OPERATIVA DE COMANDOS
COMMANDS OPERATIVE

CRYPTOSEC LAN & CRYPTOSEC BANKING

REALIA TECHNOLOGIES, SL

21/03/2013

Contenido

I. ESPAÑOL

1. Claves LMK

1.1. Descripción

A partir de una clave maestra, cargada por custodios por el interfaz del HSM, se diversifican otras que quedan almacenadas dentro del módulo, llamadas claves LMK.

Estas claves sirven, en el almacenamiento externo, para cifrar otras claves que serán almacenadas en una base de datos. De esta manera se agrupan las claves por uso evitando el utilizar una clave que tiene un uso determinado, para otra función.

En el almacenamiento interno, las claves LMK sirven como referencia para agrupar las claves por funciones.

Todas las claves diversificadas son de triple longitud. Cada clave almacenada en la BB.DD. irá asociada a una de las claves LMK.

La siguiente tabla muestra los tipos de LMKs:

Tipo	Descripción	Operativa	Sólo en Autorizado
LMK 0	Clave Maestra	Cálculo KCV (6 caracteres) de la LMK 0	NO
LMK 1	Claves de Custodio	Generación/Borrado	SI
		Importación/Exportación (RSA)	SI
		Importación/Exportación por componentes	SI
		Cálculo KCV (6 caracteres)	NO
LMK 2	Claves de Transporte de claves	Generación/Borrado	NO
		Importación/Exportación por componentes	SI
		Importación/Exportación cifrada con clave custodio	NO
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
LMK 3 *	Claves de Transporte de bloque de PIN	Generación/Borrado	NO
		Importación/Exportación cifrada con clave de transporte	NO
		Importación/Exportación (RSA)	NO
		Diversificación	NO
		Cálculo KCV (6 caracteres)	NO
		Conversiones del bloque de PIN	NO
LMK 4*	Claves de PIN	Generación/Borrado	NO

Tipo	Descripción	Operativa	Sólo en Autorizado
		Importación/Exportación por componentes	SI
		Importación/Exportación cifrada con clave de transporte	NO
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
		Construcción del bloque de PIN	NO
		Verificación del bloque de PIN	NO
LMK 5*	Claves de CVV	Generación/Borrado	NO
		Importación/Exportación por componentes	SI
		Importación/Exportación cifrada con clave de transporte	NO
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
		Cálculo de CSS/CVC/CVV	NO
		Verificación de CSS/CVC/CVV	NO
LMK 6*	Claves de Autorización EMV	Generación/Borrado	NO
		Importación/Exportación por componentes	SI
		Importación/Exportación cifrada con clave de transporte	SI
		Importación/Exportación (RSA)	NO
		Diversificación	NO
		Cálculo KCV (6 caracteres)	NO
		Comandos de autorización transacciones EMV	NO
		Comandos de tratamiento de seguridad de los scripts	NO
LMK 7	Claves de cifrado de datos	Generación/Borrado	NO
		Importación/Exportación por componentes	SI
		Importación/Exportación cifrada con clave de transporte	NO
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
		Diversificación	NO
		Comandos de cifrado/descifrado/CBC	NO
LMK 8	Claves de MAC	Generación/Borrado	NO
		Importación/Exportación cifrada con clave de transporte	NO
		Importación/Exportación (RSA)	NO
		Diversificación	NO
		Cálculo KCV (6 caracteres)	NO
		Generación de MAC	NO
		Verificación de MAC	NO
LMK 9*	Claves Monedero	Generación/Borrado	NO
		Importación/Exportación por componentes	SI
		Importación/Exportación cifrada con clave de transporte	NO
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
		Diversificación	NO
LMK 10*	Claves de personalización EMV	Generación/Borrado	NO
		Importación/Exportación por componentes	SI
		Importación/Exportación cifrada con clave de transporte	SI
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
		Diversificación	NO
LMK 11	Claves de TAF	Generación/Borrado	NO
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
LMK 12	Claves de VIAT	Importación/Exportación cifrada con clave de transporte	SI

Tipo	Descripción	Operativa	Sólo en Autorizado
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
LMK 13	Claves de PIN Irreversible	Generación/Borrado	NO
		Importación/Exportación por componentes	NO
		Importación/Exportación cifrada con clave de transporte	SI
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
LMK 14*	Claves de Autenticación de Terminal	Generación/Borrado	NO
		Importación/Exportación por componentes	NO
		Importación/Exportación cifrada con clave de transporte	SI
		Importación/Exportación (RSA)	SI
		Cálculo KCV (6 caracteres)	NO
		Diversificación	NO
LMK 15	Claves de Tabla de Decimalización	Generación/Borrado	NO
		Importación/Exportación por componentes	NO
		Importación/Exportación cifrada con clave de transporte	NO
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
		Comandos de bloque de PIN	NO
LMK 16	Claves de Firma de Log	Generación/Borrado	SI
		Importación/Exportación por componentes	SI
		Importación/Exportación cifrada con clave de transporte	SI
		Importación/Exportación (RSA)	SI
		Cálculo KCV (6 caracteres)	NO
LMK 17	Claves de cifrado de contraseñas	Generación/Borrado	NO
		Importación/Exportación por componentes	SI
		Importación/Exportación cifrada con clave de transporte	NO
		Importación/Exportación (RSA)	NO
		Cálculo KCV (6 caracteres)	NO
		Comandos de cifrado	NO
LMK 60	Claves RSA Cifrar/Descifrar	Generación	NO
		Importación/Exportación cifrada con clave de transporte	NO
		Cálculo KCV (6 caracteres)	SI
		Trasladar clave privada de una LMK a otra LMK	NO
		Cifrado/descifrado RSA	NO
		Calculo/verificación firmas	NO
LMK 61	Claves RSA Firmar/Verificar	Generación	NO
		Importación/Exportación cifrada con clave de transporte	NO
		Cálculo KCV (6 caracteres)	SI
		Trasladar clave privada de una LMK a otra LMK	NO
		Cifrado/descifrado RSA	NO
		Calculo/verificación firmas	NO
LMK 62	Claves RSA Firmar/Verificar Cifrar/Descifrar	Generación	NO
		Importación/Exportación cifrada con clave de transporte	NO
		Cálculo KCV (6 caracteres)	SI
		Trasladar clave privada de una LMK a otra LMK	NO
		Cifrado/descifrado RSA	NO
		Calculo/verificación firmas	---
LMK 62-98	Reservadas para uso futuro	----------	SI
LMK 99	Claves de uso libre	Generación/Borrado	NO

Tipo	Descripción	Operativa	Sólo en Autorizado
	(Pruebas)	Importación/Exportación por componentes	NO
		Importación/Exportación cifrada con clave de transporte	NO
		Importación/Exportación (RSA)	NO
		Cálculo KCV (valor parametrizable)	NO
		Diversificación	NO
		Comandos uso libre (todos los comandos definidos)	

* SOLO CryptosecBANKING

2. Protocolo de Comunicación con los HSM

2.1. Estructura de los mensajes

Tipos de datos:

Tipo	Comentario
A	Dato alfanumérico
H	Dato hexadecimal
N	Dato numérico
B	Dato binario
K	Estructura de clave definida a continuación

Estructura de Mensaje Comando:

Tipo	Longitud	Comentario
D	6	Longitud del mensaje
A	var	Cabecera (*)
H	4	Identificador comando
--	var	Datos del comando

Estructura del Mensaje Respuesta:

Tipo	Longitud	Comentario
D	6	Longitud del mensaje
A	var	Cabecera
H	4	Identificador comando
H	8	Estado del comando
--	var	Datos de la respuesta (2*)

(*)El formato de los mensajes admite una cabecera de longitud parametrizable por la Entidad. En la respuesta del HSM llegará esta cabecera sin modificaciones, para que la aplicación que realizó la petición compruebe que el mensaje que le han devuelto es realmente para ella.

(2*)Este campo sólo estará presente si el comando se ha resuelto satisfactoriamente, esto es, si el estado del comando es '00000000'.

2.2. Estructura de Clave

La estructura de presentación de claves a/desde el HSM será la siguiente:

Tipo	Longitud	Comentario
A	1	Indicador de tipo de clave (*)
N	4	Longitud de la clave (longitud del campo siguiente)(2*)
H	var	Valor de la clave(3*)
A	3	Cifrado de la clave (4*)
N	2	Longitud del KCV o MAC (longitud del campo siguiente) (5*)
H	var	Valor del KCV o MAC (5*)

(*) El indicador de tipo de clave puede tomar los siguientes valores:

- D – para claves DES almacenadas externamente.
- S – para claves DES almacenadas internamente.
- R – para claves RSA privadas.
- P – para claves RSA públicas.
- I – para claves RSA privadas en formato IBM estructuradas en CRT.
- V – para clave contenidas en DB.

(2*)Cuando el tipo de clave es I (RSA formato IBM) entonces la longitud de la clave deberá ser de 5000 bytes.

(3*) Dependiendo del tipo de clave, esta vendrá formateada en:

- Claves DES almacenadas externamente o a almacenar internamente, ningún formato en especial.
- Claves DES almacenadas internamente, cinco dígitos correspondientes al identificador de la clave devuelto por el HSM.
- Claves RSA públicas, cuyo valor es la codificación de DER de la estructura ASN.1 definida en el estándar PKCS#1.
- Claves RSA privadas (y no en formato IBM estructuradas en CRT), cuyo valor es la codificación en DER de la estructura ASN.1 PrivateKeyInfo definida en el estándar PKCS#8.
- Claves RSA privadas en formato IBM estructuradas en CRT: estructura propietaria de IBM.

(4*) En este campo estará codificado el tipo de cifrado de la clave:

- L00, L01 …… L99 indica que la clave está cifrada con una LMK alojada en el HSM.
- D00 indica que la clave está cifrada con algoritmo DES, con una clave externa al HSM.
- El campo vale N00 en claves DES almacenadas internamente.
- N00 indica que la clave no está cifrada.
- R00 indica que la clave está cifrada con algoritmo RSA, con una clave externa al HSM.

(5*) El KCV de la clave vendrá dado siempre que la clave pueda admitir este tipo de valor y se quiera especificar como un sistema de verificación de clave. Para no especificar KCV, este campo se deja vacío, siendo entonces su longitud 00. En el caso de que la clave no admita el KCV entonces el valor no se tendrá en cuenta.

Para el caso de claves públicas RSA, en este campo se pasará el MAC.

3. Formato de Comandos

3.1. Administración de Claves

3.1.1. Generación de una Clave - 0101 -

Permite generar todos los tipos de claves que es capaz de manejar el sistema: claves DES de distinta longitud tanto internas como externas y claves RSA haciendo uso de distintos tipos de test.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0101.
LMK	N	2	Etiqueta LMK asociada a la clave a generar. El valor deberá ser distinto de 00.
T	A	1	Tipo de clave a generar. - D para generar claves DES (almacenamiento externo). - S para generar claves DES (almacenamiento interno). - R para generar claves RSA.
L	N	4	Indicador longitud en bits de la clave a generar: - 0064 para claves DES simples. - 0128 para claves DES dobles. - 0192 para claves DES triples. - Desde 0512 hasta 2048 en múltiplos de 32 bits para claves RSA.
E	N	1	(*)Exponente de la clave RSA a generar: - 0 si el exponente es 3. - 1 si es Fermat4.
M	N	2	(2*)Modo de generación de clave RSA: - 00 Modo Fermat. - 01 Modo automático. - Desde 02 hasta 50, número de pasadas test de Miller-Rabin.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
K1	K	var	Clave generada.
K2	K	var	Clave pública (*).

(*) Sólo si T = R.

(2*) Sólo si T = R. El modo de generación consiste en los test de primalidad que se les hace pasar a las componentes que forman una clave RSA. El modo más sencillo es el test de Fermat; el siguiente modo es el automático en el cual, el módulo calcula el número de pasadas del test de Miller-Rabin para que la probabilidad de falsos positivos sea menor de 2^{-100}. En todos los modos se lleva a cabo un test frente a primos pequeños.

3.1.2. Borrado de Clave - 0102 -

Permite borrar una clave DES almacenada internamente.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0102.
K1	K	var	Clave DES interna a borrar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.

3.1.3. Listado de Claves - 0103 -

Permite obtener un listado de claves DES almacenadas internamente.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0103.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
NK	N	5	Número de claves DES internas.
HK1	N	5	(*) IDentificador de clave DES interna.
LMKK1	N	2	(*) LMK de dicha clave.
…	…	…	…
HKNK	N	5	(*) Identificador de clave DES interna.
LMKKNK	N	2	(*) LMK de dicha clave.

(*) Tantos pares de campos identificador/LMK como número de claves indique el campo NK.

3.1.4. Borrado de Clave DB - 0104 -

Permite borrar una clave RSA interna.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0104.
LEN	N	2	Longitud de la etiqueta

Dato	Tipo	Longitud	Comentario
L	A	var	Etiqueta

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.

3.1.5. Listado de Claves DB - 0105 -

Permite obtener un listado de claves RSA almacenadas internamente.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0105.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
NK	N	5	Número de claves Internas.
LEN	N	2	(*) Longitud del campo siguiente
L	A	var	(*) Etiqueta
…	…	…	…
LEN	N	2	(*) Longitud del campo siguiente
L	A	var	(*) Etiqueta

(*) Tantos pares de campos como número de claves indique el campo NK.

3.1.6. Obtención de una clave DB - 0106 -

Permite obtener un listado de claves RSA almacenadas internamente.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0106.
LEN	N	2	Longitud del campo siguiente
L	A	var	Etiqueta

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.

Dato	Tipo	Longitud	Comentario
K1	K	var	Clave

3.1.7. Almacenamiento de una clave DB - 0107 -

Permite almacenar una clave en el interior.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0107.
K1	K	var	Clave a almacenar.
LEN	N	2	Longitud del campo siguiente
L	A	var	Etiqueta

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.

3.1.8. Importación de Clave - 0201 -

Permite incorporar en el sistema claves externas. La clave a importar vendrá cifrada con una clave de transporte, de custodio o RSA.

Para nuevas implementaciones se sugiere utilizar el comando 0202.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0201.
LMK	N	2	Etiqueta LMK para la clave importada. El valor deberá ser distinto de 00.
K1	K	var	Clave a importar.
K2	K	var	Clave de descifrado, bajo LMK 01, LMK 02, LMK 60 o LMK 62.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
K	K	var	Clave importada.

3.1.9. Importación de Clave v2 - 0202 -

Permite incorporar en el sistema claves externas. La clave a importar vendrá cifrada con una clave de transporte, de custodio o RSA.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0202.
LMK	N	2	Etiqueta LMK para la clave importada. El valor deberá ser distinto de 00.
ALG	H	2	Algoritmo de importación: - 00 EBC. - 01 RSA-PKCS#1v1.5 - 02 RSA-OAEP.
K1	K	var	Clave a importar.
K2	K	var	Clave de descifrado, bajo LMK 01, LMK 02, LMK 60 o LMK 62.
F_1	A	1	(2*)Indicador de almacenamiento interno de clave DES importada: - S.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
K1	K	var	Clave importada.
K2	K	var	Clave pública (*).

(*) Sólo si K1 es una clave RSA.

(2*) Sólo si la clave DES importada se debe almacenar internamente.

3.1.10. Importación de Clave Pública RSA firmada - 0203

Permite incorporar en el sistema claves públicas RSA externas, firmadas por otra clave RSA.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0203.
LMK	N	2	Etiqueta LMK para la clave a importar. El valor deberá ser 60, 61 o 62.
T1	N	1	Tipo de dato contenedor de clave pública: - 0 Estructura de clave pública - 1 Certificado
L1	N	4	(*) Longitud del campo siguiente
K1	H	var	Contenedor de la clave pública
L2	N	4	Longitud de la firma (campo siguiente)

Dato	Tipo	Longitud	Comentario
F1	H	var	Firma de la clave a importar
K2	K	var	Clave de verificación, bajo LMK 61 o 62.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
K1	K	var	Clave importada.

(*) Sólo si T1 = 1.

3.1.11. Importación de Clave Pública RSA sin firmar - 0204 -

Permite incorporar en el sistema claves públicas RSA externas sin que vayan firmadas por otra clave RSA.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0204.
LMK	N	2	Etiqueta LMK para la clave a importar. El valor deberá ser 60, 61 ó 62.
K1	K	var	Clave a importar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
K1	K	var	Clave importada.

3.1.12. Exportación por Componentes - 0301 -

Permite exportar claves DES por componentes a través del terminal. Además, es capaz de generar la clave a exportar.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0301.
M	N	1	Acción a realizar: - 0 Genera clave y la exporta en componentes. - 1 Exporta la clave incluida en el mensaje.
LMK	N	2	(*)Etiqueta LMK para la clave generada. El valor deberá ser distinto de 00.
L1	N	2	(*)Longitud de la clave a generar en bytes.
C	N	1	Número de componentes a devolver, entre 2 y 9.
K	K	var	(2*)Clave a exportar.
L3	N	4	Longitud del siguiente campo en bytes.
T	A	var	Tabla de datos variables de usuario (nombre, custodio, cajero, fechas, finalidad, clave,…). El formato dependerá de la definición del formato de impresión.
F_1	A	1	(3*)Indicador de almacenamiento interno de clave generada: - S.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
K	K	var	(*)Clave generada.

(*) Sólo si M = 0.

(2*) Sólo si M = 1.

(3*) Sólo si M = 0 y si la clave generada se debe almacenar internamente.

Notas sobre la impresión

Es necesario tener previamente definida la cadena de formato de impresión, mediante un comando 1301.

En la exportación por componentes, el argumento T contiene las cadenas de impresión 0 a 15 que se referencian en la cadena de formatos de impresión. No es preciso definir las 16 cadenas, sin embargo, hay que tener en cuenta que estas se numeran de forma correlativa, comenzando por cero. La secuencia “\0” se usa como delimitador de cadenas. El símbolo ^P induce la impresión de la componente de clave. EL símbolo ^T implica la impresión del KCV de la componente de clave.

Para aclarar el funcionamiento de la impresión de componentes, supóngase que se quieren generar e imprimir dos componentes de una clave, acompañadas por su KCV con fines de control, en la forma siguiente:

COMPONENTE: XXXXXXXXXXXXXXXX KCV: YYYYYY

Donde X...X es la componente e Y...Y es el KCV. En este caso va a optarse por mantener la cadena de formato reducida al mínimo:

>L^0^P^1^T>F

Donde ^0 denota el punto de inclusión de la primera cadena de la tabla de datos y ^1 denota el punto de inclusión de la segunda cadena de la tabla de datos. ^P denota impresión de componente de clave y ^T denota impresión del KCV de la componente.

Con esta cadena de formato, para conseguir la impresión buscada, la tabla de datos a incluir en el presente comando será:

COMPONENTE: \0 KCV:

Se ve que el \0 se usa como separador, y que no es necesario incluirlo detrás de la última cadena. La longitud de esta cadena es 00000020.

La alternativa para simplificar (en este caso, eliminar) la tabla de datos sería incluir las cadenas de datos en la cadena de formato:

>L COMPONENTE: ^P KCV: ^T>F

aunque esta aproximación es formalmente menos adecuada.

3.1.13. Exportación de Clave - 0302 -

Permite exportar una clave DES bajo una clave DES o RSA, o exportar una clave RSA bajo una clave DES. La clave exportada irá cifrada bajo una clave externa de transporte, de custodio o RSA.

Para nuevas implementaciones se sugiere utilizar el comando 0303.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0302.
K1	K	var	Clave a exportar.
K2	K	var	Clave bajo la que se exporta la anterior. Bajo LMK 01, LMK 02, LMK 60 o LMK 62.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
K	K	var	Clave exportada.

3.1.14. Exportación de Clave v2 - 0303 -

Permite exportar una clave DES bajo una clave DES o RSA, o exportar una clave RSA bajo una clave DES. La clave exportada irá cifrada bajo una clave externa de transporte, de custodio o RSA.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0303.
ALG	H	2	Algoritmo de exportación: - 00 EBC. - 01 RSA-PKCS#1v1.5 - 02 RSA-OAEP.
K1	K	var	Clave a exportar.
L1	N	4	Longitud del dato siguiente.
D1	H	var	Dato a concatenar.
K2	K	var	Clave bajo la que se exporta la anterior. Bajo LMK 01, LMK 02, LMK 60 o LMK 62.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
K	K	var	Clave exportada.

3.1.15. Cálculo del KCV de Clave - 0401 -

Devuelve el KCV de una clave DES dada.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0401.
K	K	var	Clave de la que se quiere obtener el KCV.
L	N	2	Longitud del KCV a devolver. Este campo sólo se tendrá en cuenta para claves de uso libre, LMK 99. Para las demás claves, no se tendrá en cuenta y se tomará como valor 6.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
L	N	2	Longitud del siguiente campo en bytes.
KCV	H	var	KCV calculado.

3.1.16. Cálculo del KCV de LMK - 0402 -

Devuelve el KCV de una LMK.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0402.
IMM	N	2	Índice de la LMK de la cual se quiere obtener el KCV.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
KCV	H	6	KCV devuelto.

3.1.17. Actualización de almacenamiento de clave - 1801 -

Permite recifrar una clave cifrada bajo una LMK del conjunto antiguo bajo la misma LMK del conjunto actual. De esta manera es posible actualizar, cada vez que se modifique la clave maestra del sistema, todas las claves almacenadas externamente.

Para que este comando funcione correctamente, hay que tener en cuenta:

- Haber cargado una nueva clave maestra del sistema.

- En ese proceso, haber mantenido las LMKs derivadas de la clave maestra anterior.

- La clave a actualizar debe haber sido almacenada bajo una LMK asociada a la citada clave maestra anterior: el proceso de actualización es sucesivo, ya que sólo se almacena con tal fin el conjunto de LMKs más recientemente sustituidas.

- No haber eliminado dichas LMKs previamente a la ejecución de este comando. Una vez eliminadas, sólo será posible recuperar las claves cifradas bajo ese conjunto de claves volviendo a cargar en el sistema la clave maestra asociada.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1801.
K	K	var	Clave que se quiere actualizar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
K	K	var	Clave actualizada.

3.2. Autorización de Transacciones EMV (*CryptosecBANKING)

3.2.1. Cálculo y Validación de DAC - 0501 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0501
K	K	var	Clave de DAC (bajo LMK6).
M	N	1	Modo en el que se ejecuta la función: - 0 cálculo del DAC. - 1 validación del DAC.
D1	H	16	PAN
DAC	H	4	(*)DAC a validar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
DAC	H	4	(2*)Valor DAC calculado.
VER	N	1	(*)Resultado de la validación: - 0 correcto. - 1 incorrecto.

(*) Sólo si M = 1.

(2*) Sólo si M = 0.

3.2.2. Cálculo y Validación de IDN - 0502 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0502.
K	K	var	Clave de IDN (bajo LMK6).
M	N	1	Modo en el que se ejecuta la función: - 0 cálculo del IDN. - 1 validación del IDN.
D1	H	16	PAN
D2	H	16	ATC || 00 || 00 || UN. Normalmente UN=00 00 00 00
IDN	H	4	(*)IDN a validar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
IDN	H	16	(2*)Valor IDN calculado.
VER	N	1	(*)Resultado de la validación: - 0 correcto. - 1 incorrecto.

(*) Sólo si M = 1.

(2*) Sólo si M = 0.

3.2.3. Verificación de ARQC y Generación de ARPC - 0503 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0503.
M	N	1	Modo en el que se ejecuta la función: - 0 Verificar ARQC. - 1 Verificar ARQC y generar ARPC. - 2 Generar ARPC. - 3 Calcular ARQC.
E	N	1	Esquema: - 0 Visa VSDC. - 1 Mastercard. - 2 Diners.
K	K	var	Clave de operación (bajo LMK6).
D1	H	16	PAN / PANseq No – pre-formateado
D2	H	4	(*)Contador de transacciones (ATC).
D3	H	8	(*)Número aleatorio. (UN)
L4	N	3	(2*)Longitud del siguiente campo.
D4	H	var	(2*)Datos de la transacción (NO PADEAR)
D5	H	16	(3*)ARQC/ TC/ AAC a validar y/ o a usar para generar el ARPC.
D6	H	16	(4*)Código de respuesta usado para el ARPC (ARC||000000000000).

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
ARQC	H	16	(5*)ARQC calculado.
VER	N	1	(6*)Resultado de la validación: - 0 correcto. - 1 incorrecto.
ARPC	H	16	(4*)ARPC generado.

(*) Sólo si E = 1 y M ≠ 2, o si E=2.

(2*) Sólo si M = 0, M = 1 ó M = 3.

(3*) Sólo si M = 0, M = 1 ó M = 2.

(4*) Sólo si M = 1 ó M = 2.

(5*) Sólo si M = 3.

(6*) Sólo si M = 0 ó M = 1.

3.3. Seguridad de los Scripts (*CryptosecBANKING)

3.3.1. Firma de Script - 0504 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0504.
E	N	1	Esquema: - 0 Visa VSDC. - 1 Mastercard.
K	K	var	Clave de operación (bajo LMK6).
D1	H	16	Datos de diversificación 1.
D2	H	16	Datos de diversificación para firma script.
L3	N	6	Longitud del siguiente campo.
D3	H	var	Datos a firmar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
F	H	16	Firma.

3.3.2. Cifrado de Script - 0505 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0505.
E	N	1	Esquema: - 0 Visa VSDC. - 1 Mastercard.
K	K	var	Clave de operación (bajo LMK6).
D1	H	16	Datos de diversificación 1.
D2	H	16	Datos de diversificación para cifrado de script.
L3	N	6	Longitud del siguiente campo.
D3	H	var	Datos a cifrar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
L	N	6	Longitud del siguiente campo en bytes.
F	H	var	Datos cifrados.

3.3.3. Script de Cambio de PIN - 0506 - (*CryptosecBANKING)

Para nuevas implementaciones se sugiere utilizar el comando 0507.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0506.
E	N	1	Esquema: - 0 Visa VSDC. - 1 Mastercard.
KT	K	var	Clave de transporte ATM de PIN (bajo LMK3).
K1	K	var	(*)Clave de operaciones 1 (bajo LMK6).
K2	K	var	Clave de operaciones 2 (bajo LMK6).
K3	K	var	Clave de operaciones 3 (bajo LMK6).
F1	N	1	Formato de bloque de PIN entrante: - 1 ISO1. - 2 ISO2. - 4 IBM 3624 - 5 Diebold
F2	N	1	Formato de bloque de PIN saliente: - 0 bloque de PIN I: Bloque de PIN estándar EMV. - 1 bloque de PIN II: VISA sin usar PIN actual. - 2 bloque de PIN III: VISA usando PIN actual.
D1	H	16	Bloque de PIN recibido del cajero (construido con el PIN nuevo).
D2	H	16	(2*)Bloque de PIN recibido del cajero (construido con el PIN antiguo).
D3	H	16	Dato de diversificación 3: 14 dígitos más a la derecha del PAN || PSN.
D4	H	16	Dato de diversificación 4: ATC si VISA, ARQC si Mastercard.
L5	N	6	Longitud del siguiente campo.
D5	H	var	Datos a firmar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
L1	N	6	Longitud del siguiente campo.
PIN	H	var	Bloque de PIN cifrado.
FS	H	16	Firma comando script.

(*) Sólo si E = 0.

(2*) Sólo si F2 = 2.

3.3.4. Script de Cambio de PIN v2 - 0507 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0507.
E	N	1	Esquema: - 0 Visa VSDC. - 1 Mastercard.
KT	K	var	Clave de transporte ATM de PIN (bajo LMK3).
K1	K	var	(*)Clave de operaciones 1 (bajo LMK6).
K2	K	var	Clave de operaciones 2 (bajo LMK6).
K3	K	var	Clave de operaciones 3 (bajo LMK6).
F1	N	1	Formato de bloque de PIN entrante: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM 3624 - 5 Diebold
F2	N	1	Formato de bloque de PIN saliente: - 0 bloque de PIN I: Bloque de PIN estándar EMV. - 1 bloque de PIN II: VISA sin usar PIN actual. - 2 bloque de PIN III: VISA usando PIN actual. - 3 bloque de PIN IV: Mastercard Pay Now & Pay Later.
D1	H	16	Bloque de PIN recibido del cajero (construido con el PIN nuevo).
D2	H	16	(2*)Bloque de PIN recibido del cajero (construido con el PIN antiguo).
D3	H	16	Dato de diversificación 3: PAN.
D3b	H	2	Dato de diversificación 3b: PSN.
D4	H	16	Dato de diversificación 4: ATC si VISA, ARQC si Mastercard.
L5	N	6	Longitud del siguiente campo.
D5	H	var	Datos a firmar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
L1	N	6	Longitud del siguiente campo.
PIN	H	var	Bloque de PIN cifrado.
FS	H	16	Firma comando script.

(*) Sólo si E = 0.

(2*) Sólo si F2 = 2.

3.3.5. Verificación de ARQC y Generación de ARPC (EMV 4.1) - 0508 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0508.
M	N	1	Modo en el que se ejecuta la función: - 0 - Verificar ARQC. - 1 - Verificar ARQC y generar ARPC método 1 EMV 4.1. - 2 - Generar ARPC método 1 EMV 4.1. - 3 - Verificar ARQC y generar ARPC método 2 EMV 4.1. - 4 – Generar ARPC método 2 EMV 4.1.
E	N	1	Esquema: - 2 – VIS1.4.0 y M/CHIP4 usando “Card Key Derivation Method A” y “EMV Common Session Key Derivation Method” - 3 – VIS1.4.0 y M/CHIP4 usando “Card Key Derivation Method B” y “EMV Common Session Key Derivation Method”
K	K	var	Clave de operación (bajo LMK6).
L1	N	2	(*)Longitud PAN/PANseqNo – preFormateado (08 a 20)
D1	N	var	PAN / PANseq No – preFormateado
D2	H	4	Contador de transacciones (ATC).
L3	N	3	(2*)Longitud del siguiente campo.
D3	H	var	(2*)Datos de la transacción (PADEAR)
D4	H	16	ARQC/ TC/ AAC a validar y/o a usar para generar el ARPC.
D5	H	4	(3*)ARC. Usado para generar ARPC
D6	H	8	(4*)CSU. Usado para generar ARPC
L7	N	2	(4*)Longitud Datos Autenticación Propietaria (0…16)
D7	H	var	(4*)Datos Autenticación Propietaria

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
VER	N	1	(5*)Resultado de la validación: - 0 correcto. - 1 incorrecto.
ARPC	H	16	(6*)ARPC generado.
ARQC	H	16	(7*)ARQC calculado.

(*) Sólo sil E=3

(2*) Sólo si M = 0, M = 1 ó M = 3.

(3*) Sólo si M = 1 ó M = 2.

(4*) Sólo si M = 3 ó M = 4.

(5*) Sólo si M = 0, M=1 ó M = 3.

(6*) Sólo si M = 1, M=2 ó M = 3.

(7*) Sólo si la verificación ha sido incorrecta y el HSM está en estado de autorización.

3.4. Funciones de PIN(*CryptosecBANKING)

3.4.1. Cálculo de PIN - 0601 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0601.
K1	K	var	Clave de cálculo de PIN (bajo LMK4).
K2	K	var	Clave de cifrado de PIN (bajo LMK3).
TD	N	16	Tabla de decimalización.
F	N	1	Formato de bloque de PIN: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold
ALG	N	1	Algoritmo de generación de PIN: - 0 IBM3624. - 1 IBM3624 PIN Offset. - 4 IBM German Bank Pool Institution. - 5 Interbank.
LPAN	N	2	Longitud del siguiente campo.
PAN	N	var	PAN.
D1	H	1	(*)Valor de padeo.
LPIN	N	1	(2*)Longitud del PIN.
D2	H	1	(3*)Indicador de clave.
D3	H	3	(3*)Campo de validación.
L5	N	1	(4*)Longitud del PIN Offset.
D5	H	var	(4*)PIN Offset.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
PIN	H	16	Bloque de PIN cifrado.

(*) Sólo si F = 4.

(2*) Sólo si ALG = 0.

(3*) Sólo si ALG = 5.

(4*) Sólo si ALG = 1.

3.4.2. Cálculo de PIN dado un PIN en claro o aleatorio - 0611 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0611.
K1	K	var	Clave de cifrado de PIN (bajo LMK3).
F	N	1	Formato de bloque de PIN: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.
LPAN	N	2	Longitud del PAN.
PAN	N	var	PAN.
D1	H	1	(*)Valor de padeo.
EPIN	N	1	Entrega del PIN - 0 PIN en claro. - 1 random.
LPIN	N	1	Longitud del PIN.
PIN	H	Var	(2*)PIN en claro

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
PIN	H	16	Bloque de PIN cifrado.

(*) Sólo si F = 4.

(2*) Sólo si EPIN = 0.

3.4.3. Verificación de PIN - 0602 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0602.
K1	K	var	Clave de cálculo de PIN (bajo LMK4).
K2	K	var	Clave de cifrado de PIN (bajo LMK3).
TD	N	16	Tabla de decimalización.
F	N	1	Formato de bloque de PIN: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624.

Dato	Tipo	Longitud	Comentario
			- 5 Diebold.
ALG	N	1	Algoritmo de generación de PIN: - 0 IBM3624. - 1 IBM3624 PIN Offset. - 4 IBM German Bank Pool Institution. - 5 Interbank. - 6 ALT_1 PIN Offset.
UKP	N	1	Indicador de UKPT: - 0 no se usa. - 1 se usa.
PIN	H	16	Bloque de PIN.
D1	H	20	(*)Current Key Sequence Number.
PEM	N	2	(2*)PIN Extraction Method: - 00 no usa el valor de padeo. Equivalente a 03. - 01 usa el valor de padeo. - 02 hasta encontrar el primer dígito hexadecimal. - 03 el último carácter del bloque descifrado es el de padeo. - 04 – 16 longitud delpin indicada por su valor.
LPN	N	2	Longitud del siguiente campo.
PAN	N	var	PAN.
PAD	H	1	(3*)Valor de padeo.
LPIN	N	1	(4*)Longitud del PIN.
LPIF	N	1	(5*)Longitud del siguiente campo.
PIF	H	var	(5*)PIN Offset.
D2	H	1	(6*)Indicador de clave.
D3	H	3	(6*)Campo de validación.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
VPIN	N	1	Resultado de la verificación de PIN: - 0 correcto. - 1 incorrecto.

(*) Sólo si UKP = 1.

(2*) Sólo si F= 4.

(3*) Sólo si F = 4 y PEM = 00 ó PEM = 01.

(4*) Sólo si ALG = 0.

(5*) Sólo si ALG = 1 ó ALG = 6.

(6*) Sólo si ALG =5.

3.4.4. Verificación de PIN proveniente del TPV - 0606 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0606.
K1	K	var	Clave de cálculo de PIN (bajo LMK4).
K2	K	var	Clave de cifrado de PIN (bajo LMK3).
ND	N	1	(*) Número de datos de derivación de K2 (0..9).
DD1	H	16	(*) Dato de derivación 1.
…	…	…	…
DDN	H	16	(*) Dato de derivación nd.
TD	N	16	Tabla de decimalización.
F	N	1	Formato de bloque de PIN: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.
ALG	N	1	Algoritmo de generación de PIN: - 0 IBM3624. - 1 IBM3624 PIN Offset. - 3 Visa PVV. - 4 IBM German Bank Pool Institution. - 5 Interbank. - 6 ALT_1 PIN Offset.
UKP	N	1	Indicador de UKPT: - 0 no se usa. - 1 se usa.
PIN	H	16	Bloque de PIN.
DV	H	16	Dato de Validación.
D1	H	20	(2*)Current Key Sequence Number.
PEM	N	2	(3*)PIN Extraction Method: - 00 usa el valor de padeo. - 01 usa el valor de padeo. - 02 hasta encontrar el primer dígito hexadecimal. - 03 el último carácter del bloque descifrado es el de padeo. - 04 – 16 longitud de padeo indicada por su valor.
LPN	N	2	(4*) Longitud del siguiente campo.
PAN	N	var	(4*) PAN.
PAD	H	1	(5*)Valor de padeo.
LPIN	N	1	(6*)Longitud del PIN.
LPIF	N	1	(7*)Longitud del siguiente campo.
PIF	H	var	(7*)PIN Offset.
D2	H	1	(8*)Indicador de clave.
D3	H	3	(8*)Campo de validación.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
VPIN	N	1	Resultado de la verificación de PIN: - 0 correcto. - 1 incorrecto.

(*) Tantos campos dato de derivación como número de datos indique el campo ND.

(2*) Sólo si UKP = 1.

(3*) Sólo si F= 4.

(4*) Sólo si F= 0 ó F=3.

(5*) Sólo si F = 4 y PEM = 00 ó PEM = 01.

(6*) Sólo si ALG = 0.

(7*) Sólo si ALG = 1, ALG = 3 ó ALG=6.

(8*) Sólo si ALG =5.

3.4.5. Verificación de PVV - 0607 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0607.
K1	K	var	Clave de cálculo de PIN (bajo LMK4).
K2	K	var	Clave de cifrado de PIN (bajo LMK3).
F	N	1	Formato de bloque de PIN: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.
PIN	H	16	Bloque de PIN.
PEM	N	2	(*)PIN Extraction Method: - 00 usa el valor de padeo. - 01 usa el valor de padeo. - 02 hasta encontrar el primer dígito hexadecimal. - 03 el último carácter del bloque descifrado es el de padeo. - 04 – 16 longitud de padeo indicada por su valor.
LPN	N	2	Longitud del siguiente campo.
PAN	N	var	PAN.
PAD	H	1	(2*)Valor de padeo.
PVK	N	1	Dígito de índice de clave utilizado.
PVV	H	4	PVV

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
VPIN	N	1	Resultado de la verificación de PVV: - 0 correcto. - 1 incorrecto.

(*) Sólo si F= 4.

(2*) Sólo si F = 4 y PEM = 00 ó PEM = 01.

3.4.6. Carga Lista de PINes débiles - 0612 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0607.
LPIN	N	2	Longitud del PIN
NPIN	N	4	Número de PINes a incluir a continuación.
TPIN	N	var	Lista con los PINes débiles a cargar en el HSM

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.

3.5. Protección PIN en Intercambio: TDES (*CryptosecBANKING)

3.5.1. Gestión de PIN - 0603 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0603.
K1	K	var	Clave de Transporte de PIN del ATM (bajo LMK3).
K2	K	var	Clave de Cifrado de Bloque de PIN (bajo LMK3).
F1	N	1	Formato de bloque de PIN entrante: - 0. ISO0. - 1. ISO1. - 2. ISO2. - 3. ISO3. - 4. IBM3624. - 5. Diebold
F2	N	1	Formato de bloque de PIN saliente: - 0. ISO0. - 1. ISO1. - 2. ISO2. - 3. ISO3. - 4. IBM3624. - 5. Diebold
PIN	H	16	Bloque de PIN cifrado, recibido del ATM.
L1	N	2	Longitud del siguiente campo.
D1	N	var	PAN.

Dato	Tipo	Longitud	Comentario
L2	N	2	(*)Longitud del siguiente campo.
D2	H	var	(*)Número aleatorio usado para formar el bloque de PIN saliente.
D3	H	1	(2*)Valor de padeo.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
PIN	H	16	Bloque de PIN cifrado.

(*) Sólo si F2=1 o F2 = 3. Longitud máxima de D2 es 16.

(2*) Sólo si F2 = 4.

3.5.2. Gestión de PIN proveniente de TPVs - 0610 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0610.
K1	K	var	Clave Maestra de Transporte de PIN(bajo LMK3).
K2	K	var	Clave de Cifrado de Bloque de PIN (bajo LMK3).
F1	N	1	Formato de bloque de PIN entrante: - 0. ISO0. - 1. ISO1. - 2. ISO2. - 3. ISO3. - 4. IBM3624. - 5. Diebold
F2	N	1	Formato de bloque de PIN saliente: - 0. ISO0. - 1. ISO1. - 2. ISO2. - 3. ISO3. - 4. IBM3624. - 5. Diebold.
PIN	H	16	Bloque de PIN cifrado, recibido del ATM.
L1	N	2	Longitud del siguiente campo.
D1	N	var	PAN.
L2	N	2	(*)Longitud del siguiente campo.
D2	H	var	(*)Número aleatorio usado para formar el bloque de PIN saliente.
D3	H	1	(2*)Valor de padeo.
ND	N	1	Número de datos de derivación de K1 (0..9).
DD1	H	16	Dato de derivación 1.
…	…	…	…
DDN	H	16	Dato de derivación nd.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
PIN	H	16	Bloque de PIN cifrado.

(*) Sólo si F2=1 o F2 = 3. Longitud máxima de D2 es 16.

(2*) Sólo si F2 = 4.

3.6. Cálculo de Offset - 0604 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0604.
K1	K	var	Clave de generación de PIN (bajo LMK4).
K2	K	var	Clave de transporte de PIN (bajo LMK3).
TD	N	16	Tabla de decimalización
F	N	1	Formato de bloque de PIN entrante: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.
ALG	N	1	Algoritmo de generación de PIN: - 1 IBM3624-PIN Offset. - 2 Netherlands PIN 1. - 4 IBM German Bank Pool Institution.
PIN	H	16	Bloque de PIN cifrado.
L1	N	2	Longitud del siguiente campo.
PAN	N	var	PAN.
PAD	H	2	(*)Padeo.
LPIN	N	1	(2*)Longitud del PIN.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
OFFSET	N	4	Valor offset devuelto.

(*) Sólo si F = 4.

(2*) Sólo si ALG = 1.

3.7. Cálculo de PVV - 0608 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0608.
K1	K	var	Clave de generación de PIN (bajo LMK4).
K2	K	var	Clave de transporte de PIN (bajo LMK3).
F	N	1	Formato de bloque de PIN entrante: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.

Dato	Tipo	Longitud	Comentario
PIN	H	16	Bloque de PIN cifrado.
L1	N	2	Longitud del siguiente campo.
PAN	N	var	PAN.
PAD	H	2	(*)Padeo.
PVK	N	1	Dígito de índice de clave a usar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
OFFSET	N	4	Valor PVV devuelto.

(*) Sólo si F = 4.

3.8. Exportación de Pines(*CryptosecBANKING)

3.8.1. Exportación de PIN - 0605 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0605.
K1	K	var	Clave de cálculo de PIN (bajo LMK4).
K2	K	var	Clave de exportación de PIN (bajo LMK3).
TD	N	16	Tabla de decimalización.
F	N	1	Formato de bloque de PIN saliente: - 0 formato ISO0. - 3 formato ISO3.
LPN	N	2	Longitud del siguiente campo.
PAN	N	var	PAN.
LPIF	N	2	Longitud del siguiente campo, hasta 12.
OFF	N	var	Offset.
D1	H	10	(*)Número aleatorio para formar el bloque de PIN.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
PIN	H	16	Bloque de PIN cifrado.

(*) Sólo si F = 3.

3.8.2. Impresión de PIN - 0609 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0609.
K1	K	var	Clave de Cifrado de Bloque de PIN (bajo LMK3).
F1	N	1	Formato de bloque de PIN entrante: - 0. ISO0. - 1. ISO1. - 2. ISO2. - 3. ISO3. - 4. IBM3624. - 5. Diebold.
PIN	H	16	Bloque de PIN cifrado.
L1	N	2	Longitud del siguiente campo.
D1	N	var	PAN.
L2	N	4	Longitud del siguiente campo en bytes.
T	A	var	Tabla de datos variables de usuario (nombre, custodio, cajero, fechas, finalidad, clave,…). El formato dependerá de la definición del formato de impresión.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.

Notas sobre la impresión

Es necesario tener previamente definida la cadena de formato de impresión, mediante un comando 1301.

En la impresión de pines, el argumento T contiene las cadenas de impresión 0 a 15 que se referencian en la cadena de formatos de impresión. No es preciso definir las 16 cadenas, sin embargo, hay que tener en cuenta que estas se numeran de forma correlativa, comenzando por cero. La secuencia "\0" se usa como delimitador de cadenas. El símbolo ^P induce la impresión del PIN.

Para aclarar el funcionamiento de la impresión de pines, supóngase que se quiere enviar su PIN al señor Thomas M Smith. Se pretende imprimir tanto la dirección del cliente como su PIN, en la forma siguiente:

THOMAS M SMITH

APT 4B XXXX

39 ELM DR

MEDIA PA 19063

YOUR FULL SERVICE BANK

Donde XXXX es el PIN, y se ha añadido un mensaje corporativo final. Por otro lado, habrá que enviar información análoga a otros clientes. Puede considerarse que el formato, esto es, la posición de los distintos campos de impresión, es común a todos los clientes. Igualmente, el mensaje corporativo es también común a toda la serie de impresiones. La forma más cómoda de resolver todo esto será incluir el mensaje corporativo en la cadena de formato. (Sin embargo, es posible también incluirlo como un campo más de impresión si se prefiere por razones de espacio o de otra índole, ver comando de exportación de componentes, 3.1.12). La cadena de formato será en este caso:

>L>013^0>L>013^1>041^P>L>013^2>L>013^3>L>013YOUR FULL SERVICE BANK>L>F

Donde puede verse como ^0 denota el punto de inclusión de la primera cadena de la tabla de datos, ^1 denota el punto de inclusión de la segunda cadena de la tabla de datos y así sucesivamente. ^P denota el punto de inclusión del PIN.

Con esta cadena de formato, para conseguir la impresión buscada, la tabla de datos a incluir en el presente comando será:

THOMAS M SMITH\0APT 4B\039 ELM DR\0MEDIA PA 19063

Se ve que el \0 se usa como separador, y que no es necesario incluirlo detrás de la última cadena. La longitud de esta cadena es 00000049.

La alternativa de situar el mensaje corporativo fuera de la cadena de formato, haría que éste se sustituyese por ^4 en dicha cadena y se incorporase a la tabla de datos del comando de impresión, para cada una de las impresiones.

3.9. Cálculo de Códigos de Validación (*CryptosecBANKING)

3.9.1. Cálculo y Verificación de Códigos de Validación - 0701 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0701.
K	K	var	Clave de CVV/CVC/CSS (bajo LMK5).
D_1	N	4	Fecha de Caducidad.
D_2	N	3	Código de Servicio.
PAN	N	16	PAN.
M	N	1	Modo de funcionamiento: - 0 cálculo. - 1 Verificación.
CVV	N	3	(*)CVV/CVC/CSS a verificar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
CVV	N	3	(2*)Código de validación calculado.
VER	N	1	(*)Verificación de CVV/CVC/CSS: - 0 correcto. - 1 incorrecto.

(*) Sólo si M = 1.

(2*) Sólo si M = 0.

3.9.2. Cálculo y Verificación de CVC 3 – 0702 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0702.
K1	K	var	Clave de CVC3 (bajo LMK 5)
M	N	1	Indicador de modo de funcionamiento: - 0 Cálculo. - 1 Verificación.
D1	H	16	Datos de cálculo 1.
D2	H	12	Datos de cálculo 2.
L3	N	2	Longitud del siguiente campo.
D3	H	var	Datos de cálculo 3.

Dato	Tipo	Longitud	Comentario
CVC3	N	3	(*)CVC3 a verificar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
CVC3	H	4	(2*)CVC3 calculado.
VER	N	1	(*)Verificación de CVC 3: - 0 correcto. - 1 incorrecto.

(*) Sólo si M = 1.

(2*) Sólo si M = 0.

3.9.3. Cálculo y Verificación de CSC – 0703 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0703.
K1	K	var	Clave de CSC (bajo LMK 5)
M	N	1	Indicador de modo de funcionamiento: - 0 Cálculo. - 1 Verificación.
PAN	N	16	PAN
D1	N	4	Fecha de caducidad
L2	N	2	(*)Longitud del siguiente campo.
CSC	N	var	(*)CSC a verificar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
VER	N	1	(*)Verificación de CSC: - 0 correcto. - 1 incorrecto.
CSC3	H	3	(2*)CSC3 calculado.
CSC4	H	4	(2*)CSC4 calculado.
CSC5	H	5	(2*)CSC5 calculado.

(*) Sólo si M = 1.

(2*) Sólo si M = 0.

3.10. Securización de Mensajes

3.10.1. Securización de Mensajes – 0801 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0801.
K1	K	var	Clave de MAC (bajo LMK 8).
I1	N	1	Indicador del algoritmo a usar: - 0 MAC ANSI X9.9-1.

Dato	Tipo	Longitud	Comentario
			- 1 MAC ANSI X9.19-1. - 2 MAC TDES.
I2	N	1	Indicador del relleno a utilizar: - 0 método 1 de ISO/IEC 9797. - 1 método 2 de ISO/IEC 9797.
I3	N	1	Indicador si necesario Hash previo: - 0 No Hash. - 1 SHA-1. - 2 MD5.
IV	H	16	Vector de Inicialización.
L1	N	6	Longitud del siguiente campo.
D1	H	var	Datos a firmar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
MAC	H	16	MAC calculado.

3.10.2. Securización de Mensajes del TPV – 0802 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0802.
K1	K	var	Clave de MAC (bajo LMK 8).
ND	N	1	(*) Número de datos de derivación de K1 (0..9).
DD1	H	16	(*) Dato de derivación 1.
…	…	…	…
DDND	H	16	(*) Dato de derivación nd.
I1	N	1	Indicador del algoritmo a usar: - 0 MAC ANSI X9.9-1. - 1 MAC ANSI X9.19-1. - 2 MAC TDES.
I2	N	1	Indicador del relleno a utilizar: - 0 método 1 de ISO/IEC 9797. - 1 método 2 de ISO/IEC 9797.
I3	N	1	Indicador si necesario Hash previo: - 0 No Hash. - 1 SHA-1. - 2 MD5.
IV	H	16	Vector de Inicialización.
L1	N	6	Longitud del siguiente campo.
D1	H	var	Datos de MAC.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
MAC	H	16	MAC calculado.

(*) Tantos campos dato de derivación como número de datos indique el campo ND.

3.10.3. HMAC – 0803 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0803.
K1	K	var	Clave de MAC (bajo LMK 8).
I1	N	1	Indicador del algoritmo de hash a usar: - 0 MD5. - 1 SHA-1. - 2 SHA-256.
L1	N	6	Longitud del siguiente campo.
D1	H	var	Datos.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
HMAC	H	(*)var	HMAC calculado.

(*) Es función del algoritmo de hash empleado:

- MD5: 32.

- SHA-1: 40.

- ShA-256: 64.

3.11. Cifrado y Descifrado de Datos

3.11.1. Cifrado y Descifrado con DES - 0901 -

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0901.
K1	K	var	Clave a utilizar (bajo LMK 7).
I1	N	1	Indicador de función: - 1 Cifrar. - 0 Descifrar.
I2	N	1	Algoritmo a utilizar: - 1 ECB. - 0 CBC (vector de inicialización a ceros).
L1	N	6	Longitud del siguiente campo.

Dato	Tipo	Longitud	Comentario
D1	H	var	Datos a cifrar/descifrar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
LCIF	N	6	Longitud del siguiente campo.
CIF	H	var	Datos cifrados/descifrados.

3.11.2. Cifrado y Descifrado con DES v2 - 0902 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0902.
K1	K	var	Clave a utilizar (bajo LMK 7).
ND	N	1	(*) Número de datos de derivación de K1 (0..9).
DD1	H	16	(*) Dato de derivación 1.
…	…	…	…
DDND	H	16	(*) Dato de derivación nd.
I1	N	1	Indicador de función: - 1 Cifrar. - 0 Descifrar.
I2	N	1	Algoritmo a utilizar: - 1 ECB. - 0 CBC.
IV	H	16	(2*) Vector de inicialización.
L1	N	6	Longitud del siguiente campo.
D1	H	var	Datos a cifrar/descifrar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
LCIF	N	6	Longitud del siguiente campo.
CIF	H	var	Datos cifrados/descifrados.

(*) Tantos campos dato de derivación como número de datos indique el campo ND.

(2*) Sólo si I2=0.

3.11.3. Cifrado y Descifrado con DES v3 - 0903 -

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0903.
K1	K	var	Clave a utilizar (bajo LMK 7).
I1	N	1	Indicador de función: - 0Descifrar. - 1 Cifrar. - 2 Descifrar binario. - 3 Cifrar binario.
I2	N	1	Algoritmo a utilizar: - 1 ECB. - 0 CBC.
IV	H	16	(*)Vector de inicialización.
L1	N	6	(2*)Longitud del siguiente campo.
D1	H/B	var	(3*)Datos a cifrar/descifrar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
LCIF	N	6	Longitud del siguiente campo.
CIF	H	var	Datos cifrados/descifrados.

(*) Sólo si I2 = 0.

(2*)Si I1=0 ó 1, la longitud de los datos a enviar debe ser múltiplo de 16.

Si I1=2 ó 3, la longitud de los datos a enviar debe ser múltiplo de 8.

(3*)Si I1=0 ó 1, los datos deben ser hexadecimales (descomprimidos) y se comprimirán antes de cifrar/descifrar.

Si I1=2 ó 3, los datos pueden ir en binario y se cifrarán/descifrarán tal y como se envíen.

3.11.4. Securizar Password - 0904 –

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0904.
K	K	var	Clave de cifrado (bajo LMK 17).
LEN	N	2	Longitud de los datos siguientes.
P	B	var	Password a proteger.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
LEN	N	2	Longitud de los datos siguientes.
PP	H	var	Password protegido.

3.11.5. Verificar Password - 0905 –

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 0905.
K	K	var	Clave de cifrado (bajo LMK 17).
L1	N	2	Longitud de los datos siguientes.
P	B	var	Password a verificar.
N1	N	2	Numero de passwords protegidos que se van a pasar.
LP1	N	2	Longitud del password #1
PP1	H	var	Password protegido #1.
…	….	…..	……..
LPN	N	2	Longitud del password #N.
PPN	H	var	Password protegido #N.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
VER	N	1	Resultado de la validación: - 1 correcto (al menos unos de los password es correcto) - 0 incorrecto(ningún password se ha verificado)

3.11.6. Cifrado y Descifrado con RSA - 1001 -

Para nuevas implementaciones se sugiere utilizar el comando 1002.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1001.
K1	K	var	Clave a utilizar (bajo LMK 60 o 62).
I2	N	1	Indicador de función: - 0 Cifrar con clave pública. - 1 Descifrar con clave privada.
L1	N	6	Longitud del siguiente campo.
D1	H	var	Datos a cifrar/descifrar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
LCIF	N	6	Longitud del siguiente campo.
CIF	H	var	Datos cifrados/descifrados.

3.11.7. Cifrado y Descifrado con RSA v2 – 1002 -

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1002.
K1	K	var	Clave a utilizar (bajo LMK 60 o 62).
I1	N	1	Indicador de función: - 0 Cifrar con clave pública. - 1 Descifrar con clave privada.
I2	N	1	Identificador de algoritmo - 0 PKCS#1 v1.5. - 1 PKCS#1 OAEP.
L1	N	6	Longitud del siguiente campo.
D1	H	var	Datos a cifrar/descifrar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
LCIF	N	6	Longitud del siguiente campo.
CIF	H	var	Datos cifrados/descifrados.

3.12. Firma y Verificación Datos

3.12.1. Firma y Verificación con RSA – 1101 -

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1101.
K1	K	var	Clave a utilizar (bajo LMK 61 o 62).
M	N	1	Indicador de función: - 0 Firmar con clave privada. - 1 Verificar con clave pública.
AL1	N	1	(2*)Indicador de función HASH: - 0 SHA1. - 1 MD5
AL2	N	1	(2*)Indicador de algoritmo de firma: - 0 pkcs#1 v1.5. - 1 X9.31.
L1	N	6	Longitud del siguiente campo.
D1	H	var	Datos a firmar o verificar.
L1	N	6	(*)Longitud del siguiente campo.
D1	H	var	(*)Firma.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
LFIR	N	6	(2*)Longitud del siguiente campo.
FIR	H	var	(2*)Datos firmados.
VER	N	1	(*)Resultado de la verificación: - 0 – correcto. - 1 - incorrecto.

(*) Sólo si M = 1.

(2*) Sólo si M = 0.

3.12.2. Firma con Clave Asimétrica – 1103 -

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1103.
K1	K	var	Clave a utilizar (bajo LMK 61 o 62).
MEC	H	2	Mecanismo de hash: - 00 No Hash

Dato	Tipo	Longitud	Comentario
			- 01 MD5. - 10 SHA-1 - 11 SHA-224 - 12 SHA-256 - 13 SHA-386 - 14 SHA-512
ALG	H	2	Mecanismo de firma: - 01 RSA.
PAD	N	2	Indicador de padeo de bloque a firmar - 00Raw - 01 pkcs#1 v1.5. - 10 X9.31.
L1	N	6	Longitud del siguiente campo.
D1	H	var	Datos a firmar ya resumidos según el mecanismo especificado en MEC

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
LFIR	N	6	Longitud del siguiente campo.
FIR	H	var	Firma

3.12.3. Verificación con Clave Asimétrica – 1104 -

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1104.
K1	K	var	Clave a utilizar (bajo LMK 61 o 62).
MEC	H	2	Mecanismo de hash utilizado: - 00 No Hash - 01 MD5. - 10 SHA-1 - 11 SHA-224 - 12 SHA-256 - 13 SHA-386 - 14 SHA-512
MEC	H	2	Mecanismo de firma: - 01 RSA.
AL2	N	2	Indicador de padeo de bloque a firmar - 00 Raw - 01 pkcs#1 v1.5. - 10 X9.31.
L1	N	6	Longitud del siguiente campo.
D1	H	var	Datos a firmados ya resumidos según el mecanismo especificado en MEC
L1	N	6	Longitud del siguiente campo.
D1	H	var	Firma

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
VER	N	1	Resultado de la verificación: - 0 – correcto. - 1 - incorrecto.

3.13. Monedero y Transporte para TIBC o Advantis

3.13.1. Generación de Certificado de Monedero y Transporte– 1201 - (*CryptosecBANKING)

Mensaje de Petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1201.
M	N	1	Acción a realizar: 0 Genera certificado para monederos TIBC o Advantis. 1 Genera certificado para abono transporte TIBC o Advantis (pasada 2). 2 Genera certificado para abono transporte Advantis (pasada 1).
B0	H	16	Bloque de datos (Bloque0).
B1	H	16	Bloque de datos (Bloque1).
B2	H	16	(*)Bloque de datos (Bloque2).
B3	H	16	(2*)Bloque de datos (Bloque 3).
Kc	K	var	Clave de carga.
PAN	N	16	Número de la tarjeta.
ID	N	4	Identificador del fichero de carga.
RN1	H	16	(3*)Número aleatorio 1.
RN2	H	16	Número aleatorio 2.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
CC	H	16	Certificado calculado.

(*) Sólo si M=0 o M=1

(2*) Sólo si M=1

(3*) Sólo si M=0

3.14. Carga de la Cadena de Formato de Impresión – 1301 -

Mensaje de Petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1301.
LEN	N	3	Longitud de la cadena de formato de impresión. Hasta 400 caracteres.
STR	H	var	Cadena de formato de impresión.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.

3.15. Testeo del HSM

3.15.1. Testeo del HSM – 1401 -

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1401.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
KCV	N	6	Tres primeros bytes del identificador del HSM.

3.16. Diversificación de Clave

3.16.1. Diversificación de Clave – 1601 -

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1601.
Kd	K	var	(*) Clave a diversificar (bajo LMK correspondiente)
I_0	N	1	(2*)Comprobación de la paridad de la clave: - 0 No comprobar paridad. - 1 Comprobar paridad.
I_1	N	2	Identificador del algoritmo de diversificación: - 12 TDES II. - 22 Claves SIM.
LMK	N	2	(3*)Número de LMK bajo la que saldrá cifrada la clave.
I_2	N	1	(4*)Indicador de ajuste de paridad de clave diversificada:

Dato	Tipo	Longitud	Comentario
			- 0 No ajustar paridad de clave diversifica. - 1 Ajustar paridad de clave diversificada.
L_1	N	2	(6*)Longitud dato diversificador 1.
D_1	H	N	(6*)Dato diversificador 1.
L_2	N	2	(6*)Longitud del dato diversificador 2.
D_2	H	N	(6*)Dato diversificador 2.
ICCID	H	20	(7*)ICCID de la SIM.
F_1	A	1	(5*)Indicador de almacenamiento interno de clave diversificada: - S.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
Kr	K	var	Clave diversificada.

- (*) Clave a diversificar. Sólo se pueden diversificar claves que estén cifradas bajo LMK 2, 3, 6, 7, 8, 9, 10, 14 y 99.
- (2*) Solo se podrá dar la opción de comprobar la paridad de la clave a diversificar si está cifrada bajo LMK09 o LMK99.
- (3*) Las claves diversificadas sólo podrán estar cifradas bajo las LMK 2, 3, 6, 7, 8, 9, 10, 14 y 99. Además solo si la clave a diversificar es de tipo 99, la diversificada podrá ser del mismo tipo.
- (4*) Este indicador sólo estará presente si la clave diversificada está cifrada bajo LMK09 o LMK99.
- (5*) Este indicador sólo estará presente si la clave diversificada se debe almacenar internamente.
- (6*) Sólo para identificador de algoritmo de diversificación 12.
- (7*) Sólo para identificador de algoritmo de diversificación 22.

3.17. Números Aleatorios

3.17.1. Generación de Número Aleatorio – 1701 -

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 1701.
LEN	N	6	Longitud en bits del número aleatorio a generar.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
LEN	N	6	Longitud de los datos siguientes.
RND	H	var	Número aleatorio generado.

3.18. Función resumen (HASH)

3.18.1. Hash – 2000 -

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 2000.
MEC	H	2	Mecanismo de hash: - 01 MD5. - 10 SHA-1 - 11 SHA-224 - 12 SHA-256 - 13 SHA-386 - 14 SHA-512
FLAG	N	2	Flag de estado del hash: - 01 Init. - 02 Update - 03 Final
F1	N	2	Formato de los datos a hacer el hash: - 01 Hexadecimal. - 02 Binario
L1	N	6	Longitud del siguiente campo. (*)
D1	H	var	Datos intermedios (estado). (*)
L2	N	6	Longitud del siguiente campo.
D2	H	var	Datos a hacer el hash.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
FR1	H	2	Formato de los datos a recibir: - 01 Hexadecimal. - 02 Binario.
LR1	N	6	Longitud de los datos siguientes. (2*)
DR1		var	Datos intermedios (estado). (2*)
LR2	N	6	Longitud de los datos siguientes. (3*)
DR2		var	Hash. (3*)

(*) Sólo si FLAG = 02 ó 03.

(2*) Sólo si FLAG = 01 ó 02

(3*) Sólo si FLAG = 03

3.19. Claves de TPV

3.19.1. Petición de claves de TPV – 6001 - (*CryptosecBANKING)

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 6001.
MOD	N	1	Modo de operación para la CTC: - 0 Inicializar CTC - 1 Renovar CTC
K1	K	var	Clave Maestra de inicialización. (*)
K2	K	var	Clave de transporte antigua. (2*)
NKM	N	1	Número de claves a pedir (1..5) (3*)
OPMn	N	1	Modo de generación: - 0 Derivar - 1 Aleatoria
KMn	K	Var	Clave Maestra a derivar. (4*)
LMKn	N	2	Etiqueta LMK asociada a la clave a pedir. Deberá ser distinto de 00.(5*)
SPP	H	32	Número de serie del PINPAD (6*)
NKS	N	1	Número de claves CPE a pedir (0..2) (7*).
OPSn	N	1	Modo de generación: - 0 Derivar - 1 Aleatoria
KSn	K	Var	Clave Maestra a derivar. (8*)
SSM	H	32	Número de serie del SAM (8*)

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
NK	N	2	Número de claves
Kn	K	var	Clave de n generada (9*)

(*) Sólo presente si MOD=0

(2*) Sólo presente si MOD=1

(3*) Número de claves a solicitar. Al menos se solicitará la petición de la CTC.

En este grupo se pueden pedir:

- Clave de transporte de claves (CTC)
- Clave de transporte de PIN (CP)
- Clave de MAC (CM)

- Clave de telecarga de software (CAT)
- Clave de cifrado de datos (CAU)

Deben existir tantos campos OPMn, y KMn o LMKn como número de claves se haya indicado y SIEMPRE deberá ser una CTC la primera.

(4*) Sólo presente si OPMn=0

(5*) Sólo presente si OPMn=1

(6*) Sólo presente si MOD=0 y/u OPMn=0

(7*) Número de claves a solicitar. En este grupo se pueden pedir:

- Clave de transporte de PINoffset entre PINPAD y tarjeta (CPE1 y CPE2)

Deben existir tantos campos OPSn y KSn como número de claves se haya indicado.

(8*) Sólo presente si OPSn=0

(9*) Tantos bloque se claves como el campo NK indique y en el orden en el que se pidieron.

3.20. Tablas de decimalización

3.20.1. Actualización de tabla de decimalización - 2101 - (*CryptosecBANKING)

Permite recifrar una tabla de decimalización (TD en adelante) que viene cifrada bajo una LMK del conjunto antiguo a la misma LMK del conjunto actual. De esta manera es posible actualizar, cada vez que se modifique la clave maestra del sistema, todas las TD que tengamos almacenadas externamente.

Para que este comando funcione correctamente, hay que tener en cuenta:

- Haber cargado una nueva clave maestra del sistema.
- La TD a actualizar debe haber sido almacenada bajo una LMK asociada a la citada clave maestra anterior: el proceso de actualización es sucesivo, ya que sólo se almacena con tal fin el conjunto de LMKs más recientemente sustituidas.
- No haber eliminado dichas LMKs previamente a la ejecución de este comando. Una vez eliminadas, sólo será posible recuperar las TD cifradas bajo ese conjunto de claves volviendo a cargar en el sistema la clave maestra asociada.

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 2101.
TD	H	16	Tabla de decimalización cifrada con LMK antigua.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
TD	H	16	Tabla de decimalización cifrada con LMK nueva.

3.21. Administrador de logs

3.21.1. Envío de logs – 2201 –

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 2201.
A	N	8	Fecha de la que se quiere el log (AAAAMMDD)
S	N	8	Índice de línea a partir de la que se enviará el log.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
N	N	4	Número de líneas a devolver.
F	L	8	Longitud de la línea.
D	A	var	Línea del log
RES	N	8	(*)Índice de la última línea enviada.

(*) Este valor es distinto de cero si la longitud total de la respuesta está próxima a 512KB, es decir, si no se ha podido enviar desde la línea indicada hasta el final del fichero.

3.21.2. Índice de logs – 2202 –

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 2202.

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
N	N	4	Número de ficheros de logs encontrados.
D	A	var	Ficheros de logs encontrados (AAAAMMDD).

3.21.3. Borrar fichero de logs – 2203 –

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 2203.
A	N	8	Fecha del log que se quiere borrar (AAAAMMDD)

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.

3.22. Administrador Certificados X509

3.22.1. Generación de request (PKCS#10) – 2301 –

Mensaje de petición:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador del comando. Valor 2301.
L	N	4	Indicador longitud en bits de la clave a generar: - Debe ser de 1024 ó 2048 bits.
E	N	1	(*)Exponente de la clave RSA a generar: - 0 si el exponente es 3. - 1 si es Fermat4.
I	N	2	Número de atributos del subject.
LO1	N	2	Longitud del OID del atributo #1 (*)
O1	A	var	OID del atributo #1 (*)
LN1	N	2	Longitud del short name del OID del atributo #1
N1	A	var	Short Name del OID del atributo #1
LV1	N	2	Longitud del valor del atributo #1
V1	A	var	Valor del atributo #1
…	…	….	……
LON	N	2	Longitud del OID del atributo #n
ON	A	var	OID del atributo #n
LNN	N	2	Longitud del short name del OID del atributo #n
NN	A	var	Short Name del OID del atributo #n
LVN	N	2	Longitud del valor del atributo #n
VN	A	var	Valor del atributo #n
LPASS	N	2	Longitud del passwordChallenge
PASS	A	var	PasswordChallenge
LPK	N	2	Longitud del campo siguiente
LABPK	A	var	Etiqueta clave pública
LPRK	N	2	Longitud del campo siguiente
LABPRK	A	var	Etiqueta Clave privada
OF	N	1	Formato de salida de la request: - 0 si DER - 1 si PEM

Mensaje de respuesta:

Dato	Tipo	Longitud	Comentario
IC	H	4	Identificador de comando.
RV	H	8	Valor de retorno.
L1	N	4	Lognitud de la request
R1	B	var	Request generada

(*) Este valor se informará en el caso de ser un OID nuevo o propietario. En el caso de existir, sólo se informará que la longitud de este campo es cero (00).

4. CODIGOS DE ERROR

Los códigos de error devueltos por el HSM a la hora de realizar un comando son los mostrados en la siguiente tabla:

Código	Descripción
0x00000000	OK
0x00000001	MESSAGE FORMAT ERROR
0x00000002	SERVICE UNAVAILABLE
0x00000005	PC MEMORY ERROR
0x00000018	TIME OUT
0x00000019	INVALID FIRMWARE
0x0000001A	DRIVER ERROR
0x0000001B	ARGUMENT ERROR
0x0000001C	OPEN WD ERROR
0x0000001D	INCORRECT WD VERSION
0x00000023	NO FIRMWARE
0x00000025	DATA MISSALIGNMENT
0x00000026	CIPHER SESSION NOT INITIALIZED
0x00013000	HASH ALGORITHM NOT SUPPORTED
0x00013400	SIGNATURE FORMAT NOT SUPPORTED
0x00001000	ERROR TEST HASH SHA1
0x00001400	ERROR TEST HASH MD5
0x00001800	ERROR TEST HASH RIPEMD 128
0x00001C00	ERROR TEST HASH RIPEMD 160
0x00002000	ERROR DES ECB S
0x00002400	ERROR DES ECB D
0x00002800	ERROR DES ECB T
0x00002C00	ERROR DES CBC S
0x00003000	ERROR DES CBC D
0x00003400	ERROR DES CBC T
0x00003800	ERROR DES CFB64 S
0x00003C00	ERROR DES CFB64 D
0x00004000	ERROR DES CFB64 T
0x00004400	ERROR DES OFB64 S
0x00004800	ERROR DES OFB64 D
0x00004C00	ERROR DES OFB64 T
0x00006400	PAIR WISE CONSISTENCE TEST ERROR
0x00006800	SIGN VERIFY TEST ERROR
0x00007C00	WRONG NUMBER OF CUSTODIANS
0x00008C00	DEFAULT PIN ERROR

Código	Descripción
0x00009000	PIN SIZE ERROR
0x00009800	USER EXISTS ERROR
0x00009C00	MEMORY USER FULL ERROR
0x0000A000	USER NOT EXISTS ERROR
0x0000A800	INVALID LEN DES KEY ERROR
0x0000AC00	CONFIG BCHU ERROR
0x0000B000	INVALID DES KEY ERROR
0x0000B400	RSA KEYS GENERATION ERROR
0x0000B800	INVALID DATA SIZE ERROR
0x0000BC00	PUBLIC KEY CIPHER ERROR
0x0000C000	WRONG ALGORITHM SELECTED ERROR
0x0000C400	PRIVATE KEY CIPHER ERROR
0x0000CC00	KEY NOT EXISTS
0x0000D400	INVALID KEY NUMBER
0x0000E000	WRONG COMMAND ERROR
0x0000EC00	LEN RSA KEY ERROR
0x0000F000	PASSWORD ERROR
0x00010000	WRONG PAD CHARACTER ERROR
0x00010400	WRONG PINBLOCK ERROR
0x00010800	INVALID LEN PIN ERROR
0x00010C00	TOO MANY ONES ERROR
0x00011000	INVALID LEN PAN ERROR
0x00011400	INVALID LEN CVV ERROR
0x00011800	WRONG PAN ERROR
0x00011C00	WRONG PARAMETER ERROR
0x00012000	WRONG PINBLOCK FORMAT ERROR
0x00012400	PASSWORD SIZE ERROR
0x00012C00	ERROR RNG TEST
0x00013000	HASH ALGORITHM NOT SUPPORTED
0x00013400	SIGNATURE FORMAT NOT SUPPORTED
0x00015800	LNAU HARDWARE ERROR
0x00015C00	WRONG CONFIGURATION DATA
0x00016000	WRONG CMM KEY ERROR
0x00016400	WRONG OPTIONS INDICATOR ERROR
0x00016800	WRONG METHOD INDICATOR ERROR
0x00016C00	WRONG DINAMIC OBE ERROR
0x00017000	VERIFICATION FAILED
0x00017400	PRINT STRING TOO LONG ERROR
0x00017800	WRONG PRINT STRING ERROR
0x00017C00	WRONG PKCS8 INFO ERROR
0x00018000	WRONG DECIMALIZATION TABLE ERROR
0x00018400	NOT ALLOWED IN PRODUCTIONSTATE
0x00018800	INVALID LEN NA ERROR

Código	Descripción
0x00019000	PRINTER NOT ENABLED ERROR
0x00019400	OLD CMMS NOT EXIST ERROR
0x00019800	PIN VERIFICATION LOCKED
0x00019C00	PINBLOCK LOCKED
0x0001A000	NO LOG SPACE ERROR
0x0001A400	CARDACCESSERROR
0x0001A800	CARDTIMEOUTERROR
0x0001AC00	CARDAUTHENTICATIONERROR
0x0001B000	CARDPINERROR
0x0001B400	CARDCANNOTCHANGEPINERROR
0x0001B800	CARDSETERROR
0x0001BC00	REPEATEDCARDERROR
0x0001C000	BLANKCARDERROR
0x0001C400	CARDALREADYUPDATEDERROR
0x0001C800	EXCEPTION ERROR
0x0001CC00	INPUT DATA TIMEOUT ERROR
0x0001D000	OUTPUT DATA TIMEOUT ERROR
0x0001D400	HSM SERVING CONSOLE COMMAND ERROR

5. OPERATIVA DE LOS COMANDOS

5.1. Cálculo de DAC

DAC = (TDES(**K**) [**D1**])$_{2MSB}$

5.2. Cálculo de IDN

KIDN-IZQ = TDES (K) [D1]

KIDN-DER = TDES (K) [$\overline{\text{D1}}$]

KIDN = KIDN-IZQ || KIDN-DER

IDN =TDES (KIDN)[D2]

5.3. Cálculo y Verificación de ARQC y ARPC

KAC-IZQ = TDES (K) [D18LSB]

KAC-DER = TDES (K) [$\overline{\text{D18LSB}}$]

KAC = KAC-IZQ || KAC-DER

Para MasterCard:

SKAC-IZQ = TDES (KAC) [D2 || F000 || D3]

SKAC DER = TDES (KAC) [D2 || 0F00 || D3]

SKAC = SKAC-IZQ || SKAC-DER

Cálculo del ARQC para Visa:

ARQC = MAC(KAC) [D4]

Padeo de D4: D4 || 00...00

Cálculo del ARQC para MasterCard:

ARQC = MAC(SKAC) [D4]

Padeo de D4: D4 || 8000...00

Cálculo del ARPC:

ARPC = TDES(KAC) [D5 XOR D6]

5.4. Firma de Script

KSF-IZQ = TDES (K) [D18LSB]

KSF-DER = TDES (K) [$\overline{\text{D18LSB}}$]

KSF = KSF-IZQ || KSF-DER

Para Visa:

SKSF-IZQ = (KSF-IZQ) XOR (D2)

SKSF-DER = (KSF- DER) XOR (000000000000 || (D22LSB XOR FFFF)

SKSF = SKSF-IZQ || SKSF-DER

Para MasterCard:

SKSF-IZQ = TDES (KSF) [Bytes 8/7 D2 || F0 || Bytes 5/1 D2]

SKSF-DER = TDES (KSF) [Bytes 8/7 D2 || 0F || Bytes 5/1 D2]

SKSF = SKSF-IZQ || SKSF-DER

Firma:

F = MAC (SKSF) [D3]

Padeo de D3: D3 || 8000...00

5.5. Cifrado de Script.

KSC-IZQ=TDES (K) [D18LSB]

KSC-DER = TDES (K) [$\overline{\text{D18LSB}}$]

KSC = KSC-IZQ || KSC-DER

Para Visa:

SKSC-IZQ = (KSC-IZQ) XOR (D2)

SKSC-DER = (KSC- DER) XOR (000000000000 || (D22LSB XOR FFFF)

SKSC = SKSC-IZQ || SKSC-DER

Para MasterCard:

SKSC-IZQ = TDES (KSC) [Bytes 8/7 D2 || F0 || Bytes 5/1 D2]

SKSC-DER = TDES (KSC) [Bytes 8/7 D2 || 0F || Bytes 5/1 D2]

SKSC = SKSC-IZQ || SKSC-DER

Cifrado para Visa:

F = TDESECB (SKSC) [D3]

Padeo de D3: D3 || 8000...00

Cifrado para MasterCard:

F = TDESCBC (SKSC) [D3]

Padeo de D3: D3 || 8000...00

5.6. Script de cambio de PIN.

Se requieren tres pasos para el cifrado.

- **Cálculo del bloque de PIN.**

F2=0: Bloque de PIN I (sólo Mastercard)

D' = TDES-1(KT)[D1]

D'' = extraer de D' el nuevo valor del PIN de acuerdo a F1

Bloque de PIN I (8 bytes) = 2 || longitud D'' || D'' || relleno a Fs

F2=3: Bloque de PIN IV (sólo Mastercard)

D' = TDES-1(KT)[D1]

D'' = extraer de D' el nuevo valor del PIN de acuerdo a F1

Bloque de PIN IV (8 bytes) = 0 || longitud D'' || D'' || relleno a Fs

F2=1: Bloque de PIN II (sólo Visa)

D' = TDES-1(KT)[D1]

D'' = extraer de D' el nuevo valor del PIN de acuerdo a F1

Bloque de PIN II (8 bytes) = (BloquePIN A) XOR (BloquePIN B) siendo:

BloquePIN A (8 bytes) = 00 00 00 00 || 4 bytes menos significativos de la primera semiclave de MKAC (MKAC-IZQ):

MKAC-IZQ = TDES (K1) [PAN || PSN]

BloquePIN B (8 bytes) = 0 || longitud(D'') || D'' || relleno a Fs

F2=2: Bloque de PIN III (sólo Visa)

D' = TDES-1(KT)[D1]

D'' = extraer de D' el nuevo valor del PIN de acuerdo a F1

Bloque de PIN III (8 bytes) = (BloquePIN A) XOR (BloquePIN B) XOR (BloquePIN C) siendo:

BloquePIN A el calculado en el punto anterior.

BloquePIN B el calculado en el punto anterior.

BloquePIN C (8 bytes) = D0'' || relleno a 0s siendo D0'' el valor del PIN antiguo que se obtiene según:

D0'=TDES-1(KT)[D2]

D0'' = extraer de D0' el antiguo valor del PIN de acuerdo a F1

- **Cifrado del bloque de PIN.**

Cálculo de la clave de cifrado SKSMC

Obtención de la clave de confidencialidad-script por tarjeta:

MKSMC = MKSMC-IZQ || MKSMC-DER

MKSMC-IZQ = TDES (K2) [PAN || PSN]

MKSMC-DER = TDES (K2) [PAN || PSN']

donde:

PAN || PSN' = (PAN || PSN XOR FFFFFFFFFFFFFFFF)

La diversificación de esta clave resultará ser la clave de sesión de confidencialidad-script:

Esquema Mastercard:

SKSMC = SKSMC-IZQ || SKSMC-DER

SKSMC-IZQ = TDES (MKSMC) [D4']

SKSMC-DER = TDES (MKSMC) [D4'']

siendo:

D4' = Bytes 8/7 D4 || F0 || Bytes 5/1 D4

D4'' = Bytes 8/7 D4 || 0F || Bytes 5/1 D4

Esquema Visa:

SKSMC = SKSMC-IZQ || SKSMC-DER

SKSMC-IZQ = (MKSMC -IZQ) XOR (D4)

SKSMC-DER = (MKSMC -DER) XOR (D4')

siendo:

D4' = 000000000000 || (D4 – 2LSB XOR 'FF FF')

- ## Cifrado del Bloque de PIN de Script

Esquema Mastercard y F2 =0:

BloquePIN Script-Cifrado = TDESCBC (SKSMC) [BloquePIN I]

Esquema Mastercard y F2 =3:

BloquePIN Script-Cifrado = TDESCBC (SKSMC) [BloquePIN IV]

Esquema Visa y F2=1:

BloquePIN Script-Cifrado = TDESECB (SKSMC) [08 || BloquePIN II || 80000000000000]

Esquema Visa y F2=2:

BloquePIN Script-Cifrado = TDESECB (SKSMC) [08 || BloquePIN III || 80000000000000]

- ## Firma del script de cambio de PIN.

Cálculo de la clave de firma SK_{SMI}

Obtención de la clave de integridad-script por tarjeta: se obtiene igual que la clave de confidencialidad-script por tarjeta, empleando K3 en lugar de K2.

La diversificación de esta clave resultará ser la clave de sesión de integridad-script:

Esquema Mastercard:

SKSMI = SKSMI-IZQ || SKSMI-DER

SKSMI-IZQ = TDES (MKSMI) [D4']

SKSMI-DER = TDES (MKSMI) [D4'']

siendo:

D4' = Bytes 8/7 D4 || F0 || Bytes 5/1 D4

D4'' = Bytes 8/7 D4 || 0F || Bytes 5/1 D4

Esquema Visa:

SKSMI = SKSMI-IZQ || SKSMI-DER

SKSMI-IZQ = (MKSMI -IZQ) XOR (D4)

SKSMI-DER = (MKSMI -DER) XOR (D4')

siendo:

D4' = 000000000000 || (D4 – 2LSB XOR 'FF FF')

- **Firma del Script de Cambio de PIN**

Firma Script = MAC (SKSMI) [D5 || BloquePIN Script-Cifrado]

El algoritmo MAC a utilizar, será el definido por la norma ANSI X9.19-1. En cuanto al relleno a utilizar, éste se construirá según el método 2 de la norma ISO/IEC 9797: se añade un byte a la derecha con el valor '80' hexadecimal; a continuación, se añade a la derecha el menor número de bytes a '00' hexadecimales que garanticen que la longitud del mensaje final, mensaje original más relleno, sea múltiplo de 8 bytes.

5.7. Cálculo de Códigos de Validación.

5.7.1. Cálculo y validación de CSS/CVV/CVC.

K = KIZQ || KDER

B1 = PAN

B2 = D1 || D2 || 0 00 00 00 00

R1 = DES (KIZQ) [B1]

R2 = R1 (XOR) B2

R3 = DES (KIZQ) [R2]

R4 = DES-1 (KDER) [R3]

R5 = DES (KIZQ) [R4]

R6 = Extraer de R5 los dígitos numéricos (0-9), de izquierda a derecha. Justificar a la izquierda estos dígitos en un campo de 16 posiciones.

R7 = Extraer de R5 los caracteres (A-F), de izquierda a derecha. Convertir a decimal cada uno de estos dígitos restando 10 (decimal).

Concatenar este resultado, al resultado del paso anterior:

R8 = R6 || R7

El CVC/CVV/CSS consistirá en los tres primeros dígitos (de la izquierda) de R8.

5.7.2. Cálculo y validación de CVC3.

KIZQ = TDES (K1) [D1]

KDER = TDES (K1) [$\overline{D1}$]

K = KIZQ || KDER

D = (MAC(K)[D3])2LSB || D2

CVC3 = (TDES(K)[D])2LSB

5.8. Generación de Certificado de Monedero y Transporte TIBC

5.8.1. Certificado de Monedero.

DD= ID || PAN4-15

Ktj = DESECB[Kc](DD)

Ktr = DESECB[Ktj](RN1)

CM=MAC|RN2[Ktr](Bloque0 || Bloque1 || Bloque2)

5.8.2. Certificado de Transporte.

DD= ID || PAN4-15

Ktj = DESECB[Kc](DD)

CT=MAC|RN2[Ktj](Bloque0 || Bloque1 || Bloque2 || Bloque3)

5.9. Generación de Certificado de Monedero y Transporte Advantis

5.9.1. Certificado de Monedero.

DD= ID || PAN4-15

Ktj = DESECB[Kc](DD)

Ktr = DESECB[Ktj](RN1)

CM=MAC|RN2[Ktr](Bloque0 || Bloque1 || Bloque2)

5.9.2. Certificado de Transporte.

DD= ID || PAN4-15

Ktj = DESECB[Kc](DD)

Pasada 1:

CT=MAC|RN2[Ktj](Bloque0 || Bloque1)

Pasada 2:

CT=MAC|RN2[Ktj](Bloque0 || Bloque1 || Bloque2 || Bloque3)

5.10. Carga de Formato de Impresión

Es posible distribuir componentes de claves de forma segura imprimiéndolas, por ejemplo, en sobres ciegos. Con este propósito, puede conectarse una impresora serie al puerto RS-232 de Cryptosec.

Antes de usar los comandos de impresión, es necesario definir el formato del sobre o documento. Un formato se mantiene hasta que se sobrescribe por otro. Se emplean dos comandos para enviar los símbolos de formato al módulo. Los símbolos de formato que definen tanto campos de impresión como cualquier cadena fija se proporcionan en la tabla que se acompaña más adelante.

El comando 1301 debe usarse para crear el formato del documento. La cadena de formato puede contener cualquier texto fijo, si bien se recomienda restringirla a caracteres de formato. Esto es debido a que está limitada a 400 caracteres. El segundo comando de impresión está integrado con la función que hace uso de las capacidades de impresión. En ese punto se pueden incluir hasta 16 cadenas de texto fijas de hasta 252 caracteres cada una, que intervendrán en la impresión de acuerdo a la cadena de formato definida, junto con la información que aporte el propio comando.

Los símbolos de formato de impresión se incluyen en la siguiente tabla:

Símbolo	ASCII	Significado
>L	3E 4C	Salto de línea, retorno de carro
>V	3E 56	Tabulador vertical
>H	3E 48	Tabulador Horizontal
>F	3E 46	Salto de página
>nnn	3E 3n 3n 3n	Salta a la columna nnn desde le margen izquierdo, donde nnn es un número decimal de tres dígitos
^M	5E 49	Imprime la tercera componente de una clave en claro.
^P	5E 50	Imprime el PIN en claro para el sobre 1, o bien imprime una componente de clave en claro.
^Q	5E 51	Imprime el PIN en claro para el sobre 2, o bien imprime una componente de clave en claro.
^R	5E 52	Imprime la referencia del sobre 1
^S	5E 53	Imprime la referencia del sobre 2
^T	5E 54	Imprime los últimos 6 dígitos del número de cuenta en el sobre 1, o bien imprime el KCV de una componente de clave
^U	5E 55	Imprime los últimos 6 dígitos del número de cuenta en el sobre 2
|<L><hh hh hh ..>	7C<L><hh hh hh ..>	Manda datos binarios a la impresora, por ejemplo secuencias de control. L contiene el número de bytes a enviar, hasta 255 bytes. Siguen los bytes a enviar.
^0	5E 30	Inserta el campo de impresión 0 definido en la función que hace uso de la impresión
^1	5E 31	Inserta el campo de impresión 1 definido en la función que hace uso de la

Símbolo	ASCII	Significado
		impresión
…	…	…
^F	5E 46	Inserta el campo de impresión 15 definido en la función que hace uso de la impresión

Notar que aunque todos los símbolos están soportados, en algunos casos el firmware no soporta su uso. En cada comando se detalla de qué símbolos se vale para devolver su información.

5.11. Diversificación de Clave

Identificador de algoritmo 12:

$Kr = Kr_{IZQ} || Kr_{DER}$

$Kd = Kd_{IZQ} || Kd_{DER}$

$Kr_{IZQ} = TDES_{ECB} (Kd) [D_1]$

$Kr_{DER} = TDES_{ECB} (Kd) [D_2]$

Identificador de algoritmo 22:

$Kr = Kr_{IZQ} || Kr_{DER}$

$Kd = Kd_{IZQ} || Kd_{DER}$

$ER = TDES_{CBC(0000000000000000)}(Kd) [ICCID_{1-8} || ICCID_{9-10} || ICCID_{1-6} || ICCID_{7-10} || FFFFFFFF]$

$Kr_{IZQ} = ER_{9-16}$

$Kr_{DER} = ER_{17-24}$

5.12. Método de cálculo ALT_1 PIN Offset

En lugar de calcular el PIN Offset como diferencia entre el PIN elegido por el cliente (C-PIN) y el PIN calculado a partir de los datos de validación (A-PIN), se calcula como la suma de dichos valores, esto es:

O-PIN = **C-PIN** + **A-PIN**

Tanto la suma como la resta se entienden dígito a dígito, modulo 10 y sin acarreo.

En cuanto al cálculo del A-PIN, este se describe a continuación:

Cifrar el dato de validación (16 dígitos hexadecimales) con la clave de generación de PIN.

Decimalizar el resultado del paso anterior, recorriendo los 16 dígitos hexadecimales de izquierda a derecha, obviando todo dígito mayor que 0x9, hasta que se encuentren cuatro dígitos decimales (dígitos que tienen valores desde 0x0 hasta 0x9).

Si se han recorrido todos los dígitos pero no se han encontrado cuatro dígitos decimales, repetir el proceso, obviando todos los dígitos desde 0x0 hasta 0x9. Restar (módulo 10 y sin acarreo) 0xA a cada dígito seleccionado en este recorrido.

El PIN se forma concatenando el resultado de ambas pasadas y reteniendo los cuatro primeros dígitos.

II. English

1. LMK KEYS

1.1. Description

From a Master Key, loaded by custodians through the HSM interface, it is possible to diversify other keys that can be stored inside the module, called LMK Keys.

The LMK Keys are used in external storage, to encrypt operational keys to be stored in a database. Thus, the keys are grouped by area of use. Grouping the keys ensures that a Key is not used for anything else than itsparticular function.

All diversified keys are triple length. Each Key stored in databases will be associated with one LMK Key.

The following table shows the types of LMKs:

Type	Description	Operation	Only in Authorization Mode
LMK 0	Master Key	LMK 0 KCV Calculation (6 characters)	NO
LMK 1	Custodian Keys	Generation/Deletion	YES
		Import/Export (RSA)	YES
		Import/Export by components	YES
		KCV Calculation (6 characters)	NO
LMK 2	Key Transport Keys	Generation/Deletion	NO
		Import/Export by components	YES
		Import/Export encrypted with custodian key	NO
		Import/Export (RSA)	NO
		KCV Calculation (6 characters)	NO
LMK 3 *	PIN Block Transport Keys	Generation/Deletion	NO
		Import/Export encrypted with transport Key	NO
		Import/Export (RSA)	NO
		Diversification	NO
		KCV Calculation (6 characters)	NO
		PIN Block Conversions	NO
LMK 4*	PIN Keys	Generation/Deletion	NO
		Import/Export by components	YES
		Import/Export encrypted with transport Key	NO
		Import/Export (RSA)	NO
		KCV Calculation (6 characters)	NO
		PIN Block Construction	NO
		PIN Block Verification	NO
LMK 5*	CVV Keys	Generation/Deletion	NO

Type	Description	Operation	Only in Authorization Mode
		Import/Export by components	YES
		Import/Export encrypted with transport Key	NO
		Import/Export (RSA)	NO
		KCV Calculation (6 characters)	NO
		CSS/CVC/CVV Calculation	NO
		CSS/CVC/CVV Verification	NO
LMK 6*	EMV Authorization Keys	Generation/Deletion	NO
		Import/Export by components	YES
		Import/Export encrypted with transport Key	YES
		Import/Export (RSA)	NO
		Diversification	NO
		KCV Calculation (6 characters)	NO
		EMV Authorization Transaction Commands	NO
		Script Security Treatment Commands	NO
LMK 7	Data Encryption Keys	Generation/Deletion	NO
		Import/Export by components	YES
		Import/Export encrypted with transport Key	NO
		Import/Export (RSA)	NO
		KCV Calculation (6 characters)	NO
		Diversification	NO
		Encryption/Decryption CBC Commands	NO
LMK 8	MAC Keys	Generation/Deletion	NO
		Import/Export encrypted with transport Key	NO
		Import/Export (RSA)	NO
		Diversification	NO
		KCV Calculation (6 characters)	NO
		MAC Generation	NO
		MAC Verification	NO
LMK 9*	Cash Keys	Generation/Deletion	NO
		Import/Export by components	YES
		Import/Export encrypted with transport Key	NO
		Import/Export (RSA)	NO
		KCV Calculation (6 characters)	NO
		Diversification	NO
LMK 10*	EMV Personalization Keys	Generation/Deletion	NO
		Import/Export by components	YES
		Import/Export encrypted with transport Key	YES
		Import/Export (RSA)	NO
		KCV Calculation (6 characters)	NO
		Diversification	NO
LMK 11	TAF Keys	Generation/Deletion	NO
		Import /Export (RSA)	NO
		KCV Calculation (6 characters)	NO
LMK 12	VIAT Keys	Import/Export encrypted with transport key	YES
		Import/Export (RSA)	NO
		KCV Calculation (6 characters)	NO
LMK 13	Irreversible PIN Keys	Generation/Deletion	NO
		Import/Export by components	NO
		Import/Export encrypted with transport key	YES
		Import/Export (RSA) key	NO

Type	Description	Operation	Only in Authorization Mode
		KCV Calculation (6 characters)	NO
LMK 14*	Terminal Authentication Keys	Generation/Deletion	NO
		Import/Export by components	NO
		Import/Export encrypted with transport Key	YES
		Import/Export (RSA)	YES
		KCV Calculation (6 characters)	NO
		Diversification	NO
LMK 15	Decimalization Table Keys	Generation/Deletion	NO
		Import/Export by components	NO
		Import/Export encrypted with transport key	NO
		Import/Export (RSA)	NO
		KCV Calculation (6 characters)	NO
		PN block commands	NO
LMK 16	Log Signature Keys	Generation/Deletion	YES
		Import/Export by components	YES
		Import/Export encrypted with transport key	YES
		Import/Export with (RSA)	YES
		KCV Calculation (6 characters)	NO
LMK 17	Password Encryption Keys	Generation/Deletion	NO
		Import/Export by components	YES
		Import/Export encrypted with transport Key	NO
		Import/Export (RSA)	YES
		KCV Calculation (6 characters)	NO
		Encryption Commands	NO
LMK 60	RSA Keys	Generation	NO
		Import/Export encrypted with transport key	NO
		KCV Calculation (6 characters)	NO
		Transfer private key from a LMK to other LMK	YES
		Encryption/Decryption RSA	NO
		Signature Calculation/Verification	NO
LMK 61	RSA Keys Sign/Verify	Generation	NO
		Import/Export encrypted with transport key	NO
		KCV Calculation (6 characters)	NO
		Transfer private key from a LMK to other LMK	YES
		Encryption/Decryption RSA	NO
		Signature Calculation/Verification	NO
LMK 62	RSA Keys Sign/Verify Encrypt/Decrypt	Generation	NO
		Import/Export encrypted with transport key	NO
		KCV Calculation (6 characters)	NO
		Transfer private key from a LMK to other LMK	YES
		Encryption/Decryption RSA	NO
		Signature Calculation/Verification	NO
LMK 62-98	Reserved for future use	----------	---
LMK 99	Free use Keys (Test)	Generation/Deletion	YES
		Import/Export by components	NO
		Import/Export encrypted with transport Key	NO
		Import/Export (RSA)	NO
		KCV Calculation (custom value)	NO
		Diversification	NO
		Free use commands (Every commands defined)	NO

* CryptosecBANKING ONLY

2. Comunication Protocol with the HSM

2.1. Structure of Messages

The tables acronyms used for the operations are structured and named according to the following tables.

Data types:

These are the abbreviations for the letters that are used in the Type column.

Type	Comment
A	Alfanumeric Data
H	Hexa Data
N	Numeric Data
B	Binary Data
K	Key framework defined below

Command Message structure:

This table shows how the request messages are structured.

Type	Length	Comment
D	6	Message length
A	var	Header (*)
H	4	Command Identifier
--	var	Command Data

Response Message structure:

This table shows how the response messages are structured.

Type	Length	Comment
D	6	Message length
A	var	Header
H	4	Command Identifier
H	8	Command State
--	var	Response Data(2*)

(*)The messages format allows a header length parameterizedby the Entity. In HSM response arrive this header unchanged, so that the application that made the request verify that the message actually returned is that one which expects.

(2*)This field will be present only if the command has been successfully resolved, that is, if the state of the command is '00000000'.

2.2. Key Structure

The presentation of a key in data form from and to the HSM is structured in the following manner:

Type	Length	Comment
A	1	Type of Key (*)
N	4	Key Length (length of next field) (2*)
H	var	Key Value (3*)
A	3	Key Encryption (4*)
N	2	KCV Length (length of next field)
H	var	KCV Value (5*)

(*) Type of Key can take the following values:

- D – DES keys stored externally.
- S – DES keys stored internally..
- R – RSA private keys.
- P – RSA public keys.
- I – IBM format RSA private keys structured in CRT.
- V –DB stored keys.

(2*)When key type is I (IBM RSA format) then key length must be 5000 bytes.

(3*)Dependent on the key type, it will appear in the following format:

- DES Keys stored externally or to be stored internally, in hexadecimal format.
- DES Key stored internally, five digits corresponding to the key identifier returned by the HSM.
- RSA Public Keys, the value of which is the codification DER in the structure ASN.1 defined in the standard PKCS#1.
- RSA Private Keys (not in IBM format structured in CRT), whose value is the DER codification in the structure ASN.1 PrivateKeyInfo defined in the standard PKCS#8.
- RSA Private Keys in IBM format structured in CRT: IBM proprietary structure.

(4*)In this field will be coded by a type encryption key:

- L00, L01 …… L99 indicates that Key is encrypted with a LMK stored in HSM.
- D00 indicates that Key is encrypted with DES algorithm, with an external Key to HSM.
- N00 indicates DES Keys internally stored.
- N00 also indicates that Key is not encrypted.
- R00 indicates that Key is encrypted with RSA algorithm, with an external Key to HSM.

(5*)KCV of the Key will lie provided whenever the Key can support this type of value and wish to specify a key verification system.To not specify KCV, this field is left empty, and then its length 00. In case that the key does not admit the KCV then the value is not taken into account.(For RSA public keys, this field will contain the MAC.)

Commands Format

3.1. Key Management

3.1.1. Key Generation - 0101 -

Allows the generationof all types of Keys that the system is capable of managing:

- DES keys of several lengths both internal and external, and
- RSA keys using different types of test.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0101.
LMK	N	2	LMK label associated with the key to generate. The value should be different from 00.
T	A	1	Type of key to generate. - D to generate DES keys (external storage). - S to generate DES keys (internal storage). - R to generate RSA keys.
L	N	4	Length in bits of the Key to generate: - 0064 for simple DES Keys. - 0128 for double DES Keys. - 0192 for triple DES Keys. - From 0512 to 2048 in multiples of 32-bits for RSA Keys.
E	N	1	(*)Exponent of the RSA Key to generate: - 0 if the exponent is 3. - 1 if Fermat4.
M	N	2	(2*) RSA Key generation mode: - 00 Fermat Mode. - 01 Automatic Mode. - From 02 to 50, number of cycles Miller-Rabintest.

Response message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return value
K1	K	var	Generated Key
K2	K	var	Public Key (*).

(*)Only if T = R.

(2*) Only if T = R. Generation mode consist of the primality test that are made turn to the components of a RSA Key. The easiest mode is Fermat test; the following is automatic mode in which the module calculates the number of passes of Miller-Rabin test so that the probability of false positives is less than 2^{-100}. In all modes carried out a test in front of small prime factors.

3.1.2. Delete Key - 0102 -

Allows the deletion of a DES Key stored internally.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0102.
K1	K	var	Internal DES Key to erase.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier
RV	H	8	Return Value.

3.1.3. List of Keys - 0103 -

Retrieve a list of the DES keys stored internally.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0103.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier
RV	H	8	Return Value.
NK	N	5	Number of internal DES Keys.
HK1	N	5	(*) Identifier of internal DES Key.
LMKK1	N	2	(*) LMK of that Key.
…	…	…	…
HKNK	N	5	(*) Identifier of internal DES Key.
LMKKNK	N	2	(*) LMK of that Key.

(*) Needs equal amount of fields as number of internal DES keys specified in the field NK .

3.1.4. Delete DB Key- 0104 -

Allows the deletion of an internal RSA key stored in the DB.

Submission message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0104.
LEN	N	2	Label length
L	A	var	Label

Response message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return value.

3.1.5. List of DB Keys - 0105 -

Allows the retrieval of an internally stored RSA keys list.

Submission message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0105.

Response message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return value.
NK	N	5	Number of internal keys.
LEN	N	2	(*) Next field length
L	A	var	(*) Label
…	…	…	…
LEN	N	2	(*) Next field length
L	A	var	(*) Label

(*) As pairs of fields as number of keys indicated in field NK.

3.1.6. Obtain a DB Key - 0106 -

Allows the retrieval ofan internally stored RSA key.

Submission message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0106.
LEN	N	2	Next field length
L	A	var	Label

Response message:

Data	Type	Length	Comment
IC	H	4	Command identfier
RV	H	8	Return value.
K1	K	var	Key

3.1.7. Store DB Key - 0107 -

Allows the storageof a DB key.

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0107.
K1	K	var	Key to store.
LEN	N	2	Next field lengh
L	A	var	Label

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.

3.1.8. Import Key - 0201 -

Provides the ability to incorporate external keys in the system. The Key to be imported is encrypted with a transport, custodian or RSA key.

For new implementations we suggest the use of the command 0202.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0201.
LMK	N	2	LMK label for the imported Key.Value different to 00.
K1	K	var	Key to import.
K2	K	var	Decryption Key, under LMK 01, LMK 02, LMK 60 or LMK 62.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identfier
RV	H	8	Return Value
K	K	var	Imported Key

3.1.9. Import Key v2 - 0202 -

Provides the ability to incorporate external keys in the system. Key to import is encrypted with a transport, custodian or RSA key.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0202.
LMK	N	2	LMK label for the imported key. Value different to 00.
ALG	H	2	Importation Algorithm: - 00 EBC. - 01 RSA-PKCS#1v1.5 - 02 RSA-OAEP.
K1	K	var	Key to import.
K2	K	var	Key of decryption, under LMK 01, LMK 02, LMK 60 or LMK 62.
F_1	A	1	(2*)Indicator of internal storageof imported DES key: - S.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier
RV	H	8	Return Value
K1	K	var	Imported Key
K2	K	var	Public Key (*).

(*) Only if K1 is of type RSA.

(2*) Only if imported DES Key must be stored internally.

3.1.10. Import Signed RSA Public Key - 0203 -

Allows the importation of external signed RSA public keysto the system.

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0203.
LMK	N	2	Key to impport LMK label. Value must be 60, 61 or 62.
T1	N	1	Data type which contains public key: - 0 Public key structure - 1 Certificate
L1	N	4	(*) Next field length
K1	H	var	Public key container
L2	N	4	Signature Length (next field)
F1	H	var	The signature of the Key to import
K2	K	var	Verification key, under LMK 61 or 62.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier

Data	Type	Length	Comment
RV	H	8	Return value.
K1	K	var	Imported key.

(*) Only if T1 = 1.

3.1.11. Import Unsigned RSA Public Key - 0204 -

Allows the importation of external RSA public keys without the signature of another RSA key to the system.

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0204.
LMK	N	2	Key to import LMK label. Value must be 60, 61 or 62.
K1	K	var	Key to import.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier
RV	H	8	Return value
K1	K	var	Imported key.

3.1.12. Export in Components - 0301 -

Exports a DES key in components through the Terminal. Moreover, it can generate the key to export.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0301.
M	N	1	Action to make: - 0 Generates key and exports it in components. - 1 Exports key included in the message.
LMK	N	2	(*)LMK label for the generated key. Value must be different to 00.
L1	N	2	(*)Length of the key to generate in bytes.
C	N	1	Number of components to return, between 2 and 9.
K	K	var	(2*)Key to export.
L3	N	4	Length of the following field in bytes.
T	A	var	Table of variable user data (name, custodian, ATM machine, dates, finality, Key…). Format depends on definition of printing format.
F_1	A	1	(3*)Indicator of internal storage of generated key:

Data	Type	Length	Comment
			- S.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier
RV	H	8	Return Value
K	K	var	(*)Generated Key.

(*) Only if M = 0.

(2*) Only if M = 1.

(3*) Only if M = 0 and generated key have to be stored internally.

Notes about Printing

It is necessary to define the template for printing by running the command 1301 before performing printing operation.

In the Export in Components command, the argument T contains print strings 0 to 15 that are referenced in the template.There is no need to define all of the 16 strings.However, one must take into account that they are numbered in a sequential manner, starting from 0.Sequence "\0" is used as a string delimiter. The symbol ^P induces the printing of the key component.Symbol ^T involves printing the KCV Key component.

To clarify the operation of the Export in Components command there is an example below.Suppose you want to generate and print two components of a key, accompanied by the KCV for control purposes, it would be thefollowing:

COMPONENT: XXXXXXXXXXXXXXXX KCV: YYYYYY

Where X...X is the component and Y...Y is the KCV. In this case we keep the template to a minimum:

>L^0^P^1^T>F

Where ^0 denotes the inclusion of the first string and the data table ^1 denotes the point of includingthe second channel of the data table. ^P denotes printing the key component and ^T denotes printing the component KCV.

With this template, to obtain the wanted printing, the data table to include in the command is:

COMPONENT: \0 KCV:

The \0 is used as separator, and it is not necessary to include following the last string.This string has a length of 00000020.

The alternative to simplify(in this case, eliminate) the data table would be including data strings into the templates:

>L COMPONENT: ^P KCV: ^T>F

Although, this approach is less proper froma formal perspective.

3.1.13. Export Key - 0302 -

Exports a DES key encrypted witha DES or RSA key.Can also be used to export a RSA key encrypted with a DES key. The exported key will be encrypted under an external transport, custodian or RSA key.

For new implementations we suggest using command 0303.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0302.
K1	K	var	Key to export.
K2	K	var	Key used to export (encrypted under)the key above. Encrypted under LMK 01, LMK 02, LMK 60 o LMK 62

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier
RV	H	8	Return Value
K	K	var	Exported Key

3.1.14. Export Key v2 - 0303 -

Export a DES key encryptedwitha DES or RSA key.Can also be used to export a RSA key encrypted with a DES key. The exported key is encrypted under an external transport, custodian or RSA key.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0303.
ALG	H	2	Export Algorithm: - 00 EBC. - 01 RSA-PKCS#1v1.5 - 02 RSA-OAEP.
K1	K	var	Key to export
L1	N	4	Length of the following field.

Data	Type	Length	Comment
D1	H	var	Data to concatenate
K2	K	var	Key under which is exported the key above. Encrypted under LMK 01, LMK 02, LMK 60 o LMK 62

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value
K	K	var	Exported Key

3.1.15. Key KCV Calculation - 0401 -

Returns the KCV of a given key.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0401.
K	K	var	The Key of which you want to get the KCV
L	N	2	Returned KCV length.This field will only be taken into account for key free use, LMK 99. For other keys, is not taken into account and used as the value 6.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier
RV	H	8	Return Value
L	N	2	Length of the following field in bytes
KCV	H	var	Calculated KCV.

3.1.16. KCV of LMK Calculation - 0402 -

Returns the KCV of a LMK.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0402.
IMM	N	2	Index of the LMK of which you want to obtain the KCV.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier
RV	H	8	Return Value
KCV	H	6	KCV returned.

3.1.17. Update Key Storage - 1801 -

Allows the re-encryption of a key, that was encrypted witha LMK of an old set, withthe same LMK of the current set. This makes it possible to update all the keys stored externally every time you change the master key of the system.

For this command to work properly, the followingmust be taken into account:

- Have uploaded a new master key of the system.
- In the process, have kept the derived LMKs from the earlier Master Key.
- The key to update should have stored under a LMK associated with the Master key cited above: the update process is continuous because it is stored only for this purpose the most recently replaced set of LMKs.
- Not having eliminated these LMKs prior to the execution of this command. Once removed, it will only be possible to recover the keys encrypted under this set of keys loading again the associated Master Key into the system.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 1801.
K	K	var	Key you want to update.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
K	K	var	Updated Key.

3.2. Authorization of EMV Transactions(*CryptosecBANKING)

3.2.1. DAC Calculation and Validation - 0501 – (*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0501
K	K	var	DAC Key (under LMK6).
M	N	1	Execution Mode: - 0 DAC Calculation. - 1 DAC Validation.
D1	H	16	PAN
DAC	H	4	(*)DAC to validate.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier
RV	H	8	Return Value.
DAC	H	4	(2*)Calculated DAC value.
VER	N	1	(*)Validation result: - 0 correct. - 1 incorrect.

(*) Only if M = 1.

(2*) Only if M = 0.

3.2.2. IDN Calculation and Validation - 0502 - (*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier 0502.
K	K	var	IDN key (under LMK6).
M	N	1	Execution mode: - 0 IDN calculation. - 1 IDN validation.
D1	H	16	PAN
D2	H	16	ATC || 00 || 00 || UN. Usually UN=00 00 00 00
IDN	H	4	(*)IDN to validate.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
IDN	H	16	(2*) Calculated IDN Value.
VER	N	1	(*)Validation Result: - 0 correct. - 1 incorrect.

(*) Only if M = 1.

(2*) Only if M = 0.

3.2.3. ARQC Verification and ARPC Generation - 0503 - (*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0503.
M	N	1	Execution Mode: - 0 Verificate ARQC. - 1 Verification ARQC and generate ARPC. - 2 Generate ARPC. - 3 Calculate ARQC.
E	N	1	Scheme: - 0 Visa VSDC. - 1 Mastercard.
K	K	var	Operation Key (under LMK6).
D1	H	16	Pre-formatted PAN / PANseq No
D2	H	4	(*)Transactions Counter (ATC)
D3	H	8	(*)Unpredictable Number (UN).
L4	N	3	(2*)Next field Length.
D4	H	var	(2*) Transaction Data (DO NO PAD).
D5	H	16	(3*)ARQC/ TC/ AAC to validate and/ or use to generate ARPC.
D6	H	16	(4*)Response code used for ARPC (ARC || 00 00 00 00 00 00).

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
ARQC	H	16	(5*) Calculated ARQC.
VER	N	1	(6*) Validation Result: - 0 correct. - 1 incorrect.
ARPC	H	16	(4*)Generated ARPC.

(*) Only if E = 1 and M ≠ 2.

(2*) Only if M = 0, M = 1 or M = 3.

(3*) Only if M = 0, M = 1 or M = 2.

(4*) Only if M = 1 or M = 2.

(5*) Only if M = 3.

(6*) Only if M = 0 or M = 1.

3.3. Scripts Security(*CryptosecBANKING)

3.3.1. Script Signature - 0504 -(*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0504.
E	N	1	Scheme: - 0 Visa VSDC. - 1 MasterCard.
K	K	var	Operation Key (under LMK6).
D1	H	16	Diversification Data 1.
D2	H	16	Diversification Data to script signature.
L3	N	6	Next field Length.
D3	H	var	Data to sign.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
F	H	16	Signature.

3.3.2. Script Encryption - 0505 -(*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0505.
E	N	1	Scheme: - 0 Visa VSDC. - 1 MasterCard.
K	K	var	Operation Key (under LMK6).
D1	H	16	Diversification Data 1.
D2	H	16	Diversification Data to script encryption.
L3	N	6	Next field Length.
D3	H	var	Data to encrypt.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
L	N	6	Next field Length in bytes..

Data	Type	Length	Comment
F	H	var	Encrypted Data.

3.3.3. PIN Exchange Script - 0506 - (*CryptosecBANKING)

For new implementations we suggest the use of the command 0507.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0506.
E	N	1	Scheme: - 0 Visa VSDC. - 1 MasterCard.
KT	K	var	ATM Transport Key of PIN (under LMK3).
K1	K	var	(*)Operation Key 1 (under LMK6).
K2	K	var	Operation Key 2 (under LMK6).
K3	K	var	Operation Key 3 (under LMK6).
F1	N	1	Input PIN Block Format: - 1 ISO1. - 2 ISO2. - 4 IBM 3624 - 5 Diebold
F2	N	1	Output PIN Block Format: - 0 PIN block I: Standard EMV PIN Block. - 1 PIN block II: Visa Format Without Using Current PIN. - 2 PIN block III: Visa Format using Current PIN.
D1	H	16	PIN block received from the ATM machine (built with the new PIN).
D2	H	16	(2*) PIN block received from the ATM machine (built with the old PIN).
D3	H	16	Diversification Data 3: the 14 most significant PAN digits on the right. PAN || PSN.
D4	H	16	Diversification Data 4: ATC if VISA, ARQC if Mastercard.
L5	N	6	Next field Length.
D5	H	var	Data to sign.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
PIN	H	16	Encrypted PIN block.
FS	H	16	Script command Signature.

(*) Only if E = 0.

(2*) Only if F2 = 2.

3.3.4. PIN Exchange Script v2 - 0507 - (*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0507.
E	N	1	Scheme: - 0 Visa VSDC. - 1 MasterCard.
KT	K	var	ATM Transport Key of PIN (under LMK3).
K1	K	var	(*)Operation Key 1 (under LMK6).
K2	K	var	Operation Key 2 (under LMK6).
K3	K	var	Operation Key 3 (under LMK6).
F1	N	1	Input PIN Block Format: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM 3624 - 5 Diebold
F2	N	1	Output PIN Block Format: - 0 PIN block I: Standard EMV PIN Block. - 1 PIN block II: Visa Format Without Using Current PIN. - 2 PIN block III: Visa Format using Current PIN. - 3 PIN block IV: Mastercard Pay Now & Pay Later.
D1	H	16	PIN block received from ATM (built with new PIN).
D2	H	16	(2*) PIN block received from ATM (built with old PIN).
D3	H	16	Diversification Data 3: PAN.
D3b	H	2	Diversification Data 3b: PSN.
D4	H	16	Diversification Data 4: ATC if VISA, ARQC if Mastercard.
L5	N	6	Next field Length.
D5	H	var	Data to sign.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
L1	N	6	Next field length.
PIN	H	var	Encrypted PIN block.
FS	H	16	Script command signature.

(*) Only if E = 0.

(2*) Only if F2 = 2.

3.3.5. ARQC Verification and ARPC Generation (EMV 4.1) - 0508 - (*CryptosecBANKING)

Request message::

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0508.
M	N	1	Mode: 0 – Verify ARQC. 1 – Verify ARQC & perform ARPC generation method 1 EMV 4.1. 2 – Perform ARPC generation method 1 EMV 4.1. 3 – Verify ARQC & Perform ARPC generation method 2 EMV 4.1. 4 – Perform ARPC generation method 2 EMV 4.1.
E	N	1	Esquema: 2 – VIS1.4.0 and M/CHIP4 using Card Key Derivation Method A and EMV Common Session Key Derivation Method. 3 – VIS1.4.0 and M/CHIP4 using Card Key Derivation Method B and EMV Common Session Key Derivation Method.
K	K	var	Key for Application Cryptograms (under LMK6).
L1	N	2	(*)PAN / PANseqNo length – already formated (08 a 20)
D1	N	var	PAN / PANseq No – already formatted
D2	H	4	Application Transaction Counter (ATC).
L3	N	3	(2*)Transaction Data Length.
D3	H	var	(2*) Transaction Data (MUST BE PROPERLY PADDED)
D4	H	16	ARQC/ TC/ AAC to be validated and/or used for ARPC generation.
D5	H	4	(3*)ARC. Used for ARPC generation.
D6	H	8	(4*)CSU. Used for ARPC generation.
L7	N	2	(4*)Propietary Authentication Data Length (0…16)
D7	H	var	(4*) Propietary Authentication Data

Mensaje de respuesta:

Data	Type	Lengh	Comment
IC	H	4	Command Identifier.
RV	H	8	Response Code.
VER	N	1	(5*)Validation result: - 0 No error. - 1 ARQC validation failure.
ARPC	H	16	(6*)ARPC generated.
ARQC	H	16	(7*)ARQC calculated.

(*) Only ifl E=3

(2*) Only if M = 0, M = 1 ó M = 3.

(3*) Only if M = 1 ó M = 2.

(4*) Only if M = 3 ó M = 4.

(5*) Only if M = 0, M=1 ó M = 3.

(6*) Only if M = 1, M=2 ó M = 3.

(7*) Only if ARQC verification failure and the HSM is in Authorised state.

PIN Functions(*CryptosecBANKING)

3.3.6. PIN Calculation - 0601 -(*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0601.
K1	K	var	PIN Calculation Key (under LMK4).
K2	K	var	PIN Encryption Key (under LMK3).
TD	N	16	Decimation Table.
F	N	1	PIN Block Format: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold
ALG	N	1	PIN Generation Algorithm: - 0 IBM3624. - 1 IBM3624 PIN Offset. - 4 IBM German Bank Pool Institution. - 5 Interbank.
LPAN	N	2	Next field Length.
PAN	N	var	PAN.
D1	H	1	(*)Padding Value.
LPIN	N	1	(2*)PIN Length.
D2	H	1	(3*)Key Indicator.
D3	H	3	(3*)Validation field.
L5	N	1	(4*)PIN Offset Length.
D5	H	var	(4*)PIN Offset.

Response message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
PIN	H	16	Encrypted PIN block.

(*) Only if F = 4.

(2*) Only if ALG = 0.

(3*) Only if ALG = 5.

(4*) Only if ALG = 1.

3.3.7. PIN Calculation given a decrypted or random PIN - 0611 -(*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0611.
K1	K	var	PIN encryption key (under LMK3).
F	N	1	PIN block format: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.
LPAN	N	2	PAN length.
PAN	N	var	PAN.
D1	H	1	(*)Padding value.
EPIN	N	1	PIN presentation: - 0 PIN en claro. - 1 random.
LPIN	N	1	PIN length.
PIN	H	Var	(2*)PIN without encryption

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
PIN	H	16	Encrypted PIN block.

(*) Only if F = 4.

(2*) Only if EPIN = 0.

3.3.8. PIN Verification - 0602 -(*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0602.
K1	K	var	PIN Calculation Key(under LMK4).
K2	K	var	PIN Encryption Key (under LMK3).
TD	N	16	Decimation Table.
F	N	1	PIN block format: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.

Data	Type	Length	Comment
ALG	N	1	PIN Generation Algorithm: - 0 IBM3624. - 1 IBM3624 PIN Offset. - 4 IBM German Bank Pool Institution. - 5 Interbank. - 6 ALT_1 PIN Offset.
UKP	N	1	UKPT Indicator: - 0 do not use. - 1 use.
PIN	H	16	PIN Block.
D1	H	20	(*)Current Key Sequence Number.
PEM	N	2	(2*)PIN Extraction Method: - 00 does not use padding value. Equivalent to 03. - 01 uses padding value. - 02 until first hex digit found. - 03 last character of decrypted block is the padding one. - 04 – 16 pin length indicated by its value.
LPN	N	2	Next field Length.
PAN	N	var	PAN.
PAD	H	1	(3*)Padding Value.
LPIN	N	1	(4*)PIN Length.
LPIF	N	1	(5*)Next field Length.
PIF	H	var	(5*)PIN Offset.
D2	H	1	(6*)Key Indicador.
D3	H	3	(6*)Validation field.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
VPIN	N	1	PIN Verification result: - 0 correct. - 1 incorrect.

(*) Only if UKP = 1.

(2*) Only if F= 4.

(3*) Only if F = 4 and PEM = 00 or PEM = 01.

(4*) Only if ALG = 0.

(5*) Only if ALG = 1or ALG = 6.

(6*) Only if ALG =5.

3.3.9. PIN Verification from Point of Sale Terminal - 0606 -(*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0606.
K1	K	var	PIN calculation key (under LMK4).
K2	K	var	PIN encryption key (under LMK3).
ND	N	1	(*) Number of derivation data of K2 (0..9).
DD1	H	16	(*) Derivation data 1.
…	…	…	…
DDN	H	16	(*)Derivation data nd.
TD	N	16	Decimalization table.
F	N	1	PIN block format: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.
ALG	N	1	PIN generation algorithm: - 0 IBM3624. - 1 IBM3624 PIN Offset. - 3 Visa PVV. - 4 IBM German Bank Pool Institution. - 5 Interbank. - 6 ALT_1 PIN Offset.
UKP	N	1	UKPT indicator: - 0 not use. - 1 use.
PIN	H	16	PIN Block.
DV	H	16	Validation Data.
D1	H	20	(2*)Current Key Sequence Number.
PEM	N	2	(3*)PIN Extraction Method: - 00 uses padding value. - 01 uses padding value. - 02 until first hex digit found. - 03 last character of decrypted block is the padding one. - 04 – 16 padding length indicated by its value.
LPN	N	2	(4*) Next field length.
PAN	N	var	(4*) PAN.
PAD	H	1	(5*)Padding value.
LPIN	N	1	(6*) PIN length.
LPIF	N	1	(7*) Next field length.
PIF	H	var	(7*)PIN Offset.
D2	H	1	(8*)Key indicator.
D3	H	3	(8*)Validation field.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
VPIN	N	1	PIN Verification result: - 0 correct. - 1 incorrect.

(*) As many derivation data fields as number of data indicated in field ND.

(2*) Only if UKP = 1.

(3*) Only if F= 4.

(4*) Only if F= 0 ó F=3.

(5*) Only if F = 4 and PEM = 00 or PEM = 01.

(6*) Only if ALG = 0.

(7*) Only if ALG = 1, ALG = 3 or ALG=6.

(8*) Only if ALG =5.

3.3.10. PVV Verification - 0607 -(*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0607.
K1	K	var	PIN Calculation Key(under LMK4).
K2	K	var	PIN Encryption Key (under LMK3).
F	N	1	PIN block format: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.
PIN	H	16	PIN Block.
PEM	N	2	(*)PIN Extraction Method: - 00 uses padding value. - 01 uses padding value. - 02 until first hex digit found. - 03 last character of decrypted block is the padding one. - 04 – 16 padding length indicated by its value.
LPN	N	2	Next field Length.
PAN	N	var	PAN.
PAD	H	1	(2*) Padding value.
PVK	N	1	Used Key index digit.
PVV	H	4	PVV

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier
RV	H	8	Return Value.
VPIN	N	1	PIN verification result: - 0 correct. - 1 incorrect.

(*) Only if F= 4.

(2*) Only if F = 4 and PEM = 00 or PEM = 01..

3.3.11. Load weak PINs List - 0612 -(*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0607.
LPIN	N	2	PIN Length
NPIN	N	4	Number of PINs to be included next.
TPIN	N	var	Weak PINs list to load into the HSM.

Response message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return value.

3.4. PIN Protection in Exchange: TDES(*CryptosecBANKING)

3.4.1. PIN Management - 0603 -(*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0603.
K1	K	var	ATM PIN transport key (under LMK3).
K2	K	var	PIN block encryption key (under LMK3).
F1	N	1	Input PIN block format: - 0. ISO0. - 1. ISO1. - 2. ISO2. - 3. ISO3. - 4. IBM3624. - 5. Diebold
F2	N	1	Output PIN block format: - 0. ISO0. - 1. ISO1. - 2. ISO2. - 3. ISO3. - 4. IBM3624. - 5. Diebold
PIN	H	16	Encrypted PIN block, received from ATM.
L1	N	2	Next field length.
D1	N	var	PAN.
L2	N	2	(*)Next field length.
D2	H	var	(*) Random number used to form outgoing PIN block.
D3	H	1	(2*) Padding value.

Response message:

Data	Typo	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Returned value.
PIN	H	16	Encrypted PIN block.

(*) Only if F2=1 or F2 = 3. D2 maximum length is 16.

(2*) Only if F2 = 4.

3.4.2. PIN from Point of Sale Terminal management - 0610 -(*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0610.
K1	K	var	Transport PIN master key (under LMK3).
K2	K	var	PIN block encryption key (under LMK3).
F1	N	1	Input PIN block format: - 0. ISO0. - 1. ISO1. - 2. ISO2. - 3. ISO3. - 4. IBM3624. - 5. Diebold
F2	N	1	Output PIN block format: - 0. ISO0. - 1. ISO1. - 2. ISO2. - 3. ISO3. - 4. IBM3624. - 5. Diebold.
PIN	H	16	Encrypted PIN block, received from ATM.
L1	N	2	Next field length.
D1	N	var	PAN.
L2	N	2	(*)Next field length.
D2	H	var	(*) Random number used to form the outgoing PIN block.
D3	H	1	(2*) Padding value.
ND	N	1	Number of derivation data of K1 (0..9).
DD1	H	16	Derivation data 1.
…	…	…	…
DDN	H	16	Derivation data nd.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return Value.
PIN	H	16	Encrypted PIN block.

(*) Only if F2=1 or F2 = 3.D2 maximum length is 16.

(2*) Only if F2 = 4.

3.4.3. Offset Calculation - 0604 - (*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0604.
K1	K	var	PIN Generation Key (under LMK4).
K2	K	var	PIN Transport Key (under LMK3).
TD	N	16	Decimation Table
F	N	1	Input PIN Block format: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.
ALG	N	1	PIN generation Algorithm:: - 1 IBM3624-PIN Offset. - 2 Netherlands PIN 1. - 4 IBM German Bank Pool Institution.
PIN	H	16	Encrypted PIN Block.
L1	N	2	Next field Length.
PAN	N	var	PAN.
PAD	H	2	(*)Padding.
LPIN	N	1	(2*) PIN Length.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
OFFSET	N	4	Returned offset value.

(*) Only if F = 4.

(2*) Only if ALG = 1.

3.5. PVV Calculation - 0608 - (*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0608.
K1	K	var	PIN Generation Key (under LMK4).
K2	K	var	PIN Transport Key (under LMK3).
F	N	1	Input PIN Block format: - 0 ISO0. - 1 ISO1. - 2 ISO2. - 3 ISO3. - 4 IBM3624. - 5 Diebold.
PIN	H	16	Encrypted PIN Block.

Data	Type	Length	Comment
L1	N	2	Next field Length.
PAN	N	var	PAN.
PAD	H	2	(*)Padding.
PVK	N	1	Used Key index digit.

Response message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
OFFSET	N	4	Returned PVV value.

(*) Only if F = 4.

3.6. PINs Export(*CryptosecBANKING)

3.6.1. Export PIN - 0605 -(*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0605.
K1	K	var	PIN Calculation Key (under LMK4).
K2	K	var	PIN Export Key (under LMK3).
TD	N	16	Decimation table.
F	N	1	Output PIN format: - 0 format ISO0. - 3 format ISO3.
LPN	N	2	Next field Length.
PAN	N	var	PAN.
LPIF	N	2	Next field length, up to 12.
OFF	N	var	Offset.
D1	H	10	(*)Random number to build PIN block.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
PIN	H	16	Encrypted PIN Block.

(*) Only if F = 3.

3.6.2. PIN Printing - 0609 - (*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0609.
K1	K	var	PIN Encryption key (under LMK3).
F1	N	1	Input PIN block format: - 0. ISO0. - 1. ISO1. - 2. ISO2. - 3. ISO3. - 4. IBM3624. - 5. Diebold.
PIN	H	16	Encrypted PIN block.
L1	N	2	Next field Length.
D1	N	var	PAN.
L2	N	4	Next field Length (bytes).
T	A	var	Variable user data table (name, custodian, ATM, dates, uses, key…). Format will depend on printing format already defined.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.

Notes about printing

You must have previously defined template for printing, using command 1301.

In PIN printing, argument T contains printing strings 0 to 15, whichare referenced in string format for printing. It is not necessary to define all 16 strings, however, one must take into account that they are numbered consecutively starting from0. Sequence “\0” is used as string delimiter. Symbol ^P induces PIN printing.

To clarify the PIN printing operation, see the below example. Suppose that you want to send yourPIN to Mr Thomas M Smith. It is advisedto print both the customer’s address as well as your PIN, in the following manner:

THOMAS M SMITH

APT 4B XXXX

39 ELM DR

MEDIA PA 19063

YOUR FULL SERVICE BANK

Where XXXX is the PIN and a final corporative messagehas been added.If you have to send similar information to other customers,you can consider that format as a template, i.e. the position of the different printing fields, for all customers. Similarly, corporative message is also the same for all types of printing. The easiest way to resolve all of this is including thecorporative message in template. (However, it is possible to include it as an extra printing field if you prefer, because of space or otherwise, it can be seen in Export byComponents command, 3.1.12). Templatewill be in this case:

>L>013^0>L>013^1>041^P>L>013^2>L>013^3>L>013YOUR FULL SERVICE BANK>L>F

Where it can be seen how ^0 denotes the entry point for the first string in the data table, ^1 denotes the entry point for the second string in thetable and so on. ^P denotes the entry point for the PIN.

If we want to print the desired format with this template, the data table to be included in this command will be:

THOMAS M SMITH\0APT 4B\039 ELM DR\0MEDIA PA 19063

Where \0 is used as a separator, and it is not necessary to include it after the last string. This string length is 00000049.

The alternative is to position the message outside the template and include another unspecified string in the template (^4 in this chase). The string has to be sent to the printing command in that case for each print performed.

3.7. Validation Codes Calculation(*CryptosecBANKING)

3.7.1. Validation Codes Calculation and Verification - 0701 -(*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0701.
K	K	var	CVV/CVC/CSS Key (under LMK5).
D_1	N	4	Expiration Date.
D_2	N	3	Service Code.
PAN	N	16	PAN.
M	N	1	Mode : - 0 Calculation. - 1 Verification.
CVV	N	3	(*)CVV/CVC/CSS to verify.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
CVV	N	3	(2*)Calculated Validation code.
VER	N	1	(*)CVV/CVC/CSS Verification: - 0 correct. - 1 incorrect.

(*) Only if M = 1.

(2*) Only if M = 0.

3.7.2. Calculation and Verification of CVC 3 – 0702 - (*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0702.
K1	K	var	CVC3 Key(under LMK 5)
M	N	1	Mode Indicator: - 0 Calculation. - 1: Verification.
D1	H	16	Calculation Data 1.
D2	H	12	Calculation Data 2.
L3	N	2	Next field Length.
D3	H	var	Calculation Data 3.
CVC3	N	3	(*)CVC3 to verify.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
CVC3	H	4	(2*) Calculated CVC3.
VER	N	1	(*)CVC 3 Verification: - 0 – correct. - 1 – incorrect.

(*) Only if M = 1.

(2*) Only if M = 0.

3.7.3. Calculation and Verification of CSC– 0703 - (*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0703.
K1	K	var	CSC key (under LMK 5)
M	N	1	Indicates operation mode: - 0 Calculation. - 1 Verification.
PAN	N	16	PAN
D1	N	4	Expiration Date.
L2	N	2	(*)Next field length.
CSC	N	var	(*)CSC to verify.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
VER	N	1	(*)Verification of CSC: - 0 correct. - 1 incorrect.
CSC3	H	3	(2*)Calculated CSC3.
CSC4	H	4	(2*) Calculated CSC4.
CSC5	H	5	(2*)Calculated CSC5.

(*) Only if M = 1.

(2*) Only if M = 0.

3.8. Securing Messages

3.8.1. Securing Messages – 0801 - (*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0801.
K1	K	var	MAC Key (under LMK 8).
I1	N	1	Algorithm Indicator: - 0: MAC ANSI X9.9-1. - 1: MAC ANSI X9.19-1. - 2: MAC TDES.
I2	N	1	Padding Method Indicator: - 0: method 1 ISO/IEC 9797. - 1: method 2 ISO/IEC 9797.
I3	N	1	Previous Hash(if necessary) Indicator: - 0: No Hash. - 1: SHA-1.

Data	Type	Length	Comment
			- 2: MD5.
IV	H	16	Initialization Vector.
L1	N	6	Next field Length.
D1	H	var	Data to sign.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
MAC	H	16	Calculated MAC.

3.8.2. Securing Point of Sale Terminal Messages– 0802 - (*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0802.
K1	K	var	MAC key (under LMK 8).
ND	N	1	(*) Number of derivation data of K1 (0...9).
DD1	H	16	(*) Derivation data 1.
…	…	…	…
DDND	H	16	(*) Derivation data nd.
I1	N	1	Algorithm Indicator: - 0: MAC ANSI X9.9-1. - 1: MAC ANSI X9.19-1. - 2: MAC TDES.
I2	N	1	Padding Method Indicator: - 0: method 1 ISO/IEC 9797. - 1: method 2 ISO/IEC 9797.
I3	N	1	Previous Hash(if necessary) Indicator: - 0: No Hash. - 1: SHA-1. - 2: MD5.
IV	H	16	Initialization Vector.
L1	N	6	Next field Length.
D1	H	var	MAC data.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
MAC	H	16	Calculated MAC.

(*) As derivation data fields as indicated by number in field ND.

3.8.3. HMAC – 0803 - (*CryptosecBANKING)

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0803.
K1	K	var	MAC Key (under LMK 8).
I1	N	1	Hash algorithm Indicator: - 0: MD5. - 1: SHA-1. - 2: SHA-256.
L1	N	6	Next field Length.
D1	H	var	Data.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
HMAC	H	(*)var	Calculated HMAC.

(*) Depends on the hash algorithm selected as follows:

- MD5: 32.

- SHA-1: 40.

- SHA-256: 64.

3.9. Data Encryption and Decryption

3.9.1. DES Encryption and Decryption - 0901 -

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 0901.
K1	K	var	Key to use (under LMK 7).
I1	N	1	Function: - 1 Encrypt. - 0 Decrypt.
I2	N	1	Algorithm to use: - 1 ECB. - 0 CBC (Initialization Vector, zeros).
L1	N	6	Next field Length.
D1	H	var	Data toencrypt/decrypt.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.

Data	Type	Length	Comment
RV	H	8	Return Value.
LCIF	N	6	Next field Length.
CIF	H	var	Encrypted/Decrypted Data.

3.9.2. DES Encryption and Decryption v2 - 0902 - (*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0902.
K1	K	var	Key to use (under LMK 7).
ND	N	1	(*) Number of derivation data of K1 (0...9).
DD1	H	16	(*) Derivation data 1.
…	…	…	…
DDND	H	16	(*) Derivation data nd.
I1	N	1	Function Indicator: - 1 Encrypt. - 0 Decrypt.
I2	N	1	Algorithm: - 1 ECB. - 0 CBC.
IV	H	16	(2*) Initialization vector.
L1	N	6	Next field length.
D1	H	var	Data to encrypt/decrypt.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
LCIF	N	6	Next field length.
CIF	H	var	Encrypted/Decrypted data.

(*) As derivation data fields as indicated by number in field ND.

(2*) Only if I2=0.

3.9.3. DES Encryption and Decryption v3 - 0903 -

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 0903.
K1	K	var	Key to use (under LMK 7).
I1	N	1	Function indicator:

Data	Type	Length	Comment
			- 1 Encrypt. - 0 Decrypt.
I2	N	1	Algorithm: - 1 ECB. - 0 CBC.
IV	H	16	(*) Initialization vector.
L1	N	6	(2*) Next field length.
D1	H	var	(3*) Data to Encrypt/Decrypt.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
LCIF	N	6	Next field length.
CIF	H	var	Encrypted/Decrypted data.

(*) Only if I2 = 0.

(2*)If I1=0 or 1, data length shall be multiple of 16.

If I1=2 or 3 data length shall be multiple of 8.

(3*)If I1=0 or 1, data must be hexadecimal. Data shall be compressed before encryption/decryption.

If I1=2 or3, data can be binary. Data shall be encrypted/decrypted as received.

3.9.4. Securing Password - 0904 -

Submission message:

Dato	Tipo	Longitud	Comentario
IC	H	4	Command Identifier. Value 0904.
K	K	var	Key to use (under LMK 17).
LEN	N	2	Next field Length.
P	B	var	Password.

Response message:

Dato	Tipo	Longitud	Comentario
IC	H	4	Command Identifier.
RV	H	8	Return Value.
LEN	N	2	Next field Length.
PP	B	var	Protected Password.

3.9.5. Password Verification - 0905 -

Submission message:

Dato	Tipo	Longitud	Comentario
IC	H	4	Command Identifier. Value 0905.
K	K	var	Key to use (under LMK 17).
L1	N	2	Next field Length.
P	B	var	Password to verify.
N1	N	2	Number of protected passwords to be passed.
LP1	N	2	Length of the password #1
PP1	H	var	Protected password #1.
…	….	…..	……..
LPN	N	2	Length of the password #N.
PPN	H	var	Protected password #N.

Response message:

Dato	Tipo	Longitud	Comentario
IC	H	4	Command Identifier.
RV	H	8	Return Value.
VER	N	1	Verification of password: - 1 correct. (One of the protected password has been verified succesfully) - 0 incorrect. (None of the protected password has been verified succesfully)

3.9.6. RSA Encryption and Decryption- 1001 -

For new implementations we suggest the use of command 1002.

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 1001.
K1	K	var	Key to use (under LMK 60).
I2	N	1	Function Indicator: - 0 Encrypt with Public Key. - 1 Decrypt with Private Key.
L1	N	6	Next field Length.

Data	Type	Length	Comment
D1	H	var	Data toencrypt/decrypt.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
LCIF	N	6	Next field Length.
CIF	H	var	Encrypted/Decrypted Data.

3.9.7. RSA Encryption and Decryption v2 – 1002 -

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 1002.
K1	K	var	Key to use (under LMK 60).
I1	N	1	Function Indicator: - 0 Encrypt with Public Key. - 1 Decrypt with Private Key.
I2	N	1	Algorithm Identifier: - 0 PKCS#1 v1.5. - 1 PKCS#1 OAEP.
L1	N	6	Next field Length.
D1	H	var	Data toencrypt/decrypt.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
LCIF	N	6	Next field Length.
CIF	H	var	Encrypted/Decrypted Data.

3.10. Data Signature and Verification

3.10.1. RSA Signature and Verification – 1101 -

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 1101.
K1	K	var	Key to use (under LMK 60).
M	N	1	Function indicator: - 0 Sign with Private Key. - 1 Verificate with Public Key.
AL1	N	1	(2*)HASH function indicator: - 0 SHA1. - 1 MD5
AL2	N	1	(2*) Signature algorithm indicator: - 0 pkcs#1 v1.5. - 1 X9.31.
L1	N	6	Next field Length.
D1	H	var	Data to sign or to be verified.
L1	N	6	(*)Next field Length.
D1	H	var	(*)Sign.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
LFIR	N	6	(2*)Next field Length.
FIR	H	var	(2*)Signed Data.
VER	N	1	(*) Verification result: - 0 – correct. - 1 – incorrect.

(*) Only if M = 1.

(2*) Only if M = 0.

3.10.2. Asymmetric Key Signature – 1103 -

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 1103.
K1	K	var	Key to use (under LMK 61 or 62).
MEC	H	2	Hash mechanism: - 00 No Hash - 01 MD5. - 10 SHA-1 - 11 SHA-224 - 12 SHA-256 - 13 SHA-386

Data	Type	Length	Comment
			- 14 SHA-512
ALG	H	2	Signature mechanism: - 01 RSA.
PAD	N	2	Padding indicator of block to sign - 00Raw - 01 pkcs#1 v1.5. - 10 X9.31.
L1	N	6	Next field length.
D1	H	var	Data to sign, already digested (HASH) with mechanism indicated in MEC

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
LFIR	N	6	Next field length.
FIR	H	var	Signature

3.10.3. Verification with Asymmetric Key – 1104 –

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 1104.
K1	K	var	Key to use (under LMK 61 or 62).
MEC	H	2	Hash mechanism: - 00 No Hash - 01 MD5. - 10 SHA-1 - 11 SHA-224 - 12 SHA-256 - 13 SHA-386 - 14 SHA-512
MEC	H	2	Signature mechanism: - 01 RSA.
AL2	N	2	Padding indicator of block to sign - 00 Raw - 01 pkcs#1 v1.5. - 10 X9.31.
L1	N	6	Next field length.
D1	H	var	Signed and resumed data about mechanism specified in MEC
L1	N	6	Next field length.
D1	H	var	Signature

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
VER	N	1	Verification result:

Data	Type	Length	Comment
			- 0 – correct. - 1 – incorrect.

3.11. Cash Back and Transport Certificates Generation for TIBC or Advantis

3.11.1. Cash Back and Transport Certificates Generation– 1201 -(*CryptosecBANKING)

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 1201
M	N	1	Action: 0 Generate certificate for TIBC or Advantis Cash Back. 1 Generate certificate for TIBC or Advantis (Pass 2) Transport. 2 Generate certificate for Advantis (Pass 1) Transport.
B0	H	16	Data block (Block0).
B1	H	16	Data block (Block1).
B2	H	16	(*)Data block (Block2).
B3	H	16	(2*) Data block (Block3).
Kc	K	var	Load key.
PAN	N	16	Card number.
ID	N	4	Load file identifier.
RN1	H	16	(3*)Random number 1.
RN2	H	16	Random number 2.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
CC	H	16	Certificate generated.

(*) Only if M=0 or M=1

(2*) Only if M=1

(3*) Only if M=0

3.12. Load of Printing Template– 1301 -

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 1301.
LEN	N	3	Printing template length. Up to 400 characters.
STR	H	var	Printing template.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.

3.13. HSM Test

3.13.1. HSM Test – 1401 -

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 1401.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
KCV	N	6	First three bytes from HSM ID.

3.14. Key Diversification

3.14.1. Key Diversification – 1601 -

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 1601.
Kd	K	var	(*) Key to diversify (under corresponding LMK)
I_0	N	1	(2*)Check Key Parity: - 0 Not check. - 1 Check.
I_1	N	2	Diversify Algorithm Identifier: - 12 TDES II. - 22 SIM Keys.
LMK	N	2	(3*)Number of LMK under which the key is encrypted.
I_2	N	1	(4*)Parity adjustment of diversified key Indicator: - 0 Do not adjust parity of diversified key. - 1 Adjust parity of diversified key.
L_1	N	2	(6*)Diversify Data Length 1.
D_1	H	N	(6*)Diversify Data 1.
L_2	N	2	(6*)Diversify Data Length 2.
D_2	H	N	(6*)Diversify Data 2.
ICCID	H	20	(7*)ICCID of the SIM.
F_1	A	1	(5*)Diversified Key Internal storage Indicator : - S.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
Kr	K	var	Diversified Key.

- (*) Key to diversify. Only can diversify keys that are encrypted under LMK 2, 3,6, 7, 8, 9, 10, 14 and 99.
- (2*) Only can give the option of check key parity if the key to diversify is encrypted under LMK09 orLMK99.
- (3*) Diversified keys are only encrypted under LMK 2, 3, 6, 7, 8, 9, 10, 14 and 99. Moreover only if the key to diversify is type 99, the diversified key must be the same type.
- (4*) This indicator is only present if diversified key is encrypted under LMK09 orLMK99.
- (5*) This indicator is only present if diversified key have to be stored internally.
- (6*) Only for algorithm identifier 12.
- (7*) Only for algorithm identifier 22.

3.15. Random Number

3.15.1. Random Number Generation – 1701 -

Submission Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 1701.
LEN	N	6	Random Number Length in bits.

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier.
RV	H	8	Return Value.
LEN	N	6	Next Data Length.
RND	H	var	Generated Random Number.

3.16. Digest Function (HASH)

3.16.1. Hash – 2000 -

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 2000.
MEC	H	2	Hash Mechanism: - 01 MD5. - 10 SHA-1 - 11 SHA-224 - 12 SHA-256 - 13 SHA-386 - 14 SHA-512

Data	Type	Length	Comment
FLAG	N	2	Hash status flag: - 01 Init. - 02 Update - 03 Final
F1	N	2	Data Format: - 01 Hexadecimal. - 02 Binario
L1	N	6	Next field length. (*)
D1	H	var	Intermediate data (status). (*)
L2	N	6	Next field length.
D2	H	var	Data.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
FR1	H	2	Received data format: - 01 Hexadecimal. - 02 Binary.
LR1	N	6	Next field length. (2*)
DR1		var	Intermediate data (estatus). (2*)
LR2	N	6	Next field length. (3*)
DR2		var	Hash. (3*)

(*) Only if FLAG = 02 or 03.

(2*) Only if FLAG = 01 or02

(3*) Only if FLAG = 03

3.17. POS Keys

3.17.1. POS Terminal Key Request– 6001 - (*CryptosecBANKING)

Submission message:

Dat	Type	Length	Comment
IC	H	4	Command identifier. Value 6001.
MOD	N	1	CTC Operation mode: - 0 Initialize CTC - 1 Renew CTC
K1	K	var	Initialization Master key. (*)
K2	K	var	Old Transport Key. (2*)
NKM	N	1	Number of requested keys (1..5) (3*)
OPMn	N	1	Generation mode: - 0 Derive - 1 Random
KMn	K	Var	Master key to derive. (4*)
LMKn	N	2	LMK label associated to the requested key. Must be different to 00.(5*)

Dat	Type	Length	Comment
SPP	H	32	PINPAD serial number(6*)
NKS	N	1	Number of CPE requested keys (0...2) (7*).
OPSn	N	1	Generation mode: - 0 Derive - 1 Random
KSn	K	Var	Master key to derive. (8*)
SSM	H	32	SAM serial number (8*)

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
NK	N	2	Number of keys
Kn	K	var	n key generated (9*)

(*) Only present if MOD=0

(2*) Only present if MOD=1

(3*) Number of requested keys. At least the CTC key has to be requested.

This group may ask:

- Keys' Transport key (CTC)
- PIN Transport key (CP)
- MAC key (CM)
- Software Load key (CAT)
- Data Encryption key (CAU)

The fields OPMn, KMn and LMKn depends on the number of keys requested and ALWAYS the first key is a CTC.

(4*) Only present if OPMn=0

(5*) Only present if OPMn=1

(6*) Only present if MOD=0 and/or OPMn=0

(7*)Number of keys to be requested. This group may ask:

- PINoffset Transport key, communication between PINPAD and card (CPE1 y CPE2)

There must be as fields OPSn and KSn as many keys.

(8*) Only present if OPSn=0

(9*) So many key blocks as indicated in NK field in the order as they were applied for.

3.18. Decimalization Tables

3.18.1. Decimalization Table Update - 2101 - (*CryptosecBANKING)

Re-encrypt a Decimalization Table (DT) that is encrypted under a LMK from an old set .This makes it possible to upgradeall the DTs that are externally stored every the master key is changed.

To make this command run, you must take into account:

- A new module master key must be loaded.
- A DT must be stored under a LMK associated with the previous master key.
- The previous LMKs shouldn't be deleted before executing this command. Once deleted, you can only recover DT encrypted under that deleted key by reloading the associated master key.

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 2101.
TD	H	16	DT encrypted with old LMK,

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.
TD	H	16	DT encrypted with new LMK.

3.19. Log Management

3.19.1. Send Log – 2201 –

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 2201.
A	N	8	Log date (YYYYMMDD)
S	N	8	Line index from which log is sent.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.

Data	Type	Length	Comment
N	N	4	Number of lines returned.
F	L	8	Line length.
D	A	var	Log line
RES	N	8	(*)Index to the last line sent.

(*)This value is nonzero if the total length of the response is close to 512KB, i.e. if it was not possible to send from the indicated line to the end of the file.

3.19.2. Log Index – 2202 –

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 2202.

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return Value.
N	N	4	Number of log files found.
D	A	Var	Log files found (YYYYMMDD).

3.19.3. Delete Logs File – 2203 –

Submission message:

Data	Type	Length	Comment
IC	H	4	Command identifier. Value 2203.
A	N	8	Log file date to delete (YYYYMMDD)

Response message:

Data	Type	Length	Comment
IC	H	4	Command identifier.
RV	H	8	Return value.

3.20. X509 Certificates Management

3.20.1. Request Generation (PKCS#10) – 2301 –

Submission message:

Data	Type	Length	Comment
IC	H	4	Command Identifier. Value 2301.
L	N	4	Number of bits of the key to be generated: - Must be 1024 or 2048 bits.
E	N	1	(*)Public Exponent of the key to be generated: - 0 if exponent 3. - 1 if Fermat4.
I	N	2	Number of attributes of the subject.
LO1	N	2	OID Length from the attribute #1 (*)
O1	A	var	OID of the attribute #1 (*)
LN1	N	2	Length of the OID's short name from attribute #1
N1	A	var	OID's Short Name of the atrribute #1
LV1	N	2	Length of the value of the attribute#1
V1	A	var	Value of the attribute #1
…	…	….	……
LON	N	2	OID Length from the attribute #n (*)
ON	A	var	OID of the attribute #n(*)
LNN	N	2	Length of the OID's short name from attribute #n
NN	A	var	OID's Short Name of the atrribute #n
LVN	N	2	Length of the value of the attribute #n
VN	A	var	Value of the attribute #n
LPASS	N	2	Length of the passwordChallenge
PASS	A	var	PasswordChallenge
LPK	N	2	Length of the next field
LABPK	A	var	Public Key Label
LPRK	N	2	Length of the next field
LABPRK	A	var	Private Key Label
OF	N	1	Request Format: - 0 if DER - 1 if PEM

Response Message:

Data	Type	Length	Comment
IC	H	4	Command Identifier
RV	H	8	Return Value
L1	N	4	Length of the request
R1	B	var	Request

(*) These values must inform in case of new or propietarie's OIDs. If the OID to be used exists, the length of the filed must be zero (00) with no value at all.

4. ERRORCODES

Error codes returned by the HSM when a command is executed are shown in the following table:

Code	Description
0x00000000	OK
0x00000001	MESSAGE FORMAT ERROR
0x00000002	SERVICE UNAVAILABLE
0x00000005	PC MEMORY ERROR
0x00000018	TIME OUT
0x00000019	INVALID FIRMWARE
0x0000001A	DRIVER ERROR
0x0000001B	ARGUMENT ERROR
0x0000001C	OPEN WD ERROR
0x0000001D	INCORRECT WD VERSION
0x00000023	NO FIRMWARE
0x00000025	DATA MISSALIGNMENT
0x00000026	CIPHER SESSION NOT INITIALIZED
0x00013000	HASH ALGORITHM NOT SUPPORTED
0x00013400	SIGNATURE FORMAT NOT SUPPORTED
0x00001000	ERROR TEST HASH SHA1
0x00001400	ERROR TEST HASH MD5
0x00001800	ERROR TEST HASH RIPEMD 128
0x00001C00	ERROR TEST HASH RIPEMD 160
0x00002000	ERROR DES ECB S
0x00002400	ERROR DES ECB D
0x00002800	ERROR DES ECB T
0x00002C00	ERROR DES CBC S
0x00003000	ERROR DES CBC D
0x00003400	ERROR DES CBC T
0x00003800	ERROR DES CFB64 S
0x00003C00	ERROR DES CFB64 D
0x00004000	ERROR DES CFB64 T
0x00004400	ERROR DES OFB64 S
0x00004800	ERROR DES OFB64 D
0x00004C00	ERROR DES OFB64 T
0x00006400	PAIR WISE CONSISTENCE TEST ERROR
0x00006800	SIGN VERIFY TEST ERROR
0x00007C00	WRONG NUMBER OF CUSTODIANS
0x00008C00	DEFAULT PIN ERROR
0x00009000	PIN SIZE ERROR

Code	Description
0x00009800	USER EXISTS ERROR
0x00009C00	MEMORY USER FULL ERROR
0x0000A000	USER NOT EXISTS ERROR
0x0000A800	INVALID LEN DES KEY ERROR
0x0000AC00	CONFIG BCHU ERROR
0x0000B000	INVALID DES KEY ERROR
0x0000B400	RSA KEYS GENERATION ERROR
0x0000B800	INVALID DATA SIZE ERROR
0x0000BC00	PUBLIC KEY CIPHER ERROR
0x0000C000	WRONG ALGORITHM SELECTED ERROR
0x0000C400	PRIVATE KEY CIPHER ERROR
0x0000CC00	KEY NOT EXISTS
0x0000D400	INVALID KEY NUMBER
0x0000E000	WRONG COMMAND ERROR
0x0000EC00	LEN RSA KEY ERROR
0x0000F000	PASSWORD ERROR
0x00010000	WRONG PAD CHARACTER ERROR
0x00010400	WRONG PINBLOCK ERROR
0x00010800	INVALID LEN PIN ERROR
0x00010C00	TOO MANY ONES ERROR
0x00011000	INVALID LEN PAN ERROR
0x00011400	INVALID LEN CVV ERROR
0x00011800	WRONG PAN ERROR
0x00011C00	WRONG PARAMETER ERROR
0x00012000	WRONG PINBLOCK FORMAT ERROR
0x00012400	PASSWORD SIZE ERROR
0x00012C00	ERROR RNG TEST
0x00013000	HASH ALGORITHM NOT SUPPORTED
0x00013400	SIGNATURE FORMAT NOT SUPPORTED
0x00015800	LNAU HARDWARE ERROR
0x00015C00	WRONG CONFIGURATION DATA
0x00016000	WRONG CMM KEY ERROR
0x00016400	WRONG OPTIONS INDICATOR ERROR
0x00016800	WRONG METHOD INDICATOR ERROR
0x00016C00	WRONG DINAMIC OBE ERROR
0x00017000	VERIFICATION FAILED
0x00017400	PRINT STRING TOO LONG ERROR
0x00017800	WRONG PRINT STRING ERROR
0x00017C00	WRONG PKCS8 INFO ERROR
0x00018000	WRONG DECIMALIZATION TABLE ERROR
0x00018400	NOT ALLOWED IN PRODUCTIONSTATE
0x00018800	INVALID LEN NA ERROR
0x00019000	PRINTER NOT ENABLED ERROR

Code	Description
0x00019400	OLD CMMS NOT EXIST ERROR
0x00019800	PINVERIFICATIONLOCKED
0x00019C00	PINBLOCKLOCKED
0x0001A000	NO LOG SPACE ERROR
0x0001A400	CARDACCESSERROR
0x0001A800	CARDTIMEOUTERROR
0x0001AC00	CARDAUTHENTICATIONERROR
0x0001B000	CARDPINERROR
0x0001B400	CARDCANNOTCHANGEPINERROR
0x0001B800	CARDSETERROR
0x0001BC00	REPEATEDCARDERROR
0x0001C000	BLANKCARDERROR
0x0001C400	CARDALREADYUPDATEDERROR
0x0001C800	EXCEPTION ERROR
0x0001CC00	INPUT DATA TIMEOUT ERROR
0x0001D000	OUTPUT DATA TIMEOUT ERROR
0x0001D400	HSM SERVING CONSOLE COMMAND ERROR

5. COMMANDS EXPLAINED

5.1. DACCalculation

DAC = (TDES(K) [D1])$_{2MSB}$

5.2. IDNCalculation

KIDN-LEFT = TDES (K) [D1]

KIDN-RIGHT = TDES (K) [$\overline{\text{D1}}$]

KIDN = KIDN-LEFT || KIDN-RIGHT

IDN = TDES (KIDN) [D2]

5.3. ARQC and ARPCCalculation and Verification

KAC-LEFT = TDES (K) [D18LSB]

KAC-RIGHT = TDES (K) [$\overline{\text{D18LSB}}$]

KAC = KAC-LEFT || KAC-RIGHT

ForMasterCard:

SKAC-LEFT = TDES (KAC) [D2 || F000 || D3]

SKAC RIGHT = TDES (KAC) [D2 || 0F00 || D3]

SKAC = SKAC-LEFT || SKAC-RIGHT

Calculation of the ARQC for Visa:

ARQC = MAC(KAC) [D4]

Padeo de D4: D4 || 00...00

Calculation of the ARQC for MasterCard:

ARQC = MAC(SKAC) [D4]

D4 Padding: D4 || 8000...00

Calculation of the ARPC:

ARPC = TDES(KAC) [D5 XOR D6]

5.4. Script Signature

KSF-LEFT = TDES (K) [D18LSB]

KSF-RIGHT = TDES (K) [$\overline{\text{D18LSB}}$]

KSF = KSF-LEFT || KSF-RIGHT

For Visa:

SKSF-LEFT = (KSF-LEFT) XOR (D2)

SKSF-RIGHT = (KSF- RIGHT) XOR (000000000000 || (D22LSB XOR FFFF)

SKSF = SKSF-LEFT || SKSF-RIGHT

For MasterCard:

SKSF-LEFT = TDES (KSF) [Bytes 8/7 D2 || F0 || Bytes 5/1 D2]

SKSF-RIGHT = TDES (KSF) [Bytes 8/7 D2 || 0F || Bytes 5/1 D2]

SKSF = SKSF-LEFT || SKSF-RIGHT

Sign:

F = MAC (SKSF) [D3]

D3 Padding: D3 || 8000...00

5.5. ScriptEncryption.

KSC-LEFT = TDES (K) [D18LSB]

KSC-RIGHT = TDES (K) [$\overline{\text{D18LSB}}$]

KSC = KSC-LEFT || KSC-RIGHT

For Visa:

SKSC-LEFT = (KSC-LEFT) XOR (D2)

SKSC-RIGHT = (KSC- RIGHT) XOR (000000000000 || (D22LSB XOR FFFF)

SKSC = SKSC-LEFT || SKSC-RIGHT

For MasterCard:

SKSC-LEFT = TDES (KSC) [Bytes 8/7 D2 || F0 || Bytes 5/1 D2]

SKSC-RIGHT = TDES (KSC) [Bytes 8/7 D2 || 0F || Bytes 5/1 D2]

SKSC = SKSC-LEFT || SKSC-RIGHT

Visa Encryption:

F = TDESECB (SKSC) [D3]

Padding D3: D3 || 8000...00

MasterCard Encryption:

F = TDESCBC (SKSC) [D3]

Padding D3: D3 || 8000...00

5.6. PIN Exchange Script.

Three steps are required for encryption.

PIN block calculation.

F2=0: PIN block I (only Mastercard)

D' = TDES-1(KT)[D1]

D'' = extract from D' new PIN value according to F1

PINblock I (8 bytes) = 2 || length D'' || D'' || stuffed to Fs

F2=3: PIN block IV (only Mastercard)

D' = TDES-1(KT)[D1]

D'' = extract from D' new PIN value according to F1

PINblock IV (8 bytes) = 0 || length D'' || D'' || stuffed to Fs

F2=1: PIN block II (only Visa)

D' = TDES-1(KT)[D1]

D'' = extract from D' new PIN value according to F1

PINblock II (8 bytes) = (PINblock A) XOR (PINblock B) being:

PINblock A (8 bytes) = 00 00 00 00 || 4 bytes less significative fo the first semi-key of MKAC (MKAC-LEFT):

MKAC-LEFT = TDES (K1) [PAN || PSN]

PINblock B (8 bytes) = 0 || length(D'') || D'' || stuffed to Fs

F2=2: PIN block III (only Visa)

D' = TDES-1(KT)[D1]

D'' = extract from D' new PIN value according to F1

PIN block III (8 bytes)=(PINblock A) XOR (PINblock B) XOR(PIN block C) being:

PINblock A calculated before.

PINblock B calculated before.

PINblock C (8 bytes) = D0'' || stuffed to 0s being D0'' old PIN value obtained:

D0'=TDES-1(KT)[D2]

D0'' = extract from D0' older PIN value according to F1

- **PIN block encryption.**

Encryption key SKSMC Calculation

Obtains the confidentiality-script key from card:

MKSMC = MKSMC-LEFT || MKSMC-RIGHT

MKSMC-LEFT = TDES (K2) [PAN || PSN]

MKSMC-RIGHT = TDES (K2) [PAN || PSN']

where:

PAN || PSN' = (PAN || PSN XOR FFFFFFFFFFFFFFFF)

Diversification of this key will be confidentiality-script session key:

- **Mastercard scheme:**

SKSMC = SKSMC-LEFT || SKSMC-RIGHT

SKSMC-LEFT = TDES (MKSMC) [D4']

SKSMC-RIGHT = TDES (MKSMC) [D4'']

Being:

D4' = Bytes 8/7 D4 || F0 || Bytes 5/1 D4

D4'' = Bytes 8/7 D4 || 0F || Bytes 5/1 D4

Visa scheme:

SKSMC = SKSMC-LEFT || SKSMC-RIGHT

SKSMC-LEFT = (MKSMC -LEFT) XOR (D4)

SKSMC-RIGHT = (MKSMC -RIGHT) XOR (D4')

Being:

D4' = 000000000000 || (D4 – 2LSB XOR 'FF FF')

- **Script PINblock Encryption**

- **Mastercard scheme and F2=0:**

PINblock Script-Encryption= TDESCBC (SKSMC) [PINblock I]

- **Mastercard scheme and F2=3:**

PINblock Script-Encryption= TDESCBC (SKSMC) [PINblock IV]

- **Visa scheme and F2=1:**

PINblock Script-Cifrado = TDESECB (SKSMC) [08 || PINblock II || 80000000000000]

- **Visa scheme and F2=2:**

PINblock Script-Encryption = TDESECB (SKSMC) [08 || PINblock III || 80000000000000]

- **PIN exchange script signature.**

Sign SKSMI key Calculation

Obtaining the integrity key-card script: Obtained as the key privacy-card script, using K3 instead of K2.

The diversification of this key will result in the session key for integrity-script:

- **Mastercard scheme:**

SKSMI = SKSMI-LEFT || SKSMI-RIGHT

SKSMI-LEFT = TDES (MKSMI) [D4']

SKSMI-RIGHT = TDES (MKSMI) [D4'']

Being:

D4' = Bytes 8/7 D4 || F0 || Bytes 5/1 D4

D4'' = Bytes 8/7 D4 || 0F || Bytes 5/1 D4

- **Visa scheme:**

SKSMI = SKSMI-LEFT || SKSMI-RIGHT

SKSMI-LEFT = (MKSMI -LEFT) XOR (D4)

SKSMI-RIGHT = (MKSMI -RIGHT) XOR (D4')

Being:

D4' = 000000000000 || (D4 – 2LSB XOR 'FF FF')

- **Sign PIN exchange Script**

Script Sign = MAC (SKSMI) [D5 || PINBlock Script-Encryption]

Used MAC algorithm, will be defined in ANSI X9.19-1.

As for the stuffing to use, it will be built in accordance with the method 2 of the ISO/IEC 9797: adding a byte to the right with the hex value '80'; then is added to the right the fewer number of bytes '00'(hexadecimal) that ensure that the length of the final message, original message with stuff, is multiple of 8 bytes

5.7. Validation Codes Calculation.

5.7.1. CSS/CVV/CVC Calculation and validation.

K = KLEFT || KRIGHT

B1 = PAN

B2 = D1 || D2 || 0 00 00 00 00

R1 = DES (KLEFT) [B1]

R2 = R1 (XOR) B2

R3 = DES (KLEFT) [R2]

R4 = DES-1 (KRIGHT) [R3]

R5 = DES (KLEFT) [R4]

R6 = Extract from R5 numeric digits (0-9), from left to right. Flush left these digits in a 16 positions field.

R7 = Extract from R5 characters (A-F), from left to right. Decimation each of these digits subtracting 10 (decimal).

Concatenate this result, with the previous step result:

R8 = R6 || R7

CVC/CVV/CSS consists of the three first digits (from the left) of R8

5.7.2. CVC3 Calculation and Validation.

KLEFT = TDES (K1) [D1]

KDER = TDES (K1) [$\overline{D1}$]

K = KLEFT || KDER

D = (MAC(K)[D3])2LSB || D2

CVC3 = (TDES(K)[D])2LSB

5.8. Cash Back and Transport Certificates GenerationTIBC

5.8.1. Cash Back Certificate.

DD= ID || PAN4-15

Ktj = DESECB[Kc](DD)

Ktr = DESECB[Ktj](RN1)

CM=MAC|RN2[Ktr](Bloque0 || Bloque1 || Bloque2)

5.8.2. Transport Certificate.

DD= ID || PAN4-15

Ktj = DESECB[Kc](DD)

CT=MAC|RN2[Ktj](Block0 || Block1 || Block2 || Block3)

5.9. Cash Back and Transport Certificates Generation Advantis

5.9.1. Cash Back Certificte.

DD= ID || PAN4-15

Ktj = DESECB[Kc](DD)

Ktr = DESECB[Ktj](RN1)

CM=MAC|RN2[Ktr](Block0 || Block1 || Block2)

5.9.2. Transport Certificate.

DD= ID || PAN4-15

Ktj = DESECB[Kc](DD)

Pass 1:

CT=MAC|RN2[Ktj](Block0 || Block1)

Pass 2:

CT=MAC|RN2[Ktj](Block0 || Block1 || Block2 || Block3)

5.10. Load Printing Format

It is possible to distribute key components safely by printing them, for example, blind envelopes. To this end, you can connect a printer to the serial port RS-232 Cryptosec.

Before using the print command, it is necessary to define the paper or envelope format. A format is maintained until it is overwritten by another. Two commands are used to send the format symbols to the module. The symbols that define either format fields or any fixed string are provided in the table presented below.

Command 1301 must be used to create the document layout. The format strip can contain any fixed text, although it is suggested to restrict to formatting characters. This is because it is limited to 400 characters. The second command is integrated with the print function that makes use of the capabilities of printing. At that point you can include up to 16 text strings fixed to 252 characters each, involved with the printing according to the template set, together with the information supplied by the same command.

Printing format symbols are included in the following table:

Symbol	ASCII	Meaning
>L	3E 4C	Leap line, carriage return
>V	3E 56	Tab vertical
>H	3E 48	Tab Horizontal
>F	3E 46	Page break
>nnn	3E 3n 3n 3n	Jump to the column nnn from left margin, where nnn is a three digit decimal number
^M	5E 49	Print the third component of a key clear.
^P	5E 50	Print PIN clear for envelope 1 or prints a key component clear.
^Q	5E 51	Print PIN clear for envelope 2 or prints a key component clear.
^R	5E 52	Prints reference for envelope 1
^S	5E 53	Prints reference for envelope 2
^T	5E 54	Print the last 6 digits of the account number on the envelope 1, or print the KCV key component
^U	5E 55	Print the last 6 digits of the account number on the envelope 2
|<L> <hh hh hh ..>	7C<L> <hh hh hh..>	Send binary data to the printer, for example control sequences. L contains the number of bytes to send up to 255 bytes. Followed by the bytes to send.
^0	5E 30	Insert the field of printing 0 defined in the function which makes use of the print

Symbol	ASCII	Meaning
^1	5E 31	Insert the field of printing 1 defined in the function which makes use of the print
...	...	...
^F	5E 46	Insert the field of printing 15 defined in the function which makes use of the print

Note that, although all symbols are supported, in some cases firmware does not support their use. In each command is detailed what symbols are used to return your information.

5.11. Key Diversification

Algorithm identifier 12:

$\mathbf{Kr = Kr_{LEFT} \,||\, Kr_{RIGHT}}$

$\mathbf{Kd = Kd_{LEFT} \,||\, Kd_{RIGHT}}$

$\mathbf{Kr_{LEFT} = TDES_{ECB}\,(Kd)\,[D_1]}$

$\mathbf{Kr_{RIGHT} = TDES_{ECB}\,(Kd)\,[D_2]}$

Algoritm identifier 22:

$\mathbf{Kr = Kr_{LEFT} \,||\, Kr_{RIGHT}}$

$\mathbf{Kd = Kd_{LEFT} \,||\, Kd_{RIGHT}}$

$\mathbf{ER = TDES_{CBC(0000000000000000)}(Kd)\,[ICCID_{1\text{-}8} \,||\, ICCID_{9\text{-}10} \,||\, ICCID_{1\text{-}6} \,||\, ICCID_{7\text{-}10} \,||\, FFFFFFFF]}$

$\mathbf{Kr_{LEFT} = ER_{9\text{-}16}}$

$\mathbf{Kr_{RIGHT} = ER_{17\text{-}24}}$

5.12. ALT_1 PIN Offset Calculation Method

Instead of calculating PIN Offset as a difference between PIN cosen by the customer (C-PIN) and the calculated PIN from validation data (A-PIN), is calculated as the sum of these values, as follows:

O-PIN = C-PIN + A-PIN

Both the amount and the remainder are understood digit by digit, module 10 and without carry.

As for the calculation of A-PIN, this is described bellow:

Encrypt the validation data (16 hexadecimal digits) with the PIN generation Key.

The result of the decimation previous step, scanning the 16 hexadecimal digits from left to right, ignoring every digit larger than 0x9 until 4 decimal digits are found (digits values are from 0x0 to 0x9).

If all digits have been scanned but that 4 decimal digits have not been found, repeat the process, ignoring all digits from 0x0 to 0x9. Subtract (module 10 and without carry) 0xA each digit selected in this way.

PIN is formed concatenating the result of both revisions and retaining the first four digits