UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T Q.815

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

(02/2000)

SÉRIE Q: COMMUTATION ET SIGNALISATION

Spécifications du système de signalisation n° 7 – Interface Q3

Spécification d'un module de sécurité pour la protection globale des messages

Recommandation UIT-T Q.815

(Antérieurement Recommandation du CCITT)

RECOMMANDATIONS UIT-T DE LA SÉRIE Q

COMMUTATION ET SIGNALISATION

SIGNALISATION DANS LE SERVICE MANUEL INTERNATIONAL	Q.1–Q.3
EXPLOITATION INTERNATIONALE AUTOMATIQUE ET SEMI-AUTOMATIQUE	Q.4–Q.59
FONCTIONS ET FLUX D'INFORMATION DES SERVICES DU RNIS	Q.60–Q.99
CLAUSES APPLICABLES AUX SYSTÈMES NORMALISÉS DE L'UIT-T	Q.100–Q.119
SPÉCIFICATIONS DES SYSTÈMES DE SIGNALISATION N° 4 ET N° 5	Q.120–Q.249
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 6	Q.250–Q.309
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R1	Q.310–Q.399
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION R2	Q.400–Q.499
COMMUTATEURS NUMÉRIQUES	Q.500–Q.599
INTERFONCTIONNEMENT DES SYSTÈMES DE SIGNALISATION	Q.600–Q.699
SPÉCIFICATIONS DU SYSTÈME DE SIGNALISATION N° 7	Q.700–Q.849
Généralités	Q.700
Sous-système transport de messages	Q.701–Q.709
Sous-système commande des connexions sémaphores	Q.711–Q.719
Sous-système utilisateur de téléphonie	Q.720–Q.729
Services complémentaires du RNIS	Q.730–Q.739
Sous-système utilisateur de données	Q.740–Q.749
Gestion du système de signalisation n° 7	Q.750–Q.759
Sous-système utilisateur du RNIS	Q.760–Q.769
Sous-système application de gestion des transactions	Q.770–Q.779
Spécification des tests	Q.780–Q.799
Interface Q3	**Q.800–Q.849**
SYSTÈME DE SIGNALISATION D'ABONNÉ NUMÉRIQUE N° 1	Q.850–Q.999
RÉSEAUX MOBILES TERRESTRES PUBLICS	Q.1000–Q.1099
INTERFONCTIONNEMENT AVEC LES SYSTÈMES MOBILES À SATELLITES	Q.1100–Q.1199
RÉSEAU INTELLIGENT	Q.1200–Q.1699
PRESCRIPTIONS ET PROTOCOLES DE SIGNALISATION POUR LES IMT-2000	Q.1700–Q.1799
RNIS À LARGE BANDE	Q.2000–Q.2999

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

Recommandation UIT-T Q.815

Spécification d'un module de sécurité pour la protection globale des messages

Résumé

La présente Recommandation spécifie un module de sécurité facultatif à utiliser avec la Recommandation UIT-T Q.814, Spécification d'un agent interactif d'échange informatisé de données, qui fournit des services de sécurité pour l'ensemble des unités de données protocolaires (PDU). Le module de sécurité prend notamment en charge la non-répudiation de l'origine et de la réception, ainsi que l'intégrité globale des messages.

Source

La Recommandation Q.815 de l'UIT-T, élaborée par la Commission d'études 4 (1997-2000) de l'UIT-T, a été approuvée le 4 février 2000 selon la procédure définie dans la Résolution 1 de la CMNT.

AVANT-PROPOS

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

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

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

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

NOTE

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

DROITS DE PROPRIÉTÉ INTELLECTUELLE

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

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

TABLE DES MATIÈRES

Recommandation UIT-T Q.815

Spécification d'un module de sécurité pour la protection globale des messages

1 Domaine d'application

Le module de sécurité fournit des services de sécurité pour l'ensemble des unités de données protocolaires (PDU, *protocol data unit*). Il prend notamment en charge la non-répudiation de l'origine et de la réception, ainsi que l'intégrité globale des messages.

Dans le contexte de transactions RGT fondées sur des échanges informatisé de données (EDI, *electronic data interchange*), à l'extrémité émettrice, le module de sécurité accepte comme données d'entrée les données de sortie du traducteur EDI, effectue les transformations de sécurité demandées et fournit la chaîne d'octets qui en résulte à l'agent interactif (IA, *interactive agent*).

A l'extrémité réceptrice, le module de sécurité reçoit une chaîne d'octets de l'agent IA qu'il interprète comme une unité PDU de module de sécurité. Il procède ensuite à la vérification de la validité du message sous-jacent. Dans le cas d'une protection d'intégrité, si le message est valide, il le transmet (sans le code d'intégrité du message) au traducteur EDI.

Le module de sécurité conserve le journal de tous les messages reçus de l'agent IA. Il transmet ces messages, avec, pour chaque message, une indication du résultat de la procédure de vérification. Ce journal est disponible pour l'utilisateur EDI local.

Les aspects spécifiques du journal, ainsi que l'interface avec le journal, sont un sujet annexe qui sort du cadre de la présente Recommandation UIT-T. Le comportement du module de sécurité en cas d'échec des vérifications constitue un sujet annexe qui sort du cadre de la présente Recommandation UIT-T.

La Figure 1 ci-dessous, copiée de la Figure 2/Q.814, est utilisée comme référence.

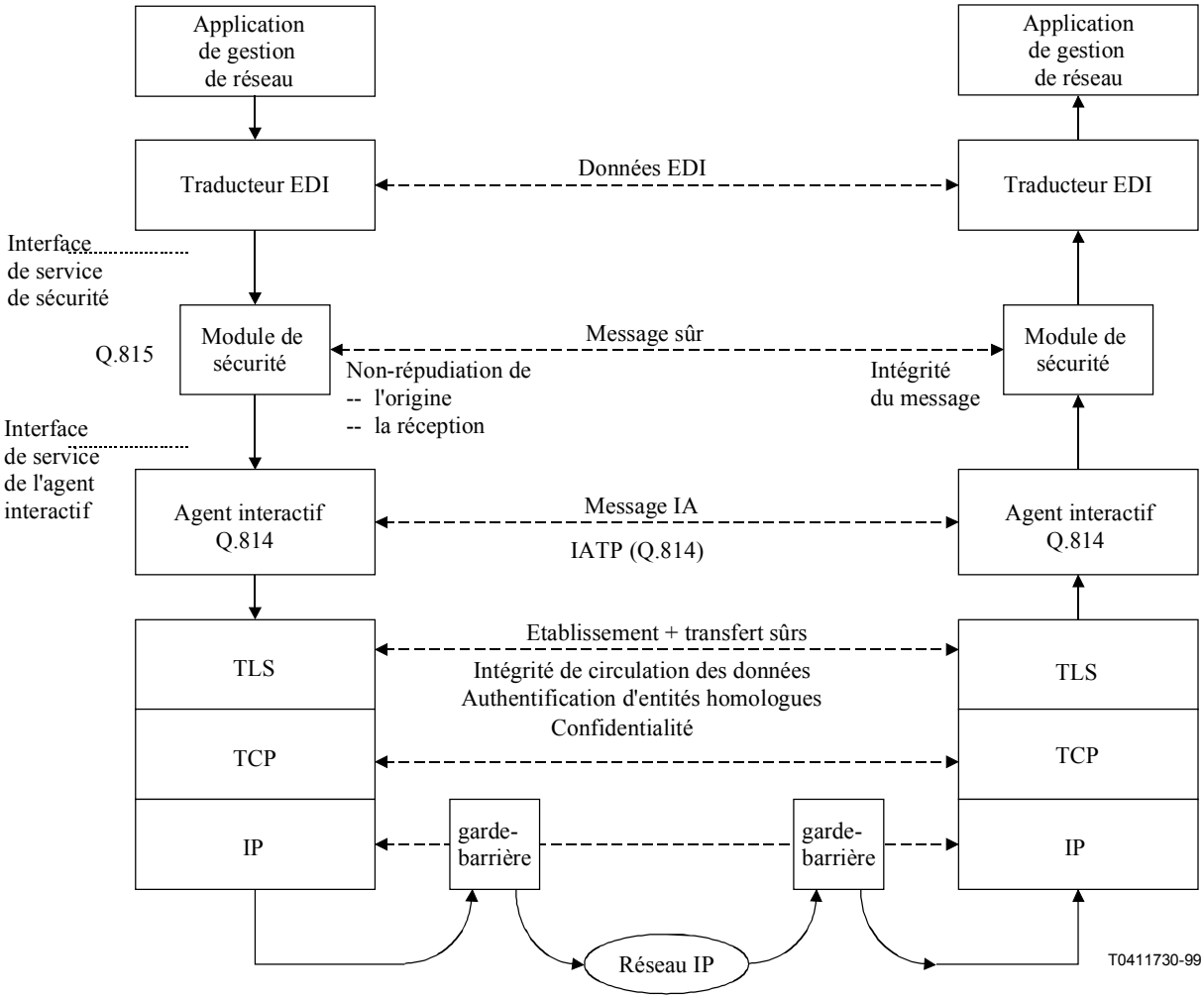

Figure 1/Q.815 – Relations pendant la circulation des messages (avec services de sécurité des messages)

2 Références

2.1 Références normatives

La présente Recommandation se réfère à certaines dispositions des Recommandations UIT-T et textes suivants qui, de ce fait, en sont partie intégrante. Les versions indiquées étaient en vigueur au moment de la publication de la présente Recommandation. Toute Recommandation ou tout texte étant sujet à révision, les utilisateurs de la présente Recommandation sont invités à se reporter, si possible, aux versions les plus récentes des références normatives suivantes. La liste des Recommandations de l'UIT-T en vigueur est régulièrement publiée.

– Recommandation UIT-T Q.812 (1997)/Amd.3 (2000), *Profils des protocoles des couches supérieures pour les interfaces Q3 et X – Amendement 3: profil des protocoles pour l'agent interactif de communications électroniques*.

– Recommandation UIT-T Q.814 (2000), *Spécification d'un agent interactif d'échange informatisé de données*.

– Recommandation UIT-T X.509 (1997) | ISO/CEI 9495-8:1998, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: cadre d'authentification*.

– Recommandation UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification de la notation de base*.

– Recommandation UIT-T X.681 (1997) | ISO/CEI 8824-2:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des objets informationnels*.

– Recommandation UIT-T X.682 (1997) | ISO/CEI 8824-3:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: spécification des contraintes*.

– Recommandation UIT-T X.683 (1997) | ISO/CEI 8824-4:1998, *Technologies de l'information – Notation de syntaxe abstraite numéro un: paramétrage des spécifications de la notation de syntaxe abstraite numéro un*.

– Recommandation UIT-T X.690 (1997) | ISO/CEI 8825-1:1998, *Technologies de l'information – Règles de codage ASN.1: spécification des règles de codage de base, des règles de codage canoniques et des règles de codage distinctives*.

ISO – Organisation internationale de normalisation:

– ISO 3166 (toutes les parties), *Codes pour la représentation des noms de pays et de leurs subdivisions*.

2.2 Références informatives

– *Directory Implementors Guide (Version 11) (1998)*.

3 Définitions

3.1 unité de données protocolaire d'application: paquet de données échangé entre deux programmes d'application sur un réseau. Ceci représente le niveau le plus élevé de communication dans le modèle OSI à sept couches; un paquet échangé à ce niveau peut en fait être transmis en plusieurs paquets sur une couche inférieure, il peut également comporter des informations supplémentaires (en-têtes) ajoutées pour l'acheminement, etc.

3.2 règles de codage distinctives: forme restreinte des règles de codage de base définies dans la Recommandation UIT-T X.690 pour éliminer les options des règles BER.

3.3 échange informatisé de données: échange de documents sous une forme électronique normalisée, entre des organismes, de manière automatique, directement depuis une application informatique d'un organisme vers une application d'un autre organisme.

3.4 échange informatisé de données pour l'administration, le commerce et le transport: cette syntaxe provient d'une Norme ISO (ISO 9735), et a été adoptée par l'Organisation des Nations Unies (ONU) comme base pour le développement des messages commerciaux pour l'EDI (UN/EDIFACT). EDIFACT a été développé dans le but de regrouper des normes précédentes et l'ASC X12.

3.5 agent interactif: l'agent interactif prend en charge le déroulement de transactions (ANSI ASC X12 EDI ou EDIFACT) par échange informatisé de données entre des entités homologues dans le secteur des télécommunications.

3.6 RSA: le système RSA est un système de chiffrement par clef publique développé par Ronald L. Rivest, Adi Shamir et Leonard M. Adleman en 1977 afin de contribuer à assurer la sécurité sur Internet. Un système de chiffrement est simplement un algorithme qui peut convertir des données d'entrée en quelque chose de non reconnaissable (chiffrement), et reconvertir ces données non reconnaissables dans leur forme d'origine (déchiffrement). Les techniques de chiffrement RSA sont décrites dans la Recommandation UIT-T X.509.

3.7 algorithme de hachage de sécurité, révision 1: fonction de hachage de 160 bits, mandatée par le National Institute for Standards Technology (NIST), avec des mécanismes de sécurité similaires à l'algorithme MD5. L'algorithme SHA-1 est défini par le gouvernement américain dans le document FIPS 180-1. C'est un mécanisme permettant de réduire un message long en un court résumé de 160 bits unidirectionnel (c'est-à-dire non réversible) et non susceptible de subir des collisions avec des textes multiples différents. Etant donné que l'algorithme SHA génère un hachage de 160 bits (résumé de message), il est beaucoup mieux protégé des attaques cryptographiques de force brute que l'algorithme MD5.

La meilleure comparaison que l'on puisse faire d'un résumé de message est celle de l'empreinte numérique d'un message. C'est un algorithme relativement rapide, de surdébit faible et sûr. L'algorithme SHA-1 peut être utilisé pour assurer la protection de l'intégrité (tout seul) ou pour la non-répudiation (associé à un chiffrement de clé publique). L'algorithme de résumé de message SHA-1 est décrit dans l'Appendice II.

4 Abréviations

La présente Recommandation UIT-T utilise les abréviations suivantes:

APDU unité de données protocolaire d'application (*application protocol data unit*)

ASC comité de normalisation reconnu (*accredited standards committee*)

DER règles de codage distinctives (de ASN.1) (*distinguished encoding rules*)

EDI échange informatisé de données (*electronic data interchange*)

EDIFACT échange informatisé de données pour l'administration, le commerce et le transport (*electronic data interchange for administration, commerce and transport*)

IA agent interactif (*interactive agent*)

IP protocole Internet (*Internet protocol*)

PDU unité de données protocolaire (*protocol data unit*)

RGT réseau de gestion des télécommunications

RSA Rivest, Shamir, Aldeman

SHA-1 algorithme de hachage de sécurité, révision 1 (*secure hash algorithm, revision 1*)

SM module de sécurité (*security module*)

SR élément STASE ROSE

TCP protocole de commande de transmission (*transmission control protocol*)

TLS sécurité de la couche de transport (*transport layer security*)

5 Conventions

Les conventions suivantes sont utilisées: les références aux paragraphes, sous-paragraphes, annexes et appendices correspondent aux éléments présents dans la présente Recommandation UIT-T, sauf spécification contraire explicitement indiquée.

6 Détails Q.815

6.1 Types de messages de module de sécurité

Le module de sécurité Q.815 spécifie les directives suivantes:

les services d'intégrité et/ou de non-répudiation de message peuvent être fournis par le module de sécurité (SM). Si le module SM fournit un service d'intégrité de message, il utilise un algorithme SHA-1 pour produire le résumé de message (MD, *message digest*). Si le module SM fournit un service de non-répudiation d'origine, il utilise un algorithme SHA-1 pour produire un résumé de message puis utilise le mécanisme de signature numérique RSA. Le service de non-répudiation de réception est fourni au moyen d'un mécanisme décrit dans la présente Recommandation UIT-T. Un quatrième ensemble d'amélioration de la sécurité est disponible par l'utilisation de l'élément STASE-ROSE (voir Recommandation UIT-T Q.813).

6.1.1 Service d'intégrité de message

L'utilisateur direct fournit au module SM un message EDIFACT ou ASC X12 EDI. Le module SM calcule un résumé de message en utilisant les données EDIFACT ou ASC X12 EDI comme données d'entrée pour l'algorithme du résumé. Le module SM code un message haché en utilisant les règles DER conformément au 7.1. L'agent IA utilise la chaîne d'octets codée selon les règles DER comme le contenu d'un message avancé, tel que défini dans la Recommandation UIT-T Q.814.

6.1.2 Service de non-répudiation d'origine

L'utilisateur direct fournit au module SM un message EDIFACT ou ASC X12 EDI. Le module SM calcule un résumé de message en utilisant les données EDIFACT ou ASC X12 EDI comme données d'entrée pour l'algorithme du résumé. Le module effectue le chiffrement du résumé du message au moyen des règles de codage DER conformément à l'algorithme de signature numérique en utilisant la clé privée de l'émetteur. Le module SM code un message signé en utilisant les règles DER conformément au 7.2. L'agent IA utilise la chaîne d'octets codée selon les règles DER comme le contenu d'un message avancé, comme défini dans la Recommandation UIT-T Q.814.

6.1.3 Service de non-répudiation de réception

L'utilisateur direct fournit au module SM un identificateur de message unique avec la date et l'heure de réception du message correspondant. Voir 7.3.2 pour de plus amples informations concernant l'identificateur unique et la date/l'heure. En option, dans le cadre d'un accord bilatéral, l'utilisateur direct peut être obligé de fournir un résumé de message ou une signature numérique comme une des composantes de la réception. Le résumé ou la signature doit correspondre au message identifié de manière unique tel que décrit ci-dessus. Le module SM code un message IA reçu en utilisant les règles DER conformément au 7.3. L'agent IA utilise la chaîne d'octets codée selon les règles DER comme le contenu d'un message avancé, comme défini dans la Recommandation UIT-T Q.814.

6.2 Caractéristiques générales

Les résumés de message et les signatures numériques sont calculés sur le texte en clair (non chiffré) des messages EDIFACT/ASC X12 EDI/Chaîne générale.

Les certificats numériques doivent être compatibles avec la Recommandation UIT-T X.509 version 3.

7 Syntaxe générale

La syntaxe générale du message du module de sécurité se présente comme suit:

```
SecureMessage ::=  CHOICE {
hashedMessage      [0] EXPLICIT HashedMessage,
signedMessage      [1] EXPLICIT SignedMessage,
messageReceipt     [2] EXPLICIT IaReceiptMessage,
sr-APDU            [3] EXPLICIT SR-APDU,
…
}
```

NOTE – Un module ASN.1 comprenant une syntaxe générale et chacune des syntaxes spécifiques définies dans le présent paragraphe se trouve à l'Annexe A.

7.1 Messages hachés

Les services d'intégrité de message sont fournis par le message *HashedMessage* défini comme suit:

```
HashedMessage :: =              SEQUENCE {
        hashedVersion           Version DEFAULT v1999 ,
        hashAlgorithmIdentifier AlgorithmIdentifier,
        hashedContent           HashedContent,          -- Données
        messageDigest           OCTET STRING ( SIZE (20) )
  }
HashedContent :: =      CHOICE {
hashedContent1          GeneralString,
hashedContent2          IA5String
}
```

7.1.1 Identificateurs d'objet référencés par messages hachés

Dans le message *hashedMessage*, la valeur à utiliser pour l'identificateur hashAlgorithmIdentifier est:

```
hashAlgorithmIdentifier OBJECT IDENTIFIER :: = {
        iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 26 }    -- SHA-1
```

7.1.2 Informations sur les valeurs pour les messages hachés

hashedContent est un message EDIFACT, ASC X12 EDI ou autre chaîne.

messageDigest est un résumé SHA-1 du message *hashedContent*.

7.2 Messages signés

Des services de non-répudiation d'origine sont fournis par le message *SignedMessage* défini comme suit:

```
SignedMessage :: = SEQUENCE {
        signedVersion                   Version DEFAULT v1999 ,
        signedDigestAlgorithms          SET OF AlgorithmIdentifier,
        signedContent                   SignedContent,      -- Données
        signerInfos                     SET OF SEQUENCE {
                signerVersion           Version DEFAULT v1999 ,
                issuerAndSerialNumber   SEQUENCE {
                        issuerCountry   SEQUENCE OF SET OF SEQUENCE {
                                country         OBJECT IDENTIFIER,
                                countryValue    PrintableString
                        },
```

```
                issuerOrg                   SEQUENCE OF SET OF SEQUENCE {
                    organizationName        OBJECT IDENTIFIER,
                    organizationValue       PrintableString
                },
                serialNumber                INTEGER
            },
            signedDigestAlgorithm           AlgorithmIdentifier,
            digestEncryptionAlgorithm       AlgorithmIdentifier,
            encryptedDigest                 OCTET STRING
        }
}
SignedContent :: =      CHOICE {
signedContent1          GeneralString,
signedContent2          IA5String
}
```

7.2.1 Identificateurs d'objet référencés par messages signés

Dans le message *SignedMessage*, la valeur à utiliser pour l'algorithme signedDigestAlgorithms est:

```
signedDigestAlgorithms OBJECT IDENTIFIER :: = {
   iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 26 }    -- SHA-1
country OBJECT IDENTIFIER :: = { 2 5 4 6 }
organizationName OBJECT IDENTIFIER :: = { 2 5 4 10 }
```

Dans le message *SignedMessage*, la valeur à utiliser pour l'algorithme signedDigestAlgorithm est:

```
signedDigestAlgorithm OBJECT IDENTIFIER :: = {
   iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 26 }    -- SHA-1
```

Dans le message *SignedMessage*, la valeur à utiliser pour l'identificateur digestEncryptionAlgorithmIdentifier est:

```
digestEncryptionAlgorithmIdentifier OBJECT IDENTIFIER :: = {
   iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) 1 1 } -- RSA
```

7.2.2 Informations sur les valeurs pour les messages signés

*signedConten*t est un message EDIFACT, ASC X12 EDI, ou autre chaîne.

countryValue est la représentation à deux caractères du nom du pays de l'émetteur, telle que définie dans l'ISO 3166.

organizationValue est le nom de l'organisme de l'émetteur du certificat.

serialNumber est un numéro unique attribué au certificat par l'émetteur.

encryptedDigest est un codage DER du résumé SHA-1 du message *signedContent*, chiffré par RSA avec la clé privée de la partie émettrice.

signedDigestAlgorithms – cet ensemble doit être composé d'un membre unique (AlgorithmIdentifier pour l'algorithme du résumé de message). L'algorithme doit être le même pour signedDigestAlgorithms et signedDigestAlgorithm.

signerInfos – cet ensemble doit être composé d'un membre unique.

La paire clef privée/clef publique associée au certificat numérique utilisée pour l'authentification doit être utilisée pour la signature/vérification.

issuerCountry – cet ensemble doit être composé d'un membre unique.

issuerOrg – cet ensemble doit être composé d'un membre unique.

NOTE – L'algorithme de signature numérique RSA convertit le type de données INTEGER d'une signature numérique en une chaîne OCTET STRING.

7.3 Message de réception IA

Les services de non-répudiation de réception sont fournis par le message *IaReceiptMessage* défini comme suit:

```
IaReceiptMessage :: =     SEQUENCE {
        uniqueIdentifier  OCTET STRING, -- Identificateur unique dans le message
        dateTimeStamp     PrintableString ( SIZE(15) ),
        enhancements      Enhancements OPTIONAL
}
Enhancements :: =  CHOICE {
        withDigest   [0] EXPLICIT WithDigest,
        withDigSig   [1] EXPLICIT WithDigSig
}
WithDigest :: =                   SEQUENCE {
        receiptDigestAlgorithm    OBJECT IDENTIFIER,
        receiptMessageDigest      OCTET STRING
}
WithDigSig :: =                   SEQUENCE {
        receiptSignatureAlgorithm OBJECT IDENTIFIER,
        receiptDigitalSignature   OCTET STRING
}
```

7.3.1 Identificateurs d'objet référencés par messages de réception IA

Dans le message *IaReceiptMessage*, la valeur à utiliser pour l'algorithme receiptDigestAlgorithm est:

```
receiptDigestAlgorithm OBJECT IDENTIFIER :: = {
  iso(1) identified-organization(3) oiw(14) secsig(3) algorithm(2) 26 }   -- SHA-1
```

Dans le message *IaReceiptMessage*, la valeur à utiliser pour l'algorithme receiptSignatureAlgorithm est:

```
receiptSignatureAlgorithm OBJECT IDENTIFIER :: = {
  iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) 1 5 }
```

7.3.2 Informations sur les valeurs pour les messages de réception IA

uniqueIdentifier est un champ quelconque dans les données du message qui change d'un message à l'autre (par exemple segment ASC X12 EDIs ISA). Des entités homologues déterminent le champ de données à utiliser dans ce but au moyen d'un mécanisme sortant du cadre de la présente Recommandation UIT-T.

dateTimeStamp est formaté comme suit: CCYYMMDDhhmmssz

où: CC = siècle

YY = année

MM = mois

DD = jour

hh = heure

mm = minute

ss = seconde

z = indicateur de fuseau horaire alpha.

Un blanc pour l'indicateur de fuseau horaire indique l'heure locale observée. Il convient que la date et l'heure de l'horodatage correspondent à la date et l'heure de réception du message complet par la partie réceptrice. Il est recommandé d'utiliser le temps universel coordonné (Z) à la réception pour éviter toute ambiguïté de fuseau horaire.

receiptMessageDigest est le résumé du message reçu. Si le message était de format SimpleHashed, le résumé est celui qui a été reçu avec le message et qui a été vérifié par le destinataire. Si le message original était de tout autre type de données, le résumé doit être calculé par le destinataire en utilisant les données du message EDIFACT/ASC X12 EDI/Chaîne générale comme données d'entrée pour l'algorithme du résumé de message spécifié.

receiptDigitalSignature est calculé en formant une concaténation de l'identificateur uniqueIdentifier du message EDIFACT/ASC X12 EDI/Chaîne générale, du champ dateTimeStamp de la réception (15 octets) et de la signature numérique reçue avec le message original. Cette structure en octets est ensuite signée par chiffrement conformément à l'algorithme de signature numérique RSA, en utilisant SHA-1 comme l'algorithme du résumé et la clé privée de la partie générant la réception. Il convient que la partie réceptrice construise une concaténation de l'identificateur uniqueIdentifier (à partir du message transmis ou du corps de la réception), du champ dateTimeStamp de 15 octets (à partir du corps de la réception) et de encryptedDigest (signature numérique) à partir du message original. Il convient de transmettre cette structure concaténée, avec la signature numérique du message de réception proprement dit (receiptDigitalSignature) à l'algorithme de vérification de la signature numérique. La clé publique de la partie réceptrice est utilisée pour vérifier la signature. Ce type de réception implique que le message original reçu soit de type SignedMessage.

7.4 Unité PDU Stase-Rose

La syntaxe pour ce type de message est tirée de la Recommandation UIT-T Q.813.

ANNEXE A

Module de production ASN.1

```
SecurityModule {itu-t(0) recommendation(0) q(17) q815(815) sm(0) messages(0)}
DEFINITIONS IMPLICIT TAGS :: = BEGIN

-- EXPORTER tout

IMPORTS
SR-APDU FROM Secure-Remote-Operations-APDUs
        {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0) stase-data(2)};

-- Types utiles

AlgorithmIdentifier :: = SEQUENCE {
                algorithm OBJECT IDENTIFIER,
                parameters NULL
}

Version :: = INTEGER
v1999 Version :: = 0

-- Syntaxe générale

SecureMessage :: = CHOICE {
hashedMessage       [0] EXPLICIT HashedMessage,
signedMessage       [1] EXPLICIT SignedMessage,
messageReceipt      [2] EXPLICIT IaReceiptMessage,
sr-APDU             [3] EXPLICIT SR-APDU,
…
}
```

-- Syntaxe message haché

```
HashedMessage :: =                  SEQUENCE {
        hashedVersion               Version DEFAULT v1999 ,
        hashAlgorithmIdentifier     AlgorithmIdentifier,
        hashedContent               HashedContent,          -- Données
        messageDigest               OCTET STRING ( SIZE (20) )
        }
HashedContent :: =          CHOICE {
hashedContent1              GeneralString,
hashedContent2              IA5String
}
```

-- Syntaxe message signé

```
SignedMessage :: =                      SEQUENCE {
        signedVersion                   Version DEFAULT v1999 ,
        signedDigestAlgorithms          SET OF AlgorithmIdentifier,
        signedContent                   SignedContent,          -- Données
        signerInfos                     SET OF SEQUENCE {
                signerVersion           Version DEFAULT v1999 ,
                issuerAndSerialNumber   SEQUENCE {
                        issuerCountry   SEQUENCE OF SET OF SEQUENCE {
                          country       OBJECT IDENTIFIER,
                  countryValue          PrintableString
                },
                issuerOrg               SEQUENCE OF SET OF SEQUENCE {
                  organizationName      OBJECT IDENTIFIER,
                  organizationValue     PrintableString
                },
                serialNumber            INTEGER
        },
        signedDigestAlgorithm           AlgorithmIdentifier,
        digestEncryptionAlgorithm       AlgorithmIdentifier,
        encryptedDigest                 OCTET STRING
  }
}

SignedContent :: =          CHOICE {
signedContent1              GeneralString,
signedContent2              IA5String
}
```

-- Syntaxe message de réception

```
IaReceiptMessage :: =       SEQUENCE {
        uniqueIdentifier    OCTET STRING, -- Identificateur unique dans le message
        dateTimeStamp       PrintableString ( SIZE(15) ),
        enhancements        Enhancements OPTIONAL
}

Enhancements :: =           CHOICE {
  withDigest                [0] EXPLICIT WithDigest,
  withDigSig                [1] EXPLICIT WithDigSig
}

WithDigest :: =                     SEQUENCE {
  receiptDigestAlgorithm            OBJECT IDENTIFIER,
  receiptMessageDigest              OCTET STRING
}
```

```
WithDigSig :: =                    SEQUENCE {
  receiptSignatureAlgorithm        OBJECT IDENTIFIER,
  receiptDigitalSignature          OCTET STRING
}

FIN
```

APPENDICE I

Références non normatives

– ISO 9735:1988, *Echange de données informatisées pour l'administration, le commerce et le transport (EDIFACT) – Règles de syntaxe au niveau de l'application*.

– ANSI ASC X12: American National Standards Institute (ANSI) Accredited Standards Committee X12. Ce comité a été mandaté par l'ANSI en 1979 pour développer des normes uniformes concernant les échanges électroniques de documents commerciaux.

APPENDICE II

Algorithme de résumé de message SHA-1

II.1 Introduction

Pour un message de longueur < 2^{64} bits, l'algorithme SHA-1 produit une représentation condensée de 160 bits du message appelé résumé de message. Le résumé de message est utilisé pendant la génération d'une signature pour le message. L'algorithme SHA-1 est également utilisé pour calculer un résumé de message pour la version reçue du message pendant le processus de vérification de la signature. Toute modification du message en transit se traduira, selon une très forte probabilité, par un résumé de message différent et l'échec du processus de vérification de la signature.

L'algorithme SHA-1 est conçu pour présenter les propriétés suivantes: impossibilité (sur le plan calcul) de trouver un message qui corresponde à un résumé de message donné, ou deux messages différents qui produisent le même résumé de message.

II.2 Chaînes binaires et nombres entiers

La terminologie suivante relative aux chaînes binaires et aux nombres entiers sera utilisée:

a) Un chiffre hexadécimal est un élément de l'ensemble {0, 1, ... , 9, A, ... , F}. Un chiffre hexadécimal est la représentation d'une chaîne de 4 bits.

Exemples: 7 = 0111, A = 1010

b) Un mot est égal à une chaîne de 32 bits qui peut être représentée sous la forme d'une séquence de 8 chiffres hexadécimaux. Pour convertir un mot en 8 chiffres hexadécimaux, on convertit chaque chaîne de 4 bits en son équivalent sous forme hexadécimale comme indiqué en a) ci-dessus.

Exemple: 1010 0001 0000 0011 1111 1110 0010 0011 = A103FE23

c) Un nombre entier entre 0 et $2^{32} - 1$ inclus peut être représenté sous la forme d'un mot. Les quatre bits de plus faible poids du nombre entier sont représentés par le chiffre hexadécimal de droite de la représentation de ce mot.

Exemple: le nombre entier 291 = $2^8+2^5+2^1+2^0$ = 256+32+2+1 est représenté par le mot hexadécimal 00000123

Si z est un nombre entier, $0 \leq z < 2^{64}$, alors $z = 2^{32}x + y$ où $0 \leq x < 2^{32}$ et $0 \leq y < 2^{32}$. Comme x et y peuvent être représentés sous la forme des mots X et Y, respectivement, z peut être représenté comme la paire des mots (X, Y).

d) block = chaîne de 512 bits. Un bloc (B, par exemple) peut être représenté sous la forme d'une séquence de 16 mots.

II.3 Opérations sur les mots

Les opérateurs logiques suivants s'appliqueront aux mots:

a) Opérations logiques sur les mots au niveau du bit

X ^ Y = "et" logique de X et Y au niveau du bit.

X ∨ Y = "ou inclusif" logique de X et Y au niveau du bit.

X XOR Y = "ou exclusif" logique de X et Y au niveau du bit.

~ X = "complément" logique de X au niveau du bit.

Exemple:

```
        01101100101110011101001001111011
XOR     01100101110000010110100110110111
        ------------------------------------------------
        =  00001001011110001011101111001100
```

b) L'opération X + Y est définie comme suit: les mots X et Y représentent les nombres entiers x et y, où $0 \leq x < 2^{32}$ et $0 \leq y < 2^{32}$. Pour les nombres entiers positifs n et m, on admet que n mod m est ce qui reste après division de n par m. Calculons $z = (x + y) \bmod 2^{32}$.

On obtient alors $0 \leq z < 2^{32}$. Convertissons z en un mot, Z, et définissons Z = X + Y.

c) L'opération de décalage circulaire à gauche S^n (X), où X est un mot et n est un nombre entier tel que $0 \leq n^{32}$, est définie par $S^n(X) = (X<<n)$ OU $(X >>32-n)$.

Dans la formule ci-dessus, X << n est obtenu comme suit: on supprime les n bits de gauche de X puis on complète le résultat en ajoutant n zéros à droite (le résultat sera toujours de 32 bits). On obtient X >> n en supprimant les n bits de droite de X puis en complétant le résultat en ajoutant n zéros à gauche. Ainsi, $S^n(X)$ équivaut à un décalage circulaire de X de n positions vers la gauche.

II.4 Remplissage d'un message

L'algorithme SHA-1 est utilisé pour calculer un résumé de message pour un message ou fichier de données fourni en entrée. Le message ou fichier de données doit être considéré comme étant une chaîne binaire. La longueur du message équivaut au nombre de bits composant le message (la longueur d'un message vide est égale à 0). Si le nombre de bits d'un message est un multiple de 8, on peut représenter ce message sous forme hexadécimale pour le rendre plus compact. Le remplissage des messages a pour objet de faire en sorte que la longueur totale d'un message dûment rempli soit un multiple de 512. L'algorithme SHA-1 traite séquentiellement des blocs de 512 bits pendant le calcul du résumé du message. Il est indiqué ci-dessous comment ce remplissage doit être effectué. En bref, on ajoute à la fin du message un "1" suivi de m "0", suivis d'un nombre entier de 64 bits, pour obtenir un message complet d'une longueur de n 512*. Le nombre entier de 64 bits est égal à 1, soit la longueur du message initial. Le message dûment rempli est ensuite traité par l'algorithme SHA-1 comme n blocs de 512 bits.

Supposons qu'un message ait une longueur de $1 < 2^{64}$. Avant d'être soumis à l'algorithme SHA-1, le message est complété à droite comme suit:

a) On ajoute "1". Exemple: si le message initial est "01010000", il devient après remplissage "010100001".

b) On ajoute des "0". Le nombre de "0" dépendra de la longueur initiale du message. Les 64 derniers bits du dernier bloc de 512 bits sont réservés pour la longueur 1 du message initial. **Exemple**: supposons que le message initial soit constitué de la chaîne binaire suivante:

   01100001 01100010 01100011 01100100 01100101

   A l'issue de l'étape a), on obtient:

   01100001 01100010 01100011 01100100 01100101 1

   Comme 1 = 40, le nombre de bits de la chaîne binaire ci-dessus est de 41 et on ajoute 407 "0", ce qui porte maintenant le total à 448. Cela donne (sous forme hexadécimale)

   61626364 65800000 00000000 00000000

   00000000 00000000 00000000 00000000

   00000000 00000000 00000000 00000000

   00000000 00000000

c) Calculons la représentation en deux mots de 1, le nombre de bits composant le message initial. Si $1 < 2^{32}$, le premier mot est composé uniquement de zéros. Ajouter ces deux mots au message complété.

   Exemple: supposons que le message initial soit tel qu'indiqué en b). Alors, 1 = 40 (à noter que 1 est calculé avant toute opération de remplissage). La représentation en deux mots de 40, sous forme hexadécimale, est la suivante: 00000000 00000028. Le message final après remplissage se présente donc comme suit, sous forme hexadécimale:

   61626364 65800000 00000000 00000000

   00000000 00000000 00000000 00000000

   00000000 00000000 00000000 00000000

   00000000 00000000 00000000 00000028

Le message dûment complété contiendra n mots de 16* pour certaines valeurs de n > 0. Ce message est considéré comme une séquence de n blocs M_1, M_2, ... , M_n, dans laquelle chaque bloc M_i contient 16 mots, le bloc M_1 contenant les premiers caractères (ou bits) du message.

II.5 Fonctions utilisées

Une séquence de fonctions logiques f_0, f_1, ... , f_{79} est utilisée dans l'algorithme SHA-1. Chaque fonction f_t, $0 \leq t \leq 79$, agit sur trois mots de 32 bits (B, C, D), produisant comme résultat un mot de 32 bits. La fonction $f_t(B,C,D)$ est définie comme suit: pour les mots B, C, D,

$f_t(B,C,D)$ = (B ET C) OU ((PAS B) ET D) $(0 \leq t \leq 19)$

$f_t(B,C,D)$ = B OU exclusif C OU exclusif D $(20 \leq t \leq 39)$

$f_t(B,C,D)$ = (B ET C) OU (B ET D) OU (C ET D) $(40 \leq t \leq 59)$

$f_t(B,C,D)$ = B OU exclusif C OU exclusif D $(60 \leq t \leq 79)$.

II.6 Constantes utilisées

Une séquence de mots constants K(0), K(1), ... , K(79) est utilisée dans l'algorithme SHA-1. Sous forme hexadécimale, ces mots sont exprimés comme suit:

$K = 5A827999$ ($0 \leq t \leq 19$)

$K_t = 6ED9EBA1$ ($20 \leq t \leq 39$)

$K_t = 8F1BBCDC$ ($40 \leq t \leq 59$)

$K_t = CA62C1D6$ ($60 \leq t \leq 79$)

II.7 Calcul du résumé de message

Le résumé de message est calculé à l'aide du message final dûment complété. Le calcul utilise deux tampons, contenant chacun 5 mots de 32 bits, et une séquence de 80 mots de 32 bits. Les mots du premier tampon de 5 mots sont étiquetés A, B, C, D, E. Les mots du second tampon de 5 mots sont étiquetés H_0, H_1, H_2, H_3, H_4. Les mots de la séquence de 80 mots sont étiquetés W_0, W_1, ... , W_{79}. Un tampon d'un seul mot (TEMP) est également utilisé.

Pour établir le résumé de message, on traite séquentiellement les blocs M_1, M_2, ... , M_n de 16 mots définis dans le paragraphe 4. Le traitement de chaque bloc M_i comporte 80 étapes.

Préalablement au traitement de tout bloc, les mots $\{H_i\}$ sont initialisés comme suit: sous forme hexadécimale,

$H_0 = 67452301$

$H_1 = EFCDAB89$

$H_2 = 98BADCFE$

$H_3 = 10325476$

$H_4 = C3D2E1F0$

On traite alors les blocs M_1, M_2, ... , M_n en procédant comme suit pour le bloc M_i:

a) On divise le bloc M_i en 16 mots W_0, W_1, ..., W_{15}, W_0 étant le mot de gauche.

b) Pour t = 16 à 79, on admet que $W_t = S^1(W_{t-3}$ OU exclusif W_{t-8} OU exclusif W_{t-14} OU exclusif $W_{t-16})$.

c) On admet que $A = H_0$, $B = H_1$, $C = H_2$, $D = H_3$, $E = H_4$.

d) Pour t = 0 à 79, on fait en sorte que:

$TEMP = S^5(A) + f_t(B,C,D) + E + Wt + K_t$;

$E = D$; $D = C$; $C = S^{30}(B)$; $B = A$; $A = TEMP$;

e) On admet que $H_0 = H_0 + A$, $H_1 = H_1 + B$, $H_2 = H_2 + C$, $H_3 = H_3 + D$, $H_4 = H_4 + E$.

Après traitement du bloc M_n, le résumé de message est la chaîne de 160 bits représentée par les 5 mots:

$H_0\ H_1\ H_2\ H_3\ H_4$.

II.8 Autre méthode de calcul

Les éléments ci-dessus supposent que la séquence W_0, ... , W_{79} soit implémentée sous la forme d'un ensemble de 80 mots de 32 bits. Cette forme d'implémentation permet de minimaliser dûment le temps d'exécution, du fait que les adresses de W_{t-3}, ... , W_{t-16} de l'étape b) sont faciles à calculer. Si on privilégie l'espace, on peut aussi considérer {Wt} comme une file d'attente circulaire, pouvant être implémentée dans un ensemble de 16 mots de 32 bits W[0], ... , W[15]. Dans ce cas, on admet que MASK = 0000000F, sous forme hexadécimale.

On traite ensuite le bloc M_i comme suit:

a) On divise M_i en 16 mots W[0], ... , W[15], W[0] étant le mot de gauche.

b) On admet que $A = H_0$, $B = H_1$, $C = H_2$, $D = H_3$, $E = H_4$.

c) Pour t = 0 à 79, on fait en sorte que:

s = t ^ MASK;

si ($t \geq 16$) $W[s] = S^1$(W[(s + 13) ^ MASK] OU exclusif W[(s + 8) ET MASK] OU exclusif W[(s + 2) ^ MASK] OU exclusif W[s]);

$TEMP = S^5(A) + f_t(B,C,D) + E + W[s] + K_t$;

$E = D$; $D = C$; $C = S^{30}(B)$; $B = A$; $A = TEMP$;

d) On admet que $H_0 = H_0 + A$, $H_1 = H_1 + B$, $H_2 = H_2 + C$, $H_3 = H_3 + D$, $H_4 = H_4 + E$.

II.9 Comparaison des méthodes

Les méthodes décrites en II.7 et II.8 donnent le même résumé de message. Bien que la méthode décrite dans II.8 permette d'économiser 64 mots de 32 bits de mémoire, le recours à cette méthode est susceptible d'allonger le temps d'exécution du fait de la complexité accrue des calculs des adresses pour le mot {W[t]} au cours de l'étape c). D'autres méthodes de calcul qui donnent des résultats identiques peuvent être implémentées conformément à la norme.

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

Genève, 2001