(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 442 630**

(51) Int. CI.:

H04N 5/00	(2011.01)	***H04N 21/845***	(2011.01)
H04N 7/16	(2011.01)		
G06F 9/445	(2006.01)		
G06F 21/00	(2013.01)		
H04H 60/23	(2008.01)		
H04N 21/418	(2011.01)		
H04N 21/4623	(2011.01)		
H04N 21/262	(2011.01)		
H04N 21/458	(2011.01)		
H04N 21/6543	(2011.01)		

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **21.12.2010 E 10812872 (9)**

(97) Fecha y número de publicación de la concesión europea: **06.11.2013 EP 2517452**

(54) Título: **Procedimiento de actualización de un procesador de seguridad, sistema, programa de ordenador y procesador de seguridad correspondientes**

(30) Prioridad:

23.12.2009 FR 0959511

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
12.02.2014

(73) Titular/es:

VIACCESS (100.0%)
Les Collines de l'Arche Opéra C
92057 Paris La Defense Cedex , FR

(72) Inventor/es:

PRESTY, RENAUD y
GRANET, OLIVIER

(74) Agente/Representante:

CURELL AGUILÁ, Mireia

DESCRIPCIÓN

Procedimiento de actualización de un procesador de seguridad, sistema, programa de ordenador y procesador de seguridad correspondientes.

La presente invención se refiere a un procedimiento de actualización de un procesador de seguridad.

Se refiere asimismo a un sistema de actualización, a un programa de ordenador y a un procesador de seguridad correspondientes.

La invención se aplica en particular a la actualización de un procesador de seguridad para un decodificador apropiado para recibir y desaleatorizar una señal multimedia aleatorizada, siendo difundida dicha señal multimedia por una cabecera de red y comprendiendo dicho procesador una memoria que contiene un código de una aplicación que, cuando es ejecutado o interpretado por un microprocesador, permite ejecutar un conjunto de operaciones necesarias para la desaleatorización de la señal multimedia aleatorizada, comprendiendo dicho procedimiento una etapa de descarga de por lo menos un módulo del código a partir de la cabecera de red.

Un decodificador del tipo citado previamente forma parte de un sistema de acceso condicional tal como el dado a conocer en el documento “Functional model of a conditional access system”, EBU *Project Group* B/CA, Winter 1995.

La invención se aplica en particular al campo de la televisión de pago. En este contexto, el decodificador dotado del procesador de seguridad constituido, a título de ejemplo, por una tarjeta chip, se encuentra en la vivienda de los clientes y su vida útil sobre el terreno supera generalmente los cinco años.

Ante las necesidades de corrección y/o de evolución, es necesario disponer de un procedimiento de actualización del procesador de seguridad en la vivienda del cliente sin intervención alguna por parte de este último, descargando en el procesador de seguridad las partes del código que requieren una actualización.

Los procedimientos convencionales de descarga de código de microprocesador son de dos tipos.

El primer tipo de descarga condicional consiste en efectuar la descarga gracias a un mecanismo dentro de la aplicación del procesador de seguridad, que permite descargar un código suplementario y reencaminar ciertas funciones del código existente hacia el código descargado.

Este primer tipo de descarga plantea problemas de seguridad. En efecto, es posible reencaminar la totalidad de las funciones del código existente hacia un código sustitutivo nuevo y descargar un código nuevo que permite volcar al contenido de las memorias. Además, si el mecanismo de descarga presenta un error informático o un fallo de seguridad, el conjunto del código del procesador de seguridad queda expuesto a ataques.

El segundo tipo de descarga convencional consiste en utilizar un mecanismo dentro del sistema operativo del procesador de seguridad para sustituir la totalidad del código existente por un código sustitutivo nuevo. También en esta circunstancia, en caso de error informático o de fallo de seguridad del mecanismo, el conjunto del código del procesador de seguridad queda expuesto a ataques.

El documento WO 2004/051983 describe un procedimiento de protección de las actualizaciones del código de un procesador de seguridad del tipo citado previamente al descargar por lo menos un módulo del código cifrado con la ayuda de una clave pública que se invalida después de la actualización.

Este procedimiento presenta el inconveniente, durante el agotamiento de las claves públicas almacenadas, de necesitar una actualización de por lo menos una nueva clave pública de cifrado, lo cual introduce así un potencial fallo de seguridad a nivel de la descarga de esta clave.

El objetivo de la invención es por lo tanto resolver estos problemas.

Con este fin, la invención tiene por objeto, un procedimiento de actualización de un procesador de seguridad para un decodificador apropiado para recibir y desaleatorizar una señal multimedia aleatorizada, siendo difundida dicha señal multimedia por una cabecera de red y comprendiendo dicho procesador una memoria que contiene un código de una aplicación que, cuando es ejecutado o interpretado por un microprocesador, permite ejecutar un conjunto de operaciones necesarias para la desaleatorización de la señal multimedia aleatorizada, comprendiendo dicho procedimiento una etapa de descarga de por lo menos un módulo del código a partir de la cabecera de red, caracterizado porque la etapa de descarga del módulo comprende las subetapas de:

- cifrado, por parte de la cabecera de red, del módulo con la ayuda de una clave y de un algoritmo de cifrado apropiados para dicho módulo;

- segmentación, por parte de la cabecera de red, del módulo en una pluralidad de bloques de tamaño

apropiado para dicho módulo;

- integración, por parte de la cabecera de red, de cada uno de los bloques del módulo en un mensaje específico;

- difusión, por parte de la cabecera de red hacia el decodificador, de cada uno de estos mensajes;

- recepción de estos mensajes por el decodificador;

- extracción, por parte del procesador de seguridad del decodificador, de los bloques del módulo a partir de estos mensajes;

- descifrado, por parte del procesador de seguridad, de los bloques del módulo;

- almacenamiento, por parte del procesador de seguridad, de los bloques del módulo en un espacio de la memoria dedicado a dicho módulo; y

- obtención a partir de estos bloques de un módulo de código actualizado.

Según modos particulares de realización, el procedimiento comprende una o varias de las siguientes características, considerada(s) de forma aislada o según todas las combinaciones técnicamente posibles:

- la etapa de descarga de módulo comprende además las subetapas de:

  - cifrado, por parte de la cabecera de red, de los mensajes que contienen los bloques del módulo con la ayuda de claves de transporte apropiadas para el módulo; y

  - descifrado, por parte del procesador de seguridad, de los mensajes que contienen los bloques del módulo.

- los mensajes que contienen los bloques del módulo se seleccionan de tipo ECM (“Entitlement Control Message”) y/o EMM (“Entitlement Management Message”) según una elección apropiada para el módulo.

- comprende además una etapa de activación, por parte del procesador de seguridad, del módulo descargado almacenado en la memoria.

- la etapa de activación del módulo descargado es activada por el procesador de seguridad automáticamente o como consecuencia de la recepción de una orden de activación emitida desde el decodificador o desde la cabecera de red.

- el módulo se desarrolla y compila en un lenguaje y con la ayuda de un compilador apropiados para el módulo.

La invención tiene también por objeto un sistema de actualización de un procesador de seguridad para un decodificador apropiado para recibir y desaleatorizar una señal multimedia aleatorizada, siendo difundida dicha señal multimedia por una cabecera de red y comprendiendo dicho procesador una memoria que contiene un código de una aplicación que, cuando es ejecutado o interpretado por un microprocesador, permite ejecutar un conjunto de operaciones necesarias para la desaleatorización de la señal multimedia aleatorizada, comprendiendo dicho sistema unos medios de descarga de por lo menos un módulo del código a partir de la cabecera de red, caracterizado porque los medios de descarga del módulo comprenden:

a nivel de la cabecera de red:

- unos medios de cifrado del módulo con la ayuda de una clave y de un algoritmo de cifrado apropiados para dicho módulo;

- unos medios de segmentación del módulo en una pluralidad de bloques de tamaño apropiado para dicho módulo;

- unos medios de integración de cada uno de los bloques del módulo en un mensajes específico; y

- unos medios de difusión hacia el decodificador de cada uno de estos mensajes

a nivel del decodificador, unos medios de recepción de estos mensajes; y

a nivel del procesador de seguridad:

- unos medios de extracción de los bloques del módulo a partir de estos mensajes;

- unos medios de descifrado de los bloques del módulo;

- unos medios de almacenamiento de los bloques del módulo en un espacio de la memoria dedicado a dicho módulo; y

- unos medios de obtención, a partir de estos bloques, de un módulo de código actualizado.

La invención tiene todavía por objeto un programa de ordenador que comprende instrucciones de código, que, cuando este programa es ejecutado en un ordenador, permiten poner en práctica el procedimiento de actualización de un procesador de seguridad.

Según modos particulares de realización, el programa de ordenador comprende una o varias de las siguientes características considerada(s) de forma aislada o según todas las combinaciones técnicamente posibles:

- comprende un subprograma ejecutado por una cabecera de red para permitir poner en práctica las subetapas de la etapa de descarga del módulo que consisten en:

- cifrar el módulo con la ayuda de una clave y de un algoritmo de cifrado apropiados para dicho módulo;

- segmentar el módulo en una pluralidad de bloques de tamaño apropiado para dicho módulo;

- integrar cada uno de los bloques del módulo en un mensaje específico; y

- difundir hacia el decodificador cada uno de estos mensajes.

- comprende un subprograma ejecutado o interpretado por un procesador de seguridad para permitir poner en práctica las subetapas de la etapa de descarga del módulo que consisten en:

  - extraer los bloques del módulo a partir de estos mensajes;

  - descifrar los bloques del módulo;

  - almacenar los bloques del módulo en un espacio de la memoria dedicado a dicho módulo; y

  - obtener, a partir de estos bloques, un módulo de código actualizado.

La invención tiene también por objeto un procesador de seguridad, caracterizado porque es apto para ejecutar o para interpretar el código de un subprograma con el fin de permitir poner en práctica las subetapas de la etapa de descarga del módulo que consisten en:

- extraer los bloques del módulo a partir de estos mensajes;

- descifrar los bloques del módulo;

- almacenar los bloques del módulo en un espacio de la memoria dedicado a dicho módulo; y

- obtener, a partir de estos bloques, un módulo de código actualizado.

Así, la invención permite remediar los inconvenientes antes citados de los procedimientos del estado de la técnica al proponer un mecanismo de descarga del código de un procesador de seguridad que dispone de elementos diferenciadores predefinidos para cada etapa de la descarga en función del módulo de código a descargar.

La invención se comprenderá mejor al leer la descripción que se ofrece seguidamente, proporcionada únicamente a título de ejemplo y realizada en referencia a los dibujos adjuntos en los cuales:

- la figura 1 es un esquema sinóptico que ilustra la estructura de un sistema de acceso condicional convencional en el cual se pone en práctica la invención;

- la figura 2 es un organigrama que ilustra el funcionamiento del procedimiento de actualización de un procesador de seguridad según la invención;

- la figura 3 es un organigrama que ilustra las etapas del procedimiento de actualización referentes a la generación de los mensajes de actualización del código del procesador de seguridad; y

- la figura 4 es un organigrama que ilustra las etapas del procedimiento de actualización puestas en práctica por el procesador de seguridad del decodificador del sistema de acceso condicional.

La figura 1 representa la estructura de un sistema de acceso condicional 2 que permite la transmisión de señales multimedia aleatorizadas. Estas señales multimedia son, por ejemplo, señales audiovisuales que se corresponden con programas audiovisuales de cadenas de televisión generados por lo menos por un operador 4.

El sistema 2 comprende una cabecera de red 6 que comprende unos medios de multiplexado y de aleatorización 8 adaptados para aleatorizar las señales multimedia 4 que a continuación son transmitidas por la cabecera de red 6 hacia un número elevado de receptores por medio de una red 10 de transmisión de datos, por ejemplo una red inalámbrica o una red por cable.

Para simplificar la figura 1, se representa solamente un receptor 12.

El receptor 12 comprende un decodificador 14 apropiado para recibir y desaleatorizar las señales multimedia aleatorizadas, transmitidas desde la cabecera de red 6.

El decodificador 14 comprende un procesador de seguridad 16, por ejemplo una tarjeta chip, que comprende una memoria 18 que contiene un código de una aplicación que permite ejecutar un conjunto de operaciones necesarias para la desaleatorización de la señal multimedia aleatorizada. El código de esta aplicación comprende típicamente varios módulos de código que se corresponden con agrupamientos de subprogramas o funciones diferentes. Estos subprogramas o funciones pueden tener una naturaleza diversa, y son por ejemplo algoritmos de cifrado o de verificación de autenticación, analizadores léxicos de mensajes a procesar, y módulos de registro de títulos de acceso. La segmentación de la aplicación de módulos de códigos predefinidos se realiza según el arbitraje del arquitecto del software durante la fase de diseño.

Además, el sistema 2 de acceso condicional comprende unos medios de generación de mensajes de acceso condicional 20 de tipo ECM (“Entitlement Control Message”) y EMM (“Entitlement Management Message”).

Los medios de multiplexado y de aleatorización 8 de la cabecera de red 6 permiten multiplexar estos mensajes de acceso condicional con las señales multimedia aleatorizadas en un mismo paquete de datos.

La figura 2 ilustra el funcionamiento del procedimiento de actualización del procesador de seguridad 16 del decodificador 14 según la invención.

El procedimiento de actualización comienza con una etapa 30 de identificación del módulo o de los módulos de código que se deben descargar como sustitutivos de un módulo o de módulos existentes con el fin de efectuar la actualización del procesador de seguridad. Esta etapa 30 permite activar las etapas siguientes de descarga del módulo o de los módulos identificados.

En 32, la cabecera de red 6 lleva a cabo un formateo del módulo o de los módulos a descargar según un formato apropiado (por ejemplo, valores hexadecimales del código binario codificado en ASCII, balizas que marcan el inicio o el final del bloque de código, número de líneas, presencia de retorno de línea...) para cada módulo. En el transcurso de esta etapa 32, la cabecera de red 6 cifra el módulo o los módulos con la ayuda de una clave y de un algoritmo de cifrado apropiados para cada módulo. Así, al final de la etapa 32, los módulos a descargar quedan protegidos en cuanto a integridad y/o en cuanto a confidencialidad.

En 34, la cabecera de red 6 segmenta cada uno de los módulos a descargar en bloques de tamaño apropiado para dicho módulo e integra cada uno de los bloques en un mensaje específico de tipo EMM o ECM, pudiendo aceptar cada módulo un tipo de mensaje y no pudiendo aceptar el otro o bien pudiendo aceptar los dos tipos de mensajes.

Así, gracias a la asociación de un tamaño de bloque apropiado para cada módulo, un bloque de un módulo dado no es utilizable para otro módulo.

También en 34, la cabecera de red 6 codifica los encabezamientos de todos los mensajes que comprenden los bloques de datos de un módulo dado según una codificación apropiada para dicho módulo. Esta codificación permite indicar la acción a realizar por el procesador de seguridad 16.

Según un modo de realización preferido de la invención, la cabecera de red 6 cifra los mensajes que contienen los bloques de cada módulo con la ayuda de claves de transporte apropiadas para el módulo.

En 36, la cabecera de red 6 difunde, a través de la red 10, los mensajes que contienen los bloques de los módulos a descargar en un flujo de transporte de tipo MPEG 2 (MPEG2-TS) por ejemplo. Para ello, la cabecera de red 6 envía los mensajes de cada módulo a descargar a todos los abonados o a un grupo de abonados determinados o a un solo abonado según reglas de direccionamiento apropiadas para cada módulo. Por ejemplo, un módulo poco sensible puede tener una difusión global mientras que un módulo crítico puede tener una difusión muy dirigida.

En 38, el decodificador 14 recibe el flujo de transporte MPEG 2 que comprende las señales multimedia y los mensajes EMM y/o ECM aleatorizados.

En 40 extrae los mensajes que contienen los bloques de los módulos a descargar. Se prevén dos modos de tratamiento de estos mensajes por parte del decodificador 14. Según el primer modo, el decodificador 14 almacena todos los mensajes que contienen todos los bloques de los módulos a descargar, y los transmite a continuación de una sola vez al procesador de seguridad 16. De acuerdo con el segundo modo, el decodificador 14 transmite al procesador de seguridad 16 los mensajes en cuanto se produzca la recepción. Según la invención, cada módulo de código está asociado a un tratamiento específico, de entre los dos tratamientos, por parte del decodificador 14.

En 42, el procesador de seguridad 16 verifica y descifra los mensajes que contienen los bloques de cada módulo con la ayuda de las propiedades asociadas a dicho módulo. Extrae los bloques del módulo a partir de estos mensajes. A continuación, descifra y verifica la integridad de todos los bloques.

En 44, el procesador de seguridad 16 almacena los bloques de cada uno de los módulos descargados en espacios de la memoria 18 apropiados para cada módulo. Durante el almacenamiento, cada módulo puede disponer además de su propio formato de almacenamiento.

Preferentemente, la integridad de los módulos almacenados está protegida con el fin de garantizar que los mismos no se corromperán durante su ejecución.

Según un modo de realización de la invención, cada módulo dispone de su propia clave y de su propio algoritmo de integridad y procede a verificaciones de integridad en momentos diferentes, por ejemplo, durante un acceso, durante una lectura, durante una ejecución, durante un control, durante una reinicialización, etc.

Además, cada módulo almacenado está protegido preferentemente en cuanto al acceso con el fin de que ningún otro módulo pueda llegar a ejecutarlo sin autorización. Según la invención, cada módulo dispone preferentemente de sus propias reglas de acceso utilizando técnicas de protección convencionales de tipo cortafuegos (“firewall”) o de una protección por medio de MPU/MMU (“Memory Protection Unit/Memory Management Unit”).

En 46, el procesador de seguridad 16 activa los módulos de código descargados, almacenados en la memoria 18, automáticamente o como consecuencia de la recepción de una orden de activación emitida desde el decodificador 14 o desde la cabecera de red 6. Según la invención, cada módulo dispone preferentemente de sus propias reglas de activación.

La figura 3 detalla las etapas que permiten generar los mensajes de actualización del código del procesador de seguridad 16. El código del procesador de seguridad 16 se segmenta en un número determinado N de módulos, constituyendo los N módulos la totalidad del código. En 50, se toma una decisión de actualizar el código del procesador de seguridad 16 con el fin de hacerlo evolucionar o de aportarle correcciones.

En 52, el módulo o los módulos a modificar se identifican a partir de una lista 54 de los N módulos de código.

En 55, los módulos a modificar son desarrollados y compilados. Cada módulo posee un mecanismo de actualización, un lenguaje (por ejemplo, el lenguaje C) y un compilador (por ejemplo, un compilador estándar ANSI) que están dedicados al mismo y que se agrupan en 56.

A título de ejemplos de mecanismos de actualización, se puede considerar:

- una actualización parcial por tablas de direccionamiento indirecto, es decir, que la dirección de una función sea sustituida por una nueva dirección almacenada en la tabla de direccionamiento indirecto; o
- una actualización parcial por sustitución de bloques de código existentes; o incluso,
- una actualización completa del módulo.

La etapa 55 de desarrollo y de compilación da como resultado un código nativo binario 58 de los módulos a modificar.

En 60, la cabecera de red 6 define un identificador de la actualización según un formato apropiado para cada módulo en cuestión y protege el código binario 58 en cuanto a integridad y confidencialidad según propiedades 62 propias de cada módulo.

A título de ejemplos de formato de identificador de la actualización, se puede considerar un identificador sobre 10 caracteres ASCII = MOD1_1.6.9 o un identificador sobre 7 caracteres ASCII = 230MOD2. También es posible no prever ningún identificador de la actualización para ciertos módulos.

A título de ejemplos del algoritmo y de la clave de la protección de confidencialidad, se puede considerar un algoritmo AES CBC con una clave dedicada almacenada de forma no flexible en el código inicial del procesador de seguridad 16 o un algoritmo 3DES CBC. También es posible no prever ninguna protección de confidencialidad del módulo.

A título de ejemplos del algoritmo y de la clave de integridad, se puede considerar una integridad AES-MAC con una clave dedicada o almacenada de forma no flexible en el código inicial del procesador de seguridad 16 o una integridad 3DES MAC con claves comunes para todos los procesadores de seguridad de todos los decodificadores de los abonados. También es posible no prever ninguna protección de integridad del módulo.

El resultado de la etapa 60 es un código 63 de módulos a modificar identificado y cifrado (es decir, protegido en cuanto a confidencialidad e integridad).

En 64, la cabecera de red 6 construye los mensajes de difusión de bloques de los módulos a modificar segmentando el código binario identificado y cifrado 63 en bloques, integrando estos bloques en mensajes EMM y/o ECM y cifrando estos mensajes.

La etapa 64 utiliza propiedades 66 propias de cada módulo. Las propiedades 66 se refieren al tipo de mensajes utilizado (EMM o ECM), a las reglas de direccionamiento (a un solo abonado, a un grupo de abonados o a todos los abonados), a las claves de transporte utilizadas para cifrar los mensajes (por ejemplo, claves específicas poseídas por la sociedad depositaria o bien las mismas claves que las utilizadas para transportar derechos del abonado o bien las claves de explotación), al tamaño de los bloques (por ejemplo, tamaño fijo de 128 o de 200 octetos o tamaño variable) y al formato de encabezamiento de los mensajes (por ejemplo, formato “TLV” que se corresponde con una codificación “Tipo, Longitud, Valor” con T = ’56 y L = ’80, o T= ’57 o T = ‘AO).

La etapa 64 da como resultado una pluralidad de mensajes 68 preparados para ser difundidos por la cabecera de red 6.

La figura 4 detalla las etapas del procedimiento según la invención llevadas a la práctica por el procesador de seguridad 16.

En 70, el procesador de seguridad 16 determina, al producirse la recepción de un mensaje EMM o ECM, si un mensaje de ese tipo contiene bloques de actualización de un módulo de código en función de los tipos de direccionamiento de los diferentes módulos.

En 72, el procesador de seguridad 16 identifica el módulo en cuestión con la ayuda de los parámetros recibidos asociados a este módulo.

Las etapas 70 y 72 utilizan así la lista 54 de los N módulos, y las propiedades 56, 62 y 66 de cada módulo.

El resultado de estas etapas es un conjunto de módulos identificados 74.

En 76, el procesador de seguridad 16 decide a partir de propiedades 78 de recepción y de procesado de cada módulo, si debe procesar el bloque recibido del módulo inmediatamente o si el procesado de los bloques del módulo debe comenzar después de haber recibido la totalidad de los bloques.

La etapa 76 da como resultado un conjunto de bloques de código 80 que pueden ser procesados por el procesador de seguridad 16.

En 82, el procesador de seguridad descifra y almacena los bloques de código del conjunto 80 en espacios dedicados a los diferentes módulos dentro de la memoria 18. Esta etapa 82 utiliza propiedades de almacenamiento 84 de cada módulo.

Las propiedades de almacenamiento 84 se refieren al espacio de memoria dedicado (por ejemplo, una zona de tamaño fijo o variable por reasignación dinámica, o bien ninguna zona dedicada), al formato de almacenamiento (por ejemplo, almacenamiento de los bloques de manera contigua y ordenada o almacenamiento de los bloques en FIFO), a la protección eventual en cuanto a confidencialidad y/o integridad de los bloques del módulo (por ejemplo, utilizando algoritmos de protección de confidencialidad AES CBC o 3DES CBC y de integridad AES MAC o 3DES MAC), a la protección eventual en cuanto a acceso de lectura o lectura/escritura de los bloques del módulo (por ejemplo, por medio de un microprocesador de procesador de seguridad 16 con un acceso limitado al módulo en escritura).

En 86, el procesador de seguridad activa cada uno de los módulos actualizados almacenados en la memoria 18 utilizando propiedades de activación 88 propias de cada módulo. A título de ejemplos, se puede considerar una activación inmediata como consecuencia de la recepción de todos los bloques del módulo o una activación como

consecuencia de la recepción de una orden de activación emitida desde la cabecera de red 6 y contenida en un mensaje EMM o ECM, o bien emitida desde el decodificador 14.

Así, el procedimiento según la invención proporciona una solución de descarga de código de un procesador de seguridad al disponer de elementos diferenciadores para cada etapa definida en función del módulo de código a descargar.

La solución de la invención permite así disponer de un proceso único y centralizado de alto nivel cuya robustez se valida de una vez por todas para una mejor seguridad del procesador de seguridad.

Esta solución también permite mejorar la seguridad al compartimentar el riesgo de un fallo de seguridad. Así, un fallo en una etapa del mecanismo de seguridad puede estar abierto a vulnerabilidades en un módulo de código aunque puede estar cerrado a las mismas en los otros módulos.

La solución de la invención permite además mejorar la seguridad global del procesador de seguridad en el caso de una ingeniería inversa, de uno de los mecanismos diferenciados propuestos. La ingeniería inversa de un mecanismo de seguridad no permite así llevar a cabo la ingeniería inversa de los otros mecanismos.

El procedimiento de la invención comprende los programas de ordenador correspondientes y se aplica ventajosamente en el contexto de la televisión de pago (“Pay TV”) con el fin de mejorar la seguridad y la corrección de los fallos de seguridad de los mecanismos de seguridad llevados a la práctica en el procesador de seguridad (generalmente una tarjeta chip).

Evidentemente, son todavía posibles otros modos de realización y aplicaciones.

REIVINDICACIONES

1. Procedimiento de actualización de un procesador de seguridad (16) para un decodificador (14) apropiado para recibir y desaleatorizar una señal multimedia aleatorizada, siendo dicha señal multimedia difundida por una cabecera de red (6) y comprendiendo dicho procesador (16) una memoria (18) que contiene un código de una aplicación que, cuando es ejecutado o interpretado por un microprocesador, permite ejecutar un conjunto de operaciones necesarias para la desaleatorización de la señal multimedia aleatorizada, comprendiendo dicho procedimiento una etapa de descarga de por lo menos un módulo del código a partir de la cabecera de red (6), caracterizado porque la etapa de descarga del módulo comprende las subetapas de:

- cifrado (32), por parte de la cabecera de red (6), del módulo con la ayuda de una clave y de un algoritmo de cifrado apropiados para dicho módulo;

- segmentación (34, 64), por parte de la cabecera de red (6), del módulo en una pluralidad de bloques de tamaño apropiado para dicho módulo, extraído de un conjunto memorizado de propiedades (66) propias de dicho módulo;

- integración (34, 64), por parte de la cabecera de red (6), de cada uno de los bloques del módulo en un mensaje específico;

- difusión (36), por parte de la cabecera de red (6) hacia el decodificador (14), de cada uno de estos mensajes;

- recepción (38, 70) de estos mensajes por el decodificador (14);

- extracción (42, 72), por parte del procesador de seguridad (16) del decodificador (14), de los bloques del módulo a partir de estos mensajes;

- descifrado (42, 82), por parte del procesador de seguridad (16), de los bloques del módulo;

- almacenamiento (44, 82), por parte del procesador de seguridad (16), de los bloques del módulo en un espacio de la memoria (18) dedicado a dicho módulo; y

- obtención, a partir de estos bloques, de un módulo de código actualizado.

2. Procedimiento de actualización según la reivindicación 1, caracterizado porque la etapa de descarga del módulo comprende además las subetapas de:

- cifrado (34), por parte de la cabecera de red (6), de los mensajes que contienen los bloques del módulo con la ayuda de claves de transporte apropiadas para el módulo; y

- descifrado (42), por parte del procesador de seguridad (16), de los mensajes que contienen los bloques del módulo.

3. Procedimiento de actualización según la reivindicación 1 o 2, caracterizado porque los mensajes que contienen los bloques del módulo se seleccionan de tipo ECM (“Entitlement Control Message”) y/o EMM (“Entitlement Management Message”) según una elección apropiada para el módulo.

4. Procedimiento de actualización según cualquiera de las reivindicaciones anteriores, caracterizado porque comprende además una etapa de activación (46), por parte del procesador de seguridad (16), del módulo descargado almacenado en la memoria (18).

5. Procedimiento de actualización según la reivindicación 4, caracterizado porque la etapa de activación (46) del módulo descargado es activada por el procesador de seguridad (16) automáticamente o como consecuencia de la recepción de una orden de activación emitida desde el decodificador (14) o desde la cabecera de red (6).

6. Procedimiento de actualización según cualquiera de las reivindicaciones anteriores, caracterizado porque el módulo se desarrolla y se compila en un lenguaje y con la ayuda de un compilador apropiados para el módulo.

7. Sistema de actualización de un procesador de seguridad (16) para un decodificador (14) apropiado para recibir y desaleatorizar una señal multimedia aleatorizada, siendo dicha señal multimedia difundida por una cabecera de red (6) y comprendiendo dicho procesador (16) una memoria (18) que contiene un código de una aplicación que, cuando es ejecutado o interpretado por un microprocesador, permite ejecutar un conjunto de operaciones necesarias para la desaleatorización de la señal multimedia aleatorizada, comprendiendo dicho sistema unos medios de descarga de por lo menos un módulo del código a partir de la cabecera de red (6), caracterizado porque los medios de descarga del módulo comprenden:

a nivel de la cabecera de red (6):

- unos medios de cifrado del módulo con la ayuda de una clave y de un algoritmo de cifrado apropiados para dicho módulo;
- unos medios de segmentación del módulo en una pluralidad de bloques de tamaño apropiado para dicho módulo, extraído de un conjunto memorizado de propiedades (66) propias de dicho módulo;
- unos medios de integración de cada uno de los bloques del módulo en un mensaje específico; y
- unos medios de difusión hacia el decodificador (14) de cada uno de estos mensajes

a nivel del decodificador (14), unos medios de recepción de estos mensajes; y

a nivel del procesador de seguridad (16):

- unos medios de extracción de los bloques del módulo a partir de estos mensajes;
- unos medios de descifrado de los bloques del módulo;
- unos medios de almacenamiento de los bloques del módulo en un espacio de la memoria (18) dedicado a dicho módulo; y
- unos medios de obtención, a partir de estos bloques, de un módulo de código actualizado.

8. Programa de ordenador que comprende instrucciones de código, que, cuando este programa es ejecutado en un ordenador, permiten realizar el procedimiento de actualización de un procesador de seguridad (16) según cualquiera de las reivindicaciones 1 a 6.

9. Programa de ordenador según la reivindicación 8, caracterizado porque comprende un subprograma ejecutado por una cabecera de red (6) para permitir realizar las subetapas de la etapa de descarga del módulo que consisten en:

- cifrar el módulo con la ayuda de una clave y de un algoritmo de cifrado apropiados para dicho módulo;
- segmentar el módulo en una pluralidad de bloques de tamaño apropiado para dicho módulo;
- integrar cada uno de los bloques del módulo en un mensaje específico; y
- difundir hacia el decodificador (14) cada uno de estos mensajes.

10. Programa de ordenador según la reivindicación 8 o 9, caracterizado porque comprende un subprograma ejecutado o interpretado por un procesador de seguridad (16) para permitir realizar las subetapas de la etapa de descarga del módulo que consisten en:

- extraer los bloques del módulo a partir de estos mensajes, utilizando un tamaño de bloques apropiado para dicho módulo, extraído de un conjunto memorizado de propiedades (66) propias de dicho módulo;
- descifrar los bloques del módulo;
- almacenar los bloques del módulo en un espacio de la memoria (18) dedicado a dicho módulo; y
- obtener, a partir de estos bloques, un módulo de código actualizado.

11. Procesador de seguridad (16), caracterizado porque es apto para ejecutar o para interpretar el código de un subprograma con el fin de permitir la realización de las subetapas de la etapa de descarga del módulo de acuerdo con la reivindicación 10.

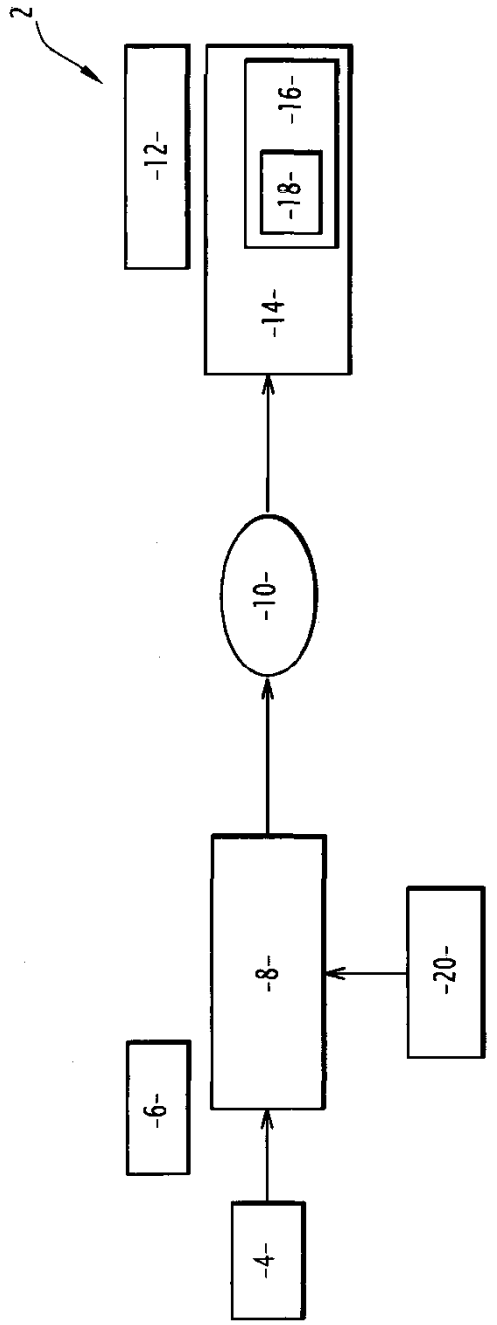

FIG.1

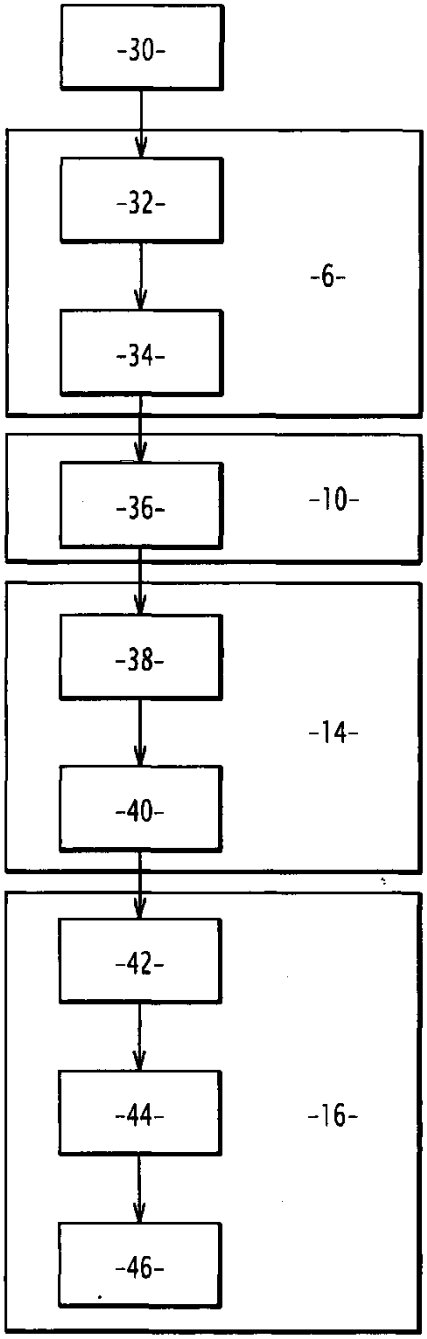

FIG.2

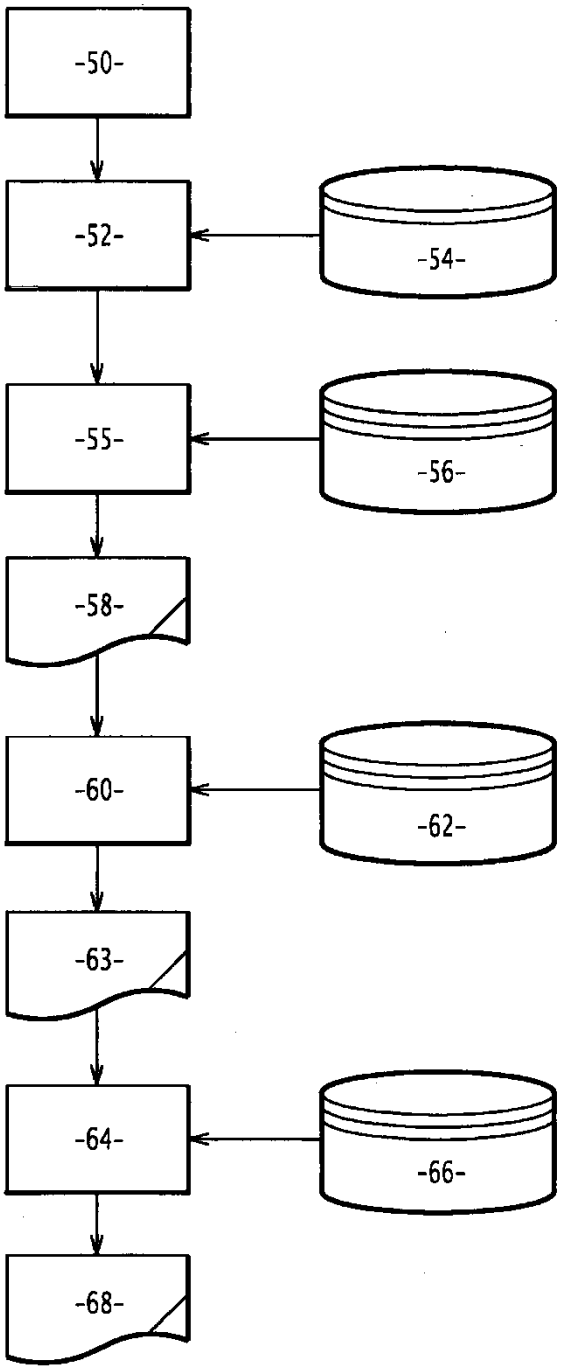

FIG.3

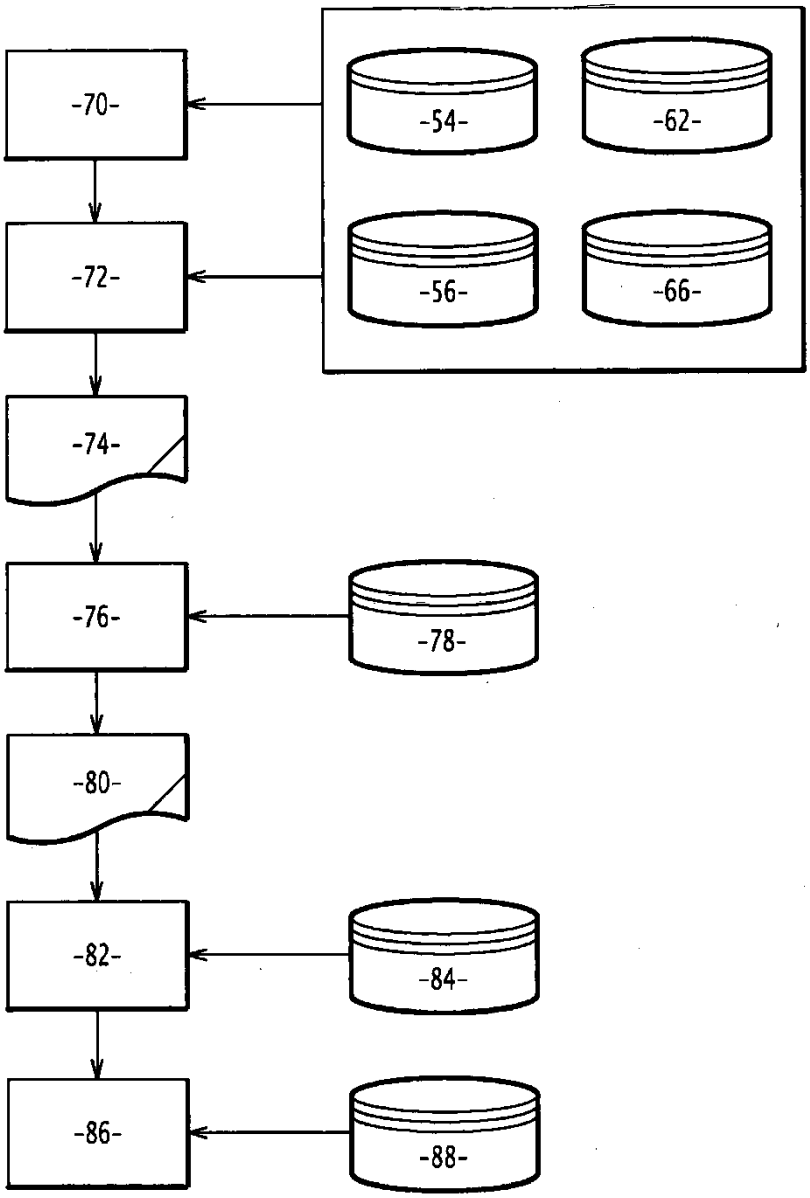

FIG.4