UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T Q.815

(02/2000)

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

SERIE Q: CONMUTACIÓN Y SEÑALIZACIÓN

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

Especificación de un módulo de seguridad para la protección del mensaje completo

Recomendación UIT-T Q.815

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES UIT-T DE LA SERIE Q

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
RED MÓVIL TERRESTRE PÚBLICA	Q.1000–Q.1099
INTERFUNCIONAMIENTO CON SISTEMAS MÓVILES POR SATÉLITE	Q.1100–Q.1199
RED INTELIGENTE	Q.1200–Q.1699
REQUISITOS Y PROTOCOLOS DE SEÑALIZACIÓN PARA IMT-2000	Q.1700–Q.1799
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.815

Especificación de un módulo de seguridad para la protección del mensaje completo

Resumen

La presente Recomendación UIT-T especifica un módulo de seguridad opcional utilizable con la Recomendación Q.814, Especificación de un agente interactivo de intercambio electrónico de datos, que proporciona servicios de seguridad a unidades de datos de protocolo (PDU) completas. En particular, el módulo de seguridad sustenta no repudio de origen y de recibo, así como integridad del mensaje completo.

Orígenes

La Recomendación UIT-T Q.815, preparada por la Comisión de Estudio 4 (1997-2000) del UIT-T, fue aprobada por el procedimiento de la Resolución 1 de la CMNT el 4 de febrero de 2000.

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 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

Recomendación UIT-T Q.815

Especificación de un módulo de seguridad para la protección del mensaje completo

1 Alcance

El módulo de seguridad proporciona servicios de seguridad a unidades de datos de protocolo (PDU) completas. En particular, el módulo de seguridad soporta no repudio de origen y de recibo, así como integridad del mensaje completo.

En el contexto de las transacciones RGT por EDI y, en el lado emisor, acepta como entrada la salida del traductor EDI, efectúa las transformaciones de seguridad solicitadas, y proporciona la cadena de octetos resultante al agente interactivo (IA).

En el lado receptor, recibe del IA una cadena de octetos que interpreta como PDU del módulo de seguridad. Procede luego a verificar la validez del mensaje subyacente. En el caso de protección de la integridad, si el mensaje es válido, pasa entonces ese mensaje (sin el código de integridad del mensaje) al traductor EDI.

El módulo de seguridad mantiene un registro cronológico de todos los mensajes que recibe del IA. Proporciona esos mensajes, junto con una indicación por cada mensaje, que representa el resultado del procedimiento de verificación. Este registro cronológico está a disposición del usuario EDI local.

Las características de registro cronológico, así como la interfaz con ese registro, son un asunto local que cae fuera del alcance de esta Recomendación UIT-T. El comportamiento del módulo de seguridad, cuando falla la verificación, es también un asunto de carácter local que también está fuera del alcance de esta Recomendación UIT-T.

La figura 1 que sigue, idéntica a la figura 2/Q.814, se utiliza aquí como referencia.

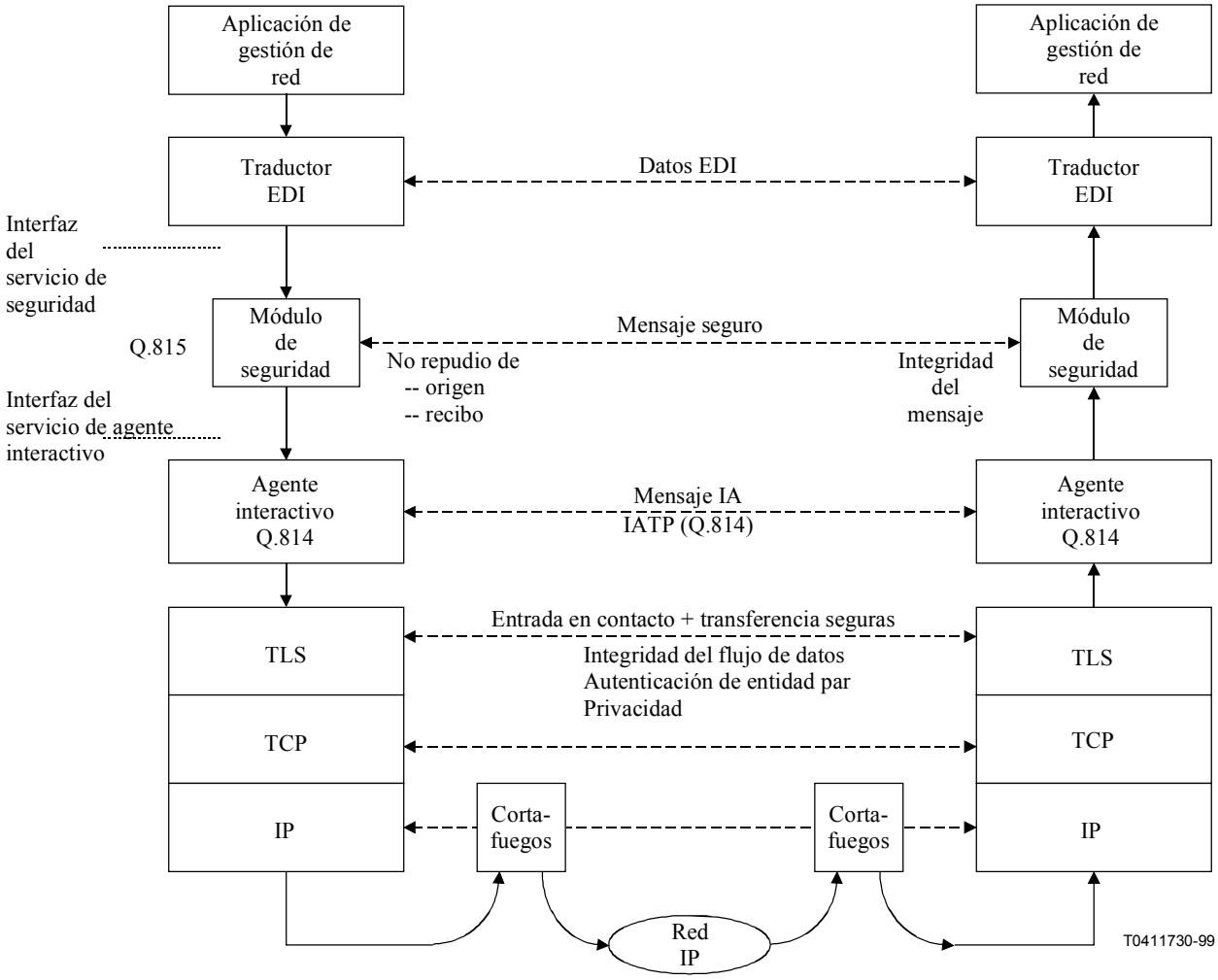

Figura 1/Q.815 – Relación de los flujos de mensajes con servicios de seguridad del mensaje

2 Referencias

2.1 Referencias normativas

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.

– Recomendación UIT-T Q.812 (1997)/enm.3 (2000), *Perfiles de protocolo de capa superior para la interfaz Q.3 – Enmienda 3: Perfil de protocolos para el agente interactivo de comunicaciones electrónicas*.

– Recomendación UIT-T Q.814 (2000), *Especificación de un agente interactivo de intercambio electrónico de datos*.

– Recomendación UIT-T X.509 (1997) | ISO/CEI 9495-8:1998, *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 especificaciones de 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 de sintaxis abstracta uno: Especificación de las reglas de codificación básicas, de las reglas de codificación canónica y de las reglas de codificación distinguida*.

ISO – International Organization for Standardization:

– ISO 3166: (Todas las partes), *Codes for the representation of names of countries and their subdivisions.*

2.2 Referencias informativas

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

3 Definiciones

En esta Recomendación UIT-T se definen los siguientes términos.

3.1 unidad de datos de protocolo de aplicación: Paquete de datos intercambiados entre dos programas de aplicación a través de una red. Ésta es la vista del nivel más alto de la comunicación en el modelo de siete capas de OSI y un solo paquete intercambiado en este nivel puede en realidad ser transmitido como varios paquetes a una capa inferior, al igual que puede añadírsele información complementaria (encabezamientos) para encaminamiento, etc.

3.2 reglas de codificación distinguida: Forma restringida de las reglas de codificación básica definidas en la (Recomendación UIT-T X.690) para eliminar las opciones en la BER.

3.3 intercambio electrónico de datos: Intercambio de documentos en forma electrónica normalizada, entre organizaciones, de manera automatizada, directamente desde una aplicación informática de una organización a una aplicación de otra.

3.4 intercambio electrónico de datos para administración, comercio y transporte: La sintaxis es una norma ISO (ISO 9735) y fue adoptada por las Naciones Unidas como base para el desarrollo internacional de mensajes comerciales de EDI (UN/EDIFACT). EDIFACT nació del deseo de agrupar normas anteriores y ASC X12.

3.5 agente interactivo: El agente interactivo (IA) soporta el intercambio de transacciones de intercambio electrónico de datos (ANSI ASC X12 EDI o EDIFACT) entre entidades pares dentro de la industria de telecomunicaciones.

3.6 RSA: RSA es un criptosistema de claves públicas desarrollado por Ronald L. Rivest, Adi Shamir y Leonard M. Adleman en 1977 en un esfuerzo por contribuir a garantizar la seguridad de Internet. Un criptosistema es simplemente un algoritmo que puede convertir datos de entrada en algo irreconocible (encriptación) y convertir los datos irreconocibles a su forma original (descriptación). En la Recomendación UIT-T X.509 se describen técnicas de encriptación RSA.

3.7 algoritmo de troceo seguro, revisión 1: Función de troceo de 160 bits, encomendada por el National Institute for Standards Technology (NIST) con mecanismos de seguridad similares al MD5. SHA-1 lo ha definido el gobierno de Estados Unidos en FIPS 180-1. Es un mecanismo para reducir un mensaje de texto extenso en un corto digesto de 160 bits que es al mismo tiempo unidireccional (es decir, no reversible) y no susceptible a colisiones debidas a múltiples textos diferentes. Como el SHA genera un troceo de 160 bits (resumen del mensaje), está mucho más a salvo de los ataques criptográficos a viva fuerza que el MD5.

Los digestos están mejor pensados como distintivo digital de un mensaje. Se trata de un algoritmo relativamente rápido, de baja tara y seguro. El SHA-1 se puede utilizar para soportar la protección de seguridad (por si mismo), o para el no rechazo (junto con la encriptación de una clave pública). En el apéndice II se describe el algoritmo de resumen de mensaje del SHA-1.

4 Abreviaturas

En esta Recomendación UIT-T se utilizan las siglas siguientes.

APDU Unidad de datos de protocolo de aplicación (*application protocol data unit*)

ASC Comité de normalización acreditado (*accredited standards committee*)

DER Reglas de codificación distinguida (de la ASN.1) (*distinguished encoding rules)*

EDI Intercambio electrónico de datos (*electronic data interchange*)

EDIFACT Intercambio electrónico de datos para administración, comercio y transporte (*electronic data interchange for administration, commerce and transport)*

IA Agente interactivo (*interactive agent*)

IP Protocolo Internet (*Internet protocol*)

PDU Unidad de datos de protocolo (*protocol data unit*)

RGT Red de gestión de las telecomunicaciones

RSA Rivest, Shamir, Aldeman

SHA-1 Algoritmo de troceo seguro, Revisión 1 (*secure hash algorithm, revision 1*)

SM Módulo de seguridad (*security module*)

SR STASE ROSE

TCP Protocolo de control de transmisión (*transmission control protocol*)

TLS Seguridad de la capa de transporte (*transport layer security*)

5 Convenios

Se aplican los siguientes convenios: las referencias a cláusulas, subcláusulas, anexos y apéndices se refieren a dichos elementos de la presente Recomendación UIT-T, a menos que se incluya explícitamente otra especificación.

6 Detalles de la Recomendación Q.815

6.1 Tipos de mensajes de módulo de seguridad

El módulo de seguridad Q.815 especifica las siguientes directrices:

Los servicios de integridad y/o no repudio del mensaje pueden ser proporcionados por el módulo de seguridad (SM). Si el SM proporciona integridad del mensaje, utiliza el SHA-1 para producir el

resumen del mensaje (MD, *message digest*). Si el SM proporciona no repudio del origen, utiliza el SHA-1 para producir un MD que utilice el mecanismo de firma digital de RSA. El no repudio de recibo es proporcionado por un mecanismo descrito en la presente Recomendación UIT-T. Está disponible un cuarto conjunto de mejoras de seguridad mediante el uso de STASE-ROSE (véase la Recomendación UIT-T Q.813).

6.1.1 Servicio de integridad del mensaje

El usuario directo suministra al SM un mensaje EDIFACT o ASC X12 EDI. El SM calculará un digesto del mensaje utilizando los datos EDIFACT o ASC X12 EDI como entrada al algoritmo del digesto. Las DER del SM codifican un mensaje troceado de acuerdo con 7.1. El IA utiliza la cadena de octetos con codificación DER como contenido de un mensaje mejorado, que se define en la Recomendación UIT-T Q.814.

6.1.2 No repudio del servicio de origen

El usuario directo suministra al SM un mensaje EDIFACT o ASC X12 EDI. El SM calculará un digesto del mensaje utilizando los datos EDIFACT o ASC X12 EDI como entrada al algoritmo del digesto. El SM encripta una codificación DER del digesto del mensaje de acuerdo con el algoritmo de firma digital utilizando la clave privada del emisor. Las DER del SM codifican un mensaje firmado de acuerdo con 7.2. El IA utiliza la cadena de octetos con codificación DER como contenido de un mensaje mejorado, que se define en la Recomendación UIT-T Q.814.

6.1.3 No repudio del servicio de recibo

El usuario directo suministra al SM un identificador de mensaje único y la fecha y hora en que se recibió el mensaje correspondiente. Véase en 7.3.2 información relativa al identificador único y a la fecha/hora. Opcionalmente, puede requerirse del usuario directo, mediante acuerdo bilateral, que suministre un digesto del mensaje o una firma digital como componente del recibo. El digesto o la firma deben corresponder al mensaje unívocamente identificado antes descrito. Las DER del SM codifican un mensaje recibido por el IA de acuerdo con 7.3. El IA utiliza la cadena de octetos con codificación DER como contenido de un mensaje mejorado, que se define en la Recomendación Q.814.

6.2 Características generales

Los digestos de mensajes y las firmas digitales se calculan en el texto claro (no encriptado) de los mensajes EDIFACT o ASC X12 EDI/cadena general.

Los certificados digitales serán compatibles con la Recomendación UIT-T X.509 versión 3.

7 Sintaxis general

La sintaxis general del mensaje del módulo de seguridad es la siguiente:

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

NOTA – En el anexo A puede verse un módulo ASN.1 que contiene la sintaxis general y cada una de las sintaxis específicas definidas en esta cláusula.

7.1 Mensajes troceados (*Hashed Messages*)

Los servicios de integridad del mensaje son proporcionados por el *HashedMessage* definido como:

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

7.1.1 Identificadores de objeto referenciados por mensajes troceados

En el hashedMessage el valor a utilizar para el hashAlgorithmIdentifier es:

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

7.1.2 Información de valor de los mensajes troceados

hashedContent consiste en un mensaje EDIFACT, ASC X12 EDI o de otro mensaje en cadena.

messageDigest consiste en el digesto SHA-1 del hashedContent.

7.2 Mensajes firmados

Los servicios de no repudio de origen son proporcionados por el *SignedMessage* definido como:

```
SignedMessage :: = SEQUENCE {
        signedVersion                   Version DEFAULT v1999 ,
        signedDigestAlgorithms          SET OF AlgorithmIdentifier,
        signedContent                   SignedContent,      -- Data
        signerInfos                     SET OF SEQUENCE {
                signerVersion           Version DEFAULT v1999 ,
                issuerAndSerialNumber   SEQUENCE {
                        issuerCountry   SEQUENCE OF SET OF SEQUENCE {
                                country         OBJECT IDENTIFIER,
                                countryValue    PrintableString
                        },
                        issuerOrg       SEQUENCE OF SET OF SEQUENCE {
                                organizationName  OBJECT IDENTIFIER,
                                organizationValue PrintableString
                        },
                        serialNumber    INTEGER
                },
                signedDigestAlgorithm   AlgorithmIdentifier,
                digestEncryptionAlgorithm AlgorithmIdentifier,
                encryptedDigest         OCTET STRING
        }
}
SignedContent :: =      CHOICE {
signedContent1          GeneralString,
signedContent2          IA5String
}
```

7.2.1 Identificadores de objetos referenciados por mensajes firmados

En el signedMessage el valor a utilizar para los signedDigestAlgorithms es:

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

En el SignedMessage el valor a utilizar para los signedDigestAlgorithm es:

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

En el SignedMessage el valor a utilizar para los digestEncryptionAlgorithmIdentifier es:

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

7.2.2 Información de valor para mensajes firmados

signedContent consiste en un mensaje EDIFACT, ASC X12 EDI, o de otro mansaje en cadena.

countryValue es la representación en dos caracteres de un nombre del país del emisor que figura en ISO 3166.

organizationValue es el nombre de la organización del emisor de certificado.

serialNumber es un número único asignado al certificado por el emisor.

encryptedDigest consiste en una codificación del resumen SHA-1 del signedContent encriptado en RSA con la clave privada de la parte emitente.

signedDigestAlgorithms – el conjunto está compuesto por un único miembro algorithmIdentifier del algoritmo del digesto del mensaje. El algoritmo será el mismo en signedDigestAlgorithms y en signedDigestAlgorithm.

signerInfos – el conjunto lo compondrá un solo miembro.

El par clave privada/pública asociado con el certificado digital utilizado para autenticación ha de ser utilizado para firmar/verificar.

issuerCountry – el conjunto lo compondrá un solo miembro.

Organización del emisor (issuerOrg) – el conjunto lo compondrá un solo miembro.

NOTA – El algoritmo de firma digital RSA convierte el tipo de datos ENTERO de una firma digital en una CADENA DE OCTETOS.

7.3 Mensaje de recibo IA

Los servicios de no repudio de recibo son proporcionados por el *IaReceiptMessage* definido como:

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

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

7.3.1 Identificadores de objetos referenciados por mensajes de recibo IA

En el IaReceiptMessage el valor a utilizar para el receiptDigestAlgorithm es:

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

En el IaReceiptMessage, el valor a utilizar para el receiptSignatureAlgorithm es:

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

7.3.2 Información de valor para mensajes de recibo IA

uniqueIdentifier consiste en cualquier campo de los datos del mensaje que sea único de un mensaje al siguiente (por ejemplo, segmento ISA ASC X12 EDI). Las entidades pares determinarán el campo de datos a utilizar para este fin mediante un mecanismo que cae fuera del alcance de esta Recomendación UIT-T.

dateTimeStamp tiene el formato siguiente: CCYYMMDDhhmmssz

donde: CC = siglo
YY= año
MM = mes
DD = día
hh = hora
mm = minuto
ss = segundo
z = indicador de zona horaria alfa.

Un carácter en blanco en el indicador de zona horaria indica hora observada local. La fecha y hora de la indicación de hora debe especificar la hora en que la parte recibiente recibió el mensaje completo de recibo. El uso de la hora universal coordinada (Z) en el recibo se recomienda para evitar ambigüedades de zonas horarias.

El receiptMessageDigest es el digesto del mensaje de recibo. Si el mensaje era del formato SimpleHashed, este digesto es el que se recibió con el mensaje y fue verificado por el recibiente. Si el mensaje original era de cualquier otro tipo, el digesto necesitará ser calculado por el recibiente utilizando el mensaje EDIFACT/ASC X12 EDI/cadena general como entrada al algoritmo del digesto del mensaje especificado.

receiptDigitalSignature se calcula formateando una concatenación del uniqueIdentifier del mensaje EDIFACT/ASC X12 EDI/cadena general, la dateTimeStamp de recibo (15 octetos) y la firma digital recibida con el mensaje original. Esta estructura de octetos es entonces firmada criptándola de acuerdo con el algoritmo de firma digital RSA utilizando SHA-1 como algoritmo del digesto y la clave privada de la parte generadora del recibo. La parte que recibe el recibo debe construir una concatenación del: uniqueIdentifier (del mensaje transmitido o del cuerpo del recibo), la dateTimeStamp de 15 octetos (del cuerpo del recibo) y el encryptedDigest (firma digital) del mensaje original. Esta estructura concatenada, junto con la firma digital del propio mensaje de recibo (receiptDigitalSignature) debe pasarse al algoritmo de verificación de la firma digital. La clave pública de la parte receptante se utiliza para verificar la firma. Este tipo de recibo requiere que el mensaje original del que se acusa recibo deba haber sido del tipo SignedMessage.

7.4 PDU Stase-Rose

La sintaxis de este tipo de mensaje es importada de la Recomendación UIT-T Q.813.

ANEXO A

Módulo de producción ASN.1

```
DEFINITIONS IMPLICIT TAGS :: = BEGIN

-- EXPORTS everything

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

-- Useful Types

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

Version :: = INTEGER
v1999 Version :: = 0

-- General Syntax

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

-- Hashed Message Syntax

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

-- Signed Message Syntax

SignedMessage :: =                  SEQUENCE {
        signedVersion               Version DEFAULT v1999 ,
        signedDigestAlgorithms      SET OF AlgorithmIdentifier,
        signedContent               SignedContent,          -- Data
        signerInfos                 SET OF SEQUENCE {
                signerVersion       Version DEFAULT v1999 ,
```

```
            issuerAndSerialNumber   SEQUENCE {
                  issuerCountry     SEQUENCE OF SET OF SEQUENCE {
                    country         OBJECT IDENTIFIER,
              countryValue          PrintableString
            },
            issuerOrg               SEQUENCE OF SET OF SEQUENCE {
              organizationName      OBJECT IDENTIFIER,
              organizationValue     PrintableString
            },
            serialNumber            INTEGER
      },
      signedDigestAlgorithm         AlgorithmIdentifier,
      digestEncryptionAlgorithm     AlgorithmIdentifier,
      encryptedDigest               OCTET STRING
  }
}

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

-- Receipt Message Syntax

IaReceiptMessage :: =     SEQUENCE {
      uniqueIdentifier    OCTET STRING, -- A unique identifier within the message
      dateTimeStamp       PrintableString ( SIZE(15) ),
      enhancements        Enhancements OPTIONAL
}

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

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

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

END
```

APÉNDICE I

Referencias no normativas

– ISO 9735:1998, *Electronic Data Interchange For Administration, Commerce and Transport* (*EDIFACT*) – *Application level syntax rules*.

– ANSI ASC X12: American National Standards Institute (ANSI) Accredited Standards Committee X12. The Committee was chartered by ANSI in 1979 to develop uniform standards for the electronic interchange of business documents.

APÉNDICE II

Algoritmo del digesto de mensaje del SHA-1

II.1 Introducción

Para un mensaje de longitud < 2^{64} bits, el SHA-1 produce una representación condensada de 160 bits del mensaje llamada digesto (resumen muy condensado) del mensaje. El digesto del mensaje se utiliza durante la generación de una firma para el mensaje. El SHA-1 se utiliza también para calcular un digesto de la versión recibida del mensaje durante el proceso de verificación de la firma. Cualquier cambio introducido en el mensaje en tránsito dará lugar, con mucha probabilidad, a un digesto de mensaje diferente y fallará la verificación de la firma.

El SHA-1 se ha concebido de manera que tenga la propiedad siguiente: inviabilidad de encontrar, mediante computación, el mensaje que corresponde a un digesto de mensaje dado o de encontrar dos mensajes diferentes que produzcan el mismo digesto de mensaje.

II.2 Cadenas de bits y enteros

Se utilizará la terminología siguiente en relación con las cadenas de bits y los enteros:

a) Un dígito hexadecimal es un elemento del conjunto {0, 1, ..., 9, A, ..., F}. Un dígito hexadecimal es la representación de una cadena de 4 bits.

Ejemplos: 7 = 0111, A = 1010.

b) Una palabra es igual a una cadena de 32 bits que puede representarse como una secuencia de 8 dígitos hexadecimales. Para convertir una palabra en 8 dígitos hexadecimales, se convierte cada cadena de 4 bits en su equivalente hexadecimal como se ha indicado en a).

Ejemplo: 1010 0001 0000 0011 1111 1110 0010 0011 = A103FE23.

c) Un entero entre 0 y $2^{32} – 1$ inclusive se puede representar como una palabra. Los cuatro bits menos significativos del entero representan mediante el dígito hexadecimal situado más a la derecha en la representación de la palabra.

Ejemplo: El entero $291 = 2^8+2^5+2^1+2^0 = 256+32+2+1$ se representa mediante la palabra hexadecimal 00000123.

Si z es un entero, $0 \leq z < 2^{64}$, entonces $z = 2^{32} x + y$ donde $0 \leq x < 2^{32}$ y $0 \leq y < 2^{32}$. Puesto que x e y se pueden representar como las palabras X e Y, respectivamente, z se puede representar como el par de palabras (X, Y).

d) Bloque = cadena de 512 bits. Un bloque (por ejemplo, B) se puede representar como una secuencia de 16 palabras.

II.3 Operaciones con palabras

A las palabras se les aplicarán los operadores lógicos siguientes:

a) Operaciones con palabras lógicas binarias:

X ^ Y = "and" (y) lógica binaria de X e Y.

X ∨ Y = "inclusive-or" (o inclusivo) lógica binaria de X e Y.

X XOR Y = "exclusive-or" (o exclusivo) lógica binaria de X e Y.

~X = "complement" (complemento) lógica binaria de X.

Ejemplo:

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

b) La operación X + Y se define como sigue: las palabras X e Y representan los enteros x e y, donde $0 \leq x < 2^{32}$ y $0 \leq y < 2^{32}$. Para los enteros positivos n y m, sea n módulo m el resto de dividir n por m. Calcular $z = (x + y)$ módulo 2^{32}.

Entonces $0 \leq z < 2^{32}$. Convertir z en una palabra, Z, y definir Z = X + Y.

c) La operación de desplazamiento circular a la izquierda $S^n(X)$, donde X es una palabra y n es un entero con $0 \leq n^{32}$, se define de la siguiente manera: $S^n(X) = (X << n)$ OR $(X >> 32–n)$.

En lo anterior, X << n se obtiene como sigue: se descartan los n bits de X situados más a la izquierda y se rellena a continuación el resultado con n ceros a la derecha (el resultado sigue siendo de 32 bits). X >> n se obtiene descartando los n bits de X situados más a la derecha y rellenando a continuación el resultado con n ceros a la izquierda. De esta manera, $S^n(X)$ es equivalente a un desplazamiento circular de X en n posiciones a la izquierda.

II.4 Relleno del mensaje

El SHA-1 se utiliza para calcular un digesto de mensaje de un mensaje o fichero de datos proporcionado a modo de entrada. El mensaje o el fichero de datos deberán considerarse como una cadena de bits. La longitud del mensaje es el número de bits del mismo (el mensaje vacío tiene una longitud 0). Si el número de bits de un mensaje es un múltiplo de 8, se puede representar el mensaje en hexadecimal para una mayor compresión. El relleno del mensaje tiene por objeto hacer que la longitud total de un mensaje rellenado sea un múltiplo de 512. El SHA-1 procesa de manera secuencial los bloques de 512 bits cuando calcula el digesto de un mensaje. En lo que sigue se especifica cómo deberá llevarse a cabo la operación de relleno. De forma breve, se agrega un "1" (uno) seguido de m "0" (ceros) seguidos de un entero de 64 bits al final del mensaje para producir un mensaje rellenado de 512*n de longitud. El entero de 64 bits es l, la longitud del mensaje original. El mensaje rellenado es procesado a continuación por el SHA-1 como n bloques de 512 bits.

Supongamos que el mensaje tiene una longitud $l < 2^{64}$. Antes de introducirlo en el SHA-1, se rellena a la derecha como sigue:

a) Se agrega "1" (uno). Ejemplo: si el mensaje original es "01010000", se rellena hasta "010100001".

b) Se agregan "0" (ceros). El número de "0" (ceros) dependerá de la longitud original del mensaje. Los últimos 64 bits del último bloque de 512 bits se reservan para la longitud l del mensaje original.

Ejemplo: Supongamos que el mensaje original es la cadena de bits:

01100001 01100010 01100011 01100100 01100101

Después del paso a), queda en:

01100001 01100010 01100011 01100100 01100101 1

Puesto que l = 40, el número de bits en el mensaje anterior es de 41 y se agregan 407 "0" (ceros), con lo que el número total de bits es ahora de 448. Esto da, en hexadecimal, lo siguiente:

61626364 65800000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000

d) Se obtiene la representación en 2 palabras de l, el número de bits del mensaje original. Si $l < 2^{32}$ la primera palabra es todo ceros. Agregar estas dos palabras al mensaje rellenado.

Ejemplo: Supongamos que el mensaje original es como en b). La longitud es entonces l = 40 (se señala que l se calcula antes de efectuar cualquier relleno). La representación de dos palabras en 40 es 00000000 00000028 en hexadecimal. Con ello, el mensaje rellenado final en hexadecimal es:

61626364 65800000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000028

El mensaje rellenado contendrá 16* n palabra, siendo n < 0. El mensaje rellenado se considera como una secuencia de n bloques M_1, M_2, ..., M_n, donde cada M_i contiene 16 palabras y M_1 contiene los primeros caracteres (o bits) del mensaje.

II.5 Funciones utilizadas

En el SHA-1 se utiliza una secuencia de funciones lógicas f_0, f_1, ..., f_{79}. Cada f_t, $0 \le t \le 79$, opera en tres palabras de 32 bits B, C, D y produce una palabra de 32 bits como salida. f_t (B,C,D) se define como sigue: para las palabras B, C, D:

f_t(B,C,D) = (B AND C) OR ((NOT B) AND D) ($0 \le t \le 19$)

f_t(B,C, D) = B XOR C XOR D ($20 \le t \le 39$)

f_t(B,C,D) = (B AND C) OR (B AND D) OR (C AND D) ($40 \le t \le 59$)

f_t(B,C, D) = B XOR C XOR D ($60 \le t \le 79$).

II.6 Constantes utilizadas

En el SHA-1 se utiliza una secuencia de palabras constantes K(0), K(1), ..., K(79). En hexadecimal, vienen dadas por

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

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

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

K_t= CA62C1D6 ($60 \le t \le 79$).

II.7 Cálculo del digesto del mensaje

El digesto del mensaje se calcula utilizando el mensaje rellenado final. En el cálculo se utilizan dos memorias tampón intermedias, formada cada una de ellas por cinco palabras de 32 bits, y una secuencia de ochenta palabras de 32 bits. Las palabras de la primera memoria tampón intermedia de 5 palabras se designan por A, B, C, D, E. Las palabras de la segunda memoria tampón intermedia de

5 palabras se designan por H_0, H_1, H_2, H_3, H_4. Las palabras de la secuencia de 80 palabras se designan por W_0, W_1, ..., W_{79}. También se emplea una memoria tampón única TEMP.

Para generar el digesto del mensaje, se procesan en orden los bloques de 16 palabras M_1, M_2, ..., M_n definidos en la cláusula 4. El procesamiento de cada M_i se lleva a cabo en 80 pasos.

Antes de procesar cualquier bloque, las $\{H_i\}$ se inicializan como sigue en hexadecimal:

H_0 = 67452301

H_1 = EFCDAB89

H_2 = 98BADCFE

H_3 = 10325476

H_4 = C3D2E1F0

A continuación se procesan M_1, M_2 ..., M_n. Para procesar M_i, se procede como sigue:

a) Se divide M_i en 16 palabras W_0, W_1, ..., W_{15}, donde W_0 es la palabra situada más a la izquierda.

b) Para t = 16 a 79, sea $W_t = S^1 (W_{1-3}$ XOR W_{t-8} XOR W_{t-14} XOR $W_{t-16})$.

c) Sea $A = H_0$, $B = H_1$, $C = H_2$, H_3, $E = H_4$.

d) Para t = 0 a 79 hacer:

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

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

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

Después de procesar M_n, el digesto del mensaje es la cadena de 160 bits representada por las 5 palabras:

H_0, H_1, H_2, H_3, H_4.

II.8 Método de cálculo alternativo

En lo anterior se supone que la secuencia W_0, ..., W_{79} se implementa como una disposición de 80 palabras de 32 bits. Tal cosa es válida desde el punto de vista de la minimización del tiempo de ejecución, ya que las direcciones de W_{t-3}, ..., W_{t-16} del paso b) se calculan fácilmente. Si lo que tiene prioridad es el espacio, una alternativa consiste en considerar la {Wt} como una cola circular, que se puede implementar utilizando una disposición de 16 palabras de 32 bits W[0], ... W[15]. En este caso, sea MASK = 0000000F en hexadecimal.

El procesamiento entonces de M_i es como sigue:

a) Dividir M_i en 16 palabras (W[0], ..., W[15], donde W[0] es la palabra situada más a la izquierda.

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

c) Para t = 0 a 79 hacer:

s = t ^ MASK;

si ($t \geq 16$) $W[s] = S^1 (W[(s + 13) \text{ ^ } MASK] \text{ XOR } W[(s + 8) \text{ AND } MASK] \text{ XOR } W[(s + 2) \text{ ^MASK}] \text{ XOR } W[s])$;

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

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

d) Sea $H_0 = H_0 + A$, $H_1 = H_1 + B$, $H_2 = H_2 + C$, $H_3 = H3 + D$, $H_4 = H_4 + E$.

II.9 Comparación de los métodos

Con los métodos de las cláusulas II.7 y II.8 se obtiene el mismo digesto de mensaje. Aunque utilizando el método de la cláusula II.8 se ahorra el almacenamiento de 64 palabras de 32 bits, es probable que prolongue el tiempo de ejecución debido a la mayor complejidad de los cálculos de direcciones de la {W[t]} en el paso c). Se pueden aplicar otros métodos de cálculo, de conformidad con la norma, que dan resultados idénticos.

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 y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación

Ginebra, 2001