(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 692 435**

(51) Int. CI.:

G06F 21/60 (2013.01)
G06F 21/62 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **19.05.2008 PCT/AU2008/000702**

(87) Fecha y número de publicación internacional: **27.11.2008 WO08141376**

(96) Fecha de presentación y número de la solicitud europea: **19.05.2008 E 08747971 (3)**

(97) Fecha y número de publicación de la concesión europea: **25.07.2018 EP 2153369**

(54) Título: **Token de seguridad y sistema y procedimiento para la generación y decodificación del token de seguridad**

(30) Prioridad:

18.05.2007 AU 2007902671 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
03.12.2018

(73) Titular/es:

SECURE KEYS PTY LIMITED (100.0%)
ISIS Partners Pty Ltd Unit 49 4 Central Avenue
Thornleigh, NSW 2120, AU

(72) Inventor/es:

FINLAYSON, DAVID, IAN y
STOCKS, MARK, ALEXANDER

(74) Agente/Representante:

PONS ARIÑO, Ángel

DESCRIPCIÓN

Token de seguridad y sistema y procedimiento para la generación y decodificación del token de seguridad

CAMPO DE LA INVENCIÓN

La presente invención se refiere a un token de seguridad y a un sistema y un procedimiento para la generación y decodificación del token de seguridad. El sistema proporciona un uso particular, aunque no exclusivo, en un sistema informático donde se utiliza un sistema de seguridad de 'paradigma de confianza'.

ANTECEDENTES DE LA INVENCIÓN

Los sistemas informáticos se utilizan para almacenar, clasificar y proporcionar diversos tipos de información a una gran variedad de usuarios. La llegada de sofisticadas estructuras de bases de datos, un mayor poder de computación, nuevos algoritmos de búsqueda y un crecimiento exponencial de redes informáticas disponibles públicamente, como Internet, ha traído consigo que muchas empresas proporcionen al menos una parte de sus productos, servicios e información a través de Internet.

La seguridad de la información que alberga un sistema informático es una cuestión fundamental para muchas empresas. Por ejemplo, en el ámbito médico, el acceso a la información del paciente se rige por unas estrictas normas legales según las cuales los usuarios pueden acceder a la información.

Esto ha llevado a la creación de sistemas complejos para el establecimiento de "niveles de permiso" especiales en sistemas informáticos. Es decir, muchos sistemas informáticos incluyen un sistema de acceso múltiple, donde distintos usuarios pueden tener distintos niveles de acceso a distintos tipos de información. La capacidad de acceder a la información puede ser un nivel de permiso atribuido a un dato, o una combinación de ambos.

Los niveles de permiso suelen estar codificados de forma particular. Una de estas formas es la codificación de los niveles de permiso como "vector de bits". Esto es, un array de bits, donde cada bit representa un aspecto diferente del nivel de permiso.
Hwang J J et al. "A NEW ACCESS CONTROL METHOD USING PRIME FACTORISATION", Computer Journal, Oxford University Press, Surrey, GB, Vol. 35, No. 1, 1 February 1992 (1992-02-01), pages 16-20, XP000339251, ISSN: 0010-4620, DOI: 10.1093/COMJNL/35.1.16, describe un procedimiento de protección para el control del acceso a los datos, basado en la factorización de números primos en la teoría numérica. Hwang describe un mecanismo de bloqueo y clave para determinar la capacidad de un usuario de llevar a cabo ciertas acciones en un archivo. El sistema determina un par de clave/bloqueo para cada usuario, de forma que cuando dicho par se ejecuta en su algoritmo de acceso al archivo, el algoritmo tiene como resultado un valor que se compara luego con otro valor asociado al archivo y a la operación solicitada.

Chang C K et al. "A BINARY SINGLE-KEY-LOCK SYSTEM FOR ACCESS CONTROL", IEEE Transactions on Computers, IEEE Service Center, Los Alamitos, CA, US, Vol. 38, No. 10, 1 October 1989 (1989-10-01), pages 1462-1466, XP000070485, ISSN: 0018-9340, DOI: 10.1109/12.35842, describe un procedimiento de codificación binaria aplicado a un sistema de bloqueo de clave única para lograr el control del acceso asociando con cada accessor una única clave y con cada recurso un único bloqueo. Se utiliza una representación binaria para implementar el sistema de bloqueo de clave única. La representación binaria no solo implementa el sistema de protección, sino que también implica relaciones jerárquicas entre los accessors, importantes para controlar los cambios de derechos de acceso.

El documento "Linux Permissions", XP003025645, describe una "guía de ayuda" para usuarios para moverse por los permisos Linux y explica cómo funcionan los permisos en Linux. Cada dato por separado se define como una estructura de 9 bits. Esta estructura de bits se define como basada en pruebas, binaria o de formato óptimo. La estructura de 9 bits define los permisos en base al valor de cada bit. Cada bit o conjunto de bits define quién puede acceder y qué tipo de acceso le está permitido.

El documento "Learning the shell", XP003025646, describe cómo se configuran los permisos para Unix. Una vez más, los permisos se configuran como series de bits que representan ajustes de permisos.

Ninguno de los documentos anteriores describe un sistema o un procedimiento para la codificación de una etiqueta de seguridad que identifica niveles de acceso a datos en un sistema informático.

RESUMEN DE LA INVENCIÓN

La invención se define mediante las reivindicaciones independientes del sistema 1, 9, 10 y 11 y la reivindicación independiente del procedimiento 12. En un primer aspecto, la presente invención proporciona un - sistema para codificar una etiqueta de seguridad para un dato utilizable en un sistema informático (100), dicha etiqueta de seguridad contiene características del dato, y el sistema comprende:

medios para la asignación (102, 106, 104, 108, 116, 118, 120, 122) de un único número entero a cada característica de un conjunto de características, dicho conjunto comprende diversas características; y

medios para la combinación (102, 106, 104, 108, 116, 118, 120, 120), donde dicho dato posee más de una característica, los números enteros asignados a cada característica del conjunto de características descritas en la etiqueta de seguridad informativa, en una etiqueta de seguridad informativa codificada que comprende un único número entero, para confundir cada conjunto de características codificado en la etiqueta de seguridad.

El número entero asignado a cada conjunto de características puede determinarse exponencialmente como x^y, donde x e y son números enteros. Los valores enteros pueden combinarse sumando los números enteros.

En una forma de realización alternativa, el número entero asignado a cada conjunto de características es un número primo único. En esta forma de realización, los valores enteros se combinan multiplicando los números enteros.

En una forma de realización, al asignar un número entero a cada conjunto de características que puede describir la etiqueta, el sistema se configura para incrementar el exponencial y para cada conjunto de características.

El conjunto de características puede ser de nivel de permiso o un compartimento. La combinación conocida como etiqueta de seguridad.

En un segundo aspecto, la presente invención proporciona un sistema para permitir a un usuario acceder a un dato contenido en un sistema informático (100), comprendiendo los medios (102, 104, 106, 108, 116, 118, 120, 122) para asignar una etiqueta de seguridad de acuerdo con el primer aspecto de la invención, para todos los datos específicos del sistema informático (100), los medios (102, 104, 106, 108, 116, 118, 120, 122) para asignar una etiqueta de seguridad a todos los usuarios del sistema informático (100), y los medios (102, 104, 106, 108, 116, 118, 120, 122) para asignar una etiqueta de seguridad a todas las aplicaciones de software que residen en un sistema informático (100), donde, para acceder a un dato específico, la etiqueta de seguridad del usuario y la aplicación de software deben dominar la etiqueta de seguridad asignada al dato específico.

En un tercer aspecto, la presente invención proporciona un sistema para construir una lista que contenga información sobre si cada uno de los diversos usuarios pueden acceder a cada uno de los diversos datos contenidos en un sistema informático (100), comprendiendo, medios de determinación (102, 104, 106, 108, 116, 118, 120, 122) para determinar un token de seguridad para cada uno de los usuarios y un token de seguridad para cada uno de los datos de acuerdo con el primer aspecto de la invención, medios de determinación (102, 104, 106, 108, 116, 118, 120, 122) para determinar si cada uno de los distintos usuarios puede acceder a cada uno de los diversos datos, y medios de construcción (102,104, 106, 108, 116, 118, 120, 122) para construir una lista de los tokens de seguridad y los permisos de acceso. En un cuarto aspecto, la presente invención proporciona un sistema para la decodificación de una etiqueta de seguridad de la forma descrita en el primer aspecto de la invención, comprendiendo los medios de determinación (102, 104, 106, 108, 116, 118, 120, 122) para determinar el suelo de la base del logaritmo del único número entero con el fin de determinar un valor para la primera del conjunto de características, los medios de sustracción (102, 104, 106, 108, 116, 118, 120, 122) para restar el suelo de la base del logaritmo del único número entero del número entero con el fin de producir un número entero de resto, y los medios (102, 104, 106, 108, 116, 118, 120, 122) para determinar si el valor del resto es distinto de cero, lo cual invoca los medios de determinación (102, 104, 106, 108, 116, 118, 120, 122) para determinar el suelo del logaritmo del número entero de resto para determinar otra característica del conjunto de características, donde el proceso se repite hasta que el valor del resto sea cero. De acuerdo con un quinto aspecto, la presente invención proporciona proporciona un procedimiento para la codificación de una etiqueta de seguridad para un dato adecuado en un sistema informático (100) para proteger los datos a los que acceden uno o varios usuarios, la etiqueta de seguridad contiene características del dato, el procedimiento comprende:
asignar un único número entero a cada característica de un conjunto de características, el conjunto comprende una diversidad de características; y
donde un dato posee más de una característica, combinando los números enteros asignados a cada característica del conjunto de características descritas en la etiqueta de seguridad informativa, en una etiqueta de seguridad codificada que comprende un único número entero, para confundir cada conjunto de características codificado en la etiqueta de seguridad.

En un sexto aspecto, la presente invención proporciona una aplicación de software que incluye instrucciones para controlar un sistema informático, para realizar un procedimiento de acuerdo con un quinto aspecto de la invención.

En un séptimo aspecto, la presente invención proporciona un soporte grabable que incorpora una aplicación de software de acuerdo con un sexto aspecto de la invención.

DESCRIPCIÓN DETALLADA DE LOS DIBUJOS

Se describirán las características de la presente invención en una forma de realización de la misma, a título de ejemplo, con referencia a los dibujos adjuntos, en los cuales:

La Figura 1 representa un sistema informático adecuado para ejecutar una aplicación de software de acuerdo con una forma de realización de la presente invención;
Las Figuras 2 y 5 son diagramas de flujo que describen un proceso alternativo para codificar un conjunto de etiquetas para un sistema de seguridad de acuerdo con realizaciones de la presente invención;
Las Figuras 3 y 6 son diagramas de flujo que describen procesos alternativos para codificar un token en particular (etiqueta de seguridad) para un sistema de seguridad de acuerdo con realizaciones de la presente invención;
Las Figuras 4 y 7 son diagramas de flujo que describen procesos alternativos para decodificar un token en particular (etiqueta de seguridad) para un sistema de seguridad de acuerdo con realizaciones de la presente invención;
La Figura 8 es un diagrama de flujo que describe el proceso para determinar una relación de dominancia binaria para un sistema de seguridad de acuerdo con una forma de realización de la presente invención;
La Figura 9 es un diagrama de flujo que describe el proceso para determinar una relación de dominancia triádica para un sistema de seguridad de acuerdo con una forma de realización de la presente invención;

DESCRIPCIÓN DE UNA FORMA DE REALIZACIÓN

En la Figura 1 se muestra un diagrama esquemático de un sistema informático 100 adecuado para su uso con una forma de realización del a presente invención. El sistema informático 100 puede emplearse para ejecutar aplicaciones y/o servicios de sistema tales como una estructura de torneos de acuerdo con una forma de realización de la presente invención. El sistema informático 100 preferentemente comprende un procesador 102, memoria de solo lectura (ROM) 104, memoria de acceso aleatorio (RAM) 106, y dispositivos de entrada/salida tales como unidades de disco 108, periféricos de entrada tales como un teclado 110 y una pantalla (u otro dispositivo de salida) 112. El ordenador incluye aplicaciones de software que pueden almacenarse en RAM 106, ROM 104, o unidades de disco 108 y pueden ser ejecutados por el procesador 102.

Un enlace de comunicación 114 se conecta con una red de ordenadores como Internet. Sin embargo, el enlace de comunicación 114 puede conectarse a una línea telefónica, una antena, una puerta de enlace u otro tipo de enlace de comunicaciones. Las unidades de disco 108 pueden incluir cualquier soporte de almacenamiento adecuado, como, por ejemplo, unidades de disquete, unidades de disco duro, unidades de CD ROM o unidades de cinta magnética de almacenamiento de datos. El sistema informático 100 puede utilizar una única unidad de disco 108 o varias. El sistema informático 100 puede utilizar cualquier sistema operativo adecuado 116, como Microsoft Windows™ o un sistema operativo basado en Unix™.

El sistema comprende además una aplicación de software 118, que en la presente forma de realización incluye una base de datos. La aplicación de software 118 puede conectarse mediante interfaz con otras aplicaciones de software 120 o con un ordenador remoto (no mostrado) a través de un enlace de comunicaciones 114.

La aplicación de software 118 representa un ejemplo de la presente invención, y se utiliza para controlar el acceso a la base de datos, empleando, entre otros dispositivos, un paradigma de sistema de confianza. Un paradigma de sistema de confianza es aquel donde cada token específico (etiquetas de seguridad) se encuentra definido, siendo cada token representativo de un conjunto de características únicas de la información a la que va asociado. El acceso al sistema está restringido por una serie de niveles (y compartimentos como tokens). Un usuario debe tener un nivel mayor que el nivel de información al que está intentando acceder. La noción de que un usuario precisa un nivel mayor que el nivel de información al que está accediendo se denomina, a veces, control de acceso obligatorio. Por ejemplo, un token puede indicar un 'nivel de seguridad' o un 'nivel de permiso' de un archivo en particular guardado en la base de datos. El token puede también utilizarse para indicar otras características de los datos, como un 'compartimento' en particular dentro de un nivel de seguridad.

Por ejemplo, empleando nomenclatura matemática, un token puede estar representado por la siguiente expresión

semántica:

Token = security_level {conjunto de compartimentos de seguridad}

Esta expresión puede comprenderse más fácilmente con referencia al Ejemplo 1. En el Ejemplo 1, los tipos de datos y tokens están personalizados para poder aplicarse a una base de datos de seguridad para guardar información médica/biológica.

'Altamente sensible' {'id personal', 'resultados del análisis de datos'}
'Sensible' {'resultados de ADN', 'resultados de colposcopia'}
'Sensible' {'gestión de seguridad'}
'Restringida' {'estadísticas de VPH'}
'Pública' {'factores de riesgo del VPH'}

Ejemplo 1 - Estructura de token de muestra 1

En otras palabras, una serie de tokens puede definirse, a nivel conceptual, como un número de niveles de permiso, estando asociado cada nivel de permiso con uno o más compartimentos (esto es, tipos de información que se encuentran dentro del nivel de permiso). Por ejemplo, la información sobre identificación personal y los resultados de análisis de datos se consideran información altamente sensible, mientras que la información general sobre los factores de riesgo del VPH se consideraría información pública.

En la técnica anterior, un procedimiento de codificación de los niveles de permiso/seguridad y compartimentos en un formato que sea compatible con sistemas informáticos ha sido utilizar un vector de bits, donde cada bit (o series de bits) representa las categorías de nivel y compartimento. El vector de bits que incluye la información codificada es denominado, en general, "token".

Las realizaciones de la presente invención, por el contrario, codifican toda la semántica del token en un único número entero.

Una forma de realización del algoritmo utiliza una forma exponencial de x^y para crear una separación entre la semántica de la etiqueta, donde x es cualquier base de número entero común e y es cualquier exponente de un número.

Así, en referencia al Ejemplo 1, los niveles y compartimentos pueden codificarse según se muestra a continuación, en el Ejemplo 2.

'identificadores personales'	$= 2^1 = 2$
'resultados del análisis de datos'	$= 2^2 = 4$
'resultados de ADN'	$= 2^3 = 8$
'resultados de colposcopia'	$= 2^4 = 16$
'gestión de seguridad'	$= 2^5 = 32$
'estadísticas sobre VPH'	$= 2^6 = 64$
'factores de riesgo del VPH'	$= 2^7 = 128$
'Altamente sensible'	$= 2^{11} = 2048$
'Sensible'	$= 2^{10} = 1024$
'Restringida'	$= 2^9 = 512$
'Pública'	$= 2^8 = 256$

Ejemplo 2 - Valores codificados para cada nivel/compartimento

Los pasos del procedimiento de asignación de valores a cada etiqueta en un conjunto de etiquetas se muestran en la figura 2. En los pasos 200 y 202 se selecciona la lista de compartimentos y se elige un exponente inicial de 1 respectivamente. En otras palabras, el valor para el primer compartimento pasa a ser $2^1 = 2$. En 204, se comprueba la lista de compartimentos para determinar si hay algún compartimento. En caso afirmativo, se asigna el primer token al primer compartimento (206).

A partir de ahí, el exponente se incrementa y se selecciona el siguiente compartimento de la lista (208). Posteriormente,

el procedimiento vuelve al paso 204, para determinar si la lista está vacía. Este proceso se repite las veces necesarias hasta que la lista de compartimentos quede vacía (210).

Una vez la lista de compartimentos está vacía, se utiliza una serie similar de pasos del procedimiento para asignar valores de token a cada uno de los niveles. Primero, se selecciona el primer nivel de la lista de niveles (212).

En 214, se comprueba la lista de niveles para determinar si hay más niveles. En caso afirmativo, se asigna el token al primer nivel (216).

A partir de ahí, el exponente se incrementa y se selecciona el siguiente nivel de la lista (218). Posteriormente, el procedimiento vuelve al paso 214, para determinar si la lista está vacía. Este proceso se repite las veces necesarias hasta que la lista de niveles quede vacía (220).

Una vez asignado un valor en forma exponencial x^y a cada nivel/compartimento, puede codificarse un token específico que se asignará a un archivo/usuario específico añadiendo los valores de nivel/compartimento resultantes.

Los pasos del procedimiento para este proceso se describen en la figura 3, donde se ha codificado una etiqueta particular. En primer lugar, en 300, se asigna al token el valor para el nivel. En segundo lugar, se accede a la lista de compartimentos para determinar si está vacía (302). Si no lo está, se añade el valor de la etiqueta al valor del token (304). Este proceso se repite hasta que todos los valores de compartimento se han añadido al token. Una vez añadidos todos los valores (esto es, cuando la lista de compartimentos esté vacía), el proceso finalizará (306).

El proceso también puede describirse por referencia al siguiente ejemplo. A los datos altamente sensibles relacionados con identificadores personales y resultados de análisis de datos se les asigna la etiqueta 'Altamente sensible' {'identificadores personales', 'resultados de análisis de datos'} y se codifican como:

$$2^{11} + 2^1 + 2^2 = 2048 + 2 + 4 = 2054$$

En otro ejemplo, a los datos de la etiqueta que se encuentran a disposición pública y están relacionados con los factores de riesgo del VPH se les puede asignar la etiqueta 'Pública' {'factores de riesgo del VPH'}, que se codifica como:

$$2^8 + 2^7 = 256 + 128 = 384$$

La distancia numérica entre cada valor de x^y garantiza que cada combinación posible de valores añadidos sea representada por un único número entero.

Si las etiquetas iniciales están encriptadas, el proceso de codificación sigue siendo válido y los algoritmos aún pueden utilizarse.

Por ejemplo, en referencia al Ejemplo 2, la etiqueta 'identificadores personales', puede transformarse en primer lugar en una cadena encriptada empleando un algoritmo de encriptación unidireccional, antes de ser codificada como un número entero.

'identificadores personales' -> '$#&RERs*Er%$m<ydfg'
que se codifica a continuación como un número entero:
'$#&RERs*Er%$m<ydfg' = $2^1 = 2$

De este modo, las etiquetas pueden confundirse aún más.

La decodificación del número entero para llegar a la estructura semántica apropiada (esto es, combinación de niveles y compartimentos) puede alcanzarse identificando de forma repetitiva el exponente más significativo asociado al número entero (ignorando el resto) para llegar al primer nivel/compartimento, quitando posteriormente el componente más significativo del total del entero para llegar a un "resto" y repitiendo luego el proceso anterior, hasta que no quede resto alguno.

La metodología de decodificación se muestra en la figura 4. En el paso 400, el proceso recibe el número entero del token. Se toma el logaritmo del número entero (402) y se toma el suelo del logaritmo del número entero (404). Este valor se convierte en el primer número entero de la lista (que representa le nivel del token) (406). Posteriormente, se sustrae el suelo del logaritmo del número entero del número entero del token original, para producir un token de

diferencia (408). Si el token de diferencia es distinto de cero (se sabe que los compartimentos están codificados en el token de diferencia) (410), y a continuación se toma el token de diferencia del logaritmo (412) y se toma el suelo del logaritmo del token de diferencia (414). Este valor pasa a ser el siguiente número entero de la lista (que representa el primero de los compartimentos en una lista de compartimentos) (416) (esto es, el valor del segundo token). Posteriormente, se sustrae el suelo del logaritmo del número entero del número entero del token de diferencia, para producir un segundo token de diferencia (418). Si el segundo token de diferencia también es distinto de cero (410), entonces los pasos del procedimiento se repiten hasta que el token de diferencia es cero (420), punto en el que todos los tokens (compartimentos) se han decodificados y finaliza el proceso.

En otro ejemplo, para decodificar el número entero 2054 se necesitarían los siguientes pasos:

1. $\log_2(2054) = 11.004220466318195$;
2. floor(log) = 11;
3. Así, el elemento más significativo es 2^{11}, que corresponde al nivel 'altamente sensible';
4. Sustraer 2^{11} de 2054 = 2054 - 2048 = 6;
5. $\log_2(6) = 2.5849625007211561$;
6. floor(log) = 2;
7. Así, el elemento más significativo es 2^2, que corresponde al compartimento 'resultados del análisis de datos';
8. Sustraer 2^2 de 6 = 6 - 4 = 2;
9. $\log_2(2) = 1$;
10. floor(log) = 1 = 2^1;
11. Así, el elemento más significativo es 2^1, que corresponde al compartimento 'identificadores personales';
12. Al no haber resto, se han identificado todos los niveles/compartimentos y se ha reconstruido el original.

En el anterior ejemplo se puede ver que le token del número entero 2054 es
'Altamente sensible' {'id personal', 'resultados del análisis de datos'}

Ejemplo 3 - Estructura de token de muestra 2

Otra representación del algoritmo utiliza números primos para crear una separación entre la semántica de etiquetas.

Así, en referencia nuevamente al Ejemplo 1, los niveles y compartimentos pueden codificarse según se muestra a continuación, en el Ejemplo 3.

'identificadores personales'	= 1
'resultados del análisis de datos'	= 3
'resultados de ADN'	= 7
'resultados de colposcopia'	= 11
'gestión de seguridad'	= 13
'estadísticas sobre VPH'	= 17
'factores de riesgo del VPH'	= 19
'Altamente sensible'	= 23
'Sensible'	= 29
'Restringida'	= 31
'Pública'	= 37

Ejemplo 4 - Valores codificados para cada nivel/compartimento utilizando números primos

Los pasos del procedimiento de asignación de valores a cada etiqueta en un conjunto de etiquetas se muestran en la figura 5. En el paso 500, se selecciona la lista de compartimentos y se realiza una determinación con respecto a si la lisa está vacía (502). De lo contrario, se asigna un único número primo a la etiqueta de compartimento. Posteriormente, el procedimiento vuelve al paso 502, para determinar si la lista está vacía. Este proceso se repite las veces necesarias hasta que la lista de compartimentos quede vacía.

Una vez la lista de etiquetas está vacía, se utiliza una serie similar de pasos del procedimiento para asignar valores de token a cada uno de los niveles. Si la lista está vacía, todos los compartimentos están asignados y se selecciona el primer nivel de la lista de etiquetas (506). Se realiza una determinación con respecto a si la lista está vacía (508).

De lo contrario, se incrementa el número primo y se asigna a la siguiente etiqueta (510). Posteriormente, el

procedimiento vuelve al paso 508, para determinar si la lista está vacía. Este proceso se repite las veces necesarias hasta que la lista de compartimentos quede vacía.

Cuando la lista de etiquetas está vacía, los niveles se agrupan por conjuntos relevantes (512). Por ejemplo, 'altamente sensible' y 'sensible' y 'restringida' y 'pública' se codifican como un conjunto, esto es, está implícito que si se necesita un nivel de 'altamente sensible', las demás etiquetas del conjunto de etiquetas también estarán codificadas.

Una vez asignado un valor de número primo a cada nivel/compartimento, puede codificarse un token específico que se asignará a un archivo/usuario específico añadiendo los valores de nivel/compartimento resultantes.

Los pasos del procedimiento para este proceso se describen en la figura 6, donde se ha codificado una etiqueta particular. En primer lugar, en 600, se configura el valor para el primer compartimento de la etiqueta de seguridad y se configura el total de la etiqueta en cero (602). En segundo lugar, se accede a la lista de compartimentos para determinar si hay más compartimentos (604). En caso afirmativo, el total de la etiqueta pasa a ser el total de la etiqueta multiplicado por el valor de número primo asignado a la etiqueta del compartimento (606). Este proceso se repite hasta que todos los valores de compartimento se han añadido al token. Cuando ya no haya más compartimentos, se configurará el valor del primer nivel de la etiqueta de seguridad (608), y se realizará una determinación con respecto a si hay más niveles en el conjunto de etiquetas (610). En caso afirmativo, el total de la etiqueta pasa a ser el total de la etiqueta multiplicado por el valor de número primo asignado a la etiqueta del nivel (612). Cuando ya no haya más niveles en el conjunto L, el proceso habrá terminado (614).

El proceso también puede describirse por referencia al siguiente ejemplo. A los datos altamente sensibles relacionados con identificadores personales y resultados de análisis de datos se les asigna la etiqueta 'Altamente sensible' {'identificadores personales', 'resultados de análisis de datos'} y se codifican como:

$$37 \times 31 \times 29 \times 23 \times 3 \times 5 = 11475735$$

Las etiquetas 'altamente sensible', 'sensible', 'restringida' y 'pública' se codifican como un conjunto. Esto es, si se precisa el nivel de 'altamente sensible', por implicación, todos los demás niveles estarán incluidos en el conjunto de etiquetas. En otro ejemplo, a los datos de la etiqueta que se encuentran a disposición pública y están relacionados con los factores de riesgo del VPH se les puede asignar la etiqueta 'Pública' {'factores de riesgo del VPH'}, que se codifica como:

$$23 \times 19 = 437$$

La naturaleza atómica de los números primos garantiza que todas las combinaciones estén representadas por un único número entero.

Si las etiquetas iniciales están encriptadas, el proceso de codificación sigue siendo válido y los algoritmos aún pueden utilizarse.

Por ejemplo, en referencia al Ejemplo 3, la etiqueta 'identificadores personales', puede transformarse en primer lugar en una cadena encriptada empleando un algoritmo de encriptación unidireccional, antes de ser codificada como un número entero.
'identificadores personales' -> '$#&RERs*Er%$m<ydfg'
que se codifica a continuación como un número entero:
'$#&RERs*Er%$m<ydfg' = 1

De este modo, las etiquetas pueden confundirse aún más.

La decodificación del número entero para llegar a la estructura semántica adecuada (esto es, combinación de niveles y compartimentos) puede alcanzarse identificando de forma reiterada cada etiqueta empleando una función de módulo.

Por ejemplo, la decodificación de 11475735 se produciría utilizando la siguiente metodología:

módulo(11475735, 3) identificadores personales - resto 0; es un token
módulo(11475735, 5) resultados del análisis de datos - resto 0; es un token
módulo(11475735, 7) resultados de ADN - resto 5; no es un token
módulo(11475735, 11) resultados de colposcopia - resto 7; no es un token

módulo(11475735, 13) gestión de seguridad - resto 11; no es un token
módulo(11475735, 17) estadística VPH - resto 4; no es un token
módulo(11475735, 19) factores de riesgo del VPH - resto 1; no es un token
módulo(11475735, 23) Pública - resto 0; es un token
módulo(11475735, 29) Restringida - resto 0; es un token
módulo(11475735, 31) Sensible - resto 0; es un token
módulo(11475735, 37) Altamente sensible - resto 0; es un token

Así, el token original se reconstruye: 'Altamente sensible' {'identificadores personales', 'resultados del análisis de datos'}.

La metodología de decodificación se muestra en la figura 7. En el paso 700, el proceso recibe el número entero del token. Se examina el primer compartimento de la lista (702) y se realiza una determinación con respecto a si la lisa está vacía (704). Si la lista está vacía, la etiqueta pasa a ser el primer nivel de la lista (706). Si la lista no está vacía, se toma el módulo del token codificado (708). Si el resultado de la función de módulo es distinto de cero, el proceso vuelve al paso correspondiente para determinar si la lista está vacía (704), de lo contrario, la etiqueta de compartimento actual es válida y se añade a la lista de etiquetas de compartimento (710) y el proceso vuelve al paso correspondiente para determinar si la lista está vacía. Volviendo al caso donde la lista de etiquetas de compartimento esté vacía y la etiqueta de nivel pase a ser el siguiente nivel de la lista (706), se realiza una determinación con respecto a si la lista está vacía (712). Si la lista está vacía, no hay más etiquetas y el proceso habrá terminado (714). Si la lista no está vacía, se toma el módulo del token codificado (716). Si el resultado de la función de módulo es distinto de cero, el proceso vuelve al paso correspondiente para determinar si la lista está vacía (712), de lo contrario, la etiqueta actual es válida y se añade a la lista de etiquetas (710) y el proceso vuelve al paso correspondiente para determinar si la lista está vacía.

Estos procedimientos de creación, codificación y decodificación de tokens de seguridad proporcionan diversas ventajas en relación a la codificación tradicional de 'vectores de bits'.

En primer lugar, mejora la facilidad de transmisión, puesto que solo es necesario transmitir un único número entero. Del mismo modo, las necesidades de almacenamiento también se reducen (o, en otras palabras, se maximiza la compresión de la información), puesto que solo se necesita almacenar un único número entero para describir cada token.

La manipulación de los tokens también se simplifica de forma ideal, puesto que se utiliza una metodología de codificación y decodificación sencilla.

Además, la mecánica de codificación y decodificación de tokens se desacopla de la semántica del token. Como el token se representa como un número entero único, la semántica del token se confunde ante los ojos del observador casual. El observador casual no puede derivar información alguna del token per se, porque se precisan conocimientos sobre metodología de codificación, la base utilizada y el modo en el que los números enteros se mapean sobre la estructura del token.

Por último, el procedimiento de vectores de bits de la técnica anterior es inherentemente limitador, puesto que el número de niveles y compartimentos disponibles debe fijarse antes de implementar un sistema de seguridad. La forma de realización de la presente invención puede modificarse in situ para añadir niveles y compartimentos adicionales, sin necesidad de cambiar las estructuras fundamentales del sistema de seguridad. Así pues, la forma de realización aquí descrita ofrece más flexibilidad cuando se codifica la semántica.

Se entenderá que si bien los anteriores ejemplos utilizan bien una forma exponencial de base 2 o una base de número primo por conveniencia, puede utilizarse cualquier base.

El sistema de codificación arriba descrito también puede utilizarse en combinación con una relación de dominancia para mejorar el flujo de seguridad en una base de datos. Una relación de dominancia es aquella donde la capacidad de un usuario de acceder a la información viene determinada por el hecho de si tiene "dominancia" con respecto a la información. Dicho de otro modo, el usuario debe tener un nivel de permiso igual o superior al nivel necesario para acceder a la información.

Un token de usuario debe dominar el token de información para que el usuario obtenga acceso a la información. El token de usuario debe ser de nivel superior al del token de información y los compartimentos deben ser un superconjunto de los compartimentos del token de información.

En el contexto del sistema de etiquetado arriba descrito, la relación es binaria y el acceso está garantizado si la etiqueta del usuario domina la etiqueta asociada con la información solicitada.

Por ejemplo, si el usuario Bob tiene una etiqueta de seguridad que incluye las categorías 'Altamente sensible' {'identificadores personales', 'resultados del análisis de datos', 'factores de riesgo de VPH}, entonces Bob dominará un documento de Microsoft Word™ que tenga la etiqueta 'Pública' {'factores de riesgo del VPH'}.

El anterior es un ejemplo de relación de dominancia binaria.

La forma de realización aquí descrita utiliza una relación de sistema de dominancia triádica.

En dicha relación, hay tres tipos de tokens. Hay un token de usuario (asociado con un usuario), un token de información (asociado con un fragmento de información concreto, como una entrada de la base de datos o un archivo) y un token de aplicación (asociado con la acción solicitada por el usuario, como escribir en el archivo o la base de datos, leer desde el archivo o la base de datos, utilizar una aplicación de software concreta para acceder a la información, etc.).

En dicha relación triádica, tanto el 'token de usuario' como el 'token de aplicación' (acción deseada) de la aplicación solicitada por el usuario para acceder/manipular la información debe dominar el 'token de información' para que la acción pueda permitirse.

Una relación triádica se explica mejor por referencia a un ejemplo. En el ejemplo, se asume que el usuario quiere leer o actualizar un dato. Para acceder a un dato, el usuario no solo debe dominar la información, sino también poseer la "herramienta" adecuada (recurso de procesamiento) para acceder a la información.

Por ejemplo, el usuario Bob quizá desee acceder a un documento titulado "security101" para pasar el documento por un procesador de texto (p. ej., para añadir una entrada). El usuario Bob posee un token de seguridad, la aplicación de procesado de texto posee un token de seguridad y el documento security101 posee un token de seguridad.

Para que Bob pueda pasar el documento security101 por un procesador de texto:

1. El token de Bob debe dominar el token de información; y
2. El token de procesado de texto debe dominar al token de información.

Bob podrá pasar el documento security 101 por un 'procesador de texto' solo si ambas relaciones de dominancia son ciertas.

El proceso se ilustra con referencia a las Figuras 8 y 9. Cabe destacar que el proceso ilustrado con referencia a las Figuras 8 y 9 es relevante para una forma de realización donde se utilice un proceso de codificación exponencial.

En referencia a la Figura 8, el número entero se descodifica en una lista de etiquetas de seguridad de información - 'Info_list' (801). El número entero se decodifica a continuación en una lista de etiquetas de seguridad ('Dom_list') del usuario o de la aplicación cuya dominancia se comprobará contrastándola con la info_list (802). Abra el primer elemento de info_list (803). Abra el primer elemento de Dom_list (804). Compruebe que el primer elemento de Dom_list es mayor que el primer elemento de info_list. Si el Dom_level es inferior a info_level, la función dominar no se aprueba (807) y el algoritmo termina (808). Por el contrario, si Dom_level es superior a info_level, podrá pasar a determinar si la lista de compartimentos (info_list) está vacía (810). El flag dominar está configurado para fallo (806) asumiendo que los compartimentos del usuario o de la aplicación (Comp_dom) no sean un superconjunto de los compartimentos de info_list (Comp_info) o que la info_list restante tenga un conjunto vacío de compartimentos.

Si el compartimento está vacío (810), el algoritmo terminará (818). Si info_list está vacía (810), la relación de dominancia habrá fallado (Flag dominar = fallo) o la relación de dominancia se habrá aprobado (Flag dominar = aprobado).

Si info_list no está vacía (810), se abrirá el primer compartimento de Comp_info (811) y se realizará una comprobación para determinar si Dom_list está vacía (812). Si Dom_list está vacía (812) la función dominar se comprobará para garantizar que no ha fallado (817) y el algoritmo pasará al siguiente elemento de info_list (810) repitiéndose el proceso para todos los elementos de la lista Comp_info.

Si la función dominar ha fallado (817) (esto es, si el flag dominar se ha configurado en fallo), el algoritmo terminará (818). Si la lista Dom_list restante no está vacía (812), el siguiente compartimento (elemento Dom) se abrirá en Dom_list (813) y se comprobará para ver coincidencias con el compartimento de la lista Comp_info (814). Si no se encuentran coincidencias, el flag dominar se configurará en 'fallo' (816), de lo contrario, se configurará en aprobado (815). El proceso (812, 813, 814) se repetirá utilizando la lista de Comp_Dom buscando una coincidencia con el info_element actual que haya configurado el flag dominar en aprobado (815).

En relación ahora a la figura 9, se describe una relación triádica. El flag 'does_dominate' se pone inicialmente como 'no' (901). Usando el algoritmo de la figura 8, la función 'user_token domina el info_token' se ejecuta devolviendo un resultado user_dominate (802). Usando el algoritmo de la figura 8, la función 'application_token domina el info_token' se ejecuta devolviendo un resultado application_dominate (903).

Si el flag user_dominate es verdadero (904) se realiza una comprobación para determinar si el flag application_dominate es verdadero (905).

Si es verdadero, la relación triádica del flag 'Does_dominate' (906) se pone como verdadera y el algoritmo termina (907).

En caso contrario, si el flag user_dominate es 'falso' y/o si el flag application_dominate es falsa, el flag Does_dominate es falso (tal y como se puso inicialmente) y el algoritmo termina (907).

La relación de dominancia se puede combinar con la codificación de la etiqueta para mejorar la seguridad en un sistema informático o en una estructura de bases de datos. Para conjuntos de datos grandes como el componente de base de datos de un sistema informático se representan todas las combinaciones de etiquetas de seguridad como números enteros codificados (pares ordenados) y todas las combinaciones de las relaciones de dominancia se ejecutan antes de utilizarse en la base de datos y el resultado de acceso que aprueba (1) o falla (0) se guarda con el par ordenado. Esto se ilustra mejor mediante el siguiente ejemplo (donde se utiliza la base 2 simplemente por conveniencia):

Paso 1 - Las etiquetas de seguridad se codifican como números enteros:

compartimentos	= [patología, cardio]
patología	= 2
cardio	= 4
niveles	= [público, secreto]
público	= 8
secreto	= 16

Paso 2 - Así los compartimentos se pueden combinar:

secreto [patología]	= 16 + 2 = 18
secreto [Cardio]	= 16 + 4 = 20
secreto [patología, cardio]	= 16 + 2 + 4 = 22
público [patología]	= 8 + 2 = 10
público [Cardio]	= 8 + 4 = 12
público [patología, cardio]	= 8 + 2 + 4 = 14

Paso 3 - Esto permite la creación de tuplas de dominancia de combinación:

secreto [patología] domina público [patología] = < 18, 10, 1 >
secreto [Cardio] domina secreto [patología, cardio] = < 20, 22, 0 >
secreto [patología, cardio] domina secreto [Cardio] = < 22, 20, 1 >
público [cardio] domina secreto [Cardio] = < 12, 20, 0 >
secreto [cardio] domina público [Cardio] = < 20, 12, 1 >

Tal y como se puede ver en el ejemplo anterior, la función de dominancia se llama para devolver un 1 o un 0 al crear la lista de tuplas de combinación de seguridad. Las tuplas de combinación y su permiso de acceso asociado (1 o 0) se cargan en la base de datos para que se le conceda o se le deniegue a la búsqueda rápida en la tabla el acceso del registro y el filtrado asociado a cualquier declaración seleccionada.

Se puede obtener velocidad de acceso adicional filtrando las tuplas de combinación de seguridad en una lista basada

en la sesión de la base de datos del usuario, y guardar como una tabla en la memoria residente cuando el usuario inicia la sesión.

Por ejemplo, si un usuario tiene la etiqueta 'secreto [cardio]' solo las tuplas < 20, 22, 0 > y < 20, 12, 1 > (del ejemplo anterior) necesitarían almacenarse en una tabla de memoria no volátil para usarla en esa sesión de base de datos. Un mecanismo similar se puede utilizar para las etiquetas de seguridad en las funciones (aplicaciones) que el usuario tiene permiso para utilizar.

Una vez se ha cargado la "tabla de tupla de seguridad" en la base de datos, una sola línea de código SQL adicional en la cláusula *where* proporciona el filtrado necesario para la protección de la información.

La relación de dominancia triádica previamente definida se implementa ampliando el concepto de 'tupla de seguridad' para cubrir "la aplicación domina la información" además de el usuario domina la información".

Por ejemplo, continuando con el ejemplo anterior:

Usuario = secreto [patología, cardio] = 22
Aplicaciones = secreto [cardio] = 20
Información = secreto [cardio] = 20
secreto [patología, cardio] domina secreto [Cardio] = < 22, 20, 1 >
secreto [cardio] domina secreto [Cardio] = < 20, 20, 1 >

Por tanto, el usuario tendría acceso utilizando el conjunto particular de etiquetas de seguridad de información, aplicación y usuario.

Donde el token se forma utilizando número primos y aritmética de módulo, por ejemplo:

'Etiqueta 1 = Altamente sensible' {'identificadores personales', 'resultados del análisis de datos'}
= 37 x 31 x 29 x 23 x 3 x 5
= 11475735
etiqueta 2 = 'Público' {'Factores de riesgo del VPH'}
= 23 x 19
= 437

Ahora la función de dominancia se puede realizar como una llamada de la función módulo, en concreto:

Si (información = etiqueta 1) (usuario = etiqueta 2) entonces Etiqueta 2 domina etiqueta 1 = módulo(437, 11475735) fallaría (ya que no es igual a cero);
Si (información = etiqueta 2) (usuario = etiqueta 1) entonces Etiqueta 1 domina etiqueta 2 = módulo(11475735, 437) también fallaría (no es igual a cero) dado que los 'factores de riesgo del VPH' no forman parte de la etiqueta de usuario para ver esos detalles;
Si los factores de riesgo del VPH se incluyen como parte de la etiqueta 1, es decir, 11475735 * 19 = 218038965 entonces
Etiqueta 1 domina etiqueta 2 = módulo(218038965, 437) aprobaría entonces (igual a cero) dado que los 'factores de riesgo del VPH' forman parte ahora de la etiqueta de usuario para ver esos detalles.

Este enfoque permite construir un envoltorio de base de datos proporcionando seguridad a nivel de fila para cualquier base de datos SQL.

Además, la relación triádica proporciona la capacidad de construir un servidor proxy que actúe como una capa transparente (desde el punto de vista del usuario) entre la base de datos y el usuario final, de forma que ese contenido se filtre en base a la función de dominancia aquí descrita. Es decir, el usuario final solo tiene permitido el acceso a la información en base a si el usuario "domina" la información.

Utilizando cualquiera de las metodologías antes descritas, se obtiene una gran cantidad de ventajas. Por ejemplo, la implementación del procedimiento no requiere ningún cambio fundamental en la forma en la que se organiza una base de datos, y se puede incluir en las estructuras de bases de datos existentes.

El procedimiento también tiene el efecto de ocultar (haciendo que no existan) los datos para cualquier usuario o aplicación donde las características de tupla de seguridad del usuario o la aplicación no dominen el token de

información almacenado a nivel de fila. Esto proporciona un nivel de confusión que puede disuadir a los "hackers" casuales.

Si fuera necesario, se puede implementar una relación de dominio binario o triádico a nivel de mensajería de objetos o incluso integrada a nivel del sistema operativo y/o a nivel de comunicaciones, para proporcionar protección y seguridad adicionales.

Se proporciona una mayor flexibilidad a la hora de gestionar recursos asignados para manejar el procesamiento de información etiquetada de seguridad.

Además, allí donde se utiliza el procedimiento exponencial, una unión de tabla utilizando información codificada es más rápida que una función de base de datos ejecutándose a nivel de fila, cuando se devuelven o se manipulan grandes cantidades de filas de la tabla. Esto acelera, preferiblemente, el procesamiento de cualquier solicitud.

Las realizaciones de la invención, cuando se implementen, dan lugar a la generación y utilización de un gran número de niveles de seguridad. Esto a cambio precisa un sistema de gestión para administrar las etiquetas de forma correcta y eficiente.

El sistema de gestión puede estar construido con referencia a un "modelo de seguridad", construido para una organización o institución particular implementando las etiquetas de seguridad dentro de sus sistemas informáticos. El modelo formaría un patrón de arquitectura de seguridad, implementación y funcionamiento.

Una organización identificaría los riesgos (amenazas) y los reduciría mediante el diseño de una arquitectura adecuada de etiquetas de seguridad. La arquitectura de etiquetas de seguridad podría incluir una cantidad de agrupamientos conceptuales, incluidos:

- roles de persona;
- niveles de rol;
- compartimentos; y
- aplicaciones (funciones).

Así, los agrupamientos pueden estar organizados en una jerarquía de dominancia de etiquetas de seguridad adecuada y se puede concebir un conjunto de reglas para permitir la interacción necesaria de las etiquetas para reducir los riesgos identificados (amenazas).

Para permitir una implementación exitosa de la invención, se puede implementar un modelo de etiquetas de seguridad utilizando la notación del modelado del UML que se puede ampliar para cubrir todos los conceptos de etiquetas de seguridad consagrados en una forma de realización de la invención aquí descrita.

Los siguientes pasos describen una metodología que se puede utilizar para crear una arquitectura de modelo de seguridad organizativa adecuada

1. "*Diagramas de casos de uso"* En particular el análisis de los "participantes" (y jerarquía de participantes) del sistema identificará los distintos niveles iniciales de seguridad necesarios para reducir cualquier lista de candidatos de amenazas (o árbol de amenazas) en términos de los "participantes" (usuarios y demás sistemas) del sistema a estudiar. Los distintos 'niveles' de seguridad se anotarían o registrarían en el diagrama de casos de uso.

2. "*Diagramas de casos de uso"* En particular el análisis de "casos de uso" (procesos o servicios) identificará los distintos niveles iniciales de seguridad necesarios para reducir cualquier lista de candidatos de amenazas (o árbol de amenazas) en términos de aplicaciones (funciones). Los distintos 'niveles' de seguridad se anotarían o registrarían en el diagrama de casos de uso.

3. "*Diagramas de casos de uso"* En particular el análisis de "participante" y "casos de uso" interactuando (la interacción entre el usuario, otros sistemas y servicios) aclarará los distintos niveles de seguridad identificados en el PASO 1 y el PASO 2 e identificará el conjunto adecuado de compartimentos necesarios para reducir cualquier lista de candidatos de amenazas (o árbol de amenazas) en términos de aplicaciones e interacciones de usuario. Los distintos 'compartimentos' y 'niveles' de seguridad se anotarían o registrarían en el diagrama de casos prácticos.

4. "*Diagramas de clase"* En particular el análisis de "clases de objetos" y la "asociación de clases de objetos" permitirá la identificación de dónde implementar etiquetas de seguridad (nivel y compartimentos) en la arquitectura de software.

Los distintos 'niveles' y 'compartimentos' de seguridad se podrían anotar o registrar en el diagrama de clase como 'procedimientos' y 'atributos' de clases de objeto.

5. Los "*Diagramas de interacción de objetos"* y "*Diagramas de secuencia de objetos»* se usarían para revisar las dinámicas de dominancia de etiquetas de seguridad (sujeto domina al objeto), tanto binaria como triádica tal y como se describe anteriormente. El fin es entender las dinámicas de programación (acceso y cierre) producidas por la interacción de las etiquetas de seguridad.

6. Los "*Diagramas de modelado de relación de entidad"* (que no forman parte del procedimiento UML) se deben anotar con etiquetas de seguridad (atributos de tablas) tanto modelados de forma independiente para una arquitectura de datos específica como derivada del modelado del UML previo, PASO 1 a PASO 5 anteriores.

7. Los "*Diagramas de interacción de objetos"* y los "*Diagramas de secuencia de objetos"* se debería volver a visitar después del PASO 7, esta vez con el fin de revisar la arquitectura de mensajería. Reducir los riesgos de envío de mensajes entres sistemas y entre clases de objetos que interactúen, es decir, "procedimientos de clase" además de las interacciones de "procedimiento de objeto".

Llevando a cabo los PASOS anteriores, el modelo de seguridad para una arquitectura de sistema informático de 'n' capas proporciona gran cantidad de ventajas. Las amenazas de la "capa de datos" se reducen mediante una configuración de etiquetas de seguridad asociada tanto a nivel de fila como de tabla.

Las amenazas de la "capa intermedia" se reducen mediante una disposición de etiquetas de seguridad en instanciaciones de clase de objeto, dando lugar a controles de seguridad en las interacciones de mensajería de objetos y de clases.

Las etiquetas de seguridad integradas en la capa de comunicaciones encriptadas reducen las amenazas de las interacciones de "Usuario" y "Sistema externo". Los algoritmos de encriptado estándar tales como el RSA se pueden utilizar para encapsular las etiquetas de seguridad.

Se entenderá que aunque en los ejemplos anteriores se utiliza una forma exponencial de base 2 por comodidad, se podrá utilizar cualquier base.

También se entenderá que aunque los tokens aquí descritos son etiquetas utilizadas para controlar el acceso a la información en una base de datos, el mismo concepto se puede aplicar a la clasificación y protección de la información en otras áreas informáticas, como controlar la mensajería entre objetos de software, o controlar las comunicaciones entre sistema informáticos.

También se entenderá que mientras las realizaciones descritas se refieren a un sistema de clasificación/aplicación médica, las realizaciones del procedimiento y del sistema de acuerdo con la presente invención se pueden utilizar en cualquier tipo de sistema informático, o cualquier tipo de base de datos, para guardar cualquier tipo de dato, utilizando cualquier combinación adecuada de compartimentos y niveles de seguridad.

REIVINDICACIONES

1. Un sistema para codificar un token de seguridad asociado con un conjunto de información en una base de datos en un sistema informático (100), conteniendo el token de seguridad características del dato, comprendiendo el sistema:

los medios para asignar (102, 106, 104, 108, 116, 118, 120, 122) un número entero único a cada uno de los conjuntos de características, comprendiendo el conjunto diversas características;
los medios para asignar a cada uno de los datos en una base de datos como mínimo una característica que describa el dato;
los medios para crear un token de seguridad de información combinando (102, 106, 104, 108, 116, 118, 120, 120), donde el dato tiene más de una característica, los número enteros asignados a cada característica del conjunto de características que describe el token de seguridad de información, en un token de seguridad de información codificado que comprende un único número entero, para confundir cada uno de los conjuntos de características codificadas en el token de seguridad;
los medios para crear un token de usuario asociado con un usuario, conteniendo el token de seguridad de usuario características que describan el conjunto de información al que se le permite acceder al usuario, comprendiendo el sistema los medios para combinar los números enteros únicos de las características a las que se le permite acceder al usuario en un token de usuario codificado; y
los medios para utilizar el token de seguridad de información codificado y el token de usuario codificado en un sistema informático y para controlar el acceso del usuario a la información de la base de datos para los usuarios cuyo token de seguridad de usuario sea un superconjunto del token de información, teniendo el token de usuario, como mínimo, el conjunto de características del token de información codificado.

2. Un sistema de acuerdo con la reivindicación 1, donde el número entero asignado a cada uno de los conjuntos de características está determinado por una exponencial con la forma x^y, donde x e y son números enteros.

3. Un sistema de acuerdo con la reivindicación 1, donde, al asignar un número entero a cada uno de los conjuntos de características capaces de describir el token, el exponencial y se incrementa para cada uno de los conjuntos de características.

4. Un sistema de acuerdo con la reivindicación 1, la reivindicación 2 o la reivindicación 3, donde los números enteros se combinan mediante la suma de los números enteros.

5. Un sistema de acuerdo con la reivindicación 1, donde el número entero asignado a cada uno de los conjuntos de características es un único número primo.

6. Un sistema de acuerdo con la reivindicación 5, donde los números enteros se combinan mediante la multiplicación de los números enteros.

7. Un sistema de acuerdo con cualquiera de las reivindicaciones 1 a 6, donde el conjunto de características es uno de los niveles de permiso y un compartimento.

8. Un sistema para permitir a un usuario acceder a un dato contenido en un sistema informático (100), que comprende:

el sistema para codificar un token de seguridad de acuerdo con la reivindicación 1;
los medios (102, 104, 106, 108, 116, 118, 120, 122) para asignar un token de seguridad a todos los datos discretos en el sistema informático (100), asignando un token de seguridad a todos los usuarios del sistema informático (100), y asignando un token de seguridad a todas las aplicaciones de software que residen en el sistema informático (100), donde, para acceder a un dato discreto, el token de seguridad del usuario y la aplicación de software deben dominar el token de seguridad asignado al dato discreto.

9. Un sistema para construir una lista que contenga información referente a si cada uno de los múltiples usuarios puede acceder a cada uno de los múltiples datos contenidos en un sistema informático (100), que comprende:

los medios para determinar (102, 104, 106, 108, 116, 118, 120, 122) un token de seguridad para cada uno de los usuarios y un token de seguridad para cada uno de los datos de acuerdo con el sistema de cualquiera de las reivindicaciones 1 a 8,
los medios para determinar (102, 104, 106, 108, 116, 118, 120, 122) si cada uno de los múltiples usuarios puede

acceder a cada uno de los múltiples datos de acuerdo con el sistema de la reivindicación 1; y
los medios para construir (102, 104, 106, 108, 116, 118, 120, 122) una lista de tokens de seguridad y permisos de acceso.

10. Un sistema para decodificar un token de seguridad de la forma descrita en la reivindicación 3, que comprende:

los medios de determinación (102, 104, 106, 108, 116, 118, 120, 122) para determinar el suelo de la base del logaritmo del número entero único para determinar un valor para la primera del conjunto de características,
los medios de sustracción (102, 104, 106, 108, 116, 118, 120, 122) para restar el suelo de la base del logaritmo del número entero único del número entero para producir un número entero de resto, y
los medios (102, 104, 106, 108, 116, 118, 120, 122) para determinar si el valor de resto es distinto de cero, lo que invoca los medios de determinación (102, 104, 106, 108, 116, 118, 120, 122) para determinar el suelo del logaritmo del número entero de resto para determinar otro más de los conjuntos de características,
donde el proceso se repite hasta que el valor del resto es cero.

11. Un sistema para decodificar un token de seguridad de la forma descrita en la reivindicación 5, que comprende:

los medios de determinación (102, 104, 106, 108, 116, 118, 120, 122) para determinar el módulo del número entero único, y
los medios (102, 104, 106, 108, 116, 118, 120, 122) para determinar si el valor del resto es distinto de cero,
donde, si el valor es cero, la característica es inherente en el token de seguridad.

12. Un procedimiento para codificar un token de seguridad asociado con un conjunto de información en una base de datos en un sistema informático (100), conteniendo el token de seguridad características del dato, comprendiendo el procedimientos los pasos de:

asignación de un número entero único a cada uno de los conjuntos de características, comprendiendo el conjunto múltiples características;
asignación a cada uno de los datos en una base de datos como mínimo una característica que describa el dato;
creación de un token de seguridad de información combinando, donde el dato tiene más de una característica, los números enteros asignados a cada característica del conjunto de características que describe el token de seguridad de información, en un token de seguridad de información codificado que comprende un único número entero, para confundir cada uno de los conjuntos de características codificadas en el token de seguridad;
creación de un token de usuario asociado con un usuario, conteniendo el token de usuario características que describan el conjunto de información al que se le permite acceder al usuario, comprendiendo el procedimiento los pasos de combinación de los números enteros únicos de las características a las que se le permite acceder al usuario en un token de usuario codificado; y
la utilización del token de seguridad de información codificado y el token de usuario codificado en un sistema informático y para controlar el acceso del usuario a la información de la base de datos para los usuarios cuyo token de seguridad de usuario sea un superconjunto del token de información, teniendo el token de usuario, como mínimo, el conjunto de características del token de información codificado.

13. Un procedimiento de acuerdo con la reivindicación 12, donde el número entero asignado a cada uno de los conjuntos de características está determinado por una exponencial con la forma x^y, donde x e y son números enteros.

14. Un procedimiento de acuerdo con la reivindicación 13, que comprende el paso adicional de, para el paso de asignar un número entero a cada uno de los conjuntos de características capaces de describir el token, incrementar el exponencial y para cada uno de los conjuntos de características.

15. Un procedimiento de acuerdo con la reivindicación 12, la reivindicación 13 o la reivindicación 14, donde los números enteros se combinen mediante la suma de los números enteros.

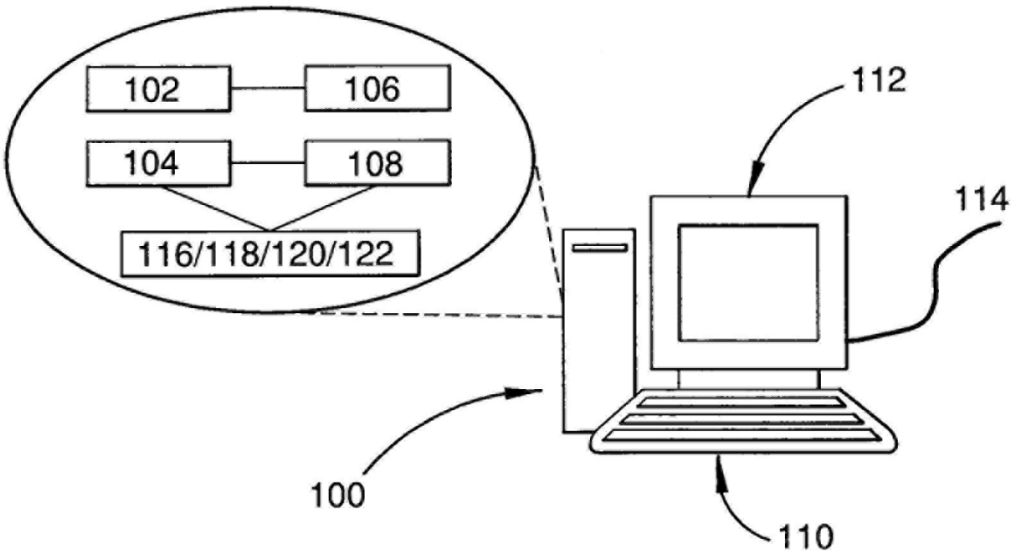

FIGURA 1

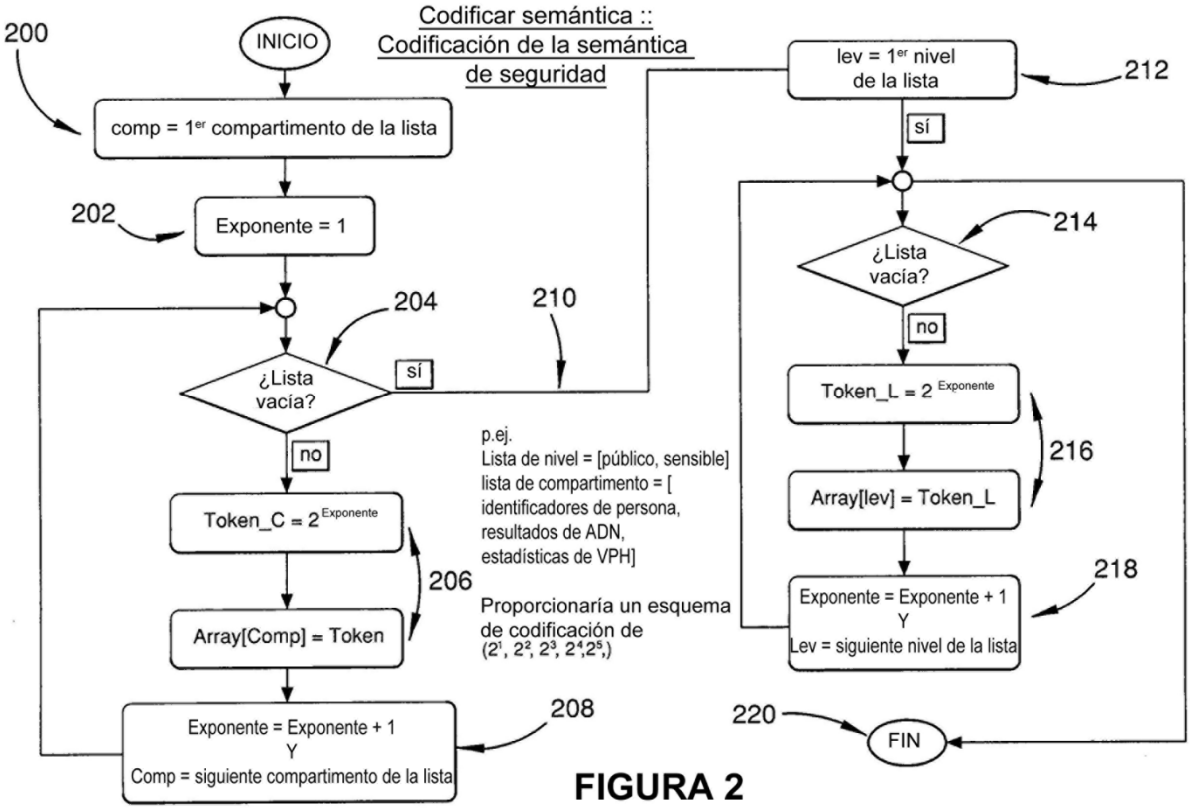
Codificar semántica ::
Codificación de la semántica de seguridad
200
INICIO
comp = 1er compartimento de la lista
202
Exponente = 1
204
¿Lista vacía?
sí
210
no
Token_C = 2 Exponente
206
Array[Comp] = Token
Exponente = Exponente + 1
Y
Comp = siguiente compartimento de la lista
208
p.ej.
Lista de nivel = [público, sensible]
lista de compartimento = [
identificadores de persona,
resultados de ADN,
estadísticas de VPH]
Proporcionaría un esquema de codificación de
(2¹, 2², 2³, 2⁴,2⁵,)
lev = 1er nivel de la lista
212
sí
214
¿Lista vacía?
no
Token_L = 2 Exponente
216
Array[lev] = Token_L
218
Exponente = Exponente + 1
Y
Lev = siguiente nivel de la lista
220
FIN

FIGURA 2

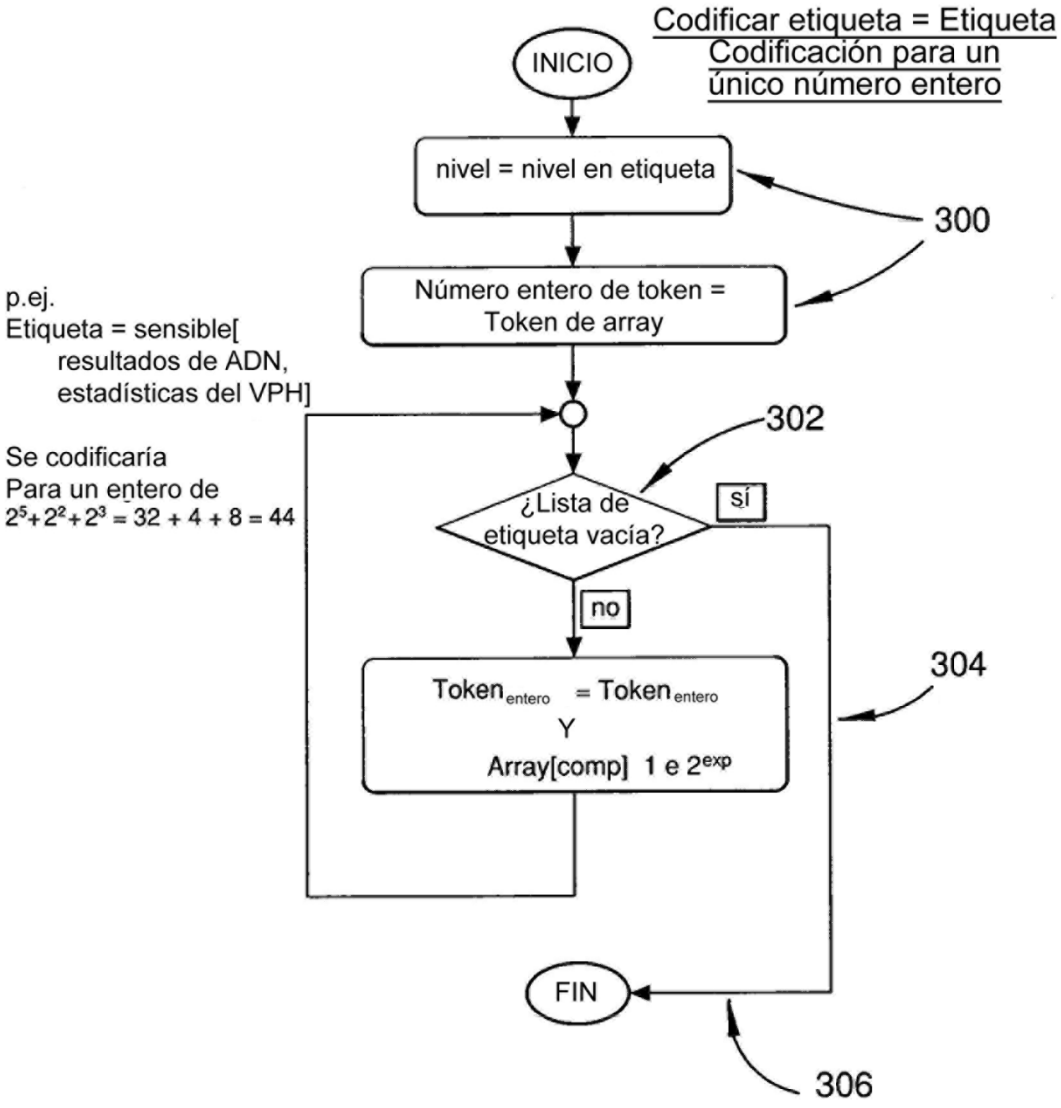
Codificar etiqueta = Etiqueta
Codificación para un
único número entero
INICIO
nivel = nivel en etiqueta
300
Número entero de token =
Token de array
p.ej.
Etiqueta = sensible[
resultados de ADN,
estadísticas del VPH]
Se codificaría
Para un entero de
$2^5+2^2+2^3 = 32 + 4 + 8 = 44$
302
¿Lista de
etiqueta vacía?
sí
no
304
$Token_{entero} = Token_{entero}$
Y
Array[comp] 1 e 2^{exp}
FIN
306

FIGURA 3

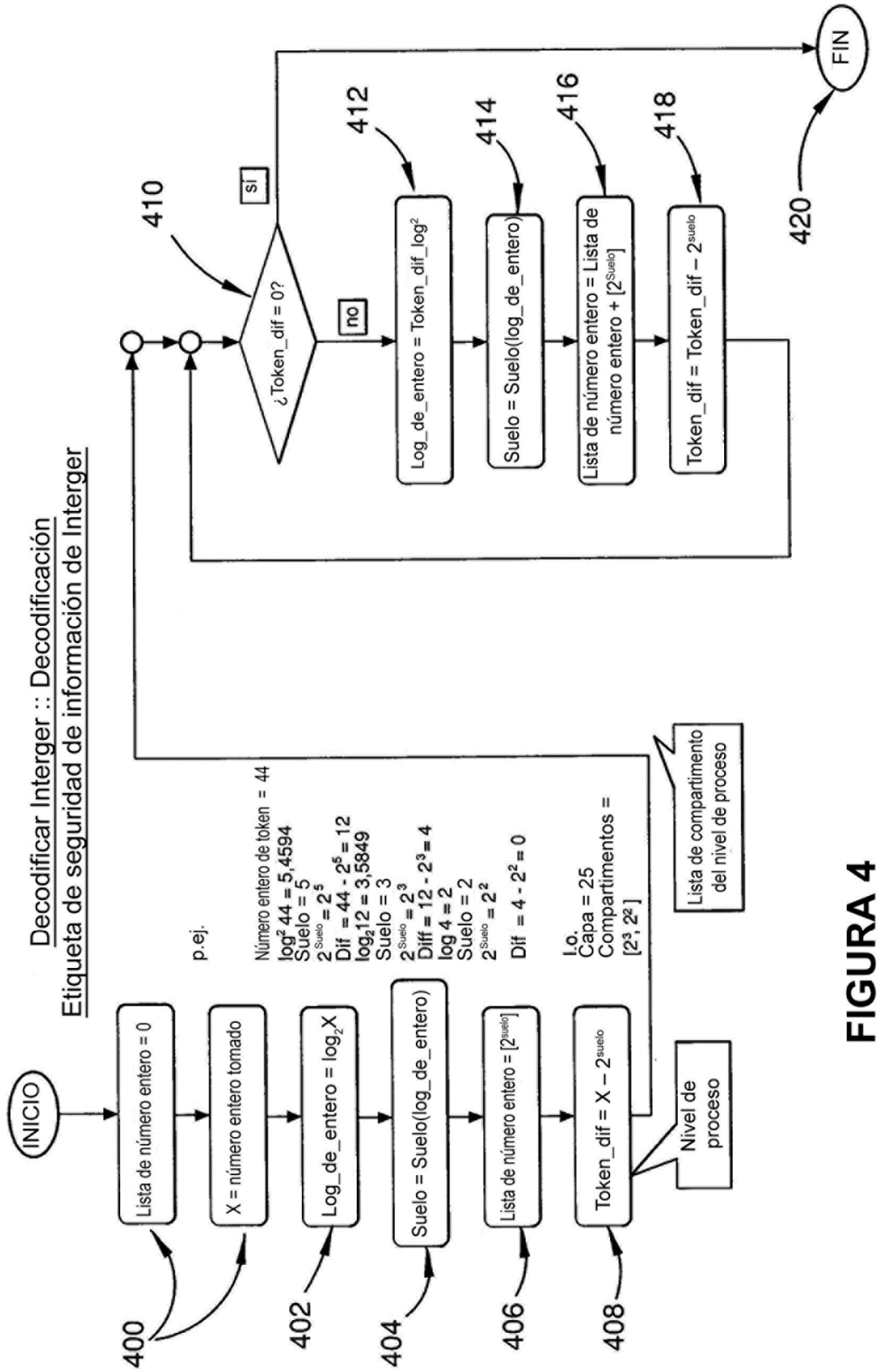
Decodificar Interger :: Decodificación
Etiqueta de seguridad de información de Interger
INICIO
Lista de número entero = 0
X = número entero tomado
Log_de_entero = log2X
Suelo = Suelo(log_de_entero)
Lista de número entero = [2suelo]
Token_dif = X – 2suelo
Nivel de proceso
Lista de compartimento del nivel de proceso
p.ej.
Número entero de token = 44
log2 44 = 5,4594
Suelo = 5
2 Suelo = 25
Dif = 44 - 25 = 12
log212 = 3,5849
Suelo = 3
2Suelo = 23
Diff = 12 - 23 = 4
log 4 = 2
Suelo = 2
2Suelo = 22
Dif = 4 - 22 = 0
I.o.
Capa = 25
Compartimentos = [23; 22]
400
402
404
406
408
410
¿Token_dif = 0?
sí
no
412
Log_de_entero = Token_dif_log2
414
Suelo = Suelo(log_de_entero)
416
Lista de número entero = Lista de número entero + [2Suelo]
418
Token_dif = Token_dif – 2suelo
420
FIN

FIGURA 4

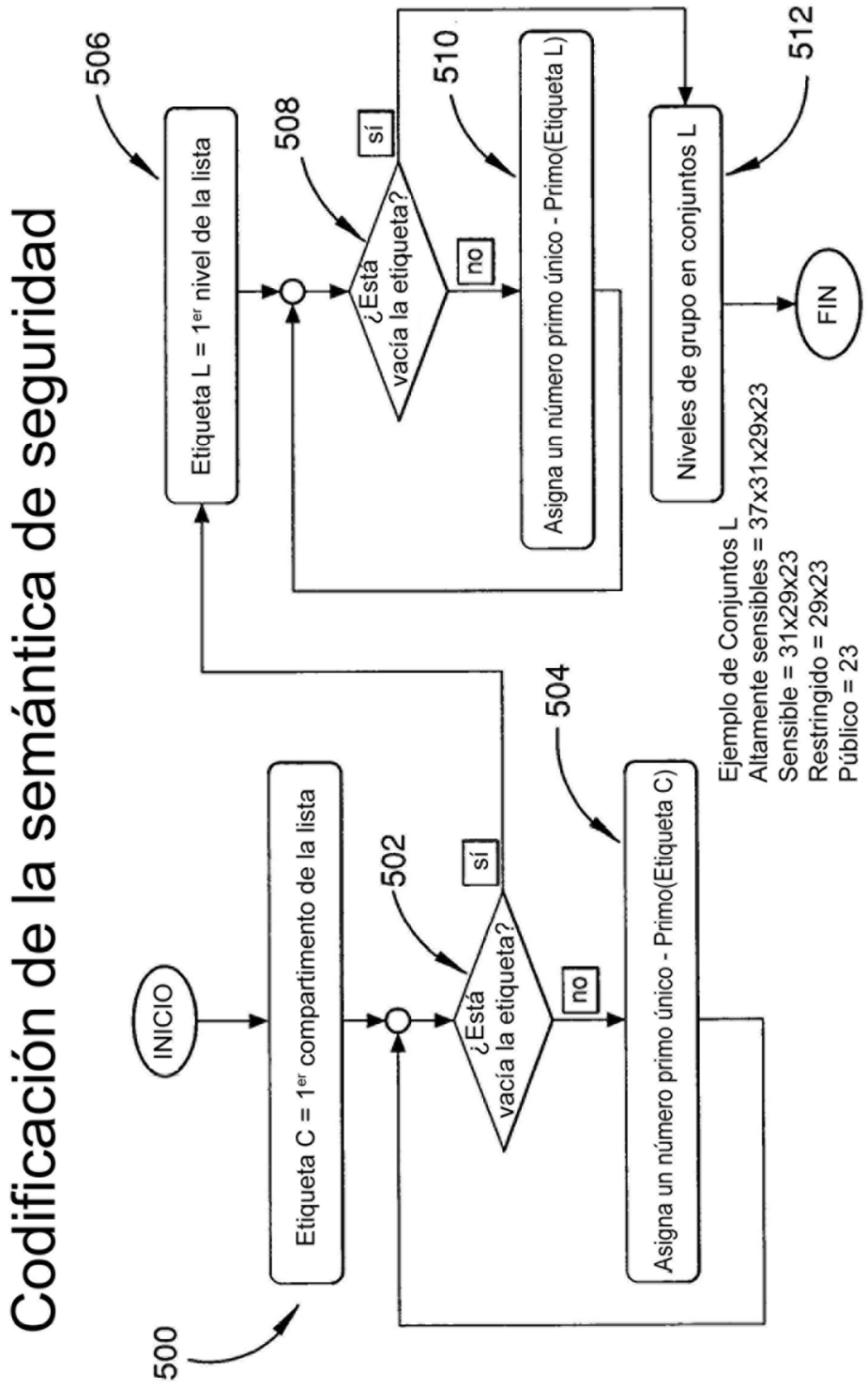
Codificación de la semántica de seguridad
500
INICIO
Etiqueta C = 1er compartimento de la lista
502
¿Está vacía la etiqueta?
sí
no
504
Asigna un número primo único - Primo(Etiqueta C)
506
Etiqueta L = 1er nivel de la lista
508
¿Está vacía la etiqueta?
sí
no
510
Asigna un número primo único - Primo(Etiqueta L)
512
Niveles de grupo en conjuntos L
FIN
Ejemplo de Conjuntos L
Altamente sensibles = 37x31x29x23
Sensible = 31x29x23
Restringido = 29x23
Público = 23

FIGURA 5

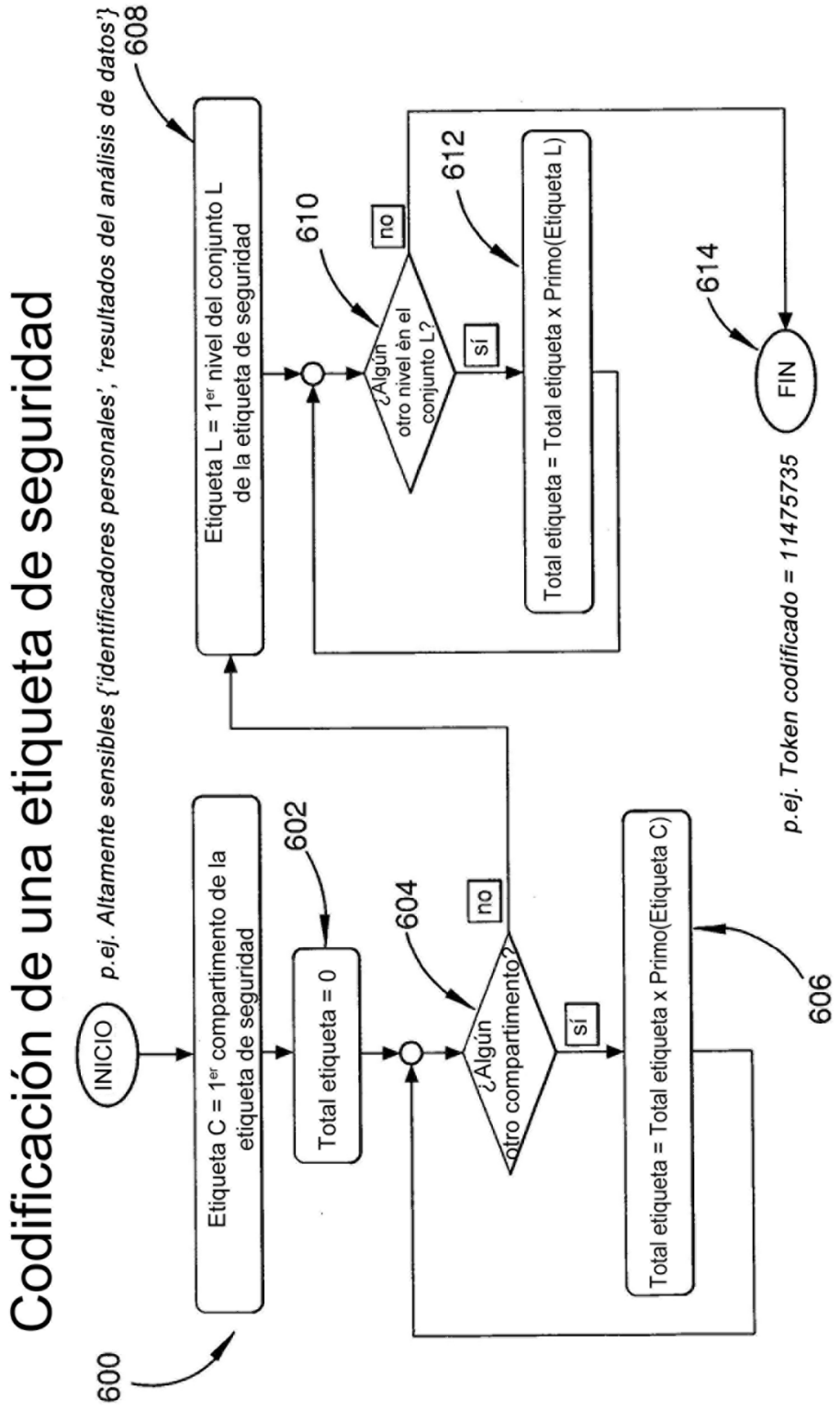
Codificación de una etiqueta de seguridad
600
INICIO
p.ej. Altamente sensibles {'identificadores personales', 'resultados del análisis de datos'}
608
Etiqueta C = 1er compartimento de la etiqueta de seguridad
602
Total etiqueta = 0
604
¿Algún otro compartimento?
no
sí
606
Total etiqueta = Total etiqueta x Primo(Etiqueta C)
Etiqueta L = 1er nivel del conjunto L de la etiqueta de seguridad
610
¿Algún otro nivel èn el conjunto L?
no
sí
612
Total etiqueta = Total etiqueta x Primo(Etiqueta L)
614
FIN
p.ej. Token codificado = 11475735

FIGURA 6

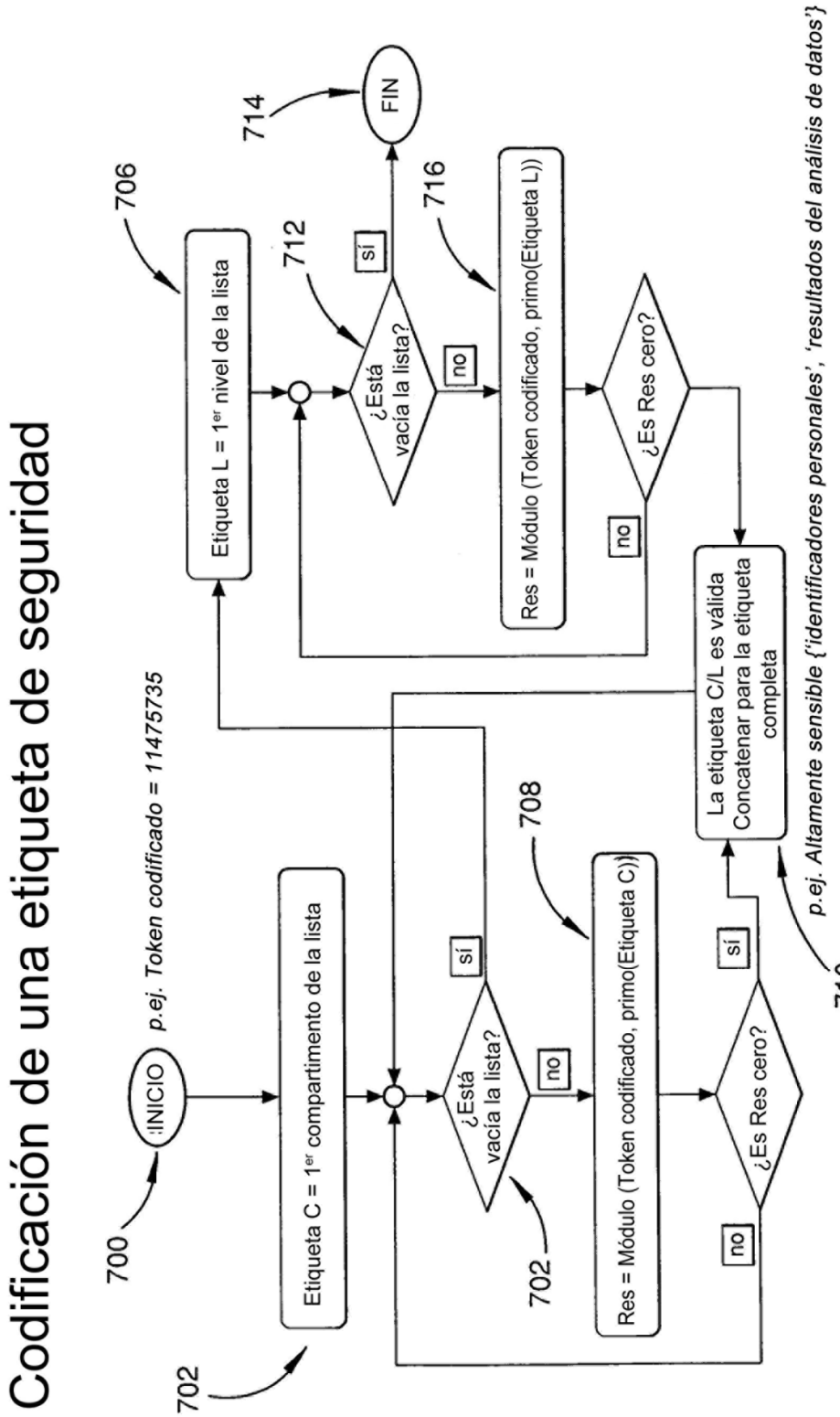
Codificación de una etiqueta de seguridad
¡INICIO
p.ej. Token codificado = 11475735
700
Etiqueta C = 1er compartimento de la lista
702
¿Está vacía la lista?
sí
no
702
Res = Módulo (Token codificado, primo(Etiqueta C))
708
¿Es Res cero?
sí
no
La etiqueta C/L es válida
Concatenar para la etiqueta completa
710
p.ej. Altamente sensible {'identificadores personales', 'resultados del análisis de datos'}
706
Etiqueta L = 1er nivel de la lista
712
¿Está vacía la lista?
sí
no
714
FIN
716
Res = Módulo (Token codificado, primo(Etiqueta L))
¿Es Res cero?
no

FIGURA 7

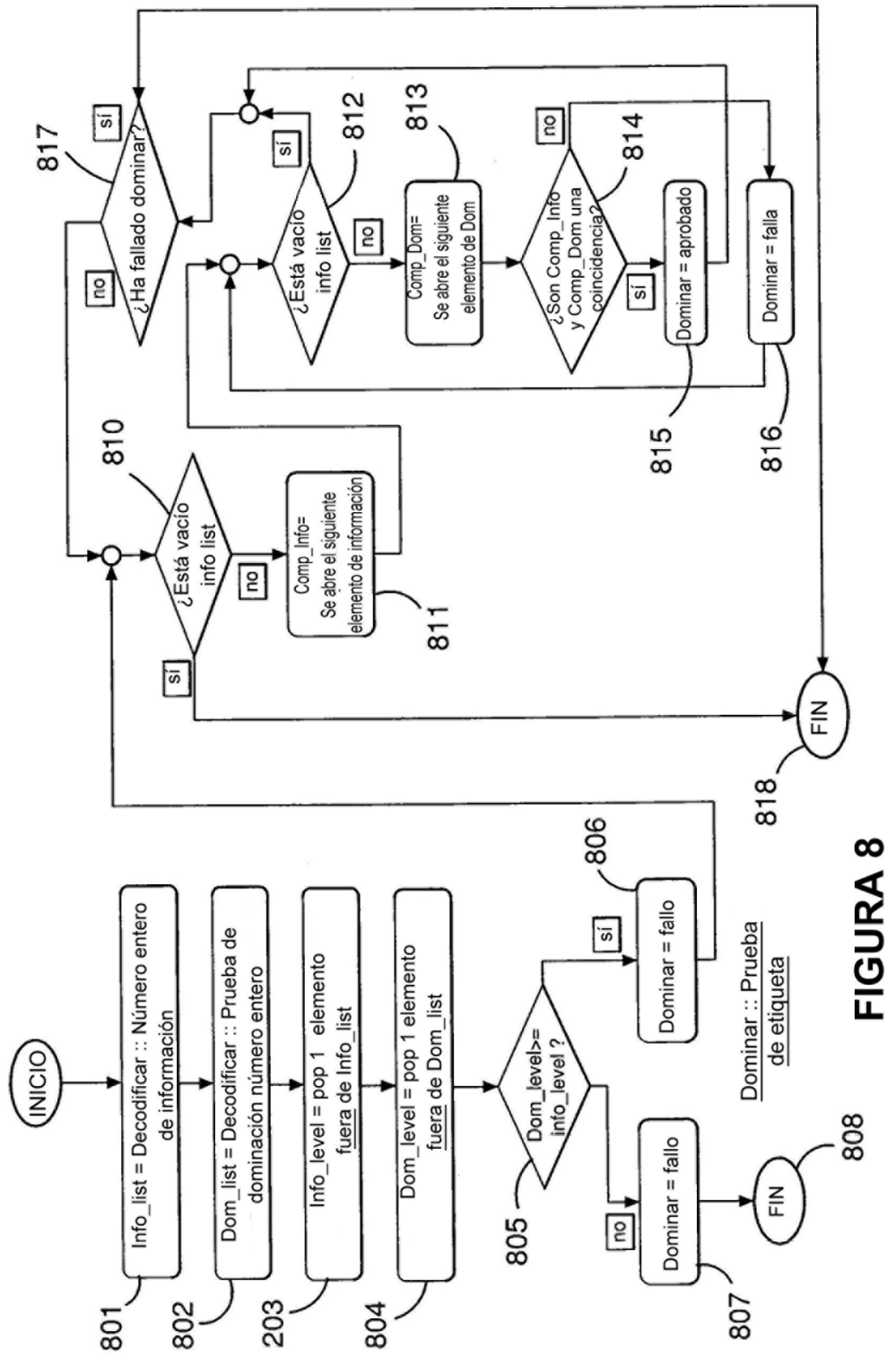
INICIO
Info_list = Decodificar :: Número entero de información
Dom_list = Decodificar :: Prueba de dominación número entero
Info_level = pop 1 elemento fuera de Info_list
Dom_level = pop 1 elemento fuera de Dom_list
Dom_level>= info_level ?
sí
no
Dominar = fallo
Dominar = fallo
FIN
Dominar :: Prueba de etiqueta
¿Está vacío info list
no
Comp_Info= Se abre el siguiente elemento de información
sí
¿Está vacío info list
no
Comp_Dom= Se abre el siguiente elemento de Dom
¿Son Comp_Info y Comp_Dom una coincidencia?
no
sí
Dominar = aprobado
Dominar = falla
¿Ha fallado dominar?
no
sí
FIN
801
802
203
804
805
806
807
808
810
811
812
813
814
815
816
817
818

FIGURA 8

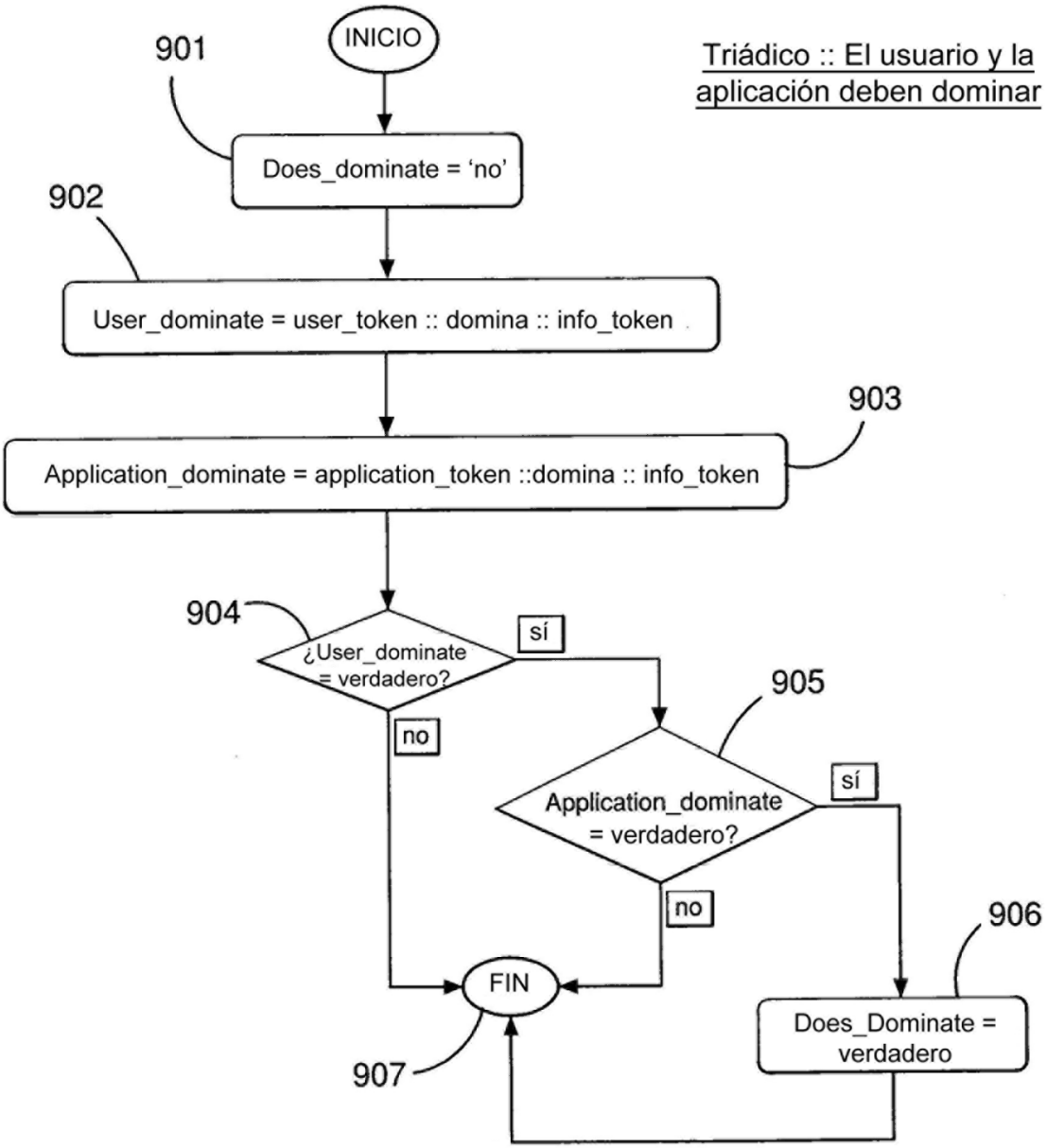
Triádico :: El usuario y la aplicación deben dominar
INICIO
901
Does_dominate = 'no'
902
User_dominate = user_token :: domina :: info_token
903
Application_dominate = application_token ::domina :: info_token
904
¿User_dominate = verdadero?
sí
no
905
Application_dominate = verdadero?
sí
no
906
Does_Dominate = verdadero
FIN
907

FIGURA 9