(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 763 090**

(51) Int. CI.:

H04L 9/32 (2006.01)
H04L 9/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **11.07.2017 E 17180837 (1)**

(97) Fecha y número de publicación de la concesión europea: **06.11.2019 EP 3429121**

(54) Título: **Generación de firma digital**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
27.05.2020

(73) Titular/es:

NAGRAVISION S.A. (100.0%)
22-24, route de Genève
1033 Cheseaux-sur-Lausanne, CH

(72) Inventor/es:

VILLEGAS, KARINE y
MACCHETTI, MARCO

(74) Agente/Representante:

TOMAS GIL, Tesifonte Enrique

DESCRIPCIÓN

Generación de firma digital

CAMPO

[0001] La presente descripción se refiere a la generación de firmas digitales, en particular aunque no exclusivamente a la generación de firmas en el Algoritmo de Firma Digital (DSA) y el Algoritmo de Firma Digital de Curva Elíptica (ECDSA).

ANTECEDENTES

[0002] El DSA (véase FIPS 186-4, Digital Signature Standard (DSS). National Institute of Standards and Technologies edition, julio 2013, incorporado por referencia aquí para los detalles de las aplicaciones convencionales del DSA) y el ECDSA (véase *ídem* y Public Key Cryptography For The Financial Services Industry: X9-62-2005 The Elliptic Curve Digital Signature ECDSA, American National Standards Institute edition, noviembre 2005, para los detalles de las aplicaciones convencionales del EDSA) son ejemplos de un tipo de algoritmos de firma digital que dependen de una forma del problema de logaritmo discreto. Estos algoritmos usan un grupo de números enteros, por ejemplo el grupo cíclico de orden o, $\mathbb{Z}_p^*$ en el caso del DSA y del grupo abeliano de puntos en una curva elíptica sobre un campo F_q en el caso del ECDSA, para los que la aplicación repetida m veces de la operación de grupo, o potenciación, es más fácil de computar que su inverso, el logaritmo. Por ejemplo, en el caso del DSA, la operación de grupo es una multiplicación módulo p, la operación o potenciación repetida es x^k mod p y el logaritmo k es $\log_x x^k$. En la patente FR3027752 y Fan, J *et al*, 2010 IEEE HOST Symposium, pp76-87 se discuten contramedidas contra ataques de canal lateral en las aplicaciones del ECDSA .

[0003] Mediante una elección adecuada de parámetros, se puede asegurar que la computación del logaritmo sea computacionalmente difícil hasta el punto de ser intratable en la práctica. Esto permite que estos algoritmos definan una clave pública D como el resultado de la aplicación repetida d veces de la operación de grupo, la potenciación por d, donde d es una clave privada secreta usada para generar una firma para un mensaje y D puede, a continuación, ser usada por un receptor del mensaje para verificar la firma. De este modo, en estos algoritmos, el problema del logaritmo discreto –ya sea en el contexto de elevar una base a una como en el DSA o multiplicación de puntos como en el ECDSA (véase más adelante)– se usa para distribuir una clave pública que se puede usar para verificar una firma generada con una clave privada correspondiente. La dificultad matemática del problema de logaritmo discreto asegura que un atacante no pueda tener acceso a la clave privada a partir de los datos disponibles para la verificación de la firma.

[0004] Un atacante que quiera obtener la clave privada, por ejemplo para forjar firmas válidas aparentemente procedentes del propietario de la clave privada, puede intentar resolver matemáticamente el problema de logaritmo discreto para recuperar la clave privada. Los intentos de llevar esto a cabo así se pueden frustrar mediante una elección adecuada de los parámetros del algoritmo para asegurar que no es computacionalmente posible recuperar la clave privada en la práctica. Sin embargo, aunque el propietario de la clave privada sea diligente en la elección de los parámetros del algoritmo, si un atacante tiene acceso a un dispositivo que usa la clave privada para generar una firma, se puede conseguir información acerca de los números aleatorios usados en la computación de la firma y, en última instancia, la clave privada usando ataques físicos; si se consigue información suficiente de esta manera, la clave privada se puede recuperar.

[0005] Ejemplos de tales ataques físicos incluyen Análisis de Canal Lateral (SCA, por sus siglas en inglés) tales como Ataque de Potencia Única/Electromagnético (SPA/SEMA); Ataque Template/Horizontal (TA/HA); y Ataque de Potencia de Correlación/Electromagnético (CPA, CEMA), Análisis de fallos (FA), o ambos combinados. En Cryptography and Security: From theory to Applications, el artículo de Junfeng Fan e Ingrid Verbauwhede: An Updated Survey on Secure ECC Implementations: Attacks, Countermeasures and Cost propone una visión de conjunto de las posibles amenazas.

[0006] Los algoritmos actuales que presentan esquemas de firmas digitales como, por ejemplo, DSA y ECDSA, muestran varias debilidades a los ataques físicos. Sería deseable generar firmas digitales con una menor vulnerabilidad a tales ataques físicos.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

[0007] A continuación, se describirán formas de realización mediante ejemplos para ilustrar aspectos de la divulgación con referencia a los dibujos anexos donde:

la figura 1 ilustra un algoritmo de generación de firma digital basado en operaciones en un grupo;
la figura 2 ilustra un algoritmo de verificación de firma digital correspondiente;
la figura 3 ilustra una modificación del algoritmo de generación de firma digital;

la figura 4 ilustra un dispositivo de computación para implementar los algoritmos y métodos descritos; y
la figura 5 ilustra una implementación específica de un circuito de procesamiento criptográfico.

DESCRIPCIÓN DETALLADA DE LAS FORMAS DE REALIZACIÓN

[0008] En general, aspectos de la divulgación mejoran la resiliencia a los ataques físicos de los algoritmos de firma digital del mismo tipo que el DSA y el ECDSA reordenando las computaciones de modo que nunca se procese información secreta como la clave privada solamente con información no secreta como el mensaje que se va a firmar, un *hash* del mensaje que se va a firmar o una parte de la firma producida. En algunas formas de realización, adicionalmente, una clave efímera, un número aleatorio usado en el algoritmo y generado de nuevo para cada firma, se manipula de modo que el valor real subyacente a las computaciones no se use nunca realmente. Individualmente y en combinación, estas medidas aseguran que los datos privados se usen solo en combinación con números aleatorios de modo que se reduce la vulnerabilidad a los ataques físicos. El método comprende la computación de una firma (r,s), donde s se calcula usando tres variables aleatorias intermedias. El método también puede comprender el cálculo de una clave efímera usada en la computación como un producto de dos números aleatorios.

[0009] Antes de tratar estos aspectos detalladamente, será útil considerar la clase general de algoritmos de firma digital basados en operaciones en grupos algebraicos y, en particular, grupos abelianos, a los que pertenecen el DSA y el ECDSA. Con referencia a la figura 1, tales algoritmos toman como entrada en una primera etapa 102 un mensaje m por firmar, una función *hash* $h_\tau(\cdot)$ por aplicar al mensaje, al igual que parámetros de dominio que incluyen un generador g del grupo y el orden del grupo generado o y cualesquiera otros parámetros requeridos por el algoritmo específico. La función *hash* $h_\tau(\cdot)$ es una función *hash* criptográfica que produce bits truncados en la longitud de bit de o. En la etapa 104, se obtiene la clave privada $d \in [1;o-1]$, destacando que la clave pública del algoritmo se obtiene a partir de la clave privada d mediante la aplicación d veces de la operación de grupo al generador: D=g*d, donde *d designa la operación de grupo aplicada d veces.

[0010] En la etapa 106 se genera una clave efímera k como un número aleatorio o pseudoaleatorio con $k \in [1;o-1]$ y una primera parte r de la firma digital se genera en la etapa 108: $r=(g^*k)_x \bmod o$, donde $(g^*k)_x$ es al menos una porción de (g*K), por ejemplo la coordenada x de este punto en ECDSA, o el propio (g*k) en el caso del DSA –véase más adelante para una descripción detallada de la correspondencia entre el algoritmo generalizado descrito y el DSA y el ECDSA–. Si r es cero se genera un nuevo k, de otro modo, se genera un resumen e del mensaje m en la etapa 110: $e=h_\tau(m)$ y se genera una segunda parte s de la firma digital en la etapa 112: $s=k^{-1}(e+d\cdot r) \bmod o$, donde k^{-1} es el inverso modular de k. A menos que s sea cero, en cuyo caso el algoritmo comienza nuevamente en la etapa 106 para generar un k- diferente, se devuelve la firma digital (r,s) que comprende ambas partes r y s.

[0011] Con referencia a la figura 2, el algoritmo de verificación de firma generalizado correspondiente comprende las primeras etapas 202 y 204 en las que se obtienen respectivamente la función *hash* $h_\tau(\cdot)$, la clave pública D, los parámetros de dominio y el mensaje m y la firma (r,s). La etapa 206 verifica que r y s están dentro del rango y termina la verificación con un fallo de verificación en la etapa 218 en caso contrario. De otro modo, el resumen e se genera a partir del mensaje m usando la función *hash* en la etapa 208. El inverso de la primera parte de la firma se utiliza para calcular variables intermedias u_1 y u_2 en las etapas respectivas 210 y 212 mediante el módulo de multiplicación o con, respectivamente, e y r. Una variable de verificación v se genera a partir de u_1 y u_2 en la etapa 214, tal y como se ha detallado en la figura 2. Si la variable de verificación coincide con r, entonces la firma se verifica en la etapa 216, por ejemplo devolviendo una bandera adecuada; de otro modo, la verificación de firma falla en la etapa 218.

[0012] Con referencia a la figura 3, en algunas formas de realización, las etapas 106 y 112 se modifican tal y como se describirá ahora para dificultar la obtención de información acerca de información secreta, en particular la clave privada d, usando ataques físicos. Deberá entenderse que cada modificación en sí misma mejora la resiliencia a los ataques y que se pueden combinar para mejorar adicionalmente la resiliencia. De este modo, las formas de realización abarcan algoritmos modificados con solo la etapa 106 modificada, solo la etapa 112 modificada o ambas etapas modificadas. En aras de una mayor concisión ahora solo se describirá la última detalladamente, dado que las formas de realización se obtienen simplemente omitiendo una u otra modificación. Las etapas 102, 104, 108 y 110 no están modificadas y, por lo tanto, las etapas correspondientes 302, 304, 308 y 310 no se describen nuevamente. Del mismo modo, el algoritmo de verificación de firma descrito previamente con referencia a la figura 2 sigue siendo aplicable. Nótese que las etapas del método se pueden reordenar para dificultar los ataques físicos implementando computaciones relacionadas lógicamente en momentos diferentes o agrupadas en ubicaciones diferentes, por ejemplo.

[0013] En algunas formas de realización, a continuación, la etapa 106 se modifica en el sentido de que en la etapa 306 se generan dos números aleatorios o pseudoaleatorios $k_1,k_2 \in [1;o-1]$ y la clave efímera k se computa como el producto de estos dos números aleatorios. Esto reduce la posibilidad de que un ataque FA tenga éxito ya que el generador de número aleatorio tendría que manipularse dos veces para reducir la entropía del la k resultante. Asimismo, esto aumentaría también el número de bits por recuperar utilizando SPA/SEMA, así como el número de

bits que necesitarían ser recuperados utilizando un rastro de SPA/TA. De hecho, en ambos casos, más de la mitad de los bits de k necesitarían ser recuperados de la etapa 308 para romper la clave privada que tiene más bits que el tamaño completo de k en la etapa 108. Del mismo modo, se necesitaría recuperar más bits para un ataque en la etapa 318 (véase más adelante) que para un ataque en la etapa 112 anteriormente descrita, nuevamente más del tamaño completo de k en la etapa 108.

[0014] En algunas formas de realización, la etapa 112 se modifica adicional o alternativamente dividiéndola en tres etapas, donde cada una calcula en efecto una variable aleatoria. En la etapa se computa 312 $s_0=k^j \cdot d \bmod o$, con j como un número entero positivo, por ejemplo unitario en algunas formas de realización. Cabe destacar que, aunque la computación implica la clave secreta d, el resultado es, de hecho, un número aleatorio, debido a la multiplicación con k o una potencia de número entero de la misma. En la etapa 314, se calcula otra variable aleatoria $s_1 = r \cdot s_0 \bmod o$. La variable s_1 es una variable aleatoria ya que s_0 es una variable aleatoria. En la etapa 316, se computa $s_2= k^j \cdot e \bmod o$, nuevamente una variable aleatoria debido a la multiplicación con k^j. Finalmente, en la etapa 318, se usan s_1 y s_2 para computar $s=k^{-(j+i)}(s_1+s_2) \bmod o$, donde $k^{-(j+1)}$ es el módulo o computado, por ejemplo usando un algoritmo euclidiano extendido con $k^2 \bmod o$ para encontrar su inverso o computando $k^{-(j+1)}=k^{-o-(j+2)} \bmod o$ usando la potenciación modular y el teorema de Fermat. Cabe destacar que, aunque la computación de $k^{-(j+1)}$ se podría interferir para producir una constante conocida por el atacante, la etapa 318 ya no se puede explotar para encontrar d, ya que s_1 y s_2 son variables aleatorias. Además, por la misma razón, los ataques basados en la reducción de la entropía de k para esta fase o en el hallazgo de r y, por lo tanto d en la etapa 112, no son posibles con estas etapas modificadas.

[0015] Con s computado en la etapa 318, la etapa 320 corresponde a la etapa 112 y se devuelve la firma (r, s) si s es no nulo. De otro modo, el algoritmo se reinicia en la etapa 306 tal y como se ha descrito previamente con referencia a la figura 1 y la etapa 106.

[0016] La resiliencia a los ataques mejorada del algoritmo generalizado descrito previamente con referencia a la figura 3 también se puede aplicar a los casos especiales del DSA, basado en el grupo cíclico y el ECDSA basado en el grupo del grupo abeliano de puntos en una curva elíptica definida sobre el campo finito F_q, y la siguiente tabla expone la correspondencia entre las variables, los parámetros y las operaciones mencionados previamente y las variables, parámetros y operaciones correspondientes en el DSA y el ECDSA, de modo que el experto en la materia recupera fácilmente los casos especiales sustituyendo las variables, los parámetros y las operaciones respectivos en la descripción anterior de la siguiente manera:

	DSA	ECDSA
Parámetros dominio DOMINIO	- p - un número primo que define el grupo $\mathbb{Z}_p^*$ de números enteros módulo p con la operación de multiplicación - q - el orden del subgrupo definido por el generador a - a - el generador	- q - característica de campo finito F_q; q es 2^m con m un número primo o un número primo >3 - a,b - parámetros que definen una curva elíptica E definida sobre el campo finito F_q; $a,b \in F_q x F_q$ - G - punto generador en E - n - el orden de G - h - cofactor $h=\#E(F_q)/n$; $\#E(F_q)$ es el número de puntos en E
Generador g	a	Punto G: (x_g,y_g) en curva E; $(x_g,y_g) \in F_q x F_q$
Orden de grupo o	q	n
Operación de grupo	Multiplicación modular con módulo igual a p o q	Adición de punto de G en E
Operación de grupo repetida k veces g*k (potenciación)	$a^k \bmod p$	Multiplicación de punto escalar [k]G
Componente para cálculo de r: $(g*k)_x$	$a^k \bmod p$	r_x con $[k]G \rightarrow (r_x;r_y)$
Clave pública D para clave privada d	$D=a^d \bmod p$	Punto en E: $D=[d]G$ o $D=[d^{-1} \bmod q]G$
Función *hash* $h_\tau(\cdot)$	Función *hash* criptográfica truncada en longitud de bit de q	Función *hash* criptográfica truncada en longitud de bit de n

[0017] De este modo, en resumen, aspectos de la divulgación proporcionan un método para firmar digitalmente un mensaje, donde el método comprende:

a) obtener parámetros de dominio que definen un grupo asociado a una operación de grupo y generado por un generador de orden o, donde los parámetros de dominio comprenden el generador g y el orden o;
b) obtener una clave de número entero privada d en el intervalo [1,o-1], donde una clave pública correspondiente es el resultado de aplicar la operación de grupo al generador d veces;
c) generar un número aleatorio k;

d) generar una primera parte de firma r aplicando la operación de grupo al generador g k veces y determinando el resto módulo o de al menos una parte del resultado;
e) generar un valor inverso $k^{-(j+1)}$ evaluando el inverso módulo o de k elevado a la potencia de j+1, donde j es un número entero positivo, por ejemplo unitario, y el producto de $k^{-(j+1)}$ y k elevado a la potencia de j+1 evaluado módulo o es unitario;
f) generar un valor s0 por evaluación, módulo o, del producto de k elevado a la potencia de j y d;
g) generar un valor s1 por evaluación, módulo o, del producto de r y s0;
h) generar un valor de resumen de mensaje del mensaje usando una función *hash* criptográfica;
i) generar un valor s2 por evaluación, módulo o, del producto de k elevado a la potencia de j y el valor de resumen de mensaje;
j) generar una segunda parte de firma s por evaluación, módulo o, del producto de $k^{-(j+1)}$ y la suma de los dos valores s1 y s2; y
k) producir r y s como la firma, donde el método comprende asegurar que r y s son no nulos.

[0018] En algunas formas de realización, k se genera generando dos números enteros aleatorios k1 y k2, cada uno en el rango de [1,o-1] y evaluando el producto de k1 y k2.

[0019] En algunas formas de realización, j se elige para que sea unidad, j=1. Este ajuste de j se usa, por ejemplo, en aplicaciones convencionales del DSA y el ECDSA. Sin embargo, cabe destacar que j se puede establecer en cualquier número entero positivo. Esto se puede llevar a cabo sin afectar a la verificación de firma, ya que la etapa 318 elimina eficazmente cualquier potenciación adicional que corresponda a j>1. Por ejemplo, j se puede generar de forma aleatoria, por ejemplo escogido de forma aleatoria a partir de un conjunto finito de valores J tales como los números enteros positivos hasta J: { 1,2,3; J}. J se puede elegir para que sea cualquier número entero positivo adecuado teniendo en cuenta la capacidad de procesamiento, por ejemplo J=2, J=3 o J=4. Por supuesto, el valor aleatorio de j también puede elegirse de cualquier otra manera adecuada. El valor de j se puede cambiar de esta manera, o basado en un programa fijo o programa predeterminado de otro modo. El valor de j se puede cambiar entre repeticiones sucesivas del algoritmo de generación de firma, cada dos repeticiones, cada tres repeticiones, o cada enésima repetición, en ciertos momentos predeterminados o de cualquier otro modo. Al cambiar el valor de j, la huella digital física de la ejecución del algoritmo cambia en consecuencia. Esto puede ayudar a dificultar los ataques de canal lateral.

[0020] En algunas formas de realización, el grupo puede ser el conjunto de puntos generados por el generador g y la operación de grupo en una curva elíptica definida sobre un campo finito F_q. En estas formas de realización, el generador g es un punto G en la curva elíptica, el orden o es el orden n de los puntos de grupo en la curva elíptica generada por G con la operación de grupo de adición de puntos en la curva elíptica, y la al menos parte del resultado en la etapa d es un componente de coordenada del punto en la curva elíptica obtenido por multiplicación de punto escalar de k con G. En algunas formas de realización, las etapas a, b, d y k se implementan conforme a las etapas correspondientes del ECDSA.

[0021] En algunas formas de realización, el grupo puede ser un subgrupo de orden q en el grupo Zp* de números enteros positivos módulo un número primo p, definido por una base de número entero positivo a como el generador G. En estas formas de realización, la operación de grupo es multiplicación modular y la al menos parte del resultado en la etapa d es a^k módulo p. En algunas formas de realización, las etapas a, b, d y k se implementan conforme a las etapas correspondientes del DSA.

[0022] Las etapas del método pueden, en algunas formas de realización, estar dispuestas en vista de la resiliencia a los ataques físicos o en cualquier otro orden temporal de acuerdo con las computaciones o cualquier otra distribución entre componentes de procesamiento. Por ejemplo, tanto la etapa f como la i se pueden llevar a cabo antes de la etapa d y/o la etapa e se puede llevar a cabo antes de la etapa d. Una o más de las etapas e, f e I se puede llevar a cabo usando un primer circuito de procesamiento, que lleva a cabo una o más de las etapas restantes usando un segundo circuito de procesamiento diferente del primer circuito.

[0023] Los aspectos de la divulgación se extienden además a un procesador o dispositivo de computación configurado para implementar un método tal y como se ha descrito anteriormente. En algunas formas de realización, un primer circuito de procesamiento se configura para implementar una o más de las etapas e, f e I y un segundo circuito de procesamiento se configura para implementar una o más de las etapas restantes. Las formas de realización incluyen, por ejemplo, una tarjeta inteligente que comprenda tal procesador y a un aparato de acceso al medio que comprenda tal procesador. Los aspectos de la divulgación se extienden además a un producto de programa informático que comprende instrucciones de computación codificadas que, cuando se ejecutan en un procesador, implementan un método tal y como se ha descrito anteriormente y/o a uno o más medios legibles por ordenador no transitorios que codifican tales instrucciones.

[0024] La figura 4 ilustra un diagrama de bloques de una implementación de un dispositivo de computación 400 en el cual se puede ejecutar un conjunto de instrucciones para provocar que el dispositivo de computación ejecute cualquiera o más de las metodologías tratadas aquí. En aplicaciones alternativas, el dispositivo de computación puede estar conectado (por ejemplo en red) a otras máquinas en una Red de Área Local (LAN), una intranet, una

extranet o Internet. El dispositivo de computación puede operar en calidad de un servidor o una máquina cliente en un entorno de red cliente-servidor, o como una máquina par en un entorno de red entre pares (o distribuido). El dispositivo de computación puede ser un ordenador personal (PC), una tableta, un descodificador (STB), un Asistente Digital Personal (PDA), un teléfono celular, un dispositivo web, un servidor, un router de red, un interruptor o puente o cualquier máquina capaz de ejecutar un conjunto de instrucciones (secuencial o de otro modo) que especifique acciones que deben ser realizadas por dicha máquina. Además, aunque solo se ilustre un dispositivo de computación, el término "dispositivo de computación" debe también ser entenderse incluyendo cualquier grupo de máquinas (por ejemplo ordenadores) que ejecutan individual o conjuntamente un conjunto (o múltiples conjuntos) de instrucciones para llevar a cabo cualquiera o más de las metodologías tratadas aquí. El dispositivo de computación puede comprender o realizarse en un elemento seguro para proporcionar computaciones seguras aisladas de otras partes de un dispositivo de computación en el que se integra el elemento seguro. El dispositivo de computación puede ser un sistema en chip.

[0025] El dispositivo de computación 400 ejemplar incluye un dispositivo de procesamiento 402, una memoria principal 404 (por ejemplo una memoria de solo lectura (ROM), una memoria *flash*, una memoria de acceso aleatorio dinámica (DRAM) tal como una DRAM sincrónica (SDRAM) o una Rambus DRAM (RDRAM), etc.), una memoria estática 406 (por ejemplo una memoria *flash*, una memoria de acceso aleatorio estática (SRAM), etc.), y una memoria secundaria (por ejemplo un dispositivo de almacenamiento de datos 418), que comunican entre sí mediante un bus 430.

[0026] El dispositivo de procesamiento 402 representa uno o más procesadores universales tales como un microprocesador, unidad central de procesamiento o similar. Más particularmente, el dispositivo de procesamiento 402 puede ser un microprocesador de ordenador de conjunto de instrucciones complejo (CISC), un microprocesador de ordenador de conjunto de instrucciones reducido (RISC), un microprocesador de palabra de instrucción muy larga (VLIW), un procesador que implemente otros conjuntos de instrucciones o procesadores que implementen una combinación de conjuntos de instrucciones. El dispositivo de procesamiento 402 también puede ser uno o más dispositivos de procesamiento especializados tales como un circuito integrado de aplicación específica (ASIC), una matriz de puertas programable en campo (FPGA), un procesador de señales digitales (DSP), un procesador de red o similar. El dispositivo de procesamiento 402 está configurado para ejecutar la lógica de procesamiento (instrucciones 422) para llevar a cabo las operaciones y etapas tratadas aquí.

[0027] El dispositivo de computación 400 puede incluir además un dispositivo de interfaz de red 408. El dispositivo de computación 400 puede incluir también una unidad de visualización de vídeo 410 (por ejemplo, una pantalla de cristal líquido (LCD) o un tubo de rayos catódicos (CRT)), un dispositivo de entrada alfanumérica 412 (por ejemplo, un teclado o una pantalla táctil), un dispositivo de control del cursor 414 (por ejemplo, un ratón o una pantalla táctil) y un dispositivo de audio 416 (por ejemplo un altavoz).

[0028] El dispositivo de almacenamiento de datos 418 puede incluir uno o más medios de almacenamiento legibles por máquina (o más específicamente uno o más medios de almacenamiento legibles por ordenador no transitorios) 428 en los que están almacenados uno o más conjuntos de instrucciones 422 que concretan cualquiera o más de las metodologías o funciones descritas aquí. Las instrucciones 422 también pueden residir, completa o al menos parcialmente, en la memoria principal 404 y/o en el dispositivo de procesamiento 402 durante su ejecución por el sistema informático 400, donde la memoria principal 404 y el dispositivo de procesamiento 402 constituyen también medios de almacenamiento legibles por ordenador.

[0029] Con referencia a la figura 5, una implementación de *hardware* específica 500 comprende una CPU 502, una RAM 504, una ROM o una memoria FLASH 506 (o cualquier otra disposición de memoria adecuada) y una pluralidad de (dos o más) subprocesadores criptográficos o conjunto de circuitos de procesamiento 508, 510, 512 que se comunican a través de un bus 514. La implementación 500 puede estar configurada como procesador criptográfico o elemento de seguridad separados aislados, implementados, por ejemplo, en un dispositivo de computación 400 o de forma autónoma, y que se comunican a través de una interfaz segura adecuada. De forma alternativa, los subprocesadores criptográficos 508, 510, 512 y/o componentes de memoria de seguridad 504 o 506 pueden estar integrados de cualquier otra forma en un dispositivo de computación tal como el dispositivo de computación 400. Los procesos criptográficos están distribuidos entre los subprocesadores criptográficos 508, 510, 512 bajo control de la CPU 502 o en una disposición entre pares entre los subprocesadores, con el propósito de acelerar el procesamiento y/o aumentar la seguridad. Por ejemplo, se pueden distribuir etapas de los métodos descritos entre los procesadores tal y como se ha descrito anteriormente. Por ejemplo, las etapas 306 y 312 se pueden ejecutar mediante el subprocesador 508, las etapas 308 y 316 se pueden ejecutar mediante el subprocesador 510, las etapas 310 y 312 se pueden ejecutar mediante el subprocesador 512 y las etapas restantes de la figura 3 se pueden ejecutar mediante la CPU 502.

[0030] Los diferentes métodos anteriormente descritos se pueden implementar mediante un programa informático. El programa informático puede incluir código máquina dispuesto para dar instrucciones a un ordenador para que lleve a cabo las funciones de uno o más de los diferentes métodos anteriormente descritos. El programa informático y/o el código para llevar a cabo tales métodos pueden proporcionarse a un equipo, tal como un ordenador, en uno o más medios legibles por ordenador o, más generalmente, un producto de programa informático. Los medios

legibles por ordenador pueden ser transitorios o no transitorios. El uno o más medios legibles por ordenador podría ser, por ejemplo, un sistema electrónico, magnético, óptico, electromagnético, infrarrojo o semiconductor, o un medio de propagación para la transmisión de datos, por ejemplo para descargar el código a través de Internet. De forma alternativa, el uno o más medios legibles por ordenador podrían tomar la forma de uno o más medios legibles por ordenador físicos tal como una memoria semiconductora o en estado sólido, una cinta magnética, un disquete de ordenador extraíble, una memoria de acceso aleatorio (RAM), una memoria de solo lectura (ROM), un disco magnético rígido y un disco óptico, tal como un CD-ROM, un CD-R/W o un DVD.

[0031] En una implementación, los módulos, componentes y otras características descritos aquí se pueden implementar como componentes discretos o integrados en la funcionalidad de los componentes de *hardware* tales como, ASIC, FPGA, DSP o dispositivos similar.

[0032] Un "componente de *hardware*" es un componente físico tangible (por ejemplo no transitorio) (por ejemplo, un conjunto de uno o más procesadores) capaz de llevar a cabo ciertas operaciones y se puede configurar o disponer de una manera física determinada. Un componente de hardware puede incluir un conjunto de circuitos o lógica dedicado que está permanentemente configurado para ejecutar determinadas operaciones. Un componente de *hardware* puede ser o incluir un procesador especializado, tal como una matriz de puertas programable en campo (FPGA) o un ASIC. Un componente de *hardware* también puede incluir lógica o circuito programable configurado temporalmente mediante *software* para ejecutar determinadas operaciones.

[0033] Por consiguiente, la frase "componente de *hardware*" debería entenderse englobando una entidad tangible que puede estar construida físicamente, configurada permanentemente (por ejemplo cableada) o configurada temporalmente (por ejemplo programada) para operar de una manera determinada o para ejecutar determinadas operaciones descritas en la presente.

[0034] Además, los módulos y componentes se pueden implementar como *firmware* o conjunto de circuitos funcional dentro de dispositivos *hardware*. Además, los módulos y componentes se pueden implementar en cualquier combinación de dispositivos *hardware* y componentes de *software*, o solamente en *software* (por ejemplo, código almacenado o realizado de otro modo en un medio legible por máquina o en un medio de transmisión).

[0035] A menos que se especifique lo contrario, como se desprende de la discusión siguiente, se observa que, a lo largo de la descripción, las expresiones que usen términos tales como "recibir", "determinar", "comparar", "permitir", "mantener", "identificar", "computar", "generar", "obtener" o similar, se refieren a las acciones y procesos de un sistema informático, o dispositivo de computación electrónico similar, que manipula y transforma datos representados como cantidades (electrónicas) físicas en los registros y memorias del sistema informático en otros datos de forma similar representados como cantidades físicas en las memorias o registros del sistema informático u otros dispositivos de almacenamiento, transmisión o visualización de información de este tipo.

[0036] Debe entenderse que la descripción anterior se destina a ser ilustrativa, y no restrictiva. Los expertos en la materia deducirán muchas otras aplicaciones leyendo y comprendiendo la descripción anterior. Aunque la presente descripción se ha descrito con referencia a aplicaciones ejemplares específicas, se reconocerá que la divulgación no está limitada a las aplicaciones descritas, pero se puede llevar a la práctica con modificación y alteración dentro del espíritu y el alcance de las reivindicaciones anexas. Por consiguiente, la especificación y los dibujos se deben considerar en un sentido ilustrativo en lugar de en un sentido restrictivo.

REIVINDICACIONES

1. Método para firmar digitalmente un mensaje, donde el método comprende:

a) obtener (302) parámetros de dominio que definen un grupo asociado a una operación de grupo y generado por un generador de orden o, donde los parámetros de dominio comprenden el generador g y el orden o;
b) obtener (304) una clave de número entero privada d en el intervalo [1,o-1], donde una clave pública correspondiente es el resultado de aplicar la operación de grupo al generador d veces;
c) generar (306) un número aleatorio k;
d) generar (308) una primera parte de firma r aplicando la operación de grupo al generador g k veces y determinando el resto módulo o de al menos una parte del resultado;
e) generar un valor inverso $k^{-(j+1)}$ evaluando el inverso módulo o de k elevado a la potencia de j+1, donde j es un número entero positivo y el producto de $k^{-(j+1)}$ y k elevado a la potencia de j+1 evaluado módulo o es unitario;
f) generar (312) un valor s0 evaluando, módulo o, el producto de k elevado a la potencia de j y d;
g) generar (314) un valor s1 evaluando, módulo o, el producto de r y s0;
h) generar (310) un valor de resumen de mensaje del mensaje usando una función *hash* criptográfica;
i) generar (316) un valor s2 evaluando, módulo o, el producto de k elevado a la potencia de j y el valor de resumen de mensaje;
j) generar (318) una segunda parte de firma s evaluando, módulo o, el producto de $k^{-(j+1)}$ y de la suma de dos valores s1 y s2; y
k) generar (320) r y s como la firma, donde el método comprende asegurar que r y s son no nulos.

2. Método según la reivindicación 1, donde k se genera generando dos números enteros aleatorios k1 y k2, cada uno en el rango de [1,o-1] y evaluando el producto de k1 y k2.

3. Método según la reivindicación 1 o 2, donde j es un número entero positivo generado de forma aleatoria.

4. Método según cualquiera de las reivindicaciones anteriores, donde el grupo es el conjunto de puntos en una curva elíptica definida sobre un campo finito F_q generado por el generador, el generador g es un punto G en la curva elíptica, el orden o es el orden n de los puntos de grupo en la curva elíptica generada por G con la operación de grupo de adición de punto en la curva elíptica, y donde la al menos parte del resultado en la etapa d es un componente de coordenada del punto en la curva elíptica obtenida por multiplicación de punto escalar de k con G.

5. Método según cualquiera de las reivindicaciones anteriores, donde las etapas a, b, d y k se implementan conforme a las etapas correspondientes del ECDSA.

6. Método según cualquiera de las reivindicaciones 1 a 3, donde el grupo es un subgrupo de orden q en el grupo Zp* de números enteros positivos módulo un número primo p, definido por una base de número entero positivo a como el generador g, la operación de grupo es una multiplicación modular y la al menos parte del resultado en la etapa d es a^k módulo p.

7. Método según cualquiera de las reivindicaciones 1 a 3 y 6, donde las etapas a, b, d y k se implementan conforme a las etapas correspondientes del DSA.

8. Método según cualquiera de las reivindicaciones precedentes, donde el método comprende llevar a cabo una o ambas de las etapas f e i antes de la etapa d.

9. Método según cualquiera de las reivindicaciones precedentes, donde el método comprende llevar a cabo la etapa e antes de la etapa d.

10. Método según cualquiera de las reivindicaciones anteriores, donde el método comprende llevar a cabo una o más de las etapas e, f e l usando un primer circuito de procesamiento (508) y llevando a cabo una o más de las etapas restantes usando un segundo circuito de procesamiento (510) diferente del primer circuito.

11. Procesador (400, 500) configurado para implementar un método según cualquiera de las reivindicaciones precedentes.

12. Procesador según la reivindicación 11, donde un primer circuito de procesamiento (508) está configurado para implementar una o más de las etapas e, f e l y un segundo circuito de procesamiento (510) está configurado para implementar una o más de las etapas restantes.

13. Tarjeta inteligente que comprende un procesador según la reivindicación 11 o 12.

14. Aparato de acceso al medio que comprende un procesador según la reivindicación 11 o 12.

15. Producto de programa informático que comprende instrucciones de computación codificadas que, cuando se ejecutan en un procesador, implementan un método tal y como se reivindica en cualquiera de las reivindicaciones 1 a 10.

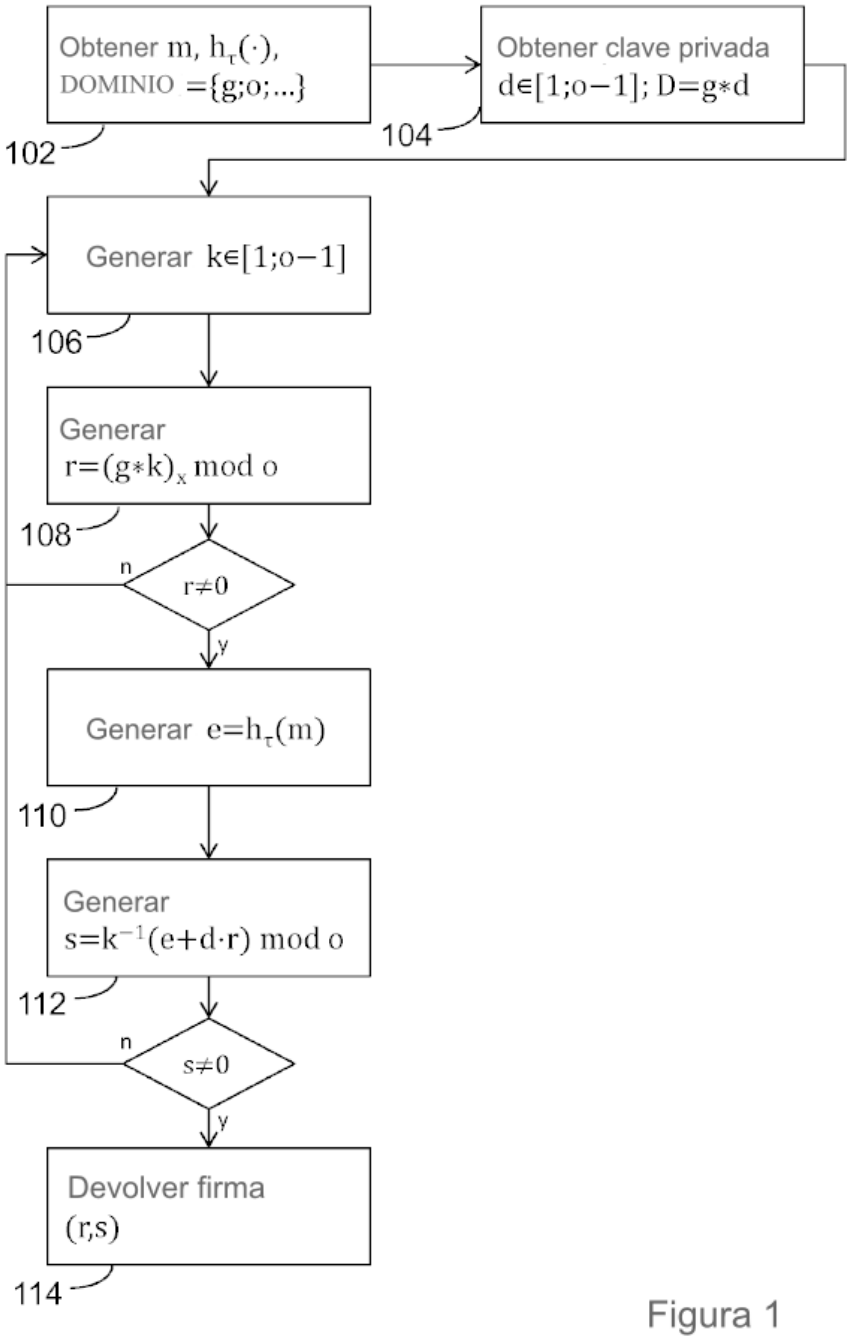
Obtener m, $h_\tau(\cdot)$, DOMINIO ={g;o;...}
102
Obtener clave privada d∈[1;o−1]; D=g∗d
104
Generar k∈[1;o−1]
106
Generar $r=(g*k)_x \bmod o$
108
n
r≠0
y
Generar $e=h_\tau(m)$
110
Generar $s=k^{-1}(e+d\cdot r) \bmod o$
112
n
s≠0
y
Devolver firma (r,s)
114

Figura 1

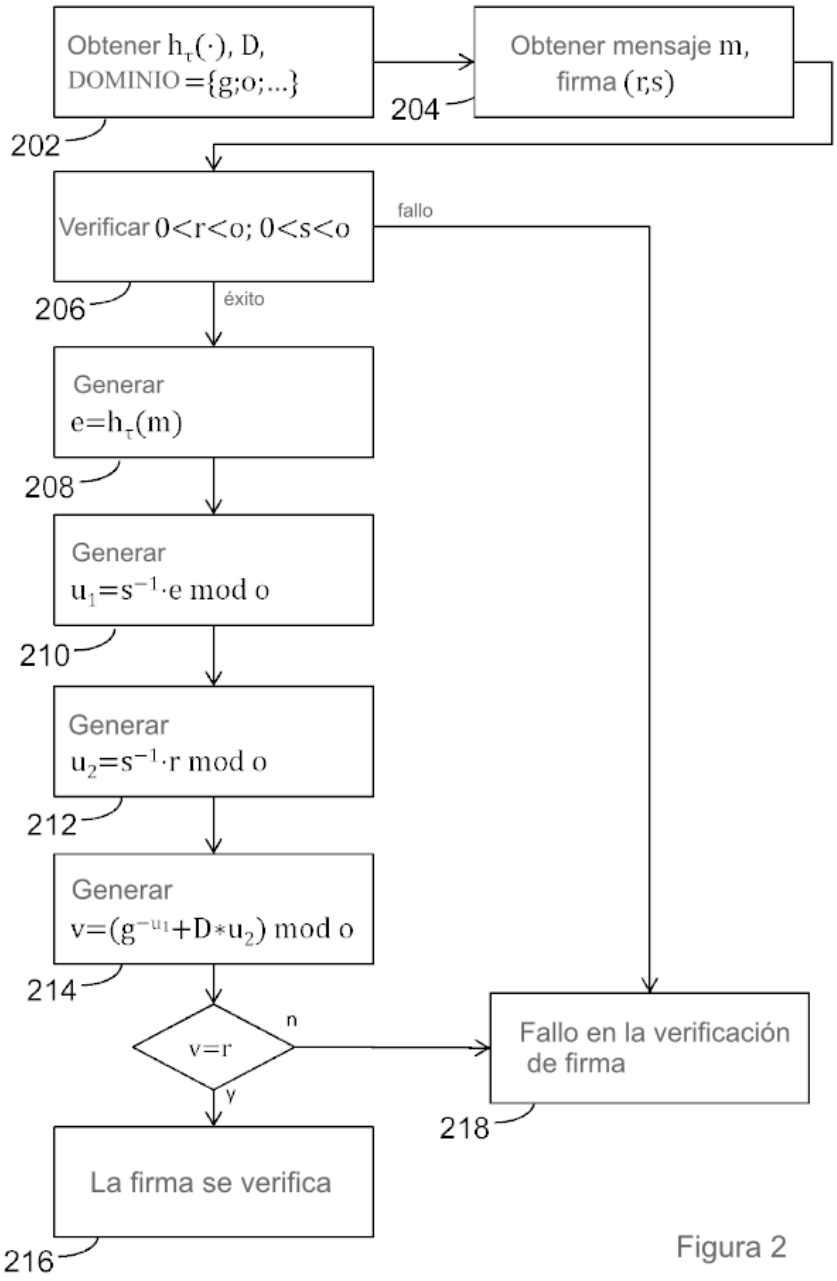
Obtener $h_\tau(\cdot)$, D, DOMINIO $=\{g;o;...\}$
202
Obtener mensaje m, firma (r,s)
204
Verificar $0<r<o$; $0<s<o$
206
fallo
éxito
Generar $e=h_\tau(m)$
208
Generar $u_1=s^{-1}\cdot e \bmod o$
210
Generar $u_2=s^{-1}\cdot r \bmod o$
212
Generar $v=(g^{-u_1}+D*u_2) \bmod o$
214
v=r
n
y
Fallo en la verificación de firma
218
La firma se verifica
216

Figura 2

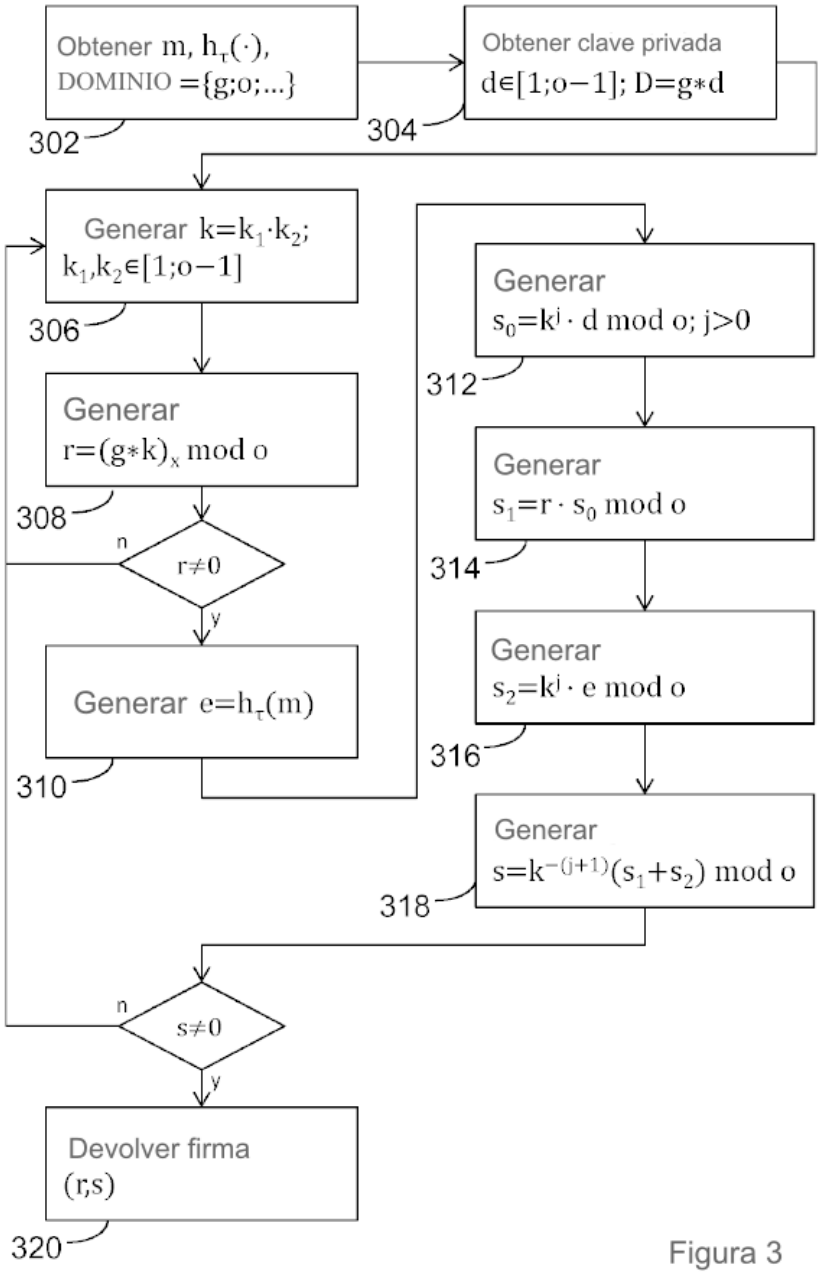
Obtener m, $h_\tau(\cdot)$, DOMINIO ={g;o;...}
302
Obtener clave privada
d∈[1;o−1]; D=g∗d
304
Generar $k=k_1 \cdot k_2$; $k_1,k_2 \in [1;o-1]$
306
Generar $r=(g*k)_x \bmod o$
308
n
r≠0
y
Generar $e=h_\tau(m)$
310
Generar $s_0=k^j \cdot d \bmod o$; j>0
312
Generar $s_1=r \cdot s_0 \bmod o$
314
Generar $s_2=k^j \cdot e \bmod o$
316
Generar $s=k^{-(j+1)}(s_1+s_2) \bmod o$
318
n
s≠0
y
Devolver firma (r;s)
320

Figura 3

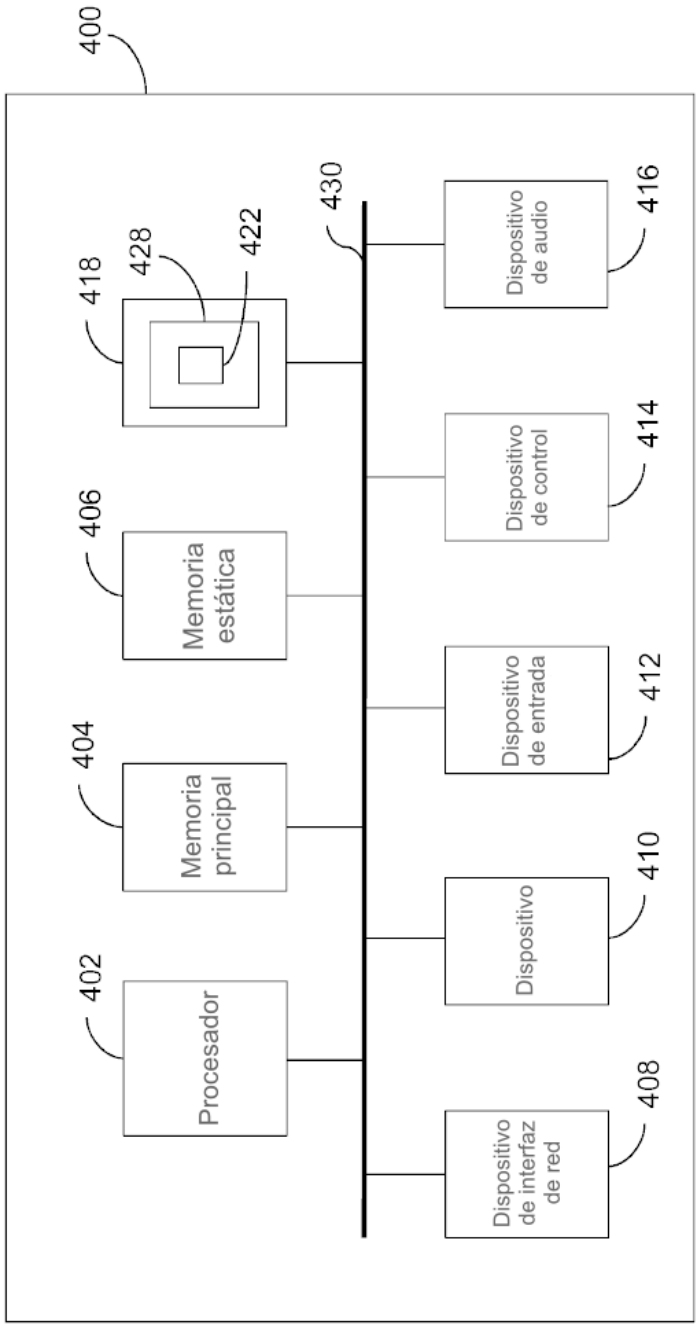
400
430
418
428
422
406
Memoria estática
404
Memoria principal
402
Procesador
Dispositivo de interfaz de red
408
Dispositivo
410
Dispositivo de entrada
412
Dispositivo de control
414
Dispositivo de audio
416

Fig. 4

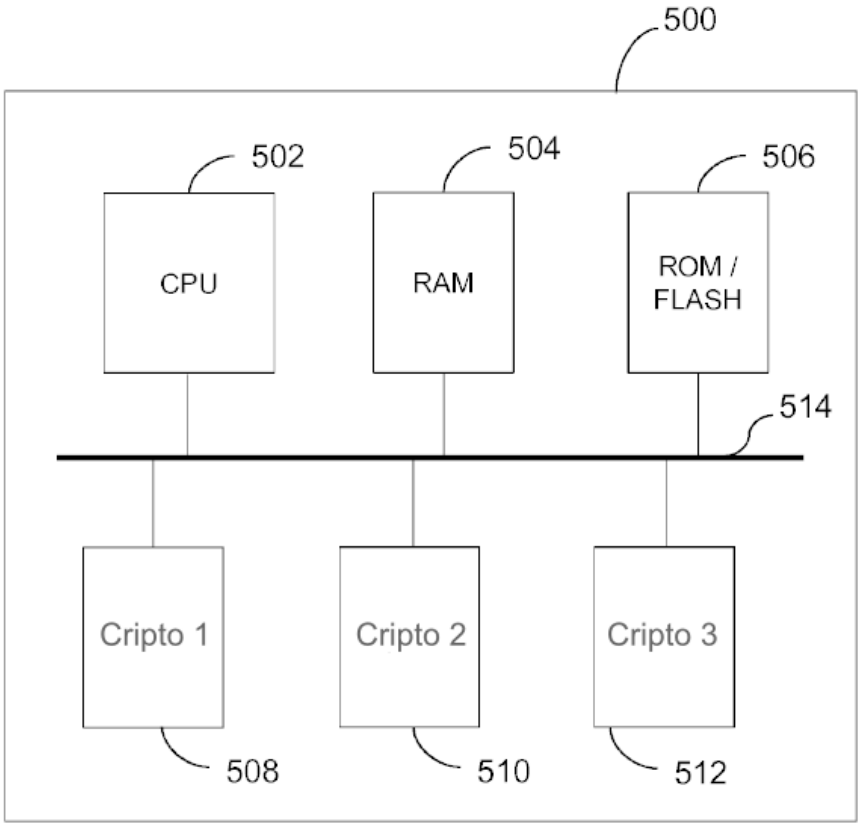
500
502
504
506
CPU
RAM
ROM / FLASH
514
Cripto 1
Cripto 2
Cripto 3
508
510
512

Fig. 5