(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 629 195**

(51) Int. Cl.:

H04H 60/73	(2008.01)
G10L 19/16	(2013.01)
H04N 21/84	(2011.01)
H04N 21/435	(2011.01)
H04L 9/32	(2006.01)
H04N 21/235	(2011.01)
H04N 19/40	(2014.01)
H04L 29/06	(2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **15.01.2014** **PCT/US2014/011695**

(87) Fecha y número de publicación internacional: **24.07.2014** **WO14113478**

(96) Fecha de presentación y número de la solicitud europea: **15.01.2014** **E 14702396 (4)**

(97) Fecha y número de publicación de la concesión europea: **17.05.2017** **EP 2946495**

(54) Título: **Codificación y descodificación de una secuencia de bits según un nivel de confianza**

(30) Prioridad:

21.01.2013 US 201361754893 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
07.08.2017

(73) Titular/es:

DOLBY LABORATORIES LICENSING CORPORATION (50.0%)
100 Potrero Avenue
San Francisco, CA 94103-4813, US y
DOLBY INTERNATIONAL AB (50.0%)

(72) Inventor/es:

SCHNEIDER, ANDREAS;
FERSCH, CHRISTOF;
WOLTERS, MARTIN;
RIEDMILLER, JEFFREY;
NORCROSS, SCOTT GREGORY y
GRANT, MICHAEL

(74) Agente/Representante:

LEHMANN NOVO, María Isabel

ES 2 629 195 T3

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Codificación y descodificación de una secuencia de bits según un nivel de confianza.

Referencia cruzada a solicitudes relacionadas

La presente solicitud invoca prioridad sobre la Solicitud de Patente Provisional de Estados Unidos No. 61/754,893, presentada el día 21 de enero de 2013.

Campo técnico

El presente documento se refiere a la transcodificación de metadatos. En particular, el presente documento se refiere a un método y sistema para transcodificar metadatos con complejidad computacional reducida.

Antecedentes

Actualmente se usan varios sistemas de renderización de audio de un solo canal y/o multicanal como, por ejemplo, los sistemas de renderización de audio multicanal 5,1, 7,1 o 9,1. Los sistemas de renderización de audio permiten p.ej. la generación de un sonido ambiente que se origina a partir de 5+1, 7+1 o 9+1 lugares de altavoz, respectivamente. Para una transmisión eficiente o para un almacenamiento eficiente de las señales de audio de un solo canal o multicanal correspondientes, se están usando sistemas de códec de audio (codificador / descodificador) como, por ejemplo, Dolby Digital (DD) o Dolby Digital Plus (DD+).

Puede haber una base significativa instalada de dispositivos de renderización de audio que se configuran para descodificar señales de audio que se han codificado usando un sistema de códec de audio particular (p.ej. Dolby Digital). Se puede hacer referencia al sistema de códec de audio particular como p.ej. un segundo códec de audio. Por otro lado, la evolución de los sistemas de códec de audio puede llevar a un sistema de códec de audio actualizado (p.ej. Dolby Digital Plus), al cual se puede hacer referencia p.ej. como un primer sistema de códec de audio. El sistema de códec de audio actualizado puede proveer características adicionales (p.ej. un número aumentado de canales) y/o calidad de codificación mejorada. Como tales, los proveedores de contenidos pueden inclinarse a proveer sus contenidos según el sistema de códec de audio actualizado.

Sin embargo, el usuario que tiene un dispositivo de renderización de audio con un descodificador del segundo sistema de códec de audio debería aún poder renderizar el contenido de audio que se ha codificado según el primer sistema de códec de audio. Ello se puede lograr por el así llamado transcodificador o convertidor que se configura para convertir el contenido de audio que se codifica según el primer sistema de códec de audio en contenido de audio modificado que se codifica según el segundo sistema de códec de audio.

Una necesidad adicional para la transcodificación puede surgir a lo largo de la cadena de distribución del contenido de audio. El contenido de audio se puede codificar por un proveedor de contenidos que usa un códec de audio que es apropiado para la producción y transmisión de contenido de audio (como, por ejemplo, el códec de audio Dolby E). El contenido de audio se puede distribuir usando dicho códec de audio orientado a la producción y el contenido de audio se puede transcodificar según un segundo códec de audio (como, por ejemplo, el códec sin pérdida Dolby TrueHD o como, por ejemplo, el códec Dolby Digital Plus o Dolby Digital).

El contenido de audio se relaciona, normalmente, con metadatos que se codifican en la secuencia de bits que representa el contenido de audio. Normalmente, el contenido de audio se divide en una secuencia de tramas, donde cada trama de contenido de audio comprende un número predeterminado de muestras (p.ej. 1024 muestras). Una trama de la secuencia de tramas se puede asociar a un contenedor respectivo o trama de metadatos. El contenedor de metadatos puede ser indicativo de información que describe la trama del contenido de audio con la cual se asocia al contenedor. Un ejemplo de dicha información que describe la trama pueden ser datos de sonoridad acerca de algunas o todas las muestras de la trama. De manera alternativa o adicional, el contenedor de metadatos se puede usar para transmitir datos auxiliares que pueden no asociarse de forma directa a la trama de contenido de audio correspondiente. Dichos datos auxiliares pueden p.ej. usarse para proveer a un descodificador de un sistema de códec de audio una actualización de firmware.

Además de transcodificar el contenido de audio de un primer sistema de códec de audio a un segundo sistema de códec de audio, el transcodificador normalmente también necesita transcodificar los metadatos asociados. Con el fin de reducir el costo de los transcodificadores / convertidores (que se implementan p.ej. dentro de descodificadores de salón), la complejidad computacional de la conversión entre un primer sistema de códec de audio y un segundo sistema de códec de audio debería ser relativamente baja. Este también debería ser el caso para la transcodificación de los metadatos. En el presente documento, se describen métodos y sistemas para la transcodificación, los cuales permiten la transcodificación de metadatos con una complejidad computacional reducida.

Según la solicitud internacional WO2011/015369A1, existen técnicas conocidas para la autenticación de secuencias de datos mediante el uso de valores criptográficos insertados en la secuencia. La solicitud de patente

US2003/0126432A1 describe el uso de claves criptográficas que comprenden claves públicas y privadas para la autenticación de contenido. La solicitud internacional WO2006/118896A2 describe técnicas para procesar metadatos en archivos, los metadatos incluyen valores *hash* y firmas digitales para autenticar los metadatos. La solicitud de patente US2010/0205162A1 describe una técnica de almacenamiento de archivos de medios de calidad asegurada que usa valores *hash* generados por la función *hash*. La solicitud de patente US2013/0123701A1 describe la aplicación de la función *hash* a una imagen que comprende datos de imagen y metadatos asociados.

Compendio de la invención

La invención provee un codificador configurado para generar una secuencia de bits codificada según la reivindicación 1 y un método correspondiente según la reivindicación 7.

Según un aspecto adicional, se proveen un descodificador relacionado configurado para recibir una secuencia de bits codificada según la reivindicación 10 y un método correspondiente según la reivindicación 14.

Según otro aspecto, se provee un medio de almacenamiento según la reivindicación 15.

Se debe notar que los métodos y sistemas, incluidas sus realizaciones preferidas según se describen en la presente solicitud de patente, se pueden usar de manera autónoma o en combinación con otros métodos y sistemas descritos en el presente documento. Además, todos los aspectos de los métodos y sistemas descritos en la presente solicitud de patente se pueden combinar de manera arbitraria. En particular, las características de las reivindicaciones se pueden combinar entre sí de manera arbitraria.

Breve descripción de las figuras

La invención se explica más abajo a modo de ejemplo con referencia a los dibujos anexos, en donde

la Figura 1a muestra un diagrama de bloques de una cadena de distribución de contenido de audio a modo de ejemplo que comprende un transcodificador;

la Figura 1b muestra una estructura a modo de ejemplo de una trama de metadatos;

las Figuras 2a y 2b muestran ejemplos de una propiedad de sello temporal usada en una trama de metadatos;

las Figuras 3a y 3b muestran ejemplos de una propiedad de desduplicación usada en una trama de metadatos;

las Figuras 4a y 4b muestran ejemplos de una propiedad de duplicación usada en una trama de metadatos; y

las Figuras 5a a 5d y la Figura 6 muestran transcodificadores conectados a MIC a modo de ejemplo.

Descripción detallada

Según se describe en la sección de antecedentes, el contenido de audio se asocia, normalmente, a metadatos y se codifica en una secuencia de bits conjunta que comprende una secuencia de tramas de contenido codificado y una secuencia asociada de contenedores de metadatos (a los que también se hace referencia como tramas de metadatos). La Figura 1 muestra un diagrama de bloques de un sistema de distribución 100 a modo de ejemplo para el contenido de audio. Los métodos y sistemas descritos en el presente documento se describen en el contexto de contenido de audio. Se debe notar, sin embargo, que los métodos y sistemas son aplicables a otros tipos de contenido como, por ejemplo, contenido de vídeo. En términos más generales, los métodos y sistemas descritos en el presente documento son aplicables a contenidos multimedia como, por ejemplo, audio y/o vídeo, en donde el contenido multimedia se asocia a metadatos.

El sistema de distribución 100 comprende un codificador 101 que se configura para codificar el contenido de audio y proveer una secuencia de bits codificada 110 (a la que también se hace referencia como la primera secuencia de bits codificada 110 o la secuencia de bits codificada de entrada 110). La primera secuencia de bits codificada 110 normalmente comprende una secuencia de tramas de contenido codificadas 111, en donde una trama de contenido codificada 111 se puede asociar a una trama de metadatos 112 correspondiente. El codificador 101 se configura para proveer una primera secuencia de bits codificada 110 que se codifica según un primer sistema de códec de audio. El primer sistema de códec de audio puede ser p.ej. uno de: Dolby E, Dolby Digital Plus, Dolby Digital Dolby True HD, Dolby Pulse, AAC y/o HE-AAC. Las tramas de contenido 111 pueden representar o pueden comprender un número predeterminado de muestras de contenido de audio, p.ej. 1536, 1024 o 512 muestras de contenido de audio.

La primera secuencia de bits codificada 110 se provee a un transcodificador 103 mediante un medio de transmisión o mediante un medio de almacenamiento 102. El transcodificador 103 se configura para transcodificar o convertir la primera secuencia de bits codificada 110 en una segunda secuencia de bits codificada 120 (a la que también se hace referencia como una secuencia de bits de salida 120), en donde la segunda secuencia de bits codificada 120 se codifica según un segundo sistema de códec de audio. El segundo sistema de códec de audio puede ser

diferente del primer sistema de códec de audio. Por otro lado, el segundo sistema de códec de audio puede ser igual al primer códec de audio, pero usa una configuración diferente, p.ej. una configuración diferente de velocidad binaria, de frecuencia de trama y/o de canal. El segundo sistema de códec de audio puede ser p.ej. uno de: Dolby E, Dolby Digital Plus, Dolby Digital Dolby True HD, Dolby Pulse, AAC y/o HE-AAC. De manera similar a la primera secuencia de bits codificada 110, la segunda secuencia de bits codificada 120 comprende una secuencia de tramas de contenido 121 y una secuencia de tramas de metadatos 122 correspondiente. Las tramas de contenido 121 de la segunda secuencia de bits codificada 120 pueden tener un tamaño de trama diferente del tamaño de trama de las tramas de contenido 111 de la primera secuencia de bits codificada 110. La segunda secuencia de bits codificada 120 se puede proveer a un descodificador 104 para renderizar el contenido de audio.

Las tramas de metadatos 112, 122 pueden tener una estructura predeterminada. En otras palabras, las tramas de metadatos 112, 122 pueden seguir una sintaxis predeterminada. A modo de ejemplo, las tramas de metadatos 112, 122 pueden seguir la así llamada sintaxis de trama de evolución ilustrada en la Tabla 1. La sintaxis de trama de evolución puede p.ej. usarse en el contexto de sistemas de códec de contenido multimedia estandarizados como, por ejemplo, el sistema de Difusión de Vídeo Digital (DVB, por sus siglas en inglés) y/o un sistema de códec de Grupo de Expertos en Imágenes en Movimiento (MPEG, por sus siglas en inglés). Se debe notar que la sintaxis de trama de metadatos que se muestra en la Tabla 1 y las siguientes tablas es solamente un ejemplo. Son posibles variaciones o modificaciones de la sintaxis. En particular, la sintaxis que se muestra en el presente documento se puede extender mediante campos adicionales, p.ej. para proveer funcionalidades adicionales.

Tabla 1

Sintaxis	No. de bits	Comentarios
evo_trama()		
{		
clave_id = variable_bits (3);		
mientras ((id = **carga útil_id)** != FIN) {	5	
si (carga útil_id == 31) {		
carga útil_id += variable_bits (5);		
}		
carga útil_config();		
carga útil_tamaño = variable_bits (8);		
carga útil (carga útil_id, carga útil_tamaño);		
}		
protección();		
}		

La semántica de los parámetros de la trama de evolución que se muestra en la Tabla 1 puede ser la siguiente:

- clave_id puede ser un identificador de la clave criptográfica usada para la función *hash* (a saber, usada para calcular la protección_bits del campo de protección()).

- carga útil_id puede ser un identificador de la siguiente carga útil de aplicación; una carga útil_id FIN = "0000b" puede tener el significado de que ninguna otra carga útil se contiene en la presente evo_trama();

- carga útil_tamaño puede indicar el número de bytes en el siguiente campo de carga útil.

La sintaxis de trama de evolución especifica una trama de metadatos 112, 122 que puede comprender múltiples bloques de metadatos, en donde también se hace referencia a un bloque de metadatos como una carga útil. Como tal, una trama de metadatos 112, 122 puede comprender cero, uno o más bloques de metadatos, en donde cada bloque de metadatos es indicativo de un tipo particular y/o un aspecto particular de metadatos. Los ejemplos de los tipos de metadatos son

- metadatos descriptivos que describen aspectos particulares de la trama de contenido 111 con la cual se asocia a la trama de metadatos 112 (p.ej. información de tiempo y/o armónicos);

- metadatos no relacionados que comprenden datos auxiliares, los cuales no se relacionan directamente con la trama de contenido 111 (como, por ejemplo, actualizaciones de firmware para un descodificador objetivo del contenido de audio codificado);

- metadatos de control que se pueden usar para controlar la renderización de una o más muestras de la trama de contenido 111 con la cual se asocia a la trama de metadatos 112 (p.ej. valores de sonoridad para una o más muestras de la trama de contenido 111).

Como tal, la trama de metadatos 112 provee una estructura flexible que se puede expandir mediante bloques adicionales de metadatos, según sea necesario, para describir características adicionales del contenido de audio codificado o para transmitir datos auxiliares adicionales dentro de la secuencia de bits 110. En caso de que no se transmitan metadatos junto con una trama de contenido 111, la trama de metadatos 112 puede no comprender bloques de metadatos, lo cual se puede indicar en la sintaxis de la trama de evolución de la Tabla 1 mediante una carga útil_id que corresponde a un "FIN" ID (identificador) predeterminado.

En el presente documento, se propone añadir un descriptor a un bloque de metadatos, en donde el descriptor describe una o más características o propiedades de los metadatos comprendidos dentro del bloque de metadatos asociados. Se hace referencia a dicho descriptor como "carga útil_config()" en la sintaxis de la trama de evolución que se muestra en la Tabla 1. El descriptor se puede usar por un transcodificador para llevar a cabo una transcodificación eficiente del bloque de metadatos, sin necesidad de analizar los metadatos comprendidos dentro del bloque de metadatos asociados. Como resultado, la complejidad de la transcodificación de metadatos se puede recudir de manera significativa.

En otras palabras, el presente documento describe métodos de transcodificación de bloques de metadatos (a los que también se hace referencia como cargas útiles) dentro de una trama de metadatos 112 (p.ej. dentro de la trama de evolución que se muestra en la Tabla 1) de una secuencia de bits codificada a otra secuencia de bits. Las funciones de transcodificación se pueden guiar por campos específicos dentro de cada carga útil (p.ej. el campo "carga útil_config()" de un bloque de metadatos, como se muestra en la Tabla 1). Las funciones de transcodificación se pueden entonces especificar de modo que las cargas útiles individuales se pueden transcodificar, de manera apropiada, de una secuencia codificada a otra secuencia codificada, sin necesidad de extraer o interpretar la esencia de los parámetros de metadatos subyacentes del bloque de metadatos (a saber, sin la necesidad de extraer o interpretar la esencia de las cargas útiles subyacentes).

La Figura 1b muestra una estructura a modo de ejemplo de una trama de metadatos 130 (p.ej. la trama de metadatos 112). La trama de metadatos 130 puede comprender un encabezamiento de trama 131 que es indicativo de información genérica acerca de la estructura de la trama de metadatos 130 y la asociación de la trama de metadatos 130 con la trama de contenido 111 de la secuencia de bits codificada 110. El encabezamiento de trama 131 puede comprender algunos o todos los campos de la trama de evolución de la Tabla 1 que no se relacionan con las cargas útiles de la trama. Además, la trama de metadatos 130 puede comprender uno o más bloques 140 de metadatos (a los que también se hace referencia como carga útil de metadatos 140). Un bloque 140 de metadatos puede comprender un encabezamiento de bloque 141, el cual puede ser indicativo del tamaño del bloque 140 de metadatos (al que se hace referencia como carga útil_tamaño en la Tabla 1). Asimismo, el bloque 140 de metadatos puede comprender un descriptor 142 (al que se hace referencia como carga útil_config() en la Tabla 1), en donde el descriptor 142 puede ser indicativo del tipo de metadato y/o de una o más propiedades del metadato, el cual se comprende en el campo de datos 143 (a saber, la carga útil() que se muestra en la Tabla 1) del bloque 140 de metadatos.

Un descriptor 142 a modo de ejemplo de un bloque 140 de metadatos para una trama de evolución, a saber, un campo "carga útil_config()" a modo de ejemplo, se muestra en la Tabla 2. Se puede ver que el descriptor 142 puede comprender o puede ser indicativo de una o más propiedades de los metadatos comprendidos dentro del bloque 140. En el ejemplo de la Tabla 2, las propiedades son

- un parámetro de sello temporal indicativo de una muestra del contenido de audio, a la cual se aplica el metadato del bloque 140. El sello temporal puede indicar una muestra que se comprende dentro de la trama de contenido 111

que se asocia a la trama de metadatos 112 del bloque 140. De manera alternativa o adicional, el sello temporal se puede configurar para asumir valores suficientemente grandes, para indicar una muestra que se comprende dentro de una trama de contenido que sucede a la trama de contenido 111 que se asocia a la trama de metadatos 112 del bloque 140.

- un parámetro de duración indicativo del número de muestras (comenzando por la muestra indicada por el sello temporal), para las cuales se aplica el metadato del bloque 140.

- una bandera de transcodificación (a la que se hace referencia como una bandera "no transcodificar" en la Tabla 2) que provee una orden a un transcodificador sobre si transcodificar o no el bloque 140 de metadatos. Si se establece la bandera "no transcodificar", el transcodificador puede simplemente ignorar o eliminar el bloque 140 de metadatos cuando transcodifica la secuencia de bits de entrada 110. Ello puede ser útil en caso de metadatos que son relevantes solamente para el primer sistema de códec de la secuencia de bits de entrada 110 y no tiene sentido para cualquier otro sistema de códec a los cuales se puede transcodificar la secuencia de bits 110 (como es el caso p.ej. de un control de redundancia cíclica (CRC, por sus siglas en inglés) que se genera en datos comprendidos dentro de la secuencia de bits de entrada 110. Un CRC normalmente solo tiene sentido si los datos codificados no se modifican, de modo que no exista necesidad de transcodificar el CRC). En términos más generales, la bandera de transcodificación se puede usar para identificar metadatos que son solo útiles durante el proceso de descodificación de la secuencia de bits de entrada dentro del transcodificador (y, por lo tanto, no requerida para el proceso de recodificación subsiguiente para generar la secuencia de bits de salida).

- una bandera de duplicar que provee una orden a un transcodificador sobre si duplicar o no los metadatos comprendidos dentro del bloque 140, cuando el tamaño de la trama de contenido 111 antes y después de la transcodificación difiere.

- una bandera de desduplicar que provee una orden a un transcodificador sobre si eliminar o no los duplicados de los metadatos comprendidos dentro del bloque 140, cuando el tamaño de la trama de contenido 111 antes y después de la transcodificación difiere.

- un parámetro de prioridad que provee una indicación de la importancia relativa de los metadatos comprendidos dentro del bloque 140. El transcodificador puede usar el parámetro de prioridad para seleccionar uno o más bloques 140 de una trama de metadatos 130, p.ej. si la velocidad binaria permitida de la segunda secuencia de bits 120 transcodificada se reduce con respecto a la velocidad binaria de la primera secuencia de bits 110.

- una bandera de asociación (a la que se hace referencia como la bandera "ahora_o_nunca" en la Tabla 2) que provee una indicación al transcodificador sobre si los metadatos comprendidos dentro del bloque 140 se asocian o no a la trama de contenido 111 correspondiente. Como tal, si se establece la bandera "ahora_o_nunca", el transcodificador es consciente del hecho de que los metadatos comprendidos dentro del bloque 140 deben transcodificarse inmediatamente o deben suprimirse (ya que la bandera "ahora_o_nunca" indica que el descodificador no puede usar los metadatos si los metadatos se retrasan).

Tabla 2

Sintaxis	No. de bits	Comentarios
carga útil_config()		
{		
sello temporal_presente;	1	
si (sello temporal_presente) {		
sello temporal = variable_bits (11);		
}		
duración presente;	1	
si (duración_presente) {		

duración = variable-bits (11);		
}		
no_transcodificar;	1	
si (!no_transcodificar) {		
duplicar;	1	
desduplicar;	1	
prioridad;	5	
ahora_o_nunca;	1	
ajustado_acoplamiento	2	
}		
}		

En otras palabras, la semántica de los parámetros de propiedad del descriptor 142 que se muestra en la Tabla 2 puede ser la siguiente:

- un parámetro de sello temporal que indica el desplazamiento en muestras del comienzo de la trama de contenido 111 a la cual pertenece la carga útil 143 en cuestión;

- un parámetro de duración que indica el tiempo en muestras para las cuales la carga útil 143 en cuestión permanece válida;

- una bandera no_transcodificar que señaliza si la carga útil 143 en cuestión debe descartarse cuando se transcodifica (bandera = 1) o si la transcodificación puede ocurrir (bandera = 0);

- una bandera de duplicar que - cuando se establece en 1 - señaliza que la carga útil 143 en cuestión necesita repetirse durante la transcodificación de modo que aparece en bloques transcodificados 140 entre el sello temporal y sello temporal + duración. La bandera de duplicar se puede establecer, p.ej. para los datos de sonoridad para indicar que las tramas tienen la misma normalización de diálogo. En términos más generales, la bandera de duplicar se puede establecer para metadatos que no tienen una noción de tiempo. La bandera de duplicar no se establece normalmente para datos que admiten el concepto de tiempo por sí mismos, como p.ej. la secuencia de bits de un códec. En otras palabras, los metadatos que se miden internamente pueden no proveerse con una bandera de duplicar y/o desduplicar que está establecida, en donde el término "que se miden internamente" significa que solamente la secuencia exacta de bloques de metadatos es significativa, a saber, una repetición o desduplicación invalidaría los metadatos. Un ejemplo de metadatos que se miden internamente es una secuencia de bits diferente (diferente del contenido comprendido dentro de las tramas de contenido) que se incorpora a una secuencia de bloques de metadatos de una secuencia de tramas de metadatos. La carga útil de dicha secuencia de bits no debe repetirse o desduplicarse nunca. De lo contrario, la secuencia de bits se repetirá en partes o se troceará parcialmente. Otro ejemplo de datos medidos internamente son los datos binarios, como un programa ejecutable. Si dichos datos binarios se transmiten en múltiples bloques de metadatos de múltiples tramas de metadatos, entonces la duplicación o desduplicación de bloques de metadatos invalidará el significado de los datos binarios.

- una bandera de desduplicar: la bandera de desduplicar puede asegurar que durante la transcodificación cada bloque de metadatos de un id particular dentro de la misma trama de metadatos de salida más allá de la primera que tiene dicha bandera establecida en uno se puede eliminar. La bandera de desduplicar se puede establecer p.ej. para datos de sonoridad como normalización de diálogo que no necesitan ser tiempos múltiples presentes por trama de metadatos de salida 122.

- una bandera "ahora_o_nunca" que indica que una carga útil no debe retrasarse durante la transcodificación.

- un parámetro de procesamiento MIC, al que se hace referencia como un parámetro "ajustado_acoplamiento" en la Tabla 2. El parámetro de procesamiento MIC puede p.ej. usarse en el contexto de un transcodificador conectado a MIC según se describe más abajo, para informar al transcodificador conectado a MIC sobre cómo manejar los metadatos de una trama de metadatos particular que se asocia a una trama de contenido particular, en caso de una modificación de las muestras de la señal comprendida dentro de la trama de contenido. La función del parámetro de procesamiento MIC se describirá en mayor detalle más abajo, cuando se describan las funciones de un transcodificador conectado a MIC.

Tabla 3

Sintaxis	No. de bits	Comentarios
carga útil (id, tamaño)		
{		
para (i = 0; i < tamaño; i++) {		
carga útil_bytes[i];	**8**	
}		
}		

La Tabla 3 muestra la sintaxis de un campo de datos 143 a modo de ejemplo de un bloque 140 de metadatos.

Según se describe más arriba, la sintaxis de secuencia de bits para transportar metadatos (a saber, la trama de metadatos 130 que comprende un bloque 140 de metadatos) puede definir propiedades genéricas de metadatos (p.ej. comprendidas en el descriptor 142, a saber, en el campo carga útil_config() que se muestra en la Tabla 2). Dichas propiedades permiten un copiado simple de los metadatos de una (a saber, primera) secuencia de bits de entrada 110 a una (a saber, segunda) secuencia de bits de salida 120, incluso si el primer códec (usado para codificar la secuencia de bits de entrada 110) y el segundo códec (usado para codificar la secuencia de bits de salida 120) usan diferentes alineaciones de tramas. La manera en la que el copiado de los metadatos se lleva a cabo se guía por las propiedades comprendidas dentro del descriptor 142. Lo único que podría necesitar cambiar durante el proceso de transcodificación pueden ser las mismas propiedades. Sin embargo, la modificación de las propiedades comprendidas dentro del descriptor 142 no requiere conocimiento sobre el significado real de los metadatos comprendidos dentro del campo de datos 143 del bloque 140.

A continuación, las propiedades a modo de ejemplo que se muestran en la Tabla 2 se describen en mayor detalle. En particular, se describe cómo el transcodificador 103 puede hacer uso de una o más de las propiedades indicadas por el descriptor 142 para llevar a cabo una transcodificación eficiente de los metadatos comprendidos dentro de un bloque 140 de metadatos.

Las Figuras 2a y 2b ilustran el uso del parámetro de sello temporal comprendido dentro del descriptor 142 de un bloque 140 de metadatos. En la Figura 2a se ilustra cómo el parámetro de sello temporal 201 se puede actualizar por un transcodificador 103, cuando se transcodifican metadatos de la primera secuencia de bits 110 a la segunda secuencia de bits 120. En el ejemplo ilustrado, el parámetro de sello temporal 201 indica la posición de una muestra particular 202 respecto al fin de la trama de contenido 111 asociada (a saber, respecto a la muestra más reciente). Como tal, el parámetro de sello temporal 201 es indicativo de un "retardo" de la muestra 202 con respecto a la muestra más reciente comprendida dentro de la trama de contenido 111. En el ejemplo ilustrado de la Figura 2, las tramas de contenido 121 de la segunda secuencia de bits 120 tienen un tamaño diferente, en particular mayor, respecto a las tramas de contenido 111 de la primera secuencia de bits 110. Como resultado, la muestra particular 202 se puede ubicar en una posición relativa diferente dentro de la trama de contenido 121 de la segunda secuencia de bits 120 en comparación con la posición relativa dentro de la trama de contenido 111 de la primera secuencia de bits 110. En particular, la muestra particular 202 puede mostrar un "retardo" diferente con respecto a la muestra más reciente comprendida dentro de la trama de contenido de salida 121 que con respecto a la muestra más reciente comprendida dentro de la trama de contenido de entrada 111. Como resultado, el parámetro de sello temporal 201 comprendido dentro de la trama de metadatos 112 de la primera secuencia de bits 110 puede necesitar modificarse, cuando se inserta en la trama de metadatos 122 de la segunda secuencia de bits 120, y producir así el parámetro de sello temporal transcodificado 203.

La Figura 2b ilustra la posibilidad de mover bloques de metadatos 140 dentro de la secuencia de bits 110, 120. Ello puede ser útil para suavizar la velocidad binaria de la secuencia de bits 120 luego de la transcodificación. A modo de ejemplo, los metadatos de un bloque particular 140 en la trama de metadatos 112 se pueden asociar a la muestra particular 202 en la trama de contenido 111 (indicada por el parámetro de sello temporal 211). Como se describe más arriba, la ubicación de la muestra particular 202 se puede indicar respecto a la última muestra, a saber, la más reciente, de la trama de contenido de entrada 111. Si no es esencial que los metadatos del bloque particular 140 lleguen directamente después a la trama de contenido 121 que comprende la muestra 202 (como se puede indicar por la bandera de asociación (a la que se hace referencia como bandera "ahora_o_nunca" en la Tabla 2), el bloque particular 140 se puede mover por el transcodificador a una trama de metadatos 222 de una trama de contenido 221 que es subsiguiente a la trama de contenido 121 que comprende la muestra 202. El transcodificador 103 puede actualizar el parámetro de sello temporal 213 de modo que señale la muestra correcta 202.

En particular, el parámetro de sello temporal 213 puede indicar la ubicación de la muestra 202 respecto a la última muestra, a saber, la más reciente, de la trama de contenido de salida 221 que la trama de metadatos de salida 222 que comprende el parámetro de sello temporal 213 con la que se lo asocia. A tal fin, el parámetro de sello temporal 213 puede asumir valores que superan el número de muestras comprendidas dentro de una trama de contenido 221. De manera similar, el parámetro de sello temporal 213 se puede configurar para asumir valores negativos. Dichos valores negativos se podrían usar para indicar una muestra 202 que se comprende en una trama de contenido futura, a saber, en una trama de contenido que es subsiguiente a la trama de contenido 221 asociada a la trama de metadatos 222 que comprende el parámetro de sello temporal 213. Mediante ello, los metadatos se pueden transmitir antes que la única o más muestras con las que se asocia (p.ej. a la que se aplicará).

Como tal, el parámetro de sello temporal 211 (posiblemente en combinación con la bandera de asociación) permite a un transcodificador 103 transmitir los metadatos asociados a un sello temporal 211 en una trama de metadatos 222 subsiguiente o precedente y ajustar el sello temporal 213 de modo que se refiera a la misma muestra MIC 202 (aunque después de la transcodificación, la muestra 202 no se comprende en la trama de contenido 221 que se asocia a la trama de metadatos 222 que comprende el bloque particular 140). Como resultado, el transcodificador 103 se provee con algo de flexibilidad para suavizar la velocidad binaria de la segunda secuencia de bits 120.

Se debe notar que -de manera similar al transcodificador 103- el codificador 101 se puede configurar para incluir metadatos para una muestra en una trama de metadatos subsiguiente. Como tal, el codificador 101 se puede configurar para generar un sello temporal 213 que señala una muestra 202 que se comprende en una trama de contenido 121 que no es la trama de contenido con la que se asocia la trama de metadatos que comprende el sello temporal 213.

Las Figuras 3a y 3b ilustran casos posibles de uso de la bandera de desduplicar indicada por el descriptor 142 de un bloque 140 de metadatos. En los casos ilustrados, las tramas de contenido 121 de la segunda secuencia de bits 120 representan un número mayor de muestras (a saber, tienen un tamaño de trama mayor) que las tramas de contenido 111 de la primera secuencia de bits 110. Si los tamaños de trama difieren, pueden ocurrir situaciones donde una sola trama de contenido 121 de la segunda secuencia de bits 120 comprende muestras de más de una trama de contenido 111 de la primera secuencia de bits 110. En dichos casos, los bloques 140 de metadatos pueden estar disponibles desde las más de una trama de metadatos 112 asociadas a las más de una trama de contenido 111 de la primera secuencia de bits 110. El transcodificador 103 tiene que decidir cuáles de los bloques 140 de metadatos se incluirán en la única trama de metadatos 122 de la única trama de contenido 121 de la segunda secuencia de bits 120. La bandera de desduplicar de un boque 140 particular puede indicar al transcodificador 103 que el bloque 140 particular no necesita insertarse en una trama de metadatos 122 de la segunda secuencia de bits 120 si los bloques de metadatos 140 de múltiples tramas de metadatos 112 de la primera secuencia de bits 110 se fusionarán. Como tal, el transcodificador 103 se puede configurar para suprimir o ignorar los bloques de metadatos 140 de tramas de metadatos 112 adicionales para los cuales se establece la bandera de desduplicar.

Ello se ilustra en la Figura 3a, donde la trama de contenido de salida 121 (a saber, la trama de contenido 121 de la secuencia de bits de salida 120) comprende las muestras de las tramas de contenido de entrada 111 y 311 (a saber, de las tramas de contenido 111, 311 de la secuencia de bits de entrada 110). El transcodificador 103 tiene que decidir cuáles de los bloques 140 de las tramas de metadatos de entrada 112, 312 (a saber, de las tramas de metadatos 112, 312 de la secuencia de bits de entrada 110) se incluirán en la trama de metadatos de salida 122 (a saber, de la trama de metadatos 122 de la secuencia de bits de salida 120) asociada a la trama de contenido de salida 121. En el ejemplo ilustrado de la Figura 3a, se supone que la bandera de desduplicar se establece al menos para uno o más bloques 140 de la trama de metadatos de entrada 312. Como tal, el transcodificador 103 se puede configurar para suprimir los bloques 140 de la trama de metadatos de entrada 312.

Se debe notar que la bandera de desduplicación del único o más bloques 140 de la trama de metadatos de entrada 112 también se puede establecer. El transcodificador 103 se puede configurar para suprimir solamente los bloques 140 de una segunda (o más) trama de metadatos 312 usada para construir la trama de metadatos de salida 122. En otras palabras, el transcodificador 103 se puede configurar para considerar la bandera de desduplicar solamente si se considerará más de una trama de metadatos de entrada 112 para generar una trama de metadatos de salida 122.

Como tal, la bandera de desduplicar se puede usar para evitar "duplicados" de un tipo particular de bloque de metadatos 140 mientras aún asegura que se incluya al menos un bloque de metadatos 140 del tipo particular.

La Figura 3b ilustra un caso a modo de ejemplo, donde no se establece la bandera de desduplicar. En el presente caso, el transcodificador 103 se puede configurar para considerar los bloques 140 de las múltiples tramas de metadatos de entrada 112 y 312 para construir la trama de metadatos de salida 122. En particular, el transcodificador 103 se puede configurar para insertar un bloque 140 de la trama de metadatos de entrada 312 en la trama de metadatos de salida 122, si no se establece la bandera de desduplicar (incluso en situaciones donde la trama de metadatos de salida 122 se genera a partir de múltiples tramas de metadatos de entrada 112, 312).

La bandera de desduplicar puede p.ej. usarse para identificar bloques de metadatos 140 que se insertan en múltiples tramas de metadatos 112, 312 subsiguientes (p.ej. en cada trama de metadatos 112, 312 de una secuencia de bits 110). Como tal, la bandera de desduplicar permite a un transcodificador 103 identificar fácilmente bloques de metadatos 140 que se pueden descartar (sin la necesidad de analizar los metadatos almacenados en el campo de datos 143 del bloque de metadatos 140). Como resultado, la complejidad computacional para tanscodificar metadatos se reduce. Por otro lado, una bandera de desduplicar que no se encuentra establecida indica que un bloque 140 correspondiente de metadatos no debería suprimirse. Ello se puede usar para datos auxiliares para asegurar que los datos auxiliares no se supriman, incluso si múltiples tramas de metadatos de entrada 112, 312 se transcodifican en una sola trama de metadatos de salida 122.

Las Figuras 4a y 4b ilustran un uso a modo de ejemplo de la bandera de duplicar indicada en el descriptor 142 de un bloque 140 de metadatos. En el caso ilustrado, la trama de contenido de entrada 111 comprende un número mayor de muestras (a saber, tiene un tamaño de trama más grande) que la trama de contenido de salida 121. Si los tamaños de trama difieren, pueden ocurrir situaciones donde las muestras de una sola trama de contenido de entrada 111 se comprenden dentro de más de una trama de contenido de salida 121, 321. Como consecuencia, el transcodificador 103 recibe una sola trama de metadatos de entrada 112 y tiene que decidir en cuál de las múltiples tramas de metadatos de salida 122, 322 ubicar un bloque 140 particular de metadatos. La bandera de duplicar se puede usar para indicar al transcodificador 130 si duplicar o no un bloque 140 particular de la trama de metadatos de entrada 112. Al establecer la bandera de duplicar, se puede indicar que los metadatos comprendidos dentro del bloque 140 se deben comprender dentro de cada trama de metadatos de salida 122, 322, como se muestra en la Figura 4a. Por otro lado, una bandera de duplicar no establecida indica que el bloque de metadatos 140 solo debería transmitirse una vez. Como tal, el transcodificador 103 inserta el bloque 140 de la trama de metadatos de entrada 112 únicamente en una sola de las múltiples tramas de metadatos de salida 122, 322 (como se ilustra en la Figura 4b).

Como se describe más arriba, el descriptor 142 de un bloque 140 de metadatos puede ser indicativo de una bandera de asociación (a la que se hace referencia como la bandera "ahora_o_nunca" en la Tabla 2). La bandera de asociación puede indicar que los metadatos comprendidos dentro del bloque 140 se pueden retrasar sin impactar en el contenido comprendido en la trama de contenido asociada. Como tal, la sintaxis del descriptor 142 puede permitir a un transcodificador 103 retrasar metadatos por una cantidad de tiempo arbitraria, si ello es una propiedad de los metadatos. Ello se puede indicar estableciendo la bandera ahora_o_nunca en 0. La bandera de asociación permite al transcodificador 103 transmitir los metadatos comprendidos dentro del bloque 140 p.ej. cuando el códec de audio subyacente puede "permitirse" la transmisión de los metadatos, p.ej. cuando las tramas de contenido comprenden silencio. Un ejemplo de metadatos que se pueden retrasar son datos auxiliares o datos binarios, como una actualización de firmware, que no necesitan transmitirse junto con una trama de contenido 121 particular.

Como se describe en el contexto de la Tabla 2, el descriptor 142 de un bloque 140 de metadatos puede ser indicativo de o puede comprender una propiedad de prioridad o un parámetro de prioridad. El parámetro de prioridad puede indicar una importancia relativa de los metadatos de un bloque 140 particular (p.ej. respecto a la importancia de otros bloques 140). Un transcodificador 103 puede decidir transcodificar solamente cierto número de bloques de metadatos 140 y descartar todos los otros bloques de metadatos en la trama de metadatos 112. Ello puede p.ej. requerirse cuando se transcodifica de una secuencia de bits de entrada 110 con velocidad binaria más alta a una secuencia de bits de salida 120 con velocidad binaria más baja. El parámetro de prioridad puede permitir al transcodificador 103 seleccionar aquellos bloques 140 de una trama de metadatos de entrada 112 que tengan las prioridades relativas más altas y descartar (o retrasar) aquellos bloques 140 que tengan las prioridades relativas más bajas.

Las aplicaciones y/o codificadores 101 pueden proveer múltiples grupos de metadatos en la misma trama de metadatos 112, cada uno con una prioridad diferente. Los múltiples grupos de metadatos se pueden asociar a diferentes calidades de metadatos. La prioridad de metadatos de calidad superior puede ser más baja que la prioridad de metadatos de calidad inferior. Como tal, el transcodificador 103 se puede configurar para degradar la calidad de los metadatos considerando el parámetro de prioridad. A modo de ejemplo, si las prioridades se establecen de modo tal que la escalabilidad es posible, a saber, cada grupo de metadatos se puede aplicar si todos los grupos de metadatos de la misma aplicación de una prioridad superior se transmiten, entonces un transcodificador puede, elegantemente, degradar la calidad de los metadatos sin tener que conocer el significado de

los metadatos. En particular, los múltiples grupos de metadatos pueden comprender metadatos incrementales, a saber, cada grupo de metadatos puede añadir algo de calidad al grupo de metadatos con la siguiente prioridad más alta. La calidad más alta de metadatos puede entonces proveerse combinando todos los grupos de metadatos (de la prioridad más alta a la prioridad más baja). Como tal, una trama de metadatos de entrada 112 puede comprender múltiples bloques 140 de metadatos incrementales, en donde el bloque 140 de metadatos con la prioridad más alta comprende una versión de los metadatos con una calidad mínima aceptable y en donde los bloques 140 con prioridad sucesivamente más baja comprenden versiones incrementales de metadatos que permiten aumentar, de forma incremental, la calidad de los metadatos. Como tal, el transcodificador 103 puede decidir sobre la calidad de metadatos que se incluyen en la segunda secuencia de bits 120 considerando los parámetros de prioridad de los múltiples bloques 140 de metadatos incrementales.

Como se indica en la sintaxis a modo de ejemplo de una trama de metadatos 112 que se muestra en la Tabla 1, la trama de metadatos 130 puede comprender un campo de protección. El campo de protección se puede usar para permitir al descodificador 104 verificar si el contenido de la trama de metadatos 130 y/o el contenido de la trama de contenido asociada se ha modificado y puede, por lo tanto, ser inválido. En otras palabras, el campo de protección puede permitir a un descodificador 104 verificar si los metadatos comprendidos dentro de una trama de metadatos 130 y/o dentro de una trama de contenido asociada es confiable o no. La Tabla 4 muestra una sintaxis a modo de ejemplo de un campo de protección de una trama de metadatos 130. El campo de protección se puede comprender dentro del encabezamiento 131 de la trama de metadatos 130.

Tabla 4

Sintaxis	No. de bits	Comentarios
protección()		
{		
protección_config_trama;	**2**	
protección_config_historia;	**2**	
conmutación (protección_config_trama) {		
caso 0:		
protección_bits_trama;	**0**	
corte;		
caso 1:		
protección_bits_trama;	**8**	
corte;		
caso 2:		
protección_bits_trama;	**32**	
corte;		
caso 3:		
protección_bits_trama;	**128**	

corte;		
}		
conmutación (protección_config_historia) {		
caso 0:		
protección_bits_historia;	**0**	
corte;		
caso 1:		
protección_bits_historia;	**8**	
corte;		
caso 2:		
protección_bits_historia;	**32**	
corte;		
caso 3:		
protección_bits_historia;	**128**	
corte;		
}		
}		

La semántica del campo de protección puede ser la siguiente:

- protección_bits_trama puede comprender la carga útil de protección truncada de la trama actual (que comprende la trama de contenido y/o la trama de metadatos asociados).

- protección_bits_historia puede comprender la carga útil de protección truncada de la trama actual y de la trama antes de la trama actual (que comprende la trama de contenido y/o la trama de metadatos asociados). Un esquema a modo de ejemplo para asegurar una secuencia de tramas se describe en el documento WO2011/0153 69.

Como tal, el campo de protección puede comprender uno o más valores criptográficos. Uno de los valores criptográficos se puede generar según los metadatos comprendidos dentro de una trama de metadatos actual (que comprende el campo de protección) y/o según la trama de contenido asociada a la trama de metadatos actual. Como tal, se puede asegurar que una trama de metadatos aislada y/o la trama de contenido asociada no se modifican. Otro de los valores criptográficos se puede generar según los metadatos comprendidos dentro de la trama de metadatos actual y dentro de una o más tramas de metadatos precedentes (así como en las respectivas tramas de contenido asociado). Como tal, se puede asegurar que las secuencias de tramas de contenido y/o tramas de metadatos no se modifican.

Un valor criptográfico se puede determinar en un codificador 101 aplicando una función unidireccional a un grupo de una o más tramas de metadatos 112, 312 y/o tramas de contenido asociadas 111, 311. En particular, un valor criptográfico se puede generar usando un valor clave y una función *hash* criptográfica (la así llamada función unidireccional). En particular, el valor criptográfico se puede generar calculando un valor HMAC-MD5 (código de

autenticación de mensaje *hash*) para los datos comprendidos dentro de una o más tramas de metadatos 112, 312 y para los datos comprendidos dentro de una o más tramas de contenido asociadas 111, 311. Asimismo, la generación del valor criptográfico puede comprender truncar el valor HMAC-MD5, p.ej. truncar a 16, 24, 32, 48, 64 o 128 bits. La truncación puede ser beneficiosa con el fin de reducir la sobrecarga requerida para el valor criptográfico en la secuencia de bits codificada 110 que comprende las tramas de metadatos 112, 312. Se debe notar que otras funciones *hash* como, por ejemplo, SHA-1 o SHA-256, se pueden usar en lugar de MD5. Asimismo, se debe notar que el codificador 101 se puede configurar para transmitir cero bits de un valor criptográfico, a saber, no transmitir valor criptográfico alguno, p.ej. en una situación donde no se requiere ninguna protección de los metadatos.

En mayor detalle, el valor criptográfico para una o más tramas de contenido 111, 311 y de una o más tramas de metadatos 112, 312 se puede determinar usando una función *hash* criptográfica *H*(.) y una clave "secreta" *C* (a la que también se hace referencia como clave de seguridad) que, normalmente, se rellena a la derecha con ceros adicionales para el tamaño de bloque de la función *hash H*(.) para determinar un código de autenticación de mensaje *hash* (HMAC, por sus siglas en inglés) de la única o más tramas de contenido 111, 311 y de una o más tramas de metadatos 112, 312. Dejemos que el signo || denote una concatenación y el signo ⊕ denote un exclusivo o, y el relleno externo *opad* = 0*x*5*c*5*c*5*c*...5*c*5*c* y el relleno interno *ipad* = 0*x*363636...3636 sean constantes de la longitud del tamaño de bloque de la función *hash H*(.), entonces el valor HMAC de la única o más tramas de contenido 111, 311 y de una o más tramas de metadatos 112, 312 se puede escribir como

$$HMAC(m) = H\left(\left(C \oplus opad\right) \| H\left(\left(C \oplus ipad\right) \| m\right)\right),$$

donde *m* es la secuencia de bits combinada de la única o más tramas de contenido 111, 311 y de una o más tramas de metadatos 112, 312. El tamaño de bloque usado con las funciones *hash* MD5 o SHA-1 o SHA-256 es, normalmente, de 512 bits. El tamaño de la salida de la función HMAC es igual al de la función *hash* subyacente, a saber, 128 bits en caso de MD5 o 160 bits en caso de SHA-1.

Como tal, el campo de protección puede comprender al menos dos valores criptográficos

- un valor criptográfico de trama (al que se hace referencia como "protección_bits_trama" en la Tabla 4) que es indicativo de la autenticidad de una trama de contenido individual 111 y su trama de metadatos asociados 112. El valor criptográfico de trama se puede usar para identificar si los datos de la trama de contenido individual 111 y su trama de metadatos asociados 112 se han cambiado. El valor criptográfico de trama se puede determinar usando un mensaje *m* que comprende la secuencia de bits de la trama de contenido individual 111 y de su trama de metadatos asociados 112 (o de la carga útil comprendida dentro de la trama de contenido individual 111 y de su trama de metadatos asociados 112.

- un valor criptográfico de historia (al que se hace referencia como "protección_bits_historia" en la Tabla 4) que es indicativo de la autenticidad de una secuencia de al menos dos tramas de contenido 111, 311 y sus al menos dos tramas de metadatos asociados 112, 312. El valor criptográfico de historia se puede usar para identificar si la secuencia de las al menos dos tramas de contenido 111, 311 y sus tramas de metadatos asociados 112, 312 se ha cambiado. El valor criptográfico de historia se puede determinar usando un mensaje *m* que comprende la secuencia de bits de las al menos dos tramas de contenido 111, 311 y sus al menos dos tramas de metadatos asociados 112, 312 (o de la carga útil allí comprendida).

Como se describe más arriba, los valores criptográficos se determinan usando una clave segura *C*, la cual es conocida normalmente solo para el codificador 101 y el descodificador 104. Según una realización de la invención, se propone permitir múltiples niveles de confianza permitiendo el uso de diferentes claves seguras *C* que proveen diferentes niveles de confianza. A modo de ejemplo, al menos se pueden proveer dos niveles de claves confiables

- una clave altamente segura C_1, que no se puede divulgar a las partes fuera de la entidad que provee los componentes 101, 103, 104 a lo largo de una cadena de distribución 100. Dicha entidad puede ser un proveedor de los sistemas de códec usados a lo largo de la cadena de distribución 100 (p.ej. Dolby Laboratories). En particular, dicha entidad puede ser el proveedor de los codificadores y los descodificadores usados a lo largo de la cadena de distribución 100. Al mantener la clave altamente segura sin divulgarse, se puede asegurar que un descodificador 104 que renderiza la señal de audio comprendida dentro de la secuencia de bits 120 recibida puede estar seguro de que los metadatos comprendidos dentro de las tramas de metadatos 122, 322 de la secuencia de bits 120 recibida son auténticos y no se han modificado de manera no autorizada a lo largo de la cadena de distribución 100.

- una clave segura moderada C_2, que se puede divulgar a otras partes, p.ej. partes que manejan algunos de los componentes 101, 103, 104 a lo largo de la cadena de distribución 100 (p.ej. licenciatarios del proveedor de los sistemas de códec). Si el descodificador 104 recibe una secuencia de bits 120 que se ha protegido usando la clave segura moderada C_2, el descodificador 104 sabe que la secuencia de bits 120 comprende metadatos (en las tramas de metadatos 122, 322) que se han manejado según algunas políticas del operador de la cadena de distribución

100, las cuales pueden ser diferentes de las políticas del proveedor de los sistemas de códec (que tiene la clave altamente segura C_1).

Una indicación de la clave segura *C* usada por el codificador 101 se puede proveer dentro de una trama de metadatos 130 (p.ej. dentro del encabezamiento 131 de la trama de metadatos 130). Ello se ilustra en la Tabla 1 que muestra el parámetro clave_id. El parámetro clave_id puede comprender un índice de un número predeterminado de claves seguras y permitir así al descodificador 104 determinar la clave segura *C*, la cual se ha usado para determinar el único o más valores criptográficos, en donde el único o más valores criptográficos se pueden comprender en el campo de protección() de la trama de metadatos 130, como se muestra en la Tabla 4). El descodificador 104 puede entonces usar la clave segura identificada para determinar el único o más valores criptográficos de igual manera que mediante el codificador 101 correspondiente. Se puede hacer referencia a los valores criptográficos que se determinan por el descodificador 104 como los valores criptográficos de verificación. Los valores criptográficos de verificación se comparan luego con los valores criptográficos almacenados en la trama de metadatos 103. En caso de concordancia, se confirma que la trama individual y/o que la secuencia de tramas no se han modificado. Por otro lado, en caso de discordancia, se confirma que la trama individual y/o que la secuencia de tramas se han modificado.

De manera alternativa o adicional a la provisión de una indicación de la clave segura dentro de la trama de metadatos 130, el descodificador 104 se puede configurar para determinar múltiples grupos de valores criptográficos de verificación usando múltiples claves seguras predeterminadas conocidas para el descodificador 104. Si uno de los grupos de valores criptográficos de verificación coincide con los valores criptográficos comprendidos en la trama de metadatos 130, el descodificador 104 sabe qué clave segura se ha usado y que la trama individual y/o que la secuencia de tramas no se han modificado. Por otro lado, una discordancia para todos los grupos de valores criptográficos de verificación indica que la trama individual y/o que la secuencia de tramas se han modificado.

La capacidad de detectar qué clave se ha usado para asegurar una secuencia de bits 110, 120 en los descodificadores 104 y transcodificadores 103 permite a las aplicaciones tomar decisiones más detalladas sobre qué hacer con los datos de diferente fiabilidad. Las decisiones podrían ser diferentes dependiendo de la clave segura detectada. En particular, se puede detectar la clave altamente segura, se puede detectar la clave segura moderada o puede no detectarse ninguna clave válida y el control de seguridad puede no pasarse.

Como tal, se pueden proveer niveles de fiabilidad cuando se usan múltiples claves seguras diferentes (que se fijan a diferentes niveles de confianza), en comparación con una solución que solo usa una sola clave segura, donde solo se puede tomar una decisión binaria sobre si se puede confiar o no en los datos.

Como se describe en el contexto de la Figura 1, una cadena de distribución 100 para el contenido de audio puede comprender un transcodificador 103 que se configura para convertir una secuencia de bits de entrada 110 en una secuencia de bits de salida 120. La transcodificación llevada a cabo por el transcodificador 103 puede relacionarse con la transcodificación de un primer sistema de códec de audio a un segundo, posiblemente diferente, sistema de códec de audio. De manera alternativa o adicional, la transcodificación se puede relacionar con un cambio de velocidad binaria de la secuencia de bits de salida 120 con respecto a la velocidad binaria de la secuencia de bits de entrada 110. El transcodificador 103 puede comprender un descodificador para descodificar la secuencia de bits de entrada 110 en una señal de audio MIC (modulación por impulso codificado). Además, el transcodificador 103 puede comprender un codificador para codificar la señal de audio MIC en la secuencia de bits de salida 120. Se puede hacer referencia a dicho transcodificador 103 como un transcodificador "conectado a MIC", ya que el único o más descodificadores (para descodificar la única o más secuencias de bits de entrada 110) se conectan al único o más codificadores (para codificar la única o más secuencias de bits de salida 120) mediante una MIC lineal.

El transcodificador 103 puede ser el así llamado transcodificador profesional que es un dispositivo usado por proveedores de contenido profesionales como, por ejemplo, las emisoras. Como se describe más arriba, el transcodificador 103 se puede configurar para aceptar la secuencia de bits de entrada 110 en un primer formato (p.ej. Dolby E) y para transcodificar la secuencia de bits de entrada 110 en un formato diferente (p.ej. Dolby Digital Plus). Dichos transcodificadores 103 normalmente incorporan uno o más descodificadores (para descodificar la secuencia de bits de entrada 110) y uno o más codificadores (para codificar la secuencia de bits de salida 120).

Un transcodificador conectado a MIC puede tener una o más etapas de procesamiento MIC entre el descodificador y el codificador. La nivelación de sonoridad es un ejemplo de dicho procesamiento MIC. Otros ejemplos de procesamiento MIC son la conversión de tasa, la submezcla de canales y/o la mezcla ascendente de canales de muestra.

Dichos transcodificadores conectados a MIC 103 plantean un desafío con respecto a cuestiones de autenticidad, protección y confianza descritas más arriba. Como se describe más arriba, una secuencia de bits de entrada 110 puede comprender tramas de metadatos 112, 312 que se protegen usando uno o más valores criptográficos (comprendidos p.ej. en el campo de protección de las tramas de metadatos 112, 312 como se muestra en las Tablas 1 y 4). Un transcodificador 103 conectado a MIC permite a un usuario modificar datos MIC derivados de las tramas

de contenido 111, 311 y posiblemente invalidar así los metadatos comprendidos dentro de las tramas de metadatos asociados 112, 312 y posiblemente comprometer así la fiabilidad de los metadatos.

En el presente documento, se describen un método y un sistema para asegurar la fiabilidad de metadatos en un transcodificador 103. En particular, el método y sistema descritos permiten mantener la fiabilidad de metadatos comprendidos en tramas de metadatos 112, 312, incluso cuando se usa un transcodificador 103 conectado a MIC.

Las Figuras 5a a 5d muestran transcodificadores conectados a MIC 503, 513, 523, 533 a modo de ejemplo, respectivamente. Los transcodificadores comprenden un descodificador 504 que se configura para convertir la secuencia de bits de entrada 110 (que comprende una secuencia de tramas de contenido 111 y una secuencia de tramas de metadatos asociados 112) en datos MIC y metadatos, respectivamente. El descodificador 504 se puede configurar para verificar la exactitud de la secuencia de bits de entrada 110 usando el esquema de protección descrito más arriba. A tal fin, el descodificador 504 puede ser consciente de algunas o todas las claves seguras predeterminadas.

Normalmente, un descodificador 504 provee un grupo no protegido de datos MIC y metadatos (p.ej. en una trama por base de trama). En otras palabras, el descodificador 504 normalmente descodifica cada trama de contenido 111 y trama de metadatos asociados 112 y provee el grupo respectivo de datos MIC y metadatos sin protección. Como tal, el descodificador 504 provee una secuencia de grupos de datos MIC y metadatos a partir de una secuencia correspondiente de tramas de contenido 111 y tramas de metadatos 112. La secuencia de grupos de datos MIC y metadatos se puede modificar por el transcodificador y puede entonces pasarse a un codificador 501 que se configura para convertir la secuencia de grupos (posiblemente modificados) de datos MIC y metadatos en la secuencia de bits de salida 120. En este contexto, el codificador 501 no puede, normalmente, verificar si la secuencia de grupos (posiblemente modificados) de datos MIC y metadatos se ha modificado de manera apropiada. En otras palabras, el codificador 501 puede no verificar la fiabilidad de la secuencia de grupos (posiblemente modificados) de datos MIC y metadatos.

En el presente documento, se propone permitir al descodificador 504 proveer uno o más valores de firma basados en uno o más grupos de datos MIC y metadatos y permitir así la protección de la conexión MIC entre el descodificador 504 y el codificador 501. Los valores de firma se pueden determinar de manera similar a los valores criptográficos, según se describe más arriba. Sin embargo, los valores de firma pueden hacer uso de un mensaje *m* que comprende uno o más grupos de datos MIC y metadatos (a diferencia de una o más tramas de contenido y tramas de metadatos asociados). En particular, el descodificador 504 se puede configurar para determinar

- un valor de firma de trama según un grupo individual de datos MIC y metadatos asociados; y
- un valor de firma de historia según dos o más grupos secuenciales de datos MIC y metadatos asociados.

En otras palabras, dentro del dominio MIC de un transcodificador 503 conectado a MIC (a saber, entre el descodificador 504 y el codificador 501), la fiabilidad del contenido se puede "proteger" usando una o más firmas (a las que también se hace referencia como valores de firma). El descodificador 504 se puede configurar para producir uno o más valores de firma como una salida. El único o más valores de firma se pueden calcular en la unión de datos MIC y metadatos regulares (tomados de la trama de contenido) y metadatos adicionales (tomados de la trama de metadatos asociados), producidos por el descodificador 504. Como tal, para cada trama de la secuencia de bits de entrada 110, uno o más valores de firma se pueden determinar según los grupos descodificados de datos MIC y metadatos. Dicho único o más valores de firma se pueden usar por el codificador 501 correspondiente para verificar si un grupo recibido de datos MIC y metadatos se ha modificado o no y/o si es confiable o no.

El codificador 501 acepta uno o más valores de firma como una entrada, junto con datos MIC, metadatos regulares y los metadatos adicionales. El codificador 501 puede entonces controlar los valores de firma contra las otras entradas (a saber, contra los grupos recibidos de datos MIC y metadatos). Si las otras entradas se han modificado / alterado, el control de firma fallará y el codificador tomará la acción apropiada. La verificación del único o más valores de firma se puede llevar a cabo en el codificador 501 determinando valores de firma de verificación según el o los grupos recibidos de datos MIC y metadatos (de manera similar, según se describe para los valores criptográficos).

Como tal, la fiabilidad de los datos MIC descodificados (y los metadatos asociados) se puede mantener dentro de un transcodificador 503 conectado a MIC permitiendo al descodificador 504 determinar uno o más valores de firma según los datos MIC descodificados y los metadatos asociados y permitiendo al codificador 501 correspondiente verificar la autenticidad de los datos MIC a codificarse (y los metadatos asociados) según el único o más valores de firma. La determinación del único o más valores de firma y su verificación se puede llevar a cabo según una o múltiples calves de seguridad niveladas C_1 y C_2, como se describe más arriba, en donde la única o más claves de seguridad pueden ser solamente conocidas para el descodificador 504 y codificador 501 y son, normalmente, desconocidas para una entidad que lleva a cabo el procesamiento MIC en la conexión entre el descodificador 504 y el codificador 501.

El uso de uno o más valores de firma permite la implementación de varios casos de uso como se ilustra en las Figuras 5a a 5d. La Figura 5a ilustra un transcodificador 503, donde ningún procesamiento MIC se lleva a cabo entre el descodificador 504 y el codificador 503. Como consecuencia, los datos protegidos 510 (que comprenden uno o más grupos de datos MIC y metadatos asociados, así como una o más firmas asociadas) no se modifican y la cadena de confianza se mantiene dentro del transcodificador 503. Como resultado, el transcodificador 503 de la Figura 5a se configura para recibir una secuencia de bits de entrada 110 que comprende una secuencia protegida y de confianza de tramas de contenido de entrada 111 y tramas de metadatos de entrada asociados 112 (a las que también se hace referencia como tramas de evolución) y para proveer una secuencia de bits de salida 120 que comprende una secuencia protegida y de confianza de tramas de contenido de salida 121 y tramas de metadatos de salida asociados 122. Ello se asegura protegiendo los datos MIC descodificados, los metadatos regulares y los metadatos adicionales (a los que también se hace referencia como metadatos de evolución) usando una o más firmas. El codificador 501 verifica la única o más firmas y pasa los metadatos adicionales como tramas de metadatos de salida 122 a la secuencia de bits de salida 120. El caso de uso que se muestra en la Figura 5a puede p.ej. ser aplicable a la transcodificación de una secuencia de bits de una primera velocidad binaria a una segunda velocidad binaria.

La Figura 5b muestra un transcodificador conectado a MIC 513 donde la cadena de confianza se rompe por una etapa de procesamiento MIC 505 no confiable. La etapa de procesamiento MIC 505 recibe los datos protegidos 510 y modifica los datos 510. La etapa de procesamiento MIC 505 es "no confiable" en el sentido de que la etapa de procesamiento MIC 505 no es consciente de la clave segura *C* usada por el descodificador 504. Como consecuencia, los datos modificados 511 comprenden uno o más grupos de datos MIC modificados y metadatos asociados, así como una o más firmas inválidas. El codificador 501 se configura para determinar la invalidez de las firmas y se puede configurar para tomar la acción apropiada. En particular, el codificador 501 se puede configurar para suprimir los metadatos adicionales de las tramas de metadatos de entrada 112 y proveer así una secuencia de bits de salida 120 que solo comprende una secuencia de tramas de contenido 121, pero que no comprende las tramas de metadatos asociados 122. Mediante ello, se asegura que el transcodificador 513 no reenvíe metadatos adicionales no confiables. Además, debido al hecho de que la secuencia de bits 120 no comprende tramas de metadatos 122, la secuencia de bits 120 no comprende los valores criptográficos mencionados más arriba (de los campos de protección de las tramas de metadatos 122). Como tal, la secuencia de bits 120 se puede identificar por un descodificador 104 como no confiable.

Como se indica más arriba, el codificador 501 se puede configurar para suprimir los metadatos adicionales de las tramas de metadatos de entrada 112, si el único o más valores de firma no son válidos. Como se describe en el contexto de la Tabla 2, los bloques de metadatos 140 de una trama de metadatos de entrada 112 pueden ser indicativos de descriptores 142 respectivos que describen una o más propiedades de los bloques de metadatos 140 correspondientes. Una de dichas propiedades puede ser el parámetro de procesamiento MIC (al que se hace referencia como el parámetro ajustado_acoplamiento en la Tabla 2). El codificador 501 se puede configurar para usar el parámetro de procesamiento MIC de un bloque de metadatos 142 para decidir si incluir o no los metadatos comprendidos dentro del bloque de metadatos 142 en la secuencia de bits de salida 120. En particular, el parámetro de procesamiento MIC puede indicar al codificador 501 que incluya metadatos de un bloque 140 de la trama de metadatos de entrada 112 en la secuencia de bits de salida 120, aunque las muestras MIC de la trama de contenido asociada 111 se hayan modificado.

La Tabla 5 muestra una semántica a modo de ejemplo del parámetro de procesamiento MIC (a saber, del parámetro ajustado_acoplamiento de la Tabla 2). En el ejemplo ilustrado, un valor "0" del parámetro de procesamiento MIC indica que la carga útil 143 (a saber, los metadatos) de un bloque 140 de metadatos se debe incluir en la secuencia de bits de salida 120 solamente si no hubiera ocurrido ningún procesamiento MIC, p.ej. solamente si el único o más valores de firma se han verificado por el codificador 501. Por otro lado, un valor "3" del parámetro de procesamiento MIC puede indicar que la carga útil 143 del bloque 140 se debe incluir siempre en la secuencia de bits de salida 120, incluso si las muestras MIC se han modificado, p.ej. incluso si el único de más valores de firma no se ha modificado. Además, el parámetro de procesamiento MIC puede asumir valores que indican situaciones intermedias, a saber, el parámetro de procesamiento MIC puede asumir valores que indican que las condiciones de procesamiento MIC que necesitan cumplirse para que la carga útil 143 se incluya en la secuencia de bits de salida 120 o que indican las condiciones de procesamiento MIC, en cuyo caso la carga útil 143 no se incluye en la secuencia de bits de salida 120.

La etapa de procesamiento MIC 505 se puede configurar para informar al codificador 501 del procesamiento que se ha llevado a cabo en las muestras MIC en la etapa de procesamiento MIC 505. En otras palabras, la etapa de procesamiento MIC 505 se puede configurar para informar al codificador 501 sobre las condiciones de procesamiento MIC (p.ej. conversión de la tasa de muestreo de las muestras MIC, inclusión de un sonido de sistema en las muestras MIC, modificación de los metadatos, modificación de una configuración de canal (p.ej. modificación de una señal mono a una señal estéreo, o submezcla de una señal multicanal 5.1 a una señal estéreo), nivelación de la sonoridad, etc.). Como tal, el codificador 501 se puede configurar para recibir indicaciones de las condiciones de procesamiento MIC de la etapa de procesamiento MIC 505. Además, el codificador 501 se puede configurar para

procesar los metadatos de un bloque 140 de metadatos, según las condiciones de procesamiento MIC recibidas y según el valor del parámetro de procesamiento MIC (p.ej. según la semántica de la Tabla 5).

Tabla 5

0	mantener la carga útil solo si no ocurre ningún procesamiento MIC	
1	mantener la carga útil si ocurre uno o más de los siguientes cambios a MIC:	
	•	la tasa de muestreo se ha convertido
2	mantener la carga útil si ocurre uno o más de los siguientes cambios a MIC:	
	•	cualquiera de los cambios mencionados para el caso "1" más arriba
	•	los sonidos de sistema se mezclan en MIC
	•	los metadatos se han modificado
	•	la configuración de canal se ha cambiado
	•	la sonoridad se ha nivelado
3	mantener la carga útil independientemente de si se lleva a cabo el procesamiento MIC	

La Figura 5c ilustra el caso de un transcodificador conectado a MIC 523 que se configura para llevar a cabo el procesamiento MIC confiable. Ello se puede lograr combinando la etapa de procesamiento MIC 506 con una etapa de refirma adicional 507. A tal fin, se puede proveer a una parte confiable una o más de las claves seguras y permitir así que la parte confiable refirme los datos modificados 511. A modo de ejemplo, se puede proveer a la parte confiable la clave segura moderada C_2. Como resultado, los datos modificados 511 se pueden refirmar (a saber, se pueden determinar uno o más valores de firma según los datos modificados 511 usando la clave segura moderada C_2) y proveer así los datos modificados protegidos 512 (que comprenden una secuencia de grupos de datos MIC modificados y metadatos asociados, así como la única o más firmas nuevas). El codificador 501 se puede configurar para verificar la nueva firma y generar la secuencia de bits de salida 120 confiable que comprende la secuencia de tramas de contenido 121 y la secuencia asociada de tramas de metadatos 122. Además, el codificador 501 se puede configurar para determinar que la cadena de confianza se ha roto y que se ha creado una nueva cadena, porque la etapa de refirma 507 puede haber usado una clave segura (p.ej. la clave segura moderada C_2) diferente a la del descodificador 504 (que puede haber usado la clave altamente segura C_1).

La Figura 5d muestra un diagrama de bloques de un transcodificador conectado a MIC 533 con una etapa de procesamiento MIC 509 comprendida dentro del codificador 501. En particular, el transcodificador 533 se configura para mantener una cadena de confianza asegurando que el procesamiento MIC se lleva a cabo por una entidad (p.ej. el codificador 501) que es consciente de la clave segura usada por el descodificador 504 para determinar el único o más valores de firma. El codificador 501 se configura para verificar la única o más firmas de los datos protegidos 510. La etapa de procesamiento MIC interno 508 puede entonces modificar los grupos recibidos de datos MIC y metadatos asociados. Además, el codificador 501 puede comprender una unidad de actualización de metadatos 509 que se configura para actualizar las tramas de metadatos, con sujeción a las modificaciones llevadas a cabo en la etapa de procesamiento MIC 508. En particular, la unidad de actualización de metadatos 509 se puede configurar para determinar valores criptográficos actualizados según las tramas de contenido transcodificadas 121 y las tramas de metadatos 122. Los valores criptográficos actualizados se pueden incluir entonces en las tramas de metadatos 122 para la comunicación al descodificador 104.

La Figura 6 provee otra representación de los transcodificadores 503, 513, 523 y 533, respectivamente.

En el presente documento, se han descrito los métodos y sistemas para transcodificar metadatos. Los métodos y sistemas permiten una transcodificación de metadatos con una complejidad computacional reducida. En particular, se propone proveer descriptores para bloques de metadatos y así permitir a un transcodificador transcodificar los metadatos según los descriptores solamente, sin la necesidad de analizar los metadatos reales comprendidos dentro de un bloque de metadatos. Mediante ello, la complejidad de un transcodificador se puede reducir de manera

significativa. Además, el presente documento provee métodos y sistemas para proteger tramas de metadatos y para proteger datos MIC en un transcodificador conectado a MIC. Como resultado, se puede asegurar que un receptor de metadatos de transcodificador se provee con una indicación de confiabilidad de los metadatos recibidos.

Los métodos y sistemas descritos en el presente documento se pueden implementar como software, firmware y/o hardware. Ciertos componentes pueden p.ej. implementarse como software que se ejecuta en un procesador de señal digital o microprocesador. Otros componentes pueden p.ej. implementarse como hardware y/o como circuitos integrados para aplicaciones específicas. Las señales encontradas en los métodos y sistemas descritos se pueden almacenar en medios como, por ejemplo, memoria de acceso aleatorio o medios de almacenamiento óptico. Se pueden transferir mediante redes como, por ejemplo, redes radioeléctricas, redes de satélite, redes inalámbricas o redes alámbricas, p.ej. Internet. Los dispositivos típicos que usan los métodos y sistemas descritos en el presente documento son dispositivos electrónicos portátiles u otros equipos de consumidor que se usan para almacenar y/o renderizar señales de audio.

REIVINDICACIONES

1. Un codificador (101) configurado para generar una secuencia de bits codificada (110) que comprende una trama de contenido (111) y una trama de metadatos asociados (112); en donde la trama de contenido (111) es indicativa de una señal codificada según un primer sistema de códec; en donde el codificador (101) se configura para

- generar un bloque (140) de metadatos;

- insertar el bloque (140) de metadatos en la trama de metadatos (112);

- seleccionar una clave segura de múltiples claves seguras predeterminadas; en donde las múltiples claves seguras predeterminadas proveen diferentes niveles de confianza;

- generar un valor criptográfico según al menos la trama de contenido (111), en la trama de metadatos asociados (112) y en la clave segura seleccionada; e

- insertar el valor criptográfico generado en la trama de metadatos (112).

2. El codificador (101) de la reivindicación 1, en donde las múltiples claves seguras predeterminadas comprenden

- una clave altamente segura conocida solamente para un desarrollador del codificador (101); y

- una clave segura moderada conocida para un operador del codificador (101).

3. El codificador (101) de cualquiera de la reivindicaciones 1 a 2, en donde el codificador (101) se configura para

- generar múltiples tramas de contenido subsiguientes (111, 311) y tramas de metadatos asociados (112, 312) para la secuencia de bits codificada (110);

- generar un valor criptográfico de trama según una sola trama de contenido (111) y su trama de metadatos asociados (112) y según la clave segura seleccionada; y

- generar un valor criptográfico histórico según al menos algunas de las múltiples tramas de contenido subsiguientes (111, 311) y sus tramas de metadatos asociados (112, 312) y según el valor seguro seleccionado.

4. El codificador (101) de cualquiera de las reivindicaciones 1 a 3, en donde el codificador (101) se configura para calcular un valor HMAC-MD5 o un valor HMAC-SHA256 para generar un valor criptográfico.

5. El codificador (101) de la reivindicación 4, en donde el codificador (101) se configura para truncar el valor HMAC-MD5 o el valor HMAC-SHA256 para producir el valor criptográfico.

6. El codificador (101) de cualquiera de las reivindicaciones 1 a 5, en donde el codificador (101) se configura para insertar una indicación de la clave segura seleccionada en la trama de metadatos (112).

7. Un método para generar una secuencia de bits codificada (110) que comprende una trama de contenido (111) y una trama de metadatos asociados (112); en donde la trama de contenido (111) es indicativa de una señal codificada según un primer sistema de códec; en donde el método comprende

- generar un bloque (140) de metadatos;

- insertar el bloque (140) de metadatos en la trama de metadatos (112);

- seleccionar una clave segura de múltiples claves seguras predeterminadas; en donde las múltiples claves seguras predeterminadas proveen diferentes niveles de confianza;

- generar un valor criptográfico según al menos la trama de contenido (111) en la trama de metadatos asociados (112) y en la clave segura seleccionada; e

- insertar el valor criptográfico generado en la trama de metadatos (112).

8. El método de la reivindicación 7, en donde las múltiples claves seguras predeterminadas comprenden

- una clave altamente segura conocida solamente para un desarrollador del codificador (101); y

- una clave segura moderada conocida para un operador del codificador (101).

9. El método de la reivindicación 7 u 8, que además comprende

- generar múltiples tramas de contenido subsiguientes (111, 311) y tramas de metadatos asociados (112, 312) para la secuencia de bits codificada (110);

- generar un valor criptográfico de trama según una sola trama de contenido (111) y su trama de metadatos asociados (112) y según la clave segura seleccionada; y

- generar un valor criptográfico histórico según al menos algunas de las múltiples tramas de contenido subsiguientes (111, 311) y sus tramas de metadatos asociados (112, 312) y según el valor seguro seleccionado.

10. Un descodificador (104) configurado para recibir una secuencia de bits codificada (110) que comprende una trama de contenido (111) y una trama de metadatos asociados (112); en donde la trama de contenido (111) es indicativa de una señal codificada según un primer sistema de códec; en donde el descodificador (104) se configura para

- extraer un valor criptográfico de la trama de metadatos (112);

- determinar una clave segura a partir de múltiples claves seguras predeterminadas; en donde las múltiples claves seguras predeterminadas proveen diferentes niveles de confianza;

- generar un valor criptográfico de verificación según al menos la trama de contenido (111), la trama de metadatos asociados (112) y la clave segura determinada; y

- comparar el valor criptográfico extraído y el valor criptográfico de verificación para determinar si se puede confiar en la secuencia de bits codificada recibida (110).

11. El descodificador (104) de la reivindicación 10, en donde el descodificador (104) se configura para determinar la clave segura extrayendo la clave segura de la trama de metadatos (112).

12. El descodificador (104) de la reivindicación 10 u 11, en donde el descodificador (104) se configura además para determinar cuál de las múltiples claves seguras predeterminadas se ha usado para generar el valor criptográfico extraído, para determinar un nivel de confianza de la secuencia de bits codificada recibida (110).

13. El descodificador (104) de cualquiera de la reivindicaciones 10 a 12, en donde el descodificador (104) se configura para

- generar múltiples valores criptográficos de verificación para las múltiples claves seguras predeterminadas, respectivamente;

- comparar cada uno de los múltiples valores criptográficos de verificación con el valor criptográfico extraído; y

- determinar que una de las múltiples claves seguras predeterminadas se ha usado para generar el valor criptográfico extraído, si uno de los múltiples valores criptográficos de verificación concuerda con el valor criptográfico extraído.

14. Un método para determinar un nivel de confianza de una secuencia de bits codificada recibida (110) que comprende una trama de contenido (111) y una trama de metadatos asociados (112); en donde la trama de contenido (111) es indicativa de una señal codificada según un primer sistema de códec; en donde el método comprende

- extraer un valor criptográfico de la trama de metadatos (112);

- determinar una clave segura a partir de múltiples claves seguras predeterminadas; en donde las múltiples claves seguras predeterminadas proveen diferentes niveles de confianza;

- generar un valor criptográfico de verificación según al menos la trama de contenido (111), la trama de metadatos asociados (112) y la clave segura seleccionada; y

- comparar el valor criptográfico extraído y el valor criptográfico de verificación para determinar si se puede confiar en la secuencia de bits codificada recibida (110) según el nivel de confianza de la clave segura determinada.

15. Un medio de almacenamiento que comprende un programa de software ejecutable en un procesador, en donde la ejecución del programa de software en el procesador origina llevar a cabo las etapas del método de cualquiera de las reivindicaciones 7 a 9 o 14.

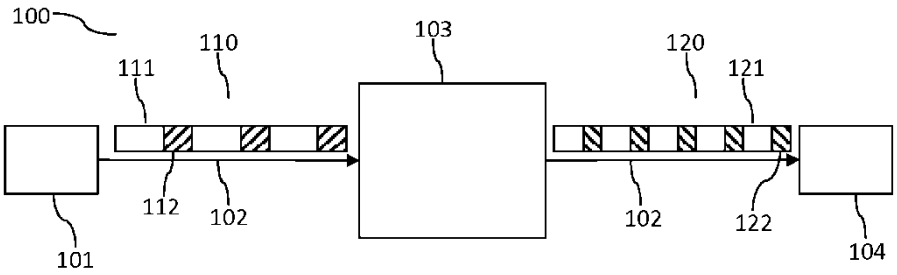

100
110
103
120
111
121
112
102
102
122
101
104

Fig. 1a

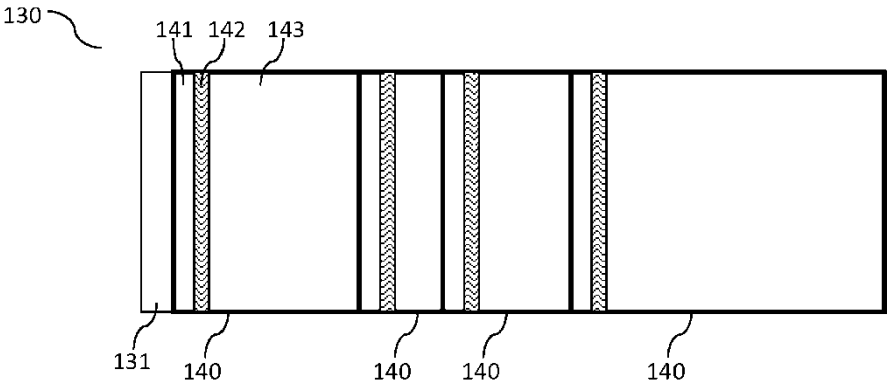

130
141
142
143
131
140
140
140
140

Fig. 1b

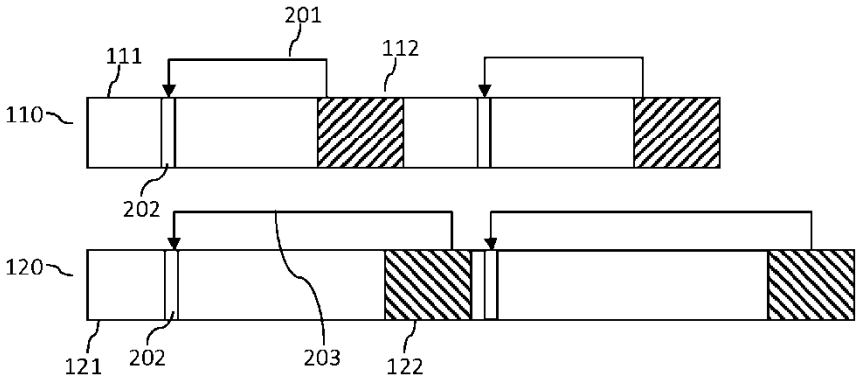

201
111
112
110
202
120
121
202
203
122

Fig. 2a

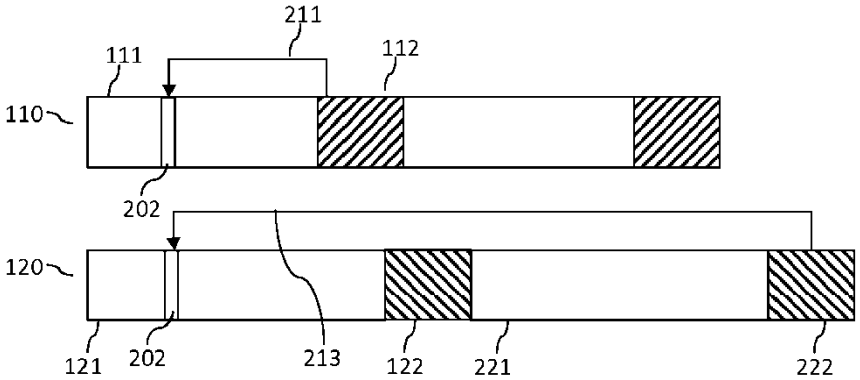

211
111
112
110
202
120
121
202
213
122
221
222

Fig. 2b

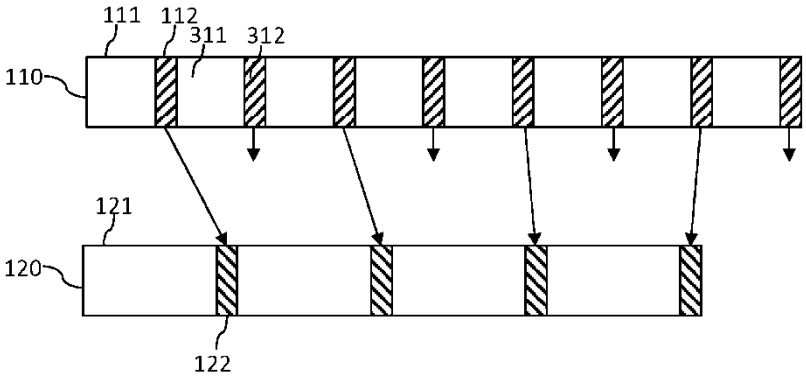


Fig. 3a

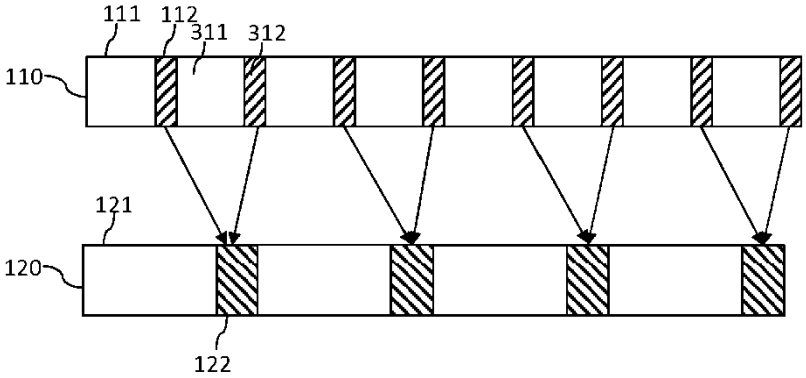


Fig. 3b

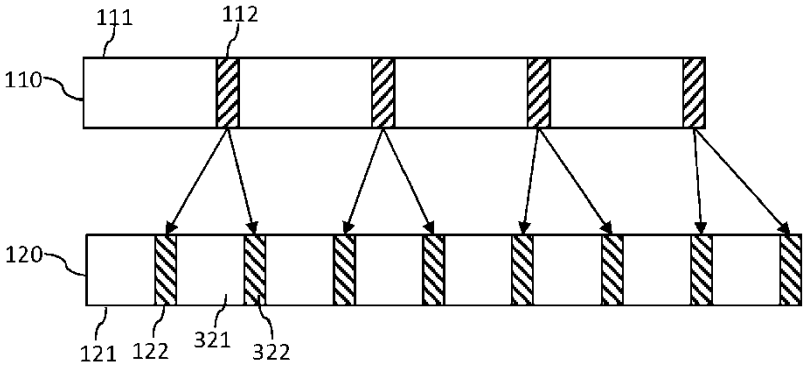


Fig. 4a

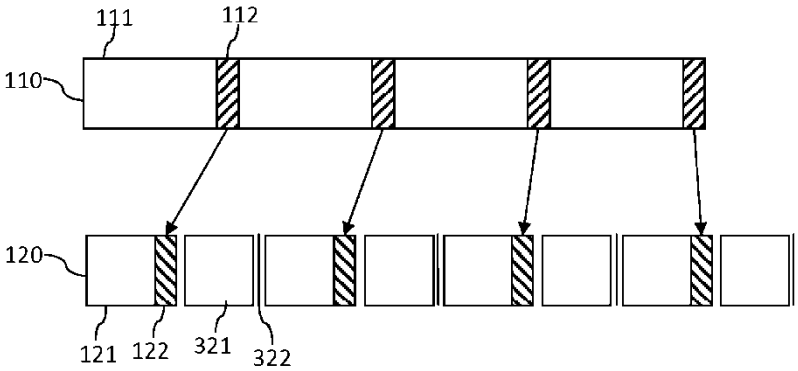


Fig. 4b

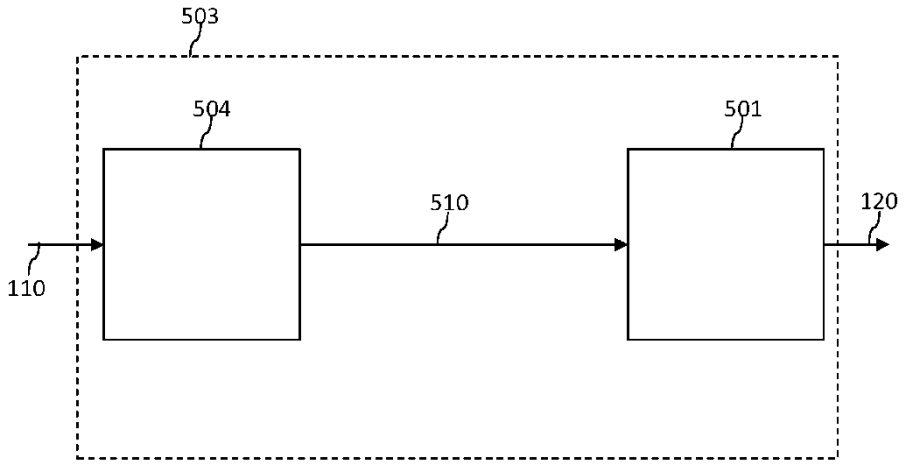


Fig. 5a

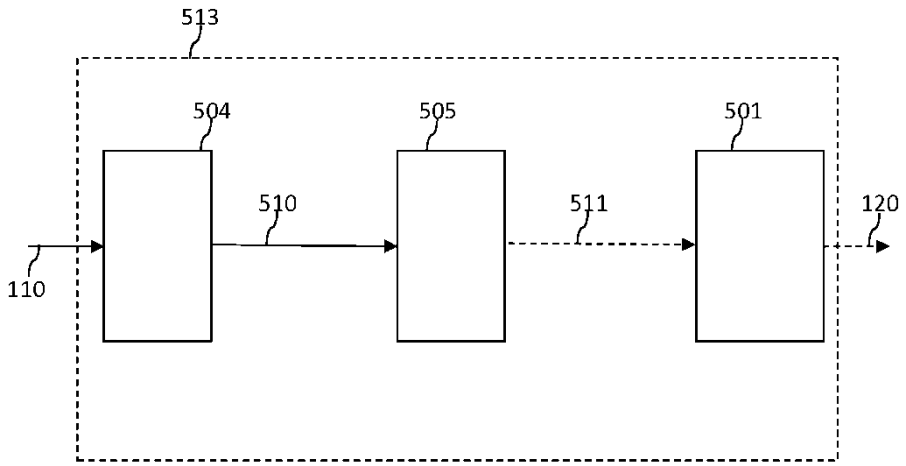


Fig. 5b

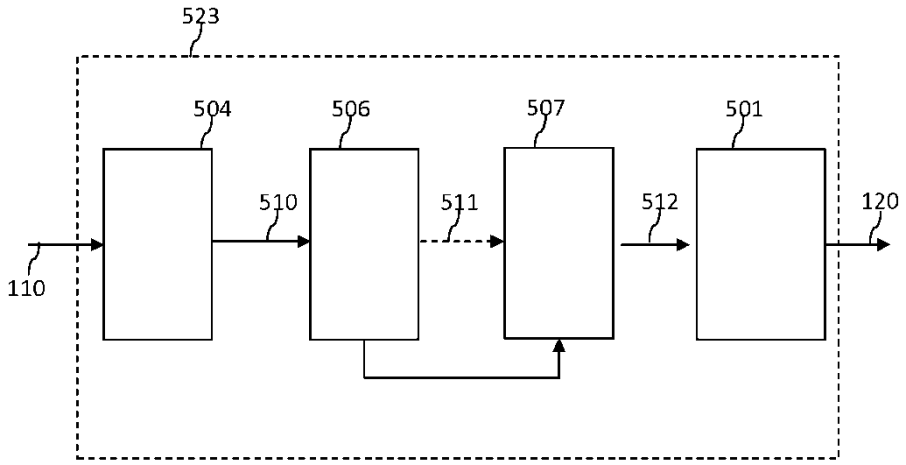

523
504
506
507
501
510
511
512
120
110

Fig. 5c

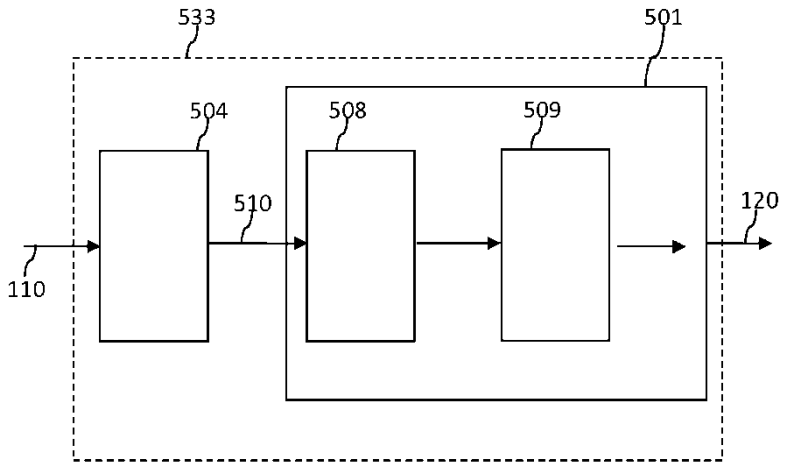
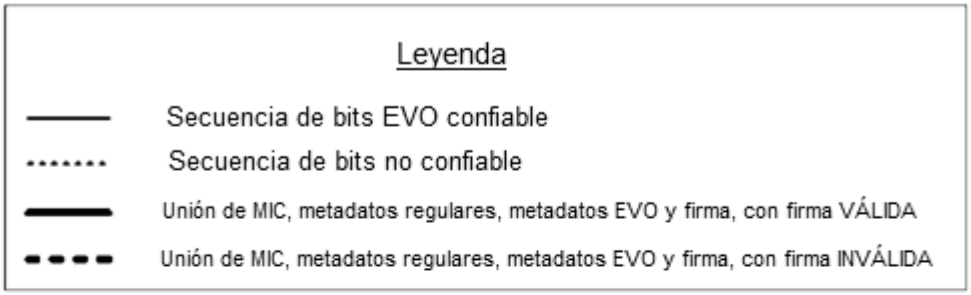
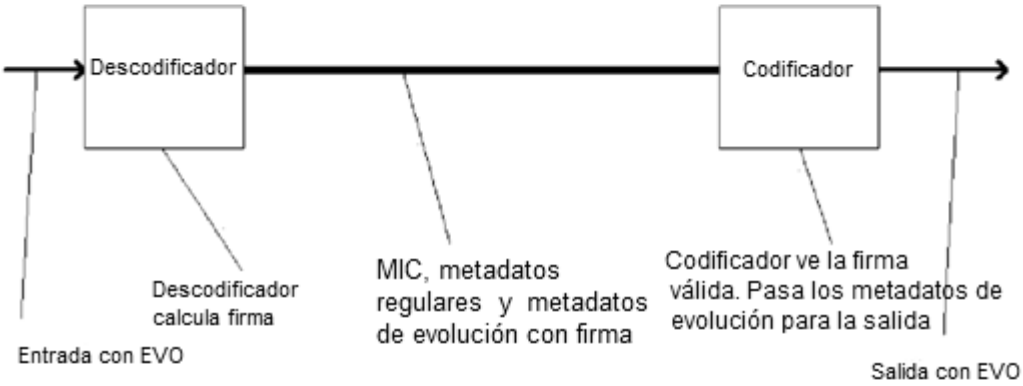

533
501
504
508
509
510
120
110

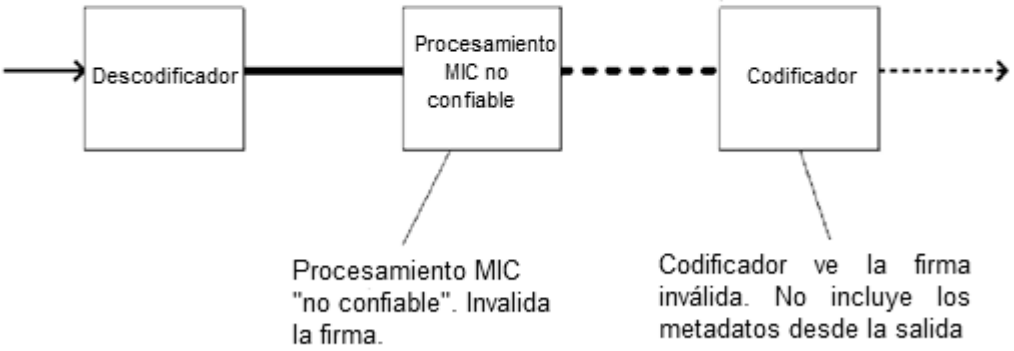
Fig. 5d



Caso de uso #1: sin procesamiento MIC (se mantiene la cadena de confianza)



Caso de uso #2: procesamiento MIC no confiable (se rompe la cadena de confianza)



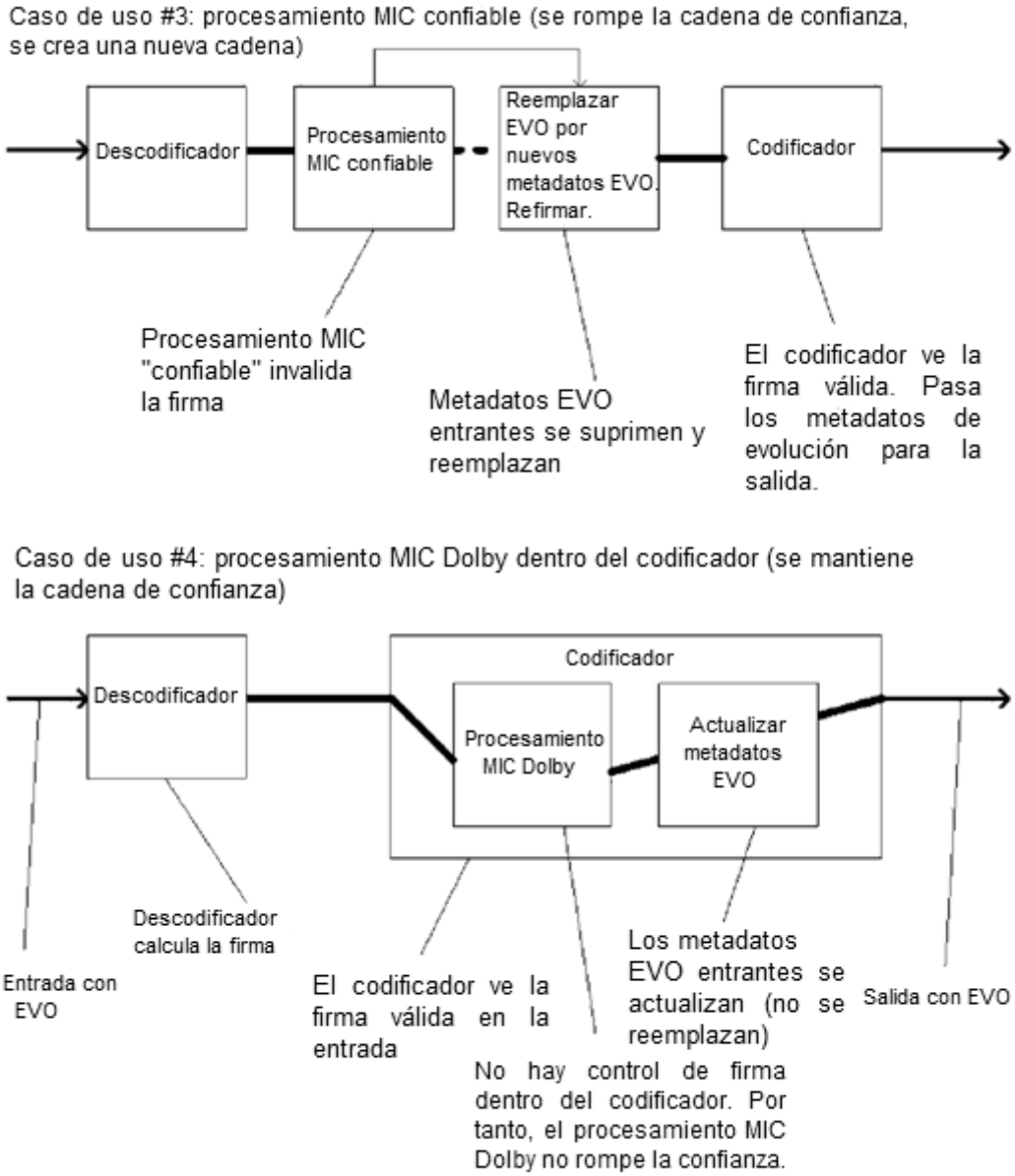

Caso de uso #3: procesamiento MIC confiable (se rompe la cadena de confianza, se crea una nueva cadena)
Descodificador
Procesamiento MIC confiable
Reemplazar EVO por nuevos metadatos EVO. Refirmar.
Codificador
Procesamiento MIC "confiable" invalida la firma
Metadatos EVO entrantes se suprimen y reemplazan
El codificador ve la firma válida. Pasa los metadatos de evolución para la salida.
Caso de uso #4: procesamiento MIC Dolby dentro del codificador (se mantiene la cadena de confianza)
Descodificador
Codificador
Procesamiento MIC Dolby
Actualizar metadatos EVO
Descodificador calcula la firma
Entrada con EVO
El codificador ve la firma válida en la entrada
Los metadatos EVO entrantes se actualizan (no se reemplazan)
Salida con EVO
No hay control de firma dentro del codificador. Por tanto, el procesamiento MIC Dolby no rompe la confianza.

Fig. 6