(19) OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 737 703**

(51) Int. CI.:

H04L 9/12 (2006.01)
H04L 9/32 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **12.10.2015** **E 15189420 (1)**

(97) Fecha y número de publicación de la concesión europea: **10.04.2019** **EP 3010175**

(54) Título: **Reinyección de un lote de comandos seguros en un canal seguro**

(30) Prioridad:

13.10.2014 FR 1459800

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
15.01.2020

(73) Titular/es:

IDEMIA FRANCE (100.0%)
2 place Samuel de Champlain
92400 Courbevoie, FR

(72) Inventor/es:

VALLIERES, JEAN-PHILIPPE y
NEROT, SÉBASTIEN

(74) Agente/Representante:

ISERN JARA, Jorge

DESCRIPCIÓN

Reinyección de un lote de comandos seguros en un canal seguro

Campo de la invención

La presente invención se refiere al campo de las comunicaciones y más particularmente el de las comunicaciones entre un dispositivo maestro, por ejemplo un servidor, y un dispositivo esclavo, por ejemplo un cliente.

CONTEXTO DE LA INVENCIÓN

Dos entidades comunicantes definen una capa de comunicación para intercambiar mensajes. Esta capa de comunicación está definida, a menudo con ayuda de un protocolo llamado de canal seguro (o *"Secure Channel*" de acuerdo con la terminología anglosajona), que permite ofrecer diferentes niveles de protección e las comunicaciones y de los datos, por ejemplo en términos de autentificación de las entidades, de integridad de los datos y de confidencialidad.

Dicho protocolo se utiliza ampliamente en la industria de tarjetas con chip para intercambiar datos de acuerdo con una relación maestro-esclavo. El maestro inicia el canal de comunicación y a continuación la comunicación mediante el envío de comandos al esclavo, respondiendo el esclavo solo a él.

A título ilustrativo, la tarjeta con chip, y de forma general cualquier elemento seguro (SE para "*Secure Element*"), tarjeta con circuito integrado (ICC para "*Integrated Circuit Card*"), universal (UICC) incluso incorporada (eUICC), puede hacer las veces de esclavo, por ejemplo en modo "servidor" en un contexto cliente/servidor. Un equipo remoto puede desempeñar el papel de maestro y, por lo tanto, de "cliente" en un contexto cliente/servidor.

Para alcanzar un nivel de seguridad elevado, los protocolos de canal seguro emplean mecanismos de seguridad que se apoyan en una o varias variables dinámicas, es decir en variables que evolucionan en el tiempo.

Los comandos y mensajes intercambiados entre el maestro y el esclavo se generan entonces a partir de esta o estas variables dinámicas, opcionalmente mediante claves de sesión, que impiden de este modo reinyectarlos si las variables dinámicas (y, por lo tanto, las claves de sesión en el ejemplo) han evolucionado.

Por ejemplo, estas variables dinámicas pueden incluir factores aleatorios intercambiados entre el maestro y el esclavo en un proceso de autentificación (tipo *challenge-response*), y/o incluir un contador de secuencia que se incrementa con cada nuevo establecimiento de un canal seguro entre las dos entidades. Este es, por ejemplo, el caso en los protocolos SCP02 y SCP03 (para "*Secure Channel Protocol*" 02 y 03) ampliamente utilizados en la industria de las tarjetas con chip y definidos en la norma GlobalPlatform.

En estos protocolos, las variables dinámicas se utilizan para generar una o varias claves de sesión utilizadas para autentificar y/o cifrar los datos intercambiados y para generar un código de autentificación de mensaje (o MAC para "*Message Authentication Code*"). Los comandos son seguros o están protegidos de este modo.

En varios de estos protocolos de canal seguro, los diferentes comandos seguros enviados por el maestro (ídem para las respuestas seguras enviadas por el esclavo) están encadenados, de modo que la generación del comando (de la respuesta) N+1 necesita que el comando (respuesta) anterior N sea válido.

Por ejemplo, el código de autentificación de un mensaje (comando o respuesta) puede ser de tipo CBC-MAC (para "*Chained Block Ciffering*") en el que el valor de inicio es el CBC-MAC del mensaje anterior.

Asimismo, el cifrado CBC de los datos puede hacer intervenir un contador interno que se incrementa para cada nuevo comando, de manera que, en un mismo canal seguro, los comandos seguros son diferentes incluso aunque los datos que contienen sean los mismos.

El objetivo de la cadena es asegurar que los comandos (respuestas) seguros son enviados y procesados en un orden previsto. Permite también una detección más fácil de comandos (respuestas) no previstos (por ejemplo un comando de reinicialización), por ejemplo enviados por una persona malintencionada.

Las explicaciones a continuación se apoyan principalmente en ejemplos en los que los comandos están encadenados. No obstante, la presente invención no está limitada a dichos comandos encadenados.

Ciertos de los protocolos de canal seguro permiten de pre-generar comandos seguros, sin necesitar en esta fase comandos o respuestas anteriores. Para hacer esto, la entidad en cuestión, generalmente el maestro, dispone de mecanismos que le permiten predecir (o simular o reproducir) el comportamiento de la otra entidad, en este contexto el esclavo, en concreto la evolución de las variables dinámicas y las respuestas que el esclavo debería generar. Se habla de un modo predictivo del protocolo de canal seguro.

Por ejemplo, los datos necesarios para el modo predictivo pueden comprender una o varias claves secretas compartidas entre el maestro y el esclavo, a partir de las cuales se generan las claves de sesión; las variables dinámicas, normalmente un contador de secuencia incrementado para cada nuevo establecimiento de un canal seguro, que permite también generar las claves de sesión; también los algoritmos que permiten predecir (o pregenerar) valores seudo-aleatorios utilizados durante intercambios de autentificación (*challenge-response*).

El modo predictivo es utilizado a menudo por el dispositivo maestro para pregenerar un lote (o juego o conjunto o serie) de comandos seguros, generalmente encadenados, que están empaquetados en un fichero o "script" o "batch" como se muestra en la figura 1. En el lote de esta figura, el primer comando permite iniciar y establecer un canal seguro, el comando siguiente, que es seguro, contribuye a un mecanismo de autentificación, a continuación los otros comandos seguros pretenden gobernar procesamientos particulares en el dispositivo esclavo.

La figura 1A muestra otro ejemplo de lote de comandos seguros sin proceso de autentificación. El comando SELECT APPLICATION es opcional ya que se puede realizar de forma descorrelacionada, antes del lote de los comandos seguros STORE DATA. Además, no consta de ninguna indicación sobre una eventual securización del canal de comunicación.

De este modo, el primer comando seguro STORE DATA (identificable gracias al número de bloque = 0 contenido en el parámetro P2) es el primer comando que indica el establecimiento de un canal seguro, ya que su campo CLA contiene una indicación de que es seguro.

Los otros comandos STORE DATA también son seguros y están encadenados.

En la práctica, se recurre a los scripts de comandos cuando los dispositivos maestro y esclavo son relativamente remotos y cuando una o varias entidades intermedias, o dispositivos terceros, se intercalan entre ellos, como se muestra en la figura 2. De este modo, un dispositivo tercero en enlace directo (que emplea por ejemplo el protocolo SSL - para "*Secure Sockets Layer*" o capa de tomas seguras) con el dispositivo esclavo interactúa con este último para la cuenta del dispositivo maestro.

A modo de ejemplo, el dispositivo tercero puede ser una aplicación (y/o sistema) implementada en un terminal en el que está incorporado el dispositivo esclavo (SE, UICC, etc.), en concreto una midlet (aplicación) encargada de despertar al dispositivo esclavo antes de comunicarse con él. Como variante, el dispositivo tercero puede ser una máquina cualquiera en una red de comunicación, es decir un intermedio de transporte.

El dispositivo tercero no es necesariamente un dispositivo de confianza. En concreto, no dispone de los datos que garantizan la protección del canal seguro (de hecho, comandos seguros del lote), en particular secretos compartidos, claves de sesión, algoritmos utilizados, etc.

En la práctica, el dispositivo tercero se encarga de transferir los comandos y eventuales respuestas seguras, ya que no conoce estos datos que garantizan la protección del canal seguro y es, por lo tanto, incapaz de procesar los comandos y respuestas transferidos de este modo. Además, puede verificar un estatus general de procesamiento de los comandos para determinar si estos pueden ser procesados con éxito, y puede reenviar al dispositivo maestro, al finalizar el procesamiento, un mensaje de estatus general que indica por ejemplo el número de comandos transmitidos. El dispositivo maestro puede, de este modo, tomar decisiones a partir del mensaje de estatus general recibido.

En el marco de los lotes de comandos, el dispositivo tercero puede desempeñar un papel de temporizador de los comandos, mediante el que decide enviar los comandos al dispositivo esclavo en un momento oportuno, por ejemplo a horas en las que las comunicaciones con el dispositivo esclavo son escasas en número. De este modo, el dispositivo maestro puede recibir el mensaje de estatus general solamente varios días después de haber enviado el script de comandos al dispositivo tercero.

Además o como variante, la transferencia de los comandos se puede realizar de forma regular, por ejemplo de forma similar a las técnicas CRON o CRONTABLE de un sistema Unix.

El modo predictivo de un protocolo de canal seguro se utiliza durante etapas de producción de chips (dispositivo esclavo) para acelerar el proceso de producción. En efecto, los comandos pre-generados pueden ser, de este modo, enviados en lote, minimizando el tiempo de procesamiento de los comandos.

Otra utilización del modo predictivo atañe a los componentes (tarjetas, SE, eUICC) desplegados, por ejemplo tarjetas SIM de un operador telefónico o elementos seguros que equipan dispositivos de recuento a distancia (máquina a máquina). En este caso de despliegue, es difícil establecer un enlace de comunicación continuo (a causa de un cortafuegos, de una zona informática desmilitarizada DMZ, etc.). Puede mostrarse interesante transmitir el lote de comandos a un dispositivo tercero cercano al componente esclavo. Se conocen, en concreto, los métodos OTA ("*Over-The-Air*") en la industria móvil en los que un operador (el maestro) pre-genera una serie de comandos para las tarjetas SIM (esclavas) de los abonados para programarlas /reconfigurarlas. Esta serie de comandos puede ser suministrada en cualquier momento a un dispositivo tercero, generalmente el terminal del abonado, responsable de transmitir los comandos a la tarjeta SIM diana.

El diagrama de la figura 3 ilustra una utilización convencional de un lote de comandos en el sistema de la figura 2.

En este ejemplo, el dispositivo maestro genera un lote de comandos a enviar al dispositivo esclavo, por ejemplo el de la figura 1 en la que el lote de comandos incluye un comando, llamado primer comando (por ejemplo el comando de inicialización INITIALIZE UPDATE según SCP03), que indica el establecimiento, con el dispositivo esclavo, de un canal de comunicación seguro gracias a, al menos, una clave de sesión (por ejemplo la clave S-MAC según SCP03), siendo esta clave de sesión dependiente de un valor actual de una variable dinámica compartida entre el dispositivo maestro y el dispositivo esclavo, e incluye al menos un comando, llamado segundo comando, seguro (o protegido) gracias a un código de autentificación de mensaje calculado con ayuda de la clave de sesión.

Esta generación implementa el modo predictivo del protocolo de canal seguro, que implica la actualización de la variable dinámica compartida (el contador de secuencia en SCP03).

De este modo, el dispositivo tercero que recibe, del dispositivo maestro, el lote de comandos a enviar al dispositivo esclavo, lo inyecta para enviar, al dispositivo esclavo, dichos comandos del lote de forma secuencial, comando por comando, en tanto que no se detecte ningún error. Como se muestra en la figura, inyectar el lote de comandos vuelve en bucle en cada comando, para que este último sea enviado y que se devuelva una respuesta (un mensaje de estatus de ejecución generalmente).

El canal seguro se cierra en cuanto aparece un error. Un ejemplo de error es la expiración de un temporizador (*timer out*) después del envío de un comando que quedó sin respuesta, o un mensaje de error devuelto por el dispositivo esclavo.

Por su lado, el dispositivo esclavo recibe los comandos de forma secuencial, es decir sin tener consciencia de la existencia de un lote o script de comandos inyectado por el dispositivo tercero.

De este modo, el dispositivo esclavo recibe, del dispositivo tercero, al menos un comando, llamado primer comando, que indica el establecimiento de un canal de comunicación seguro gracias a, al menos, una clave de sesión; calcula un valor de actualización de la o las variables dinámicas (en concreto el contador de secuencia) a partir de un valor actual de la o las variables dinámicas almacenadas en memoria no volátil; genera la clave de sesión a partir del valor de actualización calculado. A continuación teniendo en cuenta el envío secuencial (bucle) de los comandos, el dispositivo esclavo recibe también, del dispositivo tercero, al menos un comando, llamado segundo comando, seguro gracias a un código de autentificación de mensaje (tipo C-MAC según SCPP03); y verifica el código de autentificación de mensaje con ayuda de la clave de sesión antes de ejecutar dicho segundo comando.

Generalmente, el dispositivo esclavo reenvía un mensaje de estatus de ejecución (a veces datos) en respuesta a cada comando procesado.

Diferentes situaciones pueden conducir a que el dispositivo tercero emita un estatus de fracaso de la ejecución del lote de comandos, y por lo tanto al cierre del canal seguro: ausencia de respuesta del dispositivo esclavo a los comandos, mensaje de error recibido del dispositivo esclavo, desconexión de la red del dispositivo esclavo por ejemplo como consecuencia de falta de batería, etc.

Cuando el dispositivo maestro recibe dicho mensaje de estatus de fracaso, puede desear ejecutar de nuevo los mismos comandos, mientras que el canal seguro ha estado cerrado. Ahora bien, las variables dinámicas a partir de las cuales se definen los mecanismos de seguridad del canal seguro han evolucionado: las claves de sesión de un nuevo canal seguro serán diferentes de las del canal seguro que acaba de estar cerrado. De este modo, es habitual que el dispositivo maestro tenga que generar, en modo predictivo, un nuevo lote de los mismos comandos, pero calculando nuevos códigos de autentificación y recifrando los datos de los comandos con ayuda de nuevas claves de sesión predichas.

Ahora bien, esta situación no es satisfactoria ya que, para que el lote de comandos sea ejecutado con éxito, nuevas conexiones, nuevos procesamientos criptográficos (y por lo tanto costosos) son necesarios para que el dispositivo maestro pueda generar el nuevo lote de comandos. Esto disminuye la capacidad del dispositivo maestro para procesar otras solicitudes u operaciones estándar.

Además, teniendo en cuenta que el procesamiento del lote de comandos por el dispositivo tercero y el dispositivo esclavo puede estar descorrelacionado, en el tiempo, del envío de este mismo lote por el dispositivo maestro, este último es informado tardíamente del fracaso, conduciendo de este modo a una pérdida importante de tiempo (a veces en días) para el dispositivo maestro. Esta pérdida de tiempo puede mostrarse, por ejemplo, perjudicial si se deben realizar actualizaciones, por lote de comandos, en una flota de componentes desplegados. En efecto, esta situación disminuye las capacidades del dispositivo maestro para programar o controlar rápidamente la flota de componentes desplegados geográficamente.
Este perjuicio es tanto más significativo en cuanto que las razones del fracaso de la ejecución del lote de comandos

pueden ser mínimas y a veces muy puntuales: tentativa de ataque por una persona malintencionada, perturbación (por ejemplo interferencias) en el enlace de comunicación entre los dispositivos tercero y esclavo, etc.

La invención pretende resolver todos o parte de los inconvenientes mencionados anteriormente.

Se conoce también el documento US 2004/148502 que describe la transferencia de código fuente encriptado mediante claves de sesión, entre una tarjeta con chip y un servidor de compilación, y la transferencia del código compilado encriptado por otras claves de sesión, entre el servidor y la tarjeta.

El documento US 2012/159502 describe, por su parte, la asignación de recursos CPU en función de dos contadores, siendo actualizado uno cuando el segundo alcanza un valor umbral.

Sumario de la invención

El objeto de la invención se define mediante las reivindicaciones. En este contexto, la invención propone un procedimiento de comunicación que comprende, a nivel de un dispositivo esclavo, las etapas mencionadas anteriormente, y además las etapas siguientes:

incrementar, en un incremento positivo o negativo (en cuyo caso se habla también de decrecer), un contador de ensayos en la recepción del primer comando que indica el establecimiento de un canal de comunicación seguro; y sobreescribir el valor actual de la variable dinámica en memoria no volátil con el valor de actualización calculado cuando el contador de ensayos alcanza un valor umbral. En otros términos, registrar el valor de actualización calculado (de la variable dinámica), en memoria no volátil como nuevo valor actual de la variable dinámica cuando el contador de ensayos alcanza un valor umbral.

Gracias a la utilización de un contador de ensayos que retarda la actualización concreta de la variable dinámica en memoria no volátil (y, por lo tanto, antes de que cualquier nuevo canal seguro necesita nuevos datos que garanticen la protección del canal seguro, por ejemplo nuevas claves de sesión, y, por lo tanto, antes de que el lote de comandos esté obsoleto), el dispositivo esclavo permite al dispositivo tercero poder realizar varios ensayos de la inyección de comandos a hacer ejecutar por el dispositivo esclavo. En efecto, las claves de sesión e los canales seguros establecidos durante estas reinyecciones son las mismas. Esta pluralidad de ensayos permite, en concreto, superar eventuales ataques y/o perturbaciones de comunicación para ejecutar la totalidad del lote de comandos.

De este modo, la reinyección de un lote de comandos por el dispositivo tercero es posible, lo que reduce el recurso al dispositivo maestro para esta ejecución integral. El número de conexiones, la cantidad de procesamientos criptográficos necesarios para el dispositivo maestro y el tiempo necesario para esta ejecución integral, por lo tanto, se reducen. Además, el dispositivo maestro está más disponible para procesar otros comandos.

La validación de la actualización de la variable dinámica compartida cambiando el valor de actualización como nuevo valor actual en memoria no volátil es también conocido con el término de "*commitment*" en la terminología anglosajona.

De forma simétrica, la invención prevé también un procedimiento de comunicación que comprende, a nivel de un dispositivo tercero, las etapas mencionadas anteriormente, y además la etapa siguiente:
reinyectar el lote de comandos en tanto que el dispositivo tercero no reciba, del dispositivo esclavo, una indicación de una actualización del valor actual de la variable dinámica por el dispositivo esclavo. En otros términos, en tanto que el dispositivo tercero no reciba, del dispositivo esclavo, un acontecimiento de salida de un bucle de reinyección del lote de comandos.

Como se verá a continuación, las indicaciones pueden ser indirectas, es decir implícitas a informaciones transmitidas.

La invención permite una reinyección controlada de un lote de comandos seguros, gracias a que se tienen en cuenta una indicación del dispositivo esclavo sobre la actualización efectiva o no de la variable dinámica compartida a partir de la cual se establecen las protecciones de seguridad del canal seguro.

Este comportamiento del dispositivo tercero permite de este modo a un dispositivo esclavo utilizar un contador de ensayos como se ha mencionado anteriormente, para reducir el recurso al dispositivo maestro para ejecutar con éxito la totalidad del lote de comandos.

Correlativamente, la invención también se refiere a un dispositivo de procesamiento, por ejemplo una tarjeta con chip, que comprende una memoria no volátil que memoriza un valor actual de una variable dinámica, y un procesador configurado para:

recibir, de un dispositivo tercero, al menos un comando, llamado primer comando, que indica el establecimiento de un canal de comunicación seguro gracias a, al menos, una clave de sesión;
calcular un valor de actualización de la variable dinámica a partir del valor actual almacenado en memoria no volátil;
generar la clave de sesión a partir del valor de actualización calculado;

recibir, del dispositivo tercero, al menos un comando, llamado segundo comando, seguro gracias a un código de autentificación de mensaje; y
verificar el código de autentificación de mensaje con ayuda de la clave de sesión antes de ejecutar dicho segundo comando;
caracterizado por que el dispositivo de procesamiento comprende además un contador de ensayos, y el procesador está configurado además para:

incrementar el contador de ensayos en la recepción del primer comando que indica el establecimiento de un canal de comunicación seguro; y
sobreescribir el valor actual de la variable dinámica en memoria no volátil con el valor de actualización calculado cuando el contador de ensayos alcanza un valor umbral.

Simétricamente, la invención también se refiere a un dispositivo de procesamiento que comprende un procesador configurado para:

recibir, de un dispositivo maestro, un lote de comandos a enviar a un dispositivo esclavo, incluyendo el lote de comandos un comando, llamado primer comando, que indica el establecimiento, con el dispositivo esclavo, de un canal de comunicación seguro gracias a, al menos, una clave de sesión dependiente de un valor actual de una variable dinámica compartida entre el dispositivo maestro y el dispositivo esclavo, y al menos un comando, llamado segundo comando, seguro gracias a un código de autentificación de mensaje calculado con ayuda de la clave de sesión;
inyectar el lote de comandos para enviar, al dispositivo esclavo, dichos comandos del lote de forma secuencial, comando por comando, en tanto que no se detecte ningún error;
caracterizado por que el proceso está configurado además para:
reinyectar el lote de comandos en tanto que el dispositivo de procesamiento no reciba, del dispositivo esclavo, una indicación de una actualización del valor actual de la variable dinámica por el dispositivo esclavo.

Asimismo, la invención propone un sistema que comprende un dispositivo esclavo tal como se ha definido anteriormente, un dispositivo tercero tal como se ha definido anteriormente, y un dispositivo maestro configurado para generar y enviar el lote de comandos al dispositivo tercero.

La invención también tiene por objeto un producto de programa informático que comprende instrucciones adaptadas a la implementación de cada una de las etapas de uno de los procedimientos descritos anteriormente cuando dicho programa es ejecutado en un ordenador.

Los dispositivos y producto de programa informático de acuerdo con la invención presentan ventajas similares a las expuestas anteriormente en relación con los procedimientos.

Otras características de los procedimientos y dispositivos de acuerdo con realizaciones se describen en las reivindicaciones dependientes, esencialmente con ayuda de una terminología de procedimiento, extrapolables a los dispositivos.

En una realización, la etapa de cálculo de un valor de actualización comprende el registro del valor de actualización calculado en una memoria volátil del dispositivo esclavo.

Esta disposición garantiza que el valor actual de la variable dinámica no es modificado por el o los valores calculados para su actualización. De este modo, se conserva la posibilidad de establecer uno o varios canales seguros con ayuda el mismo valor actual de la variable dinámica, que permanece inalterado en memoria no volátil. De ello resulta que la reinyección, por el dispositivo tercero, de un lote de comandos que requiere el establecimiento de dicho canal seguro se hace posible y está garantizada.

En otra realización, el valor actual de la variable dinámica en memoria no volátil se sobreescribe en memoria no volátil con el valor de actualización calculado cuando el último comando seguro del conjunto de los segundos comandos seguros (por ejemplo mediante un indicador en el comando en cuestión) comprende un código de autentificación de mensaje verificado válidamente y es ejecutado con éxito.

Esta situación corresponde a la ejecución integral de los comandos seguros del lote. El canal seguro no es, en adelante, necesario y se puede cerrar, en cuyo caso conviene actualizar concretamente la o las variables dinámicas para que los próximos canales seguros se establezcan con nuevos datos de seguridad (claves de sesión por ejemplo). Esta situación es un ejemplo de acontecimiento de salida del bucle de reinyección, ya que dicha reinyección se muestra inútil teniendo en cuenta la ejecución realizada con éxito.
En una realización particular, el procedimiento (del lado del dispositivo esclavo) comprende además, la etapa siguiente: transmitir, al dispositivo tercero y en respuesta a dicho último comando seguro, un mensaje que comprende una indicación de que la ejecución de dicho o de dichos segundos comandos del lote se ha realizado con éxito. Esta indicación permite, en concreto, al dispositivo tercero se indirectamente informado de la actualización efectiva de la variable dinámica (el contador de secuencia en SCP03) en memoria no volátil del dispositivo esclavo.

En todavía otra realización, el valor actual de la variable dinámica en memoria no volátil se sobreescribe en memoria no volátil con el valor de actualización calculado cuando un llamado segundo comando cuyo código de autentificación de mensaje ha sido verificado con éxito conduce a un error o excepción de ejecución de este segundo comando, como por ejemplo cuando un criptograma de anfitrión (procedimiento de autentificación descrito más adelante) se declara inválido durante un comando EXTERNAL AUTHENTICATE (en SCP03) o cuando el comando desencriptado demuestra ser erróneo. Estas situaciones están libres de acontecimiento de salida del bucle de reinyección, ya que, en estas situaciones, la reinyección del mismo comando conducirá al mismo error: la reinyección del lote de comandos seguros se muestra, por lo tanto, inútil.

Esta configuración fuerza la actualización efectiva de la variable dinámica en memoria no volátil (commitment o cambio), y, por lo tanto, hace imposible, en adelante, la reinyección del lote de comandos, ya que el error o excepción de ejecución traduce un error intrínseco a los comandos seguros del lote, que no podrá ser corregido por una simple reinyección, sino por el dispositivo maestro generando un nuevo lote de los mismos comandos. La disposición en este contexto propuesta permite, por lo tanto, evitar una pérdida de tiempo vinculada a la reinyección de un lote de comandos intrínsecamente erróneo.

En una realización particular, el procedimiento (del lado del dispositivo esclavo) comprende además, la etapa siguiente: transmitir, al dispositivo tercero y en respuesta a dicho segundo comando que conduce a un error o excepción de ejecución, un mensaje que comprende una indicación del error o excepción de ejecución a pesar de un código de autentificación de mensaje verificado con éxito. Esta indicación permite también al dispositivo tercero se indirectamente informado de la actualización efectiva de la variable dinámica (el contador de secuencia en SCP03) en memoria no volátil del dispositivo esclavo.

Se comprende de estas diversas disposiciones que el commitment o cambio de la variable dinámica (el contador de secuencias en SCP02 o SCP03 por ejemplo) solamente tiene lugar cuando se produce uno de los acontecimientos de salida del bucle de reinyección (alcance del valor umbral de contador, fin del lote de comandos, detección de un comando erróneo en el lote a pesar de su autenticidad y su integridad). Fuera de dichos acontecimientos, el lote de comandos puede ser reinyectado si se produce un error inesperado, y esto en tanto que el número de ensayos autorizado no se alcance.

En una realización, el contador de ensayos se reinicializa a un valor por defecto cuando el valor actual de la variable dinámica en memoria no volátil se sobreescribe con el valor de actualización calculado. Por ejemplo, este valor por defecto, memorizado en memoria EEPROM, puede ser modificable en el tiempo.

Esta disposición permite reconfigurar el sistema para una nueva serie de reinyecciones de un nuevo lote de comandos, debiendo los comandos de ce nuevo lote ser seguros con ayuda de la variable dinámica tal como efectivamente actualizada (es decir con el nuevo valor actual).

En una realización particular, la etapa que consiste en sobreescribir el valor actual en memoria no volátil con el valor de actualización calculado y la etapa que consiste en reinicializar el contador de ensayos se realizan antes de la etapa de generación de la clave de sesión a partir del valor de actualización calculado.

Esta disposición garantiza una fuerte seguridad del procedimiento de acuerdo con la invención, reduciendo los riesgos de que una persona malintencionada pueda reinyectar indefinidamente un conjunto de operaciones por el dispositivo esclavo, ya que el commitment o cambio de la variable dinámica se realiza como muy pronto en el procedimiento de inicialización del canal seguro.

En concreto la etapa de registro puede preceder a la de incremento.

En una realización particular, la etapa que consiste en sobreescribir el valor actual en memoria no volátil con el valor de actualización calculado y la etapa que consiste en reinicializar el contador de ensayos se realizan en una etapa atómica.

Esta disposición permite reducir significativamente los riesgos de una desincronización entre la variable dinámica y el contador de ensayos, y como consecuencia las posibilidades de que una persona malintencionada reinyecte indefinidamente el lote de comandos.

En una realización, dicho primer comando es un llamado segundo comando seguro gracias a un código de autentificación de mensaje. Una indicación de la implementación de un mecanismo de securización del canal (encriptado y MAC por clave) puede estar previsto en un campo no seguro y, por lo tanto, legible (por ejemplo el código CLA en SCP03) de la cabecera del comando, permitiendo activar la generación de las claves de sesión para desencriptar el comando y verificar el código MAC.

En una realización relativa al dispositivo tercero, recibir la indicación de una actualización (efectiva) de la variable dinámica (compartida) por el dispositivo esclavo comprende recibir, del dispositivo esclavo, un valor actual de la variable dinámica y comparar el valor actual recibido con un valor local de la variable dinámica. El dispositivo tercero puede, de este modo, verificar directamente si el terminal esclavo ha actualizado, de forma pública, su variable dinámica por comparación con el último valor (local) del que tiene conocimiento.

En concreto, el valor actual de la variable dinámica puede ser recibido, por el dispositivo tercero, en una respuesta del dispositivo esclavo al primer comando. Esta disposición se apoya en los protocolos ya existentes sin modificarlos, al tiempo que necesita pocos procesamientos adicionales para el dispositivo tercero. En efecto, es habitual que el contador de secuencia (es decir una variable dinámica compartida en el caso de SCP02 y SCP03) sea devuelto en respuesta al comando de inicialización (INITIALIZE UPDATE en estos mismos protocolos), permitiendo al dispositivo tercero, por simple comparación con el último valor del contador del que tiene conocimiento, determinar eficazmente si una actualización efectiva (commitment) del contador de secuencia ha tenido o no lugar en el intervalo.

En otra realización, la indicación de una actualización del valor actual (que permite, de este modo, decidir sobre la reinyección de un lote actual de comandos) comprende una indicación de que la ejecución de dicho o de dichos segundos comandos del lote por el dispositivo esclavo se ha realizado con éxito.

En otra realización, la indicación de una actualización del valor actual comprende una indicación de que la ejecución, por el dispositivo esclavo, de un segundo comando del lote cuyo código de autentificación de mensaje ha sido verificado con éxito ha conducido a un error de este segundo comando. Como se ha expuesto más arriba, esta configuración incluye el caso en el que la verificación del criptograma de anfitrión del comando EXTERNAL AUTHENTICATE (SCP03) ha fracasado a pesar de un código MAC válido.

Preferentemente, los tres ejemplos de indicación definidos anteriormente para el dispositivo tercero (valor actual, indicación del éxito o de un error a pesar de un código MAC válido) son los únicos acontecimientos de salida del bucle de reinyección a partir de los cuales el dispositivo tercero sabe que no puede reinyectar el lote de comandos. En efecto, como se ha explicado más arriba, la variable dinámica compartida ha sido, en estos casos, actualizada, lo que hace que los códigos de autentificación MAC incluso los cifrados de los comandos en el lote actual están, en adelante, desincronizados respecto a las nuevas claves de sesión.

En otra realización, recibir la indicación de una actualización (efectiva) de la variable dinámica por el dispositivo esclavo comprende recibir, del dispositivo esclavo, un mensaje de error (por ejemplo 0x6982 en SCP03 de GlobalPlatform) en respuesta a un llamado segundo comando inmediatamente subsiguiente al primer comando, indicando el mensaje de error un código de autentificación de mensaje erróneo de dicho segundo comando inmediatamente subsiguiente al primer comando.

En esta configuración, el dispositivo tercero considera que si, desde el primer comando protegido por un código MAC (basado en la variable dinámica, mediante las claves de sesión), el código MAC es erróneo (mensaje de error 0x6982 en la norma GlobalPlatform), entonces es probable que las claves de sesión (y en consecuencia la variable dinámica) no sean las previstas para los códigos MAC generados previamente en el lote de comandos. En efecto, es probable que la variable dinámica se haya incrementado del lado del dispositivo esclavo dando como resultado una desincronización con el lote de comandos predicho anteriormente. Se comprenderá que esta disposición no es óptima ya que un error de código MAC también puede resultar de una perturbación del canal de comunicación.

En una realización, los códigos de autentificación de mensaje de varios segundos comandos están encadenados, en concreto, en el sentido de que el cálculo del código de autentificación MAC de un segundo comando siguiente depende del código MAC de un segundo comando precedente. Esta disposición securiza la ejecución del lote de comandos contra ataques que pretenden insertar comandos no previstos o desordenar los comandos definidos en el lote.

Breve descripción de las figuras

Otras particularidades y ventajas de la invención aparecerán aún en la descripción a continuación, ilustrada por los dibujos adjuntos, en los que:

- las figuras 1 y 1A ilustran ejemplos de lotes o batchs de comandos pre-generados;
- la figura 2 representa esquemáticamente un sistema en el que se implementan realizaciones de la invención;
- la figura 3 ilustra una utilización convencional de un lote de comandos en el sistema de la figura 2;
- la figura 4 ilustra un ejemplo de arquitectura material de cada dispositivo constitutivo del sistema descrito en referencia a la figura 2;
- la figura 5 ilustra esquemáticamente los intercambios convencionales de acuerdo con el protocolo SCP03 definido en GlobalPlatform;
- la figura 5A ilustra esquemáticamente intercambios en un modo predictivo del protocolo SCP03;
- la figura 5B ilustra esquemáticamente intercambios en un modo predictivo de un lote de comandos sin procedimiento de autentificación;
- la figura 6 ilustra, en el mismo esquema que la figura 3, la posibilidad de reinyección de un lote de comandos de acuerdo con realizaciones de la invención;
- la figura 7 ilustra, en forma de ordinograma, etapas generales de una realización de la invención del lado del

dispositivo esclavo;
- la figura 8 ilustra, en forma de ordinograma, etapas generales de una realización de la invención del lado del dispositivo tercero;
- la figura 9 ilustra, retomando el esquema de la figura 5A, intercambios de una realización de la invención que se apoya en el modo predictivo del protocolo SCP03;
- la figura 9A ilustra, retomando el esquema de la figura 5B, intercambios de una realización de la invención de un lote de comandos predichos sin procedimiento de autentificación; y
- la figura 10 ilustra el efecto de temporización de la actualización efectiva en memoria no volátil del contador de secuencias durante una implementación de la invención.

Descripción detallada de la invención

La figura 2 ilustra, de manera esquemática, un ejemplo de sistema en el que se pueden implementar realizaciones de la presente invención. El sistema 10 comprende un dispositivo maestro 12, por ejemplo un servidor de un operador telefónico, y un dispositivo esclavo 14, por ejemplo una tarjeta SIM, que el operador telefónico desea actualizar. El dispositivo esclavo 14 está, por ejemplo, incorporado en un terminal móvil, que desempeña el papel de dispositivo tercero 16 en intercambios entre los dispositivos maestro 12 y esclavo 14. Otros dispositivos intermedios (no representados) entre los dispositivos maestro 12 y esclavo 14 pueden existir en el marco de la invención.

En este ejemplo, el servidor del operador telefónico se denomina "maestro" ya que suya es la iniciativa de los intercambios con la tarjeta SIM, que es por lo tanto "esclavo". Para realizar estos intercambios, el dispositivo maestro inicia un canal seguro, por ejemplo según el protocolo SCP03, a continuación envía una serie de comandos a la tarjeta SIM. Otros protocolos y modos degradados de SCP02 y SCP03 prevén librarse de un procedimiento de inicialización del canal seguro, y permiten el envío directo de comandos seguros. En este caso, el primer comando seguro indica que un canal seguro está implícitamente establecido, generalmente a partir de secretos compartidos entre el maestro y el esclavo.

El enlace entre el dispositivo maestro 12 y el dispositivo tercero 16 puede ser por cable (conexión Ethernet) o inalámbrica (red móvil). En el ejemplo en el que el dispositivo tercero 16 es un terminal que incorpora una tarjeta SIM 14, el enlace de comunicación entre estas dos entidades es de tipo con contacto. Por supuesto, pueden estar previstas variantes sin contacto (inalámbricas).

La figura 4 ilustra un ejemplo de arquitectura material de cada dispositivo constitutivo del sistema descrito en referencia a la figura 2.

En este ejemplo, el dispositivo, es decir la tarjeta SIM 14 o el terminal 16 o el servidor 12, comprende un bus de comunicación al que están conectados:

- una unidad de procesamiento -o microprocesador- denominada CPU (siglas de *Central Processing Unit* en terminología anglosajona);
- una o varias memorias no volátiles por ejemplo ROM (acrónimo de *Read Only Memory* en terminología anglosajona) que puede constituir un soporte en el sentido de la invención, es decir que puede comprender un programa informático que comprende instrucciones para la implementación de un procedimiento de acuerdo con una realización de la invención. Esta memoria no volátil también puede ser una memoria EEPROM (acrónimo de *Electrically Erasable Read Only Memory* en terminología anglosajona) o también una memoria Flash. En concreto, esta memoria no volátil del dispositivo esclavo 14 memoriza el valor actual de la o las variables dinámicas compartidas, por ejemplo del contador de secuencia en el caso SCP02 o SCP03 mencionado más adelante.
- una memoria viva o memoria caché o memoria volátil por ejemplo RAM (acrónimo de *Random Access Memory* en terminología anglosajona) que comprende registros adaptados al registro de las variables y parámetros creados y modificados durante la ejecución del programa mencionado anteriormente; durante la implementación de la invención, los códigos de instrucciones del programa almacenado en memoria muerta ROM se cargan en la memoria RAM en vista de ser ejecutados por la unidad de procesamiento CPU;
- una interfaz de comunicación adaptada para transmitir y para recibir datos, por ejemplo mediante una red de telecomunicaciones o una interfaz de lectura/escritura de un elemento seguro;
- una interfaz de entradas/salidas I/O (para *Input/Output* en terminología anglosajona), por ejemplo una pantalla, un teclado, un ratón u otro dispositivo de punteo tal como una pantalla táctil o un mando a distancia; esta interfaz I/O permite al usuario interactuar con el sistema durante la implementación del procedimiento mediante una interfaz gráfica.

El bus de comunicación permite la comunicación y la interoperabilidad entre los diferentes elementos incluidos en el equipo o conectados a este. La representación del bus no es limitante y, en concreto, la unidad de procesamiento es susceptible de comunicar instrucciones a cualquier elemento del equipo directamente o por medio de otro elemento de este equipo.

La figura 5 ilustra esquemáticamente los intercambios convencionales de acuerdo con el protocolo SCP03 definido en GlobalPlatform. Cabe destacar que en el modo predictivo mencionado anteriormente, el conjunto de los comandos enviados por el dispositivo maestro en esta figura puede agruparse en un lote de comandos inyectado por un dispositivo tercero, por ejemplo un terminal que incorpora un dispositivo esclavo de tipo tarjeta SIM.

Como se muestra esquemáticamente en la figura 1, estos intercambios pueden comprender una primera parte dedicada al establecimiento de un canal seguro, que implica un procedimiento de autentificación de tipo *"Challenge-Response*" bien conocido. Aunque no ilustrado, estos comandos con destino a una aplicación diana del dispositivo esclavo tiene lugar después de un comando inicial de selección de dicha aplicación diana, normalmente un comando SELECT ampliamente conocido. En una variante, no está previsto ningún comando SELECT, una aplicación del dispositivo esclavo 14 que puede seleccionarse por defecto (es decir en tanto que no se reciba ningún comando SELECT para otra aplicación) en la conexión a la red del dispositivo esclavo 14.

De este modo, el dispositivo maestro 12 genera, en la etapa 500, una interrogación de anfitrión (de forma aleatoria o seudo-aleatoria) que transmite al dispositivo esclavo 14 durante la etapa 502, con ayuda de un comando INITIALIZE UPDATE.

En la recepción de este comando, el dispositivo esclavo 14 actualiza, en la etapa 504, su contador de secuencias (en tres octetos) por incremento (el valor actual se incrementa en memoria no volátil), y a continuación calcula, en la etapa 506, una interrogación de tarjeta en función del contador de secuencias y de un identificador (AID) de una aplicación seleccionada en la tarjeta. La generación de la interrogación de tarjeta es generalmente seudo-aleatoria y predictiva (para el dispositivo maestro 12 en el caso de un canal predictivo).

En la etapa 507, el dispositivo esclavo 14 genera las claves de sesión (en concreto, descritas en la sección 6 de la norma GlobalPlatform) para el canal seguro, con ayuda, en concreto, del valor actual del contador de secuencias, interrogaciones de anfitrión y de tarjeta, así como uno o varios secretos estáticos compartidos entre los dispositivos maestro y esclavo.

En la etapa 508, el dispositivo esclavo 14 calcula un criptograma de autentificación para la tarjeta, en función de las interrogaciones de anfitrión y de tarjeta, así como de una constante (compartida). Por ejemplo la función de generación del criptograma puede ser el algoritmo S-MAC basado en una de las claves de sesión generadas.

El dispositivo esclavo 14 reenvía al dispositivo maestro 12 la interrogación de tarjeta, el criptograma de la tarjeta, así como el valor actual del contador de secuencias. Esta es la etapa 510.

En la recepción de estos datos, el dispositivo maestro 12 verifica (511) el estatus de la respuesta antes de continuar, indicado, por ejemplo, en el argumento opcional. Si el estatus es positivo (campo SW=9000), el dispositivo maestro 12 genera, en la etapa 512, las claves de sesión de forma similar al dispositivo esclavo 14.

A continuación, el dispositivo maestro 12 calcula y verifica el criptograma de tarjeta, aplicando el mismo algoritmo (compartido) que el del dispositivo esclavo 14. Esta es la etapa 514.

En la etapa 516, el dispositivo maestro 12 calcula un criptograma de autentificación para el anfitrión, en función de las claves de sesión (ellas mismas en función de las interrogaciones de anfitrión y de tarjeta, estando la interrogación de tarjeta además en función del contador de secuencias en modo predictivo), así como de una constante (compartida). Por ejemplo la función de generación del criptograma puede ser el algoritmo S-MAC basado en una de las claves de sesión generadas.

Este criptograma de anfitrión es enviado al dispositivo esclavo 14 por un comando EXTERNAL AUTHENTICATE, el cual está protegido en integridad y en autenticidad mediante la adición de un código de autentificación MAC calculado en el criptograma de anfitrión, con ayuda de una clave de sesión dedicada al MAC. El envío corresponde a la etapa 518.

En la etapa 520, el dispositivo esclavo 14 verifica el código MAC, a continuación el criptograma de anfitrión recibido para confirmar la creación del canal seguro (respuesta de acuse de recibo 522). El dispositivo maestro 12 verifica el estatus de la respuesta recibida.

En paralelo o después de la verificación positiva del estatus de la respuesta, en la etapa 524, el dispositivo maestro 12 genera y opcionalmente cifra (con ayuda de las claves de sesión) uno o varios comandos. Además, se calcula un código MAC de seguridad para cada uno de ellos y a continuación se les añade. El cifrado de los comandos y sus códigos MAC están general, pero no necesariamente, encadenados en el sentido de que cada uno de ellos depende del anterior (y, por lo tanto, inicialmente del MAC del comando EXTERNAL AUTHENTICATE).

Por supuesto, en el modo predictivo, estos comandos, incluyendo su cifrado y su código MAC, se predicen y, por lo tanto, se generan previamente al establecimiento del canal seguro.

La etapa 526 representa esquemáticamente el envío sucesivo de estos comandos y la recepción, por el dispositivo maestro 12, de sus respuestas. Por ejemplo, un comando puede ser el comando STORE DATA definido en GlobalPlatform para el almacenamiento de nuevos datos en memoria no volátil del dispositivo esclavo 14.

Para cada uno de estos comandos seguros, el dispositivo esclavo 14 verifica el código MAC (etapa 528), a continuación lo descifra (etapa 530) antes de ejecutarlo. En respuesta a esta ejecución, devuelve un acuse de recibo positivo (OK, el campo SW vale 9000) o un mensaje de error (NOK), que es procesado por el dispositivo maestro (etapa 532), incluso datos cifrados y protegidos por MAC en cuyo caso el dispositivo maestro 12 verifica su integridad (por el código MAC) a continuación procede a su descifrado (534).

La figura 5A ilustra una realización del modo predictivo para SCP03 que conduce al envío de un lote de comandos a un dispositivo tercero 16.

Cabe destacar que en el modo predictivo, el dispositivo maestro 12 no tiene en cuenta los elementos de la respuesta 510. En efecto, retiene localmente una imagen de las variables y valores retenidos por el dispositivo esclavo 14, permitiendo calcular localmente la interrogación de tarjeta, el criptograma de tarjeta en conocimiento del nuevo valor del contador de secuencias.

En el ejemplo representado, estos diferentes procesamientos que el dispositivo maestro 12 realiza de forma predictiva se efectúan antes de transmitir el lote (batch) de comandos: recuperación del valor público del contador de secuencias compartido con la aplicación diana, generación seudo-aleatoria de las interrogaciones de anfitrión y de tarjeta, generación de las claves de sesión, generación de los comandos INITIALIZE UPDATE y EXTERNAL AUTHENTICATE, con su código MAC, generación de otros comandos seguros para formar el lote de comandos.

El dispositivo tercero 16 desempeña entonces un papel intermedio simple: este último recibe, del dispositivo maestro 12, únicamente el lote de comandos a inyectar por su cuenta frente al dispositivo esclavo 14. Puede verificar el estatus de respuesta recibida para determinar si la ejecución del lote se debe abortar. Como variante, se puede contentar con retransmitir el conjunto de las respuestas a los comandos que transmite sucesivamente.

Al final de la ejecución del lote de comandos, el dispositivo tercero 16 puede enviar un mensaje de estatus general al dispositivo maestro 12, para indicarle, por ejemplo, el número de comandos enviados y si la ejecución del lote completo se ha efectuado o no con éxito. Como variante, no se envía ningún mensaje de estatus general, pudiendo solicitarse la verificación del estado (éxito o fracaso) durante un comando siguiente.

En estas realizaciones de SCP03, los comandos STORE DATA son seguros, es decir está dotados de un código MAC encadenado. En una variante degradada, el protocolo SCP03 se puede utilizar solamente para el proceso de autentificación (comandos INITIALIZE UPDATE y EXTERNAL AUTHENTICATE) de modo que los comandos STORE DATA generados en la etapa 524 no ni están cifrados ni son seguros gracias a un código MAC. De este modo, solo el comando EXTERNAL AUTHENTICATE es seguro gracias a MAC. Este indica además, mediante el parámetro P1 (definido en Global Platform), que los comandos posteriores no son seguros.

En realizaciones sin proceso de autentificación, los comandos INITIALIZE UPDATE y EXTERNAL AUTHENTICATE no están previstos, y el lote de comandos comienza directamente por comandos seguros, de tipo STORE DATA (véase la figura 1A donde el comando SELECT APPLICATION es opcional ya que puede realizarse de forma descorrelacionada, antes de la transmisión de los comandos seguros STORE DATA). Esto se hace posible gracias a la predicción de las variables dinámicas que permiten a los dispositivos maestro y esclavo de disponer de las mismas claves de sesión sin ni siquiera intercambiar el mensaje. La información de securización de los comandos es por ejemplo notificada en el parámetro P1 del primer comando STORE DATA (identificable gracias al parámetro "*block number*" igual a 0 para este comando) recibido de este modo, permitiendo activar las etapas 504-508 en el dispositivo esclavo 14. Véase en este sentido la figura 5B. En la configuración de esta figura, la interrogación de anfitrión puede ser (seudo-aleatoria) predecible, permitiendo al dispositivo esclavo 14 obtenerlo mediante un proceso interno. Como variante, se puede obtener durante una operación anterior (no ilustrada) con el dispositivo maestro 12.

El protocolo SCP02 también definido en GlobalPlatform se distingue del protocolo SCP03 de la figura 5 en que el contador de secuencias no se incrementa en la recepción del comando INITIALIZE UPDATE, sino en la recepción del comando EXTERNAL AUTHENTICATE.

La presente invención se inscribe en el marco de un modo predictivo de un protocolo de canal seguro, por ejemplo del protocolo SCP02 o SCP03. En las técnicas conocidas, un lote de comandos pre-generados solamente puede ser inyectado una sola vez, conduciendo a numerosos sobrecostes, principalmente para el dispositivo maestro 12.

Para permitir una reinyección de este lote de comandos, la invención prevé utilizar un contador de reinyecciones o de ensayos. El número de reinyecciones es preferentemente razonable, en concreto para impedir que una persona malintencionada de pruebe hasta el infinito una secuencia (lote) de comandos. En este caso, este número está limitado por un valor umbral o "límite de reinyección", como se ilustra más adelante.

El contador de ensayos permite temporizar el incremento concreto del contador de secuencias, es decir su actualización efectiva en memoria no volátil. De este modo, durante la utilización del contador de ensayos, el valor actual del contador de secuencias que sirve de referencia para el establecimiento de un canal seguro permanece inalterado en memoria no volátil, lo que permite reinyectar el mismo lote de comandos seguro con las claves de sesión predichas de este canal seguro. Solamente cuando este valor en memoria no volátil es actualizado (modificado), el lote de comandos se vuelve obsoleto, necesitando solicitar al dispositivo maestro para la obtención de un lote actualizado para corresponder a las claves de sesión predichas a partir del nuevo valor actual del contador de secuencias.

Como se definió más arriba, un dispositivo esclavo 14 de acuerdo con la invención incrementa un contador de ensayos en la recepción de un comando, llamado primer comando (en concreto un comando de inicialización INITIALIZE UPDATE en SCP02 o SCP03), que indica el establecimiento de un canal de comunicación seguro; y sobreescribe el valor actual de la variable dinámica en memoria no volátil con el valor de actualización calculado cuando el contador de ensayos alcanza un valor umbral, es decir registra un valor de actualización calculado, en memoria no volátil como nuevo valor actual de la variable dinámica (por ejemplo el contador de secuencias utilizado para securizar el canal de comunicación) cuando el contador de ensayos alcanza un valor umbral.

Gracias a la utilización de este contador de ensayos que retarda la validación (commitment) del valor de actualización como nuevo valor actual de la variable dinámica en memoria no volátil, los mismos comandos (seguros gracias al valor actual de la variable dinámica) pueden ser reinyectados. De este modo, el dispositivo tercero 16 puede reinyectar el lote de comandos en tanto que el dispositivo esclavo no haya actualizado el valor actual de la variable dinámica en memoria no volátil, es decir en tanto que no reciba, del dispositivo esclavo 14, una indicación que indica una actualización del valor actual de la variable dinámica (el contador de secuencias) por el dispositivo esclavo. Como se verá a continuación, conviene por ejemplo para el dispositivo tercero 16 ser capaz de analizar los estatus de respuesta del dispositivo esclavo 14 para detectar en ellos ciertos estatus correspondientes a situaciones en las que el dispositivo esclavo ha actualizado concretamente el valor actual del contador de secuencias ya que una reinyección del lote de comandos no es oportuna.

La figura 6 ilustra, en el mismo esquema que la figura 3, la posibilidad de reinyección de un lote de comandos por el dispositivo tercero 16, representado en esta figura por un bucle (flecha en negrita) en tanto que el límite del número de ensayos contabilizado por el dispositivo esclavo 14 no se haya alcanzado (en cuyo caso una indicación es transmitida al dispositivo tercero 16) o que se realice un acontecimiento de salida (que indica que una reinyección del lote de comandos no es útil).

En esta configuración, la aparición de un acontecimiento inesperado (timer out, reset, perturbación del canal de comunicación, etc.) no conduce directamente a solicitar el dispositivo maestro 12 para la generación de un nuevo lote de comandos. En el presente documento, el lote de comandos para el que se produce el acontecimiento inesperado se puede reinyectar si no se trata un acontecimiento llamado "de salida". Si este acontecimiento es puntual, la nueva reinyección del lote de comandos debería conducir a que sea inyectado en su totalidad. De este modo, se ve que se evita solicitar al dispositivo maestro, cuando aparecen acontecimientos incontrolados.

En el enfoque propuesto en el presente documento, el dispositivo tercero 16 dispone, por lo tanto, de la capacidad de reinyectar, hasta cierto límite, un lote de comandos cuando recibe, de parte del dispositivo esclavo 14, una notificación de error en los comandos reinyectados, es decir un acontecimiento de salida del bucle de reinyección.

Cuando dicho límite del número de reinyecciones se alcanza, el dispositivo esclavo 14 lo indica (explícita o implícitamente) de modo que el dispositivo tercero 16 no reinyecta el lote actual de comandos, sino que solicita al dispositivo maestro 12 para generar un nuevo lote de comandos actualizado (si el lote actual no se ha podido ejecutar normalmente). Los comandos del nuevo lote pueden ser similares a los del lote actual; siendo la diferencia que el cifrado y el cálculo de los códigos MAC para los comandos seguros se basan, en adelante, en un nuevo valor del contador de secuencias, en SCP02 o SCP03.

Por supuesto, cuando la ejecución del lote completo de comandos se ha desarrollado con éxito, está previsto que la reinyección no sea posible, para evitar cualquier análisis por una persona malintencionada. Un estatus que indica la ejecución con éxito del lote de comandos constituye de este modo un acontecimiento de salida del bucle de reinyección. Otros acontecimientos de salida están previstos a continuación para evitar una reinyección cuando esta se muestra inútil. Este es por ejemplo el caso cuando un comando seguro, íntegro y auténtico (es decir con un código MAC válido) conduce a un error o excepción de ejecución.

En estos casos, el dispositivo esclavo 14 solo tiene que actualizar el contador de secuencias (la variable dinámica) en memoria no volátil, haciendo obsoleto el lote actual de comandos. El dispositivo tercero 16 es notificado de esta situación gracias a los estatus de respuesta que recibe del dispositivo esclavo 14.

La figura 7 ilustra, en forma de ordinograma, etapas generales de una realización de la invención del lado del dispositivo esclavo 14.

El procedimiento de la figura comienza con una fase de inicialización (etapas 700 a 704) que se puede producir durante la personalización del dispositivo esclavo, normalmente una tarjeta con chip o un elemento seguro. Durante esta fase, un contador de ensayos se crea en memoria no volátil, y se inicializa a un valor por defecto, por ejemplo 10, en la etapa 700. A continuación, en la etapa 702, un valor umbral límite del número de ensayos (y, por lo tanto, de reinyecciones) se define y se almacena en memoria no volátil, por ejemplo 10. En esta realización en la que el contador de ensayos se inicializa al número de ensayos máximo autorizados, este contador de ensayos decrecerá para alcanzar, al final, el valor 0, lo que es fácilmente detectable ya que, en ese momento, el paso del contador de 0 a -1 necesita modificar todos los bits (por ejemplo para un contador en 32 bits: 00000000 -> FFFFFFFF si en cuatro octetos). Por supuesto, otras realizaciones pueden inicializar el contador a 0 e incrementarlo hasta el valor umbral límite igual a 10.

En la etapa 704, la o las variables dinámicas, normalmente el contador de secuencias, se inicializan. Por ejemplo, el contador de secuencias se inicializa a 0.

Puede ser durante esta etapa 704 que la variable dinámica y otros parámetros eventuales (por ejemplo un secreto) son compartidos entre el dispositivo esclavo 14 y el dispositivo maestro 12 para permitir la implementación del modo predictivo de los protocolos de canal seguro.

Tras su personalización, el dispositivo esclavo 14 se pone en servicio donde esté en espera del establecimiento de un nuevo canal seguro. Se trata, por ejemplo, de la recepción de un comando INITIALIZE UPDATE en el caso de SCP02 o SCP03 ilustrado en la figura 5A. Se trata de la recepción del primer comando (por ejemplo STORE DATA) seguro en el ejemplo de la figura 5B. Esta es la etapa 706.

Cuando dicho comando es recibido, las etapas 708 y 710 consisten en verificar que el contador de ensayos y el número máximo de ensayos no están ambos dos en 0, en cuyo caso no se puede establecer ningún canal seguro (etapa 712) y el procedimiento se termina.

En el caso general, el número máximo de ensayos no es nulo. De este modo, en tanto que el contador de ensayos (en un modo en el que ha decrecido) no es nulo (salida "no" de la prueba 708), el contador de ensayos decrece en la etapa 714, seguida de la etapa 716 que consiste en actualizar las variables dinámicas necesarias para el canal seguro, es decir en calcular un valor de actualización de la o las variables dinámicas (en concreto el contador de secuencia) a partir de un valor actual de la o las variables dinámicas almacenadas en memoria no volátil. El o los valores de actualización de estas variables dinámicas se almacenan entonces en memoria volátil, garantizando que sus valores actuales permanecen inalterados, y por lo tanto adaptados a una reinyección del mismo lote de comandos.

El dispositivo esclavo 14 procesa entonces el comando recibido de forma convencional, así como los comandos seguros posteriores tales como los recibidos del dispositivo tercero 16. Estas operaciones en el canal seguro se representan esquemáticamente mediante la etapa 718, y se han descrito anteriormente en referencia a las figuras 5 a 5B.

La ejecución de los comandos seguros del lote actual conduce a dos situaciones principales, y una situación tratada de forma particular en este contexto.

Como se representa en la figura en la etapa 720, se determina si la ejecución del lote de comandos seguros se ha desarrollado correctamente (salida "normal" en la prueba 720) o si se ha producido un acontecimiento imprevisto (salida "inesperada" en la prueba 720).

Dicho acontecimiento puede corresponder a un error de transmisión de un comando seguro del lote (error detectable a través del código MAC que acompaña al comando). En este caso, el canal seguro se cierra por el envío de un mensaje de error al dispositivo tercero 16 (etapa 722), antes de que el dispositivo esclavo 14 se vuelva a poner en espera de un nuevo comando para un nuevo canal seguro.

La ejecución "normal" del lote de comandos seguros se produce cuando el dispositivo esclavo 14 ha recibido el conjunto de los comandos seguros, con códigos de autentificación MAC válidos. El último comando puede comprender, en concreto, una indicación de fin de comandos seguros (un indicador binario puede estar previsto a tal efecto en el comando; el campo P1 puede indicar que los comandos siguientes ya no son seguros), permitiendo al dispositivo esclavo 14 detectar el final de los comandos seguros. Cualquier otro medio para que el dispositivo esclavo 14 identifique el último comando seguro puede estar previsto.

La ejecución normal del conjunto de los comandos seguros constituye un acontecimiento de salida del bucle de reinyección, que conduce al cierre del canal seguro. En este caso, el dispositivo esclavo 14 registra el valor de actualización calculado (que está en memoria volátil en ese momento, debido a la etapa 716), en memoria no volátil como nuevo valor actual de la variable dinámica, normalmente el contador de secuencias. El antiguo valor actual se sobrescribe con el valor de actualización calculado. Se trata de la etapa 724.

Generalmente, el conjunto de los comandos del lote (después de EXTERNAL AUTHENICATE) son seguros. De este modo, la ejecución con éxito del conjunto de los comandos seguros corresponde a la ejecución con éxito del lote de comandos.

Otro ejemplo corresponde al caso en el que el nivel de seguridad definido en el parámetro P1 del comando EXTERNAL AUTHENTICATE indica "*No Secure Messaging Expected*", es decir un valor "0" (véase la sección 7.1.2 de GlobalPlatform). En este caso, solo el comando EXTERNAL AUTHENTICATE es seguro (los siguientes no lo son). La invención permite, por lo tanto, la reinyección del lote de comandos únicamente cuando se realiza un acontecimiento inesperado para este comando EXTERNAL AUTHENICATE, que se limita de hecho a la reinyección del único comando protegido, a saber EXTERNAL AUTHENICATE.

Tras la validación (commitment) del nuevo valor actual del contador de secuencias (etapa 724), el contador de ensayos se reinicializa al valor por defecto, "10" en el ejemplo anterior, durante una etapa 726.

A continuación, en la etapa 722, el dispositivo esclavo 14 envía, al dispositivo tercero 16 y en respuesta al último comando seguro de los comandos seguros del lote, un mensaje que comprende una indicación de que la ejecución de todos los comandos seguros del lote se ha realizado con éxito. Este es por ejemplo un mensaje de acuse de recibo en respuesta a este último comando del lote actual. Como se ha indicado más arriba, el último comando seguro se puede identificar como tal con ayuda de un indicador, o si no derivarse de la aparición del comando seguro con respecto a un número previsto de comandos seguros.

La tercera situación corresponde a un comando seguro del lote, cuyo código de autentificación de mensaje se verifica con éxito (por lo tanto, el comando seguro es íntegro y auténtico) y que conduce a un error o excepción de ejecución de este comando. Es decir, que la ejecución del comando, una vez descifrado, fracasa. En otros términos, este comando tal como es definido y cifrado por el dispositivo maestro 12 es erróneo, contaminando el lote actual de comandos que es, por lo tanto, también erróneo. Este es, por ejemplo, el caso cuando el comando indica una función inexistente en el dispositivo esclavo. Como alternativa, puede tratarse de un comando mal formado, con parámetros erróneos. En este caso, no resulta oportuno proceder a reinyecciones de este lote. Para evitar estas reinyecciones, el dispositivo esclavo 14 puede forzar al dispositivo tercero 16 a solicitar de nuevo al dispositivo maestro 12, actualizando, en memoria no volátil, el valor actual del contador de secuencias con el valor de actualización actualmente almacenado en memoria volátil y/o reenviando un estatus de error que el dispositivo tercero 16 es capaz de interpretar como acontecimiento de salida del bucle de reinyección.

Dicha situación también puede ser detectada durante la etapa 720, que conduce a la etapa 724 para la actualización efectiva del contador de secuencias en memoria no volátil, a continuación a la reinicialización del contador de ensayos en la etapa 726 y finalmente en la etapa 722 donde un mensaje es enviado, al dispositivo tercero en respuesta a este comando que conduce a un error o excepción de ejecución, comprendiendo el mensaje una indicación del error o excepción de ejecución a pesar de un código de autentificación de mensaje verificado con éxito.

Para evitar una posible desincronización del contador de secuencias con el contador de ensayos (el contador de ensayos se reinicializa cuando el contador de secuencias en memoria no volátil se incrementa), se conviene que la etapa que consiste en sobreescribir el valor actual en memoria no volátil y la etapa que consiste en reinicializar el contador de ensayos se realizan en una etapa atómica, es decir no divisible, para impedir que cualquier persona malintencionada de distorsione el procedimiento en este momento.

Otros acontecimientos de salida que conducen a la actualización efectiva (commitment) del contador de secuencias en memoria no volátil pueden estar previstos.

Otro ejemplo de acontecimiento de salida es el caso de un código MAC del comando EXTERNAL AUTHENTICATE que es detectado como válido, mientras que el criptograma de anfitrión que contiene es detectado como erróneo.

De vuelta a las etapas 708 y 710, si el contador de ensayos indica el valor "0", es decir que el contador de ensayos alcanza un valor umbral y el número máximo de reinyecciones se alcanza, el dispositivo esclavo 14 calcula el valor de actualización del contador de secuencias a partir de su valor actual almacenado en memoria no volátil; a continuación registra este valor de actualización calculado en memoria no volátil como nuevo valor actual del contador de secuencias (que, por lo tanto, se sobreescribe). Se trata de la etapa 728.

El procedimiento prosigue entonces en la etapa 730 similar a la etapa 726 para reinicializar el contador de ensayos; a continuación se prueba de nuevo el valor del contador de ensayos en la etapa 708. Este bucle garantiza el commitment del nuevo valor actual del contador de secuencias cuando se ha alcanzado el número máximo de reinyecciones.

Para evitar una posible desincronización del contador de secuencias con el contador de ensayos, la etapa que consiste en sobreescribir el valor actual en memoria no volátil y la etapa que consiste en reinicializar el contador de ensayos se realizan antes de la etapa de generación de la clave de sesión a partir del valor de actualización calculado. Esta disposición se aplica cuando el alcance del número máximo de reinyecciones es detectado con ocasión de un nuevo primer comando que indica el establecimiento de un canal seguro, por ejemplo un comando INITIALIZE UPDATE ("sí" en la prueba 708). De manera general, estas etapas se realizan en una etapa atómica, es decir no divisible, para impedir que cualquier persona malintencionada de distorsione el procedimiento en este momento.

El retorno en bucle a la etapa 708 permite procesar el lote actual (todos los comandos seguros del lote) si este ya está en línea con el nuevo valor actual del contador de secuencias. En efecto, según el nivel de información que devuelve el dispositivo esclavo 14 al dispositivo tercero 16, es posible para este último anticipar la llegada al número límite de ensayos, de modo que se puede anticipar la obtención de un lote de comandos actualizado. Cualquiera que sea, si el lote actual aún no está actualizado (y, por lo tanto, es obsoleto), lo estará en la próxima iteración ya que el dispositivo tercero 16 será informado del nuevo valor actual del contador de secuencias durante la recepción del mensaje 510 (véase la figura 5A) en el transcurso de la etapa 718 siguiente.

Cabe destacar que en la implementación de la figura 5B, este intercambio 510 no ha tenido lugar. Como el contador de secuencias es público, un mecanismo de sincronización puede prever por iniciativa del dispositivo tercero o del dispositivo esclavo, comunicar al dispositivo tercero el valor actual del contador de secuencias. Por ejemplo, el valor actual puede ser devuelto automáticamente en respuesta a cada primer comando de un lote, o ser solicitado por un comando GET DATA del dispositivo tercero.

En el ejemplo de la figura 7, se compara el valor del contador de ensayos con el número máximo de reinyecciones posibles antes de decrecer (o incrementar) dicho contador de ensayos. Una variante puede consistir en invertir el orden de estas dos etapas.

La figura 8 ilustra, en forma de ordinograma, etapas generales de una realización de la invención del lado del dispositivo tercero 16.

Después de una inicialización, el dispositivo tercero 16 está en espera de la recepción de uno o varios comandos por parte del dispositivo maestro 12. Esta es la etapa 800.

En la recepción, se determina, en la etapa 802, si se trata de un lote (script o batch) de comandos pre-generado en modo predictivo. Si este no es el caso, los comandos son procesados de forma convencional en la etapa 804, antes de volver a la espera de un nuevo comando (etapa 800).

Si no, un índice "i" se inicializa a 0 en la etapa 806, índice que permite seleccionar y procesar de forma secuencial cada uno de los comandos del lote recibido.

En la etapa 808, el dispositivo tercero 16 envía el comando "i" al dispositivo esclavo 14.

Si no se recibe ninguna respuesta (prueba 810), el dispositivo tercero 16 considera que se ha producido un acontecimiento imprevisto, y procede a la reinyección (flecha 811) del lote actual de comandos volviendo a la etapa 806.

Si se recibe una respuesta, el dispositivo tercero 16 determina, en la etapa 812, si se ha producido un acontecimiento imprevisto, por ejemplo porque la respuesta indica que el código MAC del comando es erróneo (se ha producido un error de transmisión). Si es el caso, el dispositivo tercero 16 procede a la reinyección (811) del lote actual de comandos volviendo a la etapa 806.

Si el comando ha sido recibido correctamente por el dispositivo esclavo 14, el dispositivo tercero 16 determina, en la etapa 814, si la respuesta contiene un valor del contador de secuencias del dispositivo esclavo 14. Dicha información está presente en la respuesta al primer comando INITIALIZE UPDATE como se ha descrito anteriormente en relación con la figura 5.

Si el valor del contador de secuencias del dispositivo esclavo 14 está indicado en la respuesta, se compara, en la etapa 816, con un valor local que almacena el dispositivo tercero 16.

Si los dos valores son idénticos, esto significa que el dispositivo esclavo 14 no ha actualizado (en memoria no volátil) su contador de secuencias después de la última iteración, es decir que el procedimiento se sitúa en plena reinyección de un mismo lote de comandos. En este caso, se ejecuta el comando si fuera necesario (etapa 817) y a continuación se pasa al comando siguiente de la inyección incrementando el índice "i" en la etapa 818, que retorna en bucle a la etapa 808.
Si, por el contrario, los dos valores son diferentes, esto significa que el dispositivo esclavo 14 ha actualizado (en memoria no volátil) su contador de secuencias después de la última iteración (bien por alcance del número máximo de reinyecciones, bien porque el último lote de comandos ha sido enteramente procesado, o bien finalmente porque el último lote de comandos era intrínsecamente erróneo). En este caso, dicho valor local está actualizado, en la etapa 820, para asumir el valor de contador de secuencias recibido. Continuando en la etapa 820, el dispositivo tercero 16 toma contacto con el dispositivo maestro 12 opcionalmente para solicitarle un lote de comandos actualizado con el nuevo valor actual del contador de secuencias.

De vuelta a la prueba 814, si la respuesta recibida no contiene valor de contador de secuencias (por ejemplo si no se trata de la respuesta al primer comando INITIALIZE UPDATE en el caso de la figura 5A), el dispositivo tercero 16 determina, en la etapa 821, si la respuesta recibida indica un acontecimiento de salida del bucle de reinyección.

En el ejemplo de la figura, se proponen dos pruebas. Por supuesto, se pueden implementar pruebas suplementarias para detectar otros acontecimientos de salida.

En la etapa 822, el dispositivo tercero 16 determina si la respuesta recibida indica un error intrínseco de comando (el estatus de respuesta asume un código predefinido), en cuyo caso el valor local de contador de secuencias se actualiza, en la etapa 820, de forma similar a una actualización por el dispositivo esclavo 14 (generalmente por simple incremento). En efecto, esta operación permite reflejar la actualización efectiva 724 del contador de secuencias operada por el dispositivo esclavo 14. De este modo, se garantiza una sincronización del contador de secuencias entre el dispositivo tercero 16 y el dispositivo esclavo 14.

Asimismo, si la respuesta recibida indica que el conjunto de los comandos seguros se ha procesado completamente con éxito (prueba 824 - por ejemplo estatus OK en respuesta al último comando seguro), el valor local del contador de secuencias también se actualiza, en la etapa 820, de forma similar a una actualización por el dispositivo esclavo 14, para reflejar los cambios en este último dispositivo.

De lo contrario, no se ha producido ningún acontecimiento inesperado o de salida, que conduce a continuar el procesamiento de los otros comandos mediante la etapa 818.

La figura 9 ilustra, retomando el esquema de la figura 5A, intercambios de una realización de la invención que se apoya en el protocolo SCP03. Las etapas 5xx permanecen inalteradas con respecto a la figura 5A, mostrando que la presente invención es compatible con los protocolos de canal seguro existentes.

Cabe destacar que las explicaciones a continuación se aplican también al modo predictivo con la secuenciación de la figura 5, o al modo predictivo sin autentificación de la figura 5B. Además, se aplican a otros protocolos donde uno o varios comandos seguros son transmitidos en un canal seguro.

Para sincronizarse de forma apropiada con el dispositivo esclavo 14, el dispositivo maestro 12 es capaz de recuperar el valor actual del contador de secuencias (etapa 499), desde el dispositivo esclavo 14, mediante el dispositivo tercero 16. Por ejemplo, este valor actual puede ser solicitado con ayuda de un comando GET DATA (con un indicador 9F70), ilustrado por los intercambios 900 en la figura.

El contador de ensayos decrece (etapa 714) inmediatamente después de recepción del primer comando que indica el establecimiento de un canal seguro, en el presente documento el comando INITIALIZE UPDATE, como se muestra mediante la etapa 902.

Cuando este contador de ensayos alcanza el número máximo de reinyecciones posibles, el contador de secuencias se actualiza en memoria no volátil (etapa 728) y el contador de ensayos se reinicializa (etapa 730). Estas dos etapas, que corresponden a la etapa 903, se ilustran mediante las letras "MAJ" en la figura.

La etapa 904 se distingue de la etapa 504 en que el valor de actualización del contador de secuencias se memoriza en memoria volátil (y no en memoria no volátil como en la etapa 504), para permitir una eventual reinyección del lote actual de comandos.

Las etapas siguientes 506-511, 518, 520 son idénticas a las de la figura 5A, habiendo el dispositivo esclavo 14 entonces verificado el MAC y el criptograma contenidos en el comando EXTERNAL AUTHENTICATE, es decir todo el primer comando seguro por un código MAC.

En la etapa 906 que termina el procesamiento del comando EXTERNAL AUTHENTICATE, el dispositivo esclavo 14 determina si se ha producido un acontecimiento de salida del bucle de reinyección. Estos acontecimientos están, en concreto, predefinidos.

En caso de acontecimiento de salida del bucle de reinyección, el dispositivo esclavo 14 actualiza el contador de secuencias en memoria no volátil (etapa 724), y reinicializa el contador de ensayos (etapa 730) ["MAJ" de la etapa 906].
Un acontecimiento de salida del bucle de reinyección es por ejemplo una verificación negativa del criptograma de anfitrión del comando EXTERNAL AUTHENTICATE.

Opcionalmente (en concreto si el comando EXTERNAL AUTHENTICATE indica que los comandos siguientes son seguros), un acontecimiento de salida del bucle de reinyección puede ser la detección de un MAC erróneo del comando EXTERNAL AUTHENTICATE. No obstante, generalmente, un MAC erróneo (que conduce al cierre del canal seguro), y que, por lo tanto, incluye el del comando EXTERNAL AUTHENTICATE, será considerado representativo de un acontecimiento inesperado, el cual permite una reinyección del lote de comandos.

La respuesta 522 negativa (NOK) puede comprender entonces (en el parámetro opcional SW) un código de error que es diferente si se trata de un acontecimiento de salida que conduce a la actualización del valor actual del contador de secuencias o de un acontecimiento inesperado que permite la reinyección. Estos códigos de error son conocidos por el dispositivo tercero 16.

A continuación, cada uno de los comandos seguros posteriores del lote de comandos es procesado (526), procesamiento que implica, para el dispositivo esclavo 14 verificar el código MAC (528), descifrar y ejecutar el comando (530) y determinar si se ha producido un acontecimiento de salida del bucle de reinyección, en cuyo caso se efectúan una actualización del valor actual del contador de secuencias en memoria no volátil y una reinicialización del contador de ensayos ["MAJ" de la etapa 908].

Un código MAC erróneo conduce al cierre del canal de comunicación. Se considera como un acontecimiento inesperado que permite la reinyección, ya que este error de código MAC puede resultar simplemente de un error de transmisión, y no de un error en el lote de comandos a transmitir.

Un acontecimiento de salida del bucle de reinyección es, por ejemplo, un error de ejecución de un comando seguro que dispone de un código MAC válido. En este caso en efecto, el comando seguro es intrínsecamente erróneo, no permitiendo una reinyección resolver la fuente del error (por ejemplo un parámetro erróneo o faltante en el comando). La respuesta negativa reenviada (NOK) puede comprender entonces (en el parámetro opcional SW) un código de error que es diferente si se trata de un acontecimiento de salida que conduce a la actualización del valor actual del contador de secuencias o de un acontecimiento inesperado que permite la reinyección. Estos códigos de error son conocidos por el dispositivo tercero 16.

Otro acontecimiento de salida del bucle de reinyección es simplemente la correcta ejecución del último comando seguro del lote actual. Un dicho "último comando" puede indicarse directamente con ayuda de un indicador apropiado, o puede ser implícito, por ejemplo cuando un comando inicial indica el número de comandos seguros venideros. De forma correspondiente, el dispositivo tercero 16 recibe los estatus de respuesta del dispositivo esclavo 14, que le permiten determinar si un acontecimiento inesperado o un acontecimiento de salida del bucle de reinyección se ha producido en el dispositivo esclavo 14.

La detección de un acontecimiento de salida conduce a la actualización, por el dispositivo tercero 16, de su contador local de secuencias ("MAJloc" en la figura).

En concreto, gracia a las respuestas reenviadas por el dispositivo esclavo 14, el dispositivo tercero 16 detecta, cuando recibe un valor de contador de secuencias diferente del valor localmente almacenado (véase 912), un código de error del criptograma de anfitrión (véase 914), un mensaje que indica un error de ejecución de un comando en el código MAC válido (véase 916) o un mensaje de éxito para el último comando seguro del lote (véase 916 también).

La figura 9A ilustra las mismas explicaciones en ausencia de procedimiento de autentificación (figura 5B), es decir cuando el primer comando que indica el establecimiento de un canal seguro es, a su vez, un comando seguro (figura 1 B).

La figura 10 ilustra el efecto de temporización de la actualización efectiva en memoria no volátil del contador de secuencias. El ejemplo de la figura se apoya en un número máximo de reinyecciones igual a 4, e ilustra la ejecución de cinco lotes de comandos seguros.

El contador de ensayos, inicialmente en 4, decrece a 3 durante el primer ensayo de inyección para el primer lote de comandos, durante el cual un valor de actualización del contador de secuencias se almacena en memoria volátil. Esta primera inyección (#1) es un fracaso (sea cual sea la razón), conduciendo a la reinyección del lote actual.

El contador de ensayos decrece entonces a 2, pero esta segunda inyección (#2) del lote es también un fracaso (el mismo valor de actualización del contador de secuencias se almacena en memoria volátil). El contador de ensayos decrece entonces a 1, pero esta tercera inyección (#3) del lote es también un fracaso. El contador de ensayos decrece entonces a 0, pero esta cuarta inyección (#4) del lote es también un fracaso.

En la detección del contador de ensayos a 0, el contador de secuencias se actualiza en memoria no volátil, haciendo al lote actual de comandos obsoleto. El dispositivo tercero 16 obtiene un nuevo lote de comandos (opcionalmente los mismos comandos, cifrados de forma diferente). El contador de ensayos se reinicializa también a 4.

El contador de ensayos decrece entonces a 3, peo la primera inyección (#5) del nuevo lote es un fracaso (un nuevo valor de actualización del contador de secuencias se almacena en memoria volátil). El contador de ensayos decrece entonces a 2. La segunda inyección (#6) del nuevo lote es en el presente documento un éxito, conduciendo a actualizar el contador de secuencias en memoria no volátil. El dispositivo tercero 16 obtiene un tercer y nuevo lote de comandos. El contador de ensayos se reinicializa a 4.

El contador de ensayos decrece entonces a 3. La primera inyección (#7) del nuevo lote es este contexto un éxito, conduciendo a actualizar el contador de secuencias en memoria no volátil. El dispositivo tercero 16 obtiene un cuarto y nuevo lote de comandos. El contador de ensayos se reinicializa a 4.

El contador de ensayos decrece entonces a 3, pero la primera inyección (#8) del nuevo lote es un fracaso. El contador de ensayos decrece entonces a 2, pero la segunda inyección (#9) del nuevo lote es un fracaso. El contador de ensayos decrece entonces a 1, pero la tercera inyección (#10) del nuevo lote es un fracaso. El contador de ensayos decrece entonces a 0, pero la cuarta y última inyección (#11) del nuevo lote es también un fracaso.

En la detección del contador de ensayos a 0, el contador de secuencias se actualiza en memoria no volátil, haciendo al nuevo lote actual de comandos obsoleto. El dispositivo tercero 16 obtiene un nuevo lote de comandos (opcionalmente los mismos comandos, cifrados de forma diferente). El contador de ensayos se reinicializa a 4.

El procedimiento continúa de la misma forma, con el decrecimiento del contador de ensayos a 3 y con la primera inyección (#12) del nuevo lote.

Los ejemplos anteriores son solamente realizaciones de la invención que no se limita a ellas.

En particular, los ejemplos anteriormente muestran una reinyección completa de un lote de comandos, es decir que el primer comando de la reinyección es el primer comando que indica el establecimiento del canal seguro, es decir INITIALIZE UPDATE en determinados modos de SCP02 o SCP03. Ahora bien, la invención se aplica también cuando el primer comando es un comando seguro que indica el establecimiento de un canal seguro (véase la figura 9A). De este modo, como variante, se puede prever efectuar reinyecciones a partir de un punto de restauración previsto en el script de comandos, es decir a partir de un comando seguro predefinido (varios comandos seguros que forma puntos de restauración pueden estar previstos en el lote).

Por ejemplo, determinados comandos seguros pueden estar indicados, por el dispositivo maestro, como comandos "punto de restauración", por ejemplo todos los 10 comandos. Esta indicación puede estar prevista en forma de indicador binario en la cabecera no cifrada de los comandos, para que el dispositivo tercero 16 pueda ser capaz de interpretarla.

Esta indicación es procesada de forma similar por el dispositivo tercero 16 y el dispositivo esclavo 14, a saber por la memorización de un contexto actual cuando este comando "marcado" es ejecutado (el dispositivo tercero recibe un acuse de recibo). El contexto actual contiene, en concreto, el número "i" del comando "punto de restauración", las claves de sesión, una copia del último comando y de la última respuesta (que incluye sus códigos MAC y uno o varios contadores CBC) para permitir un descifrado/cifrado de los comandos/respuestas siguientes.

De este modo, cuando se produce un acontecimiento inesperado (corte de alimentación, código MAC erróneo), los dispositivos efectúan una nueva inyección del lote de comandos a partir del último comando "punto de restauración" memorizado en el contexto guardado.

Cabe destacar que si el dispositivo esclavo no está al corriente de este acontecimiento inesperado (por ejemplo no recepción de un comando), el próximo comando que recibe del dispositivo tercero es el comando "punto de restauración", que conduce, por lo tanto, a una verificación errónea del código MAC de este comando. Los dos dispositivos vuelven a empezar entonces en una nueva reinyección del lote a partir de este comando "punto de restauración", garantizando una resincronización eficaz.

REIVINDICACIONES

1. Procedimiento de comunicación que comprende, a nivel de un dispositivo esclavo (14), las etapas siguientes:

recibir (502), de un dispositivo tercero (16), un comando, llamado primer comando, de un lote de comandos, indicando dicho primer comando el establecimiento de un canal de comunicación seguro gracias a, al menos, una clave de sesión;
calcular (904) un valor de actualización de una variable dinámica a partir de un valor actual de la variable dinámica almacenado en memoria no volátil; y
generar (506) la clave de sesión a partir del valor de actualización calculado;
a continuación recibir (518, 526), del dispositivo tercero y en el canal de comunicación seguro, al menos otro comando del lote de comandos, llamado segundo comando, seguro gracias a un código de autentificación de mensaje; y
que comprende además, a nivel del dispositivo esclavo (14), las etapas siguientes:

verificar el código de autentificación de mensaje con ayuda de la clave de sesión antes de ejecutar dicho segundo comando;
incrementar (902) un contador de ensayos en la recepción del primer comando que indica el establecimiento de un canal de comunicación seguro; y
sobreescribir (724, 728, 903, 906, 908) el valor actual de la variable dinámica en memoria no volátil con el valor de actualización calculado cuando el contador de ensayos alcanza un valor umbral.

2. Procedimiento según la reivindicación 1, en el que la etapa de cálculo de un valor de actualización (904) comprende el registro del valor de actualización calculado en una memoria volátil del dispositivo esclavo.

3. Procedimiento según la reivindicación 1 o 2, en el que el valor actual de la variable dinámica en memoria no volátil se sobreescribe (908) en memoria no volátil con el valor de actualización calculado cuando el último comando seguro del conjunto de los segundos comandos seguros comprende un código de autentificación de mensaje verificado válidamente y es ejecutado con éxito.

4. Procedimiento según la reivindicación 3, que comprende además, la etapa siguiente: transmitir, al dispositivo tercero y en respuesta a dicho último comando seguro, un mensaje que comprende una indicación de que la ejecución de dicho o de dichos segundos comandos del lote se ha realizado con éxito.

5. Procedimiento según una de las reivindicaciones 1 a 4, en el que el valor actual de la variable dinámica en memoria no volátil se sobreescribe (908) en memoria no volátil con el valor de actualización calculado cuando un llamado segundo comando cuyo código de autentificación de mensaje ha sido verificado con éxito conduce a un error o excepción de ejecución de este segundo comando.

6. Procedimiento según la reivindicación 5, que comprende además, la etapa siguiente: transmitir, al dispositivo tercero y en respuesta a dicho segundo comando que conduce a un error o excepción de ejecución, un mensaje que comprende una indicación del error o excepción de ejecución a pesar de un código de autentificación de mensaje verificado con éxito.

7. Procedimiento según una de las reivindicaciones 1 a 6, en el que el contador de ensayos se reinicializa (726, 730, 903, 906, 908) a un valor por defecto cuando el valor actual de la variable dinámica en memoria no volátil se sobreescribe con el valor de actualización calculado.

8. Procedimiento según la reivindicación 7, en el que la etapa que consiste en sobreescribir el valor actual en memoria no volátil con el valor de actualización calculado (726, 903) y la etapa que consiste en reinicializar el contador de ensayos (730, 903) se realizan antes de la etapa de generación (506) de la clave de sesión a partir del valor de actualización calculado.

9. Procedimiento según la reivindicación 7, en el que la etapa que consiste en sobreescribir el valor actual en memoria no volátil con el valor de actualización calculado (724, 728, 903, 906, 908) y la etapa que consiste en reinicializar el contador de ensayos (726, 730, 903, 906, 908) se realizan en una etapa atómica.

10. Procedimiento según una de las reivindicaciones 1 a 6, en el que dicho primer comando es un llamado segundo comando seguro gracias a un código de autentificación de mensaje.

11. Procedimiento de comunicación que comprende, a nivel de un dispositivo tercero (16), las etapas siguientes:

recibir (800), de un dispositivo maestro (12), un lote de comandos a enviar a un dispositivo esclavo (14), incluyendo el lote de comandos un comando, llamado primer comando (502), que indica el establecimiento, con el dispositivo esclavo, de un canal de comunicación seguro gracias a, al menos, una clave de sesión dependiente de un valor actual de una variable dinámica compartida entre el dispositivo maestro y el dispositivo esclavo, y al menos otro

comando (518, 526), llamado segundo comando, seguro gracias a un código de autentificación de mensaje calculado con ayuda de la clave de sesión;
inyectar (808) el lote de comandos para enviar, al dispositivo esclavo, los comandos del lote de forma secuencial, comando por comando, en tanto que no se detecte ningún error;
que comprende además, a nivel del dispositivo tercero (16), la etapa siguiente:
reinyectar (811) el lote de comandos en tanto que el dispositivo tercero no reciba, del dispositivo esclavo (14), una indicación de una actualización del valor actual de la variable dinámica por el dispositivo esclavo.

12. Procedimiento según la reivindicación 11, en el que recibir la indicación de actualización de la variable dinámica por el dispositivo esclavo comprende recibir (510, 814), del dispositivo esclavo (14), un valor actual de la variable dinámica y comparar (816) el valor actual recibido con un valor local de la variable dinámica.

13. Procedimiento según la reivindicación 12, en el que el valor actual de la variable dinámica es recibido, por el dispositivo tercero (16), en una respuesta (510) del dispositivo esclavo (14) al primer comando (502).

14. Procedimiento según la reivindicación 11, en el que recibir la indicación de una actualización de la variable dinámica por el dispositivo esclavo comprende recibir (522), del dispositivo esclavo (14), un mensaje de error en respuesta a un llamado segundo comando (518) inmediatamente subsiguiente al primer comando (502), indicando el mensaje de error un código de autentificación de mensaje erróneo de dicho segundo comando inmediatamente subsiguiente al primer comando.

15. Procedimiento según una de las reivindicaciones 11 a 14, en el que la indicación de una actualización del valor actual comprende una indicación de que la ejecución de dicho o de dichos segundos comandos del lote por el dispositivo esclavo se ha realizado con éxito.

16. Procedimiento según una de las reivindicaciones 11 a 15, en el que la indicación de una actualización del valor actual comprende una indicación de que la ejecución, por el dispositivo esclavo (14), de un segundo comando del lote cuyo código de autentificación de mensaje ha sido verificado con éxito ha conducido a un error de este segundo comando.

17. Procedimiento según una de las reivindicaciones 1 a 16, en el que los códigos de autentificación de mensaje de varios segundos comandos están encadenados.

18. Dispositivo de procesamiento (14) que comprende una memoria no volátil que memoriza un valor actual de una variable dinámica, y un procesador configurado para:

recibir, de un dispositivo tercero (16), al menos un comando, llamado primer comando, de un lote de comandos, indicando dicho primer comando el establecimiento de un canal de comunicación seguro gracias a, al menos, una clave de sesión;
calcular un valor de actualización de la variable dinámica a partir del valor actual almacenado en memoria no volátil; y
generar la clave de sesión a partir del valor de actualización calculado;
a continuación recibir, del dispositivo tercero y en el canal de comunicación seguro, al menos otro comando del lote de comandos, llamado segundo comando, seguro gracias a un código de autentificación de mensaje; y
comprendiendo el dispositivo de procesamiento además un contador de ensayos, y el procesador está configurado además para:

verificar el código de autentificación de mensaje con ayuda de la clave de sesión antes de ejecutar dicho segundo comando;
incrementar el contador de ensayos en la recepción del primer comando que indica el establecimiento de un canal de comunicación seguro; y
sobreescribir el valor actual de la variable dinámica en memoria no volátil con el valor de actualización calculado cuando el contador de ensayos alcanza un valor umbral.

19. Dispositivo de procesamiento (16) que comprende un procesador configurado para:

recibir, de un dispositivo maestro (12), un lote de comandos a enviar a un dispositivo esclavo (14), incluyendo el lote de comandos un comando, llamado primer comando, que indica el establecimiento, con el dispositivo esclavo, de un canal de comunicación seguro gracias a, al menos, una clave de sesión dependiente de un valor actual de una variable dinámica compartida entre el dispositivo maestro y el dispositivo esclavo, y al menos otro comando, llamado segundo comando, seguro gracias a un código de autentificación de mensaje calculado con ayuda de la clave de sesión;
inyectar el lote de comandos para enviar, al dispositivo esclavo, dichos comandos del lote de forma secuencial, comando por comando, en tanto que no se detecte ningún error;
estando el proceso configurado además para: reinyectar el lote de comandos en tanto que el dispositivo de procesamiento no reciba, del dispositivo esclavo, una indicación de una actualización del valor actual de la variable

dinámica por el dispositivo esclavo.

20. Sistema (10) que comprende un dispositivo esclavo (14) según la reivindicación 18, un dispositivo tercero (16) según la reivindicación 19, y un dispositivo maestro (12) configurado para generar y enviar el lote de comandos al dispositivo tercero (16).

21. Producto de programa informático que comprende instrucciones adaptadas a la implementación de cada una de las etapas del procedimiento según una cualquiera de las reivindicaciones 1 a 17, cuando dicho programa es ejecutado en un ordenador.

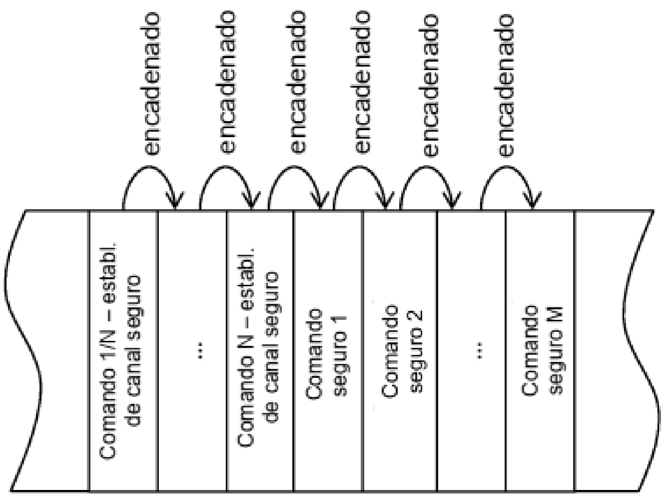


Fig. 1

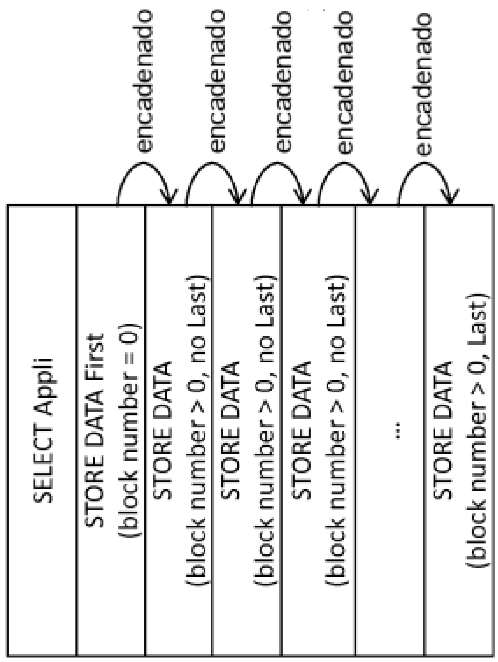


Fig. 1A

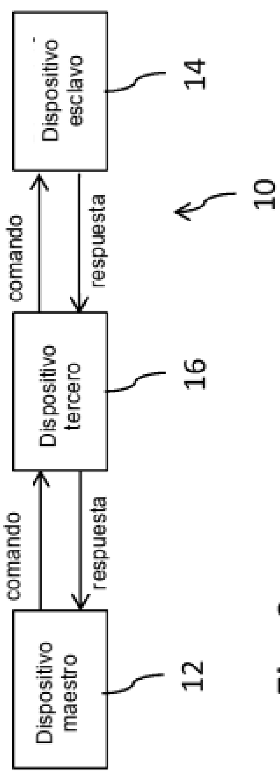


Fig. 2

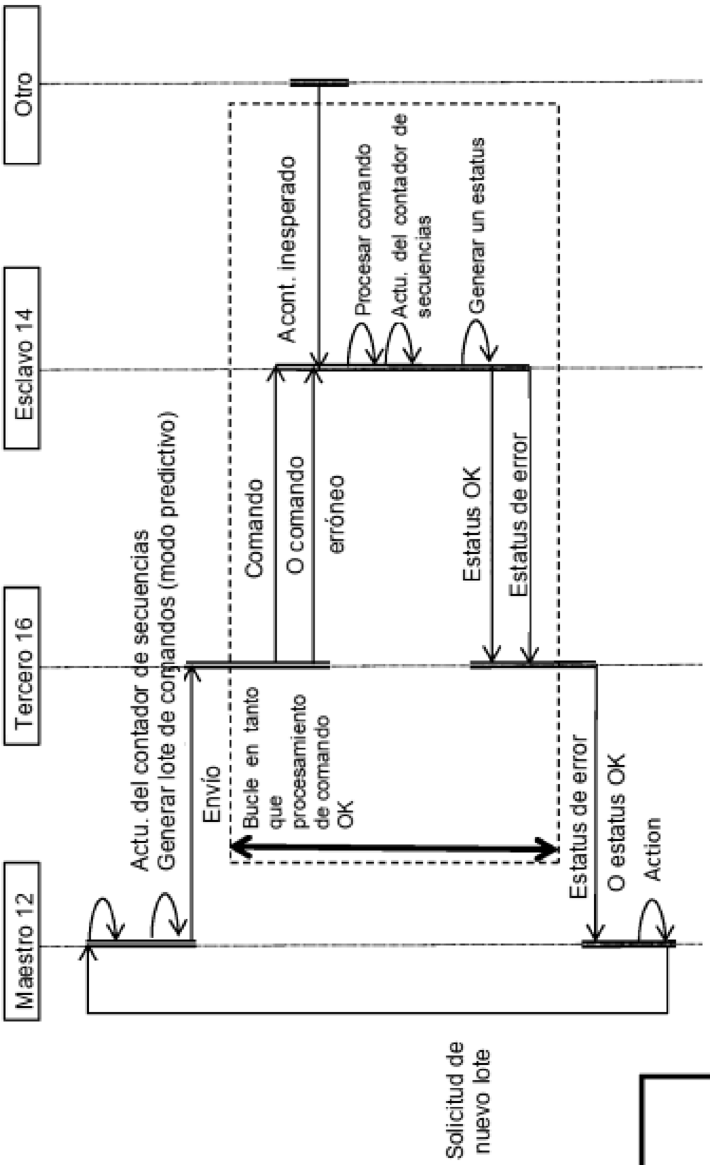

Maestro 12
Tercero 16
Esclavo 14
Otro
Actu. del contador de secuencias
Generar lote de comandos (modo predictivo)
Envío
Bucle en tanto que procesamiento de comando OK
Comando
O comando erróneo
Acont. inesperado
Procesar comando
Actu. del contador de secuencias
Generar un estatus
Estatus OK
Estatus de error
Estatus de error
O estatus OK
Action
Solicitud de nuevo lote

Fig. 3

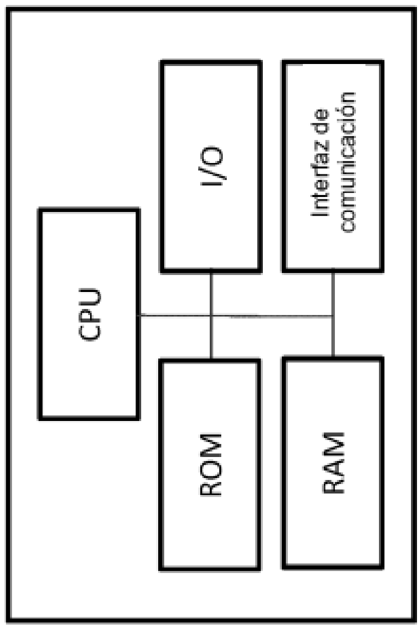

CPU
I/O
Interfaz de comunicación
ROM
RAM

Fig. 4

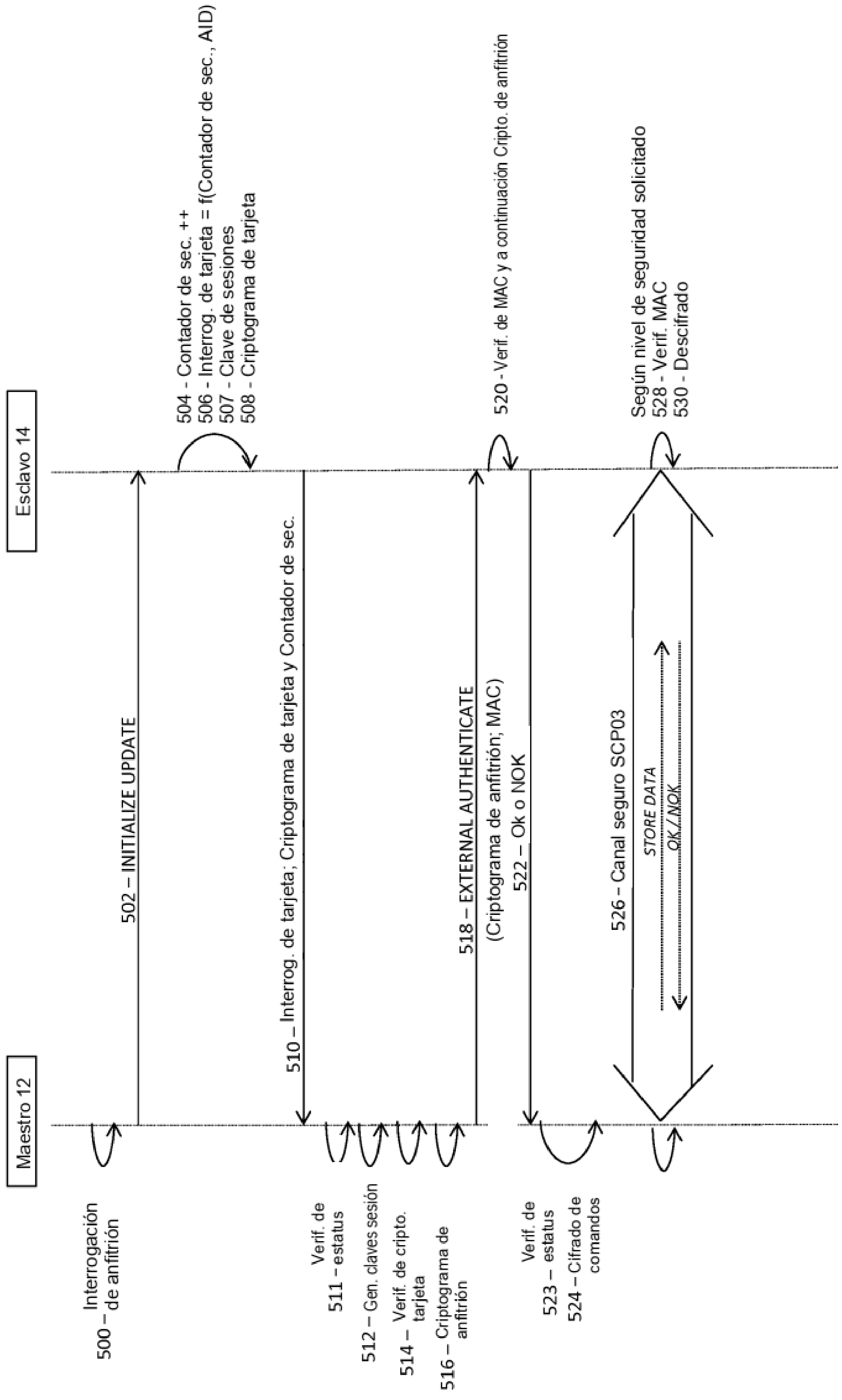
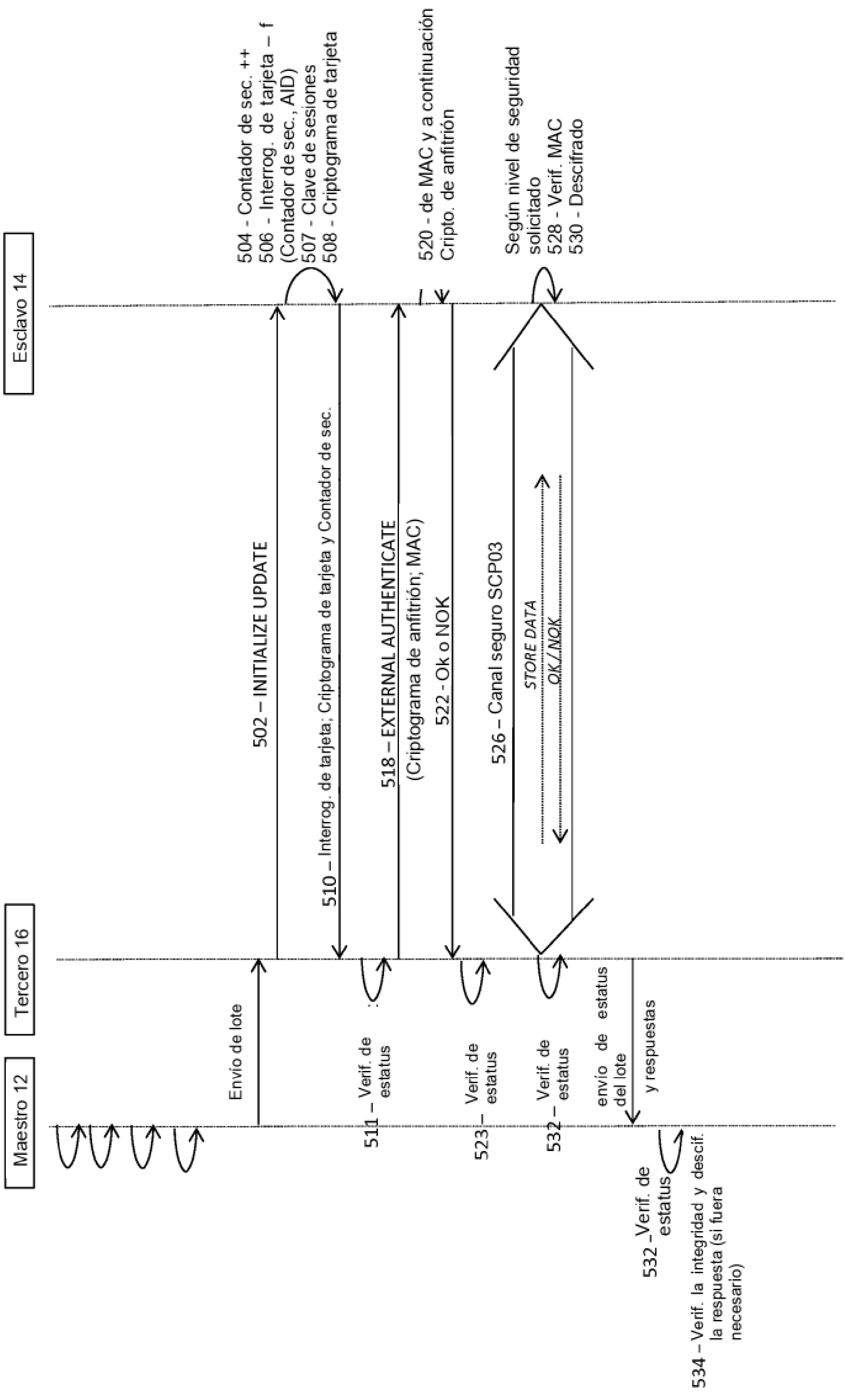

Maestro 12
Esclavo 14
500 – Interrogación de anfitrión
502 – INITIALIZE UPDATE
504 - Contador de sec. ++
506 - Interrog. de tarjeta = f(Contador de sec., AID)
507 - Clave de sesiones
508 - Criptograma de tarjeta
510 – Interrog. de tarjeta; Criptograma de tarjeta y Contador de sec.
511 – Verif. de estatus
512 – Gen. claves sesión
514 – Verif. de cripto. tarjeta
516 – Criptograma de anfitrión
518 – EXTERNAL AUTHENTICATE
(Criptograma de anfitrión; MAC)
520 - Verif. de MAC y a continuación Cripto. de anfitrión
522 – Ok o NOK
523 – Verif. de estatus
524 – Cifrado de comandos
526 – Canal seguro SCP03
STORE DATA
OK / NOK
Según nivel de seguridad solicitado
528 - Verif. MAC
530 - Descifrado

Fig. 5



Maestro 12
Tercero 16
Esclavo 14
Envío de lote
502 – INITIALIZE UPDATE
504 - Contador de sec. ++
506 - Interrog. de tarjeta – f (Contador de sec., AID)
507 - Clave de sesiones
508 - Criptograma de tarjeta
510 – Interrog. de tarjeta; Criptograma de tarjeta y Contador de sec.
511 – Verif. de estatus
518 – EXTERNAL AUTHENTICATE (Criptograma de anfitrión; MAC)
520 - de MAC y a continuación Cripto. de anfitrión
522 - Ok o NOK
523 – Verif. de estatus
526 – Canal seguro SCP03
STORE DATA
OK / NOK
Según nivel de seguridad solicitado
528 - Verif. MAC
530 - Descifrado
532 – Verif. de estatus
envío de estatus del lote y respuestas
532 – Verif. de estatus
534 – Verif. la integridad y descif. la respuesta (si fuera necesario)

Fig. 5A

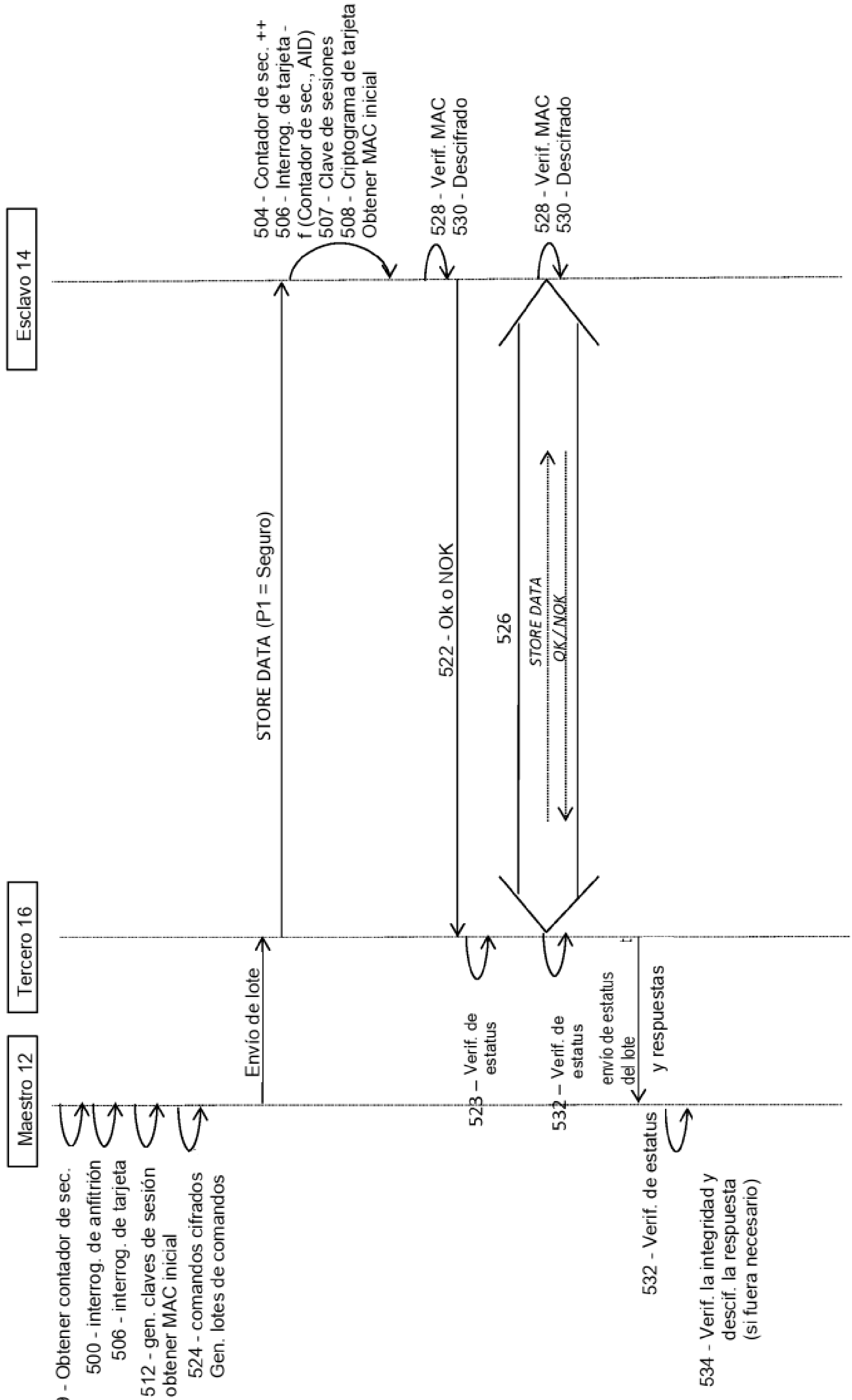
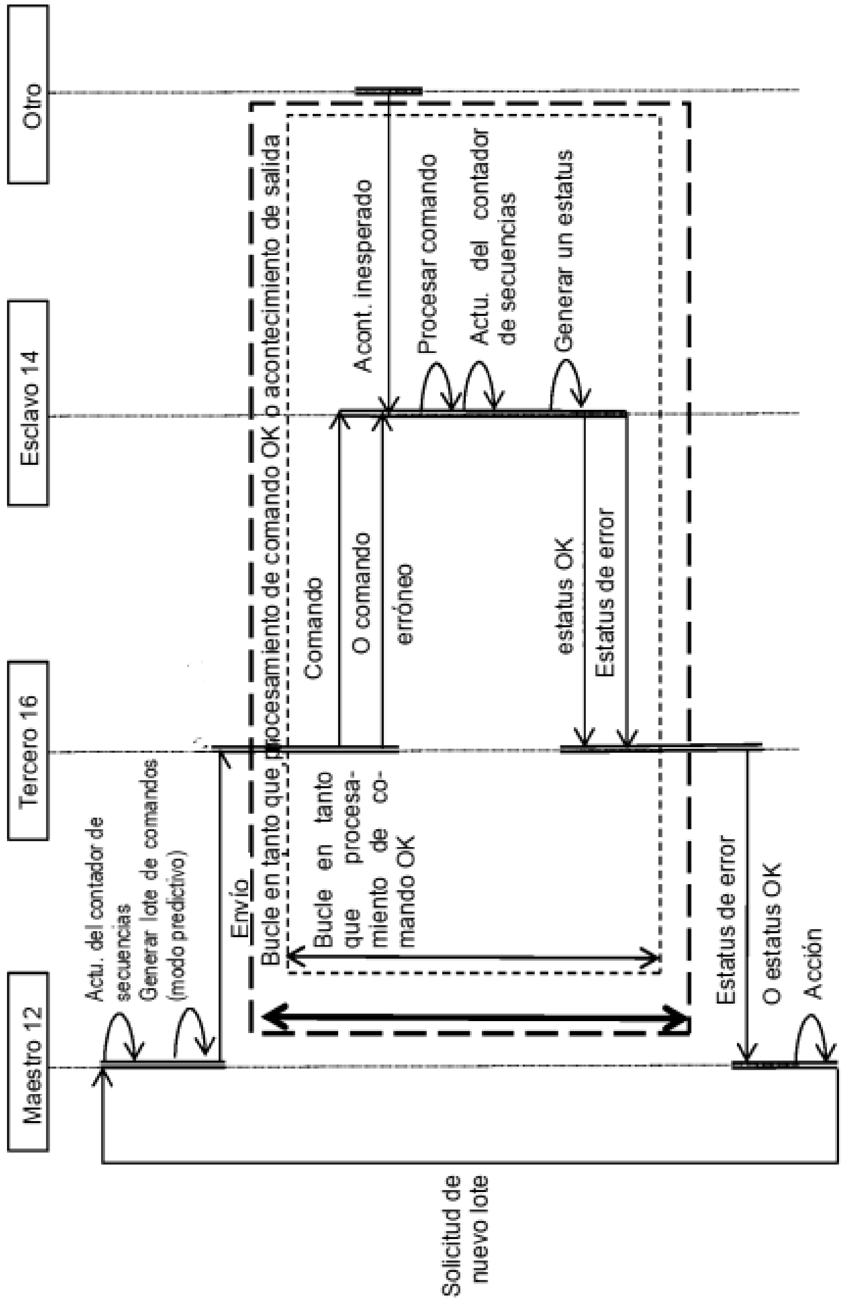

Maestro 12
Tercero 16
Esclavo 14
499 - Obtener contador de sec.
500 - interrog. de anfitrión
506 - interrog. de tarjeta
512 - gen. claves de sesión
obtener MAC inicial
524 - comandos cifrados
Gen. lotes de comandos
Envío de lote
STORE DATA (P1 = Seguro)
504 - Contador de sec. ++
506 - Interrog. de tarjeta -
f (Contador de sec., AID)
507 - Clave de sesiones
508 - Criptograma de tarjeta
Obtener MAC inicial
528 - Verif. MAC
530 - Descifrado
522 - Ok o NOK
523 – Verif. de
estatus
526
STORE DATA
OK / NOK
528 - Verif. MAC
530 - Descifrado
532 – Verif. de
estatus
envío de estatus
del lote
y respuestas
532 - Verif. de estatus
534 - Verif. la integridad y
descif. la respuesta
(si fuera necesario)

Fig. 5B



Maestro 12
Tercero 16
Esclavo 14
Otro
Actu. del contador de secuencias
Generar lote de comandos (modo predictivo)
Envío
Bucle en tanto que procesamiento de comando OK o acontecimiento de salida
Bucle en tanto que procesamiento de comando OK
Comando
O comando erróneo
Acont. inesperado
Procesar comando
Actu. del contador de secuencias
Generar un estatus
estatus OK
Estatus de error
Estatus de error
O estatus OK
Acción
Solicitud de nuevo lote

Fig. 6

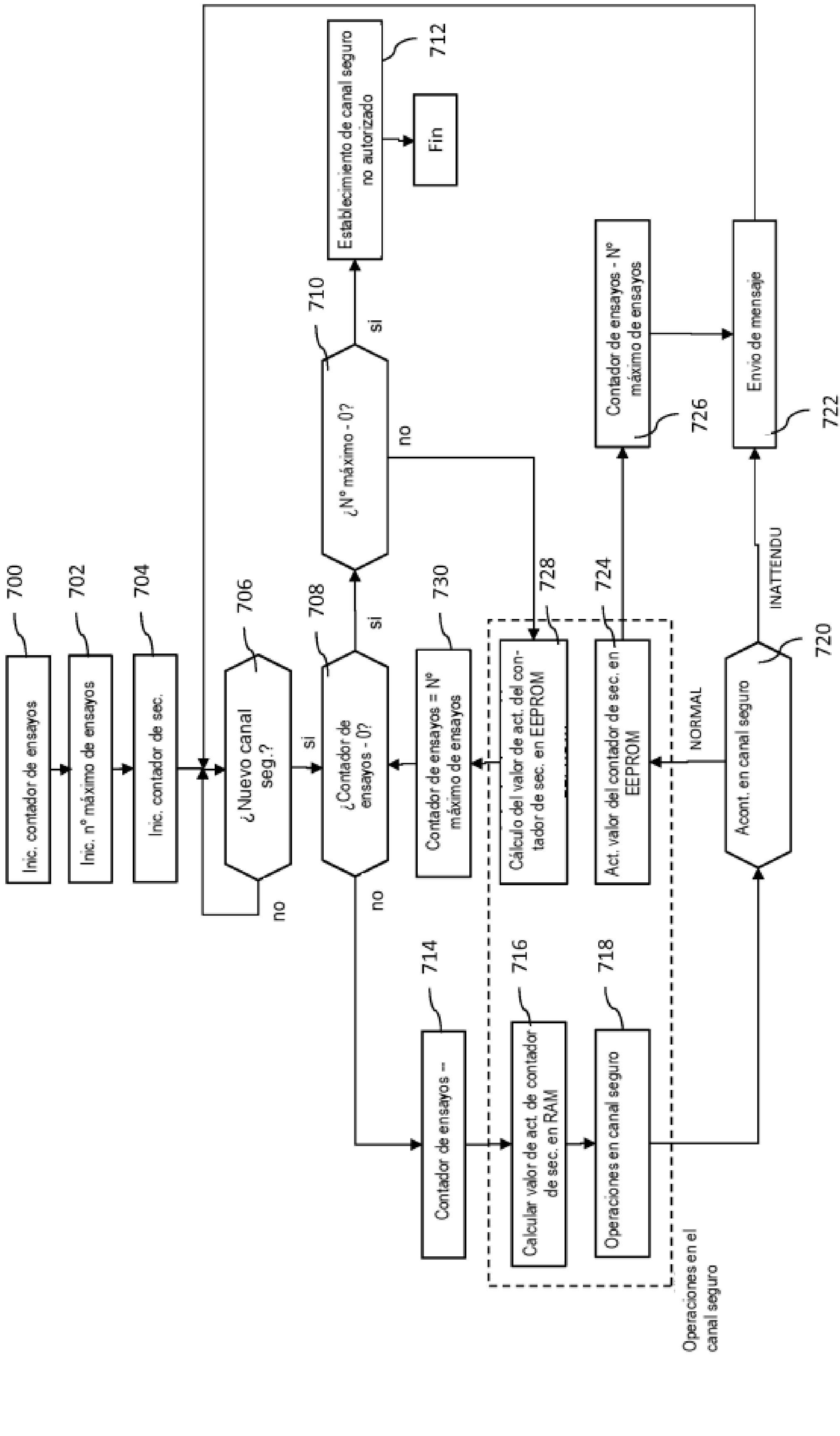

Inic. contador de ensayos
700
Inic. nº máximo de ensayos
702
Inic. contador de sec.
704
¿Nuevo canal seg.?
706
no
si
¿Contador de ensayos - 0?
708
si
no
¿Nº máximo - 0?
710
si
no
Establecimiento de canal seguro no autorizado
712
Fin
Contador de ensayos --
714
Calcular valor de act. de contador de sec. en RAM
716
Operaciones en canal seguro
718
Operaciones en el canal seguro
Acont. en canal seguro
720
NORMAL
INATTENDU
Envío de mensaje
722
Act. valor del contador de sec. en EEPROM
724
Contador de ensayos - Nº máximo de ensayos
726
Cálculo del valor de act. del contador de sec. en EEPROM
728
Contador de ensayos = Nº máximo de ensayos
730

Fig. 7

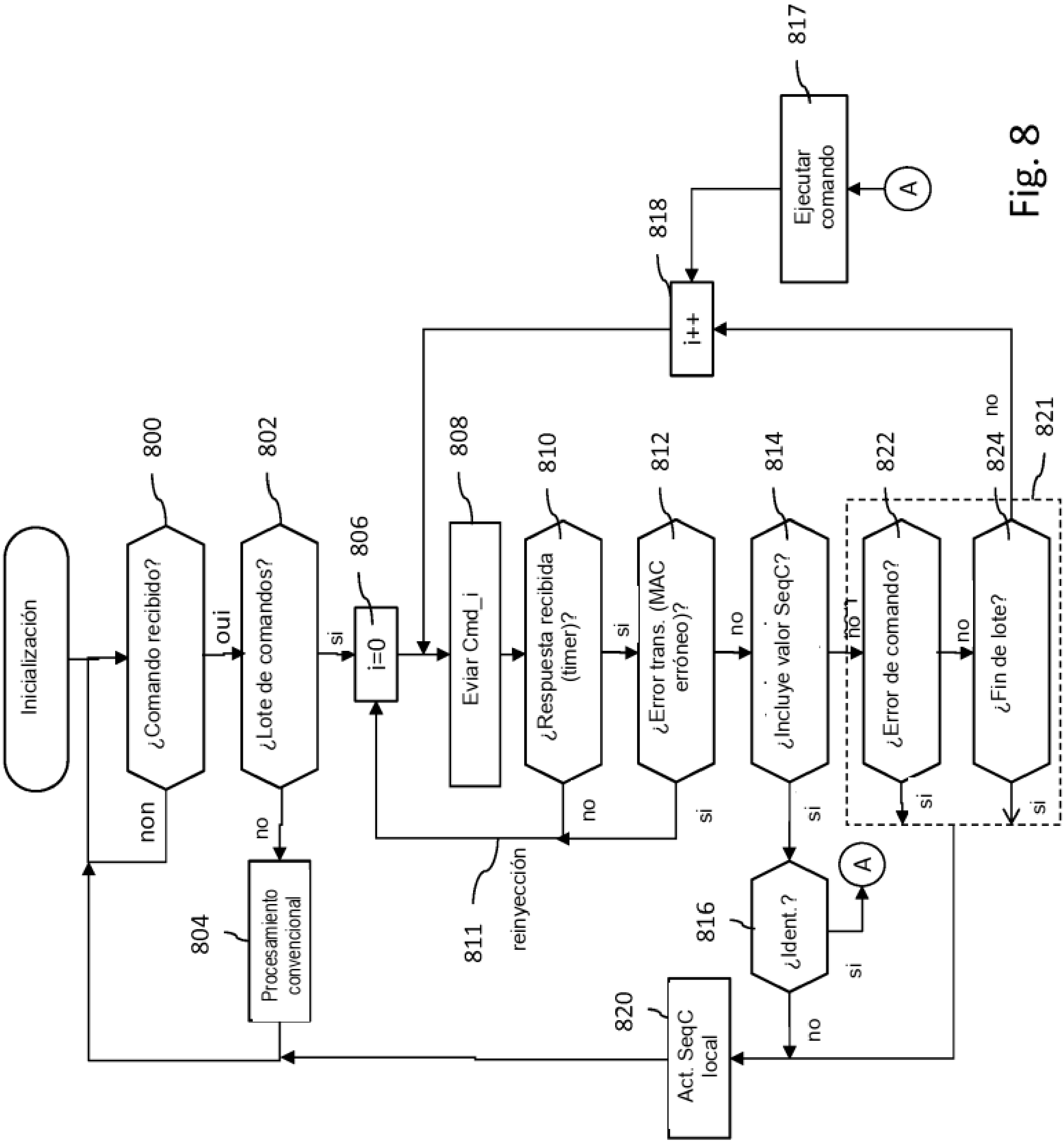

Inicialización
¿Comando recibido?
800
oui
non
¿Lote de comandos?
802
si
no
Procesamiento convencional
804
i=0
806
Eviar Cmd_i
808
¿Respuesta recibida (timer)?
810
no
si
reinyección
811
¿Error trans. (MAC erróneo)?
812
si
no
¿Incluye valor SeqC?
814
si
no
¿Ident.?
816
si
no
A
Act. SeqC local
820
¿Error de comando?
822
si
no
¿Fin de lote?
824
si
no
821
i++
818
Ejecutar comando
817
A

Fig. 8

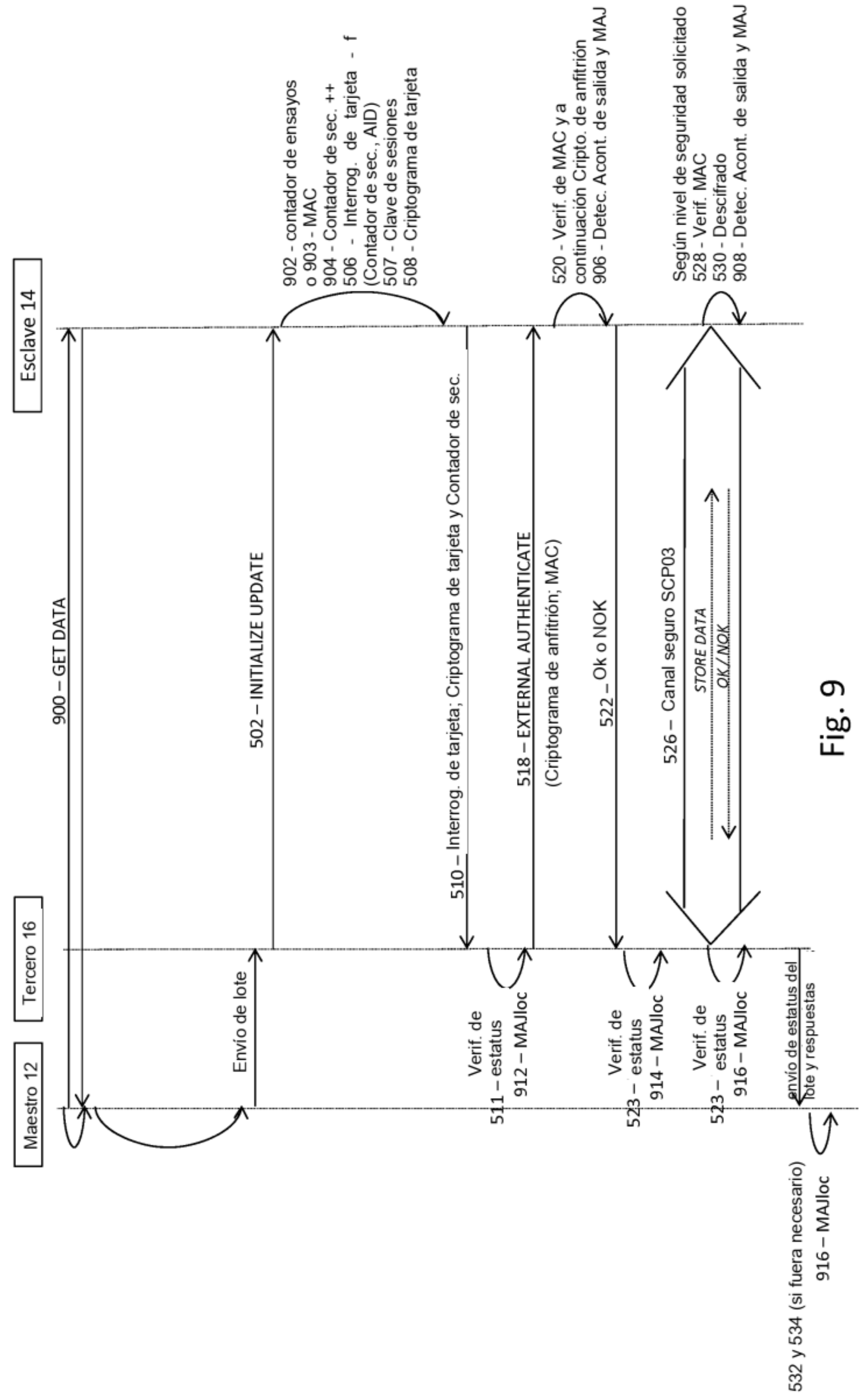

Maestro 12
Tercero 16
Esclave 14
900 – GET DATA
Envío de lote
502 – INITIALIZE UPDATE
902 - contador de ensayos
o 903 - MAC
904 - Contador de sec. ++
506 - Interrog. de tarjeta - f
(Contador de sec., AID)
507 - Clave de sesiones
508 - Criptograma de tarjeta
510 – Interrog. de tarjeta; Criptograma de tarjeta y Contador de sec.
Verif. de
511 – estatus
912 – MAJloc
518 – EXTERNAL AUTHENTICATE
(Criptograma de anfitrión; MAC)
520 - Verif. de MAC y a
continuación Cripto. de anfitrión
906 - Detec. Acont. de salida y MAJ
522 – Ok o NOK
Verif. de
523 – estatus
914 – MAJloc
526 – Canal seguro SCP03
STORE DATA
OK / NOK
Según nivel de seguridad solicitado
528 - Verif. MAC
530 - Descifrado
908 - Detec. Acont. de salida y MAJ
Verif. de
523 – estatus
916 – MAJloc
envío de estatus del
lote y respuestas
532 y 534 (si fuera necesario)
916 – MAJloc

Fig. 9

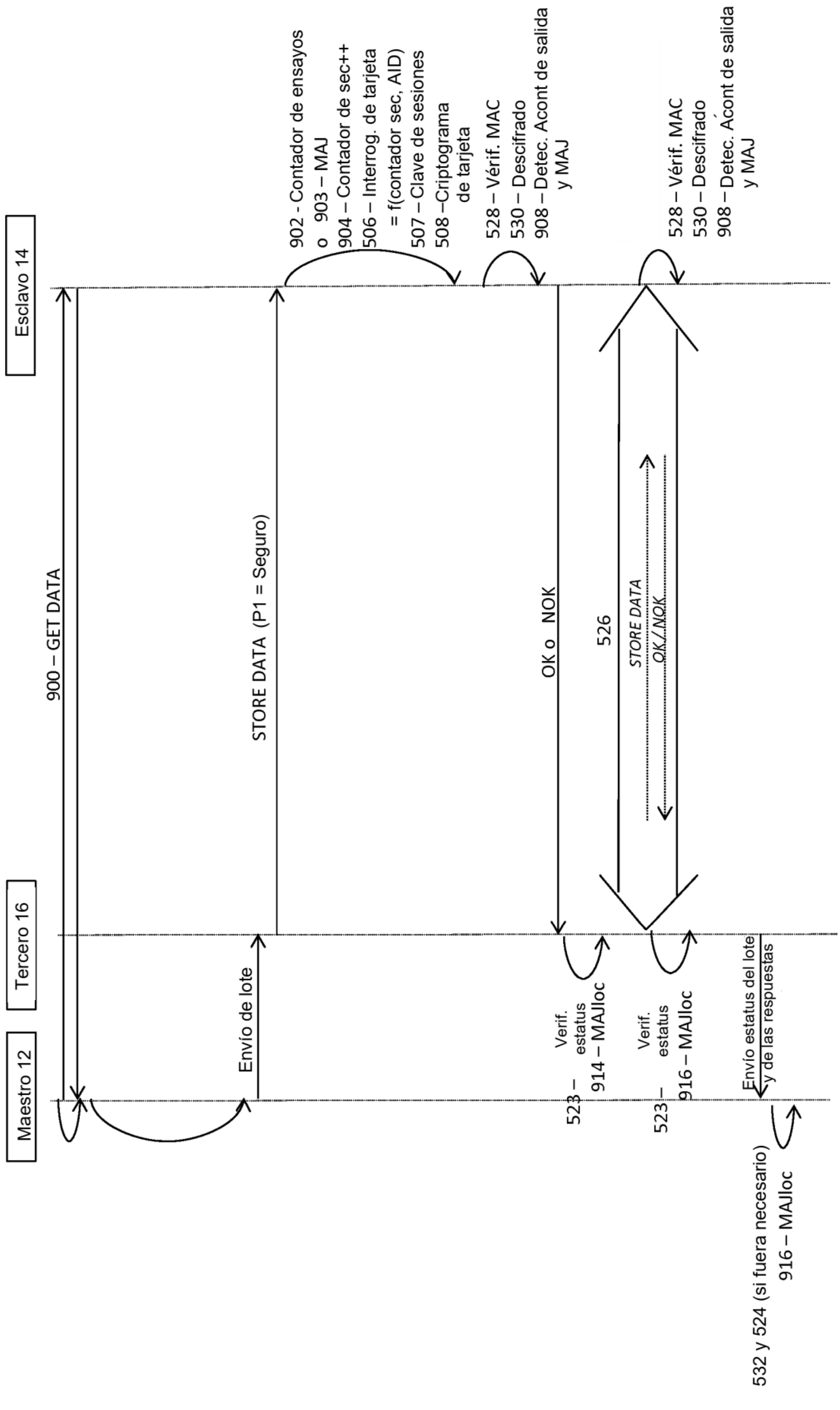
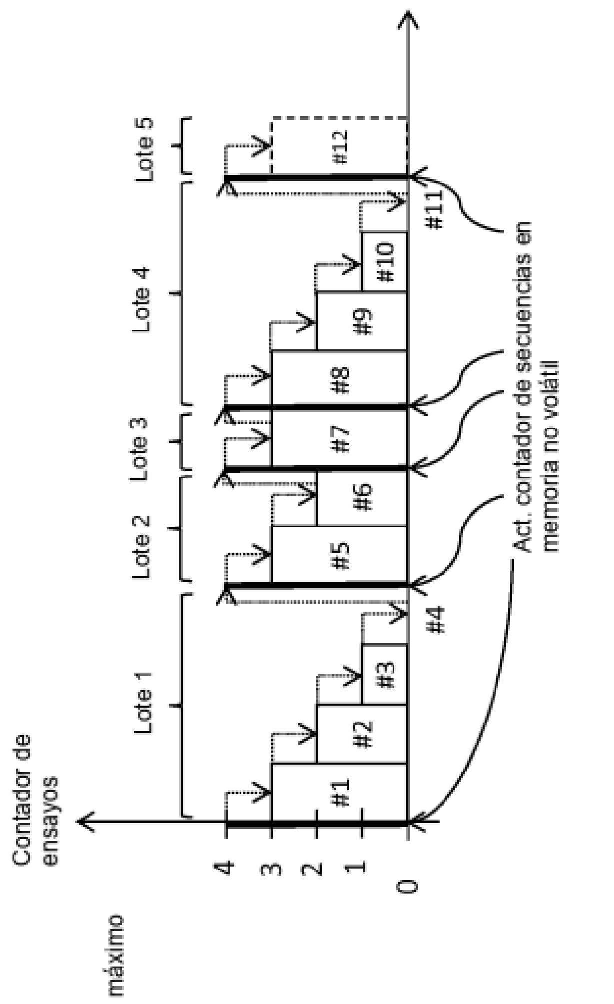

Maestro 12
Tercero 16
Esclavo 14
900 – GET DATA
Envío de lote
STORE DATA (P1 = Seguro)
902 - Contador de ensayos
o 903 – MAJ
904 – Contador de sec++
506 – Interrog. de tarjeta
= f(contador sec, AID)
507 – Clave de sesiones
508 –Criptograma
de tarjeta
528 – Vérif. MAC
530 – Descifrado
908 – Detec. Acont de salida
y MAJ
OK o NOK
526
STORE DATA
OK / NOK
528 – Vérif. MAC
530 – Descifrado
908 – Detec. Ácont de salida
y MAJ
523 –
Verif.
estatus
914 – MAJloc
523–
Verif.
estatus
916 – MAJloc
Envío estatus del lote
y de las respuestas
532 y 524 (si fuera necesario)
916 – MAJloc

Fig. 9A



Contador de ensayos
máximo
4
3
2
1
0
Lote 1
Lote 2
Lote 3
Lote 4
Lote 5
#1
#2
#3
#4
#5
#6
#7
#8
#9
#10
#11
#12
Act. contador de secuencias en memoria no volátil

Fig. 10