(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 404 410**

(51) Int. CI.:

G06F 7/72 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **28.06.2006 E 06013333 (7)**

(97) Fecha y número de publicación de la concesión europea: **27.02.2013 EP 1746495**

(54) Título: **Uso de un coprocesador para inversión modular**

(30) Prioridad:

29.06.2005 DE 102005030286

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
27.05.2013

(73) Titular/es:

GIESECKE & DEVRIENT GMBH (100.0%)
PRINZREGENTENSTRASSE 159
81677 MÜNCHEN, DE

(72) Inventor/es:

SEYSEN, MARTIN, DR.

(74) Agente/Representante:

TORNER LASALLE, Elisabet

DESCRIPCIÓN

Uso de un coprocesador para inversión modular

La invención se refiere, en general, al campo técnico de la criptografía y, en especial, a una técnica prevista con fines criptográficos para la inversión modular mediante el uso de un coprocesador. En particular, la invención está prevista para su utilización en soportes de datos portátiles que, por ejemplo, pueden estar configurados como tarjetas con chip en distintas formas de construcción o como módulos de chip.

Como los soportes de datos portátiles deben ser económicos y pequeños, por regla general presentan una capacidad de cálculo sólo relativamente reducida y relativamente poco espacio de memoria. Por otro lado, los soportes de datos portátiles se utilizan a menudo para aplicaciones críticas para la seguridad tales como, por ejemplo, transacciones financieras o funciones de identificación electrónicas. Por ello es deseable realizar en soportes de datos portátiles procedimientos criptográficos seguros (“intensos”).

Una serie de procedimientos criptográficos conocidos con clave pública, por ejemplo los procedimientos conocidos por el nombre de RSA y Diffie-Hellman, contienen cálculos en un grupo multiplicativo con un número entero grande como módulo. Para poder realizar tales procedimientos de manera eficaz se han creado microcontroladores, que además del propio núcleo de procesador presentan un coprocesador criptográfico. Este tipo de coprocesadores están optimizados por regla general para multiplicaciones de números enteros modulares con longitudes de bits de aproximadamente 512 bits a 1024 bits. La patente estadounidense 5.961.578 muestra a modo de ejemplo un microcontrolador de este tipo.

Un desarrollo más reciente es el uso de procedimientos criptográficos, que se basan en curvas elípticas (“procedimientos EC”). A diferencia del procedimiento RSA mencionado anteriormente los procedimientos EC requieren longitudes de bits de clave considerablemente menores de normalmente 160 bits a 256 bits. Sin embargo, en los procedimientos EC otras opciones como la multiplicación modular, concretamente en particular la suma, resta e inversión modular, desempeñan un papel importante.

La duración de ejecución total del procedimiento EC se ve influida en particular mediante las inversiones modulares necesarias. Cuando una inversión modular no se realiza con el apoyo de un coprocesador, por ejemplo puede requerir un factor 100 más de tiempo que una multiplicación modular. Así, por ejemplo, en el procedimiento de firma ECDSA conocido con una longitud de bits de clave de 160 bits a 192 bits se usa aproximadamente el 20% de la duración de ejecución para inversiones modulares. Por tanto, cada aceleración de las inversiones modulares afecta en una medida perceptible sobre la duración de ejecución total del procedimiento de firma ECDSA y otros procedimientos EC.

Así, existe la necesidad de un procedimiento eficaz para la inversión modular. En particular existe la necesidad de utilizar coprocesadores en sí conocidos de manera eficaz para la inversión modular.

Un procedimiento conocido y a menudo utilizado para la inversión modular es el procedimiento euclídeo ampliado, que se basa en el procedimiento ya propuesto por Euclides para la determinación del máximo común divisor (gcd = *greatest common divisor*) de dos valores de entrada *u,* v. Mediante una operación adicional, en el procedimiento euclídeo ampliado se determinan además del valor gcd(*u,* v) también los coeficientes λ, μ de una combinación lineal de los valores de entrada *u,* v con $\lambda u - \mu v = \gcd(u, v)$ y $|\lambda| < v$. Como existe un inverso modular u^{-1} del valor *u* con el módulo *v* sólo para gcd(*u,* v) = 1, se aplica $\lambda u - \mu v = 1$ y por tanto $u^{-1} = \lambda \pmod{v}$ para el coeficiente λ calculado mediante el procedimiento euclídeo ampliado.

El procedimiento euclídeo no ampliado y el procedimiento euclídeo ampliado se describen como los algoritmos A y X en las páginas 334 - 343 del libro “The Art of Computer Programming” de D. E. Knuth, tomo 2, 3ª edición, Addison-Wesley, 1997. Por las operaciones adicionales necesarias el esfuerzo de cálculo necesario para el procedimiento euclídeo ampliado es claramente mayor que el esfuerzo para el procedimiento euclídeo no ampliado. Por tanto es deseable proporcionar un procedimiento para la inversión modular, que al menos para algunos casos de aplicación requiera menos tiempo de cálculo que el procedimiento euclídeo ampliado.

Por tanto la invención tiene como objetivo proponer una técnica para la inversión modular, mediante la que pueda utilizarse un coprocesador de una manera especialmente favorable.

Según la invención este objetivo se soluciona por completo o en parte mediante un procedimiento con las características de la reivindicación 1, un producto de programa informático según la reivindicación 6 y un soporte de datos portátil según la reivindicación 7. Las reivindicaciones dependientes definen características opcionales de algunas configuraciones de la invención.

La invención parte de la idea básica de realizar etapas de procesamiento de un procedimiento euclídeo no ampliado con valores, que presentan una longitud de bits mayor que el módulo y/o el valor que va a invertirse y que por ello en

el presente documento se denominan “valores ampliados”. Es un hallazgo sorprendente que en estos valores ampliados puede incluirse información a partir de la que finalmente puede determinarse fácilmente el inverso deseado, aún cuando no se realice el procedimiento euclídeo ampliado.

En general mediante la invención se aumenta la longitud de bits de los cálculos, aunque se reduce el número de operaciones de cálculo en comparación con un procedimiento euclídeo ampliado. Así, por ejemplo, en algunas formas de realización de la invención puede dividirse por la mitad aproximadamente el número de las operaciones necesarias, mientras que aproximadamente se duplica la longitud de bits. En el caso de constelaciones como las denominadas al principio, en las que, por ejemplo, debe realizarse un procedimiento EC con una longitud de bits de clave de 160 bits a 256 bits mediante el uso de un coprocesador, que está configurado para cálculos con 512 bits ó 1024 bits, sin embargo, la longitud de bits mayor no desempeña ningún papel, mientras que el número reducido de operaciones lleva a una reducción aproximadamente lineal del tiempo de cálculo.

En algunas configuraciones en los valores ampliados la información del valor de entrada y/o del módulo se ha desplazado a posiciones de bit más elevadas. Esto puede ocurrir mediante una multiplicación por un factor de ampliación, que a su vez puede ser una segunda potencia o un múltiplo de una segunda potencia. Se entiende que esta multiplicación puede implementarse de manera eficaz, por ejemplo mediante una operación de desplazamiento. Según la invención al menos uno de los valores ampliados contiene además una perturbación en una posición de bit de valor bajo. Así, por ejemplo, puede fijarse el bit de valor más bajo del primer valor ampliado.

Las etapas de procesamiento corresponden según la invención a las etapas de procesamiento de un procedimiento euclídeo, por ejemplo de un procedimiento euclídeo clásico no ampliado o de un procedimiento euclídeo con la configuración según Lehmer. Sin embargo, esto no quiere decir necesariamente que las etapas de procesamiento se realicen con la misma frecuencia que en el caso de un procedimiento euclídeo. Más bien, en algunas configuraciones el procedimiento descrito en el presente documento se finaliza antes de lo que sería el caso en un procedimiento euclídeo habitual.

Según la invención el coprocesador está optimizado para cálculos de números enteros con al menos la longitud de bits aumentada, por ejemplo, duplicada. Este puede ser el caso por ejemplo cuando el coprocesador está configurado para cálculos RSA o procedimientos similares y se utiliza el procedimiento según la invención para cálculos EC.

El producto de programa informático según la invención presenta instrucciones de programa para implementar el procedimiento según la invención. Un producto de programa informático de este tipo puede ser, por ejemplo, una memoria de semiconductor o un disquete o un CD-ROM. En particular un producto de programa informático de este tipo puede estar previsto para su uso en la fabricación o inicialización de tarjetas con chip.

En configuraciones preferidas el producto de programa informático y/o el soporte de datos portátil está perfeccionado con características que corresponden a las características descritas anteriormente y/o mencionadas en las reivindicaciones de procedimiento dependientes.

Características, ventajas y objetivos adicionales de la invención se deducen de la siguiente descripción más exacta de un ejemplo de realización y de varias alternativas de realización. Se remite a los dibujos esquemáticos, en los que:

la figura 1 muestra un diagrama de bloques con unidades funcionales de un soporte de datos portátil según un ejemplo de realización de la invención, y

la figura 2 muestra un diagrama de flujo de un procedimiento según un ejemplo de realización de la invención en el soporte de datos representado en la figura 1.

El soporte 10 de datos mostrado en la figura 1 presenta un microcontrolador 12. De manera en sí conocida en el microcontrolador 12 en un único chip de semiconductor está integrado un núcleo 14 de procesador, un coprocesador 16, una memoria 18 y una conexión 20 de interfaz, que están conectados entre sí a través de un bus 22. Mientras que el núcleo 14 de procesador está diseñado para la realización de programa general, el coprocesador 16 sirve especialmente para la realización de cálculos de números enteros con valores binarios largos. La memoria 18 está dividida en campos de memoria configurados con diferentes tecnologías, por ejemplo, ROM, EEPROM y RAM. La memoria 18 contiene entre otras cosas instrucciones de programa, que implementan el procedimiento descrito a continuación.

El microcontrolador 12 se conoce como tal; en algunas configuraciones por ejemplo puede utilizarse el microcontrolador conocido bajo la marca Infineon SLE66CX322P, que presenta un coprocesador 16 denominado *Advanced Crypto Engine* (ACE, motor de criptografía avanzada).

La figura 2 ilustra el procedimiento del presente ejemplo de realización. Partiendo de un valor de entrada *u* y un módulo *v* el procedimiento determina el inverso modular *x* de *u* con respecto al módulo *v* con - v < *x* < v. El valor de entrada *u* y/o el módulo *v* pueden tener, por ejemplo, una longitud de bits de 160 bits ó 192 bits o 256 bits. Tales longitudes de bits son habituales para los procedimientos EC utilizados actualmente, por ejemplo el procedimiento de firma ECDSA.

El procedimiento según la figura 2 comienza en la etapa 30 porque a partir del valor de entrada *u* y el módulo *v* se generan un primer valor ampliado *U* o un segundo valor ampliado V. Los valores ampliados U y V se obtienen a partir de los valores *u* y *v* mediante la multiplicación por un factor de ampliación *f,* que es más de dos veces el módulo *v*. Además en el cálculo del primer valor ampliado U se introduce una pequeña perturbación, en el presente ejemplo de realización, una perturbación aditiva con el valor de perturbación 1.

En general, mediante la etapa 30 se desplaza información que corresponde al valor de entrada *u* y el módulo *v*, en ubicaciones de bit de valores más altos de los valores ampliados U y V. Las ubicaciones de bit de valores más bajos del segundo valor ampliado *V* quedan libres en una magnitud mayor que la longitud de bits del módulo *v*, mientras que la ubicación de bit del valor más bajo del primer valor ampliado U se ocupa con el valor de perturbación 1. Las ubicaciones de bit de valor más bajo de ambos valores ampliados U y V sirven en última instancia para el cálculo del inverso modular. Mediante el tamaño mínimo indicado anteriormente del factor de ampliación *f* se garantiza que en los valores ampliados *U* y *V* aquellas secciones de bits en las que se encuentra la información de los valores *u* y *v*, y aquellas secciones de bits, que se utilizan para el cálculo del inverso y en las que inicialmente se encuentra la perturbación, estén distanciadas de manera suficiente.

En la etapa 32 se realiza un bucle de programa con las etapas 34 de procesamiento, hasta que el segundo valor ampliado *V* es al menos igual de grande que *f* + *v*. Las etapas 34 de procesamiento corresponden exactamente al procedimiento euclídeo no ampliado, sustituyéndose en cada pasada de bucle el segundo valor ampliado V por U mod V y el primer valor ampliado U por V. Sin embargo, la condición de realización $V \geq f$ + v del bucle de programa se diferencia de la del procedimiento euclídeo en la medida en que en el presente caso el bucle se finaliza antes que en el procedimiento euclídeo.

Tras la finalización del bucle de programa se comprueba en la etapa 36, si el primer valor ampliado *V* es mayor que *f* - *v*. En este caso, en la etapa 38 se emite la diferencia *V* - *f* como resultado, concretamente como inverso modular de *u* con respecto al módulo v. En caso contrario el valor de entrada *u* no era coprimo con el módulo v*,* es decir, no podía invertirse. Entonces, en la etapa 40 se emite un mensaje de error.

Resumiendo el procedimiento recién descrito y mostrado en la figura 2 para la inversión modular mediante el uso del coprocesador 16 puede definirse en pseudocódigo de la siguiente manera:

Entradas: números enteros $u \geq 0$ y v > 1 así como un factor de ampliación aleatorio *f* con *f* > 2v

Salidas: inverso modular $x = u^{-1}$ (mod v) con -v < *x* < v*,* o error, cuando *u* y *v* no son coprimos

Procedimiento:

SEA *U*:= *f u* + 1; *V* = *f v* (1)

SIEMPRE QUE $V \geq f + v$ REALIZAR (2)

SEA *T* := *V; V* := *UmodV; U* := *T* (3)

SI *V* > *f-v* (4)

ENTONCES EMITIR EL RESULTADO *V-f* (5)

SI NO, EMITIR MENSAJE DE ERROR (6)

La línea (1) del pseudocódigo anterior corresponde a la etapa 30 en la figura 2. Las líneas (2) y (3) definen el bucle según la etapa 32; a este respecto en la línea (3) se introduce una variable auxiliar T, para implementar la asignación según las etapas 34 de procesamiento. Las líneas (4) a (6) corresponden a la diferenciación de casos de la etapa 36.

A continuación se designan los valores de las variables de programa U y V antes del inicio del bucle con U_0 y V_0 o tras la pasada de bucle de orden (*i*+1) con U_{i+1} y V_{i+1}. Se aplican por tanto las siguientes designaciones:

U_0=*fu*+1 y V_0=*f*v

$U_{i+1}=V_i$ y $V_{i+1} = U_i \bmod V_i$ para $i \geq 0$

Como ya se ha mencionado, la serie de U_i y V_i corresponde exactamente a la serie de los resultados intermedios que se obtienen cuando se aplica el procedimiento euclídeo no ampliado a los valores $f\,u + 1$ y f v. El número de pasadas de bucle, es decir, el número de las reducciones modulares necesarias, en el procedimiento descrito en este caso y en el procedimiento euclídeo aplicado a los valores u y v es aproximadamente igual.

Para explicar el funcionamiento del procedimiento de manera intuitiva, en primer lugar se remite al procedimiento euclídeo ampliado, que con respecto al valor de entrada u y el módulo v además del divisor común más grande gcd(u, v) también determina los coeficientes λ, μ de una combinación lineal con $\lambda u - \mu v = \gcd(u, v)$ y $|\lambda| < v$. Cuando se aplica este procedimiento euclídeo ampliado a $U' = f\,u$ y $V = f$ v en lugar de u y v, se obtienen exactamente los mismos resultados intermedios en las reducciones modulares y finalmente una combinación lineal $\lambda U' - \mu V = \gcd(U', V) = f \cdot \gcd(u, v)$ con los mismos valores de λ y μ que anteriormente.

Ahora se supone el procedimiento euclídeo no ampliado con el valor de entrada U = U' + 1 con la perturbación y el módulo V. Si se considera que mediante la perturbación reducida cambian los resultados intermedios de las reducciones modulares sólo en un factor multiplicativo f, en el procedimiento euclídeo no ampliado tras una fase de iteración i aparece el resto $Vi = \lambda U - \mu V = f \cdot \gcd(u, v) + \lambda$. Como gcd($u$, v) = 1, entonces puede determinarse el inverso modular λ de manera sencilla a partir de este resto, que aproximadamente presenta la misma magnitud que f.

Si bien la suposición recién realizada no es válida en general por completo, sin embargo, sí en una medida suficiente. De manera formal la exactitud del procedimientos se obtiene del hecho de que para gcd(u, v) = 1 existe un i, para el que se aplica que $V_{i-1} > 2f$ - v y $f + v > V_i > f$ - v y $(V_i - f)\cdot u = 1 \pmod v$. Si por el contrario gcd(u, v) ≠ 1, entonces se aplica que para cada V_i o bien $V_i > 2f$ - v o bien $V_i \leq$ v.

El factor de ampliación f puede seleccionarse en principio de manera aleatoria, siempre que sólo presente la magnitud mínima $f > 2$v mencionada anteriormente. En algunas configuraciones el factor de ampliación f puede ser una segunda potencia, de modo que pueda implementarse la multiplicación de los valores u y v por el factor de ampliación f mediante una sencilla operación de desplazamiento. Sin embargo, en el ejemplo de realización descrito en el presente documento el factor de ampliación f se determina como $f = 3 \cdot 2^k$ para un número entero k con $2^k > v$. Entonces se aplican las inecuaciones $2f - v > 2^{k+2} > f + v$ y $f - v > 2^{k+1}$. Por tanto es suficiente en las etapas 32 y 36, comprobar la longitud de bits de V. Mediante esta medida se reduce el esfuerzo de acondicionamiento en el coprocesador 16.

La reducción modular U mod V realizada como parte de las etapas 34 de procesamiento puede implementarse mediante cualquier procedimiento en sí conocido. En el ejemplo de realización descrito en el presente documento se realiza una división de números enteros, cuyo resto representa el resultado deseado. Como procedimiento de división se utiliza un procedimiento de resta y suma binario sencillo tal como se indica por ejemplo como el algoritmo D en las páginas 272 - 275 del libro ya citado de D. E. Knuth. Este procedimiento es adecuado en particular cuando el coprocesador 16 puede realizar rápidamente sumas, restas y operaciones de desplazamiento en números enteros largos.

En una alternativa de realización puede modificarse el procedimiento descrito hasta ahora porque se utiliza la configuración propuesta por Lehmer del procedimiento euclídeo. Esta configuración, que es especialmente adecuada para números grandes, se indica como algoritmo L en las páginas 346 - 348 del libro ya citado de D. E. Knuth. En este caso se obtienen los mismos cocientes que en el procedimiento euclídeo “clásico”, descrito anteriormente, de modo que en general no cambia el flujo de cálculo.

En una implementación a modo de ejemplo se utilizó como coprocesador 16 el *Advanced Crypto Engine* (ACE) del microcontrolador Infineon-SLE66CX322P. Este coprocesador 16 realiza las operaciones básicas habituales, por ejemplo la suma, resta y desplazamiento por bits, siempre con el ancho de registro completo de 560 ó 1120 bits. La implementación está prevista para procedimientos EC, en los que casi no se producen cálculos con longitudes de bits de más de 512 bits. Aun cuando el procedimiento descrito en el presente documento lleva aproximadamente a una duplicación de las longitudes de bits de los valores que van a procesarse, esto prácticamente no tiene importancia debido al ancho de registro aún mayor del coprocesador 16.

El procedimiento descrito en el presente documento requiere aproximadamente el mismo número de operaciones que un procedimiento euclídeo no ampliado. De este modo se evita el esfuerzo adicional requerido por lo demás para al realización de un procedimiento euclídeo ampliado. El ahorro real depende de la configuración exacta del coprocesador 16 y del ahorro en el control del coprocesador 16 (“*instrucciones de pegado*” (*glue instructions*) necesarias). En la implementación mencionada a modo de ejemplo en el presente documento se obtienen las siguientes duraciones de ejecución:

Longitud de bits del módulo →	160 bits	192 bits	256 bits	320 bits
Procedimiento euclídeo ampliado	4,80 ms	5,73 ms	7,46 ms	9,16 ms
Procedimiento del presente ejemplo de realización	2,09 ms	2,43 ms	3,16 ms	4,45 ms

De este modo, en general mediante el procedimiento descrito en el presente documento la velocidad de la inversión modular con longitudes de bits típicas para los procedimientos EC pudo más que duplicarse.

Se entiende que los detalles de la descripción anterior sólo son ejemplos de posibles configuraciones de la presente invención tal como se define mediante las reivindicaciones adjuntas. Son posibles y evidentes para el experto modificaciones adicionales, en particular en cuanto al cálculo de los valores ampliados, las etapas de procesamiento realizadas y la condición de realización.

REIVINDICACIONES

1. Procedimiento para el uso de un coprocesador (16) para la determinación del inverso modular *x* de un valor de entrada *u* con respecto a un módulo *v*, en el que:

- a partir del valor de entrada *u* se determina un primer valor ampliado *U* con una longitud de bits aumentada con respecto al valor de entrada *u*, de tal manera que en una sección de bits del primer valor ampliado *U* se encuentra la información del valor de entrada *u*, en el que la determinación del primer valor ampliado *U* comprende la multiplicación del valor de entrada *u* por un factor de ampliación *f* y en el que $f > 2v$,

- a partir del módulo *v* se determina un segundo valor ampliado *V* con una longitud de bits aumentada con respecto al módulo *v*, de tal manera que en una sección de bits del segundo valor ampliado *V* se encuentra la información del módulo *v*, en el que la determinación del segundo valor ampliado *V* comprende la multiplicación del módulo *v* por el factor de ampliación *f*,

- el coprocesador (16) está previsto para cálculos de números enteros con al menos la longitud de bits aumentada,

- al menos uno de los valores ampliados *U, V* contiene una perturbación en una posición de bit, que está distanciada de aquellas posiciones de bit, en las que se encuentra la información del valor de entrada *u* o del módulo *v*,

- partiendo de los dos valores ampliados *U* y *V* mediante el uso del coprocesador (16) se realizan etapas (34) de procesamiento de un procedimiento euclídeo, siempre que se cumpla una condición de realización predeterminada, y

- el inverso modular *x* se determina en función del resultado de las etapas (34) de procesamiento realizadas.

2. Procedimiento según la reivindicación 1, caracterizado porque el factor de ampliación *f* es una segunda potencia, de modo que en los valores ampliados *U*, *V* la información del valor de entrada *u* y del módulo *v* se ha desplazado a posiciones de bit más elevadas.

3. Procedimiento según la reivindicación 1 ó 2, caracterizado porque las etapas (34) de procesamiento contienen una reducción modular realizada mediante el uso del coprocesador (16).

4. Procedimiento según una de las reivindicaciones 1 a 3, caracterizado porque las etapas (34) de procesamiento, partiendo de los dos valores ampliados *U* y *V,* corresponden a una serie de asignaciones $(U, V) := (V, U \bmod V)$.

5. Procedimiento según una de las reivindicaciones 1 a 4, caracterizado porque el procedimiento está previsto para una aplicación criptográfica en un soporte (10) de datos portátil, en particular para un procedimiento EC.

6. Producto de programa informático, que presenta instrucciones de programa, para hacer que un microcontrolador (12) realice un procedimiento con las características de una de las reivindicaciones 1 a 5.

7. Soporte (10) de datos portátil, en particular una tarjeta con chip o módulo de chip, que está configurado para la realización de un procedimiento con las características de una de las reivindicaciones 1 a 5.

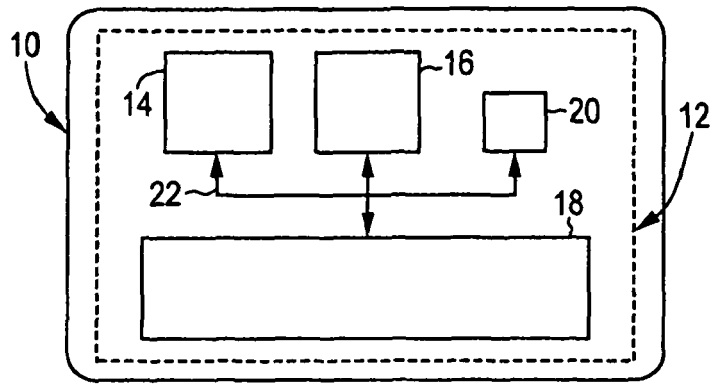
10
14
16
20
12
22
18

Fig. 1

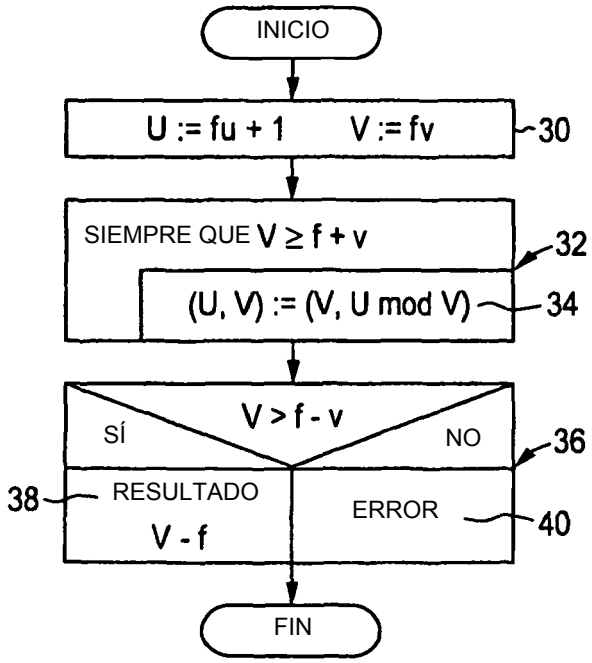
INICIO
U := fu + 1 V := fv
30
SIEMPRE QUE V ≥ f + v
32
(U, V) := (V, U mod V)
34
V > f - v
SÍ
NO
36
38
RESULTADO
V - f
ERROR
40
FIN

Fig. 2