(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 426 946**

(51) Int. CI.:

***H04L 9/32*** (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **09.04.2008 E 08734046 (9)**

(97) Fecha y número de publicación de la concesión europea: **19.06.2013 EP 2209253**

(54) Título: **Un método, sistema, servidor y terminal para poner en práctica una autenticación**

(30) Prioridad:

**08.11.2007 CN 200710170309**
**27.11.2007 CN 200710195462**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
**25.10.2013**

(73) Titular/es:

**HUAWEI TECHNOLOGIES CO., LTD. (100.0%)**
**Huawei Administration Building Bantian**
**Longgang District, Shenzhen**
**Guangdong 518129 , CN**

(72) Inventor/es:

**CHAI, XIAOQIAN;**
**GAO, HONGTAO;**
**LI, KEPENG y**
**TIAN, LINYI**

(74) Agente/Representante:

**LEHMANN NOVO, María Isabel**

## DESCRIPCIÓN

Un método, sistema, servidor y terminal para poner en práctica una autenticación

La presente invención se refiere a tecnologías de comunicaciones y en particular, a un método de autenticación basado en un protocolo de Sincronización de Datos (DS) y un protocolo de Gestión de Dispositivo (DM) y a un sistema, a un servidor y un cliente.

ANTECEDENTES DE LA INVENCIÓN

El Lenguaje de Marcado de Sincronización (SyncML) es un protocolo desarrollado para sincronizar información personal y datos intra-empresa entre múltiples plataformas y redes. El protocolo SyncML define una serie de operaciones entre las entidades participantes y define un conjunto de formatos de mensajes para realizar dichas operaciones. Sobre la base de SyncML, la denominada Alianza Móvil Abierta (OMA) desarrolla el protocolo DS y el protocolo DM.

El protocolo DS puede sincronizar información personal y datos intra-empresa entre múltiples plataformas y redes. El protocolo DS se suele aplicar a la sincronización de datos entre un dispositivo móvil o servidor de aplicación y un servidor de red o la sincronización de datos entre dos Ordenadores Personales (PCs).

El protocolo DM es una solución de gestión a distancia rentable que descarga datos de instrucciones de gestión desde la red al cliente y permite al cliente ejecutar la instrucción de gestión automáticamente para actualizar, configurar y diagnosticar el software y hardware del cliente. Además, el DM transfiere la información de servicio requerida por el operador y la información sobre las funciones del cliente desde el cliente al servidor, con lo que se soporta la operación de otros servicios.

Un mecanismo de autenticación de seguridad similar se aplica al protocolo DS y al protocolo DM para la autenticación del servidor y del cliente de forma efectiva, según se ilustra en la Figura 1:

Etapa 101: El servidor envía un mensaje de disparo operativo al cliente para iniciar una sesión;

El mensaje de disparo operativo transmite: un extracto de acceso denominado Digest generado utilizando un número aleatorio del servidor (*s_nonce*) y la información de disparo operativo (TriggerInfo). El mensaje de disparo operativo puede soportarse en un mensaje corto u otro mensaje de tipo denominado Push.

El *s_nonce* es un número aleatorio (*nonce*) generado por el cliente y disponible para el servidor.

Etapa 102: El cliente envía una demanda de sesión al servidor.

Después de recibir el mensaje de disparo operativo, el cliente utiliza el *s_nonce* memorizado para generar información del extracto de acceso Digest y para autenticar el mensaje Trigger de disparo operativo. Si la autenticación se realiza satisfactoriamente, el cliente envía una demanda de sesión al servidor para iniciar una sesión.

La demanda de sesión transmite: un identificador de sesión SessionID e información de autenticación (Authenticate) del cliente. La información de autenticación es un Digest generado utilizando el *número aleatorio* del cliente (*c_nonce*).

El *c_nonce* se genera por el servidor y está disponible para el cliente.

En esta etapa, se establece una conexión de sesión entre el cliente y el servidor.

Etapa 103: El servidor reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación.

En conformidad con la información de autenticación enviada por el cliente, el servidor realiza la autenticación del cliente y luego, reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación del servidor.

Más concretamente, la respuesta transmite: un resultado de la autenticación por el servidor del cliente, un identificador SessionID y la información de autenticación del servidor (esto es, un Digest generado utilizando el *s_nonce*).

Etapa 104: El cliente reenvía un mensaje que transmite el resultado de la autenticación al servidor.

En conformidad con la información de autenticación enviada por el servidor, el cliente realiza la autenticación del servidor y luego, reenvía un mensaje que transmite el resultado de la autenticación al servidor.

Más concretamente, este mensaje transmite: un resultado de la autenticación del cliente del servidor y otra información pertinente.

Si el servidor realiza la autenticación del cliente, de forma insatisfactoria, o el cliente realiza la autenticación del servidor también de forma insatisfactoria, a modo de ejemplo, la contraseña es incorrecta o el valor del número aleatorio *nonce* es incorrecto, el servidor o el cliente puede enviar una demanda a la parte opuesta directamente para realizar la autenticación de nuevo.

Si el servidor conoce que el *s_nonce* utilizado en el mensaje Trigger es incorrecto, a modo de ejemplo, si el servidor no recibe ninguna respuesta normal desde el cliente después de enviar repetidamente el mensaje Trigger, el servidor cree que el *s_nonce* es incorrecto y genera el Digest del mensaje Trigger utilizando un número aleatorio por defecto "0x00000000". Después de la autenticación insatisfactoria del mensaje Trigger, en conformidad con el Digest generado utilizando el *s_nonce,* el cliente utiliza el número aleatorio por defecto para generar un Digest y para realizar una nueva autenticación del mensaje Trigger. Si la autenticación es satisfactoria, el número aleatorio *nonce* por defecto se utiliza para la autenticación del servidor y del cliente y luego, se actualizan *s_nonce* y el *c_nonce*. El proceso de actualización se ilustra en la Figura 2:

Etapa 201: El servidor envía un mensaje Trigger al cliente para iniciar una sesión.

Después de determinar que el *s_nonce* anterior es incorrecto, el servidor utiliza el número aleatorio por defecto para generar un mensaje Trigger y envía el mensaje al cliente. El mensaje Trigger transmite: el Digest generado utilizando el número aleatorio por defecto y la información TriggerInfo.

Etapa 202: El cliente realiza la autenticación del mensaje Trigger de forma insatisfactoria y utiliza el número aleatorio *nonce* por defecto para una nueva autenticación.

Después de recibir el mensaje Trigger, el cliente utiliza el *s_nonce* memorizado para realizar la autenticación del mensaje Trigger. Si la autenticación falla por cualquier motivo, el cliente utiliza el número aleatorio por defecto para una nueva autenticación del mensaje Trigger.

Si la autenticación es satisfactoria, ello indica que el *s_nonce* anteriormente utilizado por el servidor es incorrecto y el cliente envía una demanda de sesión al servidor.

Etapa 203: El cliente envía una demanda de sesión al servidor.

Después de la autenticación satisfactoria utilizando el número aleatorio por defecto, el cliente envía una demanda de sesión al servidor para iniciar una sesión.

La demanda de sesión transmite: el identificador SessionID y el Digest generado utilizando el número aleatorio *nonce* por defecto.

Etapa 204: El servidor reenvía una respuesta que transmite el resultado de la autenticación, la demanda de autenticación y la orden para actualizar *c_nonce*.

El servidor realiza la autenticación del cliente utilizando el número aleatorio *nonce* por defecto y luego, reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación al cliente.

Más concretamente, la respuesta transmite: un resultado de la autenticación por el servidor del cliente, una demanda de actualización de *c_nonce* y un Digest generado utilizando el número aleatorio *nonce* por defecto.

Etapa 205: El cliente reenvía un mensaje que transmite el resultado de la autenticación y la orden para actualizar *s_nonce* al servidor.

El cliente realiza la autenticación del servidor utilizando el número aleatorio por defecto. Una vez realizada satisfactoriamente la autenticación, el cliente actualiza el *c_nonce* y luego, reenvía un mensaje que transmite el resultado de autenticación y la orden para actualizar *s_nonce* al servidor.

Etapa 206: El servidor reenvía un resultado de actualización de *s_nonce* al cliente.

En el proceso de desarrollo de la presente invención, el inventor encuentra al menos los siguientes defectos en la técnica anterior:

El número aleatorio *nonce* por defecto se utiliza para la autenticación en el caso de que *s_nonce* sea incorrecto. El número aleatorio por defecto es un valor fijo abierto y el servidor, operativamente malicioso, puede interceptar el mensaje que utiliza el número aleatorio por defecto y enviar el mensaje repetidamente para atacar al servidor o al cliente.

En la técnica anterior, dos valores de *número aleatorio* se utilizan en una sola sesión: *s_nonce* y *c_nonce*, que se generan y actualizan por el servidor y el cliente respectivamente, con lo que se impone una carga de gestión importante en el cliente y en el servidor.

El documento WO 02/25899 A da a conocer un sistema en el que un cliente/servidor/red puede poner en práctica una sesión de gestión de claves cuando el servidor inicia la sesión de gestión de claves utilizando un número aleatorio. El *número aleatorio* permite que se transmita un mensaje de disparo operativo o de salida del estado de latencia al cliente, de modo que pueda evitarse un ataque de servicio sobre el servidor cuando se recibe un número aleatorio falso por el servidor con un mensaje de demanda de AP. De este modo, el servidor puede rechazar los mensajes de demanda de AP que no vayan acompañados por un número aleatorio memorizado por el servidor. El método puede ponerse en práctica mediante circuitos, señales eléctricas y un código para realizar los actos descritos en el método.

SUMARIO DE LA INVENCIÓN

Las formas de realización de la presente invención dan a conocer un método de autenticación, un sistema, un servidor y un cliente en función de un protocolo DS o de un protocolo DM para optimizar el proceso de autenticación que se realiza entre el cliente y el servidor y sobre la base del protocolo DS o DM.

El método de autenticación, basado en el protocolo DS o DM, en una forma de realización de la presente invención, comprende:

la recepción, por un cliente, de un mensaje Trigger desde un servidor que utiliza un número aleatorio de mensaje Trigger para generar el mensaje Trigger; la extracción del número aleatorio del mensaje Trigger;

después de determinar que el número aleatorio del mensaje Trigger es válido, la utilización del número aleatorio del mensaje Trigger para generar un Digest y la autenticación del mensaje Trigger generado utilizando el número aleatorio de mensaje Trigger y

una vez realizada satisfactoriamente la autenticación, el envío de una demanda de sesión al servidor indicada por el mensaje Trigger, en donde la demanda de sesión transmite un identificador ID de sesión.

El cliente dado a conocer en una forma de realización de la presente invención comprende:

una unidad de recepción, adaptada para recibir un mensaje Trigger que se genera por un servidor utilizando un número aleatorio de mensaje Trigger y en conformidad con un protocolo DS o DM y

una primera unidad de autenticación, adaptada para: extraer el número aleatorio del mensaje Trigger; después de la determinación de que el número aleatorio del mensaje Trigger es válido, utilizar el número aleatorio del mensaje Trigger para generar un Digest y realizar la autenticación del mensaje Trigger generado utilizando el número aleatorio del mensaje Trigger; después de la autenticación operativamente satisfactoria, enviar una demanda de sesión al servidor indicada por el mensaje Trigger.

La solución técnica anterior mejora efectivamente la seguridad del sistema.

Mediante la solución técnica anterior, el servidor y el cliente comparten un número aleatorio en el proceso de sesión en lugar del *s_nonce* y del *c_nonce* en la técnica anterior para poner en práctica la autenticación entre el cliente y el servidor, con lo que se libera efectivamente parte de la carga del sistema.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La Figura 1 es un diagrama de flujo de un método de autenticación en la técnica anterior;

La Figura 2 es un diagrama de flujo de utilización de un número aleatorio por defecto para la autenticación y actualización de *s_nonce* y *c_nonce* en la técnica anterior;

La Figura 3 es un diagrama de flujo de un método de autenticación en la forma de realización 1 de la presente invención;

La Figura 4 ilustra una estructura de un formato de mensaje después de que se añada un número aleatorio *nonce*;

La Figura 5 es un diagrama de flujo de un método de autenticación cuando el *s_nonce* es incorrecto en la forma de realización 1 de la presente invención;

La Figura 6 es un diagrama de flujo de un método de autenticación en la forma de realización 2 de la presente invención;

La Figura 7 es un diagrama de flujo de un método de autenticación en la forma de realización 4 de la presente invención;

La Figura 8 es un diagrama de flujo de un método de autenticación en la forma de realización 5 de la presente invención;

La Figura 9 es un diagrama de flujo de un método de autenticación en la forma de realización 6 de la presente invención;

La Figura 10 representa un formato de mensaje de respuesta de estado que transmite un nuevo *s_nonce* y en un método de autenticación en la forma de realización 7 de la presente invención,

La Figura 11 es un diagrama de flujo de un método de autenticación en la forma de realización 8 de la presente invención;

La Figura 12 es un diagrama de flujo de un método de autenticación en la forma de realización 9 de la presente invención;

La Figura 13 es un diagrama de flujo de un método de autenticación en la forma de realización 10 de la presente invención;

La Figura 14 representa una estructura de un sistema de autenticación en la forma de realización 1 de la presente invención;

La Figura 15 representa una estructura de un cliente en la forma de realización 1 de la presente invención;

La Figura 16 representa una estructura de un cliente en la forma de realización 2 de la presente invención y

La Figura 17 representa una estructura de un sistema de autenticación en la forma de realización 2 de la presente invención.

DESCRIPCIÓN DETALLADA DE LA INVENCIÓN

Las formas de realización de la presente invención dan a conocer un método de autenticación basado en un protocolo DS o protocolo DM, un sistema, un servidor y un cliente para optimizar el proceso de autenticación que se realiza entre el cliente y el servidor y sobre la base del protocolo DS o protocolo DM.

El mecanismo de seguridad aplicado a la autenticación del mensaje en una sesión aquí mencionada, es un mecanismo de seguridad de capa de aplicación.

En el método de autenticación en la forma de realización 1 de la presente invención, el servidor genera un número aleatorio para el mensaje Trigger, en donde el número aleatorio es diferente del *s_nonce* y del *c_nonce* y está disponible para el mensaje Trigger. El número aleatorio puede denominarse un número aleatorio de mensaje Trigger. El servidor utiliza este número aleatorio para generar información de autenticación y envía el nuevo número aleatorio y la información de autenticación, junto con el mensaje Trigger, al cliente. El cliente utiliza el nuevo número aleatorio para la autenticación del mensaje Trigger.

Según se ilustra en la Figura 3, el método de autenticación en la forma de realización 1 de la presente invención comprende las etapas siguientes:

Etapa 301: El servidor envía un mensaje Trigger al cliente. El mensaje transmite un número aleatorio de mensaje Trigger.

Antes de enviar el mensaje Trigger, el servidor genera un número aleatorio de mensaje Trigger y utiliza este número aleatorio para generar un Digest y luego, utiliza el Digest para generar un mensaje Trigger.

En esta forma de realización, existen tres tipos de método de utilizar el número aleatorio del mensaje Trigger.

(1) Cuando se genera el mensaje Trigger, el servidor utiliza su propio tiempo del sistema (Ts) como un número aleatorio de mensaje Trigger y añade el tiempo Ts en el mensaje Trigger. Por lo tanto, después de recibir el mensaje Trigger, el cliente puede determinar la validez del número aleatorio comparando el tiempo local (Tc) con el tiempo del sistema (Ts). Para el número aleatorio, su validez se suele denominar freshness (carácter reciente) del número aleatorio *nonce*. Un número aleatorio reciente es válido y un número aleatorio no reciente no es válido.

Después de recibir el mensaje Trigger, el cliente calcula la diferencia entre Ts y Tc, esto es, |Ts-Tc|. Si el |Ts-Tc| es menor que un umbral preestablecido “Diff”, el número aleatorio del mensaje Trigger es válido; si el |Ts-Tc| no es menor que un umbral preestablecido “Diff”, el número aleatorio del mensaje Trigger no es válido.

El valor Diff umbral se suele configurar en el cliente y puede ser un valor empírico determinado en función de las condiciones de la red. Puesto que la propia red móvil no es estable y tiende a generar retardos de transmisión del mensaje de disparo Trigger. Umbrales demasiado pequeños tienden a hacer que no sea válido el número aleatorio del mensaje Trigger; si el umbral es demasiado grande, en el caso de un servidor operativamente malicioso intercepte el mensaje Trigger y conserve el mensaje para el cliente de forma repetida, el cliente considera los mensajes como información válida y los gestiona en tanto que el valor |Ts-Tc| caiga dentro del margen del umbral. Umbrales mayores son más vulnerables a los ataques.

(2) Antes de generar un mensaje Trigger, el servidor generar un identificador ID de sesión para el mensaje Trigger, en primer lugar, en conformidad con determinadas reglas. Las reglas le hacen factible para deducir el ID de sesión anterior a partir del ID de sesión actual. El ID de sesión sirve como un número aleatorio de mensaje Trigger. El servidor utiliza el número aleatorio para generar un Digest y utiliza el Digest para generar un mensaje Trigger.

Después de recibir el mensaje Trigger, el cliente extrae el identificador ID de sesión de la sesión a iniciarse por el mensaje Trigger y utiliza este identificador ID de sesión, el ID del servidor, la contraseña del servidor y otros campos del mensaje Trigger para generar un Digest para la autenticación del mensaje. Después de la autenticación operativamente satisfactoria, el cliente envía una demanda de sesión para establecer la sesión correspondiente al ID de sesión. El servidor extrae el ID de sesión desde la demanda de sesión para identificar dicha sesión.

Además, después de que el cliente extraiga el ID de sesión de la sesión a iniciarse por el mensaje Trigger, el cliente puede inferir el carácter reciente del ID de sesión en conformidad con las reglas de codificación del ID de sesión o el cliente memoriza los identificadores IDs de sesión utilizados y compara el ID de sesión del mensaje Trigger con los identificadores IDs de sesión memorizados para determinar sui carácter reciente.

En este método, el ID de sesión puede sustituirse por un ID de mensaje Trigger (NotificationID). Este ID de mensaje Trigger asocia el resultado del procesamiento del mensaje Trigger reenviado por el cliente con el mensaje Trigger.

(3) Cuando se genera el mensaje Trigger, el servidor numera cada mensaje Trigger y utiliza el número como un número aleatorio de mensaje Trigger exclusivo. El servidor utiliza el número aleatorio para generar un Digest y utiliza el Digest para generar un mensaje Trigger.

El número puede estar en orden ascendente o en orden descendente. Después de recibir un mensaje Trigger, el cliente compara el número aleatorio transmitido en el mensaje con el número aleatorio memorizado. En el caso de que el número esté en orden ascendente, si el nuevo número aleatorio es mayor, el número aleatorio es válido; en caso contrario, no es válido. En el caso de que el número esté en orden descendente, si el nuevo número aleatorio es más pequeño, el número aleatorio es válido y en caso contrario, no es válido.

Después de determinar que el nuevo número aleatorio es válido y de la autenticación satisfactoria del servidor, el cliente memoriza el nuevo número aleatorio, que está disponible para la comparación con el siguiente número aleatorio del mensaje Trigger.

En este método, si un servidor, operativamente malicioso, intercepta el mensaje Trigger y ataca al cliente enviando el mensaje al cliente de forma repetida, puesto que el número aleatorio utilizado por este mensaje Trigger ha sido registrado, todos los mensajes operativamente maliciosos se determinan como no válidos con lo que se impiden ataques procedentes de servidores operativamente maliciosos.

Además, debido a la inestabilidad de las redes móviles, el mensaje enviado más tarde puede llegar primero al cliente y los mensajes Trigger enviados por el servidor para diferentes sesiones pueden llegar al cliente en orden cambiado y en consecuencia, el cliente determina los mensajes válidos como mensajes no válidos de forma errónea.

A modo de ejemplo, el servidor envía 3 mensajes Trigger para 3 sesiones diferentes de forma consecutiva. El número aleatorio del mensaje Trigger utilizado por los 3 mensajes Trigger es 30, 31 y 32 respectivamente. Sin embargo, debido a la inestabilidad de las redes móviles, el cliente recibe primero el mensaje Trigger cuyo número aleatorio es 32. Por lo tanto, el cliente determina este mensaje como válido y registra este número aleatorio. Cuando los otros dos mensajes Trigger llegan al cliente, el cliente compara su número aleatorio con el número aleatorio registrado. Puesto que su número aleatorio es más pequeño que el número aleatorio registrado, el cliente determina, de forma errónea, que los mensajes no son válidos.

Para dichos problemas, las formas de realización de la presente invención ofrecen tres soluciones:

Solución 1: El cliente memoriza todos los valores de números aleatorios de mensajes Trigger o el último número aleatorio de mensaje Trigger recibido y compara el número aleatorio de mensaje Trigger determinado como no válido con el número aleatorio memorizado. Si el número aleatorio es diferente del número aleatorio memorizado, el cliente determina al número aleatorio como válido y lo memoriza.

Cuando el espacio de almacenamiento está limitado, se establece un espacio de almacenamiento y se suprime el número aleatorio mínimo memorizado cuando la cantidad de valores de números aleatorios memorizados alcanza el límite superior.

Solución 2: En el caso de que los valores de números aleatorios de mensajes Trigger estén numerados en un orden ascendente, el cliente memoriza el número aleatorio máximo recibido y la totalidad o parte de los valores de números aleatorios que son más pequeños que el valor máximo actual y no han sido recibidos; el cliente compara el número aleatorio de mensaje Trigger determinado como no válido con los valores de números aleatorios memorizados y, si el número aleatorio es diferente de los valores de números aleatorios memorizados, determina el número aleatorio como

válido y lo memoriza. En el caso de que los valores de números aleatorios de mensajes Trigger estén numerados en un orden descendente, el cliente memoriza el número aleatorio mínimo recibido y la totalidad o parte de los valores de números aleatorios que son mayores que el valor mínimo actual y no han sido recibidos; el cliente compara el número aleatorio del mensaje Trigger determinado como no válido con los valores de números aleatorios memorizados y, si el número aleatorio es diferente de los valores de números aleatorios memorizados, determina el número aleatorio como válido y lo memoriza.

A modo de ejemplo, se supone que el valor inicial es 1, el modo de numeración es el orden ascendente y el cliente recibe secuencialmente estos valores de números aleatorios de mensajes Trigger: 1, 2, 4, 5 y 7. En este caso, el cliente registra el número aleatorio máximo “7” y los valores de números aleatorios que son más pequeños que 7 y no han sido recibidos, esto es, “3” y “6”. Cuando el cliente recibe un número aleatorio de mensaje Trigger “6”, el cliente compara “6” con el valor máximo “7”. Puesto que 6 es menor que 7, el número aleatorio no es válido. A continuación, el cliente compara “6” con “3” y “6” y encuentra el mismo valor y por lo tanto, el cliente determina que el número aleatorio del mensaje Trigger es válido y suprime el “6” registrado. En el caso de que el modo de numeración esté en orden descendente, el método de determinación es similar y por ello no se repite en esta descripción.

Solución 3: En el caso de que los valores de números aleatorios de mensajes Trigger estén numerados en orden ascendente, el cliente memoriza el número aleatorio máximo y considera todos los mensajes Trigger cuyo número aleatorio sea menor que el número aleatorio máximo memorizado como no válidos. En el caso de que los valores de números aleatorios de mensajes Trigger estén numerados en orden descendente, el cliente memoriza el número aleatorio mínimo y considera a todos los mensajes Trigger cuyo número aleatorio sea mayor que el número aleatorio mínimo memorizado como no válidos. Si el servidor no recibe ninguna respuesta desde el cliente en un periodo de tiempo, el servidor genera un nuevo número aleatorio en conformidad con las reglas de numeración y envía el mensaje Trigger que transmite el nuevo número aleatorio.

Lo anteriormente descrito se refiere a un método de utilización de un número aleatorio de mensaje Trigger en una forma de realización de la presente invención.

Si el tiempo del sistema o el número de mensaje Trigger se utiliza como el número aleatorio del mensaje Trigger, el número aleatorio del mensaje Trigger puede transmitirse en la cabecera del mensaje o en el cuerpo del mensaje del mensaje Trigger. Tomando, a modo de ejemplo, la cabecera del mensaje, según se representa en la Figura 4, el formato del mensaje con un número aleatorio añadido comprende: un Digest, una cabecera de mensaje Trigger (Trigger-hdr) y un cuerpo de mensaje Trigger (Trigger-body).

La cabecera Trigger-hdr comprende: la versión, el modo de interacción del usuario (ui-mode), el iniciador de sesión, un número aleatorio, un campo reservado (uso-futuro), identificador ID de sesión, longitud del identificador del servidor (length-identifier) y el identificador del servidor.

Además, las formas de realización del a presente invención dan a conocer dos métodos de utilización de un número aleatorio de mensaje Trigger para generar un Digest:

Método 1: Supóngase H = MD5 como función Hashing y b64 = Base64 como función de codificación. El Digest puede expresarse como:

Digest = H(B64(H(identificador-servidor:contraseña)):*nonce*:B64(H(Trigger)),

en donde el campo del identificador del servidor es un identificador del servidor, el campo de la contraseña es una contraseña de servidor, el campo *nonce* es un número aleatorio de mensaje Trigger (esto es, el tiempo del sistema (Ts) o el identificador ID de sesión o el número de mensaje Trigger anteriormente citado) y el campo de Trigger incluye la cabecera Trigger-hdr y el cuerpo Trigger-body del mensaje Trigger.

Después de recibir el mensaje Trigger y de determinar que el número aleatorio del mensaje Trigger transmitido en el mensaje Trigger es válido, el cliente busca el árbol de gestión del cliente para la contraseña correspondiente al servidor. El cliente utiliza la contraseña encontrada, el identificador del servidor en el mensaje Trigger, el número aleatorio y el Trigger para generar un Digest y compara el Digest generado con el Digest transmitido en el mensaje. Si el Digest es el mismo, ello significa que la autenticación es operativamente satisfactoria; en caso contrario, falla la autenticación.

Método 2: Supóngase H = MD5 como función Hashing y b64 = Base64 como función de codificación.

Puesto que el número aleatorio de mensaje Trigger se transmite en la cabecera del mensaje o en el cuerpo del mensaje, el número aleatorio se hace parte del campo Trigger-hrd y del campo Trigger-body en el mensaje Trigger. Por lo tanto, para calcular el Digest, solamente necesita utilizarse el campo Trigger-hdr y el campo Trigger-body. El Digest puede expresarse como:

Digest = H(B64(H(identificador-servidor:contraseña)):B64(H(Trigger))),

en donde el campo del identificador del servidor es un identificador del servidor, el campo de contraseña es una contraseña de servidor y el campo de Trigger incluye la cabecera Trigger-hdr y el cuerpo Trigger-body del mensaje Trigger.

Después de recibir el mensaje Trigger y de determinar que el número aleatorio del mensaje Trigger transmitido en el mensaje Trigger es válido, el cliente busca el árbol de gestión del cliente para la contraseña correspondiente al servidor. El cliente utiliza la contraseña encontrada, el identificador del servidor en el mensaje Trigger y el Trigger para generar un Digest y compara el Digest generado con el Digest transmitido en el mensaje. Si el Digest es el mismo, la autenticación es operativamente satisfactoria y en caso contrario, falla la autenticación.

Etapa 302: El cliente determina que la información es válida, realiza la autenticación de la información de forma satisfactoria y luego, envía una demanda de sesión al servidor.

Después de recibir el mensaje Trigger, el cliente determina si el número aleatorio del mensaje Trigger, transmitido en el mensaje Trigger, es válido. El método de determinación es según se describió anteriormente. Si se determina que el número aleatorio del mensaje Trigger es válido, el cliente busca el árbol de gestión del cliente para la contraseña correspondiente al servidor. El cliente utiliza la contraseña encontrada, el identificador del servidor en el mensaje Trigger y el Trigger para generar un Digest y realiza la autenticación del mensaje Trigger. El método de autenticación detallado se describe en la etapa 301. El método de autenticación del cliente varía con el método de generación del Digest.

Después de la realización satisfactoria de la autenticación, el cliente envía una demanda de sesión al servidor para iniciar una sesión.

La demanda de sesión transmite: un identificador de sesión SessionID e información de autenticación (Authenticate) que incluye el Digest generado utilizando el *c_nonce*.

En esta etapa, se establece una conexión de sesión entre el cliente y el servidor.

Etapa 303: El servidor reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación.

En función de la información de autenticación enviada por el cliente, el servidor realiza la autenticación del cliente y luego, reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación al cliente.

Más concretamente, la respuesta transmite: un resultado de la autenticación por el servidor del cliente, un identificador de sesión SessionID e información de autenticación (Authenticate) que incluye el Digest generado utilizando el *s_nonce*.

Etapa 304: El cliente reenvía un mensaje que transmite el resultado de la autenticación al servidor.

Según la información de autenticación enviada por el servidor, el cliente realiza la autenticación del servidor y luego, reenvía un mensaje que transmite el resultado de la autenticación al servidor.

Más concretamente, este mensaje transmite: un resultado de la autenticación por el cliente del servidor y otra información pertinente.

Además, en el caso de que los valores de los números aleatorios del mensaje Trigger estén numerados en orden ascendente, con el aumento de los mensajes Trigger, el valor del número aleatorio será cada vez mayor; en el caso de que los valores de números aleatorios de mensajes Trigger estén numerados en orden descendente, con el aumento de los mensajes Trigger, el valor del número aleatorio disminuye hasta 0. En tales casos, necesita ajustarse el número aleatorio, a modo de ejemplo, necesita ajustarse el punto de inicio del conteo. Las formas de realización de la presente invención dan a conocer varios métodos de ajuste del valor del número aleatorio cuando sea necesario:

Método 1: El servidor actualiza su contraseña de cuenta en el cliente a intervalos. El servidor y el cliente pueden efectuar la reposición del valor del número aleatorio automáticamente cuando el servidor actualiza su contraseña de cuenta en el cliente.

Método 2: Cuando necesita ajustarse el número aleatorio (a modo de ejemplo, cuando llega el tiempo preestablecido o el conteo alcanza el valor preestablecido), el servidor emite una orden para la reposición del número aleatorio. La orden puede ser una orden Alert, a modo de ejemplo:

```
<Alert>
<CmdID>1</CmdID>
<Datos>1227</Datos><!-sustituir conteo número aleatorio->
```

</Alert>

Después de ajustar el número aleatorio, el servidor emite una orden para cambiar su contraseña de cuenta en el cliente, con lo que se impide la intercepción del mensaje y los ataques por servidores operativamente maliciosos.

Método 3: Puesto que el servidor puede operar directamente con el árbol de gestión del cliente, el servidor puede añadir un nodo en su información de cuenta en el árbol de gestión de cliente y utilizar el nodo para memorizar los valores de números aleatorios recibidos y retenidos por el cliente. El nodo puede ser:

<X>/AppAuth/<X>/SNNAAuthCount

En adelante, cuando necesite ajustarse el número aleatorio (a modo de ejemplo, cuando llegue el tiempo preestablecido o el conteo alcance el valor preestablecido), el servidor emite una orden Replace (Sustituir) para el nodo. Una instancia operativa de la orden es como sigue:

<Replace>

<CmdID>4</CmdID>

<Item>

<Objetivo>

<LocURI>./DMAcc/servidorA/AppAuth/1/SNAAuthCount</LocURI>

</Objetivo>

<Datos>1</Datos>

</Item>

</Replace>

Después de ajustar el número aleatorio, el servidor emite una orden para cambiar su contraseña de cuenta en el cliente, con lo que se impide la intercepción de mensajes y los ataques por servidores operativamente maliciosos.

Método 4: Cuando necesite ajustarse el número aleatorio (a modo de ejemplo, cuando llegue el tiempo preestablecido o el conteo alcance el valor preestablecido), el cliente envía una demanda de Sustitución Replace al servidor. Después de que el cliente reciba una confirmación del servidor, ambas partes ajustan el número aleatorio. A la terminación del ajuste, el servidor actualiza su contraseña de cuenta en el cliente, con lo que se impide la intercepción de mensajes y los ataques por servidores operativamente maliciosos.

En el método de autenticación dado a conocer en la forma de realización 1 de la presente invención, un número aleatorio diferente del *s_nonce* y del *c_nonce* y disponible para el mensaje Trigger se proporciona en este momento. Una vez iniciada una nueva sesión, el servidor genera un número aleatorio para ser exclusivamente utilizado por el mensaje Trigger para iniciar una sesión. El cliente utiliza el número aleatorio para la autenticación del mensaje Trigger. Aún cuando el *s_nonce* memorizado por el servidor sea incorrecto, el cliente puede iniciar todavía una sesión. En este caso, si el *s_nonce* o el *c_nonce* es incorrecto, el *s_nonce* o el *c_nonce* puede actualizarse mediante una interacción para poner en práctica la autenticación.

Tomando, a modo de ejemplo, un *s_nonce* incorrecto, según se representa en la Figura 5, el método de autenticación dado a conocer en la forma de realización 1 de la presente invención, comprende las etapas siguientes:

Etapa 501: El servidor envía un mensaje Trigger al cliente. El mensaje transmite un número aleatorio de mensaje Trigger.

Antes del envío, el servidor genera un número aleatorio de mensaje Trigger y utiliza este número aleatorio para generar un Digest y luego, utiliza el Digest para generar un mensaje Trigger.

Etapa 502: El cliente determina que el número aleatorio del mensaje Trigger transmitido en el mensaje Trigger es válido, realiza la autenticación satisfactoria del mensaje y luego, envía una demanda de sesión al servidor.

Después de recibir el mensaje Trigger, el cliente determina si el número aleatorio del mensaje Trigger, transmitido en el mensaje Trigger, es válido. El método de determinación es según se describió anteriormente. Si se determina que el número aleatorio del mensaje Trigger es válido, el cliente busca el árbol de gestión del cliente para la contraseña correspondiente al servidor. El cliente utiliza la contraseña encontrada, el identificador del servidor en el mensaje Trigger

y el Trigger para generar un Digest y realiza la autenticación del mensaje Trigger. El método de autenticación detallado se describe en la etapa 301. El método de autenticación del cliente varía con el método de generación del Digest.

Una vez realizada satisfactoriamente la autenticación, el cliente envía una demanda de sesión al servidor para iniciar una sesión.

La demanda de sesión transmite: un identificador de sesión SessionID e información de autenticación (Authenticate) que incluye el Digest generado utilizando el *c_nonce*.

En esta etapa, se establece una conexión de sesión entre el cliente y el servidor.

Etapa 503: El servidor reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación.

En conformidad con la información de autenticación enviada por el cliente, el servidor realiza la autenticación del cliente y luego, reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación al cliente.

Más concretamente, la respuesta transmite: un resultado de la autenticación por el servidor del cliente, un identificador SessionID e información de autenticación (Authenticate) que incluye el Digest generado utilizando el *s_nonce*.

Etapa 504: El cliente utiliza el *s_nonce* memorizado para la autenticación del servidor, pero falla la autenticación.

Etapa 505: El cliente envía un mensaje Challenge y un mensaje de actualización de *s_nonce* al servidor.

Etapa 506: El servidor utiliza un nuevo *s_nonce* para generar información de autenticación y envía una demanda de autenticación al cliente de nuevo.

Después de recibir el mensaje de actualización, el servidor actualiza el *s_nonce* memorizado según se indica por el cliente, utiliza el *s_nonce* actualizado para generar una nueva demanda de autenticación y envía el resultado de la actualización y la nueva demanda de autenticación al servidor.

Etapa 507: El cliente reenvía un mensaje que transmite el resultado de la autenticación al servidor.

En conformidad con la demanda de autenticación que se envía por el servidor y se genera utilizando el *s_nonce* actualizado, el cliente realiza la autenticación del servidor y luego, reenvía un mensaje que transmite el resultado de la autenticación al servidor.

Más concretamente, este mensaje transmite: un resultado de la autenticación por el cliente del servidor y otra información pertinente.

En el método de autenticación dado a conocer en la forma de realización 1 de la presente invención, una vez que se inicia una nueva sesión, el servidor genera un número aleatorio exclusivamente disponible para el mensaje Trigger para iniciar una sesión. En el método de autenticación dado a conocer en la forma de realización 2 de la presente invención, el número aleatorio exclusivamente disponible para el mensaje Trigger, para iniciar una sesión, se genera solamente cuando el servidor considera el *s_nonce* como incorrecto.

Según se ilustra en la Figura 6, el método de autenticación dado a conocer en la forma de realización 2 de la presente invención comprende las etapas siguientes:

Etapa 601: El servidor envía un mensaje Trigger al cliente. El mensaje transmite un *s_nonce*.

Etapa 602 El servidor descubre que falla la autenticación.

Si el servidor no recibe ningún mensaje enviado desde el cliente dentro de un periodo específico, el servidor considera que ha fallado la autenticación.

Etapa 603: El servidor envía un nuevo mensaje Trigger al cliente. El mensaje transmite un número aleatorio de mensaje Trigger.

Antes del envío, el servidor genera un número aleatorio de mensaje Trigger y utiliza este número aleatorio para generar un Digest y luego, utiliza el Digest para generar un mensaje Trigger.

Etapa 604: El cliente determina que el número aleatorio del mensaje Trigger, transmitido en el mensaje Trigger es válido, realiza la autenticación satisfactoria del mensaje y luego, envía una demanda de sesión al servidor.

Después de recibir el mensaje Trigger, el cliente decide si utilizar el *s_nonce* o el número aleatorio del mensaje Trigger para la autenticación, determinando si el mensaje Trigger utiliza, o no, el número aleatorio de mensaje Trigger.

El método de determinación es: El cliente determina si el mensaje Trigger utiliza el número aleatorio del mensaje Trigger determinando si el mensaje Trigger incluye un campo de número aleatorio. Es decir, si el mensaje Trigger incluye un campo de número aleatorio, el mensaje Trigger utiliza el número aleatorio del mensaje Trigger. Como alternativa, el cliente determina si el mensaje Trigger utiliza el número aleatorio del mensaje Trigger determinando la información del campo de la versión en el mensaje Trigger. Es decir, puesto que la versión del mensaje revela si el mensaje utiliza, o no, el número aleatorio del mensaje Trigger.

Después de recibir el mensaje Trigger, el cliente determina si el número aleatorio del mensaje Trigger, transmitido en el mensaje Trigger, es, o no, válido. El método de determinación es según se describió anteriormente. Si se determina que el número aleatorio del mensaje Trigger es válido, el cliente busca el árbol de gestión del cliente para la contraseña correspondiente al servidor. El cliente utiliza la contraseña encontrada, el identificador del servidor en el mensaje Trigger y el Trigger para generar un Digest y realiza la autenticación del mensaje Trigger. El método de autenticación detallado se describe en la etapa 301. El método de autenticación del cliente varía con el método de generación del Digest.

Después de la autenticación operativamente satisfactoria, el cliente envía una demanda de sesión al servidor para iniciar una sesión.

La demanda de sesión transmite: un identificador SessionID e información de autenticación (Authenticate) que incluye el Digest generado utilizando el *c_nonce*.

En esta etapa, se establece una conexión de sesión entre el cliente y el servidor.

Etapa 605: El servidor reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación.

Esta etapa es similar a la etapa 503 y por ello no se detalla en esta descripción.

Etapa 606: El cliente utiliza el *s_nonce* memorizado para la autenticación del servidor, pero falla la autenticación.

Etapa 607: El cliente envía un mensaje que transmite una demanda Challenge y un nuevo *s_nonce* al servidor.

Etapa 608: El servidor reenvía un resultado de actualización y una nueva demanda de autenticación al cliente.

Esta etapa es similar a la etapa 506 y por ello no se detalla en esta descripción.

Etapa 609: El cliente reenvía un mensaje que transmite el resultado de la autenticación al servidor.

Esta etapa es similar a la etapa 507 y por ello no se detalla en esta descripción.

En el método de autenticación, dado a conocer en la forma de realización 2 de la presente invención, cuando el servidor considera que el *s_nonce* es incorrecto, el servidor genera un número aleatorio exclusivamente disponible para el mensaje Trigger para iniciar una sesión. El cliente procesa el mensaje Trigger en conformidad con el método de autenticación dado a conocer en la forma de realización 1 de la presente invención. En el método de autenticación dado a conocer en la forma de realización 3 de la presente invención, el cliente puede decidir si actualizar, o no, el *s_nonce* determinando si el mensaje Trigger utiliza el número aleatorio exclusivamente disponible para el mensaje Trigger. Si el mensaje Trigger utiliza el número aleatorio exclusivamente disponible para el mensaje Trigger, el cliente actualiza el *s_nonce* directamente, de modo que el servidor pueda utilizar directamente el *s_nonce* actualizado para la autenticación en la sesión.

En los métodos de autenticación dados a conocer en la primera y segunda formas de realización de la presente invención, cuando falla la autenticación, no se requiere ningún número aleatorio por defecto para poner en práctica la autenticación, con lo que se mejora la seguridad del sistema.

En el método de autenticación dado a conocer en la forma de realización 3 de la presente invención, los valores de *s_nonce* y el *c_nonce* se actualizan sobre la base de la técnica anterior y la contraseña del servidor y la contraseña del cliente se actualizan en consecuencia. La contraseña del servidor actualizada da lugar a un Digest del servidor diferente generado utilizando el número aleatorio por defecto e impide que el mensaje sea sensible a los ataques al cliente. La contraseña del cliente actualizada da lugar a un diferente Digest del cliente generado utilizando el número aleatorio por defecto, impide la reproducción de los ataques al servidor y mejora la seguridad del sistema.

En el método de autenticación dado a conocer en la forma de realización 4 de la presente invención, se refuerza la correlación entre etapas y la etapa anterior se utiliza como una base de autenticación de la siguiente etapa, con miras a

mejorar la seguridad del sistema. Según se ilustra en la Figura 7, el método de autenticación comprende las etapas siguientes:

Etapa 701: El cliente recibe un mensaje Trigger para iniciar una sesión y realiza la autenticación del mensaje.

Etapa 702: El cliente realiza la autenticación del mensaje Trigger de forma operativamente insatisfactoria y utiliza el número aleatorio por defecto para realizar una nueva autenticación.

Etapa 703: El cliente envía una demanda de sesión a un servidor, en donde la demanda de sesión se genera utilizando un número aleatorio por defecto.

Si la autenticación realizada utilizando el número aleatorio por defecto es satisfactoria, el cliente envía una demanda de sesión al servidor que se indica por el mensaje Trigger. Si la sesión utiliza el mecanismo de seguridad de capa de aplicación, el mensaje transmite la información de autenticación generada utilizando el número aleatorio por defecto y el proceso prosigue con la etapa 704. Si la autenticación realizada utilizando el número aleatorio por defecto falla operativamente, el cliente ignora el mensaje Trigger sin iniciar ninguna sesión y se finaliza el proceso.

Etapa 704: El servidor realiza la autenticación de la demanda de sesión enviada por el cliente.

El servidor realiza la autenticación de la demanda de sesión en dos métodos:

Método 1: El servidor utiliza un *c_nonce* para generar información de autenticación para la autenticación y, si la autenticación es satisfactoria, realiza el proceso normal según la técnica anterior. Si falla la autenticación, el servidor utiliza el número aleatorio por defecto para generar información de autenticación y realiza de nuevo la autenticación. Si la autenticación es satisfactoria, el servidor utiliza el número aleatorio por defecto para generar una demanda de autenticación del servidor y el proceso prosigue con la etapa 705.

Método 2: Si el servidor determina que la sesión utiliza el mecanismo de seguridad de la capa de aplicación, el servidor ha enviado el mensaje Trigger generado utilizando el número aleatorio por defecto al cliente y el mensaje Trigger es un mensaje Trigger para iniciar la demanda de sesión, el servidor utiliza el número aleatorio por defecto para realizar la autenticación de la demanda de sesión. Después de la autenticación operativamente satisfactoria, el proceso prosigue con la etapa 705.

El método de determinación de si una demanda de sesión se inicia, o no, por un mensaje Trigger es: Cada demanda de sesión tiene un identificador ID de sesión único. El identificador ID de sesión, transmitido en la demanda de sesión, se compara con el identificador ID de sesión transmitido en el mensaje Trigger. Si el identificador ID de sesión es el mismo, ello indica que la sesión se inicia por el mensaje Trigger.

La etapa de determinación de si el servidor ha enviado, o no, el mensaje Trigger generado utilizando el número aleatorio por defecto al cliente y si el mensaje Trigger es un mensaje Trigger para iniciar la demanda de sesión que puede producirse después de la demanda de sesión, es objeto de autenticación satisfactoria utilizando el número aleatorio por defecto. Después de la determinación satisfactoria, el proceso prosigue con la etapa 705.

La finalidad de esta etapa es: Si el servidor no ha enviado el mensaje Trigger que se genera utilizando el número aleatorio por defecto y diseñado para iniciar la sesión al cliente, pero recibe la demanda de sesión que se envía por el cliente y que se genera utilizando el número aleatorio por defecto, ello indica que el mensaje no es normal ni seguro y probablemente, es un mensaje fraudulento enviado por una tercera parte operativamente maliciosa y se puede rechazar. Por lo tanto, esta etapa mejora la seguridad del sistema.

Etapa 705: El servidor reenvía un mensaje de respuesta al cliente.

El servidor reenvía una respuesta que transmite el resultado de la autenticación, la demanda de autenticación y la orden para actualizar *c_nonce* al cliente.

Etapa 706: El cliente realiza la autenticación de respuesta enviada por el servidor.

Si la sesión utiliza el mecanismo de seguridad de capa de aplicación, el cliente utiliza un número aleatorio por defecto para la autenticación del servidor. El método de autenticación es: El cliente utiliza un *s_nonce* para generar información de autenticación para realizar la autenticación y, si la autenticación es satisfactoria, realiza el proceso normal según la técnica anterior. Si falla la autenticación, el cliente utiliza el número aleatorio por defecto para generar información de autenticación y realiza de nuevo la autenticación. Si la autenticación es satisfactoria, el cliente actualiza el *c_nonce* y el proceso prosigue con la etapa 707.

Una alternativa a esta etapa es: Si la sesión utiliza el mecanismo de seguridad de la capa de aplicación y el cliente ha enviado la demanda de sesión generada utilizando el número aleatorio por defecto al servidor, el cliente utiliza el número

aleatorio por defecto para la autenticación de la respuesta enviada por el servidor. Después de la autenticación operativamente satisfactoria, el cliente actualiza el *c_nonce* y el proceso prosigue con la etapa 707.

La etapa de determinación de si el cliente ha enviado, o no, la demanda de sesión generada utilizando el número aleatorio por defecto al servidor puede ocurrir después de que se realice la autenticación utilizando el número aleatorio por defecto. Después de la autenticación operativamente satisfactoria, se realiza la determinación. Después de la determinación satisfactoria, el proceso prosigue con la etapa 707.

Etapa 707: El cliente reenvía una respuesta al servidor.

El cliente reenvía una respuesta que transmite el resultado de la autenticación, el resultado de la actualización de *c_nonce* y la orden para actualizar *s_nonce* al servidor.

Etapa 708: El servidor reenvía un resultado de actualización de *s_nonce* al cliente.

Una vez terminada la etapa anterior, para impedir nuevos ataques, puede actualizarse la contraseña del servidor o, al mismo tiempo la contraseña del servidor y la contraseña del cliente se actualizan.

En la etapa anterior, el identificador ID de sesión puede servir como un número aleatorio o el identificador ID del mensaje Trigger puede servir como un número aleatorio en lugar del número aleatorio por defecto, con lo que se evita un número aleatorio público invariable y se consigue una mayor seguridad.

Los métodos de autenticación dados a conocer en la tercera y cuarta formas de realización de la presente invención, mejoran efectivamente la seguridad del sistema.

En la técnica anterior, cuando el *s_nonce* es incorrecto y necesita actualizarse, la orden necesita intercambiarse, por cuatro veces, para realizar la actualización, según se indica en las etapas 203-204 en la Figura 2. Puesto que el número aleatorio por defecto necesita utilizarse antes de que se actualice, el riesgo es alto. Con el mensaje intercambiándose en la red móvil numerosas veces, la carga de la red es más elevada.

Una solución técnica para añadir un nuevo *s_nonce* a la demanda de sesión enviada por el cliente se da a conocer en las formas de realización del método de autenticación aquí descrito. De este modo, el servidor puede actualizar directamente el *s_nonce* y utilizar el nuevo *s_nonce* para la autenticación, con lo que se reduce la frecuencia de interacción de señalización y la frecuencia de utilizar el número aleatorio por defecto, con lo que se mejora la seguridad del sistema y se reduce la carga de la red.

Según se representa en la Figura 8, un método de autenticación dado a conocer en la forma de realización 5 de la presente invención, comprende las etapas siguientes:

Etapa 801: El cliente conoce que necesita actualizarse un *s_nonce*.

El cliente determina que el *s_nonce* ha caducado o encuentra que el *s_nonce* memorizado en el servidor es diferente del memorizado en el cliente y por lo tanto, conoce que el *s_nonce* necesita actualizarse.

El cliente descubre una incoherencia entre el *s_nonce* memorizado en el servidor y el memorizado en el cliente en la forma siguiente:

Después de recibir el mensaje Trigger, el cliente utiliza el *s_nonce* memorizado para realizar la autenticación del mensaje Trigger. Si la autenticación falla por cualquier motivo, el cliente utiliza el número aleatorio por defecto o utiliza el identificador ID de sesión o el ID de mensaje Trigger como un número aleatorio para generar un Digest y se realiza la nueva autenticación del mensaje Trigger.

Si la autenticación es satisfactoria, ello indica que el *s_nonce* anteriormente utilizado por el servidor es incorrecto y el *s_nonce* memorizado en el servidor es diferente del memorizado en el cliente.

Etapa 802: El cliente envía una demanda de sesión que transmite información de actualización al servidor.

Después de conocer que el *s_nonce* necesita actualizarse, el cliente genera un nuevo *s_nonce*, añade el *s_nonce* a la demanda de sesión y envía la demanda de sesión al servidor, demandando al servidor que inicie una sesión y actualice el *s_nonce*.

La demanda de sesión transmite: un identificador SessionID, un *s_nonce* recientemente generado e información de autenticación (Authenticate) que incluye el Digest generado utilizando el *c_nonce*. En esta demanda de sesión, el número aleatorio por defecto o el identificador ID de sesión o el identificador ID de mensaje Trigger pueden utilizarse como un número aleatorio para generar el Digest.

El *s_nonce* recientemente generado puede transmitirse en una cabecera SyncHdr o cuerpo SyncBody de la demanda de sesión (SyncML).

El método de transmisión se describe a continuación, suponiendo que el *s_nonce* recientemente generado se transmite en un SyncHdr.

Para la transmisión de *s_nonce*, el SyncHdr se modifica como sigue:

SyncHdr (VerDTD, VerProto, SessionID, MsgID, Target, Source, RespURI?, NoResp?, Cred?, Chal?, Meta?)>

El mensaje SyncML que transmite el *s_nonce* es:

```
<SyncML xmlns ='SYNCML:SYNCML1.2'>
<SyncHdr>
…
<Chal>
<Meta>
<NextNonce xmlns= 'syncml:metinf'> LG3iZQhhdmKNHg==</siguiente número aleatorio>
</Meta>
</Chal>
</SyncHdr>
<SyncBody>
…
</SyncBody>
</SyncML>
```

Etapa 803: El servidor reenvía una respuesta que transmite el resultado de la autenticación, el resultado de la actualización y la demanda de autenticación.

Después de recibir la demanda de sesión, el servidor utiliza el *c_nonce* para la autenticación del cliente y utiliza el *s_nonce* actualizado, transmitido en la demanda de sesión, para actualizar el *s_nonce* memorizado. Después de la autenticación satisfactoria y de que se concluya la actualización, el servidor utiliza el *s_nonce* actualizado para generar una demanda de autenticación y reenvía una respuesta que transmite el resultado de la autenticación, la orden de actualización y la demanda de autenticación al cliente. En una forma de realización preferida, el servidor utiliza un *c_nonce* para la autenticación de la demanda de sesión. Después de la autenticación satisfactoria, el servidor actualiza el *s_nonce*, con lo que se mantiene la sincronización entre el *s_nonce* memorizado en el servidor y el memorizado en el cliente.

Más concretamente, la respuesta transmite: un resultado de la autenticación por el servidor del cliente, un resultado de actualización de *s_nonce* e información de autenticación (Authenticate) que incluye el Digest generado utilizando el *s_nonce* actualizado.

Etapa 804: El cliente reenvía un mensaje que transmite el resultado de la autenticación al servidor.

El cliente utiliza el *s_nonce* actualizado para realizar la autenticación del servidor. Después de la autenticación operativamente satisfactoria, el cliente reenvía un resultado de autenticación al servidor.

Además, el método de autenticación dado a conocer en la forma de realización 5 de la presente invención, puede aplicarse al método de autenticación dado a conocer en la forma de realización 2 de la presente invención para reducir la frecuencia de interacción de señalización.

Según se representa en la Figura 9, el método de autenticación dado a conocer en la forma de realización 6 de la presente invención comprende las etapas siguientes:

Etapa 901: El servidor envía un mensaje Trigger al cliente. El mensaje transmite un *s_nonce*.

Etapa 902: El servidor descubre que falla la autenticación.

A modo de ejemplo, si el servidor no recibe ninguna demanda de sesión desde el cliente dentro de un periodo de tiempo específico, el servidor considera que ha fallado la autenticación.

Etapa 903: El servidor envía un mensaje Trigger al cliente. El mensaje transmite un número aleatorio de mensaje Trigger.

Antes del envío, el servidor genera un número aleatorio de mensaje Trigger y utiliza este número aleatorio para generar un Digest y luego, utiliza el Digest para generar un mensaje Trigger.

Etapa 904: El cliente descubre que necesita actualizarse un *s_nonce*.

Después de recibir el mensaje Trigger, el cliente determina si el mensaje Trigger utiliza el número aleatorio exclusivamente disponible para el mensaje Trigger y decide si necesita actualizarse el *s_nonce*. Como resultado, el cliente descubre que el mensaje Trigger utiliza el número aleatorio exclusivamente disponible para el mensaje Trigger y necesita actualizarse el *s_nonce*.

El método de determinación de si el mensaje Trigger utiliza, o no, el número aleatorio exclusivamente disponible para el mensaje Trigger es: El cliente determina si el mensaje Trigger utiliza el número aleatorio de mensaje Trigger determinando si el mensaje Trigger incluye, o no, un campo de número aleatorio. Es decir, si el mensaje Trigger incluye un campo de número aleatorio, el mensaje Trigger utiliza el número aleatorio del mensaje Trigger. Como alternativa, el cliente determina si el mensaje Trigger utiliza el número aleatorio de mensaje Trigger determinando la información del campo de la versión en el mensaje Trigger. Es decir, puesto que el campo de la versión revela si el mensaje utiliza, o no, el número aleatorio de mensaje Trigger.

Si el cliente descubre que el mensaje Trigger no utiliza el número aleatorio del mensaje Trigger, lo que indica que no necesita actualizarse el *s_nonce* y se realiza el proceso ordinario.

Etapa 905: El cliente envía una demanda de sesión que transmite información de actualización al servidor.

Después de recibir el mensaje Trigger y de determinar que el mensaje Trigger utiliza el número aleatorio de mensaje Trigger exclusivamente disponible para dicho mensaje Trigger, el cliente determina si el número aleatorio del mensaje Trigger, transmitido en el mensaje Trigger, es válido. El método de determinación se describió anteriormente. Si se determina que el número aleatorio de mensaje Trigger es válido, el cliente busca el árbol de gestión del cliente para la contraseña correspondiente al servidor. El cliente utiliza la contraseña encontrada, el identificador del servidor y el Trigger para generar un Digest y realiza la autenticación del mensaje Trigger. El método de autenticación detallado se describe en la etapa 301. El método de autenticación del cliente varía con el método de generación del Digest.

Después de la autenticación satisfactoria del mensaje Trigger, el cliente genera un nuevo *s_nonce*, añade el *s_nonce* a una demanda de sesión y envía la demanda de sesión que transmite información de actualización al servidor, demandando al servidor que inicie una sesión y actualice el *s_nonce*.

La demanda de sesión transmite: un identificador SessionID, un *s_nonce* actualizado e información de autenticación (Authenticate) que incluye el Digest generado utilizando el *c_nonce*.

El *s_nonce* recientemente generado puede transmitirse en el SyncHdr o SyncBoby de la demanda de sesión.

Etapa 906: El servidor reenvía una respuesta que transmite el resultado de la autenticación, el resultado de la actualización y la demanda de autenticación.

Después de recibir la demanda de sesión, el servidor utiliza el *c_nonce* para la autenticación del cliente y utiliza el *s_nonce* actualizado transmitido en la demanda de sesión, para actualizar el *s_nonce* memorizado. Después de la autenticación satisfactoria y de concluirse la actualización, el servidor utiliza el *s_nonce* actualizado para generar una demanda de autenticación y reenvía una respuesta que transmite el resultado de autenticación, el resultado de la actualización y la demanda de autenticación al cliente.

Más concretamente, la respuesta transmite: un resultado de la autenticación por el servidor para el cliente, un resultado de actualización de *s_nonce* e información de actualización (Authenticate) que incluye el Digest generado utilizando el *s_nonce* actualizado.

Etapa 907: El cliente reenvía un mensaje que transmite el resultado de la autenticación al servidor.

El cliente utiliza el *s_nonce* actualizado para realizar la autenticación del servidor. Después de la autenticación satisfactoria, el cliente reenvía un resultado de la autenticación al servidor.

Más concretamente, este mensaje transmite: un resultado de la autenticación del servidor y otra información pertinente.

En la técnica anterior, a veces, el cliente decide no iniciar la sesión después de la autenticación satisfactoria del servidor. En este caso, si el cliente descubre que el *s_nonce* caduca o es incorrecto y necesita actualizarse, resulta imposible actualizar el *s_nonce* o mantener efectivamente el *s_nonce*.

Por lo tanto, el método de autenticación en la forma de realización 7 de la presente invención da a conocer la solución correspondiente.

En el método de autenticación dado a conocer en la forma de realización 7 de la presente invención, después de que el cliente realice la autenticación del servidor satisfactoriamente y decida no iniciar una sesión, el cliente envía una respuesta de estado al servidor. Si el cliente determina que el *s_nonce* caduca o es incompatible con el *s_nonce* memorizado en el servidor, el cliente genera un nuevo *s_nonce* y añade el nuevo *s_nonce* en la respuesta del estado. El cliente utiliza el *c_nonce*, el nombre de usuario del cliente y su contraseña y el cuerpo del mensaje de respuesta para calcular el Digest. Por lo tanto, después de recibir la respuesta de estado, el servidor puede realizar la autenticación de la información en función del Digest calculado utilizando el *c_nonce*, el nombre de usuario del cliente y su contraseña y el cuerpo del mensaje de respuesta. Después de la autenticación satisfactoria, el servidor actualiza el *s_nonce* memorizado en función del nuevo *s_nonce* transmitido en la respuesta de estado.

Según se ilustra en la Figura 10, la respuesta de estado con el nuevo *s_nonce* comprende: un Digest, una cabecera de notificación (notification-hdr) y un cuerpo de notificación (notification-body).

La cabecera notification-hdr comprende: la versión, el código de estado operativo, el ID de notificación, el nuevo número aleatorio (Next-*nonce*), reservado (uso futuro>), ID de sesión (la longitud de SessionID del ID de autenticación (lenght-authname) y el ID de autenticación (authname).

El Next*Nonce* es el nuevo *s_nonce*.

En la técnica anterior, después de que el *s_nonce* sea incorrecto, el *s_nonce* y el *c_nonce* nunca se utilizarán de nuevo y el cliente y el servidor utilizan el número aleatorio por defecto para generar información de autenticación. En consecuencia, el servidor operativamente malicioso puede atacar al servidor o al cliente interceptando cualquier mensaje.

El *s_nonce* es diferente del *c_nonce* y se generan por el servidor y el cliente respectivamente y están disponibles en la parte opuesta. Por lo tanto, cuando uno de ellos es erróneo, el otro no resulta afectado. En los métodos de autenticación dados a conocer en la octava y novena forma de realización de la presente invención, se da a conocer una solución para actualizar el *s_nonce* por separado cuando el *s_nonce* es erróneo.

El *s_nonce* es erróneo si el cliente determina que ha caducado el *s_nonce* o descubre que el *s_nonce* memorizado en el servidor es erróneo, a modo de ejemplo, el cliente determina que el *s_nonce* memorizado en el servidor es incompatible o asíncrono con el *s_nonce* memorizado en el cliente.

El método de determinación de la coherencia y sincronización entre el *s_nonce* memorizado en el servidor y el *s_nonce* memorizado en el cliente puede ser: El servidor no recibe ninguna respuesta del cliente dentro de un periodo de tiempo específico después de utilizar el *s_nonce* para enviar un mensaje Trigger o el cliente descubre que el mensaje Trigger enviado por el servidor transmite un Digest generado utilizando el número aleatorio por defecto o el cliente descubre que no se utiliza ningún número aleatorio en el mensaje Trigger enviado por el servidor.

Después de descubrir el error del *s_nonce* y la autenticación satisfactoria del servidor, el cliente inicia una demanda de sesión. En la demanda de sesión, la información de autenticación para la autenticación del cliente se genera utilizando el *c_nonce* o el modo de autenticación básico (esto es, nombre de usuario más contraseña) se aplica y el *s_nonce* se actualiza de nuevo.

Dos métodos de autenticación del *s_nonce* se dan a conocer en la forma de realización 7 y la forma de realización 8 del método de autenticación, respectivamente.

En la forma de realización 8, el servidor utiliza un número aleatorio por defecto para generar un mensaje Trigger. Según se representa en la Figura 11, el método de autenticación comprende las etapas siguientes:

Etapa 1101: El servidor envía un mensaje Trigger al cliente para iniciar una sesión.

Después de determinar que el *s_nonce* anterior es incorrecto, el servidor utiliza el número aleatorio por defecto para generar un mensaje Trigger y envía el mensaje al cliente. El mensaje Trigger transmite: el Digest generado utilizando el número aleatorio por defecto, la información TriggerInfo y otra información pertinente.

Etapa 1102: El cliente realiza la autenticación insatisfactoria del mensaje Trigger y utiliza el número aleatorio por defecto para realizar una nueva autenticación.

Después de recibir el mensaje Trigger, el cliente utiliza el *s_nonce* memorizado para generar un Digest y para la autenticación del mensaje Trigger. Si falla la autenticación por algún motivo, el cliente utiliza el nonce por defecto para generar un Digest y volver a realizar la autenticación del mensaje Trigger.

Si la autenticación es operativamente satisfactoria, ello indica que el *s_nonce* anteriormente utilizado por el servidor es incorrecto y el *s_nonce* memorizado en el servidor es diferente del memorizado en el cliente.

Etapa 1103: El cliente envía una demanda de sesión que transmite información de actualización al servidor.

Después de la autenticación satisfactoria del mensaje Trigger utilizando el número aleatorio por defecto, el cliente genera un nuevo *s_nonce*, añade el *s_nonce* a una demanda de sesión y envía la demanda de sesión que transmite información de actualización al servidor, demandando al servidor para que inicie una sesión y actualice el *s_nonce*.

La demanda de sesión transmite: un identificador SessionID, un *s_nonce* actualizado e información de autenticación (Authenticate) que incluye el Digest generado utilizando el *c_nonce*.

En la etapa anterior, el ID de sesión puede servir como un número aleatorio o el ID del mensaje Trigger puede servir como un número aleatorio en lugar del número aleatorio por defecto, con lo que se evita un número aleatorio público invariable y se consigue una mayor seguridad.

El método de añadir el *s_nonce* actualizado a la demanda de sesión es esencialmente el mismo que el descrito en la forma de realización 3 y la forma de realización 4 y por ello, no se repite con detalle en esta descripción.

Etapa 1104: El servidor reenvía una respuesta que transmite el resultado de la autenticación, el resultado de la actualización y la demanda de autenticación.

Después de recibir la demanda de sesión, el servidor utiliza el *c_nonce* para la autenticación del cliente y utiliza el *s_nonce* actualizado, transmitido en la demanda de sesión, para actualizar el *s_nonce* memorizado. Después de la autenticación satisfactoria y de concluirse la actualización, el servidor utiliza el *s_nonce* actualizado para generar una demanda de autenticación y reenvía una respuesta que transmite el resultado de la autenticación, el resultado de la actualización y la demanda de autenticación al cliente.

Más concretamente, la respuesta transmite: un resultado de la autenticación del servidor para el cliente, un resultado de actualización de *s_nonce* y la información de autenticación (Authenticate) que incluye el Digest generado utilizando el *s_nonce* actualizado.

Etapa 1105: El cliente reenvía un mensaje que transmite el resultado de la autenticación al servidor.

El cliente el *s_nonce* actualizado para la autenticación del servidor. Después de la autenticación satisfactoria, el cliente reenvía un resultado de la autenticación al servidor.

En la forma de realización 9, el servidor utiliza un número aleatorio por defecto para generar un mensaje Trigger, pero la demanda de sesión no transmite ninguna información de actualización. Según se ilustra en la Figura 12, el método de autenticación comprende las etapas siguientes:

Etapa 1201: El servidor envía un mensaje Trigger al cliente para iniciar una sesión.

Después de la determinación de que el *s_nonce* anterior es incorrecto, el servidor utiliza el número aleatorio por defecto para generar un mensaje Trigger y envía el mensaje al cliente. El mensaje Trigger transmite: el Digest generado utilizando el número aleatorio por defecto, la información TriggerInfo y otra información pertinente.

Etapa 1202: El cliente realiza la autenticación insatisfactoria del mensaje Trigger y utiliza el número aleatorio por defecto para una nueva autenticación.

Después de recibir el mensaje Trigger, el cliente utiliza el *s_nonce* memorizado para generar un Digest y para la autenticación del mensaje Trigger. Si falla la autenticación por cualquier motivo, el cliente utiliza el número aleatorio por defecto para generar un Digest y para una nueva autenticación del mensaje Trigger.

Si la autenticación es satisfactoria, ello indica que el *s_nonce* anteriormente utilizado por el servidor es incorrecto y el *s_nonce* memorizado en el servidor es diferente del memorizado en el cliente.

Etapa 1203: El cliente realiza la autenticación satisfactoria de la información y luego, envía una demanda de sesión al servidor.

Después de la autenticación satisfactoria, el cliente envía una demanda de sesión al servidor para iniciar una sesión.

La demanda de sesión transmite: un identificador SessionID y la información de autenticación (Authenticate) que incluye el Digest generado utilizando el *c_nonce*.

En esta etapa, se establece una conexión de sesión entre el cliente y el servidor.

Etapa 1204: El servidor reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación.

En conformidad con la información de autenticación enviada por el cliente, el servidor realiza la autenticación del cliente. Después de la autenticación satisfactoria, el cliente utiliza un número aleatorio por defecto para generar información de autenticación (Authenticate) y luego, reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación al cliente.

Más concretamente, la respuesta transmite: un resultado de la autenticación del servidor para el cliente, un identificador SessionID y la información de autenticación (Authenticate) que incluye el Digest generado utilizando el número aleatorio por defecto.

Etapa 1205: El cliente reenvía una orden de actualización y un resultado de la autenticación al servidor.

El cliente realiza la autenticación del servidor utilizando el número aleatorio por defecto. Después de la autenticación satisfactoria, el cliente genera un nuevo *s_nonce* y envía una orden para actualizar *s_nonce* y un resultado de la autenticación del servidor al servidor.

Etapa 1206: El servidor reenvía un resultado de actualización al cliente.

Después de recibir el mensaje de actualización, el servidor actualiza el *s_nonce* memorizado según se indica por el cliente y reenvía un resultado de la actualización al cliente.

En la etapa anterior, el ID de sesión puede servir como un número aleatorio o el ID del mensaje Trigger puede servir como un número aleatorio en lugar del número aleatorio por defecto, con lo que se evita un número aleatorio público invariable y se consigue una mayor seguridad.

En este caso, la contraseña del servidor puede actualizarse para mejorar la fiabilidad del sistema.

En otra forma de realización, el servidor utiliza un número aleatorio por defecto para generar un mensaje Trigger, pero la demanda de sesión no transmite ningún número aleatorio por defecto. El método de autenticación incluye las etapas siguientes:

Después de determinar que el *s_nonce* anterior es incorrecto, el servidor utiliza el número aleatorio por defecto o utiliza el identificador ID de sesión o el identificador ID de mensaje Trigger como un número aleatorio para generar un mensaje Trigger y envía el mensaje al cliente. El mensaje Trigger transmite: el Digest generado utilizando el número aleatorio por defecto o el ID de sesión o el dispositivo objetivo del mensaje Trigger, la información TriggerInfo y otra información pertinente.

Después de recibir el mensaje Trigger, el cliente utiliza el *s_nonce* memorizado para generar un Digest y la autenticación del mensaje Trigger. Si falla la autenticación por cualquier motivo, el cliente utiliza el número aleatorio por defecto o utiliza el ID de sesión o el ID de mensaje Trigger como un número aleatorio para generar un Digest y para una nueva autenticación del mensaje Trigger. Si la autenticación es satisfactoria, el cliente envía una demanda de sesión al servidor para iniciar una sesión. La demanda de sesión transmite: el identificador SessionID y la información de autenticación (Authenticate) que incluye el Digest generado utilizando el *c_nonce*.

El servidor utiliza el *c_nonce* para la autenticación de la demanda de sesión. Si falla la autenticación, el servidor envía un mensaje Challenge para actualizar el *c_nonce* y requerir la nueva autenticación. Después de la autenticación satisfactoria, el servidor envía una demanda de autenticación generada utilizando el *s_nonce* y el cliente utiliza el *s_nonce* para la autenticación. Si falla la autenticación, el cliente envía un mensaje Challenge para actualizar el *s_nonce* y requerir una nueva autenticación. Después de la autenticación satisfactoria, el cliente reenvía un resultado.

Además, en la etapa anterior, el cliente puede añadir el *s_nonce* actualizado a la demanda de sesión y enviar la demanda al servidor. La demanda de autenticación enviada por el servidor, utiliza el nuevo *s_nonce*.

Según se describió anteriormente, cuando el *s_nonce* es erróneo, solamente se actualiza *s_nonce* y no se actualiza el *c_nonce.* Cuando el sistema gestiona el error de *s_nonce*, aún cuando el sistema utilice un número aleatorio por defecto o utilice el ID de sesión o el ID de mensaje Trigger como un número aleatorio para la autenticación, puesto que el *c_nonce* no necesita actualizarse, el cliente puede utilizar el *c_nonce* para generar una demanda de sesión, con lo que se reduce la frecuencia de utilización del número aleatorio por defecto o la utilización del ID de sesión o del ID de mensaje Trigger como un número aleatorio y se mejora la seguridad del sistema.

En la técnica anterior, el *s_nonce* y el *c_nonce* utilizados en una sesión se generan y actualizan por el cliente y el servidor respectivamente, con lo que se impone una mayor carga de gestión en el cliente y en el servidor.

En la forma de realización 10 de la presente invención, se utiliza un solo número aleatorio en una sesión para sustituir los *s_nonce* y *c_nonce* en la técnica anterior y para poner en práctica la autenticación entre el cliente y el servidor. La misma sesión está asegurada por un mecanismo de seguridad de capa de transmisión o un mecanismo de seguridad de aplicación y por lo tanto, el mismo número aleatorio se puede utilizar para poner en práctica la autenticación entre el cliente y el servidor.

El número aleatorio puede generarse por el servidor o el cliente. Suponiendo que el número aleatorio se genere por el servidor, el método de autenticación en la forma de realización 10 de la presente invención se elabora como sigue.

En la forma de realización 10, se dan a conocer dos métodos de actualización del número aleatorio, según se describe a continuación:

Método 1: El servidor actualiza el número aleatorio.

En primer lugar, el servidor emite una orden de actualización de número aleatorio (Next*Nonce*). La orden Next*Nonce* transmite un nuevo número aleatorio.

Después de recibir la orden Next*Nonce*, el cliente utiliza el nuevo número aleatorio transmitido en la orden Next*Nonce* para actualizar el número aleatorio memorizado.

La orden de actualización puede transmitirse en un mensaje de autenticación, esto es, el mensaje transmite la orden de actualización y la información de autenticación del servidor. La orden de actualización puede transmitirse también en otro mensaje de gestión, esto es, el mensaje no transmite la información de autenticación del servidor. Si la orden de actualización se transmite en el mensaje de autenticación, después de que el cliente reciba el mensaje, el cliente actualiza el número aleatorio en conformidad con la orden Next*Nonce* y luego, utiliza el número aleatorio actualizado para generar un Digest y para la autenticación de la información. La autenticación es operativamente satisfactoria. En este caso, si un servidor operativamente malicioso intercepta el mensaje, dicho servidor puede iniciar ataques sobre el cliente en cualquier momento. Para impedir dichos ataques, cuando la orden Next*Nonce* se transmite en el mensaje de autenticación, el cliente utiliza el Digest generado empleando el número aleatorio no actualizado para la autenticación de la información. Después de recibir el mensaje, el cliente utiliza el Digest generado empleando el número aleatorio no actualizado para la autenticación en primer lugar. Después de la autenticación satisfactoria, el cliente actualiza el número aleatorio memorizado en conformidad con la orden Next*Nonce*. Si otro mensaje de gestión transmite la orden de actualización del número aleatorio, puesto que un mensaje que transmite el nuevo número aleatorio y otro mensaje que transmite la información de autenticación se envían a la parte opuesta por separado, no existe ningún riesgo de ataques para la reproducción.

Según se indica en la Figura 13, suponiendo que la respuesta se transmite en la orden Next*Nonce*, el proceso es como sigue:

Etapa 1301: El servidor envía un mensaje Trigger al cliente para iniciar una sesión.

El mensaje Trigger transmite: un Digest generado utilizando el número aleatorio compartido y la información TriggerInfo.

El número aleatorio compartido se genera por el servidor y está disponible para el servidor y el cliente.

En la práctica, el número aleatorio compartido, en esta etapa, puede ser un número aleatorio de mensaje Trigger o número aleatorio por defecto. A veces, el servidor no puede utilizar un número aleatorio, sino que utiliza el ID del servidor y la contraseña para generar el mensaje Trigger para iniciar una sesión y el cliente utiliza el ID del servidor y su contraseña para generar un Digest para la autenticación del mensaje Trigger.

Etapa 1302: El cliente envía una demanda de sesión al servidor.

Después de recibir el mensaje Trigger, el cliente utiliza el *s_nonce* memorizado para generar un Digest y para la autenticación del mensaje Trigger. Si la autenticación es operativamente satisfactoria, el cliente envía una demanda de sesión al servidor para iniciar una sesión,.

La demanda de sesión transmite: un identificador SessionID y la información de autenticación (Authenticate) que incluye el Digest generado utilizando el número aleatorio compartido.

En esta etapa, se establece una conexión de sesión entre el cliente y el servidor.

Etapa 1303: El servidor envía una respuesta que transmite el resultado de autenticación y la demanda de autenticación y la respuesta transmite la orden Next*Nonce*.

En conformidad con la información de autenticación enviada por el cliente, el servidor realiza la autenticación del cliente. Después de la autenticación satisfactoria, si el servidor descubre que necesita actualizarse el número aleatorio compartido, el servidor genera un nuevo número aleatorio compartido y luego, reenvía una respuesta que transmite el resultado de la autenticación y la demanda de autenticación al cliente. La respuesta transmite la orden Next*Nonce*.

Más concretamente, la respuesta transmite: un resultado de la autenticación del servidor para el cliente, un identificador SessionID, información de autenticación (Authenticate) que incluye el Digest generado utilizando el número aleatorio no actualizado y una orden Next*Nonce* que incluye el nuevo número aleatorio.

Etapa 1304: Después de recibir la respuesta, el cliente utiliza el número aleatorio no actualizado para la autenticación del mensaje.

Etapa 1305: La autenticación es operativamente satisfactoria y el cliente utiliza el nuevo número aleatorio transmitido en la orden Next*Nonce* para actualizar el número aleatorio compartido memorizado según se indica por la orden Next*Nonce*.

Etapa 1306: El cliente reenvía un mensaje que transmite el resultado de la autenticación y el resultado de la actualización al servidor.

Más concretamente, este mensaje transmite: un resultado de la autenticación del cliente para el servidor, un resultado de la actualización del número aleatorio compartido y otra información pertinente.

El servidor y el cliente pueden definir el periodo de validez del número aleatorio compartido de forma diferente. Cuando el servidor determina que el número aleatorio compartido es válido, el periodo de validez del número aleatorio compartido puede haber finalizado para el cliente. Por lo tanto, para mantener la validez del número aleatorio compartido para el cliente, la forma de realización 10 de la presente invención da a conocer una solución técnica en la que el cliente demanda al servidor para actualizar el número aleatorio compartido.

El cliente puede utilizar la orden Alert entre las órdenes DM para demandar al servidor que actualice el número aleatorio compartido. Para hacer que el servidor entienda la orden, se añade un tipo de alerta en la orden. El tipo de Alerta es una indicación de solicitar al servidor la actualización del número aleatorio.

Cuando el cliente cree que necesita actualizarse el número aleatorio, el cliente envía una demanda de actualización del número aleatorio compartido al servidor a través de la orden Alert. La demanda puede transmitirse en el mensaje de autenticación u otro mensaje de gestión. Después de recibir la demanda, el servidor decide si actualizar, o no, el número aleatorio según las condiciones específicas.

El tipo de Alerta puede definirse como: org.openmobilealliance.Next*Nonce*.

A continuación, se proporciona una instancia operativa de un mensaje del tipo Alerta:

```
<Alert>
<CmdID>2</CmdID>
<Datos>1226</Datos><!- - Alerta genérica - ->
<Item>
<Meta>
<Type xmlns= “syncml:metinf”>
org.openmobilealliance.NextNonce
</Tipo>
</Meta>
<Datos/>
</Item>
</Alerta>
```

El método de la actualización del cliente del número aleatorio es similar al método del servidor para actualizar el número aleatorio y por ello no se repite en esta descripción.

Método 2: El servidor y el cliente actualizan conjuntamente el número aleatorio.

El servidor y el cliente pueden generar un nuevo número aleatorio para su actualización cuando determinen que necesita actualizarse el número aleatorio compartido.

El número aleatorio puede actualizarse mediante una orden Next*Nonce*. A continuación se proporciona una instancia operativa de actualización:

<Chal>

<Meta>

<NextNonce xmlns= ‘syncml:metinf’>LG3iZQhhdmKNHg==</NextNonce>

</Meta>

</Chal>

La orden Next*Nonce* puede transmitirse en un mensaje en el proceso de sesión. A modo de ejemplo, el cliente puede añadir la orden Next*Nonce* a una demanda de sesión y enviar la demanda de sesión al servidor, solicitando al servidor que actualice el número aleatorio compartido o que añade la orden Next*Nonce* en otro mensaje.

Sin importar que el servidor o el cliente actualice, o no, el número aleatorio compartido, si la orden de actualización se transmite en el mensaje de autenticación, después de que la parte opuesta reciba el mensaje, la parte opuesta actualiza primero el número aleatorio y luego, utiliza el número aleatorio actualizado para generar un Digest y para la autenticación de la información. La autenticación es así operativamente satisfactoria. En este caso, si un servidor operativamente malicioso intercepta el mensaje, dicho servidor puede iniciar ataques de reproducción sobre el cliente en cualquier momento. Para evitar dichos ataques, cuando la orden Next*Nonce* se transmita en el mensaje de autenticación, la parte opuesta puede utilizar el Digest generado utilizando el número aleatorio no actualizado para la autenticación de la información. Después de la autenticación operativamente satisfactoria, la parte opuesta actualiza el número aleatorio memorizado en conformidad con la orden Next*Nonce*. Si otro mensaje de gestión transmite la orden de actualización de número aleatorio, puesto que un mensaje que transmite el nuevo número aleatorio y otro mensaje que transmite la información de autenticación se envían por separado a la parte opuesta, no existe ningún riesgo de ataques.

En la forma de realización 10 de la presente invención, el servidor y el cliente utilizan un número aleatorio compartido para la autenticación. En este caso, si el número aleatorio compartido es erróneo, el error puede gestionarse mediante cualquiera de los métodos anteriormente descritos. En la etapa 803, en la forma de realización 5 de la presente invención, el mensaje transmite un nuevo número aleatorio, un Digest generado utilizando el nuevo número aleatorio. En este caso, si un servidor, operativamente malicioso, intercepta el mensaje, dicho servidor puede iniciar un ataque enviando el mensaje al servidor o al cliente, de forma repetida. El servidor o el cliente es incapaz de identificar el mensaje, cree que el mensaje es válido y realiza las operaciones correspondientes. Para evitar dichos ataques, el número aleatorio no actualizado puede utilizarse para generar un Digest. De este modo, después de recibir el mensaje, el servidor utiliza el número aleatorio no actualizado para calcular el Digest y para la autenticación del remitente del mensaje, esto es, el cliente y luego, actualiza el número aleatorio memorizado en conformidad con la orden Next*Nonce*.

La forma de realización 10 de la presente invención libera efectivamente la carga de trabajo del sistema.

Según se indica en la Figura 14, un sistema de autenticación dado a conocer en la forma de realización 1 de la presente invención comprende:

un servidor 1410, adaptado para: utilizar el número aleatorio del mensaje Trigger para generar un mensaje Trigger y para enviar el mensaje Trigger generado y

un cliente 1420, adaptado para: recibir el mensaje Trigger generado utilizando el número aleatorio de mensaje Trigger y para utilizar el número aleatorio del mensaje Trigger para efectuar la autenticación del mensaje Trigger generado, esto es, verificar la validez del mensaje Trigger.

El servidor 1410 comprende, además:

una primera unidad de generación 1412, adaptada para generar un mensaje Trigger utilizando un número aleatorio de mensaje Trigger;

una unidad de envío 1411, adaptada para enviar el mensaje Trigger generado utilizando el número aleatorio del mensaje Trigger;

una segunda unidad de generación 1417, adaptada para: generar un mensaje Trigger utilizando un número aleatorio de servidor y para enviar el mensaje Trigger generado al cliente;

una unidad de determinación 1413, adaptada para controlar la primera unidad de generación 1412 para utilizar el número aleatorio de mensaje Trigger para generar un mensaje Trigger después de determinar que el mensaje Trigger generado utilizando el número aleatorio del servidor tiene una autenticación operativamente insatisfactoria;

una unidad de tiempo 1414, adaptada para: determinar el tiempo del sistema del servidor cuando la primera unidad de generación 1412 utiliza el número aleatorio de mensaje Trigger para generar un mensaje Trigger y para añadir el tiempo del sistema en el mensaje Trigger generado utilizando el número aleatorio del mensaje Trigger;

una unidad de codificación 1415, adaptada para: numerar los mensajes Trigger generados por la primera unidad de generación 1412 utilizando el número aleatorio de mensaje Trigger y empleando el número como el número aleatorio de mensaje Trigger;

una unidad de restablecimiento de número aleatorio 1416, adaptada para el restablecimiento del número aleatorio de mensaje Trigger generado por la unidad de codificación 1415 cuando sea necesario y

una unidad de ID de sesión para número aleatorio 1418 adaptada para: utilizar el ID de sesión de la sesión iniciada por el mensaje Trigger como un número aleatorio de mensaje Trigger, de modo que el cliente realice la autenticación del mensaje Trigger utilizando el número aleatorio de mensaje Trigger después de recibir el mensaje Trigger y envía una demanda de sesión para demandar el establecimiento de una sesión correspondiente al ID de sesión después de la autenticación operativamente satisfactoria.

La unidad de codificación 1415 comprende, además:

una unidad de numeración en orden ascendente 14151, adaptada para numerar los mensajes Trigger generados utilizando el número aleatorio de mensaje Trigger en orden ascendente y

una unidad de numeración en orden descendente 14152, adaptada para numerar los mensajes Trigger generados utilizando el número aleatorio de mensaje Trigger en orden descendente.

El cliente 1420 comprende, además:

una unidad de recepción 1421, adaptada para recibir el mensaje Trigger generado utilizando el número aleatorio de mensaje Trigger;

una primera unidad de autenticación 1422, adaptada para: utilizar el número aleatorio de mensaje Trigger para la autenticación del mensaje Trigger generado empleando el número aleatorio de mensaje Trigger, esto es, verificar la validez del mensaje Trigger generado utilizando el número aleatorio de mensaje Trigger;

una segunda unidad de ate 1425, adaptada para utilizar el número aleatorio del servidor para la autenticación del mensaje Trigger después de recibir el mensaje Trigger y, si falla la autenticación, emplear el número aleatorio de mensaje Trigger para una nueva autenticación del mensaje Trigger;

una primera unidad de determinación de la validez 1423, adaptada para: determinar el tiempo local del cliente cuando la unidad de recepción 1421 recibe el mensaje Trigger generado utilizando el número aleatorio de mensaje Trigger y para comparar el valor absoluto de la diferencia entre el tiempo local y el tiempo del sistema con un umbral preestablecido; si el valor absoluto es menor que el umbral preestablecido, para determinar que el número aleatorio de mensaje Trigger es válido y para controlar la primera unidad de autenticación 1422 para utilizar el número aleatorio de mensaje Trigger para la autenticación del mensaje Trigger generado utilizando el número aleatorio de mensaje Trigger;

una segunda unidad de determinación de la validez 1424, adaptada para: memorizar el número aleatorio de mensaje Trigger transmitido en el mensaje Trigger si se determina que el número aleatorio de mensaje Trigger transmitido en el mensaje Trigger es válido en conformidad con el mensaje Trigger memorizado después de que la unidad de recepción 1421 reciba el mensaje generado utilizando el número aleatorio de mensaje Trigger y para controlar la primera unidad de autenticación 1422 para utilizar el número aleatorio de mensaje Trigger para la autenticación del mensaje Trigger generado empleando el número aleatorio de mensaje Trigger y

una unidad de identificador ID de sesión para número aleatorio 1428, adaptada para: utilizar el ID de sesión de la sesión iniciada por el mensaje Trigger como un número aleatorio de mensaje Trigger para la autenticación del mensaje Trigger empleando el número aleatorio de mensaje Trigger y para enviar una demanda de sesión para solicitar el establecimiento de una sesión correspondiente al ID de sesión después de la autenticación operativamente satisfactoria.

La segunda unidad de determinación de la validez 1424 comprende:

una primera unidad de determinación de la numeración 14241, adaptada para: comparar el número aleatorio de mensaje Trigger memorizado en el cliente con el número aleatorio de mensaje Trigger transmitido en el mensaje Trigger y para determinar que el número aleatorio de mensaje Trigger, transmitido en el mensaje Trigger, es válido, si el número aleatorio de mensaje Trigger, transmitido en el mensaje Trigger, es mayor que el número aleatorio de mensaje Trigger máximo memorizado en el cliente o si los valores del número aleatorio de mensaje Trigger que han sido recibidos y memorizados por el cliente no incluyen el número aleatorio de mensaje Trigger transmitido en el mensaje Trigger o si los valores de los números aleatorios que no se han recibido por el cliente incluyen el número aleatorio de mensaje Trigger transmitido en el mensaje Trigger y

una segunda unidad de determinación de la numeración 14242, adaptada para: comparar el número aleatorio de mensaje Trigger memorizado en el cliente con el número aleatorio de mensaje Trigger transmitido en el mensaje Trigger y para determinar que el número aleatorio de mensaje Trigger transmitido en el mensaje Trigger es válido si el número aleatorio de mensaje Trigger transmitido en el mensaje Trigger es menor que el número aleatorio de mensaje Trigger mínimo memorizado en el cliente o si los valores de los números aleatorios de mensajes Trigger que se han recibido y memorizado por el cliente no incluyen el número aleatorio de mensaje Trigger transmitido en el mensaje Trigger o si los valores de números aleatorios que no se han recibido por el cliente incluyen el número aleatorio de mensaje Trigger transmitido en el mensaje Trigger.

El modo de funcionamiento del sistema de autenticación en la forma de realización 1, es similar al modo operativo del método de autenticación en la forma de realización 1 y en la forma de realización 2 de la presente invención y no se repite por ello en esta descripción.

Por intermedio del sistema de autenticación dado a conocer en la forma de realización 1 de la presente invención, cuando falla la autenticación, no se requiere ningún número aleatorio por defecto para poner en práctica la autenticación, con lo que se mejora la seguridad del sistema.

El servidor dado a conocer en la forma de realización 1 de la presente invención es básicamente el mismo que el servidor en el sistema de autenticación dado a conocer en la forma de realización 1 de la presente invención y no se repite por ello en esta descripción.

Según se ilustra en la Figura 15, el cliente, dado a conocer en la forma de realización 1 de la presente invención comprende:

una unidad de recepción 1501, adaptada para recibir un mensaje Trigger enviado por el servidor;

una primera unidad de generación 1512, adaptada para: generar un nuevo número aleatorio de servidor si se determina que el número aleatorio del servidor necesita actualizarse después de recibir el mensaje Trigger, para añadir el nuevo número aleatorio de servidor a una demanda de sesión y para enviar la demanda de sesión al servidor, de modo que el servidor pueda utilizar el nuevo número aleatorio de servidor para actualizar el número aleatorio de servidor memorizado después de recibir la demanda de sesión que transmite el nuevo número aleatorio de servidor y

una segunda unidad de generación 1503, adaptada para: generar un nuevo número aleatorio de servidor si se decide no iniciar una sesión y la determinación de que necesita actualizarse el número aleatorio de servidor después de recibir el mensaje Trigger, para añadir el nuevo número aleatorio de servidor en una respuesta de estado y para enviar la respuesta de estado al servidor, de modo que el servidor pueda utilizar el nuevo número aleatorio de servidor para actualizar el número aleatorio de servidor memorizado después de recibir la respuesta de estado que transmite el nuevo número aleatorio de servidor.

El modo operativo del cliente, dado a conocer en la forma de realización 1, es similar al modo operativo del cliente en el método de autenticación dado a conocer en la cuarta, quinta y sexta formas de realización de la presente invención y por ello no se repite en esta descripción.

A través del cliente dado a conocer en la forma de realización 1 de la presente invención, cuando necesita actualizarse el *s_nonce*, la demanda de sesión transmite directamente una orden de actualización, con lo que se reduce la frecuencia de interacción de señalización, se reduce la carga de trabajo del sistema, se disminuye la frecuencia de utilización del número aleatorio por defecto para realizar la autenticación y se mejora la seguridad del sistema.

Según se ilustra en la Figura 16, el cliente 1600 dado a conocer en la forma de realización 2 de la presente invención comprende:

una unidad de recepción 1601, adaptada para recibir un mensaje Trigger enviado por el servidor;

una unidad de generación 1602, adaptada para: utilizar un número aleatorio de servidor para la autenticación del mensaje Trigger después de recibir el mensaje Trigger; si falla la autenticación, para utilizar el número aleatorio por defecto para la autenticación del mensaje Trigger; después de la autenticación operativamente satisfactoria, para utilizar

un número aleatorio de cliente para generar una demanda de sesión y para enviar la demanda de sesión al servidor, de modo que el servidor pueda utilizar el número aleatorio del cliente para la autenticación del cliente y

una unidad de cambio de contraseña 1603, adaptada para cambiar la contraseña del servidor y la contraseña del cliente después de la actualización del número aleatorio del servidor con el nuevo número aleatorio del servidor.

El modo operativo del cliente dado a conocer en la forma de realización 2 es similar al modo operativo del cliente en el método de autenticación dado a conocer en la séptima y octava formas de realización de la presente invención y por ello, no se repite en esta descripción.

A través del cliente dado a conocer en la forma de realización 2 de la presente invención, cuando necesita actualizarse *s_nonce*, solamente se actualiza el *s_nonce* y no se actualiza el *c_nonce*. Aún cuando el sistema utilice el número aleatorio por defecto para la autenticación cuando se gestiona el error de *s_nonce*, puesto que el *c_nonce* no necesita actualizarse, el cliente puede utilizar el *c_nonce* para generar una demanda de sesión, con lo que se reduce la frecuencia de utilización del número aleatorio por defecto y se mejora la seguridad del sistema.

Según se indica en la Figura 17, un sistema de autenticación dado a conocer en la forma de realización 2 de la presente invención incluye un servidor 1710 y un cliente 1720.

El servidor 1710 comprende, además:

una unidad de disparo operativo 1711, adaptada para: utilizar un número aleatorio compartido por el servidor y el cliente para generar un mensaje Trigger y para enviar el mensaje Trigger al cliente, de modo que el cliente pueda utilizar el número aleatorio compartido para la autenticación del mensaje Trigger después de recibir el mensaje Trigger;

una unidad de recepción 1712, adaptada para recibir la demanda de sesión generada utilizando el número aleatorio compartido desde el cliente;

una unidad de autenticación 1713, adaptada para la autenticación de la demanda de sesión utilizando el número aleatorio compartido;

una unidad de generación 1714, adaptada para: utilizar un número aleatorio compartido para generar una respuesta después de que se realice la autenticación satisfactoria de la demanda de sesión y para enviar la respuesta al cliente, de modo que el cliente puede utilizar el número aleatorio compartido para la autenticación de la respuesta después de recibir dicha respuesta;

una unidad de actualización 1715, adaptada para: generar un número aleatorio compartido; cuando necesite actualizarse el número aleatorio compartido, para generar un nuevo número aleatorio compartido y para enviar un mensaje de actualización de número aleatorio que transmita el nuevo número aleatorio compartido al cliente, de modo que el cliente pueda utilizar el nuevo número aleatorio compartido para actualizar el número aleatorio compartido después de recibir el mensaje de actualización del número aleatorio y

una unidad de demanda 1716, adaptada para: cuando se determine que necesita actualizarse el número aleatorio compartido, para enviar una demanda de actualización de número aleatorio al cliente, de modo que el cliente pueda generar un nuevo número aleatorio compartido después de recibir la demanda de actualización de número aleatorio y para decidir la actualización del número aleatorio y para enviar un mensaje de actualización del número aleatorio que transmite el nuevo número aleatorio compartido.

El cliente 1720 comprende, además:

una unidad de recepción 1721, adaptada para recibir el mensaje Trigger que se envía por el servidor y se genera utilizando el número aleatorio compartido por el servidor y el cliente;

una primera unidad de autenticación 1722, adaptada para la autenticación del mensaje Trigger utilizando el número aleatorio compartido después de la recepción del mensaje Trigger;

una unidad de generación 1723, adaptada para: utilizar un número aleatorio compartido para generar una demanda de sesión después de la autenticación operativamente satisfactoria y para enviar la demanda de sesión al servidor, de modo que el servidor pueda utilizar el número aleatorio compartido para la autenticación de la demanda de sesión después de recibir la demanda de sesión, esto es, para verificar la validez de la demanda de sesión;

una segunda unidad de autenticación 1724, adaptada para utilizar el número aleatorio compartido para la autenticación de la respuesta después de recibir la respuesta generada por el servidor utilizando el número aleatorio compartido;

una unidad de actualización 1725, adaptada para: generar un número aleatorio compartido; cuando necesite actualizarse el número aleatorio compartido, para generar un nuevo número aleatorio compartido y para enviar un mensaje de

actualización de número aleatorio que transmita el nuevo número aleatorio compartido al servidor, de modo que el servidor pueda utilizar el nuevo número aleatorio compartido para actualizar el número aleatorio compartido después de recibir el mensaje de actualización del número aleatorio y

una unidad de demanda 1726, adaptada para: cuando se determine que necesita actualizarse el número aleatorio, para enviar una demanda de actualización del número aleatorio al servidor, de modo que el servidor pueda generar un nuevo número aleatorio compartido después de recibir la demanda de actualización de número aleatorio y para decidir actualizar el número aleatorio y enviar un mensaje de actualización de número aleatorio que transmita el nuevo número aleatorio compartido.

El modo operativo del sistema de autenticación, en la forma de realización 2, es similar al modo operativo del método de autenticación en la forma de realización 9 de la presente invención y por ello, no se repite en esta descripción.

A través del sistema de autenticación dado a conocer en la forma de realización 2 de la presente invención, el servidor y el cliente comparten un número aleatorio en el proceso de sesión en lugar de *s_nonce* y del *c_nonce* en la técnica anterior para poner en práctica la autenticación entre el cliente y el servidor, con lo que se libera efectivamente la carga de trabajo del sistema.

El servidor dado a conocer en la forma de realización 2 de la presente invención y el cliente dado a conocer en la forma de realización 3 de la presente invención son esencialmente el mismo que el servidor y el cliente en el sistema de autenticación dado a conocer en la forma de realización 2 de la presente invención y por ello no se repiten en esta descripción.

Es comprensible para los expertos en esta técnica que la totalidad o parte de las etapas de las formas de realización anteriores pueden ponerse en práctica por hardware bajo las instrucciones de un programa informático. El programa puede memorizarse en un medio de almacenamiento legible por ordenador. El medio de memorización puede ser una Memoria de Solamente Lectura (ROM), un disco magnético o un Disco Compacto (CD).

Lo que antecede es un método de autenticación basado en un protocolo DS o un protocolo DM, un sistema, un servidor y un cliente bajo la presente invención. Aunque la invención se describe a través de algunas formas de realización, a modo de ejemplo, la invención no está limitada a dichas formas de realización. Es evidente para los expertos en esta técnica que se pueden realizar modificaciones y variaciones a la invención sin desviarse por ello del alcance de protección de la invención. La invención cubrirá las modificaciones y variaciones a condición de que caigan dentro del alcance de protección definido por las siguientes reivindicaciones o sus equivalentes.

**REIVINDICACIONES**

**1.** Un método de autenticación basado en el protocolo de Sincronización de Datos, DS, o de Gestión de Dispositivo, DM, estando dicho método caracterizado por cuanto que comprende:

la recepción (301), por un cliente, de un mensaje de disparo operativo procedente de un servidor que utiliza un número aleatorio del mensaje de disparo operativo para generar el mensaje de disparo;

la extracción, por el cliente, del *número aleatorio* de mensaje de disparo operativo;

la utilización (302), por el cliente, del *número aleatorio* de mensaje de disparo operativo con el fin de generar un extracto de acceso y la autenticación del mensaje de disparo operativo generado utilizando el *número aleatorio* de mensaje de disparo operativo después de determinar que dicho *número aleatorio* de mensaje de disparo es válido y

el envío (302), por el cliente, de una demanda de sesión al servidor indicado por el mensaje de disparo operativo después de la realización satisfactoria de la autenticación, en donde la demanda de sesión soporta un identificador ID de sesión.

**2.** El método según la reivindicación 1, caracterizado por cuanto que, antes de utilizar el *número aleatorio* de mensaje de disparo operativo para generar un mensaje de disparo, el método comprende, además:

la utilización, por el servidor, de un número aleatorio de servidor para generar un mensaje de disparo operativo;

el envío, por el servidor, del mensaje de disparo operativo generado utilizando el *número aleatorio* del servidor al cliente y

la utilización, por el servidor, del *número aleatorio* de mensaje de disparo operativo para generar el mensaje de disparo después de determinar que el mensaje de disparo generado mediante el uso del *número aleatorio* del servidor es objeto de autenticación insatisfactoria.

**3.** El método según la reivindicación 1 o 2, caracterizado por cuanto que un tiempo de sistema del servidor se utiliza por el servidor como el *número aleatorio* de mensaje de disparo operativo y se transmite en el mensaje de disparo;

el cliente determina si el *número aleatorio* de mensaje de disparo operativo es válido comparando un tiempo local del cliente con el *número aleatorio* de mensaje de disparo operativo después de recibir el mensaje de disparo generado utilizando el *número aleatorio* de mensaje de disparo operativo y

el cliente utiliza el *número aleatorio* de mensaje de disparo operativo para generar un extracto de acceso y autenticar el mensaje de disparo generado utilizando el *número aleatorio* de mensaje de disparo operativo con el empleo del extracto de acceso después de determinar que el *número aleatorio* del mensaje de disparo es válido.

**4.** El método según la reivindicación 1 o 2, caracterizado por cuanto que la utilización del *número aleatorio* del mensaje de disparo operativo para generar un mensaje de disparo por el servidor comprende:

la transmisión, por el servidor, del *número aleatorio* del mensaje de disparo operativo en una cabecera de mensaje o un cuerpo de mensaje del mensaje de disparo; la generación del extracto de acceso utilizando el *número aleatorio* del mensaje de disparo operativo, la cabecera del mensaje y el cuerpo del mensaje del mensaje de disparo y generando un mensaje de disparo utilizando el extracto de acceso o

la transmisión, por el servidor, del *número aleatorio* del mensaje de disparo operativo en una cabecera de mensaje o un cuerpo de mensaje del mensaje de disparo operativo; la generación del extracto de acceso utilizando la cabecera de mensaje y el cuerpo de mensaje del mensaje de disparo operativo y la generación del mensaje de disparo utilizando el extracto de acceso.

**5.** Un cliente, caracterizado por cuanto que comprende:

una unidad de recepción (1421), adaptada para recibir un mensaje de disparo operativo que se genera por un servidor utilizando un *número aleatorio* del mensaje de disparo operativo y en cumplimiento de un protocolo de Sincronización de Datos, DS o de Gestión de Dispositivo, DM y

una primera unidad de autenticación (1422), adaptada para extraer el *número aleatorio* del mensaje de disparo operativo; para generar un extracto de acceso utilizando el *número aleatorio* del mensaje de disparo operativo para autenticar el mensaje de disparo generado usando el *número aleatorio* del mensaje de disparo operativo después de determinar que el *número aleatorio* del mensaje de disparo operativo es válido; para enviar una demanda de sesión al servidor indicado por el mensaje de disparo operativo después de que se realice satisfactoriamente la autenticación.

**6.** El cliente según la reivindicación 5, caracterizado por cuanto que el cliente comprende, además:

una segunda unidad de autenticación (1425),adaptada para utilizar un *número aleatorio* de servidor para la autenticación del mensaje de disparo operativo después de recibir dicho mensaje y para realizar una nueva autenticación del mensaje de disparo operativo utilizando el *número aleatorio* del mensaje de disparo si falla la autenticación.

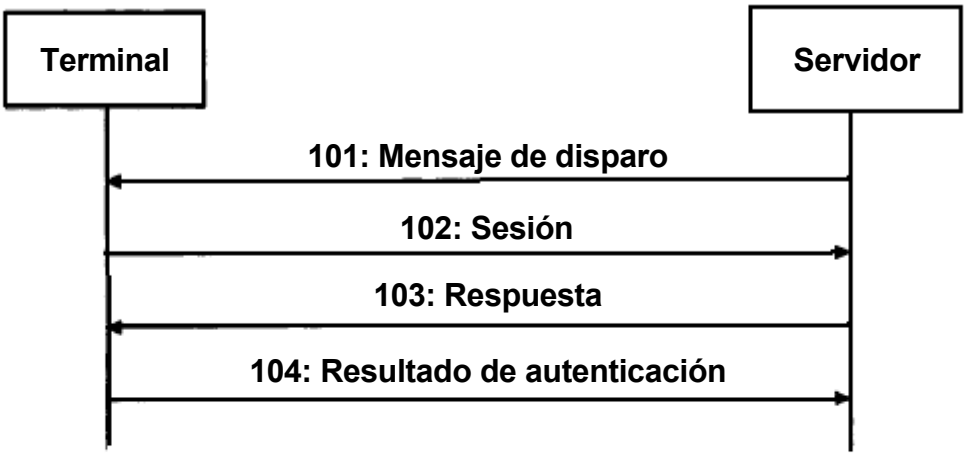
Terminal
Servidor
101: Mensaje de disparo
102: Sesión
103: Respuesta
104: Resultado de autenticación

FIG. 1

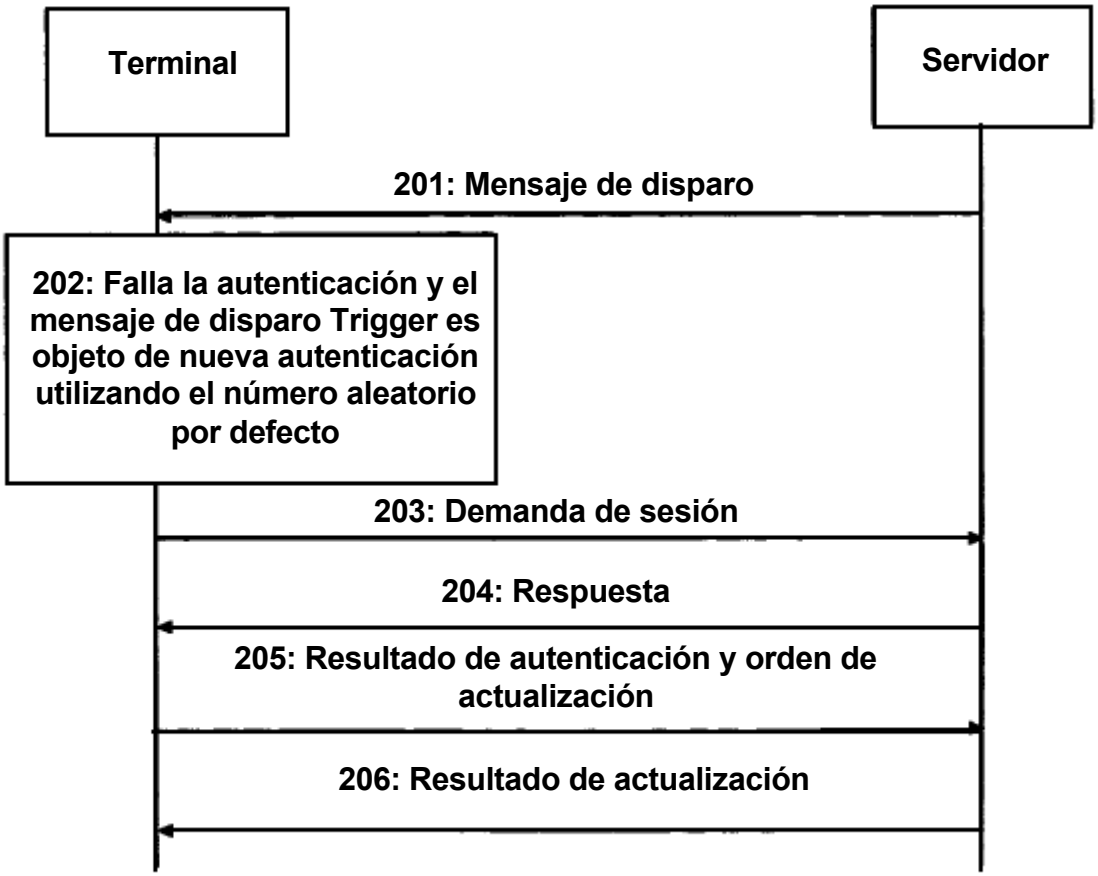
Terminal
Servidor
201: Mensaje de disparo
202: Falla la autenticación y el mensaje de disparo Trigger es objeto de nueva autenticación utilizando el número aleatorio por defecto
203: Demanda de sesión
204: Respuesta
205: Resultado de autenticación y orden de actualización
206: Resultado de actualización

FIG. 2

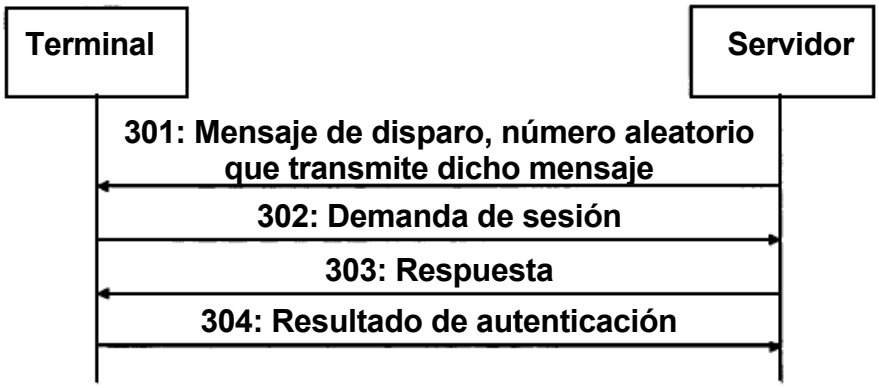


FIG. 3

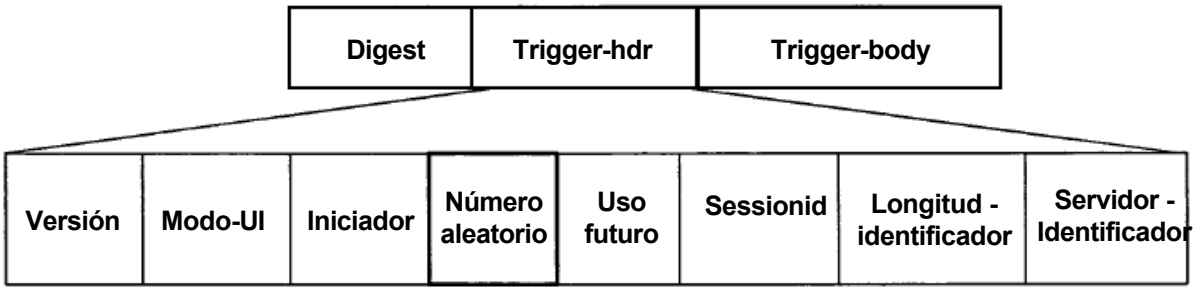


FIG. 4

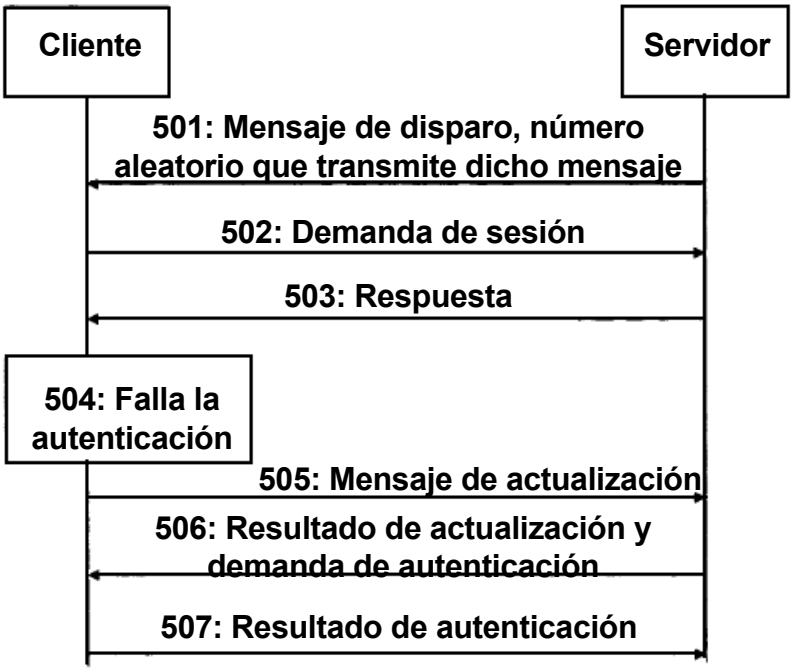


FIG. 5

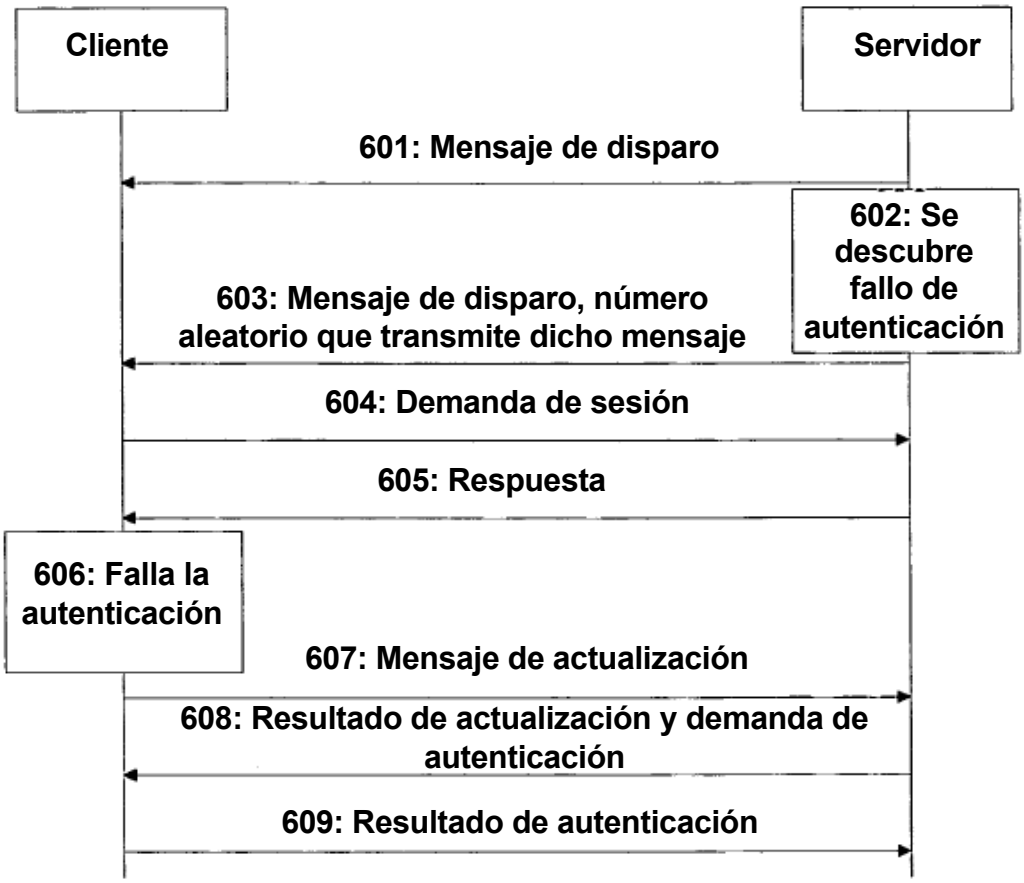
Cliente
Servidor
601: Mensaje de disparo
602: Se descubre fallo de autenticación
603: Mensaje de disparo, número aleatorio que transmite dicho mensaje
604: Demanda de sesión
605: Respuesta
606: Falla la autenticación
607: Mensaje de actualización
608: Resultado de actualización y demanda de autenticación
609: Resultado de autenticación

FIG. 6

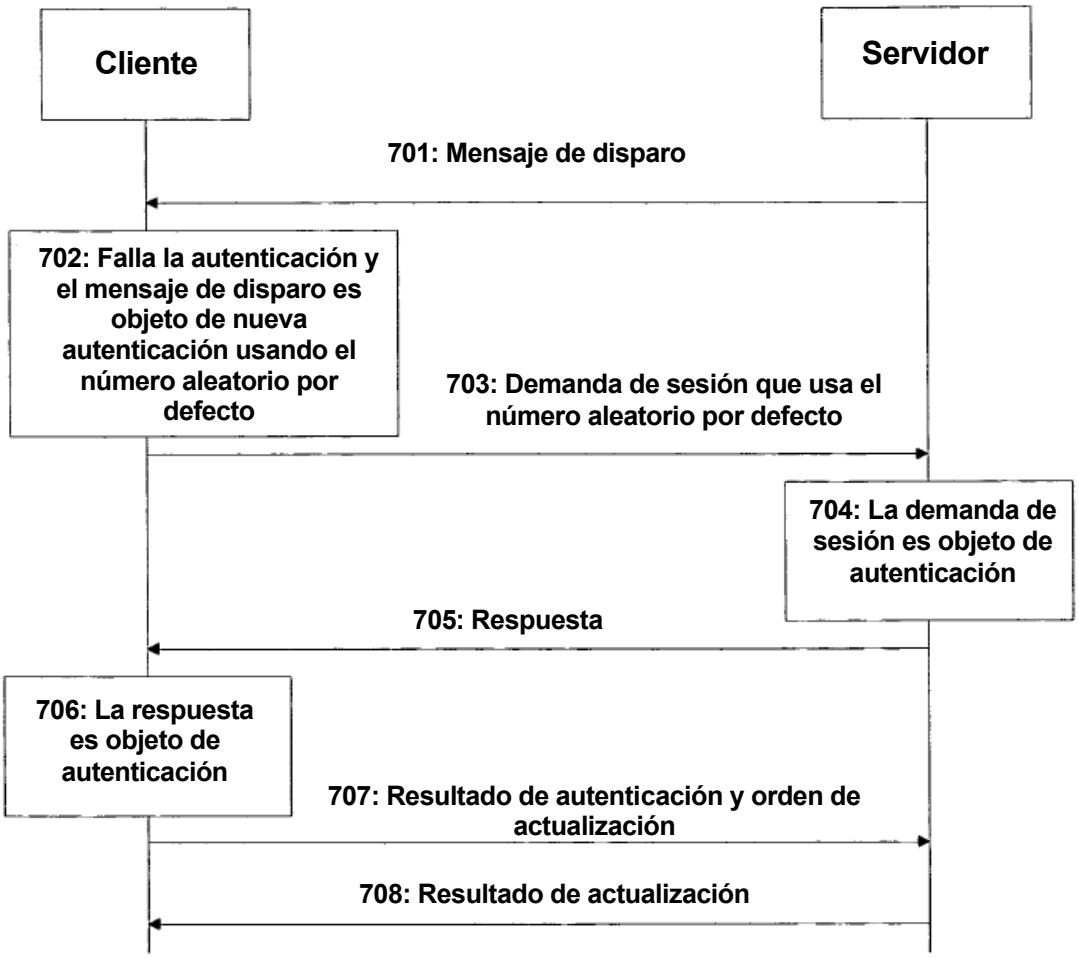
Cliente
Servidor
701: Mensaje de disparo
702: Falla la autenticación y el mensaje de disparo es objeto de nueva autenticación usando el número aleatorio por defecto
703: Demanda de sesión que usa el número aleatorio por defecto
704: La demanda de sesión es objeto de autenticación
705: Respuesta
706: La respuesta es objeto de autenticación
707: Resultado de autenticación y orden de actualización
708: Resultado de actualización

FIG. 7

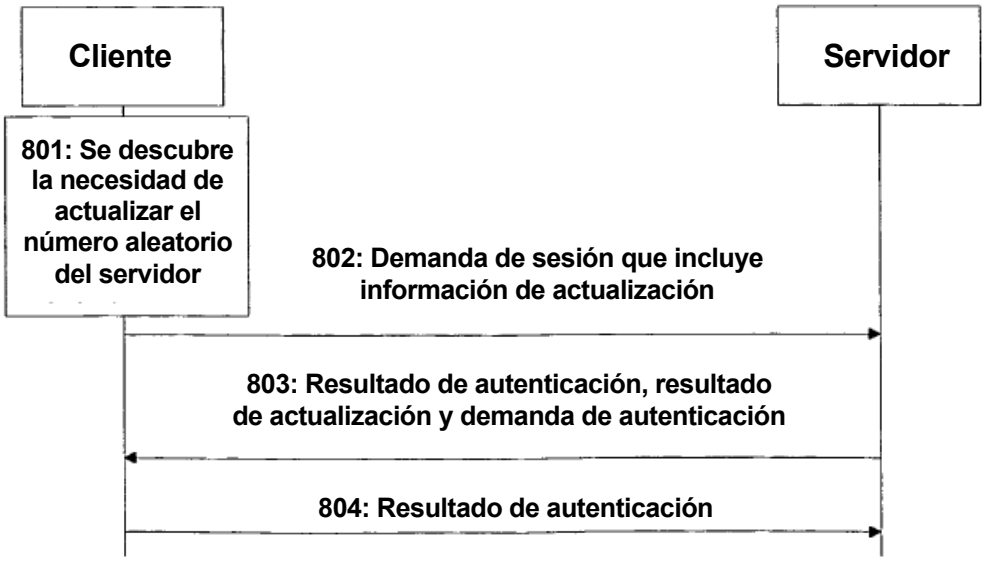
Cliente
Servidor
801: Se descubre la necesidad de actualizar el número aleatorio del servidor
802: Demanda de sesión que incluye información de actualización
803: Resultado de autenticación, resultado de actualización y demanda de autenticación
804: Resultado de autenticación

FIG. 8

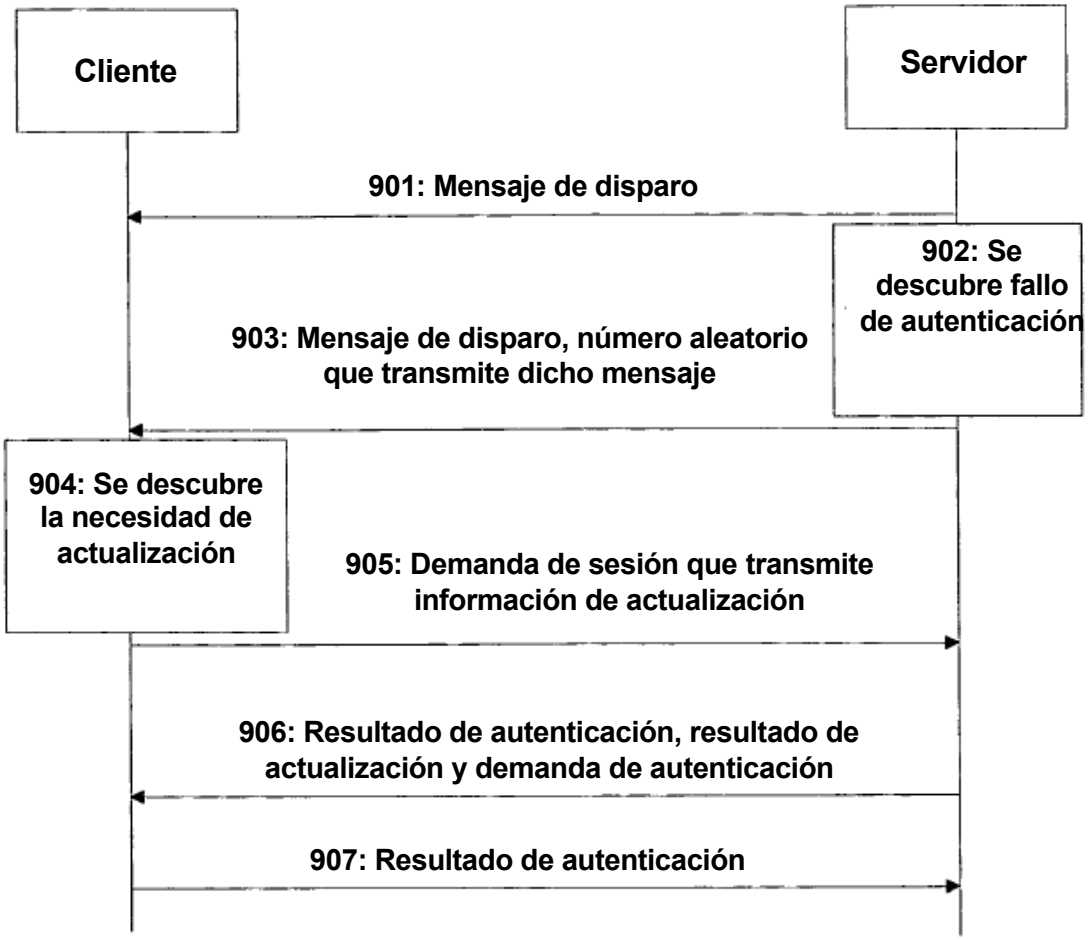
Cliente
Servidor
901: Mensaje de disparo
902: Se descubre fallo de autenticación
903: Mensaje de disparo, número aleatorio que transmite dicho mensaje
904: Se descubre la necesidad de actualización
905: Demanda de sesión que transmite información de actualización
906: Resultado de autenticación, resultado de actualización y demanda de autenticación
907: Resultado de autenticación

FIG. 9

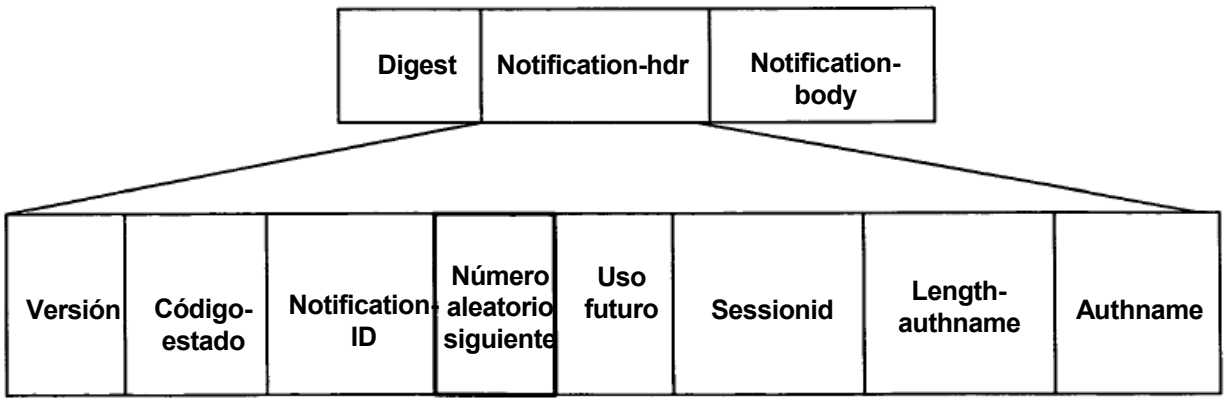
Digest
Notification-hdr
Notification-body
Versión
Código-estado
Notification-ID
Número aleatorio siguiente
Uso futuro
Sessionid
Length-authname
Authname

FIG. 10

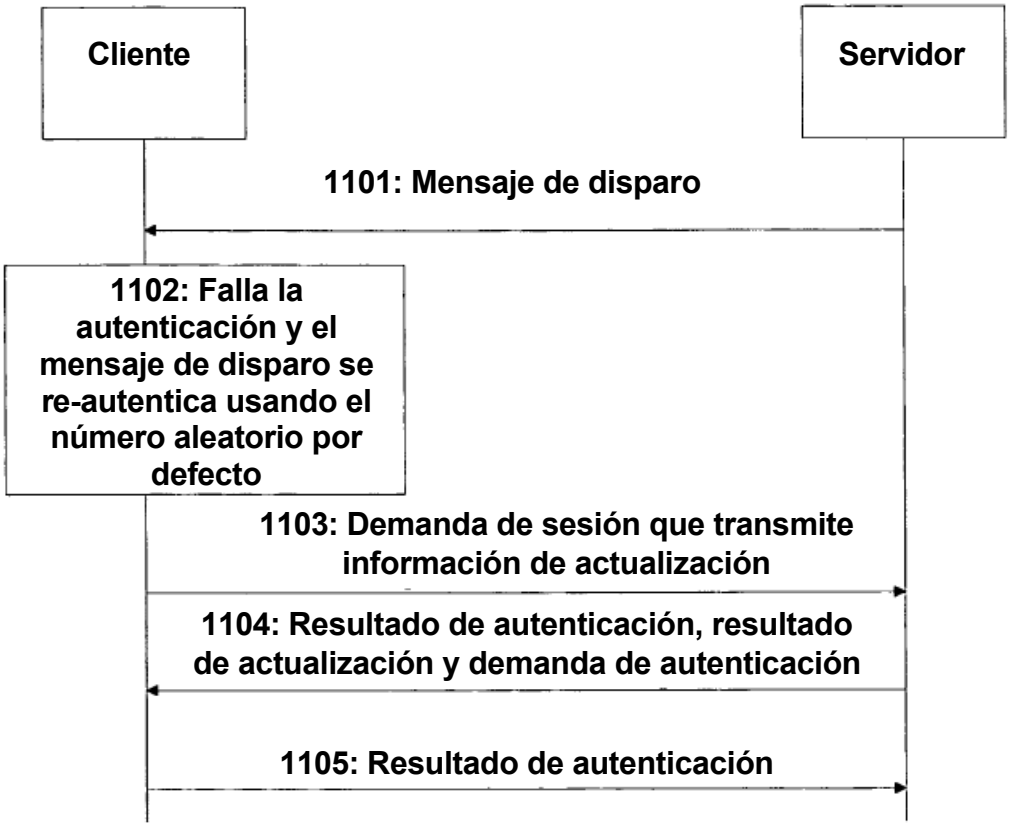
Cliente
Servidor
1101: Mensaje de disparo
1102: Falla la autenticación y el mensaje de disparo se re-autentica usando el número aleatorio por defecto
1103: Demanda de sesión que transmite información de actualización
1104: Resultado de autenticación, resultado de actualización y demanda de autenticación
1105: Resultado de autenticación

FIG. 11

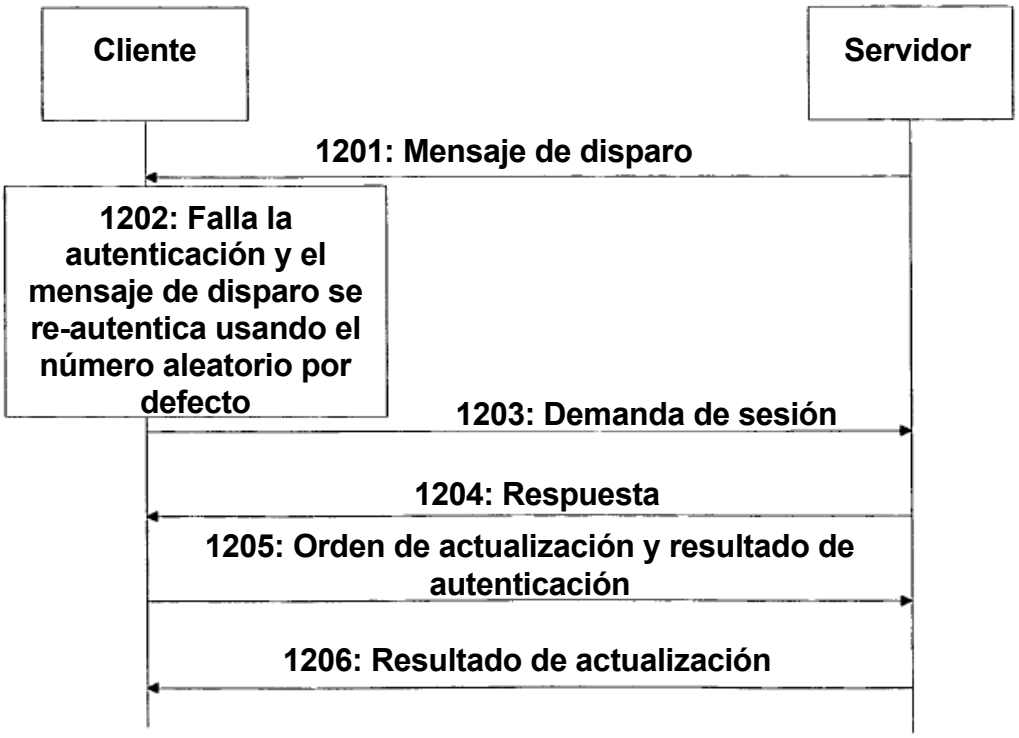
Cliente
Servidor
1201: Mensaje de disparo
1202: Falla la autenticación y el mensaje de disparo se re-autentica usando el número aleatorio por defecto
1203: Demanda de sesión
1204: Respuesta
1205: Orden de actualización y resultado de autenticación
1206: Resultado de actualización

FIG. 12

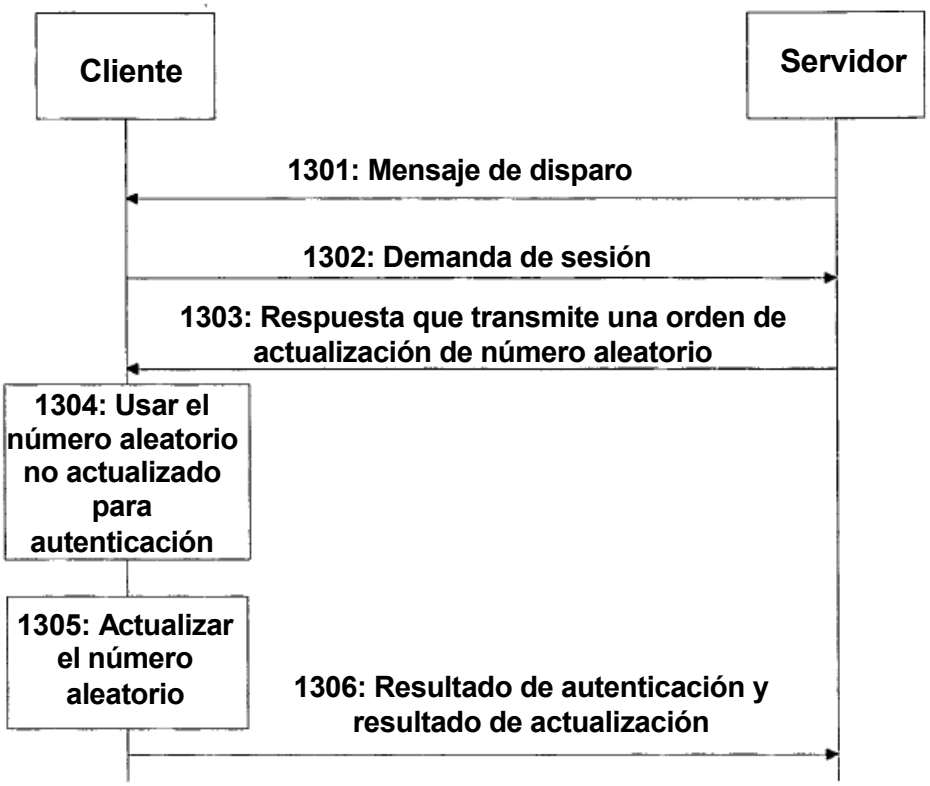
Cliente
Servidor
1301: Mensaje de disparo
1302: Demanda de sesión
1303: Respuesta que transmite una orden de actualización de número aleatorio
1304: Usar el número aleatorio no actualizado para autenticación
1305: Actualizar el número aleatorio
1306: Resultado de autenticación y resultado de actualización

FIG. 13

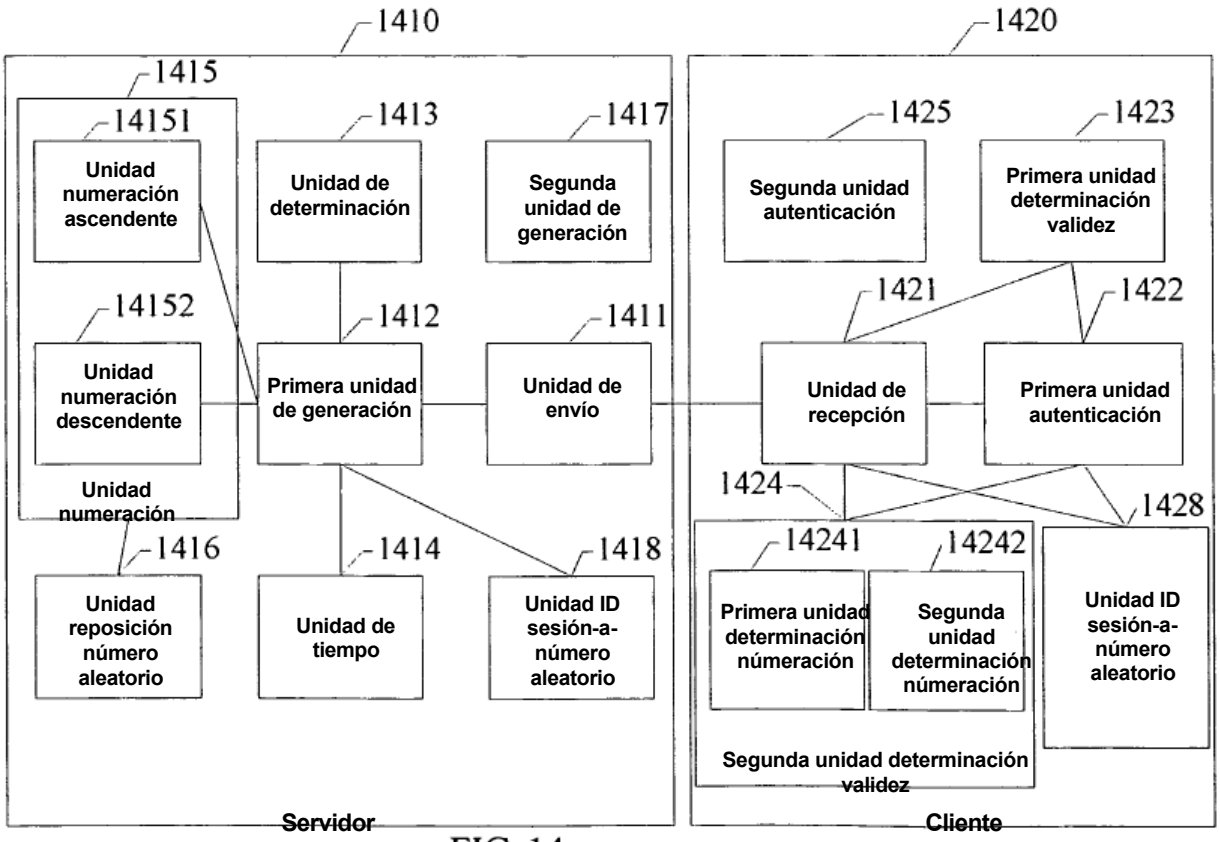
1410
1415
14151
Unidad numeración ascendente
14152
Unidad numeración descendente
Unidad numeración
1413
Unidad de determinación
1417
Segunda unidad de generación
1412
Primera unidad de generación
1411
Unidad de envío
1416
Unidad reposición número aleatorio
1414
Unidad de tiempo
1418
Unidad ID sesión-a-número aleatorio
Servidor
1420
1425
Segunda unidad autenticación
1423
Primera unidad determinación validez
1421
1422
Unidad de recepción
Primera unidad autenticación
1424
1428
14241
Primera unidad determinación numeración
14242
Segunda unidad determinación numeración
Segunda unidad determinación validez
Unidad ID sesión-a-número aleatorio
Cliente

FIG. 14

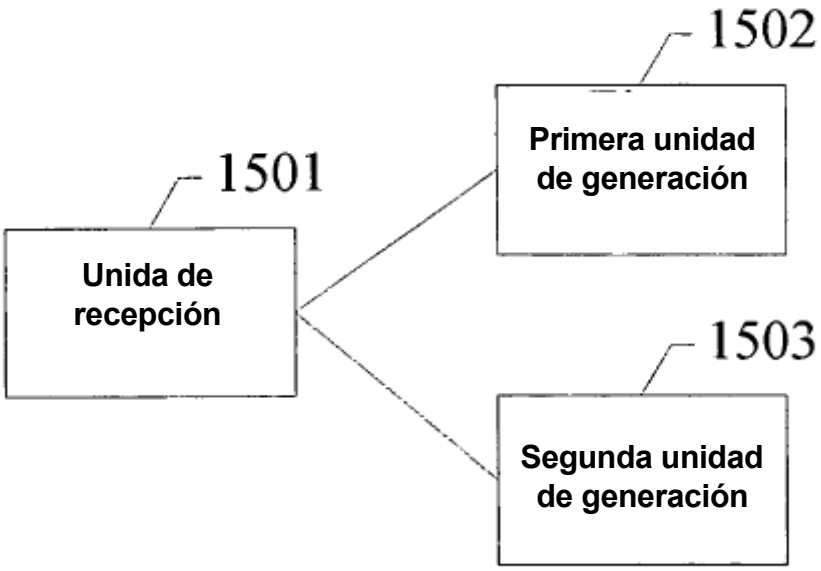
1502
Primera unidad de generación
1501
Unida de recepción
1503
Segunda unidad de generación

FIG. 15

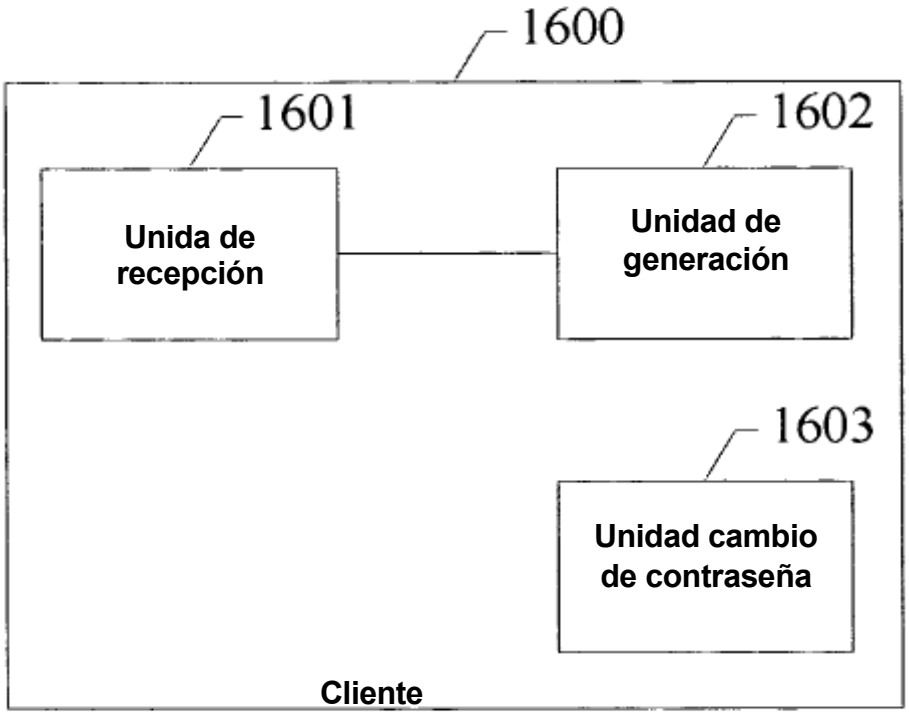
1600
1601
1602
Unida de recepción
Unidad de generación
1603
Unidad cambio de contraseña
Cliente

FIG. 16

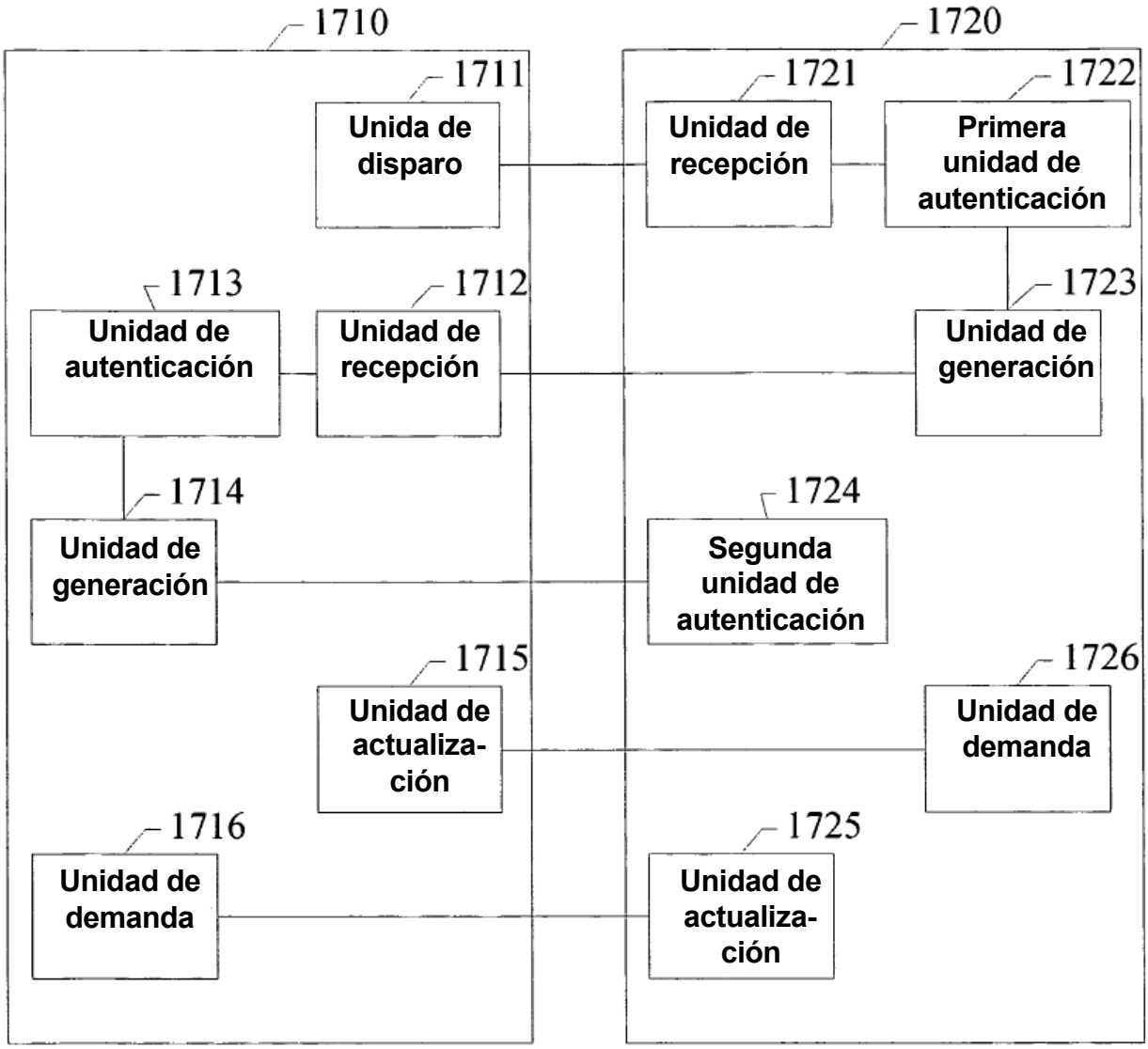
1710
1711
Unida de disparo
1713
Unidad de autenticación
1712
Unidad de recepción
1714
Unidad de generación
1715
Unidad de actualiza-ción
1716
Unidad de demanda
1720
1721
Unidad de recepción
1722
Primera unidad de autenticación
1723
Unidad de generación
1724
Segunda unidad de autenticación
1726
Unidad de demanda
1725
Unidad de actualiza-ción

FIG. 17