(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 808 404**

(51) Int. CI.:

H04L 29/06 (2006.01)
H04L 9/32 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **04.01.2012 PCT/EP2012/050104**

(87) Fecha y número de publicación internacional: **12.07.2012 WO12093144**

(96) Fecha de presentación y número de la solicitud europea: **04.01.2012 E 12700036 (2)**

(97) Fecha y número de publicación de la concesión europea: **20.05.2020 EP 2661858**

(54) Título: **Método para comunicarse entre un servidor y un cliente y cliente, servidor y sistema correspondientes**

(30) Prioridad:

05.01.2011 EP 11305010

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
26.02.2021

(73) Titular/es:

THALES DIS FRANCE SA (100.0%)
6, rue de la Verrerie
92190 Meudon, FR

(72) Inventor/es:

LU, HONGQIAN KAREN y
POTONNIEE, OLIVIER

(74) Agente/Representante:

ELZABURU, S.L.P

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

Método para comunicarse entre un servidor y un cliente y cliente, servidor y sistema correspondientes

Campo de la invención:

La invención se refiere, de manera general, a un método para comunicarse entre un servidor y un cliente.

Además, la invención también pertenece a un servidor para comunicarse con un cliente.

Además, la invención también se refiere a un cliente para comunicarse con un servidor.

Finalmente, la invención también se refiere a un sistema de comunicación que comprende un servidor y un cliente.

Estado de la técnica:

Una solución conocida para la comunicación entre un servidor (web) y un cliente se basa en el envío, desde el cliente, de una solicitud de conexión al servidor que envía al cliente una página (web), como respuesta a la solicitud. El cliente es ejecutado por un ordenador principal. Una vez conectado al servidor, un usuario del ordenador tiene que cerrar una sesión de comunicación abierta (web) entre el servidor y el cliente presionando un botón dedicado.

Sin embargo, si el usuario del ordenador no presiona el botón, a continuación, la sesión de comunicación abierta no se cierra. Por lo tanto, la sesión de comunicación abierta aún está abierta para (un) usuario o usuarios y/o aplicación o aplicaciones no autorizados y posiblemente maliciosos.

Así, existe la necesidad de comunicarse, de manera segura, entre el servidor y el cliente. El documento US 2008/0086771 A1 describe una solución en la que un servidor y un cliente acceden a una clave de extensión de sesión y/o a una clave asociada adjunta, con el objetivo de comunicarse de forma segura.

Compendio de la invención:

La invención propone una solución para satisfacer la necesidad especificada anteriormente, proporcionando un método para comunicarse entre un servidor y un cliente, siendo definido el método en la reivindicación 1, y proporcionando un sistema correspondiente como se define en la reivindicación 5.

El principio de la invención consiste en una transmisión, desde el cliente al servidor, de datos que permiten al servidor controlar la sesión de comunicación abierta con el cliente y, si el servidor autentica al cliente, el servidor autoriza a continuar la sesión durante un período de tiempo de continuación predefinido.

De lo contrario, es decir, si el servidor no autentica al cliente, entonces el servidor le prohíbe al cliente continuar la sesión de comunicación abierta, es decir, la detiene.

Se ha de observar que la autenticación es posible gracias, bien a una clave, denominada clave de extensión de sesión, o bien a otra clave asociada con la clave de extensión de sesión compartida entre el servidor y el cliente y que permite al servidor y/o al cliente generar los datos autenticación esperados.

El método de la invención permite extender de forma segura una comunicación abierta entre el servidor y el cliente, es decir, cuando el servidor autentica de forma eficaz al cliente.

Un usuario de un dispositivo de ordenador principal, como un terminal, que ejecuta el cliente no necesita estar involucrado.

En particular, al contrario de la solución conocida especificada anteriormente en la presente memoria, el usuario del terminal no tiene que cerrar explícitamente la sesión de comunicación abierta, ya que el método de la invención permite cerrar automáticamente la sesión al no proporcionar al servidor datos que autentiquen particularmente al cliente.

Por lo tanto, el método de la invención es conveniente para el usuario que no necesita involucrarse.

Breve descripción de los dibujos:

Las características y ventajas adicionales de la invención serán más claramente comprensibles después de leer una descripción detallada de una realización preferida de la invención, dada como un ejemplo indicativo y no limitativo, junto con los siguientes dibujos:

- La fig. 1 ilustra un diagrama simplificado de una realización ejemplar de un sistema que comprende un Ordenador Personal (o PC), como terminal, que ejecuta un navegador (web), como cliente, una tarjeta con chip, como token, acoplado al PC y un servidor, adaptándose el sistema, una vez que se abre una sesión de comunicación entre el servidor y el cliente, para comunicarse de forma segura entre el servidor y el cliente mientras se involucra el token para generar los datos de autenticación esperados, según la invención; y

- La fig. 2 representa un ejemplo de un flujo de mensajes intercambiados entre el token, el terminal y el servidor del sistema de la fig. 1, de manera que el servidor autoriza o no al cliente a continuar, durante un período de tiempo de extensión predeterminado, la sesión de comunicación abierta gracias a los datos de autenticación enviados o no por el token, a través del cliente, al servidor respectivamente.

Descripción detallada:

En la presente memoria se considera un chip transportado por una tarjeta, como un chip medio y token.

Sin embargo, el chip puede ser transportado por cualquier otro medio que tenga diferentes factores de forma.

Dentro de la presente descripción, un token, también denominado elemento seguro, es un objeto electrónico destinado a, por un lado, comunicar datos con el mundo exterior y, por otro lado, llevar a cabo preferentemente al menos una operación de seguridad, tal como una protección de los datos que almacena el token.

En lugar de ser un chip transportado por una tarjeta, como medio, el token puede ser un Elemento Seguro Insertado, como un chip soldado, posiblemente de forma extraíble, en una Placa de Circuito Impreso (o PCB) de un dispositivo de ordenador principal.

En lugar de ser transportado por una tarjeta, el token puede ser transportado por otro medio, tal como un adaptador (dongle), por ejemplo, del tipo Bus Universal en Serie (o USB).

Naturalmente, la realización descrita en la presente memoria a continuación es solamente para fines de ejemplo y no se considera que reduzca el alcance de la presente invención.

La fig. 1 muestra esquemáticamente un sistema 10 para comunicarse entre un cliente ejecutado por un PC 14 y un servidor (web) ejecutado por un ordenador, también denominado servidor 18.

El sistema 10 incluye una tarjeta 12 con chip, el PC 14, como terminal, y el servidor 18.

En aras de la concisión, la tarjeta 12 con chip se denomina en lo sucesivo el token 12.

El token 12 está acoplado al PC 14.

En lugar de estar constituido por un PC, el terminal puede estar constituido, por ejemplo, por un teléfono móvil, un teléfono inteligente (es decir, un teléfono móvil con capacidad de Asistente digital personal (PDA)), un decodificador, una tableta, un ordenador, un ordenador portátil, un reproductor de video, un reproductor de audio, un reproductor multimedia, una consola de juegos, un ordenador portátil de dimensiones reducidas (netbook) y/o una PDA.

En aras de la simplicidad, solamente se ha representado un PC 14, como dispositivo cliente y terminal, que ejecuta una aplicación de navegador (web) y solamente un token acoplado al PC 14.

Sin embargo, varios PC, como dispositivos cliente, pueden comunicarse, al mismo tiempo o en diferentes momentos, con el servidor 18, con el objetivo de acceder a uno o diversos servicios proporcionados por el servidor 18.

El PC 14, como dispositivo de ordenador principal, puede acomodar una pluralidad de tokens.

La aplicación del navegador puede ser, como se conoce, una aplicación Microsoft Internet Explorer (marca registrada) o una aplicación Mozilla Firefox (marca registrada).

La aplicación del navegador del dispositivo cliente desempeña el papel de un cliente que intenta cargar una o diversas páginas (web) del servidor 18.

Según una alternativa, en lugar de la aplicación del navegador, el cliente es una aplicación propietaria, es decir, una aplicación a la que solamente se puede acceder mediante un distribuidor (o proveedor) de la aplicación considerada.

El servicio o servicios proporcionados por el servidor 18 pueden satisfacer las reglas de tipo de Word Wide Web Consortium (o W3C).

El PC 14 está conectado, a través de un contacto o un enlace 15 bidireccional sin contacto, al menos a una red de comunicaciones 16.

Como enlace de contacto, puede ser un enlace de cable.

Como enlace sin contacto, puede estar constituido por un enlace de tipo Wifi (marca registrada) o Bluetooth (marca registrada), como un enlace de radiofrecuencia con un alcance bastante corto (típicamente hasta 10 m).

El PC 14 incluye al menos un microprocesador (no representado), al menos una memoria (no representada) y al menos dos interfaces de Entrada/Salida (o I/O) (no representadas).

La interfaz de I/O del PC comprende preferiblemente una pantalla de visualización 142 y un teclado 144, como Interfaz hombre máquina (o MMI), con el objetivo de interactuar con un usuario de PC que desea beneficiarse del servicio o servicios suministrados por el servidor 18.

La interfaz de I/O del PC incluye una interfaz de I/O para intercambiar datos, mediante la red de comunicación 16, con el servidor 18. La interfaz de I/O del PC con el servidor 18 puede utilizar un Protocolo de Transferencia de Hipertexto (o HTTP), un Protocolo de Transferencia de Archivos (o FTP) y/u otro u otros protocolos de comunicaciones de datos, con el objetivo de comunicarse con el servidor 18.

El PC 14 está equipado con un lector de tarjetas (no representado), de manera que el PC 14 puede interactuar con el token 12 insertado dentro del lector de tarjetas.

La interfaz de I/O de PC también incluye una interfaz de I/O, tal como una interfaz de Ordenador Personal/Tarjeta Inteligente (o PC/SC), para intercambiar datos con el token 12.

Un componente de aplicación o de software ejecutado por el PC 14 intercambia con un componente de aplicación o de software ejecutado por el token 12 mientras que utiliza preferiblemente la Unidad de Datos de Protocolo de Aplicación (o APDU).

La interfaz de I/O del PC con el token 12 puede utilizar un Protocolo de Cable Único (o SWP), un Protocolo de Internet (o IP), un protocolo USB, un Protocolo Portador Independiente (o BIP) y/u otro u otros protocolos de comunicaciones de datos.

La memoria del PC almacena preferiblemente una aplicación de navegador y un motor de script.

La memoria del PC puede almacenar una clave privada asociada con una así llamada clave de extensión de sesión, como una clave pública asociada.

El microprocesador de PC procesa datos que se originan a partir de y/o están destinados a cualquier componente interno y datos que se originan y/o están destinados a cualquier dispositivo externo a través de una interfaz de I/O de PC.

El microprocesador del PC ejecuta particularmente, además de un Sistema Operativo (u OS), la aplicación del navegador, como cliente, y preferiblemente el motor de script, como intérprete de script, al que se puede acceder desde el cliente.

El microprocesador de PC ejecuta preferiblemente una función Hash criptográfica, como, por ejemplo, Algoritmo Hash Seguro (o SHA) -1, con el objetivo de generar una cadena de bits de tamaño fijo, como un valor hash criptográfico o resumen de mensaje (mensaje). La función Hash criptográfica permite reducir un tamaño de datos, tal como un valor aleatorio (como un desafío), como una entrada a la función Hash criptográfica. El resumen, como salida correspondiente de la función criptográfica Hash, gracias a su tamaño fijo, puede ser utilizado así por el token 12, como una entidad con capacidades de procesamiento de datos más débiles que los medios de procesamiento de datos del PC 14.

El microprocesador del PC ejecuta preferiblemente, después de la función Hash criptográfica, una función de formateo, como, por ejemplo, Métodos de Codificación para Firmas del Apéndice - Esquemas de Firma Probabilístico (o EMSA-PSS) como se define en PKCS # 1 v2.1, con el objetivo de formatear el resumen, como entrada a la función de formateo. La función de formateo puede ser una función para codificar la entrada en un formato interpretable por el token 12, cuando el token 12 soporta la aplicación de autenticación.

Un script es una cadena de caracteres y constituye datos ejecutables. El script está insertado dentro de una página (web), tal como una página que es escrita en un Lenguaje de marcas de hipertexto (o HTML). El script está constituido, por ejemplo, por un script de tipo Javascript (marca registrada), un ActiveX (marca registrada) o un Adobe (marca registrada).

El motor de script interpreta y ejecuta dinámicamente, es decir, durante la comunicación abierta con el servidor 18, al menos en parte el script.

Según una característica interesante de la invención, el PC 14, y más exactamente el cliente (aplicación), está adaptado para enviar al servidor 18 (aplicación) datos para autenticar, antes del servidor 18, al menos el cliente, como un interlocutor autorizado del servidor 18, con el objetivo de extender o prolongar una sesión de comunicación abierta con el servidor 18 durante un período de tiempo de extensión predefinido.

Cuando se permite que la sesión de comunicación se extienda por primera vez, la sesión de comunicación se extiende, una vez por el período de tiempo de extensión predefinido, a un tiempo de vencimiento correspondiente a un momento en que la sesión de comunicación se abre completada por un período de tiempo de extensión.

Cuando la sesión de comunicación se permite durante al menos otro momento, la sesión de comunicación se extiende al menos por el período de tiempo de extensión predefinido desde el último tiempo de vencimiento

calculado. La sesión de comunicación abierta puede así, extenderse periódicamente mientras el cliente envía al servidor 18 los datos de autenticación esperados.

Cualquier solicitud emitida por el cliente es procesada por el servidor 18 mientras no haya transcurrido el tiempo de vencimiento.

Tan pronto como el cliente, posiblemente con ayuda de otro componente de software y/u otra aplicación ejecutada por el PC 14 y/o el token 12, no envíe los datos de autenticación esperados, el servidor 18 cierra la sesión de comunicación abierta al final de un último tiempo de vencimiento calculado.

Para generar datos de autenticación, el PC 14 puede adaptarse para acceder preferiblemente a una clave privada asociada con la clave de extensión de sesión, como una clave pública asociada.

Según otra realización, en lugar de ser una clave pública, la clave de extensión de sesión es una clave secreta. La clave secreta puede compartirse previamente bien entre el servidor 18 y el cliente o bien generarse a partir de otros datos secretos compartidos entre el servidor 18 y el cliente. Los datos secretos pueden ser una clave madre, a partir de la cual se genera una clave hija gracias a un algoritmo predeterminado, tal como, por ejemplo, la Función de Derivación de Clave 1 (o KDF1) como se describe particularmente en http: //www.di-mgt .com.au/cryptoKDFs.html. Los datos secretos pueden generarse alternativamente mediante el uso de, bien una clave secreta combinada con un contador, como dos entradas a un algoritmo de generación, como, por ejemplo, un Código de Autenticación de Mensaje Basado en Hash (o HMAC) de Autenticación Abierta (o OATH) basándose en un algoritmo de Contraseña de un Solo Uso (o HOTP)/Solicitud de Comentarios del Grupo de Trabajo de Ingeniería de Internet (o IETF RFC) 4226, o una clave secreta combinada con una marca de tiempo, como dos entradas a un algoritmo de generación, como, por ejemplo, una algoritmo de contraseña de un solo uso basada en tiempo OATH (o TOTP).

Según una realización preferida, el cliente delega en el motor de scripts una interpretación de un script emitido por el servidor 18, con el objetivo de dirigir al token 12 datos ejecutables comprendidos dentro del script, destinados al token 12, y permitiendo al token 12 generar datos de autenticación, como un criptograma o como resultado de un cálculo criptográfico.

Alternativamente, el cliente delega en el motor de scripts una interpretación de un script emitido por el servidor 18, con el objetivo de dirigir a una aplicación de autenticación que se ha de ejecutar por los datos ejecutables del PC 14 comprendidos dentro del script, destinados al PC 14, y permitiendo al PC 14 generar datos de autenticación. La aplicación de autenticación soportada por el PC 14 puede estar constituida por una biblioteca. Contrariamente a la realización preferida en la que una aplicación de autenticación es soportada por un microprocesador de dispositivo, a saber, un microprocesador de token 12, diferente del microprocesador de PC 14, el PC 14 soporta la aplicación de autenticación dedicada a generar datos de autenticación. Antes de generar datos de autenticación, el PC 14 puede adaptarse para generar o dejar de generar un desafío utilizando datos secretos previamente compartidos con el servidor 18. Los datos secretos se pueden generar utilizando, bien una clave secreta combinada con un contador, como dos entradas a un algoritmo de generación, como, por ejemplo, un algoritmo OATH HOTP de IETF RFC 4226, o bien una clave secreta combinada con una marca de tiempo, como dos entradas a un algoritmo de generación, como, por ejemplo, un algoritmo OATH TOTP. El desafío se utiliza, a continuación, para generar los datos de autenticación correspondientes.

Según una realización preferida, el PC 14, y más exactamente el cliente, está adaptado para enviar al servidor 18 una solicitud, como una solicitud de desafío, para generar y obtener un desafío para cada extensión de tiempo de la sesión de comunicación abierta. Preferiblemente, el cliente delega en el motor de scripts una interpretación de un script emitido por el servidor 18, con el objetivo de dirigir al servidor 18 una solicitud de desafío comprendida dentro del script, y permitir que al servidor 18 generar un desafío y enviar de vuelta el desafío generado.

Alternativamente, el cliente delega en el motor de scripts una interpretación de un script emitido por el servidor 18, con el objetivo de dirigir al token 12 y los datos ejecutables del servidor 18 comprendidos dentro del script y permitir que al token 12 y al servidor 18 generar independientemente un desafío. Para generar un desafío, cada uno de los token 12 y el servidor 18 utilizan datos secretos previamente compartidos, como una clave secreta combinada con un contador (como se define en OATH HOTP/IETF RFC4226) o una marca de tiempo (como se define en OATH TOTP), y un algoritmo predeterminado, tal como un algoritmo de generación de Contraseña a Tiempo (u OTP), como un Código de Autenticación de Mensaje Basado en Hash (o HMAC) basado en OTP (o HOTP).

Para generar los datos de autenticación, el token 12 (o el PC 14) y el lado del servidor 18 pueden utilizar, además de un desafío generado, la clave de extensión de sesión o la clave asociada y un algoritmo predeterminado, información contextual.

El token 12 puede almacenar, preferiblemente de manera segura, la información contextual.

Como información contextual, puede haber información relacionada con el servidor 18, tal como un identificador del servidor 18, como su dirección IP y/o un nombre de dominio del servidor, y/o un desafío del usuario, tal como el éxito de una autenticación de usuario. La autenticación del usuario se realiza preferiblemente mediante el token 12 verificando, por ejemplo, si los datos introducidos por el usuario coinciden con los datos almacenados (y/o

generados) por el token 12, tal como un Número de Identidad Personal (o PIN) y/o impresión biométrica.

Según una realización preferida, el token 12 está adaptado para generar y proporcionar el script con datos de autenticación posiblemente mientras que utiliza un desafío generado, bien por el token 12 o bien por el servidor 18. Cuando lo genera el servidor 18, el servidor 18 proporciona además el token 12 con el desafío generado. Los datos de autenticación generados son suministrados a continuación por el token 12, mediante el script, al servidor 18.

El token 12 incluye un chip. El chip comprende al menos una memoria 122, al menos una interfaz 124 de I/O para comunicarse con el exterior del token 12 y al menos un microprocesador 126, como medio para procesar datos, que están vinculados internamente a través de un canal (bus) 123 de control y datos.

El token 12 almacena y lleva a cabo una o diversas funciones de seguridad.

Las funciones de seguridad pueden incluir un proceso de autenticación de usuario que ha de ser utilizado, con el fin de acceder a datos y/o aplicación o aplicaciones administradas por el token 12 y/o el servidor 18 que ha de ser direccionado.

Para autenticar al usuario, el token 12 puede almacenar una aplicación para verificar un PIN. El PIN se almacena de forma segura en la memoria 122 del chip y debe introducirse por medio de un usuario de token 12. El token 12 compara los datos de entrada con el PIN almacenado y, cuando los datos de entrada coinciden con el PIN almacenado, autoriza una ejecución de la aplicación, tal como una aplicación para autenticar un token 12.

Las funciones de seguridad incluyen preferentemente un proceso de encriptación/desencriptación. El proceso de encriptación/desencriptación se ha de utilizar para intercambiar datos, a través del PC 14, con el servidor 18. Antes de enviar cualquier dato, los datos se encriptan utilizando una clave y un algoritmo de encriptación.

Los algoritmos para encriptar/desencriptar datos se comparten entre el token 12 y el servidor 18.

El proceso de encriptación/desencriptación se ha de utilizar antes de enviar, a través del PC 14, al servidor 18, datos y, después de recibir, a través del PC 14, desde el servidor 18 datos respectivamente, con el objetivo de proteger un acceso a los datos así intercambiados.

La memoria 122 de chip puede estar constituida por una o diversas EEPROM (acrónimo de "Memoria de Solo Lectura Programable y Borrable Eléctricamente"), una o diversas ROM ("Memoria de Solo Lectura"), una o diversas memorias Flash, como memoria o memorias no volátiles, y/o cualquier otra memoria o memorias de diferentes tipos, como una o diversas RAM ("Memoria de Acceso Aleatorio"), como memoria o memorias volátiles.

La memoria 122 de chip almacena preferiblemente una clave privada que se utiliza para generar datos de autenticación. La memoria 122 de chip almacena una clave de extensión de sesión, como una clave pública asociada, que está asociada con la clave privada.

La memoria 122 de chip almacena preferiblemente, además de un sistema operativo (OS), al menos un algoritmo de aplicación de autenticación accesible desde el exterior, en particular, a través del script, desde el servidor 18. El algoritmo de aplicación de autenticación permite, cuando se ejecuta por el microprocesador 126 de chip, generar datos de autenticación y solicitar al PC 14 (y más exactamente, a través del script, al cliente) enviar uno (o diversos) mensaje(s) junto con los datos autenticación generados y posiblemente información adicional.

El algoritmo de la aplicación de autenticación se puede escribir en un lenguaje orientado a objetos, tal como Java, también denominado applet cuando se desarrolla en Java. Según tal realización correspondiente, la memoria 122 de chip almacena una Máquina Virtual Java (o JVM) que interpreta y ejecuta el applet.

Para generar los datos de autenticación, la aplicación de autenticación utiliza preferiblemente un desafío (generado, bien por el token 12 o bien por el servidor 18) y la clave privada asociada con la clave de extensión de sesión, como dos entradas a un algoritmo de encriptación, tal como un Estándar de Encriptación de Datos (o DES), un triple DES o un Rivest Shamir y Adleman (o RSA) con una longitud de clave seleccionada, como 1024 o más. Los datos de autenticación así generados, como un criptograma resultante, constituyen un desafío firmado también denominado firma del desafío.

Cuando el token 12 firma el desafío y envía al cliente (y más exactamente el script) la firma del desafío, asegura que el token 12 esté acoplado de manera eficaz al PC 14 y, por lo tanto, presente.

Tan pronto como el token 12 ya no esté presente, el cliente (gracias al script) ya no puede ser capaz de entregar datos de autenticación al servidor 18. Por consiguiente, el servidor 18 prohíbe que su interlocutor continúe la sesión de comunicación abierta entre el cliente y el servidor 18 enviando preferiblemente a su interlocutor un mensaje para indicar que la sesión de comunicación abierta ha finalizado.

El desafío se utiliza, para la realización descrita preferida, para generar los datos de autenticación correspondientes. Cuando el desafío lo proporciona el servidor 18 y ha de ser recibido por el token 12 antes de generar datos de autenticación. Por lo tanto, una solicitud correspondiente del desafío emitida por el cliente ha de enviarse antes del

tiempo de vencimiento. Cuando el tiempo de extensión ha de actualizarse al menos una vez mediante una solicitud adicional de otro desafío, el cliente sondea al servidor 18 con un valor del período de tiempo que es menor que el valor del período de tiempo de extensión. El valor del período de tiempo de tal sondeo puede ser preferiblemente igual al valor del período de tiempo de extensión. El sondeo se ha de ejecutar antes del tiempo de vencimiento menos un tiempo máximo estimado para recuperar el desafío y entregar los datos de autenticación correspondientes.

Como información adicional, puede haber un certificado. El certificado puede cumplir con un formato definido, por ejemplo, por las especificaciones de tipo X.509. El certificado contiene una clave de extensión de sesión, como clave pública. La clave privada se almacena de forma segura en la memoria 122 del chip. La clave privada es una clave asociada con la clave pública del certificado.

Como información adicional, puede haber información contextual. La memoria 122 de chip almacena, preferiblemente de manera segura, la información contextual, tal como un identificador del servidor 18, como destinatario del mensaje o mensajes que se han de enviar desde el cliente.

La interfaz 124 de I/O del chip incluye una interfaz de I/O para intercambiar datos con el ordenador principal, a saber, el PC 14, mientras se utiliza preferiblemente APDU.

La interfaz 124 de I/O del chip con el PC 14 puede utilizar un SWP, una IP, un protocolo USB, un BIP y/u otro u otros protocolos de comunicaciones de datos.

La interfaz 124 de I/O de chip puede incluir una interfaz o interfaces de I/O adicionales, con el objetivo de comunicarse con otra entidad o entidades respectivas externas.

El microprocesador 126 de chip procesa datos que se originan desde y/o están destinados a cualquier componente interno y datos que se originan desde y/o están destinados a cualquier dispositivo externo a través de la interfaz 124 de I/O de chip.

El microprocesador 126 de chip ejecuta, además de un sistema operativo (OS), preferiblemente una aplicación de autenticación que es accesible, es decir, que se puede iniciar, desde el PC 14, y más exactamente el script se ejecuta por el PC 14.

El microprocesador 126 de chip ejecuta, además del sistema operativo (OS) y la aplicación de autenticación, preferiblemente una aplicación de autenticación de usuario.

La aplicación de autenticación de usuario permite, cuando es ejecutada por el microprocesador 126 de chip, autenticar a un usuario de token al menos una vez, por ejemplo, después de una conexión al servidor 18 para iniciar sesión en el servidor 18. La autenticación del usuario del token verifica preferiblemente que el usuario del token ha proporcionado, a través de una MMI del PC, datos que coinciden con los datos esperados, tal como un PIN y/o impresión o impresiones biométricas, almacenadas previamente, preferiblemente de manera segura, dentro de la memoria 122 del chip.

El servidor 18 es remoto, es decir, es accesible a través de al menos una red de comunicaciones, tal como una red de Intranet, una red de Internet y/o una red de radiocomunicaciones móviles.

Según una alternativa, el servidor 18 es local. Por ejemplo, el servidor 18 está insertado dentro del PC 14.

El servidor 18 puede ser operado o administrado por un Operador de Red Móvil (o MNO), un Operador de Red Virtual Móvil (o MVNO), un Operador bancario, un operador de red de comunicaciones por cable, un Operador de servicios o en nombre de un Operador de servicios, como Un proveedor de servicios.

El servidor 18 comprende al menos una interfaz de I/O (no representada) para comunicarse con el exterior del servidor 18 y al menos un microprocesador (no representado), como medio para procesar datos, que están vinculador internamente entre sí.

El servidor 18 puede comprender al menos una memoria.

El servidor 18 almacena y lleva a cabo preferiblemente una o diversas funciones de seguridad. Las funciones de seguridad pueden incluir el envío a un token, al que se accede a través de un script insertado dentro de una página web, de una solicitud para llevar a cabo un proceso de autenticación de usuario que se ha de utilizar, con el fin de acceder a datos y/o a una aplicación o aplicaciones administradas por el token 12 y/o el servidor 18 que ha de ser direccionado.

Las funciones de seguridad incluyen preferentemente un proceso de encriptación/desencriptación. El proceso de encriptación/desencriptación se ha de utilizar para intercambiar datos, a través del PC 14, con el token 12. Antes de enviar cualquier dato, los datos se encriptan utilizando una clave y un algoritmo de encriptación.

Los algoritmos para encriptar/desencriptar datos se comparten entre el token 12 y el servidor 18.

El proceso de encriptación/desencriptación se ha de utilizar antes de enviar, a través del PC 14, al token 12, datos y, después de recibir, a través del PC 14, desde el token 12 datos respectivamente, con el objetivo de proteger un acceso a los datos así intercambiados.

La memoria del servidor almacena preferiblemente, además de un sistema operativo (OS), al menos un algoritmo de aplicación de autenticación accesible desde el microprocesador del servidor 18. El algoritmo de aplicación de autenticación permite, cuando el microprocesador del servidor 18 lo ejecuta, generar datos de autenticación, comparar los datos recibidos con los datos de autenticación generados y enviar preferiblemente uno (o diversos) mensaje(s) junto con un resultado de la comparación entre los datos recibidos y los datos de autenticación generados.

El servidor 18 está conectado preferiblemente, a través de un enlace 19 bidireccional, a una memoria 110.

La memoria 110 almacena una base de datos. La base de datos registra, al menos para cada sesión de comunicación abierta pendiente, un valor del último desafío enviado al cliente 14, un valor de una clave de extensión de sesión y un valor de un tiempo de vencimiento correspondiente.

La base de datos también registra, al menos para cada sesión de comunicación abierta pendiente, preferiblemente un valor de un identificador de la sesión de comunicación abierta, un valor de un desafío asociado con la sesión de comunicación abierta así identificada.

La clave de extensión de sesión se proporciona (o se genera por) preferiblemente al servidor 18 antes de una verificación de los datos recibidos del interlocutor.

El servidor 18 utiliza preferentemente el certificado que se recibe, a través del cliente, desde el token 12. El certificado recibido que contiene la clave de extensión de sesión, como clave pública, permite al servidor 18 determinar si la clave privada del token 12 se ha utilizado o no para generar un resultado criptográfico, como la firma del desafío.

El servidor 18 utiliza la clave de extensión de sesión para verificar si los datos recibidos de un interlocutor, como el cliente, para la sesión de comunicación abierta, coinciden con los datos de autenticación generados por el servidor 18.

Una coincidencia de los datos recibidos por el servidor 18 con los datos de autenticación, como datos de referencia, generados por el servidor 18, permite al servidor 18 extender la sesión de comunicación abierta con su interlocutor.

Para autenticar a su interlocutor, el servidor 18 utiliza la clave de extensión de sesión, como una entrada a un algoritmo de encriptación o desencriptación, tal como un DES, un DES triple o un RSA con una longitud de clave seleccionada, como 1024 o más. Los datos de autenticación así generados constituyen un criptograma resultante que también se ha de obtener y entregar por un interlocutor del servidor 18 para una comunicación adicional.

El servidor 18 tiene que recibir, de su interlocutor, datos que coincidan con los datos de referencia, de manera que el servidor 18 permita a su interlocutor debatir hasta que se determine un tiempo de vencimiento.

El servidor 18 está dispuesto a autorizar (o prohibir) que su interlocutor intercambie datos adicionales mientras genera un tiempo de vencimiento cada vez que el servidor 18 autentica a su interlocutor, como cliente. La autenticación del cliente significa que el servidor 18 verifica que los datos suministrados por su interlocutor corresponden a los datos de autenticación generados.

Para generar un tiempo de vencimiento, el servidor 18 añade preferiblemente a un tiempo en el que el cliente ha abierto una sesión de comunicación con el servidor 18 un período de tiempo de extensión predefinido. El servidor 18 activa un temporizador de sesión de comunicación preferiblemente tan pronto como un cliente se registra en el servidor 18.

Preferiblemente, el período de tiempo de extensión predefinido es corto. Tal período de tiempo de extensión predefinido corto obliga a un interlocutor, a saber, un cliente, a autenticarse regularmente ante el servidor 18, con el objetivo de poder continuar, de manera progresiva, comunicándose con el servidor 18.

El servidor 18 procesa cualquier solicitud emitida por el cliente mientras no haya transcurrido el tiempo de vencimiento. De lo contrario, es decir, cuando ha transcurrido el tiempo de vencimiento, el servidor 18 no procesa ninguna solicitud originada por el cliente.

Tan pronto como el cliente no entrega datos de autenticación esperados por el servidor 18, el servidor 18 cierra la sesión de comunicación abierta al final del último tiempo de vencimiento calculado.

Tal cierre puede deberse a la no entrega de datos de autenticación esperados o a una entrega de datos de autenticación inesperados. Así, incluso si un usuario de PC olvida activar un botón de "cierre de sesión" para cerrar la sesión de comunicación abierta, entonces el servidor 18 cierra automáticamente la sesión de comunicación abierta. La sesión de comunicación se detiene así y no se deja abierta para ningún usuario y/o aplicación (o

cualquier componente de software) no autorizado y posiblemente malicioso.

Cuando el servidor 18 autoriza a extender una sesión de comunicación abierta por primera vez, el servidor 18 extiende la sesión de comunicación abierta, una vez por el período de tiempo de extensión predefinido, a un tiempo de vencimiento correspondiente a un momento en el que la sesión de comunicación está abierta completada por un período de tiempo de extensión.

Cuando el servidor 18 autoriza a extender una sesión de comunicación abierta por al menos otro período de tiempo, la sesión de comunicación abierta se extiende al menos por el período de tiempo de extensión predefinido desde el último tiempo de vencimiento calculado. La sesión de comunicación abierta puede así extenderse además periódicamente mientras el cliente envía al servidor 18 los datos de autenticación esperados.

La fig. 2 representa un ejemplo de un flujo 20 de mensajes que involucra el token 12, el PC 14, como cliente y el servidor 18, durante un período de tiempo de extensión de una sesión de comunicación abierta entre el PC 14 y el servidor 18.

Se supone que un usuario de PC posee una tarjeta inteligente, como el token 12.

El PC 14 ejecuta un navegador, como cliente, para acceder al servidor 18, por ejemplo, más allá de un lanzamiento, por parte del usuario del PC, a través de una MMI del PC, de una ejecución del navegador.

Opcionalmente, una vez que el cliente está conectado al servidor 18, el servidor 18 involucra el token 12 utilizando una extensión de navegador, con el fin de que el token 12 autentique al usuario del token antes de que el cliente inicie sesión en el servidor 18. Tal solución, también conocida como SConnect y descrita particularmente en el documento US 7 748 609 B2, permite acceder, desde el servidor 18, a una función o funciones y datos del token 12. Para direccionar el token 12 desde el servidor 18, primero se instala una extensión del navegador, como aplicación, dentro del navegador. La extensión del navegador proporciona una Interfaz de Programación de Aplicaciones (o API) para comunicarse con el token 12. A continuación se puede emitir un script desde el servidor 18 e insertar dentro de una página servida por el servidor 18. El script trae a la página cargada desde el servidor 18 un código de aplicación que se comunica con el token 12 (utilizando la API de extensión del navegador) y el servidor 18.

Cuando el cliente 14 se conecta al servidor 18, el servidor 18 genera una cookie, como un identificador de la sesión de comunicación abierta. La cookie es una cadena de bits de información. Para generar una cookie, el servidor 18 puede utilizar un algoritmo de generación aleatorio. La cookie incluye un identificador del servidor 18, como direccionador de la cookie generada. El servidor 18 envía la cookie generada al cliente 14. El cliente 14, cuando se direcciona al servidor 18, tiene que utilizar la cookie generada cuando el cliente envía al servidor 18 cualquier dato durante la sesión de comunicación abierta. De lo contrario, el servidor 18 rechaza un acceso a su interlocutor que no proporciona la cookie. La cookie se utiliza como una evidencia que permite autenticar al cliente, como interlocutor del servidor 18.

Tan pronto como el cliente inicia sesión en el servidor 18, el servidor 18 envía una página que comprende un script para solicitar al servidor 18 otro script para extender una sesión de comunicación abierta. Los dos scripts se han de interpretar por un motor de scripts soportado por el PC 14 y ejecutado por el microprocesador del PC 14.

Una vez que el cliente 14 ha recibido el script para solicitar al servidor 18 otro script para extender una sesión de comunicación abierta, el cliente envía al servidor 18 una solicitud 22 para obtener un script para extender una sesión de comunicación abierta, tal como, por ejemplo, "Obtener script de extensión de sesión" (“Get sesión-extension script”).

Además de recibir la solicitud 22 para obtener un script para extender una sesión de comunicación abierta, el servidor 18 inicia el temporizador de sesión de comunicación. El temporizador de sesión de comunicación está configurado para contar desde cero hasta un valor del período de tiempo de extensión predeterminado. Alternativamente, el temporizador de sesión de comunicación está configurado para contar hacia atrás desde un valor del período de tiempo de extensión predeterminado.

El valor del período de tiempo de extensión predeterminado es corto, es decir, menor de unos pocos minutos. Preferentemente, el valor del período de tiempo de extensión predefinido es menor a un minuto. El valor del período de tiempo de extensión predefinido se establece, por ejemplo, en 30 s. La sesión de comunicación abierta se prolonga desde un valor del período de tiempo en el que la sesión de comunicación está abierta con el valor del período de tiempo de extensión.

A continuación, el servidor 18 envía de vuelta al cliente 14 una página que incluye un script 24 para extender una sesión de comunicación abierta.

Por ejemplo, el script 24 para extender una sesión de comunicación abierta es el siguiente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">

<html>
```

```
<head>
 <meta http-equiv="content-type" content="text/html; charset=UTF-8">
  < script src="sconnect.js"></script>
  <script src="sha1.js"></script>
  <script src="padding.js"></script>
  <script>
    //  Obtener referencia Sconnect
    var scardComm = new Sconnect.PCSC();
    // esta función inicia el diálogo de extensión de sesión con la función del servidor extendSession(){
      // enviar una solicitud de desafío al servidor
      var challengeRequest = new XMLHttpRequest ();
      challengeRequest.open("GET",http://www.myserver.com/session/getchallenge");
      challengeRequest.onreadystatechange = function() {
        if ((challengeRequest.readyState == 4) && (challengeRequest.status == 200 )) {
          // hemos recibido el desafío del servidor, ahora tenemos que firmarlo
          // calcular el resumen SHA-1 de la var hashedRandom = Sha1.digest (challengeRequest.responseText) aleatoria;
          var paddedHash = encodeHashForSigning(hashedRandom);
          // enviar tarjeta inteligente APDU para firmar (RSA-1024 y Key 9C)
          var resp = scardComm.exchangeAPDU("00879C887C81858200818180" + paddedHash);
          if (resp.statusWord == "9000") {
            // ahora tenemos la firma, enviarla al servidor:
            var authRequest = new XMLHttpRequest ();
            // añadir datos de autenticación a la URL de solicitud HTTP authRequest.open("GET", "http://www.myserver.com/session/authData?data="+ resp.dataOut);
            authRequest.onreadystatechange = function () {
              if ((authRequest.readyState == 4) && (authRequest.status == 200))
                  // ¡Éxito! extendemos la sesión
                  // Registre la siguiente extensión de sesión en 30 segundos
                  setTimeout ("extendSession ()", 30000);
               }
            }
          }
        }
      }
  }
 // conectarse a la tarjeta insertada
```

```
 if (scardComm.connect(listReaders(1)[0])) {
  // seleccionar la aplicación de firma de la tarjeta
  scardcomm.exchangeAPDU("00A404000BA000000308000010000100");
  extendSession ();
 }
 </script>
 </head>
<body>
 Has iniciado sesión
 </body>
</html>
```

El script 24 para extender una sesión de comunicación abierta puede incluir las siguientes operaciones a ser ejecutadas por el microprocesador del PC 14:

- un envío, al servidor 18, de una solicitud de desafío,
- una recepción de un desafío (procedente del servidor 18),
- una determinación de un desafío hash,
- un formato del desafío hash,
- un envío, al token 12, de un comando APDU para solicitar al token 12 para firmar un desafío hash formateado,
- una recepción de una firma correspondiente (procedente del token 12),
- un envío, al servidor 18, de una solicitud de datos de autenticación junto con la firma correspondiente,
- una recepción de un resultado (procedente del servidor 18) relacionado con la solicitud de autenticación enviada,
- una configuración de un temporizador de vencimiento a un valor del período tiempo de vencimiento predefinido, solamente si el resultado corresponde a un éxito, es decir, la autenticación de la firma enviada coincide con los datos de autenticación esperados generados por el servidor 18, y
- en tal caso, las operaciones enumeradas anteriormente se repetirán al menos una vez después de la expiración del temporizador de vencimiento.

El temporizador de vencimiento está configurado para contar desde cero hasta el valor del período de tiempo de vencimiento predefinido. Alternativamente, el temporizador de vencimiento está configurado para contar hacia atrás desde el valor del tiempo de vencimiento predefinido hasta cero.

El valor del período de tiempo de vencimiento predefinido es corto como, por ejemplo, 30 s.

Antes de las operaciones enumeradas anteriormente, puede ser requerido seleccionar la aplicación de autenticación que se ha de ejecutar en el token 12.

Antes del vencimiento del valor del período de tiempo de vencimiento, el cliente 14 envía al servidor 18 una solicitud 26 para obtener un desafío, tal como, por ejemplo, "Obtener un nuevo desafío".

Una vez que el servidor 18 recibe la solicitud 26 para obtener un desafío, el servidor 18 genera, por ejemplo, un valor aleatorio, como un desafío. El desafío puede ser alternativamente un resultado de un algoritmo de generación de OTP, tal como un OTP basado en HMAC.

El servidor 18 almacena el último desafío generado. El servidor 18 también puede almacenar un tiempo en el que se solicita el desafío.

El servidor 18 envía de vuelta al cliente 14 el último desafío 28 generado, como respuesta a la solicitud 26 para obtener un desafío, con el objetivo de permitir que su interlocutor suministre al servidor 18 los datos de autenticación correspondientes.

Tan pronto como el cliente 14 recibe el desafío, el cliente 14 determina un desafío hash, con el objetivo de adaptar el tamaño del desafío generado a un tamaño suficientemente reducido. Tal tamaño reducido del desafío generado permite que el desafío generado se procese por el token 12 (que generalmente ha reducido la capacidad de procesamiento de datos con respecto a un ordenador terminal). A continuación, el cliente 14 formatea el desafío hash resultante. El cliente 14 obtiene así un desafío hash formateado.

A continuación, el cliente 14 envía al token 12 un comando APDU 210 para solicitar al token 12 que firme el desafío hash formateado.

Opcionalmente, el cliente 14 envía al token 12 información contextual adicional, por ejemplo, información relacionada con el servidor 18, tal como un identificador del servidor 18, como su dirección IP, su nombre de dominio y/o su nombre. Tal envío de información contextual adicional se puede realizar concatenando a la información relacionada con el desafío la información contextual adicional.

Cuando se acopla al PC 14, el token 12 genera una firma correspondiente, como datos de autenticación, encriptando el desafío hash formateado mientras que utiliza un algoritmo de encriptación y una clave privada relacionada con el token 12. La firma puede cumplir con un estándar, tal como PKCS 1 como se describe en el siguiente sitio http://www.rsa.com/rsalabs/node.asp?id = 2125.

El token 12 envía de vuelta al cliente 14 la firma 212 generada, como datos de autenticación, acompañada de un certificado que incluye una clave pública, como clave de extensión de sesión, correspondiente a la clave privada almacenada dentro del token 12.

La clave de extensión de sesión es preferiblemente una clave dedicada. Según otra realización, la clave de extensión de sesión es una clave para iniciar sesión en el servidor 18.

Alternativamente, en lugar de una clave pública, la clave de extensión de sesión es una clave secreta que, bien se comparte previamente con el token 12 o bien se genera por el servidor 18 y por el token 12 mientras utiliza los mismos datos.

Alternativamente, en lugar de proporcionar la clave de extensión de sesión y los datos de autenticación al mismo tiempo, el token 12 transmite la clave de extensión de sesión, a través del cliente 14, al servidor 18 antes de una transmisión de los datos de autenticación. Puede producirse una transmisión de la clave de extensión de sesión, por ejemplo, durante una fase de inicio de sesión entre el cliente 14 y el servidor 18.

Según otra realización, en lugar del token 12, otro servidor, como sistema de soporte, proporciona al servidor 18 la clave de extensión de sesión antes de una verificación (llevada a cabo por el servidor 18) de los datos recibidos con respecto a los datos de autenticación esperados.

El cliente 14 recibe una firma generada correspondiente y el certificado. El cliente 14 almacena el certificado.

A continuación, el cliente 14 envía, al servidor 18, una solicitud 214 para autenticar datos acompañados con la firma generada correspondiente y el certificado.

El servidor 18 (y más exactamente una aplicación de autenticación ejecutada por el servidor 18) recibe la firma generada correspondiente y el certificado. El servidor 18 extrae del certificado la clave de extensión de sesión.

El servidor 18 analiza si los datos recibidos (o no) de su interlocutor corresponden o no con los últimos datos de autenticación almacenados, con el objetivo de permitir o bloquear un acceso adicional al servidor 18.

Si el token 12 ya no está o no está acoplado al PC 14, el servidor 18 no recibe ningún dato, a través del cliente 14, del token 12, a continuación, el servidor 18 cierra automáticamente la sesión de comunicación abierta. La sesión de comunicación abierta se detiene, reduciendo, por lo tanto, un riesgo de ser atacado por un usuario y/o software no autorizado.

El servidor 18 desencripta preferentemente datos, como firma de un desafío, recibidos de su interlocutor, utilizando la clave de extensión de sesión. El servidor 18 obtiene datos desencriptados para compararlos con el desafío almacenado generado, como datos de autenticación. Si los datos desencriptados son el desafío generado, entonces la autenticación tiene éxito y el servidor 18 autoriza a continuar un intercambio de datos con su interlocutor generando otro desafío y reiniciando un temporizador de sesión de comunicación.

Cuando la autenticación tiene éxito, el servidor 18 permite guardar dentro de la base de datos almacenada dentro de la memoria 110 al menos el identificador de la sesión de comunicación abierta asociada con el último desafío generado y la última clave de extensión de sesión recibida, como datos registrados.

De lo contrario, es decir, si, bien el servidor 18 no recibe datos o bien los datos desencriptados no son el desafío generado, el interlocutor no se autentica y el servidor 18 prohíbe intercambiar más datos con su interlocutor cerrando automáticamente la sesión de comunicación abierta.

Basándose en un resultado de la comparación llevada a cabo por el servidor 18, el servidor 18 envía preferiblemente de vuelta al cliente 14 un mensaje 216 que incluye un resultado de comparación, es decir, bien la autenticación tiene éxito o bien la autenticación ha fallado.

Mientras que el usuario del PC necesita intercambiar, a través del cliente 14, con el servidor 18, y el token 12 está presente, el PC 14 (microprocesador) ejecuta el script para extender una sesión de comunicación abierta repitiendo las operaciones mencionadas anteriormente en la presente memoria. Más exactamente, el PC 14 solicita y obtiene periódicamente del servidor 18 un nuevo desafío, solicita y obtiene, del token 12, una firma correspondiente y entrega la firma al servidor 18 y recibe el correspondiente resultado de autenticación con éxito. Para cada extensión de tiempo autorizada, el servidor 18 calcula un valor de tiempo de vencimiento correspondiente de la sesión de comunicación abierta añadiendo al último valor de tiempo de vencimiento un valor del período de tiempo de extensión predeterminado. El cliente 14 y el servidor 18 intercambian, por un lado, solicitudes 26 adicionales para obtener un desafío y 214 para autenticar datos y las respuestas 28 y 216 correspondientes y, por otro lado, el cliente 14 y el token 12 intercambian comandos 210 adicionales por solicitar que el token 12 firme el desafío hash formateado y la respuesta 212 correspondiente.

Una vez que se alcanza el último valor de tiempo de vencimiento generado (almacenado en el lado del servidor 18) para la sesión de comunicación abierta, el servidor 18 no responde a ninguna solicitud adicional procedente del cliente 14. El servidor 18 puede enviar al cliente 14 de vuelta un mensaje para informar a este último que la sesión de comunicación abierta llega al final de la sesión. El final de la sesión de comunicación es el último valor de tiempo de extensión. El último valor de tiempo de extensión es, bien el primer valor de tiempo de extensión de sesión o bien un valor de extensión de sesión adicional.

El servidor 18 verifica durante la sesión de comunicación abierta si se recibe una solicitud procedente del cliente 14 antes de que el temporizador de la sesión de comunicación llegue al último tiempo de vencimiento generado. En caso afirmativo, el servidor 18 responde a la solicitud recibida.

Gracias a la invención, no es necesario confiar en el cliente 14 para mantener una sesión de comunicación abierta entre el cliente 14 y el servidor 18. De hecho, el servidor 18 cierra automáticamente la sesión de comunicación abierta tan pronto como el cliente 14 no entrega los datos de autenticación esperados.

Se pueden introducir muchas enmiendas de la realización descrita anteriormente sin desviarse del espíritu de la invención.

Por ejemplo, según una primera realización, una vez se ha iniciado sesión en el servidor 18, el script de extensión de sesión se incorpora en cada página servida por el servidor 18. Sin embargo, tal primera realización tipo implica establecer una comunicación entre el PC 14 y el token 12 para que cada página se ralentice y así, disminuya la representación de cada página.

Según una segunda realización, una vez que se ha iniciado sesión en el servidor 18, el script de extensión de sesión se incorpora solamente a una página cargada desde el servidor 18 y se mantiene abierta durante la sesión de comunicación abierta. Tal uso de una sola página, como una ventana emergente, permite minimizar la comunicación de datos entre el PC 14 y el token 12. La ventana emergente puede incluir un botón de "cerrar sesión" dedicado que permite, cuando se presiona, cerrar explícitamente la sesión de comunicación abierta. Para abrir la ventana emergente durante el inicio de sesión en el servidor 18, puede ser un script window.open("http://www.myserver.com/extend-session","session-window").

REIVINDICACIONES

1. Un método (20) para comunicarse entre un servidor (18) y un cliente (14), accediendo el servidor y el cliente al menos a una clave de extensión de sesión o a una clave asociada con la clave de extensión de sesión, como una clave asociada,

caracterizado por que el método comprende las siguientes operaciones:

- el servidor envía, al cliente, un script (24);

- el cliente envía, al servidor, al menos una solicitud (26) de desafío;

- el servidor genera, basándose en la solicitud de desafío, un desafío;

- el cliente que ejecuta el script recibe el desafío;

- el cliente que ejecuta el script envía, a una aplicación de autenticación, el desafío (210);

- la aplicación de autenticación genera datos de autenticación utilizando el desafío, la clave de extensión de sesión o la clave asociada, y el algoritmo predeterminado;

- la aplicación de autenticación envía, al cliente que ejecuta el script, los datos (212) de autenticación;

- el servidor recibe, del cliente que ejecuta el script, los datos (214) de autenticación;

- el servidor autentica con éxito al menos al cliente sobre la base de los datos de autenticación recibidos, el desafío generado y la clave de extensión de sesión o la clave asociada;

- el servidor genera un tiempo de vencimiento, siendo el tiempo de vencimiento un tiempo en el que una sesión de comunicación está abierta completada por un período de tiempo de extensión predeterminado; y

- el servidor autoriza a extender la sesión de comunicación abierta con el cliente autenticado hasta el tiempo de vencimiento.

2. Método según la reivindicación 1, en donde el cliente y el servidor utilizan además de la clave de extensión de sesión o la clave asociada, el desafío y un algoritmo predeterminado, información contextual, con el objetivo de generar los datos de autenticación correspondientes.

3. Método según la reivindicación 1 o 2, en donde un token (12) que se acopla al terminal (14), almacenando el token la clave de extensión de sesión o la clave asociada, el token soporta la aplicación de autenticación.

4. Método según cualquiera de las reivindicaciones 1 a 3, en donde el cliente que ejecuta el script envía al servidor, además de los datos de autenticación, un identificador de la sesión de comunicación abierta.

5. Un sistema (10) de comunicaciones que comprende un servidor (18) y un cliente (14), comprendiendo el servidor y el cliente medios para acceder al menos a una clave de extensión de sesión o a una clave asociada con una clave de extensión de sesión, como una clave asociada,

caracterizado por que el sistema está adaptado para llevar a cabo el método de las operaciones de la reivindicación 1.

6. Sistema de comunicaciones según la reivindicación 5, en donde el cliente incluye o está acoplado a un token.

7. Sistema de comunicaciones según la reivindicación 6, en donde, el token que comprende medios para almacenar la clave de extensión de sesión o la clave asociada, el token comprende la aplicación de autenticación.

8. Sistema de comunicaciones según la reivindicación 6 o 7, en donde el token incluye al menos un elemento de un grupo que comprende:

- un elemento seguro;
- una tarjeta con chip;
- un adaptador (dongle);
- un adaptador (dongle) USB.

9. Sistema de comunicaciones según cualquiera de las reivindicaciones 5 a 8, en donde el cliente incluye un terminal.

10. Sistema de comunicaciones según la reivindicación 9, en donde el terminal incluye al menos un elemento de un

grupo que comprende:

- un PC;
- un teléfono inteligente;
- una PDA
- un decodificador;
- una tableta, un ordenador;
- un ordenador portátil;
- un reproductor de video;
- un reproductor de audio;
- un reproductor multimedia;
- una consola de juegos;
- un ordenador portátil de dimensiones reducidas (netbook)

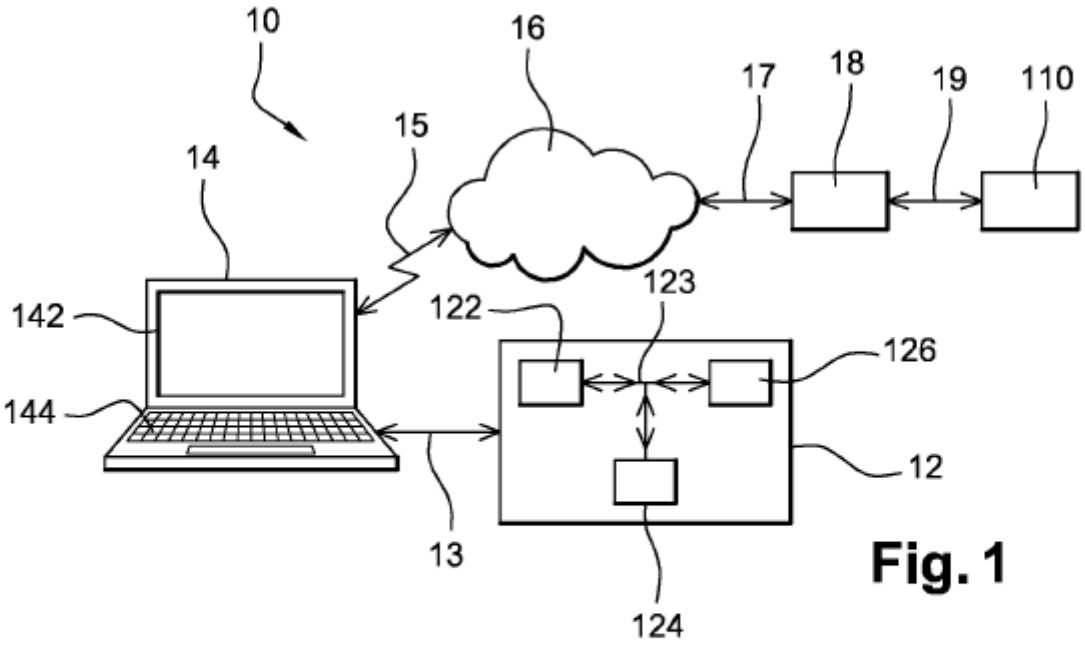
10
16
17
18
19
110
15
14
142
122
123
126
144
12
13
124

Fig. 1

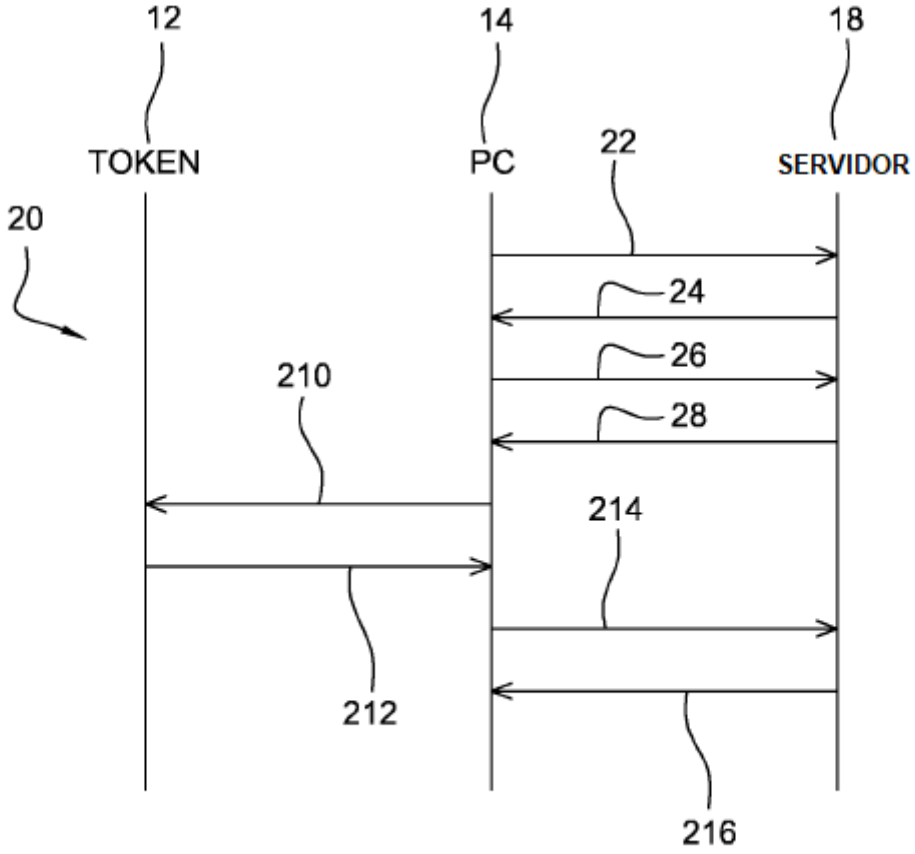
12
TOKEN
14
PC
18
SERVIDOR
22
20
24
26
210
28
214
212
216

Fig. 2