UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

UIT-T Q.813

(06/98)

SECTEUR DE LA NORMALISATION
DES TÉLÉCOMMUNICATIONS
DE L'UIT

SÉRIE Q: COMMUTATION ET SIGNALISATION

Spécifications du système de signalisation n° 7 – Interface Q3

Elément de service d'application des transformations de sécurité pour l'élément de service d'opérations distantes (STASE-ROSE)

Recommandation UIT-T Q.813

(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
Généralités	Q.850–Q.919
Couche Liaison de données	Q.920–Q.929
Couche Réseau	Q.930–Q.939
Gestion utilisateur-réseau	Q.940–Q.949
Description d'étape 3 des services complémentaires utilisant le système DSS 1	Q.950–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.1999
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.813

ELÉMENT DE SERVICE D'APPLICATION DES TRANSFORMATIONS DE SÉCURITÉ POUR L'ÉLÉMENT DE SERVICE D'OPÉRATIONS DISTANTES (STASE-ROSE)

Résumé

La présente Recommandation fournit des spécifications pour la prise en charge de transformations de sécurité, telles que le chiffrement, le hachage, le scellé et la signature, en se concentrant sur l'unité de données protocolaires (PDU, *protocol unit data*) de l'élément de service d'opérations distantes (ROSE, *remote operations service element*) considérée comme un tout. Les transformations de sécurité sont utilisées pour fournir divers services de sécurité tels que l'authentification, la confidentialité, l'intégrité et la non-répudiation. La présente Recommandation décrit une démarche pour la fourniture de transformations de sécurité qui est mise en œuvre au niveau de la couche Application et ne fait appel à aucune fonctionnalité spécifique de la sécurité dans l'une quelconque des couches sous-jacentes de la pile OSI.

Source

La Recommandation UIT-T Q.813, élaborée par la Commission d'études 4 (1997-2000) de l'UIT-T, a été approuvée le 26 juin 1998 selon la procédure définie dans la Résolution n° 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'études à traiter par les Commissions d'études de l'UIT-T, lesquelles élaborent en retour des Recommandations sur ces thèmes.

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

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

NOTE

Dans la présente Recommandation, 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 Q.813

ELÉMENT DE SERVICE D'APPLICATION DES TRANSFORMATIONS DE SÉCURITÉ POUR L'ÉLÉMENT DE SERVICE D'OPÉRATIONS DISTANTES (STASE-ROSE)

(Genève, 1998)

1 Domaine d'application, but et utilisation

1.1 Domaine d'application

Les transformations de sécurité (ST, *security transformation*) sont utilisées pour fournir divers services de sécurité tels que l'authentification d'entité homologue, l'authentification de l'origine, la confidentialité, l'intégrité et la non-répudiation des données. Les transformations de sécurité englobent le chiffrement, le hachage, les scellés numériques et les signatures numériques.

La présente Recommandation prend en charge des services de sécurité pour des unités PDU de l'élément ROSE au sein de la couche Application. Elle est indépendante de la pile de protocoles de communication sous-jacente. La présente Recommandation définit un nouvel élément de service d'application (ASE, *application service element*) appelé "élément de service d'application de transformations de sécurité pour l'élément ROSE" (STASE-ROSE), qui est localisé dans la pile de protocoles OSI entre l'élément ROSE et la couche Présentation. La présente Recommandation fournit une démarche pour la réalisation de transformations de sécurité (ST) qui n'impose aucune contrainte à l'une quelconque des six couches inférieures de la pile de communication. Elle diffère en cela de certaines méthodes [par exemple, la sécurité générique de couche supérieure (GULS, *generic upper layers security*)] qui prennent en charge des transformations de sécurité au moyen d'une fonctionnalité incorporée dans la pile de communication au niveau de la couche Présentation.

La présente Recommandation fournit en outre une fonctionnalité d'authentification d'entité homologue au moment de l'établissement de l'association, une fonctionnalité de négociation des paramètres de sécurité (tels que les algorithmes de sécurité) qui seront utilisés durant l'association ainsi qu'une fonctionnalité de mise à jour dynamique durant l'association de paramètres de sécurité utilisés dans les unités de données protocolaires individuelles.

La méthode décrite dans la présente Recommandation peut être adaptée pour des entités ASE autres que l'élément ROSE, qui interagissent directement avec la couche Présentation. La présente Recommandation se concentre toutefois sur l'élément ROSE et ne traite aucune extension ou généralisation éventuelle.

La réalisation effective des transformations de sécurité (par exemple la création et la vérification de signatures numériques) est un problème local qui est en dehors du domaine d'application de la présente Recommandation. En particulier, l'utilisation d'un module générique de sécurité tel que l'interface de programmation du service générique de sécurité (GSS-API, *generic security service – application programming interface*) pour la réalisation de transformations de sécurité est un problème local. La présente Recommandation ne prescrit pas l'utilisation de l'interface API du service GSS, mais fournit toutefois le cadre général nécessaire à son utilisation avec l'élément STASE-ROSE (voir l'Appendice III).

La gestion des clés est une composante importante d'une infrastructure de sécurité. La présente Recommandation prend en charge l'échange d'informations liées aux clés de chiffrement. Un cadre général pour la gestion des clés est toutefois en dehors du domaine d'application de la présente Recommandation.

1.2 But

La présente Recommandation a pour but de protéger des unités PDU ROSE dans leur intégralité.

La Recommandation Q.812 spécifie la fonctionnalité de transfert et d'administration de fichier (FTAM, *file transfer administration and management*), l'élément de service d'application de gestion d'informations communes (CMISE, *common information management application service element*) et l'annuaire X.500 dans la couche Application pour les interfaces Q3 et X du réseau de gestion des télécommunications (RGT). La présente Recommandation traite de la sécurité des unités de données protocolaires (PDU) de l'élément ROSE. Bien qu'elle soit guidée par le besoin de sécuriser les interactions et les échanges de message du RGT, la présente Recommandation peut toutefois fournir des fonctionnalités de sécurité pour toute application qui utilise l'élément ROSE.

1.3 Utilisation

La présente Recommandation est utilisable par des applications qui emploient l'élément ROSE, telles que des applications utilisateur qui mettent en œuvre l'élément CMISE ou l'annuaire X.500. Elle a pour objectif essentiel la protection d'unités PDU du protocole CMIP. Etant donné que le protocole CMIP est fondé sur la version 1998 de l'élément ROSE (voir les Recommandations X.219 et X.229), la présente Recommandation est aussi axée sur cette version, plutôt que sur celle de 1994 (voir les Recommandations X.880, X.881 et X.882). En conséquence, il se peut que la présente Recommandation ne s'applique pas à la version actuelle de la Recommandation X.500 qui est fondée sur la version de 1994 de l'élément ROSE.

2 Références

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.

2.1 Références normatives

– Recommandation UIT-T M.3010 (1996), *Principes des réseaux de gestion des télécommunications.*

– Recommandation UIT-T Q.811 (1997), *Profils des protocoles des couches inférieures pour les interfaces Q3 et X*.

– Recommandation UIT-T Q.812 (1997), *Profils des protocoles des couches supérieures pour les interfaces Q3 et X*.

– Recommandation UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base: le modèle de référence de base.*

– Recommandation X.208 du CCITT (1988), *Spécification de la syntaxe abstraite numéro un (ASN.1).*

– Recommandation UIT-T X.210 (1993) | ISO/CEI 10731:1994, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de référence de base: conventions pour la définition des services de l'interconnexion de systèmes ouverts.*

– Recommandation UIT-T X.217 (1995) | ISO/CEI 8649:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Définition de service applicable à l'élément de service de contrôle d'association*.

– Recommandation X.219 du CCITT (1988), *Opérations distantes: modèle, notation et définition du service*.

– Recommandation UIT-T X.227 (1995) | ISO/CEI 8650-1:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Protocole en mode connexion applicable à l'élément de service de contrôle d'association: spécification du protocole.*

– Recommandation X.229 du CCITT (1988), *Opérations distantes: spécification du protocole*.

– Recommandation UIT-T X.500 (1997) | ISO/CEI 9594-1:1997, *Technologies de l'information – Interconnexion des systèmes ouverts – L'annuaire: vue d'ensemble des concepts, modèles et services.*

– Recommandation UIT-T X.509 (1997) | ISO/CEI 9594-8:1997, *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.*

– Recommandation UIT-T X.710 (1997) | ISO/CEI 9595:1998, *Technologies de l'information – Interconnexion des systèmes ouverts – Service commun de transfert d'informations de gestion*.

– Recommandation UIT-T X.711 (1997) | ISO/CEI 9596-1:1998, *Technologies de l'information – Interconnexion des systèmes ouverts – Spécification du protocole commun de transfert d'informations de gestion*.

– ISO/CEI 9979:1991, *Techniques cryptographiques – Procédures pour l'enregistrement des algorithmes cryptographiques*.

2.2 Références informatives

– Recommandation UIT-T X.803 (1994) | ISO/CEI 10745:1995, *Technologies de l'information – Interconnexion des systèmes ouverts – Modèle de sécurité pour les couches supérieures.*

– Recommandation UIT-T X.830 (1995) | ISO/CEI 11586-1:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Sécurité générique des couches supérieures: aperçu général, modèles et notation.*

– Recommandation UIT-T X.831 (1995) | ISO/CEI 11586-2:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Sécurité générique des couches supérieures: définition du service assuré par l'élément de service d'échange de sécurité.*

– Recommandation UIT-T X.832 (1995) | ISO/CEI 11586-3:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Sécurité générique des couches supérieures: spécification du protocole d'élément de service d'échange de sécurité.*

– Recommandation UIT-T X.833 (1995) | ISO/CEI 11586-4:1996, *Technologies de l'information – Interconnexion des systèmes ouverts – Sécurité générique des couches supérieures: spécification de la syntaxe de protection du transfert.*

– ISO/CEI 9798-3:1993, *Technologies de l'information – Techniques de sécurité – Mécanisme d'authentification d'entité – Partie 3: authentification d'entité utilisant un algorithme à clé publique*.

– ISO/CEI 11770-1:1996, *Technologies de l'information – Techniques de sécurité – Gestion des clés – Partie 1: cadre général.*

– ANSI X3.92-1981, Data Encryption Algorithm.

– ANSI X3.106-1983, Data Encryption Algorithm – Modes of Operation.

– NBS FIPS PUB 46-1, Data Encryption Standard, *National Bureau of Standards*, US Department of Commerce, janvier 1988.

– NBS FIPS PUB 74, Guidelines for Implementing and Using the NBS Data Encryption Standard, *National Bureau of Standards*, US Department of Commerce, avril 1981.

– NBS FIPS PUB 81, DES Modes of Operation, *National Bureau of Standards*, US Department of Commerce, décembre 1980.

– NIST FIPS PUB 46-2, Data Encryption Standard, *National Institute of Standards and Technology*, US Department of Commerce, décembre 1993.

– NIST FIPS PUB 180-1, Secure Hash Standard, *National Institute of Standards and Technology*, US Department of Commerce, mai 1994.

– NIST FIPS PUB 186, Digital Signature Standard, *National Institute of Standards and Technology*, US Department of Commerce, mai 1995.

– IETF RFC 2078, Generic Security Service Application Program Interface, Version 2, *Internet Engineering Task Force*, janvier 1997.

3 Définitions

La présente Recommandation définit les termes suivants:

3.1 entité d'application initiant l'association; initiateur de l'association: entité d'application qui prend l'initiative de l'association d'application.

3.2 entité d'application qui répond à l'association; **répondeur de l'association**: entité d'application qui répond à l'initiation d'une association d'application faite par une autre entité d'application.

3.3 entité d'application émettrice; émetteur: entité d'application qui émet l'unité APDU à destination de l'entité d'application réceptrice.

3.4 entité d'application réceptrice; récepteur: entité d'application qui reçoit l'unité APDU en provenance de l'entité d'application émettrice.

3.5 **demandeur**: entité d'application qui émet une primitive de transfert STASE-ROSE.

3.6 **accepteur**: entité d'application qui reçoit la primitive d'indication.

3.7 **élément STASE-ROSE**: élément de service d'application, localisé entre la couche de présentation OSI et l'élément ROSE, qui fournit les transformations nécessaires au transfert sécurisé d'unités PDU ROSE.

3.8 **transfert sécurisé**: mécanisme qui fournit d'une manière sécurisée un transfert d'unités de données protocolaires d'application (APDU) entre systèmes ouverts.

3.9 **utilisateur STASE-ROSE; utilisateur SR**: élément de service d'application qui utilise les services de l'élément STASE-ROSE. L'élément de service d'opérations distantes (ROSE) est le seul utilisateur des services STASE-ROSE.

3.10 **fournisseur STASE-ROSE; fournisseur SR**: fournisseur de l'élément de service d'application de transformations de sécurité pour l'élément ROSE.

3.11 **fournisseur ACSE**: fournisseur de l'élément de commande de service d'association.

La présente Recommandation utilise les définitions de services de sécurité et de mécanismes de sécurité telles qu'elles sont spécifiées dans les Recommandations X.800 et M.3016.

4 Abréviations

La présente Recommandation utilise les abréviations suivantes:

AARE	réponse d'association pour l'élément ACSE (*ACSE association response*)
AARQ	demande d'association pour l'élément ACSE (*ACSE association request*)
ACSE	élément de commande d'association (*association control service element*)
AE	entité d'application (*application entity*)
APDU	unité de données protocolaires d'application (*application protocol data unit*)
API	interface de programmation d'application (*application programming interface*)
ASCII	code normalisé américain pour l'échange d'informations (*american standard code for information interchange*)
ASN.1	notation de syntaxe abstraite n° 1 (*abstract syntax notation one*)
BER	règles de codage de base (*basic encoding rules*)
CBC	chaînage de blocs de chiffre (*cipher block chaining*)
CEI	Commission électrotechnique internationale
CMIP	protocole d'informations communes de gestion (*common management information protocol*)
CMISE	élément de service d'informations communes de gestion (*common management information service element*)
DER	règles de codage distinctives (*distinguished encoding rules*)
DES	norme de chiffrement de données (*digital encryption standard*)
EBCDIC	code d'échange binaire étendu codé en décimal (*extended binary coded decimal interchange code*)

FIPS PUB	publication de normes fédérales de traitement de l'information (*federal information processing standards publication*)
FTAM	administration et gestion de transfert de fichier (*file transfer administration and management*)
GSS-API	interface de programmation d'application du service générique de sécurité (*generic security service – application programming interface*)
GULS	sécurité générique des couches supérieures (*generic upper layers security*)
IETF	groupe de travail d'ingénierie Internet (*Internet engineering task force*)
IS	Norme internationale (*international standard*)
ISO	Organisation internationale pour la normalisation (*International organization for standardization*)
MAC	code d'authentification de message (*message authentication code*)
MD	résumé de message (*message digest*)
NBS	Bureau national de normalisation (*national bureau of standards*)
NIST	Institut national pour la normalisation et la technologie (*national institute for standards and technology*)
PDU	unité de données protocolaires (*protocol data unit*)
PKCS	norme de chiffrement par clé publique (*public key cryptography standard*)
QoP	qualité de protection (*quality of protection*)
RFC	demande de commentaires (norme Internet) (*request for comments*)
ROSE	élément de service d'opérations distantes (*remote operations service element*)
RSA	Rivest, Shamir et Adelman
SR	élément STASE-ROSE
ST	transformation de sécurité (s*ecurity transformation*)
STASE-ROSE	élément ROSE de service d'application de transformations de sécurité (*security transformations application service element – ROSE*)
UIT-T	Union internationale des télécommunications – Secteur de normalisation des télécommunications

5 Aperçu général

Le présent paragraphe fournit un aperçu général à haut niveau du service STASE-ROSE. Elle se concentre sur les actions de l'élément STASE-ROSE, en laissant aux sous-paragraphes suivants la description plus précise de la manière dont l'élément STASE-ROSE effectue les actions.

L'élément STASE-ROSE est un élément de service d'application, localisé entre la couche Présentation OSI et l'élément ROSE, qui fournit les transformations nécessaires à un transfert sécurisé d'unités PDU ROSE. L'élément STASE-ROSE fournit en outre un moyen d'échange d'informations sur la sécurité fournie. L'élément STASE-ROSE est invoqué par l'élément ROSE du côté émetteur et fournit une indication à l'élément ROSE du côté récepteur. La demande et l'indication contiennent toutes deux l'unité PDU ROSE protégée ainsi que des informations optionnelles concernant le type de sécurité fourni.

Les caractéristiques des transformations de sécurité et de l'échange d'informations de sécurité de l'élément STASE-ROSE sont traitées ci-après.

5.1 Transformations de sécurité

L'élément STASE-ROSE protège les unités PDU ROSE en appliquant des transformations de sécurité (ST, *security transformation*) sélectionnées à des unités PDU ROSE codées au moyen des règles de codage distinctives (DER, *distinguished encoding rules*). L'élément STASE-ROSE prend en charge en particulier les transformations de sécurité suivantes:

- **confidentiel** (*confidential*): l'unité PDU ROSE avec codage sous forme DER est chiffrée à des fins de protection de la confidentialité en utilisant un algorithme de chiffrement par clé symétrique;
- **chiffré public** (*public enciphered*): l'unité PDU ROSE avec codage sous forme DER est chiffrée à des fins de protection de la confidentialité en utilisant un algorithme de chiffrement par clé publique;
- **haché** (*hashed*): l'élément STASE-ROSE calcule un code d'identification de message (MAC, *message authentication code*) basé sur un résidu de hachage de l'unité PDU ROSE avec codage sous forme DER ainsi qu'un mot de passe secret et ajoute le résultat à l'unité PDU ROSE à des fins de protection de l'intégrité;
- **scellé** (*sealed*): l'élément STASE-ROSE calcule le scellé numérique de l'unité PDU ROSE avec codage sous forme DER et ajoute le résultat à l'unité PDU ROSE à des fins de protection de l'intégrité;
- **signé** (*signed*): l'élément STASE-ROSE calcule la signature numérique de l'unité PDU ROSE avec codage sous forme DER et ajoute le résultat à l'unité PDU ROSE à des fins de protection contre la répudiation;
- **confidentiel signé** (*confidential signed*): l'élément STASE-ROSE calcule la signature numérique de l'unité PDU ROSE avec codage sous forme DER et ajoute le résultat à l'unité PDU ROSE chiffrée (voir "confidentiel" ci-dessus) à des fins de protection de la confidentialité et contre la répudiation;
- **confidentiel haché** (*confidential hashed*): l'élément STASE-ROSE calcule le code MAC de l'unité PDU ROSE avec codage sous forme DER et ajoute le résultat à la fin de l'unité PDU ROSE chiffrée (voir "confidentiel" ci-dessus) à des fins de protection de la confidentialité et de l'intégrité;
- **confidentiel scellé** (*confidential sealed*): l'élément STASE-ROSE calcule le scellé numérique de l'unité PDU ROSE avec codage sous forme DER et ajoute le résultat à l'unité PDU ROSE chiffrée (voir "confidentiel" ci-dessus) à des fins de protection de la confidentialité et de l'intégrité.

L'élément STASE-ROSE peut également retransmettre des unités PDU ROSE en **clair** sans codage et transformations de sécurité.

5.2 Echange des informations de sécurité

Les messages suivants spécifient quelles sont les transformations de sécurité énumérées ci-dessus qui seront utilisées pour protéger l'unité PDU ROSE lors de son échange entre des éléments STASE-ROSE ou entre un élément ROSE et un élément STASE-ROSE:

- l'élément ROSE invoque l'élément STASE-ROSE du côté origine;
- l'élément STASE-ROSE d'origine émet une unité PDU STASE-ROSE à destination de l'élément STASE-ROSE du côté récepteur;

• l'élément STASE-ROSE du côté récepteur fournit une indication à l'élément ROSE.

La connaissance de la transformation de sécurité utilisée est nécessaire mais non suffisante pour une communication correcte. Les deux côtés doivent connaître également les algorithmes utilisés ainsi que les valeurs de tous les paramètres utilisés (par exemple: clés de chiffrement, vecteurs d'initialisation). La présente Recommandation fournit un certain nombre de valeurs et de mécanismes par défaut qui ne nécessitent que le strict minimum d'échanges d'informations en relation avec la sécurité. Elle fournit également des fonctionnalités de négociation, au moment de l'établissement d'une association, des algorithmes pris en charge, avec la possibilité de modifier et de spécifier de telles informations pour chaque unité PDU ROSE.

5.2.1 Valeurs par défaut des informations de sécurité

L'utilisation de la capacité de négociation de l'élément STASE-ROSE est optionnelle. Si deux entités communicantes n'utilisent pas cette capacité, elles doivent avoir conclu un accord concernant un ensemble de paramètres de sécurité, tels que des algorithmes de sécurité, qui seront utilisés durant l'association. Les deux participants peuvent conclure un accord pour tout ensemble de valeurs de tels paramètres en utilisant des moyens qui sont en dehors du domaine d'application de la présente Recommandation.

Les conventions, algorithmes de sécurité et mécanismes de sécurité suivants seront utilisés, sauf accord contraire conclu entre les entités communiquantes:

– l'algorithme de chiffrement par défaut pour un chiffrement symétrique sera la norme de chiffrement numérique (DES, *digital encryption standard*) dans le mode de chaînage de blocs de chiffre (CBC, *cipher block chaining*);

– si un chiffrement DES triple est nécessaire, l'algorithme par défaut sera l'algorithme chiffrement-déchiffrement-chiffrement (EDE, *encryption decryption encryption*) dans le mode CBC avec feed-back extérieur utilisant trois clés DES différentes;

– si aucun vecteur d'initialisation (IV, *initialization vector*) n'est spécifié, le vecteur d'initialisation utilisé pour la première association se constituera d'une valeur de 64 bits tous nuls; chaque vecteur d'initialisation suivant se constituera des 8 premiers octets de l'unité PDU ROSE chiffrée précédemment;

– l'algorithme de chiffrement par clé publique sera l'algorithme RSA[1, 2];

– l'algorithme de hachage par défaut sera l'algorithme MD5[3];

– le code MAC par défaut (pour un hachage avec clé) sera le code HMAC[4];

– le scellé par défaut sera le hachage MD5 de l'unité PDU ROSE codée selon les règles DER et chiffrée au moyen de l'algorithme DES;

– la signature par défaut sera le hachage MD5 de l'unité PDU ROSE codée selon les règles DER et chiffrée au moyen de l'algorithme RSA avec la clé privée de l'utilisateur;

– l'authentification de l'entité homologue se fera au moment de l'établissement de l'association. L'unité fonctionnelle (FU, *functional unit*) optionnelle d'authentification de l'élément ACSE

---

1 RIVEST (R.), SHAMIR (A.) et ADELMAN (L. M.): A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Communications of the ACM*, Version 21, N. 2 p. 120-126, février 1978.

2 RSA Data Security Inc., PKCS No. 1: RSA Encryption Standard, Version 1.5, novembre 1993.

3 RIVEST (R.), IETF RFC 1319: The MD5 Message Digest Algorithm, avril 1992.

4 KRAWCZYK (H.), BELLARE (M.), CANETTI (R.), IETF RFC 2104, HMAC: Keyed-Hashing for Message Authentication, février 1997.

sera utilisée. Les informations d'authentification seront véhiculées dans les champs "valeur d'authentification appelante" et "valeur d'authentification de réponse" des unités PDU ACSE de demande AARQ et de réponse AARE respectivement. Les chaînes de bit des champs "prescription d'élément ACSE du demandeur" et "prescription d'élément ACSE du répondeur" de l'unité fonctionnelle d'authentification seront positionnées de manière à contenir l'unité fonctionnelle d'authentification. Les champs "valeur d'authentification appelante" et "valeur d'authentification de réponse" sont du type "valeur d'authentification" défini dans l'ISO 8650 sous la forme d'une expression CHOICE. L'expression CHOICE pour la "valeur d'authentification" sera du type EXTERNAL. Le contexte de présentation contiendra une référence à la syntaxe abstraite qui est utilisée pour le type EXTERNAL. Il n'est pas nécessaire d'utiliser le champ optionnel "nom de mécanisme" de l'unité fonctionnelle d'authentification des unités PDU ACSE lorsque les valeurs et les conventions par défaut sont utilisées;

– si elle est requise, l'authentification de l'entité homologue avec chiffrement par clé publique se constituera des informations suivantes:

  • identificateur non ambigu de l'émetteur;
  • identificateur non ambigu du récepteur;
  • horodatage utilisant un temps généralisé;
  • clé de chiffrement symétrique optionnelle obtenue par chiffrement utilisant la clé publique du récepteur et qui sera utilisée par l'émetteur durant l'association;
  • signature numérique des champs précédents, obtenue en utilisant la clé privée de l'émetteur;
  • certificat optionnel de la clé publique de l'émetteur.

La signature numérique sera calculée en utilisant l'algorithme MD5 pour le hachage et l'algorithme RSA pour le chiffrement par clé publique, sauf accord contraire entre les entités communicantes conclu par des moyens qui sont en dehors du domaine d'application de la présente Recommandation. La syntaxe de cet élément d'authentification est donnée au 5.2.1.1. Les clés de chiffrement symétriques optionnelles, obtenues par chiffrement au moyen de la clé publique, peuvent être différentes dans les messages des entités AARQ et AARE, ce qui permet à l'initiateur et au répondeur de l'association d'utiliser des clés différentes durant l'association.

Il est possible d'utiliser des horodatages pour les éléments d'authentification proposés dans la présente Recommandation, ainsi que dans d'autres parties de la présente Recommandation. Les horloges système peuvent être retardées si leur fonctionnement est trop rapide, ou elles peuvent prendre du retard à la suite d'un mauvais fonctionnement. La présente Recommandation prescrit que les horodatages consécutifs produits par un système auront des valeurs uniformément croissantes, même si de telles perturbations se manifestent. L'Appendice I présente une illustration d'une génération possible d'un tel temps uniformément croissant.

La présente Recommandation ne précise pas lequel des deux éléments d'authentification doit être utilisé. Les entités communicantes peuvent déterminer l'élément d'authentification qu'elles utiliseront en concluant un accord par des moyens qui sont en dehors du domaine d'application de la présente Recommandation.

Bien que la présente Recommandation définisse deux éléments d'authentification, les parties communicantes peuvent convenir d'en utiliser un autre, auquel cas il faudra spécifier la syntaxe ASN.1 applicable à cet élément. Il faut aussi attribuer et enregistrer un nom de syntaxe abstraite applicable à l'élément d'authentification qui servira lors de la négociation de la couche Application.

Les entités communicantes peuvent conclure un accord, portant sur un ensemble différent de valeurs par défaut, par des moyens qui sont en dehors du domaine d'application de la présente Recommandation.

5.2.1.1 Syntaxe abstraite pour l'élément d'authentification par clé publique

Le module suivant d'authentification par clé publique doit être véhiculé dans le champ "valeur d'authentification" de l'unité fonctionnelle d'authentification de l'élément ACSE lorsqu'une authentification de l'entité homologue avec chiffrement par clé publique est prescrit. Ce module prend en charge les deux cas suivants:

– tous les champs de l'élément d'authentification sont spécifiés de manière explicite;

– les deux entités communicantes utilisent l'interface (GSS-API, *generic security service – application programming interface*) et les informations d'authentification sont transportées sous la forme d'une chaîne d'octets qui est interprétée localement par l'interface GSS-API.

```
STASE-ROSE-Authentication-value {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0)
abstractSyntax(1) stase-authentication-value(0) }
DEFINITIONS IMPLICIT TAGS ::= BEGIN
-- EXPORTE tout

IMPORTS

SenderId, ReceiverId, Signature, SignatureCertificate
FROM Secure-Remote-Operations-APDUs {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0) stase-
data(2)};

Authentication-value ::= CHOICE {
                explicit                        [0] ExplicitAuthenticator,
                 gssAuthenticator               [1] GssAuthenticator,
        -- utilisé uniquement si les deux entités communicantes utilisent l'interface GSS-API
                ...
                 }

ExplicitAuthenticator ::= SEQUENCE {
                                        senderId        [0] SenderId,
                                        receiverId      [1] ReceiverId,
                                        time            [3] GeneralizedTime,
                                        encryptedSymmetricKey [4] INTEGER    OPTIONAL,
        -- clé de chiffrement symétrique chiffrée avec la clé publique du récepteur
                                        signature       [5] Signature,
        -- signature par l'émetteur des champs précédents codés sous forme de caractères ASCII
                                        certificate     [6] SignatureCertificate        OPTIONAL
        -- certificat de la clé publique de l'émetteur pour la clé utilisée pour la signature
        }

GssAuthenticator ::= SEQUENCE {
                        gssMechanism            [0] OBJECT IDENTIFIER OPTIONAL,
                        gssInitialContextToken  [1] OCTET STRING
                        }

END
```

5.2.1.2 Nom de syntaxe abstraite

La présente Recommandation attribue la valeur d'identificateur d'objet ASN.1 suivante:

{itu-t recommendation q(17) q813(813) stase(1) abstractSyntax(1) stase-authentication-value(0)}

comme nom de syntaxe abstraite pour l'ensemble des valeurs de données de présentation, chacune de ces dernières étant une valeur de type ASN.1

STASE-ROSE-Authentication-value.Authentication-value.

La valeur de descripteur d'objet correspondante sera "STASE-ROSE-Authenticator".

5.2.2 Négociation d'algorithmes de sécurité

Les valeurs par défaut, spécifiées par la présente Recommandation ou ayant fait l'accord entre les entités communicantes, peuvent être remplacées de manière dynamique comme décrit ci-dessous.

Les entités communicantes peuvent négocier au moment de l'établissement de l'association les algorithmes qu'elles prendront en charge durant l'association. L'entité qui est à l'origine de l'association peut placer de manière optionnelle les informations suivantes dans le champ utilisateur de demande d'association de l'élément ACSE:

- un ensemble d'algorithmes de chiffrement symétrique acceptables;
- un ensemble d'algorithmes de chiffrement par clé publique acceptables;
- un ensemble d'algorithmes de hachage acceptables;
- un ensemble d'algorithmes de hachage avec clé (MAC) acceptables;
- un ensemble d'algorithmes de scellé numérique acceptables;
- un ensemble d'algorithmes de signature numérique acceptables.

La réponse AARE contiendra un ensemble d'algorithmes acceptables qui constitue un sous-ensemble de celui figurant dans la demande AARQ. Si cette capacité optionnelle est utilisée par l'initiateur de l'association dans la demande AARQ, elle doit alors également être utilisée par le répondeur de l'association. Une réponse d'association qui n'utilise pas la capacité optionnelle de négociation est équivalente à une réponse contenant des valeurs nulles (c'est-à-dire un ensemble vide). Si l'un quelconque des ensembles énumérés précédemment ne figure pas dans la demande AARQ, alors seule la valeur par défaut de l'algorithme correspondant sera prise en charge. Si l'un des ensembles énumérés précédemment figure dans la demande AARQ, il doit alors figurer également dans la réponse AARE, faute de quoi son absence est équivalente à l'ensemble vide. Si l'un quelconque des ensembles énumérés précédemment figure dans la réponse AARE, mais qu'il est vide, alors aucun algorithme ne peut être utilisé pour la transformation de sécurité correspondante, de sorte que cette transformation de sécurité ne peut pas être utilisée durant l'association. Si l'un quelconque des ensembles énumérés précédemment contient un élément et un seul dans la réponse AARE, cet élément désigne alors la valeur par défaut de la transformation de sécurité correspondante pour la durée de l'association. Si l'un quelconque des ensembles énumérés précédemment contient plus d'un élément dans la réponse AARE et si l'un de ces éléments correspond à la valeur par défaut de la transformation de sécurité correspondante (telle qu'elle est spécifiée dans la présente Recommandation ou ayant fait d'une autre manière l'objet d'un accord entre deux entités communicantes), cet élément désigne alors la valeur par défaut de la transformation de sécurité correspondante pour la durée de l'association. Une entité qui reçoit un message peut mettre fin à la session si elle n'est pas d'accord avec le choix des algorithmes. Une erreur est détectée si l'un quelconque des ensembles énumérés précédemment figure dans la réponse AARE et contient des éléments qui ne figurent pas dans la demande AARQ. Le Tableau 5-1 résume le processus de négociation d'algorithme de sécurité qui est illustré dans l'Appendice II.

Tableau 5-1/Q.813 – Algorithmes négociés

<table>
<tr><td colspan="4" rowspan="3">Ensemble d'algorithmes acceptables dans AARE</td><td colspan="4">Ensemble d’algorithmes acceptables dans AARQ</td></tr>
<tr><td colspan="3">Présent</td><td>Absent</td></tr>
<tr><td colspan="2">Non vide</td><td>Vide</td><td rowspan="2"></td></tr>
<tr><td>2 éléments ou plus</td><td>1 élément</td><td>(NULL)</td></tr>
<tr><td rowspan="4">Présent</td><td rowspan="3">Non vide (sous-ensemble de l'ensemble AARQ)</td><td rowspan="2">2 éléments ou plus</td><td>Ne contient pas la valeur par défaut prédéfinie</td><td>Algorithmes de réponse AARE, pas de valeur par défaut</td><td>Erreur</td><td>Erreur</td><td>Erreur</td></tr>
<tr><td>Contient la valeur par défaut prédéfinie</td><td>Algorithmes de réponse AARE avec la valeur par défaut prédéfinie</td><td>Erreur</td><td>Erreur</td><td>Erreur</td></tr>
<tr><td colspan="2">1 élément</td><td>Algorithmes de réponse AARE est la valeur par défaut</td><td>L'algorithme sélectionné est la valeur par défaut</td><td>Erreur</td><td>Erreur</td></tr>
<tr><td colspan="3">Vide (NULL)</td><td>Néant</td><td>Néant</td><td>Néant</td><td>Néant</td></tr>
<tr><td colspan="4">Absent</td><td>Néant</td><td>Néant</td><td>Néant</td><td>Uniquement la valeur par défaut prédéfinie, sinon néant</td></tr>
</table>

L'élément STASE-ROSE prend également en charge, en plus de la négociation d'algorithmes de sécurité, la négociation de divers paramètres de chiffrement:

- identification des clés de chiffrement symétrique pouvant être utilisées;
- identification des clés publiques pouvant être utilisées;
- identification des clés de scellé pouvant être utilisées;
- identification des identificateurs de mot de passe pouvant être utilisés;
- spécification des tailles de clés publiques pouvant être utilisées;
- spécification des clés publiques pouvant être utilisées;
- clé privée de l'émetteur.

La présente Recommandation ne fournit aucune valeur par défaut pour ces paramètres, contrairement à ce qui est le cas pour les algorithmes de chiffrement.

L'élément STASE-ROSE prend également en charge la transmission des paramètres de chiffrement supplémentaires suivants:

- spécification d'un vecteur d'initialisation (pour le chiffrement DES) devant être utilisé;
- spécification des bits de feed-back qui doivent être utilisés pour les modes de feed-back de sortie à k bits ou de feed-back de chiffrement à k bits de l'algorithme DES;

- spécification d'un résumé de clé pour la vérification d'une clé publique;
- spécification d'un numéro de séquence pour le message en cours;
- spécification d'un horodatage pour le message en cours;
- spécification d'une clé symétrique chiffrée au moyen d'un chiffrement par clé symétrique;
- spécification d'une clé symétrique chiffrée au moyen de la clé publique du récepteur;
- spécification d'un identificateur de clé de chiffrement de clé;
- fourniture de certificats X.509 ou d'itinéraires de certification des clés publiques de l'émetteur pouvant être utilisés sans restrictions;
- fourniture de certificats X.509 ou d'itinéraires de certification des clés publiques de l'émetteur pouvant être utilisés uniquement pour le chiffrement;
- fourniture de certificats X.509 ou d'itinéraires de certification des clés publiques de l'émetteur pouvant être utilisés uniquement pour des signatures numériques;
- spécification d'une clé symétrique de session chiffrée au moyen de la clé publique du récepteur et signée au moyen de la clé privée de l'émetteur.

Les valeurs de ces paramètres peuvent être fournies par l'un ou l'autre des participants pendant l'établissement de l'association, mais elles ne peuvent pas faire l'objet d'une négociation. Chaque participant peut, par exemple, envoyer son, ou ses certificats de clé publique à l'autre participant.

Une fois la phase de négociation achevée (c'est-à-dire au moment de l'établissement de l'association), les deux entités ont conclu un accord au sujet des transformations de sécurité et des algorithmes de ces transformations de sécurité qu'elles prendront en charge. Elles ont également conclu un accord, dans certains cas mais pas nécessairement dans tous, au sujet des algorithmes par défaut pour tout ou partie des transformations de sécurité qu'elles ont décidé de prendre en charge. Elles peuvent également avoir conclu un accord portant sur les valeurs de tout ou partie des paramètres de sécurité.

L'élément STASE-ROSE prend en charge la spécification et l'utilisation d'algorithmes différents pour des unités PDU différentes. Il permet en outre à chaque entité communicante d'utiliser des algorithmes différents de ceux utilisés par l'entité homologue pour les mêmes transformations de sécurité. Ceci n'est évidemment pertinent que si plusieurs algorithmes ont fait l'objet d'un accord pour une transformation de sécurité donnée durant la phase de négociation. L'élément STASE-ROSE fournit un certain nombre de règles simples concernant la spécification dynamique d'algorithmes pendant la durée d'une association:

- si un algorithme et un seul a fait l'objet d'un accord pour une transformation de sécurité donnée, l'algorithme pour cette transformation de sécurité ne sera alors pas spécifié une deuxième fois pendant la durée de l'association;
- si plusieurs algorithmes ont fait l'objet d'un accord pour une transformation de sécurité donnée et si un algorithme par défaut a également fait l'objet d'un accord pour cette transformation de sécurité, la spécification de l'algorithme pour cette transformation de sécurité après l'établissement de l'association est alors optionnelle. L'algorithme par défaut sera en vigueur pour la transformation de sécurité après l'établissement de l'association s'il n'est pas spécifié;
- si plusieurs algorithmes ont fait l'objet d'un accord pour une transformation de sécurité donnée et si aucun algorithme par défaut n'a fait l'objet d'un accord pour cette transformation de sécurité, chaque participant de la communication doit alors spécifier l'algorithme qu'il utilise pour cette transformation de sécurité la première fois qu'il émet un message utilisant cette transformation de sécurité. Chaque entité communicante peut spécifier, sans que ceci soit obligatoire, l'algorithme qu'elle utilise pour cette transformation de sécurité dans des messages ultérieurs;

- si plusieurs algorithmes ont fait l'objet d'un accord pour une transformation de sécurité donnée et si aucun algorithme par défaut n'a fait l'objet d'un accord pour cette transformation de sécurité, l'algorithme qui a été utilisé la dernière fois par cette entité pour cette transformation de sécurité est alors l'algorithme par défaut pour cette entité.

La procédure de négociation utilise le paramètre "sélection de paramètres de chiffrement" défini au 5.3. Ce paramètre est transmis à l'élément ACSE sous la forme de données utilisateur et sera véhiculé dans le champ "données utilisateur" des unités PDU ACSE de demande AARQ et de réponse AARE.

5.3 Syntaxe abstraite pour la négociation de paramètres de sécurité

Le module suivant est enregistré pour la négociation de paramètres de sécurité; il doit être utilisé dans le champ "informations utilisateur" de l'élément ACSE.

```
STASE-A-ASSOCIATE-Information {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0) stase-
userinfo(1)}
DEFINITIONS IMPLICIT TAGS ::= BEGIN
-- EXPORTE tout

IMPORTS

SenderId, ReceiverId, Signature, KeyId , PublicKeyCertificate, EncryptionCertificate, SignatureCertificate,
EncryptedAuthenticatedSymmetricKey
FROM Secure-Remote-Operations-APDUs {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0) stase-
data(2)};

EncryptionParametersSelection ::= SET
        {symmetricKeyIds        [0] SET OF KeyId                        OPTIONAL,
        publicKeyIds            [1] SET OF KeyId                        OPTIONAL,
        sealKeyIds              [2] SET OF KeyId                        OPTIONAL,
        signatureKeyIds         [3] SET OF KeyId                        OPTIONAL,
        passwordIds             [4] SET OF KeyId                        OPTIONAL,
        initializationVector    [5] OCTET STRING (SIZE(8))              OPTIONAL,
        feedBackBits            [6] INTEGER (1..63)                     OPTIONAL,
        -- pour le mode de feed-back de sortie à k bits ou
        -- le mode de feed-back de chiffrement à k bits de l'algorithme DES
        symmetricAlgorithms     [7] SET OF OBJECT IDENTIFIER            OPTIONAL,
        publicKeyAlgorithms     [8] SET OF OBJECT IDENTIFIER            OPTIONAL,
        signatureAlgorithms     [9] SET OF OBJECT IDENTIFIER            OPTIONAL,
        sealAlgorithms          [10] SET OF OBJECT IDENTIFIER           OPTIONAL,
        hashAlgorithms          [11] SET OF OBJECT IDENTIFIER           OPTIONAL,
        keyDigest               [12] OCTET STRING (SIZE(8..64))         OPTIONAL,
        -- pour la vérification de clés publiques
        blockSize               [13] INTEGER                            OPTIONAL,
        -- pour un hachage carré modulo n
        keySizes                [14] SET OF INTEGER                     OPTIONAL,
        -- pour l'algorithme RSA
         publicKeys             [15] SET OF SEQUENCE
                        {modulus            INTEGER,
                         exponent           INTEGER
                        }                                               OPTIONAL,
        sequenceNumber          [16] INTEGER                            OPTIONAL,
        timeStamp               [17] GeneralizedTime                    OPTIONAL,
        encryptedKey            [18] OCTET STRING (SIZE(64..128))       OPTIONAL,
        -- clé de session symétrique, chiffrée en utilisant la clé de chiffrement de clé
        encryptedSymmetricKey   [19] INTEGER                            OPTIONAL,
        -- clé de session symétrique, chiffrée en utilisant la clé publique du récepteur
        keyEncryptionKey        [20] SEQUENCE (SIZE (1..3)) OF KeyId    OPTIONAL,
```

```
        -- une à trois clés symétriques utilisées pour le chiffrement d'une clé de session
        keyListIds                  [21] SET OF KeyListId  OPTIONAL,
        -- liste de clés de chiffrement pouvant être utilisées pendant la durée de l'association
        encryptionCertificate       [22] SET OF EncryptionCertificate       OPTIONAL,
-- certificats X.509 ou itinéraires de certification des clés publiques de l'émetteur
-- utilisés uniquement pour le chiffrement
        signatureCertificate        [23] SET OF SignatureCertificate        OPTIONAL,
-- certificats X.509 ou itinéraires de certification des clés publiques de l'émetteur
-- utilisés uniquement pour des signatures numériques
        encryptedAuthenticatedSymmetricKeys [24] SET OF EncryptedAuthenticatedSymmetricKey
                                                                            OPTIONAL,
-- clé de session symétrique, chiffrée en utilisant la clé publique du récepteur
-- et signée avec la clé de l'émetteur
        macAlgorithms               [25] SET OF OBJECT IDENTIFIER   OPTIONAL,
        publicKeyCertificate        [26] SET OF PublicKeyCertificate        OPTIONAL,
-- certificats X.509 ou itinéraires de certification des clés publiques de l'émetteur sans restriction
-- d'utilisation
        ...
        }

-- La sélection de paramètres de chiffrement est utilisée de manière optionnelle pendant
-- l'établissement de l'association pour négocier les algorithmes et les autres paramètres de
-- chiffrement qui seront pris en charge pendant la durée de l'association. Cette sélection n'est pas
-- utilisée pour les unités PDU de l'élément STASE-ROSE.

KeyListId ::= CHOICE    {identifier     OBJECT IDENTIFIER,
                         name           GraphicString,
                         number         INTEGER
                        }

END
```

5.3.1 Nom de syntaxe abstraite

La présente Recommandation attribue la valeur d'identificateur d'objet ASN.1 suivante:

{itu-t recommendation q(17) q813(813) stase(1) abstractSyntax(1) stase-userinfo(1) }

comme nom de syntaxe abstraite pour l'ensemble des valeurs de données de présentation, chacune de ces dernières étant une valeur de type ASN.1.

STASE-A-ASSOCIATE-Information.EncryptionParametersSelection

La valeur de descripteur d'objet correspondante sera "STASE-ROSE-User-Information".

6 Modèle

La communication entre des processus d'application dans l'environnement OSI est représentée sous la forme d'une communication entre deux entités d'application (AE, *application entity*) qui utilisent le service de présentation. La communication entre certaines applications peut nécessiter un transfert sécurisé d'unité de données protocolaires d'application (APDU).

Les unités APDU émises par l'une des entités d'application (l'émetteur) sont reçues par l'autre entité d'application (le récepteur). Le transfert sécurisé garantit que des unités APDU transférées par l'émetteur peuvent être soumises aux opérations de vérification de leur intégrité et de leur non-répudiation et qu'elles ne seront comprises que par le destinataire prévu pour une unité APDU donnée. Le transfert sécurisé implique des transformations de sécurité (par exemple, un chiffrement) sur les unités APDU de l'entité émettrice avant leur transfert et l'application des transformations de

sécurité inverse (par exemple, un déchiffrement) avant leur livraison à l'entité d'application réceptrice. L'élément STASE-ROSE traite uniquement le transfert sécurisé d'unités de données protocolaires d'application de l'élément de service d'opérations distantes (ROSE).

Le transfert sécurisé s'exécute dans le contexte de l'association d'application. Une association d'application définit la relation entre deux entités d'application et se constitue d'un échange d'informations de commande de protocole d'application qui utilise les services de la couche Présentation. L'entité d'application qui initialise l'association est appelée "entité d'application initiatrice" ou initiateur de l'association; l'entité d'application qui répond à l'initialisation d'une association par une autre entité d'application est appelée "entité d'application répondante" ou répondeur de l'association.

Les fonctionnalités d'une entité d'application se divisent en un processus d'application et un ensemble d'éléments de service d'application (ASE). Chaque élément ASE peut être subdivisé à son tour en un ensemble d'éléments ASE (plus élémentaires). Les interactions entre les entités d'application sont décrites sous la forme de leur utilisation d'éléments ASE.

La combinaison particulière d'un processus d'application et l'ensemble d'éléments ASE qui constituent une entité d'application sont définis par le contexte d'application.

La Figure 1 présente un exemple de contexte d'application utilisant l'élément STASE-ROSE (seule la relation entre les entités homologues au niveau le plus élevé est représentée).

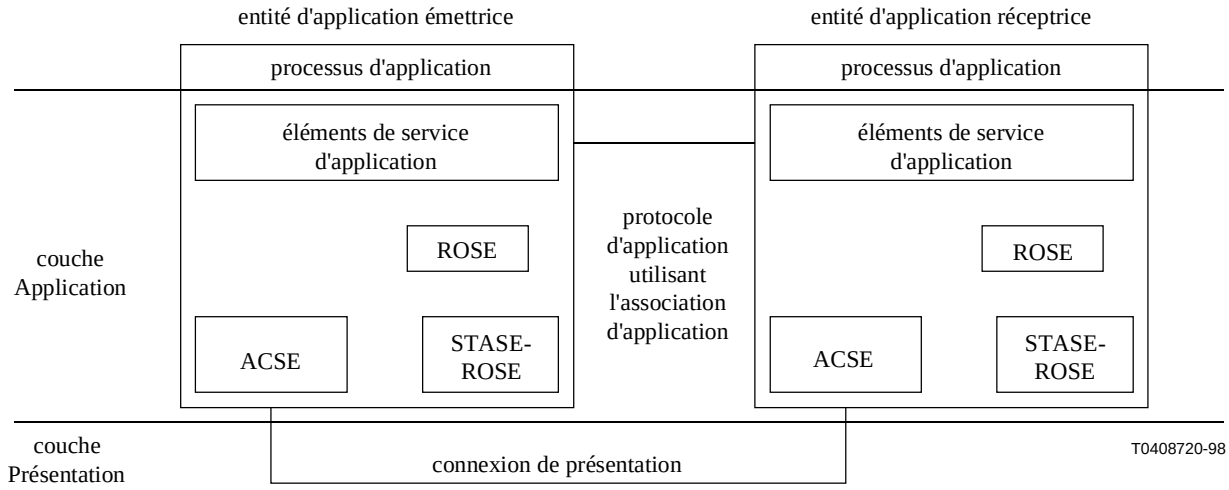

Figure 1/Q.813 – Eléments ASE obligatoires lorsque l'élément STASE-ROSE est utilisé

Les éléments ASE dont dispose un processus d'application doivent pouvoir communiquer au moyen d'une association d'application. La commande de l'association d'application est effectuée par les services de traitement d'application fournis par l'élément de commande de service d'application (ACSE).

La Figure 2 décrit les éléments ASE qui doivent être présents lorsque l'élément STASE-ROSE est utilisé. Elle présente les éléments ASE pour le réseau de gestion des télécommunications (voir la Recommandation M.3010) utilisant l'élément de service d'informations communes de gestion (CMISE) en plus des éléments ACSE, ROSE et STASE-ROSE.

La présente Recommandation peut être utilisée pour toute application qui utilise l'élément ROSE. Le texte du présent paragraphe se limite toutefois, pour simplifier la présentation, à l'élément CMISE comme élément ASE utilisateur de l'élément ROSE.

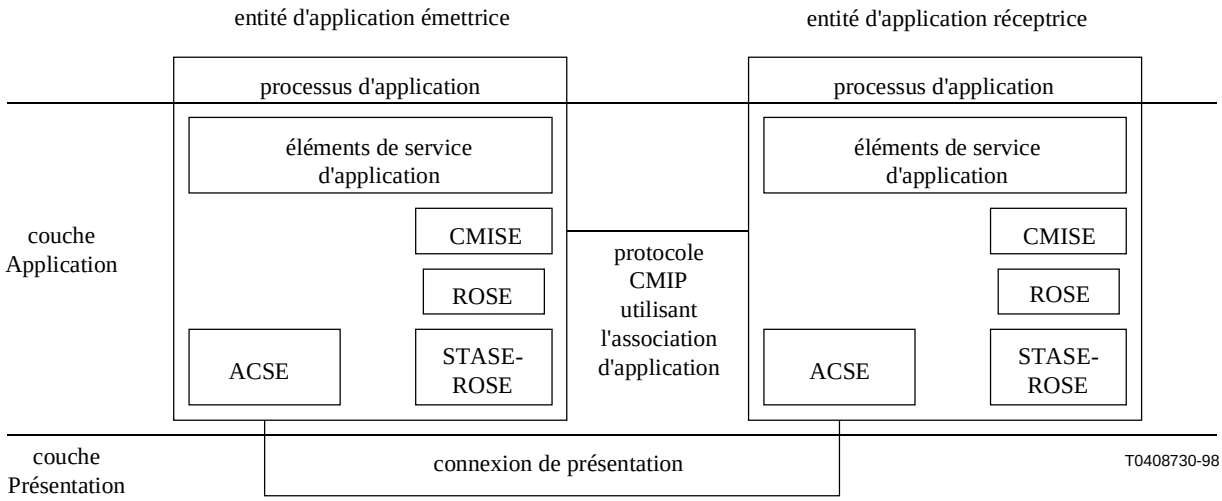

Figure 2/Q.813 – Eléments ASE présents lorsque l'élément STASE-ROSE est utilisé pour sécuriser l'élément CMISE

7 Aperçu général du service

7.1 Services d'association

La présente Recommandation ne fournit pas de services distincts pour l'établissement et la libération d'associations d'application. Le processus d'application du contexte d'application qui utilise l'élément STASE-ROSE fait appel aux services décrits dans la Recommandation X.217 pour la commande d'associations d'application.

Diverses entités ASE du contexte d'application échangent des informations d'initialisation durant la phase d'établissement de l'association afin d'établir une association utilisant des entités ACSE. Les prescriptions des contextes d'application, de présentation et de session sont véhiculées au moyen des paramètres du service A-ASSOCIATE.

Les services A-RELEASE et A-ABORT décrits dans la Recommandation X.217 sont utilisés pour mettre fin à une association. Ils peuvent être invoqués par les éléments utilisateur.

7.2 Services STASE-ROSE

Le Tableau 7-1 énumère les services STASE-ROSE.

Le paragraphe qui suit donne une description sommaire du service STASE-ROSE:

– SR-TRANSFER

  Le service SR-TRANSFER fournit à un élément ROSE le moyen d'initialiser un transfert sécurisé d'unités PDU ROSE vers un élément ROSE homologue.

Tableau 7-1/Q.813 – Services STASE-ROSE

Service	Type
SR-TRANSFER	sans confirmation

7.3 Relation avec le service de présentation

Le service STASE-ROSE nécessite l'accès au service P-DATA.

Le contexte de présentation se constitue d'une syntaxe abstraite nommée avec une syntaxe de transfert compatible négociée par la couche Présentation. La syntaxe de transfert BER est négociée d'ordinaire au moment de l'établissement d'une association, avec un contexte d'application qui contient l'élément STASE-ROSE. Bien que les règles DER soient utilisées par l'élément STASE-ROSE pour réaliser le codage avant d'appliquer les transformations de sécurité, ces règles ne font pas partie du contexte d'application (à moins qu'elles ne soient également utilisées par la couche Présentation).

7.4 Définition du service

7.4.1 Conventions

La présente Recommandation définit des services pour l'élément STASE-ROSE en utilisant les conventions de description définies dans la Recommandation X.210. La définition du service de l'élément STASE-ROSE contient un tableau qui donne la liste des paramètres des primitives. La présence d'un paramètre dans un tableau est définie par l'une des valeurs suivantes:

-- ne s'applique pas

M obligatoire

U option utilisateur

C conditionnel

La notation (=) indique en outre qu'une valeur de paramètre est identique à la valeur qui se trouve à sa gauche dans le tableau.

Le caractère "**.**" est utilisé dans la présente Recommandation pour indiquer des champs dans un type ASN.1 type. L'expression **a.x** est utilisé, par exemple, pour indiquer le champ **x** contenu dans le champ **a** des types Exemple1 et Exemple2 ASN.1 définis ci-dessous.

```
Exemple1 ::= SEQUENCE {
        a  SEQUENCE {
                x INTEGER,
                y BOOLEAN
        },
        b  INTEGER
}

A ::= SEQUENCE {
                x INTEGER,
                y BOOLEAN
}

Exemple2 := SEQUENCE {
                a A,
                b INTEGER
}
```

7.4.2 Service d'association

7.4.2.1 Etablissement de l'association

Le service A-ASSOCIATE décrit dans la Recommandation X.217 est invoqué par le processus d'application d'un contexte d'application impliquant l'élément STASE-ROSE pour établir une

association avec un processus d'application homologue. L'établissement de l'association est la première phase de l'activité de toute instance de transfert sécurisé.

Le Tableau 7-2 donne la liste des paramètres qui sont définis par la présente Recommandation comme devant constituer la partie spécifique de l'élément STASE-ROSE pour le paramètre "informations utilisateur" du service A-ASSOCIATE. Ces informations sont spécifiées par l'initiateur de l'association et sont échangées au moment de l'établissement d'une association. L'échange de ces informations d'initialisation se fait de manière optionnelle avant l'utilisation des services STASE-ROSE.

Tableau 7-2/Q.813 – Informations utilisateur du service A-ASSOCIATE

Nom du paramètre	Signification	Demande/ Indication	Réponse/ Confirmation
symmetricKeyIds	*identificateurs de clé symétrique*	U	C
publicKeyIds	*identificateurs de clé publique*	U	C
sealKeyIds	*identificateurs de clé de scellé*	U	C
signatureKeyIds	*identificateurs de clé de signature*	U	C
passwordIds	*identificateurs de mot de passe*	U	C
initializationVector	*vecteur d'initialisation*	U	U
feedBackBits	*bits de feed-back*	U	U
symmetricAlgorithms	*algorithmes symétriques*	U	C
publicKeyAlgorithms	*algorithmes avec clé publique*	U	C
signatureAlgorithms	*algorithmes de signature*	U	C
sealAlgorithms	*algorithmes de scellé*	U	C
hashAlgorithms	*algorithmes de hachage*	U	C
keyDigest	*résumé de clé*	U	U
blockSize	*taille de bloc*	U	U
keySize	*taille de clé*	U	C
publicKeys	*clés publiques*	U	U
sequenceNumber	*numéro de séquence*	U	U
timeStamp	*horodatage*	U	U
encryptedKey	*clé chiffrée*	U	U
encryptedSymmetricKey	*clé symétrique chiffrée*	U	U
keyEncryptionKey	*clé de chiffrement de clé*	U	U
keyListIds	*identificateurs de liste de clés*	U	C
publicKeyCertificates	*certificats de clé publique*	U	U
encryptionCertificates	*certificats de chiffrement*	U	U
signatureCertificates	*certificats de signature*	U	U
encryptedAuthenticated SymmetricKeys	*clés symétriques authentifiées chiffrées*	U	U
macAlgorithms	*algorithmes de bloc MAC*	U	C

La condition C dans le Tableau 7-2 signifie qu'un paramètre ne figure dans la primitive de réponse ou de confirmation que s'il figure dans la primitive de demande ou d'indication. S'il figure dans la primitive de demande ou d'indication sans figurer dans la primitive de réponse ou de confirmation, la primitive de réponse ou de confirmation est alors interprétée comme si le paramètre était présent avec une valeur nulle.

La signification de ces paramètres et des réponses attendues de la part du récepteur est détaillée ci-dessous:

- **symmetricKeyIds**: ensemble d'identificateurs de clé devant être utilisé pour cette association pour un chiffrement symétrique. Le répondeur de l'association répondra en indiquant le même ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre symmetricKeyIds figure dans une primitive d'indication A-ASSOCIATE.
- **publicKeyIds**: ensemble d'identificateurs de clé devant être utilisé pour cette association pour un chiffrement par clé publique. Le répondeur de l'association répondra en indiquant le même ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre publicKeyIds figure dans une primitive d'indication A-ASSOCIATE.
- **sealKeyIds**: ensemble d'identificateurs de clé devant être utilisé pour cette association pour un scellé. Le répondeur de l'association répondra en indiquant le même ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre sealKeyIds figure dans une primitive d'indication A-ASSOCIATE.
- **signatureKeyIds**: ensemble d'identificateurs de clé devant être utilisé pour cette association pour une signature numérique. Le répondeur de l'association répondra en indiquant le même ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre signatureKeyIds figure dans une primitive d'indication A-ASSOCIATE.
- **passwordIds**: ensemble d'identificateurs de mot de passe pour les mots de passe devant être utilisés pour cette association. Le répondeur de l'association répondra en indiquant le même ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre figure dans une primitive d'indication A-ASSOCIATE.
- **initializationVector**: vecteur d'initialisation (IV) devant être utilisé pour un chiffrement DES dans le mode de chaînage de blocs de chiffre (CBC, *ciphered block chaining*). Chaque participant peut utiliser un vecteur d'initialisation différent pour les messages qu'il émet.
- **feedBackBits**: bits de feed-back devant être utilisés pour un chiffrement DES dans les modes avec feed-back de chiffrement à k bits ou un feed-back de sortie à b bits. Chaque participant peut utiliser une valeur différente pour les bits de feed-back des messages qu'il émet.
- **symmetricAlgorithms**: ensemble d'algorithmes symétriques que l'initiateur de l'association est en mesure de prendre en charge. Le répondeur de l'association répondra en indiquant le même ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre symmetricAlgorithms figure dans une primitive d'indication A-ASSOCIATE.
- **publicKeyAlgorithms**: ensemble d'algorithmes avec clé publique que l'initiateur de l'association est en mesure de prendre en charge. Le répondeur de l'association répondra en indiquant le même ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre publicKeyAlgorithms figure dans une primitive d'indication A-ASSOCIATE.
- **signatureAlgorithms**: ensemble d'algorithmes de signature que l'initiateur de l'association est en mesure de prendre en charge. Le répondeur de l'association répondra en indiquant le même ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre signatureAlgorithms figure dans une primitive d'indication A-ASSOCIATE.
- **sealAlgorithms**: ensemble d'algorithmes de scellé que l'initiateur de l'association est en mesure de prendre en charge. Le répondeur de l'association répondra en indiquant le même ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre sealAlgorithms figure dans une primitive d'indication A-ASSOCIATE.
- **hashAlgorithms**: ensemble d'algorithmes de hachage que l'initiateur de l'association est en mesure de prendre en charge. Le répondeur de l'association répondra en indiquant le même

ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre hashAlgorithms figure dans une primitive d'indication A-ASSOCIATE.

- **keyDigest**: résumé de message (empreinte digitale) d'une clé publique utilisé pour vérifier la validité d'une clé publique.
- **blockSize**: taille de bloc devant être utilisé pour un hachage "carré modulo n". Chaque participant peut choisir une taille de bloc différente pour les messages qu'il émet.
- **keySize**: taille de clé pour l'algorithme de chiffrement RSA. Le répondeur de l'association répondra avec une taille de clé identique ou différente si le paramètre keySize figure dans la primitive d'indication A-ASSOCIATE.
- **publicKeys**: ensemble de clés publiques devant être utilisé par l'émetteur sur cette association. Le répondeur de l'association peut répondre en émettant son propre ensemble de clés publiques.
- **sequenceNumber**: numéro de séquence de départ pour les unités PDU ROSE si l'association doit être protégée contre des attaques par répétition et suppression. Chaque participant peut choisir un numéro de séquence différent au démarrage. Si ce numéro est présent, le fournisseur du service STASE-ROSE de tout participant qui le fournit attribuera un numéro de séquence à toute unité APDU du service STASE-ROSE qui est émise sur l'association d'application.
- **timeStamp**: date et heure à laquelle la primitive de demande A-ASSOCIATE a été initialisée par l'initiateur de l'association. L'interprétation de ce paramètre dépend de l'implémentation et elle est en dehors du domaine d'application de la présente Recommandation. Si le répondeur de l'association émet un horodatage, la valeur de ce dernier correspondra à l'instant d'émission de la primitive de réponse A-ASSOCIATE.
- **encryptedKey**: clé symétrique utilisée pour (faisant partie de) l'association et chiffrée en utilisant une clé de chiffrement de clé symétrique (KEK, *key encryption key*).
- **encryptedSymmetricKey**: clé symétrique utilisée pour (faisant partie de) l'association et chiffrée en utilisant la clé publique du récepteur.
- **keyEncryptionKey**: identifie une à trois clés symétriques devant être utilisées comme clé KEK symétrique.
- **keyListIds**: ensemble d'identificateurs de listes de clés de chiffrement symétrique dont l'initiateur de l'association propose l'utilisation. Le répondeur de l'association répondra en indiquant le même ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre keyListIds figure dans la primitive d'indication A-ASSOCIATE.
- **publicKeyCertificates**: itinéraire de certification X.509 qui certifie la clé publique de l'émetteur.
- **encryptionCertificates**: itinéraire de certification X.509 qui certifie la clé publique de l'émetteur pouvant être utilisée uniquement pour le chiffrement.
- **signatureCertificates**: itinéraire de certification X.509 qui certifie la clé publique de l'émetteur pouvant être utilisé uniquement pour des signatures numériques.
- **encryptedAuthenticatedSymmetricKeys**: clé symétrique utilisée pour (faisant partie de) l'association et chiffrée en utilisant la clé publique du récepteur, suivie d'un horodatage (temps généralisé), de l'identificateur de l'émetteur, de l'identificateur du récepteur et d'une signature calculée à partir de la représentation ASCII de ces quatre champs en utilisant la clé privée de l'émetteur.
- **macAlgorithms**: ensemble d'algorithmes de bloc MAC que l'initiateur de l'association est en mesure de prendre en charge. Le répondeur de l'association répondra en indiquant le même

ensemble ou un sous-ensemble de l'ensemble précédent si le paramètre macAlgorithms figure dans la primitive d'indication A-ASSOCIATE.

7.4.2.2 Libération de l'association

Le service A-RELEASE décrit dans la Recommandation X.217 est invoqué par le processus d'application d'un contexte d'application impliquant l'élément STASE-ROSE pour demander la clôture ordonnée d'une association entre des entités d'application homologues. La présente Recommandation ne spécifie aucune utilisation de paramètre pour le service A-RELEASE.

Le service A-ABORT est invoqué par le processus d'application pour demander l'abandon brutal de l'association d'application.

7.4.3 Service SR-TRANSFER

Le service SR-TRANSFER est utilisé par un utilisateur du service STASE-ROSE (élément ROSE) pour transférer de manière sécurisée une unité PDU ROSE vers un utilisateur du service STASE-ROSE (élément ROSE) homologue.

La structure du service se constitue de deux primitives de service, comme indiqué dans la Figure 3.

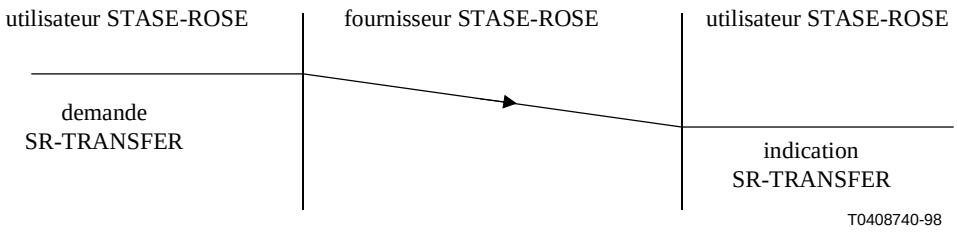

Figure 3/Q.813 – Primitives du service SR-TRANSFER

7.4.4 Paramètres du service SR-TRANSFER

Le Tableau 7-3 donne la liste des paramètres du service SR-TRANSFER.

Tableau 7-3/Q.813 – Paramètres du service SR-TRANSFER

Nom du paramètre	Demande	Indication
Unité PDU ROSE	M	M(=)
Type de chiffrement	M	M(=)
Paramètres de chiffrement	U	C(=)

7.4.4.1 Unité PDU ROSE

Ce paramètre identifie l'unité PDU ROSE qui doit être transférée. Il doit être fourni par le demandeur du service et les valeurs des données seront conformes à la définition des unités APDU ROSE de la Recommandation X.229.

7.4.4.2 Type de chiffrement

Ce paramètre identifie le type de transformations de sécurité demandé par l'utilisateur du service pour l'unité PDU ROSE courante. La liste qui suit indique des valeurs valides pour le type (Note, l'utilisation de valeurs par défaut pour les paramètres de chiffrement est spécifiée au 5.2.1):

- **clear** (*clair*): aucune transformation de sécurité n'est demandée;
- **simpleConfidential** (*confidentiel simple*): protection de confidentialité de l'ensemble de l'unité PDU en utilisant les paramètres par défaut pris en charge par le fournisseur du service;
- **confidential** (*confidentiel*): protection de confidentialité de l'ensemble de l'unité PDU en utilisant les valeurs fournies dans les paramètres de chiffrement;
- **simplePublicEnciphered** (*chiffrement public simple*): protection de confidentialité de l'ensemble de l'unité PDU en utilisant la clé publique par défaut prise en charge par le fournisseur du service;
- **publicEnciphered** (*chiffrement public*): protection de confidentialité de l'ensemble de l'unité PDU en utilisant la clé publique fournie dans les paramètres de chiffrement;
- **simpleHashed** (*haché simple*): bloc MAC obtenu par hachage de l'unité PDU en utilisant les valeurs par défaut;
- **hashed** (*haché*): bloc MAC obtenu par hachage de l'unité PDU en utilisant les valeurs fournies par les paramètres de chiffrement;
- **simpleSealed** (*scellé simple*): scellé de l'unité PDU en utilisant les valeurs par défaut;
- **sealed** (*scellé*): scellé de l'unité PDU en utilisant les valeurs fournies par les paramètres de chiffrement;
- **simpleSigned** (*signé simple*): signature numérique de l'unité PDU en utilisant les valeurs par défaut:
- **signed** (*signé*): signature numérique de l'unité PDU en utilisant les valeurs fournies par les paramètres de chiffrement;
- **simpleConfidentialSigned** (*confidentiel signé simple*): protection de confidentialité de l'ensemble de l'unité PDU et signature numérique de l'unité PDU en utilisant les valeurs par défaut;
- **confidentialSigned** (*confidentiel signé*): protection de confidentialité de l'ensemble de l'unité PDU et signature numérique de l'unité PDU en utilisant les valeurs fournies par les paramètres de chiffrement;
- **simpleConfidentialMAC** (*confidentiel MAC simple*): protection de confidentialité de l'ensemble de l'unité PDU et code MAC de l'unité PDU en utilisant les valeurs par défaut;
- **confidentialMAC** (*confidentiel MAC*): protection de confidentialité de l'ensemble de l'unité PDU et code MAC de l'unité PDU en utilisant les valeurs fournies par les paramètres de chiffrement;
- **simpleConfidentialSealed** (*confidentiel scellé simple*): protection de confidentialité de l'ensemble de l'unité PDU et scellé de l'unité PDU en utilisant les valeurs par défaut;
- **confidentialSealed** (*confidentiel scellé*): protection de confidentialité de l'ensemble de l'unité PDU et scellé de l'unité PDU en utilisant les valeurs fournies par les paramètres de chiffrement.

7.4.4.3 Paramètres de chiffrement

Ce paramètre identifie les paramètres devant être utilisés pour les transformations de sécurité. La présence de ce paramètre dépend du type de chiffrement choisi par l'utilisateur (tel que ce type est décrit dans le sous-paragraphe précédent).

8 Interactions entre éléments de service d'application

Le présent paragraphe décrit les interactions entre les processus d'application, les éléments ACSE, ROSE et STASE-ROSE, l'usager ROSE (par exemple, l'élément CMISE) et les services de la couche Présentation pendant les différentes phases de la communication entre deux entités d'application. D'autres interactions sont possibles; elles conduisent à l'échange d'un ensemble de messages similaires avec des fonctionnalisés identiques entre les systèmes communicants. Le choix de ces interactions est une affaire locale. L'Annexe A traite ce point plus en profondeur dans le cas où l'élément CMISE est l'utilisateur du service ROSE.

8.1 Interactions lors de l'établissement de l'association

La Figure 4 présente la succession, lors de la phase d'établissement de l'association, des interactions entre le processus d'application, divers éléments ASE et le fournisseur du service de présentation.

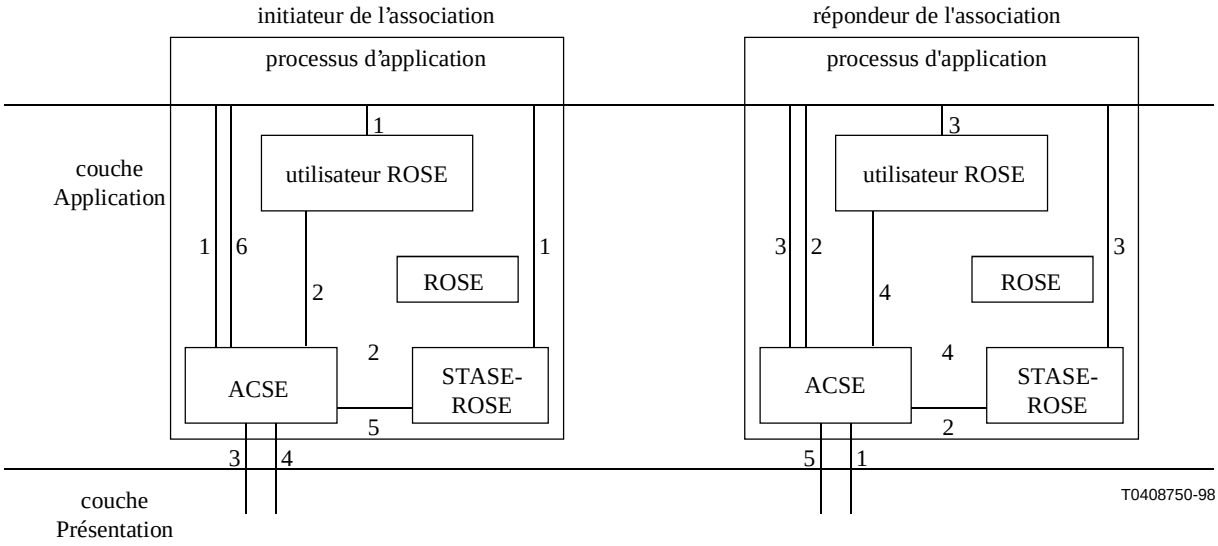

Figure 4/Q.813 – Interactions durant l'établissement de l'association

8.1.1 Initiateur de l'association

Le texte qui suit décrit les interactions du côté initiateur de l'association dans la Figure 4:

1) le processus d'application de l'entité d'application qui fait appel au service STASE-ROSE émet une primitive de demande A-ASSOCIATE à destination de l'élément ACSE afin d'établir une association d'application. Si une authentification de l'entité homologue est souhaitée, le processus d'application fournit alors à l'élément ACSE la valeur de l'élément d'authentification devant être véhiculée dans le champ "valeur d'authentification" de l'unité PDU de demande AARQ (en utilisant l'élément ACSE "unité fonctionnelle d'authentification"). Le processus d'application peut également, durant la même phase, informer l'élément STASE-ROSE et les éléments ASE qui utilisent le service ROSE (par exemple, l'élément CMISE) au sujet de l'association demandée et fournir à l'élément STASE-ROSE toute valeur proposée pour tout ou partie des paramètres de chiffrement;

2) le service STASE-ROSE fournit à l'élément ACSE toute valeur proposée pour tout ou partie des paramètres de chiffrement. Le mécanisme utilisé par le service STASE-ROSE pour informer l'élément ACSE est une affaire d'implémentation qui n'est pas traitée par la présente Recommandation. Ces informations seront véhiculées dans le champ "informations utilisateur" de l'élément ACSE en utilisant le paramètre "sélection de paramètres de

chiffrement" défini au 5.3. L'entité ASE, ou les entités ASE utilisatrices du service ROSE peuvent également fournir à l'élément ACSE des informations relatives à ces entités ASE. Toute information de ce type est véhiculée dans le champ "informations utilisateur" de l'élément ACSE. Le champ "informations utilisateur" de l'élément ACSE défini dans la Rec. UIT-T X.227 | ISO/CEI 8650-1 se constitue d'une expression SEQUENCE OF EXTERNAL. Les informations destinées le cas échéant à l'élément STASE-ROSE seront véhiculées dans le premier terme EXTERNAL. Le contexte d'application spécifiera le, ou les, termes EXTERNAL qui véhiculeront des informations pour chacune des autres entités ASE;

3) l'entité ACSE émet une primitive de demande P-CONNECT à destination du fournisseur de présentation afin d'établir une association d'application;

   le fournisseur du service de présentation transfère la primitive de demande P-CONNECT et reçoit une réponse (qui n'est pas indiquée ci-dessous);

4) le fournisseur de présentation émet une primitive de confirmation P-CONNECT à destination de l'élément ACSE qui confirme l'établissement d'une connexion de présentation;

5) l'élément ACSE informe le service STASE-ROSE de l'établissement d'une nouvelle association d'application et fournit à l'élément STASE-ROSE les valeurs éventuelles des paramètres de chiffrement. Le mécanisme utilisé par le service STASE-ROSE pour informer l'élément ACSE est une affaire d'implémentation qui n'est pas traitée par la présente Recommandation;

6) l'élément ACSE émet une primitive de confirmation A-ASSOCIATE à destination du processus d'application pour confirmer l'établissement de l'association. L'élément ACSE fournit au processus d'application les valeurs éventuelles des paramètres de chiffrement. Le mécanisme utilisé par l'élément ACSE pour informer le processus d'application est une affaire d'implémentation qui n'est pas traitée par la présente Recommandation.

8.1.2 Répondeur de l'association

Le texte qui suit décrit les interactions du côté répondeur de l'association dans la Figure 4.

Le fournisseur du service de présentation reçoit une demande de connexion en provenance du fournisseur de présentation distant:

1) le fournisseur de présentation émet une primitive d'indication P-CONNECT à destination de l'élément ACSE indiquant qu'un utilisateur de service distant souhaite établir une association;

2) l'élément ACSE émet une primitive d'indication A-ASSOCIATE à destination du processus d'application. Durant la même phase, l'élément ACSE informe le service STASE-ROSE de la demande d'établissement d'une nouvelle association d'application et fournit à l'élément STASE-ROSE les valeurs éventuelles des paramètres de chiffrement. L'élément ACSE fournit ensuite aux utilisateurs de l'élément ROSE les informations éventuelles spécifiques d'élément ASE (véhiculées dans le champ "informations utilisateur"). Le mécanisme utilisé par le service STASE-ROSE pour informer l'élément ACSE est une affaire d'implémentation qui n'est pas traitée par la présente Recommandation;

3) le processus d'application émet une primitive de réponse A-ASSOCIATE à destination de l'élément ACSE pour accepter ou rejeter l'association d'application. Si la primitive d'indication A-ASSOCIATE contient des valeurs proposées pour tout ou partie des paramètres de chiffrement, le processus d'application peut alors indiquer à l'élément STASE-ROSE quelles sont les valeurs de ces paramètres de chiffrement qui doivent être acceptées. Le mécanisme utilisé par le processus d'application pour informer l'élément STASE-ROSE est une affaire d'implémentation qui n'est pas traitée par la présente

Recommandation. Le processus d'application peut également informer durant cette phase le ou les éléments ASE utilisateurs du service ROSE au sujet de l'association;

4) l'élément STASE-ROSE fournit à l'élément ACSE les valeurs éventuelles qui ont été acceptées pour les paramètres de chiffrement dans la demande AARQ. Le mécanisme utilisé par l'élément STASE-ROSE pour informer l'élément ACSE est une affaire d'implémentation qui n'est pas traitée par la présente Recommandation. Ces informations seront véhiculées dans le champ "informations utilisateur" de l'élément ACSE en utilisant le paramètre "sélection de paramètres de chiffrement" défini au 5.3. Le ou les éléments ASE utilisateurs de l'élément ROSE peuvent également fournir durant la même phase à l'élément ACSE des informations pertinentes pour ce ou ces éléments ASE. Le champ "informations utilisateur" de l'élément ACSE défini dans la Rec. UIT-T X.227 | ISO/CEI 8650-1 se constitue d'une expression SEQUENCE OF EXTERNAL. Les informations destinées le cas échéant à l'élément STASE-ROSE seront véhiculées dans le premier terme EXTERNAL. Le contexte d'application spécifiera le ou les termes EXTERNAL qui véhiculeront des informations pour chacune des autres entités ASE;

5) l'élément ACSE émet une primitive de réponse P-CONNECT à destination du fournisseur de présentation pour accepter ou rejeter l'établissement de l'association.

8.2 Libération de l'application

La Figure 5 présente la succession, lors de la phase de libération de l'association, des interactions entre le processus d'application, divers éléments ASE et le fournisseur du service de présentation.

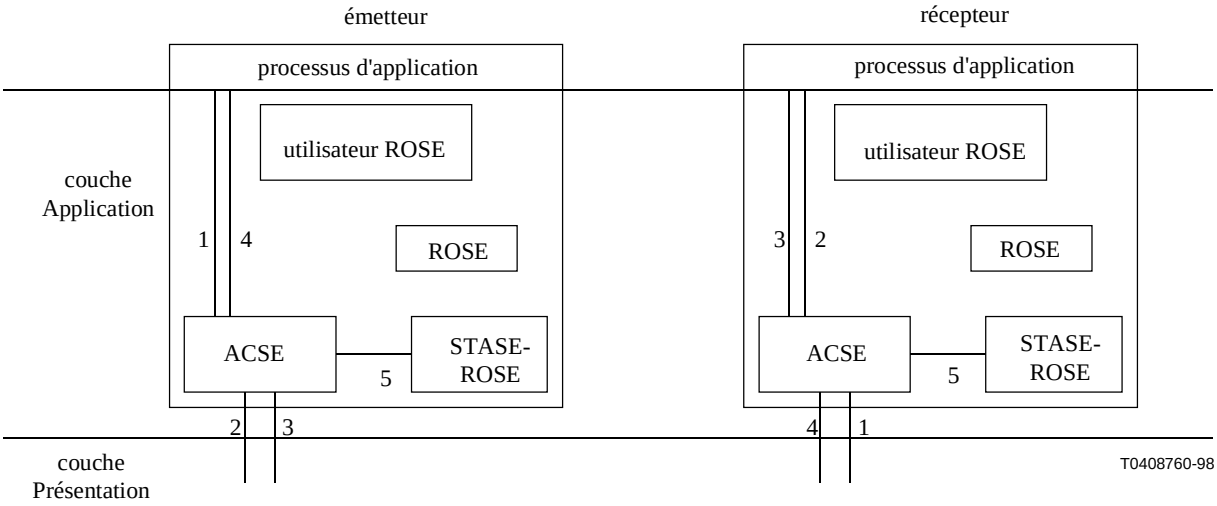

Figure 5/Q.813 – Interactions durant la libération de l'association

8.2.1 Emetteur

Le texte qui suit décrit les interactions du côté initiateur de l'association dans la Figure 5:

1) le processus d'application du contexte d'application qui fait appel au service STASE-ROSE émet une primitive de demande A-RELEASE à destination de l'élément ACSE pour libérer une association d'application;

2) l'élément ACSE émet une primitive de demande P-RELEASE à destination du fournisseur de présentation pour libérer une association d'application.

Le fournisseur du service de présentation transfère ensuite la primitive de demande P-RELEASE vers l'entité d'application homologue et reçoit une réponse (non indiquée ci-dessous);

3) le fournisseur de présentation émet une primitive de confirmation P-RELEASE à destination de l'élément ACSE qui confirme la libération d'une connexion de présentation;

4) l'élément ACSE émet une primitive de confirmation A-RELEASE à destination du processus d'application qui confirme la libération de l'association d'application;

5) l'élément ACSE informe l'élément STASE-ROSE et d'autres éléments ASE (non indiqués dans la figure) de la libération de l'association d'application. Le mécanisme utilisé par l'élément ACSE pour informer l'élément STASE-ROSE et d'autres éléments ASE est une affaire d'implémentation qui n'est pas traitée par la présente Recommandation.

8.2.2 Récepteur

Le texte qui suit décrit les interactions du côté récepteur de la Figure 5.

Le fournisseur du service de présentation reçoit une demande de libération en provenance du fournisseur de présentation distant:

1) le fournisseur de présentation émet une primitive d'indication P-RELEASE à destination de l'élément ACSE pour indiquer le souhait d'un utilisateur du service distant de libérer une association;

2) l'élément ACSE émet une primitive d'indication A-RELEASE à destination du processus d'application;

3) le processus d'application émet une primitive de réponse A-RELEASE à destination de l'élément ACSE pour accepter la libération de l'association d'application;

4) l'élément ACSE émet une primitive de réponse P-RELEASE à destination du fournisseur de présentation pour accepter la libération de l'association;

5) l'élément ACSE informe l'élément STASE-ROSE et d'autres éléments ASE (non indiqués dans la figure) de la libération de l'association d'application. Le mécanisme utilisé par l'élément ACSE pour informer l'élément STASE-ROSE et d'autres éléments ASE est une affaire d'implémentation qui n'est pas traitée par la présente Recommandation.

8.3 Abandon de l’association

La Figure 6 présente la succession, lors de la phase d'abandon de l'association, des interactions entre le processus d'application, divers éléments ASE et le fournisseur du service de présentation.

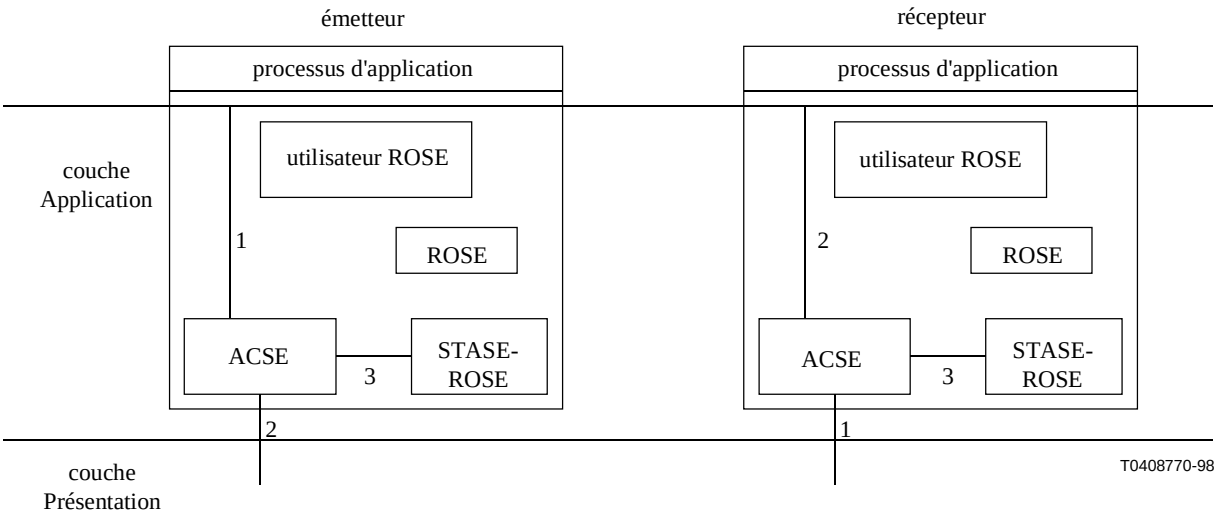

Figure 6/Q.813 – Interactions durant l'abandon de l'association

8.3.1 Emetteur

Le texte qui suit décrit les interactions du côté initiateur de l'association dans la Figure 6:

1) le processus d'application du contexte d'application qui fait appel au service STASE-ROSE émet une primitive de demande A-ABORT à destination de l'élément ACSE afin d'abandonner une association d'application;

2) l'élément ACSE émet une primitive de demande P-ABORT à destination du fournisseur de présentation pour abandonner la connexion de présentation;

3) l'élément ACSE informe l'élément STASE-ROSE et d'autres éléments ASE de l'abandon de l'association d'application. Le mécanisme utilisé par l'élément ACSE pour informer l'élément STASE-ROSE et d'autres éléments ASE est une affaire d'implémentation qui n'est pas traitée par la présente Recommandation.

8.3.2 Récepteur

Le texte qui suit décrit les interactions du côté récepteur de la Figure 6.

Le fournisseur du service de présentation détecte l'abandon d'une connexion de présentation:

1) le fournisseur de présentation émet une primitive d'indication P-ABORT à destination de l'élément ACSE pour indiquer qu'une connexion de présentation a été abandonnée;

2) l'élément ACSE émet une primitive d'indication A-ABORT à destination du processus d'application;

3) l'élément ACSE informe l'élément STASE-ROSE et d'autres éléments ASE de l'abandon de l'association d'application. Le mécanisme utilisé par l'élément ACSE pour informer l'élément STASE-ROSE et d'autres éléments ASE est une affaire d'implémentation qui n'est pas traitée par la présente Recommandation.

8.4 Transfert de données

La Figure 7 présente la succession, lors de la phase d'abandon de l'association, des interactions entre le processus d'application, divers éléments ASE et le fournisseur du service de présentation.

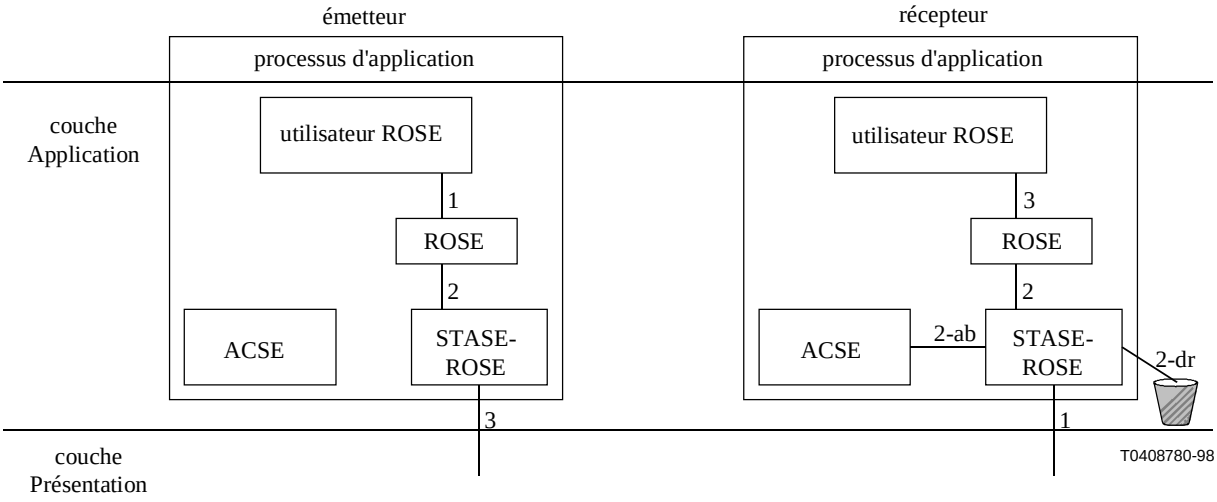

Figure 7/Q.813 – Interactions durant le transfert de données

8.4.1 Emetteur

Le texte qui suit décrit les interactions du côté initiateur de l'association dans la Figure 7:

1) l'utilisateur du service ROSE émet une primitive ROSE de demande ou de réponse de transfert de données, pour demander un transfert de données à la suite d'une initiative du processus d'application;

2) l'élément ROSE émet une primitive de demande SR-TRANSFER à destination de l'élément STASE-ROSE pour demander un transfert sécurisé de données;

3) l'élément STASE-ROSE effectue sur l'unité PDU ROSE (figurant dans les paramètres de la demande) le codage et les transformations de sécurité nécessaires, puis émet une primitive de demande P-DATA à destination du fournisseur de présentation. L'élément STASE-ROSE peut, tel qu'il est défini à l'heure actuelle, être utilisé par des services correspondant à toutes les unités fonctionnelles du service CMIS, à l'exception des services étendus.

8.4.2 Récepteur

Le texte qui suit décrit les interactions du côté récepteur de la Figure 7:

1) le fournisseur de présentation émet une primitive d'indication P-DATA à destination de l'élément STASE-ROSE pour l'informer de l'arrivée de données en provenance de l'application homologue sur une connexion d'application;

2) l'élément STASE-ROSE effectue les transformations de sécurité inverses sur les données entrantes, vérifie la validité de l'unité PDU (par exemple, la validité du scellé ou de la signature, l'actualité de l'horodatage, la valeur du numéro de séquence) et émet une primitive d'indication SR-TRANSFER à destination de l'élément ROSE si l'unité PDU est reconnue valide.

   L'élément ROSE émet une primitive d'indication ou de confirmation du service ROSE à destination de l'utilisateur du service ROSE qui informe ensuite (ceci n'est pas indiqué dans la figure) le processus d'application de l'arrivée de données en provenance d'une entité d'application homologue.

   L'élément STASE-ROSE peut établir que l'unité APDU reçue n'est pas acceptable (par exemple, si le déchiffrement échoue). L'action qui doit être effectuée dans un tel cas par l'élément STASE-ROSE est une affaire locale. La présente Recommandation recommande toutefois l'une des actions suivantes:

- l'implémentation de l'entité STASE-ROSE réceptrice peut rejeter l'unité APDU entrante comme indiqué en **2-dr** dans la Figure 7;
- l'implémentation de l'entité STASE-ROSE réceptrice peut émettre une primitive A-ABORT à destination de l'élément ACSE comme indiqué en **2-ab** dans la Figure 7.

Dans l'un ou l'autre cas, il est recommandé que l'événement fasse l'objet d'un compte rendu à destination de l'élément utilisateur, que l'événement soit journalisé dans une trace de vérification de sécurité et qu'une alarme de sécurité soit émise à destination de l'administrateur de sécurité local (action non représentée dans la figure).

9 Protocole STASE-ROSE

Le protocole spécifié dans le présent paragraphe prend en charge les services STASE-ROSE décrits précédemment. Comme il a été indiqué, l'élément STASE-ROSE utilise le service P-DATA de la couche Présentation pour le transfert d'unités PDU ROSE sécurisées.

La machine de protocole STASE-ROSE (SRPM) communique avec l'élément ROSE en utilisant les primitives du service SR-TRANSFER décrites précédemment. La machine SRPM est activée par des demandes de service issues de l'élément ROSE et par des primitives d'indication issues du service de présentation. Elle émet à son tour des primitives d'indication à destination de l'élément ROSE et des primitives de demande à destination du service de présentation. Les primitives de demande et d'indication P-DATA du service de présentation sont utilisées.

La réception d'une primitive du service STASE-ROSE ou la réception d'une primitive du service de présentation ainsi que la génération des actions précédentes sont des affaires locales qui sont en dehors du domaine d'application de la présente Recommandation.

On suppose qu'une association d'application existe entre les entités d'application homologues durant l'échange des unités APDU. Cette association sera établie en utilisant les paramètres spécifiés au 7.4.2.

NOTE – Toute association d'application peut être identifiée par un système de terminaison au moyen d'un mécanisme interne dépendant de l'implémentation et qui permet à l'utilisateur du service STASE-ROSE (l'élément ROSE) et à la machine SRPM d'y faire référence.

9.1 Définition de la syntaxe abstraite des unités APDU

Les types ASN.1 suivants sont définis pour l'élément STASE-ROSE en plus de ceux définis dans la Recommandation X.229.

```
Secure-Remote-Operations-APDUs {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0) stase-data(2)}
DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTE tout

IMPORTS

ROSEapdus
FROM Remote-Operations-ADPUs {joint-iso-ccitt remote-operations(4) apdus(1)}

AE-title
FROM ACSE-1 {joint-iso-ccitt association-control (2) abstract-syntax(1) apdus(0) version(1)}

DistinguishedName
FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework (1)}

-- le module cité et la syntaxe correspondante sont définis dans l'Annexe D/X.711 – 1998.
```

```
Certificate, CertificationPath
FROM AuthenticationFramework {joint-iso-ccitt ds(5) modules(1) authenticationFramework(7)};

SR-APDU ::= CHOICE
        { clear                     [0] ROSEapdus,
          simpleConfidential        [1] OCTET STRING,
          confidential              [2] Enciphered ,
          simplePublicEnciphered    [3] SimplePublicEnciphered,
          publicEnciphered          [4] PublicEnciphered ,
          hashed                    [5] HashedROSEpdu ,
          sealed                    [6] SealedROSEpdu ,
          signed                    [7] SignedROSEpdu ,
          confidentialSigned        [8] ConfidentialSigned,
         confidentialMAC            [9] ConfidentialMAC,
          confidentialSealed        [10] ConfidentialSealed,
          gssToken                  [11] GssToken,
           ...
          }

Enciphered ::= SEQUENCE
        {encrypted                  OCTET STRING,
         encryptionParameters       EncryptionParameters OPTIONAL
        }

-- encrypted représente une unité PDU ROSE codée selon les règles DER et chiffrée.
-- encryptionParameters représente les paramètres utilisés pour le chiffrement.

SimplePublicEnciphered ::= CHOICE
                          { integers    SEQUENCE OF INTEGER,
                            string      OCTET STRING
                          }

-- SimplePublicEnciphered représente une unité PDU ROSE codée selon les règles DER
-- et chiffrée avec la clé publique.
-- Une unité PDU longue peut être fragmentée en blocs plus petits dont chacun peut être chiffré
-- sous la forme d'un type INTEGER. La taille de ces blocs dépend de l'algorithme de chiffrement
-- par clé publique utilisé et de la taille de la clé publique. La taille de tels blocs est
-- en dehors du domaine d'application de la présente Recommandation.
-- Le résultat du chiffrement par clé publique peut être représenté dans certains cas par un type
-- OCTET STRING.

PublicEnciphered ::= SEQUENCE
                          {publicEncrypted   SimplePublicEnciphered,
                           encryptionParameters     EncryptionParameters OPTIONAL
                          }

-- publicEncrypted représente une unité PDU ROSE codée selon les règles DER
-- et chiffrée avec la clé publique.
-- encryptionParameters représente les paramètres utilisés pour le chiffrement.

Hash ::= SEQUENCE{
              hashValue             OCTET STRING (SIZE(8..64)),
              encryptionParameters  EncryptionParameters OPTIONAL
              }

-- hashValue représente le résumé de message obtenu par hachage de l'unité PDU ROSE
-- codée selon les règles DER.
-- encryptionParameters représente les paramètres utilisés par l'algorithme de hachage.
```

```
HashedROSEpdu ::= SEQUENCE
                {data       OCTET STRING,
                 hash       CHOICE { hash          Hash,
                                     simpleHash    OCTET STRING (SIZE (8..64))
                                     }
        }
```

-- data représente l'unité PDU ROSE codée selon les règles DER.
-- hash représente le résultat du hachage sous la forme d'une chaîne OCTET STRING simple
-- ou de la structure Hash décrite ci-dessus.

```
Seal ::= SEQUENCE
                {sealValue                  OCTET STRING (SIZE(8..128)),
                 encryptionParameters       EncryptionParameters OPTIONAL
                }
```

-- sealValue représente la valeur du scellé pour l'unité PDU ROSE codée selon les règles DER.
-- encryptionParameters représente les paramètres utilisés par l'algorithme de génération du scellé.

```
SealedROSEpdu ::= SEQUENCE
                {data       OCTET STRING,
                 seal       CHOICE {seal          Seal,
                                    simpleSeal    OCTET STRING (SIZE(8..64))
                                    }
        }
```

-- data représente l'unité PDU ROSE codée selon les règles DER.
-- seal représente la valeur du scellé sous la forme d'une chaîne OCTET STRING simple
-- ou de la structure Seal décrite ci-dessus.

```
Signature ::= SEQUENCE
                        {signatureValue             SEQUENCE (SIZE(1..4)) OF INTEGER,
                         encryptionParameters       EncryptionParameters OPTIONAL
                        }
```

-- signatureValue représente la signature de l'unité PDU ROSE codée selon les règles DER.
-- encryptionParameters représente les paramètres de l'algorithme de signature.

```
SignedROSEpdu ::= SEQUENCE
        {data           OCTET STRING,
         signature      CHOICE {signature           [1] Signature,
                                simpleSignature     [2] SEQUENCE (SIZE(1..4)) OF INTEGER
                                }
        }
```

-- data contient le codage de l'unité PDU ROSE selon les règles DER.
-- signature représente la signature de l'unité PDU ROSE codée selon les règles DER,
-- sous la forme d'un type INTEGER simple ou de la structure Signature définie ci-dessus.

```
ConfidentialSigned ::= SEQUENCE
        { encrypted     OCTET STRING,
          signature     CHOICE {signature           [1] Signature,
                                simpleSignature     [2] SEQUENCE (SIZE(1..4)) OF INTEGER
                                }
        }
```

-- encrypted représente le chiffrement de l'unité PDU ROSE codée selon les règles DER.
-- signature représente la signature de l'unité PDU ROSE codée selon les règles DER sous une forme
-- simple ou sous la forme de la structure Signature définie ci-dessus.

```
ConfidentialMAC ::= SEQUENCE
        { encrypted   OCTET STRING,
          mac         CHOICE {mac             [1] Hash,
                              simpleMAC       [2] OCTET STRING (SIZE (8..64))
                              }
        }

-- encrypted représente le chiffrement de l'unité PDU ROSE codée selon les règles DER.
-- mac représente le code MAC de l'unité PDU ROSE codée selon les règles DER sous forme simple
-- ou sous la forme de la structure Hash définie ci-dessus.

ConfidentialSealed ::= SEQUENCE
        { encrypted   OCTET STRING,
          seal        CHOICE {sealed      [1] Seal,
                              simpleSealed  [2] OCTET STRING (SIZE (8..64))
                              }
        }

-- encrypted représente le chiffrement de l'unité PDU ROSE codée selon les règles DER.
-- seal représente le scellé de l'unité PDU ROSE codée selon les règles DER sous une forme simple
-- ou sous la forme de la structure Seal ci-dessus.

EncryptionParameters ::= SET
        {symmetricKeyId         [0] KeyId               OPTIONAL,
        publicKeyId             [1] KeyId               OPTIONAL,
        sealKeyId               [2] KeyId               OPTIONAL,
        signatureKeyId          [3] KeyId               OPTIONAL,
        passwordId              [4] KeyId               OPTIONAL,
         initializationVector   [5] OCTET STRING (SIZE(8))  OPTIONAL,
        feedBackBits            [6] INTEGER (1..63)     OPTIONAL,
        -- pour le mode de feed back de sortie à k bits ou
        -- le mode de feed back de chiffrement à k bits de l'algorithme DES
        symmetricAlgorithm      [7] OBJECT IDENTIFIER   OPTIONAL,
        publicKeyAlgorithm      [8] OBJECT IDENTIFIER   OPTIONAL,
        signatureAlgorithm      [9] OBJECT IDENTIFIER   OPTIONAL,
        sealAlgorithm           [10] OBJECT IDENTIFIER  OPTIONAL,
        hashAlgorithm           [11] OBJECT IDENTIFIER  OPTIONAL,
        keyDigest               [12] OCTET STRING (SIZE(8..64)) OPTIONAL,
        -- pour la vérification des clés publiques
        blockSize               [13] INTEGER            OPTIONAL,
        -- pour le hachage carré modulo n
        keySize                 [14] INTEGER            OPTIONAL,
        -- pour l'algorithme RSA
        publicKey               [15] SEQUENCE
                                        {modulus    INTEGER,
                                         exponent   INTEGER
                                        }               OPTIONAL,
        sequenceNumber          [16] INTEGER            OPTIONAL,
        timeStamp               [17] GeneralizedTime    OPTIONAL,
        encryptedKey            [18] OCTET STRING (SIZE(64..128)) OPTIONAL,
        -- clé de session symétrique, chiffrée en utilisant la clé de chiffrement de clé
        encryptedSymmetricKey   [19] INTEGER    OPTIONAL,
        -- clé de session symétrique, chiffrée en utilisant la clé publique du récepteur
        keyEncryptionKey        [20] SEQUENCE (SIZE (1..3)) OF KeyId        OPTIONAL,
        -- une à trois clés symétriques utilisées pour le chiffrement d'une clé de session
        publicKeyCertificate    [21]    PublicKeyCertificate                OPTIONAL,
-- certificat X.509 ou itinéraire de certification de la clé publique de l'émetteur
-- sans restrictions d'utilisation
        encryptionCertificate   [22] EncryptionCertificate          OPTIONAL,
-- certificat X.509 ou itinéraire de certification de la clé publique de l'émetteur
-- utilisé uniquement pour le chiffrement
```

```
        signatureCertificate        [23] SignatureCertificate                    OPTIONAL,
-- certificat X.509 ou itinéraire de certification de la clé publique de l'émetteur
-- utilisé uniquement pour les signatures numériques
encryptedAuthenticatedSymmetricKey [24] EncryptedAuthenticatedSymmetricKey OPTIONAL,
-- clé de session symétrique, chiffrée en utilisant la clé publique du récepteur
-- et signée avec la clé privée de l'émetteur
        macAlgorithm                [25] OBJECT IDENTIFIER                       OPTIONAL,
                ...
                }

-- le type extensible EncryptionParameters est utilisé comme conteneur pour tous les paramètres
-- pouvant être utilisés pour toutes les transformations de sécurité. La plupart des applications
-- n'utiliseront aucune ou une faible partie des composants du type EncryptionParameters.

KeyId ::= CHOICE        {
                         name        GraphicString,
                         number      INTEGER
                        }

PublicKeyCertificate ::= CHOICE {certificate            [0] Certificate,
                                 certificationPath      [1] CertificationPath
                                 }

EncryptionCertificate ::= CHOICE {certificate           [0] Certificate,
                                  certificationPath     [1] CertificationPath
                                  }

SignatureCertificate ::= CHOICE {certificate            [0] Certificate,
                                 certificationPath      [1] CertificationPath
                                }

EncryptedAuthenticatedSymmetricKey ::= SEQUENCE {
                                 encryptedSymmetricKey  INTEGER,
        -- clé de session symétrique, chiffrée en utilisant la clé publique du récepteur
                                 time           GeneralizedTime,
                                 sender         SenderId,
                                 receiver       ReceiverId,
                                 signature      Signature
-- la signature est calculée à partir de la représentation ASCII des quatre champs précédents en utilisant la clé
-- privée de l'émetteur
                                 }
SenderId ::= CHOICE {
                        identifier      [1] DistinguishedName,
                        name            [2] GraphicString,
                        application     [3] AE-title
                        }

ReceiverId ::= SenderId

GssToken::=CHOICE {
        micToken    [1]     MicToken ,
        wrapToken   [2]     OCTET STRING
    }

MicToken::=SEQUENCE {
        rosePDU     [1]     OCTET STRING,
        token       [2]     OCTET STRING
    }

END
```

9.2 Nom de syntaxe abstraite

La présente Recommandation attribue l'identificateur suivant:

{itu-t recommendation q(17) q813(813) stase(1) abstractSyntax(1) stase-data(2)}

comme nom de syntaxe abstraite pour l'ensemble de valeurs de données de présentation, chacune des valeurs étant du type ASN.1
Secure-Remote-Operations-APDUs.SR-APDU

dans lequel les composantes d'argument des unités PDU ROSE sont fournies par l'utilisateur de l'élément ROSE.

La valeur de descripteur d'objet correspondante sera "STASE-ROSE-Data".

9.3 Identificateurs d'algorithme

Sauf accord contraire conclu entre les entités communicantes (par des moyens qui sont en dehors du domaine d'application de la présente Recommandation), les algorithmes de transformation de sécurité seront limités à ceux qui sont indiqués dans l'ISO/CEI 9979 et seront identifiés par les identificateurs d'objet fournis dans cette norme internationale.

9.4 Noms de contextes d'application

9.4.1 Contexte sécurisé de RGT

La valeur d'identificateur d'objet suivante sera attribuée au nom du contexte d'application dont l'entité d'application se constitue des éléments SMASE, CMISE, ROSE, STASE-ROSE et ACSE:

{itu-t recommendation q(17) q813(813) stase(1) stase-application-context (2) secureTMNContext(0)}

et la valeur du descripteur d'objet correspondante sera "Contexte sécurisé d'application interactive du RGT".

Le champ "informations utilisateur" de l'élément ACSE défini dans la Rec. UIT-T X.227 | ISO/CEI 8650-1 se constitue d'une expression SEQUENCE OF EXTERNAL. La présente Recommandation spécifie l'ordre suivant des expressions EXTERNAL figurant dans le champ "informations utilisateur" de l'élément ACSE:

– données fournies éventuellement pour l'élément STASE-ROSE;
– données fournies éventuellement pour l'élément CMISE, telles qu'elles sont définies dans la Rec. UIT-T X.711 | ISO/CEI 9596-1;
– données fournies éventuellement pour l'élément SMASE, telles qu'elles sont définies dans la Rec. UIT-T X.701 | ISO/CEI 10040.

9.4.2 Contexte d'application d'annuaire sécurisé

La valeur d'identificateur d'objet suivante sera attribuée au nom du contexte d'application dont l'entité d'application se constitue des éléments ou entités X.500, ROSE, STASE-ROSE et ACSE:

{itu-t recommendation q(17) q813(813) stase(1) stase-application-context (2) secureDirectoryContext(1)}

et la valeur du descripteur d'objet correspondante sera "Secure-Directory-Application-Context".

9.5 Procédures du service STASE-ROSE

Le protocole STASE-ROSE fournit l'élément de procédure suivant:

transfert.

Les sous-paragraphes qui suivent fournissent, pour cet élément de procédure, un résumé constitué d'un sommaire des unités APDU adéquates, d'un aperçu à haut niveau des primitives du service STASE-ROSE, des unités APDU impliquées et du service de transfert utilisé.

9.5.1 Transfert

9.5.1.1 But

La procédure de transfert est invoquée par un utilisateur du service STASE-ROSE (élément ROSE) pour transférer une unité APDU ROSE de manière sécurisée. L'élément STASE-ROSE appliquera à l'unité PDU ROSE les transformations de sécurité nécessaires et la transférera à l'élément STASE-ROSE homologue.

9.5.1.2 Unités APDU utilisées

Le transfert utilise l'unité APDU STASE-ROSE.

Le Tableau 9-1 qui suit donne la liste des unités APDU STASE-ROSE (unités APDU SR). Un seul des champs sera utilisé dans une unité APDU SR unique. Le sous-paragraphe 9.5.1.4 décrit l'utilisation des champs de l'unité APDU SR. Voir le 9.1 en ce qui concerne la définition ASN.1 de l'unité APDU SR.

Tableau 9-1/Q.813 – Champs des unités APDU STASE-ROSE

Nom du champ	Source	Puits
clear	demande	indication
simpleConfidential	demande	indication
confidential	demande	indication
simplePublicEnciphered	demande	indication
publicEnciphered	demande	indication
hashed	demande	indication
sealed	demande	indication
signed	demande	indication
confidentialSigned	demande	indication
confidentialMAC	demande	indication
confidentialSealed	demande	indication

9.5.1.3 Procédure de transfert

Cette procédure est activée par les événements suivants:

a) primitive de demande SR-TRANSFER en provenance du demandeur;

b) primitive d'indication P-DATA en provenance du service de présentation.

9.5.1.3.1 Aperçu général

Dans un environnement de système ouvert, tout système peut utiliser sa propre représentation interne pour des éléments d'information tels que des caractères ou des nombres entiers et réels. Il est possible de spécifier de tels éléments d'information en utilisant la notation ASN.1 et de les échanger en utilisant une syntaxe de transfert (selon les règles BER ou DER) ayant fait l'objet d'un accord préalable lorsqu'une communication entre systèmes ouverts hétérogènes est prise en charge. Si une transformation de sécurité telle que le chiffrement est effectuée sur un élément d'information dans la

couche Application en utilisant la représentation de données internes du système (par exemple, une représentation ASCII des caractères), le résultat de cette transformation sera alors incompréhensible pour tout autre système ouvert utilisant une représentation différente au niveau de la couche Application (par exemple, une représentation EBCDIC des caractères). Les transformations de sécurité doivent, pour cette raison, être effectuées sur une syntaxe de transfert d'éléments d'information.

La présente Recommandation spécifie que les transformations de sécurité sont effectuées sur des unités PDU ROSE codées selon les règles DER.

9.5.1.3.2 Primitive de demande SR-TRANSFER

La machine SRPM demandeuse génère une unité APDU SR en utilisant les valeurs de paramètre de la primitive de demande SR-TRANSFER.

Les champs de l'unité APDU SR sont produits comme suit:

1) si la valeur du paramètre "type de chiffrement" (voir 7.4.4) est égale à "clair", le paramètre de l'unité PDU ROSE sera alors assigné directement au champ **clear** de l'unité **APDU SR**;

2) les procédures suivantes sont appliquées pour toute autre valeur du paramètre "type de chiffrement":

   – l'élément STASE-ROSE codera tout d'abord l'unité PDU ROSE selon les règles DER.

   • Si une protection de la confidentialité de l'unité PDU ROSE est exigée (c'est-à-dire si la valeur du paramètre "type de chiffrement" est égale à "confidentiel simple", "confidentiel", "chiffré public simple" ou "chiffré public") l'élément STASE-ROSE chiffrera alors le flux d'octets codé au préalable selon les règles DER (*sous forme DER*) et assignera le résultat à l'un des champs de l'unité **APDU SR** de la manière suivante:

      – si la valeur du paramètre "type de chiffrement" est égale à "confidentiel simple", le flux d'octets sous forme DER sera chiffré en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. Les données chiffrées sont assignées au champ **simpleConfidential** de la structure de l'unité APDU SR;

      – si la valeur du paramètre "type de chiffrement" est égale à "confidentiel", le flux d'octets sous forme DER sera chiffré en utilisant les informations fournies par le paramètre "paramètres de chiffrement". Les données chiffrées et les paramètres de chiffrement seront assignés respectivement aux champs **confidential.encrypted** et **confidential.encryptionParameters** de la structure de l'unité APDU SR;

      – si la valeur du paramètre "type de chiffrement" est égale à "chiffré public simple", le flux d'octets sous forme DER sera chiffré en utilisant l'information de clé publique par défaut comme décrit au 5.2. Les données chiffrées sont assignées au champ **simplePublicEnciphered** de la structure de l'unité APDU SR;

      – si la valeur du paramètre "type de chiffrement" est égale à "chiffré public", le flux d'octets sous forme DER sera chiffré en utilisant le paramètre "paramètres de chiffrement". Les données chiffrées et les paramètres de chiffrement sont assignés respectivement aux champs **enciphered.publicEncrypted** et **enciphered.encryptionParameters** de la structure de l'unité APDU SR.

   • Si un contrôle de l'intégrité est exigé (c'est-à-dire si la valeur du paramètre "type de chiffrement" est égale à "haché simple", "haché", "scellé simple" ou "scellé"), l'élément STASE-ROSE calculera alors le scellé numérique ou le hachage du flux d'octets sous forme DER et assignera le résultat à l'un des champs de l'unité **APDU SR** de la manière suivante:

– si la valeur du paramètre "type de chiffrement" est égale à "haché simple", le flux d'octets sous forme DER est haché en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. Le flux d'octets sous forme DER et la valeur du résultat de hachage sont assignés respectivement aux champs **hashed.data** et **hashed.hash.simpleHash** de la structure de l'unité **APDU SR**;

– si la valeur du paramètre "type de chiffrement" est égale à "haché", le flux d'octets sous forme DER est haché en utilisant les informations fournies par le paramètre "paramètres de chiffrement". Le flux d'octets sous forme DER, sa valeur de hachage et les paramètres de chiffrement sont assignés respectivement aux champs **hashed.data**, **hashed.hash.hash.hashValue** et **hashed.hash.hash.encryptionParameters** de la structure de l'unité **APDU SR**;

– si la valeur du paramètre "type de chiffrement" est égale à "scellé simple", le flux d'octets sous forme DER est scellé en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. Le flux d'octets sous forme DER et sa valeur de scellé sont assignés respectivement aux champs **sealed.data** et **sealed.seal.simpleSeal** de la structure de l'unité **APDU SR**;

– si la valeur du paramètre "type de chiffrement" est égale à "scellé", le flux d'octets sous forme DER est scellé en utilisant les informations fournies par le paramètre "paramètres de chiffrement". Le flux d'octets sous forme DER, sa valeur de scellé et les paramètres de chiffrement sont assignés respectivement aux champs **sealed.data**, **sealed.seal.seal.sealValue** et **sealed.seal.seal.encryptionParameters** de la structure de l'unité **APDU SR**.

• Si la non-répudiation est exigée (c'est-à-dire si la valeur du paramètre "type de chiffrement" est égale à "signé simple" ou "signé"), l'élément STASE-ROSE calculera alors la signature numérique du flux d'octets sous forme DER et assignera le résultat à l'un des champs de l'unité APDU SR de la manière suivante:

– si la valeur du paramètre "type de chiffrement" est égale à "signé simple", la signature numérique du flux d'octets sous forme DER est calculée en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. Le flux d'octets sous forme DER et sa signature sont assignés respectivement aux champs **signed.data** et **signed.signature.simpleSignature** de la structure de l'unité **APDU SR**;

– si la valeur du paramètre "type de chiffrement" est égale à "signé", la signature numérique du flux d'octets, sous forme DER, est calculée en utilisant les informations fournies par le paramètre "paramètres de chiffrement". Le flux d'octets sous forme DER, sa signature et les paramètres de chiffrement sont assignés respectivement aux champs **signed.data, signed.signature.signature.signatureValue** et **signed.signature.signature.encryptionParameters** de la structure de l'unité **APDU SR**.

• Si la confidentialité et la non-répudiation sont exigées toutes deux (c'est-à-dire si la valeur du paramètre "type de chiffrement" est égale à "signé confidentiel simple" ou "signé confidentiel"), l'élément STASE-ROSE calculera alors le chiffrement et la signature de l'unité PDU sous forme DER. Sauf accord contraire conclu entre les entités communicantes par des moyens qui sont en dehors du domaine d'application de la présente Recommandation, la signature sera calculée sur la valeur en clair, c'est-à-dire non chiffrée, de l'unité PDU ROSE sous forme DER. Le résultat du chiffrement et de la signature sera assigné à la structure de l'unité **APDU SR** de la manière suivante:

– si la valeur du paramètre "type de chiffrement" est égale à "signé confidentiel simple", la signature numérique et la valeur chiffrée du flux d'octets sous forme DER, sont calculées en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. Les données chiffrées et la signature numérique seront assignées respectivement aux champs **confidentialSigned.encrypted** et **confidentialSigned.signature.simpleSignature** de la structure de l'unité **APDU SR**;

– si la valeur du paramètre "type de chiffrement" est égale à "signé confidentiel", la signature numérique et la valeur chiffrée du flux d'octets sous forme DER sont calculées en utilisant les informations fournies par le paramètre "paramètres de chiffrement". Les données chiffrées, la signature numérique et les paramètres de chiffrement seront assignés respectivement aux champs **confidentialSigned.encrypted**, **confidentialSigned.signature.signature.signatureValue** et **confidentialSigned.signature.signature.encryptionParameters** de la structure de l'unité **APDU SR**.

• Si la confidentialité et l'intégrité sont exigées toutes deux (c'est-à-dire si la valeur du paramètre "type de chiffrement" est égale à "MAC confidentiel simple" ou "MAC confidentiel"), l'élément STASE-ROSE calculera alors le chiffrement et le code MAC de l'unité PDU sous forme DER. Sauf accord contraire conclu entre les entités communicantes par des moyens qui sont en dehors du domaine d'application de la présente Recommandation, le code MAC sera calculé sur la valeur en clair, c'est-à-dire non chiffrée, de l'unité PDU ROSE sous forme DER. Le résultat du chiffrement et de la signature sera assigné à la structure de l'unité **APDU SR** de la manière suivante:

– si la valeur du paramètre "type de chiffrement" est égale à "MAC confidentiel simple", le code MAC et la valeur chiffrée du flux d'octets sous forme DER sont calculés en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. Les données chiffrées et le code MAC seront assignés respectivement aux champs **confidentialMAC.encrypted** et **confidentialMAC.mac.simpleMAC** de la structure de l'unité **APDU SR**;

– si la valeur du paramètre "type de chiffrement" est égale à "MAC confidentiel", le code MAC et la valeur chiffrée du flux d'octets sous forme DER sont calculés en utilisant les informations fournies par le paramètre "paramètres de chiffrement". Les données chiffrées, le code MAC et les paramètres de chiffrement seront assignés respectivement aux champs **confidentialMAC.encrypted**, **confidentialMAC.mac.mac.hashValue** et **confidentialMAC.mac.mac.encryptionParameters** de la structure de l'unité **APDU SR**.

• Si la confidentialité et le scellé numérique sont exigés tous deux (c'est-à-dire si la valeur du paramètre "type de chiffrement" est égale à "scellé confidentiel simple" ou "scellé confidentiel"), l'élément STASE-ROSE calculera alors le chiffrement et le scellé de l'unité PDU sous forme DER. Sauf accord contraire conclu entre les entités communicantes par des moyens qui sont en dehors du domaine d'application de la présente Recommandation, le scellé sera calculé sur la valeur en clair, c'est-à-dire non chiffrée, de l'unité PDU ROSE sous forme DER. Le résultat du chiffrement et de la signature sera assigné à la structure de l'unité **APDU SR** de la manière suivante:

– si la valeur du paramètre "type de chiffrement" est égale à "scellé confidentiel simple", le scellé et la valeur chiffrée du flux d'octets sous forme DER sont calculés en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. Les données

chiffrées et le scellé seront assignés respectivement aux champs **confidentialSealed.encrypted** et **confidentialSealed.seal.simpleSealed** de la structure de l'unité **APDU SR**;

– si la valeur du paramètre "type de chiffrement" est égale à "scellé confidentiel", le scellé et la valeur chiffrée du flux d'octets sous forme DER sont calculés en utilisant les informations fournies par le paramètre "paramètres de chiffrement". Les données chiffrées, le scellé et les paramètres de chiffrement seront assignés respectivement aux champs **confidentialSealed.encrypted**, **confidentialSealed.seal.seal.sealValue** et **confidentialSealed.seal.seal.encryptionParameters** de la structure de l'unité **APDU SR**.

L'unité APDU SR ainsi constituée est transférée vers l'élément STASE-ROSE homologue dans le paramètre "données utilisateur" de la primitive de demande de transfert P-DATA du service de présentation.

La machine SRPM passe alors en attente d'une primitive d'indication P-DATA en provenance de la couche Présentation ou d'une primitive de demande SR-TRANSFER en provenance du demandeur.

9.5.1.3.3 Primitive d'indication P-DATA

La machine SRPM acceptante reçoit de son homologue une unité APDU SR dans les données utilisateur d'une primitive d'indication de transfert P-DATA. L'élément STASE-ROSE appliquera les procédures suivantes pour récupérer les primitives d'indication SR-TRANSFER:

1) la valeur du paramètre "type de chiffrement" sera positionnée sur "clair" et le paramètre de l'unité PDU ROSE sera positionné sur le champ **clear** de l'unité **APDU SR** si ce champ a été sélectionné dans l'unité entrante;

2) la procédure suivante est appliquée si tout autre champ est sélectionné dans l'unité APDU entrante:

   • si le champ **simpleConfidential** a été sélectionné, le flux d'octets sous forme DER qui correspond à l'unité PDU ROSE sera récupéré par déchiffrement du champ **simpleConfidential** en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. La valeur du paramètre "type de chiffrement" sera égale à "confidentiel simple". La procédure à appliquer en cas d'échec du déchiffrement est décrite à la fin du présent sous-paragraphe;

   • si le champ **confidential** a été sélectionné, le flux d'octets sous forme DER qui correspond à l'unité PDU ROSE sera récupéré par déchiffrement du champ **confidential.encrypted** en utilisant le champ **confidential.encryptionParameters**. La valeur du paramètre "type de chiffrement" sera égale à "confidentiel". La valeur du paramètre "paramètres de chiffrement" sera égale à celle du champ **confidential.encryptionParameters**. La procédure à appliquer en cas d'échec du déchiffrement est décrite à la fin du présent sous-paragraphe;

   • si le champ **simplePublicEnciphered** a été sélectionné, le flux d'octets sous forme DER qui correspond à l'unité PDU ROSE sera récupéré par déchiffrement du champ **simplePublicEnciphered** en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. La valeur du paramètre "type de chiffrement" sera égale à "chiffré public simple". La procédure à appliquer en cas d'échec du déchiffrement est décrite à la fin du présent sous-paragraphe;

   • si le champ **publicEnciphered** a été sélectionné, le flux d'octets sous forme DER qui correspond à l'unité PDU ROSE sera récupéré par déchiffrement du champ **publicEnciphered.publicEncrypted** en utilisant le champ **publicEnciphered.encryptionParameters**. La valeur du paramètre "type de

chiffrement" sera égale à "confidentiel". La valeur du paramètre "paramètres de chiffrement" sera égale à celle du champ **publicEnciphered.encryptionParameters**. La procédure à appliquer en cas d'échec du déchiffrement est décrite à la fin du présent sous-paragraphe.

- Si le champ **hashed** a été sélectionné dans l'unité APDU reçue et:
  - si le champ **hashed.hash.simpleHash** a été sélectionné, la valeur du champ **hashed.data** sera alors utilisée comme codage DER de l'unité PDU ROSE. La valeur du hachage du flux d'octets sous forme DER sera calculée en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2 et sera comparée à celle du champ **hashed.hash.simpleHash**. La valeur du paramètre "type de chiffrement" sera égale à "haché simple". La procédure à appliquer en cas d'échec de la comparaison de la valeur du hachage est décrite à la fin du présent sous-paragraphe;
  - si le champ **hashed.hash.hash** a été sélectionné, la valeur du champ **hashed.data** sera alors utilisée comme codage DER de l'unité PDU ROSE. La valeur du hachage du flux d'octets sous forme DER sera calculée en utilisant le champ **hashed.hash.hash.encryptionParameters** et sera comparée à celle du champ **hashed.hash.hash.hashValue**. La valeur du paramètre "type de chiffrement" sera égale à "haché". La valeur du paramètre "paramètres de chiffrement " sera assignée à la valeur du champ **hashed.hash.hash.encryptionParameters**. La procédure à appliquer en cas d'échec de la comparaison de la valeur du hachage est décrite à la fin du présent sous-paragraphe.
- Si le champ **sealed** a été sélectionné dans l'unité APDU reçue et:
  - si le champ **sealed.seal.simpleSeal** a été sélectionné, la valeur du champ **sealed.data** sera alors utilisée comme codage DER de l'unité PDU ROSE. La valeur du scellé numérique du flux d'octets sous forme DER sera calculée en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2 et sera comparée à celle du champ **sealed.seal.simpleSeal**. La valeur du paramètre "type de chiffrement" sera égale à "scellé simple". La procédure à appliquer en cas d'échec de la comparaison de la valeur du scellé est décrite à la fin du présent sous-paragraphe;
  - si le champ **sealed.seal.seal** a été sélectionné, la valeur du champ **sealed.data** sera alors utilisée comme codage DER de l'unité PDU ROSE. La valeur du scellé numérique du flux d'octets sous forme DER sera calculée en utilisant le champ **sealed.seal.seal.encryptionParameters** et sera comparée à celle du champ **sealed.seal.seal.sealValue**. La valeur du paramètre "type de chiffrement" sera égale à "scellé". La valeur du paramètre "paramètres de chiffrement" sera assignée à la valeur du champ **sealed.seal.seal.encryptionParameters**. La procédure à appliquer en cas d'échec de la comparaison de la valeur du scellé est décrite à la fin du présent sous-paragraphe.
- Si le champ **signed** a été sélectionné dans l'unité APDU reçue et:
  - si le champ **signed.signature.simpleSignature** a été sélectionné, la valeur du champ **signed.data** sera alors utilisée comme codage DER de l'unité PDU ROSE. La signature numérique du flux d'octets sous forme DER sera calculée en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2 et sera comparée à celle du champ **signed.signature.simpleSignature**. La valeur du paramètre "type de chiffrement" sera égale à "signé simple". La procédure à appliquer en cas d'échec de la comparaison de la valeur de la signature numérique est décrite à la fin du présent sous-paragraphe;

– si le champ **signed.signature.signature** a été sélectionné, la valeur du champ **signed.data** sera alors utilisée comme codage DER de l'unité PDU ROSE. La signature numérique du flux d'octets sous forme DER sera calculée en utilisant le champ **signed.signature.signature.encryptionParameters** et sera comparée à celle du champ **signed.signature.signature.signatureValue**. La valeur du paramètre "type de chiffrement" sera égale à "signé". La valeur du paramètre "paramètres de chiffrement" sera assignée à la valeur du champ **signed.signature.signature.encryptionParameters**. La procédure à appliquer en cas d'échec de la comparaison de la valeur de la signature numérique est décrite à la fin du présent sous-paragraphe.

- Si le champ **confidentialSigned** a été sélectionné dans l'unité APDU reçue et:
  – si le champ **confidentialSigned.signature.simpleSignature** a été sélectionné, le flux d'octets sous forme DER qui correspond à l'unité PDU ROSE sera récupéré par déchiffrement du champ **confidentialSigned.encrypted** en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. La signature numérique du flux d'octets sous forme DER sera calculée en utilisant les valeurs par défaut et sera comparée à celle du champ **confidentialSigned.signature.simpleSignature**. La valeur du paramètre "type de chiffrement" sera positionnée sur "signé confidentiel simple". La procédure à appliquer en cas d'échec du déchiffrement ou de la comparaison de la valeur de la signature numérique est décrite à la fin du présent sous-paragraphe;
  – si le champ **confidentialSigned.signature.signature** a été sélectionné, le flux d'octets sous forme DER qui correspond à l'unité PDU ROSE sera récupéré par déchiffrement du champ **confidentialSigned.encrypted** en utilisant le champ **confidentialSigned.signature.signature.encryptionParameters**. La signature numérique du flux d'octets sous forme DER sera calculée en utilisant le champ **confidentialSigned.signature.signature.encryptionParameters** et sera comparée à celle du champ **confidentialSigned.signature.signature.signatureValue**. La valeur du paramètre "type de chiffrement" sera positionnée sur "signé confidentiel" et la valeur du paramètre "paramètres de chiffrement" sera positionnée sur **confidentialSigned.signature.signature.encryptionParameters**. La procédure à appliquer en cas d'échec du déchiffrement ou de la comparaison de la valeur de la signature numérique est décrite à la fin du présent sous-paragraphe.
- Si le champ **confidentialMAC** a été sélectionné dans l'unité APDU reçue et:
  – si le champ **confidentialMAC.mac.simpleMAC** a été sélectionné, le flux d'octets sous forme DER qui correspond à l'unité PDU ROSE sera récupéré par déchiffrement du champ **confidentialMAC.encrypted** en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. La valeur du code MAC du flux d'octets sous forme DER sera calculée en utilisant les valeurs par défaut et sera comparée à celle du champ **confidentialMAC.mac.simpleMAC**. La valeur du paramètre "type de chiffrement" sera positionnée sur "MAC confidentiel simple". La procédure à appliquer en cas d'échec du déchiffrement ou de la comparaison du code MAC est décrite à la fin du présent sous-paragraphe;
  – si le champ **confidentialMAC.mac.mac** a été sélectionné, le flux d'octets sous forme DER qui correspond à l'unité PDU ROSE sera récupéré par déchiffrement du champ **confidentialMAC.encrypted** en utilisant le champ **confidentialMAC.mac.mac.encryptionParameters**. La valeur du code MAC du flux d'octets sous forme DER sera calculée en utilisant le champ **confidentialMAC.mac.mac.encryptionParameters** et sera comparée à celle du champ **confidentialMAC.mac.mac.hashValue**. La valeur du paramètre "type de

chiffrement" sera positionnée sur "MAC confidentiel" et la valeur du paramètre "paramètres de chiffrement" sera positionnée sur **confidentialMAC.mac.mac.encryptionParameters**. La procédure à appliquer en cas d'échec du déchiffrement ou de la comparaison du code MAC est décrite à la fin du présent sous-paragraphe.

- Si le champ **confidentialSealed** a été sélectionné dans l'unité APDU reçue et:
  - si le champ **confidentialSealed.seal.simpleSealed** a été sélectionné, le flux d'octets sous forme DER qui correspond à l'unité PDU ROSE sera récupéré par déchiffrement du champ **confidentialSealed.encrypted** en utilisant les valeurs par défaut telles qu'elles sont décrites au 5.2. La valeur du scellé du flux d'octets sous forme DER sera calculée en utilisant les valeurs par défaut et sera comparée à celle du champ **confidentialSealed.seal.simpleSealed**. La valeur du paramètre "type de chiffrement" sera positionnée sur "scellé confidentiel simple". La procédure à appliquer en cas d'échec du déchiffrement ou de la comparaison du scellé est décrite à la fin du présent sous-paragraphe;
  - si le champ **confidentialSealed.seal.seal** a été sélectionné, le flux d'octets sous forme DER qui correspond à l'unité PDU ROSE sera récupéré par déchiffrement du champ **confidentialSealed.encrypted** en utilisant le champ **confidentialSealed.seal.seal.encryptionParameters**. Le scellé du flux d'octets sous forme DER sera calculé en utilisant le champ **confidentialSealed.seal.seal.encryptionParameters** et sera comparé à la valeur du champ **confidentialSealed.seal.seal.sealValue**. La valeur du paramètre "type de chiffrement" sera positionnée sur "scellé confidentiel" et la valeur du paramètre "paramètres de chiffrement" sera positionnée sur **confidentialSealed.seal.seal.encryptionParameters**. La procédure à appliquer en cas d'échec du déchiffrement ou de la comparaison du scellé est décrite à la fin du présent sous-paragraphe.
- L'élément STASE-ROSE décodera le flux d'octets sous forme DER qui est récupéré dans l'unité APDU reçue et assignera le résultat au paramètre de l'unité PDU ROSE.

La machine SRPM émettra une primitive d'indication SR-TRANSFER à destination de l'accepteur contenant les paramètres récupérés si elle a réussi à exécuter correctement les procédures décrites ci-dessus.

La procédure à appliquer si l'une quelconque des valeurs de paramètre de l'unité APDU SR n'est pas acceptable pour la machine SRPM est une affaire locale. La présente Recommandation recommande toutefois que l'implémentation de la machine SRPM prenne l'une des deux actions suivantes lorsqu'une unité APDU SR reçue n'est pas acceptable.

1) La machine SRPM rejettera l'unité APDU SR. L'entité d'application locale associée peut être informée, d'une manière qui dépend de l'implémentation, qu'une unité APDU non acceptable a été reçue et rejetée.

2) La machine SRPM invoquera le service A-ABORT fourni par l'élément de commande d'association pour abandonner l'association. L'entité d'application locale associée peut être informée, d'une manière qui dépend de l'implémentation, qu'une unité APDU non acceptable a été reçue et rejetée et que l'association d'application a été abandonnée.

Dans l'un ou l'autre cas, il est recommandé que l'événement soit journalisé dans une trace de vérification de sécurité et qu'une alarme de sécurité soit émise à destination de l'administrateur de sécurité local.

La machine SRPM acceptante passe en attente d'une primitive d'indication P-DATA en provenance de la couche Présentation ou d'une primitive de demande SR-TRANSFER en provenance de l'utilisateur du service.

9.5.1.4 Utilisation des champs de l'unité APDU SR

Les champs de l'unité APDU SR sont utilisés de la manière suivante:

1) **clear** (*clair*): ce champ est utilisé si aucune transformation de sécurité n'est demandée par l'utilisateur de l'élément STASE-ROSE.

2) **simpleConfidential** (*confidentiel simple*): ce champ est utilisé si une protection de la confidentialité est demandée par l'utilisateur de l'élément STASE-ROSE et si les paramètres de chiffrement par défaut sont utilisés.

3) **confidential** (*confidentiel*): ce champ est utilisé si une protection de la confidentialité est demandée par l'utilisateur de l'élément STASE-ROSE et si les paramètres de chiffrement sont fournis par cet utilisateur.

4) **simplePublicEnciphered** (*chiffré public simple*): ce champ est utilisé lorsqu'un chiffrement par clé publique est demandé par l'utilisateur de l'élément STASE-ROSE et si les paramètres de chiffrement par défaut sont utilisés.

5) **publicEnciphered** (*chiffré public*): ce champ est utilisé lorsqu'un chiffrement par clé publique est demandé par l'utilisateur de l'élément STASE-ROSE et si les paramètres de chiffrement sont fournis par cet utilisateur.

6) **hashed** (*haché*): ce champ est utilisé lorsqu'une protection par hachage est demandée par l'utilisateur de l'élément STASE-ROSE.

7) **sealed** (*scellé*): ce champ est utilisé lorsqu'une protection par scellé numérique est demandée par l'utilisateur de l'élément STASE-ROSE.

8) **signed** (*scellé*): ce champ est utilisé lorsque la non-répudiation est demandée par l'utilisateur de l'élément STASE-ROSE.

9) **confidentialSigned** (*signé confidentiel*): ce champ est utilisé lorsque la non-répudiation et la protection de la confidentialité sont demandées par l'utilisateur de l'élément STASE-ROSE.

10) **confidentialMAC** (*MAC confidentiel*): ce champ est utilisé lorsque la protection de l'intégrité par hachage et la protection de la confidentialité sont demandées par l'utilisateur de l'élément STASE-ROSE.

11) **confidentialSealed** (*scellé confidentiel*): ce champ est utilisé lorsque la protection de l'intégrité par sceau et la protection de la confidentialité sont demandées par l'utilisateur de l'élément STASE-ROSE.

9.6 Mappage du service STASE-ROSE sur service de présentation

Le présent sous-paragraphe définit de quelle manière les primitives du service de présentation décrites dans la Recommandation X.216 sont utilisées par la machine SRPM. Le Tableau 9-2 qui suit définit le mappage des primitives et des unités APDU du service STASE-ROSE sur les primitives du service de présentation.

Le service P-DATA est un service sans confirmation. L'utilisation des paramètres des primitives d'indication et de demande P-DATA est la suivante:

- **données utilisateur**: unité APDU à transférer. Le présent mappage n'impose pas de contrainte de taille maximale.

Tableau 9-2/Q.813 – Aperçu général du mappage du service de présentation

Service STASE-ROSE	Unité APDU	Service de présentation
demande ou indication SR-TRANSFER	APDU SR	demande ou indication P-DATA

10 Mappage des services ROSE sur les services de l'élément STASE-ROSE

Le présent paragraphe décrit de quelle manière les primitives du service STASE-ROSE décrites dans la présente Recommandation sont utilisées par les services ROSE définis dans la Recommandation X.219. Le mappage est donné dans le tableau qui suit:

Tableau 10-1/Q.813 – Mappage des services ROSE sur le service STASE-ROSE

Service ROSE	Unité APDU	Service STASE-ROSE
demande ou indication RO-INVOKE	ROIV	demande ou indication SR-TRANSFER
demande ou indication RO-RESULT	RORS	demande ou indication SR-TRANSFER
demande ou indication RO-ERROR	ROER	demande ou indication SR-TRANSFER
demande ou indication RO-REJECT-U	RORJ	demande ou indication SR-TRANSFER
demande ou indication RO-REJECT-P	RORJ	demande ou indication SR-TRANSFER

Le service SR-TRANSFER est un service sans confirmation.

11 Conformité

Une implémentation qui déclare la conformité à la présente Recommandation se conformera aux prescriptions suivantes:

- **prescriptions de déclaration**: l'implémentation fera les déclarations suivantes:
  a) contexte d'application pour lequel la conformité est déclarée;
  b) prise en charge, ou non, de la négociation de paramètres de sécurité au moment de l'établissement de l'association;
  c) algorithmes de transformation de sécurité éventuels fournis par l'implémentation et possibilité d'utilisation par l'implémentation d'algorithmes de transformation de sécurité supplémentaires fournis par l'utilisateur;
- **conformité statique**: le système:
  d) se conformera à la définition de la syntaxe abstraite des unités APDU figurant au paragraphe 9;

     prendra en charge les règles de codage distinctives spécifiées dans la Recommandation X.690 avec l'identificateur d'objet {joint-iso-ccitt asn1(1) ber-derived(2) distinguished-encoding(1)} et le descripteur d'objet "Codage distinctif d'un type ASN.1 unique" aux fins de génération et d'interprétation des informations du protocole d'application (par exemple, les informations du protocole de l'élément CMISE) en plus de l'utilisation des règles BER dans la couche Présentation pour le codage des unités PDU STASE-ROSE;
  e) prendra en charge le protocole de l'élément ACSE, défini dans la Recommandation X.227, pour établir et supprimer une association;

- **conformité dynamique**: le système:

  f) se conformera aux éléments de procédure définis dans le paragraphe 9;

  g) se conformera, comme défini dans les paragraphes 9 et 10, aux mappages d'utilisation des services pour lesquels la conformité est déclarée.

12 Tables d'états de la machine SRPM

Le présent paragraphe décrit une machine de protocole STASE-ROSE (SRPM) unique sous la forme d'une table d'états. La table d'états décrit la relation entre l'état d'une association d'application, les événements entrants qui surviennent dans le protocole, les actions effectuées et l'état résultant de l'association d'application.

La table d'états de la machine SRPM ne constitue pas une définition formelle de la machine SRPM. Elle est utilisée pour fournir une spécification plus précise des procédures définies au paragraphe 9.

Le présent paragraphe contient les tableaux suivants:

a) le Tableau 12-1 donne le nom abrégé, la source, la description et le nom de chaque événement entrant. Les sources sont les suivantes:
   - utilisateur du service STASE-ROSE (utilisateur SR);
   - élément ACSE (ACSE);
   - fournisseur du service de présentation (fournisseur PS);

b) le Tableau 12-2 donne le nom abrégé de chaque état de la machine SRPM;

c) le Tableau 12-3 donne le nom abrégé, la cible, le nom et la description de chaque événement entrant. Les cibles sont les suivantes:
   - utilisateur du service STASE-ROSE (utilisateur SR);
   - élément ACSE (ACSE);
   - fournisseur du service de présentation (fournisseur PS);

d) le Tableau 12-4 donne les prédicats;

e) le Tableau 12-5 spécifie la table d'états de la machine SRPM en utilisant les abréviations définies dans les tables précédentes.

Tableau 12-1/Q.813 – Liste d'événements entrants

Nom abrégé	Source	Nom et description
AA-ESTAB	ACSE	primitive de réponse A-ASSOCIATE positive ou primitive de confirmation A-ASSOCIATE positive
SRreq	utilisateur SR	primitive de demande SR-TRANSFER
APDUua	machine SRPM homologue	unité APDU non acceptable dans les données utilisateur d'une indication P-DATA
P-DATAind	fournisseur PS	primitive d'indication P-DATA
AA-REL	ACSE	primitive de réponse A-RELEASE positive ou primitive de confirmation A-RELEASE
ABORTind	ACSE	primitive d'indication A-ABORT ou primitive d'indication A-ABORT-P

Tableau 12-2/Q.813 – Etats de la machine SRPM

Nom abrégé	Nom et description
STA01	libre; non associé
STA02	associé

Tableau 12-3/Q.813 – Liste d'événements sortants

Nom abrégé	Cible	Nom et description
SRind	utilisateur SR	primitive d'indication SR-TRANSFER
P-DATAreq	fournisseur PS	primitive de demande P-DATA
ABORTreq	ACSE	primitive de demande A-ABORT

Tableau 12-4/Q.813 – Prédicats

Code	Nom et description
p1	unité APDU non acceptable, décision locale de rejet de l'unité APDU
p2	unité APDU non acceptable, décision locale d'abandon de l'association

Tableau 12-5/Q.813 – Table d'états de la machine SRPM

Evénements entrants	STA01	STA02
AA-ESTAB	STA02	
SRreq		P-DATAreq STA02
P-DATAind		SRind STA02
APDUua		p1: STA02 p2: ABORTReq STA01
AA-REL		STA01
ABORTind		STA01

12.1 Conventions

L'intersection d'un événement entrant (ligne) et d'un état (colonne) constitue une cellule.

Une cellule vide dans la table d'états représente une combinaison d'un événement entrant et d'un état qui n'est pas définie pour la machine SRPM.

Une cellule non vide représente une combinaison d'un événement entrant et d'un état qui est définie pour la machine SRPM. Une liste d'actions peut être obligatoire ou conditionnelle. Si une cellule contient une liste d'actions obligatoire, cette action est alors la seule figurant dans la cellule.

Une liste d'actions obligatoire possède le contenu suivant:

a) de manière optionnelle, un ou plusieurs événements sortants;

b) un état résultant.

Une liste d'actions conditionnelle possède le contenu suivant:

a) une expression de prédicats comprenant des prédicats et des opérateurs booléens (Ø représente l'opérateur booléen de négation);

b) une liste d'actions obligatoire (qui n'est utilisée que si l'expression de prédicats est vraie).

12.2 Actions effectuées par la machine SRPM

La table d'états de la machine SRPM définit les actions devant être effectuées par la machine SRPM sous la forme d'un événement sortant optionnel et d'un état résultant de l'association d'application.

12.2.1 Intersections non valides

Une cellule vide indique une intersection non valide d'un événement entrant et d'un état. L'action suivante est effectuée si une telle intersection est rencontrée:

a) si l'événement entrant provient de l'utilisateur SR, l'action effectuée par la machine SRPM est alors une affaire locale;

b) si l'événement entrant est lié à une unité APDU reçue, un fournisseur PS ou un élément ACSE, la machine SRPM émet alors une primitive de demande d'abandon à destination de l'élément ACSE.

12.2.2 Intersections valides

Une des actions suivantes est effectuée si l'intersection d'un état et d'un événement entrant est valide:

a) si la cellule contient une liste d'actions obligatoire, la machine SRPM effectue alors l'action spécifiée;

b) si la cellule contient une ou plusieurs listes d'actions conditionnelles, la machine SRPM effectue alors l'action spécifiée pour chaque expression de prédicats qui est vraie. La machine SRPM effectue l'une des actions spécifiées au 12.2.1 si aucune des expressions de prédicat n'est vraie.

13 Tables d'états de la machine de protocole d'opérations distantes

Le présent paragraphe est une extension de l'Annexe A "tables d'états ROPM" de la Recommandation X.229. Le présent paragraphe fournit la table d'états de la partie de transfert de la machine de protocole d'opérations distantes (ROPM-TR), dans le cas où l'élément STASE-ROSE figure dans le contexte d'application et où l'élément RTSE n'y figure pas.

Le présent paragraphe importe les définitions, les conventions et les états définis dans la Recommandation X.229 (prière de se référer à la Recommandation X.229 pour le contenu des informations). Le présent paragraphe contient les tableaux suivants:

a) le Tableau 13-1 indique l'événement entrant reçu du fournisseur du service STASE-ROSE (fournisseur SR) par l'élément ROSE en plus de ceux spécifiés dans le Tableau A.1/X.229;

b) le Tableau 13-2 indique les événements sortants en plus de ceux spécifiés dans le Tableau A.4/X.229;

c) le Tableau 13-3 spécifie la table d'états ROPM-TR, si l'élément STASE-ROSE figure dans le contexte d'application et que l'élément RTSE n'y figure pas.

Tableau 13-1/Q.813 – Liste d'événements entrants

Nom abrégé	Source	Nom et description
SR-TransInd	fournisseur SR	primitive d'indication SR-TRANSFER

Tableau 13-2/Q.813 – Liste d'événements sortants

Nom abrégé	Cible	Nom et description
SR-TransReq	fournisseur SR	primitive de demande SR-TRANSFER

Tableau 13-3/Q.813 – Table d'états ROPM-TR pour le transfert par l'élément STASE-ROSE

Evénements entrants	STA100	STA200
AA-ESTAB	STA200	
TRANSreq		SR-TransReq STA200
SR-TransInd		TRANSind STA200
AA-REL		STA100
AA-ABreq		ABORTreq STA100
ABORTind		AA-ABind STA100

ANNEXE A

Elément CMISE sécurisé

La présente annexe décrit l'utilisation de l'élément STASE-ROSE pour l'implémentation d'applications de réseau de gestion de télécommunications sécurisées. La Figure 2 donne le modèle d'un contexte d'application impliquant les éléments ACSE, CMISE, ROSE et STASE-ROSE. La présente annexe définit le contexte d'application, les règles d'établissement de l'association et la conformité pour l'élément CMISE sécurisé.

A.1 Contexte d'application

Le contexte d'application fourni ici est repris à partir du paragraphe 9.

Le nom de contexte d'application dont les entités d'application se composent des éléments SMASE, CMISE, ROSE, STASE et ACSE recevra l'attribution suivante de valeur d'identificateur d'objet:

{itu-t recommendation q8xx(8xx) stase(1) stase-application-context (2) secureTMNContext(1)}

et la valeur suivante de descripteur d'objet "Contexte sécurisé d'application interactive du RGT".

A.2 Règles d'établissement d'association

La Recommandation X.710 définit les paramètres d'établissement d'une association pour l'élément CMISE. La présente Recommandation exige en outre l'échange des paramètres

d'association définis au 7.4.2 si une négociation de paramètres est souhaitée au moment de l'établissement de l'association.

Comme spécifié dans le paragraphe 8, l'élément STASE-ROSE fournit à l'élément ACSE toute valeur proposée pour tout ou partie des paramètres de chiffrement. Le mécanisme utilisé par l'élément STASE-ROSE pour informer l'élément ACSE est une affaire d'implémentation qui n'est pas traitée par la présente Recommandation. Ces informations seront véhiculées dans le champ "informations utilisateur" de l'élément ACSE en utilisant le paramètre "sélection de paramètres de chiffrement" défini dans le paragraphe 5. L'élément CMISE peut également fournir durant la même phase à l'élément ACSE des informations pertinentes concernant l'élément CMISE homologue. Toutes ces informations sont véhiculées dans le champ "informations utilisateur" de l'élément ACSE. Le champ "informations utilisateur" de l'élément ACSE se constitue d'une expression SEQUENCE OF EXTERNAL. La présente Recommandation spécifie l'ordre suivant des expressions EXTERNAL figurant dans le champ "informations utilisateur" de l'élément ACSE: données fournies éventuellement pour l'élément STASE-ROSE, données fournies éventuellement pour l'élément CMISE, données fournies éventuellement pour l'élément SMASE.

A.3 Conformité

Un système déclarant la conformité à l'élément CMISE sécurisé se conformera aux prescriptions suivantes.

A.3.1 Prescriptions statiques

a) le système se conformera à toutes les prescriptions définies au 8.1/X.711;

b) le système prendra en charge les règles de codage distinctives définies dans la Recommandation X.690;

c) le système prendra en charge le protocole STASE-ROSE défini dans le paragraphe 10.

A.3.2 Prescriptions statiques

a) le système se conformera à toutes les prescriptions définies au 8.2/X.711;

b) le système prendra en charge les procédures STASE-ROSE définies aux paragraphes 7 et 9.5.

ANNEXE B

Syntaxes ASN.1 définies dans la présente Recommandation

La présente annexe rassemble les diverses définitions de syntaxe ASN.1 fournies dans la présente Recommandation.

B.1 Syntaxe abstraite pour l'élément d'authentification par clé publique

Le module d'authentification qui suit doit être véhiculé dans le champ "valeur d'authentification" de l'unité fonctionnelle d'authentification de l'élément ACSE lorsqu'une authentification de l'entité homologue par clé publique est requise.

```
STASE-ROSE-Authentication-value {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0)
abstractSyntax(1) stase-authentication-value(0) }

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTE tout

IMPORTS

SenderId, ReceiverId, Signature, SignatureCertificate
FROM Secure-Remote-Operations-APDUs {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0) stase-
data(2)};

Authentication-value ::= CHOICE {
                                    explicit            [0] ExplicitAuthenticator,
                                    gssAuthenticator    [1] GssAuthenticator
    -- à n'utiliser que si les deux entités communicantes utilisent GSS-API.
            }

ExplicitAuthenticator ::= SEQUENCE {
                                    senderId    [0] SenderId,
                                    receiverId  [1] ReceiverId,
                                    time        [3] GeneralizedTime,
                                    encryptedSymmetricKey [4] INTEGER   OPTIONAL,
    -- clé de chiffrement symétrique chiffrée avec la clé publique du récepteur
                                   signature    [5] Signature,
    -- signature par l'émetteur des champs précédents codés sous forme de caractères ASCII
                                   certificate  [6] SignatureCertificate       OPTIONAL
    -- certificat de la clé publique de l'émetteur pour la clé utilisée pour la signature
    }

GssAuthenticator ::= SEQUENCE {
                    gssMechanism            [0] OBJECT IDENTIFIER   OPTIONAL,
                    gssInitialContextToken  [1] OCTET STRING
                    }

END
```

La présente Recommandation attribue la valeur d'identificateur d'objet ASN.1 suivante:

{itu-t recommendation q(17) q813(813) stase(1) abstractSyntax(1) stase-authentication-value(0)}

comme nom de syntaxe abstraite pour l'ensemble de toutes les valeurs de données de présentation, chacune de ces dernières étant une valeur de type ASN.1

STASE-ROSE-Authentication-value.Authentication-value.

La valeur de descripteur d'objet correspondante sera "STASE-ROSE-Authenticator".

B.2 Syntaxe abstraite pour la négociation de paramètres de sécurité

Le module suivant de négociation de paramètres de sécurité est enregistré à des fins d'utilisation dans le champ "informations utilisateur" de l'élément ACSE.

```
STASE-A-ASSOCIATE-Information
        {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0) stase-userinfo(1)}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTE tout

IMPORTS

SenderId, ReceiverId, Signature, KeyId , PublicKeyCertificate, EncryptionCertificate, SignatureCertificate,
EncryptedAuthenticatedSymmetricKey
FROM Secure-Remote-Operations-APDUs {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0)
stase-data(2)};

EncryptionParametersSelection ::= SET
    {symmetricKeyIds          [0] SET OF KeyId                          OPTIONAL,
     publicKeyIds             [1] SET OF KeyId                          OPTIONAL,
     sealKeyIds               [2] SET OF KeyId                          OPTIONAL,
     signatureKeyIds          [3] SET OF KeyId                          OPTIONAL,
     passwordIds              [4] SET OF KeyId                          OPTIONAL,
     initializationVector     [5] OCTET STRING (SIZE(8))                OPTIONAL,
     feedBackBits             [6] INTEGER (1..63)                       OPTIONAL,
-- pour le mode de feed back de sortie à k bits ou
-- le mode de feed back de chiffrement à k bits de l'algorithme DES
    symmetricAlgorithms       [7] SET OF OBJECT IDENTIFIER              OPTIONAL,
    publicKeyAlgorithms       [8] SET OF OBJECT IDENTIFIER              OPTIONAL,
    signatureAlgorithms       [9] SET OF OBJECT IDENTIFIER              OPTIONAL,
    sealAlgorithms            [10] SET OF OBJECT IDENTIFIER             OPTIONAL,
    hashAlgorithms            [11] SET OF OBJECT IDENTIFIER             OPTIONAL,
    keyDigest                 [12] OCTET STRING (SIZE(8..64))           OPTIONAL,
    -- pour la vérification de clés publiques
    blockSize                 [13] INTEGER                              OPTIONAL,
    -- pour un hachage carré modulo n
    keySizes                  [14] SET OF INTEGER                       OPTIONAL,
    -- pour l'algorithme RSA
    publicKeys                [15] SET OF SEQUENCE
                                       {modulus          INTEGER,
                                        exponent         INTEGER
                                       }                                 OPTIONAL,
    sequenceNumber            [16] INTEGER                              OPTIONAL,
    timeStamp                 [17] GeneralizedTime                      OPTIONAL,
    encryptedKey              [18] OCTET STRING (SIZE(64..128))         OPTIONAL,
    -- clé de session symétrique, chiffrée en utilisant la clé de chiffrement de clé
    encryptedSymmetricKey     [19] INTEGER                              OPTIONAL,
    -- clé de session symétrique, chiffrée en utilisant la clé publique du récepteur
    keyEncryptionKey          [20] SEQUENCE (SIZE (1..3)) OF KeyId      OPTIONAL,
    -- une à trois clés symétriques utilisées pour le chiffrement d'une clé de session
    keyListIds                [21] SET OF KeyListId                     OPTIONAL,
    -- liste de clés de chiffrement pouvant être utilisées pendant la durée de l'association
    encryptionCertificate     [22] SET OF EncryptionCertificate         OPTIONAL,
-- certificats X.509 ou itinéraires de certification des clés publiques de l'émetteur utilisés uniquement
-- pour le chiffrement
    signatureCertificate      [23] SET OF SignatureCertificate          OPTIONAL,
-- certificats X.509 ou itinéraires de certification des clés publiques de l'émetteur
-- utilisés uniquement pour des signature numériques
    encryptedAuthenticatedSymmetricKeys [24] SET OF
    EncryptedAuthenticatedSymmetricKey                                  OPTIONAL,
-- clé de session symétrique, chiffrée en utilisant la clé publique du récepteur et signée avec la clé de l'émetteur
    macAlgorithms             [25] SET OF OBJECT IDENTIFIER             OPTIONAL,
    publicKeyCertificate      [26] SET OF PublicKeyCertificate          OPTIONAL,
```

```
-- certificats X.509 ou itinéraires de certification des clés publiques de l'émetteur
-- sans restrictions d'utilisation
        ...
        }
-- La sélection de paramètres de chiffrement est utilisée de manière facultative pendant
-- l'établissement de l'association pour négocier les algorithmes et les autres paramètres de
-- chiffrement qui seront pris en charge pendant la durée de l'association. Cette sélection n'est pas
-- utilisée pour les unités PDU de l'élément STASE-ROSE.

KeyListId ::= CHOICE    {identifier    OBJECT IDENTIFIER,
                         name          GraphicString,
                         number        INTEGER
                        }

END
```

La présente Recommandation attribue la valeur d'identificateur d'objet ASN.1 suivante:

{itu-t recommendation q(17) q813(813) stase(1) abstractSyntax(1) stase-userinfo(1)}

comme nom de syntaxe abstraite pour l'ensemble de toutes les valeurs de données de présentation, chacune de ces dernières étant une valeur de type ASN.1.

STASE-A-ASSOCIATE-Information.EncryptionParametersSelection

La valeur de descripteur d'objet correspondante sera "STASE-ROSE-User-Information".

B.3 Définition de la syntaxe abstraite des unités APDU

Les types ASN.1 suivants sont définis pour l'élément STASE-ROSE en plus de ceux qui sont définis dans la Recommandation X.229.

```
Secure-Remote-Operations-APDUs {itu-t recommendation q(17) q813(813) stase(1) stase-pci(0) stase-data(2)}

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTE tout

IMPORTS

ROSEapdus
FROM Remote-Operations-ADPUs {joint-iso-ccitt remote-operations(4) apdus(1)}

AE-title
FROM ACSE-1 {joint-iso-ccitt association-control (2) abstract-syntax(1) apdus(0) version(1)}

DistinguishedName
FROM InformationFramework {joint-iso-ccitt ds(5) modules(1) informationFramework (1)}

-- le module cité et la syntaxe correspondante sont définis dans l'Annexe  D/X.711 – 1998.

Certificate, CertificationPath
FROM AuthenticationFramework {joint-iso-ccitt ds(5) modules(1) authenticationFramework(7)};

SR-APDU ::= CHOICE
        { clear                     [0] ROSEapdus,
          simpleConfidential        [1] OCTET STRING,
          confidential              [2] Enciphered ,
          simplePublicEnciphered    [3] SimplePublicEnciphered,
          publicEnciphered          [4] PublicEnciphered ,
          hashed                    [5] HashedROSEpdu ,
```

```
        sealed                  [6] SealedROSEpdu ,
        signed                  [7] SignedROSEpdu ,
        confidentialSigned      [8] ConfidentialSigned,
        confidentialMAC         [9] ConfidentialMAC,
        confidentialSealed      [10] ConfidentialSealed,
        gssToken                [11] GssToken,
        ...
       }
Enciphered ::= SEQUENCE
                {encrypted              OCTET STRING,
                 encryptionParameters   EncryptionParameters OPTIONAL
                }

-- encrypted représente une unité PDU ROSE codée selon les règles DER et chiffrée
-- encryptionParameters représente les paramètres de chiffrement

SimplePublicEnciphered ::= CHOICE
                           { integers   SEQUENCE OF INTEGER,
                             string     OCTET STRING
                           }

-- SimplePublicEnciphered représente une unité PDU ROSE codée selon les règles DER
-- et chiffrée avec la clé publique.
-- Une unité PDU longue peut être fragmentée en blocs plus petits dont chacun peut être chiffré
-- sous la forme d'un type INTEGER. La taille de ces blocs dépend de l'algorithme de chiffrement
-- par clé publique utilisé et de la taille de la clé publique. La taille de tels blocs est
-- en dehors du domaine d'application de la présente Recommandation.
-- Le résultat du chiffrement par clé publique peut être représenté dans certains cas par un type
-- OCTET STRING.

PublicEnciphered ::= SEQUENCE
                     {publicEncrypted   SimplePublicEnciphered,
                      encryptionParameters     EncryptionParameters OPTIONAL
                     }

-- publicEncrypted représente une unité PDU ROSE codée selon les règles DER
-- et chiffrée avec la clé publique.
-- encryptionParameters représente les paramètres utilisés pour le chiffrement.

Hash ::= SEQUENCE{
                 hashValue              OCTET STRING (SIZE(8..64)),
                 encryptionParameters   EncryptionParameters OPTIONAL
                }

-- hashValue représente le résumé de message obtenu par hachage de l'unité PDU ROSE
-- codée selon les règles DER.
-- encryptionParameters représente les paramètres utilisés par l'algorithme de hachage.

HashedROSEpdu ::= SEQUENCE
          {data         OCTET STRING,
           hash         CHOICE { hash           Hash,
                                 simpleHash     OCTET STRING (SIZE (8..64))
                               }
          }

-- data représente l'unité PDU ROSE codée selon les règles DER.
-- hash représente le résultat du hachage sous la forme d'une chaîne OCTET STRING simple
-- ou de la structure Hash décrite ci-dessus.
```

```
Seal ::= SEQUENCE
        {sealValue              OCTET STRING (SIZE(8..64)),
         encryptionParameters   EncryptionParameters OPTIONAL
        }

-- sealValue représente la valeur du scellé pour l'unité PDU ROSE codée selon les règles DER
-- encryptionParameters représente les paramètres utilisés par l'algorithme de génération du scellé

SealedROSEpdu ::= SEQUENCE
        {data           OCTET STRING,
         seal           CHOICE  {seal       Seal,
                                 simpleSeal OCTET STRING (SIZE(8..128))
                                  }
        }

-- data représente l'unité PDU ROSE codée selon les règles DER.
-- seal représente la valeur du scellé sous la forme d'une chaîne OCTET STRING simple
-- ou de la structure Seal décrite ci-dessus.

Signature ::= SEQUENCE
                {signatureValue         SEQUENCE (SIZE(1..4)) OF INTEGER,
                 encryptionParameters   EncryptionParameters OPTIONAL
                }

-- signatureValue représente la signature de l'unité PDU ROSE codée selon les règles DER.
-- encryptionParameters représente les paramètres de l'algorithme de signature.

SignedROSEpdu ::= SEQUENCE
    {data           OCTET STRING,
     signature      CHOICE {signature          [1] Signature,
                            simpleSignature   [2] SEQUENCE (SIZE(1..4)) OF INTEGER
                            }
    }

-- data contient le codage de l'unité PDU ROSE selon les règles DER.
-- signature représente la signature de l'unité PDU ROSE codée selon les règles DER,
-- sous la forme d'un type INTEGER simple ou de la structure Signature définie ci-dessus.

ConfidentialSigned ::= SEQUENCE
        { encrypted  OCTET STRING,
          signature  CHOICE {signature          [1] Signature,
                             simpleSignature    [2] SEQUENCE (SIZE(1..4)) OF INTEGER
                             }
    }

-- encrypted représente le chiffrement de l'unité PDU ROSE codée selon les règles DER.
-- signature représente la signature de l'unité PDU ROSE codée selon les règles DER sous une forme
-- simple ou sous la forme de la structure Signature définie ci-dessus.

ConfidentialMAC ::= SEQUENCE
    { encrypted  OCTET STRING,
      mac        CHOICE {mac          [1] Hash,
                         simpleMAC    [2] OCTET STRING (SIZE (8..64))
                         }
    }

-- encrypted représente le chiffrement de l'unité PDU ROSE codée selon les règles DER.
-- mac représente le code MAC de l'unité PDU ROSE codée selon les règles DER sous une
-- forme simple ou sous la forme de la structure Hash définie ci-dessus.
```

```
ConfidentialSealed ::= SEQUENCE
        { encrypted     OCTET STRING,
          seal          CHOICE {sealed          [1] Seal,
                                simpleSealed    [2] OCTET STRING (SIZE (8..64))
                                }
        }

-- encrypted représente le chiffrement de l'unité PDU ROSE codée selon les règles DER.
-- seal représente le scellé de l'unité PDU ROSE codée selon les règles DER sous une forme simple
-- ou sous la forme de la structure Seal ci-dessus.

EncryptionParameters ::= SET
        {symmetricKeyId         [0] KeyId                               OPTIONAL,
         publicKeyId            [1] KeyId                               OPTIONAL,
         sealKeyId              [2] KeyId                               OPTIONAL,
         signatureKeyId         [3] KeyId                               OPTIONAL,
         passwordId             [4] KeyId                               OPTIONAL,
         initializationVector   [5] OCTET STRING (SIZE(8))              OPTIONAL,
         feedBackBits           [6] INTEGER (1..63)                     OPTIONAL,
-- pour le mode de feed back de sortie à k bits ou
-- le mode de feed back de chiffrement à k bits de l'algorithme DES
        symmetricAlgorithm      [7] OBJECT IDENTIFIER                   OPTIONAL,
        publicKeyAlgorithm      [8] OBJECT IDENTIFIER                   OPTIONAL,
        signatureAlgorithm      [9] OBJECT IDENTIFIER                   OPTIONAL,
        sealAlgorithm           [10] OBJECT IDENTIFIER                  OPTIONAL,
        hashAlgorithm           [11] OBJECT IDENTIFIER                  OPTIONAL,
        keyDigest               [12] OCTET STRING (SIZE(8..64))         OPTIONAL,
-- pour la vérification des clés publiques
        blockSize               [13] INTEGER                            OPTIONAL,
-- pour le hachage carré modulo n
        keySize                 [14] INTEGER                            OPTIONAL,
-- pour l'algorithme RSA
        publicKey               [15] SEQUENCE
                                        {modulus        INTEGER,
                                         exponent       INTEGER
                                }                                       OPTIONAL,
        sequenceNumber          [16] INTEGER                            OPTIONAL,
        timeStamp               [17] GeneralizedTime                    OPTIONAL,
        encryptedKey            [18] OCTET STRING (SIZE(64..128))       OPTIONAL,
-- clé de session symétrique, chiffrée en utilisant la clé de chiffrement de clé
        encryptedSymmetricKey   [19] INTEGER                            OPTIONAL,
-- clé de session symétrique, chiffrée en utilisant la clé publique du récepteur
        keyEncryptionKey        [20] SEQUENCE (SIZE (1..3)) OF KeyId
                                                                        OPTIONAL,
-- une à trois clés symétriques utilisées pour le chiffrement d'une clé de session
        publicKeyCertificate    [21]    PublicKeyCertificate            OPTIONAL,
-- certificat X.509 ou itinéraire de certification de la clé publique de l'émetteur sans restrictions d'utilisation
        encryptionCertificate   [22] EncryptionCertificate      OPTIONAL,
-- certificat X.509 ou itinéraire de certification de la clé publique de l'émetteur utilisé uniquement pour le
-- chiffrement
        signatureCertificate    [23] SignatureCertificate               OPTIONAL,
-- certificat X.509 ou itinéraire de certification de la clé publique de l'émetteur utilisé uniquement pour les
-- signatures numériques
        encryptedAuthenticatedSymmetricKey
                                [24] EncryptedAuthenticatedSymmetricKey         OPTIONAL,
-- clé de session symétrique, chiffrée en utilisant la clé publique du récepteur et signée avec la clé privée
-- de l'émetteur
        macAlgorithm            [25] OBJECT IDENTIFIER                  OPTIONAL,

        ...
        }
```

```
-- le type extensible EncryptionParameters est utilisé comme conteneur pour tous les paramètres
-- pouvant être utilisés pour toutes les transformations de sécurité. La plupart des applications
-- n'utiliseront aucune ou une faible partie des composants du type EncryptionParameters

KeyId ::= CHOICE        {
                         name        GraphicString,
                         number      INTEGER
                        }

PublicKeyCertificate ::= CHOICE {certificate            [0] Certificate,
                                 certificationPath      [1] CertificationPath
                                 }

EncryptionCertificate ::= CHOICE {certificate           [0] Certificate,
                                  certificationPath     [1] CertificationPath
                                  }

SignatureCertificate ::= CHOICE {certificate            [0] Certificate,
                                 certificationPath      [1] CertificationPath
                                 }

EncryptedAuthenticatedSymmetricKey ::= SEQUENCE {
                                 encryptedSymmetricKey              INTEGER,
                    -- clé de session symétrique, chiffrée en utilisant la clé publique du récepteur
                                 time                   GeneralizedTime,
                                 sender                 SenderId,
                                 receiver               ReceiverId,
                                 signature              Signature
-- la signature est calculée à partir de la représentation ASCII des quatre champs précédents
-- en utilisant la clé privée de l'émetteur
                                 }
SenderId ::= CHOICE {
                         identifier     [1] DistinguishedName,
                         name           [2] GraphicString,
                         application    [3] AE-title                    }
ReceiverId ::= SenderId

GssToken::=CHOICE {
        micToken        [1]     MicToken ,
        wrapToken       [2]     OCTET STRING
        }

MicToken::=SEQUENCE {
                rosePDU     [1]     OCTET STRING ,
                token       [2]     OCTET STRING
        }

END
```

B.4 Identificateur d'objet de syntaxe abstraite

La présente Recommandation attribue l'identificateur suivant:

{itu-t recommendation q(17) q813(813) stase(1) abstractSyntax(1) stase-data(2)}

comme nom de syntaxe abstraite pour l'ensemble de valeurs de données de présentation, chacune des valeurs étant du type ASN.1

Secure-Remote-Operations-APDUs.SR-APDU

dans lequel les composants d'argument des unités PDU ROSE sont fournis par l'utilisateur de l'élément ROSE.

La valeur du descripteur d'objet correspondante sera "STASE-ROSE-Data".

B.5 Noms de contextes d'application

La valeur d'identificateur d'objet suivante sera attribuée au nom du contexte d'application dont l'entité d'application se constitue des éléments SMASE, CMISE, ROSE, STASE-ROSE et ACSE:

{itu-t recommendation q(17) q813(813) stase(1) stase-application-context(2) secureTMNContext(0)}

et la valeur du descripteur d'objet correspondante sera "Secure-TMN-Interactive-Application-Context".

La valeur d'identificateur d'objet suivante sera attribuée au nom du contexte d'application dont l'entité d'application se constitue des éléments ou entités X.500, ROSE, STASE-ROSE et ACSE:

{itu-t recommendation q(17) q813(813) stase(1) stase-application-context (2) secureDirectoryContext(1)}

et la valeur du descripteur d'objet correspondante sera 'Secure-Directory-Application-Context".

APPENDICE I

Temps uniformément croissant utilisé à des fins de sécurité

La présente Recommandation spécifie l'utilisation d'un temps uniformément croissant pour certains buts de sécurité. Le présent appendice décrit une réalisation possible pour un tel paramètre de temps. Cette description est donnée uniquement dans un but d'illustration et d'autres solutions peuvent être possibles.

L'horloge d'un système réel peut subir diverses dégradations:

- fluctuation de rythme, qui peut conduire à une avance ou à un retard par rapport au temps exact;
- arrêt de fonctionnement pendant une certaine durée;
- perte de la date et de l'heure courantes, auquel cas l'horloge revient à un instant bien déterminé dans le passé; cette perte de la date actuelle peut s'accompagner, ou non, d'un arrêt.

Le présent appendice décrit la construction d'une horloge, utilisable par les mécanismes de sécurité, qui assure un service interrompu même si l'une des dégradations décrites plus haut affecte l'horloge du système réel.

Nous distinguerons quatre types de temps dans le présent appendice:

1) Le temps GMT qui est le temps astronomique correct.
2) L'horloge système (SC, *system clock*) qui est le temps indiqué par l'horloge du système.
3) Le temps virtuel (VT, *virtual time*) qui est le seul temps utilisé par les mécanismes de sécurité (et le cas échéant par d'autres composants du système).
4) Le temps externe (ET, *external time*) qui est le temps apparaissant dans une unité PDU entrante.

Le temps virtuel est le temps qui est lu chaque fois qu'une unité PDU sortante est générée avec un horodatage et chaque fois qu'une unité PDU entrante contenant un horodatage est reçue (ainsi qu'à d'autres fins qui sont en dehors du domaine d'application de la présente Recommandation). Chaque

fois que le temps virtuel est accédé en lecture, sa valeur est d'abord mise à jour, après quoi la valeur mise à jour est utilisée dans la réponse à la demande de lecture. La valeur mise à jour est également stockée dans une mémoire non volatile. Le pseudo-code suivant définit la procédure de mise à jour du temps virtuel VT:

si

$$VT < SC$$

alors

$$VT = SC$$

sinon

$$VT = VT + 1 \text{ battement}$$

dans lequel 1 battement est la plus petite valeur possible de l'incrément de temps de l'horloge virtuelle; sa valeur doit être suffisamment petite pour que le rythme du battement virtuel soit supérieur au rythme de crête de lecture du temps virtuel. Une valeur usuelle du battement peut être de 10 ms.

Si l'horloge système s'arrête ou si elle est réinitialisée à sa valeur par défaut, le temps virtuel continuera alors à progresser à un rythme virtuel correspondant à sa fréquence d'utilisation. Le temps virtuel effectue un rattrapage du temps effectif lorsque l'horloge système est mise à jour au moyen du temps GMT.

L'objectif étant de prendre en compte des dérives importantes de l'horloge système (par exemple, de 30 minutes) entre des remises successives à l'heure GMT de l'horloge système, des systèmes peuvent accepter des unités PDU contenant un temps externe qui diffère du temps virtuel d'une valeur pouvant aller jusqu'au double de la dérive autorisée (par exemple, 1 heure). La valeur exacte de ce paramètre de tolérance peut être ajustée de manière à s'adapter aux caractéristiques des horloges des systèmes communicants. La détection de retard se fera évidemment avec une précision moindre en cas de déviation des horloges.

Des tolérances plus larges (par exemple, de 4 heures) peuvent être utilisées pour prendre en compte des défaillances catastrophiques de l'horloge système (arrêts prolongés ou perte du temps actuel, ou les deux). Une unité PDU qui est reçue avec une valeur de temps externe située entre les deux limites de tolérance sera acceptée, mais l'événement peut être journalisé dans une trace de vérification de sécurité. Si le temps externe d'une unité PDU est en dehors de la deuxième limite de tolérance, une alerte de sécurité peut alors être émise en plus de la journalisation; la décision de poursuivre, de libérer ou d'abandonner l'association est une affaire de politique de sécurité locale.

Toutes les journalisations de trace de vérification de sécurité peuvent être faites en utilisant le temps virtuel, ce qui garantit le maintien strict de l'ordre relatif des événements.

L'instant de chaque remise à l'heure GMT de l'horloge système peut être journalisé dans la trace de vérification de sécurité. Le journal peut contenir les valeurs de l'horloge système et du temps virtuel avant et après la remise à l'heure. Ces informations peuvent être utiles pour rapprocher le temps virtuel et le temps GMT lors d'une analyse de la trace de vérification de sécurité.

APPENDICE II

Exemple de négociation d'algorithmes de sécurité

La Figure II.1 présente un scénario possible pour la négociation d'algorithmes de sécurité[5].

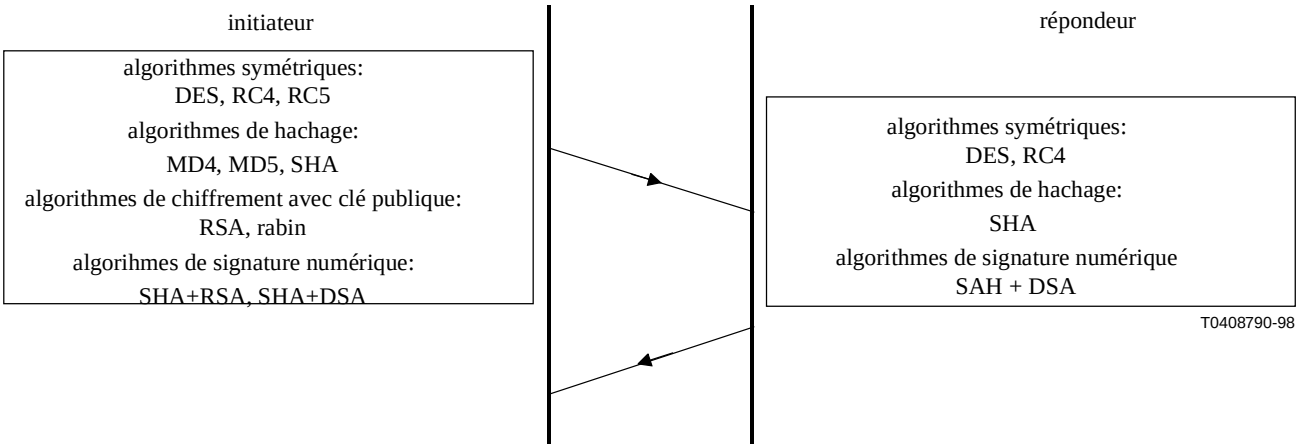

Figure II.1/Q.813 – Négociation d'algorithmes de sécurité

A la fin de l'échange illustré par la Figure II.1, l'initiateur peut décider que les ensembles d'algorithmes que le répondeur est prêt à prendre en charge pour l'association proposée ne sont pas acceptables, auquel cas il rejettera l'association. L'association se poursuivra si l'initiateur accepte les ensembles d'algorithmes proposés par le répondeur, avec les valeurs par défaut suivantes:

- algorithme symétrique: DES, étant donné qu'il s'agit de l'algorithme spécifié par défaut par la présente Recommandation et qu'il se trouve dans les options négociées pour les algorithmes symétriques;
- algorithme de hachage: SHA, étant donné qu'il s'agit du seul algorithme de hachage acceptable pour les deux parties;
- algorithme de chiffrement par clé publique: aucun algorithme de chiffrement par clé publique ne peut être utilisé durant cette association, car aucun n'a fait l'objet d'un accord;
- algorithme de signature numérique: SHA+DSA, étant donné qu'il s'agit du seul algorithme de signature numérique acceptable par les deux parties;
- algorithme de scellé numérique: MD5+DES, étant donné qu'il s'agit de la valeur par défaut spécifiée dans la présente Recommandation et que la négociation n'a pas porté sur des algorithmes de scellé numérique.

---

[5] RABIN (M.O.), Digital Signatures and Public Key Functions as Intractable as Factorization, *MIT Laboratory for Computer Science*, Technical Report, MIT/LCS/TR-212, janvier 1979.

RIVEST (R. L.), The MD4 Message Digest Algorithm, RFC 1320, avril 1992.

RIVEST (R. L.), The RC4 Encryption Algorithm, *RSA Data Security Inc.*, mars 1993.

RIVEST (R. L.), The RC5 Encryption Algorithm, *Dr. Dobb's Journal*, volume 20, No. 1, pp. 146-148, janvier 1995.

APPENDICE III

Utilisation de l'interface GSS-API avec l'élément STASE-ROSE

L'interface GSS-API est une interface de programmation d'application de haut niveau pour l'intégration de services de sécurité de communications. La dernière version (GSS-API version 2) est documentée dans la norme RFC 2078.

L'utilisation de l'interface GSS-API peut présenter un certain nombre d'avantages pour des fournisseurs de piles OSI qui souhaitent implémenter le service STASE-ROSE. En premier lieu, étant donné qu'il s'agit d'une interface de programmation d'application de haut niveau, l'interface GSS-API fournit aux réalisateurs d'applications un moyen très simple pour l'intégration de services de sécurité. L'utilisation de l'interface GSS-API pour le service STASE-ROSE peut garantir que des algorithmes de sécurité, ou même des mécanismes de sécurité complets, peuvent être modifiés sans avoir à modifier l'élément STASE-ROSE.

Le présent appendice décrit de quelle manière l'élément STASE-ROSE peut fournir les fonctionnalités de chiffrement (transformations de sécurité sur les unités PDU ROSE) au moyen de l'interface GSS-API (interface API pour les services génériques de sécurité). Le présent appendice décrit l'utilisation de l'interface GSS-API pendant les diverses phases de la communication.

III.1 Phase d'établissement de l'association

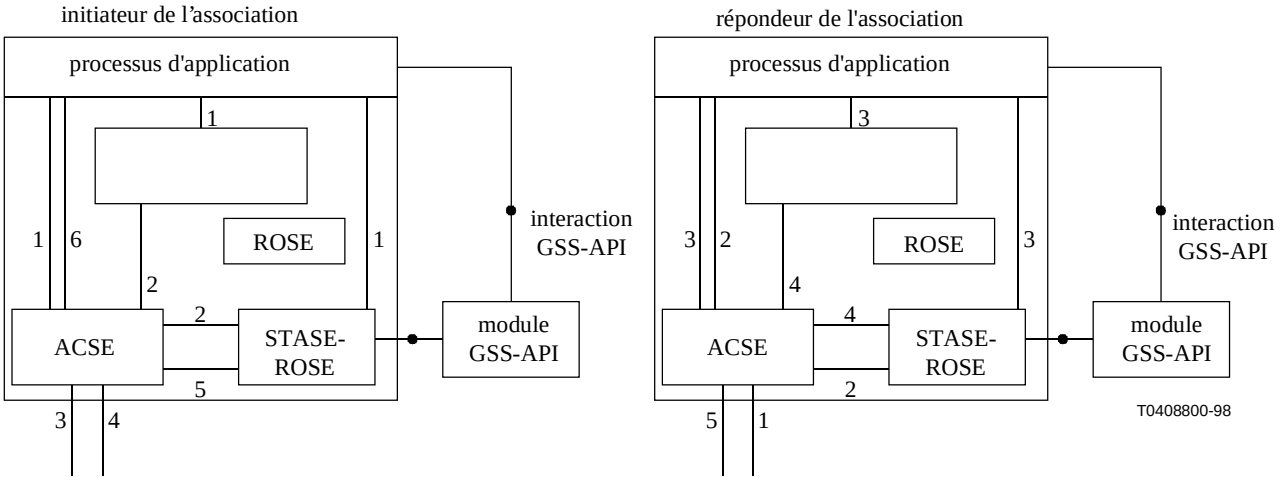

Figure III.1/Q.813 – Utilisation de l'interface GSS-API avec l'élément STASE-ROSE au moment de l'établissement de l'association

La Figure III.1 se base sur la Figure 4. Elle présente une utilisation possible de l'interface GSS-API pendant la phase d'établissement de l'association. Elle indique que le processus d'application et l'élément STASE-ROSE auront tous deux besoin d'accéder à la même interface GSS-API de prise en charge de module de chiffrement. Le processus d'application utilisera le module d'interface GSS-API à des fins d'authentification pendant l'établissement de l'association, alors que l'élément STASE-ROSE l'utilisera durant la phase de transfert de données. Les descriptions des interactions entre les différents composants indiqués dans la Figure III.1 sont décrites ci-dessous pour le côté initiateur (gauche) et le côté répondeur (droit).

Les interactions au niveau de l'initiateur de l'association sont les suivantes:

a) l'application invoque une primitive GSS_aquire_cred() pour obtenir ses données d'accréditation du module d'interface GSS-API;

b) l'application invoque une primitive GSS_init_sec_context() pour initialiser un contexte de sécurité avec un répondeur d'association spécifié. L'application doit faire le choix d'un contexte de sécurité pour l'association (c'est-à-dire, choisir entre une authentification unilatérale ou mutuelle et déterminer si une protection est nécessaire pour la succession des messages et leur reproduction). Le module d'interface GSS-API renverra à l'application un jeton "contexte initial";

c) l'application émettra la primitive de demande A-ASSOCIATE pour l'élément ACSE à destination du répondeur de l'application, ce qui fournit à cet élément ACSE le jeton "contexte initial" devant être véhiculé dans le champ "valeur d'authentification". La structure de jeton proposée au 2.1 doit dans ce cas être prise en charge par l'élément ACSE. L'application peut fournir simultanément à l'élément STASE-ROSE des informations concernant la protection des données durant la phase de transfert de données (le paramètre "descripteur opaque de contexte" de l'interface GSS-API doit être fourni au minimum comme référence de contexte de sécurité). Cette étape correspond à l'étape 1 décrite au 8.1.1;

d) les étapes suivantes sont identiques aux étapes 2 à 6 décrites au 8.1.1.

Les interactions au niveau du répondeur de l'association sont les suivantes:

a) les étapes 1 et 2 sont identiques aux deux premières étapes décrites au 8.1.2;

b) lorsque l'association reçoit une primitive d'indication A-ASSOCIATE, elle invoquera une primitive GSS_aquire_cred() pour obtenir les données d'accréditation en provenance du module d'interface GSS-API (si ce n'est déjà fait à cet instant). L'application invoquera ensuite une primitive GSS_accept_sec_context() avec, comme l'un des paramètres d'entrée, le jeton reçu de l'initiateur (dans de champ "valeur d'authentification de l'élément ACSE). Le module d'interface GSS-API authentifiera l'initiateur en vérifiant que le jeton est valide. Si une authentification mutuelle est demandée, l'association recevra alors du module d'interface GSS-API un deuxième jeton qui devra être retransmis vers l'initiateur.

Les étapes 3 à 5 sont identiques à celles décrites au 8.1.2. L'application fournira, comme partie de la réponse A-ASSOCIATE émise dans l'étape 3, un deuxième jeton (en cas d'authentification mutuelle) qui sera véhiculé dans le champ "valeur d'authentification". La syntaxe de structure de jeton proposée au 2.1 doit être prise en charge par le paramètre "valeur d'authentification" de la réponse ACSE A-ASSOCIATE de l'élément ACSE.

Négociation de contexte de sécurité

L'initiateur et les modules d'interface GSS-API cibles négocieront (de manière transparente pour l'élément STASE-ROSE), dans le cadre de l'échange initial de jetons au moment de l'établissement de l'association, un ensemble commun d'algorithmes d'intégrité et de confidentialité pour l'association de sécurité établie. L'ensemble valide d'algorithmes négocié sera toujours, par défaut, le plus grand ensemble commun d'algorithmes pris en charge par les deux parties. Cette négociation d'algorithmes est faite de manière automatique par les modules d'interface GSS-API qui communiquent, sans intervention de la part de l'utilisateur de l'interface GSS-API (application). Ce niveau de négociation est suffisant du point de vue de l'interfonctionnement, mais ne fournit toutefois pas la souplesse nécessaire pour que les applications qui communiquent puissent, par exemple, limiter le nombre des algorithmes valides.

Il est possible de mettre en œuvre une politique de sécurité plus souple en utilisant de manière optionnelle un deuxième mécanisme de négociation au niveau de l'élément STASE-ROSE

(c'est-à-dire, extérieur aux modules d'interface GSS-API) qui utilise les paramètres de négociation définis au 5.3. Les paramètres de négociation pertinents sont en particulier les suivants:

– algorithme symétrique;
– algorithme de clé publique;
– algorithmes de signature;
– algorithmes de scellé;
– algorithmes de hachage.

Cette fonctionnalité de négociation signifie que deux entités STASE-ROSE peuvent conclure un accord sur l'utilisation d'un ensemble d'algorithmes qui est un sous-ensemble de celui qui a été négocié par les modules d'interface GSS-API. Une condition préalable est, dans ce cas, que les entités STASE-ROSE sachent quels sont les algorithmes que leurs modules d'interface GSS-API locaux prennent en charge. Les principes de négociation sont les mêmes que pour l'entité STASE-ROSE dans le cas général.

III.2 Phase de transfert de données

La Figure III.2 se base sur la Figure 7. Elle présente l'utilisation de l'interface GSS-API pendant le transfert de données. Les interactions entre les différents composants du côté initiateur (gauche) et du côté répondeur (droit) seront les mêmes que celles décrites aux 8.4.1 et 8.4.2.

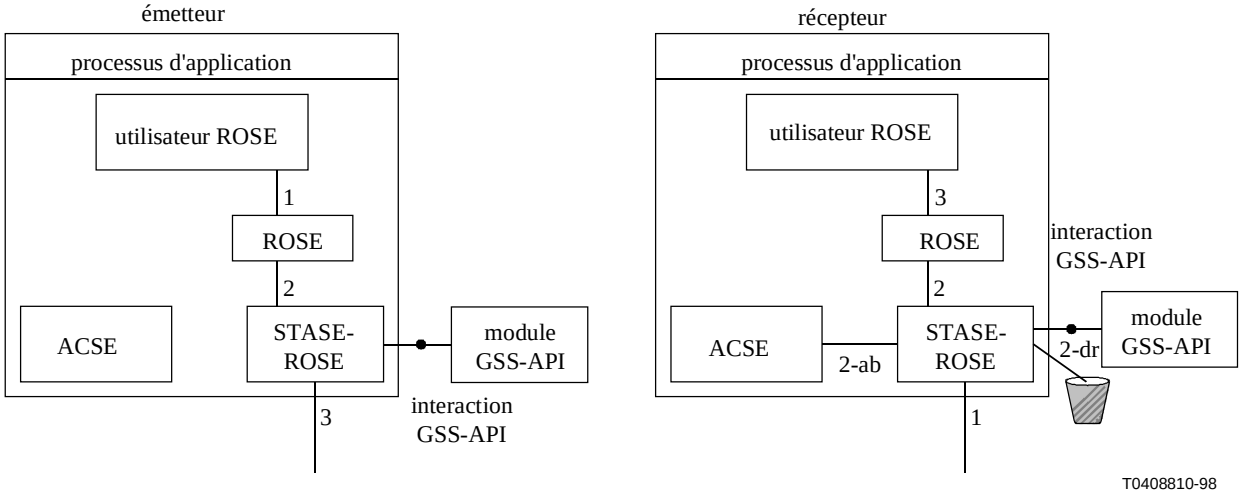

Figure III.2/Q.813 – Utilisation de l'interface GSS-API avec l'élément STASE-ROSE pendant le transfert de données

Durant cette phase, chaque entité STASE-ROSE interfacera son module d'interface GSS-API locale en utilisant les primitives GSS_wrap() et GSS_get_mic() de l'interface GSS-API pour la fourniture de transformations de sécurité. Chacune des applications peut fournir à ses entités ASE locales de l'entité STASE-ROSE le descripteur opaque de contexte de l'interface GSS-API qui est nécessaire comme paramètre local pour toutes les invocations des primitives GSS_wrap() et GSS_get_mic(). Les modules d'interface GSS-API qui communiquent et les entités ASE (si une négociation supplémentaire de contexte a été effectuée durant la phase d'établissement de l'association, voir 3.1.1) ont déjà négocié l'ensemble de variantes d'algorithmes pour l'association et, en conséquence, les possibilités pour la protection de la qualité (QoP, *quality of protection*).

L'élément ROSE demandera à l'élément STASE-ROSE un certain niveau de protection pour chaque message émis. La manière dont l'élément ROSE choisit le niveau de protection nécessaire n'est pas traitée ici (cette information est fournie de préférence par l'application locale).

L'élément ROSE doit indiquer au minimum à l'élément STASE-ROSE quel est, le cas échéant, le type de chiffrement à utiliser. L'élément STASE-ROSE reçoit ces informations dans le paramètre "type de chiffrement" de la primitive SR-TRANSFER. Ces demandes doivent être mappées vers une invocation de primitive gss_wrap()/gss_get_mic(). L'élément STASE-ROSE doit utiliser la primitive gss_wrap() lorsqu'une protection de la confidentialité est demandée, avec ou sans protection de l'intégrité. L'élément STASE-ROSE doit utiliser la primitive gss_get_mic() lorsqu'une protection de l'intégrité ou de non-répudiation est demandée sans protection de confidentialité.

Toutefois, la connaissance des conditions d'utilisation des primitives gss_wrap() et gss_get_mic() n'est pas suffisante pour l'élément STASE-ROSE. L'entité STASE-ROSE initiatrice doit également savoir quel est le niveau de qualité de protection (niveau QoP) qu'elle doit demander lorsqu'elle invoque ces fonctions [paramètre d'entrée qop_req des primitives gss_wrap() et gss_get_mic()]. L'élément STASE-ROSE peut en principe prendre la décision concernant la qualité de protection des deux manières suivantes:

1) l'élément STASE-ROSE est informé de la qualité de protection demandée par l'élément ROSE au moyen du paramètre "paramètres de chiffrement" de la primitive SR-TRANSFER (voir 7.4.4 de la spécification de l'élément STASE-ROSE);

2) si le paramètre "paramètres de chiffrement" de la primitive SR-TRANSFER n'est pas utilisé, le niveau de protection par défaut pour le type de chiffrement indiqué par la primitive SR-TRANSFER sera alors utilisé. On suppose dans ce cas que les entités STASE-ROSE ont déjà conclu un accord sur un niveau de protection par défaut au moment de l'établissement du contexte.

Quelle que soit la solution adoptée, il est important d'assurer que l'élément STASE-ROSE fasse le choix d'un niveau de qualité de protection qui reste dans les limites de la protection qui a été négociée au moment de l'établissement de l'association. Le réalisateur mettant en œuvre l'élément STASE-ROSE doit décider en conséquence de ce qu'il doit advenir si, par exemple, une primitive SR-TRANSFER pour l'élément ROSE demande un niveau de qualité de protection qui est supérieur à celui que le contexte existant est en mesure ou en droit de fournir.

Une fois qu'un jeton de fonction gss_wrap() ou gss_get_mic() a été créé, ce jeton sera inséré dans le protocole STASE-ROSE en utilisant le champ de protocole gssToken défini au 2.3. L'entité réceptrice invoquera les primitives gss_unwrap() or gss_verify_mic() pour vérifier ces jetons.

SERIES DES RECOMMANDATIONS UIT-T

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