(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 536 654**

(51) Int. CI.:

G06F 17/30 (2006.01)
G06F 21/00 (2013.01)
H04L 29/06 (2006.01)
H04L 29/08 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **06.05.2011 E 11729353 (0)**

(97) Fecha y número de publicación de la concesión europea: **11.02.2015 EP 2705445**

(54) Título: **Procedimiento para el almacenamiento de datos en un servidor central**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
27.05.2015

(73) Titular/es:

SECLOUS GMBH (100.0%)
Heimeranstrasse 39
80339 München, DE

(72) Inventor/es:

RAUDASCHL, WOLFGANG

(74) Agente/Representante:

DÍAZ NUÑEZ, Joaquín

PROCEDIMIENTO PARA EL ALMACENAMIENTO DE DATOS EN UN SERVIDOR CENTRAL

DESCRIPCIÓN

[0001]La invención se refiere a un procedimiento para el almacenamiento de datos en un servidor central. Este procedimiento se caracteriza porque hay múltiples usuarios, cada uno con una contraseña, que será utilizada al menos para crear una clave de usuario, y al menos a algunos de estos usuarios les corresponde al menos una máquina-cliente con datos almacenados, y al menos partes de estos datos, en caso necesario, se dividen en al menos un bloque de datos para su subida, y cada bloque de datos se compara, con ayuda de un valor ID del bloque de datos, para determinar si hay otro bloque de datos idéntico en el servidor, y los bloques de datos no disponibles se suben al servidor central. Se genera entonces una lista de bloques de datos que se sube al servidor central, a partir de la cual, cuando el usuario solicita datos que están almacenados en el servidor central, dichos datos pueden volver a generarse en su formato original mediante un proceso de recuperación de datos.

[0002] En la patente US 2010/332401 ya ha sido publicado un procedimiento de este tipo.

[0003] En la patente US 2009/313483 A1 se muestra cómo se almacenan el valor ID del bloque de datos y el valor ID de la clave del bloque de datos en una lista de claves del bloque de datos siempre y cuando este valor ID del bloque de datos todavía no esté disponible, así como cómo se sube la lista de claves del bloque de datos de la máquina-cliente al servidor central.

[0004] De la patente US 2010/058013 A1 se deriva una clave de bloque de datos que se genera exclusivamente a partir de un bloque de datos cifrado mediante el uso de un procedimiento de generación de claves.

[0005] De manera muy general, se entiende por *datos* cualquier tipo de información, sobre todo cuando se presenta de forma estructurada como ficheros, como por ejemplo imágenes, documentos, programas y archivos. Asimismo, los datos también pueden ser registros de un servidor de bases de datos u otro tipo de información que, por ejemplo, deba almacenarse en el servidor central mediante la memoria de trabajo de la máquina-cliente. En el presente documento el término ficheros se emplea como sinónimo de datos y viceversa. El nombre de los ficheros equivaldría por ejemplo a la descripción de los datos.

[0006] Por *servidor central* se entiende también una infraestructura en la nube, en la que un conjunto de servidores o granjas de servidores están conectados entre sí, pudiendo estar también repartidos en diferentes lugares geográficos. En lo sucesivo, el término servidor se utilizará como sinónimo de servidor central. Los servidores centrales son controlados por proveedores de espacio de almacenamiento en línea (proveedores de servicio) para que en estos espacios de almacenamiento en línea se puedan cubrir además los recursos necesarios para las operaciones del lado del servidor.

[0007] En la mayoría de ejemplos de aplicación, el concepto de servidor central hace referencia a un servidor externo que se comunica con el usuario a través de una red de información, como por ejemplo Internet. Asimismo, el servidor central puede estar disponible en un edificio de una empresa solo para sus empleados.

[0008] Se entiende como *usuario* tanto los usuarios particulares, como las empresas y sus empleados; un proveedor de servicios de almacenamiento en línea tiene una multitud de usuarios como clientes.

[0009] En la presente solicitud se utiliza el término *máquina cliente* para describir ordenadores personales, ordenadores portátiles, netbooks, iPhones y similares, es decir, cualquier dispositivo en el que los usuarios han almacenado sus datos, por ejemplo en forma de ficheros, y desde los cuales quieren guardarlos en el servidor. En las máquinas-cliente, una aplicación ejecuta los pasos del procedimiento en el modelo de aplicación correspondiente.

[0010] Se parte de que, para la transmisión de datos entre el servidor y la máquina-cliente, se emplea una conexión cifrada que proporciona el proveedor de servicio y que se adecua al estado actual de la técnica, por ejemplo una conexión SSL (Secure Sockets Layer).

[0011] Cuando en la solicitud se haga referencia al cifrado, se puede utilizar cualquier algoritmo de cifrado apropiado, aunque preferentemente el AES de 256 bits.

[0012] De forma análoga, si deben calcularse valores hash, el inventor recomienda uno de los candidatos finalistas SHA3 de 256 bits, ya que son considerablemente más rápidos que los SHA2.

[0013] Cabe recalcar que en la presente invención no es necesaria la utilización de RSA o curvas elípticas, si bien su empleo es opcional como algoritmo de cifrado. Los algoritmos de cifrado asíncronos son unas 1000 veces más lentos, por lo que se recomienda evitar su uso.

[0014] Según el Grupo Gartner y otras empresas de investigación de mercados de tecnología, la computación en la nube es una de las tendencias tecnológicas más relevantes del futuro próximo.

[0015] La computación en la nube ofrece a los clientes el acceso dinámico y escalable a recursos en la red, con la ventaja de que el usuario solo paga por aquellos recursos que realmente necesita y únicamente durante el tiempo que los necesita de verdad. Además es posible compartir recursos, ya sean en el seno interno de la empresa o con otros clientes o socios, pudiéndose definir estos recursos como espacio de almacenamiento o capacidad del procesamiento o servicios, entre otros.

[0016] Un aspecto fundamental para trabajar en la nube es la elevada disponibilidad de recursos y la máxima seguridad para que no pueda perderse ningún dato.

[0017] Los diferentes modelos de la computación en la nube se clasifican en:

• SaaS - Software como servicio (aplicación alojada en la red);

• DaaS - Datos como servicio (consultas del cliente a la base de datos del proveedor) ;

• PaaS - Plataforma como servicio (plataforma de desarrollo de software alojada en la red);

• IaaS - Infraestructura como servicio (el proveedor aloja las máquinas virtuales del cliente o proporciona almacenamiento en la red);

• IPMaaS - Gestión de identidad y política como servicio (el proveedor gestiona la identidad o la política de control de accesos para el cliente);

• NaaS - Red como servicio (el proveedor ofrece redes virtuales, como redes VPN), que son utilizadas en la actualidad por millones de clientes en todo el mundo.

[0018] En el ámbito de la computación en la nube, la presente invención hace referencia principalmente al modelo DaaS, si bien no está limitada únicamente a este modelo ni a la computación en la nube.

[0019] Por ejemplo, la empresa norteamericana Dropbox lleva trabajando en este ámbito desde 2008 y en 2011 ya había alcanzado 25 millones de usuarios que almacenaban diariamente 200 millones de ficheros. Dropbox ofrece a sus usuarios espacio de almacenamiento en línea con control simultáneo de versiones, con la particularidad de que este espacio virtual de almacenamiento no solo se puede utilizar para guardar datos en línea, sino para mantener sincronizados los ficheros en este espacio de almacenamiento en todas las máquinas-cliente que desee el usuario.

[0020] Otros ejemplos de empresas que ofrecen servicios en la nube son Amazon, Google, Cisco o Microsoft, entre muchas otras. Empresas como IBM, Microsoft, Oracle o Apple ven la computación en la nube como el servicio informático con más crecimiento de los próximos años y así lo han promocionado.

[0021] Sin embargo, a pesar de las ventajas de la computación en la nube, existen ciertos obstáculos para los usuarios potenciales (desde usuarios con fines particulares hasta grandes empresas internacionales). Estos obstáculos son, entre otros:

• El cumplimiento de la legislación exige la soberanía de los datos: los datos deben ponerse a disposición de las autoridades estatales si así lo solicitan.

• Debido a optimizaciones del espacio de almacenamiento por parte del proveedor de espacio de almacenamiento, la seguridad y la protección de los datos están limitadas.

• No siempre es posible garantizar los tiempos de respuesta y los anchos de banda, ya que los servidores del proveedor deben dar servicio a millones de usuarios.

[0022] Las empresas que ofrecen espacio de almacenamiento en línea tienen desde cientos de miles a millones de usuarios que desean guardar, sincronizar y distribuir sus datos a través de sus servidores.

[0023] Muchos de estos proveedores de almacenamiento en línea ofrecen un control de versiones a sus usuarios, lo que significa que todas las versiones de los ficheros se mantienen al menos durante un determinado espacio de tiempo, lo que le permite al usuario recuperarlos en caso necesario. Es habitual que se guarde un historial de los datos de manera ilimitada, cumpliendo así con los requisitos legales.

[0024] La seguridad desempeña un papel fundamental, sobre todo con datos sensibles, respecto tanto a su recuperabilidad como a la protección de tales datos frente a terceros no autorizados. En cualquier caso, en principio las autoridades estatales pueden acceder a los datos almacenados de cualquier cliente. Entendemos como terceros

aquellos que, bien sea con permiso (por ejemplo, las autoridades estatales), bien sin él (por ejemplo, espionaje industrial o cibercriminales), consiguen tener acceso a los datos.

[0025] Por esta razón, muchos servicios cifran los datos de sus usuarios con una clave propietaria, idéntica para todos los clientes de la empresa y que estos no pueden conocer. Esto implica al mismo tiempo que los datos pueden enviarse sin cifrar mediante una conexión cifrada al servidor, donde serán cifrados por este. El mismo principio se aplica en los casos en los que el servidor crea una clave para cada cliente o incluso para cada fichero: al asumir el servidor la generación de la clave, este la reconoce y debe transmitirla, en caso de duda, a terceros. Así es como es posible para los terceros recibir los datos descifrados.

Implementaciones actuales, variante 1:

[0026] La variante 1 de las implementaciones actuales permite al usuario cifrar sus datos en la máquina-cliente y transferirlos al servidor en forma ya cifrada. Cada usuario elige un nombre de usuario y una contraseña. Esta última no se transfiere al servidor. En las máquinas-cliente (PC, Mac-Books...) del usuario un programa ejecuta los siguientes pasos:

- Los datos del usuario se dividen en bloques de datos;

- se determina qué bloques de datos ya ha creado y subido este usuario y qué bloques de datos todavía deben subirse al servidor;

- los bloques de datos que deben subirse al servidor se cifran todos con la misma clave, solo conocida por el usuario;

- se genera una lista de bloques de datos, donde aparecen de forma consecutiva los bloques de datos de cada uno de los datos o ficheros;

- la lista de bloques de datos se cifra con la misma clave, conocida por el usuario, y en el servidor se cargan al menos las modificaciones que se producen en esta lista.

[0027] Todos los datos generados así se cifran con una clave que se genera a partir de la contraseña del usuario y, a continuación, se suben al servidor. De esta forma no se necesita en ningún momento transferir al servidor una contraseña de usuario.

[0028] Este procedimiento puede considerarse como muy seguro, ya que se aplica una filosofía de conocimiento cero, en la que la empresa que pone a disposición el espacio de almacenamiento no tiene información aprovechable sobre el contenido de los datos guardados en su servidor.

[0029] Otro aspecto importante de este planteamiento es que, de esta forma, los servidores no se sobrecargan cifrando y descifrando cada uno de los bloques de datos sin cifrar de los usuarios. Esto supone el ahorro de muchos recursos con millones de usuarios y enormes transferencias de datos y, por tanto, una reducción de costes para el proveedor.

[0030] Sin embargo, este planteamiento tiene una gran desventaja respecto a las implementaciones actuales de la variante 2: como los bloques de datos de cada usuario se cifran individualmente en su máquina-cliente, no cabe posibilidad de reconocer si los bloques de datos que ha generado un usuario ya han sido creados anteriormente por otros muchos usuarios.

[0031] En resumen, se puede considerar que esta variante optimiza la seguridad y ahorra capacidad de cálculo.

Implementaciones actuales, variante 2:

[0032] Este planteamiento persigue el almacenamiento de los datos en el servidor optimizando sus recursos. Esto se consigue mediante una compresión inteligente. Un bloque de datos de un tamaño determinado (por ejemplo, 8 KB) se sustituye por un valor unívoco de 256 bits o menor, según el bloque de datos. Debido a su identificación, los bloques de datos ya existentes en el servidor se ven como compresión basada en diccionario. Esto significa que, al igual que ocurre en un índice de palabras clave, una cadena de caracteres corta puede representar a una cadena de caracteres muy larga. Mediante la consulta de abreviaturas se encuentra la cadena de caracteres muy larga correspondiente. Obviamente se produce un solapamiento mínimo en cada bloque, que utiliza un único usuario una sola vez. Pero este solapamiento se compensa mucho más gracias a las ventajas, ya que se permite una compresión adicional de los bloques de datos ya comprimidos anteriormente a un nivel muy elevado.

[0033]En concreto, los proveedores de espacios de almacenamiento de datos que aplican el principio de la variante 2 pueden cobrar a sus usuarios por el espacio de almacenamiento usado por estos, aunque en algunos casos ocuparán mucho menos espacio para ciertos usuarios. La probabilidad de encontrar bloques de datos idénticos

dentro de los datos del mismo usuario es muy elevada simplemente debido al control de versiones, pero por causa del elevado número de datos generados por la gran cantidad de usuarios, aumenta enormemente la probabilidad de encontrar bloques de datos idénticos entre varios usuarios, según el principio del ataque de cumpleaños: ciertos bloques de datos de ciertos clientes pueden ser idénticos a otros bloques de datos de otros clientes. Un bloque de datos de los usuarios se parece, por la elevada compresión que se produce, al solapamiento de muchos bloques de datos no duplicados. En general, mediante este procedimiento de almacenamiento en línea puede ahorrarse un porcentaje de dos dígitos en comparación con la variante 1 antes mencionada, lo que supone una gran ventaja para el proveedor de servicios de la variante 2.

[0034] La desventaja de esta variante es que la seguridad descrita en la versión 1 en gran medida no se consigue. Asimismo, este procedimiento hace necesario que los bloques de datos se envíen de forma no cifrada al servidor y que el servidor cifre estos bloques de datos con una clave que solo pueda conocer la empresa encargada de gestionar el servidor. En esta variante, el usuario no puede cifrar por sí mismo los datos, ya que si no debería conocer la clave común y, en caso de un acceso, podría descifrar todos los datos de todos los usuarios. Sin embargo, la variante 1 contradice a la variante 2 por completo, ya que un mismo bloque de datos se cifrará por un primer usuario con la clave del primer usuario y por un segundo usuario con la clave del segundo usuario. Así pues, mientras ambos usuarios no compartan por casualidad la misma contraseña, se descarta que el mismo bloque de datos pueda ser reconocido por todos los usuarios. El servidor no puede determinar si dos usuarios utilizan la misma contraseña; por esta razón, la clave de usuario debería tener en cuenta el nombre de usuario. Solo existe una única posibilidad de que dos bloques de datos con un cifrado idéntico sean dos bloques de datos diferentes sin cifrar que, debido a las claves diferentes de los usuarios, se conviertan en el mismo bloque de datos cifrado; sin embargo, como esto prácticamente está descartado, no puede producirse una compresión mediante bloques de datos idénticos.

Objetivo de la invención:

[0035] Así pues, el objetivo de la invención es proporcionar un procedimiento para el segundo modelo que, a pesar de una filosofía de conocimiento cero, posibilita una mejor protección de datos de los clientes frente a terceros, incluso cuando toda la información sobre las actividades del usuario y los datos guardados por este están disponibles para estos terceros.

[0036] Otro de los objetivos de la invención es darle una oportunidad al proveedor de espacio de almacenamiento para aumentar la confianza de sus clientes finales, mostrándoles un procedimiento de fácil comprensión y fiable respecto a seguridad. Así se consigue superar uno de los mayores obstáculos para la utilización de la nube por parte de los clientes finales.

[0037] Asimismo, la invención tiene como objetivo mostrar un procedimiento que traslada a la máquina-cliente gran parte de las operaciones, en especial la compresión, el cifrado y el descifrado de bloques de datos, tal y como es posible únicamente en la variante 1, descrita anteriormente. Al mismo tiempo, el esfuerzo de procesamiento en la máquina-cliente debe reducirse en la medida en que se pueda reconocer, a partir de un bloque sin cifrar y una subida mínima al servidor, si este bloque de datos ya existe o si debe subirse al servidor preferiblemente en formato comprimido o, al menos, cifrado.

[0038] La invención también debe mostrar un procedimiento que en ningún caso permita que un usuario pueda provocar daños en el servidor mediante el envío de datos manipulados.

[0039] Resumiendo, el procedimiento debe ofrecer la mayor protección posible de los datos frente a terceros, cumpliendo con los siguientes requisitos:

a) Estos terceros tienen acceso a todas las claves, conocen todos los algoritmos utilizados y todo el procedimiento que se aplica en el servidor.

b) Estos terceros pueden ver en cualquier momento la información almacenada en el servidor, en especial el conjunto de todos los datos y bloques de datos, así como la lista de bloques de datos que afectan al usuario que ha aplicado el procedimiento. Así pues, el procedimiento debe garantizar que los datos guardados en el servidor sean cifrados con una clave desconocida en todo momento por el proveedor del servicio, o con una clave que solo sea conocida para aquellos que han guardado la información y que se deriva del conocimiento de qué aspecto tiene el bloque de datos cifrado, bit por bit, en forma no cifrada.

C) Para estos terceros es posible acceder a todas las configuraciones e información del usuario en cualquier momento, sobre todo los nombres de usuario y las contraseñas, por lo que el procedimiento debe evitar que sea necesario comunicar al servidor la contraseña ni, a ser posible, el nombre del usuario en una forma utilizable por este.

d) En estos terceros es posible reconocer un fichero de registro o protocolo en el cual se ha registrado cada consulta al servidor, cada subida y cada descarga, así como cada modificación del usuario en orden

cronológico. Así pues, el procedimiento debe garantizar que estos terceros no reciban ninguna información cuando por ejemplo un usuario modifique únicamente un pequeño fragmento de un fichero y solo guarde esta pequeña modificación en el servidor.

e) Esta protección de los datos debe quedar garantizada, si bien el procedimiento debe además ofrecer la posibilidad de dividir datos y bloques de datos de un tamaño determinado o variable y reconocer si el propio usuario o cualquier otro usuario ya ha creado este bloque de datos y lo ha subido anteriormente. De esta forma es posible garantizar al proveedor de espacio de almacenamiento algo fundamental para este: la reducción de la sobrecarga de sus recursos.

f) Así pues, una de las reivindicaciones más importantes es que la contraseña de usuario no puede guardarse en el servidor en ningún caso y, preferiblemente, tampoco se guardará el nombre de usuario, sino solo su valor hash, por ejemplo. Esto permite recurrir al nombre de usuario y a la contraseña para generar la clave principal del usuario sin transferir información relacionada al servidor. Gracias al valor hash del nombre de usuario guardado en el servidor, los usuarios pueden identificarse igualmente de forma unívoca. Para poder identificar al usuario con su contraseña correspondiente en el servidor, se puede generar por ejemplo una cadena de caracteres que incluya el nombre de usuario, la contraseña y una constante que contenga una cadena de caracteres proporcionada por el servidor y, a partir de esta cadena, generarse un valor hash. A continuación, este valor hash se comunica al servidor, garantizándose así la identificación del usuario mediante su contraseña.

[0040] La invención logra la transmisión de los bloques de datos subidos en el servidor central y la lista de bloques de datos ejecutando los siguientes pasos para cada bloque de datos no cifrado en la máquina-cliente:

- Generación de una clave del bloque de datos exclusivamente mediante el bloque de datos no cifrado utilizando un procedimiento de generación de claves, para que no puedan utilizarse ni la contraseña ni otros datos relevantes del usuario;

- cifrado del bloque de datos sin cifrar con la clave generada del bloque de datos en un bloque de datos cifrado, utilizando un procedimiento de cifrado;

- generación de un valor ID unívoco del bloque de datos sin cifrar y, en caso necesario, del procedimiento para la recogida de información, la generación de la clave o el cifrado, y asignación de este valor ID del bloque de datos al bloque de datos cifrado utilizando un procedimiento predeterminado de generación de ID de bloques de datos para el usuario;

- transmisión de este valor ID unívoco del bloque de datos de la máquina-cliente al servidor central para recibir una respuesta de este relativa a si ese bloque de datos cifrado al que se le ha asignado un valor ID del bloque de datos ya está disponible, así como subir el bloque de datos cifrado siempre y cuando este no esté disponible en el servidor central;

- inclusión de este valor ID unívoco del bloque de datos en la lista de bloques de datos para su subida;

- inclusión del valor ID del bloque de datos y la clave del bloque de datos en una lista de claves de bloques de datos, siempre y cuando este valor ID del bloque de datos no esté ya disponible en la lista de claves de bloques de datos, teniendo en cuenta que la lista de claves de bloques de datos puede ser parte de la lista de bloques de datos y que, utilizando todos los bloques de datos seleccionados para su subida, se ejecute lo siguiente:

- cifrar la lista de bloques de datos con una clave de lista de bloques de datos, que se genera utilizando un procedimiento de claves de listas de bloques de datos que solo conoce y gestiona el usuario;

- cifrar la lista de claves del bloque de datos con la clave de usuario;

- subir la lista cifrada de bloques de datos, así como la lista cifrada de claves de bloques de datos de la máquina-cliente al servidor central,

teniendo en cuenta que los bloques de datos no cifrados, la lista de bloques de datos no cifrados, la lista de claves de bloques de datos no cifrados y las claves de bloques de datos generadas, así como la clave de usuario, solo permanecerán en la máquina-cliente,

y que, en el procedimiento de recuperación de los datos, el servidor envía a la máquina-cliente los datos cifrados almacenados en este y que ni siquiera él puede descifrar, de manera que la recuperación de los datos únicamente puede ocurrir en la máquina-cliente.

[0041] El orden de los pasos del procedimiento de la invención no tiene ninguna limitación y puede elegirse de forma libre en la medida de lo posible.

[0042] A continuación se explica en detalle el procedimiento de la invención y sus ventajas a partir de ejemplos de aplicación. En la imagen 1 se muestra una posible disposición del servidor central y los usuarios o máquinas-cliente.

[0043] Para implementar el procedimiento de la invención se parte de una interacción entre el servidor y la máquina-cliente.

[0044] Según una de las posibles aplicaciones de la invención, el servidor distribuye los bloques de datos en tablas de una base de datos; otra posible aplicación puede incluir el almacenamiento de bloques de datos en ficheros, utilizándose por ejemplo el valor ID del bloque de datos como nombre del fichero. Debido al elevado número de bloques de datos que se podrían generar, es preferible no implementar esta solución.

[0045] Para el procedimiento no es relevante a qué algoritmo se recurrirá para el cifrado, siempre y cuando el algoritmo sea compatible con la retroalimentación de bloques y sea lo suficientemente seguro y rápido. Sin duda, uno de los algoritmos de cifrado predilectos es el AES, ya que realiza un buen análisis y el algoritmo estándar está en este ámbito.

[0046] Asimismo, se puede emplear cualquier algoritmo para el cálculo de los valores hash que devuelva, con el máximo rendimiento, un valor hash unívoco lo suficientemente extenso. En este caso se preferirá el uso de uno de los candidatos finalistas SHA-3, ya que son más seguros y eficaces que el SHA-2.

[0047] En el procedimiento de la invención se utiliza una clave para el cifrado solo conocida por el usuario. Esta será, preferentemente, un valor hash del nombre de usuario y la contraseña, pero también se puede emplear cualquier otro procedimiento para generar una clave que solo sea conocida por el usuario. En lo sucesivo, esta clave se denominará clave de usuario.

[0048] Asimismo, en este procedimiento se genera una clave para cada bloque de datos no cifrado que se obtiene al menos con la ayuda del bloque de datos no cifrado. En lo sucesivo, esta clave se denominará clave del bloque de datos y garantiza que puedan descifrarse adecuadamente cada uno de los bloques de datos solo cuando un usuario sepa o supiera cómo era el bloque de datos sin cifrar. Esto permite, por una parte, mostrar públicamente los bloques de datos cifrados, así como reconocer qué bloques de datos ya han sido creados y cargados por un usuario cualquiera. Al mismo tiempo esto provoca que sea prácticamente imposible para un tercero averiguar el contenido de determinados bloques de datos, ya que este no dispone de ninguna información sobre la clave.

[0049] Para aumentar todavía más la seguridad también se cifra la lista de bloques de datos, así como la lista de claves de bloques de datos. Ambas listas están cifradas con la clave del usuario, ya que no pueden ser conocidas por terceros. De esta forma, el propio usuario puede averiguar fácilmente qué bloques de datos necesita para poder recuperar un fichero concreto y solicitarlos y recibirlos del servidor. Para ello, el usuario utiliza la información de la lista de bloques de datos. Para poder volver a descifrar cada uno de los bloques de datos cifrados, el usuario lee las claves necesarias de su lista de claves de bloques de datos y aplica cada clave al bloque de datos correspondiente.

[0050] Como tanto el cifrado como el descifrado se producen en el lado de la máquina-cliente, y no en el servidor, en ningún momento se necesita informar al servidor sobre ninguna de las claves empleadas, ni sobre la clave de usuario ni ninguna de las claves del bloque de datos.

[0051] Si los bloques de datos cifrados están almacenados, en uno de los posibles modelos de aplicación, en una base de datos en el servidor, en esta se crearán preferiblemente al menos las siguientes tablas y columnas:

Tabla "bloques de datos cifrados":

Columna: Bloque de datos cifrado

Columna: Hash de 256 bits del bloque de datos cifrado (clave primaria: el cálculo de esta columna puede realizarse de manera automática con un disparador o *trigger*, evitando así manipulaciones por parte de terceros, por ejemplo mediante la subida de datos falsos.) También puede asignarse un índice a una función, como por ejemplo SHA256 (bloque de datos cifrado), suprimiéndose así los 256 bits para esta columna.

Tabla "listas de ficheros de usuario cifrados":

Columna: Indicador del usuario (indizado con el hash del nombre del fichero, preferiblemente un valor ID, que se relaciona con una tabla con el hash del nombre de usuario, contraseña y cadena de caracteres).

Columna: Hash del nombre del fichero (indizado con el identificador del usuario, idealmente cifrado con la clave de usuario, ya que el usuario ha cifrado el nombre del fichero en su máquina-cliente y

puede encontrarlo en el servidor en cualquier momento a través del índice. Si el nombre del fichero se cifra con la clave del fabricante, no se garantiza la protección frente a terceros).

Columna: Nombre del fichero (idealmente cifrado, completo con ruta de carpetas).

Columna: Listado de los valores ID del bloque de datos (cifrados con clave de usuario)

Columna: Información sobre la versión del fichero (preferiblemente fecha de la subida del fichero, para que, a partir de la última fecha de un fichero, se pueda reconocer cuál es la última versión. Esto solo se necesita si se ofrece control de versiones).

Columna: Hash del fichero (para comprobar rápidamente si este fichero ha sido modificado y si el fichero ha sido recuperado de forma adecuada)

Tabla "Lista de claves del bloque de datos":

Columna: Indicador del usuario

Columna: Valor ID del bloque de datos (indizado, cifrado con clave de usuario)

Columna: Clave del bloque de datos (cifrado con clave de usuario)

Tabla "Usuario":

Columna: Contraseña unívoca del usuario

Columna: Datos de acceso del usuario (indizados, por ejemplo, hash del nombre de usuario, contraseña y cadena de caracteres, relacionada con la tabla "listas de ficheros cifrados de usuarios")

Columnas: Otros datos del usuario (apellidos, nombre, espacio de almacenamiento disponible..., preferiblemente cifrados con una clave conocida únicamente por el servidor y sus administradores)

[0052] El modelo de aplicación descrito anteriormente solo es uno de muchas posibles. Este modelo de aplicación ofrece las siguientes ventajas:

1. Los datos no se almacenan de manera redundante, por lo que solo se ocupa un espacio de almacenamiento mínimo.

2. Los datos se almacenan en una forma muy apropiada para ampliaciones y otras aplicaciones de esta invención, tal como se mostrará a continuación.

3. La carga del servidor se limita a dar respuesta a las consultas SQL y a introducir los datos en las tablas. El servidor no debe almacenar de forma temporal ningún dato de usuario, ya que en la tabla "Usuario" se puede ejecutar un JOIN mediante una consulta SQL en la tabla "Listas de ficheros cifrados del usuario".

[0053] Un modelo alternativa de aplicación para un modelo de datos en el servidor implica ampliar la tabla "Listas de ficheros cifrados del usuario" con la columna "Claves del bloque de datos", eliminándose así la tabla "Lista de claves del bloque de datos". Esta forma de aplicación ofrece pocas ventajas, ya que se elimina un VÍNCULO entre estas dos tablas mediante el valor ID del bloque de datos. Una de las desventajas es que las cantidades de datos que deben guardarse y, al mismo tiempo, la transferencia de datos entre la máquina-cliente y el servidor aumentan considerablemente. En este caso, una compresión de la columna "Listado de los valores ID del bloque de datos" tampoco aporta más ventajas en ningún otro modelo de aplicación.

[0054] Otro modelo alternativa de aplicación puede ser guardar la columna "Listado de los valores ID del bloque de datos" en la tabla "Listas de ficheros cifrados del usuario", en lugar de como un listado de cadenas de caracteres en formato binario de los valores ID del bloque de datos, también como registro de datos por cada valor ID del bloque de datos de un fichero. Esta forma de aplicación no ofrece ninguna ventaja y sí algunas desventajas.

[0055] Los especialistas conocen otras posibilidades cuyas ventajas y desventajas deben valorarse, sobre todo respecto a los requisitos actuales, con el fin de poder seleccionar el modelo de datos óptimo en un caso concreto.

[0056] De forma ideal, si el nombre del fichero se guarda en cada ocasión con un cifrado diferente, también puede añadirse un hash del nombre del fichero no cifrado como columna adicional, posibilitando así el control de versiones para un fichero en el lado del servidor, sin necesidad de conocer una clave. Este mismo principio es aplicable para la

mayoría de las columnas de la mayoría de tablas. Así se consigue que toda la información relativa a un fichero se cifre por completo en el servidor y no la pueda utilizar nadie excepto el usuario que ha creado dicho fichero.

[0057] Cada usuario tiene un nombre de usuario y una contraseña. La contraseña no se guarda en la nube y, a ser posible, tampoco el nombre de usuario. En uno de los modelos de aplicación se puede guardar en el servidor un hash de 256 bits con la siguiente información: nombre de usuario y contraseña junto con una cadena de caracteres constante. Así se pueden identificar de manera unívoca tanto el usuario como su contraseña para poder devolver una lista de ficheros e índices al usuario.

[0058] Para recibir bloques de datos cifrados y guardados en el servidor como datos no cifrados o ficheros en su máquina-cliente, se ejecutan los siguientes pasos en la máquina-cliente, idealmente en este mismo orden:

1. Recuperar la lista cifrada de ficheros de bloques de datos para el fichero correspondiente.

2. Descifrar la lista de ficheros de bloques de datos con la clave de usuario.

3. Recuperar del servidor los bloques de datos en esta lista a partir del valor ID del bloque de datos de la lista descifrada de ficheros de bloques de datos.

4. Recuperar la lista cifrada de claves del bloque de datos del servidor y descifrarla con la clave de usuario.

5. Descifrar los bloques de datos cifrados con la clave del bloque de datos correspondiente de la lista de claves de bloques de datos.

6. Encadenar los bloques de datos sin cifrar y guardarlos con el nombre de fichero deseado bien en la memoria de trabajo, bien en el disco duro.

[0059] Si el usuario quiere modificar su contraseña, a pesar de haber guardado ya datos en el servidor, se mantendrá la contraseña original pero se cifrará con una nueva contraseña como modelo de aplicación. De forma alternativa la lista de bloques de datos y la lista de claves de bloques de datos pueden volverse a cifrar.

[0060] Los propios bloques de datos cifrados no deben volver a cifrarse, ya que cada bloque de datos tiene su propia clave.

[0061] Se trata de un modelo de aplicación en el que se mantiene la lista de bloques de datos recuperada o creada directamente en la máquina-cliente, así como la lista de claves de bloques de datos. Así se elimina un viaje de ida y vuelta al servidor. Si los ficheros de un usuario se modifican y sincronizan en diferentes máquinas-cliente, el servidor únicamente recupera los registros modificados o que faltan de la lista de bloques de datos y de la lista de claves de bloques de datos. La lista solo se completará o modificará con los registros modificados en otras máquinas-cliente.

[0062] Es posible detectar fácilmente una modificación del bloque de datos en otra máquina-cliente comparando la lista de bloques de datos con la del servidor. Asimismo, es posible detectar, a partir del valor hash del fichero completo o del valor hash de los datos completos, si un fichero ha sufrido alguna modificación y si ha de dividirse en bloques.

[0063] Otro modelo de aplicación consiste en que se genere y se envíe al servidor el valor hash de la lista cifrada de bloques de datos, para determinar directamente si se ha modificado el fichero. Asimismo es posible guardar en el servidor el hash de todo un fichero no cifrado que alguna vez haya sido modificado ligeramente, ya que así puede reconocerse muy rápidamente si un fichero ha sido modificado antes de que el fichero sea dividido en bloques de datos. En este sentido se amplía por ejemplo la tabla “lista de bloques de datos” añadiendo una columna que contiene el valor hash descrito del fichero sin cifrar. Este valor hash puede guardarse en el servidor tanto cifrado como sin cifrar, ya que no puede extraerse ninguna información del mismo. Si el valor hash no está cifrado, la máquina-cliente puede solicitarlo directamente al servidor subiendo el valor hash; si el valor hash está cifrado, este debe cargarse primero en la máquina-cliente y, a continuación, descifrarse en este.

[0064] Sin embargo, el cifrado incluso de este valor hash presenta la ventaja de que en la recuperación de un fichero por parte de terceros, por ejemplo al adivinar cada una de las claves del bloque de datos o mediante el intento de reproducir un fichero idéntico, no puede verificarse si este es el mismo fichero que el usuario en realidad ha grabado.

[0065] En un modelo de aplicación, el especialista solo subirá al servidor aquellos registros de la lista de bloques de datos y de la lista de claves del bloque de datos donde pueda determinar que han sufrido alguna modificación.

[0066] Otro modelo de aplicación consiste en cargar siempre en el servidor todo el bloque de datos cifrado para comprobar si este ya está disponible en el servidor y, en caso contrario, puede guardarse inmediatamente en el servidor. En este caso, el valor ID de los bloques de datos sería idéntico al propio bloque de datos cifrado. Este modelo de aplicación solo se nombra con afán de ser exhaustivo, ya que en la práctica no se llevará a cabo.

[0067] En resumen se aplica que, en total, por cada tamaño de bloque libremente elegible:

En el lado de la máquina-cliente: hash de 256 bits del bloque sin cifrar + hash de 256 bits del bloque cifrado

En el lado del servidor: hash de 256 bits del bloque cifrado (indizado)

- es decir, al final se juntan aprox. 1024 bits adicionales

[0068] El tamaño de los bloques se ponderará, siendo un valor razonable entre 8 KB y 16 KB, ya que así la probabilidad de encontrar bloques idénticos es todavía relativamente alta.

[0069] El solapamiento se sitúa en estos casos entre el 0,08 % y el 0,04 % aproximadamente; por el contrario, la tasa de compresión se sitúa en un factor entre 12,8 y 25,6.

[0070] Otra forma de aplicación puede ser que el servidor devuelva para este bloque un valor ID unívoco al encontrar o crear un bloque de datos comprimido y cifrado. Esto requiere una nueva columna indizada que incluya el valor ID en la tabla 1 y, en lugar del HASH, se escribirá el ID del bloque al componer el fichero.

[0071] Una de las ventajas de este procedimiento es que no se necesitan más de 256 bits por bloque para el usuario, sino únicamente un mínimo de 64 bits y, como mucho, 128 bits. Los costes de memoria adicionales en la tabla 1 merecen la pena cuando un bloque aparece tres veces o más. (Versiones de ficheros)

[0072] Por otro lado, se descarta una manipulación por parte de los usuarios, ya que los valores ID se generan exclusivamente en el servidor y se envían a la máquina-cliente solo para hacer una consulta.

[0073] Otro modelo de aplicación puede ser que los bloques no sean consultados en el servidor en el orden en el que están, sino en un orden aleatorio u ordenado.

[0074] Aunque esto requiere de más almacenamiento en la máquina-cliente (ya que se han de mantener los bloques hasta que se ultimen uno a uno), la recuperación de un fichero implica algo más, ya que no se puede saber en qué orden se recomponen los bloques en el fichero.

[0075] Otro modelo de aplicación puede ser que la máquina-cliente guarde de forma local todos los bloques que haya generado alguna vez o sobre los que sepa, por otros medios, que existen (por ejemplo en una base de datos local de SQLite). Así se consigue de nuevo ahorrar viajes de ida y vuelta al servidor y, al mismo tiempo, se da a conocer menos información.

[0076] Otro modelo de aplicación puede ser que por ejemplo los primeros x bits (por ejemplo, los primeros 128 bits) de cada bloque se dejen y solo se añadan a la lista de bloques de datos.

[0077] Esto, junto con los valores ID para los bloques hash, hace que tenga sentido comprimir la lista de bloques de ficheros (ya que no solo aparecen valores hash en esta).

[0078] Si bien esto supone un coste de más memoria por bloque de datos de un usuario (reducible por compresión), (solapamiento: 16 bytes comprimidos más y un bloque menos, probabilidad de acierto mayor, para los mismos bloques algo mayor, -> prácticamente 0 ...), por otro lado ofrece algunas ventajas:

1. Los bloques comprimidos y cifrados se reducen de nuevo un par de bytes, aumentando aún más la probabilidad de encontrar bloques idénticos de usuarios iguales o diferentes.

2. Si un bloque en forma no cifrada ya es conocido (porque este bloque es conocido como texto fuente), todavía faltan 16 bytes por bloque para los que no hay ningún tipo de información, ya que estos están cifrados con la clave de usuario.

[0079] En un fichero de 1 MB y un bloque de tamaño 16 KB, todavía hay 16 * 64 bytes = 1 KB totalmente desconocidos (0,1 % de todo el fichero), por lo que nunca puede confirmarse al 100 % que fue justo este fichero el que se utilizó.

[0080] Otro modelo de aplicación consiste en que no se eliminen de forma fija los primeros 16 bytes del bloque, sino que, de los 16384 bytes (16 KB), se escoja una zona cualquiera de posibles zonas de 256 (u otra cantidad) (1 byte para la posición en la que se eliminó, determinada cada vez por un generador aleatorio).

[0081] Ventaja: Esta eliminación de ciertas zonas en el bloque y el almacenamiento extra tiene la gran ventaja de que no se puede saber qué zona ha sido eliminada exactamente.

[0082] Desventaja: hasta 1 byte más de almacenamiento por cada bloque de fichero (ya que está comprimido), la probabilidad de encontrar los mismos bloques disminuye, ya que los mimos bloques fuente pueden estar divididos hasta de 256 maneras diferentes (u otra cantidad).

[0083] Por causa de la variedad de posibilidades, al especialista le quedan otras posibilidades de aplicación y mejora que no detallaremos en este documento.

[0084]) Se recomienda que el nombre de usuario nunca se guarde en el servidor, sino únicamente su valor hash.

[0085] En la siguiente tabla se ilustra de forma ejemplar la aplicación de los pasos del procedimiento de la invención.

	Cliente	Servidor/Nube				
Paso nº	Ámbito	Acción en el cliente	Acción en el cliente Modelos de aplicación	Comentarios, detalles	Acción en el servidor	Ejemplo
1	Todos los datos (fichero)			Inicialización: Vaciar la lista de bloques de datos para este fichero		Un fichero con por ejemplo 100 KB
2	Todos los datos (fichero)	Dividir datos en bloques de datos de un tamaño determinado		También pueden seleccionare diferentes tamaños		Por ejemplo bloques de datos de 10 * 10 KB
3	Opcional Todos los bloques de datos	Comprimir el bloque de datos				Por ejemplo 10 KB se transforman en 5 KB
4	Opcional Todos los bloques de datos	Generar una información base de clave a partir de un bloque de datos no cifrado utilizando un procedimiento de obtención de información	Si se ejecuta el paso 3, alternativamente: bloque de datos comprimido o bloque de datos modificado según un sistema determinado como base de un procedimiento de obtención de información	Permite pequeños valores para el almacenamiento de información de descifrado; opcionalmente están disponibles varios procedimientos de obtención de información y cada vez se elige un nuevo procedimiento de obtención de información con un		Por ejemplo un fragmento de 128 bits del hash skein del bloque de datos no cifrado = por ejemplo "UVWXYZ0987654321"; el hash del bloque comprimido es más rápido, ya que el tamaño del bloque es más pequeño y mejor, pues el texto fuente para la fuerza bruta no es conocido directamente, pero el resultado de la compresión no siempre es el mismo a pesar de ser los mismos datos

				generador aleatorio		
5	Si se ejecuta el paso 4 - Todos los bloques de datos	Generar una clave del bloque de datos a partir de la información base generada de clave utilizando un procedimiento de generación de claves	En este paso se puede introducir una ralentización voluntaria para reforzar la información base de clave.	El procedimiento de generación de claves se selecciona de un número predeterminado de procedimientos de generación de claves		Para poder generar una clave de 256 bits, pero ralentizando el camino para reforzar la información base de clave de 128 bits: SHA256(SHA512(SHA512(SHA512 (SHA512(información base de clave) +'A') +'BC') + 'DEF') + 'Z') por ejemplo = "werfufng9otrilekU5util128exs2lpy"
6	Si se ejecuta el paso 4 - Todos los bloques de datos	Generar una clave del bloque de datos a partir del bloque de datos no cifrado utilizando un procedimiento de generación de claves	Si se ejecuta el paso 3, el procedimiento de generación de claves puede hacer referencia también al bloque de datos comprimido	El procedimiento de generación de claves se selecciona de un número predeterminado de procedimientos de generación de claves		Por ejemplo los candidatos Skein 256 (bloque de datos no cifrado) SHA3 son más rápidos que SHA2 = por ejemplo "+Z(9)S"
7	Si se ejecuta el paso 3 - Todos los bloques de datos	Cifrar el bloque de datos comprimido con la clave generada del bloque de datos en un bloque de datos cifrado utilizando un procedimiento de cifrado		El procedimiento de cifrado se selecciona de un número predeterminado de procedimientos de cifrado		Por ejemplo AES256 (bloque de datos comprimido) con la clave del bloque de datos y el algoritmo CFB
8	Si no se ejecuta el paso 3 - Todos los bloques	Cifrar el bloque de datos no cifrado con la clave generada del bloque de datos en un bloque de datos cifrado, utilizando un procedimiento de		El procedimiento de cifrado se selecciona de un número predeterminado de procedimientos de cifrado		Por ejemplo AES256 (bloque de datos no cifrado) con la clave del bloque de datos y el algoritmo CFB

	de datos	cifrado				
9	Todos los bloques de datos	Generar un valor ID unívoco a partir del bloque de datos sin cifrar y del procedimiento elegido de obtención de información, de generación de la clave de usuario o de cifrado, y asignar el valor ID generado del bloque de datos al bloque de datos cifrado utilizando un procedimiento predeterminado para el usuario de generación de ID de bloques de datos	Generar un valor ID unívoco del bloque de datos a partir del bloque de datos cifrado (excluye manipulaciones posteriores mediante subidas fraudulentas en el servidor) como procedimiento de generación de ID de bloques de datos	Si el valor ID del bloque de datos se genera a partir del bloque de datos no cifrado, el servidor no puede validarlo, por lo que el servidor se ve obligado a confiar en todos los usuarios, pudiendo estos asignar bloques de datos cifrados a valores ID erróneos de bloques de datos, lo que perjudicaría a otros usuarios		Por ejemplo los candidatos Skein 256 (bloque de datos cifrado) SHA3 son más rápidos que SHA2 = por ejemplo "1234567890ABCDEFGHIJKLMNOPQRST UV"
9	Todos los bloques de datos	Transmitir este valor ID unívoco del bloque de datos al servidor		Debería transferirse mediante una conexión cifrada y la información transferida debería ser mínima		Por ejemplo "? " + valor ID del bloque de datos- "?1234567890ABCDEFGHIJKLMNOPQRSTU V"
10	Todos los bloques de datos			Como se especifica en la reivindicación 1, el servidor puede devolver como respuesta SÍ o NO; entonces el valor ID del bloque de datos permanece sin cambios en el cliente durante los siguientes pasos. Así pues, como	Comprobar si el bloque de datos cifrado en el servidor, correspondiente al valor ID del bloque de datos, ya está disponible y responder al cliente	Por ejemplo RETURN "SELECT" "valor ID del bloque de datos incrementados" FROM "Bloque de datos cifrado" WHERE "bloque de datos de hash cifrado"=" + valor ID del bloque de datos

				consecuencia, la ventaja de este valor ID incrementado del bloque de datos es que son valores ID del bloque de datos más pequeños y comprimibles		
11	Todos los bloques de datos	Comprobar la respuesta del servidor y subir el bloque de datos cifrado siempre y cuando este no esté disponible todavía en el servidor				Por ejemplo solo cuando el valor devuelto del servidor = 0: subir bloque de datos cifrado
12				Devolver el valor ID del bloque de datos incrementado al el cliente como confirmación de que el proceso ha concluido	Nueva comprobación del valor hash del bloque de datos (el cliente puede subir datos de forma duplicada) Guardar el bloque de datos cifrado	Por ejemplo en la base de datos del sistema de ficheros En el primer caso se ejecutan los correspondientes comandos SQL (Insert into ...)
13		Incluir el valor ID unívoco del bloque de datos en la lista de bloques de datos para su subida	Opcional: el valor ID del bloque de datos se sustituye por el valor ID del bloque de usuarios incrementado, que será devuelto por el servidor			Por ejemplo el valor ID del bloque de datos binario "1234567890ABCDEFGHIJKLMNOPQRST UV" se sustituirá por el valor numérico 4567 que ha sido devuelto por el servidor -> añadir 4567 al final de la lista de bloques de datos
14	Si se ejecuta el paso 4	Guardar el valor ID del bloque de datos y la información base de	La información base de clave forma parte de la lista de bloques	Preferiblemente, el valor hash del bloque de datos cifrado		Por ejemplo ..., 4567-UVWXYZ0987654321, ... El valor ID del bloque de datos ha sido sustituido por el

	- Todos los bloques de datos	clave en una lista de claves de bloques de datos, que puede ser parte de la lista de bloques de datos, siempre y cuando no esté ya en esta lista	de datos (por ejemplo se añade detrás de cada valor ID del bloque de datos)	calculado por el cliente será añadido también a esta lista, ya que después puede ser comparado con esta lista en el cliente, con lo que ya no será necesario comprobar en el servidor la existencia de los bloques de datos conocidos.		valor devuelto por el servidor
	Si no se ejecuta el paso 4 - Todos los bloques de datos	Guardar el valor ID del bloque de datos y de la clave del bloque de datos en una lista de claves de bloques de datos, que puede ser parte de la lista de bloques de datos, siempre y cuando no esté ya en esta lista	La clave del bloque de datos forma parte de la lista de bloques de datos (por ejemplo se añade detrás de cada valor ID del bloque de datos)	Preferiblemente, el valor hash del bloque de datos cifrado calculado por el cliente será añadido también a esta lista, ya que después puede ser comparado con esta lista en el cliente, con lo que ya no será necesario comprobar en el servidor la existencia de los bloques de datos conocidos.		Por ejemplo.:., 1234567890ABCDEFGHIJKLMNOPQRSTU V - 58thglifewe8fdmnre539128wialqwy<, ... por ejemplo El servidor devuelve solo SÍ/NO, por lo que debe guardarse el valor ID del bloque de datos calculado en un principio
				Pasos 3 hasta 14 para todos los bloques de datos		
15		Cifrar la lista de bloques de datos con una clave de lista de bloques de datos, que	La clave de la lista de bloques de datos será calculada de forma diferente bien	La ventaja de estas formas de aplicación es que los terceros, en cualquier lista de		El vector de entrada o base para una clave de listas de bloques de datos es, por ejemplo, la fecha de modificación actual grabada en el servidor de cada

		se genera utilizando un procedimiento de claves de listas de bloques de datos que solo conoce y gestiona el usuario	a partir de la lista de bloques de datos, o bien utilizando un vector de inicialización según la lista de bloques de datos	bloques de datos con historial y en la que solo cambia uno de los últimos bloques de datos, no pueden saber a partir de qué valor ID del bloque de datos se ha producido la primera edición, eliminación o escritura.		una de las listas de bloques de datos
16		Cifrar la lista de claves del bloque de datos con la clave de usuario		Aquí no hay ningún valor añadido por la utilización de vectores de inicialización o claves diferentes, ya que no hay historial		
17		Subir la lista cifrada de bloques de datos, así como la lista cifrada de claves de bloques de datos, del cliente al servidor central		Los bloques de datos no cifrados, la lista de bloques de datos no cifrados, la lista de claves de bloques de datos no cifrados y las claves de bloques de datos, la clave de usuario y el procedimiento de claves de listas de bloques de datos y, si se ha generado, la información base de clave, permanecen exclusivamente en el servidor.		
18				La lista de claves de bloques de datos puede guardarse en la	Guardar la lista de bloques de datos cifrados y la lista	Por ejemplo una versión de un fichero es una línea en la tabla "lista de ficheros de usuarios cifrados" y la lista de bloques de datos será guardada como

				de claves de bloques de datos cifrados	columna de texto cifrada
			tabla "lista de claves de bloques de datos" como un registro en cada bloque de datos diferente		

REIVINDICACIONES

1. Procedimiento para grabar datos en un servidor central, cada uno de una pluralidad de usuarios que tiene una palabra de paso de usuario que es utilizada por lo menos para generar una llave de usuario y por lo menos un cliente sobre el cual son grabados los datos siendo atribuido a cada uno por lo menos una parte de los dichos usuarios y por lo menos una parte de los dichos datos que son atribuidos según las necesidades por lo menos a un bloque de datos a ser transferido y cada bloque de datos que es comparado con los bloques de datos sobre el servidor con la ayuda de un valor único de identidad de bloque de datos para determinar si los bloques de datos ya están presentes sobre el servidor y los bloques de datos que no están presentes sobre el servidor que son transferidos sobre el servidor central y una lista de datos / bloques de datos que hay que transferir sobre el servidor, con la ayuda de la cual los datos pueden ser regenerados en su estado original en el curso de una etapa de recuperación de datos después de una petición del usuario, estando establecida y transferida sobre el servidor central,

Caracterizado por

que cada bloque de datos no cifrado sobre el cliente está sometido a las etapas siguientes para identificar los bloques de datos que hay que transferir sobre el servidor central y establecer la lista de datos / bloques de datos que hay que transferir sobre el servidor central:

- Generar una llave de bloque de datos teniendo exclusivamente como base un bloque de datos no cifrado utilizando un protocolo de generación de llave, de manera que la palabra de paso del usuario y otros datos del usuario no son utilizados para generar dicha llave,
- Cifrar el bloque de datos no cifrado utilizando un protocolo de encriptación teniendo como base dicha llave de bloque de datos generada para obtener un bloque de datos cifrado,
- Generar un valor único de identificación de bloque de datos teniendo como base el bloque de datos no cifrado y, según el caso, sobre un protocolo de recuperación de informaciones y/o de generación de llave y/o de encriptación y atribuir dicho valor de identidad de bloque de datos al bloque de datos cifrado utilizando un protocolo de generación de identidad de bloque de datos predefinido para el usuario,
- Transferir dicho valor único de identificación de bloque de datos del cliente sobre el servidor central para obtener una respuesta del servidor si el bloque de datos cifrado atribuido a dicho valor de identificación de bloque de datos está ya presente sobre el servidor central y transferir el bloque de datos cifrado sobre el servidor central si no está todavía allí presente,
- Inscribir dicho valor único de identificación de bloque de datos en la lista de datos / bloques de datos que hay que transferir sobre el servidor central,
- Inscribir dicho valor de identificación de bloque de datos y dicha llave de datos en una lista de bloques de datos / llaves si dicho valor de identificación de bloque de datos no está comprendido todavía en dicha lista de bloques de datos / llaves, dicha lista de bloques de datos / llaves que puede formar parte de la lista de datos / bloques de datos,

y **por que** las etapas siguientes pueden ser realizadas para todos los bloques de datos identificados para ser transferidos sobre el servidor central:

- Cifrar la lista de datos / bloques de datos utilizando una llave de lista de datos / bloques de datos que es generada utilizando un protocolo de generación de llave de lista de datos / bloques de datos exclusivamente por el usuario,
- Cifrar dicha lista de bloques de datos / llaves utilizando la llave de usuario,
- Transferir la lista cifrada de datos / bloques de datos y la lista cifrada de bloques de datos / llaves del cliente sobre el servidor central,

Los bloques de datos no cifrados, la lista no cifrada de datos / bloques de datos, la lista no cifrada de bloques de datos / llaves y las llaves fechadas de bloques generadas y la llave de usuario que se quede sobre el cliente,

el servidor que enviaba datos que son grabados sobre el servidor en su estado cifrado y no pueden ser descifrados por el servidor al cliente en el curso de la etapa de recuperación de datos, para que sea solo sobre el cliente se puedan recuperar los datos.

2. Procedimiento según la reivindicación 1, **caracterizado por que** la llave de bloque de datos es un valor hash del bloque de datos no cifrado.

3. Procedimiento según la reivindicación 1, **caracterizado por que** informaciones de base de llaves son generadas con la ayuda del bloque de datos no cifrado utilizando un protocolo de recuperación de informaciones y **por que** la llave de bloque de datos es generada exclusivamente con la ayuda de las dichas informaciones de base de llaves, preferentemente utilizando una parte predeterminada suficientemente larga de un valor hash del bloque de datos no cifrado y las informaciones de base de llaves que son preferentemente inscritas en la lista de bloques de datos / llaves en lugar de la llave de bloque de datos.

4. Procedimiento según la reivindicación 1 ó 3, **caracterizado por que** la llave de bloque de datos es generada con la ayuda de las informaciones de llave de datos en un número predeterminado de ciclos de generación de valores hash que proceden del valor hash de las informaciones de llave de datos, el valor hash que resulta del ciclo hash precedente siendo utilizado como nuevo valor hash de salida.

5. Procedimiento según una de las reivindicaciones precedentes, **caracterizado por que** el bloque de datos no cifrado primero es comprimido y luego cifrado utilizando la llave de bloque de datos.

6. Procedimiento según la reivindicación 5, **caracterizado por que** la llave de bloque de datos es un valor hash del bloque de datos comprimido.

7. Procedimiento según la reivindicación 5, **caracterizado por que** las informaciones de base de llave son una parte definida suficientemente larga de un valor hash del bloque de datos comprimido y **por que** la verdadera llave de bloque de datos es calculada teniendo como base este valor hash.

8. Procedimiento según una de las reivindicaciones precedentes, **caracterizado por que** el valor de identificación de bloque de datos corresponde a un valor hash del valor hash del bloque de datos no cifrado.

9. Procedimiento según una de las reivindicaciones 1 a 7, **caracterizado por que** el valor de identificación de bloque de datos corresponde al valor hash del bloque de datos cifrado.

10. Procedimiento según una de las reivindicaciones precedentes, **caracterizado por que** el valor de identificación de bloque de datos es enviado al servidor y el servidor anuncia si el bloque de datos correspondiente está ya presente sobre el servidor o debe ser transferido allí y **por que** el servidor le envía un número único para cada valor de identificación de bloque de datos, atribuido a dicho valor de identificación de bloque de datos por el servidor, al cliente, donde dicho número es inscrito sobre la lista de datos / bloques de datos y la lista de bloques de datos / llaves como nuevo valor de identificación de bloque de datos y utilizado para la etapa de recuperación de datos.

11. Procedimiento según una de las reivindicaciones precedentes, **caracterizado por que** todo valor de identificación de bloque de datos, con el que el servidor determina si los bloques correspondientes de datos están ya presentes sobre el servidor, son generados primero sobre el cliente y luego enviados al servidor en una orden aleatoria para la petición.

12. Procedimiento según una de las reivindicaciones 1 a 11, **caracterizado por que** un número variable de bytes de cada bloque de datos no cifrado es quitado de una posición variable y añadido como información suplementaria para cada bloque de datos a la lista de datos / bloques de datos con las informaciones que conciernen al número de bytes quitados y su posición.

13. Procedimiento según una de las reivindicaciones precedentes, **caracterizado por que** un valor que no puede ser modificado para la lista de datos / bloques de datos y es conocido públicamente es utilizado como una base para calcular el vector de inicialización de un procedimiento de encadenamiento para cifrar dicha lista de datos / bloques de datos.

14. Procedimiento según una de las reivindicaciones precedentes, **caracterizado por que** datos aleatorios, específicamente marcados, pero de otro modo insignificantes, de una longitud variable son añadidos a la lista de datos / bloques de datos.

15. Procedimiento según una de las reivindicaciones precedentes, **caracterizado por que** un valor aleatorio de manipulación de llave para modificar la llave de bloque de datos es determinado para cada bloque de datos no cifrado antes de cada encriptación y **por que** dicho valor aleatorio de manipulación de llave es registrado en la lista de datos / bloques de datos.

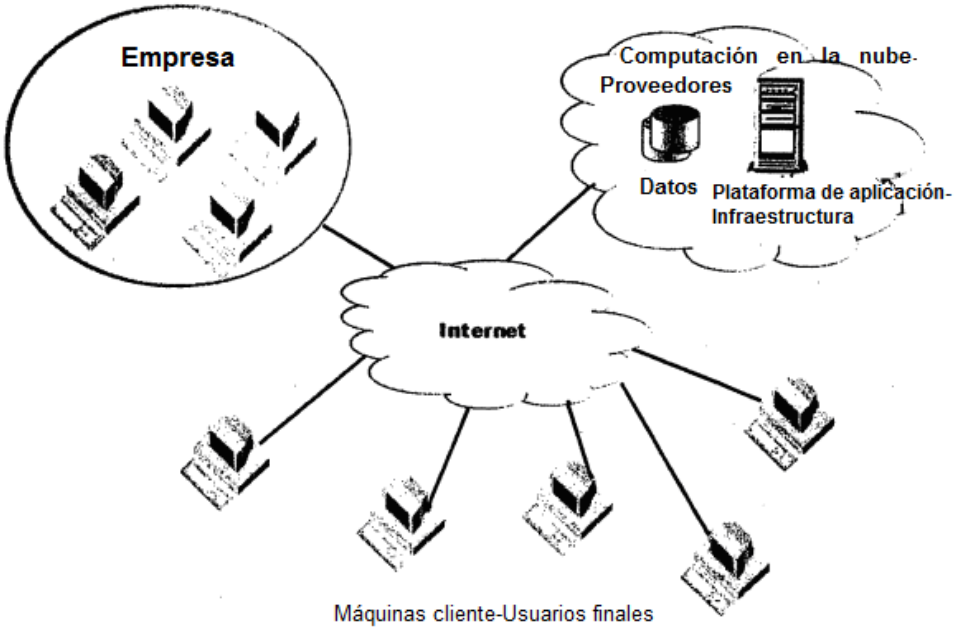
Empresa
Computación en la nube-
Proveedores
Datos
Plataforma de aplicación-
Infraestructura
Internet
Máquinas cliente-Usuarios finales

Figura 1