Guía para programar la lectura de nb octetos de un fichero de la tarjeta ACOS3 UTILIZANDO SECURE MESSAGING, es decir: la tarjeta enviará cifrados los datos al terminal.

****

Esto es lo que el manual de la tarjeta se denomina:
ISO-OUT (pág 37 del manual de la ACOS3)

****

Pista o truco:
Se puede escribir o leer con Secure Messaging en un fichero, incluso si no tiene puesto los correspondientes flags en su record de definición.
Los flags; si se ponen, obligan a usar Secure Messaging, pero si no se ponen no lo impiden.
Por tanto para probar será conveniente hacerlo en un fichero que no tenga ninguna restricción de acceso.
Escribiremos en él con o sin Secure Messaging y podremos leer de él (con Secure Messaging) para comprobar que la lectura ha funcionado.
Si esto fuese un desarrollo real, por supuesto que habría que poner los flags de Secure Messaging en su valor para que impidieran que se leyera de otra forma.
****

Se trata de leer de un fichero de la tarjeta un mensaje de nb bytes, de longitud menor o igual a 240 bytes.

Suponemos que la tarjeta está configurada para utilizar 3DES.

Hay que realizar la Autenticacion Mutua con éxito, calcular la clave de sesión KS. Poner la clave de sesión en un objeto para poder utilizarla.

A la tarjeta hay que enviarle una APDU de lectura como la siguiente:

CLA*	INS	P1	P2	P3*	97H	01H	P3	8EH	04H	MAC(4 octetos)
1	1	1	1	1	1	1	1	1	1	4 bytes

CLA* = 8C, el CLA de la apdu de lectura es 80 modificado para secure messaging es 80H ORX 0CH = 8CH.
INS es B0, es la instrucción para leer de un fichero binario. Si fuese leer en un fichero de records la instrucción sería B2.
P1, P2 son los bytes de la posición del fichero donde se va a leer, parte High y Low respectivamente es decir si vamos a leer desde la posición 0000H del fichero P1=00 y P2=00. Si fueramos a leer a partir de la posición 0129H P1=01H y P2=20H.
P3* Es igual a 9. Cuando se envíe con sendApdu no se escribe en la instrucción.

El string | 97H | 01H | P3 | es una TLV donde 97H significa P3 original en un ISO-OUT comando.

P3 Es el número de bytes a leer del fichero. Se pasa a un byteString de un octeto, para incorporarlo a la TLV, por ejemplo con P3 = ByteString.valueOf(nb, 1).

El string | 8EH | 04H | MAC| es otra TLV donde 8EH significa Cryptographic Checksum o MAC que es lo mismo.

Para calcular el MAC hay que efectuar las siguientes operaciones:

Se generan las siguientes TLV's concatenadas:

| 89H | 04H | CLA* | INS | P1 | P2 | 97H | 01H | P3|

La primera TLV (| 89H | 04H | CLA* | INS | P1 | P2 |) tiene de etiqueta 89H, que significa command header.

La segunda TLV (| 97H | 01H | P3 |) es la TLV de P3 original.

Ahora hay que cifrar estas dos TLV's concatenadas; con 3DES (rellenando al final con el metodo: pad(Crypto.ISO9797_METHOD_2, true)), y con el modo CBC, con la clave de sesión KS, y con el IV calculado según:
IV se calcula haciéndole un AND lógico bit a bit al RNDC (el aleatorio de la tarjeta) con el byteString: 00 00 00 00 00 00 FF FF. Al resultante se le incrementa en uno para obtener IV.

El resultado de este cifrado será un string. Nos fijaremos en el último octeto de ese string. Si a ese string lo llamamos "leecifrado":
El último octeto se obtiene con la intrucción:

```
octetoparacifra = leecifrado.right(8);
```

Supongamos que es: F9 54 05 97 0F 1B 0B 85

El MAC que buscamos son los 4 bytes más a la izquierda de estos 8 bytes. Sería: F9 54 05 97 se puede obtener con la intrucción: MAC = octetoparacifra.left(4);

Estos 4 bytes son los que como MAC hay que hay que enviarle a la tarjeta en la APDU de lectura con secure messaging.

Si el comando es correcto la respuesta será 61xxH. Solo indica que ha tenido éxito. En el manual dice que xxH es el número de octetos que hay que pedir en GET RESPONSE. Ni caso, lo haremos de otra manera.
Si devuelve 6882H ("SECURE MESSAGING NOT ALLOWED") no hacer ni caso porque funciona.

A continuación hay que "sacar los datos" del fichero con GET RESPONSE. El valor de P3 que hay que pasarle a GET RESPONSE es nbp+15. nb es el número de bytes que se quieren leer del fichero y nbp es el número de bytes cuando se lleva al valor del siguiente múltiplo de ocho. Hay que calcularlo. 15 es la suma del resto de octetos que enviará la tarjeta en la petición de respuesta como se ve a continuación.

Es decir a la tarjeta hay que enviarle una APDU GET RESPONSE así:

card.sendApdu(0x80, 0xC0, 0x00, 0x00, nbp+15);

La respuesta de la tarjeta será así:

87H	L87	Pi	Encripted(padding)	99H	02H	Sw1Sw2	8EH	04H	MAC	9000H
1	1	1	n*	1	1	2	1	1	4	2

Esta respuesta tiene n* + 15 bytes, donde:
n* es el número de bytes que antes hemos llamado nbp: el número de bytes que se ha pedido leer en la GET RESPONSE, llevado al múltiplo de ocho de valor inmediato superior.
15 es el número de octetos que suman el resto de los campos que no son datos cifrados.
Pi: octetos de relleno en el encripted data.
Sw1Sw2: Respuesta al comando original.
MAC: Checksum criptográfico calculado así:

La tarjeta genera las tres TLV's encadenadas:

89H	04H	CLA*	INS	P1	P2	87H	L87	Pi	Encripted(padding)	99H	02H	Sw1Sw2

Rellena al primer múltiplo de ocho por encima y lo cifra con CBC, con el IV+1, descrito antes.
Del resultado de este cifrado los cuatro bytes más a la izquierda del último octeto será el MAC.

El programador, del software del terminal, deberá verificar el MAC calculándolo por su cuenta y comparándolo con el que le envía la tarjeta. Si coinciden procederá a descifrar los datos con CBC, la clave de sesión (KS) y el vector de inicialización (IV+1) (el mismo usado antes para el cálculo del MAC).

Una vez descifrados los datos estos traen Pi octetos de relleno.
Hay que descontárselos por la derecha al mensaje descifrado y obtendremos el mensaje en claro y limpio.

****
*   *
*****Resumiendo y escribiendo la secuencia de pasos quedaría así: *
*   *
****

Paso 1) Realizar la AUTENTICACIÓN MUTUA con éxito para obtener una clave de sesión KS.
Paso 2) Calcular el MAC de la APDU de lectura de Secure Messaging.
Paso 3) Formar las TLV's para la APDU de lectura y enviarle la APDU de lectura a la tarjeta.
Paso 4) Calcular el valor múltiplo de ocho por encima del número de datos a leer, sumarle 15.
Paso 5) Enviar a la tarjeta la APDU GET RESPONSE con P3 = nbp+15.

```
card.sendApdu(0x80, 0xC0, 0x00, 0x00, nbp+15);
```

Paso 6) La tarjeta envía un string, terminado con un MAC. Verificar el MAC.
Paso 7) Si el MAC es correcto descifrar Encripted data (con el padding).
Paso 8) Eliminar Pi octetos de relleno a la derecha de los datos descifrados.
Paso 9) Sacar el mensaje a la pantalla para leerlo en claro.

**********SUERTE**********