(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 814 275**

(51) Int. CI.:

H04L 9/30 (2006.01)
H04L 9/32 (2006.01)
H04L 29/06 (2006.01)
G06F 21/36 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **28.09.2016 PCT/US2016/054186**

(87) Fecha y número de publicación internacional: **22.06.2017 WO17105579**

(96) Fecha de presentación y número de la solicitud europea: **28.09.2016 E 16876204 (5)**

(97) Fecha y número de publicación de la concesión europea: **06.05.2020 EP 3369208**

(54) Título: **Sistema y método de autenticación y encriptación a prueba de intercepción**

(30) Prioridad:

28.10.2015 US 201514925769
03.11.2015 US 201514931613

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
26.03.2021

(73) Titular/es:

NI, MIN (100.0%)
1050 Creekdale Drive
Clarkston, Georgia 30021, US

(72) Inventor/es:

NI, MIN

(74) Agente/Representante:

TOMAS GIL, Tesifonte Enrique

DESCRIPCIÓN

Sistema y método de autenticación y encriptación a prueba de intercepción

ANTECEDENTES DE LA INVENCIÓN

1. Campo de la invención

[0001] La presente invención se refiere a sistemas y métodos de autenticación y cifrado y, más particularmente, a sistemas y métodos de autenticación y cifrado a prueba de intercepciones.

2. Antecedentes de la técnica relacionada

[0002] En la sociedad moderna, la vida cotidiana requiere el uso de una amplia variedad de dispositivos de información, como teléfonos móviles, PC, ordenadores portátiles y cajeros automáticos, por nombrar algunos. Los dispositivos de información pueden conservar los datos personales de los usuarios. Debido a la importancia de proteger estos datos personales, existen métodos para bloquear y desbloquear estos dispositivos de forma segura.

[0003] En la actualidad, el método más utilizado para bloquear y desbloquear estos dispositivos es un procedimiento de autenticación de desafío basado en contraseña, por el cual un dispositivo generalmente requiere que, antes de acceder a sus servicios, los usuarios introduzcan una identificación de usuario y una contraseña para el reconocimiento de identidad. Esto se conoce como inicio de sesión. Este proceso de inicio de sesión está diseñado para evitar que los datos personales de los usuarios sean robados o cambiados de manera fraudulenta.

[0004] Con el rápido aumento diario de la cobertura y la accesibilidad de la red, es más probable que los piratas informáticos intenten obtener las contraseñas de los usuarios para obtener acceso a su información privada. Además, los piratas informáticos se están volviendo cada vez más sofisticados para adivinar y descifrar las contraseñas de los usuarios. Por lo tanto, las contraseñas simples ya no brindan una protección adecuada contra las amenazas cibernéticas y el espionaje.

[0005] Ante esto, se han implementado diversos mecanismos para brindar una mejor protección. Por ejemplo, los usuarios deben crear una contraseña que cumpla con los requisitos de longitud, complejidad e imprevisibilidad de la contraseña, de modo que la fuerza de la contraseña sea, en teoría, suficiente para defenderse de los ataques de búsqueda por fuerza bruta y los ataques de diccionario. Además, los usuarios deben cambiar sus contraseñas con regularidad para invalidar las contraseñas antiguas, reduciendo así la posibilidad de que sus contraseñas sean descifradas. Estos mecanismos mejoran la seguridad hasta cierto punto y, por lo tanto, ayudan a los usuarios a proteger sus cuentas.

[0006] Sin embargo, cada organización puede tener un conjunto diferente de reglas de contraseña. Algunos requieren que la longitud de la contraseña sea de al menos 6 u 8 caracteres. Algunos requieren el uso de letras mayúsculas y minúsculas mezcladas, así como números. Algunos requieren al menos un carácter especial, pero otros no permiten caracteres especiales, por lo que cuando, crea que acaba de crear una contraseña estrella muy fuerte que puede usar en todos los lugares, habrá un siguiente lugar que tiene un conjunto diferente de requisitos que invalidará su contraseña estrella.

[0007] Como resultado de estas diferentes reglas de contraseñas, puede ser difícil, si no imposible, que los usuarios recuerden la multitud de contraseñas que han configurado con diferentes sitios/organizaciones. Por tanto, los usuarios normalmente almacenarán sus contraseñas, como en un archivo que se almacena en su dispositivo de información y/o en una aplicación de almacenamiento de contraseñas que se ejecuta en su dispositivo de información. Los piratas informáticos pueden atacar las contraseñas almacenadas, y si obtienen acceso al dispositivo en el que se almacenan las contraseñas, obtendrán acceso a todas las contraseñas y tendrán acceso a todas las cuentas/sitios protegidos por contraseña del usuario. Por lo tanto, implementar reglas estrictas para las contraseñas para evitar contraseñas que son demasiado débiles puede tener el efecto opuesto al esperado (un mayor riesgo de exponer más información).

[0008] En vista de estos problemas con las contraseñas tradicionales, se han desarrollado nuevos métodos para intentar resolver estos problemas. Estos métodos pueden incluir, entre otros, el uso de fotos, imágenes gráficas o diferentes formas y tonos para que a los piratas informáticos les resulte más difícil espiar o robar. Algunas técnicas incluso utilizan gestos y posicionamiento de información en determinadas ubicaciones de la pantalla de entrada para validar el acceso del usuario. Sin embargo, ninguno de estos métodos puede anular una cámara oculta que puede registrar cada movimiento de los usuarios cada vez que inician sesión en un dispositivo. Si un pirata informático puede reproducir todas las grabaciones y analizar cada movimiento de un usuario, el pirata informático eventualmente obtendrá acceso.

[0009] En la publicación internacional n.º WO2013/163285 se describe un método convencional para la autenticación de contraseñas de usuario.

[0010] Los principales problemas con los métodos de autenticación existentes son:

(1) Las contraseñas tradicionales y las preguntas de seguridad (el método más utilizado) no son a prueba de copia;
(2) Las imágenes gráficas y los métodos basados en fotografías pueden requerir que los usuarios carguen una imagen o un archivo de fotografía, y el sistema debe guardar y mantener las imágenes y/o fotografías. Esto aumenta la carga del usuario y del sistema, y si los piratas informáticos graban y reproducen el proceso de inicio de sesión, las imágenes aún se pueden reconocer;
(3) Los nuevos métodos de autenticación basados en gráficos, gestos y/o ubicación solo se pueden usar entre humanos y ordenadores y, por lo tanto, no se pueden usar de máquina a máquina.

[0011] Por tanto, existe la necesidad de sistemas y métodos de autenticación y cifrado que no presenten los problemas descritos anteriormente.

RESUMEN DE LA INVENCIÓN

[0012] Un objeto de la invención es resolver al menos los problemas y/o desventajas anteriores y proporcionar al menos las ventajas que se describen a continuación.

[0013] Según un aspecto de la presente invención, se proporciona un método como se define en la reivindicación 1 a continuación.

[0014] Según otro aspecto de la presente invención, se proporciona un sistema como se define en la reivindicación 8 a continuación.

[0015] Por tanto, un objeto de la presente invención es proporcionar un sistema y un método para la autenticación de un usuario.

[0016] Otro objeto de la presente invención es proporcionar un sistema y método para la autenticación de un usuario que intenta acceder a un dispositivo electrónico.

[0017] Otro objeto de la presente invención es proporcionar un sistema y un método para la autenticación de un usuario que solicita acceso a información almacenada electrónicamente.

[0018] Otro objeto de la presente invención es proporcionar un sistema y método para la autenticación de un usuario que solicita acceso a un dispositivo en una red.

[0019] Otro objeto de la presente invención es proporcionar un sistema y método para la autenticación de un usuario que utiliza códigos de acceso que contienen un número predeterminado de símbolos.

[0020] Otro objeto de la presente invención es proporcionar un sistema y método para la autenticación de un usuario que utiliza múltiples *tokens*, donde cada *token* es un grupo de al menos dos símbolos de un conjunto de símbolos usados para crear un código de acceso de usuario.

[0021] Otro objeto de la presente invención es proporcionar un sistema y método para cifrar y descifrar información electrónica.

[0022] Otro objeto de la presente invención es proporcionar un sistema y método para cifrar y descifrar información electrónica que utiliza un código de acceso que contiene un número predeterminado de símbolos para cifrar y descifrar la información electrónica.

[0023] Otro objeto de la presente invención es proporcionar un sistema y método para el cifrado y descifrado de información electrónica que utiliza un código de acceso que contiene un número predeterminado de símbolos en combinación con múltiples *tokens*, donde cada *token* es un grupo de al menos dos símbolos de un conjunto de símbolos utilizados para crear el código de acceso.

[0024] Para lograr al menos los objetos anteriores, en su totalidad o en parte, se proporciona un método de autenticación de un usuario usando un código de acceso predeterminado que comprende un número predeterminado de símbolos ("símbolos de código de acceso") seleccionados de un conjunto de símbolos, en el que cada uno de los símbolos de código de acceso predeterminados se caracterizan por una posición de pin predeterminada, que comprende presentar un conjunto de *tokens* al usuario, en el que el conjunto de *tokens* comprende al menos dos *tokens*, y en el que cada *token* en el conjunto de *tokens* comprende al menos dos símbolos que pertenecen al conjunto de símbolos, que requieren que el usuario seleccione un *token* del conjunto de *tokens* para cada posición de pin en el código de acceso predeterminado, y autenticar al usuario basándose en los *tokens* que el usuario seleccionó.

[0025] Para lograr al menos los objetos anteriores, en su totalidad o en parte, también se proporciona un sistema para autenticar a un usuario usando un código de acceso predeterminado que comprende un número predeterminado de símbolos ("símbolos de código de acceso") seleccionados de un conjunto de símbolos, en el que cada de los símbolos de código de acceso predeterminados se caracteriza por una posición de pin predeterminada, que comprende un procesador, una memoria accesible por el procesador y un módulo de autenticación/encriptación que comprende un conjunto de instrucciones legibles por ordenador almacenadas en la memoria que son ejecutables por el procesador para presentar un conjunto de *tokens* para el usuario, en el que el conjunto de *tokens* comprende al menos dos *tokens*, y donde cada *token* en el conjunto de *tokens* comprende al menos dos símbolos que pertenecen al conjunto de símbolos, requiere que el usuario seleccione un *token* del conjunto de *tokens* para cada posición de pin en el código de acceso predeterminado y autenticar al usuario basándose en los *tokens* que el usuario seleccionó.

[0026] Ventajas, objetos y características adicionales de la invención se expondrán en parte en la descripción que sigue y en parte resultarán evidentes para los expertos en la técnica al examinar lo siguiente o se pueden aprender de la práctica de la invención. Los objetos y ventajas de la invención pueden realizarse y lograrse como se señala particularmente en las reivindicaciones adjuntas.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0027] La invención se describirá en detalle con referencia a los siguientes dibujos, en los que los números de referencia iguales se refieren a elementos iguales, en los que:

La figura 1A es un diagrama de bloques que ilustra un sistema de autenticación/cifrado a prueba de intercepciones ejemplar que puede incorporarse en un dispositivo o servidor al que se accede mediante un sistema cliente, de acuerdo con una realización de la presente invención;
La figura 1B es un diagrama de bloques que ilustra el sistema de autenticación/cifrado a prueba de intercepciones incorporado en un dispositivo, de acuerdo con una realización de la presente invención;
La figura 1C es un diagrama de bloques que ilustra el sistema de autenticación/cifrado a prueba de intercepciones incorporado en un servidor al que accede un dispositivo cliente a través de una red, de acuerdo con una realización de la presente invención;
La Figura 1D es un diagrama esquemático de una implementación de *hardware* ejemplar del sistema de autenticación/cifrado a prueba de intercepciones, de acuerdo con una realización de la presente invención;
La figura 2A muestra ejemplos de símbolos agrupados en cuatro dimensiones, de acuerdo con una realización de la presente invención;
La figura 2B muestra ejemplos de símbolos agrupados en cinco dimensiones, de acuerdo con una realización de la presente invención;
La Figura 3 es un diagrama de flujo que ilustra un proceso ejemplar implementado por el módulo de autenticación/encriptación para permitir que un usuario cree un código de acceso, de acuerdo con una realización de la presente invención;
La figura 4 es un diagrama de flujo que ilustra un proceso ejemplar implementado por el módulo de autenticación/encriptación para autenticar a un usuario, de acuerdo con una realización de la presente invención;
La figura 5 es una tabla que enumera reglas de generación de *tokens* a modo de ejemplo utilizadas por el módulo de autenticación/cifrado, de acuerdo con una realización de la presente invención;
La figura 6 es una tabla que enumera reglas de selección de *tokens* de ejemplo utilizadas por el módulo de autenticación/cifrado, de acuerdo con una realización de la presente invención;
La figura 7 es una tabla que enumera las reglas de validación de *tokens* a modo de ejemplo utilizadas por el módulo de autenticación/cifrado, de acuerdo con una realización de la presente invención;
Las figuras 8A y 8B son capturas de pantalla de muestra del proceso de creación (registro) de identificación de usuario en la realización GATE_4, de acuerdo con una realización de la presente invención;
Las figuras 9A-9D son capturas de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_4, de acuerdo con una realización de la presente invención;
Las figuras 10A-10D son capturas de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_4 en formato de texto, de acuerdo con una realización de la presente invención;
Las figuras 11A y 11B son capturas de pantalla de muestra del proceso de creación (registro) de identificación de usuario en la realización GATE_5, de acuerdo con una realización de la presente invención;
Las figuras 12A-12D son capturas de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_5, de acuerdo con una realización de la presente invención;
Las figuras 13A-13D son capturas de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_5 en formato de texto, de acuerdo con una realización de la presente invención;
La figura 14 es una captura de pantalla de muestra de un proceso de cifrado de mensajes que utiliza la realización GATE_4 en la que un mensaje de texto sin formato se cifra con un código de acceso del emisor, de acuerdo con una realización de la presente invención;

Las figuras 15A y 15B son capturas de pantalla de muestra de un proceso de descifrado de mensajes utilizando la realización GATE_4 en la que tiene lugar un descifrado exitoso, de acuerdo con una realización de la presente invención;
Las figuras 16A y 16B son capturas de pantalla de muestra de un proceso de descifrado de mensajes que utiliza la realización GATE_4 en la que un mensaje de relleno cifrado se invalida en el lado del receptor, de acuerdo con una realización de la presente invención;
La figura 17 es una captura de pantalla de muestra de un proceso de descifrado de mensajes que utiliza la realización GATE_4 en la que el descifrado falla debido a un código de acceso del receptor que es diferente al código de acceso del emisor, de acuerdo con una realización de la presente invención;
La figura 18 es una captura de pantalla de muestra de un proceso de cifrado de mensajes que utiliza la realización GATE_5 en la que un mensaje de texto sin formato se cifra con un código de acceso del emisor, de acuerdo con una realización de la presente invención;
Las figuras 19A y 19B son capturas de pantalla de muestra de un proceso de descifrado de mensajes usando la realización GATE_5 en la que tiene lugar un descifrado exitoso, de acuerdo con una realización de la presente invención;
Las figuras 20A y 20B son capturas de pantalla de muestra de un proceso de descifrado de mensajes que utiliza la realización GATE_5 en la que un mensaje de relleno cifrado se invalida en el lado del receptor, de acuerdo con una realización de la presente invención; y
La figura 21 es una captura de pantalla de muestra de un proceso de descifrado de mensajes que utiliza la realización GATE_5 en la que el descifrado falla debido a un código de acceso del receptor que es diferente al código de acceso del emisor, de acuerdo con una realización de la presente invención.

DESCRIPCIÓN DETALLADA DE REALIZACIONES PREFERIDAS

[0028] La presente invención proporciona un mecanismo novedoso de autenticación y encriptación a prueba de intercepciones que se implementa con un aparato especialmente programado. Para la autenticación, la presente invención utiliza "códigos de acceso" que se forman con símbolos que forman parte del sistema operativo de un dispositivo. Como ejemplo, un código de acceso puede verse tener este aspecto:

ⓘ ♥ 2 ✉

[0029] Para un usuario, el código de acceso anterior puede significar "Me encanta enviar correos electrónicos", que es fácil de recordar pero difícil de conocer para cualquier otra persona. Cada símbolo del código de acceso se denominará un "pin" con una posición de pin correspondiente en relación con los otros pines. En el ejemplo anterior, el símbolo "ⓘ" está en la primera posición del pin, el símbolo " ♥ " está en la posición del segundo pin, el símbolo "2" está en la posición del tercer pin y el símbolo " ✉ " está en la posición del cuarto pin.

[0030] La presente invención es preferiblemente "dentro del sistema" porque utiliza un conjunto selecto de símbolos que son preferiblemente parte del sistema operativo del dispositivo y, por lo tanto, no requiere que los usuarios carguen ninguna foto o imagen. Los símbolos utilizados para crear los códigos de acceso o el mensaje cifrado son de diferentes tipos que se agrupan preferiblemente en dos o más grupos que se denominarán en el presente documento "dimensiones". Esto brinda a los usuarios una mayor variedad de formas de expresarse al crear sus contraseñas.

[0031] La presente invención proporciona un mecanismo novedoso de autenticación y cifrado a prueba de intercepciones al "ocultar" los símbolos utilizados para los pines que componen el código de acceso de un usuario entre otros múltiples símbolos que no forman parte del código de acceso del usuario. Es como, en esencia, esconder una aguja en un pajar. Específicamente, la presente invención utiliza lo que se denominará aquí como "*tokens*". Un *token* es un grupo de al menos dos símbolos. Se presentan varios *tokens* (un "conjunto de *tokens*") a un usuario, con algunos o todos los pines preseleccionados de un usuario insertados aleatoriamente en algunos o en todos los *tokens*. Específicamente, cada pin (representado por un símbolo preseleccionado) en el código de acceso de un usuario puede incluirse en uno de los *tokens* que se le presentan al usuario. El usuario elige los *tokens* que contienen los pines individuales que componen el código de acceso, de modo que la posición del pin de cada *token* seleccionado corresponde a la posición del pin del código de acceso que contiene. Debido a que cada *token* elegido contiene no solo uno de los pines preseleccionados en el código de acceso del usuario, sino también otros símbolos generados aleatoriamente que no son uno de los pines preseleccionados en el código de acceso del usuario, alguien que observe qué *tokens* ha elegido el usuario no puede determinar cuál es la contraseña real del usuario.

[0032] Cada vez que se le pide al usuario que proporcione el código de acceso, se genera otro conjunto de *tokens* generados aleatoriamente con los pines individuales preseleccionados en el código de acceso del usuario distribuidos al azar entre los múltiples *tokens*. Por lo tanto, el usuario elegirá un conjunto diferente de *tokens* cada vez que se le solicite que proporcione el código de acceso, evitando así que un observador determine el código de acceso del usuario en función de la elección de los *tokens*.

[0033] Como ejemplo ilustrativo, se pueden utilizar 4 grupos de símbolos (4 dimensiones), cada dimensión contiene 36 símbolos. Durante un proceso de inicio de sesión, se muestran 36 *tokens* para el usuario, y cada *token* contiene un símbolo de cada una de las cuatro dimensiones de los símbolos. Un símbolo determinado se muestra preferiblemente en un solo *token* (es decir, si el símbolo aparece en un *token*, no vuelve a aparecer en otro *token*). Dado que, en este ejemplo, se muestran 36 *tokens*, se muestra cada símbolo en cada una de las cuatro dimensiones (un símbolo de cada una de las dimensiones en cada uno de los 36 *tokens*).

[0034] Si el número de pines (representados por símbolos) en el código de acceso de un usuario está representado por la variable "N", entonces el usuario deberá elegir N *tokens* (los *tokens* que contienen los pines individuales que componen el código de acceso, de modo que la posición del pin de cada *token* seleccionado corresponde a la posición del pin del código de acceso que contiene).

[0035] Como se ha mencionado anteriormente, la presente invención reducirá la probabilidad de "espiar e interceptar" el código de acceso del usuario porque el usuario introduce un *token* que contiene 4 símbolos, incluido uno de los pines en el código de acceso, para cada pin en el código de acceso. Por lo tanto, incluso si un pirata informático observa la entrada de un usuario mirando la pantalla de inicio de sesión, o un pirata informático intercepta el tráfico de la red, el pirata informático no podría determinar cuál de los 4 símbolos en cada *token* corresponde a cada uno de los pines preseleccionados que componen el código de acceso del usuario. En consecuencia, si el pirata informático intentara iniciar sesión en la cuenta del usuario, se le presentaría otro conjunto de *tokens* generados aleatoriamente, algunos de los cuales contienen los pines preseleccionados del usuario que componen el código de acceso del usuario, y el pirata informático no sabría qué *tokens* seleccionar.

[0036] Sin embargo, si el pirata informático observa el proceso de inicio de sesión del usuario suficientes veces, el pirata informático puede comparar todas las sesiones de inicio de sesión registradas para averiguar qué es cada pin en el código de acceso, porque cada pin seguramente aparecerá en un *token* que introduzca el usuario, y si el pirata informático compara todos los primeros *tokens* de diferentes sesiones de inicio de sesión, el pirata informático eventualmente determinará el primer pin, y si el pirata informático compara todos los segundos *tokens* de todas las sesiones, eventualmente determinará el segundo pin, etc.

[0037] Por lo tanto, para evitar que un pirata informático descubra los pines en el código de acceso con el tiempo, el número de *tokens* generados aleatoriamente que se presentan al usuario es preferiblemente menor que el número de símbolos en cada dimensión. Por ejemplo, si cada una de las dimensiones contiene 36 símbolos, se puede optar por presentar solo 16 *tokens* al usuario. El resultado de esto es que no hay garantía de que un pin de usuario aparezca en un *token*. En esta realización, si el usuario está intentando iniciar sesión y uno o más pines en el código de acceso del usuario no están presentes en ninguno de los *tokens*, entonces el usuario selecciona cualquier *token* como un *token* "comodín" para los pines que no están presentes en ningún de los *tokens* (de modo que la posición del pin del *token* comodín seleccionado corresponda a la posición del pin del código de acceso que falta). Esto hace que la suposición de un pirata informático sea mucho más difícil, porque puede haber un *token* elegido al azar en lugar de uno de los pines del usuario que en realidad no contiene el pin.

[0038] Otro beneficio de usar menos *tokens* que el número de símbolos en una o más dimensiones (por ejemplo, usar 16 *tokens* cuando el número de símbolos en una o más dimensiones es 36) es que facilita a los usuarios averiguar rápidamente si los pines preseleccionados están en los *tokens* o no, y la pantalla parece más simple para el usuario.

[0039] La presente invención preferiblemente usa símbolos que son parte del sistema operativo del dispositivo que incorpora el presente sistema. Por lo tanto, no es necesario que el usuario cargue gráficos o fotos especiales en el sistema y, por lo tanto, reduce la carga para el usuario y el sistema para almacenar y mantener esas propiedades.

[0040] El sistema y los métodos de la presente invención pueden usarse no solo para autenticar usuarios, sino también para autenticar múltiples informaciones y, por lo tanto, pueden usarse para cifrar mensajes.

[0041] La figura 1A es un diagrama de bloques que ilustra un sistema 100 de autenticación/cifrado a prueba de intercepciones ejemplar que puede incorporarse en un dispositivo o servidor al que se accede mediante un sistema cliente, de acuerdo con una realización preferida de la presente invención. El sistema 100 incluye un módulo 110 de autenticación/encriptación que proporciona funcionalidad de autenticación y/o encriptación a prueba de intercepciones. El sistema 100 se puede conectar opcionalmente a una red 130.

[0042] La figura 1B es un diagrama de bloques que ilustra el sistema de autenticación/cifrado a prueba de intercepciones incorporado en un dispositivo 150, de acuerdo con una realización preferida de la presente invención. El dispositivo 150 incluye preferiblemente un módulo 120 de interfaz de usuario. El módulo 110 de autenticación/encriptación proporciona una autenticación segura de la contraseña de un usuario y/o encriptación de un mensaje, como se explicará con más detalle a continuación. El módulo 110 de autenticación/encriptación puede conectarse opcionalmente a una red 130.

[0043] El módulo de interfaz de usuario 120 se puede implementar con cualquier tipo de interfaz de usuario conocido en la técnica, como, por ejemplo, una interfaz gráfica de usuario, una interfaz basada en web y similares. En general, el módulo 120 de interfaz de usuario puede implementarse con cualquier tipo de interfaz que permita a un usuario interactuar con el módulo 110 de autenticación.

[0044] La figura 1C es un diagrama de bloques que ilustra el sistema de autenticación/cifrado a prueba de intercepciones incorporado en un servidor 160 al que accede un dispositivo cliente 170 a través de una red 130, de acuerdo con una realización preferida de la presente invención. El dispositivo cliente 170 es cualquier tipo de ordenador o dispositivo que pueda acceder al servidor 160 a través de la red 130, y preferiblemente incluye un módulo 120 de interfaz de usuario que permite al usuario interactuar con el dispositivo 170 cliente. El módulo 110 de autenticación/cifrado proporciona seguridad autenticación de la contraseña de un usuario y/o cifrado de un mensaje, como se explicará con más detalle a continuación.

[0045] Los enlaces de comunicación 140 se utilizan para las comunicaciones entre el módulo de autenticación/cifrado 110, el módulo de interfaz de usuario 120, la red 130 y el dispositivo cliente 170. Los enlaces de comunicación 140 pueden ser enlaces cableados, enlaces inalámbricos, enlaces inductivos inalámbricos, enlaces capacitivos o cualquier otro mecanismo conocido en la técnica para conectar componentes electrónicos. Un enlace de cableado podría implementarse adecuadamente con un bus como, por ejemplo, un bus de Arquitectura estándar de la industria (ISA), un bus de Arquitectura de microcanal (MCA), un bus ISA mejorado, un bus local de la Asociación de estándares electrónicos y de video (VESA) o un bus de interconexión de componentes periféricos (PCI).

[0046] Los ejemplos de enlaces inalámbricos incluyen, entre otros, un enlace WAP (Protocolo de aplicaciones inalámbricas), un enlace GPRS (Servicio general de radio por paquetes), un enlace GSM (Sistema global para comunicaciones móviles), CDMA (Acceso múltiple por división de código) o Enlace TDMA (Acceso múltiple por división de tiempo), como un canal de teléfono celular, un enlace GPS (Sistema de posicionamiento global), CDPD (datos de paquetes digitales celulares), un dispositivo de tipo de mensajería dúplex RIM (Research in Motion, Limited), un enlace radio Bluetooth o un enlace de radiofrecuencia basado en IEEE 802.11(WiFi).

[0047] La red 130 puede ser una red por cable o inalámbrica, y puede incluir o interactuar con uno o más de, por ejemplo, Internet, una intranet, una PAN (red de área personal), una LAN (red de área local), una WAN (Red de área amplia) o MAN (red de área metropolitana), una red de área de almacenamiento (SAN), una conexión de retransmisión de tramas, una conexión de red inteligente avanzada (AIN), una conexión de red óptica síncrona (SONET), una línea T1, T3, E1 o E3 digital, una conexión de servicio de datos digitales (DDS), una conexión DSL (línea de abonado digital), una conexión Ethernet, una línea ISDN (red digital de servicios integrados), un puerto de acceso telefónico como V.90, una conexión de módem analógico V.34bis, un módem por cable, una conexión ATM (modo de transferencia asincrónica), una conexión FDDI (interfaz de datos distribuida por fibra) o CDDI (interfaz de datos distribuida por cobre).

[0048] El módulo de interfaz de usuario 120 se puede implementar con cualquier tipo de interfaz de usuario conocido en la técnica, como, por ejemplo, una interfaz gráfica de usuario, una interfaz basada en web y similares. En general, el módulo 120 de interfaz de usuario puede implementarse con cualquier tipo de interfaz que permita a un usuario interactuar con el módulo 110 de autenticación/cifrado.

[0049] El término "módulo", como se usa en este documento, significa un dispositivo, componente o disposición del mundo real de componentes implementados usando *hardware*, que puede incluir un circuito integrado específico de aplicación (ASIC) o una matriz de puertas programables en campo (FPGA), por ejemplo, o un sistema de microprocesador y un conjunto de instrucciones para implementar la funcionalidad del módulo, que (al ejecutarse) transforman el sistema de microprocesador en un dispositivo especial para realizar las funciones del módulo.

[0050] Un módulo también se puede implementar como una combinación de *hardware* solo y *hardware* controlado por *software*, con ciertas funciones facilitadas por el *hardware* solo, y otras funciones facilitadas por una combinación de *hardware* y *software*. En ciertas implementaciones, al menos una parte, y en algunos casos, la totalidad, de un módulo se puede ejecutar en el procesador(es) de un ordenador o dispositivo (como, por ejemplo, el servidor 160 y el dispositivo cliente 170) que ejecuta un sistema operativo, programas del sistema y programas de aplicación, mientras que también implementa el módulo mediante el uso de múltiples tareas, subprocesos múltiples, procesamiento distribuido (por ejemplo, nube) u otras técnicas similares. Ejemplos de dicho ordenador o dispositivo incluyen, entre otros, un ordenador personal (por ejemplo, un ordenador de escritorio o un ordenador portátil), un servidor, un cajero automático (ATM), una terminal de punto de venta, un aparato y un dispositivo informático móvil, como un teléfono inteligente, una tableta o un asistente digital personal (PDA). Además, el servidor 160 es adecuadamente cualquier tipo de servidor, tal como un servidor Windows, un servidor Linux, un servidor Unix o similar.

[0051] La figura 1D es un diagrama esquemático de una implementación de *hardware* ejemplar del sistema 100 de

autenticación/encriptación a prueba de intercepciones, de acuerdo con una realización de la presente invención. En la realización de la figura 1D, el módulo 110 de autenticación/cifrado es implementado por la CPU 118 y la memoria 112.

[0052] La CPU 118 accede al código 114 del sistema operativo y otro código 116 de programa almacenado en la memoria 112 para su ejecución. El código de programa 116 que implementa la funcionalidad del módulo de autenticación/encriptación 110 se almacena en la memoria 112, o en un dispositivo de almacenamiento externo (no mostrado), para el acceso y ejecución por la CPU 118.

[0053] La memoria 112 se puede implementar mediante, por ejemplo, memoria de acceso aleatorio (RAM), memoria caché, medios de almacenamiento extraíbles/no extraíbles, medios de almacenamiento volátiles/no volátiles, como una unidad de disco duro no extraíble no volátil, una unidad de disquete no volátil extraíble, una unidad de disco óptico (como CD-ROM, DVD-ROM o cualquier otro medio de almacenamiento óptico), una unidad *flash* USB y una tarjeta de memoria.

[0054] A continuación se describirá la funcionalidad del módulo 110 de autenticación/encriptación. El módulo 110 de autenticación/encriptación proporciona autenticación de contraseña/desafío, en respuesta a una solicitud de inicio de sesión de usuario, mostrando al usuario (a través del módulo 120 de interfaz de usuario) múltiples *tokens* con pines preseleccionados de un usuario insertados aleatoriamente en los *tokens*. Como se ha mencionado anteriormente, cada pin (representado por un símbolo preseleccionado) en el código de acceso de un usuario puede incluirse en uno de los *tokens* que se presentan al usuario. Al menos un pin está presente en uno de los *tokens*. El usuario elige los *tokens* que contienen los pines individuales que componen el código de acceso, de modo que la posición del pin de cada *token* seleccionado corresponde a la posición del pin del código de acceso que contiene. Debido a que cada *token* elegido contiene símbolos generados aleatoriamente que no son uno de los pines preseleccionados en el código de acceso del usuario, así como posiblemente uno de los pines preseleccionados en el código de acceso del usuario, alguien que observe qué *tokens* ha elegido el usuario no puede determinar cuál es la contraseña real del usuario.

[0055] A continuación se describirán dos realizaciones ilustrativas de la funcionalidad del módulo de autenticación/cifrado, y se las denominará en el presente documento Entrada tabular de acceso gráfico_4 ("GATE_4") y Entrada tabular de acceso gráfico_5 ("GATE_5"). Como se muestra en la Figura 2A, la realización GATE_4 usa 4 dimensiones de símbolos, con 36 símbolos incluidos en cada dimensión. Como se muestra en la Figura 2B, la realización GATE_5 usa 5 dimensiones de símbolos, con 26 símbolos incluidos en cada dimensión. Los símbolos mostrados en las Figs. 2A y 2B son ejemplos de los tipos de símbolos que se pueden usar, y debe apreciarse que se puede usar cualquier otro tipo de símbolos sin dejar de estar dentro del alcance de la presente invención.

[0056] Los símbolos mostrados en las Figs. 2A y 2B están generalmente disponibles en los sistemas operativos de ordenador modernos y no requieren ningún proceso especial para crear o cargar/guardar en un sistema existente que incorpore la presente invención. Son un subconjunto del sistema Unicode estándar que se encuentra en la mayoría de los sistemas operativos de ordenador. Cada símbolo tiene un ID Unicode, los caracteres familiares: a, b, ... z, 0, 1, 9, @, +, <, % forman todos parte del sistema Unicode. Por ejemplo:

Unicode \u0062 es para el carácter: b
Unicode \u2206 es para el carácter: Δ
Unicode \u0040 es para el carácter: @

Los símbolos mostrados en las realizaciones de las Figs. 2A y 2B se incluyeron porque son diversos, distintivos y fáciles de recordar.

[0057] La figura 3 es un diagrama de flujo que ilustra un proceso ejemplar implementado por el módulo 110 de autenticación/encriptación para permitir que un usuario cree un código de acceso, de acuerdo con una realización de la presente invención. El proceso comienza en el paso 310, donde se le pide al usuario que introduzca un ID de usuario deseado. En el paso 320, el módulo 110 de autenticación/cifrado determina si el ID de usuario ya existe en la memoria 112. Si existe el ID de usuario, el proceso continúa hasta el paso 330. De lo contrario, el proceso continúa hasta el paso 340.

[0058] En el paso 330, se le pregunta al usuario si desea sobrescribir el código de acceso existente asociado con la identificación del usuario. Si el usuario indica "no", el proceso vuelve al paso 310. Si el usuario indica "sí", el proceso pasa al paso 340.

[0059] En el paso 340, se muestran al usuario los símbolos disponibles para que el usuario elija para cada pin en su código de acceso. Luego, en el paso 350, se le pide al usuario que seleccione uno de los símbolos mostrados como uno de los pines para su código de acceso. En el paso 360, el proceso determina si el usuario ha solicitado que los pines seleccionados actualmente se guarden como código de acceso. Esto puede implementarse, por ejemplo, mostrando un icono que el usuario puede seleccionar cuando el usuario está listo para guardar la

contraseña. Si el usuario no ha indicado que se debe guardar el código de acceso, el proceso vuelve al paso 350, donde el usuario selecciona otro símbolo para otro pin en el código de acceso. Si el usuario indica, en el paso 360, que se debe guardar el código de acceso, entonces el proceso pasa al paso 370.

[0060] En el paso 370, se determina si el código de acceso seleccionado cumple con un requisito de longitud predeterminado (es decir, un número mínimo predeterminado de pines). Si el código de acceso seleccionado lo cumple, entonces el código de acceso se guarda en el paso 380. Si el código de acceso no lo cumple, el proceso vuelve al paso 350, en el que se solicita al usuario que elija un símbolo para un pin adicional.

[0061] La figura 4 es un diagrama de flujo que ilustra un proceso ejemplar implementado por el módulo 110 de autenticación/encriptación para autenticar a un usuario, de acuerdo con una realización de la presente invención. El proceso comienza en el paso 410, donde se presenta al usuario una pantalla de inicio de sesión en la que se solicita al usuario que introduzca una identificación de usuario. A continuación, el proceso pasa al paso 420, donde el módulo 110 de autenticación/cifrado determina si existe la identificación de usuario introducida por el usuario. Si es así, el proceso pasa al paso 430. Si no es así, el proceso vuelve al paso 410, donde se solicita al usuario que introduzca otro ID de usuario.

[0062] En el paso 430, el módulo de autenticación/encriptación genera un número predeterminado de *tokens* basado en el número de pines usados para el código de acceso del usuario. En la realización descrita en la figura 4, se generan 16 *tokens* y se muestran preferiblemente al usuario en una tabla de *tokens* de 4 x 4, como se describirá con más detalle a continuación. Los *tokens* se generan en base a las reglas de generación de *tokens* que se muestran en la Fig. 5. En los ejemplos, hay 16 *tokens* en la tabla, pero en realidad, podría ser cualquier número de *tokens* mayor que el número de pines del usuario, y se presenta en 3 x 4, 2 x 5 o cualquier otra combinación, 4 x 4 es solo una forma preferida de mostrarlos.

[0063] A continuación, el proceso pasa al paso 440, donde el usuario selecciona los *tokens* que contienen los pines en su código de acceso, en el orden en que aparecen los pines en el código de acceso, de acuerdo con las reglas de selección de *tokens* que se muestran en la Figura 6. En el paso 450, el módulo 110 de autenticación/cifrado determina si los *tokens* seleccionados por el usuario siguen las reglas de selección de *tokens* mostradas en la figura 6. Si se siguen las reglas de selección de *tokens*, el proceso pasa al paso 460, donde el usuario está autenticado y se le permite el acceso. Si no se siguen las reglas de selección de *tokens*, el proceso pasa al paso 470, donde el usuario no está autenticado y se le niega el acceso.

[0064] Como se ha mencionado anteriormente, la Fig. 5 es una tabla que enumera las reglas de generación de *tokens* a modo de ejemplo utilizadas por el módulo 110 de autenticación/cifrado, de acuerdo con una realización de la presente invención. Al menos uno de los pines en el código de acceso del usuario estará en al menos uno de los 16 *tokens*. A veces, todos los pines del usuario se encontrarán en los *tokens*, y la mayoría de las veces entre 1 y N (siendo N la longitud del código de acceso del usuario con N pines), los pines preseleccionados por el usuario estarán en los 16 *tokens*.

[0065] En la realización mostrada en la Fig. 5, se generan 16 *tokens*, no 36 (en el caso de la realización GATE_4) y no 26 (en el caso de la realización GATE_5). Por lo tanto, en el caso de la realización GATE_4, solo 16/36 = 44 % de las veces puede aparecer uno de los pines en el código de acceso del usuario en uno de los 16 *tokens*. En el caso de la realización GATE_5, solo 16/26 = 62 % de las veces aparecerá uno de los pines en el código de acceso del usuario en uno de los 16 *tokens*. Existe la posibilidad de que todos los pines en el código de acceso de un usuario aparezcan en la tabla de *tokens*, y se garantiza que al menos un pin de usuario estará presente en la tabla de *tokens*. La mayoría de las veces, faltarán algunos de los pines en el código de acceso del usuario y algunos aparecerán en los *tokens*. En realizaciones alternativas, las reglas podrían modificarse de modo que al menos 2 o 3 de los pines en el código de acceso de un usuario estén presentes en la tabla de *tokens*.

[0066] Es esta incertidumbre la que hace que la presente invención sea eficaz. Si el proceso de inicio de sesión es visto o se intercepta, lo único seguro para el pirata informático es la longitud del código de acceso, porque si el usuario introduce más o menos *tokens* que la longitud del código de acceso, el inicio de sesión fallará. Lo único que conducirá, pero no garantizará, un inicio de sesión exitoso es introducir la misma cantidad de *tokens* que pines en la contraseña del usuario.

[0067] Sin embargo, incluso si un pirata informático se entera de la longitud del código de acceso, el pirata informático no podrá determinar la identidad de los pines individuales. Esto se debe a que, aunque es posible que un pin no aparezca necesariamente en uno de los 16 *tokens*, el usuario puede iniciar sesión correctamente. Esto se debe a que, como se indica en las reglas de selección de *tokens* de la Fig. 6, el usuario puede elegir un *token* aleatoria para un pin que no está presente en uno de los *tokens* que se presentan.

[0068] Además, incluso si todos los pines del código de acceso del usuario aparecen en los 16 *tokens*, el pirata informático aún no podrá saber qué símbolo en cada *token* es un pin preseleccionado, porque hay 4 símbolos en la realización GATE_4, y 5 símbolos en la realización GATE_5. Esta incertidumbre hace que este sistema y método de la presente invención sea a prueba de intercepciones y visualizaciones.

[0069] Como se muestra en las reglas de selección de *tokens* enumeradas en la Fig.6, las reglas para seleccionar un *token* válido se pueden resumir de la siguiente manera:

- Un usuario debe seleccionar N *tokens* de la tabla de *tokens*, correspondientes a los N pines en el código de acceso del usuario. Por lo tanto, si el código de acceso del usuario tiene 4 pines, el usuario selecciona 4 *tokens*. Del mismo modo, si el código de acceso del usuario tiene 6 pines, el usuario debe seleccionar 6 *tokens*.
- Si el pin de un usuario aparece en uno de los 16 *tokens*, el usuario debe seleccionar ese *token* para introducir el pin.
- Si uno de los pines en el código de acceso del usuario no está presente en ninguno de los 16 *tokens*, el usuario debe seleccionar cualquiera de los 16 *tokens* para ese pin (en lo sucesivo denominado "*token* comodín").

[0070] Como se ha mencionado anteriormente, la Fig. 7 es una tabla que enumera reglas de validación de *token* ejemplares utilizadas por el módulo 110 de autenticación/cifrado, de acuerdo con una realización de la presente invención. Estas reglas son para validar los *tokens* que el usuario seleccionó en el proceso de inicio de sesión. Por ejemplo, las reglas determinan si el número de *tokens* introducidos por el usuario es igual a la longitud del código de acceso del usuario. De lo contrario, el inicio de sesión del usuario fallará. Si es así, entonces las reglas requieren verificar cada pin en el código de acceso del usuario para ver si está en uno de los 16 *tokens*. Si uno de los pines en el código de acceso del usuario no está presente en uno de los *tokens*, el usuario debe seleccionar un *token* aleatorio. Si aparece un pin en el código de acceso del usuario en uno de los *tokens*, el usuario debe seleccionar ese *token*.

[0071] La figura 8A es una captura de pantalla de muestra de una pantalla de registro vacía para introducir una identificación de usuario, según el paso 310 de la figura 3, de acuerdo con una realización de la presente invención. La figura muestra un ejemplo de cómo puede aparecer la pantalla de registro de la realización GATE_4 antes de que un usuario introduzca una identificación de usuario.

[0072] La figura 8B es una captura de pantalla de muestra de una trama de ejecución del proceso de registro para crear una identificación de usuario presentada por el sistema 100, de acuerdo con una realización de la presente invención. Muestra un ejemplo de cómo puede verse la pantalla de realización de GATE_4 a medida que el usuario pasa por el proceso de registro (crear identificación de usuario) de la Fig.3 de la siguiente manera:

- El usuario introduce una nueva identificación de usuario: "admin" (G4 206), luego hace clic en el botón "Verificar disponibilidad" (G4 208). El sistema comprueba si el ID "admin" ya está en su memoria (paso 320 de la figura 3). Si es así, mostrará un cuadro de diálogo (no se muestra) que pregunta: "La identificación de usuario ya existe, ¿desea sobrescribir el código de acceso existente?" Si el usuario no desea sobrescribir la contraseña anterior, el proceso cerrará el cuadro de diálogo y esperará a que el usuario introduzca otra identificación de usuario. Si el ID de usuario "admin" no existe, o si existe, pero el usuario desea sobrescribir el código de acceso existente, el sistema habilitará los botones en G4 212, G4 222, G4 232 y G4 242, que tienen cada uno 36 símbolos de una dimensión predefinida, como se muestra en la Fig. 2A. Por ejemplo, G4 212 incluye los 36 números de la 1ª dimensión, y esos símbolos se mostrarán en un *token* en la posición "[1]" (superior izquierda) como se muestra en G4 210 (los números son del 1 al 36). G4 222 tiene 36 símbolos de "Ⓐ" a "?" y aparecerán en cualquier *token* en la posición "[2]" (G4 220: arriba a la derecha). G4 232 tiene 36 símbolos de "○" a "° F", que se mostrarán en la posición "[3]" (G4 230: abajo a la izquierda) en un *token*, y G4 242 muestra 36 símbolos de "+" a "₪" y aparecerán en cualquier *token* en la posición " [4] "(G4 240: abajo a la derecha). En esta etapa del proceso, el sistema habilitará los botones anteriores en G4 212, G4 222, G4 232 y G4 242, para que el usuario pueda hacer clic en cualquiera de ellos. A modo de comparación, en la Fig. 8A esos botones no están habilitados y se ven pálidos, porque el usuario aún no ha introducido una identificación de usuario. Sin una identificación de usuario, el usuario no está permitido crear un código de acceso.
- El usuario hace clic en "①" entre los símbolos en G4 222 para seleccionar el primer pin, y aparece en G4 250 (paso 350 de la Fig. 3).
- El usuario hace clic en " ♥ " entre los símbolos en G4 232 para seleccionar el segundo pin, y aparece en G4 252 (paso 350 de la Fig. 3).
- El usuario hace clic en "2" entre los símbolos en G4 212 para seleccionar el tercer pin, y aparece en G4 254 (paso 350 de la Fig. 3).
- El usuario hace clic en " ✉ " entre los símbolos en G4 242 para seleccionar el cuarto pin, y aparece en G4 256 (paso 350 de la Fig. 3).
- En este ejemplo, el usuario eligió tener 4 pines en el código de acceso, por lo que la longitud del código de acceso es 4.
- En este ejemplo, las posiciones G4 258 y G4 260 se dejan en blanco.

- A continuación, el usuario finaliza el proceso de creación (registro) de la identificación de usuario haciendo clic en el botón "Guardar" - G4 270 (paso 370 de la figura 3). El sistema guardará el código de acceso "① ♥ 2 ☒ " con la identificación de usuario "admin" en la memoria (paso 380 de la figura 3).

[0073] Como se ha mencionado anteriormente, las opciones dimensionales no se limitan a los símbolos mostrados en el ejemplo de la Fig. 8B. Las opciones dimensionales pueden incluir cualquier otro símbolo. Hay 4 pines de usuario en la realización ejemplar descrita anteriormente, sin embargo, se puede usar cualquier número de pines sin dejar de estar dentro del alcance de la presente invención. Si el número de pines es demasiado bajo, el código de acceso será demasiado vulnerable. Si el número de pines es demasiado alto, es posible que el usuario no recuerde el código de acceso. Por consiguiente, una longitud preferida está entre 4 y 6 pines.

[0074] La figura 9A es una captura de pantalla de muestra de un marco de ejecución de la pantalla de inicio de sesión presentada por el sistema 100, de acuerdo con una realización de la presente invención, antes de que el usuario introduzca cualquier información. La figura 9A se utiliza como comparación con la figura 9B.

[0075] La figura 9B es una captura de pantalla de muestra de una trama de ejecución del proceso de inicio de sesión ejecutado por el sistema 100, de acuerdo con una realización de la presente invención. Muestra un ejemplo de cómo puede verse la pantalla de realización de GATE_4 a medida que el usuario pasa por el proceso de inicio de sesión de la Fig.4 de la siguiente manera:

- El usuario introduce una identificación de usuario: "admin" (G4 306) (paso 410 de la Fig. 4).
- El usuario hace clic en el botón "Entrar" (G4 309). El sistema verifica si el ID "admin" ya está en su memoria (paso 420 de la Fig. 4), si no existe, mostrará un mensaje (no mostrado) mostrando "El ID de usuario no existe, introduzca una identificación de usuario válida ". Si existe, el sistema mostrará una tabla de 4 x 4 (G4 320). Para describir mejor la tabla, las filas se marcan preferiblemente de arriba a abajo como sigue: A, B, C, D. Las columnas también se marcan preferiblemente de izquierda a derecha como sigue: 1, 2, 3, 4. Los *tokens* en esta tabla se genera de acuerdo con las reglas descritas en la Fig.5.
- Como sabemos por la Fig. 8B que el código de acceso asociado con la identificación de usuario "admin" es: "① ♥ 2 ☒ ", el usuario debe comenzar revisando los 16 *tokens* de la tabla para encontrar el primer pin:" ① ". Debido a que el símbolo ① pertenece a la 2ª dimensión, en este ejemplo aparece en la posición superior derecha de cualquier *token*, por lo que el usuario solo necesita escanear la parte superior derecha de cada *token* para ver si ① existe. En este ejemplo, está en el *token* D2. Esta captura de pantalla se tomó en modo de demostración, y en el modo de demostración el programa resalta el símbolo correspondiente para el usuario para comprender mejor el proceso. En tiempo real, esto no necesita ser resaltado. En esta ilustración, el ① en D2 está resaltado. Dado que se encuentra en el *token* D2, el usuario debe hacer clic en este *token* de acuerdo con las reglas descrito en la Fig.6.
- Después de que el usuario hace clic en D2, el *token* se copia en la primera posición del código de acceso (G4 350) (paso 440 de la figura 4).
- El segundo pin en la contraseña del usuario es: " ♥ ", y este símbolo pertenece a la tercera dimensión. En este ejemplo, los símbolos de la tercera dimensión aparecen en el lado inferior izquierdo de cualquier *token*. Por lo tanto, el usuario solo necesita mirar el lado inferior izquierdo de cada uno de los 16 *tokens*. En este ejemplo, está en el *token* D3, por lo que el usuario debe hacer clic en D3 (paso 440 de la figura 4), que se resalta en esta ilustración.
- Después de que el usuario haga clic en D3 (paso 440 de la figura 4), el *token* se copia en la segunda posición del código de acceso (G4 353).
- El tercer pin de la contraseña es "2" y pertenece a la 1ª dimensión. Por lo tanto, en este ejemplo, el usuario solo necesita mirar la posición superior izquierda de cada uno de los 16 *tokens*. En este ejemplo, no existe. De acuerdo con las reglas de selección de *tokens* (Fig. 6), el usuario puede y debe seleccionar un *token* comodín (cualquier *token*) para la posición de ese pin. En este ejemplo, el usuario hace clic aleatoriamente en el *token* C3. Ese *token* se copia en la posición del tercer pin: G4 356.
- El cuarto y último pin es " ☒ ", y este símbolo pertenece a la cuarta dimensión. Los símbolos de la cuarta dimensión en este ejemplo aparecen en el lado inferior derecho de cualquier *token*. Por lo tanto, el usuario solo necesita marcar el lado inferior derecho de cada uno de los 16 *tokens*. En este ejemplo, está en el *token* B3, por lo que el usuario debe hacer clic en B3 (paso 440 de la figura 4). En esta ilustración, se resalta. Después de que el usuario haga clic en B3, el *token* se copia en la cuarta posición del código de acceso (G4 359).
- Dado que solo hay 4 pines en el código de acceso, las posiciones G4 362 y G4 365 se dejan en blanco y deben permanecer en blanco. Si el usuario introduce un *token* en uno o ambos lugares, el sistema le negará el acceso al usuario, porque el usuario introdujo más *tokens* que los 4 pines originales (paso 450 de la figura 4).

- Después de que el usuario introduzca los 4 *tokens*, el usuario hará clic en el botón "Iniciar sesión" (G4 370) para que el sistema sepa que el usuario ha finalizado el proceso de selección de *tokens*, y el sistema verificará si los *tokens* introducidos son válidos según a las reglas descritas en la Fig. 7 (paso 450 de la Fig. 4).
- En este ejemplo, los *tokens* introducidos por el usuario son válidos y el sistema muestra un mensaje de "Inicio de sesión exitoso" (G4 380) y otorga acceso al usuario (paso 460 de la figura 4).

[0076] La figura 9C es una captura de pantalla de muestra de una trama de ejecución del proceso de inicio de sesión ejecutado por el sistema 100 para un proceso de inicio de sesión fallido, de acuerdo con una realización de la presente invención. Muestra un ejemplo de cómo puede verse la pantalla de realización de GATE_4 si el usuario pasa por un proceso de inicio de sesión fallido (Fig.4) de la siguiente manera:

- El usuario introduce una identificación de usuario: "admin" (G4 307) (paso 410 de la Fig. 4).
- El usuario hace clic en el botón "Entrar" (G4 310). El sistema verifica si el ID "admin" ya está en su memoria (paso 420 de la Fig. 4), si no existe, mostrará un mensaje (no mostrado) mostrando "El ID de usuario no existe, introduzca una identificación de usuario válida ". Si existe, el sistema mostrará una tabla de 4 x 4 (G4 321). Para describir mejor la tabla, las filas se marcan preferiblemente de arriba a abajo como sigue: A, B, C, D. Las columnas también se marcan preferiblemente de izquierda a derecha como sigue: 1, 2, 3, 4. Los *tokens* en esta tabla se genera de acuerdo con las reglas descritas en la Fig.5.
- Como sabemos por la Fig. 8B que el código de acceso asociado con la identificación de usuario "admin" es: "① ♥ 2 ☒ ", el usuario debe comenzar revisando los 16 *tokens* en la tabla para encontrar el primer pin:" ① ". Debido a que el símbolo ① pertenece a la segunda dimensión, en este ejemplo aparece en la posición superior derecha de cualquier *token*, por lo que el usuario solo necesita escanear la parte superior derecha de cada *token* para ver si ① existe. En este ejemplo, está en el *token* A1. El usuario debe hacer clic en este *token* de acuerdo con las reglas descritas en la Fig. 6.
- Después de que el usuario haga clic en A1, el *token* se copia en la primera posición del código de acceso (G4 351) (paso 440 de la figura 4).
- El segundo pin en el código de acceso del usuario es: " ♥ ", y este símbolo pertenece a la tercera dimensión. En este ejemplo, los símbolos de la tercera dimensión aparecen en el lado inferior izquierdo de cualquier *token*. Por lo tanto, el usuario solo necesita mirar el lado inferior izquierdo de cada uno de los 16 *tokens*. En este ejemplo, está en el *token* D1. Por lo tanto, el usuario debe hacer clic en D1 (paso 440 de la Fig. 4). Sin embargo, en este ejemplo, el usuario no hizo clic en este *token* y, en cambio, hizo clic en B4. Este es el *token* incorrecto, y el sistema lo registrará como un error y denegará el acceso del usuario.
- Después de que el usuario hace clic en B4 (paso 440 de la figura 4), el *token* se copia en la segunda posición del código de acceso (G4 354).
- El tercer pin de la contraseña es "2" y pertenece a la 1ª dimensión. Por lo tanto, en este ejemplo, el usuario solo necesita mirar la posición superior izquierda de cada uno de los 16 *tokens*. En este ejemplo, no existe. De acuerdo con las reglas de selección de *tokens* (Fig. 6), el usuario puede y debe seleccionar un *token* comodín (cualquier *token*) para la posición de ese pin. En este ejemplo, el usuario hace clic aleatoriamente en el *token* C4. Ese *token* se copia en la posición del tercer pin: G4 357.
- El cuarto y último pin es " ☒ ", y este símbolo pertenece a la cuarta dimensión. Los símbolos de la cuarta dimensión en este ejemplo aparecen en el lado inferior derecho de cualquier *token*. Por lo tanto, el usuario solo necesita marcar el lado inferior derecho de cada uno de los 16 *tokens*. En este ejemplo, está en el *token* A2, por lo que el usuario debe hacer clic en A2 (paso 440 de la figura 4). Después de que el usuario haga clic en A2, el *token* se copia en la cuarta posición del código de acceso (G4 360).
- Dado que solo hay 4 pines en el código de acceso, las posiciones G4 363 y G4 366 se dejan en blanco y deben permanecer en blanco. Si el usuario introduce un *token* en uno o ambos lugares, el sistema le negará el acceso al usuario, porque el usuario introdujo más *tokens* que los 4 pines originales (paso 450 de la figura 4).
- Después de que el usuario introduce los 4 *tokens*, el usuario hará clic en el botón "Iniciar sesión" (G4 371) para que el sistema sepa que el usuario ha finalizado el proceso de selección de *tokens*, y el sistema verificará si los *tokens* introducidos son válidos según a las reglas descritas en la Fig. 7 (paso 450 de la Fig. 4).
- En este ejemplo, los *tokens* introducidos por el usuario no son válidos y el sistema muestra un mensaje de "Inicio de sesión fallido" (G4 381) y deniega el acceso al usuario (paso 470 de la figura 4).

[0077] La figura 9D es una captura de pantalla de muestra de una trama de ejecución del proceso de inicio de sesión ejecutado por el sistema 100 para otro proceso de inicio de sesión fallido, de acuerdo con una realización de la presente invención. Muestra un ejemplo de cómo puede verse la pantalla de realización de GATE_4 si el usuario pasa por un proceso de inicio de sesión fallido (Fig.4) de la siguiente manera:

- El usuario introduce una identificación de usuario: "admin" (G4 308) (paso 410 de la Fig. 4).

- El usuario hace clic en el botón "Entrar" (G4 311). El sistema verifica si el ID "admin" ya está en su memoria (paso 420 de la Fig. 4), si no existe, mostrará un mensaje (no mostrado) mostrando "El ID de usuario no existe, introduzca una identificación de usuario válida ". Si existe, el sistema mostrará una tabla de 4 x 4 (G4 322). Para describir mejor la tabla, las filas se marcan preferiblemente de arriba a abajo como sigue: A, B, C, D. Las columnas también se marcan preferiblemente de izquierda a derecha como sigue: 1, 2, 3, 4. Los *tokens* en esta tabla se genera de acuerdo con las reglas descritas en la Fig.5.
- Como sabemos por la Fig. 8B que el código de acceso asociado con la identificación de usuario "admin" es: "① ♥ 2 ☒ ", el usuario debe comenzar revisando los 16 *tokens* de la tabla para encontrar el primer pin:" ① ". Debido a que el símbolo ① pertenece a la 2ª dimensión, en este ejemplo aparece en la posición superior derecha de cualquier *token*, por lo que el usuario solo necesita escanear la parte superior derecha de cada *token* para ver si ① existe. En este ejemplo, está en el *token* B2. El usuario debe hacer clic en este *token* de acuerdo con las reglas descritas en la Fig.6.
- Después de que el usuario haga clic en B2, el *token* se copia en la primera posición del código de acceso (G4 352) (paso 440 de la figura 4).
- El segundo pin en la contraseña del usuario es: " ♥ ", y este símbolo pertenece a la tercera dimensión. En este ejemplo, los símbolos de la tercera dimensión aparecen en el lado inferior izquierdo de cualquier *token*. Por lo tanto, el usuario solo necesita mirar el lado inferior izquierdo de cada uno de los 16 *tokens*. En En este ejemplo, no existe. De acuerdo con las reglas de selección de *tokens* (Fig. 6), el usuario puede y debe seleccionar un *token* comodín (cualquier *token*) para la posición de ese pin. En este ejemplo, el usuario hace clic aleatoriamente en el *token* A4. el *token* se copia a la 2ª posición del pin - G4 355.
- El tercer pin de la contraseña es "2" y pertenece a la 1ª dimensión. Por lo tanto, en este ejemplo, el usuario solo necesita mirar la posición superior izquierda de cada uno de los 16 *tokens*. En este ejemplo, está en el *token* B3. En este ejemplo, el usuario hace clic en B3 y ese *token* se copia en la posición del tercer pin: G4 358.
- El cuarto y último pin es " ☒ ", y este símbolo pertenece a la cuarta dimensión. Los símbolos de la cuarta dimensión en este ejemplo aparecen en el lado inferior derecho de cualquier *token*. Por lo tanto, el usuario solo necesita marcar el lado inferior derecho de cada uno de los 16 *tokens*. En este ejemplo, está en el *token* D1, por lo que el usuario debe hacer clic en D1 (paso 440 de la figura 4). Después de que el usuario hace clic en D1, el *token* se copia en la cuarta posición del código de acceso (G4 361).
- Dado que solo hay 4 pines en el código de acceso, las posiciones G4 364 y G4 367 deben dejarse en blanco. En este ejemplo, el usuario introdujo un *token* adicional D2 y, por lo tanto, el sistema negará el acceso al usuario, porque el usuario introdujo más de los 4 pines originales.
- Después de que el usuario introduce los 5 *tokens*, el usuario hará clic en el botón "Iniciar sesión" (G4 372) para que el sistema sepa que el usuario ha finalizado el proceso de selección de *tokens*, y el sistema verificará si los *tokens* introducidos son válidos según a las reglas descritas en la Fig. 7 (paso 450 de la Fig. 4).
- En este ejemplo, el usuario introdujo demasiados *tokens* y el sistema muestra un mensaje de "Inicio de sesión fallido" (G4 382) y deniega el acceso al usuario (paso 470 de la figura 4).

[0078] La figura 10A es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_4 en formato de texto. Refleja una pantalla vacía cuando se inician los programas. Esta imagen se utiliza como base de comparación para las Figuras 10B, 10C y 10D a continuación. Esta pantalla solo se muestra como una explicación de lo que sucede de forma oculta. No se muestra durante el inicio de sesión del usuario en tiempo real.

[0079] La figura 10B es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_4 en formato de texto. Muestra lo que sucede de forma oculta cuando se está llevando a cabo el proceso de inicio de sesión de usuario de la figura 9B y muestra cómo se ve el flujo de proceso de la figura 4 en una realización de muestra. Esta es solo una demostración y no se muestra durante el inicio de sesión del usuario en tiempo real. Se utiliza para ilustrar visualmente las reglas de validación de *tokens* que se muestran en la Fig.7.

[0080] Hay 3 columnas: "Lado del cliente" (lado izquierdo), "Conexión de red" (centro) y "Lado del servidor" (lado derecho). El proceso comienza desde el lado del cliente cuando el usuario introduce la identificación del usuario, luego la información pasa a través de la conexión de red al lado del servidor. El servidor genera 16 *tokens* y los pasa a la red, luego los *tokens* se pasan al lado del cliente.

[0081] El usuario selecciona los *tokens* de acuerdo con las reglas de selección de *tokens* que se muestran en la Fig.6. Los *tokens* seleccionados se pasan a la red, luego se pasan al lado del servidor para ser validados, el resultado de otorgar o denegar el acceso se pasa a través de la red a el lado del cliente. El flujo del proceso está marcado con flechas en la Fig. 10B. A continuación se proporciona una explicación más detallada.

- El usuario introduce una identificación de usuario: "admin" (G4 406).
- El usuario hace clic en el botón "Entrar" (G4 409).

- La identificación de usuario "admin" (G4 406) se muestra en el lado del cliente (G4 411) y se pasa (G4 421) a la conexión de red (G4 413), luego se pasa nuevamente (G4 422) al lado del servidor (G4 415)).
- En el lado del servidor, el sistema verifica su memoria para ver si existe la identificación de usuario "admin", si no, el sistema mostrará un mensaje: "La identificación de usuario no existe, introduzca una identificación de usuario válida" (no se muestra). Se utiliza el mismo ejemplo de la Fig. 8B, el código de acceso en la memoria es: ""① ♥ 2 ✉ ", el sistema lo encuentra en la memoria (G4 417).
- El sistema genera 16 *tokens* (G4 423) de acuerdo con las reglas de generación de *tokens* que se muestran en la Fig.5.
- Los 16 *tokens* se pasan (G4 424) a la red.
- La red pasa (G4 425) los *tokens* al lado del cliente.
- Los 16 *tokens* se muestran en la pantalla de inicio de sesión del usuario, como se muestra en la Fig. 9B, en una tabla de 4 x 4 (G4 320 en la Fig. 9B).
- El usuario selecciona (G4 426) los 4 *tokens*: G4 350, G4 353, G4 356 y G4 359.
- Los 4 *tokens* seleccionados por el usuario se pasan (G4 427) a la red después de que el usuario haga clic en "Iniciar sesión" (G4 370 en la Fig. 9B).
- Los 4 *tokens* seleccionados por el usuario se pasan (G4 428) al lado del servidor.
- En el lado del servidor, el sistema verifica los 4 *tokens* uno por uno: C11, C12, C13 y C14, en este ejemplo todos son correctos.
- El resultado anterior de un inicio de sesión exitoso (G4 429) se pasa (G4 430) a la red.
- La red pasa (G4 431) el resultado al lado del cliente y muestra (G4 432) un mensaje que se muestra en G4 380 de la Fig. 9B.

[0082] La Figura 10C es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_4 en formato de texto. Muestra lo que sucede de forma oculta cuando se está llevando a cabo el proceso de inicio de sesión del usuario de la figura 9C y muestra cómo se ve el flujo del proceso de la figura 4 en una realización de muestra. Esta es solo una demostración y no se muestra durante el inicio de sesión del usuario en tiempo real. Se utiliza para ilustrar visualmente las reglas de validación de *tokens* que se muestran en la Fig.7.

[0083] Hay 3 columnas: "Lado del cliente" (lado izquierdo), "Conexión de red" (centro) y "Lado del servidor" (lado derecho). El proceso comienza desde el lado del cliente cuando el usuario introduce la identificación del usuario, luego la información pasa a través de la conexión de red al lado del servidor. El servidor genera 16 *tokens* y los pasa a la red, luego los *tokens* se pasan al lado del cliente.

[0084] El usuario selecciona los *tokens* de acuerdo con las reglas de selección de *tokens* que se muestran en la Fig.6. Los *tokens* seleccionados se pasan a la red, luego se pasan al lado del servidor para ser validados, el resultado de otorgar o denegar el acceso se pasa a través de la red a el lado del cliente. El flujo del proceso está marcado por flechas en la Fig. 10C. A continuación se proporciona una explicación más detallada.

- El usuario introduce una identificación de usuario: "admin" (G4 506).
- El usuario hace clic en el botón "Entrar" (G4 509).
- La identificación de usuario "admin" (G4 506) se muestra en el lado del cliente (G4 511) y se pasa (G4 521) a la conexión de red (G4 513), luego se pasa nuevamente (G4 522) al lado del servidor (G4 515)).
- En el lado del servidor, el sistema verifica su memoria para ver si existe la identificación de usuario "admin", si no, el sistema mostrará un mensaje: "La identificación de usuario no existe, introduzca una identificación de usuario válida" (no se muestra). Se utiliza el mismo ejemplo de la Fig. 8B, el código de acceso en la memoria es: "① ♥ 2 ✉ ", el sistema lo encuentra en la memoria (G4 517).
- El sistema genera 16 *tokens* (G4 523) de acuerdo con las reglas de generación de *tokens* que se muestran en la Fig.5.
- Los 16 *tokens* se pasan (G4 524) a la red.
- La red pasa (G4 525) los *tokens* al lado del cliente.
- Los 16 *tokens* se muestran en la pantalla de inicio de sesión del usuario, como se muestra en la Fig. 9C, en una tabla de 4 x 4 (G4 321 en la Fig. 9C).
- El usuario selecciona (G4 526) los 4 *tokens*: G4 351, G4 354, G4 357 y G4 360.
- Los 4 *tokens* seleccionados por el usuario se pasan (G4 527) a la red después de que el usuario haga clic en "Iniciar sesión" (G4 371 en la Fig. 9C).
- Los 4 *tokens* seleccionados por el usuario se pasan (G4 528) al lado del servidor.
- En el lado del servidor, el sistema verifica los 4 *tokens* uno por uno: C21, C22, C23 y C24, en este ejemplo los 2ª el *token* es incorrecto (porque el segundo pin " ♥ " existe en el *token* D1, pero el usuario seleccionó el *token* B4 que estaba mal. Por lo tanto, el resultado es un inicio de sesión fallido).

- El resultado anterior de un inicio de sesión fallido (G4 529) se pasa (G4 530) a la red.
- La red pasa (G4 531) el resultado al lado del cliente y muestra (G4 532) un mensaje que se muestra en G4 381 de la Fig. 9C.

[0085] La Figura 10D es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_4 en formato de texto. Muestra lo que sucede de forma oculta cuando tiene lugar el proceso de inicio de sesión del usuario de la figura 9D y muestra cómo se ve el flujo del proceso de la figura 4 en una realización de muestra. Esta es solo una demostración y no se muestra durante el inicio de sesión del usuario en tiempo real. Se utiliza para ilustrar visualmente las reglas de validación de *tokens* que se muestran en la Fig.7.

[0086] Hay 3 columnas: "Lado del cliente" (lado izquierdo), "Conexión de red" (centro) y "Lado del servidor" (lado derecho). El proceso comienza desde el lado del cliente cuando el usuario introduce la identificación del usuario, luego la información pasa a través de la conexión de red al lado del servidor. El servidor genera 16 *tokens* y los pasa a la red, luego los *tokens* se pasan al lado del cliente.

[0087] El usuario selecciona los *tokens* de acuerdo con las reglas de selección de *tokens* que se muestran en la Fig.6. Los *tokens* seleccionados se pasan a la red, luego se pasan al lado del servidor para ser validados, el resultado de otorgar o denegar el acceso se pasa a través de la red a el lado del cliente. El flujo del proceso está marcado con flechas en la Fig. 10D. A continuación se proporciona una explicación más detallada.

- El usuario introduce una identificación de usuario: "admin" (G4 606).
- El usuario hace clic en el botón "Entrar" (G4 609).
- La identificación de usuario "admin" (G4 606) se muestra en el lado del cliente (G4 611) y se pasa (G4 621) a la conexión de red (G4 613), luego se pasa nuevamente (G4 622) al lado del servidor (G4 615)).
- En el lado del servidor, el sistema verifica su memoria para ver si existe la identificación de usuario "admin", si no, el sistema mostrará un mensaje: "La identificación de usuario no existe, introduzca una identificación de usuario válida" (no se muestra). Se utiliza el mismo ejemplo de la Fig. 8B, el código de acceso en la memoria es: ""ⓘ♥2✉", el sistema lo encuentra en la memoria (G4 617).
- El sistema genera 16 *tokens* (G4 623) de acuerdo con las reglas de generación de *tokens* que se muestran en la Fig.5.
- Los 16 *tokens* se pasan (G4 624) a la red.
- La red pasa (G4 625) los *tokens* al lado del cliente.
- Los 16 *tokens* se muestran en la pantalla de inicio de sesión del usuario, como se muestra en la Fig. 9D, en una tabla de 4 x 4 (G4 322 en la Fig. 9D).
- El usuario selecciona (G4 626) los 5 *tokens*: G4 352, G4 355, G4 358, G4 361 y G4 364.
- Los 5 *tokens* seleccionados por el usuario se pasan (G4 627) a la red después de que el usuario haga clic en "Iniciar sesión" (G4 372 en la Fig. 9D).
- Los 5 *tokens* seleccionados por el usuario se pasan (G4 628) al lado del servidor.
- En el lado del servidor, el sistema verifica los 5 *tokens* uno por uno: C31, C32, C33, C34 y C35, en este ejemplo el 5° *token* es incorrecto (debido a que el código de acceso tiene solo 4 pines, pero el usuario introdujo un quinto *token*, eso estaba mal. Por lo tanto, el resultado es un inicio de sesión fallido).
- El resultado anterior de un inicio de sesión fallido (G4 629) se pasa (G4 630) a la red.
- La red pasa (G4 631) el resultado al lado del cliente y muestra (G4 632) un mensaje que se muestra en G4 382 de la Fig. 9D.

[0088] La Figura 11A es una captura de pantalla de muestra del proceso de creación (registro) de identificación de usuario en la realización GATE_5. Refleja una pantalla vacía cuando se inician los programas. Esta imagen se utiliza como base de comparación para la Figura 11B a continuación.

[0089] La figura 11B es una captura de pantalla de muestra del proceso de creación (registro) de identificación de usuario en la realización GATE_5. Muestra dónde se encuentra cada dimensión de los símbolos en un *token*. También refleja cómo un usuario crea una nueva identificación de usuario y cómo un usuario selecciona y guarda pines para formar un código de acceso asociado con la identificación de usuario. El proceso procede de la siguiente manera:

- El usuario introduce una nueva identificación de usuario: "admin" (G5 206), luego haga clic en el botón "Verificar disponibilidad" (G5 208). El sistema verifica si la identificación "admin" ya está en su memoria, si es así, mostrará un cuadro de diálogo (no se muestra) que pregunta: "La identificación de usuario ya existe, ¿desea sobrescribir la contraseña existente?" Si el usuario no desea sobrescribir la contraseña anterior, el proceso cerrará el cuadro de diálogo y esperará a que el usuario introduzca otra identificación de usuario. Si el ID de usuario "admin" no existe, o si existe, pero el usuario desea sobrescribir el código de acceso existente, el sistema habilitará los botones en G5 212, G5 222, G5 232, G5 242 y G5 248, cada

uno tiene 26 símbolos de una dimensión predefinida como se describe en la Fig. 2B. Por ejemplo, G5 212 incluye los 26 símbolos de la 1ª dimensión, esos símbolos se mostrarán en un *token* en la posición "[1]" (superior izquierda) como se muestra en G5 210. Los 26 símbolos son de "Ⓐ" a " ⓩ ".

G5 222 muestra 26 símbolos de "α" a "ϟ", son de la 2ª dimensión y aparecerán en cualquier *token* en la posición" [2] "(G5 220: arriba a la derecha). G5 232 muestra 26 números del 1 al 26, son de la 3ra dimensión que aparecen en la posición "[3]" (G5 230: en el medio) en un *token*, y G5 242 muestra 26 símbolos de "○" a "≠", son de la 4ta dimensión y se mostrarán en cualquier *token* en la posición "[4]" (G5 240: abajo a la izquierda). G5 248 muestra 26 símbolos de "+" a "₪", son de la quinta dimensión y aparecerán en cualquier *token* en la posición" [5] "(G5 246: abajo a la derecha). En este momento del proceso, el sistema habilitará los botones anteriores en G5 212, G5 222, G5 232, G5 242 y G5 248, para que el usuario pueda hacer clic en cualquiera de ellos. Como comparación, en la Fig. 11A esos botones no están habilitados y se ven pálidos, porque el usuario aún no ha introducido ninguna identificación de usuario. Sin una identificación de usuario, no permitirá que el usuario seleccione ningún pin.

- El usuario hace clic en "$"entre los símbolos en G5 248 para seleccionar el primer pin, y aparece en G5 250.
- El usuario hace clic en "=" entre los símbolos en G5 242 para seleccionar el segundo pin, y aparece en G5 252.
- El usuario hace clic en "Ⓜ" entre los símbolos en G5 212 para seleccionar el tercer pin, y aparece en G5 254.
- El usuario hace clic en "©" entre los símbolos en G5 212 para seleccionar el cuarto pin, y aparece en G5 256.
- El usuario hace clic en "2" entre los símbolos en G5 232 para seleccionar el quinto pin, y aparece en G5 258.
- El usuario hace clic en "☺"entre los símbolos en G5 242 para seleccionar el sexto pin, y aparece en G5 260.
- En este ejemplo, el usuario eligió tener 6 pines en su contraseña, por lo que la longitud de su contraseña es 6.
- Luego, el usuario finaliza el proceso de creación (registro) de identificación de usuario haciendo clic en el botón "Guardar" (G5 270). El sistema guardará la contraseña "$= Ⓜ © 2☺" con la identificación de usuario" admin "en la memoria.

[0090] La figura 12A es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_5. Refleja una pantalla vacía cuando se inician los programas. Esta imagen se utiliza como base de comparación para las Figuras 12B, 12C y 12D a continuación.

[0091] La figura 12B es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_5. Muestra una tabla de 4 x 4 de 16 *tokens* de los que un usuario selecciona el código de acceso. Refleja cómo funciona el proceso de selección de *tokens* y resalta los símbolos en los *tokens* que un usuario selecciona y tiene pines de usuario en ellos como parte del código de acceso del usuario. También muestra cómo puede ser un inicio de sesión exitoso. Este proceso de ejemplo sigue el ejemplo de la Fig. 11B, por lo que se utilizará la misma contraseña. El proceso procede de la siguiente manera: El usuario introduce una nueva identificación de usuario: "admin" (G5 306).

- El usuario hace clic en el botón "Entrar" (G5 309). El sistema verifica si la identificación "admin" ya está en su memoria, si no existe, mostrará un mensaje (no se muestra) mostrando "La identificación de usuario no existe, introduzca una identificación de usuario válida". Si existe, el sistema mostrará una tabla de 4 x 4 (G5 320), para describir mejor la tabla, las filas están marcadas de arriba hacia abajo: A, B, C, D. las columnas también están marcadas de izquierda a derecha: 1,2,3,4. Los *tokens* de esta tabla se generan de acuerdo con las reglas descritas en la Fig.5.
- Dado que sabemos por la Fig. 11B que el código de acceso asociado con la identificación de usuario "admin" es: "$= Ⓜ © 2☺", por lo que el usuario debe comenzar revisando los 16 *tokens* de la tabla para encontrar el primer pin: "$" . Como el símbolo $ pertenece a la quinta dimensión, solo aparecerá en la posición inferior derecha de cualquier *token*, por lo que el usuario solo necesita escanear la parte inferior derecha de cada *token* para ver si $ existe. En nuestro ejemplo, está en el *token* B4, la pantalla se toma en modo de demostración y, en el modo de demostración, el programa resalta el símbolo correspondiente para que el usuario comprenda mejor el proceso. En tiempo real, no se resaltará. En nuestro caso, el $

en B4 está resaltado, ya que se encuentra en el *token* B4, el usuario debe hacer clic en este *token* de acuerdo con las reglas descritas en la Fig.6.

- Después de que el usuario hace clic en B4, el *token* se copia en la primera posición del código de acceso (G5 350).
- El segundo pin en el código de acceso del usuario es: "=", y este símbolo pertenece a la cuarta dimensión, los símbolos de la cuarta dimensión solo aparecen en el lado inferior izquierdo de cualquier *token*, por lo que el usuario solo necesita mirar el lado inferior izquierdo de cada uno de los 16 *tokens*, en nuestro caso está en el *token* D2, por lo que el usuario debe hacer clic en D2. Está resaltado en la captura de pantalla.
- Después de que el usuario hace clic en D2, el *token* se copia en la segunda posición del código de acceso (G5 353).
- El tercer pin en el código de acceso es: "Ⓜ", y pertenece a la 1ª dimensión, por lo que el usuario solo necesita mirar la posición superior izquierda de cada uno de los 16 *tokens*, en nuestro caso está en el *token* D4, por lo que el usuario debe hacer clic en D4. Está resaltado en la captura de pantalla.
- Después de que el usuario hace clic en D4, el *token* se copia en la tercera posición del código de acceso (G5 356).
- El cuarto pin es: "©", y este símbolo pertenece a la 1ª dimensión, los símbolos de la 1ª dimensión solo aparecen en el lado superior izquierdo de cualquier *token*, por lo que el usuario solo necesita marcar el lado superior izquierdo de cada uno de los 16 *tokens*, en En nuestro caso, no existe, de acuerdo con las reglas de selección de *tokens*, el usuario puede y debe seleccionar un *token* comodín (cualquier *token*) en la posición de ese pin, por lo que el usuario hace clic en un *token* aleatorio A4, ese *token* se copia en la posición del 4to pin G5 359.
- El quinto pin en el código de acceso es: "2", y pertenece a la tercera dimensión, por lo que el usuario solo necesita mirar el centro de cada uno de los 16 *tokens*, en nuestro caso está en el *token* D2, por lo que el usuario debe hacer clic en D2. Está resaltado en la captura de pantalla. Después de que el usuario haga clic, el *token* se copia en la quinta posición del código de acceso (G5 362).
- El sexto y último pin es: "☺", y este símbolo pertenece a la cuarta dimensión, los símbolos de la cuarta dimensión solo aparecen en el lado inferior izquierdo de cualquier *token*, por lo que el usuario solo necesita marcar el lado inferior izquierdo de cada uno de los 16 *tokens*, en nuestro caso está en el *token* A4, por lo que el usuario debe hacer clic en A4. También se resalta en el programa de demostración. Después de hacer clic en el usuario, el *token* se copia en la sexta y última posición de la contraseña (G5 365).
- Después de que el usuario introduzca los 6 *tokens*, hará clic en el botón "Iniciar sesión" (G5 370) para que el sistema sepa que ha terminado el proceso de selección de *tokens*, y el sistema verificará si los *tokens* introducidos son válidos de acuerdo con las reglas descritas. en la figura 7.
- En este ejemplo, los *tokens* introducidos por el usuario son válidos y el sistema mostró un mensaje de "Inicio de sesión exitoso" y le otorgó acceso al usuario (G5 380).
- En este ejemplo, los *tokens* en G5 353 y G5 362 son iguales, también lo son los *tokens* en G5 359 y G5 365, es solo una coincidencia, situaciones como esta pueden suceder con bastante frecuencia. Esto podría confundir a cualquiera que esté tratando de adivinar el código de acceso.

[0092] La figura 12C es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_5. Muestra una tabla de 4 x 4 de 16 *tokens* de los que un usuario selecciona el código de acceso. Refleja cómo funciona el proceso de selección de *tokens*. También muestra cómo puede verse un inicio de sesión fallido con 3 pines incorrectos. Este proceso de ejemplo sigue el ejemplo de la Fig. 11B, por lo que se utilizará la misma contraseña. El proceso procede de la siguiente manera:

- El usuario introduce una identificación de usuario: "admin" (G5 307).
- El usuario hace clic en el botón "Entrar" (G5 310). El sistema verifica si la identificación "admin" ya está en su memoria, si no existe, mostrará un mensaje (no se muestra) mostrando "La identificación de usuario no existe, introduzca una identificación de usuario válida". Si existe, el sistema mostrará una tabla de 4 x 4 (G5 321), para describir mejor la tabla, las filas están marcadas de arriba a abajo: A, B, C, D. las columnas también están marcadas de izquierda a derecha: 1,2,3,4. Los *tokens* de esta tabla se generan de acuerdo con las reglas descritas en la Fig.5.
- Dado que sabemos por la Fig. 11B que el código de acceso asociado con la identificación de usuario "admin" es: "$= Ⓜ © 2 ☺", por lo que el usuario debe comenzar revisando los 16 *tokens* de la tabla para encontrar el primer pin: "$" . Como el símbolo $ pertenece a la quinta dimensión, solo aparecerá en la posición inferior derecha de cualquier *token*, por lo que el usuario solo necesita escanear la parte inferior derecha de cada *token* para ver si $ existe. En nuestro ejemplo, está en el *token* A2, el usuario debe hacer clic en este *token* de acuerdo con las reglas descritas en la Fig.6.

- Después de que el usuario hace clic en A2, el *token* se copia en la primera posición del código de acceso (G5 351).
- El segundo pin en el código de acceso del usuario es: "=", este símbolo pertenece a la cuarta dimensión, y los símbolos de la cuarta dimensión solo aparecen en el lado inferior izquierdo de cualquier *token*, por lo que el usuario solo necesita mirar el lado inferior izquierdo de cada uno de los 16 *tokens*, en nuestro caso no existe, de acuerdo con las reglas de selección de *tokens*, el usuario puede y debe seleccionar un *token* comodín (cualquier *token*) en la posición de ese pin, por lo que el usuario hace clic en un *token* aleatorio A3.
- Después de que el usuario hace clic en A3, el *token* se copia en la segunda posición del código de acceso (G5 354).
- El tercer pin en el código de acceso es: "Ⓜ", y pertenece a la 1ra dimensión, por lo que el usuario solo necesita mirar la posición superior izquierda de cada uno de los 16 *tokens*, en nuestro caso está en el *token* C1, el usuario debe hacer clic en C1. En nuestro caso, el usuario hace clic en C1.
- Después de que el usuario hace clic en C1, el *token* se copia en la tercera posición del código de acceso (G5 357).
- El cuarto pin es: "©", y este símbolo pertenece a la 1ª dimensión, el usuario solo necesita mirar la posición superior izquierda de cada uno de los 16 *tokens*, en nuestro caso está en el *token* D2, el usuario debe hacer clic en D2. En nuestro caso, el usuario no hizo clic en D2, sino que hizo clic en C2. Esto está mal y el sistema denegará el acceso de los usuarios.
- Después de que el usuario hace clic en C2, el *token* se copia en la cuarta posición del código de acceso (G5 360).
- El quinto pin en el código de acceso es: "2", y pertenece a la tercera dimensión, por lo que el usuario solo necesita mirar el centro de cada uno de los 16 *tokens*, en nuestro caso está en el *token* C3, y de acuerdo con la selección del *token*. Las reglas de la Fig. 6 el usuario debe seleccionar este *token*, pero en el ejemplo, el usuario seleccionó el *token* B2 en su lugar, esto es incorrecto, y el sistema lo comprobará y lo notará.
- Una vez que el usuario hace clic en el *token* B2 incorrecto, se copia en la quinta posición del código de acceso (G5 363).
- El sexto y último pin es: "☺", y este símbolo pertenece a la cuarta dimensión, los símbolos de la cuarta dimensión solo aparecen en el lado inferior izquierdo de cualquier *token*, por lo que el usuario solo necesita marcar el lado inferior izquierdo de cada uno de los 16 *tokens*, en nuestro caso está en el *token* D1, el usuario debe hacer clic en D1. En nuestro caso, el usuario no hizo clic en D1, sino que hizo clic en el *token* C4, esto es incorrecto y el sistema lo notará.
- Después de que el usuario hace clic en el *token* C4 incorrecto, se copia en la sexta posición del código de acceso (G5 366).
- Después de que el usuario introduce los 6 *tokens*, hace clic en el botón "Iniciar sesión" (G5 371) para que el sistema sepa que ha finalizado el proceso de selección de *tokens*, y el sistema verificará si los *tokens* introducidos son válidos de acuerdo con las reglas descritas en Figura 7.
- En este ejemplo, los *tokens* introducidos por el usuario no son válidos y el sistema mostró un mensaje de "Error de inicio de sesión" y denegó el acceso del usuario (G5 381).

[0093] La figura 12D es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_5. Muestra una tabla de 4 x 4 de 16 *tokens* de los que un usuario selecciona el código de acceso. Refleja cómo funciona el proceso de selección de *tokens*. También muestra cómo puede ser un inicio de sesión fallido al que le falta un pin. Este proceso de ejemplo sigue el ejemplo de la Fig. 11B, por lo que se utilizará la misma contraseña. El proceso procede de la siguiente manera:

- El usuario introduce una identificación de usuario: "admin" (G5 308).
- El usuario hace clic en el botón "Entrar" (G5 311). El sistema verifica si la identificación "admin" ya está en su memoria, si no existe, mostrará un mensaje (no se muestra) mostrando "La identificación de usuario no existe, introduzca una identificación de usuario válida". Si existe, el sistema mostrará una tabla de 4 x 4 (G5 322), para describir mejor la tabla, las filas están marcadas de arriba a abajo: A, B, C, D. las columnas también están marcadas de izquierda a derecha: 1,2,3,4. Los *tokens* de esta tabla se generan de acuerdo con las reglas descritas en la Fig.5.
- Dado que sabemos por la Fig. 11B que el código de acceso asociado con la identificación de usuario "admin" es: "$= Ⓜ © 2 ☺", por lo que el usuario debe comenzar revisando los 16 *tokens* de la tabla para encontrar el primer pin:"$" . Como el símbolo $ pertenece a la quinta dimensión, solo aparecerá en la posición inferior derecha de cualquier *token*, el usuario solo necesita escanear la parte inferior derecha de cada *token* para ver si $ existe. En nuestro ejemplo, no existe, el usuario puede y debe seleccionar

un *token* comodín (cualquier *token*) en la posición de ese pin, por lo que el usuario hace clic en un *token* aleatorio D3.

- Después de que el usuario hace clic en D3, el *token* se copia en la primera posición del código de acceso (G5 352).
- El segundo pin en el código de acceso del usuario es: "=", este símbolo pertenece a la cuarta dimensión, los símbolos de la cuarta dimensión solo aparecen en el lado inferior izquierdo de cualquier *token*, por lo que el usuario solo necesita mirar el lado inferior izquierdo de cada uno de los 16 *tokens*, en nuestro caso está en el *token* C1, el usuario debe hacer clic en este *token*.
- Después de que el usuario hace clic en C1, el *token* se copia en la segunda posición del código de acceso (G5 355).
- El tercer pin en el código de acceso es: "Ⓜ", y pertenece a la 1ra dimensión, por lo que el usuario solo necesita mirar la posición superior izquierda de cada uno de los 16 *tokens*, en nuestro caso está en el *token* A1, por lo que el usuario debe haga clic en A1. En nuestro caso, el usuario hace clic en A1.
- Después de que el usuario hace clic en A1, el *token* se copia en la tercera posición del código de acceso (G5 358).
- El cuarto pin es: "©", y este símbolo pertenece a la 1ª dimensión, el usuario solo necesita mirar la posición superior izquierda de cada uno de los 16 *tokens*, en nuestro caso está en el *token* D3, por lo que el usuario debe hacer clic en D3. En nuestro caso, el usuario hizo clic en D3.
- Después de que el usuario hace clic en D3, el *token* se copia en la cuarta posición del código de acceso (G5 361).
- El quinto pin en el código de acceso es: "2", y pertenece a la tercera dimensión, el usuario solo necesita mirar el centro de cada uno de los 16 *tokens*, en nuestro caso está en el *token* C4, y de acuerdo con las reglas de selección de *tokens*. en la figura 6, el usuario debe seleccionar este *token*, y en el ejemplo, el usuario seleccionó el *token* C4.
- Después de que el usuario hace clic en el *token* C4, se copia en la quinta posición del código de acceso (G5 364).
- El sexto y último pin es: "☺", y este símbolo pertenece a la 4ª dimensión, y los símbolos de la 4ª dimensión solo aparecen en el lado inferior izquierdo de cualquier *token*, por lo que el usuario solo necesita marcar el lado inferior izquierdo de cada uno de los 16 *tokens*, en nuestro caso está en *token* C2, por lo que el usuario debe hacer clic en C2. Pero en nuestro caso, el usuario no hizo clic en C2, sino que dejó la última posición en blanco y solo introdujo 5 pines. Esto es incorrecto.
- Después de que el usuario introduce los 5 *tokens* anteriores, hizo clic en el botón "Iniciar sesión" (G5 372) para que el sistema sepa que ha terminado el proceso de selección de *tokens*, y el sistema verificará si los *tokens* introducidos son válidos de acuerdo con las reglas descritas. en la figura 7.
- En este ejemplo, los *tokens* introducidos por el usuario no son válidos, porque el código de acceso original tenía 6 pines, pero el usuario en nuestro ejemplo solo introdujo 5 *tokens*, por lo que la solicitud de acceso del usuario fue denegada y el sistema mostró un mensaje de "Error al iniciar sesión" (G5 382).

[0094] La figura 13A es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_5 en formato de texto. Refleja una pantalla vacía cuando se inician los programas. Esta imagen se utiliza como base de comparación para las Figuras 13B, 13C y 13D a continuación. Esta pantalla solo se muestra como una explicación de lo que sucede de forma oculta. No se muestra durante el inicio de sesión del usuario en tiempo real.

[0095] La figura 13B es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_5 en formato de texto. Muestra lo que sucede entre bastidores cuando se lleva a cabo el proceso de inicio de sesión del usuario de la figura 12B y muestra cómo se ve el flujo del proceso de la figura 4 en una realización de muestra. Esta es solo una demostración y no se muestra durante el inicio de sesión del usuario en tiempo real. Se utiliza para ilustrar visualmente las reglas de validación de *tokens* que se muestran en la Fig.7.

[0096] Hay 3 columnas: "Lado del cliente" (lado izquierdo), "Conexión de red" (centro) y "Lado del servidor" (lado derecho). El proceso comienza desde el lado del cliente cuando el usuario introduce la identificación del usuario, luego la información pasa a través de la conexión de red al lado del servidor. El servidor genera 16 *tokens* y los pasa a la red, luego los *tokens* se pasan al lado del cliente.

[0097] El usuario selecciona los *tokens* de acuerdo con las reglas de selección de *tokens* que se muestran en la figura 6, luego los *tokens* seleccionados se pasan a la red y luego al lado del servidor para ser validados. El resultado de otorgar o denegar el acceso se transmite a través de la red al lado del cliente. El flujo del proceso está marcado con flechas en la Fig. 13B. El proceso procede de la siguiente manera:

- El usuario introduce la identificación de usuario "admin" (G5 406).
- El usuario hace clic en "Enter" (G5 409).

- La identificación de usuario "admin" (G5 406) se muestra en el lado del cliente (G5 411) y se pasa (G5 421) a la conexión de red (G5 413), luego se pasa nuevamente (G5 422) al lado del servidor (G5 415).
- En el lado del servidor, el sistema verifica su memoria para ver si existe la identificación de usuario "admin", si no lo hace, el sistema mostrará un mensaje: "La identificación de usuario no existe, introduzca una identificación de usuario válida" (no se muestra). Se utiliza el mismo ejemplo que en la Fig. 11B, por lo que el código de acceso en la memoria es: "$= Ⓜ © 2☺", el sistema lo encuentra en la memoria (G5 417).
- El sistema genera 16 *tokens* (G5 423) según la Fig.5.
- Los 16 *tokens* se pasan (G5 424) a la red.
- La red pasó (G5 425) los *tokens* al lado del cliente.
- Los 16 *tokens* se muestran en la pantalla de inicio de sesión del usuario, como se muestra en la Fig. 12B, en una tabla de 4 x 4 (G5 320).
- El usuario selecciona (G5 426) los 6 *tokens*: G5 350, G5 353, G5 356, G5 359, G5 362 y G5 365.
- Los 6 *tokens* seleccionados por el usuario se pasan (G5 427) a la red después de que el usuario haga clic en "Iniciar sesión" (G5 370) en la Fig. 12B.
- Los 6 *tokens* seleccionados por el usuario se pasan (G5 428) al lado del servidor.
- En el lado del servidor, el sistema verifica los 6 *tokens* uno por uno: K11, K12, K13, K14, K15 y K16; en nuestro ejemplo, todos son correctos.
- El resultado anterior de un inicio de sesión exitoso (G5 429) se pasa (G5 430) a la red.
- La red pasa (G5 431) el resultado al lado del cliente y muestra (G5 432) un mensaje que se muestra en el G5 380 de la Fig. 12B.

[0098] La Figura 13C es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_5 en formato de texto. Muestra lo que sucede de forma oculta cuando se lleva a cabo el proceso de inicio de sesión de usuario de la figura 12C y muestra cómo se ve el flujo de proceso de la figura 4 en una realización de muestra. Esta es solo una demostración y no se muestra durante el inicio de sesión del usuario en tiempo real. Se utiliza para ilustrar visualmente las reglas de validación de *tokens* que se muestran en la Fig.7.

[0099] Hay 3 columnas: "Lado del cliente" (lado izquierdo), "Conexión de red" (centro) y "Lado del servidor" (lado derecho). El proceso comienza desde el lado del cliente cuando el usuario introduce la identificación del usuario, luego la información pasa a través de la conexión de red al lado del servidor. El servidor genera 16 *tokens* y los pasa a la red, luego los *tokens* se pasan al lado del cliente.

[0100] El usuario selecciona los *tokens* de acuerdo con las reglas de selección de *tokens* que se muestran en la Fig.6. Los *tokens* seleccionados se pasan a la red, luego se pasan al lado del servidor para ser validados, el resultado de otorgar o denegar el acceso se pasa a través de la red a el lado del cliente. El flujo del proceso está marcado con flechas en la Fig. 13C. A continuación se proporciona una explicación más detallada.

- El usuario introduce una identificación de usuario: "admin" (G5 506).
- El usuario hace clic en el botón "Entrar" (G5 509).
- La identificación de usuario "admin" (G5 506) se muestra en el lado del cliente (G5 511) y se pasa (G5 521) a la conexión de red (G5 513), luego se pasa nuevamente (G5 522) al lado del servidor (G5 515)).
- En el lado del servidor, el sistema verifica su memoria para ver si existe la identificación de usuario "admin", si no, el sistema mostrará un mensaje: "La identificación de usuario no existe, introduzca una identificación de usuario válida" (no se muestra). Se utiliza el mismo ejemplo de la Fig. 11B, el código de acceso en la memoria es: "$= Ⓜ © 2☺", el sistema lo encuentra en la memoria (G5 517).
- El sistema genera 16 *tokens* (G5 523) de acuerdo con las reglas de generación de *tokens* que se muestran en la Fig.5.
- Los 16 *tokens* se pasan (G5 524) a la red.
- La red pasa (G5 525) los *tokens* al lado del cliente.
- Los 16 *tokens* se muestran en la pantalla de inicio de sesión del usuario, como se muestra en la Fig. 12C, en una tabla de 4 x 4 (G5 321 en la Fig. 12C).
- El usuario selecciona (G5 526) los 6 *tokens*: G5 351, G5 354, G5 357, G5 360, G5 363 y G5 366.
- Los 6 *tokens* seleccionados por el usuario se pasan (G5 527) a la red después de que el usuario hace clic en "Iniciar sesión" (G5 371 en la Fig. 12C).
- Los 6 *tokens* seleccionados por el usuario se pasan (G5 528) al lado del servidor.
- En el lado del servidor, el sistema verifica los 6 *tokens* uno por uno: K21, K22, K23, K24, K25 y K26. En este ejemplo, los últimos 3 *tokens* seleccionados son incorrectos (el usuario necesitaba seleccionar el *token* D2, el *token* C3 y el *token* D1 para los 4°, 5° y 6° *tokens*, respectivamente, pero en su lugar el usuario

seleccionó el *token* C2, el *token* B2 y el *token* C4, que son incorrectos. Por lo tanto, el resultado es un inicio de sesión fallido).

- El resultado anterior de un inicio de sesión fallido (G5 529) se pasa (G5 530) a la red.
- La red pasa (G5 531) el resultado al lado del cliente y muestra (G5 532) un mensaje que se muestra en G5 381 de la Fig. 12C.

[0101] La figura 13D es una captura de pantalla de muestra del proceso de inicio de sesión del usuario en la realización GATE_5 en formato de texto. Muestra lo que sucede entre bastidores cuando se lleva a cabo el proceso de inicio de sesión del usuario de la figura 12D y muestra cómo se ve el flujo del proceso de la figura 4 en una realización de muestra. Esta es solo una demostración y no se muestra durante el inicio de sesión del usuario en tiempo real. Se utiliza para ilustrar visualmente las reglas de validación de *tokens* que se muestran en la Fig.7.

[0102] Hay 3 columnas: "Lado del cliente" (lado izquierdo), "Conexión de red" (centro) y "Lado del servidor" (lado derecho). El proceso comienza desde el lado del cliente cuando el usuario introduce la identificación del usuario, luego la información pasa a través de la conexión de red al lado del servidor. El servidor genera 16 *tokens* y los pasa a la red, luego los *tokens* se pasan al lado del cliente.

[0103] El usuario selecciona los *tokens* de acuerdo con las reglas de selección de *tokens* que se muestran en la figura 6, luego los *tokens* seleccionados se pasan a la red y luego al lado del servidor para ser validados. El resultado de otorgar o denegar el acceso se transmite a través de la red al lado del cliente. El flujo del proceso está marcado por flechas en la Fig. 13D. El proceso procede de la siguiente manera:

- El usuario introduce la identificación de usuario "admin" (G5 606).
- El usuario hace clic en "Enter" (G5 609).
- La identificación de usuario "admin" (G5 606) se muestra en el lado del cliente (G5 611) y se pasa (G5 621) a la conexión de red (G5 613), luego se pasa nuevamente (G5 622) al lado del servidor (G5 615)).
- En el lado del servidor, el sistema verifica su memoria para ver si existe la identificación de usuario "admin", si no lo hace, el sistema mostrará un mensaje: "La identificación de usuario no existe, introduzca una identificación de usuario válida" (no se muestra). Se utiliza el mismo ejemplo de la Fig. 11B, por lo que el código de acceso en la memoria es: "$= Ⓜ © 2☺", el sistema lo encuentra en la memoria (G5 617).
- El sistema genera 16 *tokens* (G5 623) de acuerdo con las reglas de generación de *tokens* de la Fig.5.
- Los 16 *tokens* se pasan (G5 624) a la red.
- La red pasó (G5 625) los *tokens* al lado del cliente.
- Los 16 *tokens* se muestran en la pantalla de inicio de sesión del usuario, como se muestra en la Fig. 12D, en una tabla de 4 x 4 (G5 322).
- El usuario selecciona (G5 626) los 5 *tokens*: G5 352, G5 355, G5 358, G5 361 y G5 364. Fíjese en G5 367, dice "[☺] falta la entrada", es decir, el sistema espera el símbolo "☺" como el último pin, por lo tanto, debería haber un sexto *token* y, sin embargo, falta la entrada del sexto *token*, esto es un error y el sistema negará el acceso del usuario.
- Los 5 *tokens* seleccionados por el usuario se pasan (G5 627) a la red después de que el usuario haga clic en "Iniciar sesión" (G5 372) en la Fig. 12D.
- Los 5 *tokens* seleccionados por el usuario se pasan (G5 628) al lado del servidor.
- En el lado del servidor, el sistema verifica los 5 *tokens* uno por uno: K31, K32, K33, K34 y K35, en nuestro ejemplo, todos los 5 *tokens* son correctos y, sin embargo, falta el sexto *token* (el usuario no hizo clic en C2, y en su lugar el usuario dejó la última posición en blanco y solo introdujo 5 pines), por lo tanto, K36 obtuvo una marca [x] que indica un error. Esto está mal. Por lo tanto, el resultado es un inicio de sesión fallido.
- El resultado anterior de un inicio de sesión fallido (G5 629) se transmite (G5 630) a la red.
- La red pasa (G5 631) el resultado al lado del cliente y muestra (G5 632) un mensaje que se muestra en G5 382 de la Fig. 12D.

[0104] El método se puede ampliar aún más para usar la siguiente función para hacer que adivinar el código de acceso sea aún más difícil: asigne cierto número de pines para que estén "ocultos", de modo que cuando esos pines ocultos aparezcan en la tabla de selección, no deban ser seleccionados. En cambio, el usuario debe seleccionar cualquier otro *token* que no tenga estos pines y evitar los *tokens* que los tengan.

[0105] Entonces, por ejemplo, si el usuario tiene la contraseña "123 ($#) 456", la longitud del código de acceso es 8, los 2 pines en el medio son pines ocultos que se muestran entre "(" y ")". Para los pines 1, 2, 3, 4, 5, 6 se

siguen las reglas anteriores para "$" y "#", se siguen las "reglas del pin oculto", que son: si ninguno de ellos aparece en ninguno de los 16 *tokens*, el usuario puede y debe seleccionar un *token* comodín en su lugar, pero si alguno de ellos aparece en una de las 16 *tokens* de la tabla de selección, el usuario debe evitar ese pin y seleccionar una de las otras 15 *tokens*.

[0106] Para convertir un pin en un pin oculto, se puede mostrar una casilla de verificación debajo de cada pin en la pantalla de creación (registro) del código de acceso, de modo que cuando el usuario marca una casilla debajo de un pin, ese pin se convierte en un pin oculto y, durante la validación del *token* proceso, utilice las reglas de pin ocultas anteriores para validar el inicio de sesión del usuario.

[0107] La presente invención también se puede utilizar en cualquier proceso de comunicación para adjuntar una tabla de selección con 16 *tokens* [generados con un código de acceso del emisor siguiendo las reglas de generación de *tokens* descritas en la Fig. 5] y una clave con algunos *tokens* junto con un mensaje. Si esa clave es válida en relación aesa tabla, entonces ese mensaje es un mensaje verdadero. Si la clave no es válida para la tabla de selección, entonces el mensaje adjunto es un mensaje falso. Manteniendo el mensaje verdadero y descartando el mensaje falso, se obtendrá el mensaje final [original] correcto. el receptor usa el mismo código de acceso para descifrar el mensaje, por lo que el proceso puede continuar como en los siguientes ejemplos:

- Message_1: Me iré a casa a las 7 pm [+ tabla de *tokens* con 16 *tokens* + clave inválida] -> Eliminar mensaje
  Message_2: Regresaré a casa a las 3 pm [+ tabla de *tokens* con 16 *tokens* + clave válida] ==> Regresaré a casa a las 3 pm
  Message_3: Abandonaremos el ataque el día 3 [+ tabla de *tokens* con 16 *tokens* + clave inválida] -> Eliminar mensaje
  Message_4: Atacaremos al mediodía del día 3 [+ tabla de *tokens* con 16 *tokens* + clave válida] ==> Atacaremos al mediodía del día 3.
  Por lo tanto, el mensaje final correcto es: Me iré a casa a las 3 pm. Atacaremos al mediodía del día 3.
- Message_1: Voy [+ tabla de *tokens* con 16 *tokens* + clave válida] ==> Voy
  Message_2: a [+ tabla de *tokens* con 16 *tokens* + clave válida] ==> a
  Message_3: pasar de [+ tabla de *tokens* con 16 *tokens* + clave no válida] -> Eliminar mensaje
  Message_4: ir [+ tabla de *tokens* con 16 *tokens* + clave válida] ==> ir
  Por tanto, el mensaje final correcto es: voy a ir.
- Message_1: n [+ tabla de *tokens* con 16 *tokens* + clave no válida] -> Eliminar mensaje
  Message_2: o [+ tabla de *tokens* con 16 *tokens* + clave no válida] -> Eliminar mensaje
  Message_3: v [+ tabla de *tokens* con 16 *tokens* + clave válida] ==> v
  Message_4: a [+ tabla de *tokens* con 16 *tokens* + clave válida] ==> a
  Message_5: l [+ tabla de *tokens* con 16 *tokens* + clave válida] ==> l
  Message_6: e [+ tabla de *tokens* con 16 *tokens* + clave válida] ==> e
  Por lo tanto, el mensaje final correcto es: vale.

[0108] La presente invención también puede vencer el *phishing*, que es un intento de adquirir información confidencial como nombres de usuario, contraseñas y detalles de tarjetas de crédito, a menudo por razones maliciosas, haciéndose pasar por una entidad fiable en una comunicación electrónica. Debido a que, con la presente invención, toda la información que se pasa entre el usuario y el servidor está en forma de *tokens*, y cada *token* tiene múltiples símbolos, no se deja descubrir ningún pin de usuario claro.

[0109] La figura 14 es una captura de pantalla de muestra de un proceso de cifrado de mensajes que utiliza la realización GATE_4. Muestra un ejemplo de cómo se cifra un mensaje de texto sin formato con una contraseña de emisor y cómo puede verse el mensaje cifrado. El proceso procede de la siguiente manera:

- El emisor del mensaje introduce un mensaje de texto sin formato "secreto" en G4 700.
- El emisor introduce un código de acceso "123" en G4 702 y hace clic en el botón "Encriptar" [G4 703].
- El mensaje original "secreto" se mezcla con algunos caracteres de relleno aleatorios y se convierte en el siguiente mensaje de resultado "sLeQWNcrMfYeMtHQr" como se muestra en G4 704.
- El receptor utilizará el mismo código de acceso "123" [G4 706] en el lado del receptor para descifrar el mensaje.

[0110] La figura 15A es una captura de pantalla de muestra de un proceso de descifrado de mensajes utilizando la realización GATE_4. Muestra un ejemplo de cómo un mensaje original cifrado se puede descifrar con éxito con un código de acceso del receptor. El proceso procede de la siguiente manera:

- Cada carácter en el mensaje de resultado [G4 704] se adjunta con una tabla de *tokens* de 4 x 4, cada *token* en la tabla tiene 4 símbolos, cada carácter en el mensaje también se adjunta a una "clave", la clave tiene algunos *tokens* en ella, el número de *tokens* en la clave puede oscilar entre 2 y 6.
- El receptor usa el mismo código de acceso "123" [G4 706] para descifrar el mensaje, el mensaje descifrado se muestra en G4 704 como caracteres resaltados. El resultado del mensaje descifrado es "secreto" [G4 708].
- La figura 15A muestra un ejemplo de cómo podría verse para cada carácter. En la captura de pantalla, el primer carácter del mensaje "s" [G4 710] se muestra como ejemplo después de que el usuario haga clic en él [G4 710], G4 712 muestra que el carácter de visualización actual es "s". La tabla de *tokens* de 4 x 4 adjunta a este carácter se muestra en la tabla G4 714.
- Los *tokens* de clave adjuntos a "s" también se muestran como G4 720, G4 722 y G4 724.
- Los caracteres de relleno en el mensaje: L, Q, W, N, M, f, Y, M, H, Qyr se adjuntan intencionalmente con tablas de *tokens* y claves que no son válidas, por lo que se invalidarán en el lado del receptor.
- En el ejemplo que se muestra, el usuario puede hacer clic en cada carácter en G4 704 para mostrar su contenido y *tokens* de clave, luego hacer clic en el botón "Verificar" [G4 726] para ver si el carácter es válido. En la captura de pantalla, muestra que el carácter "s" es válido y que la verificación fue exitosa [G4 730].

[0111] La figura 15B es una captura de pantalla de muestra de un proceso de descifrado de mensajes usando la realización GATE_4. Muestra lo que sucede de forma oculta cuando se lleva a cabo el proceso mostrado en la Fig. 15A y cómo se valida un mensaje en el lado del receptor. El proceso procede de la siguiente manera:

- El carácter de mensaje "s" [G4 750] se cifra con el código de acceso del emisor "123" [G4 752] y se adjunta con 16 *tokens* [G4 754] y una clave con algunos *tokens* [G4 720, G4 722 y G4 724]. Esta información se envía a la red [G4 756], luego al receptor [G4 758].
- En el lado del receptor, se utiliza la misma contraseña "123" [G4 760] para decodificar el mensaje. Los *tokens* de clave pasan por [G4 762] el proceso de validación G4 764, G4 766, para verificar cada *token* de clave. De C51, C52 y C53, se puede ver que todos son válidos, por lo que se llega a una conclusión final [G4 768] de que el mensaje es válido [G4 770], como se muestra en la Fig. 15A [G4 730].

[0112] La figura 16A es una captura de pantalla de muestra de un proceso de descifrado de mensajes utilizando la realización GATE_4. Muestra un ejemplo de cómo un mensaje de relleno cifrado puede ser descifrado y reconocido como información no válida por un código de acceso del receptor. El proceso procede de la siguiente manera:

- La figura 16A muestra un ejemplo de cómo podría verse para cada carácter de relleno que no es parte del mensaje original. Muestra el contenido del segundo carácter "L" [G4 711] en el mensaje [G4 704] después de que el usuario hace clic en él. G4 713 muestra que el carácter es "L". La tabla de 4 x 4 de 16 *tokens* adjunta a este carácter se muestra en G4 715. Los 4 *tokens* de clave adjuntos a "L" se muestran como G4 721, G4 723, G4 725 y G4 727.
- Dado que el carácter "L" es un carácter de relleno y no es parte del mensaje original, el emisor lo adjuntó intencionalmente con una tabla de *tokens* de 4 x 4 y una clave que no validaría. Se puede ver claramente que la contraseña del emisor [G4 702] y la contraseña del receptor [G4 706] son iguales y tienen 3 pines "1", "2" y "3", por lo que una clave válida no debe tener más ni menos que 3 *tokens*. En este ejemplo, tiene 4 *tokens* de clave, por lo que no es válido y el carácter "L" debe ignorarse y no formaría parte del mensaje descifrado final.
- En la captura de pantalla, cuando el usuario hace clic en el botón "Verificar" G4 726, muestra que el proceso de validación ha fallado [G4 731].

[0113] La figura 16B es una captura de pantalla de muestra de un proceso de descifrado de mensajes utilizando la realización GATE_4. Muestra lo que sucede cuando tiene lugar el proceso mostrado en la figura 16A y cómo se invalida un mensaje de relleno en el lado del receptor. El proceso procede de la siguiente manera:

- El carácter de mensaje "L" [G4 751] está encriptado con el código de acceso del emisor "123" [G4 752] y se adjunta con una tabla de 4 x 4 de 16 *tokens* [G4 755] y una clave con algunos *tokens* [G4 721, G4 723, G4 725 y G4 727]. La información anterior se envía a la red [G4 757] y luego al receptor [G4 759].
- En el lado del receptor, se utiliza la misma contraseña "123" [G4 760] para decodificar el mensaje. Los *tokens* de clave pasan por [G4 763] el proceso de validación G4 765, G4 767, para verificar cada *token* de clave, desde C61, C62, C63 y C64. Se puede ver que los últimos 2 *tokens* de clave no son válidos.
- El tercer código de acceso "3" apareció en el tercer *token* [G4 790] y debe seleccionarse. Sin embargo, el octavo *token* [G4 792] fue seleccionado y apareció en el tercer *token* de clave G4 725. Esto es incorrecto.
- El código de acceso del emisor es igual al código de acceso del receptor y tiene 3 pines, pero la clave adjunta tiene 4 *tokens*. El último *token* [G4 727] tampoco es válido.

- Se llega a una conclusión final [G4 769] de que el mensaje no es válido [G4 771], como se muestra en la Fig. 16A [G4 731].

[0114] La figura 17 es una captura de pantalla de muestra de un proceso de descifrado de mensajes utilizando la realización GATE_4. Muestra un ejemplo de cómo un mensaje original cifrado puede no ser descifrado correctamente por un código de acceso del receptor que es diferente del código de acceso del emisor. El proceso procede de la siguiente manera:

- El usuario escribió un mensaje de texto sin formato "secreto" en G4 700, luego introdujo la contraseña "123" [G4 702] y hizo clic en el botón "Encriptar" [G4 703].
- El mensaje se encripta, se envía y se recibe y se muestra en G4 705 como: "sLeQWNcrMfYeMtHQr" [G4 705].
- El receptor usa la contraseña "567" [G4 707] para descifrar el mensaje recibido del emisor, encriptado con la contraseña "123" [G4 702].
- El mensaje descifrado se resalta en G4 705: "ecrQ".
- El mensaje de resultado se muestra como "ecrQ" [G4 709].
- El mensaje de resultado es diferente del mensaje original del emisor: "secreto"
-

[G4 700], porque el receptor utilizó un código de acceso diferente para descifrar el mensaje.

[0115] La figura 18 es una captura de pantalla de muestra de un proceso de cifrado de mensajes que utiliza la realización GATE_5. Muestra un ejemplo de cómo se cifra un mensaje de texto sin formato con una contraseña de emisor y cómo puede verse el mensaje cifrado. El proceso procede de la siguiente manera:

- El emisor del mensaje introduce un mensaje de texto sin formato "FYEO" en G5 700.
- El emisor introduce una contraseña "123" en G5 702 y hace clic en el botón "Encriptar" [G5 703].
- El mensaje original "FYEO" se mezcla con algunos caracteres de relleno aleatorios y se convierte en el siguiente mensaje de resultado "F1PRojcYnEbAO" [G5 704].
- El receptor utilizará el mismo código de acceso "123" [G5 706] en el lado del receptor para descifrar el mensaje.

[0116] La figura 19A es una captura de pantalla de muestra de un proceso de descifrado de mensajes utilizando la realización GATE_5. Muestra un ejemplo de cómo un mensaje original cifrado se puede descifrar con éxito con un código de acceso del receptor. El proceso procede de la siguiente manera:

- Cada carácter del mensaje recibido [G5 704] se adjunta con una tabla de *tokens* de 4 x 4, cada *token* de la tabla tiene 5 símbolos, cada carácter del mensaje también se adjunta a una "clave", la clave tiene algunos *tokens*, el número de *tokens* en la clave puede oscilar entre 2 y 6.
- El receptor usa el mismo código de acceso "123" [G5 706] para decodificar el mensaje, el mensaje descifrado se muestra en G5 704 como caracteres resaltados. El resultado del mensaje descifrado es "FYEO" [G5 708].
- La Fig. 19A muestra un ejemplo de cómo podría verse para cada carácter. En la captura de pantalla, el primer carácter del mensaje "F" [G5 710] se muestra como ejemplo. G5 712 muestra que el carácter de visualización actual es "F", y la tabla de *tokens* de 4 x 4 adjunta a este carácter se muestra en la tabla G5 714.
- Los *tokens* de clave adjuntos a "F" también se muestran como G5 720, G5 722 y G5 724.
- Los caracteres de relleno en el mensaje: 1, P, R, o, j, c, n, b y A se adjuntan intencionalmente con tablas de *tokens* y claves que no son válidas, por lo que se invalidarán en el lado del receptor.
- En este ejemplo, el usuario puede hacer clic en cada carácter en G5 704 para mostrar su contenido y *tokens* de clave, luego hacer clic en el botón "Verificar" [G5 726] para ver si el carácter es válido. En la captura de pantalla, se muestra que el carácter "F" es válido y la verificación se realizó correctamente [G5 730].

[0117] La figura 19B es una captura de pantalla de muestra de un proceso de descifrado de mensajes utilizando la realización GATE_5. Muestra lo que sucede de forma oculta cuando tiene lugar el proceso mostrado en la figura 19A y cómo se valida un mensaje en el lado del receptor. El proceso procede de la siguiente manera:

- El carácter de mensaje "F" [G5 750] está encriptado con la contraseña del emisor "123" [G5 752] y adjunta con una tabla 4 x 4 de 16 *tokens* [G5 754] y una clave con algunos *tokens* [G5 720, G5 722 y G5 724]. Esta información se envía a la red [G5 756] y luego al receptor [G5 758].
- En el lado del receptor, utiliza el mismo código de acceso "123" [G5 760] para descifrar el mensaje. Los *tokens* de clave pasan por [G5 762] el proceso de validación G5 764, G5 766, para verificar cada *token*

de clave. De K51, K52 y K53 se puede ver que todos son válidos, por lo que se llega a una conclusión final [G5 768] de que el mensaje es válido [G5 770], como se muestra en la Fig. 19A [G5 730].

[0118] La figura 20A es una captura de pantalla de muestra de un proceso de descifrado de mensajes utilizando la realización GATE_5. Muestra un ejemplo de cómo un mensaje de relleno cifrado puede ser descifrado y reconocido como información no válida por un código de acceso del receptor. El proceso procede de la siguiente manera:

- La figura 20A muestra un ejemplo de cómo podría verse para cada carácter de relleno que no es parte del mensaje original. Muestra el contenido del tercer carácter "P" [G5 711] en el mensaje [G5 704]. G5 713 muestra que el carácter es "P", y la tabla 4 x 4 de 16 *tokens* adjunta a este carácter se muestra en G5 715. Los 3 *tokens* de clave adjuntos a "P" se muestran como G5 721, G5 723 y G5 725 .
- Dado que el carácter "P" es un carácter de relleno y no es parte del mensaje original, el emisor lo adjuntó intencionalmente con una tabla de *tokens* de 4 x 4 y una clave que no validaría, ya que se puede ver claramente que la contraseña del emisor [G5 702] y el código de acceso del receptor [G5 706] son iguales y tienen 3 pines "1", "2" y "3". El primer pin "1" en el código de acceso aparece en el último *token* en la tabla 4 x 4 [G5 716], y ese *token* debe seleccionarse como el primer *token* de clave. Sin embargo, se seleccionó el segundo *token* [G5 718] en la tabla, y se muestra en la posición del primer *token* de clave G5 721. Esto es incorrecto e invalidaría este mensaje.
- En la captura de pantalla, cuando el usuario hace clic en el botón "Verificar" G5 726, muestra que el proceso de validación ha fallado [G5 731] para este carácter "P".

[0119] La figura 20B es una captura de pantalla de muestra de un proceso de descifrado de mensajes utilizando la realización GATE_5. Muestra lo que sucede de forma oculta cuando tiene lugar el proceso mostrado en la Fig. 20A y cómo se invalida un mensaje de relleno en el lado del receptor. El proceso procede de la siguiente manera:

- El carácter de mensaje "P" [G5 751] está encriptado con el código de acceso del emisor "123" [G5 752] y se adjunta con una tabla 4 x 4 de 16 *tokens* [G5 755] y una clave con algunos *tokens* [G5 721, G5 723 y G5 725]. Esta información se envía a la red [G5 757] y luego al receptor [G5 759].
- En el lado del receptor, utiliza el mismo código de acceso "123" [G5 760] para descifrar el mensaje. Los *tokens* de clave pasan por [G5 763] el proceso de validación G5 765, G5 767, para verificar cada *token* de clave. Desde K61, K62 y K63 se puede ver que el primer *token* no es válido.
- El primer código de acceso "1" apareció en el último *token* [G5 790] y debe seleccionarse. Sin embargo, se seleccionó el segundo *token* [G5 792] y aparece en la primera posición del *token* de clave [G5 721]. Por tanto, es inválido.
- Se llega a una conclusión final [G5 769] de que el mensaje no es válido [G5 771], como se muestra en la Fig. 20A [G5 731].

[0120] La figura 21 es una captura de pantalla de muestra de un proceso de descifrado de mensajes utilizando la realización GATE_5. Muestra un ejemplo de cómo un mensaje original cifrado puede no ser descifrado correctamente por un código de acceso del receptor que es diferente del código de acceso del emisor. El proceso procede de la siguiente manera:

- El usuario escribió un mensaje de texto sin formato "FYEO" en G5 700, luego introdujo la contraseña "123" [G5 702] y presionó el botón "Encriptar" [G5 703].
- El mensaje se encripta, se envía y se recibe, y se muestra en G5 705 como: "F1PRojcYnEbAO" [G5 705].
- El receptor utiliza el código de acceso "680" [G5 707] para descifrar el mensaje recibido del emisor, que está cifrado con el código de acceso "123" [G5 702].
- El mensaje descifrado se resalta en G5 705: "nE".
- El mensaje de resultado se muestra como "nE" [G5 709].
- El mensaje de resultado es diferente del mensaje original del emisor: "FYEO" [G5 700], porque el receptor utilizó un código de acceso diferente para descifrar el mensaje.

[0121] Los siguientes pasos se utilizan preferiblemente, para cada pin en el código de acceso, para generar *tokens* de clave válidos en relación a una tabla de 4 x 4 con 16 *tokens* para cada mensaje válido:

- Revisar los 16 *tokens*: (a) si el pin se encuentra en un *token*, elegir ese *token*; y (b) si el pin no está en ningun *token*, elegir un *token* al azar de las 16 *tokens* de la tabla.

[0122] Los siguientes pasos se utilizan preferiblemente para generar *tokens* de clave no válidos en una tabla de 4 x 4 con 16 *tokens* para cada mensaje no válido:

<1> Establecer booleano "Done_Fixing" en falso
<2> Revisar los 16 *tokens*, seguir los pasos <3> y <4> a continuación para cada pin del código de acceso

<3>
<A> Si el pin se encuentra en un *token*:
(1) Si Done_Fixing es igual a falso, elegir cualquier otro *token* excepto este para elegir intencionalmente un *token* incorrecto y establecer Done_Fixing en verdadero.
(2) Si Done_Fixing es igual a verdadero, elegir ese *token*.
<B> Si el pin no está en ningun *token*, elegir un *token* aleatoria del 16.
<4> Guardar el *token* de clave generado anteriormente en un vector.
<5> Generar un número aleatorio N en el rango de: -1 a 1
<A> Si N = -1, borrar el último *token* de clave del vector.
<B> Si N = 0, no hacer nada.
<C> Si N = 1 y la longitud del pin de usuario <6, agregar un *token* aleatorio del 16 en la tabla al vector.
<6> Los *tokens* en el vector serán los *tokens* de clave finales.

[0123] Las realizaciones y ventajas anteriores son meramente ilustrativas y no deben interpretarse como limitantes de la presente invención. La descripción de la presente invención pretende ser ilustrativa y no limitar el alcance de las reivindicaciones. A los expertos en la técnica les resultarán evidentes muchas alternativas, modificaciones y variaciones. Pueden realizarse varios cambios sin apartarse del alcance de la invención, como se define en las siguientes reivindicaciones.

[0124] Por ejemplo, aunque la presente invención se ha descrito en relación con las realizaciones GATE_4 y GATE_5, en las que se utilizan 4 dimensiones y 5 dimensiones de símbolos, respectivamente, se puede utilizar cualquier número de dimensiones (incluyendo sólo una dimensión) sin dejar de estar dentro del alcance de la presente invención. En general, siempre que cada *token* tenga más de un símbolo, se puede utilizar cualquier número de símbolos categorizados en cualquier número de dimensiones. Además, las realizaciones de GATE_4 y GATE_5 descritas anteriormente, así como las capturas de pantalla asociadas, pretenden ser ilustrativas y no limitar el alcance de la presente invención.

REIVINDICACIONES

1. Método para autenticar a un usuario mediante el uso de un código de acceso predeterminado almacenado electrónicamente que comprende un número predeterminado de símbolos de código de acceso seleccionados de un conjunto de símbolos preseleccionados, en el que cada uno de los símbolos de código de acceso está **caracterizado por** una posición predeterminada del pin, método que comprende:

presentar un conjunto de *tokens* al usuario a través de una interfaz de usuario (120) de un dispositivo electrónico (150), en el que el conjunto de *tokens* comprende al menos dos *tokens*, y en el que cada *token* del conjunto de *tokens* comprende al menos dos símbolos que pertenecen al conjunto de símbolos;
requerir que el usuario seleccione un *token* del conjunto de *tokens* para cada posición de pin en el código de acceso a través de la interfaz de usuario; y
autenticar al usuario en base a los *tokens* que el usuario seleccionó, en donde el usuario se autentica si:

el número de *tokens* seleccionados por el usuario es igual al número de símbolos en el código de acceso,
al menos uno de los *tokens* seleccionados contiene uno de los símbolos de código de acceso respectivos, y
para cada posición de pin, el símbolo de código de acceso en dicha posición de pin está comprendido en el *token* seleccionado para esta posición de pin o no está comprendido en absoluto en ninguno de los *tokens* del conjunto de *tokens* presentado al usuario.

2. Método de la reivindicación 1, en el que el número de símbolos en el conjunto de símbolos es igual al número de *tokens* presentados al usuario

3. Método de la reivindicación 1, en el que el número de símbolos en el conjunto de símbolos es mayor que el número de *tokens* presentados al usuario.

4. Método de la reivindicación 1, en el que el conjunto de símbolos se divide en al menos dos subconjuntos, en lo sucesivo denominados dimensiones, y cada *token* comprende un símbolo de cada una de las al menos dos dimensiones.

5. Método de la reivindicación 1, en el que cada *token* comprende cuatro símbolos que pertenecen al conjunto de símbolos, y opcionalmente
en el que cada conjunto de símbolos se divide en cuatro subconjuntos, en lo sucesivo denominados dimensiones, y cada *token* comprende un símbolo de cada dimensión de símbolos.

6. Método de la reivindicación 1, en el que cada *token* comprende cinco símbolos que pertenecen al conjunto de símbolos, y opcionalmente
en el que cada conjunto de símbolos se divide en cinco subconjuntos, en lo sucesivo denominados dimensiones, y cada *token* comprende un símbolo de cada dimensión de símbolos.

7. Método de la reivindicación 1, en el que el conjunto de símbolos se basa en el sistema Unicode.

8. Sistema para autenticar a un usuario mediante el uso de un código de acceso predeterminado almacenado electrónicamente que comprende un número predeterminado de símbolos de código de acceso seleccionados de un conjunto de símbolos preseleccionado, en el que cada uno de los símbolos de código de acceso es **caracterizado por** una posición predeterminada del pin, sistema que comprende:

un procesador;
memoria accesible por el procesador; y
un módulo de autenticación/encriptación (110) que comprende un conjunto de instrucciones legibles por ordenador almacenadas en la memoria que son ejecutables por el procesador para:

presentar un conjunto de *tokens* al usuario, en el que el conjunto de *tokens* comprende al menos dos *tokens*, y en el que cada *token* en el conjunto de *tokens* comprende al menos dos símbolos que pertenecen al conjunto de símbolos;
requerir que el usuario seleccione un *token* del conjunto de *tokens* para cada posición de pin en el código de acceso a través de una interfaz de usuario (120); y
autenticar al usuario en base a los *tokens* que el usuario seleccionó, en donde el procesador determina que el usuario está autenticado si:

el número de *tokens* seleccionados por el usuario es igual al número de símbolos en el código de acceso,
al menos uno de los *tokens* seleccionados contiene uno de los símbolos de código de acceso respectivos, y

para cada posición de pin, el símbolo de código de acceso en dicha posición de pin está comprendido en el *token* seleccionado para esta posición de pin o no está comprendido en absoluto en ninguno de los *tokens* del conjunto de *tokens* presentado al usuario.

9. Sistema de la reivindicación 8, en el que el número de símbolos en el conjunto de símbolos es igual al número de *tokens* presentados al usuario, o
en el que el número de símbolos en el conjunto de símbolos es mayor que el número de *tokens* presentados al usuario.

10. Sistema de la reivindicación 8, en el que el conjunto de símbolos se divide en al menos dos subconjuntos, denominados en lo sucesivo dimensiones, y cada *token* comprende un símbolo de cada una de las al menos dos dimensiones.

11. Sistema de la reivindicación 8, en el que cada *token* comprende cuatro símbolos que pertenecen al conjunto de símbolos y, opcionalmente, cada conjunto de símbolos se divide en cuatro subconjuntos, en lo sucesivo denominados dimensiones, y cada *token* comprende un símbolo de cada dimensión de símbolos.

12. Sistema de la reivindicación 8, en el que cada *token* comprende cinco símbolos que pertenecen al conjunto de símbolos y, opcionalmente,
en el que cada conjunto de símbolos se divide en cinco subconjuntos, en lo sucesivo denominados dimensiones, y cada *token* comprende un símbolo de cada dimensión de símbolos.

13. Sistema de la reivindicación 8, en el que el conjunto de símbolos se basa en el sistema Unicode.

Fig. 1A

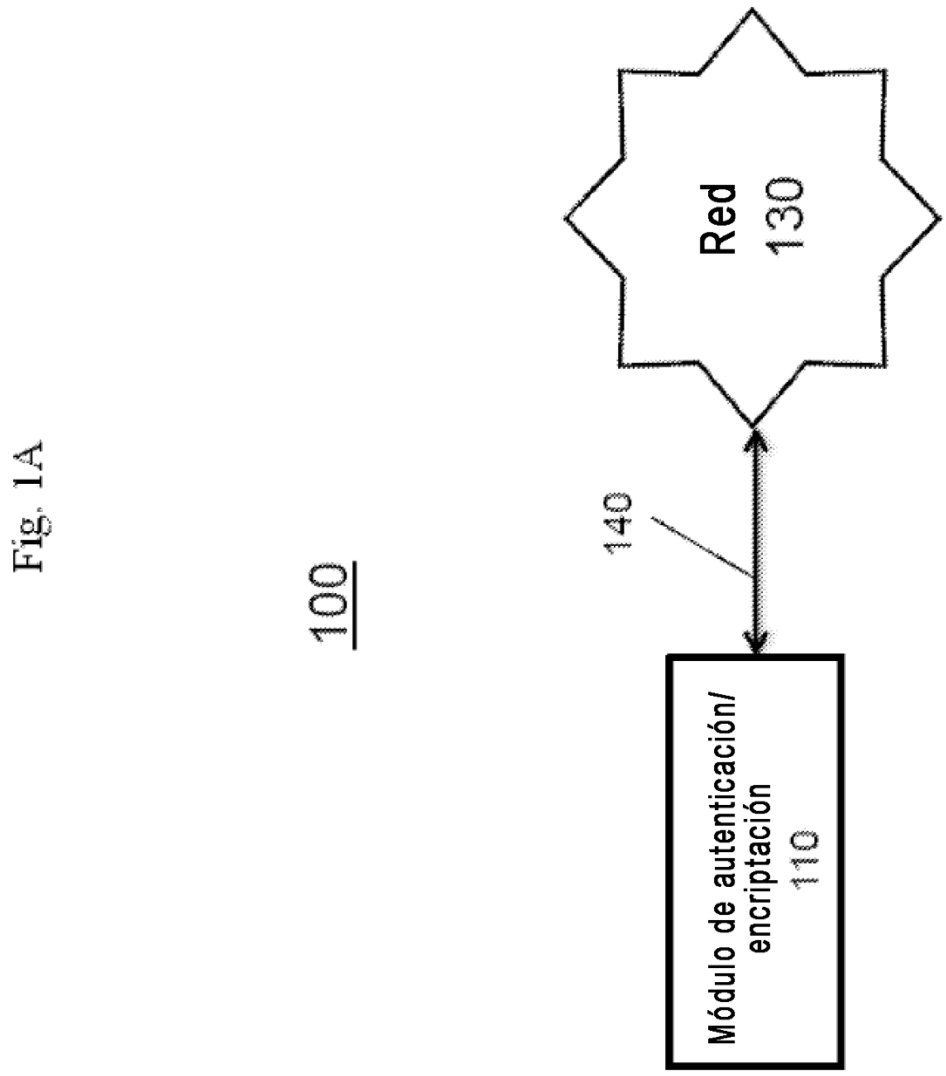


Fig. 1B

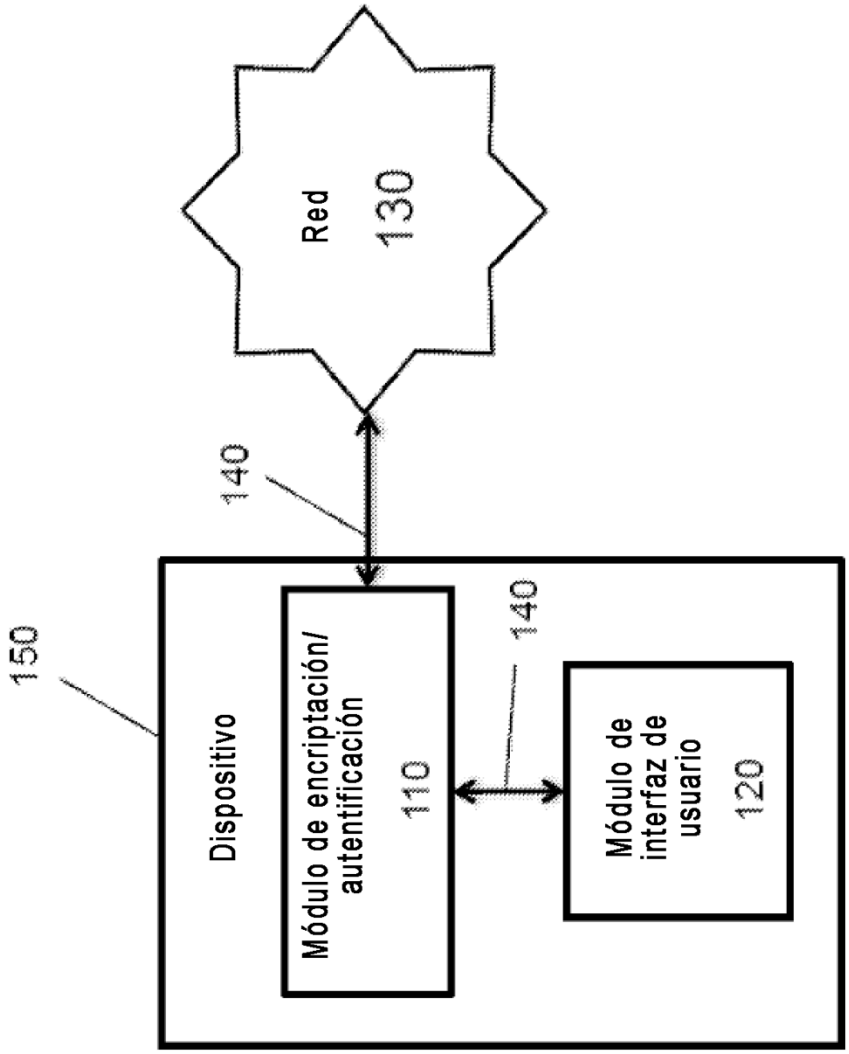

Red
130
140
150
Dispositivo
Módulo de encriptación/ autentificación
110
140
Módulo de interfaz de usuario
120

Fig. 1C

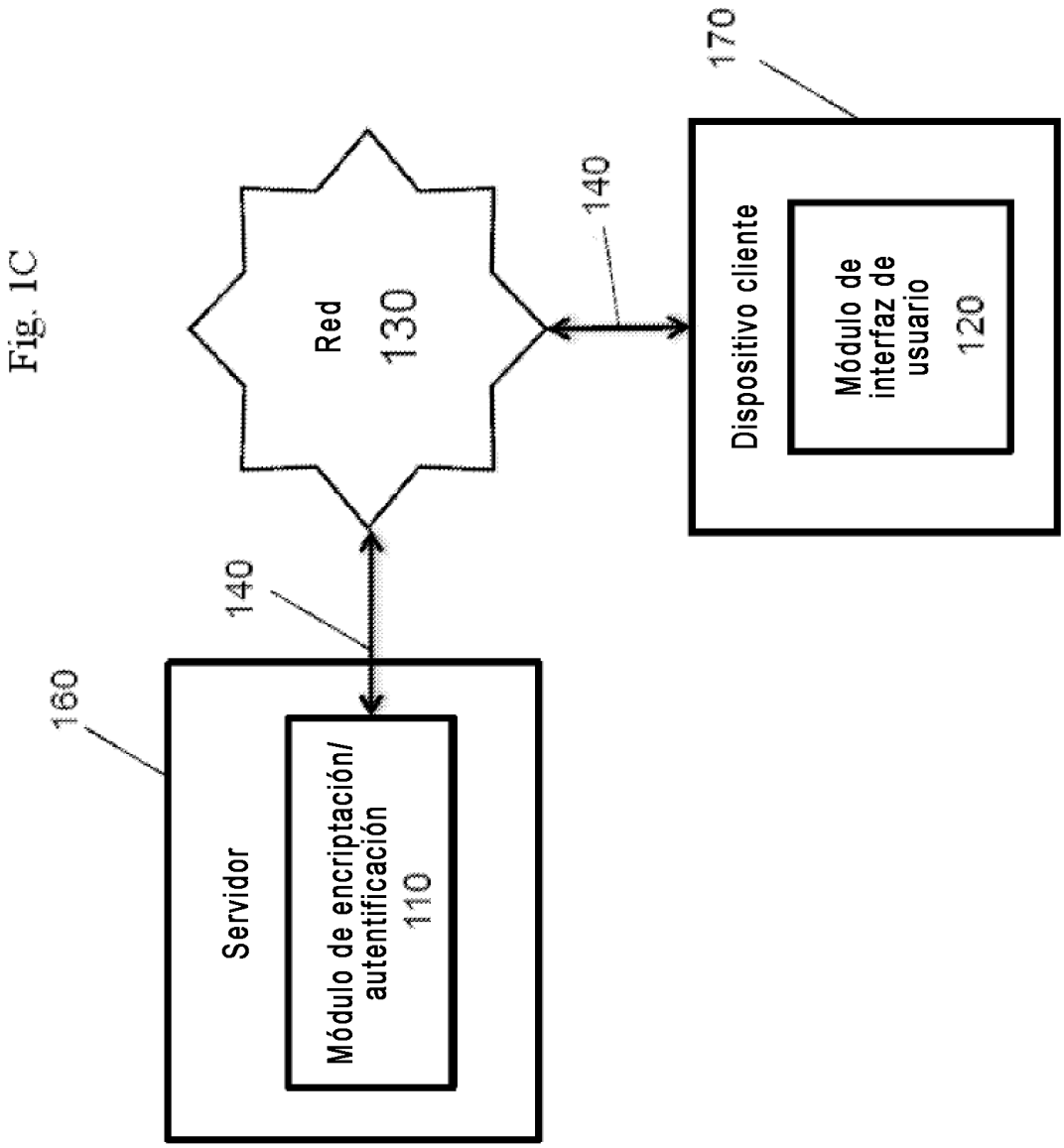

170
Dispositivo cliente
Módulo de interfaz de usuario
120
140
Red
130
140
160
Servidor
Módulo de encriptación/ autentificación
110

Fig. 1D

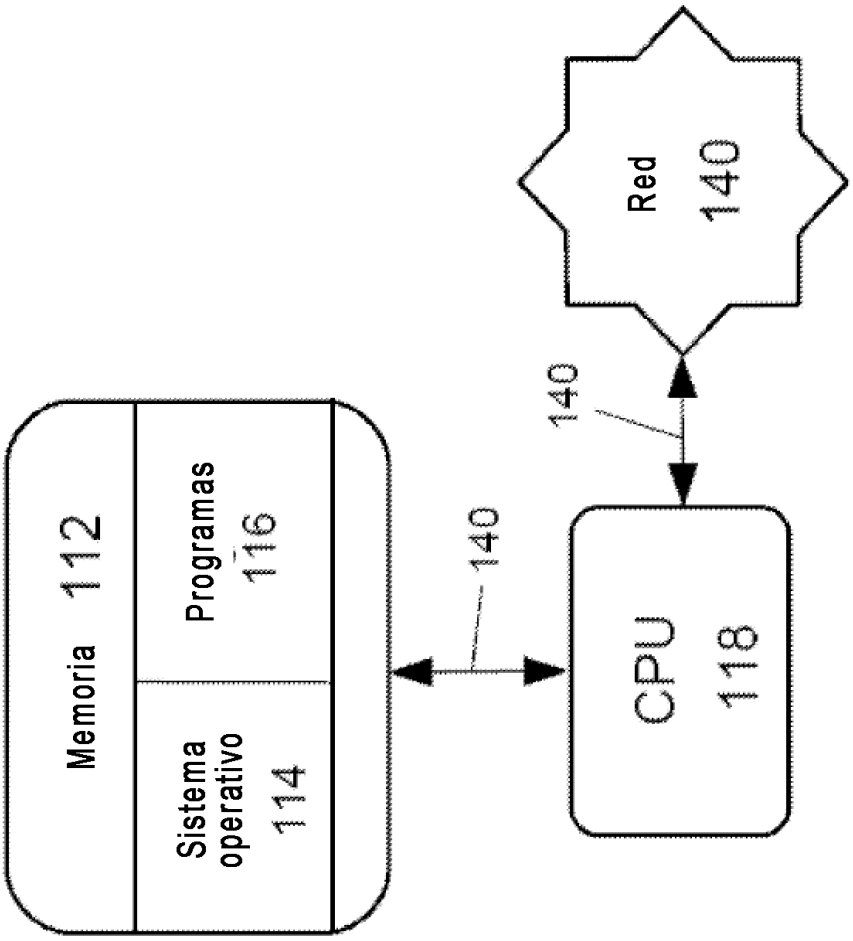

Red
140
140
Memoria 112
Programas
116
Sistema
operativo
114
140
CPU
118

Fig. 2A

(GATE_4)

1ª Dimensión

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36

2ª Dimensión

Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ Ⓚ Ⓛ Ⓜ Ⓝ Ⓞ Ⓟ Ⓠ Ⓡ Ⓢ Ⓣ Ⓤ Ⓥ Ⓦ Ⓧ Ⓨ Ⓩ ~ ! @ # % ^ & * : ?

3ª Dimensión

○ ● △ ▲ □ ■ ☆ ★ ♤ ♠ ♧ ♣ ♡ ♥ ♢ ♦ ☾ ☪ ☹ ☻ ♂ ♀ ∵ ∴ = ≠ ← → ↑ ↓ ✓ X ☼ ☂ ☾ ☤

4ª Dimensión

+ − × ÷ © ® ✂ ✉ ☏ ⁙ ♫ ♨ ☮ ☯ ✡ ❋ ☣ ☨ ☸ ⛀ ☢ ∑ ∞ π λ ψ ☜ ☞ ☝ ☟ $ ¥ € £ ₣ ₪

Fig. 2B

(GATE_5)

1ª Dimensión

Ⓐ Ⓑ Ⓒ Ⓓ Ⓔ Ⓕ Ⓖ Ⓗ Ⓘ Ⓙ Ⓚ Ⓛ Ⓜ Ⓝ Ⓞ Ⓟ Ⓠ Ⓡ Ⓢ Ⓣ Ⓤ Ⓥ Ⓦ Ⓧ Ⓨ Ⓩ

2ª Dimensión

α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ ς σ τ υ φ χ ψ ω ϟ

3ª Dimensión

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26

4ª Dimensión

○ ● △ ▲ □ ■ ☆ ★ ♤ ♠ ♧ ♣ ♡ ♥ ♢ ♦ ☺ ☻ ♂ ♀ ☂ ☼ ∵ ∴ = ≠

5ª Dimensión

+ − × ÷ ← → ↑ ↓ ✂ ✉ ✌ ⁙ ♫ ☮ ☯ ✡ ☾ ☤ ☣ ☐ ☢ $ ¥ € £ ₣ ₪

Fig. 3

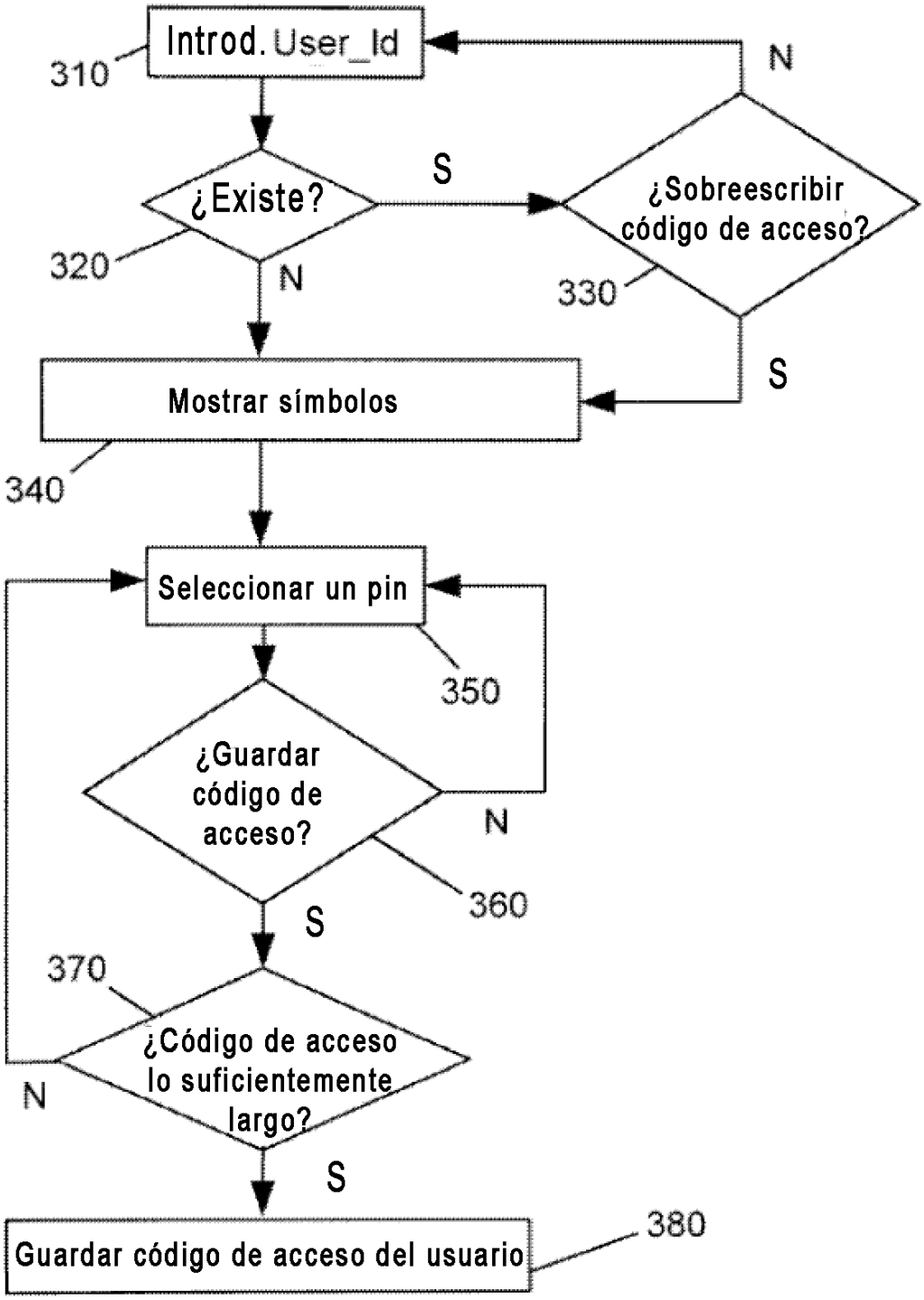

Introd. User_Id
310
¿Existe?
320
S
N
¿Sobreescribir código de acceso?
330
N
S
Mostrar símbolos
340
Seleccionar un pin
350
¿Guardar código de acceso?
N
360
S
370
¿Código de acceso lo suficientemente largo?
N
S
Guardar código de acceso del usuario
380

Fig. 4

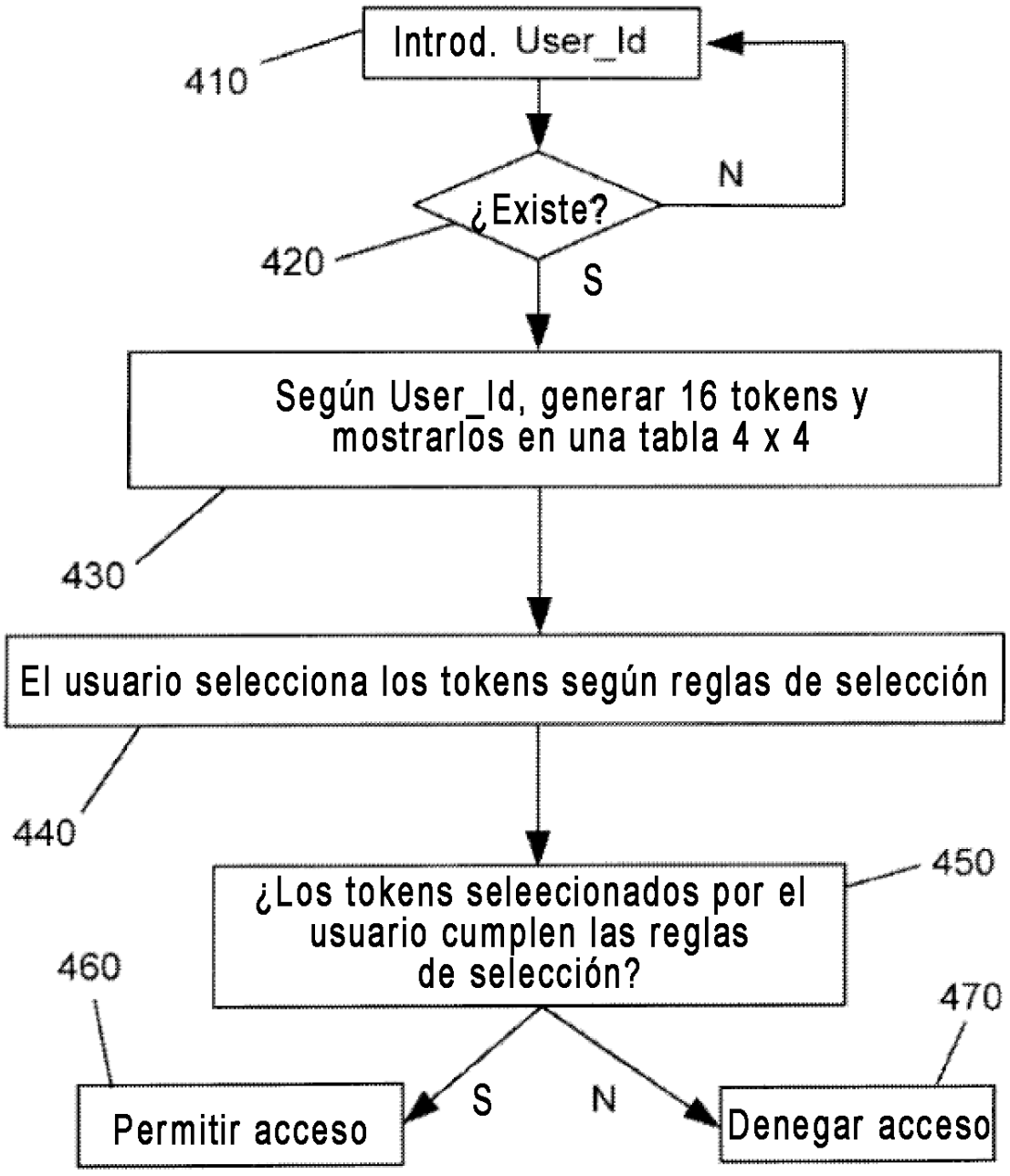

Introd. User_Id
410
¿Existe?
N
420
S
Según User_Id, generar 16 tokens y mostrarlos en una tabla 4 x 4
430
El usuario selecciona los tokens según reglas de selección
440
¿Los tokens seleecionados por el usuario cumplen las reglas de selección?
450
460
470
S
N
Permitir acceso
Denegar acceso

Fig. 5

(Reglas de generación de tokens)

[Con fines ilustrativos, se usa la realización GATE_4 (véase Fig. 2A)]

El objetivo es generar 16 tokens sin símbolos redundantes entre ellos, donde cada token tendrá 4 símbolos, uno de cada dimensión, a saber:Token_1(2 ♡ +), Token_2(30 ? ♦ λ), ... , Token_16(16 @ ✓ ☻)

Al menos un pin* de usuario debe estar en uno de los 16 tokens, posiblemente más, incluso todos.

Obsérvese el orden de los símbolos anteriores, el 1º siempre es de la 1ª_Dim, el 2º es de la 2ª_Dim...

Reglas y Pasos	**Ejemplos de resultados**
[1] Crear 4 Vectores, cada uno que contiene 36 símbolos de cada una de las 4 dimensiones descritas en la FIG 2: Available_Symbols para GATE_4. Cada vez que un símbolo se elimina de un Vector, su tamaño se reducirá en uno. El motivo de eliminar símbolos de los vectores es para evitar la duplicación. Se hará referencia a los vectores anteriores como vectores dimensionales: V_1, V_2, V_3, V_4. Cada paso de generación de token eliminará un símbolo restante de cada uno de los 4 vectores anteriores, para que no haya dos tokens con los mismos símbolos.	[1] V_1(1, 2, ... 35, 36) V_2(Ⓐ, Ⓑ, ... ;, ?) V_3(☼, ♠, ... ℃, ℉) V_4(+, −, ... ₣, ₪)
[2] Obtener el User_Id introducido por el usuario durante el inicio de sesión	[2] admin
[3] Obtener el código de acceso** de User_id en la memoria durante el registro	[3] ① ♥ 2 ✉
[4] Guardar un pin aleatorio del código de acceso en : User_Pin_Vector***	[4] ♥
[5] Guardar un número aleatorio [de 1 a 16] en User_Pin_Show_Up_Location	[5] 7
[6] Para (i=1; i<=16; i++) hacer los pasos [7] y [8] para generar 16 tokens	[6]
[7] Crear un token vacío: un vector para contener 4 símbolos	[7] Token_i()
[8] para j=1; j<=4; j++) hacer paso [8.1] u [8.2] para añadir un símbolo de cada dimensión al token. El paso [8.1] asegura que se use al menos 1 pin de usuario.	[8] Añadirle 4 símbolos
[8.1] si i es igual a User_Pin_Show_Up_Location y ♥ de User_Pin_Vector aún está en V_j, eliminarlo de V_j y añadirlo al token actual.	[8.1] Token_7(15 ~ ♥ ☙)
[8.2] si no, eliminar un símbolo aleatorio de V_j para añadirlo al token actual. El tamaño de V_j se reducirá en uno tras eliminar ese símbolo.	[8.2] Token_i(8 € ★ ⊖)
[9] Después de estos pasos, habrá 16 tokens con al menos un pin de usuario	[9] Token_1, ..., Token_16

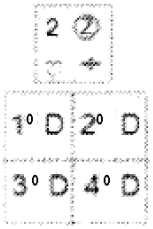
NOTA: * pin - cada símbolo de un código de acceso** es un pin. Por ejemplo, ① es el 1º pin y ✉ es el 4º pin en el siguiente código de acceso : ① ♥ 2 ✉

** código de acceso - similar a una contraseña común, pero puede contener símbolos procedentes de las selecciones de cada dimensión [e.g. ① ♥ 2 ✉], las dimensiones se describen en las FIG. 2 y 2B.

*** User_pin_vector - un Vector en lenguaje Java que puede contener cualquier número de elementos de cualquier tipo, en este caso un símbolo.

Fig. 6

(Reglas de selección de tokens)



[1] Se mostrará al usuario 16 tokens en una tabla 4 x 4, donde cada token tendrá 4 símbolos, por ejemplo, un token se presentará así:
Cada dimensión tiene una posición fija en un token, la parte superior izquierda es para símbolos de la 1ª dimensión, la parte superior derecha es para símbolos de la 2ª dimensión, la parte inferior izquierda es para símbolos de la 3ª dimensión, la parte inferior derecha es para símbolos de la 4ª dimensión.
La posición fija ayudará al usuario a localizar rápidamente los símbolos de un token.

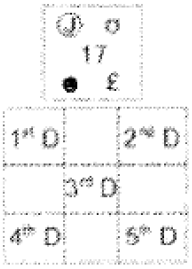
[2] El usuario debe seguir el orden del código de usuario, p.ej. , para seleccionar los tokens que contienen pins de usuario. Por ejemplo, la tabla de la derecha tiene 16 tokens, al menos uno de los cuales contiene un pin de usuario, el usuario puede seguir los siguientes pasos para seleccionar tokens.
Para facilitar las referencias, las filas se denominarán: A, B, C, D, y las columnas: 1, 2, 3, 4.

	1	2	3	4
A	13 &	34 Ⓜ	29 Ⓟ	28 !
B	3 Ⓠ	36 Ⓥ	6 :	19 Ⓦ
C	20 Ⓩ	26 Ⓙ	17 Ⓢ	12 Ⓒ
D	32 Ⓚ	35 Ⓕ	16 Ⓑ	10 Ⓤ

[3] Como el 1º pin es , que pertenece a la 2ª dimensión, el usuario puede buscarlo en la esquina superior derecha de cada token, ya que no está en ninguno de los 16, puede y debe escoger cualquier token en su lugar.

Se puede escoger A2: .
Igual con el 2º pin de usuario , como falta, se puede escoger en su lugar D3 : . El 3º pin también falta, así que se puede escoger B1 : . Sin embargo, el 4º pin de usuario está en el token A1, el usuario debe escogerlo para que sea válido, así que se puede elegir A1 : . El usuario ha escogido estos 4 tokens : , y estos 4 tokens se enviarán al servidor para su validación.

Para GATE_5: El proceso es el mismo que para GATE_4, solo se añade una dimensión más. A la derecha se muestran un ejemplo de token y las localizaciones de las 5 dimensiones. Un token puede ser como este : .



Cada tabla tiene 16 tokens, pero cada token contiene 5 símbolos - uno de cada una de las 5 dimensiones de la FIG. 2B.

Fig. 7

(Reglas de validación de tokens)

Para GATE_4: continuar con el ejemplo de la FIG 6

[1] El usuario debe seguir el orden del código de acceso ⓘ ♥ 2 ✉ para seleccionar los tokens que contienen pines de usuario.

[2] Los 16 tokens originales se muestran en la tabla de la derecha

[3] El usuario seleccionó los 4 tokens: (34 Ⓜ ♢ ♫), (16 Ⓑ ● 👁), (3 Ⓠ ← ®) y (13 & ☺ ✉).

[4] El proceso de validación los comprobará uno a uno y, si uno de ellos no es válido, la solicitud de inicio de sesión será denegada.

[5] Si el usuario selecciona más o menos tokens que el número de pines de su código, la solicitud de inicio de sesión también será denegada.

[6] En la implementación de muestra, los códigos de acceso pueden ser de hasta 6 pines.

	1	2	3	4
A	13 & ☺ ✉	34 Ⓜ ♢ ♫	29 Ⓟ → ☢	28 ! ✓ ÷
B	3 Ⓠ ← ®	36 Ⓥ = ◒	6 : ↓ π	19 Ⓦ X +
C	20 Ⓩ ☪ ☞	26 Ⓙ ₸ €	17 Ⓢ ♤ X	12 Ⓒ ♡ ₪
D	32 Ⓚ ♠ $	35 Ⓕ ☆ ❁	16 Ⓑ ● 👁	10 Ⓤ ℃ ☎

[7] Como el 1º pin de usuario es ⓘ se observarán todos los símbolos de todos los 16 tokens para comprobar si ⓘ existe. Como no existe, el usuario puede y debe escoger un token comodín en su lugar, y en este ejemplo es válido el 1º token de usuario (34 Ⓜ ♢ ♫)

[8] El 2º pin ♥ de usuario también falta en los 16 tokens, así que el 2º token elegido por el usuario (16 Ⓑ ● 👁) también es válido, como el 3º token (3 Ⓠ ← ®) elegido por el usuario.

[9] El 4º pin del usuario era ✉, y al observar todos los símbolos de los 16 tokens anteriores, se puede ver que está en el token A1, así que, para que sea válido, el usuario debe y puede elegir solo A1; en el ejemplo, escogió (13 & ☺ ✉) como su 4º y último token, ni más ni menos que el número de pines (4) de su código de acceso original; por lo tanto, se aprueba la solicitud de inicio de sesión

Para GATE_5: El proceso es el mismo, solo se añade una dimensión

Fig. 8A GATE_4_Create_Id

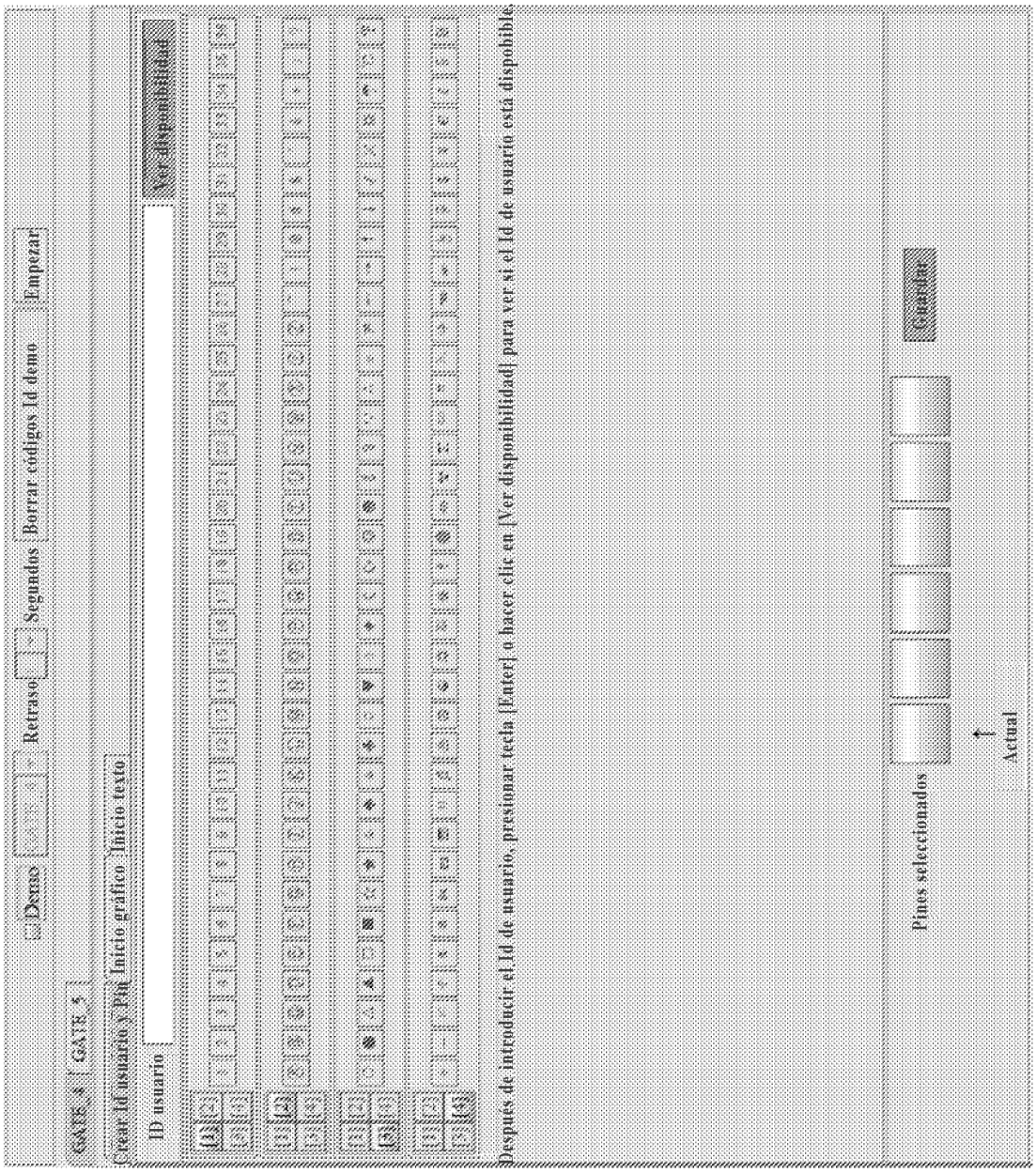


Fig. 8B GATE_4_Create_Id

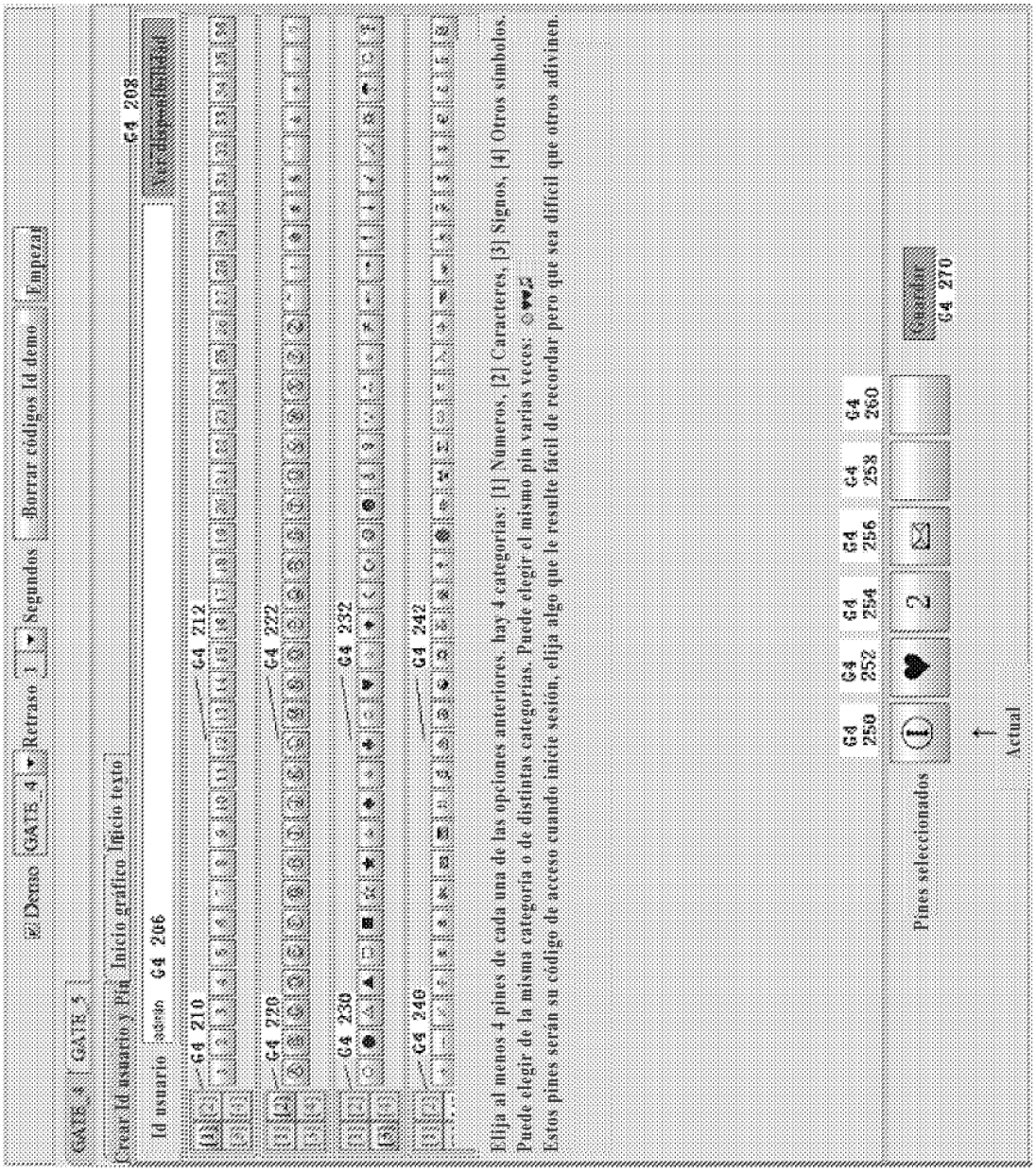


Fig. 9A GATE_4_Graphic_Login

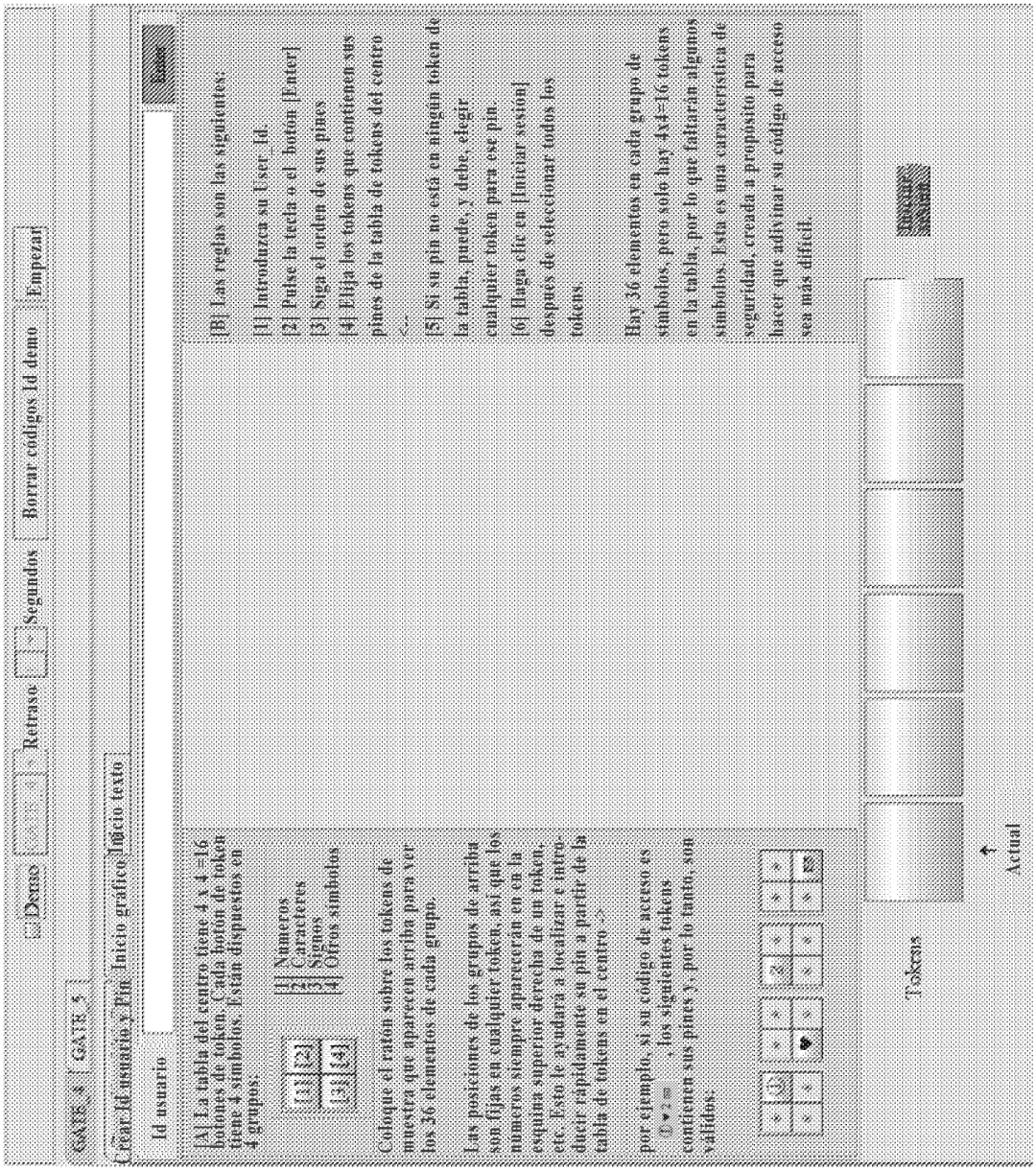


Fig. 9B GATE_4_Graphic_Login

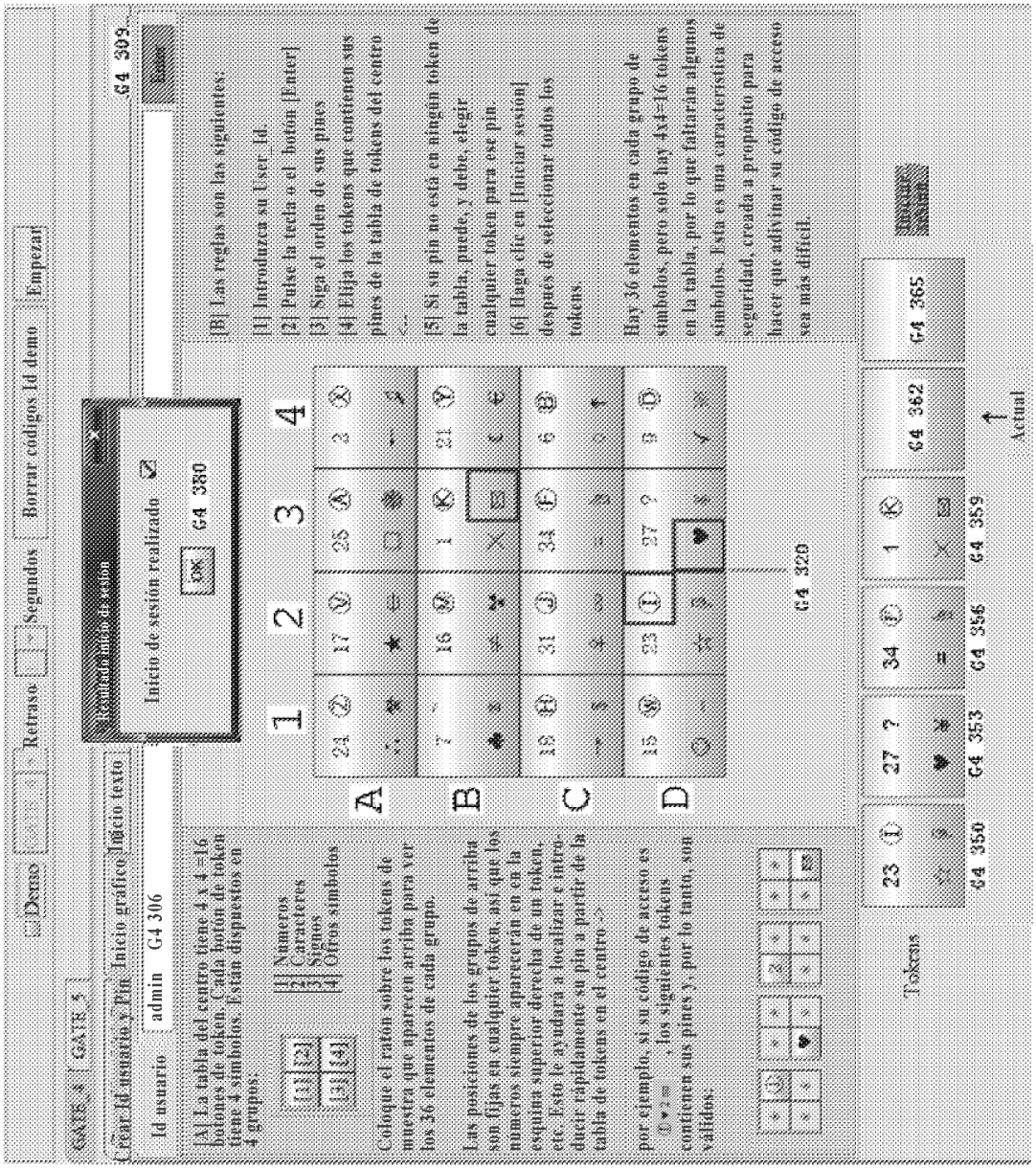


Fig. 9C GATE_4_Graphic_Login

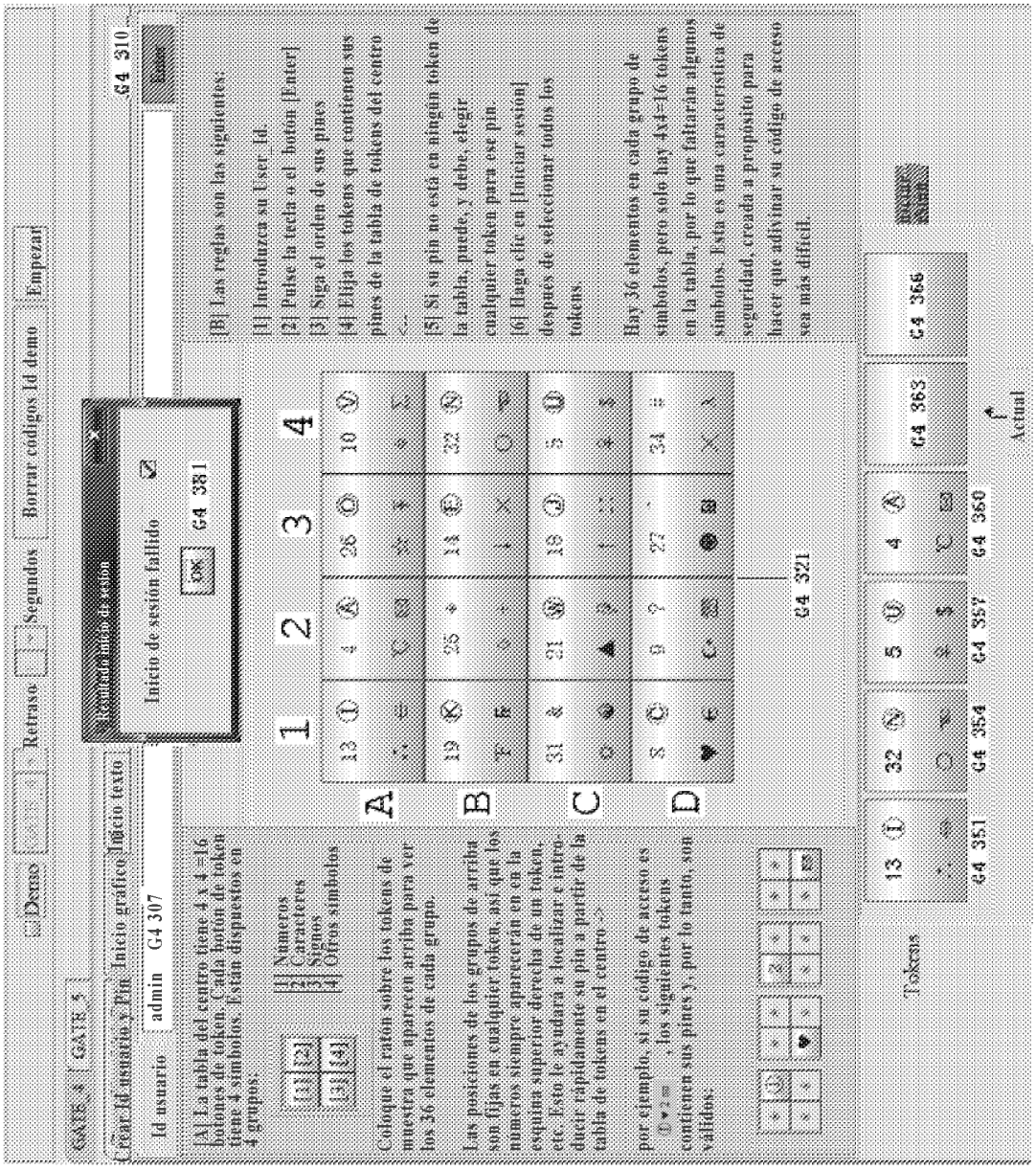


Fig. 9D GATE_4_Graphic_Login

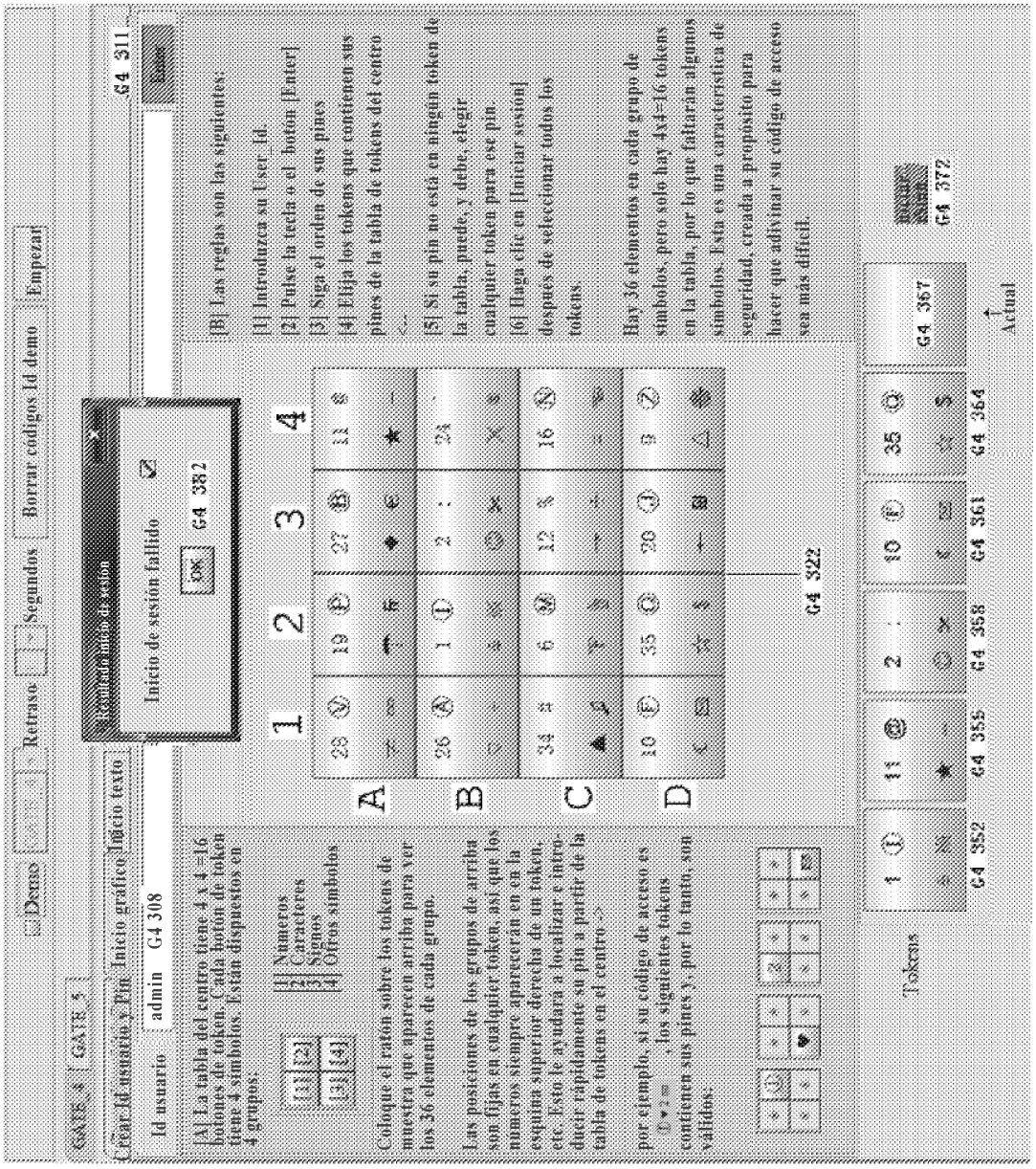


Fig. 10A GATE_4_Text_Login

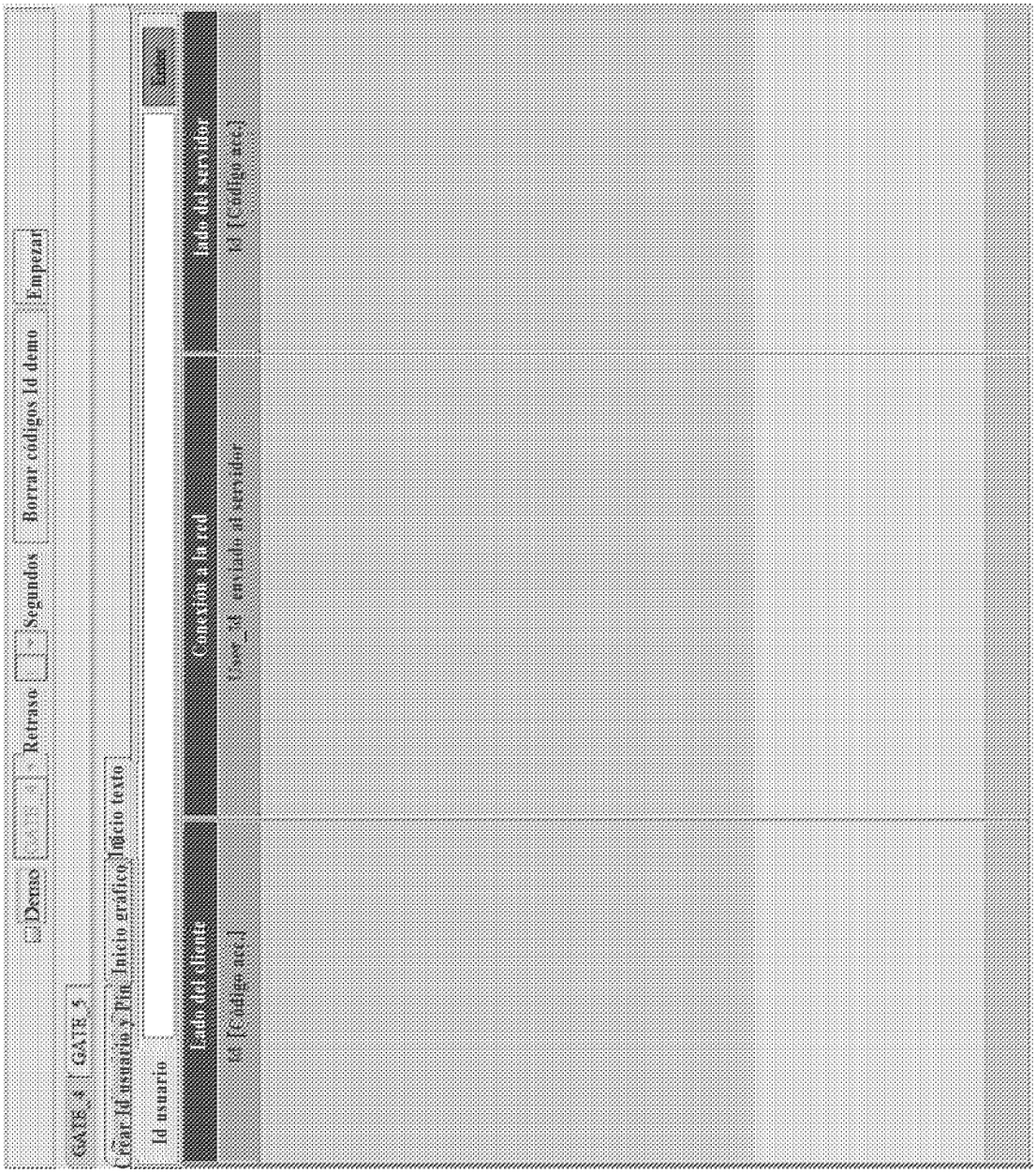


Fig. 10B GATE_4_Text_Login

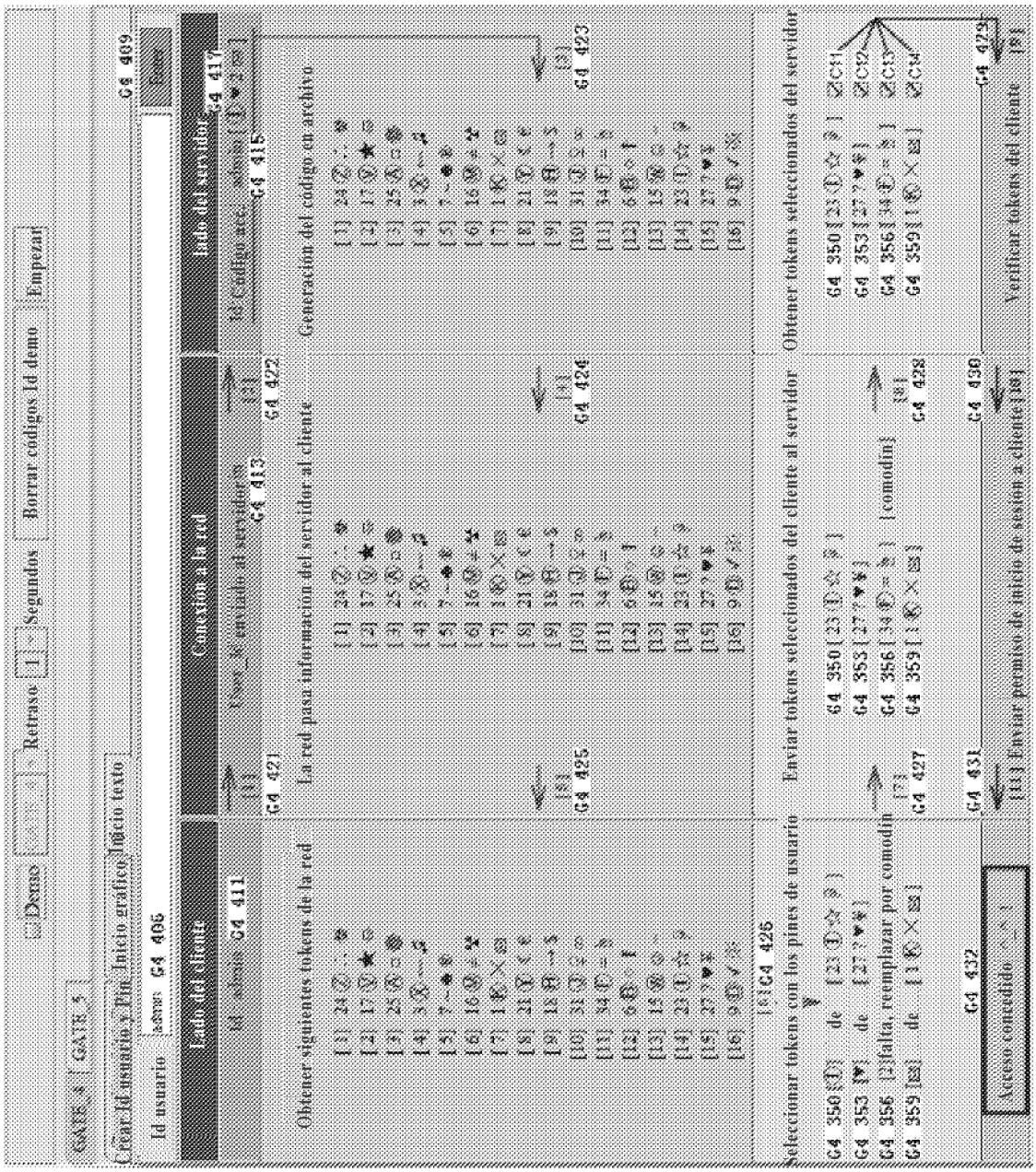


Fig. 10C GATE_4_Text_Login

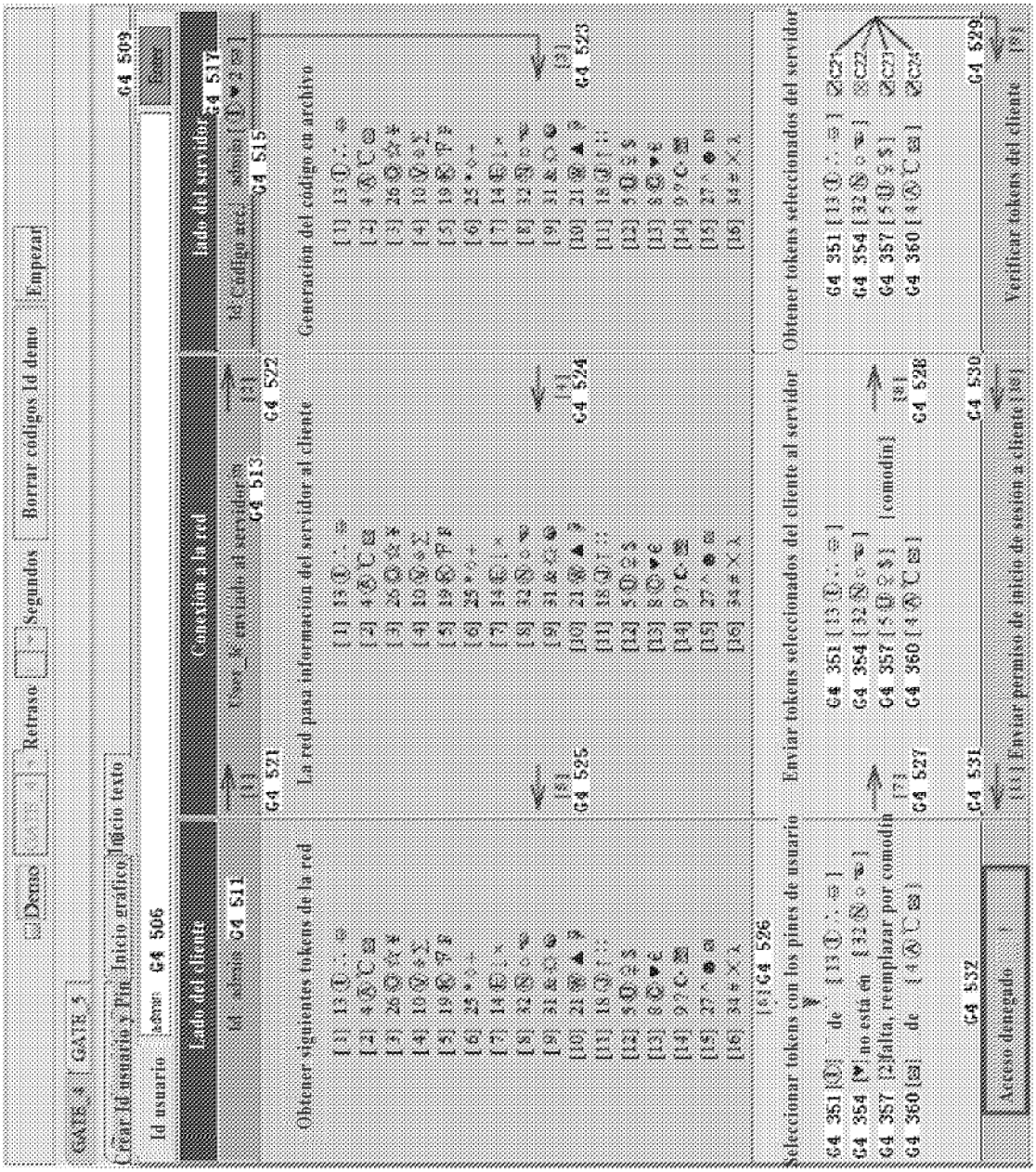


Fig. 10D GATE_4_Text_Login

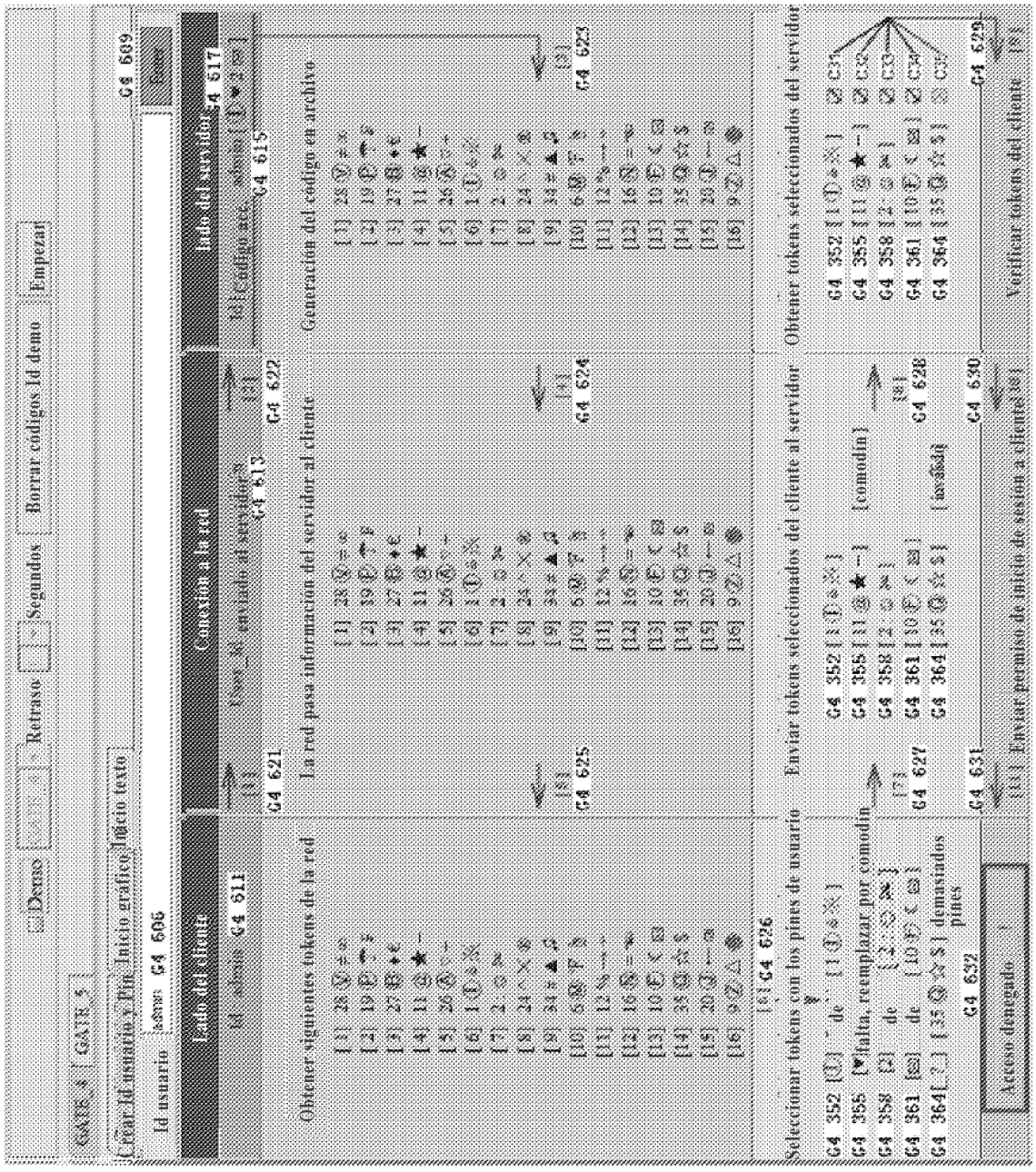


Fig. 11A GATE_5_Create_Id

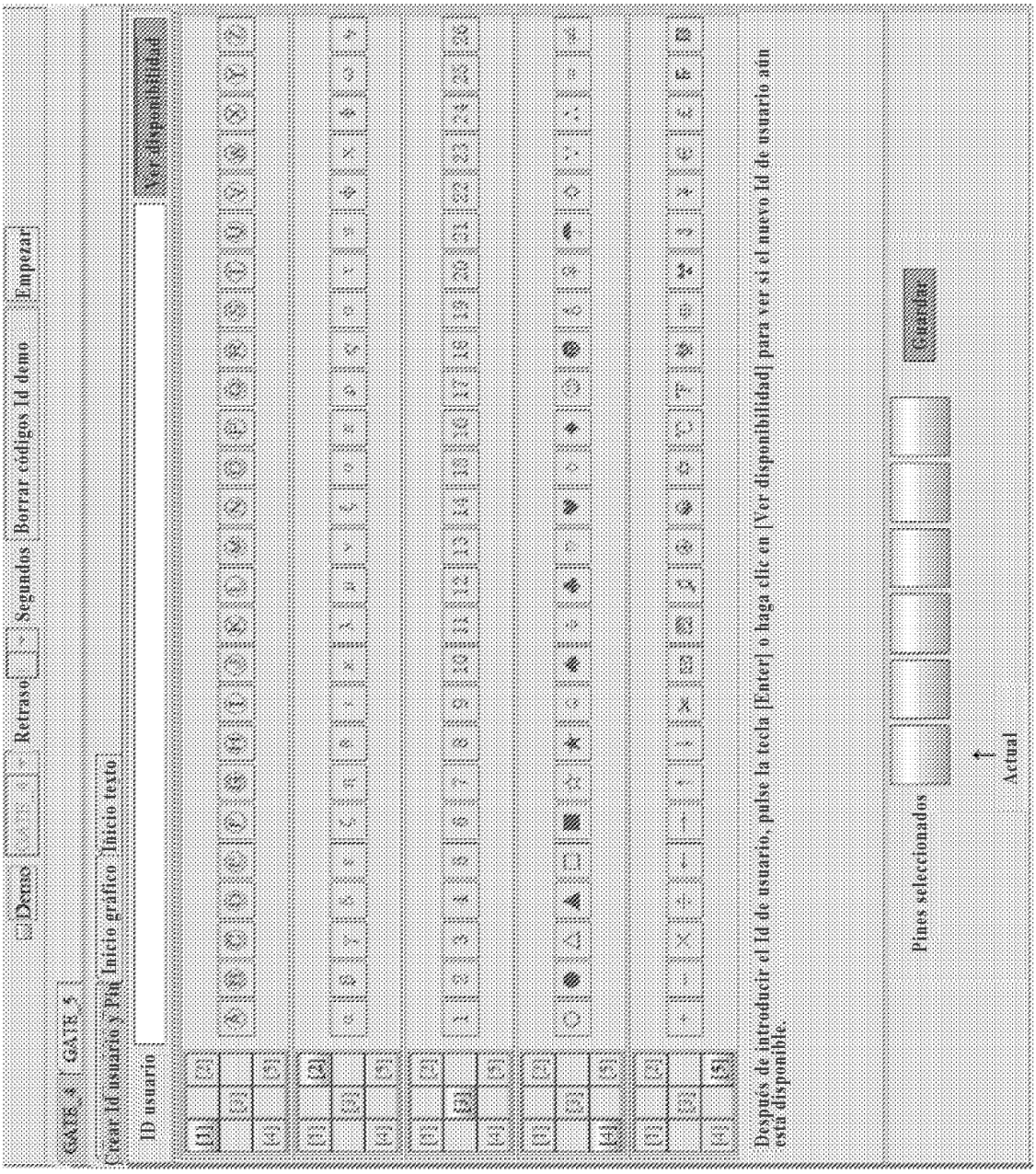


Fig. 11B GATE_5_Create_Id

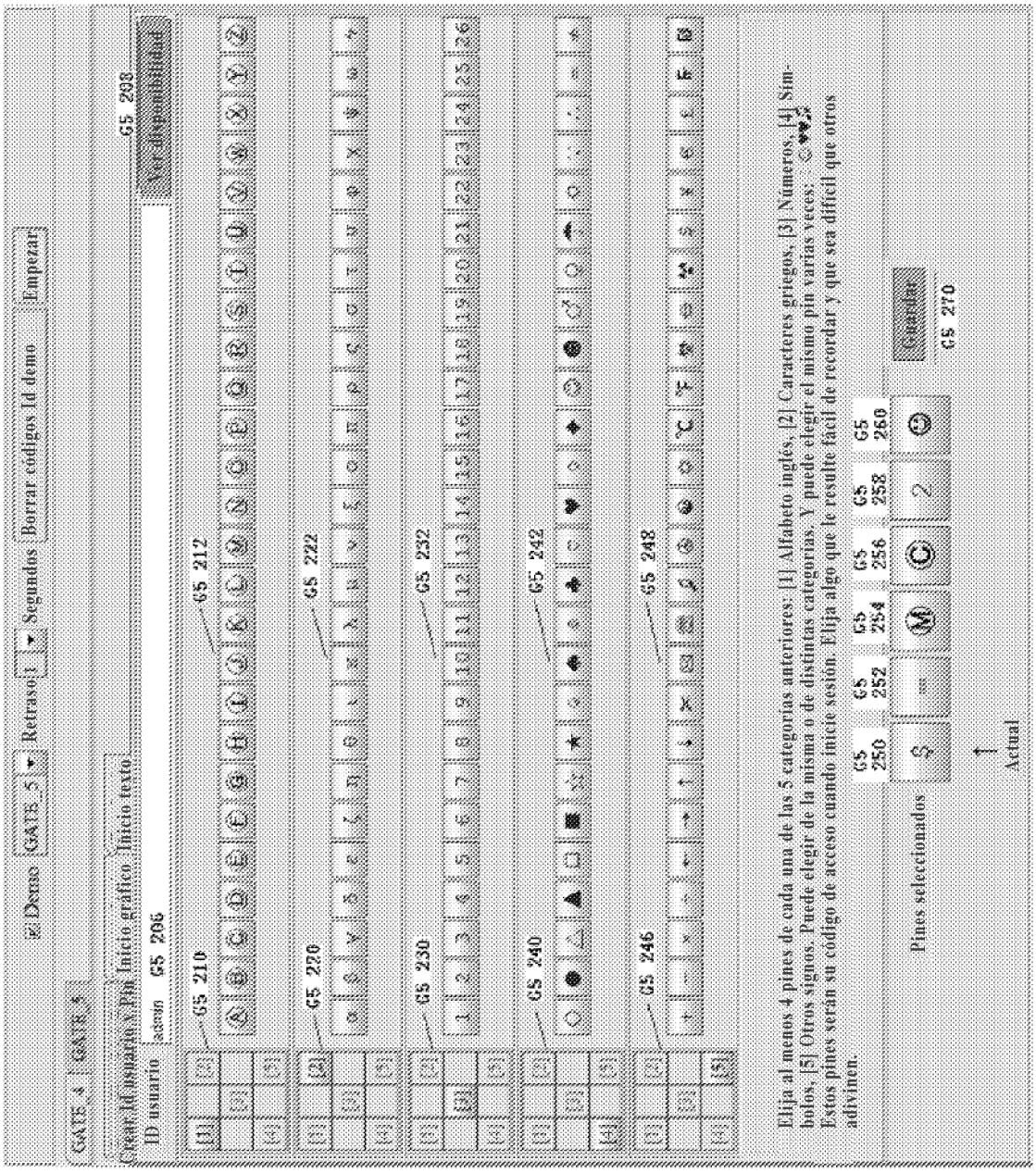


Fig. 12A GATE_5_Graphic_Login

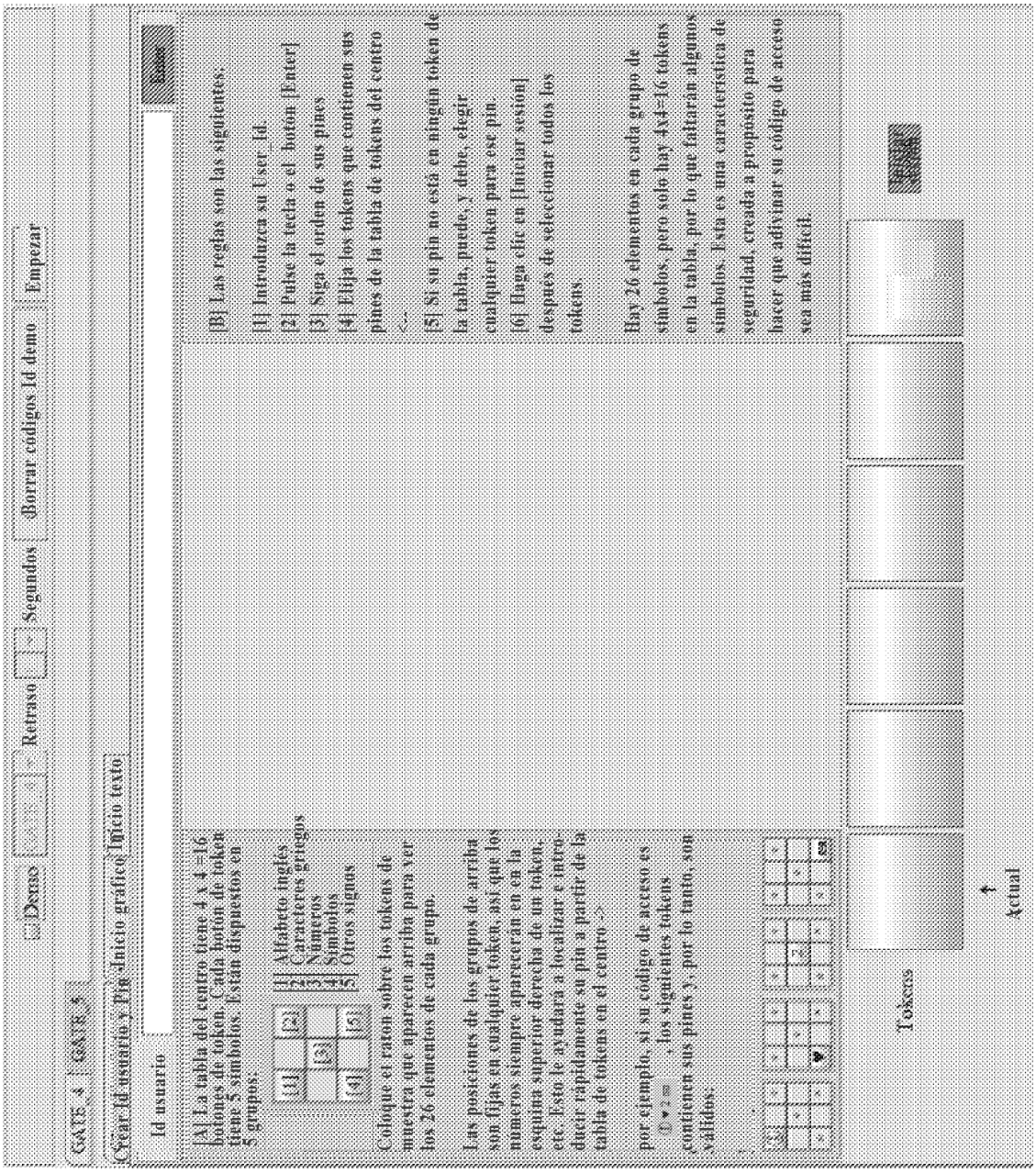


Fig. 12B GATE_5_Graphic_Login

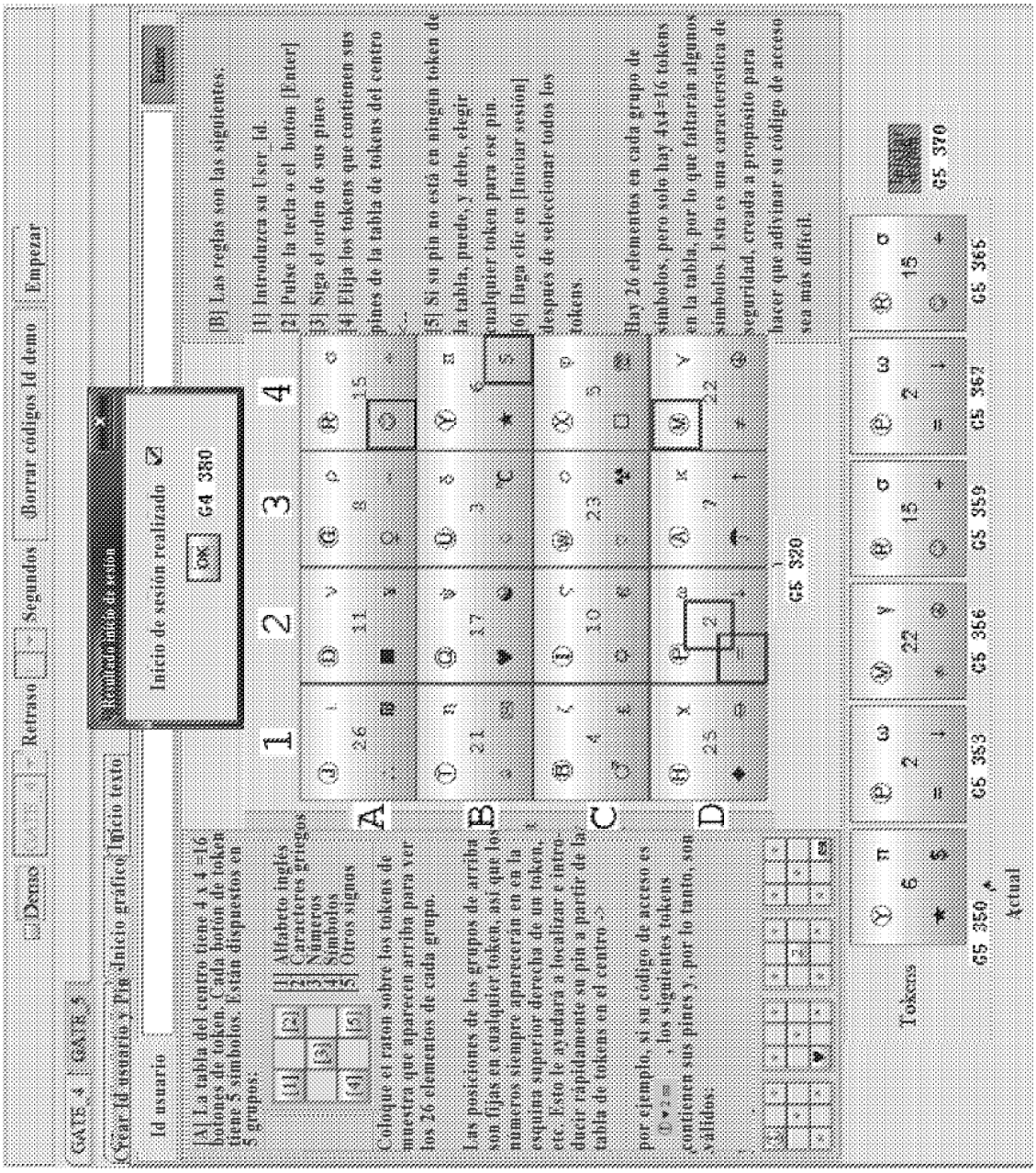


Fig. 12C GATE_5_Graphic_Login

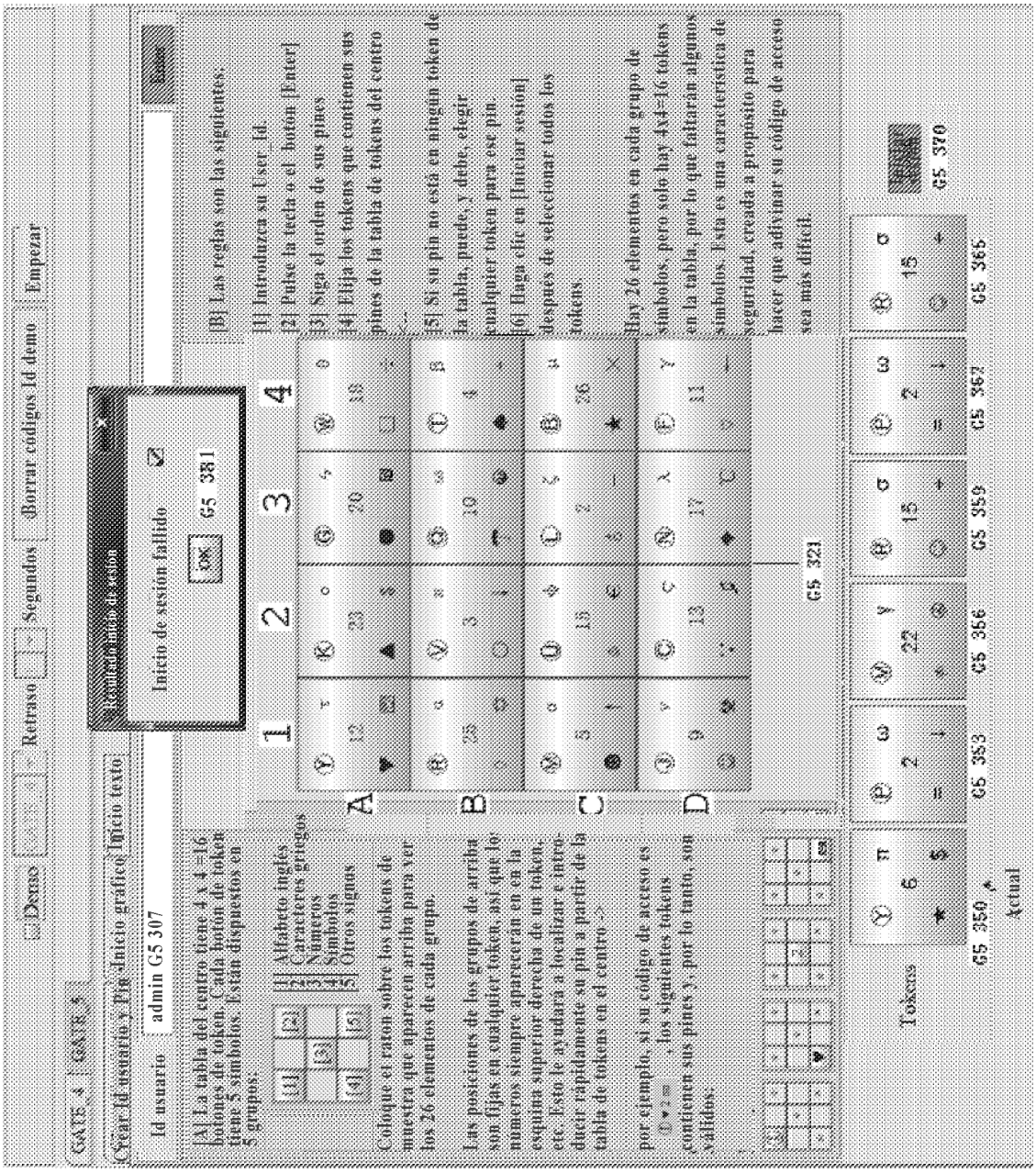


Fig. 12D GATE_5_Graphic_Login

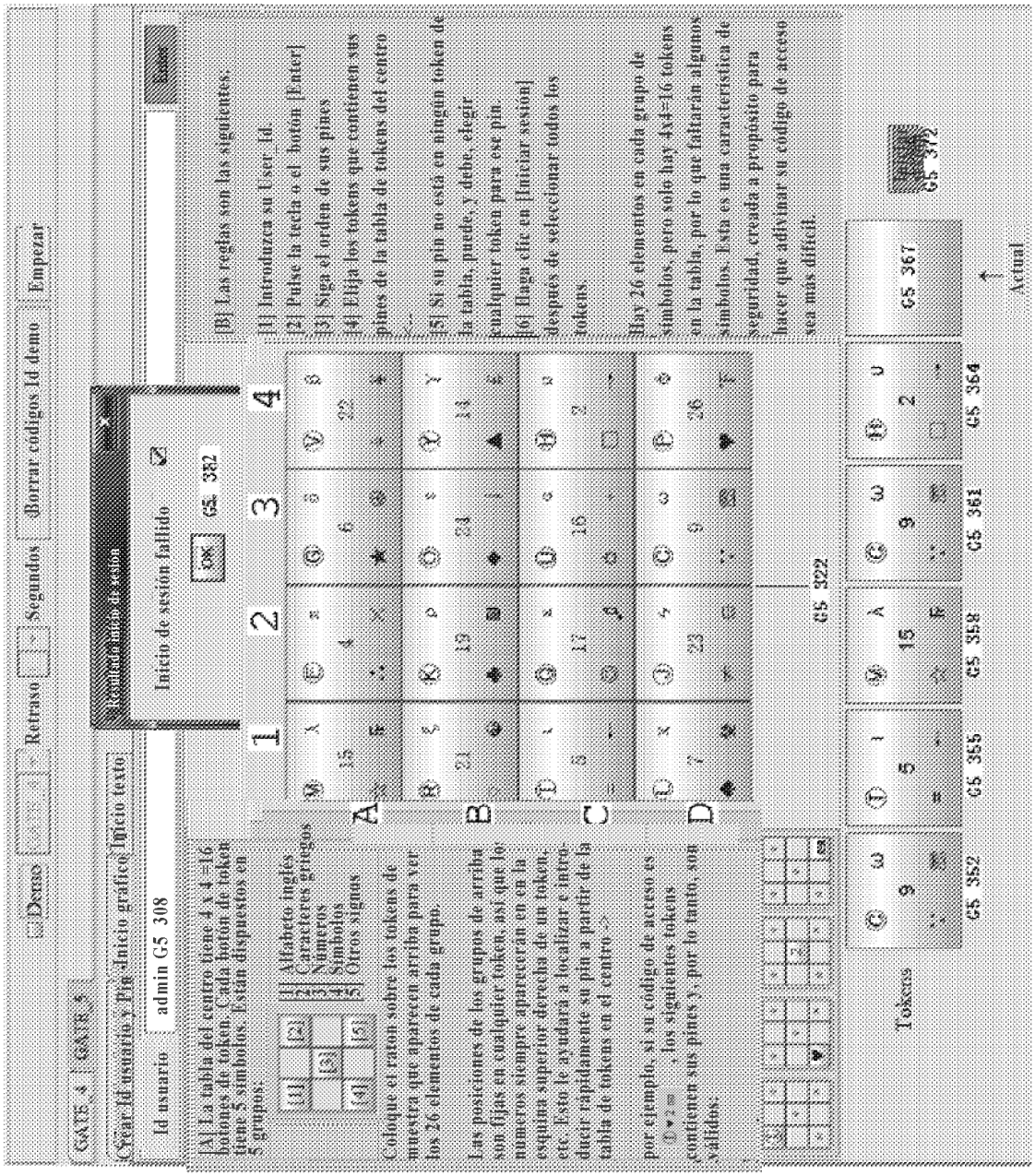


Fig. 13A GATE_5_Text_Login

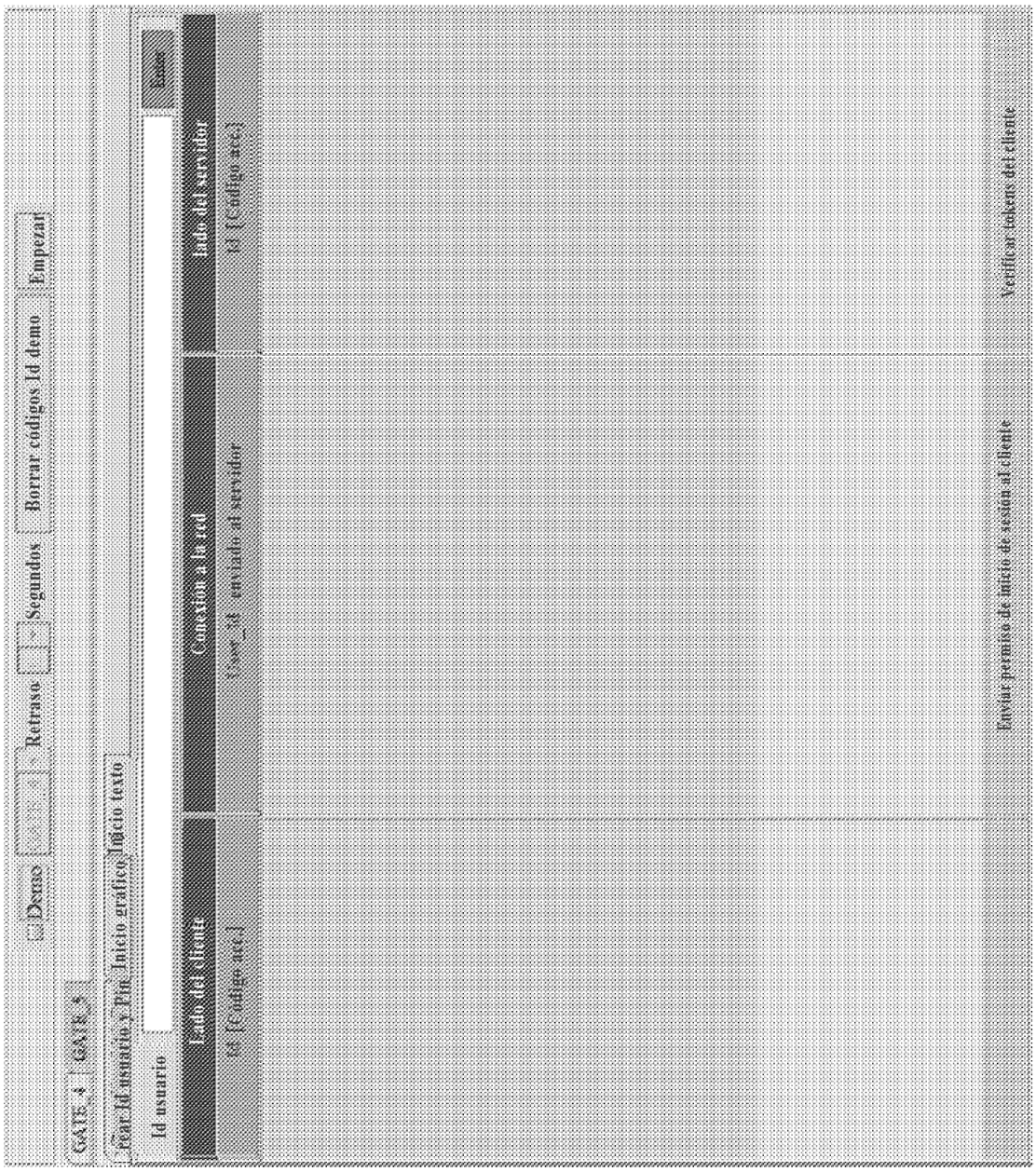


Fig. 13B GATE_5_Text_Login

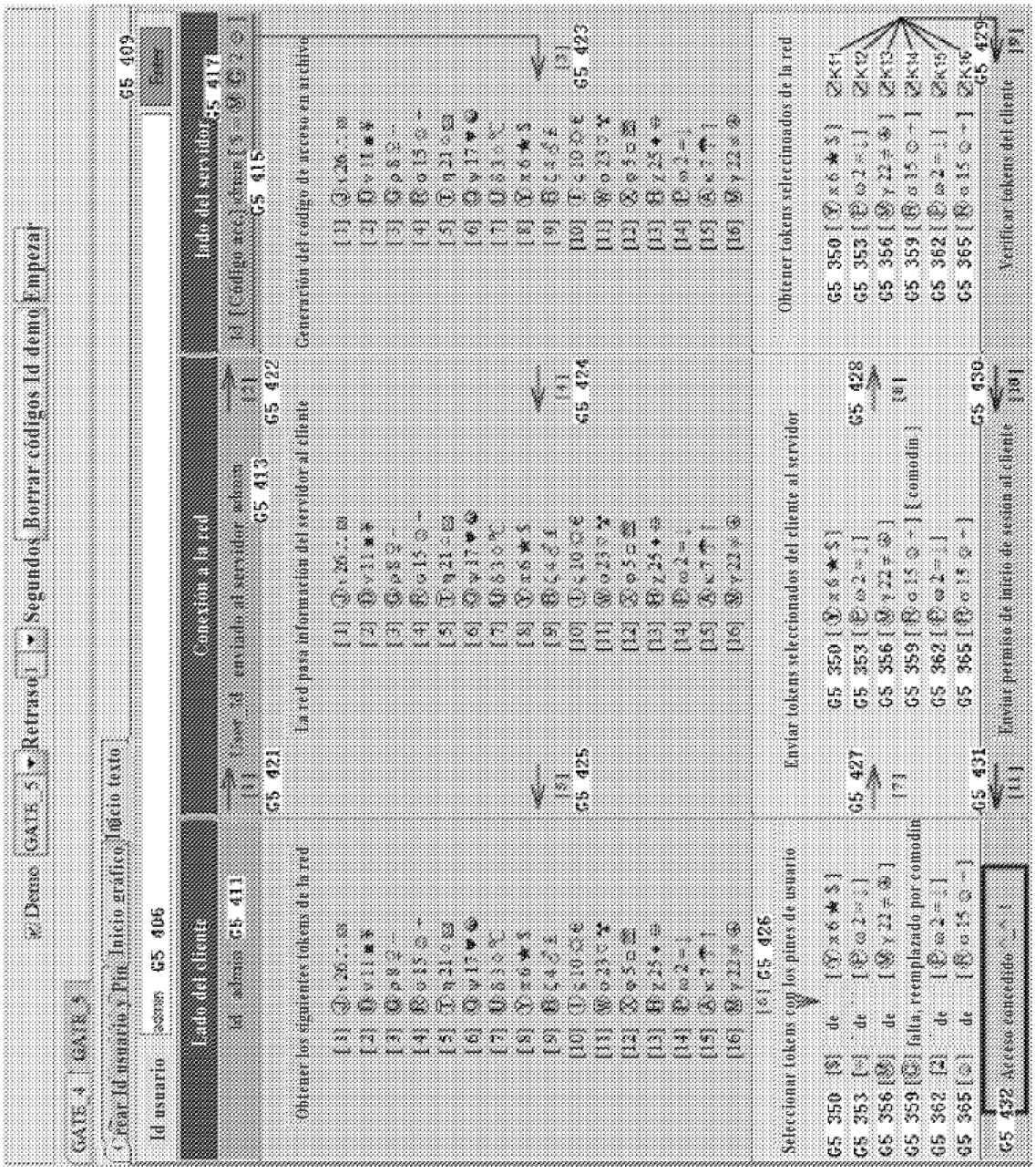


Fig. 13C GATE_5_Text_Login

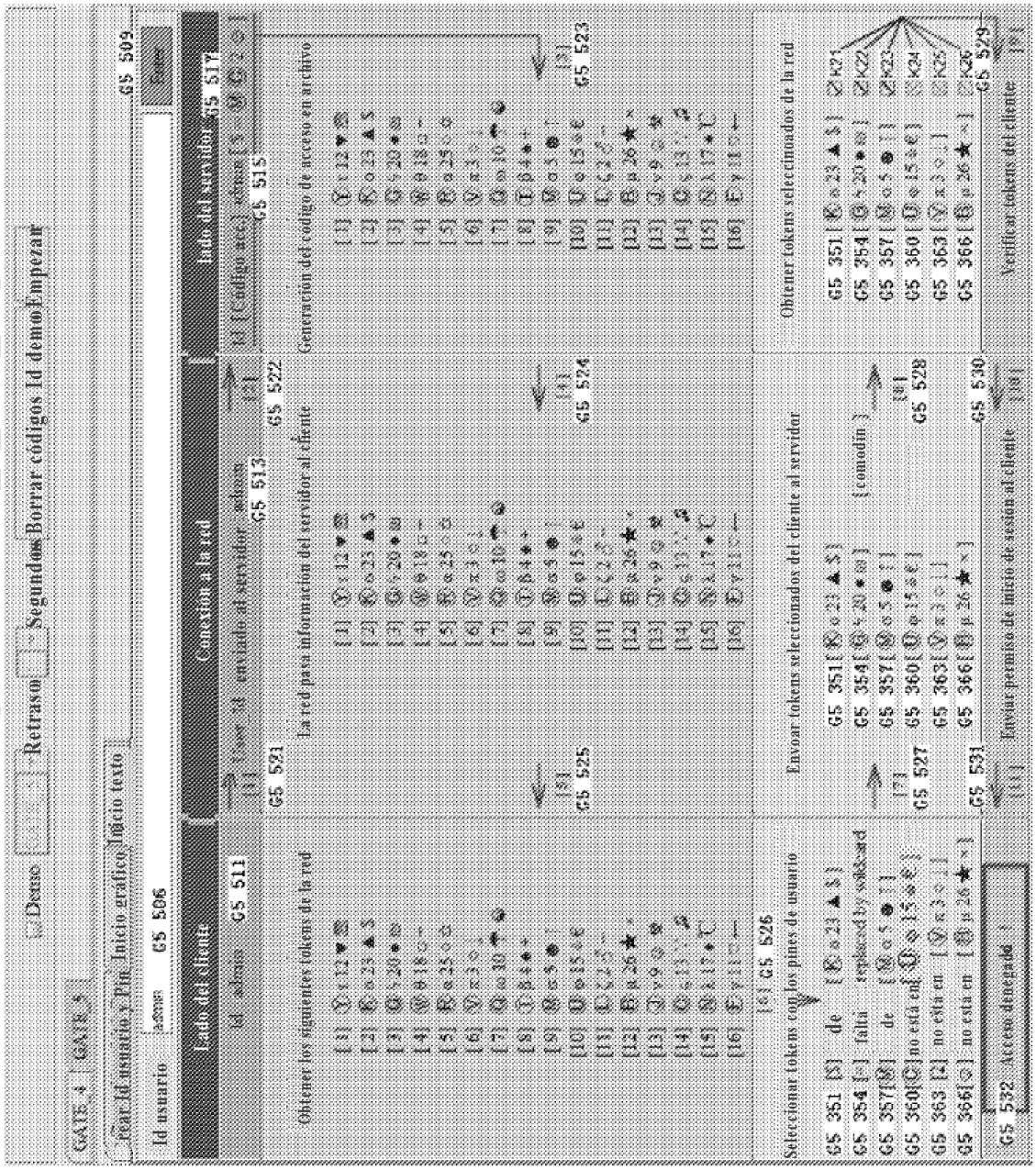


Fig. 13D GATE_5_Text_Login

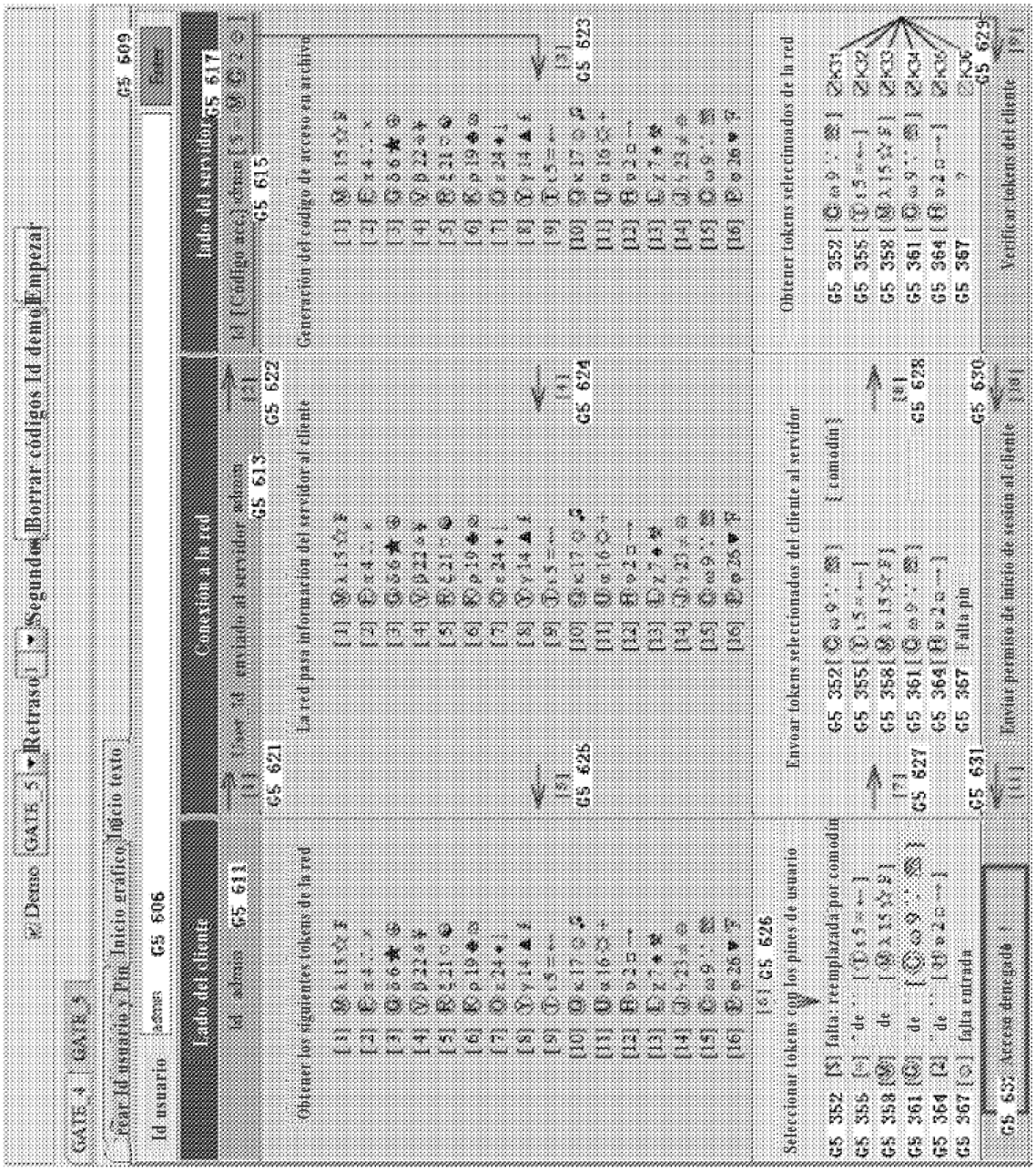


Fig. 14 GATE_4_Encryption

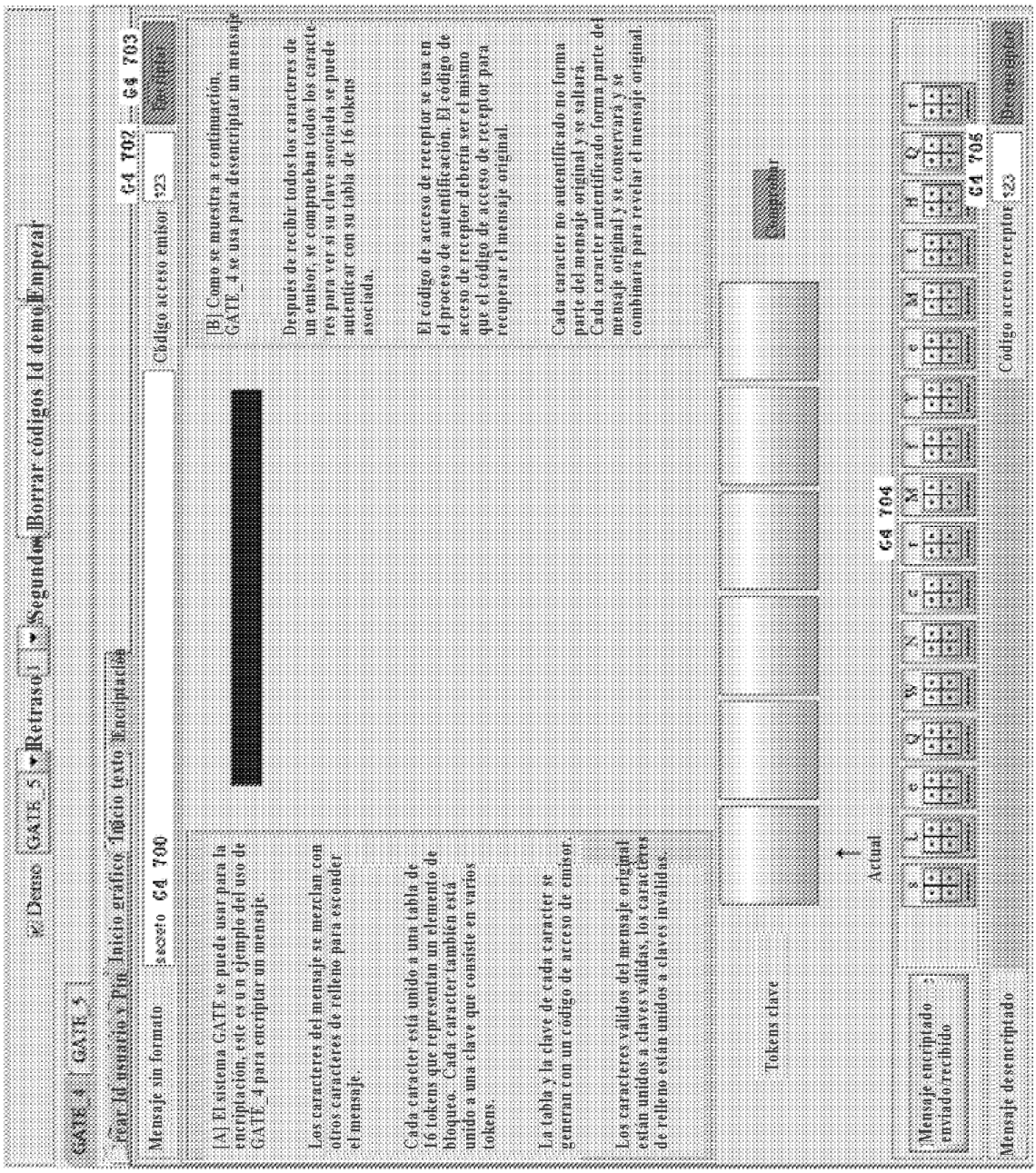


Fig. 15A GATE_4_Encryption

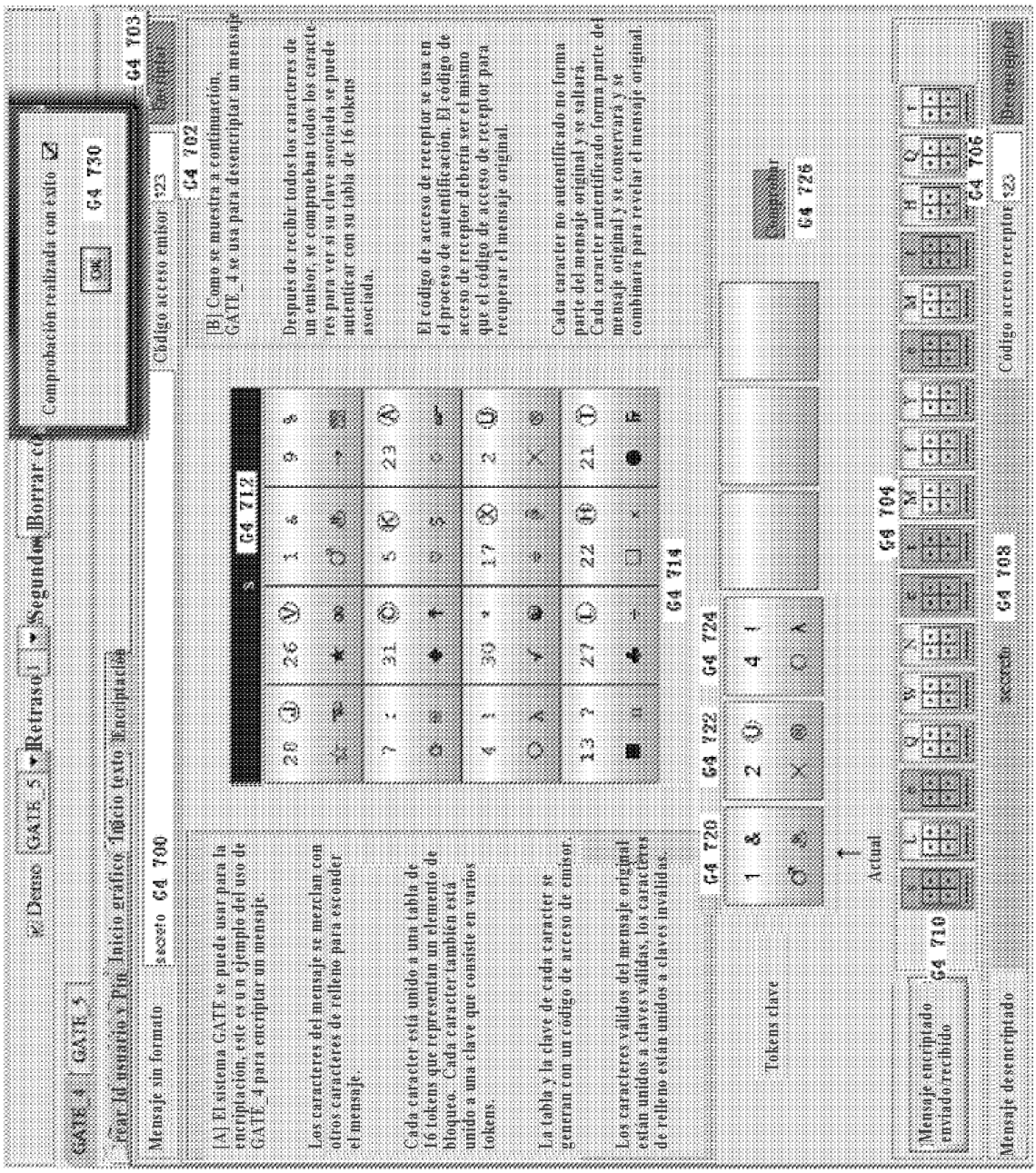


Fig. 15B GATE_4_Encryption

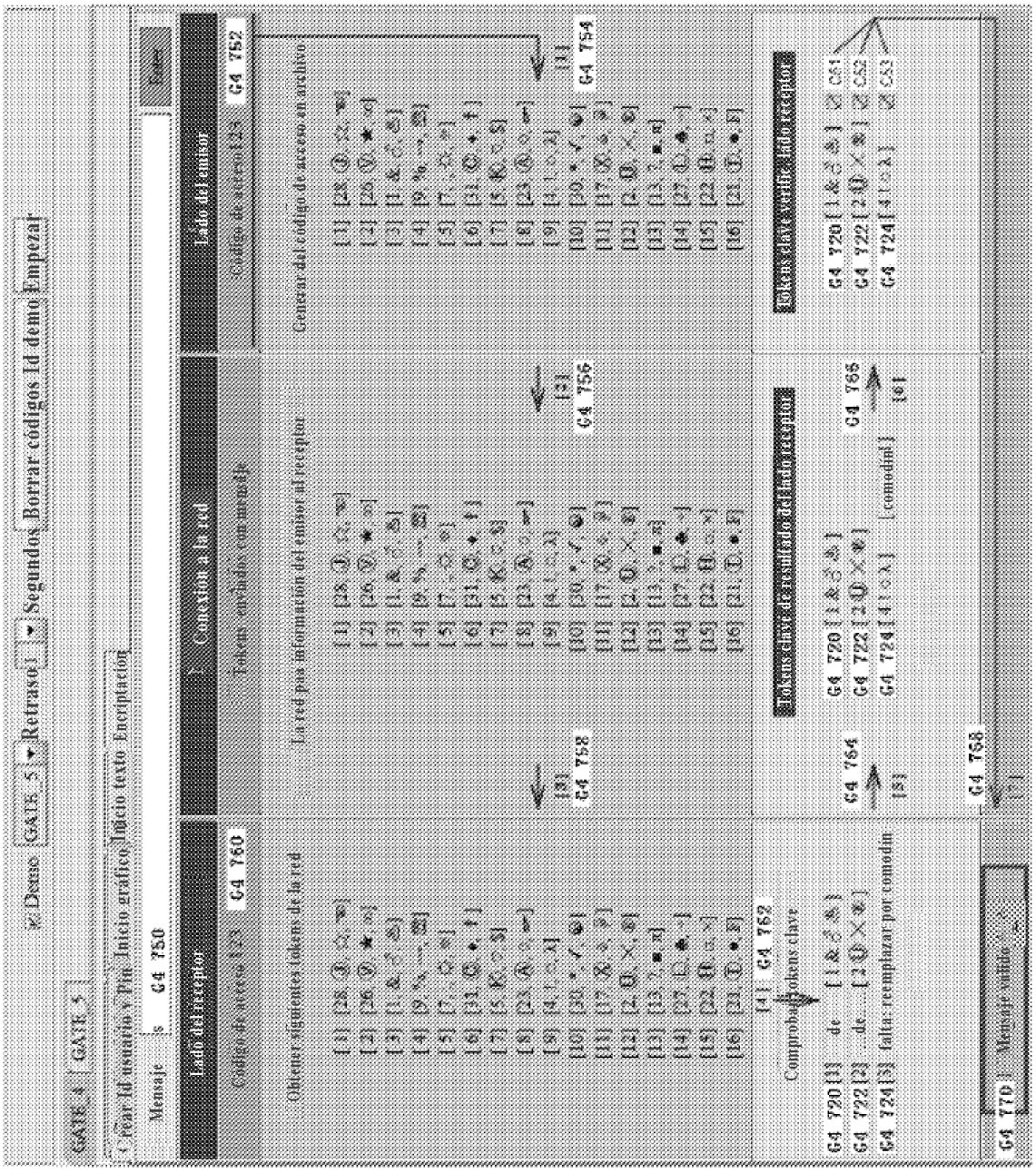


Fig. 16A GATE_4_Encryption

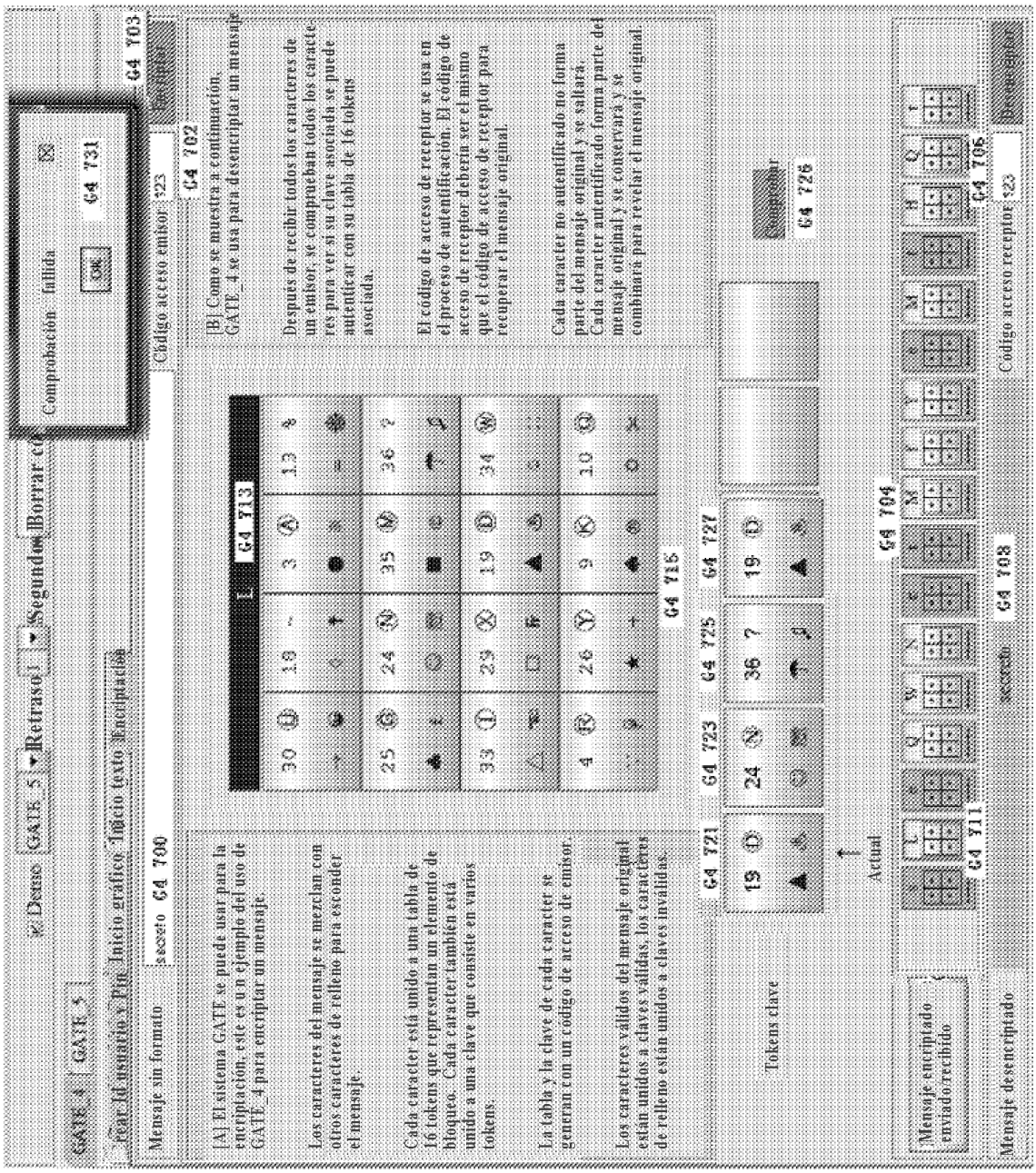


Fig. 16B GATE_4_Encryption

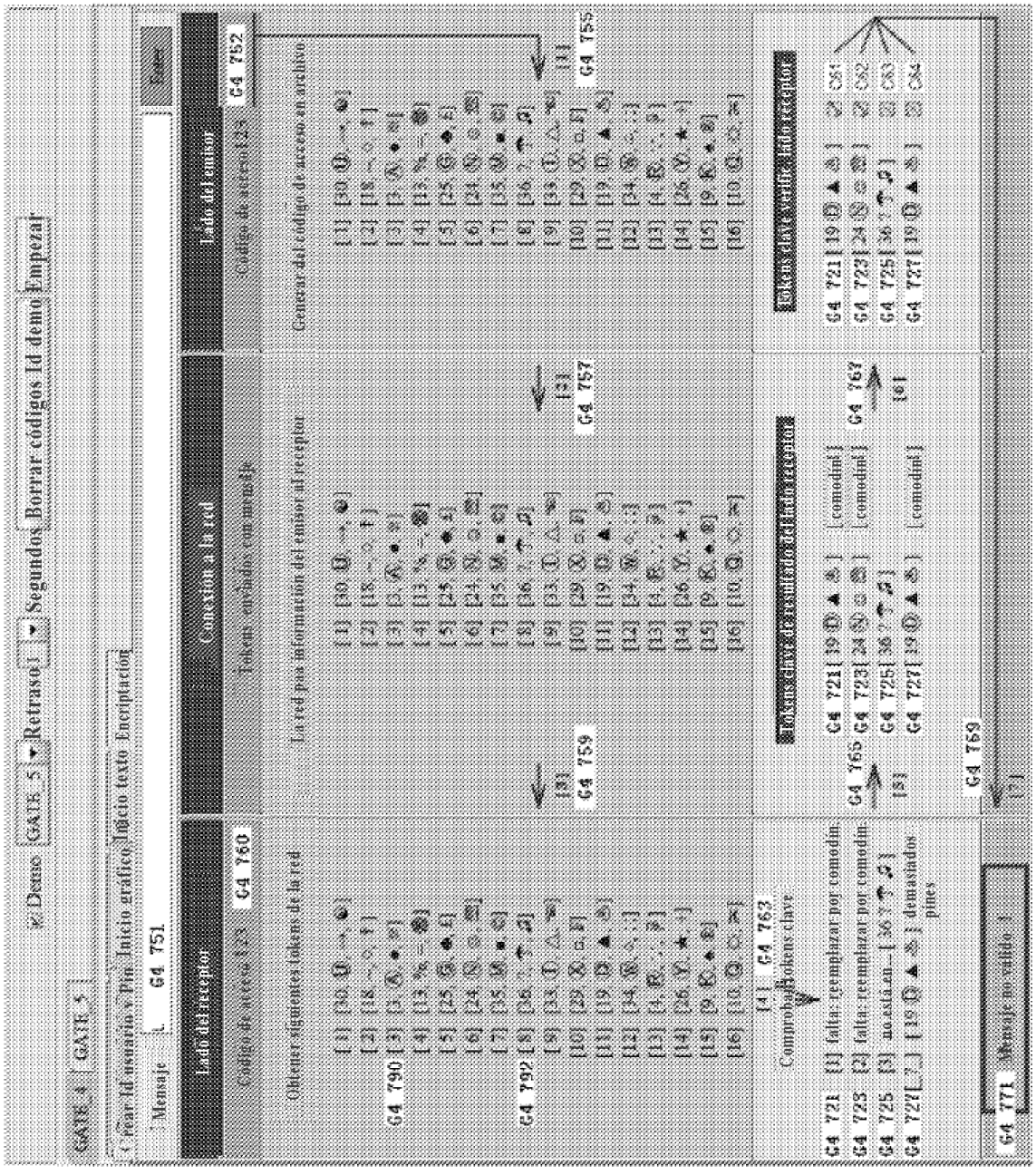


Fig. 17 GATE_4_Encryption

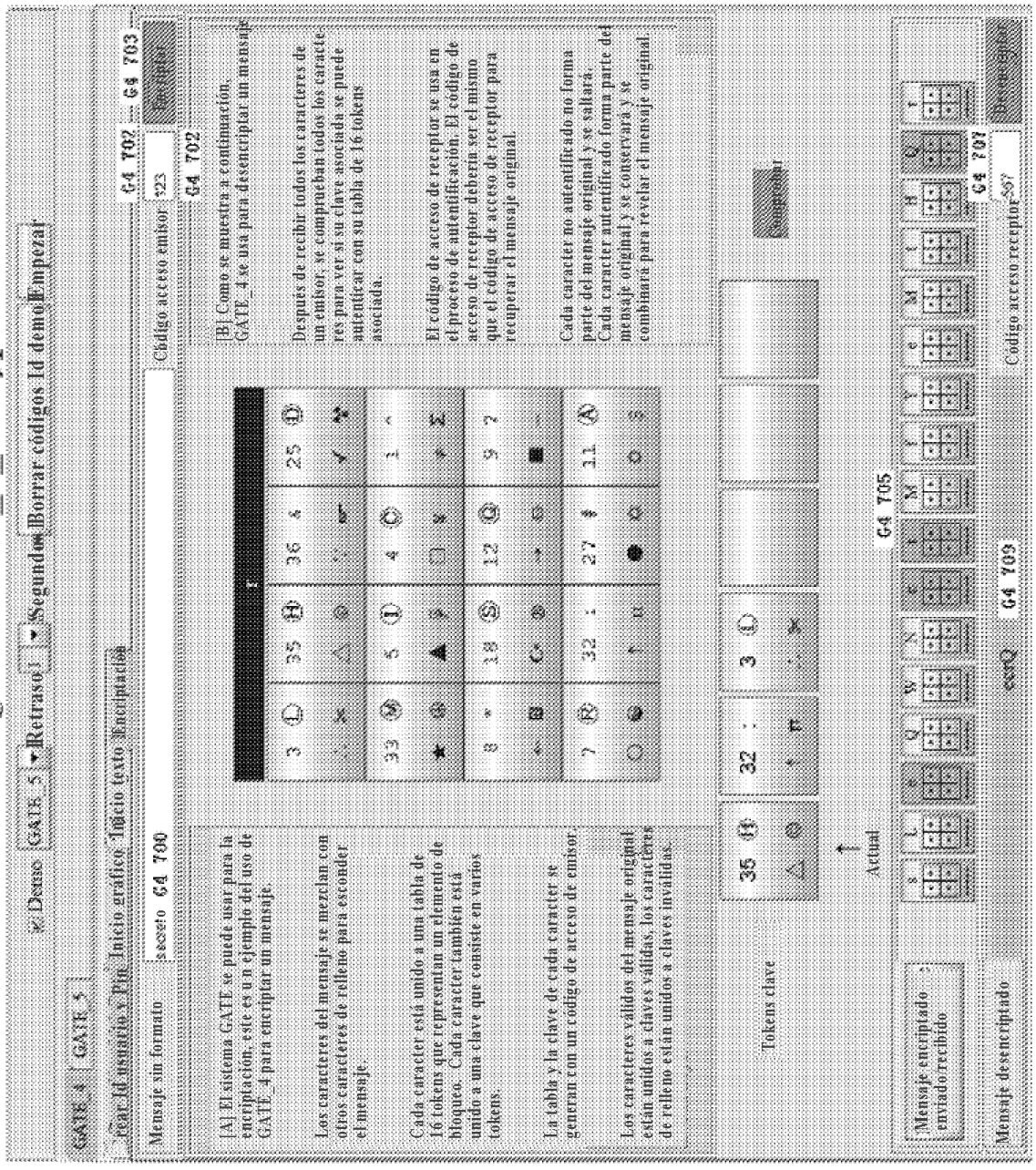


Fig. 18 GATE_5_Encryption

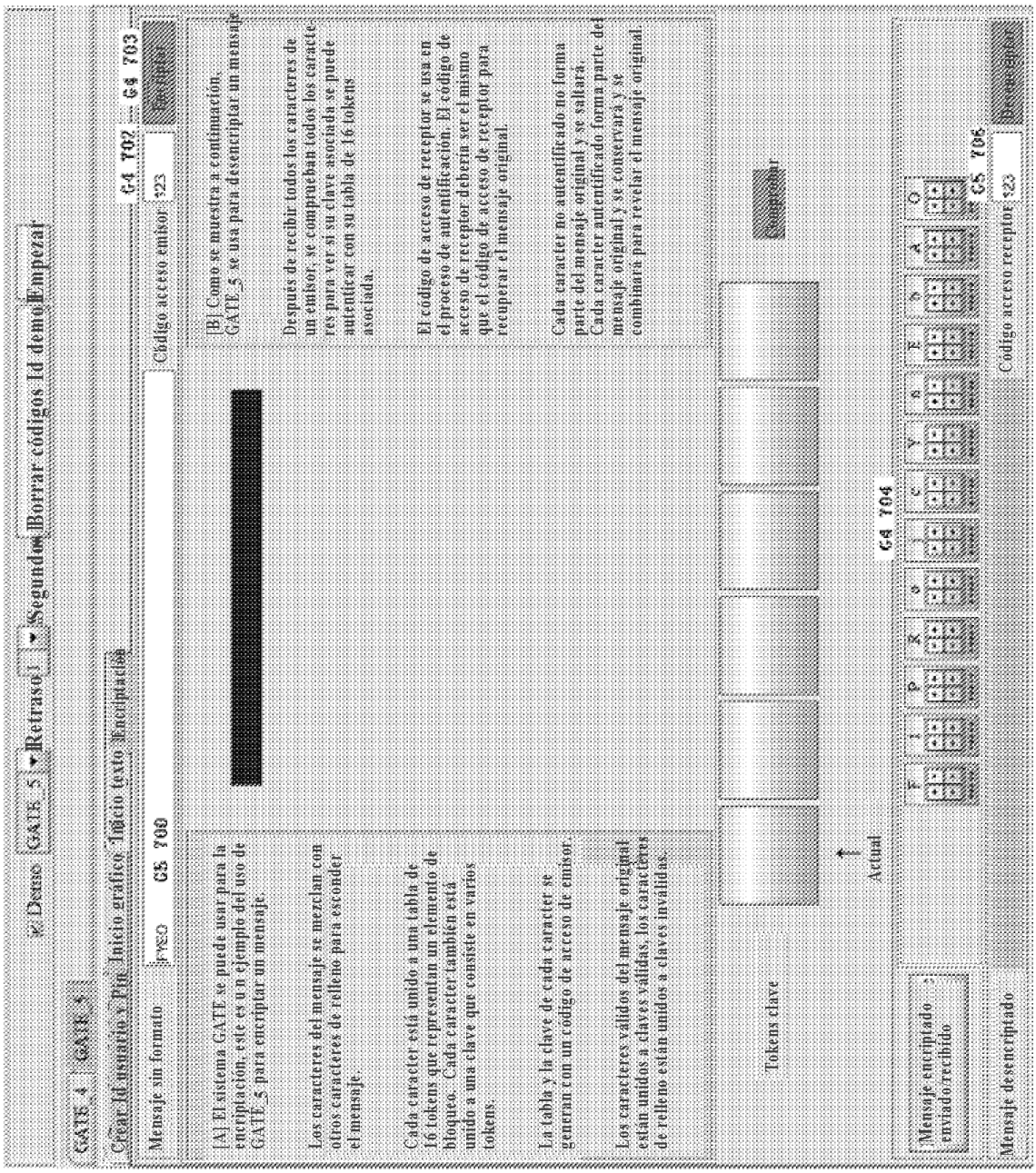


Fig. 19A GATE_5_Encryption

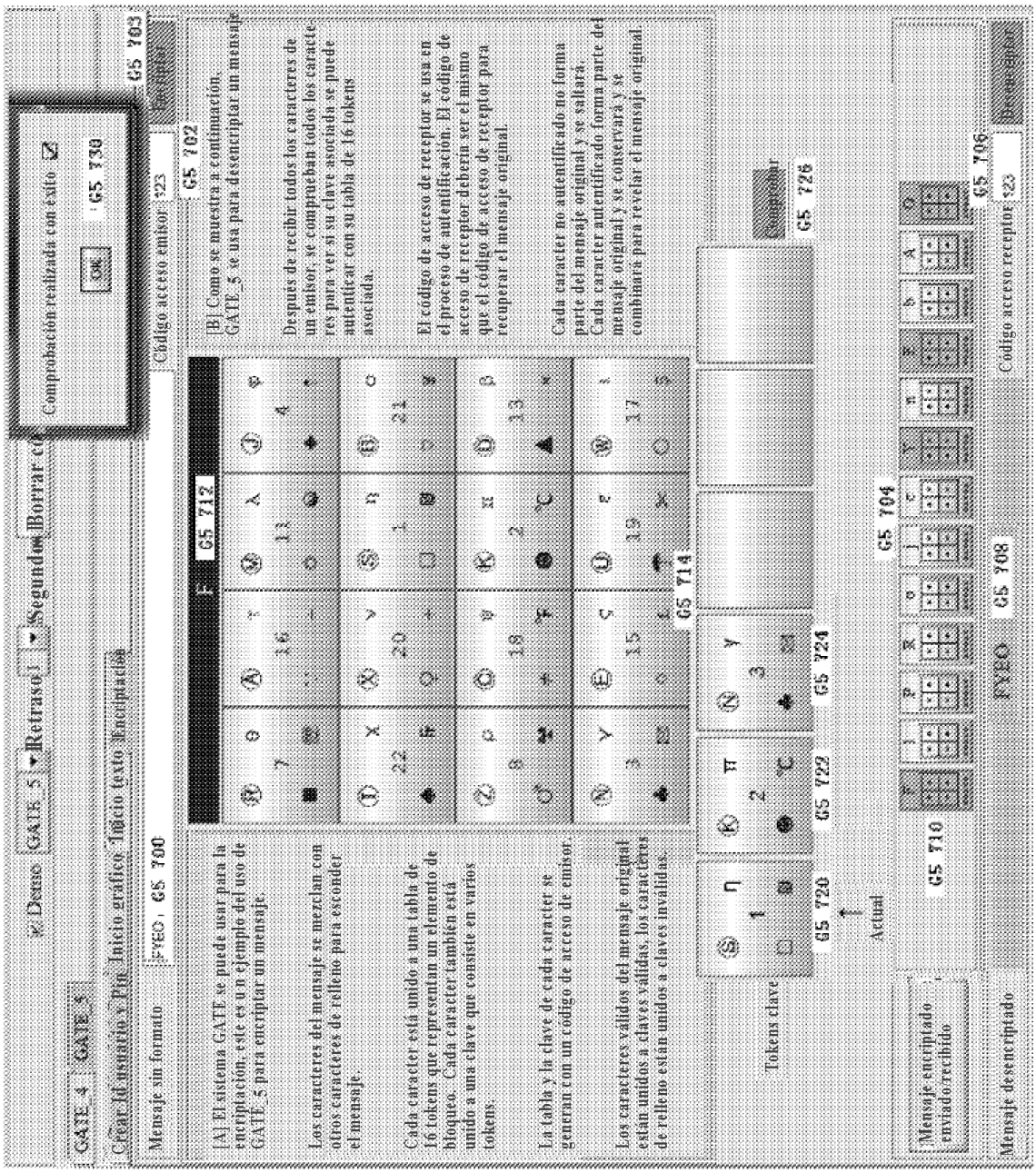


Fig. 19B GATE_5 Encryption

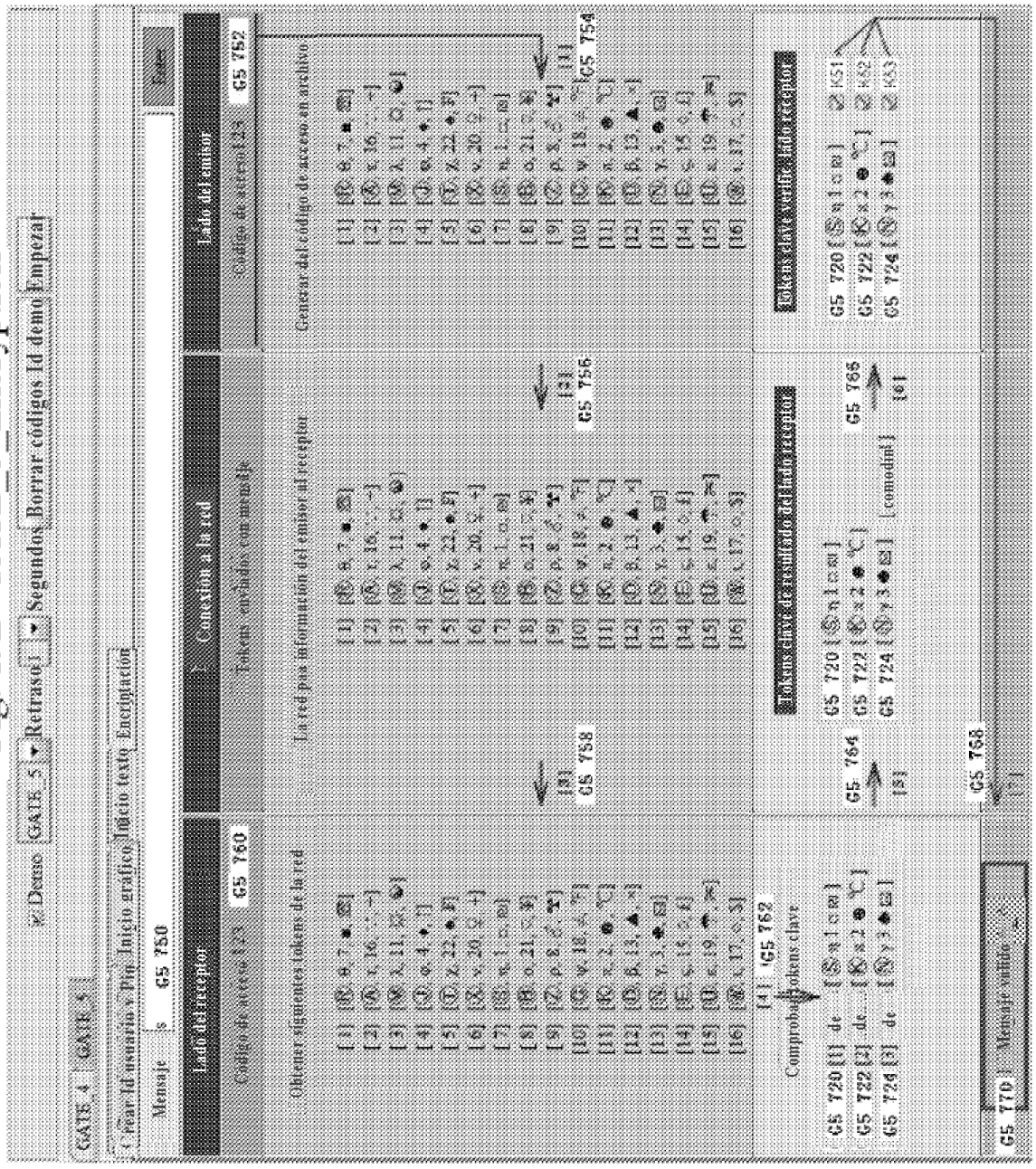


Fig. 20A GATE_5_Encryption

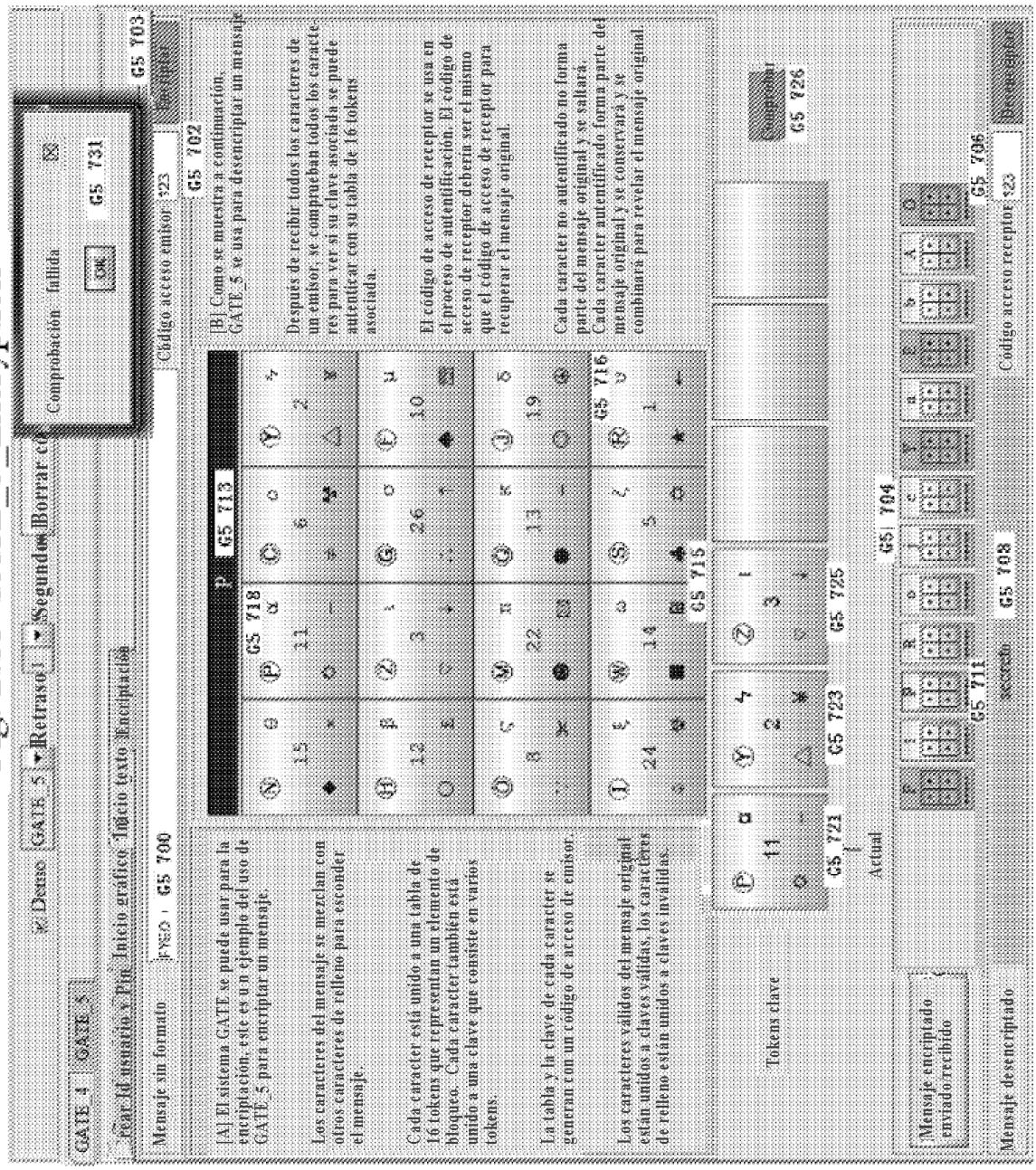


Fig. 20B GATE_5_Encryption

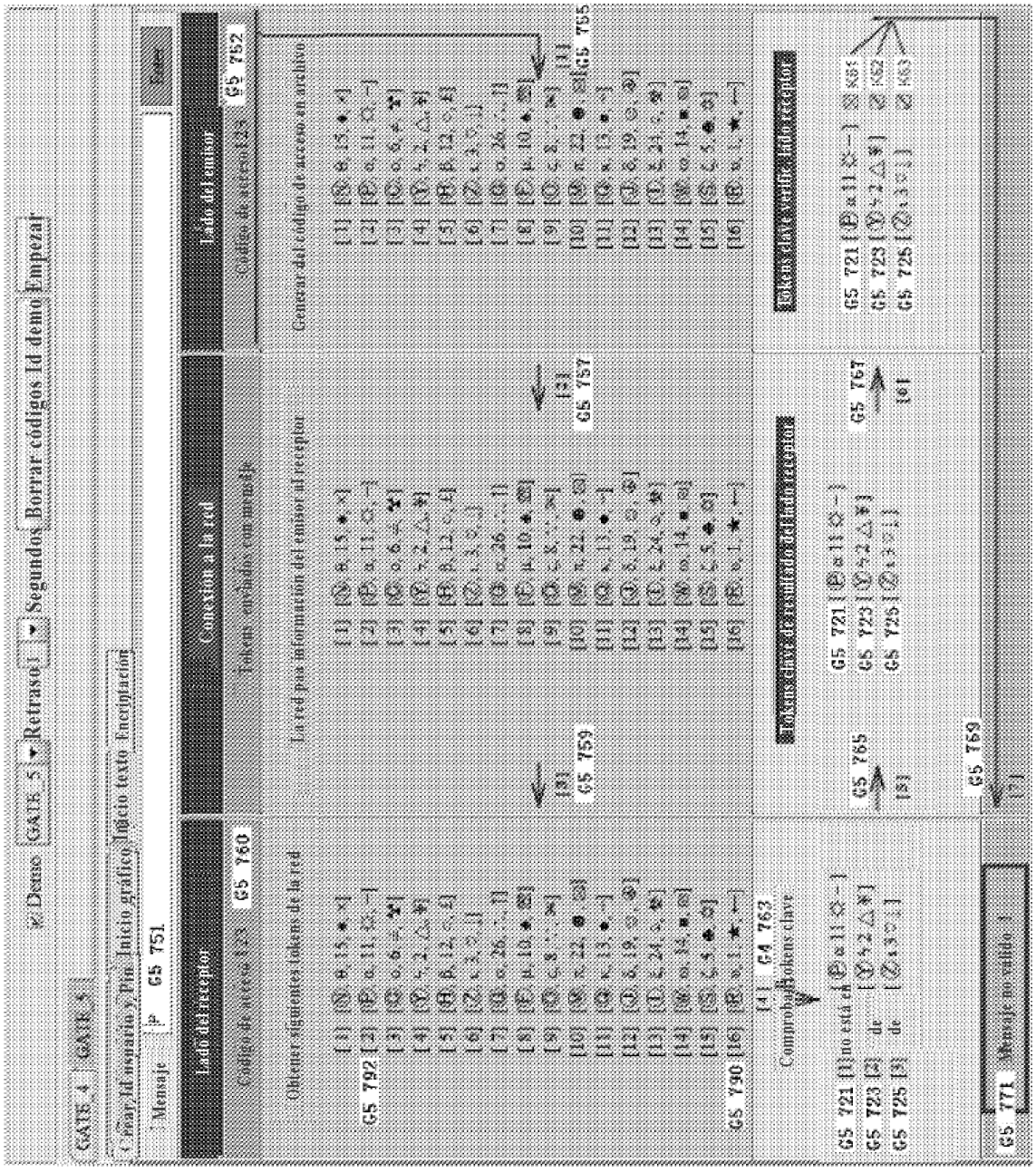


Fig. 21 GATE_5_Encryption