

DESCRIPCIÓN

Título de la invención: ejecución mejorada de una operación en un elemento seguro**Campo técnico**

La presente invención se refiere, de manera general, al campo de los elementos seguros y, más particularmente, a la gestión de la ejecución de tareas u operaciones informáticas, tales como tratamientos criptográficos, en elementos seguros.

Técnica anterior

El ETSI (por "European Telecommunications Standards Institute") trabaja actualmente en las especificaciones técnicas de las redes de 5ª generación (5G).

5G prevé concretamente que cada abonado a un servicio dado de una red de comunicación, por ejemplo radio o WiFi (marca comercial), tenga un identificador 5G único en el mundo, indicado SUPI (por "SUbscriber Permanent Identifier"), incorporado en un elemento seguro, de tipo tarjeta SIM (por "Subscriber Identity Module").

Este identificador de abonado incluye, por ejemplo, la IMSI (por "International Mobile Subscriber Identity") que es un identificador único asignado a cada abonado móvil de una red móvil GSM (2G), UMTS (3G) o LTE (4G), o el NAI (por "Network Access Identifier") introducido por RFC 4282.

En un intento por mejorar la seguridad, la especificación técnica ETSI TS 133 501 V15.1.0 (2018-07) prevé la autenticación del abonado, con la ayuda de este identificador único, en múltiples ocasiones, concretamente a lo largo del proceso de autenticación y de intercambio de claves entre un equipo usuario anfitrión (y más particularmente el elemento seguro insertado o soldado en el mismo) y la red, es decir en cada conexión a la red.

Más particularmente, la especificación recomienda un cifrado de este identificador en curvas elípticas, antes de enviarse a la red.

Es posible prever, por ejemplo a lo largo de la personalización del elemento seguro del equipo usuario anfitrión, un juego de identificadores previamente cifrados que van a usarse para estas autenticaciones y conectarse de ese modo a la red. No obstante, la memoria del elemento seguro es limitada y no es posible prever un número de identificadores ya cifrados suficiente para todas las conexiones que tendrán lugar.

Por tanto, el elemento seguro debe cifrar necesariamente de manera regular el identificador "en el momento". Ahora bien, una operación de cifrado de este tipo, realizada mediante software, resulta costosa en cuanto a recursos de cálculo, tanto más cuando se realiza de manera frecuente. Además, acapara los recursos durante un tiempo prolongado.

La operación de cifrado es tan solo un ejemplo de operaciones costosas en cuanto a recursos de cálculo. Las operaciones periódicas de limpieza de la memoria, conocidas con la denominación "garbage collector" (o recolector de basura), también consumen muchos recursos de tratamiento.

De manera general, un elemento seguro se enfrenta a numerosas operaciones que acaparan los recursos de tratamiento durante duraciones prolongadas.

Esta inmovilización de los recursos del elemento seguro reduce en gran medida la reactividad del elemento seguro frente a solicitudes del equipo usuario anfitrión o de la red de telefonía móvil. Por tanto, esta situación no está adaptada a 5G que tiene como objetivo abordar las necesidades aumentadas de aplicaciones complejas que necesitan una alta reactividad.

Por consiguiente, existe una necesidad de mejorar la gestión de este tipo de operaciones en el interior de los elementos seguros.

Descripción de la invención

Por tanto, la presente invención tiene como objetivo aliviar al menos uno de los inconvenientes anteriormente mencionados.

En este contexto, un primer aspecto de la invención se refiere a un procedimiento de gestión de la ejecución de una operación

principal por un elemento seguro, estando el procedimiento caracterizado porque:

la operación principal está formada por una pluralidad de operaciones elementales, y

el procedimiento comprende las siguientes etapas:

- a. recibir, de un equipo anfitrión, un comando en el formato de unidad de datos de protocolo de aplicación, APDU,
- b. activar un contador de tiempo durante la ejecución del comando de APDU con el fin de determinar una duración real de tratamiento del comando de APDU por el elemento seguro;
- c. determinar si la duración real de tratamiento indicada por el contador de tiempo es inferior a una duración predefinida asociada al comando de APDU; y
- d. en caso de determinación positiva, ejecutar una o varias de dichas operaciones elementales durante el tiempo restante de dicha duración predefinida.

De manera correlativa, un segundo aspecto de la invención se refiere a un elemento seguro configurado para ejecutar una operación principal formada por una pluralidad de operaciones elementales, comprendiendo el elemento seguro un procesador configurado para:

- a. recibir, de un equipo anfitrión, un comando en el formato de unidad de datos de protocolo de aplicación, APDU,
- b. activar un contador de tiempo durante la ejecución del comando de APDU con el fin de determinar una duración real de tratamiento del comando de APDU por el elemento seguro;
- c. determinar si la duración real de tratamiento indicada por el contador de tiempo es inferior a una duración predefinida asociada al comando de APDU; y
- d. en caso de determinación positiva, ejecutar una o varias de dichas operaciones elementales durante el tiempo restante de dicha duración predefinida.

La invención así reivindicada permite mejorar la gestión de las operaciones puestas en práctica en el interior de los elementos seguros.

En efecto, la operación principal, por ejemplo que consume muchos recursos de tratamiento, se realiza por partes durante el tiempo restante tras la ejecución del comando de APDU recibido.

Además, la operación principal puede realizarse en varias veces a lo largo de una pluralidad de iteraciones de ejecución de comandos de APDU sucesivos, sin por ello haber impedido la puesta en práctica de otros tratamientos (comandos de APDU) antes del final de la operación principal.

La invención recupera de ese modo un alto grado de reactividad del elemento seguro.

Todavía según la invención, el procedimiento comprende además una etapa e. de enviar una respuesta al comando de APDU con destino al equipo anfitrión, tras la etapa d.

En una realización particular, la respuesta al comando se envía al caducar la duración predefinida asociada al comando de APDU. Controlando el instante de emisión de la respuesta de APDU hacia el equipo anfitrión, el elemento seguro prolonga artificialmente el tiempo asignado al tratamiento del comando de APDU con el fin de permitir la ejecución de operaciones elementales al final de este tratamiento. Este tiempo asignado corresponde a la duración predefinida asociada al comando de APDU.

Esta prolongación controlada presenta además ventajas de seguridad. En efecto, permite modificar (y por tanto ocultar) el tiempo real de tratamiento o de ejecución de un comando de APDU, visto desde el exterior del elemento seguro, ya que este tiempo se define habitualmente entre el envío del comando y la recepción de su respuesta. Por tanto, no se revelan las características de tratamiento del elemento seguro; una persona malintencionada ya no puede identificar el tratamiento realmente puesto en práctica.

En las reivindicaciones dependientes se describen otras características del procedimiento y del elemento seguro según realizaciones de la invención.

En otra realización, el contador de tiempo se activa tras la recepción del comando de APDU. Esto permite seguir el tiempo de ejecución del comando de APDU tal como se observa desde el exterior del elemento seguro, con el fin, por ejemplo, de adaptarse más fácilmente a duraciones máximas normalizadas. Evidentemente, como variante, podrá activarse al mismo tiempo que la ejecución del comando de APDU por el elemento seguro, tras su recepción.

En otra realización, el procedimiento comprende una pluralidad de iteraciones de las etapas a. a d. de manera que se ejecuta dicha pluralidad de operaciones elementales que forman la operación principal. Todas las operaciones elementales pueden ejecutarse durante los tiempos restantes tras la ejecución de varios comandos de APDU sucesivos.

No obstante, como variante, algunas de las operaciones elementales pueden ejecutarse por el elemento seguro entre la gestión de dos comandos de APDU sucesivos. Concretamente, el procedimiento puede comprender, tras el envío de una respuesta a un comando de APDU recibido y previamente a la recepción de un siguiente comando de APDU, la ejecución adicional de una o varias de dichas operaciones elementales. Además, en cuanto el elemento seguro ya no está solicitado por comandos de APDU, puede hacer avanzar la ejecución de las operaciones elementales que componen las operaciones principales que deben realizarse.

En una realización particular, la ejecución adicional de una o varias de dichas operaciones elementales comprende la ejecución sucesiva de una o varias operaciones elementales hasta la recepción del siguiente comando de APDU. Dicho de otro modo, el elemento seguro ejecuta operaciones elementales mientras no se recibe ningún nuevo comando de APDU, que debe tratar. En este caso, detiene el tratamiento de las operaciones elementales y

trata el nuevo comando de APDU recibido según las etapas a. a d. anteriores.

En una realización, la operación principal se divide previamente para dar dicha pluralidad de operaciones elementales, estando cada una asociada a una duración de tratamiento elemental, y la etapa d. comprende la selección de la una o varias de dichas operaciones elementales que van a ejecutarse en función de sus duraciones de tratamiento elemental asociadas y de dicho tiempo restante.

En este modo, las operaciones elementales se conocen por adelantado y, por tanto, están predefinidas, con sus costes (duraciones de tratamiento teóricas) asociados. Cuando resulta estar disponible tiempo restante tras la ejecución de un comando de APDU, el elemento seguro puede entonces buscar, preferiblemente en un orden lógico, operaciones elementales que puede ejecutar durante este tiempo restante, teniendo en cuenta el coste de tiempo de cada operación elemental.

Esto hace posible la ejecución "simultánea" de varias operaciones principales. En este caso, la etapa d. puede comprender la ejecución de al menos dos operaciones elementales procedentes de al menos dos operaciones principales distintas. En efecto, el elemento seguro puede buscar operaciones elementales de varias operaciones principales con el fin de optimizar el uso del tiempo restante tras la ejecución del comando de APDU.

En una realización distinta de la de la división de la operación principal (que consume muchos recursos), la etapa d. comprende la continuación de una ejecución de la operación principal desde un punto de suspensión anterior (inicialmente el comienzo de la operación), suspendiéndose dicha ejecución de la operación principal en un siguiente punto de suspensión que depende del tiempo restante.

En este caso, no se divide la operación principal en principio sino que se ejecuta de manera progresiva continuando, mediante

suspensiones sucesivas, su ejecución. Por tanto, esta ejecución se reanuda tras la ejecución del comando de APDU, durante el tiempo restante, después se suspende de nuevo al final del tiempo restante, preferiblemente en un punto de suspensión que no compromete la ejecución global de la operación principal.

Además, el punto de suspensión denominado "anterior" durante una nueva iteración de las etapas a. a d. puede ser el punto de suspensión denominado "siguiente" resultante de la última iteración de las etapas a. a d. Evidentemente, si el elemento seguro llega a ejecutar una parte de la operación principal entre el tratamiento de dos comandos de APDU, el punto de suspensión "anterior" durante una nueva iteración de las etapas a. a d. será el punto en el que se haya detenido esta ejecución durante recepción del nuevo comando de APDU que va a tratarse.

Las ventajas, objetivos y características particulares del elemento seguro son similares a los del procedimiento anteriormente mencionado. Un sistema puede comprender el elemento seguro y un equipo anfitrión de dicho elemento seguro. En una realización particular, las diferentes etapas del procedimiento anteriormente mencionado están determinadas por instrucciones de programas informáticos.

En consecuencia, la invención también se refiere a un soporte no transitorio legible por ordenador que almacena un programa informático que, cuando se ejecuta por un elemento seguro dotado de un procesador, hace que el elemento seguro realice el procedimiento tal como se definió anteriormente.

Este programa puede usar cualquier lenguaje de programación y estar en forma de código fuente, código objeto o de código intermedio entre código fuente y código objeto, tal como en una forma parcialmente compilada, o en cualquier otra forma deseable.

El soporte de información puede ser cualquier entidad o dispositivo que pueda almacenar el programa. Por ejemplo, el soporte puede comprender un medio de almacenamiento, tal como

una ROM, por ejemplo una ROM de microcircuito, o incluso un medio de grabación magnético, por ejemplo un disco duro, o incluso una memoria flash.

Alternativamente, el soporte de información puede ser un circuito integrado en el que está incorporado el programa, estando el circuito adaptado para ejecutar o para usarse en la ejecución del procedimiento en cuestión.

Breve descripción de los dibujos

Otras particularidades y ventajas de la invención se desprenderán adicionalmente de la siguiente descripción, ilustrada por las figuras adjuntas que ilustran ejemplos de realización carentes de cualquier carácter limitativo.

[Figura 1] La figura 1 representa, esquemáticamente, un elemento seguro en el que puede ponerse en práctica la presente invención.

[Figura 2] La figura 2 ilustra, de manera esquemática, módulos funcionales del elemento seguro para una puesta en práctica de la invención.

[Figura 3] La figura 3 ilustra, con la ayuda de un diagrama de flujo, etapas generales de un procedimiento según la invención.

[Figura 4] La figura 4 ilustra, con la ayuda de un diagrama de flujo, etapas generales de un procedimiento alternativo según la invención.

[Figura 5] La figura 5 ilustra una secuencia cronológica del tratamiento de varios comandos de APDU sucesivos mientras que tienen que tratarse operaciones principales, según puestas en práctica de la invención.

Descripción detallada

Un elemento seguro, SE (por "secure element" en inglés), es un componente o plataforma de hardware inviolable (normalmente un chip o una tarjeta de chip) usado en un dispositivo o terminal anfitrión (normalmente un terminal móvil) y que puede albergar, de manera segura bajo el control de un sistema operativo de la tarjeta, aplicaciones y datos de acuerdo con reglas y exigencias de seguridad fijadas por autoridades de confianza. Los elementos

seguros se usan concretamente para tratamientos criptográficos con fines de identificación, autenticación, firma, etc.

Un factor de forma ampliamente usado del SE es el elemento seguro incorporado o integrado, eSE (por "embedded Secure Element"). Este elemento seguro incorporado está generalmente soldado al terminal anfitrión.

Los elementos seguros están programados según las aplicaciones deseadas.

A modo de ejemplo, un eSE puede formar el elemento seguro necesario para numerosos usos o servicios que se basan en una comunicación NFC (por "Near Field Communication" en inglés) puesta en práctica por un terminal móvil anfitrión. Por ejemplo, un servicio de pago de NFC necesita informaciones bancarias secretas del usuario que están ventajosamente almacenadas en el eSE, protegidas de cualquier acceso inoportuno. También es el caso de un servicio de transporte público en el que el eSE permite identificar al usuario ante detectores.

Otro ejemplo de elemento seguro es la UICC incorporada (por "Universal Integrated Circuit Card") que proporciona las referencias de un abonado para autenticarse en una red de telefonía móvil. Se trata, por ejemplo, de un eSE configurado como una tarjeta SIM (por "Subscriber Identity Module", o módulo de identidad de abonado). Se habla entonces de eUICC (por "embedded Universal Integrated Circuit Card").

La figura 1 representa, esquemáticamente, un elemento seguro 100 en el que puede ponerse en práctica la presente invención. Este elemento seguro 100 comprende un microprocesador 101, al que están asociadas, por un lado, una o varias memorias de acceso aleatorio 102 (por ejemplo de tipo RAM, por "Random Access Memory"), por ejemplo por medio de un bus 103, y, por otro lado, una o varias memorias no volátiles 104 (por ejemplo de tipo ROM (por "Read Only Memory") y/o EEPROM (por "Electrically Erasable Programmable Read Only Memory") y/o flash), por ejemplo a través de un bus 105. La memoria de almacenamiento no volátil 104

almacena datos y software destinados a poner en práctica la invención, pero también cualquier dato necesario para la ejecución de los programas. La memoria volátil de trabajo 102 almacena el código ejecutable de los programas informáticos así como los registros adecuados para registrar variables y parámetros necesarios para su ejecución.

Los buses 103 y 105 pueden ser un único bus. El microprocesador 101 puede ser un criptoprocador seguro.

El elemento seguro 100 comprende además, conectados al/a los bus(es), una o varias interfaces de comunicación 106 con el terminal anfitrión (no representado) en el que está incorporado el elemento seguro.

La interfaz de comunicación 106 está realizada, por ejemplo, por medio de contactos previstos en el elemento seguro (normalmente una tarjeta de chip) según la norma ISO 7816 y/o por medio de una antena de radiofrecuencia (sin contacto) según la norma ISO 14443.

El elemento seguro 100 se comunica, a través del interfaz de comunicación 106, con un lector de tarjeta (según la norma ISO 7816 y/o 14443) con el que está equipado el terminal anfitrión, normalmente en un esquema de maestro-esclavo en el que el lector es el maestro. Además, el lector del terminal anfitrión envía un comando X, normalmente en el formato APDU (por "Application Protocol Data Unit"), y recibe de vuelta una respuesta Y, normalmente en el formato APDU, que es el resultado de la ejecución del comando X. Un comando de este tipo se denomina a continuación "comando de APDU".

Un comando de APDU también se conoce con la denominación de instrucción de APDU debido a la presencia de un código de instrucción "INS" en el comando.

En la práctica, el lector solo puede enviar un nuevo comando de APDU tras haber recibido una respuesta al comando de APDU anterior.

El elemento seguro 100 puede ser un eSE ("embedded SE"), una eUICC ("embedded Universal Integrated Circuit Card"), una eSIM ("embedded SIM"), destinado a integrarse, por ejemplo mediante soldadura, en un terminal anfitrión tal como un teléfono inteligente (smartphone), una tableta digital, un ordenador portátil, un asistente personal, un dispositivo de entretenimiento (por ejemplo, consola de videojuegos).

La figura 2 ilustra, de manera esquemática, módulos funcionales del elemento seguro 100 para una puesta en práctica de la invención. Evidentemente, pueden existir funciones adicionales no representadas, tal como se desprenderá de la siguiente descripción.

El elemento seguro 100 comprende un módulo de tratamiento o de ejecución 110, un estimador de duración asignada por un comando de APDU 120, un contador de tiempo 130, una pila o memoria intermedia de tareas/operaciones principales 140 y, opcionalmente, un gestor de tareas secundarias u "operaciones elementales" 150. Este gestor 150 comprende un módulo de división y de estimación de tiempo de tratamiento 152, una pila o memoria intermedia de operaciones elementales 154 y un selector de operaciones elementales 156. Una pila o memoria intermedia es una zona de memoria que almacena datos temporalmente. Puede ser de tipo FIFO (por "first in, first out").

El módulo de tratamiento o de ejecución 110 controla un procedimiento de gestión de la ejecución de una operación principal almacenada en la pila 140 según la invención. Normalmente, puede poner en práctica las operaciones de la figura 3 descrita a continuación.

El módulo de tratamiento 110 está conectado a la interfaz de comunicación 106 con el fin de recibir comandos de APDU X del terminal anfitrión, ejecutar estos comandos y proporcionar de vuelta respuestas APDU Y.

El módulo de tratamiento 110 puede solicitar al estimador de duración 120 con el fin de obtener una duración asignada para un comando de APDU X recibido.

El estimador de duración 120 puede comprender de manera sencilla una tabla de correspondencia que asocia una duración asignada $T_{\text{máx}}$ a un comando de APDU o un tipo de comando de APDU. A modo de ejemplo, puede preverse una misma duración asignada $T_{\text{máx}}$ para el conjunto de los comandos de APDU. Como variante, puede definirse una duración asignada $T_{\text{máx}}$ diferente para cada tipo de comando de APDU, normalmente mediante código de instrucción "INS" (se definen 18 códigos diferentes en la parte 3 de la norma ISO/IEC 7816, por ejemplo "read binary", "read record", "write binary", "open channel", "select file", "external authenticate", etc.). Evidentemente, pueden preverse otras definiciones de "tipo de comando de APDU", tales como, por ejemplo, las cuatro estructuras de comandos de APDU diferentes (formato corto, formato orientado a escritura, formato orientado a lectura, formato completo).

La duración asignada $T_{\text{máx}}$ puede definirse con respecto a un tiempo de respuesta máximo definido en la norma ISO/IEC 7816 (véase, por ejemplo, la sección 10.5.3.2 de la norma ISO/IEC 7816-3). $T_{\text{máx}}$ puede ser igual a ese tiempo de respuesta máximo o a un porcentaje del mismo, por ejemplo el 70% o el 80%. El porcentaje puede variar de un tipo de comando de APDU a otro, incluso de un comando de APDU a otro.

Como variante, la duración asignada $T_{\text{máx}}$ puede calcularse a partir de un tiempo medio de ejecución de un comando de APDU, de un tipo de comando de APDU o del conjunto de los comandos de APDU. $T_{\text{máx}}$ puede ser igual a ese tiempo de respuesta medio aumentado en un porcentaje, por ejemplo del 10%, el 15%, el 20%, el 25% incluso el 50%. El porcentaje puede variar de un tipo de comando de APDU a otro, incluso de un comando de APDU a otro. El tiempo medio puede determinarse de manera previa o a lo largo de la vida útil del elemento seguro.

El estimador de duración 120 también puede comprender un submódulo de aprendizaje que tiene como objetivo ajustar o actualizar la duración asignada $T_{\text{máx}}$ en el tiempo, concretamente en función de las duraciones reales de ejecución de los comandos de APDU que ejecuta el elemento seguro 100. El aprendizaje se realiza con la misma granularidad que la asignación de las duraciones asignadas a los comandos de APDU, a saber por comando de APDU, por tipo de comando de APDU o de manera global para el conjunto de los comandos de APDU.

Una de las duraciones $T_{\text{máx}}$ definidas anteriormente puede usarse como duración inicial.

Una media de las duraciones reales de ejecución (que incluye eventualmente la duración inicial) aumentada en un porcentaje, por ejemplo del 10%, el 15%, el 20%, el 25% incluso el 50%, puede definir la duración asignada $T_{\text{máx}}$ para el siguiente comando de APDU (o el tipo de comando de APDU, o el conjunto de los comandos de APDU). Como variante al uso de la media, puede usarse la duración real de ejecución máxima (según la granularidad conservada).

La actualización de las duraciones asignadas $T_{\text{máx}}$ puede realizarse en cada comando de APDU X tratado, o después de un número predefinido de comandos de APDU tratados (eventualmente idénticos o del mismo tipo) o de manera periódica.

Este aprendizaje permite "adaptar" el estimador 120 a las características del elemento seguro 100.

Como variante al aprendizaje autónomo, las duraciones asignadas $T_{\text{máx}}$ pueden actualizarse (periódicamente o bajo demanda) mediante transferencia a distancia ("over-the-air" u OTA) concretamente cuando el terminal anfitrión es un terminal de comunicación móvil, o mediante fichero cargado directamente por el terminal anfitrión en el elemento seguro 100.

El módulo de tratamiento 110 puede solicitar el contador de tiempo 130 con el fin de medir la duración real de ejecución o de tratamiento de un comando de APDU X recibido.

El contador de tiempo 130 está normalmente conectado a la línea de reloj CLK del elemento seguro, que generalmente se proporciona por el terminal anfitrión.

El contador de tiempo 130 puede activarse tras la recepción de un nuevo comando de APDU X. Como variante, puede activarse únicamente cuando el módulo de tratamiento 110 comienza a leer el comando (por tanto, el encabezado "CLA", "INS", "P1" y "P2" seguido por el cuerpo "Lc", "DATA", "Le").

El contador de tiempo 130 se detiene cuando se obtiene el resultado de ejecución por el módulo de tratamiento 110 o la respuesta de APDU Y está lista para transmitirse. En los dos casos, la respuesta de APDU Y aún no se ha transmitido al terminal anfitrión.

Tal como se describe a continuación, la duración medida por el contador de tiempo 130 permite al módulo de tratamiento 110 saber fácilmente si queda tiempo de tratamiento disponible en vista de la duración $T_{\text{máx}}$ asignada al comando de APDU tratado. Este tiempo disponible permite hacer avanzar parcialmente la ejecución de una operación principal almacenada en la memoria intermedia 140. Tal como se describió anteriormente, esta duración medida también permite, en determinadas realizaciones, actualizar la duración $T_{\text{máx}}$ asignada a este comando de APDU (o a un tipo de comando particular, incluso al conjunto de los comandos de APDU). La memoria intermedia 140 es una zona de memoria en la que el elemento seguro 100 almacena operaciones que desea realizar cuando hay recursos de tratamiento disponibles. Normalmente, una rutina del elemento seguro 100 almacena en la misma las operaciones que consumen muchos recursos.

A modo de ejemplo, la generación de identificadores cifrados, por ejemplo SUPI o IMSI, monopoliza el microprocesador 101 durante una duración prolongada. Una combinación de identificadores cifrados (según técnicas clásicas) puede estar presente en el elemento seguro 100 para facilitar las operaciones de autenticación ante una red de comunicación móvil. Cualquier

identificador cifrado usado se suprime de la combinación. Entonces, una rutina puede garantizar que la combinación contiene de manera permanente al menos N identificadores cifrados. Además, cuando este número disminuye hasta N , la rutina puede lanzar una operación de generación de M nuevos identificadores cifrados. Esta operación se graba en la memoria intermedia 140. De manera similar, operaciones de cifrado, de descifrado, de firma, ..., pueden estar compuestas por múltiples operaciones elementales.

En otro ejemplo, las operaciones, por ejemplo periódicas, de recogida de basura (en inglés "garbage collector" o GC) responsable del reciclado de la memoria previamente asignada y después inutilizada pueden indicarse en la memoria intermedia 140.

Estas operaciones denominadas "principales" están generalmente compuestas por una pluralidad de tareas u operaciones elementales. Por ejemplo, el cifrado de un identificador es un cifrado por fragmentos o bloques: el cifrado de cada bloque que compone el identificador puede asimilarse a una operación elemental. Asimismo, la recogida de basura puede tratar la memoria por bloques o zonas: la limpieza de cada zona que compone la memoria que va a limpiarse puede asimilarse a una operación elemental. Más generalmente, una operación principal es una serie de operaciones o etapas que pueden realizarse de manera progresiva.

Un entorno de ejecución preventivo (en el elemento seguro) puede, en particular, suspender la ejecución de una operación y reanudarla posteriormente. Esta suspensión puede realizarse, por ejemplo, en la frontera de cada operación elemental o etapa que constituye la operación principal. Más generalmente, pueden identificarse puntos de suspensión en estas operaciones principales, donde la ejecución puede interrumpirse y suspenderse sin dañar la ejecución en su globalidad (concretamente sin impacto sobre el resultado final de la

ejecución). La suspensión puede requerir opcionalmente la grabación de un entorno de ejecución (registros, parámetros o valores intermedios).

Además, el gestor de operaciones elementales 150 puede convertir, con la ayuda del módulo de división y de estimación de tiempo de tratamiento 152, una operación principal de la memoria intermedia 140 en una pluralidad de operaciones elementales que van a ejecutarse. En los siguientes ejemplos, se trata concretamente de definir varias operaciones elementales para el cifrado de todos los bloques de un identificador que va a cifrarse, o de definir varias operaciones elementales para la limpieza de todas las zonas de memoria.

El módulo 152 estima por otro lado un tiempo de tratamiento necesario para cada operación elemental obtenida. De manera similar al módulo 120, puede preverse una tabla de correspondencia que asocia un tiempo T_{estim} a cada operación elemental (o tipo de operación elemental). Puede preverse un tiempo por defecto para cualquier operación sin referencia. Un aprendizaje a lo largo del tiempo puede permitir afinar estos tiempos, incluso definirlos para las operaciones inicialmente sin referencia. En este caso, el contador de tiempo 130 puede usarse para calcular la duración real de ejecución de cada operación elemental.

Este aprendizaje permite "adaptar" el estimador 152 a las características del elemento seguro 100.

Según las técnicas de división de la operación principal, las operaciones elementales pueden tener tamaños (en el sentido de cantidad de tratamiento necesaria) diferentes y, por tanto, asociarse a duraciones estimadas de tratamiento T_{estim} diferentes. La memoria intermedia de operaciones elementales 154 almacena estas operaciones elementales que van a ejecutarse, con su tiempo estimado T_{estim} asociado. Preferiblemente, los almacena en un orden lógico de ejecución correspondiente a su orden en la operación principal. Esto puede ser necesario si determinadas

operaciones elementales usan el resultado de operaciones elementales anteriores. Para garantizar que la selección de las operaciones elementales respeta este orden, la memoria intermedia 154 puede ser de tipo FIFO (por "First In, First Out", primero en entrar, primero en salir). Puede preverse una FIFO por cada operación principal en caso de una pluralidad de tales operaciones principales.

En una realización, la memoria intermedia 154 almacena simultáneamente de las operaciones elementales de dos o más operaciones principales. Esto permite realizar simultáneamente la ejecución progresiva de varias operaciones principales.

El módulo de tratamiento 110 puede solicitar entonces al selector de operaciones elementales 156 para elegir qué operaciones elementales ejecutar cuando hay tiempo de tratamiento disponible en una puesta en práctica de la invención.

Normalmente, el módulo de tratamiento 110 indica al selector 156 un tiempo disponible T_{dispo} . El selector 156 puede seleccionar de ese modo operaciones elementales en función de sus tiempos estimados propios T_{estim} y del tiempo disponible T_{dispo} .

Si el tiempo asignado $T_{m\acute{a}x}$ comprende el tiempo de envío $T_{env\acute{i}o}$ de la respuesta de APDU Y, el tiempo disponible T_{dispo} se reduce en este tiempo $T_{env\acute{i}o}$ antes de la selección de las operaciones elementales. Como variante, el tiempo de envío $T_{env\acute{i}o}$ puede ser despreciable.

Normalmente, el selector 156 selecciona un número máximo de operaciones elementales cuya suma de tiempos de tratamiento estimados correspondientes no supera el tiempo disponible: $\text{suma}(T_{estim}) \leq T_{dispo}$.

Durante una operación de selección para un tiempo disponible T_{dispo} , el selector 156 puede seleccionar operaciones elementales asociadas a una única operación principal. Como variante, el selector 156 puede seleccionar operaciones elementales asociadas a una segunda operación principal únicamente si no hay más operaciones elementales que van a ejecutarse para una primera

operación principal. Como variante adicional, el selector 156 puede seleccionar operaciones elementales asociadas a dos o más operaciones principales.

El selector 156 puede respetar el orden de las operaciones elementales en la o las operaciones principales, y por tanto seleccionar únicamente el o los primeros elementos (operaciones elementales) en la o las memorias intermedias FIFO.

Opcionalmente, las operaciones principales pueden estar asociadas a prioridades. Entonces, el selector 156 puede seleccionar el máximo de operaciones elementales de la operación principal que tienen la prioridad más alta que satisfacen T_{dispo} , después seleccionar el máximo de operaciones elementales de la siguiente operación principal que tiene la prioridad más alta que satisfacen el tiempo restante (T_{dispo} reducido en los tiempos estimados para las operaciones elementales ya seleccionadas) y así sucesivamente.

La figura 3 ilustra, con la ayuda de un diagrama de flujo, etapas generales de un procedimiento según la invención. Estas etapas se controlan por el módulo de tratamiento 110.

En la etapa 300, el módulo de tratamiento 110 recibe un comando de APDU X del terminal anfitrión.

En la etapa 305, el módulo de tratamiento 110 activa el contador de tiempo 130 durante la ejecución del comando de APDU con el fin de determinar una duración real de tratamiento del comando de APDU. Tal como se indicó anteriormente, esta activación puede realizarse en reacción a la recepción 300 del comando de APDU o ser simultánea al inicio de la ejecución 310 del comando de APDU.

En la etapa 310, el módulo de tratamiento 110 ejecuta el comando de APDU X según mecanismos clásicos. Por ejemplo, para un comando "read binary", se trata de leer octetos de un archivo en memoria. Para un comando "write binary", se trata de escribir octetos en un archivo en memoria.

La ejecución del comando de APDU se termine cuando se ha realizado la operación solicitada y se ha obtenido el resultado

(por ejemplo, escritura realizada u octetos leídos), y opcionalmente cuando está lista para enviarse la respuesta de APDU Y que comprende este resultado.

La etapa 315 se produce en este instante de final de ejecución, mediante la detención del contador de tiempo 130 y, por tanto, la obtención de la duración real $T_{ejecución}$ de tratamiento del comando de APDU X.

En la etapa 320, el módulo de tratamiento 110 recupera a partir del estimador 120 la duración $T_{máx}$ asignada al tratamiento de este comando de APDU X. En una realización, esta duración define el instante en el que el módulo de tratamiento 110 va a enviar la respuesta de APDU Y y, por tanto, finalizar el tratamiento del comando de APDU X visto desde el exterior (es decir, desde el terminal anfitrión).

Una vez recuperada la duración $T_{máx}$, el módulo de tratamiento 110 la compara con la duración de ejecución del comando $T_{ejecución}$. Concretamente, determina si la duración real de tratamiento indicada por el contador de tiempo (por tanto, $T_{ejecución}$) es inferior a la duración predefinida $T_{máx}$ asociada al comando de APDU, esto con el fin de determinar si queda tiempo de tratamiento disponible T_{dispo} : $T_{dispo} = T_{máx} - T_{ejecución}$.

En caso negativo (no hay tiempo disponible), el módulo de tratamiento 110 envía la respuesta de APDU Y con destino al equipo anfitrión en la etapa 330.

En caso afirmativo (hay tiempo disponible), el módulo de tratamiento 110 ejecuta una o varias de dichas operaciones elementales durante el tiempo restante T_{dispo} de dicha duración predefinida $T_{máx}$.

En una realización en la que las operaciones principales se dividen previamente en pluralidades de operaciones elementales por el módulo 152, el módulo de tratamiento 110 solicita al selector 156 indicándole la duración disponible T_{dispo} con el fin de obtener la o las operaciones elementales que van a ejecutarse. Anteriormente se describieron mecanismos de selección de las

operaciones elementales en relación con la figura 2. Concretamente, el selector 156 selecciona una o varias de dichas operaciones elementales que van a ejecutarse en función de sus duraciones de tratamiento elemental asociadas T_{estim} y de dicho tiempo T_{dispo} restante tras la ejecución del comando de APDU.

La selección, y por tanto la ejecución que conlleva, pueden comprender la selección y la ejecución de operaciones elementales procedentes de operaciones principales distintas.

La etapa 325 consiste entonces en ejecutar estas operaciones elementales seleccionadas y devueltas por el selector 156.

El contador de tiempo 130 puede usarse en esta ocasión para medir la duración real de ejecución de cada una de las operaciones elementales, con el fin concretamente de ajustar, si es necesario, el tiempo estimado T_{estim} para estas operaciones elementales.

Debe observarse que, si el selector 156 indica que no hay ninguna operación elemental que va a ejecutarse (o bien la memoria intermedia 154 está vacía o bien ninguna operación elemental en la salida de la memoria intermedia FIFO satisface la duración disponible - $T_{estim} > T_{dispo}$), el procedimiento puede pasar directamente a la etapa 330.

En otra realización, sin división previa de las operaciones principales, la etapa 325 puede consistir en continuar, durante el tiempo disponible T_{dispo} , la ejecución de una operación principal anteriormente suspendida.

Si aún no se ha ejecutado parcialmente (y por tanto suspendido) ninguna operación principal, la etapa 325 consiste en iniciar la ejecución de una operación principal almacenada en la memoria intermedia 140, por ejemplo la operación principal que tiene la prioridad más alta.

En cuanto ya se ha ejecutado parcialmente una operación principal, la etapa puede consistir en continuar la ejecución de esta operación principal desde un punto de suspensión anterior. La ejecución de la operación principal se realiza durante el

tiempo disponible T_{dispo} , es decir que se suspende de nuevo en un siguiente punto de suspensión que depende del tiempo restante T_{dispo} . Por tanto, se habrán ejecutado una o varias operaciones elementales en el interior de la operación principal.

Durante la suspensión, se memoriza el nuevo punto de suspensión, así como eventuales datos o valores de registro necesarios para la reanudación de la ejecución. Entonces, durante esta reanudación (por tanto, en una etapa 325 posterior), se recuperan los datos o valores.

Tras la etapa 325 de ejecución de operaciones elementales, la etapa 330 consiste, para el elemento seguro 100, en enviar la respuesta de APDU Y con destino al equipo anfitrión. Se observa que este envío es posterior a la etapa 325 de ejecución de las operaciones elementales. Preferiblemente, la respuesta de APDU Y se envía al caducar la duración predefinida $T_{máx}$ asociada al comando de APDU. Aplazando el envío de esta respuesta al final de la duración $T_{máx}$, se prolonga artificialmente la duración de ejecución del comando de APDU, vista desde el exterior (terminal anfitrión). De este modo, no se recibe ningún otro comando de APDU a partir del terminal anfitrión durante el periodo $T_{máx}$. Esto permite en última instancia al elemento seguro poder ejecutar siempre (de manera progresiva) una operación principal sin bloquear el tratamiento de comando de APDU pero también sin verse bloqueado por una serie continua de comandos de APDU.

Debe observarse que el procedimiento de la figura 3 puede ponerse en práctica tras la recepción de cualquier nuevo comando de APDU. En este caso, la gestión de la ejecución de una operación principal por el elemento seguro 100 comprende una pluralidad de iteraciones de las etapas anteriores de manera que se ejecutan (de manera progresiva) la pluralidad de operaciones elementales que forman la operación principal.

La figura 4 ilustra, con la ayuda de un diagrama de flujo, otra realización de un procedimiento según la invención. En esta realización, pueden ejecutarse una o varias operaciones

elementales entre el tratamiento de dos comandos de APDU vistos desde el terminal anfitrión. Dicho de otro modo, la ejecución adicional de una o varias de dichas operaciones elementales se realiza tras el envío de una respuesta a un comando de APDU recibido y previamente a la recepción de un siguiente comando de APDU.

Esta realización también ilustra las iteraciones sucesivas (mediante la flecha desde la etapa 330 hacia la etapa 400) de tratamiento de comandos de APDU recibidos con el fin de ejecutar de manera progresiva una operación principal.

En la etapa 400, el módulo de tratamiento 110 determina si se ha recibido un comando de APDU.

En caso afirmativo, se ponen en práctica las etapas 305 a 330 descritas anteriormente, observándose que, si no existe ninguna operación principal que va a ejecutarse, la etapa 325 se reduce a no hacer nada, y puede transmitirse la respuesta de APDU Y, durante la etapa 330, directamente tras la ejecución del comando, es decir sin esperar al final de la duración asignada $T_{\text{máx}}$. Dicho de otro modo, se asigna una duración prolongada $T_{\text{máx}}$ a un comando de APDU por el estimador 120 únicamente cuando hay una o varias operaciones principales que van a ejecutarse.

En caso negativo, el módulo de tratamiento 110 determina en la etapa 450 si existe una operación principal que va a ejecutarse, tanto si ya se ha ejecutado parcialmente (y por tanto suspendido) como si no. Esto equivale, por ejemplo, a verificar si la memoria intermedia 140 o 154 está vacía o no.

En caso negativo, el procedimiento vuelve a la etapa 400. Además, mientras no se recibe ningún comando de APDU u operación principal, el módulo de tratamiento 110 no realiza ningún tratamiento.

En caso afirmativo de la prueba 450, el módulo de tratamiento 110 ejecuta, en la etapa 455, una operación elemental de manera similar a lo que se describió anteriormente para la etapa 325 para una o varias operaciones elementales.

En caso de división de la operación principal, la etapa 455 consiste en recuperar la primera operación elemental de la memoria intermedia 154 y en ejecutarla.

En ausencia de división, la etapa 455 puede consistir en reanudar la ejecución de la operación principal a partir del último punto de suspensión y continuarla hasta el próximo punto de suspensión posible.

Tras esta ejecución, el procedimiento vuelve a la etapa 400. Esto permite verificar si se recibe un nuevo comando de APDU en el intervalo.

Evidentemente, la etapa 455 puede prever la ejecución de N (dos o más) operaciones elementales, en lugar de una sola, o la continuación de la ejecución hasta el Nésimo (segundo o más) próximo punto de suspensión hacia delante, antes de volver a la etapa 400.

Si se recibe un comando de APDU durante la ejecución 455, el elemento seguro 100 puede indicar al terminal anfitrión que tratará el comando un poco más tarde.

El experto en la técnica conoce normas de los mecanismos que permiten proporcionar esta indicación. A modo de ejemplo, el elemento seguro puede reenviar un estado (status word) 9300 que indica un estado ocupado (busy). Como variante, puede reenviar un octeto de procedimiento (Procedure Byte) de tipo "NULL" (valor '60'). Como variante adicional, puede usar el comando proactivo "MORE TIME" mediante una o varias secuencias 91xx/FETCH/MORE TIME/Terminal Response[9000/91xx].

Entonces, el elemento seguro puede ejecutar el comando de APDU tras la prolongación del tiempo.

De manera preferida, el elemento seguro que reenvía esta indicación como respuesta al comando de APDU, espera a que el terminal anfitrión reenvíe de nuevo el comando de APDU, comando que se tendrá en cuenta durante la próxima etapa 400 (sin que el elemento seguro solicite una nueva prolongación del tiempo).

La realización en bucle de las etapas 400-450-455 permite la ejecución sucesiva de una o varias operaciones elementales hasta la recepción de un nuevo comando de APDU.

La figura 5 ilustra una secuencia cronológica del tratamiento de varios comandos de APDU sucesivos mientras que tienen que tratarse dos operaciones principales OP1 y OP2, por ejemplo dos cifrados de identificador, dos operaciones de recogida de basura (en dos zonas de memoria diferentes) o dos operaciones de naturalezas diferentes.

Aunque las dos operaciones principales deben ejecutarse en este caso al mismo tiempo, una puede aparecer posteriormente a la otra.

Las dos operaciones principales OP1 y OP2 se representan en la parte inferior de la figura como formadas respectivamente por n y m operaciones elementales. Estas operaciones elementales se almacenan, por ejemplo, en dos memorias intermedias FIFO 154.

Tras la recepción de un comando de APDU X1 (etapa 300), el elemento seguro 100 obtiene el tiempo $T_{\text{máx1}}$ asociado, activa el contador 130 (etapa 305), ejecuta el comando de APDU X1 (etapa 310) sin enviar la respuesta de APDU Y1, después calcula el tiempo restante disponible T_{dispo} (etapas 315, 320). En el ejemplo, el tiempo disponible permite al selector 156 seleccionar las operaciones elementales 1 y 2 de OP1 (OP1_1 y OP1_2) y la operación elemental 1 de OP2, indicada OP2_1 (ya que, por ejemplo, la operación elemental 3 de OP1 necesita un tiempo T_{estim} superior al tiempo restante).

Entonces se ejecutan estas tres operaciones elementales (etapa 325). Al caducar el tiempo asignado $T_{\text{máx1}}$, el elemento seguro 100 envía la respuesta de APDU Y1 al comando de APDU X1.

Después se recibe un segundo comando de APDU X2 (etapa 300), el elemento seguro 100 obtiene el tiempo $T_{\text{máx2}}$ asociado, activa el contador 130 (etapa 305), ejecuta el comando de APDU X2 (etapa 310) sin enviar la respuesta de APDU Y2 correspondiente, después calcula el tiempo restante disponible T_{dispo} (etapas 315, 320).

En el ejemplo, el tiempo disponible T_{dispo} permite al selector 156 seleccionar una única operación elemental (la operación elemental 3 de OP1, indicada OP1_3). Entonces se ejecuta esta operación elemental (etapa 325). Al caducar el tiempo asignado $T_{máx2}$, el elemento seguro 100 envía la respuesta de APDU Y2 al comando de APDU X2.

Se realizan las mismas etapas para el comando de APDU X3 y los siguientes. Por tanto, de manera progresiva se ejecutan las operaciones elementales de OP1 y OP2.

Llega el comando de APDU XN (etapa 300). El elemento seguro 100 obtiene el tiempo $T_{máxN}$ asociado, activa el contador 130 (etapa 305), ejecuta el comando de APDU XN (etapa 310) sin enviar la respuesta de APDU YN correspondiente, después calcula el tiempo restante disponible T_{dispo} (etapas 315, 320). En el ejemplo, el tiempo disponible T_{dispo} permite al selector 156 seleccionar las dos últimas operaciones elementales de OP1 (OP1_n-1 y OP1_n) y la operación elemental j de OP2, indicada OP2_j. Entonces se ejecutan estas operaciones elementales (etapa 325). Al caducar el tiempo asignado $T_{máxN}$, el elemento seguro 100 envía la respuesta de APDU YN al comando de APDU XN.

En este estado, se termina la ejecución de la operación principal OP1 ya que se han ejecutado todas sus operaciones elementales. El resultado de la operación principal puede usarse entonces por el elemento seguro. Por ejemplo, este último dispone de uno o varios nuevos identificadores cifrados, o dispone de nuevas zonas de memoria recicladas.

Evidentemente, comandos de APDU posteriores van a permitir terminar la ejecución progresiva de OP2.

Los ejemplos anteriores son simplemente realizaciones de la invención.

Se hace constar que con relación a esta fecha, el mejor método conocido por la solicitante para llevar a la práctica la citada invención, es el que resulta claro de la presente descripción de la invención.