OACI

Doc 9303

Documentos de viaje de lectura mecánica

Octava edición, 2021

Parte 11: Mecanismos de seguridad para los MRTD

Aprobado por la Secretaria General y publicado bajo su responsabilidad

ORGANIZACIÓN DE AVIACIÓN CIVIL INTERNACIONAL

OACI

Doc 9303

Documentos de viaje de lectura mecánica

Octava edición, 2021

Parte 11: Mecanismos de seguridad para los MRTD

Aprobado por la Secretaria General y publicado bajo su responsabilidad

ORGANIZACIÓN DE AVIACIÓN CIVIL INTERNACIONAL

Publicado por separado en español, árabe, chino, francés, inglés y ruso
por la ORGANIZACIÓN DE AVIACIÓN CIVIL INTERNACIONAL
999 Robert-Bourassa Boulevard, Montreal, Quebec, Canadá H3C 5H7

En el sitio web www.icao.int/Security/FAL/TRIP pueden obtenerse descargas
e información adicional

Doc 9303, *Documentos de viaje de lectura mecánica*
Parte 11 — *Mecanismos de seguridad para los MRTD*
Pedido núm.: 9303P11
ISBN 978-92-9265-532-7 (versión impresa)

ENMIENDAS

La publicación de enmiendas se anuncia periódicamente en los suplementos del *Catálogo de productos y servicios*; el Catálogo y sus suplementos pueden consultarse en el sitio web de la OACI: www.icao.int. Las casillas en blanco facilitan la anotación de estas enmiendas.

REGISTRO DE ENMIENDAS Y CORRIGENDAS

ENMIENDAS		
Núm.	Fecha	Anotada por

CORRIGENDAS		
Núm.	Fecha	Anotado por

Las denominaciones empleadas en esta publicación y la forma en que aparecen presentados los datos que contiene no implican, de parte de la OACI, juicio alguno sobre la condición jurídica de ninguno de los países, territorios, ciudades o áreas, o de sus autoridades, ni respecto a la delimitación de sus fronteras o límites.

ÍNDICE

Página

1. ALCANCE

En la Parte 11 del Doc 9303 se proporcionan especificaciones para permitir que los Estados y los proveedores implanten elementos de seguridad criptográfica para documentos de viaje de lectura mecánica electrónicos (“eMRTD”) que ofrecen acceso a circuito integrado (CI) sin contacto. Se especifican protocolos criptográficos para:

- impedir el despumado de datos del CI sin contacto;
- impedir la escucha furtiva de la comunicación entre el CI sin contacto y el lector;
- proporcionar autenticación de los datos almacenados en el CI sin contacto basada en la infraestructura de clave pública (PKI) que se describe en la Parte 12; y
- proporcionar autenticación del propio CI sin contacto.

La octava edición del Doc 9303 incorpora las especificaciones relativas a los registros de viaje y registros de visado opcionales y a las aplicaciones biométricas adicionales (conocidas como aplicaciones LDS2) como extensión opcional del eMRTD. Esta parte del Doc 9303 incluye los protocoles necesarios de control de acceso ampliado para proteger la escritura y lectura de los datos de las respectivas aplicaciones LDS2. Estos protocolos de control de acceso también pueden usarse para la protección de los elementos biométricos secundarios de la aplicación eMRTD.

La autenticación de los datos almacenados en el CI sin contacto es el elemento de seguridad básico para permitir el uso del CI para inspección manual o automática. Por consiguiente, este elemento ES EXIGIDO.

ES OBLIGATORIO implantar un protocolo para impedir el despumado de los datos almacenados en el CI sin contacto así como la escucha furtiva de la comunicación entre el CI y el terminal.

La implantación de los otros protocolos ES OPCIONAL, permitiéndose al Estado expedidor u organización expedidora que determine el conjunto de elementos de seguridad necesarios con arreglo a los reglamentos o exigencias nacionales.

Esta Parte deberá leerse conjuntamente con las siguientes partes del Doc 9303:

- Parte 1 — *Introducción*;
- Parte 10 — *Estructura lógica de datos (LDS) para el almacenamiento de datos biométricos y de otro tipo en el circuito integrado (CI) sin contacto*; y
- Parte 12 — *Infraestructura de clave pública para los MRTD*.

2. HIPÓTESIS Y NOTACIONES

Se supone que el lector de este documento está familiarizado con los conceptos y mecanismos ofrecidos por la criptografía de clave pública y las infraestructuras de clave pública.

Si bien el uso de técnicas criptográficas de clave pública añade complicaciones en la implantación de eMRTD, dichas técnicas añaden también valor en el sentido de que proporcionarán a los puntos de control fronterizo de primera línea una medida adicional para determinar la autenticidad del eMRTD. Se supone que su utilización no es la única medida para determinar la autenticidad y que NO DEBERÍA considerarse como único factor determinante en el cual basarse.

En el caso de que los datos del CI sin contacto no puedan utilizarse, por ejemplo, como resultado de una revocación de certificados o de una verificación de firma inválida, o si el CI sin contacto fue dejado intencionalmente en blanco (véase la sección 4.5.4 del Doc 9303-10), el eMRTD no es necesariamente invalidado. En tales casos un Estado receptor PODRÍA basarse en otras características de seguridad del documento para fines de validación.

2.1 Requisitos para microplaquetas y terminales de eMRTD

En esta Parte del Doc 9303 se especifican requisitos para implantaciones de microplaquetas (o, lo que es equivalente, el CI) y terminales (o sistemas de inspección) de eMRTD. Si bien las microplaquetas de eMRTD deben cumplir dichos requisitos con arreglo a la terminología descrita en el 9303-1, los requisitos para terminales deben interpretarse como orientación, es decir, el interfuncionamiento de la microplaqueta del eMRTD y el terminal solo se garantizan si este último cumple dichos requisitos, de otra forma la interacción con la microplaqueta del eMRTD o bien fallará o bien el comportamiento de microplaqueta del eMRTD será indefinido. En general, la microplaqueta del eMRTD no tiene por qué cumplir requisitos relacionados con los terminales a menos que la seguridad de dicha microplaqueta se vea directamente afectada.

2.2 Notaciones

Las siguientes notaciones se utilizan para denotar primitivas criptográficas en una manera que no depende del algoritmo:

- cifrado de texto claro S con clave simétrica K: **E** (K, S);
- descifrado de texto C cifrado con clave simétrica K: **D** (K, C);
- la operación de calcular una condensación sobre un mensaje *m* se denota por **H** *(m)*.
- cálculo del código de autenticación de mensaje con clave simétrica K sobre el mensaje M: **MAC** (K, M);
- acuerdo de claves basado en pares de claves asimétricas (SK, PK) y (SK', PK') y parámetros de dominio D: **KA**(SK, PK',D) / **KA**(SK',PK,D);
- obtención de clave a partir de un secreto S compartido: **KDF** *(S);*
- la firma de un mensaje *m* con la clave privada S_{KIFD} se indica mediante la *s* = **Sign**(S_{KIFD}, *m*);
- verificación de la firma resultante *s* con la clave pública P_{KIFD} y el mensaje *m*: **Verify**(P_{KIFD}, *s*, *m*).
- cómputo de la representación comprimida de una clave pública *PK*: **Comp***(PK).*

3. SEGURIDAD DE LOS DATOS ELECTRÓNICOS

Además de la autenticación pasiva mediante firmas digitales y el control de acceso a la microplaqueta, los Estados expedidores o las organizaciones expedidoras PUEDEN optar por una seguridad adicional, utilizando formas más complejas de proteger el CI sin contacto y sus datos.

El acceso a un eMRTD comprende las etapas siguientes:

1. Obtención de acceso al CI sin contacto del eMRTD (sección 4)
2. Autenticación de los datos (sección 5)
3. Autenticación de la microplaqueta (sección 6)
4. Mecanismos de control de acceso adicionales (sección 7)
5. Lectura de los datos (véase el Doc 9303-10).

Se dispone de diferentes protocolos para las diferentes etapas. La configuración exacta del eMRTD es determinada por el Estado expedidor u organización expedidora. Las opciones presentadas en la tabla 1 pueden combinarse adecuadamente para lograr una seguridad adicional con arreglo a las necesidades de los expedidores.

En el apéndice J se describen los procedimientos de inspección para diferentes configuraciones de los eMRTD.

Tabla 1. Seguridad de los datos electrónicos (Resumen)

Método	***CI sin contacto***	***Sistema de inspección***	***Ventajas***	***Nota***
MÉTODO DE SEGURIDAD BÁSICO				
Autenticación pasiva (sección 5.1)	m	m	Prueba que el contenido de la SO_D y la LDS son auténticos y no se han modificado.	No impide una copia exacta o sustitución de CI. No impide acceso no autorizado. No impide despumado.
MÉTODOS DE SEGURIDAD AVANZADOS				
Comparación de ZLM convencional (OCR-B) y ZLM basada en IC (LDS)	n/a	o	Prueba que el contenido del CI sin contacto y el eMRTD físico se corresponden.	Añade complejidad (menor). No impide una copia exacta del CI sin contacto y del documento convencional.
Autenticación activa (sección 6.1)	o	o	Impide la copia del SO_D y prueba que ha sido leído del CI sin contacto auténtico. Prueba que el CI sin contacto no ha sido substituido.	No impide el acceso no autorizado. Añade complejidad. Para la LDS2 se EXIGE la autenticación de microplaqueta.
Autenticación de la microplaqueta (sección 6.2)	o/c	o		

Método	*CI sin contacto*	*Sistema de inspección*	*Ventajas*	*Nota*
Control de acceso de base (BAC) (sección 4.3)	c (véase también 4.1)	m (véase también 4.1)	Impide el despumado y el uso impropio. Impide la escucha furtiva de las comunicaciones entre el eMRTD y el sistema de inspección (cuando se usa para establecer un canal de sesión cifrado).	No impide una copia exacta o substitución de CI (también requiere copiar el documento convencional). Añade complejidad. Al menos uno, el BAC o el PACE, SERÁ admitido por el MRTD. El PACE se EXIGE para el LDS2. PACE ofrece mejor protección contra la intromisión que BAC. Véase también el Apéndice A.
Establecimiento de conexión autenticada por contraseña (PACE) (sección 4.4)	r/c (véase también 4.1)	r (véase también 4.1)		
Autenticación del terminal (sección 7.1)	o	o	Impide el acceso no autorizado a datos sensibles. Impide el despumado de datos sensibles.	Exige gestión de claves adicional. No impide una copia exacta o substitución del CI (también exige copiar el documento convencional). Añade complejidad. La autenticación del terminal es OBLIGATORIA para el LDS2
Cifrado de datos (Sección 7.2)	o	o	Protege las características biométricas adicionales. No exige CI con procesador.	Exige gestión de claves de descifrado complejas. No impide una copia exacta o substitución del CI. Añade complejidad.
m = OBLIGATORIO, r = RECOMENDADO, o = OPCIONAL, c = CONDICIONAL, n/a = no aplicable.				

Nota.— En la sección 4 figuran detalles sobre configuraciones normalizadas de CI sin contacto con respecto a la implantación del control de acceso de base y al establecimiento de conexión autenticada por contraseña.

La implantación de los métodos de seguridad avanzados que figuran en la Tabla 1 no afecta el cumplimiento de las normas de la OACI.

4. ACCESO AL CI SIN CONTACTO

La adición de un CI sin contacto y sin control de acceso a un eMRTD introduce dos nuevas posibilidades de ataque:

- los datos almacenados en el CI sin contacto pueden leerse electrónicamente sin que se haya autorizado esta lectura del documento (despumado); y
- la comunicación no cifrada entre un CI sin contacto y un lector puede escucharse furtivamente a una distancia de varios metros.

Aunque hay medidas físicas posibles para impedir el despumado (p. ej., blindaje utilizando una malla metálica en la cubierta de una libreta pasaporte), estas no se aplican a la escucha furtiva. Por consiguiente, se entiende que los Estados expedidores u organizaciones expedidoras DEBERÁN optar por implantar un mecanismo de control de acceso a la microplaqueta, es decir un mecanismo de control de acceso que en efecto requiera el conocimiento del titular del eMRTD de que los datos almacenados en el CI sin contacto son leídos en forma segura. Este mecanismo de control de acceso a la microplaqueta impide el despumado así como la escucha furtiva.

Un CI sin contacto protegido por un mecanismo de control de acceso a la microplaqueta impide el acceso a su contenido a menos que el sistema de inspección pueda probar que está autorizado para acceder al CI sin contacto. Esta prueba se obtiene en un protocolo criptográfico, en el que el sistema de inspección demuestra conocer la información obtenida del documento físico.

DEBE proporcionarse esta información al sistema de inspección antes de que pueda leer el CI sin contacto. La información debe obtenerse en forma óptica/visual del eMRTD (p. ej., de la ZLM). También DEBE ser posible para un inspector ingresar esta información manualmente en el sistema de inspección en caso de no poderse leer mecánicamente la información.

Suponiendo que en la información del documento físico no pueda obtenerse de un documento cerrado (p. ej., dado que la información se obtiene de la ZLM de lectura óptica), se acepta que el eMRTD fue entregado a inspección con pleno conocimiento. Debido al cifrado del canal, la escucha furtiva de la comunicación exigiría un esfuerzo considerable.

En esta sección se definen dos mecanismos para el control de acceso a la microplaqueta:

- Control de acceso de base (BAC, sección 4.3), basado exclusivamente en criptografía simétrica; y
- Establecimiento de conexión autenticada por contraseña (PACE, sección 4.4), que emplea criptografía asimétrica para proporcionar claves de sesión de entropía mayor.

En el Apéndice A figura información adicional sobre la fuerza de las claves de sesión.

4.1 Configuraciones conformes

Las siguientes configuraciones cumplen esta especificación:

- microplaquetas de eMRTD con BAC solamente;
- microplaquetas de eMRTD con PACE *y* BAC;
- a partir del 1 de enero de 2018, microplaquetas de eMRTD con PACE solamente.

Nota.— El BAC puede resultar obsoleto en el futuro. En este caso, el PACE será el mecanismo de control de acceso por defecto.

Los sistemas de inspección conformes DEBEN apoyar todas las configuraciones del eMRTD conformes. Si un eMRTD apoya PACE y BAC, el sistema de inspección UTILIZARÁ el BAC o el PACE pero no ambos en la misma sesión.

Nota 1.— Versiones anteriores del Doc 9303 aceptaban las microplaquetas de eMRTD sin control de acceso a la microplaqueta (“eMRTD simples”). Esto se considera obsoleto en la octava edición. No obstante, los sistemas de inspección conformes DEBEN apoyar eMRTD sin control de acceso a la microplaqueta.

Nota 2.— Para el acceso a las aplicaciones LDS2, el IC DEBE requerir también la ejecución del PACE.

4.2 Procedimiento de acceso a la microplaqueta

El procedimiento de acceso a la microplaqueta para autenticar el sistema de inspección consiste en las etapas siguientes.

1. **Lectura de EF.CardAccess** **(EXIGIDO)**

Si el PACE es apoyado por el eMRTD, la microplaqueta de éste DEBE proporcionar los parámetros que han de utilizarse para el PACE en el fichero EF.CardAccess.

Si se dispone de EF.CardAccess, el sistema de inspección LEERÁ el fichero EF.CardAccess (véase la sección 9.2.11) para determinar los parámetros (es decir, cifrado simétrico, algoritmos de acuerdo de clave, parámetros de dominio y correspondencias) apoyados por la microplaqueta eMRTD. El sistema de inspección puede seleccionar cualquiera de estos parámetros.

Si no se dispone del fichero EF.CardAccess o éste no contiene parámetros para PACE, el sistema de inspección DEBERÍA tratar de leer el eMRTD con control de acceso de base (omitir las etapas hasta la 4).

2. **Lectura de EF.DIR** **(OPCIONAL)**

El sistema de inspección PUEDE leer el EF.DIR (si está presente) para recuperar una lista de aplicaciones presentes en la microplaqueta del eMRTD.

3. **PACE** **(CONDICIONAL)**

Esta etapa se RECOMIENDA si la microplaqueta del eMRTD apoya PACE. Este paso se EXIGE si se pretende acceder a las aplicaciones LDS2.

- El sistema de inspección DEBERÍA obtener la clave K_{π} de la ZLM. En vez de la ZLM PUEDE usar el número de acceso a la tarjeta (CAN) si éste es conocido por el sistema de inspección.

- La microplaqueta del eMRTD ACEPTARÁ la ZLM como contraseñas para PACE. También PUEDE aceptar el CAN en lugar de la ZLM.

- El sistema de inspección y la microplaqueta del eMRTD autentican mutuamente el uso de K_{π} y obtienen las claves de sesión KS_{Enc} y KS_{MAC}. se UTILIZARÁ el protocolo PACE según se describe en la sección 4.4.

Si tiene éxito, la microplaqueta del eMRTD ejecuta lo siguiente:

- INICIARÁ la construcción segura de mensajes (Construcción segura de mensajes).

- DARÁ acceso a datos menos sensibles (p. ej., EF.DG1, EF.DG2, EF.DG14, EF.DG15, etc. de la aplicación eMRTD, y al objeto de seguridad de documento. Para la definición de “datos sensibles”, véase el Doc 9303-1).

- RESTRINGIRÁ los derechos de acceso para exigir protección segura de mensajes.

El sistema de inspección DEBE verificar la autenticidad del contenido del fichero EF.CardAccess utilizando el EF.DG14 o el EF.CardSecurity, y del EF.DIR (si está presente y se ha leído) utilizando el EF.CardSecurity.

Nota.— Si en la microplaqueta del eMRTD no está presente ninguna aplicación LDS2, puede que el EF.CardSecurity no contenga una copia segura del EF.DIR.

4. **Control de acceso de base** **(CONDICIONAL)**

Esta etapa es EXIGIDA si la microplaqueta del eMRTD impone el control de acceso a la microplaqueta y no se ha utilizado PACE. Si PACE se ejecutó satisfactoriamente o si el eMRTD no impone el control de acceso a la microplaqueta, puede omitirse esta etapa.

La aplicación eMRTD DEBE seleccionarse antes de que se lleve a cabo el control de acceso de base.

- El sistema de inspección DEBERÍA obtener las claves de acceso de base al documento (K_{Enc} y K_{MAC}) de la ZLM.

- El sistema de inspección y la microplaqueta del eMRTD autentican mutuamente el uso de las claves de acceso de base al documento y obtienen las claves de sesión KS_{Enc} y KS_{MAC}.

Si tiene éxito, la microplaqueta del eMRTD ejecuta lo siguiente:

- INICIARÁ la construcción segura de mensajes.

- OTORGARÁ acceso a datos menos sensibles (p. ej., EF.DG1, EF.DG2, EF.DG14, EF.DG15, etc. de la aplicación eMRTD, y al objeto de seguridad de documento).

- RESTRINGIRÁ los derechos de acceso para exigir construcción segura de mensajes.

Nota.— Como resultado del procedimiento de acceso a la microplaqueta, el actual DF puede ser o bien el fichero maestro (si se usa el PACE) o la aplicación eMRTD (si se usa el BAC).

4.3 Control de acceso de base

4.3.1 Especificación del protocolo

La autenticación y el establecimiento de claves se proporcionan mediante un protocolo de puesta a prueba-respuesta de tres pasadas con arreglo a [ISO/IEC 11770-2] mecanismo de establecimiento de claves 6 utilizando 3DES [FIPS 46-3] como cifra en bloque. Una suma de control criptográfica con arreglo a [ISO/IEC 9797-1] El algoritmo MAC 3 se calcula y se añade a los textos cifrados. DEBEN utilizarse los modos de operación descritos en la sección 4.3.3. Los números aleatorios utilizados solo una vez (nonces) intercambiados DEBEN ser de 8 bytes, el material de cifrado intercambiado DEBE ser de 16 bytes. El IFD (i.e., el sistema de inspección) y el CI sin contacto NO DEBEN utilizar identificadores distintivos como nonces.

Con más detalle, el IFD y el CI EJECUTARÁN las etapas siguientes:

1) El IFD pide una puesta a prueba en el RND.IC enviando el comando GET CHALLENGE. El CI genera y responde con un nonce RND.IC.

2) El IFD ejecuta las operaciones siguientes:

   a) genera un nonce RND.IFD y material de cifrado K.IFD.

   b) genera la concatenación S = RND.IFD || RND.IC || K.IFD.

   c) calcula el criptograma E_{IFD} = **E**($K_{Enc,}$ S).

d) calcula la suma de control M_{IFD} = **MAC**($K_{MAC,}$ E_{IFD}).

e) envía el comando EXTERNAL AUTHENTICATE con función de autenticación mutua utilizando los datos E_{IFD} || M_{IFD}.

3) El CI ejecuta las operaciones siguientes:

a) verifica la suma de control M_{IFD} del criptograma E_{IFD}.

b) descifra el criptograma E_{IFD}.

c) extrae el RND.IC de S y verifica si IFD devolvió el valor correcto.

d) genera material de cifrado K.IC.

e) genera la concatenación R = RND.IC || RND.IFD || K.IC.

f) calcula el criptograma E_{IC} = **E**(K_{Enc}, R).

g) calcula la suma de control M_{IC} = **MAC**(K_{MAC}, E_{IC}).

h) envía la respuesta utilizando los datos E_{IC} || M_{IC}.

4) El IFD ejecuta las operaciones siguientes:

a) verifica la suma de control M_{IC} del criptograma E_{IC}.

b) descifra el criptograma E_{IC}.

c) extrae RND.IFD de R y verifica si el CI devolvió el valor correcto.

5) El IFD y el CI obtienen las claves de sesión KS_{Enc} y KS_{MAC} utilizando el mecanismo de obtención de claves que se describe en las secciones 9.7.1 y 9.7.4 con (K.IC xor K.IFD) como secreto compartido.

4.3.2 Proceso de inspección

Cuando un eMRTD con control de acceso de base se ofrece al sistema de inspección, la información leída óptica o visualmente se utiliza para obtener las claves de acceso de base al documento (K_{Enc} y K_{MAC}) a efectos de acceder al CI sin contacto y establecer un canal seguro de comunicación entre el CI sin contacto del eMRTD y el sistema de inspección.

Un CI sin contacto del eMRTD que apoya el control de acceso de base DEBE responder a los intentos de lectura no autenticados, es decir, intentos de lectura enviados sin construcción segura de mensajes [incluyendo la selección de ficheros (protegidos) en la LDS], con el mensaje “condición de seguridad no satisfecha” (0x6982) una vez establecido el canal seguro. Si el CI recibe un comando SELECT, es decir, sin construcción segura de mensajes aplicada, en el canal seguro, el CI ELIMINARÁ el canal seguro. Cuando se envía un comando SELECT antes de establecerse el canal seguro, o cuando el canal seguro ha sido eliminado, el CI PUEDE devolver los 0x6982 y 0x9000, es decir, son respuestas conformes a la OACI.

Para autenticar el sistema de inspección DEBEN ejecutarse las etapas siguientes:

1) El sistema de inspección lee la “Información ZLM” (MRZ_information) que consiste en la concatenación del número de documento, fecha de nacimiento y fecha de caducidad, incluyendo sus respectivos dígitos de verificación, según se describen en el Doc 9303-4, Doc 9303-5 o Doc 9303-6 para los formatos de documento DV3, DV1 y DV2, respectivamente, de la ZLM utilizando un lector OCR-B. Alternativamente, la información requerida puede escribirse; en este caso, se ESCRIBIRÁ como aparece en la ZLM. Los 16 bytes más significativos de la condensación SHA-1 de esta “Información ZLM” se utilizan como semilla de claves para obtener las claves de acceso de base al documento utilizando el mecanismo de obtención de clave que se describe en la sección 9.7.2.

2) El sistema de inspección y el CI sin contacto del eMRTD autentican y obtienen mutuamente claves de sesión. DEBE utilizarse el protocolo de autenticación y establecimiento de claves descrito anteriormente.

3) Después de una autenticación exitosa del protocolo de autenticación tanto el IFD como el CI calculan las claves de sesión KS_{Enc} y KS_{MAC} utilizando un mecanismo de extensión de claves que se describe en las secciones 9.7.1 y 9.7.4 con (K.IC xor K.IFD) como secreto compartido. Todas las comunicaciones subsiguientes DEBEN estar protegidas por la construcción segura de mensajes según se describe en la sección 9.8.

4.3.3 Especificaciones criptográficas

4.3.3.1 Cifrado de puesta a prueba y respuesta

Para el cálculo de E_{IFD} y E_{IC} se utilizarán 3DES de dos claves en modo CBC con IV cero (es decir, 0x00 00 00 00 00 00 00 00) con arreglo a [ISO/IEC 11568-2]. NO DEBE utilizarse relleno para los datos de entrada al ejecutar el comando EXTERNAL AUTHENTICATE.

4.3.3.2 Autenticación de puesta a prueba y respuesta

Las sumas de control criptográficas M_{IFD} y M_{IC} se CALCULARÁN utilizando el algoritmo 3 MAC de [ISO/IEC 9797-1] con cifra en bloque DES, IV cero (8 bytes) y el método de relleno 2 de [ISO/IEC 9797-1]. La longitud MAC DEBE ser de 8 bytes.

4.3.4 Unidades de datos del protocolo de aplicación

El control de acceso de base se ejecuta utilizando los comandos GET CHALLENGE y EXTERNAL AUTHENTICATE con función de autenticación mutua. Los comandos se CODIFICARÁN según se especifica en [ISO/IEC 7816-4].

4.3.4.1 OBTENER PUESTA A PRUEBA (GET CHALLENGE)

Comando		
CLA		Específica de contexto
INS	0x84	GET CHALLENGE
P1/P2	0x0000	—
Datos		*Ausentes*
Respuesta		
Datos	Nonce aleatorio	
Bytes de estado	0x9000	*Procesamiento normal* Nonce aleatorio general y transmitido satisfactoriamente.
	Otros	*Error dependiente del sistema operativo* No pudo transmitirse el nonce aleatorio.

4.3.4.2 AUTENTICACIÓN EXTERNA (EXTERNAL AUTHENTICATE)

Comando			
CLA		Específica de contexto	
INS	0x82	EXTERNAL AUTHENTICATE	
P1/P2	0x0000	—	
Datos		Datos de comando E_{IFD} || M_{IFD}	EXIGIDO
Respuesta			
Datos		Datos de respuesta E_{IC} || M_{IC}	EXIGIDO
Bytes de estado	0x9000	*Procesamiento normal* El protocolo se ha ejecutado con éxito.	
	Otros	*Error dependiente del sistema de operación* El protocolo falló.	

4.4 Establecimiento de conexión autenticada por contraseña

PACE es un protocolo de acuerdo de claves Diffie-Hellman autenticado por contraseña que proporciona comunicación segura y autenticación basada en contraseña de la microplaqueta del eMRTD y del sistema de inspección (es decir, que la microplaqueta del eMRTD y el sistema de inspección comparten la misma contraseña π).

PACE establece una construcción segura de mensajes entre una microplaqueta de eMRTD y un sistema de inspección basada en contraseñas débiles (breves). El contexto de seguridad se establece en el fichero maestro. El protocolo permite que la microplaqueta del eMRTD verifique que el sistema de inspección está autorizado para acceder a los datos almacenados y tiene las características siguientes:

- Se proporcionan claves de sesión fuertes independientemente de la fuerza de la contraseña.
- La entropía de las contraseñas utilizadas para autenticar el sistema de inspección puede ser muy baja (p. ej., en general alcanza con 6 dígitos).

PACE utiliza claves K_π obtenidas de contraseñas mediante una función de obtención de claves $\mathbf{KDF_\pi}$ (véase la sección 9.7.3). Para los documentos de viaje de lectura mecánica de interfuncionamiento mundial se dispone de las siguientes dos contraseñas y sus claves correspondientes:

- ZLM: la clave K_π definida por $K_\pi = \mathbf{KDF_\pi}(\text{ZLM})$ es EXIGIDA. Se obtienen de la zona de lectura mecánica (ZLM) en forma similar al control de acceso de base, es decir, la clave se obtiene del número de documento, la fecha de nacimiento y la fecha de caducidad.
- CAN: la clave K_π definida por $K_\pi = \mathbf{KDF_\pi}(\text{CAN})$ es OPCIONAL. Se obtiene del número de acceso de la tarjeta (CAN). El CAN es un número impreso en el documento y DEBE escogerse en forma aleatoria o pseudoaleatoria (p. ej., utilizando una función pseudoaleatoria criptográficamente fuerte). Las Partes 4, 5 y 6 del Doc 9303 especifican el campo CAN.

Nota.— En contraste con la ZLM (número de documento, fecha de nacimiento, fecha de caducidad) el CAN tiene la ventaja de que puede escribirse manualmente con facilidad.

PACE apoya dos correspondencias diferentes como parte de la ejecución del protocolo:

- *correspondencia genérica* basada en el acuerdo de claves Diffie-Hellman;
- *correspondencia integrada* basada en la correspondencia directa de un elemento del campo con el grupo criptográfico;
- *correspondencia de autenticación de microplaqueta* amplía la correspondencia genérica e integra la autenticación de microplaqueta en el protocolo PACE.

Si la microplaqueta apoya la correspondencia de autenticación de microplaqueta, por lo menos una correspondencia genérica o correspondencia integrada y autenticación de microplaqueta DEBE también ser apoyada por la microplaqueta. Esto entraña que, para los sistemas de inspección que apoyan PACE, solo se EXIGE apoyo para correspondencia genérica y correspondencia integrada. El apoyo para la correspondencia de autenticación de microplaqueta es OPCIONAL.

4.4.1 Especificación de protocolo

El sistema de inspección lee los parámetros para PACE apoyados por la microplaqueta del eMRTD del fichero EF.CardAccess (véase la sección 9.2.11) y seleccione los parámetros que han de utilizarse, seguido de la ejecución del protocolo.

Se UTILIZARÁN los comandos siguientes:

- READ BINARY según se especifica en el Doc 9303-10;

- MSE:Set AT (comando MANAGE SECURITY ENVIRONMENT con función de plantilla de autenticación Set) según se especifica en la sección 4.4.4.1;

- Los pasos siguientes se EJECUTARÁN por el sistema de inspección y la microplaqueta del eMRTD utilizando una cadena de comandos GENERAL AUTHENTICATE según se especifica en la sección 4.4.4.2:

  1) La microplaqueta del eMRTD elige en forma aleatoria y uniforme un nonce *s*, cifra el nonce a $z = \mathbf{E}(K_\pi, s)$, donde $K_\pi = \mathbf{KDF}_\pi$ (π) se obtiene de la contraseña compartida π, y envía el texto cifrado *z* al sistema de inspección.

  2) Sistema de inspección recupera el texto simple $s = \mathbf{D}(K_\pi, z)$ con ayuda de la contraseña compartida π.

  3) Tanto la microplaqueta del eMRTD como el sistema de inspección ejecutan los pasos siguientes:

     a) Intercambio en datos adicionales requeridos para la correspondencia del nonce:

        i) para la correspondencia genérica, la microplaqueta del eMRTD y el sistema de inspección intercambian claves públicas efímeras.

        ii) para la correspondencia integrada, el sistema de inspección envía un nonce adicional a la microplaqueta del eMRTD.

     b) Calculan los parámetros de dominio efímero $D = \mathbf{Map}(D_{IC}, s, \ldots)$ según se describe en la sección 4.4.3.3.

     c) Ejecutan un acuerdo de clave Diffie-Hellman anónimo (véase la sección 9.6) basado en los parámetros de dominio efímero y generan el secreto compartido $K = \mathbf{KA}(S_{KDH,IC}, P_{KDH,IFD,D}) = \boldsymbol{KA}(SK_{DH,IFD}, P_{KDH,IC}, D)$.

     d) Durante el acuerdo de clave Diffie-Hellman, el CI y el sistema de inspección DEBERÍAN verificar que las dos claves públicas $PK_{DH,IC}$ y $PK_{DH,IFD}$ son diferentes.

     e) Obtienen claves de sesión $KS_{MAC} = \mathbf{KDF}_{\mathrm{MAC}}(K)$ y $KS_{Enc} = \mathbf{KDF}_{\mathrm{Enc}}(K)$ según se describe en la sección 9.7.1.

     f) Intercambian y verifican el testigo (token) de autenticación $T_{IFD} = \boldsymbol{MAC}(K_{SMAC,PKDH,IC})$ y $T_{IC} = \boldsymbol{MAC}(K_{SMAC}, P_{KDH,IFD})$ según se describe en la sección 4.4.3.4.

  4) Condicionalmente, la microplaqueta eMRTD calcula los datos de autenticación de microplaqueta CA_{IC}, los cifra y $A_{IC} = \mathbf{E}(KS_{Enc}, CA_{IC})$ y los envía al terminal (véase la sección 4.4.3.5.1). El terminal descifra A_{IC} y verifica la autenticidad de la microplaqueta utilizando los datos de autenticación de microplaqueta recuperado CA_{IC} (véase la sección 4.4.3.5.2).

En la figura 1 también se muestra una versión simplificada del protocolo.

Microplaqueta denotada por IC		***Sistema de inspección denotado por IFD***
Parámetros de dominio estático D_{IC} Elegir nonce s aleatorio Calcular z = **E**(K_π,s)		
	— z →	
		Calcular s = **D**(K_π,z)
	← datos adicionales para **Map** →	
D = **Map**(D_{IC},s,…) Elegir par de claves efímeras aleatorias ($SK_{DH,IC}$, $PK_{DH,IC}$,D)		D = **Map**(D_{IC},s,…) Elegir par de claves efímeras aleatorias ($SK_{DH,IFD}$, $PK_{DH,IFD}$,D)
	← $PK_{DH,IC}$, $PK_{DH,\ IFD}$ →	
Verificar $PK_{DH,IC} \neq PK_{DH,\ IFD}$ K = **KA**($SK_{DH,IC}$, $PK_{DH,\ IFD}$,D) Calcular T_{IC} = **MAC**(KS_{MAC}, $PK_{DH,IFD}$) [calcular CA_{IC} y cifrar como A_{IC} = **E**(KS_{Enc}, CA_{IC}).]		Verificar $PK_{DH,IC} \neq PK_{DH,IFD}$ K = **KA**($SK_{DH,IFD}$, $PK_{DH,IC}$,D) Calcular T_{IFD} = **MAC**(KS_{MAC}, $PK_{DH,IC}$)
	← T_{IC}, T_{IFD} → [— A_{IC} →]	
Verificar T_{IFD}		Verificar T_{IC} [descifrar A_{IC} y autenticar microplaqueta]

Figura 1. Establecimiento de conexión autenticada por contraseña

4.4.2 Condición de seguridad

Un eMRTD que apoye PACE RESPONDERÁ a los intentos de lectura no autenticados [incluyendo la selección de ficheros (protegidos) en la LDS] con el mensaje “condición de seguridad no satisfecha” (0x6982).

Nota.— Esta especificación es más restrictiva que la especificación correspondiente para eMRTD con BAC solamente.

Si PACE fue ejecutado con éxito, la microplaqueta eMRTD ha verificado la contraseña utilizada. La construcción segura de mensajes se inicia utilizando las claves de sesión obtenidas KS_{MAC} y KS_{Enc}.

4.4.3 Especificaciones criptográficas

En esta sección figuran los detalles criptográficos de la especificación.

Los algoritmos particulares son seleccionados por el Estado expedidor u organización expedidora. El sistema de inspección DEBE apoyar todas las combinaciones que se describen en las subsecciones siguientes, con la excepción de la correspondencia de autenticación de microplaqueta, que es OPCIONAL. La microplaqueta del eMRTD PUEDE admitir más de una combinación de algoritmos.

Nota.— Algunos algoritmos no están disponibles para la correspondencia de autenticación de microplaqueta: por razones de seguridad, ya no se recomienda el uso de 3DES. No se dispone de variantes de DH para reducir el número de variantes que han de implantar los terminales.

4.4.3.1 DH

Para PACE con DH DEBEN utilizarse los respectivos algoritmos y formatos de la sección 9.6 y de la tabla 2.

Tabla 2. Algoritmos y formatos para DH

OID	*Correspon-dencia*	*Cifrado simétrico*	*Longitud de clave*	*Construcción segura de mensajes*	*Testigo de autenti-cación*
id-PACE-DH-GM-3DES-CBC-CBC	Genérica	3DES	112	CBC / CBC	CBC
id-PACE-DH-GM-AES-CBC-CMAC-128	Genérica	AES	128	CBC / CMAC	CMAC
id-PACE-DH-GM-AES-CBC-CMAC-192	Genérica	AES	192	CBC / CMAC	CMAC
id-PACE-DH-GM-AES-CBC-CMAC-256	Genérica	AES	256	CBC / CMAC	CMAC
id-PACE-DH-IM-3DES-CBC-CBC	Integrada	3DES	112	CBC / CBC	CBC
id-PACE-DH-IM-AES-CBC-CMAC-128	Integrada	AES	128	CBC / CMAC	CMAC
id-PACE-DH-IM-AES-CBC-CMAC-192	Integrada	AES	192	CBC / CMAC	CMAC
id-PACE-DH-IM-AES-CBC-CMAC-256	Integrada	AES	256	CBC / CMAC	CMAC

4.4.3.2 ECDH

Para PACE con ECDH DEBEN utilizarse los respectivos algoritmos y formatos de la sección 9.6 y de la tabla 3.

Sólo se UTILIZARÁN curvas de números primos. DEBERÍAN utilizarse los parámetros de dominio normalizados que se describen en la sección 9.5.1.

Tabla 3. Algoritmos y formatos para ECDH

OID	***Correspon-dencia***	***Cifrado simétrico***	***Longitud de clave***	***Construcción segura de mensajes***	***Testigo de autenti-cación***
id-PACE-ECDH-GM-3DES-CBC-CBC	Genérica	3DES	112	CBC / CBC	CBC
id-PACE-ECDH-GM-AES-CBC-CMAC-128	Genérica	AES	128	CBC / CMAC	CMAC
id-PACE-ECDH-GM-AES-CBC-CMAC-192	Genérica	AES	192	CBC / CMAC	CMAC
id-PACE-ECDH-GM-AES-CBC-CMAC-256	Genérica	AES	256	CBC / CMAC	CMAC
id-PACE-ECDH-IM-3DES-CBC-CBC	Integrada	3DES	112	CBC / CBC	CBC
id-PACE-ECDH-IM-AES-CBC-CMAC-128	Integrada	AES	128	CBC / CMAC	CMAC
id-PACE-ECDH-IM-AES-CBC-CMAC-192	Integrada	AES	192	CBC / CMAC	CMAC
id-PACE-ECDH-IM-AES-CBC-CMAC-256	Integrada	AES	256	CBC / CMAC	CMAC
id-PACE-ECDH-CAM-AES-CBC-CMAC-128	Autentica-ción de micro-plaqueta	AES	128	CBC / CMAC	CMAC
id-PACE-ECDH-CAM-AES-CBC-CMAC-192		AES	192	CBC / CMAC	CMAC
id-PACE-ECDH-CAM-AES-CBC-CMAC-256		AES	256	CBC / CMAC	CMAC

4.4.3.3 Nonces de cifrado y correspondencia

La microplaqueta eMRTD SELECCIONARÁ en forma aleatoria y uniforme el nonce *s* como cadena de bits binarios de longitud *l*, donde *l* es múltiplo del tamaño de bloque en bits del cifrado de bloque **E**() respectivo elegido por la microplaqueta del eMRTD.

- El nonce s se CIFRARÁ en modo CBC con arreglo a [ISO/IEC 10116] utilizando la clave $K_\pi = KDF_\pi(\pi)$ obtenida de la contraseña π y IV puesto a la cadena de todos 0.
- El nonce s se CONVERTIRÁ en generador aleatorio utilizando una función de correspondencia **Map** específica del algoritmo.
- Para la correspondencia integrada se seleccionará un nonce *t* adicional en forma aleatoria y uniforme como cadena de bits binarios de longitud *k* y se enviará en claro. En este caso *k* es el tamaño de clave en bits del respectivo cifrado en bloque **E**() y *l* SERÁ el menor múltiplo del tamaño de bloque de **E**() de modo que $l \geq k$.

Para ejecutar la correspondencia del nonce *s* o los nonces *s*, *t* en el grupo criptográfico se UTILIZARÁ uno de los siguientes modos de correspondencia:

- *Correspondencia genérica* (sección 4.4.3.3.1);
- *Correspondencia integrada* (sección 4.4.3.3.2);
- *Correspondencia de autenticación de mensaje* (sección 4.4.3.3.3).

4.4.3.3.1 Correspondencia genérica

ECDH

La función **Map**:G → Ĝ se define como Ĝ = s×G+H, donde H en <G> se elige de modo que se desconoce el $\log_G H$. El punto H se CALCULARÁ mediante un acuerdo de clave Diffie-Hellman anónimo [TR-03111] como H = **KA**$(SK_{Map,IC}, PK_{Map,\ IFD}, D_{IC})$ = **KA**$(SK_{Map,IFD}, PK_{Map,\ IC}, D_{IC})$.

Nota.— El algoritmo de acuerdo de clave ECKA impide pequeños ataques de subgrupo utilizando multiplicación de cofactores compatibles.

DH

La función **Map**:g → ĝ se define como $\hat{g} = g^s \times h$, donde h en <g> se elige de modo que se desconoce el $\log_g h$. El elemento h del grupo se CALCULARÁ mediante un acuerdo de clave Diffie-Hellman anónimo como h = **KA**$(SK_{Map,IC}, PK_{Map,IFD}, D_{IC})$ = **KA**$(SK_{Map,IFD}, PK_{Map,IC}, D_{IC})$.

Nota.— El método de validación de clave pública que se describe en [RFC 2631] DEBE utilizarse para impedir ataques de subgrupos pequeños.

4.4.3.3.2 Correspondencia integrada

ECDH

La función **Map**:G → Ĝ se define como $\hat{G} = f_G(\mathbf{R}_p(s,t))$, donde $\mathbf{R}_p()$ es una función pseudoaleatoria que transforma cadenas de octetos en elementos de GF(p) y $f_G()$ es una función que transforma elementos de GF(p) a <G>. El nonce t aleatorio se ELEGIRÁ en forma aleatoria por el sistema de inspección y se enviará a la microplaqueta eMRTD. La función pseudoaleatoria $\mathbf{R}_p()$ se describe a continuación. La función $f_G()$ se define en [BCIMRT2010]. En el Apéndice B figura una descripción informativa al respecto.

DH

La función **Map**:g → ĝ se define como $\hat{g} = f_g(\mathbf{R}_p(s,t))$, donde $\mathbf{R}_p()$ es una función pseudoaleatoria que transforma cadenas de octetos en elementos de GF(p) y $f_g()$ es la función que transforma elementos de GF(p) en <g>. El nonce t aleatorio se ELEGIRÁ en forma aleatoria por el sistema de inspección y se enviará a la microplaqueta eMRTD. La función pseudoaleatoria $\mathbf{R}_p()$ se describe a continuación. La función $f_g()$ se define como $f_g(x)=x^a \bmod p$, y a = (p-1)/q es el cofactor. Las implantaciones DEBEN verificar $\hat{g} \neq 1$.

Correspondencia de números pseudoaleatorios

La función $\mathbf{R}_p(s,t)$ es una función que transforma cadenas de octetos s (de longitud de bits l) y t (de longitud de bits k) en un elemento $\text{int}(x_1||x_2||...||x_n) \bmod p$ of GF(p). La función $\mathbf{R}_p(s,t)$ se especifica a continuación en la figura 2.

La construcción se basa en el respectivo cifrado de bloques **E**() en modo CBC con arreglo a [ISO/IEC 10116] con IV=0, donde k es el tamaño de clave (en bits) de **E**(). Cuando se requiera, el resultado k_i DEBE truncarse para llevarlo a un tamaño de clave k. El valor n se SELECCIONARÁ como número más pequeño, de modo que $n*l \geq \log_2 p + 64$.

Nota.— El truncamiento solo es necesario para AES-192: Se emplean octetos de 1 a 24 de k_i; no se utilizan octetos adicionales. En caso de DES, k se considera igual a 128 bits, y el resultado de R(s,t) será de 128 bits.

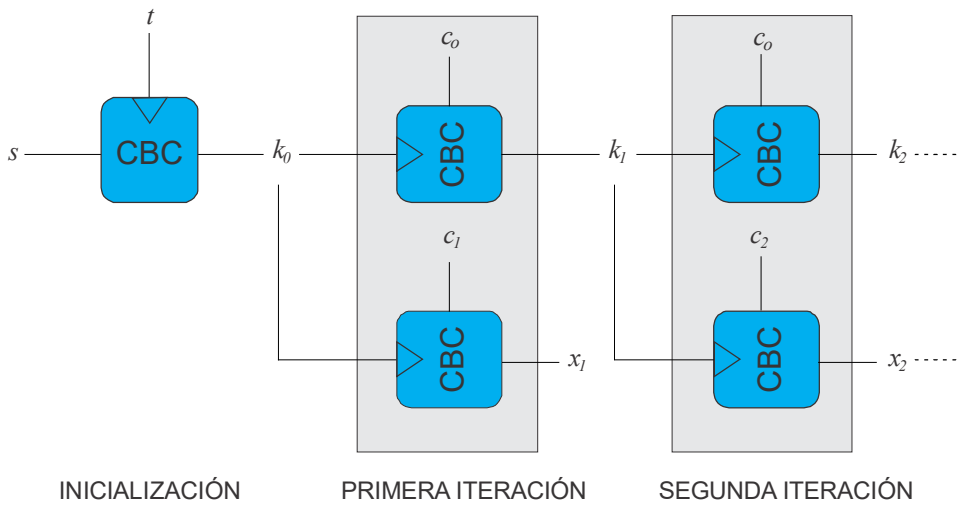

Figura 2. Correspondencia de números pseudoaleatorios

Las constantes c_0 y c_1 se definen como sigue:

- Para 3DES y AES-128 (l=128):
  - `c0=0xa668892a7c41e3ca739f40b057d85904`
  - `c1=0xa4e136ac725f738b01c1f60217c188ad`
- Para AES-192 y AES-256 (l=256):
  - `c0=`
    `0xd463d65234124ef7897054986dca0a174e28df758cbaa03f240616414d5a1676`
  - `c1=`
    `0x54bd7255f0aaf831bec3423fcf39d69b6cbf066677d0faae5aadd99df8e53517`

4.4.3.3.3 Correspondencia de autenticación de microplaqueta

La fase de correspondencia del PACE-CAM es idéntica a la fase de correspondencia de PACE-GM (véase la sección 4.4.3.3.1).

4.4.3.4 Testigo de autenticación

El testigo de autenticación se CALCULARÁ sobre un objeto de datos de clave pública (véase la sección 9.4) que contiene el identificador de objeto según se indica en MSE:Set AT (véase la sección 4.4.4.1) y la clave pública efímera recibida (es decir, excluyendo los parámetros de dominio, véase la sección 9.4.5) utilizando un código de autenticación y la clave KS_{MAC} obtenida del acuerdo de claves.

Nota.— El código de autenticación de mensajes ejecuta internamente el relleno, es decir, no se ejecuta relleno específico de la aplicación.

3DES

Se UTILIZARÁ 3DES [FIPS 46-3] en modo minorista con arreglo a [ISO/IEC 9797-1] MAC algoritmo 3 / relleno método 2 con cifrado de bloque DES y IV=0.

AES

La AES [FIPS 197] se UTILIZARÁ en modo CMAC [SP 800-38B] con una longitud MAC de 8 bytes.

4.4.3.5 Datos de autenticación de microplaqueta cifrados

La microplaqueta eMRTD DEBE proporcionar pares de claves estáticas SK_{IC}, PK_{IC} según se describe en la sección 6.2. Los datos de autenticación de claves cifrados se EXIGEN para PACE con correspondencia de autenticación de microplaqueta.

4.4.3.5.1 Generación por la microplaqueta eMRTD

Los datos de autenticación de microplaqueta se CALCULARÁN como $CA_{IC} = (SK_{IC})^{-1} * SK_{Map,IC} \bmod p$, donde SK_{IC} es la clave privada estática de la microplaqueta, $SK_{Map,IC}$ es la clave privada efímera utilizada por la microplaqueta para calcular H en la etapa de correspondencia de PACE (véase la sección 4.4.3.3.1) y p es el orden del grupo criptográfico utilizado. Los datos de autenticación de microplaqueta se CIFRARÁN utilizando la clave KS_{Enc} obtenida del acuerdo de clave como $A_{IC} = \mathbf{E}(KS_{Enc}, CA_{IC})$ para producir los datos de autenticación de microplaqueta cifrado.

Nota.— $(SK_{IC})^{-1}$ *puede calcularse previamente durante la personalización de la microplaqueta eMRTD y almacenarse en forma segura en la microplaqueta, evitando la inversión modular durante el tiempo de ejecución.*

4.4.3.5.2 Verificación por el terminal

El terminal DESCIFRARÁ A_{IC} para recuperar CA_{IC} y verificar $PK_{Map,IC} = \mathbf{KA}(CA_{IC}, PK_{IC}, D_{IC})$, donde PK_{IC} es la clave pública estática de la microplaqueta eMRTD.

Nota.— La autenticación pasiva DEBE realizarse en combinación con la correspondencia de autenticación de microplaqueta. Solo después de una validación exitosa del objeto de seguridad respectivo puede considerarse que la microplaqueta eMRTD es genuina.

4.4.3.5.3 Relleno

Los datos que han de cifrarse se RELLENARÁN con arreglo a [ISO/IEC 9797-1] “Método de relleno 2”.

4.4.3.5.4 AES

La AES [19] se UTILIZARÁ en modo CBC con arreglo a [ISO/IEC 10116] con $IV=E(KS_{Enc},-1)$, donde *-1* es la cadena de bits de longitud 128 con todos los bits puestos a 1.

4.4.4 Unidades de datos de protocolo de aplicación

Se UTILIZARÁ la siguiente secuencia de comandos para implantar PACE:

1. MSE:Set AT
2. GENERAL AUTHENTICATE

4.4.4.1 MSE:Set AT

El comando MSE:Set AT se utiliza para seleccionar e inicializar el protocolo PACE. El uso de MSE:Set AT para el PACE se indica mediante un identificador de objeto PACE (véanse las secciones 4.4.3 y 9.2.3) que figura como referencia de mecanismo criptográfico con el rótulo 0x80, como se muestra en la siguiente tabla.

Comando			
CLA		Específico de contexto	
INS	0x22	Gestionar entorno de seguridad	
P1/P2	0xC1A4	Establecer plantillas de autenticación para autenticación mutua	
Datos	0x80	*Referencia de mecanismo criptográfico* Identificador de objeto de protocolo a seleccionar (valor solamente, se omite el rótulo 0x06).	EXIGIDO
	0x83	*Referencia de una clave pública/clave secreta* La contraseña que se ha de utilizar se indica mediante los siguientes valores de este objeto de datos: 0x01: MRZ_information 0x02: CAN	EXIGIDO
	0x84	*Referencia a una clave privada/Referencia para calcular una clave de sesión.* Este objeto de datos se EXIGE para indicar el identificador de los parámetros de dominio que ha de utilizarse si los parámetros son ambiguos, es decir, más de un conjunto de parámetros de dominio disponible para PACE.	CONDICIONAL
	0x7F4C	*Plantilla de autorización de la persona titular del certificado* Este objeto de datos (definido en el Doc 9303-12) DEBE estar presente si la terminal pide referencia(s) de la autoridad de certificación para su uso en la autenticación del terminal con vistas a su devolución como parte del PACE (cf. sección 4.4.5). El identificador de objeto que figura en este objeto de datos se PONDRÁ en id-IS (cf. Doc 9303-10). La terminal PONDRÁ todos los bits de acceso de la plantilla de datos discrecionales a 1.	CONDICIONAL
Respuesta			
Datos	–	Ausentes	
Bytes de estado	0x9000	*Procesamiento normal* El protocolo se ha seleccionado e inicializado.	

	0x6A80	*Parámetros incorrectos en el campo de datos de comandos* El algoritmo no se apoya o la inicialización falló.
	0x6A88	*Datos de referencia no hallados* Los datos de referencia (es decir, contraseña o parámetro de dominio) no están disponibles.
	otros	*Error dependiente del sistema de operación* La inicialización del protocolo falló.

Nota 1.— Algunos sistemas de operación aceptan la selección de una clave no disponible y devuelven un error sólo cuando la clave se utiliza para el fin seleccionado.

Nota 2.— Para el comando MSE:Set, el IC DEBERÍA hacer caso omiso de los objetos de datos con rótulos no especificados para este comando. La terminal NO DEBERÍA incluir objetos de datos con rótulos desconocidos para que los entienda el IC.

4.4.4.2 GENERAL AUTHENTICATE

Se utiliza una cadena de comandos GENERAL AUTHENTICATE para ejecutar el protocolo PACE.

Comando			
CLA		Específico de contexto.	
INS	0x86	GENERAL AUTHENTICATE	
P1/P2	0x0000	Claves y protocolo implícitamente conocidos	
Datos	0x7C	*Datos de autenticación dinámica* Objetos de datos específicos del protocolo	EXIGIDO
Respuesta			
Datos	0x7C	*Datos de autenticación dinámica* Objetos de datos específicos del protocolo como se describe en la sección 4.4.5.	EXIGIDO
Bytes de estado	0x9000	*Procesamiento normal* El protocolo (etapa) tuvo éxito.	
	0x6300	*La autenticación falló* El protocolo (etapa) falló.	
	0x6A80	*Parámetros incorrectos en el campo de datos de comandos* Los datos proporcionados son inválidos.	
	otros	*Error dependiente del sistema de operación* El protocolo (etapa) falló.	

4.4.4.3 Encadenamiento de órdenes

El encadenamiento de comandos DEBE utilizarse para que el comando GENERAL AUTHENTICATE enlace la secuencia de comandos con la ejecución del protocolo. El encadenamiento de órdenes NO DEBE utilizarse para otros fines a menos que se indique claramente en la microplaqueta. En la [ISO/IEC 7816-4] figuran detalles sobre el encadenamiento de comandos.

4.4.5 Datos intercambiados

Los objetos de datos específicos del protocolo se INTERCAMBIARÁN en una cadena de comandos GENERAL AUTHENTICATE, con los datos de comando y respuesta específicos del protocolo encapsulado en un objeto de datos de autenticación dinámica (véase la sección 4.4.4.2) con rótulos específicos de contexto según se indica en la tabla 4:

Tabla 4. Datos intercambiados para PACE

Etapa	***Descripción***	***Datos de comando del protocolo***		***Datos de respuesta del protocolo***	
1.	Nonce cifrado	-	Ausentes[1]	0x80	Nonce cifrado
2.	Nonce de correspondencia	0x81	Datos de correspondencia	0x82	Datos de correspondencia
3.	Ejecución de acuerdo de clave	0x83	Clave pública efímera	0x84	Clave pública efímera
4.	Autenticación mutua	0x85	Testigo de autenticación	0x86	Testigo de autenticación
				0x87	Referencia de la autoridad de certificación (CONDICIONAL)
				0x88	Referencia de la autoridad de certificación (CONDICIONAL)
				0x8A	Datos de autenticación de microplaqueta cifrados (CONDICIONAL)

La(s) referencia(s) de la autoridad de certificación DEBE(N) estar presente(s) si se transmitió un objeto de datos 0x7F4C al IC durante el establecimiento del PACE (cf. sección 4.4.4.1) y la autenticación del terminal es admitida por el IC. En este caso, el objeto de datos 0x87 CONTENDRÁ la referencia de la autoridad de certificación más reciente. El objeto de datos 0x88 PUEDE contener la referencia de la autoridad de certificación previa.

Los datos de autenticación de microplaqueta cifrados (véase la sección 4.4.3.5) DEBEN estar presentes si se utiliza correspondencia de autenticación de microplaqueta y NO DEBEN estar presentes en otros casos.

1. Esto implica un objeto de datos de autenticación dinámica vacío.

4.4.5.1 Nonce cifrado

El nonce cifrado (véase la sección 4.4.3.3) se CODIFICARÁ como cadena de octeto.

4.4.5.2 Datos de correspondencia

Los datos intercambiados son específicos de la correspondencia utilizada:

4.4.5.2.1 Correspondencia genérica

Las claves públicas efímeras (véanse la sección 4.4.3.3 y la sección 9.4.5 se CODIFICARÁN como punto de curva elíptica (ECDH) o entero sin signo (DH).

4.4.5.2.2 Correspondencia integrada

El nonce *t* se CODIFICARÁ como cadena de octetos.

Nota.— El objeto de datos específico del contexto 0x82 ESTARÁ vacío para la correspondencia integrada.

4.4.5.2.3 Correspondencia de autenticación de microplaqueta

La codificación de los datos de correspondencia es idéntica a la correspondencia genérica (véase la sección 4.4.5.2.1).

4.4.5.3 Claves públicas

Las claves públicas se CODIFICARÁN según se describe en la sección 9.4.5

4.4.5.4 Testigo de autenticación

El testigo de autenticación (véase la sección 4.4.3.4) se CODIFICARÁ como cadena de octetos.

4.4.5.5 Referencia de la autoridad de certificación

Los objetos de datos de la referencia de la autoridad de certificación (CAR) SE CODIFICARÁN según se especifica en el Doc 9303-12.

4.4.5.6 Datos de autenticación de microplaqueta

Los datos de autenticación de microplaqueta se CODIFICARÁN como cadena de octetos utilizando la función FE2OS() especificado en [TR-03111] antes del cifrado. Obsérvese que FE2OS() exige que la codificación tenga el mismo número de octetos que el orden primo del grupo, es decir, posiblemente incluyendo 0x00 iniciales. Los datos de autenticación de microplaqueta cifrados se CODIFICARÁN como cadena de octetos.

5. AUTENTICACIÓN DE LOS DATOS

Además de los grupos de datos de LDS, el CI sin contacto también contiene un objeto de seguridad de documento (SO_D). Este objeto se firma en forma digital por el Estado expedidor u organización expedidora y contiene representaciones condensadas del contenido de la LDS (véase el Doc 9303-10).

Un sistema de inspección, que contenga la clave pública del firmante de documentos de cada Estado, o que haya leído el certificado del firmante de documentos (C_{DS}) en el eMRTD, podrá verificar el objeto de seguridad de documento (SO_D). De esta manera, mediante el contenido del objeto de seguridad de documento (SO_D), se autentica el contenido de la LDS.

Este mecanismo de verificación no exige capacidades de procesamiento en el CI sin contacto del eMRTD. Por consiguiente, se denomina “autenticación pasiva” del contenido del CI.

La autenticación pasiva prueba que el contenido del objeto de seguridad de documento (SO_D) y la LDS es auténtico y no se ha modificado. No impide la copia exacta del contenido del CI sin contacto o la sustitución de la microplaqueta.

Por consiguiente, un sistema de autenticación pasiva DEBERÍA estar apoyado por una inspección física adicional del eMRTD.

5.1 Autenticación pasiva

5.1.1 Proceso de inspección

El sistema de inspección ejecuta las etapas siguientes:

1. El sistema de inspección LEERÁ del CI sin contacto el objeto de seguridad de documento (SO_D) [que DEBE contener el certificado del firmante del documento (C_{DS}), véase también el Doc 9303-10].

2. El sistema de inspección CONSTRUIRÁ y VALIDARÁ un trayecto de certificación desde un punto de confianza al certificado del firmante del documento utilizado para firmar el objeto de seguridad de documento (SO_D) con arreglo al Doc 9303-12.

3. El sistema de inspección UTILIZARÁ la clave pública del firmante del documento verificada para revisar la firma del objeto de seguridad de documento (SO_D).

4. El sistema de inspección PUEDE leer del CI sin contacto los grupos de datos pertinentes.

5. El sistema de inspección GARANTIZARÁ que el contenido del grupo de datos es auténtico y no se ha modificado mediante condensación de dicho contenido y comparación del resultado con el valor de condensación correspondiente en el objeto de seguridad de documento (SO_D).

Las siguientes verificaciones adicionales se consideran mejores prácticas:

1. El sistema de inspección o el funcionario inspector DEBERÍA verificar la presencia de una DocumentTypeExtension en el certificado del firmante del documento.

    - En caso afirmativo, el sistema de inspección DEBERÍA verificar la coherencia del DocumentTypeExtension, el tipo de documento del Grupo de datos 1 y el tipo de documento de la ZLM (véanse los Docs 9303-12, 9303-10 y 9303-3, respectivamente).

- En caso negativo, el sistema de inspección DEBERÍA verificar que el KeyUsage del certificado de firmante de documentos está establecido en digitalSignature y que el certificado del firmante del documento no contiene ExtendedKeyUsage-Extension (véase el Doc 9303-12).

2. El sistema de inspección o el funcionario inspector DEBERÍA verificar la coherencia de los códigos de país en:

   - el campo Asunto y, si está presente, el SubjectAltName del certificado del firmante del documento;
   - el campo Asunto y, si está presente, el SubjectAltName del punto de confianza (certificado CSCA);
   - el Grupo de datos 1 leído del CI sin contacto; y
   - la ZLM visual.

   Además, el sistema de inspección o el funcionario inspector PUEDE comparar el contenido del Grupo de datos 1 con la ZLM visual (véanse los Docs 9303-12, 9303-10 y 9303-3, respectivamente).

3. El sistema de inspección DEBERÍA verificar que la fecha de expedición del eMRTD está incluida en el período de uso de la clave privada que figura en el certificado del firmante del documento (véase el Doc 9303-12).

La información biométrica puede utilizarse ahora para que ejecute la verificación de las características biométricas con la persona que presenta el eMRTD.

5.1.2 Proceso de inspección adicional para las aplicaciones LDS2

Los datos escritos después de la expedición del eMRTD no están protegidos por el objeto de seguridad del documento, que está firmado por el expedidor del documento. Para verificar la autenticidad de los datos escritos después de la expedición, el sistema de inspección DEBE realizar los siguientes pasos para cada objeto de datos escrito.

1. El sistema de inspección CONSTRUIRÁ y VALIDARÁ un trayecto de certificación desde un punto de confianza al certificado de firmante utilizado para firmar el objeto de datos con arreglo al Doc 9303-12. El sistema de inspección PUEDE usar ambos certificados, conocidos de antemano, y los certificados recuperados de la microplaqueta para construir el trayecto (véase el Doc 9303-10).

2. El sistema de inspección UTILIZARÁ la clave pública verificada de firmante para verificar la firma del objeto de datos.

Nota.— Este procedimiento puede omitirse en el caso de los objetos de datos cuya autenticidad el Estado receptor o la organización receptora no consideran relevante para el proceso de inspección.

6. AUTENTICACIÓN DEL CI SIN CONTACTO

El Estado expedidor u organización expedidora PUEDE optar por proteger sus eMRTD contra sustitución de microplaqueta.

Se dispone de los siguientes mecanismos para verificar la autenticidad de la microplaqueta.

1. *Autenticación activa*, como se define en la sección 6.1. El apoyo de autenticación activa se indica por la presencia del EF.DG15. Si está disponible, el terminal PUEDE leer y verificar EF.DG15 y ejecutar la autenticación activa.

2. *Autenticación de microplaqueta*, como se define en la sección 6.2. El apoyo de la autenticación de microplaqueta se indica por la presencia de los correspondientes `SecurityInfos` en EF.DG14/EF.CardSecurity. Si se dispone del mismo, el terminal PUEDE leer y verificar el EF.DG14/EF.CardSecurity.CardSecurity y ejecutar la autenticación de microplaqueta.

3. *PACE con correspondencia de autenticación de microplaqueta* (PACE-CAM) según se define en la sección 4.4. El apoyo está indicado por la presencia de una estructura `PACEInfo` correspondiente en EF.CardAccess. Si el PACE-CAM se ejecutó con éxito en el procedimiento de acceso a la microplaqueta, el terminal PUEDE ejecutar lo siguiente para autenticar la microplaqueta:

   - leer y verificar EF.CardSecurity;

   - utilizar la clave pública de EF.CardSecurity conjuntamente con los datos de correspondencia y los datos de autenticación de microplaquetas recibidos como parte de PACE-CAM para autenticar la microplaqueta (sección 4.4.3.5.2).

6.1 Autenticación activa

La autenticación activa autentica el CI sin contacto firmando una puesta a prueba enviada por el IFD (sistema de inspección) con una clave privada conocida solamente por el CI.

Para este fin, el CI sin contacto contiene su propio par de claves de autenticación activa (KPr_{AA} y KPu_{AA}). Una representación de condensación del Grupo de datos 15 [información de clave pública (KPu_{AA})] se almacena en el objeto de seguridad de documento (SO_D) y por consiguiente es autenticado por la firma digital del expedidor. La clave privada (KPr_{AA}) correspondiente se almacena en la memoria segura del CI sin contacto.

Mediante la autenticación de la ZLM visual [a través de la ZLM condensada en el objeto de seguridad de documento (SO_D)] en combinación con la respuesta a la puesta a prueba, utilizando el par de claves de autenticación activa (KPr_{AA} y KPu_{AA}) del eMRTD, el sistema de inspección verifica que el objeto de seguridad de documento (SO_D) se ha leído del CI sin contacto genuino, almacenado en el eMRTD genuino.

La autenticación activa requiere capacidades de procesamiento en el CI sin contacto del eMRTD.

6.1.1 Especificación del protocolo

La autenticación activa se ejecuta utilizando el comando INTERNAL AUTHENTICATE de [ISO/IEC 7816-4].

Si la autenticación activa se realiza después de haberse establecido la construcción segura de mensajes, todos los comandos y respuestas DEBEN transmitirse como APDU de construcción segura de mensajes con arreglo a la sección 9.8.

Más detalladamente, el IFD (sistema de inspección) y el CI (CI sin contacto del eMRTD) ejecuta las etapas siguientes:

1. El IFD genera un nonce RND.IFD y lo envía al CI utilizando el comando INTERNAL AUTHENTICATE.

2. El CI ejecuta las operaciones siguientes:

   a) genera el mensaje M;

b) calcula h(M);

c) calcula la firma σ y envía la respuesta al IFD.

3. El IFD verifica la respuesta al comando INTERNAL AUTHENTICATE enviada y verifica si el CI devolvió el valor correcto.

6.1.2 Especificaciones criptográficas

6.1.2.1 Nonce

La entrada es un nonce (RND.IFD) que DEBE ser de 8 bytes.

Nota.— Los nonces NO DEBEN volver a utilizarse, p. ej., el nonce utilizado para BAC/PACE NO DEBE ser reutilizado para autenticación activa.

6.1.2.2 RSA

El CI CALCULARÁ una firma, cuando se utilice un mecanismo basado en factorización de enteros, con arreglo a [ISO/IEC 9796-2] plan 1 de firma digital.

En lo que sigue, k indica la longitud de clave para generación de firma y L_h la longitud de la salida de la función de condensación H utilizada durante la generación de firma. La opción 1 del indicador de fin DEBE utilizarse (y poner t a 1) si se utiliza SHA-1 durante la generación de firma; la opción 2 del indicador de fin DEBE utilizarse en todos los demás casos (y t ponerse a 2).

En la opción 2 se USARÁN los siguientes valores para el indicador de fin:

Función de condensación	SHA-224	SHA-256	SHA-384	SHA-512
Indicador de fin	0x38CC	0x34CC	0x36CC	0x35CC

Por motivos de interoperabilidad, solo SHA-1, SHA-224, SHA-256, SHA-384 y SHA-512 son admitidos como funciones de condensación para la autenticación activa con RSA.

El mensaje M que ha de firmarse SERÁ la concatenación de M1 y M2, donde M1 DEBE ser un nonce de longitud c - 4 bits (RND.IC) generado por el eMRTD, donde c (*capacidad de la firma*) está dada por $c = k - L_h - (8 \times t) - 4$, y M2 es RND.IFD generado por el sistema de inspección.

El resultado del cálculo de la firma DEBE ser una firma σ sin la parte no recuperable M2 del mensaje.

Los eMRTD DEBERÍAN implantar el plan de generación de firmas especificado en [ISO/IEC 9796-2], párrafo B.6, y no DEBERÍAN utilizar el plan de generación de firmas especificado en [ISO/IEC 9796-2], párrafo B.4. Los eMRTD NO IMPLANTARÁN otros planes de generación de firma.

Los sistemas de inspección IMPLANTARÁN el plan de generación de firmas especificado en [ISO/IEC 9796-2], párrafo B.6, y DEBERÍAN implantar el plan de generación de firma especificado en [ISO/IEC 9796-2], párrafo B.4.

6.1.2.3 ECDSA

Para ECDSA, se utilizará el formato de firma sencillo con arreglo a [TR-03111]. Sólo se utilizarán curvas de primos con puntos sin comprimir. Se UTILIZARÁ un algoritmo de condensación, cuya longitud de salida es igual o menor que la longitud de la clave ECDSA en uso. Solo SHA-224, SHA-256, SHA-384 o SHA-512 son admitidos como funciones de condensación. No se USARÁN RIPEMD-160 ni SHA-1.

El mensaje M que ha de firmarse es el nonce RND.IFD proporcionado por el sistema de inspección.

6.1.3 Unidades de datos del protocolo de aplicación

La autenticación activa se ejecuta mediante una única invocación del comando INTERNAL AUTHENTICATE según se especifica en [ISO/IEC 7816-4].

Comando			
CLA		Específico de contexto	
INS	0x88	INTERNAL AUTHENTICATE	
P1/P2	0x0000	—	
Datos		*RND.IFD*	EXIGIDO
Respuesta			
Datos		Firma σ generada por el CI	EXIGIDO
Bytes de estado	0x9000	*Procesamiento normal* El protocolo se ha ejecutado con éxito.	
	Otros	*Error dependiente del sistema de operación* El protocolo falló.	

6.1.4 Claves de autenticación activa

Los pares de claves de autenticación activa (KPr_{AA} y KPu_{AA}) SERÁN generados en forma segura.

Tanto la clave pública de autenticación activa (KPu_{AA}) como la clave privada de autenticación activa (KPr_{AA}) se almacenan en el CI sin contacto del eMRTD. Después de eso, no se aplica gestión de clave para estas claves.

Nota.— Cabe señalar que cuando se utilizan longitudes de clave superiores a 1 848 bits (si se usa construcción segura de mensajes con 3DES)/1 792 bits (si se utiliza construcción segura de mensajes con AES) en la autenticación activa con construcción segura de mensajes, las APDU de longitud ampliada DEBEN estar apoyadas por la microplaqueta de eMRTD y el sistema de inspección.

Los Estados expedidores u organizaciones expedidoras ELEGIRÁN longitudes de clave apropiadas que ofrezcan protección contra ataques durante la vida útil del eMRTD. DEBERÍAN tenerse en cuenta catálogos criptográficos adecuados.

6.1.5 Información de clave pública de autenticación activa

La clave pública de autenticación activa se almacena en el Grupo de datos 15 de la LDS. El formato de la estructura (`SubjectPublicKeyInfo`) se especifica en [RFC 5280], véase la sección 9.1. Todos los objetos de seguridad DEBEN producirse en formato de regla de codificación distinguida (DER) para preservar la integridad de las firmas que contienen.

```
ActiveAuthenticationPublicKeyInfo ::= SubjectPublicKeyInfo
```

6.1.6 Proceso de inspección

Cuando un eMRTD con el Grupo de datos 15 se ofrece al sistema de inspección, el mecanismo de autenticación activa PUEDE aplicarse para asegurar que los datos se leen del CI sin contacto genuino y que este y el documento físico se corresponden.

El sistema de inspección y el CI sin contacto ejecutan las etapas siguientes:

1. La ZLM en su totalidad se lee visualmente del eMRTD (si ya no se ha leído como parte del procedimiento de control de acceso de base) y se compara con el valor ZLM en el Grupo de datos 1. Dado que la autenticidad e integridad del Grupo de datos 1 ya han sido verificadas mediante la autenticación pasiva, la similitud asegura que la ZLM visual es auténtica y no ha sido modificada.

2. La autenticación pasiva también ha demostrado la autenticidad e integridad del Grupo de datos 15. Esto asegura que la clave pública de autenticación activa (KPu_{AA}) es auténtica y no ha sufrido modificaciones.

3. Para asegurar que el objeto de seguridad de documentos (SO_D) no es una copia, el sistema de inspección utiliza el par de claves de autenticación activa (KPr_{AA} y KPu_{AA}) del eMRTD en un protocolo de puesta a prueba-respuesta con el CI sin contacto del eMRTD, como se describe anteriormente.

Una aplicación positiva del protocolo de puesta a prueba-respuesta comprueba que el objeto de seguridad de documento (SO_D) corresponde al documento físico, el CI sin contacto es genuino y que este y el documento físico se corresponden.

6.2 Autenticación de microplaqueta

El protocolo de autenticación de microplaqueta es un protocolo de acuerdo de clave Diffie-Hellman efímero y estático que proporciona comunicación segura y autenticación unilateral de la microplaqueta del eMRTD.

as diferencias principales con respecto a la autenticación activa son:

- Se impide la semántica de puesta a prueba porque las transcripciones producidas por este protocolo no son transferibles.

- Además de la autenticación de la microplaqueta del eMRTD, este protocolo también proporciona claves de sesión fuertes.

En el Apéndice C se describen detalles sobre la semántica de puesta a prueba.

Los pares de clave de autenticación de microplaquetas estáticos DEBEN almacenarse en la microplaqueta del eMRTD.

- La clave privada se ALMACENARÁ en condiciones de seguridad en la memoria de la microplaqueta del eMRTD.

- La clave pública se PROPORCIONARÁ como `SubjectPublicKeyInfo` en la estructura `ChipAuthenticationPublicKeyInfo` (véase la sección 9.2.6).

Este protocolo proporciona autenticación implícita de la propia microplaqueta del eMRTD y de los datos almacenados ejecutando confusión segura de mensajes por medio de las nuevas claves de sesión.

Si el IC admite la autenticación de microplaqueta, PUEDE admitir esa autenticación en el fichero maestro y/o en la aplicación eMRTD. Si la autenticación de microplaqueta se usa junto con el acceso a los grupos de datos en las aplicaciones LDS2, el IC DEBE apoyar la autenticación de microplaqueta en el fichero maestro.

Nota.— Si se exige la compatibilidad con el mecanismo de control de acceso ampliado de la Unión Europea [TR-03110], el IC DEBE admitir la autenticación del terminal en la aplicación eMRTD.

6.2.1 Especificación del protocolo

El terminal y la microplaqueta del eMRTD ejecutan las etapas siguientes.

1. La microplaqueta del eMRTD envía al terminal su clave pública Diffie-Hellman estática PK_{IC} y los parámetros de dominio D_{IC}.

2. El terminal genera un par de claves Diffie-Hellman efímero $(SK_{DH,IFD}, PK_{DH,IFD}, D_{IC})$ y envía la clave pública efímera $PK_{DH,\ IFD}$ a la microplaqueta del eMRTD.

3. La microplaqueta del eMRTD y el terminal calculan lo siguiente:

   a) La clave secreta compartida K = **KA**$(SK_{IC}, PK_{DH,IFD}, D_{IC})$ = **KA**$(SK_{DH,IFD}, PK_{IC}, D_{IC})$

   b) Las claves de sesión KS_{MAC} = **KDF**$_{MAC}(K)$ and KS_{Enc} = **KDF**$_{Enc}(K)$ obtenidas de K para la construcción segura de mensajes.

En la figura 3 se muestra una versión simplificada:

CI (microplaqueta)		***IFD (sistema de inspección)***
Par de claves estáticas $(SK_{IC}, PK_{IC}, D_{IC})$		
	— PK_{IC}, D_{IC} →	
		Elección de par de claves efímeras aleatorias $(SK_{DH,IFD}, PK_{DH,IFD}, D_{IC})$
	← $PK_{DH,IFD}$ —	
K = **KA**$(SK_{IC}, PK_{DH,IFD}, D_{IC})$		K = **KA**$(SK_{DH,IFD}, PK_{IC}, D_{IC})$

Figura 3. Autenticación de microplaqueta

Para verificar la autenticidad de la PK_{IC} el terminal EJECUTARÁ autenticación pasiva.

6.2.2 Estado de seguridad

Si la autenticación de microplaqueta se realizó con éxito, se reinicia la construcción segura de mensajes utilizando las claves de sesión obtenidas KS_{MAC} y KS_{Enc}. En caso contrario, la construcción segura de mensajes continúa utilizando las claves de sesión establecidas previamente (PACE o control de acceso de base).

Nota.— La autenticación pasiva DEBE realizarse en combinación con la autenticación de microplaqueta. Sólo después de una exitosa validación del objeto de seguridad respectivo puede considerarse genuina la microplaqueta del eMRTD.

6.2.3 Especificaciones criptográficas

El Estado expedidor u organización expedidora selecciona algoritmos particulares. El sistema de inspección DEBE apoyar todas las combinaciones que se describen en las subsecciones siguientes. La microplaqueta del eMRTD PUEDE apoyar más de una combinación de algoritmos.

6.2.3.1 Autenticación de microplaqueta con DH

Para la autenticación de microplaqueta con DH DEBEN utilizarse los algoritmos y formatos respectivos de la sección 9.6. Para las claves públicas, DEBE utilizarse PKCS#3 [PKCS#3] en vez de X9.42 [X9.42].

Tabla 5. Identificadores de objetos para autenticación de microplaqueta con DH

OID	*Cifrado simétrico*	*Longitud de clave*	*Construcción segura de mensajes*
id-CA-DH-3DES-CBC-CBC	3DES	112	CBC / CBC
id-CA-DH-AES-CBC-CMAC-128	AES	128	CBC / CMAC
id-CA-DH-AES-CBC-CMAC-192	AES	192	CBC / CMAC
id-CA-DH-AES-CBC-CMAC-256	AES	256	CBC / CMAC

6.2.3.2 Autenticación de microplaqueta con ECDH

Para la autenticación de microplaqueta con ECDH DEBEN utilizarse los algoritmos y formatos respectivos de la sección 9.6.

Tabla 6. Identificadores de objeto para autenticación de microplaqueta con ECDH

OID	*Cifrado simétrico*	*Longitud de clave*	*Construcción segura de mensajes*
id-CA-ECDH-3DES-CBC-CBC	3DES	112	CBC / CBC
id-CA-ECDH-AES-CBC-CMAC-128	AES	128	CBC / CMAC
id-CA-ECDH-AES-CBC-CMAC-192	AES	192	CBC / CMAC
id-CA-ECDH-AES-CBC-CMAC-256	AES	256	CBC / CMAC

6.2.4 Unidades de datos del protocolo de aplicaciones

Dependiendo del algoritmo simétrico que se utilice, se dispone de dos implantaciones de autenticación de microplaqueta.

- El comando siguiente se UTILIZARÁ para implantar la autenticación de microplaqueta con construcción segura de mensajes 3DES:

  1. MSE:Set KAT

- La siguiente secuencia de comandos se UTILIZARÁ para implantar autenticación de microplaqueta con construcción segura de mensajes AES y PUEDE utilizarse para implantar autenticación de microplaqueta con construcción segura de mensajes 3DES:

  1. MSE:Set AT
  2. GENERAL AUTHENTICATE

6.2.4.1 Implantación utilizando MSE:Set KAT

Nota.— MSE:Set KAT sólo puede utilizarse para `id-CA-DH-3DES-CBC-CBC` *y para* `id-CA-ECDH-3DES-CBC-CBC`*, es decir, la construcción segura de mensajes se restringe a 3DES.*

Comando			
CLA		Específico de contexto	
INS	0x22	Gestionar entorno de seguridad	
P1/P2	0x41A6	Establecimiento de plantilla de acuerdo de claves para cálculo.	
Datos	0x91	*Clave pública efímera* Clave pública efímera $PK_{DH,IFD}$ (véase la sección 9.4.5) codificada como valor de clave pública sencilla.	EXIGIDO
	0x84	*Referencia de clave privada* El objeto de datos se EXIGE si la clave privada es ambigua, es decir, más de un par de claves disponibles para autenticación de microplaqueta (véanse las Secciones 6.2 y 9.2.6).	CONDICIONAL
Respuesta			
Datos	–	Ausente	
Bytes de estado	0x9000	*Procesamiento normal* La operación de acuerdo de claves se realizó con éxito. Se obtuvieron nuevas claves de sesión.	
	0x6A80	*Parámetros incorrectos en el campo de datos de comandos* La validación de la clave pública efímera falló.	
	Otros	*Error dependiente del sistema de operación* Las claves de sesión establecidas previamente siguen siendo válidas.	

6.2.4.2 Implantación utilizando MSE:Set AT y GENERAL AUTHENTICATE

1. MSE:Set AT: El comando MSE:Set AT se utiliza para seleccionar e inicializar el protocolo. El uso del MSE:Set AT para la autenticación de microplaqueta se indica por medio de un identificador de objeto de autenticación de microplaqueta (véanse las secciones 6.2.3 y 9.2.7) incluido como referencia de mecanismo criptográfico con el rótulo 0x80; véase la tabla a continuación.

Comando			
CLA		Específico de contexto	
INS	0x22	Gestionar entorno de seguridad	
P1/P2	0x41A4	*Autenticación de microplaqueta*: Establecimiento de plantilla de autenticación para autenticación interna.	
Datos	0x80	*Referencia de mecanismo criptográfico* Identificador de objeto de protocolo a seleccionar (valor solamente, se omite rótulo 0x06).	EXIGIDO
	0x84	*Referencia a la clave privada* El objeto de datos se EXIGE para indicar el identificador de la clave privada que ha de utilizarse si la clave privada es ambigua, es decir, más de una clave privada disponible para autenticación de microplaqueta.	CONDICIONAL
Respuesta			
Datos	–	Ausente	
Bytes de estado	0x9000	*Procesamiento normal* El protocolo se ha seleccionado e inicializado.	
	0x6A80	*Parámetros incorrectos en el campo de datos de comandos* El algoritmo no se apoya o la inicialización falló.	
	0x6A88	*Datos de referencia no hallados* Los datos de referencia (es decir, clave privada) no están disponibles.	
	otros	*Error dependiente del sistema de operación* La inicialización del protocolo falló.	

Nota.— Algunos sistemas de operación aceptan la selección de una clave no disponible y devuelven un error sólo cuando la clave se utiliza para el fin seleccionado.

2. GENERAL AUTHENTICATE: El comando GENERAL AUTHENTICATE se utiliza para ejecutar la autenticación de microplaqueta.

Comando				
CLA		Específico de contexto		
INS	0x86	GENERAL AUTHENTICATE		
P1/P2	0x0000	Claves y protocolo implícitamente conocidos.		
Datos	0x7C	*Datos de autenticación dinámica* Objetos de datos específicos del protocolo.		EXIGIDO
		0x80	Clave pública efímera	
Respuesta				
Datos	0x7C	*Datos de autenticación dinámica* Objetos de datos específicos del protocolo		EXIGIDO
Bytes de estado	0x9000	*Procesamiento normal* El protocolo (etapa) tuvo éxito.		
	0x6300	*La autenticación falló* El protocolo (etapa) falló.		
	0x6A80	*Parámetros incorrectos en el campo de datos* Los datos proporcionados son inválidos.		
	0x6A88	*Datos de referencia no hallados* Los datos de referencia (es decir, clave privada) no están disponibles.		
	Otros	*Error dependiente del sistema de operación* El protocolo (etapa) falló.		

Nota.— Las claves públicas para autenticación de microplaqueta apoyadas por la microplaqueta están disponibles en el objeto de seguridad (véase la sección 9.2.11). Si se apoya más de una clave pública, el terminal DEBE seleccionar la correspondiente clave privada de la microplaqueta que ha de utilizarse dentro de MSE:Set AT.

6.2.4.3 Clave pública efímera

Las claves públicas efímeras (véase la sección 9.4.5) se CODIFICARÁN como punto de curva elíptica (ECDH) o entero sin signo (DH).

7. MECANISMOS DE CONTROL DE ACCESO ADICIONALES

Los datos personales almacenados en el CI sin contacto definidos como mínimo obligatorio para interfuncionamiento mundial son la ZLM y la imagen almacenada en forma digital del rostro del titular. Ambos elementos también pueden verse (leerse) visualmente después de abierto el eMRTD y presentado para inspección.

Además de la imagen facial almacenada en forma digital como principal característica biométrica para interfuncionamiento mundial, la OACI ha aprobado el uso de imágenes de dedos o iris almacenadas en forma digital además del rostro. Para uso nacional o bilateral, los Estados PUEDEN optar por almacenar plantillas o pueden optar por limitar el acceso a estos datos o cifrarlos, según decidan los propios Estados.

El acceso a estos datos personales más sensibles DEBERÍA ser más restringido. Esto puede lograrse de dos maneras: mediante la ampliación del control de acceso o el cifrado de los datos. En la sección 7.1 se define la autenticación del terminal como un mecanismo interoperable para el control de acceso ampliado. Si no se exige la interoperabilidad, pueden usarse otros mecanismos.

7.1 Autenticación del terminal

El mecanismo de autenticación del terminal es CONDICIONAL. Para las aplicaciones LDS2 se EXIGE la implementación. La autenticación del terminal PUEDE usarse para proteger las características biométricas secundarias en la aplicación eMRTD.

El protocolo de autenticación del terminal es un protocolo de dos pasos, interrogación-respuesta, que proporciona una autenticación unilateral explícita del terminal. Se basa en el control de acceso ampliado, según se especifica en [TR-03110]. Si el IC admite este protocolo, este DEBE admitir la autenticación de microplaqueta o PACE con correspondencia de autenticación de microplaqueta.

Este protocolo permite que el IC verifique si el terminal está habilitado para acceder a datos sensibles. Como el terminal puede acceder luego a datos sensibles, toda comunicación ulterior DEBE protegerse adecuadamente. Así pues, la autenticación del terminal también autentica una clave pública efímera, elegida por el terminal, que se usa para la mensajería segura con autenticación de microplaqueta o PACE con correspondencia de autenticación de microplaqueta. El IC DEBE vincular los derechos de acceso del terminal a la mensajería segura establecida por la clave pública efímera autenticada del terminal.

El IC PUEDE apoyar la autenticación del terminal en el fichero maestro y/o la aplicación eMRTD. Si la autenticación del terminal se usa para proteger los grupos de datos de otras aplicaciones que no sean la aplicación eMRTD, el IC DEBE admitir la autenticación del terminal en el fichero maestro.

Nota.— Si se exige la compatibilidad con el control de acceso ampliado de la Unión Europea [TR-03110], el IC DEBE admitir la autenticación del terminal en la aplicación eMRTD.

7.1.2 Especificación del protocolo

Las siguientes etapas son ejecutadas por el terminal y el IC:

1. El terminal envía una cadena de certificado al IC. La cadena empieza con un certificado verificable cuya clave pública de la autoridad de certificación de verificación de país (CVCA) está almacenada en la microplaqueta y termina con el certificado de terminal.

2. El IC verifica los certificados y extrae la clave pública PK_{IFD} del terminal.

3. El IC elige al azar un interrogante r_{IC} y lo envía al terminal.

4. El terminal responde con la firma s_{IFD} = **Sign**(SK_{IFD}, $ID_{IC}||r_{IC}||$**Comp**($PK_{DH,IFD}$)).

5. El IC comprueba que **Verify**(PK_{IFD}, s_{IFD}, ID_{IC} $||r_{IC}||$**Comp**($PK_{DH,IFD}$)) = verdadero.

Nota.— La clave $PK_{DH,IFD}$ se genera durante la autenticación de microplaqueta o PACE con correspondencia de autenticación de microplaqueta. Si se genera más de una clave (p. ej., la autenticación de microplaqueta se realiza después del PACE con correspondencia de autenticación de microplaqueta), DEBE usarse la clave más reciente.

En este protocolo, ID_{IC} es un identificador del IC:

- Si se usa BAC, ID_{IC} es el número de documento del eMRTD contenido en la ZLM, incluido el dígito de verificación.

- Si se usa PACE, ID_{IC} se calcula usando la clave pública efímera PACE, i.e. ID_{IC} = **Comp**($PK_{DH,IC}$).

Nota.— Se EXIGE la ejecución correcta del protocolo PACE antes de que pueda realizarse la autenticación del terminal en el fichero maestro.

A continuación se muestra una versión simplificada:

IC (microplaqueta)		**IFD (sistema de inspección)**
Elegir r_{IC} al azar		
	— r_{IC} →	
		s_{IFD} = **Sign**(SK_{IFD},ID_{IC}||r_{IC}||**Comp**($PK_{DH,IFD}$))
	← s_{IFD} —-	
Verify(PK_{IFD} , s_{IFD}, ID_{IC}||r_{IC}||**Comp**($PK_{DH,IFD}$)) = verdadero		

Figura 4. Autenticación del terminal

7.1.3 Estado de seguridad

Si la autenticación del terminal se llevó a cabo correctamente, el IC OTORGARÁ acceso a datos sensibles almacenados con arreglo a la autorización efectiva del terminal autenticado. Si la autorización efectiva no otorga derechos de acceso a ningún dato de una aplicación LDS2, la selección de esta aplicación DEBE ser rechazada por el IC.

No obstante, el IC RESTRINGIRÁ los derechos de acceso del terminal a la mensajería segura establecida por la clave pública efímera autenticada, i. e. la clave pública efímera proporcionada por el terminal como parte de la autenticación de microplaqueta o PACE con correspondencia de autenticación de microplaqueta. El IC NO DEBE aceptar que se realice más de una ejecución de autenticación del terminal en la misma sesión (cf. Véase la definición de “sesión” en la sección 9.8.1 y la sección 9.8.3).

Nota 1.— Los derechos de acceso son válidos siempre que esté activa la mensajería segura establecida por las claves públicas efímeras autenticadas, así el estado de seguridad no se ve afectado por la selección de aplicaciones o su eliminación de la selección.

Nota 2.— La mensajería segura no se ve afectada por la autenticación del terminal. La microplaqueta del eMRTD RETENDRÁ los mensajes seguros aun cuando la autenticación del terminal falle (a menos que se produzca un error de la mensajería segura).

7.1.4 Especificaciones criptográficas

7.1.4.1 Autenticación del terminal con RSA

Para la autenticación del terminal con RSA, DEBEN usarse los siguientes algoritmos y formatos.

7.1.4.1.1 Algoritmo de firma

Se USARÁ RSA [RFC-3447], [PKCS#1] según se especifica en la tabla 7.

Tabla 7. Identificadores de objeto para la autenticación del terminal con RSA

OID	Firma	Condensación	Parámetros
id-TA-RSA-PSS-SHA-256	RSASSA-PSS	SHA-256	por defecto
id-TA-RSA-PSS-SHA-512	RSASSA-PSS	SHA-512	por defecto

Los parámetros por defecto que han de usarse con RSA-PSS se definen de la siguiente manera:

- Algoritmo de condensación: El algoritmo de condensación se selecciona con arreglo a la tabla 7.
- Algoritmo de generación de máscaras: MGF1 [RFC-3447], [PKCS#1] mediante el uso del algoritmo de condensación seleccionado.
- Longitud del salt: longitud de octetos del resultado del algoritmo de condensación seleccionado.
- Indicador de fin: 0xBC

7.1.4.1.2 Formato de clave pública

Se USARÁ el formato TLV [ISO/IEC 7816-8] descrito en el Doc 9303-12.

- El identificador de objeto se TOMARÁ de la tabla 7.
- La longitud de bits del módulo SERÁ de 2 048 o 3 072.
- La longitud de bits del exponente SERÁ como mucho de 32.

7.1.4.1.3 Clave pública comprimida

La clave pública efímera comprimida **Comp**($P_{KDH,IFD}$) del terminal se define como la condensación SHA-1 del valor público DH, i. e. una cadena de octetos de longitud fija 20.

7.1.4.2 Autenticación del terminal con el ECDSA

Para la autenticación del terminal con el ECDSA, DEBEN usarse los siguientes algoritmos y formatos.

7.1.4.2.1 Algoritmo de firma

El ECDSA se usará con el formato de firma de texto plano [TR-03111] especificado en la tabla 8.

Tabla8. Identificadores de objeto para la autenticación del terminal con el ECDSA

OID	**Firma**	**Condensación**
id-TA-ECDSA-SHA-224	ECDSA	SHA-224
id-TA-ECDSA-SHA-256	ECDSA	SHA-256
id-TA-ECDSA-SHA-384	ECDSA	SHA-384
id-TA-ECDSA-SHA-512	ECDSA	SHA-512

7.1.4.2.2 Formato de clave pública

SE USARÁ el formato TLV [ISO/IEC 7816-8] descrito en el Doc 9303-12.

- El identificador de objeto SE TOMARÁ de la tabla 8.
- La longitud de bits de la curva SERÁ de 224, 256, 320, 384 o 512.
- Los parámetros de dominio SERÁN conformes al [TR-03111].

7.1.4.2.3 Compresión de clave pública

La clave pública efímera comprimida **Comp**(PKDH,IFD) del terminal se define como la coordenada x del punto público ECDH, i. e. una cadena de octetos de longitud fija ⌈log256p⌉.

7.1.4.3 Validación de la certificación

Para validar un certificado de terminal, el IC DEBE proporcionarse con una cadena de certificado que empieza en un punto de confianza almacenado en el IC. Esos puntos de confianza son claves públicas más o menos recientes de la CVCA del IC.

7.1.4.3.1 Estado inicial del/de los punto(s) de confianza del IC

El/los punto(s) de confianza inicial(es) SE ALMACENARÁN en condiciones de seguridad en la memoria del IC en la producción de la fase de(pre)personalización.

El agente de (pre)personalización SE ENCARGARÁ DE:

- fijar la fecha corriente del IC en la fecha de la (pre)personalización; y
- personalizar la clave CVCA con la fecha efectiva más reciente como punto de confianza.

El agente de (pre)personalización PUEDE además personalizar la llave previa CVCA como punto de confianza.

7.1.4.3.2 Certificados de enlace

Como el par de claves utilizado por la CVCA cambia con el tiempo, tienen que producirse certificados de enlace CVCA. Los certificados de enlace CVCA DEBEN firmarse con la clave CVCA previa, i. e. la clave CVCA con la fecha efectiva más reciente. Se EXIGE que el IC actualice internamente su(s) punto(s) de confianza con arreglo a los certificados de enlace válidos recibidos.

El IC DEBE ser capaz de almacenar hasta dos puntos de confianza.

Nota.— Debido a la programación de los certificados de enlace CVCA (véase el Doc 9303-12), como mucho es necesario almacenar dos puntos de confianza en el IC.

7.1.4.3.3 Fecha actual

El IC DEBE aceptar certificados de enlace CVCA caducados, pero NO DEBE aceptar certificados DV ni de terminal caducados. Para determinar si un certificado está caducado, el IC USARÁ su fecha actual.

Fecha actual: Si el IC no tiene un reloj interno, la fecha actual del IC SE APROXIMARÁ de la siguiente manera. El IC aproxima la fecha de forma autónoma utilizando la fecha efectiva del certificado más reciente, contenida en un certificado de enlace a la CVCA válido, un certificado DV o un certificado de terminal preciso.

Certificado de terminal preciso: Un certificado de terminal es preciso si el IC confía en la autoridad expedidora del verificador del documento (DV) para producir certificados de terminal con la fecha efectiva correcta del certificado. Los certificados de enlace CVCA, los certificados DV y los certificados de terminal expedidos por un DV nacional SERÁN CONSIDERADOS precisos por el IC. Otros certificados NO DEBEN considerarse exactos.

Un terminal PUEDE enviar certificados de enlace CVCA, certificados DV y certificados de terminal a un IC para actualiza la fecha actual y el punto de confianza almacenado en el IC aun cuando no tenga intención o no sea capaz de seguir con la autenticación del terminal.

Nota.— El IC solo verifica que un certificado sea aparentemente reciente (i. e. con respecto a la fecha actual aproximada), a menos que el IC contenga un reloj interno.

7.1.4.3.4 Procedimiento de validación general

El procedimiento de validación del certificado consiste en tres etapas:

1. **Verificación del certificado**: La firma DEBE ser válida y, a menos que el certificado sea un certificado de enlace CVCA, el certificado NO DEBE estar caducado. Si falla la verificación, se INTERRUMPIRÁ el procedimiento.

   Nota.— Solo puede darse el caso de que un certificado de enlace CVCA esté caducado si el IC tiene una fuente de tiempo que supere a la fecha actual aproximada descrita anteriormente.

2. **Actualización del estado interno**: La fecha actual DEBE estar *actualizada*, la clave pública y los atributos (incluidas las ampliaciones de certificados pertinentes) DEBEN ser importados, DEBEN *activarse* nuevos puntos de confianza y DEBEN *desactivarse* puntos de confianza caducados para la verificación de certificados DV.

3. **Limpieza**: La microplaqueta PROPORCIONARÁ como máximo dos puntos de confianza activados por aplicación. Si permanecen activados más de dos puntos de confianza para una aplicación después de la actualización del estado interno, *se DESACTIVARÁ* el punto de confianza con la fecha efectiva menos reciente.

La operación de *actualizar* la fecha actual y las operaciones de *activar* y *desactivar* un punto de confianza DEBEN implementarse como una operación atómica.

Activación de un punto de confianza: se AÑADIRÁ el nuevo punto de confianza a la lista de puntos de confianza.

Desactivación de un punto de confianza: Los puntos de confianza caducados NO DEBEN usarse para la verificación de los certificados DV. En el caso de los IC en los que la fecha actual pueda superar a la fecha de caducidad de un punto de confianza, p. ej., los IC que usan un reloj interno, los puntos de confianza caducados DEBEN seguir siendo utilizables para la verificación de los certificados de enlace CVCA. Los puntos de confianza desactivados PUEDEN borrarse después de importarse correctamente el certificado de enlace sucesivo.

7.1.4.3.5 Ejemplo de procedimiento de validación

El siguiente procedimiento de validación, proporcionado a modo de ejemplo, PUEDE usarse para validar una cadena de certificados. Para cada certificado recibido el IC ejecuta las etapas siguientes:

1. El IC verifica la firma del certificado. Si la firma es incorrecta, fracasa la verificación.

2. Si el certificado no es un certificado de enlace CVCA, su fecha de caducidad se compara con la fecha actual del IC. Si la fecha de caducidad precede a la fecha corriente, la verificación falla.

3. Se da por válido el certificado y se importan la clave pública y los atributos (incluidas las ampliaciones pertinentes del certificado) que contiene el certificado.

    - Para la CVCA, el DV y los certificados de terminal precisos: La fecha efectiva del certificado se compara con la fecha actual del IC. Si la fecha actual precede a la fecha efectiva, la fecha corriente se actualiza a la fecha efectiva.

    - Para los certificados de enlace CVCA: La nueva clave pública de la CVCA se añade a la lista de puntos de confianza almacenados en condiciones de seguridad en la memoria del IC. Y entonces se procede a activar el nuevo punto de confianza.

    - Para los certificados DV y certificados de terminal: La nueva clave pública del DV o del terminal se importa temporalmente para la ulterior verificación del certificado o la autenticación del terminal, respectivamente.

4. Los puntos de confianza caducados que están almacenados en condiciones de seguridad en la memoria del IC se desactivan para la verificación de los certificados DV y pueden suprimirse de la lista de puntos de confianza.

7.1.4.3.6 Autorización efectiva

Cada certificado CONTENDRÁ una plantilla de autorización de la persona titular del certificado (véase el Doc 9303-12) y PUEDE contener ampliaciones de autorización (véase el Doc 9303-12, sección 7.2.2.6).

- En la plantilla de autorización de la persona titular del certificado se identifica el tipo de terminal (esta especificación solo considera los sistemas de inspección, pero otras especificaciones pueden usar diferentes tipos de terminal).

- La plantilla de autorización de la persona titular del certificado y las ampliaciones de autorización determinan la *autorización relativa* de la persona titular del certificado asignada por la autoridad que expide el certificado.

Para determinar la *autorización efectiva* de la persona titular de un certificado, el IC DEBE calcular un operador booleano "Y" a nivel de bit de la autorización relativa contenida en el certificado de terminal, el certificado DV referenciado y el certificado de la CVCA referenciado.

El IC INTERPRETARÁ la autorización efectiva de la siguiente manera:

- La función efectiva es una CVCA:
  - Este certificado de enlace fue expedido por la CVCA nacional.
  - El IC DEBE actualizar su punto de confianza interno, i. e. la clave pública y la autorización efectiva.
  - El expedidor de certificado es una fuente fiable de tiempo y el IC DEBE actualizar su fecha actual usando la fecha efectiva del certificado.
  - El IC NO DEBE otorgar el acceso de la CVCA a los datos sensibles (i.e. DEBERÍA hacerse caso omiso de la autorización efectiva).
- La función efectiva es un DV:
  - El certificado fue expedido por la CVCA nacional para un DV autorizado.
  - El expedidor de certificado es una fuente fiable de tiempo y el IC DEBE actualizar su fecha actual usando la fecha efectiva del certificado.
  - El IC NO DEBE otorgar un acceso DV a datos sensibles (i. e. DEBERÍA hacerse caso omiso de la autorización efectiva).
- La función efectiva es un terminal:
  - El certificado fue expedido por un DV nacional o extranjero.
  - Si el certificado es un certificado de terminal preciso (cf. sección 7.1.4.3.3), el expedidor es una fuente fiable de tiempo y el IC DEBE actualizar su fecha actual usando la fecha efectiva del certificado.
  - El IC DEBE otorgar el acceso del terminal autenticado a los datos sensibles con arreglo a la autorización efectiva.

Nota.— La plantilla de autorización de la persona titular del certificado y las ampliaciones de la autorización pueden contener bits no asignados a un derecho de acceso (bits RFU). El IC no DEBE tener en cuenta esos bits durante la evaluación de los derechos de acceso.

7.1.4.3.7 Importación de la clave pública

Las claves públicas importadas por el procedimiento de validación de certificado se almacenan *permanente* o *temporalmente* en el IC.

El IC DEBERÍA rechazar la importación de una clave pública si ya conoce la referencia de la persona titular del certificado.

Importación permanente: El IC IMPORTARÁ de forma permanente las llaves públicas contenidas en los certificados de enlace CVCA, que DEBERÁN almacenarse en condiciones de seguridad en la memoria del IC. Una llave pública importada de forma permanente y sus metadatos DEBERÁN CUMPLIR las siguientes condiciones:

- Una vez caducada PUEDE sobrescribirse con una clave pública importada posteriormente de forma permanente.

- DEBE sobrescribirse con la clave pública importada posteriormente de forma permanente con la misma referencia de persona titular del certificado o DEBE rechazarse la importación.

- NO DEBE sobrescribirse con una clave pública importada temporalmente.

Activar y desactivar una clave pública importada de forma permanente DEBE ser una operación atómica.

Importación temporal: El IC IMPORTARÁ temporalmente las llaves públicas contenidas en los certificados DV y de terminal.

Una clave pública importada temporalmente y sus metadatos DEBERÁN CUMPLIR las siguientes condiciones:

- NO SERÁ seleccionable ni utilizable si el IC ha sido inhabilitado.

- DEBE seguir siendo utilizable hasta que se complete correctamente la subsiguiente operación criptográfica (i. e. PSO:Verify certificate o External authenticate).

- PUEDE sobrescribirse con una clave pública temporalmente importada ulteriormente.

Un terminal NO DEBE hacer uso de ninguna clave pública importada temporalmente, sino de la importada más recientemente.

Metadatos importados: Para cada clave pública importada de forma permanente o temporal, DEBEN almacenarse los siguientes datos adicionales contenidos en el certificado (véase el Doc 9303-12):

- referencia de la persona titular del certificado

- autorización de la persona titular del certificado (función efectiva y autorización efectiva)

- fecha efectiva del certificado

- fecha de caducidad del certificado

- ampliaciones del certificado (cuando proceda)

El cálculo de la función efectiva (CVCA, DV o terminal) y la autorización efectiva de la persona titular del certificado se describen en la sección 7.1.4.3.6.

Nota.— El formato de los datos almacenados depende del sistema operativo y queda fuera del alcance de esta especificación.

7.1.5 Unidades de datos del protocolo de aplicación

La siguiente secuencia de comandos SE USARÁ con la mensajería segura para implementar la autenticación del terminal:

- MSE:Set DST

- PSO:Verify Certificate

- MSE:Set AT
- Get Challenge
- External Authenticate

Los pasos 1 y 2 se repiten para cada certificado verificable mediante tarjeta (CV) que haya que verificar (certificados de enlace CVCA, certificado DV, certificado de terminal).

7.1.5.1 MSE:Set DST

El comando MSE:Set DST se usa para establecer la verificación de los certificados.

Comando			
CLA		Específico para un contexto	
INS	0x22	MANAGE SECURITY ENVIRONMENT	
P1/P2	0x81B6	Establecimiento de una plantilla de firma digital para verificación.	
Datos	0x83	*Referencia de una clave pública* ISO 8859-1, nombre codificado de la clave pública que ha de establecerse	EXIGIDO
Respuesta			
Datos	–	Ausente	
Bytes de estado	0x9000 0x6A88 otros	*Operación normal* Se ha seleccionado la clave para el propósito definido. *Datos de referencia no hallados* La selección falló al no estar disponible la clave pública. *Error dependiente del sistema de operación* La clave no se ha seleccionado.	

Nota.— Algunos sistemas de operación aceptan la selección de una clave no disponible y solo devuelven un error cuando la clave pública se usa para el fin seleccionado.

7.1.5.2 PSO:Verify Certificate

El comando PSO:Verify Certificate se usa para verificar e importar certificados.

Comando			
CLA		Específico para un contexto	
INS	0x2A	PERFORM SECURITY OPERATION	
P1/P2	0x00BE	Verificar certificado autodescriptivo.	
Datos	0x7F4E 0x5F37	*Cuerpo del certificado* *Cuerpo del certificado que ha de verificarse* *Firma* *Firma del certificado que ha de verificarse.*	EXIGIDO EXIGIDO

Respuesta		
Datos	–	Ausente
Bytes de estado	0x9000 otros	*Procesamiento normal* El certificado se ha validado correctamente y la clave pública se ha importado. *Error dependiente del sistema de operación* No se pudo importar la clave pública (p. ej., el certificado no fue aceptado).

7.1.5.3 MSE:Set AT

El uso de MSE:Set AT para la autenticación del terminal se indica por medio de P1/P2 fijado en 0x81A4 (véase la tabla que figura a continuación):

Comando			
CLA		Específico para un contexto	
INS	0x22	MANAGE SECURITY ENVIRONMENT	
P1/P2	0x81A4	Autenticación del terminal:	
Datos	0x83	*Referencia de una clave pública/ clave secreta* Este objeto de datos sirve para seleccionar la clave pública del terminal por su nombre codificado con arreglo a la norma ISO 8859-1.	EXIGIDO
Respuesta			
Datos	–	Ausente	
Bytes de estado	0x9000 0x6A80 0x6A88 otros	*Procesamiento normal* *El protocolo se ha seleccionado e inicializado.* *Parámetros incorrectos en el campo de datos del comando* El algoritmo no se admite o falló la inicialización *Datos de referencia no hallados* Los datos de referencia no están disponibles. *Error dependiente del sistema de operación.* La inicialización del protocolo falló.	

Nota.— Algunos sistemas de operación aceptan la selección de una clave pública no disponible y devuelven un error solo cuando la clave se utiliza para el fin seleccionado.

7.1.5.4 Get Challenge

Comando			
CLA		Específico para un contexto	
INS	0x84	GET CHALLENGE	
P1/P2	0x0000		
Datos	–	Ausente	
Le	0x08		EXIGIDO
Respuesta			
Datos	r_{IC}	8 bytes de aleatoriedad	
Bytes de estado	0x9000 otros	*Procesamiento normal* *Error dependiente del sistema de operación.*	

7.1.5.5 EXTERNAL AUTHENTICATE

Comando			
CLA		Específico para un contexto	
INS	0x82	EXTERNAL AUTHENTICATE	
P1/P2	0x0000	Claves y algoritmos conocidos implícitamente.	
Data		Firma generada por el terminal.	EXIGIDO
Respuesta			
Datos	–	Ausente	
Bytes de estado	0x9000 0x6300 0x6982 otros	*Procesamiento normal* La autenticación se realizó correctamente. El acceso a los grupos de datos se otorgará con arreglo a la autorización efectiva del certificado verificado correspondiente. *Aviso* La verificación de la firma falló. *Estado de seguridad no satisfecho* La autenticación falló debido a que el actual nivel de autenticación del terminal no permite utilizar la autenticación del terminal (p. ej., la autenticación del terminal ya se había realizado, etc.). *Error dependiente del sistema de operación.* La autenticación falló.	

7.2 Cifrado de características biométricas adicionales

La restricción del acceso a las características biométricas adicionales también PUEDE efectuarse mediante cifrado de las mismas. Para poder descifrar los datos cifrados, el sistema de inspección DEBE contar con una clave de descifrado. La definición del algoritmo de cifrado/descifrado y de las claves que han de utilizarse corresponden al Estado que las implemente y queda fuera del alcance de este documento.

La implantación de la protección de las características biométricas adicionales depende de las especificaciones internas del Estado o de especificaciones convenidas bilateralmente entre Estados que comparten esta información.

8. SISTEMA DE INSPECCIÓN

Para apoyar el funcionamiento requerido y las opciones definidas que pueden implantarse en los eMRTD que se ofrecerán, el sistema de inspección deberá satisfacer ciertas condiciones previas.

8.1 Control de acceso de base

Los sistemas de inspección que apoyen el control de acceso de base DEBEN satisfacer las siguientes condiciones previas:

1. El sistema de inspección está equipado con medios para adquirir la ZLM del documento físico a efectos de obtener las claves de acceso de base al documento (K_{Enc} y K_{MAC}) del eMRTD.

2. El soporte lógico del sistema de inspección apoya el protocolo descrito en la sección 4.3, en caso de que se ofrezca al sistema un eMRTD con control de acceso de base, incluyendo el cifrado del canal de comunicación con construcción segura de mensajes.

8.2 Establecimiento de conexión autenticada por contraseña

Los sistemas de inspección PACE DEBEN satisfacer las siguientes condiciones previas:

1. El sistema de inspección está equipado con medios para adquirir la ZLM o el CAN del documento físico.

2. El soporte lógico del sistema de inspección apoya el protocolo descrito en la sección 4.4, en el caso de que se ofrezca al sistema un eMRTD con PACE, incluyendo el cifrado del canal de comunicación con construcción segura de mensajes.

8.3 Autenticación pasiva

Para poder realizar la autenticación pasiva de los datos almacenados en el CI sin contacto del eMRTD, el sistema de inspección debe tener el conocimiento de la información de claves de los Estados expedidores u organizaciones expedidoras:

1. Para cada Estado expedidor u organización expedidora, el certificado de CA de firma de país o la información pertinente extraída del certificado se ALMACENARÁ en condiciones de seguridad en el sistema de inspección.

2. Alternativamente, para cada Estado expedidor u organización expedidora, los certificados del firmante del documento (C_{DS}) o la información pertinente extraída de los certificados se ALMACENARÁ en condiciones de seguridad en el sistema de inspección.

Antes de utilizar una clave pública de CA de firma de país de un Estado expedidor u organización expedidora, el Estado receptor u organización receptora DEBE establecer confianza en esa clave.

Antes de utilizar un certificado del firmante del documento (C_{DS}) para verificación de un SO_D, el sistema de inspección VERIFICARÁ su firma digital, utilizando la clave pública de CA de firma de país.

Además, los sistemas de inspección TENDRÁN acceso a la información de revocación verificada.

8.4 Autenticación activa

El apoyo de autenticación activa en los sistemas de inspección es OPCIONAL.

Si el sistema de inspección apoya la autenticación active, se EXIGE que el sistema de inspección tenga capacidad de leer la ZLM visual.

Si el sistema de inspección apoya la autenticación activa, el soporte lógico del sistema de inspección APOYARÁ el protocolo de autenticación activa que se describe en la sección 6.1.

8.5 Autenticación de microplaqueta

El apoyo de autenticación de microplaqueta en los sistemas de inspección es OPCIONAL.

Si el sistema de inspección apoya la autenticación de microplaqueta, SE EXIGE que el sistema de inspección tenga capacidad de leer la ZLM visual.

Si el sistema de inspección apoya la autenticación de microplaqueta, el soporte lógico del sistema de inspección APOYARÁ el protocolo de autenticación de microplaqueta que se describe en la sección 6.2.

8.6 Autenticación del terminal

La admisión de la autenticación del terminal por los sistemas de inspección es OPCIONAL.

Si el sistema de inspección admite la autenticación del terminal, SE EXIGE que el sistema de inspección tenga capacidad para almacenar la clave privada del sistema de inspección en condiciones de seguridad. El sistema de inspección DEBE tener acceso a su DV a intervalos regulares para renovar el certificado de terminal.

Si el sistema de inspección admite la autenticación del terminal, el programa informático del sistema de inspección ADMITIRÁ el protocolo de autenticación del terminal descrito en la sección 7.1.

8.7 Descifrado de las características biométricas adicionales

La implantación de la protección de las características biométricas opcionales depende de las especificaciones internas del Estado o de especificaciones convenidas bilateralmente entre Estados que comparten esta información.

9. ESPECIFICACIONES COMUNES

9.1 Estructuras ASN.1

Las estructuras de datos `SubjectPublicKeyInfo` y `AlgorithmIdentifier` se definen como sigue:

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

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

En [X9.42] y [TR-03111] figuran detalles sobre los `parameters`.

9.2 Información sobre los protocolos y las aplicaciones admitidos

La estructura de datos ASN.1 `SecurityInfos` SERÁ proporcionada por la microplaqueta del eMRTD para indicar protocolos de seguridad apoyados. La estructura de datos se especifica como sigue:

```
SecurityInfos  ::= SET OF SecurityInfo

  SecurityInfo ::= SEQUENCE {
  protocol  OBJECT IDENTIFIER,
  requiredData ANY DEFINED BY protocol,
  optionalData ANY DEFINED BY protocol OPTIONAL
}
```

Los elementos contenidos en una estructura de datos `SecurityInfo` tienen el significado siguiente:

- El identificador de objeto `protocol` identifica el protocolo apoyado.
- El `requiredData` de tipo abierto contiene datos obligatorios específicos del protocolo.
- El `optionalData` de tipo abierto contiene datos opcionales específicos del protocolo.

Informaciones de seguridad para PACE

Para indicar el apoyo a PACE las `SecurityInfos` pueden contener las entradas siguientes:

- Por lo menos una `PACEInfo` con un parámetro de dominio normalizado DEBE estar presente.
- Para cada conjunto apoyado de parámetros de dominio explícito DEBE estar presente una `PACEDomainParameterInfo`.

Informaciones de seguridad para autenticación activa

Si la microplaqueta del eMRTD utiliza un algoritmo de firma basado en ECDSA para autenticación activa, las `SecurityInfos` DEBEN contener la siguiente entrada `SecurityInfo`:

- `ActiveAuthenticationInfo`

Informaciones de seguridad para autenticación de microplaqueta

Para indicar el apoyo de la autenticación de microplaqueta las `SecurityInfos` pueden contener las entradas siguientes:

- Por lo menos una `ChipAuthenticationInfo` y la correspondiente `ChipAuthenticationPublicKeyInfo` utilizando parámetros de dominio explícitos DEBEN estar presentes.

Informaciones de seguridad para autenticación del terminal

Para indicar la admisión de la autenticación del terminal, las `SecurityInfos` pueden contener las entradas siguientes:

- Por lo menos una `TerminalAuthenticationInfo` estará presente.

Informaciones de seguridad para las aplicaciones presentes

En la sección 3.11.2 del Doc 9303-10 se recomienda la presencia de un fichero elemental transparente EF.DIR para indicar las aplicaciones admitidas. El fichero es obligatorio si está presente alguna aplicación LDS2. Puesto que el EF.DIR no está firmado y, por tanto, puede ser manipulado, p. ej., para ocultar aplicaciones existentes del IFD, se proporciona una copia segura del EF.DIR como `SecurityInfo` si está presente alguna aplicación LDS2.

Informaciones de seguridad para otros protocolos

Las `SecurityInfos` PUEDEN contener entradas adicionales que indican la admisión de otros protocolos o proporcionan otra información. El sistema de inspección PUEDE descartar toda entrada que desconozca.

9.2.1 PACEInfo

Esta estructura de datos proporciona información detallada sobre una implantación de PACE.

- El identificador de objeto `protocol` IDENTIFICARÁ los algoritmos que han de utilizarse (es decir, acuerdo de claves, cifrado simétrico y MAC).
- El entero `versión` IDENTIFICARÁ la versión del protocolo. Sólo la versión 2 es apoyada por esta especificación.
- El entero `parameterId` se utiliza para indicar el identificador de parámetro de dominio. DEBE utilizarse si la microplaqueta del eMRTD emplea parámetros de dominio normalizados (véase la sección 9.5.1), proporciona múltiples parámetros de dominio explícito para PACE o si `protocol` es uno de los OID `*-CAM-*`. En caso de PACE con correspondencia de autenticación de microplaqueta, el `parameterID` también indica la identidad de la clave de autenticación de microplaqueta utilizada, es decir, la microplaqueta DEBE proporcionar una `ChipAuthentication PublicKeyInfo` con `keyID` igual a `parameterID` de esta estructura de datos.

```
PACEInfo ::= SEQUENCE {
    protocol    OBJECT IDENTIFIER(
                id-PACE-DH-GM-3DES-CBC-CBC |
                id-PACE-DH-GM-AES-CBC-CMAC-128 |
                id-PACE-DH-GM-AES-CBC-CMAC-192 |
                id-PACE-DH-GM-AES-CBC-CMAC-256 |
                id-PACE-ECDH-GM-3DES-CBC-CBC |
                id-PACE-ECDH-GM-AES-CBC-CMAC-128 |
                id-PACE-ECDH-GM-AES-CBC-CMAC-192 |
                id-PACE-ECDH-GM-AES-CBC-CMAC-256 |
                id-PACE-DH-IM-3DES-CBC-CBC |
                id-PACE-DH-IM-AES-CBC-CMAC-128 |
                id-PACE-DH-IM-AES-CBC-CMAC-192 |
                id-PACE-DH-IM-AES-CBC-CMAC-256 |
                id-PACE-ECDH-IM-3DES-CBC-CBC |
```

```
                id-PACE-ECDH-IM-AES-CBC-CMAC-128 |
                id-PACE-ECDH-IM-AES-CBC-CMAC-192 |
                id-PACE-ECDH-IM-AES-CBC-CMAC-256
                id-PACE-ECDH-CAM-AES-CBC-CMAC-128 |
                id-PACE-ECDH-CAM-AES-CBC-CMAC-192 |
                id-PACE-ECDH-CAM-AES-CBC-CMAC-256),
    version     INTEGER, -- MUST be 2
    parameterId INTEGER OPTIONAL
}
```

9.2.2 PACEDomainParameterInfo

Esta estructura de datos es EXIGIDA si la microplaqueta del eMRTD proporciona parámetros de dominio explícitos para PACE y DEBE omitirse en caso contrario.

- El identificador de objeto `protocol` IDENTIFICARÁ el tipo de parámetros de dominio (es decir, DH o ECDH).
- La secuencia `domainParameter` CONTENDRÁ los parámetros de dominio.
- El entero `parameterId` PUEDE utilizarse para indicar identificador de parámetro de dominio local. DEBE utilizarse si la microplaqueta del eMRTD proporciona múltiples parámetros de dominio explícitos para PACE.

```
PACEDomainParameterInfo ::= SEQUENCE {
  protocol          OBJECT IDENTIFIER(
                    id-PACE-DH-GM |
                    id-PACE-ECDH-GM |
                    id-PACE-DH-IM |
                    id-PACE-ECDH-IM
                    id-PACE-ECDH-CAM),
  domainParameter   AlgorithmIdentifier,
  parameterId       INTEGER OPTIONAL
}
```

Nota.— La microplaqueta del eMRTD PUEDE apoyar más de un conjunto de parámetros de dominio explícitos (es decir, la microplaqueta puede apoyar diferentes algoritmos o longitudes de clave). En este caso, el identificador DEBE revelarse en el correspondiente `PACEDomainParameterInfo`.

Los parámetros de dominio contenidos en `PACEDomainParameterInfo` no están protegidos y pueden no ser seguros. El uso de parámetros de dominio que no son seguros para PACE llevará que se filtre la contraseña utilizada. Las microplaquetas de eMRTD DEBEN apoyar por lo menos un conjunto de parámetros de dominio normalizados como se especifica en la sección 9.5.1. Los sistemas de inspección NO DEBEN utilizar parámetros de dominio explícitos proporcionados por la microplaqueta del eMRTD a menos que los sistemas de inspección sepan explícitamente que dichos parámetros de dominio son seguros.

Las claves públicas efímeras DEBEN intercambiarse como valores de clave pública llanos. En la sección 9.4.5 figura más información sobre la codificación.

9.2.3 Identificador de objeto PACE

Los identificadores de objeto utilizados para PACE figuran en el subárbol de `bsi-de`:

```
bsi-de OBJECT IDENTIFIER ::= {
    itu-t(0) identified-organization(4) etsi(0)
    reserved(127) etsi-identified-organization(0) 7
}
```

Se UTILIZARÁ el siguiente identificador de objeto:

```
id-PACE OBJECT IDENTIFIER ::= {
    bsi-de protocols(2) smartcard(2) 4
}

id-PACE-DH-GM                     OBJECT IDENTIFIER ::= {id-PACE 1}
id-PACE-DH-GM-3DES-CBC-CBC        OBJECT IDENTIFIER ::= {id-PACE-DH-GM 1}
id-PACE-DH-GM-AES-CBC-CMAC-128    OBJECT IDENTIFIER ::= {id-PACE-DH-GM 2}
id-PACE-DH-GM-AES-CBC-CMAC-192    OBJECT IDENTIFIER ::= {id-PACE-DH-GM 3}
id-PACE-DH-GM-AES-CBC-CMAC-256    OBJECT IDENTIFIER ::= {id-PACE-DH-GM 4}
id-PACE-ECDH-GM                   OBJECT IDENTIFIER ::= {id-PACE 2}
id-PACE-ECDH-GM-3DES-CBC-CBC      OBJECT IDENTIFIER ::= {id-PACE-ECDH-GM 1}
id-PACE-ECDH-GM-AES-CBC-CMAC-128  OBJECT IDENTIFIER ::= {id-PACE-ECDH-GM 2}
id-PACE-ECDH-GM-AES-CBC-CMAC-192  OBJECT IDENTIFIER ::= {id-PACE-ECDH-GM 3}
id-PACE-ECDH-GM-AES-CBC-CMAC-256  OBJECT IDENTIFIER ::= {id-PACE-ECDH-GM 4}

id-PACE-DH-IM                     OBJECT IDENTIFIER ::= {id-PACE 3}
id-PACE-DH-IM-3DES-CBC-CBC        OBJECT IDENTIFIER ::= {id-PACE-DH-IM 1}
id-PACE-DH-IM-AES-CBC-CMAC-128    OBJECT IDENTIFIER ::= {id-PACE-DH-IM 2}
id-PACE-DH-IM-AES-CBC-CMAC-192    OBJECT IDENTIFIER ::= {id-PACE-DH-IM 3}
id-PACE-DH-IM-AES-CBC-CMAC-256    OBJECT IDENTIFIER ::= {id-PACE-DH-IM 4}

id-PACE-ECDH-IM                   OBJECT IDENTIFIER ::= {id-PACE 4}
id-PACE-ECDH-IM-3DES-CBC-CBC      OBJECT IDENTIFIER ::= {id-PACE-ECDH-IM 1}
id-PACE-ECDH-IM-AES-CBC-CMAC-128  OBJECT IDENTIFIER ::= {id-PACE-ECDH-IM 2}
id-PACE-ECDH-IM-AES-CBC-CMAC-192  OBJECT IDENTIFIER ::= {id-PACE-ECDH-IM 3}
id-PACE-ECDH-IM-AES-CBC-CMAC-256  OBJECT IDENTIFIER ::= {id-PACE-ECDH-IM 4}

id-PACE-ECDH-CAM                  OBJECT IDENTIFIER ::= {id-PACE 6}
id-PACE-ECDH-CAM-AES-CBC-CMAC-128 OBJECT IDENTIFIER ::= {id-PACE-ECDH-CAM 2}
id-PACE-ECDH-CAM-AES-CBC-CMAC-192 OBJECT IDENTIFIER ::= {id-PACE-ECDH-CAM 3}
id-PACE-ECDH-CAM-AES-CBC-CMAC-256 OBJECT IDENTIFIER ::= {id-PACE-ECDH-CAM 4}
```

9.2.4 ActiveAuthenticationInfo

Si la microplaqueta del eMRTD utiliza el algoritmo de firma basado en ECDSA para autenticación activa, las `SecurityInfos` en el Grupo de datos 14 de la LDS de la microplaqueta del eMRTD DEBEN contener la siguiente entrada de `SecurityInfo`:

```
ActiveAuthenticationInfo ::= SEQUENCE {
    protocol OBJECT IDENTIFIER (id-icao-mrtd-security-aaProtocolObject
    version INTEGER -- MUST be 1
    signatureAlgorithm OBJECT IDENTIFIER
}

id-icao-mrtd-security-aaProtocolObject OBJECT IDENTIFIER ::=
                                    { id-icao-mrtd-security 5 }
```

Para `signatureAlgorithm`, se UTILIZARÁN los identificadores de objeto definidos en [TR-03111].

Nota.— El identificador de objeto `id-icao-mrtd-security` se define en el Doc 9303-10.

9.2.5 ChipAuthenticationInfo

Esta estructura de datos proporciona información detallada sobre una implantación de autenticación de microplaqueta.

- El identificador de objeto `protocol` IDENTIFICARÁ los algoritmos que han de utilizarse (es decir acuerdo de claves, cifrado simétrico y MAC).
- El entero `version` IDENTIFICARÁ la versión del protocolo. Actualmente, esta especificación solo apoya la versión 1.
- El entero `keyId` PUEDE utilizarse para indicar el identificador de clave local. DEBE utilizarse si la microplaqueta del eMRTD proporciona múltiples claves públicas para autenticación de microplaqueta.

```
ChipAuthenticationInfo ::= SEQUENCE {
  protocol   OBJECT IDENTIFIER(
             id-CA-DH-3DES-CBC-CBC |
             id-CA-DH-AES-CBC-CMAC-128 |
             id-CA-DH-AES-CBC-CMAC-192 |
             id-CA-DH-AES-CBC-CMAC-256 |
             id-CA-ECDH-3DES-CBC-CBC |
             id-CA-ECDH-AES-CBC-CMAC-128 |
             id-CA-ECDH-AES-CBC-CMAC-192 |
             id-CA-ECDH-AES-CBC-CMAC-256),
  version    INTEGER, -- MUST be 1
  keyId      INTEGER OPTIONAL
}
```

9.2.6 ChipAuthenticationPublicKeyInfo

Esta estructura de datos proporciona una clave pública para la autenticación de microplaqueta o PACE con correspondencia de autenticación de microplaqueta de la microplaqueta del eMRTD.

- El identificador de objeto `protocol` IDENTIFICARÁ el tipo de clave pública (es decir, DH o ECDH).
- La secuencia `chipAuthenticationPublicKey` CONTENDRÁ la clave pública en forma codificada.
- El entero `keyId` PUEDE utilizarse para indicar el identificador de clave local. DEBE utilizarse si la microplaqueta del eMRTD proporciona múltiples claves públicas para autenticación de microplaqueta o si esta clave se utiliza para PACE con correspondencia de autenticación de microplaqueta.

```
ChipAuthenticationPublicKeyInfo ::= SEQUENCE {
    protocol                    OBJECT IDENTIFIER(id-PK-DH | id-PK-ECDH),
    chipAuthenticationPublicKey SubjectPublicKeyInfo,
    keyId                       INTEGER OPTIONAL
}
```

Nota.— La microplaqueta del eMRTD PUEDE apoyar más de un par de claves de autenticación de microplaqueta (es decir, la microplaqueta puede apoyar diferentes algoritmos o longitudes de claves). En este caso, el identificador de clave local DEBE revelarse en la correspondiente `ChipAuthenticationInfo` y `ChipAuthenticationPublicKeyInfo`.

9.2.7 Identificador de objeto de autenticación de microplaqueta

Se UTIIZARÁ el siguiente identificador de objeto:

```
id-PK OBJECT IDENTIFIER ::= {
  bsi-de protocols(2) smartcard(2) 1
}

id-PK-DH                    OBJECT IDENTIFIER ::= {id-PK 1}
id-PK-ECDH                  OBJECT IDENTIFIER ::= {id-PK 2}

id-CA OBJECT IDENTIFIER ::= {
  bsi-de protocols(2) smartcard(2) 3
}

id-CA-DH                    OBJECT IDENTIFIER ::= {id-CA 1}
id-CA-DH-3DES-CBC-CBC       OBJECT IDENTIFIER ::= {id-CA-DH 1}
id-CA-DH-AES-CBC-CMAC-128   OBJECT IDENTIFIER ::= {id-CA-DH 2}
id-CA-DH-AES-CBC-CMAC-192   OBJECT IDENTIFIER ::= {id-CA-DH 3}
id-CA-DH-AES-CBC-CMAC-256   OBJECT IDENTIFIER ::= {id-CA-DH 4}

id-CA-ECDH                  OBJECT IDENTIFIER ::= {id-CA 2}
id-CA-ECDH-3DES-CBC-CBC     OBJECT IDENTIFIER ::= {id-CA-ECDH 1}
id-CA-ECDH-AES-CBC-CMAC-128 OBJECT IDENTIFIER ::= {id-CA-ECDH 2}
id-CA-ECDH-AES-CBC-CMAC-192 OBJECT IDENTIFIER ::= {id-CA-ECDH 3}
id-CA-ECDH-AES-CBC-CMAC-256 OBJECT IDENTIFIER ::= {id-CA-ECDH 4}
```

9.2.8 TerminalAuthenticationInfo

Esta estructura de datos proporciona información detallada sobre la implementación de la autenticación del terminal.

- El identificador de objeto `protocol` IDENTIFICARÁ el protocolo *general* de autenticación del terminal ya que el protocolo específico puede cambiar con el tiempo.
- El número entero de `version` IDENTIFICARÁ la versión del protocolo. Actualmente, esta especificación solo admite la versión 1. Nótese que versiones posteriores de [TR-03110] definen la version 2 de este protocolo, que queda fuera del alcance de esta especificación.

```
TerminalAuthenticationInfo ::= SEQUENCE {
   protocol OBJECT IDENTIFIER(id-TA),
   version  INTEGER        -- MUST be 1
 }
```

9.2.9 Identificadores de objeto de autenticación del terminal

SE UTILIZARÁ el siguiente identificador de objeto:

```
id-TA OBJECT IDENTIFIER ::= {
  bsi-de protocols(2) smartcard(2) 2
}

id-TA-RSA               OBJECT IDENTIFIER ::= {id-TA 1}
id-TA-RSA-PSS-SHA-256   OBJECT IDENTIFIER ::= {id-TA-RSA 4}
id-TA-RSA-PSS-SHA-512   OBJECT IDENTIFIER ::= {id-TA-RSA 6}

id-TA-ECDSA             OBJECT IDENTIFIER ::= {id-TA 2}
id-TA-ECDSA-SHA-224     OBJECT IDENTIFIER ::= {id-TA-ECDSA 2}
id-TA-ECDSA-SHA-256     OBJECT IDENTIFIER ::= {id-TA-ECDSA 3}
id-TA-ECDSA-SHA-384     OBJECT IDENTIFIER ::= {id-TA-ECDSA 4}
id-TA-ECDSA-SHA-512     OBJECT IDENTIFIER ::= {id-TA-ECDSA 5}
```

9.2.10 EFDIRInfo

Esta estructura de datos encierra una copia completa del contenido del fichero elemental transparente EF.DIR contenido en el fichero maestro.

```
  EFDIRInfo ::= SEQUENCE {
          protocol                          OBJECT IDENTIFIER(id-EFDIR),
          eFDIR                             OCTET STRING
  }

id-EFDIR OBJECT IDENTIFIER ::={
  id-icao-mrtd-security 13
}
```

9.2.11 Almacenamiento en la microplaqueta

Para indicar el apoyo a los protocolos y parámetros apoyados, la microplaqueta del eMRTD PROPORCIONARÁ `SecurityInfos` en ficheros elementales transparentes (la estructura genérica de estos ficheros figura en el el Doc 9303-10):

- El fichero EF.CardAccess contenido en el fichero maestro se EXIGE si la microplaqueta del eMRTD apoya PACE y CONTENDRÁ las `SecurityInfos` pertinentes que se requieren para PACE:
  - `PACEInfo`
  - `PACEDomainParameterInfo`
- El fichero EF.*CardSecurity* contenido en el fichero maestro se EXIGE si
  - la microplaqueta del eMRTD admite PACE con correspondencia de autenticación de microplaqueta, o
  - la microplaqueta del eMRTD admite la autenticación del terminal en el fichero maestro, o
  - el eMRTD admite la autenticación de la microplaqueta en el fichero maestro

  y CONTENDRÁ las siguientes `SecurityInfos`:
  - `ChipAuthenticationInfo` según requiere la autenticación de microplaqueta
  - `ChipAuthenticationPublicKeyInfo` según se requiere para PACE-CAM/autenticación de microplaqueta
  - `TerminalAuthenticationInfo` según requiere la autenticación del terminal
  - `EFDIRInfo` si hay más aplicaciones, además del eMRTD, presentes en la microplaqueta
  - Las `SecurityInfos` contenidas en EF.CardAccess.
- El fichero EF.DG14 contenido en la aplicación del eMRTD SE EXIGE si
- La microplaqueta del eMRTD admite PACE con correspondencia genérica/integrada
- La microplaqueta del eMRTD admite la autenticación del terminal en la aplicación eMRTD, o
- La microplaqueta del eMRTD admite la autenticación de la microplaqueta en la aplicación eMRTD

y CONTENDRÁ las siguientes `SecurityInfos`:

- `ChipAuthenticationInfo` según requiere la autenticación de microplaqueta
- `ChipAuthenticationPublicKeyInfo` según requiere la autenticación de microplaqueta
- `TermninalAuthenticationInfo` según requiere la autenticación del terminal
- Las `SecurityInfos` contenidas en EF.CardAccess.

- El conjunto completo de `SecurityInfos` (incluyendo las `SecurityInfos` contenidas en EF.CardAccess no especificadas en el Doc 9303) se ALMACENARÁ adicionalmente en EF.DG14 de la aplicación eMRTD (véase el Doc 9303-10).

Los ficheros PUEDEN contener `SecurityInfos` adicionales que caen fuera del alcance de esta especificación.

Nota.— Si bien la autenticidad de las `SecurityInfos` *almacenadas en EF.DG14 y EF.CardSecurity está protegida por la autenticación pasiva, el fichero EF.CardAccess no está protegido.*

9.3 APDU

9.3.1 Longitud ampliada

Dependiendo del tamaño de los objetos criptográficos (p. ej., claves públicas, firmas), las APDU con campos de longitud ampliada DEBEN utilizarse para enviar estos datos a la microplaqueta del eMRTD. En [ISO/IEC 7816-4] figuran detalles sobre la longitud ampliada.

9.3.1.1 Microplaquetas del eMRTD

Para las microplaquetas del eMRTD, el apoyo de la longitud ampliada es CONDICIONAL. Si los algoritmos criptográficos y los tamaños de clave seleccionados por el Estado expedidor requieren el uso de longitud ampliada, las microplaquetas del eMRTD APOYARÁN la longitud ampliada. Si la microplaqueta del eMRTD apoya la longitud ampliada, ello DEBE indicarse en el ATR/ATS o en EF.ATR/INFO según se especifica en [ISO/IEC 7816-4].

9.3.1.2 Terminales

Para los terminales se exige el apoyo de longitud ampliada. Un terminal DEBERÍA examinar si el apoyo de longitud ampliada está indicado o no en la ATR/ATS de la microplaqueta del eMRTD o en EF.ATR/INFO antes de utilizar esta opción. El terminal NO DEBE utilizar la longitud ampliada para APDU distintos de los comandos siguientes a menos que los tamaños exactos de memoria intermedia de entrada y salida de la microplaqueta del eMRTD se declaren explícitamente en la ATR/ATS o en EF.ATR/INFO.

- MSE:Set KAT
- GENERAL AUTHENTICATE

9.3.2 Encadenamiento de comandos

El encadenamiento de comandos DEBE utilizarse para que el comando GENERAL AUTHENTICATE enlace la secuencia de comandos con la ejecución del protocolo PACE. El encadenamiento de comandos NO DEBE utilizarse para otros fines a menos que se indique claramente en la microplaqueta. En la [ISO/IEC 7816-4] figuran detalles sobre el encadenamiento de comandos.

9.3.3 Objetos de datos

El emisor de un comando o respuesta APDU DEBE transmitir los objetos de datos del campo de datos en el orden definido en las descripciones de la APDU.

Nota.— No se exige aceptar los objetos de datos en cualquier orden, aunque así se aumenta la interoperabilidad de algunos comandos, p. ej., en el caso de MSE:Set AT/GENERAL AUTHENTICATE. No obstante, debe procederse con cautela en caso de comandos como PSO:Verify Certifcate, en que el orden se ha fijado por motivos criptográficos.

9.4 Objetos de datos de clave pública

Un objeto de datos de clave pública es una estructura BER TLV construida que contiene un identificador de objeto y varios objetos de datos específicos del contexto anidados dentro de la plantilla de clave pública del titular de la tarjeta 0x7F49.

- El identificador de objeto es específico de cada aplicación y se refiere no solamente al formato de clave pública (es decir, los objetos de datos específicos de contexto) sino también a su uso.
- Los objetos de datos específicos del contexto son definidos por el identificador de objeto y contienen el valor de clave pública y los parámetros de dominio.

El formato de los objetos de datos de clave pública utilizados en esta especificación se describe a continuación.

9.4.1 Codificación del objeto de datos

Un entero sin signo se CONVERTIRÁ en una cadena de octetos utilizando la representación binaria del entero en un formato big-endian. Se UTILIZARÁ el número mínimo de octetos, es decir, NO DEBEN utilizarse los octetos iniciales de valor 0x00.

Para codificar puntos de curva elíptica, se UTILIZARÁ la codificación no comprimida con arreglo a [TR-03111].

9.4.2 Claves públicas RSA

En la tabla 9 se muestran los objetos de datos contenidos en una clave pública RSA. El orden de los objetos de datos es fijo.

Tabla 9. Clave pública RSA

Objeto de datos	Notación	Rótulo	Tipo	Certificado CV
Identificador de objetos		0x06	Identificador de objetos	m
Módulo compuesto	*n*	0x81	Número entero sin signo	m
Exponente público	*e*	0x82	Número entero sin signo	m

9.4.3 Claves públicas Diffie Hellman

Los objetos de datos contenidos en una clave pública DH se muestran en la tabla 10. El orden de los objetos de datos es fijo.

Tabla 10. Objetos de datos para claves públicas DH

Objeto de datos	Notación	Rótulo	Tipo
Identificador de objeto		0x06	Identificador de objeto
Módulo primo	p	0x81	Número entero sin signo
Orden del subgrupo	q	0x82	Número entero sin signo
Generador	g	0x83	Número entero sin signo
Valor público	y	0x84	Número entero sin signo

Nota.— La codificación de los componentes de la clave como enteros no firmados implica que cada uno de ellos se codifica sobre el menor número de bytes posible, es decir, sin los bytes precedentes puestos a 0x00. En particular, la clave pública DH puede codificarse sobre un número de bytes menor que el número de bytes del primo.

9.4.4 Claves públicas de curva elíptica

Los objetos de datos contenidos en una clave pública EC se muestran en la tabla 11. El orden de los objetos de datos es fijo, los parámetros de dominio CONDICIONALES DEBEN estar todos presentes, excepto el cofactor, o todos ausentes como sigue:

Tabla 11. Objetos de datos para claves públicas ECDH

Objeto de datos	Notación	Rótulo	Tipo
Identificador de objeto		0x06	Identificador de objeto
Módulo primo	p	0x81	Número entero sin signo
Primer coeficiente	a	0x82	Número entero sin signo
Segundo coeficiente	b	0x83	Número entero sin signo
Punto de base	G	0x84	Punto de curva elíptica
Orden del punto de base	r	0x85	Número entero sin signo
Punto público	Y	0x86	Punto de curva elíptica
Cofactor	f	0x87	Número entero sin signo

9.4.5 Claves públicas efímeras

Para las claves públicas efímeras el formato y los parámetros de dominio ya se conocen. Por consiguiente, solo el valor de clave pública llano, es decir, el valor público y para las claves públicas Diffie-Hellman y el punto público Y para las claves públicas de curva elíptica, se utilizan para indicar la clave pública efímera en un objeto de datos específico de contexto.

Nota.— La validación de claves públicas efímeras es RECOMENDADA. Para DH, el algoritmo de validación requiere que la microplaqueta del eMRTD tenga un conocimiento más detallado de los parámetros de dominio (es decir, el orden del subgrupo utilizado) que es el que normalmente proporciona PKCS#3.

9.5 Parámetros de dominio

Con la excepción de los parámetros de dominio contenidos en `PACEInfo`, todos los parámetros de dominio se PROPORCIONARÁN como `AlgorithmIdentifier` (véase la sección 9.1).

Dentro del `PACEInfo`, se HARÁ referencia directamente a los ID de parámetros de dominio normalizados que se describen en la tabla 12. Los parámetros de dominio explícitos proporcionados por `PACEDomainParameterInfo` NO DEBEN utilizar estos ID reservados para los parámetros de dominio normalizados.

9.5.1 Parámetros de dominio normalizados

DEBERÍAN utilizarse los ID de parámetros de dominio normalizados descritos en la tabla siguiente. Los parámetros de dominio explícitos NO DEBEN utilizar dichos ID reservados para parámetros de dominio normalizados.

El siguiente identificador de objeto DEBERÍA utilizarse para hacer referencia a los parámetros de dominio normalizados en un `AlgorithmIdentifier` (véase la sección 9.1):

```
standardizedDomainParameters OBJECT IDENTIFIER ::= {
  bsi-de algorithms(1) 2
}
```

Dentro de un `AlgorithmIdentifier`, este identificador de objeto HARÁ referencia al ID del parámetro de dominio normalizado según figura en la Tabla 12 como `INTEGER`, contenido como `parameters` en el `AlgorithmIdentifier`.

Tabla 12. Parámetros de dominio normalizados

ID	*Nombre*	*Tamaño (bit)*	*Tipo*	*Referencia*
0	Grupo MODP de 1024-bits con subgrupo de orden primo de 160-bits	1024/160	GFP	[RFC 5114]
1	Grupo MODP de 2048-bits con subgrupo de orden primo de 224-bits	2048/224	GFP	[RFC 5114]
2	Grupo MODP de 2048-bits con subgrupo de orden	2048/256	GFP	[RFC 5114]

ID	*Nombre*	*Tamaño (bit)*	*Tipo*	*Referencia*
	primo de 256-bits			
3-7	RFU			
8	NIST P-192 (secp192r1)	192	ECP	[RFC 5114], [FIPS 186-4]
9	BrainpoolP192r1	192	ECP	[RFC 5639]
10	NIST P-224 (secp224r1) *	224	ECP	[RFC 5114], [FIPS 186-4]
11	BrainpoolP224r1	224	ECP	[RFC 5639]
12	NIST P-256 (secp256r1)	256	ECP	[RFC 5114], [FIPS 186-4]
13	BrainpoolP256r1	256	ECP	[RFC 5639]
14	BrainpoolP320r1	320	ECP	[RFC 5639]
15	NIST P-384 (secp384r1)	384	ECP	[RFC 5114], [FIPS 186-4]
16	BrainpoolP384r1	384	ECP	[RFC 5639]
17	BrainpoolP512r1	512	ECP	[RFC 5639]
18	NIST P-521 (secp521r1)	521	ECP	[RFC 5114], [FIPS 186-4]
19-31	RFU			

* Esta curva no puede utilizarse con la correspondencia integrada.

9.5.2 Parámetros de dominio explícitos

El Identificador de objeto `dhpublicnumber` o `ecPublicKey` para DH o ECDH, respectivamente, SE UTILIZARÁ para hacer referencia a los parámetros de dominio explícito en un `AlgorithmIdentifier` (véase la sección 9.1):

```
dhpublicnumber OBJECT IDENTIFIER ::= {
  iso(1) member-body(2) us(840) ansi-x942(10046) number-type(2) 1
}

ecPublicKey OBJECT IDENTIFIER ::= {
  iso(1) member-body(2) us(840) ansi-x962(10045) keyType(2) 1
}
```

En el caso de curvas elípticas, los parámetros de dominio DEBEN describirse explícitamente en la estructura `ECParameters`, contenida como `parameters` en el `AlgorithmIdentifier`, es decir, NO DEBEN utilizarse curvas nombradas y parámetros de dominio implícitos.

9.6 Algoritmos de acuerdo de clave

La especificación apoya el acuerdo de claves Diffie-Hellman y el Diffie-Hellman de curva elíptica según se resume en la tabla siguiente:

Tabla 7. Algoritmos de acuerdo de clave

Algoritmo/Formato	DH	ECDH
Algoritmo de acuerdo de clave	[PKCS#3]	ECKA [TR-03111]
Formato de clave pública X.509	[X9.42]	[TR-03111]
Formato de clave pública TLV	TLV, véase la sección 9.4.3	TLV, véase la sección 9.4.4
Validación de clave pública efímera	[RFC 2631]	[TR-03111]

9.7 Mecanismo de obtención de clave

9.7.1 Función de obtención de claves

La función de obtención de claves **KDF**(K,c), se define como sigue:

Entrada: Se requieren las entradas siguientes:

- El valor de secreto compartido K (EXIGIDO)
- Un contador c de enteros big-endian de 32-bits (EXIGIDO)

Salida: Cadena de octetos keydata.

Acciones: Se ejecutan las acciones siguientes:

- keydata = **H**(K || c)
- Salida de cadena de octetos keydata

La función de obtención de claves que el **KDF**(K,c) requiere una función de condensación adecuada indicada por **H**(), es decir, la longitud de bits de la función de condensación SERÁ mayor o igual que la longitud de bits de la clave obtenida. El valor de condensación se INTERPRETARÁ como salida de bytes big-endian.

Nota.— El secreto compartido K se define como cadena de octetos. Si el secreto compartido se genera con ECKA [TR-03111], se UTILIZARÁ la coordenada x del punto generado.

9.7.1.1 3DES

Para obtener claves 3DES [FIPS 46-3] de 128-bits (112-bits excluyendo bits de paridad) se EMPLEARÁ la función de condensación SHA-1 [FIPS 180-4] y DEBEN ejecutarse las siguientes etapas adicionales:

- Utilización de octetos 1 a 8 de keydata para formar keydataA y octetos 9 a 16 de keydata para formar keydataB; no se utilizan octetos adicionales.

- Ajustar los bits de paridad de keydataA y keydataB para formar claves DES correctas (OPCIONAL).

9.7.1.2 AES

Para obtener claves AES [FIPS 197] de 128-bits se UTILIZARÁ la función de condensación SHA-1 [FIPS 180-4] y se DEBE ejecutar la siguiente etapa adicional:

- Utilización de octetos 1 a 16 de keydata; no se utilizan octetos adicionales.

Para obtener claves AES [FIPS 197] de 192-bits y 256-bits se UTILIZARÁ la función SHA-256 [FIPS 180-4]. Para las claves AES de 192-bits DEBE ejecutarse la siguiente etapa adicional:

- Utilización de octetos 1 a 24 de keydata; no se utilizan octetos adicionales.

9.7.2 Claves de acceso de base al documento

El cálculo de dos claves 3DES a partir de una semilla de claves (K) se utiliza para establecer las claves de acceso de base al documento K_{Enc} = **KDF**(K,1) y K_{MAC} = **KDF**(K,2).

9.7.3 PACE

Sea $\mathbf{KDF}_{\pi}(\pi)$ = **KDF**(f(π),3) una función de obtención de claves para obtener claves de cifrado de una contraseña π. La codificación de las contraseñas, es decir, K= f(π) se especifica en la tabla 14:

Tabla 8. Codificación de contraseñas

Contraseña	**Codificación**
ZLM (MRZ)	SHA-1(número de documento || fecha de nacimiento || fecha de caducidad)
CAN	Cadena de caracteres codificada según [ISO/IEC 8859-1]

Nota.— El número del documento que ha de utilizarse como entrada siempre es el número completo del documento. En el caso de los documentos TD1 cuyo número de documento consta de más de nueve caracteres, el número del documento tiene que concatenarse a partir del campo del número del documento y el campo de datos opcionales de la ZLM, excluido el carácter de relleno. Véase también la nota j) en la sección 4.2.2 del Doc 9303-5.

9.7.4 Claves de construcción segura de mensajes

Las claves para el cifrado de autenticación se obtienen como $\mathbf{KDF}_{Enc}(K) = \mathbf{KDF}(K,1)$ y $\mathbf{KDF}_{MAC}(K) = \mathbf{KDF}(K,2)$ respectivamente, a partir de un secreto compartido K.

9.8 Construcción segura de mensajes

9.8.1 Iniciación de la sesión

La sesión se inicia cuando se establece la construcción segura de mensajes. Dentro de una sesión, las claves de construcción segura de mensajes (es decir, establecidas por el control de acceso de base, PACE o autenticación de microplaqueta) pueden modificarse.

La construcción segura de mensajes se basa en 3DES [FIPS 46-3] o AES [FIPS 197] en modo cifrado y luego autenticado, es decir, los datos se rellenan, cifran y posteriormente los datos cifrados y formateados se ingresan en el cálculo de autenticación. Estas claves de sesión se OBTENDRÁN utilizando la función de obtención de claves que se describe en la sección 9.7.1.

Nota.— El relleno siempre es ejecutado por la capa de construcción segura de mensajes, por consiguiente, el código de autenticación de mensajes subyacente no necesita ejecutar ningún relleno interno.

9.8.2 Contador de secuencia de envío

Se UTILIZARÁ un entero sin signo como contador de secuencia de envío (SSC). El tamaño de bits del SSC SERÁ igual al tamaño de bloque del cifrado de bloque utilizado para la construcción segura de mensajes, es decir, 64 bits para 3DES y 128 bits para AES.

El SSC se AUMENTARÁ cada vez antes de generarse un comando o respuesta APDU, es decir, si el valor inicial es x, en el primer comando el valor del SSC es x+1. El valor del SSC para primera respuesta es x+2.

Si se reinicia la construcción segura de mensajes, el SSC se utiliza como sigue:

- Los comandos utilizados para el acuerdo de claves están protegidas con las antiguas claves de sesión y el antiguo SSC. Esto se aplica en particular a la respuesta a la última orden utilizada para el acuerdo de clave de sesión.

- El contador de secuencia de envío se establece en su nuevo valor de inicio, véase la sección 9.8.6.3 para 3DES y la sección 9.8.7.3 para AES.

- Las nuevas claves de sesión y el nuevo SSC se utilizan para proteger los comandos y respuestas subsiguientes.

9.8.3 Terminación de la sesión

La microplaqueta del eMRTD DEBE cancelar la construcción segura de mensajes únicamente si ocurre un error de construcción segura de mensajes o si se recibe una APDU llana.

Si la construcción segura de mensajes se cancela, la microplaqueta del eMRTD ELIMINARÁ las claves de sesión almacenadas y repondrá los derechos de acceso del terminal.

Nota.— La microplaqueta eMRTD PUEDE seleccionar implícitamente el fichero maestro cuando se termine una sesión.

9.8.4 Estructura de mensajes de las APDU SM

Los objetos de datos SM [véase la (ISO/IEC 7816-4)] DEBEN utilizarse en el orden siguiente:

- Comando APDU: [DO‘85’ o DO‘87’] [DO‘97’] DO‘8E’.
- Respuesta APDU: [DO‘85’ o DO‘87’] [DO‘99’] DO‘8E’.

En caso de que INS sea par, SE USARÁ el DO‘87’ y, en caso de que INS sea impar, SE USARÁ el DO’85’.

Todos los objetos de datos SM DEBEN codificarse en BER TLV según se especifica en [ISO/IEC 7816-4]. El encabezamiento de comando DEBE incluirse en el cálculo de MAC, por consiguiente, DEBE utilizarse el byte de clase CLA = 0x0C.

El valor real de Lc se modificará a Lc’ después de la aplicación de la construcción segura de mensajes. Si se requiere, puede incluirse opcionalmente un objeto de datos apropiado en la parte de datos de la APDU para indicar el valor original de Lc.

En la figura 5 se muestra la transformación de un comando APDU no protegido en comando APDU protegido en el caso de que se disponga de *Data* y *Le*. Si no se dispone de *Data*, se dejará fuera la construcción de DO ‘87’. Si no se dispone de *Le,* se dejará fuera la construcción de DO ‘97’. Para evitar ambigüedades se RECOMIENDA no utilizar un campo de valor vacío del objeto de datos Le (véase también la sección 10.4 de [ISO/IEC 7816-4]).

En la figura 6 se muestra la transformación de un comando APDU no protegido en respuesta APDU protegida en caso de que se disponga de *Data*. Si no se dispone de *Data*, se dejará fuera la construcción de DO ‘87’.

9.8.5 Errores SM

La cancelación del canal protegido para la aplicación eMRTD ocurre cuando:

- el CI sin contacto no está alimentado; o
- el CI sin contacto reconoce un error SM mientras interpreta un comando. En este caso, los bytes de estado deben devolverse sin SM.

Si la construcción segura de mensajes se cancela, la microplaqueta del eMRTD ELIMINARÁ las claves de sesión almacenadas y repondrá los derechos de acceso al terminal.

Nota.— PUEDE haber otras circunstancias en las cuales el CI sin contacto cancela la sesión. No es posible proporcionar una lista completa de tales circunstancias.

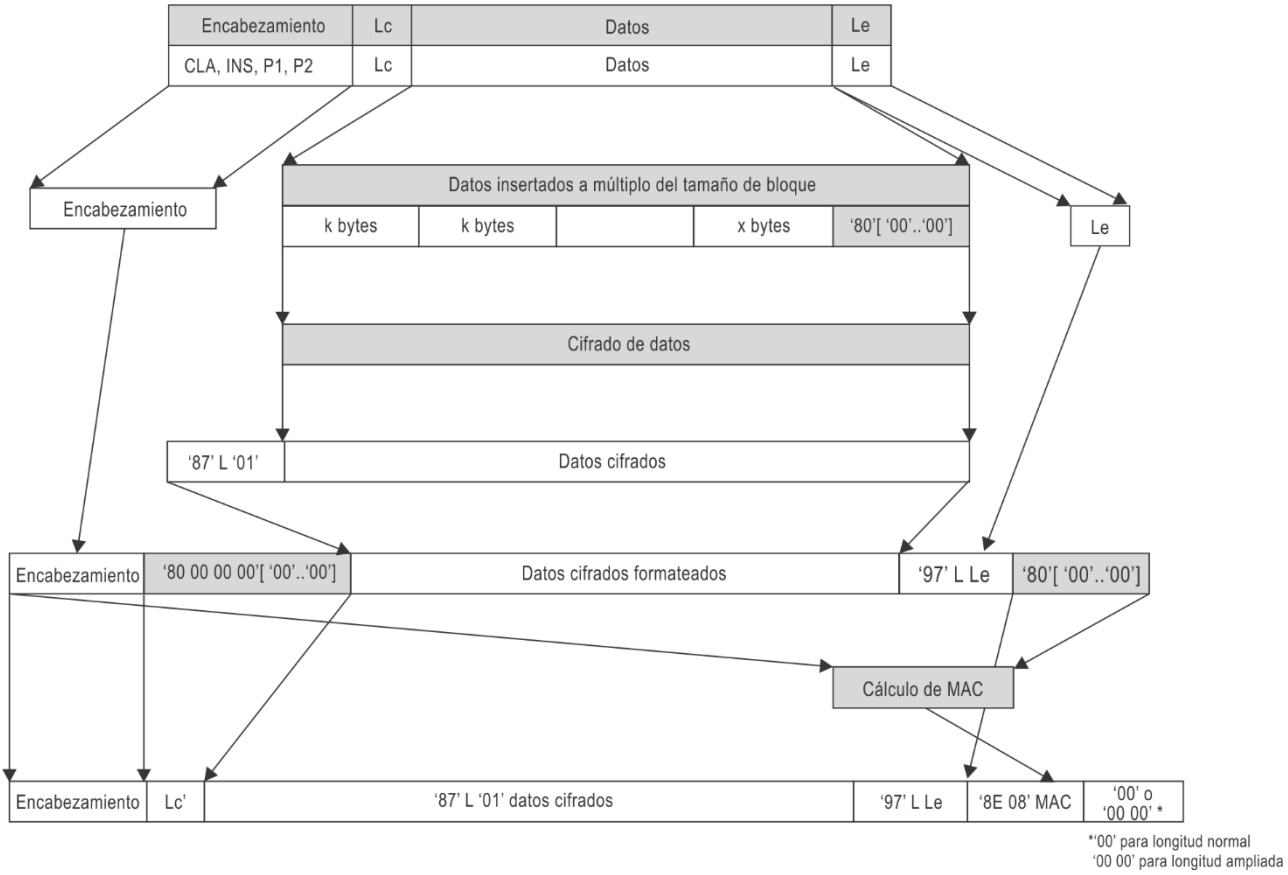

Figura 5. Cálculo de comando APDU SM para byte INS par

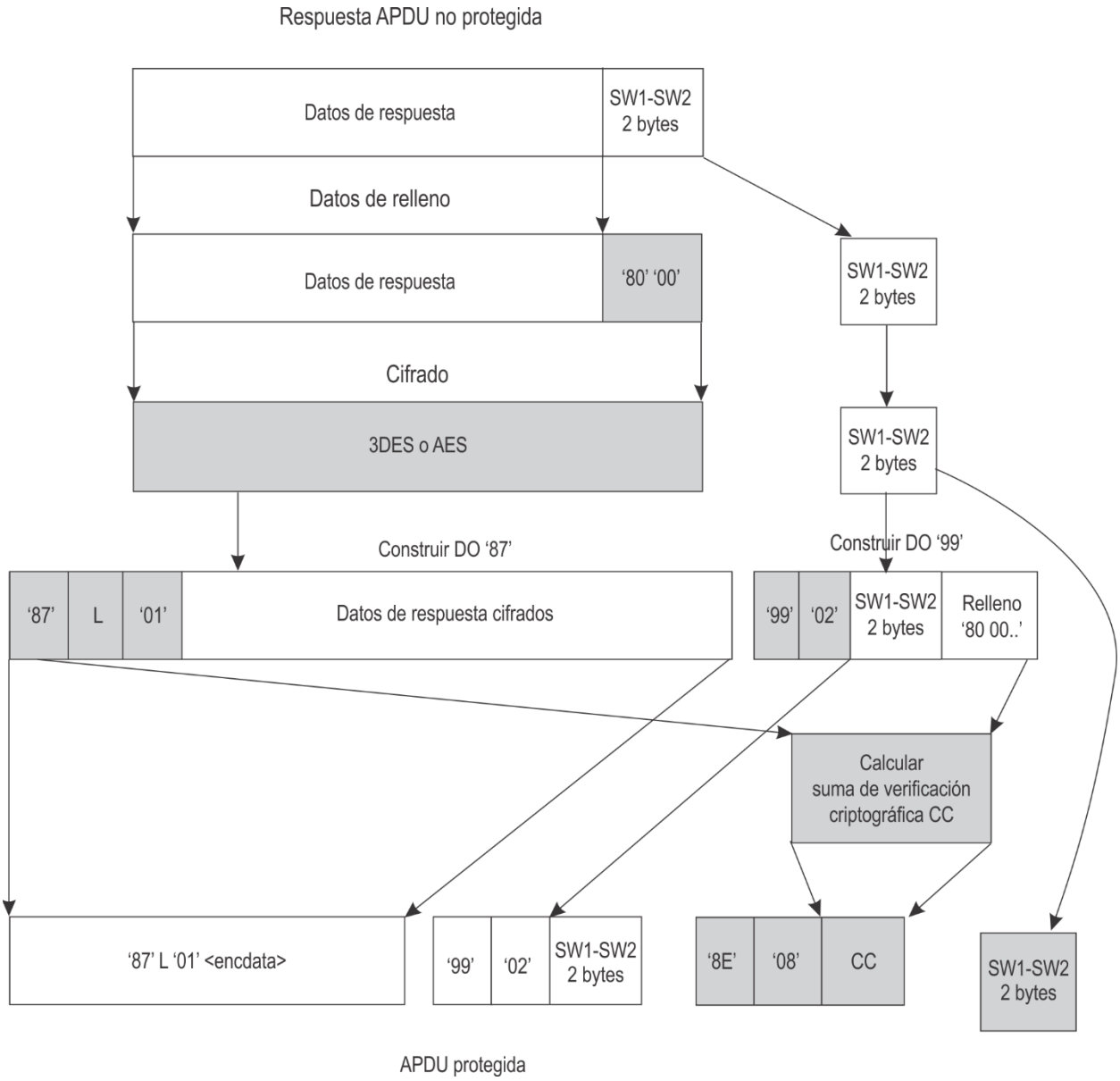

Figura 6. Cálculo de respuesta APDU SM para byte INS par

9.8.6 Modos de operación 3DES

9.8.6.1 Cifrado

Se utiliza 3DES de dos claves en modo CBC con IV cero (es decir, 0x00 00 00 00 00 00 00 00) con arreglo a [ISO/IEC 11568-2]. Se utiliza relleno con arreglo al método de relleno 2 de [ISO/IEC 9797-1].

9.8.6.2 Autenticación de mensaje

Se calculan las sumas de verificación criptográfica utilizando el algoritmo 3 MAC de [ISO/IEC 9797-1] con cifrado de bloques DES, y de cero IV (8 bytes), y método de relleno 2 de [ISO/IEC 9797-1]. La longitud MAC DEBE ser de 8 bytes.

Después de una autenticación exitosa el datagrama que ha de transformarse en MACed DEBE ser antepuesto por el contador de secuencia de envío.

9.8.6.3 Contador de secuencia de envío

Para la construcción segura de mensajes después de BAC, el contador de secuencia de envío se INICIALIZARÁ concatenando los cuatro bytes menos significativos de RND.IC y RND.IFD, respectivamente:

SSC = RND.IC (4 bytes menos significativos) || RND.IFD (4 bytes menos significativos).

En todos los demás casos, el SSC se INICIALIZARÁ a cero (es decir 0x00 00 00 00 00 00 00 00).

9.8.7 Modos de operación AES

9.8.7.1 Cifrado

Para el cifrado de mensajes se UTILIZARÁ AES [FIPS 197] en modo CBC con arreglo a [ISO/IEC 10116] con claves KS_{Enc} y IV = **E**(KS_{Enc}, SSC).

9.8.7.2 Autenticación de mensajes

Para la autenticación de mensajes se UTILIZARÁ AES en modos CMAC [SP 800-38B] con KS_{MAC} con una longitud MAC de 8 bytes. El datagrama que ha de autenticarse SERÁ antepuesto por el contador de secuencia de envío.

9.8.7.3 Contador de secuencia de envío

El contador de secuencia de envío se INICIALIZARÁ a cero (es decir, 0x00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00).

10. REFERENCIAS (NORMATIVA)

[X9.42] ANSI: X9.42, Public Key Cryptography for the Financial Services Industry: Agreement of Symmetric Keys Using Discrete Logarithm Cryptography, 1999

[ISO/IEC 7816-4] ISO/IEC 7816-4:2013 Identification cards — Integrated circuit cards — Part 4: Organization, security and commands for interchange

[ISO/IEC 7816-8] ISO/IEC 7816-8:2019 Identification cards — Integrated circuit cards — Part 8: Commands and mechanisms for security operations

[ISO/IEC 8859-1] ISO/IEC 8859-1:1998 Information technology — 8-bit single-byte coded graphic character sets — Part 1: Latin alphabet No. 1

[ISO/IEC 9796-2] ISO/IEC 9796-2:2010 Information technology — Security techniques — Digital signature schemes giving message recovery — Part 2: Integer factorization based mechanisms

[ISO/IEC 9797-1] ISO/IEC 9797-1:2011 Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher

[ISO/IEC 10116] ISO/IEC 10116:2017 Information technology — Security techniques — Modes of operation for an n-bit block cipher

[ISO/IEC 11568-2] ISO/IEC 11568-2:2012 Financial services — Key management (retail) — Part 2: Symmetric ciphers, their key management and life cycle

[ISO/IEC 11770-2] ISO/IEC 11770-2:2018 IT Security techniques — Key management — Part 2: Mechanisms using symmetric techniques

[FIPS 46-3] NIST FIPS PUB 46-3, Data Encryption Standard (DES), 1999

[FIPS 180-4] NIST FIPS PUB 180-4, Secure hash standard, 2015

[FIPS 186-4] NIST FIPS PUB 186-4, Digital Signature Standard (DSS), 2013

[FIPS 197] NIST FIPS PUB 197, Specification for the Advanced Encryption Standard (AES), 2001

[SP 800-38B] NIST Special Publication 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication, 2005

[RFC 2631] Rescorla, Eric: RFC 2631 Diffie-Hellman key agreement method, 1999

[RFC 3447] Jonsson, Jakob and Kaliski, Burt: RFC 3447, Public-key cryptography standards (PKCS) #1: RSA cryptography specifications version 2.1, 2003

[RFC 5114] Lepinski, Matt; Kent, Stephen: RFC 5114 Additional Diffie-Hellman Groups for Use with IETF Standards, 2008

[RFC 5280] D. Cooper, S. Santesson, S. Farrell, S. Boyen, R. Housley, W. Polk, RFC 5280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, 2008

[RFC 5639] Lochter, Manfred; Merkle, Johannes: RFC 5639 Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, 2010

[TR-03110] BSI: Technical Guideline TR-03110: Advanced Security Mechanisms for Machine Readable Travel Documents

[TR-03111] BSI: Technical Guideline TR-03111: Elliptic Curve Cryptography, Version 2.0, 2012

[PKCS#1] RSA Laboratories, PKCS#1 v2.2: RSA Cryptography Standard, 2012

[PKCS#3] RSA Laboratories, PKCS#3: Diffie-Hellman key-agreement standard, 1993

[Keesing2009] J. Bender, D. Kügler: Introducing the PACE solution, in: Keesing Journal of Documents & Identity, Issue 30, Keesing, 2009.

[BFK2009] J. Bender, M. Fischlin, D. Kügler: Security Analysis of the PACE Key-Agreement Protocol, in: Proceedings ISC 2009, LNCS volume 5735, Springer, 2009.

[BCIMRT2010] Brier, Eric; Coron, Jean-Sébastien; Icart, Thomas; Madore, David; Randriam, Hugues; and Tibouch, Mehdi, Efficient Indifferentiable Hashing into Ordinary Elliptic Curves, Advances in Cryptology – CRYPTO 2010, Springer-Verlag, 2010

— — — — — — — —

Apéndice A de la Parte 11

ENTROPÍA DE LAS CLAVES DE ACCESO OBTENIDAS DE LA ZLM (INFORMATIVO)

Debido a su sencillez, el control de acceso de base resultó ser un protocolo muy exitoso y se ha implantado en casi todos los eMRP.

La seguridad proporcionada por el control de acceso de base está limitada por el diseño de protocolo. Las claves de acceso de base al documento (K_{Enc} y K_{MAC}) se generan a partir de datos impresos con carácter aleatorio muy limitado. Los datos que se utilizan para la generación de las claves son el número de documento, fecha de nacimiento y fecha de caducidad. Como consecuencia, las claves resultantes tienen una entropía relativamente baja y son débiles desde el punto de vista criptográfico. La entropía real depende del tipo del número de documento. Para un documento de viaje válido por 10 años la fuerza máxima de las claves es de aproximadamente:

- 56 bits para un número de documento numérico ($365^2 * 10^{12}$ posibilidades)
- 73 bits para un número de documento alfanumérico ($365^2*36^9*10^3$ posibilidades).

Especialmente en el segundo caso, esta estimación requiere que el número de documento se elija en forma aleatoria y uniforme, lo que por lo general no sucede. Dependiendo del conocimiento del atacante, la entropía real de la clave de acceso de base al documento puede ser inferior, p. ej., si el atacante conoce todos los números de documento en uso o es capaz de correlacionar números de documento con fechas de caducidad.

No existe una forma directa de fortalecer el control de acceso de base dado que sus limitaciones son inherentes al diseño del protocolo basado en criptografía simétrica (“clave secreta”). Un mecanismo de control de acceso criptográficamente fuerte debe (además) utilizar criptografía asimétrica (“clave pública”).

El establecimiento de conexión autenticada por contraseña (PACE) se diseñó para solucionar este problema. Emplea criptografía asimétrica para establecer claves de sesión, cuya fuerza es independiente de la entropía de la contraseña utilizada. Si PACE se implanta con criptografía de curva elíptica con curvas de 256 bits y AES-128 (una elección común), las claves de sesión tendrán entropía de 128 bits.

Deben distinguirse dos tipos de ataque:

- Despumado: es un ataque en línea, es decir, el atacante trata de acceder al CI sin contacto en tiempo real, p. ej., adivinando la contraseña. Si el protocolo utilizado para proteger el CI sin contacto no tiene debilidad criptográfica, la probabilidad de éxito del atacante está dada por el tiempo en que el atacante tiene acceso al CI, la duración de un único intento para adivinar la contraseña y la entropía de pasaporte.
- Escucha furtiva: es un ataque fuera de línea, es decir, el atacante trata de descifrar la comunicación interceptada sin acceder al CI sin contacto. Si el protocolo utilizado para establecer las claves de sesión no tiene debilidad criptográfica, la probabilidad de éxito está dada por la fuerza de las claves de sesión y la potencia de cálculo de que dispone el atacante.

Para más información, véase [Keesing2009] donde figura un análisis general sobre la entropía de las claves de sesión y una comparación de BAC y PACE, y [BFK2009] que contiene un análisis criptográfico de PACE.

— — — — — — — —

Apéndice B de la Parte 11

CODIFICACIÓN DE PUNTOS PARA CORRESPONDENCIA INTEGRADA DE ECDH (INFORMATIVO)

B.1 DESCRIPCIÓN DE ALTO NIVEL DEL MÉTODO DE CODIFICACIÓN DE PUNTOS

El algoritmo toma como entradas los parámetros de curva (a, b, p, f) donde (a, b) son los coeficientes de la curva y p es la característica del campo primo sobre el cual está definida la curva.

$$E : y^2 \equiv x^3 + ax + b \ (mod\ p)$$

El orden de E es siempre de la forma fq para algún primo q y f se denomina cofactor. PACE v2 exige la generación de un punto que pertenece al subgrupo q de E que indicamos con $E[q]$. La codificación de puntos también toma como entrada un número t tal que

$$0 < t < p$$

y devuelve, en tiempo constante, un punto que pertenece a $E[q]$. Según se describe en [BCIMRT2010], la codificación de puntos se presenta en dos tipos, dependiendo del sistema de coordenadas preferido por la implantación:

- Una primera implantación, que se describe en la sección B.2, tiene como salida el punto de curva elíptico en coordenadas afines (x, y);
- Otra implantación, que se presenta en la sección B.3, tiene como salida el mismo punto en coordenadas Jacobianas (X, Y, Z).

Cualquiera sea la opción adoptada, el punto generado es idéntico en el sentido de que

$$x = XZ^{-2} \ mod\ p \text{ y } y = YZ^{-3} \ mod\ p$$

y la implantación de la fase subsiguiente de PACE v2 (la fase de intercambio de claves Diffie-Hellman de curva elíptica) puede por lo tanto aprovechar el uso de la opción que mejor corresponda a la interfaz de la API criptográfica que realiza las operaciones de curva elíptica.

Como se observa a continuación, la codificación de puntos para coordenadas afines requiere aproximadamente dos exponenciaciones modulares en módulo p mientras que la codificación de puntos para coordenadas Jacobianas requiere solamente una.

Obsérvese que para las dos implantaciones disponibles, la codificación de puntos requiere explícitamente que $p \equiv 3 \ mod\ 4$.

B.2 IMPLANTACIÓN PARA COORDENADAS AFINES

El algoritmo se implanta como sigue:

Entradas: parámetros de curva (a, b, p, f) y t tal que $0 < t < p$

Salida: un punto (x, y) en el subgrupo de orden primo $E[q]$ de E

1. Calcular $\alpha = -t^2 \bmod p$
2. Calcular $X_2 = -ba^{-1}(1+(\alpha+\alpha^2)^{-1}) \bmod p$
3. Calcular $X_3 = \alpha X_2 \bmod p$
4. Calcular $h_2 = (X_2)^3 + a X_2 + b \bmod p$
5. Calcular $h_3 = (X_3)^3 + a X_3 + b \bmod p$
6. Calcular $U = t^3 h_2 \bmod p$
7. Calcular $A = (h_2)^{p-1-(p+1)/4} \bmod p$
8. Si $A^2 h_2 = 1 \bmod p$ defínase $(x, y) = (X_2, A h_2 \bmod p)$
9. En caso contrario defínase $(x, y) = (X_3, A U \bmod p)$
10. Salida $(x, y) = [f] (x, y)$.

Notas sobre la implantación

Sin contar las multiplicaciones y adiciones modulares, el tiempo de ejecución de la implantación anterior está dominado por dos exponenciaciones modulares:

- la Etapa 2 puede escribirse

$$X_2 = -ba^{-1}(1+(\alpha+\alpha^2)^{-1}) = -b(1+\alpha+\alpha^2)\,(a(\alpha+\alpha^2))^{p-2} \bmod p$$

  que esencialmente significa una exponenciación modular con exponente $p-2$;

- la Etapa 7 es una exponenciación modular con exponente $p-1-(p+1)/4$.

Nota.— La Etapa 10 requiere una multiplicación escalar por el cofactor f. *Para muchas curvas, el cofactor es igual a 1 de modo que esta multiplicación escalar puede evitarse.*

B.3 IMPLANTACIÓN PARA COORDENADAS JACOBIANAS

El algoritmo se implanta como sigue:

Entradas: parámetros de curva (a, b, p, f) y t tal que $0 < t < p$

Salida: un punto (X, Y, Z) en el subgrupo E de orden primo $E[q]$ de E

1. Calcular $\alpha = -t^2 \bmod p$
2. Calcular $Z = a(\alpha+\alpha^2) \bmod p$
3. Calcular $X_2 = -bZ\,(1+\alpha+\alpha^2) \bmod p$
4. Calcular $X_3 = \alpha X_2 \bmod p$
5. Calcular $h_2 = (X_2)^3 + a X_2 Z^4 + b Z^6 \bmod p$
6. Calcular $h_3 = (X_3)^3 + a X_3 Z^4 + b Z^6 \bmod p$
7. Calcular $U = -\alpha t h_2 \bmod p$

8. Calcular $A = (h_2)^{p-1-(p+1)/4} \bmod p$
9. Si $A^2 h_2 = 1 \bmod p$ defínase $(X, Y, Z) = (X_2, A\, h_2 \bmod p, Z)$
10. En caso contrario defínase $(X, Y, Z) = (X_3, A\, U \bmod p, Z)$
11 Salida $(X, Y, Z) = [f](X, Y, Z)$.

Notas sobre la implantación

Sin contar las multiplicaciones y adiciones modulares, el tiempo de ejecución de la implantación anterior está dominado por la única exponenciación modular de la Etapa 7. Por consiguiente, se prevé que sea aproximadamente el doble de rápido que la implantación para coordenadas afines.

Nota.— La multiplicación escalar de la Etapa 10 puede evitarse completamente cuando el cofactor f *es igual a 1.*

— — — — — — — —

Apéndice C de la Parte 11

SEMÁNTICA DE PUESTA A PRUEBA (INFORMATIVO)

Considérese un protocolo de puesta a prueba-respuesta basado en firma entre una microplaqueta (CI) del eMRTD y un terminal (IFD), donde la microplaqueta del eMRTD quiere demostrar conocimiento de su clave privada SK_{IC}:

- El terminal envía una puesta a prueba *c* elegida aleatoriamente a la microplaqueta del eMRTD.
- La microplaqueta del eMRTD responde con la firma s=**Sign**(SK_{IC},c).

Si bien este es un protocolo muy sencillo y eficiente, la microplaqueta del eMRTD realmente firma el mensaje *c* sin conocer la semántica del mismo. Dado que las firmas proporcionan una prueba de autenticidad transferible, cualquier tercera parte puede – en principio – convencerse de que la microplaqueta del eMRTD ha firmado efectivamente este mensaje.

Aunque *c* debería ser una cadena de bits aleatoria, el terminal puede también generar esta cadena de bits en forma impredecible pero (públicamente) verificable, p. ej., sea SK_{IFD} la clave privada del terminal y la puesta a prueba generada utilizando un plan de firmas con recuperación de mensaje.

$$c = \textbf{Sign}(SK_{IFD}, ID_{IC}||Date||Time||Location)$$

La firma garantiza que el terminal ha generado verdaderamente esta puesta a prueba. Debido al carácter transferible de la firma del terminal, cualquier tercera parte con confianza en el terminal y conocimiento de la clave pública correspondiente PK_{IFD} puede verificar que la puesta a prueba fue creada correctamente mediante la verificación de esta firma. Además, debido al carácter transferible de la firma de la microplaqueta del eMRTD en la puesta a prueba, la tercera parte puede concluir que la afirmación es verdadera: la microplaqueta del eMRTD estaba verdaderamente en cierta fecha y hora en un determinado lugar.

En el lado positivo, los Estados pueden utilizar semántica de puesta a prueba para su uso interno, p. ej., para demostrar que una determinada persona ha inmigrado verdaderamente. En el lado negativo, esas pruebas pueden utilizarse indebidamente para rastrear personas. En particular, dado que la autenticación activa no está limitada a las terminales autorizadas, el uso indebido es posible. El peor escenario sería unas microplaquetas de eMRTD que proporcionen la autenticación activa sin control de acceso de base. En este caso, un sistema de rastreo muy poderoso puede establecerse colocando módulos de soporte físico seguros en lugares prominentes. Los registros resultantes no pueden falsificarse debido a las firmas. El control de acceso de base disminuye este problema en cierta medida, dado que se requiere interacción con el titular. No obstante, el problema permanece, pero se limita a lugares donde el documento de viaje del titular es leído de todas maneras, p. ej., por líneas aéreas u hoteles.

Se puede objetar que especialmente en un escenario sin contacto, las puestas a prueba pueden escucharse furtivamente y volverse a utilizar en una fecha, hora o lugar diferentes y así hacer que la prueba sea por lo menos no fiable. Si bien las puestas a prueba de escucha furtiva son técnicamente posibles, el argumento sigue siendo inválido. Por hipótesis, se confía en que un terminal produce puestas a prueba correctamente y que puede suponerse que ha verificado la identidad de la microplaqueta del eMRTD antes de iniciar la autenticación activa. Entonces, la puesta a prueba escuchada en forma furtiva contendrá una identidad diferente de la identidad del demostrador que firma la puesta a prueba.

— — — — — — — —

Apéndice D de la Parte 11

EJEMPLO ELABORADO: CONTROL DE ACCESO DE BASE (INFORMATIVO)

D.1 CÁLCULO DE LAS CLAVES A PARTIR DE UNA CLAVE SEMILLA (K_{seed})

Esta sección proporciona un ejemplo para calcular claves 3DES a partir de un valor semilla K_{seed}. Este procedimiento se utilizará como "subrutina" en los ejemplos para control de acceso de base.

Entrada:

K_{seed} = '239AB9CB282DAF66231DC5A4DF6BFBAE'

Calcular clave de cifrado (c = '00000001'):

1. Concatenar K_{seed} y c:
   D = '239AB9CB282DAF66231DC5A4DF6BFBAE00000001'

2. Calcular la condensación SHA-1 de D:
   H_{SHA-1}(D) = 'AB94FCEDF2664EDFB9B291F85D7F77F27F2F4A9D'

3. Formar claves DES K_a y K_b, para utilizar como primera y segunda claves para 3DES (es decir, la clave 3DES es la concatenación de K_a y K_b):
   $K_{a'}$ = 'AB94FCEDF2664EDF'
   $K_{b'}$ = 'B9B291F85D7F77F2'

4. Ajustar bits de paridad:
   K_a = 'AB94FDECF2674FDF'
   K_b = 'B9B391F85D7F76F2'

Calcular clave de cálculo MAC (c = '00000002'):

1. Concatenar K_{seed} y c:
   D = '239AB9CB282DAF66231DC5A4DF6BFBAE00000002'

2. Calcular la condensación SHA-1 de D:
   H_{SHA-1}(D) = '7862D9ECE03C1BCD4D77089DCF131442814EA70A'

3. Formar claves $K_{a'}$ y $K_{b'}$:
   $K_{a'}$ = '7862D9ECE03C1BCD'
   $K_{b'}$ = '4D77089DCF131442'

4. Ajustar bits de paridad:
   K_a = '7962D9ECE03D1ACD'
   K_b = '4C76089DCE131543'

D.2 OBTENCIÓN DE CLAVES DE ACCESO DE BASE AL DOCUMENTO (K_{Enc} Y K_{MAC})

En esta sección se proporcionan ejemplos de la forma en que se obtienen las claves de acceso de base de la ZLM.

ZLM del DV2, el número de documento supera los 9 caracteres

1. Leer la ZLM
   ZLM = `I<UTOSTEVENSON<<PETER<JOHN<<<<<<<<<<`
   `D23145890<UTO3407127M95071227349<<<8`

2. Construir la 'ZLM_information' a partir de la ZLM
   Número de documento = `D23145890734` dígito de verificación = `9`
   Fecha de nacimiento = `340712` dígito de verificación = `7`
   Fecha de caducidad = `950712` dígito de verificación = `2`
   MRZ_information = `D23145890734934071279507122`

Continuar con la Etapa 3.

ZLM de DV2, número de documento con 9 caracteres

1. Leer la ZLM:
   ZLM = `I<UTOERIKSSON<<ANNA<MARIA<<<<<<<<<<`
   `L898902C<3UTO6908061F9406236<<<<<<<8`

2. Construir la 'ZLM_information' de la ZLM:
   Número de documento = `L898902C<` dígito de verificación = `3`
   Fecha de nacimiento = `690806` dígito de verificación = `1`
   Fecha de caducidad = `940623` dígito de verificación = `6`
   MRZ_information = `L898902C<3690806194062 36`

Continuar con la Etapa 3.

ZLM de DV1, número de documento con 9 caracteres

1. Leer la ZLM
   ZLM = `I<UTOD23145890<7349<<<<<<<<<<`
   `3407127M9507122UTO<<<<<<<<<<2`
   `STEVENSON<<PETER<JOHN<<<<<<<<<`

2. Construir la 'ZLM_information' de la ZLM
   Número de documento = `D23145890734` dígito de verificación = `9`
   Fecha de nacimiento = `340712` dígito de verificación = `7`
   Fecha de caducidad = `950712` dígito de verificación = `2`
   MRZ_information = `D23145890734934071279507122`

Continuar con la Etapa 3.

ZLM de DV1, número de documento con 9 caracteres

1. Leer la ZLM
   ZLM = `I<UTOL898902C<3<<<<<<<<<<<<<<`
   `6908061F9406236UTO<<<<<<<<<<1`
   `ERIKSSON<<ANNA<MARIA<<<<<<<<<<`

2. Construir la 'ZLM_information' de la ZLM
   Número de documento = L898902C< dígito de verificación = 3
   Fecha de nacimiento = 690806 dígito de verificación = 1
   Fecha de caducidad = 940623 dígito de verificación = 6
   MRZ_information = L898902C<3690806194062360

3. Calcular la condensación SHA-1 para 'MRZ_information':
   H_{SHA-1}(MRZ_information) = '239AB9CB282DAF66231DC5A4DF6BFBAEDF477565'

4. Tomar los 16 bytes más significativos para formar K_{seed}:
   K_{seed} = '239AB9CB282DAF66231DC5A4DF6BFBAE'

5. Calcular las claves de acceso de base (K_{Enc} y K_{MAC}) con arreglo a la sección 9.7.1/Apéndice D.1:
   K_{Enc} = 'AB94FDECF2674FDFB9B391F85D7F76F2'
   K_{MAC} = '7962D9ECE03D1ACD4C76089DCE131543'

D.3 AUTENTICACIÓN Y ESTABLECIMIENTO DE CLAVES DE SESIÓN

En esta sección se proporciona un ejemplo para ejecutar el control de acceso de base.

Sistema de inspección:

1. Pedir un número aleatorio de 8 bytes del CI sin contacto del eMRTD:

Comando APDU:				
CLA	**INS**	**P1**	**P2**	**Le**
00	84	00	00	08

Respuesta APDU:	
Campo de datos de respuesta	**SW1-SW2**
RND.IC	9000

   RND.IC = '4608F91988702212'

2. Generar un aleatorio de 8 bytes y un aleatorio de 16 bytes:
   RND.IFD = '781723860C06C226'
   K_{IFD} = '0B795240CB7049B01C19B33E32804F0B'

3. Concatenar RND.IFD, RND.IC y K_{IFD}:
   S = '781723860C06C2264608F91988702212
   0B795240CB7049B01C19B33E32804F0B'

4. Cifrar S con clave 3DES K_{Enc}:
   E_{IFD} = '72C29C2371CC9BDB65B779B8E8D37B29
   ECC154AA56A8799FAE2F498F76ED92F2'

5. Calcular MAC por encima de E_{IFD} con clave 3DES K_{MAC}:
   M_{IFD} = '5F1448EEA8AD90A7'

6. Construir datos de comandos para EXTERNAL AUTHENTICATE y enviar comando APDU al CI sin contacto del eMRTD:
   cmd_data = '72C29C2371CC9BDB65B779B8E8D37B29ECC154AA
   56A8799FAE2F498F76ED92F25F1448EEA8AD90A7'

Comando APDU:						
CLA	**INS**	**P1**	**P2**	**Lc**	**Campo de datos de comando**	**Le**
00	82	00	00	28	cmd_data	28

CI sin contacto del eMRTD:

1. Descifrar y verificar datos recibidos y comparar RND.IC con la respuesta a GET CHALLENGE.

2. Generar un aleatorio de 16 bytes:
   K_{IC} = '0B4F80323EB3191CB04970CB4052790B'

3. Calcular XOR de K_{IFD} y K_{IC}:
   K_{seed} = '0036D272F5C350ACAC50C3F572D23600'

4. Calcular claves de sesión (KS_{Enc} y KS_{MAC}) con arreglo a la sección 9.7.1/Apéndice D.1:
   KS_{Enc} = '979EC13B1CBFE9DCD01AB0FED307EAE5'
   KS_{MAC} = 'F1CB1F1FB5ADF208806B89DC579DC1F8'

5. Calcular contador de secuencia de envío:
   SSC = '887022120C06C226'

6. Concatenar RND.IC, RND.IFD y K_{IC}:
   R = '4608F91988702212781723860C06C226
   0B4F80323EB3191CB04970CB4052790B'

7. Cifrar R con clave 3DES K_{Enc}:
   E_{IC} = '46B9342A41396CD7386BF5803104D7CE
   DC122B9132139BAF2EEDC94EE178534F'

8. Calcular MAC por encima de E_{IC} con clave 3DES K_{MAC}:
   M_{IC} = '2F2D235D074D7449'

9. Construir datos de respuesta a EXTERNAL AUTHENTICATE y enviar respuesta APDU al sistema de inspección:
   resp_data = '46B9342A41396CD7386BF5803104D7CEDC122B91
   32139BAF2EEDC94EE178534F2F2D235D074D7449'

Respuesta APDU:	
Campo de datos de respuesta	**SW1-SW2**
resp_data	9000

Sistemas de inspección:

1. Descifrar y verificar datos recibidos y comparar RND.IFD recibidos con RND.IFD generados.

2. Calcular XOR de K_{IFD} y K_{IC}:
   K_{seed} = '0036D272F5C350ACAC50C3F572D23600'

3. Calcular claves de sesión (KS_{Enc} y KS_{MAC}) con arreglo a la Sección 9.7.1/Apéndice D.1:
   KS_{Enc} = '979EC13B1CBFE9DCD01AB0FED307EAE5'
   KS_{MAC} = 'F1CB1F1FB5ADF208806B89DC579DC1F8'

4. Calcular contador de secuencia de envío:
   SSC = '887022120C06C226'

D.4 CONSTRUCCIÓN SEGURA DE MENSAJES

Después de autenticar y establecer las claves de sesión, el sistema de inspección selecciona el EF.COM (fichero ID = '011E') y lee los datos utilizando construcción segura de mensajes. Se utilizarán los cálculos KS_{Enc}, KS_{MAC} y SSC (Etapas 3 y 4 anteriores del sistema de inspección).

Primero se seleccionará EF.COM, después se leerán los 4 primeros bytes de este fichero de modo de poderse determinar la longitud de la estructura de los ficheros y después se leerán los bytes restantes.

1. Seleccionar EF.COM

   Comando APDU no protegida:

CLA	INS	P1	P2	Lc	Campo de datos de orden
00	A4	02	0C	02	01 1E

   a) Enmascarar byte de clase y rellenar encabezamiento de comado:
      CmdHeader = '0CA4020C80000000'

   b) Rellenar datos:
      Data = '011E800000000000'

   c) Cifrar datos con KS_{Enc}:
      EncryptedData = '6375432908C044F6'

   d) Construir DO'87':
      DO87 = '8709016375432908C044F6'

   e) Concatenar CmdHeader y DO'87':
      M = '0CA4020C8000000087090163754329 08C044F6'

f) Calcular MAC de M:

i) Incrementar SSC en 1:

SSC = '887022120C06C227'

ii) Concatenar SSC y M y añadir relleno:

N = '887022120C06C2270CA4020C80000000
8709016375432908C044F68000000000'

iii) Calcular MAC por encima de N con KS_{MAC}:

CC = 'BF8B92D635FF24F8'

g) Construir DO'8E':

DO8E = '8E08BF8B92D635FF24F8'

h) Construir y enviar APDU protegida:

ProtectedAPDU = '0CA4020C158709016375432908C0
44F68E08BF8B92D635FF24F800'

i) Recibir respuesta APDU del CI sin contacto del eMRTD:

RAPDU = '990290008E08FA855A5D4C50A8ED9000'

j) Verificar RAPDU CC calculando MAC de DO'99':

i) Incrementar SSC en 1:

SSC = '887022120C06C228'

ii) Concatenar SSC y DO'99' y añadir relleno:

K = '887022120C06C22899029000800000000'

iii) Calcular MAC con KS_{MAC}:

CC' = 'FA855A5D4C50A8ED'

iv) Comparar CC' con datos de DO'8E' de RAPDU.

'FA855A5D4C50A8ED' == 'FA855A5D4C50A8ED' ? SÍ.

2. Leer binario de cuatro primeros bytes:

Comando APDU no protegida:

CLA	INS	P1	P2	Le
00	B0	00	00	04

a) Enmascarar byte de clase y rellenar encabezamiento de comando:

CmdHeader = '0CB0000080000000'

b) Construir DO'97':

DO97 = '970104'

c) Concatenar CmdHeader y DO'97':

M = '0CB000008000000970104'

d) Calcular MAC de M:

i) Incrementar SSC en 1:
SSC = ‘887022120C06C229’

ii) Concatenar SSC y M y añadir relleno:
N = ‘887022120C06C2290CB00000
80000000970104800000000’

iii) Calcular MAC por encima de N con KSMAC:
CC = ‘ED6705417E96BA55’

e) Construir DO‘8E’:
DO8E = ‘8E08ED6705417E96BA55’

f) Construir y enviar APDU protegida:
ProtectedAPDU = ‘0CB000000D9701048E08ED6705417E96BA5500’

g) Recibir respuesta APDU del CI sin contacto del eMRTD:
RAPDU = ‘8709019FF0EC34F992265199029000
8E08AD55CC17140B2DED9000’

h) Verificar RAPDU CC calculando MAC de concatenación DO‘87’ y DO‘99’:

i) Incrementar SSC en 1:
SSC = ‘887022120C06C22A’

ii) Concatenar SSC, DO‘87’ y DO‘99’ y añadir relleno:
K = ‘887022120C06C22A8709019F
F0EC34F99226519902900080’

iii) Calcular MAC con KS_{MAC}:
CC’ = ‘AD55CC17140B2DED’

iv) Comparar CC’ con datos de DO‘8E’ de RAPDU:
‘AD55CC17140B2DED’ == ‘AD55CC17140B2DED’ ? SÍ.

i) Descifrar datos de DO‘87’ con KS_{Enc}:
DecryptedData = ‘60145F01’

j) Determinar longitud de estructura:
L = ‘14’ + 2 = 22 bytes

3. Leer binario de restantes 18 bytes del desplazamiento 4:

Comando APDU no protegida:

CLA	INS	P1	P2	Le
00	B0	00	04	12

a) Enmascarar byte de clase y rellanar encabezamiento de comando:
CmdHeader = ‘0CB0000480000000’

b) Construir DO‘97’:
DO97 = ‘970112’

c) Concatenar CmdHeader y DO‘97’:
M = ‘0CB000048000000970112’

d) Calcular MAC de M:

i) Incrementar SSC en 1:
SSC = ‘887022120C06C22B’

ii) Concatenar SSC y M y añadir relleno:
N = ‘887022120C06C22B0CB00004
80000000970112800000000’

iii) Calcular MAC por encima de N con KS_{MAC}:
CC = ‘2EA28A70F3C7B535’

e) Construir DO‘8E’:
DO8E = ‘8E082EA28A70F3C7B535’

f) Construir y enviar APDU protegida:
ProtectedAPDU = ‘0CB000040D9701128E082EA28A70F3C7B53500’

g) Recibir respuesta APDU del CI sin contacto del eMRTD:
RAPDU = ‘871901FB9235F4E4037F2327DCC8964F1F9B8C30F42
C8E2FFF224A990290008E08C8B2787EAEA07D749000’

h) Verificar RAPDU CC calculando MAC de concatenación DO‘87’ y DO‘99’:

i) Incrementar SSC en 1:
SSC = ‘887022120C06C22C’

ii) Concatenar SSC, DO‘87’ y DO‘99’ y añadir relleno:
K = ‘887022120C06C22C871901FB9235F4E4037F232
7DCC8964F1F9B8C30F42C8E2FFF224A99029000’

iii) Calcular MAC con KS_{MAC}:
CC’ = ‘C8B2787EAEA07D74’

iv) Comparar CC’ con datos de DO‘8E’ de RAPDU:
‘C8B2787EAEA07D74’ == ‘C8B2787EAEA07D74’ ? SÍ.

i) Descifrar datos de DO‘87’ con KS_{Enc}:
DecryptedData = ‘04303130365F36063034303030305C026175’

RESULTADOS:

Datos EF.COM = ‘60145F0104303130365F36063034303030305C026175’

— — — — — — — —

Apéndice E de la Parte 11

EJEMPLO ELABORADO: AUTENTICACIÓN PASIVA (INFORMATIVO)

Etapa 1. Leer el objeto de seguridad de documento (SO_D) [contiene opcionalmente el certificado del firmante del documento (C_{DS})] del CI sin contacto.

Etapa 2: Leer el firmante del documento (DS) del objeto de seguridad del documento (SO_D).

Etapa 3: El sistema de inspección verifica SO_D utilizando la clave pública del firmante del documento.

Etapa 4: El sistema de inspección verifica C_{DS} utilizando la clave pública de CA de firma de país.

Si ambas verificaciones en las Etapas 3 y 4 son correctas, esto asegura que el contenido de SO_D es fiable y puede utilizarse en el proceso de inspección.

Etapa 5: Leer los grupos de datos de la LDS pertinentes.

Etapa 6: Calcular las condensaciones de los grupos de datos pertinentes.

Etapa 7: Comparar las condensaciones calculadas con los valores de condensación correspondientes en la SO_D.

Si los valores de condensación de la Etapa 7 son idénticos, esto asegura que el contenido de grupo de datos es auténtico y no ha sufrido cambios.

— — — — — — — —

Apéndice F de la Parte 11

EJEMPLO ELABORADO: AUTENTICACIÓN ACTIVA (INFORMATIVO)

Este ejemplo elaborado utiliza los valores siguientes:

1. Mecanismo basado en factorización de enteros: RSA
2. Longitud de módulo (k): 1 024 bits (128 bytes)
3. Algoritmo de condensación: SHA-1

Sistema de inspección:

Etapa 1. Generar un aleatorio de 8 bytes:
RND.IFD = 'F173589974BF40C6'

Etapa 2. Construir comando para autenticación interna y enviar comando APDU al CI sin contacto del eMRTD:

Comando APDU

CLA	INS	P1	P2	Lc	Campo de datos de comando	Le
00	88	00	00	08	RND.IFD	00

CI sin contacto del eMRTD:

Etapa 3. Determinar M_2 de APDU recibido:
M_2 = 'F173589974BF40C6'

Etapa 4. Crear indicador de fin:
T = 'BC' (es decir, SHA-1)
t (longitud de T en octetos) = 1

Etapa 5. Determinar longitudes:
a. $c = k - L_h - 8t - 4 = 1024 - 160 - 8 - 4 = 852$ bits
b. $L_{M1} = c - 4 = 848$ bits

Etapa 6. Generar nonce M_1 de longitud L_{M1}:

```
M1 =  '9D2784A67F8E7C659973EA1AEA25D95B
       6C8F91E5002F369F0FBDCE8A3CEC1991
       B543F1696546C5524CF23A5303CD6C98
       599F40B79F377B5F3A1406B3B4D8F967
       84D23AA88DB7E1032A405E69325FA91A
       6E86F5C71AEA978264C4A207446DAD4E
       7292E2DCDA3024B47DA8'
```

Etapa 7. Crear M:

M = M_1 | M_2 = ‘9D2784A67F8E7C659973EA1AEA25D95B
6C8F91E5002F369F0FBDCE8A3CEC1991
B543F1696546C5524CF23A5303CD6C98
599F40B79F377B5F3A1406B3B4D8F967
84D23AA88DB7E1032A405E69325FA91A
6E86F5C71AEA978264C4A207446DAD4E
7292E2DCDA3024B47DA8F173589974BF
40C6’

Etapa 8. Calcular resumen SHA-1 de M:

H = SHA-1(M) = ‘C063AA1E6D22FBD976AB0FE73D94D2D9
C6D88127’

Etapa 9.[2] Construir representación del mensaje:

F = ‘6A’ | M_1 | H | T =
‘6A9D2784A67F8E7C659973EA1AEA25D9
5B6C8F91E5002F369F0FBDCE8A3CEC19
91B543F1696546C5524CF23A5303CD6C
98599F40B79F377B5F3A1406B3B4D8F9
6784D23AA88DB7E1032A405E69325FA9
1A6E86F5C71AEA978264C4A207446DAD
4E7292E2DCDA3024B47DA8C063AA1E6D
22FBD976AB0FE73D94D2D9C6D88127BC’

Etapa 10. Cifrar F con la clave privada de autenticación activa para formar la firma:

S = ‘756B683B036A6368F4A2EB29EA700F96
E26100AFC0809F60A91733BA29CAB362
8CB1A017190A85DADE83F0B977BB513F
C9C672E5C93EFEBBE250FE1B722C7CEE
F35D26FC8F19219C92D362758FA8CB0F
F68CEF320A8753913ED25F69F7CEE772
6923B2C43437800BBC9BC028C49806CF
2E47D16AE2B2CC1678F2A4456EF98FC9’

Etapa 11. Construir datos de respuesta para INTERNAL AUTHENTICATE y enviar respuesta APDU al sistema de inspección:

Respuesta APDU:

Campo de datos de respuesta	**SW1-SW2**
S	9000

2. Dado que no se devuelve la parte conocida (RND.IFD), pero debe añadirse por el propio IFD, se aplica la recuperación parcial (‘6A’).

Sistema de inspección:

Etapa 12. Descifrar la firma con la clave pública:

F = ‘6A9D2784A67F8E7C659973EA1AEA25D9
5B6C8F91E5002F369F0FBDCE8A3CEC19
91B543F1696546C5524CF23A5303CD6C
98599F40B79F377B5F3A1406B3B4D8F9
6784D23AA88DB7E1032A405E69325FA9
1A6E86F5C71AEA978264C4A207446DAD
4E7292E2DCDA3024B47DA8C063AA1E6D
22FBD976AB0FE73D94D2D9C6D88127BC’

Etapa 13. Determinar algoritmo de condensación por indicador de fin T*:

T = ‘BC’ (es decir, SHA-1)

Etapa 14. Extraer resumen:

D = ‘C063AA1E6D22FBD976AB0FE73D94D2D9
C6D88127’

Etapa 15. Extraer M_1:

M_1 = ‘9D2784A67F8E7C659973EA1AEA25D95B
6C8F91E5002F369F0FBDCE8A3CEC1991
B543F1696546C5524CF23A5303CD6C98
599F40B79F377B5F3A1406B3B4D8F967
84D23AA88DB7E1032A405E69325FA91A
6E86F5C71AEA978264C4A207446DAD4E
7292E2DCDA3024B47DA8’

Etapa 16. El encabezamiento indica recuperación parcial, pero la firma tiene longitud de módulo, entonces concatenar M_1 con M_2 conocido (es decir, RND.IFD):

M* = ‘9D2784A67F8E7C659973EA1AEA25D95B
6C8F91E5002F369F0FBDCE8A3CEC1991
B543F1696546C5524CF23A5303CD6C98
599F40B79F377B5F3A1406B3B4D8F967
84D23AA88DB7E1032A405E69325FA91A
6E86F5C71AEA978264C4A207446DAD4E
7292E2DCDA3024B47DA8F173589974BF
40C6’

Etapa 17. Calcular resumen SHA-1 de M*:

D* = ‘C063AA1E6D22FBD976AB0FE73D94D2D9
C6D88127’

Etapa 18. Comparar D y D*:

Si D es igual a D*, entonces la verificación ha sido exitosa.

— — — — — — — —

Apéndice G de la Parte 11

EJEMPLO ELABORADO: PACE – CORRESPONDENCIA GENÉRICA (INFORMATIVO)

En el presente apéndice se proporcionan dos ejemplos elaborados para el protocolo PACE según se define en la sección 4.4 utilizando la correspondencia genérica. El primer ejemplo se basa en ECDH mientras que el segundo utiliza DH. Todos los números contenidos en las tablas están en notación hexadecimal.

En ambos ejemplos, la ZLM se utiliza como contraseña. Esto también conduce a la misma clave simétrica K_{π}. Los campos de datos pertinentes de la ZLM incluyendo los dígitos de verificación son:

- Número de documento: T220001293;
- Fecha de nacimiento: 6408125;
- Fecha de caducidad: 1010318.

Por consiguiente, la codificación K de la ZLM y la clave de cifrado obtenida K_{π} son:

K	7E2D2A41 C74EA0B3 8CD36F86 3939BFA8 E9032AAD
K_{π}	89DED1B2 6624EC1E 634C1989 302849DD

G.1 EJEMPLO BASADO EN ECDH

El ejemplo se basa en ECDH aplicando los parámetros de dominio normalizados BrainpoolP256r1 (véase [RFC 5639]).

La primera sección introduce la correspondiente `PACEInfo`. Posteriormente, se enumeran y examinan las APDU intercambiadas, incluyendo todos los nonces generados y claves efímeras.

Parámetros de curva elíptica

Utilizando parámetros de dominio normalizados, toda la información requerida para ejecutar PACE está dada por la estructura de datos `PACEInfo`. En particular, no se requiere `PACEDomainParameterInfo`.

PACEInfo	3012060A 04007F00 07020204 02020201 0202010D

La estructura detallada de `PACEInfo` se desglosa en la tabla siguiente.

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
30	12		SECUENCIA	PACEInfo
06	0A	04 00 7F 00 07 02 02 04 02 02	IDENTIFICADOR DE OBJETO	PACE con ECDH, correspondencia genérica y claves de sesión AES 128
02	01	02	ENTERO	Versión 2
02	01	0D	ENTERO	Parámetros de dominio normalizados Brainpool P256r1

Por conveniencia, a continuación se presenta la codificación ASN.1 de los parámetros de dominio BrainpoolP256r1.

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
30	81 EC		SECUENCIA	Parámetro de dominio
06	07	2A 86 48 CE 3D 02 01	IDENTIFICADOR DE OBJETO	Algorithm id-ecPublicKey
30	81 E0		SECUENCIA	Parámetro de dominio
02	01	01	ENTERO	Versión
30	2C		SECUENCIA	Campo subyacente
06	07	2A 86 48 CE 3D 01 01	IDENTIFICADOR DE OBJETO	Campo primo
02	21	00 A9 FB 57 DB A1 EE A9 BC 3E 66 0A 90 9D 83 8D 72 6E 3B F6 23 D5 26 20 28 20 13 48 1D 1F 6E 53 77	ENTERO	Primo p
30	44		SECUENCIA	Ecuación de curva
04	20	7D 5A 09 75 FC 2C 30 57 EE F6 75 30 41 7A FF E7 FB 80 55 C1 26 DC 5C 6C E9 4A 4B 44 F3 30 B5 D9	CADENA DE OCTETOS	Parámetro a
04	20	26 DC 5C 6C E9 4A 4B 44 F3 30 B5 D9 BB D7 7C BF 95 84 16 29 5C F7 E1 CE 6B CC DC 18 FF 8C 07 B6	CADENA DE OCTETOS	Parámetro b

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
04	41		CADENA DE OCTETOS	Grupo generador G
		04	-	Punto no comprimido
		8B D2 AE B9 CB 7E 57 CB 2C 4B 48 2F FC 81 B7 AF B9 DE 27 E1 E3 BD 23 C2 3A 44 53 BD 9A CE 32 62	-	Coordenada x
		54 7E F8 35 C3 DA C4 FD 97 F8 46 1A 14 61 1D C9 C2 77 45 13 2D ED 8E 54 5C 1D 54 C7 2F 04 69 97	-	Coordenada y
02	21	00 A9 FB 57 DB A1 EE A9 BC 3E 66 0A 90 9D 83 8D 71 8C 39 7A A3 B5 61 A6 F7 90 1E 0E 82 97 48 56 A7	ENTERO	Orden de grupo n
02	01	01	ENTERO	Cofactor f

Flujo de aplicación en el ejemplo basado en ECDH

Para inicializar PACE, el terminal envía el comando MSE:Set AT a la microplaqueta.

T>C :	`00 22 C1 A4 0F 80 0A 04 00 7F 00 07 02 02 04 02 02 83 01 01`
C>T :	`90 00`

Aquí, T>C es una abreviatura para la APDU enviada del terminal a la microplaqueta mientras que C>T denota la respuesta correspondiente enviada por la microplaqueta al terminal. La codificación del comando se explica en la tabla siguiente.

Comando				
CLA	00		Llano	
INS	22		Gestionar entorno de seguridad	
P1/P2	C1 A4		Establecer plantillas de autenticación para autenticación mutua	
Lc	0F		Longitud del campo de datos	
Datos	Rótulo	Longitud	Valor	Comentario
	80	0A	04 00 7F 00 07 02 02 04 02 02	Mecanismo criptográfico: PACE con ECDH, correspondencia genérica y claves de sesiones AES128
	83	01	01	Contraseña: MRZ (ZLM)

Respuesta		
Bytes de estado	90 00	Procesamiento normal

Nonce cifrado

A continuación, la microplaqueta genera en forma aleatoria el nonce s y lo cifra mediante K_π.

Nonce s descifrado	3F00C4D3 9D153F2B 2A214A07 8D899B22
Nonce z cifrado	95A3A016 522EE98D 01E76CB6 B98B42C3

El nonce cifrado es interrogado por el terminal.

T>C :	10 86 00 00 02 7C 00 00
C>T :	7C 12 80 10 95 A3 A0 16 52 2E E9 8D 01 E7 6C B6 B9 8B 42 C3 90 00

La codificación del comando APDU y la respuesta correspondiente figuran en la tabla siguiente.

Comando				
CLA	10		Encadenamiento de comandos	
INS	86		GENERAL AUTHENTICATE	
P1/P2	00 00		Claves y protocolo implícitamente conocidos	
Lc	02		Longitud de los datos	
Datos	Rótulo	Longitud	Valor	Comentario
	7C	00	-	Ausente
Le	00		La longitud máxima de bytes prevista del campo de datos de respuesta es 256	
Respuesta				
Datos	Rótulo	Longitud	Valor	Comentario
	7C	12		Datos de autenticación dinámica
	80	10	95 A3 A0 16 52 2E E9 8D 01 E7 6C B6 B9 8B 42 C3	Nonce cifrado
Bytes de estado	90 00		Procesamiento normal	

Nonce de correspondencia

El nonce se hace corresponder con un generador de grupo efímero a través de correspondencia genérica. Las claves efímeras requeridas elegidas en forma aleatoria también se recogen en la tabla siguiente.

Clave privada del terminal	7F4EF07B 9EA82FD7 8AD689B3 8D0BC78C F21F249D 953BC46F 4C6E1925 9C010F99
Clave pública del terminal	7ACF3EFC 982EC455 65A4B155 129EFBC7 4650DCBF A6362D89 6FC70262 E0C2CC5E, 544552DC B6725218 799115B5 5C9BAA6D 9F6BC3A9 618E70C2 5AF71777 A9C4922D
Clave privada de la microplaqueta	498FF497 56F2DC15 87840041 839A8598 2BE7761D 14715FB0 91EFA7BC E9058560
Clave pública de la microplaqueta	824FBA91 C9CBE26B EF53A0EB E7342A3B F178CEA9 F45DE0B7 0AA60165 1FBA3F57, 30D8C879 AAA9C9F7 3991E61B 58F4D52E B87A0A0C 709A49DC 63719363 CCD13C54
Secreto compartido H	60332EF2 450B5D24 7EF6D386 8397D398 852ED6E8 CAF6FFEE F6BF85CA 57057FD5, 0840CA74 15BAF3E4 3BD414D3 5AA4608B 93A2CAF3 A4E3EA4E 82C9C13D 03EB7181
Generador de correspondencia $\hat{G}$	8CED63C9 1426D4F0 EB1435E7 CB1D74A4 6723A0AF 21C89634 F65A9AE8 7A9265E2, 8C879506 743F8611 AC33645C 5B985C80 B5F09A0B 83407C1B 6A4D857A E76FE522

Las APDU siguientes se intercambian entre el terminal y la microplaqueta para hacer la correspondencia con el nonce.

T>C :	10 86 00 00 45 7C 43 81 41 04 7A CF 3E FC 98 2E C4 55 65 A4 B1 55 12 9E FB C7 46 50 DC BF A6 36 2D 89 6F C7 02 62 E0 C2 CC 5E 54 45 52 DC B6 72 52 18 79 91 15 B5 5C 9B AA 6D 9F 6B C3 A9 61 8E 70 C2 5A F7 17 77 A9 C4 92 2D 00
C>T :	7C 43 82 41 04 82 4F BA 91 C9 CB E2 6B EF 53 A0 EB E7 34 2A 3B F1 78 CE A9 F4 5D E0 B7 0A A6 01 65 1F BA 3F 57 30 D8 C8 79 AA A9 C9 F7 39 91 E6 1B 58 F4 D5 2E B8 7A 0A 0C 70 9A 49 DC 63 71 93 63 CC D1 3C 54 90 00

La estructura de las APDU puede describirse como sigue:

Comando					
CLA	10		Encadenamiento de comandos		
INS	86		GENERAL AUTHENTICATE		
P1/P2	00 00		Claves y protocolo implícitamente conocidos		
Lc	45		Longitud de los datos		
Datos	Rótulo	Longitud	Valor	Comentario	
	7C	43	-	Datos de autenticación dinámica	
	81	41		Datos de correspondencia	
			04		Punto no comprimido
			7A CF 3E FC 98 2E ... C2 CC 5E		Coordenada x
			54 45 52 DC B6 72 ... C4 92 2D		Coordenada y
Le	00		La longitud de bytes máxima prevista del campo de datos de respuesta es 256		
Respuesta					
Datos	Rótulo	Longitud	Valor	Comentario	
	7C	43		Datos de autenticación dinámica	
	82	41		Datos de correspondencia	
			04		Punto no comprimido
			82 4F BA 91 C9 CB ... BA 3F 57		Coordenada x
			30 D8 C8 79 AA A9 ... D1 3C 54		Coordenada y
Bytes de estado	90 00		Procesamiento normal		

Ejecución del acuerdo de claves

En la tercera etapa, la microplaqueta y el terminal ejecutan un acuerdo de claves ECDH anónimo utilizando los nuevos parámetros de dominio determinados por el generador de grupo efímero de la etapa previa. Solamente se requiere la coordenada x como secreto compartido dado que KDF utiliza solo la primera coordenada para obtener las claves de sesión.

Clave privada del terminal	A73FB703 AC1436A1 8E0CFA5A BB3F7BEC 7A070E7A 6788486B EE230C4A 22762595
Clave pública del terminal	2DB7A64C 0355044E C9DF1905 14C625CB A2CEA487 54887122 F3A5EF0D 5EDD301C, 3556F3B3 B186DF10 B857B58F 6A7EB80F 20BA5DC7 BE1D43D9 BF850149 FBB36462
Clave privada de la microplaqueta	107CF586 96EF6155 053340FD 633392BA 81909DF7 B9706F22 6F32086C 7AFF974A
Clave pública de la microplaqueta	9E880F84 2905B8B3 181F7AF7 CAA9F0EF B743847F 44A306D2 D28C1D9E C65DF6DB, 7764B222 77A2EDDC 3C265A9F 018F9CB8 52E111B7 68B32690 4B59A019 3776F094
Secreto compartido	28768D20 701247DA E81804C9 E780EDE5 82A9996D B4A31502 0B273319 7DB84925

El acuerdo de claves se ejecuta como sigue:

T>C :	10 86 00 00 45 7C 43 83 41 04 2D B7 A6 4C 03 55 04 4E C9 DF 19 05 14 C6 25 CB A2 CE A4 87 54 88 71 22 F3 A5 EF 0D 5E DD 30 1C 35 56 F3 B3 B1 86 DF 10 B8 57 B5 8F 6A 7E B8 0F 20 BA 5D C7 BE 1D 43 D9 BF 85 01 49 FB B3 64 62 00
C>T :	7C 43 84 41 04 9E 88 0F 84 29 05 B8 B3 18 1F 7A F7 CA A9 F0 EF B7 43 84 7F 44 A3 06 D2 D2 8C 1D 9E C6 5D F6 DB 77 64 B2 22 77 A2 ED DC 3C 26 5A 9F 01 8F 9C B8 52 E1 11 B7 68 B3 26 90 4B 59 A0 19 37 76 F0 94 90 00

La codificación del acuerdo de claves se examina en la tabla siguiente:

Comando				
CLA	10		Encadenamiento de órdenes	
INS	86		GENERAL AUTHENTICATE	
P1/P2	00 00		Claves y protocolo implícitamente conocidos	
Lc	45		Longitud de los datos	
Datos	Rótulo	Longitud	Valor	Comentario
	7C	43	-	Datos de autenticación dinámica
	83	41		Clave pública efímera del terminal

			04		Punto no comprimido
			2D B7 A6 4C 03 55 ... DD 30 1C		Coordenada x
			35 56 F3 B3 B1 86 ... B3 64 62		Coordenada y
Le	00		La longitud de bytes máxima prevista del campo de datos de respuesta es 256		
Respuesta					
Datos	Rótulo	Longitud	Valor	Comentario	
	7C	43		Datos de autenticación dinámica	
	84	41		Clave pública efímera de la microplaqueta	
			04		Punto no comprimido
			9E 88 0F 84 29 05 ... 5D F6 DB		Coordenada x
			77 64 B2 22 77 A2 ... 76 F0 94		Coordenada y
Bytes de estado	90 00		Procesamiento normal		

Por medio de KDF, las claves de sesión AES 128 KS_{Enc} y KS_{MAC} se obtienen del secreto compartido y son:

KS_{Enc}	F5F0E35C 0D7161EE 6724EE51 3A0D9A7F
KS_{MAC}	FE251C78 58B356B2 4514B3BD 5F4297D1

Autenticación mutua

Los testigos de autenticación se obtienen mediante KS_{MAC} utilizando

Datos de entrada para T_{IFD}	7F494F06 0A04007F 00070202 04020286 41049E88 0F842905 B8B3181F 7AF7CAA9 F0EFB743 847F44A3 06D2D28C 1D9EC65D F6DB7764 B22277A2 EDDC3C26 5A9F018F 9CB852E1 11B768B3 26904B59 A0193776 F094
Datos de entrada para T_{IC}	7F494F06 0A04007F 00070202 04020286 41042DB7 A64C0355 044EC9DF 190514C6 25CBA2CE A4875488 7122F3A5 EF0D5EDD 301C3556 F3B3B186 DF10B857 B58F6A7E B80F20BA 5DC7BE1D 43D9BF85 0149FBB3 6462

como entrada. La codificación de los datos de entrada figura a continuación:

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
7F49	4F		CLAVE PÚBLICA	Datos de entrada para T_{IFD}
06	0A	04 00 7F 00 07 02 02 04 02 02	IDENTIFICADOR DE OBJETO	PACE con ECDH, correspondencia genérica y clave de sesión AES 128
86	41		PUNTO DE CURVA ELÍPTICA	Punto público efímero de la microplaqueta
		04		Punto no comprimido
		9E 88 0F 84 29 ... 5D F6 DB		Coordenada x
		77 64 B2 22 77 ... 76 F0 94		Coordenada y

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
7F49	4F		CLAVE PÚBLICA	Datos de entrada para T_{IC}
06	0A	04 00 7F 00 07 02 02 04 02 02	IDENTIFICADOR DE OBJETO	PACE con ECDH, correspondencia genérica y claves de sesión AES 128
86	41		PUNTO DE CURVA ELÍPTICA	Punto público efímero del terminal
		04		Punto no comprimido
		2D B7 A6 4C 03 ... DD 30 1C		Coordenada x
		35 56 F3 B3 B1 ... B3 64 62		Coordenada y

Los testigos de autenticación calculados son:

T_{IFD}	C2B0BD78 D94BA866
T_{IC}	3ABB9674 BCE93C08

Finalmente, estos testigos se intercambian y verifican.

T>C :	00 86 00 00 0C 7C 0A 85 08 C2 B0 BD 78 D9 4B A8 66 00
C>T :	7C 0A 86 08 3A BB 96 74 BC E9 3C 08 90 00

G.2 EJEMPLO BASADO EN DH

El segundo ejemplo se basa en DH utilizando el grupo MODP de 1024 bits con subgrupo de orden primo de 160 bits especificado por [RFC 5114]. Los parámetros del grupo son:

Primo *p*	B10B8F96 A080E01D DE92DE5E AE5D54EC 52C99FBC FB06A3C6 9A6A9DCA 52D23B61 6073E286 75A23D18 9838EF1E 2EE652C0 13ECB4AE A9061123 24975C3C D49B83BF ACCBDD7D 90C4BD70 98488E9C 219A7372 4EFFD6FA E5644738 FAA31A4F F55BCCC0 A151AF5F 0DC8B4BD 45BF37DF 365C1A65 E68CFDA7 6D4DA708 DF1FB2BC 2E4A4371
Generador de subgrupo *g*	A4D1CBD5 C3FD3412 6765A442 EFB99905 F8104DD2 58AC507F D6406CFF 14266D31 266FEA1E 5C41564B 777E690F 5504F213 160217B4 B01B886A 5E91547F 9E2749F4 D7FBD7D3 B9A92EE1 909D0D22 63F80A76 A6A24C08 7A091F53 1DBF0A01 69B6A28A D662A4D1 8E73AFA3 2D779D59 18D08BC8 858F4DCE F97C2A24 855E6EEB 22B3B2E5
Orden primo *q* o *g*	F518AA87 81A8DF27 8ABA4E7D 64B7CB9D 49462353

La primera sección introduce la `PACEInfo`. Posteriormente, las APDU intercambiadas incluyendo todos los nonces generados y claves efímeras se enumeran y examinan.

Parámetros Diffie Hellman

La información pertinente para PACE está dada por la estructura de datos `PACEInfo`.

PACEInfo	3012060A 04007F00 07020204 01020201 02020100

La estructura detallada de `PACEInfo` es:

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
30	12		SECUENCIA	PACEInfo
06	0A	04 00 7F 00 07 02 02 04 01 02	IDENTIFICADOR DE OBJETO	OID: PACE con DH, correspondencia genérica y clave de sesión AES 128
02	01	02	ENTERO	Versión 2
02	01	00	ENTERO	Grupo normalizado de 1024-bits especificado por RFC 5114

Flujo de aplicación del ejemplo basado en DH

Para inicializar PACE, el terminal envía el comando MSE:AT a la microplaqueta.

T>C :	00 22 C1 A4 0F 80 0A 04 00 7F 00 07 02 02 04 01 02 83 01 01
C>T :	90 00

La codificación del comando se describe en la tabla siguiente.

Comando				
CLA	00		Llano	
INS	22		Gestionar entorno de seguridad	
P1/P2	C1 A4		Establecer plantillas de autenticación para autenticación mutua	
Lc	0F		Longitud de campo de datos	
Datos	Rótulo	Longitud	Valor	Comentario
	80	0A	04 00 7F 00 07 02 02 04 01 02	OID: mecanismo criptográfico: PACE con DH, correspondencia genérica y AES128
	83	01	01	Contraseña: MRZ (ZLM)
Respuesta				
Bytes de estado	90 00		Procesamiento normal	

Nonce cifrado

A continuación, el terminal interroga un nonce de microplaqueta.

Nonce s descifrado	FA5B7E3E 49753A0D B9178B7B 9BD898C8
Nonce z cifrado	854D8DF5 827FA685 2D1A4FA7 01CDDDCA

La comunicación tiene el aspecto siguiente.

T>C :	10 86 00 00 02 7C 00 00
C>T :	7C 12 80 10 85 4D 8D F5 82 7F A6 85 2D 1A 4F A7 01 CD DD CA 90 00

La codificación del comando APDU y la correspondiente respuesta se describen en la tabla siguiente.

Comando				
CLA	10		Encadenamiento de órdenes	
INS	86		GENERAL AUTHENTICATE	
P1/P2	00 00		Claves y protocolo implícitamente conocidos	
Lc	02		Longitud de los datos	
Datos	Rótulo	Longitud	Valor	Comentario
	7C	00	-	Ausente
Le	00		La longitud de byte máxima prevista del campo de datos de respuesta es 256	
Respuesta				
Datos	Rótulo	Longitud	Valor	Comentario
	7C	12		Datos de autenticación dinámica
	80	10	85 4D 8D F5 82 7F A6 85 2D 1A 4F A7 01 CD DD CA	Nonce cifrado
Bytes de estado	90 00		Procesamiento normal	

Correspondencia de nonce

Mediante la correspondencia genérica, se hace corresponder el nonce con un generador de grupo efímero. Para ese fin, las claves efímeras siguientes se generan de forma aleatoria por el terminal y la microplaqueta.

Clave privada del terminal	5265030F 751F4AD1 8B08AC56 5FC7AC95 2E41618D
Clave pública del terminal	23FB3749 EA030D2A 25B278D2 A562047A DE3F01B7 4F17A154 02CB7352 CA7D2B3E B71C343D B13D1DEB CE9A3666 DBCFC920 B49174A6 02CB4796 5CAA73DC 702489A4 4D41DB91 4DE9613D C5E98C94 160551C0 DF86274B 9359BC04 90D01B03 AD54022D CB4F57FA D6322497 D7A1E28D 46710F46 1AFE710F BBBC5F8B A166F431 1975EC6C
Clave privada de la microplaqueta	66DDAFEA C1609CB5 B963BB0C B3FF8B3E 047F336C
Clave pública de la microplaqueta	78879F57 225AA808 0D52ED0F C890A4B2 5336F699 AA89A2D3 A189654A F70729E6 23EA5738 B26381E4 DA19E004 706FACE7 B235C2DB F2F38748 312F3C98 C2DD4882 A41947B3 24AA1259 AC22579D B93F7085 655AF308 89DBB845 D9E6783F E42C9F24 49400306 254C8AE8 EE9DD812 A804C0B6 6E8CAFC1 4F84D825 8950A91B 44126EE6
Secreto compartido H	5BABEBEF 5B74E5BA 94B5C063 FDA15F1F 1CDE9487 3EE0A5D3 A2FCAB49 F258D07F 544F13CB 66658C3A FEE9E727 389BE3F6 CBBBD321 28A8C21D D6EEA3CF 7091CDDF B08B8D00 7D40318D CCA4FFBF 51208790 FB4BD111 E5A968ED 6B6F08B2 6CA87C41 0B3CE0C3 10CE104E ABD16629 AA48620C 1279270C B0750C0D 37C57FFF E302AE7F
Generador de correspondencia $\hat{G}$	7C9CBFE9 8F9FBDDA 8D143506 FA7D9306 F4CB17E3 C71707AF F5E1C1A1 23702496 84D64EE3 7AF44B8D BD9D45BF 6023919C BAA027AB 97ACC771 666C8E98 FF483301 BFA4872D EDE9034E DFACB708 14166B7F 36067682 9B826BEA 57291B5A D69FBC84 EF1E7790 32A30580 3F743417 93E86974 2D401325 B37EE856 5FFCDEE6 18342DC5

Las APDU siguientes se intercambian entre el terminal y la microplaqueta para hacer la correspondencia con nonce.

T>C :	10 86 00 00 86 7C 81 83 81 81 80 23 FB 37 49 EA 03 0D 2A 25 B2 78 D2 A5 62 04 7A DE 3F 01 B7 4F 17 A1 54 02 CB 73 52 CA 7D 2B 3E B7 1C 34 3D B1 3D 1D EB CE 9A 36 66 DB CF C9 20 B4 91 74 A6 02 CB 47 96 5C AA 73 DC 70 24 89 A4 4D 41 DB 91 4D E9 61 3D C5 E9 8C 94 16 05 51 C0 DF 86 27 4B 93 59 BC 04 90 D0 1B 03 AD 54 02 2D CB 4F 57 FA D6 32 24 97 D7 A1 E2 8D 46 71 0F 46 1A FE 71 0F BB BC 5F 8B A1 66 F4 31 19 75 EC 6C 00
C>T :	7C 81 83 82 81 80 78 87 9F 57 22 5A A8 08 0D 52 ED 0F C8 90 A4 B2 53 36 F6 99 AA 89 A2 D3 A1 89 65 4A F7 07 29 E6 23 EA 57 38 B2 63 81 E4 DA 1 9E0 04 70 6F AC E7 B2 35 C2 DB F2 F3 87 48 31 2F 3C 98 C2 DD 48 82 A4 19 47 B3 24 AA 12 59 AC 22 57 9D B9 3F 70 85 65 5A F3 08 89 DB B8 45 D9 E6 78 3F E4 2C 9F 24 49 40 03 06 25 4C 8A E8 EE 9D D8 12 A8 04 C0 B6 6E 8C AF C1 4F 84 D8 25 89 50 A9 1B 44 12 6E E6 90 00

La estructura de los APDU puede describirse como sigue:

Comando				
CLA	10		Encadenamiento de órdenes	
INS	86		GENERAL AUTHENTICATE	
P1/P2	00 00		Claves y protocolo implícitamente conocidos	
Lc	86		Longitud de los datos	
Datos	Rótulo	Longitud	Valor	Comentario
	7C	81 83	-	Datos de autenticación dinámica
	81	81 80	23 FB 37 49 EA 03 ... 75 EC 6C	Datos de correspondencia
Le	00		La longitud de bytes máxima prevista del campo de datos de respuesta es 256	
Respuesta				
Datos	Rótulo	Longitud	Valor	Comentario
	7C	81 83		Datos de autenticación dinámica
	82	81 80	ED 0F C8 90 A4 B2 ... 12 6E E6	Datos de correspondencia
Bytes de estado	90 00		Procesamiento normal	

Ejecución del acuerdo de claves

Posteriormente, la microplaqueta y el terminal ejecutan un acuerdo de claves DH anónimo utilizando los nuevos parámetros de dominio determinados por el generador del grupo efímero de la etapa previa.

Clave privada del terminal	89CCD99B 0E8D3B1F 11E1296D CA68EC53 411CF2CA
Clave pública del terminal	00907D89 E2D425A1 78AA81AF 4A7774EC 8E388C11 5CAE6703 1E85EECE 520BD911 551B9AE4 D04369F2 9A02626C 86FBC674 7CC7BC35 2645B616 1A2A42D4 4EDA80A0 8FA8D61B 76D3A154 AD8A5A51 786B0BC0 71470578 71A92221 2C5F67F4 31731722 36B7747D 1671E6D6 92A3C7D4 0A0C3C5C E397545D 015C175E B5130551 EDBC2EE5 D4
Clave privada de la microplaqueta	A5B78012 6B7C980E 9FCEA1D4 539DA1D2 7C342DFA
Clave pública de la microplaqueta	075693D9 AE941877 573E634B 6E644F8E 60AF17A0 076B8B12 3D920107 4D36152B D8B3A213 F53820C4 2ADC79AB 5D0AEEC3 AEFB9139 4DA476BD 97B9B14D 0A65C1FC 71A0E019 CB08AF55 E1F72900 5FBA7E3F A5DC4189 9238A250 767A6D46 DB974064 386CD456 743585F8 E5D90CC8 B4004B1F 6D866C79 CE0584E4 9687FF61 BC29AEA1
Secreto compartido	6BABC7B3 A72BCD7E A385E4C6 2DB2625B D8613B24 149E146A 629311C4 CA6698E3 8B834B6A 9E9CD718 4BA8834A FF5043D4 36950C4C 1E783236 7C10CB8C 314D40E5 990B0DF7 013E64B4 549E2270 923D06F0 8CFF6BD3 E977DDE6 ABE4C31D 55C0FA2E 465E553E 77BDF75E 3193D383 4FC26E8E B1EE2FA1 E4FC97C1 8C3F6CFF FE2607FD

El acuerdo de claves se ejecuta como sigue:

T>C :	10 86 00 00 86 7C 81 83 83 81 80 90 7D 89 E2 D4 25 A1 78 AA 81 AF 4A 77 74 EC 8E 38 8C 11 5C AE 67 03 1E 85 EE CE 52 0B D9 11 55 1B 9A E4 D0 43 69 F2 9A 02 62 6C 86 FB C6 74 7C C7 BC 35 26 45 B6 16 1A 2A 42 D4 4E DA 80 A0 8F A8 D6 1B 76 D3 A1 54 AD 8A 5A 51 78 6B 0B C0 71 47 05 78 71 A9 22 21 2C 5F 67 F4 31 73 17 22 36 B7 74 7D 16 71 E6 D6 92 A3 C7 D4 0A 0C 3C 5C E3 97 54 5D 01 5C 17 5E B5 13 05 51 ED BC 2E E5 D4 00
C>T :	7C 81 83 84 81 80 07 56 93 D9 AE 94 18 77 57 3E 63 4B 6E 64 4F 8E 60 AF 17 A0 07 6B 8B 12 3D 92 01 07 4D 36 15 2B D8 B3 A2 13 F5 38 20 C4 2A DC 79 AB 5D 0A EE C3 AE FB 91 39 4D A4 76 BD 97 B9 B1 4D 0A 65 C1 FC 71 A0 E0 19 CB 08 AF 55 E1 F7 29 00 5F BA 7E 3F A5 DC 41 89 92 38 A2 50 76 7A 6D 46 DB 97 40 64 38 6C D4 56 74 35 85 F8 E5 D9 0C C8 B4 00 4B 1F 6D 86 6C 79 CE 05 84 E4 96 87 FF 61 BC 29 AE A1 90 00

Comando				
CLA	10		Encadenamiento de órdenes	
INS	86		GENERAL AUTHENTICATE	
P1/P2	00 00		Claves y protocolo implícitamente conocidos	
Lc	86		Longitud de los datos	
Datos	Rótulo	Longitud	Valor	Comentario
	7C	81 83	-	Datos de autenticación dinámica
	83	81 80	90 7D 89 E2 D4 25 ... 2E E5 D4	Clave pública efímera del terminal
Le	00		La longitud de bytes máxima prevista del campo de datos de respuesta es 256	
Respuesta				
Datos	Rótulo	Longitud	Valor	Comentario
	7C	81 83		Datos de autenticación dinámica
	84	81 80	07 56 93 D9 AE 94 ... 29 AE A1	Clave pública efímera de la microplaqueta
Bytes de estado	90 00		Procesamiento normal	

Las claves de sesión AES 128 KS_{Enc} y KS_{MAC} se obtienen del secreto compartido utilizando el KDF.

KS_{Enc}	2F7F46AD CC9E7E52 1B45D192 FAFA9126
KS_{MAC}	805A1D27 D45A5116 F73C5446 9462B7D8

Autenticación mutua

Los testigos de autenticación se construyen a partir de los siguientes datos de entrada.

Datos de entrada para T_{IFD}	7F49818F 060A0400 7F000702 02040102 84818007 5693D9AE 94187757 3E634B6E 644F8E60 AF17A007 6B8B123D 9201074D 36152BD8 B3A213F5 3820C42A DC79AB5D 0AEEC3AE FB91394D A476BD97 B9B14D0A 65C1FC71 A0E019CB 08AF55E1 F729005F BA7E3FA5 DC418992 38A25076 7A6D46DB 97406438 6CD45674 3585F8E5 D90CC8B4 004B1F6D 866C79CE 0584E496 87FF61BC 29AEA1
Datos de entrada para T_{IC}	7F49818F 060A0400 7F000702 02040102 84818090 7D89E2D4 25A178AA 81AF4A77 74EC8E38 8C115CAE 67031E85 EECE520B D911551B 9AE4D043 69F29A02 626C86FB C6747CC7 BC352645 B6161A2A 42D44EDA 80A08FA8 D61B76D3 A154AD8A 5A51786B 0BC07147 057871A9 22212C5F 67F43173 172236B7 747D1671 E6D692A3 C7D40A0C 3C5CE397 545D015C 175EB513 0551EDBC 2EE5D4

La codificación de los datos de entrada se indica a continuación:

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
7F49	81 8F		CLAVE PÚBLICA	Datos de entrada para T_{IFD}
06	0A	04 00 7F 00 07 02 02 04 01 02	IDENTIFICADOR DE OBJETO	PACE con DH, correspondencia genérica y claves de sesión AES 128
84	81 80	07 56 93 D9 AE ... 29 AE A1	ENTERO SIN SIGNO	Clave pública efímera de la microplaqueta

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
7F49	81 8F		CLAVE PÚBLICA	Datos de entrada para T_{IC}
06	0A	04 00 7F 00 07 02 02 04 01 02	IDENTIFICADOR DE OBJETO	PACE con DH, correspondencia genérica y claves de sesión AES 128
84	81 80	90 7D 89 E2 D4 ... 2E E5 D4	ENTERO SIN SIGNO	Clave pública efímera del terminal

Los testigos de autenticación calculados son:

T_{IFD}	B46DD9BD 4D98381F
T_{IC}	917F37B5 C0E6D8D1

Finalmente, estos testigos se intercambian y verifican.

T>C :	00 86 00 00 0C 7C 0A 85 08 B4 6D D9 BD 4D 98 38 1F 00
C>T :	7C 1B 86 08 91 7F 37 B5 C0 E6 D8 D1 87 0F 44 45 54 45 53 54 43 56 43 41 30 30 30 30 33

Comando				
CLA	00		Llano	
INS	86		GENERAL AUTHENTICATE	
P1/P2	00 00		Claves y protocolo implícitamente conocidos	
Lc	0C		Longitud de los datos	
Datos	Rótulo	Longitud	Valor	Comentario
	7C	0A	-	Datos de autenticación dinámica
	85	08	B4 6D D9 BD 4D 98 38 1F	Testigo de autenticación del terminal
Le	00		La longitud de bytes máxima prevista del campo de datos de respuesta es 256	
Respuesta				
Datos	Rótulo	Longitud	Valor	Comentario
	7C	0A		Datos de autenticación dinámica
	86	08	91 7F 37 B5 C0 E6 D8 D1	Testigo de autenticación de la microplaqueta
Bytes de estado	90 00		Procesamiento normal	

— — — — — — — —

Apéndice H de la Parte 11

EJEMPLO ELABORADO: PACE – CORRESPONDENCIA INTEGRADA (INFORMATIVO)

En el presente apéndice se proporcionan dos ejemplos para el protocolo PACE con correspondencia integrada. El primero se basa en la curva elíptica Diffie-Hellman (ECDH) y el segundo en Diffie-Hellman (DH). Se utiliza la clave *K* obtenida de la ZLM en el ejemplo anterior.

H.1 EJEMPLO BASADO EN ECDH

Este ejemplo se basa en la curva elíptica BrainpoolP256r1. La cifra de bloque utilizada en este ejemplo es AES-128. Cabe recordar que los parámetros de la curva son los siguientes:

Primo p	A9FB57DB A1EEA9BC 3E660A90 9D838D72 6E3BF623 D5262028 2013481D 1F6E5377
Parámetro a	7D5A0975 FC2C3057 EEF67530 417AFFE7 FB8055C1 26DC5C6C E94A4B44 F330B5D9
Parámetro b	26DC5C6C E94A4B44 F330B5D9 BBD77CBF 95841629 5CF7E1CE 6BCCDC18 FF8C07B6
Coordenada x del generador de grupo G	8BD2AEB9 CB7E57CB 2C4B482F FC81B7AF B9DE27E1 E3BD23C2 3A4453BD 9ACE3262
Coordenada y del generador de grupo G	547EF835 C3DAC4FD 97F8461A 14611DC9 C2774513 2DED8E54 5C1D54C7 2F046997
Orden del grupo n	A9FB57DB A1EEA9BC 3E660A90 9D838D71 8C397AA3 B561A6F7 901E0E82 974856A7
Cofactor f	01

La clave de cifrado es la siguiente:

K_π	591468CD A83D6521 9CCCB856 0233600F

Nonce cifrado

La microplaqueta elige en forma aleatoria un nonce s que se cifra utilizado K_{IT}. Luego se envía al terminal el nonce cifrado z.

Nonce de descifrado s	2923BE84 E16CD6AE 529049F1 F1BBE9EB
Nonce cifrado z	143DC40C 08C8E891 FBED7DED B92B64AD

Nonce de correspondencia

Un nonce t se elige en forma aleatoria y se envía en claro, luego se usan t y s para calcular la correspondencia integrada. Primero, s y t se aplican a la función pseudoaleatoria R_p, obtenida de AES. Luego, la codificación de punto f_G se utiliza sobre resultado para calcular el generador de correspondencia $\hat{G}=f_G(R_p(s,t))$.

Nonce t	5DD4CBFC 96F5453B 130D890A 1CDBAE32
R (s,t) pseudoaleatorio	E4447E2D FB3586BA C05DDB00 156B57FB B2179A39 49294C97 25418980 0C517BAA 8DA0FF39 7ED8C445 D3E421E4 FEB57322
$R_p(s,t)$	A2F8FF2D F50E52C6 599F386A DCB595D2 29F6A167 ADE2BE5F 2C3296AD D5B7430E
Coordenada x del generador de correspondencia $\hat{G}$	8E82D315 59ED0FDE 92A4D049 8ADD3C23 BABA94FB 77691E31 E90AEA77 FB17D427
Coordenada y del generador de correspondencia $\hat{G}$	4C1AE14B D0C3DBAC 0C871B7F 36081693 64437CA3 0AC243A0 89D3F266 C1E60FAD

Ejecución del acuerdo de clave

La microplaqueta y el terminal ejecutan un acuerdo de clave Diffie-Hellman anónimo utilizando sus claves secretas y el generador de correspondencia $\hat{G}$. El secreto compartido K es la coordenada x del acuerdo.

Clave privada de la microplaqueta SK_{IC}	107CF586 96EF6155 053340FD 633392BA 81909DF7 B9706F22 6F32086C 7AFF974A
Clave pública de la microplaqueta PK_{IC}	67F78E5F 7F768608 2B293E8D 087E0569 16D0F74B C01A5F89 57D0DE45 691E51E8 932B69A9 62B52A09 85AD2C0A 271EE6A1 3A8ADDDC D1A3A994 B9DED257 F4D22753
Clave privada del terminal SK_{IFD}	A73FB703 AC1436A1 8E0CFA5A BB3F7BEC 7A070E7A 6788486B EE230C4A 22762595
Clave pública del terminal PK_{IFD}	89CBA23F FE96AA18 D824627C 3E934E54 A9FD0B87 A95D1471 DC1C0ABF DCD640D4 6755DE9B 7B778280 B6BEBD57 439ADFEB 0E21FD4E D6DF4257 8C13418A 59B34C37
Secreto compartido K	4F150FDE 1D4F0E38 E95017B8 91BAE171 33A0DF45 B0D3E18B 60BA7BEA FDC2C713

Aplicando las especificaciones de [1], las claves de sesión K_{Enc} y K_{MAC} se obtienen de K utilizando la función de condensación SHA-1: K_{Enc}=SHA-1(K||0x00000001) y K_{MAC}=SHA-1(K||0x00000002). Luego, solo se utilizan los primeros 16 octetos del resumen con el resultado siguiente:

K_{Enc}	0D3FEB33 251A6370 893D62AE 8DAAF51B
K_{MAC}	B01E89E3 D9E8719E 586B50B4 A7506E0B

Autenticación mutua

Los testigos de autenticación se calculan utilizando un CMAC sobre las entradas siguientes con la clave K_{MAC}.

Datos de entrada para T_{IC}	7F494F06 0A04007F 00070202 04040286 410489CB A23FFE96 AA18D824 627C3E93 4E54A9FD 0B87A95D 1471DC1C 0ABFDCD6 40D46755 DE9B7B77 8280B6BE BD57439A DFEB0E21 FD4ED6DF 42578C13 418A59B3 4C37
Datos de entrada para T_{IFD}	7F494F06 0A04007F 00070202 04040286 410467F7 8E5F7F76 86082B29 3E8D087E 056916D0 F74BC01A 5F8957D0 DE45691E 51E8932B 69A962B5 2A0985AD 2C0A271E E6A13A8A DDDCD1A3 A994B9DE D257F4D2 2753

Los testigos de autenticación correspondientes son:

T_{IC}	75D4D96E 8D5B0308
T_{IFD}	450F02B8 6F6A0909

H.2 EJEMPLO BASADO EN DH

Este ejemplo se basa en el grupo MODP de 1 024 bits con subgrupo de orden primo de 160-bits. La cifra de bloque utilizada en este ejemplo es AES-128.

Los parámetros de grupo son:

Primo p	B10B8F96 A080E01D DE92DE5E AE5D54EC 52C99FBC FB06A3C6 9A6A9DCA 52D23B61 6073E286 75A23D18 9838EF1E 2EE652C0 13ECB4AE A9061123 24975C3C D49B83BF ACCBDD7D 90C4BD70 98488E9C 219A7372 4EFFD6FA E5644738 FAA31A4F F55BCCC0 A151AF5F 0DC8B4BD 45BF37DF 365C1A65 E68CFDA7 6D4DA708 DF1FB2BC 2E4A4371
Generador de subgrupo g	A4D1CBD5 C3FD3412 6765A442 EFB99905 F8104DD2 58AC507F D6406CFF 14266D31 266FEA1E 5C41564B 777E690F 5504F213 160217B4 B01B886A 5E91547F 9E2749F4 D7FBD7D3 B9A92EE1 909D0D22 63F80A76 A6A24C08 7A091F53 1DBF0A01 69B6A28A D662A4D1 8E73AFA3 2D779D59 18D08BC8 858F4DCE F97C2A24 855E6EEB 22B3B2E5
Orden primo q o g	F518AA87 81A8DF27 8ABA4E7D 64B7CB9D 49462353

Se utiliza la siguiente clave de cifrado:

K_π	591468CD A83D6521 9CCCB856 0233600F

Nonce cifrado

La microplaqueta elige en forma aleatoria un nonce s que se cifra utilizando K_π. Luego, el nonce cifrado z se envía al terminal.

Nonce descifrado s	FA5B7E3E 49753A0D B9178B7B 9BD898C8
Nonce cifrado z	9ABB8864 CA0FF155 1E620D1E F4E13510

Nonce de correspondencia

Un nonce t se elige en forma aleatoria y se envía en claro. Luego, t y s se utilizan para calcular la correspondencia integrada. Primero, s y t se aplican a la función pseudoaleatoria R_p, obtenida de AES. Después, se utiliza la codificación del punto f_g sobre el resultado.

Nonce t	B3A6DB3C 870C3E99 245E0D1C 06B747DE
R(s,t) pseudoaleatorio	EAB98D13 E0905295 2AA72990 7C3C9461 84DEA0FE 74AD2B3A F506F0A8 3018459C 38099CD1 F7FF4EA0 A078DB1F AC136550 5E3DC855 00EF95E2 0B4EEF2E 88489233 BEE0546B 472F994B 618D1687 02406791 DEEF3CB4 810932EC 278F3533 FDB860EB 4835C36F A4F1BF3F A0B828A7 18C96BDE 88FBA38A 3E6C35AA A1095925 1EB5FC71 0FC18725 8995944C 0F926E24 9373F485
$R_p(s,t)$	A0C7C50C 002061A5 1CC87D25 4EF38068 607417B6 EE1B3647 3CFB800D 2D2E5FA2 B6980F01 105D24FA B22ACD1B FA5C8A4C 093ECDFA FE6D7125 D42A843E 33860383 5CF19AFA FF75EFE2 1DC5F6AA 1F9AE46C 25087E73 68166FB0 8C1E4627 AFED7D93 570417B7 90FF7F74 7E57F432 B04E1236 819E0DFE F5B6E77C A4999925 328182D2
Generador de correspondencia $\hat{g}= f_g(R_p(s,t))$	1D7D767F 11E333BC D6DBAEF4 0E799E7A 926B9697 3550656F F3C83072 6D118D61 C276CDCC 61D475CF 03A98E0C 0E79CAEB A5BE2557 8BD4551D 0B109032 36F0B0F9 76852FA7 8EEA14EA 0ACA87D1 E91F688F E0DFF897 BBE35A47 2621D343 564B262F 34223AE8 FC59B664 BFEDFA2B FE7516CA 5510A6BB B633D517 EC25D4E0 BBAA16C2

Ejecución del acuerdo de claves

La microplaqueta y el terminal ejecutan un acuerdo de claves Diffie-Hellman anónimo utilizando sus claves secretas y el generador de correspondencia $\hat{g}$.

Clave privada de la microplaqueta SK_{IC}	020F018C 7284B047 FA7721A3 37EFB7AC B1440BB3 0C5252BD 41C97C30 C994BB78 E9F0C5B3 2744D840 17D21FFA 6878396A 6469CA28 3EF5C000 DAF7D261 A39AB886 0ED4610A B5343390 897AAB5A 7787E4FA EFA0649C 6A94FDF8 2D991E8E 3FC332F5 142729E7 040A3F7D 5A4D3CD7 5CBEE1F0 43C1CAD2 DD484FEB 4ED22B59 7D36688E

Clave pública de la microplaqueta PK_{IC}	928D9A0F 9DBA450F 13FC859C 6F290D1D 36E42431 138A4378 500BEB4E 0401854C FF111F71 CB6DC1D0 335807A1 1388CC8E AA87B079 07AAD9FB A6B169AF 6D8C26AF 8DDDC39A DC3AD2E3 FF882B84 D23E9768 E95A80E4 746FB07A 9767679F E92133B4 D379935C 771BD7FB ED6C7BB4 B1708B27 5EA75679 524CDC9C 6A91370C C662A2F3
Clave privada del terminal SK_{IFD}	4BD0E547 40F9A028 E6A515BF DAF96784 8C4F5F5F FF65AA09 15947FFD 1A0DF2FA 6981271B C905F355 1457B7E0 3AC3B806 6DE4AA40 6C1171FB 43DD939C 4BA16175 103BA3DE E16419AA 248118F9 0CC36A3D 6F4C3736 52E0C3CC E7F0F1D0 C5425B36 00F0F0D6 A67F004C 8BBA33F2 B4733C72 52445C1D FC4F1107 203F71D2 EFB28161
Clave pública del terminal PK_{IFD}	0F0CC629 45A80292 51FB7EF3 C094E12E C68E4EF0 7F27CB9D 9CD04C5C 4250FAE0 E4F8A951 557E929A EB48E5C6 DD47F2F5 CD7C351A 9BD2CD72 2C07EDE1 66770F08 FFCB3702 62CF308D D7B07F2E 0DA9CAAA 1492344C 85290691 9538C98A 4BA4187E 76CE9D87 832386D3 19CE2E04 3C3343AE AE6EDBA1 A9894DC5 094D22F7 FE1351D5
Secreto compartido K	419410D6 C0A17A4C 07C54872 CE1CBCEB 0A2705C1 A434C8A8 9A4CFE41 F1D78124 CA7EC52B DE7615E5 345E48AB 1ABB6E7D 1D59A57F 3174084D 3CA45703 97C1F622 28BDFDB2 DA191EA2 239E2C06 0DBE3BBC 23C2FCD0 AF12E0F9 E0B99FCF 91FF1959 011D5798 B2FCBC1F 14FCC24E 441F4C8F 9B08D977 E9498560 E63E7FFA B3134EA7

Las claves de sesión K_{Enc} y K_{MAC} se obtienen de K utilizando la función de condensación SHA-1: K_{Enc}=SHA-1(K||0x00000001) y K_{MAC}=SHA-1(K||0x00000002). Después, solo se utilizan los primeros 16 octetos del resumen con el resultado siguiente:

K_{Enc}	01AFC10C F87BE36D 8179E873 70171F07
K_{MAC}	23F0FBD0 5FD6C7B8 B88F4C83 09669061

Autenticación mutua

Los testigos de autenticación se calculan utilizando una CMAC en las siguientes entradas con la clave K_{MAC}.

Datos de entrada para T_{IC}	7F49818F 060A0400 7F000702 02040302 8481800F 0CC62945 A8029251 FB7EF3C0 94E12EC6 8E4EF07F 27CB9D9C D04C5C42 50FAE0E4 F8A95155 7E929AEB 48E5C6DD 47F2F5CD 7C351A9B D2CD722C 07EDE166 770F08FF CB370262 CF308DD7 B07F2E0D A9CAAA14 92344C85 29069195 38C98A4B A4187E76 CE9D8783 2386D319 CE2E043C 3343AEAE 6EDBA1A9 894DC509 4D22F7FE 1351D5
Datos de entrada para T_{IFD}	7F49818F 060A0400 7F000702 02040302 84818092 8D9A0F9D BA450F13 FC859C6F 290D1D36 E4243113 8A437850 0BEB4E04 01854CFF 111F71CB 6DC1D033 5807A113 88CC8EAA 87B07907 AAD9FBA6 B169AF6D 8C26AF8D DDC39ADC 3AD2E3FF 882B84D2 3E9768E9 5A80E474 6FB07A97 67679FE9 2133B4D3 79935C77 1BD7FBED 6C7BB4B1 708B275E A7567952 4CDC9C6A 91370CC6 62A2F3

Las correspondientes testigos de autenticación son:

T_{IC}	C2F04230 187E1525
T_{IFD}	55D61977 CBF5307E

— — — — — — — —

Apéndice I de la Parte 11

EJEMPLO ELABORADO: PACE – CORRESPONDENCIA CA (INFORMATIVO)

En el presente apéndice se proporciona un ejemplo elaborado para el protocolo PACE con Correspondencia de autenticación de microplaqueta basada en Diffie-Hellman de curva elíptica (ECDH). Todos los números de las tablas están en notación hexadecimal.

La ZLM se utiliza como contraseña. Los campos de datos pertinentes de la ZLM incluyendo los dígitos de verificación son:

- Número de documento: C11T002JM4;
- Fecha de nacimiento: 9608122;
- Fecha de caducidad: 2310314.

Por consiguiente, la codificación K de la ZLM y la clave de cifrado obtenida K_π son:

K	894D03F1 48C6265E 89845B21 8856EA34 D00EF8E8
K_π	4E6F6FBF 7BE748B9 32C7B741 61BBA9DF

I.1 EJEMPLO BASADO EN ECDH

El ejemplo se basa en ECDH aplicando los parámetros de dominio normalizados BrainpoolP256r1 (véase [RFC 5639]).

La primera sección introduce la correspondiente `PACEInfo`. Posteriormente, se enumeran y examinan las APDU intercambiadas, incluyendo todos los nonces generados y claves efímeras.

Parámetros de curva elíptica

Utilizando parámetros de dominio normalizados, toda la información requerida para ejecutar PACE está dada por la estructura de datos `PACEInfo`. En particular, no se requiere `PACEDomainParameterInfo`.

PACEInfo	3012060A 04007F00 07020204 06020201 0202010D

La estructura detallada de `PACEInfo` se desglosa en la tabla siguiente.

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
30	12		SECUENCIA	PACEInfo
06	0A	04 00 7F 00 07 02 02 04 06 02	IDENTIFICADOR DE OBJETO	PACE con ECDH, correspondencia de autenticación de microplaqueta y claves de sesión AES 128
02	01	02	ENTERO	Versión 2
02	01	0D	ENTERO	Parámetros de dominio normalizados Brainpool P256r1

Por conveniencia, a continuación se presenta la codificación ASN.1 de los parámetros de dominio BrainpoolP256r1.

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
30	81 EC		SECUENCIA	Parámetro de dominio
06	07	2A 86 48 CE 3D 02 01	IDENTIFICADOR DE OBJETO	Algorithm id-ecPublicKey
30	81 E0		SECUENCIA	Parámetro de dominio
02	01	01	ENTERO	Versión
30	2C		SECUENCIA	Campo subyacente
06	07	2A 86 48 CE 3D 01 01	IDENTIFICADOR DE OBJETO	Campo primo
02	21	00 A9 FB 57 DB A1 EE A9 BC 3E 66 0A 90 9D 83 8D 72 6E 3B F6 23 D5 26 20 28 20 13 48 1D 1F 6E 53 77	ENTERO	Primo p
30	44		SECUENCIA	Ecuación de curva
04	20	7D 5A 09 75 FC 2C 30 57 EE F6 75 30 41 7A FF E7 FB 80 55 C1 26 DC 5C 6C E9 4A 4B 44 F3 30 B5 D9	CADENA DE OCTETOS	Parámetro a
04	20	26 DC 5C 6C E9 4A 4B 44 F3 30 B5 D9 BB D7 7C BF 95 84 16 29 5C F7 E1 CE 6B CC DC 18 FF 8C 07 B6	CADENA DE OCTETOS	Parámetro b

04	41		CADENA DE OCTETOS		Grupo generador G	
		04	-			Punto no comprimido
		8B D2 AE B9 CB 7E 57 CB 2C 4B 48 2F FC 81 B7 AF B9 DE 27 E1 E3 BD 23 C2 3A 44 53 BD 9A CE 32 62	-			Coordenada x
		54 7E F8 35 C3 DA C4 FD 97 F8 46 1A 14 61 1D C9 C2 77 45 13 2D ED 8E 54 5C 1D 54 C7 2F 04 69 97	-			Coordenada y
02	21	00 A9 FB 57 DB A1 EE A9 BC 3E 66 0A 90 9D 83 8D 71 8C 39 7A A3 B5 61 A6 F7 90 1E 0E 82 97 48 56 A7	ENTERO		Orden de grupo n	
02	01	01	ENTERO		Cofactor f	

Flujo de aplicación en el ejemplo basado en ECDH

Para inicializar PACE, el terminal envía la orden MSE:Set AT a la microplaqueta.

T>C :	`00 22 C1 A4 0F 80 0A 04 00 7F 00 07 02 02 04 06 02 83 01 01`
C>T :	`90 00`

Aquí, T>C es una abreviatura para la APDU enviada del terminal a la microplaqueta mientras que C>T denota la respuesta correspondiente enviada por la microplaqueta al terminal. La codificación de la orden se explica en la tabla siguiente.

Orden				
CLA	00		Llano	
INS	22		Gestionar entorno de seguridad	
P1/P2	C1 A4		Establecer plantillas de autenticación para autenticación mutua	
Lc	0F		Longitud del campo de datos	
Datos	Rótulo	Longitud	Valor	Comentario
	80	0A	04 00 7F 00 07 02 02 04 02 02	Mecanismo criptográfico: PACE con ECDH, correspondencia de autenticación de microplaqueta y claves de sesiones AES128
	83	01	01	Contraseña: MRZ (ZLM)

Respuesta		
Bytes de estado	90 00	Procesamiento normal

Nonce cifrado

A continuación, la microplaqueta genera en forma aleatoria el nonce s (palabra aleatori de un úico uso) y lo cifra mediante K_{π}.

Nonce s descifrado	658B860B C94DF6F0 44FCE6D5 C82CF8E5
Nonce z cifrado	CB60E8E0 D85B76A9 BD304747 C2AD42E2

El nonce cifrado es interrogado por el terminal.

T>C :	10 86 00 00 02 7C 00 00
C>T :	7C 12 80 10 CB 60 E8 E0 D8 5B 76 A9 BD 30 47 47 C2 AD 42 E2 90 00

La codificación de la orden APDU y la respuesta correspondiente figuran en la tabla siguiente.

Comando				
CLA	10		Encadenamiento de órdenes	
INS	86		GENERAL AUTHENTICATE	
P1/P2	00 00		Claves y protocolo implícitamente conocidos	
Lc	02		Longitud de los datos	
Datos	Rótulo	Longitud	Valor	Comentario
	7C	00	-	Ausente
Le	00		La longitud máxima de bytes prevista del campo de datos de respuesta es 256	
Respuesta				
Datos	Rótulo	Longitud	Valor	Comentario
	7C	12		Datos de autenticación dinámica
	80	10	CB60E8E0 D85B76A9 BD304747 C2AD42E2	Nonce cifrado
Bytes de estado	90 00		Procesamiento normal	

Nonce de correspondencia

El nonce se hace corresponder con un generador de grupo efímero a través de correspondencia genérica. Las claves efímeras requeridas elegidas en forma aleatoria también se recogen en la tabla siguiente.

Clave privada del terminal	5D8BB87B D74D985A 4B7D4325 B9F7B976 FE835122 77340079 8914AA22 738135CC
Clave pública del terminal	7F1D410A DB7DDB3B 84BF1030 800981A9 105D7457 B4A3ADE0 02384F30 86C67EDE 1AB88910 4A27DB6D 842B0190 20FBF3CE ACB0DC62 7F7BDCAC 29969E19 D0E553C1
Clave privada de la microplaqueta	9E56A6B5 9C95D06E CE5CD10F 983BB2F4 F1943528 E577F238 81D89D8C 3BBEE0AA
Clave pública de la microplaqueta	A234236A A9B9621E 8EFB73B5 245C0E09 D2576E52 77183C12 08BDD552 80CAE8B3 04F36571 3A356E65 A451E165 ECC9AC0A C46E3771 342C8FE5 AEDD0926 85338E23
Secreto compartido H	2C1DCC17 73346492 C6636A36 EE4B965E 292E9AAE 7EE37736 EF58B9D0 A043F348 403A8CF3 3CA7DC0D 9DF61D08 89CE2442 4FF97C1A AD48A5CA 2A554B07 1EF7638D
Generador de correspondencia $\hat{G}$	89F0B5EA BF3BE293 C75903A3 98613192 5C9F5B51 5CA95AF4 85DC7E88 6F03245D 44BEFB2D D3A0DBD7 1CB5E618 971CF474 7F12B79E 548379A4 0E45963B AAF3E829

Las APDU siguientes se intercambian entre el terminal y la microplaqueta para hacer la correspondencia con el nonce.

T>C :	10 86 00 00 45 7C 43 81 41 04 7F 1D 41 0A DB 7D DB 3B 84 BF 10 30 80 09 81 A9 10 5D 74 57 B4 A3 AD E0 02 38 4F 30 86 C6 7E DE 1A B8 89 10 4A 27 DB 6D 84 2B 01 90 20 FB F3 CE AC B0 DC 62 7F 7B DC AC 29 96 9E 19 D0 E5 53 C1 00
C>T :	7C 43 82 41 04 A2 34 23 6A A9 B9 62 1E 8E FB 73 B5 24 5C 0E 09 D2 57 6E 52 77 18 3C 12 08 BD D5 52 80 CA E8 B3 04 F3 65 71 3A 35 6E 65 A4 51 E1 65 EC C9 AC 0A C4 6E 37 71 34 2C 8F E5 AE DD 09 26 85 33 8E 23 90 00

La estructura de las APDU puede describirse como sigue:

Comando					
CLA	10		Encadenamiento de órdenes		
INS	86		GENERAL AUTHENTICATE		
P1/P2	00 00		Claves y protocolo implícitamente conocidos		
Lc	45		Longitud de los datos		
Datos	Rótulo	Longitud	Valor	Comentario	
	7C	43	-	Datos de autenticación dinámica	
	81	41		Datos de correspondencia	
			04		Punto no comprimido
			7F 1D 41 0A ... 86 C6 7E DE		Coordenada x
			1A B8 89 10... D0 E5 53 C1		Coordenada y
Le	00		La longitud de bytes máxima prevista del campo de datos de respuesta es 256		
Respuesta					
Datos	Rótulo	Longitud	Valor	Comentario	
	7C	43		Datos de autenticación dinámica	
	82	41		Datos de correspondencia	
			04		Punto no comprimido
			A2 34 23 6A ... 80 CA E8 B3		Coordenada x
			04 F3 65 71... 85 33 8E 23		Coordenada y
Bytes de estado	90 00		Procesamiento normal		

Ejecución del acuerdo de claves

En la tercera etapa, la microplaqueta y el terminal ejecutan un acuerdo de claves ECDH anónimo utilizando los nuevos parámetros de dominio determinados por el generador de grupo efímero de la etapa previa. Solamente se requiere la coordenada x como secreto compartido dado que KDF utiliza solo la primera coordenada para obtener las claves de sesión.

Clave privada del terminal	76ECFDAA 9841C323 A3F5FC5E 88B88DB3 EFF7E35E BF57A7E6 946CB630 006C2120
Clave pública del terminal	446C9340 84D9DAB8 63944F21 9520076C 29EE3F7A E6722B11 FF319EC1 C7728F95 5483400B FF60BF0C 59292700 09277DC2 A515E125 75010AD9 BA916CF1 BF86FEFC
Clave privada de la microplaqueta	CD626EF3 C256E235 FE8912CA C28279E6 26008EDA 6B3A05C4 CF862A3B DAB79E78
Clave pública de la microplaqueta	02AD566F 3C6EC7F9 324509AD 50A51FA5 2030782A 4968FCFE DF737DAE A9933331 11C3B9B4 C2287789 BD137E7F 8AA882E2 A3C633CC D6ECC2C6 3C57AD40 1A09C2E1
Secreto compartido	67950559 D0C06B4D 4B86972D 14460837 461087F8 419FDBC3 6AAF6CEA AC462832

El acuerdo de claves se ejecuta como sigue:

T>C :	10 86 00 00 45 7C 43 83 41 04 44 6C 93 40 84 D9 DA B8 63 94 4F 21 95 20 07 6C 29 EE 3F 7A E6 72 2B 11 FF 31 9E C1 C7 72 8F 95 54 83 40 0B FF 60 BF 0C 59 29 27 00 09 27 7D C2 A5 15 E1 25 75 01 0A D9 BA 91 6C F1 BF 86 FE FC 00
C>T :	7C 43 84 41 04 02 AD 56 6F 3C 6E C7 F9 32 45 09 AD 50 A5 1F A5 20 30 78 2A 49 68 FC FE DF 73 7D AE A9 93 33 31 11 C3 B9 B4 C2 28 77 89 BD 13 7E 7F 8A A8 82 E2 A3 C6 33 CC D6 EC C2 C6 3C 57 AD 40 1A 09 C2 E1 90 00

La codificación del acuerdo de claves se examina en la tabla siguiente:

Comando				
CLA	10		Encadenamiento de órdenes	
INS	86		GENERAL AUTHENTICATE	
P1/P2	00 00		Claves y protocolo implícitamente conocidos	
Lc	45		Longitud de los datos	
Datos	Rótulo	Longitud	Valor	Comentario
	7C	43	-	Datos de autenticación dinámica
	83	41		Clave pública efímera del terminal
			04	Punto no comprimido

			44 6C 93 40 ... C7 72 8F 95		Coordenada x
			54 83 40 0B ... BF 86 FE FC		Coordenada y
Le	00		La longitud de bytes máxima prevista del campo de datos de respuesta es 256		
Respuesta					
Datos	Rótulo	Longitud	Valor	Comentario	
	7C	43		Datos de autenticación dinámica	
	84	41		Clave pública efímera de la microplaqueta	
			04		Punto no comprimido
			02 AD 56 6F ... A9 93 33 31		Coordenada x
			11 C3 B9 B4 ... 1A 09 C2 E1		Coordenada y
Bytes de estado	90 00		Procesamiento normal		

Por medio de KDF, las claves de sesión AES 128 KS_{Enc} y KS_{MAC} se obtienen del secreto compartido y son:

KS_{Enc}	0A9DA4DB 03BDDE39 FC5202BC 44B2E89E
KS_{MAC}	4B1C0649 1ED5140C A2B537D3 44C6C0B1

Autenticación mutua

Los testigos de autenticación se obtienen mediante KS_{MAC} utilizando como entrada:

Datos de entrada para T_{IFD}	7F494F06 0A04007F 00070202 04060286 410402AD 566F3C6E C7F93245 09AD50A5 1FA52030 782A4968 FCFEDF73 7DAEA993 333111C3 B9B4C228 7789BD13 7E7F8AA8 82E2A3C6 33CCD6EC C2C63C57 AD401A09 C2E1
Datos de entrada para T_{IC}	7F494F06 0A04007F 00070202 04060286 4104446C 934084D9 DAB86394 4F219520 076C29EE 3F7AE672 2B11FF31 9EC1C772 8F955483 400BFF60 BF0C5929 27000927 7DC2A515 E1257501 0AD9BA91 6CF1BF86 FEFC

La codificación de los datos de entrada figura a continuación.

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
7F49	4F		CLAVE PÚBLICA	Datos de entrada para T_{IFD}
06	0A	04 00 7F 00 07 02 02 04 06 02	IDENTIFICADOR DE OBJETO	PACE con ECDH, correspondencia de autenticación de microplaqueta y clave de sesión AES 128
86	41		PUNTO DE CURVA ELÍPTICA	Punto público efímero de la microplaqueta
		04		Punto no comprimido
		02 AD 56 6F... A9 93 33 31		Coordenada x
		11 C3 B9 B4 ... 1A 09 C2 E1		Coordenada y

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
7F49	4F		CLAVE PÚBLICA	Datos de entrada para T_{IC}
06	0A	04 00 7F 00 07 02 02 04 06 02	IDENTIFICADOR DE OBJETO	PACE con ECDH, correspondencia de autenticación de microplaqueta y claves de sesión AES 128
86	41		PUNTO DE CURVA ELÍPTICA	Punto público efímero del terminal
		04		Punto no comprimido
		44 6C 93 40 ... C7 72 8F 95		Coordenada x
		54 83 40 0B ... BF 86 FE FC		Coordenada y

Los testigos de autenticación calculados son los siguientes:

T_{IFD}	E86BD060 18A1CD3B
T_{IC}	8596CF05 5C67C1A3

Finalmente, estos testigos se intercambian y verifican.

T>C :	00 86 00 00 0C 7C 0A 85 08 E8 6B D0 60 18 A1 CD 3B 00
C>T :	7C 3C 86 08 85 96 CF 05 5C 67 C1 A3 8A 30 1E EA 96 4D AA E3 72 AC 99 0E 3E FD E6 33 33 53 BF C8 9A 67 04 D9 3D A8 79 8C F7 7F 5B 7A 54 BD 10 CB A3 72 B4 2B E0 B9 B5 F2 8A A8 DE 2F 4F 92 90 00

La codificación de la autenticacióin mutua se examina en la tabla siguiente:

Comando					
CLA	10		No se produce encadenamiento de órdenes (última orden de la cadena)		
INS	86		GENERAL AUTHENTICATE		
P1/P2	00 00		Claves y protocolo implícitamente conocidos		
Lc	0C		Longitud de los datos		
Datos	Rótulo	Longitud	Valor	Comentario	
	7C	0A	-	Datos de autenticación dinámica	
	85	08		Clave pública efímera del terminal	
			E8 6B D0 60 18 A1 CD 3B		T_{IFD}
Le	00		La longitud de bytes máxima prevista del campo de datos de respuesta es 256		
Respuesta					
Datos	Rótulo	Longitud	Valor	Comentario	
	7C	3C		Datos de autenticación dinámica	
	86	08		Clave pública efímera de la microplaqueta	
			85 96 CF 05 5C 67 C1 A3		T_{IC}
	8A	30			Coordenada x
			1E EA 96 4D … DE 2F 4F 92		Datos de autenticación de microplaqueta cifrada
Bytes de estado	90 00		Procesamiento normal		

Autenticación de microplaqueta

Obtiene `ChipAuthenticationPublicKeyInfo`de EF.CardSecurity

ChipAuthenticationPublicKeyInfo	`30620609 04007F00 07020201 02305230 0C060704 007F0007 01020201 0D034200 04187270 9494399E 7470A643 1BE25E83 EEE24FEA 568C2ED2 8DB48E05 DB3A610D C884D256 A40E35EF CB59BF67 53D3A489 D28C7A4D 973C2DA1 38A6E7A4 A08F68E1 6F02010D`

La estructura detallada de `ChipAuthenticationPublicKeyInfo` se desglosa por elementos en la table siguiente.

Rótulo	***Longitud***	***Valor***	***Tipo ASN.1***	***Comentario***
30	62		SECUENCIA	ChipAuthenticationPublicKeyInfo
06	09	04 00 7F 00 07 02 02 01 02	IDENTIFCADOR DE OBJETO	id-PK-ECDH
30	52		SECUENCIA	SubjectPublicKeyInfo
30	0C		SECUENCIA	Parámetros de dominio normalizados Brainpool P256r1
06	07	04 00 7F 00 07 01 02	IDENTIFCADOR DE OBJETO	standardizedDomainParameters
02	01	0D	ENTERO	Brainpool256r1
03	42	00 04 18 72 70 … 8F 68 E1 6F	CADENA DE BITS	Clave pública CA
02	01	0D	ENTERO	keyID 13

Para la autenticación de microplaqueta se utilizan los datos siguientes:

Datos de autenticación de microplaqueta cifrados	`1EEA964D AAE372AC 990E3EFD E6333353 BFC89A67 04D93DA8 798CF77F 5B7A54BD 10CBA372 B42BE0B9 B5F28AA8 DE2F4F92`
Datos de autenticación de microplaqueta descifrados	`85DC3FA9 3D0952BF A82F5FD1 89EE75BD 82F11D1F 0B8ED4BF 5319AC9B 53C426B3`
IV para descrifrado/cifrado de datos CA IV = $E(KS_{ENC}, -1)$	`F6A3B75A1 E933941 DD7A13E2 520779DF`

Clave pública de microplaqueta de la nonce de correspondencia de GENERAL AUTHENTICATE $PK_{MAP,IC}$	A234236A A9B9621E 8EFB73B5 245C0E09 D2576E52 77183C12 08BDD552 80CAE8B3 04F36571 3A356E65 A451E165 ECC9AC0A C46E3771 342C8FE5 AEDD0926 85338E23
Clave CA pública de la microplaqueta de ChipAuthenticationPublicKeyInfo PK_{IC}	18727094 94399E74 70A6431B E25E83EE E24FEA56 8C2ED28D B48E05DB 3A610DC8 84D256A4 0E35EFCB 59BF6753 D3A489D2 8C7A4D97 3C2DA138 A6E7A4A0 8F68E16F

El terminal verifica que $PK_{MAP,IC} = KA(CA_{IC}, PK_{IC}, D_{IC})$.

— — — — — — — —

Apéndice J de la Parte 11

PROCEDIMIENTOS DE INSPECCIÓN (INFORMATIVO)

J.1 PROCEDIMIENTO DE INSPECCIÓN PARA LA APLICACIÓN eMRTD

En esta sección se describe un procedimiento de inspección que contiene solo una aplicación eMRTD (“documentos LDS1”).

1. Obtención de acceso al IC sin contacto (véase la sección 4.2)

   - Si el acceso al IC está protegido, pueden usarse el PACE o el BAC en esta etapa, aunque se recomienda utilizar el PACE por razones de seguridad. Desde el 1 de enero de 2018 los eMRTD admiten únicamente el PACE.

   - Si es compatible con el IC y el terminal, debería utilizarse el PACE-CAM por razones de rendimiento.

   - El IC otorga acceso a datos menos sensibles en la aplicación eMRTD y a EF.CardSecurity en el fichero maestro, si está presente.

2. Inicio de la autenticación de datos

   - Leer el objeto de seguridad del documento y verificar la firma, incluida la verificación en cadena de los certificados del firmante del documento.

3. Autenticación de la microplaqueta

   - Dependiendo de qué admita el IC, realizar una autenticación de la microplaqueta o autenticación activa. La presencia del EF.DG15 en la aplicación eMRTD indica si se admite la autenticación activa y la presencia de los `SecurityInfos` correspondientes en EF.DG14 indica si se admite la autenticación de la microplaqueta.

   - Esta etapa también puede realizarse como parte de la etapa 1 si se usa el PACE con correspondencia de autenticación de microplaqueta.

   - La autenticación solo está completa en combinación con la autenticación del fichero que contiene la clave pública (EF.CardSecurity, EF.DG14 o EF.DG15) usada para esta etapa.

4. Control de acceso adicional

   - Es necesario realizar la autenticación del terminal si lo requiere la configuración del eMRTD para acceder a los datos sensibles, i. e. EF.DG3 y/o EF.DG4.

5. Lectura de datos

   - La lectura de datos puede iniciarse en cuanto se otorguen los derechos de acceso necesarios, p. ej., los datos menos sensibles pueden leerse después de la etapa 1.

   - Los datos no deben considerarse genuinos sin la autenticación de los datos leídos (etapa 2).

J.2 PROCEDIMIENTO DE INSPECCIÓN PARA LOS eMRTD CON MUCHAS APLICACIONES

En esta sección se describe un procedimiento de inspección diseñado para los eMRTD que contienen una o más aplicaciones además de la aplicación eMRTD (“documentos LDS2”). Este procedimiento también puede emplearse para acceder a la aplicación eMRTD solamente.

1. Obtención de acceso al IC sin contacto (véase la sección 4.2)

   - En este contexto solo está disponible el PACE para obtener acceso al IC.
   - Si lo admiten el IC y el terminal, debería usarse el PACE-CAM por razones de rendimiento.
   - El IC otorga acceso a datos menos sensibles en la aplicación eMRTD y a EF.CardSecurity en el fichero maestro.

2. Verificar la presencia de EF.CardSecurity

   - Si EF.CardSecurity no está presente, el eMRTD no admite la autenticación en el fichero maestro (lo que implica que el IC solo contiene una aplicación eMRTD). En este caso, seleccionar la aplicación eMRTD y seguir con la etapa 2 del procedimiento de la sección J.1 del presente apéndice.

3. Inicio de la autenticación de datos

   - Leer EF.CardSecurity y verificar la firma, incluida la verificación en cadena del certificado del firmante del documento.
   - Los datos de la aplicación eMRTD están protegidos por medio del objeto de seguridad del documento, que debe verificarse cuando se lean los datos de esta aplicación. Los datos de otras aplicaciones están protegidos por las firmas de los datos, que también deben verificarse tras haber leído estos datos.

4. Autenticación de la microplaqueta

   - Realizar la autenticación de la microplaqueta en el fichero maestro. Si la información necesaria no está contenida en los `SecurityInfos` en EF.CardSecurity, el IC no admite la autenticación en el fichero maestro. En este caso seleccionar la aplicación eMRTD y continuar con la etapa 2 del procedimiento de la sección J.1 del presente apéndice.
   - Esta etapa puede realizarse como parte de la etapa 1, si se usa el PACE con correspondencia de autenticación de microplaqueta.
   - La autenticación solo está completa en combinación con la autenticación del fichero que contiene la clave pública (EF.CardSecurity) usada para esta etapa.

5. Control de acceso adicional

   - Realizar la autenticación del terminal.
   - Si solo se requiere el acceso de lectura a datos menos sensibles en la aplicación eMRTD, puede saltarse esta etapa.

6. Lectura/Escritura de datos

   - La lectura/escritura de datos incluye la selección de las aplicaciones que contienen los ficheros.

   - La lectura de datos puede iniciarse en cuanto se hayan garantizado los derechos de acceso necesarios, p. ej., los datos menos sensibles de la aplicación eMRTD pueden leerse después de la etapa 1.

   - los datos no deben considerarse genuinos sin la autenticación de los datos leídos (etapa 3).

— — — — — — — —

Apéndice K de la Parte 11

CONTROL DE ACCESO AMPLIADO DE LA UNIÓN EUROPEA (INFORMATIVO)

La definición de la autenticación del terminal del presente documento se basa en el control de acceso ampliado tal como se usa en la Unión Europea (véase [TR-03110]), con el fin de proteger el acceso a las huellas digitales almacenadas en la aplicación LDS1. En este apéndice se señalan las diferencias entre [TR-03110] y los protocolos definidos en el presente documento.

El procedimiento avanzado de inspección utilizado para acceder a los eMRTD equipados con control de acceso ampliado con arreglo a [TR-03110] comprende las siguientes etapas:

1. Realizar el procedimiento de acceso a la microplaqueta (véase la sección 4.2) y seleccionar la aplicación eMRTD;

2. Realizar la autenticación de la microplaqueta en la aplicación eMRTD (véase la sección 6.2) e iniciar la autenticación pasiva (véase la sección 5.1);

3. Realizar la autenticación del terminal (véase más adelante) en la aplicación eMRTD (véase la sección 7.1).

Nota.— En el control de acceso ampliado de la Unión Europea, tanto la autenticación de la microplaqueta como la del terminal se ejecutan en la aplicación eMRTD. Las especificaciones de este documento permiten que estos protocolos, en función del contexto, se ejecuten en la aplicación eMRTD o en el fichero maestro.

K.1 DERECHOS DE ACCESO

Tabla K-1. Autorización de los sistemas de inspección

7	6	5	4	3	2	1	0	Descripción
x	x	-	-	-	-	-	-	Función (véase el Doc 9303-12)
-	-	x	x	x	x	x	x	Derechos de acceso
-	-	x	x	x	x	-	-	RFU
-	-	-	-	-	-	1	-	Acceso de lectura a la aplicación eMRTD: DG4 (iris)
-	-	-	-	-	-	-	1	Acceso de lectura a la aplicación eMRTD: DG3 (huella dactilar)

Los derechos de acceso a los grupos de datos en aplicaciones distintas de la del eMRTD se transmiten a través de las ampliaciones de autorización definidas en las Partes 12 y 10 del Doc 9303. Los derechos de acceso a las huellas dactilares (y el iris) se transmiten a través de la plantilla de autorización de la persona titular del certificado:

Para el cálculo de los derechos de acceso efectivos, véase la sección 7.1.4.3.6.

K.2 EF.CVCA

De acuerdo con la especificación, los puntos de confianza (referencias de la autoridad de certificación) conocidos del IC para la verificación del certificado como parte de la autenticación del terminal se transmiten al IFD como parte del protocolo PACE (véase la sección 4.4.3.5).

En lugar de eso, el control de acceso ampliado de la Unión Europea define un fichero transparente EF.CVCA en la aplicación eMRTD. A continuación se reproduce la especificación:

Tabla K-2. Fichero elemental EF.CVCA

Nombre de fichero	EF.CVCA
ID de fichero	0x011C (por defecto)
ID de fichero breve	0x1C (por defecto)
Acceso de lectura	PACE
Acceso de escritura	NUNCA (actualización únicamente interna)
Tamaño	36 bytes (fijos) rellenado con octetos de valor 0x00
Contenido	[CARi][||CARi-1][||0x00..00]

Si el IC admite la autenticación del terminal en la aplicación eMRTD, DEBE hacer que las referencias de las claves públicas de la CVCA sean adecuadas para los sistemas de inspección disponibles en un fichero elemental transparente EF.CVCA en la aplicación eMRTD, según se especifica en la tabla K-2.

Este fichero CONTENDRÁ una secuencia de objetos de datos de las referencias a la autoridad de certificación (CAR) (véase el Doc 9303-12) adecuados para la autenticación del terminal.

- CONTENDRÁ como mucho dos objetos de datos de las referencias a la autoridad de certificación.
- La referencia a la autoridad de certificación más reciente SERÁ el primer objeto de datos de esta lista.
- El fichero DEBE rellenarse añadiendo octetos de valor 0x00.

El fichero EF.CVCA tiene un identificador EF por defecto y un identificador EF breve. Si los valores por defecto no pueden utilizarse, el identificador EF (breve) SE ESPECIFICARÁ en el parámetro OPCIONAL `efCVCA` de la `TerminalAuthenticationInfo`. Si se usa el `efCVCA` para indicar el identificador EF que ha de utilizarse, entonces el identificador EF por defecto queda anulado. Si en `efCVCA` no se da un identificador EF breve, el fichero EF.CVCA DEBE seleccionarse explícitamente utilizando el identificador EF que se haya dado.

```
TerminalAuthenticationInfo ::= SEQUENCE {
  protocol OBJECT IDENTIFIER(id-TA),
  version  INTEGER, -- MUST be 1
  efCVCA   FileID OPTIONAL
}

FileID ::= SEQUENCE {
  fid  OCTET STRING (SIZE(2)),
  sfid OCTET STRING (SIZE(1)) OPTIONAL
}
```

— FIN —

ISBN 978-92-9265-532-7