Unión Internacional de Telecomunicaciones

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

X.1365

(03/2020)

SERIE X: REDES DE DATOS, COMUNICACIONES DE SISTEMAS ABIERTOS Y SEGURIDAD

Aplicaciones y servicios con seguridad (2) – Seguridad en la Internet de las cosas (IoT)

Metodología de seguridad para el uso de criptografía basada en la identidad para dar soporte a servicios de Internet de las cosas mediante redes de telecomunicaciones

Recomendación UIT-T X.1365

Unión Internacional de Telecomunicaciones

RECOMENDACIONES UIT-T DE LA SERIE X

REDES DE DATOS, COMUNICACIONES DE SISTEMAS ABIERTOS Y SEGURIDAD

REDES PÚBLICAS DE DATOS	X.1–X.199
INTERCONEXIÓN DE SISTEMAS ABIERTOS	X.200–X.299
INTERFUNCIONAMIENTO ENTRE REDES	X.300–X.399
SISTEMAS DE TRATAMIENTO DE MENSAJES	X.400–X.499
DIRECTORIO	X.500–X.599
GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS	X.600–X.699
GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	X.700–X.799
SEGURIDAD	X.800–X.849
APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS	X.850–X.899
PROCESAMIENTO DISTRIBUIDO ABIERTO	X.900–X.999
SEGURIDAD DE LA INFORMACIÓN Y DE LAS REDES	
Aspectos generales de la seguridad	X.1000–X.1029
Seguridad de las redes	X.1030–X.1049
Gestión de la seguridad	X.1050–X.1069
Telebiometría	X.1080–X.1099
APLICACIONES Y SERVICIOS CON SEGURIDAD (1)	
Seguridad en la multidifusión	X.1100–X.1109
Seguridad en la red residencial	X.1110–X.1119
Seguridad en las redes móviles	X.1120–X.1139
Seguridad en la web	X.1140–X.1149
Protocolos de seguridad (1)	X.1150–X.1159
Seguridad en las comunicaciones punto a punto	X.1160–X.1169
Seguridad de la identidad en las redes	X.1170–X.1179
Seguridad en la TVIP	X.1180–X.1199
SEGURIDAD EN EL CIBERESPACIO	
Ciberseguridad	X.1200–X.1229
Lucha contra el correo basura	X.1230–X.1249
Gestión de identidades	X.1250–X.1279
APLICACIONES Y SERVICIOS CON SEGURIDAD (2)	
Comunicaciones de emergencia	X.1300–X.1309
Seguridad en las redes de sensores ubicuos	X.1310–X.1339
Seguridad de las redes eléctricas inteligentes	X.1330–X.1339
Recomendaciones relacionadas con la PKI	X.1340–X.1349
Seguridad en la Internet de las cosas (IoT)	**X.1360–X.1369**
Seguridad en los sistemas de transporte inteligente (ITS)	X.1370–X.1379
Seguridad de tecnología de libro mayor distribuido	X.1400–X.1429
Seguridad de tecnología de libro mayor distribuido	X.1430–X.1449
Protocolos de seguridad (2)	X.1450–X.1459
INTERCAMBIO DE INFORMACIÓN DE CIBERSEGURIDAD	
Aspectos generales de la ciberseguridad	X.1500–X.1519
Intercambio de estados/vulnerabilidad	X.1520–X.1539
Intercambio de eventos/incidentes/heurística	X.1540–X.1549
Intercambio de políticas	X.1550–X.1559
Petición de heurística e información	X.1560–X.1569
Identificación y descubrimiento	X.1570–X.1579
Intercambio asegurado	X.1580–X.1589
SEGURIDAD DE LA COMPUTACIÓN EN NUBE	
Visión general de la seguridad de la computación en nube	X.1600–X.1601
Diseño de la seguridad de la computación en nube	X.1602–X.1639
Prácticas óptimas y directrices en materia de seguridad de la computación en nube	X.1640–X.1659
Aplicación práctica de la seguridad de la computación en nube	X.1660–X.1679
Otras cuestiones de seguridad de la computación en nube	X.1680–X.1699
COMUNICACIÓN CUÁNTICA	
Terminologías	X.1700–X.1701
Generador de números aleatorio cuántico	X.1702–X.1709
Marco de seguridad QKDN	X.1710–X.1711
Diseño de seguridad para QKDN	X.1712–X.1719
Técnicas de seguridad para QKDN	X.1720–X.1729
SEGURIDAD DE LOS DATOS	
Seguridad de los macrodatos	X.1750–X.1759
SEGURIDAD DE 5G	X.1800–X.1819

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

Recomendación UIT-T X.1365

Metodología de seguridad para el uso de criptografía basada en la identidad para dar soporte a servicios de Internet de las cosas mediante redes de telecomunicaciones

Resumen

La Recomendación UIT-T X.1365 describe una metodología de seguridad para utilizar la tecnología de clave pública IBC (criptografía basada en la identidad) en los servicios de Internet de las cosas (IoT) a través de las redes de telecomunicaciones, incluidos los mecanismos de gestión de identidades, la arquitectura de gestión de claves, las operaciones de gestión de claves y la autentificación.

La metodología de seguridad tradicional basada en certificados implica pesadas operaciones de gestión de claves, que comprende la emisión, consulta y revocación de certificados. Estos sistemas se enfrentan a grandes dificultades para seguir el ritmo del creciente número de dispositivos conectados a la IoT al tiempo que mantiene un rendimiento suficiente.

La tecnología IBC es otra metodología de seguridad que utiliza como clave pública la identidad de la entidad. Una característica esencial de la IoT es que todo tiene un identificador único (ID). Si se utilizan los ID como claves públicas, no se requieren certificados. En consecuencia, la solución de seguridad IBC utiliza una gestión de claves más sencilla, permite a las autoridades distribuidas controlar sus propios dispositivos y se adapta bien tanto a un gran número de puntos terminales como a diversos dispositivos.

Historia

Edición	Recomendación	Aprobación	Comisión de Estudio	ID único*
1.0	ITU-T X.1365	2020-03-26	17	11.1002/1000/14089

Palabras clave

Criptografía basada en la identidad (IBC), IoT, metodología de seguridad, seguridad de los datos de usuario.

* Para acceder a la Recomendación, sírvase digitar el URL http://handle.itu.int/ en el campo de dirección del navegador, seguido por el identificador único de la Recomendación. Por ejemplo, http://handle.itu.int/11.1002/1000/11830-en.

PREFACIO

La Unión Internacional de Telecomunicaciones (UIT) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones y de las tecnologías de la información y la comunicación. El Sector de Normalización de las Telecomunicaciones de la UIT (UIT-T) 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 Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), 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 AMNT.

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.

La observancia de esta Recomendación es voluntaria. Ahora bien, la Recomendación puede contener ciertas disposiciones obligatorias (para asegurar, por ejemplo, la aplicabilidad o la interoperabilidad), por lo que la observancia se consigue con el cumplimiento exacto y puntual de todas las disposiciones obligatorias. La obligatoriedad de un elemento preceptivo o requisito se expresa mediante las frases "tener que, haber de, hay que + infinitivo" o el verbo principal en tiempo futuro simple de mandato, en modo afirmativo o negativo. El hecho de que se utilice esta formulación no entraña que la observancia se imponga a ninguna de las partes.

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 [ha recibido/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 en la dirección http://www.itu.int/ITU-T/ipr/.

ÍNDICE

Página

Recomendación UIT-T X.1365

Metodología de seguridad para el uso de criptografía basada en la identidad para dar soporte a servicios de Internet de las cosas mediante redes de telecomunicaciones

1 Alcance

En esta Recomendación se proporciona una metodología de seguridad para el uso de la tecnología de criptografía basada en la identidad (IBC) en apoyo de los servicios de Internet de las cosas (IoT) a través de las redes de telecomunicaciones. Esta metodología de seguridad abarca los mecanismos para la identificación de dispositivos, la cuestión de la clave privada, el parámetro público de búsqueda y los protocolos de autentificación.

NOTA – Esta metodología no se limita al servicio IoT, también puede ser utilizado por otros servicios.

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. En esta Recomendación, la referencia a un documento, en tanto que autónomo, no le otorga el rango de una Recomendación.

[IETF RFC 4764] IETF RFC 4764 (2007), *The EAP-PSK protocol: A Pre-Shared Key Extensible Authentication Protocol (EAP) Method*.

[IETF RFC 5091] IETF RFC 5091 (2007), *Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems*.

[IETF RFC 5216] IETF RFC 5216 (2008), *The EAP-TLS Authentication Protocol*.

[IETF RFC 5280] IETF RFC 5280 (2008), *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.

[IETF RFC 5408] IETF RFC 5408 (2009), *Identity-Based Encryption Architecture and Supporting Data Structures*.

[IETF RFC 5480] IETF RFC 5480 (2009), *Elliptic Curve Cryptography Subject Public Key Information*.

[IETF RFC 5958] IETF RFC 5958(2010), *Asymmetric Key Packages*.

[IETF RFC 6507] IETF RFC 6507 (2012), *Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)*.

[IETF RFC 6508] IETF RFC 6508 (2012), *Sakai–Kasahara Key Encryption (SAKKE)*.

[IETF RFC 6960] IETF RFC 6960 (2013), *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*.

[IETF RFC 7250] IETF RFC 7250 (2014), *Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)*.

[IETF RFC 8446] IETF RFC 8446 (2018), *The Transport Layer Security (TLS) Protocol Version 1.3*.

[ISO/CEI 11770-3] ISO/CEI 11770-3:2015, *Information technology – Security techniques – Key management – Part 3: Mechanisms using asymmetric techniques*.

[ISO/CEI 14888-3] ISO/CEI 14888-3:2018, *IT Security techniques – Digital signatures with appendix – Part 3: Discrete logarithm based mechanisms*.

[ISO/CEI 18033-5] ISO/CEI 18033-5:2015, *Information technology – Security techniques – Encryption algorithms – Part 5: Identity-based ciphers*.

3 Definiciones

3.1 Términos definidos en otros documentos

En la presente Recomendación se utilizan los siguientes términos definidos en otros documentos:

3.1.1 proveedor de identidad (IdP) [b-UIT-T Y.2720]: entidad que crea, mantiene y gestiona información digna de confianza sobre la identidad de otras entidades (por ejemplo, usuarios/abonados, organizaciones y dispositivos) y ofrece servicios de identidad basados en relaciones de confianza, negocio y otros tipos de relaciones.

3.1.2 identificador (ID) [b-UIT-T E.101]: serie de cifras, caracteres y símbolos utilizados para identificar inequívocamente a un abonado, un usuario, un elemento de red, una función, una entidad de red, un servicio o una aplicación. Los identificadores pueden utilizarse para el registro y la autorización. Pueden ser públicos para todas las redes o privados para una red específica (normalmente los identificadores privados no se revelan a terceros).

3.1.3 clave pública maestra (MPK) [ISO/CEI 18033-5]: el valor público determinado con carácter único por la correspondiente clave maestra secreta.

3.1.4 clave secreta maestra (MSK) [ISO/CEI 18033-5]: el valor secreto utilizado por el generador de claves privadas para calcular claves privadas para un algoritmo de criptografía basado en la identidad.

3.1.5 generador de claves privadas (PKG) [ISO/CEI 18033-5]: entidad o función que genera un conjunto de claves privadas.

3.2 Términos definidos en esta Recomendación

En la presente Recomendación se definen los siguientes términos:

3.2.1 dominio de identidad: colección de entidades que comparten el mismo conjunto de parámetros públicos y reglas de denominación de identidades.

3.2.2 parámetro público: uno de los parámetros para el cálculo criptográfico, incluida una selección de un esquema o función criptográfico particular de una familia de esquemas o funciones criptográficos o de una familia de espacios matemáticos y la clave pública maestra.

3.2.3 servidor de parámetros públicos: entidad que proporciona parámetros públicos previa solicitud.

3.2.4 módulo de seguridad (SecM): un elemento de *software* o de *hardware*, o la composición de *software* y *hardware* que implementa de manera segura mecanismos criptográficos y ofrece servicios de seguridad.

4 Abreviaturas y acrónimos

En la presente Recomendación se utilizan las siguientes abreviaturas y acrónimos:

4G	Cuarta generación
5G	Quinta generación
AuC	Centro de autentificación (*authentication centre*)
AGW	Pasarela combinada (*aggregate gateway*)
AK	Clave de autentificación (*authentication key*)
AKA	Acuerdo de clave autentificada (*authenticated key agreement*)
AN	Nodo de acceso (*access node*)
AS	Sistema de acceso (*access system*)
ASN.1	Notación de sintaxis abstracta uno (*abstract syntax notation one*)
AU	Unidad de autentificación (*authentication unit*)
BN	Barreto-Naehrig
BLS-12	Grado de inserción Barreto-Lynn-Scott 12 (*Barreto-Lynn-Scott embedding degree 12*)
BLS-24	Grado de inserción Barreto-Lynn-Scott 24 (*Barreto-Lynn-Scott embedding degree 24*)
CRL	Lista de revocación de certificados (*certificate revocation list*)
DER	Normas de codificación distinguida (*distinguished encoding rules*)
EAP	Protocolo de autentificación extensible (*extensible authentication protocol*)
ECCSI	Firmas sin certificado basadas en la curva elíptica para encriptación basada en la identidad (*elliptic curve-based certificateless signatures for identity-based encryption*)
EID	Identificador de eUICC (*eUICC-ID*)
EIS	Conjunto de información de eUICC (*eUICC information set*)
eUICC	Tarjeta de circuito integrado universal integrada (*embedded universal integrated circuit card*)
EUM	Fabricante de eUICC (*eUICC manufacturer*)
GW	Pasarela (*gateway*)
HSM	Módulo de seguridad de *hardware* (*hardware security module*)
HTTP	Protocolo de transferencia de hipertexto (*hypertext transfer protocol*)
IBAKA	Acuerdo sobre claves autentificadas basadas en la identidad (*identity-based authenticated key agreement*)
IBC	Criptografía basada en la identidad (*identity-based cryptography*)
IBE	Encriptación basada en la identidad (*identity-based encryption*)
IBS	Firma basada en la identidad (*identity-based signature*)
ID	Identificador
IdP	Proveedor de identidad (*identity provider*)
IMSI	Identidad internacional de suscripción al servicio móvil (*international mobile subscription identity*)
IoT	Internet de las cosas (*Internet of things*)

ISP	Plataforma de servicio IoT (*IoT service platform*)
IRL	Lista de revocación de identidades (*identity revocation list*)
ISD	Dominio de seguridad del expedidor (*issuer security domain*)
KDF	Función de cálculo de claves (*key derivation function*)
KDK	Clave de cálculo de claves (*key derivation key*)
KEK	Clave de encriptado de claves (*key encryption key*)
KEM	Mecanismo de encapsulado de clave (*key encapsulation mechanism*)
KMIP	Protocolo de interoperabilidad de la gestión de claves (*key management interoperability protocol*)
KMS	Servicio de gestión de claves (*key management service*)
KPAK	Clave de autentificación pública del KMS (*KMS public authentication key*)
KSS-16	Grado de inserción Kachisa-Schaefer-Scott 16 (*Kachisa-Schaefer-Scott embedding degree 16*)
KSS-18	Grado de inserción Kachisa-Schaefer-Scott 18 (*Kachisa-Schaefer-Scott embedding degree 18*)
LTE	Evolución a largo plazo (*long-term evolution*)
LTE-M	Evolución a largo plazo, categoría M1 (*long-term evolution, category M1*)
MAC	Control de acceso a los medios (*media access control*)
MNO	Operador de red móvil (*mobile network operator*)
MSK	Clave secreta maestra (*master secret key*)
NB-IoT	Internet de las cosas de banda estrecha (*narrowband Internet of things*)
OCSP	Protocolo de estado de certificado en línea (*online certificate status protocol*)
OID	Identificador de objeto (*object identifier*)
OISP	Protocolo de estado de identidad en línea (*online identity status protocol*)
PKG	Generador de claves privadas (*private key generator*)
PKI	Infraestructura de clave pública (*public key infrastructure*)
PPS	Servidor de parámetros públicos (*public parameter server*)
PVT	Testigo de verificación pública (*public verification token*)
RSF	Función de servidor de revocación (*revocation server function*)
SecM	Módulo de seguridad (*security module*)
SK	Sakai-Kasahara
SM-DP	Preparación de los datos de gestor de abono (*subscription manager data preparation*)
SM-SR	Encaminamiento seguro del gestor de abono (*subscription manager secure routing*)
SOK	Sakai-Ohgishi-Kasahara
SSK	Clave de firma secreta (*secret signing key*)
TLS	Seguridad de la capa de transporte (*transport layer security*)
TLV	Etiqueta, longitud y vector (*tag, length and vector*)
TVP	Parámetro de variante temporal (*time-variant parameter*)

UE Equipo de usuario (*user equipment*)

UICC Tarjeta universal de circuito integrado (*universal integrated circuit card*)

5 Convenios

Ninguno.

6 Presentación

De conformidad con la cláusula 6.1 de [b-UIT-T Y.4000], la IoT "puede concebirse como una infraestructura global de la sociedad de la información, que permite ofrecer servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperabilidad de tecnologías de la información y la comunicación (TIC) presentes y futuras". La seguridad de la IoT es una de las preocupaciones prioritarias debido a la naturaleza ubicua de los dispositivos y a la creciente sensibilidad de los datos de usuario. En [b-UIT-T Y.4100] se describen los requisitos comunes de seguridad de alto nivel de la IoT, relacionados entre otros aspectos con la seguridad de las comunicaciones, la seguridad de la gestión de datos, la seguridad de la prestación de servicios y la autentificación y autorización mutuas. En [b-UIT-T X.1361] se analizan con mayor detalle las amenazas y los problemas de seguridad en un entorno de Internet de las cosas y se describen las capacidades que permitirían abordar y reducir esas amenazas y resolver esos problemas, por ejemplo:

– una capacidad de comunicación segura para soportar comunicaciones seguras, fiables y con protección de la privacidad;
– una capacidad de gestión de claves segura para soportar comunicaciones seguras;
– una capacidad de gestión de datos segura para proporcionar una gestión de datos segura, fiable y con protección de privacidad;
– una capacidad de autenticación para la autenticación de dispositivos;
– una capacidad de autorización (control de acceso) para autorizar dispositivos;
– una capacidad de implementar protocolos seguros basándose en algoritmos criptográficos ligeros.

Los dispositivos IoT se caracterizan por la limitación de recursos, como las funciones de cálculo y comunicación. Debido a la naturaleza de estos dispositivos, la gestión de los requisitos de seguridad en un sistema IoT se enfrenta a dificultades nuevas. Al evaluar las soluciones de seguridad para la IoT hay que tener en cuenta algunos factores clave, en especial la facilidad máxima de despliegue, las operaciones de gestión ligeras y la autoridad distribuida.

Como se describe en [b-UIT-T X.1361], la autentificación, el control de acceso y la integridad y confidencialidad de los datos son algunos de los servicios esenciales necesarios para la seguridad de la IoT. Para prestar esos servicios se pueden utilizar mecanismos de criptografía de claves tanto públicas como simétricas.

La solución de seguridad basada en claves simétricas es relativamente sencilla. Sin embargo, no responde bien en los escenarios entre pares, como las aplicaciones máquina-máquina en la IoT sin servicio en línea que actúe como intermediario de confianza o sin compartición previa de los secretos entre dispositivos pares. La comunicación segura entre sistemas también resulta más difícil si no se exponen los secretos de usuario a los pares.

Una solución tradicional de criptografía de claves públicas con certificados conlleva operaciones complejas de gestión de claves, como emisión, consulta, distribución, verificación y revocación de certificados. A estos sistemas les resulta muy difícil seguir el ritmo de aumento del número de dispositivos y funcionalidades en la IoT y mantener al mismo tiempo un rendimiento adecuado. La sobrecarga que supone el intercambio de certificados en los protocolos de seguridad también genera

problemas, en especial en las redes de Internet de las cosas de banda estrecha (NB-IoT) con unidad de paquete de datos pequeño.

La criptografía basada en la identidad (IBC) es otro tipo de tecnología que utiliza la identidad de una entidad como clave pública. Una característica esencial de la IoT es que todo tiene un identificador (ID) único. Si se utilizan estos ID como claves públicas, no se requieren certificados. En consecuencia, una solución de seguridad de IBC utiliza una gestión de claves más sencilla, permite a las autoridades distribuidas controlar sus propios dispositivos y se adapta bien tanto a un gran número de puntos terminales como a diversos dispositivos. Al no transmitirse certificados, los protocolos de seguridad se pueden ejecutar con más eficacia.

En un sistema de IBC, existe una parte de confianza denominada servicio de gestión de claves (KMS) que se ocupa de generar la clave privada de cada entidad. Antes de prestar el servicio de generación de claves, el KMS pone en marcha un proceso de inicialización del sistema mediante la invocación de una función **IBSetup** que determina, sobre la base de un parámetro de seguridad dado, un conjunto de parámetros de sistema y genera una clave secreta maestra (MSK) y una clave pública maestra (MPK). Es importante destacar que el KMS tiene la misma función como generador de claves privadas (PKG). En consecuencia, en aras de la simplicidad, en esta Recomendación se utilizará KMS y PKG indistintamente, y la combinación de los parámetros de sistema y la MPK se denominará parámetros públicos. El KMS trata la MSK con total confidencialidad e informa abiertamente de los parámetros públicos que, en caso necesario, pueden ser publicados a través de un servidor de parámetros públicos (PPS) de servicio dedicado.

Un sistema de seguridad de IBC típico puede utilizar múltiples mecanismos de IBC, como encriptación basada en la identidad (IBE), firma basada en la identidad (IBS) y acuerdo sobre claves autentificadas basadas en la identidad (IBAKA), para prestar diferentes servicios de seguridad, por ejemplo, confidencialidad de los datos, autentificación de entidades y establecimiento de canales seguros. Todos estos algoritmos de IBC pueden considerarse como una composición de dos conjuntos de funciones. Uno de estos conjuntos está formado por funciones de generación de claves, que generan pares de clave privada y clave pública basadas en la identidad. La función de generación de clave privada (**IBExtract**) genera una clave privada partir de un ID, la MSK y los parámetros públicos. La función de derivación de clave pública de identidad (**IBDerivate**) calcula una clave pública a partir de un ID y los parámetros públicos. El otro conjunto de funciones, por ejemplo, encriptado o desencriptado (**IBEnc/IBDec**), firma o verificación (**IBSign/IBVerify**) y protocolo de establecimiento de clave de sesión autentificada, utiliza los pares de claves generados para completar las operaciones criptográficas correspondientes.

Varias organizaciones de elaboración de normas han uniformizado la tecnología de IBC, entre las que destacan la Organización Internacional de Normalización (ISO), la Comisión Electrotécnica Internacional (CEI), el Grupo de Tareas sobre Ingeniería de Internet (IETF), el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE), el Instituto Europeo de Normas de Telecomunicaciones (ETSI) y la Administración Nacional de Normas de China (SAC). En la bibliografía se incluye una lista de algunas normas pertinentes elaboradas por estas organizaciones. OneM2M también contempla la posibilidad de utilizar tecnologías de IBC para la versión 4 de las redes de IoT, cuyo análisis de seguridad figura en [b-ETSI TR 118 508].

En esta Recomendación se describe un marco de seguridad para el uso de la tecnología de IBC en la provisión de capacidades de seguridad para servicios de IoT a través de las redes de telecomunicaciones. El marco abarca la gestión de identidades, la arquitectura de gestión de claves, las operaciones de gestión de claves y la autentificación, así como los protocolos de acuerdo de clave para el uso de la IBC.

7 Arquitectura de referencia del sistema para servicios de IoT a través de las redes de telecomunicaciones

En esta cláusula se presenta una arquitectura general de referencia del sistema para servicios de IoT a través de las redes de telecomunicaciones. En la Figura 1 se muestra un esquema de la arquitectura de referencia del sistema para servicios de IoT. El sistema está formado por tres dominios: dispositivo IoT, sistema de acceso (AS) y plataforma de servicio IoT (ISP).

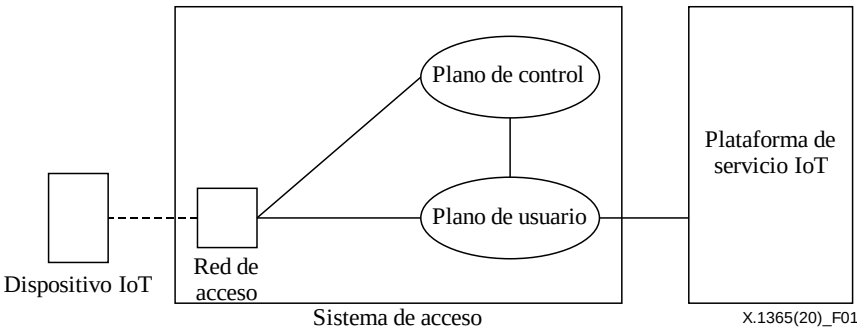

Figura 1 – Esquema de la arquitectura del sistema para servicios de Internet de las cosas

Los dispositivos IoT se ocupan de recopilar datos o ejecutar acciones. La mayoría de ellos pueden establecer una conexión con un sistema de telecomunicaciones y comunicar con una ISP. Actualmente, la mayor parte de los dispositivos IoT se conectan con una ISP a través de un enlace inalámbrico establecido con una red de telecomunicaciones. En esta Recomendación, el AS hace referencia a redes de telecomunicaciones. Por regla general, consta de dos partes: red de acceso (AN) y red medular. La red medular puede dividirse a su vez en dos partes: plano de control y plano de usuario, que son responsables de la señalización de control y la transmisión de datos respectivamente.

La red de telecomunicaciones es una conexión inalámbrica tradicional en uso desde hace varias generaciones. Históricamente, las redes de telecomunicaciones se han diseñado en apoyo de la comunicación móvil de las personas con funciones de itinerancia ininterrumpida. En los últimos años, desde la aparición de las redes de evolución a largo plazo (LTE) de cuarta generación (4G), el soporte de los dispositivos IoT también se tiene en cuenta en el diseño. Por ejemplo, en la 4G-LTE se han desarrollado las tecnologías de categoría M1 (LTE-M) y NB-IoT para dar soporte a los dispositivos IoT.

La mayoría de los sistemas de telecomunicaciones actuales están formados por tres componentes: terminales o equipo de usuario (UE), un AN y redes medulares. En este ejemplo se presupone que tanto el AN como las redes medulares pertenecen al AS, como se muestra en la Figura 1. Normalmente, los servicios de IoT se encuentran fuera de las redes de telecomunicaciones y cuentan con interfaces para la transmisión de datos y la gestión del servicio. Para mejorar el apoyo a los servicios de IoT, las especificaciones de sistema de las redes de telecomunicaciones incorporan ahora un diseño de IoT más específico. La integración entre las redes de telecomunicaciones y los servicios de IoT se ha estrechado en los últimos años.

Se han desarrollado especificaciones de sistema para redes de quinta generación (5G) y los servicios de IoT admiten tecnologías de clave pública, como la autentificación de acceso a la red. Como se señala en la cláusula 6, en comparación con otras tecnologías de clave pública, la IBC se caracteriza por la gestión más sencilla y la transmisión más eficaz. Por consiguiente, para utilizar la IBC en servicios de IoT a través de las redes de telecomunicaciones, es necesario que la especificación sea un complemento estándar de las especificaciones existentes.

8 Marco para la utilización de la criptografía basada en la identidad en servicios de IoT a través de las redes de telecomunicaciones

En esta cláusula se proporciona un marco para la utilización de tecnologías de clave pública de IBC en servicios de IoT a través de las redes de telecomunicaciones. El marco incluye una arquitectura de sistema con los componentes de red necesarios para utilizar las tecnologías de IBC. Además, se especifica un marco de gestión de claves para la IBC, esencial para los sistemas que utilizan tecnología de IBC. Se abordan también otras cuestiones importantes, como gestión de claves, denominación de identidades y protocolos de autentificación.

8.1 Arquitectura de sistema de IoT con criptografía basada en la identidad

En los servicios de IoT que se ejecutan en redes de telecomunicaciones, la IBC posibilita la autentificación de acceso a la red, la autentificación de acceso al servicio o ambas. La autentificación de acceso a la red determina si un dispositivo tiene autorización para acceder a la red, y la autentificación de acceso al servicio establece si un dispositivo puede acceder a una ISP.

Los dispositivos IoT pueden acceder a la red de telecomunicaciones de manera directa o indirecta. Existen por tanto dos modelos de acceso:

– modelo de conexión directa: los dispositivos IoT se conectan al AS de forma directa;

– modelo de conexión indirecta: los dispositivos IoT se conectan al AS a través de una pasarela combinada (AGW).

En la Figura 2 se muestra una arquitectura de referencia de sistema IoT donde la IBC se utiliza para proteger la seguridad tanto del AS como de la ISP. En lo referente a la seguridad, el AS y la ISP pueden tener requisitos de seguridad propios para los servicios de IoT. Dada la posibilidad de que el AS o la ISP proporcionen credenciales de seguridad, la utilización de la IBC para redes de IoT puede basarse en uno de los tres escenarios siguientes.

– Utilización de la IBC para la protección de la seguridad del AS.

  En este escenario, el AS proporciona y gestiona las credenciales de seguridad para el acceso a la red almacenadas en dispositivos IoT. El AS autentifica los dispositivos IoT al conectarse a ellos. Por ejemplo, un dispositivo IoT calcula la firma IBS de acuerdo con la clave privada proporcionada por el AS y la envía al AS. En consecuencia, el AS puede autentificar el dispositivo IoT basándose en la firma IBS indicada en los mensajes de autentificación. Si la verificación se completa correctamente, el AS envía los datos procedentes del dispositivo IoT al servidor de IoT.

– Utilización de la IBC para la protección de la seguridad de la ISP.

  La ISP proporciona y gestiona las credenciales de IBC almacenadas en dispositivos IoT para poder acceder al servicio. La ISP autentifica los dispositivos IoT basándose en la firma generada con las credenciales de IBC.

– Utilización de la IBC para la protección de la seguridad del AS y la ISP.

  El AS o la ISP, por separado o de manera conjunta, proporcionan y gestionan las credenciales de IBC almacenadas en dispositivos IoT para poder acceder al servicio. Tanto el AS como la ISP pueden autentificar el dispositivo IoT con el mismo conjunto de credenciales.

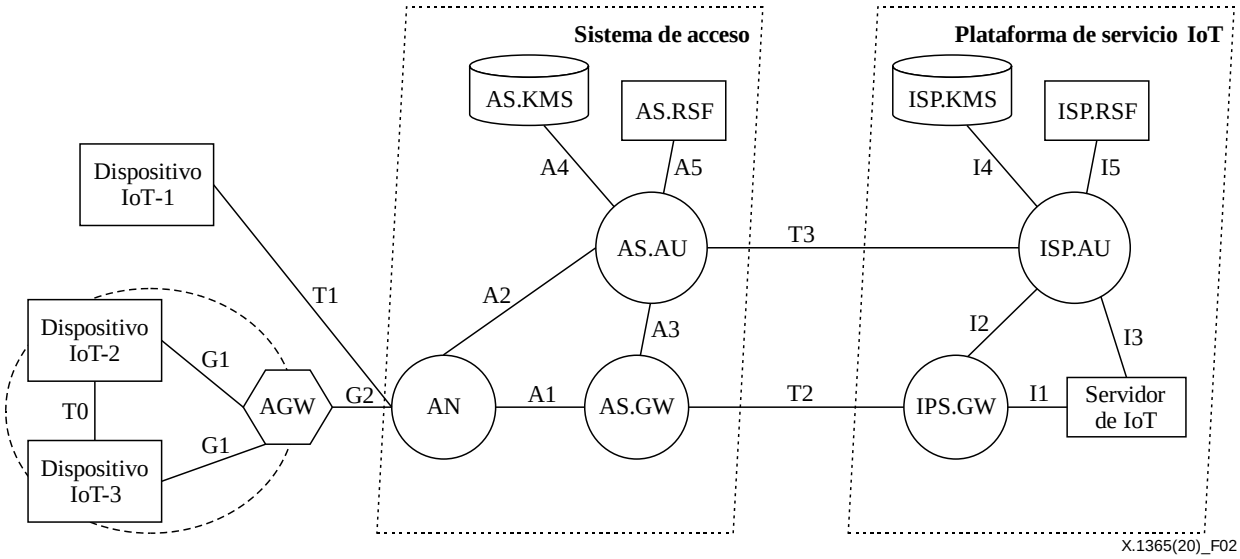

Figura 2 – Arquitectura de sistema de IoT con criptografía basada en la identidad en escenarios de protección de la seguridad tanto del sistema de acceso como de la plataforma de servicio IoT

Los tres escenarios descritos más arriba abarcan la mayoría de los casos de uso de IBC para el acceso a redes y servicios. No obstante, podrían acontecer otros escenarios ajenos al ámbito de la presente Recomendación.

La arquitectura de sistema IoT basada en IBC consta de las funciones de red (NF) y los dispositivos siguientes:

– Sistema de acceso (AS): sistema de acceso para dispositivos IoT o AGW, que incluye un nodo de acceso (AS.AN), una función de sistema de gestión de claves (AS.KMS), una unidad de autentificación (AS.AU), una función de servidor de revocación (AS.RSF) y una pasarela (AS.GW).

– Plataforma de servicio IoT (ISP): plataforma para la gestión de servicios de IoT, que incluye una función de sistema de gestión de claves (ISP.KMS), una unidad de autentificación (isp.au), una función de servidor de revocación (ISP.RSF), una pasarela (ISP.GW) y un servidor de IoT. La ISP dará apoyo a la gestión, distribución, autentificación de identidad, encriptado o desencriptado de claves, a la firma o verificación de firmas, etc.

– Pasarela combinada (AGW): nodo de agregación responsable de la conexión de dispositivos IoT, que combina y envía todos los datos de dispositivo IoT al AS. La AGW actúa como intermediario de la transmisión de datos entre los dispositivos IoT y el AN.

– Nodo de acceso (AN): nodo de acceso para los dispositivos IoT o la AGW; puede ser un punto de acceso de red fija o inalámbrica.

– Función de servicio de gestión de claves (KMS): sistema de gestión responsable de la generación, distribución y actualización de las claves y los parámetros de IBC para dispositivos IoT y funciones de red.

– Unidad de autentificación (AU): una AU autentifica los dispositivos IoT basándose en el sistema de IBC.

– Función de servidor de revocación (RSF): servidor que conserva una lista de revocación de identidades (IRL). No se pueden utilizar las claves públicas o las identidades de la lista de revocación.

NOTA – Tanto el AS como la ISP pueden tener KMS, AU y RSF propios.

– Pasarela del sistema de acceso (AS.GW): elemento de red conectado con la IoT GW, responsable de la transmisión de datos de usuario de IoT.
– Pasarela de IoT (IoT GW): pasarela responsable de enviar o combinar los datos y transmitirlos al servidor de IoT, o de enviar los datos y las señales procedentes del servidor de IoT a los dispositivos IoT.
– Servidor de IoT: servidor ubicado en el lado proveedor de servicios de IoT que recopila los datos IoT procedentes de la IoT GW.
– Dispositivo IoT: dispositivo extremo que se utiliza para recopilar datos y establecer la conexión con un AN y un servidor de IoT, que proporciona protección de datos, como negociación y encriptado o desencriptado de claves y firma o verificación de firmas.

Las funciones de los puntos de referencia que se muestran en la Figura 2 se describen de la manera siguiente:

– G1: punto de referencia entre un dispositivo IoT y una AGW, que se utiliza para la autentificación y la comunicación de seguridad.
– G2: punto de referencia entre una AGW y un AN, que se utiliza para la señalización y comunicación de datos entre la AGW y el AN.
– T0: punto de referencia entre dispositivos IoT, que se utiliza para la señalización y el intercambio de datos.
– T1: punto de referencia entre dispositivos IoT y un AN, que se utiliza para la autentificación y la comunicación de seguridad.
– T2: puntos de referencia entre la AS.GW y la ISP.GW, que proporcionan un túnel de datos en el plano de usuario entre la AS.GW y la ISP.
– T3: puntos de referencia entre la AS.AU y la ISP.AU para el intercambio de señalizaciones, que incluye el intercambio de identidades o el provisionamiento de claves.
– A1: puntos de referencia entre el AN y la AS.GW para la tunelización de datos del plano de usuario.
– A2: puntos de referencia entre la AS.AU y el AN para la señalización del plano de control.
– A3: puntos de referencia entre la AS.AU y la AS.GW para el protocolo de asignación y gestión de GW en el AS.
– A4: puntos de referencia entre la AS.AU y el AS.KMS para el protocolo de provisionamiento de claves en el AS.
– A5: puntos de referencia entre la AS.AU y la AS.RSF para el protocolo de revocación de identidades o claves en el AS.
– I1: puntos de referencia entre el servidor de IoT y la ISP.G para la tunelización de datos del plano de usuario.
– I2: puntos de referencia entre la ISP.AU y la ISP.GW para el protocolo de asignación y gestión de GW en la ISP.
– I3: puntos de referencia entre la ISP.AU y el servidor de IoT para el intercambio de información, como la información de suscripción relacionada con servicios que se transfiere del servidor de IoT a la ISP.AU, y el mensaje de notificación de autentificación de la ISP.AU al servidor de IoT.
– I4: puntos de referencia entre la ISP.AU y el ISP.KMS para el protocolo de provisionamiento de claves en la ISP.
– I5: puntos de referencia entre la ISP.AU y la ISP.RSF para el protocolo de revocación de identidades o claves en la ISP.

8.2 Arquitectura de gestión de claves

En esta cláusula se describe la arquitectura funcional necesaria para permitir la gestión de claves al utilizar mecanismos de IBC en la IoT. Se consideran dos variantes de arquitectura de gestión de claves, dependiendo de si un dispositivo IoT tiene una tarjeta universal de circuito integrado (UICC) integrada [b-GSMA SGP.02]: a) gestión de claves con IBC en dispositivos IoT con UICC integrada (eUICC); y b) gestión de claves con IBC en dispositivos IoT con UICC no integrada.

Si se utiliza la IBC en dispositivos IoT con eUICC, la arquitectura repite la estructura de aprovisionamiento remoto de eUICC general que figura en [b-GSMA SGP.02] y añade dos entidades de función nuevas: el KMS y el PPS. Este escenario se divide en dos subcasos en función de la ubicación del KMS.

1) El KMS está gestionado por la entidad que se ocupa también del operador de red móvil (MNO) – Figura 3.
2) El KMS está gestionado por la entidad que se ocupa de la preparación de los datos de gestor de abono (SM-DP) – Figura 4.

En ambos subcasos, las claves (esto es, los parámetros públicos y las claves privadas) se generan cuando un MNO emite una solicitud de perfil. Las claves se suministran de manera remota a los dispositivos eUICC en forma de claves instaladas con arreglo a la especificación de provisionamiento de claves a distancia de [b-GSMA SGP.02]. En [b-GSMA SGP.02] se ofrece información detallada sobre los cometidos, las funciones asociadas y las interfaces del provisionamiento a distancia de eUICC. La especificación del perfil, el formato de almacenamiento y la utilización de estas claves en eUICC es ajena al ámbito de la presente Recomendación.

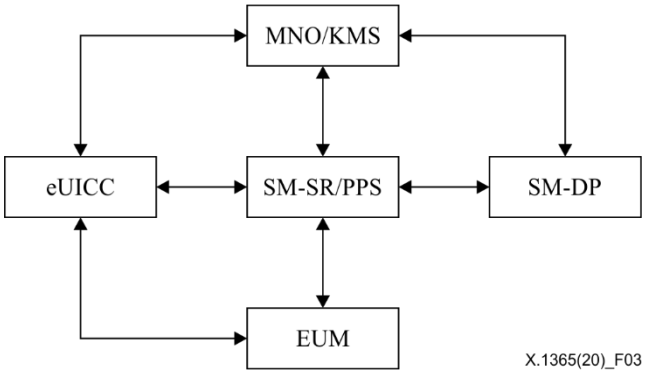

Figura 3 – Arquitectura A de gestión de claves de criptografía basada en la identidad, para dispositivos de IoT con UICC integrada

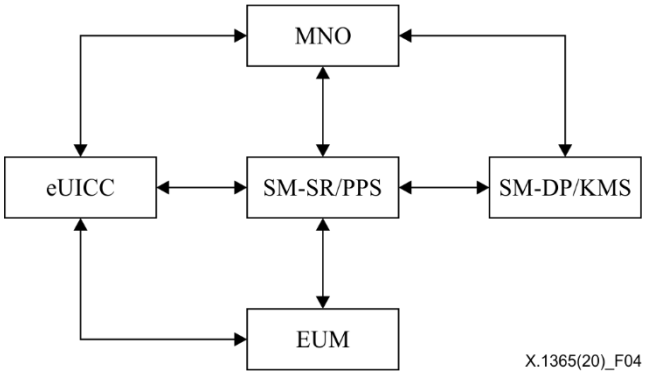

Figura 4 – Arquitectura B de gestión de claves de criptografía basada en la identidad, para dispositivos de IoT con UICC integrada

En la Figura 5 se muestra una arquitectura general para la utilización de la IBC en dispositivos IoT con UICC no integrada, con los componentes básicos siguientes:

- SecM: un módulo de seguridad (SecM) es un elemento que puede almacenar claves de forma segura y, con ellas, ejecutar mecanismos de seguridad para completar operaciones de seguridad. Los dispositivos IoT deben tener un SecM.
- IdP: un proveedor de identidad (IdP) es una entidad que crea, mantiene y gestiona información de identidad.
- AuC: un centro de autentificación (AuC) ofrece el servicio de autentificación de entidades.

El IdP utiliza el servicio de autentificación del AuC para autentificar los dispositivos IoT. Tras el proceso de autentificación inicial, el IdP presta el servicio de provisionamiento de identidades al SecM, que incluye funciones de creación, designación, sustitución y revocación de identidades. Tras crear y designar una identidad nueva para un dispositivo IoT, el IdP invoca el servicio de generación de clave privada que presta el KMS para generar la clave privada del ID recién asignado y distribuir las claves al SecM de forma segura. Además, extrae los parámetros públicos del KMS y los comunica al PPS, que publica los parámetros públicos a entidades externas. El IdP también puede prestar servicio de autentificación a otras entidades ejecutando protocolos de autentificación específicos con el SecM, entre ellos los definidos en la presente Recomendación.

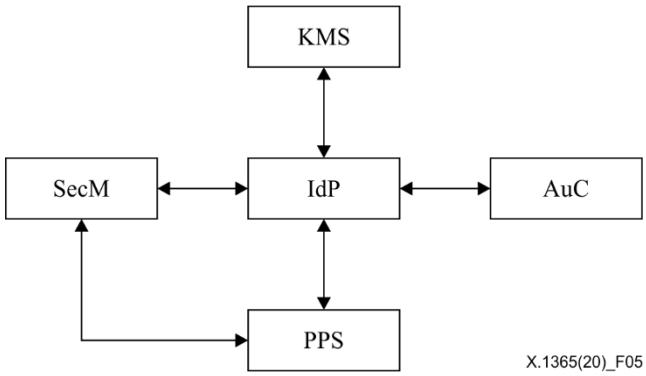

Figura 5 – Arquitectura de gestión de claves de criptografía basada en la identidad, para dispositivos de IoT con UICC no integrada

8.3 Denominación de identidades

Si se utiliza la tecnología de IBC para servicios de IoT a través de una red de telecomunicaciones, los nombres de las identidades pueden contener información útil que ayude a los operadores a gestionar la red. Se pueden integrar diferentes elementos de información (por ejemplo, tipo de servicio, ubicación, ID de dispositivo y hora de validez) en una identidad. La tecnología de IBC necesita una parte de esta información para su funcionamiento, como la hora de validez. Al disponer de la información de identidad, el operador puede optimizar la gestión de redes, por ejemplo, asignando una conexión a segmentos de red específicos en función del tipo de servicio. También resulta sencillo localizar un dispositivo basándose en su información de ubicación. En el Apéndice I se ofrece una definición de ejemplo de la identidad.

8.4 Gestión de claves

Además del valor de identidad, un sistema de IBC consta de tres tipos de valores de clave criptográfica: la MSK, los parámetros públicos y la clave privada. En el Anexo B se recoge la definición de notación de sintaxis abstracta uno (ASN.1) aplicable a estas estructuras de clave.

Para gestionar estas claves, el sistema de IBC se sirve de cinco operaciones de gestión de claves:

1) operación de inicialización de sistema;
2) inicialización de dispositivo;
3) búsqueda de parámetro público;
4) provisionamiento de identidad y clave;
5) revocación de identidad y clave.

El protocolo de interoperabilidad de la gestión de claves (KMIP) [b-OASIS KMIP] permite intercambiar mensajes entre la entidad de gestión y el KMS. No obstante, es preciso determinar la extensión del KMIP necesaria para satisfacer los requisitos nuevos de la función **IBSetup** y **IBExtract**. En los dispositivos IoT con eUICC, se utilizan procedimientos normalizados para el aprovisionamiento de claves a distancia [b-GSMA SGP.02]. En los dispositivos IoT con UICC no integrada, los protocolos de interacción entre los SecM y las entidades de gestión se definen de acuerdo con el protocolo de transferencia de hipertexto (HTTP). En el Anexo C se presentan las especificaciones de estas operaciones.

La operación de inicialización de sistema inicia un sistema de IBC mediante la generación de la MSK y los parámetros públicos. Se presupone que hay una entidad gestora, como el IdP, SM-DP o el MNO, responsable del proceso de inicialización del sistema de IBC. Esta entidad establece un canal seguro con una entidad KMS que ejecuta la función **IBSetup**. Las dos partes ejecutan el KMIP con la operación de creación de par de claves. La entidad gestora facilita al KMS la información necesaria para invocar la función **IBCSetup** y generar la MSK y los parámetros públicos. El KMIP se amplía para admitir funciones de configuración de diferentes elementos, por ejemplo, de algoritmos de IBC normalizados. En la cláusula C.1 se ofrece información detallada sobre esta operación.

La operación de inicialización de dispositivo prepara un dispositivo IoT para el aprovisionamiento de identidad y clave. Presenta dos variantes: inicialización de dispositivos IoT con eUICC e inicialización de dispositivos IoT con UICC no integrada. En la primera variante, la eUICC completa el registro en el encaminamiento seguro del gestor de abono (SM-SR) y queda preparada para la descarga de perfil [b-GSMA SGP.02]. No se requieren operaciones adicionales para los dispositivos con eUICC estándar. En la segunda variante, el SecM debe registrarse primero en el AuC para conseguir un ID de aprovisionamiento (PROV.ID) y una credencial de aprovisionamiento (PROV.CRED). Este par, PROV.ID/PROV.CRED, se utiliza para la autentificación de entidades durante el proceso de aprovisionamiento de identidad y clave. Adicionalmente, cuando los dispositivos IoT no pueden establecer un canal seguro con el IdP utilizando seguridad de la capa de transporte (TLS), durante el proceso de inicialización de dispositivo se debe instalar en el SecM una identidad de clave IdP.ID y una clave pública IdP.PUK conexa que pertenezcan al IdP o al parámetro público. En la cláusula C.2 se ofrece información detallada sobre la operación.

La operación de búsqueda de parámetro público permite recuperar los parámetros públicos de IBC. Un dispositivo IoT utilizará el procedimiento de aprovisionamiento de identidad y clave para obtener los parámetros públicos del sistema de IBC al que pertenece. Puede aplicar las especificaciones detalladas en la cláusula 4 de [IETF RFC 5408] para recuperar del PPS conocido los parámetros públicos de otros sistemas de IBC. En la cláusula C.3 se ofrece información detallada sobre la operación.

La operación de aprovisionamiento de identidad y clave incluye los procedimientos de asignación de identidad, extracción de clave privada y distribución de clave. Tras el proceso de inicialización, los dispositivos IoT solo tienen una identidad provisional. El IdP, SM-DP o MNO determinará qué entidad debe asignarse al dispositivo solicitante, se comunicará con el KMS para generar la clave privada correspondiente y, por último, distribuirá la identidad, la clave privada y los parámetros públicos al dispositivo de manera segura. En la cláusula C.4 se ofrece información detallada sobre la operación.

La operación revocación de identidad y clave se utiliza cuando una política de seguridad estricta exige que se revoque una identidad de manera oportuna. Si se revoca una identidad, se le asignará el estado revocado. Si una entrada consulta el estado de una identidad revocada, el IdP, SM-DP o MNO devolverá el valor correcto definido en el protocolo de estado de identidad en línea (OISP). Para comprobar un lote de estados de identidad de manera más eficiente, una entidad puede recuperar periódicamente la IRL del IdP, SM-DP o MNO y almacenarla en local, y utilizar la IRL más reciente para comprobar si una identidad está revocada sin consultar el estado en línea de cada identidad. En la cláusula C.5 se ofrece información detallada sobre la operación.

8.5 Autentificación

La autentificación es el proceso para determinar si una entidad (un dispositivo o un usuario) tiene derecho a acceder a unos recursos concretos. En las redes de telecomunicaciones, hay dos tipos de autentificación en relación con los dispositivos IoT: autentificación de acceso a red y autentificación a servicio. La autentificación de acceso a red establece si un dispositivo tiene permiso para acceder a la red, y la autentificación de acceso a servicio, si un dispositivo puede acceder a una ISP.

Dado que la IBC puede reducir en gran medida la carga de gestión de identidades y claves de un número elevado de dispositivos IoT, los protocolos de autentificación basados en tecnologías de IBC son adecuados para las autentificaciones de IoT en redes de telecomunicaciones. Otra ventaja de la IBC es que permite la autentificación distribuida, lo que no solo reduce de manera considerable el tiempo de autentificación, sino que también posibilita escenarios de aplicación nuevos, por ejemplo, la autentificación de dispositivo a dispositivo o de vehículo a vehículo. En las redes de telecomunicaciones actuales, como las redes 4G LTE, se puede utilizar la IBC para la autentificación entre dispositivos IoT e ISP. En las redes celulares 5G, la IBC puede aplicarse a la autentificación tanto de acceso a la red como de acceso al servicio. La especificación de seguridad 5G vigente, [b ETSI TS 133.501], define un marco de autentificación unificado que admite los métodos de protocolo de autentificación extensible (EAP). En el anexo de [b-ETSI TS 133.501] se explica con más detalle cómo utilizar el EAP-TLS en la 5G para redes de IoT.

El marco EAP es abierto y admite múltiples protocolos de autentificación, incluido el EAP-TLS. Los métodos de autentificación de EAP admiten claves tanto simétricas como asimétricas.

La IBC es una tecnología de claves públicas relativamente nueva que los protocolos de autentificación actuales no soportan bien. Por ello, en el Anexo D se presentan cuatro protocolos existentes que se han modificado para admitir la IBC durante la autentificación:

1) cláusula D.1: protocolo de transporte de clave secreta en un paso [ISO/CEI 11770-3];
2) cláusula D.2: TLS con clave pública original [IETF RFC 8446];
3) cláusula D.3: EAP-TLS [IETF RFC 5216];
4) cláusula D.4: EAP-PSK [IETF RFC 4764].

9 Requisitos de seguridad

En esta Recomendación solo se abordan los requisitos de seguridad para utilizar la IBC en IoT. Las amenazas y requisitos de seguridad generales para la IoT se detallan en [b-UIT-T X.1361]. Al tratarse de un criptosistema, las consideraciones de seguridad más importantes son la integridad y la autenticidad de las claves públicas y la confidencialidad de las claves secretas efímeras y a largo plazo. Un sistema de IBC está formado por los componentes siguientes: la MSK, los parámetros públicos, los ID, las claves privadas y los secretos efímeros utilizados en las operaciones criptográficas.

9.1 Requisitos de seguridad de la clave secreta maestra

Todas las claves privadas son generadas por la MSK. En particular, si se desvela la MSK, el adversario tiene la capacidad para recrear la clave privada de una entidad y, por tanto, puede desencriptar todos los mensajes protegidos con la clave pública correspondiente o suplantar cualquier entidad. Cualquier acceso ilícito a la MSK pondría en entredicho la seguridad del sistema de IBC. En consecuencia, la MSK se almacenará en un entorno reforzado, como un módulo de seguridad de *hardware* (HSM). Todos los accesos a la clave se autentificarán con mecanismos de alta seguridad.

9.2 Requisito de seguridad de los parámetros públicos

La clave pública se calcula a partir de los parámetros públicos y de un ID con la operación **IBDerivate**. Por tanto, la utilización de un conjunto falso de parámetros públicos generados por un adversario para encriptar un mensaje o verificar una firma pondría en peligro la confidencialidad del mensaje encriptado o conduciría a una conclusión falsa sobre el origen de una firma. En consecuencia, los parámetros públicos se transmitirán a través de un canal seguro o con una firma válida, y una entidad comprobará la entidad par del canal seguro o la validez de la firma con respecto a una clave pública segura antes de aceptar los parámetros públicos.

9.3 Requisito de seguridad del identificador

En la IoT, cada entidad posee un ID. Si se asigna el mismo ID a más de una entidad y se facilita la clave privada correspondiente a todas ellas, se pueden producir ataques de suplantación de identidad o una fuga de información sensible. Por ello, se asignará un ID único a cada dispositivo.

9.4 Requisito de seguridad de la clave privada

Cuando se compromete el entorno de seguridad de un dispositivo de ID, existe la posibilidad de que se filtre la clave privada. Por tanto, la clave privada se enviará a través de un canal seguro y se almacenará en un entorno seguro.

9.5 Requisito de seguridad de los secretos efímeros

Pueden producirse filtraciones de los secretos efímeros, como el secreto aleatorio que se utiliza en los procesos de encriptado o firma, cuando se compromete el entorno de seguridad de un dispositivo IoT. Por consiguiente, se garantizará la aleatoriedad de los secretos efímeros.

Anexo A

Formulación genérica y algoritmos de criptografía basada en identidad

(El presente anexo forma parte integrante de la presente Recomendación.)

En el presente anexo se ofrece una formulación genérica de la IBC y una lista de los algoritmos de la IBC admitidos en esta Recomendación. Los algoritmos que siguen esa formulación genérica, pero que no figuran en la lista, pueden incorporarse fácilmente en el futuro como extensiones a este marco. La formulación genérica especificada aquí también sirve de guía a las descripciones de estructuras de datos clave relacionadas, a las operaciones de gestión de claves, así como a los protocolos de autenticación y de establecimiento de claves definidos en los Anexos B a D.

Un criptosistema IBC incluye los siguientes tipos de datos clave, cuya categorización de esas claves se ajusta a la [ISO/CEI 18033-5].

– *ib.msk*: la MSK es el valor secreto utilizado por el KMS para calcular una clave privada basada en la identidad. *ib.msk* se genera durante el proceso de inicialización del sistema y solo es conocido por el KMS.

– *ib.mpk*: la MPK, que se determina unívocamente por la MSK correspondiente. *ib.mpk* se calcula por el KMS durante el proceso de inicialización del sistema.

– *ib.sysparam*: los parámetros del sistema para la computación criptográfica, incluida una selección de un esquema criptográfico o una función particular de una familia de esquemas criptográficos o de funciones, o de una familia de espacios matemáticos. *ib.sysparam* es elegido por el KMS durante el proceso de inicialización del sistema.

– *ib.pubparam*: los parámetros públicos son la combinación de los parámetros del sistema *ib.sysparam* con la MPK *ib.mpk*. Este tipo de clave se define para proporcionar una visión unificada entre las Normas Internacionales tales como [ISO/CEI 18033-5], y RFC relacionadas con la IBC, como [IETF RFC 5091].

– *ib.prk*: la clave privada basada en la identidad, que es generada por el KMS con *ib.msk* e *ib.pubparam*, correspondiente a un identificador *ID.*

– *ib.pub*: la clave pública basada en la identidad, que se calcula a partir de un identificador *ID* y de *ib.pubparam* mediante una función definida por un esquema criptográfico basado en la identidad.

Un criptosistema IBC puede incluir las siguientes funciones que se especifican con entradas y salidas.

IBSetup

entrada: parámetro de seguridad

salida: *ib.pubparam, ib.msk*

IBExtract

entrada: *ib.pubparam, ib.msk, ID*

salida: *ib.prk*

IBDerivate

entrada: *ib.pubparam, ID*

salida: *ib.puk*

IBEnc

entrada: *ib.pubparam, ID, mensaje M*

salida: texto cifrado *C*

IBDec

entrada: *ib.pubparam, ID, ib.prk,* texto cifrado C

salida: plantext *M* o error

IBSign

entrada: *ib.pubparam, ID, ib.prk,* mensaje *M*

salida: firma *S*

IBVerify

Entrada: *ib.pubparam, ID,* mensaje *M,* firma *S*

Salida: válida o no válida

La presente Recomendación soportará el uso de los siguientes algoritmos basados en la identidad:

– BB1-KEM (mecanismo de encapsulado de claves, KEM) [IETF RFC 5091].
– BF-IBE [IETF RFC 5091].
– SK-KEM [IETF RFC 6508].
– SM9-IBE [b-GM/T 0044.2].
– Cha-Cheon-IBS (IBS2) [ISO/CEI 14888-3].
– ECCSI (firmas sin certificado basadas en curvas elípticas para cifrado basado en identidad) [IETF RFC 6507].
– Hess-IBS (IBS1) [ISO/CEI 14888-3].
– SM9-IBS (IBS china) [ISO/CEI 14888-3].
– Fujioka-Suzuki-Ustaoglu-AKA (acuerdo de clave autenticada, AKA) [ISO/CEI 11770-3].
– Smart-Chen-Cheng-AKA [ISO/CEI 11770-3].
– SM9-AKA [b-GM/T 0044.2].
– Wang-AKA [b-IEEE P1363.3].

Todos esos algoritmos se basan en la suposición del logaritmo discreto y normalmente se implementan en el grupo de puntos sobre una curva elíptica. Muchos de estos algoritmos utilizan además un "emparejamiento criptográfico" sobre una curva elíptica [b-Galbraith]. Un emparejamiento criptográfico *e* es un mapa bilineal eficientemente computable e: $G1 \times G2 \rightarrow G3$, que satisface la ecuación:

$$e([a]P1, [b]P2) = e(P1, P2)^{a*b}$$

donde P1 y P2 son los generadores del grupo *G*1 y *G*2, respectivamente. [*a*]P1 denota el número *a* de veces de operaciones de grupo con P1, del mismo modo [*b*]P2 es la operación de grupo con P2.

Un emparejamiento criptográfico puede ser creado como instancia por el emparejamiento Weil, emparejamiento Tate, emparejamiento óptimo Ate, etc., sobre curvas elípticas fáciles de emparejar [b-Freeman]. Algunas de las curvas elípticas de fácil emparejamiento utilizadas habitualmente son las curvas elípticas de extrema singularidad, curvas Barreto-Naehrig (BN), curvas Barreto-Lynn-Scott con un grado de inserción de 12 (BLS-12), curvas Kachisa-Schaefer-Scott con un grado de inserción de 16 (KSS-16), curvas Kachisa-Schaefer-Scott con un grado de inserción de 18 (KSS-18) y curvas Barreto-Lynn-Scott con un grado de inserción de 24 (BLS-24). Todas esas

curvas *E* se basan en un campo primo, un campo finito de la característica primo p, F_p, donde p es un entero primo. *G*1 es el subgrupo de puntos en la curva *E*. *G*2 es el mismo que *G*1 si se utilizan las curvas de extrema singularidad o es un subgrupo de puntos en la curva de torsión *E*. *E* se construye a partir de algún campo de extensión del campo base F_p. *G*3 es la extensión de campo F_p^k del campo base F_p, donde k es el grado de inserción.

Hay algoritmos IBC que se construyen con otros mecanismos matemáticos como retículas, por ejemplo, [b-Ducas]. Ese tipo de algoritmo es eficiente en cuanto a la computación, al tiempo que tiene un tamaño de clave y salida mayor que los basados en el logaritmo discreto sobre curvas elípticas. Se cree que los algoritmos son resistentes a los ataques que se ejecutan en ordenadores cuánticos. Con todo, los algoritmos de esa categoría todavía están en fase de desarrollo. Por lo tanto, parece prematuro tenerlos presentes para la estandarización, si bien esos algoritmos IBC basados en retículas pueden tenerse en cuenta para una futura inclusión.

Anexo B

Especificación de datos clave de la criptografía basada en la identidad

(El presente anexo forma parte integrante de la presente Recomendación.)

Utilizando el método estándar ASN.1, [IETF RFC 5408] ha definido una estructura genérica para los parámetros del sistema, incluyendo *ib.pubparam* y otra información auxiliar, y [IETF RFC 5091] ha definido dos conjuntos de estructuras de datos clave, incluyendo *ib.msk* e *ib.prk* para dos algoritmos IBE, es decir, BF-IBE y BB1-IBE. Aunque mantiene la compatibilidad con las definiciones vigentes, esta Recomendación amplía la definición de los parámetros del sistema y define nuevas estructuras de datos clave para soportar más algoritmos y diversas implementaciones eficientes con diferentes curvas y emparejamientos.

A continuación, se define una estructura genérica de parámetros del sistema:

```
IBSysParams ::= SEQUENCE {
        version                 INTEGER { v3(3) },
        domainName              IA5String,
        domainSerial            INTEGER,
        validity                ValidityPeriod,
        ibPublicParameters      IBPublicParameters,
        ibIdentityType          OBJECT IDENTIFIER,
        ibParamExtensions       [0] IMPLICIT IBParamExtensions OPTIONAL,
        signatureAlgorithm      [1] IMPLICIT AlgorithmIdentifier OPTIONAL,
        signature               [2] IMPLICIT BIT STRING OPTIONAL
}
```

IBSysParams corresponde a la definición IBESysParams en [IETF RFC 5408], pero la versión se modifica a v3 (3) y se añaden dos campos adicionales. districtName y districtSerial han sido renombrados como domainName y domainSerial, respectivamente. La definición de IBPublicParameter ha sido modificada del tipo OCTET STRING al tipo recién definido IBParameterData, que es una CHOICE determinada por el valor de pkgAlgorithm. Esa definición elimina la doble codificación innecesaria causada por la definición anterior, a saber, la codificación de publicParameterData como una SEQUENCE de, por ejemplo, BFPublicParameters y la codificación adicional del resultado como una OCTET STRING. Excepto los dos nuevos campos, el significado de los otros campos permanece sin cambios como en [IETF RFC 5408]. Los significados de los dos nuevos campos son los siguientes:

– *Signature Algorithm* designa el algoritmo de firma utilizado para generar el valor de firma. Este campo es opcional, ya que el campo de firma no es obligatorio.

– El campo de firma contiene la firma digital calculada sobre la base de las normas de codificación distinguida (DER) ASN.1 resultantes de la versión de campo de *ibParamExtensions*. Este campo está codificado como BIT STRING y es opcional.

Si se presenta, el campo de firma se utiliza para ayudar a una entidad a comprobar la autenticidad de los parámetros públicos del sistema sin recurrir a otros métodos. Por ejemplo, si un dispositivo de IoT no tiene la capacidad de establecer un canal seguro basado en TLS como se requiere en [IETF RFC 5408] para recuperar los parámetros públicos de otro sistema IBC, puede consultar su PPS con HTTP. En ese caso, el PPS servidor firmará los parámetros públicos solicitados con su clave privada

de firma. El dispositivo de IoT puede verificar la firma para comprobar la autenticidad de la respuesta. Si un PPS está publicando los parámetros públicos de otro sistema IBC a sus entidades servidoras, se recomienda que el mensaje de firma se trate como una identidad y que el algoritmo **IBExtract** se utilice como algoritmo de firma para generar la clave privada como el valor de firma correspondiente. De ese modo, los dispositivos de IoT verifican si el valor de la firma es una clave privada válida que corresponde al resultado DER ASN.1 desde la versión de campo *ibParamExtensions* y no necesita una clave pública de verificación adicional para verificar la firma.

```
ValidityPeriod ::= SEQUENCE {
        notBefore               GeneralizedTime,
        notAfter                GeneralizedTime
}
IBPublicParameters ::= SEQUENCE SIZE (1..MAX) OF IBPublicParameter
IBPublicParameter ::= SEQUENCE {
        pkgAlgorithm            OBJECT IDENTIFIER,
        publicParameterData     IBParameterData
}
```

El valor de publicParameterData se define por pkgAlgorithm. Puede ser una de las siguientes opciones.

```
IBParameterData ::= CHOICE {
        bb1ParameterData        [0] IMPLICIT BB1PublicParameters,
        bfParameterData         [1] IMPLICIT BFPublicParameters,
        eccsiParameterData      [2] IMPLICIT ECCSIPublicParameters,
        skParameterData         [3] IMPLICIT SKPublicParameters,
        sm9ParameterData        [4] IMPLICIT SM9PublicParameters
}
IBParamExtensions ::= SEQUENCE OF IBParamExtension
IBParamExtension ::= SEQUENCE {
        ibParamExtensionOID     OBJECT IDENTIFIER,
        ibParamExtensionValue   OCTET STRING
}
AlgorithmIdentifier ::= SEQUENCE {
        algorithm               OBJECT IDENTIFIER,
        parameters              ANY DEFINED BY algorithm OPTIONAL
}
```

En [IETF RFC 5091], dos conjuntos de MSK, parámetros públicos y bloque de claves privadas, es decir:

– BB1MasterSecret, BB1PublicParameters, BB1PrivateKeyBlock;

– BFMasterSecret, BFPublicParameters, BFPrivateKeyBlock están definidos para las funciones de generación de claves BF y BB1. Esas designaciones solo encajan con las implementaciones de las funciones con emparejamientos simétricos en curvas elípticas de extrema singularidad definidas sobre campos primos. En la presente Recomendación se especifican nuevas estructuras con versión modificada a v3 para soportar implementaciones de esos algoritmos con emparejamientos asimétricos. Para los emparejamientos simétricos sobre curvas elípticas de extrema singularidad, el campo correspondiente en las estructuras de datos clave BB1 y BF permanece inalterable como en [IETF RFC 5091]. Se definen tres conjuntos más de estructuras de datos clave para ECCSI, SM9 y SK-KEM, respectivamente.

```
BB1MasterSecret ::= SEQUENCE {
        version         INTEGER { v3(3) },
        alpha           INTEGER,
        beta            INTEGER,
        gamma           INTEGER
}
```

Para las implementaciones con emparejamientos asimétricos, alfa será s1, beta será s2 y gamma será s3 en la cláusula 9.3 de [ISO/CEI 18033-5].

```
BB1PublicParameters ::= SEQUENCE {
        version         INTEGER { v3(3) },
        curve           OBJECT IDENTIFIER,
        hashfcn         OBJECT IDENTIFIER,
        pairing         PAIRING OPTIONAL,
        p               INTEGER OPTIONAL,
        q               [0] IMPLICIT INTEGER OPTIONAL,
        pointP          FpPoint,
        pointQ          [1] EXPLICIT FpxPoint OPTIONAL
        pointP1         FpPoint,
        pointP2         [2] EXPLICIT FpxPoint OPTIONAL,
        pointP3         FpPoint,
        v               FpxElement
}
```

– El emparejamiento especifica qué tipo de mapa bilineal se utilizará con los parámetros generados. Se admiten tres tipos de emparejamiento: emparejamiento de Weil, emparejamiento de Tate y emparejamiento óptimo de Ate.

– p y q se convierten en opcionales. Para algunos tipos de curvas, como BN, BLS-12, etc., p y q están predeterminados por identificadores de objeto de curva (OID) y, por lo tanto, no es necesario volver a especificarlos.

– pointP y pointQ, para la realización con emparejamientos asimétricos, serán Q1 en *G*1 y Q2 en *G*2 en la cláusula 9.3 de [ISO/CEI 18033-5]. Para los emparejamientos simétricos, pointP es igual a pointQ, por lo que pointQ es OPTIONAL.
– pointP1 y pointP3, para la implementación con emparejamientos asimétricos, serán R y T en la cláusula 9.3 de [ISO/CEI 18033-5].
– pointP2, para la implementación con emparejamientos asimétricos como el emparejamiento óptimo de Ate sobre curvas BN, toma valor de un campo de extensión de F_p. pointP2 es opcional porque si se da v, pointP2 es innecesario para que el algoritmo BB1-KEM lo lleve a cabo.
– v es el resultado del emparejamiento, que es un elemento del campo de extensión de F_p. Para la implementación con emparejamientos asimétricos, como el emparejamiento óptimo de Ate sobre curvas BN, el campo de extensión es F_p^k, donde k es el grado de inserción. En este caso, v será J en la cláusula 9.3 de [ISO/CEI 18033-5].
– El significado de otros campos permanece inalterado como en [IETF RFC 5091].

```
PAIRING ::= ENUMERATED {
        weil        (1),    --Weil pairing
        tate        (2),    --Tate pairing
        optimalAte  (3)     --Optimal Ate pairing
}
FpPoint ::= SEQUENCE {
        x       INTEGER,
        y       INTEGER
}
```

FpPoint define un punto en una curva elíptica sobre un campo primo. Un punto tiene dos coordenadas, que se designan como la coordenada x y la coordenada y. Las dos coordenadas toman valores enteros grandes.

```
FpxPoint ::= CHOICE {
        fpPoint     [1] EXPLICIT FpPoint,
        fp2Point    [2] EXPLICIT Fp2Point,
        fp3Point    [3] EXPLICIT Fp3Point,
        fp4Point    [4] EXPLICIT Fp4Point
}
```

– Fp2Point define un punto en una curva elíptica sobre un campo F_p^2. Cada coordenada de un punto toma valor de un elemento de F_p^2.
– Fp3Point define un punto en una curva elíptica sobre un campo F_p^3. Cada coordenada de un punto toma valor de un elemento de F_p^3.
– Fp4Point define un punto en una curva elíptica sobre un campo F_9p^4. Cada coordenada de un punto toma valor de un elemento de F_p^4.

```
Fp2Point ::= SEQUENCE {
        x       Fp2Element,
        y       Fp2Element
}
```

– Fp2Point define un punto en una curva elíptica sobre un campo F_p^2. Un punto tiene dos coordenadas que se denominan coordenada x y coordenada y. Las dos coordenadas toman valores de F_p^2.

```
Fp3Point ::= SEQUENCE {
        x       Fp3Element,
        y       Fp3Element
}
```

– Fp3Point define un punto en una curva elíptica sobre un campo F_p^3. Las dos coordenadas de un punto toman valores de F_p^3.

```
Fp4Point ::= SEQUENCE {
        x       Fp4Element,
        y       Fp4Element
}
```

– Fp4Point define un punto en una curva elíptica sobre un campo F_p^4. Las dos coordenadas de un punto toman valores de F_p^4.

```
Fp2Element ::= SEQUENCE {
        a       INTEGER,
        b       INTEGER
}
```

– Fp2Element define un elemento de un campo F_p^2, que se representa como a+bα donde α es raíz no cuadrática en F_p.

```
Fp3Element ::= SEQUENCE {
        a       INTEGER,
        b       INTEGER,
        c       INTEGER
}
```

– Fp3Element define un elemento de un campo F_p^3, que se representa como a+bβ+cβ^2 donde β es raíz no cúbica en F_p.

```
Fp4Element ::= SEQUENCE {
        a       Fp2Element,
        b       Fp2Element
}
```

– Fp4Element define un elemento de un campo F_p^4, que se representa como torre de dos elementos de F_p^2.

```
FpxElement ::= CHOICE {
        fp2Elemt        [1] EXPLICIT Fp2Element,
                        --para implementación de curva elíptica de extrema singularidad
        fp12Elemt       [2] EXPLICIT Fp12Element,
                        --utilizando representación de torre F_p->F_p^2->F_p^6->F_p^12
```

fp16Elemt [3] EXPLICIT Fp16Element,

--utilizando representación de torre F_p->F_p^2->F_p^4->F_p^8->F_p^16

fp18Elemt [4] EXPLICIT Fp18Element,

--utilizando representación de torre F_p->F_p^3->F_p^6->F_p^18

fp24Elemt [5] EXPLICIT Fp24Element

--utilizando representación de torre F_p->F_p^2->F_p^6->F_p^12->F_p^24

}

– FpxElement define la representación de torre de un elemento en *G*3. El emparejamiento *e* asigna dos entradas de *G*1 y *G*2, respectivamente, a un elemento en *G*3. Para las curvas de emparejamiento utilizadas habitualmente, los elementos en *G*3 se representan normalmente en un método de torre. Para diferentes grados de inserción, puede haber diferentes representaciones de torres. En la presente Recomendación se define una representación de torre utilizada habitualmente de elementos en campo con grados de inserción 12, 16, 18 y 24.

Fp12Element ::= SEQUENCE {

a Fp6Element,

b Fp6Element

}

– Fp12Element define un elemento de F_p^12 con una representación de torre de 2x3x2 y se utilizará en la implementación con curvas BN o curvas BLS-12 o curvas BLS-24.

Fp6Element ::= SEQUENCE {

a Fp2Element,

b Fp2Element,

c Fp2Element

}

– Fp6Element define un elemento de F_p^6 con una representación de torre 3x2 y se utilizará en la implementación con curvas BN o curvas BLS-12 o curvas BLS-24.

Fp16Element ::= SEQUENCE {

a Fp8Element,

b Fp8Element

}

– Fp16Element define un elemento de F_p^16 con una representación de torre de 2x2x2x2 y se utilizará en la implementación con curvas KSS-16.

Fp8Element ::= SEQUENCE {

a Fp4Element,

b Fp4Element

}

– Fp8Element define un elemento de F_p^8 con una representación de torre de 2x2x2 y se utilizará en la implementación con curvas KSS-16.

Fp18Element ::= SEQUENCE {

a Fp6bElement,

b Fp6bElement,

c Fp6bElement

}

– Fp18Element define un elemento de F_p^18 con una representación de torre de 3x2x3 y se utilizará en la implementación con curvas KSS-18.

Fp6bElement ::= SEQUENCE {

a Fp3Element,

b Fp3Element

}

– Fp6bElement define un elemento de F_p^6 con una representación de torre 2x3 y se utilizará en la implementación con curvas KSS-18.

Fp24Element ::= SEQUENCE {

a Fp12Element,

b Fp12Element

}

– Fp24Element define un elemento de F_p^24 con una representación de torre de 2x2x3x2 y se utilizará en la implementación con curvas BLS-24.

BB1PrivateKeyBlock ::= SEQUENCE {

version INTEGER { v3(3) },

pointD0 FpxPoint,

pointD1 FpxPoint

}

– Los significados de pointD0 y pointD1 permanecen inalterados como en [IETF RFC 5091], pero se toman de *G*2 si BB1-KEM se implementa con emparejamientos asimétricos. En este caso, pointD0 y pointD1 serán dID0 y dID1 respectivamente en la cláusula 9.3 de [ISO/CEI 18033-5].

BFMasterSecret ::= SEQUENCE {

version INTEGER {v3(3) },

masterSecret INTEGER

}

– El significado de otros campos permanece inalterado como en [IETF RFC 5091].

BFPublicParameters ::= SEQUENCE {

version INTEGER { v3(3) },

curve OBJECT IDENTIFIER,

hashfcn OBJECT IDENTIFIER,

pairing PAIRING OPTIONAL,

p INTEGER OPTIONAL,

```
    q           [0] IMPLICIT INTEGER OPTIONAL,
    pointP      FpxPoint,
    pointPpub   FpxPoint
}
```

– El significado de cada campo permanece inalterado como en [IETF RFC 5091], pero pointP y pointPpub se toman de *G*2 si BF-IBE se implementa con emparejamientos asimétricos. En este caso, los puntos P y Ppub serán Q y R, respectivamente, en la cláusula 8.2 de [ISO/CEI 18033-5].

```
BFPrivateKeyBlock ::= SEQUENCE {
    version     INTEGER { v3(3) },
    privateKey  FpPoint
}
```

– Los significados de cada campo permanecen inalterados como en [IETF RFC 5091]. Para implementaciones con emparejamientos asimétricos, privateKey será skID en la cláusula 8.2 de [ISO/CEI 18033-5].

```
ECCSIMasterSecret ::= SEQUENCE {
    version     INTEGER {v3(3) },
    masterSecret INTEGER
}
```

– masterSecret será KSAK en [IETF RFC 6507].

```
ECCSIPublicParameters ::= SEQUENCE {
    version     INTEGER { v2(2) },
    curve       OBJECT IDENTIFIER,
    hashfcn     OBJECT IDENTIFIER,
    pointP      FpPoint,
    pointPpub   FpPoint
}
```

– pointP será G en [IETF RFC 6507].

– pointPpub será la clave de autenticación pública KMS (KPAK) en [IETF RFC 6507].

```
ECCSIPrivateKeyBlock ::= SEQUENCE {
    version     INTEGER { v2(2) },
    ssk         INTEGER ,
    pvt         OCTET STRING
}
```

– ssk y pvt serán una clave de firma secreta (SSK) y un testigo de verificación pública (PVT) en [IETF RFC 6507], respectivamente.

```
SKMasterSecret ::= SEQUENCE {
        version         INTEGER {v3(3) },
        masterSecret    INTEGER
}
```

– masterSecret será z_T en [IETF RFC 6508] y s en la cláusula 9.2 de [ISO/CEI 18033-5].

```
SKPublicParameters ::= SEQUENCE {
        version         INTEGER { v3(3) },
        curve           OBJECT IDENTIFIER,
        hashfcn         OBJECT IDENTIFIER,
        pairing         PAIRING OPTIONAL,
        p               INTEGER OPTIONAL,
        q               [0] IMPLICIT INTEGER OPTIONAL,
        pointP1         FpPoint,
        pointP1pub      [1] EXPLICIT FpPoint OPTIONAL,
        pointP2         [2] EXPLICIT FpxPoint OPTIONAL,
        pointP2pub      [3] EXPLICIT FpxPoint OPTIONAL,
        v               [4] EXPLICIT FpxElement
}
```

– Para implementaciones con emparejamientos simétricos sobre curvas de extrema singularidad, p y q se definen en [IETF RFC 5091]. Para implementaciones con emparejamientos asimétricos, p y q están predeterminados por la curva utilizada y se convierten en opcionales.

– pointP1 será P en [IETF RFC 6508] y Q1 en *G*1 en la cláusula 9.2 de [ISO/CEI 18033-5].

– pointP1pub será Z_T en [IETF RFC 6508] y R en la cláusula 9.2 de [ISO/CEI 18033-5]. pointP1pub puede ser innecesario para otros algoritmos, como los algoritmos de firma, basados en la función de generación Sakai-Kasahara (SK), por lo que es opcional.

– pointP2 será Q2 en *G*2 en la cláusula 9.2 de [ISO/CEI 18033-5] si el SK-KEM se implementa con emparejamientos asimétricos. pointP2 no es necesario para que el SK-KEM lo lleve a cabo, por lo que es opcional.

– pointP2pub será [*ib.msk*]Q2, que es innecesario para SK-KEM, pero puede ser necesario para otros algoritmos, como los algoritmos de firma, basados en la función de generación de claves SK, por lo que es opcional.

```
SKPrivateKeyBlock ::= SEQUENCE {
        version         INTEGER { v3(3) },
        privateKey      FpxPoint
}
```

– privateKey será RSK en [IETF RFC 6508] y skID en la cláusula 9.2 de [ISO/CEI 18033-5].

```
SM9MasterSecret ::= SEQUENCE {
        version         INTEGER {v3(3) },
        masterSecret    INTEGER
}
```

– masterSecret será *ib.msk* que es U, definida en la cláusula 7.4 de [b-ISO/CEI 14888-3a].

```
SM9PublicParameters ::= SEQUENCE {
        version         INTEGER { v3(3) },
        curve           OBJECT IDENTIFIER,
        hashfcn         OBJECT IDENTIFIER,
        pairing         PAIRING OPTIONAL,
        p               INTEGER OPTIONAL,
        q               [0] IMPLICIT INTEGER OPTIONAL,
        pointP1         FpPoint,
        pointP1pub      [1] EXPLICIT FpPoint OPTIONAL,
        pointP2         [2] EXPLICIT FpxPoint OPTIONAL,
        pointP2pub      [3] EXPLICIT FpxPoint OPTIONAL,
        v               [4] EXPLICIT FpxElement
}
```

– Para implementaciones con emparejamiento simétrico sobre curvas de extrema singularidad, p y q son como se definen en [IETF RFC 5091]. Para implementaciones con emparejamientos asimétricos, p y q están predeterminados por la curva utilizada.

– pointP1 será P en la cláusula 7.4 de [ISO/CEI 14888-3].

– pointP1pub es innecesario para SM9-IBS, pero necesario para SM9-IBE y en este caso pointP2pub será [*ib.msk*]P.

– pointP2 será Q en la cláusula 7.4 de [ISO/CEI 14888-3]. pointP2 no es necesario para que **SM9-IBE** lo lleve a cabo, por lo que es opcional.

– pointP2pub será V en la cláusula 7.4 de [ISO/CEI 14888-3a]. pointP2pub no es necesario para que **SM9-IBE** lo lleve a cabo, por lo que es opcional.

```
SM9PrivateKeyBlock ::= SEQUENCE {
        version         INTEGER { v3(3) },
        privateKey      FpxPoint
}
```

– privateKey será X en la cláusula 7.4 de [ISO/CEI 14888-3a] para la firma y será el *ib.prvk* en *G*1 para SM9-IBE y SM9-AKA.

La definición de BFMasterSecret, BFPublicParameters y BFPrivateKeyBlock se utilizará para los algoritmos que utilicen la generación de claves Sakai-Ohgishi-Kasahara (SOK), tales como BF-IBE, Cha-Cheon-IBS, Hess-IBS, Fujioka-Suzuki-Ustaoglu-AKA, Smart-Chen-Cheng-AKA y Wang-AKA. La definición de BB1MasterSecret, BB1PublicParameters y BB1PrivateKeyBlock se utilizará para los algoritmos que utilicen la generación de claves BB1, como por ejemplo, BB1-KEM. Los parámetros de SKMasterSecret, SKPublicParameters y SKPrivateKeyBlock se utilizarán para SK-KEM y posiblemente para otros algoritmos basados en la función de generación de claves SK.

SM9MasterSecret, SM9PublicParameters y SM9PrivateKeyBlock se utilizarán para los algoritmos SM9, incluidos SM9-IBE, SM9-IBS y SM9-AKA. ECCSIMasterSecret, ECCSIPublicParameters y ECCSIPrivateKeyBlock se utilizarán para ECCSI.

Si es necesario proteger la clave privada, se utilizará la estructura EncryptedPrivateKeyInfo definida en [IETF RFC 5958].

```
EncryptedPrivateKeyInfo ::= SEQUENCE {
        encryptionAlgorithm  EncryptionAlgorithmIdentifier,
        encryptedData        EncryptedData
}
EncryptionAlgorithmIdentifier ::= AlgorithmIdentifier
EncryptedData ::= OCTET STRING
AlgorithmIdentifier ::= SEQUENCE {
        algorithm            OBJECT IDENTIFIER,
        parameters           ANY DEFINED BY algorithm OPTIONAL
}
```

Anexo C

Operaciones de gestión de claves

(El presente anexo forma parte integrante de la presente Recomendación.)

En los sistemas de IBC, las operaciones de gestión de claves son la inicialización de sistema, el aprovisionamiento de identidad y clave privada, la revocación de identidad y clave privada y la publicación de parámetros de sistema. La inicialización de sistema requiere la invocación de la función **IBSetup**, y el aprovisionamiento de clave privada se basa en la invocación de la función **IBExtract**. Estas operaciones precisan de la interacción entre una entidad de gestión y el KMS. En la presente Recomendación se utiliza el KMIP para el intercambio de mensajes entre estas dos partes. En el Apéndice II se establece la extensión del protocolo necesaria para satisfacer los nuevos requisitos de los algoritmos **IBSetup** e **IBExtract** admitidos. Los protocolos para la interacción entre los SecM y las entidades de gestión se definen con arreglo al HTTP para dispositivos IoT con UICC no integrada. Para las eUICC, se utilizan las normas de [b-GSMA SGP.02], ampliadas cuando proceda.

C.1 Inicialización de sistema

En cualquier sistema de IBC, es obligatorio completar un proceso de inicialización de sistema antes de prestar KMS a sus usuarios. Durante el proceso, el KMS ejecuta una o más funciones **IBSetup** para generar uno o más conjuntos de pares de clave *ib.msk* e *ib.pubparam*. El método para reforzar la seguridad del KMS es ajeno al ámbito de la presente Recomendación. Una buena práctica consiste en generar y almacenar la clave *ib.msk* en un HSM. Siempre que sea posible, se preparará un esquema de generación de claves distribuidas que aplique un esquema de compartición de secretos para dividir *ib.msk* y propagar la función de compartición de secretos y generación de claves privadas a varios KMS. En este caso, solo se podrá generar correctamente una clave privada para un ID si el número de KMS en correcto funcionamiento supera un umbral definido.

Véase la Figura C.1.

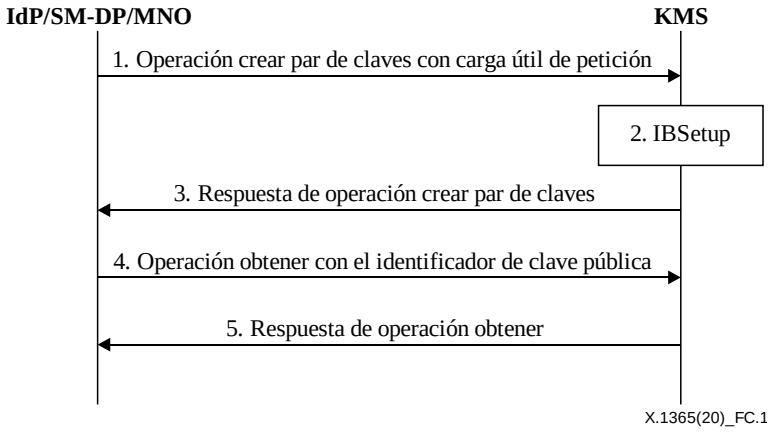

Figura C.1 – Inicialización de sistema con protocolo de interoperabilidad para la gestión de claves

Condiciones iniciales

Se presupone que el IdP/SM-DP/MNO actúa como iniciador de sistema y se encarga del proceso de inicialización de sistema. Para que el IdP/SM-DP/MNO pueda invocar la función **IBSetup** en un KMS, se deberán satisfacer las condiciones siguientes:

a) Se ha establecido un canal seguro entre el IdP/SM-DP/MNO y el KMS.

b) El IdP/SM-DP/MNO ha completado un proceso de autentificación con el KMS, y el IdP/SM-DP/MNO autentificado tiene autorización para ejecutar la petición **IBSetup**.

Procedimiento

1) El IdP/SM-DP/MNO preparará la carga útil de petición e invocará la operación crear par de claves para enviar un mensaje de petición codificado al KMS.
2) El KMS comprobará si la petición es válida y el IdP/SM-DP/MNO tiene autorización para invocar esta operación. Si no se cumple alguna de estas condiciones, el KMS enviará una respuesta con una indicación de fallo. En caso contrario, el KMS ejecutará **IBSetup** con los parámetros especificados en la petición.
3) El KMS devolverá la respuesta de ejecución al IdP/SM-DP/MNO. Si la operación se completa correctamente, el KMS devolverá como mínimo un ID único de clave privada a *ib.msk* y un ID único de clave pública a *ib.pubparam*.
4) Alternativamente, si la operación crear par de claves se completa correctamente, el IdP/SM-DP/MNO podrá invocar la operación obtener con el ID único de clave pública proporcionado por la última respuesta a fin de recuperar los parámetros públicos *ib.pubparam*.
5) El KMS devolverá el valor de clave de los parámetros públicos que se acaban de generar.

La extensión del KMIP como apoyo a esta operación se trata en el Apéndice II.

Condición final: El KMS se inicia correctamente y el IdP/SM-DP/MNO conoce el ID único de clave privada y el ID único de clave pública necesarios para acceder al valor *ib.msk* y los parámetros públicos *ib.pubparam* de la MSK generada, respectivamente. El IdP/SM-DP/MNO utilizará el ID único de clave privada para llamar a la operación firmar a fin de generar claves privadas de identidad, y el ID único de clave pública para llamar a la operación obtener y así recuperar los parámetros públicos.

C.2 Inicialización de dispositivo

La operación inicializar dispositivo consiste en preparar el dispositivo para el aprovisionamiento de identidad y clave. Los dispositivos IoT con eUICC y los dispositivos IoT con UICC no integrada utilizan procedimientos de inicialización de dispositivo diferentes.

C.2.1 Caso 1: Inicialización para eUICC

En las eUICC, la identidad y clave privada correspondiente *ib.prk* y los parámetros públicos *ib.sysparam* se descargan a un perfil de dominio de seguridad del expedidor (ISD). Por consiguiente, tras el proceso de inicialización de dispositivo, la eUICC debería estar preparada para crear el perfil del ISD. La operación de registro se completará con arreglo a [b-GSMA SGP.02]. A continuación, se reproduce la cláusula 3.5.1 de [b-GSMA SGP.02].

- **Registro de la eUICC en el SM-SR**

Condición inicial

a) Se producen las eUICC y se carga y activa un perfil de aprovisionamiento en la red del operador de aprovisionamiento. Las eUICC están probadas y listas para su envío. Cada eUICC tiene un conjunto de información de eUICC (EIS) propio.

Procedimiento

1) El fabricante de eUICC (EUM) envía una petición de registro de eUICC al SM-SR seleccionado, que contiene el EIS.
2) El SM-SR almacena el EIS en su base de datos, con el identificador de eUICC (EID) como parámetro de clave.

3) El SM-SR confirma al EUM que el registro se ha completado con éxito. El mensaje de confirmación incluye el EID.

Condición final: La eUICC está registrada en el SM-SR y lista para la descarga del perfil. Ya se puede enviar al fabricante de dispositivos máquina a máquina.

C.2.2 Caso 2: Inicialización para dispositivos IoT con UICC no integrada

En el caso de los dispositivos IoT con UICC no integrada, se completará la operación de registro siguiente:

- **Registro del SecM en el AuC**

Condición inicial

a) El SecM se genera y el dispositivo IoT podrá comunicarse con el IdP en la red del operador.

Procedimiento

1) El SecM envía una petición de adquisición de datos de aprovisionamiento del SecM al AuC.
2) El AuC genera un ID de aprovisionamiento (PROV.ID) y una credencial de autentificación (PROV.CRED) asociada para el SecM solicitante.
3) El AuC envía PROV.ID y PROV.CRED al SecM. En el mismo mensaje, el AuC también envía una identidad de clave IdP.ID y una clave pública asociada IdP.PUK o *ib.sysparam* al SecM, si este no tiene capacidad para ejecutar el protocolo de TLS.
4) El SecM almacena PROV.ID y PROV.CRED de manera segura, así como IdP.ID e IdP.PUK o *ib.sysparam* si se han facilitado. El SecM protegerá el IdP.ID y la IdP.PUK o *ib.sysparam* contra modificaciones no autorizadas.

Condición final: El SecM está registrado en el AuC y listo para el aprovisionamiento de identidad y clave.

C.3 Búsqueda de parámetros públicos

Una entidad utilizará el procedimiento de aprovisionamiento de identidad o clave para obtener los parámetros públicos para el sistema de IBC con el que se ha registrado la entidad. Una entidad, que puede ser un dispositivo IoT o una entidad de gestión de un sistema de IBC, seguirá la especificación definida en la cláusula 4 de [IETF RFC 5408] para recuperar del PPS conocido los parámetros públicos correspondientes a otro sistema de IBC. Los IBESysParams de la respuesta de [IETF RFC 5408] se sustituirán por los IBSysParams definidos en la presente Recomendación. En [IETF RFC 5408] se presupone que la consulta al dispositivo IoT puede establecer un canal seguro basado en TLS con el PPS consultado. Si no se puede cumplir este requisito, signatureAlgorithm y el campo de firma de IBSysParams existirán y serán válidos. Una vez recuperados los IBSysParams, se ejecutará un proceso de verificación de firma adecuado y solo se aceptarán los parámetros públicos recuperados si la firma de IBSysParams es válida y la clave pública que verifica la firma es auténtica y válida.

C.4 Aprovisionamiento de identidad y clave

El aprovisionamiento de identidad y clave incluye los procedimientos de asignación de identidad, extracción de clave privada y distribución de clave. Tras el proceso de inicialización, los dispositivos solo tienen una identidad provisional. El IdP, SM-DP o MNO determinará qué entidad debe asignarse al dispositivo solicitante, se comunicará con el KMS para generar la clave privada correspondiente y, por último, distribuirá la identidad, la clave privada y los parámetros públicos al dispositivo de manera segura.

Véase la Figura C.2.

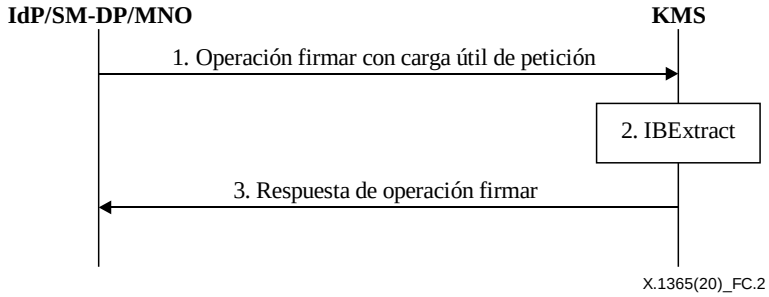

Figura C.2 – Generación de clave privada con protocolo de interoperabilidad para la gestión de claves

- **Generación de clave privada con KMIP**

Condiciones iniciales

Se parte de la base de que el IdP/SM-DP/MNO actúa como generador de la clave privada *ib.prk*. Para que el IdP/SM-DP/MNO pueda invocar la función IBExtract en un KMS, se deberán satisfacer las condiciones siguientes:

a) Se ha establecido un canal seguro entre el IdP/SM-DP/MNO y el KMS.

b) El IdP/SM-DP/MNO ha completado un proceso de autentificación con el KMS, y el IdP/SM-DP/MNO autentificado tiene autorización para ejecutar la petición IBExtract.

Procedimiento

1) El IdP/SM-DP/MNO preparará la carga útil de petición e invocará la operación de firma para enviar el mensaje de petición codificado al KMS.

2) El KMS comprobará si la petición es válida y el IdP/SM-DP/MNO tiene autorización para invocar esta operación. Si no se cumple alguna de estas condiciones, el KMS devolverá una respuesta con una indicación de fallo. En caso contrario, el KMS ejecutará **IBExtract** con *ib.msk*, *ib.pubparam* y los parámetros especificados en la petición.

3) El KMS devolverá la respuesta de ejecución al IdP/SM-DP/MNO. Si la operación se completa correctamente, el KMS devolverá la clave privada generada *ib.prk* con forma de IBPrivateKeyBlock, que es un valor CHOICE definido con ASN.1 como sigue:

```
IBPrivateKeyBlock ::= CHOICE {
    bb1PrivateKeyBlock      BB1PrivateKeyBlock,
    bfPrivateKeyBlock       BFPrivateKeyBlock,
    eccsiPrivateKeyBlock    ECCSIPrivateKeyBlock,
    skPrivateKeyBlock       SKPrivateKeyBlock,
    sm9PrivateKeyBlock      SM9PrivateKeyBlock
}
```

La extensión del KMIP como apoyo a esta operación se trata en el Apéndice II.

Condición final: El IdP/SM-DP/MNO recuperó la clave privada correspondiente a la identidad de la petición.

- **Aprovisionamiento de identidad y clave para la eUICC**

Condiciones iniciales

a) La eUICC está registrada en el SM-SR y lista para la descarga del perfil.

b) El SM-DP ha creado un perfil no personalizado de acuerdo con la descripción de perfil facilitada por el MNO.

c) El MNO tiene una petición para un número de perfiles de eUICC.

d) El perfil no personalizado se ha validado en el tipo de eUICC objetivo con el procedimiento de verificación de perfil no personalizado.

Procedimiento

1) El MNO posibilita que un SM-DP seleccionado realice el proceso de ordenación de perfiles. Los detalles de este proceso se presentan en la cláusula 3.5.3 de [b-GSMA SGP.02].

2) El SM-DP crea un perfil personalizado con los datos recibidos del MNO. En concreto, el SM-DP utilizará la Identidad internacional de suscripción al servicio móvil (IMSI) seleccionada para completar la operación de firma con el KMS con arreglo a lo especificado en la generación de clave privada con el KMIP y generar una clave privada para la IMSI seleccionada. La clave privada generada y los ibPublicParameters de IBSysParams se incluirán como claves en el perfil.

3) El perfil objetivo se reserva en la eUICC procedente del MNO. Los detalles del perfil para el proceso de descarga e instalación se recogen en la cláusula 3.5.4 de [b-GSMA SGP.02].

4) El perfil objetivo de la eUICC se habilita a través del SM-SR o del SM-DP y el SM-SR. Los pasos específicos para la habilitación de perfil se presentan en la cláusula 3.5.6 o 3.5.7 de [b-GSMA SGP.02].

Condición final: El perfil objetivo está habilitado en la eUICC. El perfil habilitado anteriormente se deshabilita. El EIS está actualizado.

- **Aprovisionamiento de identidad y clave para dispositivos con UICC no integrada**

Caso 1: El SecM tiene la capacidad de establecer una sesión TLS con el IdP

Condiciones iniciales

a) El SecM se ha registrado en el AuC.

Procedimiento

1) El SecM establece una sesión TLS con el IdP y verificará correctamente la validez del certificado TLS del IdP.

2) El SecM ejecuta un procedimiento de autentificación web con el IdP utilizando PROV.ID y PROV.CRED.

3) El IdP selecciona una identidad asignada al dispositivo solicitante y completa la operación de firma con el KMS con arreglo a lo especificado en la generación de clave privada con KMIP a fin de generar una clave privada para la identidad seleccionada.

4) El IdP envía la identidad asignada, la clave privada generada y los parámetros públicos al SecM a través de la sesión TLS.

5) El SecM almacena la clave privada de manera segura y los parámetros públicos estarán protegidos contra modificaciones no autorizadas.

Condición final: La clave objetivo se reserva en el SecM.

El SecM y el IdP aplicarán el protocolo definido en la cláusula 5 y en [IETF RFC 5408] para completar el procedimiento de aprovisionamiento de identidad y clave. En la respuesta, la estructura IBEPrivateKeyReply definida en [IETF RFC 5408] se reemplazará por IBPrivateKeyReply.

IBPrivateKeyReply ::= SEQUENCE SIZE (1..MAX) OF IBPrivateKey

IBPrivateKey ::= SEQUENCE {

pkgIdentity IBIdentityInfo OPTIONAL,

```
    pkgAlgorithm        OBJECT IDENTIFIER,
    pkgKeyData          IBPrivateKeyBlock,   --defined by pkgAlgorithm
    pkgOptions          SEQUENCE SIZE (1..MAX) OF PKGOption,
    ibSysParams         IBSysParams OPTIONAL
}
PKGOption ::= SEQUENCE {
    optionID            OBJECT IDENTIFIER,
    optionValue         OCTET STRING
}
```

Caso 2: El SecM no tiene implementación de TLS

Condiciones iniciales

a) El SecM se ha registrado en el AuC.

Procedimiento

1) El SecM genera la clave de encriptación de claves (KEK) y codifica la petición de aprovisionamiento de clave (IBKeyProvRequest). La petición incluye la KEK, el ID de aprovisionamiento (PROV.ID) y la credencial de aprovisionamiento (PROV.CRED), que se encriptan por medio de la clave pública del IdP identificada mediante IdP.ID. El resultado de la encriptación se codifica como EncryptedMsg. Envía al IdP la petición encriptada en el cuerpo de una petición HTTP POST.
2) El IdP descodifica el texto cifrado con la clave de privacidad identificada como IdP.ID en la petición, y comprueba que la indicación de tiempo está actualizada, que el contador es correcto, o ambos. Si la petición no supera estas verificaciones, el IdP devolverá una respuesta con indicación de fallo. El IdP también comprobará con el AuC si PROV.ID y PROV.CRED son correctos. Si esta verificación fracasa, el IdP devolverá una respuesta con indicación de fallo. El IdP selecciona una identidad asignada al dispositivo solicitante y completa la operación de firma con el KMS con arreglo a lo especificado en la generación de clave privada con KMIP a fin de generar una clave privada para la identidad seleccionada.
3) El IdP encripta la clave privada generada y, en caso necesario, la identidad y los parámetros públicos, que se codifican como IBKeyProvisionData, con la clave de encriptado de claves (KEK) mediante el algoritmo (keyProtAlg) especificado en la petición. El texto cifrado se codifica como EncryptedMsg. El IdP envía al SecM la petición encriptada en el cuerpo de una petición HTTP.
4) El SecM desencripta la respuesta y obtiene la identidad asignada, la clave privada y los parámetros públicos. Almacena la clave privada de manera segura y los parámetros públicos estarán protegidos contra modificaciones no autorizadas.

Condición final: La clave objetivo se reserva en el SecM.

```
IBKeyProvisionRequest ::= SEQUENCE {
    version         INTEGER { v1(1) },
    timer           Time OPTIONAL,
    counter         INTEGER     OPTIONAL,
    identity        OCTET STRING,
    credential      OCTET STRING,
```

```
        keyProtAlg      OBJECT IDENTIFIER,
        kek             OCTET STRING
}
Time ::= CHOICE {
        utcTime         UTCTime,
        generalTime     GeneralizedTime
}
IBKeyProvisionResponse ::= SEQUENCE SIZE(1..MAX) OF IBKeyProvisionData
IBKeyProvisionData ::= SEQUENCE {
        identity        OCTET STRING OPTIONAL,
        ibSysParams     IBSysParams OPTIONAL,
        ibPrivateKey    IBPrivateKeyBlock
}
EncryptedMsg ::= SEQUENCE {
        encryptionAlgorithm  EncryptionAlgorithmIdentifier,
        encryptedData        EncryptedData
}
EncryptionAlgorithmIdentifier ::= AlgorithmIdentifier
EncryptedData ::= OCTET STRING
```

C.5 Revocación de identidad y clave

Si se debe desautorizar una identidad en el sistema IBC por diferentes motivos, por ejemplo, porque el propietario de la identidad ha cancelado su suscripción al servicio o la clave privada correspondiente ha quedado comprometida, se debe revocar la identidad y puede ser necesario destruir la clave privada asociada por razones de seguridad. Si se revoca una identidad, se le asignará el estado revocado. Si una entrada consulta el estado de una identidad revocada, el IdP/SM-DP/MNO devolverá el valor correcto definido en el OISP. Para comprobar el estado de una identidad de manera más eficiente, una entidad puede recuperar periódicamente la IRL del IdP/SM-DP/MNO y almacenarla en local, y utilizar la IRL más reciente para determinar si una identidad está revocada sin consultar el estado en línea de cada identidad. En las eUICC, antes de destruir la clave privada es preciso deshabilitarla y, a continuación, eliminar el perfil de la eUICC.

- **Revocación de identidad y clave para eUICC**

Condiciones iniciales

a) El perfil objetivo está habilitado en la eUICC.

Procedimiento

1) El MNO inicia la deshabilitación de perfil con el proceso de SM-DP. Los datos del proceso de deshabilitación de perfil figuran en la cláusula 3.5.8 de [b-GSMA SGP.02]. SM-DP asignará el estado revocado a la identidad.
2) El MNO inicia el proceso de eliminación de perfil. Los pasos específicos de la eliminación de ISD-P se detallan en la cláusula 3.5.10 de [b-GSMA SGP.02]. SM-DP asignará el estado revocado a la identidad y, si el proceso de eliminación de ISD-P se completa con éxito, también se asigna el estado eliminado a la identidad. Cuando una entidad consulta el estado

de una identidad, SM-DP responde en consecuencia de acuerdo con el registro de estado. SM-DP publicará periódicamente una lista de estados de las identidades revocadas durante el periodo.

Condición final: El perfil objetivo está deshabilitado y se elimina de la eUICC.

- **Revocación de identidad y clave para dispositivos IoT con UICC no integrada**

Si se revoca una identidad, el IdP le asignará el estado revocado. Cuando una entidad consulta el estado de una identidad, el IdP responde en consecuencia de acuerdo con el registro de estado. El IdP publicará periódicamente una lista de estados de las identidades revocadas durante el periodo.

El proceso de activación de revocación y el mantenimiento de los estados de identidad son ajenos al ámbito de la presente Recomendación.

- **Protocolo de estado de identidad en línea**

Cuando el número de dispositivos IoT que se conectan con un operador de telecomunicaciones es elevado, puede ser necesario que un SM-DP, IdP o dispositivo IoT obtenga información de manera oportuna sobre el estado de revocación de una identidad de dispositivo IoT. En la presente Recomendación se especifica un OISP para habilitar el SM-DP, IdP o dispositivo IoT y determinar el estado actual de una identidad mediante consultas en línea. Un cliente OISP emite una petición de estado a un respondedor OISP y suspende la aceptación de la identidad en cuestión hasta recibir la respuesta a la petición. El OISP tiene características en común con el protocolo de estado de certificación en línea (OCSP) [IETF RFC 6960].

Una petición OISP contiene los datos siguientes:

```
OISPRequest ::= SEQUENCE {
        version             INTEGER { v1(1) },
        identity            IBIdentityInfoSet
}
```

– version indica la versión del protocolo, que en este documento es v1(1).

– identity es la petición OISP.

```
IBIdentityInfoSet ::= SEQUENCE SIZE(1..MAX) OF IBIdentityInfo

IBIdentityInfo ::= SEQUENCE {
        domainName          IA5String OPTIONAL,
        domainSerial        INTEGER OPTIONAL,
        identityType        OBJECT IDENTIFIER OPTIONAL,
        identityData        OCTET STRING
}
```

– domainName es OPTIONAL e IA5String representa el URI [b-URI] o IRI [b-IRI].

– domainSerial es OPTIONAL e incluye un INTEGER que define un conjunto único de parámetros públicos de IBC si se utiliza más de un conjunto de parámetros en un único dominio.

– identityType es OPTIONAL y contiene un OBJECT IDENTIFIER que define el formato con el que está codificado el campo identityData. Si falta este campo, se utiliza el tipo de identidad por omisión.

– identityData son los datos de la identidad objetivo.

Al recibir una petición, un respondedor OISP comprueba si el mensaje está bien formado y contiene la información que necesita el respondedor. Si el resultado de la verificación es fallido, el respondedor OISP produce un mensaje de error; en caso contrario, devuelve una respuesta definitiva de acuerdo con el estado de las identidades consultadas en la petición.

```
OISPResponse ::= SEQUENCE {
        responseStatus          OISPResponseStatus,
        responseData            OISPResponseData OPTIONAL
}
```

– responseStatus indica el estado de procesamiento de la petición anterior.

– responseData es OPTIONAL e incluye los datos de respuesta de la petición. Si el valor de responseStatus es una condición de error, no se define el campo responseData.

```
OISPResponseStatus ::= ENUMERATED {
      successful            (0),   -- Response has valid confirmations
      malformedRequest      (1),   -- Illegal confirmation request
      internalError         (2),   -- Internal error in issuer
      tryLater              (3),   -- Try again later
                                   -- value 4 is not used
      unauthorized          (5),   -- Request unauthorized
}
OISPResponseData ::= SEQUENCE {
      version               INTEGER { v1(1) },
      producedAt            GeneralizedTime,
      hashAlgorithm         AlgorithmIdentifier OPTIONAL,
      tbsIdStatus           SEQUENCE OF SingleIdStatus,
      signatureAlgorithm    AlgorithmIdentifier OPTIONAL,
      signature             BIT STRING OPTIONAL,
      certs                 [0] EXPLICIT SEQUENCE OF Certificate OPTIONAL
}
```

– version DEBE ser v1(1) para esta versión de la sintaxis de respuesta básica.

– producedAt es la hora a la que el respondedor OISP firmó esta respuesta.

– hashAlgorithm define un algoritmo hash para generar idHash en tbsIdStatus, si existe dicho campo. El campo es opcional y el valor por omisión es OBJECT IDENTIFIER para SHA256 sin parámetros.

– tbsldStatus indica las respuestas de cada identidad en una petición.

– signatureAlgorithm es OPTIONAL e incluye el algoritmo que se utilizó para firmar la respuesta.

– signature se calcula sobre la base del resultado ASN.1 DER del campo producedAt para tbsIdStatus con el algoritmo de firma especificado. Este campo es OPTIONAL y puede no estar definido si el cliente OISP cuenta con otros métodos para garantizar la autenticidad de la respuesta. Por ejemplo, la respuesta se transmite a través de un canal TLS seguro entre el cliente y el respondedor.

– certs es OPTIONAL e indica el certificado que ayuda al cliente OISP a verificar la firma del respondedor. La estructura de Certificate se define en [IETF RFC 5280].

```
SingleIdStatus ::= SEQUENCE {
    idHash          OCTET STRING OPTIONAL,
    identityID      IBIdentityInfo OPTIONAL,
    identityStatus  IdentityStatus,
}
```

– idHash es OPTIONAL e incluye el hash de la identidad de la petición. Si identityID es demasiado largo, se puede utilizar idHash para representar la identidad consultada. identityID es OPTIONAL y contiene el campo IBIdentityInfo de la identidad objetivo en la petición.

– identityStatus indica el estado de la identidad en la petición anterior.

```
IdentityStatus ::= CHOICE {
    good                [0]  IMPLICIT NULL,
    revoked             [1]  IMPLICIT RevokedInfo,
    unknown             [2]  IMPLICIT UnknownInfo,
    updated             [3]  IMPLICIT IBIdentityInfo,
    revokedAndDeleted   [4]  IMPLICIT RevokedInfo
}
UnknownInfo ::= NULL
```

– El estado "good" indica una respuesta positiva a la consulta de estado.

– El estado "revoked" indica que la identidad se ha revocado, temporal o permanentemente, y el valor es la información de revocación.

– El estado "unknown" indica que el respondedor no tiene información sobre el certificado al que se refiere la petición.

– El estado "updated" indica que se ha actualizado la identidad y el valor es una identidad asignada recientemente a la identidad de la consulta.

– El estado "revokedAndDeleted" indica que la identidad se ha revocado y se ha destruido la clave privada desde el dispositivo remoto.

```
RevokedInfo ::= SEQUENCE {
    revocationTime      GeneralizedTime,
    revocationReason    [0] EXPLICIT IRLReason OPTIONAL
}
IRLReason ::= ENUMERATED {
        unspecified             (0),
        keyCompromise           (1),
        pkgCompromise           (2),
        affiliationChanged      (3),
        superseded              (4),
        cessationOfOperation    (5),
        identityHold            (6),
```

```
                                -- value 7 is not used
    removeFromIRL               (8),
    privilegeWithdrawn          (9)
}
```

- **Lista de revocación de identidades**

Además de utilizar el OSIP para responder a las consultas de estado de identidad, una entidad como el IdP o SM-DP puede publicar regularmente una lista completa de las identidades revocadas durante un periodo. Para agilizar el proceso de comprobación de estado de las identidades, una entidad de comprobación de estado con gran capacidad de almacenamiento puede consultar la lista de revocación de identidades (IRL) y almacenarla en local. A partir de la IRL, la entidad de comprobación puede determinar si una identidad puede utilizarse para unas operaciones concretas, como autorización de acceso a red. Si la identidad en cuestión no está en la IRL, se presupone que su identidad es válida. Para que el sistema sea más eficaz, el IdP/SM-DP/MNO puede publicar la lista de identidades revocadas desde un momento concreto; esta lista se denomina IRL delta. Una IRL delta contiene la información de las identidades revocadas desde la publicación de una IRL completa. La utilización de IRL delta puede reducir de forma significativa la sobrecarga de las comunicaciones y el tiempo de procesamiento de las IRL. Una IRL es similar a una lista de revocación de certificados (CRL) [IETF RFC 5280].

La IRL se define como sigue:

```
IdentityRevocationList ::= SEQUENCE {
    tbsIdentityList         TBSIdentityRevocationList,
    signatureAlgorithm      AlgorithmIdentifier OPTIONAL,
    signatureValue          BIT STRING OPTIONAL
}
```

– tbsIdentityList es la lista de identidades revocadas con información adicional, como la hora de revocación.

– signatureAlgorithm define el algoritmo que utilizó el emisor de la IRL para firmar la lista. Este campo es opcional y no aparece si no existe signatureValue.

– signatureValue define el valor de la firma generada por el emisor en tbsIdentityList. Este campo es opcional y no aparece si el cliente de la petición tiene otros métodos para garantizar la autenticidad de la lista recuperada.

```
TBSIdentityRevocationList ::= SEQUENCE {
    version                 INTEGER { v1(1) },
    issuer                  Name,
    irlNumber               INTEGER OPTIONAL,
    deltaList               BOOLEAN OPTIONAL,
    thisUpdate              Time,
    nextUpdate              Time OPTIONAL,
    domainName              IA5String OPTIONAL,
    domainSerial            INTEGER OPTIONAL,
    revokedIdentities       SEQUENCE OF SEQUENCE {
      identity              IBIdentityInfo,
```

```
        revocationDate      Time,
        irlEntryExtensions  Extensions OPTIONAL
    } OPTIONAL,
    irlExtensions           [0] EXPLICIT Extensions OPTIONAL
}
Name ::= CHOICE {--imported from [IETF RFC 5280]
    rdnSequence RDNSequence
}
RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
RelativeDistinguishedName ::= SET SIZE (1..MAX) OF AttributeTypeAndValue
AttributeTypeAndValue ::= SEQUENCE {
    type    AttributeType,
    value   AttributeValue
}
AttributeType ::= OBJECT IDENTIFIER
AttributeValue ::= ANY -- DEFINED BY AttributeType
Extensions ::= SEQUENCE SIZE (1..MAX) OF Extension
Extension ::= SEQUENCE {--imported from [IETF RFC 5280]
    extnID      OBJECT IDENTIFIER,
    critical    BOOLEAN DEFAULT FALSE,
    extnValue   OCTET STRING
                -- contains the DER encoding of an ASN.1 value
                -- corresponding to the extension type identified
                -- by extnID
}
```

– version indica la versión de la estructura de la IRL.
– issuer es el nombre de la entidad que emite la IRL.
– irlNumber es el número de emisor de la IRL actual. Empieza en 0. Para cada publicación de IRL completa, el número aumenta en 1. Es opcional.
– deltaList indica si la IRL actual es una IRL delta. La lista contiene únicamente la información de las identidades revocadas desde que se publicó una IRL completa indexada según irlNumber.
– thisUpdate especifica la hora de generación de esta IRL.
– nextUpdate define la hora de generación de la próxima IRL. Es opcional.
– domainName define el dominio de identidad IBC.
– domainSerial define el número de dominio de identidad IBC.
– revokedIdentities es el conjunto de identidades revocadas.
  • identity es el detalle de la identidad revocada.
  • revocationDate es la hora a la que se revocó la identidad.

- irlEntryExtensions define posibles extensiones de revokedIdentity. Actualmente no hay ninguna extensión definida.

– irlExtensions define posibles extensiones de IRL. Actualmente no hay ninguna extensión definida.

Anexo D

Autenticación

(El presente anexo forma parte integrante de la presente Recomendación.)

En el presente anexo se amplían cuatro protocolos de autenticación vigentes para dar soporte a la IBC.

D.1 Protocolo de transporte secreto de una pasada

Este protocolo corresponde al mecanismo de transporte de claves secretas 2 en [ISO/CEI 11770-3]. Transfiere una clave secreta, generada, encriptada y firmada por la entidad A, de la entidad A a la entidad B, con autenticación de clave explícita de la entidad A a la entidad B y autenticación de clave implícita de la entidad B a la entidad A. La autenticación de clave explícita de la entidad A a la entidad B se consigue mediante la firma por la entidad A del secreto encriptado y de un parámetro de variante temporal (TVP). La autenticación de clave implícita de la entidad B a la entidad A se consigue encriptando el secreto con el ID de B, lo que implica que solo B puede recuperar el secreto. Véase la Figura D.1.

Para llevar a cabo el protocolo, deberán cumplirse los siguientes requisitos:

– La entidad A tiene una clave privada de firma *A.ib.prk* correspondiente a su ID y a los parámetros públicos relacionados *A.ib.pubparam.*

– La entidad B tiene una clave privada de desencriptación *B.ib.prk* correspondiente a su ID y a los parámetros públicos relacionados *B.ib.pubparam*.

– La entidad A tiene acceso a una copia autenticada del parámetro público de la entidad B para el encriptado *B.ib.pubparam* y el ID de la entidad B.

– La entidad B tiene acceso a una copia autenticada del parámetro público de la entidad A para la firma *A.ib.pubparam* y el ID de A.

– El TVP opcional será una indicación de tiempo o un número de secuencia. Si se utilizan indicaciones de tiempo, entonces las entidades A y B necesitan tener relojes sincrónicos o utilizar una indicación de tiempo de un tercero de confianza.

– A y B pueden compartir los mismos parámetros públicos, es decir, *A.ib.pubparam = B.ib.pubparam*.

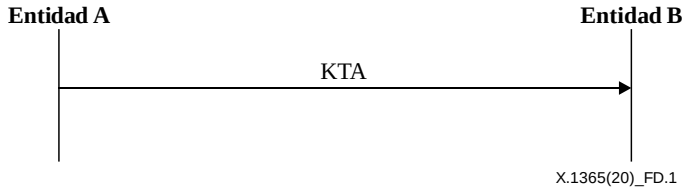

Figura D.1 – Protocolo de transporte secreto de una pasada

1) La Entidad A genera un secreto aleatorio *K* de la longitud requerida.

2) La entidad A genera *BE*=**IBEnc**(*B.ib.pubparam, ID_B, [ID_A]||K|||Text1*). Text1 puede estar vacío, e *ID_A* es opcional si la entidad B tiene otros medios para obtener el ID de la entidad A.

3) La entidad A genera *S*=**IBSign**(*A.ib.pubparam, ID_A, A.ib.prk, [ID_B]|||TVP||BE|||Text2*). Text2 puede estar vacío e *ID_B* es opcional si la entidad B conoce el ID utilizado, *ID_B*, para el encriptado.

4) La entidad A genera el testigo KTA=[*ID_B*]||TVP||BE|||*Text2*|||S||*Text3*.

5) Cuando el TVP es una indicación de tiempo, la entidad B verifica si el TVP está dentro de la diferencia de tiempo permitida. De lo contrario, la entidad B rechaza el testigo.

6) Si la entidad B puede obtener *ID_A* por otros medios y el TVP es un número de secuencia, la entidad B comprueba primero si el número de secuencia es mayor que el que se mantiene para la entidad B. De lo contrario, la entidad B rechaza el testigo.

7) Si la entidad B puede obtener *ID_A* por otros medios, la entidad B verifica la firma *S* en KTA por **IBVerify**(*A.ib.pubparam, ID_A,* [*ID_B*]||TVP||BE||*Text2,* S). Si la firma no es válida, la entidad B rechaza el testigo.

8) La entidad B descifra *BE* por [*ID_A*]||*K*|*Text1=***IBDec**(*B.ib.pubparam, ID_B, B.ib.prk,* BE).

9) Si la entidad B solo puede obtener *ID_A* después del paso 8, la entidad B comprueba la frescura del TVP, si el TVP es un número de secuencia. Si el TVP no está fresco, la entidad B rechaza el testigo. La entidad B verifica además la firma *S*. Si la firma no es válida, la entidad B rechaza el testigo.

10) Si se superan todas las comprobaciones y verificaciones, la entidad A y la entidad B utilizan *K* para proteger los siguientes mensajes. Ambas entidades pueden utilizar una función de cálculo de claves (KDF) [b-IEEE 1363] para generar claves para el encriptado y la autenticación de mensajes.

NOTA 1 – El protocolo puede convertirse en un protocolo de autenticación de entidad unilateral eliminando BE del mensaje firmado por la entidad A y KTA. Esa modificación se convierte en el sistema de autenticación de una sola pasada definido en [b-ISO/CEI 9798-3].

NOTA 2 – El protocolo puede convertirse en un protocolo de autenticación de entidad bilateral exigiendo que la entidad B devuelva *K* a la entidad A. La entidad B recibe autenticación de la entidad A demostrando que puede recuperar *K*, lo que le obliga a poseer la clave privada *B.ib.prk*.

NOTA 3 – Los algoritmos de encriptación de firmas basados en la identidad, como el algoritmo de encriptación de firmas BLMQ [b-Barreto], el algoritmo de encriptación de firmas Chen-Malone-Lee [b-Chen], pueden utilizarse para mejorar la eficiencia.

D.2 TLS-IBS

En esta cláusula se especifica otro protocolo de autenticación denominado TLS-IBS. Se supone que tanto el lado del servidor como el lado del dispositivo de IoT disponen de credenciales basadas en la identidad, que incluyen una identidad, una clave privada para la firma y parámetros públicos KMS (por ejemplo, KPAK según se define en [IETF RFC 6507] como parámetro de cálculo). Las definiciones de la estructura de parámetro público KMS para los algoritmos soportados se encuentran en el Anexo B.

La TLS-IBS se ha desarrollado sobre la base de [IETF RFC 7250]. Tradicionalmente, el servidor y el cliente TLS intercambian claves públicas aprobadas mediante certificados de infraestructura de clave pública (PKI). Se considera que es complicado y puede causar carencias en la seguridad con el uso de certificados PKI. Para simplificar el intercambio de certificados, en [IETF RFC 7250] se ha especificado el uso de una clave pública bruta en TLS. Es decir, en lugar de transmitir un certificado completo en los mensajes TLS, solo se intercambian claves públicas entre el cliente y el servidor. Con todo, se supone que existe un mecanismo fuera de banda de vinculación de clave pública e identidad. Para las redes de IoT, la TLS con una clave pública bruta es particularmente atractiva, pero las identidades vinculantes con claves públicas pueden ser un problema. El coste de mantener un cuadro grande para la correspondencia de identidades y claves públicas en el lado del servidor conlleva un coste de mantenimiento adicional, por ejemplo, los dispositivos tienen que registrarse previamente con el servidor. Para simplificar la vinculación entre la clave pública y la entidad que presenta la clave pública, una mejor manera sería utilizar la IBC, como la clave pública ECCSI especificada en [IETF RFC 6507] para la autenticación. A diferencia de los certificados UIT-T X.509 y las claves públicas brutas, una clave pública en la IBC adopta la forma de identidad de la entidad.

Eso ayuda a eliminar la necesidad de vinculación entre una clave pública y la entidad que presenta la clave pública.

Cuando la IBS se utiliza como clave pública bruta para TLS, la firma y los algoritmos *hash* se negocian durante la toma de contacto. La toma de contacto entre el servidor y el cliente TLS sigue los procedimientos definidos en [IETF RFC 7250] y TLS 1.3 [IETF RFC 8446], pero con el soporte de los algoritmos IBS como esquemas de firma.

A continuación, el protocolo TLS-IBS desarrollado sobre la base de [IETF RFC 7250] y TLS 1.3 con ECCSI [IETF RFC 6507], IBS1 (Hess-IBS), IBS1 (Cha-Cheon-IBS) y SM9-IBS [ISO/CEI 14888-3] como algoritmos de firma se especifica a continuación:

1) El dispositivo de IoT envía ClientHello al servidor, incluyendo la extensión key_share, signature_algorithms, server_certificate_type y client_certificate_type, e indica que soporta los algoritmos IBS y clave pública bruta.

2) El servidor envía ServerHello al dispositivo IoT, incluyendo la extensión key_share, server_certificate_type, client_certificate_type, Certificate, CertificateRequest, CertificateVerify y Finished, e indica que soporta la clave pública bruta e incluye su identidad (ServerID), parámetros KMS (OID, parámetros KMS) en la parte de certificado. Las estructuras de datos para los parámetros KMS se definen en la cláusula D.2.3. En el mensaje CertificateVerify se incluye una firma generada con la clave privada del servidor.

3) Después de verificar la identidad y la firma del servidor, el dispositivo de IoT envía su clave pública bruta como Certificate, CertificateVerify y Finished al servidor. El dispositivo IoT incluye sus parámetros de identidad (ClientID) y KMS (OID, KMS) en el área de certificados, que es la clave pública bruta del cliente. Las estructuras de datos para los parámetros KMS se definen en la cláusula D.2.3. Se incluye una firma generada con la clave privada del cliente.

4) Los pasos restantes son los mismos que los de TLS 1.3 en [IETF RFC 8446].

Véase la Figura D.2.

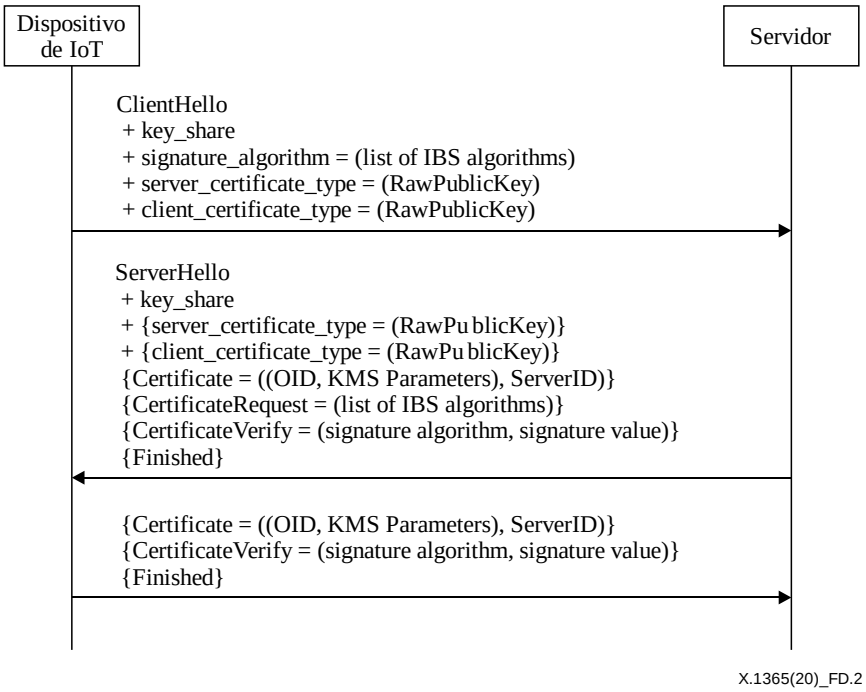

Figura D.2 – TLS-IBS

D.2.1 ClientHello

El formato de mensaje ClientHello es el mismo que el especificado en TLS 1.3 [IETF RFC 8446], pero los valores del algoritmo de firma deben ampliarse para la IBS.

El mensaje ClientHello indica al servidor los tipos de certificado o clave pública bruta soportados por el cliente, así como los tipos de certificado que el cliente espera recibir del servidor. El mensaje ClientHello incluye los algoritmos IBS deseados sobre la base del orden de preferencia del cliente. En TLS 1.3, se define una estructura de datos denominada SignatureScheme para los algoritmos de firma. Para soportar el algoritmo IBS, debe ser extendido de la siguiente manera:

```
enum {
...
/* IBS signature algorithm */
eccsi_sha256 (0x0704)
ibs1_sha256 (0x0705)
ibs2_sha256 (0x0706)
sm9_ibs_sm3 (0x0707)
/* Reserved Code Points */
private_use (0xFE00..0xFFFF),
(0xFFFF)
} SignatureScheme;
```

Los detalles de los puntos de código para los algoritmos de firma extendidos pueden encontrarse en el registro TLS [b-IANA TLS REG].

D.2.2 ServerHello

El formato de mensaje de ServerHello es el mismo que el especificado en TLS 1.3 [IETF RFC 8446]. El SignatureScheme se extiende de la misma manera que en Client_Hello.

D.2.3 Certificado de servidor

Para el certificado de servidor, una estructura de certificado se define como RawPublicKey en [b-IEF RFC 7250]. Como en [IETF RFC 7250], se utiliza una estructura de datos subjectPublicKeyInfo para especificar la clave pública bruta y su algoritmo criptográfico. Dentro de la estructura subjectPublicKeyInfo, se definen dos campos, algoritmo y parámetros. El algoritmo especifica el algoritmo criptográfico utilizado con una clave pública bruta que se representa mediante OID; y el campo de parámetros proporciona los parámetros necesarios asociados con el algoritmo. La identidad del servidor debe estar en la parte subjectPublicKey.

NOTA – La identidad debe seguir el formato definido en el Apéndice I.

```
subjectPublicKeyInfo ::= SEQUENCE {
    algorithm     AlgorithmIdentifier,
    subjectPublicKey                BIT STRING
}
```

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

Cuando se utiliza un algoritmo IBS, se utiliza una identidad como clave pública bruta que puede convertirse en una cadena OCTET. Por lo tanto, el certificado y la estructura subjectPublicKey pueden ser reutilizados sin cambios.

El campo de algoritmo en la estructura AlgorithmIdentifier es el ID de objeto del algoritmo IBS utilizado. Además de eso, es necesario comunicar al par el conjunto de parámetros públicos utilizados por el firmante. La información puede ser transportada en la carga útil del campo de parámetros en AlgorithmIdentifier. De acuerdo con los algoritmos anteriores, las estructuras de parámetros públicos son ECCSIPublicParameters, BFPublicParameters, BFPublicParameters y SM9PublicParameters, respectivamente, como se definen en el Anexo B.

Para soportar los algoritmos IBS sobre el protocolo TLS para generar el mensaje CeritificateVerify, es necesario definir una estructura de datos para el valor de firma.

- Una estructura de datos para ECCSI se define como sigue (basada en [IETF RFC 6507]):

  ```
  ECCSI-Sig-Value ::= SEQUENCE {
    r INTEGER,
    s INTEGER,
    pvt OCTET STRING
  }
  ```

  donde pvt (como se define en [IETF RFC 6507]) se codifica como 0x04 || coordenada x de [v]G || coordenada y de [v]G.

- A continuación se define una estructura de datos para IBS1:

  ```
  IBS1-Sig-Value ::= SEQUENCE {
    r INTEGER,
    s ECPoint
  }
  ```

  ECPoint ::= OCTET STRING como se define en [IETF RFC 5480]

- A continuación se define una estructura de datos para IBS2:

  ```
  IBS2-Sig-Value ::= SEQUENCE {
    r ECPoint,
    s ECPoint
  }
  ```

- A continuación se define una estructura de datos para SM9-IBS:

  ```
  SM9-Sig-Value ::= SEQUENCE {
    r INTEGER,
    s ECPoint
  }
  ```

Para utilizar un algoritmo de firma con TLS, es necesario proporcionar un OID para el algoritmo de firma. El Cuadro D.1 muestra la información básica necesaria para los algoritmos de firma IBS que se utilizarán para la TLS.

Cuadro D.1 – Algoritmos de firma basados en la identidad

Tipo de clave	Documento	OID
ISO/CEI 14888-3 ibs-1	ISO/CEI 14888-3: Mecanismo IBS-1	1.0.14888.3.0.7
ISO/CEI 14888-3 ibs-2	ISO/CEI 14888-3: Mecanismo IBS-2	1.0.14888.3.0.8
SM9-IBS	ISO/CEI 14888-3: Mecanismo IBS china	1.2.156.10197.1.302.1
Firmas sin certificado basadas en la curva elíptica para encriptación basada en la identidad (ECCSI)	Sección 5.2 de [IETF RFC 6507]	1.3.6.1.5.5.7.6.29

D.2.4 Certificado de cliente

Para soportar IBS, el certificado de cliente se amplía del mismo modo que el certificado de servidor.

D.3 EAP-TLS-IBS

En esta cláusula, el protocolo de autenticación EAP-TLS se amplía para soportar IBS. Tanto el lado de la red como el del UE disponen de credenciales basadas en la identidad, que incluyen una identidad, una clave privada para la firma y parámetros públicos KMS (por ejemplo, KPAK, como se define en [IETF RFC 6507]). Véase la Figura D.3.

La EAP-TLS se modifica como sigue.

1) Lo mismo que EAP-TLS.
2) Después de recibir la respuesta EAP con la identidad UE, ID_UE.
3) La AU envía ID_UE a la RSF para su validación.
4) La RSF valida el ID_UE basándose en la lista de revocación almacenada.
5) La RSF envía el resultado de validación a la AU.
6) Si ID_UE es válido, la AU envía el mensaje de inicio EAP-TLS al UE.

7-9) Lo mismo que lo descrito para TLS-IBS.

10) EAP-Success.

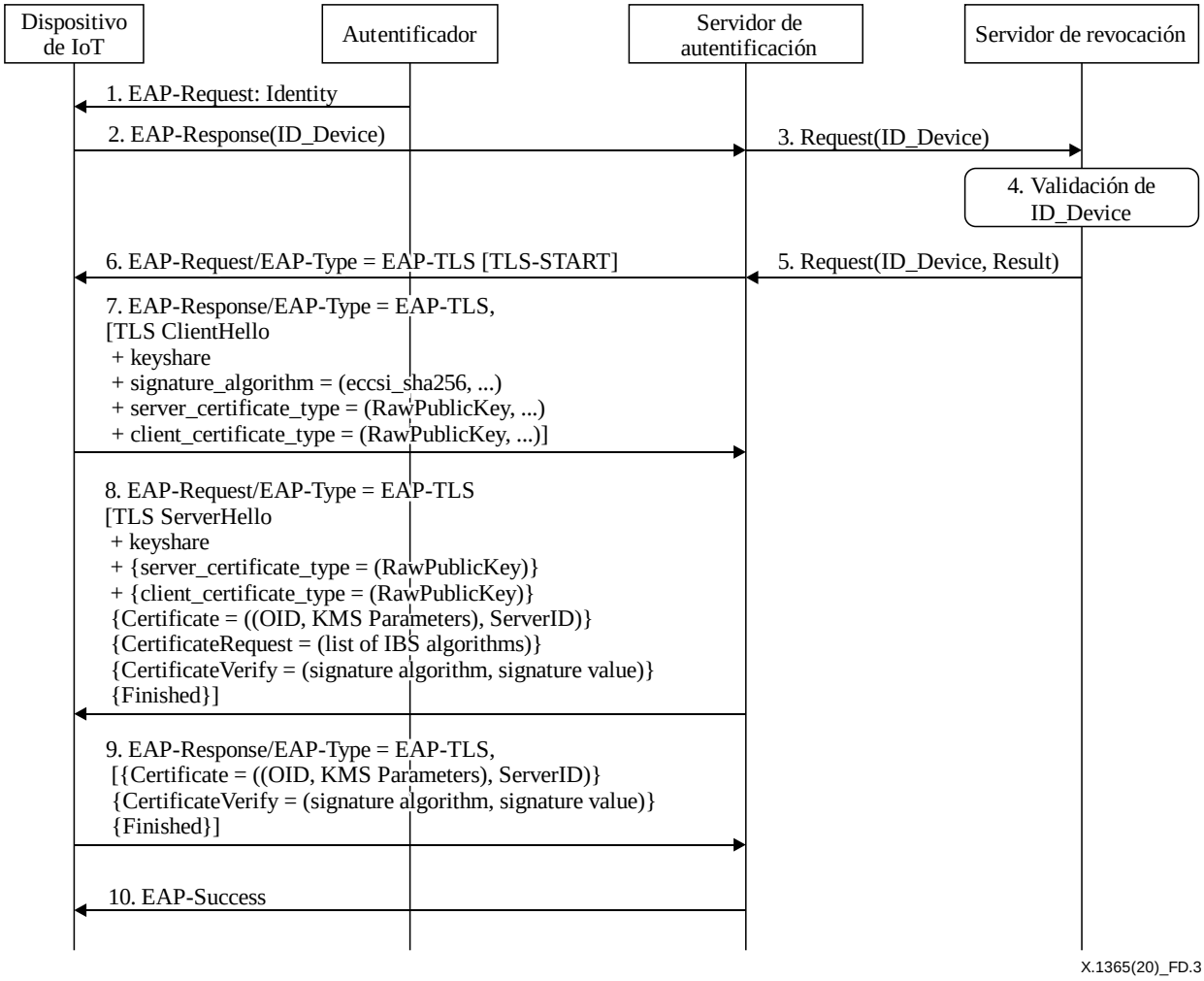

Figura D.3 – EAP-TLS-IBS

D.3.1 EAP-Request

El formato de mensaje EAP-Request es el mismo que el especificado en [IETF RFC 5216].

D.3.2 EAP-Response

El formato de mensaje de EAP-Response es el mismo que el especificado en [IETF RFC 5216].

D.3.3 ClientHello

El formato de mensaje ClientHello es el mismo que el de la cláusula D.2.1.

D.3.4 ServerHello

El formato de mensaje ServerHello es el mismo que el de la cláusula D.2.2.

D.3.5 Certificado de servidor

El formato de certificado de servidor es el mismo que el de la cláusula D.2.3.

D.3.7 Certificado de cliente

El formato de certificado de servidor es el mismo que el de la cláusula D.2.4.

D.4 EAP-PSK-ECCSI

En esta cláusula, EAP-PSK se amplía para soportar uno de los algoritmos IBS, ECCSI, para autenticación. Tanto UE como la AU disponen de credenciales basadas en la identidad con una

identidad, una SSK, un PVT y una KPAK, como se define en [IETF RFC 6507] como parámetro de computación.

Con las credenciales proporcionadas, UE y AU pueden calcular claves simétricas basadas en Diffie-Hellman estático mediante el intercambio de información de la identidad y el PVT, y luego utilizar la SSK propiedad de cada entidad. Por ejemplo, una UE puede calcular una clave después de recibir la identidad de la AU y su PVT, expresada como ID_AU y PVT_AU respectivamente, del siguiente modo:

K_UE = [SSK_UE](KPAK+[hash(G || KPAK || ID_AU || PVT_AU)]PVT_AU))

donde G es un punto de generación en la curva elíptica utilizada por el KMS para generar claves para UE y redes. Es suministrado a UE y a AU por el KMS junto con SSK, PVT y KPAK, etc. El uso de la función *hash* puede seguir el Anexo A de [IETF RFC 6507].

Del mismo modo, AU también puede calcular K_AU después de recibir la identidad y PVT de UE, como se indica a continuación:

K_AU = [SSK_AU](KPAK+[hash(G || KPAK || ID_UE || PVT_UE)]PVT_UE))

Puede demostrarse que K_UE es igual a K_AU.

Con las propiedades anteriores, EAP-PSK puede utilizarse para autenticación mutua de la siguiente manera.

1) El UE envía una solicitud de anexo a la AU e indica que EAP-PSK se utilizará para autenticación mutua.
2) La AU verifica el tipo de autenticación y decide un método de autenticación.
3) La AU envía el primer mensaje sobre el EAP-PSK al UE con un campo de identidad que contiene la ID_AU y la PVT_AU, así como un número aleatorio RAND_S, como requiere el EAP-PSK.
4) El UE obtiene una clave simétrica como K = [SSK_UE](KPAK+[hash(G || KPAK || ID_AU || PVT_AU)]PVT_AU)). El UE genera un número aleatorio RAND_P y calcula K' = KDF(K, RAND_P, RAND_S). El UE calcula una clave de autenticación (AK) y una clave de cálculo de clave (KDK) basadas en [b-IETF RFC4764] para EAP-PSK.
5) El UE envía el segundo mensaje sobre EAP-PSK a la AU, que contiene RAND_S, RAND_P, un MAC_P (MAC_P=CMAC-AES-128(AK, ID_P||ID_S||RAND_S||RAND_P)) para autenticación, y un campo de identidad que consta de ID_UE y PVT_UE.
6) La AU envía ID_UE a la RSF para su validación.
7) La RSF valida el ID_UE según su lista de revocación.
8) La RSF envía los resultados de validación a la AU.
9) Si el ID es válido, la AU obtiene una clave simétrica como K = [SSK_AU](KPAK+[hash(G || KPAK || ID_UE || PVT_UE)]PVT_UE)). La AU calcula además K' = KDF(K, RAND_P, RAND_S). La AU obtiene una AK y una KDK basadas en [IETF RFC 4764] para EAP-PSK. La AU autentica el UE basándose en el MAC_P recibido del mensaje. La AU calcula además una clave de sesión basada en RAND_P y la KDK.
10) La AU envía el tercer mensaje sobre el EAP-PSK al UE con un MAC_S (MAC_S=CMAC-AES-128(AK, ID_S|RAND_P)) para la autenticación y otros campos requeridos por el EAP-PSK.
11) El UE autentica la AU con el MAC_S recibido y calcula una clave de sesión con RAND_P y KDK calculados previamente.
12) El UE envía el último mensaje sobre el EAP-PSK a la AU para finalizar el procedimiento de autenticación EAP-PSK.

Véase la Figura D.4.

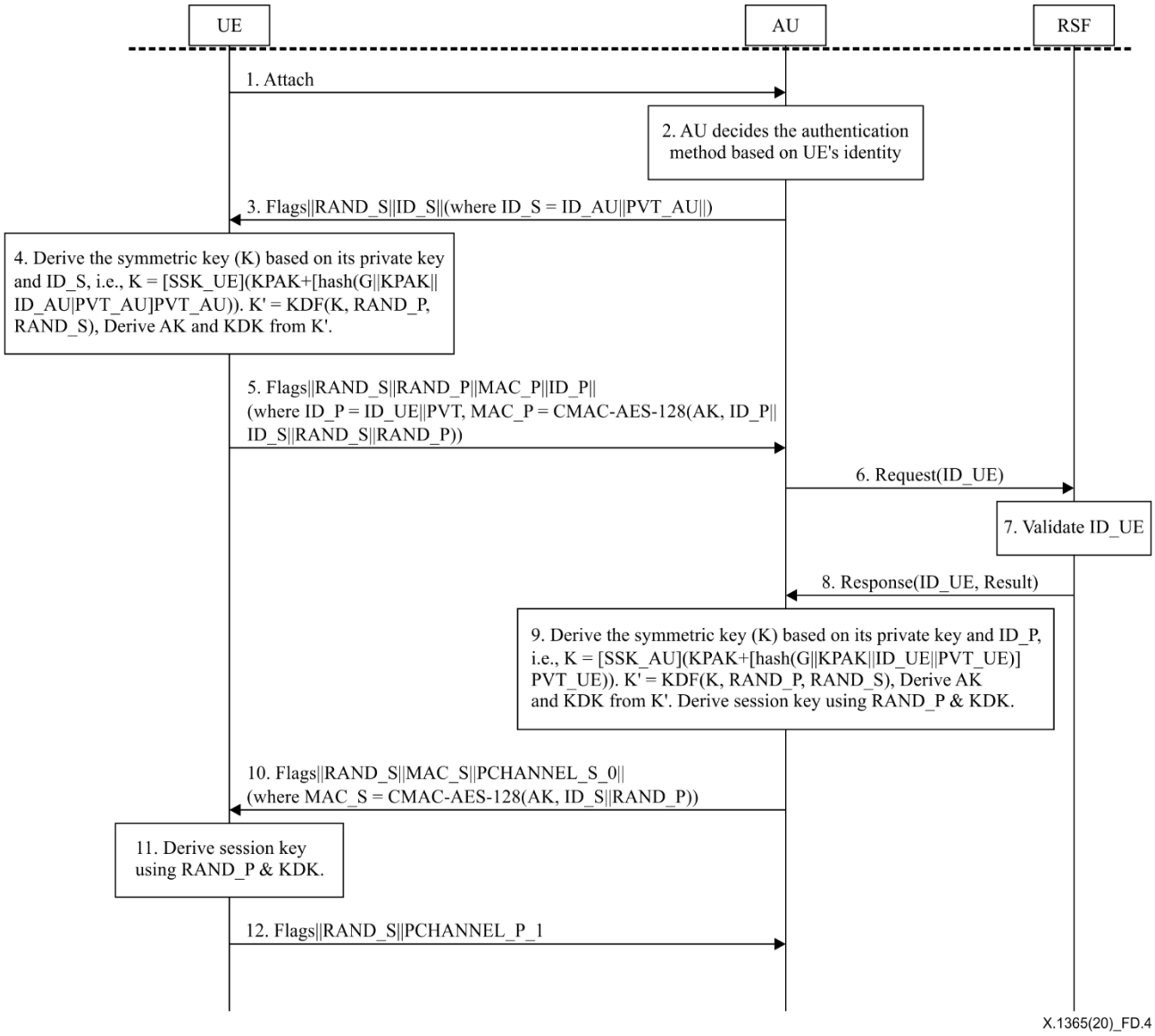

Figura D.4 – EAP-PSK--ECCSI

D.4.1 Adjuntar

Este mensaje imita el procedimiento de autenticación.

D.4.2 EAP-PSK--ECCSI primer mensaje (mensaje 3 de la Figura D.4)

El primer mensaje EAP-PSK--ECCSI es enviado por el servidor al par. El formato es el siguiente:

El primer mensaje EAP-PSK--ECCSI consiste en:

Un campo Flags de 1 byte

Un número aleatorio de 16 bytes: RAND_S

Un campo de longitud variable que transmite el identificador de acceso a la red del servidor: ID_S. La longitud de este campo se deduce del campo longitud EAP. La longitud de este identificador de acceso a la red no debe exceder los 966 bytes. Esta restricción tiene por objeto evitar problemas de fragmentación.

La Figura D.5 muestra un formato de ejemplo del primer mensaje sobre el EAP-PSK.

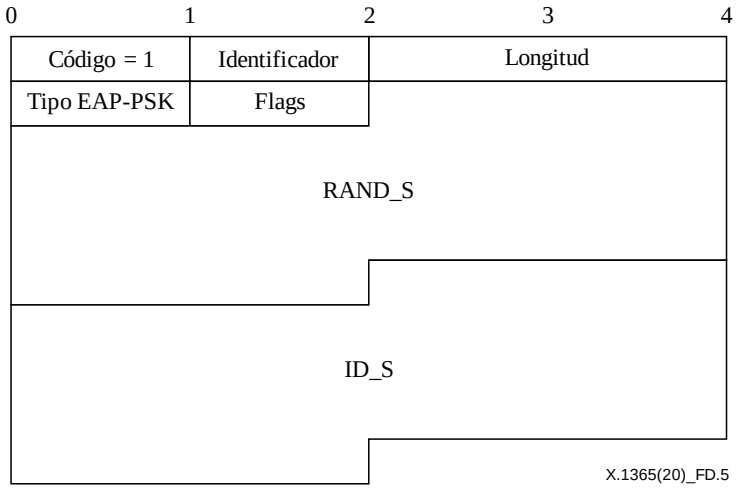

Figura D.5 – Formato de EAP-PSK

Para admitir la autenticación EAP-PSK basada en IBC, se utiliza el ID_S para el protocolo de EAP-PSK para transportar el ID_AU y PVT_AU. ID_S y PVT_AU se llevan en la estructura de datos de etiqueta, longitud y vector (TLV, por sus siglas en inglés), donde el primer octeto lleva un indicador de etiqueta y el segundo un campo de longitud, indicando la longitud del campo seguido. El campo vectorial tiene el valor.

El Cuadro D.2 define el TLV para ID y PVT utilizado con el EAP-PSK.

Cuadro D.2 – Definición de etiqueta, longitud y vector para identidad y testigo de verificación pública

	Etiqueta	**Longitud**	**Valor**
Identidad	1	Variable (≤255)	Definido por el proveedor de servicios
PVT	2	65	Número en hexadecimal

La Figura D.6 muestra el formato de mensaje EAP-PSK--ECCSI que contiene la identidad y PVT dentro del campo ID_S.

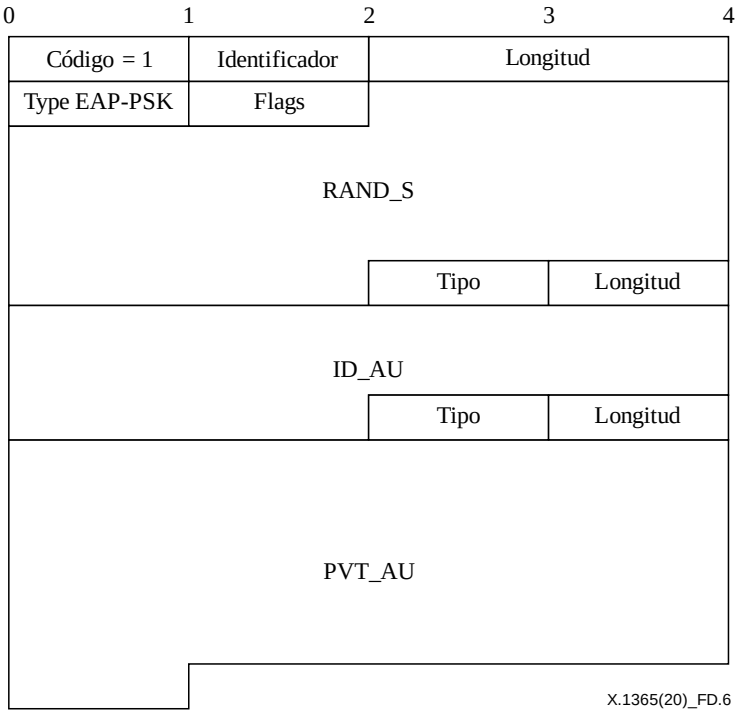

Figura D.6 – Formato de mensaje de EAP-PSK--ECCSI

D.4.3 Segundo mensaje EAP-PSK--ECCSI (mensaje 5 de la Figura D.4)

El segundo mensaje EAP-PSK-ECCSI es enviado por el par al servidor. El formato consiste en:

– un campo de flags de 1 byte;

– el número aleatorio de 16 bytes enviado por el servidor en el primer mensaje EAP-PSK--ECCSI (RAND_S) que sirve como ID de sesión;

– un número aleatorio de 16 bytes: RAND_P;

– un control de acceso a los medios de 16 bytes (MAC): MAC_P;

– un campo de longitud variable que transmite el identificador de acceso a la red del par: ID_P. La longitud de este campo se deduce del campo longitud EAP. La longitud de este identificador de acceso a la red no debe exceder los 966 bytes.

Del mismo modo, el campo ID_S de EAP-PSK se utiliza para llevar el campo ID_UE y PVT_UE. La Figura D.7 muestra el formato del segundo mensaje EAP-PSK.

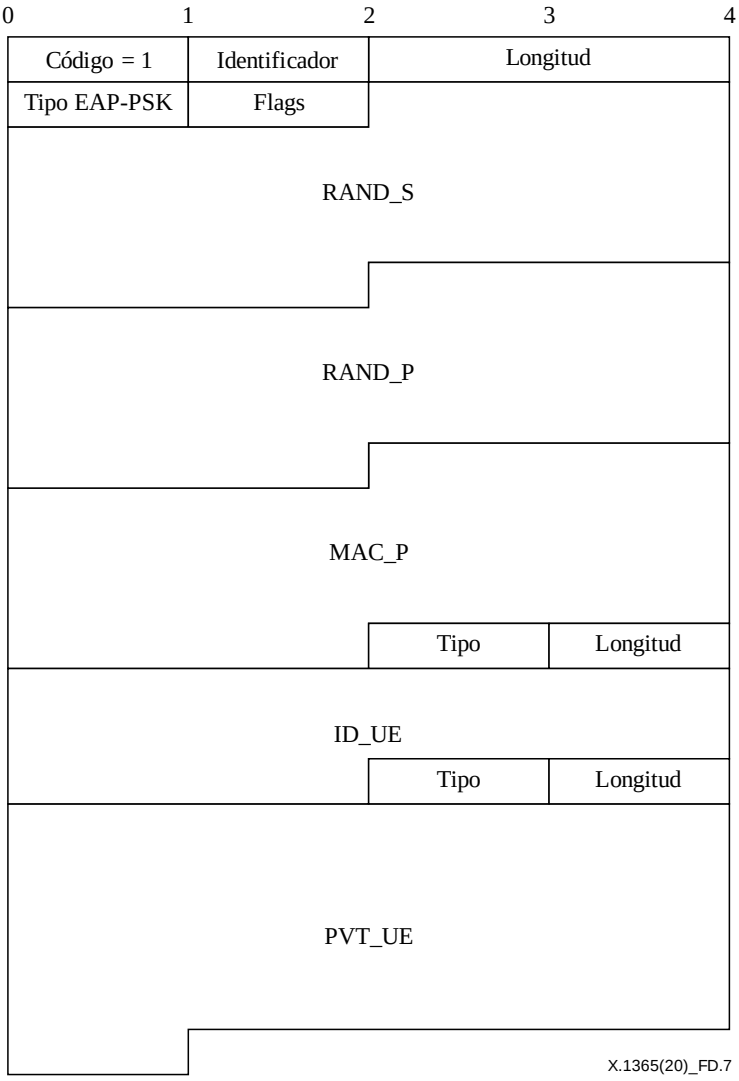

Figura D.7 – Formato de mensaje para el segundo mensaje sobre EAP-PSK--ECCSI

D.4.4 Tercer mensaje EAP-PSK--ECCSI (mensaje 10 de la Figura D.4)

El tercer mensaje EAP-PSK--ECCSI es enviado por el servidor al par. El formato es el mismo que el presentado en [IETF RFC 4764].

D.4.5 Cuarto mensaje EAP-PSK--ECCSI (mensaje 12 de la Figura D.4-1)

El cuarto mensaje EAP-PSK-ECCSI es enviado por el par al servidor. El formato es el mismo que el presentado en [IETF RFC 4764].

Apéndice I

Denominación de la identidad

(El presente apéndice no forma parte integrante de la presente Recomendación.)

El ID de una aplicación IoT puede ser el ID de un terminal o el ID de una plataforma IoT. El ID es un nombre que cumple el propósito de identificación. Un ID es una representación práctica del objeto y permite que el objeto sea referenciado o tratado en una base de datos o en protocolos de comunicación. Para cumplir este propósito, los ID tienen que ser únicos, o que el ID sea único en un sistema independiente. Por ejemplo, el código postal es único en un país, el carácter único del ID se da en un ámbito determinado. Además, un ID no es solo para un objeto singular, sino también para un grupo de objetos, lo cual permite la gestión y operación uniformes para dicho grupo.

Los OID [b-UIT-T X.660], [b-UIT-T X-Sup.31] son creados conjuntamente por la ISO/CEI y el UIT-T, y presentan numerosas características. Un OID tiene una estructura de árbol jerárquico, que puede extender con facilidad sus capas y la longitud de los ID. Un OID corresponde a un nodo en el árbol OID, que es capaz de identificar cualquier cosa (física o virtual, dispositivo o no dispositivo), y que puede conectarla con infraestructuras globales de información y comunicación. La raíz del árbol contiene los tres arcos siguientes: 0 (UIT-T), 1 (ISO) y 2 (conjunto ISO-UIT-T). Cada nodo del árbol está representado por una serie de números enteros separados por puntos, que corresponden al trayecto desde la raíz a través de la serie de nodos antecesores, hasta el nodo. Cada nivel de ID de autoridad de registro debe ser atribuido por la autoridad de registro del nivel superior. Por ejemplo, el OID que indica el Centro de Registro de Tarjetas IC de China, 1.2.156.20005, es atribuido por 1.2.156 (Miembro de la ISO, China), el OID del Centro Nacional de Registro de OID de China.

Un OID completo sería una combinación del ID de la autoridad de registro y del ID de la entidad, y estos dos componentes están separados por un punto, según se muestra en la Figura I.1. Si la empresa ha registrado un OID por la autoridad de registro de nivel superior, solo es necesario designar el ID de la entidad.

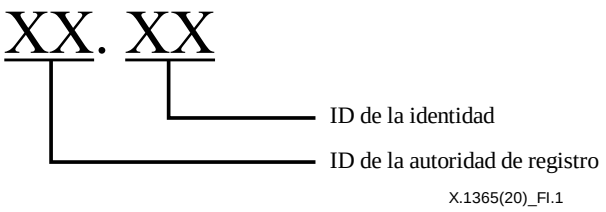

Figura I.1 – Estructura de OID completo para objetos

Por ejemplo, el ID de la entidad tendrá la estructura que se enumera en el Cuadro I.1.

Cuadro I.1 – Información detallada del ID de entidad

Byte	Componente	Interpretación
1	versión y reservado	4 bits para la versión del ID de la entidad, y 4 bits para dígitos reservados para el futuro
2	empresa	tipo de empresa
3~11	tiempo expirado	tiempo no válido de la identidad, 5 bytes para publicar el tiempo en tiempo Unix, y 4 bytes para el periodo de validez en segundos
12	tipo	valor 0 para número insignificante, 1 para el MAC y 2 para la IMSI
13	longitud (valor *l*)	el tamaño del valor (en bytes), 6 para el MAC y 8 para la IMSI
14~13+*l*	valor	número de identificación individual

El ID de la entidad tiene una longitud de 19 bytes cuando se utiliza el MAC como número de identificación individual, y de 21 bytes cuando se utiliza la IMSI. Normalmente, una IMSI se presenta como un número de 15 dígitos o menos y, salvo la red de prueba, el primer dígito no es cero [b-UIT-T E.212]. Ceros de relleno antes de la IMSI hasta 16 dígitos y utilizando 4 bits para un dígito, 8 bytes son suficientes para una IMSI.

La plataforma de IoT mantiene una lista para el direccionamiento. Cuando se registra por primera vez un dispositivo terminal, la plataforma añadirá una fila que contiene el ID y la dirección IP del dispositivo. Mediante la búsqueda del ID de un dispositivo de la lista puede obtenerse la dirección IP correspondiente al dispositivo. Véase el Cuadro I.2.

Cuadro I.2 – Ejemplo de la lista para el direccionamiento

Identificador	Dirección IP
1.2.9c.4e25.10.1.5b3e408003c26700.1.6.38B1DBC3156F	192.168.0.1

Apéndice II

Extensiones KMIP para dar soporte a la IBC

(El presente apéndice no forma parte integrante de la presente Recomendación.)

El KMIP puede extenderse como sigue para dar soporte a las operaciones requeridas de la IBC con el KMS, en particular la inicialización del sistema con el KMIP y la generación de clave privada con las operaciones del KMIP definidas en C.1 y C.4, respectivamente.

La composición de la carga útil requerida para crear el par de claves se refleja en el Cuadro II.1

Cuadro II.1 – Carga útil requerida

Objeto	Requerido	Descripción
Atributo de plantilla de clave privada	Sí	Especifica los atributos cuando la función **IBSetup** genera *ib.msk* e *ib.pubparam.*

El atributo de plantilla de clave privada incluirá los atributos que se recogen en el Cuadro II.2.

Cuadro II.2 – Atributo de plantilla de clave privada

Objeto	Requerido	Codificación	Descripción
Algoritmo criptográfico	Sí	Enumeración, véase el Cuadro II.3	Especifica la función **IBSetup**.
Longitud criptográfica	No	Entero	Especifica la longitud de bit de la característica del campo primo en que se basa la curva elíptica.
Máscara de utilización criptográfica	Sí	Entero	Especifica el uso del *ib.msk* que será Señal para la generación de claves. En lo esencial, **IBExtract** es un proceso de señalización.
Parámetros de dominio criptográfico	Sí	Objeto	Especifica más parámetros para escoger parámetros de sistema, como la curva elíptica utilizada.
Parámetros criptográficos	Sí	Objeto	Especifica otras funciones, como una función hash, que se utilizarán con las funciones de **IBExtract**.

El algoritmo criptográfico será uno de los valores que figuran en el Cuadro II.3.

Cuadro II.3 – Algoritmo criptográfico (generación de claves)

Nombre	Valor
IBC-KGA-BB1	00000030
IBC-KGA-BF	00000031
IBC-KGA-ECCSI	00000032
IBC-KGA-SK	00000033
IBC-KGA-SM9	00000034

La longitud criptográfica será un valor igual o mayor que 110.

El uso criptográfico se fijará como 00000001 (Señal).

Los parámetros de dominio criptográfico incluirán los atributos enumerados en el Cuadro II.4.

Cuadro II.4 – Parámetros de dominio criptográfico

Objeto	Requerido	Codificación	Descripción
Longitud de Q	No	Entero	Especifica la longitud en bit del orden del grupo del que se escoge *ib.msk*.
Curva recomendada	Sí	Enumeración, véase el Cuadro II.5	Especifica la curva utilizada.
Tipo de emparejamiento	No	Enumeración, véase el Cuadro II.6	Cuando se utiliza, especifica el emparejamiento en un algoritmo basado en la identidad.
Nombre de dominio	No	CADENA DE TEXTO	Especifica un nombre único para los parámetros de sistema *ib.pubparam* generados.
Serie de dominio	No	ENTERO	Especifica un número de versión para los parámetros de sistema *ib.pubparam* generados.

La curva recomendada será uno de los valores recogidos en el Cuadro II.5.

Cuadro II.5 – Curva recomendada

Nombre	Valor
IBC-CURVE-SS1	00000070
IBC-CURVE-SS2	00000071
IBC-CURVE-BN-254-1	00000072
IBC-CURVE-BN-256-1	00000073
IBC-CURVE-BN-256-2	00000074
IBC-CURVE-BN-382-1	00000077
IBC-CURVE-BLS-12-381-1	0000007A
IBC-CURVE-BLS-12-442-1	0000007B
IBC-CURVE-BLS-12-455-1	0000007C
IBC-CURVE-BLS-12-461-1	0000007D
IBC-CURVE-KSS-16-340-1	0000007E
IBC-CURVE-KSS-18-348-1	0000007F

El tipo de emparejamiento será uno de los valores recogidos en el Cuadro II.6.

Cuadro II.6 – Tipo de emparejamiento

Nombre	Valor
Emparejamiento Weil	00000001
Emparejamiento Tate	00000002
Emparejamiento óptimo Ate	00000003

Los parámetros criptográficos incluirán los atributos recogidos en el Cuadro II.7.

Cuadro II.7 – Parámetros criptográficos

Objeto	Requerido	Codificación	Descripción
Algoritmo de la función hash	Sí	Enumeración, véase el Cuadro II.8	Especifica la función hash que será utilizada con la función de generación de claves.
Grupo de clave privada	No	Enumeración, véase el Cuadro II.9	Especifica en qué grupo se genera la clave privada si se utiliza un emparejamiento.

Al algoritmo de la función hash será uno de los valores que figuran en el Cuadro II.8.

Cuadro II.8 – Algoritmo criptográfico (hash)

Nombre	Valor
SHA224	00000040
SHA256	00000041
SHA384	00000042
SHA512	00000043
SHA3-224	00000044
SHA3-256	00000045
SHA3-384	00000046
SHA3-512	00000047
SM3	00000048

El grupo de clave privada será uno de los valores del Cuadro II.9.

Cuadro II.9 – Grupo de clave privada

Nombre	Valor
IBC-PRK-GROUP1	00000001
IBC-PRK-GROUP2	00000002
IBC-PRK-TWOGROUPS	00000003

La composición de la carga útil de respuesta del par de claves creada se indica en el Cuadro II.10.

Table II.10 – Carga útil de respuesta

Objeto	Requerido	Descripción
Identificador único de clave privada	Sí	El identificador único del objeto de clave privada nuevamente creado que puede utilizarse para acceder a *ib.msk*. El identificador está codificado como una cadena de texto.
Identificador único de clave pública	Sí	El identificador único del objeto de clave pública nuevamente creado que puede utilizarse para acceder a *ib.pubparam*. El identificador está codificado como una cadena de texto.

La composición de la carga útil requerida para obtener funcionamiento es la que figura en el Cuadro II.11.

Cuadro II.11 – Carga útil requerida

Objeto	Requerido	Descripción
Identificador único de clave pública	Sí	El identificador único del objeto de clave pública que puede utilizarse para acceder a *ib.pubparam*. El identificador está codificado como una cadena de texto.

La composición de la carga útil para obtener respuesta es la que figura en el Cuadro II.12.

Cuadro II.12 – Carga útil de respuesta

Objeto	Requerido	Descripción
Tipo de objeto	Sí	Tipo de objeto
Identificador único	Sí	El identificador único del objeto
Clave pública	Sí	Una estructura de clave pública que encapsula los datos de los parámetros públicos *ib.pubparam* de la IBC

El ID único será el mismo que el ID único de clave pública enviado en la carga útil de la solicitud obtener.

El tipo de objeto será 00000003 (clave pública).

La composición del bloque esencial del campo clave pública es la que figura en el Cuadro II.13.

Cuadro II.13 – Bloque esencial del campo clave pública

Objeto	Requerido	Codificación	Descripción
Tipo de formato de clave	Sí	Enumerar, véase el Cuadro II.14	Especifica el formato del valor de la clave.
Compresión de clave	No	Enumerar	Especifica si debe comprimirse el valor de la clave.
Valor de la clave	Sí	Estructura de clave para los parámetros públicos transparentes de la IBC	Una estructura de clave transparente de nueva creación para la clave pública IBC.
Algoritmo criptográfico	Sí	Enumerar, véase el Cuadro II.15	Lo mismo que crear carga útil de solicitud de par de claves

El tipo de formato de clave será el valor del Cuadro II.14.

Cuadro II.14 – Tipo de formato de clave

Nombre	Valor
Parámetros públicos transparentes de la IBC	00000016

La compresión de la clave será o bien 00000001 (sin comprimir) o 00000002 (comprimida original).

El valor de la clave tendrá los atributos recogidos en el Cuadro II.15.

Cuadro II.15 – Valor de la clave

Objeto	Requerido	Codificación	Descripción
P	No	Gran entero	Para curvas basadas en un campo primo, P es la (p) característica del campo primo.
Q	No	Gran entero	Q es el orden del subgrupo del punto (*G*1) en el que se calculan las operaciones criptográficas.
J	No	Gran entero	J es el cofactor tal que J*Q = X-1, donde X es el orden del grupo del punto de la curva especificada.
CADENA P1	Sí	CADENA DE BYTE	Para los algoritmos basados en emparejamiento, P1 es el generador del grupo *G*1 de emparejamiento. Para el algoritmo no basado en emparejamiento, P1 es un generador del subgrupo del punto de trabajo.
CADENA P2	No	CADENA DE BYTE	Para los algoritmos basados en emparejamiento, P2 es el generador del grupo *G*2 de emparejamiento.
CADENA sP1	No	CADENA DE BYTE	sP1 es el resultado escalar de [*ib.msk*]P1 o el resultado escalar de un componente entero de *ib.msk* con P1.
CADENA sP2	No	CADENA DE BYTE	Para los algoritmos basados en emparejamiento, sP2 es el resultado escalar de [*ib.msk*]P2 o el resultado escalar de un componente entero de *ib.msk* con P2.
CADENA sP3	No	CADENA DE BYTE	Para algunos algoritmos basados en el emparejamiento (particularmente los algoritmos que utilizan la función de generación de claves BB1), sP3 es el resultado escalar de otro componente entero de *ib.msk* con P1.
CADENA de emparejamiento público	No	CADENA DE BYTE	Para algunos algoritmos basados en el emparejamiento, el emparejamiento público es el resultado de emparejar (P1, [s]P2) o de emparejar ([s]P1, P2) o de emparejar (P1, P2), donde s es *ib.msk*, para algoritmos como SM9, SK-KEM o SK-KEM o ([s1]P1, [s2]P2) para BB1-KEM, donde s1, s2 son componentes enteros de *ib.msk*.

Las nuevas definiciones de etiquetas figuran en el Cuadro II.16.

Cuadro II.16 – Definiciones de etiquetas

Objeto	Valor de la etiqueta
Tipo de emparejamiento	420100
Grupo de clave privada	420101
Nombre de dominio	420102
Serie de dominio	420103
CADENA P1	420104
CADENA sP1	420105
CADENA P2	420106
CADENA sP2	420107
CADENA sP3	420108
CADENA de emparejamiento público	420109

La composición de la carga útil requerida para señal es la del Cuadro II.17.

Cuadro II.17 – Carga útil requerida para señal

Objeto	Requerido	Descripción
Identificador único	No	El identificador único del objeto criptográfico gestionado que es la clave *ib.msk* que hay que utilizar para la operación **IBExtract**. De omitirse, el servidor utilizará el valor del marcador de posición del ID como identificador único.
Parámetros criptográficos	No	Los parámetros criptográficos pueden especificar el grupo a partir del cual se generará la clave privada.
Datos	Sí	Los datos especifican el valor de identidad a partir del cual se extraerá la clave privada.

Los parámetros criptográficos incluirán los atributos que figuran en el Cuadro II.18.

Cuadro II.18 – Parámetros criptográficos

Objeto	Requerido	Codificación	Descripción
Grupo de clave privada	No	Enumeración, véase el Cuadro II.9.	Especifica el grupo (*G*1 o *G*2) a partir del cual se generará la clave privada.

Bibliografía

[b-ITU-T E.101] Recomendación UIT-T E.101 (2009), *Definición de los términos utilizados para los identificadores (nombres, números, direcciones y otros identificadores) para los servicios y redes públicos de telecomunicación en las Recomendaciones de la serie E*.

[b-UIT-T E.212] Recomendación UIT-T E.212 (2016), *Plan de identificación internacional para redes públicas y suscripciones*.

[b-UIT-T X.509] Recomendación UIT-T X.509 (2019), *Tecnología de la información – Interconexión de sistemas abiertos – El directorio: Marcos para certificados de claves públicas y atributos*.

[b-UIT-T X.660] Recomendación UIT-T X.660 (2011), *Tecnología de la información – Procedimientos para el funcionamiento de las autoridades de registro de los identificadores de objeto: Procedimientos generales y arcos superiores del árbol de identificadores de objeto internacionales*.

[b-UIT-T X.1361] Recomendación UIT-T X.1361 (2018), *Marco de seguridad para la Internet de las cosas basado en el modelo de pasarela*.

[b-UIT-T X-Sup.31] Recomendaciones UIT-T de la serie X – Suplemento 31 (2017), *UIT-T X.660 – Suplemento sobre directrices para utilizar identificadores de objeto para la Internet de las cosas.*

[b-ITU-T Y.2720] Recomendación UIT-T Y.2720 (2009), *Marco general para la gestión de identidades en las redes de la próxima generación*.

[b-ITU-T Y.4000] Recomendación UIT-T Y.4000/Y.2060 (2012), *Visión general de Internet de las cosas*.

[b-UIT-T Y.4100] Recomendación UIT-T Y.4100/Y.2066 (2014), *Requisitos comunes de la Internet de las cosas*.

[b-ISO/IEC 9798-3] ISO/IEC 9798-3:2019. *IT Security techniques – Entity authentication – Part 3: Mechanisms using digital signature techniques*.

[b-ETSI TR 118 508] ETSI TR 118 508 V1.0.0 (2014), *Analysis of Security Solutions for the oneM2M System*.
<https://www.etsi.org/deliver/etsi_tr/118500_118599/118508/01.00.00_60/tr_118508v010000p.pdf>

[b-ETSI TS 133.501] ETSI TS 133 501 V15.2.0 (2018), *5G; Security architecture and procedures for 5G system (3GPP TS 33.501 version 15.1.0 Release 15)*.
<https://www.etsi.org/deliver/etsi_ts/133500_133599/133501/15.01.00_60/ts_133501v150100p.pdf>

[b-GM/T 0044.2] GM/T 0044.2-2016, *Identity-based cryptographic algorithms SM9 – Part 2: Digital signature algorithm*.

[b-GSMA SGP.02] GSMA Official Document SGP.02 Version 3.1 (2016), *Remote Provisioning Architecture for Embedded UICC – Technical Specification*.

[b-IANA TLS REG] Internet Assigned Numbers Authority (IANA), *Transport Layer Security (TLS) Parameters*. Website available, last viewed 2019-07-12.
<https://www.iana.org/assignments/tls-parameters/tls-parameters.xhtml>

[b-IEEE 1363] IEEE 1363-2000, *IEEE Standard Specifications for Public-Key Cryptography*.

[b-IEEE P1363.3] IEEE P1363.3/D9 (May 2013), *IEEE Standard for Identity-Based Cryptographic Techniques using Pairings*.

[b-IETF RFC 3748] IETF RFC 3748 (2004). *Extensible Authentication Protocol (EAP)*.

[b-OASIS KMIP] OASIS (2016), *Key Management Interoperability Protocol Specification Version 1.3*. <http://docs.oasis-open.org/kmip/spec/v1.3/os/kmip-spec-v1.3-os.pdf>

[b-Barreto] Barreto, P. S. L. M., Libert, B., McCullagh, N., Quisquater, J.-J. (2005). Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. En: Roy B. (director de la publicación). *Advances in Cryptology – ASIACRYPT 2005*, pp. 515-532. *Lecture Notes in Computer Science*, vol. 3788. Berlín: Springer

[b-Chen] Chen, L., Malone-Lee, J. (2005). Improved identity-based signcryption. En: Vaudenay S. (director de la publicación). *Public Key Cryptography – PKC 2005*, pp. 362-379. *Lecture Notes in Computer Science*, vol. 3386. Berlín: Springer.

[b-Ducas] Ducas, L., Lyubashevsky, V., Prest, T. (2014). Efficient identity-based encryption over NTRU lattices. En: Sarkar P., Iwata T. (directores de la publicación). *Advances in Cryptology – ASIACRYPT 2014*, pp. 22-41. *Lecture Notes in Computer Science*, vol. 8874. Berlín: Springer.

[b-Freeman] Freeman, D., Scott, M., Teske, E. (2010). A taxonomy of pairing-friendly elliptic curves. *J. Cryptol*. **23**, págs. 224-280.

[b-Galbraith] Galbraith, S.D., Paterson, K.G., Smart, N.P. (2008). Pairings for cryptographers. *Discrete Appl. Math.*, **156**, págs. 3113-3121.

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie D	Principios de tarificación y contabilidad y cuestiones económicas y políticas de las telecomunicaciones/TIC internacionales
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 multimedia
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedia
Serie K	Protección contra las interferencias
Serie L	Medio ambiente y TIC, cambio climático, ciberdesechos, eficiencia energética, construcción, instalación y protección de los cables y demás elementos de planta exterior
Serie M	Gestión de las telecomunicaciones, incluida la RGT y el mantenimiento de redes
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de la transmisión telefónica, instalaciones telefónicas y redes de líneas locales
Serie Q	Conmutación y señalización, y mediciones y pruebas asociadas
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, comunicaciones de sistemas abiertos y seguridad**
Serie Y	Infraestructura mundial de la información, aspectos del protocolo Internet, redes de próxima generación, Internet de las cosas y ciudades inteligentes
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación

Impreso en Suiza
Ginebra, 2020