Remplacée par une version plus récente

UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

# UIT-T X.509

(08/97)

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

SÉRIE X: RÉSEAUX POUR DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS

Annuaire

---

# Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: cadre d'authentification

Recommandation UIT-T X.509

Remplacée par une version plus récente

(Antérieurement Recommandation du CCITT)

---

Remplacée par une version plus récente

RECOMMANDATIONS UIT-T DE LA SÉRIE X

**RÉSEAUX POUR DONNÉES ET COMMUNICATION ENTRE SYSTÈMES OUVERTS**

| | |
|---|---|
| RÉSEAUX PUBLICS POUR 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.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.*

Remplacée par une version plus récente

**NORME INTERNATIONALE 9594-8**

**RECOMMANDATION UIT-T X.509**

# TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES OUVERTS – L'ANNUAIRE: CADRE D'AUTHENTIFICATION


## Résumé

La présente Recommandation | Norme internationale définit un cadre de travail pour la fourniture par l'Annuaire de services d'authentification 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 renforcée, faisant intervenir des justificatifs codés par des techniques cryptographiques. Si l'authentification simple offre une certaine protection (limitée) contre les accès non autorisés, seule l'authentification renforcée doit être utilisée pour la fourniture de services sûrs.


## Source

La Recommandation X.509 de l'UIT-T a été approuvée le 9 août 1997. Un texte identique est publié comme Norme internationale ISO/CEI 9594-8.

Remplacée par une version plus récente

## 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'études 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 n° 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, le terme *exploitation reconnue (ER)* désigne tout particulier, toute entreprise, toute société ou tout organisme public qui exploite un service de correspondance publique. Les termes *Administration*, *ER* et *correspondance publique* sont définis dans la *Constitution de l'UIT (Genève, 1992)*.

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 n'avait pas 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.

© UIT 1999

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

Remplacée par une version plus récente

## TABLE DES MATIÈRES


*Page*

SECTION 1 – GÉNÉRALITÉS ..... 1

1 Domaine d'application ..... 1

2 Références normatives ..... 1

2.1 Recommandations | Normes internationales identiques ..... 1

2.2 Paires de Recommandations | Normes internationales équivalentes par leur contenu technique ..... 1

2.3 Autres références ..... 1

3 Définitions ..... 1

3.1 Définitions relatives à l'architecture de sécurité du modèle de référence OSI ..... 1

3.2 Définitions relatives au modèle d'Annuaire ..... 1

3.3 Définitions relatives au cadre d'authentification ..... 1

4 Abréviations ..... 1

5 Conventions ..... 1

SECTION 2 – AUTHENTIFICATION SIMPLE ..... 1

6 Procédure d'authentification simple ..... 1

6.1 Production de l'information d'identification protégée ..... 1

6.2 Procédure d'authentification simple protégée ..... 1

6.3 Type d'attribut de mot de passe d'utilisateur ..... 1

SECTION 3 – AUTHENTIFICATION RENFORCÉE ..... 1

7 Base de l'authentification renforcée ..... 1

8 Obtention d'une clé publique d'utilisateur ..... 1

8.1 Optimisation de la quantité d'information obtenue de l'Annuaire ..... 1

8.2 Exemple ..... 1

9 Signatures numériques ..... 1

10 Procédures d'authentification renforcée ..... 1

10.1 Aperçu général ..... 1

10.2 Authentification à une voie ..... 1

10.3 Authentification à deux voies ..... 1

10.4 Authentification à trois voies ..... 1

11 Gestion des clés et des certificats ..... 1

11.1 Production de paires de clés ..... 1

11.2 Gestion des certificats ..... 1

12 Extensions de certificat et de liste CRL ..... 1

12.1 Introduction ..... 1

12.2 Informations de clé et de politique ..... 1

12.2.1 Prescriptions ..... 1

12.2.2 Champs des extensions de certificat et de liste CRL ..... 1

12.3 Attributs d'entité titulaire de certificat et d'émetteur de certificat ..... 1

12.3.1 Prescriptions ..... 1

12.3.2 Champs d'extension de certificat ou de liste CRL ..... 1

12.4 Contraintes sur le chemin de certification ..... 1

12.4.1 Prescriptions ..... 1

12.4.2 Champs d'extension de certificat ..... 32

12.4.3 Procédure de traitement du chemin de certification ..... 35

12.5 Extensions de base des listes CRL ..... 37

12.5.1 Prescriptions ..... 37

12.5.2 Champs d'extension de liste et d'entrée de liste CRL ..... 38

Remplacée par une version plus récente

*Page*

12.6 Points de répartition de listes CRL et listes CRL delta ..... 40
12.6.1 Prescriptions ..... 40
12.6.2 Champs d'extension de certificat ..... 41
12.6.3 Champs d'extension de liste CRL et d'entrée de liste CRL ..... 42
12.6.4 Type d'attribut pour listes CRL delta ..... 44
12.7 Règles de correspondance ..... 44
12.7.1 Correspondance exacte de certificat ..... 44
12.7.2 Correspondance de certificat ..... 44
12.7.3 Correspondance exacte d'une paire de certificats ..... 45
12.7.4 Correspondance d'une paire de certificats ..... 45
12.7.5 Correspondance exacte de listes de certificats ..... 46
12.7.6 Correspondance de listes de certificats ..... 46
12.7.7 Correspondance d'identificateurs d'algorithme ..... 47

13 Obtention d'attributs certifiés ..... 47
13.1 Certificats d'attribut ..... 47
13.2 Attribut de certificat d'attribut ..... 48
13.3 Règle de correspondance des certificats d'attribut ..... 48
13.4 Chemins de certificat d'attribut ..... 48
13.5 Liste d'annulations des certificats d'attribut ..... 49

Annexe A – Cadre d'authentification en ASN.1 ..... 50

Annexe B – Prescriptions de sécurité ..... 60
B.1 Dangers ..... 60
B.2 Services de sécurité ..... 60
B.3 Mécanismes de sécurité ..... 61
B.4 Dangers contre lesquels la protection est assurée par les services de sécurité ..... 62
B.5 Négociation des services et des mécanismes de sécurité ..... 62

Annexe C – Introduction à la cryptographie de clé publique ..... 63

Annexe D – Le système cryptographique RSA de clé publique ..... 65
D.1 Objectif et domaine d'application ..... 65
D.2 Définitions ..... 65
D.3 Symboles et abréviations ..... 65
D.4 Description ..... 66
D.5 Prescriptions de sécurité ..... 66
D.5.1 Longueurs de clé ..... 66
D.5.2 Production de clés ..... 66
D.6 Exposant public ..... 66
D.7 Conformité ..... 67

Annexe E – Fonctions de hachage ..... 68
E.1 Prescriptions pour les fonctions de hachage ..... 68

Annexe F – Dangers contre lesquels la protection est assurée par les méthodes d'authentification renforcée ..... 69

Annexe G – Confidentialité des données ..... 70
G.1 Introduction ..... 70
G.2 Confidentialité des données par chiffrement asymétrique ..... 70
G.3 Confidentialité des données par chiffrement symétrique ..... 70

Annexe H – Définition de référence des identificateurs d'objet d'algorithme ..... 71

Annexe I – Bibliographie ..... 72

Annexe J – Exemples d'utilisation de contraintes de chemin de certification ..... 73
J.1 Exemple 1: utilisation de contraintes de base ..... 73
J.2 Exemple 2: utilisation de contraintes nominatives ..... 73
J.3 Exemple 3: utilisation de mappage de politiques et de contraintes de politiques ..... 73

Annexe K – Amendements et corrigenda ..... 75

## Introduction

La présente Recommandation | Norme internationale a été élaborée, ainsi que d'autres Recommandations | Normes internationales, pour faciliter l'interconnexion des systèmes de traitement de l'information et permettre ainsi d'assurer des services d'annuaire. L'ensemble de tous ces systèmes, avec les informations d'annuaire qu'ils contiennent, peut être considéré comme un tout intégré, appelé *Annuaire*. Les informations de l'Annuaire, appelées collectivement base d'informations d'Annuaire (DIB) sont généralement utilisées pour faciliter la communication entre, avec ou à propos d'objets tels que des entités d'application, des personnes, des terminaux et des listes de distribution.

L'Annuaire joue un rôle important dans l'interconnexion des systèmes ouverts, dont le but est de permettre, moyennant un minimum d'accords techniques en dehors des normes d'interconnexion proprement dites, l'interconnexion des systèmes de traitement de l'information:

- provenant de divers fabricants;
- gérés différemment;
- de niveaux de complexité différents;
- de générations différentes.

Un grand nombre d'applications comportent des prescriptions de sécurité pour assurer leur protection contre les dangers susceptibles de porter atteinte à la communication de l'information. L'Annexe B contient une brève description des dangers généralement connus ainsi que des services et mécanismes de sécurité qu'on peut utiliser pour la protection contre ces dangers. Presque tous les services de sécurité reposent sur la fiabilité de la connaissance des identités des parties en communication, c'est-à-dire sur leur authentification.

La présente Recommandation | Norme internationale définit un cadre de fourniture de services d'authentification par l'Annuaire à ses utilisateurs. Ces derniers utilisent l'Annuaire lui-même ainsi que d'autres applications et services. L'Annuaire peut utilement contribuer à répondre à leur besoins en services d'authentification et autres services de sécurité car c'est un emplacement naturel à partir duquel les parties en communication peuvent obtenir les informations d'authentification des unes et des autres - connaissance sur laquelle repose l'authentification. L'Annuaire est un emplacement naturel du fait qu'il contient d'autres informations nécessaires à la communication et obtenues avant que celle-ci ait lieu. L'obtention, à partir de l'Annuaire, des informations d'authentification d'un éventuel correspondant de communication est, avec cette méthode, semblable à l'obtention d'une adresse. En raison de la vaste portée de l'Annuaire aux fins des communications, on prévoit que le présent cadre d'authentification sera largement utilisé par toute une gamme d'applications.

Cette troisième édition révise techniquement et améliore, mais ne remplace pas, la deuxième édition de la présente Recommandation | Norme internationale. Les implémentations peuvent encore revendiquer la conformité à la deuxième édition mais celle-ci finira par ne plus être prise en compte (c'est-à-dire que les erreurs signalées ne seront plus corrigées). Il est recommandé que les implémentations se conforment, dès que possible, à la présente troisième édition.

Cette troisième édition spécifie les versions 1 et 2 des protocoles de l'Annuaire.

Les première et deuxième éditions spécifiaient également la version 1. La plupart des services et protocoles spécifiés dans la présente édition sont conçus pour fonctionner selon la version 1. Lors de la négociation de celle-ci, on a traité les différences entre les services et entre les protocoles, définis dans les trois éditions, en utilisant les règles d'extensibilité définies dans l'édition actuelle de la Rec. UIT-T X.519 | ISO/CEI 9594-5. Certains services et protocoles améliorés, par exemple les erreurs signées, ne fonctionneront cependant pas avant que toutes les entités d'annuaire mises en jeu dans l'exploitation aient négocié la version 2.

Les réalisateurs voudront bien noter qu'un processus de résolution des erreurs existe et que des corrections pourront être apportées à la présente Recommandation | Norme internationale sous la forme de corrigenda techniques. Les mêmes corrections seront apportées à la présente Recommandation | Norme internationale sous la forme d'un *Guide du réalisateur*. Le Secrétariat du sous-comité peut fournir une liste des corrigenda techniques approuvés pour cette Recommandation | Norme internationale. Les corrigenda techniques publiés peuvent être obtenus auprès de votre organisation nationale de normalisation. Le Guide du réalisateur peut être obtenu par consultation du site Internet de l'UIT.

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

L'Annexe B, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, décrit les prescriptions de sécurité.

L'Annexe C, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, est une introduction à la cryptographie à clé publique.

Remplacée par une version plus récente

L'Annexe D, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, décrit le système cryptographique RSA à clé publique.

L'Annexe E, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, décrit les fonctions de hachage.

L'Annexe F, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, décrit les dangers contre lesquels la protection est assurée par les méthodes d'authentification renforcée.

L'Annexe G, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, décrit la confidentialité des données.

L'Annexe H, qui fait partie intégrante de la présente Recommandation | Norme internationale, définit des identificateurs d'objet assignés aux algorithmes d'authentification et de chiffrement, en l'absence d'un registre formel.

L'Annexe I, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, contient une bibliographie.

L'Annexe J, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, contient des exemples d'emploi de contraintes sur le chemin de certification.

L'Annexe K, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, énumère les amendements et les relevés d'erreurs qui ont été intégrés dans la présente édition de la présente Recommandation | Norme internationale.

**NORME INTERNATIONALE**

**RECOMMANDATION UIT-T**

# TECHNOLOGIES DE L'INFORMATION – INTERCONNEXION DES SYSTÈMES OUVERTS – L'ANNUAIRE: CADRE D'AUTHENTIFICATION

## SECTION 1 – GÉNÉRALITÉS

## 1 Domaine d'application

La présente Recommandation | Norme internationale:

- spécifie la forme sous laquelle les informations d'authentification sont conservées par l'Annuaire;
- décrit la façon dont on peut obtenir de l'Annuaire ces informations d'authentification;
- énonce les hypothèses faites au sujet de la manière dont ces informations d'authentification sont constituées et au sujet de leur emplacement dans l'Annuaire;
- définit les trois manières dont les applications peuvent employer ces informations pour effectuer des authentifications et décrit la façon dont d'autres services de sécurité peuvent être assurés par l'authentification.

La présente Recommandation | Norme internationale décrit deux niveaux d'authentification: l'authentification simple, qui utilise un mot de passe pour la vérification de l'identité et l'authentification renforcée, qui fait intervenir des habilitations établies au moyen de techniques cryptographiques. Tandis que l'authentification simple n'offre qu'une protection limitée contre l'accès non autorisé, seule l'authentification renforcée devra servir de base à la prestation de services sûrs. Il ne s'agit pas d'en faire un cadre général d'authentification, mais ce niveau peut se prêter à une utilisation générale pour des applications qui considèrent que ces techniques sont appropriées.

On ne peut fournir d'authentification (et d'autres services de sécurité) que dans le contexte d'une politique de sécurité définie pour une application. Il appartient aux utilisateurs de cette application de définir leur propre politique de sécurité, qui peut dépendre des services fournis conformément à une norme.

Il appartient aux normes définissant des applications qui utilisent le cadre d'authentification de spécifier les échanges de protocole qu'il convient d'effectuer afin de parvenir à une authentification fondée sur l'information d'authentification obtenue de l'Annuaire. Le protocole utilisé par les applications pour obtenir des justificatifs de l'Annuaire est le protocole d'accès à l'Annuaire (DAP, *directory access protocol*) spécifié dans la Rec. UIT-T X.519 | ISO/CEI 9594-5.

La méthode d'authentification renforcée qui est spécifiée dans la présente Recommandation | Norme internationale repose sur des systèmes cryptographiques à clé (de codage) publique. C'est le principal avantage de ces systèmes que les certificats d'utilisateur puissent être conservés dans l'Annuaire et être obtenus par les utilisateurs de l'Annuaire de la même manière que d'autres informations de l'Annuaire. On admet que les certificats d'utilisateur sont constitués par des moyens indépendants et qu'ils peuvent ensuite être placés dans l'Annuaire. La production de certificats d'utilisateur est effectuée par une autorité de certification autonome qui est complètement distincte des agents DSA de l'Annuaire. En particulier, aucune exigence spéciale n'est prescrite aux fournisseurs de l'Annuaire pour mémoriser ou communiquer les certificats d'utilisateur de façon sûre.

Une brève introduction à la cryptographie à clé publique est donnée dans l'Annexe C.

En général, le cadre d'authentification ne dépend pas de l'utilisation d'un algorithme particulier pour autant qu'il possède les propriétés décrites à l'article 7. Il est possible dans la pratique d'utiliser un certain nombre d'algorithmes différents. Toutefois, deux utilisateurs qui désirent s'authentifier doivent utiliser le même algorithme cryptographique pour effectuer correctement l'authentification. De cette façon, dans le contexte d'un ensemble d'applications voisines, le choix d'un algorithme unique servira à élargir au maximum la communauté d'utilisateurs capables de s'authentifier et de communiquer en sécurité. Un exemple d'algorithme cryptographique de clé publique est spécifié dans l'Annexe D.

De même, deux utilisateurs qui désirent s'authentifier doivent utiliser la même fonction de hachage (voir 3.3.14) (utilisée pour former des justificatifs et des jetons d'authentification). De nouveau, en principe, on peut utiliser un certain nombre de variantes de fonction de hachage au prix d'un rétrécissement des communautés d'utilisateurs capables de s'authentifier. Une brève introduction aux fonctions de hachage et un exemple de fonction de hachage sont donnés dans l'Annexe E.

## 2 Références normatives

Les Recommandations et Normes internationales suivantes contiennent des dispositions qui, par suite de la référence qui y 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 | Norme internationale 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 tient à jour une liste des Recommandations de l'UIT-T en vigueur.

### 2.1 Recommandations | Normes internationales identiques

- Recommandation UIT-T X.411 (1995) | ISO/CEI 10021-4:1997, *Technologies de l'information – Systèmes de messagerie: système de transfert de messages: définition et procédures du service abstrait.*
- Recommandation UIT-T X.500 (1997) | ISO/CEI 9594-1:1999, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: vue d'ensemble des concepts, modèles et services*.
- Recommandation UIT-T X.501 (1997) | ISO/CEI 9594-2:1997, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: les modèles*.
- Recommandation UIT-T X.511 (1997) | ISO/CEI 9594-3:1997, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: définition du service abstrait*.
- Recommandation UIT-T X.518 (1997) | ISO/CEI 9594-4:1997, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: procédures pour le fonctionnement réparti*.
- Recommandation UIT-T X.519 (1997) | ISO/CEI 9594-5:1997, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: spécification du protocole*.
- Recommandation UIT-T X.520 (1997) | ISO/CEI 9594-6:1997, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: types d'attributs sélectionnés*.
- Recommandation UIT-T X.521 (1997) | ISO/CEI 9594-7:1997, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: classes d'objets sélectionnées*.
- Recommandation UIT-T X.525 (1997) | ISO/CEI 9594-9:1999, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: duplication*.
- Recommandation UIT-T X.530 (1997) | ISO/CEI 9594-10:1999, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: utilisation de la gestion de systèmes pour l'administration de l'Annuaire.*
- Recommandation X.660 du CCITT (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.*
- Recommandation UIT-T X.680 (1994) | ISO/CEI 8824-1:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base.*
- Recommandation UIT-T X.681 (1994) | ISO/CEI 8824-2:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels.*
- Recommandation UIT-T X.682 (1994) | ISO/CEI 8824-3:1995, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes.*
- Recommandation UIT-T X.683 (1994) | ISO/CEI 8824-4:1995, *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 (1994) | ISO/CEI 8825-1:1995, *Technologies de l'information – Règles de codage de la notation de syntaxe abstraite numéro un: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*
- Recommandation UIT-T X.690[1)] Cor.2 | ISO/CEI 8825-1[1)] Cor.2.
- 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 X.800 du CCITT (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 des systèmes ouverts – Modèle de référence de base – Partie 2: Architecture de sécurité.*

### 2.3 Autres références

- ISO/CEI 11770-1:1996, *Technologies de l'information – Techniques de sécurité – Partie 1: Cadre général*.
- POSTEL (J.B.), Internet Protocol (Protocole Internet), RFC 791, *Internet Activities Board*, 1981.
- CROCKER (D.H.), Standard for the Format of ARPA Internet Text Messages (Norme pour le format des messages de l'agence ARPA en mode texte sur le réseau Internet), RFC 822, *Internet Activities Board*, 1982.
- MOCKAPETRIS (P.), Domain Names – Implementation and Specification (Noms de domaines – Mise en œuvre et spécification), RFC 1035, *Internet Activities Board*, 1987.
- BERNERS-LEE (T.), *Universal Resource Identifiers in WWW* (Identificateurs universels de ressource dans le réseau WWW), RFC 1630, *Internet Activities Board*, 1994.

## 3 Définitions

Pour les besoins de la présente Recommandation | 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. X.800 du CCITT et l'ISO 7498-2:

a) asymétrique (chiffrement);
b) échange d'authentifications;
c) information d'authentification;
d) confidentialité;
e) 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).

---

1) Actuellement à l'état de projet.

## 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 relatives au cadre d'authentification

Les termes suivants sont définis dans la présente Recommandation | Norme internationale.

**3.3.1 certificat d'attributs**: ensemble d'attributs d'un utilisateur associé à quelques autres informations, qui est rendu infalsifiable par la signature numérique créée au moyen de la clé privée de l'autorité de certification qui a émis ce certificat.

**3.3.2 jeton d'authentification; jeton**: information acheminée au cours d'un échange d'authentification renforcée, qui peut être utilisée pour authentifier son expéditeur.

**3.3.3 certificat d'utilisateur; certificat de clé publique; certificat**: clé publique d'un utilisateur, ainsi que certaines autres informations, rendue infalsifiable par chiffrement avec la clé secrète de l'autorité de certification qui l'a délivrée.

**3.3.4 certificat d'autorité de certification**: certificat délivré par une autorité de certification pour une autre autorité de certification.

**3.3.5 politique de certification**: ensemble nommé de règles qui indique l'applicabilité d'un certificat à une communauté particulière et/ou à une classe d'application particulière ayant des exigences de sécurité communes. Par exemple, une politique de certification donnée peut indiquer l'applicabilité d'un type de certificat à l'authentification de transactions électroniques par transfert de données pour le courtage de marchandises dans une fourchette de prix déterminée.

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

**3.3.7 système utilisateur de certificat**: mise en œuvre des fonctions définies dans la présente spécification d'Annuaire et employées par un utilisateur de certificat.

**3.3.8 autorité de certification**: autorité chargée par un ou par plusieurs utilisateurs de créer et d'attribuer les certificats. Cette autorité peut, facultativement, créer les clés d'utilisateur.

**3.3.9 chemin de certification**: séquence ordonnée de certificats d'objets contenus dans l'arbre DIT, qui peut être traitée pour obtenir la clé publique de l'objet final contenu dans le chemin, en même temps que celle de l'objet initial contenu dans le chemin.

**3.3.10 point de répartition de liste CRL**: entrée de répertoire ou autre source de répartition pour des listes d'annulation de certificat (CRL). Une liste CRL répartie à partir d'un point de répartition CRL peut contenir des entrées d'annulation pour un sous-ensemble seulement de l'ensemble complet des certificats émis par une même autorité de certification ou des entrées d'annulation pour plusieurs autorités de certification.

**3.3.11 système cryptographique; cryptosystème**: recueil de transformations du texte en clair au texte chiffré et réciproquement, les transformations particulières à utiliser étant sélectionnées par des clés. Les transformations sont normalement définies par un algorithme mathématique.

**3.3.12 liste CRL delta**: liste CRL partielle, n'indiquant que les modifications intervenues depuis une création de liste CRL antérieure.

**3.3.13 entité finale**: entité titulaire de certificat qui se sert de sa clé publique pour un autre usage que la signature de certificats.

**3.3.14 fonction (de) hachage**: fonction (mathématique) qui met en correspondance les valeurs d'un grand (et éventuellement très grand) domaine avec une gamme de valeurs plus petite. Une "bonne" fonction de hachage est telle que les résultats de l'application de la fonction à un (grand) ensemble de valeurs du domaine seront régulièrement (et apparemment aléatoirement) répartis sur toute l'étendue de ce domaine.

**3.3.15 agrément de clé**: méthode de négociation directe d'une valeur de clé sans transfert de la clé, même sous forme chiffrée, par exemple par la méthode de Diffie-Hellman (on trouvera de plus amples renseignements sur les mécanismes d'agrément de clé dans l'ISO/CEI 11770-1).

**3.3.16 fonction à une voie**: fonction (mathématique) f qui est facile à calculer mais pour laquelle, à une valeur générale y de l'étendue, il est difficile de faire correspondre par le calcul une valeur x du domaine telle que f(x) = y. Il peut exister un petit nombre de valeurs de y pour lesquelles la détermination de x n'est pas difficile à obtenir par le calcul.

**3.3.17 équivalence de politiques**: reconnaissance du fait que, lorsqu'une autorité de certification d'un domaine donné certifie une autorité de certification dans un autre domaine, une certaine politique de certification dans ce second domaine peut être considérée, par l'autorité de certification du premier domaine, comme équivalente (mais non nécessairement identique à tous points de vue) à une politique de certification particulière dans le premier domaine.

**3.3.18 clé publique**: clé d'une paire de clés d'utilisateur qui est publiquement connue (dans un système cryptographique à clé publique).

**3.3.19 clé privée; clé secrète** (déconseillé): clé d'une paire de clés d'utilisateur qui n'est connue que de cet utilisateur (dans un système cryptographique à clé publique).

**3.3.20 authentification simple**: authentification obtenue au moyen de simples arrangements de mot de passe.

**3.3.21 politique de sécurité**: ensemble de règles établies par l'organisme de sécurité qui régit l'utilisation et la fourniture de services et de facilités de sécurité.

**3.3.22 authentification renforcée**: authentification obtenue au moyen de justificatifs déterminés par cryptographie.

**3.3.23 confiance**: généralement, on peut dire qu'une entité se fie à une deuxième entité lorsqu'elle (la première entité) formule l'hypothèse que la deuxième entité se comportera exactement comme le prévoit la première entité. Cette confiance ne peut s'appliquer qu'à une fonction particulière. Le rôle principal de la confiance dans le cadre d'authentification consiste à décrire la relation entre une entité d'authentification et une autorité de certification; une entité d'authentification doit être certaine de pouvoir se fier à l'autorité de certification pour qu'elle crée uniquement des certificats valides et fiables.

**3.3.24 numéro de série de certificat**: valeur entière, unique pour l'autorité de certification qui l'émet et associée sans ambiguïté au certificat émis par cette autorité.

## 4 Abréviations

Pour les besoins de la présente Recommandation | Norme internationale, les abréviations suivantes s'appliquent.

CA Autorité de certification (*certification authority*)

CRL Liste d'annulation de certificats (*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 d'utilisateur d'annuaire (*directory user agent*)

PKCS Système cryptographique à clé publique (*public key cryptosystem*)

## 5 Conventions

A quelques exceptions mineures près, la présente Spécification d'Annuaire a été élaborée conformément aux directives concernant la "présentation des textes communs UIT-T | ISO/CEI", qui figurent dans le Guide relatif à la coopération entre l'UIT-T et l'ISO/CEI JTC 1, mars 1993.

Le terme "Spécification d'Annuaire" (comme dans "la présente Spécification d'Annuaire") s'entend selon l'acception de la présente Recommandation | Norme internationale. Le terme "Spécifications d'Annuaire" s'entend selon l'acception de toutes les Recommandations de la série X.500 | ISO/CEI 9594.

La présente Spécification d'Annuaire utilise le terme "systèmes de l'édition 1988" pour désigner les systèmes conformes à la première édition (1988) des Spécifications d'Annuaire, c'est-à-dire à l'édition 1988 des Recommandations de la série 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 désigner les systèmes conformes à la deuxième édition (1993) des Spécifications d'Annuaire, c'est-à-dire l'édition 1993 des Recommandations UIT-T de la série X.500 et de l'ISO/CEI 9594:1995. Les systèmes conformes à la présente troisième édition des Spécifications d'Annuaire sont désignés par le terme "systèmes de l'édition 1997".

Cette Spécification d'Annuaire présente la notation ASN.1 en caractères gras de la police Times Roman, 9 points. Lorsque des types et des valeurs ASN.1 sont cités dans le texte normal, ils en sont différenciés par leur présentation en caractères gras Times Roman, 9 points. Les noms des procédures, normalement cités lors de la spécification des sémantèmes de traitement, sont différenciés du texte normal par une présentation en caractères gras de la police Helvetica. Les autorisations de contrôle d'accès sont présentées en caractères italiques de la police Helvetica.

Si, dans une liste, les points sont numérotés (au lieu d'utiliser des tirets ou des lettres), ils sont considérés comme des étapes d'une procédure.

La notation utilisée dans la présente Spécification d'Annuaire est définie dans le Tableau 1 ci-après.

**Tableau 1 – Notation**

| Notation | Signification |
|---|---|
| Xp | Clé publique d'un utilisateur X. |
| Xs | Clé privée de X. |
| Xp[I] | Chiffrement d'une certaine information, I, au moyen de la clé publique de X. |
| Xs[I] | Chiffrement de I au moyen de la clé privée de X. |
| X{I} | Signature de I par l'utilisateur X. Elle se compose de l'information I, assortie d'un sommaire chiffré. |
| CA(X) | Autorité de certification de l'utilisateur X. |
| $CA^n$ (X) | (où 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 (pouvant être de longueur arbitraire) où chaque élément est le certificat pour l'autorité de certification qui a produit le certificat suivant. Il est fonctionnellement équivalent au certificat ci-après: $X_1$«$X_{n+1}$». Par exemple, la possession de A«B»B«C» fournit la même capacité que A«C», à savoir la possibilité de découvrir Cp étant donné Ap. |
| $X_1$p • $X_1$«$X_2$» | Opération de dévoilement d'un certificat (ou d'une chaîne de certificats) pour en extraire une clé publique. C'est un opérateur infixe dont l'opérande gauche est la clé publique d'une autorité de certification, et dont l'opérande droit est un certificat délivré par cette autorité de certification. Le résultat est la clé publique de l'utilisateur dont le certificat est l'opérande droit. Par exemple:<br><br>Ap • A«B» B«C»<br><br>indique l'opération de l'utilisation de la clé publique de A pour obtenir la clé publique Bp de B, à partir de son certificat, suivie de l'utilisation de Bp pour dévoiler le certificat de C. Le résultat de l'opération est la clé publique Cp de C. |
| A→B | Chemin de certification de A vers B, composé d'une chaîne de certificats débutant par: CA(A)«$CA^2$(A)» et finissant par CA(B)«B». |
| NOTE – Dans ce tableau, les symboles X, $X_1$, $X_2$, etc., remplacent les noms des utilisateurs; le symbole I remplace une information quelconque. | |

SECTION 2 – AUTHENTIFICATION SIMPLE

## 6 Procédure d'authentification simple

L'authentification simple vise à fournir une autorisation locale reposant sur un nom distinctif d'utilisateur, sur un mot de passe faisant l'objet (facultativement) d'un accord bilatéral et sur un accord bilatéral quant aux modalités d'emploi et de traitement de ce mot de passe dans un domaine donné. L'utilisation de l'authentification simple est essentiellement destinée à l'emploi local, c'est-à-dire pour authentification d'entités homologues entre un agent DUA et un agent DSA. L'authentification simple peut être réalisée de plusieurs manières:

a) transfert du nom distinctif et du mot de passe (facultatif) de l'utilisateur en clair (sans protection) au destinataire pour évaluation;

b) transfert du nom distinctif de l'utilisateur, du mot de passe de l'utilisateur et d'un numéro aléatoire et/ou d'une indication horaire, qui sont tous protégés par application d'une fonction à une voie;

c) transfert de l'information protégée décrite en b) ainsi que d'un numéro aléatoire et (ou) d'une indication horaire, qui sont tous protégés par application d'une fonction à une voie.

NOTE 1 – Il n'est pas exigé que les fonctions à une voie appliquées soient différentes.

NOTE 2 – La signalisation des procédures de protection des mots de passe pourra faire l'objet d'une extension de la présente Recommandation | Norme internationale.

Quand les mots de passe ne sont pas protégés, un niveau de sécurité minimal est assuré pour empêcher un accès non autorisé. Ce niveau ne doit pas être considéré comme la base de services sûrs. La protection du nom distinctif et du mot de passe de l'utilisateur assure une plus grande sécurité. Les algorithmes à utiliser pour le mécanisme de protection sont en général des fonctions à une voie sans chiffrement, qui sont très simples à mettre en œuvre.

La Figure 1 montre la procédure générale à appliquer pour obtenir une authentification simple.

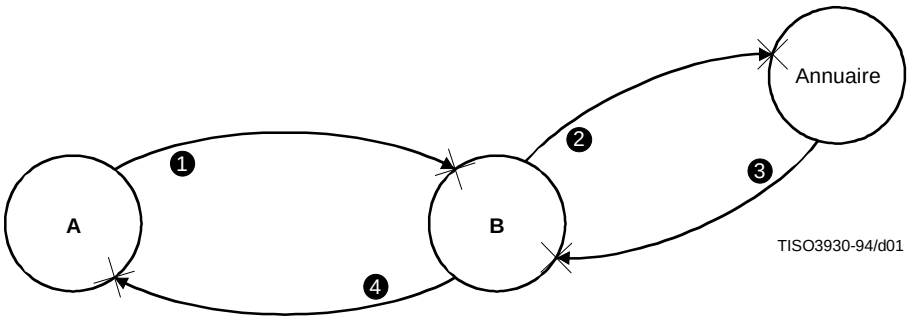



**Figure 1 – Procédure d'authentification simple sans protection**

Les étapes de cette procédure sont les suivantes:

1) un expéditeur A envoie son nom distinctif et son mot de passe à un destinataire B;

2) B envoie le nom distinctif visé et le mot de passe de A à l'Annuaire, où le mot de passe est comparé avec celui qui est contenu en tant qu'attribut **UserPassword** dans l'entrée d'annuaire concernant A (au moyen de l'opération Compare de l'Annuaire);

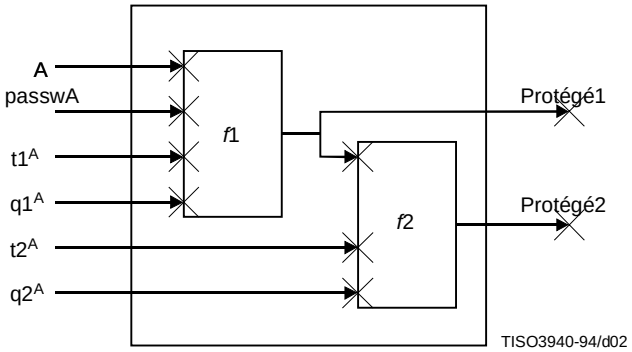
3) l'Annuaire confirme (ou infirme) à B que les habilitations sont valides;

4) le succès (ou l'échec) de l'authentification est communiqué à A.

La forme fondamentale d'authentification simple ne comporte que l'étape 1); elle peut aussi comporter l'étape 4) après que B a vérifié le nom distinctif et le mot de passe.

## 6.1 Production de l'information d'identification protégée

La Figure 2 montre deux méthodes de production d'une information d'identification protégée. $f1$ et $f2$ sont des fonctions à une voie (identiques ou différentes); les indications horaires et les nombres aléatoires sont facultatifs et dépendent d'accords bilatéraux.




| | |
|---|---|
| A | Nom distinctif de l'utilisateur |
| $t^A$ | Indications horaires |
| passwA | Mot de passe de A |
| $q^A$ | Numéros aléatoires avec inclusion d'un compteur (facultatif) |

**Figure 2 – Authentification simple protégée**

## 6.2 Procédure d'authentification simple protégée

La Figure 3 montre la procédure d'authentification simple protégée.

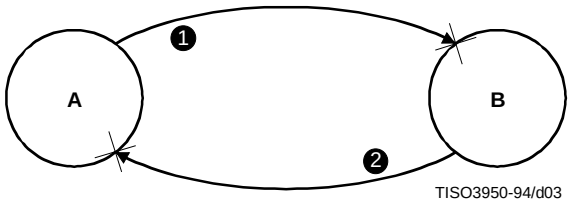



**Figure 3 – Procédure d'authentification simple protégée**

Cette procédure comporte les étapes suivantes (avec, initialement, utilisation de $f1$ seulement):

1) l'expéditeur (A) envoie à l'utilisateur B son information d'identification protégée (Authenticator1). La protection est assurée par application de la fonction à une voie ($f1$) de la Figure 2, où l'indication horaire et (ou) le nombre aléatoire (s'il est utilisé) ont pour objet de réduire au minimum les répétitions et de cacher le mot de passe.

   La protection du mot de passe de A a la forme:

   $$\text{Protected1} = f1\ (t1^A, q1^A, \text{A}, \text{passwA})$$

   L'information transmise à B prend la forme:

   $$\text{Authenticator1} = t1^A, q1^A, \text{A}, \text{Protected1}$$

2) B vérifie l'information d'identification protégée offerte par A en produisant une copie locale protégée du mot de passe de A (de la forme Protected1) (au moyen du nom distinctif et, facultativement, d'une

indication horaire et/ou d'un nombre aléatoire fourni par A, ainsi qu'une copie locale du mot de passe de A). B compare l'information d'identification visée (Protected1) avec la valeur produite localement, afin de s'assurer de leur égalité.

3) B confirme (ou infirme) à A la vérification de l'information d'identification protégée.

La procédure peut être modifiée de manière à assurer une plus grande protection (au moyen de $f1$ et $f2$). Les principales différences sont les suivantes:

1) A envoie son information d'identification protégée supplémentaire (Authenticator2) à B. Une protection supplémentaire est obtenue en appliquant une autre fonction $f2$ comme le montre la Figure 2. La protection supplémentaire prend la forme:

$$\text{Protected2} = f2\ (t2^A, q2^A, \text{Protected1})$$

L'information transmise à B prend la forme:

$$\text{Authenticator2} = t1^A, t2^A, q1^A, q2^A, A, \text{Protected2}$$

Pour comparaison, B émet la valeur locale du mot de passe protégé supplémentaire de A et la compare (pour en contrôler l'égalité) avec celle de Protected2 [le principe étant le même que pour l'étape 1) du 6.2].

2) B confirme (ou infirme) à A la vérification de l'information d'identification protégée.

NOTE – Les procédures définies dans les présents articles sont spécifiées en fonction de A et de B. Appliqué à l'Annuaire (spécifié dans les Rec. UIT-T X.511 | ISO/CEI 9594-3 et UIT-T X.518 | ISO/CEI 9594-4), A pourrait être un agent DUA lié à un agent DSA, B; en variante, A pourrait être un agent DSA lié à un autre agent DSA, B.

### 6.3 Type d'attribut de mot de passe d'utilisateur

Un type d'attribut de mot de passe d'utilisateur contient le mot de passe d'un objet. Une valeur d'attribut pour le mot de passe de l'utilisateur est une chaîne spécifiée par l'objet.

```
userPassword ATTRIBUTE ::= {
    WITH SYNTAX                 OCTET STRING (SIZE (0..ub-user-password))
    EQUALITY MATCHING RULE      octetStringMatch
    ID                          id-at-userPassword }
```

## SECTION 3 – AUTHENTIFICATION RENFORCÉE

## 7 Base de l'authentification renforcée

La façon d'aborder une authentification renforcée au sens de la présente Spécification d'Annuaire fait usage des propriétés de la famille des systèmes cryptographiques connus sous le nom de systèmes cryptographiques à clé publique (PKCS, *public key cryptosystems*). Ces systèmes cryptographiques, également décrits comme asymétriques, font intervenir une paire de clés, l'une privée et l'autre publique, plutôt qu'une seule clé comme dans les systèmes cryptographiques classiques. L'Annexe C donne une brève introduction à ces systèmes cryptographiques et aux propriétés qui les rendent utiles pour l'authentification. Pour qu'un système PKCS soit actuellement utilisable dans ce cadre d'authentification, il doit avoir la propriété de permettre l'usage des deux clés de la paire pour le chiffrement, la clé privée étant utilisée pour le déchiffrement si c'est la clé publique qui a été utilisée, et la clé publique étant utilisée pour le déchiffrement si c'est la clé privée qui a été utilisée. Autrement dit, $X_p \bullet X_s = X_s \bullet X_p$ si $X_p/X_s$ sont des fonctions de chiffrement/déchiffrement utilisant les clés publique/privée de l'utilisateur X.

NOTE – On pourra ultérieurement envisager d'autres types de système PKCS, c'est-à-dire qui n'exigent pas la propriété de permutabilité et qui peuvent être mis en œuvre sans grande modification de la présente Spécification d'Annuaire.

Le cadre d'authentification ne prescrit pas l'emploi d'un système cryptographique particulier. Il est prévu que ce cadre s'appliquera à tout système cryptographique à clé publique et qu'il suivra ainsi les modifications des méthodes utilisées à la suite des progrès à venir en cryptographie, en techniques mathématiques ou en capacités de calcul. Toutefois deux utilisateurs qui désirent s'authentifier doivent utiliser le même algorithme cryptographique pour que les authentifications soient effectuées correctement. De cette façon, dans le contexte d'un ensemble d'applications voisines, le choix d'un algorithme unique servira à élargir au maximum la communauté des utilisateurs capables de s'authentifier et de communiquer en sécurité. Un exemple d'algorithme cryptographique est donné dans l'Annexe D.

L'authentification repose sur la possession d'un nom distinctif unique pour chaque utilisateur. La responsabilité de l'attribution de noms distinctifs revient aux autorités de désignation. Chaque utilisateur doit donc se fier aux autorités de dénomination pour que les noms distinctifs ne soient pas émis en double.

Chaque utilisateur est identifié par la possession de sa clé privée. Un deuxième utilisateur est capable de déterminer si un partenaire d'une communication est en possession de la clé privée et peut utiliser ce renseignement pour confirmer que le partenaire de la communication est en fait l'utilisateur. La validité de cette confirmation dépend de la mesure dans laquelle la clé privée demeure confidentielle au niveau de l'utilisateur.

Pour qu'un utilisateur puisse déterminer si un partenaire de communication est en possession de la clé privée d'un autre utilisateur, il doit être lui-même en possession de la clé publique de cet utilisateur. S'il est simple d'obtenir la valeur de cette clé publique d'après l'inscription de l'utilisateur dans l'Annuaire, il est plus problématique d'en vérifier l'exactitude. Il existe pour cela de nombreuses possibilités; l'article 8 décrit un traitement au moyen duquel une clé publique d'utilisateur peut être contrôlée par référence à l'Annuaire. Ce traitement ne peut remplir son rôle que s'il existe une chaîne continue de points de confiance dans l'Annuaire entre les utilisateurs demandant à s'authentifier. Pour constituer cette chaîne, on peut identifier un point de confiance commun. Ce point doit être relié à chaque utilisateur par une chaîne continue de points de confiance.

## 8 Obtention d'une clé publique d'utilisateur

Pour qu'un utilisateur puisse avoir confiance en la procédure d'authentification, il doit obtenir la clé publique de l'autre utilisateur à partir d'un point de confiance. Une telle origine, appelée autorité de certification (CA, *certification authority*), utilise l'algorithme de clé publique pour certifier la clé publique en produisant un *certificat*. Ce certificat, dont la forme est spécifiée plus loin, possède les propriétés suivantes:

- tout utilisateur ayant accès à la clé publique de l'autorité de certification peut recouvrer la clé publique qui a été certifiée;
- aucune partie autre que l'autorité de certification ne peut modifier le certificat sans que cela soit détecté (les certificats sont infalsifiables).

Du fait que les certificats sont infalsifiables, on peut les publier en les introduisant dans l'Annuaire sans que ce dernier ait à prendre de dispositions particulières pour assurer leur protection.

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

Une autorité de certification produit les certificats de l'utilisateur en signant (voir l'article 9) un recueil d'informations comprenant le nom distinctif d'utilisateur et la clé publique, ainsi qu'un *identificateur unique* facultatif comprenant des informations supplémentaires concernant l'utilisateur. La forme exacte du contenu de l'identificateur unique n'est pas spécifiée ici; elle est laissée aux soins de l'autorité de certification et peut être par exemple un identificateur d'objet, un certificat, une date ou une autre forme de certification de la validité du nom distinctif. Plus précisément, le certificat d'utilisateur ayant A comme nom distinctif et UA comme identificateur unique, produit par l'autorité de certification ayant CA comme nom et UCA comme identificateur unique, prend la forme suivante:

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

où V est la version du certificat, SN le numéro de série du certificat, AI l'identificateur de l'algorithme servant à signer le certificat, UCA l'identificateur unique facultatif de l'autorité CA, UA l'identificateur unique facultatif de l'utilisateur A; $T^A$ indique la période de validité du certificat et se compose de deux dates, correspondant au début et à la fin de cette validité. Comme on suppose que $T^A$ est modifié par période d'au moins 24 heures, on prévoit que les systèmes utiliseront le temps universel coordonné comme base de temps de référence. Tout utilisateur ayant connaissance de CAp peut vérifier la validité de la signature du certificat. Le type de données ASN.1 suivant peut être utilisé pour représenter les 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 présent, la version doit être v2 ou v3
        subjectUniqueIdentifier [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                                        -- si présent, la version doit être v2 ou v3
        extensions              [3]  Extensions OPTIONAL
                                        -- si 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 }
-- L'ensemble d'objets d'information ci-après sera défini ultérieurement, peut-être selon des protocoles
-- normalisés ou selon des déclarations de conformité d'instance de protocole. Cet ensemble est nécessaire
-- pour spécifier une contrainte du tableau applicable à l'élément parameters d'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

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

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

Avant qu'une valeur de temps soit utilisée dans une opération de comparaison, par exemple dans le cadre d'une règle de correspondance pour une recherche, et si la syntaxe du type **Time** a été choisie comme étant du type **UTCTime**, la valeur du champ d'année à deux chiffres doit être converti en champ d'année à quatre chiffres comme suit:

- si la valeur à 2 chiffres va de 00 à 49 inclus, on ajoute le nombre 2000 à cette valeur;
- si la valeur à 2 chiffres va de 50 à 99 inclus, on ajoute le nombre 1900 à cette valeur.

NOTE 2 – L'utilisation de l'élément **GeneralizedTime** peut empêcher l'interfonctionnement avec des réalisations ne possédant pas la possibilité de choisir entre **UTCTime** et **GeneralizedTime**. Il appartient aux spécificateurs des domaines d'utilisation des certificats définis dans la présente Spécification d'Annuaire de déterminer à quel moment le temps généralisé pourra être utilisé, par exemple dans des groupes de profilage. En aucun cas le temps UTC ne doit être utilisé pour représenter des dates ultérieures à 2049.

Le champ d'**extension** permet d'ajouter de nouveaux champs à la structure sans modifier la définition ASN.1. Un champ d'extension se compose d'un identificateur d'extension, d'un fanion de criticité et d'un codage canonique d'une valeur de donnée de type ASN.1 associée à l'extension désignée. Lorsqu'une implémentation effectuant le traitement d'un certificat ne reconnaît pas une extension et que le fanion de criticité est à la valeur Faux, cette implémentation peut ignorer cette extension. Si le fanion de criticité est à la valeur Vrai, des extensions non reconnues peuvent faire que la structure soit considérée comme invalide; en d'autres termes, une extension critique non reconnue dans un certificat entraînera l'échec de la validation d'une signature utilisant ce certificat.

Si des éléments inconnus apparaissent dans l'extension et que celle-ci ne soit pas marquée comme étant critique, ces éléments inconnus doivent être ignorés conformément aux règles d'extensibilité indiquées au 7.5.2.2 de la Rec. UIT-T X.519 | ISO/CEI 9594-5.

Des extensions spécifiques peuvent être définies dans les Recommandations | Normes internationales, ou par toute organisation qui en a besoin. L'identificateur d'objet qui désigne une extension doit être défini conformément à la Rec. X.660 du CCITT| ISO/CEI 9834-1. L'article 12 définit des extensions normalisées pour certificats.

La classe d'objets suivante est utilisée pour définir des extensions spécifiques:

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

NOTE 3 – Dans les cas où l'autorité de dénomination peut réattribuer le nom distinctif à un autre utilisateur, les autorités d'authentification peuvent utiliser l'identificateur unique pour distinguer les occurrences réutilisées. Toutefois, si plusieurs autorités d'authentification fournissent des certificats au même utilisateur, il est recommandé que ces autorités d'authentification effectuent une coordination de l'attribution des identificateurs uniques dans le cadre de leurs procédures d'enregistrement d'utilisateur.

L'entrée d'annuaire de chaque utilisateur, A, qui participe à une authentification renforcée, contient le(s) certificat(s) de A. Ce certificat est produit par une autorité de certification de A, qui est une entité de l'arbre DIT. L'autorité de certification de A, qui n'est pas forcément unique, est indiquée par CA(A) ou simplement par CA si A est sous-entendu. La clé publique de A peut ainsi être découverte par tout utilisateur qui connaît la clé publique de CA. La découverte des clés publiques est donc récursive.

Si l'utilisateur A, qui essaie d'obtenir la clé publique de l'utilisateur B, a déjà obtenu la clé publique de CA(B), le processus est achevé. Pour permettre à A d'obtenir la clé publique de CA(B), l'entrée d'annuaire de chaque autorité de certification X contient plusieurs certificats, qui sont de deux types. D'abord les certificats progressifs de X produits par d'autres autorités de certification, ensuite les certificats régressifs produits par X, qui sont les clés publiques certifiées d'autres autorités de certification. L'existence de ces certificats permet aux utilisateurs d'élaborer des chemins de certification entre deux points.

La liste des certificats nécessaires pour permettre à un utilisateur d'obtenir la clé publique d'un autre utilisateur est appelée *chemin de certification*, chaque élément de la liste étant un certificat de l'autorité de certification pour le suivant. Un chemin de certification de A à B (désigné par $A \rightarrow B$):

- débute avec le certificat produit par CA(A), à savoir: CA(A)«$X^1$» pour une entité $X^1$;
- continue avec d'autres certificats $X^i$«$X^{i+1}$»;
- finit avec le certificat de B.

Un chemin de certification forme logiquement une chaîne continue de points de confiance dans l'arbre d'informations d'Annuaire entre deux utilisateurs qui désirent s'authentifier. La méthode précise utilisée par les utilisateurs A et B pour obtenir les chemins de certification $A \rightarrow B$ et $B \rightarrow A$ peut varier. Une façon d'y parvenir plus facilement consiste à établir une hiérarchie d'autorités de certification pouvant coïncider ou non avec une partie ou avec la totalité de la hiérarchie de l'arbre DIT. L'avantage pour les utilisateurs qui ont des autorités de certification dans la hiérarchie est de pouvoir établir un chemin de certification entre eux au moyen de l'Annuaire sans information préalable. Pour cela, chaque autorité de certification peut stocker un certificat et un certificat inverse désigné comme correspondant à son autorité de certification supérieure.

Les certificats sont conservés dans les entrées d'annuaire en tant qu'attributs des types **UserCertificate**, **CACertificate** et **CrossCertificatePair**. Ces types d'attribut sont connus de l'Annuaire. On peut effectuer des opérations sur ces attributs en utilisant les mêmes opérations de protocole que pour d'autres attributs. La définition de ces types se trouve au 3.3; leur spécification est la suivante:

```
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 }

CertificatePair ::= SEQUENCE {
    forward [0] Certificate OPTIONAL,
    reverse [1] Certificate OPTIONAL
        -- au moins un des certificats de la paire doit être présent -- }
```

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

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

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

De plus, le type de données ASN.1 suivant peut être utilisé pour représenter le chemin de certification progressif. Il s'agit d'une composante contenant le chemin de certification qui peut pointer vers l'arrière jusqu'à l'expéditeur.

```
ForwardCertificationPath ::= SEQUENCE OF CrossCertificates

CrossCertificates ::= SET OF Certificate
```

## 8.1 Optimisation de la quantité d'information obtenue de l'Annuaire

Dans le cas général, avant que les utilisateurs puissent mutuellement s'authentifier, l'Annuaire doit fournir les chemins complets de certification pour l'aller et le retour. Toutefois, dans la pratique, on peut réduire la quantité d'informations que doit fournir l'Annuaire dans un cas donné d'authentification:

a) si les deux utilisateurs qui désirent s'authentifier sont desservis par la même autorité de certification, le chemin de certification devient commun et les utilisateurs se révèlent directement l'un à l'autre leur certificat;

b) si les autorités de certification des utilisateurs sont disposées en ordre hiérarchique, un utilisateur pourra mémoriser les clés publiques, les certificats et les certificats inverses de toutes les autorités de certification comprises entre les utilisateurs et la racine de l'arbre DIT. Cela amène en général l'utilisateur à connaître les clés publiques et les certificats de trois ou quatre autorités de certification seulement. L'utilisateur dans ce cas n'aura besoin d'obtenir les chemins de certification qu'à partir du point commun de confiance;

c) si un utilisateur est fréquemment en communication avec d'autres utilisateurs certifiés par une autre autorité de certification donnée, il pourra connaître le chemin de certification vers cette autorité et le chemin de retour de certification provenant de cette autorité, ne rendant ainsi nécessaire que l'obtention du certificat de l'autre utilisateur lui-même à partir de l'Annuaire;

d) les autorités de certification peuvent se communiquer réciproquement leurs certificats par accord bilatéral, ce qui raccourcit le chemin de certification;

e) si deux utilisateurs ont déjà communiqué et connaissent réciproquement leurs certificats, ils ont la possibilité de s'authentifier sans avoir recours à l'Annuaire.

De toute façon, après avoir pris mutuellement connaissance de leurs certificats d'après le chemin de certification, les utilisateurs doivent vérifier la validité des certificats reçus.

## 8.2 Exemple

La Figure 4 représente un exemple fictif de fragment d'arbre DIT, dans lequel les autorités de certification forment une hiérarchie. Outre l'information indiquée aux autorités de certification, on admet que chaque utilisateur a connaissance de la clé publique de son autorité de certification, ainsi que de ses propres clés, publique et privée.

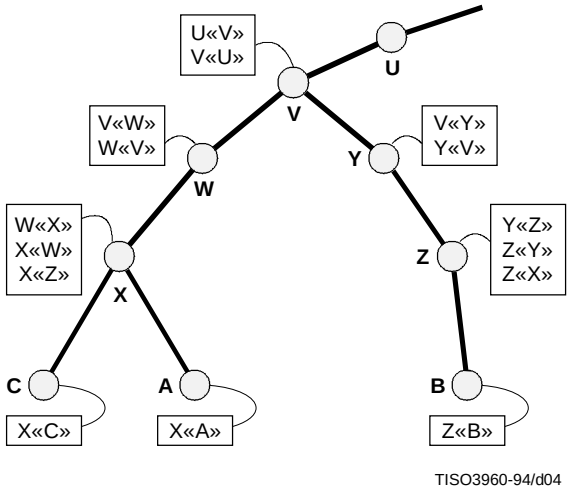



**Figure 4 – Hiérarchie des CA – Exemple théorique**

Si les autorités de certification des utilisateurs sont disposées hiérarchiquement, A peut acquérir les certificats suivants en provenance de l'Annuaire pour établir un chemin de certification de A vers B:

X«W», W«V», V«Y», Y«Z», Z«B»

Lorsque A a obtenu ces certificats, il peut dévoiler le chemin de certification en séquence pour livrer le contenu du certificat de B y compris Bp:

Bp = Xp • X«W» W«V» V«Y» Y«Z» Z«B»

En général, A doit également acquérir les certificats suivants en provenance de l'Annuaire pour établir le chemin de retour de certification de B vers A:

Z«Y», Y«V», V«W», W«X», X«A»

Lorsque B reçoit ces certificats de A, il peut dévoiler le chemin de retour de certification en séquence pour livrer le contenu du certificat de A y compris Ap:

Ap = Zp • Z«Y» Y«V» V«W» W«X» X«A»

En appliquant les optimisations du 8.1:

a) considérons A et C par exemple: tous les deux ont connaissance de Xp, de sorte qu'il ne reste à A qu'à acquérir directement le certificat de C. Le dévoilement de le chemin de certification se réduit à:

   Cp = Xp • X«C»

   et le dévoilement du chemin de retour de certification se réduit à:

   Ap = Xp • X«A»

b) en admettant que A ait connaissance de la même façon de W«X», Wp, V«W», Vp, U«V», Up, etc., l'information que A doit obtenir de l'Annuaire pour former le chemin de certification, se réduit à:

   V«Y», Y«Z», Z«B»

et l'information que A doit obtenir de l'Annuaire pour former le chemin de retour de certification, se réduit à:

Z«Y», Y«V»

c) en admettant que A soit fréquemment en communication avec les utilisateurs certifiés par Z, il peut être au courant [en plus des clés publiques décrites à l'alinéa b) ci-dessus], de V«Y», Y«V», Y«Z», et Z«Y». Pour communiquer avec B, il lui suffit donc d'obtenir Z«B» de l'Annuaire;

d) en admettant que les utilisateurs certifiés par X et Z soient fréquemment en communication, X«Z» sera conservé dans l'entrée d'annuaire pour X et réciproquement (cela est représenté dans la Figure 4). Si A désire procéder aux authentifications avec B, il lui suffit d'obtenir:

X«Z», Z«B»

pour former le chemin de certification, et:

Z«X»

pour former le chemin de retour de certification;

e) en admettant que les utilisateurs A et C aient communiqué auparavant et aient été mis au courant réciproquement de leurs certificats, ils peuvent utiliser directement la clé publique de l'autre, c'est-à-dire:

$Cp = Xp \bullet X«C»$

et

$Ap = Xp \bullet X«A»$

Dans le cas plus général, les autorités de certification n'ont pas entre elles de relations hiérarchiques. Dans l'exemple théorique de la Figure 5, supposons que l'utilisateur D, certifié par U, désire s'authentifier avec l'utilisateur E certifié par W. L'entrée d'Annuaire de l'utilisateur D contiendra le certificat U«D» et celle de l'utilisateur E contiendra le certificat W«E».

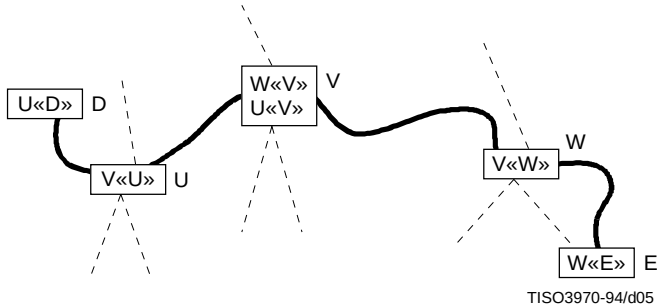



**Figure 5 – Exemple d'itinéraire de certification non hiérarchique**

Soit V une autorité de certification avec laquelle U et W ont auparavant échangé des clés publiques en confiance. Il s'ensuit que des certificats U«V», V«U», W«V» et V«W» ont été produits et mémorisés dans l'Annuaire. En supposant que U«V» et W«V» sont stockés dans l'entrée de V, V«U» est stocké dans l'entrée de U et V«W» est stocké dans l'entrée de W.

L'utilisateur D doit trouver un chemin de certification vers E. Plusieurs méthodes peuvent être utilisées. L'une consiste à considérer les utilisateurs et les autorités de certification comme des nœuds et les certificats comme des arcs dans un graphe orienté. Dans ces conditions, D doit rechercher dans le graphique un chemin de U vers E, qui pourra être U«V», V«W», W«E». Une fois que ce chemin a été découvert, le chemin inverse W«V», V«U», U«D» peut aussi être constitué.

# 9 Signatures numériques

Cet article est destiné à spécifier non une norme traitant des signatures numériques en général, mais les moyens permettant de signer les jetons dans l'Annuaire.

Pour signer une information (info) on lui ajoute un résumé chiffré de cette information. Ce résumé est produit au moyen d'une fonction de hachage à une voie, tandis que le chiffrement est effectué au moyen de la clé privée du signataire (voir la Figure 6) ainsi:

$$X\{Info\} = Info, Xs[h(Info)]$$

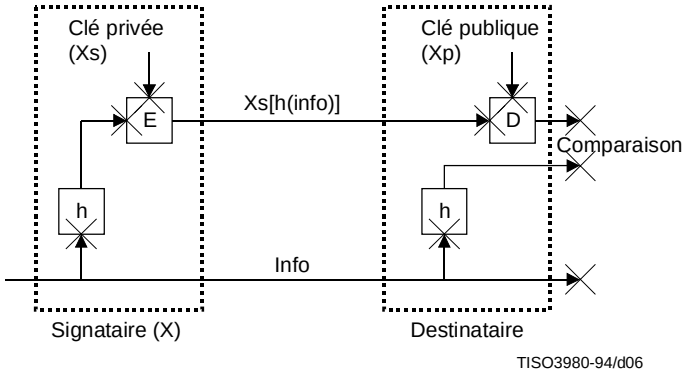



**Figure 6 – Signatures numériques**

NOTE 1 – Le chiffrement utilisant la clé privée garantit que la signature ne peut pas être falsifiée. La nature à une seule voie de la fonction de hachage fait en sorte qu'une information fausse, produite de façon à obtenir le même résultat de hachage (et ainsi la signature), ne puisse être substituée.

Le destinataire de l'information signée vérifie la signature:

- en appliquant la fonction de hachage à une voie à l'information;
- en comparant le résultat avec celui que l'on a obtenu par déchiffrement de la signature au moyen de la clé publique du signataire.

Ce cadre d'authentification ne prescrit pas d'utiliser pour la signature une fonction de hachage unique à une voie. Il est prévu que ce cadre s'appliquera à toute fonction de hachage appropriée et suivra ainsi les modifications des méthodes utilisées par suite des progrès futurs en cryptographie, dans les techniques mathématiques ou dans les capacités de calcul. Toutefois, deux utilisateurs qui désirent s'authentifier doivent utiliser la même fonction de hachage pour que les authentifications soient effectuées correctement. De cette façon, dans le contexte d'un ensemble d'applications voisines, le choix d'une fonction unique servira à élargir au maximum la communauté des utilisateurs capables de s'authentifier et de communiquer en sécurité.

L'information signée comprend des indicateurs qui identifient l'algorithme de hachage et l'algorithme de chiffrement servant à établir la signature numérique.

Le chiffrement de certains éléments de données peut être décrit au moyen de la notation ASN.1 suivante:

```
ENCRYPTED { ToBeEnciphered }        ::=          BIT STRING ( CONSTRAINED BY {
    -- doit résulter de l'application d'une procédure de chiffrement aux octets codés selon
    -- les règles de codage de base d'une valeur de l'élément  --  ToBeEnciphered } )
```

La valeur de la chaîne de bits est produite ainsi: on prend les octets qui forment le codage complet (à l'aide des règles de codage de base ASN.1 – Rec. UIT-T X.690 | ISO/CEI 8825-1) de la valeur du type **ToBeEnciphered** et on applique une procédure de chiffrement à ces octets.

NOTE 2 – La procédure de chiffrement exige un accord sur l'algorithme à employer et, le cas échéant, sur ses paramètres, comme les clés nécessaires, les valeurs d'initialisation et les instructions de remplissage. Il appartient aux procédures de chiffrement de spécifier les moyens qui permettent d'obtenir la synchronisation de l'émetteur et du récepteur des données, ce qui peut inclure une information dans les bits à transmettre.

NOTE 3 – La procédure de chiffrement doit prendre comme entrée une chaîne d'octets et engendrer une seule chaîne de bits en tant que résultat.

NOTE 4 – Les mécanismes d'obtention d'accord sur l'algorithme de chiffrement et ses paramètres par l'expéditeur et par le destinataire des données n'entrent pas dans le cadre de la présente Spécification d'Annuaire.

La signature d'un item de données est formée par chiffrement d'une transformation abrégée (ou "hachée") de cet item. On peut la décrire par la notation ASN.1 suivante:

```
ENCRYPTED-HASH { ToBeSigned }     ::=    BIT STRING ( CONSTRAINED BY {
    -- doit résulter de l'application d'une procédure de hachage aux octets codés selon les règles DER d'une valeur
    -- de l'élément -- ToBeSigned -- puis de l'application d'une procédure de chiffrement à ces octets -- })

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

NOTE 5 – La procédure de chiffrement exige les accords énumérés dans la Note 2 ainsi que l'obtention d'un accord sur la question de savoir si les octets hachés sont chiffrés directement ou seulement après avoir été recodés en chaîne binaire au moyen des règles de codage de base ASN.1.

Au cas où une signature doit être ajoutée à un type de données, la notation ASN.1 suivante peut être utilisée pour définir le type de données qui résulte de l'application d'une signature au type de données indiqué.

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

Pour permettre la validation des types **SIGNED** et **SIGNATURE** dans un contexte réparti, un codage spécifique est nécessaire. Un tel codage d'une valeur de données **SIGNED** ou **SIGNATURE** est obtenu par application des règles de codage de base définies dans l'ISO 8825, moyennant les restrictions suivantes:

a) on utilisera la forme définie de codage de longueur, codée dans un nombre minimal d'octets;

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

c) si la valeur d'un type est sa valeur par défaut, elle sera absente;

d) les éléments d'un type d'ensemble seront codés dans l'ordre ascendant de leurs valeurs d'étiquette;

e) les éléments d'un type d'ensemble seront codés dans l'ordre ascendant de leurs valeurs d'octet;

f) si la valeur d'un type booléen est Vrai, le codage aura son contenu d'octets mis sur "FF"$_{16}$;

g) tous les bits inutilisés du dernier octet lors du codage d'une valeur de chaîne de bits, sont, le cas échéant, mis sur zéro;

h) le codage du type réel sera tel que les bases 8, 10 et 16 ne seront pas utilisées et le facteur de proportionnalité binaire aura la valeur zéro;

i) le codage d'un temps UTC doit être conforme à la Rec. UIT-T X.690/Cor.2 | ISO/CEI 8825-1/Cor.2;

j) le codage d'un temps généralisé doit être tel que spécifié dans la Rec. UIT-T X.690 | ISO/CEI 8825-1.

La production d'un codage distinctif implique que la syntaxe abstraite des données à coder soit entièrement comprise. L'Annuaire peut être appelé à signer des données ou à contrôler la signature de données contenant des extensions de protocole inconnues ou des syntaxes d'attribut inconnues. L'Annuaire doit suivre les règles suivantes:

– préserver le codage des informations reçues, dont il ne connaît pas entièrement la syntaxe abstraite et qu'il prévoit de signer ultérieurement;

– envoyer avec un codage distinctif, au moment de la signature des données à envoyer, les données dont il connaît entièrement la syntaxe et toute autre donnée avec son codage préservé; signer le codage qu'il envoie réellement;

– lors de la vérification des signatures dans les données reçues, vérifier chaque signature en fonction des données réellement reçues, plutôt qu'en fonction de la forme convertie des donnée reçues en codage distinctif.

# 10 Procédures d'authentification renforcée

## 10.1 Aperçu général

La manière d'aborder fondamentalement l'authentification a été exposée ci-dessus, à savoir la confirmation de l'identité en faisant apparaître la possession d'une clé privée. Or, il existe de nombreuses procédures possibles qui emploient cette approche. En général, c'est la tâche d'une application particulière de déterminer les procédures appropriées de façon à répondre à la politique de sécurité pour l'application. Cet article contient la description de trois procédures particulières d'authentification qui peuvent s'avérer utiles au travers d'une gamme d'applications.

NOTE – La présente Spécification d'Annuaire ne spécifie pas les procédures en précisant les détails nécessaires à leur mise en œuvre. Toutefois, des Recommandations | Normes internationales complémentaires pourraient être envisagées pour le faire soit dans le cas d'une application particulière, soit d'une façon qui viserait un objectif général.

Les trois procédures font intervenir différents nombres d'échanges d'information d'authentification et en conséquence offrent différents types d'assurance à leurs participants:

a) une authentification à une voie, décrite au 10.2, fait intervenir un transfert d'information unique d'un utilisateur (A) à un autre utilisateur (B), et établit:
   - l'identité de A, et le fait que le jeton d'authentification a été effectivement produit par A;
   - l'identité de B, et le fait que le jeton d'authentification a été effectivement prévu pour être envoyé à B;
   - l'intégrité et "l'originalité" (la propriété de ne pas avoir été envoyé deux fois ou plus) du jeton d'authentification en cours de transfert.

     Ces dernières propriétés peuvent également être établies pour des données additionnelles arbitraires qui accompagnent le transfert;

b) une authentification à deux voies, décrite au 10.3, fait intervenir en plus une réponse de B en A. Elle établit en plus:
   - que le jeton d'authentification produit dans la réponse l'a effectivement été par B et qu'il est destiné à être envoyé en A;
   - l'intégrité et l'originalité du jeton d'authentification envoyé dans la réponse;
   - à titre d'option, le secret mutuel d'une partie des jetons;

c) une authentification à trois voies, décrite au 10.4, fait intervenir, en plus, un autre transfert de A en B. Elle établit les mêmes propriétés que l'authentification à deux voies mais le fait sans qu'il soit nécessaire d'associer une vérification de la date et de l'heure.

Dans chaque cas d'authentification renforcée, A doit obtenir la clé publique de B et le chemin de retour de certification de B en A avant tout échange d'information. Cela peut faire intervenir l'accès à l'Annuaire comme décrit à l'article 7, mais cet accès, s'il existe, n'est pas mentionné de nouveau dans la description des procédures ci-dessous.

La vérification des dates et heures, citée dans les paragraphes suivants, s'applique uniquement si des horloges synchronisées sont employées dans un environnement local ou si des horloges sont synchronisées logiquement par des accords bilatéraux. Dans tous les cas, il est recommandé d'utiliser le temps universel coordonné.

Pour chacune des trois procédures d'authentification décrites ci-après, on admet que l'utilisateur A a vérifié la validité de tous les certificats sur le trajet de certification.

## 10.2 Authentification à une voie

L'authentification à une voie comprend les étapes suivantes, représentées à la Figure 7:

1) A produit $r^A$, nombre non récurrent qui est utilisé pour détecter les attaques par redéfilement et pour empêcher les falsifications;

2) A envoie le message suivant en B:

$$B \rightarrow A, A\{t^A, r^A, B\}$$

où $t^A$ représente la date et l'heure. $t^A$ comprend une ou deux indications chronologiques, l'heure de production du jeton (facultatif) et la date d'expiration. Ou bien, si l'authentification de l'origine des données de "sgnData" doit être fournie par la signature numérique:

$$B \rightarrow A, A\{t^A, r^A, B, sgnData\}$$

Si l'information à transmettre est utilisée par la suite comme clé privée (cette information est désignée par "encData"):

$$B \rightarrow A,\ A\{t^A,\ r^A,\ B,\ sgnData,\ Bp[encData]\}$$

L'emploi de "encData" comme clé privée exige un soin particulier, par exemple afin qu'il s'agisse d'une clé solide pour tout système cryptographique utilisé comme indiqué dans le champ "sgnData" du jeton;

3) B effectue les opérations suivantes:

   a) obtient Ap à partir de B → A, vérifiant que le certificat de A n'est pas périmé;

   b) vérifie la signature et ainsi l'intégrité de l'information signée;

   c) vérifie que B est bien le destinataire prévu;

   d) vérifie que la date et l'heure sont "à jour";

   e) facultativement, vérifie que $r^A$ n'a pas été soumis à un redéfilement. Ceci peut, par exemple, s'obtenir en ayant introduit dans $r^A$ une partie séquentielle qui est testée par une implémentation locale, uniquement pour vérifier cette unicité de valeur.

      $r^A$ est valide jusqu'à la date d'expiration indiquée par $t^A$. $r^A$ est toujours accompagné d'une partie séquentielle qui indique que A ne répétera pas le jeton pendant la durée $t^A$ et ainsi, que la vérification de la valeur de $r^A$ n'est pas nécessaire.

      De toute façon, B a intérêt à stocker la partie séquentielle avec l'indication horaire $t^A$ en clair et avec la partie hachée du jeton pendant la durée $t^A$.

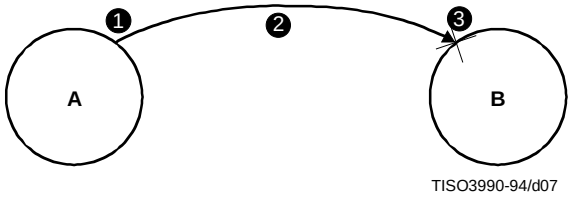



**Figure 7 – Authentification à une voie**

## 10.3 Authentification à deux voies

L'authentification à deux voies comprend les étapes suivantes, représentées à la Figure 8:

1) comme pour 10.2;

2) comme pour 10.2;

3) comme pour 10.2;

4) B produit $r^B$, nombre non récurrent utilisé aux mêmes fins que $r^A$;

5) B envoie en A le jeton d'authentification suivant:

$$B\{t^B,\ r^B,\ A,\ r^A\}$$

où $t^B$ est une indication horaire, définie comme $t^A$.

Ou bien, si l'authentification d'origine des données de "sgnData" doit être fournie par la signature numérique:

$$B\{t^B,\ r^B,\ A,\ r^A,\ sgnData\}$$

Si l'information à transmettre est utilisée par la suite comme clé privée (cette information est désignée par "encData"):

$$B\{t^B,\ r^B,\ A,\ r^A,\ sgnData,\ Ap[encData]\}$$

L'emploi de "encData" comme clé privée implique un choix approprié, par exemple, pour obtenir une clé solide pour tout système cryptographique utilisé comme indiqué dans le champ "sgnData" du jeton;

6) A effectue les opérations suivantes:

   a) vérifie la signature et ainsi l'intégrité de l'information signée;

   b) vérifie que A est bien le destinataire prévu;

   c) vérifie que la date et l'heure $t^B$ sont "à jour";

   d) facultativement, vérifie que $r^B$ n'a pas été soumis à un redéfilement [voir en 10.2 l'étape 3) d)].

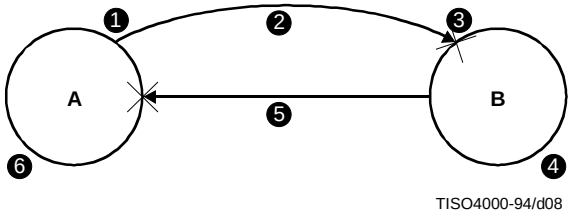



**Figure 8 – Authentification à deux voies**

## 10.4 Authentification à trois voies

L'authentification à trois voies comprend les étapes suivantes, représentées à la Figure 9:

1) comme pour 10.3;

2) comme pour 10.3. L'indication horaire $t^A$ peut avoir la valeur zéro;

3) comme pour 10.3, excepté qu'il n'est pas nécessaire de vérifier la date ni l'heure;

4) comme pour 10.3;

5) comme pour 10.3. L'indication horaire $t^B$ peut avoir la valeur zéro;

6) comme pour 10.3, excepté qu'il n'est pas nécessaire de vérifier la date ni l'heure;

7) A vérifie que le nombre $r^A$ reçu est identique au nombre $r^A$ émis;

8) A envoie en B le jeton d'authentification suivant:

$$A\{r^B, B\}$$

9) B effectue les opérations suivantes:

   a) vérifie la signature et ainsi vérifie l'intégrité de l'information signée;

   b) vérifie que le nombre $r^B$ reçu est identique au nombre $r^B$ émis par B.

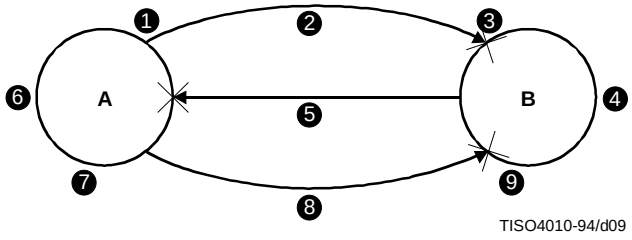



**Figure 9 – Authentification à trois voies**

# 11 Gestion des clés et des certificats

## 11.1 Production de paires de clés

La politique de gestion de sécurité totale d'une mise en œuvre définira le cycle de vie des paires de clés et se trouve donc hors du domaine d'application du cadre d'authentification. Cependant, il est vital pour la sécurité globale que les clés privées ne soient connues que de l'utilisateur auquel ces clés appartiennent.

Il n'est pas facile pour un utilisateur qui est un être humain de se souvenir des données des clés. Aussi convient-il d'employer une méthode appropriée pour mémoriser ces données de manière à les transporter facilement. Un mécanisme réalisable consisterait à utiliser une "carte à mémoire". Elle conserverait les clés privée et (facultativement) publique de l'utilisateur, le certificat d'utilisateur et une copie de la clé publique de l'autorité de certification. L'emploi de cette carte devrait comporter une sécurité complémentaire, par exemple par l'utilisation d'un numéro d'identification personnel (PIN, *personal identification number*), ce qui augmenterait la sécurité du système en exigeant de l'utilisateur qu'il possède la carte et qu'il sache comment y accéder. La méthode à choisir pour mémoriser ces données n'entre pas dans le cadre de la présente Spécification d'Annuaire.

Il y a trois façons de produire une paire de clés d'utilisateur:

a) l'utilisateur produit sa propre paire de clés – Cette méthode a l'avantage de ne jamais livrer une clé privée d'utilisateur à une autre entité, mais elle nécessite un certain niveau de compétence de la part de l'utilisateur comme indiqué dans l'Annexe D;

b) la paire de clés est produite par un tiers – Celui-ci doit livrer la clé privée à l'utilisateur de manière matérielle et sûre, puis détruire résolument toute information relative à la création de la paire de clés, ainsi que les clés elles-mêmes. Il conviendra de prendre les mesures de sécurité matérielles convenables, de manière que le tiers et les opérations sur les données soient exempts de toute fraude;

c) la paire de clés est produite par l'autorité de certification – C'est un cas particulier du point b) et les mêmes considérations s'appliquent dans ce cas.

NOTE – L'autorité de certification présente déjà un potentiel de confiance auprès de l'utilisateur et sera soumise aux mesures matérielles de sécurité nécessaires. Cette méthode a l'avantage de ne pas nécessiter de transfert en sûreté de données à l'autorité de certification.

Le système cryptographique en usage impose des contraintes (techniques) particulières à la production des clés.

## 11.2 Gestion des certificats

Un certificat associe la clé publique et le nom distinctif unique de l'utilisateur. De la sorte:

a) une autorité de certification doit s'assurer de l'identité d'un utilisateur avant de créer un certificat à son intention;

b) une autorité de certification ne doit pas délivrer de certificat pour deux utilisateurs ayant le même nom.

La production d'un certificat se présente comme une opération indépendante et ne doit pas être effectuée au moyen d'un mécanisme à question/réponse automatique. L'avantage de cette certification est que la clé privée de l'autorité de certification (CA) n'est jamais connue si ce n'est de cette autorité de certification, isolée et matériellement sûre. Le secret de la clé privée ne peut donc être percé que par une attaque contre l'autorité de certification elle-même, ce qui rend une compromission improbable.

Il importe que le transfert d'information à l'autorité de certification ne soit pas une source de compromission et il convient que des mesures matérielles de sécurité appropriées soient prises. A cet égard:

a) ce serait un grave manquement à la sécurité si l'autorité de certification livrait un certificat à un utilisateur ayant une clé publique qui aurait été falsifiée;

b) si le moyen de produire des paires de clés selon 11.1 b) ou 11.1 c) est employé, la clé privée de l'utilisateur doit être transférée à l'autre utilisateur de façon sûre;

c) en cas d'emploi du moyen de production des paires de clés du 11.1 a) ou du 11.1 b), l'utilisateur peut recourir à diverses méthodes (directes ou différées) pour communiquer sa clé publique à l'autorité de certification en toute sécurité. Les méthodes "en ligne" sont parfois plus souples pour les opérations effectuées à distance entre l'utilisateur et l'autorité de certification.

Un certificat est un élément d'information publiquement disponible et aucune mesure particulière de sécurité n'a besoin d'être prise en ce qui concerne son transport à l'Annuaire. Etant donné qu'il est produit par une autorité de certification indépendante, au nom d'un utilisateur qui en recevra copie, il suffit à l'utilisateur de mémoriser cette information dans son entrée d'annuaire lors d'un accès subséquent à l'Annuaire. Ou bien l'autorité de certification pourra conserver le certificat pour l'utilisateur et, dans ce cas, il convient de donner à cet agent des droits d'accès appropriés.

Les certificats auront une durée de vie qui leur sera associée et à la fin de laquelle leur validité expirera. Pour assurer la continuité du service, l'autorité de certification fera en sorte que des certificats de remplacement soient disponibles à temps pour remplacer les certificats dont la validité expire (ou a expiré). Les deux aspects suivants sont liés à cette procédure:

– la validité des certificats peut être prévue de manière que chacun devienne valide au moment où expire la validité du précédent, mais un chevauchement des validités peut être autorisé. Dans ce dernier cas, cela épargne à l'autorité de certification la nécessité d'installer et de distribuer un grand nombre de certificats qui peuvent ne plus être en vigueur à la date d'expiration;

– les certificats dont la validité a expiré sont en principe enlevés de l'Annuaire. Il incombe à l'autorité de certification de conserver les anciens certificats pendant quelque temps s'il existe un service de non-rejet des données.

Les certificats peuvent être annulés avant d'être périmés, par exemple si l'on suppose que la clé privée de l'utilisateur a fait l'objet d'une compromission, ou si l'utilisateur ne doit plus être certifié par l'autorité de certification, ou encore si l'on suppose que le certificat de l'autorité de certification a donné lieu à une compromission. Les quatre aspects suivants sont liés à cette procédure:

– l'annulation d'un certificat d'utilisateur ou d'un certificat de CA sera communiquée par l'autorité de certification et un nouveau certificat sera mis à disposition, si besoin est. L'autorité de certification peut alors informer l'entité titulaire du certificat que celui-ci est annulé en appliquant une procédure différée;

– l'autorité de certification conservera:

  a) une liste datée des certificats qu'elle a émis et qui ont été annulés;

  b) une liste datée des certificats annulés de toutes les CA dont elle a connaissance, qu'elle a certifiés.

  Ces deux listes certifiées doivent exister, même si elles sont vides;

– le maintien des entrées d'Annuaire affectées par les listes d'annulation de l'autorité de certification incombe à l'Annuaire et à ses utilisateurs, qui agiront en conformité avec les dispositions de sécurité adoptées. Par exemple, l'utilisateur peut modifier son entrée d'objet en remplaçant l'ancien certificat par un nouveau. Ce dernier servira alors à authentifier l'utilisateur vis-à-vis de l'Annuaire;

– les listes d'annulation ("listes noires") sont conservées dans les entrées comme des attributs des types "CertificateRevocationList" et "AuthorityRevocationList". Ces attributs peuvent être exploités selon les mêmes opérations que les autres attributs. Ces types d'attribut sont définis ainsi:

```
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 }

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

NOTE 1 – La vérification de l'ensemble de la liste des certificats est une question locale. On ne supposera la liste ordonnée d'une manière quelconque, à moins que des règles d'ordonnancement particulières n'aient été spécifiées par le CA émetteur, par exemple dans la politique de cette autorité de certification.

NOTE 2 – Si le service de non-rejet des données dépend des clés fournies par l'autorité de certification, le service doit s'assurer que toutes les clés pertinentes de l'autorité de certification (annulées ou périmées) et les listes d'annulation datées, sont archivées et certifiées par une autorité actuelle.

NOTE 3 – Si des extensions incluses dans une liste de type **CertificateList** sont définies comme étant critiques, l'élément **version** de cette liste doit être présent. Si aucune extension définie comme étant critique n'est incluse dans la liste, l'élément **version** sera absent. Ceci permettra à une implémentation qui ne prend en charge que les listes CRL de version 1 de continuer à utiliser une liste CRL tant qu'elle ne rencontrera pas d'extension au cours de l'examen de la séquence des certificats révoqués de cette liste. Une implémentation prenant en charge les listes CRL de version 2 (ou plus) peut en optimiser le traitement si elle détermine suffisamment tôt qu'aucune extension critique n'est présente dans cette liste.

NOTE 4 – Lorsqu'une implémentation qui traite une liste d'annulation de certificats ne reconnaît pas une extension critique dans le champ **crlEntryExtensions**, cette mise en œuvre doit partir du principe qu'au minimum le certificat identifié a été annulé et n'est plus valide. Elle doit ensuite effectuer les actions supplémentaires concernant ce certificat annulé, selon les instructions de la politique locale. Lorsqu'une implémentation ne reconnaît pas une extension critique dans le champ **crlExtensions**, elle doit partir du principe que les certificats identifiés ont été annulés et ne sont plus valides. Dans ce dernier cas cependant, comme la liste n'est peut-être pas complète, les certificats qui n'ont pas été identifiés comme étant annulés ne peuvent pas être considérés comme étant valides. Dans ce cas, c'est la politique locale qui doit indiquer les mesures à prendre. De toute façon, la politique locale peut indiquer des actions supplémentaires et/ou plus poussées que celles qui sont exposées dans la présente Spécification.

NOTE 5 – Si une extension affecte le traitement de la liste (par exemple si plusieurs listes CRL doivent être analysées pour établir la liste complète des certificats annulés ou si une entrée peut représenter une série de certificats), cette extension doit être indiquée comme étant critique dans le champ **crlExtensions**, quel que soit l'emplacement de l'extension dans la liste CRL. Une extension indiquée dans le champ **crlEntryExtensions** d'une entrée doit être placée dans cette entrée et ne doit affecter que le(s) certificat(s) spécifié(s) dans cette entrée.

NOTE 6 – Les extensions normales pour les listes CRL sont définies dans l'article 12.

# 12 Extensions de certificat et de liste CRL

## 12.1 Introduction

Cet article spécifie les extensions apportées dans les domaines suivants:

a) *informations de clé et de politique*: ces extensions de certificat et de liste CRL acheminent des informations supplémentaires sur les clés mises en jeu, y compris les identificateurs de clé d'utilisateur et d'émetteur, les indicateurs d'usage prévu ou restreint de clé, et les indicateurs de politique de certification;

b) *attribut d'entité titulaire de certificat et d'émetteur de certificat*: ces extensions de certificat et de liste CRL prennent en charge les variantes de dénomination, sous différentes formes nominatives, pour une entité titulaire de certificat, un émetteur de certificat ou un émetteur de liste CRL. Ces extensions peuvent également acheminer des informations supplémentaires d'attribut sur l'entité titulaire de certificat, afin de donner à l'utilisateur du certificat des assurances sur le fait que l'entité titulaire de certificat est une personne ou une entité existante;

c) *contraintes sur le chemin de certification*: ces extensions de certificat permettent de spécifier des contraintes dans des certificats d'autorité de certification, c'est-à-dire dans des certificats émis par des autorités de certification pour d'autres autorités de certification, afin de faciliter le traitement automatique des chemins de certification lorsque plusieurs politiques de certification sont en jeu. Cela se produit lorsque les politiques varient selon les applications dans un même environnement ou lorsqu'il y a lieu d'assurer l'interfonctionnement avec des environnements extérieurs. Les contraintes peuvent limiter les types de certificat qui peuvent être émis par l'autorité de certification en cause ou qui peuvent s'enchaîner dans un chemin de certification;

d) *extensions de base des listes CRL*: ces extensions de liste CRL permettent d'insérer dans une telle liste des indications sur les motifs d'annulation, d'effectuer une suspension provisoire d'un certificat, et d'attribuer des numéros aux listes CRL pour permettre aux utilisateurs de certificat de détecter des listes CRL manquantes dans une séquence issue d'un même 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 en listes CRL distinctes l'ensemble complet des informations d'annulation issues d'une même autorité de certification. Elles permettent également de combiner en une seule liste CRL les informations d'annulation issues de diverses autorités de certification. Ces extensions permettent également d'utiliser des listes CRL partielles (delta), qui n'indiquent que les modifications intervenues depuis une édition antérieure de liste CRL.

L'inclusion d'une extension quelconque dans un certificat ou dans une liste CRL relève d'une décision de l'autorité émettrice de ce certificat ou de cette liste CRL.

Dans un certificat ou dans une liste CRL, une extension est étiquetée comme étant critique ou non critique. Si une extension est étiquetée comme étant critique et qu'un système utilisateur de certificat ne reconnaisse pas le type du champ d'extension ou n'implémente pas les sémantèmes de l'extension, ce système doit considérer le certificat comme invalide. Si une extension est étiquetée comme étant non critique, un système utilisateur de certificat qui ne reconnaît ni n'implémente ce type d'extension peut traiter le reste du certificat sans tenir compte de l'extension. Les définitions des types d'extension figurant dans la présente Spécification d'Annuaire indiquent si l'extension est toujours critique, jamais critique, ou si la criticité peut être déterminée par l'émetteur du certificat ou de la liste CRL. La raison de la prescription que certaines extensions ne soient jamais critiques est de permettre aux implémentations utilisatrices de certificat, qui n'ont pas besoin d'utiliser de telles extensions, de ne pas en tenir compte sans remettre en question leur capacité d'interfonctionnement avec toutes les autorités de certification.

NOTE – Un système utilisateur de certificat peut exiger que certaines extensions non critiques soient présentes dans un certificat afin que celui-ci soit considéré comme acceptable. La nécessité d'inclure de telles extensions peut être dictée par les règles de politique locale de l'utilisateur du certificat ou par une règle de politique de l'autorité de certification, indiquée au système utilisateur de certificat par l'inclusion d'un identificateur particulier de politique de certification dans l'extension relative aux politiques de certification, cette extension étant alors étiquetée comme étant critique.

Pour toutes les extensions de certificat, de liste CRL et d'entrée de liste CRL définies dans la présente Spécification d'Annuaire, il ne doit pas y avoir plus d'une seule instance de chaque type d'extension dans, respectivement, un certificat, une liste CRL ou une entrée de liste CRL.

Cet article définit également des règles de mise en correspondance destinées à faciliter la sélection de certificats ou de listes CRL ayant des caractéristiques spécifiques, à partir d'attributs à valeurs multiples contenant plusieurs certificats ou listes CRL.

## 12.2 Informations de clé et de politique

### 12.2.1 Prescriptions

Les prescriptions suivantes s'appliquent aux informations de clé et de politique:

a) la mise à jour d'une paire de clés d'autorité de certification peut intervenir à intervalles réguliers ou dans des circonstances particulières. Il faut pour cela qu'un champ du certificat achemine un identificateur de la clé publique à utiliser pour vérifier la signature du certificat. Un système utilisateur de certificat peut utiliser de tels identificateurs pour trouver le certificat d'autorité de certification correct, permettant de valider la clé publique de l'émetteur du certificat;

b) en général, une entité titulaire de certificat possède différentes clés publiques et, corrélativement, des certificats pour différents objets, par exemple pour la signature numérique et pour l'agrément de clé de chiffrement. Un champ est nécessaire dans le certificat pour aider un utilisateur de certificat à sélectionner le certificat correct pour une entité donnée et à une fin donnée, ou pour permettre à une autorité de certification de stipuler qu'une clé certifiée ne peut être utilisée que pour un objectif particulier;

c) la mise à jour d'une paire de clés d'entité titulaire de certificat peut intervenir à intervalles réguliers ou dans des circonstances particulières. Il faut pour cela qu'un champ du certificat achemine un identificateur permettant de distinguer différentes clés publiques pour la même entité, utilisées à différents moments. Un système utilisateur de certificat peut utiliser de tels identificateurs pour trouver le certificat correct;

d) la clé privée correspondant à une clé publique certifiée est normalement utilisée pendant une période différente de celle de la clé publique. Avec les clés à signature numérique, la période d'utilisation de la clé privée de signature est normalement plus courte que celle de la clé publique de vérification. La période de validité du certificat indique une période pendant laquelle la clé publique peut être utilisée, cette période n'étant pas nécessairement la même que la période d'utilisation de la clé privée. En cas de compromission d'une clé privée, la période de risque peut être limitée si le vérificateur de signature connaît la période d'utilisation légitime de cette clé privée. Il est donc nécessaire que la période d'utilisation de la clé privée soit indiquée dans un certificat;

e) étant donné que les certificats peuvent être utilisés dans des environnements où de multiples politiques de certification s'appliquent, il est nécessaire de prévoir l'inclusion, dans les certificats, des informations de politique de certification;

f) lors d'une certification réciproque entre deux organisations, il est possible de convenir que certaines des politiques de ces deux organisations peuvent être considérées comme équivalentes. Un certificat d'autorité de certification doit permettre à son émetteur d'indiquer que l'une de ses propres politiques de certification est équivalente à une autre politique de certification, relevant du domaine de cette autorité de certification. Ce mappage est appelé équivalence de politiques;

g) l'utilisateur d'un système de chiffrement ou de signature numérique, utilisant les certificats définis dans la présente Spécification d'Annuaire, doit être en mesure de déterminer d'avance les algorithmes pris en charge par d'autres utilisateurs.

### 12.2.2 Champs des extensions de certificat et de liste CRL

Les champs d'extension suivants sont définis:

a) *identificateur de clé d'autorité*;

b) *identificateur de clé d'entité*;

c) *type d'utilisation de clé publique*;

d) *période d'utilisation de clé privée*;

e) *politiques de certification*;

f) *mappages de politiques*.

Ces champs d'extension ne doivent être utilisés que dans des certificats, sauf pour l'identificateur de clé d'autorité, qui peut aussi être utilisé comme extension de liste CRL. Sauf indication contraire, ces extensions peuvent être utilisées aussi bien dans les certificats d'autorité de certification que dans les certificats d'entité finale.

Par ailleurs, un attribut d'Annuaire est défini pour prendre en charge la sélection de l'algorithme à utiliser pour communiquer avec une entité finale distante qui utilise des certificats conformes à la présente Spécification d'Annuaire.

#### 12.2.2.1 Champ d'identificateur de clé d'autorité

Ce champ, qui peut servir d'extension de certificat ou d'extension de liste CRL, indique la clé publique à utiliser pour vérifier la signature relative à ce certificat ou à cette liste CRL. Il permet à une autorité de certification donnée de distinguer les différentes clés qu'elle utilise (par exemple lorsqu'une mise à jour de clé intervient). Ce champ 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 identifiée par un identificateur explicite de clé, contenu dans la composante **keyIdentifier**, par l'identification d'un certificat pour cette clé (indiquant l'émetteur du certificat dans la composante **authorityCertIssuer** et le numéro de série dans la composante **authorityCertSerialNumber**) ou par ces deux moyens: identificateur explicite de clé et identification d'un certificat pour la clé. Si ces deux formes d'identification sont utilisées, l'émetteur du certificat ou de la liste CRL doit en vérifier la cohérence. Un identificateur de clé doit être unique par rapport à tous les identificateurs de clé dans le cadre du certificat ou de la liste CRL contenant l'extension pour l'autorité émettrice. Une implémentation qui prend en charge cette extension n'est pas tenue de pouvoir traiter toutes les formes nominatives contenues dans la composante **authorityCertIssuer**. (On trouvera au 12.3.2.1 des détail sur le type **GeneralNames**.)

Les autorités de certification doivent attribuer des numéros en série aux certificats de manière que chaque paire (émetteur-numéro de série de certificat) désigne un seul et même certificat.

Cette extension n'est jamais critique.

#### 12.2.2.2 Champ d'identificateur de clé d'entité

Ce champ désigne la clé publique qui est en cours de certification. Il permet à la même entité de différencier les différentes clés qu'elle utilise (par exemple lors d'une mise à jour de clé). Ce champ est défini comme suit:

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

SubjectKeyIdentifier            ::= KeyIdentifier
```

Un identificateur de clé doit être unique par rapport à tous les identificateurs de clé pour l'entité avec laquelle il est utilisé. Cette extension n'est jamais critique.

#### 12.2.2.3 Champ de type d'utilisation de clé publique

Ce champ, qui indique la fin à laquelle la clé publique certifiée est utilisée, 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) }
```

Les bits contenus dans le type **KeyUsage** sont les suivants:

a) **digitalSignature**: ce bit permet de vérifier les signatures numériques qui ont des fins autres que celles qui sont indiquées en b), f) ou g) ci-dessous;

b) **nonRepudiation**: ce bit permet de vérifier les signatures numériques utilisées pour fournir un service de non-rejet protégeant contre un faux déni, de la part de l'entité signataire, d'une action quelconque (à l'exception d'une signature de certificat ou de liste CRL, comme en f) ou en g) ci-dessous);

c) **keyEncipherment**: ce bit permet de chiffrer des clés ou d'autres informations de sécurité, par exemple pour le transport de clés;

d) **dataEncipherment**: ce bit permet de chiffrer des données d'utilisateur, mais non des clés ou d'autres informations de sécurité, comme en c) ci-dessus;

e) **keyAgreement**: ce bit est utilisé comme clé publique d'agrément de clé;

f) **keyCertSign**: ce bit permet de vérifier une signature d'autorité de certification concernant des certificats;

g) **cRLSign**: ce bit permet de vérifier une signature d'autorité de certification concernant des listes CRL;

h) **encipherOnly**: bit d'acceptation de clé publique pour utilisation seulement dans des données de chiffrage avec bit **keyAgreement** également activé (c'est-à-dire que le bit d'utilisation de l'autre clé est mis à "valeur non définie");

i) **decipherOnly**: bit d'acceptation de clé publique pour utilisation seulement dans des données de déchiffrage avec bit **keyAgreement** également activé (c'est-à-dire que le bit d'utilisation de l'autre clé est mis à "valeur non définie").

Le bit **keyCertSign** n'est utilisé que dans des certificats d'autorité de certification. Si l'élément **KeyUsage** est mis à la valeur **keyCertSign** et que l'extension de contraintes de base est présente dans le même certificat, la valeur de la composante **cA** de cette extension doit être mise à la valeur **Vrai**. Les autorités de certification peuvent aussi définir dans l'élément **KeyUsage** d'autres utilisations des bits d'utilisation de clé. Elles peuvent par exemple utiliser le bit **digitalSignature** pour assurer l'authentification et l'intégrité de transactions administratives directes.

Cette extension peut, sur option de l'émetteur de certificat, être soit critique soit non critique.

Si l'extension est étiquetée comme étant critique, le certificat ne doit être utilisé que pour une fin pour laquelle le bit d'utilisation de clé correspondant est mis à 1.

Si l'extension est étiquetée comme étant non critique, ce champ indique la ou les fins prévues pour la clé et peut être utilisé pour trouver la clé ou le certificat correct d'une entité qui possède plusieurs clés ou certificats. Il s'agit d'un champ de type consultatif qui n'implique pas la limitation de l'usage de la clé à la fin indiquée. Un bit mis à zéro indique que la clé n'est pas destinée à cette fin. Si tous les bits sont à zéro, cela indique que la clé est prévue pour une fin autre que celles qui sont énumérées.

#### 12.2.2.4 Champ de type d'utilisation de clé publique étendu

Ce champ indique une ou plusieurs fins auxquelles la clé publique certifiée peut être utilisée, en plus ou à la place des fins de base indiquées dans le champ d'extension de type d'utilisation de clé publique. Ce champ est défini comme suit:

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

KeyPurposeId ::= OBJECT IDENTIFIER
```

Des fins peuvent être définies par toute organisation ayant un besoin donné. Les identificateurs d'objet utilisés pour identifier ces fins doivent être attribués conformément à la Rec. X.660 du CCITT | ISO/CEI 9834-1.

Cette extension peut, sur option de l'émetteur de certificat, être soit critique soit non critique.

Si l'extension est étiquetée comme étant critique, le certificat ne doit être utilisé qu'à l'une des fins indiquées.

Si l'extension est étiquetée comme étant non critique, ce champ indique la ou les fins prévues pour la clé et peut être utilisé pour trouver la clé ou le certificat correct d'une entité qui possède plusieurs clés ou certificats. Il s'agit d'un champ de type consultatif qui n'implique pas que l'usage de la clé est limité par l'autorité de certification à la fin indiquée. (Les applications utilisatrices peuvent néanmoins nécessiter qu'une fin donnée soit indiquée pour que le certificat soit acceptable pour ces applications.)

Si un certificat contient à la fois un champ de type d'utilisation de clé publique critique et un champ de type d'utilisation de clé publique étendu critique, les deux champs doivent être traités de manière indépendante et le certificat ne doit être utilisé qu'à une fin compatible avec les deux champs. Si aucune fin n'est compatible avec les deux champs, le certificat ne doit être utilisé à aucune fin.

#### 12.2.2.5 Champ de période d'utilisation de clé privée

Ce champ indique la période d'utilisation de la clé privée correspondant à la clé publique certifiée. Il n'est applicable qu'aux clés de signature numérique et 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} )
```

La composante **notBefore** (pas avant le … à …) indique la date et l'heure à partir desquelles la clé privée pourra être utilisée pour la signature. Si la composante **notBefore** n'est pas présente, aucune information n'est donnée quant au moment à partir duquel commence la période de validité de la clé privée. La composante **notAfter** (pas après le … à …) indique la date et l'heure après lesquelles la clé privée ne pourra plus être utilisée pour la signature. Si la composante **notAfter** n'est pas présente, aucune information n'est donnée quant au moment d'expiration de la période de validité de la clé privée.

Cette extension n'est jamais critique.

NOTE 1 – La période de validité de la clé privée peut être différente de la validité certifiée de la clé publique, telle qu'indiquée par la période de validité du certificat. La période d'utilisation des clés numériques de signature est normalement plus courte que celle des clés publiques de vérification.

NOTE 2 – Si le vérificateur d'une signature numérique veut vérifier que la clé n'a pas été annulée, par exemple en raison d'une compromission de clé avant le moment de la vérification, un certificat valide doit exister pour la clé publique au moment de la vérification. Après l'expiration du ou des certificats pour une clé publique, un vérificateur de signature ne peut plus accepter de compromis notifiés par des listes CRL.

#### 12.2.2.6 Champ de politiques de certification

Ce champ énumère les politiques de certification reconnues par l'autorité de certification émettrice et applicables aux certificats, ainsi que les informations facultatives qualifiant ces politiques de certification. Normalement, différentes politiques de certification se rapporteront à différentes applications pouvant utiliser la clé certifiée. 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** identifie et achemine les informations de qualification concernant une politique de certification donnée. La composante **policyIdentifier** contient un identificateur de politique de certification et la composante **policyQualifiers** contient des valeurs de qualificateur de politique pour cet élément.

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

Si l'extension est étiquetée comme étant critique, cela indique que le certificat ne doit être utilisé que pour son objet et conformément aux règles impliquées par l'une des politiques de certification indiquées. Les règles d'une politique particulière peuvent prescrire que le système utilisateur du certificat traite la valeur du qualificateur d'une manière particulière.

Si l'extension est étiquetée comme n'étant pas critique, son utilisation ne contraint pas nécessairement l'usage du certificat selon les politiques énumérées. Un utilisateur de certificat peut toutefois exiger qu'une politique particulière soit présente afin d'utiliser le certificat (voir 12.4.3). Les qualificateurs de politique peuvent, au choix de l'utilisateur du certificat, être traités ou être ignorés.

Les types des politiques de certification et de leurs qualificateurs peuvent être définis par toute organisation qui en a besoin. Les identificateurs d'objet utilisés pour identifier des types de politiques de certification et des types de qualificateurs de politique de certification doivent être attribués conformément à la Rec. X.660 du CCITT | ISO/CEI 9834-1. La classe d'objets ASN.1 suivante est utilisée pour définir les types de qualificateur de politique de certification:

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

Une définition de type de qualificateur de politique doit comporter:

- une déclaration sur la sémantique des valeurs possibles;
- une indication précisant si l'identificateur de qualificateur peut apparaître dans une extension de politique de certification sans valeurs d'accompagnement et, si tel est le cas, indiquant la sémantique impliquée en l'occurrence.

NOTE – Un qualificateur peut être spécifié comme étant d'un type ASN.1 quelconque. Lorsqu'il est prévu d'utiliser surtout le qualificateur avec des applications qui ne possèdent pas de fonctions de décodage ASN.1, il est recommandé de spécifier le type **OCTET STRING**. La valeur en notation ASN.1 du type **OCTET STRING** peut en effet acheminer une valeur de qualificateur codée conformément à une convention quelconque, spécifiée par l'élément de politique définissant l'organisation.

#### 12.2.2.7 Champ de mappage de politiques

Ce champ, qui ne doit être utilisé que dans les certificats d'autorité de certification, permet à un émetteur de certificat d'indiquer qu'aux fins de l'utilisateur d'un chemin de certification contenant ce certificat, l'une des politiques de certification de l'émetteur peut être considérée comme équivalente à une autre politique de certification, utilisée dans le domaine de l'autorité de certification en cause. 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 }
```

La composante **issuerDomainPolicy** indique une politique de certification qui est reconnue dans le domaine de l'autorité de certification émettrice et qui peut être considérée comme équivalente à la politique de certification indiquée dans la composante **subjectDomainPolicy** qui est reconnue dans le domaine de l'autorité de certification en cause.

Cette extension n'est jamais critique.

NOTE 1 – Un exemple de mappage de politiques est donné ci-dessous. Le domaine du gouvernement des Etats-Unis d'Amérique peut avoir une politique dite *CanadianTrade* et le gouvernement canadien peut avoir une politique appelée *U.S. Trade*. Bien que ces deux politiques soient clairement identifiées et définies, les deux gouvernements peuvent s'entendre pour accepter des chemins de certification allant au-delà de leurs propres frontières, dans le cadre des règles impliquées par ces politiques à des fins particulières.

NOTE 2 – Le mappage de politiques implique de notables corollaires administratifs et la mise en œuvre d'un personnel suffisamment qualifié et habilité à prendre les décisions nécessaires. En général, il est préférable de convenir d'un usage des politiques communes plus étendu que pour appliquer le mappage des politiques. Dans l'exemple précédent, il serait préférable que les Etats-Unis d'Amérique, le Canada et le Mexique conviennent d'une politique commune pour les échanges commerciaux en Amérique du Nord (*North American Trade*).

NOTE 3 – L'on s'attend que le mappage de politiques ne sera réalisable que dans des environnement limités, dans lesquels les déclarations de politiques sont très simples.

#### 12.2.2.8 Attribut d'algorithmes pris en charge

On définit un attribut d'Annuaire pour prendre en charge la sélection d'un algorithme à utiliser lors de la communication avec une entité finale distante, utilisant des certificats comme défini dans la présente Spécification d'Annuaire. La notation ASN.1 suivante définit cet attribut (à valeurs multiples):

```
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 }
```

Chaque valeur de l'attribut à valeurs multiples doit correspondre à une valeur distincte de la composante **algorithmIdentifier**. La valeur de la composante **intendedUsage** donne une indication sur l'usage prévu de l'algorithme (voir 12.2.2.3 pour les usages reconnus). La valeur de la composante **intendedCertificatePolicies** identifie les politiques de certification et, facultativement, les qualificateurs de politique avec lesquels l'algorithme identifié peut être utilisé.

## 12.3 Attributs d'entité titulaire de certificat et d'émetteur de certificat

### 12.3.1 Prescriptions

Les prescriptions suivantes se rapportent aux attributs d'entité titulaire de certificat et d'émetteur de certificat:

a) les certificats doivent être utilisables par les applications qui font appel à diverses formes nominatives, y compris les noms de courrier électronique par Internet, les noms de domaines Internet, les adresses d'expéditeur ou destinataire X.400 et les noms des partenaires de transfert électronique de données (EDI, *electronic data interchange*). Il est donc nécessaire de pouvoir associer de manière sûre de multiples noms ayant divers formats avec une entité titulaire de certificat ou avec un émetteur de certificat ou liste CRL;

b) un utilisateur de certificat peut avoir besoin de connaître de manière sûre certaines informations d'identification sur une entité, afin d'avoir l'assurance que cette entité est bien la personne ou la chose recherchée. Par exemple, des informations telles que l'adresse postale, le poste occupé dans l'entreprise ou la photographie peuvent être requises. De telles informations peuvent être aisément représentées par des attributs d'annuaire, mais ces attributs ne font pas forcément partie du nom distinctif. Un champ du certificat est donc nécessaire pour acheminer d'autres attributs d'annuaire que ceux du nom distinctif.

### 12.3.2 Champs d'extension de certificat ou de liste CRL

Les champs d'extension suivants sont définis:

a) *variante nominative (alias) d'entité*;

b) *variante nominative (alias) d'émetteur*;

c) *attributs d'annuaire d'une entité*.

Ces champs ne doivent être utilisés que comme des extensions de certificat, sauf dans le cas de la variante nominative d'émetteur, qui peut également être utilisée comme extension de liste CRL. En tant qu'extensions de certificat, ces champs peuvent être présents dans des certificats d'autorité de certification ou dans des certificats d'entité finale.

#### 12.3.2.1 Champ de variante nominative d'entité

Ce champ contient une ou plusieurs variantes nominatives, faisant appel à diverses formes nominatives, pour l'entité qui est associée par l'autorité de certification à la clé publique certifiée. Ce champ 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 contenues dans les variantes du type **GeneralName** sont des noms de diverses formes comme suit:

- **otherName** est un nom de forme quelconque, défini comme une instance de la classe d'objets informationnels **OTHER-NAME**;
- **rfc822Name** est une adresse de courrier 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 ayant une forme convenue entre partenaires de transfert électronique de données (EDI); la composante **nameAssigner** désigne une autorité qui attribue des valeurs nominatives uniques dans la composante **partyName**;
- **uniformResourceIdentifier** est un identificateur de ressource uniforme pour le réseau mondial WWW, défini conformément au document Internet RFC 1630;
- **iPAddress** est une adresse de protocole Internet définie conformément au document Internet RFC 791, représentée par une chaîne d'éléments binaires;
- **registeredID** est un identificateur d'objet enregistré quelconque, attribué conformément à la Rec. X.660 du CCITT | ISO/CEI 9834-1.

Pour chaque forme nominative utilisée dans le type **GeneralName**, il doit exister un système d'enregistrement de nom garantissant que chaque nom utilisé désigne sans équivoque une seule entité, aussi bien auprès de l'émetteur qu'auprès des utilisateurs du certificat.

Cette extension peut, au choix de l'émetteur du certificat, être soit critique soit non critique. Une mise en œuvre qui prend en charge cette extension n'est pas tenue de traiter toutes les formes nominatives. Si l'extension est étiquetée comme étant critique, au moins une des formes nominatives présentes doit être reconnue et traitée, à défaut de quoi le certificat est considéré comme invalide. En dehors de la restriction précédente, un système utilisateur de certificat est autorisé à ignorer tout nom dont la forme n'est pas reconnue ou pas prise en charge. Il est recommandé que ce champ soit étiqueté comme étant non critique, à condition que le champ d'entité d'annuaire du certificat contienne un nom d'annuaire identifiant sans ambiguïté l'entité.

NOTE 1 – L'utilisation de la classe **TYPE-IDENTIFIER** 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 est étiqueté comme étant critique, le champ **subject** du certificat ou de la liste CRL peut contenir un nom de valeur nulle (par exemple des noms distinctifs relatifs formés d'une séquence de zéros). Dans ce cas, l'entité n'est identifiée que par le ou les noms contenus dans cette extension.

#### 12.3.2.2 Champ de variante nominative d'émetteur

Ce champ contient une ou plusieurs variantes nominatives, utilisant l'une quelconque de diverses formes nominatives, pour l'émetteur du certificat ou de la liste CRL.

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

Cette extension peut, au choix de l'émetteur du certificat ou de la liste CRL, être soit critique soit non critique. Une implémentation qui prend en charge cette extension n'est pas tenue de traiter toutes les formes nominatives. Si l'extension est étiquetée comme étant critique, au moins une des formes nominatives présentes doit être reconnue et traitée, à défaut de quoi le certificat ou la liste CRL est considéré comme invalide. En dehors de la restriction précédente, un système utilisateur de certificat est autorisé à ignorer tout nom dont la forme n'est pas reconnue ou pas prise en charge. Il est recommandé que ce champ soit étiqueté comme étant non critique, à condition que le champ d'émetteur du certificat ou de la liste CRL contienne un nom d'annuaire identifiant sans ambiguïté l'autorité émettrice.

NOTE – Si ce champ d'extension est présent et est étiqueté comme étant critique, le champ **issuer** du certificat ou de la liste CRL peut contenir un nom de valeur nulle (par exemple des noms distinctifs relatifs formés d'une séquence de zéros). Dans ce cas, l'émetteur n'est identifié que par le ou les noms contenus dans cette extension.

#### 12.3.2.3 Champ d'attributs d'annuaire d'entité

Ce champ achemine toutes valeurs souhaitées d'attribut d'Annuaire pour l'entité titulaire du certificat. Ce champ est défini comme suit:

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

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

Cette extension n'est jamais critique.

## 12.4 Contraintes sur le chemin de certification

### 12.4.1 Prescriptions

Pour le traitement du chemin de certification:

a) il faut que les certificats d'entité finale puissent être distingués des certificats d'autorité d'authentification afin d'éviter que des entités finales ne s'établissent elles-mêmes comme autorités de certification sans autorisation. Il faut également qu'une autorité de certification puisse limiter la longueur d'une chaîne résultant d'une autorité de certification constituant une entité certifiée, par exemple pas plus d'un seul certificat supplémentaire ou pas plus de deux certificats supplémentaires;

b) une autorité de certification doit être en mesure de spécifier des contraintes permettant à un utilisateur de certificat de vérifier que des autorités de certification offrant un moindre degré de fiabilité dans un chemin de certification (c'est-à-dire des autorités de certification se trouvant plus en aval dans le chemin de certification commençant par l'autorité de certification dont l'utilisateur de certificat utilise la clé publique pour commencer ses opérations) ne violent pas la règle de confiance dont elles bénéficient en émettant des certificats pour des entités situées dans un espace nominatif inapproprié. L'observation de ces contraintes doit pouvoir être vérifié automatiquement par l'utilisateur du certificat;

c) le traitement du chemin de certification doit pouvoir être mis en œuvre dans un module automatisé et autonome. Cela est nécessaire pour que l'on puisse mettre en œuvre des modules matériels ou logiciels fiabilisés, accomplissant les fonctions de traitement du chemin de certification;

d) il doit être possible de mettre en œuvre le traitement du chemin de certification indépendamment d'interactions en temps réel avec l'utilisateur local;

e) il doit être possible de mettre en œuvre le traitement du chemin de certification indépendamment de l'utilisation de bases de données locales fiabilisées, contenant des informations descriptives de politiques. (Certaines informations locales fiabilisées – au moins une clé publique initiale – sont nécessaires pour le traitement du chemin de certification mais il y a lieu de minimiser la quantité de ces informations);

f) les chemins de certification ont besoin de fonctionner dans des environnements où plusieurs politiques de certification 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 à quelles fins. La concaténation de plusieurs domaines de politique doit être possible;

g) une flexibilité totale des modèles habilités est requise. Un modèle hiérarchique strict, convenant à une organisation donnée, n'est pas approprié lorsque l'on considère les besoins de multiples entreprises interconnectées. La flexibilité est requise lors de la sélection de la première autorité de certification fiabilisée dans un chemin de certification. Il doit en particulier être possible de prescrire que le chemin de certification commence dans le domaine de sécurité local du système utilisateur de clé publique;

h) les structures de dénomination ne doivent pas être contraintes d'utiliser des noms dans les certificats. En d'autres termes, les structures nominatives d'Annuaire considérées comme naturelles pour certaines organisations ou zones géographiques ne doivent pas avoir besoin 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 rétrocompatibles avec le système de type chemin de certification non contraint, tel que spécifié dans des éditions antérieures de la présente Recommandation | Norme internationale;

j) une autorité de certification doit pouvoir interdire l'utilisation des mappages de politiques et doit pouvoir prescrire la présence – dans les certificats ultérieurs d'un chemin de certification – d'identificateurs explicites de la politique de certification.

NOTE – Dans tout système utilisateur de certificat, le traitement d'un chemin de certification nécessite un niveau d'assurance approprié. La présente Spécification d'Annuaire définit des fonctions qui peuvent être utilisées dans des mises en œuvre appelées à être conformes à des déclarations d'assurance spécifiques. Par exemple, une prescription d'assurance peut déclarer que le chemin de certification doit toujours être protégé contre la manipulation du processus (comme l'intrusion logicielle ou la modification de données). Le niveau d'assurance doit être compatible avec le risque commercial. Par exemple:

a) un traitement interne par un module cryptographique approprié peut être requis pour des clés publiques utilisées afin de valider des transferts de fonds de montant élevé;

b) tandis qu'un traitement logiciel peut convenir pour des demandes d'état de compte bancaire à domicile.

Par conséquent, les fonctions de traitement du chemin de certification doivent pouvoir être mises en œuvre dans des modules cryptographiques matériels ou dans des jetons cryptographiques, pour ne citer que cette option.

### 12.4.2 Champs d'extension de certificat

Les champs d'extension suivants sont définis:

a) *contraintes de base*;

b) *contraintes de nom*;

c) *contraintes de politique*.

Ces champs d'extension ne doivent être utilisés que pour des extensions 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 aussi être utilisées dans des certificats d'entité finale. L'Annexe L contient des exemples d'utilisation de ces contraintes.

#### 12.4.2.1 Champ de contraintes de base

Ce champ indique si l'entité peut faire office d'autorité de certification, la clé publique certifiée étant utilisée pour vérifier des signatures de certificat. Dans ce cas, une contrainte de longueur du chemin de certification peut aussi être spécifiée. 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 }
```

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

La composante **pathLenConstraint** ne doit être présente que si la composante **cA** est mise à la valeur Vrai. Elle indique le nombre maximal de certificats d'autorité de certification qui peuvent venir à la suite de ce certificat dans un chemin de certification. La valeur 0 indique que l'entité titulaire de ce certificat ne peut émettre des certificats que pour des entités finales et non pour d'autres autorités de certification. Si aucun champ **pathLenConstraint** n'apparaît dans un certificat de chemin de certification, aucune limite n'est imposée à la longueur autorisée du chemin de certification.

Cette extension peut, au choix de l'émetteur du certificat, être soit critique soit non critique. Il est recommandé qu'elle soit étiquetée comme critique, sinon une entité non autorisée à être une autorité de certification pourrait émettre des certificats et un système utilisateur de certificat pourrait utiliser par inadvertance un tel certificat.

Si cette extension est présente et est étiquetée comme critique:

- si la valeur de la composante **cA** n'est pas mise à Vrai, la clé publique certifiée ne doit pas être utilisée pour vérifier une signature de certificat;
- si la valeur de la composante **cA** est mise à Vrai et que la composante **pathLenConstraint** soit présente, le système utilisateur du certificat doit vérifier que le chemin de certification en cours de traitement est compatible avec la valeur de la composante **pathLenConstraint**.

NOTE 1 – Si cette extension n'est pas présente ou est étiquetée comme non critique et n'est pas reconnue par un système utilisateur de certificat, il y a lieu que ce système utilise d'autres moyens pour déterminer si la clé publique certifiée peut être utilisée pour vérifier des signatures de certificat.

NOTE 2 – Pour contraindre une entité titulaire de certificat à n'être qu'une entité finale, c'est-à-dire à ne pas être une autorité de certification, l'émetteur peut inclure ce champ d'extension avec une valeur vide de la composante **SEQUENCE**.

#### 12.4.2.2 Champ de contraintes de nom

Ce champ, qui ne doit être utilisé que dans un certificat d'autorité de certification, indique un espace nominatif à l'intérieur duquel doivent se trouver tous les noms d'entité contenus dans les certificats ultérieurs d'un chemin 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)
```

Si elles sont présentes, les composantes **permittedSubtrees** et **excludedSubtrees** spécifient chacune un ou plusieurs sous-arbres de dénomination, définis chacun par le nom de sa racine et, facultativement, par une zone limitée par des niveaux supérieurs ou inférieurs dans chaque sous-arbre. Si la composante **permittedSubtrees** est présente, seuls sont acceptables les certificats, parmi tous les certificats émis par l'autorité de certification titulaire et par les autorités de certification subséquentes dans le chemin de certification, qui possèdent des noms d'entité à l'intérieur de ces sous-arbres. Si la composante **excludedSubtrees** est présente, aucun certificat émis par l'autorité de certification titulaire ou par des autorités de certification subséquentes dans le chemin de certification, n'est acceptable s'il possède un nom d'entité à l'intérieur de ces sous-arbres. Si les deux composantes **permittedSubtrees** et **excludedSubtrees** sont présentes et que les espaces nominatifs se superposent, la déclaration d'exclusion a priorité.

Parmi les formes nominatives disponibles dans le type **GeneralName**, seules celles qui ont une structure hiérarchique bien définie peuvent être utilisées dans ces domaines. La forme nominative **directoryName** répond à cette prescription. Lorsque l'on utilise cette forme nominative, un sous-arbre de dénomination correspond à un sous-arbre de l'arbre DIT. Les mises en œuvre conformes ne sont pas tenues de reconnaître toutes les formes nominatives possibles. Si l'extension est étiquetée comme critique et qu'une mise en œuvre utilisatrice de certificat ne reconnaisse pas une forme nominative utilisée dans une composante **base** quelconque, le certificat doit être traité comme si une extension critique non reconnue

avait été rencontrée. Si l'extension est étiquetée non critique et qu'une mise en œuvre utilisatrice de certificat ne reconnaisse pas une forme nominative utilisée dans une composante **base** quelconque, cette spécification de sous-arbre peut être ignorée. Lorsqu'une entité titulaire de certificat possède plusieurs noms de la même forme (y compris, dans le cas de la forme nominative de type **directoryName**, le nom contenu dans le champ d'entité titulaire du certificat, si ce champ n'est pas vide) tous les noms de cette forme doivent subir un test quant à leur conformité à une contrainte de nom sur cette forme nominative.

NOTE – Lors du test des noms d'entité titulaire de certificat quant à leur conformité à une contrainte de nom, les noms contenus dans les extensions non critiques de variante nominative d'entité titulaire doivent être traités et non ignorés.

Le champ **minimum** spécifie la limite supérieure de la zone contenue à l'intérieur du sous-arbre. Tous les noms dont la dernière composante se trouve au-dessus du niveau spécifié ne sont pas contenus dans cette zone. Une valeur zéro de la composante **minimum** (valeur par défaut) correspond à la base, c'est-à-dire au sommet nodal du sous-arbre. Par exemple, si le champ **minimum** est mis à 1, le sous-arbre de dénomination exclut la base nodale mais inclut les nœuds subordonnés.

Le champ **maximum** spécifie la limite inférieure de la zone contenue à l'intérieur du sous-arbre. Tous les noms dont la dernière composante se trouve au-dessous du niveau spécifié ne sont pas contenus dans cette zone. Une valeur zéro de la composante **maximum** correspond à la base, c'est-à-dire au sommet du sous-arbre. Une composante **maximum** absente indique qu'aucune limite inférieure ne doit être imposée à la zone contenue dans le sous-arbre. Par exemple, si le champ **maximum** est mis à 1, le sous-arbre de dénomination exclut tous les nœuds sauf la base (sommet) du sous-arbre et ses subordonnés immédiats.

Cette extension peut, au choix de l'émetteur du certificat, être soit critique soit non critique. Il est recommandé qu'elle soit étiquetée critique, sinon un utilisateur de certificat pourrait ne pas vérifier que les certificats subséquents dans un chemin de certification sont inclus dans l'espace nominatif imparti par l'autorité de certification émettrice.

Si cette extension est présente et est étiquetée critique, un système utilisateur de certificat doit vérifier que le chemin de certification en cours de traitement est compatible avec la valeur contenue dans cette extension.

#### 12.4.2.3 Champ de contraintes de politique

Ce champ spécifie les contraintes qui peuvent nécessiter une identification explicite de la politique de certification ou interdire le mappage de politiques pour le reste du chemin de certification. Ce champ est défini comme suit:

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

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

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

Si la composante **requireExplicitPolicy** est présente, elle indique que tous les certificats à partir de celui qui a été émis dans le chemin de certification par une autorité de certification désignée jusqu'à la fin de ce chemin doivent contenir, 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 ou celui d'une politique qui a été déclarée équivalente à la première grâce à la fonction de mappage de politiques. L'autorité de certification désignée est soit l'autorité titulaire du certificat contenant cette extension (si la valeur de la composante **requireExplicitPolicy** est zéro) soit l'autorité de certification qui est titulaire d'un certificat subséquent dans le chemin de certification (tel qu'indiqué par une valeur non nulle).

Si la composante **inhibitPolicyMapping** est présente, elle indique que, dans tous les certificats, à partir de celui qui a été émis dans le chemin de certification par une autorité de certification désignée jusqu'à la fin de ce chemin, le mappage des politiques n'est pas autorisé. L'autorité de certification désignée est soit l'autorité titulaire du certificat contenant cette extension (si la valeur de la composante **inhibitPolicyMapping** est zéro), soit l'autorité de certification qui est titulaire d'un certificat subséquent dans le chemin de certification (tel qu'indiqué par une valeur non nulle).

Une valeur du type **SkipCerts** indique le nombre de certificats à sauter dans le chemin de certification avant qu'une contrainte devienne effective.

Cette extension peut, au choix de l'émetteur, être critique ou non critique. Il est recommandé qu'elle soit étiquetée comme étant critique afin qu'un utilisateur de certificat puisse interpréter correctement la stipulation de l'autorité de certification émettrice.

### 12.4.3 Procédure de traitement du chemin de certification

Le traitement du chemin de certification est effectué dans un système qui a besoin d'utiliser la clé publique d'une entité finale distante, par exemple un système qui vérifie une signature numérique produite par une entité distante. Les extensions de politique de certification, de contrainte de base, de contrainte de nom et de contrainte de politique ont été conçues pour faciliter une mise en œuvre automatique et autonome de la logique de traitement du chemin de certification.

On trouvera ci-après la description d'une procédure de validation de chemins de certification. Une mise en œuvre doit être fonctionnellement équivalente au comportement externe résultant de cette procédure. L'algorithme utilisé par une mise en œuvre donnée pour calculer la ou les valeurs de sortie correctes à partir des entrées indiquées, n'est pas normalisé.

Les données d'entrée de la procédure de traitement du chemin de certification sont les suivantes:

a) ensemble des certificats composant un chemin de certification;

b) valeur ou identificateur fiable de clé publique (si la clé est mise en mémoire interne par le module de traitement du chemin de certification), pour usage lors de la vérification du premier certificat du chemin de certification;

c) ensemble initial de politiques (*initial-policy-set*) composé d'un ou de plusieurs identificateurs de politique de certification, indiquant que l'une quelconque de ces politiques serait acceptable par l'utilisateur du certificat en vue du traitement du chemin de certification; cette entrée peut également prendre la valeur spéciale toute politique (*any-policy*);

d) valeur indicatrice de politique initiale explicite (*initial-explicit-policy*), précisant si un identificateur de politique acceptable doit apparaître explicitement dans le champ d'extension de politiques de certification de tous les certificats contenus dans le chemin de certification;

e) valeur indicatrice d'interdiction de mappage avec la politique initiale (*initial-policy-mapping-inhibit*), qui indique si le mappage de politiques est interdit dans le chemin de certification;

f) la date et l'heure actuelles (si elles ne sont pas fournies par voie interne au module de traitement du chemin de certification).

Les valeurs mentionnées aux points c), d) et e) dépendront des prescriptions imposées par la politique de la combinaison utilisateur-application qui a besoin d'utiliser la clé publique d'entité finale certifiée.

Les données de sortie de la procédure sont les suivantes:

a) indication du succès ou de l'échec de la validation du chemin de certification;

b) en cas d'échec de validation, code de diagnostic indiquant la cause de l'échec;

c) si la validation a réussi, ensemble des politiques soumises aux contraintes des autorités de certification dans le chemin de certification, ainsi que tous les qualificateurs de ces politiques rencontrés dans le chemin de certification, ou la valeur spéciale *any-policy*. Si la valeur *any-policy* n'est pas renvoyée, l'utilisateur du certificat ne doit utiliser le certificat que conformément à l'une des politiques identifiées et doit traiter tous les qualificateurs présents dans le chemin de certification pour cette politique;

d) si la validation a réussi et que la valeur *any-policy* ait été renvoyée selon le point c), l'ensemble de tous les qualificateurs d'élément de politique rencontrés dans le chemin de certification;

e) détails de tout mappage de politiques qui s'est produit lors du traitement du chemin de certification.

NOTE – Si la validation réussit, le système utilisateur du certificat peut encore choisir de ne pas utiliser le certificat à la suite de certaines valeurs des qualificateurs de politique ou d'autres informations contenues dans le certificat.

La procédure fait appel à l'ensemble de variables d'état suivant:

a) *user-constrained-policy-set* (ensemble des politiques contraintes par l'utilisateur): ensemble des identificateurs de politique de certification comprenant les politiques actuellement considérées comme acceptables par l'utilisateur du certificat; cette variable d'état peut également prendre la valeur spéciale *any-policy*;

b) *authority-constrained-policy-set* (ensemble des politiques contraintes par l'autorité): ensemble des identificateurs de politique de certification comprenant les politiques actuellement considérées comme acceptables par les autorités de certification placées dans le chemin de certification; cette variable d'état peut également prendre la valeur spéciale *any-policy*;

c) *permitted-subtrees* (sous-arbres permis): ensemble de spécifications sous-arborescentes définissant des sous-arbres à l'intérieur desquels doivent se trouver tous les noms des entités titulaires des certificats subséquents dans le chemin de certification, ou des sous-arbres pouvant prendre la valeur spéciale non borné (*unbounded*);

d) *excluded-subtrees* (sous-arbres exclus): ensemble (éventuellement vide) des spécifications de sous-arbre (comprenant chacune un nom de base de sous-arbre et des indicateurs de niveau maximum et minimum), définissant les sous-arbres dans lesquels ne peut se trouver aucun nom d'entité titulaire d'un certificat subséquent dans le chemin de certification;

e) *explicit-policy-indicator* (indicateur explicite de politique): cet indicateur précise si une politique acceptable doit être explicitement identifiée dans chaque certificat;

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

g) *pending-constraints* (contraintes imminentes): détails concernant les contraintes de politique explicite ou d'interdiction de mappage de politiques, qui ont été stipulées mais qui n'ont pas encore été appliquées. Il existe deux fanions d'un bit, appelés *explicit-policy-pending* (politique explicite imminente) et *policy-mapping-inhibit-pending* (interdiction imminente de mappage de politiques), assortis l'un et l'autre d'un entier appelé *skip-certificates* (sauter des certificats) qui indique le nombre de certificats qui restent à sauter avant que la contrainte soit appliquée.

La procédure implique une étape d'initialisation, suivie d'une série d'étapes de traitement de chaque certificat. L'étape d'initialisation se compose des opérations suivantes:

a) initialisation de la variable *user-constrained-policy-set* à la valeur de la variable *initial-policy-set*;

b) initialisation de la variable *authority-constrained-policy-set* à la valeur de la variable *any-policy*;

c) initialisation de la variable *permitted-subtrees* à la valeur de la variable *unbounded*;

d) initialisation de la variable *excluded-subtrees* à la valeur d'un ensemble vide;

e) initialisation de la variable *explicit-policy-indicator* à la valeur de la variable *initial-explicit-policy*;

f) initialisation de la variable *policy-mapping-inhibit-indicator* à la valeur de la variable *initial-policy-mapping-inhibit*;

g) initialisation des deux indicateurs *pending-constraints* à la valeur de non-activation.

Chaque certificat est ensuite traité à son tour, à partir du certificat signé, au moyen de la clé publique d'entrée vérifiée. Le dernier certificat est considéré comme étant le *certificat final*; tous les autres certificats sont considérés comme des *certificats intermédiaires*.

Les vérifications suivantes sont appliquées à un certificat:

a) vérifier que la signature est efficace, que les dates sont valides, que les noms de l'entité titulaire du certificat et de l'émetteur du certificat s'enchaînent correctement, et que le certificat n'a pas été annulé;

b) vérifier, pour un certificat intermédiaire où le champ d'extension de contraintes de base est présent, que la composante **cA** est présente et mise à la valeur Vrai. Si la composante **pathLenConstraint** est présente, vérifier que le chemin de certification actuel ne viole pas cette contrainte;

c) si la variable *explicit-policy-indicator* est activée et que la variable *user-constrained-policy-set* ne soit pas mise à *any-policy*, vérifier que l'extension de politiques de certification est présente et qu'au moins un membre de la variable *user-constrained-policy-set* apparaît dans le champ des politiques de certification;

d) si l'extension de politiques de certification est présente et est étiquetée comme étant critique, calculer l'intersection entre les politiques indiquées dans cette extension et l'ensemble *authority-constrained-policy-set*, puis insérer le résultat en tant que nouvelle valeur de la variable *authority-constrained-policy-set*. Vérifier que l'intersection entre la variable *authority-constrained-policy-set* et la variable *user-constrained-policy-set* n'est pas vide;

e) vérifier que le nom de l'entité titulaire figure dans l'espace nominatif indiqué par la valeur de la variable *permitted-subtrees* et qu'il n'apparaît pas dans l'espace nominatif indiqué par la valeur de la variable *excluded-subtrees*.

Si l'une quelconque des vérifications ci-dessus échoue, la procédure se termine et renvoie une indication d'échec et un code de cause approprié. Si aucune des vérifications ci-dessus n'échoue au sujet d'un certificat d'entité finale, la procédure se termine, renvoie une indication de succès ainsi que l'ensemble des identificateurs de politique extraits de l'ensemble *authority-constrained-policy-set*, les qualificateurs requis d'élément de politique, et les détails relatifs à tout mappage de politiques qui pourrait s'être produit.

Pour un certificat intermédiaire, les actions d'enregistrement de contrainte ci-après sont alors exécutées afin de régler correctement les variables d'état pour le traitement du prochain certificat:

a) si l'extension **nameConstraints** est présente dans le certificat avec une composante **permittedSubtrees**, la variable d'état *permitted-subtrees* est mise à l'intersection de sa précédente valeur et de la valeur indiquée dans l'extension du certificat;

b) si l'extension **nameConstraints** est présente dans le certificat avec une composante **excludedSubtrees**, la variable d'état *excluded-subtrees* est mise à la réunion de sa précédente valeur et de la valeur indiquée dans l'extension du certificat;

c) si la variable *explicit-policy-indicator* n'est pas activée:

- et que l'indicateur *explicit-policy-pending* soit activé, décrémenter la valeur correspondante de la variable *skip-certificates* et, si cette valeur atteint zéro, activer l'indicateur *explicit-policy-indicator*;
- et que la contrainte **requireExplicitPolicy** soit présente dans le certificat, exécuter les opérations suivantes: pour une valeur 0 de la composante **SkipCerts**, activer l'indicateur *explicit-policy-indicator*; pour toute autre valeur de la composante **SkipCerts**, activer l'indicateur *explicit-policy-pending* et régler la valeur correspondante de la variable *skip-certificates* à la valeur de **SkipCerts** ou à la valeur précédente de *skip-certificates*, selon celle qui est la plus petite (si l'indicateur *explicit-policy-pending* était déjà activé);

d) si la variable *policy-mapping-inhibit-indicator* n'est pas activée:

- traitement de toute extension de mappage de politiques en fonction des politiques indiquées dans la variable *user-constrainted-policy-set* et adjonction des identificateurs de politique appropriés à la variable *user-constrained-policy-set*;
- traitement de toute extension de mappage de politiques en fonction des politiques indiquées dans la variable *authority-constrained-policy-set* et adjonction des identificateurs de politique appropriés à la variable *authority-constrained-policy-set*;
- si l'indicateur *policy-mapping-inhibit-pending* est activé, régression de la valeur correspondante de la variable *skip-certificates* et, si cette valeur arrive à zéro, activer l'indicateur *policy-mapping-inhibit-indicator*;
- si la contrainte **inhibitPolicyMapping** est présente dans le certificat, exécuter les opérations suivantes: pour une valeur 0 de la composante **SkipCerts**, activer l'indicateur *policy-mapping-inhibit-indicator*; pour toute autre valeur de la composante **SkipCerts**, activer l'indicateur *policy-mapping-inhibit-pending* et régler la valeur correspondante de la variable *skip-certificates* à la valeur de **SkipCerts** ou à la valeur précédente de *skip-certificates*, selon celle qui est la plus petite (si l'indicateur *policy-mapping-inhibit-pending* était déjà activé).

## 12.5 Extensions de base des listes CRL

### 12.5.1 Prescriptions

Les prescriptions suivantes s'appliquent aux listes CRL:

a) les utilisateurs de certificat doivent être en mesure de suivre la trace de toutes les listes CRL émises par un émetteur de liste CRL ou par un point de répartition de listes CRL (voir 12.6) et être en mesure de détecter l'absence d'une liste CRL dans la séquence. Il faut donc attribuer des numéros en séquence aux listes CRL;

b) certains utilisateurs de liste CRL peuvent vouloir répondre différemment à une annulation, selon la raison de celle-ci. Il est donc prescrit qu'une entrée de liste indique la raison de l'annulation;

c) une autorité de certification doit être en mesure de suspendre temporairement la validité d'un certificat puis de l'annuler ou de le remettre en circulation. Les raisons possibles d'une telle action sont les suivantes:

- désir de diminuer la responsabilité en cas d'annulation erronée lorsqu'une demande d'annulation n'est pas authentifiée et qu'il n'y a pas assez de renseignements pour déterminer si elle est valide;
- autres besoins d'entreprise, tels que l'inactivation temporaire d'un certificat d'entité en attendant une vérification des comptes ou une recherche;

d) une liste CRL contient, pour chaque certificat annulé, la date d'expédition de l'annulation par l'autorité de certification. D'autres renseignements peuvent être connus, comme le moment où une compromission réelle ou suspectée de clé s'est produite. Ces renseignements peuvent être utiles à un utilisateur de certificat. La date d'annulation n'est pas suffisante pour résoudre certains litiges car, au pire, toutes les signatures émises au cours de la période de validité du certificat doivent être considérées comme invalides. Il peut cependant être important, pour un utilisateur, qu'un document signé soit reconnu comme valide même si la clé utilisée pour signer le message a fait l'objet d'une compromission après la production de la signature. Pour aider à résoudre ce problème, une entrée de liste CRL peut comporter une seconde date qui indique le moment où l'on a appris ou suspecté que la clé privée avait été compromise.

### 12.5.2 Champs d'extension de liste et d'entrée de liste CRL

Les champs d'extension suivants sont définis:

a) *numéro de liste CRL*;

b) *code de cause*;

c) *code d'instruction de maintien*;

d) *date d'invalidité*.

Le champ de numéro de liste CRL ne doit être utilisé que comme champ d'extension de liste CRL et les autres champs ne doivent être utilisés que comme champs d'extension d'entrée de liste CRL.

#### 12.5.2.1 Champ de numéro de liste CRL

Ce champ d'extension de liste CRL achemine un numéro progressant en séquence monotone pour chaque liste CRL émise par un émetteur de listes CRL donné, par l'intermédiaire d'un attribut d'annuaire d'autorité de certification donné ou d'un point de répartition de liste CRL donné. Ce numéro permet à un utilisateur de liste CRL de détecter si les listes CRL émises avant celle qui est en cours de traitement ont également été vues et traitées. Ce champ est défini comme suit:

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

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

Cette extension n'est jamais critique.

#### 12.5.2.2 Champ de code de cause

Ce champ d'extension d'entrée de liste CRL indique la cause de l'annulation du certificat. Le code de cause peut être utilisé par des applications pour déterminer, sur la base de la politique locale, comment réagir à des annulations reçues. 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) }
```

Les valeurs suivantes du code de cause indiquent pourquoi un certificat a été annulé:

– la valeur **keyCompromise** est utilisée pour annuler un certificat d'entité finale; elle indique que l'on sait ou que l'on soupçonne que la clé privée du titulaire, ou d'autres aspects de l'entité titularisée dans le certificat, ont été compromis;

– la valeur **cACompromise** est utilisée pour annuler un certificat d'autorité de certification; elle indique que l'on sait ou que l'on soupçonne que la clé privée du titulaire, ou d'autres aspects de l'entité titularisée dans le certificat, ont été compromis;

– la valeur **affiliationChanged** indique que le nom de l'entité titulaire, ou d'autres informations contenues dans le certificat, ont été modifiés mais qu'il n'y a pas de raison de penser que la clé privée a été compromise;

– la valeur **superseded** indique que le certificat a été remplacé mais qu'il n'y a pas de raison de penser que la clé privée a été compromise;

– la valeur **cessationOfOperation** indique que le certificat n'est plus nécessaire pour la fin en vue de laquelle il a été émis mais qu'il n'y a pas de raison de penser que la clé privée a été compromise.

On peut mettre un certificat en instance en émettant une entrée de liste CRL affectée d'un code de cause de type **certificateHold**. Le message de mise en instance d'un certificat peut comprendre un code facultatif d'instruction de maintien afin d'acheminer des informations supplémentaires vers les utilisateurs (voir 12.5.2.3). Une fois qu'un code de maintien a été émis, ce code peut être traité de l'une des trois façons suivantes:

a) le code peut rester affecté à la liste CRL, sans autre action que le rejet, par les utilisateurs, des transactions émises au cours de la période de maintien;

b) le code peut être remplacé par une annulation (finale) pour le même certificat, auquel cas la cause sera l'une des causes normales d'annulation, la date d'annulation sera celle à laquelle le certificat a été mis en instance, et le champ d'extension par code facultatif d'instruction de maintien ne doit pas apparaître;

c) le code peut être explicitement libéré et l'entrée supprimée de la liste CRL.

Le code de cause **removeFromCRL** n'est utilisé qu'avec des listes CRL de type delta (voir 12.6). Il indique qu'il y a lieu de supprimer une entrée de liste CRL existante en raison de l'expiration ou de la fin du maintien d'un certificat. Une entrée affectée de ce code de cause doit être utilisée dans les listes CRL delta pour lesquelles la liste CRL de base correspondante contient une entrée affectée du code de cause **certificateHold** pour le même certificat.

Cette extension n'est jamais critique.

#### 12.5.2.3 Champ de code d'instruction de maintien

Ce champ d'extension d'entrée de liste CRL permet d'inclure un identificateur d'instruction enregistré indiquant la mesure à prendre en présence d'un certificat maintenu. Il n'est applicable que dans une entrée ayant un code de cause **certificateHold**. Ce champ est défini comme suit:

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

HoldInstruction ::= OBJECT IDENTIFIER
```

Cette extension n'est jamais critique. Aucun code normalisé d'instruction de maintien n'est défini dans la présente Spécification d'Annuaire.

NOTE – Exemples d'instructions de maintien: "veuillez vous mettre en communication avec l'autorité de certification" ou "récupération du jeton d'usager".

#### 12.5.2.4 Champ de date d'invalidité

Ce champ d'extension d'entrée de liste CRL indique la date à laquelle l'on a appris ou formé le soupçon que la clé privée avait été compromise ou que le certificat devait, pour une autre raison, être considéré comme invalide. Cette date peut être antérieure à la date d'annulation indiquée dans l'entrée de liste CRL, c'est-à-dire à la date à laquelle l'autorité de certification a procédé à l'annulation. Ce champ est défini comme suit:

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

Cette extension n'est jamais critique.

NOTE 1 – La date indiquée dans cette extension n'est pas intrinsèquement suffisante pour justifier un non-rejet. Cette date peut, par exemple, avoir été signalée par le détenteur de la clé privée et il est possible que cette personne revendique à tort qu'une clé a été compromise à un moment donné dans le passé, afin de rejeter une signature légitimement formée.

NOTE 2 – Lorsqu'une annulation est affichée pour la première fois par une autorité de certification dans une liste CRL, la date d'invalidité peut précéder la date d'émission de listes CRL antérieures. La date d'annulation ne doit pas précéder la date d'émission de listes CRL antérieures.

## 12.6 Points de répartition de listes CRL et listes CRL delta

### 12.6.1 Prescriptions

Etant donné qu'il est possible que des listes d'annulations deviennent longues et peu maniables, il faut pouvoir représenter des listes CRL partielles. Différentes solutions sont requises pour deux types différents de mises en œuvre traitant des listes CRL.

Le premier type de mise en œuvre est celui des postes de travail individuels, éventuellement dans un jeton cryptographique attaché. Ces mises en œuvre sont susceptibles d'avoir une capacité limitée, si tant est qu'elle existe, de stockage fiabilisé. Il faut donc examiner la liste CRL tout entière pour déterminer si elle est valide, puis voir si le certificat est valide. Ce traitement peut prendre du temps si la liste CRL est longue. Le partitionnement des listes CRL est requis pour éliminer ce problème avec ces mises en œuvre.

Le deuxième type de mise en œuvre est celui des serveurs très performants, qui traitent un grand volume de messages, comme les serveurs de traitement transactionnel. Dans cet environnement, les listes CRL sont normalement traitées en tâche d'arrière-plan. Après leur validation, le contenu de ces listes est stocké localement sous une représentation qui facilite leur examen, par exemple avec un fanion pour chaque certificat, indiquant si celui-ci a été annulé. Cette représentation est conservée dans une mémoire fiabilisée. Ce type de serveur nécessitera normalement des listes CRL mises à jour pour un grand nombre d'autorités de certification. Comme ce serveur possède déjà une liste des certificats déjà annulés, il lui suffit de reconstituer une liste des certificats récemment annulés. Cette liste, dite liste CRL delta, sera plus courte et exigera moins de ressources qu'une liste CRL complète, en termes de consultation et de traitement.

Les prescriptions suivantes s'appliquent donc aux points de répartition de listes CRL et aux listes CRL delta:

a) de façon à limiter la longueur des listes CRL, il doit être possible d'attribuer, à différentes listes CRL, des sous-ensembles de l'ensemble de tous les certificats émis par une autorité de certification donnée. Cette attribution peut s'effectuer par association de chaque certificat avec un point de répartition de liste CRL qui est:

   – soit une entrée d'Annuaire dont la liste CRL possède un attribut qui contient une entrée d'annulation pour le certificat, si celui-ci a été annulé;

   – soit un emplacement tel qu'une adresse de courrier électronique ou un identificateur Internet de ressource uniforme permettant d'obtenir la liste CRL applicable;

b) pour des raisons de performance, il est souhaitable de réduire le nombre de listes CRL nécessitant une vérification lors de la validation de certificats multiples, par exemple lors de la validation d'un chemin de certification. Pour réaliser cette réduction, un même émetteur de liste CRL peut signer et émettre des listes CRL contenant des annulations issues de multiples autorités de certification;

c) il est prescrit que des listes CRL distinctes s'appliquent aux certificats d'autorité de certification annulés et aux certificats d'entité finale annulés. Cette distinction facilite le traitement des chemins de certification car la liste CRL des certificats d'autorité de certification annulés sera vraisemblablement très courte (et généralement vide). Les attributs **authorityRevocationList** et **certificateRevocationList** ont été spécifiés à cette fin. Mais pour que cette distinction soit sûre, il est nécessaire que chaque liste CRL comporte un indicateur précisant de quelle liste il s'agit. Sinon, on ne pourrait pas détecter d'illégitimes substitutions de listes;

d) il faut prévoir la possibilité d'existence, pour d'éventuelles situations de compromis (lorsqu'il y a un risque notable d'utilisation frauduleuse de clé privée), d'une liste CRL différente de celle qui inclut toutes les terminaisons d'association de routine (lorsqu'il n'y a pas de risque notable d'utilisation frauduleuse de clé privée);

e) il faut également prévoir la possibilité d'avoir des listes CRL partielles (appelées listes CRL delta) qui ne contiennent des entrées que pour les certificats qui ont été annulés depuis l'émission d'une liste CRL de base;

f) alors que les prescriptions de a) à e) peuvent conduire à la mise en place facultative de mécanismes d'annulation, chaque autorité de certification doit, dans le cadre d'un plan commun de convergence, émettre des listes CRL complètes, couvrant tous les certificats émis par cette autorité.

### 12.6.2 Champs d'extension de certificat

L'extension de points de répartition de listes CRL ne doit être utilisée que comme extension de certificat. Elle peut être utilisée aussi bien dans des certificats d'autorité de certification que dans des certificats d'entité finale. Ce champ désigne le ou les points de répartition de listes CRL auquel (auxquels) un utilisateur de certificat doit se rapporter pour vérifier si le certificat a été annulé. Un utilisateur de certificat peut obtenir une liste CRL à partir d'un point de distribution applicable. Il peut également obtenir une liste CRL complète et actuelle à partir de l'entrée d'annuaire de l'autorité de certification. 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) }
```

La composante **distributionPoint** désigne l'emplacement à partir duquel on peut obtenir la liste CRL. Si cette composante est absente, le nom du point de répartition passe par défaut à celui de l'émetteur de la liste CRL.

Lorsque la variante **fullName** est utilisée ou lorsque le nom par défaut s'applique, le nom du point de répartition peut avoir plusieurs formes nominatives. Le même nom, sous au moins une de ses formes nominatives, doit être présent dans le champ **distributionPoint** de l'extension de la liste CRL concernant le point de répartition d'origine. Un système utilisateur de certificat n'est pas tenu d'être en mesure de traiter toutes les formes nominatives. Il peut utiliser un point de répartition à condition qu'au moins une forme nominative puisse être traitée. Si aucune forme nominative ne peut être traitée pour un point de répartition, un système utilisateur de certificat peut quand même utiliser le certificat, à condition que l'information d'annulation nécessaire puisse être obtenue d'une autre source, par exemple un autre point de répartition ou l'entrée d'annuaire de l'autorité de certification.

La composante **nameRelativeToCRLIssuer** ne peut être utilisée que si le point de répartition de listes CRL est affecté d'un nom d'annuaire qui est directement subordonné au nom d'annuaire de l'émetteur de listes CRL. Dans ce cas, la composante **nameRelativeToCRLIssuer** achemine le nom relatif distinctif associé au nom d'annuaire de l'émetteur de la liste CRL.

La composante **reasons** indique les raisons d'annulation gérées par cette liste CRL. Si la composante **reasons** est absente, le point de répartition de listes CRL correspondant répartit une liste CRL qui contient une entrée pour ce certificat si celui-ci a été annulé, quelle qu'en soit la raison. Sinon, la valeur **reasons** indique les raisons d'annulation qui sont gérées par le point de répartition de liste CRL correspondant.

La composante **cRLIssuer** désigne l'autorité qui émet et signe la liste CRL. Si cette composante est absente, le nom de l'émetteur de la liste CRL passe par défaut à celui de l'émetteur du certificat.

Cette extension peut, au choix de l'émetteur du certificat, être critique ou non critique. Aux fins de l'interopérabilité, il est recommandé qu'elle soit étiquetée "non critique". Si l'extension de point de répartition de liste CRL est rendue critique, l'autorité de certification doit veiller à ce que les points de répartition intègrent les codes de cause **keyCompromise** ou **cACompromise**, selon ce qui est applicable.

Si cette extension est étiquetée "critique", un système utilisateur de certificat ne doit pas utiliser le certificat sans avoir au préalable consulté et vérifié une liste CRL à partir d'un des points de répartition désignés, intégrant au moins le code de cause **keyCompromise** (pour un certificat d'entité finale) ou le code de cause **cACompromise** (pour un certificat d'autorité de certification).

Si cette extension est étiquetée "non critique" et qu'un système utilisateur de certificat ne reconnaisse pas le type de champ d'extension, ce système ne doit utiliser le certificat que dans les conditions suivantes:

- s'il peut recevoir de l'autorité de certification une liste CRL complète et la vérifier (le fait que cette liste CRL est complète est indiqué par l'absence, dans cette liste, de champ d'extension concernant le point de répartition d'origine);
- si la vérification d'annulation n'est pas prescrite par la politique locale;
- ou si la vérification d'annulation est réalisée par d'autres moyens.

NOTE – Il est possible d'avoir, pour le même certificat, plusieurs listes CRL issues de plusieurs émetteurs. La coordination entre ces émetteurs de listes CRL et l'autorité de certification d'origine relève de la politique de l'autorité de certification.

### 12.6.3 Champs d'extension de liste CRL et d'entrée de liste CRL

Les champs d'extension de liste CRL ou d'entrée de liste CRL suivants sont définis:

a) *point de répartition d'origine*;

b) *émetteur de certificat*;

c) *indicateur de liste CRL delta*.

Les champs point de répartition d'origine et indicateur de liste CRL delta ne doivent être utilisés que pour des extensions de liste CRL. Le champ émetteur de certificat ne doit être utilisé que pour une extension d'entrée de liste CRL.

#### 12.6.3.1 Champ de point de répartition d'origine

Ce champ d'extension de liste CRL désigne le point de répartition correspondant à cette liste CRL particulière. Il indique si celle-ci est limitée aux seules annulations de certificats d'entité finale, aux seules annulations de certificats d'autorité de certification, ou à un seul ensemble limité de causes. La liste CRL est signée par la clé de l'émetteur de cette liste car les points de répartition de listes CRL ne possèdent pas leurs propres paires de clés. Une liste CRL répartie par l'intermédiaire de l'Annuaire est stockée dans l'entrée du point de répartition de liste CRL, qui ne peut pas être l'entrée d'annuaire de l'émetteur de la liste CRL. 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,
    onlyContainsCACerts        [2] BOOLEAN DEFAULT FALSE,
    onlySomeReasons            [3] ReasonFlags OPTIONAL,
    indirectCRL                [4] BOOLEAN DEFAULT FALSE }
```

La composante **distributionPoint** contient le nom du point de répartition sous une ou plusieurs formes nominatives. Si ce champ est absent, la liste CRL doit contenir des entrées pour tous les certificats annulés mais non arrivés à expiration, qui sont issus de l'émetteur de la liste CRL.

Si la composante **onlyContainsUserCerts** est Vrai, la liste CRL ne contient d'annulations que pour des certificats d'entité finale. Si la composante **onlyContainsCACerts** est Vrai, la liste CRL ne contient d'annulations que pour des certificats d'autorité de certification. Si la composante **onlySomeReasons** est présente, la liste CRL ne contient d'annulations que pour la ou les causes identifiées; sinon, la liste d'annulations contient des annulations pour toutes les causes.

Si la composante **indirectCRL** est Vrai, cette liste CRL peut contenir des notifications d'annulation issues d'autorités de certification autres que l'émetteur de la liste CRL. L'autorité de certification particulière qui est responsable de chaque entrée est conforme aux indications données dans l'extension d'entrée CRL issue de l'émetteur de la liste CRL ou conforme aux règles de configuration par défaut décrites au 12.6.3.2. Dans une telle liste CRL, il appartient à l'émetteur de la liste de s'assurer que celle-ci est complète en ce sens qu'elle contient toutes les entrées d'annulation, compatibles avec les indicateurs **onlyContainsUserCerts**, **onlyContainsCACerts** et **onlySomeReasons**, issues de toutes les autorités de certification qui désignent cet émetteur de liste CRL dans leurs certificats.

Pour les listes CRL réparties par l'intermédiaire de l'Annuaire, les règles suivantes s'appliquent concernant l'utilisation des attributs. Une liste CRL dont la composante **onlyContainsCACerts** est Vrai doit être répartie via l'attribut **authorityRevocationList** du point de répartition associé ou, si aucun point de répartition n'est identifié, via l'attribut **authorityRevocationList** de l'entrée d'émetteur de liste CRL. Sinon, celle-ci doit être répartie via l'attribut **certificateRevocationList** du point de répartition associé ou, si aucun point de répartition n'est identifié, via l'attribut **certificateRevocationList** de l'entrée d'autorité de certification.

Cette extension est toujours critique. Un utilisateur de certificat qui ne comprend pas cette extension ne peut pas supposer que la liste CRL contient une énumération complète des certificats annulés de l'autorité de certification identifiée. Les listes CRL ne contenant pas d'extensions critiques doivent contenir toutes les entrées de liste CRL actuelles pour l'autorité de certification émettrice, y compris les entrées pour tous les certificats d'utilisateur et d'autorité de certification annulés.

NOTE 1 – Le moyen permettant aux autorités de certification de communiquer les informations d'annulation aux émetteurs de listes CRL est hors du domaine d'application de la présente Recommandation | Norme internationale.

NOTE 2 – Si une autorité de certification émet, à partir de sa propre entrée d'annuaire (c'est-à-dire à partir d'un point autre que celui de répartition CRL distinctement nommé), une liste CRL dont la composante **onlyContainsUserCerts** ou **onlyContainsCACerts** est Vraie, cette autorité de certification doit veiller à ce que tous les certificats visés par cette liste CRL contiennent l'extension **basicConstraints**.

#### 12.6.3.2 Champ d'émetteur de certificat

Ce champ d'extension d'entrée de liste CRL désigne l'émetteur de certificat associé à une entrée dans une liste CRL indirecte, c'est-à-dire une liste CRL dont l'indicateur **indirectCRL** est activé dans son extension de point de répartition d'origine. Si cette extension n'est pas présente dans la première entrée d'une liste CRL indirecte, le nom de l'émetteur de certificat passe par défaut à celui de l'émetteur de la liste CRL. Dans les entrées suivantes d'une liste CRL indirecte, si cette extension n'est pas présente, l'émetteur de certificat pour cette entrée est le même que celui de l'entrée précédente. Ce champ est défini comme suit:

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

Cette extension est toujours critique. Si une mise en œuvre ne tient pas compte de cette extension, elle ne peut pas attribuer correctement des entrées de liste CRL aux certificats.

#### 12.6.3.3 Champ d'indicateur de liste CRL delta

Ce champ d'extension de liste CRL indique qu'une liste CRL est de type delta seulement. Ce champ est défini comme suit:

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

BaseCRLNumber ::= CRLNumber
```

La valeur du type **BaseCRLNumber** désigne le numéro de la liste CRL de base qui a été utilisée comme point de départ pour produire cette liste CRL delta. En d'autres termes, cette liste CRL delta contient les modifications intervenues entre la liste CRL de base et la liste CRL complète qui a été émise en même temps que cette liste CRL delta. Il appartient à une autorité de certification de fournir les listes CRL delta. Une liste CRL delta ne doit cependant pas être émise sans qu'une liste CRL complète et correspondante soit émise en même temps. La valeur du numéro de liste CRL, telle qu'acheminée dans le champ d'extension de numéro de liste CRL (s'il est présent), doit être identique aussi bien pour la liste CRL delta que pour la liste CRL complète correspondante.

L'utilisateur de liste CRL qui construit une liste maintenue localement à partir de listes CRL delta doit considérer sa liste CRL construite comme incomplète et inutilisable si les deux conditions suivantes sont Vrai:

- la valeur du numéro **CRLNumber** de la liste CRL delta reçue est supérieure d'au moins deux unités à la valeur du numéro **CRLNumber** de la dernière liste CRL delta traitée;
- et la valeur du numéro **BaseCRLNumber** de la liste CRL delta reçue a changé par rapport à la valeur du numéro **BaseCRLNumber** de la dernière liste CRL delta traitée.

Cette extension est toujours critique. Un utilisateur de certificat qui ne comprend pas l'utilisation des listes CRL delta ne doit pas utiliser de liste CRL contenant cette extension car la liste CRL n'est peut-être pas aussi complète que l'utilisateur le croit. Les listes CRL ne contenant pas d'extensions critiques doivent toujours contenir toutes les entrées de liste CRL actuelles pour l'autorité de certification émettrice, y compris les entrées pour tous les certificats d'utilisateur annulés et pour tous les certificats d'autorité de certification annulés.

NOTE – Il appartient à l'autorité de certification de déterminer s'il y a lieu de répartir les listes CRL émises entre deux listes CRL de base. Par exemple, l'autorité de certification peut avoir pour politique de distribuer des CRL de base sur des CD-ROM et des CD-ROM delta par un service en ligne comme l'annuaire. Même si l'autorité de certification a émis sa liste CRL avec la liste CRL delta associée, sa politique peut imposer à l'utilisateur de construire la liste CRL actuelle en appliquant la liste CRL delta à la liste CRL de base distribuée sur le CD-ROM.

### 12.6.4 Type d'attribut pour listes CRL delta

Le type d'attribut suivant est défini pour maintenir une liste CRL delta dans une entrée d'annuaire d'autorité de certification:

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

## 12.7 Règles de correspondance

La présente Spécification d'Annuaire définit des règles de correspondance à utiliser avec les attributs de type **Certificate**, **CertificatePair**, **CertificateList**, et **SupportedAlgorithm**, respectivement.

### 12.7.1 Correspondance exacte de certificat

Cette règle compare, quant à leur égalité, une valeur présentée et une valeur d'attribut de type **Certificate.** Elle sélectionne de façon univoque un certificat et un seul.

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

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

Cette règle de correspondance renvoie la valeur Vrai si les composantes contenues dans la valeur d'attribut correspondent à celles qui se trouvent dans la valeur présentée.

### 12.7.2 Correspondance de certificat

Cette règle compare une valeur présentée et une valeur d'attribut de type **Certificate**. Elle sélectionne un ou plusieurs certificats sur la base de divers critères.

```
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 }

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 (1..MAX) OF CertPolicyId
```

Cette règle de correspondance renvoie la valeur Vrai si toutes les composantes qui sont présentes dans la valeur présentée correspondent aux composantes homologues de la valeur d'attribut, comme suit:

a) la composante **serialNumber** correspond si sa valeur dans celle de l'attribut est égale à celle qui se trouve dans la valeur présentée;

b) la composante **issuer** correspond si sa valeur dans celle de l'attribut est égale à celle qui se trouve dans la valeur présentée;

c) la composante **subjectKeyIdentifier** correspond si sa valeur mémorisée dans l'attribut est égale à celle qui se trouve dans la valeur présentée; il n'y a pas de correspondance si la valeur mémorisée dans l'attribut ne contient aucune extension d'identificateur de clé d'entité titulaire;

d) la composante **authorityKeyIdentifier** correspond si sa valeur mémorisée dans l'attribut est égale à celle qui se trouve dans la valeur présentée; il n'y a pas de correspondance si la valeur mémorisée dans l'attribut ne contient aucune extension d'identificateur de clé d'autorité ou si toutes les composantes de la valeur présentée ne se trouvent pas dans la valeur mémorisée dans l'attribut;

e) la composante **certificateValid** correspond si la valeur présentée s'inscrit dans la période de validité de la valeur mémorisée dans l'attribut;

f) la composante **privateKeyValid** correspond si la valeur présentée s'inscrit dans la période indiquée par l'extension de période d'utilisation de clé privée contenue dans la valeur mémorisée dans l'attribut ou si celle-ci ne contient aucune extension de période d'utilisation de clé privée;

g) la composante **subjectPublicKeyAlgID** correspond si elle est égale à la composante **algorithm** de l'identificateur **algorithmIdentifier** de la composante **subjectPublicKeyInformation** contenue dans la valeur mémorisée dans l'attribut;

h) la composante **keyUsage** correspond si tous les éléments binaires activés dans la valeur présentée le sont également dans l'extension d'utilisation de clé privée contenue dans la valeur mémorisée dans l'attribut, ou si celle-ci ne contient aucune extension d'utilisation de clé privée;

i) la composante **subjectAltName** correspond si la valeur mémorisée dans l'attribut contient l'extension de variante nominative d'entité titulaire avec une composante **AltNames** du même type nominatif qu'indiqué dans la valeur présentée;

j) la composante **policy** correspond si tous les éléments de politique désignés dans une des valeurs présentées se trouvent dans l'ensemble d'identificateurs d'éléments de politique **policyElementsIds** contenu dans une des valeurs de composante **policyInformation** de l'extension relative aux politiques de certification contenue dans la valeur mémorisée dans l'attribut; il n'y a pas de correspondance si celle-ci ne contient aucune extension de politique de certification;

k) la composante **pathToName** correspond si le certificat ne possède pas d'extension de contraintes nominatives interdisant la construction d'un chemin de certification jusqu'à la valeur nominative présentée.

### 12.7.3 Correspondance exacte d'une paire de certificats

La règle de correspondance exacte d'une paire de certificats compare, quant à leur égalité, une valeur présentée et une valeur d'attribut de type **CertificatePair**. Elle sélectionne uniquement une paire de certificats homologues.

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

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

Cette règle de correspondance renvoie la valeur Vrai si les composantes présentes dans les composantes **forwardAssertion** et **reverseAssertion** de la valeur présentée correspondent aux composantes homologues des composantes **forward** et **reverse** respectivement, dans la valeur mémorisée dans l'attribut.

### 12.7.4 Correspondance d'une paire de certificats

La règle de correspondance d'une paire de certificats compare une valeur présentée avec une valeur d'attribut de type **CertificatePair**. Elle sélectionne une ou plusieurs paires de certificats homologues sur la base de divers critères applicables au certificat progressif ou au certificat régressif de la paire.

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

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

Cette règle de correspondance renvoie la valeur Vrai si toutes les composantes présentes dans les composantes **forwardAssertion** et **reverseAssertion** de la valeur présentée correspondent aux composantes homologues des composantes **forward** et **reverse**, respectivement, contenues dans la valeur d'attribut mémorisée, conformément aux règles indiquées en 12.7.2.

### 12.7.5 Correspondance exacte de listes de certificats

La règle de correspondance exacte de listes de certificats compare, quant à leur égalité, une valeur présente et une valeur d'attribut de type **CertificateList**. Elle sélectionne uniquement une liste CRL donnée.

```
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 composantes contenues dans la valeur mémorisée dans l'attribut correspondent à celles qui sont contenues dans la valeur présentée. Si la composante **distributionPoint** est présente, elle doit toujours correspondre sous au moins une forme nominative.

### 12.7.6 Correspondance de listes de certificats

La règle de correspondance de listes de certificats compare une valeur présentée à une valeur d'attribut de type **CertificateList**. Elle sélectionne une ou plusieurs listes CRL sur la base de divers critères.

```
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 }
```

La règle de correspondance renvoie la valeur Vrai si toutes les composantes qui sont présentes dans la valeur présentée correspondent aux composantes homologues de la valeur mémorisée dans l'attribut, comme suit:

a) la composante **issuer** correspond si sa valeur, contenue dans celle de l'attribut, est égale à celle de la valeur présentée;

b) la composante **minCRLNumber** correspond si sa valeur est inférieure ou égale à celle qui est contenue dans l'extension de numéro de liste CRL de la valeur mémorisée dans l'attribut; il n'y a pas de correspondance si la valeur mémorisée dans l'attribut ne contient aucune extension de numéro de liste CRL;

c) la composante **maxCRLNumber** correspond si sa valeur est supérieure ou égale à celle qui est contenue dans l'extension de numéro de liste CRL de la valeur mémorisée dans l'attribut; il n'y a pas de correspondance si la valeur mémorisée dans l'attribut ne contient aucune extension de numéro de liste CRL;

d) la composante **reasonFlags** correspond si l'un quelconque des éléments binaires qui sont activés dans la valeur présentée est également activé dans les composantes **onlySomeReasons** de l'extension relative au point de répartition d'origine, contenue dans la valeur mémorisée dans l'attribut; il n'y a pas de correspondance si la valeur mémorisée dans l'attribut ne contient aucune extension de point de répartition d'origine;

e) la composante **dateAndTime** correspond si sa valeur est égale ou ultérieure à la valeur contenue dans la composante **thisUpdate** de la valeur mémorisée dans l'attribut et si elle est antérieure à la valeur contenue dans la composante **nextUpdate** de la valeur mémorisée dans l'attribut; il n'y a pas de correspondance si la valeur mémorisée dans l'attribut ne contient aucune composante **nextUpdate**;

f) la composante **distributionPoint** correspond si la valeur mémorisée dans l'attribut contient une extension de point de répartition d'origine et que la valeur de cette composante dans la valeur présente soit égale à la valeur correspondante dans cette extension, sous au moins une forme nominative.

### 12.7.7 Correspondance d'identificateurs d'algorithme

La règle de correspondance d'identificateurs d'algorithme compare, quant à leur égalité, une valeur présentée et une valeur d'attribut de type **SupportedAlgorithm**.

```
algorithmIdentifierMatch MATCHING-RULE      ::=  {
    SYNTAX              AlgorithmIdentifier
    ID                  id-mr-algorithmIdentifierMatch }
```

La règle renvoie la valeur Vrai si la valeur présentée est égale à la composante **algorithmIdentifier** de la valeur mémorisée dans l'attribut.

## 13 Obtention d'attributs certifiés

Les identités des utilisateurs sont associées à leurs certificats de clé publique aux fins de l'authentification et de l'identification. Chaque certificat de clé publique achemine les informations nécessaires pour exécuter certaines fonctions cryptographiques. Les attributs certifiés qui sont associés à une entité titulaire, comme les habilitations, peuvent être acheminés dans une structure distincte, définie sous la forme d'un certificat d'attribut **AttributeCertificate**.

### 13.1 Certificats d'attribut

Un certificat d'attribut est une structure distincte d'un certificat de clé publique d'entité titulaire selon la présente Recommandation | Norme internationale. Une entité titulaire peut avoir plusieurs certificats d'attribut, associés à chacun de ses certificats de clé publique. Il n'est pas prescrit que la même autorité de certification crée à la fois le certificat de clé publique et le ou les certificats d'attribut pour un utilisateur. En fait, la séparation des tâches obligera souvent à procéder autrement. Il convient cependant de tenir compte, lors de la définition d'une politique locale, des implications de l'exécution de multiples validations de chemin de certification. Les échanges entre une entité titulaire et l'autorité de certification traitant de la production de certificats d'attribut sont hors du domaine d'application de la présente Recommandation | Norme internationale.

```
AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE {
    version     Version DEFAULT v1,
    subject     CHOICE {
        baseCertificateID   [0]   IssuerSerial,  -- numéro associé à un certificat de clé publique
        subjectName         [1]   GeneralNames },  -- associé à un nom
    issuer                  GeneralNames,  -- CA émettant le certificat d'attribut
    signature               AlgorithmIdentifier,
    serialNumber            CertificateSerialNumber,
    attrCertValidityPeriod  AttCertValidityPeriod,
    attributes              SEQUENCE OF Attribute,
    issuerUniqueID          UniqueIdentifier OPTIONAL,
    extensions              Extensions OPTIONAL }

IssuerSerial  ::= SEQUENCE {
    issuer          GeneralNames,
    serial          CertificateSerialNumber,
    issuerUID       UniqueIdentifier OPTIONAL}

AttCertValidityPeriod ::= SEQUENCE {
    notBeforeTime       GeneralizedTime,
    notAfterTime        Generalized Time }
```

Le numéro de version **version** permet de distinguer différentes versions du certificat d'attribut. Pour la présente Recommandation | Norme internationale, le numéro de version doit être **v1**.

Le champ **subject** contient l'identité de l'entité titulaire du certificat, désignée soit par le numéro de série d'émetteur **serialNumber** qui est associé au certificat de clé publique de l'entité titulaire, soit par le nom de cette entité **subjectName**.

Le champ **issuer** contient le nom de l'autorité de certification qui a créé le certificat d'attribut.

Le champ **serialNumber** contient le numéro de série qui identifie de manière unique le certificat d'attribut.

Le champ **signature** identifie l'algorithme cryptographique utilisé pour signer numériquement le certificat d'attribut.

Le champ **attCertValidityPeriod** achemine la période, exprimée en temps généralisé, pendant laquelle le certificat d'attribut est considéré comme valide.

Le champ **attributes** contient tout attribut d'annuaire concernant l'entité titulaire (sans limitation aux seuls attributs stockés dans un système d'Annuaire, mais avec reconnaissance obligatoire de l'attribut dans le cadre du domaine considéré).

Le champ **issuerUniqueId** identifie de manière unique l'émetteur du certificat d'attribut lorsque le nom de cet émetteur n'est pas suffisant.

Le champ **extensions** permet d'ajouter de nouveaux champs au certificat d'attribut.

## 13.2 Attribut de certificat d'attribut

Un certificat d'attribut peut être utilisé pour certifier certaines valeurs d'attribut et les associer soit à un certificat X.509 soit directement à un nom d'entité titulaire.

```
attributeCertificateAttribute ATTRIBUTE  ::= {
    WITH SYNTAX                    AttributeCertificate
    EQUALITY MATCHING-RULE         certificateExactMatch
    ID                             id-at-attributeCertificate }
```

## 13.3 Règle de correspondance des certificats d'attribut

La règle de correspondance des certificats d'attribut compare une valeur présentée à une valeur d'attribut du type **AttributeCertificate**. Cette règle de correspondance permet une correspondance plus complexe que la règle **certificateExactMatch**.

```
attributeCertificateMatch  MATCHING-RULE  ::= {
    SYNTAX AttributeCertificateAssertion
    ID     id-mr-attributeCertificateMatch  }

AttributeCertificateAssertion  ::=  SEQUENCE  {
    subject
    [0]  CHOICE {
                    baseCertificateID   [0]  IssuerSerial,
                    subjectName         [1]  GeneralNames} OPTIONAL,
    issuer          [1]  GeneralNames OPTIONAL,
    attCertValidity[2]   GeneralizedTime OPTIONAL,
    attType         [3]  SET OF AttributeType OPTIONAL}
     -- Au moins une composante de la séquence doit être présente
```

La règle de correspondance renvoie la valeur Vrai si toutes les composantes se trouvant dans la valeur présentée correspondent aux composantes homologues de la valeur d'attribut, comme suit:

a) la composante **baseCertificateID** correspond si elle est égale à la composante **IssuerSerial** de la valeur mémorisée dans l'attribut;

b) la composante **subjectName** correspond si la valeur mémorisée dans l'attribut contient une extension nominative contenant le même type de nom que dans la valeur présentée;

c) la composante **issuer** correspond si la valeur mémorisée dans l'attribut contient une composante nominative ayant le même type de nom qu'indiqué dans la valeur présentée;

d) la composante **attCertValidity** correspond si sa valeur s'inscrit dans la période de validité spécifiée par la valeur mémorisée dans l'attribut;

e) pour chaque composante **attType** contenue dans la valeur présentée, il existe un attribut de ce type qui est présent dans la composante **attributes** de la valeur mémorisée.

## 13.4 Chemins de certificat d'attribut

Un utilisateur peut obtenir un ou plusieurs certificats d'attribut auprès d'une ou de plusieurs autorités de certification. Chaque certificat d'attribut porte le nom de l'autorité de certification qui l'a émis. Le type de données ASN.1 suivant peut être utilisé pour représenter un chemin de certificat d'attribut.

```
AttributeCertificationPath  ::=  SEQUENCE {
    attributeCertificate  AttributeCertificate,
    acPath        SEQUENCE OF ACPathData OPTIONAL }
```

```
ACPathData ::= SEQUENCE {
    certificate            [0] Certificate OPTIONAL,
    attributeCertificate   [1] AttributeCertificate OPTIONAL }
```

## 13.5 Liste d'annulations des certificats d'attribut

Les certificats d'attribut peuvent être annulés avant leur date d'expiration si les attributs ne sont plus applicables. Les certificats d'attribut peuvent être annulés sans incidence sur le certificat de clé publique de l'utilisateur. Les listes d'annulations sont conservées dans des entrées sous forme d'attribut de type **attributeCertificateRevocationList**. Ce type d'attribut est défini comme suit:

```
attributeCertificateRevocationList    ATTRIBUTE ::= {
    WITH SYNTAX    CertificateList
    ID             id-at-attributeCertificateRevocationList}
```

NOTE – Tous les amendements appliqués aux listes d'annulations de certificats selon la version 2 sont applicables aux listes d'annulations de certificats d'attribut.

# Annexe A

## Cadre d'authentification en ASN.1

(Cette annexe fait partie intégrante de la présente Recommandation | Norme internationale)

La présente annexe contient toutes les définitions de type, de valeur et de classe d'objets informationnels ASN.1 qui figurent dans la présente Spécification d'Annuaire, sous la forme des deux modules ASN.1 **AuthenticationFramework** et **CertificateExtensions**.

---

```
AuthenticationFramework {joint-iso-ccitt ds(5) module(1) authenticationFramework(7) 3}
DEFINITIONS ::=
BEGIN

-- EXPORTE 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'obligeront pas
-- à apporter les extensions et les modifications nécessaires pour tenir à jour ou améliorer le service d'Annuaire.

IMPORTS
    id-at, informationFramework, upperBounds, selectedAttributeTypes, basicAccessControl,
    certificateExtensions
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}

    Name, ATTRIBUTE
        FROM InformationFramework informationFramework

    ub-user-password
        FROM UpperBounds upperBounds

    AuthenticationLevel
        FROM BasicAccessControl basicAccessControl

    UniqueIdentifier, octetStringMatch
        FROM SelectedAttributeTypes selectedAttributeTypes

        certificateExactMatch, certificatePairExactMatch, certificateListExactMatch
        FROM CertificateExtensions certificateExtensions ;
-- définition du certificat de base

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 présent, la version doit être v2 ou v3
    subjectUniqueIdentifier      [2]  IMPLICIT UniqueIdentifier OPTIONAL
                                      -- si présent, la version doit être v2 ou v3
    extensions                   [3]  Extensions OPTIONAL
                                      -- si 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 }
-- L'ensemble d'objets informationnels ci-après sera défini ultérieurement, peut-être selon des protocoles
-- normalisés ou des déclarations de conformité d'instance de protocole. Cet ensemble est nécessaire
-- pour spécifier une contrainte du tableau applicable à l'élément parameters de l'identificateur 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
-- Pour les extensions dont l'ordre individuel dans la structure SEQUENCE est significatif, la spécification de
-- ces extensions individuelles doit comprendre les règles régissant la signification de cet ordre.

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

    ExtensionSet    EXTENSION   ::=   { ... }

EXTENSION ::= CLASS {
    &id         OBJECT IDENTIFIER UNIQUE,
    &ExtnType }
WITH SYNTAX {
    SYNTAX              &ExtnType
    IDENTIFIED BY       &id }
-- autres structures de certificats

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
                    -- au moins un des éléments de la paire doit être présent -- }
-- Liste d'annulations de certificats (CRL)

CertificateList         ::=   SIGNED { SEQUENCE {
    version                 Version OPTIONAL,
                -- si cet élément est présent, la version doit être v2
    signature                   AlgorithmIdentifier,
    issuer                      Name,
    thisUpdate                  Time,
    nextUpdate                  Time OPTIONAL,
    revokedCertificates         SEQUENCE OF SEQUENCE {
            userCertificate         CertificateSerialNumber,
            revocationDate          Time,
            crlEntryExtensions      Extensions OPTIONAL } OPTIONAL,
    crlExtensions       [0]  Extensions OPTIONAL }}
-- certificat d'attribut

AttributeCertificationPath  ::=  SEQUENCE {
    attributeCertificate  AttributeCertificate,
    acPath          SEQUENCE OF ACPathData OPTIONAL }
```

```
ACPathData ::= SEQUENCE {
    certificate            [0] Certificate OPTIONAL,
    attributeCertificate   [1] AttributeCertificate OPTIONAL }

attributeCertificate ATTRIBUTE ::= {
    WITH SYNTAX                    AttributeCertificate
    EQUALITY MATCHING-RULE         attributeCertificateMatch
    ID                             id-at-attributeCertificate }

AttributeCertificate ::= SIGNED {AttributeCertificateInfo}

AttributeCertificateInfo ::= SEQUENCE {
    version                Version DEFAULT v1,
    subject CHOICE {
        baseCertificateID  [0] IssuerSerial, -- numéro associé à un certificat de clé publique
        subjectName        [1] GeneralNames }, -- associé à un nom
    issuer                 GeneralNames, -- CA émettrice du certificat d'attribut
    signature              AlgorithmIdentifier,
    serialNumber           CertificateSerialNumber,
    attCertValidityPeriod  AttCertValidityPeriod,
    attributes             SEQUENCE OF Attribute,
    issuerUniqueID         UniqueIdentifier OPTIONAL,
    extensions             Extensions OPTIONAL }

IssuerSerial ::= SEQUENCE {
    issuer          GeneralNames,
    serial          CertificateSerialNumber,
    issuerUID       UniqueIdentifier OPTIONAL}

AttCertValidityPeriod ::= SEQUENCE{
    notBeforeTime    GeneralizedTime,
    notAfterTime     GeneralizedTime }

attributeCertificateMatch MATCHING-RULE ::= {
    SYNTAX      AttributeCertificateAssertion
    ID          id-mr-attributeCertificateMatch }

AttributeCertificateAssertion ::= SEQUENCE {
    subject
    [0]  CHOICE {
                    baseCertificateID  [0] IssuerSerial,
                    subjectName        [1] Name} OPTIONAL,
    issuer          [1] Name OPTIONAL,
    attCertValidity
                    [2] GeneralizedTime OPTIONAL,
    attType         [3] SET OF AttributeType OPTIONAL}
-- au moins une composante de la séquence doit être présente
-- types d'attribut --

userPassword                   ATTRIBUTE ::=          {
    WITH SYNTAX                    OCTET STRING (SIZE (0..ub-user-password))
    EQUALITY MATCHING RULE         octetStringMatch
    ID                             id-at-userPassword }

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 }

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

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

attributeCertificateRevocationList      ATTRIBUTE ::= {
      WITH SYNTAX      CertificateList
      ID               id-at-attributeCertificateRevocationList}
-- classes d'objets informationnels --

ALGORITHM ::=    TYPE-IDENTIFIER
-- types paramétrés --

ENCRYPTED-HASH { ToBeSigned }     ::=    BIT STRING ( CONSTRAINED BY {
      -- doit être le résultat de l'application d'une procédure de hachage aux octets codés
      -- selon les règles de codage de base --
      -- d'une valeur de l'élément -- ToBeSigned -- puis de l'application d'une procédure
      -- de chiffrements à ces octets -- })

ENCRYPTED { ToBeEnciphered }        ::=    BIT STRING ( CONSTRAINED BY {
      -- doit être le résultat de l'application d'une procédure de chiffrement --
      -- aux octets codés selon les règles de codage de base d'une valeur de l'élément -- ToBeEnciphered})

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

SIGNED { ToBeSigned }               ::=    SEQUENCE {
      toBeSigned                    ToBeSigned,
      COMPONENTS OF                 SIGNATURE { ToBeSigned }}
-- assignations d'identificateur d'objet --

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-attributeCertificate          OBJECT IDENTIFIER       ::=    {id-at 58}

END

-- module d'extensions de certificat

CertificateExtensions {joint-iso-ccitt ds(5) module(1) certificateExtensions(26) 0}
DEFINITIONS IMPLICIT TAGS ::=
BEGIN

-- EXPORTE TOUT --

IMPORTS
      id-at, id-ce, id-mr, informationFramework, authenticationFramework,
            selectedAttributeTypes, upperBounds
            FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1)
            usefulDefinitions(0) 3}
      Name, RelativeDistinguishedName, ATTRIBUTE, Attribute,
            MATCHING-RULE FROM InformationFramework informationFramework
      CertificateSerialNumber, CertificateList, AlgorithmIdentifier,
            EXTENSION
            FROM AuthenticationFramework authenticationFramework
      DirectoryString
            FROM SelectedAttributeTypes selectedAttributeTypes
      ub-name
            FROM UpperBounds upperBounds
      ORAddress
            FROM MTSAbstractService {joint-iso-ccitt mhs(6) mts(3)
            modules(0) mts-abstract-service(1) version-1994 (0) } ;
-- Sauf indication explicite par ailleurs, l'ordre des composantes de la structure
-- SEQUENCE OF n'a aucune signification particulière dans la présente Spécification.
-- Extensions relatives aux informations de clé et de politique --
```

```
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 ::= { ... }
```

```
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 }

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 }
-- Extensions relatives aux attributs de titulaire et d'émetteur de certificat --

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
-- Extensions relatives aux contraintes du chemin de certification --

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)

CertPolicySet ::= SEQUENCE (1..MAX) OF CertPolicyId
-- Extensions relatives aux listes CRL de base --

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) }

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

HoldInstruction ::= OBJECT IDENTIFIER

invalidityDate EXTENSION ::= {
    SYNTAX      GeneralizedTime
    IDENTIFIED BY id-ce-invalidityDate }
-- Extensions relatives aux points de répartition de listes CRL et aux listes CRL delta --

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) }

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

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

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

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

BaseCRLNumber ::= CRLNumber

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

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 }

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

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

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

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

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

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

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

algorithmIdentifierMatch MATCHING-RULE ::= {
    SYNTAX          AlgorithmIdentifier
    ID              id-mr-algorithmIdentifierMatch }
-- Assignations d'identificateurs d'objet --

id-at-supportedAlgorithms           OBJECT IDENTIFIER ::= {id-at 52}
id-at-deltaRevocationList           OBJECT IDENTIFIER ::= {id-at 53}
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-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}
-- Les identificateurs d'objet suivants ne sont pas utilisés par 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
```

# Annexe B

## Prescriptions de sécurité[2)]

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

De nombreuses applications OSI, des services définis par l'UIT-T et des services non définis par l'UIT-T comportent des prescriptions de sécurité. Ces prescriptions proviennent du besoin de protéger le transfert de l'information contre une gamme de dangers potentiels.

### B.1 Dangers

Certains dangers sont bien connus:

a) *interception d'identité* – L'identité d'un ou de plusieurs des utilisateurs intervenant dans une communication est notée en vue d'un usage frauduleux;

b) *usurpation d'identité* – Prétention d'un utilisateur à se faire passer pour un autre utilisateur afin d'avoir accès à l'information ou d'acquérir des privilèges supplémentaires;

c) *redéfilement* – Enregistrement et écoute ultérieure d'une communication;

d) *interception de données* – Observation de données d'utilisateur au cours d'une communication par un utilisateur non autorisé;

e) *manipulation* – Remplacement, insertion, suppression ou mise en désordre de données d'utilisateur au cours d'une communication par un utilisateur non autorisé;

f) *rejet* – Démenti d'un utilisateur d'avoir participé en partie ou pendant toute sa durée, à une communication;

g) *refus de service* – Empêchement ou interruption d'une communication ou encore retard d'opérations temporelles critiques;

NOTE 1 – Ce danger est très général et dépend de l'application sur le plan individuel ou de l'intention de la brusque interruption non autorisée; il ne fait donc pas explicitement partie de l'objectif du cadre d'authentification.

h) *acheminement erroné* – Acheminement erroné d'un chemin de communication prévu pour un usager vers un autre usager;

NOTE 2 – L'acheminement erroné apparaîtra naturellement dans les couches OSI 1 à 3. L'acheminement erroné est donc en dehors de l'objectif du cadre d'authentification. Cependant, il peut être possible d'éviter les conséquences d'un acheminement erroné en utilisant les services de sécurité fournis dans le cadre d'authentification.

i) *analyse de trafic* – Observation de l'information relative à une communication entre utilisateurs (c'est-à-dire absence ou présence, fréquence, sens, séquence, type, volume, etc.).

NOTE 3 – Les dangers provenant de l'analyse de trafic ne concernent naturellement pas exclusivement une couche OSI déterminée. Donc l'analyse du trafic est généralement en dehors de l'objectif du cadre d'authentification. Toutefois, on peut assurer partiellement une protection contre l'analyse de trafic par la production d'un trafic additionnel inintelligible (remplissage de trafic) en utilisant des données chiffrées ou aléatoires.

### B.2 Services de sécurité

Afin d'assurer la protection contre les dangers perçus, il est nécessaire de prévoir divers services de sécurité. Les services de sécurité que fournit le cadre d'authentification sont assurés au moyen des mécanismes de sécurité décrits au B.3:

a) *authentification des entités homologues* – Ce service donne la confirmation qu'un utilisateur, dans un certain cas de communication, est bien l'utilisateur demandé. Deux services différents d'authentification des entités homologues peuvent être demandés:

- *authentification d'entité unique* (soit authentification d'entité d'*expéditeur de données* soit authentification d'entité de *destinataire de données*);
- *authentification mutuelle* dans laquelle les deux utilisateurs en communication s'authentifient mutuellement;

2) Pour plus d'information, voir l'ISO 7498-2.

lorsqu'ils demandent un service d'authentification des entités homologues, les deux utilisateurs décident de concert si leurs identités seront protégées ou non;

le service d'authentification des entités homologues est assuré par le cadre d'authentification. On peut l'utiliser pour la protection contre une fausse identité et un redéfilement en ce qui concerne l'identité des utilisateurs;

b) *commande d'accès* – On peut employer ce service contre l'utilisation non autorisée de ressources. Le service de commande d'accès est fourni par l'Annuaire ou par une autre application et ne concerne donc pas le cadre d'authentification;

c) *confidentialité de données* – On peut employer ce service pour assurer la protection des données contre la divulgation non autorisée des données. Le service de confidentialité des données est assuré par le cadre d'authentification. On peut l'employer pour la protection contre l'interception de données;

d) *intégrité de données* – Ce service fournit la preuve de l'intégrité des données dans une communication. Le service d'intégrité de données est assuré par le cadre d'authentification. On peut l'employer pour déceler des manipulations et assurer une protection contre celles-ci;

e) *non-rejet* – Ce service fournit la preuve de l'intégrité et de l'origine de données – les deux dans une relation infalsifiable – qui peuvent être vérifiées par un tiers à tout moment.

## B.3 Mécanismes de sécurité

Les mécanismes de sécurité indiqués dans ce paragraphe assurent les services de sécurité décrits au B.2:

a) *échange d'authentifications* – Les mécanismes d'authentification fournis par le cadre d'authentification comportent deux niveaux:

- *authentification simple* – S'appuie sur la fourniture par l'expéditeur de son nom et de son mot de passe, qui sont vérifiés par le destinataire;
- *authentification renforcée* – S'appuie sur l'utilisation des techniques cryptographiques pour protéger l'échange de validation d'information. Dans le cadre d'authentification, l'authentification renforcée est basée sur un schéma asymétrique;

le mécanisme d'échange d'authentifications est employé pour assurer le service d'authentification d'entité homologue;

b) *chiffrement* – Le cadre d'authentification couvre le chiffrement du transfert de données. On peut utiliser soit un schéma asymétrique soit un schéma symétrique. L'échange de clés nécessaire pour chacun des cas est assuré soit au cours d'un échange d'authentifications antérieur, soit indépendamment à n'importe quel moment avant la communication prévue. Ce dernier cas est en dehors de l'objectif du cadre d'authentification. Le mécanisme de chiffrement assure le service de la confidentialité de données;

c) *intégrité de données* – Le mécanisme fait intervenir le chiffrement d'une chaîne comprimée des données pertinentes à transférer. En même temps que les données en clair, ce message est envoyé au destinataire. Le destinataire reproduit la compression et le chiffrement qui suit des données en clair et compare le résultat avec celles qui ont été créées par l'expéditeur, pour en vérifier l'intégrité;

le mécanisme d'intégrité de données peut être fourni par le chiffrement des données en clair comprimées, soit par un schéma asymétrique, soit par un schéma symétrique (avec le schéma symétrique, la compression et le chiffrement de données peuvent être effectués simultanément). Le mécanisme n'est pas explicitement fourni par le cadre d'authentification. Cependant, il est totalement fourni en tant que partie du mécanisme de signature numérique (voir ci-dessous) comportant un schéma asymétrique;

le mécanisme d'intégrité de données assure le service d'intégrité de données. Il assure aussi partiellement le service de non-rejet (ce service a besoin également du mécanisme de signature numérique pour que ses exigences soient entièrement satisfaites);

d) *signature numérique* – Ce mécanisme fait intervenir le chiffrement au moyen de la clé privée de l'expéditeur, d'une chaîne comprimée des données pertinentes à transférer. La signature numérique de même que les données en clair sont envoyées au destinataire. De la même façon que dans le cas du mécanisme d'intégrité de données, ce message est traité par le destinataire pour en vérifier l'intégrité. Le mécanisme de signature numérique prouve également l'authenticité de l'expéditeur et les relations sans ambiguïté entre l'expéditeur et les données qui ont été transférées;

le cadre d'authentification assure le mécanisme de signature numérique comportant un schéma asymétrique;

le mécanisme de signature numérique assure le service d'intégrité de données et assure également le service de non-rejet.

## B.4 Dangers contre lesquels la protection est assurée par les services de sécurité

Le Tableau B.1 indique les dangers susceptibles de porter atteinte à la sécurité, contre lesquels chaque service de sécurité peut assurer la protection. La présence d'un point centré ("•") indique qu'un service donné de sécurité assure la protection contre un danger donné.

**Tableau B.1 – Dangers et protection**

| Dangers | Services | | | |
|---|---|---|---|---|
| | Authentification d'entité | Confidentialité des données | Intégrité des données | Non-rejet |
| Interception d'identité | • (si nécessaire) | | | |
| Interception de données | | • | | |
| Usurpation d'identité | • | | | |
| Redéfilement | • (identité) | | • (données) | • |
| Manipulation | | | • | |
| Rejet | | | | • |

## B.5 Négociation des services et des mécanismes de sécurité

La fourniture des caractéristiques de sécurité au cours d'un cas de communication nécessite la négociation du contexte dans lequel les services de sécurité sont exigés. Cela conduit à un accord sur le type de mécanismes de sécurité et les paramètres de sécurité qui sont nécessaires pour fournir ces services de sécurité. Les procédures requises pour ces négociations des mécanismes et des paramètres peuvent être appliquées en considérant soit qu'elles font corps avec la procédure normale d'établissement de communication, soit qu'elles constituent un traitement à part. Les détails précis de ces procédures de négociation ne sont pas spécifiés dans la présente annexe.

# Annexe C

## Introduction à la cryptographie de clé publique[3)]

(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 à 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 C.1.

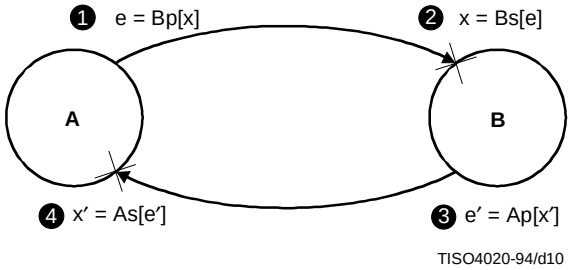



**Figure C.1 – Utilisation d'un PKCS pour l'échange d'information secrète**

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 C.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:

$$x = Bs[e], \text{ ou } x = Bs[Bp[x]]$$

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.

---

3) Pour complément d'information, consulter:

DIFFIE (W.) et HELLMAN (M. E.): New Directions in Cryptography, *IEEE Transactions on Information Theory*, IT-22 n° 6 (novembre 1976).

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 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 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 D

## Le système cryptographique RSA[4] de clé publique[5]

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

### D.1 Objectif et domaine d'application

Un exposé complet sur le système RSA dépasse l'objectif de la présente annexe. Toutefois, on trouvera ci-après la description succincte de la méthode, qui repose sur l'utilisation d'une élévation à une puissance d'un module.

### D.2 Définitions

Les termes suivants sont définis dans la présente annexe.

**D.2.1** **clé publique**: paire de paramètres qui se compose de l'exposant public et du module arithmétique.

NOTE – L'élément de données **subjectPublicKey** ASN.1, défini comme **BIT STRING** (voir l'Annexe A) doit être interprété, s'agissant de RSA, comme étant du type:

**SEQUENCE {INTEGER, INTEGER}**

où le premier nombre entier est le module arithmétique, le second étant l'exposant public. La séquence est représentée au moyen des règles de codage de base ASN.1.

**D.2.2** **clé privée**: paire de paramètres qui se compose de l'exposant secret et du module arithmétique.

### D.3 Symboles et abréviations

Pour les besoins de la présente annexe, les symboles et abréviations suivants s'appliquent:

| | |
|---|---|
| X, Y | Blocs de données qui sont arithmétiquement inférieurs au module |
| n | Module arithmétique |
| e | Exposant public |
| d | Exposant secret |
| p, q | Nombres premiers dont le produit forme le module arithmétique (n) |
| | NOTE – Bien que les nombres premiers soient de préférence au nombre de deux, l'utilisation d'un module avec trois ou plusieurs facteurs premiers n'est pas à écarter. |
| lcm | Plus petit commun multiple |
| mod n | Modulo arithmétique n |

---

4) Le système cryptographique spécifié dans la présente annexe, inventé par R. L. Rivest, A. Shamir et L. Adleman, est bien connu sous le sigle *RSA*.

5) Pour un complément d'information, consulter:

Généralités:

RIVEST (R. L.), SHAMIR (A.) et ADLEMAN (L.): A Method for Obtaining Digital Signatures and Public-key Cryptosystems, *Communications of the ACM*, 21, 2, 120-126, février 1978.

Production des clés:

GORDON (J.): Strong RSA Keys, *Electronics Letters*, 20, 5, 514-516.

Référence – Déchiffrement:

QUISQUATER (J. J.) et COUVREUR (C.): Fast Decipherment Algorithm for RSA Public-key Cryptosystems, *Electronics Letters*, 18, 21, 905-907, 14 octobre 1982.

## D.4 Description

Cet algorithme asymétrique utilise la fonction puissance pour les transformations de blocs de données telles que:

– $Y = X^e \bmod n$ avec $0 \le X < n$;

– $X = Y^d \bmod n$ avec $0 \le Y < n$,

qui peuvent être satisfaites par exemple pour:

– $ed \bmod \text{lcm}(p-1,q-1) = 1$; ou

– $ed \bmod (p-1)(q-1) = 1$.

Pour effectuer ce processus, un bloc de données doit être interprété comme un nombre entier. Pour cela, on considère que la totalité du bloc de données est une séquence de bits ordonnée (par exemple de longueur $\lambda$) dont le premier bit est l'élément binaire de poids le plus élevé dans le premier octet du bloc de données. Le nombre entier est alors formé comme étant la somme des bits après avoir donné au premier bit le poids $2^{\lambda-1}$ et divisé ce poids par 2 pour chaque bit suivant (le dernier bit aura le poids 1).

La longueur du bloc de données doit être le plus grand nombre d'octets contenant moins de bits que le module. Les blocs incomplets doivent être complétés de la façon que l'on désire. Un nombre de blocs de remplissage additionnel quelconque peut être ajouté.

## D.5 Prescriptions de sécurité

### D.5.1 Longueurs de clé

Il est reconnu que la longueur de clé acceptable est susceptible de changer avec le temps, en fonction du coût et de la disponibilité du matériel, du temps passé, des progrès techniques et du niveau de sécurité requis. Il est recommandé d'adopter initialement une valeur de 512 bits pour la longueur de n, mais sous réserve d'un complément d'étude.

### D.5.2 Production de clés

La sécurité du système RSA repose sur la difficulté d'effectuer la factorisation de n. De nombreux algorithmes permettent d'effectuer cette opération et, afin de faire obstacle à l'emploi de toute technique actuellement connue, les valeurs p et q doivent être minutieusement choisies compte tenu des règles suivantes (par exemple voir la Note 5 de bas de page, "production des clés"):

a) elles devront être choisies au hasard;

b) elles devront être élevées;

c) elles devront être des nombres premiers;

d) |p−q devra avoir une valeur élevée;

e) (p+1) devra posséder un facteur premier élevé;

f) (q+1) doit posséder un facteur premier élevé;

g) (p−1) doit posséder un facteur premier élevé, soit r;

h) (q−1) doit posséder un facteur premier élevé, soit s;

i) (r−1) doit posséder un facteur premier élevé;

j) (s−1) doit posséder un facteur premier élevé.

Après la production des clés publique et privée, par exemple "Xp" et "Xs" définies aux articles 3 et 4 et qui sont constituées par d, e, et n, il serait préférable de détruire les valeurs p et q de même que toutes les autres données produites telles que le produit (p−1)(q−1) et les facteurs premiers de valeur élevée. Néanmoins, la conservation de p et de q localement peut multiplier par 2 ou par 4 le débit du déchiffrement. La décision de conserver p et q est considérée comme un problème à résoudre à l'échelon local (voir la Note 5 de bas de page "Référence – Déchiffrement").

Il convient de faire en sorte que e soit $> \log_2(n)$ afin d'éviter une attaque préparée en prenant la racine $e\text{-}^e$ de mod n pour découvrir le texte en clair.

## D.6 Exposant public

L'exposant public (e) pourra être commun à tout l'environnement afin de minimiser la longueur de la partie de clé publique qui doit être effectivement diffusée et de réduire la capacité de transmission et la complexité de la transformation (voir la Note 1).

L'exposant e devra être assez grand mais tel que l'élévation à une puissance puisse être effectuée efficacement en ce qui concerne la durée du traitement et la capacité de mémoire. Si l'on désire un exposant public fixe e, il y a grand avantage à utiliser le nombre de Fermat $F_4$ (voir la Note 2).

$$F_4 = 2^{2^4} + 1$$

= 65537 en numérotation décimale, et

= 1 0000 0000 0000 0001 en numérotation binaire

NOTE 1 – Bien que le module n et l'exposant e soient tous les deux publics, le module ne doit pas être la partie commune à un groupe d'utilisateurs. La connaissance du module "n", de l'exposant public "e" et de l'exposant secret "d" suffit pour déterminer la factorisation de "n". Par conséquent, si le module est commun, chacun peut déduire ses facteurs et par-là découvrir l'exposant secret de tous les autres.

NOTE 2 – L'exposant fixé devra avoir une valeur élevée et être un nombre premier, mais il devra également assurer un traitement efficace. Le nombre de Fermat $F_4$ répond à ces exigences, par exemple l'authentification nécessite seulement 17 multiplications et est en moyenne 30 fois plus rapide que le déchiffrement.

## D.7 Conformité

Si la présente annexe spécifie un algorithme pour les fonctions publiques et secrètes, elle ne définit pas la méthode employée pour effectuer les calculs; il est donc possible qu'il y ait des produits différents qui satisfassent à cette annexe et qui soient mutuellement compatibles.

# Annexe E

# Fonctions de hachage

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

La fonction de hachage carré-mod qui était décrite dans la présente annexe de la première édition de la présente Spécification d'Annuaire est déconseillée.

## E.1 Prescriptions pour les fonctions de hachage

Pour utiliser une fonction de hachage comme fonction sûre à une voie, il ne doit pas être possible d'obtenir facilement le même résultat de hachage à partir de combinaisons différentes du message d'entrée.

Une fonction de hachage à haut niveau de sécurité devra satisfaire aux prescriptions suivantes:

a) la fonction doit être à une voie, c'est-à-dire que, quel que soit le résultat de hachage possible, il doit être impossible par calcul de construire un message d'entrée qui soit réduit à ce résultat;

b) la fonction de hachage doit être exempte de collision, c'est-à-dire qu'il doit être impossible par calcul de construire deux messages d'entrée distincts qui se réduisent au même résultat.

# Annexe F

## Dangers contre lesquels la protection est assurée par les méthodes d'authentification renforcée

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

La méthode d'authentification renforcée décrite dans la présente Spécification d'Annuaire assure une protection contre les dangers, comme cela est décrit dans la partie de l'Annexe B relative à l'authentification renforcée.

De plus, il existe une gamme de dangers potentiels qui tiennent à la méthode d'authentification renforcée elle-même, à savoir:

- *compromission de la clé privée d'utilisateur* – Un des principes de base de l'authentification renforcée est que la clé privée reste sûre. Plusieurs méthodes pratiques sont à la disposition de l'utilisateur pour qu'il conserve sa clé privée de façon qu'elle assure une sécurité satisfaisante. Les conséquences de la compromission sont limitées à la déstabilisation de communication touchant l'utilisateur concerné;
- *compromission de la clé privée d'une CA* – Un principe de base d'une authentification renforcée est également que la clé privée d'une CA reste sûre. Une sécurité matérielle et des méthodes de "nécessité d'accès" sont applicables. Les conséquences de la compromission sont limitées à la déstabilisation de communication touchant tout utilisateur certifié par cette CA;
- *fait d'induire en erreur une CA par la délivrance d'un certificat non valide* – Le fait que les CA soient indépendantes offre une certaine protection. La CA a la responsabilité de vérifier que les accréditations données comme sérieuses sont valides avant d'établir un certificat. Les conséquences de la compromission sont limitées à la déstabilisation de communication touchant l'utilisateur pour lequel le certificat a été établi et quiconque ayant été concerné par le certificat non valide;
- *collusion entre une CA malhonnête et un utilisateur* – Une telle attaque collusoire mettra la méthode en défaut. Cela constituera une trahison de la confiance témoignée à la CA. Les conséquences d'une CA malhonnête sont limitées à la déstabilisation de communication touchant n'importe quel utilisateur certifié par la CA;
- *falsification d'un certificat* – La méthode d'authentification renforcée assure la protection contre la falsification d'un certificat, du fait que la CA doit le signer. La méthode dépend de la conservation du secret de la clé privée de la CA;
- *falsification d'un jeton* – La méthode d'authentification renforcée assure la protection contre la falsification d'un jeton, du fait que l'expéditeur doit le signer. La méthode dépend de la conservation du secret de la clé privée de l'expéditeur;
- *redéfilement d'un jeton* – Les méthodes d'authentification à une ou deux voies assurent la protection contre le redéfilement d'un jeton par l'introduction de la date et de l'heure dans le jeton. La méthode à trois voies assure également la protection par vérification des nombres aléatoires;
- *attaque du système cryptographique* – Les possibilités d'une analyse cryptographique efficace du système dépendant des progrès réalisés dans la théorie du calcul des nombres et conduisant à la nécessité d'une clé de plus grande longueur peuvent valablement être annoncées à l'avance.

# Annexe G

# Confidentialité des données

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

## G.1 Introduction

Le traitement de la confidentialité des données peut être initialisé après que les clés nécessaires au chiffrement ont été échangées. Celles-ci pourraient être fournies lors d'un échange d'authentifications précédent, comme décrit à l'article 9 ou par tout autre processus d'échange de clés (ce dernier n'entre pas dans le cadre de la présente Spécification d'Annuaire).

La confidentialité des données peut être assurée en appliquant soit un schéma de chiffrement symétrique, soit un schéma de chiffrement asymétrique.

## G.2 Confidentialité des données par chiffrement asymétrique

Dans ce cas, la confidentialité des données est assurée au moyen du chiffrement des données à transmettre par l'expéditeur qui utilise la clé publique du destinataire auquel les données sont destinées: le destinataire les déchiffrera alors en utilisant sa clé privée.

## G.3 Confidentialité des données par chiffrement symétrique

Dans ce cas, la confidentialité des données s'obtient au moyen d'un algorithme de chiffrement symétrique. Ce choix est en dehors de l'objectif du cadre d'authentification.

Au cas où l'échange d'authentifications conformément à l'article 9 a été effectué par les deux parties concernées, une clé pour l'utilisation d'un algorithme symétrique peut être déterminée. Le choix des clés privées dépend de la transformation à effectuer. Les parties doivent être sûres que ce sont des clés à haut niveau de sécurité. La présente Spécification d'Annuaire ne spécifie pas la manière dont s'effectue ce choix bien qu'il soit manifestement nécessaire que cela ait l'accord des parties concernées ou que cela soit spécifié dans d'autres normes.

# Annexe H

## 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**".

---

```
AlgorithmObjectIdentifiers {joint-iso-ccitt ds(5) module(1) algorithmObjectIdentifiers(8) 3}
DEFINITIONS ::=
BEGIN

-- EXPORTE 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'obligeront pas
-- à apporter les extensions et les modifications nécessaires pour tenir à jour ou améliorer le service d'Annuaire.

IMPORTS
    algorithm, authenticationFramework
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 3}
    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 --

rsa   ALGORITHM                            ::=     {
      KeySize
      IDENTIFIED BY    id-ea-rsa }
KeySize                                    ::=     INTEGER
-- assignation d'identificateurs d'objet --

id-ea-rsa               OBJECT IDENTIFIER  ::=     {id-ea 1}
-- l'assignation 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 I

## Bibliographie

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

KENT (S.): Privacy Enhancement for Internet Electronic Mail, Part II: Certificate-Based Key Management, RFC 1422, *Internet Activities Board*, 1993.

Nations Unies, EDIFACT Security Implementation Guidelines, (*Conseil économique et social des Nations Unies*), TRADE/WP.4/R.1026, 7 février 1994. (A incorporer dans la syntaxe EDIFACT selon ISO 9735.)

# Annexe J

## Exemples d'utilisation de contraintes de chemin de certification

(Cette annexe ne fait pas partie intégrante de la présente Recommandation | Norme internationale)

### J.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 }**

### J.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}}}**

### J.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 K

## 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 modifications suivantes:

- Amendement 1 pour les extensions de certificat;
- Amendement 2 pour le renforcement de la sécurité opérationnelle de l'Annuaire.

Cette édition de la présente Spécification d'Annuaire comprend les corrigenda techniques suivants, concernant les erreurs signalées dans le Relevé d'erreurs suivant (dont certaines parties peuvent avoir été rendues caduques par les amendements qui ont formé la présente édition de cette Spécification d'Annuaire):

- Corrigendum technique 1 (reprenant le Relevé d'erreurs 128);
- Corrigendum technique 2 (reprenant les Relevés d'erreurs 077, 078, 083, 084);
- Corrigendum technique 3 (reprenant les Relevés d'erreurs 080, 092, 100, 177);
- Corrigendum technique 1 de l'édition 97 (reprenant les Relevés d'erreurs 183, 194).

# Remplacée par une version plus récente

## 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 pour données et communication entre systèmes ouverts** |
| Série Y | Infrastructure mondiale de l'information |
| Série Z | Langages et aspects informatiques généraux des systèmes de télécommunication |