

Guía de adaptación para el tratamiento en terminales de validación de tarjeta de transporte basada en virtualización HCE para entornos Android

Versión 2.5.3

04/07/2023

Archivo	Guía de adaptación para el tratamiento en terminales de validación de tarjeta virtual basado en Android HCE
Título	Guía de adaptación para el tratamiento en terminales de validación de tarjeta de transporte basada en virtualización HCE para entornos Android
Versión	[2.5.3]
Autor	María del Amor León Fariña y Luis Criado Fernández

Control de Versiones

Versión	Descripción	Fecha
1.0	Creación del documento	11/07/2016
2.0	Se ha incluido el algoritmo para adaptarse a SAM tipo 2 y SAM tipo 4	16/05/2018
2.1	FCI	03/07/2018
2.2	Optimización algoritmo epígrafe 5	04/07/2018
2.3	Fijando el parámetro P2 al valor 0x0C en el algoritmo del epígrafe 5	10/07/2018
2.4	Aclaración formato Timestamp	22/08/2018
2.5	Se ha corregido el FCI	13/09/2018
[2.5.1]	Se ha corregido pequeña errata en el diagrama de flujo de "lógica BIT" (bifurcación Timestamp)	31/10/2018
[2.5.2]	Se ha indicado que por defecto se utilizará SetKey5 (situación definitiva) y en caso de fallo, el SetKey1	13/09/2021
[2.5.3]	En el algoritmo se indica SAM tipo 4 o superior	04/07/2023

Copyright © CRTM. Todos los derechos reservados. Solo para uso interno.
Prohibida su distribución sin autorización expresa

1. Antecedentes

El CRTM cuenta con la tecnología necesaria para virtualizar una tarjeta TTP o Multi, en un entorno basado en Android HCE utilizando el canal NFC del móvil, para interactuar con los terminales de validación inspección o incluso de carga /recarga distribuidos en la red de transporte y de ventas del CRTM. Sin embargo, para estos equipos y en particular los de validación, esta virtualización, queda oculta ya que no están adaptados. Afortunadamente esta adaptación es sencilla y se reduce a modificar algunas llamadas a bajo nivel.

Hasta la fecha se habían realizado diferentes pilotos en NFC basados en la SIM de telefonía puesto que esta tecnología no necesitaba adaptar los protocolos ya establecidos a bajo nivel en los validadores. Sin embargo, la virtualización de tarjetas en SIM presenta otros problemas que no detallaremos en este documento puesto que exceden su ámbito.

HCE son las siglas de **Host Card Emulation** y corresponden con una técnica que se utiliza para virtualizar tarjetas inteligentes sin contacto en terminales móviles por software seguro. Es válido para entornos Blackberry 10 o Android 4.4 o superior.

2. Despertar HCE

La virtualización en un entorno HCE a alto nivel consiste en implementar un servicio seguro (equivalente a nivel software a una tarjeta física) y darlo de alta en el sistema operativo del móvil.

El problema viene con las tarjetas Desfire y los comandos nativos de estas. El modelo HCE desarrollado en Android sigue por completo la norma ISO-7816-4, esto incide en cómo el dispositivo maneja la emulación de una determinada tarjeta. Los distintos emuladores que pueden convivir en un dispositivo se asocian a un determinado AID. Cuando Android detecta un Select de ese AID, cede el control al emulador en cuestión. La dificultad es que las Desfire no usan APDU's 7816-4 y Android no sabe interpretar los comandos nativos Desfire, en esta situación el emulador Desfire nunca recibiría el control. Para solventar esta situación es necesario incluir por delante de los comandos nativos Desfire un Select

AID, cuya única función, es indicar a Android que ceda el control al emulador Desfire. Lo que hemos denominado “despertar HCE”.

A continuación, se describe cómo se realiza este proceso:

Se denomina AID (Application ID) al identificador de la aplicación, que a diferencia del mundo de los comandos nativos de la familia de las tarjetas NXP DESFIRE este identificador no puede ser de 3 bytes, la longitud mínima permitida es de 5 Byte llegando a 16 Byte como máximo.

El CRTM ha seleccionado como AID para las virtualizaciones en HCE uno de 7 bytes que corresponde con AID= DE5C0D1F1CADA5 expresado en hexadecimal y que, si leemos los 5 como “s”, los 0 como “o” y los 1 como “i” obtenemos la palabra **DESCODIFICADAS** como identificador de la aplicación.

El control del canal de comunicaciones NFC pertenece al sistema operativo del móvil y es necesario indicarle al sistema operativo del móvil que pase el control del canal al emulador DESFIRE o a la tarjeta HCE. Para ello se necesita comunicarlo por el APDU correspondiente en formato de trama ISO 7816 que es la que entiende el entorno de HCE. El comando para seleccionar un AID se codifica con la instrucción A4 y el mensaje en APDU quedaría:

- CLA -> 00
- INS -> A4
- P1 -> 04 (Indica selección por el nombre indicado en el campo de datos, es decir por el AID).
- P2 -> 00
- lc -> 07 (Longitud del AID)
- Data field -> **DE5C0D1F1CADA5**
- Le -> 00

La respuesta de una tarjeta física a este comando será errónea, mientras que la virtualización devolverá un FCI

3. Estructura FCI

The structure of the FCI which will be retrieved as a response to the command ISO Select is listed in Table 5. A detailed explanation of all fields which are contained in the FCI can also be found below.

Some parts of the FCI will always be different for physical MIFARE DESFire EV1/EV2 ICs whereas a MIFARE 2GO implementation. The reason therefore is the ability to distinguish between which device type is currently presented to the reader terminal, and for being able to follow-up during the transaction accordingly.

	TLV Header					Payload = Info PDCap1 VCUID Zero Padding			
						Timestamp			
Longitud en bytes	1	1	1	1	1	2	7	1	5
Tipo	tag	length	tag	length	info	PDCap1 (PDCap1.1 PDCap1.2)	VCUID	MIFARE2GO Version Identifier	Timestamp
Valor	0x6F	0x12	0x85 or 0x86	0x10	0x14	0x020C	dynamic	dynamic	dynamic

Explicación payload of the FCI:

- VCUID (Virtual Card UID)

- 7 bytes length

- PDCap1

- PDCap1.1 values

- 0x00 - RFU, MIFARE DESFire default value

- 0x01 - MIFARE DESFire standard Timestamp (4 bytes, where leftmost byte is set to 0x00 in order to receive 5 bytes in total for the Timestamp)

- 0x02 - MIFARE DESFire extended Timestamp (5 bytes to prevent the millenium bug)

- PDCap1.2 values

- 0x0A - MIFARE DESFire 2kB default value

- 0x0B - MIFARE DESFire 4kB default value

- 0x0C - MIFARE DESFire 8kB default value

- MIFARE 2GO Version Identifier

- MIFARE DESFire physical IC = 0x00

- MIFARE 2GO = increasing version with each new software release, starting from 0x00

- Timestamp: Utiliza la marca de tiempo de UNIX, es decir, es un número que representa los segundos transcurridos desde el 1 de enero de 1970 (UTC)

4. Uso de comandos para garantizar la compatibilidad entre tarjetas físicas y virtuales

El validador está continuamente radiando y explorando su campo para detectar una tarjeta, en el nuevo escenario, no solo puede aparecer una tarjeta Desfire, también puede aparecer una emulación HCE, es decir tarjetas virtuales.

El objetivo es enviar siempre el APDU comentado anteriormente en este documento, aunque tiene un inconveniente que puede resolverse.

Si se envía un APDU 7816 a una Desfire, esta dejará de aceptar comandos nativos, por lo tanto, deben ser encapsulados en el APDU definido por NXP a tal efecto, esto es:

- CLA -> 90.
- INS -> comando nativo de la Desfire
- P1 -> 00.
- P2 -> 00.
- Ic -> Longitud de los argumentos.
- Data field -> Argumentos.
- le -> Longitud de los datos esperado en la respuesta. Se pasa un 00 indicando que se espera recibir cualquier longitud.

Como ejemplo de encapsulación de comandos nativos, se han elegido estos tres:

- Comando de obtención de información:
 - Forma nativa: 60
 - Forma nativa encapsulada: 90 60 00 00 00.
- Autenticación nativa:
 - Forma nativa: 0A xx (xx indica el índice de la clave)
 - Forma nativa encapsulada: 90 0A 00 00 01 xx 00.
- Selección de aplicación
 - Forma nativa: 5A xx xx xx (xx xx xx indica el AID de la aplicación Desfire)
 - Forma nativa encapsulada: 90 5A 00 00 03 xx xx xx 00.

Conclusión: Para garantizar el funcionamiento dual entre tarjetas físicas y virtuales se deben encapsular los comandos en el APDU definido por NXP

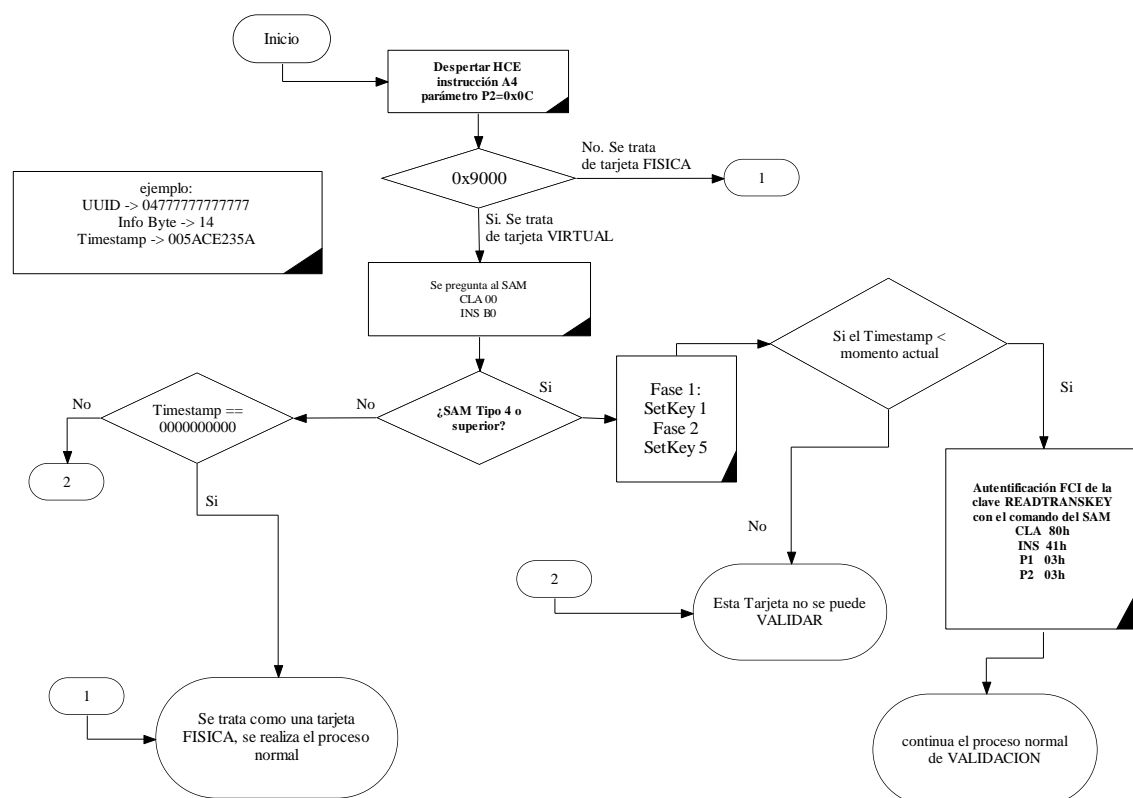
5. Lógica BIT

La estructura de la tarjeta virtualizada es idéntica 100x100 a la tarjeta física.

Los algoritmos utilizados en los distintos aplicativos (validación, inspección, carga/recarga) son idénticos a los definidos para las tarjetas físicas.

Las transacciones generadas son iguales en formato y contenido a las que se realizan con tarjetas físicas.

El número de serie de la tarjeta virtual se obtiene en el FCI



6. Conclusiones

A modo de resumen se hace hincapié en los siguientes puntos.

La lógica BIT se mantiene completamente intacta, tanto si se opera con una tarjeta Desfire física, como si se opera con una tarjeta Desfire emulada en HCE.

Es obligatorio el uso de comandos encapsulados según estándar 7816 y no en forma nativa del fabricante de tarjetas, puesto que el mundo HCE no ofrece garantías de continuación de utilización de comandos que no se ajusten a estándares.

En cualquier caso y debido a que, en las especificaciones de Android para el HCE, se especifica que el UUID intercambiado será aleatorio, es necesario obtener el número de serie de la tarjeta virtual a partir del FCI.