



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

CCITT

COMITÉ CONSULTATIF
INTERNATIONAL
TÉLÉGRAPHIQUE ET TÉLÉPHONIQUE

X.509

(11/1988)

SÉRIE X: RÉSEAUX DE COMMUNICATIONS DE
DONNÉES: ANNUAIRE

ANNUAIRE – CADRE D'AUTHENTIFICATION

Réédition de la Recommandation X.509 du CCITT publiée
dans le Livre Bleu, Fascicule VIII.8 (1988)

NOTES

1 La Recommandation X.509 du CCITT a été publiée dans le fascicule VIII.8 du Livre Bleu. Ce fichier est un extrait du Livre Bleu. La présentation peut en être légèrement différente, mais le contenu est identique à celui du Livre Bleu et les conditions en matière de droits d'auteur restent inchangées (voir plus loin).

2 Dans la présente Recommandation, le terme «Administration» désigne indifféremment une administration de télécommunication ou une exploitation reconnue.

ANNUAIRE – CADRE D'AUTHENTIFICATION ¹⁾

(Melbourne, 1988)

SOMMAIRE

0 *Introduction*

1 *Objectif et domaine d'application*

2 *Références*

3 *Définitions*

4 *Notation et abréviations*

SECTION 1 – *Authentification simple*

5 *Procédure d'authentification simple*

SECTION 2 – *Authentification poussée*

6 *Bases de l'authentification poussée*

7 *Obtention d'une clé publique d'utilisateur*

8 *Signatures numériques*

9 *Procédures d'authentification poussée*

10 *Gestion des clés et des certificats*

Annexe A – Exigences de sécurité

Annexe B – Introduction à la cryptographie de clé publique

Annexe C – Système cryptographique de clé publique RSA

Annexe D – Fonctions hachage

Annexe E – Dangers contre lesquels la protection est assurée par les services de sécurité

Annexe F – Confidentialité des données

Annexe G – Cadre d'authentification en ASN.1

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

¹⁾ La Recommandation X.509 et l'ISO 9594-8, Systèmes de traitement de l'information – Interconnexion des systèmes ouverts – L'annuaire – Cadre d'authentification, ont été élaborées en étroite collaboration et sont alignées du point de vue technique.

0 Introduction

0.1 La présente Recommandation, de même que les autres de la série, a été élaborée pour faciliter l'interconnexion des systèmes de traitement de l'information en vue d'assurer la prestation des services d'annuaire. L'ensemble de tous ces systèmes de même que l'information d'annuaire qu'ils conservent, peuvent être considérés comme un tout intégré appelé l'annuaire. Cette information détenue par l'*annuaire* connue généralement sous le nom de base d'information d'annuaire (DIB) est en général utilisée pour faciliter la communication entre, avec ou au sujet d'objets tels que des entités d'application OSI, des personnes, des terminaux et des listes de diffusion.

0.2 L'annuaire joue un rôle important dans l'interconnexion des systèmes ouverts, dont le but est de permettre sur une base de concertation technique minimale – mises à part les normes d'interconnexion elles-mêmes – l'interconnexion des systèmes de traitement de l'information:

- provenant de fabricants différents;
- placés sous des gestions différentes;
- de niveaux de complexité différents; et
- d'âge différent.

0.3 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 A 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.

0.4 La présente Recommandation 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. 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.

1 Objectif et domaine d'application

1.1 La présente Recommandation:

- 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.

1.2 La présente Recommandation 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.

1.3 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 à la ou aux normes de sécurité définissant cette application de définir leur propre politique de sécurité.

1.4 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 Recommandation X.519.

1.5 La méthode d'authentification poussée spécifiée dans la présente Recommandation 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 B.

1.6 En général, le cadre d'authentification ne dépend pas de l'utilisation d'un algorithme particulier pour autant qu'il possède les propriétés décrites au § 6.1. Il est possible en 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 C.

1.7 De même, deux utilisateurs qui désirent s'authentifier doivent utiliser la même fonction hachage [voir le § 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 D.

2 Références

2.1 ISO 7498-2: Systèmes de traitement de l'information – Interconnexion des systèmes ouverts – Architecture de sécurité.

3 Définitions

3.1 La présente Recommandation utilise les termes généraux relatifs à la sécurité définis dans la partie 2 du modèle de référence OSI sur la sécurité:

- a) *asymétrique* (chiffrement);
- b) *échange d'authentifications*;
- c) *information d'authentification*;
- d) *confidentialité*;
- e) *accréditation*;
- f) *cryptographie*;
- g) *authentification de l'origine des données*;
- h) *déchiffrement*;
- i) *chiffrement*;
- j) *clé*;
- k) *mot de passe*;
- l) *authentification d'entité homologue*;
- m) *symétrique* (chiffrement).

3.2 Les termes suivants, utilisés dans la présente Recommandation, sont définis dans la Recommandation X.501:

- a) *attribut*;
- b) *base de données d'annuaire*;
- c) *arbre d'information d'annuaire*;
- d) *nom spécifique*;
- e) *entrée*;
- f) *objet*;
- g) *racine*.

- 3.3 Les termes spécifiques suivants sont définis et utilisés dans la Recommandation:
- a) *jeton d'authentification (jeton)*: Information acheminée au cours d'un échange d'authentifications, qui peut être utilisée à authentifier son expéditeur;
 - b) *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;
 - c) *autorité de certification*: Autorité chargée par un ou plusieurs utilisateurs de créer et d'attribuer leur clé publique et leur certificat. Cette autorité peut, facultativement, créer les clés d'utilisateur;
 - d) *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;
 - e) *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;
 - f) *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;
 - g) *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;
 - h) *clé publique*: (dans un système cryptographique de clé publique) Clé d'une paire de clés d'utilisateur qui est publiquement connue;
 - i) *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;
 - j) *authentification simple*: Authentification obtenue au moyen de simples arrangements de mot de passe;
 - k) *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;
 - l) *authentification poussée*: Authentification obtenue au moyen d'accréditations déterminées par cryptographie;
 - m) *confiance*: Généralement, on peut dire qu'une entité se fie à une deuxième entité lorsqu'elle (la première entité) fait 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;
 - n) *numéro de série d'utilisateur*: Valeur entière, unique pour la CA qui l'émet et associée sans ambiguïté au certificat émis par cette CA.

4 Notation et abréviations

4.1 La notation employée dans la Recommandation est définie dans le tableau 1/X.509.

Remarque – Dans l'introduction des notations, les symboles X , X_1 , X_2 , etc. apparaissent à la place des noms d'utilisateur tandis que le symbole I est mis à la place d'une information arbitraire.

4.2 Les abréviations suivantes sont utilisées dans la Recommandation.

CA	Autorité de certification ("Certification Authority")
DIB	Base de données d'annuaire ("Directory Information Base")
DIT	Arbre d'information d'annuaire ("Directory Information Tree")
PKCS	Système cryptographique à clé publique ("Public key cryptosystem").

TABLEAU 1/X.509

Notation

NOTATION	SIGNIFICATION
X_p	Clé publique d'un utilisateur X.
X_s	Clé secrète de X.
$X_p[I]$	Chiffrement d'une certaine information, I, au moyen de la clé publique de X.
$X_s[I]$	Chiffrement de [I] au moyen de la clé secrète 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))).
$X_1 \ll X_2 \gg$	Certificat de l'utilisateur X_2 émis par l'autorité de certification X_1 .
$X_1 \ll X_2 \gg X_2 \ll X_3 \gg$	Chaine de certificats (pouvant être de longueur arbitraire) où chaque article est le certificat pour l'autorité de certification qui produit le texte. Il est fonctionnellement équivalent au certificat suivant $X_1 \ll X_{n+1} \gg$. Par exemple, la possession $A \ll B \gg B \ll C \gg$ fournit la même capacité que $A \ll C \gg$, à savoir la possibilité de découvrir C_p de A_p donné.
$X_{1p} \cdot X_1 \ll X_2 \gg$	Opération de dévoilement d'un certificat (ou d'une chaine 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: $A_p \cdot A \ll B \gg B \ll C \gg$ indique l'opération de l'utilisation de la clé publique de A pour obtenir la clé publique B_p de B, à partir de son certificat, suivi de l'utilisation de B_p pour dévoiler le certificat de C. Le résultat de l'opération est la clé publique C_p de C.
$A \rightarrow B$	Itinéraire de certification de A en B composé d'une chaine de certificats, débutant avec: $CA(A) \ll CA^2(A)$ et finissant avec $CA(B) \ll B \gg$.

5 Procédure d'authentification simple

5.1 L'authentification simple vise à fournir une autorisation locale reposant sur un nom spécifique 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 spécifique de l'utilisateur et du mot de passe (facultatif) en clair (sans protection) au destinataire pour évaluation;
- b) transfert du nom spécifique 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.

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

Remarque 2 – La signalisation des procédures de protection des mots de passe pourra faire l'objet d'un complément à la présente Recommandation.

5.2 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 spécifique 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 oeuvre.

5.3 La figure 1/X.509 montre la procédure générale à appliquer pour obtenir une authentification simple.

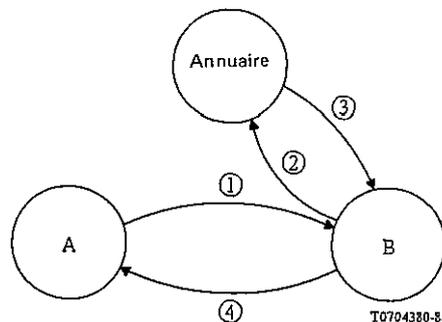


FIGURE 1/X.509

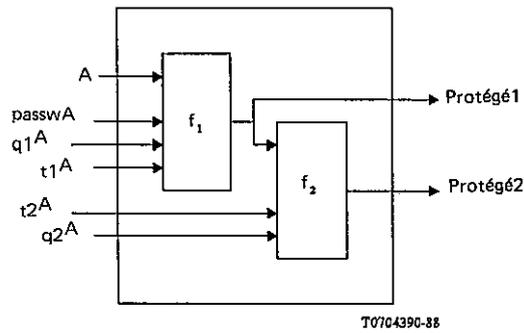
Procédure d'authentification simple sans protection

5.3.1 Les étapes de cette procédure sont les suivantes:

- 1) un utilisateur expéditeur A envoie son nom spécifique et son mot de passe à un utilisateur destinataire B;
- 2) B envoie le nom spécifique visé et le mot de passe de A à l'annuaire, où le mot de passe est comparé avec celui qui détient l'attribut **mot de passe de l'utilisateur** 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.

5.3.2 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 spécifique et le mot de passe.

5.4 La figure 2/X.509 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 numéros aléatoires sont facultatifs et dépendent d'accords bilatéraux.



A = nom spécifique de l'utilisateur
 t^A = indications horaires
 passwdA = mot de passe A
 q^A = numéros aléatoires avec inclusion d'un compteur (facultatif)

FIGURE 2/X.509

Authentification simple protégée

5.4.1 La figure 3/X.509 montre la procédure d'authentification simple protégée.

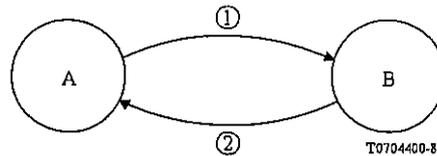


FIGURE 3/X.509

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/X.509, où l'indication horaire et (ou) le numéro aléatoire (s'il est utilisé) ont pour objet de réduire à un minimum les répétitions et de cacher le mot de passe.

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

$$\text{Protected1} = f_1(t1^A, q1^A, A, \text{passwdA}).$$

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 de l'indication horaire, du nom spécifique et, facultativement, d'une indication horaire additionnelle et/ou d'un numéro 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.

5.4.2 La procédure du § 5.4.1 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/X.509. La protection supplémentaire prend la forme:

$$\text{Protected2} = f_2(t2^A, q2^A, \text{Protected1}).$$

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 le compare (pour en contrôler l'égalité) avec celle de Protected2 (le principe étant le même que pour l'étape 1 du § 5.4.1).

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

Remarque – Les procédures définies dans le présent paragraphe sont spécifiées en fonction de A et B. Appliqué à l'annuaire (spécifié dans les Recommandations X.511 et X.518), A pourrait être un DUA lié à un DSA, B ou A pourrait être un DSA lié à un autre DSA, B.

5.5 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 ATTRIBUTE-SYNTAX
OCTET STRING (SIZE (0..ub-user-password))
MATCHES FOR EQUALITY**

5.6 La macro ASN.1 suivante peut être utilisée pour définir le type de données provenant de l'application d'une fonction à une voie à un autre type de données donné.

PROTECTED MACRO ::= SIGNATURE

SECTION 2 – *Authentication poussée*

6 Bases de l'authentification poussée

6.1 La façon d'aborder une authentification poussée au sens de la présente Recommandation 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 secrète et l'autre publique, plutôt qu'une seule clé comme dans les systèmes cryptographiques classiques. L'annexe B 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 secrète 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 secrète qui a été utilisée. Autrement dit, $X_p \cdot X_s = Y_s \cdot Y_p$ si X_p/X_s sont des fonctions de chiffrement/déchiffrement utilisant les clés publique/secrète de l'utilisateur X.

Remarque – 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 oeuvre sans grande modification de la présente Recommandation.

6.2 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 C.

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

6.4 Chaque utilisateur est identifié par la possession de sa clé secrète. Un deuxième utilisateur est capable de déterminer si un partenaire d'une communication est en possession de la clé secrète 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 secrète demeure confidentielle au niveau de l'utilisateur.

6.5 Pour qu'un utilisateur puisse déterminer si un partenaire de communication est en possession de la clé secrète 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; le § 7 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.

7 Obtention d'une clé publique d'usager

7.1 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 au § 7.2, 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.

Remarque – Bien que les CA soient définies sans ambiguïté par un nom spécifique dans le DIT, cela n'implique nullement une relation entre l'organisation des CA et le DIT.

7.2 Une autorité de certification produit les certificats de l'utilisateur en signant (voir le § 8) un recueil d'informations comprenant le nom spécifique d'utilisateur et la clé publique. Plus précisément, le certificat d'un utilisateur ayant A pour nom spécifique produit par l'autorité de certification CA prend la forme:

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

où SN est le numéro de série du certificat, AI est l'identificateur de l'algorithme servant à signer le certificat: T^A indique la période de validité du certificat et comprend deux dates, correspondant au début et à la fin de cette validité. Comme on suppose que T^A est modifié par période d'au moins 24 heures, on prévoit que les systèmes utiliseront le temps universel coordonné comme base de temps de référence. Tout utilisateur ayant connaissance de CAP peut vérifier la validité de la signature du certificat. Le type de données ASN.1 suivant peut être utilisé pour représenter les certificats.

```

Certificate ::= SIGNED SEQUENCE{
    version [0]Version DEFAULT 1988,
    serialNumber SerialNumber,
    signature AlgorithmIdentifier
    issuer Name
    validity Validity,
    subject Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo}

Version ::= INTEGER { 1988(0)}
SerialNumber ::= INTEGER

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

SubjectPublicKeyInfo ::=
    SEQUENCE{
        algorithm AlgorithmIdentifier
        subjectKey BIT STRING}

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

```

7.3 L'entrée d'annuaire de chaque utilisateur, A, qui participe à une authentification poussée contient le certificat 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.

7.4 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.

7.5 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 (indiqué pour $A \rightarrow B$):

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

Un itinéraire de certification forme logiquement une chaîne continue de points de confiance dans l'arbre d'information de l'annuaire entre deux utilisateurs qui désirent s'authentifier. La méthode précise utilisée par les utilisateurs A et B pour obtenir les itinéraires de certification $A \rightarrow B$ et $B \rightarrow A$ 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 son CA supérieure.

7.6 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 de la présente Recommandation, leur spécification est la suivante:

```

UserCertificate ::= ATTRIBUTE
WITH ATTRIBUTE-SYNTAX Certificate
CACertificate ::= ATTRIBUTE
WITH ATTRIBUTE-SYNTAX Certificate
CrossCertificatePair ::= ATTRIBUTE
WITH ATTRIBUTE-SYNTAX CertificatePair
CertificatePair ::=
SEQUENCE{
forward [o] Certificate OPTIONAL
reverse [1] Certificate OPTIONAL
-- at least one must be present --}

```

Un utilisateur peut obtenir une 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.

7.7 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.

7.8 (Exemple). La figure 4/X.509 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 secrète.

7.8.1 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:

$$X\langle\langle W \rangle\rangle, W\langle\langle V \rangle\rangle, V\langle\langle Y \rangle\rangle, Y\langle\langle Z \rangle\rangle, Z\langle\langle B \rangle\rangle.$$

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:

$$B_p = X_p \cdot X\langle\langle W \rangle\rangle W\langle\langle V \rangle\rangle V\langle\langle Y \rangle\rangle Y\langle\langle Z \rangle\rangle Z\langle\langle B \rangle\rangle.$$

En général, A doit également acquérir les certificats suivants provenant de l'annuaire pour établir l'itinéraire retour de certification de B vers A:

$$Z\langle\langle Y \rangle\rangle, Y\langle\langle V \rangle\rangle, V\langle\langle W \rangle\rangle, W\langle\langle X \rangle\rangle, X\langle\langle A \rangle\rangle.$$

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:

$$A_p = Z_p \cdot Z\langle\langle Y \rangle\rangle Y\langle\langle V \rangle\rangle V\langle\langle W \rangle\rangle W\langle\langle X \rangle\rangle X\langle\langle A \rangle\rangle.$$

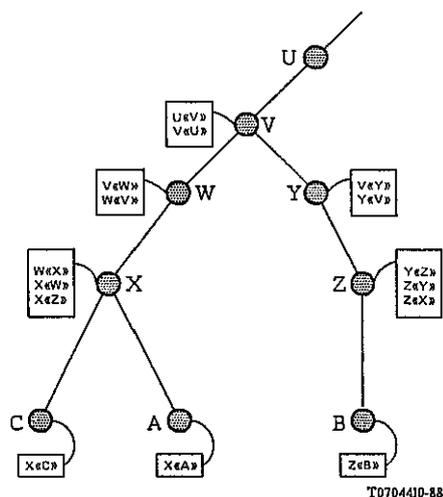


FIGURE 4/X.509

Hiérarchie des CA – Exemple fictif

7.8.2 En appliquant les optimisations du § 7.7:

- a) Considérons A et C par exemple: tous les deux ont connaissance de X_p , 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:

$$C_p = X_p \cdot X\langle\langle C \rangle\rangle$$

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

$$A_p = X_p \cdot X\langle\langle A \rangle\rangle.$$

- b) En admettant que A ait connaissance de la même façon de $W\langle\langle X \rangle\rangle$, W_p , $V\langle\langle W \rangle\rangle$, V_p , $U\langle\langle V \rangle\rangle$, U_p , etc., l'information que A doit obtenir de l'annuaire pour former l'itinéraire de certification, se réduit à:

$$V\langle\langle Y \rangle\rangle, Y\langle\langle Z \rangle\rangle, Z\langle\langle B \rangle\rangle$$

et l'information que A doit obtenir de l'annuaire pour former l'itinéraire retour de certification, se réduit à:

$$Z\langle\langle Y \rangle\rangle, Y\langle\langle V \rangle\rangle.$$

- 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) cidessus], de $V\langle\langle Y \rangle\rangle$, $Y\langle\langle V \rangle\rangle$, $Y\langle\langle Z \rangle\rangle$ et $Z\langle\langle Y \rangle\rangle$. Pour communiquer avec B, il lui suffit donc d'obtenir seulement $Z\langle\langle B \rangle\rangle$ de l'annuaire.
- d) En admettant que les utilisateurs certifiés par X et Z soient fréquemment en communication, $X\langle\langle Z \rangle\rangle$ sera conservé dans l'entrée d'annuaire pour X et réciproquement (ceci est représenté dans la figure 4/X.509). Si A désire procéder aux authentifications avec B, il lui suffit d'obtenir:

$$X\langle\langle Z \rangle\rangle, Z\langle\langle B \rangle\rangle$$

pour former l'itinéraire de certification, et

$$Z\langle\langle X \rangle\rangle$$

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:

$$C_p = X_p \cdot X\langle\langle C \rangle\rangle$$

et

$$A_p = X_p \cdot X\langle\langle A \rangle\rangle.$$

7.8.3 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/X.509, 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 $U\langle\langle D \rangle\rangle$ et celle de l'utilisateur E contiendra le certificat $W\langle\langle E \rangle\rangle$.

Soit V une CA avec laquelle U et W ont auparavant échangé des clés publiques en confiance. Il s'ensuit que des certificats $U\langle\langle V \rangle\rangle$, $V\langle\langle U \rangle\rangle$, $W\langle\langle V \rangle\rangle$ et $V\langle\langle W \rangle\rangle$ ont été produits et mémorisés dans l'annuaire. En supposant que $U\langle\langle V \rangle\rangle$ et $W\langle\langle V \rangle\rangle$ sont stockés dans l'entrée de V, $V\langle\langle U \rangle\rangle$ est stocké dans l'entrée de U et $V\langle\langle W \rangle\rangle$ 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 noeuds 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 $U\langle\langle V \rangle\rangle$, $V\langle\langle W \rangle\rangle$, $W\langle\langle E \rangle\rangle$. Une fois que cet itinéraire a été découvert, l'itinéraire inverse $W\langle\langle V \rangle\rangle$, $V\langle\langle U \rangle\rangle$, $U\langle\langle D \rangle\rangle$ peut aussi être constitué.

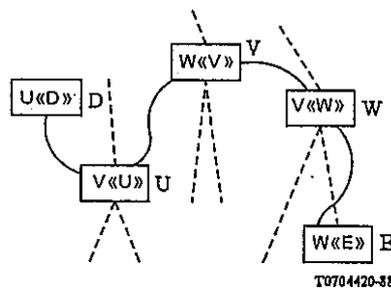


FIGURE 5/X.509

Exemple d'itinéraire de certification non hiérarchique

8 Signatures numériques

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

8.1 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 secrète du signataire (voir figure 6/X.509) ainsi:

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

Remarque – Le chiffrement utilisant la clé secrète 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.

8.2 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.

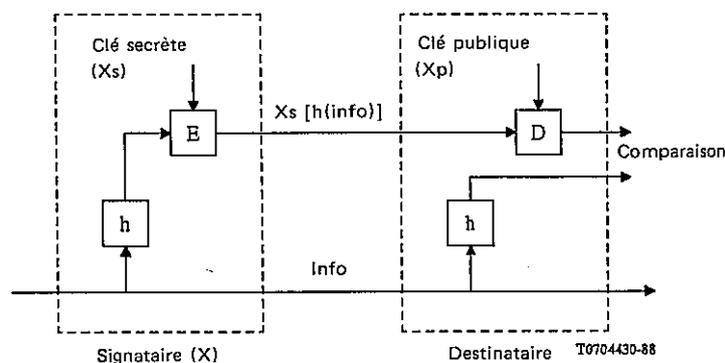


FIGURE 6/X.509

Signatures numériques

8.3 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 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. Une fonction hachage vraisemblablement appelée à être largement utilisée est spécifiée dans l'annexe D.

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

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

```

ENCRYPTED MACRO ::=
BEGIN

TYPE NOTATION ::= type (ToBeEnciphered)
VALUE NOTATION ::= value (VALUE BIT STRING)
END
    
```

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) de la valeur du **type à chiffrer** et on applique une procédure de chiffrement à ces octets.

Remarque 1 – 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.

Remarque 2 – La procédure de chiffrement doit prendre comme entrée une chaîne d'octets et engendrer une seule chaîne de bits en tant que résultat.

Remarque 3 – 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 Recommandation.

8.5 Au cas où une signature doit être ajoutée à un type de données, 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.

```

SIGNED MACRO ::=
BEGIN

TYPE NOTATION ::= type (ToBeSigned)

VALUE NOTATION ::= value (VALUE
    SEQUENCE{
        ToBeSigned,
        AlgorithmIdentifier,
        -- of the algorithm used to compute
        -- the signature
        ENCRYPTED OCTET STRING
        -- where the octet string is the result
        -- of the hashing of the value of
        -- 'ToBeSigned' --}
    )
END -- of SIGNED.

```

8.6 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 MACRO ::=
BEGIN

TYPE NOTATION ::= type (OfSignature)

VALUE NOTATION ::= value (VALUE
    SEQUENCE{
        AlgorithmIdentifier,
        -- of the algorithm used to compute
        -- the signature
        ENCRYPTED OCTET STRING
        -- where the octet string is a function (e.g. a
        -- compressed or hashed version) of the
        -- value 'OfSignature', which may include
        -- the identifier of the algorithm used to
        -- compute the signature --}
    )
END -- of SIGNATURE.

```

8.7 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 la Recommandation X.209, 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.

9 Procédures d'authentification poussée

9.1 Présentation générale

9.1.1 La manière d'aborder fondamentalement l'authentification a été exposée cidessus, à savoir la confirmation de l'identité en faisant apparaître la possession d'une clef secrète. 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. Le § 9 contient la description de trois procédures particulières d'authentification qui peuvent s'avérer utiles au travers d'une gamme d'applications.

Remarque – La présente Recommandation ne spécifie pas les procédures en précisant les détails nécessaires à leur mise en oeuvre. 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.

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

- a) Une authentification à une voie, décrite au § 9.2, 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 au § 9.3, fait intervenir en plus une réponse de B en A. Elle établit en plus:
 - que le jeton d'authentification produit dans la réponse l'a effectivement été par B et qu'il est destiné à être envoyé en A;
 - l'intégrité et l'originalité du jeton d'authentification envoyé dans la réponse;
 - à titre d'option, le secret mutuel d'une partie des jetons.
- c) Une authentification à trois voies, décrite au § 9.4, 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 clef 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 au § 7, mais cet accès, s'il existe, n'est pas mentionné de nouveau dans la description des procédures ci-dessous.

La vérification des dates et heures citée dans les paragraphes suivants, s'applique uniquement 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.

9.2 *Authentification à une voie*

Déroulement des opérations comme représenté à la figure 7/X.509.

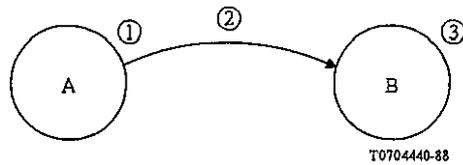


FIGURE 7/X.509

Authentification à une voie

- 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 secrète (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 secrète 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 A_p à partir de $B \rightarrow A$, 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 oeuvre 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 .

9.3 *Authentification à deux voies*

Déroulement des opérations comme représenté à la figure 8/X.509.

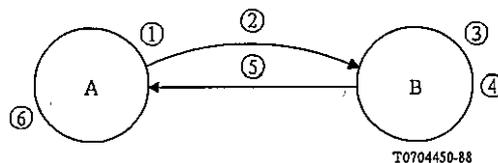


FIGURE 8/X.509

Authentification à deux voies

- 1) Comme pour le § 9.2.
- 2) Comme pour le § 9.2.
- 3) Comme pour le § 9.2.
- 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:

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

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

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

L'emploi de "encData" comme clef secrète implique un choix approprié, par exemple pour obtenir une clef 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 au § 9.2 l'étape 3) e)].

9.4 Authentification à trois voies

Déroulement des opérations comme représenté à la figure 9/X.509.

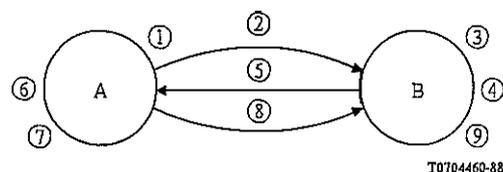


FIGURE 9/X.509

Authentification à trois voies

- 1) Comme pour le § 9.3.
- 2) Comme pour le § 9.3. L'indication horaire t^A peut avoir la valeur zéro.
- 3) Comme pour le § 9.3, excepté qu'il n'est pas nécessaire de vérifier la date ni l'heure.
- 4) Comme pour le § 9.3.
- 5) Comme pour le § 9.3. L'indication horaire t^B peut avoir la valeur zéro.
- 6) Comme pour le § 9.3 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:
 $A \{r^B\}$.
- 9) B effectue les opérations suivantes:
 - a) vérifie la signature et ainsi vérifie l'intégrité de l'information signée;
 - b) vérifie que le r^B reçu est identique au r^B émis par B.

10 Gestion des clefs et des certificats

10.1 Génération de paires de clefs

10.1.1 La politique de gestion de sécurité totale d'une mise en oeuvre définira le cycle de vie des paires de clefs et se trouve donc en dehors de l'objectif du cadre d'authentification. Cependant, il l'utilisateur auquel ces clefs appartiennent.

Il n'est pas facile pour un utilisateur qui est un être humain de se souvenir des données des clefs, 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 clefs secrète et (facultativement) publique de l'utilisateur, le certificat d'utilisateur et une copie de la clef publique de l'autorité de certification. L'emploi de cette carte devrait comporter une sécurité complémentaire, par exemple par l'utilisation d'un PIN ("Personal Identification Number": Numéro d'identification personnel), 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 Recommandation.

10.1.2 Il y a trois façons de produire une paire de clefs comme indiqué aux § 10.1.2.1 à 10.1.2.3.

10.1.2.1 L'utilisateur génère sa propre paire de clefs. Cette méthode a l'avantage de ne jamais livrer une clef secrète 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 C.

10.1.2.2 La paire de clefs est générée par un tiers. Celui-ci doit livrer la clef secrète à l'utilisateur de manière matérielle et sûre, puis détruire résolument toute information relative à la création de la paire de clefs, ainsi que les clefs 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.

10.1.2.3 La paire de clefs est générée par la CA. C'est un cas particulier du § 10.1.2.2 et les mêmes considérations s'appliquent dans ce cas.

Remarque – 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.

10.1.2.4 Le système cryptographique en usage impose des contraintes (techniques) particulières à la génération des clefs.

10.2 Gestion des certificats

10.2.1 Un certificat associe la clef publique et le nom spécifique 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.

10.2.2 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 clef secrète 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 clef secrète ne peut être percé que par une attaque contre la CA elle-même, ce qui rend une compromission improbable.

10.2.3 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 clef publique qui aurait été falsifiée;
- b) si le moyen de générer des paires de clefs donné au § 10.1.2.3 est utilisé, aucun transfert sûr n'est nécessaire;
- c) en cas d'emploi du moyen de production des paires de clefs du § 10.1.2.1 ou du § 10.1.2.2, l'utilisateur peut recourir à diverses méthodes (directes ou différées) pour communiquer sa clef 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.

10.2.4 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.

10.2.5 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 § 10.2.5.1 et 10.2.5.2 en montrent les différents aspects.

10.2.5.1 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.

10.2.5.2 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.

10.2.6 Les certificats peuvent être annulés avant d'être périmés, par exemple si l'on suppose que la clef secrète 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 divers aspects en sont présentés aux § 10.2.6.1 à 10.2.6.4.

10.2.6.1 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.

10.2.6.2 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 clefs secrètes dont elle a connaissance, qu'elle a certifiés.

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

10.2.6.3 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.

10.2.6.4 Les listes d'annulation ("listes noires") sont conservées dans les entrées comme des attributs des types "**Liste d'annulation de certificats**" et "**Liste d'annulation d'autorité**". Ces attributs peuvent être exploités selon les mêmes opérations que les autres attributs. Ces types d'attribut sont définis ainsi:

```
CertificateRevocationList ::= ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX CertificateList  
  
AuthorityRevocationList ::= ATTRIBUTE  
WITH ATTRIBUTE-SYNTAX CertificateList  
  
CertificateList ::= SIGNED SEQUENCE{  
signature AlgorithmIdentifier,  
issuer Name,  
lastUpdate UTCTime,  
revokedCertificates  
    SIGNED SEQUENCE OF SEQUENCE{  
        signature AlgorithmIdentifier,  
        issuer Name, CertificateSerialNumber subject,  
        revocationDate UTCTime}  
    OPTIONAL}
```

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

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

ANNEXE A
(à la Recommandation X.509)

Exigences de sécurité

La présente annexe ne fait pas partie intégrante de la présente Recommandation.

[On peut trouver le texte additionnel se rapportant à ce sujet dans le document ISO 7498 – Systèmes de traitement de l'information – Modèle de référence OSI – partie 2, Architecture de sécurité.]

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.

A.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) *redéfilingement*: 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;

Remarque – 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;

Remarque – 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.).

Remarque – 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.

A.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 § A.3 de la présente annexe.

- 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.

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.

A.3 Mécanismes de sécurité

Les mécanismes de sécurité indiqués dans ce paragraphe assurent les services de sécurité décrits au § A.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 nonrejet (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 secrète 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.

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

Le tableau figurant à la fin de la présente annexe indique les dangers susceptibles de porter atteinte à la sécurité contre lesquels chaque service de sécurité peut assurer la protection. La présence d'un astérisque (*) indique qu'un service donné de sécurité assure la protection contre un danger donné.

A.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.

SERVICES

DANGERS	Authentification d'entité	Secret des données	Intégrité des données	Non-rejet
Interception d'identité	*			
	(si nécessaire)			
Interception de données		*		
Fausse identité	*			
Redéfilement	*		*	*
	(identité)		(données)	
Manipulation			*	*
Rejet				*

ANNEXE B

(à la Recommandation X.509)

Introduction à la cryptographie de clef publique

La présente annexe ne fait pas partie intégrante de la présente Recommandation.

Dans les systèmes cryptographiques classiques, la clef 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 clef publique (PKCS), cependant, les clefs vont par paires, l'une des clefs étant utilisée pour le chiffrement et l'autre pour le déchiffrement. Chaque paire de clefs est associée à un utilisateur particulier X. L'une des clefs connue sous le nom de clef publique (X_p) est connue publiquement et peut être employée par n'importe quel utilisateur pour chiffrer les données. Seul X qui possède la clef secrète complémentaire (X_s) peut déchiffrer les données. (Ceci est représenté par la notation $D = X_s[X_p[D]]$.) Il est impossible de découvrir par un calcul la clef secrète à partir de la connaissance de la clef publique. Tout utilisateur peut ainsi communiquer un élément d'information que seul X peut découvrir en la chiffrant au moyen de X_p . Par extension, deux utilisateurs peuvent communiquer en secret en utilisant l'un et l'autre la clef publique pour chiffrer les données comme indiqué dans la figure B-1/X.509.

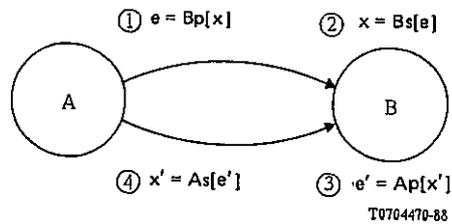


FIGURE B-1/X.509

Utilisation d'un PKCS pour l'échange d'information secrète

L'utilisateur A possède une clef publique A_p et une clef secrète A_s et l'utilisateur B possède un autre jeu de clefs B_p et B_s . A et B connaissent tous les deux les clefs publiques de l'un et de l'autre, mais ignorent la clef secrète 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 ciaprès (représentées dans la figure B-1/X.509):

- 1) A désire envoyer une certaine information secrète x à B. A chiffre donc x avec la clef de chiffrement de B et envoie l'information chiffrée e à B. Ceci est représenté par:

$$e = B_p[x].$$

- 2) B peut maintenant déchiffrer cette information chiffrée e pour obtenir l'information x au moyen de la clef secrète de déchiffrement B_s . Il convient de noter que B est le seul possesseur de B_s et, étant donné que cette clef ne peut jamais être découverte ou envoyée, il est impossible à une autre partie d'obtenir l'information x . La possession de B_s détermine l'identité de B. L'opération de déchiffrement est représentée par:

$$x = B_s[e] \text{ ou } x = B_s[B_p[x]].$$

- 3) B peut maintenant envoyer de même une certaine information secrète x' à A avec la clef de chiffrement A_p de A:

$$e' = A_p[x'].$$

- 4) A obtient x' par déchiffrement de e' :

$$x' = A_s[e'], \text{ ou } x' = A_s[A_p[x']].$$

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 clefs secrètes 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 clefs secrètes de déchiffrement A_s et B_s . A peut déterminer si B est en possession de la clef secrète de déchiffrement B_s 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 B_s . 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 $D = X_p[X_s[D]]$. Ainsi, un élément d'information qui pourrait avoir été expédié uniquement par X, soit lisible par tout autre utilisateur (qui soit en possession de X_p). 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 C.

Pour complément d'information, consulter:

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

ANNEXE C

(à la Recommandation X.509)

Système cryptographique de clef publique (RSA)

La présente annexe ne fait pas partie intégrante de la présente Recommandation.

Remarque – 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".

C.1 *Objectif et domaine d'application*

Un exposé complet sur le système RSA dépasse l'objectif de ce document. Toutefois, on trouvera ciaprès la description succincte de la méthode qui repose sur l'utilisation d'une élévation à une puissance d'un module.

C.2 *Références*

Pour un complément d'information, consulter:

1) Généralités

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

2) Production des clefs

GORDON, J. – Strong RSA Keys, *Electronics Letters*, 20, 5, 514-516.

3) Déchiffrement

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

C.3 *Définitions*

a) *clef publique*: Paire de paramètres qui se compose de l'exposant public et du module arithmétique;

Remarque – L'élément de données **subjectPublicKey** ASN.1, défini comme **BIT STRING** (voir l'annexe G) 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.

b) *clef secrète*: Paire de paramètres qui se compose de l'exposant secret et du module arithmétique.

C.4 *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)

Remarque – 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.

mod n modulo arithmétique n.

C.5 *Description*

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

$$Y = X^e \text{ mod } n \quad \text{avec} \quad 0 \leq X < n$$

$$X = Y^d \text{ mod } n \quad 0 \leq Y < n$$

qui peuvent être satisfaites par exemple pour:

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

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

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 l). Le nombre entier est alors formé comme étant la somme des bits après avoir donné au premier bit le poids 2^{l-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.

C.6 Exigences de sécurité

C.6.1 Longueurs de clef

Il est reconnu que la longueur de clef 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*.

C.6.2 Génération de clefs

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 référence 2), § C.2]:

- a) elles devront être choisies au hasard;
- b) elles devront avoir 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 clefs publique et secrète, par exemple " X_p " et " X_s " définis aux § 3.3 et 4.1 de la présente Recommandation et qui sont constitués 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 [référence 3)].

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

C.7 Exposant public

L'exposant public (e) pourra être commun à tout l'environnement afin de minimiser la longueur de la partie de clef publique qui doit être effectivement diffusée et de réduire la capacité de transmission et la complexité de la transformation (voir la remarque 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 format F_4 (voir la remarque 2).

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

$$= 65537 \text{ en numérotation décimale, et}$$

$$= 1\ 0000\ 0000\ 0000\ 0001 \text{ en numérotation binaire.}$$

Remarque 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.

Remarque 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 format 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.

C.8 Conformité

Si la présente annexe spécifie un algorithme pour les fonctions publiques et secrètes, il 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 D

(à la Recommandation X.509)

Fonctions hachage

La présente annexe ne fait pas partie intégrante de la présente Recommandation.

D.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.

D.2 Description d'une fonction hachage

La fonction hachage suivante effectue la compression des données sur une base: bloc par bloc.

Ce hachage s'effectue en trois opérations principales:

- 1) la chaîne de données à hacher est divisée en blocs B d'égale longueur. Cette longueur est déterminée par les caractéristiques du système asymétrique de cryptographie utilisé pour la signature. Avec le système cryptographique RSA, cette longueur (en octets) est le plus grand nombre entier, l , tel qu'avec le module n , $16 \mid l < \log_2 n$;
- 2) pour des raisons d'impossibilité d'inversion, chaque octet du bloc est divisé en deux parties égales. Chacune de ces moitiés de bloc est précédée (remplie) de "uns" binaires. Grâce à cette répartition en zones, une inflexibilité ou redondance est introduite, qui augmente considérablement la qualité de l'impossibilité d'inversion que présente la fonction hachage. Chaque bloc généré au cours de l'opération 1) s'étend sur la longueur du module n ;
- 3) chaque bloc résultant de l'opération 2) s'ajoute au bloc précédent modulo 2, élevé au carré et réduit modulo n jusqu'à ce que tous les m blocs soient traités.

Le résultat est la valeur H_m , où:

$$H_0 = 0$$

$$H_i = (H_{i-1} \oplus B_i)^2 \bmod n, \text{ pour } 1 \leq i \leq m$$

Si le dernier bloc est incomplet, il est rempli avec "1".

ANNEXE E

(à la Recommandation X.509)

Dangers contre lesquels la protection est assurée par les services de sécurité

La présente annexe ne fait pas partie intégrante de la présente Recommandation.

La méthode d'authentification poussée décrite dans la présente Recommandation assure une protection contre les dangers, comme cela est décrit dans la partie de l'annexe A 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 clef secrète d'utilisateur – Un des principes de base de l'authentification poussée est que la clef secrète reste sûre. Plusieurs méthodes pratiques sont à la disposition de l'utilisateur pour qu'il conserve sa clef secrète 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 clef secrète d'une CA – Un principe de base d'une authentification poussée est également que la clef secrète 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 clef secrète 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 secrète 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 clef de plus grande longueur peuvent valablement être annoncées à l'avance.

ANNEXE F
(à la Recommandation X.509)
Confidentialité des données

La présente annexe ne fait pas partie intégrante de la présente Recommandation.

F.1 *Introduction*

Le traitement de la confidentialité des données peut être initialisé après que les clefs nécessaires au chiffrement ont été échangées. Celles-ci pourraient être fournies lors d'un échange d'authentifications précédent, comme décrit au § 9 ou par tout autre processus d'échange de clefs (ce dernier n'entre pas dans le cadre de la présente Recommandation).

La confidentialité des données peut être assurée en appliquant soit un schéma de chiffrement symétrique, soit un schéma de chiffrement asymétrique.

F.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 clef publique du destinataire auquel les données sont destinées: le destinataire les déchiffre alors en utilisant sa clef secrète.

F.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 au § 9 a été effectué par les deux parties concernées, une clef pour l'utilisation d'un algorithme symétrique peut être déterminée. Le choix des clefs secrètes dépend de la transformation à effectuer. Les parties doivent être sûres que ce sont des clefs à haut niveau de sécurité. La présente Recommandation 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 G
(à la Recommandation X.509)
Cadre d'authentification en ASN.1

La présente annexe fait partie intégrante de la présente Recommandation.

Cette annexe comprend la totalité des définitions ASN.1 des types de macros et des valeurs contenues dans la Recommandation sous la forme du module ASN.1 "**Cadre d'authentification**".

AuthenticationFramework {joint-iso-ccitt ds(5) modules(1)
authenticationFramework(7)}

DEFINITIONS ::=
BEGIN

EXPORTS AlgorithmIdentifier, AuthorityRevocationList, CACertificate, Certificate,
Certificates, CertificationPath, CertificateRevocationList,
UserCertificate, CrossCertificatePair, UserPassword, ALGORITHM,
ENCRYPTED, PROTECTED, SIGNATURE, SIGNED;

```

IMPORTS
  informationFramework, selectedAttributeTypes, upperBounds
  FROM UsefulDefinitions {joint-iso-ccitt ds(5)modules(1)
                        usefulDefinitions(0)}
  Name, ATTRIBUTE,ATTRIBUTE-SYNTAX
  FROM InformationFramework informationFramework
ub-user-passwordFROM UpperBounds upperBounds;
-- types
Certificate ::= SIGNED SEQUENCE{
                version [0] Version DEFAULT 1988,
                serialNumber SerialNumber,
                signature AlgorithmIdentifier,
                issuer Name,
                validity Validity,
                subject Name,
                subjectPublicKeyInfo SubjectPublicKeyInfo}

Version ::= INTEGER { 1988(0)}
SerialNumber ::= INTEGER
Validity ::= SEQUENCE{
                notBefore UTCTime
                notAfter UTCTime}

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

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

Certificates ::= SEQUENCE{
                certificate 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,
                lastUpdate UTCTime,
                revokedCertificates SIGNEDSEQUENCE OF SEQUENCE{
                signature AlgorithmIdentifier,
                issuer Name,
                userCertificate SerialNumber,
                revocationDate UTCTime}
                OPTIONAL}

CertificatePair ::= SEQUENCE{
                forward [0] Certificate OPTIONAL,
                reverse [1] Certificate OPTIONAL
                -- at least one of the pair must be present --}

-- attribute types
UserCertificate ::= ATTRIBUTE
                WITH ATTRIBUTE-SYNTAXCertificate
CACertificate ::= ATTRIBUTE
                WITH ATTRIBUTE-SYNTAXCertificate

```

```

CrossCertificatePair ::= ATTRIBUTE
                      WITH ATTRIBUTE-SYNTAXCertificatePair
CertificateRevocationList ::= ATTRIBUTE
                           WITH ATTRIBUTE-SYNTAXCertificateList
AuthorityRevocationList ::= ATTRIBUTE
                          WITH ATTRIBUTE-SYNTAXCertificateList
UserPassword ::= ATTRIBUTE
              WITH ATTRIBUTE-SYNTAX
              OCTETSTRING(SIZE(0...ub-user-password))
              MATCHES FOR EQUALITY

-- macros
ALGORITHM MACRO ::=
BEGIN
TYPE NOTATION ::= "PARAMETER" type
VALUE NOTATION ::= value(VALUE OBJECT IDENTIFIER)
END -- of ALGORITHM

ENCRYPTED MACRO ::=
BEGIN
TYPE NOTATION ::= type (ToBeEnciphered)
VALUENOTATION ::= value (VALUE BIT STRING
-- the value of the bit string is generated by
-- taking the octets which form the complete
-- encoding (using the ASN.1 Basic Encoding Rules)
-- of the value of the ToBeEnciphered type and
-- applying an encipherment procedure to those octets --
END

SIGNED MACRO ::=
BEGIN
TYPE NOTATION ::= type (ToBeSigned)
VALUE NOTATION ::= value(VALUE
SEQUENCE{
  ToBeSigned,
  AlgorithmIdentifier, -- of the algorithm used to generate the signature
  ENCRYPTED OCTET STRING
  -- where the octet string is the result
  -- of the hashing of the value of
  -- "ToBeSigned" --}
)
END -- of SIGNED

SIGNATURE MACRO ::=
BEGIN
TYPE NOTATION ::= type (OfSignature)
VALUE NOTATION ::= value(VALUE
  SEQUENCE{
    AlgorithmIdentifier,
    -- of the algorithm used to compute the signature
    ENCRYPTED OCTET STRING
    -- where the octet string is a function (e.g. a compressed or hashed version)
    -- of the value "OfSignature", which may include the identifier of the
    -- algorithm used to compute the signature --}
  )
END -- of SIGNATURE

PROTECTED MACRO ::= SIGNATURE

END -- of Authentication Framework Definitions

```

ANNEXE H

(à la Recommandation X.509)

Définition de référence des identificateurs d'objet d'algorithme

La présente annexe ne fait pas partie intégrante de la présente Recommandation.

Elle définit les identificateurs d'objet affectés aux algorithmes d'authentification et de chiffrement, en l'absence d'un enregistreur officiel. Il est prévu de recourir à un tel enregistreur quand il sera disponible. Les définitions prennent la forme du module ASN.1 **AlgorithmObjectIdentifiers**.

```

AlgorithmObjectIdentifiers    {joint-iso-ccitt ds(5) modules(1)
                                algorithmObjectIdentifiers(8)}

DEFINITIONS ::=
BEGIN

EXPORTS
    encryptionAlgorithm, hashAlgorithm, signatureAlgorithm,
    rsa, squareMod-n, sqMod-nWithRSA;

IMPORTS
    algorithm, authenticationFramework
        FROM UsefulDefinitions {joint-iso-ccitt ds(5) modules(1)
                                usefulDefinitions(0)}

    ALGORITHM FROM AuthenticationFramework authenticationFramework;

-- categories of object identifier

encryptionAlgorithm OBJECT IDENTIFIER ::= {algorithm 1}

hashAlgorithm OBJECT IDENTIFIER      ::= {algorithm 2}

signatureAlgorithm OBJECT IDENTIFIER  ::= {algorithm 3}

-- algorithms

rsa ALGORITHM
    PARAMETER KeySize
    ::= {encryptionAlgorithm 1}

KeySize ::= INTEGER

sqMod-n ALGORITHM
    PARAMETER BlockSize
    ::= {hashAlgorithm 1}

BlockSize ::= INTEGER

sqMod-nWithRSA ALGORITHM
    PARAMETER KeyAndBlockSize
    ::= {signatureAlgorithm 1}

KeyAndBlockSize ::= INTEGER

END -- of Algorithm Object Identifier Definitions

```

SÉRIES DES RECOMMANDATIONS UIT-T

Série A	Organisation du travail de l'UIT-T
Série B	Moyens d'expression: définitions, symboles, classification
Série C	Statistiques générales des télécommunications
Série D	Principes généraux de tarification
Série E	Exploitation générale du réseau, service téléphonique, exploitation des services et facteurs humains
Série F	Services de télécommunication non téléphoniques
Série G	Systèmes et supports de transmission, systèmes et réseaux numériques
Série H	Systèmes audiovisuels et multimédias
Série I	Réseau numérique à intégration de services
Série J	Transmission des signaux radiophoniques, télévisuels et autres signaux multimédias
Série K	Protection contre les perturbations
Série L	Construction, installation et protection des câbles et autres éléments des installations extérieures
Série M	RGT et maintenance des réseaux: systèmes de transmission, de télégraphie, de télécopie, circuits téléphoniques et circuits loués internationaux
Série N	Maintenance: circuits internationaux de transmission radiophonique et télévisuelle
Série O	Spécifications des appareils de mesure
Série P	Qualité de transmission téléphonique, installations téléphoniques et réseaux locaux
Série Q	Commutation et signalisation
Série R	Transmission télégraphique
Série S	Equipements terminaux de télégraphie
Série T	Terminaux des services télématiques
Série U	Commutation télégraphique
Série V	Communications de données sur le réseau téléphonique
Série X	Réseaux de données et communication entre systèmes ouverts
Série Y	Infrastructure mondiale de l'information et protocole Internet
Série Z	Langages et aspects informatiques généraux des systèmes de télécommunication