(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 747 758**

(51) Int. CI.:

G06F 7/523 (2006.01)
G06F 7/72 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **29.03.2017 E 17163465 (2)**

(97) Fecha y número de publicación de la concesión europea: **17.07.2019 EP 3226120**

(54) Título: **Multiplicador no modular, procedimiento para multiplicación no modular y dispositivo computacional**

(30) Prioridad:

30.03.2016 IL 24484216

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
11.03.2020

(73) Titular/es:

WINBOND ELECTRONICS CORP. (100.0%)
No. 8 Keya 1st Rd., Daya District, Central Taiwan Science Park
Taichung City, Taiwan, TW

(72) Inventor/es:

KALUZHNY, URI

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Multiplicador no modular, procedimiento para multiplicación no modular y dispositivo computacional

Campo de la invención

La presente invención se refiere en general a la seguridad de datos, y particularmente a la multiplicación no modular eficiente que está protegida contra ataques de canal lateral.

Antecedentes de la invención

Muchos criptosistemas importantes, tales como el RSA, usan aritmética modular, así como no modular, incluyendo la exponenciación y la multiplicación, con valores de módulo importantes. El procedimiento clásico de calcular un producto no modular implica dividir los operandos en bloques o "dígitos" y aplicar una suma ponderada sobre los productos cruzados de los dígitos. Sin embargo, este enfoque de multiplicación simplista es computacionalmente costoso en muchos casos prácticos.

Para la multiplicación modular, por ejemplo, en cálculos criptográficos, es una práctica común usar un procedimiento eficiente, conocido como multiplicación modular de Montgomery (o simplemente multiplicación de Montgomery). Para realizar la multiplicación de Montgomery, los operandos se convierten a una forma especial de Montgomery usando un algoritmo conocido como reducción de Montgomery. La multiplicación de los operandos en forma de Montgomery evita la necesidad de reducción modular como se requiere en la aritmética convencional (aunque todavía se requiere una reducción condicional más simple si el producto resultante es mayor que el módulo). Los algoritmos de reducción y multiplicación de Montgomery se describen, por ejemplo, en Menezes et al., en el Handbook of Applied Cryptography (1996), sección 14.3.2, páginas 600-603.

Los criptosistemas pueden estar sujetos a varios tipos de ataques destinados a exponer información interna secreta. En un ataque denominado ataque de canal lateral (SCA), la información secreta puede deducirse analizando el comportamiento del consumo de energía durante la ejecución de una función criptográfica subyacente. Por ejemplo, Amiel et al., describen en un artículo titulado "Power Analysis for Secret Recovering and Reverse Engineering of Public Key Algorithms", actas de la 14ª conferencia internacional sobre áreas seleccionadas en criptografía, SAC 2007, LNCS, volumen 4876, páginas 110-125, Springer, Heidelberg, ataques de análisis de potencia diferencial (DPA), aplicados a cálculos de multiplicación no modulares.

El documento US 2004/059767 A1 divulga un procedimiento y un circuito para enmascarar datos digitales manejados por un algoritmo y factorizados por un sistema de numeración de residuos basado en una base finita de números o polinomios primos entre sí, que comprende hacer variable la base de factorización.

Sumario de la invención

Una realización que se describe en la presente memoria proporciona un multiplicador que incluye una interfaz y un circuito. La interfaz está configurada para recibir números enteros de n bits A y B. La circuitería está configurada para calcular un producto no modular ($A * B$) realizando una secuencia de cálculos y aleatorizando un patrón de una energía eléctrica consumida por el multiplicador al realizar la secuencia, incluyendo la secuencia: generar un número aleatorio w, determinar los módulos M1 y M2 que dependen de un número $R = 2^k$, con k igual a una longitud de bits de M1 y M2, y del número aleatorio w, y calcular un primer producto modular $C = A * B \% M1$ y un segundo producto modular $D = A * B \% M2$, y, producir y dar salida al producto no modular ($A * B$) en base al primer y al segundo producto modular.

En algunas realizaciones, la circuitería incluye un multiplicador Montgomery, que está configurado para calcular los productos modulares primero y segundo. En otras realizaciones, la circuitería se configura para convertir A a A' en el módulo M1 del dominio de Montgomery calculando $w*A$, para convertir B a B' en el módulo M2 del dominio de Montgomery calculando $(wd)*B$, siendo d un número entero menor que w, y para calcular el primer y el segundo producto modular calculando los respectivos productos de Montgomery $A'⊙B$ y $A⊙B'$ usando el multiplicador de Montgomery. En otras formas de realización, $(A*B)$ se representa como una combinación de una parte menos significativa AB_L y una parte más significativa AB_H que satisfacen que $(A * B) = AB_H * R + AB_L$, y la circuitería está configurada para calcular AB_H y AB_L en base al primer y al segundo producto modular.

En una realización, la circuitería está configurada para calcular la parte más significativa como $AB_H = (C-D)/d$, siendo d un número entero utilizado en la conversión de B al dominio de Montgomery, y para calcular la parte menos significativa como $AB_L=C-w*AB_H$. En otra realización, la circuitería está configurado para determinar el módulo M1 como M1=R-w, y para determinar el módulo M2 como M2=R-(w-d), siendo d un número entero menor que w.

Se proporciona adicionalmente, de acuerdo con una realización que se describe en la presente memoria, un procedimiento de multiplicación no modular, que incluye recibir números enteros de n bits A y B. Se calcula un producto no modular ($A * B$), usando un multiplicador, realizando una secuencia de cálculos y aleatorizando un patrón de una potencia eléctrica consumida por el multiplicador al realizar la secuencia, incluyendo la secuencia:

generar un número aleatorio w, determinar los módulos M1 y M2 que dependen de un número $R=2^k$, con k igual a una longitud de bits de M1 y M2, y del número aleatorio w, y calcular un primer producto modular C = A ∗ B % M1 y un segundo producto modular D = A ∗ B % M2, y, producir y dar salida al producto modular (A ∗ B) en base al primer y al segundo producto modular.

Se proporciona adicionalmente, de acuerdo con una realización que se describe en la presente memoria, un dispositivo computacional, que incluye una interfaz y un multiplicador. La interfaz está configurada para recibir números enteros de n bits A y B. El multiplicador está configurado para calcular un producto no modular (A∗B) realizando una secuencia de cálculos y aleatorizando un patrón de una energía eléctrica consumida por el multiplicador cuando realiza la secuencia, incluyendo la secuencia: generar un número aleatorio w, determinar los módulos M1 y M2 que dependen de un número $R=2^k$, con k igual a una longitud de bits de M1 y M2, y del número aleatorio w, y calcular un primer producto modular C=A∗B % M1 y un segundo producto modular D=A∗B % M2, y, producir y dar salida al producto no modular (A∗B) en base al primer y al segundo producto modular.

La presente invención se entenderá más plenamente a partir de la siguiente descripción detallada de las realizaciones de la misma, tomada junto con los dibujos en los que:

Breve descripción de los dibujos

La figura 1 es un diagrama de bloques que ilustra esquemáticamente un dispositivo computacional que tiene un multiplicador no modular eficiente que está protegido contra ataques de canal lateral, de acuerdo con una realización que se describe en la presente memoria; y

la figura 2 es un diagrama de flujo que ilustra esquemáticamente un procedimiento de multiplicación no modular en base a las multiplicaciones modulares de Montgomery, de acuerdo con una realización que se describe en la presente memoria.

Descripción detallada de las realizaciones

Visión general

Las realizaciones que se describen en la presente memoria proporcionan procedimientos de multiplicación no modular eficiente en base a las multiplicaciones modulares de Montgomery. Los procedimientos incluyen cálculos modulares con módulos aleatorios, que protegen el cálculo contra ataques de canal lateral.

La multiplicación modular de Montgomery, también conocida simplemente como la multiplicación de Montgomery por brevedad, es un procedimiento de multiplicación eficiente. Para resumir brevemente, dados dos números enteros grandes A y B, en lugar de calcular A ∗ B, la multiplicación de Montgomery (denotada Θ) produce A Θ B = A ∗ B ∗ R^{-1} % M, en donde $R=2^k$, y k es la longitud de bit del módulo M. (El símbolo "%" se usa en la presente descripción y en las figuras para denotar "módulo" y el símbolo "∗" o "·" se usa para denotar multiplicación no modular).

En las realizaciones divulgadas, la multiplicación de Montgomery se usa para calcular eficientemente expresiones de la forma A ∗ B % M. Para este fin, solamente uno de los operandos se convierte al dominio de Montgomery y se multiplica con el otro operando de la siguiente manera:

Ecuación 1:

$$A' \Theta B = A * R * B * R^{-1} \% M = A * B \% M$$

En algunas realizaciones, un multiplicador no modular calcula dos productos modulares en forma de la Ecuación 1, para dos módulos diferentes M1 y M2, y combina los dos productos modulares para obtener el producto no modular - (A ∗ B). El producto no modular se representa usando partes más y menos significativas como $AB_H * R + AB_L$, en donde las partes más y menos significativas se evalúan en base a los productos modulares A ∗ B % M1 y A ∗ B % M2. Las partes más y menos significativas se pueden ver como subconjuntos respectivos de menor y mayor significación del producto de 2n bits (A∗B).

Los módulos M1 y M2 se determinan en base a $R=2^k$, y a un pequeño entero impar w elegido al azar. Como se describirá en detalle a continuación, el cálculo de los productos modulares se simplifica considerablemente seleccionando k (y, por lo tanto, R) lo suficientemente grande. El componente aleatorio w en los módulos M1 y M2 hace que el cálculo esté protegido contra ataques de canal lateral. Más específicamente, bajo ciertas condiciones de w, los esquemas propuestos dan como resultado el resultado de multiplicación no modular correcto independientemente del w específico seleccionado, pero dado que para diferentes valores de w el cálculo comprende diferentes cálculos, el patrón de consumo de energía subyacente será diferente también, lo que protege el cálculo frente a los ataques de análisis de potencia.

En el contexto de la presente divulgación y en las reivindicaciones, se puede derivar un patrón de consumo de energía analizando un parámetro adecuado tal como la potencia o la corriente consumida durante el cálculo.

Descripción del sistema

La figura 1 es un diagrama de bloques que ilustra esquemáticamente elementos de circuito de un dispositivo 20 computacional que incluye un multiplicador no modular eficiente que está protegido frente a ataques de canal lateral, de acuerdo con una realización que se describe en la presente memoria. Los elementos de circuito que se muestran en la figura generalmente se implementan como circuitos lógicos de hardware en un dispositivo de circuito integrado (CI), tal como en un circuito integrado específico de aplicación (ASIC) o una matriz de puertas programable en campo (FPGA), pero pueden implementarse alternativamente en software en un procesador programable adecuado, o como una combinación de elementos de hardware y software.

Los circuitos representados llevan a cabo una función de multiplicación no modular que puede integrarse en el dispositivo computacional en una amplia variedad de configuraciones y aplicaciones diferentes, para realizar operaciones relacionadas con el cifrado, el descifrado y/o la autentificación, por ejemplo. Solamente los elementos del dispositivo 20 que son directamente relevantes para la multiplicación no modular se muestran en la figura, y la integración de estos elementos con otros componentes del dispositivo 20 será evidente para los expertos en la materia.

El dispositivo 20 extrae energía eléctrica de una fuente de energía, por ejemplo, de una fuente de alimentación o batería (no mostrada). Las técnicas descritas a continuación protegen el dispositivo 20 de ataques de canal lateral que intentan acceder a información secreta mediante el análisis del patrón temporal o espectral de la energía consumida por el dispositivo 20. Tal ataque puede detectar el consumo de energía de varias maneras, por ejemplo, al detectar la corriente en las líneas de suministro de energía que alimentan el dispositivo 20, o midiendo la radiación electromagnética en las proximidades del dispositivo.

El dispositivo 20 computacional tiene un par de entradas 28A, 28B (implementadas como posiciones en una matriz de memoria, por ejemplo) para recibir las entradas A y B de n bits. El dispositivo 20 comprende un multiplicador 24 no modular, que acepta A y B a través de una interfaz 32, y envía el producto no modular ($A*B$) a una salida 36 (tal como otra posición en la matriz de memoria), cuyos contenidos pueden enviarse a otros componentes del dispositivo 20 o retroalimentarse a una o ambas entradas 28A, 28B para cálculos posteriores, tales como multiplicaciones sucesivas múltiples que se usan en la exponenciación.

El multiplicador 24 no modular comprende un motor 40 de Montgomery, que está configurado con un parámetro 46 de módulo M y un número 48 entero k, en el que k es igual a la longitud de bits de M, que se utilizará en la aritmética de Montgomery como se describirá a continuación. El número entero k determina el valor de R como $R=2^k$.

El motor 40 de Montgomery recibe las entradas A y B de la interfaz 32, y calcula $A * B \% M$. (El símbolo "%" se usa en la presente descripción y en las figuras para denotar "módulo" y el símbolo "$*$" o "$\cdot$" se usa para denotar la multiplicación no modular.) Para llevar a cabo la multiplicación no modular entre A y B, el dispositivo 20 aplica el motor 40 de Montgomery para A y B con dos módulos diferentes indicados M1 y M2, y entrega los productos modulares A * B % M1 y A * B % M2 a un combinador 50. Como se describirá en detalle a continuación, el combinador 50 aplica cálculos adicionales a los productos modulares para obtener el producto no modular (A * B).

El motor 40 de Montgomery comprende un multiplicador 52 Montgomery, que acepta los operandos 54A y 54B de k bits. Suponiendo que las entradas al multiplicador 54 Montgomery se denotan como X e Y, el multiplicador Montgomery realiza el cálculo:

Ecuación 2:

$$X \odot Y = X * Y * R^{-1} \% M$$

El motor 40 de Montgomery comprende además un convertidor 56, que recibe el módulo 46 M. El convertidor 56 recibe una entrada X de n bits (es decir, A o B) y la convierte a X' en el dominio Montgomery dada por:

Ecuación 3:

$$X' = X * R \% M$$

X' tiene la misma longitud de bits que M, es decir, k bits. Como se describirá a continuación, al seleccionar M=(R-w), el cálculo de la Ecuación 3 es equivalente a calcular $X'=w*X$, que no requiere cálculos modulares y, por lo tanto, es considerablemente más simple.

El multiplicador 24 de Montgomery aplica el motor 40 por separado a los módulos M1 y M2. Las operaciones descritas a continuación se pueden llevar a cabo en varios esquemas de programación. En una realización de

ejemplo, cuando se configura en el módulo M1, el motor 40 de Montgomery entrega la entrada B al operando 54B, aplica el convertidor 56 a la entrada A y entrega A' = A ∗ R % M1 al operando 54A. La salida del multiplicador de Montgomery en este caso viene dada por:

Ecuación 4:

$$A' \odot B = A * R * B * R^{-1} \% M1 = A * B \% M1$$

Cuando se configura para el módulo M2, el motor 40 de Montgomery entrega la entrada A al operando 54B, aplica el convertidor 56 a la entrada B y entrega B' = B ∗ R % M2 al operando 54A. La salida del multiplicador de Montgomery en este caso viene dada por:

Ecuación 5:

$$B' \odot A = B * R * A * R^{-1} \% M2 = A * B \% M2$$

En las ecuaciones 4 y 5, el valor (A * B) de 2n bits se reduce a productos modulares de k bits con módulo M1 y M2, respectivamente. Como se describirá en detalle a continuación, el combinador 50 recupera el valor (A ∗ B) no modular de 2n bits de los dos productos modulares de las Ecuaciones 4 y 5.

Como se describió anteriormente, el parámetro 46 de módulo M está configurado para uno de los módulos M1 y M2. El multiplicador 24 no modular comprende un aleatorizador 60 de módulo que calcula un módulo 64 M1 y un módulo 68 M2 de la siguiente manera:

Ecuación 6:

$$M1 = R - w$$

$$M2 = R - (w - d)$$

en donde d es un número entero adecuado, por ejemplo, d=2 o d es alguna potencia de 2, y un número 72 w es un entero aleatorio o pseudoaleatorio elegido por un generador 76 aleatorio. (El término "aleatorio", como se usa en la presente descripción y en las reivindicaciones, debe entenderse como que incluye también números "pseudoaleatorios", a menos que el contexto indique lo contrario). En la presente divulgación, w se elige como un entero impar aleatorio mayor que d.

El número w se selecciona como impar porque la multiplicación de Montgomery requiere módulos impares. Además, w se elige lo suficientemente pequeño, de modo que el convertidor 56 pueda calcular w∗A y (w-d)∗B de forma rápida y eficiente. El intervalo de w es suficientemente grande para lograr una aleatorización efectiva en M1 y M2. En una realización de ejemplo, w comprende un número aleatorio de 32 bits.

Obsérvese que el resultado al calcular el producto no modular (A ∗ B) usando el motor 40 de Montgomery y el combinador 50 es independiente del valor del w 72 aleatorio (en ciertas condiciones de w), aunque los cálculos subyacentes reales dependen de w. En otras palabras, la aplicación del esquema a las mismas entradas A y B con pares de módulos M1 y M2 derivados de diferentes números w1 y w2 da como resultado el mismo resultado no modular (A ∗ B). Sin embargo, las operaciones aritméticas realizadas en el cálculo (A ∗ B) con w1 son diferentes de aquellas realizadas con w2, lo que da como resultado diferentes patrones de consumo de energía respectivos.

El multiplicador 24 no modular comprende un módulo 80 aritmético que comprende un sumador 82 y un multiplicador 84 no modular. El sumador y el multiplicador operan típicamente en bloques de un tamaño predefinido, tal como treinta y dos bits. El módulo 80 aritmético aplica cálculos aritméticos no modulares para el multiplicador 24 como lo requieren sus diversos elementos, tales como el multiplicador Montgomery 52, el convertidor 56, el combinador 50 y el aleatorizador 60 de módulo.

Un controlador 88 programa y coordina las operaciones de los diferentes elementos del multiplicador 24 para llevar a cabo la multiplicación no modular.

Multiplicación no modular eficiente en base a las multiplicaciones de Montgomery

La figura 2 es un diagrama de flujo que ilustra esquemáticamente un procedimiento de multiplicación no modular en base a las multiplicaciones modulares de Montgomery, de acuerdo con una realización que se describe en la presente memoria. En la descripción que sigue, el procedimiento se describe como ejecutado por el multiplicador 24 no modular de la figura 1.

El procedimiento comienza con un multiplicador 24 no modular que recibe entradas A y B de números enteros de n bits, en un paso 100 de entrada. En un paso 104 de aleatorización, el multiplicador no modular genera un número (72) entero impar aleatorio w, por ejemplo, utilizando el generador 76 aleatorio. Además, en el paso 104, el multiplicador no modular establece el número 48 entero k, que determina $R=2^k$. En una realización, k se

establece como n más la longitud de bits de w. Por ejemplo, cuando w es un entero de 32 bits, k=n+32. Al seleccionar k como n más la longitud de bits de w, se cumplen las siguientes condiciones.

Ecuación 7:

$$w * A < R$$

$$(w - d) * B < R$$

Téngase en cuenta que en la Ecuación 7, suponiendo que ambas entradas A y B sean números enteros de n bits, si la condición superior es verdadera, también lo es la condición inferior, y por lo tanto, solamente se debe considerar la condición superior al seleccionar R lo suficientemente grande.

En un paso 108 de selección de módulo, el multiplicador no modular determina los módulos M1 (64) y M2 (68) en base a $R=2^k$ y al número entero aleatorio w, de acuerdo con la Ecuación 6 anterior, es decir, M1=R-w, M2=R- (w-d). En un paso 112 de conversión, el motor 40 de Montgomery convierte A y B en módulos de dominio de Montgomery M1 y M2, respectivamente, usando el convertidor 56. Al calcular A', particionamos R=(R-w)+w, y usando la condición $w*A<R$ de la Ecuación 7, obtenemos con alta probabilidad también que $w*A<(R-w)$ o:

Ecuación 8:

$$A' = A * R \% M1 = w * A \% (R - w) = w * A$$

En la Ecuación 8, la operación del módulo por (R-w) se requiere solamente cuando $w*A>(R-w)$, y puede omitirse en caso contrario. Dado que tenemos que $w*A<R$, la probabilidad de que $w*A$ exceda (R-w) es w/R, que para un w de 32 bits y un k de 512 bits es menor que $2^{32}/2^{512}$, que es prácticamente igual a cero.

De manera similar, al calcular B' particionamos R=[R-(w-d)]+(w-d), y usando la condición $(w-d)*B<R$ de la Ecuación 7, obtenemos también con alta probabilidad que $(w-d)*B<[R-(w-d)]$ o:

Ecuación 9:

$$A' = B * R \% M2 = (w - d) * B \% [R - (w - d)] = (w - d) * B$$

En algunas realizaciones, las multiplicaciones $w*A$ y $(w-d)*B$ en las Ecuaciones 8 y 9, respectivamente, se llevan a cabo mediante el módulo 80 aritmético.

En algunas realizaciones, el motor 40 de Montgomery aplica el convertidor 56 por separado a la entrada A para producir A' y a la entrada B para producir B'. La salida del convertidor 56, es decir, A' o B' se entrega al operando 54A del multiplicador 52 de Montgomery.

En un paso 114 de cálculo modular, el multiplicador 52 de Montgomery calcula productos modulares como se indica en las Ecuaciones 4 y 5 anteriores:

Ecuación 10:

$$C = A' \odot B = A * B \% M1$$

Ecuación 11:

$$D = B' \odot A = A * B \% M2$$

En el apéndice a continuación, presentamos un procedimiento en el que la reducción de un número entero Z de 2m bits usando diferentes módulos M1 y M2, se utiliza para representar Z por partes menos y más significativas de m bits Z1 y Z0 que pueden resolverse dados Z%M1 y Z%M2.

Al aplicar los resultados del apéndice a $Z = (A * B)$ podemos dividir el producto no modular en partes más y menos significativas como se describirá en la presente memoria.

En un paso 116 de partición más significativo/menos significativo, el producto $(A * B)$ de 2n bits se divide en partes menos y más significativas designadas AB_H y AB_L, respectivamente, en el que $(A * B) = AB_H * R + AB_L$. En base al apéndice, tenemos:

Ecuación 12:

$$AB_H = (C - D) / d$$

Ecuación 13:

$$AB_L = C - w * AB_H$$

En las ecuaciones 12 y 13, $R=2^k$, AB_L es un número entero de k bits y AB_H es un número entero de (2n-k) bits. En algunas realizaciones, d es una potencia de 2, y la división por d en la Ecuación 12 se implementa eficientemente como operaciones de desplazamiento binario. En un paso 120 de salida, el multiplicador 24 no modular emite el producto no modular $(A * B)$ como:

Ecuación 14:

$$(A * B) = AB_H * R + AB_L$$

y luego el procedimiento termina.

El multiplicador 24 típicamente elige un valor diferente de w cada vez que se invoca el procedimiento de la figura 2. Como tal, el patrón temporal o espectral del consumo de energía del multiplicador 24 y del dispositivo 20 en su conjunto varía aleatoriamente. El patrón de consumo de energía es aleatorio incluso si los valores de entrada A y B siguen siendo los mismos. Por lo tanto, es poco probable que un ataque de canal lateral deduzca información secreta, por ejemplo, los valores de A y B, del análisis del perfil de consumo de energía.

Las realizaciones descritas anteriormente se proporcionan a modo de ejemplo, y también se pueden usar otras realizaciones adecuadas. Por ejemplo, en la descripción anterior, el motor 40 de Montgomery comprende un único multiplicador de Montgomery y un único convertidor al dominio de Montgomery que se aplican ambos dos veces para calcular los dos productos modulares $A * B \% M1$ y $A * B \% M2$. En realizaciones alternativas, en las que el motor de Montgomery comprende dos multiplicadores 52 de Montgomery y dos convertidores 56, los dos productos modulares se calculan en paralelo, lo que reduce el tiempo de cálculo a la mitad.

Apéndice

Sea Z un número entero de 2m bits. Z puede representarse como $Z=Z1*R+Z0$, en donde una parte menos significativa Z0 y una parte más significativa Z1 comprenden enteros de m bits, y $R=2^m$. La reducción del número entero Z de 2m bits mediante los módulos M1=R-w y M2=R-(w-d), en donde w>d es un número entero impar que es muy pequeño en comparación con R da como resultado:

Ecuación 15:

$$Z\%M1 = (w * Z1 + Z0)\% M1$$

Ecuación 16:

$$Z\%M2 = [(w - d) * Z1 + Z0]\%M2$$

Se puede demostrar que si la longitud m de bits se selecciona lo suficientemente grande para que Z1<<M1, tenemos, con alta probabilidad,

$$Z\%M1 = w * Z1 + Z0$$

$$Z\%M2 = (w - d) * Z1 + Z0$$

Y luego las siguientes fórmulas producen las partes más y menos significativas de Z dados Z%M1 y Z%M2:

Ecuación 17:

$$Z1 = (Z\%M1 - Z\%M2)/d$$

Ecuación 18:

$$Z0 = Z\%M1 - w * [(Z\%M1 - Z\%M2)/d]$$

Como se indicó anteriormente, en una realización alternativa de la presente invención, los pasos y operaciones descritos anteriormente son llevados a cabo por un procesador programable adecuado bajo el control de las instrucciones del programa de software. El software se puede descargar al procesador de forma electrónica, por ejemplo, a través de una red. Además, o alternativamente, el software puede almacenarse en medios tangibles, no transitorios, legibles por computadora, tales como medios ópticos, magnéticos o de memoria electrónica.

Aunque las realizaciones descritas en la presente memoria abordan principalmente una operación de multiplicación no modular eficiente que está protegida frente a los ataques de canal lateral, los procedimientos y sistemas descritos en la presente memoria también pueden usarse en otras aplicaciones, tales como en la implementación de la multiplicación no modular en software, en cuyo caso la multiplicación simplista puede ser más rápida pero no está protegida contra ataques DPA.

REIVINDICACIONES

1. Un multiplicador (24) no modular, que comprende:

una interfaz (32), que está configurada para recibir números A y B enteros de n bits; y

circuitería (40), que está configurada para calcular un producto no modular (A ∗ B) mediante la realización de una secuencia de cálculos, y para aleatorizar un patrón de energía eléctrica consumida por el multiplicador (24) no modular al realizar la secuencia, comprendiendo la secuencia:

generar un número aleatorio w;

determinar los módulos M1 y M2 que dependen de un número $R=2^k$, en donde k es igual a una longitud de bits de M1 y M2, y del número aleatorio w, y calcular un primer producto modular C = A ∗ B % M1 y un segundo producto modular D = A ∗ B % M2; y

producir y dar salida al producto no modular (A ∗ B) en base al primer y al segundo producto modular.

2. El multiplicador (24) no modular de acuerdo con la reivindicación 1, en el que la circuitería (40) comprende un multiplicador (52) de Montgomery, que está configurado para calcular el primer y el segundo producto modular.

3. El multiplicador (24) no modular de acuerdo con la reivindicación 2, en el que la circuitería (40) está configurada para convertir A a A' en el módulo M1 del dominio de Montgomery calculando w∗A, para convertir B a B' en el módulo M2 del dominio de Montgomery calculando (w-d)∗B, en donde d es un número entero menor que w, y para calcular el primer y el segundo producto modular calculando los respectivos productos de Montgomery A'ΘB y AΘB' usando el multiplicador (52) de Montgomery.

4. El multiplicador (24) no modular de acuerdo con la reivindicación 1, en el que (A ∗ B) se representa como una combinación de una parte menos significativa AB_L y una parte más significativa AB_H que satisface (A ∗ B) = AB_H ∗ R + AB_L, y en donde la circuitería (40) está configurada para calcular AB_H y AB_L en base al primer y al segundo producto modular.

5. El multiplicador (24) no modular de acuerdo con la reivindicación 4, en el que el circuito está configurado para calcular la parte más significativa como $AB_H=(C-D)/d$, en donde d es un número entero utilizado para convertir B al dominio de Montgomery, y para calcular la parte menos significativa como $AB_L=Cw*AB_H$.

6. El multiplicador (24) no modular de acuerdo con la reivindicación 1, en el que el circuito está configurado para determinar el módulo M1 como M1=R-w, y para determinar el módulo M2 como M2=R-(w-d), en donde d es un número entero menor que w.

7. El multiplicador (24) no modular de acuerdo con la reivindicación 1, en el que el multiplicador está configurado para aplicar las primera y segunda operaciones de multiplicación no modular entre A y B, con diferentes valores de w respectivos, en las que los patrones de consumo de energía primero y segundo, correspondientes la energía consumida durante la ejecución de las primera y segunda operaciones de multiplicación no modular comprende diferentes patrones de consumo de energía respectivos.

8. Un procedimiento de multiplicación no modular, que comprende:

recibir (100) números A y B enteros de n bits; y

usar un multiplicador, calcular un producto no modular (A ∗ B) realizando una secuencia de cálculos y aleatorizar un patrón de energía eléctrica consumida por el multiplicador al realizar la secuencia, comprendiendo la secuencia:

generar un número aleatorio w;

determinar los módulos M1 y M2 que dependen de un número $R=2^k$, en donde k es igual a una longitud de bits de M1 y M2, y del número aleatorio w, y calcular un primer producto modular C = A ∗ B % M1 y un segundo producto modular D = A ∗ B % M2; y

producir y dar salida (120) al producto no modular (A ∗ B) en base al primer y al segundo producto modular.

9. El procedimiento de multiplicación no modular de acuerdo con la reivindicación 8, en el que calcular el primer y el segundo producto modular comprende calcular el primer y segundo producto modular usando un multiplicador (52) de Montgomery.

10. El procedimiento de multiplicación no modular de acuerdo con la reivindicación 9, en el que el cálculo del primer y del segundo producto modular comprende convertir (112) A a A' en el módulo M1 del dominio de

Montgomery calculando w∗A, convertir (112) B a B' en el módulo M2 del dominio de Montgomery calculando (w-d)∗B, en donde d es un número entero menor que w, y calcular (114) el primer y el segundo producto modular calculando los respectivos productos de Montgomery A'ʘB y AʘB' utilizando el multiplicador (52) de Montgomery.

11. El procedimiento de multiplicación no modular de acuerdo con la reivindicación 8, en el que (A ∗ B) se representa como una combinación de una parte menos significativa AB_L y una parte más significativa AB_H que satisfacen que (A ∗ B) = AB_H ∗ R + AB_L, y que comprende calcular AB_H y AB_L en base al primer y al segundo producto modular.

12. El procedimiento de multiplicación no modular de acuerdo con la reivindicación 11, en el que calcular la parte más significativa comprende calcular (116) AB_H=(C-D)/d, en donde d es un número entero usado para convertir B al dominio de Montgomery, y en el que calcular la parte menos significativa comprende calcular (116) AB_L=C-w∗AB_H.

13. El procedimiento de multiplicación no modular de acuerdo con la reivindicación 8, en el que determinar los módulos M1 y M2 comprende determinar (108) el módulo M1 como M1=R-w, y determinar (108) el módulo M2 como M2=R-(w-d), en donde d es un número entero menor que w.

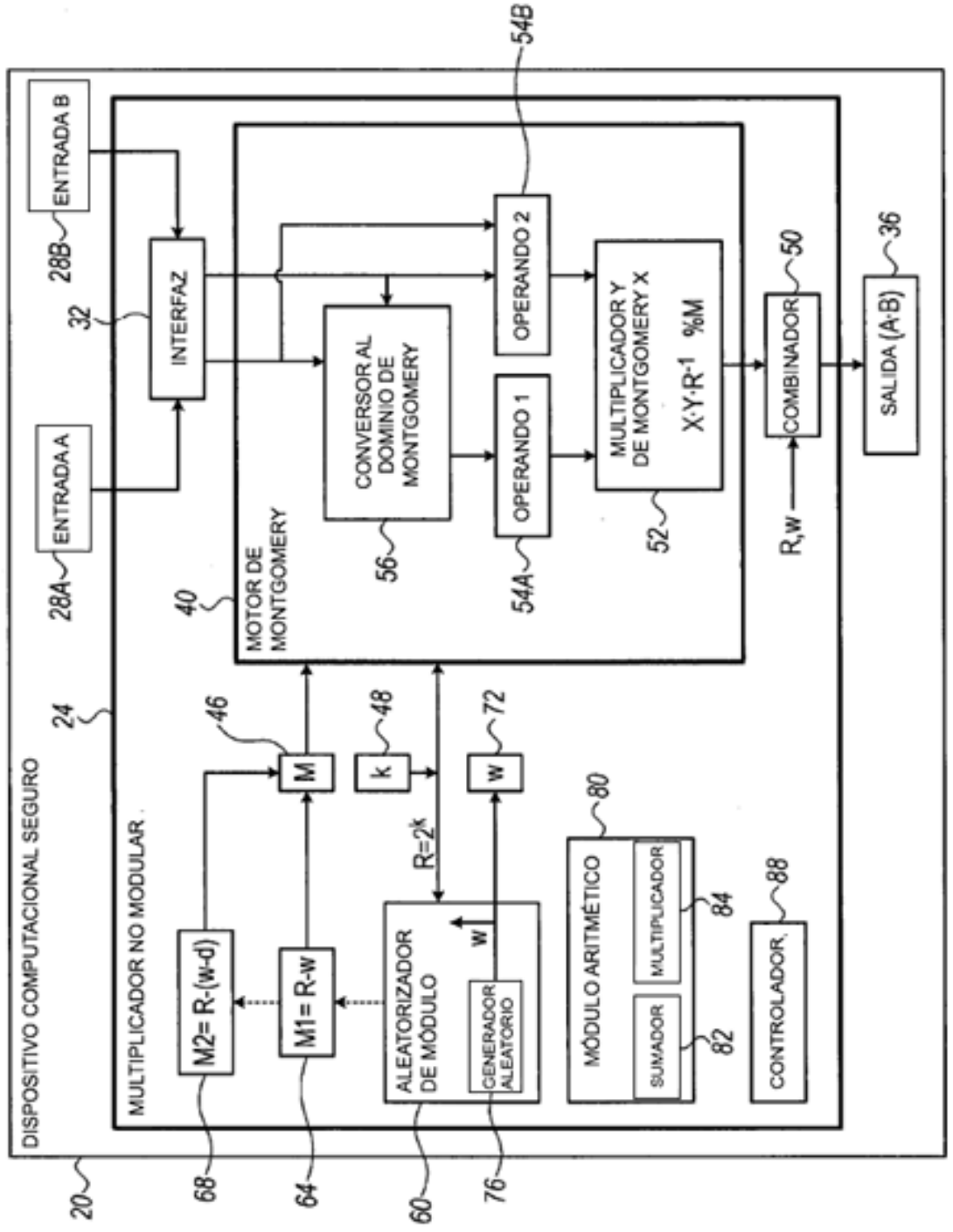
DISPOSITIVO COMPUTACIONAL SEGURO
MULTIPLICADOR NO MODULAR
ENTRADA B
ENTRADA A
INTERFAZ
MOTOR DE MONTGOMERY
CONVERSOR AL DOMINIO DE MONTGOMERY
OPERANDO 2
OPERANDO 1
MULTIPLICADOR Y DE MONTGOMERY X
X·Y·R-1 %M
COMBINADOR
SALIDA (A·B)
M
k
w
R=2k
M2= R-(w-d)
M1= R-w
ALEATORIZADOR DE MÓDULO
GENERADOR ALEATORIO
MÓDULO ARITMÉTICO
MULTIPLICADOR
SUMADOR
CONTROLADOR
R,w
20
24
28A
28B
32
36
40
46
48
50
52
54A
54B
56
60
64
68
72
76
80
82
84
88

FIG. 1

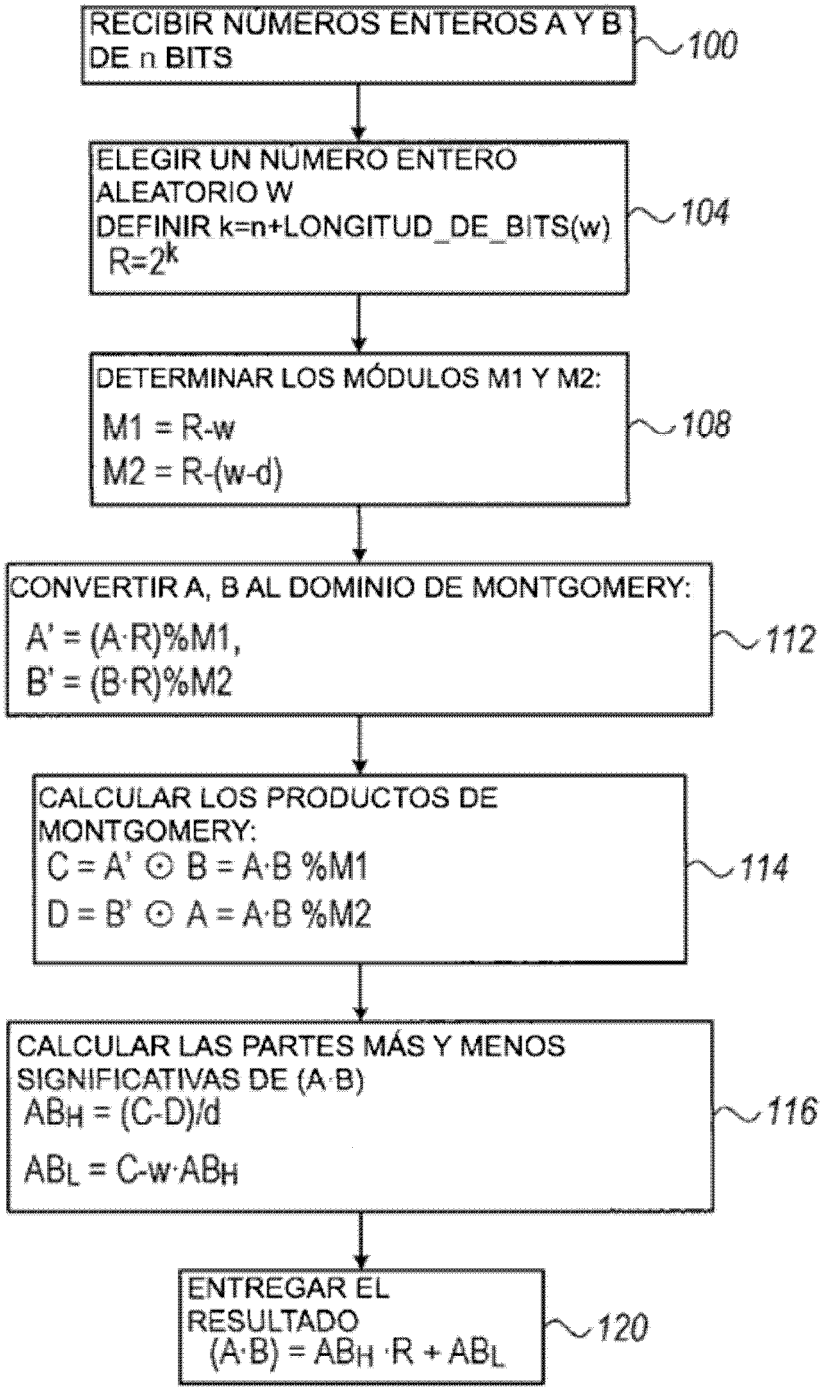
RECIBIR NÚMEROS ENTEROS A Y B DE n BITS
100
ELEGIR UN NÚMERO ENTERO ALEATORIO W
DEFINIR k=n+LONGITUD_DE_BITS(w)
R=2^k
104
DETERMINAR LOS MÓDULOS M1 Y M2:
M1 = R-w
M2 = R-(w-d)
108
CONVERTIR A, B AL DOMINIO DE MONTGOMERY:
A' = (A·R)%M1,
B' = (B·R)%M2
112
CALCULAR LOS PRODUCTOS DE MONTGOMERY:
C = A' ⊙ B = A·B %M1
D = B' ⊙ A = A·B %M2
114
CALCULAR LAS PARTES MÁS Y MENOS SIGNIFICATIVAS DE (A·B)
AB_H = (C-D)/d
AB_L = C-w·AB_H
116
ENTREGAR EL RESULTADO
(A·B) = AB_H ·R + AB_L
120

FIG. 2