Remplacée par une version plus récente

UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T

X.509

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(11/93)

RÉSEAUX DE COMMUNICATION DE 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

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. Au sein de l'UIT-T, qui est l'entité qui établit les normes mondiales (Recommandations) sur les télécommunications, participent quelque 179 pays membres, 84 exploitations de télécommunications reconnues, 145 organisations scientifiques et industrielles et 38 organisations internationales.

L'approbation des Recommandations par les Membres de l'UIT-T s'effectue selon la procédure définie dans la Résolution n° 1 de la Conférence mondiale de normalisation des télécommunications (CMNT), (Helsinki, 1993). De plus, la CMNT, qui se réunit tous les quatre ans, approuve les Recommandations qui lui sont soumises et établit le programme d'études pour la période suivante.

Dans certains secteurs de la technologie 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. Le texte de la Recommandation X.509 de l'UIT-T a été approuvé le 16 novembre 1993. Son texte est publié, sous forme identique, comme Norme internationale ISO/CEI 9594-8.

NOTE

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

© UIT 1995

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

RECOMMANDATIONS UIT-T DE LA SÉRIE X

RÉSEAUX POUR DONNÉES ET INTERCONNEXION DES SYSTÈMES OUVERTS

(Février 1994)

ORGANISATION DES RECOMMANDATIONS DE LA SÉRIE X

Domaine	Recommandations
RÉSEAUX PUBLICS POUR DONNÉES	
Services et services complémentaires	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
Définition 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 de couche	X.280-X.289
Test de conformité	X.290-X.299
INTERFONCTIONNEMENT DES RÉSEAUX	
Considérations générales	X.300-X.349
Systèmes mobiles de transmission de données	X.350-X.369
Gestion	X.370-X.399
SYSTÈMES DE MESSAGERIE	X.400-X.499
ANNUAIRE	X.500-X.599
RÉSEAUTAGE OSI ET ASPECTS DES SYSTÈMES	
Réseautage	X.600-X.649
Dénomination, adressage et enregistrement	X.650-X.679
Notation de syntaxe abstraite numéro un (ASN.1)	X.680-X.699
GESTION OSI	X.700-X.799
SÉCURITÉ	X.800-X.849
APPLICATIONS OSI	
Engagement, concomitance et rétablissement	X.850-X.859
Traitement des transactions	X.860-X.879
Opérations distantes	X.880-X.899
TRAITEMENT OUVERT RÉPARTI	X.900-X.999

Remplacée par une version plus récente

TABLE DES MATIÈRES

Page

SECTION 1 – CONSIDÉRATIONS GÉNÉRALES 1

1 Domaine d'application 1

2 Références normatives 2

2.1 Recommandations | Normes internationales identiques 2

2.2 Paires de Recommandations | Normes internationales équivalentes par leur contenu technique 2

3 Définitions 3

3.1 Définitions relatives à l'architecture de sécurité du modèle de référence OSI 3

3.2 Définitions relatives au modèle d'Annuaire 3

3.3 Définitions relatives au cadre d'authentification 3

4 Abréviations 4

5 Conventions 4

SECTION 2 – AUTHENTIFICATION SIMPLE 5

6 Procédure d'authentification simple 5

6.1 Génération de l'information d'identification protégée 6

6.2 Procédure d'authentification simple protégée 7

6.3 Type d'attribut de mot de passe d'utilisateur 7

SECTION 3 – AUTHENTIFICATION POUSSÉE 8

7 Bases de l'authentification poussée 8

8 Obtention d'une clé publique d'usager 8

8.1 Optimisation de la quantité d'information obtenue de l'Annuaire 10

8.2 Exemple 11

9 Signatures numériques 13

10 Procédures d'authentification poussée 14

10.1 Vue d'ensemble 14

10.2 Authentification à une voie 15

10.3 Authentification à deux voies 16

10.4 Authentification à trois voies 16

11 Gestion des clés et des certificats 17

11.1 Génération de paires de clés 17

11.2 Gestion des certificats 17

Annexe A – Cadre d'authentification en ASN.1 19

Annexe B – Exigences de sécurité 22

B.1 Dangers 22

B.2 Services de sécurité 22

B.3 Mécanismes de sécurité 23

B.4 Dangers contre lesquels la protection est assurée par les services de sécurité 24

B.5 Négociation des services et des mécanismes de sécurité 25

Annexe C – Introduction à la cryptographie de clé publique 25

Remplacée par une version plus récente

Page

Annexe D – Le système RSA cryptographique de clé publique 27
D.1 Objectif et domaine d'application 27
D.2 Définitions 27
D.3 Symboles et abréviations 27
D.4 Description 28
D.5 Exigences de sécurité 28
D.6 Exposant public 29
D.7 Conformité 29

Annexe E – Fonctions hachage 30
E.1 Exigences pour les fonctions hachage 30

Annexe F – Dangers contre lesquels la protection est assurée par les méthodes d'authentification poussée 31

Annexe G – Confidentialité des données 32
G.1 Introduction 32
G.2 Confidentialité des données par chiffrement asymétrique 33
G.3 Confidentialité des données par chiffrement symétrique 33

Annexe H – Définition de référence des identificateurs d'objet d'algorithme 34

Annexe J – Modifications et Corrigendums 35

Remplacée par une version plus récente

Résumé

La présente Recommandation | Norme internationale définit un cadre de travail pour l'authentification des services par l'Annuaire au bénéfice de ses utilisateurs. Elle décrit deux niveaux d'authentification: l'authentification simple, utilisant un mot de passe pour vérifier l'identité déclarée, et l'authentification renforcée, faisant intervenir des pouvoirs 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.

Remplacée par une version plus récente

Introduction

La présente Recommandation | Norme internationale a été élaborée, de concert avec les 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 ces systèmes, avec les informations d'annuaire qu'ils contiennent, peut être considéré comme un tout intégré, appelé l'*Annuaire*. Les informations contenues dans l'Annuaire, appelées collectivement base d'informations d'Annuaire (DIB) sont généralement utilisées pour faciliter la communication entre des objets tels que entités d'application, individus, terminaux, listes de distribution, ainsi que les communications avec ces objets ou au sujet de ces objets.

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; et
- d'âges différents.

Un grand nombre d'applications comportent des exigences 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 menaces généralement connues ainsi que les services et les mécanismes de sécurité qu'on peut utiliser pour la protection contre ces dernières. En fin de compte, 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 d'authentification pour assurer la prestation des services d'authentification par l'Annuaire à ses utilisateurs. Ces utilisateurs comprennent l'Annuaire lui-même ainsi que d'autres applications et services. L'Annuaire peut utilement contribuer à répondre à leurs besoins en authentification et en d'autres services de sécurité car c'est un emplacement naturel à partir duquel les parties en communication peuvent obtenir l'information d'authentification des uns et des autres: connaissance sur laquelle repose l'authentification. L'Annuaire est l'emplacement naturel du fait qu'il détient d'autres informations qui sont nécessaires à la communication et qui sont obtenues avant l'établissement de la communication. L'obtention de l'information d'authentification d'un partenaire d'une communication potentielle à partir de l'Annuaire est, avec cette approche, semblable à l'obtention d'une adresse. En raison du domaine étendu recouvert par l'Annuaire, on prévoit que ce cadre d'authentification sera très utilisé par toute une gamme d'applications.

Cette seconde édition révise techniquement et améliore, mais ne remplace pas, la première édition de la présente Recommandation | Norme internationale. Les mises en œuvre peuvent encore prétendre à la conformité à la première édition.

Cette seconde édition spécifie la version 1 des protocoles et services de l'Annuaire. La première édition spécifie également version 1. On a traité les différences entre les services et les protocoles définis dans les deux éditions en utilisant les règles d'extensibilité définies dans la présente version de la Rec. X.519 | ISO/CEI 9594-5.

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 exigences 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.

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

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

Remplacée par une version plus récente

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 de consignation.

L'Annexe J, qui ne fait pas partie intégrante de la présente Recommandation | Norme internationale, recense les modifications et les rapports de défaut qui ont été incorporés dans cette édition de la présente Recommandation | Norme internationale.

NORME INTERNATIONALE

RECOMMANDATION UIT-T

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

SECTION 1 – CONSIDÉRATIONS GÉNÉRALES

1 Domaine d'application

La présente Spécification | Norme internationale:

- spécifie la forme sous laquelle l'information d'authentification est conservée par l'Annuaire;
- décrit la façon dont on peut obtenir de l'Annuaire cette information d'authentification;
- établit les hypothèses faites au sujet de la manière dont est constituée cette information d'authentification et de son emplacement dans l'Annuaire;
- définit trois manières dont les applications peuvent employer cette information d'authentification pour effectuer des authentifications et explique comment d'autres services de sécurité peuvent être assurés par l'authentification.

La présente Recommandation | Norme internationale contient les descriptions de deux niveaux d'authentification: l'authentification simple qui utilise un mot de passe pour la vérification de l'identité et l'authentification poussée qui fait intervenir des accréditations 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 poussée devra servir de base à la prestation de services sûrs. Il ne s'agit pas ici d'établir un cadre général d'authentification. Celui-ci peut toutefois se prêter à une utilisation générale pour des applications qui jugent 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 particulière. Il appartient aux utilisateurs de cette application de définir leur propre politique de sécurité qui peut dépendre des services fournis par une norme.

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

La méthode d'authentification poussée 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 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 sont mis en place dans l'Annuaire par leur créateur. La génération de certificats d'utilisateur est effectuée par une autorité de certification autonome qui est complètement distincte des 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 en 7.1. 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 des 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 hachage [voir 3.3 f)] (utilisée pour former des accréditations et des jetons d'authentification). De nouveau, en principe, on peut utiliser un certain nombre de variantes de fonction hachage au prix d'un rétrécissement des communautés des 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.500 (1993) | ISO/CEI 9594-1:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Vue d'ensemble des concepts, modèles et services*.
- Recommandation UIT-T X.501 (1993) | ISO/CEI 9594-2:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Les modèles*.
- Recommandation UIT-T X.511 (1993) | ISO/CEI 9594-3:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Définition du service abstrait*.
- Recommandation UIT-T X.518 (1993) | ISO/CEI 9594-4:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Procédures pour le fonctionnement réparti*.
- Recommandation UIT-T X.519 (1993) | ISO/CEI 9594-5:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Spécifications du protocole*.
- Recommandation UIT-T X.520 (1993) | ISO/CEI 9594-6:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Types d'attributs sélectionnés*.
- Recommandation UIT-T X.521 (1993) | ISO/CEI 9594-7:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Classes d'objets sélectionnées*.
- Recommandation UIT-T X.525 (1993) | ISO/CEI 9594-9:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – L'Annuaire: Duplication*.
- Recommandation UIT-T X.680 (1994) | ISO/CEI 8824-1:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation de syntaxe abstraite numéro un: Spécification de la notation de base.*
- Recommandation UIT-T X.681 (1994) | ISO/CEI 8824-2:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation de syntaxe abstraite numéro un: Spécification des objets informationnels.*
- Recommandation UIT-T X.682 (1994) | ISO/CEI 8824-3:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation de syntaxe abstraite numéro un: Spécification des contraintes.*
- Recommandation UIT-T X.683 (1994) | ISO/CEI 8824-4:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Notation de syntaxe abstraite numéro un: Paramétrage des spécifications de la notation de syntaxe abstraite n° 1.*
- Recommandation UIT-T X.690 (1994) | ISO/CEI 8825-1:1994, *Technologie de l'information – Interconnexion des systèmes ouverts – Règles de codage de l'ASN.1: Spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives.*
- Recommandation UIT-T X.880 (1994) | ISO/CEI 13712-1:1994, *Technologie de l'information – Opérations distantes: Concepts, modèle et notation.*
- Recommandation UIT-T X.881 (1994) | ISO/CEI 13712-2:1994, *Technologie 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 UIT-T 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.*

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 | ISO 7498-2:

a) *asymétrique (chiffrement)*;

b) *échange d'authentifications*;

c) *information d'authentification*;

d) *confidentialité*;

e) *justificatif d'identité*;

f) *cryptographie*;

g) *authentification de l'origine des données*;

h) *déchiffrement*;

i) *chiffrement*;

j) *clé*;

k) *mot de passe*;

l) *authentification de l'entité homologue*;

m) *symétrique (chiffrement)*.

3.2 Définitions relatives au modèle d'Annuaire

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

a) *attribut*;

b) *base d'informations d'Annuaire*;

c) *arbre d'informations d'Annuaire*;

d) *agent de système d'Annuaire*;

e) *agent d'utilisateur d'Annuaire*;

f) *nom distintif*;

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 jeton d'authentification (jeton): Information acheminée au cours d'un échange d'authentifications, qui peut être utilisée à authentifier son expéditeur.

3.3.2 certificat d'utilisateur (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.

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

3.3.4 itinéraire de certification: Séquence ordonnée de certificats d'objets dans le DIT qui en plus de la clé publique de l'objet initial dans l'itinéraire, peut être traitée pour obtenir celle de l'objet final dans l'itinéraire.

3.3.5 système cryptographique: 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.

3.3.6 fonction hachage: Fonction (mathématique) qui met en correspondance les valeurs d'un grand (et éventuellement très grand) domaine avec une gamme 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 la gamme.

3.3.7 fonction à une voie: Fonction (mathématique) f qui est facile à calculer mais pour laquelle, à une valeur générale y de la gamme, 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.

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

3.3.9 clé privée; clé secrète (déconseillée): (dans un système cryptographique de clé publique) Clé d'une paire de clés d'utilisateur qui n'est connue que de cet utilisateur.

3.3.10 authentification simple: Authentification obtenue au moyen de simples arrangements de mot de passe.

3.3.11 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.

3.3.12 authentification poussée: Authentification obtenue au moyen d'accréditations déterminées par cryptographie.

3.3.13 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 certaine fonction particulière. Le rôle de confiance qui revient à la clé 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.

3.3.14 numéro de série de certificat: Valeur entière, unique pour la CA qui l'émet et associée sans ambiguïté au certificat émis par cette CA.

4 Abréviations

Les abréviations suivantes sont utilisées dans la présente Recommandation | Norme internationale:

CA	Autorité de certification (*certification authority*)
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

Sauf exceptions mineures, 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, de mars 1993.

Le terme «Spécification d'Annuaire» (comme dans «la présente Spécification d'Annuaire») s'entend selon l'acception de la Rec. UIT-T X.509 | ISO/CEI 9594-8. Le terme «Spécifications d'Annuaire» s'entend selon l'acception des Recommandations de la série X.500 et de toutes les parties de l'ISO/CEI 9594.

Dans la présente Spécification d'Annuaire le terme «systèmes de l'édition 1988» désigne les systèmes conformes à l'édition précédente (1988), c'est-à-dire l'édition 1988 des Recommandations de la série X.500 du CCITT et ISO/CEI 9594: édition 1990. Pour les systèmes conformes aux Spécifications actuelles d'Annuaire on utilise le terme «systèmes de l'édition 1993».

Si, dans une liste, les points sont numérotés (au lieu d'utiliser des tirets ou des lettres), ils seront alors 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 I avec un sommaire chiffré attaché.
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 produit le suivant. Il est fonctionnellement équivalent au certificat suivant 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_1p • 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 d'infixe, dont l'opérande de gauche est la clé publique d'une autorité de certification, et dont l'opérande de droite est un certificat délivré par cette autorité de certification. Le résultat est la clé publique de l'utilisateur dont le certificat est l'opérande de droite. Par exemple: Ap • A<<B>> B<<C>> 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	Itinéraire de certification de A en B composé d'une chaîne de certificats, débutant avec: CA(A)<<CA^2(A)>> et finissant avec CA(B)<<B>>.
NOTE – Dans ce tableau, les symboles X, X_1, X_2, etc., remplacent les noms des usagers; le symbole I remplace toute information arbitraire.	

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, un mot de passe faisant l'objet (facultativement) d'un accord bilatéral et 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 DUA et un DSA. L'authentification simple peut être réalisée de plusieurs manières:

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

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

NOTES

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

2 La signalisation des procédures de protection des mots de passe pourra faire l'objet d'un complément au présent document.

Quand les mots de passe ne sont pas protégés, un niveau de sécurité minimal est assuré pour empêcher un accès non autorisé. Il ne doit pas être considéré comme la base de services sûrs. La protection du nom distinctif de l'utilisateur et du mot de passe assure une sécurité plus grande. 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.

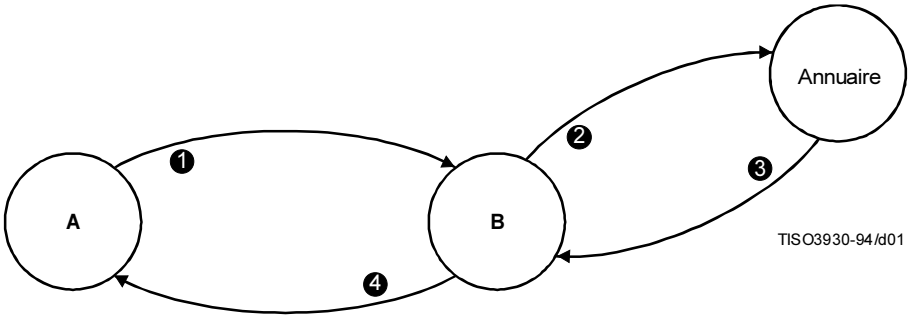


Figure 1 – Procédure d'authentification simple sans protection

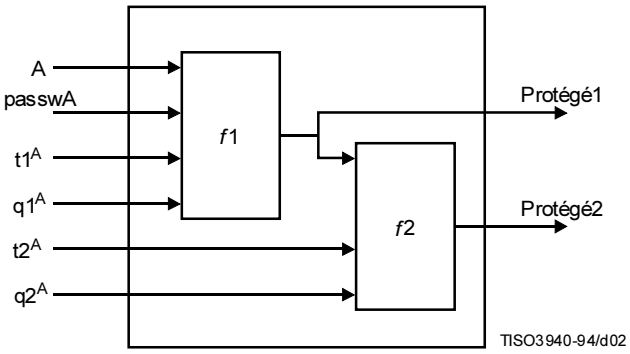
Les étapes de cette procédure sont les suivantes:

1) un utilisateur expéditeur A envoie son nom distinctif et son mot de passe à un utilisateur 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 détenu en tant qu'attribut **UserPassword** dans l'entrée d'annuaire concernant A (en utilisant l'opération Compare de l'Annuaire);
3) l'Annuaire confirme (ou dément) à B que les accréditations 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 Génération de l'information d'identification protégée

La Figure 2 montre deux méthodes de production d'une information d'identification protégée. *f*1 et *f*2 sont des fonctions à une voie (identiques ou différentes) et les indications horaires et les nombres aléatoires sont facultatifs et dépendent d'accords bilatéraux.



A	Nom distinctif de l'utilisateur
t^A	Indications horaires
$passw^A$	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.

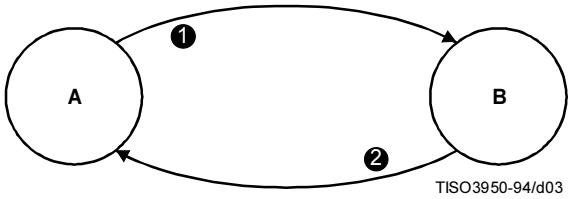


Figure 3 – Procédure d'authentification simple protégée

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

1) L'utilisateur d'origine (utilisateur 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:

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

   L'information transmise à B prend la forme:

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

   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 de 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.

2) B confirme (ou dément) 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 additionnellement 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 6.4.1].

2) B confirme (ou dément) 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 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 DUA lié à un DSA, B ou bien A pourrait être un DSA lié à un autre 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 POUSSÉE

7 Bases de l'authentification poussée

La façon d'aborder une authentification poussée 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). 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 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, $Xp \bullet Xs = Xs \bullet Xp$ si Xp/Xs sont des fonctions de chiffrement/déchiffrement utilisant les clés publique/privée de l'utilisateur X.

NOTE – On pourra ultérieurement envisager d'autres types de 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 d'appellation 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'usager

Pour qu'un utilisateur puisse avoir confiance en la procédure d'authentification, il doit obtenir la clé publique de l'autre utilisateur, d'une origine en laquelle il a confiance. Une telle origine, appelée autorité de certification (CA), 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 est certifiée;
- aucune partie autre que l'autorité de certification ne peut modifier le certificat sans que cela ne 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 n'ait à prendre des dispositions particulières pour assurer leur protection.

NOTE 1 – Bien que les CA soient définies sans ambiguïté par un nom distinctif dans le DIT, cela n'implique nullement une relation entre l'organisation des CA et le 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 à 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'un 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 est le numéro de série du certificat, AI est l'identificateur de l'algorithme servant à signer le certificat, UCA est l'identificateur unique facultatif de l'autorité CA, UA est 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 certificates.

```
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
        subjectUniqueIdentifier     [2]     IMPLICIT UniqueIdentifier OPTIONAL
                                                        -- si présent, la version doit être v2  -- }}

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

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 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       UTCTime,
        notAfter        UTCTime }

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

NOTE 2 – Dans les cas où l'autorité d'appellation peut réattribuer le nom distinctif à un autre utilisateur, les autorités CA peuvent utiliser l'identificateur unique pour distinguer les occurrences réutilisées. Toutefois, si plusieurs autorités CA fournissent des certificats au même utilisateur, il est recommandé que les autorités CA effectuent une coordination de l'attribution des identificateurs uniques, en tant que partie de leurs procédures d'enregistrement d'utilisateur.

L'entrée d'annuaire de chaque utilisateur, A, qui participe à une authentification poussée contient le(s) certificat(s) de A. Ce certificat est produit par une autorité de certification de A, qui est une entité du 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 ainsi 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 vers l'avant de X produits par d'autres autorités de certification, ensuite les certificats inverses 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 itinéraires 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 *itinéraire de certification*, chaque élément de la liste étant un certificat de l'autorité de certification pour le suivant. Un itinéraire de certification de A en B (désigné par A→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 itinéraire 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 itinéraires de certification A→B et B→A peut varier. Une façon d'y parvenir plus facilement consiste à établir une hiérarchie de CA pouvant coïncider ou non avec une partie, ou la totalité de la hiérarchie du DIT. L'avantage pour les utilisateurs qui ont des CA dans la hiérarchie est de pouvoir établir un itinéraire de certification entre eux au moyen de l'Annuaire sans information préalable. Pour cela, chaque CA peut stocker un certificat et un certificat inverse désigné comme correspondant à sa CA 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
    ID                      id-at-userCertificate }

cACertificate               ATTRIBUTE   ::=     {
    WITH SYNTAX             Certificate
    ID                      id-at-cAcertificate }

crossCertificatePair        ATTRIBUTE   ::=     {
    WITH SYNTAX             CertificatePair
    ID                      id-at-crossCertificatePair }

CertificatePair             ::=     SEQUENCE {
    forward         [0]     Certificate OPTIONAL,
    reverse         [1]     Certificate OPTIONAL
                                -- au moins une doit être présente -- }
```

Un utilisateur peut obtenir un ou plusieurs certificats en provenance d'une ou 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 itinéraires 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 l'itinéraire de certification vers l'avant. Il s'agit d'une composante contenant l'itinéraire de certification qui peut pointer vers l'arrière jusqu'à l'émetteur.

```
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 itinéraires 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, l'itinéraire de certification devient commun et les utilisateurs révèlent directement l'un à l'autre leur certificat;

b) si les CA des utilisateurs sont disposées par 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 du 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 que d'obtenir les itinéraires de certification à partir du point commun de confiance;

c) si un utilisateur est fréquemment en communication avec d'autres utilisateurs certifiés par une autre CA donnée, il pourra connaître l'itinéraire de certification vers cette CA et l'itinéraire retour de certification provenant de cette CA, 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 l'itinéraire 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 l'itinéraire de certification, les utilisateurs doivent vérifier la validité des certificats reçus.

8.2 Exemple

La Figure 4 représente un exemple fictif de fragment de DIT, dans lequel les CA forment une hiérarchie. Outre l'information indiquée aux CA, 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.

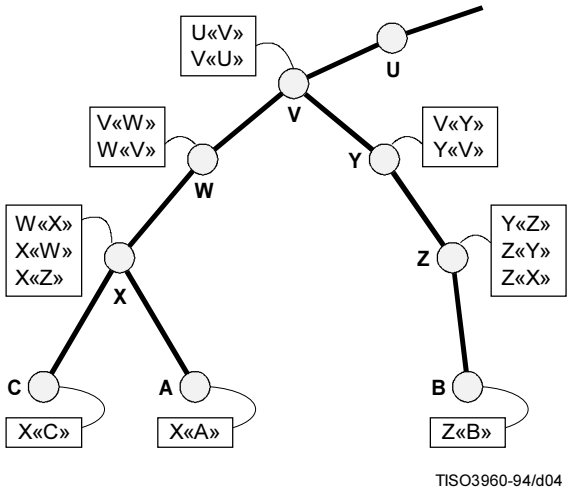


Figure 4 – Hiérarchie des CA – Exemple fictif

Si les CA des utilisateurs sont disposées hiérarchiquement, A peut acquérir les certificats suivants en provenance de l'Annuaire pour établir un itinéraire 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 l'itinéraire 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 provenant de l'Annuaire pour établir l'itinéraire 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 l'itinéraire 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 l'itinéraire de certification se réduit à:

$$Cp = Xp \bullet X\langle\langle C\rangle\rangle$$

et le dévoilement de l'itinéraire retour de certification se réduit à:

$$Ap = Xp \bullet X\langle\langle A\rangle\rangle$$

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 l'itinéraire de certification, se réduit à:

V<<Y>>, Y<<Z>>, Z<<B>>

et l'information que A doit obtenir de l'Annuaire pour former l'itinéraire 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 seulement 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 (ceci 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 l'itinéraire de certification, et:

Z<<X>>

pour former l'itinéraire 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:

$$Cp = Xp \bullet X\langle\langle C\rangle\rangle$$

et

$$Ap = Xp \bullet X\langle\langle A\rangle\rangle$$

Dans le cas plus général, les autorités de certification n'ont pas entre elles de relations hiérarchiques. Dans l'exemple fictif 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>>.

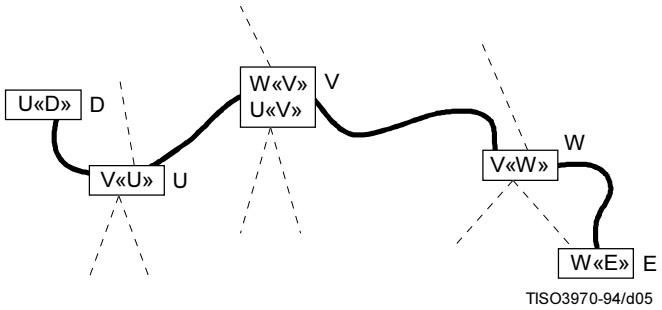


Figure 5 – Exemple d'itinéraire de certification non hiérarchique

Soit V une CA 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 itinéraire de certification vers E. Plusieurs méthodes peuvent être utilisées. L'une consiste à considérer les utilisateurs et les CA comme des nœuds et les certificats comme des arcs dans un graphique dirigé. Dans ces conditions, D doit rechercher dans le graphique un itinéraire de U vers E, qui pourra être U<<V>>, V<<W>>, W<<E>>. Une fois que cet itinéraire a été découvert, l'itinéraire 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 l'information. Ce résumé est produit au moyen d'une fonction 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)]

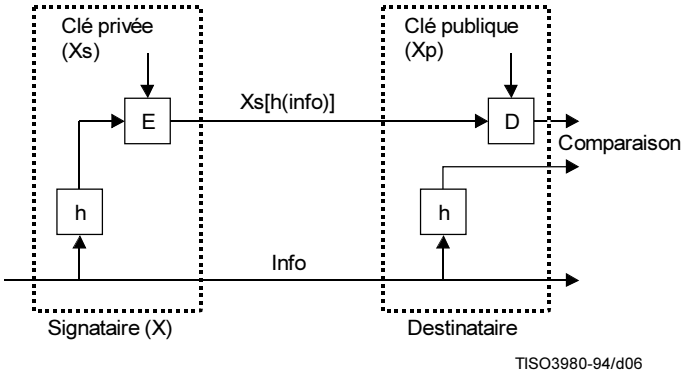


Figure 6 – Signatures numériques

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 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 hachage à une voie à l'information;
- 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 hachage unique à une voie. Il est prévu que ce cadre s'appliquera à toute fonction 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 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 l'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  --  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 le destinataire des données n'entrent pas dans le cadre de la présente Spécification d'Annuaire.

Au cas où une signature doit être ajoutée à un type de données, l'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 }}
```

Au cas où seule la signature est exigée, la macro 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.

```
SIGNATURE { OfSignature }      ::=   SEQUENCE {
    algorithmIdentifier              AlgorithmIdentifier,
    encrypted                        ENCRYPTED { HASHED { OfSignature }}}
```

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 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 leur valeur d'étiquette;

e) les éléments d'un type d'ensemble seront codés par ordre ascendant de leur valeur d'octet;

f) si la valeur d'un type booléen est vraie, 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.

10 Procédures d'authentification poussée

10.1 Vue d'ensemble

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 normes 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 en 10.2, fait intervenir un transfert d'information unique d'un utilisateur (A) destiné à un autre utilisateur (B), et établit:

   – l'identité de A, et que le jeton d'authentification a été effectivement produit par A;

   – l'identité de B, et 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 en 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 en 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 poussée, A doit obtenir la clé publique de B et l'itinéraire retour de certification de B en A avant tout échange d'information. Ceci 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 articles suivants, s'applique uniquement au cas où 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 génère 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:

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

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

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

L'emploi de «encData» comme clé privée exige un soin particulier, par exemple 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 mise en œuvre 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.

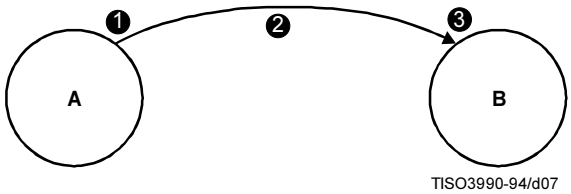


Figure 7 – Authentification à une voie

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 génère 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)].

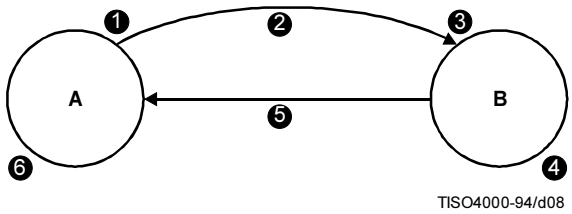


Figure 8 – Authentification à deux voies

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 r^A reçu est identique au 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 r^B reçu est identique au r^B émis par B.

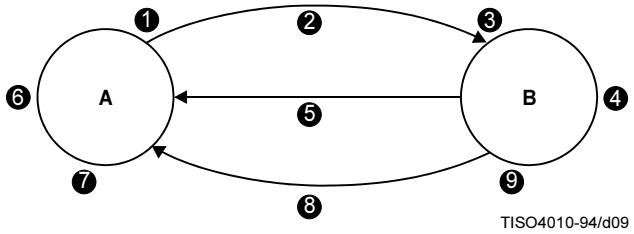


Figure 9 – Authentification à trois voies

11 Gestion des clés et des certificats

11.1 Génération 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 en dehors de l'objectif 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 il convient 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 génère 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 générée 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 générée par la CA. 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 nécessaires matérielles de sécurité. Cette méthode a l'avantage de ne pas nécessiter de transfert sur des données à la CA pour la certification.

Le système cryptographique en usage impose des contraintes (techniques) particulières à la génération 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, CAs, n'étant jamais connue si ce n'est de la CA isolée et matériellement sûre, le secret de la clé privée ne peut être percé que par une attaque contre la CA 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 appropriées de sécurité soient prises. A cet égard:

a) ce serait un grave manquement à la sécurité si la CA livrait un certificat à un utilisateur ayant une clé publique qui aurait été falsifiée;

b) si le moyen de générer des paires de clés donné au 11.1 c) est utilisé, aucun transfert sûr n'est nécessaire;

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 la CA en toute sécurité. Les méthodes «en ligne» sont parfois plus souples pour les opérations effectuées à distance entre l'utilisateur et la CA.

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 la CA 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, la CA 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 la CA 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 la CA 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 la CA, ou encore si l'on suppose que le certificat de la CA 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 la CA et un nouveau certificat sera mis à disposition, si besoin est. La CA peut alors informer le titulaire du certificat que celui-ci est annulé en appliquant une procédure différée.

– La CA 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 la CA 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
    ID                              id-at-certificateRevocationList }

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

CertificateList             ::=     SIGNED { SEQUENCE {
    signature                       AlgorithmIdentifier,
    issuer                          Name,
    thisUpdate                      UTCTime,
    nextUpdate                      UTCTime OPTIONAL,
    revokedCertificates             SEQUENCE OF SEQUENCE {
        userCertificate             CertificateSerialNumber,
        revocationDate              UTCTime } OPTIONAL }}
```

NOTES

1 La vérification de toute la liste des certificats est une question locale.

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

Annexe A

Cadre d'authentification en ASN.1

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

Cette annexe contient toutes les définitions de type, de valeur et de classe d'objet d'information ASN.1 qui figurent dans la présente Spécification d'Annuaire, sous la forme du module ASN.1 «**AuthenticationFramework**».

```
AuthenticationFramework {joint-iso-ccitt ds(5) module(1) authenticationFramework(7) 2}
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
                FROM UsefulDefinitions {joint-iso-ccitt ds(5) module(1) usefulDefinitions(0) 2}

        Name, ATTRIBUTE
                FROM InformationFramework informationFramework

        ub-user-password
                FROM UpperBounds upperBounds

        AuthenticationLevel
                FROM BasicAccessControl basicAccessControl

        UniqueIdentifier, octetStringMatch
                FROM SelectedAttributeTypes selectedAttributeTypes        ;

-- types --

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
        subjectUniqueIdentifier    [2]     IMPLICIT UniqueIdentifier OPTIONAL
                                                   -- si présent, la version doit être v2 -- }}

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

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 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     UTCTime,
    notAfter      UTCTime }

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

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

ForwardCertificationPath ::=    SEQUENCE OF CrossCertificates

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

CrossCertificates       ::=     SET OF Certificate

CertificateList         ::=     SIGNED { SEQUENCE {
    signature                     AlgorithmIdentifier,
    issuer                        Name,
    thisUpdate                    UTCTime,
    nextUpdate                    UTCTime OPTIONAL,
    revokedCertificates           SEQUENCE OF SEQUENCE {
        userCertificate                 CertificateSerialNumber,
        revocationDate                  UTCTime } OPTIONAL }}

CertificatePair         ::=     SEQUENCE {
    forward     [0]     Certificate OPTIONAL,
    reverse     [1]     Certificate OPTIONAL
                        -- au moins une des paires doit être présente -- }

-- types d'attributs --

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

userCertificate         ATTRIBUTE   ::=     {
    WITH SYNTAX         Certificate
    ID                  id-at-userCertificate }

cACertificate           ATTRIBUTE   ::=     {
    WITH SYNTAX         Certificate
    ID                  id-at-cAcertificate }

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

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

crossCertificatePair ATTRIBUTE  ::=     {
    WITH SYNTAX         CertificatePair
    ID                  id-at-crossCertificatePair }

-- classes d'objets d'information --

ALGORITHM ::=   TYPE-IDENTIFIER

-- types paramétrés --

HASHED { ToBeHashed }           ::=     OCTET STRING ( CONSTRAINED BY {
    -- doit résulter de l'application d'une procédure de hachage --
    -- aux octets codés selon les règles de codage de base (voir 8.7) --
    -- d'une valeur de -- ToBeHashed })
```

```
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 -- ToBeEnciphered })

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

SIGNATURE { OfSignature }      ::=     SEQUENCE {
        algorithmIdentifier    AlgorithmIdentifier,
        encrypted              ENCRYPTED { HASHED { OfSignature }}}

-- assignation d'identificateurs d'objets --

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}

END
```

Annexe B

Exigences de sécurité[1)]

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

De nombreuses applications OSI, des services définis par le CCITT et des services non définis par le CCITT comportent des exigences de sécurité. Ces exigences 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 pour mauvaise utilisation;

b) *fausse 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 pour la sécurité 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 itinéraire 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* – L'observation de l'information relative à une communication entre utilisateurs (c'est-à-dire absence/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.

---

1) Pour plus d'information, voir 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 poussée – S'appuie sur l'utilisation des techniques cryptographiques pour protéger l'échange de validation d'information. Dans le cadre d'authentification, l'authentification poussée 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 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 bullet («•») 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		•		
Fausse 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. Ceci 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[2)]

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

Dans les systèmes cryptographiques classiques, la clé utilisée par l'expéditeur d'un message secret pour effectuer le chiffrement est la même que celle qui est utilisée par le destinataire légitime pour effectuer le déchiffrement.

Dans les systèmes cryptographiques à clé publique (PKCS), 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. (Ceci 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.

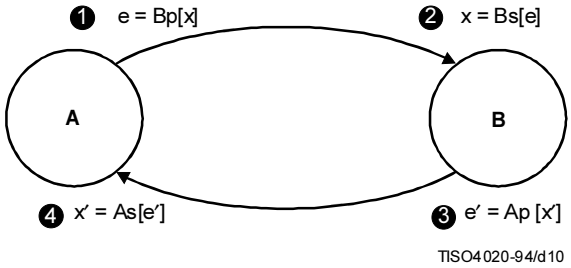


Figure C.1 – Utilisation d'un PKCS pour l'échange d'information secrète

L'utilisateur A possède une clé publique Ap et une clé privée As et 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 l'information secrète l'un avec l'autre 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. Ceci 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':

$$x' = As[e'], \text{ ou } x' = As[Ap[x']]$$

---

2) 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).

Par ce moyen, A et B ont échangé l'information secrète x et x'. Cette information ne peut être obtenue par personne d'autre que A et B du moment que leurs clés privées ne sont pas révélées.

Alors qu'un tel échange transfère l'information secrète entre deux parties, il peut aussi servir à vérifier leurs identités. Plus précisément, A et B sont respectivement identifiés par la possession de leurs clés privées de déchiffrement As et Bs. A peut déterminer si B est en possession de la clé privée de déchiffrement Bs par l'obtention de la partie x de son message envoyé en retour dans le message x' de B. Cela indique à A que la communication est établie avec le possesseur de Bs. B peut de même vérifier l'identité de A.

Certains PKCS possèdent cette 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 pourrait avoir été expédié uniquement par X, soit lisible par tout autre utilisateur (qui soit en possession de 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 RSA[3] cryptographique de clé publique[4]

(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 cette 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

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

---

[3] Le système de cryptographie spécifié dans la présente annexe, inventé par R. L. Rivest, A. Shamir et L. Adleman, est bien connu sous le sigle: «RSA».

[4] 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, (février 1978) 120-126.

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, (14 octobre 1982) 905-907.

D.4 Description

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

$$Y = X^e \bmod n \quad \text{avec} \quad 0 \le X < n$$

$$X = Y^d \bmod n \quad \text{avec} \quad 0 \le Y < n$$

qui peuvent être satisfaites par exemple pour:

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

$$ed \bmod (p-1)(q-1) = 1.$$

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 λ). 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 Exigences 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 Génération de clés

La sécurité du système RCA 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 4 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 génération 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 4) 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-ième 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.

NOTES

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.

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 hachage

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

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

E.1 Exigences pour les fonctions hachage

Pour utiliser une fonction 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 hachage à haut niveau de sécurité devra satisfaire aux exigences 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 est réduit à ce résultat;

b) la fonction 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 poussée

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

La méthode d'authentification poussée 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 poussée.

De plus, il existe une gamme de dangers potentiels qui tiennent à la méthode d'authentification poussée elle-même, à savoir:

Compromission de la clé privée d'utilisateur – Un des principes de base de l'authentification poussée 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 poussée 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 poussée 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 poussée 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)

Cette 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) 2}
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) 2}
        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 J

Modifications et Corrigendums

(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:

– Modification 1 pour le contrôle d'accès

Cette édition de la présente Spécification d'Annuaire comprend les corrigendums concernant les défauts signalés dans les Rapports de défaut suivants:

– Corrigendum technique 1 (reprenant les Rapports de défaut 009, 015, 016, 019, 031).