(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 368 580**

(51) Int. CI.:
H04N 7/167 (2011.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Número de solicitud europea: **06760123 .7**
(96) Fecha de presentación: **18.05.2006**
(97) Número de publicación de la solicitud: **1889478**
(97) Fecha de publicación de la solicitud: **20.02.2008**

(54) Título: **CIFRADO/DESCIFRADO DE DATOS DE PROGRAMA PERO NO DE DATOS PSI.**

(30) Prioridad:
25.05.2005 US 137272
30.01.2006 US 342460
30.01.2006 US 342479
30.01.2006 US 343060
31.01.2006 US 342472

(45) Fecha de publicación de la mención BOPI:
18.11.2011

(45) Fecha de la publicación del folleto de la patente:
18.11.2011

(73) Titular/es:
Zenith Electronics LLC
C/O The Corporation Trust Company New Castle County 1209 Orange Street
Wilmington DE 19801, US y
Lewis, Richard

(72) Inventor/es:
HAUGE, Raymond, C. y
TURNER, Rudolf

(74) Agente: **Arias Sanz, Juan**

DESCRIPCIÓN

Cifrado/descifrado de datos de programa pero no de datos PSI.

Solicitudes relacionadas

Esta solicitud se refiere a la solicitud de patente estadounidense número de serie 11/137.272, presentada el 25 de mayo de 2005.

Campo técnico de la invención

La presente invención se refiere al cifrado y descifrado de datos transmitidos entre un transmisor y un receptor y, más particularmente, al cifrado y descifrado tanto de datos como de claves de cifrado usadas para cifrar los datos.

Antecedentes de la invención

Existen numerosos sistemas en los que la copia no autorizada de datos tiene consecuencias indeseables. Por ejemplo, en los sistemas de pago por visión tales como los que ofrecen hoteles, hostales y sistemas de cable, el proveedor que ofrece una programación de pago por visión pierde importantes beneficios si se piratean sus programas.

Están disponibles habitualmente numerosas herramientas en ferreterías, tiendas de artículos para ocio, laboratorios universitarios, y que proporcionan hackers y expertos para permitir la ingeniería inversa de todos los aspectos de los sistemas de transmisión de datos, incluyendo los sistemas de pago por visión. Por consiguiente, los proveedores de pago por visión y otros interesados en la protección frente al copiado implementan diversos sistemas de protección frente al copiado con el fin de impedir un copiado no autorizado.

Los sistemas protección frente al copiado tienen varios objetivos de seguridad. Por ejemplo, los sistemas de protección frente al copiado pretenden impedir el robo de contenido digital comprimido de alta calidad, impedir el robo de contenido digital no comprimido de alta calidad, y limitar las pérdidas provocadas por los accesos indebidos.

La solicitud estadounidense publicada 2004/0268117 A1 describe un transmisor en el que (i) se genera aleatoriamente una clave K_i j, donde i es un índice de clave, (ii) se genera un | número de serie cifrado S_n según | S_n = F(W|| n) donde W es un número ID secreto, n es el número de serie del receptor, || indica concatenación, y F() es una función *hash*, (iii) la clave K_i se cifra para producir una semilla cifrada C_{ni} según $C_{ni} = K_x$ XOR F (S_n || i), y (iv) la semilla cifrada C_{ni}, el índice i, el número de serie n se transmiten en un flujo de autorización al receptor. El transmisor también usa la clave K_i para cifrar datos y transmite los datos cifrados al receptor. Un receptor usa S_n, que se ha calculado previamente y almacenado en el receptor, y el índice i recibido para descifrar la semilla cifrada recibida C_{ni} para así recuperar la clave K_i según $K_i = C_{ni}$ XOR F(S_n || i). El receptor descifra entonces los datos cifrados recibidos usando la clave K_A recuperada.

La solicitud de patente europea 1187483 A2 describe una disposición en la que se reciben datos cifrados a través de un canal de transmisión de datos, se reciben claves a través de un canal de transmisión de claves, y se reciben mapas a través de un canal de sincronización de bloques 78. Los mapas indican qué claves deben usarse para descifrar datos en qué tramas de datos. El receptor usa los mapas recibidos para determinar cuál de las claves recibidas debe usarse para descifrar datos en cuál de las tramas de datos recibidas, y el receptor usa las claves recibidas para descifrar los datos cifrados en las correspondientes tramas de datos recibidas.

La solicitud de patente europea 0706118 A1 describe una disposición en la que un proveedor de software cifra un programa P y el usuario del software lo descifra. Se genera una primera clave de cifrado Kup según un algoritmo secreto y un ID de usuario o programa. La primera clave de cifrado Kup se usa por un primer algoritmo de cifrado para cifrar una segunda clave de cifrado r que se genera de manera arbitraria para producir así una segunda clave de cifrado Er cifrada. Una tercera clave de cifrado K2 se genera de manera arbitraria. La segunda clave de cifrado r se usa por un segundo algoritmo de cifrado para cifrar la tercera clave de cifrado K2 para producir así una tercera clave de cifrado K2' cifrada. El programa P se cifra mediante la tercera clave de cifrado K2 para producir así un programa cifrado P'. La segunda clave de cifrado Er cifrada, la tercera clave de cifrado K2' cifrada y el programa cifrado P' se proporcionan al usuario del software quien entonces descifra el programa cifrado P'.

El sistema de protección frente al copiado de la presente invención pretende frustrar el copiado no autorizado de contenido.

Sumario de la invención

La presente invención proporciona un método, implementado por un receptor, de descifrado de datos cifrados tal como se expone en la reivindicación 1.

Las realizaciones preferidas se definen mediante las reivindicaciones dependientes.

Breve descripción de los dibujos

Estas y otras características y ventajas resultarán más evidentes a partir de una consideración detallada de la invención tomada en conjunción con los dibujos, en los que:

la figura 1 ilustra un codificador de cifrado de un transmisor con protección frente al copiado según una realización de la presente invención;

la figura 2 ilustra el bloque de cifrado de datos de la figura 1 en mayor detalle;

la figura 3 ilustra el bloque de claves dinámicas de la figura 1 en mayor detalle;

la figura 4 ilustra el bloque de expansión de claves de la figura 3 en mayor detalle;

la figura 5 ilustra partes de la figura 1 en mayor detalle;

la figura 6 ilustra el modificador de claves de la figura 5 en mayor detalle;

la figura 7 ilustra un mensaje modificador MM de ejemplo usado en el sistema de protección frente al copiado de la figura 1;

la figura 8 ilustra una parte de control del mensaje modificador MM ilustrado en la figura 7;

la figura 9 ilustra una definición de ejemplo de los bytes de control de sistema del mensaje modificador MM ilustrado en la figura 8;

la figura 10 ilustra un segmento de mensaje MS de ejemplo usado en el sistema de protección frente al copiado de la figura 1;

la figura 11 ilustra el bloque de cifrado de la clave de programa, la clave de modificación y el mensaje modificador MM de la figura 1 en mayor detalle;

la figura 12 ilustra un mensaje de clave de ejemplo que forma parte del segmento de mensaje MS ilustrado en la figura 10;

la figura 13 ilustra un par de segmentos de mensaje MS de ejemplo usado para transmitir claves de programa y claves de modificación;

la figura 14 ilustra el sincronismo del transmisor y el receptor con respecto a la generación y uso de mensajes;

la figura 15 ilustra una rotación de ejemplo para aplicar las claves de programa PK durante el cifrado de datos de programa;

la figura 16 ilustra un ejemplo de las partes de un segmento de datos de programa de un campo al que se aplica la rotación;

la figura 17 ilustra un decodificador de descifrado de un receptor de protección frente al copiado según una realización de la presente invención;

la figura 18 ilustra el bloque de descifrado de datos de la figura 17 en mayor detalle;

la figura 19 ilustra partes del decodificador de descifrado de la figura 17 en mayor detalle; y,

la figura 20 ilustra el bloque de descifrado de la clave y el mensaje modificador de la figura 17 en mayor detalle.

Descripción detallada

En la figura 1, un codificador de cifrado 8 de ejemplo de un transmisor con protección frente al copiado incluye un filtro PID 10 que recibe un flujo de transporte MPEG y que determina qué paquetes en el flujo de transporte MPEG contienen datos que deben cifrarse. Tal como se comenta más adelante, el filtro PID 10 también identifica paquetes nulos que deben sustituirse con segmentos de mensaje MS que dan al receptor suficiente información para descifrar los datos de programa cifrados en la señal recibida, y el filtro PID 10 identifica además paquetes que contienen información que no debe cifrarse.

Un generador dinámico de claves de programa y claves de modificación 12 genera dinámicamente claves de programa PK que se aplican por un primer motor de cifrado 14 con el fin de cifrar los datos de programa en el flujo de transporte MPEG que se ha seleccionado para el cifrado. El primer motor de cifrado 14, por ejemplo, puede ser un motor de cifrado de una única envoltura, y puede disponerse para aplicar el proceso de cifrado de una única envoltura especificado en la norma Advanced Encryption Standard (AES). Los paquetes de datos de programa cifrados se suministran a una entrada de un multiplexador de salida 16.

Las claves de programa PK generadas de manera dinámica se aplican a través de un multiplexador 24 tras lo cual ellas mismas se cifran por un segundo motor de cifrado 18. El segundo motor de cifrado 18 puede ser un motor de cifrado de triple envoltura, y puede disponerse para aplicar el proceso de cifrado de triple envoltura especificado en la norma Advanced Encryption Standard.

A diferencia de las claves de programa PK generadas de manera dinámica que se usan por el primer motor de cifrado 14 para cifrar los datos de programa, las claves usadas por el segundo motor de cifrado 18 para cifrar las claves de programa PK generadas de manera dinámica son claves de segmentos de mensaje. En una memoria 20 están almacenadas claves fijas, estas claves fijas se usan por un control y generador de claves de segmentos de mensaje 22 para generar claves de segmentos de mensaje, y las claves de segmentos de mensaje se suministran al segundo motor de cifrado 18.

Las claves fijas almacenadas en la memoria 20 tienen, por ejemplo, 128 bits de longitud, y hay, por ejemplo, sesenta y cuatro claves fijas almacenadas en la memoria 20. Los valores *hash* comentados en el presente documento son, por ejemplo, de sesenta y cuatro bits cada uno y se derivan como partes seleccionadas de las claves fijas. Alternativamente, pueden almacenarse por separado valores *hash* en la memoria 20, y las claves fijas y los valores *hash* pueden ser de cualquier longitud y número deseado.

Por tanto, el control y generador de claves de segmentos de mensaje 22 selecciona las claves fijas que deben usarse por el segundo motor de cifrado 18 a partir de la memoria 20, las usa para generar claves de segmentos de mensaje, y suministra las claves de segmentos de mensaje al segundo motor de cifrado 18. El segundo motor de cifrado 18 cifra las claves de programa PK generadas de manera dinámica basándose en las claves de segmentos de mensaje del control y generador de claves de segmentos de mensaje 22.

Tal como se comenta más adelante, también se aplican un mensaje modificador MM y claves de modificación MK a través del multiplexador 24 y se cifran por el segundo motor de cifrado 18. Las claves de programa PK generadas de manera dinámica cifradas y el mensaje modificador MM cifrado se ensamblan en segmentos de mensaje de clave de programa PKMS que se reenvían al receptor. Tal como se comentará adicionalmente más adelante, las claves de modificación MK cifradas, una suma de comprobación cifrada, y el mensaje modificador MM cifrado se ensamblan de forma similar en segmentos de mensaje de clave de modificación MKMS que también se reenvían al receptor.

Las claves de modificación, generadas de manera dinámica por el generador de claves de programa y de modificación 12, se usan con las claves fijas para generar las claves de segmentos de mensaje que se usan para cifrar las claves de programa, y la suma de comprobación se basa en las claves fijas almacenadas en la memoria 20. La suma de comprobación, por ejemplo, puede comprender 128 bits, y puede generarse a partir de todas las claves fijas almacenadas en la memoria 20. Por consiguiente, el receptor puede comparar la suma de comprobación desde el transmisor con una suma de comprobación generada a partir de sus propias claves fijas para comprobar que sus claves fijas coinciden con las claves fijas del transmisor. La suma de comprobación también puede usarse para determinar errores en la transmisión.

Tal como se indicó anteriormente, el segmento de mensaje de clave de programa PKMS y el segmento de mensaje de clave de modificación MKMS dan al receptor la información que requiere para descifrar los datos de programa cifrados en la señal recibida.

La figura 2 muestra el primer motor de cifrado 14 en mayor detalle. Tal como se muestra en las figuras 1 y 2, el primer motor de cifrado 14 está acoplado entre el filtro PID 10 y el multiplexador de salida 16.

El primer motor de cifrado 14 tiene tres secciones 14A, 14B y 14C. La sección 14A incluye un demultiplexador 30, memorias 32 y 34, y un multiplexador 36. La sección 14B incluye un retardo RAM 38, un bloque de cifrado 40, y un multiplexador 42. La sección 14C incluye un demultiplexador 44, memorias 46 y 48, y un multiplexador 50.

El filtro PID 10 pasa los paquetes de transporte en el flujo de transporte MPEG al demultiplexador 30. Los paquetes de transporte se demultiplexan y se almacenan en las memorias 32 y 34 que operan a modo de ping-pong. Los paquetes de transporte en las memorias 32 y 34 se suministran al multiplexador 36.

El multiplexador 36 pasa todos los paquetes desde las memorias 32 y 34 tanto al retardo RAM 38 como al bloque de cifrado 40. Estos paquetes incluyen paquetes de programa, paquetes nulos y paquetes no de programa tales como los PID, PSIP, PMT y PAT. El bloque de cifrado 40 usa las claves de programa PK generadas de manera dinámica para cifrar todos los paquetes que recibe y suministra los paquetes cifrados al multiplexador 42. En respuesta a un indicador de cifrado desde el filtro PID 10, el multiplexador 42 selecciona sólo los paquetes cifrados del bloque de cifrado 40 que corresponden al programa o programas seleccionados que deben cifrarse. Se entenderá que el flujo de transporte MPEG puede contener uno o más programas y que uno cualquiera o más de estos programas pueden estar indicados para el cifrado. Todos los demás paquetes (aquéllos que no corresponden al programa que va a cifrarse) se seleccionan por el multiplexador 42 desde el retardo RAM 38. Por tanto, la salida del multiplexador 42 es el flujo de transporte MPEG de entrada salvo porque los paquetes correspondientes al programa seleccionado están cifrados. El multiplexador 42 pasa los paquetes cifrados y no cifrados al demultiplexador 44.

Entre los paquetes que el multiplexador 42 selecciona del retardo RAM 38 se encuentran paquetes PSI (*Program Specific Information*, información específica de programa) tales como paquetes PSIP (*Program and System Information Protocol*, protocolo de información de programa y sistema), paquetes PAT (*Program Association Table*, tabla de asociación de programas), paquetes PMT (*Program Map Table*, tabla de mapa de programas) y/o paquetes que contienen campos de adaptación. Por consiguiente, los paquetes no cifrados emitidos por el multiplexador 42 incluyen paquetes PSI. Los paquetes PSI contienen información que ayuda al receptor a determinar qué canal y qué partes del flujo de transporte contienen el programa seleccionado por el usuario. Si los paquetes PSI estuvieran cifrados, el receptor no podría localizar los programas seleccionados por usuario para su descifrado.

Los paquetes cifrados y no cifrados desde el demultiplexador 44 se almacenan en las memorias 46 y 48 que operan a modo de ping-pong. Los paquetes cifrados y no cifrados en las memorias 46 y 48 se suministran a través del multiplexador 50 al multiplexador de salida 16.

Las secciones 14A y 14C del primer motor de cifrado 14 se controlan para que mantengan un sincronismo, unas tasas de flujo de datos y una sincronización apropiados.

La figura 3 muestra una parte dinámica del generador de claves de programa 12A del generador dinámico de claves de programa y claves de modificación 12 en más detalle. La parte dinámica del generador de claves de programa 12A incluye un generador de semillas 60 que suministra una semilla a un generador de números aleatorios 62. Por ejemplo, el generador de semillas 60 puede seleccionar, según se desee, la semilla a partir de cualquier parte del flujo de transporte MPEG 61, tal como vídeo y/o audio, en uno o más paquetes de datos de programa.

Un demultiplexador 64 selecciona cuatro números aleatorios de 128 bits desde el generador de números aleatorios 62 y almacena estos cuatro números aleatorios de 128 bits como cuatro claves de programa generadas de manera dinámica en una siguiente parte de una memoria 66 mientras que el bloque de cifrado 40 usa las cuatro claves de programa generadas de manera dinámica previamente almacenadas en una parte activa de la memoria 66 para cifrar datos de programa. Por tanto, mientras que las cuatro claves de programa PK generadas de manera dinámica almacenadas en la parte activa de la memoria 66 están usándose actualmente para cifrar datos de programa, el demultiplexador 64 selecciona otros cuatro números aleatorios de 128 bits desde el generador de números aleatorios 62 y almacena estos cuatro números aleatorios de 128 bits adicionales como cuatro claves de programa PK generadas de manera dinámica en la siguiente parte de la memoria 66.

Tal como se explica más adelante en relación con la figura 14, en el momento que se transmite un segmento de mensaje de clave de modificación MKMS, se interrumpe el uso de la cuatro claves de programa PK generadas de manera dinámica almacenadas en la parte activa de la memoria 66, y empieza el uso de las cuatro nuevas claves de programa PK generadas de manera dinámica almacenadas en la siguiente parte de la memoria 66. En este punto de transición, la antigua siguiente parte de la memoria 66 pasa a ser la nueva parte activa de la memoria 66, y la antigua parte activa de la memoria 66 pasa a ser la nueva siguiente parte de la memoria 66. Además, mientras que estas cuatro nuevas claves de programa PK generadas de manera dinámica están usándose para cifrar datos de programa, se generan dinámicamente cuatro claves de programa PK más y se almacenan en la nueva siguiente parte de la memoria 66.

Un multiplexador 68 suministra las cuatro claves de programa dinámicas desde la parte activa de la memoria 66 a un expansor de claves 70 tal como el mostrado en la figura 4. Según sea necesario, el expansor de claves 70 expande cada una de las claves de programa PK dinámicas desde claves de 128 bits hasta, por ejemplo, claves expandidas de 1408 bits. Las claves de programa PK dinámicas expandidas se suministran al bloque de cifrado 40 de la figura 2.

El expansor de claves 70 tal como se muestra en la figura 4 incluye un bloque de clave inversa. El bloque de clave inversa se habilita durante el cifrado de programa y se deshabilita durante el cifrado del segmento de mensaje de clave de programa PKMS y el segmento de mensaje de clave de modificación MKMS.

De esta manera, se usan cuatro claves de programa PK generadas de manera dinámica para cifrar datos de programa mientras que las siguientes cuatro claves de programa PK están generándose dinámicamente. Las cuatro claves de programa PK generadas de manera dinámica que están usándose desde la parte activa de la memoria 66 continúan usándose hasta que se genera el segmento de mensaje de clave de modificación MKMS.

El tiempo entre segmentos de mensaje, por ejemplo, puede hacerse que dependa de la disponibilidad de paquetes nulos en el flujo de transporte MPEG entrante porque se transmiten segmentos de mensaje en lugar de paquetes nulos seleccionados. El filtro PID 10 detecta el paquete nulo e indica al multiplexador de salida 16 que pase un segmento de mensaje en lugar de paquetes desde el multiplexador 50.

Tal como se muestra en la figura 5, un selector de claves fijas 80 usa números aleatorios generados por el generador de números aleatorios 62 con el fin de acceder a la memoria 20 para seleccionar las claves fijas de la memoria 20. Por ejemplo, cada clave fija almacenada en la memoria 20 puede ser de 128 bits, y pueden usarse cuatro palabras de dirección de 32 bits para leer cada clave fija de la memoria 20. Estas claves fijas se usan para cifrar las claves de programa y las claves de modificación (descritas en más detalle más adelante en el presente documento) que se envían al receptor y que el receptor requiere para descifrar los datos de programa cifrados

recibidos.

Más específicamente, se seleccionan tres claves fijas de la memoria 20 por el selector de claves fijas 80 y se almacenan como claves fijas K_A en una memoria de claves fijas 82. Se seleccionan tres claves fijas más de la memoria 20 por el selector de claves fijas 80 y se almacenan como claves fijas K_B en una memoria de claves fijas 84. Por ejemplo, cada una de estas tres claves fijas K_A y tres claves fijas K_B puede ser de 128 bits de longitud. Las tres claves fijas K_A almacenadas en la memoria de claves fijas 82 y las tres claves fijas K_B almacenadas en la memoria de claves fijas 84 se seleccionan basándose en direcciones aleatorias desde el generador de números aleatorios 62.

Además, se seleccionan tres valores *Hash* A, B y C por el selector de claves fijas 80 y se almacenan en una memoria de claves de segmentos de mensaje y valores *hash* 86. Los tres valores *Hash* A, B y C también se seleccionan basándose en direcciones aleatorias desde el generador de números aleatorios 62. Por ejemplo, cada uno de los tres valores *Hash* A, B y C puede ser de 64 bits o 1/2 de una clave fija. Además, se almacenan tres números aleatorios desde el generador de números aleatorios 62 en una memoria de claves de modificación 88 como claves de modificación K_M. Cada una de las claves de modificación, por ejemplo, puede ser de 128 bits de longitud.

Un generador de claves de segmentos de mensaje 90, que se muestra en más detalle en la figura 6, incluye *latches* 92_1, 92_2 y 92_3 y una tabla de consulta 94 de 96 x 32. El *latch* 92_1 retiene los primeros 32 bits de una primera de las tres claves fijas K_A almacenadas en la memoria de claves fijas 82, el *latch* 92_2 retiene los primeros 32 bits de una primera de las tres claves fijas K_B almacenadas en la memoria de claves fijas 84, y el *latch* 92_3 retiene los primeros 32 bits de una primera de las tres claves de modificación K_M almacenadas en la memoria de claves de modificación 88. Estos 96 bits retenidos forman una dirección de 96 bits que extrae mediante lectura los primeros 32 bits de una primera clave de segmento de mensaje para su almacenamiento en la memoria de claves de segmentos de mensaje y valores *hash* 86.

La figura 6 muestra también, en forma simplificada, cuatro de las tablas de consulta que están almacenadas en la tabla de consulta 94. Una de las tablas se selecciona para proporcionar las tres claves de segmentos de mensaje que están almacenadas en la memoria de claves de segmentos de mensaje y valores *hash* 86. La forma simplificada de la tabla 0 en la figura 6 muestra la relación entre la dirección y los bits que están almacenados en la tabla 0. Por tanto, si el primer bit K_M de una dirección es 0 y e primer bit K_A de una dirección es 0 y el primer bit K_B de una dirección es 0, la tabla 0 extraerá mediante lectura un bit 0 para el primer bit K_0 de una clave de segmento de mensaje. Sin embargo, si el primer bit K_M de una dirección es 1 y el primer bit K_A de una dirección es 1 y el primer bit K_B de una dirección es 0, la tabla 0 extraerá mediante lectura en lugar de ello un bit 1 para el primer bit K_0 de una clave de segmento de mensaje. Si el siguiente bit K_M de una dirección es 0 y el siguiente bit K_A de una dirección es 0 y el siguiente bit K_B de una dirección es 0, la tabla 0 extraerá mediante lectura un bit 0 para el siguiente bit K_0 de la clave de segmento de mensaje. Sin embargo, si el siguiente bit K_M de una dirección es 0 y el siguiente bit K_A de una dirección es 1 y el siguiente bit K_B de una dirección es 0, la tabla 0 extraerá mediante lectura en lugar de ello un bit 1 para el siguiente bit K_0 de una clave de segmento de mensaje.

Los bits que están almacenados en las tablas pueden tener cualquier relación deseada respecto a sus direcciones. La relación puede ser una relación aleatoria, O, XOR, Y, NAND, NOT, MUX, de complemento a uno, de complemento a dos, o de escala de grises, y cada tabla puede tener una relación diferente entre la dirección y los bits almacenados.

Una vez que los primeros 32 bits de la primera clave de segmento de mensaje se han extraído mediante lectura de la tabla de consulta 94 y se han almacenado en la memoria de claves de segmentos de mensaje y valores *hash* 86, el *latch* 92_1 retiene los segundos 32 bits de la primera de las tres claves fijas K_A almacenadas en la memoria de claves fijas 82, el *latch* 92_2 retiene los segundos 32 bits de la primera de las tres claves fijas K_B almacenadas en la memoria de claves fijas 84, y el *latch* 92_3 retiene los segundos 32 bits de la primera de las tres claves de modificación K_M almacenadas en la memoria de claves de modificación 88. Estos 96 bits retenidos forman una segunda dirección de 96 bits que extrae mediante lectura los segundos 32 bits de la primera clave de segmento de mensaje para su almacenamiento en la memoria de claves de segmentos de mensaje y valores *hash* 86.

Los terceros y cuartos 32 bits de la primera de las tres claves fijas K_A almacenadas en la memoria de claves fijas 82, de la primera de las tres claves fijas K_B almacenadas en la memoria de claves fijas 84, y de la primera de las tres claves de modificación K_M almacenadas en la memoria de claves de modificación 88 se usan para extraer mediante lectura los terceros y cuartos 32 bits de la primera clave de segmento de mensaje a partir de la tabla de consulta 94. Estos terceros y cuartos 32 bits de la primera clave de segmento de mensaje también se almacenan en la memoria de claves de segmentos de mensaje y valores hash 86 para formar todos los 128 bits de la primera clave de segmento de mensaje. Las claves de segmentos de mensaje segunda y tercera se extraen mediante lectura de forma similar de la tabla de consulta 94 y se almacenan en la memoria de claves de segmentos de mensaje y valores *hash* 86. Estas tres claves de segmentos de mensaje se usan para cifrar las claves de programa. Otras tres claves de segmentos de mensaje se usan para cifrar un conjunto de claves de modificación tal como se explica en más detalle más adelante.

Tal como se muestra en la figura 5, un multiplexador 96 multiplexa de manera apropiada las cuatro siguientes claves de programa PK generadas de manera dinámica de la memoria 66, un control de claves 98, las claves de modificación de la memoria de claves de modificación 88, la suma de comprobación de la memoria 20 y un mensaje modificador MM de una memoria de mensajes modificadores 99 para crear el segmento de mensaje de clave de programa PKMS y el segmento de mensaje de clave de modificación MKMS que se comentan con más exhaustividad más adelante.

Un ejemplo del mensaje modificador MM se muestra en la figura 7. Tal como se muestra, el mensaje modificador MM contiene un valor inicial de 64 bits y un control de 192 bits. El uso del valor inicial se describe a continuación. Tal como se muestra en la figura 8, los bits de control del mensaje modificador MM comprenden, por ejemplo, cuatro bytes para el control de sistema, nueve bytes para punteros de dirección que apuntan a las direcciones de memoria para las claves fijas y valores *hash*, y once bytes que pueden usarse para cualquier fin.

Los punteros de dirección comentados anteriormente apuntan a las direcciones en la memoria 20 correspondientes a (i) las seis claves fijas que se almacenan en las memorias de claves fijas 82 y 84 y que, en combinaciones seleccionadas, se usan por el generador de claves de segmentos de mensaje 90 para generar las claves de segmentos de mensaje A, B y C almacenadas en la memoria de claves de segmentos de mensaje y valores hash 86 y (ii) los valores *hash* A, B y C que también se almacenan en la memoria de claves de segmentos de mensaje y valores *hash* 86. Estos punteros de dirección se envían en el mensaje modificador MM al receptor de modo que el receptor puede regenerar las claves de segmentos de mensaje A, B y C y los correspondientes valores *hash* A, B y C que se requieren para descifrar las claves de programa y claves de modificación, tal como se explica a continuación.

Los 32 bits del control de sistema del mensaje modificador MM se muestran a modo de ejemplo en la figura 9. Los bits 0 y 1 se usan para designar el control de copias asignado a los datos de programa. Los bits 2-7 están reservados, excepto que al menos uno de estos bits reservados se fije con un valor para indicar que el segmento de mensaje correspondiente es un segmento de mensaje de clave de modificación MKMS y se fije con otro valor para indicar que el segmento de mensaje correspondiente es un segmento de mensaje de clave de programa PKMS.

Cuando este al menos un bit reservado se fija con un valor que indica que el segmento de mensaje correspondiente es un segmento de mensaje de clave de modificación MKMS, los bits K_M proporcionados en la tabla de consulta 94 se fijan con un valor predeterminado tal como todo ceros mientras que se están produciendo las tres claves de segmentos de mensaje para su almacenamiento en la memoria de claves de segmentos de mensaje y valores *hash* 86. De hecho, las claves de segmentos de mensaje que se usan para cifrar el segmento de mensaje de clave de modificación MKMS se producen con claves de modificación que tienen un valor predeterminado conocido tanto para el transmisor como el receptor.

Cuando las claves de modificación tienen este valor predeterminado, la tabla de consulta 94 puede pasar sólo las claves fijas K_A como las claves de segmentos de mensaje. Alternativamente, cuando las claves de modificación tienen este valor predeterminado, la tabla de consulta 94 podría en cambio pasar sólo las claves fijas K_B como las claves de segmentos de mensaje, o la tabla de consulta 94 podría leer claves de segmentos de mensaje basándose en las claves fijas tanto K_A como K_B de las memorias de claves fijas 82 y 84. Estas alternativas se basan en cuál de las tablas en la tabla de consulta 94 se selecciona tal como se indica por los bits 8-11 del control de sistema del mensaje modificador MM tal como se comenta a continuación. Las claves de segmentos de mensaje producidas con estas claves de modificación que tienen el valor predeterminado se usan para cifrar los mensajes de clave de modificación MK1, MK2 y MK3 y el mensaje de suma de comprobación CRC.

Cuando este al menos un bit reservado se fija con el valor que indica que el segmento de mensaje correspondiente es un segmento de mensaje de clave de programa PKMS, los bits K_M proporcionados a la tabla de consulta 94 son las claves de modificación generadas aleatoriamente almacenadas en la memoria de claves de modificación 88, y estas claves de modificación generadas aleatoriamente se usan junto con las claves fijas K_A y K_B para producir las tres claves de segmentos de mensaje almacenadas en la memoria de claves de segmentos de mensaje y valores *hash* 86. Por tanto, las claves de segmentos de mensaje que se usan para cifrar el segmento de mensaje de clave de programa PKMS se producen con las claves de modificación generadas aleatoriamente almacenadas en la memoria de claves de modificación 88 además de las claves fijas K_A y K_B de las memorias de claves fijas 82 y 84. Las claves de segmentos de mensaje producidas con las claves de modificación generadas aleatoriamente almacenadas en la memoria de claves de modificación 88 se usan para cifrar los mensajes de clave de programa PK1, PK2, PK3 y PK4.

Las claves fijas usadas para generar las claves de segmentos de mensaje que cifran el segmento de mensaje de clave de programa PKMS pueden ser iguales o diferentes de las claves fijas usadas para generar las claves de segmentos de mensaje que cifran el segmento de mensaje de clave de modificación MKMS.

Los bits 8, 9, 10 y 11 designan cuál de las dieciséis tablas posibles almacenadas en la tabla de consulta 94 se usa para producir las claves de segmentos de mensaje almacenadas en la memoria de claves de segmentos de mensaje y valores *hash* 86.

Los bits 12-15 pueden usarse para cualquier fin tal como indicar al receptor una rotación de claves de programa particular, tal como se comenta a continuación.

Los bits 16-31 son una suma de comprobación producida por un generador de CRC de la memoria de mensajes modificadores 99. Específicamente, el generador de CRC de la memoria de mensajes modificadores 99 aplica un código de CRC a los bits 0-15 del byte de control de sistema mostrado en la figura 9 con el fin de generar una suma de comprobación. Esta suma de comprobación comprende los bits 16-31 tal como se muestra en la figura 9. El generador de CRC agrega esta suma de comprobación a los bits 0-15 no modificados para formar el control de sistema completo del mensaje modificador MM. Este control de sistema completo del mensaje modificador MM se usa por el receptor para determinar si el segmento de mensaje de clave de programa PKMS y/o el segmento de mensaje de clave de modificación MKMS no se recibe de manera apropiada debido, por ejemplo, a ruido en el canal y se describe en más detalle a continuación.

Tal como se muestra en la figura 5, un multiplexador 100 recibe las claves de segmentos de mensaje y los valores *hash* almacenados en la memoria de claves de segmentos de mensaje y valores *hash* 86. El multiplexador 100 también recibe tres claves fijas A', B' y C' y tres valores *Hash* A', B' y C' almacenados en una memoria 102. Por ejemplo, cada una de las tres claves fijas A', B' y C' almacenadas en la memoria 102 comprende una clave fija de 128 bits, y cada uno de los tres valores *Hash* A', B' y C' almacenados en la memoria 102 comprende un valor Hash de 64 bits.

Los multiplexadores 96 y 100 funcionan junto con el segundo motor de cifrado 18 para cifrar la parte cifrada de los segmentos de mensaje MS mostrados en la figura 10. En el caso del segmento de mensaje de clave de programa PKMS, la parte cifrada del segmento de mensaje MS mostrada en la figura 10 incluye el mensaje modificador MM y cuatro mensajes de clave de programa KM1, KM2, KM3 y KM4. En el caso del segmento de mensaje de clave de modificación MKMS, la parte cifrada del segmento de mensaje MS mostrada en la figura 10 incluye el mensaje modificador MM, los tres mensajes de clave de modificación MK1, MK2 y MK3, y la suma de comprobación de claves fijas CRC. Los mensajes modificadores MM incluyen el valor inicial y el control de 192 bits tal como se muestra en las figuras 7 y 8. El valor inicial, por ejemplo, puede incluir 64 bits arbitrarios predeterminados.

Con el fin de cifrar el mensaje modificador MM, el multiplexador 100 pasa las tres claves fijas A', B' y C' y los tres valores *Hash* A', B' y C' desde la memoria 102 a través de un expansor de claves 104 hasta el segundo motor de cifrado 18. El expansor de claves 104, por ejemplo, puede ser similar al expansor de claves 70 y expande sólo las claves fijas A', B' y C'. El expansor de claves 104 no expande los valores *Hash* A', B' y C'. Además, el multiplexador 96 pasa el mensaje modificador MM al segundo motor de cifrado 18.

El segundo motor de cifrado 18 se muestra en más detalle en la figura 11. El valor *Hash* A' se aplica a un EXCLUSIVE OR (O EXCLUSIVO) 106, el valor *Hash* B' se aplica a un EXCLUSIVE OR 108, y el valor *Hash* C' se aplica a un EXCLUSIVE OR 110. Los EXCLUSIVE OR 106, 108 y 110 procesan por bits sus respectivas entradas. La clave fija expandida A' se aplica a un cifrador AES 112, la clave fija expandida B' se aplica a un cifrador AES 114, y la clave fija expandida C' se aplica a un cifrador AES 116.

El valor inicial del mensaje modificador MM se aplica al EXCLUSIVE OR 106, un primer 1/3 de los bits de control del mensaje modificador MM se aplica al cifrador AES 112, un segundo 1/3 de los bits de control del mensaje modificador MM se aplica al cifrador AES 114, y un tercer 1/3 de los bits de control del mensaje modificador MM se aplica al cifrador AES 116.

El cifrador AES 112 cifra una salida del EXCLUSIVE OR 106 y el primer 1/3 de los bits de control del mensaje modificador MM según la clave fija expandida A', y suministra la mitad del resultado de cifrado al EXCLUSIVE OR 108 y la otra mitad como el segundo 1/4 del mensaje modificador MM cifrado. El cifrador AES 114 cifra una salida del EXCLUSIVE OR 108 y el segundo 1/3 de los bits de control del mensaje modificador MM según la clave fija expandida B', y suministra la mitad del resultado de cifrado al EXCLUSIVE OR 110 y la otra mitad como el tercer 1/4 del mensaje modificador MM cifrado. El cifrador AES 116 cifra una salida del EXCLUSIVE OR 110 y el tercer 1/3 de los bits de control del mensaje modificador MM según la clave fija expandida C', y suministra la mitad del resultado de cifrado como el primer 1/4 del mensaje modificador MM cifrado y la otra mitad como el cuarto 1/4 del mensaje modificador MM cifrado.

Cada mensaje de clave en el segmento de mensaje de clave de programa PKMS tiene la construcción de ejemplo de la figura 12. Según este ejemplo, un mensaje de clave de programa KM1 incluye un valor inicial de 64 bits, que puede ser el mismo valor inicial comentado anteriormente o un valor inicial diferente, un control de claves 98 de 64 bits, y una de las claves de programa de 128 bits dividida en dos partes de 64 bits. Los mensajes de clave de programa KM2, KM3 y KM4 que contienen las otras tres claves de programa están construidos de manera similar.

El control de claves 98 se usa para designar si el mensaje de clave contiene una clave de programa, una clave de modificación, o la suma de comprobación.

Con el fin de cifrar el mensaje de clave de programa KM1, el multiplexador 100 pasa las tres claves de segmentos de mensaje A, B y C y los tres valores *Hash* A, B y C de la memoria de claves de segmentos de mensaje y valores

hash 86 a través del expansor de claves 104 hasta el segundo motor de cifrado 18. Tal como se explicó anteriormente, las tres claves de segmentos de mensaje A, B y C que se usan para cifrar los mensajes de clave de programa son las claves de segmentos de mensaje leídas de la tabla 94 mediante el uso de las claves de modificación generadas aleatoriamente K_M almacenadas en la memoria de claves de modificación 88, las claves fijas K_A de la memoria de claves fijas 82, y las claves fijas K_B de la memoria de claves fijas 84. El expansor de claves 104 expande sólo las claves de segmentos de mensaje A, B y C. El expansor de claves 104 no expande los valores *Hash* A, B y C. Además, el multiplexador 96 pasa la primera de las cuatro claves de programa generadas de manera dinámica desde la siguiente parte de la memoria 66 hasta el segundo motor de cifrado 18.

En el segundo motor de cifrado 18, el valor *Hash* A se aplica al EXCLUSIVE OR 106, el valor *Hash* B se aplica al EXCLUSIVE OR 108, y el valor *Hash* C se aplica al EXCLUSIVE OR 110. La clave de segmento de mensaje expandida A se aplica al cifrador AES 112, la clave de segmento de mensaje expandida B se aplica al cifrador AES 114, y la clave de segmento de mensaje expandida C se aplica al cifrador AES 116. El valor inicial se aplica al EXCLUSIVE OR 106, la palabra de control se aplica al cifrador AES 112, un primer 1/2 de la primera de las cuatro claves de programa generadas de manera dinámica se aplica al cifrador AES 114, y una segunda mitad de la primera de las cuatro claves de programa generadas de manera dinámica se aplica al cifrador AES 116.

El cifrador AES 112 cifra una salida del EXCLUSIVE OR 106 y la palabra de control según la clave de segmento de mensaje expandida A, y suministra la mitad del resultado de cifrado al EXCLUSIVE OR 108 y la otra mitad como el segundo 1/4 del mensaje de clave de programa KM1. El cifrador AES 114 cifra una salida del EXCLUSIVE OR 108 y el primer 1/2 de la primera de las cuatro claves de programa generadas de manera dinámica según la clave de segmento de mensaje expandida B, y suministra la mitad del resultado de cifrado al EXCLUSIVE OR 110 y la otra mitad como el tercer 1/4 del mensaje de clave de programa KM1. El cifrador AES 116 cifra una salida del EXCLUSIVE OR 110 y el segundo 1/2 de la primera de las cuatro claves de programa generadas de manera dinámica según la clave de segmento de mensaje expandida C, y suministra la mitad del resultado de cifrado como el primer 1/4 del mensaje de clave de programa KM1 y la otra mitad como el cuarto 1/4 del mensaje de clave de programa KM1.

Los otros tres mensajes de clave de programa KM2, KM3 y KM4 se generan de manera similar.

Cada mensaje de clave de modificación en el segmento de mensaje de clave de modificación MKMS tiene también la construcción de ejemplo de la figura 12. Según este ejemplo, un mensaje de clave de modificación MK1 incluye un valor inicial de 64 bits, que puede ser el mismo valor inicial comentado anteriormente o un valor inicial diferente, un control de claves 98 de 64 bits, y una de las claves de modificación de 128 bits dividida en dos partes de 64 bits. Los mensajes de clave de modificación MK2 y MK3 que contienen las otras dos claves de modificación están construidos de manera similar.

De nuevo, el control de claves 98 se usa para designar si el mensaje de clave contiene una clave de programa, una clave de modificación, o la suma de comprobación.

Con el fin de cifrar el mensaje de clave de modificación MK1, el multiplexador 100 pasa las tres claves de segmentos de mensaje A, B y C y los tres valores *Hash* A, B y C de la memoria de claves de segmentos de mensaje y valores *hash* 86 a través del expansor de claves 104 hasta el segundo motor de cifrado 18. Tal como se explicó anteriormente, las tres claves de segmentos de mensaje A, B y C que se usan para cifrar los mensajes de clave de modificación son las claves de segmentos de mensaje leídas de la tabla 94 mediante el uso de las claves de modificación con el valor predeterminado. Por tanto, las claves fijas K_A de la memoria de claves fijas 82 pueden leerse de la tabla 94 como las claves de segmentos de mensaje. Alternativamente, tal como se explicó anteriormente, las claves fijas K_B de la memoria de claves fijas 84 pueden leerse de la tabla 94 como las claves de segmentos de mensaje o una combinación de las claves fijas K_A y K_B puede usarse para leer las claves de segmentos de mensaje de la tabla 94. El expansor de claves 104 expande sólo las claves de segmentos de mensaje A, B y C. El expansor de claves 104 no expande los valores *Hash* A, B y C. Además, el multiplexador 96 pasa la primera de las claves de modificación de la memoria de claves de modificación 88 hacia el segundo motor de cifrado 18.

Los valores *Hash* A, B y C se aplican a los EXCLUSIVE OR 106, 108 y 110 tal como anteriormente. Además, las claves de segmentos de mensaje expandidas A, B y C se aplican a los cifradores AES 112, 114 y 116 tal como anteriormente. El valor inicial se aplica al EXCLUSIVE OR 106, la palabra de control se aplica al cifrador AES 112, un primer 1/2 de la primera de las tres claves de modificación se aplica al cifrador AES 114, y una segunda mitad de la primera de las tres claves de modificación se aplica al cifrador AES 116.

El cifrador AES 112 suministra la mitad de su resultado de cifrado al EXCLUSIVE OR 108 y la otra mitad como el segundo 1/4 del mensaje de clave de modificación MK1. El cifrador AES 114 suministra la mitad de su resultado de cifrado al EXCLUSIVE OR 110 y la otra mitad como el tercer 1/4 del mensaje de clave de modificación MK1. El cifrador AES 116 suministra la mitad de su resultado de cifrado como el primer 1/4 del mensaje de clave de modificación MK1 y la otra mitad como el cuarto 1/4 del mensaje de clave de modificación MK1.

Los otros dos mensajes de clave de modificación MK2 y MK3 y el mensaje de suma de comprobación CRC se

generan de manera similar.

El multiplexador de salida 16 de la figura 1 multiplexa los datos de programa cifrados, la cabecera de PID de MPEG del flujo de transporte, 192 bits de reloj que pueden suministrarse por un generador separado y que pueden ser el código de tiempo SMPTE (si hay alguno), y 20 bytes de corrección de error hacia delante del flujo de transporte con el segmento de mensaje de clave de programa cifrado PKMS y el segmento de mensaje de clave de modificación cifrado MKMS para formar el flujo de transporte cifrado. Tanto el segmento de mensaje de clave de programa PKMS como el segmento de mensaje de clave de modificación MKMS está contenido en un segmento de datos de ATSC completos correspondiente.

El segundo motor de cifrado 18 genera los segmentos de mensaje MS por pares, es decir, el segmento de mensaje de clave de programa PKMS y el segmento de mensaje de clave de modificación MKMS. Este par de segmentos de mensaje MS se muestra en la figura 13. El mensaje modificador MM en cada segmento de mensaje MS se proporciona según las figuras 8 y 9. El primer segmento de mensaje mostrado en la figura 13 es el segmento de mensaje de clave de modificación MKMS y contiene una forma cifrada de las tres claves de modificación almacenadas en la memoria de claves de modificación 88 y la suma de comprobación (CRC) de la memoria 20. El segundo segmento de mensaje mostrado en la figura 13 es el segmento de mensaje de clave de programa PKMS y contiene una forma cifrada de las cuatro nuevas claves de programa cifradas que van a aplicarse por el receptor para descifrar los datos de programa cifrados.

Por tanto, tal como se muestra en la figura 10, el mensaje modificador MM y los cuatro mensajes de clave de programa KM1, KM2, KM3 y KM4 del segmento de mensaje de clave de programa PKMS están cifrados. De manera similar, el mensaje modificador MM, los tres mensajes de clave de modificación MK1, MK2 y MK3, y el mensaje de suma de comprobación CRC del segmento de mensaje de clave de modificación MKMS están cifrados.

La cabecera de cuatro bytes del segmento de mensaje MS mostrado en la figura 10 es el PID de MPEG. El mensaje modificador MM incluye los bytes de control de mensajes mostrados en la figura 9. Este byte de control identifica el segmento de mensaje MS en un par o bien como el segmento de mensaje de clave de programa PKMS o bien como el segmento de mensaje de clave de modificación MKMS, tal como se explicó anteriormente.

La figura 14 muestra el sincronismo de transmisión y recepción de pares relativos de mensajes con el que se determina la sincronización de clave. Cuando sucede el evento 1, que puede ser un paquete nulo en el flujo de transporte MPEG, se transmite un segmento de mensaje de clave de programa PKMS tal como se muestra en la figura 14. El receptor recibe este segmento de mensaje de clave de programa PKMS, lo descifra, y almacena las claves de programa que estaban contenidas en el segmento de mensaje de clave de programa PKMS como siguientes claves de programa. Sin embargo, el receptor no empieza a usar estas siguientes claves de programa todavía.

Después de que el transmisor transmite el segmento de mensaje de clave de programa PKMS, el codificador de cifrado 8 del transmisor crea las tres claves de modificación y el mensaje modificador MM, y cifra el mensaje modificador MM y las tres claves de modificación usando las claves de segmentos de mensaje y los valores *Hash* tal como se describió anteriormente. El codificador de cifrado 8 ensambla entonces el segmento de mensaje de clave de modificación MKMS que contiene el mensaje modificador MM cifrado y las tres claves de modificación tal como se describió anteriormente. Cuando se detecta un paquete nulo (evento 2), el transmisor transmite el segmento de mensaje de clave de modificación MKMS en lugar del paquete nulo y, al mismo tiempo, el codificador de cifrado 8 empieza a usar las siguientes claves de programa almacenadas en la memoria 66 como las claves de programa activas para cifrar datos de programa. Por tanto, las siguientes claves de programa se convierten en las claves de programa activas.

Al mismo tiempo, el receptor recibe este segmento de mensaje de clave de modificación MKMS e inmediatamente empieza a usar sus claves de programa previamente almacenadas como las claves de programa activas para descifrar el contenido de programa. Por consiguiente, la sustitución de las claves de programa activas por las siguientes claves de programa se realiza al mismo tiempo en el transmisor y receptor de modo que el transmisor y el receptor usan las mismas claves de programa para cifrar y descifrar el mismo contenido de programa.

Después de que el transmisor transmite el segmento de mensaje de clave de modificación MKMS y cambia de claves de programa, el codificador de cifrado 8 del transmisor crea nuevas claves de programa, y guardas las nuevas claves de programa en la memoria 66 como las siguientes claves de programa. El codificador de cifrado 8 cifra las nuevas claves de programa y ensambla otro segmento de mensaje de clave de programa PKMS que contiene las nuevas claves de programa y espera una oportunidad (evento 3 tal como un paquete nulo) para transmitir este segmento de mensaje de clave de programa PKMS.

Aunque el codificador de cifrado 8 del transmisor crea nuevas claves de programa, guarda las nuevas claves de programa, y ensambla el siguiente segmento de mensaje de clave de programa PKMS, el receptor descifra el segmento de mensaje de clave de modificación MKMS que se acaba de recibir, y guarda el mensaje modificador MM y las claves de modificación contenidas en este mensaje.

Durante los segmentos en los que el codificador de cifrado 8 no está transmitiendo segmentos de mensaje de clave de programa PKMS y segmentos de mensaje de clave de modificación MKMS, el codificador de cifrado 8 está usando las claves de programa activas para cifrar datos de programa y está transmitiendo los datos de programa cifrados al receptor.

Durante los segmentos en los que el receptor no está recibiendo segmentos de mensaje de clave de programa PKMS y segmentos de mensaje de clave de modificación MKMS, el receptor está usando las claves de programa activas para descifrar datos de programa.

En una realización en la que la transmisión de mensajes y uso de claves está sincronizado con la aparición de paquetes nulos, pueden darse ocasiones en las que aparecen paquetes nulos con una frecuencia elevada de manera no deseada. Por ejemplo, durante periodos en los que hay poca actividad en el vídeo, pueden aparecer muchos paquetes nulos durante una sola trama. Por tanto, puede ser deseable añadir una función de retardo de manera que la transmisión de mensajes y el cambio de claves no se produzcan con mayor frecuencia que con una frecuencia predeterminada. Por ejemplo, esta función de retardo puede fijarse de modo que la transmisión de mensajes y el cambio de claves no se produzcan con mayor frecuencia que una vez cada dos o tres tramas ATSC.

Durante cifrado de datos de programa, el bloque de cifrado 40 rota las cuatro claves de programa activas PK. La figura 15 muestra la rotación. Tal como se muestra en la figura 16, cada segmento de datos de programa de un campo que va a transmitirse al receptor incluye una cabecera de MPEG de cuatro bytes no cifrada que identifica el segmento como un segmento de datos de programa, once bloques conteniendo cada uno 128 bits cifrados de datos de programa, ocho bytes de datos de programa no cifrados, y veinte bytes de datos de corrección de error hacia delante no cifrados.

Tal como se muestra en la figura 15, las cuatro claves de programa activas A, B, C y D se aplican en el siguiente orden a los once bloques de datos en el primer segmento de datos de programa: A, B, C, D, A, B, C, D, A, B, C. Por consiguiente, la clave de programa activa A se aplica al primero de los once bloques de datos que deben cifrarse, la clave de programa activa B se aplica al segundo de los once bloques de datos que deben cifrarse, ..., y la clave de programa activa C se aplica al undécimo de los once bloques de datos que deben cifrarse.

Este mismo esquema de rotación ABCDABCDABC puede usarse para los segmentos de datos de programa siguientes y subsiguientes de un campo.

Alternativamente, el siguiente segmento de datos de programa puede continuar la rotación. Por tanto, las claves de programa activas A, B, C y D se aplican en el siguiente orden a los once bloques de datos que deben cifrarse en el segundo segmento de datos de programa: D, A, B, C, D, A, B, C, D, A, B. Por consiguiente, la clave de programa activa D se aplica al primero de los once bloques de datos que deben cifrarse, la clave de programa activa A se aplica al segundo de los once bloques de datos que deben cifrarse, ..., y la clave de programa activa B se aplica al undécimo de los once bloques de datos que deben cifrarse. La rotación puede continuar entonces para los segmentos de datos de programas subsiguientes tal como se indica mediante la figura 15.

Como alternativa adicional, pueden usarse otras secuencias de rotación. Los bits 12-15 del byte de control de sistema mostrados en la figura 9 pueden usarse para indicar al receptor la rotación particular que está usándose en el transmisor.

El multiplexador de salida 16 transmite segmentos de datos de programa cifrados de manera continua hasta que surge una oportunidad (evento) para transmitir un segmento de mensaje MS (o bien un segmento de mensaje de clave de programa PKMS o bien un segmento de mensaje de clave de modificación MKMS). La aparición de un paquete nulo da lugar a la oportunidad de transmitir uno de estos segmentos de mensaje, la aparición del siguiente paquete nulo da lugar a la oportunidad de transmitir el otro de los segmentos de mensaje MS en el par, etc. Puede establecerse un objetivo para transmitir un segmento de mensaje MS de una manera periódica que depende de la aparición de un paquete nulo. Por ejemplo, el objetivo puede ser transmitir un segmento de mensaje MS no con mayor frecuencia que una vez por campo de 312 segmentos.

Un decodificador de descifrado 180 de ejemplo de un receptor de protección frente al copiado se muestra en la figura 17. El decodificador de descifrado 180 incluye un filtro PID 182 que, basándose en números PID, detecta y reenvía datos de programa cifrados a un primer motor de descifrado 184 y detecta y reenvía segmentos de mensaje de clave de programa PKMS y segmentos de mensaje de clave de modificación MKMS a un segundo motor de descifrado 186. El primer motor de descifrado 184 efectúa un proceso de descifrado de una única envoltura que es complementario al proceso de cifrado de una única envoltura realizado por el primer motor de cifrado 14.

Cuando se recibe el segmento de mensaje de clave de modificación MKMS, el segundo motor de descifrado 186 descifra (desenvuelve) este segmento de mensaje con el fin de recuperar las claves de modificación y la clave fija y direcciones de valor *Hash* de una memoria 188. Un selector de claves fijas y un generador de claves de segmentos de mensaje 190 usa estas direcciones de valor *Hash* y clave fija para recuperar claves fijas y valores *Hash* de la memoria 188. En el caso de descifrar el segmento de mensaje de clave de modificación MKMS, el selector de claves fijas y el generador de claves de segmentos de mensaje 190 usan las claves fijas y los valores *Hash* recuperados de

la memoria 188 junto con las claves de modificación conocidas anteriores, es decir, las claves de modificación que tienen el valor predeterminado conocido, con el fin de regenerar las claves de segmentos de mensaje que se usaron en el codificador de cifrado 8 para cifrar las claves de modificación y el mensaje de suma de comprobación CRC y que el decodificador de descifrado 180 requiere para descifrar las claves de modificación cifradas y el mensaje de suma de comprobación CRC. En el caso de descifrar el segmento de mensaje de clave de programa PKMS, el selector de claves fijas y el generador de claves de segmentos de mensaje 190 usa las claves fijas y los valores *Hash* recuperados de la memoria 188 basándose en las direcciones de memoria contenidas en el mensaje modificador del segmento de mensaje de clave de programa PKMS junto con las claves de modificación descifradas con el fin de regenerar las claves de segmentos de mensaje que se usaron en el codificador de cifrado 8 para cifrar las claves de programa y que el decodificador de descifrado 180 requiere para descifrar los mensajes de claves de programa cifrado KM1, KM2, KM3 y KM4.

Cuando se recibe el segmento de mensaje de clave de programa PKMS, el segundo motor de descifrado 186 descifra las claves de programa en el segmento de mensaje MS usando las claves de segmentos de mensaje a partir del selector de claves fijas y el generador de claves de segmentos de mensaje 190 y almacena las claves de programa descifradas en la siguiente parte de una memoria 192. Mientras tanto, el primer motor de descifrado 184 usa las claves de programa activas almacenadas en la memoria 192 para descifrar los datos cifrados a partir de los segmentos de datos de programa del campo que está recibiéndose.

Tal como se muestra en la figura 18, el primer motor de descifrado 184 incluye tres secciones 184A, 184B y 184C. La sección 184A incluye un demultiplexador 200, memorias 202 y 204 y un multiplexador 206. La sección 184B incluye una memoria 208, un bloque de descifrado 210 y un multiplexador 212. La sección 184C incluye un demultiplexador 214, memorias 216 y 218 y un multiplexador 220. Las secciones 184A, 184B y 184C se controlan mediante el filtro PID 182.

El filtro PID 182 pasa todos los paquetes en el flujo de transporte MPEG al demultiplexador 200. Todos los paquetes se demultiplexan y se almacenan en las memorias 202 y 204 que operan a modo de ping-pong. Todos los paquetes en las memorias 202 y 204 se suministran al multiplexador 206.

El multiplexador 206 pasa todos los paquetes desde las memorias 202 y 204 hacia la memoria 208 y hacia el bloque de descifrado 210. Estos paquetes incluyen paquetes de programa (uno o más de los cuales pueden estar cifrados), segmentos de mensaje, y paquetes no de programa tales como PID, PSIP, PMT y PAT. El bloque de descifrado 210 usa las claves de programa descifradas PK para descifrar todos los paquetes que éste recibe y suministra los paquetes descifrados al multiplexador 212. El multiplexador 212, en respuesta a un indicador de descifrado desde el filtro PID 182, selecciona sólo los paquetes descifrados del bloque de descifrado 210 que corresponden al programa o programas seleccionados que tenían que descifrarse. Todos los demás paquetes (aquéllos que no corresponden al programa que va a descifrarse) se seleccionan por el multiplexador 212 desde la memoria 208. Por tanto, la salida del multiplexador 212 es el flujo de transporte MPEG original menos los paquetes nulos y que incluye segmentos de mensaje. El multiplexador 212 pasa los paquetes descifrados y no cifrados al demultiplexador 214.

Los paquetes descifrados y no cifrados del demultiplexador 214 se almacenan en las memorias 216 y 218 que operan a modo de ping-pong. Los paquetes descifrados y no cifrados en las memorias 216 y 218 se suministran a través del multiplexador 220 a un insertador de nulos 222.

El insertador de nulos 222 se controla mediante el filtro PID 182 para retirar los segmentos de mensaje de clave de programa PKMS y los segmentos de mensaje de clave de modificación MKMS desde el flujo de transporte, y para insertar los paquetes nulos de nuevo en el flujo de transporte en lugar de los segmentos de mensaje de clave de programa PKMS retirados y los segmentos de mensaje de clave de modificación MKMS retirados. La salida del insertador de nulos es el flujo de transporte MPEG descifrado.

Las secciones 184A y 184C del primer motor de descifrado 184 se controlan mediante los paquetes de mensaje para que mantengan un sincronismo, unas tasas de flujo de datos y una sincronización apropiados.

El selector de claves fijas y el generador de claves de segmentos de mensaje 190 se muestra más detalladamente en la figura 19. Tal como se muestra en la figura 19, los segmentos de mensaje de clave de programa PKMS y los segmentos de mensaje de clave de modificación MKMS se suministran al segundo motor de descifrado 186. Cada uno de estos segmentos de mensaje tiene la forma mostrada en la figura 10. Por consiguiente, tal como se muestra en la figura 20, el mensaje modificador MM en el segmento de mensaje recibido se descifra usando las tres claves fijas A', B' y C' y los tres valores *Hash* A', B' y C' que se almacenan en una memoria 230. Las tres claves fijas A', B y C' y los tres valores *Hash* A', B' y C' almacenados en la memoria 230 son las mismas claves fijas y valores *Hash* que se almacenan en la memoria 102.

El mensaje modificador MM descifrado indica al receptor, entre otros, si el segmento de mensaje correspondiente es un segmento de mensaje de clave de programa PKMS o un segmento de mensaje de clave de modificación MKMS. Si el segmento de mensaje correspondiente es un segmento de mensaje de clave de programa PKMS, el receptor sabe usar las claves de modificación descifradas K_M así como las claves fijas K_A y K_B para producir las claves de segmentos de mensaje que se requieren para el descifrado de los mensajes de claves de programa. Si el segmento

de mensaje correspondiente es un segmento de mensaje de clave de modificación MKMS, el receptor sabe usar las claves de modificación conocidas que tienen el valor predeterminado con el fin de extraer las claves fijas K_A, K_B o alguna combinación de K_A y K_B así como las claves de segmentos de mensaje que se requieren para el descifrado de los mensajes de clave de modificación y el mensaje de suma de comprobación CRC.

Con el fin de descifrar el mensaje modificador MM en uno recibido de los segmentos de mensaje de clave de modificación MKMS o de los segmentos de mensaje de clave de programa PKMS, un multiplexador 232 pasa las tres claves fijas A', B' y C' y los tres valores *Hash* A', B' y C' desde la memoria 230 a través de un expansor de claves 234 hacia el segundo motor de cifrado 186. El expansor de claves 234, por ejemplo, puede ser similar al expansor de claves 104 y expande sólo las claves fijas A', B' y C'. El expansor de claves 234 no expande los valores *Hash* A', B' y C'.

El segundo motor de cifrado 186 que efectúa una operación complementaria a la efectuada por el motor de cifrado 18 se muestra más detalladamente en la figura 20. Tal como se muestra en la figura 20, el valor *Hash* C' se aplica a un EXCLUSIVE OR 236, el valor *Hash* B' se aplica a un EXCLUSIVE OR 238, y el valor *Hash* A' se aplica a un EXCLUSIVE OR 240. Los EXCLUSIVE OR 236, 238 y 240 procesan por bits sus respectivas entradas. La clave fija expandida C' se aplica a un descifrador AES 242, la clave fija expandida B' se aplica a un descifrador AES 244, y la clave fija expandida A' se aplica a un descifrador AES 246.

El primer 1/4 del mensaje modificador MM cifrado se aplica al descifrador AES 242, el segundo 1/4 del mensaje modificador MM cifrado se aplica al descifrador AES 246, el tercer 1/4 del mensaje modificador MM cifrado se aplica al descifrador AES 244, y el cuarto 1/4 del mensaje modificador MM cifrado se aplica al descifrador AES 242.

El descifrador AES 242 descifra el primer 1/4 y el cuarto 1/4 del mensaje modificador MM cifrado según la clave fija expandida C', y suministra la mitad del resultado de descifrado al EXCLUSIVE OR 236 y la otra mitad como el tercer 1/3 de los bits de control del mensaje modificador MM descifrado. El descifrador AES 244 descifra una salida del EXCLUSIVE OR 236 y el tercer 1/4 del mensaje modificador MM cifrado según la clave fija expandida B', y suministra la mitad del resultado de descifrado al EXCLUSIVE OR 238 y la otra mitad como el segundo 1/3 de los bits de control del mensaje modificador MM descifrado. El cifrador AES 246 descifra una salida del EXCLUSIVE OR 238 y el segundo 1/4 del mensaje modificador MM cifrado según la clave fija expandida A', y suministra la mitad del resultado de cifrado al EXCLUSIVE OR 240 y la otra mitad como el primer 1/3 del mensaje modificador MM descifrado. La salida del EXCLUSIVE OR 240 es el valor inicial del mensaje modificador MM. Si este valor inicial no es el mismo valor inicial que se usó durante el cifrado del mensaje modificador MM, el proceso de cifrado/descifrado tiene entonces un error que indica un descifrado de mensaje erróneo.

Tal como se muestra en la figura 19, un multiplexador 250 aplica los bits de control del mensaje modificador MM descifrado a un decodificador de mensaje modificador 252.

Tras el descifrado del mensaje modificador MM, el multiplexador 232 pasa las tres claves de segmentos de mensaje A, B y C y los tres valores *Hash* A, B y C almacenados en una memoria de clave de segmento de mensaje 254 al expansor de claves 234. Cuando está descifrándose el segmento de mensaje de clave de modificación MKMS, estas tres claves de segmentos de mensaje se producen con las claves de modificación que tienen el valor predeterminado. El expansor de claves expande sólo las tres claves de segmentos de mensaje A, B y C, no expande los tres valores *Hash* A, B y C. El segundo motor de descifrado 186 usa las tres claves de segmentos de mensaje A, B y C expandidas y los tres valores *Hash* A, B y C para descifrar el mensaje de clave de modificación MK1 en el segmento de mensaje de clave de modificación MKMS recibido. Tal como se indicó anteriormente, cada uno de los tres mensajes de clave de modificación MK1, MK2 y MK3 y el mensaje de suma de comprobación CRC tiene el formato mostrado en la figura 12, y el control de cada uno de los mensajes es el control de claves 98 que indica si el mensaje particular es un mensaje de clave de programa, un mensaje de clave de modificación o un mensaje de suma de comprobación.

Tal como se muestra en la figura 20, el valor *Hash* C se aplica al EXCLUSIVE OR 236, el valor *Hash* B se aplica al EXCLUSIVE OR 238, y el valor *Hash* A se aplica al EXCLUSIVE OR 240. La clave fija expandida C se aplica al descifrador AES 242, la clave fija expandida B se aplica al descifrador AES 244, y la clave fija expandida A se aplica al descifrador AES 246.

El primer 1/4 del mensaje de clave de modificación MK1 cifrado se aplica al descifrador AES 242, el segundo 1/4 de del mensaje de clave de modificación MK1 cifrado se aplica al descifrador AES 246, el tercer 1/4 del mensaje de clave de modificación MK1 cifrado se aplica al descifrador AES 244, y el cuarto 1/4 del mensaje de clave de modificación MK1 cifrado se aplica al descifrador AES 242.

El descifrador AES 242 suministra la mitad de su resultado de descifrado al EXCLUSIVE OR 236 y la otra mitad como el segundo 1/2 de la clave de modificación MK1 descifrada. El descifrador AES 244 suministra la mitad de su resultado de descifrado al EXCLUSIVE OR 238 y la otra mitad como el primer 1/2 de la clave de modificación descifrada. El cifrador AES 246 suministra la mitad de su resultado de cifrado al EXCLUSIVE OR 240 y la otra mitad como el control de la clave de modificación descifrada. La salida del EXCLUSIVE OR 240 es el valor inicial del mensaje de clave de modificación. Si el valor inicial no es el mismo valor inicial que se usó durante el cifrado de la

clave de modificación MK1, el proceso de cifrado/descifrado tiene entonces un error que indica la necesidad de una acción correctiva.

El motor de descifrado 186 descifra de manera similar los mensajes de clave de modificación MK2 y MK3 y el mensaje de suma de comprobación CRC. El multiplexador 250 pasa los controles y la suma de comprobación tal como se indica en la figura 19, y pasa las claves de modificación para su almacenamiento en una memoria de claves de modificación 256.

Tras el descifrado del segmento de mensaje recibido de clave de modificación MKMS, el selector de claves fijas y el generador de claves de segmentos de mensaje 190 pueden empezar la generación de nuevas claves de segmentos de mensaje que se usarán para descifrar las claves de programas a partir del siguiente segmento de mensaje recibido de clave de programa PKMS.

El decodificador de mensaje modificador 252 decodifica el mensaje modificador MM recibido y descifrado en cada uno de los segmentos de mensaje para determinar las direcciones según la definición y el formato del mensaje modificador mostrados en las figuras 8 y 9. El selector de claves fijas 260 usa estas direcciones para seleccionar, de la memoria 188, las mismas tres claves K_A, las mismas tres claves fijas K_B, y los mismos tres valores *Hash* A, B y C que se usaron para producir las claves de segmentos de mensaje A, B y C que se usaron para cifrar los segmentos de mensaje PKMS y MKMS en el codificador de cifrado 8. Una primera memoria de claves 262 almacena las tres claves K_A, una segunda memoria de claves fijas 264 almacena las tres claves fijas K_B seleccionadas y la memoria de clave de segmento de mensaje 254 almacena los tres valores *Hash* A, B y C seleccionados.

Un generador de claves de segmentos de mensaje 266 puede tener la misma construcción que el generador de claves de segmentos de mensaje 90 mostrado en la figura 6. Por consiguiente, el *latch* 92_1 retiene los primeros 32 bits de una primera de las tres claves fijas K_A almacenadas en la memoria de claves fijas 262, el *latch* 92_2 retiene los primeros 32 bits de una primera de las tres claves fijas K_B almacenadas en la memoria de claves fijas 264, y el *latch* 92_3 retiene los primeros 32 bits de una primera de las tres claves de modificación K_M almacenadas en la memoria de claves de modificación 256 cuando están produciéndose las claves de segmentos de mensaje para descifrar claves de programa (de lo contrario, las claves de modificación que tienen el valor predeterminado se usan para generar claves de segmentos de mensaje para descifrar claves de modificación). Estos 96 bits retenidos forman una dirección de 96 bits que extraen mediante lectura los primeros 32 bits de una primera clave de segmento de mensaje para su almacenamiento en la memoria de clave de segmento de mensaje 254.

La misma tabla que se seleccionó en el transmisor se selecciona en el receptor para proporcionar las tres claves de segmentos de mensaje que se almacenan en la memoria de clave de segmento de mensaje 254.

Una vez que los primeros 32 bits de la primera clave de segmento de mensaje se han extraído de la tabla de consulta 94 y se han almacenado en la memoria de clave de segmento de mensaje 254, el *latch* 92_1 retiene los segundos 32 bits de la primera de las tres claves fijas K_A almacenadas en la memoria de claves fijas 262, el *latch* 92_2 retiene los segundos 32 bits de la primera de las tres claves fijas K_B almacenadas en la memoria de claves fijas 264, y el *latch* 92_3 retiene los segundos 32 bits de la primera de las tres claves de modificación K_M almacenadas en la memoria de claves de modificación 256 cuando están produciéndose las claves de segmentos de mensaje para descifrar claves de programa (de lo contrario, las claves de modificación que tienen el valor predeterminado se usan para generar claves de segmentos de mensaje para descifrar claves de modificación). Estos 96 bits retenidos forman una segunda dirección de 96 bits que extraen mediante lectura los segundos 32 bits de la primera clave de segmento de mensaje para su almacenamiento en la memoria de clave de segmento de mensaje 254.

Los terceros y cuartos 32 bits de la primera de las tres claves fijas K_A almacenadas en la memoria de claves fijas 262, de la primera de las tres claves fijas K_B almacenadas en la memoria de claves fijas 264, y de la primera de las tres claves de modificación K_M almacenadas en la memoria de claves de modificación 256 se usan para extraer mediante lectura los terceros y cuartos 32 bits de la primera clave de segmento de mensaje desde la tabla de consulta 94 cuando las claves de segmentos de mensaje están produciéndose para descifrar claves de programa (de lo contrario, las claves de modificación que tienen el valor predeterminado se usan para generar claves de segmentos de mensaje para descifrar claves de modificación. Estos terceros y cuartos 32 bits de la primera clave de segmento de mensaje se almacenan también en la memoria de clave de segmento de mensaje 254 para formar todos los 128 bits de la primera clave de segmento de mensaje. Las claves segunda y tercera de segmentos de mensaje se extraen de manera similar de la tabla de consulta 94 y se almacenan en la memoria de clave de segmento de mensaje 254.

Cuando se recibe el siguiente segmento de mensaje de clave de programa PKMS, el mensaje modificador MM en el segmento de mensaje recibido MS se descifra tal como anteriormente usando las claves fijas A', B' y C' y los valores *Hash* A', B' y C' almacenados en la memoria 230. Posteriormente, el multiplexador 232 pasa las tres claves de segmentos de mensaje A, B y C y los tres valores *Hash* A, B y C desde la memoria de clave de segmento de mensaje 254 a través del expansor de claves 234 hacia el segundo motor de cifrado 186. El expansor de claves 234 expande sólo las claves de segmentos de mensaje A, B y C. El expansor de claves 234 no expande los valores *Hash* A, B y C.

En el segundo motor de cifrado 186, el valor *Hash* C se aplica al EXCLUSIVE OR 236, el valor *Hash* B se aplica al EXCLUSIVE OR 238, y el valor *Hash* A se aplica al EXCLUSIVE OR 240. La clave fija expandida C se aplica al descifrador AES 242, la clave fija expandida B se aplica al descifrador AES 244, y la clave fija expandida A se aplica al descifrador AES 246.

El primer 1/4 del primer mensaje de clave de programa KM1 cifrado se aplica al descifrador AES 242, el segundo 1/4 del primer mensaje de clave de programa KM1 cifrado se aplica al descifrador AES 246, el tercer 1/4 del primer mensaje de clave de programa KM1 cifrado se aplica al descifrador AES 244, y el cuarto 1/4 del mensaje de clave de programa KM1 cifrado se aplica al descifrador AES 242.

El descifrador AES 242 descifra el primer 1/4 y el cuarto 1/4 del primer mensaje de clave de programa KM1 cifrado según la clave fija expandida C, y suministra la mitad del resultado de descifrado al EXCLUSIVE OR 236 y la otra mitad como el segundo 1/2 de la primera clave de programa del primer mensaje de clave de programa KM1 descifrado. El descifrador AES 244 descifra una salida del EXCLUSIVE OR 236 y el tercer 1/4 del primer mensaje de clave de programa KM1 cifrado según la clave fija expandida B, y suministra la mitad del resultado de descifrado al EXCLUSIVE OR 238 y la otra mitad como el primer 1/2 de la primera clave de programa del primer mensaje de clave de programa KM1 descifrado. El cifrador AES 246 descifra una salida del EXCLUSIVE OR 238 y el segundo 1/4 del primer mensaje de clave de programa KM1 cifrado según la clave fija expandida A, y suministra la mitad del resultado de cifrado al EXCLUSIVE OR 240 y la otra mitad como el control del primer mensaje de clave de programa KM1 descifrado. La salida del EXCLUSIVE OR 240 es el valor inicial del primer mensaje de clave de programa KM1. Si este valor inicial no es el mismo valor inicial que se usó durante el cifrado del primer mensaje de clave de programa KM1, el proceso de cifrado/descifrado tiene entonces un error que indica la necesidad de acción correctiva.

Los otros tres mensajes de clave de programa KM2, KM3 y KM4 se descifran de manera similar.

El multiplexador 250 de la figura 19 pasa estas cuatro claves de programa a la siguiente parte de la memoria 192 y pasa el control de cada uno de los mensajes de clave de programa KM1, KM2, KM3 y KM4 descifrados.

Un multiplexador 270 pasa las claves de programa activas, usando la rotación tratada anteriormente en relación con las figuras 15 y 16, a través de un expansor de claves 272 al bloque de descifrado 210 de modo que pueden descifrarse los datos apropiados. El expansor de claves 272 puede construirse según la figura 4. Tal como en el caso de expansor de claves 70, el expansor de claves 272 incluye también un bloque de clave inversa. Este bloque de clave inversa se deshabilita durante el descifrado de programa y se habilita durante el descifrado del segmento de mensaje de clave de programa PKMS y el segmento de mensaje de clave de modificación MKMS.

Mientras que el bloque de descifrado 210 está usando las claves activas de la parte activa de la memoria 192 para descifrar datos, las siguientes claves de programa se reciben y almacenan en la siguiente parte de la memoria 192.

El decodificador de mensaje modificador 252 decodifica también todo el control de sistema del mensaje modificador MM recibido y descifrado. Tal como se trató anteriormente, el control de sistema del mensaje modificador MM se muestra en la figura 9. Por consiguiente, el decodificador de mensaje modificador 252 aplica el mismo código CRC tal como el codificador a los bits 0-15 del control de sistema del mensaje modificador MM en el segmento de mensaje recibido PKMS o MKMS con el fin de volver a calcular los bits 16-31 de suma de comprobación. El receptor compara la suma de comprobación calculada de nuevo de los bits 0-15 con los bits 16-31 de suma de comprobación en el control de sistema recibido. Si la suma de comprobación calculada de nuevo de los bits 0-15 con los bits 16-31 de suma de comprobación recibidos no coinciden, se trata el segmento de mensaje recibido como el siguiente segmento de mensaje que espera recibirse en la secuencia de los segmentos de mensaje recibidos.

Además, el decodificador de mensaje modificador 252 usa los bits 12-15 decodificados del control de sistema para determinar la rotación de claves de programa que el bloque de descifrado 210 debe usar para descifrar los paquetes de programa cifrados tal como muestra la línea que se extiende desde el decodificador de mensaje modificador 252 hacia el control del multiplexador 270 que selecciona la siguiente clave activa que va a usarse.

Se han tratado anteriormente determinadas modificaciones de la presente invención. A los expertos en la técnica de la presente invención se les ocurrirán otras modificaciones de la presente invención. Por ejemplo, las memorias tal como se describió anteriormente pueden ser ROM, RAM, RAM no volátiles y/o cualquier otro dispositivo de memoria adecuado.

Además, tal como se dio a conocer anteriormente, se usa una tabla de consulta 94 de 96 x 32 para producir las claves de segmentos de mensaje. Por consiguiente, los 96 bits de dirección se usan para leer 32 bits de una clave de segmento de mensaje. En su lugar, pueden usarse otras tablas de consulta y esquemas de dirección para producir las claves de segmentos de mensaje. Por ejemplo, puede usarse una tabla de consulta de 384 x 128 para producir las claves de segmentos de mensaje. Por consiguiente, los 384 bits de dirección que comprenden 128 bits de K_M, 128 bits de K_A y 128 bits de K_B se usan para leer una clave de segmento de mensaje de 128 bits. Independientemente de la tabla de consulta y esquema de direccionamiento que se use en el transmisor, debe usarse la misma tabla de consulta y el mismo esquema de direccionamiento en el receptor.

Por consiguiente, la descripción de la presente invención debe interpretarse sólo como ilustrativa y es para enseñar a los expertos en la técnica la mejor manera de llevar a cabo la invención.

REIVINDICACIONES

1. Método, implementado por un receptor (180), de descifrado de datos cifrados que comprende:

recibir una clave de datos (PKMS) desde un transmisor (8), en el que la clave de datos (PKMS) se genera por un generador de números aleatorios sembrado con una semilla derivada de los datos que deben cifrarse en el transmisor, y en el que los datos están contenidos en un paquete de transporte MPEG;

recibir, desde el transmisor, una señal de sincronización (MKMS) y la clave de datos en lugar de respectivos paquetes nulos en un flujo de transporte MPEG;

recibir, desde el transmisor, datos cifrados en el flujo de transporte MPEG;

sincronizar, en respuesta a la señal de sincronización (MKMS) recibida, el uso de la clave de datos (PKMS) para descifrar los datos cifrados recibidos;

sustituir la señal de sincronización recibida en el flujo de transporte MPEG por un paquete nulo, sustituyendo la clave de datos recibida en el flujo de transporte MPEG por un paquete nulo y descifrando los datos cifrados recibidos en consecuencia.

2. Método según la reivindicación 1, que comprende además recibir la clave de datos desde el transmisor antes de recibir la señal de sincronización.

3. Método según la reivindicación 1, en el que la clave de datos recibida desde el transmisor está cifrada, y en el que la sincronización del uso de la clave de datos para descifrar los datos cifrados recibidos comprende:

descifrar la clave de datos cifrada; y,

descifrar los datos cifrados recibidos según la clave de datos descifrada.

4. Método según la reivindicación 3, en el que la señal de sincronización comprende una señal de sincronización actual recibida en lugar de un paquete nulo actual en el flujo de transporte MPEG, en el que el descifrado de la clave de datos cifrada comprende descifrar la clave de datos cifrada según una clave de modificación, y en el que la clave de modificación está contenida en una señal de sincronización previamente recibida que se recibe en lugar de un paquete nulo previo en el flujo de transporte MPEG.

5. Método según la reivindicación 1, en el que la recepción de una señal de sincronización en lugar de un paquete nulo comprende recibir una señal de sincronización actual en lugar de un paquete nulo actual, en el que la recepción de la clave de datos comprende recibir una señal de clave de datos en lugar de un paquete nulo previo en el flujo de transporte MPEG, en el que el paquete nulo previo tuvo lugar antes del paquete nulo actual, y en el que la señal de clave de datos contiene la clave de datos recibida.

6. Método según la reivindicación 5, en el que la clave de datos recibida desde el transmisor está cifrada, y en el que la sincronización del uso de la clave de datos para descifrar los datos cifrados recibidos comprende;

descifrar la clave de datos cifrada; y,

descifrar los datos cifrados recibidos según la clave de datos descifrada.

7. Método según la reivindicación 6, en el que el descifrado de la clave de datos cifrada comprende descifrar la clave de datos cifrada según una clave de modificación, en el que el método comprende además recibir una señal de sincronización de clave de modificación en lugar de un paquete nulo que es anterior al paquete nulo previo, y en el que la señal de sincronización de clave de modificación contiene la clave de modificación.

8. Método según la reivindicación 1, en el que la recepción de una clave de datos comprende recibir una pluralidad de claves de datos, y en el que la sincronización del uso de la clave de datos para descifrar los datos cifrados recibidos comprende rotar las claves de datos según un patrón y descifrar los datos recibidos basándose en las claves de datos rotadas.

9. Método según la reivindicación 8, que comprende además recibir una identificación del patrón desde un transmisor.

10. Método según la reivindicación 1, en el que los datos comprenden vídeo.

11. Método según la reivindicación 1, que comprende además descifrar la clave de datos antes de descifrar los datos.

12. Método según la reivindicación 1, en el que la recepción de la clave de datos comprende recibir la clave de datos cifrada mediante una clave de segmento de mensaje, en el que la sincronización del uso de la clave de datos

para descifrar los datos cifrados recibidos comprende:

recibir una dirección de memoria que corresponde a una clave fija;

recuperar la clave fija de una memoria según la dirección de memoria;

producir la clave de segmento de mensaje a partir de la clave fija;

descifrar la clave de datos según la clave de segmento de mensaje; y,

descifrar al menos una parte de los datos según la clave de datos descifrada.

13. Método según la reivindicación 12, en el que la producción de la clave de segmento de mensaje comprende:

recibir una clave de modificación; y,

producir la clave de segmento de mensaje a partir de la clave fija y la clave de modificación.

14. Método según la reivindicación 13, en el que la dirección de memoria comprende una primera dirección de memoria, en el que la clave fija comprende una primera clave fija, en el que la clave de segmento de mensaje comprende una primera clave de segmento de mensaje, en el que la clave de modificación comprende una primera clave de modificación, en el que la primera clave de modificación recibida está cifrada, y en el que el método comprende además:

recibir una segunda dirección de memoria;

recuperar una segunda clave fija de una memoria basándose en la segunda dirección de memoria;

generar una segunda clave de segmento de mensaje según una segunda clave de modificación que tiene un valor predeterminado y según la segunda clave fija; y,

descifrar la primera clave de modificación según la segunda clave de segmento de mensaje.

15. Método según la reivindicación 14, en el que las direcciones de memoria primera y segunda son direcciones de memoria primera y segunda cifradas, y en el que el método comprende además descifrar las direcciones de memoria primera y segunda cifradas usando una tercera clave fija.

16. Método según la reivindicación 1, en el que la recepción de datos cifrados en el flujo de transporte MPEG comprende recibir al menos un programa cifrado y datos PSI no cifrados, en el que los datos PSI no cifrados se refieren al programa cifrado, y en el que la sincronización del uso de la clave de datos para descifrar los datos cifrados recibidos comprende:

localizar el programa cifrado según los datos PSI no cifrados; y,

descifrar el programa cifrado localizado según la clave de datos.

17. Método según la reivindicación 1, en el que la recepción de una señal de sincronización comprende recibir un señal de sincronización cifrada, y en el que la sincronización, en respuesta a la señal de sincronización recibida, del uso de la clave de datos para descifrar los datos cifrados recibidos comprende sincronizar, en respuesta a la señal de sincronización cifrada recibida, el uso de la clave de datos para descifrar los datos cifrados recibidos.

18. Método según la reivindicación 1, en el que la clave de datos comprende una siguiente clave de datos cifrada, y en el que la recepción de una clave de datos (PKMS) comprende:

recibir un mensaje de clave de datos que contiene la siguiente clave de datos cifrada; y,

descifrar la siguiente clave de datos cifrada según una primera clave de modificación contenida en un mensaje de primera clave de modificación recibido antes de recibir el mensaje de clave de datos; en el que la recepción de una señal de sincronización comprende recibir un mensaje de segunda clave de modificación posterior al descifrado de la siguiente clave de datos cifrada de tal manera que el mensaje de segunda clave de modificación contiene una segunda clave de modificación; y, en el que la sincronización del uso de la clave de datos para descifrar los datos cifrados recibidos comprende conmutar en respuesta al mensaje de segunda clave de modificación el uso de una clave de datos activa para el uso de la siguiente clave de datos en el descifrado de los datos cifrados recibidos.

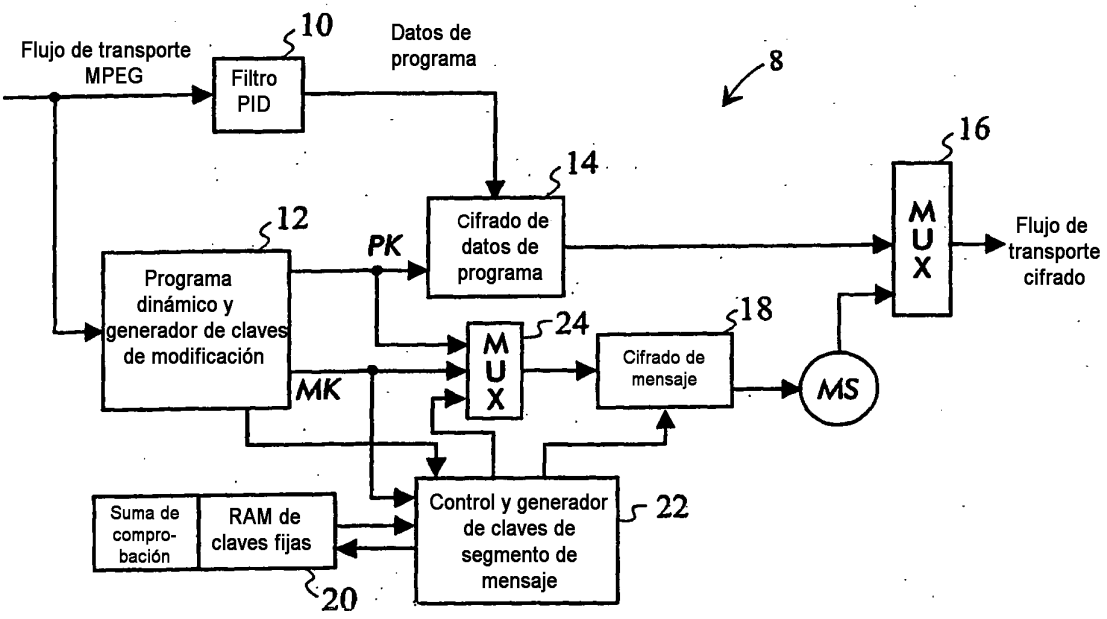
Flujo de transporte MPEG
10
Filtro PID
Datos de programa
8
16
14
12
PK
Cifrado de datos de programa
M U X
Flujo de transporte cifrado
Programa dinámico y generador de claves de modificación
24
18
MK
M U X
Cifrado de mensaje
MS
Suma de compro-bación
RAM de claves fijas
20
Control y generador de claves de segmento de mensaje
22

Figura 1

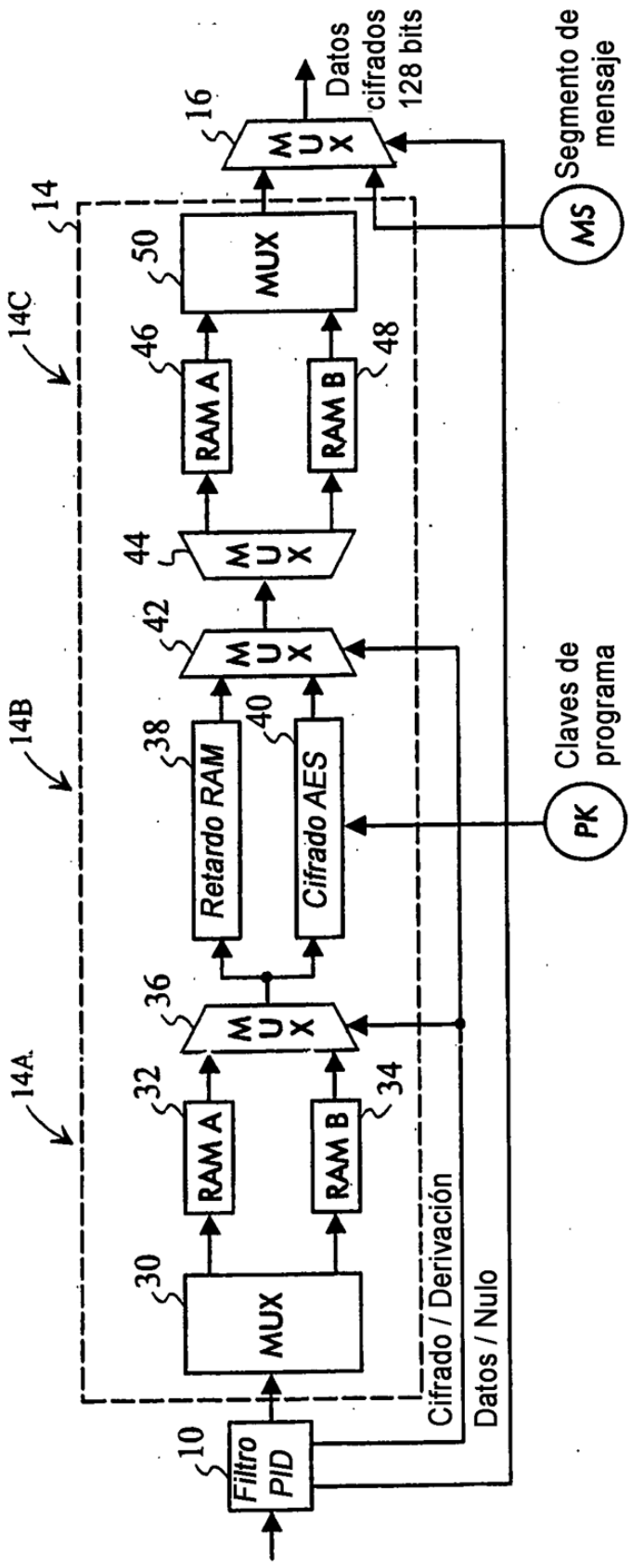
14
14A
14B
14C
10
Filtro PID
30
MUX
32
RAM A
34
RAM B
36
MUX
38
Retardo RAM
40
Cifrado AES
42
MUX
44
MUX
46
RAM A
48
RAM B
50
MUX
16
MUX
Datos cifrados 128 bits
Segmento de mensaje
MS
Claves de programa
PK
Cifrado / Derivación
Datos / Nulo

Figura 2

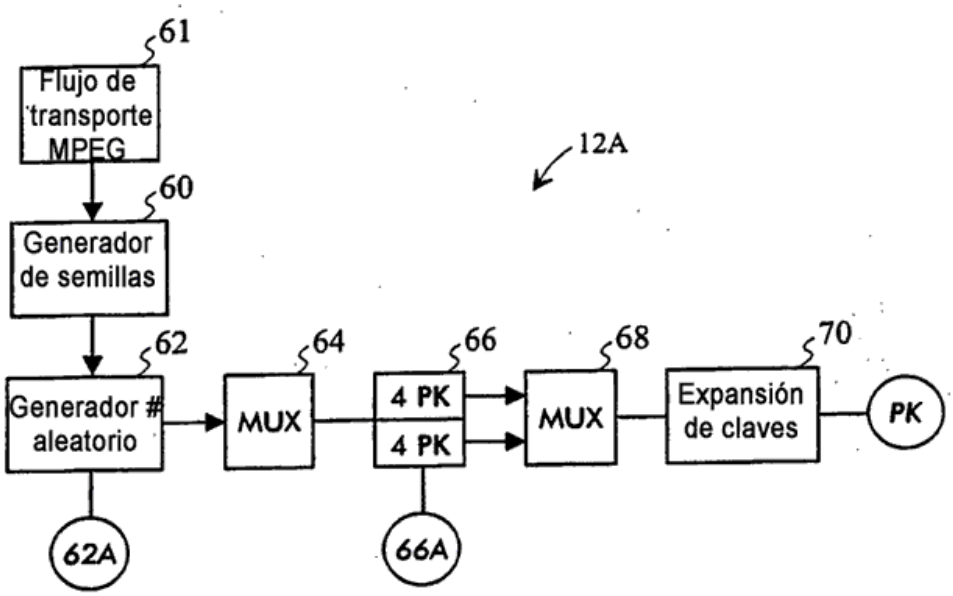

Figura 3

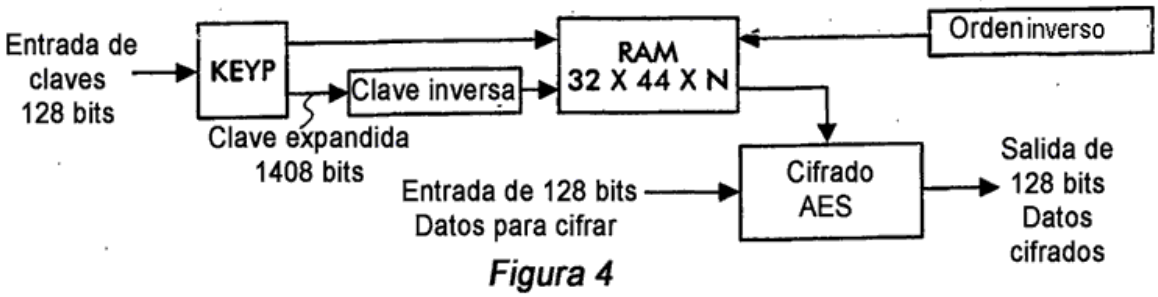

Figura 4

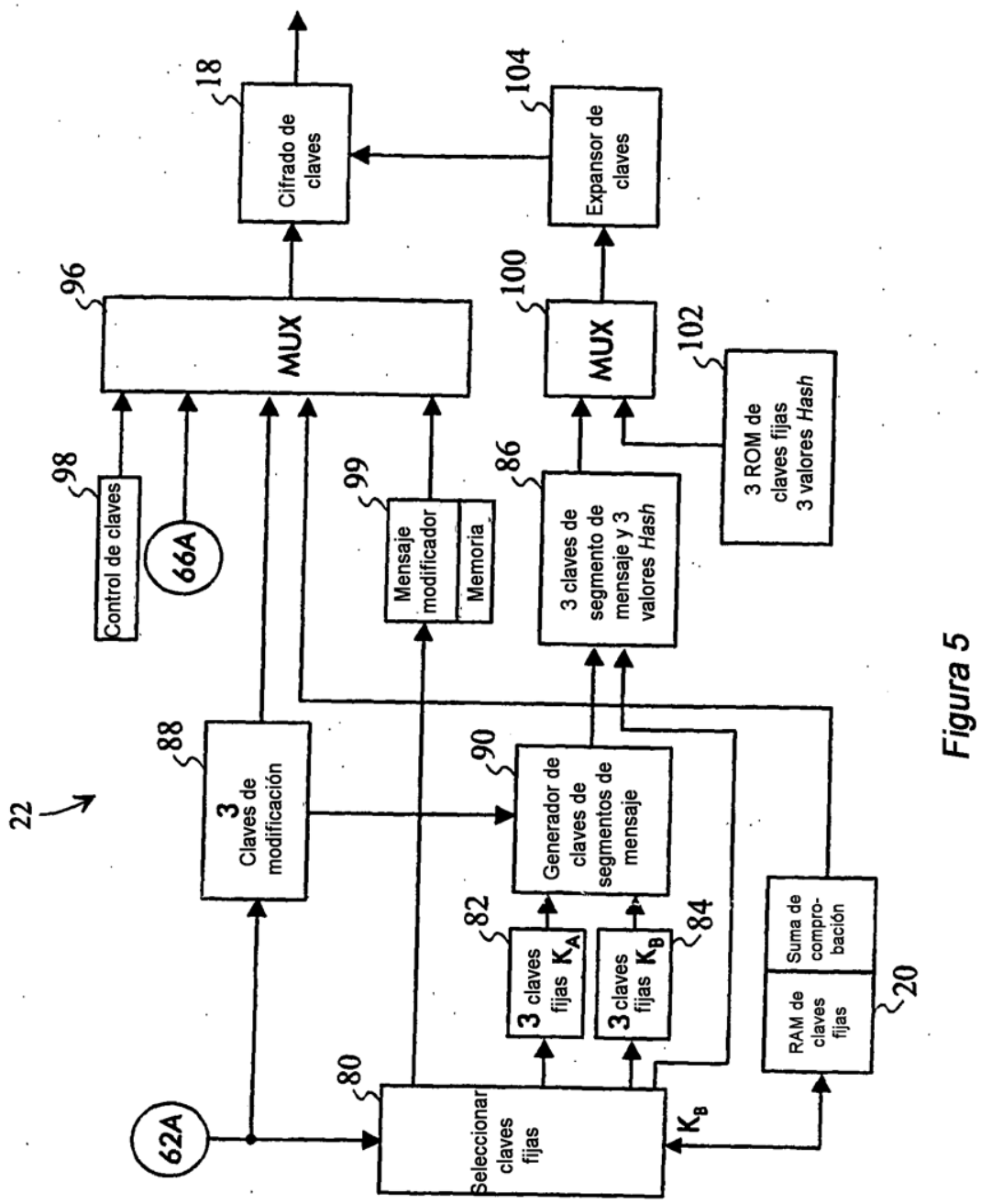
22
18
Cifrado de claves
104
Expansor de claves
96
MUX
100
MUX
102
3 ROM de claves fijas 3 valores Hash
98
Control de claves
66A
99
Mensaje modificador
Memoria
86
3 claves de segmento de mensaje y 3 valores Hash
88
3 Claves de modificación
90
Generador de claves de segmentos de mensaje
82
3 claves fijas K_A
84
3 claves fijas K_B
Suma de compro-bación
RAM de claves fijas
20
62A
80
Seleccionar claves fijas
K_B

Figura 5

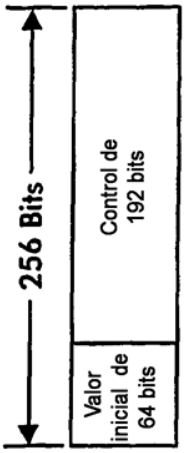

Figura 7

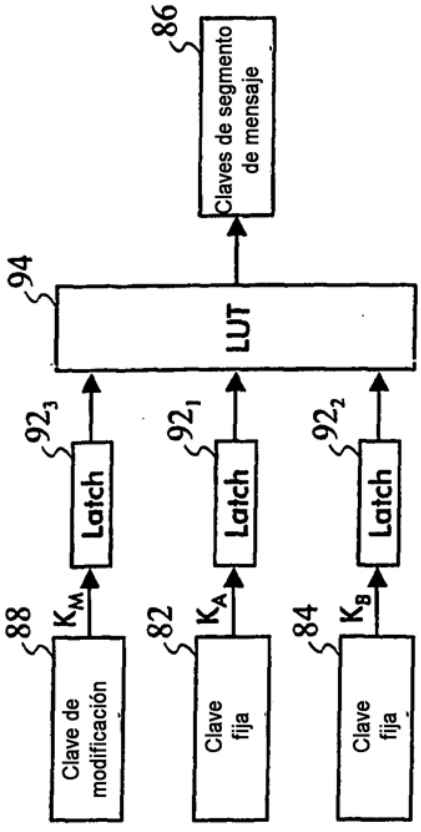

Tabla 0

K_M	K_A	K_B	K_0
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Tabla 1

K_M	K_A	K_B	K_1
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Tabla 2

K_M	K_A	K_B	K_2
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Tabla 3

K_M	K_A	K_B	K_3
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Figura 6

# de byte	Función	Descripción
0-3	Control de sistema	**Control**
4-12	Claves fijas y valores *Hash*	Punteros de dirección
13-23	TBD	

Figura 8

de byte 0-3

Bits		Acción	Función
0,1	XX	Ninguna	Control de copias
	XX	No más copias	
	XX	Copiar una vez	
	XX	Sólo mostrar	
2-7	XXXXXX	Reservado	
8-11	XXXX	# de tabla	Diseñar tabla
12-15	XXXX	Control	Rotación de claves
16-31	X—X	CRC	

Figura 9

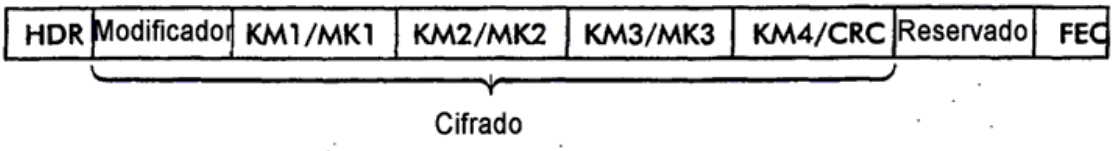

Figura 10

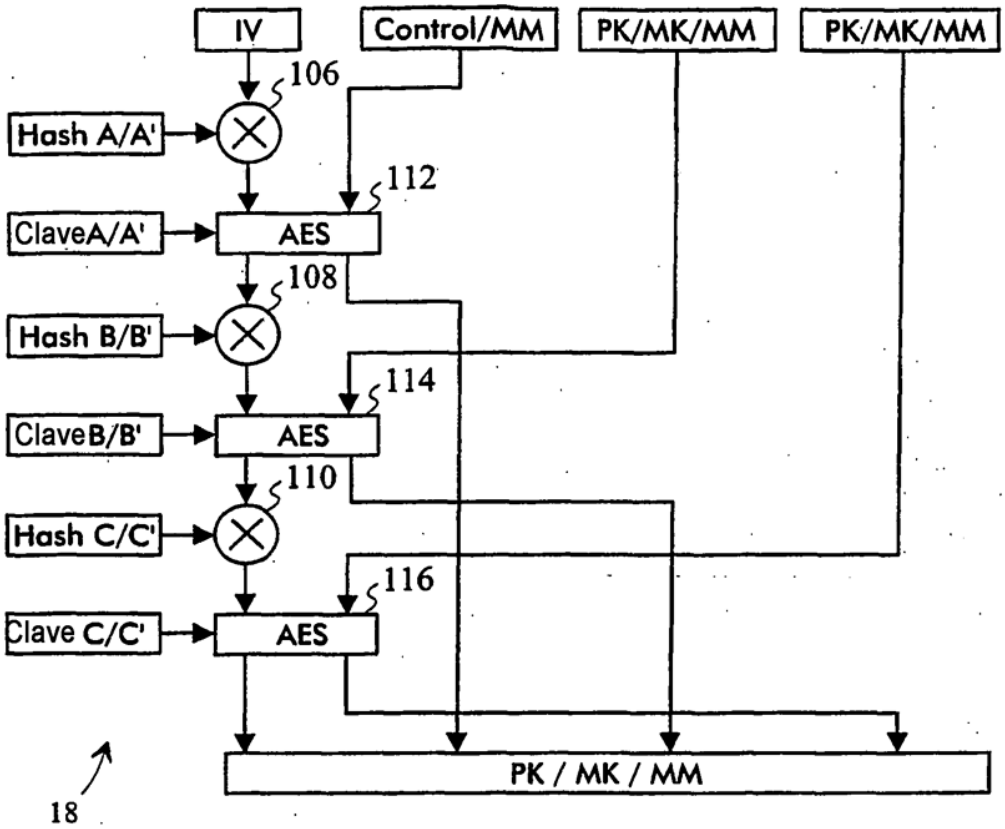

Figura 11

I.V. de 64 bits	Control de 64 bits	64 bits PK/MK MSB	64bits PK/MK LSB

Figura 12

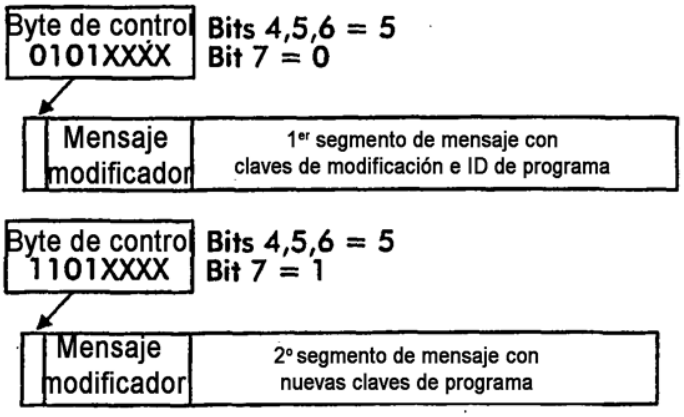

Figura 13

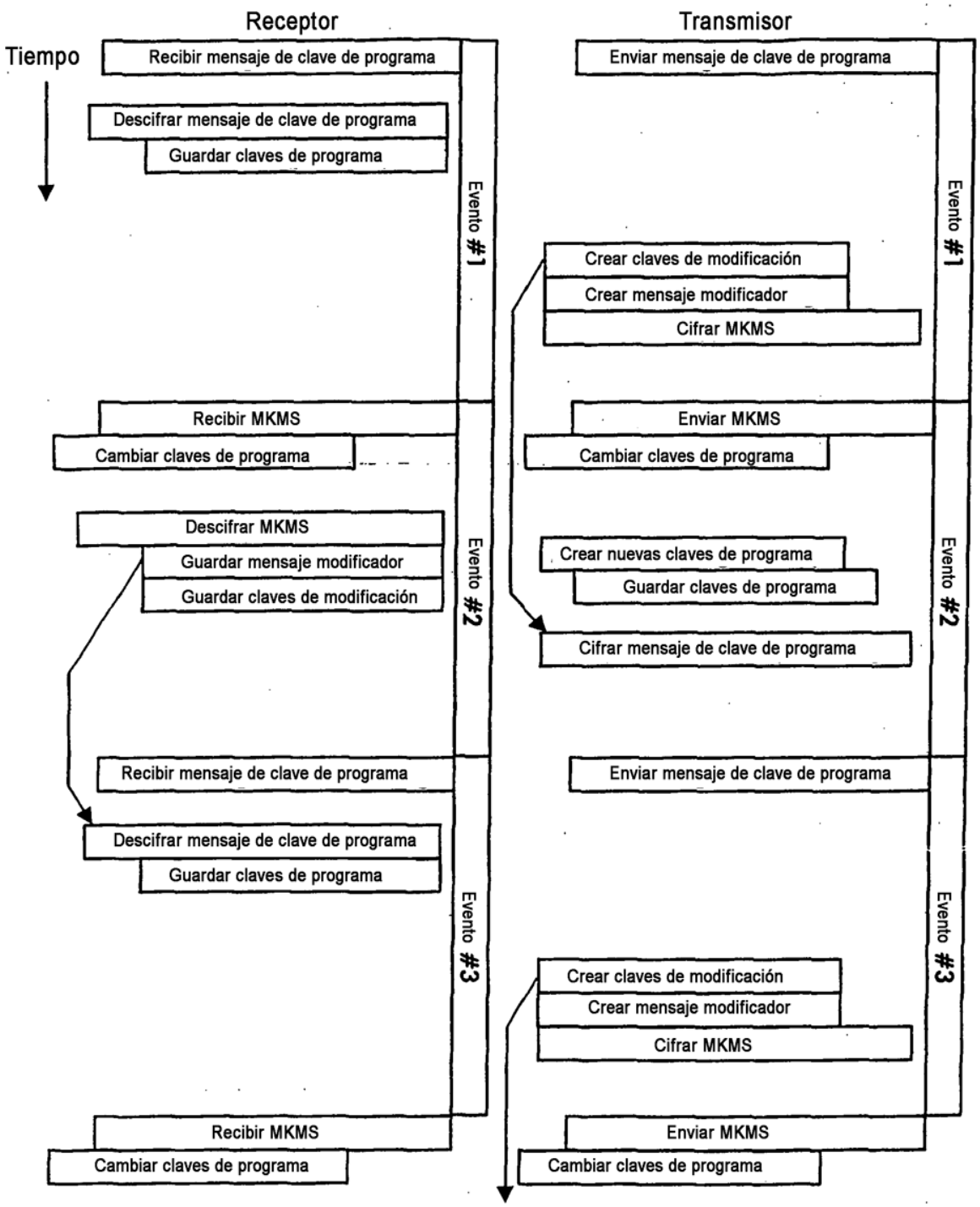
Receptor
Transmisor
Tiempo
Recibir mensaje de clave de programa
Descifrar mensaje de clave de programa
Guardar claves de programa
Evento #1
Enviar mensaje de clave de programa
Crear claves de modificación
Crear mensaje modificador
Cifrar MKMS
Recibir MKMS
Cambiar claves de programa
Enviar MKMS
Cambiar claves de programa
Descifrar MKMS
Guardar mensaje modificador
Guardar claves de modificación
Evento #2
Crear nuevas claves de programa
Guardar claves de programa
Cifrar mensaje de clave de programa
Recibir mensaje de clave de programa
Enviar mensaje de clave de programa
Descifrar mensaje de clave de programa
Guardar claves de programa
Evento #3
Crear claves de modificación
Crear mensaje modificador
Cifrar MKMS
Recibir MKMS
Cambiar claves de programa
Enviar MKMS
Cambiar claves de programa

Figura 14

	1	2	3	4	5	6	7	8	9	10	11
1	A	B	C	D	A	B	C	D	A	B	C
2	D	A	B	C	D	A	B	C	D	A	B
3	C	D	A	B	C	D	A	B	C	D	A
4	B	C	D	A	B	C	D	A	B	C	D
5	A	B	C	D	A	B	C	D	A	B	C
⋮	⋮										

Figura 15

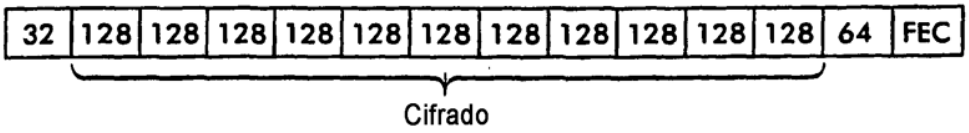

Figura 16

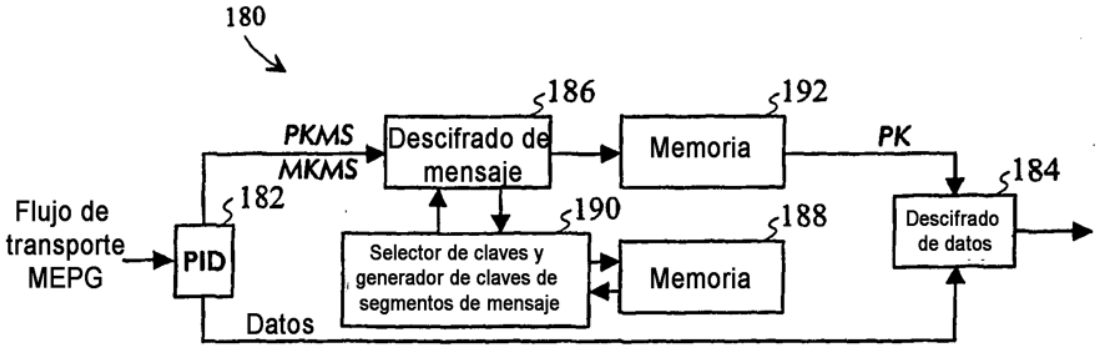

Figura 17

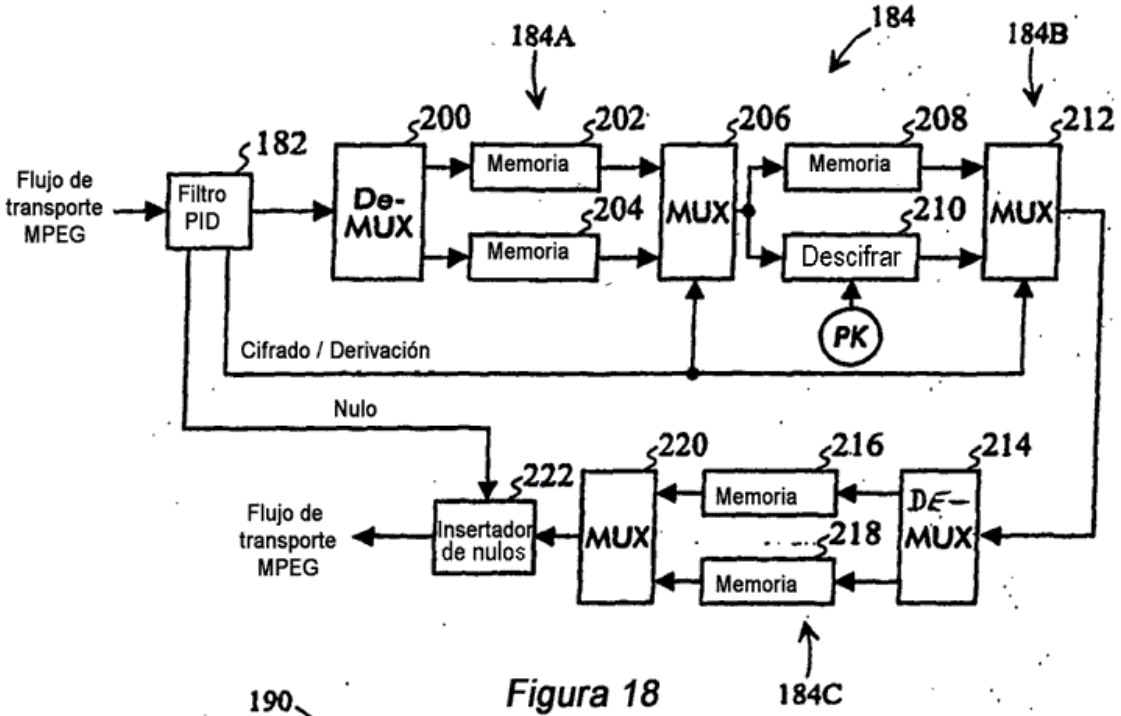

Figura 18

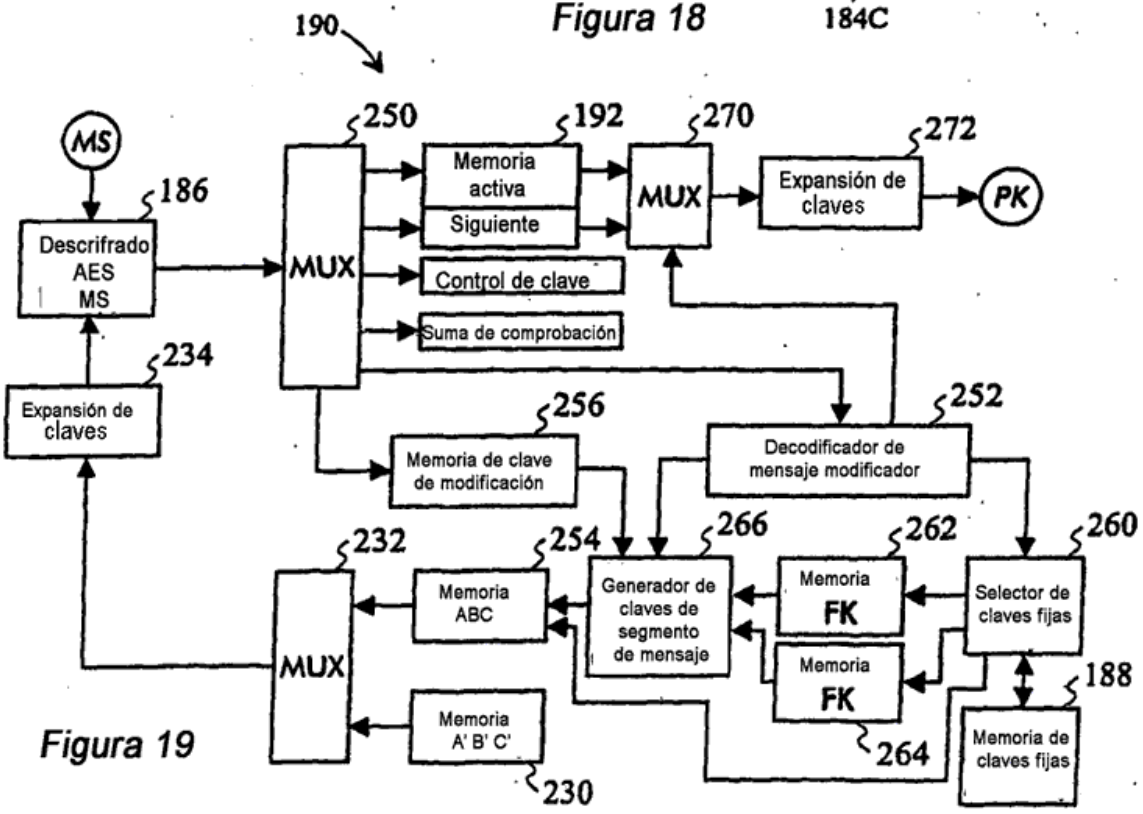

Figura 19

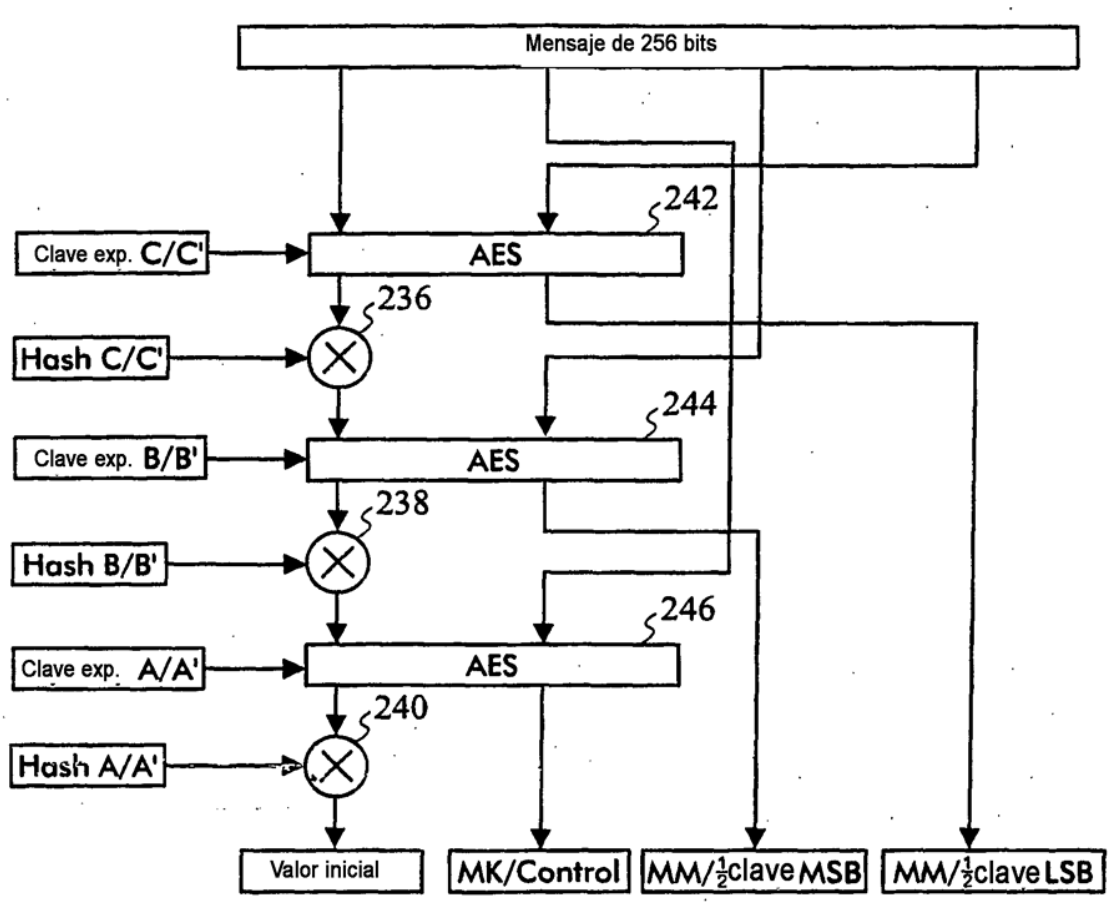
Mensaje de 256 bits
242
Clave exp. C/C'
AES
236
Hash C/C'
244
Clave exp. B/B'
AES
238
Hash B/B'
246
Clave exp. A/A'
AES
240
Hash A/A'
Valor inicial
MK/Control
MM/½clave MSB
MM/½clave LSB

Figura 20