(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 435 721**

(51) Int. CI.:

H04L 9/00 (2006.01)
H04L 9/06 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **18.01.2010 E 10701001 (9)**

(97) Fecha y número de publicación de la concesión europea: **24.07.2013 EP 2380306**

(54) Título: **Circuito criptográfico protegido contra los ataques de observación, en particular de orden elevado**

(30) Prioridad:

20.01.2009 FR 0950341

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
23.12.2013

(73) Titular/es:

INSTITUT TELECOM - TELECOM PARISTECH (100.0%)
46 Rue Barrault
75013 Paris, FR

(72) Inventor/es:

DANGER, JEAN-LUC y
GUILLEY, SYLVAIN

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Circuito criptográfico protegido contra los ataques de observación, en particular de orden elevado

La presente invención se refiere a un circuito criptográfico protegido contra los ataques de observación. Esta se aplica en particular para la protección de circuitos criptográficos contra los ataques de observación de orden elevado sobre implantaciones basadas en el enmascaramiento.

La criptografía tiene, en particular, como objetivos proteger:

- o bien la confidencialidad de una información, por medio del cifrado y de la operación dual: el descifrado;
- o bien solo la autenticidad de una información, mediante las operaciones de firma y de verificación, mediante las operaciones de firma y de verificación de las firmas.

La criptografía utiliza métodos matemáticos, los cuales se ha demostrado que conducen a sistemas seguros. Por ejemplo, un cifrado se considera seguro cuando se ha probado que no existen, en el estado actual de los conocimientos publicados, métodos de ataque significativamente más rápidos que el ataque exhaustivo, que corresponde a la prueba de todas las claves posibles.

En general, los métodos de cifrado implican unos cálculos complejos, necesarios para la seguridad de los sistemas. Esta complejidad no plantea problemas particulares a los ordenadores, pero es un grave inconveniente en el caso de dispositivos que no tienen una gran potencia de cálculo, por lo general animados por microprocesadores de « 8 bits », de bajo coste. Las consecuencias pueden ser de varios tipos, como:

- un obstáculo para un buen uso, un ejemplo típico sería una tarjeta bancaria que necesita varios minutos para firmar una factura;
- una denegación de servicio, un ejemplo típico sería un decodificador de televisión de pago que no puede seguir el flujo de información emitido.

Para resolver esta dificultad sin aumentar el precio de los sistemas, es habitual añadir un sistema de ayuda a la unidad central del dispositivo utilizado, en forma de un coprocesador dedicado a la criptografía.

Sin embargo, lo aplique la unidad central o un coprocesador especializado, el algoritmo criptográfico se aplica mediante un dispositivo físico, en la actualidad de tipo electrónico. Estos dispositivos físicos presentan imperfecciones inevitables, ligadas a las propiedades inherentes a las leyes básicas de la electricidad.

Es por ello por lo que sistemas de criptografía seguros desde el punto de vista matemático pueden ser atacados aprovechando las imperfecciones inherentes de los sistemas físicos que aplican el algoritmo, así pues:

- la duración de los cálculos puede depender de los valores de los datos, en particular en los sistemas informáticos optimizados en velocidad, lo que ha dado lugar a los ataques de tipo « Timing Attack » que se describen en particular en el documento de P. Kocher y otros: Timing Attack on Implementation of Diffie-Hellman, RSA, DSS and other systems, en Proceedings of CRYPTO'96, volumen 1.109 de LNCS, páginas 104-113, Springer-Verlag, 1996, permitiendo estos ataques en algunos casos encontrar todas las claves secretas a partir de simples mediciones de los tiempos de ejecución;
- el consumo instantáneo también puede depender de los datos, lo que ha dado lugar a una serie de ataques como:
  - el ataque SPA (Simple Power Analysis) que se describe en particular en el documento de Thomas S. Messerges y otros: Investigations of Power Analysis Attacks on Smartcards, en USENIX - Smartcard'99, páginas 151-162, mayo 10-11, 1999, Chicago USA, intentando este ataque distinguir las operaciones ejecutadas por una unidad central a partir de una medición de su consumo eléctrico durante una operación criptográfica;
  - el ataque DPA (Differential Power Analysis) que se describe en particular en el documento de S. Guilley y otros: Differential Power Analysis Model and some Results, en Proceedings of WCC/CARDIS, páginas 127-142, agosto de 2004, Toulouse, Francia, utilizando este ataque operaciones estadísticas sobre numerosas mediciones de consumo eléctrico, realizadas durante operaciones criptográficas sobre mensajes aleatorios y con una clave constante, para validar o invalidar una hipótesis hecha sobre una parte limitada de la clave;
- toda corriente eléctrica que circula por un conductor genera un campo electromagnético cuya medición puede dar lugar a ataques de tipo EMA (Electro Magnetic Attack) idénticos en su principio a los ataques que se basan en el consumo eléctrico;
- algunos ataques alteran de forma intencionada el funcionamiento de los sistemas con el fin de aprovechar los resultados erróneos para encontrar los secretos del sistema, se conoce a estos ataques bajo la denominación de ataques por inyección de errores.

Se llama canal oculto a cualquier imperfección de un dispositivo físico que aplica un algoritmo criptográfico y es susceptible de sufrir fugas de información vinculada a los secretos conservados en la memoria del dispositivo.

Se han propuesto algunas protecciones contra estos ataques en los canales ocultos, basadas en particular en:

- el ocultamiento, donde se trata de que la fuga sea constante, en este caso independiente del secreto;
- el enmascaramiento, donde se trata de que la fuga sea aleatoria, imprevisible y por lo tanto inservible.

Estas dos técnicas hacen que sean más difíciles los ataques que buscan encontrar información, pero siguen siendo no obstante vulnerables a los ataques que se beneficiarían de los fallos de implementación. Existen numerosos ejemplos de vulnerabilidades potenciales o demostradas, por ejemplo:

- el ocultamiento basado en lógica diferencial (como WDDL) puede ser vulnerable a un ataque sobre las diferencias de plazos combinatorios acumulativos entre una y otra de las fases de cálculo, fase de evaluación y fase de precarga;
- el enmascaramiento, que puede ser sensible a los ataques de orden elevado, denominados HO-DPA.

El documento WO 01/61916 A describe un método de contramedida con un enmascaramiento de los datos.

El documento XP 2 455 442: "Third order differential Analysis and split mask countermeasure for low Energy Embedded processor" describe una contramedida sobre los ataques de tercer orden por medio de varias máscaras intermedias generadas de forma iterativa.

Un objetivo de la invención es, en particular, permitir una protección eficaz contra los ataques de observación de orden elevado. Para ello, la invención tiene por objeto un circuito criptográfico que comprende al menos un registro que suministra una variable x enmascarada ($x \oplus m$) por una variable de máscara (m), caracterizado porque dicha variable enmascarada se cifra mediante una primera caja de sustitución S de forma cíclica, comprendiendo dicho circuito un registro de máscara que emite en cada ciclo una máscara transformada (m_t) a su vez transformada por una función de modificación (41, 61) en la variable de máscara m, siendo dicha función de modificación una biyección o una función de descomposición, transformándose dicha variable enmascarada ($x \oplus m$) al final de un ciclo en una nueva máscara ($S(x) \oplus m'$) de dicha variable (x), cifrada por la primera caja de sustitución S, mediante una nueva máscara (m'), calculándose dicha nueva máscara (m') en función de dicha variable de máscara (m) y de dicha variable enmascarada (x ⊕ m) por medio de una segunda caja de sustitución S'.

La nueva máscara transformada m'_t se introduce, por ejemplo, en el registro de máscara M tras haberse sometido a la modificación inversa en la salida de la segunda caja de sustitución S'.

En un primer modo posible de aplicación, la modificación puede ser una biyección B de tal modo que la máscara transformada m_t se almacena en el registro M, valiendo $B(m_t)$ la máscara m aplicada a la caja de sustitución y siendo $B^{-1}(m')$ el nuevo valor que hay que almacenar en el registro, siendo m' la salida de la segunda caja de sustitución S' (21).

La biyección B es, por ejemplo, tal que el peso de Hamming de la máscara m no siempre es idéntica al peso de Hamming de su imagen por la biyección $B(m)$.

En un segundo modo posible de aplicación, la modificación de la máscara m se puede realizar mediante la descomposición en dos submáscaras m_1 y m_2 de tal modo que $m = m_1 \theta m_2$ donde θ es una ley de composición de grupo, almacenándose la primera submáscara m_1 en un primer registro de máscara M1 y almacenándose la segunda submáscara m_2 en un segundo registro de máscara M2, siendo $m'_1 = m' \theta^{-1} m'_2$ el nuevo valor que hay que almacenar en el registro M1 y siendo m'_2 el nuevo valor que hay que almacenar, siendo m' el valor de la máscara en la salida de la segunda caja de sustitución S'.

La ley de composición es, por ejemplo, la operación « o exclusiva » XOR, la operación de suma, la operación de multiplicación o incluso la operación de tipo s=a*b + (a+b)/2, siendo * la multiplicación y + la suma.

La ley de composición tiene, por ejemplo, una distancia de Hamming entre dos submáscaras consecutivas m_2 de un ciclo a otro, Δm_2, no nula.

Al tener las submáscaras m_1, m_2 un número par de bits, la ley de composición de grupo (α) que permite obtener la máscara m a partir de las submáscaras m'_1, m_2 verifica, por ejemplo:

- $\Delta m_2 = 1$ por grupos de 2 bits ;
- $m = m_1 \oplus m_2$ si $m_2\ MSB = 1$ o $m = m_1$ si $m_2\ MSB = 0$

siendo respectivamente $m_2\ MSB$, $m_2\ LSB$ los bits de alto peso y de bajo peso de la submáscara m_2.

El circuito comprende, por ejemplo, un generador de valores aleatorios, emitiendo dicho generador el valor de la segunda submáscara m'_2.

Esta submáscara m_1 procedente del primer registro de máscara M1 se puede guardar en el segundo registro de máscara M2 al final de ciclo, recibiendo el primer registro M1, por su parte, la nueva máscara m'_1 de tal modo que se reconstruye la nueva máscara completa $m' = m'_1 \theta m'_2$.

El algoritmo de cifrado es de tipo DES, estando el camino de los datos x, m dividido en dos partes, izquierda y derecha, dicho circuito comprende, por ejemplo, además del registro de máscara izquierda ML y del registro de máscara derecha MR, un tercer registro de máscara M, conteniendo el registro MR la submáscara derecha m_{r1} y conteniendo el registro ML la submáscara izquierda m_{l2}', conteniendo el registro M la submáscara m_{r2} igual a la submáscara m_{l1}, siendo los valores efectivos de la máscara derecha m_r y de la máscara izquierda m_l tales que:

- $m_r = m_{r1}$ θ m_{r2}
- $m_l = m_{l1}$ θ m_{l2}

La segunda caja de sustitución S' comprende, por ejemplo, dos cajas idénticas a la primera caja de sustitución S y dos puertas XOR, estando las salidas de las dos cajas conectadas a una puerta XOR cuya salida emite la nueva máscara m', entrando la variable enmascarada ($x \oplus m$) en una caja y en la otra puerta XOR cuya otra entrada recibe la máscara m, entrando la salida de esta puerta entrado en la otra caja.

Una memoria ROM ejecuta, por ejemplo, la lógica de las puertas XOR y la sustitución de las cajas.

Se mostrarán otras características y ventajas de la invención mediante la descripción que sigue, hecha en referencia a los dibujos adjuntos, que representan:

- las figuras 1a y 1b, una ilustración del principio de enmascaramiento de una caja de sustitución;
- la figura 2, una ilustración del enmascaramiento al nivel de una caja de sustitución;
- la figura 3, una presentación de un ejemplo de distribución de actividad en la salida de un registro para diferentes valores de la distancia de Hamming entre dos palabras consecutivas de una variable x del registro;
- la figura 4, una ilustración de un primer modo posible de modificación del camino de una máscara en un circuito de acuerdo con la invención;
- la figura 5, una presentación de la distribución de actividad correspondiente al modo anterior;
- la figura 6, una ilustración de un segundo modo posible de realización del camino de una máscara en un circuito de acuerdo con la invención;
- la figura 7, una presentación de la distribución de actividad correspondiente al modo anterior;
- la figura 8, un ejemplo de modificación con descomposición de las máscaras sin utilizar un generador aleatorio;
- la figura 9, un ejemplo de realización de enmascaramiento de acuerdo con la invención utilizando un algoritmo DES;
- la figura 10, un ejemplo de realización de una ley de composición de grupo utilizada en un circuito de acuerdo con la invención;
- las figuras 11a, 11b y 11c, unos ejemplos de realización de una segunda caja de sustitución utilizada en un circuito de acuerdo con la invención.

Las figuras 1a y 1b ilustran el principio del enmascaramiento de una caja de sustitución 1, llamada normalmente « S-box », con una máscara constante m. Una S-box aplica a un mensaje una función no lineal que tiene en particular como objetivo volver al mensaje entrópico en la salida de la S-box.

En la figura 1a, un dato x se combina en la entrada de la S-box 1 con la clave criptográfica k. En este ejemplo el dato y la clave se combinan mediante una operación lógica « o exclusiva » también llamada xor y que se designará de aquí en adelante ⊕. En el caso de la figura 1a, el mensaje $x \oplus k$ codificado en n bits por ejemplo, se presenta por lo tanto en la entrada de la S-box. El mensaje de salida correspondiente es $S(x \oplus k)$.

La figura 1b muestra el enmascaramiento de la S-box con una máscara constante m. La técnica del enmascaramiento se basa en el cifrado de las variables internas sensibles que dependen de un secreto mediante una máscara m. En el ejemplo de las figuras 1a y 1b, x es dicha variable. Dado que m es imprevisible para un atacante, las fugas no se pueden utilizar en el marco de un ataque directo sobre x. La variable x ya no existe como tal pero se puede reconstruir mediante el par de señales (m, x_m) donde $x_m = x$ θ m es la variable que hay que enmascarar y θ es un operador de grupo que puede ser en particular lógico o aritmético. En el ejemplo de la figura 1b, el enmascaramiento utiliza la función xor, esto es $x_m = x \oplus m$.

Este tipo de enmascaramiento está especialmente adaptado para la protección de los algoritmos DES (Data Encryption Standard) y AES (Advanced Encryption Standard), en el que las operaciones lineales se realizan en un cuerpo binario. La máscara se puede aplicar a las variables internas que son, por lo general, vectores de bits, palabras, funciones criptográficas. El enmascaramiento al nivel de una palabra lo hace aplicable tanto a las implementaciones en programas informáticos como a las materiales. La realización del enmascaramiento es simple cuando la función f en la que se aplica la variable enmascarada es lineal en función de la ley de grupo, esto es $f(x$ θ $m) = f(x)$ θ $f(m)$.

El valor de $f(x)$ se puede reconstruir a partir de $f(x$ θ $m)$ y m. De este modo $f(x)$ se extrae justo al final del algoritmo de tal modo que se evita una fuga de información procedente de la variable x. Al no estar correlacionados los cálculos sobre x θ m con x, no hay fugas directas de secreto.

Si la función f es no lineal, el mecanismo de enmascaramiento se vuelve más complejo ya que $f(x)$ no se puede

recrear matemáticamente a partir de $f(x\ \theta\ m)$ y $f\ (m)$. En los algoritmos de cifrado simétrico, la parte no lineal corresponde a las cajas de sustitución o S-box, así por ejemplo la S-box 1 de las figuras 1a y 1b. Una técnica actual en programas informáticos utiliza una memoria ROM que genera la función no lineal de la S-box, designándose esta función *Sm*, tal que:

$$S_m(x \oplus m) = S(x) \oplus m$$

siendo S la función de la S-box antes del enmascaramiento.

En consecuencia, el tamaño de la memoria pasa de 2^n para *S* a 2^{2n} para S_m, siendo n el número de bits de la máscara. Las figuras 1a y 1b ilustran este cambio.

Esta realización no está protegida con una implantación material en la que se realiza una iteración del algoritmo de cifrado en un ciclo de reloj. En este caso, las transferencias al nivel de un registro descubren de forma automática los datos. En efecto, considerando a título de ejemplo el operador ⊕, una transferencia se puede describir por la siguiente relación:

$$x \oplus m \oplus S(x) \oplus m = x \oplus S(x)$$

El término $x \oplus m$ de esta relación es el valor inicial en la entrada de la S-box S_m, y el término $S(x) \oplus m$ es el valor final en la salida de la S-box, indicando el operador ⊕ entre estos dos términos la transición. La relación anterior muestra que el resultado de la transición $x \oplus S(x)$ es independiente de la máscara *m*.

La eficacia que ofrece el enmascaramiento se puede probar contra los ataques de primer orden que únicamente consideran la variable interna *x*, como se demuestra en particular en el documento de J. Blomer y otros: Provably Secure Masking of AES, en LNCS, editor, Proceedings of SAC'04, volumen 3.357, páginas 69-83, Springer, agosto de 2004, Waterloo, Canadá. Las fugas de información que proceden del circuito de enmascaramiento pueden sin embargo verse sometidas a un ataque de segundo orden, e incluso de un orden superior mucho más elevado.

Un ataque de segundo orden considera dos variables x_1 y x_2. Por ejemplo, si x_1 y x_2 están enmascaradas por la misma máscara, el ataque de segundo orden se beneficia del hecho de que $x_1 \oplus x_2 = x_1 \oplus m \oplus x_2 \oplus m$.

Se han estudiado algunas implementaciones para superar los ataques de orden elevado, como se demuestra en particular en el documento de M. Akkar y otros: A generic protection against High-oder differential Power Analysis, en LNCS, editor, Proceedings of FSE'03, volumen 2.887 de LNCS, Springer, 2003, Berlín, utilizando este documento unas máscaras constantes. Sin embargo, para obtener una gran eficacia es necesario un fuerte incremento de la complejidad. A título de ejemplo, se ha demostrado que el algoritmo DES precia al menos tres máscaras diferentes y seis S-box adicionales para cada S-box con el fin de poder resistir los ataques de orden elevado mediante este método, como se demuestra en el documento de J. Lv. y otros, Enhanced DES implementation secure against differential power analysis in smart-cards, en Information Security and Privacy, 10th Australasian Conference, volumen 3.574 de LNCS, páginas 195-206, Brisbane, julio de 2005, Springer Verlag.

En otro método conocido, se calcula una nueva máscara *m* en cada iteración. Este método se describe en particular en el documento de F-X. Standaert y otros, FPGA Implementations of the DES and Triple-DES Masked Against Power Analysis Attack, en Proceedings of FPL 2006, agosto de 2006, Madrid. La variable enmascarada $x \oplus m$ de un registro R se asocia cada vez a una nueva máscara *m* procedente de un registro M. De este modo al final de un ciclo, la variable $x \oplus m$ se transforma en $S(x) \oplus m'$ en la que la nueva máscara *m*' se calcula en función de *m* y de $x \oplus m$ por medio de una nueva S-box de función *S*'. Este método ofrece un buen compromiso de complejidad ya que solo asocia una nueva S-box *S*' a cada S-box S existente.

Sin embargo, esta implementación sigue estando sujeta a los ataques de segundo orden como se demuestra en particular en el documento de E. Peeters y otros, Improved Higher-Order Side-Channel Attacks with FPGA Experiments, en CHES, volumen 3.659 de LNCS, páginas 309-323, Springer-Verlag, 2005.

La figura 2 ilustra la implementación de las S-box *S* y *S*'. Un registro 22, designado R, suministra la variable enmascarada $x \oplus m$ con la que se combina la clave *k*, por ejemplo mediante una puerta XOR 24, en la entrada de la S-box 1, de función *S* , la cual emite en la salida la variable $S(x \oplus k) \oplus m'$ que se vuelve a introducir en la entrada del registro 22, R. En paralelo, un registro 23, designado M, suministra la máscara *m* que entra en la S-box 21, de función S', siendo la otra entrada de esta última común a la entrada de la S-box 1. La salida de la S-box 21, *S*', emite una nueva máscara *m*' y se vuelve a introducir en la entrada del registro M. Esta nueva máscara se utiliza para enmascarar la variable *x* que emite el registro R. De aquí en adelante en la descripción, las máscaras producidas por el bucle de máscara 20 que comprende el registro de máscara 23 y la S-box 21, *S*' se utilizarán para enmascarar la variable *x*.

Las reintroducciones en los registros R y M se hacen cada uno en un ciclo de reloj. En aras de la simplicidad, la representación de la figura 2 hace abstracción de las funciones de expansión y de permutación propias del algoritmo DES.

El ataque HO-DPA que describe Peeters y otros trata sobre las variables $x \oplus m$ y m que proceden de los registros. El principio consiste en estudiar las distribuciones de la actividad en la salida de los registros para diferentes valores de la variable x. En lógica CMOS, un modelo de actividad designado A puede ser la distancia de Hamming designada DH entre dos palabras consecutivas. En particular:

$$A(x \oplus m, m) = DH(x_i \oplus m_i, x_{i-1} \oplus m_{i-1}) + DH(m_i, m_{i-1}) \quad (1)$$

esto es,

$$A(x \oplus m, m) = PH(\Delta x \oplus \Delta m) + PH(\Delta m) \quad (2)$$

PH corresponde al peso de Hamming y Δx a la distancia de Hamming entre dos palabras consecutivas x_i, x_{i-1} de la variable x.

Si x y m tienen un único bit, la actividad corresponde a 2. $PH(\Delta m)$ si $\Delta x = 0$.

Si $\Delta x = 1$, la actividad corresponde a $PH(\Delta \overline{m}) + PH(\Delta m)$ y vale de forma constante 1. Así pues, el conocimiento de las distribuciones de consumo para cada Δx permite construir el ataque HO-DPA al observar las distribuciones de consumo y al compararlas con la actividad prevista para una hipótesis de clave k incluida en la variable x.

Si x y m se codifican en 4 bits, el peso de Hamming PH puede tomar cinco valores: 0, 1, 2, 3, 4. Por lo tanto, la actividad $A(x \oplus m, m) = PH(\Delta x \oplus \Delta m) + PH(\Delta m)$ puede tomar nueve valores: 0, 1, 2, 3, 4, 5, 6, 7, 8.

La figura 3 ilustra las distribuciones de la actividad A para una máscara de 4 bits para los diferentes valores posibles de Δx, tomando Δx en este caso los 2^4 valores 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15. La figura 3 presenta, por lo tanto, cinco distribuciones diferentes 31, 32, 33, 34, 35 para el conjunto de los valores de Δx. En cada distribución 31, el valor de Δx se distribuye, o reparte, en función de la actividad la cual puede tomar los nueve valores 0, 1, 2, 3, 4, 5, 6, 7 y 8. Aparecen claras diferencias entre estas distribuciones, por lo que estas diferencias las puede utilizar un ataque HO-DPA.

Para comparar la eficacia de las diferentes soluciones de enmascaramiento, se puede utilizar una dimensión métrica de comparación de las distribuciones 31, 32, 33, 34, 35 inspirada en una prueba conocida, la prueba de $\chi 2$, definida por la siguiente relación:

$$\chi 2 = \sum_{i=1}^{C} \frac{(P_i - P_i ref)^2}{P_i ref} = \frac{1}{N} \sum_{i=1}^{C} \frac{(N_i - N_i ref)}{N_i ref} \quad (3)$$

en la que :

- C representa el número de clases de la distribución, por ejemplo C = 9 para las distribuciones de actividad con una máscara codificada en 4 bits, representando C - 1 el grado de libertad de la prueba de $\chi 2$;
- N es el número total de muestras;
- P_i es la probabilidad observada de tener un elemento en la clase i, P_i también es igual a $\frac{N_i}{N}$, siendo N_i el número de elementos observados en la clase i ;
- $P_i ref$ es la probabilidad de referencia de tener un elemento en la clase i, siendo $P_i ref$ también igual a $\frac{N_i ref}{N}$, siendo $N_i ref$ el número de elementos esperados en la clase i.

Se considera la distribución de referencia como la media de las 16 distribuciones que se obtienen para todos los valores de Δx cuando la máscara se codifica en 4 bits. El valor de $\chi 2$ en este caso permite juzgar la homogeneidad de todas las distribuciones. A título de referencia, $\chi 2$ vale 21,89 para la distribución con una máscara de 4 bits. Es preciso que este valor sea lo más bajo posible para evitar discriminaciones entre las distribuciones 31, 32, 33, 34, 35 que pueden ser utilizadas por ataques HO-DPA. En otras palabras, es preciso que este valor de $\chi 2$ sea lo más pequeño posible para protegerse de los ataques HO-DPA.

La invención permite protegerse frente a dichos ataques de observación, incluidos de orden elevado, al reducir las discriminaciones, o las diferencias, entre las diferentes distribuciones de actividad para una misma máscara. Se considerará, de aquí en adelante, a título de ejemplo, una máscara codificada en 4 bits.

En particular, haciendo referencia por ejemplo a la figura 2, la invención hace que la actividad del registro M que contiene la máscara m sea independiente de la actividad del registro R que contiene la variable enmascarada $x \oplus m$.

Si la variable *x* no influye en las distribuciones de consumo propias de estos dos registros 22, 23, ya no es posible llevar a cabo con éxito un ataque HO-DPA como se describe en el documento de Peeters y otros. El equilibrio se puede conseguir modificando la estructura del camino de la máscara sin tocar el camino de la variable enmascarada. Las modificaciones propuestas se pueden hacer o bien en el espacio, transformando la máscara, o bien en el tiempo, cambiando la evolución de la máscara. De manera ventajosa, la invención se refiere a la transformación de la máscara. De acuerdo con la invención, se pueden hacer varias modificaciones de máscara.

La figura 4 ilustra una primera modificación posible de la máscara, una modificación mediante biyección. En este caso, una transformación biyectiva 41, designada *B*, se intercala entre el registro de máscara 23, registro M, y la S-box 21, de función *S*'. La transformación bijectiva inversa 42, designada B^{-1}, se aplica en la salida de la S-box. De este modo, la máscara *m*' que emite la S-box *S*' experimenta la transformación bijectiva B^{-1} antes de almacenarse en el registro M. La biyección es tal que la distancia de Hamming entre *m* y su imagen *B*(*m*) no sea siempre nula de tal modo que se modifique la correlación de la actividad entre la variable $x \oplus m$ y *B*(*m*). La implementación de la biyección *B* se puede hacer mediante una tabla predefinida.

La máscara *m*' en la salida de S-box *S*' es la que utiliza realmente la variable enmascarada. La figura 4 ilustra el camino de la máscara.

La figura 5 muestra las 16 distribuciones 51 que se obtienen tras la transformación de la máscara mediante una biyección tal y como se ilustra en la figura 4, para los 16 valores de Δx para una máscara codificada en 4 bits como en el caso de la figura 3, por lo tanto con los mismos valores de actividad que pueden ir de 0 a 8. Estas distribuciones 51 presentan ligeras diferencias entre sí, al contrario que las distribuciones de la figura 3. Las distribuciones de la actividad son mucho más homogéneas, lo que se confirma mediante la prueba de $\chi 2$ que da un valor medio de 1,85 con esta transformación bijectiva, en comparación con valor de 21,89 que corresponde a la distribución de la figura 3.

La figura 6 ilustra otra modificación posible de acuerdo con la invención. Se trata de una modificación de la máscara mediante descomposición. En este caso, la máscara *m* se descompone en dos submáscaras m_1 y m_2 tales que:

$$m = m_1 \ \theta \ m_2$$

realizando θ una ley de composición interna de grupo, como el operador θ mencionado con anterioridad. Al ser θ un operador de grupo, admite un operador inverso θ^{-1}.

El registro M de las figuras anteriores se sustituye, por lo tanto, por dos registros 231, 232. Un registro M1 contiene el valor m_1 y un registro M2 contiene el valor m_2. La salida de estos registros está unida a un circuito 61 que ejecuta la ley de composición θ. En la salida de este operador 61, la máscara *m* se recompone mediante la operación $m = m_1 \ \theta \ m_2$ antes de transformarse en una nueva máscara *m*' mediante la S-box S', 21 como en los caminos anteriores que se ilustran en particular en las figuras 2 y 4.

La operación inversa θ^{-1} la lleva a cabo un operador 62 situado en la salida de la S-box S' 21. Este operador permite separar la máscara *m*' en m'_1 y m'_2 de tal modo que $m'_1 = m' \ \theta^{-1} \ m'_2$. La máscara *m*' se puede generar mediante un generador aleatorio RNG como se ilustra en la figura 6. En cada iteración del algoritmo de cifrado el registro M2 que contiene m_2 toma un nuevo valor m'_2 que proviene del generador aleatorio 71. El registro M1 que contiene m_1 toma un nuevo valor $m'_1 = m' \ \theta^{-1} \ m'_2$, este valor lo emite el operador 62 que realiza la operación de grupo θ^{-1} a partir del dato de entrada *m'* suministrado por la S-box 21 y el dato de entrada m'_2 suministrado por el generador aleatorio 71.

La actividad A en el registro R que contiene la variable enmascarada $x \oplus m$ y de los dos registros M1 y M2 se convierte, por lo tanto, en:

$$A(x \oplus m) = PH(\Delta x \oplus \Delta m) + PH(\Delta m_1) + PH(\Delta m_2) \qquad (4)$$

Si se considera este modelo de actividad, la prueba de $\chi 2$ se aplica para diferentes leyes θ y los resultados se ilustran, a título de ejemplo, en la siguiente tabla:

θ	**Sin modificaciones**	**B(a)**	**A XOR b**	**a+b**	**a*b**	**a*b + (a+b)/2**	**a⍺b**
$\chi 2$	21,89	1,85	16,69	0,31	0,36	0,18	0

Esta tabla muestra que se pueden utilizar los operadores simples, como la suma + y la multiplicación *, para obtener unas distribuciones equilibradas, siendo los resultados obtenidos respectivamente 0,31 y 0,36, lo que rechaza el ataque en un número extremadamente alto de huellas de consumo que hay que adquirir para discriminar las distribuciones entre sí. La transformación mediante biyección B permite obtener un resultado de 1,85.

De manera ventajosa, la ley α, que se describe a continuación, permite tener un $\chi 2$ nulo, es decir unas distribuciones perfectamente idénticas en el sentido del modelo de actividad considerado y, por lo tanto, resistente a los ataques de orden elevado.

La figura 7 ilustra las 16 distribuciones posibles 70 con 4 bits por variable y, teniendo en cuenta la ley α, las distribuciones que se obtienen con el operador lógico XOR no permiten reducir de manera suficiente el valor de $\chi 2$ ya que estas hacen que aparezcan dos grandes clases de distribución en función de la paridad del peso de Hamming de Δx, *PH* ($\Delta x \oplus m$). Si Δx es par, la actividad tiene siempre unos valores pares mientras que si Δx es impar la actividad tiene siempre unos valores pares.

La figura 8 ilustra un ejemplo de aplicación que da un resultado similar a la solución anterior, pero sin utilizar un generador aleatorio, excepto para iniciar el cálculo. Para evitar el sobrecoste ligado a la utilización de un generador aleatorio en cada ciclo, una de las dos submáscaras puede sustituir la variable aleatoria generada en el ejemplo anterior. Se puede utilizar, por ejemplo, la submáscara procedente del registro M1, 231. Esta submáscara procedente del registro M1 se guarda en el registro M2, 232, al final del ciclo, recibiendo por su parte el registro M1 la nueva máscara m'_1 de tal modo que se reconstruye la máscara completa. De este modo, la nueva máscara que se genera en cada ciclo es $m' = m'_1 \theta m_1$, en el ciclo siguiente la nueva máscara generada es $m'' = m''_1 \theta m'_1$ y así sucesivamente.

La figura 9 ilustra un ejemplo de enmascaramiento para el algoritmo DES. En esta figura las funciones de expansión y de permutación se han ignorado en aras de la claridad.

En lo que se refiere al algoritmo DES, el camino de los datos se divide en dos partes, izquierda y derecha, como en cualquier esquema de Feistel. En particular, el registro de variables enmascaradas 22 de los ejemplos anteriores está dividido en dos registros, un registro izquierdo 221 y un registro derecho 222. En lo que se refiere a la parte de enmascaramiento, además de los dos registros de máscara 91, 92, un registro derecho MR y un registro izquierdo ML, se añade un tercer registro 93, designado M, de tal modo que se descomponen las dos máscaras izquierda y derecha. La implementación del algoritmo DES saca provecho de la solución expuesta con anterioridad, en relación a las figuras 7 y 8, sin utilizar un generador aleatorio en cada ciclo, sino utilizando tres registros encadenados. El registro MR contiene la submáscara m_{r1} y el registro ML contiene la submáscara m_{l2}. El registro M contiene la submáscara m_{r2} igual a la submáscara m_{l1}. Los valores efectivos de las máscaras derecha m_r e izquierda m_l respectivamente contenidos en los registros MR y ML vienen dados por las siguientes relaciones:

- $m_r = m_{r1} \theta m_{r2}$
- $m_l = m_{l1} \mu\theta m_{l2}$

Los operadores 611, 612 que realizan la relación de grupo θ están situados en la salida de los registros MR, M, ML para realizar las dos operaciones anteriores. La máscara derecha m_r se transforma mediante la S-box *S'*, 21 en una nueva máscara derecha m'_r, siendo la otra entrada de la S-box 21 la variable enmascarada $x_r \oplus m_r$ y a continuación cifrada que viene del registro R derecho 222. La nueva máscara derecha m'_r se combina con la máscara m_l para dar $m'_r \oplus m_l$ en la entrada del operador 62 que realiza la operación θ^{-1}. Este último combina $m'_r \oplus m_l$ con m'_{r2}. De forma similar al ejemplo de la figura 8, la submáscara m'_{r2} en la entrada del operador 62 no la suministra un generador aleatorio, sino que es igual a la submáscara m_{r1} emitida por el registro MR. Cada registro 221, 222, 91, 92, 93 recibe un valor inicial 94, 95, 96, 97. La siguiente tabla presenta las máscaras utilizadas tanto en los registros de máscaras 91, 92, 93, en las tres últimas columnas, como en los registros de variables enmascaradas 221, 222, en las dos columnas anteriores, para tres ciclos consecutivos:

Ciclo	Máscara derecha	Máscara izquierda	MR	M	ML
0	*ML*0	*MR*0	*M*0	*MR*0 θ^{-1} *M*0	*ML*0 θ^{-1} *M*0

1	*MR*0	m'_r	*MR*0 θ^{-1} *M*0	m'_{r1}	*M*0
2	m'_r	m''_r	m'_{r1}	m''_{r1}	*MR*0 θ^{-1} *M*0

Esta tabla muestra que las máscaras utilizadas en los registros de máscaras así como las utilizadas en los registros de variables enmascaradas son diferentes y permiten, en consecuencia, evitar en particular un ataque de tipo HO-DPA.

Una implementación de la función α permite tener unas distribuciones equilibradas, por lo tanto $\chi 2$ nulo, como se ilustra en la figura 10.

Esta ley se basa en el hecho de que Δm_2 nunca es nulo. En efecto, si la variación de *m*, $\Delta m = \Delta m_2 = 0$, la ley de grupo θ hace que Δm_1 también es igual a 0. En este caso, la actividad que viene dada por la anterior ecuación (4) solo puede ser nula si Δx es nulo. Como no existen otros valores de Δx para obtener una actividad nula, esto

muestra que la actividad no estará nunca perfectamente equilibrada para todos los Δx. Por el contrario, si el valor de Δm_2 es no nulo, pueden existir leyes que permitan verificar el equilibrio perfecto de las distribuciones de actividad y de este modo obtener $\chi 2 = 0$. Con respecto a la prueba de imposibilidad de enmascaramiento, tal y como se considera en particular en el documento de G. Piret y otros: Security Analysis of Higher-Order Boolean Masking Schemes for Block Ciphers, IET Information Security, 2(1): 1-11, 2008, la invención prevé una relación de enlace entre las dos máscaras m_1 y m_2, lo que no se tiene en cuenta en el documento de G. Piret y otros.

Si se consideran a título de ejemplo las submáscaras por paquetes sucesivos de 2 bits, teniendo entonces la máscara un número par de bits, lo que es a menudo el caso, la ley siguiente denominada α permite obtener una distribución equilibrada sea cuales sean los Δx. El grupo de 2 bits formado por el bit de alto peso de la submáscara m_2, designado m_2MSB, y del bit de bajo peso, designado m_2LSB, condiciona de la siguiente forma la generación de la máscara mediante la ley α:

- $\Delta m_2 = 1$ por grupos de 2 bits;
- $m = m_1 \oplus m_2$ si $m_2\,MSB = 1$ o $m = m_1$ si $m_2\,MSB = 0$

Esta ley α es involutiva y la aplica el operador 61 en la entrada de la S-box 21.

Por otra parte, la figura 10 presenta un ejemplo posible de implementación 100 del operador 62, situado en la salida de la S-box 21, que aplica la función $m'_1 = m' \alpha^{-1} m'_2$ para un grupo de 2 bits de máscara.

El circuito 100 utiliza un generador aleatorio 101 para producir la submáscara m_2. La variable aleatoria emitida por este generador 101 selecciona, en un codificador 102 que recibe en la entrada $m'_2\,MSB$ y $m'_2\,LSB$, bien un aumento o bien una disminución en el registro M2 que se codifica en Gray (o código binario reflejado) de tal modo que se garantiza la condición $\Delta m'_2 = 1$. Partiendo de la máscara m' producida por la S-box 21, el operador 62 emite una submáscara m'_1 cuyo bit de alto peso $m'_1\,MSB$ es igual a $m'MSB \oplus m'_2\,MSB$, siendo $m'MSB$ el bit de alto peso de m' y $m'_2\,MSB$ que toma el valor 0 o 1 según la variable aleatoria generada en la entrada del codificador 102. El bit de bajo peso $m'_1\,LSB$ es igual a $m'MSB$ o a $m'LSB \oplus m'_2\,LSB$ si el valor de $m'_2\,MSB$ es respectivamente igual a 0 o a 1.

Las figuras 11a, 11b y 11c presentan unas mejoras posibles para realizar la caja de sustitución S', 21 que realiza la nueva máscara m'. La caja S' se puede realizar con una mezcla de puertas XOR y de memorias ROM, de acuerdo con varios niveles de robustez y de complejidad.

La figura 11a presenta una primera solución que utiliza dos cajas S, 1 y dos puertas XOR 110, 111 para reconstruir la información no enmascarada x y la nueva máscara m'. De este modo, por la primera puerta 111, se obtiene $x = m \oplus x \oplus m$ y la segunda puerta 112 emite la nueva máscara m' en la salida de la caja S' con $m' = S(x \oplus m) \oplus S(x)$.

Una segunda solución, que se ilustra en la figura 11b utiliza una memoria ROM 113 para realizar la lógica de las puertas XOR y la sustitución de las cajas S del montaje de la figura 11a. La memoria recibe entonces dos palabras, m y $x \oplus m$.

La figura 11c presenta una tercera solución que se puede aplicar para una descomposición espacial sin registro aleatorio 71. Esta solución utiliza una memoria ROM 114 que recibe tres palabras $x \oplus m$, m_1 y m_2. Esta memoria integra la lógica y la función 115 llevadas a cabo por la memoria 113 de la solución de la figura 11b y la ley de grupo θ para la reconstrucción de la máscara m en la entrada de la función 115 así como, por ejemplo, la ley inversa de grupo θ^{-1} en la salida de esta función 115.

REIVINDICACIONES

1. Circuito criptográfico que comprende al menos un registro (R, 22) que suministra una variable *x* enmascarada ($x \oplus m$) por una variable de máscara (*m*), **caracterizado porque** dicha variable enmascarada se cifra mediante una primera caja de sustitución S(1) de forma cíclica, comprendiendo dicho circuito un registro de máscara (M, 23) que emite en cada ciclo una máscara transformada (m_t) a su vez transformada por una función de modificación (41, 61) en la variable de máscara *m*, siendo dicha función de modificación una biyección o una función de descomposición, transformándose dicha variable enmascarada ($x \oplus m$) al final de un ciclo en una nueva máscara ($S(x) \oplus m'$) de dicha variable (*x*), cifrada por la primera caja de sustitución *S*, mediante una nueva máscara (*m*'), calculándose dicha nueva máscara (m') en función de dicha variable de máscara (m) y de dicha variable enmascarada ($x \oplus m$) por medio de una segunda caja de sustitución *S*' (21).

2. Circuito de acuerdo con la reivindicación 1, **caracterizado porque** la nueva máscara transformada m' se introduce en el registro de máscara M tras haberse sometido a la función inversa de modificación (42, 62) de dicha función de modificación (41, 61) en la salida de la segunda caja de sustitución *S'* (21).

3. Circuito de acuerdo con una cualquiera de las reivindicaciones anteriores, **caracterizado porque** la función de modificación es una biyección (41) B tal que la máscara transformada m_t se almacena en el registro M, valiendo $B(m_t)$ la máscara *m* aplicada a la caja de sustitución y siendo $B^{-1}(m')$ el nuevo valor que hay que almacenar en el registro, siendo *m'* la salida de la segunda caja de sustitución *S'* (21).

4. Circuito de acuerdo con una cualquiera de las reivindicaciones anteriores, **caracterizado porque** la biyección B es tal que el peso de Hamming de la máscara *m* no siempre es idéntico al peso de Hamming de su imagen por la biyección B(*m*).

5. Circuito de acuerdo con una cualquiera de las reivindicaciones 1 a 2, **caracterizado porque** la modificación de la máscara m se realiza mediante la descomposición en dos submáscaras m_1 y m_2 de tal modo que $m = m_1 \theta m_2$,en la que θ es una ley de composición de grupo, almacenándose la primera submáscara m_1 en un primer registro de máscara M1 (231) y almacenándose la segunda submáscara m_2 en un segundo registro de máscara M2 (232), siendo $m'_1 = m \theta^{-1} m'_2$ el nuevo valor que hay que almacenar en el registro M1 y siendo m'_2 el nuevo valor que hay almacenar, siendo *m'* el valor de la máscara en la salida de la segunda caja de sustitución *S'* (21).

6. Circuito de acuerdo con una cualquiera de las reivindicaciones 1 a 2 y 5, **caracterizado porque** la ley de composición es la operación « o exclusiva » XOR.

7. Circuito de acuerdo con una cualquiera de las reivindicaciones 1, 2 y 5, **caracterizado porque** la ley de composición es la operación de suma.

8. Circuito de acuerdo con una cualquiera de las reivindicaciones 1, 2 y 5, **caracterizado porque** la ley de composición es la operación de multiplicación.

9. Circuito de acuerdo con una cualquiera de las reivindicaciones 1, 2 y 5, **caracterizado porque** la ley de composición es la operación de tipo s = a*b + (a+b)/2 , siendo * la multiplicación y + la suma.

10. Circuito de acuerdo con una cualquiera de las reivindicaciones 1, 2 y 5, **caracterizado porque** la ley de composición tiene una distancia de Hamming entre dos submáscaras consecutivas m_2 de un ciclo al otro, Δm_2, no nula.

11. Circuito de acuerdo con la reivindicación 10, **caracterizado porque** al tener las submáscaras m_1, m_2 un número par de bits, la ley de composición de grupo (α) que permite obtener la máscara *m* a partir de las submáscaras m_1, m_2 verifica:

- $\Delta m_2 = 1$ por grupos de 2 bits ;
- $m = m_1 \oplus m_2$ si $m_2\ MSB = 1$ o $m = m_1$ si $m_2\ MSB = 0$

siendo respectivamente $m_2\ MSB$, $m_2\ LSB$ los bits de alto peso y de bajo peso de la submáscara m_2.

12. Circuito de acuerdo con una cualquiera de las reivindicaciones anteriores salvo 3 y 4, **caracterizado porque** comprende un generador de valores aleatorios (71), emitiendo dicho generador el valor de la segunda submáscara m'_2.

13. Circuito de acuerdo con una cualquiera de las reivindicaciones anteriores salvo 3 y 4, **caracterizado porque** esta submáscara m_1 procedente del primer registro de máscara M1 se guarda en el segundo registro de máscara M2 al final del ciclo, recibiendo por su parte el primer registro M1 la nueva máscara m'_1 de tal modo que reconstruya la nueva máscara completa $m' = m'_1 \theta m'_2$.

14. Circuito de acuerdo con una cualquiera de las reivindicaciones anteriores salvo 3 y 4, **caracterizado porque** el algoritmo de cifrado es de tipo DES, estando el camino de los datos *x*, *m* dividido en dos partes, izquierda (91, 221) y

derecha (92, 222), dicho circuito comprende además del registro de máscara izquierda ML (91) y del registro de máscara derecha MR (92) un tercer registro de máscara M (93), conteniendo el registro MR la submáscara derecha m_{r1} y conteniendo el registro ML la submáscara izquierda m_{l2}, conteniendo el registro M la submáscara m_{r2} igual a la submáscara m_{l1}, siendo los valores efectivos de la máscara derecha m_r y de la máscara izquierda m_l tales que:

- $m_r = m_{r1}$ θ m_{r2}
- $m_l = m_{l1}$ θ m_{l2}

15. Circuito de acuerdo con una cualquiera de las reivindicaciones anteriores, **caracterizado porque** la segunda caja de sustitución *S'* (21) comprende dos cajas idénticas a la primera caja de sustitución *S* (1) y dos puertas XOR (111, 112), estando las salidas de las dos cajas (1) conectadas a una puerta XOR (112) cuya salida emite la nueva máscara *m'*, entrando la variable enmascarada ($x \oplus m$) en una caja (1) y en la otra puerta XOR (111) cuya otra entrada recibe la máscara *m*, entrando la salida de esta puerta (111) en la otra caja (1)

16. Circuito de acuerdo con la reivindicación 15, **caracterizado porque** una memoria ROM (113) realiza la lógica de las puertas XOR (111, 112) y la sustitución de las cajas (1).

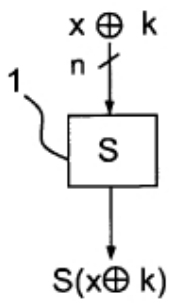

FIG.1a

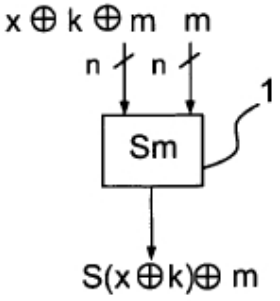

FIG.1b

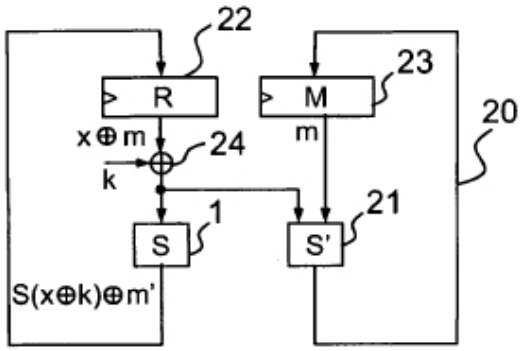

FIG.2

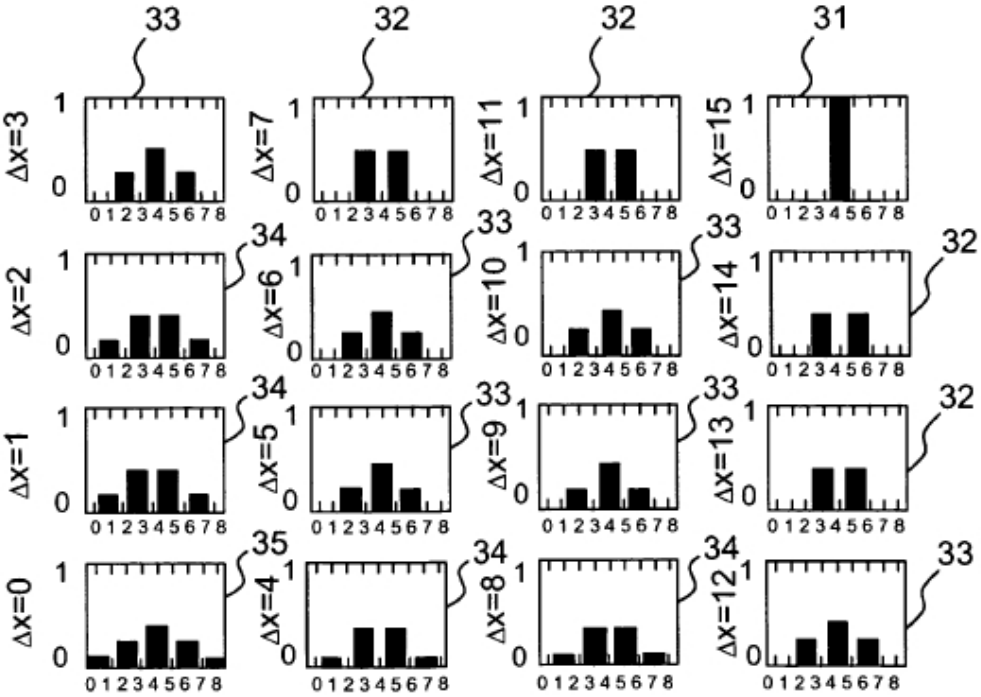

FIG.3

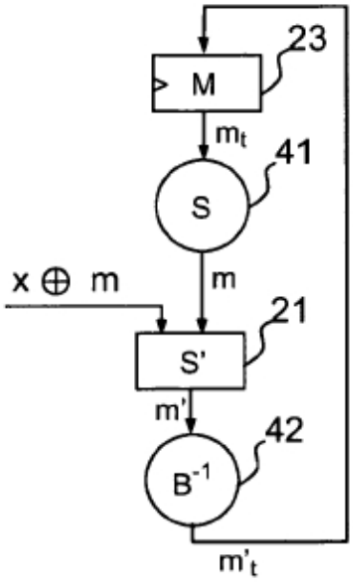
23
M
m_t
41
S
x ⊕ m
m
21
S'
m'
42
B^{-1}
m'_t

FIG.4

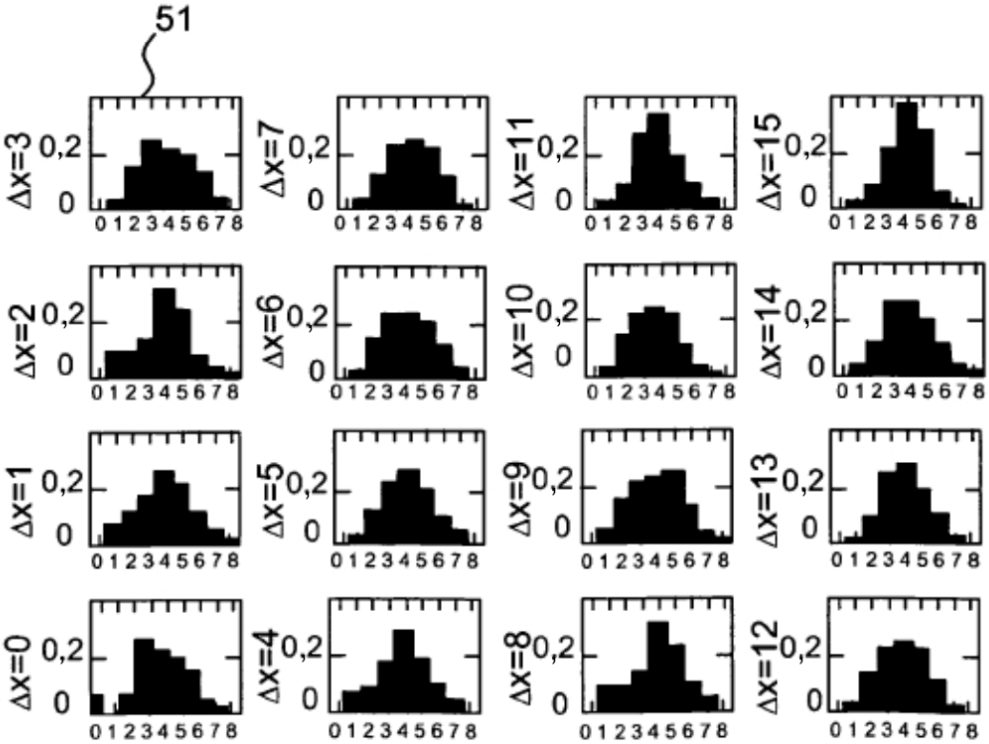
51
Δx=3
Δx=7
Δx=11
Δx=15
Δx=2
Δx=6
Δx=10
Δx=14
Δx=1
Δx=5
Δx=9
Δx=13
Δx=0
Δx=4
Δx=8
Δx=12
0,2
0
0 1 2 3 4 5 6 7 8

FIG.5

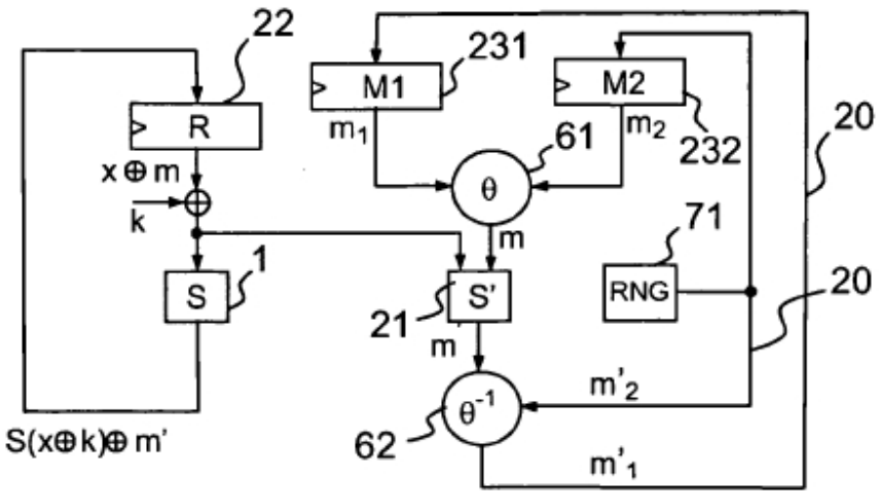
22
231
M1
M2
R
m1
m2
232
20
61
θ
x⊕m
k
m
71
1
S
S'
RNG
20
21
m
m'2
θ-1
62
S(x⊕k)⊕m'
m'1

FIG.6

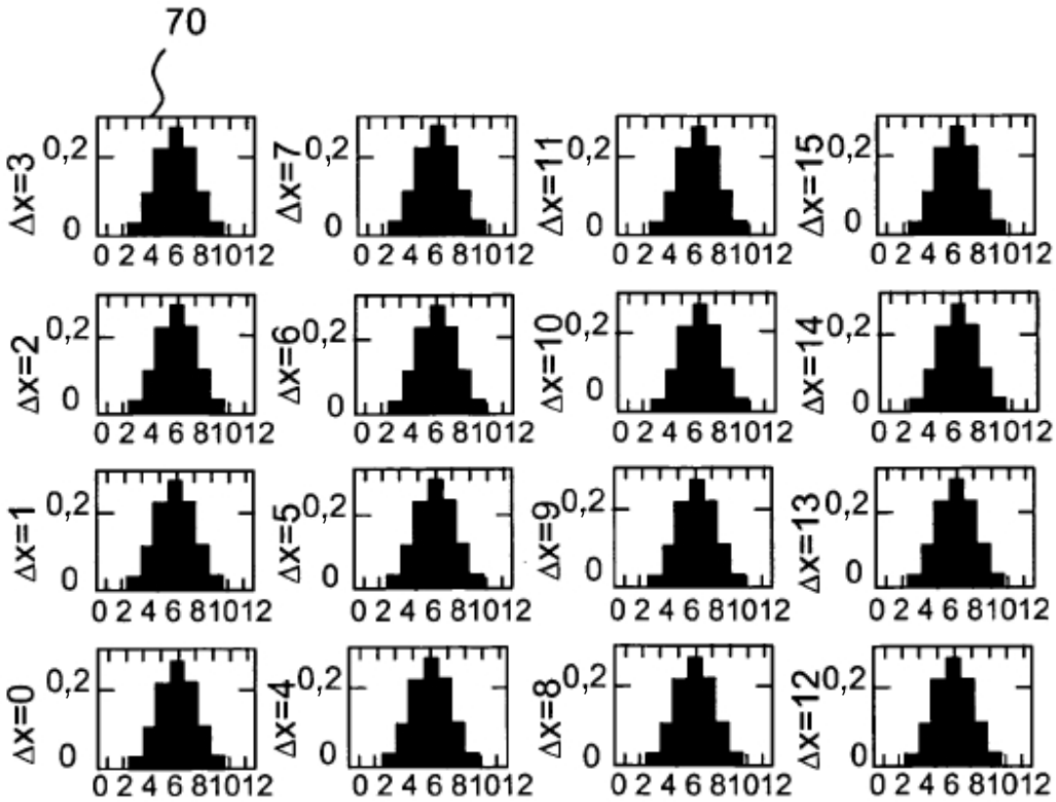
70
Δx=3
Δx=7
Δx=11
Δx=15
Δx=2
Δx=6
Δx=10
Δx=14
Δx=1
Δx=5
Δx=9
Δx=13
Δx=0
Δx=4
Δx=8
Δx=12
0,2
0
0 2 4 6 8 10 12

FIG.7

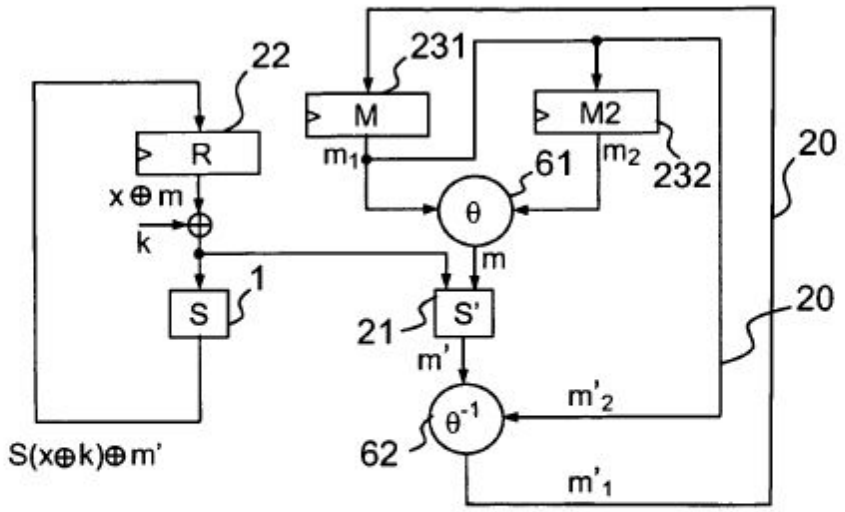
22
231
M
M2
R
m1
m2
232
20
61
x⊕m
θ
k
m
S
1
S'
21
20
m'
m'2
θ-1
62
S(x⊕k)⊕m'
m'1

FIG.8

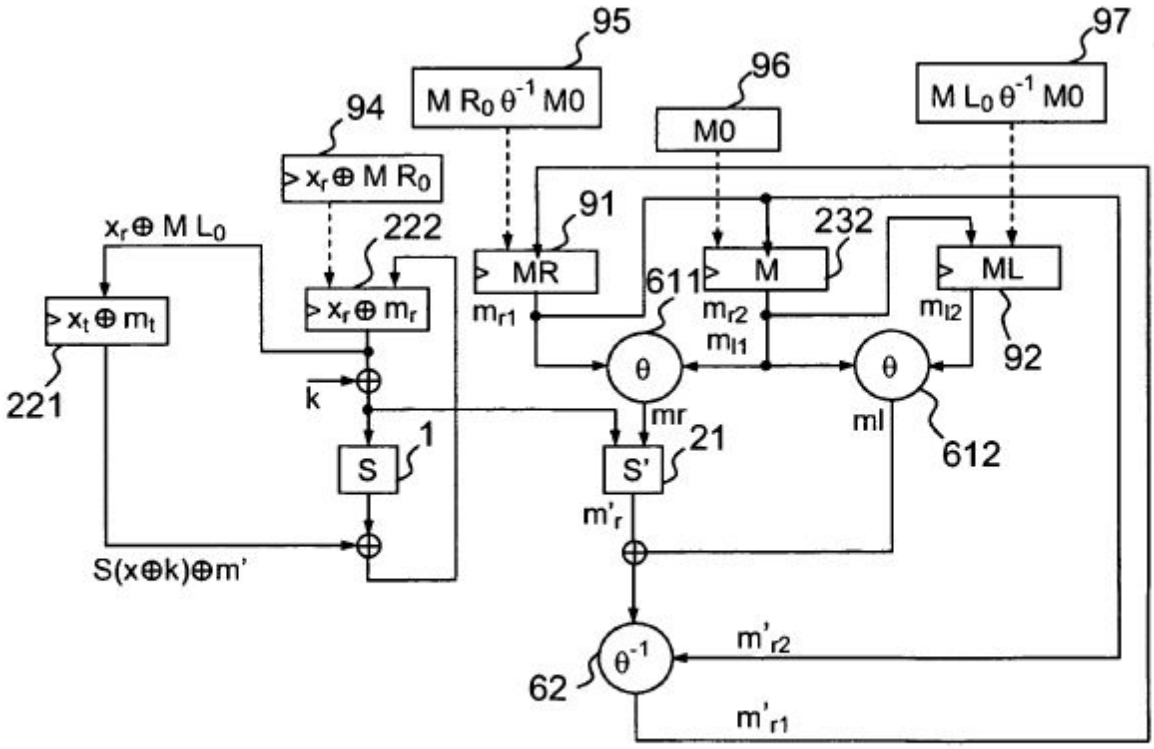
95
97
M R0 θ-1 M0
96
M L0 θ-1 M0
94
M0
xr⊕M R0
91
232
xr⊕M L0
222
MR
M
ML
611
xt⊕mt
xr⊕mr
mr1
mr2
ml2
92
ml1
θ
θ
221
k
mr
ml
612
1
21
S
S'
m'r
S(x⊕k)⊕m'
m'r2
θ-1
62
m'r1

FIG.9

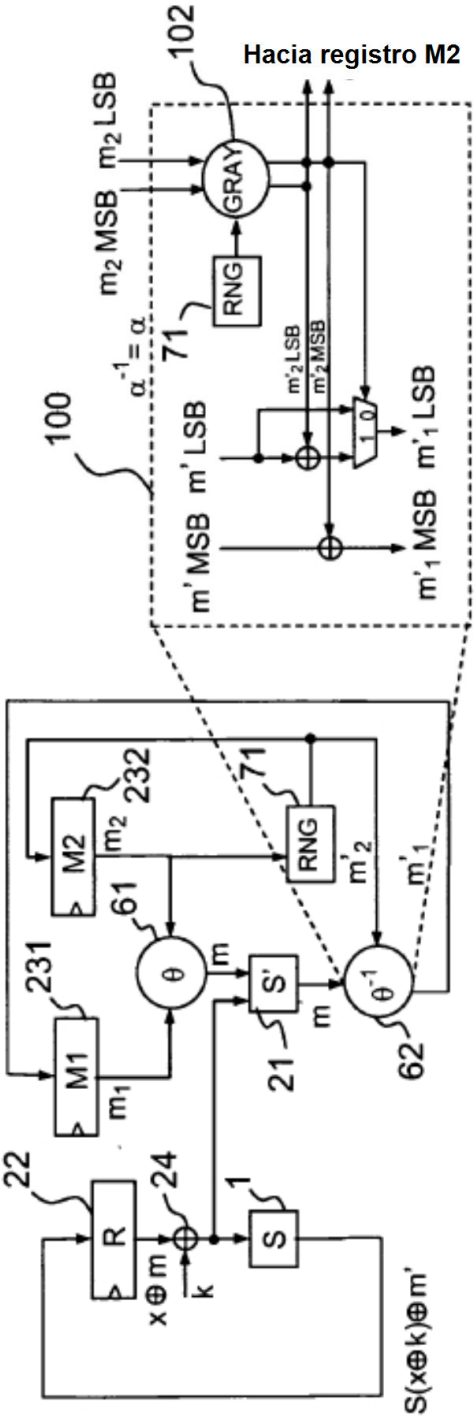
Hacia registro M2
102
GRAY
RNG
71
m_2 LSB
m_2 MSB
$\alpha^{-1}=\alpha$
100
m'LSB
m'MSB
m'_2LSB
m'_2MSB
m'_1LSB
m'_1MSB
M2
232
m_2
61
θ
m
S'
21
θ^{-1}
62
m'_2
m'_1
231
M1
m_1
22
R
24
x⊕m
k
1
S
S(x⊕k)⊕m'

FIG.10

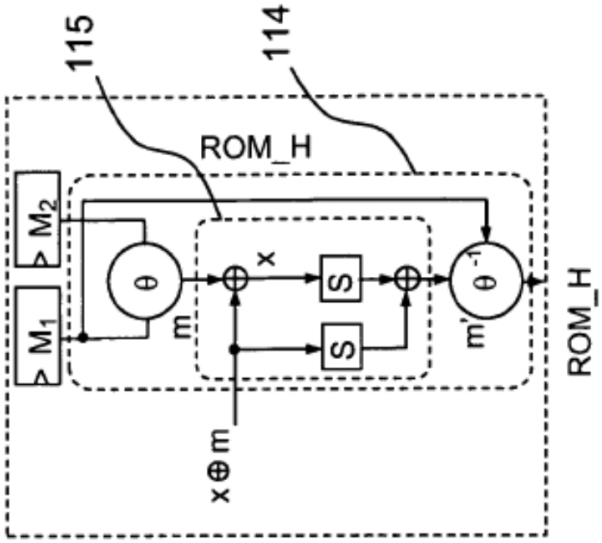

FIG.11c

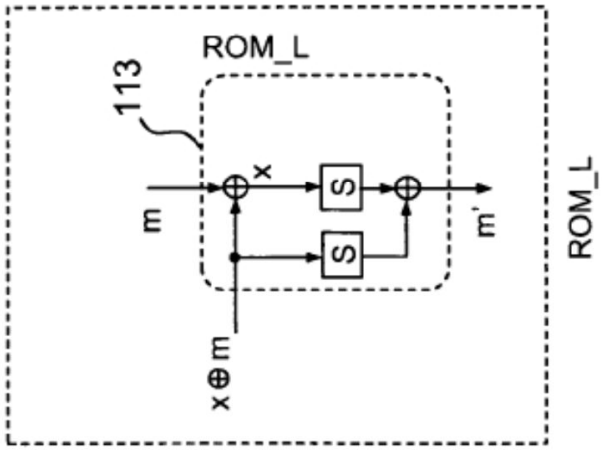

FIG.11b

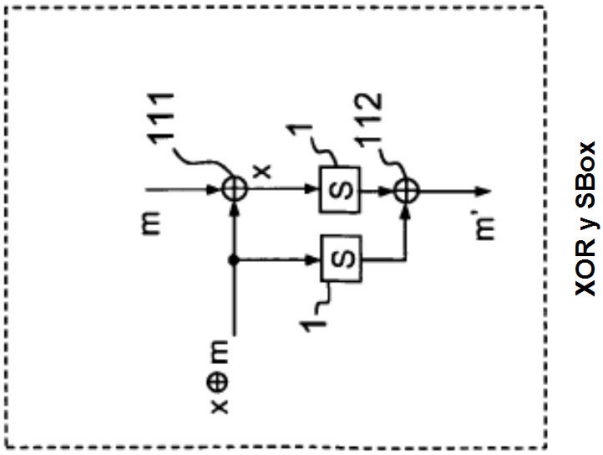

FIG.11a