(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 663 422**

(51) Int. CI.:

H04L 9/08 (2006.01)
H04L 9/32 (2006.01)
G06F 21/10 (2013.01)
G06F 21/62 (2013.01)
G09C 1/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **28.05.2013 E 13169589 (2)**

(97) Fecha y número de publicación de la concesión europea: **31.01.2018 EP 2670080**

(54) Título: **Sistema y método de protección de datos**

(30) Prioridad:

28.05.2012 IT TO20120462

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
12.04.2018

(73) Titular/es:

LEONARDO S.P.A. (100.0%)
Piazza Monte Grappa 4
00195 Roma, IT

(72) Inventor/es:

BARLETTA, ALESSANDRO

(74) Agente/Representante:

UNGRÍA LÓPEZ, Javier

DESCRIPCIÓN

Sistema y método de protección de datos

La presente invención se refiere a un sistema y método para proteger datos de información y, en particular, para verificar y conceder autorización para acceder a datos de información.

Se conocen métodos de protección de datos, en particular métodos para autorizar el acceso a información protegida y/o permitir la ejecución de programas de ordenador (software) a condición de obtener una licencia para usarlos.

Los métodos conocidos contemplan el uso de una clave no conocida por el usuario y que se puede obtener comprando una licencia de uso del software en cuestión. Este método tiene el inconveniente de que esta clave puede ser replicada de forma fraudulenta, de modo que los usuarios no autorizados que conozcan esta clave serán capaces de usar el software.

Otros métodos contemplan el uso de un dispositivo, llamado token, capaz de generar una contraseña, generalmente en un formato numérico, en base a un algoritmo desconocido para el usuario. Típicamente, en un esquema de autenticación de dos factores, esta contraseña asume un valor dependiente de parámetros que varían con el tiempo (por ejemplo, la hora, la fecha, etc), más una parte conocida, elegida por el usuario (conocida como el Número de Identificación Personal o, más simplemente, el PIN), de modo que no sea fácilmente replicable. Este método tiene el inconveniente de que todos conocen la hora y el PIN lo elige por lo general el usuario de modo que sea fácil de recordar (en la práctica, el PIN a menudo elegido es la fecha de nacimiento del usuario, los nombres de personas fallecidas, etc). Se conocen ataques de varios tipos que reducen drásticamente la protección proporcionada por este tipo de esquema.

Otros métodos conocidos incluyen los pasos de almacenar porciones enteras de software en el token, pero de esta forma hay un impacto en la velocidad de ejecución del software, unido al hecho de que hay que acceder a estas porciones, compilarlas y luego ejecutarlas. En consecuencia, la comprobación se realiza en general solamente en la fase de arranque del software y no durante todo el período de tiempo en el que se usa el software.

El objeto de la presente invención es proporcionar un sistema y método de protección de datos capaz de superar los inconvenientes de la técnica anterior.

Con más detalle, el sistema de autenticación se basa en un token accesible por medio de un PIN, y está configurado para autorizar o denegar el acceso a la información a proteger, teniendo el token una memoria que guarda: una matriz conteniendo elementos de información numéricos y/o alfanuméricos y/o alfabéticos (I_1-I_N), y datos numéricos y/o alfanuméricos y/o alfabéticos aleatorios interpuestos entre elementos de información inmediatamente consecutivos en un orden de lectura de la matriz; y un vector de esquema de sello encriptado (o matriz de esquema de sellado), conteniendo información relativa a la disposición de los elementos de información en la matriz e información relativa al tamaño (típicamente en bytes) de cada elemento de datos aleatorios, con el fin de permitir el acceso a cada uno de los elementos de información en sus posiciones respectivas en la matriz. El sistema de autenticación está configurado para: supeditar el acceso al token a introducir un PIN correcto; desencriptar el vector de esquema de sello; adquirir la información de disposición de los elementos de información y la información en el espacio ocupado por cada elemento de datos aleatorios del vector de esquema de sello; comprobar la correspondencia entre la información de disposición adquirida y la disposición efectiva de los elementos de información en la matriz, y entre la información de tamaño adquirida y el tamaño efectivo de los datos aleatorios; autorizar o denegar el acceso a la información a proteger en base a un resultado de la comprobación previa.

El documento US2011289576 se refiere a un método y aparato para proporcionar un token para acceso seguro.

Según la presente invención, se facilitan un sistema y método de protección de datos como los definidos en las reivindicaciones anexas.

Para una mejor comprensión de la presente invención, algunas realizaciones preferidas se describirán ahora, puramente a modo de ejemplo no limitativo y con referencia a los dibujos adjuntos, donde:

La figura 1 representa esquemáticamente un dispositivo, o token, para generar claves de autenticación.

La figura 2 representa un ordenador, de tipo conocido, adecuado para conectarse operativamente al token de la figura 1.

La figura 3 representa, por medio de un diagrama de flujo, los pasos de un método de protección de datos según una realización de la presente invención.

La figura 4 representa, en forma gráfica, una matriz incluyendo una pluralidad de elementos de datos/información codificados según los pasos 10 y 12 del método de la figura 3.

La figura 5 representa la matriz de la figura 4 según una forma diferente de representación gráfica.

La figura 6 representa, por medio de un diagrama de flujo, los pasos secundarios del método de la figura 3.

La figura 7 representa, por medio de un diagrama de flujo, más pasos secundarios del método de la figura 3.

Y la figura 8 representa, por medio de un diagrama de flujo, más pasos del método de protección de datos según la presente invención.

La presente descripción se refiere a un sistema y método para proteger datos y, en particular, a cifrar o encriptar datos electrónicos.

Con más detalle, los niveles de protección proporcionados por la presente invención incluyen: un nivel físico, incluyendo un dispositivo de autenticación o token, capaz de generar un código de autenticación pseudoaleatorio, por ejemplo, en un formato numérico, alfabético o alfanumérico; y un nivel de software, incluyendo pasos criptográficos realizados por medio de un programa de ordenador.

El token de autenticación, esquemáticamente representado en la figura 1 e indicado con el número de referencia 1, incluye un microprocesador 2 de tipo conocido, capaz de generar, a petición del usuario, el código de autenticación pseudoaleatorio según un cierto algoritmo, que puede tomar en cuenta una pluralidad de factores, tales como, por ejemplo, el número de serie del token de autenticación 1, un valor de fecha actual, un valor de tiempo definido por un reloj dentro del token de autenticación 1, u otros factores. Este valor es generado mediante hardware por un algoritmo basado en el tiempo y puede variar entre diferentes fabricantes de tokens. Típicamente, no se describe cómo el código de autenticación pseudoaleatorio es generado por el token y cada proveedor de tokens adopta un algoritmo de propiedad. El método según la presente invención es independiente de la forma en la que se genera este código de autenticación y por ello éste último no se describe más aquí.

El token de autenticación 1 también incluye una memoria 4, por ejemplo, una memoria de tipo flash, conectada al microprocesador 2.

El programa para generar el código de autenticación reside en la memoria 4 del token de autenticación 1, en particular en una zona privada 4a de la memoria 4, a usar para almacenar datos sensibles accesibles por medio de una palabra clave, o PIN (Número de Identificación Personal). Una zona pública 4a de la memoria 4 puede almacenar certificados de naturaleza pública e información no sensible.

El PIN es generado automáticamente y no se pone a disposición del usuario final. A la instalación, o cuando el software es ejecutado por primera vez, al usuario se le indica que elija una contraseña de su agrado que desbloquee la utilización del PIN y guarde el hash (es decir, el resultado de una operación que es no reversible o reversible “con dificultad” - en una base de datos).

La zona privada 4a de la memoria 4 puede ser del tipo oculto (el usuario no es consciente de su presencia), o visible para el usuario, pero con acceso protegido por una palabra clave o contraseña. Esta contraseña puede ser modificada, a condición de que tenga el registro semilla generado por el proveedor y la autorización del administrador de token (conocido como el Oficial de Seguridad) que está protegida por otra contraseña, típicamente modificable y no sobrescribible, que se pone a un valor predeterminado y guarda en una posición segura.

La zona privada 4a de la memoria 4 puede ser usada para almacenar una pluralidad de datos y/o información, usada durante los pasos del método criptográfico, como se ilustra mejor a continuación.

En una realización, el token de autenticación 1 puede conectarse a un ordenador 6 (figura 2), de modo que la información pueda intercambiarse entre el token de autenticación 1 y el ordenador 6. Por ejemplo, el token de autenticación 1 podría estar equipado con un puerto USB para conexión al ordenador 6.

A continuación, se hace referencia a un texto sencillo que se desea codificar con el fin de evitar que un usuario no autorizado acceda a la información contenida en el texto sencillo.

Con referencia a la figura 3, se representa un diagrama de flujo que incluye pasos del método de encriptado según una realización de la presente invención.

Según la presente invención, se genera un “sello” que incluye una pluralidad de datos e información, elegido de forma discrecional, que se encripta según un esquema aleatorio. El desencriptado y la lectura de este sello es un requisito para autorizar o denegar una o más actividades adicionales.

Según una aplicación preferida, el desencriptado y la lectura de este sello es un requisito para autorizar la ejecución de un programa de software, u obtener una licencia de uso de dicho programa de software. Los pasos de encriptado

según el método de la presente invención no se dirigen, por lo tanto, a encriptar el programa de software a ejecutar, sino a generar un sello encriptado, cuya lectura es necesaria para obtener la autorización para ejecutar dicho programa de software.

El método según la figura 3 incluye pasos de inicialización (pasos 10 y 12) durante los que el sello es generado a partir del texto sencillo y se escribe en la zona privada 4a de la memoria 4, y pasos de encriptado (pasos 14 y 16) durante los que un método criptográfico de tipo conocido (por ejemplo, un método conocido elegido a partir de RSA, AES, DES, u otros) se usa para encriptar el sello generado en los pasos 10 y 12 con el fin de garantizar otro nivel de protección.

El sello se genera en base a los datos y/o la información que constituyen el texto sencillo, disponiendo estos datos en una matriz como se describe mejor a continuación. Además, el sello tiene un tamaño, en bits, de modo que sea compatible con el algoritmo de encriptado usado en los pasos 14 y 16 (por ejemplo, una longitud de 128 bits, o 196 bits, o 256 bits, o algún otro tamaño) y deje suficientes espacios “libres” a llenar con datos aleatorios.

El texto sencillo puede incluir una pluralidad de datos y/o información de un tipo fijo (cuyo valor o contenido de información no varía con el tiempo) y/o un tipo variable (cuyo valor o contenido de información no es el mismo si se considera en diferentes momentos en el tiempo). Los datos fijos incluyen, por ejemplo, códigos de identificación predeterminados en forma de números, letras y/o alfanuméricos o algún otro formato, tal como, por ejemplo, uno o más de: denominación comercial o nombre código del software, nombre del fabricante de software, número progresivo de licencia para el usuario actual del software, nombre de cliente y/o código de identificación, fecha de inicialización, número de serie del token, módulos de software para habilitar (en el caso de software compuesto de múltiples módulos) y contraseña/códigos de licencia asociados, y otros datos opcionales.

Los datos variables incluyen, por ejemplo, valores para la fecha actual, hora actual, u otros datos que pueden tomar un valor variable en base a condiciones predeterminadas o no planificadas.

Con respecto a los campos variables, la corrección de su contenido es verificada “por cita”. En otros términos, el componente de software comprueba que, dado un valor de tiempo escrito en el tiempo t_0, después de transcurrir “x” segundos, el valor de tiempo es tiempo t_0+x (más o menos una tolerancia del orden de cientos de milisegundos). El valor t_0 es el momento en que se escribe el valor de fecha/hora y “x” es una deriva variable determinada de forma instantánea y aleatoriamente por el software según una distribución uniforme libremente elegida, cuyos límites se especifican entre un mínimo y un máximo, generalmente del orden de decenas de segundos/minutos.

A continuación, en la descripción, cada elemento de datos, fijo o variable, se considera como un solo “elemento” y se indica con las referencias I_1, I_2, I_3, …, I_N. Una pluralidad N de elementos I_1-I_N de este tipo forma un conjunto (dispuesto en forma de un vector I) incluyendo un número N de elementos I_1-I_N:

$$I=\{I_1,\ I_2,\ I_3,\ I_4,\ I_5,\ \ldots,\ I_N\}$$

Como se ha mencionado, una aplicación de la presente invención es autorizar o denegar la ejecución de un programa de software, u obtener una licencia de uso de dicho programa de software. En este caso, se guarda una copia del vector I en el programa de software de una manera que no es accesible al usuario del programa de software.

Cuando se considera en su formato digital (como una secuencia de bits con valores lógicos “1” o “0”), cada elemento I_1-I_N del conjunto de N elementos I_1-I_N tiene su propia longitud L(i), mensurable como un número de bits y/o bytes, donde 1 byte es un grupo de 8 bits. En este caso, “i” toma valores comprendidos entre 1 y N, y por lo tanto L(1) es la longitud del primer elemento del vector I, L(2) es la longitud del segundo elemento del vector I, y así sucesivamente, donde L(N) es la longitud del N-ésimo (último) elemento del vector I.

Además, cada elemento I_1-I_N tiene su propio valor J(i), o J_1-J_N. Por ejemplo, si el elemento I_1-I_N es una fecha, el valor J(i) de este elemento es una fecha real, por ejemplo, en el formato de día/mes/año. Si el elemento I_1-I_N es el nombre del fabricante de software, el valor J(i) de dicho elemento es el nombre del fabricante, y así sucesivamente. Por lo tanto, para cada elemento I_1-I_N, el valor (i) tiene, indiferentemente, un valor numérico, alfanumérico o alfabético.

Dichos elementos I_1-I_N están dispuestos en una matriz 100 que tiene una pluralidad M de campos definidos por la intersección de una fila x (con x = x_1, x_2, …, x_R) con una columna respectiva y (con y = y_1, y_2, …, y_C). La matriz 100 se representa gráficamente en la figura 4. Cada uno de los N elementos I_1-I_N ocupa un campo respectivo (x, y) de la matriz 100. La matriz 100 tiene un número de campos igual a M = $x_R \cdot y_c$ y mayor que N. En particular, el valor de M se elige de modo que los elementos I_1-I_N se almacenen en campos respectivos (x, y) que no son directamente consecutivos uno a otro y de modo que soporten el algoritmo de encriptado elegido. En otros términos, el campo inmediatamente siguiente, a lo largo de las filas de la matriz 100, un campo conteniendo un elemento I_1-I_N, no contiene otro de los elementos I_1-I_N.

Cuando todos los elementos I_1-I_N han sido escritos en la matriz 100, los campos (x, y) de la matriz 100 que quedan libres se llenan con bytes aleatoriamente generados, por medio de un generador de números aleatorios de tipo conocido. De esta forma, cada elemento I_1-I_N está separado de los otros elementos I_1-I_N por uno o más bytes aleatorios.

En la figura 4, los bytes aleatorios se indican con $RB_{x,y}$, donde los subíndices x,y toman el valor de la fila correspondiente x y la columna correspondiente y del campo en que se escriben los bytes aleatorios. Por ejemplo, uno o más bytes aleatorios $RB_{1,1}$ están almacenados en el campo identificado por la primera fila $x = x_1$ y la primera columna $y = y_1$. Igualmente, uno o más bytes aleatorios $RB_{2,2}$ están almacenados en el campo identificado por la segunda fila $x = x_2$ y la segunda columna $y = y_2$.

Siempre con referencia a la figura 4, el elemento I_1 ocupa el campo correspondiente a la primera fila $x = x_1$ y la segunda columna $y = y_2$, y se indica como $I^1_{1,2}$; igualmente, el elemento I_2 ocupa el campo correspondiente a la segunda fila $x = x_2$ y la primera columna $y = y_1$, y se indica como $I^2_{2,1}$; y así sucesivamente.

Como se puede indicar en la figura 3, los bytes aleatorios $RB_{x,y}$ se insertan tanto entre los elementos I_1-I_N (cuando se consideran en secuencia a lo largo de las filas x_1-x_R) como antes del primer elemento I_1 de la matriz 100, así como después del último elemento I_N de la matriz 100. En otros términos, según esta realización, tanto el primer campo de la matriz 100 como el último campo de la matriz 100 contienen bytes aleatorios $RB_{x,y}$.

Es claro que la longitud en bytes de cada elemento I_1-I_N y de cada secuencia de bytes aleatorios $RB_{x,y}$ puede tener cualquier longitud, en bytes. Los campos $_{x,y}$ de la matriz 100 que son diferentes uno de otro pueden contener un número diferente de bytes. Esta situación se representa por medio de un ejemplo gráfico en la figura 5, donde los bytes aleatorios se representan gráficamente por campos llenos con líneas de sombreado y los elementos I_1-I_N por campos en blanco.

Según una realización, cada campo (x, y) de la matriz 100 es identificado de forma inequívoca por un número de identificación menor o igual a M, y viceversa. La asociación de un número de identificación con cada campo de la matriz 100 se implementa, por ejemplo, pasando progresivamente la matriz 100 a lo largo de las filas x, de modo que el número “1” corresponda al campo (x = 1, y = 1), el número “2” corresponda al campo (x = 1, y = 2), y así sucesivamente, de modo que el número “M” corresponde al campo ($x = x_R$, $y = y_C$). Es claro que también es posible cualquier otra asociación.

Volviendo a la figura 3, el paso 10 incluye definir un conjunto V_I de N valores enteros que identifican el orden recíproco en el que los elementos I_1-I_N deben insertarse en la matriz 100. El orden se considera, por ejemplo, comenzando en el primer campo de la matriz 100 y atravesando las filas x. Sin embargo, puede elegirse cualquier otro orden.

Como se representa en la figura 4, cada elemento I_1-I_N está separado de los otros elementos I_1-I_N por un campo (x, y) de la matriz 100 conteniendo una secuencia de bytes aleatorios $RB_{x,y}$. Para ello, en el paso 10 de la figura 3, se genera un conjunto V_{RB} de N+1 números aleatorios en el que cada uno de estos números identifica la longitud, en bytes, de una secuencia respectiva $RB_{x,y}$ que debe insertarse en la matriz 100 como un separador de los elementos I_1-I_N.

Por lo tanto, el conjunto (vector) V_I identifica el orden en que insertar los elementos I_1-I_N en la matriz 100 y el conjunto (vector) V_{RB} identifica la longitud de cada bloque $RB_{x,y}$ que debe insertarse entre un elemento I_1-IN-$_1$ y el elemento siguiente I_2-I_N (y también al principio y al final de la matriz 100). Los vectores V_I y V_{RB} se guardan en un vector de esquema de sello SS que tiene la forma siguiente (los valores numéricos indicados en el vector de esquema de sello SS son ilustrativos de una posible realización y no son limitativos):

$$SS=\{V_I;\ V_{RB}\}=\{1,2,3,4,\ldots;\ 23,4,46,5,11,\ldots\}.$$

De esta forma, en ausencia del vector de esquema de sello SS, la simple lectura de la matriz 100 no permite recuperar fácilmente la información (es decir, no es posible seguir inmediatamente el rastro del texto sencillo, o más bien los elementos I_1-I_N) contenidos en la matriz 100.

El vector V_I incluye N campos (y por lo tanto N valores en el caso donde cada campo contiene un solo valor), y el vector V_{RB} incluye N+1 campos (y en consecuencia N+1 valores en el caso donde cada campo contiene solamente un valor); por lo tanto, el vector de esquema de sello incluye 2N+1 valores, donde el valor de SS(N+1) es el número de bytes que precede al primer elemento I_1 en la matriz 100; el valor de SS(N+2) es el número de bytes entre el primer elemento I_1 y el segundo elemento I_2 en la matriz 100; el valor de SS(N+3) es el número de bytes entre el segundo elemento I_2 y el tercer elemento I_3 en la matriz 100; y así sucesivamente. El valor de SS(2N+1) es el número de bytes añadido en la matriz 100 después del último elemento I_N.

Como se ha mencionado previamente, el paso de definir el conjunto (vector) V_I es la equivalente de definir N números de identificación de un orden en el que insertar recíprocamente los elementos I_1-I_N en la matriz 100. Se

deberá indicar que los N valores que forman el vector V_I no identifican campos respectivos de la matriz 100, sino que representan el orden de introducción de los elementos I_1-I_N en la matriz 100, por ejemplo, cuando se consideran en secuencia a lo largo de las filas x de la matriz 100. De forma igual a la descrita, el vector V_I es, a modo de ejemplo, del tipo:

$$V_I=\{1,\ 2,\ 3,\ 4,\ \ldots\}.$$

El vector V_I se interpreta de la forma siguiente: el elemento de inserción I_1 como el primer elemento en la matriz 100 ($I(V_I(1)) = \{I_1\}$, porque $V_I(1) = \{1\}$ y $I(1) = I_1$); el elemento de inserción I_2 como el segundo elemento en la matriz 100 ($I(V_I(2)) = \{I_2\}$, porque $V_I(2) = \{2\}$ y $I(2) = I_2$); el elemento de inserción I_3 como el tercer elemento en la matriz 100 ($I(V_I(3)) = \{I_3\}$, porque $V_I(3) = \{3\}$ y $I(3) = I_3$); el elemento de inserción 14 como el cuarto elemento en la matriz 100 ($I(V_I(4)) = \{I_4\}$, porque $V_I(4) = \{4\}$ e $i(4) = I_4$); y así sucesivamente para todos los N elementos I_1-I_N del vector I.

Siempre como un ejemplo cualitativo, el paso de definir el conjunto de N+1 números aleatorios en el que cada uno de estos números identifica la longitud de un bloque respectivo $RB_{x,y}$ da lugar a la generación del vector V_{RB}, del tipo siguiente:

$$V_{RB}=\{23,\ 4,\ 46,\ 5,\ 11,\ \ldots\}.$$

El vector V_{RB} se interpreta de la siguiente manera: insertar veintitrés bytes aleatorios inmediatamente antes del primer elemento en la matriz 100, definida por el vector V_I (en este ejemplo, antes del elemento I_1); insertar cuatro bytes aleatorios inmediatamente después del primer elemento en la matriz 100, definida por el vector V_I (en este ejemplo, después del elemento I_1); insertar cuarenta y seis bytes aleatorios inmediatamente después del segundo elemento en la matriz 100, definida por vector el V_I (en este ejemplo, después del elemento I_2); insertar cinco bytes aleatorios inmediatamente después del tercer elemento en la matriz 100, definida por el vector V_I (en este ejemplo, después del elemento I_3); insertar once bytes aleatorios inmediatamente después del cuarto elemento en la matriz 100, definida por el vector V_I (en este ejemplo, después del elemento I4).

La operación de ordenación de los elementos I_1-I_N en la matriz 100 (es decir, generar el vector V_I, pasos 10 y 12 en la figura 3), y/o la generación del vector V_{RB}, puede realizarse cualquier número de veces, y permite la actualización cíclica de la matriz 100, a intervalos regulares y/o eventos especiales predeterminados siguientes. El vector de esquema de sello SS también es actualizado de manera correspondiente. En particular, dados N elementos I_1-I_N, estos tienen N! posibles permutaciones antes de obtener de nuevo el orden de inicio de la matriz 100. Esto quiere decir que la ordenación de los elementos I_1-I_N definidos por el vector V_I puede cambiarse N! veces antes de volver al orden inicial.

En la práctica, habiendo elegido dos primeros números enteros “A” y “B”, con valores entre 1 y M, se realiza una función shuffle de modo que las operaciones siguientes se realizan A veces:

(1) permutando la posición de un primer valor del vector V_I con la posición de un segundo valor del vector V_I (eligiéndose arbitrariamente las dos posiciones o usando dos contadores diferentes); y

(2) permutando la posición de uno o más valores del vector V_I con la posición de uno o más valores respectivos del vector V_I, estando estas posiciones después de la posición en la que se contenía originalmente el primer valor mencionado en el punto anterior (1).

El paso de permutación (1) se realiza “B” veces en “B” valores diferentes, con el fin de generar un vector V_I que contiene valores ordenados de una manera que no puede predeterminarse de antemano.

El valor de B se elige aleatoriamente y de tal manera que B no sea un divisor de $N!\cdot(N+1)!$. De esta forma, los dos números B y $N!\cdot(N+1)!$ son primos mutuos y, explotando el hecho de que no son perfectamente divisibles, se evita la vuelta de nuevo a los mismos valores después de un cierto número de repeticiones.

Con más detalle, la operación de permutación (1) incluye realizar una permutación en los valores numéricos contenidos en posiciones respectivas del vector V_I. Como se ha descrito previamente, el vector V_I es realmente un vector de números enteros, cada uno de los cuales describe en qué posición de la matriz 100 se inserta un elemento respectivo I_1-I_N (o, más bien, el valor J_1-J_N asociado con cada elemento respectivo I_1-I_N). Por lo tanto, el paso de permutación incluye los pasos de:

(a) - seleccionar un primer valor numérico contenido en un primer campo respectivo del vector V_I (por ejemplo, $V_I(1)$, conteniendo el número “1”);

(b) - seleccionar un segundo valor numérico contenido en un segundo campo respectivo, diferente del primer campo, del vector V_I;

(c) - permutar el primer valor numérico contenido en el primer campo y el segundo valor numérico contenido en el segundo campo uno con otro.

El paso de permutación también incluye el paso opcional de:

(d)- repetir los pasos (a)-(c).

Las figuras 6 y 7 muestran dichos pasos (a)-(d) con más detalle, por medio de diagramas de flujo.

Con referencia a la figura 6, en el paso 20, se supone que se está en la primera iteración del método de la figura 6. El Vector V_I es, por ejemplo, del tipo $V_I = \{1, 2, 3, 4\}$. La variable p es un índice que identifica, en este ejemplo, un campo del vector V_I y puede tomar valores de 1 a (N-1), en este caso (N-1) = 4. Se selecciona la penúltima posición del vector V_I, que es $V_I(p) = V_I(N-1) = V_I(3) = \{3\}$.

Entonces, en el paso 22, se selecciona el último campo del vector V_I, es decir, $V_I(p+1) = V_I(N) = V_I(4) = \{4\}$.

Posteriormente, en el paso 24, los valores contenidos en los campos $V_I(N-1)$ y $V_I(N)$ son comparados uno con otro. Si $V_I(N-1)$ es menor que $V_I(N)$, en el paso 26 se selecciona $V_I(N-1)$ como el primer elemento según dicho paso (a).

De otro modo, en el paso 28, el valor p se decrementa una unidad y se verifica, en el paso 29, si se ha llegado al inicio del vector V_I. Si el valor actual de p es igual a 0, se ha salido del vector V_I y se han hecho todas las permutaciones posibles (el método finaliza). Si p es mayor o igual a 1, en el paso 30 se selecciona $V_I(p)$; y, en el paso 32, se selecciona $V_I(p+1)$. El procesado vuelve entonces al paso 24 para comparar $V_I(p)$ con $V_I(p+1)$.

Continuando con el ejemplo anterior, dado el vector $V_I = \{1, 2, 3, 4\}$, el paso 24 conduce al paso 26, como $V_I(N-1)<V_I(N)$.

Los pasos 28-32 no se realizan y el valor $V_I(N-1) = 3$ se selecciona, así como el primer valor numérico de dicho paso (a).

El procesado pasa después a la selección del segundo valor numérico, según dicho paso (b). Con referencia a la figura 7, en el paso 36, el último valor del vector V_I (es decir, $V_I(N)$) se selecciona en la primera iteración del método de la figura 7. Según el método de la figura 7, el índice q se usa para indexar el vector V_I. Por lo tanto, en el paso 36, q = N.

Entonces, en el paso 38, el valor $V_I(p)$ seleccionado en el paso 26 en la figura 6 se compara con el valor actual $V_I(q)$. Si $V_I(q)>V_I(p)$, entonces, en el paso 40, el valor contenido en el campo $V_I(p)$ es intercambiado con el valor contenido en el campo $V_I(q)$. En este caso, considerando siempre el vector $V_I = \{1, 2, 3, 4)$, el paso 40 conduce a la generación del vector $V_I = \{1, 2, 4, 3\}$.

Si la comparación en el paso 38 pone de manifiesto $V_I(q)<V_I(p)$, entonces, en el paso 42, el valor q se decrementa una unidad. En el paso 43 se verifica si el valor actual de q es mayor que el valor actual de p. Si es verdadero, el procesado vuelve al paso 38 para una nueva comparación entre $V_I(p)$ y $V_I(q)$; si es falso, el método finaliza.

A continuación, el procesado pasa al paso de “permutación” previamente mencionado. Durante este paso, los valores numéricos contenidos en las N-p posiciones del vector V_I que siguen a la p'-ésima posición son subdivididos en dos subgrupos de (N-p)/2 valores cada uno. La posición de los dos subgrupos de (N-p)/2 valores se intercambia entonces. En el caso donde N-p es un número impar, es posible considerar N-p+1 campos, o elegir un grupo diferente.

Volviendo al vector V_I generado después del paso 40 en la figura 7, es decir, $V_I = \{1, 2, 4, 3\}$, el valor del índice p es igual a 3 y el paso de permutación no tiene efecto.

Iterando los pasos 20-32 en la figura 6 y la figura 7 de nuevo (segunda iteración), el nuevo vector V_I que se obtiene tiene la forma siguiente:

$V_I = \{1, 3, 4, 2\}$.

Entonces, la realización del paso de permutación da:

$V_I^{(2)} = \{1, 3, 2, 4\}$.

Iterando los pasos 20-32 en la figura 6 y la figura 7 de nuevo (tercera iteración), el vector V_I que se obtiene tiene la forma siguiente:

$V_I^{(3)} = \{1, 3, 4, 2\}$.

En este caso, el paso de permutación no cambia el vector V_I = {1, 3, 2, 4}.

Iterando los pasos 20-32 en la figura 6 y la figura 7 de nuevo (cuarta iteración), el vector V_I que se obtiene tiene la forma siguiente:

V_I = {1, 4, 3, 2}.

Entonces, la realización del paso de permutación da:

$V_I^{(4)}$ = {1, 4, 2, 3}.

El procesado continúa de esta forma, obteniendo todas las permutaciones posibles de los valores numéricos contenidos en el vector V_I.

Con el fin de hacer aleatorio el orden de los elementos I_1-I_N en la matriz 100 al tiempo de su primera generación, es posible realizar iteraciones de los pasos de los métodos de las figuras 6-7 y del paso de permutación “A” veces (donde A es un número aleatorio), con el fin de generar un orden de los valores numéricos en el vector V_I que no sea predeterminable de antemano.

Según otra realización, cada iteración de los pasos de los métodos de las figuras 6-7 se realiza “B” veces antes de realizar el paso de permutación.

El paso de definir el conjunto (vector) V_{RB} de N+1 números aleatorios, donde cada uno de estos números identifica la longitud en bytes de un bloque respectivo $RB_{x,y}$, cumple al menos uno de los requisitos siguientes:

(1) se inserta al menos un bloque $RB_{x,y}$ (de cualquier tamaño/longitud en bytes) antes del primer elemento I_1-I_N en la matriz 100;

(2) se inserta al menos un bloque $RB_{x,y}$ (de cualquier tamaño/longitud en bytes) entre dos elementos consecutivos I_1-I_N a lo largo de las filas x de la matriz 100, con el fin de separar los dos elementos consecutivos I_1-I_N uno de otro;

(3) se inserta al menos un bloque $RB_{x,y}$ (de cualquier tamaño/longitud en bytes) después del último elemento I_1-I_N en la matriz 100;

(4) la distribución de probabilidad de las longitudes en bytes de cada bloque $RV_{x,y}$ es uniforme; y

(5) la suma S_{RB} de las longitudes, en bytes, de todos los bloques $RB_{x,y}$ es tal que la matriz 100 se llena completamente (es decir, $\Sigma_i L(i)+S_{RB} = M$).

(6)

Según una realización de la presente invención, la longitud de un bloque respectivo $RB_{x,y}$ cumple todos los requisitos (1)-(5) previos.

Volviendo al diagrama de flujo de la figura 3, en el paso 14, la matriz 100 generada según los pasos 10 y 12 es encriptada usando una clave k_s generada aleatoriamente por el token de autenticación 1 en la figura 1. Dependiendo del algoritmo de encriptado elegido, la clave k_s puede generarse según un método estándar. En el caso descrito a modo de ejemplo, es el token el que maneja su generación. Así, el paso de encriptado usando la clave k_s invoca esta rutina estándar (y por lo tanto, en sí mismo, no es la materia de la presente invención). Por ejemplo, en el método criptográfico AES, la clave es tan larga como el bloque mínimo de datos encriptables (típicamente 32 bytes) y se genera a partir de un vector de inicialización que el usuario puede elegir o haber generado de forma aleatoria.

Además, la matriz 100 se encripta usando un método criptográfico de tipo conocido, por ejemplo, un algoritmo de encriptado asimétrico, tal como RSA, o un algoritmo de encriptado a base de bloques como AES (Encriptado Estándar Avanzado), o el algoritmo DES (Estándar de Encriptado de Datos), o cualquier otro algoritmo de encriptado conocido en la literatura. Dichos algoritmos de encriptado RSA, AES y DES son conocidos en la literatura y los pasos de encriptado asociados no se especifican aquí por razones de brevedad. La generación de las claves k_s también se realiza, como se ha indicado, con métodos conocidos en la literatura.

Una matriz encriptada (o, en general, un objeto texto encriptado), a continuación indicado como “C_data”, se obtiene como resultado del paso 14 de la figura 3.

Tanto la clave k_s como el objeto texto encriptado C_data están almacenados en la zona privada 4a de la memoria 4 del token de autenticación 1.

De forma similar a la descrita con referencia al encriptado de la matriz 100, el vector de esquema de sello SS, que representa la clave leída para la matriz 100, también está encriptado (paso 14) por medio de una clave respectiva k_v generada aleatoriamente por el token de autenticación 1 de la figura 1. La generación de la clave k_v tiene lugar de manera similar a la generación de la clave k_s, por medio de un método conocido de libre elección. Por ejemplo, con referencia de nuevo al algoritmo AES, esta clave k_v es generada por un vector de inicialización que es diferente del vector de inicialización usado para la clave k_s.

El vector SS también está encriptado con algoritmos conocidos, de forma similar a la matriz 100. Por ejemplo, podría usarse el algoritmo de encriptado AES. No obstante, es claro que pueden usarse otros algoritmos.

Tanto la clave k_v como el vector de esquema de sello encriptado SS (a continuación también indicado como "C_scheme") se guardan en la zona privada 4a de la memoria 4 del token de autenticación 1. El objeto texto encriptado C_data constituye un "sello" que contiene información (elementos I_1-I_N) cuyo conocimiento es una condición necesaria para permitir el acceso a más datos y/o información que se desea proteger. Por ejemplo, siguiendo haciendo referencia a la aplicación preferida de la presente invención, el desencriptado y la lectura de este sello es un requisito para autorizar la ejecución de un programa de software, u obtener una licencia de uso de este programa de software. Como se ha indicado previamente, una copia del vector I que contiene elementos I_1-I_N se guarda en el programa de software de una manera que no es accesible a un usuario del programa de software. La verificación de la lectura correcta de la matriz 100 (usando el esquema de sello SS) se realiza comparando los elementos I_1-I_N leídos en la matriz 100 con los elementos I_1-I_N de vector I guardados en el programa de software.

Para autorizar el acceso a estos datos/información/programa de software adicionales, se realizan los pasos de un método según la figura 8.

Ante todo, en el paso 50, se realiza una comprobación para establecer si el token de autenticación 1 usado corresponde al token de autenticación esperado 1.

Esta comprobación puede realizarse verificando que el modelo y/o el número de serie del token de autenticación 1 en uso son consistentes y compatibles con los esperados.

En el caso donde la comprobación en el paso 50 proporciona un resultado positivo, entonces, en el paso 52, se accede a la zona privada 4a de la memoria 4 para verificar si el sello es integral o ha sido alterado. El acceso a la zona privada 4a de la memoria 4 tiene lugar, como se ha indicado, con un PIN que se genera en el paso de inicialización de licencia y consta de un conjunto de caracteres imprimibles (símbolos más alfanuméricos) de longitud suficiente (por ejemplo, 127 caracteres) para hacer inefectivo o especialmente difícil cualquier intento de ataque. De hecho, en caso de intentar averiguar este PIN por intentos sucesivos, si se supera un número predeterminado de intentos fallidos, el token se bloquea de manera irrecuperable, obligando a contactar con el fabricante de software, que es el único que puede desbloquearlo.

Según una realización, se llevan a cabo los pasos siguientes: (a) cuando el usuario intenta iniciar el software a proteger, éste último indica al usuario que introduzca el token y una contraseña personal; (b) el usuario proporciona su contraseña personal pedida en el punto (a) anterior; (c) el software a proteger comprueba, con su propio método que no es parte de la presente invención, que la contraseña personal del usuario es correcta (por ejemplo, en el caso donde se usa una función hash, calcula la hash y la comprueba con una hash previamente almacenada, por ejemplo en un archivo, en una base de datos, en un servidor, etc); (d) si se pasa la comprobación en el punto (c) anterior, el software a proteger usa el PIN correspondiente a la licencia del usuario para abrir las comunicaciones con el token y realizar las comprobaciones relevantes según los pasos del método de la presente invención.

Entonces, en el paso 54, usando las claves k_s y k_v, el objeto texto encriptado C_data y el esquema de sello encriptado SS C_scheme son desencriptados, de manera conocida usando el algoritmo de encriptado usado según los pasos 14 y 16 de la figura 3. Entonces se verifica la coherencia entre las posiciones de los elementos I_1-I_N en la matriz 100 identificados por el esquema de sello SS y las posiciones efectivas de los elementos I_1-I_N en la matriz 100.

En una primera comprobación, leyendo las posiciones especificadas en el esquema de sello SS, se verifica que el contenido leído de la matriz 100 es consistente con los elementos I_1-I_N del vector I contenido en el programa de software que se desea proteger, por ejemplo, verificando uno o varios parámetros siguientes: (i) cardinalidad de campo; (ii) longitud de campo; (iii) tipo de campo (número, cadena, fecha, etc); (iv) contenido de campos fijos (por ejemplo, nombre del fabricante de software); (v) coherencia entre software y el nombre de software especificado en el sello; (vi) coherencia entre información relativa al hardware que contiene el sello (por ejemplo, el número de serie del chip token), obtenida interrogando el hardware, y los datos equivalentes grabados en el sello. En particular, este último punto (vi) implica que, en el caso donde una persona mal intencionada logra copiar totalmente (sin destruir el chip al mismo tiempo) el contenido de zona de memoria privada 4a a otro token del mismo modelo, todavía se denegaría el acceso a los datos, porque están unidos a dicho elemento concreto de hardware (token).

En este paso, el método también permite la lectura de la información variable. A modo de ejemplo no limitativo, esta información incluye uno o más de los siguientes: número de licencia; nombre de cliente; fecha/hora de generación del sello actual; módulos activados por la licencia (si es aplicable).

En el caso donde se deniega el acceso a los datos/información a proteger, paso 55, el método de la figura 8 finaliza.

En el caso donde se permite el acceso a los datos/información a proteger, en el paso 56, se genera aleatoriamente un nuevo vector de esquema de sello SS y, en base al nuevo vector de esquema de sello SS así generado, se genera una nueva matriz similar a la matriz 100. Según una realización, la nueva matriz contiene los mismos elementos I_1-I_N de la matriz 100, dispuestos según un nuevo orden y separados por bloques $RB_{x,y}$ que tienen diferentes números de bytes. Alternativamente, según una realización diferente, la nueva matriz contiene elementos $I_1^{(a)}$-$I_N^{(a)}$ diferentes de los elementos I_1-I_N de matriz 100; el número N' de elementos $I_1^{(a)}$-$I_N^{(a)}$ también puede ser diferente del número N de elementos I_1-I_N. Según otra realización, la nueva matriz contiene elementos $I_1^{(a)}$-$I_N^{(a)}$ que son parcialmente comunes a los elementos I_1-I_N de la matriz 100 (por ejemplo, los campos fijos son los mismos) y parcialmente diferentes (por ejemplo, los campos variables podrían ser diferentes). De esta forma, se genera un nuevo sello.

Los campos de datos usados para generar el nuevo sello son los mismos del sello original, como ya se ha descrito anteriormente. Los datos de fecha/hora los proporciona el dispositivo informático local en el que se ejecuta el software durante la generación del nuevo sello. El esquema se hace evolucionar a partir de la permutación actual en la que se encuentra. Los espacios reservados para los bytes aleatorios (RB_{xy}) son reasignados de manera aleatoria (usando el mismo criterio previamente descrito).

Cada vez que el sello se hace evolucionar según el paso 56 de la figura 8, también se realizan los pasos de encriptado según los pasos 14-16 de la figura 3, no descritos aquí por razones de brevedad.

Entonces, después de un cierto período de tiempo predeterminado o aleatorio, la coherencia entre el esquema de sello actual SS y la matriz actual se verifica de nuevo, como se ha descrito con referencia al paso 54.

Los pasos 54-56 siguen iterando hasta que el paso 54 proporciona un resultado que se considera aceptable, es decir, de tal manera que la disposición de los elementos $I_1^{(a)}$-$I_N^{(a)}$ en la matriz actual es consistente con la disposición identificada por el esquema de sello actual SS, y los tamaños en bytes de los bloques $RB_{x,y}$ son consistentes con los tamaños especificados por el esquema de sello actual SS.

Cuando el paso 54 proporciona un resultado positivo como salida (correspondencia entre el vector SS y la matriz 100), entonces, el token usado se considera válido y se obtiene autorización para acceder a los datos y/o información posteriores (por ejemplo, autorización para ejecutar un programa de software, u obtener una licencia de uso de este programa de software).

A partir de este momento en adelante y durante la sesión de uso actual del software, solamente se verifica que el hardware de token sigue respondiendo con su propia identidad, que ha sido previamente confirmada.

No obstante, la verificación en profundidad previamente descrita puede ser invocada de nuevo por el software en base al evento de que el usuario pida usar funciones que se consideren críticas para usar el software (o las que el fabricante considere de mayor valor).

En el caso de que el token se desconecte, se detecta esta situación y hay que volver a insertar el mismo token (por “mismo” se entiende el que tiene el número de serie concreto), con la reinicialización de todo el procedimiento de verificación una vez insertado.

De otro modo, si el token no se inserta o tiene un número de serie diferente, el software suspende su ejecución (lo que quiere decir que, una vez iniciada una sesión con un token, no es posible continuar con otro token, aunque este otro token contenga una licencia válida).

Las ventajas del sistema y método según la presente invención son evidentes.

Para la opción de ordenación en el vector conteniendo los elementos de información I_1-I_N, se explota el hecho (matemáticamente probable) de que existe una y solamente una forma de pasar por todas las permutaciones posibles de los N elementos permutando las posiciones de un par a un tiempo, según el concepto de las denominadas permutaciones lexicales (obtenible según el algoritmo de Teinhaus-Johnson-Trotter). Este método es computacionalmente eficiente y forma un recorrido hamiltoniano, es decir, visita todos los vértices de un hipotético permutaedro, pasando a través de cada vértice solamente una vez. En este caso, el permutaedro es un hipotético sólido en el que cada vértice contiene una posible combinación de los elementos de información I_1-I_N diferente de las otras combinaciones presentes en los vértices restantes.

La forma de recorrer esta secuencia se elige según un criterio ad hoc: de hecho, saltando a lo largo de la secuencia un número de pasos elegido de modo que no sea un divisor entero del número de elementos (que es N! (N+1)!), se garantiza que la secuencia original se genere en un único ciclo hamiltoniano. Esencialmente, la secuencia es un bucle acíclico. De esta forma, se define un sistema denominado de prueba de conocimiento, donde el token es el soporte del "conocimiento" (o, en otros términos, del sello).

El componente de software está configurado para comprobar la sanidad del sello, que demuestra la veracidad de la licencia (y por lo tanto sirve como "prueba" o "testimonio") presente en un token genérico válido que contiene los datos/información de prueba.

Así, tomando en consideración lo anterior, y que:

- el sistema criptográfico incluye las funciones de inicialización del primer sello, apertura del sello, y validación del sello;

- el sello cambia de forma cada vez según un esquema aleatorio que no altera el contenido de la licencia, sino que cambia la disposición con respecto al material aleatorio del resto del mensaje;

- se cumple el criterio de plenitud, es decir, es exitosa la verificación de cada sello inicial y posterior, sea cual sea el esquema adoptado para la licencia;

- se cumple el criterio de sanidad perfecta, puesto que la probabilidad de aceptar el sello es nula si el sello no tiene la forma correcta (por ejemplo, si se ha falsificado entre dos comprobaciones sucesivas), y así cada verificación de este sistema es del tipo de prueba no interactiva de conocimiento cero y se deberá especificar que usa un esquema de compromiso evolutivo. En conclusión, aparte del obvio aumento de seguridad, las ventajas obtenidas son:

(i) aunque requiere la transmisión de datos, la seguridad de la información de licencia no se pone en peligro en el intercambio de datos entre el software y el hardware necesario para validar la licencia (porque no se revela); y

(ii) al mismo tiempo, se mantiene la posibilidad de gestionar múltiples licencias flotantes en múltiples máquinas con un mismo software (es decir, la situación más general donde hay múltiples máquinas y múltiples usuarios y cada usuario puede trabajar en la máquina que prefiera).

Con respecto al uso del PIN, a la primera inicialización, el código en cuestión es generado automáticamente de manera que sea muy largo (por ejemplo, 128 caracteres) y no se describe al usuario, sino que es utilizado de forma automática y transparente a nivel bajo por el software. De esta forma, teniendo en mente que el token puede estar configurado para bloquearse automáticamente después de un número predeterminado de intentos fallidos (por ejemplo 3 o 4), se minimiza la probabilidad de acceso a la zona privada del token por un usuario no autorizado que desee acceder al contenido.

Además, según la presente invención, se ofrecen varios niveles de protección:

(1) un nivel físico, que requiere acceso al token de autenticación 1, y es tal que el chip es a prueba de manipulación, o, más bien, si se intenta extraer físicamente el chip, es prácticamente imposible recuperar la información guardada en él, que se pierde/destruye;

(2) una zona de memoria privada 4a, que está oculta al usuario y solamente es accesible mediante la contraseña establecida por el fabricante del software;

(3) un nivel criptográfico, que permite más protección de la información en el caso de que la zona privada 4a de la memoria sea descubierta y/o se conozca su contraseña de acceso;

(4) un esquema de disposición recíproca de los elementos I_1-I_N de tal manera que la simple lectura de este esquema o de la matriz generada a partir de este esquema no permite recuperar fácilmente los datos dispuestos según este esquema;

(5) un mecanismo de verificación de coherencia de tal manera que los intentos de acceso o desencriptado que producen cambios incluso mínimos en los datos almacenados en la matriz disparan la invalidación del sello.

(6) la posibilidad de instalar el software en múltiples máquinas e implementar una licencia flotante que solamente habilita las máquinas en las que se inserta o insertan el o los tokens conteniendo la licencia o las licencias. Por ejemplo, instalación en 10 máquinas, pero con 3 tokens solamente.

Finalmente, es claro que se puede hacer modificaciones y variantes en la invención descrita e ilustrada aquí sin apartarse del alcance de protección de la presente invención, definido en las reivindicaciones anexas.

Por ejemplo, la zona privada 4a de la memoria 4 podría acomodar una pluralidad de sellos generados como se ha descrito previamente, con el fin de gestionar simultáneamente una pluralidad de software y usuarios.

REIVINDICACIONES

1. Un sistema de autenticación para autorizar un acceso a información a proteger, incluyendo un token de hardware (1) que tiene una memoria (4) que incluye una zona de memoria privada (4a), accesible por medio de un número de identificación personal y que almacena:

- una pluralidad de primeros elementos de información (I_1-I_N) de tipo numérico y/o alfanumérico y/o alfabético; y

- una pluralidad de primeros elementos de datos aleatorios ($RB_{x,y}$) de tipo numérico y/o alfanumérico y/o alfabético, teniendo cada uno un tamaño respectivo, donde:

-- dichos primeros elementos de información (I_1-I_N) y dichos primeros elementos de datos aleatorios ($RB_{x,y}$) están dispuestos en una matriz (100), que tiene al menos un orden de lectura y de tal manera que cada primer elemento de información (I_1-I_N) esté separado de un primer elemento de información sucesivo (I_1-I_N), en dicho orden de lectura, por uno de dichos primeros elementos de datos aleatorios ($RB_{x,y}$), y

-- la zona de memoria privada (4a) almacena además un primer vector de esquema de sello (SS), de tipo encriptado, conteniendo primera información de disposición relativa de los primeros elementos de información (I_1-I_N) en la matriz (100) y primera información de tamaño de cada primer elemento de datos aleatorios ($RB_{x,y}$), con el fin de permitir el acceso a cada uno de los primeros elementos de información (I_1-I_N) en la matriz (100),

estando configurado además dicho sistema de autenticación para:

- supeditar el acceso a la zona de memoria privada (4a) a la introducción de dicho número de identificación personal;

- desencriptar el primer vector de esquema de sello y adquirir dicha primera información de disposición relativa de los primeros elementos de información (I_1-I_N) y dicha primera información de tamaño de cada primer elemento de datos aleatorios ($RB_{x,y}$) en la matriz (100);

- comprobar la correspondencia entre dicha primera información de disposición relativa adquirida y la disposición de los primeros elementos de información (I_1-I_N) en la matriz (100), y entre dicha primera información de tamaño adquirida y el tamaño de los primeros elementos de datos aleatorios en la matriz (100); y

- autorizar o denegar el acceso a dichos elementos de información a proteger en base a un resultado de dicha comprobación de correspondencia entre dicha primera información de disposición relativa adquirida y la disposición de los primeros elementos de información (I_1-I_N) en la matriz (100), y entre dicha primera información de tamaño adquirida y los tamaños respectivos de dichos primeros elementos de datos aleatorios ($RB_{x,y}$),

-- donde la primera información de disposición relativa de los primeros elementos de información (I_1-I_N) sigue una a otra en un orden aleatorio en dicho primer vector de esquema de sello (SS), obteniéndose dicho orden aleatorio por medio de un número aleatorio de permutaciones de dicha primera información de disposición relativa de los primeros elementos de información (I_1-I_N), y donde la realización de dichas permutaciones incluye:

i) permutar la posición de un elemento de información inicial elegido entre los primeros elementos de información (I_1-I_N) con la posición de otro de los primeros elementos de información (I_1-I_N);

ii) permutar la posición de dos o más elementos de información elegidos entre dichos primeros elementos de información (I_1-I_N), estando estas posiciones después de la posición en la matriz (100) en la que dicho elemento de información inicial fue asignado anteriormente a la operación de permutación i), y siendo el número de primeros elementos de información (I_1-I_N) igual a N, y realizándose la operación de permutación i) un número de veces B elegido aleatoriamente y de tal manera que B no sea un divisor de N!·(N+1)!.

2. Un sistema de autenticación según la reivindicación 1, configurado además, en el caso de que dicha autorización sea denegada, para:

- generar un segundo vector de esquema de sello, de tipo encriptado, conteniendo segunda información de disposición relativa de los primeros elementos de información (I_1-I_N) y segunda información de tamaño de segundos elementos de datos aleatorios ($RB_{x,y}$);

- disponer los primeros elementos de información (I_1-I_N) en la matriz (100) en base a la segunda información de disposición relativa y de tal manera que cada primer elemento de información (I_1-I_N) esté separado de un primer elemento de información sucesivo (I_1-I_N), en dicho orden de lectura, por un segundo elemento de datos aleatorios respectivo ($RB_{x,y}$) que tiene un tamaño correspondiente a dicha segunda información de tamaño;

- supeditar otra concesión de dicha autorización de acceso a dicha información a proteger a: desencriptar el primer vector de esquema de sello y adquirir la primera información de disposición relativa y la primera información de

tamaño; y verificar la correspondencia entre la primera información de disposición relativa adquirida y la disposición de los primeros elementos de información (I_1-I_N) en la matriz (100), y entre la primera información de tamaño adquirida y el tamaño de los primeros elementos de datos aleatorios en la matriz (100).

3. Un sistema de autenticación según la reivindicación 1, configurado además, en el caso de que dicha autorización sea denegada, para:

- adquirir segundos elementos de información (I_1-I_N), teniendo cada uno un valor numérico, alfanumérico, o alfabético respectivo (J_1-J_N);

- generar una pluralidad de segundos elementos de datos aleatorios ($RB_{x,y}$), teniendo cada uno un valor numérico, alfanumérico, o alfabético respectivo, y un tamaño respectivo;

- generar un segundo vector de esquema de sello (SS), de tipo encriptado, conteniendo segunda información de disposición relativa de los segundos elementos de información (I_1-I_N) y segunda información de tamaño de los segundos elementos de datos aleatorios ($RB_{x,y}$);

- disponer los segundos elementos de información (I_1-I_N) en la matriz (100) en base a la segunda información de disposición relativa y de tal manera que cada segundo elemento de información (I_1-I_N) esté separado de un segundo elemento de información sucesivo (I_1-I_N), en dicho orden de lectura, por unos segundos elementos de datos aleatorios respectivos ($RB_{x,y}$) que tiene un tamaño correspondiente a dicha segunda información de tamaño; y

- supeditar otra concesión de dicha autorización de acceso a dicha información a proteger a: desencriptar el segundo vector de esquema de sello y adquirir la segunda información de disposición relativa y la segunda información de tamaño; y verificar la correspondencia entre la segunda información de disposición relativa adquirida y la disposición de los segundos elementos de información (I_1-I_N) en la matriz (100), y entre la segunda información de tamaño adquirida y el tamaño de los segundos elementos de datos aleatorios en la matriz (100).

4. Un sistema de autenticación según cualquiera de las reivindicaciones precedentes, donde dichos primeros elementos de información (I_1-I_N) incluyen datos de tipo fijo y/o datos de tipo variable.

5. Un sistema de autenticación según cualquiera de las reivindicaciones precedentes, donde las operaciones de permutación i) e ii) se repiten un número de veces igual a M, donde M es un valor entero mayor que N.

6. Un sistema de autenticación según cualquiera de las reivindicaciones precedentes, configurado para realizar una o más permutaciones de la primera información de disposición relativa de los primeros elementos de información (I_1-I_N) del primer vector de esquema de sello (SS) con el fin de generar el segundo vector de esquema de sello.

7. Un sistema de autenticación según cualquiera de las reivindicaciones precedentes, configurado además para encriptar la matriz (100) usando una primera clave de encriptado (k_s) generada por un algoritmo de encriptado.

8. Un sistema de autenticación según la reivindicación 7, donde la primera clave de encriptado (k_s) se almacena en la zona de memoria privada (4a) de la memoria (4).

9. Un sistema de autenticación según cualquiera de las reivindicaciones precedentes, donde los vectores de esquema de sello primero y segundo (SS) son encriptados por medio de una segunda clave de encriptado respectiva (k_y) generada por un algoritmo de encriptado.

10. Un sistema de autenticación según la reivindicación 9, donde las segundas claves de encriptado (k_v) están almacenadas en la zona de memoria privada (4a) de la memoria (4).

11. Un sistema de autenticación según cualquiera de las reivindicaciones precedentes, configurado además para comprobar si el token de hardware (1) corresponde a un token de hardware esperado.

12. Un sistema de autenticación según cualquiera de las reivindicaciones precedentes, donde verificar la correspondencia entre dicha primera información de disposición relativa y la disposición de los primeros elementos de información (I_1-I_N) en la matriz (100) incluye verificar al menos uno de: la coherencia de un valor (J_1J_N) de los primeros elementos de información (I_1-I_N) en la matriz (100) con un valor esperado respectivo; y una correspondencia entre un tamaño de un elemento de información respectivo (I_1-I_N) y un tamaño esperado para dicho primer elemento de información (Il-IN).

13. Un sistema de autenticación según cualquiera de las reivindicaciones precedentes, configurado además para supeditar el acceso a la zona de memoria privada (4a) a verificar si un número de serie del token de hardware corresponde a un número de serie esperado.

14. Un método de autenticación para autorizar un acceso a información a proteger, incluyendo:

- adquirir una pluralidad de primeros elementos de información (I_1-I_N) de tipo numérico y/o alfanumérico y/o alfabético;

- generar una pluralidad de primeros elementos de datos aleatorios ($RB_{x,y}$) de tipo numérico y/o alfanumérico y/o alfabético, teniendo cada uno un tamaño respectivo;

- disponer dichos primeros elementos de información (I_1-I_N) y dichos primeros elementos de datos aleatorios ($RB_{x,y}$) en una matriz (100), que tiene al menos un orden de lectura y de tal manera que cada primer elemento de información (I_1-I_N) esté separado de un primer elemento de información sucesivo (I_1-I_N), en dicho orden de lectura, por uno de dichos primeros elementos de datos aleatorios ($RB_{x,y}$);

- generar un primer vector de esquema de sello (SS), de tipo encriptado, conteniendo primera información de disposición relativa de los primeros elementos de información (I_1-I_N) en la matriz (100) y primera información de tamaño de cada primer elemento de datos aleatorios ($RB_{x,y}$), con el fin de permitir el acceso a cada uno de los primeros elementos de información (I_1-I_N) en la matriz (100);

- supeditar la concesión de una autorización de acceso a dicha información a proteger a: (i) introducir un número de identificación personal; (ii) desencriptar el primer vector de esquema de sello y adquirir la primera información de disposición relativa y la primera información de tamaño; y (iii) verificar la correspondencia entre la primera información de disposición relativa adquirida y la disposición de los primeros elementos de información (I_1-I_N) en la matriz (100), y entre la primera información de tamaño adquirida y el tamaño de los primeros elementos de datos aleatorios en la matriz (100); y

- conceder dicha autorización en el caso donde dicha comprobación de correspondencia es positiva;

-- donde generar el primer vector de esquema de sello (SS) incluye disponer la primera información de disposición relativa de los primeros elementos de información (I_1-I_N) en un orden aleatorio que incluye realizar un número aleatorio de permutaciones en dicha primera información de disposición relativa de los primeros elementos de información (I_1-I_N), y

-- donde realizar dichas permutaciones incluye:

i) permutar la posición de un elemento de información inicial elegido entre los primeros elementos de información (I_1-I_N) con la posición de otro de los primeros elementos de información (I_1-I_N);

ii) permutar la posición de dos o más elementos de información elegidos entre dichos primeros elementos de información (I_1-I_N), estando estas posiciones después de la posición en la matriz (100) en la que dicho elemento de información inicial fue asignado anteriormente al paso de permutación i), y siendo el número de primeros elementos de información (I_1-I_N) igual a N y realizándose el paso de permutación i) un número de veces B que se elige aleatoriamente y de tal manera que B no sea un divisor de $N!\cdot(N+1)!$.

15. Un método de autenticación según la reivindicación 14, incluyendo además, en el caso donde dicha autorización es denegada:

- generar un segundo vector de esquema de sello, de tipo encriptado, conteniendo segunda información de disposición relativa de los primeros elementos de información (I_1-I_N) y segunda información de tamaño de los segundos elementos de datos aleatorios ($RB_{x,y}$);

- disponer los primeros elementos de información (I_1-I_N) en la matriz (100) en base a la segunda información de disposición relativa, y de tal manera que cada primer elemento de información (I_1-I_N) esté separado de un primer elemento de información sucesivo (I_1-I_N), en dicho orden de lectura, por un segundo elemento de datos aleatorios respectivo ($RB_{x,y}$) que tiene un tamaño correspondiente a dicha segunda información de tamaño;

- supeditar otra concesión de dicha autorización de acceso a dicha información a proteger a: desencriptar el primer vector de esquema de sello y adquirir la primera información de disposición relativa y la primera información de tamaño; y verificar la correspondencia entre la primera información de disposición relativa adquirida y la disposición de los primeros elementos de información (I_1-I_N) en la matriz (100), y entre la primera información de tamaño adquirida y el tamaño de los primeros elementos de datos aleatorios en la matriz (100).

16. Un método de autenticación según la reivindicación 14, incluyendo además, en el caso donde dicha autorización es denegada:

- adquirir segundos elementos de información (I_1-I_N) de tipo numérico y/o alfanumérico y/o alfabético;

- generar una pluralidad de segundos elementos de datos aleatorios ($RB_{x,y}$) de tipo numérico y/o alfanumérico y/o alfabético, teniendo cada uno un tamaño respectivo;

- generar un segundo vector de esquema de sello, de tipo encriptado, conteniendo segunda información de disposición relativa de los segundos elementos de información (I_1-I_N) y segunda información de tamaño de los segundos elementos de datos aleatorios ($RB_{x,y}$);

- disponer los segundos elementos de información (I_1-I_N) en la matriz (100) en base a la segunda información de disposición relativa y de tal manera que cada segundo elemento de información (I_1-I_N) esté separado de un segundo elemento de información sucesivo (I_1-I_N), en dicho orden de lectura, por un segundo elemento de datos aleatorios respectivo ($RB_{x,y}$) que tiene un tamaño correspondiente a dicha segunda información de tamaño; y

- supeditar otra concesión de dicha autorización de acceso a dicha información a proteger a: desencriptar el segundo vector de esquema de sello y adquirir la segunda información de disposición relativa y la segunda información de tamaño; y verificar la correspondencia entre la segunda información de disposición relativa adquirida y la disposición de los segundos elementos de información (I_1-I_N) en la matriz (100), y entre la segunda información de tamaño adquirida y el tamaño de los segundos elementos de datos aleatorios en la matriz (100).

17. Un método de autenticación según cualquiera de las reivindicaciones 14-16, donde adquirir la pluralidad de primeros elementos de información (I_1-I_N) incluye adquirir datos de tipo fijo y/o datos de tipo variable.

18. Un método de autenticación según cualquiera de las reivindicaciones 14-17, donde los pasos de permutación i) y ii) se repiten un número de veces igual a M, donde M es un valor entero mayor que N.

19. Un método de autenticación según cualquiera de las reivindicaciones 14-18, incluyendo además realizar un número aleatorio de permutaciones de la primera información de disposición relativa de los primeros elementos de información (I_1-I_N) del primer vector de esquema de sello (SS) con el fin de generar una ordenación aleatoria de dicha primera información de disposición relativa.

20. Un método de autenticación según cualquiera de las reivindicaciones 14-19, incluyendo además encriptar la matriz (100) usando una primera clave de encriptado (k_s) generada por un algoritmo de encriptado.

21. Un método de autenticación según cualquiera de las reivindicaciones 14-20, donde el primer vector de esquema de sello (SS) es encriptado con una segunda clave de encriptado (K_v) generada por un algoritmo de encriptado.

22. Un método de autenticación según la reivindicación 15 o 16, donde el segundo vector de esquema de sello (SS) es encriptado con una tercera clave de encriptado (k_v) generada por un algoritmo de encriptado.

23. Un método de autenticación según cualquiera de las reivindicaciones 14-22, donde verificar la correspondencia entre dicha primera información de disposición relativa adquirida y la disposición de los primeros elementos de información (I_1-I_N) en la matriz (100) incluye verificar al menos uno de: la coherencia de un valor (J_1-J_N) de los primeros elementos de información (I_1-I_N) en la matriz (100) con un valor esperado respectivo; y una correspondencia entre un tamaño de un primer elemento de información respectivo (I_1-I_N) y un tamaño esperado para dicho primer elemento de información (I_1-I_N).

24. Un producto de programa de ordenador incluyendo instrucciones de programa de ordenador cargables en medios de procesado (1, 6) y diseñadas de modo que, cuando sean ejecutadas, los medios de procesado estén configurados para realizar todos los pasos del método según cualquiera de las reivindicaciones 14-23.

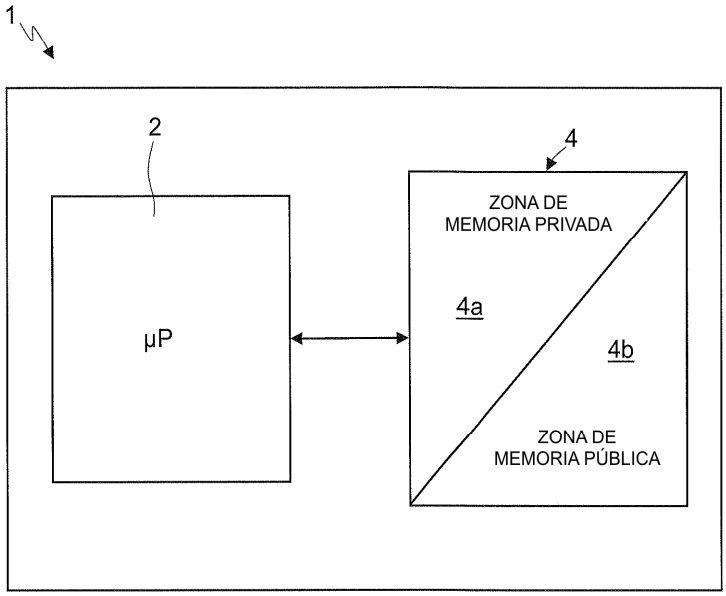
1
2
4
ZONA DE
MEMORIA PRIVADA
4a
4b
ZONA DE
MEMORIA PÚBLICA
µP

Fig.1

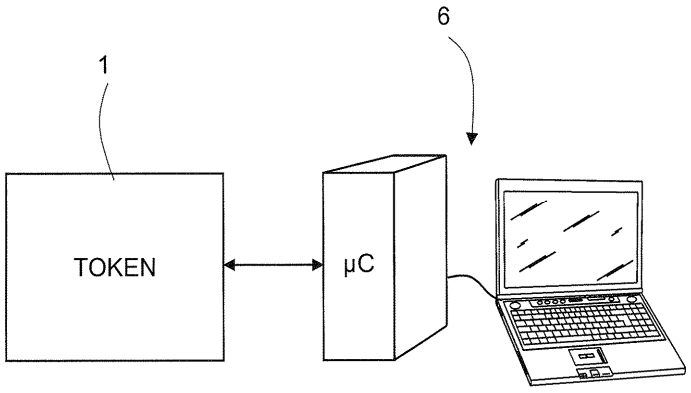
6
1
TOKEN
µC

Fig.2

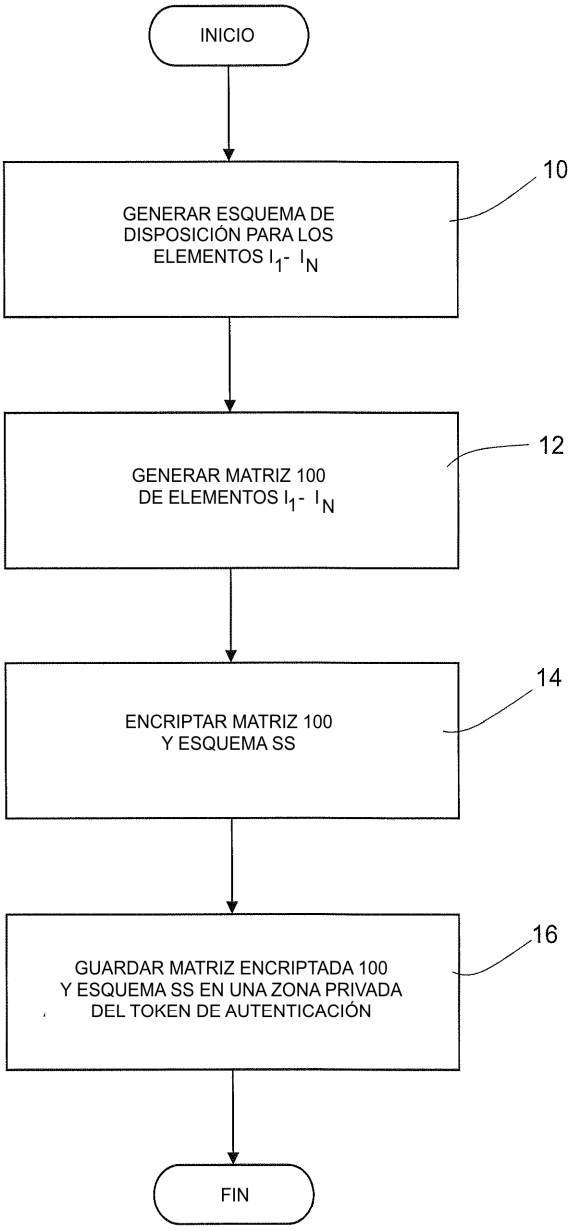
INICIO
GENERAR ESQUEMA DE DISPOSICIÓN PARA LOS ELEMENTOS I_1- I_N
10
GENERAR MATRIZ 100 DE ELEMENTOS I_1- I_N
12
ENCRIPTAR MATRIZ 100 Y ESQUEMA SS
14
GUARDAR MATRIZ ENCRIPTADA 100 Y ESQUEMA SS EN UNA ZONA PRIVADA DEL TOKEN DE AUTENTICACIÓN
16
FIN

Fig.3

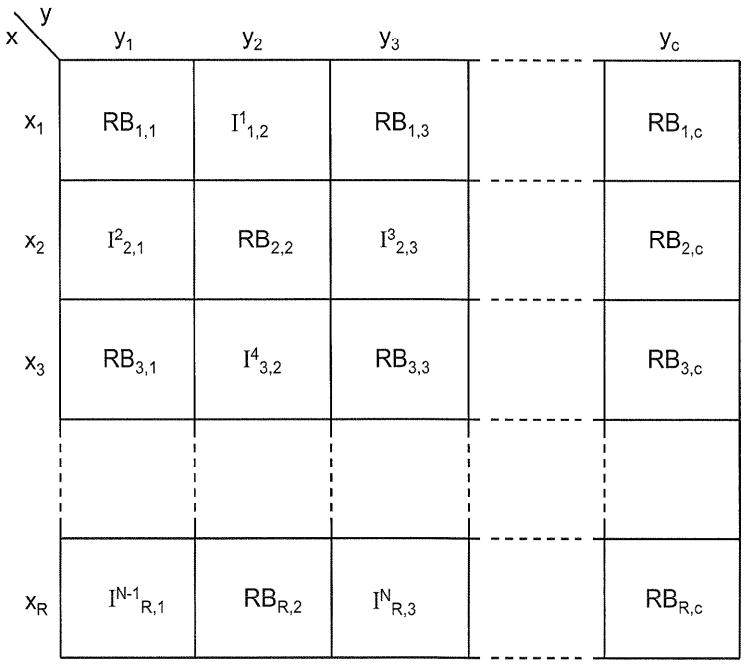
y
x
y1
y2
y3
yc
x1
RB1,1
I1 1,2
RB1,3
RB1,c
x2
I2 2,1
RB2,2
I3 2,3
RB2,c
x3
RB3,1
I4 3,2
RB3,3
RB3,c
xR
IN-1 R,1
RBR,2
IN R,3
RBR,c

Fig.4

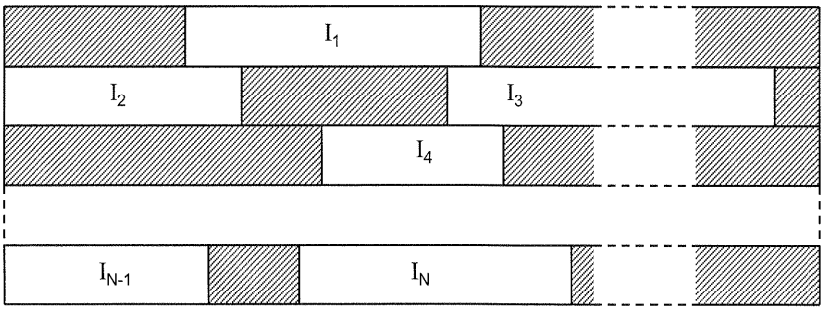
I1
I2
I3
I4
IN-1
IN

Fig.5

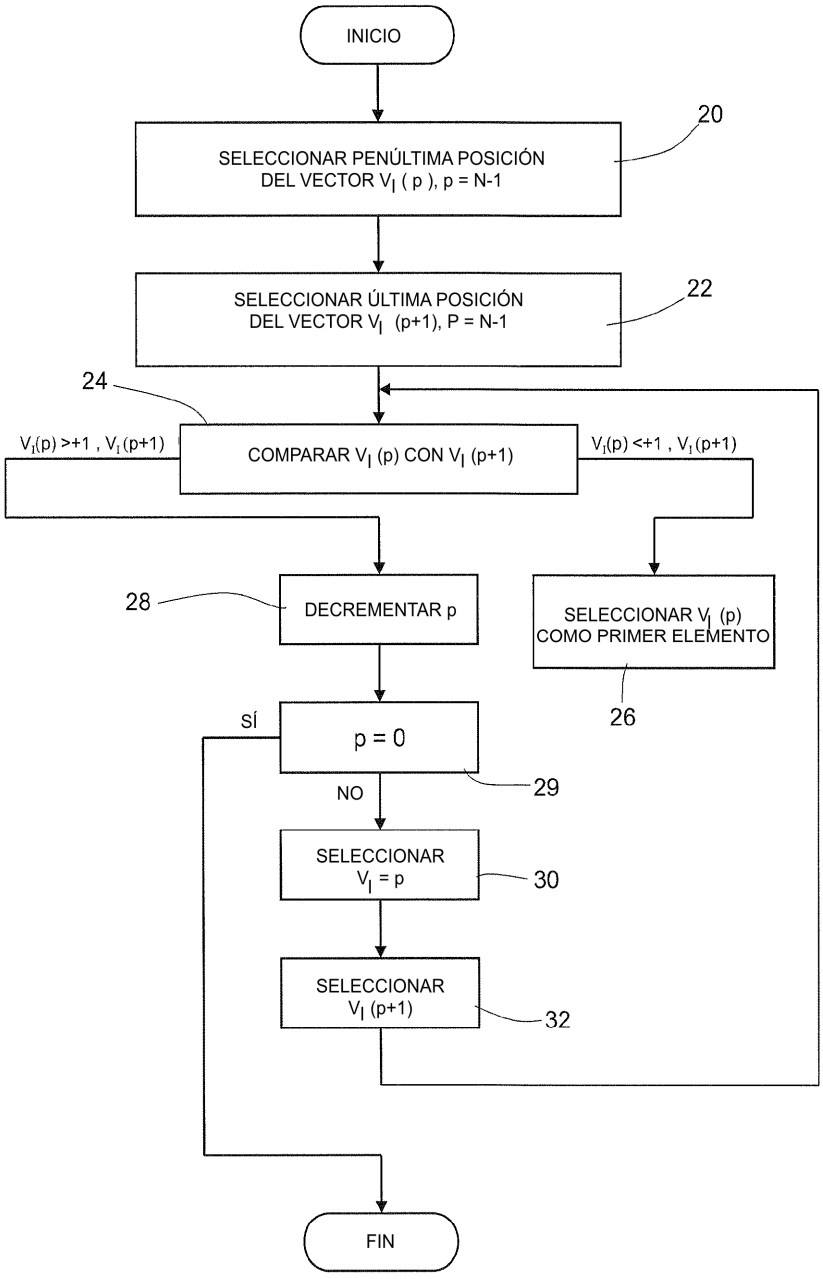
INICIO
SELECCIONAR PENÚLTIMA POSICIÓN DEL VECTOR V_I (p), p = N-1
20
SELECCIONAR ÚLTIMA POSICIÓN DEL VECTOR V_I (p+1), P = N-1
22
24
$V_I(p) > +1 , V_I(p+1)$
COMPARAR V_I (p) CON V_I (p+1)
$V_I(p) < +1 , V_I(p+1)$
28
DECREMENTAR p
SELECCIONAR V_I (p) COMO PRIMER ELEMENTO
26
SÍ
p = 0
29
NO
SELECCIONAR V_I = p
30
SELECCIONAR V_I (p+1)
32
FIN

Fig.6

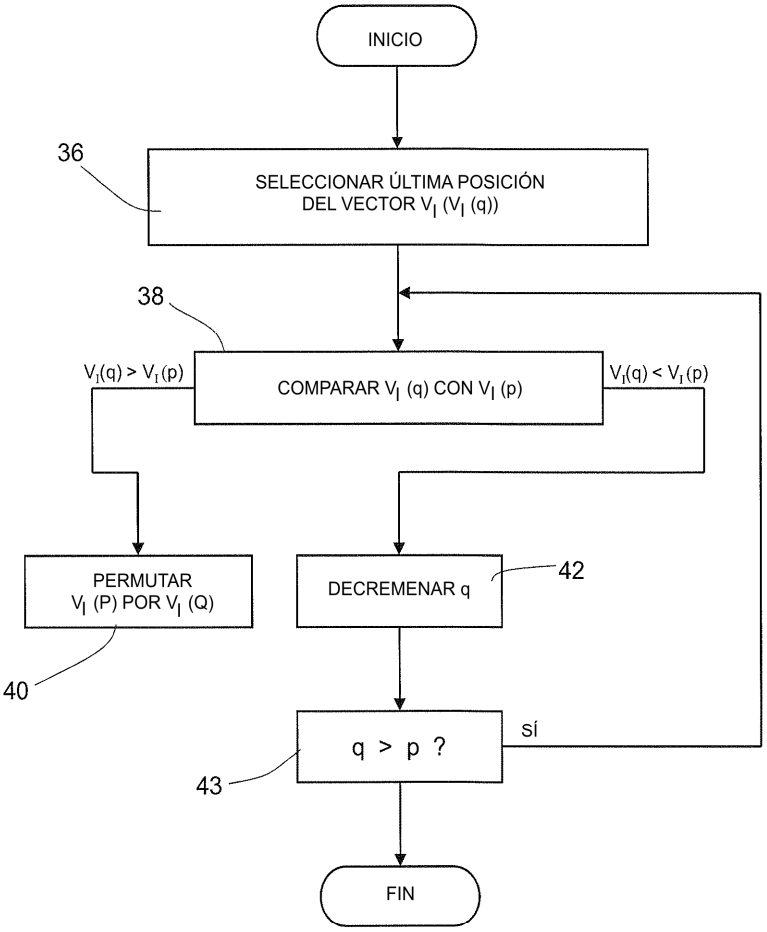
INICIO
36
SELECCIONAR ÚLTIMA POSICIÓN
DEL VECTOR V_I (V_I (q))
38
$V_i(q) > V_I(p)$
COMPARAR V_I (q) CON V_I (p)
$V_i(q) < V_I(p)$
PERMUTAR
V_I (P) POR V_I (Q)
40
DECREMENAR q
42
q > p ?
43
sí
FIN

Fig.7

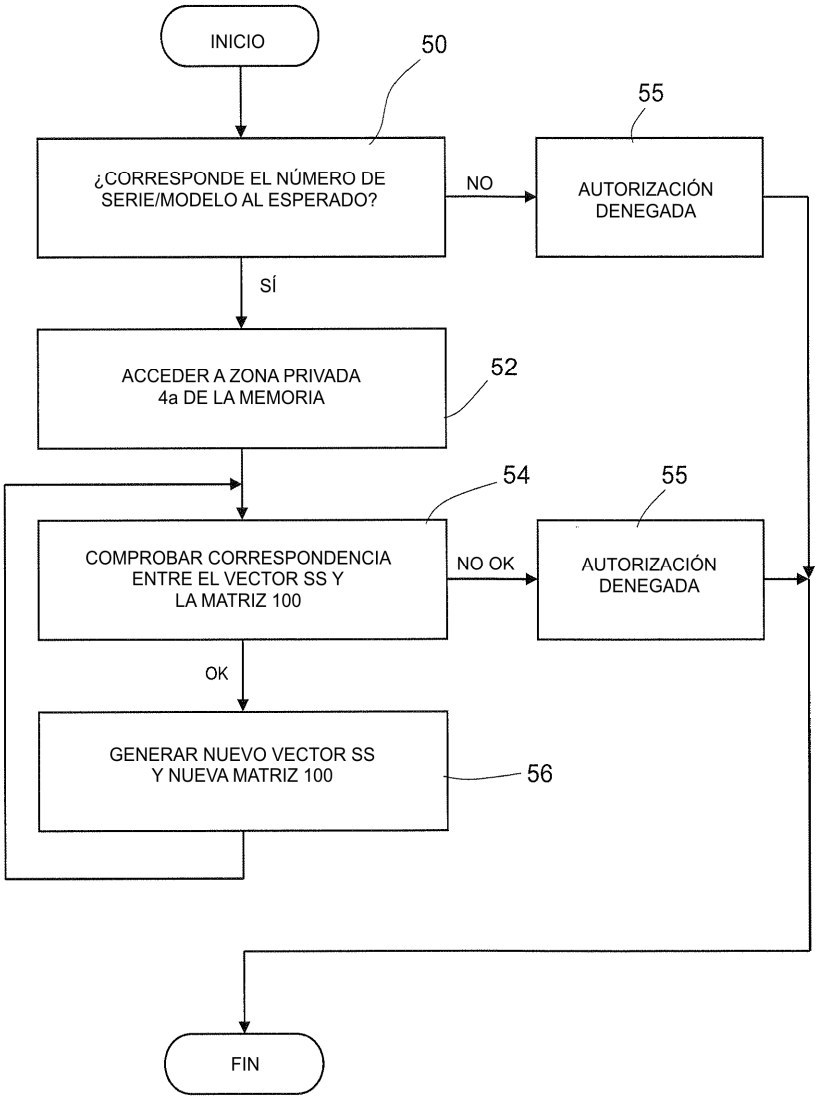
INICIO
50
55
¿CORRESPONDE EL NÚMERO DE SERIE/MODELO AL ESPERADO?
NO
AUTORIZACIÓN DENEGADA
SÍ
ACCEDER A ZONA PRIVADA 4a DE LA MEMORIA
52
54
55
COMPROBAR CORRESPONDENCIA ENTRE EL VECTOR SS Y LA MATRIZ 100
NO OK
AUTORIZACIÓN DENEGADA
OK
GENERAR NUEVO VECTOR SS Y NUEVA MATRIZ 100
56
FIN

Fig.8