(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 568 661**

(51) Int. CI.:

H04L 9/08 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **07.11.2007 E 07873471 (2)**

(97) Fecha y número de publicación de la concesión europea: **27.01.2016 EP 2100404**

(54) Título: **Sistemas y métodos para distribuir y garantizar datos**

(30) Prioridad:

07.11.2006 US 857345 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
03.05.2016

(73) Titular/es:

SECURITY FIRST CORP. (100.0%)
22362 Gilberto, Suite 130
Rancho Santa Margarita, CA 92688, US

(72) Inventor/es:

BELLARE, MIHIR y
ROGAWAY, PHILLIP

(74) Agente/Representante:

ISERN JARA, Jorge

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

DESCRIPCIÓN

Sistemas y métodos para distribuir y garantizar datos

Referencia cruzada a las solicitudes relacionadas

Esta solicitud reivindica el beneficio de la solicitud provisional de Estados Unidos N.º 60/857.345, presentada el 7 de noviembre de 2006.

Campo de la invención

La presente invención se refiere, en general, a un sistema para garantizar los datos contra el acceso o el uso no autorizado. La presente invención también se refiere, en general, a técnicas de cifrado para la construcción de esquemas de compartición de secretos, y más específicamente a sistemas y métodos para soportar un esquema de compartición de secretos que puede tolerar el daño a una o más comparticiones.

Antecedentes de la invención

En la sociedad actual, los individuos y las empresas realizan una cantidad cada vez mayor de actividades en y sobre los sistemas informáticos. Estos sistemas informáticos, que incluyen las redes informáticas propietarias y no propietarias, a menudo almacenan, archivan y transmiten todo tipo de información sensible. Por lo tanto, existe una necesidad cada vez mayor de garantizar que los datos almacenados y transmitidos a través de estos sistemas no pueden leerse o comprometerse de algún otro modo.

Una solución común para garantizar los sistemas informáticos es proporcionar de manera funcional un nombre de usuario y contraseña. Sin embargo, la gestión de contraseñas ha demostrado ser bastante costosa con un gran porcentaje de las llamadas al servicio de ayuda relacionadas con los problemas de contraseñas. Por otra parte, las contraseñas proporcionan poca seguridad en cuanto a que, en general, se almacenan en un archivo susceptible a un acceso inadecuado, a través de, por ejemplo, los ataques de fuerza bruta.

Otra solución para garantizar los sistemas informáticos es proporcionar unas infraestructuras de cifrado. La criptografía, en general, se refiere a la protección de datos mediante la transformación o el cifrado de los mismos a un formato ilegible. Solo aquellos que poseen la clave(s) del cifrado pueden descifrar los datos en un formato utilizable. La criptografía se usa para identificar usuarios, por ejemplo, la autenticación, para permitir privilegios de acceso, por ejemplo, la autorización, para crear certificados y firmas digitales, y similares. Un sistema de criptografía popular es un sistema de clave pública que usa dos claves, una clave pública conocida por todos y una clave privada que solo conoce el propietario individual o negocio de la misma. En general, los datos cifrados con una clave se descifran con la otra y la clave no puede recrearse a partir de la otra.

Desafortunadamente, incluso los sistemas criptográficos de clave pública típicos anteriores son aún muy dependientes del usuario para la seguridad. Por ejemplo, los sistemas criptográficos entregan la clave privada al usuario, por ejemplo, a través del navegador del usuario. A continuación, los usuarios no sofisticados, en general, almacenan la clave privada en un disco duro accesible a los demás a través de un sistema informático abierto, como, por ejemplo, Internet. Por otro lado, los usuarios pueden escoger nombres pobres para los archivos que contienen su clave privada, como, por ejemplo “clave”. El resultado de lo anterior y otros actos es permitir que la clave o las claves sean susceptibles de comprometerse.

Además de los compromisos anteriores, un usuario puede guardar su clave privada en un sistema informático configurado con un sistema de copia de seguridad o archivado, dando como resultado potencialmente en copias de la clave privada que viaja a través de múltiples dispositivos de almacenamiento informáticos u otros sistemas. Esta brecha de seguridad se denomina a menudo como “migración de clave”. Al igual que en la migración de clave, muchas aplicaciones proporcionan acceso a la clave privada de un usuario a través de, a lo sumo, un simple acceso de nombre de usuario y contraseña. Como se ha mencionado anteriormente, el acceso de nombre de usuario y contraseña a menudo no proporciona una seguridad adecuada.

Una solución para aumentar la seguridad de los sistemas criptográficos anteriores es incluir la biometría como parte de la autenticación o autorización. La biometría, en general, incluye unas características físicas medibles, tales como, por ejemplo, las huellas dactilares o la voz que pueden comprobarse mediante un sistema automatizado, como, por ejemplo, la coincidencia de patrones o el reconocimiento de patrones de huellas digitales o patrones de voz. En tales sistemas, los datos biométricos y/o las claves de un usuario pueden almacenarse en los dispositivos informáticos móviles, tales como, por ejemplo, una tarjeta inteligente, un ordenador portátil, un asistente digital personal o un teléfono móvil, permitiendo de este modo que los datos biométricos o las claves puedan usarse en un ambiente móvil.

El sistema criptográfico biométrico móvil anterior sufre todavía de una variedad de inconvenientes. Por ejemplo, el usuario móvil puede perder o romper la tarjeta inteligente o el dispositivo informático portátil, teniendo de este modo

cortado enteramente su acceso a los datos potencialmente importantes. Como alternativa, una persona con malas intenciones puede robar la tarjeta inteligente móvil o el dispositivo informático portátil del usuario y usarlo para robar efectivamente las credenciales digitales del usuario móvil. Por otra parte, el dispositivo informático portátil puede estar conectado a un sistema abierto, tal como Internet, y, como con las contraseñas, el archivo en el que se almacenan los datos biométricos puede ser susceptible de comprometerse a través de la falta de atención del usuario para la seguridad o los intrusos maliciosos.

Una forma de proteger los datos contra el acceso no autorizado o el uso no autorizado es usar un esquema de compartición de secretos. Un esquema de compartición de secretos es un método para dividir una pieza sensible de los datos (por ejemplo, los archivos confidenciales, una clave de cifrado, o cualquier tipo de comunicación), a veces llamado el secreto, en una recopilación de piezas, llamadas comparticiones, de tal manera que la posesión de un número suficiente de comparticiones permite la recuperación del secreto, pero la posesión de un número insuficiente de comparticiones proporciona poca o ninguna información sobre el secreto que se ha compartido. Estos esquemas son herramientas importantes en la criptografía y la seguridad de la información.

Formalmente, un esquema de compartición de secretos consiste en un par de algoritmos, el algoritmo de compartición *Compartir* y el algoritmo de recuperación *Recuperar*. El algoritmo de compartición es normalmente probabilístico (lo que significa que hace elecciones aleatorias), y el algoritmo de recuperación es normalmente determinístico. El algoritmo de compartición puede usarse para desmontar o dividir, el secreto en una recopilación de comparticiones, y el algoritmo de recuperación puede usarse para volver a montar dichas comparticiones. A la hora del montaje, cada compartición puede estar presente, en cuyo caso puede proporcionarse una cadena al algoritmo de recuperación, o es posible que falte una compartición, en cuyo caso puede suministrarse un valor designado (denominado como, (“◊”) en el presente documento) al algoritmo de recuperación. Un conjunto de actores que está autorizado a recuperar el secreto se llama conjunto autorizado, y el conjunto de todos estos actores a veces se llama estructura de acceso.

Los esquemas de compartición de secretos se han diseñado para trabajar en diversas estructuras de acceso, pero la estructura de acceso más común es una estructura de acceso umbral, en la que cualquier subconjunto de m o más actores, de un total de n actores en total, se dice que están autorizados. Un esquema de compartición de secretos de una estructura de acceso umbral a veces se llama esquema umbral. Hay dos propiedades de seguridad para cualquier esquema de compartición de secretos: una propiedad de privacidad y una propiedad de recuperabilidad. La propiedad de privacidad garantiza que las coaliciones no autorizadas de actores no aprenden nada útil sobre el secreto. La propiedad de recuperabilidad garantiza que las coaliciones autorizadas de actores en última instancia, pueden recuperar el secreto subyacente.

El esquema de compartición de secretos de Shamir se dice que es un esquema de compartición de secretos perfecto (PSS). El término “perfecto” se refiere a que la garantía de privacidad es una información teórica y sin ningún error; por lo tanto, las coaliciones no autorizadas de actores no pueden aprender nada útil sobre el secreto subyacente en los esquemas de PSS.

Una limitación con los esquemas de PSS es que el tamaño de cada compartición debe ser al menos tan largo como el tamaño del secreto que se comparte. Sin embargo, cuando el secreto incluye un archivo grande o una cadena larga de caracteres, esta limitación puede llegar a ser difícil de manejar, aumentando la complejidad global del sistema. En respuesta a esta limitación, se han desarrollado los esquemas para la compartición de secretos de cálculo (CSS).

Por ejemplo, el esquema CSS de Krawczyk permite que las comparticiones sean más cortas que el secreto. Por ejemplo, en un esquema umbral de 2 de 3 (lo que significa que dos de tres comparticiones son adecuadas para recuperar el secreto), el secreto S puede dividirse en comparticiones de un tamaño aproximadamente de |S|/2 bits, en la que |S| indica la longitud de S. Estas comparticiones cortas no son posibles en la configuración de PSS. Sin embargo, en los esquemas de CSS la propiedad de privacidad ya no puede ser una información absoluta y teórica; más bien, una coalición no autorizada de actores puede obtener una pequeña cantidad de información sobre el secreto compartido a partir de sus comparticiones. Pero, bajo un supuesto de complejidad de cálculo, la cantidad de información será insignificante y por lo tanto, en la práctica, no es una gran preocupación.

Una segunda limitación de los esquemas de PSS se refiere a la falta de robustez encomendada. La robustez significa que un participante defectuoso o contradictorio es incapaz de obligar a la recuperación de un secreto incorrecto. El modelo para PSS supone que cada compartición es o “correcto” o “perdido”, pero nunca puede ser incorrecto (por ejemplo, dañado o intencionadamente alterado). En la práctica, esta suposición es muy irrazonable porque las comparticiones pueden ser erróneas debido a una serie de factores, entre ellos, por ejemplo, los errores en el almacenamiento, el ruido en un canal de comunicaciones, o debido a las actividades genuinamente contradictorias. Además, la falta de robustez no es solo una posibilidad teórica, sino un problema genuino para los esquemas de PSS normales, incluyendo el esquema de compartición de secretos de Shamir. Con el esquema de Shamir, un adversario puede forzar de hecho la recuperación de cualquier secreto deseado cambiando de manera apropiada solo una compartición. Las aplicaciones prácticas de los esquemas de compartición de secretos normalmente necesitan robustez.

Sumario de la invención

Basándose en lo anterior, se necesitan unos sistemas de compartición de secretos de cálculos robustos que sean a la vez eficientes y tengan unas fuertes propiedades de seguridad demostrables bajo unos supuestos criptográficos débiles.

En consecuencia, un aspecto de la presente invención es proporcionar un método para garantizar de manera virtual cualquier tipo de datos de un acceso o uso no autorizado. El método comprende una o más etapas de analizar, dividir y/o separar los datos a garantizarse en dos o más partes o porciones. El método también comprende encriptar los datos a garantizarse. El cifrado de los datos puede realizarse antes o después del primer análisis, división y/o separación de los datos. Además, la etapa de cifrado puede repetirse para una o más porciones de los datos. Del mismo modo, las etapas de análisis, división y/o separación pueden repetirse para una o más porciones de los datos. El método también comprende de manera opcional almacenar los datos analizados, divididos y/o separados que se han cifrado en una localización o en múltiples localizaciones. Este método también comprende de manera opcional reconstituir o volver a montar los datos protegidos en su forma original para el acceso o uso autorizado. Este método puede incorporarse en las operaciones de cualquier ordenador, servidor, motor o similares, que sea capaz de ejecutar las etapas deseadas del método.

Otro aspecto de la presente invención proporciona un sistema para garantizar de manera virtual cualquier tipo de datos de un acceso o uso no autorizado. Este sistema comprende un módulo de división de datos, un módulo de manipulación criptográfica, y, de manera opcional, un módulo de ensamble de datos. El sistema puede, en una realización, comprender además, una o más instalaciones de almacenamiento de datos en las que pueden almacenarse datos seguros.

Otro aspecto de la invención incluye usar cualquier algoritmo de análisis y de división adecuado para generar comparticiones de datos. Cualquier combinación aleatoria, pseudo-aleatoria, determinista, de los mismos puede emplearse para analizar y dividir los datos.

En otras realizaciones más, se proporciona un esquema de compartición de secretos de n partes con un espacio de mensajes S. Puede definirse una familia de adversarios, A. El esquema de compartición de secretos de n partes puede incluir una o más de las siguientes cinco primitivas: (1) un algoritmo de cifrado simétrico con claves de k bits y un espacio de mensaje S; (2) un algoritmo de PSS de n partes sobre unos adversarios A con un espacio de mensaje $\{0,1\}^k$; (3) un algoritmo de dispersión de información de n partes (IDA); (4) un código de corrección de errores de n partes (ECC) sobre unos adversarios A con un espacio de mensaje $\{0,1\}^h$; y (5) un esquema de vinculación aleatorio (o probabilístico). Los datos pueden garantizarse aplicando primero un algoritmo de compartición de secretos de cálculo para los datos a garantizarse. A continuación, puede generarse un valor aleatorio o pseudo-aleatorio. A partir de la salida del algoritmo de compartición de secretos y el valor aleatorio o pseudo-aleatorio, puede calcularse un conjunto de valores de vinculación y de valores de no vinculación. A continuación, puede formarse una pluralidad de comparticiones combinando una salida de compartición del algoritmo de compartición de secretos, un valor de no vinculación, y uno o más valores de vinculación. A continuación, las comparticiones pueden almacenarse en una o más localizaciones físicas (por ejemplo, en una unidad de disco duro magnético), o en una o más localizaciones geográficas (por ejemplo, diferentes depósitos de datos o servidores).

En algunas realizaciones, puede usarse un esquema de vinculación probabilística para calcular el conjunto de valores de vinculación y un conjunto de valores de no vinculación. Cada compartición puede definirse por una salida de compartición de un algoritmo de compartición de secretos de cálculo, un valor de no vinculación, y uno o más valores de vinculación del conjunto de valores de vinculación.

En algunas realizaciones, puede generarse y usarse una clave de cifrado para cifrar los datos de usuario para crear una porción de texto cifrado. Un conjunto de n comparticiones de clave puede crearse aplicando un algoritmo de compartición de secretos a la clave de cifrado. A continuación, un conjunto de n fragmentos de texto cifrado puede crearse aplicando un algoritmo de dispersión de información (IDA) para el texto cifrado. Un conjunto de n valores de vinculación y de n valores de no vinculación puede calcularse aplicando un esquema de vinculación probabilística para cada una de las n comparticiones de clave y los fragmentos de texto cifrado. Pueden formarse N fragmentos de datos, en el que cada fragmento de datos puede ser una función de una compartición de clave, un texto cifrado, un valor de no vinculación, y uno o más valores de vinculación. Finalmente, los fragmentos de datos pueden almacenarse en uno o más dispositivos de almacenamiento lógicos (por ejemplo, n dispositivos de almacenamiento lógicos). Uno o más de estos dispositivos de almacenamiento lógicos pueden estar localizados en diferentes localizaciones geográficas o físicas. A continuación, los datos de usuario pueden reconstituirse combinando al menos un número predefinido de fragmentos de datos. En algunas realizaciones, pueden usarse diversos códigos de corrección de errores para proporcionar una recopilación adecuada de los valores de vinculación a cada actor.

Breve descripción de los dibujos

La presente invención se describe con más detalle a continuación en conexión con los dibujos adjuntos, que están destinados a ilustrar y no a limitar la invención, y en los que:

La figura 1 ilustra un diagrama de bloques de un sistema criptográfico, de acuerdo con los aspectos de una realización de la invención;
La figura 2 ilustra un diagrama de bloques del motor de confianza de la figura 1, de acuerdo con los aspectos de una realización de la invención;
La figura 3 ilustra un diagrama de bloques del motor de transacción de la figura 2, de acuerdo con los aspectos de una realización de la invención;
La figura 4 ilustra un diagrama de bloques del depositario de la figura 2, de acuerdo con los aspectos de una realización de la invención;
La figura 5 ilustra un diagrama de bloques del motor de autenticación de la figura 2, de acuerdo con los aspectos de una realización de la invención;
La figura 6 ilustra un diagrama de bloques del motor de cifrado de la figura 2, de acuerdo con los aspectos de una realización de la invención;
La figura 7 es un diagrama de bloques ilustrativo que representa la estructura global de un esquema de compartición de secretos de cálculo robusto (RCSS), de acuerdo con una realización de la invención;
La figura 8 ilustra el proceso de compartición de secretos, de acuerdo con una realización de la invención;
La figura 9 ilustra con mayor detalle las etapas de vinculación mostradas en la figura 8, de acuerdo con una realización de la invención;
La figura 10 ilustra el proceso de compartición basado en una abstracción diferente de la construcción de un esquema de RCSS a partir de un esquema de CSS y de un esquema de vinculación; y
La figura 11 ilustra con más detalle las etapas de verificación en el esquema de vinculación probabilística mostrado en la figura 10.

Descripción detallada de la invención

Un aspecto de la presente invención es proporcionar un sistema criptográfico en el que uno o más servidores seguros, o un motor de confianza, almacenan las claves de cifrado y los datos de autenticación de usuario. Los usuarios acceden a la funcionalidad de los sistemas criptográficos convencionales a través del acceso de red al motor de confianza, sin embargo, el motor de la confianza no divulga las claves reales y otros datos de autenticación y, por lo tanto, las claves y los datos permanecen seguros. Este almacenamiento centrado en el servidor de claves y de datos de autenticación mantiene la seguridad independiente del usuario, la portabilidad, la disponibilidad y la sencillez.

Debido a que los usuarios pueden confiar en el sistema criptográfico para realizar la autenticación de usuario y documentos y otras funciones de cifrado, puede incorporarse una amplia variedad de funcionalidades en el sistema. Por ejemplo, el proveedor del motor de confianza puede garantizar contra el acuerdo de repudio, por ejemplo, autenticando a los participantes del acuerdo, firmando de manera digital el acuerdo en nombre de o para los participantes, y almacenar un registro del acuerdo firmado de manera digital por cada participante. Además, el sistema criptográfico puede monitorizar los acuerdos y determinar la aplicación de diferentes grados de autenticación, a partir de, por ejemplo, el precio, el usuario, el proveedor, la localización geográfica, el lugar de uso, o similares.

Para facilitar una comprensión completa de la invención, el resto de la descripción detallada describe la invención con referencia a las figuras, en las que los elementos similares están referenciados con números similares completamente.

La figura 1 ilustra un diagrama de bloques de un sistema criptográfico 100, de acuerdo con los aspectos de una realización de la invención. Como se muestra en la figura 1, el sistema criptográfico 100 incluye un sistema de usuario 105, un motor de confianza 110, una autoridad de certificación 115, y un sistema de proveedor 120, que se comunica a través de un enlace de comunicaciones 125.

De acuerdo con una realización de la invención, el sistema de usuario 105 comprende un ordenador de fin general convencional que tiene uno o más microprocesadores, tales como, por ejemplo, un procesador basado en Intel. Además, el sistema de usuario 105 incluye un sistema operativo apropiado, tal como, por ejemplo, un sistema operativo capaz de incluir gráficos o ventanas, tales como Windows, Unix, Linux, o similares. Como se muestra en la figura 1, el sistema de usuario 105 puede incluir un dispositivo biométrico 107. El dispositivo biométrico 107 puede capturar de manera ventajosa unos datos biométricos del usuario y transferir los datos biométricos capturados al motor de confianza 110. De acuerdo con una realización de la invención, el dispositivo biométrico puede comprender de manera ventajosa un dispositivo que tenga atributos y características similares a los divulgados en la solicitud de patente de Estados Unidos N.º 08/926.277, presentada el 5 de septiembre de 1997, titulada “ “RELIEF OBJECT IMAGE GENERATOR”, en la solicitud de patente de Estados Unidos N.º 09/558.634, presentada el 26 de abril de 2000, titulada “IMAGING DEVICE FOR A RELIEF OBJECT AND SYSTEM AND METHOD OF USING THE IMAGE DEVICE”, en la solicitud de Patente de Estados Unidos N.º 09/435.011, presentada el 5 de noviembre de 1999, titulada “RELIEF OBJECT SENSOR ADAPTOR”, y en la solicitud de patente de Estados Unidos N.º 09/477.943, presentada el 5 de enero del año 2000, titulada “PLANAR OPTICAL IMAGE SENSOR AND SYSTEM FOR GENERATING AN ELECTRONIC IMAGE OF A RELIEF OBJECT FOR FINGER-PRINT READING”, todas las cuales son propiedad de la cesionaria presente.

Además, el sistema de usuario 105 puede conectarse al enlace de comunicaciones 125 a través de un proveedor de servicios convencional, tal como, por ejemplo, una conexión telefónica, una línea de abonado digital (DSL), un módem por cable, una conexión de fibra, o similares. De acuerdo con otra realización, el sistema de usuario 105 se conecta al enlace de comunicaciones 125 a través de una conectividad de red, tal como, por ejemplo, una red de área local o amplia. De acuerdo con una realización, el sistema operativo incluye una pila TCP/IP que maneja todo el tráfico de mensajes entrantes y salientes que pasa a través del enlace de comunicaciones 125.

Aunque el sistema de usuario 105 se divulga con referencia a las realizaciones anteriores, la invención no está destinada a limitarse por las mismas. Más bien, un experto en la materia reconocerá a partir de la divulgación en el presente documento, un gran número de realizaciones alternativas del sistema de usuario 105, incluyendo casi cualquier dispositivo informático capaz de enviar o recibir información desde otro sistema informático. Por ejemplo, el sistema de usuario 105 puede incluir, pero no se limitan a, una estación de trabajo informática, una televisión interactiva, un kiosco interactivo, un dispositivo de informatización móvil personal, como por ejemplo, un asistente digital, un teléfono móvil, un ordenador portátil, o similares, un dispositivo de comunicaciones inalámbricas, una tarjeta inteligente, un dispositivo informático integrado, o similares, que pueden interactuar con el enlace de comunicaciones 125. En estos sistemas alternativos, los sistemas operativos son probablemente diferentes y se adaptan al dispositivo específico. Sin embargo, de acuerdo con una realización, los sistemas operativos siguen proporcionando de manera ventajosa los protocolos de comunicaciones apropiados necesarios para establecer la comunicación con el enlace de comunicaciones 125.

La figura 1 ilustra el motor de confianza 110. De acuerdo con una realización, el motor de confianza 110 comprende uno o más servidores seguros para acceder y almacenar la información sensible, que puede ser cualquier tipo o forma de datos, tales como, pero no limitado a texto, audio, video, datos de autenticación de usuario y claves de cifrado públicas y privadas. De acuerdo con una realización, los datos de autenticación incluyen datos diseñados para identificar de manera única a un usuario del sistema criptográfico 100. Por ejemplo, los datos de autenticación pueden incluir un número de identificación de usuario, uno o más datos biométricos, y una serie de preguntas y respuestas generadas por el motor de confianza 110 o el usuario, pero respondidas inicialmente por el usuario en el registro. Las preguntas anteriores pueden incluir datos demográficos, tales como el lugar de nacimiento, la dirección, el aniversario, o similares, datos personales, tales como el nombre de soltera de la madre, el helado favorito, o similares, u otros datos diseñados para identificar de manera unívoca al usuario. El motor de confianza 110 compara los datos de autenticación del usuario asociados con una transacción en curso, con los datos de autenticación proporcionados en un momento anterior tales como, por ejemplo, durante el registro. El motor de confianza 110 puede necesitar de manera ventajosa que el usuario produzca los datos de autenticación en el momento de cada transacción, o, el motor de confianza 110 puede permitir de manera ventajosa que el usuario produzca de manera periódica los datos de autenticación, tal como en el principio de una cadena de transacciones o al iniciar sesión en un sitio web de proveedor específico.

De acuerdo con la realización en la que el usuario produce unos datos biométricos, el usuario proporciona una característica física; tal como, pero no limitado a, una exploración facial, una exploración de mano, una exploración de oído, una exploración del iris, una exploración de retina, el patrón vascular, el ADN, una huella digital, la escritura o el habla, al dispositivo biométrico 107. El dispositivo biométrico produce de manera ventajosa un patrón electrónico, o biométrico, de la característica física. El patrón electrónico se transfiere a través del sistema de usuario 105 al motor de confianza 110, o para fines de autenticación o de registro.

Una vez que el usuario produce los datos de autenticación adecuados y que el motor de confianza 110 determina una coincidencia positiva entre esos datos de autenticación (los datos de autenticación actuales) y los datos de autenticación proporcionados en el momento del registro (los datos de autenticación de registro), el motor de confianza 110 proporciona al usuario la funcionalidad de cifrado completa. Por ejemplo, el usuario autenticado de manera correcta puede emplear de manera ventajosa el motor de confianza 110 para realizar un cálculo de clave, una firma digital, un cifrado y un descifrado (a menudo denominado en conjunto solo como el cifrado), la creación o la distribución de certificados digitales, y similares. Sin embargo, las claves de cifrado privadas usadas en las funciones de cifrado no estarán disponibles fuera del motor de confianza 110, garantizando de este modo la integridad de las claves de cifrado.

De acuerdo con una realización, el motor de confianza 110 genera y almacena las claves de cifrado. De acuerdo con otra realización, al menos una clave de cifrado está asociada con cada usuario. Por otra parte, cuando las claves de cifrado incluyen la tecnología de clave pública, cada clave privada asociada con un usuario se genera dentro de, y no se libera de, el motor de confianza 110. Por lo tanto, siempre que el usuario tiene acceso al motor de confianza 110, el usuario puede realizar las funciones de cifrado usando su clave privada o pública. Este acceso remoto permite de manera ventajosa que los usuarios permanezcan completamente móviles y accedan a la funcionalidad de cifrado a través de prácticamente cualquier conexión a Internet, tales como los teléfonos móviles y satelitales, los kioscos, los ordenadores portátiles, las habitaciones de hotel y similares.

De acuerdo con otra realización, el motor de confianza 110 realiza la funcionalidad de cifrado usando un par de claves generadas por el motor de confianza 110. De acuerdo con esta realización, el motor de confianza 110 primero autentica al usuario, y después de que el usuario haya producido de manera correcta que los datos de autenticación

coincidan con los datos de autenticación registrados, el motor de confianza 110 usa su propio par de claves de cifrado para realizar las funciones de cifrado en nombre del usuario autenticado.

Un experto en la materia reconocerá a partir de la divulgación en el presente documento que las claves de cifrado pueden incluir de manera ventajosa algunas o todas las claves simétricas, las claves públicas y las claves privadas. Además, un experto en la materia reconocerá a partir de la divulgación en el presente documento que las claves anteriores pueden implementarse con un amplio número de algoritmos disponibles a partir de tecnologías comerciales, tales como, por ejemplo, RSA, ELGAMAL, o similares.

La figura 1 ilustra también la autoridad de certificación 115. De acuerdo con una realización, la autoridad de certificación 115 puede comprender de manera ventajosa una organización o empresa de terceros de confianza que emite los certificados digitales, tales como, por ejemplo, VeriSign, Baltimore, Entrust, o similares. El motor de confianza 110 puede transmitir de manera ventajosa las solicitudes de certificados digitales, a través de uno o más protocolos de certificados digitales convencionales, tales como, por ejemplo, PKCS10, a la autoridad de certificación 115. En respuesta, la autoridad de certificación 115 emitirá un certificado digital en uno o más de un número de diferentes protocolos, tales como, por ejemplo, PKCS7. De acuerdo con una realización de la invención, el motor de confianza 110 solicita los certificados digitales de varias o todas las autoridades de certificación prominentes 115 de tal manera que el motor de confianza 110 tiene acceso a un certificado digital correspondiente a la norma de certificación de cualquier parte solicitante.

De acuerdo con otra realización, el motor de confianza 110 realiza internamente emisiones de certificados. En esta realización, el motor de confianza 110 puede acceder a un sistema de certificados para generar los certificados y/o puede generar internamente los certificados cuando se solicitan, tal como, por ejemplo, en el momento de la generación de las claves o en el certificado convencional solicitado en el momento de la solicitud. El motor de confianza 110 se divulgará en mayor detalle a continuación.

La figura 1 también ilustra el sistema de proveedor 120. De acuerdo con una realización, el sistema de proveedor 120 comprende de manera ventajosa un servidor web. Los servidores de web normales, en general, sirven contenido a través de Internet usando uno de diversos lenguajes de marcado de Internet o las normas de formato de documento, tales como el lenguaje de marcado de hipertexto (HTML) o el lenguaje de marcado extensible (XML). El servidor web acepta las solicitudes de los navegadores como Netscape e Internet Explorer y, a continuación, devuelve los documentos electrónicos adecuados. Una serie de tecnologías del lado del servidor o del cliente pueden usarse para aumentar la potencia del servidor web más allá de su capacidad de entregar los documentos electrónicos convencionales. Por ejemplo, estas tecnologías incluyen las secuencias de comandos de interfaz de pasarela común (CGI), la capa de conexión segura (SSL) y las páginas de servidor activas (ASP). El sistema del proveedor 120 puede proporcionar de manera ventajosa contenidos electrónicos relativos a las transacciones comerciales, personales, educativas o de otro tipo.

Aunque el sistema del proveedor 120 se divulga con referencia a las realizaciones anteriores, la invención no está destinada a limitarse por las mismas. Más bien, un experto en la materia reconocerá a partir de la divulgación en el presente documento que el sistema de proveedor 120 puede comprender de manera ventajosa cualquiera de los dispositivos descritos con referencia al sistema de usuario 105 o a una combinación de los mismos.

La figura 1 también ilustra el enlace de comunicaciones 125 que conecta el sistema de usuario 105, el motor de confianza 110, la autoridad de certificación 115, y el sistema de proveedor 120. De acuerdo con una realización, el enlace de comunicaciones 125 comprende preferentemente la Internet. La Internet, tal como se usa en esta divulgación es una red mundial de ordenadores. La estructura de la Internet, que es bien conocida por los expertos en la materia, incluye una estructura principal de red con unas redes que se ramifican desde la estructura principal. Estas ramificaciones, a su vez, tienen ramificaciones de redes a partir de las mismas, y así sucesivamente. Los routers mueven los paquetes de información entre los niveles de la red, y a continuación de una red a otra red, hasta que el paquete llega a la vecindad de su destino. Desde el destino, el host de la red de destino dirige el paquete de información al terminal apropiado, o nodo. En una realización ventajosa, los hubs de enrutamiento de Internet comprenden los servidores del sistema de nombres de dominio (DNS) que usan el protocolo de control de transmisión/protocolo de internet (TCP/IP) como es bien conocido en la técnica. Los hubs de enrutamiento se conectan a uno o más hubs de enrutamiento a través de los enlaces de comunicaciones de alta velocidad.

Una parte popular de la Internet es la World Wide Web. La World Wide Web contiene diferentes ordenadores, que almacenan los documentos capaces de visualizar información gráfica y textual. Los ordenadores que proporcionan información sobre la World Wide Web normalmente se denominan “sitios web”. Un sitio web está definido por una dirección de Internet que tiene una página electrónica asociada. La página electrónica puede identificarse por un localizador de recursos uniforme (URL). En general, una página electrónica es un documento que organiza la presentación del texto, de las imágenes gráficas, el audio, el vídeo, y así sucesivamente.

Aunque el enlace de comunicaciones 125 se divulga en términos de su realización preferida, un experto en la materia reconocerá a partir de la divulgación en el presente documento que el enlace de comunicaciones 125 puede incluir una amplia gama de enlaces de comunicaciones interactivos. Por ejemplo, el enlace de comunicaciones 125

puede incluir redes de televisión interactivas, redes telefónicas, sistemas de transmisión inalámbrica de datos, sistemas de cable de dos vías, redes informáticas privadas o públicas personalizadas, redes de kioscos interactivos, redes de cajeros automáticos, enlaces directos, redes por satélite o móviles, y similares.

La figura 2 ilustra un diagrama de bloques del motor de confianza 110 de la figura 1, de acuerdo con los aspectos de una realización de la invención. Como se muestra en la figura 2, el motor de confianza 110 incluye un motor de transacción 205, un depositario 210, un motor de autenticación 215, y un motor de cifrado 220. De acuerdo con una realización de la invención, el motor de confianza 110 incluye también un almacenamiento masivo 225. Como se muestra además en la figura 2, el motor de transacción 205 se comunica con el depositario 210, el motor de autenticación 215, y el motor de cifrado 220, junto con el almacenamiento masivo 225. Además, el depositario 210 se comunica con el motor de autenticación 215, el motor de cifrado 220, y el almacenamiento masivo 225. Por otra parte, el motor de autenticación 215 se comunica con el motor de cifrado 220. De acuerdo con una realización de la invención, algunas o todas de las comunicaciones anteriores pueden comprender de manera ventajosa la transmisión de documentos XML a direcciones IP que corresponden al dispositivo de recepción. Como se ha mencionado anteriormente, los documentos XML permiten de manera ventajosa a los diseñadores crear sus propias etiquetas de documentos personalizadas, lo que permite la definición, la transmisión, la validación y la interpretación de los datos entre las aplicaciones y entre las organizaciones. Además, algunas o todas de las comunicaciones anteriores pueden incluir las tecnologías SSL convencionales.

De acuerdo con una realización, el motor de transacción 205 comprende un dispositivo de enrutamiento de datos, tal como un servidor web convencional disponible de Netscape, Microsoft, Apache, o similares. Por ejemplo, el servidor web puede recibir de manera ventajosa unos datos de entrada desde el enlace de comunicaciones 125. De acuerdo con una realización de la invención, los datos de entrada se dirigen a un sistema de seguridad de front-end para el motor de confianza 110. Por ejemplo, el sistema de seguridad de front-end puede incluir de manera ventajosa un cortafuegos, un sistema de detección de intrusiones que busca los perfiles de ataque conocidos, y/o un escáner de virus. Después de limpiar el sistema de seguridad de front-end, los datos se reciben por el motor de transacción 205 y se enrutan a uno de entre el depositario 210, el motor de autenticación 215, el motor de cifrado 220, y el almacenamiento masivo 225. Además, el motor de transacción 205 monitoriza los datos de entrada del motor de autenticación 215 y del motor de cifrado 220, y enruta los datos hacia los sistemas específicos a través del enlace de comunicaciones 125. Por ejemplo, el motor de transacción 205 puede enrutar de manera ventajosa los datos hacia el sistema de usuario 105, la autoridad de certificación 115, o el sistema de proveedor 120.

De acuerdo con una realización, los datos se enrutan usando técnicas de enrutamiento HTTP convencionales, tales como, por ejemplo, las que emplean las URL o los indicadores de recursos uniforme (URI). Los URI son similares a las URL, sin embargo, los URI indican normalmente la fuente de los archivos o las acciones, tal como, por ejemplo, ejecutables, secuencias de comandos, y similares. Por lo tanto, de acuerdo con la una realización, el sistema de usuario 105, la autoridad de certificación 115, el sistema de proveedor 120, y los componentes del motor de confianza 210, incluyen de manera ventajosa datos suficientes dentro de las URL o los URI de comunicación al motor de transacción 205 para enrutar adecuadamente los datos en todo el sistema criptográfico.

Aunque el enrutamiento de datos se divulga con referencia a su realización preferida, un experto en la materia reconocerá un gran número de posibles soluciones o estrategias de enrutamiento de datos. Por ejemplo, XML u otros paquetes de datos pueden desempaquetarse y reconocerse de manera ventajosa por su formato, su contenido, o similares, de tal manera que el motor de transacción 205 puede enrutar datos adecuadamente a través del motor de confianza 110. Por otra parte, un experto en la materia reconocerá que el enrutamiento de datos puede adaptarse de manera ventajosa a los protocolos de transferencia de datos que se ajustan a los sistemas de red específicos, tales como, por ejemplo, cuando el enlace de comunicaciones 125 comprende una red local.

De acuerdo con otra realización más de la invención, el motor de transacción 205 incluye las tecnologías de cifrado SSL convencionales, de tal manera que los sistemas anteriores pueden autenticarse a sí mismos, y viceversa, con el motor de transacción 205, durante las comunicaciones específicas. Tal como se usa en esta divulgación, el término “SSL ½” se refiere a las comunicaciones en las que se autentica con SSL un servidor pero no necesariamente el cliente, y el término “SSL COMPLETA” se refiere a las comunicaciones en las que el cliente y el servidor se autentican con SSL. Cuando la presente divulgación usa el término “SSL”, la comunicación puede comprender una SSL ½ o COMPLETA.

A medida que el motor de transacción 205 enruta los datos a los diversos componentes del sistema criptográfico 100, el motor de transacción 205 pueden crear de manera ventajosa una pista de auditoría. De acuerdo con una realización, la pista de auditoría incluye un registro de al menos el tipo y el formato de los datos enrutados por el motor de transacción 205 en todo el sistema criptográfico 100. Estos datos de auditoría pueden almacenarse de manera ventajosa en el almacenamiento masivo 225.

La figura 2 ilustra también el depositario 210. De acuerdo con una realización, el depositario 210 comprende una o más instalaciones de almacenamiento de datos, tales como, por ejemplo, un servidor de directorio, un servidor de base de datos, o similares. Como se muestra en la figura 2, el depositario 210 almacena claves de cifrado y datos de autenticación de registro. Las claves de cifrado pueden corresponder de manera ventajosa al motor de confianza

110 o a los usuarios del sistema criptográfico 100, tal como el usuario o el proveedor. Los datos de autenticación de registro pueden incluir de manera ventajosa unos datos diseñados para identificar de manera única a un usuario, tal como por ejemplo, el ID de usuario, contraseñas, respuestas a preguntas, datos biométricos, o similares. Estos datos de autenticación de registro pueden adquirirse de manera ventajosa en el registro de un usuario o en otro momento posterior alternativo. Por ejemplo, el motor de confianza 110 puede incluir la renovación o la reedición de los datos de autenticación de registro.

De acuerdo con una realización, la comunicación del motor de transacción 205 hacia y desde el motor de autenticación 215 y el motor de cifrado 220 comprende una comunicación segura, tal como, por ejemplo la tecnología SSL convencional. Además, como se ha mencionado anteriormente, los datos de las comunicaciones hacia y desde el depositario 210 pueden transferirse usando las URL, los URI, HTTP o documentos XML, con cualquiera de los anteriores, teniendo de manera ventajosa unas solicitudes de datos y formatos integrados en los mismos.

Como se ha mencionado anteriormente, el depositario 210 puede comprender de manera ventajosa una pluralidad de instalaciones de almacenamiento de datos seguros. En tal realización, las instalaciones de almacenamiento de datos seguros pueden configurarse de tal manera que un compromiso de la seguridad en una instalación de almacenamiento de datos individual no comprometerá las claves de cifrado o los datos de autenticación almacenados en las mismas. Por ejemplo, de acuerdo con esta realización, las claves de cifrado y los datos de autenticación se operan de manera matemática con el fin de aleatorizar estadística y sustancialmente los datos almacenados en cada instalación de almacenamiento de datos. De acuerdo con una realización, la aleatorización de los datos de una instalación de almacenamiento de datos individual convierte a estos datos en indescifrables. Por lo tanto, el compromiso de una instalación de almacenamiento de datos individual produce solo un número aleatorio indescifrable y no compromete la seguridad de ninguna de las claves de cifrado o los datos de autenticación en su conjunto.

La figura 2 ilustra también el motor de confianza 110 que incluye el motor de autenticación 215. De acuerdo con una realización, el motor de autenticación 215 comprende un comparador de datos configurado para comparar los datos del motor de transacción 205 con los datos del depositario 210. Por ejemplo, durante la autenticación, un usuario suministra unos datos de autenticación actuales al motor de confianza 110 de tal manera que el motor de transacción 205 recibe los datos de autenticación actuales. Como se ha mencionado anteriormente, el motor de transacción 205 reconoce las solicitudes de datos, preferentemente en la URL o el URI, y enruta los datos de autenticación al motor de autenticación 215. Por otra parte, tras la solicitud, el depositario los 210 reenvía los datos de autenticación de registro que corresponden al usuario al motor de autenticación 215. De este modo, el motor de autenticación 215 tiene tanto los datos de autenticación actuales como los datos de autenticación de registro para su comparación.

De acuerdo con una realización, las comunicaciones hacia el motor de autenticación comprenden unas comunicaciones seguras, tales como, por ejemplo, la tecnología SSL. Además, la seguridad puede proporcionarse dentro de los componentes del motor de confianza 110, tales como, por ejemplo, el súper cifrado que usa tecnologías de clave pública. Por ejemplo, de acuerdo con una realización, el usuario cifra los datos de autenticación actuales con la clave pública del motor de autenticación 215. Además, el depositario 210 también cifra los datos de autenticación de registro con la clave pública del motor de autenticación 215. De esta manera, solamente la clave privada del motor de autenticación puede usarse para descifrar las transmisiones.

Como se muestra en la figura 2, el motor de confianza 110 incluye también el motor de cifrado 220. De acuerdo con una realización, el motor de cifrado comprende un módulo de manipulación criptográfica, configurado para proporcionar de manera ventajosa unas funciones de cifrado convencionales, tales como, por ejemplo, una funcionalidad de infraestructura de clave pública (PKI). Por ejemplo, el motor de cifrado 220 puede emitir de manera ventajosa las claves públicas y privadas para los usuarios del sistema criptográfico 100. De esta manera, las claves de cifrado se generan en el motor de cifrado 220 y se reenviarán al depositario 210 de tal manera que al menos las claves de cifrado privadas no están disponibles fuera del motor de confianza 110. De acuerdo con otra realización, el motor de cifrado de 220 aleatoriza y divide al menos los datos de clave de cifrado privadas, almacenando de este modo solamente los datos divididos aleatoriamente. De igual manera que la división de los datos de autenticación de registro, el proceso de división garantiza que las claves almacenadas no están disponibles fuera el motor de cifrado 220. De acuerdo con otra realización, las funciones del motor de cifrado pueden combinarse con y realizarse por el motor de autenticación 215.

De acuerdo con una realización, las comunicaciones hacia y desde el motor de cifrado incluyen unas comunicaciones seguras, tal como la tecnología SSL. Además, los documentos XML pueden emplearse de manera ventajosa para transferir datos y/o hacer solicitudes de función de cifrado.

La figura 2 ilustra también el motor de confianza 110 que tiene el almacenamiento masivo 225. Como se ha mencionado anteriormente, el motor de transacción 205 mantiene los datos que corresponden a una pista de auditoría y almacena estos datos en el almacenamiento masivo 225. Del mismo modo, de acuerdo con una realización de la invención, el depositario 210 mantiene los datos que corresponden a una pista de auditoría y

almacena estos datos en el dispositivo de almacenamiento masivo 225. Los datos de pista de auditoría de depositario son similares a los del motor de transacción 205 en que los datos de pista de auditoría comprenden un registro de las solicitudes recibidas por el depositario 210 y la respuesta a las mismas. Además, el almacenamiento masivo 225 puede usarse para almacenar certificados digitales que tienen la clave pública de un usuario contenida en los mismos.

Aunque el motor de confianza 110 se divulga con referencia a sus realizaciones preferidas y alternativas, la invención no está destinada a limitarse por las mismas. Más bien, un experto en la materia reconocerá en la divulgación en el presente documento, un gran número de alternativas para el motor de confianza 110. Por ejemplo, el motor de confianza 110, puede realizar de manera ventajosa solo la autenticación, o como alternativa, solo algunas o la totalidad de las funciones de cifrado, tales como el cifrado y el descifrado de los datos. De acuerdo con estas realizaciones, uno de los motores de autenticación 215 y el motor de cifrado 220 pueden eliminarse de manera ventajosa, creando de este modo un diseño más sencillo del motor de confianza 110. Además, el motor de cifrado 220 también puede comunicarse con una autoridad de certificación de tal manera que la autoridad de certificación está incorporada en el motor de confianza 110. De acuerdo con otra realización más, el motor de confianza 110 puede realizar de manera ventajosa la autenticación y una o más funciones de cifrado, tales como, por ejemplo, la firma digital.

La figura 3 ilustra un diagrama de bloques del motor de transacción 205 de la figura 2, de acuerdo con los aspectos de una realización de la invención. De acuerdo con esta realización, el motor de transacción 205 comprende un sistema operativo 305 que tiene un subproceso de manipulación y un subproceso de escucha. El sistema operativo 305 puede ser de manera ventajosa similar a los encontrados en los servidores convencionales de gran volumen, tales como, por ejemplo, servidores web disponibles de Apache. El subproceso de escucha monitoriza la comunicación entrante de uno de entre el enlace de comunicaciones 125, el motor de autenticación 215, y el motor de cifrado 220 para el flujo de datos entrante. El subproceso de manipulación reconoce las estructuras de datos específicas del flujo de datos entrante, tal como, por ejemplo, las estructuras de datos precedentes, enrutando de este modo los datos de entrada a uno de entre el enlace de comunicaciones 125, el depositario 210, el motor de autenticación 215, el motor de cifrado 220, o el almacenamiento masivo 225. Como se muestra en la figura 3, los datos entrantes y salientes pueden garantizarse de manera ventajosa a través de, por ejemplo, la tecnología SSL.

La figura 4 ilustra un diagrama de bloques del depositario 210 de la figura 2 de acuerdo con los aspectos de una realización de la invención. De acuerdo con esta realización, el depositario 210 comprende uno o más servidores de protocolo ligero de acceso a directorios (LDAP). Los servidores de directorio LDAP están disponibles en una amplia variedad de fabricantes tales como Netscape, ISO, y otros. La figura 4 muestra también que el servidor de directorio almacena preferentemente los datos 405 correspondientes a las claves de cifrado y los datos 410 correspondientes a los datos de autenticación de registro. De acuerdo con una realización, el depositario 210 comprende una única estructura de memoria lógica que indexa los datos de autenticación y los datos de clave de cifrado para un único ID de usuario. La única estructura de memoria lógica incluye preferentemente unos mecanismos para garantizar un alto grado de confianza o seguridad, en los datos almacenados en la misma. Por ejemplo, la localización física del depositario 210 puede incluir de manera ventajosa un amplio número de medidas de seguridad convencionales, tales como el acceso limitado de empleados, los sistemas de vigilancia modernos, y similares. Además de, o en lugar de, las garantías físicas, el sistema o servidor informático puede incluir de manera ventajosa unas soluciones de software para proteger los datos almacenados. Por ejemplo, el depositario 210 puede crear y almacenar los datos 415 de manera ventajosa que corresponden a una pista de auditoría de las acciones tomadas. Además, las comunicaciones entrantes y salientes pueden cifrarse de manera ventajosa con el cifrado de clave pública junto con las tecnologías SSL convencionales.

De acuerdo con otra realización, el depositario 210 puede comprender unas instalaciones de almacenamiento de datos distintas y físicamente separadas, como se divulga adicionalmente con referencia a la figura 7.

La figura 5 ilustra un diagrama de bloques del motor de autenticación 215 de la figura 2, de acuerdo con los aspectos de una realización de la invención. Al igual que en el motor de transacción 205 de la figura 3, el motor de autenticación 215 comprende un sistema operativo 505 que tiene al menos un subproceso de escucha y un subproceso de manipulación de una versión modificada de un servidor web convencional, tal como, por ejemplo, los servidores web disponibles de Apache. Como se muestra en la figura 5, el motor de autenticación 215 incluye el acceso a al menos una clave privada 510. La clave privada 510 puede usarse de manera ventajosa, por ejemplo, para descifrar los datos del motor de transacción 205 o del depositario 210, que se han cifrado con una clave pública correspondiente del motor de autenticación 215.

La figura 5 ilustra también el motor de autenticación 215 que comprende un comparador 515, un módulo de división de datos 520, y un módulo de ensamblaje de datos 525. De acuerdo con la realización preferida de la invención, el comparador 515 incluye una tecnología capaz de comparar los patrones potencialmente complejos relacionados con los datos de autenticación biométricos anteriores. La tecnología puede incluir hardware, software, o soluciones combinadas para las comparaciones de patrones, tales como, por ejemplo, los que representan los patrones de huellas digitales o los patrones de voz. Además, de acuerdo con una realización, el comparador 515 del motor de autenticación 215 puede comparar de manera ventajosa cálculos de clave convencionales de documentos con el fin

de hacer un resultado de comparación. De acuerdo con una realización de la invención, el comparador 515 incluye la aplicación de la heurística 530 para la comparación. La heurística 530 puede abordar de manera ventajosa las circunstancias que rodean un intento de autenticación, tales como, por ejemplo, la hora del día, la dirección IP o la máscara de subred, el perfil de compra, la dirección de correo electrónico, el número de serie o el ID del procesador, o similares.

Por otra parte, la naturaleza de las comparaciones de datos biométricos puede resultar en un grado de variación de la confianza que se produce a partir de la coincidencia de los datos de autenticación biométricos actuales con los datos de registro. Por ejemplo, a diferencia de una contraseña tradicional, que solo puede devolver una coincidencia positiva o negativa, una huella digital puede determinarse para que sea una coincidencia parcial, por ejemplo, una coincidencia del 90 %, una coincidencia del 75 %, o una coincidencia del 10 %, en lugar de simplemente que sea correcta o incorrecta. Otros identificadores biométricos tales como el análisis de impresión de voz o el reconocimiento de rostros pueden compartir esta característica de autenticación probabilística, en lugar de una autenticación absoluta.

Cuando se trabaja con esta autenticación probabilística o en otros casos en los que una autenticación se considera menos que absolutamente fiable, es deseable aplicar la heurística 530 para determinar si el nivel de confianza proporcionado en la autenticación es suficientemente alto como para autentificar la transacción que se está realizando.

A veces será el caso de que la transacción en cuestión es una transacción de valor relativamente bajo, en la que es aceptable para autenticarse un menor nivel de confianza. Esto podría incluir una transacción que tiene un bajo valor en dólares asociado con la misma (por ejemplo, una compra de 10 $) o una transacción con bajo riesgo (por ejemplo, la admisión en un sitio web solo para miembros).

Por el contrario, para autenticar otras transacciones, puede ser deseable necesitar un alto grado de confianza en la autenticación antes de permitir que se produzca la transacción. Estas operaciones pueden incluir transacciones de un gran valor en dólares (por ejemplo, firmar un contrato de suministro de múltiples millones de dólares) o una transacción con un alto riesgo si se produce una autenticación irregular (por ejemplo, de manera remota iniciar sesión en un ordenador del gobierno).

El uso de la heurística 530 en combinación con los niveles de confianza y los valores de transacciones pueden usarse como se describirá a continuación para permitir que el comparador proporcione un sistema de autenticación dinámico sensible al contexto.

De acuerdo con otra realización de la invención, el comparador 515 puede realizar un seguimiento de los intentos de autenticación de manera ventajosa para una transacción específica. Por ejemplo, cuando una transacción falla, el motor de confianza 110 puede solicitar al usuario que vuelva a introducir sus datos de autenticación actuales. El comparador 515 del motor de autenticación 215 puede emplear de manera ventajosa un limitador de intentos 535 para limitar el número de intentos de autenticación, prohibiendo de este modo los intentos de fuerza bruta para suplantar a los datos de autenticación de un usuario. De acuerdo con una realización, el limitador de intentos 535 comprende un módulo de software que monitoriza las transacciones para repetir los intentos de autenticación y, por ejemplo, limitar los intentos de autenticación a tres para una transacción dada. De este modo, el limitador de intentos 535 limitará un intento automatizado para suplantar a los datos de autenticación de un usuario para, por ejemplo, simplemente a tres “conjeturas”. Tras tres fallos, el limitador de intentos 535 puede denegar de manera ventajosa los intentos de autenticación adicionales. Esta negación puede implementarse de manera ventajosa a través de, por ejemplo, el comparador 515 que devuelve un resultado negativo, independientemente de los datos de autenticación actuales que se transmiten. Por otra parte, el motor de transacción 205 puede bloquear de manera ventajosa cualquier intento de autenticación adicional relativo a una transacción en la que tres intentos han fallado anteriormente.

El motor de autenticación 215 incluye también el módulo de división de datos 520 y el módulo de ensamblaje de datos 525. El módulo de división de datos 520 comprende de manera ventajosa un software, un hardware, o un módulo de combinación que tiene la capacidad de operar de manera matemática con diversos datos con el fin de aleatorizar y dividir sustancialmente los datos en porciones. De acuerdo con una realización, los datos originales no pueden recrearse a partir de una porción individual. El módulo de ensamblaje de datos 525 comprende de manera ventajosa un software, un hardware, o un módulo de combinación configurado para operar de manera matemática en las porciones sustancialmente aleatorias anteriores, de tal manera que la combinación de las mismas proporciona los datos descifrados originales. De acuerdo con una realización, el motor de autenticación 215 emplea el módulo de división de datos 520 para aleatorizar y dividir los datos de autenticación de registro de división en porciones, y emplea el módulo de ensamblaje de datos 525 para reensamblar las porciones en los datos de autenticación de registro utilizables.

La figura 6 ilustra un diagrama de bloques del motor de cifrado 220 del motor de confianza 200 de la figura 2 de acuerdo con los aspectos de una realización de la invención. Al igual que en el motor de transacción 205 de la figura 3, el motor de cifrado 220 comprende un sistema operativo 605 que tiene al menos un subproceso de escucha y un

subproceso de manipulación de una versión modificada de un servidor web convencional, tal como, por ejemplo, los servidores web disponibles de Apache. Como se muestra en la figura 6, el motor de cifrado 220 comprende un módulo de división de datos 610 y un módulo de ensamblaje de datos 620 con una función similar a los de la figura 5. Sin embargo, de acuerdo con una realización, el módulo de división de datos 610 y el módulo de ensamblaje de datos 620 procesan los datos de clave de cifrado, en oposición a los datos de autenticación de registro anteriores. Aunque, un experto en la materia reconocerá a partir de la divulgación en el presente documento que el módulo de división de datos 910 y el módulo de división de datos 620 pueden combinarse con los del motor de autenticación 215.

El motor de cifrado 220 comprende también un módulo de manipulación criptográfica 625 configurado para realizar una, algunas o la totalidad de un gran número de funciones de cifrado. De acuerdo con una realización, el módulo de manipulación criptográfica 625 puede comprender unos módulos o programas de software, hardware, o ambos. De acuerdo con otra realización, el módulo de manipulación criptográfica 625 puede realizar comparaciones de datos, análisis de datos, división de datos, separación de datos, cálculos de clave de datos, cifrado o descifrado de datos, verificación o creación de firmas digitales, generación de certificados digitales, almacenamiento o solicitudes, generación de claves de cifrado, o similares. Por otra parte, un experto en la materia reconocerá a partir de la divulgación en el presente documento que el módulo de manipulación criptográfica 825 puede comprender de manera ventajosa una infraestructura de clave pública, tal como la Pretty Good Privacy (privacidad bastante buena) (PGP), un sistema de clave pública basado en RSA, o un gran número de sistemas de gestión de claves alternativas. Además, el módulo de manipulación criptográfica 625 puede realizar un cifrado de clave pública, un cifrado de clave simétrica, o ambos. Además de lo anterior, el módulo de manipulación criptográfica 625 puede incluir uno o más programas o módulos informáticos, hardware, o ambos, para implementar unas funciones de interoperabilidad sin fisuras y transparentes.

Un experto en la materia reconocerá también a partir de la divulgación en el presente documento que la funcionalidad de cifrado puede incluir un gran número o variedad de funciones, en general, relacionadas con los sistemas de gestión de claves de cifrado.

Un esquema de compartición de secretos de cálculo robusto (RCSS) se ilustra en la figura 7. Un parte denominada como el distribuidor 700 tiene un secreto 701 que el distribuidor desea distribuir. Con este fin, el distribuidor 700 puede aplicar el mecanismo de compartición de un esquema de RCSS 702. El mecanismo de compartición 702 puede dar como resultado algún número, n, de las comparticiones que se generan, como se indica por las comparticiones 704, 705, y 706. 703. La recopilación de todas las comparticiones puede ser un vector S derivado de manera probabilística del secreto 701. A continuación, la recopilación 703 de las comparticiones puede enviarse a través de una red o distribuirse fuera de banda, de tal manera que cada compartición se almacena en su propio depósito de datos (o en diferentes localizaciones físicas o geográficas en uno o más depósitos de datos). El almacenamiento de las comparticiones en un depósito de datos lógico 720 puede tener el beneficio de una mayor seguridad, por que puede ser más difícil para un adversario obtener un acceso a todas las comparticiones que pueden estar almacenados en los servidores de datos 721, 722, y 723, que a un subconjunto propio de estas comparticiones. Uno o más de los servidores 721, 722 y 723 pueden estar localizados en lugares físicamente diferentes, operados bajo un control administrativo diferente, o protegidos por unos controles de acceso de hardware y software heterogéneos. El depósito de datos lógico 720 puede incluir también un sistema de archivos distribuido o en red.

Cuando una parte quiere recuperar el secreto que se ha distribuido en el depósito de datos lógico 720, la entidad 740 puede intentar recopilar las comparticiones. La primera compartición recopilada *S**[1] 744 puede ser la misma que la compartición 704, pero también podría diferir debido a la modificación no intencionada en la transmisión o el almacenamiento (por ejemplo, una corrupción de datos), o una modificación intencionada debido a las actividades de un agente adversario. Del mismo modo, una segunda compartición recopilada *S**[2] 745 puede ser la misma que la compartición 705, y una última compartición recopilada *S**[*n*] 746 puede ser la misma que la compartición 706, pero estas comparticiones podrían diferir también por razones similares. Además de la posibilidad de ser una compartición “errónea”, una o más comparticiones en la recopilación 743 podrían ser también el valor distinguido “desaparecido”, representado por el símbolo, (“◊”). Este símbolo puede indicar que el sistema (por ejemplo, la entidad 740) no es capaz de encontrar o recopilar esta compartición específica. A continuación, puede proporcionarse el vector de comparticiones pretendidas *S** al algoritmo de recuperación 742 del esquema de RCSS, que puede devolver o el secreto recuperado *S** 741 o el valor designado como inválido 747. El secreto compartido 701 debería ser igual al secreto recuperado 741, a menos que el grado de actividad del adversario en las comparticiones corruptas exceda del que se ha diseñado que puede soportar el esquema.

El objetivo de RCSS es útil a través de dos dominios principales: garantizar los datos en reposo y garantizar los datos en movimiento. En el primer escenario, un servidor de archivos, por ejemplo, mantiene sus datos en una variedad de servidores remotos. Incluso si algún subconjunto de esos servidores está dañado (por ejemplo, por unos administradores deshonestos) o no está disponible (por ejemplo, debido a una caída de la red), los datos aún pueden estar disponibles y ser privados. En el escenario de datos en movimiento, el remitente de un mensaje secreto y el receptor del mensaje pueden estar conectados por una multiplicidad de rutas, solo algunas de las cuales pueden observarse por el adversario. Enviando las comparticiones a través de estas rutas diferentes, el remitente

puede transmitir de manera segura el secreto S a pesar de la posibilidad de que algunas rutas estén temporalmente no disponibles o controladas por el adversario. Por ejemplo, en algunas realizaciones, cada compartición puede transmitirse a través de un canal lógico de comunicaciones diferente. Los sistemas y métodos para garantizar los datos, y en particular los sistemas y métodos para garantizar los datos en movimiento específicos, se describen en más detalle en la solicitud de patente de Estados Unidos N.º 10/458.928, presentada el 11 de junio 2003, en la solicitud de patente de Estados Unidos N.º 11/258.839, presentada el 25 de octubre de 2005, y en la solicitud de patente de Estados Unidos N.º 11/602.667, presentada el 20 de noviembre de 2006.

A pesar de que al menos se ha propuesto por Krawczyk un esquema de RCSS con tamaños de comparticiones cortos, el estudio científico de este esquema revela que no es un esquema de RCSS válido bajo supuestos débiles en el esquema de cifrado, y no se sabe que sea una esquema válido para todas las estructuras de acceso (por ejemplo, unas estructuras de acceso distintas de los esquemas de umbral). Por al menos estas razones, las figuras 8-11 describen otros enfoques para la compartición de secretos. Estos otros enfoques a veces se denominan en el presente documento como ESX o HK2.

El mecanismo del enfoque ESX o HK2 puede incluir un esquema de compartición de secretos de cálculo robusto que puede construirse a partir de las siguientes cinco primitivas: (1) un generador de números aleatorios o pseudo-aleatorios, (2) un esquema de cifrado; (3) un esquema de compartición de secretos perfecto (PSS); (4) un algoritmo de dispersión de información (IDA); y (5) un esquema de vinculación probabilística. Estas cinco primitivas se describen con más detalle a continuación.

(1) Un generador de números aleatorios o pseudo-aleatorios, Rand. Un generador de números de este tipo puede tomar un número *k* como entrada y devolver *k* bits aleatorios o pseudo-aleatorios. En las figuras 8-11, se evita la entrada *k* para facilitar la ilustración.

(2) Un esquema de cifrado, que puede incluir un par de algoritmos, uno llamado Cifrar y el otro llamado Descifrar. El algoritmo de cifrado Cifrar puede tomar una clave *K* de una longitud dada *k* y un mensaje de entrada *M* que se denomina como el texto sin cifrar. El algoritmo Cifrar puede devolver una cadena *C* que se conoce como el texto cifrado. El algoritmo Cifrar puede emplear de manera opcional bits aleatorios, pero estos bits aleatorios no se muestran de manera explícita en los dibujos. El algoritmo de descifrado Descifrar puede tomar una clave *K* de una longitud dada *k* y un mensaje de entrada *C* que se denomina como el texto cifrado. El algoritmo Descifrar puede devolver una cadena *M* que se denomina como el texto sin cifrar. En algunos casos, el algoritmo de descifrado puede devolver un valor de fallo designado, que puede indicar que el texto cifrado *C* no se corresponde con el cifrado de cualquier posible texto sin cifrar.

(3) Un esquema de compartición de secretos perfecto (PSS), que puede incluir un par de algoritmos Compartir^{PSS} y Recuperar^{PSS}. El primero de estos algoritmos, conocido como el *algoritmo de compartición* de la PSS, puede ser un mapa probabilístico que toma como entrada una cadena *K*, llamada el secreto, y devuelve una secuencia de *n* cadenas, *K*[1],..., *K*[*n*], denominada como *comparticiones*. Cada *K*[*i*] puede incluir una compartición o las *n* comparticiones que se han tratado, o distribuido, por el distribuidor (la entidad que realiza el proceso de compartición). El número *n* puede ser un parámetro programable por el usuario del sistema de compartición de secretos, y puede incluir cualquier número positivo adecuado. En algunas realizaciones, el algoritmo de compartición es probabilístico por que emplea bits aleatorios o pseudo-aleatorios. Una dependencia de este tipo puede realizarse proporcionando los bits aleatorios o pseudo-aleatorios del algoritmo de compartición, como proporcionados por el algoritmo Rand. El segundo algoritmo, conocido como el *algoritmo de recuperación* de la PSS, puede tomar como entrada un vector de *n* cadenas denominadas como las *comparticiones pretendidas*. Cada compartición pretendida es o una cadena o un símbolo distinguido ("◊") que se lee como *desaparecido*. Este símbolo puede usarse para indicar que alguna compartición específica no está disponible. El algoritmo de recuperación para el esquema de compartición de secretos perfecto puede devolver una cadena *S*, o el *secreto recuperado*. Pueden suponerse dos propiedades del esquema de PSS. La primera propiedad, la propiedad de privacidad, garantiza que ningún conjunto no autorizado de usuarios obtenga cualquier información útil sobre el secreto que se ha compartido a partir de sus comparticiones. La segunda propiedad, la propiedad de recuperación, garantiza que un conjunto autorizado de partes siempre puede recuperar el secreto, suponiendo que las partes autorizadas contribuyen con las comparticiones correctas al algoritmo de recuperación y que cualquier parte adicional contribuye o con una compartición correcta o con el valor desaparecido distinguido ("◊"). Este esquema de PSS puede incluir el esquema de Shamir, denominado comúnmente como "compartición de secretos de Shamir" o el esquema de compartición de secretos Blakley.

(4) Un algoritmo de dispersión de información (IDA), que puede incluir un par de algoritmos Compartir^{IDA} y Recuperar^{IDA}. El primero de estos algoritmos, conocido como el *algoritmo de compartición* del IDA, puede incluir un mecanismo que toma como entrada una cadena *C*, el mensaje a dispersarse, y devuelve una secuencia de n cadenas, *C*[1],..., *C*[*n*], que se denominan como los *fragmentos* de los datos que han resultado de la dispersión. El valor de n puede ser un parámetro programable por el usuario del IDA, y puede ser cualquier número positivo adecuado. El algoritmo de compartición del IDA puede ser probabilístico o determinista. En las figuras 8-11, no se muestra de manera explícita la posibilidad de usar bits aleatorios en el IDA; sin embargo, debería entenderse que los bits aleatorios pueden usarse en el IDA en otras realizaciones.

El segundo algoritmo, conocido como el *algoritmo de recuperación* del IDA, podrá tomar como entrada un vector de *n* cadenas, los fragmentos suministrados. Cada fragmento suministrado puede ser una cadena o el símbolo distinguido ("◊") que se lee como desaparecido y se usa para indicar que algún fragmento de datos específico no

está disponible. El algoritmo de recuperación para el IDA puede devolver una cadena S, el *secreto de recuperación*. El IDA puede suponer que tiene una propiedad de recuperabilidad; por lo tanto, un conjunto autorizado de partes siempre puede recuperar los datos a partir de los fragmentos suministrados, suponiendo que las partes autorizadas contribuyen con los fragmentos correctos al algoritmo de recuperación del IDA y que cualquier parte adicional que participa en la reconstrucción contribuye o con un fragmento correcto o si no con el valor desaparecido distinguido ("◊"). A diferencia del caso de un esquema de PSS, puede no haber una propiedad de privacidad asociada con el IDA y, de hecho, un IDA simple y práctico es replicar la entrada *C* durante n veces, y tener el algoritmo de recuperación que usar el valor que se produce más a menudo como los datos recuperados. Se conocen unos IDA más eficientes (por ejemplo, el IDA de Rabin).

(5) Un esquema de vinculación probabilística, que puede incluir un par de algoritmos, Ct y Vf, llamados el *algoritmo de vinculación* y el *algoritmo de verificación*. El algoritmo de vinculación Ct puede ser un algoritmo probabilístico que toma una cadena *M* para vincular y devuelve un valor de vinculación, *H* (la cadena que un actor puede usar para vincular *M*) y también un valor de no vinculación, *R* (la cadena que un actor puede usar para no vincular el *H* de vinculación de *M*). El algoritmo de vinculación puede ser probabilístico y, como tal, puede tomar un argumento final, *R**, que se denomina como las monedas del algoritmo. Estas monedas pueden generarse anteriormente por una llamada a un generador de números aleatorios o pseudo-aleatorios, Rand. La notación "Ct(*M*; *R**)" se usa a veces en el presente documento para indicar de manera explícita el valor de retorno del algoritmo de vinculación Ct en la entrada *M* con las monedas aleatorias *R**. El algoritmo de verificación, Vf, puede ser un algoritmo determinista que tiene tres cadenas de entrada: un valor de vinculación *H*, una cadena *M*, y un valor de no vinculación *R*. Este algoritmo puede devolver un bit 0 o 1, con 0 indicando que la no vinculación es inválida (no convincente) y 1 indicando que la no vinculación es válida (convincente).

En general, un esquema de vinculación puede satisfacer dos propiedades: una propiedad de ocultación y una propiedad de unión. La propiedad de ocultación implica que, dada una vinculación determinada de manera aleatoria *H* para un mensaje elegido de manera adversa M_0 o M_1, el adversario no puede determinar qué mensaje *H* corresponde a la vinculación. La propiedad de unión implica que un adversario, que ha vinculado un mensaje M_0 por medio de una vinculación H_0 y la no vinculación correspondiente R_0, es incapaz de encontrar algún mensaje M_1 distinto de M_0 y cualquier no vinculación R_1 de tal manera que $Vf(H_0, M_1, R_1) = 1$. En la mayoría de los casos, el valor de no vinculación *R* producido por un esquema de vinculación Ct (*M*; *R**) es, precisamente, las monedas aleatorias *R** proporcionadas al algoritmo (es decir, *R* = *R**). Sin embargo, esta propiedad no se necesita en todos los casos. Los esquemas de vinculación probabilísticas más naturales pueden obtenerse por medio de las funciones hash de cifrado adecuadas, tales como SHA-1. Hay una variedad de técnicas naturales para procesar el valor que se ha vinculado para, *M*, y las monedas, *R**, antes de aplicar las funciones hash de cifrado. Cualquier esquema de vinculación que contiene un mecanismo de vinculación Ct y un algoritmo de verificación Vf puede producir un mecanismo de vinculación Vincular y un mecanismo de verificación Verificar que se aplica a los vectores de las cadenas en lugar de a las cadenas individuales. El algoritmo de vinculación Vincular puede aplicar el algoritmo Ct por componentes, y el algoritmo de verificación Verificar puede aplicar el algoritmo Vf por componentes. Para Ct, pueden usarse las monedas aleatorias independientes para cada cadena de componente en algunas realizaciones.

La figura 8 muestra un diagrama de bloques simplificado del mecanismo de compartición del esquema de RCSS de acuerdo con una realización de la invención. El secreto, *S*, 800 puede incluir el secreto que el distribuidor desea distribuir o compartir. El secreto 800 puede ser un archivo en un sistema de archivos, un mensaje procedente de un protocolo de comunicaciones, o cualquier otra pieza de datos sensible. El secreto 800 puede representarse como cualquier cadena codificada adecuada (por ejemplo, una cadena ASCII o codificada en binario). Sin embargo, en implementaciones reales las cadenas binarias pueden usarse como el secreto 800 para facilidad de la implementación. Primero, el secreto S puede cifrarse usando el algoritmo de cifrado 803 de un esquema de cifrado de clave compartida para obtener un texto cifrado *C* 804. La clave *K* 802 para realizar este cifrado puede obtenerse usando la salida del generador de números aleatorios o pseudo-aleatorios 801 con el fin de producir el número apropiado de bits aleatorios o pseudo-aleatorios para la clave 802.

La clave 802 puede usarse para solo una compartición, y por lo tanto puede denominarse como una clave de una vez. Además de usarse para cifrar el secreto 800, la clave 802 puede compartirse o distribuirse también usando un esquema de compartición de secretos perfecto (PSS) 806. El esquema de PSS 806 puede incluir cualquier esquema de compartición de secretos perfecto, incluyendo los esquemas de compartición de secretos de Shamir o Blakley. El esquema de compartición de secretos perfecto 806 puede ser aleatorio, lo que requiere su propia fuente de bits aleatorios (o pseudo-aleatorios). Los bits aleatorios o pseudo-aleatorios pueden proporcionarse por un generador de números aleatorios o pseudo-aleatorio independiente, tal como generador de números 805. El esquema de PSS 806 puede emitir un vector de comparticiones de clave $K = K[1],..., K[n]$ 808, que, conceptualmente, puede enviarse a los diferentes "actores", una compartición por cada actor. En primer lugar, sin embargo, las comparticiones de clave pueden combinarse con información adicional en algunas realizaciones. El texto cifrado *C* 804 puede dividirse en fragmentos 809 usando un algoritmo de dispersión de información (IDA) 807, tal como el mecanismo de IDA de Rabin. El IDA 807 puede emitir un vector de fragmentos de texto cifrado $C[1],..., C[n]$ 809. A continuación, puede emplearse el mecanismo de vinculación 812 de un esquema de vinculación probabilística. Un número suficiente de bits aleatorios se generan durante el proceso de vinculación usando el generador de números aleatorios o pseudo-aleatorio 810, y la cadena aleatoria resultante 811 se usa para todas las vinculaciones en el mecanismo de vinculación 812. El mecanismo de vinculación 812 puede determinar un valor de vinculación $H[i]$ y un valor de no

vinculación *R*[*i*], mostrados de manera conjunta en el vector 813, para cada mensaje *M*[*i*] = *K*[*i*] *C*[*i*] (extendido a través de 808 y 809). La *i*-ésima compartición (que no se representa de manera explícita en la figura 8) puede codificar *K*[*i*] 808, *C*[*i*] 809, *R*[*i*], y *H*[1],..., *H*[*n*] 813. Cada parte *i* puede recibir en su compartición la vinculación *H*[*j*] para cada *K*[*j*], *C*[*j*] (para *j* en 1... *n*) y no simplemente la vinculación de su propia compartición.

La figura 9 muestra, con mayor detalle, el proceso de vinculación ilustrativo del mecanismo de vinculación 812 (figura 8). El proceso de vinculación implica n diferentes llamadas al mecanismo de Ct de nivel inferior del esquema de vinculación. La aleatoriedad se genera por el generador de números aleatorios o pseudo-aleatorios 900 y la cadena aleatoria o pseudo-aleatoria resultante *R** se particiona en n segmentos, *R**[1] *R**[2],..., *R**[*n*] 901. La *i*-ésima porción de la aleatoriedad (una de las porciones 921, 922, o 923 cuando i es 1, 2 o *n*) se usa para vincular el *i*-ésimo mensaje que se está procesando, *M*[*i*] = *K*[*i*] *C*[*i*] (mostrado como los mensajes 910, 911, 912) usando los algoritmos de vinculación Ct 931, 932, y 933 de un esquema de vinculación. Los pares de vinculación y de no vinculación 941, 942, y 943, pueden emitirse por el algoritmo Ct. Es probable que cada *R*[*i*] sea simplemente *R**[*i*], pero esto no es estrictamente necesario o supuesto.

El algoritmo etiquetado “Compartir” en la Tabla 1, a continuación, explica adicionalmente el sistema de compartición representado en las figuras 8 y 9. Este algoritmo toma como entrada una cadena *S*, el secreto que se va a compartir. En la línea 10, se generan un número suficiente de lanzamientos de moneda aleatorios para proporcionar una clave de cifrado *K* para un esquema de cifrado simétrico que consiste en los algoritmos de Cifrar y Descifrar. En la línea 11, la cadena sensible *S* que se va a compartir se cifra usando la clave *K* con el fin de crear un texto cifrado *C*. El cifrado puede ser aleatorio, pero no es necesario para que el mecanismo funcione de manera correcta. A continuación, en la línea 12, puede recurrirse el algoritmo de compartición de un esquema de compartición de secretos perfecto (tal como el esquema de Shamir). El algoritmo de compartición es probabilístico, aunque esto no se indica de manera explícita en el código. La compartición da como resultado un vector de comparticiones de clave, *K* = *K*[1]... *K*[*n*]. En la línea 13, el texto cifrado *C* puede dividirse en una recopilación de fragmentos a partir de la que una subrecopilación autorizado de fragmentos se adecuará para recuperar el secreto. Esto puede realizarse usando el algoritmo de compartición de un IDA (por ejemplo, el IDA 807 de la figura 8). Cualquier IDA válido puede usarse, tal como el mecanismo de Rabin, una replicación, o cualquier esquema ad hoc con el IDA descrito anteriormente de manera adecuada. Las líneas 15 y 16 comprenden una vinculación probabilística del mensaje *KC*[*i*] = *K*[*i*] *C*[*i*], con las monedas necesarias que se generan en la línea 15 y la vinculación *H*[*i*] y la no vinculación *R*[*i*] que se calculan usando estas monedas. La línea 17 calcula la compartición resultante (algunas veces denominada como “fragmento” en el presente documento) *S*[*i*] a partir de los valores ya calculados. La compartición en el esquema RCSS objeto es *S*[*i*] = *R*[*i*] *K*[*i*] *C*[*i*] H[1]... *H*[*n*]. A continuación, las comparticiones pueden devolverse al llamante, para almacenarse en diferentes localizaciones o transmitirse a través de una variedad de canales, de acuerdo con la intención del llamante.

El algoritmo de recuperación del esquema de RCSS se muestra también, a continuación, en la Tabla 1. Esta vez, el llamante proporciona un vector de la totalidad de las comparticiones pretendidas, *S* = *S*[l]... *S*[*n*]. Cada compartición pretendida *S*[*i*] puede ser una cadena o el símbolo distinguido "◊", que a su vez representa una compartición desaparecida. También puede suponerse, en algunas realizaciones, que el llamante proporciona la identidad de una compartición *j*, en la que *j* está entre 1 y *n* inclusive, que se sabe que es válido. En las líneas 20-21, cada *S*[*i*] puede analizarse en sus cadenas de componentes *R*[*i*] *K*[*i*], *C*[*i*], y *H*[1]... H[*n*]. Se entiende que el símbolo desaparecido "◊", puede analizarse en los componentes todos los cuales son a su vez el símbolo desaparecido 0. En la línea 23, puede ejecutarse el algoritmo de verificación del esquema de vinculación para determinar si el mensaje *KC*[*i*] = *K*[*i*] *C*[*i*] parece ser válido. A continuación, puede usarse la compartición “válida conocida” *j* como el “valor de referencia” para cada vinculación $H_j[i]$. Siempre que un valor *K*[*i*] *C*[*i*] parece ser inválido, puede reemplazarse por el símbolo desaparecido. El vector de los valores *K*[*i*] que se han revisado en ese sentido puede ahora suministrarse al algoritmo de recuperación del esquema de compartición de secretos en la línea 25, mientras que el vector de los valores revisados *C*[*i*] puede suministrarse con el algoritmo de recuperación del IDA en línea 26. En este punto, uno solo tiene que descifrar el texto cifrado *C* recuperado a partir de la IDA en la clave *K* recuperada a partir del esquema de PSS para obtener el valor *S* que se recupera por el propio esquema de RCSS.

Tabla 1: Mecanismos de compartición y recuperación del esquema de RCSS.

Algoritmo Compartir (*S*)		
10	*K* ←	Rand (*k*)
11	*C* ←	Cifrar$_K$(*S*)
12	*K* ←	Compartir^{PSS}(*K*)
13	*C* ←	Compartir^{IDA}(*C*)
14	for	*i*←1 to *n* do
15		*R**[*i*] ← Rand (*k'*)
16		(*H*[*i*], *R*[*i*]) ← Ct(*K*[*i*] *C*[*i*]; *R**[*i*])
17		*S*[*i*] ← *R*[*i*] *K*[*i*] *C*[*i*] *H*[1] ...*H*[*n*]

```
Algoritmo Compartir (S)
18      return S

Algoritmo Recuperar (S, j)
20      for         i←1 to n do
21                  R[i] K[i] C[i] H_i[1] ... H_i[n] ← S[i]
22      for         i←1 to n do
23                  if S[i] ≠ ◊ and Vf (H_j[i], K[i] C[i], R[i])
24                  then K[i] ← ◊, C[i] ← ◊
25      K ←         Recuperar^PSS (K)
26      C ←         Recuperar^IDA(C)
27      S ←         Descifrar_K(C)
28      return S
```

Como se ha indicado anteriormente, el algoritmo Recuperar de la Tabla 1 supone que el usuario suministra la localización de una compartición conocida válida. En la ausencia de esta, pueden emplearse otros medios para determinar un valor de consenso para $H[i]$. La posibilidad más natural usada en algunas realizaciones es el voto de la mayoría. Por ejemplo, en lugar de $H_i[i]$ en la línea 23, puede usarse un valor de $H[i]$ que se produce con mayor frecuencia entre los valores recuperados $H_j[i]$, para j variando de 1 a n.

Volviendo brevemente a la figura 8, la porción de la figura que está etiquetada de 801 a 807 puede implementarse o considerarse como un único proceso que incluye una compartición de secretos de cálculo (CSS) de S para obtener el vector de las comparticiones $KC = (KC[1],..., KC[n])$, en la que $KC[i] = K[i]\ C[i]$, con una vinculación probabilística aplicada al vector resultante de las comparticiones. La figura 10 muestra un esquema descrito a partir de esta realización alternativa. En esta realización, se emplean las tres primitivas siguientes, en lugar de las anteriores cinco primitivas definidas en relación con las figuras 8 y 9: (1) un generador de números aleatorios o pseudo-aleatorios, Rand; (2) un esquema de compartición de secretos de cálculo (CSS); y (3) un esquema de vinculación probabilística.

El generador de números aleatorios o pseudo-aleatorios, Rand, puede definirse como anteriormente. El esquema de compartición de secretos de cálculo puede incluir un par de algoritmos Compartir^{CSS} y Recuperar^{CSS}. El primero de estos algoritmos, conocido como el *algoritmo de compartición* de la CSS, puede ser un mapa probabilístico que toma como entrada una cadena K, llamada el secreto, y devuelve una secuencia de n cadenas, $K[1],..., K[n]$, denominadas como las *comparticiones*. Cada $K[i]$ puede incluir una *compartición* o las n comparticiones que se han tratado, o distribuido, por el distribuidor (la entidad que realiza el proceso de compartición). El número n puede ser un parámetro del esquema de compartición de secretos, y puede ser un número positivo arbitrario. El algoritmo de compartición puede ser probabilístico por que puede emplear unos bits aleatorios o pseudo-aleatorios. Una dependencia de este tipo puede realizarse proporcionando los bits aleatorios o pseudo-aleatorios del algoritmo de compartición, como proporcionados por el generador de números aleatorios o pseudo-aleatorios, Rand.

El segundo algoritmo, que se conoce como el *algoritmo de recuperación* de la CSS, toma como entrada un vector de n cadenas, denominado como las *comparticiones pretendidas*. Cada compartición pretendida es o una cadena o un símbolo distinguido "◊", que se lee como *desaparecido* y se usa para indicar que algunas comparticiones específicas no están disponibles o se desconocen. El algoritmo de recuperación para el esquema de compartición de secretos de cálculo puede devolver una cadena S, el *secreto recuperado*. Ya que el par de algoritmos conforman un esquema de compartición de secretos de cálculo, pueden suponerse dos propiedades. La primera propiedad, la propiedad de privacidad, puede garantizar que ningún conjunto de usuarios no autorizado obtiene alguna información significativa (que puede extraerse mediante cálculos) sobre el secreto que se ha compartido a partir de sus comparticiones. La segunda propiedad, la propiedad de recuperación, garantiza que un conjunto autorizado de partes siempre puede recuperar el secreto, suponiendo que las partes autorizadas contribuyen con las comparticiones correctas al algoritmo de recuperación y que cualquier parte adicional contribuye o con la compartición correcta o si no con el valor desaparecido distinguido ("◊").

La tercera primitiva en esta realización es un esquema de vinculación probabilística, que puede implementarse como se ha descrito anteriormente en relación con las figuras 8 y 9.

Haciendo referencia a la figura 10, la cadena secreta S 1000 puede compartirse, o distribuirse, usando el algoritmo Compartir 1001 de un esquema de compartición de secretos de cálculo (probabilístico). Esto puede dar como resultado n comparticiones, $KC[1],..., KC[n]$ 1002. A continuación, puede emplearse un esquema de vinculación probabilística 1005 para obtener el vector 1006 de vinculación y de no vinculación. La vinculación probabilística puede emplear lanzamientos de monedas 1004 generados por algún generador de números aleatorios o pseudo-

aleatorios 1003. Compartir 1 del esquema de RCSS, *S*[1], puede incluir la compartición *KC*[1] a partir del esquema de CSS 1002 junto con la no vinculación *R*[1] a partir del esquema de vinculación 1006 junto con el vector de vinculación *H*[1]... *H*[*n*] a partir del esquema de vinculación 1006. Compartir 2 del esquema de RCSS, *S*[2], pueden incluir la compartición *KC*[2] a partir del esquema de CSS 1002 junto con la no vinculación *R*[2] a partir del esquema de vinculación 1006 junto con el vector de vinculación *H*[1]... *H*[*n*] a partir del esquema de vinculación 1006. Este proceso puede continuar, con una compartición n del esquema de RCSS, *S*[*n*], que incluye la compartición *KC*[*n*] a partir del esquema de CSS 1002 junto con la no vinculación *R*[*n*] a partir del esquema de vinculación 1006 junto con el vector de vinculación *H*[1]... *H*[*n*] a partir del esquema de vinculación 1006.

La figura 11 ilustra el proceso de recuperación del esquema de RCSS que se acaba de describir. El algoritmo Recuperar 1130 está provisto de un vector de comparticiones pretendidas, que algunas veces se llaman fragmentos en el presente documento, para distinguir estas comparticiones de las comparticiones del esquema de CSS. El *i*-ésimo fragmento recibido por el algoritmo Recuperar 1130 se analiza para formar una cadena *KC*[*i*], un valor de no vinculación *R*[*i*], y un vector de vinculación $H_i = H_i[1]... H_i[n]$. A partir de la recopilación de los vectores de vinculación $H_1[i]... H_n[i]$, el algoritmo Recuperar 1130 debe determinar una vinculación de consenso *H*[*i*]. Para el escenario en el que el algoritmo Recuperar 1130 está provisto de un índice *j* para un actor cuya compartición se sabe que es válida, el valor de consenso H[*i*] puede seleccionarse para que sea $H_j[i]$. Para el caso en que no se sabe que tal compartición es auténtica, el valor de consenso puede seleccionarse como un valor de cadena que se produce con mayor frecuencia entre $H_1[i],..., H_n[i]$. La figura 11 representa las comparticiones *KC*[1] 1100, *KC*[2] 1110, y *KC*[*n*] 1120 analizadas fuera de los fragmentos primero, segundo, y *n*-ésimo proporcionados al algoritmo Recuperar de RCSS, respectivamente. El ejemplo mostrado en la figura 11 representa igualmente los valores de no vinculación *R*[1] 1102, *R*[2] 1112, y *R*[*n*] 1122 analizados fuera de los fragmentos primero, segundo, y *n*-ésimo proporcionados al algoritmo Recuperar de RCSS, respectivamente. La figura 11 representa también los valores de vinculación de consenso *H*[1] 1101, *H*[2] 1111, y *H*[*n*] 1121, determinados de la manera descrita anteriormente. Centrándose en el procesamiento del primer fragmento, el algoritmo de verificación Vf 1104 del esquema de vinculación probabilística se llama en la vinculación *H*[1], el mensaje *KC*[1], y la no vinculación *R*[1]. El algoritmo puede devolver un bit, con, por ejemplo, 0 indicando que el mensaje *KC*[1] no debería aceptarse como teniendo que no vincularse, y 1 indicando que debería. En consecuencia, se alimenta un demultiplexor 1106 con el bit de decisión del algoritmo de verificación, con, por ejemplo, un 0 indicando que el valor recuperado debería considerarse como desaparecido ("◊") 1105 y un 1 indicando que el valor recuperado debería considerarse como el propio *KC*[1] 1100. La salida A es la primera entrada suministrada al algoritmo Recuperar 1130 de un esquema de CSS. Continuando de esta manera, se procesa el fragmento 2 (mostrado en 1110-1116 en el ejemplo de la figura 11) y cada fragmento adicional se procesa, hasta que se procesa el *n*-ésimo (mostrado en 1120-1126 en el ejemplo de la figura 11). A continuación, se proporciona la recopilación de comparticiones al algoritmo Recuperar 1130 del esquema de CSS con el fin de recuperar el secreto. Este valor recuperado puede ser el valor emitido por el propio esquema de RCSS.

Los expertos en la materia se darán cuenta de que son posibles un gran número de variantes. Por ejemplo, puede usarse un código de corrección de errores en algunas realizaciones para proporcionar una recopilación adecuada de vinculaciones *H*[1]... *H*[*n*] para cada actor, sustituyendo de manera eficaz el simple pero algo ineficaz código de replicación de la realización anterior.

A pesar de que algunas aplicaciones comunes se describen anteriormente, debería entenderse claramente que la presente invención puede integrarse con cualquier aplicación de red con el fin de aumentar la seguridad, la tolerancia a fallos, el anonimato, o cualquier combinación adecuada de los anteriores.

Además, otras combinaciones, adiciones, sustituciones y modificaciones serán evidentes para los expertos en la materia en vista de la divulgación en el presente documento. En consecuencia, la presente invención no pretende estar limitada por la respuesta de las realizaciones preferidas, pero ha de definirse por referencia a las reivindicaciones adjuntas.

REIVINDICACIONES

1. Un método para garantizar datos generando una recopilación de fragmentos de los datos, comprendiendo el método:

aplicar un mecanismo de compartición de un esquema de compartición de secretos de cálculo a los datos para producir una recopilación de comparticiones;
generar un valor aleatorio o pseudo-aleatorio;
calcular un conjunto de valores de vinculación y un conjunto de valores de no vinculación a partir del valor aleatorio o pseudo-aleatorio y la recopilación de las comparticiones;
producir cada fragmento en la recopilación de fragmentos combinando una compartición, un valor de no vinculación, y al menos un valor de vinculación del conjunto de valores de vinculación; y
almacenar cada fragmento en al menos un depósito de datos.

2. El método de la reivindicación 1, en el que producir cada fragmento en la recopilación de fragmentos comprende combinar una compartición, un valor de no vinculación, y todo el conjunto de valores de vinculación.

3. El método de la reivindicación 1, en el que calcular un conjunto de valores de vinculación y un conjunto de valores de no vinculación comprende emplear un esquema de vinculación probabilística.

4. Un método para garantizar los datos, comprendiendo el método:

aplicar un mecanismo de compartición de un esquema de compartición de secretos de cálculo a los datos para producir una recopilación de comparticiones;
usar un esquema de vinculación probabilística para calcular un conjunto de valores de vinculación y un conjunto de valores de no vinculación a partir de la recopilación de las comparticiones;
producir una pluralidad de fragmentos, en el que cada fragmento comprende una compartición de la recopilación de comparticiones, un valor de no vinculación del conjunto de valores de no vinculación, y el conjunto de valores de vinculación; y
almacenar cada fragmento en al menos un depósito de datos.

5. El método de la reivindicación 1 o 4, en el que almacenar cada fragmento en al menos un depósito de datos comprende almacenar cada fragmento en diferentes localizaciones geográficas.

6. El método de la reivindicación 1 o 4, en el que almacenar cada fragmento en al menos un depósito de datos comprende almacenar cada fragmento en diferentes localizaciones físicas en el al menos un depósito de datos.

7. El método de la reivindicación 1 o 4, en el que el al menos un depósito de datos comprende un sistema de archivos distribuido.

8. El método de la reivindicación 1 o 4, en el que el esquema de compartición de secretos de cálculo se selecciona del grupo que consiste en los esquemas de compartición de secretos Shamir, Blakley, y Krawczyk.

9. El método de la reivindicación 1 o 4, que comprende además transmitir los fragmentos producidos a través de una pluralidad de canales de comunicación.

10. El método de la reivindicación 9, en el que transmitir los fragmentos producidos a través de una pluralidad de canales de comunicación comprende transmitir cada fragmento producido a través de un canal de comunicación diferente.

11. Un sistema que comprende al menos un procesador de ordenador configurado para realizar el método de acuerdo con una cualquiera de las reivindicaciones 1-10.

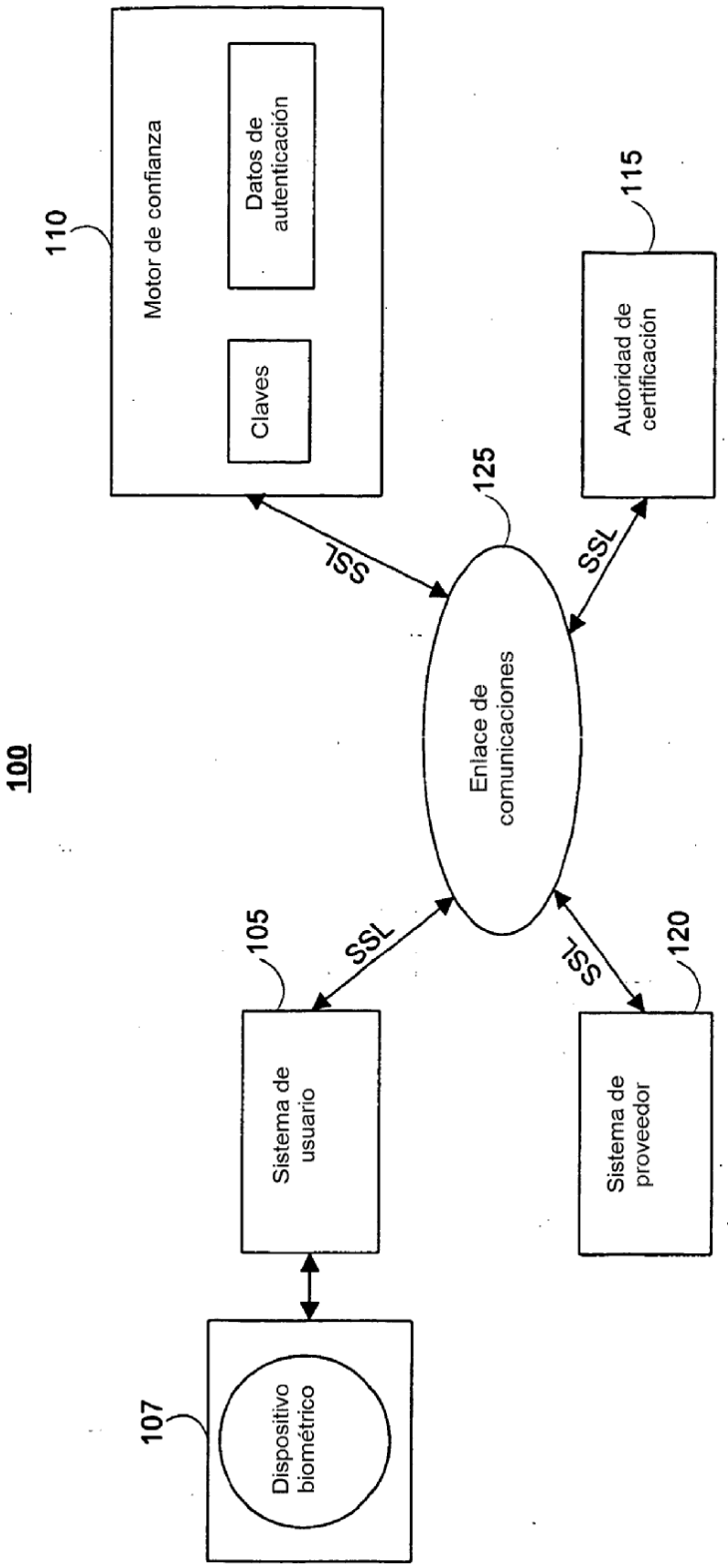
100
110
Motor de confianza
Datos de autenticación
Claves
115
Autoridad de certificación
125
Enlace de comunicaciones
SSL
SSL
SSL
SSL
105
Sistema de usuario
120
Sistema de proveedor
107
Dispositivo biométrico

FIG. 1

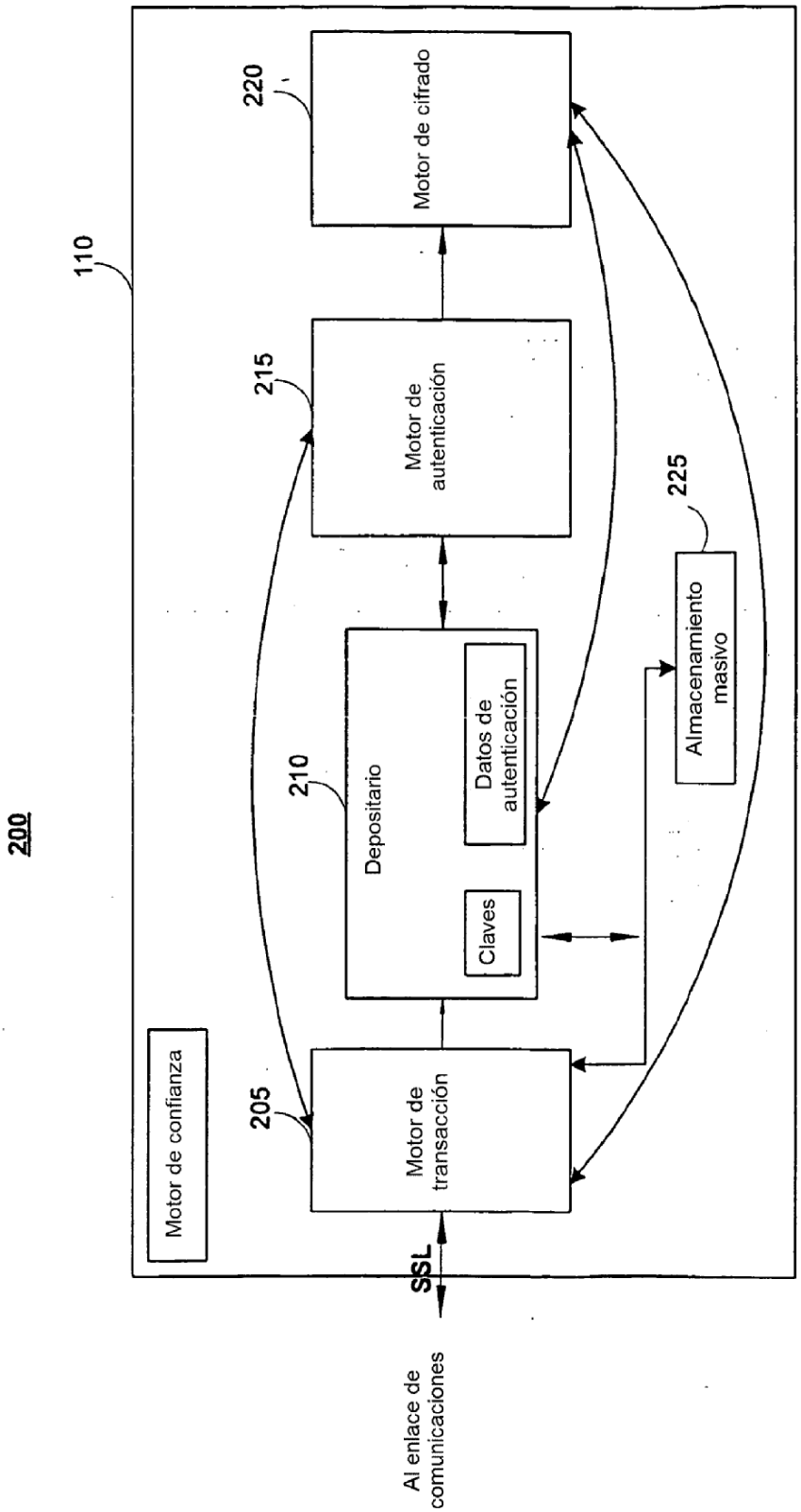
200
110
220
Motor de cifrado
215
Motor de autenticación
225
Almacenamiento masivo
210
Depositario
Datos de autenticación
Claves
205
Motor de transacción
Motor de confianza
SSL
Al enlace de comunicaciones

FIG. 2

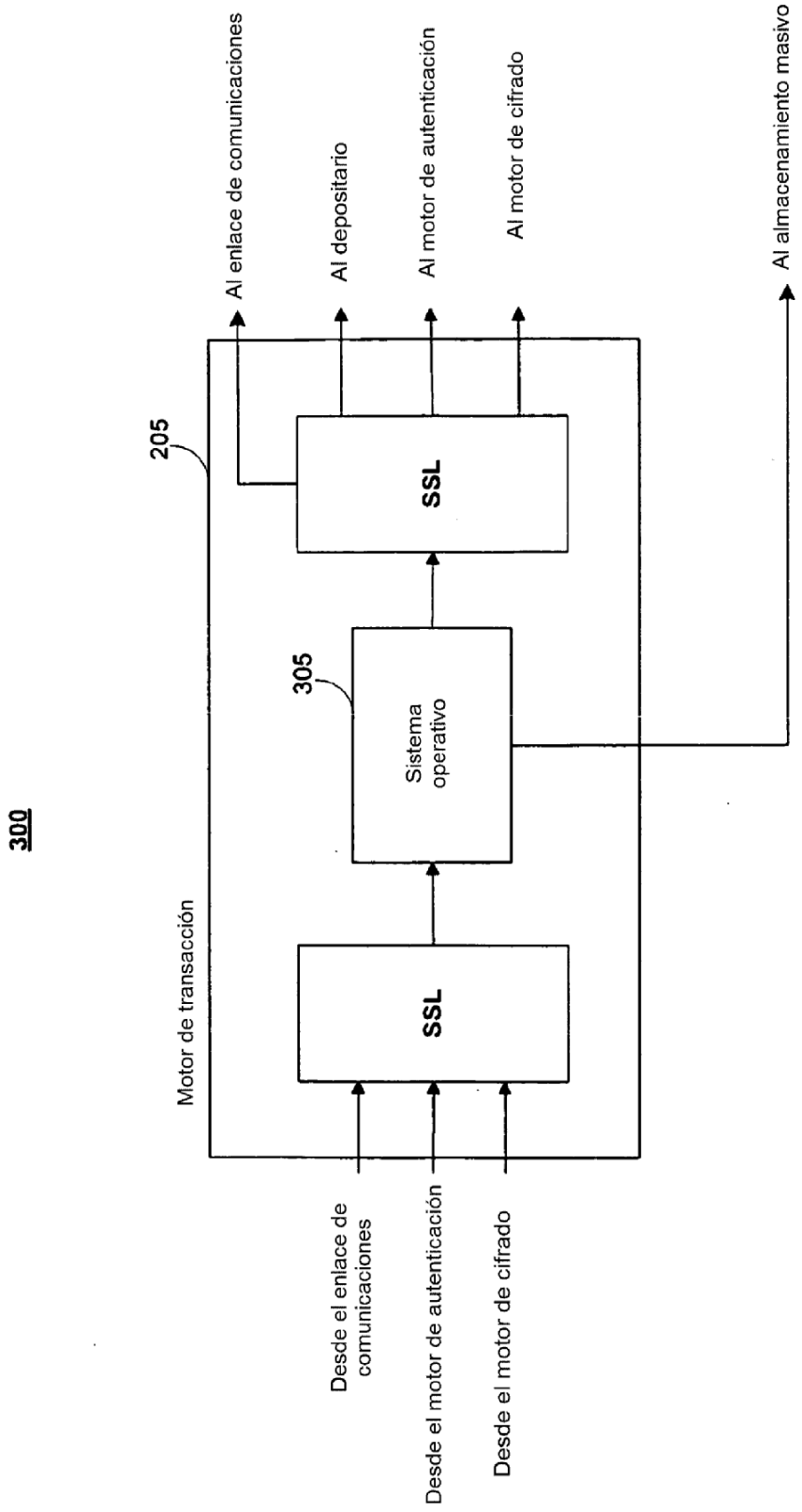
300
205
305
Motor de transacción
SSL
SSL
Sistema operativo
Desde el enlace de comunicaciones
Desde el motor de autenticación
Desde el motor de cifrado
Al enlace de comunicaciones
Al depositario
Al motor de autenticación
Al motor de cifrado
Al almacenamiento masivo

FIG. 3

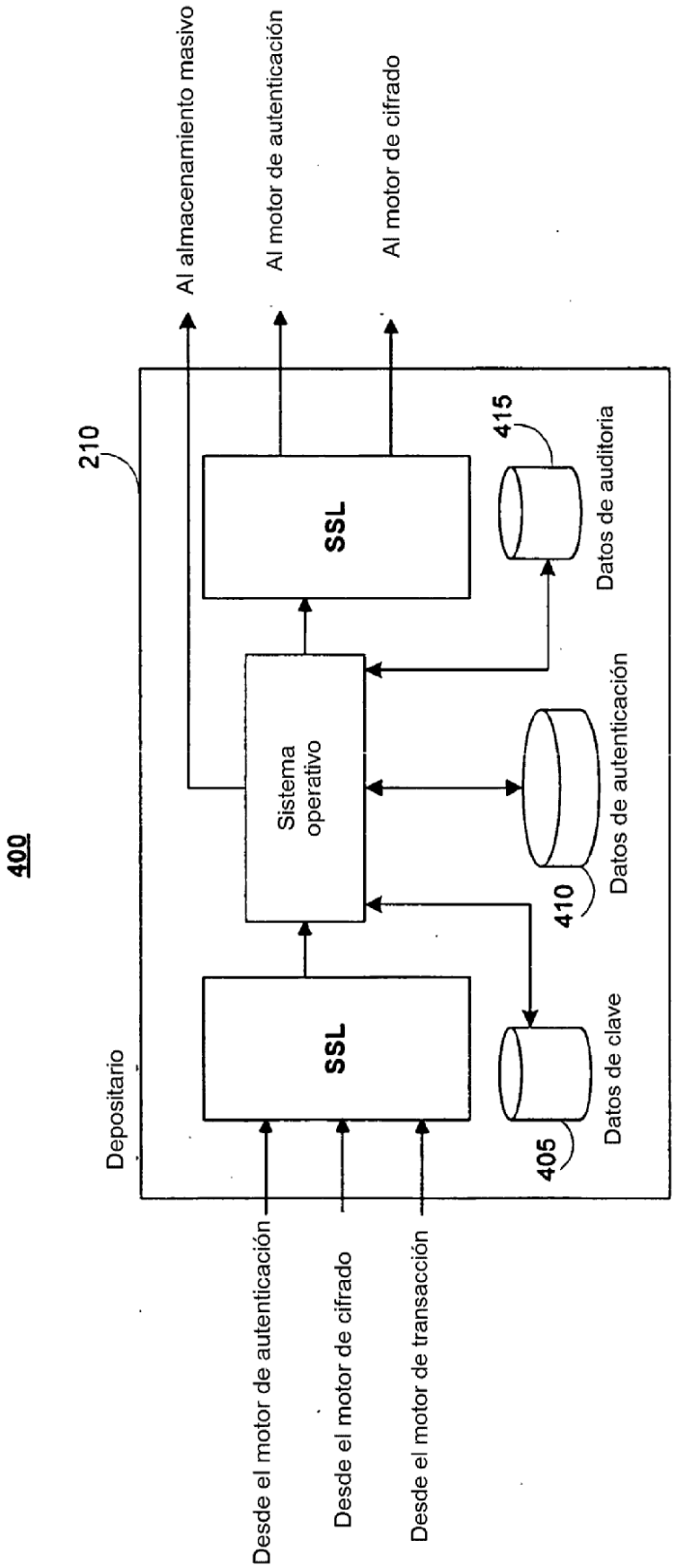
400
210
Depositario
SSL
SSL
Sistema operativo
Datos de clave
Datos de autenticación
Datos de auditoria
405
410
415
Desde el motor de autenticación
Desde el motor de cifrado
Desde el motor de transacción
Al almacenamiento masivo
Al motor de autenticación
Al motor de cifrado

FIG. 4

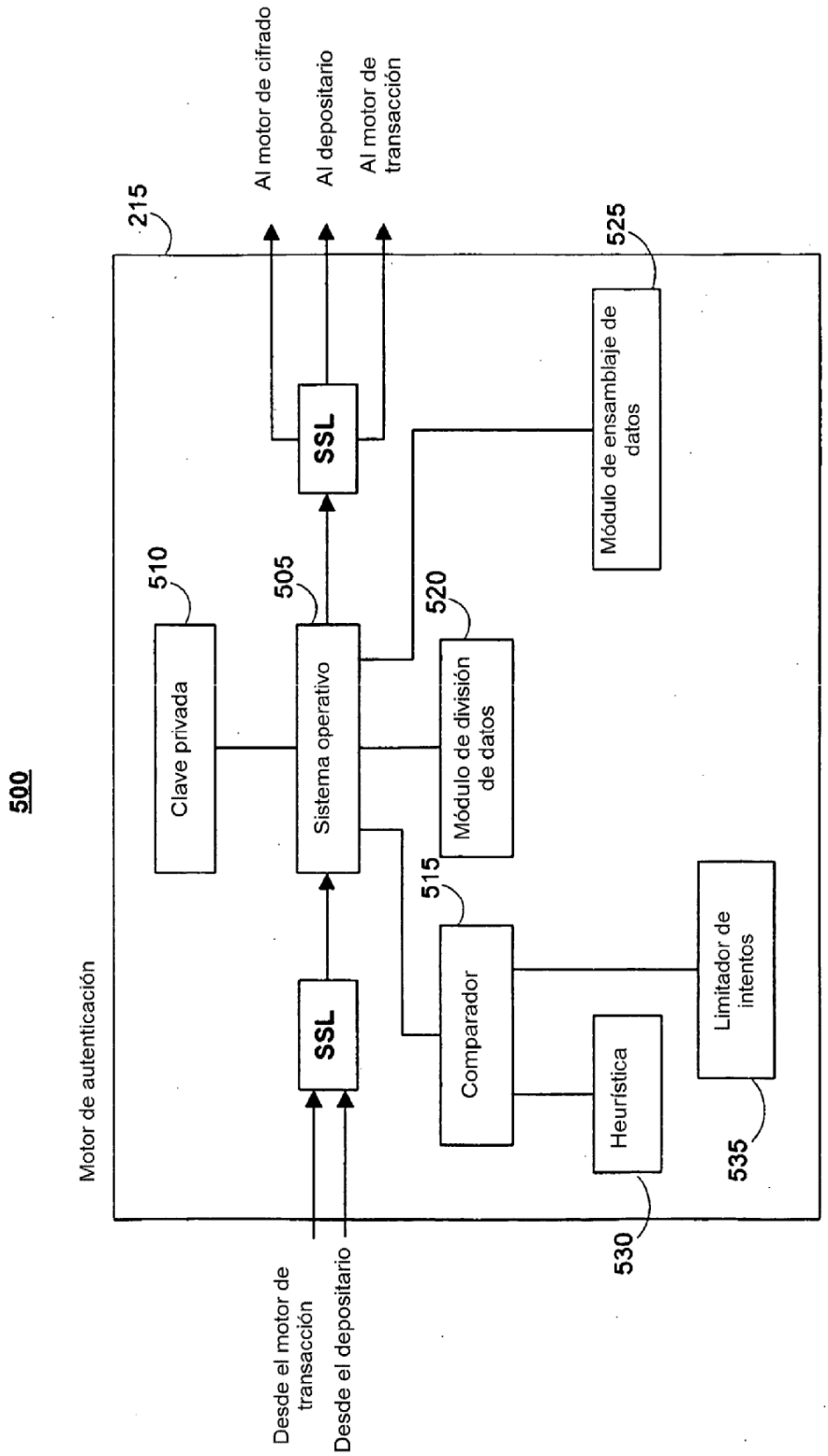
500
Motor de autenticación
215
Al motor de cifrado
Al depositario
Al motor de transacción
SSL
SSL
Clave privada
Sistema operativo
Módulo de división de datos
Módulo de ensamblaje de datos
Comparador
Heurística
Limitador de intentos
510
505
520
525
515
530
535
Desde el motor de transacción
Desde el depositario

FIG. 5

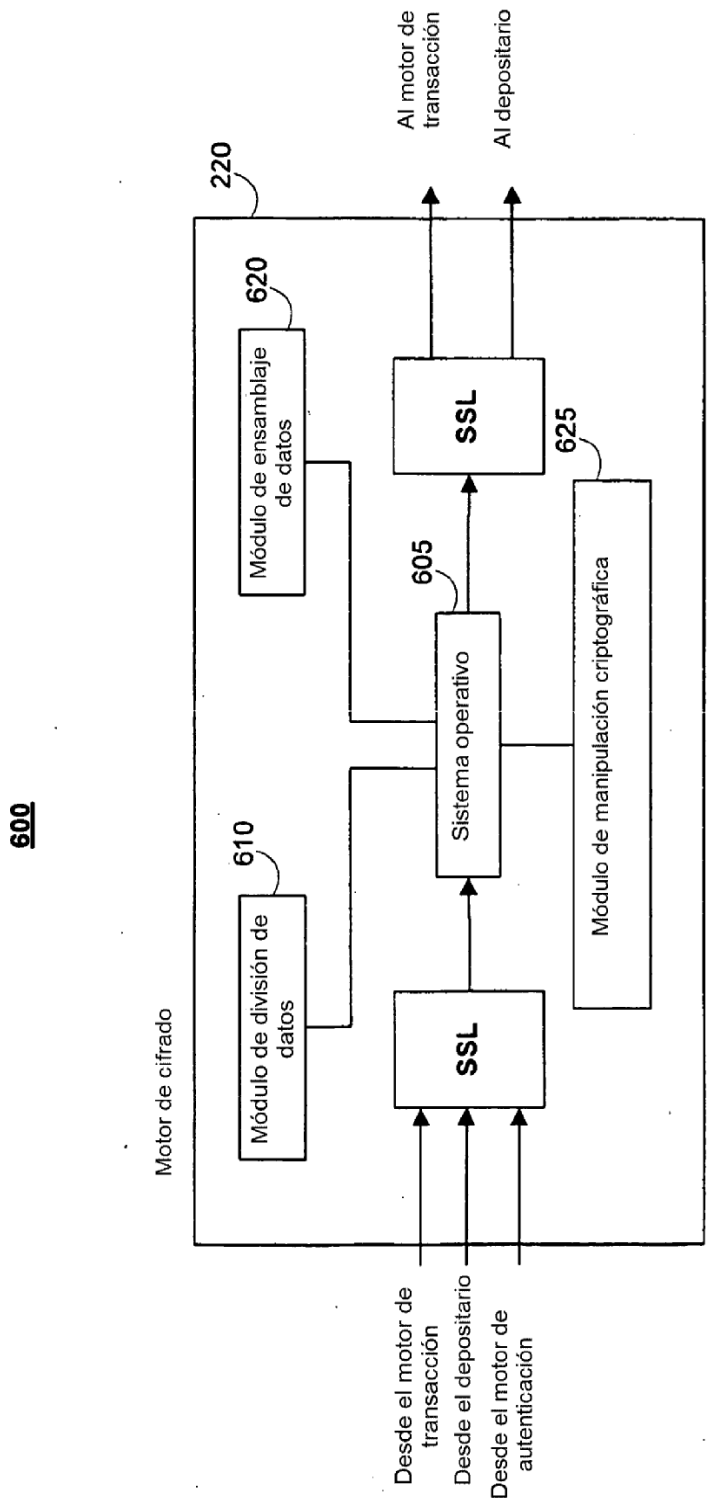
600
Motor de cifrado
220
Módulo de división de datos
610
Módulo de ensamblaje de datos
620
SSL
SSL
Sistema operativo
605
Módulo de manipulación criptográfica
625
Desde el motor de transacción
Desde el depositario
Desde el motor de autenticación
Al motor de transacción
Al depositario

FIG. 6

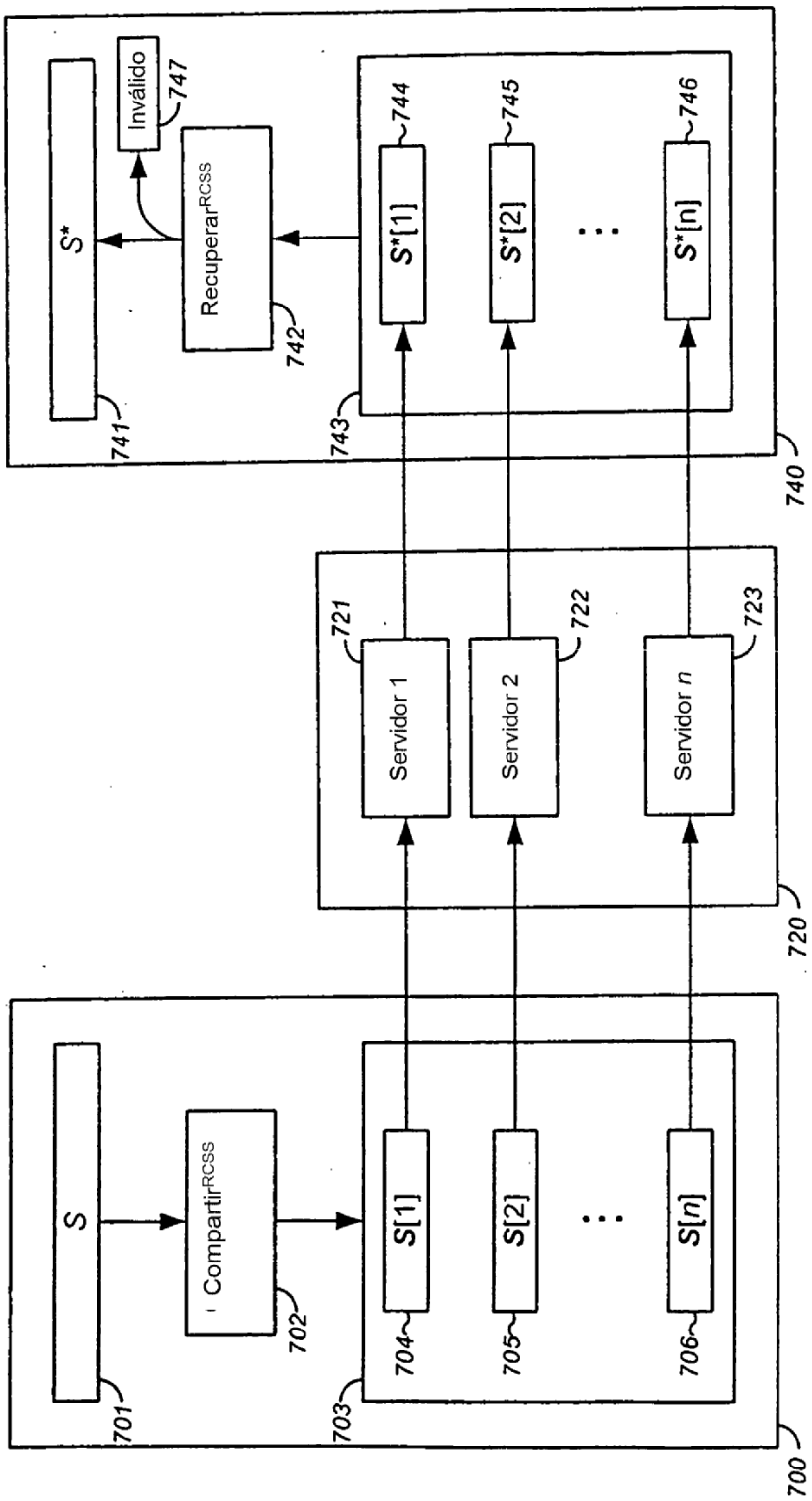
S
CompartirRCSS
S[1]
S[2]
S[n]
Servidor 1
Servidor 2
Servidor n
S*
Inválido
RecuperarRCSS
S*[1]
S*[2]
S*[n]
700
701
702
703
704
705
706
720
721
722
723
740
741
742
743
744
745
746
747

Figura 7

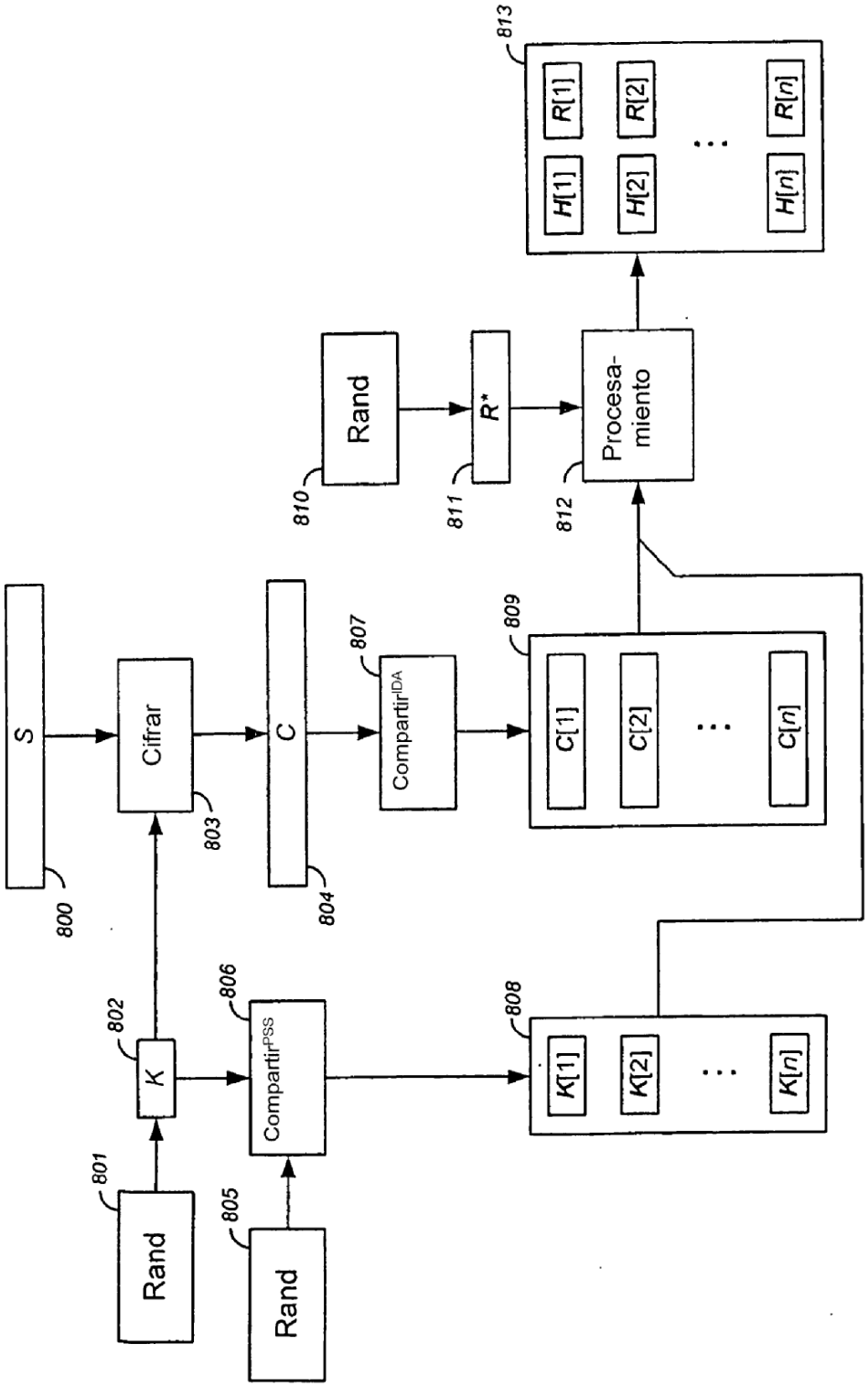
S
800
Cifrar
803
C
804
CompartirIDA
807
C[1]
C[2]
C[n]
809
Rand
810
R*
811
Procesa-
miento
812
H[1]
H[2]
H[n]
R[1]
R[2]
R[n]
813
Rand
801
K
802
CompartirPSS
806
Rand
805
K[1]
K[2]
K[n]
808

Figura 8

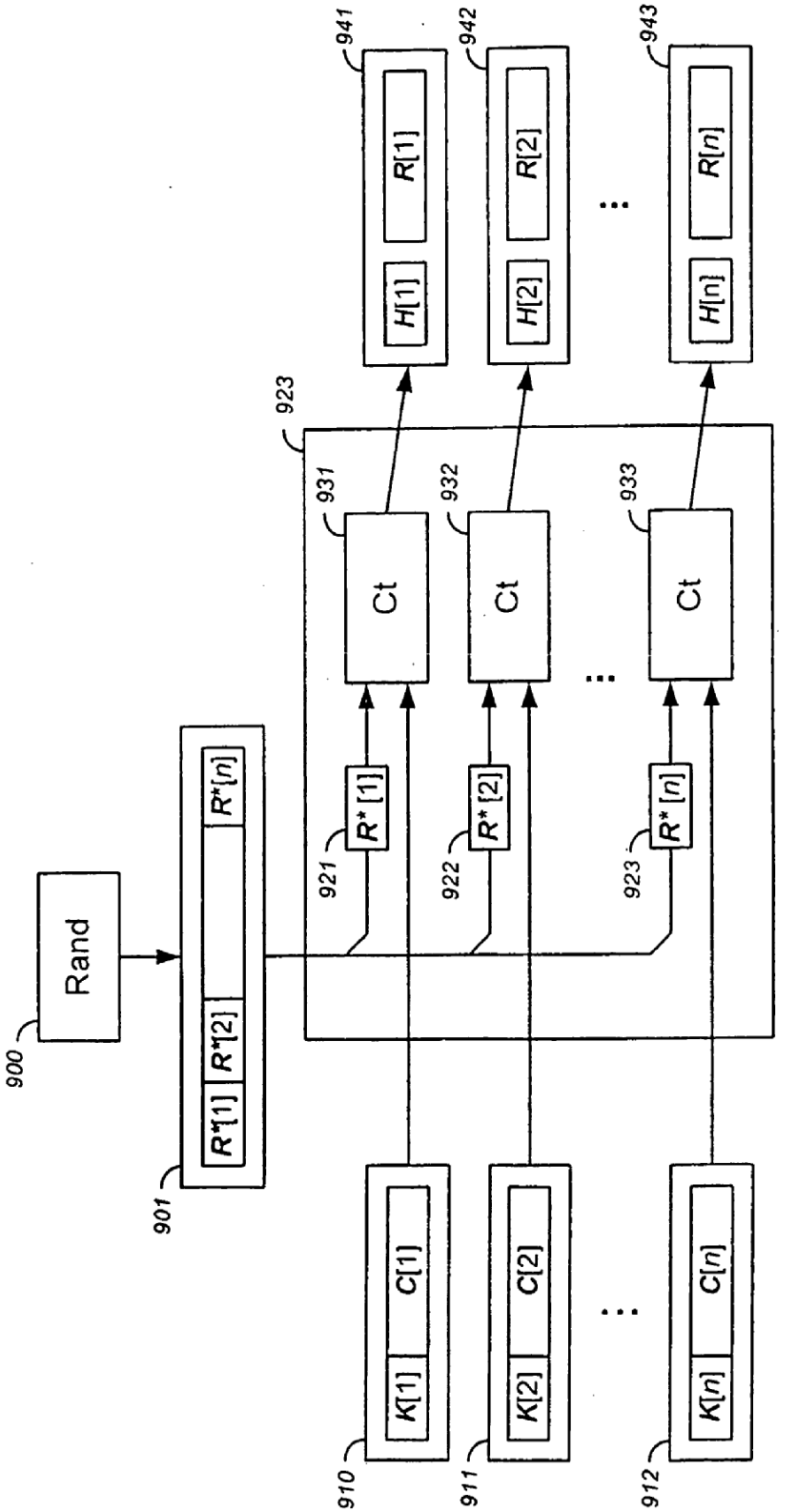
Rand
900
901
R*[1]
R*[2]
R*[n]
921
922
923
R* [1]
R* [2]
R* [n]
923
931
932
933
Ct
Ct
Ct
941
942
943
H[1]
H[2]
H[n]
R[1]
R[2]
R[n]
910
911
912
K[1]
K[2]
K[n]
C[1]
C[2]
C[n]

Figura 9

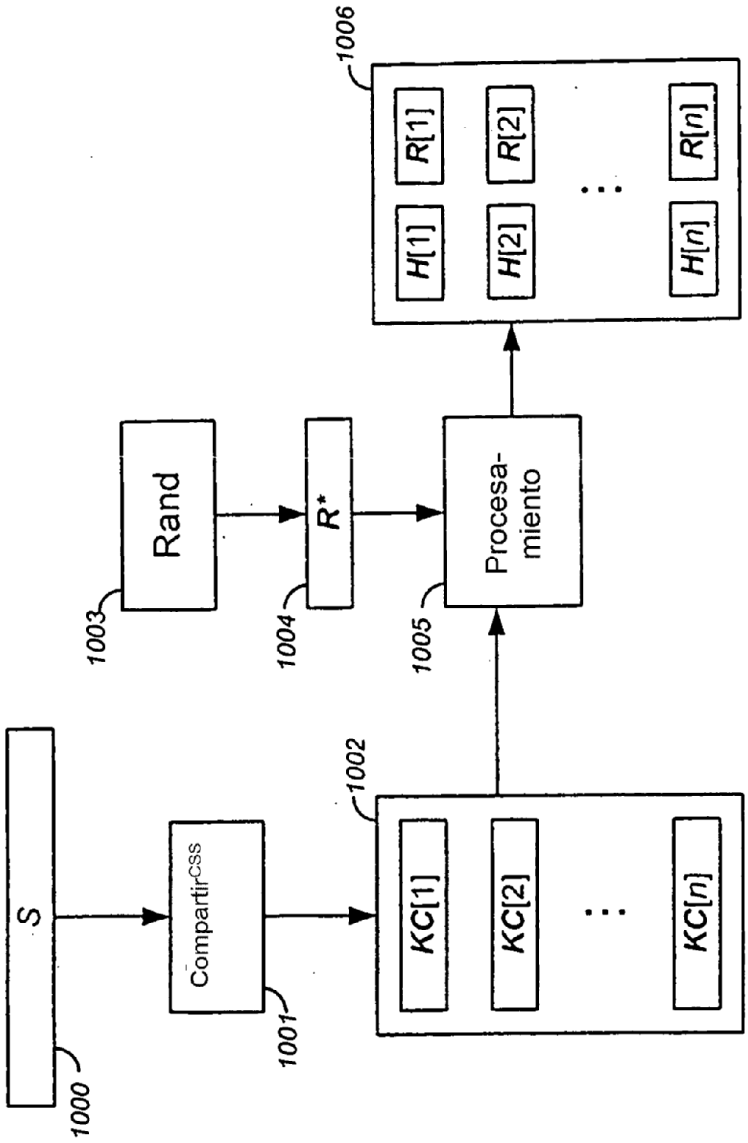
S
1000
CompartirCSS
1001
KC[1]
KC[2]
KC[n]
1002
Rand
1003
R*
1004
Procesa-
miento
1005
R[1]
R[2]
R[n]
H[1]
H[2]
H[n]
1006

Figura 10

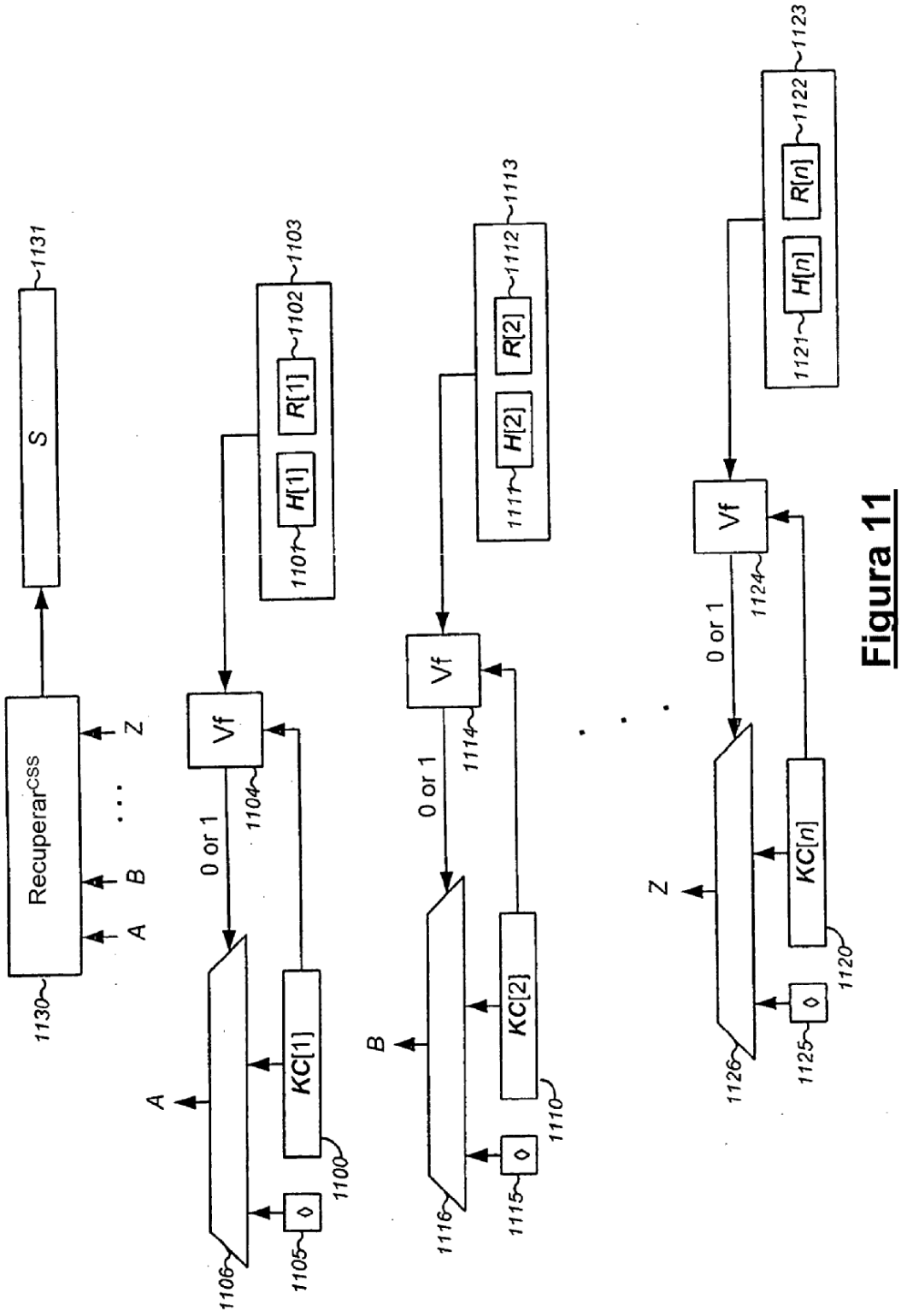
S
1131
Recuperar^CSS
1130
A
B
Z
Vf
0 or 1
1104
1103
H[1]
R[1]
1102
1101
KC[1]
1100
1105
0
1106
A
Vf
0 or 1
1114
1113
H[2]
R[2]
1112
1111
KC[2]
1110
1115
0
1116
B
Vf
0 or 1
1124
1123
H[n]
R[n]
1122
1121
KC[n]
1120
1125
0
1126
Z

Figura 11