained-policy-set* (ensemble de politiques imposées par l'autorité): table des qualificatifs et des identificateurs de politiques extraits des certificats de l'itinéraire de certification (les lignes y représentent les politiques, leurs qualificatifs et les historiques de mappage, et les colonnes les certificats et l'itinéraire de certification);

b) *permitted-subtrees* (sous-arbres autorisés): ensemble de spécifications de sous-arbre définissant les sous-arbres auxquels doit appartenir tout nom de sujet figurant dans des certificats suivants dans l'itinéraire de certification, ou la valeur spéciale *unbounded* (non lié);

c) *excluded-subtrees* (sous-arbres interdits): ensemble (éventuellement vide) de spécifications de sous-arbre (comportant chacune un nom de base de sous-arbre et des indicateurs de niveau minimal et maximal) définissant des sous-arbres auxquels ne doit pas appartenir tout nom de sujet figurant dans les certificats suivants dans l'itinéraire de certification;

d) *explicit-policy-indicator* (indicateur de politique explicite): indique si une politique acceptable doit figurer de manière explicite dans chaque certificat;

e) *path-depth* (profondeur de l'itinéraire): nombre entier égal au nombre de certificats sur l'itinéraire de certification, augmenté de un, pour lesquels le traitement a été effectué;

f) *policy-mapping-inhibit-indicator* (indicateur d'interdiction de mappage de politique): indique si le mappage de politique est interdit;

g) *inhibit-any-policy-indicator*: indique si la valeur spéciale **anyPolicy** correspond à une politique de certification spécifique;

h) *pending-constraints* (contraintes en attente): détails des contraintes de politique explicite et/ou d'interdiction de mappage de politique, qui ont été stipulées mais qui doivent prendre effet sur la suite de l'itinéraire. Il s'agit de deux indicateurs d'un bit appelés *explicit-policy-pending* (politique explicite en attente) et *policy-mapping-inhibit-pending* (interdiction de mappage de politique en attente) et *inhibit-any-policy-pending* ainsi que pour chacun d'eux, d'un entier appelé *skip-certificates* qui donne le nombre de certificats restant à ignorer avant que les contraintes prennent effet.

10.4 Etape d'initialisation

La procédure implique une étape d'initialisation suivie d'une série d'étapes de traitement. L'étape d'initialisation comprend les actions suivantes:

a) remplir les colonnes de rang zéro et un du tableau *authorities-constrained-policy-set* avec la valeur *any-policy*;

b) initialisation de la variable *permitted-subtrees* avec la valeur *unbounded*;

c) initialisation de la variable *excluded-subtrees* avec l'ensemble vide;

d) initialisation de l'indicateur *explicit-policy-indicator* avec la valeur *initial-explicit-policy*;

e) initialisation de la variable *path-depth* avec la valeur un;

f) initialisation de l'indicateur *policy-mapping-inhibit-indicator* avec la valeur *initial-policy-mapping-inhibit*;

g) initialisation de l'indicateur *inhibit-any-policy-indicator* avec la valeur *initial-inhibit-policy*;

h) initialisation des trois indicateurs *pending-constraints* sur "non positionné".

10.5 Traitement de certificat

10.5.1 Vérification de base des certificats

Les certificats sont ensuite traités un par un en, commençant par le certificat utilisant la clé publique fiable d'entrée. Le dernier certificat est considéré comme étant le certificat final; tous les autres certificats sont considérés comme étant des certificats intermédiaires.

Les vérifications suivantes s'appliquent à un certificat:

a) vérifier que la signature est correcte, que les dates sont valides, que la succession des noms de sujet de certificat et d'émetteur de certificat est correcte et que le certificat n'a pas été révoqué;

b) si la contrainte d'extension de base est présente dans un certificat intermédiaire, vérifier que le composant **cA** est présent et positionné sur "Vrai". Si le composant **pathLenConstraint** est présent, vérifier que l'itinéraire de certification actuel respecte cette contrainte de longueur (en ignorant les certificats intermédiaires auto-émis);

c) si l'extension de politiques de certificat n'est pas présente, positionner alors l'ensemble *authorities-constrained-policy-set* sur l'ensemble vide en supprimant toutes les lignes dans le tableau *authorities-constrained-policy-set*;

d) si l'extension de politiques de certificat est présente, alors pour chaque politique *P*, dans l'extension autre que **anyPolicy**, adjoindre les qualificateurs de politique associés à *P*, dans chaque rangée du tableau *authority-constrained-policy-set* dont les entrées dans la colonne [*path-depth*] contiennent la valeur *P*. Si aucune rangée du tableau précité ne contient *P* dans son entrée colonne [*path-depth*] mais que la valeur dans *authorities-constrained-policy-set* [0, *path-depth*] est *anyPolicy*, ajouter alors une nouvelle rangée dans le tableau en reproduisant la rangée 0 et en écrivant l'identificateur de politique *P* avec ses qualificatifs dans l'entrée colonne [*path-depth*] de la nouvelle rangée;

e) si l'extension politique de certification est présente mais n'inclut pas la valeur **anyPolicy** ou si l'indicateur *inhibit-any-policy-indicator* est fixé, supprimer alors toute rangée pour laquelle l'entrée colonne [*path-depth*] contient la valeur *anyPolicy* ainsi que toute rangée pour laquelle l'entrée colonne [*path-depth*] ne contient pas de valeur dans l'extension politique de certification;

f) si l'extension politique de certification est présente et inclut la valeur **anyPolicy** et que l'indicateur *inhibit-any-policy-indicator* n'est pas positionné, adjoindre alors les qualificatifs de politique associés à **anyPolicy** à chaque rangée dans le tableau *authorities-constrained-policy-set* dont l'entrée colonne [*path-depth*] contient la valeur *anyPolicy* ou contient une valeur qui n'apparaît pas dans l'extension politique de certification;

g) si le certificat n'est pas un certificat intermédiaire auto-émis, vérifier que le nom du sujet appartient à l'espace de noms indiqué par la valeur de *permitted-subtrees* et n'appartient pas à l'espace de noms indiqué par la valeur *excluded-subtrees*.

10.5.2 Traitement des certificats intermédiaires

Les actions suivantes d'enregistrement de contrainte sont effectuées ensuite, dans le cas d'un certificat intermédiaire, afin de positionner la valeur correcte des variables d'état pour le traitement du certificat suivant:

a) si l'extension **nameConstraints** (contraintes de nom) est présente dans le certificat avec un composant **permittedSubtrees** (sous-arbres autorisés), remplacer alors la valeur de la variable d'état *permitted-subtrees* par l'intersection de sa valeur précédente avec la valeur indiquée dans l'extension de certificat;

b) si l'extension **nameConstraints** est présente dans le certificat avec un composant **excludedSubtrees** (sous-arbres interdits), remplacer alors la valeur de la variable d'état *excluded-subtrees* par l'union de sa valeur précédente avec la valeur indiquée dans l'extension de certificat;

c) si l'indicateur *policy-mapping-inhibit-indicator* est positionné:

- traiter toute extension de mappage de politique, pour chaque mappage identifié dans l'extension, en localisant toutes les lignes dans le tableau *authorities-constrained-policy-set* dont l'élément dans la colonne de rang [*path-depth*] est égal à la valeur de la politique de domaine de l'émetteur et en supprimant la colonne;

d) si l'indicateur *policy-mapping-inhibit-indicator* n'est pas positionné:

- traiter de la manière suivante toute extension de mappage de politique, pour tout mappage figurant dans l'extension: localiser toutes les lignes dans le tableau *authorities-constrained-policy-set* dont l'élément de la colonne de rang [*path-depth*] est égal à la valeur de politique du domaine émetteur figurant dans l'extension et copier la valeur de la politique du domaine sujet figurant dans l'extension dans l'élément de la colonne de rang [*path-depth*+1] de la même ligne. Si l'extension mappe une politique de domaine émetteur avec plusieurs politiques de domaine sujet, la ligne concernée doit alors être copiée et le nouvel élément ajouté à chaque colonne. Si la valeur de l'élément *authorities-constrained-policy-set*[0, *path-depth*] est égale à *any-policy*, copier alors tout identificateur de politique de domaine émetteur, pour l'extension de mappages de politique, dans la colonne de rang [*path-depth*], en dupliquant les lignes si nécessaire et en conservant les qualificatifs s'ils sont présents, et copier la valeur de politique du domaine sujet de l'extension dans l'élément de la colonne de rang [*path-depth*+1] de la même ligne;
- si l'indicateur *policy-mapping-inhibit-pending* est positionné et si le certificat n'est pas auto-émis, décrémenter alors la valeur correspondante de *skip-certificates* et positionner l'indicateur *policy-mapping-inhibit-indicator* si cette valeur devient nulle;
- procéder comme suit si la contrainte **inhibitPolicyMapping** figure dans le certificat. Positionner l'indicateur *policy-mapping-inhibit-indicator* si la valeur du composant **SkipCerts** (certificats ignorés) est nulle. Pour toute autre valeur du composant **SkipCerts**, positionner l'indicateur *policy-mapping-inhibit-pending* et positionner la valeur correspondante de *skip-certificates* sur la plus petite des valeurs du composant **SkipCerts** et de la valeur précédente de la variable *skip-certificates* (si l'indicateur *policy-mapping-inhibit-pending* était déjà positionné);

e) pour toute ligne non modifiée dans l'étape c) ou d) ci-dessus (et pour toute ligne dans le cas où aucune extension de mappage ne figure dans le certificat), copier la valeur de l'identificateur de politique de la colonne de rang [*path-depth*] dans la colonne de rang [*path-depth*+1] de la ligne;

f) Si l'indicateur *inhibit-any-policy-indicator* n'est pas positionné:

- si l'indicateur *inhibit-any-policy-pending* est positionné, décrémenter la valeur *skip-certificates* correspondante et positionner l'indicateur *inhibit-any-policy-indicator* si cette valeur devient nulle;

- procéder comme suit si la contrainte **inhibitAnyPolicy** est présente dans le certificat. Pour une valeur du composant **SkipCerts** nulle, positionner l'indicateur *inhibit-any-policy-indicator* sur zéro. Pour toute autre valeur de **SkipCerts**, positionner l'indicateur *inhibit-any-policy-pending* et remplacer la valeur correspondante de *skip-certificates* par la plus petite des valeurs du composant **SkipCerts** et de la valeur précédente de la variable *skip-certificates* (si l'indicateur *inhibit-any-policy-pending* était déjà positionné).

g) incrémenter la variable *path-depth*.

10.5.3 Traitement des indicateurs de politique explicite

Les actions suivantes sont ensuite effectuées pour tous les certificats:

a) si l'indicateur *explicit-policy-indicator* n'est pas positionné:

- si l'indicateur *explicit-policy-pending* est positionné et si le certificat n'est pas un certificat intermédiaire auto-émis, décrémenter la valeur *skip-certificates* correspondante et positionner l'indicateur *explicit-policy-indicator* si cette valeur devient nulle;

- si la contrainte **requireExplicitPolicy** est présente dans le certificat, exécuter ce qui suit: Pour une valeur du composant **SkipCerts** nulle, positionner l'indicateur *explicit-policy-pending-indicator* sur zéro. Pour toute autre valeur de **SkipCerts**, positionner l'indicateur *explicit-policy-pending* et remplacer la valeur correspondante de *skip-certificates* par la plus petite des valeurs du composant **SkipCerts** et la valeur précédente de la variable *skip-certificates* (si l'indicateur *explicit-policy-pending* était déjà positionné)

- si la composante **requireExplicitPolicy** est présente, et que le chemin de certification contient un certificat émis par une autorité de certification désignée, il est nécessaire que tous les certificats du chemin contiennent, dans l'extension relative aux politiques de certification, un identificateur de politique acceptable. Cet identificateur est celui de la politique de certification requise par l'utilisateur du chemin de certification, celui d'une politique qui a été déclarée équivalente à la première, grâce à la fonction de mappage de politiques, ou la valeur spéciale *any-policy*. L'autorité de certification désignée est soit l'autorité émettrice du certificat contenant cette extension (si la valeur de la composante **requireExplicitPolicy** est zéro) soit une autorité de certification qui est titulaire d'un certificat subséquent dans le chemin de certification (tel qu'indiqué par une valeur non nulle).

10.5.4 Traitement final

Les actions suivantes sont effectuées pour les certificats d'entité finale:

a) vérifier alors que le tableau *authorities-constrained-policy-set* n'est pas vide si l'indicateur *explicit-policy-indicator* est positionné. Si l'une quelconque des vérifications précédentes échoue, la procédure se termine alors en renvoyant une indication d'échec avec un code motif adéquat, l'indicateur *explicit-policy-indicator* et des valeurs nulles dans l'ensemble *user-constrained-policy-set* et dans le tableau *authorities-constrained-policy-set*.

Si aucun des contrôles ci-dessus ne devait échouer pour le certificat final, l'ensemble *user-constrained-policy-set* doit être calculé en formant l'intersection de l'ensemble *authorities-constrained-policy-set* et *initial-policy-set* Si *authorities-constrained-policy-set[0, path-depth]* est *any-policy*, alors, *authorities-constrained-policy-set* est *any-policy*. Dans les autres cas, *authorities-constrained-policy-set* est, pour chaque rangée du tableau, égal à la valeur de la cellule la plus à gauche qui ne contient pas l'identificateur *any-policy*. La procédure doit ensuite prendre fin, en renvoyant une indication de réussite avec le *explicit-policy-indicator*, le tableau *authorities-constrained-policy-set* et le *user-constrained-policy-set*. Si l'intersection de l'ensemble *authorities-constrained-policy-set* et *user-constrained-set* est nulle, l'itinéraire est valide dans le cadre de la politique ou des politiques contraintes par l'autorité, mais aucune n'est acceptable pour l'utilisateur.

11 Schéma d'annuaire d'infrastructures PKI

Cet article définit les éléments de schéma d'annuaire utilisés pour représenter les informations d'infrastructure PKI dans l'annuaire. Il contient la spécification des classes d'objets, des attributs et des valeurs de règles de concordance d'attribut pertinentes.

11.1 Classes d'objets et formes de nom d'annuaire d'infrastructure PKI

Ce paragraphe contient les définitions des classes d'objets utilisées pour représenter les objets d'infrastructure PKI dans l'annuaire.

11.1.1 Classe d'objets "utilisateur d'infrastructure PKI"

La classe d'objets "utilisateur d'infrastructure PKI" sert à définir des entrées pour des objets pouvant être le sujet de certificats de clé publique.

```
pkiUser    OBJECT-CLASS   ::= {
    SUBCLASS OF        {top}
    KIND               auxiliary
    MAY CONTAIN        {userCertificate}
    ID                 id-oc-pkiUser }
```

11.1.2 Classe d'objets "autorité de certification d'infrastructure PKI"

La classe d'objets "autorité de certification d'infrastructure PKI" sert à définir des objets qui jouent le rôle d'autorités de certification.

```
pkiCA      OBJECT-CLASS   ::= {
    SUBCLASS OF        {top}
    KIND               auxiliary
    MAY CONTAIN        {cACertificate |
                       certificateRevocationList |
                       authorityRevocationList |
                       crossCertificatePair }
ID                     id-oc-pkiCA }
```

11.1.3 Classe d'objets et forme de nom de points de répartition de liste CRL

La classe d'objets "point de répartition de liste CRL" sert à définir des entrées pour des objets qui peuvent jouer le rôle de points de répartition de liste CRL.

```
cRLDistributionPoint       OBJECT-CLASS       ::= {
    SUBCLASS OF            { top }
    KIND                   structural
    MUST CONTAIN           { commonName }
    MAY CONTAIN            { certificateRevocationList |
                           authorityRevocationList |
                           deltaRevocationList }
    ID                     id-oc-cRLDistributionPoint }
```

La forme de nom "point de répartition de liste CRL" spécifie la manière dont peuvent être nommées les entrées appartenant à la classe d'objets **cRLDistributionPoint**.

```
cRLDistPtNameForm          NAME-FORM ::= {
    NAMES                  cRLDistributionPoint
    WITH ATTRIBUTES        { commonName}
    ID                     id-nf-cRLDistPtNameForm }
```

11.1.4 Classe d'objets "liste CRL delta"

La classe d'objets "liste CRL delta" sert à définir des entrées pour des objets qui contiennent des listes delta de révocation (par exemple, des autorités de certification, des autorités d'attribut, etc.).

```
deltaCRL                   OBJECT-CLASS   ::= {
    SUBCLASS OF            {top}
    KIND                   auxiliary
    MAY CONTAIN            {deltaRevocationList}
    ID                     id-oc-deltaCRL }
```

11.1.5 Classe d'objets "politique de certificat et déclaration de pratique de certification"

La classe d'objets "politique de certificat et déclaration de pratique de certification" (CP CPS) sert à définir des entrées pour des objets qui contiennent des politiques de certificat et/ou des informations de pratique de certification.

```
cpCps                OBJECT-CLASS   ::= {
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      {certificatePolicy |
                     certificationPracticeStmt}
    ID               id-oc-cpCps }
```

11.1.6 Classe d'objets "itinéraire de certificat d'infrastructure PKI"

La classe d'objets "itinéraire de certificat d'infrastructure PKI" sert à définir des entrées pour des objets qui contiennent des itinéraires d'infrastructure PKI. Elle sera utilisée en général conjointement aux entrées d'objets de classe de structure d'itinéraires **pkiCA** [*infrastructure PKI d'autorité de certification*].

```
pkiCertPath          OBJECT-CLASS   ::= {
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      { pkiPath }
    ID               id-oc-pkiCertPath }
```

11.2 Attributs "répertoire d'infrastructure PKI"

Ce paragraphe contient la définition d'attributs de répertoire permettant de stocker des éléments d'information d'infrastructure PKI dans l'annuaire.

11.2.1 Attribut "certificat d'utilisateur"

Un utilisateur peut obtenir un ou plusieurs certificats de clé publique émis par une ou plusieurs autorités de certification. Le type d'attribut **userCertificate** [*certificat d'utilisateur*] contient les certificats de clé publique obtenus par un utilisateur auprès d'une ou plusieurs autorités de certification.

```
userCertificate                  ATTRIBUTE   ::=   {
    WITH SYNTAX                  Certificate
    EQUALITY MATCHING RULE       certificateExactMatch
    ID                           id-at-userCertificate}
```

11.2.2 Attribut "certificat d'autorité de certification"

L'attribut **cACertificate** [*certificat d'autorité de certification*] d'une entrée d'annuaire d'autorité de certification sera utilisé pour stocker des certificats émis par cette autorité à sa propre intention (s'il en existe) et des certificats émis pour cette autorité de certification par des autorités de certification situées dans son domaine. Dans le cas de certificats de version 3, ces derniers contiendront une extension **basicConstraints** avec une valeur de composant **cA** positionnée sur "Vrai". La définition du domaine est strictement une question de politique locale.

```
cACertificate                    ATTRIBUTE   ::=   {
    WITH SYNTAX                  Certificate
    EQUALITY MATCHING RULE       certificateExactMatch
    ID                           id-at-cAcertificate }
```

11.2.3 Attribut "paire de certificats croisés"

Les élément **issuedToThisCA** [*émis à l'intention de la présente autorité de certification*] de l'attribut **crossCertificatePair** [*paire de certificats croisés*] d'une entrée d'autorité de certification dans l'annuaire seront utilisés pour stoker tous les certificats émis par une autorité de certification, à l'exception de ceux qu'elle a émis à sa propre intention. Les éléments **issuedByThisCA** [*émis par la présente autorité de certification*] de l'attribut **crossCertificatePair** d'une entrée d'annuaire d'autorité de certification peuvent éventuellement contenir un sous-ensemble de certificats émis par cette autorité de certification à l'intention d'autres autorités de certification. Si une autorité de certification émet un certificat à l'intention d'une autre autorité de certification, et si celle-ci n'est pas de rang hiérarchiquement inférieur à l'autorité de certification émettrice, cette dernière doit mettre ledit certificat dans l'élément **issuedByThisCA** de l'attribut **crossCertificatePair** de sa propre entrée d'annuaire. Lorsque les éléments **issuedToThisCA** et **issuedByThisCA** sont présents simultanément dans une même valeur d'attribut, le nom de l'émetteur de l'un des certificats doit correspondre au nom du destinataire de l'autre et réciproquement; la clé publique du destinataire de l'un des certificats permettra de vérifier la signature numérique de l'autre et réciproquement. Dans les éditions précédentes, le terme **forward** était utilisé à la place de **issuedToThisCA** et le terme **reverse** était utilisé à la place de **issuedByThisCA**.

Lorsqu'un élément **issuedByThisCA** est présent, les valeurs des éléments **issuedToThisCA** et **issuedByThisCA** ne sont pas nécessairement stockées dans la même valeur d'attribut; elles peuvent être stockées, soit dans une seule valeur d'attribut, soit dans deux valeurs d'attribut.

Dans le cas de certificats de version 3, ces derniers contiendront une extension **basicConstraints** avec une valeur de composant **cA** positionnée sur "Vrai".

```
crossCertificatePair                ATTRIBUTE  ::=  {
    WITH SYNTAX                     CertificatePair
    EQUALITY MATCHING RULE          certificatePairExactMatch
    ID                              id-at-crossCertificatePair }

CertificatePair         ::=     SEQUENCE {
issuedToThisCA          [0]     Certificate OPTIONAL,
issuedByThisCA          [1]     Certificate OPTIONAL
                                -- au moins l'une des paires sera présente -- }
(WITH COMPONENTS {..., issuedToThisCA PRESENT} |
WITH COMPONENTS {..., issuedByThisCA PRESENT})
```

11.2.4 Attribut "liste de révocation de certificat"

L'attribut suivant contient une liste de certificats révoqués.

```
certificateRevocationList           ATTRIBUTE  ::=  {
    WITH SYNTAX                     CertificateList
    EQUALITY MATCHING RULE          certificateListExactMatch
    ID                              id-at-certificateRevocationList }
```

11.2.5 Attribut "liste de révocation d'autorité"

L'attribut suivant contient une liste de certificats d'autorité révoqués.

```
authorityRevocationList             ATTRIBUTE  ::=  {
    WITH SYNTAX                     CertificateList
    EQUALITY MATCHING RULE          certificateListExactMatch
    ID                              id-at-authorityRevocationList }
```

11.2.6 Attribut "liste delta de révocation"

Le type d'attribut suivant est défini pour le stockage d'une liste dCRL dans une entrée d'annuaire:

```
deltaRevocationList                 ATTRIBUTE ::= {
    WITH SYNTAX                     CertificateList
    EQUALITY MATCHING RULE          certificateListExactMatch
    ID                              id-at-deltaRevocationList }
```

11.2.7 Attribut "algorithmes pris en charge"

Un attribut de l'annuaire est défini pour la prise en charge du choix d'un algorithme utilisable lors de la communication avec une entité finale distante qui utilise des certificats tels qu'ils sont définis dans la présente Spécification d'annuaire. Les règles ASN.1 suivantes définissent cet attribut (pouvant prendre plusieurs valeurs):

```
supportedAlgorithms ATTRIBUTE ::= {
    WITH SYNTAX                     SupportedAlgorithm
    EQUALITY MATCHING RULE          algorithmIdentifierMatch
    ID                              id-at-supportedAlgorithms }

SupportedAlgorithm ::= SEQUENCE {
    algorithmIdentifier                     AlgorithmIdentifier,
    intendedUsage                   [0]     KeyUsage OPTIONAL,
    intendedCertificatePolicies     [1]     CertificatePoliciesSyntax OPTIONAL }
```

Chacune des valeurs multiples de l'attribut possédera une valeur d'identificateur **algorithmIdentifier** [*identificateur d'algorithme*] distincte. La valeur du composant **intendedUsage** [*utilisation prévue*] fournit une indication concernant l'utilisation prévue pour l'algorithme (voir 8.2.2.3 en ce qui concerne les utilisations reconnues). La valeur du composant **intendedCertificatePolicies** [*politiques de certificat prévues*] indique les politiques de certificat et, de manière optionnelle, les qualificatifs de certificat de politique avec lesquels l'algorithme en question peut être utilisé.

11.2.8 Attribut "déclaration de pratique de certification"

L'attribut **certificationPracticeStmt** [*déclaration de pratique de certification*] est utilisé pour stocker des informations concernant la déclaration de pratique de certification d'une autorité.

```
certificationPracticeStmt      ATTRIBUTE   ::=   {
    WITH SYNTAX         InfoSyntax
    ID                  id-at-certificationPracticeStmt }

InfoSyntax          ::=         CHOICE {
    content         DirectoryString {ub-content},
    pointer         SEQUENCE {
        name                        GeneralNames,
        hash                        HASH { HashedPolicyInfo } OPTIONAL } }

POLICY    ::=   TYPE-IDENTIFIER

HashedPolicyInfo        ::=    POLICY.&Type( {Policies} )

Policies POLICY ::= {...} -- définies par les réalisateurs --
```

Le composant **content** [*contenu*] contient, s'il est présent, la totalité de l'énoncé de la déclaration de pratique de certification de l'autorité.

Si le composant **pointer** [*pointeur*] est présent, le composant **name** [*nom*] fait alors référence à un ou plusieurs emplacements au niveau desquels peut être obtenue une copie de la déclaration de pratique de certification de l'autorité. Le composant **hash** [*hachage*] contient, s'il est présent, un hachage du contenu de la déclaration de pratique de certification qui doit se trouver à l'emplacement de référence. Ce hachage peut être utilisé pour effectuer une vérification d'intégrité du document de référence.

11.2.9 Attribut "politique de certificat"

L'attribut **certificatePolicy** [*politique de certificat*] est utilisé pour stocker des informations concernant une politique de certificat.

```
certificatePolicy          ATTRIBUTE   ::=   {
    WITH SYNTAX         PolicySyntax
    ID                  id-at-certificatePolicy }

PolicySyntax    ::=         SEQUENCE {
    policyIdentifier        PolicyID,
    policySyntax            InfoSyntax
}

PolicyID    ::=   CertPolicyId
```

Le composant **policyIdentifier** contient l'identificateur d'objet enregistré pour la politique de certificat concernée.

Le composant **content** contient, s'il est présent, la totalité de l'énoncé de la politique de certificat.

Si le composant **pointer** est présent, le composant **name** fait alors référence à un ou plusieurs emplacements au niveau desquels peut être obtenue une copie de la politique de certificat. Le composant **hash** contient, s'il est présent, un hachage du contenu de politique de certificat qui doit se trouver à l'emplacement de référence. Ce hachage peut être utilisé pour effectuer une vérification d'intégrité du document de référence.

11.2.10 Attribut "itinéraire d'infrastructure PKI"

L'attribut "itinéraire d'infrastructure PKI" est utilisé pour stocker des itinéraires de certification, constitués chacun d'une succession de certificats croisés.

```
pkiPath    ATTRIBUTE   ::= {
    WITH SYNTAX         PkiPath
    ID                  id-at-pkiPath }

PkiPath    ::=   SEQUENCE OF CrossCertificates
```

Cet attribut peut être stocké dans l'entrée d'annuaire de l'autorité de certification et contenir certains itinéraires de certification partant de cette autorité vers d'autres autorités de certification. Cet attribut permet, s'il est utilisé, une extraction plus efficace des certificats croisés qui constituent des itinéraires de certification d'utilisation fréquente. Il n'existe pas de prescriptions d'utilisation propres à cet attribut et l'ensemble des valeurs qu'il stocke ne représentera probablement pas la totalité des itinéraires de certification directs pour toute autorité de certification donnée.

11.3 Règles de concordance d'annuaire d'infrastructure PKI

La présente Spécification d'annuaire définit les règles de concordance devant être utilisées avec des attributs du type **Certificate, CertificatePair**, **CertificateList, CertificatePolicy** et **SupportedAlgorithm** [respectivement *certificat, paire de certificats, liste de certificat, politique de certificat* et *algorithme pris en charge*]. Ce paragraphe définit également les règles de concordance facilitant le choix de certificats ou de listes CRL qui possèdent des caractéristiques spécifiques, parmi des certificats ou des listes CRL utilisant des attributs avec des valeurs multiples.

11.3.1 Concordance exacte de certificat

La règle de concordance exacte de certificat compare une valeur présentée avec la valeur d'un attribut du type **Certificate**. Elle choisit de manière non ambiguë un certificat unique.

```
certificateExactMatch MATCHING-RULE ::= {
    SYNTAX      CertificateExactAssertion
    ID          id-mr-certificateExactMatch }

CertificateExactAssertion ::= SEQUENCE {
    serialNumber    CertificateSerialNumber,
    issuer          Name }
```

Cette règle de concordance renvoie la valeur "Vrai" si les composants de la valeur d'attribut concordent avec la valeur d'attribut présentée.

11.3.2 Concordance de certificat

La règle de concordance de certificat compare une valeur présentée avec une valeur d'attribut du type **Certificate**. Elle sélectionne un ou plusieurs certificats en fonction de caractéristiques diverses.

```
certificateMatch MATCHING-RULE ::= {
    SYNTAX      CertificateAssertion
    ID          id-mr-certificateMatch }

CertificateAssertion ::= SEQUENCE {
    serialNumber            [0]     CertificateSerialNumber     OPTIONAL,
    issuer                  [1]     Name                        OPTIONAL,
    subjectKeyIdentifier    [2]     SubjectKeyIdentifier        OPTIONAL,
    authorityKeyIdentifier  [3]     AuthorityKeyIdentifier      OPTIONAL,
    certificateValid        [4]     Time                        OPTIONAL,
    privateKeyValid         [5]     GeneralizedTime             OPTIONAL,
    subjectPublicKeyAlgID   [6]     OBJECT IDENTIFIER           OPTIONAL,
    keyUsage                [7]     KeyUsage                    OPTIONAL,
    subjectAltName          [8]     AltNameType                 OPTIONAL,
    policy                  [9]     CertPolicySet               OPTIONAL,
    pathToName              [10]    Name                        OPTIONAL,
    subject                 [11]    Name                        OPTIONAL,
    nameConstraints         [12]    NameConstraintsSyntax       OPTIONAL
}

AltNameType ::= CHOICE {
    builtinNameForm     ENUMERATED {
                        rfc822Name              (1),
                        dNSName                 (2),
                        x400Address             (3),
                        directoryName           (4),
                        ediPartyName            (5),
                        uniformResourceIdentifier (6),
                        iPAddress               (7),
                        registeredId            (8) },
    otherNameForm       OBJECT IDENTIFIER }

CertPolicySet ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId
```

Cette règle de concordance renvoie la valeur "Vrai" si tous les composants figurant dans la valeur présentée concordent de la manière suivante avec les composants figurant dans la valeur de l'attribut concerné:

le composant **serialNumber** est en concordance si la valeur de ce composant dans la valeur de l'attribut est égale à la valeur présentée;

le composant **issuer** est en concordance si la valeur de ce composant dans la valeur de l'attribut est égale à la valeur présentée;

le composant **subjectKeyIdentifier** [*identificateur de clé de sujet*] est en concordance si la valeur de ce composant dans la valeur de l'attribut stocké est égale à la valeur présentée; il n'y a pas concordance si la valeur de l'attribut stocké ne contient pas d'extension d'identificateur de clé de sujet;

le composant **authorityKeyIdentifier** [*identificateur de clé d'autorité*] est en concordance si la valeur de ce composant dans la valeur de l'attribut stocké est égale à la valeur présentée; il n'y a pas concordance si la valeur de l'attribut stocké ne contient pas d'extension d'identificateur de clé d'autorité ou si les composants de la valeur présentée ne figurent pas tous dans la valeur de l'attribut stocké;

le composant **certificateValid** [*validité du certificat*] est en concordance si la valeur présentée appartient à la durée de validité de la valeur de l'attribut stocké;

le composant **privateKeyValid** [*validité de la clé privée*] est en concordance si la valeur présentée appartient à la durée de validité indiquée par l'extension de durée d'utilisation de clé privée de la valeur de l'attribut stocké ou si une telle extension ne figure pas dans la valeur de l'attribut stocké;

le composant **subjectPublicKeyAlgID** [*identificateur d'algorithme de clé publique du sujet*] est en concordance en cas d'égalité avec le composant **algorithmIdentifier** du composant **subjectPublicKeyInformation** [*information de clé publique du sujet*] de la valeur de l'attribut stocké;

le composant **keyUsage** [*utilisation de la clé*] est en concordance si tous les bits positionnés dans la valeur présentée sont également positionnés dans l'extension d'utilisation de clé de la valeur de l'attribut stocké ou s'il n'existe pas de telle extension dans la valeur de l'attribut stocké;

le composant **subjectAltName** [*autre nom du sujet*] est en concordance si la valeur de l'attribut stocké contient l'extension d'autre nom de sujet avec un composant **AltNames** [*autres noms*] du même type de nom que celui qui est indiqué dans la valeur présentée;

le composant **policy** est en concordance si au moins l'un des éléments de l'ensemble **CertPolicySet** [*ensemble de politiques de certificat*] présenté figure dans l'extension de politiques de certificat dans la valeur de l'attribut stocké ou si, soit la valeur stockée, soit la valeur présentée, contient la valeur spéciale **any-policy** dans le composant **policy** [*politique*]. Il n'y a pas concordance s'il n'existe pas d'extension de politiques de certificat normalisée dans la valeur de l'attribut stocké;

le composant **pathToName** [*itinéraire vers le nom*] est en concordance, sauf si le certificat possède une extension de contraintes de nom qui interdit la construction d'un itinéraire de certification vers la valeur de nom présentée;

le composant **subject** est en concordance si la valeur de ce composant dans la valeur de l'attribut est égale à celle dans la valeur présentée;

le composant **nameConstraints** est en concordance si les noms de sujet dans la valeur de l'attribut stocké appartiennent à l'espace de noms indiqué par la valeur du composant **permitted-subtrees** de la valeur présentée et n'appartiennent pas à l'espace de noms indiqué par la valeur du composant **excluded-subtrees** de la valeur présentée.

11.3.3 Concordance exacte de paire de certificats

La règle de concordance exacte de paire de certificats compare une valeur présentée avec une valeur d'attribut du type **CertificatePair**. Elle sélectionne de manière non ambiguë une paire unique de certificats croisés.

```
certificatePairExactMatch MATCHING-RULE          ::=  {
    SYNTAX     CertificatePairExactAssertion
    ID         id-mr-certificatePairExactMatch }

CertificatePairExactAssertion ::= SEQUENCE {
    issuedToThisCAAssertion     [0] CertificateExactAssertion OPTIONAL,
    issuedByThisCAAssertion     [1] CertificateExactAssertion OPTIONAL }
    ( WITH COMPONENTS                {..., issuedToThisCAAssertion PRESENT} |
    WITH COMPONENTS                  {..., issuedByThisCAAssertion PRESENT} )
```

Cette règle de concordance renvoie la valeur "Vrai" si les composants figurant dans les composants **issuedToThisCAAssertion** et **issuedByThisCAAssertion** de la valeur présentée correspondent respectivement aux composants adéquats des composants **issuedToThisCA** et **issuedByThisCA** dans la valeur de l'attribut stocké.

11.3.4 Concordance de paire de certificats

La règle de concordance de paire de certificats compare une valeur présentée avec une valeur d'attribut du type **CertificatePair**. Elle sélectionne une ou plusieurs paires de certificats croisés en fonction de diverses caractéristiques du certificat **issuedToThisCA** ou **issuedByThisCA** de la paire.

```
certificatePairMatch MATCHING-RULE ::=  {
    SYNTAX      CertificatePairAssertion
    ID          id-mr-certificatePairMatch }

CertificatePairAssertion ::= SEQUENCE {
    issuedToThisCAAssertion     [0] CertificateAssertion OPTIONAL,
    issuedByThisCAAssertion     [1] CertificateAssertion OPTIONAL }
    ( WITH COMPONENTS               {..., issuedToThisCAAssertion PRESENT} |
    WITH COMPONENTS                 {..., issuedByThisCAAssertion PRESENT} )
```

Cette règle de concordance renvoie la valeur "Vrai" si tous les composants qui figurent dans les composants **issuedToThisCAAssertion** et **issuedByThisCAAssertion** de la valeur présentée correspondent respectivement aux composants adéquats des composants **issuedToThisCA** et **issuedByThisCA** de la valeur de l'attribut stocké en utilisant les règles données.

11.3.5 Concordance exacte de liste de certificats

La règle de concordance exacte de liste de certificats compare une valeur présentée avec une valeur d'attribut du type **CertificateList**. Elle sélectionne de manière non ambiguë une liste CRL unique.

```
certificateListExactMatch MATCHING-RULE ::=     {
    SYNTAX      CertificateListExactAssertion
    ID          id-mr-certificateListExactMatch }

CertificateListExactAssertion ::= SEQUENCE {
    issuer              Name,
    thisUpdate          Time,
    distributionPoint   DistributionPointName OPTIONAL }
```

La règle renvoie la valeur "Vrai" si les composants de la valeur de l'attribut stocké correspondent à ceux figurant dans la valeur présentée. Le composant **distributionPoint** doit correspondre, s'il est présent, à au moins l'une des formes de nom.

11.3.6 Concordance de liste de certificats

La règle de concordance de liste de certificats compare une valeur présentée avec une valeur d'attribut du type **CertificateList**. Elle sélectionne une ou plusieurs listes CRL en fonction de plusieurs caractéristiques.

```
certificateListMatch MATCHING-RULE ::= {
    SYNTAX      CertificateListAssertion
    ID          id-mr-certificateListMatch }

CertificateListAssertion ::= SEQUENCE {
    issuer                      Name                OPTIONAL,
    minCRLNumber        [0]     CRLNumber           OPTIONAL,
    maxCRLNumber        [1]     CRLNumber           OPTIONAL,
    reasonFlags                 ReasonFlags         OPTIONAL,
    dateAndTime                 Time                OPTIONAL,
    distributionPoint   [2]     DistributionPointName OPTIONAL,
    authorityKeyIdentifier [3]  AuthorityKeyIdentifier  OPTIONAL }
```

La règle de concordance renvoie la valeur "Vrai" si tous les composants figurant dans la valeur présentée concordent de la manière suivante avec les composants adéquats de la valeur de l'attribut stocké:

le composant **issuer** est en concordance si la valeur de ce composant dans la valeur de l'attribut est égale à celle dans la valeur présentée;

le composant **minCRLNumber** [*numéro minimal de liste CRL*] est en concordance si sa valeur est inférieure ou égale à celle figurant dans l'extension de numéro de liste CRL de la valeur de l'attribut stocké; il n'y a pas concordance si la valeur de l'attribut stocké ne contient pas d'extension de numéro de liste CRL;

le composant **maxCRLNumber** [*numéro maximal de liste CRL*] est en concordance si sa valeur est supérieure ou égale à celle figurant dans l'extension de numéro de liste CRL de la valeur de l'attribut stocké; il n'y a pas concordance si la valeur de l'attribut stocké ne contient pas d'extension de numéro de liste CRL;

le composant **reasonFlags** [*fanions de motif*] est en concordance si chacun des bits positionnés dans la valeur présentée est également positionné dans les composants **onlySomeReasons** de l'extension de point de répartition émetteur de la valeur de l'attribut stocké; il y a également concordance si la valeur de l'attribut stocké contient les fanions **reasonFlags** dans l'extension de point de répartition émetteur ou si la valeur de l'attribut stocké ne contient pas d'extension de point de répartition émetteur;

NOTE – Même si une liste CRL correspond à une valeur particulière de fanion **reasonFlags**, il se peut que la liste CRL ne contienne pas de notification de révocation avec ce code motif.

le composant **dateAndTime** [*date et heure*] est en concordance si la valeur est égale ou postérieure à celle figurant dans le composant **thisUpdate** de la valeur de l'attribut stocké et antérieure à celle figurant dans le composant **nextUpdate** de la valeur de l'attribut stocké; il n'y a pas concordance si la valeur de l'attribut stocké ne contient pas de composant **nextUpdate**;

le composant **distributionPoint** est en concordance si la valeur de l'attribut stocké contient une extension de point de répartition émetteur et si la valeur de ce composant dans la valeur présentée est égale à la valeur correspondante d'au moins l'une des formes de nom figurant dans cette extension;

le composant **authorityKeyIdentifier** est en concordance si la valeur de ce composant dans la valeur de l'attribut stocké est égale à celle figurant dans la valeur présentée; il n'y a pas concordance si la valeur de l'attribut stocké ne contient pas d'extension d'identificateur de clé d'autorité ou si la totalité des composants dans la valeur présentée n'est pas présente dans la valeur de l'attribut stocké.

11.3.7 Concordance d'identificateur d'algorithme

La règle de concordance d'identificateur d'algorithme compare une valeur présentée avec une valeur d'attribut du type **SupportedAlgorithms**.

```
algorithmIdentifierMatch MATCHING-RULE::=  {
    SYNTAX     AlgorithmIdentifier
    ID         id-mr-algorithmIdentifierMatch }
```

La règle renvoie la valeur "Vrai" si la valeur présentée est égale au composant **algorithmIdentifier** de valeur de l'attribut stocké.

11.3.8 Concordance de politique

La règle de concordance de politique compare une valeur présentée avec une valeur d'attribut du type **CertificatePolicy** ou avec une valeur d'attribut du type **privPolicy** [*politique privée*].

```
policyMatch MATCHING-RULE    ::= {
    SYNTAX     PolicyID
    ID         id-mr-policyMatch }
```

La règle renvoie la valeur "Vrai" si la valeur présentée est égale au composant **policyIdentifier** de la valeur de l'attribut stocké.

11.3.9 Concordance d'itinéraire PKI

La règle de concordance d'itinéraire PKI compare une valeur présentée avec une valeur d'attribut du type **pkiPath** [*itinéraire d'infrastructure PKI*]. Un système utilisant des certificats peut employer cette règle de concordance pour sélectionner un itinéraire partant d'un certificat émis par une autorité de certification à laquelle il fait confiance et aboutissant à un certificat émis à destination de l'autorité de certification qui a émis le certificat d'entité finale en cours de validation.

```
pkiPathMatch MATCHING-RULE  ::= {
    SYNTAX     PkiPathMatchSyntax
    ID         id-mr-pkiPathMatch }

kiPathMatchSyntax ::=  SEQUENCE {
    rstIssuer          Name,
    stSubject          Name }
```

Cette règle de concordance renvoie la valeur "Vrai" si la valeur présentée dans le composant **firstIssuer** [*premier émetteur*] correspond aux éléments adéquats du champ **issuer** du premier certificat de la **SEQUENCE** dans la valeur stockée et si la valeur présentée dans le composant **lastSubject** [*dernier sujet*] correspond aux éléments adéquats du champ sujet du dernier certificat de la **SEQUENCE** dans la valeur stockée. Cette règle de concordance renvoie la valeur "Faux" si l'une ou l'autre des comparaisons échoue.

SECTION 3 – CADRE DE CERTIFICAT D'ATTRIBUT

Le cadre de certificat d'attribut défini dans la présente Spécification fournit une base permettant d'édifier des infrastructures de gestion de privilèges (PMI, *privilege management infrastructures*). Ces infrastructures peuvent prendre en charge des applications telles que le contrôle d'accès.

La liaison d'un privilège avec une entité est fournie par une autorité au moyen d'un certificat de clé publique contenant une extension définie de manière explicite à cet effet. Le format des certificats d'attribut est défini dans la présente Spécification; il prévoit un procédé d'extension et un ensemble d'extensions de certificat spécifiques. La révocation des certificats d'attribut peut ou non être nécessaire. Les durées de validité peuvent être très brèves dans certains environnements (par exemple de l'ordre de quelques minutes), ce qui rend inutile un processus de révocation. Il est nécessaire que les utilisateurs puissent prendre connaissance d'une révocation pour éviter d'employer un certificat qui n'est plus digne de confiance. Un des procédés utilisables pour la notification de révocation aux utilisateurs met en œuvre des listes de révocation. Le format des listes défini dans la section 2 de la présente Spécification prévoit un procédé d'extension et un ensemble d'extensions de certificat spécifiques. D'autres extensions sont définies dans la présente Spécification. D'autres organismes peuvent également définir des extensions supplémentaires pour des certificats et des listes de révocation pouvant être utiles dans leurs environnements propres .

Un système utilisant des certificats d'attribut doit être en mesure de valider un certificat avant son utilisation par une application. La présente Spécification décrit également des procédures permettant d'effectuer cette validation, englobant la vérification de l'intégrité du certificat proprement dit, de son statut de révocation et de sa validité pour son utilisation prévue.

Ce cadre contient un certain nombre d'éléments optionnels qui sont pertinents uniquement dans certains environnements. Bien que les modèles soient définis de manière complète, ce cadre peut être utilisé dans des environnements qui n'en utilisent pas tous les composants. Il existe, par exemple, des environnements qui ne nécessitent pas de révocation de certificats d'attribut. La délégation de privilège et l'utilisation de rôle sont également des fonctionnalités de ce cadre qui ne s'appliquent pas dans tous les cas. Elles figurent toutefois dans la présente Spécification afin de permettre la prise en charge d'environnements qui en éprouvent le besoin.

L'annuaire utilise des certificats d'attribut pour fournir, pour les informations d'annuaire, un contrôle d'accès basé sur des règles.

12 Certificats d'attribut

Les certificats de clé publique sont conçus principalement en vue de la fourniture d'un service d'identification sur la base duquel il est possible d'édifier d'autres services, tels que l'intégrité des données, l'authentification d'entité, la confidentialité et l'autorisation. La présente Spécification fournit deux procédés permettant de lier un attribut de privilège à un détenteur.

Les certificats de clé publique peuvent fournir directement, conjointement au service d'authentification d'entité, un service d'autorisation si des privilèges sont associés au sujet par les pratiques de l'autorité de certification émettrice. Les certificats de clé publique peuvent contenir une extension **subjectDirectoryAttributes** [*attributs d'annuaire du sujet*] qui inclut des privilèges associés au sujet du certificat de clé publique. Ce procédé convient à des situations dans lesquelles l'autorité qui émet le certificat de clé publique (CA) est également l'autorité qui délègue le privilège (AA) et si la durée de validité du privilège correspond à celle du certificat de clé publique. Des entités finales ne peuvent pas agir comme autorités d'attribut. Si l'une quelconque des extensions définies à l'article 15 de la présente Spécification figure dans le certificat de clé publique, elle s'applique alors également à tous les privilèges attribués par l'extension **subjectDirectoryAttributes** de ce certificat.

Dans le cas le plus général, les durées de vie des privilèges d'entité ne correspondront pas à la durée de validité d'un certificat de clé publique. Les privilèges auront souvent une durée de vie beaucoup plus courte. L'autorité d'attribution de privilège sera souvent différente de celle qui émet un certificat de clé publique pour la même entité, et des privilèges divers peuvent être attribués par diverses autorités d'attribut (AA). Les privilèges peuvent également être attribués dans un contexte dépendant du temps et les caractéristiques de mise "en service/hors service" des privilèges peuvent également ne pas être en synchronisme avec la durée de vie du certificat de clé publique et/ou avec des privilèges d'entité émis par une autorité d'attribut différente. L'utilisation de certificats d'attribut émis par une autorité d'attribut fournit une infrastructure de gestion de privilège (PMI) flexible qui peut être mise en place et gérée indépendamment d'une infrastructure PKI. Il existe cependant une relation entre les deux dans la mesure où l'infrastructure PKI est utilisée pour authentifier les identités d'émetteurs et de détenteurs figurant dans des certificats d'attribut.

12.1 Structure du certificat d'attribut

La structure du certificat d'attribut diffère de celle du certificat de clé publique d'un sujet. Un sujet peut avoir plusieurs certificats associés à chacun de ses certificats de clé publique. Il n'est pas obligatoire que le certificat de clé publique et le ou les certificats d'attribut d'un utilisateur soient créés par une même autorité et la séparation des fonctions impose souvent qu'il n'en soit pas ainsi. Dans des environnements où des autorités différentes sont responsables de l'émission des certificats de clé publique et des certificats d'attribut, le ou les certificats de clé publique émis par une autorité de certification (CA) et le ou les certificats d'attribut émis par une autorité d'attribut (AA) sont signés au moyen de clés privées de signature différentes. Il est fortement recommandé que des clés différentes soient utilisées pour signer les certificats d'attribut et les certificats de clé publique dans des environnements où une entité unique est à la fois l'autorité de certification qui émet des certificats de clé publique et l'autorité d'attribut qui émet des certificats d'attribut. Les échanges entre l'autorité émettrice et l'entité réceptrice d'un certificat sont en dehors du domaine d'application de la présente Spécification.

Le certificat d'attribut est défini comme suit:

```
AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE
    {
    version                 AttCertVersion, -- la version est v2
    holder                  Holder,
    issuer                  AttCertIssuer,
    signature               AlgorithmIdentifier,
    serialNumber            CertificateSerialNumber,
    attrCertValidityPeriod  AttCertValidityPeriod,
    attributes              SEQUENCE OF Attribute,
    issuerUniqueID          UniqueIdentifier OPTIONAL,
    extensions              Extensions    OPTIONAL
    }

AttCertVersion ::= INTEGER { v2(1) }

Holder     ::=   SEQUENCE
        {
        baseCertificateID       [0] IssuerSerial        OPTIONAL,
            -- émetteur et numéro de série du certificat de clé publique du détenteur
        entityName              [1] GeneralNames        OPTIONAL,
            -- nom ou rôle de l'entité
        objectDigestInfo        [2] ObjectDigestInfo    OPTIONAL
            -- utilisé pour authentifier directement le détenteur, par exemple un exécutable
-- l'un au moins des composants baseCertificateID, entityName ou objectDigestInfo doit être présent --}

ObjectDigestInfo   ::= SEQUENCE {
    digestedObjectType  ENUMERATED {
        publicKey               (0),
        publicKeyCert           (1),
        otherObjectTypes        (2) },
    otherObjectTypeID           OBJECT IDENTIFIER  OPTIONAL,
    digestAlgorithm             AlgorithmIdentifier,
    objectDigest                BIT STRING }

AttCertIssuer ::=     [0]   SEQUENCE {
issuerName                  GeneralNames OPTIONAL,
baseCertificateID       [0] IssuerSerial OPTIONAL,
objectDigestInfo        [1] ObjectDigestInfo OPTIONAL }
-- au moins l'un des composants sera présent
    ( WITH COMPONENTS { ..., issuerName PRESENT } |
    WITH COMPONENTS { ..., baseCertificateID PRESENT } |
    WITH COMPONENTS { ..., objectDigestInfo PRESENT } )

IssuerSerial  ::=  SEQUENCE {
    issuer      GeneralNames,
    serial      CertificateSerialNumber,
    issuerUID   UniqueIdentifier OPTIONAL }

AttCertValidityPeriod  ::= SEQUENCE {
    notBeforeTime       GeneralizedTime,
    notAfterTime        GeneralizedTime }
```

La version permet de faire la distinction entre les diverses versions du certificat d'attribut. Pour les certificats d'attribut émis conformément à la syntaxe de la présente Spécification, le composant **version** doit être égal à **v2**.

Le composant **holder** [*détenteur*] véhicule l'identité du détenteur du certificat d'attribut.

Le composant **baseCertificateID** indique, s'il est présent, un certificat de clé publique particulier devant être utilisé pour authentifier l'identité de ce détenteur lorsque des privilèges sont déclarés au moyen de ce certificat d'attribut.

Le composant **entityName** [*nom d'entité*] indique, s'il est présent, un ou plusieurs noms pour le détenteur. Si **entityName** est le seul composant présent dans le champ **holder**, tout certificat de clé publique possédant comme sujet l'un de ces noms peut alors être utilisé pour authentifier l'identité du détenteur qui déclare des privilèges en utilisant ce certificat d'attribut. Si les composants **baseCertificateID** et **entityName** sont tous deux présents, seul le certificat indiqué par **baseCertificateID** peut alors être utilisé. Le composant **entityName** présent dans ce cas est uniquement un outil qui facilite la localisation, par le vérificateur de privilège, du certificat de clé publique concerné.

NOTE 1 – L'utilisation du seul composant **GeneralNames** pour l'identification du détenteur présente un risque, du fait qu'il pointe uniquement sur un nom du détenteur. Ceci ne suffit pas, en général, à authentifier l'identité d'un détenteur en vue de lui accorder des privilèges. L'utilisation du nom de l'émetteur et du numéro de série d'un certificat de clé publique donné permet toutefois à l'émetteur de certificats d'attribut de faire confiance au processus d'authentification effectué par l'autorité de certification lorsque ce certificat de clé publique particulier est émis. De même, l'utilisation de certaines des options du composant **GeneralNames** (par exemple, l'adresse **IPAddress**) n'est pas adéquate pour désigner un détenteur de certificat d'attribut, en particulier lorsque le détenteur n'est pas une entité individuelle mais un rôle. Un autre problème se pose lors de l'utilisation du seul composant **GeneralNames** comme identificateur d'un détenteur du fait qu'il n'existe pas d'autorité ou de processus strict pour l'attribution de noms pour un grand nombre de formes de nom figurant dans cette structure.

Le composant **objectDigestInfo** est utilisé directement, s'il est présent, pour authentifier l'identité d'un détenteur, y compris dans le cas d'un détenteur exécutable (par exemple, un "applet"). L'authentification du détenteur se fait en comparant le contenu du composant **objectDigest** [*résumé d'objet*] avec un résumé des informations correspondantes créées par le vérificateur de privilège en utilisant le même algorithme que celui qui est indiqué dans le composant **objectDigestInfo**. Le détenteur est authentifié à des fins de déclaration de privilèges au moyen de ce certificat d'attribut si les deux sont identiques.

- le composant **publicKey** [*clé publique*] sera indiqué lorsqu'un hachage de la clé publique d'une entité est présent. Le hachage d'une clé publique peut ne pas identifier sans ambiguïté un certificat unique (c'est-à-dire qu'une valeur de clé identique peut figurer dans plusieurs certificats). La liaison d'un certificat d'attribut avec une clé publique nécessite le calcul du hachage de la représentation de cette clé publique qui figure dans un certificat de clé publique. D'une manière spécifique, les informations d'entrée de l'algorithme de hachage se constitueront du codage DER d'une représentation de la clé figurant dans le composant **SubjectPublicKeyInfo** [*informations de clé public du sujet*]. Il convient de noter que ceci inclut les éléments **AlgorithmIdentifier** et **BIT STRING**. Il convient de noter également que si la valeur de la clé publique utilisée en entrée de la fonction de hachage a été extraite d'un certificat de clé publique, il est alors possible (par exemple, si des paramètres de l'algorithme de signature numérique ont été obtenus par héritage) que ceci soit insuffisant pour le hachage. Les informations d'entrée correctes du hachage comprendront, dans un tel contexte, la valeur des paramètres hérités et peuvent différer de ce fait des informations du composant **SubjectPublicKeyInfo** figurant dans le certificat de clé publique;

- le composant **publicKeyCert** [*certificat de clé publique*] sera indiqué dans le cas d'un certificat de clé publique haché, le hachage portant sur la totalité du codage DER du certificat de clé publique, y compris les bits de signature;

- le composant **otherObjectTypes** [*autres types d'objet*] sera indiqué lorsque des objets autres que des clés publiques ou certificats de clé publique sont hachés (par exemple, des objets logiciel). L'identité du type de l'objet peut être fournie de manière optionnelle. La partie de l'objet devant être hachée peut être déterminée, soit par le type indiqué de manière explicite, soit par le contexte d'utilisation de l'objet si l'identificateur n'est pas fourni.

Le composant **issuer** véhicule l'identité de l'autorité d'attribut qui a émis le certificat.

- Le composant **issuerName** indique, s'il est présent, un ou plusieurs noms pour l'émetteur.

- Le composant **baseCertificateID** indique, s'il est présent, l'émetteur par une référence à un certificat de clé publique particulière dont cet émetteur est le sujet.

- Le composant **objectDigestInfo** indique, s'il est présent, l'émetteur en fournissant un hachage de ses informations d'identification.

Le composant **signature** indique l'algorithme de chiffrement utilisé pour la signature numérique du certificat d'attribut.

Le composant **serialNumber** contient le numéro de série qui identifie sans ambiguïté le certificat d'attribut dans le domaine d'application de son émetteur.

Le composant **attrCertValidityPeriod** [*durée de validité du certificat*] véhicule la durée pendant laquelle le certificat d'attribut est considéré comme valide, exprimée dans le format **GeneralizedTime**.

Le composant **attributes** contient les attributs associés au détenteur, qui sont en cours de certification (par exemple, les privilèges).

NOTE 2 – Cette séquence d'attributs peut être vide dans le cas d'un certificat d'attribut de descripteur d'attribut.

Le composant **issuerUniqueID** [*identificateur unique de l'émetteur*] peut être utilisé pour identifier l'émetteur du certificat d'attribut lorsque le composant **issuer** n'est pas suffisant.

Le composant **extensions** permet d'ajouter d'autres champs au certificat d'attribut.

Le cadre pour les certificats d'attribut décrit dans cette section se concentre essentiellement sur le modèle pour lequel un privilège figure au sein de certificats d'attribut. Toutefois, comme mentionné précédemment, l'extension de certificats définie dans ce paragraphe peut également figurer dans un certificat de clé publique qui utilise l'extension **subjectDirectoryAttributes**.

12.2 Itinéraires de certificat d'attribut

Il peut être nécessaire, comme pour les certificats de clé publique, de véhiculer un certificat d'attribut (par exemple, pour déclarer des privilèges dans le cadre d'un protocole d'application). Le type de données ASN.1 suivant peut être utilisé pour représenter un itinéraire de certification d'attribut.

```
AttributeCertificationPath ::= SEQUENCE {
    attributeCertificate        AttributeCertificate,
    acPath                      SEQUENCE OF ACPathData OPTIONAL }

ACPathData ::= SEQUENCE {
    certificate             [0] Certificate OPTIONAL,
    attributeCertificate    [1] AttributeCertificate OPTIONAL }
```

13 Relations entre l'autorité d'attribut, la source d'autorité et l'autorité de certification

L'autorité d'attribut (AA) et l'autorité de certification (CA) sont logiquement (et souvent aussi physiquement) totalement indépendantes. La création et la maintenance de "l'identité" peut (et doit souvent) être distincte de l'infrastructure PMI. La totalité de l'infrastructure PKI peut de ce fait être mise en place et exploitée avant la mise en place de l'infrastructure PMI. L'autorité de certification, bien qu'elle soit source d'autorité pour les identités au sein de son domaine, n'est pas automatiquement la source d'autorité pour les privilèges. L'autorité de certification ne sera de ce fait pas nécessairement elle-même une autorité d'attribut et, comme conséquence logique, ne sera pas nécessairement responsable de la décision permettant à d'autres entités d'agir comme autorité d'attribut (par exemple en les désignant en tant que telles dans leurs certificats d'identité).

La source d'autorité (SOA) est l'entité à laquelle un vérificateur de privilège fait confiance comme responsable ultime pour l'attribution d'un ensemble de privilèges. Une ressource peut imposer des limitations à la source d'autorité en faisant confiance à certaines sources d'autorité pour des fonctions données (par exemple une source pour les privilèges de lecture et une autre pour les privilèges d'écriture). Une source d'autorité est elle-même une autorité d'attribut dans la mesure où elle émet, pour d'autres entités, des certificats leur attribuant des privilèges. Une source d'autorité peut être comparée à une "autorité de certificat racine" ou à une "ancre de confiance" dans l'infrastructure PKI, du fait qu'un vérificateur de privilège fait confiance aux certificats portant sa signature. Il est nécessaire, dans certains environnements, que les autorités de certification surveillent de manière stricte les entités pouvant agir comme source d'autorité. Ce cadre fournit un procédé de la prise en charge de ce besoin. Cette surveillance n'est pas nécessaire dans d'autres environnements et les procédés permettant de déterminer les entités pouvant agir comme source d'autorité dans de tels environnements peuvent, le cas échéant, être en dehors du domaine d'application de la présente Spécification.

Ce cadre est flexible et peut répondre aux besoins d'un grand nombre de types d'environnement.

a) Dans beaucoup d'environnements, tous les privilèges seront attribués directement aux entités individuelles par une autorité d'attribut, à savoir la source d'autorité.

b) D'autres environnements peuvent nécessiter la prise en charge de la fonctionnalité optionnelle de rôle qui permet d'émettre des certificats attribuant divers rôles à des individus. Les privilèges associés au rôle sont attribués de manière implicite à ces individus. Les privilèges de rôle peuvent être attribués par un certificat émis pour le rôle lui-même ou par d'autres moyens (par exemple, par configuration locale).

c) Une autre fonctionnalité optionnelle de ce cadre est la prise en charge de la délégation de privilège. Dans le cas d'une délégation, la source d'autorité attribue un privilège à une entité qui est également autorisée à agir comme autorité d'attribut et déléguer le privilège à son tour. La délégation peut se répéter pour plusieurs autorités d'attribut intermédiaires jusqu'à ce qu'elle soit attribuée à une entité finale, qui ne peut plus déléguer ce privilège. Les autorités d'attribut intermédiaires peuvent ou non être en mesure d'agir comme déclarants de privilège pour les privilèges qu'elles délèguent.

d) Dans certains environnements, la même entité physique peut agir à la fois comme autorité d'attribut et autorité de certification. Ce double rôle logique pour une même entité physique existe toujours lorsque les privilèges sont véhiculés par une extension **subjectDirectoryAttributes** d'un certificat de clé publique. Dans d'autres environnements, des entités physiques distinctes agissent comme autorité de certification et autorité d'attribut. Dans ce cas, un privilège est attribué en utilisant un certificat d'attribut et non un certificat de clé publique.

L'infrastructure PKI est utilisée pour authentifier les détenteurs (déclarants de privilège) et vérifier les signatures numériques des émetteurs, lorsque les certificats d'attribut possèdent des pointeurs vers les certificats de clé publique de leurs émetteurs et de leurs détenteurs.

13.1 Privilège dans les certificats d'attribut

Les entités peuvent acquérir un privilège de l'une des deux manières suivantes:

- une autorité d'attribut peut attribuer un privilège de manière unilatérale à une entité en créant un certificat d'attribut (éventuellement à sa propre initiative ou à la demande d'un tiers). Ce certificat peut être stocké dans un répertoire d'accès public et peut être traité par la suite par un ou plusieurs vérificateurs de privilège à des fins de décision d'autorisation. Ceci peut s'effectuer sans que l'entité en ait connaissance ou sans action explicite de sa part.
- une entité peut, en outre, demander un privilège à une certaine autorité d'attribut. Une fois qu'il est créé, ce certificat sera renvoyé (exclusivement) à l'entité qui en fait la demande et qui le présente de manière explicite lorsqu'elle demande l'accès à une ressource protégée.

Il convient de noter que dans les deux procédures, l'autorité d'attribut doit effectuer les actions nécessaires pour garantir que l'entité se voit effectivement attribuer le privilège. Ceci peut impliquer certains procédés hors bande, comparables à la certification d'une liaison entre une identité et une clé par une autorité de certification.

L'infrastructure PMI basée sur des certificats d'attribut convient à des environnements lorsque l'une quelconque des conditions suivantes est remplie:

- une entité est responsable de l'attribution d'un privilège particulier à un détenteur et une autre, de l'émission des certificats de clé publique pour le même sujet;
- un certain nombre d'attributs de privilèges doivent être assignés à un détenteur par un certain nombre d'autorités;
- la durée de vie d'un privilège est différente de la durée de validité du certificat de clé publique du détenteur (la durée de vie du privilège est en général notablement plus réduite);
- le privilège est valable uniquement durant certains laps de temps qui ne sont pas en synchronisme avec la validité de la clé publique de l'utilisateur ou la validité d'autres privilèges.

13.2 Privilège dans des certificats de clé publique

Dans certains environnements, les privilèges sont associés au sujet par les pratiques d'une autorité de certification. De tels privilèges peuvent figurer directement dans des certificats de clé publique (et réutiliser de ce fait une grande partie d'une infrastructure existante), plutôt que d'être émis dans des certificats d'attribut. Dans un tel cas, le privilège est contenu dans l'extension **subjectDirectoryAttributes** du certificat de clé publique.

Ce procédé convient lorsqu'une ou plusieurs des conditions suivantes sont remplies:

- une même entité physique intervient à la fois comme autorité de certification et comme autorité d'attribut;
- la durée de vie du privilège est identique à celle de la clé publique contenue dans le certificat;
- la délégation de privilège n'est pas autorisée;
- la délégation est autorisée, mais pour toute délégation la totalité des privilèges figurant dans le certificat (dans l'extension **subjectDirectoryAttributes**) possèdent les mêmes paramètres de délégation et toutes les extensions pertinentes pour la délégation s'appliquent de la même manière à tous les privilèges figurant dans le certificat.

14 Modèles d'infrastructure PMI

14.1 Modèle général

Le modèle général de gestion de privilège se constitue de trois entités: l'objet, le déclarant de privilège et le vérificateur de privilège.

L'objet peut être une ressource protégée, par exemple dans une application de contrôle d'accès. La ressource protégée constitue un objet. Le type de cet objet dispose de méthodes pouvant être invoquées (il peut s'agir, par exemple, d'un pare-feu qui dispose de la méthode objet "autoriser l'entrée" ou d'un système de fichier qui dispose des méthodes objet lecture, écriture et exécution). Un autre type d'objet dans ce modèle peut être un objet qui a été signé dans une application de non-répudiation.

Le déclarant de privilège est l'entité qui détient un privilège particulier et déclare ses privilèges pour un contexte d'utilisation particulier.

Le vérificateur de privilège est l'entité qui détermine si les privilèges déclarés sont suffisants ou non pour le contexte d'utilisation donné.

La détermination de réussite ou d'échec est faite par le vérificateur de privilège en fonction des quatre facteurs suivants:

- privilège du déclarant de privilège;
- politique de privilège en vigueur;
- variables d'environnement actuelles, si elles sont pertinentes;
- sensibilité de la méthode objet, si elle est pertinente.

Le privilège d'un détenteur de privilège indique le niveau de confiance accordé par l'émetteur du certificat au détenteur, en ce qui concerne le respect par ce dernier des caractéristiques de la politique qui ne sont pas imposées par des moyens techniques. Ce privilège est encapsulé dans l'attribut du ou des certificats du détenteur de privilège (ou dans l'extension **subjectDirectoryAttributes** de son certificat de clé publique) pouvant être présentés au vérificateur de privilège dans la demande d'invocation ou distribués par un autre moyen, par exemple l'annuaire. La codification du privilège utilise la structure **Attribute** qui contient un type **AttributeType** [*type d'attribut*] et un ensemble **SET OF AttributeValue**. Certains types d'attribut utilisés pour spécifier un privilège peuvent avoir une structure très simple, telle qu'un élément unique du type **INTEGER** ou **OCTET STRING**. D'autres attributs peuvent avoir des syntaxes plus complexes. L'Annexe D en fournit un exemple.

La politique de privilège spécifie le niveau de privilège considéré comme suffisant compte tenu de la sensibilité d'une méthode objet donnée ou pour un contexte d'utilisation donné. La politique de privilège doit être protégée en ce qui concerne l'intégrité des données et l'authenticité. La politique peut être véhiculée de diverses manières. L'une des solutions extrêmes consiste à considérer que la politique n'est pas acheminée, mais simplement définie et conservée d'une manière purement locale dans l'environnement du vérificateur de privilège. L'autre extrême consiste à considérer que certaines politiques sont "universelles" et doivent être acheminées vers, ou connues de toute entité du système. Il existe un grand nombre de variantes entres ces solutions extrêmes. La présente Spécification définit des composants d'un schéma de stockage d'informations de politique de privilège dans l'annuaire.

La politique de privilège Spécifie le seuil d'acceptation pour un ensemble de privilèges donné. Ceci signifie qu'elle définit avec précision dans quelles conditions un vérificateur de privilège doit conclure qu'un ensemble de privilèges présenté est "suffisant" pour qu'il décide d'accorder l'accès au déclarant de privilège (pour sa demande d'objet, de ressource, d'application, etc.).

La présente Spécification ne normalise pas de syntaxe pour la définition d'une politique de privilège. L'Annexe D présente quelques exemples de syntaxes pouvant être utilisées à cet effet. Il s'agit toutefois uniquement d'exemples. Toute syntaxe peut être utilisée, y compris un texte normal. Toute instance de politique de privilège doit être identifiée sans ambiguïté, quelle que soit la syntaxe utilisée. Des identificateurs d'objet sont utilisés à cet effet.

PrivilegePolicy ::= OBJECT IDENTIFIER

Les variables d'environnement, si elles sont utilisées, représentent celles des caractéristiques des prescriptions de politique nécessaires à la détermination de réussite ou d'échec (par exemple, l'heure du jour ou le solde actuel d'un compte) auxquelles le vérificateur de privilège peut accéder par des moyens locaux. La représentation des variables d'environnement est un problème purement local.

La sensibilité de la méthode objet, si elle est utilisée, peut représenter des attributs du document ou de la demande à traiter, tels que le montant d'un transfert de fonds qu'il propose d'effectuer ou la confidentialité du contenu du document. La sensibilité de la méthode objet peut être codée de manière explicite dans une étiquette de sécurité associée ou dans un certificat d'attribut détenu par la méthode objet; elle peut aussi être encapsulée de manière implicite dans la structure et le contenu de l'objet de données associé. Le codage peut se faire d'un certain nombre de manières, par exemple en dehors du domaine d'application de l'infrastructure PMI, dans l'étiquette X.411 associée au document ou dans les champs d'échange de données EDIFACT; il peut également être codé de manière statique dans l'application du vérificateur de privilège. Il est possible, en variante, d'utiliser l'infrastructure PMI avec un certificat d'attribut associé à la méthode objet. Certains contextes d'utilisation ne font pas appel à la sensibilité de la méthode objet.

Un vérificateur de privilège n'est pas nécessairement lié par une relation avec une autorité d'attribut particulière. Comme les détenteurs de privilège peuvent posséder des certificats d'attribut émis à leur intention par un grand nombre d'autorités d'attribut, les vérificateurs de privilège peuvent accepter des certificats émis par de nombreuses autorités d'attribut, qui ne sont pas nécessairement liées hiérarchiquement, pour accorder l'accès à une ressource donnée.

Le cadre de certificat d'attribut est utilisable pour la gestion de privilèges de divers types et à des fins diverses. Les termes utilisés dans la présente Spécification, tels que déclarant de privilège, vérificateur de privilège, etc. ne dépendent pas de l'application ou de l'utilisation particulière.

14.1.1 Infrastructure PMI dans le contexte de contrôle d'accès

Il existe un cadre normalisé pour le contrôle d'accès (Rec. UIT-T X.812 | ISO/CEI 10181-3) qui définit un ensemble correspondant de termes propres à l'application de la commande d'accès. Un mappage des termes génériques utilisés dans la présente Spécification avec ceux du cadre de contrôle d'accès est donnée ci-dessous afin de clarifier la relation entre ce modèle et la présente Spécification.

Le déclarant de privilège de la présente Spécification joue un rôle "d'initiateur" dans le cadre du contrôle d'accès.

Le vérificateur de privilège de la présente Spécification joue le rôle d'une "fonction de décision de commande d'accès (ADF)" dans le cadre de contrôle d'accès.

La méthode objet pour laquelle un privilège est déclaré dans la présente Spécification correspond à la "cible" définie dans le cadre de contrôle d'accès.

Les variables d'environnement de la présente Spécification correspondent aux "informations de contexte" dans le cadre du contrôle d'accès.

La politique de privilège analysée dans la présente Spécification peut comprendre la "politique de contrôle d'accès" et les "règles de politique de contrôle d'accès" définies dans le cadre du contrôle d'accès.

Ce modèle permet de superposer de manière relativement homogène un réseau existant de ressources à protéger avec une infrastructure PMI. En particulier, le fait que le vérificateur de privilège intervient comme passerelle vers une méthode d'objet sensible en acceptant ou en refusant des demandes pour l'invocation de cette méthode objet permet de protéger l'objet sans impact sur l'objet lui-même ou avec un impact réduit. Le vérificateur de privilège filtre toutes les demandes concernant les méthodes objet adéquates et ne laisse passer que celles qui sont dûment autorisées.

14.1.2 Infrastructure PMI dans un contexte de non-répudiation

Il existe un cadre normalisé pour la non-répudiation (Rec. UIT-T X.813 | ISO/CEI 10181-4) qui définit un ensemble correspondant de termes propres à la non-répudiation. Un mappage des termes génériques utilisés dans la présente Spécification avec ceux du cadre de non-répudiation est donné ci-dessous afin de clarifier la relation entre ce modèle et la présente Spécification.

Le déclarant de privilège de la présente Spécification joue un rôle de "sujet de preuve" ou "d'origine" dans le cadre de non-répudiation.

Le vérificateur de privilège de la présente Spécification joue un rôle "d'utilisateur de preuve" ou de "bénéficiaire" dans le cadre de non-répudiation.

La méthode objet pour laquelle un privilège est déclaré dans la présente Spécification correspond à la "cible" définie dans le cadre de non-répudiation.

Les variables d'environnement de la présente Spécification correspondent à "la date et l'heure auxquelles la preuve a été générée ou vérifiée" dans le cadre de non-répudiation.

La politique de privilège analysée dans la présente Spécification peut comprendre la "politique de sécurité de non-répudiation" dans le cadre de non-répudiation.

14.2 Modèle de contrôle d'accès

Le modèle de contrôle couvre le contrôle de l'accès à la méthode de l'objet sensible. Le modèle se constitue de cinq composants: le déclarant de privilège, le vérificateur de privilège, la méthode objet, la politique de privilège et les variables d'environnement (voir Figure 3). Le déclarant de privilège possède un privilège; la méthode objet possède une sensibilité. Les procédés décrits ici permettent au vérificateur de privilège de contrôler l'accès du déclarant de privilège à la méthode objet, conformément à la politique de privilège. Le privilège et la sensibilité peuvent être des paramètres à valeurs multiples.

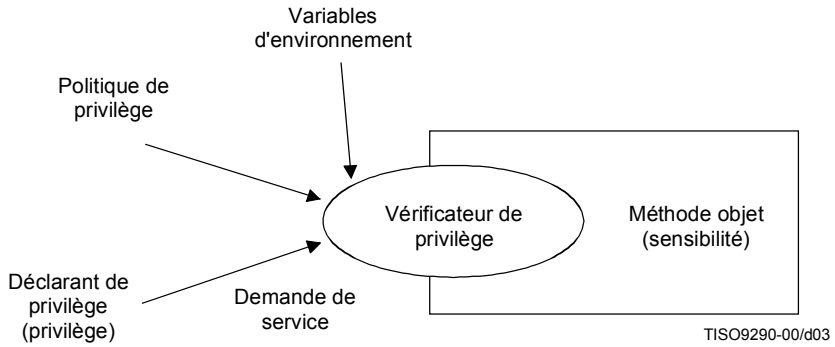


Figure 3 – Modèle de contrôle

Le déclarant de privilège peut être une entité identifiée par un certificat de clé publique ou par un objet exécutable identifié par le résumé de son image sur disque, etc.

14.3 Modèle de délégation

Il peut être nécessaire de déléguer un privilège dans certains environnements; il s'agit toutefois d'une caractéristique optionnelle du cadre qui n'est pas requise dans tous les environnements. Le modèle de délégation se constitue des quatre composants suivants: le vérificateur de privilège, la source d'autorité, les autres autorités d'attribut et le déclarant de privilège (voir Figure 4).

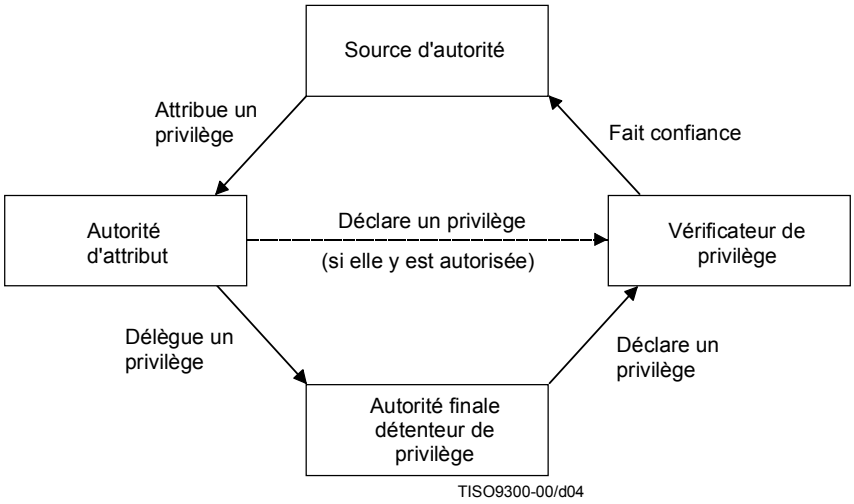


Figure 4 – Modèle de délégation

Comme dans les environnements n'utilisant pas la délégation, la source d'autorité est l'émetteur initial de certificats qui attribuent un privilège à un détenteur de privilège. Elle autorise toutefois le détenteur de privilège à agir comme autorité d'attribut et à déléguer à son tour le privilège à d'autres entités par l'émission de certificats contenant le même privilège (ou un sous-ensemble de ce dernier). La source d'autorité peut imposer des contraintes à cette délégation (par exemple, limiter la longueur de l'itinéraire ou l'espace de noms autorisé pour la délégation). Chacune de ces autorités d'attribut intermédiaires peut, dans les certificats qu'elle émet pour de nouveaux détenteurs de privilège, autoriser ces deniers à agir comme des autorités d'attribut et effectuer à leur tour une délégation. Une limitation universelle de la délégation est qu'aucune autorité d'attribut ne peut déléguer un privilège qu'elle ne détient pas. Une entité effectuant la délégation peut imposer d'autres restrictions aux capacités des autorités d'attribut situées en aval.

Lorsque la délégation est utilisée, le vérificateur de privilège fait confiance à la source d'autorité en ce qui concerne la délégation de tout ou partie de ces privilèges à des détenteurs, dont certains peuvent déléguer à leur tour tout ou partie de ces privilèges à d'autres détenteurs.

Le vérificateur de privilège fait confiance à la source d'autorité comme autorité pour un ensemble donné de privilèges pour la ressource. Si le certificat du déclarant n'est pas émis par cette source d'autorité, le vérificateur de privilège doit alors établir l'existence d'un itinéraire de délégation de certificats partant de ce déclarant de privilège et aboutissant à un certificat émis par la source d'autorité. La validation de cet itinéraire de délégation doit inclure la vérification du fait que chaque autorité d'attribut dispose de privilèges suffisants et d'une autorisation valable pour leur délégation.

Lorsque les privilèges sont véhiculés par des certificats d'attribut, l'itinéraire de délégation est différent de l'itinéraire de validation de certificat utilisé pour valider les certificats de clé publique des entités impliquées dans le processus de délégation. La qualité de l'authentification fournie par le processus de validation du certificat de clé publique doit toutefois être en rapport avec la sensibilité de l'objet protégé.

Un itinéraire de délégation se constituera en totalité, soit de certificats d'attribut, soit de certificats de clé publique. Une entité qui a obtenu un privilège dans un certificat d'attribut peut uniquement utiliser des certificats d'attribut pour effectuer à son tour une délégation, si elle y est autorisée. De même, un privilège obtenu dans un certificat de clé publique peut uniquement être délégué par l'émission de nouveaux certificats de clé publique, si cette délégation est autorisée. Seules les autorités d'attribut peuvent déléguer un privilège et non les entités finales.

14.4 Modèle de rôles

Les rôles permettent d'attribuer de manière indirecte des privilèges à des individus. Ces derniers reçoivent des certificats d'attribution de rôle qui leur assignent un ou plusieurs rôles. Des privilèges particuliers sont attribués à un nom de rôle au moyen de certificats de spécification de rôle et non à des détenteurs de privilège individuels au moyen de certificats d'attribut. Ce niveau indirect permet, par exemple, de mettre à jour les privilèges attribués à un rôle sans impact sur les certificats qui attribuent des rôles à des individus. Les certificats d'attribution de rôle peuvent être des certificats d'attribut ou des certificats de clé publique. Les certificats de spécification de rôle peuvent être des certificats d'attribut, mais non des certificats de clé publique. L'attribution de privilèges à un rôle peut se faire par d'autres moyens si les certificats de spécification de rôle ne sont pas utilisés (par exemple, par configuration locale d'un vérificateur de privilège).

Les actions suivantes sont possibles:

- toute autorité d'attribut peut définir un nombre quelconque de rôles;
- le rôle lui-même et les membres d'un rôle peuvent être définis et administrés séparément par des autorités d'attribut différentes;
- il est possible de déléguer l'appartenance à un rôle comme tout autre privilège;
- il est possible d'attribuer toute durée de vie adéquate pour les rôles et les appartenances.

L'attribut **role** [*rôle*] figure dans le composant **attributes** du certificat d'attribut si le certificat d'attribution de rôle est un certificat d'attribut. L'attribut **role** figure dans l'extension **subjectDirectoryAttributes** si le certificat d'attribution de rôle est un certificat de clé publique. Dans ce cas, tous les autres privilèges figurant dans le certificat de clé publique sont attribués directement au sujet du certificat et non au rôle.

Un déclarant de privilège peut figurer de ce fait dans un certificat d'attribution de rôle destiné au vérificateur de privilège uniquement pour indiquer que le déclarant de privilège joue un rôle particulier (par exemple, "directeur" ou "acheteur"). Lorsqu'il doit prendre une décision de réussite ou d'échec d'autorisation, le vérificateur de privilège peut avoir une connaissance *a priori* des privilèges associés au rôle déclaré ou peut être obligé de les trouver par d'autres moyens. Le certificat de spécification de rôle peut être utilisé à cet effet.

Un vérificateur de privilège doit être en mesure de comprendre les privilèges spécifiés pour le rôle. L'attribution de ces privilèges au rôle peut se faire dans l'infrastructure PMI en utilisant un certificat de spécification de rôle ou en dehors de cette infrastructure (par exemple, par configuration locale). La présente Spécification fournit les procédés permettant de lier ce certificat avec le certificat d'attribution de rôle pertinent si les privilèges du rôle sont déclarés dans le certificat de

spécification de rôle. Un certificat de spécification de rôle ne peut pas être délégué à une autre entité. L'émetteur du certificat d'attribution de rôle peut être différent de l'émetteur du certificat de spécification de rôle et ces émetteurs peuvent être administrés (pour leur expiration, révocation, etc.) d'une manière totalement indépendante. Un même certificat (d'attribut ou de clé publique) peut être un certificat d'attribution de rôle et contenir simultanément d'autres privilèges concernant uniquement le même individu. Un certificat de spécification de rôle doit toutefois constituer un certificat distinct.

NOTE – L'utilisation de rôles dans un cadre d'autorisation peut accroître la complexité du traitement d'itinéraire, parce qu'une telle fonctionnalité définit un autre itinéraire de délégation qu'il est nécessaire de suivre. L'itinéraire de délégation du certificat d'attribution de rôle peut impliquer des autorités d'attribut diverses et être indépendant de l'autorité d'attribut qui a émis le certificat de spécification de rôle.

14.4.1 Attribut "rôle"

La spécification des types d'attribut de privilège est en général un problème propre à l'application qui est en dehors du domaine d'application de la présente Spécification. La seule exception qui concerne l'attribut est définie ci-dessous pour l'attribution d'un détenteur à un rôle. La spécification des valeurs de l'attribut "rôle" est en dehors du domaine d'application de la présente Spécification.

```
role   ATTRIBUTE  ::= {
       WITH SYNTAX          RoleSyntax
       ID                   id-at-role }

RoleSyntax ::= SEQUENCE {
roleAuthority       [0]     GeneralNames    OPTIONAL,
roleName            [1]     GeneralName }
```

Cet attribut de privilège est utilisé pour remplir le champ **attributes** d'un certificat d'attribution de rôle. Si le certificat d'attribution de rôle est un certificat de clé publique, il est alors utilisé pour remplir l'extension **subjectDirectoryAttributes** de ce certificat.

Le composant **roleAuthority** [*autorité de rôle*] indique, s'il est présent l'autorité reconnue qui est responsable de l'émission du certificat de spécification de rôle.

Si le composant **roleAuthority** est présent et si un vérificateur de privilège utilise un certificat de spécification de rôle pour déterminer les privilèges attribués au rôle, au moins l'un des noms du composant **roleAuthority** doit alors figurer dans le champ **issuer** de ce certificat. Les procédés permettant de s'assurer que les privilèges ont été attribués par une autorité dont le nom figure dans ce composant sont en dehors du domaine d'application de la présente Spécification si le vérificateur de privilège utilise des moyens autres qu'un certificat de spécification de rôle pour déterminer les privilèges attribués au rôle.

L'identité de l'autorité responsable doit être déterminée par d'autres moyens si le composant **roleAuthority** est absent. L'extension **roleSpecCertIdentifier** [*identificateur de certificat de spécification de rôle*] dans un certificat d'attribution de rôle est l'une des manières d'établir cette liaison lorsque les privilèges ont été attribués au rôle en utilisant un certificat de spécification de rôle.

Le composant **roleName** [*nom de rôle*] indique le rôle assigné au détenteur d'un certificat d'attribution de rôle contenant cet attribut. Ce nom de rôle doit également figurer dans le champ **holder** du certificat de spécification de rôle si un vérificateur de privilège utilise un certificat de spécification de rôle pour déterminer les privilèges attribués à ce rôle.

15 Extensions de certificat de gestion de privilège

Les extensions de certificat suivantes peuvent figurer dans des certificats à des fins de gestion de privilège. Des règles sont également fournies pour les types de certificat dans lesquels peut figurer l'extension.

Sauf pour l'extension d'identificateur de source d'autorité, toute extension pouvant figurer dans un certificat de clé publique sera uniquement présente dans ce certificat si ce dernier attribue un privilège à son sujet (ce qui implique que l'extension **subjectDirectoryAttributes** sera présente). Cette extension s'applique à la totalité des privilèges figurant dans l'extension **subjectDirectoryAttributes** si l'une quelconque de ces extensions est présente dans le certificat de clé publique.

Les listes de révocation utilisées pour publier des notifications de révocation concernant des certificats d'attribut (listes ACRL et AARL) peuvent contenir toute liste CRL ou toute extension d'élément de liste CRL telles qu'elles sont définies dans la section 2 de la présente Spécification pour l'utilisation dans des listes CRL et CARL.

Cet article spécifie les extensions concernant les domaines suivants:

a) *gestion de privilège de base*: ces extensions de certificat véhiculent des informations concernant la déclaration d'un privilège;

b) *révocation de privilège*: ces extensions de extensions de certificat véhiculent des informations concernant l'emplacement où se trouvent les informations de statut de révocation;

c) *source d'autorité*: ces extensions de certificats concernent la source d'attribution de privilège à laquelle un vérificateur fait confiance pour une ressource donnée;

d) *rôles*: ces extensions de certificat véhiculent des informations concernant l'emplacement où se trouvent les certificats de spécification de rôle connexes.

e) *délégation*: ces extensions de certificat permettent de définir des contraintes pour une délégation ultérieure des privilèges attribués.

15.1 Extensions de gestion de privilège de base

15.1.1 Définition des besoins

Les besoins suivants sont liés à la gestion de privilège de base:

a) les émetteurs doivent pouvoir définir des contraintes pour la durée pendant laquelle un privilège peut être déclaré;

b) les émetteurs doivent pouvoir définir des certificats d'attribut comme cible pour des serveurs ou des services spécifiques;

c) les émetteurs peuvent avoir besoin de faire transporter des informations à des fins d'affichage à l'intention de déclarants de privilège et/ou de vérificateurs de privilège qui utilisent le certificat;

d) les émetteurs peuvent avoir besoin d'imposer des contraintes pour les politiques de privilège avec lesquelles les privilèges attribués peuvent être utilisés.

15.1.2 Champs de gestion d'extension de privilège de base

Les champs d'extension suivants sont définis:

a) *spécification de durée;*

b) *informations de cible;*

c) *notification d'utilisateur;*

d) *politiques de privilège acceptable*.

15.1.2.1 Extension de spécification de durée

L'extension de spécification de durée peut être utilisée par une autorité d'attribut pour limiter la durée pendant laquelle le privilège attribué dans le certificat qui contient cette extension peut être déclaré par le détenteur de privilège. Une autorité d'attribut peut, par exemple, émettre un certificat qui attribue des privilèges pouvant être déclarés uniquement du lundi au vendredi dans la tranche horaire de 9:00 heures à 17:00 heures. Un autre exemple peut être le cas d'un directeur qui délègue son autorité de signature à un subordonné pendant la durée de ses vacances.

Ce champ est défini comme suit:

```
timeSpecification EXTENSION  ::=  {
    SYNTAX              TimeSpecification
    IDENTIFIED BY       id-ce-timeSpecification }
```

Cette extension peut figurer dans des certificats d'attribut ou des certificats de clé publique émis par des autorités d'attribut, y compris des sources d'autorité, pour des entités qui peuvent agir comme déclarants de privilège, incluant d'autres autorités d'attribut et des entités finales. Cette extension ne figurera pas dans des certificats qui contiennent l'extension d'identificateur de source d'autorité ou dans des certificats émis pour des autorités d'attribut qui ne peuvent pas agir également comme déclarants de privilège.

Si cette extension figure dans un certificat émis pour une entité qui est une autorité d'attribut, elle s'applique alors uniquement à la déclaration faite par cette entité pour les privilèges figurant dans le certificat. Elle n'a aucun impact sur la durée pendant laquelle l'autorité d'attribut est en mesure d'émettre des certificats.

Comme cette extension spécifie en fait une réduction de la durée de validité du certificat qui la contient, elle doit être marquée comme critique (ce qui signifie que l'émetteur de cette extension indique de manière explicite que l'attribution de privilège n'est pas valide en dehors de la durée spécifiée).

Le certificat doit être rejeté si cette extension est présente et n'est pas comprise par le vérificateur de privilège.

15.1.2.1.1 Spécification de concordance de durée

La règle de concordance de spécification de durée compare une valeur présentée avec une valeur d'attribut du type **AttributeCertificate** [*certificat d'attribut*].

```
timeSpecificationMatch MATCHING-RULE ::= {
    SYNTAX          TimeSpecification
    ID              id-mr-timeSpecMatch }
```

Cette règle de concordance renvoie la valeur "Vrai" si la valeur stockée contient l'extension **timeSpecification** [*spécification de temps*] et si les composants figurant dans la valeur présentée et les composants correspondants de la valeur stockée concordent.

15.1.2.2 Extension d'informations de cible

L'extension d'informations de cible permet de désigner, pour un certificat d'attribut, une cible constituée d'un ensemble spécifique de serveurs ou de services. Un certificat d'attribut qui contient cette extension ne doit être utilisé que pour les serveurs ou les services spécifiés.

Ce champ est défini comme suit:

```
targetingInformation EXTENSION ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF Targets
    IDENTIFIED BY       id-ce-targetInformation }

Targets    ::=      SEQUENCE SIZE (1..MAX) OF Target

Target     ::=      CHOICE {
    targetName          [0]     GeneralName,
    targetGroup         [1]     GeneralName,
    targetCert          [2]     TargetCert }

TargetCert      ::= SEQUENCE {
    targetCertificate   IssuerSerial,
    targetName          GeneralName OPTIONAL,
    certDigestInfo      ObjectDigestInfo OPTIONAL }
```

Le composant **targetName** s'il est présent, fournit les noms des serveurs/services cibles pour lesquels le certificat d'attribut conteneur est ciblé.

Le composant **targetGroup**, s'il est présent, fornit le nom du groupe cible pour lequel le certificat d'attribut conteneur est ciblé. La manière dont la qualité de membre pour une cible contenue dans le composant **targetGroup** est déterminée ne s'inscrit pas dans le cadre de la présente Spécification.

Le composant **targetCert**, s'il est présent, identifie les serveurs cibles par référence à leur certificat.

Cette extension peut figurer dans des certificats d'attribut émis par des autorités d'attribut, y compris des sources d'autorité, pour des entités qui peuvent agir comme déclarants de privilège, incluant d'autres autorités d'attribut et des entités finales. Cette extension ne figurera pas dans des certificats de clé publique ou dans des certificats d'attribut émis pour des autorités d'attribut qui ne peuvent pas agir également comme déclarants de privilège.

Si cette extension figure dans un certificat émis pour une entité qui est une autorité d'attribut, elle s'applique alors uniquement à la déclaration par cette entité pour les privilèges présents dans le certificat. Elle n'a aucun impact sur la capacité d'émission de certificats par l'autorité d'attribut.

Cette extension est toujours critique.

Le certificat doit être rejeté si cette extension est présente mais que le vérificateur de privilège ne fait pas partie des vérificateurs de privilège spécifiés.

Le certificat d'attribut ne possède pas de cible et peut être accepté par un serveur quelconque si cette extension n'est pas présente.

15.1.2.3 Extension de notification d'utilisateur

L'extension de notification d'utilisateur permet à une autorité d'attribut d'inclure une notification devant être affichée à l'intention d'un détenteur qui déclare un privilège et/ou à l'intention d'un vérificateur de privilège lorsqu'il utilise le certificat d'attribut qui contient cette extension.

Ce champ est défini comme suit:

```
userNotice EXTENSION ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF UserNotice
    DENTIFIED BY        id-ce-userNotice }
```

Cette extension peut figurer dans des certificats d'attribut ou des certificats de clé publique émis par des autorités d'attribut, y compris des sources d'autorité, pour des entités qui peuvent agir comme déclarants de privilège, incluant d'autres autorités d'attribut et des entités finales. Cette extension ne figurera pas dans des certificats qui contiennent l'extension d'identificateur de source d'autorité ou dans des certificats d'attribut émis pour des autorités d'attribut qui ne peuvent pas agir également comme déclarants de privilège.

Si cette extension figure dans un certificat émis pour une entité qui est une autorité d'attribut, elle s'appliquera alors uniquement à la déclaration faite par cette entité pour des privilèges figurant dans le certificat. Elle n'a aucun impact sur la capacité de l'autorité d'attribut pour l'émission de certificats.

Cette extension peut être critique ou non, au choix de l'émetteur de certificat.

Si cette extension est marquée comme critique, les notifications d'utilisateur doivent être affichées à l'intention d'un vérificateur de privilège chaque fois qu'un privilège est déclaré. Si le déclarant de privilège fournit le certificat d'attribut au vérificateur de privilège (c'est-à-dire si le vérificateur de privilège ne l'extrait pas directement d'un référentiel), les notifications d'utilisateur doivent également être affichées à l'intention du déclarant de privilège.

Si cette extension est marquée comme non critique, le privilège déclaré dans le certificat peut être accordé par un vérificateur de privilège sans tenir compte du fait que les notifications ont été affichées ou non à l'intention du déclarant de privilège et/ou du vérificateur de privilège.

15.1.2.4 Extension de politiques de privilège acceptable

Le champ de politiques de privilège acceptable est utilisé pour définir des contraintes concernant la déclaration des privilèges attribués lors de l'utilisation avec un ensemble spécifique de politiques de privilège.

Ce champ est défini comme suit:

```
acceptablePrivilegePolicies EXTENSION ::= {
    SYNTAX              AcceptablePrivilegePoliciesSyntax
    IDENTIFIED BY       id-ce-acceptablePrivilegePolicies }

AcceptablePrivilegePoliciesSyntax       ::=     SEQUENCE SIZE (1..MAX) OF PrivilegePolicy
```

Cette extension peut figurer dans des certificats d'attribut ou des certificats de clé publique émis par des autorités d'attribut, y compris des sources d'autorité, pour d'autres autorités d'attribut ou des entités finales. Si cette extension figure dans un certificat de clé publique, elle concerne alors uniquement la capacité du sujet à agir comme déclarant de privilège pour les privilèges figurant dans l'extension **subjectDirectoryAttributes**.

Cette extension sera marquée comme critique si elle est présente.

Si cette extension est présente et comprise par le vérificateur de privilège, ce dernier doit s'assurer que la politique de privilège qui fait l'objet de la comparaison avec ces privilèges est l'une de celles qui sont indiquées dans cette extension.

Le certificat doit être rejeté si cette extension est présente et n'est pas comprise par le vérificateur de privilège.

15.2 Extensions de révocation de privilège

15.2.1 Définition des besoins

Les besoins suivants sont liés à la révocation de certificats d'attribut:

a) il peut être nécessaire, pour rester maître de la taille des listes CRL, d'assigner des sous-ensembles de tous les certificats émis par une même autorité d'attribut à plusieurs listes CRL;

b) les émetteurs d'attribut de certificat doivent pouvoir indiquer dans un certificat d'attribut qu'il n'existe pas d'informations de révocation pour ce certificat.

15.2.2 Champs d'extension de révocation de privilège

Les champs d'extension suivants sont définis:

a) *points de répartition de liste CRL*;

b) *absence d'informations de révocation*.

15.2.2.1 Extension de point de répartition de liste CRL

L'extension de point de répartition de liste CRL est définie dans la section 2 de la présente Spécification à des fins d'utilisation dans des certificats de clé publique. Ce champ peut également être présent dans un certificat d'attribut. Il peut figurer dans des certificats émis pour des autorités d'attribut, y compris des sources d'autorité, ainsi que dans des certificats émis pour des entités finales.

Si elle figure dans un certificat, un vérificateur de privilège traitera cette extension exactement de la manière décrite dans la section 2 pour des certificats de clé publique.

15.2.2.2 Extension d'absence d'informations de révocation

Il peut ne pas être nécessaire de révoquer des certificats (par exemple lorsque les certificats d'attribut sont émis avec des durées de validité très brèves). Une autorité d'attribut peut utiliser cette extension pour indiquer qu'aucune information de statut de révocation n'est fournie pour ce certificat d'attribut. Ce champ est défini comme suit:

```
noRevAvail EXTENSION ::= {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-noRevAvail }
```

Cette extension peut figurer dans des certificats d'attribut émis par des autorités d'attribut, y compris des sources d'autorité, pour des entités finales. Cette extension ne figurera pas dans des certificats de clé publique ou dans des certificats d'attribut émis pour des autorités d'attribut.

Cette extension est toujours non critique.

Un vérificateur de privilège n'a pas besoin de rechercher des informations de statut de révocation si cette extension figure dans un certificat d'attribut.

15.3 Extensions de source d'autorité

15.3.1 Définition des besoins

Les besoins suivants sont liés aux sources d'autorité:

a) il peut être nécessaire, dans certains environnements, qu'une autorité de certification surveille de manière stricte les entités pouvant agir comme source d'autorité.

b) il est nécessaire de fournir les définitions de syntaxe et les règles de hiérarchie valides pour les attributs de privilège disponibles auprès des sources d'autorité responsables.

15.3.2 Champs d'extension de source d'autorité

Les champs d'extension suivants sont définis:

a) *identificateur de source d'autorité*;

b) *descripteur d'attribut*.

15.3.2.1 Extension d'identificateur de source d'autorité

L'extension d'identificateur de source d'autorité indique que le sujet du certificat peut agir comme source d'autorité à des fins de gestion de privilège. Le sujet du certificat peut de ce fait définir des attributs qui assignent un privilège, émettre des certificats de descripteur d'attribut pour ces attributs et utiliser la clé privée correspondant à la clé publique certifiée pour émettre des certificats qui attribuent un privilège à des détenteurs. Ces nouveaux certificats peuvent être des certificats d'attribut ou des certificats de clé publique avec une extension **subjectDirectoryAttributes** contenant les privilèges.

Cette extension n'est pas nécessaire dans certains environnements et d'autres procédés peuvent être utilisés pour déterminer les entités qui peuvent agir comme sources d'autorité. Cette extension est nécessaire uniquement dans des environnements pour lesquels une surveillance centralisée stricte est nécessaire pour la gestion des entités qui agissent comme sources d'autorité.

Ce champ est défini comme suit:

```
sOAIdentifier EXTENSION  ::= {
    SYNTAX              NULL
    IDENTIFIED BY       id-ce-sOAIdentifier }
```

La capacité du sujet ou du détenteur à agir comme source d'autorité doit être déterminée par d'autres moyens si cette extension ne figure pas dans un certificat.

Ce champ peut uniquement être présent dans un certificat de clé publique émis pour une source d'autorité. Il ne figurera pas dans des certificats d'attribut ou des certificats de clé publique émis pour d'autres autorités d'attribut ou pour des entités finales détentrices de privilège.

Cette extension est toujours non critique.

15.3.2.2 Extension de descripteur d'attribut

Les vérificateurs de privilège ont besoin de connaître la définition d'un attribut de privilège et les règles de hiérarchie qui régissent sa délégation ultérieure, pour s'assurer que l'autorisation est faite de manière correcte. Ces définitions et règles peuvent être fournies aux vérificateurs de privilège par des moyens qui sont en dehors du domaine d'application de la présente Spécification (pouvant par exemple être configurés de manière locale par le vérificateur de privilège).

Cette extension fournit un procédé pouvant être utilisé par une source d'autorité pour fournir aux vérificateurs de privilège des définitions d'attribut et les règles de hiérarchie associées. Un certificat d'attribut qui contient cette extension est appelé certificat de descripteur d'attribut et constitue un certificat d'attribut d'un type spécial. La syntaxe d'un certificat de descripteur d'attribut est identique à celle du type **AttributeCertificate** avec les caractéristiques suivantes:

- le champ **attributes** contient une **SEQUENCE** vide;
- il s'agit d'un certificat auto-émis (c'est-à-dire que l'émetteur et le détenteur sont une même entité);
- il contient l'extension de descripteur d'attribut.

Ce champ est défini comme suit:

```
attributeDescriptor EXTENSION  ::= {
    SYNTAX              AttributeDescriptorSyntax
    IDENTIFIED BY       {id-ce-attributeDescriptor } }

AttributeDescriptorSyntax  ::= SEQUENCE {
    identifier                  AttributeIdentifier,
    attributeSyntax             OCTET STRING (SIZE(1..MAX)),
    name                [0]     AttributeName   OPTIONAL,
    description         [1]     AttributeDescription    OPTIONAL,
    dominationRule              PrivilegePolicyIdentifier}

AttributeIdentifier  ::= ATTRIBUTE.&id({AttributeIDs})

AttributeIDs ATTRIBUTE  ::= {...}

AttributeName  ::= UTF8String(SIZE(1..MAX))

AttributeDescription  ::= UTF8String(SIZE(1..MAX))

PrivilegePolicyIdentifier ::=   SEQUENCE {
    privilegePolicy             PrivilegePolicy,
    privPolSyntax               InfoSyntax }
```

Le composant **identifier** d'une valeur de l'extension **attributeDescriptor** est l'identificateur d'objet identifiant le type d'attribut.

Le composant **attributeSyntax** contient la définition ASN.1 de la syntaxe d'attribut. Cette définition ASN.1 doit être donnée comme spécifié pour le composant **information** de l'attribut opérationnel de règles de correspondance (Matching Rules) définit dans la Rec. UIT-T X.501 | ISO/CEI 9594-2.

Le composant **name** contient facultativement un nom en clair qui permet de reconnaître l'attribut.

Le composant **description** contient facultativement une description en clair de l'attribut.

Le composant **dominationRule** [*règle de hiérarchie*] indique dans quelles conditions un privilège délégué est "inférieur" au privilège correspondant détenu par l'auteur de la délégation. Le composant **privilegePolicy** [*politique de privilège*] indique, au moyen de son identificateur d'objet, l'instance de politique de privilège qui contient les règles. Le composant **privPolSyntax** [*syntaxe de politique privée*] contient, soit la politique de privilège proprement dite, soit un pointeur vers un emplacement où elle peut être trouvée. Dans le cas d'un pointeur, un hachage optionnel de la politique de privilège peut également être présent de manière à permettre une vérification de l'intégrité des données de la politique de privilège concernée.

Cette extension peut figurer uniquement dans des certificats de descripteur d'attribut. Elle ne sera pas présente dans des certificats de clé publique ou des certificats d'attribut autres que ceux émis par des sources d'autorité à leur propre intention.

Cette extension sera toujours non critique.

Le certificat de descripteur d'attribut est créé par la source d'autorité au moment de la création ou de la définition du type d'attribut correspondant; il définit la signification de la contrainte universelle de délégation vers "l'aval" et les moyens de la faire respecter dans l'infrastructure. Les certificats d'attribut contenant cette extension sont stockés dans l'attribut **attributeDescriptorCertificate** [*certificat de descripteur d'attribut*] de l'entrée d'annuaire de la source d'autorité.

15.3.2.2.1 Concordance de descripteur d'attribut

La règle de concordance de descripteur d'attribut compare une valeur présentée avec une valeur d'attribut du type **AttributeCertificate.**

```
attDescriptor MATCHING-RULE  ::= {
    SYNTAX          AttributeDescriptorSyntax
    ID              id-mr-attDescriptorMatch }
```

Cette règle de concordance renvoie la valeur "Vrai" si la valeur stockée contient l'extension **attributeDescriptor** [*descripteur d'attribut*] et si les composants figurant dans la valeur présentée et les composants correspondants de la valeur stockée concordent.

15.4 Extensions de rôle

15.4.1 Définition des besoins

Les besoins suivants sont liés aux rôles:

– si un certificat est un certificat d'attribution de rôle, un vérificateur de privilège doit alors être en mesure de localiser le certificat de spécification de rôle correspondant qui contient les privilèges spécifiques attribués au rôle proprement dit.

15.4.2 Champs d'extension de rôle

Le champ d'extension suivant est défini:

– *identificateur de certificat de spécification de rôle*.

15.4.2.1 Extension d'identificateur de certificat de spécification de rôle

Cette extension peut être utilisée par une autorité d'attribut comme pointeur vers un certificat de spécification de rôle contenant l'attribution de privilèges à un rôle. Elle peut figurer dans un certificat d'attribution de rôle (c'est-à-dire un certificat qui contient l'attribut **role**).

Lorsqu'il traite un certificat d'attribution de rôle, un vérificateur de privilège doit pouvoir prendre connaissance de l'ensemble de privilèges de ce rôle afin de déterminer si la vérification réussit ou échoue. Si les privilèges ont été attribués au rôle dans un certificat de spécification de rôle, ce champ peut alors être utilisé pour localiser ce certificat.

Ce champ est défini comme suit:

```
roleSpecCertIdentifier EXTENSION ::=
    {
    SYNTAX              RoleSpecCertIdentifierSyntax
    IDENTIFIED BY       { id-ce-roleSpecCertIdentifier }
}

RoleSpecCertIdentifierSyntax  ::= SEQUENCE SIZE (1..MAX) OF RoleSpecCertIdentifier

RoleSpecCertIdentifier  ::= SEQUENCE {
    roleName                    [0]     GeneralName,
    roleCertIssuer              [1]     GeneralName,
    roleCertSerialNumber        [2]     CertificateSerialNumber     OPTIONAL,
    roleCertLocator             [3]     GeneralNames                OPTIONAL
    }
```

Le composant **roleName** identifie le rôle. Ce nom est le même que celui qui figure dans le composant **holder** du certificat de spécification de rôle auquel cette extension fait référence.

Le composant **roleCertIssuer** [*émetteur de certificat de rôle*] identifie l'autorité d'attribut émettrice du certificat de spécification de rôle qui fait l'objet de la référence.

Le composant **roleCertSerialNumber** [*numéro de série de certificat de rôle*] contient, s'il est présent, le numéro de série du certificat de spécification de rôle. Il convient de noter que si les privilèges attribués au rôle proprement dit sont modifiés, un nouveau certificat de spécification de rôle sera alors émis pour le rôle. Tout certificat contenant cette extension avec le composant **roleCertSerialNumber** doit alors être remplacé par un certificat qui fait référence au nouveau numéro de série. Ce comportement est nécessaire dans certains environnements, mais il n'est pas souhaitable dans beaucoup d'autres. Ce composant sera en général absent, ce qui permet la mise à jour automatique des privilèges attribués au rôle proprement dit sans impact sur les certificats d'attribution de rôle.

Le composant **roleCertLocator** [*localisation du certificat de rôle*] contient, s'il est présent, des informations pouvant être utilisées pour localiser le certificat de spécification de rôle.

Cette extension peut figurer dans des certificats d'attribution de rôle qui sont des certificats d'attribut ou des certificats de clé publique émis par des autorités d'attribut, y compris des sources d'autorité, pour d'autres autorités d'attribut ou pour des entités finales détentrices de privilège. Cette extension ne figurera pas dans des certificats qui contiennent l'extension d'identificateur de source d'autorité.

Cette extension peut être utilisée, si elle est présente, par un vérificateur de privilège pour localiser le certificat de spécification de rôle.

Si cette extension n'est pas présente, soit:

a) d'autres moyens doivent alors être utilisés pour localiser le certificat de spécification de rôle; soit,

b) des procédés ne faisant pas appel à un certificat de spécification de rôle ont été utilisés pour attribuer des privilèges au rôle (les privilèges de rôle peuvent, par exemple, être configurés de manière locale pour le vérificateur de privilège).

Cette extension est toujours non critique.

15.4.2.1.1 Concordance d'identificateur de certificat de spécification de rôle

La règle de concordance d'identificateur de certificat de spécification de rôle compare une valeur présentée avec une valeur d'attribut du type **AttributeCertificate.**

```
roleSpecCertIdMatch MATCHING-RULE  ::= {
    SYNTAX          RoleSpecCertIdentifierSyntax
    ID              id-mr-roleSpecCertIdMatch }
```

Cette règle de concordance renvoie la valeur "Vrai" si la valeur stockée contient l'extension **roleSpecCertIdentifier** et si les composants figurant dans la valeur présentée et les composants correspondants de la valeur stockée concordent.

15.5 Extensions de délégation

15.5.1 Définition des besoins

Les besoins suivants sont liés à la délégation de privilèges:

a) les certificats de privilège d'entité finale doivent pouvoir être distingués des certificats d'autorité d'attribut de manière à interdire que des entités finales s'établissent elles-mêmes comme autorités d'attribut d'une manière non autorisée. Une autorité d'attribut doit également avoir la possibilité d'imposer une limite à la taille de la suite d'un itinéraire de délégation;

b) une autorité d'attribut doit pouvoir spécifier l'espace de noms adéquat au sein duquel la délégation de privilège peut s'effectuer. Le vérificateur de privilège doit pouvoir vérifier le respect de ces contraintes;

c) une autorité d'attribut doit pouvoir spécifier les politiques de certificat acceptables que les déclarants de privilège doivent utiliser en aval sur un itinéraire de délégation pour s'authentifier lorsqu'ils déclarent une délégation de privilège auprès de cette autorité d'attribut;

d) un vérificateur de privilège doit pouvoir localiser le certificat d'attribut correspondant à un émetteur, de manière à s'assurer que ce dernier dispose du privilège suffisant pour la délégation du privilège figurant dans le certificat en cours.

15.5.2 Champs d'extension de délégation

Les champs d'extension suivants sont définis:

a) *contraintes d'attribut de base;*

b) *contraintes de nom délégué;*

c) *politiques de certificat acceptable;*

d) *identificateur d'autorité d'attribut*.

15.5.2.1 Extension de contraintes d'attribut de base

Ce champ indique si une délégation ultérieure est autorisée pour des privilèges attribués dans le certificat où il figure. Dans ce cas, il peut également spécifier une contrainte de longueur pour l'itinéraire de délégation.

Ce champ est défini comme suit:

```
basicAttConstraints EXTENSION ::=
{
        SYNTAX                  BasicAttConstraintsSyntax
        IDENTIFIED BY           { id-ce-basicAttConstraints }
}

BasicAttConstraintsSyntax ::= SEQUENCE
    {
        authority               BOOLEAN DEFAULT FALSE,
        pathLenConstraint       INTEGER (0..MAX) OPTIONAL
    }
```

Le composant **authority** [*autorité*] indique si le détenteur est autorisé ou non à effectuer une nouvelle délégation du privilège. Si la valeur du composant **authority** est égale à "Vrai" le détenteur est alors également une autorité d'attribut et il est autorisé à déléguer le privilège à son tour, compte tenu des contraintes pertinentes. Si la valeur du composant **authority** est égale à "Faux", le détenteur est alors une entité finale et il n'est pas autorisé à déléguer le privilège.

Le composant **pathLenConstraint** est significatif uniquement si le composant **authority** est positionné sur "Vrai". Il donne le nombre maximal de certificats d'autorité d'attribut qui peuvent faire suite à ce certificat sur un itinéraire de délégation. Une valeur nulle indique que le sujet de ce certificat peut uniquement émettre des certificats pour des entités finales et non pour des autorités d'attribut. Si le champ **pathLenConstraint** n'est présent dans aucun des certificats d'un itinéraire de délégation, aucune limite n'est alors imposée à la longueur de l'itinéraire de délégation. Il convient de noter que la contrainte prend effet à partir du certificat suivant sur l'itinéraire. La contrainte limite le nombre de certificats d'autorité d'attribut entre le certificat de l'autorité d'attribut qui contient la contrainte et le certificat de l'entité finale. La longueur totale de l'itinéraire peut de ce fait être au plus supérieure de deux certificats à la valeur indiquée par cette contrainte. Ceci tient compte des certificats des deux points d'extrémité et des certificats d'autorité d'attribut intermédiaires entre les deux points d'extrémité qui sont soumis à la contrainte indiquée par la valeur de cette extension.

Cette extension peut figurer dans des certificats d'attribut ou des certificats de clé publique émis par des autorités d'attribut, y compris des sources d'autorité, pour d'autres autorités d'attribut ou des entités finales. Cette extension ne figurera pas dans des certificats qui contiennent l'extension d'identificateur de source d'autorité.

Si cette extension figure dans un certificat d'attribut et si la valeur du composant **authority** est égale à "Vrai", le détenteur est alors autorisé à émettre à son tour des certificats d'attribut qui délèguent les privilèges qu'il contient à d'autres entités, mais sans délégation de certificats de clé publique.

Si cette extension figure dans un certificat de clé publique et si l'extension **basicConstraints** indique que le sujet est également une autorité de certification, ce sujet est alors autorisé à émettre à son tour des certificats de clé publique qui délèguent ces privilèges à d'autres entités, mais pas à émettre des certificats d'attribut. Si la contrainte de longueur d'itinéraire est présente, le sujet peut alors uniquement effectuer une délégation pour les contraintes appartenant à l'intersection de la contrainte spécifiée dans cette extension et toutes les contraintes spécifiées dans l'extension **basicConstraints**. Si cette extension figure dans un certificat de clé publique et si l'extension **basicConstraints** n'y figure pas ou indique que le sujet est une entité finale, ce dernier n'est alors pas autorisé à déléguer les privilèges.

Cette extension peut être critique ou non, au choix de l'émetteur de certificat. Il est recommandé qu'elle soit critique, faute de quoi un détenteur qui n'est pas autorisé comme autorité d'attribut peut émettre des certificats et le vérificateur de privilège peut utiliser un tel certificat par inadvertance.

Les actions suivantes s'appliquent si cette extension est présente et marquée comme critique:

- si la valeur du composant **authority** n'est pas positionnée sur "Vrai", l'attribut délégué ne sera alors pas utilisé pour de nouvelles délégations;
- si la valeur du composant **authority** est positionnée sur "Vrai" et si le composant **pathLenConstraint** est présent, le vérificateur de privilège vérifiera alors si l'itinéraire de délégation en cours de traitement respecte la contrainte **pathLenConstraint**.

Si cette extension est présente, marquée comme non critique et si elle n'est pas reconnue par le vérificateur de privilège, le système doit alors utiliser d'autres moyens pour déterminer si l'attribut délégué peut être utilisé pour une nouvelle délégation.

Si cette extension n'est pas présente ou si elle est présente avec une valeur de **SEQUENCE** vide, le détenteur doit alors être uniquement une entité finale et non une autorité d'attribut et aucune délégation des privilèges contenus dans le certificat d'attribut n'est permise par le détenteur.

15.5.2.1.1 Concordance de contraintes d'attribut de base

La règle de concordance de contraintes d'attribut de base compare une valeur présentée avec une valeur d'attribut du type **AttributeCertificate.**

```
basicAttConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          BasicAttConstraintsSyntax
    ID              id-mr-basicAttConstraintsMatch }
```

Cette règle de concordance renvoie la valeur "Vrai" si la valeur stockée contient l'extension **basicAttConstraints** [*contraintes d'attribut de base*] et si les composants figurant dans la valeur présentée et les composants correspondants de la valeur stockée concordent.

15.5.2.2 Extension de contraintes de nom délégué

Le champ de contraintes de nom délégué indique un espace de noms auquel doivent appartenir tous les noms de détenteurs des certificats suivants sur l'itinéraire de délégation.

Ce champ est défini comme suit:

```
delegatedNameConstraints EXTENSION ::= {
    SYNTAX              NameConstraintsSyntax
    IDENTIFIED BY       id-ce-delegatedNameConstraints }
```

Cette extension est traitée de la même manière que l'extension **nameConstraints** pour des certificats de clé publique. Si le composant **permittedSubtrees** est présent, il indique alors que seuls les certificats d'attributs dont les noms de détenteur appartiennent à ces sous-arbres sont acceptables parmi tous les certificats d'attribut émis par l'autorité AA détentrice et les autorités suivantes sur l'itinéraire de délégation. Si le composant **excludedSubtrees** est présent, il indique alors que tout certificat d'attribut, émis par l'autorité AA détentrice ou par une autorité d'attribut suivante sur l'itinéraire de délégation, n'est pas acceptable si le nom du détenteur appartient à l'un de ces sous-arbres. La déclaration d'exclusion a priorité si les deux composants **permittedSubtrees** et **excludedSubtrees** sont présents et que les espaces de nom se chevauchent.

Cette extension peut figurer dans des certificats d'attribut ou des certificats de clé publique émis par des autorités d'attribut, y compris des sources d'autorité, pour d'autres autorités d'attribut. Cette extension ne figurera pas dans des certificats émis pour des entités finales ou des certificats qui contiennent l'extension d'identificateur de source d'autorité.

Si cette extension figure dans un certificat de clé publique et si l'extension **nameConstraints** est également présente, le sujet peut alors uniquement effectuer une délégation pour les contraintes appartenant à l'intersection de la contrainte spécifiée dans cette extension et de celle qui est spécifiée dans l'extension **nameConstraints**.

Cette extension peut être critique ou non, au choix de l'émetteur de certificat d'attribut. Il est recommandé qu'elle soit critique, faute de quoi un utilisateur d'attribut peut omettre de vérifier que les certificats d'attribut suivants sur un itinéraire de délégation appartiennent à l'espace de noms prévu par l'autorité d'attribut émettrice.

15.5.2.2.1 Concordance de contraintes de nom délégué

La règle de concordance de contraintes de nom délégué compare une valeur présentée avec une valeur d'attribut du type **AttributeCertificate.**

```
delegatedNameConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          NameConstraintsSyntax
    ID              id-mr-delegatedNameConstraintsMatch}
```

Cette règle de concordance renvoie la valeur "Vrai" si la valeur stockée contient l'extension **attributeNameConstraints** [*contraintes de nom d'attribut*] et si les composants figurant dans la valeur présentée et les composants correspondants de la valeur stockée concordent.

15.5.2.3 Extension de politiques de certificat acceptable

Le champ de politiques de certificat acceptable est utilisé, dans la délégation avec certificats d'attribut, pour gérer les politiques de certificat acceptable sous lesquelles ont dû être émis les certificats de clé publique pour des détenteurs suivants sur un itinéraire de délégation. L'énumération d'un ensemble de politiques dans ce champ permet à une autorité d'attribut d'exiger que les émetteurs suivants sur un itinéraire de délégation délèguent les privilèges contenus dans le certificat uniquement à des détenteurs de certificats de clé publique émis conformément à une ou plusieurs des politiques de certificat énumérées. Les politiques indiquées dans cette liste ne sont pas des politiques portant sur l'émission du certificat, mais des politiques conformément auxquelles des certificats de clé publique acceptables ont dû être émis pour des détenteurs suivants.

Ce champ est défini comme suit:

```
acceptableCertPolicies EXTENSION ::= {
    SYNTAX              AcceptableCertPoliciesSyntax
    IDENTIFIED BY       id-ce-acceptableCertPolicies }

AcceptableCertPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId

CertPolicyId ::= OBJECT IDENTIFIER
```

Cette extension peut figurer uniquement dans des certificats d'attribut émis par des autorités d'attribut, y compris des sources d'autorité, pour d'autres autorités d'attribut. Cette extension ne figurera pas dans des certificats d'entité finale d'attribut ou dans tout certificat de clé publique. La même fonctionnalité est fournie par le composant **certificatePolicies** [*politiques de certificat*] et d'autres extensions connexes dans le cas de la délégation utilisant des certificats de clé publique.

Cette extension sera marquée comme critique si elle est présente.

Si cette extension est présente et comprise par le vérificateur de privilège, ce dernier doit alors s'assurer que tous les déclarants de privilège suivants sur l'itinéraire de délégation sont authentifiés par un certificat de clé publique établi conformément à une ou plusieurs des politiques de certificat énumérées.

Le certificat doit être rejeté si cette extension est présente et n'est pas comprise par le vérificateur de privilège.

15.5.2.3.1 Concordance de politiques de certificat acceptable

La règle de concordance de politiques de certificat acceptable compare une valeur présentée avec une valeur d'attribut du type **AttributeCertificate.**

```
acceptableCertPoliciesMatch MATCHING-RULE  ::= {
    SYNTAX          AcceptableCertPoliciesSyntax
    ID              id-mr-acceptableCertPoliciesMatch }
```

Cette règle de concordance renvoie la valeur "Vrai" si la valeur stockée contient l'extension **acceptableCertPolicies** [*politiques de certificat acceptable*] et si les composants figurant dans la valeur présentée et les composants correspondants de la valeur stockée concordent.

15.5.2.4 Extension d'identificateur d'autorité d'attribut

Une autorité qui délègue des privilèges doit posséder au moins le même privilège et l'autorisation de le déléguer. Une autorité d'attribut qui délègue un privilège à une autre autorité d'attribut ou à une entité finale peut faire figurer cette extension dans le certificat d'autorité d'attribut ou le certificat d'entité finale qu'elle émet. L'extension constitue un pointeur en retour vers le certificat dans lequel l'émetteur du certificat qui contient cette extension a reçu l'attribution de son privilège correspondant. L'extension peut être utilisée par un vérificateur de privilège pour s'assurer que l'autorité d'attribut émettrice possède un privilège suffisant qui lui permet d'effectuer la délégation pour le détenteur du certificat qui contient cette extension.

Ce champ est défini comme suit:

```
authorityAttributeIdentifier EXTENSION  ::=
    {
    SYNTAX              AuthorityAttributeIdentifierSyntax
    IDENTIFIED BY       { id-ce-authorityAttributeIdentifier }
    }
AuthorityAttributeIdentifierSyntax      ::=  SEQUENCE SIZE (1..MAX) OF AuthAttId
AuthAttId       ::=  IssuerSerial
```

Un certificat contenant cette extension peut contenir plusieurs délégations de privilège pour le détenteur du certificat. Si l'attribution de ces privilèges à l'autorité d'attribut qui a émis ce certificat a été effectuée par plusieurs certificats, l'extension contiendra alors plusieurs pointeurs.

Cette extension peut figurer dans des certificats d'attribut ou des certificats de clé publique émis par des autorités d'attribut pour d'autres autorités d'attribut ou pour des entités finales détentrices de privilège. Cette extension ne figurera pas dans des certificats émis par une source d'autorité ou dans des certificats de clé publique qui contiennent l'extension d'identificateur de source d'autorité.

Cette extension est toujours non critique.

15.5.2.4.1 Concordance d'identificateur d'autorité d'attribut

La règle de concordance d'identificateur d'autorité d'attribut compare une valeur présentée avec une valeur d'attribut du type **AttributeCertificate**.

```
authAttIdMatch MATCHING-RULE  ::= {
    SYNTAX          AuthorityAttirbuteIdentifierSyntax
    ID              id-mr-authAttIdMatch }
```

Cette règle de concordance renvoie la valeur "Vrai" si la valeur stockée contient l'extension **authorityAttributeIdentifier** [*identificateur d'attribut d'autorité*] et si les composants figurant dans la valeur présentée et les composants correspondants de la valeur stockée concordent.

16 Procédure de traitement d'itinéraire de privilège

Le traitement d'itinéraire de privilège est effectué par un vérificateur de privilège. Les règles de traitement d'itinéraire pour les certificats d'attribut et les certificats de clé publique sont relativement comparables.

Les autres composants du traitement d'itinéraire, à savoir la vérification des signatures de certificat, la validation de la durée de validité des certificats, etc., ne sont pas traités dans cet article.

La procédure de base, décrite au 16.1 ci-dessous, est la seule qui soit nécessaire pour des itinéraires de privilèges constitués d'un seul certificat (dans ce cas, leur attribution au déclarant de privilège a été effectuée par la source d'autorité), à moins que le privilège n'ait été attribué à un rôle. Dans ce dernier cas, le vérificateur peut être dans l'obligation d'obtenir le certificat de spécification de rôle qui attribue au rôle le privilège en question comme décrit au 16.2 ci-dessous, s'il n'a pas été configuré avec le privilège propre au rôle. La procédure de délégation d'itinéraire du 16.3 est nécessaire en outre si le déclarant de privilège a reçu la délégation du privilège d'une autorité d'attribut intermédiaire. Ces procédures ne sont pas effectuées en séquence. La procédure de traitement de rôle et la procédure de traitement de délégation sont effectuées avant d'avoir déterminé si les privilèges déclarés sont suffisants ou non dans le contexte d'utilisation au sein de la procédure de base.

16.1 Procédure de traitement de base

Il est nécessaire de vérifier la signature de chaque certificat de l'itinéraire. Les procédures relatives à la validation des signatures et des certificats de clé publique ne sont pas reproduites dans ce paragraphe. Le vérificateur de privilège doit vérifier l'identité de chaque entité de l'itinéraire en appliquant les procédures de l'article 10. Il convient de noter que la signature d'un certificat d'attribut implique nécessairement la vérification de la validité du certificat de clé publique de référence. Lorsque des privilèges sont attribués en utilisant des certificats d'attribut, les procédures de traitement d'itinéraire devront prendre en considération, dans le processus de détermination de la validité ultime du certificat d'attribut d'un déclarant de privilège, des éléments appartenant à l'infrastructure PMI et à l'infrastructure PKI. Une fois que la validité a été confirmée, les privilèges contenus dans ce certificat peuvent éventuellement être utilisés, en fonction d'une comparaison avec la politique de privilège pertinente et d'autres informations faisant partie du contexte d'utilisation du certificat.

Le contexte d'utilisation doit déterminer si le détenteur de privilège a effectivement l'intention de déclarer le privilège contenu pour une utilisation dans ce contexte. L'existence d'une chaîne de certificats fiables vers une source d'autorité ne suffit pas pour effectuer cette détermination. L'intention du détenteur de privilège d'utiliser ce certificat doit être clairement indiquée et vérifiée. Les procédés pour s'assurer qu'une telle déclaration de privilège a été clairement exprimée par le détenteur de privilège sont toutefois en dehors du domaine d'application de la présente Spécification. Cette déclaration de privilège peut, par exemple, être vérifiée si le détenteur de privilège a signé une référence concernant ce certificat, ce qui indique son intention d'utiliser ce certificat dans ce contexte.

Le vérificateur de privilège doit s'assurer que tout certificat d'attribut de l'itinéraire qui ne contient pas l'extension **noRevAvail** [*pas de révocation disponible*] n'a pas été révoqué.

Le vérificateur de privilège doit s'assurer que le privilège déclaré est valide pour l'instant appelé "instant d'évaluation"; cette vérification peut être faite à un instant quelconque, c'est-à-dire à l'instant actuel de la vérification ou pour tout instant dans le passé. La vérification est toujours faite à l'instant actuel dans le contexte d'un service de contrôle d'accès. Toutefois, dans le contexte de non-répudiation, la vérification peut être faite pour un instant antérieur ou pour l'instant actuel. Une fois que les certificats ont été validés, le vérificateur de privilège doit s'assurer que l'instant d'évaluation appartient à la durée de validité de tous les certificats utilisés sur l'itinéraire. En outre, si l'un quelconque des certificats de l'itinéraire contient l'extension **timeSpecification**, les contraintes concernant les instants auxquels le privilège peut être déclaré doivent également assurer que la déclaration de privilège est valide à l'instant d'évaluation.

Si l'extension **targetingInformation** [*informations de cible*] figure dans le certificat utilisé pour déclarer un privilège, le vérificateur de privilège doit alors vérifier que le serveur ou le service qu'il vérifie appartient à la liste des cibles.

La procédure de traitement décrite au 16.2 est nécessaire pour s'assurer que les privilèges adéquats sont identifiés si le certificat est un certificat d'attribution de rôle. La procédure du traitement décrite au 16.3 est nécessaire pour s'assurer que la délégation a été faite correctement si le privilège a été délégué à l'entité et non attribué directement par la source d'autorité bénéficiant de la confiance du vérificateur de privilège.

Le vérificateur de privilège doit également déterminer si les privilèges déclarés sont suffisants pour le contexte d'utilisation. La politique de privilège définit les règles utilisées pour cette détermination et englobe la spécification de toutes les variables d'environnement devant être prises en considération. Les privilèges déclarés, y compris ceux qui découlent de la procédure de rôle du 16.2 et de la procédure de délégation du 16.3, ainsi que toutes les variables d'environnement pertinentes (par exemple l'heure du jour ou le solde actuel d'un compte) font l'objet d'une comparaison avec la politique de privilège pour déterminer s'ils suffisent ou non pour le contexte d'utilisation. Si l'extension **acceptablePrivilegePolicies** [*politiques de privilège acceptable*] est présente, la déclaration de privilège peut réussir uniquement si la politique de privilège utilisée par le vérificateur de privilège pour cette comparaison fait partie de celles qui figurent dans cette extension.

Toutes les notifications d'utilisateurs pertinentes sont fournies au vérificateur de privilège si la vérification réussit.

16.2 Procédure de traitement d'itinéraire de privilège

Le vérificateur de privilège doit obtenir les privilèges particuliers attribués au rôle si le certificat déclaré est un certificat d'attribution de rôle. Le nom du rôle assigné au déclarant de privilège figure dans l'attribut **role** du certificat. Si sa configuration ne contient pas déjà les privilèges du rôle nommé, le vérificateur de privilège peut alors avoir besoin de localiser le certificat de spécification de rôle qui attribue les privilèges à ce rôle. Les informations contenues dans l'attribut **role** et dans l'extension **roleSpecCertIdentifier** peuvent être utilisées pour localiser ce certificat.

Les privilèges attribués au rôle sont également attribués de manière implicite au déclarant de privilège et figurent de ce fait dans les privilèges déclarés qui font l'objet de la comparaison avec la politique de privilège de la procédure de base du 16.1, qui détermine si les privilèges déclarés suffisent dans le contexte d'utilisation.

16.3 Procédure de traitement de délégation

Le vérificateur de privilège doit s'assurer de la validité de l'itinéraire de délégation en effectuant les vérifications suivantes si les privilèges déclarés sont délégués au déclarant de privilège par une autorité d'attribut intermédiaire:

- chacune des autorités d'attribut qui a émis un certificat sur l'itinéraire de délégation était autorisée à le faire;
- chaque certificat de l'itinéraire de délégation est valide, compte tenu de l'itinéraire et des contraintes de nom qui sont imposés sur ce dernier;
- chaque entité de l'itinéraire de délégation est authentifiée par un certificat de clé publique valide, compte tenu de toutes les contraintes de politiques imposées;
- aucune autorité d'attribut n'a généré un privilège supérieur à celui qu'elle détient.

Le vérificateur de privilège doit avoir obtenu les résultats suivants avant de commencer la validation d'un itinéraire de délégation. Ces informations peuvent être fournies par le déclarant de privilège ou obtenues par le vérificateur de privilège à partir d'autres sources, telles que l'annuaire. Les attributs du service peuvent être fournis au vérificateur de privilège dans un document structuré ou par d'autres moyens:

- établissement de la confiance dans la clé publique de vérification utilisée pour la validation de la signature de la source d'autorité. Cette confiance peut être établie, soit par des moyens hors bande, soit par un certificat de clé publique émis pour la source d'autorité par une autorité de certification avec laquelle le vérificateur de privilège a déjà établi la confiance. Ce certificat éventuel contient l'extension **sOAIdentifier** [*identificateur de source d'autorité*];
- privilège du déclarant de privilège, codé dans son certificat d'attribut ou dans l'extension d'attributs d'annuaire du certificat de clé publique du sujet;
- itinéraire de délégation de certificats allant du déclarant de privilège vers la source d'autorité;
- règle de hiérarchie pour le privilège déclaré. Elle peut être obtenue à partir du descripteur d'attribut émis par la source d'autorité responsable de l'attribut en question ou par des moyens hors bande;
- politique de privilège, obtenue de l'annuaire ou par des moyens hors bande;
- variables d'environnement, par exemple, l'heure et le jour actuels, le solde actuel d'un compte, etc.

Une implémentation sera fonctionnellement équivalente au comportement externe résultant de cette procédure. L'algorithme utilisé pour une implémentation particulière pour fournir les sorties correctes à partir des entrées données n'est pas normalisé.

16.3.1 Vérification de l'intégrité des données de la règle de hiérarchie

La règle de hiérarchie est associée au privilège délégué. La syntaxe et la méthode d'obtention de la règle de hiérarchie ne sont pas normalisées. Il est toutefois possible de vérifier l'intégrité de la règle de hiérarchie qui a été extraite. Le certificat de descripteur d'attribut émis par la source d'autorité responsable de l'attribut qui est délégué peut contenir un hachage de la règle de hiérarchie. Le vérificateur de privilège peut appliquer la fonction de hachage à la copie extraite pour la règle de hiérarchie et comparer les deux hachages; le vérificateur de privilège a obtenu la règle de hiérarchie correcte si les deux sont identiques.

16.3.2 Etablir un itinéraire de délégation valide

Le vérificateur de privilège doit trouver l'itinéraire de délégation et obtenir des certificats valides pour toutes les entités situées sur l'itinéraire. L'itinéraire de délégation va du déclarant de privilège direct vers la source d'autorité. Tout certificat intermédiaire sur l'itinéraire de délégation doit contenir l'extension **basicAttConstraints** avec le composant d'autorité positionné sur la valeur "Vrai". L'émetteur de chaque certificat doit être le même que le détenteur ou le sujet du certificat adjacent sur l'itinéraire de délégation. L'extension **authorityAttributeIdentifier** est utilisée pour localiser le certificat adéquat de l'entité adjacente sur l'itinéraire de délégation. Le nombre de certificats sur l'itinéraire ne sera pas supérieur de deux unités à la valeur du composant **pathLenConstraint** figurant dans l'extension **basicAttConstraints** de l'entité. La raison en est que la contrainte **pathLenConstraint** limite le nombre de certificats intermédiaires entre les deux points d'extrémité (c'est-à-dire entre le certificat qui contient la contrainte et le certificat d'entité finale), de sorte que la longueur maximale est égale à la valeur de cette contrainte augmentée de deux pour tenir compte des certificats des points d'extrémité.

Si l'extension **delegatedNameConstraints** [*contraintes de nom délégué*] figure dans l'un quelconque des certificats de l'itinéraire de délégation, la contrainte est alors traitée de la même manière que pour l'extension **nameConstraints** dans la procédure de traitement de l'itinéraire de certification de l'article 10.

Si l'extension **acceptableCertPolicies** [*politiques de certificat acceptable*] figure dans l'un quelconque des certificats de l'itinéraire de délégation, le vérificateur de privilège s'assurera alors que l'authentification de chaque entité suivante de l'itinéraire de délégation est faite en utilisant un certificat de clé publique qui contient au minimum une des politiques acceptables.

16.3.3 Vérification de la délégation de privilège

Une entité ne peut pas déléguer de privilège supérieur à celui qu'elle possède. La règle de hiérarchie figurant dans le descripteur d'attribut fournit les règles permettant de déterminer si une valeur donnée de l'attribut qui fait l'objet d'une délégation est "inférieure" à une autre valeur.

Pour chaque certificat de l'itinéraire de délégation, y compris le certificat direct du déclarant de privilège, le vérificateur de privilège doit s'assurer que l'entité faisant la délégation est autorisée à déléguer le privilège qu'elle détient et que le privilège délégué n'est pas supérieur au privilège détenu.

Le vérificateur doit comparer, pour chacun de ces certificats, le privilège délégué avec le privilège détenu par l'auteur de la délégation, conformément à la règle de hiérarchie pour le privilège. Le privilège détenu par l'entité effectuant la délégation est obtenu à partir du certificat adjacent sur l'itinéraire de délégation, comme décrit au 16.2. La comparaison est faite sur la base de la règle de hiérarchie présentée au 16.3.1.

16.3.4 Détermination de la réussite ou de l'échec

Une fois que l'existence d'un itinéraire de délégation valide a été établie, les privilèges du déclarant de privilège direct sont fournis en entrée de la comparaison avec la politique de privilège, comme indiqué au 16.1, pour déterminer si le déclarant de privilège possède ou non un privilège suffisant pour le contexte d'utilisation.

17 Schéma d'annuaire PMI

Cet article définit les éléments de schéma d'annuaire utilisés pour représenter les informations d'infrastructure PMI dans l'annuaire. Il contient les spécifications des classes d'objets, des attributs et de règles de concordance de valeur d'attributs qui s'appliquent.

17.1 Classes d'objets "annuaire PMI"

Ce paragraphe contient les définitions de classe d'objets pour la représentation des objets d'infrastructure PMI dans l'annuaire.

17.1.1 Classe d'objets "utilisateur d'infrastructure PMI"

La classe d'objets "utilisateur d'infrastructure PMI" sert à définir des entrées pour des objets pouvant être détenteurs de certificats d'attribut.

```
pmiUser OBJECT-CLASS ::= {
-- utilisateur d'infrastructure PMI (c'est-à-dire, un "détenteur")
  SUBCLASS OF    {top}
  KIND           auxiliary
  MAY CONTAIN    {attributeCertificateAttribute}
  ID             id-oc-pmiUser }
```

17.1.2 Classe d'objets "autorité d'attribut d'infrastructure PMI"

La classe d'objets "autorité d'attribut d'infrastructure PMI" sert à définir des entrées pour des objets pouvant agir comme autorités d'attribut.

```
pmiAA OBJECT-CLASS ::= {
-- autorité d'attribut d'infrastructure PMI
  SUBCLASS OF    {top}
  KIND           auxiliary
  MAY CONTAIN    {aACertificate |
                  attributeCertificateRevocationList |
                  attributeAuthorityRevocationList}
  ID             id-oc-pmiAA }
```

17.1.3 Classe d'objets "source d'autorité d'infrastructure PMI"

La classe d'objets "source d'autorité d'infrastructure PMI" sert à définir des entrées pour des objets pouvant agir comme source d'autorité. Il convient de noter que si l'objet est autorisé à agir comme source d'autorité par l'émission d'un certificat de clé publique qui contient l'extension **sOAIdentifier**, il existera alors également une entrée d'annuaire représentant la classe d'objets **pkiCA**.

```
pmiSOA OBJECT-CLASS ::= {  -- source d'autorité d'infrastructure PMI
    SUBCLASS OF  {top}
    KIND         auxiliary
    MAY CONTAIN  {attributeCertificateRevocationList |
                  attributeAuthorityRevocationList |
                  attributeDescriptorCertificate}
    ID           id-oc-pmiSOA }
```

17.1.4 Classe d'objets "certificat d'attribut de point de répartition de liste CRL"

La classe d'objets "certificat d'attribut de point de répartition de liste CRL" sert à définir des entrées pour des objets qui contiennent des certificats d'attribut et/ou des segments de liste de révocation d'autorité d'attribut. Il est prévu que cette classe auxiliaire soit utilisée conjointement à la classe d'objets structurés **crlDistributionPoint** lorsque des entrées sont instanciées. Etant donné que les attributs **certificateRevocationList** et **authorityRevocationList** sont optionnels dans cette classe, il est possible de créer des entrées qui contiennent, par exemple, uniquement une liste de révocation d'autorité d'attribut ou des entrées qui contiennent des listes de révocation de plusieurs types, en fonction des besoins.

```
attCertCRLDistributionPt     OBJECT-CLASS ::= {
  SUBCLASS OF    {top}
  KIND           auxiliary
  MAY CONTAIN    { attributeCertificateRevocationList |
                   attributeAuthorityRevocationList }
  ID             id-oc-attCertCRLDistributionPts }
```

17.1.5 Classe d'objets "itinéraire de délégation d'infrastructure PMI"

La classe d'objets "itinéraire de délégation d'infrastructure PMI" sert à définir des entrées pour des objets pouvant contenir des itinéraires de délégation. Elle sera utilisée en général conjointement à la classe d'objets structurés **pmiAA** [*autorité d'attribut d'infrastructure PMI*].

```
pmiDelegationPath      OBJECT-CLASS   ::= {
    SUBCLASS OF        {top}
    KIND               auxiliary
    MAY CONTAIN        { delegationPath }
    ID                 id-oc-pmiDelegationPath }
```

17.1.6 Classe d'objets "politique de privilège"

La classe d'objets "politique de privilège" sert à définir des entrées pour des objets qui contiennent des informations de politique de privilège.

```
privilegePolicy      OBJECT-CLASS   ::= {
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      {privPolicy }
    ID               id-oc-privilegePolicy }
```

17.2 Attributs d'annuaire d'infrastructure PMI

Ce paragraphe définit les attributs d'annuaire utilisés pour stocker les données d'infrastructure PMI dans des entrées d'annuaire.

17.2.1 Attribut "certificat d'attribut"

L'attribut suivant contient des certificats d'attribut émis pour un détenteur spécifique et stockés dans l'entrée d'annuaire de ce dernier.

```
attributeCertificateAttribute ATTRIBUTE  ::= {
    WITH SYNTAX                    AttributeCertificate
    EQUALITY MATCHING RULE         attributeCertificateExactMatch
    ID                             id-at-attributeCertificate }
```

17.2.2 Attribut "certificats d'autorité d'attribut"

L'attribut suivant contient des certificats d'attribut émis pour une autorité d'attribut et stockés dans l'entrée d'annuaire de l'autorité d'attribut détentrice.

```
aACertificate        ATTRIBUTE   ::=  {
    WITH SYNTAX                    AttributeCertificate
    EQUALITY MATCHING RULE         attributeCertificateExactMatch
    ID                             id-at-aACertificate }
```

17.2.3 Attribut "certificat de descripteur d'attribut"

L'attribut suivant contient des certificats d'attribut émis par une source d'autorité avec l'extension **attributeDescriptor**. Ces certificats d'attribut contiennent la spécification de la syntaxe valide et la règle de hiérarchie pour des attributs de privilège; ils sont stockés dans l'entrée d'annuaire de la source d'autorité émettrice.

```
attributeDescriptorCertificate      ATTRIBUTE  ::= {
    WITH SYNTAX                    AttributeCertificate
    EQUALITY MATCHING RULE         attributeCertificateExactMatch
    ID                             id-at-attributeDescriptorCertificate }
```

17.2.4 Attribut "liste de révocation de certificat d'attribut"

L'attribut suivant contient une liste de certificats d'attribut révoqués. Ces listes peuvent être stockées dans l'entrée d'annuaire de l'autorité émettrice ou dans une autre entrée d'annuaire (par exemple, de point de répartition).

```
attributeCertificateRevocationList     ATTRIBUTE ::= {
    WITH SYNTAX                    CertificateList
    EQUALITY MATCHING RULE         certificateListExactMatch
    ID                             id-at-attributeCertificateRevocationList}
```

17.2.5 Attribut "liste de révocation de certificat d'autorité d'attribut"

L'attribut suivant contient une liste de certificats d'attribut révoqués émis pour des autorités d'attribut. Ces listes peuvent être stockées dans l'entrée d'annuaire de l'autorité émettrice ou dans une autre entrée d'annuaire (par exemple, de point de répartition).

```
attributeAuthorityRevocationList    ATTRIBUTE   ::= {
    WITH SYNTAX                     CertificateList
    EQUALITY MATCHING RULE          certificateListExactMatch
    ID                              id-at-attributeAuthorityRevocationList }
```

17.2.6 Attribut "itinéraire de délégation"

L'attribut "itinéraire de délégation" contient des itinéraires de délégation, constitués chacun d'une succession de certificats d'attribut.

```
delegationPath      ATTRIBUTE   ::= {
    WITH SYNTAX                 AttCertPath
    ID                          id-at-delegationPath }

AttCertPath     ::=   SEQUENCE OF AttributeCertificate
```

Cet attribut peut être stocké dans l'entrée d'annuaire de l'autorité d'attribut et contenir certains itinéraires de délégation partant de cette autorité d'attribut vers d'autres autorités d'attribut. Son utilisation éventuelle permet une extraction plus rapide des certificats d'attribut délégués qui constituent les itinéraires de délégation les plus utilisés. Il n'existe pas de prescriptions propres à cet attribut et l'ensemble des valeurs qu'il stocke ne représentera probablement pas l'ensemble complet des itinéraires de délégation pour toute autorité d'attribut donnée.

17.2.7 Attribut "politique de privilège"

L'attribut "politique de privilège" contient des informations concernant les politiques de privilège.

```
privPolicy ATTRIBUTE    ::= {
    WITH SYNTAX         PolicySyntax
    ID                  id-at-privPolicy }
```

Le composant **policyIdentifier** contient l'identificateur d'objet enregistré pour une politique de privilège donnée.

Le composant **content** contient, s'il est présent, la totalité de l'énoncé de la politique de certificat.

Si le composant **pointer** est présent, le composant **name** fait alors référence à un ou plusieurs emplacements au niveau desquels peut être obtenue une copie de la politique de privilège. Le composant **hash** contient, s'il est présent, un hachage du contenu de politique de privilège qui doit se trouver à l'emplacement de référence. Ce hachage peut être utilisé pour effectuer une vérification d'intégrité du document de référence.

17.3 Règles de concordance de répertoire d'infrastructure PMI

Ce paragraphe définit les règles de concordance pour les attributs d'annuaire d'infrastructure PMI.

17.3.1 Concordance exacte de certificat d'attribut

La règle de concordance exacte de certificat d'attribut compare une valeur présentée avec une valeur d'attribut du type **AttributeCertificate**.

```
attributeCertificateExactMatch MATCHING-RULE  ::= {
    SYNTAX          AttributeCertificateExactAssertion
    ID              id-mr-attributeCertificateExactMatch }

AttributeCertificateExactAssertion      ::=   SEQUENCE {
    serialNumber                CertificateSerialNumber OPTIONAL,
    issuer                      IssuerSerial }
```

Cette règle de concordance renvoie la valeur "Vrai" si les composants de la valeur d'attribut concordent avec la valeur d'attribut présentée.

17.3.2 Concordance de certificat d'attribut

La règle de concordance de certificat d'attribut compare une valeur présentée avec une valeur d'attribut du type **AttributeCertificate**. Cette règle de concordance permet de rechercher une concordance plus complexe que la règle de concordance exacte de certificat.

```
attributeCertificateMatch  MATCHING-RULE  ::=  {
    SYNTAX          AttributeCertificateAssertion
    ID              id-mr-attributeCertificateMatch  }

AttributeCertificateAssertion  ::=  SEQUENCE  {
    holder          [0]   CHOICE {
                              baseCertificateID      [0]  IssuerSerial,
                              holderName             [1]  GeneralNames} OPTIONAL,
    issuer          [1]   GeneralNames OPTIONAL,
    attCertValidity [2]   GeneralizedTime OPTIONAL,
    attType         [3]   SET OF AttributeType OPTIONAL}
-- L'un au moins des composants de la séquence doit être présent
```

La règle de concordance renvoie la valeur "Vrai" si tous les composants figurant dans la valeur présentée concordent de la manière suivante avec les composants adéquats de la valeur de l'attribut:

- le composant **baseCertificateID** est en concordance s'il est égal au composant **IssuerSerial** [*série de l'émetteur*] de la valeur de l'attribut stocké;
- le composant **holderName** est en concordance si la valeur de l'attribut stocké contient l'extension de nom avec le même type de nom que celui qui est indiqué dans la valeur présentée;
- le composant **issuer** est en concordance si la valeur de l'attribut stocké contient le composant de nom du même type de nom que celui qui est indiqué dans la valeur présentée;
- le composant **attCertValidity** [*validité du certificat d'attribut*] est en concordance si sa valeur appartient à la durée de validité de la valeur de l'attribut stocké; et
- pour chaque composant **attType** [*type d'attribut*] dans la valeur présentée, un attribut de ce type figure dans le composant **attributes** de la valeur stockée.

17.3.3 Concordance détenteur/émetteur

La règle de concordance détenteur/émetteur compare une valeur présentée des composants détenteur et/ou émetteur avec une valeur du type **AttributeCertificate**.

```
holderIssuerMatch MATCHING-RULE  ::= {
    SYNTAX          HolderIssuerAssertion
    ID              id-mr-holderIssuerMatch }

HolderIssuerAssertion  ::=  SEQUENCE {
    holder          [0]     Holder          OPTIONAL,
    issuer          [1]     AttCertIssuer   OPTIONAL }
```

Cette règle de concordance renvoie la valeur "Vrai" si tous les composants présents dans la valeur présentée concordent avec les composants adéquats de la valeur de l'attribut.

17.3.4 Concordance d'itinéraire de délégation

La règle de concordance **delegationPathMatch** [*concordance d'itinéraire de délégation*] compare une valeur présentée avec une valeur d'attribut du type **delegationPath** [*itinéraire de délégation*]. Un vérificateur de privilège peut utiliser cette règle de concordance pour choisir un itinéraire partant d'un certificat émis par sa source d'autorité et aboutissant à un certificat émis pour l'autorité d'attribut qui a émis le certificat de détenteur d'entité finale en cours de validation.

```
delegationPathMatch MATCHING-RULE  ::= {
    SYNTAX          DelMatchSyntax
    ID              id-mr-delegationPathMatch }

DelMatchSyntax     ::=  SEQUENCE {
    firstIssuer     AttCertIssuer,
    lastHolder      Holder }
```

Cette règle de concordance renvoie la valeur "Vrai" si la valeur présentée dans le composant **firstIssuer** est en concordance avec les éléments correspondants du champ d'émetteur du premier certificat de la **SEQUENCE** dans la valeur stockée et si la valeur présentée dans le composant **lastHolder** [*dernier détenteur*] concorde avec les éléments correspondants du champ de détenteur du dernier certificat stocké dans la **SEQUENCE** de la valeur stockée. Cette règle de concordance renvoie la valeur "Faux" si la comparaison échoue.

SECTION 4 – UTILISATION DES CADRES DE CLÉ PUBLIQUE ET DE CERTIFICAT D'ATTRIBUT PAR L'ANNUAIRE

L'annuaire utilise le cadre de certificat de clé publique comme base pour un certain nombre de services de sécurité concernant l'authentification forte et la protection des opérations de l'annuaire ainsi que la protection des données stockées. L'annuaire utilise le cadre de certificat d'attribut comme base pour le schéma de contrôle d'accès basé sur des règles. La relation entre les éléments du cadre de certificat de clé publique et du cadre de certificat d'attribut est définie ici pour ce qui est des divers services de sécurité de l'annuaire. Les services de sécurité spécifiques fournis par l'annuaire sont entièrement spécifiés dans l'ensemble complet de spécifications d'annuaire.

18 Authentification de l'annuaire

L'annuaire prend en charge l'authentification des utilisateurs qui y accèdent au moyen d'agents utilisateurs d'annuaire DUA ainsi que l'authentification de systèmes d'annuaires (DSA) pour des utilisateurs et d'autres systèmes DSA. Les procédures devant être utilisées par l'authentification simple ou l'authentification forte dans l'annuaire sont décrites dans les paragraphes qui suivent.

18.1 Procédure d'authentification simple

Une authentification simple est conçue pour fournir une autorisation locale basée sur le nom distinctif d'un utilisateur, un mot de passe (optionnel) faisant l'objet d'un accord bilatéral associé à une entente bilatérale pour l'utilisation et le traitement de ce mot de passe au sein d'un domaine unique. L'utilisation de l'authentification simple est prévue principalement comme un moyen local, c'est-à-dire pour l'authentification d'entités homologues entre un agent DUA et un agent DSA ou entre un agent DSA et un autre agent DSA. L'authentification simple peut être réalisée de diverses manières:

a) transfert du nom distinctif de l'utilisateur et du mot de passe (optionnel) en clair (non protégé) vers le destinataire à des fins d'évaluation;

b) transfert du nom distinctif de l'utilisateur, du mot de passe et d'un nombre aléatoire et/ou d'un horodatage, l'ensemble étant protégé par l'application d'une fonction non réversible;

c) transfert des informations protégées décrites en b) ainsi que d'un nombre aléatoire et/ou d'un horodatage, l'ensemble étant protégé par l'application d'une fonction non réversible.

NOTE 1 – Il n'existe aucune prescription pour que les fonctions non réversibles appliquées soient différentes.

NOTE 2 – La signalisation des procédures de protection des mots de passe peut faire l'objet d'une extension du document.

Un niveau minimal de sécurité est fourni contre un accès non autorisé lorsque les mots de passe ne sont pas protégés. Ce procédé ne doit pas être considéré comme une base pour des services fiables. La protection du nom distinctif de l'utilisateur et du mot de passe fournit un niveau de sécurité supérieur. Les algorithmes utilisés pour la protection sont en général des fonctions non réversibles sans chiffrement dont l'implémentation est très simple.

La Figure 5 présente la procédure de fourniture de l'authentification simple.

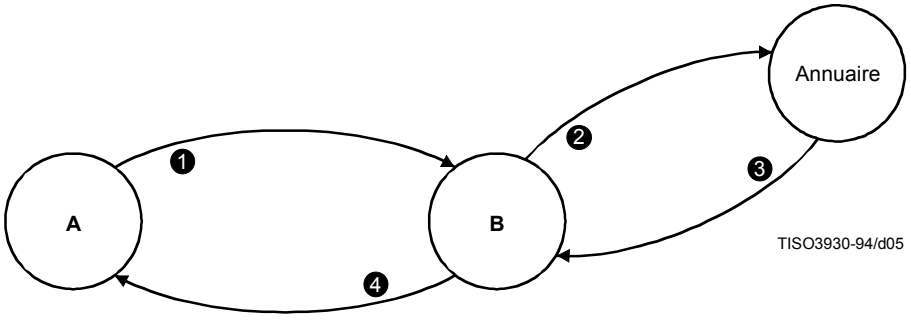


Figure 5 – Procédure d'authentification simple non protégée

Les étapes concernées sont les suivantes:

1) un utilisateur d'origine A émet son nom distinctif et son mot de passe vers un utilisateur destinataire B;

2) l'utilisateur B émet le nom distinctif et le mot de passe soumis par l'utilisateur A à destination de l'annuaire, qui compare le mot de passe par rapport à l'attribut **UserPassword** "mot de passe d'utilisateur" dans l'entrée d'annuaire de l'utilisateur A (en utilisant l'opération de comparaison de l'annuaire);

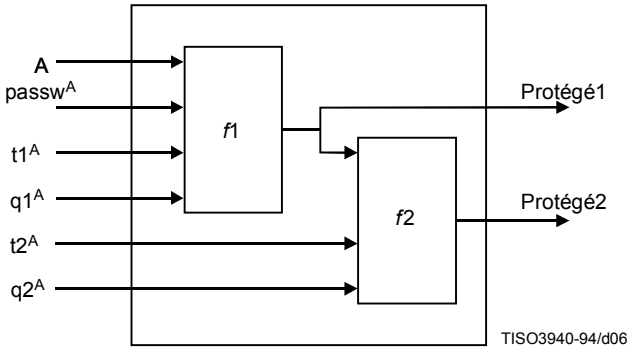
3) L'annuaire fait part à l'utilisateur B de la confirmation ou du rejet de la validité de ses justificatifs;

4) l'indication de réussite (ou d'échec) de l'authentification peut être transmise à l'utilisateur A.

La forme de base de l'authentification simple implique uniquement l'étape 1) et peut inclure l'étape 4) une fois que l'utilisateur B a vérifié le nom distinctif et le mot de passe.

18.1.1 Générations d'informations d'identification protégées

La Figure 6 présente deux démarches permettant de générer des informations d'identification protégées. Les fonctions *f*1 et *f*2 sont des fonctions non réversibles (pouvant être identiques ou différentes); les horodatages et les nombres aléatoires sont optionnels et font l'objet d'accords bilatéraux.



A	Nom distinctif de l'utilisateur
t^A	Horodatages
$passw^A$	Mot de passe de A
q^A	Numéros aléatoires contenant un compteur (optionnel)

Figure 6 – Authentification simple protégée

18.1.2 Procédure d'authentification simple protégée

La Figure 7 présente la procédure d'authentification simple protégée.

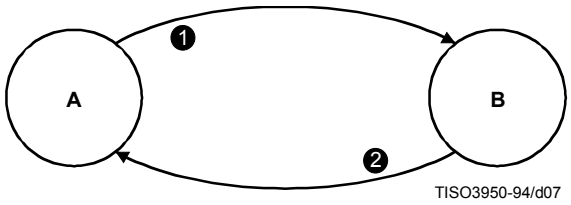


Figure 7 – Procédure d'authentification simple protégée

Les étapes concernées sont les suivantes (utilisant au départ uniquement la fonction *f*1):

1) Un utilisateur d'origine A émet ses informations d'identification protégées (Authenticator1) vers l'utilisateur B. La protection s'effectue en appliquant la fonction non réversible (*f*1) de la Figure 6, pour laquelle l'horodatage et/ou le nombre aléatoire (éventuel) sont utilisés afin de réduire les répétitions et dissimuler le mot de passe.

   La protection du mot de passe de A se fait de la manière suivante:

   Protected1 = *f*1 ($t1^A$, $q1^A$, A, $passw^A$)

   Les informations véhiculées vers B se présentent sous la forme suivante:

   Authenticator1 = $t1^A$, $q1^A$, A, Protected1

2) L'utilisateur B vérifie les informations d'identification protégées fournies par l'utilisateur A en générant (au moyen du nom distinctif, de l'horodatage optionnel et/ou du nombre aléatoire fournis par A, ainsi que d'une copie locale du mot de passe de A) une copie locale protégée du mot de passe de A (sous la forme Protected1). L'utilisateur B compare les informations d'identification soumises (Protected1) avec la valeur générée de manière locale.

3) L'utilisateur B fait part à l'utilisateur A de la confirmation ou du rejet des informations d'identification protégées.

La procédure peut être modifiée par l'utilisation des fonctions *f*1 et *f*2 pour fournir une meilleure protection. Les différences principales sont les suivantes:

1) L'utilisateur A émet ses informations d'identification protégées supplémentaires (Authenticator2) vers l'utilisateur B. La protection supplémentaire est réalisée en appliquant une autre fonction non réversible *f*2, comme indiqué par la Figure 6. La protection supplémentaire se présente sous la forme suivante:

   Protected2 = *f*2 ($t2^A$, $q2^A$, Protected1)

   Les informations véhiculées vers B se présentent sous la forme suivante:

   Authenticator2 = $t1^A$, $t2^A$, $q1^A$, $q2^A$, A, Protected2

Pour effectuer la comparaison, l'utilisateur B génère une valeur locale du mot de passe supplémentaire protégé de l'utilisateur A et la compare avec la valeur de Protected2.

2) L'utilisateur B fait part à l'utilisateur A de la confirmation ou du rejet des informations d'identification protégées.

NOTE – Les procédures définies dans ces paragraphes introduisent les utilisateurs A et B dans leur spécification. Lorsqu'elles s'appliquent à l'annuaire (spécifié dans la Rec. UIT-T X.511 | ISO/CEI 9594-3 et la Rec. X.518 | ISO/CEI 9594-4), l'utilisateur A peut être un agent DUA se liant à l'utilisateur B qui est un agent DSA; en variante, l'utilisateur A peut être un agent DSA se liant à l'utilisateur B qui est un autre agent DSA.

18.1.3 Type d'attribut "mot de passe utilisateur"

Un type d'attribut "mot de passe utilisateur" contient le mot de passe d'un objet. Une valeur d'attribut pour le mot de passe utilisateur est une chaîne d'octets spécifiée par l'objet suivant:

```
userPassword ATTRIBUTE  ::=  {
    WITH SYNTAX                 OCTET STRING (SIZE (0..ub-user-password))
    EQUALITY MATCHING RULE      octetStringMatch
    ID                          id-at-userPassword }
```

18.2 Authentification forte

Les procédures décrites dans ce paragraphe sont utilisées pour l'authentification forte entre un agent DUA et un agent DSA ainsi qu'entre deux agents DSA. Les procédures utilisent le cadre de certificat de clé publique défini dans la présente Spécification. Elles utilisent en outre l'annuaire comme référentiel pour les informations de clé publique nécessaires à l'authentification. La présence des paramètres pertinents dans des protocoles de l'annuaire est définie directement dans les spécifications de protocole. Les procédures définies ici pour l'authentification forte peuvent également être utilisées par des applications autres que l'annuaire qui utilisent également un tel référentiel. Lorsque ces procédures sont utilisées pour l'annuaire, le terme "utilisateur" peut désigner un agent DUA ou un agent DSA.

La démarche mise en œuvre dans la présente Spécification d'annuaire pour l'authentification forte utilise les propriétés d'une famille des systèmes de chiffrement appelés systèmes de chiffrement avec clé publique (PKCS, *public-key cryptosystems*). Ces systèmes de chiffrement qualifiés également d'asymétriques utilisent une paire de clés privées et publiques. L'Annexe E fournit une introduction succincte traitant de ces systèmes de chiffrement et indique les propriétés qui les rendent utilisables du point de vue de l'authentification. Pour qu'un système PKCS soit utilisable dans ce cadre

d'authentification sous sa forme actuelle, il doit avoir la propriété que les clés de la paire permettent toutes deux d'effectuer le chiffrement; la clé privée étant utilisée pour le déchiffrement lorsque la clé publique a été utilisée pour le chiffrement et la clé publique étant utilisée pour le déchiffrement lorsque la clé privée a été utilisée pour le chiffrement. En d'autres termes, elles satisfont à la relation $X_p \bullet X_s = X_s \bullet X_p$, X_p/X_s étant les fonctions de chiffrement et de déchiffrement qui utilisent les clés publiques et privées de l'utilisateur X.

NOTE – Une extension future de la présente Spécification d'annuaire est possible pour la prise en charge d'autres systèmes PKCS ne nécessitant pas la propriété de permutation.

Ce cadre d'authentification n'impose pas l'utilisation d'un système de chiffrement particulier. Il est prévu que le cadre s'appliquera à tout système de chiffrement avec clé publique et prendra de ce fait en charge les modifications apportées aux méthodes utilisées en fonction des progrès futurs du chiffrement, des techniques mathématiques et des capacités de calcul. Toutefois, deux utilisateurs souhaitant s'authentifier prendront en charge le même algorithme de chiffrement pour effectuer correctement l'authentification. Il s'ensuit que, dans le contexte d'un ensemble d'applications en relation, le choix d'une fonction unique permettra de maximiser la taille de la communauté des utilisateurs susceptibles de s'authentifier et de communiquer d'une manière fiable.

L'authentification repose sur le fait que chaque utilisateur possède un nom distinctif unique. L'attribution des noms distinctifs est de la responsabilité des autorités de dénomination. Chaque utilisateur doit en conséquence se fier au fait que les autorités de dénomination ne fournissent pas de noms distinctifs dupliqués.

Tout utilisateur est identifié par sa clé privée. Un autre utilisateur est en mesure de déterminer si un partenaire de communication est en possession de la clé privée et peut utiliser cette information pour confirmer que le partenaire en communication est effectivement l'utilisateur concerné. La validité de cette confirmation est conditionnée par le fait que la clé privée de l'utilisateur n'est pas divulguée.

Pour déterminer si un partenaire de communication est en possession de la clé privé d'un autre utilisateur, le premier utilisateur doit lui-même être en possession de la clé publique du second. Il est simple d'obtenir la valeur de cette clé publique à partir de l'entrée d'annuaire de l'utilisateur, mais la vérification de l'exactitude de cette clé est plus problématique. Elle peut se faire d'un grand nombre de manières possibles: le paragraphe 18.2.1 décrit un processus permettant de vérifier la clé publique d'un utilisateur par une référence à l'annuaire. Ce processus nécessite l'existence, entre les utilisateurs voulant s'authentifier, d'une chaîne continue de points de confiance dans l'Annuaire. Une telle chaîne peut être construite en identifiant un point de confiance commun. Ce point de confiance commun sera lié à chacun des utilisateurs par une chaîne continue de points de confiance.

18.2.1 Obtention de certificats de clé publique à partir de l'annuaire

Les certificats sont stockés dans des entrées d'annuaire sous la forme d'attributs des types **UserCertificate**, **CACertificate** et **CrossCertificatePair**. Ces types d'attributs sont connus de l'annuaire. Ils peuvent être utilisés au moyen du même protocole que les autres attributs. La définition de ces types se trouve en 3.3; leur spécification est donnée 11.2.

Avant que les utilisateurs puissent s'authentifier mutuellement, l'annuaire fournira dans le cas général la totalité des itinéraires de certification directs et en retour. Il est toutefois possible, dans la pratique, de réduire de la manière suivante le volume des informations qui seront extraites de l'annuaire pour une instance donnée d'authentification:

a) l'itinéraire de certification devient trivial si les deux utilisateurs souhaitant s'authentifier sont desservis par la même autorité de certification; ils peuvent dans ce cas ouvrir directement leurs certificats mutuels;

b) un utilisateur peut stocker les clés publiques, les certificats directs et les certificats en retour de toutes les autorités de certification se trouvant entre lui-même et la racine de l'arbre DIT si les autorités de certification des utilisateurs forment une hiérarchie. Ceci impliquera en général que l'utilisateur aura uniquement connaissance des clés publiques et des certificats de trois ou quatre autorités de certification. L'utilisateur aura alors uniquement besoin d'obtenir les itinéraires de certification à partir du point de confiance commun.

c) si un utilisateur communique fréquemment avec des utilisateurs certifiés par une autorité de certification donnée, il peut alors mémoriser l'itinéraire de certification vers cette autorité et l'itinéraire de retour correspondant et aura de ce fait besoin d'obtenir uniquement le certificat de l'autre utilisateur à partir de l'annuaire;

d) des autorités de certification peuvent se certifier mutuellement par un accord bilatéral, ce qui a pour effet de raccourcir l'itinéraire de certification;

e) deux utilisateurs peuvent s'authentifier sans recourir à l'annuaire s'ils ont communiqué précédemment et mémorisé leurs certificats mutuels.

Dans tous les cas, les utilisateurs vérifieront la validité des certificats une fois qu'ils ont pris connaissance de leurs certificats mutuels par le biais de l'itinéraire de certification.

18.2.1.1 Exemple

La Figure 8 présente un exemple fictif de fragment d'un arbre DIT dans lequel les autorités de certification forment une hiérarchie. Nous ferons l'hypothèse, en plus de l'existence des informations indiquées au niveau des autorités de certification, que chaque utilisateur connaît la clé publique de son autorité de certification ainsi que ses propres clés publiques et privées.

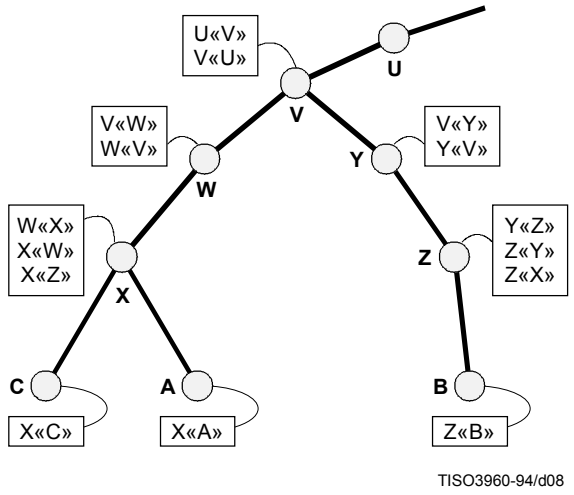


Figure 8 – Exemple de hiérarchie d'autorités de certification

Si les autorités de certification des utilisateurs constituent une hiérarchie, l'utilisateur A peut obtenir de l'annuaire les certificats suivant pour établir un itinéraire de certification vers B:

X«W», W«V», V«Y», Y«Z», Z«B»

Une fois que l'utilisateur A a obtenu ces certificats, il peut les ouvrir successivement sur l'itinéraire de certification, ce qui fournit le contenu du certificat de l'utilisateur B, y compris sa clé publique Bp:

Bp = Xp • X«W» W«V» V«Y» Y«Z» Z«B»

L'utilisateur A doit en général obtenir également de l'annuaire les certificats suivants pour établir l'itinéraire de certification en retour de B vers A:

Z«Y», Y«V», V«W», W«X», X«A»

Lorsqu'il reçoit ces certificats de l'utilisateur A, l'utilisateur B peut les ouvrir successivement sur l'itinéraire de certification en retour, ce qui fournit le contenu du certificat de l'utilisateur A, y compris sa clé publique Ap:

Ap = Zp • Z«Y» Y«V» V«W» W«X» X«A»

Les optimisations décrites au 18.2.1 s'appliquent comme suit:

a) les utilisateurs A et C, par exemple, connaissent tous deux la clé Xp de sorte que A peut obtenir directement le certificat de C. L'ouverture des certificats de l'itinéraire de certification se réduit à:

   Cp = Xp • X«C»

   et l'ouverture des certificats de l'itinéraire de certification en retour se réduit à:

   Ap = Xp • X«A»

b) si l'on fait l'hypothèse que l'utilisateur A connaît de ce fait W«X», Wp, V«W», Vp, U«V», Up, etc., les informations qu'il doit extraire de l'annuaire pour constituer l'itinéraire de certification se réduisent à:

   V«Y», Y«Z», Z«B»

   et les informations que l'utilisateur A doit extraire de l'annuaire pour constituer l'itinéraire de certification en retour se réduisent à:

   Z«Y», Y«V»

c) si l'on fait l'hypothèse que l'utilisateur A communique fréquemment avec des utilisateurs certifiés par l'autorité Z, il peut alors mémoriser les certificats V«Y», Y«V», Y«Z» et Z«Y» (en plus des clés mémorisées dans l'alinéa b) ci-dessus). Il aura uniquement besoin d'extraire Z«B» de l'annuaire pour communiquer avec l'utilisateur B.

d) si on fait l'hypothèse que des utilisateurs certifiés par les autorités X et Z communiquent fréquemment, le certificat X«Z» sera contenu dans l'entrée d'annuaire de l'autorité X et vice versa (comme indiqué par la Figure 8). Si l'utilisateur A veut s'authentifier pour l'utilisateur B, il aura alors uniquement besoin d'obtenir:

X«Z», Z«B»

pour constituer l'itinéraire de certification, et:

Z«X»

pour constituer l'itinéraire de certification en retour.

e) si on fait l'hypothèse que les utilisateurs A et C ont communiqué précédemment et ont mémorisé leurs certificats respectifs, ils peuvent alors utiliser directement leurs clés publiques respectives, c'est-à-dire:

$Cp = Xp \bullet X«C»$

et

$Ap = Xp \bullet X«A»$

Dans le cas général, les autorités de certification n'ont pas de relation hiérarchique. Dans l'exemple fictif de la Figure 9, on fait l'hypothèse qu'un utilisateur D certifié par l'autorité U souhaite s'authentifier pour l'utilisateur E certifié par l'autorité W. L'entrée d'annuaire de l'utilisateur D contiendra le certificat U«D» et l'entrée de l'utilisateur E contiendra le certificat W«E».

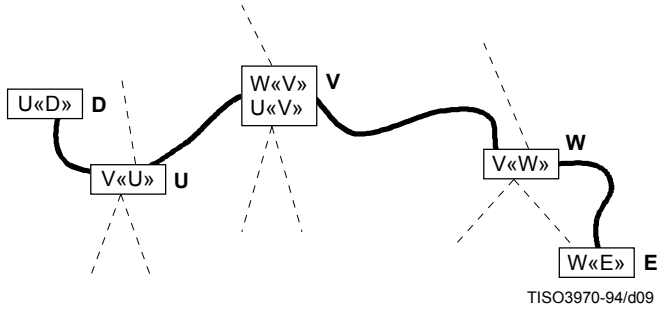


Figure 9 – Exemple d'itinéraire de certification non hiérarchique

Soit V une autorité de certification avec laquelle les autorités de certification U et W ont échangé précédemment des clés publiques de manière fiable. Il en résulte que les certificats U«V», V«U», W«V» et V«W» ont été générés et stockés dans l'annuaire. Supposons que les certificats U«V» et W«V» sont stockés dans l'entrée de l'autorité V, que le certificat V«U» est stocké dans l'entrée de l'autorité U et que le certificat V«W» est stocké dans l'entrée de l'autorité W.

L'utilisateur D doit trouver un itinéraire de certification vers l'utilisateur E. Il peut mettre en œuvre diverses stratégies. L'une d'elles consiste à considérer les utilisateurs et autorités de certification comme des nœuds et les certificats comme des arcs d'un graphe orienté. Sous cet aspect, l'utilisateur D doit alors faire une recherche dans le graphe pour trouver un itinéraire allant de U vers E, l'un de ces itinéraires étant U«V», V«W», W«E». Il est possible de construire également l'itinéraire inverse W«V», V«U», U«D» une fois que le premier a été établi.

18.2.2 Procédures d'authentification forte

La démarche de base de l'authentification a été esquissée ci-dessus, à savoir la confirmation d'une identité en prouvant la possession d'une clé privée. Diverses procédures d'authentification sont toutefois utilisables pour cette démarche. Il est en général du ressort d'une application donnée de déterminer les procédures adéquates de manière à satisfaire à sa propre politique de sécurité. Ce paragraphe décrit trois procédures d'authentification particulières pouvant être d'une utilité pour un certain domaine d'applications.

NOTE – La présente Spécification d'annuaire ne traite pas des procédures avec le niveau de détail nécessaire à leur implémentation. Il est toutefois possible d'envisager à cet effet d'autres normes, propres à une application ou d'une utilisation générale.

Les trois procédures impliquent un nombre différent d'échanges d'informations d'authentification et fournissent en conséquence divers types de garantie à leurs participants. D'une manière spécifique:

a) l'authentification en un temps décrite au 18.2.2.1 implique un seul transfert d'informations d'un utilisateur (A) vers un utilisateur (B) et établit les faits suivants:

   – identité de l'utilisateur A et le fait que le jeton d'authentification a effectivement été généré par ce dernier;

   – identité de l'utilisateur B et le fait que le jeton d'authentification était effectivement destiné à ce dernier;

   – l'intégrité et "l'originalité" (la propriété d'avoir été émis une seule fois) du jeton d'authentification transféré.

   Ces dernières propriétés peuvent également être établies pour toutes les autres données présentes dans le transfert.

b) l'authentification en deux temps décrite au 18.2.2.2 implique en outre une réponse de B vers A. Elle établit en plus les faits suivants:

   – que le jeton figurant dans la réponse a effectivement été généré par l'utilisateur B et était destiné à être émis vers l'utilisateur A;

   – l'intégrité et l'originalité du jeton d'authentification émis dans la réponse;

   – (de manière optionnelle) le secret mutuel de parties des jetons.

c) l'autorisation en trois temps décrite au 18.2.2.3 implique en outre un nouveau transfert de A vers B. Il établit les mêmes faits que l'authentification en deux temps, mais ne nécessite pas la vérification par horodatage de l'association.

Dans chaque cas où une authentification forte doit être effectuée, l'utilisateur A doit obtenir la clé publique de l'utilisateur B et l'itinéraire de certification en retour de B vers A avant tout échange d'informations. Ceci peut impliquer l'accès à l'annuaire, comme décrit au 18.2. Tout accès de ce type n'est pas mentionné de nouveau dans la description des procédures qui suivent.

La vérification des horodatages, telle qu'elle est mentionnée dans les paragraphes qui suivent, s'applique uniquement, soit dans le cas d'horloges synchronisées utilisées dans un environnement local, soit dans le cas d'horloges synchronisées de manière logique dans le cadre d'accords bilatéraux. Il est recommandé que le temps universel coordonné soit utilisé dans les deux cas.

On fait l'hypothèse, pour chacune des trois procédures d'authentification décrites ci-dessous, que le participant A a vérifié la validité de tous les certificats de l'itinéraire de certification.

18.2.2.1 Authentification en un temps

Les étapes suivantes sont effectuées, comme indiqué par la Figure 10:

1) l'utilisateur A génère un nombre r^A sans répétition utilisé pour détecter les attaques par réutilisation et éviter les contrefaçons;

2) l'utilisateur A émet le message suivant pour l'utilisateur B:

$$BA, A\{t^A, r^A, B\}$$

dans lequel t^A représente un horodatage constitué d'une ou de deux dates: l'instant (optionnel) de génération du jeton et l'instant d'expiration. La forme suivante est utilisée en variante si l'authentification des données "sgnData" doit être fournie par la signature numérique:

$$BA, A\{t^A, r^A, B, sgnData\}$$

La forme suivante est utilisée dans des cas où des données véhiculées seront utilisées ultérieurement comme clé privée (ces données sont représentées par "encData"):

$$BA, A\{t^A, r^A, B, sgnData, Bp[encData]\}$$

L'utilisation des données "encData" comme clé privée implique qu'elles doivent être choisies avec soin, par exemple pour constituer une clé forte pour le système de chiffrement indiqué par le champ "sgnData" du jeton;

3) l'utilisateur B effectue les actions suivantes:

   a) obtenir la clé Ap à partir de l'itinéraire B→A, en vérifiant que le certificat de A n'est pas caduc;

   b) vérifier la signature, et en conséquence l'intégrité des informations signées;

   c) vérifier que l'utilisateur B est effectivement le destinataire prévu;

   d) vérifier que l'horodatage est "actuel";

   e) vérifier, de manière optionnelle, que le nombre r^A n'a pas été réutilisé. Ceci peut se faire, par exemple, si r^A contient une partie séquentielle dont l'unicité de la valeur est vérifiée par une implémentation locale.

      Le nombre r^A est valide jusqu'à la date d'expiration indiquée par l'instant t^A. Le nombre r^A est toujours accompagné d'une partie séquentielle qui indique que l'utilisateur A ne doit pas répéter le jeton pendant le laps de temps t^A, de sorte que la vérification du nombre r^A lui-même n'est pas requise.

      Il est en tout cas raisonnable que l'utilisateur B stocke pendant la duré t^A la partie séquentielle avec l'horodatage t^A en clair ainsi que la partie hachée du jeton.

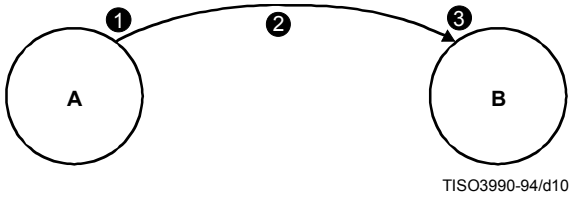


Figure 10 – Authentification en un temps

18.2.2.2 Authentification en deux temps

Les étapes suivantes sont effectuées, comme indiqué par la Figure 11:

1) comme pour le paragraphe 18.2.2.1;

2) comme pour le paragraphe 18.2.2.1;

3) comme pour le paragraphe 18.2.2.1;

4) L'utilisateur B génère un nombre r^B sans répétition, qui est utilisé à des fins similaires à celles de r^A;

5) L'utilisateur B émet le jeton d'authentification suivant pour l'utilisateur A:

$$B\{t^B, r^B, A, r^A\}$$

   dans lequel t^B est un horodatage défini de la même manière que pour t^A.

   La forme suivante est utilisée en variante si l'authentification des données "sgnData" doit être fournie par la signature numérique:

$$B\{t^B, r^B, A, r^A, sgnData\}$$

   La forme suivante est utilisée dans des cas où des données véhiculées seront utilisées ultérieurement comme clé privée (ces données sont représentées par "encData"):

$$B\{t^B, r^B, A, r^A, sgnData, Ap[encData]\}$$

   L'utilisation des données "encData" comme clé privée implique qu'elles doivent être choisies avec soin, par exemple pour constituer une clé forte pour le système de chiffrement indiqué par le champ "sgnData" du jeton.

6) L'utilisateur A effectue les actions suivantes:

   a) vérifier la signature, et en conséquence l'intégrité des informations signées;

   b) vérifier que l'utilisateur A est le destinataire prévu;

   c) vérifier que l'horodatage t^B est "actuel";

   d) vérifier, de manière optionnelle, que le nombre r^B n'a pas été réutilisé [(voir 18.2.2.1, étape 3), alinéa d)].

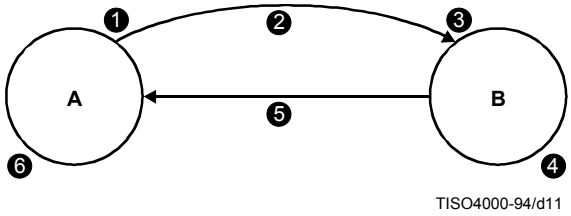


Figure 11 – Authentification en deux temps

18.2.2.3 Authentification en trois temps

Les étapes suivantes sont effectuées, comme indiqué par la Figure 12:

1) comme pour le paragraphe 18.2.2.2;
2) comme pour le paragraphe 18.2.2.2, l'horodatage t^A peut être nul;
3) comme pour le paragraphe 18.2.2.2, sauf que l'horodatage n'a pas besoin d'être vérifié;
4) comme pour le paragraphe 18.2.2.2;
5) comme pour le paragraphe 18.2.2.2, l'horodatage t^B peut être nul;
6) comme pour le paragraphe 18.2.2.2, sauf que l'horodatage n'a pas besoin d'être vérifié;
7) l'utilisateur A vérifie que le nombre r^A reçu est identique au nombre r^A émis;
8) l'utilisateur émet le jeton d'authentification suivant pour l'utilisateur B:

$$A\{r^B,B\}.$$

9) L'utilisateur B effectue les actions suivantes:
   a) vérifier la signature, et en conséquence l'intégrité des informations signées;
   b) vérifier que le nombre r^B reçu est identique au nombre r^B émis.

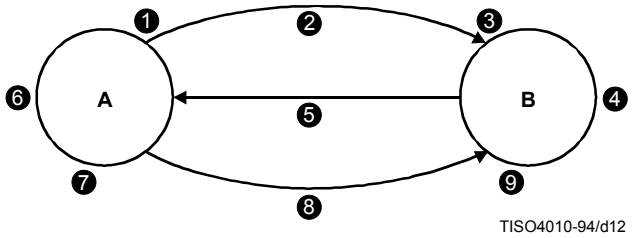


Figure 12 – Authentification en trois temps

19 Contrôle d'accès

L'annuaire existe dans un environnement dans lequel diverses autorités administratives contrôlent l'accès à leur partie de la base DIB. La définition d'un schéma de contrôle d'accès comprend des méthodes fournissant les fonctions suivantes:

- spécifier des informations de contrôle d'accès (ACI);
- faire respecter les droits d'accès définis par ces informations de contrôle d'accès;
- maintenir les informations de contrôle d'accès.

Le respect des droits d'accès s'applique au contrôle d'accès pour les informations suivantes:

- informations d'annuaire liées au nom;
- informations d'utilisateur d'annuaire;
- informations d'exploitation de l'annuaire, englobant les informations de contrôle d'accès.

Des autorités administratives peuvent utiliser tout ou partie de tout schéma de contrôle d'accès normalisé ou définir librement leurs propres schémas d'accès.

Le contrôle d'accès de base (BAC, *basic access control*) défini dans la Rec. UIT-T X.501 | ISO/CEI 9594-2 est un procédé de liste de contrôle d'accès permettant à des administrateurs d'annuaire de lier des permissions au niveau d'authentification effectué pour se lier à l'annuaire. Le cadre de certificat de clé publique défini dans la présente Spécification est utilisé pour fournir l'authentification forte utilisée pour cette liaison.

Le contrôle d'accès basé sur des règles (RBAC, *rules based access control*) défini dans la Rec. UIT-T X.501 | ISO/CEI 9594-2 utilise le cadre de certificat d'attribut défini dans la présente Spécification pour véhiculer des attributs d'habilitation utilisés pour prendre les décisions de contrôle d'accès. Le contrôle RBAC peut également être utilisé conjointement au contrôle BAC.

20 Protection des opérations d'annuaire

Le cadre de certificat de clé publique défini dans la présente Spécification est utilisé dans tous les protocoles d'Annuaire définis dans les Recommandations de cette série pour fournir une protection optionnelle des opérations englobant les demandes, les réponses et les erreurs. La protection de l'intégrité est fournie par la signature numérique de l'émetteur et la vérification de cette signature par le destinataire au moyen du certificat de clé publique de l'émetteur. La protection de la vie privée est fournie par l'utilisation d'un chiffrement avec clé publique dans lequel le contenu est chiffré au moyen de la clé publique fournie par le certificat de clé publique du destinataire souhaité et déchiffrée par ce dernier au moyen de la clé privée correspondante.

Les procédés et la syntaxe spécifique pour la demande et l'inclusion des éléments de protection dans les échanges de protocole sont définis par chacun des protocoles d'annuaire des Recommandations de cette série.

Annexe A

Cadres de certificats d'attribut et de clé publique

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

La présente annexe contient, sous la forme des trois modules ASN.1 **AuthenticationFramework** [*cadre d'authentification*], **CertificateExtensions** [*extensions de certificat*] et **AttributeCertificateDefinitions** [*définition de certificat d'attribut*], toutes les descriptions de type, de valeur et de classe d'objets d'informations utilisées dans la présente Spécification d'annuaire.

-- *A.1 Module du cadre d'authentification*

```
AuthenticationFramework {joint-iso-itu-t ds(5) module(1) authenticationFramework(7) 4}
DEFINITIONS ::=
BEGIN

-- EXPORTER TOUT --
-- Les types et les valeurs définis dans ce module sont exportés à des fins d'utilisation par d'autres
-- modules ASN.1 contenus au sein de Spécifications d'annuaire et par d'autres applications
-- qui les mettront en œuvre pour accéder à des services d'annuaire. D'autres applications peuvent les
-- utiliser à des fin propres, mais une telle utilisation n'imposera aucune contrainte aux extensions ou
-- modifications nécessaires à la maintenance ou à l'évolution du service d'annuaire.

IMPORTS
    id-at, id-nf, id-oc, informationFramework, upperBounds, selectedAttributeTypes, basicAccessControl,
    certificateExtensions
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 4}

    Name, ATTRIBUTE, OBJECT-CLASS, NAME-FORM, top
        FROM InformationFramework informationFramework

    ub-user-password, ub-content
        FROM UpperBounds upperBounds

    UniqueIdentifier, octetStringMatch, DirectoryString, commonName
        FROM SelectedAttributeTypes selectedAttributeTypes

    certificateExactMatch, certificatePairExactMatch, certificateListExactMatch, KeyUsage, GeneralNames,

        CertificatePoliciesSyntax, algorithmIdentifierMatch, CertPolicyId
        FROM CertificateExtensions certificateExtensions ;
-- définition du certificat de clé publique --

Certificate                 ::=  SIGNED { SEQUENCE {
    version                 [0]  Version DEFAULT v1,
    serialNumber                 CertificateSerialNumber,
    signature                    AlgorithmIdentifier,
    issuer                       Name,
    validity                     Validity,
    subject                      Name,
    subjectPublicKeyInfo         SubjectPublicKeyInfo,
    issuerUniqueIdentifier  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                                 -- si ce composant est présent, la version doit être v2 ou v3
    subjectUniqueIdentifier [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                                 -- si ce composant est présent, la version doit être v2 or v3
    extensions              [3]  Extensions OPTIONAL
                                 -- si ce composant est présent, la version doit être v3 -- } }

Version                     ::=  INTEGER { v1(0), v2(1), v3(2) }

CertificateSerialNumber     ::=  INTEGER

AlgorithmIdentifier              ::=  SEQUENCE {
    algorithm               ALGORITHM.&id ({SupportedAlgorithms}),
    parameters              ALGORITHM.&Type ({SupportedAlgorithms}{ @algorithm}) OPTIONAL }
-- La définition de l'objet d'information suivant est différée dans l'attente éventuelle de déclarations
-- de profil ou de conformité d'implémentation de protocole. L'ensemble est nécessaire pour la spécification
-- du composant parameters du champ AlgorithmIdentifier.
```

```
SupportedAlgorithms     ALGORITHM ::=   { ... }

Validity                ::=   SEQUENCE {
    notBefore       Time,
    notAfter        Time }

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

Time  ::=  CHOICE {
    utcTime             UTCTime,
    generalizedTime     GeneralizedTime }

Extensions ::= SEQUENCE OF Extension
-- Lorsque l'ordre des extensions au sein de l'expression SEQUENCE est significatif, la spécification de ces
-- extensions individuelles contiendra les règles de signification de l'ordre dans lequel elles figurent

Extension ::= SEQUENCE {
    extnId          EXTENSION.&id ({ExtensionSet}),
    critical        BOOLEAN DEFAULT FALSE,
    extnValue       OCTET STRING
                    -- contient un codage DER d'une valeur du type &ExtnType
                    -- pour l'objet d'extension identité par extnId -- }

ExtensionSet EXTENSION      ::=   { ... }

EXTENSION ::= CLASS {
    &id         OBJECT IDENTIFIER UNIQUE,
    &ExtnType }
WITH SYNTAX {
    SYNTAX              &ExtnType
    IDENTIFIED BY       &id }

-- autres structures de certificat PKI

Certificates                ::=     SEQUENCE {
    userCertificate                 Certificate,
    certificationPath               ForwardCertificationPath OPTIONAL}

ForwardCertificationPath            ::=     SEQUENCE OF CrossCertificates

CrossCertificates           ::=     SET OF Certificate

CertificationPath           ::=     SEQUENCE {
    userCertificate                 Certificate,
    theCACertificates               SEQUENCE OF CertificatePair OPTIONAL}

CertificatePair         ::=     SEQUENCE {
    forward             [0]     Certificate OPTIONAL,
    reverse             [1]     Certificate OPTIONAL
                                -- l'un au moins des éléments de la paire sera présent -- }
    (WITH COMPONENTS {..., forward PRESENT} |
    WITH COMPONENTS {..., reverse PRESENT})


-- liste de révocation de certificat (CRL)

CertificateList         ::=     SIGNED { SEQUENCE {
    version                     Version OPTIONAL,
                            -- si ce composant est présent, la version doit être v2
    signature                   AlgorithmIdentifier,
    issuer                      Name,
    thisUpdate                  Time,
    nextUpdate                  Time OPTIONAL,
    revokedCertificates             SEQUENCE OF SEQUENCE {
        serialNumber                    CertificateSerialNumber,
        revocationDate                  Time,
        crlEntryExtensions              Extensions OPTIONAL } OPTIONAL,
    crlExtensions       [0]         Extensions OPTIONAL }}


-- classes d'objets d'information --
```

```
ALGORITHM ::= TYPE-IDENTIFIER
-- types paramétrés --
HASH {ToBeHashed}       ::=     SEQUENCE {
    algorithmIdentifier             AlgorithmIdentifier,
    hashValue                       BIT STRING ( CONSTRAINED BY {
            -- doit être le résultat de l'application d'une procédure de hachage aux octets du codage DER --
    -- d'une valeur de -- ToBeHashed } ) }

ENCRYPTED-HASH { ToBeSigned }       ::=     BIT STRING ( CONSTRAINED BY {
    -- doit être le résultat de l'application d'une procédure de hachage aux octets du codage DER (voir 6.1) --
    -- d'une valeur de -- ToBeSigned -- suivie de l'application d'une procédure de chiffrement à ces octets -- })

ENCRYPTED { ToBeEnciphered }        ::=     BIT STRING ( CONSTRAINED BY {
    -- doit être le résultat de l'application d'une procédure de chiffrement appliquée --
    -- aux octets du codage BER d'une valeur de -- ToBeEnciphered})

SIGNATURE { ToBeSigned }            ::=     SEQUENCE {
    algorithmIdentifier             AlgorithmIdentifier,
    encrypted                       ENCRYPTED-HASH { ToBeSigned }}

SIGNED { ToBeSigned }               ::=     SEQUENCE {
    toBeSigned                      ToBeSigned,
    COMPONENTS OF                   SIGNATURE { ToBeSigned }}


-- classes d'objets PKI --

pkiUser  OBJECT-CLASS       ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {userCertificate}
    ID              id-oc-pkiUser }

pkiCA OBJECT-CLASS ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {cACertificate |
                    certificateRevocationList |
                    authorityRevocationList |
                    crossCertificatePair }
    ID              id-oc-pkiCA }

cRLDistributionPoint        OBJECT-CLASS        ::= {
    SUBCLASS OF             { top }
    KIND                    structural
    MUST CONTAIN            { commonName }
    MAY CONTAIN             { certificateRevocationList |
                            authorityRevocationList |
                            deltaRevocationList }
    ID                      id-oc-cRLDistributionPoint }

cRLDistPtNameForm  NAME-FORM    ::= {
    NAMES                   cRLDistributionPoint
    WITH ATTRIBUTES         { commonName}
    ID                      id-nf-cRLDistPtNameForm }

deltaCRL    OBJECT-CLASS        ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {deltaRevocationList}
    ID                  id-oc-deltaCRL }

cpCps   OBJECT-CLASS        ::= {
    SUBCLASS OF         {top}
    KIND                auxiliary
    MAY CONTAIN         {certificatePolicy |
                        certificationPracticeStmt}
    ID                  id-oc-cpCps }
```

```
pkiCertPath        OBJECT-CLASS        ::= {
    SUBCLASS OF            {top}
    KIND                   auxiliary
    MAY CONTAIN            { pkiPath }
    ID                     id-oc-pkiCertPath }
```

-- attributs d'annuaire PKI --

```
userCertificate                ATTRIBUTE  ::=        {
    WITH SYNTAX                           Certificate
    EQUALITY MATCHING RULE                certificateExactMatch
    ID                                    id-at-userCertificate}

cACertificate                  ATTRIBUTE  ::=        {
    WITH SYNTAX                           Certificate
    EQUALITY MATCHING RULE                certificateExactMatch
    ID                                    id-at-cAcertificate }

crossCertificatePair           ATTRIBUTE  ::=        {
    WITH SYNTAX                           CertificatePair
    EQUALITY MATCHING RULE                certificatePairExactMatch
    ID                                    id-at-crossCertificatePair }

certificateRevocationList      ATTRIBUTE  ::=        {
    WITH SYNTAX                           CertificateList
    EQUALITY MATCHING RULE                certificateListExactMatch
    ID                                    id-at-certificateRevocationList }

authorityRevocationList        ATTRIBUTE  ::=        {
    WITH SYNTAX                           CertificateList
    EQUALITY MATCHING RULE                certificateListExactMatch
    ID                                    id-at-authorityRevocationList }

deltaRevocationList ATTRIBUTE ::= {
    WITH SYNTAX                           CertificateList
    EQUALITY MATCHING RULE                certificateListExactMatch
    ID                                    id-at-deltaRevocationList }

supportedAlgorithms  ATTRIBUTE ::= {
    WITH SYNTAX                           SupportedAlgorithm
    EQUALITY MATCHING RULE                algorithmIdentifierMatch
    ID                                    id-at-supportedAlgorithms }

SupportedAlgorithm ::= SEQUENCE {
    algorithmIdentifier                           AlgorithmIdentifier,
    intendedUsage                         [0]     KeyUsage OPTIONAL,
    intendedCertificatePolicies           [1]     CertificatePoliciesSyntax OPTIONAL }

certificationPracticeStmt    ATTRIBUTE  ::=  {
    WITH SYNTAX              InfoSyntax
    ID                       id-at-certificationPracticeStmt }

InfoSyntax    ::=  CHOICE {
    content          DirectoryString {ub-content},
    pointer          SEQUENCE {
        name                 GeneralNames,
        hash                 HASH { HashedPolicyInfo } OPTIONAL } }

POLICY  ::=  TYPE-IDENTIFIER

HashedPolicyInfo       ::=   POLICY.&Type( {Policies} )
Policies POLICY ::= {...} -- défini par les réalisateurs --

certificatePolicy   ATTRIBUTE  ::=  {
    WITH SYNTAX       PolicySyntax
    ID                id-at-certificatePolicy }

PolicySyntax ::=  SEQUENCE {
    policyIdentifier      PolicyID,
    policySyntax          InfoSyntax
    }

PolicyID       ::=  CertPolicyId
```

```
pkiPath        ATTRIBUTE   ::= {
    WITH SYNTAX        PkiPath
    ID                 id-at-pkiPath }

PkiPath        ::=   SEQUENCE OF CrossCertificates

userPassword     ATTRIBUTE   ::=  {
    WITH SYNTAX                    OCTET STRING (SIZE (0..ub-user-password))
    EQUALITY MATCHING RULE         octetStringMatch
    ID                             id-at-userPassword }

-- attribution d'identificateur d'objet --
-- classes d'objets --

id-oc-cRLDistributionPoint          OBJECT IDENTIFIER ::=          {id-oc 19}
id-oc-pkiUser                       OBJECT IDENTIFIER ::=          {id-oc 21}
id-oc-pkiCA                         OBJECT IDENTIFIER ::=          {id-oc 22}
id-oc-deltaCRL                      OBJECT IDENTIFIER ::=          {id-oc 23}
id-oc-cpCps                         OBJECT IDENTIFIER ::=          {id-oc 30}
id-oc-pkiCertPath                   OBJECT IDENTIFIER ::=          {id-oc 31}

-- formes de nom --

id-nf-cRLDistPtNameForm             OBJECT IDENTIFIER ::=          {id-nf 14}


-- attributs d'annuaire --

id-at-userPassword                  OBJECT IDENTIFIER ::=          {id-at 35}
id-at-userCertificate               OBJECT IDENTIFIER ::=          {id-at 36}
id-at-cAcertificate                 OBJECT IDENTIFIER ::=          {id-at 37}
id-at-authorityRevocationList       OBJECT IDENTIFIER ::=          {id-at 38}
id-at-certificateRevocationList     OBJECT IDENTIFIER ::=          {id-at 39}
id-at-crossCertificatePair          OBJECT IDENTIFIER ::=          {id-at 40}
id-at-supportedAlgorithms           OBJECT IDENTIFIER ::=          {id-at 52}
id-at-deltaRevocationList           OBJECT IDENTIFIER ::=          {id-at 53}
id-at-certificationPracticeStmt     OBJECT IDENTIFIER ::=          {id-at 68}
id-at-certificatePolicy             OBJECT IDENTIFIER ::=          {id-at 69}
id-at-pkiPath                       OBJECT IDENTIFIER ::=          {id-at 70}

END
```

-- A.2 Module d'extension de certificat

```
CertificateExtensions {joint-iso-itu-t ds(5) module(1) certificateExtensions(26) 4}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTER TOUT --

IMPORTS
    id-at, id-ce, id-mr, informationFramework, authenticationFramework,
        selectedAttributeTypes, upperBounds
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
        usefulDefinitions(0) 4}

    Name, RelativeDistinguishedName, ATTRIBUTE, Attribute, MATCHING-RULE
        FROM InformationFramework informationFramework

    CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
        EXTENSION, Time, PolicyID
        FROM AuthenticationFramework authenticationFramework

    DirectoryString
        FROM SelectedAttributeTypes selectedAttributeTypes

    ub-name
        FROM UpperBounds upperBounds

    ORAddress
        FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3)
        modules(0) mts-abstract-service(1) version-1999 (1) } ;
-- L'ordre de succession des composants d'une expression SEQUENCE OF dans cette Spécification
-- n'est pas significatif, sauf indication explicite contraire

-- Certificat de clé publique et extensions de liste CRL --

authorityKeyIdentifier EXTENSION ::= {
    SYNTAX              AuthorityKeyIdentifier
    IDENTIFIED BY       id-ce-authorityKeyIdentifier }

AuthorityKeyIdentifier ::= SEQUENCE {
    keyIdentifier               [0] KeyIdentifier               OPTIONAL,
    authorityCertIssuer         [1] GeneralNames                OPTIONAL,
    authorityCertSerialNumber   [2] CertificateSerialNumber     OPTIONAL }
    ( WITH COMPONENTS           {..., authorityCertIssuer PRESENT,
                    authorityCertSerialNumber PRESENT} |
    WITH COMPONENTS             {..., authorityCertIssuer ABSENT,
                    authorityCertSerialNumber ABSENT} )

KeyIdentifier ::= OCTET STRING

subjectKeyIdentifier EXTENSION ::= {
    SYNTAX              SubjectKeyIdentifier
    IDENTIFIED BY       id-ce-subjectKeyIdentifier }

SubjectKeyIdentifier ::= KeyIdentifier

keyUsage EXTENSION ::= {
    SYNTAX              KeyUsage
    IDENTIFIED BY       id-ce-keyUsage }

KeyUsage ::= BIT STRING {
    digitalSignature        (0),
    nonRepudiation          (1),
    keyEncipherment         (2),
    dataEncipherment        (3),
    keyAgreement            (4),
    keyCertSign             (5),
    cRLSign                 (6),
    encipherOnly            (7),
    decipherOnly            (8) }

extKeyUsage EXTENSION ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF KeyPurposeId
    IDENTIFIED BY       id-ce-extKeyUsage }
```

```
KeyPurposeId ::= OBJECT IDENTIFIER

privateKeyUsagePeriod EXTENSION ::= {
    SYNTAX              PrivateKeyUsagePeriod
    IDENTIFIED BY       id-ce-privateKeyUsagePeriod }

PrivateKeyUsagePeriod ::= SEQUENCE {
    notBefore       [0]   GeneralizedTime OPTIONAL,
    notAfter        [1]   GeneralizedTime OPTIONAL }
    ( WITH COMPONENTS {..., notBefore PRESENT} |
    WITH COMPONENTS     {..., notAfter PRESENT} )

certificatePolicies EXTENSION ::= {
    SYNTAX              CertificatePoliciesSyntax
    IDENTIFIED BY       id-ce-certificatePolicies }

CertificatePoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF PolicyInformation

PolicyInformation ::= SEQUENCE {
    policyIdentifier      CertPolicyId,
    policyQualifiers      SEQUENCE SIZE (1..MAX) OF
                              PolicyQualifierInfo OPTIONAL }

CertPolicyId ::= OBJECT IDENTIFIER

PolicyQualifierInfo ::= SEQUENCE {
    policyQualifierId         CERT-POLICY-QUALIFIER.&id
                              ({SupportedPolicyQualifiers}),
    qualifier                 CERT-POLICY-QUALIFIER.&Qualifier
                              ({SupportedPolicyQualifiers}{@policyQualifierId})
                              OPTIONAL }

SupportedPolicyQualifiers CERT-POLICY-QUALIFIER ::= { ... }

anyPolicy     OBJECT IDENTIFIER    ::=   { 2 5 29 32 0 }

CERT-POLICY-QUALIFIER ::= CLASS {
    &id             OBJECT IDENTIFIER UNIQUE,
    &Qualifier      OPTIONAL }
WITH SYNTAX {
    POLICY-QUALIFIER-ID   &id
    [QUALIFIER-TYPE   &Qualifier] }

policyMappings EXTENSION ::= {
    SYNTAX              PolicyMappingsSyntax
    IDENTIFIED BY       id-ce-policyMappings }

PolicyMappingsSyntax ::= SEQUENCE SIZE (1..MAX) OF SEQUENCE {
    issuerDomainPolicy          CertPolicyId,
    subjectDomainPolicy         CertPolicyId }

subjectAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-subjectAltName }

GeneralNames ::= SEQUENCE SIZE (1..MAX) OF GeneralName

GeneralName ::= CHOICE {
    otherName                   [0]   INSTANCE OF OTHER-NAME,
    rfc822Name                  [1]   IA5String,
    dNSName                     [2]   IA5String,
    x400Address                 [3]   ORAddress,
    directoryName               [4]   Name,
    ediPartyName                [5]   EDIPartyName,
    uniformResourceIdentifier   [6]   IA5String,
    iPAddress                   [7]   OCTET STRING,
    registeredID                [8]   OBJECT IDENTIFIER }

OTHER-NAME ::= TYPE-IDENTIFIER

EDIPartyName ::= SEQUENCE {
    nameAssigner      [0]   DirectoryString {ub-name} OPTIONAL,
    partyName         [1]   DirectoryString {ub-name} }
```

```
issuerAltName EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-issuerAltName }

subjectDirectoryAttributes EXTENSION ::= {
    SYNTAX              AttributesSyntax
    IDENTIFIED BY       id-ce-subjectDirectoryAttributes }

AttributesSyntax ::= SEQUENCE SIZE (1..MAX) OF Attribute

basicConstraints EXTENSION ::= {
    SYNTAX              BasicConstraintsSyntax
    IDENTIFIED BY       id-ce-basicConstraints }

BasicConstraintsSyntax ::= SEQUENCE {
    cA                      BOOLEAN DEFAULT FALSE,
    pathLenConstraint       INTEGER (0..MAX) OPTIONAL }

nameConstraints EXTENSION ::= {
    SYNTAX              NameConstraintsSyntax
    IDENTIFIED BY       id-ce-nameConstraints }

NameConstraintsSyntax ::= SEQUENCE {
    permittedSubtrees  [0]      GeneralSubtrees OPTIONAL,
    excludedSubtrees   [1]      GeneralSubtrees OPTIONAL }

GeneralSubtrees ::= SEQUENCE SIZE (1..MAX) OF GeneralSubtree

GeneralSubtree ::= SEQUENCE {
    base                        GeneralName,
    minimum             [0]     BaseDistance DEFAULT 0,
    maximum             [1]     BaseDistance OPTIONAL }

BaseDistance ::= INTEGER (0..MAX)

policyConstraints EXTENSION ::= {
    SYNTAX              PolicyConstraintsSyntax
    IDENTIFIED BY       id-ce-policyConstraints }

PolicyConstraintsSyntax ::= SEQUENCE {
    requireExplicitPolicy     [0] SkipCerts OPTIONAL,
    inhibitPolicyMapping      [1] SkipCerts OPTIONAL }

SkipCerts ::= INTEGER (0..MAX)

cRLNumber EXTENSION ::= {
    SYNTAX              CRLNumber
    IDENTIFIED BY       id-ce-cRLNumber }

CRLNumber ::= INTEGER (0..MAX)

reasonCode EXTENSION ::= {
    SYNTAX              CRLReason
    IDENTIFIED BY       id-ce-reasonCode }

CRLReason ::= ENUMERATED {
    unspecified                 (0),
    keyCompromise               (1),
    cACompromise                (2),
    affiliationChanged          (3),
    superseded                  (4),
    cessationOfOperation        (5),
    certificateHold             (6),
    removeFromCRL               (8),
    privilegeWithdrawn          (9),
    aaCompromise                (10) }

holdInstructionCode EXTENSION ::= {
    SYNTAX              HoldInstruction
    IDENTIFIED BY       id-ce-instructionCode }

HoldInstruction ::= OBJECT IDENTIFIER

invalidityDate EXTENSION ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-invalidityDate }
```

```
crlScope EXTENSION  ::= {
    SYNTAX              CRLScopeSyntax
    IDENTIFIED BY       id-ce-cRLScope }

CRLScopeSyntax  ::=   SEQUENCE SIZE (1..MAX) OF PerAuthorityScope

PerAuthorityScope ::= SEQUENCE {
    authorityName           [0]     GeneralName OPTIONAL,
    distributionPoint       [1]     DistributionPointName OPTIONAL,
    onlyContains            [2]     OnlyCertificateTypes OPTIONAL,
    onlySomeReasons         [4]     ReasonFlags OPTIONAL,
    serialNumberRange       [5]     NumberRange OPTIONAL,
    subjectKeyIdRange       [6]     NumberRange OPTIONAL,
    nameSubtrees            [7]     GeneralNames OPTIONAL,
    baseRevocationInfo      [9]     BaseRevocationInfo OPTIONAL
    }

OnlyCertificateTypes   ::= BIT STRING {
    user            (0),
    authority       (1),
    attribute       (2) }

NumberRange ::= SEQUENCE {
    startingNumber      [0]     INTEGER OPTIONAL,
    endingNumber        [1]     INTEGER OPTIONAL,
    modulus                     INTEGER OPTIONAL }

BaseRevocationInfo ::= SEQUENCE {
    cRLStreamIdentifier         [0]  CRLStreamIdentifier    OPTIONAL,
    cRLNumber                   [1]  CRLNumber,
    baseThisUpdate              [2]  GeneralizedTime }

statusReferrals EXTENSION  ::= {
    SYNTAX              StatusReferrals
    IDENTIFIED BY       id-ce-statusReferrals }

StatusReferrals ::= SEQUENCE SIZE (1..MAX) OF StatusReferral

StatusReferral  ::=  CHOICE {
    cRLReferral         [0]     CRLReferral,
    otherReferral       [1]     INSTANCE OF OTHER-REFERRAL}

CRLReferral ::= SEQUENCE  {
    issuer              [0]     GeneralName OPTIONAL,
    location            [1]     GeneralName OPTIONAL,
    deltaRefInfo        [2]     DeltaRefInfo OPTIONAL,
    cRLScope                    CRLScopeSyntax,
    lastUpdate          [3]     GeneralizedTime OPTIONAL,
    lastChangedCRL      [4]     GeneralizedTime OPTIONAL}

DeltaRefInfo  ::=  SEQUENCE {
    deltaLocation       GeneralName,
    lastDelta           GeneralizedTime OPTIONAL }

OTHER-REFERRAL ::= TYPE-IDENTIFIER

cRLStreamIdentifier EXTENSION       ::= {
    SYNTAX              CRLStreamIdentifier
    IDENTIFIED BY       id-ce-cRLStreamIdentifier }

CRLStreamIdentifier  ::=  INTEGER (0..MAX)

orderedList EXTENSION      ::= {
    SYNTAX              OrderedListSyntax
    IDENTIFIED BY       id-ce-orderedList }

OrderedListSyntax  ::= ENUMERATED {
ascSerialNum        (0),
ascRevDate          (1) }

deltaInfo EXTENSION  ::=   {
    SYNTAX              DeltaInformation
    IDENTIFIED BY       id-ce-deltaInfo }
```

```
DeltaInformation ::= SEQUENCE {
    deltaLocation       GeneralName,
    nextDelta           GeneralizedTime OPTIONAL }

cRLDistributionPoints EXTENSION ::= {
    SYNTAX              CRLDistPointsSyntax
    IDENTIFIED BY       id-ce-cRLDistributionPoints }

CRLDistPointsSyntax ::= SEQUENCE SIZE (1..MAX) OF DistributionPoint

DistributionPoint ::= SEQUENCE {
    distributionPoint   [0]     DistributionPointName OPTIONAL,
    reasons             [1]     ReasonFlags OPTIONAL,
    cRLIssuer           [2]     GeneralNames OPTIONAL }

DistributionPointName ::= CHOICE {
    fullName                        [0]     GeneralNames,
    nameRelativeToCRLIssuer         [1]     RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {
    unused                  (0),
    keyCompromise           (1),
    cACompromise            (2),
    affiliationChanged      (3),
    superseded              (4),
    cessationOfOperation    (5),
    certificateHold         (6),
    privilegeWithdrawn      (7),
    aACompromise            (8) }

issuingDistributionPoint EXTENSION ::= {
    SYNTAX              IssuingDistPointSyntax
    IDENTIFIED BY       id-ce-issuingDistributionPoint }

IssuingDistPointSyntax ::= SEQUENCE {
    distributionPoint               [0] DistributionPointName OPTIONAL,
    onlyContainsUserCerts           [1] BOOLEAN DEFAULT FALSE,
    onlyContainsAuthorityCerts      [2] BOOLEAN DEFAULT FALSE,
    onlySomeReasons                 [3] ReasonFlags OPTIONAL,
    indirectCRL                     [4] BOOLEAN DEFAULT FALSE,
    onlyContainsAttributeCerts      [5] BOOLEAN DEFAULT FALSE }

certificateIssuer EXTENSION ::= {
    SYNTAX              GeneralNames
    IDENTIFIED BY       id-ce-certificateIssuer }

deltaCRLIndicator EXTENSION ::= {
    SYNTAX              BaseCRLNumber
    IDENTIFIED BY       id-ce-deltaCRLIndicator }

BaseCRLNumber ::= CRLNumber

baseUpdateTime EXTENSION    ::= {
    SYNTAX              GeneralizedTime
    IDENTIFIED BY       id-ce-baseUpdateTime }

freshestCRL EXTENSION    ::= {
    SYNTAX              CRLDistPointsSyntax
    IDENTIFIED BY       id-ce-freshestCRL }

inhibitAnyPolicy   EXTENSION   ::= {
    SYNTAX          SkipCerts
IDENTIFIED BY       id-ce-inhibitAnyPolicy }
```

-- Règles de concordance PKI --

```
certificateExactMatch MATCHING-RULE ::= {
    SYNTAX          CertificateExactAssertion
    ID              id-mr-certificateExactMatch }

CertificateExactAssertion ::= SEQUENCE {
    serialNumber        CertificateSerialNumber,
    issuer              Name }
```

```
certificateMatch MATCHING-RULE ::= {
    SYNTAX          CertificateAssertion
    ID              id-mr-certificateMatch }

CertificateAssertion ::= SEQUENCE {
    serialNumber            [0] CertificateSerialNumber     OPTIONAL,
    issuer                  [1] Name                        OPTIONAL,
    subjectKeyIdentifier    [2] SubjectKeyIdentifier        OPTIONAL,
    authorityKeyIdentifier  [3] AuthorityKeyIdentifier      OPTIONAL,
    certificateValid        [4] Time                        OPTIONAL,
    privateKeyValid         [5] GeneralizedTime             OPTIONAL,
    subjectPublicKeyAlgID   [6] OBJECT IDENTIFIER           OPTIONAL,
    keyUsage                [7] KeyUsage                    OPTIONAL,
    subjectAltName          [8] AltNameType                 OPTIONAL,
    policy                  [9] CertPolicySet               OPTIONAL,
    pathToName              [10] Name                       OPTIONAL,
    subject                 [11] Name                       OPTIONAL,
    nameConstraints         [12] NameConstraintsSyntax      OPTIONAL }

AltNameType ::= CHOICE {
    builtinNameForm    ENUMERATED {
          rfc822Name                (1),
          dNSName                   (2),
          x400Address               (3),
          directoryName             (4),
          ediPartyName              (5),
          uniformResourceIdentifier (6),
          iPAddress                 (7),
          registeredId              (8) },
    otherNameForm      OBJECT IDENTIFIER }

CertPolicySet ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId

certificatePairExactMatch MATCHING-RULE    ::=   {
    SYNTAX          CertificatePairExactAssertion
    ID              id-mr-certificatePairExactMatch }

CertificatePairExactAssertion ::= SEQUENCE {
    issuedToThisCAAssertion         [0] CertificateExactAssertion OPTIONAL,
    issuedByThisCAAssertion         [1] CertificateExactAssertion OPTIONAL }
    ( WITH COMPONENTS               {..., issuedToThisCAAssertion PRESENT} |
      WITH COMPONENTS               {..., issuedByThisCAAssertion PRESENT} )

certificatePairMatch MATCHING-RULE ::={
    SYNTAX          CertificatePairAssertion
    ID              id-mr-certificatePairMatch }

CertificatePairAssertion ::= SEQUENCE {
    issuedToThisCAAssertion    [0] CertificateAssertion OPTIONAL,
    issuedByThisCAAssertion    [1] CertificateAssertion OPTIONAL }
    ( WITH COMPONENTS   {..., issuedToThisCAAssertion PRESENT} |
      WITH COMPONENTS   {..., issuedByThisCAAssertion PRESENT} )

certificateListExactMatch MATCHING-RULE    ::=   {
    SYNTAX          CertificateListExactAssertion
    ID              id-mr-certificateListExactMatch }

CertificateListExactAssertion ::= SEQUENCE {
    issuer              Name,
    thisUpdate          Time,
    distributionPoint   DistributionPointName OPTIONAL OPTIONAL }

certificateListMatch MATCHING-RULE ::= {
    SYNTAX          CertificateListAssertion
    ID              id-mr-certificateListMatch }

CertificateListAssertion ::= SEQUENCE {
    issuer                          Name                    OPTIONAL,
    minCRLNumber            [0]     CRLNumber               OPTIONAL,
    maxCRLNumber            [1]     CRLNumber               OPTIONAL,
    reasonFlags                     ReasonFlags             OPTIONAL,
    dateAndTime                     Time                    OPTIONAL,
    distributionPoint       [2]     DistributionPointName OPTIONAL,
    authorityKeyIdentifier  [3]     AuthorityKeyIdentifier OPTIONAL}
```

```
algorithmIdentifierMatch MATCHING-RULE     ::=  {
    SYNTAX          AlgorithmIdentifier
    ID              id-mr-algorithmIdentifierMatch }

policyMatch MATCHING-RULE   ::= {
    SYNTAX          PolicyID
    ID              id-mr-policyMatch }

pkiPathMatch MATCHING-RULE  ::= {
SYNTAX        PkiPathMatchSyntax
ID            id-mr-pkiPathMatch }
PkiPathMatchSyntax    ::= SEQUENCE {
firstIssuer           Name,
lastSubject           Name }
```

-- attributions d'identificateur d'objet --

```
id-ce-subjectDirectoryAttributes        OBJECT IDENTIFIER   ::=  {id-ce 9}
id-ce-subjectKeyIdentifier              OBJECT IDENTIFIER   ::=  {id-ce 14}
id-ce-keyUsage                          OBJECT IDENTIFIER   ::=  {id-ce 15}
id-ce-privateKeyUsagePeriod             OBJECT IDENTIFIER   ::=  {id-ce 16}
id-ce-subjectAltName                    OBJECT IDENTIFIER   ::=  {id-ce 17}
id-ce-issuerAltName                     OBJECT IDENTIFIER   ::=  {id-ce 18}
id-ce-basicConstraints                  OBJECT IDENTIFIER   ::=  {id-ce 19}
id-ce-cRLNumber                         OBJECT IDENTIFIER   ::=  {id-ce 20}
id-ce-reasonCode                        OBJECT IDENTIFIER   ::=  {id-ce 21}
id-ce-instructionCode                   OBJECT IDENTIFIER   ::=  {id-ce 23}
id-ce-invalidityDate                    OBJECT IDENTIFIER   ::=  {id-ce 24}
id-ce-deltaCRLIndicator                 OBJECT IDENTIFIER   ::=  {id-ce 27}
id-ce-issuingDistributionPoint          OBJECT IDENTIFIER   ::=  {id-ce 28}
id-ce-certificateIssuer                 OBJECT IDENTIFIER   ::=  {id-ce 29}
id-ce-nameConstraints                   OBJECT IDENTIFIER   ::=  {id-ce 30}
id-ce-cRLDistributionPoints             OBJECT IDENTIFIER   ::=  {id-ce 31}
id-ce-certificatePolicies               OBJECT IDENTIFIER   ::=  {id-ce 32}
id-ce-policyMappings                    OBJECT IDENTIFIER   ::=  {id-ce 33}
-- déconseillé                          OBJECT IDENTIFIER   ::=  {id-ce 34}
id-ce-authorityKeyIdentifier            OBJECT IDENTIFIER   ::=  {id-ce 35}
id-ce-policyConstraints                 OBJECT IDENTIFIER   ::=  {id-ce 36}
id-ce-extKeyUsage                       OBJECT IDENTIFIER   ::=  {id-ce 37}
id-ce-cRLStreamIdentifier               OBJECT IDENTIFIER ::=    {id-ce 40}
id-ce-cRLScope                          OBJECT IDENTIFIER ::=    {id-ce 44}
id-ce-statusReferrals                   OBJECT IDENTIFIER ::=    {id-ce 45}
id-ce-freshestCRL                       OBJECT IDENTIFIER ::=    {id-ce 46}
id-ce-orderedList                       OBJECT IDENTIFIER ::=    {id-ce 47}
id-ce-baseUpdateTime                    OBJECT IDENTIFIER ::=    {id-ce 51}
id-ce-deltaInfo                         OBJECT IDENTIFIER ::=    {id-ce 53}
id-ce-inhibitAnyPolicy                  OBJECT IDENTIFIER ::=    {id-ce 54}
```

-- règles de concordance d'identificateurs d'objet --

```
id-mr-certificateExactMatch             OBJECT IDENTIFIER   ::=  {id-mr 34}
id-mr-certificateMatch                  OBJECT IDENTIFIER   ::=  {id-mr 35}
id-mr-certificatePairExactMatch         OBJECT IDENTIFIER   ::=  {id-mr 36}
id-mr-certificatePairMatch              OBJECT IDENTIFIER   ::=  {id-mr 37}
id-mr-certificateListExactMatch         OBJECT IDENTIFIER   ::=  {id-mr 38}
id-mr-certificateListMatch              OBJECT IDENTIFIER   ::=  {id-mr 39}
id-mr-algorithmIdentifierMatch          OBJECT IDENTIFIER   ::=  {id-mr 40}
id-mr-policyMatch                       OBJECT IDENTIFIER ::=    {id-mr 60}
id-mr-pkiPathMatch                      OBJECT IDENTIFIER   ::=  {id-mr 62}
```

-- Les identificateurs d'objets suivants ne sont pas utilisés dans la présente Spécification:
-- {id-ce 2}, {id-ce 3}, {id-ce 4}, {id-ce 5}, {id-ce 6}, {id-ce 7},
-- {id-ce 8}, {id-ce 10}, {id-ce 11}, {id-ce 12}, {id-ce 13},
-- {id-ce 22}, {id-ce 25}, {id-ce 26}

END

-- A.3 Module de cadre de certificat d'attribut

```
AttributeCertificateDefinitions {joint-iso-itu-t ds(5) module(1) attributeCertificateDefinitions(32) 4}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTER TOUT --

IMPORTS
    id-at, id-ce, id-mr, informationFramework, authenticationFramework,
        selectedAttributeTypes, upperBounds, id-oc, certificateExtensions
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1)
        usefulDefinitions(0) 4}

    Name, RelativeDistinguishedName, ATTRIBUTE, Attribute,
        MATCHING-RULE, AttributeType, OBJECT-CLASS, top
        FROM InformationFramework informationFramework

    CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
        EXTENSION, SIGNED, InfoSyntax, PolicySyntax, Extensions, Certificate
        FROM AuthenticationFramework authenticationFramework

    DirectoryString, TimeSpecification, UniqueIdentifier
        FROM SelectedAttributeTypes selectedAttributeTypes

    GeneralName, GeneralNames, NameConstraintsSyntax, certificateListExactMatch
        FROM CertificateExtensions certificateExtensions

    ub-name
        FROM UpperBounds upperBounds

    UserNotice
        FROM PKIX1Implicit93 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5)
pkix(7) id-mod(0) id-pkix1-implicit-93(4)}

    ORAddress
        FROM MTSAbstractService {joint-iso-itu-t mhs(6) mts(3)
        modules(0) mts-abstract-service(1) version-1999 (1) } ;
-- L'ordre de succession des composants d'une expression SEQUENCE OF dans cette Spécification
-- n'est pas significatif, sauf indication explicite contraire.


-- Structure de certificat d'attribut --

AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE
    {
    version                 AttCertVersion, --la  version est v2
    holder                  Holder,
    issuer                  AttCertIssuer,
    signature               AlgorithmIdentifier,
    serialNumber            CertificateSerialNumber,
    attrCertValidityPeriod  AttCertValidityPeriod,
    attributes              SEQUENCE OF Attribute,
    issuerUniqueID          UniqueIdentifier OPTIONAL,
    extensions              Extensions      OPTIONAL
    }

AttCertVersion     ::= INTEGER {v1(0), v2(1) }

Holder    ::=   SEQUENCE
          {
          baseCertificateID     [0] IssuerSerial            OPTIONAL,
               -- émetteur et numéro de série du détenteur du certificat de clé publique
          entityName            [1] GeneralNames            OPTIONAL,
               -- nom de l'entité ou du rôle
          objectDigestInfo      [2] ObjectDigestInfo        OPTIONAL
               -- utilisé pour authentifier directement le détenteur, par exemple un exécutable
-- au moins l'un des composants baseCertificateID, entityName ou objectDigestInfo doit être présent --}
```

```
ObjectDigestInfo   ::= SEQUENCE {
    digestedObjectType  ENUMERATED {
        publicKey           (0),
        publicKeyCert       (1),
        otherObjectTypes    (2) },
    otherObjectTypeID       OBJECT IDENTIFIER  OPTIONAL,
    digestAlgorithm         AlgorithmIdentifier,
    objectDigest            BIT STRING }

AttCertIssuer ::=   [0]     SEQUENCE {
    issuerName                      GeneralNames  OPTIONAL,
    baseCertificateID       [0]     IssuerSerial  OPTIONAL,
    objectDigestInfo        [1]     ObjectDigestInfo  OPTIONAL }

-- Au moins l'un des composants doit être présent
    ( WITH COMPONENTS { ..., issuerName  PRESENT } |
      WITH COMPONENTS { ..., baseCertificateID  PRESENT } |
      WITH COMPONENTS { ..., objectDigestInfo PRESENT } )

IssuerSerial  ::=  SEQUENCE {
    issuer          GeneralNames,
    serial          CertificateSerialNumber,
    issuerUID       UniqueIdentifier OPTIONAL }

AttCertValidityPeriod  ::= SEQUENCE {
    notBeforeTime       GeneralizedTime,
    notAfterTime        GeneralizedTime }

AttributeCertificationPath  ::=  SEQUENCE {
    attributeCertificate            AttributeCertificate,
    acPath                          SEQUENCE OF ACPathData OPTIONAL }

ACPathData  ::=  SEQUENCE {
    certificate           [0]  Certificate  OPTIONAL,
    attributeCertificate  [1]  AttributeCertificate  OPTIONAL }

PrivilegePolicy     ::=   OBJECT IDENTIFIER

-- attributs de privilège --

role    ATTRIBUTE::= {
        WITH SYNTAX         RoleSyntax
        ID                  id-at-role }

RoleSyntax    ::=   SEQUENCE {
    roleAuthority       [0]     GeneralNames OPTIONAL,
    roleName            [1]     GeneralName }

-- classes d'objets d'infrastructure PMI --

pmiUser OBJECT-CLASS ::= {
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {attributeCertificateAttribute}
    ID              id-oc-pmiUser
    }

pmiAA OBJECT-CLASS ::= {
 -- autorité d'attribut d'infrastructure PMI
    SUBCLASS OF     {top}
    KIND            auxiliary
    MAY CONTAIN     {aACertificate |
                     attributeCertificateRevocationList |
                     attributeAuthorityRevocationList}
    ID              id-oc-pmiAA
    }
```

```
pmiSOA OBJECT-CLASS ::= {  -- source d'autorité d'infrastructure PMI --
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      {attributeCertificateRevocationList |
                      attributeAuthorityRevocationList |
                      attributeDescriptorCertificate}
    ID               id-oc-pmiSOA
    }

attCertCRLDistributionPt   OBJECT-CLASS ::= {
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      { attributeCertificateRevocationList |
                      attributeAuthorityRevocationList }
    ID               id-oc-attCertCRLDistributionPts
    }

pmiDelegationPath          OBJECT-CLASS   ::= {
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      { delegationPath }
    ID               id-oc-pmiDelegationPath }

privilegePolicy            OBJECT-CLASS   ::= {
    SUBCLASS OF      {top}
    KIND             auxiliary
    MAY CONTAIN      {privPolicy }
    ID               id-oc-privilegePolicy }
```

-- attributs d'annuaire d'infrastructure PMI --

```
attributeCertificateAttribute          ATTRIBUTE  ::= {
    WITH SYNTAX                        AttributeCertificate
    EQUALITY MATCHING RULE             attributeCertificateExactMatch
    ID                                 id-at-attributeCertificate }

aACertificate                          ATTRIBUTE   ::=   {
    WITH SYNTAX                        AttributeCertificate
    EQUALITY MATCHING RULE             attributeCertificateExactMatch
    ID                                 id-at-aACertificate }

attributeDescriptorCertificate         ATTRIBUTE  ::= {
    WITH SYNTAX                        AttributeCertificate
    EQUALITY MATCHING RULE             attributeCertificateExactMatch
    ID                                 id-at-attributeDescriptorCertificate }

attributeCertificateRevocationList     ATTRIBUTE ::= {
    WITH SYNTAX                        CertificateList
    EQUALITY MATCHING RULE             certificateListExactMatch
    ID                                 id-at-attributeCertificateRevocationList}

attributeAuthorityRevocationList       ATTRIBUTE   ::= {
    WITH SYNTAX                        CertificateList
    EQUALITY MATCHING RULE             certificateListExactMatch
    ID                                 id-at-attributeAuthorityRevocationList }

delegationPath          ATTRIBUTE   ::= {
    WITH SYNTAX         AttCertPath
    ID                  id-at-delegationPath }

AttCertPath   ::=   SEQUENCE OF AttributeCertificate

privPolicy              ATTRIBUTE   ::= {
    WITH SYNTAX         PolicySyntax
    ID                  id-at-privPolicy }
```

-- Extensions et règles de concordance de certificat d'attribut --

```
attributeCertificateExactMatch MATCHING-RULE  ::= {
    SYNTAX          AttributeCertificateExactAssertion
    ID              id-mr-attributeCertificateExactMatch }
```

```
AttributeCertificateExactAssertion   ::=  SEQUENCE {
    serialNumber        CertificateSerialNumber OPTIONAL,
    issuer              IssuerSerial
    }

attributeCertificateMatch  MATCHING-RULE  ::=  {
    SYNTAX        AttributeCertificateAssertion
    ID            id-mr-attributeCertificateMatch  }

AttributeCertificateAssertion  ::=  SEQUENCE  {
    holder              [0]   CHOICE {
        baseCertificateID     [0]  IssuerSerial,
        holderName            [1]  GeneralNames} OPTIONAL,
    issuer              [1]   GeneralNames OPTIONAL,
    attCertValidity     [2]   GeneralizedTime OPTIONAL,
    attType             [3]   SET OF AttributeType OPTIONAL}
-- l'un au moins des composants de la séquence doit être présent

holderIssuerMatch MATCHING-RULE  ::= {
    SYNTAX        HolderIssuerAssertion
    ID            id-mr-holderIssuerMatch }

HolderIssuerAssertion ::=  SEQUENCE {
    holder        [0]       Holder    OPTIONAL,
    issuer        [1]       AttCertIssuer OPTIONAL
    }

delegationPathMatch MATCHING-RULE  ::= {
    SYNTAX        DelMatchSyntax
    ID            id-mr-delegationPathMatch }

DelMatchSyntax   ::=  SEQUENCE {
    firstIssuer   AttCertIssuer,
    lastHolder    Holder }

sOAIdentifier EXTENSION  ::= {
    SYNTAX            NULL
    IDENTIFIED BY     id-ce-sOAIdentifier }

authorityAttributeIdentifier EXTENSION  ::=
    {
    SYNTAX            AuthorityAttributeIdentifierSyntax
    IDENTIFIED BY     { id-ce-authorityAttributeIdentifier } }

AuthorityAttributeIdentifierSyntax    ::=  SEQUENCE SIZE (1..MAX) OF AuthAttId

AuthAttId     ::=  IssuerSerial

authAttIdMatch MATCHING-RULE  ::= {
    SYNTAX        AuthorityAttributeIdentifierSyntax
    ID            id-mr-authAttIdMatch }

roleSpecCertIdentifier EXTENSION ::=
    {
    SYNTAX            RoleSpecCertIdentifierSyntax
    IDENTIFIED BY     { id-ce-roleSpecCertIdentifier } }

RoleSpecCertIdentifierSyntax  ::= SEQUENCE SIZE (1..MAX) OF RoleSpecCertIdentifier

RoleSpecCertIdentifier  ::= SEQUENCE {
    roleName                  [0]     GeneralName,
    roleCertIssuer            [1]     GeneralName,
    roleCertSerialNumber      [2]     CertificateSerialNumber    OPTIONAL,
    roleCertLocator           [3]     GeneralNames               OPTIONAL }

roleSpecCertIdMatch MATCHING-RULE  ::= {
    SYNTAX        RoleSpecCertIdentifierSyntax
    ID            id-mr-roleSpecCertIdMatch }

basicAttConstraints EXTENSION ::=
    {
        SYNTAX            BasicAttConstraintsSyntax
        IDENTIFIED BY     { id-ce-basicAttConstraints }
    }
```

```
BasicAttConstraintsSyntax ::= SEQUENCE
    {
        authority               BOOLEAN DEFAULT FALSE,
        pathLenConstraint       INTEGER (0..MAX) OPTIONAL
    }

basicAttConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          BasicAttConstraintsSyntax
    ID              id-mr-basicAttConstraintsMatch }

delegatedNameConstraints EXTENSION ::= {
    SYNTAX              NameConstraintsSyntax
    IDENTIFIED BY       id-ce-delegatedNameConstraints }

delegatedNameConstraintsMatch MATCHING-RULE  ::= {
    SYNTAX          NameConstraintsSyntax
    ID              id-mr-delegatedNameConstraintsMatch}

timeSpecification EXTENSION  ::=  {
    SYNTAX              TimeSpecification
    IDENTIFIED BY       id-ce-timeSpecification }

timeSpecificationMatch MATCHING-RULE  ::= {
    SYNTAX          TimeSpecification
    ID              id-mr-timeSpecMatch }

acceptableCertPolicies EXTENSION ::= {
    SYNTAX              AcceptableCertPoliciesSyntax
    IDENTIFIED BY       id-ce-acceptableCertPolicies }

AcceptableCertPoliciesSyntax ::= SEQUENCE SIZE (1..MAX) OF CertPolicyId

CertPolicyId ::= OBJECT IDENTIFIER

acceptableCertPoliciesMatch MATCHING-RULE  ::= {
    SYNTAX          AcceptableCertPoliciesSyntax
    ID              id-mr-acceptableCertPoliciesMatch }

attributeDescriptor EXTENSION  ::= {
    SYNTAX              AttributeDescriptorSyntax
    IDENTIFIED BY       {id-ce-attributeDescriptor } }

AttributeDescriptorSyntax  ::= SEQUENCE {
    identifier                      AttributeIdentifier,
    attributeSyntax                 OCTET STRING (SIZE(1..MAX)),
    name                [0]         AttributeName   OPTIONAL,
    description         [1]         AttributeDescription    OPTIONAL,
    dominationRule                  PrivilegePolicyIdentifier}

AttributeIdentifier  ::= ATTRIBUTE.&id({AttributeIDs})

AttributeIDs ATTRIBUTE  ::= {...}

AttributeName  ::= UTF8String(SIZE(1..MAX))

AttributeDescription  ::= UTF8String(SIZE(1..MAX))

PrivilegePolicyIdentifier     ::=   SEQUENCE {
    privilegePolicy                 PrivilegePolicy,
    privPolSyntax                   InfoSyntax }

attDescriptor MATCHING-RULE  ::= {
    SYNTAX          AttributeDescriptorSyntax
    ID              id-mr-attDescriptorMatch }

userNotice  EXTENSION  ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF UserNotice
    IDENTIFIED BY       id-ce-userNotice }

targetingInformation  EXTENSION  ::= {
    SYNTAX              SEQUENCE SIZE (1..MAX) OF Targets
    IDENTIFIED BY       id-ce-targetInformation }

Targets  ::=   SEQUENCE SIZE (1..MAX) OF Target

Target::=CHOICE {
    targetName          [0]         GeneralName,
    targetGroup         [1]         GeneralName,
    targetCert          [2]         TargetCert }
```

```
TargetCert    ::=  SEQUENCE {
    targetCertificate     IssuerSerial,
    targetName            GeneralName OPTIONAL,
    certDigestInfo        ObjectDigestInfo OPTIONAL }

noRevAvail EXTENSION ::= {
    SYNTAX                NULL
    IDENTIFIED BY         id-ce-noRevAvail }

acceptablePrivilegePolicies EXTENSION ::= {
    SYNTAX                AcceptablePrivilegePoliciesSyntax
    IDENTIFIED BY         id-ce-acceptablePrivilegePolicies }

AcceptablePrivilegePoliciesSyntax   ::=  SEQUENCE SIZE (1..MAX) OF PrivilegePolicy

-- attributions d'identificateur d'objet --

-- classes d'objets --
-
id-oc-pmiUser                          OBJECT IDENTIFIER ::=   {id-oc 24}
id-oc-pmiAA                            OBJECT IDENTIFIER ::=   {id-oc 25}
id-oc-pmiSOA                           OBJECT IDENTIFIER ::=   {id-oc 26}
id-oc-attCertCRLDistributionPts        OBJECT IDENTIFIER ::=   {id-oc 27}
id-oc-privilegePolicy                  OBJECT IDENTIFIER ::=   {id-oc 32}
id-oc-pmiDelegationPath                OBJECT IDENTIFIER ::=   {id-oc 33}

-- attributs d'annuaire --

id-at-attributeCertificate             OBJECT IDENTIFIER ::=   {id-at 58}
id-at-attributeCertificateRevocationList OBJECT IDENTIFIER ::= {id-at 59}
id-at-aACertificate                    OBJECT IDENTIFIER ::=   {id-at 61}
id-at-attributeDescriptorCertificate   OBJECT IDENTIFIER ::=   {id-at 62}
id-at-attributeAuthorityRevocationList OBJECT IDENTIFIER ::=   {id-at 63}
id-at-privPolicy                       OBJECT IDENTIFIER ::=   {id-at 71}
id-at-role                             OBJECT IDENTIFIER ::=   {id-at 72}
id-at-delegationPath                   OBJECT IDENTIFIER ::=   {id-at 73}

-- extensions de certificat d'attribut --

id-ce-authorityAttributeIdentifier     OBJECT IDENTIFIER ::=   {id-ce 38}
id-ce-roleSpecCertIdentifier           OBJECT IDENTIFIER ::=   {id-ce 39}
id-ce-basicAttConstraints              OBJECT IDENTIFIER ::=   {id-ce 41}
id-ce-delegatedNameConstraints         OBJECT IDENTIFIER ::=   {id-ce 42}
id-ce-timeSpecification                OBJECT IDENTIFIER ::=   {id-ce 43}
id-ce-attributeDescriptor              OBJECT IDENTIFIER ::=   {id-ce 48}
id-ce-userNotice                       OBJECT IDENTIFIER ::=   {id-ce 49}
id-ce-sOAIdentifier                    OBJECT IDENTIFIER ::=   {id-ce 50}
id-ce-acceptableCertPolicies           OBJECT IDENTIFIER ::=   {id-ce 52}
id-ce-targetInformation                OBJECT IDENTIFIER ::=   {id-ce 55}
id-ce-noRevAvail                       OBJECT IDENTIFIER ::=   {id-ce 56}
id-ce-acceptablePrivilegePolicies      OBJECT IDENTIFIER ::=   {id-ce 57}

-- règles de concordance d'infrastructure PMI --

id-mr-attributeCertificateMatch        OBJECT IDENTIFIER ::=   {id-mr 42}
id-mr-attributeCertificateExactMatch   OBJECT IDENTIFIER ::=   {id-mr 45}
id-mr-holderIssuerMatch                OBJECT IDENTIFIER ::=   {id-mr 46}
id-mr-authAttIdMatch                   OBJECT IDENTIFIER ::=   {id-mr 53}
id-mr-roleSpecCertIdMatch              OBJECT IDENTIFIER ::=   {id-mr 54}
id-mr-basicAttConstraintsMatch         OBJECT IDENTIFIER ::=   {id-mr 55}
id-mr-delegatedNameConstraintsMatch    OBJECT IDENTIFIER ::=   {id-mr 56}
id-mr-timeSpecMatch                    OBJECT IDENTIFIER ::=   {id-mr 57}
id-mr-attDescriptorMatch               OBJECT IDENTIFIER ::=   {id-mr 58}
id-mr-acceptableCertPoliciesMatch      OBJECT IDENTIFIER ::=   {id-mr 59}
id-mr-delegationPathMatch              OBJECT IDENTIFIER ::=   {id-mr 61}

END
```

Annexe B

Règles de génération et de traitement des listes CRL

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

B.1 Introduction

Un participant faisant confiance (utilisateur de certificat) doit être en mesure de vérifier le statut d'un certificat pour décider s'il doit ou non se fier à ce certificat. Les listes de révocation de certificat sont l'un des procédés utilisables par des participants pour obtenir ces informations. Il est également possible d'utiliser d'autres procédés qui sortent du domaine d'application de la présente Spécification.

Cette annexe traite de l'utilisation des listes CRL, par des participants faisant confiance, pour la vérification du statut de révocation d'un certificat. Les autorités peuvent appliquer diverses politiques d'émission de listes de révocation. L'autorité émettrice du certificat peut, dans certains cas, permettre à une autre autorité de révoquer des certificats qu'elle a émis. Certaines autorités peuvent combiner dans une liste unique les révocations de certificats d'entités finales et de certificats d'autorité de certification, alors que d'autres peuvent utiliser des listes distinctes. Certaines autorités peuvent subdiviser l'ensemble de leurs certificats dans des sous-ensembles de liste CRL et d'autres peuvent émettre des mises à jour différentielles d'une liste de révocation entre des intervalles réguliers d'émission de liste CRL. Il en résulte que les participants faisant confiance doivent être en mesure de déterminer le domaine d'application des listes CRL auxquelles ils accèdent, afin de s'assurer qu'ils disposent de l'ensemble complet des informations de révocation couvrant le domaine d'application du certificat en question pour les motifs de révocation pertinents, compte tenu de la politique dans le cadre de laquelle ils sont actifs. L'extension **crlScope** peut servir pour la détermination du domaine d'application. Cette annexe présente un procédé qui peut être utilisé si l'extension **crlScope** ne figure pas dans les listes CRL.

Cette annexe traite de la vérification du statut de révocation de certificats de clé publique au moyen de listes CRL, EPRL et CARL. Cette description peut toutefois également s'appliquer à la vérification du statut de révocation de certificats d'attribut utilisant des listes de révocation de certificat d'attribut (ACRL, *attribute certificate revocation lists*) et des listes de révocation d'autorité d'attributs (AARL, *attribute authority revocation lists*). Nous prendrons en considération, dans la présente annexe, les listes ACRL à la place des listes CRL et les listes AARL à la place des listes CARL.

B.1.1 Types de liste CRL

Un participant faisant confiance peut disposer de listes CRL de l'un ou de plusieurs des types suivants en fonction des caractéristiques de révocation de la politique de l'autorité émettrice du certificat:

- liste CRL pleine et complète;
- liste CRL d'entité finale (EPRL) pleine et complète;
- liste de révocation d'autorité de certification (CARL) pleine et complète;
- liste CRL, EPRL ou CARL de point de répartition;
- liste CRL, EPRL ou CARL indirecte (ICRL);
- liste CRL, EPRL ou CARL delta (dCRL);
- liste dCRL, EPRL ou CARL indirecte.

Une liste CRL pleine et complète contient tous les certificats d'entité finale et d'autorité de certification révoqués, émis par une autorité pour l'ensemble des motifs.

Une liste EPRL pleine et complète contient tous les certificats d'entité finale révoqués, émis par une autorité pour l'ensemble des motifs.

Une liste CARL pleine et complète contient tous les certificats d'autorité de certification révoqués, émis par une autorité pour l'ensemble des motifs.

Une liste CRL, EPRL ou CARL de point de répartition couvre la totalité ou un sous-ensemble des certificats émis par une autorité. Le choix du sous-ensemble peut se faire selon un certain nombre de critères.

Une liste CRL, EPRL ou CARL indirecte (iCRL) contient une liste de certificats révoqués qui n'ont pas été émis, en tout ou partie, par l'autorité qui émet et signe cette liste.

Une liste CRL, EPRL ou CARL delta contient uniquement des modifications pour une liste CRL qui est complète pour le domaine d'application donné au moment indiqué par la liste CRL à laquelle la liste dCRL fait référence. Il convient de noter que la liste CRL de référence peut être une liste CRL complète pour le domaine d'application donné ou une liste dCRL utilisée de manière locale pour construire une liste CRL complète. Tous les types de liste CRL précédents (à l'exception de la liste dCRL) sont des types de listes CRL qui sont complètes pour leur domaine d'application. Une liste dCRL doit être utilisée conjointement à une liste CRL associée qui est complète pour le même domaine d'application afin de fournir une représentation complète du statut de révocation des certificats.

Une liste CRL, EPRL ou CARL delta indirecte est une liste CRL qui contient uniquement des modifications pour un ensemble d'une ou plusieurs listes CRL, qui sont complètes pour leurs domaines d'application et pour lesquelles tout ou partie des certificats ont pu être émis par une autorité différente de celle qui signe et émet cette liste CRL.

Dans cette annexe, l'expression "domaine d'application d'une liste CRL" est définie selon deux axes indépendants. L'une des dimensions concerne l'ensemble des certificats couverts par la liste. L'autre dimension concerne des codes motif couverts par la liste CRL. Le domaine d'application d'une liste CRL peut être déterminé de l'une ou de plusieurs des manières suivantes:

- extension de point de répartition émetteur (IDP) dans la liste CRL;
- extension de domaine d'application de liste CRL dans la liste CRL;
- autres moyens hors du domaine d'application de la présente Spécification.

B.1.2 Traitement de liste CRL

Lorsqu'il utilise les listes CRL pour déterminer si un certificat est révoqué, un participant faisant confiance doit avoir la certitude d'utiliser la ou les listes CRL pertinentes pour ce certificat. La présente annexe décrit une procédure, constituée d'un certain nombre d'étapes, qui permet d'obtenir et de traiter les listes CRL pertinentes. Une implémentation sera fonctionnellement équivalente au comportement externe résultant de cette procédure. L'algorithme utilisé par une implémentation particulière pour obtenir les résultats corrects (c'est-à-dire le statut de révocation d'un certificat) à partir des entrées données (le certificat lui-même et des entrées en provenance de la politique locale) n'est pas normalisé. Par exemple, bien que cette procédure soit décrite sous la forme d'une succession ordonnée d'étapes de traitement, une implémentation peut utiliser des listes CRL qui sont présentes dans son cache local plutôt que d'extraire des listes CRL chaque fois qu'elle traite un certificat, à condition que ces listes CRL soient complètes pour le domaine d'application du certificat et prennent en considération la totalité des paramètres du certificat et de la politique.

La présente annexe ne décrit pas de procédure de suivi de pointeur dans une structure de liste CRL qui contient l'extension **statusReferrals** [*références de statut*]. Une liste CRL contenant cette extension ne sera pas utilisée, par un participant faisant confiance, comme source de vérification du statut de révocation d'un certificat, mais elle peut servir d'outil supplémentaire pour localiser les listes CRL adéquates permettant la vérification du statut de révocation.

Les paragraphes B.2 à B.5 qui suivent décrivent les étapes générales suivantes:

1) détermination des paramètres pour les listes CRL;
2) détermination des listes CRL nécessaires;
3) extraction des listes CRL;
4) traitement des listes CRL.

L'étape 1) identifie les paramètres en provenance du certificat et d'autres sources, qui seront utilisés pour déterminer quels sont les types de listes CRL nécessaires.

L'étape 2) utilise la valeur de ces paramètres pour déterminer les listes CRL.

L'étape 3) identifie les attributs d'annuaire à partir desquels peuvent être extraits les types de liste CRL.

L'étape 4) décrit le traitement des listes CRL pertinentes.

B.2 Détermination des paramètres pour les listes CRL

Les paramètres permettant de déterminer la pertinence des listes CRL candidates sont fournis par des informations figurant dans le certificat proprement dit ainsi que par des informations provenant de la politique à laquelle est soumis le participant faisant confiance. Les informations suivantes sont nécessaires pour déterminer les types des listes CRL adéquates:

- type de certificat (entité finale ou autorité d'attribut);
- point de répartition de liste CRL critique;
- liste CRL critique la plus récente;
- codes motif concernés.

Le type de certificat peut être déterminé à partir de l'extension de contraintes de base qu'il contient. Cette extension indique, lorsqu'elle est présente, si le certificat est un certificat d'autorité de certification ou un certificat d'entité finale. Le certificat est considéré comme étant un certificat d'entité finale si l'extension est absente. Cette information est nécessaire pour déterminer si la liste CRL, EPRL ou CARL peut être utilisée pour vérifier la révocation du certificat.

Si le certificat contient une extension critique de point de répartition de liste CRL, le système de traitement de certificat du participant faisant confiance doit reconnaître cette extension pour se fier au certificat. Par exemple, la dépendance par rapport à une liste CRL complète ne serait pas suffisante.

Si le certificat contient une extension critique de liste CRL la plus récente, le participant faisant confiance ne peut alors pas utiliser le certificat s'il n'a pas extrait et vérifié au préalable la liste CRL la plus récente.

Les codes motif concernés sont déterminés par la politique et sont en général fournis par l'application. Il est recommandé qu'ils englobent la totalité des codes motif. Ces informations sont nécessaires pour déterminer quelles sont les listes CRL suffisantes du point de vue des codes motif.

Il convient de noter que la politique peut également imposer ou non qu'un participant faisant confiance ait l'obligation de vérifier le statut de révocation pour des listes dCRL, même si l'extension **freshestCRL** [*liste CRL la plus récente*] est marquée comme non critique ou ne figure pas dans le certificat. Le traitement de ces listes dCRL optionnelles est décrit dans l'étape 4), bien qu'il n'appartient pas à cette étape.

B.3 Détermination des listes CRL nécessaires

Les valeurs des paramètres décrits en B.2 déterminent les critères de choix des types de liste CRL nécessaires pour la vérification du statut de révocation d'un certificat donné. La détermination des types de liste CRL peut se faire sur la base des ensembles de critères suivants, de la manière décrite dans les paragraphes B.3.1 à B.3.4 qui suivent.

- certificat d'entité finale avec déclaration critique de point de répartition de liste CRL;
- certificat d'entité finale sans déclaration critique de point de répartition de liste CRL;
- certificat d'autorité de certification avec déclaration critique de point de répartition de liste CRL;
- certificat d'autorité de certification sans déclaration critique de point de répartition de liste CRL.

Le traitement des autres paramètres (extension critique de liste CRL la plus récente et ensemble des codes motif concernés) est décrit dans chaque paragraphe.

Il convient de noter qu'il se peut, dans chaque cas, que plusieurs types de liste CRL satisfassent aux prescriptions. Le participant faisant confiance peut dans ce cas choisir d'utiliser l'un quelconque des types adéquats.

B.3.1 Entité finale avec point de répartition de liste CRL critique

Les listes CRL suivantes seront extraites si le certificat est un certificat d'entité finale qui contient une extension **cRLDistributionPoints** [*points de répartition de liste CRL*] marquée comme critique:

- une liste CRL de l'un des points de répartition de liste CRL nommés couvrant un ou plusieurs des codes motif concernés;
- si cette liste CRL ne couvre pas tous les codes motif concernés, le statut de révocation des autres codes motif peut alors être obtenu par toute combinaison des listes CRL suivantes:
  - autres listes CRL de point de répartition;
  - autres listes CRL complètes;
  - autres listes EPRL complètes.

Si le certificat contient une extension de liste CRL la plus récente marquée comme critique, il sera alors nécessaire d'obtenir une ou plusieurs listes CRL à partir d'un ou de plusieurs des points de répartition nommés dans cette extension pour garantir la vérification des informations de révocation les plus récentes pour tous les codes motif concernés.

B.3.2 Entité finale sans point de répartition de liste CRL critique

Les listes CRL suivantes seront extraites si le certificat est un certificat d'entité finale dans lequel l'extension **cRLDistributionPoints** ne figure pas ou si elle est présente et marquée comme non critique:

- listes CRL de point de répartition (éventuellement présentes);
- listes CRL complètes;
- listes EPRL complètes.

Si le certificat contient également une extension de liste CRL la plus récente marquée comme critique, il sera alors nécessaire d'obtenir une ou plusieurs listes CRL à partir d'un ou de plusieurs des points de répartition nommés dans cette extension pour garantir la vérification des informations de révocation les plus récentes pour tous les codes motif concernés.

B.3.3 Autorité de certification avec point de répartition de liste CRL critique

Les listes CRL ou CARL suivantes seront extraites si le certificat d'autorité de certification contient une extension **cRLDistributionPoints** marquée comme critique:

- une liste CRL ou CARL de l'un des points de répartition nommés couvrant un ou plusieurs des codes motif concernés;
- si cette liste CRL ou CARL ne couvre pas tous les codes motif concernés, le statut de révocation des autres codes motif peut être obtenu par toute combinaison des listes CRL ou CARL suivantes:
  - autres listes CRL ou CARL de point de répartition;
  - autres listes CRL complètes;
  - autres listes EPRL complètes.

Si le certificat contient également une extension de liste CRL la plus récente marquée comme critique, il sera alors nécessaire d'obtenir une ou plusieurs listes CRL ou CARL à partir d'un ou de plusieurs des points de répartition nommés dans cette extension pour garantir la vérification des informations de révocation les plus récentes pour tous les codes motif concernés.

B.3.4 Autorité de certification sans point de répartition de liste CRL critique

Si le certificat est un certificat d'autorité de certification dans lequel l'extension **cRLDistributionPoints** ne figure pas ou est présente et marquée comme non critique, le statut de révocation pour les codes concernés peut alors être fourni par toute combinaison des listes CRL suivantes:

- listes CRL/CARL de point de répartition (éventuellement présentes);
- listes CRL complètes;
- listes CARL complètes.

Si le certificat contient également une extension de liste CRL la plus récente marquée comme critique, il sera alors nécessaire d'obtenir une ou plusieurs listes CRL ou CARL à partir d'un ou de plusieurs des points de répartition nommés dans cette extension pour garantir la vérification des informations de révocation les plus récentes pour tous les codes motif concernés.

B.4 Extraction des listes CRL

Lorsque le participant faisant confiance extrait de l'annuaire les listes CRL adéquates, ces dernières sont obtenues à partir de l'entrée d'annuaire du point de répartition de liste CRL ou de l'émetteur de certificat par extraction des attributs adéquats, c'est-à-dire, d'un ou de plusieurs des attributs suivants:

- liste de révocation de certificat;
- liste de révocation d'autorité;
- liste de révocation delta.

B.5 Traitement des listes CRL

Un participant faisant confiance est prêt à traiter les listes CRL une fois qu'il a analysé les paramètres comme indiqué en B.2, identifié les types de liste CRL adéquats comme décrit en B.3 et extrait un ensemble adéquat de listes CRL comme décrit en B.4. L'ensemble de listes CRL contiendra au moins une liste CRL de base et peut également contenir une ou plusieurs listes dCRL. Le participant faisant confiance doit s'assurer que chaque liste CRL traitée est correcte en ce qui concerne son domaine d'application. Le participant faisant confiance a déjà déterminé que la liste CRL est pertinente pour le domaine d'application du certificat concerné lors des phases de traitement B.2 et B.3 décrites précédemment. Il est en outre nécessaire d'effectuer des vérifications de validité sur les listes CRL et de déterminer si le certificat a été révoqué ou non. Ces vérifications sont décrites dans les paragraphes B.5.1 à B.5.4 qui suivent.

B.5.1 Validation du domaine d'application de liste CRL

Comme décrit en B.3, un ou plusieurs types de liste CRL peuvent être utilisés comme liste de base pour la vérification du statut de révocation d'un certificat. Les participants faisant confiance peuvent disposer d'un ou de plusieurs types de liste CRL de base suivants, en fonction de la politique d'émission de liste CRL de l'autorité émettrice:

- liste CRL complète pour toutes les entités;
- liste EPRL complète;
- liste CARL complète;
- liste CRL, EPRL ou CARL basée sur un point de répartition.

Les paragraphes B.5.1.1 à B.5.1.4 indiquent l'ensemble des conditions qui doivent être satisfaites pour qu'un participant faisant confiance puisse utiliser une liste CRL de chacun des types comme liste CRL de base pour la vérification du statut de révocation du certificat pour les codes motif concernés.

Le cas des listes CRL de base indirectes est traité dans chaque paragraphe.

B.5.1.1 Liste CRL complète

Toutes les conditions suivantes doivent être satisfaites pour déterminer si une liste CRL est complète pour les certificats d'entité finale et les certificats d'autorité de certification ainsi que pour tous les codes motif concernés:

- l'extension d'indicateur de liste CRL delta ne doit pas être présente;
- l'extension de point de répartition émetteur peut être présente;
- l'extension de point de répartition émetteur ne doit pas contenir de champ **distributionPoint**;
- l'extension de point de répartition émetteur ne doit pas contenir de champ **onlyContainsUserCerts** positionné sur "Vrai";
- l'extension de point de répartition émetteur ne doit pas contenir de champ **onlyContainsAuthorityCerts** positionné sur "Vrai";
- l'extension de point de répartition émetteur ne doit pas contenir de champ **onlyContainsAttributeCerts** positionné sur "Vrai";
- si le champ **reasonCodes** [*codes motif*] est présent dans l'extension de point de répartition émetteur, il doit alors contenir tous les motifs concernés par l'application;
- l'extension de point de répartition émetteur peut contenir ou non le champ **indirectCRL**. Il s'ensuit qu'il n'est pas nécessaire de vérifier ce champ.

B.5.1.2 Liste EPRL complète

Toutes les conditions suivantes doivent être satisfaites pour déterminer si une liste CRL est une liste EPRL complète pour tous les codes motif concernés:

- l'extension d'indicateur de liste CRL delta ne doit pas être présente;
- l'extension de point de répartition émetteur doit être présente;
- l'extension de point de répartition émetteur ne doit pas contenir de champ **distributionPoint**;
- l'extension de point de répartition émetteur doit contenir un champ **onlyContainsUserCerts**. La valeur de ce champ doit être positionnée sur "Vrai";
- l'extension de point de répartition émetteur ne doit pas contenir de champ **onlyContainsAuthorityCerts** positionné sur "Vrai";
- l'extension de point de répartition émetteur ne doit pas contenir de champ **onlyContainsAttributeCerts** positionné sur "Vrai";
- si le champ **reasonCodes** est présent dans l'extension de point de répartition émetteur, le champ de code motif doit alors contenir tous les motifs concernés par l'application;
- l'extension de point de répartition émetteur peut contenir ou non le champ **indirectCRL**. Il s'ensuit qu'il n'est pas nécessaire de vérifier ce champ.

Cette liste CRL peut uniquement être utilisée si le participant faisant confiance a déterminé que le sujet du certificat est une entité finale. Il en résulte que, pour les certificats de version 3, si le certificat sujet contient l'extension **basicConstraints**, la valeur du composant **cA** doit alors être égale à "Vrai".

B.5.1.3 Liste CARL complète

Toutes les conditions suivantes doivent être satisfaites pour déterminer si une liste CRL est une liste CARL complète pour tous les codes motif concernés:

- l'extension d'indicateur de liste CRL delta ne doit pas être présente;
- l'extension de point de répartition émetteur doit être présente;
- l'extension de point de répartition émetteur ne doit pas contenir de champ **distributionPoint**;
- l'extension de point de répartition émetteur ne doit pas contenir de champ **onlyContainsUserCerts** positionné sur "Vrai";
- l'extension de point de répartition émetteur ne doit pas contenir le champ **onlyContainsAttributeCerts** positionné sur "Vrai";
- l'extension de point de répartition émetteur doit contenir le champ **onlyContainsAuthorityCerts**. Ce champ doit être positionné sur "Vrai";
- si le champ **reasonCodes** est présent dans l'extension de point de répartition émetteur, le champ de code motif doit alors contenir tous les motifs concernés par l'application;
- l'extension de point de répartition émetteur peut contenir ou non le champ **indirectCRL**. Il s'ensuit qu'il n'est pas nécessaire de vérifier ce champ.

Cette liste CARL peut uniquement être utilisée si le participant faisant confiance a déterminé que le sujet du certificat est une autorité de certification. Il en résulte que, pour les certificats de version 3, si le certificat sujet contient l'extension **basicConstraints**, la valeur du composant **cA** doit alors être égale à "Vrai".

B.5.1.4 Liste CRL, EPRL ou CARL basée sur un point de répartition

Toutes les conditions suivantes doivent être satisfaites pour déterminer si une liste CRL est l'une de celles indiquées par une extension de point de répartition de liste CRL figurant dans le certificat:

- soit le champ **distributionPoint** est absent de l'extension de point de répartition émetteur de la liste CRL (uniquement lorsque cette extension n'est pas marquée comme critique), soit l'un des noms du champ **distributionPoint** de l'extension de point de répartition de liste CRL du certificat doit correspondre à l'un des noms figurant dans le champ **distributionPoint** de l'extension de point de répartition émetteur de la liste CRL. En variante, l'un des noms du champ **cRLIssuer** de l'extension de point de répartition de liste CRL du certificat peut correspondre à l'un des noms de point de répartition du point IDP;
- si le certificat est un certificat d'entité finale, la liste CRL ne doit alors pas contenir de champ **onlyContainsAuthorityCerts** positionné sur "Vrai" dans l'extension de point de répartition émetteur de la liste CRL;
- si le champ **onlyContainsAuthorityCerts** est positionné sur "Vrai" dans l'extension de point de répartition émetteur de la liste CRL, le certificat en cours de vérification doit alors contenir l'extension **basicConstraints** avec un composant **cA** positionné sur "Vrai";
- si le champ **reasonCodes** figure dans l'extension de point de répartition de liste CRL du certificat, il doit alors, soit être absent de l'extension de point de répartition émetteur de la liste CRL, soit contenir l'un au moins des codes motif déclarés dans l'extension de point de répartition de liste CRL du certificat;
- si le champ **cRLIssuer** ne figure pas dans l'extension de point de répartition de liste CRL du certificat, la liste CRL doit alors être signée par la même autorité de certification que celle qui a signé le certificat;
- si le champ **cRLIssuer** figure dans l'extension de point de répartition de liste CRL du certificat, la liste CRL doit alors être signée par l'émetteur de liste CRL indiqué dans l'extension de point de répartition de liste CRL du certificat et doit contenir le champ **indirectCRL** dans l'extension de point de répartition émetteur.

B.5.2 Validation du domaine d'application de liste CRL delta

Le participant faisant confiance peut également vérifier des listes dCRL, soit parce que cette vérification est exigée par une extension **freshestCRL** critique figurant dans le certificat, soit parce que la politique à laquelle est soumis le participant faisant confiance prend en charge la vérification de liste dCRL.

Un participant faisant confiance peut toujours avoir la certitude qu'il dispose des informations de liste CRL adéquates pour un certificat si toutes les conditions suivantes sont satisfaites:

- la liste CRL de base utilisée par le participant faisant confiance est pertinente pour le domaine d'application du certificat;
- la liste CRL delta utilisée par le participant faisant confiance est pertinente pour le domaine d'application du certificat;
- la liste CRL de base a été émise à un instant égal ou antérieur à celui de l'émission de la liste CRL de base référencée par la liste dCRL.

Toutes les conditions suivantes doivent être satisfaites pour déterminer que la liste dCRL est pertinente pour le certificat:

- l'extension d'indicateur de liste CRL delta doit être présente;
- la liste dCRL doit avoir été émise après la liste CRL de base. L'une des manières de s'assurer de la validité de cette condition est de vérifier que le numéro de liste CRL figurant dans l'extension **crlNumber** de la liste dCRL est supérieur au numéro de liste CRL figurant dans l'extension **crlNumber** de la liste CRL utilisée par le participant et que les champs **cRLStreamIdentifier** [*identificateur de flux de liste CRL*] de la liste de base et de la liste dCRL concordent. Cette démarche peut nécessiter un traitement du rebouclage des numéros. Une autre solution consiste à comparer les champs **thisUpdate** de la liste de base et de la liste dCRL dont dispose le participant faisant confiance;
- la liste CRL de base utilisée par le participant faisant confiance doit être celle pour laquelle la liste dCRL est émise ou une liste postérieure. L'une des manières de s'assurer de la validité de cette condition est de vérifier que le numéro de liste CRL figurant dans l'extension **deltaCRLIndicator** de la liste dCRL est inférieur ou égal au numéro de liste CRL figurant dans l'extension **crlNumber** de la liste CRL qui est utilisée par le participant et que les champs **cRLStreamIdentifier** de la liste de base et de la liste dCRL concordent. Cette démarche peut nécessiter un traitement du rebouclage des numéros. Une autre solution consiste à comparer les champs **thisUpdate** de la liste de CRL de base dont dispose le participant faisant confiance et de la liste CRL de base sur laquelle pointe la liste dCRL. Une autre solution consiste également à comparer le champ **thisUpdate** de la liste CRL de base dont dispose le participant faisant confiance avec l'extension **baseUpdateTime** [*instant de mise à jour de base*] de la liste la liste dCRL dont dispose le participant faisant confiance;

  NOTE – Un participant faisant confiance peut toujours construire une liste CRL de base en appliquant une liste dCRL à une liste CRL de base, tant que les deux règles précédentes sont valables, en utilisant la vérification au moyen des champs **crlNumber** et **cRLStreamIdentifier**. Dans un tel cas, l'extension **crlNumber** et le champ **thisUpdate** de la nouvelle liste CRL de base sont ceux de la liste dCRL. Le participant faisant confiance ne connaît pas le champ **nextUpdate** de la nouvelle liste CRL de base et n'en a pas besoin pour établir une association avec une autre liste dCRL.

- si la liste dCRL contient une extension de point de répartition émetteur, le domaine d'application du point de répartition émetteur sera alors cohérent avec le certificat, comme décrit en B.5.1.4 ci-dessous;
- si la liste dCRL contient une extension de domaine d'application de liste CRL, le certificat appartiendra alors au domaine d'application de la liste CRL;
- si la liste dCRL ne contient aucune des extensions: **streamIdentifier**, **crlScope** et **issuingDistributionPoint**, elle sera alors utilisée uniquement conjointement à une liste CRL de base pleine et complète.

B.5.3 Vérification de validité et d'actualité de la liste CRL de base

Toutes les conditions suivantes doivent être satisfaites pour vérifier qu'une liste CRL de base est exacte et n'a pas été modifiée depuis son émission:

- le participant faisant confiance doit pouvoir obtenir la clé publique de l'émetteur identifié dans la liste CRL en utilisant des moyens d'authentification;
- la signature de la liste CRL de base doit être vérifiée au moyen de cette clé publique authentifiée;
- si le champ **nextUpdate** est présent, l'instant actuel doit être antérieur à la valeur de ce champ;
- le nom de l'émetteur dans la liste CRL doit correspondre au nom de l'émetteur dans le certificat dont on vérifie la révocation, sauf si la liste CRL est extraite à partir du point de répartition de liste CRL du certificat et si l'extension de point de répartition de liste CRL contient le composant **cRLIssuer**. Dans ce cas, l'un des noms du composant **cRLIssuer** dans l'extension de point de répartition de liste CRL doit correspondre au nom d'émetteur figurant dans la liste CRL.

B.5.4 Validité et vérifications de la liste CRL delta

Toutes les conditions suivantes doivent être satisfaites pour vérifier qu'une liste dCRL est exacte et n'a pas été modifiée depuis son émission:

- le participant faisant confiance doit pouvoir obtenir la clé publique de l'émetteur identifié dans la liste CRL en utilisant des moyens d'authentification;
- la signature de la liste dCRL doit être vérifiée en utilisant cette clé publique authentifiée;
- si le champ **nextUpdate** est présent, l'instant actuel doit être antérieur à la valeur de ce champ;
- le nom de l'émetteur dans la liste CRL doit correspondre au nom de l'émetteur dans le certificat dont on vérifie la révocation, sauf si l'une des deux conditions suivantes est vérifiée:
  - une liste CRL delta est extraite du point de répartition de liste CRL figurant dans le certificat et l'extension de point de répartition de liste CRL contient le composant **cRLIssuer**. Dans ce cas, l'un des noms figurant dans le composant **cRLIssuer** de l'extension de point de répartition de liste doit correspondre au nom de l'émetteur figurant dans la liste CRL; ou
  - une liste CRL delta est extraite du champ **freshestCRL** du certificat et l'extension de domaine d'application de la liste CRL contient un composant **PerAuthorityScope** avec un champ **authorityName** qui correspond au nom de l'émetteur du certificat.

Annexe C

Exemples d'émission de liste CRL delta

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

C.1 Introduction

Il existe deux modèles d'émission de listes CRL impliquant l'utilisation de listes dCRL pour un ensemble donné de certificats.

Dans le premier modèle, chaque liste dCRL fait référence à la liste CRL la plus récente qui est complète pour le domaine d'application donné. Plusieurs listes dCRL peuvent être émises pour un même domaine d'application avant l'émission d'une nouvelle liste CRL complète pour ce domaine d'application. La nouvelle liste CRL complète pour ce domaine est utilisée comme base pour la nouvelle séquence de listes dCRL et constitue la liste CRL qui fait l'objet de la référence dans l'extension adéquate de la liste dCRL. Lorsque la nouvelle liste CRL complète est émise pour le domaine d'application, une liste dCRL finale pour la liste CRL complète est également émise pour le domaine d'application.

Le deuxième modèle, qui est très proche, diffère du précédent par le fait que la liste CRL qui fait l'objet d'une référence de la part d'une liste dCRL n'est pas nécessairement complète pour le domaine concerné (c'est-à-dire que la liste CRL de référence peut avoir été émise sous la forme d'une liste dCRL). Si la liste CRL de référence est complète pour le domaine d'application donné, elle n'est pas nécessairement la plus récente.

Un système utilisant des certificats qui traite une liste dCRL doit également disposer d'une liste CRL complète pour le domaine d'application donné et qui est au moins aussi actuelle que la liste CRL référencée par la liste dCRL. Cette liste CRL complète pour le domaine d'application donné peut avoir été émise par l'autorité responsable ou peut avoir été établie de manière locale par le système utilisant des certificats. Il convient de noter que dans certaines situations, il peut exister des informations dupliquées entre la liste dCRL et la liste CRL complète pour le domaine d'application donné, par exemple si le système utilisant des certificats dispose d'une liste CRL qui a été émise après celle qui est référencée dans la liste dCRL.

Le tableau suivant donne trois exemples d'utilisation de listes dCRL. Le premier exemple correspond au schéma classique décrit dans le premier modèle ci-dessus. Les deux exemples suivants sont des variantes du deuxième modèle décrit ci-dessus.

Dans le deuxième exemple, l'autorité émet des listes CRL complètes pour le domaine d'application donné tous les deux jours et les listes dCRL font référence à l'avant dernière liste CRL complète. Ce procédé peut être utile dans des environnements où il est nécessaire de réduire le nombre d'utilisateurs qui accèdent simultanément à un référentiel afin d'extraire une liste CRL complète pour un domaine d'application donné. Dans cet exemple, les utilisateurs qui disposent de la liste CRL complète la plus récente et ceux qui ont accès à l'avant dernière liste CRL complète peuvent utiliser la même liste dCRL. Les deux ensembles d'utilisateurs disposent d'informations complètes concernant les certificats pour le domaine donné au moment de l'émission de la liste dCRL dont ils disposent.

Dans le troisième exemple, des listes CRL complètes pour le domaine d'application donné sont émises une fois par semaine comme dans le premier exemple, mais chaque liste dCRL fait référence à une base d'informations de révocation datant de 7 jours avant cette liste dCRL.

Cette annexe ne fournit pas d'exemple d'utilisation de listes CRL indirectes, mais ce cas constitue un sur-ensemble de ces exemples.

Selon la politique locale, seuls ces exemples ou d'autres variantes sont possibles. Certains des facteurs que peuvent être pris en considération pour la mise en place de cette politique sont le nombre d'utilisateurs et leur fréquence d'accès aux listes CRL, la duplication de listes CRL, le partage de charge entre les systèmes d'annuaire qui détiennent les listes CRL, les performances, les prescriptions de temps de latence, etc.

Jour	Exemple 1 – La liste delta fait référence à la liste CRL qui est complète pour le domaine d'application donné		Exemple 2 – La liste delta fait référence à l'avant dernière liste CRL qui est complète pour le domaine d'application		Exemple 2 – La liste delta fait référence à des informations de révocation datant de 7 jours	
	Liste CRL complète	Liste CRL delta	Liste CRL complète	Liste CRL delta	Liste CRL complète	Liste CRL delta
8	thisUpdate=jour 8 nextUpdate=jour 15 crlNumber=8	thisUpdate=jour 8 nextUpdate=jour 9 crlNumber=8 BaseCRLNumber=1	thisUpdate=jour 8 nextUpdate=jour 10 crlNumber=8	thisUpdate=jour 8 nextUpdate=jour 9 crlNumber=8 BaseCRLNumber=6	thisUpdate=jour 8 nextUpdate=jour 15 cRLNumber=8	thisUpdate=jour 8 nextUpdate=jour 9 cRLNumber=8 BaseCRLNumber= 1
9	non émis	thisUpdate=jour 9 nextUpdate=jour 10 crlNumber=9 BaseCRLNumber=8	non émis	thisUpdate=jour 9 nextUpdate=jour 10 crlNumber=9 BaseCRLNumber=6	non émis	thisUpdate=jour 9 nextUpdate=jour 10 cRLNumber=9 BaseCRLNumber= 2
10	non émis	thisUpdate=jour 10 nextUpdate=jour 11 crlNumber=10 BaseCRLNumber=8	thisUpdate=jour 10 nextUpdate=jour 12 crlNumber=10	thisUpdate=jour 10 nextUpdate=jour 11 crlNumber=10 BaseCRLNumber=8	non émis	thisUpdate=jour 10 nextUpdate=jour 11 cRLNumber=10 BaseCRLNumber= 3
11-14	Mêmes profils que pour les jours précédents					
15	thisUpdate=jour 15 nextUpdate=jour 22 crlNumber=15	thisUpdate=jour 15 nextUpdate=jour 16 crlNumber=15 BaseCRLNumber=8	non émis	thisUpdate=jour 15 nextUpdate=jour 16 crlNumber=15 BaseCRLNumber=12	thisUpdate=jour 15 nextUpdate=jour 22 cRLNumber=15	thisUpdate=jour 15 nextUpdate=jour 16 cRLNumber=15 BaseCRLNumber= 8
16	non émis	thisUpdate=jour 16 nextUpdate=jour 17 crlNumber=16 BaseCRLNumber=15	thisUpdate=jour 16 nextUpdate=jour 18 crlNumber=16	thisUpdate=jour 16 nextUpdate=jour 17 crlNumber=16 BaseCRLNumber=14	non émis	thisUpdate=jour 16 nextUpdate=jour 17 cRLNumber=16 BaseCRLNumber= 9

Annexe D

Exemples de définition de politique de privilège et d'attribut de privilège

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

D.1 Introduction

La politique de privilège définit d'une manière précise, pour la gestion de privilège, dans quelles conditions un vérificateur de privilège doit conclure qu'un ensemble de privilèges présenté est suffisant pour qu'il décide d'accorder un accès au déclarant de privilège (pour la demande d'objet, de ressource ou d'application). Une spécification formelle de la politique de privilège peut aider un vérificateur de privilège à effectuer une évaluation automatisée des privilèges d'un déclarant de privilège en fonction de la sensibilité de la ressource demandée, du fait qu'elle formule les règles d'acceptation ou de rejet d'une demande du déclarant en tenant compte du privilège de ce dernier et de la sensibilité de la ressource.

Comme il est nécessaire de garantir l'intégrité de la politique de privilège utilisée pour une telle détermination, il est possible de véhiculer dans des objets signés un identificateur de la politique de privilège sous la forme d'un identificateur d'objet et d'un hachage de la totalité de la politique de privilège; ces informations peuvent être stockées dans des entrées d'annuaire. La présente Spécification ne normalise toutefois aucune syntaxe particulière pouvant être utilisée pour la définition d'une instance de politique de privilège.

D.2 Exemples de syntaxes

Une politique de privilège peut être définie en utilisant toute syntaxe, y compris un texte ordinaire. Afin de fournir aux personnes qui définissent des politiques de privilège une aide dans leur compréhension des diverses options de définitions, la présente annexe fournit deux exemples de syntaxe utilisables à ces fins. Il est nécessaire d'insister sur le fait qu'il s'agit uniquement d'exemples et que l'utilisation de certificats d'attribut ou d'extensions **subjectDirectoryAttributes** de certificats de clé publique n'est pas nécessaire pour la prise en charge de ces syntaxes ou de toute autre syntaxe particulière.

D.2.1 Premier exemple

La syntaxe ASN.1 qui suit est un exemple détaillé et souple d'outil de définition de politique de privilège.

```
PrivilegePolicySyntax       ::= SEQUENCE {
    version         Version,
    ppe             PrivPolicyExpression }

PrivPolicyExpression        ::= CHOICE {
    ppPredicate     [0] PrivPolicyPredicate,
    and             [1] SET SIZE (2..MAX) OF PrivPolicyExpression,
    or              [2] SET SIZE (2..MAX) OF PrivPolicyExpression,
    not             [3] PrivPolicyExpression,
    orderedPPE      [4] SEQUENCE OF PrivPolicyExpression }
    -- Note: le composant "sequence" définit l'ordre dans lequel les privilèges
    -- doivent être examiné dans le temps

PrivPolicyPredicate         ::= CHOICE {
    present                 [0] PrivilegeIdentifier,
    equality                [1] PrivilegeComparison, -- privilège avec valeur unique ou ensemble
    greaterOrEqual          [2] PrivilegeComparison, -- privilège avec valeur unique
    lessOrEqual             [3] PrivilegeComparison, -- privilège avec valeur unique
    subordinate             [4] PrivilegeComparison, -- privilège avec valeur unique
    substrings              [5] SEQUENCE {      -- privilège avec valeur unique
        type                        PrivilegeType,
        initial                 [0] PrivilegeValue OPTIONAL,
        any                     [1] SEQUENCE OF PrivilegeValue,
        final                   [2] PrivilegeValue OPTIONAL },
subsetOf                    [6] PrivilegeComparison, -- privilège avec valeur prise dans un ensemble
supersetOf                  [7] PrivilegeComparison, -- privilège avec valeur prise dans un ensemble
nonNullSetInter             [8] PrivilegeComparison, -- privilège avec valeur prise dans un ensemble
approxMatch                 [9] PrivilegeComparison,
-- privilège avec valeur unique ou prise dans un ensemble (approximation définie par une application)
```

```
extensibleMatch          [10] SEQUENCE {
      matchingRule            OBJECT IDENTIFIER,
      inputs                  PrivilegeComparison } }

PrivilegeComparison   ::= CHOICE {
   explicit           [0] Privilege,
   -- la ou les valeurs d'un privilège externe identifié par l'expression
   -- Privilege.privilegeId sont comparées avec la ou les valeurs
   -- fournies de manière explicite dans l'expression Privilege.privilegeValueSet
   byReference        [1] PrivilegeIdPair }
   -- la ou les valeurs d'un privilège externe identifié par l'expression
   -- PrivilegeIdPair.firstPrivilege sont comparées avec
   -- la ou les valeurs d'un autre privilège externe identifié par l'expression
   -- PrivilegeIdPair.secondPrivilege

Privilege             ::= SEQUENCE {
type                  PRIVILEGE.&id ({SupportedPrivileges}),
values                SET SIZE (0..MAX) OF
                      PRIVILEGE.&Type ({SupportedPrivileges} {@type})
}


SupportedPrivileges PRIVILEGE     ::=  { ... }
PRIVILEGE   ::=  ATTRIBUTE
-- le privilège est comparable à l'attribut

PrivilegeIdPair     ::=  SEQUENCE {
   firstPrivilege        PrivilegeIdentifier,
   secondPrivilege       PrivilegeIdentifier }

PrivilegeIdentifier ::=  CHOICE {
   privilegeType         [0] PRIVILEGE.&id ({SupportedPrivileges}),
   xmlTag                [1] OCTET STRING,
   edifactField          [2] OCTET STRING }
-- l'identificateur de privilège étend le concept de type d'attribut à d'autres environnements
-- (par exemple, avec étiquette) tels que le langage XML ou l'échange EDIFACT

Version               ::= INTEGER { v1(0) }
```

Un exemple concret peut permettre de clarifier la création et l'utilisation de la structure **PrivilegePolicy**.

Prenons le privilège d'approbation d'une augmentation de salaire. Supposons, pour simplifier, que la politique devant être respectée stipule que seuls les cadres supérieurs et les niveaux hiérarchiques plus élevés peuvent approuver une augmentation de salaire et qu'une approbation peut être faite uniquement pour une personne de niveau inférieur (un directeur peut, par exemple, approuver une augmentation pour un cadre supérieur, mais pas pour un directeur général adjoint). Supposons qu'il existe six niveaux possibles pour cet exemple ("technicien" = 0, "cadre" = 1, "cadre supérieur = 2", "directeur" = 3, "directeur général adjoint" = 4, "directeur général" = 5).

Supposons en outre que le type d'attribut (le "privilège") indiquant le niveau hiérarchique dans un certificat d'attribut possède un identificateur d'objet *OID-C* et que le type d'attribut (la "sensibilité") identifiant un niveau hiérarchique dans l'enregistrement de données contenant le champ "salaire" à modifier possède un identificateur d'objet *OID-D* (qui sera en fait remplacé par des identificateurs d'objets réels dans une implémentation concrète). L'expression booléenne suivante définit la politique "approbation de salaire" souhaitée (la codification de cette politique dans une expression de politique de privilège est une tâche relativement simple):

ET (NON (Inférieur_ou_égal (valeur correspondant à OID-C, valeur correspondant à OID-D)) sous_ensemble_de (valeur correspondant à OID-C, { 2, 3, 4, 5 }))

Ce codage de politique indique que la position hiérarchique de l'auteur de l'approbation doit être supérieure (indiqué par l'expression "NON Inférieur_ou_égal") à celle de la personne approuvée et que la position hiérarchique de l'auteur de l'approbation doit appartenir au domaine {cadre supérieur, ..., directeur général} pour que le résultat de l'évaluation de cette expression booléenne soit égal à "Vrai". La première comparaison de privilège se fait "par référence" entre les valeurs correspondantes du type d'attribut "position hiérarchique" pour les deux entités impliquées. La deuxième comparaison de privilège est "explicite"; dans ce cas la valeur correspondant au privilège "position hiérarchique" de l'auteur de l'approbation est comparée à une liste explicite de valeurs. Dans cette situation, le vérificateur de privilège a besoin d'une expression qui codifie cette politique avec deux attributs associés à l'auteur et au sujet de l'approbation. L'attribut de l'auteur de l'approbation (qui figure dans un certificat d'attribut) peut avoir la valeur {*OID-C* 3} et l'attribut

du bénéficiaire de l'approbation (qui peut être contenu dans un enregistrement de base de données) peut avoir la valeur {*OID-D* 3}. La comparaison de la valeur du type d'attribut de l'auteur de l'approbation (égale à 3 dans cet exemple) avec la valeur d'attribut correspondant au type d'attribut du bénéficiaire de l'approbation (égale aussi à 3 dans cet exemple) fournit la valeur "Faux" pour l'expression "Non Inférieur_ou_égal", de sorte que le premier directeur se voit refuser la capacité d'approuver une augmentation de salaire pour le deuxième directeur. Si par contre l'attribut du bénéficiaire de l'approbation était égal à {*OID-D* 1}, l'autorisation serait accordée au directeur pour approuver une augmentation de salaire du cadre.

Il n'est pas difficile d'imaginer des ajouts utiles à l'expression précédente. Il est possible d'ajouter, par exemple, un troisième argument à l'expression "**et**" pour indiquer que la variable d'environnement "temps actuel" obtenue à partir de l'horloge locale et codée dans un attribut identifié par un identificateur de type d'objet *OID-E* doit appartenir à un domaine particulier spécifié explicitement dans un attribut identifié par un type d'objet *OID-F*. De la sorte, des augmentations de salaire peuvent être autorisées uniquement si les conditions ci-dessus sont satisfaites et si la demande est effectuée pendant les heures de travail.

D.2.2 Deuxième exemple

Une politique de sécurité dans sa forme la plus simple se constitue d'un ensemble de critères pour la fourniture de services de sécurité. Du point de vue du contrôle de sécurité, une politique de sécurité est un sous-ensemble d'une politique de sécurité de niveau supérieur qui définit les moyens de faire respecter des politiques de contrôle d'accès entre des initiateurs et des cibles. Les procédés de contrôle d'accès doivent permettre la communication lorsqu'une politique spécifique le permet et refuser la communication lorsqu'une politique spécifique ne le permet pas de manière explicite.

Une politique de sécurité constitue la base pour les décisions faites par les procédures de contrôle d'accès. Les informations de politique de sécurité propres au domaine sont véhiculées au moyen du fichier d'informations de politique de sécurité (SPIF, *security policy information file*).

Le fichier SPIF est un objet signé en vue d'assurer sa protection contre des modifications non autorisées. Le fichier SPIF contient des informations utilisées pour interpréter les paramètres de contrôle d'accès figurant dans l'étiquette de sécurité et l'attribut d'habilitation. L'identificateur de politique de sécurité qui figure dans l'attribut d'habilitation doit être associé à une syntaxe et à une sémantique propres à l'implémentation, telles qu'elles sont définies par la politique de sécurité. Cette syntaxe d'implémentation associée à une politique de sécurité spécifique est gérée dans un fichier SPIF.

Le fichier SPIF véhicule des équivalences entre autorisations et sensibilités à travers des domaines de politique de sécurité, fournit une représentation des étiquettes de sécurité sous forme imprimable et mappe des chaînes de caractères visibles contenant des niveaux et des catégories de sécurité à des fins de présentation aux utilisateurs finaux lors de la sélection des attributs de sécurité d'un objet de données. Les mappages sont exprimés de manière à ce qu'une étiquette générée dans un domaine de politique de sécurité puisse être interprétée correctement dans un autre domaine de politique de sécurité. Le fichier SPIF peut également mapper l'attribut d'habilitation avec les champs du message de sécurité et les étiquettes de présentation qui sont affichés à destination de l'utilisateur. S'il s'applique, ce mappage vérifie que le destinataire prévu dispose des autorisations adéquates pour accepter l'objet de données.

Un fichier SPIF contient une succession des champs suivants:

- **versionInformation** [*informations de version*]: indique la version de la syntaxe ASN.1.
- **updateInformation** [*informations de mise à jour*]: indique la version de la syntaxe et de la sémantique de la spécification de fichier SPIF.
- **securityPolicyIdData** [*données d'identification de politique de sécurité*]: identifie la politique de sécurité à laquelle s'applique le fichier SPIF.
- **privilegeId** [*identificateur de privilège*]: indique l'identificateur d'objet qui caractérise la syntaxe contenue dans la catégorie de sécurité d'attribut d'habilitation.
- **rbacId** [*identificateur rbac*]: identificateur d'objet qui caractérise la syntaxe de la catégorie de sécurité qui est utilisée conjointement au fichier SPIF.
- **securityClassifications** [*classifications de sécurité*]: mappe la classification de l'étiquette de sécurité et la classification de l'attribut d'habilitation et fournit également des mappages pour des équivalences.
- **securityCategoryTagSets** [*ensembles d'étiquettes de catégorie de sécurité*]: mappe les catégories de sécurité de l'étiquette de sécurité avec les catégories de sécurité de l'attribut d'habilitation et fournit également des mappages pour des équivalences.

- **equivalentPolicies** [*politiques équivalentes*]: consolide toutes les politiques équivalentes dans le fichier SPIF.
- **defaultSecurityPolicyIdData** [*données d'identificateur de politique de sécurité par défaut*]: indique la politique de sécurité qui s'appliquera lorsque des données sont reçues sans étiquette de sécurité.
- **extensions**: fournit un procédé permettant d'inclure des fonctionnalités supplémentaires lorsque de nouveaux besoins sont mis en évidence.

Le fichier d'informations de politique de sécurité est défini par la syntaxe suivante:

```
SecurityPolicyInformationFile ::= SIGNED {SPIF}
SPIF ::= SEQUENCE    {
    versionInformation              VersionInformationData DEFAULT v1,
    updateInformation               UpdateInformationData,
    securityPolicyIdData                    ObjectIdData,
    privilegeId                             OBJECT IDENTIFIER,
    rbacId                                  OBJECT IDENTIFIER,
    securityClassifications         [0]     SEQUENCE OF SecurityClassification OPTIONAL,
    securityCategories              [1]     SEQUENCE OF SecurityCategory  OPTIONAL,
    equivalentPolicies              [2]     SEQUENCE OF EquivalentPolicy  OPTIONAL,
    defaultSecurityPolicyIdData     [3]     ObjectIdData OPTIONAL,
    extensions                      [4]     Extensions OPTIONAL }

VersionInformationData ::= INTEGER { v1(0) }

UpdateInformationData  ::= SEQUENCE    {
    sPIFVersionNumber               INTEGER,
    creationDate                    GeneralizedTime,
    originatorDistinguishedName     Name,
    keyIdentifier                   OCTET STRING OPTIONAL }

ObjectIdData ::=   SEQUENCE  {
    objectId            OBJECT IDENTIFIER,
    objectIdName        DirectoryString {ubObjectIdNameLength} }

SecurityClassification  ::=  SEQUENCE  {
    labelAndCertValue                   INTEGER,
    classificationName                  DirectoryString {ubClassificationNameLength},
    equivalentClassifications   [0]     SEQUENCE OF EquivalentClassification OPTIONAL,
    hierarchyValue                      INTEGER,
    markingData                 [1]     SEQUENCE OF MarkingData OPTIONAL,
    requiredCategory            [2]     SEQUENCE OF OptionalCategoryGroup OPTIONAL,
    obsolete                            BOOLEAN DEFAULT FALSE           }

EquivalentClassification ::= SEQUENCE {
    securityPolicyId            OBJECT IDENTIFIER,
    labelAndCertValue           INTEGER,
    applied                     INTEGER {
                                    encrypt   (0),
                                    decrypt   (1),
                                    both      (2) } }

MarkingData ::= SEQUENCE {
    markingPhrase       DirectoryString {ubMarkingPhraseLength} OPTIONAL,
    markingCodes        SEQUENCE OF MarkingCode OPTIONAL }

MarkingCode ::= INTEGER {
    pageTop             (1),
    pageBottom          (2),
    pageTopBottom       (3),
    documentEnd         (4),
    noNameDisplay       (5),
    noMarkingDisplay    (6),
    unused              (7),
    documentStart       (8),
    suppressClassName   (9)}
```

```
OptionalCategoryGroup ::= SEQUENCE {
    operation               INTEGER {
                                onlyOne         (1),
                                oneOrMore       (2),
                                all             (3)},
    categoryGroup           SEQUENCE OF OptionalCategoryData }

OptionalCategoryData ::= SEQUENCE {
    optCatDataId            OC-DATA.&id({CatData}),
    categorydata            OC-DATA.&Type({CatData}{@optCatDataId }) }

OC-DATA ::= TYPE-IDENTIFIER

CatData OC-DATA ::= { … }

EquivalentPolicy  ::= SEQUENCE {
    securityPolicyId        OBJECT IDENTIFIER,
    securityPolicyName      DirectoryString {ubObjectIDNameLength}
    OPTIONAL}

Extensions ::= SEQUENCE OF Extension

Extension ::= SEQUENCE {
    extensionId             EXTENSION.&objId ({ExtensionSet}),
    critical                BOOLEAN DEFAULT FALSE,
    extensionValue              OCTET STRING  }
```

Il convient de noter que la syntaxe de l'exemple de fichier SPIF est évolutive et que la définition et la description complète de chaque élément se trouvent dans la Rec. UIT-T X.841 | ISO/CEI 15816.

D.3 Exemple d'attribut de privilège

L'exemple qui suit présente un attribut utilisé pour véhiculer un privilège donné. Il est fourni uniquement à titre d'illustration; la spécification effective de cette syntaxe et l'attribut associé sont décrits dans le paragraphe 17.5 de la Rec. UIT-T X.501 | ISO/CEI 9594-2. Cet attribut véhicule une habilitation qui peut être associée à une autorité nommée, par exemple dans le cas d'un agent DUA souhaitant communiquer avec un agent DSA.

Un attribut d'habilitation associe une habilitation avec une entité nommée englobant des agents DUA.

```
clearance  ATTRIBUTE ::=  {
    WITH SYNTAX             Clearance
    ID                      id-at-clearance }

Clearance  ::=  SEQUENCE {
    policyId                OBJECT IDENTIFIER,
    classList               ClassList DEFAULT {unclassified},
    securityCategories      SET SIZE (1MAX) OF SecurityCategory  OPTIONAL}

ClassList  ::=  BIT STRING {
    unmarked        (0),
    unclassified    (1),
    restricted      (2),
    confidential    (3),
    secret          (4),
    topSecret       (5) }
```

Les composants individuels sont décrits avec les spécifications effectives de ce privilège dans le document référencé.

Annexe E

Introduction à la cryptographie avec clé publique[2)]

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

Dans les systèmes cryptographiques classiques, la clé utilisée par l'expéditeur d'un message secret pour effectuer le chiffrement est la même que celle qui est utilisée par le destinataire légitime pour effectuer le déchiffrement.

Dans les systèmes cryptographiques avec clé publique (PKCS, *public key cryptosystems*), cependant, les clés vont par paires, l'une des clés étant utilisée pour le chiffrement et l'autre pour le déchiffrement. Chaque paire de clés est associée à un utilisateur particulier X. L'une des clés, connue sous le nom de clé publique (Xp) est connue publiquement et peut être employée par n'importe quel utilisateur pour chiffrer les données. Seul X qui possède la clé privée complémentaire (Xs) peut déchiffrer les données. (Cela est représenté par la notation D = Xs[Xp[D]]). Il est impossible de découvrir par un calcul la clé privée à partir de la connaissance de la clé publique. Tout utilisateur peut ainsi communiquer un élément d'information que seul X peut découvrir en le chiffrant au moyen de Xp. Par extension, deux utilisateurs peuvent communiquer en secret en utilisant l'un et l'autre la clé publique pour chiffrer les données comme indiqué dans la Figure E.1.

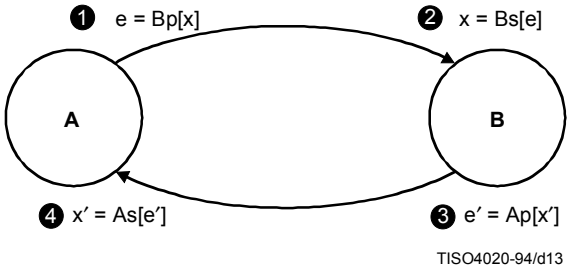


Figure E.1 – Utilisation d'un système PKCS pour l'échange d'informations secrètes

L'utilisateur A possède une clé publique Ap et une clé privée As; l'utilisateur B possède un autre jeu de clés, Bp et Bs. A et B connaissent tous les deux les clés publiques de l'un et de l'autre, mais ignorent la clé privée de l'autre partie. A et B ont donc la possibilité d'échanger des informations secrètes en accomplissant les opérations ci-après (représentées dans la Figure E.1):

1) A désire envoyer une certaine information secrète x à B. A chiffre donc x avec la clé de chiffrement de B et envoie l'information chiffrée e à B. Cela est représenté par:

$$e = Bp[x]$$

2) B peut maintenant déchiffrer cette information chiffrée e pour obtenir l'information x au moyen de la clé privée de déchiffrement Bs. Il convient de noter que B est le seul possesseur de Bs et, étant donné que cette clé ne peut jamais être découverte ou envoyée, il est impossible à une autre partie d'obtenir l'information x. La possession de Bs détermine l'identité de B. L'opération de déchiffrement est représentée par:

$$x = Bs[e], \text{ ou } x = Bs[Bp[x]]$$

---

2) Pour complément d'information, consulter:

DIFFIE (W.) et HELLMAN (M.E.): New Directions in Cryptography, *IEEE Transactions on Information Theory*, IT-22 Nº 6, novembre 1976.

3) B peut maintenant envoyer de même une certaine information secrète x′ à A avec la clé de chiffrement Ap de A:

$$e' = Ap[x']$$

4) A obtient x′ par déchiffrement de e′:

$$x' = As[e'], \text{ ou } x' = As[Ap[x']]$$

Par ce moyen, A et B ont échangé les informations secrètes x et x′. Ces informations ne peuvent être obtenues par personne d'autre que A et B du moment que leurs clés privées ne sont pas révélées.

Alors qu'un tel échange transfère l'information secrète entre deux parties, il peut aussi servir à vérifier leurs identités. Plus précisément, A et B sont respectivement identifiés par la possession de leurs clés privées de déchiffrement As et Bs. A peut déterminer si B est en possession de la clé privée de déchiffrement Bs par l'obtention de la partie x de son message envoyé en retour dans le message x′ de B. Cela indique à A que la communication est établie avec le possesseur de Bs. B peut de même vérifier l'identité de A.

Certains systèmes PKCS possèdent la propriété que leurs opérations de déchiffrement et de chiffrement peuvent être inversées de façon à avoir D = Xp[Xs[D]]. Ainsi, un élément d'information qui ne pourrait avoir été expédié que par X, pourra être lisible par tout autre utilisateur (qui est en possession de la clé Xp). Cela peut donc être utilisé pour certifier l'origine de l'information et sert de base aux signatures numériques. Seuls les systèmes PKCS qui possèdent cette propriété de permutabilité peuvent être utilisés dans le présent cadre d'authentification. Un algorithme de ce type est décrit dans l'Annexe D.

Annexe F

Définition de référence des identificateurs d'objet d'algorithme

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

La présente annexe définit des identificateurs d'objet assignés aux algorithmes d'authentification et de chiffrement, en l'absence d'un registre formel de consignation. Il est prévu d'utiliser un tel registre dès qu'il sera disponible. Les définitions prennent la forme d'un module ASN.1 appelé **AlgorithmObjectIdentifiers** [*identificateurs d'objet d'algorithme*].

```
AlgorithmObjectIdentifiers {joint-iso-itu-t ds(5) module(1) algorithmObjectIdentifiers(8) 4}
DEFINITIONS ::=
BEGIN

-- EXPORTER tout --

-- Les types et les valeurs définis dans ce module sont exportés afin d'être utilisés dans les autres
-- modules ASN.1 contenus dans les Spécifications d'annuaire, et dans d'autres applications
-- qui s'en serviront pour accéder au service d'annuaire. D'autres applications pourront les utiliser à leurs
-- propres fins, mais ces utilisations n'imposeront pas de contraintes aux extensions et modifications
-- nécessaires pour tenir à jour ou améliorer le service d'annuaire.

IMPORTS
    algorithm, authenticationFramework
        FROM UsefulDefinitions {joint-iso-itu-t ds(5) module(1) usefulDefinitions(0) 4}
    ALGORITHM
        FROM AuthenticationFramework authenticationFramework      ;

-- catégorie d'identificateurs d'objet --

encryptionAlgorithm    OBJECT IDENTIFIER    ::=    {algorithm 1}
hashAlgorithm          OBJECT IDENTIFIER    ::=    {algorithm 2}
signatureAlgorithm     OBJECT IDENTIFIER    ::=    {algorithm 3}

-- synonymes --

id-ea   OBJECT IDENTIFIER ::=   encryptionAlgorithm
id-ha   OBJECT IDENTIFIER ::=   hashAlgorithm
id-sa   OBJECT IDENTIFIER ::=   signatureAlgorithm

-- algorithmes --

rsaALGORITHM    ::=   {
    KeySize
    IDENTIFIED BYid-ea-rsa }

KeySize         ::=   INTEGER

-- attribution d'identificateurs d'objet --

id-ea-rsa       OBJECT IDENTIFIER    ::=    {id-ea 1}

-- l'attribution des identificateurs d'objet suivants réserve les valeurs assignées aux fonctions déconseillées

id-ha-sqMod-n                OBJECT IDENTIFIER    ::=        {id-ha 1}
id-sa-sqMod-nWithRSA         OBJECT IDENTIFIER    ::=        {id-sa 1}

END
```

Annexe G

Exemples d'utilisation de contraintes d'itinéraire de certification

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

G.1 Exemple 1: utilisation de contraintes de base

Supposons que la Société Widget veuille certifier en retour l'autorité de certification centrale du Groupe industriel Acme mais souhaite que la communauté Widget n'utilise que les certificats d'entité finale émis par cette autorité de certification et non pas les certificats émis par d'autres autorités de certification, certifiés par cette autorité de certification.

La Société Widget peut répondre à cette prescription en émettant un certificat pour l'autorité de certification centrale d'Acme, contenant la valeur suivante de champ d'extension:

Valeur du champ de contraintes de base:

{ cA TRUE, pathLenConstraint 0 }

G.2 Exemple 2: utilisation de contraintes nominatives

Supposons que la Société Widget veuille certifier en retour l'autorité de certification centrale du Groupe industriel Acme mais souhaite que la communauté Widget n'utilise que les certificats Acme pour les entités qui répondent aux critères suivants:

- dans le groupe Acme, Inc. (Etats-Unis), toutes les entités sont acceptables, sauf celles du département des achats;
- dans la société EuroAcme (France), seules les entités immédiatement dépendantes du siège d'EuroAcme sont acceptables (ce qui inclut les individus qui rendent compte directement au siège mais exclut ceux qui rendent compte à des filiales);
- dans la société Acme Ltd. (Royaume-Uni), toutes les entités sont acceptables, sauf celles qui rendent compte à des filiales du département d'organisation de la recherche et du développement (ce qui inclut les individus qui rendent compte directement à ce département mais exclut ceux qui rendent compte à des filiales de ce département).

La Société Widget peut répondre à ces prescriptions en émettant un certificat pour l'autorité de certification centrale d'Acme, contenant les valeurs de champ d'extension suivantes:

Valeur du champ de contraintes de base:

{ cA TRUE }

Valeur du champ de contraintes nominatives:

{ permittedSubtrees {{base -- *Country=US, Org=Acme Inc* **--},**

{base -- *Country=France, Org=EuroAcme* **--, maximum 1},**

{base -- *Country=UK, Org=Acme Ltd* **--}},**

excludedSubtrees {{base --*Country=US, Org=Acme Inc, Org. Unit=Purchasing***-},**

{base --*Country=UK Org=Acme Ltd., Org. Unit=R&D***--, minimum 2}}}**

G.3 Exemple 3: utilisation de mappage de politiques et de contraintes de politiques

Supposons que le scénario suivant de certification réciproque soit requis entre les gouvernements du Canada et des Etats-Unis d'Amérique:

a) une autorité de certification du gouvernement canadien souhaite certifier l'utilisation de signatures du gouvernement des Etats-Unis en fonction d'une politique canadienne dite *Can/US-Trade*;

b) le gouvernement des Etats-Unis a une politique dite *US/Can-Trade*, que le gouvernement canadien est disposé à considérer comme équivalente à sa propre politique *Can/US-Trade*;

c) le gouvernement canadien souhaite appliquer des mesures de sauvegarde prescrivant que tous les certificats des Etats-Unis doivent déclarer explicitement la prise en charge de la politique et interdisant le mappage avec d'autres politiques dans le domaine des Etats-Unis.

Une autorité de certification canadienne peut émettre un certificat en faveur d'une autorité de certification des Etats-Unis, contenant les valeurs de champ d'extension suivantes:

Valeur du champ de politiques de certification:

{{ policyIdentifier -- *identificateur d'objet pour Can/US-Trade* -- **}}**

Valeur du champ de mappage de politiques:

{{ issuerDomainPolicy -- *identificateur d'objet pour Can/US-Trade* -- ,

subjectDomainPolicy -- *identificateur d'objet pour US/Can-Trade* -- **}}**

Valeur du champ de contraintes de politiques:

{{ policySet { -- *identificateur d'objet pour Can/US-Trade* -- **}, requireExplicitPolicy (0),**

inhibitPolicyMapping (0)}}

Annexe H

Liste alphabétique des définitions des éléments d'information

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

Cette annexe fournit un index alphabétique des définitions de format de certificat et de liste CRL, des extensions de certificats, des classes d'objets, des formes de nom, des types d'attribut et des règles de concordance définis dans cette Spécification d'annuaire.

Elément	Paragraphe
Formats de certificat et de liste CRL	
Structure du certificat d'attribut	12.1
Liste de révocation de certificat	7.3
Clés publiques et certificats de clé publique	7
Extensions de certificat, de liste CRL et d'élément de liste CRL	
Extension de politiques de certificat acceptable	15.5.2.3
Extension de politiques de privilège acceptable	15.1.2.4
Extension de descripteur d'attribut	15.3.2.2
Extension d'identificateur d'autorité d'attribut	15.5.2.4
Extension d'identificateur de clé d'autorité	8.2.2.1
Extension de mise à jour de base	8.6.2.5
Extension de contraintes d'attribut de base	15.5.2.1
Extension de contraintes de base	8.4.2.1
Extension d'émetteur de certificat	8.6.2.3
Extension de politiques de certificat	8.2.2.6
Extension de point de répartition de liste CRL	8.6.2.1
Extension de numéro de liste CRL	8.5.2.1
Extension de domaine d'application de liste CRL	8.5.2.5
Extension d'identificateur de flux de liste CRL	8.5.2.7
Extension de contraintes de nom délégué	15.5.2.2
Extension d'indicateur de liste CRL delta	8.6.2.4
Extensions d'informations delta	8.5.2.9
Extension d'utilisation de clé étendue	8.2.2.4
Extension de liste CRL la plus récente	8.6.2.6
Extension de code d'instruction de mise en attente	8.5.2.3
Extension d'inhibition de toute politique	8.4.2.4
Extension de date de non validité	8.5.2.4
Extension d'autre nom d'émetteur	8.3.2.2
Extension de point de répartition émetteur	8.6.2.2
Extension d'utilisation de clé	8.2.2.3
Extension de contraintes de nom	8.4.2.2
Extension d'absence d'informations de révocation	15.2.2.2
Extension de liste ordonnée	8.5.2.8
Extension de contraintes de politique	8.4.2.3
Extension de mappages de politique	8.2.2.7
Extension de durée d'utilisation de clé privée	8.2.2.5

Elément	Paragraphe
Extension de code motif	8.5.2.2
Extension d'identificateur de certificat de spécification de rôle	15.4.2.1
Extension d'identificateur de source d'autorité	15.3.2.1
Extension de référence de statut	8.5.2.6
Extension d'autre nom de sujet	8.3.2.1
Extension d'identificateur de clé publique de sujet	8.2.2.2
Extension d'attributs d'annuaire du sujet	8.3.2.3
Extension d'informations de cible	15.1.2.2
Extension de spécification de durée	15.1.2.1
Extension de notification d'utilisateur	15.1.2.3
Classes d'objets et formes de nom	
Classe d'objets "certificat d'attribut de point de répartition de liste CRL"	17.1.4
Classe d'objets "politique de certificat et de déclaration CPS"	11.1.5
Classe d'objets et forme de nom de points de répartition de liste CRL	11.1.3
Classe d'objets "liste CRL delta"	11.1.4
Classe d'objets "autorité de certification d'infrastructure PKI"	11.1.2
Classe d'objets "itinéraire de certificat d'infrastructure PKI"	11.1.6
Classe d'objets "utilisateur d'infrastructure PKI"	11.1.1
Classe d'objets "autorité d'attribut d'infrastructure PMI"	17.1.2
Classe d'objets "itinéraire de délégation d'infrastructure PMI"	17.1.5
Classe d'objets "source d'autorité d'infrastructure PMI"	17.1.3
Classe d'objets "utilisateur d'infrastructure PMI"	17.1.1
Classe d'objets "politique de privilège"	17.1.6
Attributs d'annuaire	
Attribut "certificats d'autorité d'attribut"	17.2.2
Attribut "liste de révocation de certificat d'autorité d'attribut"	17.2.5
Attribut "certificat d'attribut"	17.2.1
Attribut "liste de révocation de certificat d'attribut"	17.2.4
Attribut "certificat de descripteur d'attribut"	17.2.3
Attribut "liste de révocation d'autorité"	11.2.5
Attribut "certificat d'autorité de certification"	11.2.2
Attribut "déclaration de pratique de certification"	11.2.8
Attribut "politique de certificat"	11.2.9
Attribut "liste de révocation de certificat"	11.2.4
Attribut "paire de certificats croisés"	11.2.3
Attribut "itinéraire de délégation"	17.2.6
Attribut "liste delta de révocation"	11.2.6
Attribut "itinéraire d'infrastructure PKI"	11.2.10
Attribut "politique de privilège"	17.2.7
Attribut "algorithmes pris en charge"	11.2.7
Attribut "certificat d'utilisateur"	11.2.1
Règles de concordance	
Concordance d'identificateur d'autorité d'attribut	15.5.2.4.1
Concordance de politiques de certificat acceptable	15.5.2.3.1

Elément	Paragraphe
Concordance d'identificateur d'algorithme	11.3.7
Concordance exacte de certificat d'attribut	17.3.1
Concordance de certificat d'attribut	17.3.2
Concordance de descripteur d'attribut	15.3.2.2.1
Concordance de contraintes d'attribut de base	15.5.2.1.1
Concordance exacte de certificat	11.3.1
Concordance exacte de liste de certificats	11.3.5
Concordance de liste de certificats	11.3.6
Concordance de certificat	11.3.2
Concordance exacte de paire de certificats	11.3.3
Concordance de paire de certificats	11.3.4
Concordance de contraintes de nom délégué	15.5.2.2.1
Concordance d'itinéraire de délégation	17.3.4
Concordance détenteur/émetteur	17.3.3
Concordance d'itinéraire PKI	11.3.9
Concordance de politique	11.3.8
Concordance d'identificateur de certificat de spécification de rôle	15.4.2.1.1
Spécification de concordance de durée	15.1.2.1.1

Annexe I

Amendements et corrigenda

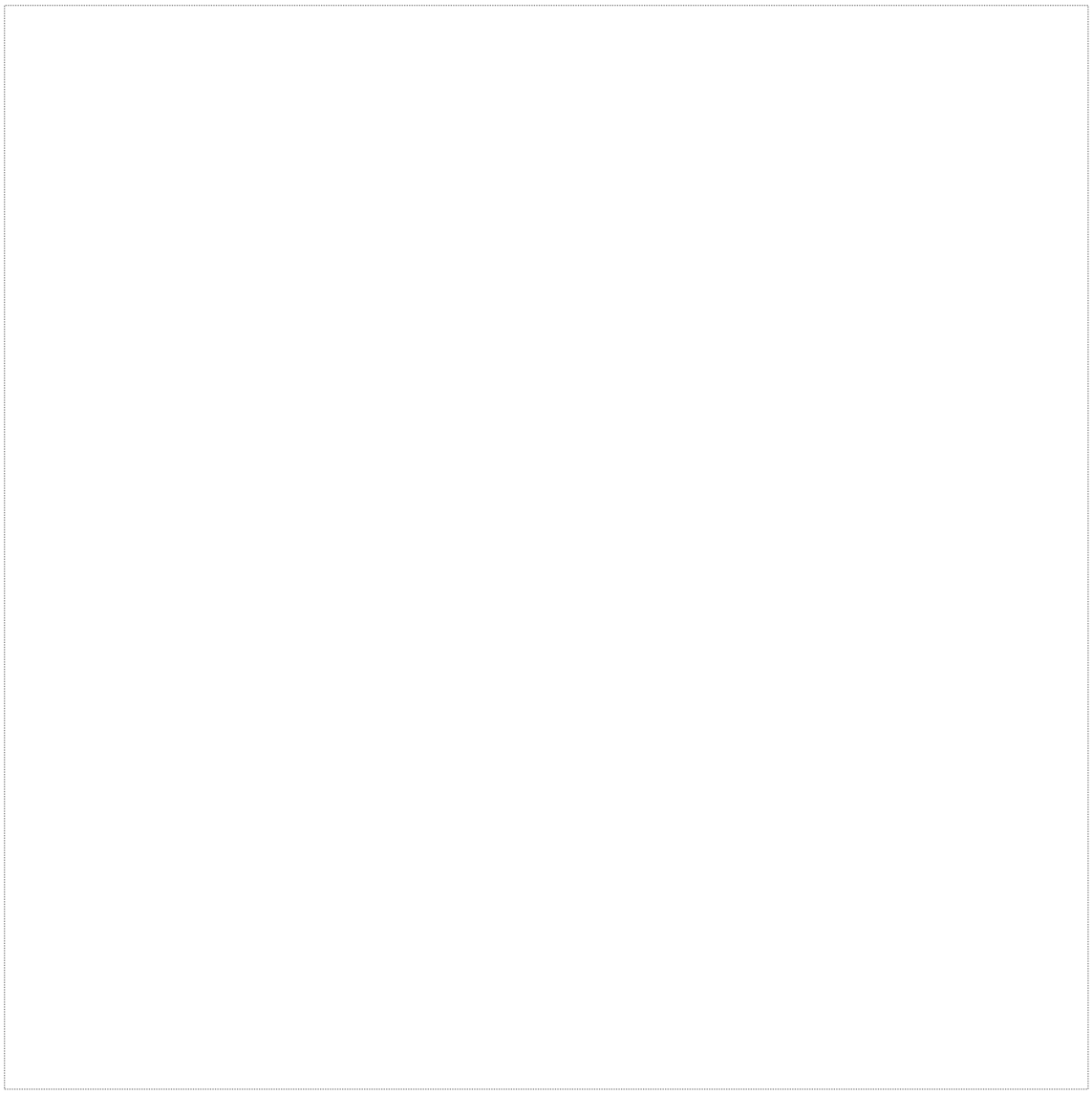
(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

Cette édition de la présente Spécification d'annuaire comprend les modification suivantes::

- Amendement 1 pour les extensions de certificat;

Cette édition de la présente Spécification d'annuaire comprend les corrigenda techniques suivants, concernant les erreurs signalées dans les relevés d'erreurs suivants (certaines parties des corrigenta techniques suivants peuvent avoir été rendues caduques par les amendements qui ont formé la présente édition de cette Spécification d'annuaire):

- Corrigendum technique 1 (reprenant les Relevés d'erreurs 183 et 194);
- Corrigendum technique 3 (reprenant les Relevés d'erreurs 200, 201, 212, 213, 218, et 220);
- Corrigendum technique 4 (reprenant les Relevés d'erreurs 185);
- Corrigendum technique 5 (reprenant les Relevés d'erreurs 204);
- Corrigendum technique 7 (reprenant les Relevés d'erreurs 222).



SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	**Réseaux de données et communication entre systèmes ouverts**
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects informatiques généraux des systèmes de télécommunication