UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T Q.813

(06/98)

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

SERIE Q: CONMUTACIÓN Y SEÑALIZACIÓN

Especificaciones del sistema de señalización N.° 7 – Interfaz Q3

Elemento de servicio de aplicación de transformaciones de seguridad para el elemento de servicio de operaciones a distancia (STASE-ROSE)

Recomendación UIT-T Q.813

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES DE LA SERIE Q DEL UIT-T

CONMUTACIÓN Y SEÑALIZACIÓN

SEÑALIZACIÓN EN EL SERVICIO MANUAL INTERNACIONAL	Q.1–Q.3
EXPLOTACIÓN INTERNACIONAL SEMIAUTOMÁTICA Y AUTOMÁTICA	Q.4–Q.59
FUNCIONES Y FLUJOS DE INFORMACIÓN PARA SERVICIOS DE LA RDSI	Q.60–Q.99
CLÁUSULAS APLICABLES A TODOS LOS SISTEMAS NORMALIZADOS DEL UIT-T	Q.100–Q.119
ESPECIFICACIONES DE LOS SISTEMAS DE SEÑALIZACIÓN N.° 4 Y N.° 5	Q.120–Q.249
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.° 6	Q.250–Q.309
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R1	Q.310–Q.399
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R2	Q.400–Q.499
CENTRALES DIGITALES	Q.500–Q.599
INTERFUNCIONAMIENTO DE LOS SISTEMAS DE SEÑALIZACIÓN	Q.600–Q.699
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.° 7	Q.700–Q.849
Generalidades	Q.700
Parte transferencia de mensajes	Q.701–Q.709
Parte control de la conexión de señalización	Q.711–Q.719
Parte usuario de telefonía	Q.720–Q.729
Servicios suplementarios de la RDSI	Q.730–Q.739
Parte usuario de datos	Q.740–Q.749
Gestión del sistema de señalización N.° 7	Q.750–Q.759
Parte usuario de la RDSI	Q.760–Q.769
Parte aplicación de capacidades de transacción	Q.770–Q.779
Especificaciones de las pruebas	Q.780–Q.799
Interfaz Q3	**Q.800–Q.849**
SISTEMA DE SEÑALIZACIÓN DIGITAL DE ABONADO N.° 1	Q.850–Q.999
Generalidades	Q.850–Q.919
Capa de enlace de datos	Q.920–Q.929
Capa de red	Q.930–Q.939
Gestión usuario-red	Q.940–Q.949
Descripción de la etapa 3 para los servicios suplementarios que utilizan el sistema de señalización digital de abonado DSS 1	Q.950–Q.999
RED MÓVIL TERRESTRE PÚBLICA	Q.1000–Q.1099
INTERFUNCIONAMIENTO CON SISTEMAS MÓVILES POR SATÉLITE	Q.1100–Q.1199
RED INTELIGENTE	Q.1200–Q.1999
RED DIGITAL DE SERVICIOS INTEGRADOS DE BANDA ANCHA (RDSI-BA)	Q.2000–Q.2999

Para más información, véase la Lista de Recomendaciones del UIT-T.

RECOMENDACIÓN UIT-T Q.813

ELEMENTO DE SERVICIO DE APLICACIÓN DE TRANSFORMACIONES DE SEGURIDAD PARA EL ELEMENTO DE SERVICIO DE OPERACIONES A DISTANCIA (STASE-ROSE)

Resumen

La presente Recomendación proporciona las especificaciones para soportar transformaciones de seguridad, como criptación, troceado, sellado y firma, centrando la atención en las unidades de datos de protocolo (PDU) del elemento de servicio de operaciones a distancia (ROSE) en su totalidad. Las transformaciones de seguridad se utilizan para facilitar la prestación de diversos servicios de seguridad, por ejemplo los de autenticación, confidencialidad, integridad y no repudio. Esta Recomendación describe una manera de realizar las transformaciones de seguridad que se implementa en la capa de aplicación y no requiere ninguna funcionalidad específica de la seguridad en ninguna de las capas de la pila OSI subyacentes.

Orígenes

La Recomendación UIT-T Q.813 ha sido preparada por la Comisión de Estudio 4 (1997-2000) del UIT-T y fue aprobada por el procedimiento de la Resolución N.° 1 de la CMNT el 26 de junio de 1998.

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución N.° 1 de la CMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

ÍNDICE

Página

Recomendación Q.813

ELEMENTO DE SERVICIO DE APLICACIÓN DE TRANSFORMACIONES DE SEGURIDAD PARA EL ELEMENTO DE SERVICIO DE OPERACIONES A DISTANCIA (STASE-ROSE)

(Ginebra, 1998)

1 Alcance, objetivo y aplicación

1.1 Alcance

Las transformaciones de seguridad (ST, *security transformation*) se utilizan para facilitar la prestación de diversos servicios de seguridad, tales como autenticación de la entidad par, autenticación del origen de los datos, confidencialidad, integridad y no repudio. Entre las transformaciones de seguridad figuran la criptación, el troceado, los sellos digitales y las firmas digitales.

Esta Recomendación se refiere a servicios de seguridad para las PDU de ROSE dentro de la capa de aplicación. Es independiente de la pila de protocolos de comunicaciones subyacente. Define un nuevo elemento de servicio de aplicación (ASE, *application service element*) llamado elemento de servicio de aplicación de transformaciones de seguridad para ROSE (STASE-ROSE, *security transformation application service element*), que reside entre un ROSE y la capa de presentación en la pila de protocolos OSI. La presente Recomendación indica una manera de efectuar las transformaciones de seguridad (ST) que no impone exigencia alguna a ninguna de las seis capas inferiores de la pila de comunicaciones, lo que contrasta con los métodos [por ejemplo, el de seguridad genérica de capa superior (GULS, *generic upper layers security*)] que soportan las transformaciones de seguridad mediante una funcionalidad insertada en la pila de comunicaciones en la capa de presentación.

La presente Recomendación prevé además la autenticación de la entidad en el momento en que se establece la asociación; la negociación de parámetros de seguridad (tales como los algoritmos de seguridad) que se utilizarán mientras dure la asociación; y la actualización dinámica, durante la asociación, de los parámetros de seguridad que se utilizan para cada una de las unidades de datos de protocolo.

El método presentado en esta Recomendación podría adaptarse para ASE que no son ROSE y que interactúan directamente con la capa de presentación. Sin embargo, esta Recomendación se centra en el ROSE y no considera ninguna posible extensión o generalización.

La manera según la cual de llevan a cabo en la práctica las transformaciones de seguridad (por ejemplo, produciendo y verificando firmas digitales) es un asunto local que queda fuera del alcance de la presente Recomendación. En particular es un asunto local la utilización de un módulo de seguridad genérico, tal como la interfaz de programación de aplicación de servicios de seguridad genéricos (GSS-API, *generic security service – application programming interface*), para efectuar las transformaciones de seguridad. No obstante, aunque esta Recomendación no impone la utilización de la GSS-API, si proporciona en cambio el marco necesario para utilizarla junto con el STASE-ROSE (véase apéndice III).

La gestión de claves es un componente importante de la infraestructura de seguridad. La presente Recomendación postula el intercambio de información relacionada con claves criptográficas. Sin embargo, el marco general de la gestión de claves queda fuera del alcance de esta Recomendación.

1.2 Objetivo

El objetivo de esta Recomendación es la protección de las PDU de ROSE en su totalidad.

La Recomendación Q.812 especifica la transferencia, el acceso y la gestión de ficheros (FTAM, *file transfer administration and management*), el elemento de servicio común de información de gestión (CMISE, *common information management application service element*) y el directorio X.500 en la capa de aplicación para las interfaces Q3 y X de la red de gestión de las telecomunicaciones (RGT). El directorio X.500 y el CMISE utilizan los servicios del elemento de servicio de operaciones a distancia (ROSE, *remote operation service element*). La presente Recomendación se refiere a la seguridad de las unidades de datos de protocolo (PDU, *protocol data units*) del ROSE. Si bien es la necesidad de asegurar las interacciones de la RGT o los intercambios de mensajes lo que motiva la presente Recomendación, también puede utilizarse ésta para proporcionar la seguridad de cualquier aplicación que utilice un ROSE.

1.3 Aplicación

Esta Recomendación es aplicable a las aplicaciones basadas en ROSE, por ejemplo las de usuario, que utilizan un CMISE o el directorio X.500. Uno de los objetivos principales de esta Recomendación es proporcionar protección a las PDU de CMIP (protocolo de información de gestión común). Puesto que el CMIP se basa en la versión de ROSE de 1988 (véanse las Recomendaciones X.219 y X.229), esta Recomendación también emplea esa versión, en vez de la de 1994 (véanse las Recomendaciones X.880, X.881 y X.882). Por consiguiente, esta Recomendación no es aplicable a la versión actual de la Recomendación X.500, que se basa en la versión de ROSE de 1994.

2 Referencias

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes.

2.1 Referencias normativas

– Recomendación UIT-T M.3010 (1996), *Principios para una red de gestión de las telecomunicaciones.*

– Recomendación UIT-T Q.811 (1997), *Perfiles de protocolo de capa inferior para las interfaces Q3 y X.*

– Recomendación UIT-T Q.812 (1997), *Perfiles de protocolo de capa superior para las interfaces Q3 y X.*

– Recomendación UIT-T X.200 (1994) | ISO/CEI 7498-1:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Modelo de referencia básico: El modelo básico.*

– Recomendación X.208 del CCITT (1988), *Especificación de la notación de sintaxis abstracta uno.*

– Recomendación UIT-T X.210 (1993) | ISO/CEI 10731:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Modelo de referencia básico: Convenios para la definición de servicios en la interconexión de sistemas abiertos.*

– Recomendación UIT-T X.217 (1995) | ISO/CEI 8649:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Definición de servicio para el elemento de servicio de control de asociación.*

– Recomendación X.219 del CCITT (1988), *Operaciones a distancia: Modelo, notación y definición del servicio.*

– Recomendación UIT-T X.227 (1995) | ISO/CEI 8650-1:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Protocolo con conexión para el elemento de servicio de control de asociación: Especificación de protocolo.*

– Recomendación X.229 del CCITT (1988), *Operaciones a distancia: Especificación del protocolo.*

– Recomendación UIT-T X.500 (1997) | ISO/CEI 9594-1:1997, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Visión de conjunto de conceptos, modelos y servicios.*

– Recomendación UIT-T X.509 (1997) | ISO/CEI 9594-8:1997, *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Marco de autenticación.*

– Recomendación UIT-T X.680 (1997) | ISO/CEI 8824-1:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de la notación básica.*

– Recomendación UIT-T X.681 (1997) | ISO/CEI 8824-2:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de objetos de información.*

– Recomendación UIT-T X.682 (1997) | ISO/CEI 8824-3:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Especificación de constricciones.*

– Recomendación UIT-T X.683 (1997) | ISO/CEI 8824-4:1998, *Tecnología de la información – Notación de sintaxis abstracta uno: Parametrización de las especificaciones de la notación de sintaxis abstracta uno.*

– Recomendación UIT-T X.690 (1997) | ISO/CEI 8825-1:1998, *Tecnología de la información – Reglas de codificación de notación se sintaxis abstracta uno: Especificación de las reglas de codificación básicas, de las reglas de codificación canónica y de las reglas de codificación distinguida.*

– Recomendación UIT-T X.710 (1997) | ISO/CEI 9595:1998, *Tecnología de la información – Interconexión de sistemas abiertos – Servicio común de información de gestión.*

– Recomendación UIT-T X.711 (1997) | ISO/CEI 9596-1:1998, *Tecnología de la información – Interconexión de sistemas abiertos – Protocolo común de información de gestión: Especificación.*

– ISO/CEI 9979:1991, *Data cryptographic techniques – Procedures for the registration of cryptographic algorithms.*

2.2 Referencias informativas

– Recomendación UIT-T X.803 (1994) | ISO/CEI 10745:1995, *Tecnología de la información – Interconexión de sistemas abiertos – Modelo de seguridad de capas superiores.*

– Recomendación UIT-T X.830 (1995) | ISO/CEI 11586-1:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Seguridad genérica de capas más altas – Sinopsis, modelo y notación.*

– Recomendación UIT-T X.831 (1995) | ISO/CEI 11586-2:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Seguridad genérica de las capas superiores: Definición de servicio del elemento de servicio de intercambio de seguridad.*

– Recomendación UIT-T X.832 (1995) | ISO/CEI 11586-3:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Seguridad genérica de las capas superiores: Especificación del protocolo de elemento de servicio de intercambio de seguridad.*

– Recomendación UIT-T X.833 (1995) | ISO/CEI 11586-4:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Seguridad genérica de las capas superiores: Especificación de la sintaxis de transferencia de protección.*

– ISO/CEI 9798-3:1993, *Information technology – Security techniques – Entity authentication mechanisms – Part 3: Entity authentication using a public key algorithm.*

– ISO/CEI 11770-1 (1996), *Information technology – Security techniques – Key management – Part 1: Framework.*

– 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, Jan. 1988.

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

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

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

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

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

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

3 Definiciones

En esta Recomendación se definen los términos siguientes.

3.1 entidad de aplicación de iniciación de asociación, iniciador de asociación: Entidad de aplicación que inicia la asociación de aplicación.

3.2 entidad de aplicación de respuesta de asociación: **respondedor de asociación**: Entidad de aplicación que responde a la iniciación de una asociación de aplicación por otra entidad de aplicación.

3.3 entidad de aplicación de emisión; emisor: Entidad de aplicación que emite la APDU hacia la entidad de aplicación de recepción.

3.4 entidad de aplicación de recepción; receptor: Entidad de aplicación que recibe la APDU procedente de la entidad de aplicación de emisión.

3.5 solicitante: Entidad de aplicación que emite una primitiva de transferencia STASE-ROSE.

3.6 aceptante: Entidad de aplicación que recibe una primitiva de indicación.

3.7 STASE-ROSE: Elemento de aplicación residente entre la capa de presentación OSI y un ROSE que proporciona las transformaciones necesarias para asegurar la transferencia de las PDU de ROSE.

3.8 transferencia segura: Mecanismo que proporciona la transferencia de unidades de datos de protocolo de aplicación (APDU) entre sistemas abiertos de manera segura.

3.9 usuario STASE-ROSE; usuario SR (SR-user): Elemento de servicio de aplicación que utiliza los servicios de un STASE-ROSE. El elemento de servicio de operaciones a distancia (ROSE) es el único usuario de los servicios STASE-ROSE.

3.10 proveedor STASE-ROSE; proveedor SR (SR-provider): Proveedor del elemento de servicio de aplicación de transformaciones de seguridad para ROSE.

3.11 proveedor ACSE: Proveedor del elemento de servicio de control de asociación.

Esta Recomendación utiliza las definiciones de servicios de seguridad y los mecanismos de seguridad especificados en las Recomendaciones X.800 y M.3016.

4 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

AARE	Respuesta de asociación ACSE (*ACSE association response*)
AARQ	Petición de asociación ACSE (*ACSE association request*)
ACSE	Elemento de servicio de control de asociación (*association control service element*)
AE	Entidad de aplicación (*application entity*)
APDU	Unidad de datos de protocolo de aplicación (*application protocol data unit*)
API	Interfaz de programación de aplicación (*application programming interface*)
ASCII	Código ASCII (*american standard code for information interchange*)
ASN.1	Notación de sintaxis abstracta uno (*abstract syntax notation one*)
BER	Reglas de codificación básica (*basic encoding rules*)
CBC	Concatenación de bloque cifrados (*cipher block chaining*)

CMIP	Protocolo de información de gestión común (*common management information protocol*)
CMISE	Elemento de servicio común de información de gestión (*common management information service element*)
DER	Reglas de codificación distinguida (*distinguished encoding rules*)
DES	Norma de criptación digital (*digital encryption standard*)
EBCDIC	Código EBCDIC (*extended binary coded decimal interchange code*)
FIPS PUB	Publicación de normas federales de tratamiento de la información (*federal information processing standards publication*)
FTAM	Transferencia, acceso y gestión de ficheros (*file transfer administration and management*)
GSS-API	Interfaz de programación de aplicación de servicios de seguridad genéricos (*generic security service – application programming interface*)
GULS	Seguridad genérica de capa superior (*generic upper layers security*)
CEI	Comisión Electrónica Internacional
IETF	Grupo de Tareas Especiales de ingeniería Internet (*Internet engineering task force*)
IS	Norma internacional (*international standard*)
ISO	Organización Internacional de Normalización (*international organization for standardization*)
MAC	Código de autenticación de mensaje (*message authentication code*)
MD	Compendio de mensajes (*message digest*)
NBS	Oficina Nacional de Normas (*national bureau of standards*)
NIST	Instituto Nacional de Normas y Tecnología (*national institute for standards and technology*)
PDU	Unidad de datos de protocolo (*protocol data unit*)
PKCS	Criptosistema de claves públicas (*public key cryptography standard*)
QoP	Calidad de protección (*quality of protection*)
RFC	Petición de comentarios (*request for comments*)
ROSE	Elemento de servicio de operaciones a distancia (*remote operations service element*)
RSA	Rivest Shamir Adelman
SR	STASE-ROSE
ST	Transformaciones de seguridad (*security transformation*)
STASE-ROSE	Elemento de servicio de aplicación de transformaciones de seguridad (STASE, *security transformation application service element*) para elemento de servicio de operaciones a distancia (ROSE, *remote operations service element*)
UIT-T	Unión Internacional de Telecomunicaciones – Sector de Normalización de las Telecomunicaciones

5 Visión de conjunto

En esta cláusula se da una visión de conjunto de alto nivel del STASE-ROSE. La atención se centra en lo que hace el STASE-ROSE, dejando para subcláusulas posteriores la exposición más pormenorizada de cómo lo hace.

El STASE-ROSE es un elemento de servicio de aplicación residente entre la capa de presentación OSI y un ROSE que proporciona las transformaciones necesarias para transferir de manera segura las PDU del ROSE. Además, el STASE-ROSE representa una manera de intercambiar información sobre la seguridad que se proporciona. El STASE-ROSE es invocado por una petición del ROSE en el lado de transmisión, y da una indicación al ROSE en el lado de recepción. Tanto la petición como la indicación contienen la PDU de ROSE que se protege, así como (opcionalmente), información relativa al tipo de seguridad que se da.

A continuación se examinan aspectos relativos a las ST y al intercambio de información de seguridad del STASE-ROSE.

5.1 Transformaciones de seguridad

El STASE-ROSE protege las PDU del ROSE aplicando las transformaciones de seguridad (ST) seleccionadas a las PDU del ROSE en su totalidad codificadas con las reglas de codificación distinguida (DER, *distinguished encoding rules*). El STASE-ROSE admite en particular las siguientes ST:

- **confidencial**: La PDU del ROSE con codificación DER es criptada a efectos de protección de la privacidad con un algoritmo de criptación de claves simétricas;
- **cifrado público**: La PDU del ROSE con codificación DER es criptada a efectos de protección de la privacidad con un algoritmo de criptación de claves públicas;
- **troceado**: El STASE-ROSE calcula un código de autenticación de mensajes (MAC, *message authentication code*) basado en la función troceado de la PDU del ROSE con codificación DER y una contraseña secreta y adjunta el resultado a la PDU del ROSE para la protección de la integridad;
- **sellado**: El STASE-ROSE calcula el sello digital de la PDU del ROSE con codificación DER y adjunta el resultado a la PDU del ROSE para la protección de la integridad;
- **firmado**: El STASE-ROSE calcula la firma digital de la PDU del ROSE con codificación DER y adjunta el resultado a la PDU del ROSE para la protección del no repudio;
- **firmado confidencial**: STASE-ROSE calcula la firma digital de la PDU de ROSE con codificación DER y adjunta al resultado PDU ROSE criptada (véase "confidencial") a efectos de protección del no repudio y la privacidad;
- **troceado confidencial**: El STASE-ROSE calcula el MAC de la PDU del ROSE con codificación DER y adjunta el resultado a la PDU del ROSE criptada (véase "confidencial") para la protección de la integridad y la privacidad;
- **sellado confidencial**: El STASE-ROSE calcula el sello digital de la PDU del ROSE con codificación DER y adjunta el resultado a la PDU del ROSE criptada (véase "confidencial") para la protección de la integridad y la privacidad.

El STASE-ROSE puede también pasar a través de las PDU del ROSE en **claro**, sin ninguna codificación ni ninguna ST.

5.2 Intercambio de información de seguridad

Los mensajes siguientes, intercambiados entre entidades STASE-ROSE o entre ROSE y STASE-ROSE, especifican cuál de las ST indicadas más arriba se utiliza para proteger la PDU del ROSE:

- el ROSE invoca STASE-ROSE en el lado de origen;
- el STASE-ROSE de origen envía una PDU de STASE-ROSE a una entidad STASE-ROSE par en el lado de recepción;
- el STASE-ROSE del lado de recepción proporciona una indicación al ROSE.

Es obligatorio saber cuáles son las ST que se efectúan, pero no es suficiente para que las comunicaciones sean satisfactorias. De hecho, ambos lados necesitan saber qué algoritmos se utilizan, así como los valores de los parámetros (por ejemplo, las claves de criptación, los vectores de inicialización) utilizados. Esta Recomendación proporciona varios valores y mecanismos por defecto que sólo requieren un mínimo intercambio de información relacionada con la seguridad. También proporciona los medios y sistemas con los que negociar en el momento en que se establece la asociación qué algoritmos serán admitidos y con los que cambiar y especificar esa información dinámicamente para cada PDU de un ROSE.

5.2.1 Valores por defecto de la información de seguridad

La utilización de la capacidad de negociación del STASE-ROSE es opcional. Si dos entidades comunicantes no utilizan dicha capacidad, deben establecer un acuerdo sobre un conjunto de parámetros de seguridad, tales como los algoritmos de seguridad, que se utilizarán mientras dure la asociación. Ambas partes se pueden poner de acuerdo sobre una serie de valores para esos parámetros por medios que quedan fuera del alcance de la presente Recomendación.

A menos que las entidades comunicantes acuerden otra cosa, se utilizarán los siguientes convenios, algoritmos de seguridad y mecanismos de seguridad:

– El algoritmo de criptación por defecto para la criptación de claves simétricas será la norma de criptación digital (DES, *digital encryption standard*) en el modo concatenación de bloques cifrados (CBC, *cipher block chaining*).

– Si se necesita una triple DES, el procedimiento por defecto será criptación, descriptación, criptación (EDE, *encryption decryption encryption*) en el modo retroalimentación externa de CBC con tres claves DES diferentes.

– Si no se especifica vector de inicialización (IV, *initialization vector*), el primer IV de la asociación constará de 64 bits de valor 0 y cada IV subsiguiente estará formado por los últimos 8 bytes de la PDU de ROSE criptada con anterioridad.

– El algoritmo de criptación de claves públicas por defecto será RSA[1, 2].

– El algoritmo de la función troceado por defecto será MD5[3].

– El MAC por defecto (para función troceado con clave) será HMAC[4].

---

1 RIVEST (R.), SHAMIR (A.), y ADELMAN (L. M.), "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, versión 21, N.º 2, págs. 120-126. febrero de 1978.

2 RSA Data Security Inc., PKCS N.º 1: RSA Encryption Standard, Versión 1.5, noviembre de 1993.

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

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

– El sello por defecto será el troceado MD5 de la PDU del ROSE con codificación DER criptada con DES.

– La firma digital por defecto será el troceado MD5 de la PDU del ROSE con codificación DER criptada con RSA y la clave privada del usuario.

– La autenticación de la entidad par deberá producirse en el momento en que se establece la asociación. Se utilizará la unidad funcional (FU, *functional unit*) autenticación opcional del ACSE. La información de autenticación deberá figurar en los campos valor de autenticación del llamante (calling-authentication-value) y valor de autenticación del respondedor (responding-authentication-value) de la FU autenticación de las PDU de la AARQ y la AARE respectivamente. Las cadena de bits de los campos requisitos ACSE del emisor (sender-acse-requirements) y requisitos ACSE del respondedor (responder-acse-requirements) de la FU autenticación deberán fijarse de modo que incluyan la FU autenticación. Los campos valor de autenticación del llamante y valor de autenticación del respondedor son del tipo valor de autenticación (Authentication-value) que en ISO 8650 se define como una OPCIÓN. La OPCIÓN del valor de autenticación deberá ser EXTERNO. El contexto de presentación deberá incluir una referencia a la sintaxis abstracta que se utiliza para el valor EXTERNO. Cuando se emplean valores y convenios por defecto no es necesario utilizar el campo (opcional) nombre del mecanismo (mechanism-name) de la FU autenticación de las PDU del ACSE.

– Si se necesita la autenticación de la entidad par con criptación de claves públicas, constará de lo siguiente:

  • identificador único del emisor;
  • identificador único del receptor;
  • una indicación de tiempo que sea la hora generalizada;
  • opcionalmente, una clave de criptación simétrica, que será utilizada por el emisor durante la asociación, encriptada con la clave pública del receptor;
  • una firma digital de los campos precedentes, firmados con la clave privada del emisor;
  • opcionalmente, el certificado de clave pública del emisor.

A menos que las entidades comunicantes acuerden otra cosa, por medios que quedan fuera del alcance de la presente Recomendación, la firma digital se calculará utilizando MD5 para el troceado y RSA para la criptación de claves públicas. La sintaxis de este autenticador figura en 5.2.1.1. Las claves (opcionales) de criptación simétrica criptada públicamente en los mensajes AARQ y AARE pueden ser diferentes, con lo que el iniciador y el respondedor de asociación pueden utilizar claves diferentes mientras dura la asociación.

Los autenticadores propuestos en esta Recomendación, así como otras partes de la misma, utilizan indicaciones de tiempo. Los relojes de sistema se pueden retrasar si van demasiado rápido, o el retraso puede ser provocado por algún funcionamiento defectuoso. La presente Recomendación exige que, con independencia de tales sucesos, las indicaciones de tiempo consecutivas producidas por un sistema aumenten de manera monótona. El apéndice I ilustra una posible construcción de tiempo monótonamente creciente.

Esta Recomendación no especifica cuál de los dos autenticadores se ha de utilizar. Las entidades comunicantes pueden ponerse de acuerdo, por medios que quedan fuera del alcance de la presente Recomendación, sobre el autenticador que utilizarán.

Si bien esta Recomendación define dos autenticadores, las partes comunicantes pueden concertar un acuerdo para utilizar otro autenticador. En ese caso, se especificará la sintaxis ASN.1 de ese

autenticador; también es necesario asignar y registrar un nombre de sintaxis abstracta para el autenticador, que se utilizará en la negociación de capa de presentación.

Las entidades comunicantes pueden ponerse de acuerdo, por medios que quedan fuera del alcance de la presente Recomendación, sobre un conjunto diferente de valores por defecto.

5.2.1.1 Sintaxis abstracta para el autenticador de claves públicas

El módulo siguiente, para la autenticación de claves públicas, se ha de llevar en el campo valor de autenticación (Authentication-value) de la FU autenticación de un ACSE cuando se requiera la autenticación de la entidad par con criptación de claves públicas. El módulo admite dos situaciones diferentes:

– todos los campos del autenticador se especifican explícitamente;

– ambas entidades comunicantes utilizan GSS-API y la información de autenticación se transporta como una cadena de octetos que es interpretada localmente por 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
-- EXPORTS everything

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,
        -- to be used only if the two communicating entities use GSS-API
                ...
                 }

ExplicitAuthenticator ::= SEQUENCE {
                                senderId        [0] SenderId,
                                receiverId      [1] ReceiverId,
                                time            [3] GeneralizedTime,
                                encryptedSymmetricKey [4] INTEGER    OPTIONAL,
        -- a symmetric encryption key encrypted with the receiver's public key
                                signature       [5] Signature,
        -- the sender's signature of the preceding fields encoded as ASCII characters
                                certificate     [6] SignatureCertificate        OPTIONAL
        -- the sender's public key certificate for the key used for the signature
        }

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

END
```

5.2.1.2 Nombre de sintaxis abstracta

Esta Recomendación asigna el valor de identificador de objeto ASN.1:

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

como un nombre de sintaxis abstracta para el conjunto de todos los valores de datos de presentación, cada uno de los cuales es un valor de tipo ASN.1.

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

El valor de descriptor de objeto correspondiente es "Autenticador STASE-ROSE" ("STASE-ROSE-Authenticator").

5.2.2 Negociación de algoritmos de seguridad

Los valores por defecto, ya sean especificados en esta Recomendación o acordados por las entidades comunicantes, pueden ser sustituidos dinámicamente como se describe más abajo.

En el momento en que se establece la asociación, las entidades comunicantes pueden negociar los algoritmos que van a utilizar mientras dura la asociación. En el campo información de usuario de la petición de asociación ACSE, el originador de la asociación puede, opcionalmente, incluir:

- un conjunto de algoritmos de criptación simétrica aceptables;
- un conjunto de algoritmos de criptación pública aceptables;
- un conjunto de algoritmos de troceado aceptables;
- un conjunto de algoritmos de troceado con clave (MAC) aceptables;
- un conjunto de algoritmos de sellos digitales aceptables;
- un conjunto de algoritmos de firmas digitales aceptables.

La AARE contendrá conjuntos de algoritmos aceptables que serán subconjuntos de los de la AARQ. Si esta capacidad de negociación opcional es utilizada por el originador de la asociación en la AARQ, debe ser utilizada por el respondedor de la asociación. Una respuesta de asociación que no utiliza la capacidad de negociación opcional equivale a una respuesta con valores nulos (es decir, conjuntos vacíos). Si en la AARQ falta algunos de los conjuntos indicados más arriba, sólo se admitirá el valor por defecto del algoritmo correspondiente. Si en la AARQ está presente alguno de los conjuntos indicados más arriba, deberá estar presente en la AARE; de no ser así, la ausencia de ese conjunto de la AARE equivale a conjunto vacío. Si alguno de los conjuntos indicados más arriba está presente en la AARE pero está vacío, no se puede utilizar ningún algoritmo para la ST correspondiente y, por consiguiente, esa ST no será utilizada mientras dura la asociación. Si alguno de los conjuntos indicados más arriba contiene exactamente un elemento (en la AARE), ese elemento designa el valor por defecto de la ST correspondiente mientras dura la asociación. Si alguno de los conjuntos indicados más arriba contiene más de un elemento (en la AARE) y uno de esos elementos corresponde al valor por defecto de la ST correspondiente (especificado en esta Recomendación o acordado entre las dos entidades comunicantes), ese elemento designa el valor por defecto de la ST correspondiente mientras dura la asociación. Si una entidad recibe un mensaje que no concuerda con los algoritmos elegidos, puede terminar la asociación. Si alguno de los conjuntos indicados más arriba está presente en la AARE y contiene elementos no contenidos en la AARQ, se declara error. Si se detecta un error, se libera la asociación. El proceso de negociación del algoritmo de seguridad se presenta de forma resumida en el cuadro 5-1 y se ilustra en el apéndice II.

Cuadro 5-1/Q.813 – Algoritmos negociados

<table>
<tr><td colspan="4" rowspan="3">Conjunto de algoritmos aceptables en AARE</td><td colspan="4">Conjunto de algoritmos aceptables en AARQ</td></tr>
<tr><td colspan="3">Presente</td><td rowspan="2">Ausente</td></tr>
<tr><td colspan="2">No vacío</td><td rowspan="2">Vacío (NULO)</td></tr>
<tr><td>2 o más elementos</td><td>1 elemento</td></tr>
<tr><td rowspan="4">Presente</td><td rowspan="3">No vacío (subconjunto de conjunto de AARQ)</td><td rowspan="2">2 o más elementos</td><td>No incluye algoritmo por defecto predefinido</td><td>Los algoritmos de AARE, sin algoritmo por defecto</td><td>Error</td><td>Error</td><td>Error</td></tr>
<tr><td>Incluye el algoritmo por defecto predefinido</td><td>Los algoritmos de AARE, con algoritmo por defecto predefinido</td><td>Error</td><td>Error</td><td>Error</td></tr>
<tr><td colspan="2">1 elemento</td><td>El algoritmo de AARE, es el algoritmo por defecto</td><td>El algoritmo seleccionado, es el algoritmo por defecto</td><td>Error</td><td>Error</td></tr>
<tr><td colspan="3">Vacío (NULO)</td><td>Ninguno</td><td>Ninguno</td><td>Ninguno</td><td>Ninguno</td></tr>
<tr><td colspan="4">Ausente</td><td>Ninguno</td><td>Ninguno</td><td>Ninguno</td><td>Sólo el algoritmo por defecto predefinido, si no hay algoritmo por defecto predefinido, ninguno</td></tr>
</table>

Además de la negociación de los algoritmos de seguridad, el STASE-ROSE sustenta también la negociación de diversos parámetros de criptación:

- identificación de las claves de criptación simétricas que pueden ser utilizadas;
- identificación de las claves públicas que pueden ser utilizadas;
- identificación de las claves de sello que pueden ser utilizadas;
- identificación de los ID de contraseña que pueden ser utilizados;
- especificación de los tamaños de claves públicas que pueden ser utilizados;
- especificación de las claves públicas que pueden ser utilizadas;
- clave secreta del emisor.

A diferencia de lo que ocurre con los algoritmos de criptación, esta Recomendación no proporciona ningún valor por defecto para ninguno de sus parámetros.

El STASE-ROSE soporta además el transporte de parámetros de criptación adicionales:

- especificación del vector de inicialización (para criptación DES) que será utilizado;
- especificación de los bits de retroalimentación que serán utilizados en el modo retroalimentación de salida de bit k o en los modos retroalimentación de cifrado de bit k de DES;
- especificación de un compendio de claves para la verificación de una clave pública;
- especificación de un número de secuencia para el mensaje en curso;

- especificación de una indicación de tiempo para el mensaje en curso;
- especificación de una clave simétrica criptada con una clave de criptación de claves simétricas;
- especificación de una clave simétrica criptada con una clave pública (del receptor);
- especificación de un identificador de clave de criptación de claves;
- provisión de certificados o trayectos de certificación de la Recomendación X.509 de las claves públicas del emisor que pueden ser utilizadas sin restricciones;
- provisión de certificados o trayectos de certificación de la Recomendación X.509 de las claves públicas del emisor que sólo pueden ser utilizadas para criptación;
- provisión de certificados o trayectos de certificación de la Recomendación X.509 de las claves públicas del emisor que sólo pueden ser utilizadas para firmas digitales;
- especificación de una clave de sesión simétrica criptada con la clave pública del receptor y firmada con la clave secreta del emisor.

Los valores de esos parámetros pueden ser proporcionados por cualquiera de las partes durante el establecimiento de la asociación, pero no son objeto de negociación. Por ejemplo, cada parte puede proporcionar sus propios certificados de claves públicas a la otra parte.

Al final de la fase de negociación (es decir, el establecimiento de la asociación), las dos entidades han llegado a un acuerdo respecto a qué ST sustentarán y qué algoritmos admitirán para esas ST. En algunos casos, pero no necesariamente en todos, llegan también a un acuerdo respecto a los algoritmos de defecto para algunas o todas las ST que han decidido sustentar. En ciertos casos también pueden llegar a un acuerdo sobre los valores de algunos o todos los parámetros de seguridad.

El STASE-ROSE admite la especificación y utilización de diferentes algoritmos para diferentes PDU. Además, permite que cada entidad comunicante utilice algoritmos distintos de los utilizados por la otra entidad para las mismas ST. Esto sólo tiene importancia naturalmente, si se ha llegado a un acuerdo respecto a más de un algoritmo para una ST determinada durante la fase de negociación. El STASE-ROSE proporciona algunas reglas directas sobre la especificación dinámica de algoritmos mientras dura la asociación:

- Cuando se haya llegado a un acuerdo respecto a exactamente un algoritmo para una determinada ST, el algoritmo para esa ST no será especificado de nuevo mientras dure la asociación.
- Si se ha llegado a un acuerdo respecto a más de un algoritmo para una determinada ST, y se ha acordado un algoritmo por defecto para esa ST, la especificación del algoritmo para esa ST, tras el establecimiento de la asociación, es de carácter opcional. Si no se especifica, entrará en vigor el algoritmo por defecto para la ST tras el establecimiento de la asociación.
- Si se ha llegado a un acuerdo respecto a más de un algoritmo para una determinada ST, y no se ha acordado ningún algoritmo por defecto para esa ST, cada una de las partes comunicantes puede especificar el algoritmo que utiliza para esa ST la primera vez que envía un mensaje que utiliza la ST. Cada unidad comunicante puede especificar el algoritmo que utiliza para esa ST en los mensajes subsiguientes, aunque no está obligada a hacerlo.
- Si se ha llegado a un acuerdo respecto a más de un algoritmo para una determinada ST, y no se ha acordado ningún algoritmo por defecto para esa ST, cada vez que la ST correspondiente sea utilizada por una unidad comunicante, después de la primera vez, el último algoritmo utilizado para esta ST por esa entidad es el algoritmo por defecto para esa entidad.

El procedimiento de negociación utiliza el parámetro selección de parámetros de criptación (EncryptionParametersSelection) definido en 5.3. Este parámetro se transfiere al ACSE como datos de usuario y se llevará en el campo información de usuario (User Information) de las PDU de la AARQ y la AARE.

5.3 Sintaxis abstracta para la negociación de parámetros de seguridad

Se registra el siguiente modulo para la negociación de parámetros de seguridad, a utilizar en el campo información de usuario (UserInformation) de un ACSE.

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

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,
        -- for k-bit output feedback mode or k-bit cipher feedback mode of 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,
         -- for verification of public keys
        blockSize               [13] INTEGER                            OPTIONAL,
        -- for square mod-n hashing
        keySizes                [14] SET OF INTEGER                     OPTIONAL,
        -- for 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,
        -- symmetric session key, encrypted with Key-Encryption-Key
        encryptedSymmetricKey   [19] INTEGER                            OPTIONAL,
        -- symmetric session key, encrypted with the receiver's public key
        keyEncryptionKey        [20] SEQUENCE (SIZE (1..3)) OF KeyId    OPTIONAL,
        -- one to three symmetric keys used for encrypting a session key
        keyListIds              [21] SET OF KeyListId                   OPTIONAL,
        -- list of encryption keys that can be used during the association
        encryptionCertificate   [22] SET OF EncryptionCertificate       OPTIONAL,

-- X.509 certificates or certification paths of the sender's public keys used for encryption only
```

```
        signatureCertificate            [23] SET OF SignatureCertificate        OPTIONAL,

-- X.509 certificates or certification paths of the sender's public keys used for digital signatures only --

        encryptedAuthenticatedSymmetricKeys [24] SET OF EncryptedAuthenticatedSymmetricKey
                                                                                OPTIONAL,

-- symmetric session key, encrypted with the receiver's public key and signed with sender's key--

        macAlgorithms                   [25] SET OF OBJECT IDENTIFIER   OPTIONAL,
        publicKeyCertificate            [26] SET OF PublicKeyCertificate        OPTIONAL,

-- X.509 certificates or certification paths of the sender's public keys with no usage restrictions --
        ...
        }

-- EncryptionParametersSelection is optionally used during association setup to negotiate which algorithms and other
-- encryption parameters will be supported during the association. It is not used in STASE-ROSE PDUs. --

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

END
```

5.3.1 Nombre de sintaxis abstracta

Esta Recomendación asigna el valor de identificador de objeto ASN.1

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

como un nombre de sintaxis abstracta para el conjunto de todos los valores de datos de presentación, cada uno de los cuales es un valor de tipo ASN.1.

STASE-A-ASSOCIATE-Information.EncryptionParametersSelection

El valor de descriptor de objeto correspondiente es "Información de usuario STASE-ROSE" ("STASE-ROSE-User-Information").

6 Modelo

En el entorno OSI, la comunicación entre procesos de aplicación se representa en términos de comunicación entre un par de entidades de aplicación (AE, *application entity*) que utilizan el servicio de presentación. La comunicación entre algunas entidades de aplicación quizás requiera transferencia segura de unidades de datos de protocolo de aplicación (APDU).

Las APDU enviadas por una AE (el emisor) son recibidas por la otra AE (el receptor). Con la transferencia segura se garantiza que las APDU transferidas por el emisor, pueden ser verificadas correctamente a efectos de integridad y/o que pueden ser comprobadas a efectos de no repudio, y/o que sólo puede ser comprendidas por el receptor al que se desea hacer llegar una determinada APDU. La transferencia segura implica transformaciones de seguridad (por ejemplo, la criptación) de las APDU provenientes de la entidad de aplicación de emisión antes de transferirlas y efectuar las transformaciones de seguridad inversas (por ejemplo, la descriptación) que preceden a su entrega a la entidad de aplicación de recepción. El STASE-ROSE sólo se ocupa de la transferencia segura de las unidades de datos de protocolo de aplicación del elemento de servicio de operaciones a distancia (ROSE).

La transferencia segura se lleva a cabo dentro del contexto de la asociación de aplicación. Una asociación de aplicación define la relación entre un par de AE, y consiste en el intercambio de información de control de protocolo de aplicación haciendo uso de los servicios de la capa de presentación. La AE que inicia la asociación se denomina entidad de iniciación de asociación, o iniciador de asociación, mientras que la AE que responde a la iniciación de una asociación de aplicación por otra AE se denomina AE de respuesta de aplicación, o respondedor de asociación.

La funcionalidad de una AE se descompone en un proceso de aplicación y un conjunto de elementos de servicio aplicación (ASE). Cada ASE se puede descomponer a su vez en un conjunto de elementos de servicio de aplicación (más primitivos). La interacción entre las AE se describe en términos de su utilización de los ASE.

La combinación específica de un proceso de aplicación y el conjunto de los ASE que comprenden una AE viene definida por el contexto de aplicación.

La figura 1 ilustra un ejemplo de contexto de aplicación en el que interviene un STASE-ROSE (sólo se muestra la relación de entidad par entre los ASE superiores).

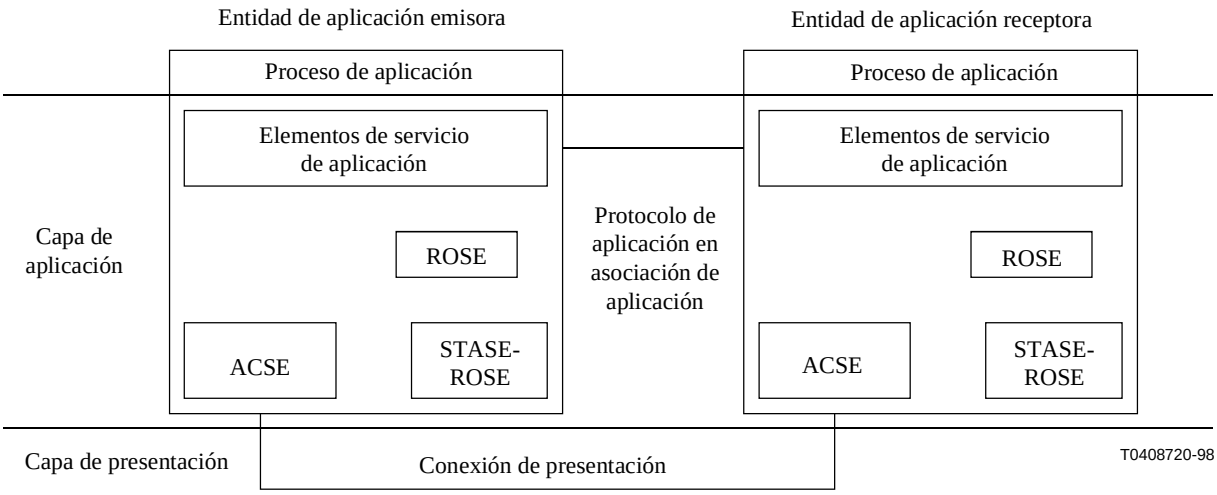

Figura 1/Q.813 – ASE obligatorios cuando se utiliza STASE-ROSE

Los ASE disponibles para un proceso de aplicación requieren comunicación por una asociación de aplicación. El control de la asociación de aplicación lo efectúa la aplicación que procesa los servicios proporcionados por el elemento de servicio de control de asociación (ACSE, *common management information service element*).

La figura 2 indica los ASE que deben estar presentes cuando se utiliza un STASE-ROSE. Muestra los ASE de la red de gestión de las telecomunicaciones (véase la Recomendación M.3010), en la que interviene el elemento de servicio común de información de gestión (CMISE) además del ACSE, el ROSE y el STASE-ROSE.

La presente Recomendación se puede utilizar con cualquier aplicación que utilice ROSE. Sin embargo, para simplificar la presentación, el texto de esta cláusula se refiere solamente al CMISE como el ASE que utiliza ROSE.

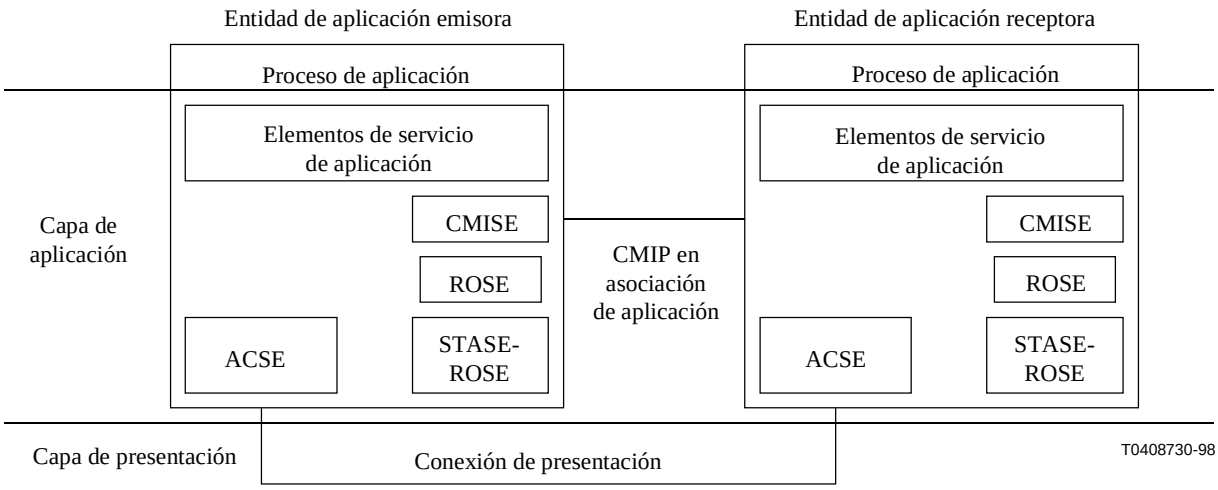

Figura 2/Q.813 – ASE presentes cuando se utiliza STASE-ROSE para asegurar el CMISE

7 Visión de conjunto de servicios

7.1 Servicios de asociación

Esta Recomendación no prevé servicios separados para el establecimiento y la liberación de asociaciones de aplicación. El proceso de aplicación del contexto de aplicación en el que interviene el STASE-ROSE se basa en los servicios de la Recomendación X.217 para el control de las asociaciones de aplicación.

Durante la fase de establecimiento de la asociación, diversos ASE del contexto de aplicación pueden intercambiar información de inicialización para establecer una asociación que utilice ACSE. Los requisitos del contexto de aplicación, de presentación y de sesión se llevan utilizando parámetros del servicio A-ASSOCIATE.

Los servicios A-RELEASE y A-ABORT de la Recomendación X.217 se utilizan para la terminación de una asociación. Pueden ser invocados por cualquier elemento de usuario.

7.2 Servicios STASE-ROSE

En el cuadro 7-1 se muestra el servicio STASE-ROSE.

Lo que sigue es una breve descripción del servicio STASE-ROSE:

– SR-TRANSFER

  El servicio SR-TRANSFER permite al ROSE iniciar la transferencia segura de las PDU de ROSE a un ROSE par.

Cuadro 7-1/Q.813 – Servicios STASE-ROSE

Servicio	Tipo
SR-TRANSFER	No confirmado

7.3 Relación con los servicios de presentación

El servicio STASE-ROSE requiere el acceso al servicio P-DATA.

Una sintaxis abstracta denominada con una sintaxis de transferencia compatible (negociada por la capa de presentación) constituye un contexto de presentación. La sintaxis de transferencia de las reglas de codificación básica (BER) será la que se negocie normalmente durante el establecimiento de una asociación con un contexto de aplicación que incluya STASE-ROSE. Incluso aunque el STASE-ROSE utilice las reglas de codificación distinguida (DER) para efectuar la codificación antes de llevar a cabo las transformaciones de seguridad, las DER no forman parte del contexto de aplicación (a menos que las DER sean utilizadas también por la capa de presentación).

7.4 Definición de servicios

7.4.1 Convenios

Esta Recomendación define servicios para el STASE-ROSE siguiendo los convenios descriptivos definidos en la Recomendación X.210. La definición del servicio STASE-ROSE incluye un cuadro en el que figura la relación de los parámetros de las primitivas. La presencia de los parámetros en un cuadro se señala mediante una de las indicaciones siguientes:

-- No aplicable

M Obligatorio (*mandatory*)

U Opción de usuario (*user option*)

C Condicional

Además, la notación (=) indica que el valor de un parámetro es idéntico al valor que se encuentra a su izquierda en el cuadro.

En esta Recomendación, el carácter "**.**" se utiliza para direccionar campos en un tipo ASN.1. Por ejemplo, **a.x** se utiliza para direccionar el campo **x** en el campo **a** de los tipos ASN.1 del ejemplo 1 y el ejemplo 2 que figuran a continuación.

```
Example1 ::= SEQUENCE {
        a  SEQUENCE {
                x INTEGER,
                y BOOLEAN
        },
        b  INTEGER
}

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

Example2 := SEQUENCE {
                a A,
                b INTEGER
}
```

7.4.2 Servicios de asociación

7.4.2.1 Establecimiento de la asociación

El servicio A-ASSOCIATE de la Recomendación X.217 es invocado por el proceso de aplicación de un contexto de aplicación en el que interviene un STASE-ROSE para establecer una asociación con un proceso de aplicación par. El establecimiento de la asociación es la primera fase de cualquier instancia de la actividad del servicio de transferencia segura.

El cuadro 7-2 contiene la relación de los parámetros definidos por esta Recomendación como la parte específica del STASE-ROSE del parámetro de información de usuario del servicio A-ASSOCIATE. Esta información es especificada por el iniciador de asociación e intercambiada cuando se establece una asociación. El intercambio de esta información de inicialización es opcional antes de utilizar los servicios STASE-ROSE.

Cuadro 7-2/Q.813 – Información de usuario de A-ASSOCIATE

Nombre de parámetro	Petición/Indicación	Respuesta/Confirmación
symmetricKeyIds	U	C
publicKeyIds	U	C
sealKeyIds	U	C
signatureKeyIds	U	C
passwordIds	U	C
initializationVector	U	U
feedBackBits	U	U
symmetricAlgorithms	U	C
publicKeyAlgorithms	U	C
signatureAlgorithms	U	C
sealAlgorithms	U	C
hashAlgorithms	U	C
keyDigest	U	U
blockSize	U	U
keySize	U	C
publicKeys	U	U
sequenceNumber	U	U
timeStamp	U	U
encryptedKey	U	U
encryptedSymmetricKey	U	U
keyEncryptionKey	U	U
keyListIds	U	C
publicKeyCertificates	U	U
encryptionCertificates	U	U
signatureCertificates	U	U
encryptedAuthenticatedSymmetricKeys	U	U
macAlgorithms	U	C

La condición C del cuadro 7-2 consiste en que el parámetro está presente en la respuesta/confirmación solamente si lo está en la petición/indicación. Si está presente en la petición/indicación pero no en la respuesta/confirmación, la respuesta/confirmación se interpreta como si el parámetro estuviera presente con un valor nulo.

A continuación se describe con más detalle el significado de estos parámetros y la respuesta que se espera del respondedor:

- **identificadores de claves simétricas (symmetricKeyIds)**: Conjunto de identificadores de claves de las claves que se han de utilizar en esta asociación para criptación simétrica. El respondedor de la aplicación responderá con el mismo conjunto de identificadores o un subconjunto del mismo si el parámetro symmetricKeyIds está presente en la indicación A-ASSOCIATE.
- **identificadores de claves públicas (publicKeyIds)**: Conjunto de identificadores de claves de las claves que se han de utilizar en esta asociación para criptación de claves públicas. El respondedor de la aplicación responderá con el mismo conjunto de identificadores o un subconjunto del mismo si el parámetro publicKeyIds está presente en la indicación A-ASSOCIATE.
- **identificadores de claves de sello (sealKeyIds)**: Conjunto de identificadores de claves de las claves que se han de utilizar en esta asociación a efectos de sellado. El respondedor de la aplicación responderá con el mismo conjunto de identificadores o un subconjunto del mismo si el parámetro sealKeyIds está presente en la indicación A-ASSOCIATE.
- **identificadores de claves de firma (signatureKeyIds)**: Conjunto de identificadores de claves de las claves que se han de utilizar en esta asociación para firma digital. El respondedor de la aplicación responderá con el mismo conjunto de identificadores o un subconjunto del mismo si el parámetro signatureKeyIds está presente en la indicación A-ASSOCIATE.
- **identificadores de contraseñas (passwordIds)**: Conjunto de identificadores de contraseñas de las contraseñas que se han de utilizar en esta asociación. El respondedor de la aplicación responderá con el mismo conjunto de identificadores o un subconjunto del mismo si el parámetro passwordIds está presente en la indicación A-ASSOCIATE.
- **vector de inicialización (initializationVector)**: Vector de inicialización (IV) que se han de utilizar para la criptación DES en el modo concatenación de bloques cifrados (CBC). Cada parte puede utilizar un IV diferente para los mensajes que envía.
- **bits de retroalimentación (feedBackBits)**: Bits de retroalimentación que se han de utilizar para la criptación DES en el modo retroalimentación de cifrado de bit k o el modo retroalimentación de salida de bit b. Cada parte puede utilizar un valor diferente para los bits de retroalimentación de los mensajes que envía.
- **algoritmos simétricos (symmetricAlgorithms)**: Conjunto de algoritmos simétricos que el iniciador de asociación puede admitir. El respondedor de asociación responderá con el mismo conjunto de algoritmos o un subconjunto del mismo si el parámetro symmetricAlgorithms está presente en la indicación A-ASSOCIATE.
- **algoritmos de claves públicas (publicKeyAlgorithms)**: Conjunto de algoritmos de claves públicas que el iniciador de asociación puede admitir. El respondedor de asociación responderá con el mismo conjunto de algoritmos o un subconjunto del mismo si el parámetro publicKeyAlgorithms está presente en la indicación A-ASSOCIATE.

- **algoritmos de firma (signatureAlgorithms)**: Conjunto de algoritmos de firma digital que el iniciador de asociación puede admitir. El respondedor de asociación responderá con el mismo conjunto o un subconjunto del mismo si el parámetro signatureAlgorithms está presente en la indicación A-ASSOCIATE.
- **algoritmos de sello (sealAlgorithms)**: Conjunto de algoritmos de sello que el iniciador de la asociación puede admitir. El respondedor de asociación responderá con el mismo conjunto de algoritmos o un subconjunto del mismo si el parámetro sealAlgorithms está presente en la indicación A-ASSOCIATE.
- **algoritmos de troceado (hashAlgorithms)**: Conjunto de algoritmos de la función troceado que el iniciador de la asociación puede admitir. El respondedor de la asociación responderá con el mismo conjunto de algoritmos o un subconjunto del mismo si el parámetros hashAlgorithms está presente en la indicación A-ASSOCIATE.
- **compendio de claves (keyDigest)**: Compendio de mensajes [imprenta digital (fingerprint)] de una clave pública, utilizado para verificar la validez de una clave pública.
- **tamaño de bloques (blockSize)**: Tamaño de los bloques que se han de utilizar para la función troceado cuadrado módulo n. Cada parte puede elegir un tamaño de bloques diferente para los mensajes que envía.
- **tamaño de clave** (**keySize)**: Tamaño de la clave del algoritmo de criptación RSA. El respondedor de asociación responderá con un tamaño de clave igual o diferente si está presente el parámetro keySize en la indicación A-ASSOCIATE.
- **claves públicas (publicKeys)**: Conjunto de claves públicas que ha de utilizar el emisor en esta asociación. El respondedor de asociación puede responder con el conjunto de claves públicas del respondedor.
- **número de secuencia (sequenceNumber)**: Número de secuencia de arranque de las PDU del ROSE, si la asociación debiera estar protegida contra ataques tales como los de reactuación y supresión. Cada parte puede optar por comenzar con un número de secuencia diferente. Si este parámetro está presente, el proveedor STASE-ROSE de cualquier parte que lo suministre asignará un número de secuencia a cada APDU del STASE-ROSE enviada en la asociación de aplicación.
- **indicación de tiempo (timeStamp)**: Hora en la que la petición A-ASSOCIATE fue iniciada por el iniciador de asociación. La interpretación de este parámetro depende de la implementación y queda fuera del alcance de la presente Recomendación. Si el respondedor de asociación envía la timeStamp, su valor corresponderá a la hora en la que se emitió la primitiva de respuesta A-ASSOCIATE.
- **clave criptada (encryptedKey)**: Clave simétrica utilizada para (parte de) la asociación y criptada con una clave de criptación de claves (KEK, *key encryption key*) simétricas.
- **clave simétrica criptada (encryptedSymmetricKey)**: Clave simétrica utilizada para (parte de) la asociación y criptada con la clave pública del receptor.
- **clave de criptación de claves (keyEncryptionKey)**: Identifica una, dos o tres claves simétricas que se han de utilizar como KEK simétricas.
- **identificadores de listas de claves (keyListIds)**: Conjunto de identificadores de listas de claves de criptación simétricas que propone utilizar el iniciador de asociación. El respondedor de asociación responderá con el mismo conjunto de identificadores o un subconjunto del mismo si el parámetro keyListIds está presente en la indicación A-ASSOCIATE.
- **certificados de claves públicas (publicKeyCertificates)**: Trayecto de certificación de la Recomendación X.509 que certifica la clave pública del emisor.

- **certificados de criptación (encryptionCertificates)**: Trayecto de certificación de la Recomendación X.509 que certifica la clave pública del emisor que sólo puede ser utilizada para criptación.
- **certificados de firmas (signatureCertificates)**: Trayecto de certificación de la Recomendación X.509 que certifica la clave pública del emisor que sólo puede ser utilizada para firmas digitales.
- **claves simétricas autenticadas criptadas (encryptedAuthenticatedSymmetricKeys)**: Clave simétrica utilizada para (parte de) la asociación y criptada con la clave pública del receptor, a la que sigue una indicación de tiempo (hora generalizada), el ID del emisor, el ID del receptor, y una firma calculada en la representación ASCII de esos cuatro campos utilizando la clave privada del emisor.
- **algoritmos MAC (macAlgorithms)**: Conjunto de algoritmos MAC que el iniciador de asociación puede admitir. El respondedor de asociación responderá con el mismo conjunto de algoritmos o un subconjunto del mismo si el parámetro macAlgorithms está presente en la indicación A-ASSOCIATE.

7.4.2.2 Liberación de la asociación

El servicio A-RELEASE de la Recomendación X.217 es invocado por el proceso de aplicación en el contexto de aplicación en el que interviene un STASE-ROSE para pedir la terminación ordenada de una asociación entre entidades de aplicación pares. Esta Recomendación no especifica ninguna utilización de los parámetros del servicio A-RELEASE.

El servicio A-ABORT es invocado por el proceso de aplicación para pedir la terminación repentina de la asociación de aplicación.

7.4.3 Servicio SR-TRANSFER

El servicio SR-TRANSFER es utilizado por un usuario STASE-ROSE (ROSE) para transferir una PDU de ROSE de manera segura al usuario STASE-ROSE (ROSE) par.

La estructura de servicio conexa consiste en dos primitivas de servicio, como se ilustra en la figura 3.

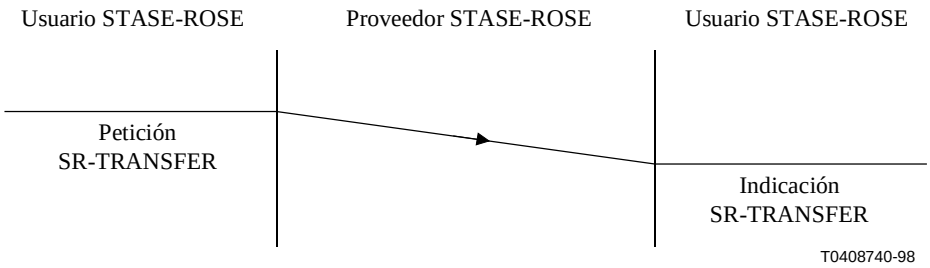

Figura 3/Q.813 – Primitivas del servicio SR-TRANSFER

7.4.4 Parámetros de SR-TRANSFER

En el cuadro 7.3 se presenta la relación de parámetros del servicio SR-TRANSFER.

Cuadro 7-3/Q.813 – Parámetros de SR-TRANSFER

Nombre de parámetro	Petición	Indicación
PDU de ROSE (ROSE-PDU)	M	M(=)
Tipo de criptación (Encryption-Type)	M	M(=)
Parámetros de criptación (Encryption-Parameters)	U	C(=)

7.4.4.1 PDU de ROSE

Este parámetro identifica la PDU de ROSE que se tiene que transferir. Ha de ser suministrado por el solicitante del servicio y los valores de los datos deberán estar de acuerdo con la definición de [unidades de datos de protocolo de aplicación de ROSE (ROSEapdus)] de la Recomendación X.229.

7.4.4.2 Tipo de criptación

Este parámetro identifica el tipo de las ST que desea el usuario del servicio para la PDU del ROSE en curso. Lo que sigue es una lista de valores válidos del tipo (se señala que la utilización de valores por defecto para los parámetros de criptación se especifica en 5.2.1):

- **claro (clear)**: No se desea ST.
- **confidencial simple (simpleConfidential)**: Protección de la privacidad de la PDU total utilizando los valores por defecto admitidos por el proveedor del servicio.
- **confidencial (confidential)**: Protección de la privacidad de la PDU total utilizando los parámetros proporcionados en el parámetro parámetros de criptación (Encryption-Parameters).
- **cifrado público simple (simplePublicEnciphered)**: Protección de la privacidad de la PDU total utilizando la clave pública por defecto admitida por el proveedor del servicio.
- **cifrado público (publicEnciphered)**: Protección de la privacidad de la PDU utilizando la clave pública proporcionada por el parámetro parámetros de criptación (Encryption-Parameters).
- **troceado simple (simpleHashed)**: MAC basado en la función troceado de la PDU utilizando valores por defecto.
- **troceado (hashed)**: MAC en la función troceado de la PDU utilizando los parámetros proporcionados en el parámetro parámetros de criptación.
- **sellado simple (simpleSealed)**: Sellado de la PDU utilizando los valores por defecto.
- **sellado (sealed)**: Sellado de la PDU utilizando los parámetros proporcionados en el parámetro parámetros de criptación.
- **firmado simple (simpleSigned)**: Firma digital de la PDU utilizando los valores por defecto.
- **firmado (signed)**: Firma digital de la PDU utilizando los parámetros proporcionados en el parámetro parámetros de criptación.
- **firmado confidencial simple (simpleConfidentialSigned)**: Protección de la privacidad de la PDU total y firma digital de la PDU utilizando los valores por defecto.
- **firmado confidencial (confidentialSigned)**: Protección de la privacidad de la PDU total y firma digital de la PDU utilizando los parámetros proporcionados en el parámetro parámetros de criptación.
- **MAC confidencial simple (simpleConfidentialMAC)**: Protección de la privacidad de la PDU total y MAC de la PDU utilizando los valores por defecto.

- **MAC confidencial (confidentialMAC)**: Protección de la privacidad de la PDU total y MAC de la PDU utilizando los parámetros proporcionados en el parámetro parámetros de criptación.
- **sellado confidencial simple (simpleConfidentialSealed)**: Protección de la privacidad de la PDU total y sello de la PDU utilizando los valores por defecto.
- **sellado confidencial (confidentialSealed)**: Protección de la privacidad de la PDU total y sello de la PDU utilizando los parámetros proporcionados en el parámetro parámetros de criptación.

7.4.4.3 Parámetros de criptación

Este parámetro identifica los parámetros que se han de utilizar para las transformaciones de seguridad. La presencia de este parámetro depende del tipo de criptación seleccionado por el usuario (como se describe en la subcláusula anterior).

8 Interacción entre elemento de servicio de aplicación

En esta cláusula se describen las interacciones entre procesos de aplicación, ACSE, ROSE, STASE-ROSE, el usuario ROSE (por ejemplo, CMISE) y los servicios de la capa de presentación de las diferentes etapas de la comunicación entre dos entidades de aplicación. Son posibles otras interacciones, que dan lugar al mismo conjunto de mensajes intercambiados entre los sistemas comunicantes y la misma funcionalidad. La elección de esas interacciones es un asunto local. El anexo A circunscribe esta discusión al caso en el que el CMISE es el usuario ROSE.

8.1 Establecimiento de la asociación

La figura 4 ilustra la secuencia de interacciones entre el proceso de aplicación, diversos ASE y el proveedor del servicio de presentación durante la fase de establecimiento de la asociación.

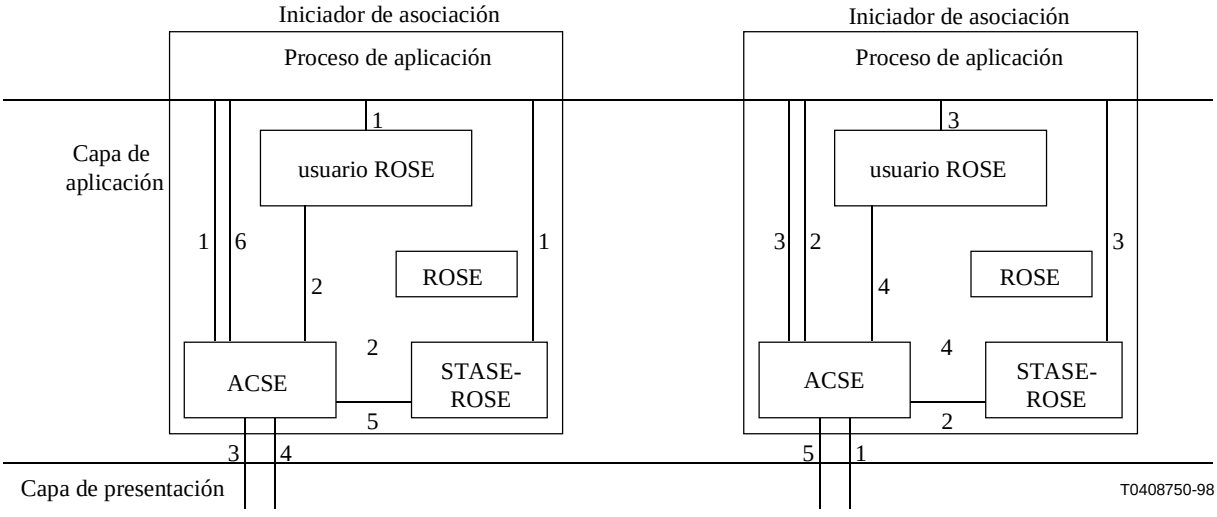

Figura 4/Q.813 – Interacción durante el establecimiento de la asociación

8.1.1 Iniciador de asociación

A continuación se describen las interacciones en el lado iniciador de asociación de la figura 4:

1) El proceso de aplicación de la entidad de aplicación en que interviene un STASE-ROSE emite una petición A-ASSOCIATE hacia el ACSE para establecer una asociación de aplicación. Si se desea la autenticación de la entidad par, el proceso de aplicación proporciona al ACSE el valor del autenticador que se ha de llevar en el campo valor de autenticación (Authentication-value) de la PDU de la AARQ (utilizando la FU autenticación del ACSE). Durante la misma fase, el proceso de aplicación puede también informar al STASE-ROSE y al o a los ASE del usuario ROSE (por ejemplo, el CMISE) de la asociación pedida y proporcionar al STASE-ROSE cualquiera de los valores propuestos para (algunos de) los parámetros de criptación.

2) El STASE-ROSE proporciona al ACSE cualesquiera valores propuestos para (algunos de) los parámetros de criptación. El mecanismo según el cual el STASE-ROSE informa al ACSE es un asunto que depende de la implementación y no se trata en la presente Recomendación. Esta información se llevará en el campo información de usuario ACSE utilizando la selección de parámetros de criptación definida en 5.3. Durante la misma fase, el o los ASE del usuario ROSE pueden proporcionar también al ACSE información de importancia para esos ASE. Toda esa información se lleva en el campo información de usuario ACSE. El campo información de usuario ACSE definido en la Rec. UIT-T X.227 | ISO/CEI 8650-1 consta de una SEQUENCE OF EXTERNAL. La información para el STASE-ROSE, si hay alguna, se llevará en el primer EXTERNAL. El contexto de aplicación deberá especificar cuál o cuáles EXTERNAL llevarán la información para cada uno de sus otros ASE.

3) El ACSE emite una petición P-CONNECT hacia el proveedor del servicio presentación para establecer una asociación de aplicación.

   El proveedor del servicio de presentación transfiere a continuación la petición P-CONNECT y recibe una respuesta (no mostrada arriba).

4) El proveedor del servicio de presentación emite una primitiva de confirmación P-CONNECT hacia el ACSE, confirmando el establecimiento de una conexión de presentación.

5) El ACSE informa al STASE-ROSE sobre el establecimiento de una nueva asociación de aplicación y se proporciona los valores (si hay alguno) de los parámetros de criptación. El mecanismo según el cual el ACSE informa al STASE-ROSE es un asunto que depende de la implementación y no se trata en la presente Recomendación.

6) El ACSE emite una primitiva de confirmación A-ASSOCIATE hacia el proceso de aplicación confirmando el establecimiento de la asociación. El ACSE proporciona al proceso de aplicación los valores (si hay alguno) de los parámetros de criptación. El mecanismo según el cual el ACSE informa al proceso de aplicación es un asunto que depende de la implementación y no se trata en la presente Recomendación.

8.1.2 Respondedor de asociación

A continuación se describen las interacciones en el lado respondedor de asociación de la figura 4.

El proveedor del servicio de presentación recibe una petición de conexión del proveedor del servicio de presentación distante:

1) El proveedor del servicio de presentación emite una primitiva de indicación P-CONNECT hacia el ACSE, informando del interés de unos usuarios del servicio distantes en establecer una asociación.

2) El ACSE emite una primitiva de indicación A-ASSOCIATE hacia el proceso de aplicación. Durante la misma fase, el ACSE informa al STASE-ROSE sobre la petición establecimiento de una nueva asociación de aplicación y proporciona los valores propuestos (si se propone alguno) de los parámetros de criptación. El ACSE proporciona además información específica de ASE (llevada en el campo información de usuario), si hay alguna, a los usuarios ROSE. El mecanismo según el cual el ACSE informa al STASE-ROSE es un asunto que depende de la implementación y no se trata en la presente Recomendación.

3) El proceso de aplicación emite una primitiva de respuesta A-ASSOCIATE hacia el ACSE aceptando o rechazando la asociación de aplicación. Si la indicación A-ASSOCIATE contiene valores propuestos para (algunos de) los parámetros de criptación, el proceso de aplicación puede indicar al STASE-ROSE qué valores de esos parámetros de criptación deberían ser aceptados. El mecanismo según el cual el proceso de aplicación informa al STASE-ROSE es un asunto que depende de la implementación y no se trata en la presente Recomendación. Durante esta fase, el proceso de aplicación puede informar también al o a los ASE de usuario ROSE de la asociación.

4) El STASE-ROSE proporciona al ACSE los valores aceptados, si se acepta alguno, para los parámetros de criptación en la AARQ. El mecanismo según el cual el STASE-ROSE informa al ACSE es un asunto que depende de la implementación y no se trata en la presente Recomendación. Esta información se llevará en el campo información de usuario ACSE utilizando la selección de parámetros de criptación definida en 5.3. Durante la misma fase, el o los ASE del usuario ROSE pueden proporcionar también al ACSE información de importancia para esos ASE. El campo información de usuario ACSE definido en la Rec. UIT-T X.227 | ISO/CEI 8650-1 consta de una SEQUENCE OF EXTERNAL. La información para el STASE-ROSE, si hay alguna, se llevará en la primera EXTERNAL. El contexto de aplicación deberá especificar cuál o cuáles EXTERNAL llevarán la información para cada uno de sus otros ASE.

5) El ACSE emite una primitiva de respuesta P-CONNECT hacia el proveedor del servicio de presentación aceptando o rechazando el establecimiento de la asociación.

8.2 Liberación de la asociación

La figura 5 ilustra la secuencia de interacciones entre el proceso de aplicación, diversos ASE y el proveedor del servicio de presentación durante la fase de liberación de la asociación.

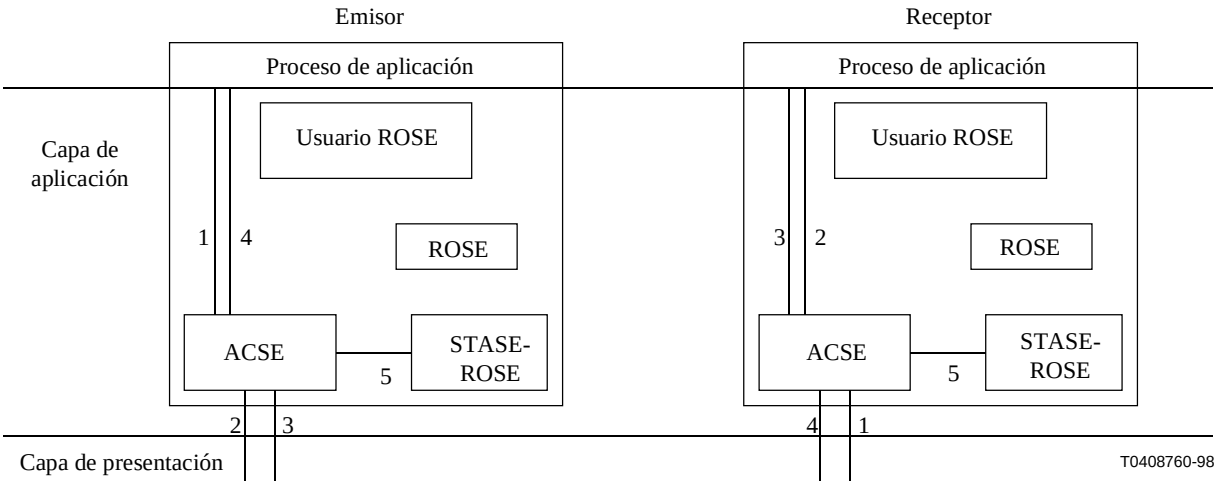

Figura 5/Q.813 – Interacción durante la liberación de la asociación

8.2.1 Emisor

A continuación se describen las interacciones en el lado emisor de la figura 5:

1) El proceso de aplicación del contexto de aplicación en el que interviene un STASE-ROSE emite una petición A-RELEASE hacia el ACSE para liberar una asociación de aplicación.

2) El ACSE emite una petición P-RELEASE hacia el proveedor del servicio de presentación para liberar una asociación de aplicación.

   El proveedor del servicio de presentación transfiere a continuación la petición P-RELEASE a la entidad de aplicación par y recibe una respuesta (no mostrada arriba).

3) El proveedor del servicio de presentación emite una primitiva de confirmación P-RELEASE hacia el ACSE, confirmando la liberación de una conexión de presentación.

4) El ACSE emite una primitiva de confirmación A-RELEASE hacia el proceso de aplicación confirmando la liberación de la asociación de aplicación.

5) El ACSE informa al STASE-ROSE y a los otros ASE (no mostrados en la figura) sobre la liberación de la asociación de aplicación. El mecanismo según el cual el ACSE informa al STASE-ROSE y a los otros ASE es un asunto que depende de la implementación y no se trata en la presente Recomendación.

8.2.2 Receptor

A continuación se describen las interacciones en el lado receptor de la figura 5.

El proveedor del servicio de presentación recibe una petición de liberación del proveedor del servicio de presentación distante.

1) El proveedor del servicio de presentación emite una primitiva de indicación P-RELEASE hacia el ACSE, informando del interés de unos usuarios del servicio distantes en liberar una asociación.

2) El ACSE emite una primitiva de indicación A-RELEASE hacia el proceso de aplicación.

3) El proceso de aplicación emite una primitiva de respuesta A-RELEASE hacia el ACSE aceptando la liberación de la asociación de aplicación.

4) El ACSE emite una primitiva de respuesta P-RELEASE hacia el proveedor del servicio de presentación aceptando la liberación de la asociación.

5) El ACSE informa al STASE-ROSE y a los otros ASE sobre la liberación de la asociación de aplicación. El mecanismo según el cual el ACSE informa al STASE-ROSE y a los otros ASE es un asunto que depende de la implementación y no se trata en la presente Recomendación.

8.3 Aborto de la asociación

La figura 6 ilustra la secuencia de interacciones entre el proceso de aplicación, diversos ASE y el proveedor del servicio de presentación durante el aborto de la asociación.

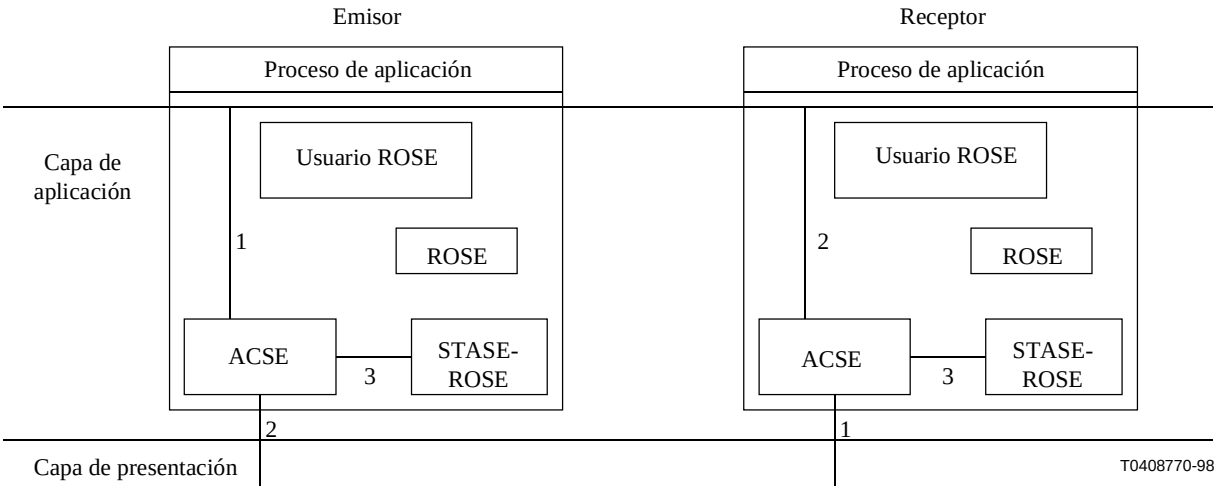

Figura 6/Q.813 – Interacción durante el aborto de la asociación

8.3.1 Emisor

A continuación se describen las interacciones en el lado de emisor de la figura 6:

1) El proceso de aplicación que utiliza el STASE-ROSE emite una petición A-ABORT hacia el ACSE para abortar una asociación de aplicación.

2) El ACSE emite una petición P-ABORT hacia el proveedor del servicio de presentación para abortar una conexión de presentación.

3) El ACSE informa al STASE-ROSE y a los otros ASE sobre el aborto de la asociación de aplicación. El mecanismo según el cual el ACSE informa al STASE-ROSE y a los otros ASE es un asunto que depende de la implementación y no se trata en la presente Recomendación.

8.3.2 Receptor

A continuación se describen las interacciones en el lado receptor de la figura 6.

El proveedor del servicio de presentación detecta el aborto de una conexión de presentación:

1) El proveedor del servicio de presentación emite una primitiva de indicación P-ABORT hacia el ACSE, informando de que ha sido abortada una conexión de aplicación.

2) El ACSE emite una primitiva de indicación A-ABORT hacia el proceso de aplicación.

3) El ACSE informa al STASE-ROSE y a los otros ASE sobre el aborto de la aplicación de asociación. El mecanismo según el cual el ACSE informa al STASE-ROSE y a los otros ASE es un asunto que depende de la implementación y no se trata en la presente Recomendación.

8.4 Transferencia de datos

La figura 7 muestra las interacciones entre el proceso de aplicación, diversos ASE y el proveedor del servicio de presentación durante la fase de transferencia de datos.

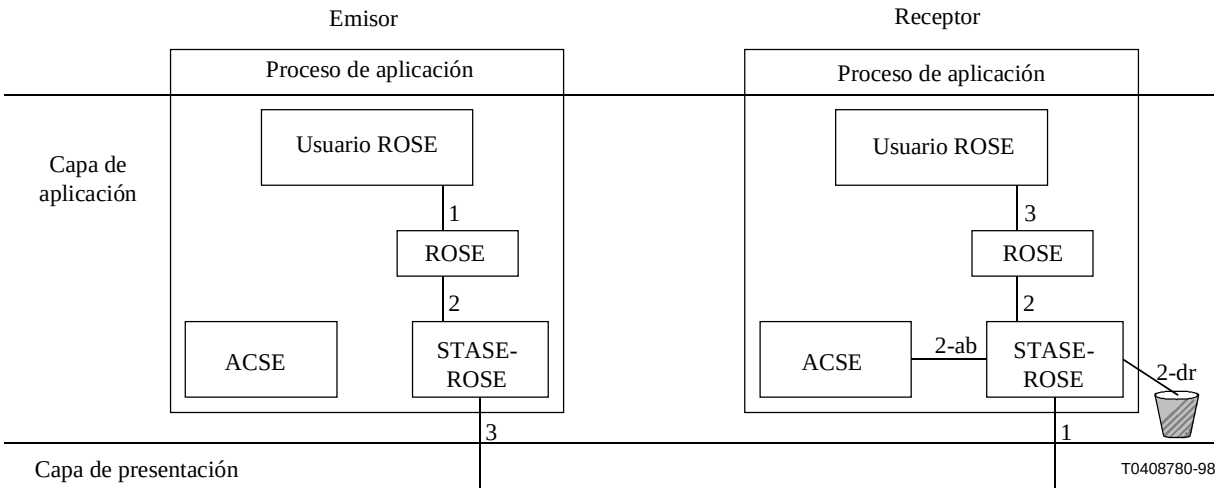

Figura 7/Q.813 – Interacción durante la transferencia de datos

8.4.1 Emisor

A continuación se describen las interacciones en el lado receptor de la figura 7:

1) El usuario ROSE, impulsado por el proceso de aplicación (no se muestra), emite una primitiva de petición o respuesta ROSE hacia el ROSE, pidiendo la transferencia de datos.

2) El ROSE emite la primitiva de petición SR-TRANSFER hacia el STASE-ROSE, pidiendo la transferencia segura de los datos.

3) El STASE-ROSE efectúa la codificación requerida y la transformación de seguridad en la PDU del ROSE (presente en los parámetros de petición) y emite una primitiva de petición P-DATA hacia el proveedor del servicio de presentación. El STASE-ROSE, tal como se define actualmente, puede ser utilizado por los servicios correspondientes a todas las unidades funcionales del CMISE excepto los servicios ampliados.

8.4.2 Receptor

A continuación se describen las interacciones en el lado emisor de la figura 7:

1) El proveedor del servicio de presentación emite una primitiva de indicación P-DATA hacia el STASE-ROSE, informando de la llegada de datos procedentes de la entidad de aplicación par en una conexión de aplicación.

2) El STASE-ROSE efectúa las transformaciones de seguridad inversas en los datos entrantes, comprueba la validez de la PDU (por ejemplo, validez del sello o de la firma, vigencia de la indicación de tiempo, valor del número de secuencia) y, si son válidos, emite una primitiva de indicación SR-TRANSFER hacia el ROSE.

   El ROSE emite una primitiva de indicación o confirmación ROSE hacia el usuario ROSE que a continuación informa (no se muestra) al proceso de aplicación de la llegada de datos procedentes de una entidad de aplicación par.

   Es posible que el STASE-ROSE receptor encuentre que la APDU entrante es inaceptable (por ejemplo, por fallos en la descriptación). En tal caso, la acción a ejecutar por el STASE-ROSE es un asunto local. No obstante, la presente Recomendación recomienda las dos alternativas siguientes:

   - La implementación del STASE-ROSE receptor puede prescindir de la APDU entrante como se muestra con **2-dr** en la figura 7.

- El STASE-ROSE receptor puede emitir una primitiva A-ABORTO hacia el ACSE, como se muestra en **2-ab** en la figura 7.

En cualquier caso, se recomienda (no se muestra) que el suceso sea notificado al elemento de usuario, que se registre en un registro de auditoría de seguridad y que se emita una alarma hacia el administrador de seguridad local.

9 Protocolo STASE-ROSE

El protocolo especificado en esta cláusula soporta los servicios STASE-ROSE descritos anteriormente. Como ya se ha mencionado, el STASE-ROSE utiliza el servicio P-DATA de la capa de presentación para transferir de manera segura las PDU de un ROSE.

La máquina de protocolo de STASE-ROSE (SRPM, *STASE-ROSE-protocol-machine*) se comunica con el ROSE mediante las primitivas del servicio SR-TRANSFER descritas más arriba. La SRPM es excitada por las peticiones de servicio procedentes del ROSE, y por las primitivas de indicación procedentes del servicio de presentación. A su vez, la SRPM emite primitivas de indicación hacia el ROSE, y primitivas de petición hacia el servicio de presentación. Se utilizan las primitivas del servicio de presentación petición P-DATA e indicación P-DATA.

La recepción de una primitiva del servicio STASE-ROSE o la recepción de una primitiva del servicio de presentación, y la generación de las acciones dependientes, son asuntos locales que quedan fuera del alcance de la presente Recomendación.

Durante el intercambio de las APDU, se supone la existencia de una asociación de aplicación entre las AE pares. La asociación se establecerá utilizando los parámetros especificados en 7.4.2.

NOTA – Cada asociación de aplicación puede ser identificada en un sistema de extremo mediante un mecanismo interno, dependiente de la implementación, al que pueden referirse el usuario del servicio STASE-ROSE (ROSE) y la SRPM.

9.1 Definición de la sintaxis abstracta de las APDU

Además de los tipos ASN.1 definidos en la Recomendación X.229, se definen los siguientes tipos para el STASE-ROSE.

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


DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS everything

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)}

-- the referenced module and corresponding syntax are found in Annex D/Rec. 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 represents the DER encoded and encrypted ROSE PDU.
-- encryptionParameters represents the parameters used for encryption.

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

-- SimplePublicEnciphered represents the DER encoded and public key encrypted ROSE PDU.
-- A large PDU may be broken into smaller blocks, each of which may be encrypted
-- as an INTEGER. The size of such blocks depends on the public key encryption algorithm
-- used and on the size of the public key; specification of such block sizes is outside the
-- scope of this Recommendation.
-- In some cases the result of public key encryption may be represented as an OCTET STRING.

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

-- publicEncrypted represents the DER encoded and public key encrypted ROSE PDU.
-- encryptionParameters represents the parameters used for encryption.

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

-- hashValue represents the message digest resulting from hashing the DER encoded
-- ROSE PDU.
-- encryptionParameters represents the parameters used for the hashing algorithm.

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

```
-- data represents the DER encoded ROSE PDU
-- hash represents the hash value either as a simple OCTET STRING or the Hash
-- structure defined above.

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

-- sealValue represents the seal value for the DER encoded ROSE PDU.
-- encryptionParameters represents the parameters used by the seal generation algorithm.

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

-- data represents the DER encoded ROSE PDU
-- seal represents the seal value either as a simple OCTET STRING or the Seal structure
-- defined above.

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

-- signatureValue represents the signature for the DER encoded ROSE PDU.
-- encryptionParameters represents the parameters for the signature algorithm.

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

-- data contains the DER encoding of the ROSE PDU.
-- signature represents the signature of the DER encoded ROSE PDU, either as a simple
-- INTEGER or the Signature structure defined above.

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

-- encrypted represents the encryption of the DER encoded ROSE PDU.
-- signature represents the signature of the DER encoded ROSE PDU in either a simple form
-- or as Signature type defined above.

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

```
-- encrypted represents the encryption of the DER encoded ROSE PDU.
-- mac represents the MAC of the DER encoded ROSE PDU in either a simple form
-- or as Hash type defined above.

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

-- encrypted represents the encryption of the DER encoded ROSE PDU.
-- seal represents the seal of the DER encoded ROSE PDU in either a simple form
-- or as Seal type defined above.

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,
        -- for k-bit output feedback mode or k-bit cipher feedback mode of 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,
        -- for verification of public keys
        blockSize                [13] INTEGER               OPTIONAL,
        -- for square mod-n hashing
        keySize                  [14] INTEGER               OPTIONAL,
        -- for 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,
        -- symmetric session key, encrypted with Key-Encryption-Key
        encryptedSymmetricKey    [19] INTEGER      OPTIONAL,
        -- symmetric session key, encrypted with the receiver's public key
        keyEncryptionKey    [20] SEQUENCE (SIZE (1..3)) OF KeyId          OPTIONAL,
        -- one to three symmetric keys used for encrypting a session key
        publicKeyCertificate     [21]    PublicKeyCertificate          OPTIONAL,
-- X.509 certificate or certification path of the sender's public key with no usage restrictions
        encryptionCertificate    [22] EncryptionCertificate       OPTIONAL,
-- X.509 certificate or certification path of the sender's public key used for encryption only
        signatureCertificate     [23] SignatureCertificate        OPTIONAL,
-- X.509 certificate or certification path of the sender's public key used for digital signatures only --
encryptedAuthenticatedSymmetricKey [24] EncryptedAuthenticatedSymmetricKey OPTIONAL,
-- symmetric session key, encrypted with the receiver's public key and signed with sender's key--
        macAlgorithm             [25] OBJECT IDENTIFIER          OPTIONAL,
                ...
                }

-- EncryptionParameters is an extensible type that is used as a catch-all for any
-- parameters that may be used by any of the STs. In most applications only a small
```

```
-- number, if any, of the components of EncryptionParameters will be used.

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,
        -- symmetric session key, encrypted with the receiver's public key
                                 time            GeneralizedTime,
                                 sender          SenderId,
                                 receiver        ReceiverId,
                                 signature       Signature
-- the signature is computed over ASCII representation of the preceding four fields with the sender's private key
                                 }
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 Nombre de sintaxis abstracta

Esta Recomendación asigna el siguiente identificador de objeto:

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

como un nombre de sintaxis abstracta para el conjunto de valores de datos de presentación, cada uno de los cuales es un valor del tipo ASN.1

Secure-Remote-Operations-APDUs.SR-APDU

donde los componentes de los argumentos de las PDU del ROSE los llena el usuario del ROSE.

El valor del descriptor de objeto correspondiente es "STASE-ROSE-Data".

9.3 Identificadores de algoritmos

A menos que las entidades comunicantes acuerden otra cosa (por medios que quedan fuera del alcance de la presente Recomendación), los algoritmos de ST se limitarán a los identificados en ISO/CEI 9979, y serán identificados por los IDENTIFICADORES DE OBJETO proporcionados en esa Norma Internacional.

9.4 Nombres de contextos de aplicación

9.4.1 Contexto RGT seguro

El nombre del contexto de aplicación, cuando la entidad de aplicación se compone de SMASE, CMISE, ROSE, STASE-ROSE y ACSE, tendrá la siguiente asignación de valor de identificador de objeto:

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

y el siguiente valor de descriptor de objeto: "Secure-TMN-Interactive-Application-Context" (Contexto de aplicación interactiva de RGT seguro).

El campo información de usuario ACSE definido en la Rec. UIT-T X.227 | ISO/CEI 8650-1 consta de una SEQUENCE OF EXTERNAL. La presente Recomendación especifica que, para este contexto de aplicación, el orden de los EXTERNAL en el campo de información de usuario sea:

– datos suministrados para STASE-ROSE, si hay alguno;
– datos suministrados para el CMISE definido en la Rec. UIT-T X.711 | ISO/CEI 9596-1, si hay alguno;
– y datos suministrados para el SMASE definido en la Rec. UIT-T X.701 | ISO/CEI 10040, si hay alguno.

9.4.2 Contexto de aplicación de directorio seguro

El nombre del contexto de aplicación, cuando la entidad de aplicación se compone de directorio X.500, ROSE, STASE-ROSE y ACSE, tendrá la siguiente asignación de valor de identificador de objeto:

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

y el siguiente valor de descriptor de objetos: "Secure–Directory-Application-Context" (Contexto de aplicación de directorio seguro).

9.5 Procedimientos STASE-ROSE

El protocolo STASE-ROSE consta de un solo elemento de procedimiento:

Transferencia.

En las subcláusulas que siguen, se hace un resumen de este elemento de procedimiento. Consta de una presentación sumaria de las APDU pertinentes y de una visión de conjunto de alto nivel de las primitivas del servicio STASE-ROSE, las PDU que intervienen y del servicio de transferencia utilizado.

9.5.1 Transferencia

9.5.1.1 Finalidad

El procedimiento de transferencia es utilizado por un usuario del servicio STASE-ROSE (ROSE) para transferir una APDU de ROSE de manera segura. El STASE-ROSE deberá aplicar las transformaciones de seguridad necesarias a la PDU del ROSE y transferirla al STASE-ROSE par.

9.5.1.2 APDU utilizadas

El procedimiento de transferencia utiliza la APDU del STASE-ROSE.

En el cuadro 9-1 se indican los campos de la APDU del STASE-ROSE (SR-APDU). Sólo deberá utilizarse uno de los campos en una única SR-APDU. En 9.5.1.4 se describe la utilización de los campos de la SR-APDU. Véase en 9.1 la definición ASN.1 de la SR-APDU.

Cuadro 9-1/Q.813 – Campos de APDU de STASE-ROSE

Nombre de campo	Fuente	Sumidero
clear (claro)	pet.	ind.
simpleConfidential (confidencial simple)	pet.	ind.
confidential (confidencial)	pet.	ind.
simplePublicEnciphered (cifrado público simple)	pet.	ind.
publicEnciphered (cifrado público)	pet.	ind.
hashed (troceado)	pet.	ind.
sealed (sellado)	pet.	ind.
signed (firmado)	pet.	ind.
confidentialSigned (firmado confidencial)	pet.	ind.
confidentialMAC (MAC confidencial)	pet.	ind.
confidentialSealed (sellado confidencial)	pet.	ind.

9.5.1.3 Procedimiento de transferencia

Este procedimiento es activado por los siguientes sucesos:

a) una primitiva de petición SR-TRANSFER del solicitante;

b) una primitiva de indicación P-DATA del servicio de presentación.

9.5.1.3.1 Visión de conjunto

En un entorno de sistemas abiertos, cada sistema puede tener su propia representación interna de elementos de información tales como los caracteres, los números enteros y los números reales. Para hacer posible las comunicaciones entre sistemas abiertos heterogéneos, esos ítems de información se pueden especificar utilizando la notación ASN.1 y se puede intercambiar aplicando una sintaxis de transferencia convenida, tal como BER o DER. Si se realiza una ST (por ejemplo, la criptación) en un ítem de información de la capa de aplicación, utilizando la representación interna del sistema (por ejemplo, la representación ASCII de caracteres), el resultado de la transformación carecería de significado para otro sistema abierto que utilizara una representación interna diferente en la capa de aplicación (por ejemplo, la representación EBCDIC de caracteres). Por este motivo, las ST debería llevarse a cabo en una sintaxis de transferencia de ítems de información.

La presente Recomendación especifica que las ST se efectúen en las PDU de ROSE con codificación DER.

9.5.1.3.2 Primitiva de petición SR-TRANSFER

La SRPM solicitante forma una SR-APDU a partir de los valores de los parámetros de la primitiva de petición SR-TRANSFER.

Los campos de la SR-APDU se construyen como sigue:

1) Si el valor del parámetro tipo de criptación (Encryption-Type) (véase 7.4.4.) es **clear**, el parámetro de ROSE-PDU será asignado directamente al campo **clear** de la estructura **SR-APDU.**

2) Para cualquier otro valor del parámetro tipo de criptación se efectúa el siguiente procedimiento:

   – El STASE-ROSE codificará primero el parámetro PDU de ROSE (ROSE-PDU) utilizando las DER.

   • Si se requiere protección de la privacidad de la PDU de ROSE (es decir, el valor del parámetro tipo de criptación es simpleConfidential, confidential, simplePublicEnciphered o publicEnciphered), el STASE-ROSE criptará el tren de octetos con codificación DER y lo asignará a uno de los campos de la estructura **SR-APDU** como sigue:

     – si el valor de parámetro tipo de criptación es **simpleConfidential**, el tren de octetos con codificación DER será criptado utilizando los valores por defecto descritos en 5.2. Los datos criptados son asignados al campo **simpleConfidential**;

     – si el valor del parámetro tipo de criptación es **confidential**, el tren de octetos con codificación DER será criptado utilizando la información proporcionada en el parámetro parámetros de criptación (Encryption-Parameters). Los datos criptados y el parámetro parámetros de criptación serán asignados a los campos **confidential.encrypted** y **confidentail.encryptionParameters**, respectivamente, de la estructura SR-APDU;

     – si el valor del parámetro tipo de criptación es **simplePublicEnciphered**, el tren de octetos con codificación DER será criptado utilizando la información de claves públicas por defecto descrita en 5.2. Los datos criptados son asignados al campo **simplePublicEnciphered**;

     – si el valor del parámetro tipo de criptación es **publicEnciphered**, el tren de octetos con codificación DER será criptado utilizando el parámetro parámetros de criptación. Los datos criptados y el parámetro parámetros de criptación son asignados a los campos **enciphered.publicEncrypted** y **enciphered.encryptionParameters** respectivamente.

   • Si se requiere la comprobación de la integridad (es decir, el valor del parámetro tipo de criptación es simpleHashed, hashed, simpleSealed o sealed), el STASE-ROSE calculará el sello digital o el troceado del tren de datos con codificación DER y lo asignará a los diferentes campos de la estructura **SR-APDU** como sigue:

     – si el valor del parámetro tipo de criptación es **simpleHashed**, el tren de octetos con codificación DER es troceado utilizando los valores por defecto descritos en 5.2. El tren de octetos con codificación DER y su valor troceado son asignados a los campos **hashed.data** y **hashed.hash.simpleHash**, respectivamente, de la estructura **SR-APDU**;

– si el valor del parámetro tipo de criptación es **hashed**, el tren de octetos con codificación DER es troceado utilizando la información proporcionada en el parámetro parámetros de criptación. El tren de octetos con codificación DER, su valor troceado y el parámetro parámetros de criptación son asignados a los campos **hashed.data, hashed.hash.hash.hashValue** y **hashed.hash.hash.encryptionParameters**, respectivamente, de la estructura **SR-APDU**;

– si el valor del parámetro tipo de criptación es **simpleSealed**, el tren de octetos con codificación DER se sella utilizando los valores por defecto descritos en 5.2. El tren de octetos con codificación DER y su valor de sello son asignados a los campos **sealed.data** y **sealed.seal.simpleSeal**, respectivamente, de la estructura **SR-APDU**;

– si el valor de parámetro tipo de criptación es **sealed**, el tren de octetos con codificación DER se sella utilizando la información proporcionada en el parámetro parámetros de criptación. El tren de octetos con codificación DER, el valor de su sello hermético y el parámetro parámetros de criptación son asignados a los campos **sealed.data, sealed.seal.seal.sealValue** y **sealed.seal.seal.encryptionParameters**, respectivamente, de la estructura **SR-APDU**.

• Si se requiere no repudio (es decir, el valor del parámetro tipo de criptación es simpleSignature o signature), el STASE-ROSE calculará la firma digital del tren de octetos con codificación DER y la asignará a la estructura SR-APDU como sigue:

– si el valor del parámetro tipo de criptación es **simpleSigned**, la firma digital del tren de octetos con codificación DER se calcula utilizando los valores por defecto descritos en 5.2. El tren de octetos con codificación DER y su firma son asignados a los campos **signed.data** y **signed.signature.simpleSignature**, respectivamente, de la estructura **SR-APDU**;

– si el valor del parámetro tipo de criptación es **signed**, la firma digital del tren de octetos con codificación DER se calcula utilizando la información proporcionada en el parámetro parámetros de criptación. El tren de octetos con codificación DER, su firma y el parámetro parámetros de criptación son asignados a los campos **signed.data, signed.signature.signature.signatureValue** y **signed.signature.signature.encryptionParameters**, respectivamente, de la estructura **SR-APDU**.

• Si se desea tanto privacidad como no repudio (es decir, el valor del parámetro tipo de criptación es simpleConfidentialSigned o confidentialSigned), el STASE-ROSE calculará la criptación de la PDU con codificación DER y la firma de la PDU con codificación DER. A menos que las entidades comunicantes acuerden otra cosa, por medios que quedan fuera del alcance de la presente Recomendación, la firma se calculará en claro, es decir, no criptada, ROSEpdu con codificación DER. Los resultados de la criptación y la firma serán asignados a la estructura **SR-APDU** como sigue:

– si el valor del parámetro tipo de criptación es igual a **simpleConfidentialSigned**, la firma digital y el valor criptado del tren de octetos con codificación DER se calculan utilizando los valores por defecto descritos en 5.2. Los datos criptados y la firma digital serán asignados a los campos **confidentialSigned.encrypted** y **confidentialSigned.signature.simple.Signature**, respectivamente, de la estructura **SR-APDU**;

– si el valor del parámetro tipo de criptación es **confidentialSigned**, la firma digital y el valor criptado del tren de octetos con codificación DER se calculan utilizando la información proporcionada en el parámetro parámetros de criptación. Los datos

criptados, la firma digital y el parámetro parámetros de criptación serán asignados a los campos **confidentialSigned.encrypted, confidentialSigned.signature.signature.signatureValue** y **confidentialSigned.signature.signature.encryptionParameters**, respectivamente, de la estructura **SR-APDU**.

- Si desea tanto privacidad como integridad (es decir, el valor del parámetro tipo de criptación es simpleConfidentialMAC o confidentialMAC), el STASE-ROSE calculará la criptación de la PDU con codificación DER y el MAC de la PDU con codificación DER. A menos que las entidades comunicantes acuerden otra cosa, por medios que quedan fuera del alcance de la presente Recomendación, el MAC se calculará en claro, es decir, no criptado, ROSEpdu con codificación DER. Los resultados de la criptación y la firma serán asignados a la estructura **SR-APDU** como sigue:
  - si el valor del parámetro tipo de criptación es **simpleConfidentialMAC**, el MAC y el valor criptado del tren de octetos con codificación DER se calculan utilizando los valores por defecto descritos en 5.2. Los datos criptados y el MAC serán asignados a los campos **confidentialMAC.encrypted** y **confidentialMAC.mac.simpleMAC**, respectivamente, de la estructura **SR-APDU**;
  - si el valor del parámetro tipo de criptación es **confidentialMAC**, el MAC y el valor criptado del tren de octetos con codificación DER se calculan utilizando la información proporcionada en el parámetro parámetros de criptación. Los datos criptados, el MAC y el parámetro parámetros de criptación serán asignados a los campos **confidentialMAC.encripted**, **confidentialMAC.mac.mac.hashValue** y **confidentialMAC.mac.mac.encryptionParameters**, respectivamente, de la estructura **SR-APDU**.
- Si se desea tanto privacidad como integridad basada en sello digital (es decir, el valor del parámetro tipo de criptación es simpleConfidentialSealed o confidentialSealed), el STASE-ROSE calculará la criptación de la PDU con codificación DER y el sello de la PDU con codificación DER. A menos que las entidades comunicantes acuerden otra cosa, por medios que quedan fuera del alcance de la presente Recomendación, el sello se calculará en claro, es decir, no criptado, ROSEpdu con codificación DER. Los resultados de la criptación y la firma serán asignados a la estructura **SR-APDU** como sigue:
  - si el valor del parámetro tipo de criptación es **simpleConfidentialSealed**, el sello y el valor criptado del tren de octetos con codificación DER se calculan utilizando los valores por defecto descritos en 5.2. Los datos criptados y el sello serán asignados a los campos **confidentialSealed.encrypted** y **confidentialSealed.seal.simpleSealed**, respectivamente, de la estructura **SR-APDU**;
  - si el valor del parámetro tipo de criptación es **confidentialSealed**, el sello y el valor criptado del tren de octetos con codificación DER se calculan utilizando la información proporcionada en el parámetro parámetros de criptación. Los datos criptados, el sello y el parámetro parámetros de criptación serán asignados a los campos **confidentialSealed.encrypted**, **confidentialSealed.seal.seal.sealValue** y **confidentialSealed.seal.seal.encryptionParameters**, respectivamente, de la estructura **SR-APDU**.

La SR-APDU así formada es transferida al STASE-ROSE par como el parámetro datos de usuario de la primitiva de petición de transferencia P-DATA del servicio de presentación.

La SRPM solicitante espera una primitiva de indicación P-DATA de la capa de presentación o una primitiva de petición SR-TRANSFER del peticionario.

9.5.1.3.3 Primitiva de indicación P-DATA

La SRPM aceptante recibe una SR-APDU de su par como datos de usuario en una primitiva de indicación de transferencia P-DATA. El procedimiento que se indica a continuación será efectuado por el STASE-ROSE para recuperar las primitivas de indicación SR-TRANSFER.

1) Si en la APDU entrante ha sido seleccionado el campo **clear** de la **SR-APDU**, el valor del parámetro tipo de criptación (Encryption-Type) se fijará a **clear** y el parámetro PDU de ROSE (ROSE-PDU) se fijará al campo **clear** en la **SR-APDU**.

2) Si en la APDU entrante se ha seleccionado cualquier otro campo, se lleva a cabo el siguiente procedimiento:

   - si se ha seleccionado el campo **simpleConfidential**, el tren de octetos con codificación DER correspondiente a la PDU del ROSE será recuperado descriptando el campo **simpleConfidential** con los valores por defecto descritos en 5.2. El valor del parámetro tipo de criptación será **simpleConfidential**. El procedimiento a seguir cuando falla la descriptación se describe más adelante en esta subcláusula;
   - si se ha seleccionado el campo **confidential**, el tren de octetos con codificación DER correspondiente a la PDU del ROSE será recuperado descriptando el campo **confidential.encrypted** con el campo **confidential.encryptionParameters**. El valor del parámetro tipo de criptación será **confidential**. El valor del parámetro parámetros de criptación será el valor del campo **confidential.encryptionParameters**. El procedimiento a seguir cuando falla la descriptación se describe más adelante en esta subcláusula;
   - si se ha seleccionado el campo **simplePublicEnciphered**, el tren de octetos con codificación DER correspondiente a la PDU del ROSE será recuperado descriptando el campo **simplePublicEnciphered** con los valores por defecto descritos en 5.2. El valor del parámetro tipo de criptación será **simplePublicEnciphered**. El procedimiento a seguir cuando falla la descriptación se describe más adelante en esta subcláusula;
   - si se ha seleccionado el campo **publicEnciphered**, el tren de octetos con codificación DER correspondiente a la PDU del ROSE será recuperado descriptando el campo **publicEnciphered.publicEncrypted** con el campo **publicEnciphered.encryptionParameters**. El valor del parámetro tipo de criptación será **confidential**. El valor del parámetro parámetros de criptación será el valor del campo **publicEnciphered.encryptionParameters**. El procedimiento a seguir cuando falla la descriptación se describe más adelante en esta subcláusula.
   - Si en la APDU entrante se ha seleccionado el campo **hashed** y:
     - si se ha seleccionado el campo **hashed.hash.simpleHash**, el valor del campo **hashed.data** será utilizado como codificación DER de la PDU del ROSE. El valor troceado del tren de octetos con codificación DER se calculará utilizando los valores por defecto descritos en 5.2 y se comparará con el campo **hashed.hash.simpleHash**. El valor del parámetro tipo de criptación será **simpleHashed**. El procedimiento a seguir cuando falla la comparación del valor troceado se describe más adelante en esta subcláusula;
     - si se ha seleccionado el campo **hashed.hash.hash**, el valor del campo **hashed.data** será utilizado como codificación DER de la PDU del ROSE. El valor troceado del tren de octetos con codificación DER se calculará utilizando el campo **hashed.hash.hash.encryptionParameters** y se comparará con el campo

hashed.hash.hash.hashValue. El valor del parámetro tipo de criptación será **hashed**. El valor del parámetro parámetros de criptación será el valor del campo **hashed.hash.hash.encryptionParameters**. El procedimiento a seguir cuando falla la comparación del valor troceado se describe más adelante en esta subcláusula.

- Si en la APDU entrante se ha seleccionado el campo **sealed** y:
  - si se ha seleccionado el campo **sealed.seal.simpleSeal**, el valor del campo **sealed.data** será utilizado como codificación DER de la PDU del ROSE. El sello digital del tren de octetos con codificación DER se calculará utilizando los valores por defecto descritos en 5.2 y se comparará con el campo **sealed.seal.simpleSeal**. El valor del parámetro tipo de criptación será **simpleSealed**. El procedimiento a seguir cuando falla la comparación del sello digital se describe más adelante en esta subcláusula;
  - si se ha seleccionado el campo **sealed.seal.seal**, el valor del campo **sealed.data** será utilizado como codificación DER de la PDU del ROSE. El sello digital del tren de octetos con codificación DER se calculará utilizando el campo **sealed.seal.seal.encryptionParameters** y se comparará con el campo **sealed.seal.seal.sealValue**. El valor del parámetro tipo de criptación será **sealed**. El valor del parámetro parámetros de criptación será el valor del campo **sealed.seal.seal.encryptionParameters**. El procedimiento a seguir cuando falla la comparación del sello digital se describe más adelante en esta subcláusula.
- Si en la APDU entrante se ha seleccionado el campo **signed** y:
  - si se ha seleccionado el campo **signed.signature.simpleSignature**, el valor del campo **signed.data** será utilizado como codificación DER de la PDU del ROSE. La firma digital del tren de octetos con codificación DER se calculará utilizando los valores por defecto descritos en 5.2 y se comparará con el campo **signed.signature.simpleSignature**. El valor del parámetro tipo de criptación será **simpleSigned**. El procedimiento a seguir cuando falla la comparación de la firma digital se describe más adelante en esta subcláusula;
  - si se ha seleccionado el campo **signed.signature.signature**, el valor del campo **signed.data** será utilizado como codificación DER de la PDU del ROSE. La firma digital del tren de octetos con codificación DER se calculará utilizando el campo **signed.signature.signature.encryptionParameters** y se comparará con el campo **signed.signature.signature.signatureValue**. El valor del parámetro tipo de criptación será **signed**. El valor del parámetro parámetros de criptación será el valor del campo **signed.signature.signature.encryptionParameters**. El procedimiento a seguir cuando falla la comparación de la firma digital se describe más adelante en esta subcláusula.
- Si en la PDU entrante se ha seleccionado el campo **confidentialSigned** y:
  - si se ha seleccionado el campo **confidentialSigned.signature.simpleSignature**, el tren de octetos con codificación DER correspondiente a la PDU del ROSE será recuperado descriptando el campo **confidentialSigned.encrypted** con los valores por defecto descritos en 5.2. La firma digital del tren de octetos con codificación DER se calculará utilizando los valores por defecto y se comparará con el campo **confidentialSigned.signature.simpleSignature**. El valor del parámetro tipo de criptación se fijará a simpleConfidentialSigned. Si falla la descriptación o la comparación de la firma digital se seguirán los procedimientos descritos al final de esta subcláusula;

– si se ha seleccionado el campo **confidentialSigned.signature.signature**, el tren de octetos con codificación DER correspondiente a la PDU del ROSE será recuperado descriptando el campo **confidentialSigned.encrypted** con el campo **confidentialSigned.signature.signature.encryptionParameters**. La firma digital del tren de octetos con codificación DER se calculará utilizando el campo **confidentialSigned.signature.signature.encryptionParameters** y se comparará con el campo **confidentialSigned.signature.signature.signatureValue**. El valor del parámetro tipo de criptación se fijará a **confidentialSigned** y el valor del parámetro parámetros de criptación se fijará a **confidentialSigned.signature.signature.encryptionParameters**. Si falla la descriptación o la comparación de la firma digital se seguirán los procedimientos descritos al final de esta subcláusula.

- Si en la APDU entrante se ha seleccionado el campo **confidentialMAC** y:
  – si se ha seleccionado **confidentialMAC.mac.simpleMAC** el tren de octetos con codificación DER correspondiente a la PDU del ROSE será recuperado descriptando el campo **confidentialMAC.encrypted** con los valores por defecto descritos en 5.2. El MAC del tren de octetos con codificación DER se calculará utilizando los valores por defecto y se comparará con el campo **confidentialMAC.mac.simpleMAC**. El valor del parámetro tipo de criptación se fijará a simpleConfidentialMAC. Si falla la descriptación o la comparación del MAC se seguirán los procedimientos descritos al final de esta subcláusula;
  – si se ha seleccionado el campo **confidentialMAC.mac.mac**, el tren de octetos con codificación DER correspondiente a la PDU del ROSE será recuperado descriptando el campo **confidentialMAC.encrypted** con el campo **confidentialMAC.mac.mac.encryptionParameters**. El MAC del tren de octetos con codificación DER se calculará utilizando el campo **confidentialMAC.mac.mac.encryptionParameters** y se comparará con el campo **confidentialMAC.mac.mac.hashValue**. El valor del parámetro tipo de criptación se fijará a **confidentialMAC** y el valor del parámetro parámetros de criptación se fijará a **confidentialMAC.mac.mac.encryptionParameters**. Si falla la descriptación o la comparación del MAC se seguirán los procedimientos descritos al final de esta subcláusula.
- Si en la APDU entrante se ha seleccionado el campo **confidentialSealed** y:
  – si se ha seleccionado el campo **confidentialSealed.seal.simpleSealed**, el tren de octetos con codificación DER correspondiente a la PDU del ROSE será recuperado descriptando el campo **confidentialSealed.encrypted** con los valores por defecto descritos en 5.2. El sello del tren de octetos con codificación DER se calculará utilizando los valores por defecto y se comparará con el campo **confidentialSealed.seal.simpleSealed**. El valor del parámetro tipo de criptación se fijará a **simpleConfidentialSealed**. Si falla la descriptación o la comparación del sello se seguirán los procedimientos descritos al final de esta subcláusula;
  – si se selecciona el campo **confidentialSealed.seal.seal**, el tren de octetos con codificación DER correspondiente a la PDU del ROSE será recuperado descriptando el campo **confidentialSealed.encrypted** con el campo **confidentialSealed.seal.seal.encrytionParameters**. El sello del tren de octetos con codificación DER se calculará utilizando el campo **confidentialSealed.seal.seal.encryptionParameters** y se comparará con el campo **confidentialSealed.seal.seal.sealValue**. El valor del parámetro tipo de criptación se fijará a **confidentialSealed** y el valor del parámetro parámetros de criptación se

fijará a **confidentialSealed.seal.seal.encryptionParameters**. Si falla la descriptación o la comparación del sello se seguirán los procedimientos descritos al final de esta subcláusula.

- El STASE-ROSE decodificará el tren de octetos con codificación DER recuperado de la APDU entrante y lo asignará al parámetro PDU de ROSE.

Si la SRPM puede efectuar los procedimientos anteriores de manera satisfactoria, emitirá una primitiva indicación SR-TRANSFER hacia el aceptante con los parámetros recuperados.

Los procedimientos que se han de efectuar cuando cualquier valor de la SR-APDU sea inaceptable para la SRPM son un asunto local. Sin embargo, esta Recomendación propone que la implementación de la SRPM ejecute una de las dos acciones indicadas a continuación cuando se reciba una SR-APDU inaceptable.

1) Prescindir de la SR-APDU. La entidad de aplicación local asociada puede ser informada, de una manera que depende de la implementación, de que se ha recibido una APDU inaceptable y de que se ha prescindido de ella.
2) Invocar el servicio A-ABORT proporcionado por el elemento de servicio de control de asociación para abortar la asociación de aplicación. La entidad de aplicación local asociada puede ser informada, de una manera que depende de la implementación, de que se ha recibido una APDU inaceptable y de que la asociación de aplicación ha sido abortada.

Se recomienda que, en cualquiera de los casos el suceso se registre cronológicamente en un registro de auditoría de seguridad y que se emita una alarma de seguridad hacia el administrador de seguridad local.

La SRPM aceptante espera una primitiva de indicación P-DATA de la capa de presentación o una primitiva de petición SR-TRANSFER del usuario del servicio.

9.5.1.4 Utilización de los campos de la SR-APDU

La utilización de los campos de la SR-APDU es como siguen:

1) **clear**: Este campo se utiliza si el usuario SR no pide transformaciones de seguridad.
2) **simpleConfidential**: Este campo se utiliza si el usuario SR pide protección de la privacidad y se utilizan parámetros de criptación por defecto.
3) **confidential**: Este campo se utiliza si el usuario SR pide protección de la privacidad y los parámetros de criptación son proporcionados por el usuario SR.
4) **simplePublicEnciphered**: Este campo se utiliza cuando el usuario SR pide PKCS y los parámetros de criptación no son proporcionados por el usuario SR.
5) **publicEnciphered**: Este campo se utiliza cuando se utiliza PKCS y los parámetros de criptación son proporcionados por el usuario SR.
6) **hashed**: Este campo se utiliza cuando el usuario SR pide protección en basada en la función troceado.
7) **sealed**: Este campo se utiliza cuando el usuario SR pide protección basada en el sello digital.
8) **signed**: Este campo se utiliza cuando el usuario SR pide el no repudio.
9) **confidentialSigned**: Este campo se utiliza cuando el usuario SR pide tanto el no repudio como la protección de la privacidad completa.
10) **confidentialMAC**: Este campo se utiliza cuando el usuario SR pide tanto la integridad basada en la función troceado como la protección de la privacidad.
11) **confidentialSealed**: Este campo se utiliza cuando el usuario SR pide tanto la integridad basada en el sello como la protección de la privacidad completa.

9.6 Correspondencia entre los servicios STASE-ROSE y el servicio de presentación

En esta subcláusula se define la utilización por la SRPM de las primitivas del servicio de presentación descritas en la Recomendación X.216. En el cuadro 9.2 se define la correspondencia entre las primitivas del servicio STASE-ROSE y las APDU y las primitivas del servicio de presentación.

El servicio P-DATA es un servicio no confirmado. La utilización de los parámetros de las primitivas de la petición P-DATA y la indicación P-DATA es como sigue:

- **Datos de usuarios**: La APDU que se va a transferir. Su tamaño máximo no está limitado por esta correspondencia.

Cuadro 9-2/Q.813 – Visión general de la correspondencia del servicio de presentación

Servicio STASE-ROSE	APDU	Servicio de presentación
Petición/indicación SR-TRANSFER	SR-APDU	Petición/indicación P-DATA

10 Correspondencia entre los servicios ROSE y los servicios STASE-ROSE

En esta cláusula se define la utilización de las primitivas de los servicios STASE-ROSE descritas en la presente Recomendación por los servicios ROSE definidos en la Recomendación X.219. La tabla que sigue define la correspondencia:

Cuadro 10-1/Q.813 – Correspondencia entre los servicios ROSE y los servicios STASE-ROSE

Servicio ROSE	APDU	Servicio STASE-ROSE
Petición/indicación RO-INVOKE	ROIV	Petición/indicación SR-TRANSFER
Petición/indicación RO-RESULT	RORS	Petición/indicación SR-TRANSFER
Petición/indicación RO-ERROR	ROER	Petición/indicación SR-TRANSFER
Petición/indicación RO-REJECT-U	RORJ	Petición/indicación SR-TRANSFER
Petición/indicación RO-REJECT-P	RORJ	Petición/indicación SR-TRANSFER

El servicio SR-TRANSFER es un servicio no confirmado.

11 Conformidad

Una implementación que alegue conformidad con esta Recomendación deberá cumplir los requisitos que se indican a continuación:

- **Requisitos de declaración:** El implementador deberá declarar lo siguiente:
  a) el contexto de aplicación con el que se alega conformidad;
  b) si se admite la negociación de los parámetros de seguridad en el momento en que se establece la asociación;
  c) qué algoritmos ST se suministran, si es que se suministra alguno, con la implementación y si la implementación puede utilizar algoritmos ST adicionales proporcionados por el usuario.

- **Requisitos estáticos:** El sistema deberá:

  d) Ser conforme a la definición de la sintaxis abstracta de las APDU formulada en la cláusula 9.

     Admitir las reglas de codificación distinguidas especificadas en la Recomendación X.690 con el identificador de objeto {joint-iso-ccitt asn1(1) ber-derived(2) distinguished-encoding(1)} y el descriptor de objeto "Codificación distinguida de un solo tipo ASN.1" para generar e interpretar información de protocolo de aplicación (por ejemplo, información de protocolo CMISE), y esto, además de la utilización de las BER en la capa de presentación para la codificación de las PDU de STASE-ROSE.

  e) Admitir el protocolo ACSE definido en la Recomendación X.227, para establecer y liberar una asociación.

- **Requisitos dinámicos:** El sistema deberá:

  f) Ser conforme al elemento de procedimiento definido en la cláusula 9.

  g) Ser conforme a las correspondencias con los servicios utilizados con los que se alega conformidad, definidas en las cláusulas 9 y 10.

12 Tablas de estados de la SRPM

En esta cláusula se define una sola máquina de protocolo STASE-ROSE (SRPM) en términos de una tabla de estados. La tabla de estados muestra la interrelación entre el estado de una asociación de aplicación, los sucesos entrantes que se producen en el protocolo, las acciones realizadas y, por último, el estado resultante de la asociación de aplicación.

La tabla de estados de la SRPM no constituye una definición formal de la SRPM. Se incluye para proporcionar una especificación más precisa del procedimiento definido en la cláusula 9.

Esta cláusula contiene los siguientes cuadros:

a) El cuadro 12-1, que especifica el nombre abreviado, fuente, y nombre/descripción de cada suceso entrante. Las fuentes son:
   - usuario STASE-ROSE (usuario SR);
   - ACSE (ACSE);
   - proveedor del servicio de presentación (proveedor PS).

b) El cuadro 12-2, que especifica el nombre abreviado de cada estado de la SRPM.

c) El cuadro 12-3, que especifica el nombre abreviado, objetivo y nombre/descripción de cada suceso entrante. Los objetivos son:
   - usuario STASE-ROSE (usuario SR);
   - ACSE(ACSE);
   - proveedor de servicio presentación (proveedor PS).

d) El cuadro 12-4, que especifica los predicados.

e) El cuadro 12-5, que especifica la tabla de estados de la SRPM utilizando las abreviaturas de las tablas anteriores.

Cuadro 12-1/Q.813 – Lista de sucesos entrantes

Nombre abreviado	Fuente	Nombre y descripción
AA-ESTAB	ACSE	Primitiva de respuesta A-ASSOCIATE o primitiva de confirmación A-ASSOCIATE positiva
SRreq	Usuario SR	Primitiva de petición SR-TRANSFER
APDUua	SRPM par	APDU inaceptable como datos de usuario en un suceso P-DATAind
P-DATAind	Proveedor PS	Primitiva de indicación P-DATA
AA-REL	ACSE	Primitiva de respuesta A-RELEASE positiva o primitiva de confirmación A-RELEASE positiva
ABORTind	ACSE	Primitiva de indicación A-ABORT o primitiva de indicación A-ABORT-P

Cuadro 12-2/Q.813 – Estados de la SRPM

Nombre abreviado	Nombre y descripción
STA01	Reposo; no asociado
STA02	Asociado

Cuadro 12-3/Q.813 – Lista de sucesos salientes

Nombre abreviado	Objetivo	Nombre y descripción
SRind	Usuario SR	Primitiva de indicación SR-TRANSFER
P-DATAreq	Proveedor PS	Primitiva de petición P-DATA
ABORTreq	ACSE	Primitiva de petición A-ABORT

Cuadro 12-4/Q.813 – Predicados

Código	Nombre y descripción
p1	APDU inaceptable y la práctica local es limitar la APDU
p2	APDU inaceptable y la práctica local es abortar la asociación

Cuadro 12-5/Q.813 – Tabla de estados de la SRPM

Sucesos entrantes	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 Convenios

La intersección de un suceso entrante (fila) y un estado (columna) forma una casilla.

En la tabla de estados, una casilla en blanco representa la combinación de un suceso entrante y un estado que no está definido para las SRPM.

Una casilla que no está en blanco representa la combinación de un suceso entrante y un estado que está definido para la SRPM. Esta casilla contiene una o más listas de acciones. Una lista de acciones puede ser obligatoria o condicional. Si una casilla contiene una lista de acciones obligatorias, ésta es la única lista de acciones en la casilla.

Una lista de acciones obligatorias contiene:

a) opcionalmente uno o más sucesos salientes, y

b) un estado resultante.

Una lista de acciones condicionales contiene:

a) una expresión de predicado que comprende predicados y operadores booleanos (Ø representa el booleano NOT), y

b) una lista de acciones obligatorias (esta lista de acciones obligatorias se utiliza solamente si la expresión de predicado es verdadera).

12.2 Acciones que ha de ejecutar la SRPM

La tabla de estados de la SRPM define la acción que ha de ejecutar la SRPM en términos de un suceso saliente opcional y el estado resultante de la asociación de aplicación.

12.2.1 Intersecciones no válidas

Una casilla en blanco indica una intersección no válida de un suceso entrante y un estado. Si se produce esta intersección, se ejecuta una de las acciones siguientes:

a) Si el suceso entrante proviene del usuario SR, cualquier acción ejecutada por la SRPM es un asunto local.

b) Si el suceso entrante está relacionado con una APDU recibida, el proveedor PS o el ACSE, la SRPM emite una ABORTReq hacia el ACSE.

12.2.2 Intersecciones válidas

Si la intersección de un estado y un suceso entrante es válida, se ejecuta una de las acciones siguientes:

a) Si la casilla contiene una lista de acciones obligatorias, la SRPM realiza la acción especificada.

b) Si la casilla contiene una o más listas de acciones condicionales, para cada expresión de predicado que sea verdadera, la SRPM ejecuta la acción especificada. Si ninguna de las expresiones de predicado es verdadera, la SRPM ejecuta una de las acciones especificadas en 12.2.1.

13 Tablas de estados de la máquina de protocolo de operaciones a distancia

Esta cláusula es una ampliación del anexo A "Tablas de estados de la ROPM" a la Recomendación X.229. Se indica aquí la tabla de estados de la parte transferencia de la máquina de protocolo de operaciones a distancia (ROPM-TR, *remote-operations-protocol-machine transfer-part*), si el STASE-ROSE está incluido en el contexto de aplicación y no lo está el RTSE (elemento de servicio de transparencia fiable).

La presente cláusula importa las definiciones, convenios y estados definidos en la Recomendación X.229 (véase en dicha Recomendación una descripción de la información). Contiene los siguientes cuadros:

a) el cuadro 13-1, que especifica los sucesos entrantes recibidos del proveedor del servicio STASE-ROSE (proveedor SR) por el ROSE además de los especificados en el cuadro A.1/X.229;

b) el cuadro 13-2, que especifica los sucesos salientes además de los especificados en el cuadro A.4/X.229;

c) el cuadro 13-3, que especifica la tabla de estados de la ROPM-TR, sí está incluido el STASE-ROSE y no lo está el RTSE en el contexto de aplicación.

Cuadro 13-1/Q.813 – Lista de sucesos entrantes

Nombre abreviado	Origen	Nombre y descripción
SR-TransInd	Proveedor SR	Primitiva de indicación SR-TRANSFER

Cuadro 13-2/Q.813 – Lista de sucesos salientes

Nombre abreviado	Objetivo	Nombre y descripción
SR-TransReq	Proveedor SR	Primitiva de petición SR-TRANSFER

Cuadro 13-3/Q.813 – Tabla de estados de la ROPM-TR para transferencia por el STASE-ROSE

Sucesos entrantes	STA01	STA02
AA-ESTAB	STA200	
TRANSreq		SR-TransReq STA200
SR-TransInd		TRANSind STA200
AA-REL		STA100
AA-ABreq		ABORTreq STA100
ABORTind		AA-Abind STA100

ANEXO A

CMISE seguro

En este anexo se describe la utilización del STASE-ROSE para la implementación de aplicaciones de red gestión de las telecomunicaciones seguras. La figura 2 muestra el modelo para un contexto de aplicación en el que intervienen ACSE, CMISE, ROSE y STASE-ROSE. En el presente anexo se define el contexto de aplicación, las reglas de establecimiento de asociación y la conformidad de un CMISE seguro.

A.1 Contexto de aplicación

El contexto de aplicación que aquí se indica procede de la cláusula 9.

El nombre del contexto de aplicación, cuando la entidad de aplicación se compone de SMASE, CMISE, ROSE, STASE y ACSE, tendrá la siguiente asignación de valor de identificador de objeto:

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

y el siguiente valor de descriptor de objeto: "Contexto de aplicación interactiva de RGT seguro".

A.2 Reglas para el establecimiento de la asociación

La Recomendación X.710 define los parámetros de establecimiento de asociación para el CMISE. La presente Recomendación requiere además que se intercambien los parámetros de asociación definidos en 7.4.2 si se desea negociar los parámetros de seguridad al establecerse la asociación.

Como se especifica en la cláusula 8, el STASE-ROSE proporciona al ACSE cualesquiera valores propuestos para (algunos de) los parámetros de criptación. El mecanismo según el cual el STASE-ROSE informa al ACSE es un asunto que depende de la implementación y no se trata en la presente Recomendación. Esta información se llevará en el campo información de usuario del ACSE utilizando la selección de parámetros de criptación definida en la cláusula 5. Durante la misma fase, el CMISE puede proporcionar también al ACSE información importante para el CMISE par. Toda esa información se lleva en el campo información de usuario ACSE. El campo información de usuario ACSE consta de una SEQUENCE OF EXTERNAL. Esta Recomendación especifica que la información suministrada por el STASE-ROSE, si hay alguna, figurará en el primer EXTERNAL,

seguida por la EXTERNAL para CMISE de alguna, seguida de la EXTERNAL para SMASE, en su caso.

A.3 Conformidad

Un sistema CMISE seguro conforme deberá cumplir los siguientes requisitos:

A.3.1 Requisitos estáticos

a) El sistema deberá cumplir todos los requisito definidos en 8.1/ X.711.

b) El sistema deberá admitir las reglas de codificación distinguida definidas en la Recomendación X.690.

c) El sistema deberá admitir el protocolo STASE-ROSE definido en la cláusula 10.

A.3.2 Requisitos dinámicos

a) El sistema deberá cumplir todos los requisito definidos en 8.2/X.711.

b) El sistema deberá admitir los procedimientos STASE-ROSE definidos en la cláusula 7 y 9.5.

ANEXO B

Sintaxis ASN.1 definida en esta Recomendación

Este anexo reúne las diversas definiciones de la sintaxis ASN.1 proporcionadas en la presente Recomendación.

B.1 Sintaxis abstracta para el autenticador de claves públicas

El siguiente módulo para la autenticación de claves públicas se ha de llevar en el campo valor de autenticación (Authentication-value) de la unidad funcional (FU) autenticación de ACSE cuando se requiera autenticación de entidad par con criptación de clave pública.

```
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

-- EXPORTS everything

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
    -- to be used only if the two communicating entities use GSS-API.
            }

ExplicitAuthenticator ::= SEQUENCE {
                                senderId    [0] SenderId,
                                receiverId  [1] ReceiverId,
                                time        [3] GeneralizedTime,
                                encryptedSymmetricKey [4] INTEGER   OPTIONAL,
    -- a symmetric encryption key encrypted with the receiver's public key
```

```
                    signature       [5] Signature,
    -- the sender's signature of the preceding fields encoded as ASCII characters
                    certificate     [6] SignatureCertificate        OPTIONAL
    -- the sender's public key certificate for the key used for the signature
    }

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

END
```

Esta Recomendación asigna el valor de identificador de objeto ASN.1.

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

como un nombre de sintasis abstracta para el conjunto de todos los valores de datos de presentación cada uno de los cuales es un valor de tipo ASN.1

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

El valor de descriptor de objeto correspondiente es "STASE-ROSE-Authenticator" (autenticador STASE-ROSE).

B.2 Sintaxis abstracta para la negociación de parámetros de seguridad

Se registra el siguiente módulo para la negociación de parámetros de seguridad, a utilizar en el campo información de usuario (UserInformation) de un ACSE.

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

DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS everything

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,
  -- for k-bit output feedback mode or k-bit cipher feedback mode of 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,
```

```
        -- for verification of public keys
        blockSize               [13] INTEGER                            OPTIONAL,
        -- for square mod-n hashing
        keySizes                [14] SET OF INTEGER                     OPTIONAL,
        -- for 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,
        -- symmetric session key, encrypted with Key-Encryption-Key
        encryptedSymmetricKey   [19] INTEGER                            OPTIONAL,
        -- symmetric session key, encrypted with the receiver's public key
        keyEncryptionKey        [20] SEQUENCE (SIZE (1..3)) OF KeyId    OPTIONAL,
        -- one to three symmetric keys used for encrypting a session key
        keyListIds              [21] SET OF KeyListId                   OPTIONAL,
        -- list of encryption keys that can be used during the association
        encryptionCertificate   [22] SET OF EncryptionCertificate       OPTIONAL,
-- X.509 certificates or certification paths of the sender's public keys used for encryption only
        signatureCertificate    [23] SET OF SignatureCertificate        OPTIONAL,
-- X.509 certificates or certification paths of the sender's public keys used for digital signatures only--
        encryptedAuthenticatedSymmetricKeys [24] SET OF
        EncryptedAuthenticatedSymmetricKey                              OPTIONAL,
-- symmetric session key, encrypted with the receiver's public key and signed with sender's key--

        macAlgorithms           [25] SET OF OBJECT IDENTIFIER           OPTIONAL,
        publicKeyCertificate    [26] SET OF PublicKeyCertificate        OPTIONAL,
-- X.509 certificates or certification paths of the sender's public keys with no usage restrictions--
        ...
        }
-- EncryptionParametersSelection is optionally used during association setup to negotiate which algorithms and other
-- encryption parameters will be supported during the association. It is not used in STASE-ROSE PDUs.--

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

END
```

Esta Recomendación asigna el valor de identificador de objeto ASN.1

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

como un nombre de sintaxis abstracta para el conjunto de todos los valores de datos de presentación cada uno de los cuales es un valor de tipo ASN.1

STASE-A-ASSOCIATE-Information.EncryptionParametersSelection

El valor de descriptor de objeto correspondiente es "STASE-ROSE-User-Information" (información de usuario STASE-ROSE).

B.3 Definición de la sintaxis abstracta de las APDU

Además de los tipos ASN.1 definidos en la Recomendación X.229, se definen los tipos siguientes para el STASE-ROSE.

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

```
DEFINITIONS IMPLICIT TAGS ::= BEGIN

-- EXPORTS everything

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)}

-- the referenced module and corresponding syntax are found in Annex 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 represents the DER encoded and encrypted ROSE PDU.
-- encryptionParameters represents the parameters used for encryption.

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

-- SimplePublicEnciphered represents the DER encoded and public key encrypted ROSE PDU.
-- A large PDU may be broken into smaller blocks, each of which may be encrypted
-- as an INTEGER. The size of such blocks depends on the public key encryption algorithm
-- used and on the size of the public key; specification of such block sizes is outside the
-- scope of this Recommendation.
-- In some cases the result of public key encryption may be represented as an OCTET STRING.

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

```
-- publicEncrypted represents the DER encoded and public key encrypted ROSE PDU.
-- encryptionParameters represents the parameters used for encryption.

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

-- hashValue represents the message digest resulting from hashing the DER encoded
-- ROSE PDU.
-- encryptionParameters represents the parameters used for the hashing algorithm.

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

-- data represents the DER encoded ROSE PDU.
-- hash represents the hash value either as a simple OCTET STRING or the Hash
-- structure defined above.

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

-- sealValue represents the seal value for the DER encoded ROSE PDU.
-- encryptionParameters represents the parameters used by the seal generation algorithm.

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

-- data represents the DER encoded ROSE PDU.
-- seal represents the seal value either as a simple OCTET STRING or the Seal structure
-- defined above.

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

-- signatureValue represents the signature for the DER encoded ROSE PDU.
-- encryptionParameters represents the parameters for the signature algorithm.

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

-- data contains the DER encoding of the ROSE PDU.
-- signature represents the signature of the DER encoded ROSE PDU, either as a simple
-- INTEGER or the Signature structure defined above.
```

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

-- encrypted represents the encryption of the DER encoded ROSE PDU.
-- signature represents the signature of the DER encoded ROSE PDU in either a simple form
-- or as Signature type defined above.

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

-- encrypted represents the encryption of the DER encoded ROSE PDU.
-- mac represents the MAC of the DER encoded ROSE PDU in either a simple form
-- or as Hash type defined above.

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

-- encrypted represents the encryption of the DER encoded ROSE PDU.
-- seal represents the seal of the DER encoded ROSE PDU in either a simple form
-- or as Seal type defined above.

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,
 -- for k-bit output feedback mode or k-bit cipher feedback mode of 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,
-- for verification of public keys
       blockSize               [13] INTEGER                             OPTIONAL,
-- for square mod-n hashing
       keySize                 [14] INTEGER                             OPTIONAL,
-- for 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,
-- symmetric session key, encrypted with Key-Encryption-Key
```

```
        encryptedSymmetricKey   [19] INTEGER                                OPTIONAL,
-- symmetric session key, encrypted with the receiver's public key
        keyEncryptionKey        [20] SEQUENCE (SIZE (1..3)) OF KeyId
                                                                            OPTIONAL,
-- one to three symmetric keys used for encrypting a session key
        publicKeyCertificate    [21]  PublicKeyCertificate                  OPTIONAL,
-- X.509 certificate or certification path of the sender's public key with no usage restrictions
        encryptionCertificate   [22] EncryptionCertificate          OPTIONAL,
-- X.509 certificate or certification path of the sender's public key used for encryption only
        signatureCertificate    [23] SignatureCertificate                   OPTIONAL,
-- X.509 certificate or certification path of the sender's public key used for digital signatures only
        encryptedAuthenticatedSymmetricKey
                                [24] EncryptedAuthenticatedSymmetricKey     OPTIONAL,
-- symmetric session key, encrypted with the receiver's public key and signed with sender's key
        macAlgorithm            [25] OBJECT IDENTIFIER                      OPTIONAL,

        ...
        }

-- EncryptionParameters is an extensible type that is used as a catch-all for any
-- parameters that may be used by any of the STs. In most applications only a small
-- number, if any, of the components of EncryptionParameters will be used.

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,
            -- symmetric session key, encrypted with the receiver's public key
                                time                    GeneralizedTime,
                                sender                  SenderId,
                                receiver                ReceiverId,
                                signature               Signature
-- the signature is computed over ASCII representation of the preceding four fields with the sender's private key.
                                }
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 Identificador de objeto de sintaxis abstracta

Esta Recomendación asigna el siguiente identificador de objeto:

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

como un nombre de sintaxis abstracta para el conjunto de valores de datos de presentación, cada uno de los cuales es un valor de tipo ASN.1

Secure-Remote-Operations-APDUs.SR-APDU

donde los componentes de los argumentos de las PDU del ROSE los llena el usuario del ROSE.

El valor del descriptor de objeto correspondiente es "STASE-ROSE-Data" (datos STASE-ROSE).

B.5 Nombres de contextos de aplicación

El nombre del contexto de aplicación, cuando la entidad de aplicación se compone de SMASE, CMISE, ROSE, STASE-ROSE, y ACSE, tendrá la siguiente asignación de valor de identificación de objeto:

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

y el siguiente valor de descriptor de objeto: "Secure-TMN-Interactive-Application-Context" (Contexto de aplicación interactiva de RGT seguro).

El nombre del contexto de aplicación, cuando la entidad de aplicación se compone de directorio X.500, ROSE, STASE-ROSE y ACSE, tendrá la siguiente asignación de valor de identificador de objeto:

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

y el siguiente valor de descriptor de objetos: "Secure–Directory-Application-Context" (Contexto de aplicación de directorio seguro).

APÉNDICE I

Tiempo monótonamente creciente para seguridad

La presente Recomendación especifica la utilización de tiempo (hora) monótonamente creciente para algunos objetivos de seguridad. En este apéndice se describe la posible construcción de un parámetro de tiempo como ese. Se da a efectos ilustrativos únicamente. Son posibles otros procedimientos.

Un reloj de sistema real puede experimentar diversas degradaciones:

- su cadencia puede fluctuar, haciendo que se adelante o se atrase con respecto a la hora real;
- puede dejar de generar señales de tiempo elementales (hacer tic-tac) durante un cierto periodo;
- puede perder la hora real en cuyo caso toma como valor por defecto una determinada hora suficientemente adelantada; esta pérdida de la hora real puede ir o no acompañada de la parada.

En este apéndice se describe la construcción de un reloj, a utilizar por los mecanismos de seguridad, que da un servicio ininterrumpido incluso si se produce algunos de los percances descritos más arriba en el reloj del sistema real.

En el presente apéndice se distingue entre cuatro tipos de tiempo u hora:

1) El UTC o tiempo universal coordinado, que es el tiempo u hora "astronómicamente correcto".

2) El tiempo del SC, que es el tiempo indicado por el reloj del sistema (SC, *system clock*).

3) El tiempo virtual (VT, *virtual time*), que es el tiempo utilizado por los mecanismos de seguridad (y posiblemente por otros componentes del sistema).

4) Tiempo externo (ET, *external time*), que es la hora indicada en una PDU entrante.

El VT se lee cada vez que se genera una PDU saliente que contiene una indicación de tiempo, y cada vez que se recibe una PDU entrante que contiene una indicación de tiempo (también se genera a otros efectos, que quedan fuera del alcance de la presente Recomendación). Cada vez que se lee el VT, primero se actualiza y a continuación se indica el valor actualizado en respuesta a la petición de lectura. El valor actualizado se almacena también en una memoria no volátil. El procedimiento para actualizar el VT viene dado por el siguiente seudocódigo:

si:

$$VT < SC$$

entonces:

$$VT = SC$$

en los demás casos:

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

siendo 1 tic la cantidad más pequeña en la que se puede incrementar el reloj (virtual); debe ser lo bastante pequeña como para que la cadencia máxima posible de los impulsos del reloj (el tic-tac virtual) sea superior a la frecuencia de cresta a la que se lee el VT. Lo normal es que 1 tic sea de 10 ms.

Si el SC se para, o se repone a su valor por defecto, el VT continúa haciendo "tic" con una cadencia "virtual" que corresponde a la frecuencia con que se utiliza. El VT recupera la hora real una vez que el SC se actualiza con el UTC.

Para hacer posible un desfase sustancial (por ejemplo, de 30 minutos) del reloj del sistema (SC) entre reajustes consecutivos del SC con respecto al UTC, los sistemas pueden aceptar unidades de datos de protocolo (PDU) con un ET que difiera del VT en hasta dos veces el desfase permitido (por ejemplo, 1 hora). El valor exacto de este parámetro de permisibilidad se puede ajustar para concordar con las características de los relojes de los sistemas comunicantes. Desde luego, con relojes afectados por una deriva, la detección del retardo se reducirá a una granularidad más gruesa.

En previsión de fallos catastróficos del SC (paradas prolongadas y/o pérdida de la hora real), se puede introducir un parámetro de permisibilidad más amplio (por ejemplo, de 4 horas). Si llegase una PDU con un valor de ET entre los dos parámetros de permisibilidad sería aceptada de todos modos, pero el suceso podría registrarse cronológicamente en un registro de auditoría de seguridad. Si la PDU entrante tiene un ET que está fuera del límite permitido por el segundo parámetro de permisibilidad, podría emitirse una alerta de seguridad además de proceder al registro cronológico del suceso en un registro de auditoría de seguridad; la decisión respecto a si se continúa, se libera o se aborta la asociación en este caso es un asunto que depende de la política de seguridad local.

Todos los registros cronológicos de auditoría de seguridad se pueden hacer con el VT. Así se garantiza el mantenimiento estricto del orden relativo de los sucesos.

Cada vez que el SC se reajusta con el UTC, el suceso puede ser registrado cronológicamente en el registro de auditoría de seguridad. El registro cronológico puede contener los valores del SC y el VT antes y justo después de la actualización. Esta información puede ser de utilidad para establecer la correlación entre el VT y el UTC a efectos de análisis del registro de auditoría de seguridad.

APÉNDICE II

Ejemplo de negociación de algoritmos de seguridad

La figura II.1 ilustra un posible escenario para la negociación de algoritmos de seguridad[5].

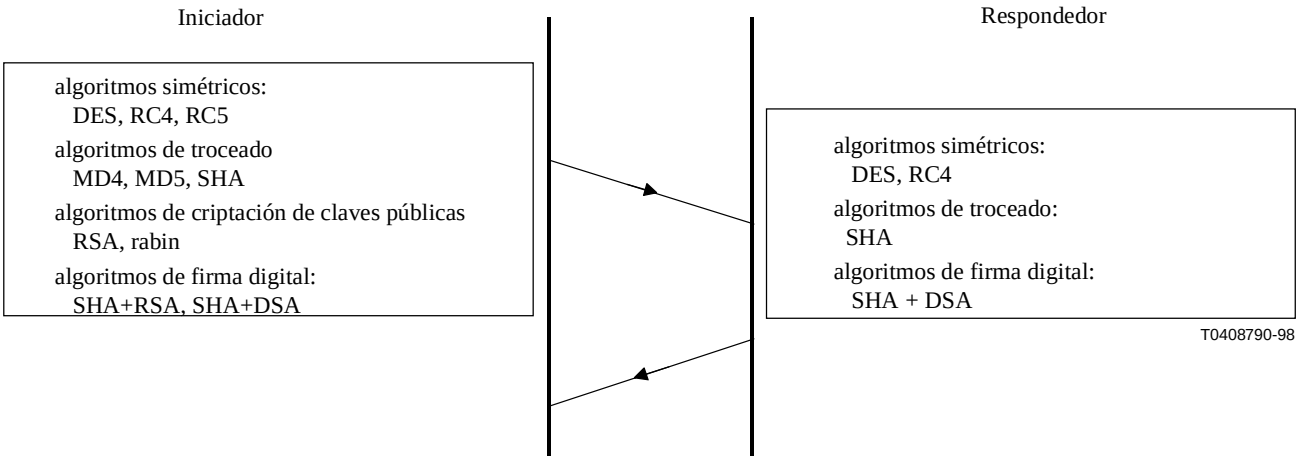

Figura II.1/Q.813 – Negociación de algoritmos de seguridad

Al final del intercambio ilustrado en la figura II-1 el iniciador puede decir que los conjuntos de algoritmos que el respondedor está dispuesto a admitir para la asociación propuesta son inaceptables. En tal caso rechazará la asociación. Si el iniciador está dispuesto a aceptar los conjuntos de algoritmos propuestos por el respondedor, la asociación seguirá su curso con los siguientes algoritmos por defecto:

- algoritmo simétrico: DES, ya que es el algoritmo por defecto especificado en esta Recomendación y se encuentra entre las opciones negociadas para algoritmos simétricos;
- algoritmo de troceado: SHA, ya que es el único algoritmo de troceado mutuamente aceptable;

---

[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, enero de 1979.

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

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

RIVEST (R. L.), The RC5 Encryption Algorithm, *Dr. Dobb's Journal*, Versión 20, N.º 1, págs. 146-148, enero de 1995.

- algoritmo de criptación de claves públicas: no se puede utilizar ningún algoritmo de criptación de claves públicas durante esta asociación ya que no se ha acordado ninguno;
- algoritmo de firma digital: SHA+DSA, ya que es el único algoritmo de firma digital mutuamente aceptable;
- algoritmo de sello digital: MD5+DES, ya que éste es el algoritmo por defecto especificado en esta Recomendación y en la negociación no se hizo referencia a algoritmos de sello digital.

APÉNDICE III

Utilización de GSS-API con STASE-ROSE

La GSS-API es una API de alto nivel para la integración de los servicios de seguridad de las comunicaciones. La versión más reciente de la API (versión 2 de GSS-API) se documenta en RFC 2078.

La utilización de la GSS-API puede reportar diversos beneficios a los vendedores de pilas OSI que desean implementar el STASE-ROSE. En primer lugar, puesto que la GSS-API es una API de alto nivel, proporciona a los implementadores de aplicaciones un medio muy sencillo de integración de los servicios de seguridad. En segundo lugar, con la utilización de la GSS-API con el STASE-ROSE se garantiza la posibilidad de cambiar los algoritmos de seguridad o incluso mecanismos completos de seguridad sin tener que modificar el STASE-ROSE.

En este apéndice se describe cómo puede realizar el STASE-ROSE su funcionalidad criptográfica (transformaciones de seguridad en las PDU de ROSE) mediante la utilización de la interfaz de programación de aplicación de servicios de seguridad genéricos (GSS-API). En este apéndice se describe además la utilización de la GSS-API durante las diferentes fases de la comunicación.

III.1 Fase de establecimiento de la asociación

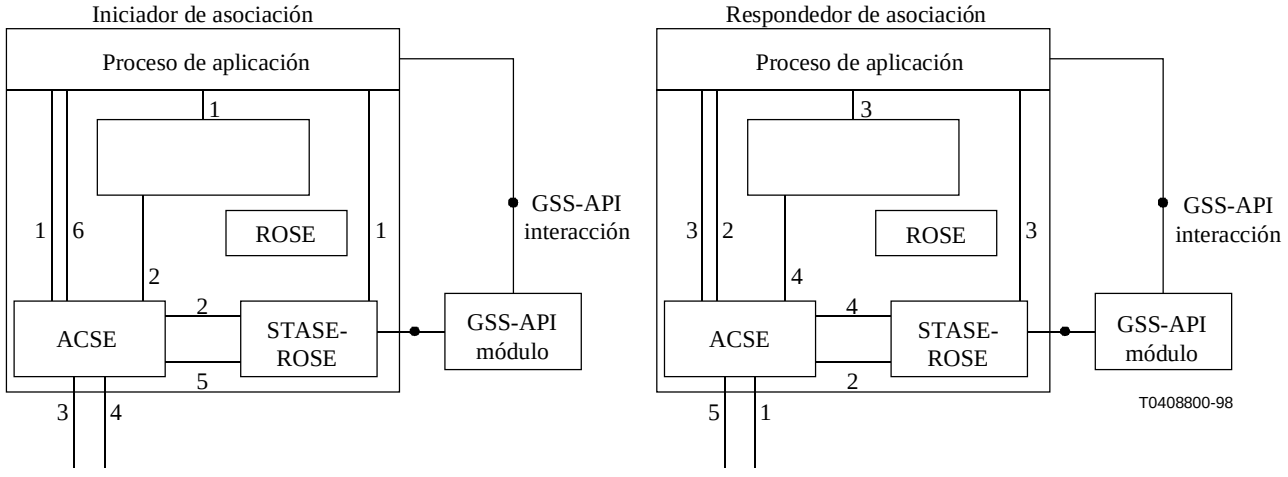

Figura III.1/Q.813 – Utilización de GSS-API con STASE-ROSE en el momento en que se establece la asociación

La figura III-1 se basa en la figura 4. Indica una posible manera de utilizar la GSS-API durante la fase de establecimiento de la asociación y muestra que, tanto el proceso de aplicación como el STASE-ROSE, necesitarán tener acceso al mismo módulo criptográfico GSS-API de soporte. El

proceso de aplicación utilizará el módulo GSS-API a efectos de autenticación durante el establecimiento de la asociación, mientras que el STASE-ROSE utilizará el módulo GSS-API durante la fase de transferencia. A continuación da una descripción de la interacción entre los diferentes componentes de la figura en el lado iniciador (izquierda) y en el lado respondedor (derecha).

En lo que sigue se describe la interacción en el lado iniciador de asociación:

a) La aplicación efectúa GSS_aquire_cred() para adquirir sus credenciales del módulo GSS-API.

b) La aplicación llama GSS_init_sec_context() para iniciar un contexto de seguridad con un respondedor de asociación especificado. La aplicación tiene que decidir respecto a un conjunto de parámetros de seguridad para la asociación (por ejemplo, si se aplica autenticación unilateral o mutua, si hace falta la protección de la secuencia de mensajes y contra la reactuacción). El módulo GSS-API devolverá un testigo initial_context a la aplicación.

c) La aplicación iniciará una petición A-ASSOCIATE de ACSE dirigida al respondedor de la asociación, proporcionando al ACSE el testigo initial_context que se ha de llevar en el campo valor de autenticación. La sintaxis de la estructura del testigo que se propone en 2.1 debe ser sustentada en este caso por el ACSE. Al mismo tiempo, la aplicación puede proporcionar al STASE-ROSE información importante sobre protección de datos durante la fase de transferencia de datos (como mínimo, debería proporcionar el parámetro context_handle de GSS-API como una referencia del contexto de seguridad). Este paso es análogo al paso 1 descrito en 8.1.1.

d) Los pasos restantes son similares a los pasos 2 a 6 de 8.1.1.

En lo que sigue se describe la interacción en el lado respondedor de asociación:

a) Los pasos 1 y 2 son idénticos a los dos primeros pasos de 8.1.2.

b) Cuando la aplicación reciba una indicación A-ASSOCIATE, llamará GSS_aquire_cred() para adquirir sus credenciales del módulo GSS-API (si la aplicación no los ha adquirido todavía). La aplicación llamará continuación a GSS_accept_sec_context() con el testigo recibido del iniciador (dentro del campo valor de autenticación del ACSE) como uno de los parámetros de entrada. El módulo GSS-API autenticará al iniciador verificando que el testigo es válido. Si el iniciador pide autenticación mutua, la aplicación recibirá del módulo GSS-API un segundo testigo que ha de ser transmitido de vuelta al iniciador.

Los pasos 3, 4 y 5 se efectuarán como se describe en 8.1.2. Como parte de la respuesta A-ASSOCIATE que se da en el paso 3, la aplicación proporcionará el segundo testigo (en caso de autenticación mutua) que se ha de llevar en el campo valor de autenticación. De nuevo, la estructura de la sintaxis del testigo propuesta en 2.1 deberá ser soportada por el parámetro valor de autenticación de la respuesta A-ASSOCIATE del ACSE.

Negociación del contexto de seguridad

Como parte del intercambio de testigos de contexto inicial en el momento en el que se establece la asociación, el iniciador y los módulos GSS-API objetivo negociarán (de manera transparente al STASE-ROSE) un conjunto común válido de algoritmos de integridad y confidencialidad para la asociación de seguridad establecida. Por defecto, el conjunto negociado válido de algoritmos será siempre el mayor conjunto común de algoritmos que admitan ambas partes. La negociación de algoritmos la llevan a cabo de manera automática los módulos GSS-API comunicantes sin ser controlados por el usuario GSS-API (aplicación). Este nivel de negociación es suficiente a efectos de interoperabilidad, pero no da a las aplicaciones comunicantes ninguna flexibilidad, por ejemplo, para restringir más aún el número de algoritmos válidos.

Para que la política de seguridad sea más flexible, una opción consistiría, por tanto, en añadir un segundo mecanismo de negociación al nivel del STASE-ROSE (externo a los módulos GSS-API), utilizando los parámetros de negociación definidos en 5.3. En particular, serían pertinentes, los siguientes parámetros de la negociación:

– algoritmo simétrico;
– algoritmo de clave pública;
– algoritmos de firma;
– algoritmos de sello;
– algoritmos de troceado.

La facilidad de negociación expuesta significaría que dos entidades STASE-ROSE pueden ponerse de acuerdo sobre la utilización de un conjunto de algoritmos que es un subconjunto del conjunto de algoritmos que está siendo negociado por los módulos GSS-API. En este caso, un requisito previo consiste en que las entidades STASE-ROSE conozcan qué algoritmos admiten sus módulos GSS-API locales. Los principios de la negociación son los mismos que para el STASE-ROSE en general.

III.2 Fase de transferencia de datos

La figura III.2 se basa en la figura 7. Indica la utilización de la GSS-API durante la fase de transferencia de datos. Las interacciones entre los diferentes componentes en el lado iniciador (izquierda) y el lado respondedor (derecha) serán las mismas que se describen en 8.4.1 y 8.4.2.

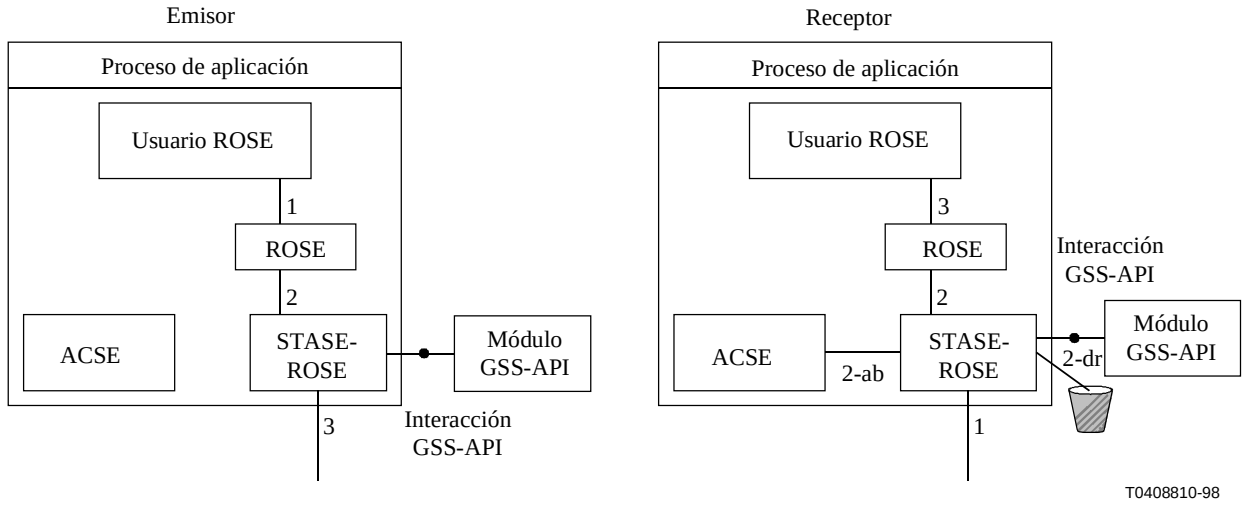

Figura III.2/Q.813 – Utilización de GSS-API con STASE-ROSE durante la transferencia de datos

Durante esta fase, cada ASE STASE-ROSE hará interfaz con su módulo GSS-API local utilizando las primitivas GSS-API GSS_wrap() y GSS_get_mic() para facilitar las transformaciones de seguridad. En este momento del proceso, cada aplicación debe haber proporcionado a su ASE STASE-ROSE local el asa de contexto GSS-API que se necesita como parámetro de entrada en todas las llamadas de la función GSS_wrap()/GSS_get_mic(). Además, los módulos GSS-API comunicantes y los ASE STASE-ROSE (si se utilizó negociación de contexto adicional durante la

fase de establecimiento de la asociación, véase 3.1.1) ya han decidido cuál es el conjunto de algoritmos alternativo para la asociación y, por tanto, las posibilidades de protección de la calidad (QoP, *quality of protection*).

Para cada mensaje enviado, el ROSE pedirá un cierto nivel de protección del STASE-ROSE. No se analiza aquí cómo decide el ROSE cuál es el nivel de protección que se necesita (preferentemente, debería conseguir esta información de la aplicación local).

Como mínimo, el ROSE debe informar al STASE-ROSE del tipo de criptación a utilizar, si es que se utiliza alguna. El STASE-ROSE recibe esta información por medio del parámetro tipo de criptación de SR-TRANSFER. Se ha de establecer la correspondencia entre estas peticiones y una llamada de la función gss_wrap()/gss_get_mic(). Cada vez que se pida algún tipo de protección de la confidencialidad (con o sin protección de la integridad), el STASE-ROSE debe utilizar la función gss_wrap(). Cada vez que se pida protección de la integridad o del no repudio sin protección alguna de la confidencialidad, el STASE-ROSE debe utilizar la función gss_get_mic().

Ahora bien, para la entidad que utiliza el STASE-ROSE no es suficiente saber si ha de utilizar gss_wrap() o gss_get_mic(). La entidad del STASE-ROSE iniciadora debe saber también qué nivel de calidad de protección ha de pedir cuando llame estas funciones [parámetro de entrada qop_req para gss_wrap() y gss_get_mic()]. El STASE-ROSE puede en principio decidir sobre el nivel de calidad de protección de dos maneras:

1) El STASE-ROSE es informado sobre la calidad de protección requerida por el ROSE mediante el parámetro parámetros de criptación de SR-TRANSFER (véase 7.4.4 de la especificación del STASE-ROSE).

2) Si no se utiliza el parámetro parámetros de criptación de SR-TRANSFER, se utilizará el nivel de protección por defecto para el tipo de criptación indicado por SR-TRANSFER. En este caso se supone que las entidades del STASE-ROSE ya han acordado un nivel de protección por defecto en el momento del establecimiento del contexto.

Cualquiera que sea la solución utilizada, es importante asegurar que el STASE-ROSE selecciona un nivel de QoP que se encuentra dentro de los límites de protección que se negociaron en el momento en que se estableció la asociación. El implementador del STASE-ROSE ha de decidir por tanto lo que debería ocurrir si, por ejemplo, la primitiva SR-TRANSFER procedente del ROSE pidiera un nivel de calidad de protección superior al que el contexto establecido puede o está autorizado a proporcionar.

Una vez creado un testigo de gss_wrap() o de gss_get_mic(), se insertará en el protocolo STASE-ROSE utilizando el campo del protocolo gssToken definido en 2.3. La entidad receptora verificará los testigos llamando gss_unwrap() o gss_verify_mic().

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Transmisiones de señales radiofónicas, de televisión y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	**Conmutación y señalización**
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información
Serie Z	Lenguajes de programación