(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 602 827**

(51) Int. CI.:

G09C 1/00 (2006.01)
H04L 9/00 (2006.01)
H04L 9/06 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **18.01.2010 PCT/EP2010/050547**

(87) Fecha y número de publicación internacional: **29.07.2010 WO10084107**

(96) Fecha de presentación y número de la solicitud europea: **18.01.2010 E 10700868 (2)**

(97) Fecha y número de publicación de la concesión europea: **17.08.2016 EP 2380305**

(54) Título: **Circuito de criptografía, protegido particularmente contra los ataques por observación de fugas de información mediante su cifrado**

(30) Prioridad:

20.01.2009 FR 0950342

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
22.02.2017

(73) Titular/es:

INSTITUT MINES-TELECOM (100.0%)
37-39 Rue Dareau
75014 Paris, FR

(72) Inventor/es:

DANGER, JEAN-LUC y
GUILLEY, SYLVAIN

(74) Agente/Representante:

CARPINTERO LÓPEZ, Mario

DESCRIPCIÓN

Circuito de criptografía, protegido particularmente contra los ataques por observación de fugas de información mediante su cifrado

La presente invención se refiere a un circuito de criptografía, protegido particularmente contra los ataques por observación de fugas de información mediante su cifrado.

Cada vez más sistemas de comunicación y de tratamiento de la información han recurrido a unos procedimientos criptográficos para prevenirse contra cualquier actuación maliciosa sobre los datos que se disponen a transitar sobre unos medios públicos. En particular, un cifrado asegura la confidencialidad de los datos, el resumen criptográfico asegura su integridad y la firma electrónica asegura su autenticidad. En cada uno de estos casos, se pone en juego un secreto común entre la parte encargada de la emisión de los datos y la parte encargada de la recepción de estos datos, estando eventualmente fusionadas estas dos partes. Para un atacante hostil a estos mecanismos de seguridad, es decir deseoso de conocer ilegítimamente el contenido del mensaje, de modificar el contenido de la transacción, de hacer impersonal o denegar la procedencia de un intercambio, un objetivo prioritario es encontrar el secreto común con el fin de beneficiarse impunemente de poderes similares a la parte receptora autorizada.
Han sido y son posibles siempre de alguna manera unos ataques directos contra unos algoritmos de criptografía. Sin embargo, se observa una disminución continua de los fallos lógicos. En particular, cada vez más algoritmos de criptografía están normalizados después de haber pasado a través de una prueba de escrutinio internacional. Este fue particularmente el caso para el cifrado simétrico AES (Advanced Encryption Standard) al final de los años 1990. Este mismo escenario se desarrolla actualmente para la versión 3 futura del algoritmo de troceado SHA (Secure Hash Algorithm).
Sin embargo, con la movilidad creciente de los medios de comunicación y de tratamiento de la información, pueden concebirse nuevos ataques. Observando el comportamiento temporal del sistema, en velocidad de ejecución, su comportamiento electrónico, el consumo de energía por un ataque DPA por ejemplo, o su comportamiento de radiación, en radiación magnética por un ataque EMA por ejemplo, pueden fugarse numerosas informaciones. Se han propuesto unas protecciones contra estos ataques en los canales auxiliares, sobre la base particularmente:

- de un disimulo, en donde se trata de hacer la fuga constante, en este caso independiente del secreto;
- de un enmascarado, donde se trata de hacer la fuga aleatoria, es decir imprevisible, por tanto inaprovechable.

Estas dos técnicas permiten hacer más difícil unos ataques que se dirijan a encontrar información, pero continúan siendo sin embargo vulnerables a los ataques que sacarían beneficio de defectos de implementación. Unos ejemplos de ataques DPA se describen en el documento de P. Kocher et ál.: Differential Power Analysis, In proceedings of CRYPT'99, volumen 1666 de LNCS, páginas 338-397, Springer-Verlag, 1999. Unos ejemplos de ataques EMA se describen en el documento de K. Gandolfi et ál.: Electromagnetic Analysis - Concrete Results, In CHES, volumen 2162 de LNCS, páginas 251-261, Springer-Verlag, 2001.
Existen numerosos ejemplos de vulnerabilidades potenciales o mostradas. Se pueden citar particularmente:

el disimulo a base de lógica diferencial (como WDDL) puede ser vulnerable a un ataque sobre unas diferencias de retardo combinatorias acumulativas entre una u otra de las fases del cálculo, fase de evaluación y fase de precarga
el enmascarado puede ser sensible a los ataques de orden elevado, denominados HO - DPA.

Un documento US 5 768 390 divulga un circuito de criptografía que incluye varias claves.

El documento FR2893796 describe un procedimiento de enmascarado de un algoritmo de cifrado en el que cada ronda necesita la utilización de 64 máscaras presentes en una tabla haciéndose la selección de la máscara de manera aleatoria. Siendo en el documento FR2893796 la máscara con todos los bits a 0 la máscara verdadera y que no enmascara la clave y por tanto corresponde al cálculo del cifrado.

Un objetivo de la invención es particularmente combatir estos ataques, particularmente del tipo DPA o EMA. Con este fin, la invención tiene por objeto un circuito de criptografía que incluye una clave funcional k_c para ejecutar un algoritmo de criptografía, caracterizado por que dicho circuito incluye una segunda clave k_i independiente de k_c y apropiada en cada instancia de dicho circuito, permitiendo proteger a éste contra los ataques que aprovechan los canales auxiliares del circuito.
Esta segunda clave puede o bien ser almacenada en una unidad de memorización dedicada o bien ser propia del componente.

La clave funcional k_c se enmascara por ejemplo mediante la segunda clave k_i combinando las dos claves mediante la operación XOR, siendo cifrada una variable de entrada x mediante la clave enmascarada $k_c \oplus k_i$.
La segunda clave k_i sirve para proteger la clave k_c gracias a una implementación confidencial.
La segunda clave k_i sirve por ejemplo para proteger un algoritmo confidencial, particularmente aquel que incluye un algoritmo criptográfico estándar personalizado por el enmarcado de dos funciones secretas protegidas por enmascarado con la clave k_i.
La segunda clave k_i se crea por ejemplo mediante una función del tipo PUF (Physically Unclonable Function) o POK (Physically Obfuscated Key).

La segunda clave k_i puede programarse también después de la fabricación del circuito, por personalización, con un valor aleatorio único en un recinto de seguridad.
El enmascarado introducido por la segunda clave k_i puede protegerse contra los ataques de orden elevado HO-DPA.
El conocimiento de la segunda clave k_i, que sirve de clave de implementación única a un circuito, permite por ejemplo la utilización de un procedimiento de control de protección a unos usuarios favorecidos que aseguran dicho control.
El circuito puede realizarse en un circuito programable de tipo FPGA.
La segunda clave k_i puede personalizarse, mediante el control del fichero de programación de una FPGA.
Ventajosamente, el circuito puede realizarse mediante una implementación de software.
Incluye por ejemplo una tercera clave k_b para cifrar el archivo de programación (25) de dicho circuito FPGA, confiriendo a este la confidencialidad de almacenamiento externo y de la transferencia de la clave k_i hacia la FPGA.
El cardinal de la segunda clave k_i es por ejemplo igual al cardinal de la clave funcional k_c, esto con el fin de hacer el ataque por los canales ocultos sobre k_i más difíciles que el ataque cripto-analítico sobre k_c.
El cardinal de la tercera clave k_b es superior o igual al cardinal de la clave funcional k_c.
El algoritmo de cifrado es el algoritmo DES.

Surgirán otras características y ventajas de la invención con la ayuda de la descripción que sigue, realizada en relación a los dibujos adjuntos que representan:

- La figura 1, un ejemplo de circuito que incluye una protección por enmascarado de la clave del algoritmo DES.
- La figura 2, el mismo circuito sin enmascarado.
- La figura 3, un ejemplo de codificación previa añadida al algoritmo de manera que proteja por enmascarado una implementación.
- La figura 4, una ilustración del principio de realización del circuito según la invención.

La figura 1 presenta un modo de enmascarado al que puede aplicarse la invención. Más particularmente, la figura 1 presenta a título de ejemplo una ilustración de enmascarado del algoritmo DES (Data Encryption Standard) implementado particularmente según la arquitectura descrita sintéticamente en el documento de S. Guilley et ál.: A fast Pipelined MultiMode DES Architecture Operating in IP Representation, Integration, The VLSI Journal, 40(4) páginas 479-489, julio de 2007, DOI. El circuito de la figura 1 se realiza por ejemplo en un circuito lógico programable del tipo FPGA (Field Programmable Gate Array). En este algoritmo, el camino de los datos se escinde en dos partes, izquierda y derecha. A título de comparación la figura 2 representa el mismo circuito en donde se pone en evidencia el sobrecoste de material para asegurar la protección por enmascarado, siendo indicados en trazos discontinuos los circuitos que implican este sobrecoste.
Por tanto un mensaje de entrada 1 se reparte entre un registro 3 de datos izquierdo y un registro 4 de datos derecho. Se reparte una máscara 2 entre un registro 5 de enmascarado izquierdo y un registro 6 de enmascarado derecho. Antes de ser almacenados en los registros de datos izquierdo y derecho, los datos del mensaje se enmascaran por combinación con los datos de la máscara por medio de una puerta 7 XOR a la izquierda y una puerta 8 XOR a la derecha. La clave 9 de cifrado, *k*, se enmascara también por la máscara *m* mediante una función de Feistel 10. El dato enmascarado del registro 6 derecho y la semi-máscara del registro 2 derecho forman las entradas de la función de Feistel donde el dato enmascarado derecho es cifrado ahí mediante una primera caja 9 de sustitución y donde la semi-máscara derecha se cifra mediante una segunda caja 16 de sustitución. Los datos de los registros 5 de datos izquierdo y 1 de la máscara izquierda se combinan respectivamente en el dato derecho y en la nueva máscara, en la salida de la función de Feistel, por medio de las puertas 11, 12 XOR y posteriormente se realimentan sobre los registros derechos, siendo recombinados a continuación los datos derecho e izquierdo mediante unas puertas 13, 14 XOR para transmitir en la salida 15 el mensaje cifrado. En un circuito del tipo de la figura 1, solo se supone que fugan los registros 5, 6 de datos.

Un circuito según la invención conserva la fuga pero la convierte en cifrada, por tanto incomprensible. De ese modo, un atacante que realice por ejemplo un ataque de tipo DPA o EMA no encuentra más que la variable:

$$K \oplus M \qquad (1)$$

es decir la clave secreta *K* cifrada a su vez mediante una máscara *M*. Este modo de protección de la clave *K* es conocido bajo el nombre de cifrado de Vernam, con la operación “O exclusiva”, también denominada XOR, y denotada por $\oplus$, siendo un código de Vernam un código que se puede cifrar con la operación XOR. Un circuito de criptografía según la invención se protege por tanto contra los ataques en los canales ocultos mediante cifrado de Vernam de las fugas de información.
Existen unos dominios de aplicación en donde el algoritmo de cifrado se personaliza completamente. Este es el caso por ejemplo en la esfera pública o privada para el GSM o la televisión de pago que se basan en la criptografía confidencial. Un argumento habitualmente puesto de relieve para activar esta elección es que los ataques sobre los canales auxiliares, denominados SCA (Side-Channel Attack), son imposibles porque la función de fuga a correlacionar con el circuito es desconocida. En el documento de K. Tiri et ál.: Side-Channel Leakage Tolerant Architectures, In ITNG’06 - Proceedings of the Third International Conference on Information Technology, New Generation, páginas 204-209, Washington DC, Estados Unidos, 2006 IEEE Computer Society, se propone modificar a la vez la implementación y la funcionalidad de un algoritmo, con o sin sobrecoste en cantidad de material. Un inconveniente de los dos procedimientos anteriores es que el cifrado se convierte en funcionalmente secreto. Esto

puede ser admisible en ciertos casos de la figura cuando unos profesionales de la seguridad implementan el sistema y su despliegue. Pero en la gran mayoría de los casos, cuando la concepción y la distribución de los sistemas de cifrado es difícil de supervisar, este escenario es muy incierto. Una vez recubierta la funcionalidad del secreto, se hace posible un ataque del tipo DPA de manera trivial. Además ciertas políticas de certificación, como por ejemplo FIPS-140, exigen la utilización no personalizada de normas de criptografía, lo que convierte a todos los procedimientos de tolerancia a las SCA preconizados, particularmente en el documento de K. Tiri et ál., en excluyentes.

Según la invención, para realizar un cifrado, respetando particularmente totalmente la especificación funcional conocida de este cifrado, se efectúa una protección por enmascarado utilizando una máscara apropiada para el circuito de criptografía a proteger. Un circuito según la invención incluye una arquitectura de enmascarado en donde la máscara *M*, propia del circuito, es simplemente constante y desconocida para el usuario o para el diseñador del circuito.

Se puede demostrar que un camino de enmascarado según la figura 1 realiza correctamente un cifrado de Vernam de la clave criptográfica de acuerdo con la ecuación (1) anterior, en el marco del ataque DPA de primer orden, es decir un ataque donde solo se supone que fugan los registros de datos 5, 6. Por otro lado, puede utilizarse igualmente cualquier variante alrededor del enmascarado para implementar la invención: es suficiente en efecto que la implementación se exprese de modo diferente a la implementación de referencia mientras se conserva la funcionalidad. En el caso del enmascarado, la implementación de referencia corresponde a aquella con una máscara nula (todo a cero); pero desde el momento en que la máscara es no nula, la implementación cambia, sin por tanto modificar la funcionalidad. Ahora bien es también posible cambiar de representación para introducir la variabilidad en la implementación. Por ejemplo, Jean-Sébastien CORON propone en “A New DPA Countermeasure Based on Permutation Tables. En SCN, volumen 5229 de Lecture Notes in Computer Science, páginas 278-292. Springer” modificar las partes de operaciones elementales del AES con la introducción de 2 biyecciones 4-bit $\rightarrow$ 4-bit, de tal manera sin embargo que al ensamblarlas, proporciona correctamente el cálculo de un AES clásico. Este cambio de representación puede dar lugar igualmente a una implementación secreta, cuya fuga de información no se estudia sin embargo en este documento.

De ese modo, los ataques en correlación del primer orden se hacen imposibles porque el modelo de fugas es desconocido. Además, los ataques que se basan en la constitución de un conjunto, o catálogo, de medidas, como los ataques denominados de “plantilla”, se convierten también en irrealizables porque al ser única cada implementación, es imposible construir un catálogo universal. Ventajosamente, con la invención, la diversidad de las implementaciones es comparable, incluso igual, al número de claves criptográficas. En particular, el ataque de tipo “segunda imagen previa” es entonces imposible. La probabilidad de encontrar por azar un circuito cuya clave sea programable que tenga la misma máscara que el circuito puesto servicio es comparable, incluso igual, a la probabilidad de adivinar por azar la clave correcta, es decir de lograr una búsqueda exhaustiva sobre la clave por ataque de fuerza bruta.
En el ejemplo de la figura 1, el material añadido para implementar el enmascarado está formado por unos registros de máscaras izquierda 1 y derecha 2 y unas puertas 12, 13, 14 XOR que combinan las máscaras con los datos, así como unos circuitos de sustitución 16 de la función de Feistel que tratan la salida del registro de máscara derecho.
En el marco de una realización mediante ASIC o FPGA, puede automatizarse el enmascarado de otros tipos de primitivas criptográficas con la asistencia de herramientas de CAO adaptadas que funcionan directamente sobre el código fuente.

Es interesante hacer notar que el procedimiento de protección puede aplicarse, de manera general a cualquier implementación que contenga un secreto susceptible de fugar a través de un canal auxiliar. Un ejemplo inmediato es la protección de claves de cifrado, aunque unas claves de firma se protegen también bien de la misma manera. Además, en lugar de proteger un parámetro de un algoritmo criptográfico, se puede igualmente proteger el algoritmo en sí, si es confidencial. Esto aparece en los sectores tales como la televisión de pago, en donde puede implementarse la criptografía no Inter-operativa porque las comunicaciones están cifradas de punto a punto (satélite hacia decodificador). Es entonces usual utilizar un algoritmo normalizado modificando en él uno o varios elementos (como las tablas de sustitución o las funciones de difusión). De esta manera, se asegura una personalización del algoritmo sin arriesgarse a debilitar su seguridad.

La figura 3 ilustra otra manera de proceder. En este ejemplo, se reutiliza un algoritmo estándar A tal cual, pero enmarcándole mediante unas codificaciones externas (EEin y EEout), de manera que la función realizada ya no sea A, sino la composición EEout $\circ$ A $\circ$ EEin. Una explicación de este principio se da en la introducción en el artículo de C. Clavier: Secret External Encodings Do Not Prevent Transient Fault Analysis, in CHES’07, volumen 4727 de Lecture Notes in Computer Science, páginas 181-194. La parte izquierda 30, 31, 32 de la figura 3 muestra cómo una técnica de enmascarado puede impedir que fuguen los valores EE(x), estando enmarcada la función 30 EE por dos registros 31, 32 en donde el primer registro 31 recibe el dato $x \oplus m$. La función 33 EE’ dispuesta en paralelo, definida como EE’(a, b) = EE(a) $\oplus$ EE(a $\oplus$ b), asegura que el desenmascarado continúa siendo posible. De ese modo, gracias al añadido del material 33, 34, 35 representado en la parte derecha de la figura 3, ninguno de los registros contiene EE(x), cualquiera que sea la entrada X del algoritmo. De esta manera, es imposible remontar a una información cualquiera sobre la codificación externa secreta EE. En lo que sigue, sin perder sin embargo generalidad, se concentrará en el caso de la figura de la protección de la fuga de una clave criptográfica.

Una solución del tipo FPGA permite ventajosamente a cada circuito tener su propia configuración, incluso durante un despliegue a gran escala. En particular con una solución FPGA, es inútil recompilar todo el sistema para modificar un valor, como particularmente la máscara propia en un componente, para personalizarlo. Esto implica que no se viola el principio de Kerckhoffs, siendo cada implementación efectivamente secreta, pero única. El compromiso de una implementación no permite el compromiso de todas las implantaciones.

La ingeniería inversa de la funcionalidad de ciertos circuitos FPGA puede hacerse posible debido a que se programa por software, en un fichero que se encuentra en una memoria legible permanentemente. Para evitar una ingeniería inversa de ese tipo, se puede utilizar un tipo de FPGA que permita cifrar este fichero, denominado de “bitstream”. De ese modo, la protección se guarda en sí misma confidencialmente mediante unos medios criptográficos. La ofuscación del código es una defensa suplementaria para completar la operación que se dirige a remontar desde lenguaje máquina hacia la especificación de alto nivel.

La figura 4 ilustra de manera esquemática y simplificada un ejemplo de circuito según la invención. Este circuito 21, de tipo FPGA, hace intervenir tres claves.

Una clave de funcionalidad k_c sirve para implementar el cifrado en el circuito 21. Este cifrado es por ejemplo el algoritmo 23 DES que transforma una variable de entrada x en una variable encriptada $y = DES(x, k_c)$ en el interior de un registro 22.

Una clave no funcional k_c, sirve para enmascarar la clave funcional k_c. Es esta clave k_i la que forma la máscara M de la clave funcional, un operador XOR combina estas dos claves en $k_c \oplus k_i$. La clave k sirve por tanto para proteger la clave funcional k_c de la implementación DES contra las fugas de información 24, por observación de radiación magnética o de consumo instantáneo particularmente.

Otra clave no funcional k_c, sirve para proteger los elementos secretos del archivo “bitstream” 25, es decir al menos k_i, incluso k_c.

Preferentemente, en este esquema, las claves se dimensionan de manera que:

$$| k_i | = | k_c | \qquad (2)$$

y

$$| k_b | \geq | k_c | \qquad (3)$$

expresando $|k_i|$, $|k_b|$, $|k_c|$ respectivamente el cardinal de k_i, de k_b y de k_c.

Según la invención la implementación del algoritmo criptográfico 23 es tal que la variable encriptada y es funcionalmente independiente de la clave k_i que protege la clave de cifrado k_c de la variable, siendo tan diversas las fugas de información de la implantación como $2^{|k_i|}$ (2 a la potencia de $|k_i|$).

En el caso de un algoritmo DES, $y = DES(x, k_c, k_i)$ con y independiente funcionalmente de ki.

Se ha de tomar nota que un ataque de primer orden no se hace simplemente más difícil sino imposible. Porque es necesario adivinar k_c conociendo $k_c \oplus k_i$, siendo k_i totalmente desconocido, incluyendo para un usuario o para un diseñador.

En eso, le invención aporta un alto grado de confianza, siendo probada la seguridad contra cualquier adversario que tenga una fuerza de cálculo inferior a $2^{|k_i|}$. Esto la devuelve al nivel de seguridad del algoritmo DES en sí mismo desde que $|k_i| = |k_c|$.

Es posible utilizar una función de tipo PUF (Physically Unclonable Functions), funciones físicamente no clonables o POK (Physically Obfuscated Key) clave física apropiada para la implementación, o cualquier otro sistema que permita generar un secreto apropiado en el circuito 21 en forma y lugar de una clave aportada desde el exterior, a través de una infraestructura de clave pública, denominada PKI, o cualquier otro mecanismo de personalización de confianza.

La segunda clave k_i puede programarse también después de la fabricación del circuito con un valor aleatorio único en un recinto de seguridad.

Es posible también utilizar un mecanismo de enmascarado de máscara constante, que utiliza como sobrecoste unas contramedidas a los ataques en la lógica combinatoria, también conocidos bajo nombre de “Shallow Attack”, o contra los ataques HO-DPA.

Se ha de tomar nota que el ataque sobre el enmascarado algorítmico que aprovecha la presencia de transiciones no funcionales, también denominados “glitchs”, poco dependientes de la máscara secreta, tal como se ha presentado particularmente en el documento de S. Mangard et ál.: Successfully Attacking Masked AES Hardware Implementations, In LNCS, editor, Proceedings of CHES’05, volumen 3659 de LNCS, páginas 157-171, Springer, septiembre de 2005, Edimburgo, Escocia, no se aplica a una implementación secreta, porque es imposible realizar una simulación del circuito, no conociéndolo. De hecho, este ataque se basa en una correlación con un modelo pre-caracterizado. Esta etapa es irrealizable con un circuito según la invención, salvo para un eventual atacante iniciado que conociera el diseño de las máscaras del ASIC realizado, o el archivo “bitstream” del FPGA, o que dispusiera de una muestra en donde la máscara puede elegirse. Para impedir esta posibilidad, se puede utilizar particularmente la función PUF descrita anteriormente.

Ciertos algoritmos propietarios, en particular los algoritmos normalizados encapsulados entre dos codificaciones secretas, no son resistentes a los ataques de perturbación como lo muestra particularmente el documento de C. Clavier: Secret External Encodings Do Not Prevent Transient Fault Analysis, In CHES, volumen 4727 de Lecture Notes in Computer Science, páginas 181-194, Springer, 2007. Esta clase de ataque necesita que el atacante pueda

fijar el valor de un registro a un valor conocido, como por ejemplo 0x00. En un circuito protegido por una clave de implementación k_i según la invención, esto es muy difícil en la práctica si los registros de datos y de la máscara son disjuntos, porque el atacante debería entonces realizar unos fallos múltiples mucho más difíciles de generar que unos fallos simples.

Un tipo de protección según la invención, de clave de implementación k_i, puede combinarse ventajosamente con otras protecciones tales como por ejemplo las protecciones usuales de detección de fallos, en el RTL para la codificación, o física para el encapsulado. Esto permite alcanzar un alto nivel de protección a la vez contra los ataques pasivos y contra los ataques activos.

REIVINDICACIONES

1. Circuito de criptografía (21) que incluye una clave funcional k_c para ejecutar un algoritmo de criptografía, incluyendo dicho circuito una segunda clave k_i, **caracterizado porque** dicha segunda clave es apropiada en cada instancia de dicho circuito, permitiendo proteger a éste contra los ataques que aprovechan los canales auxiliares de dicho circuito;
siendo enmascarada la clave funcional k_c mediante la segunda clave k_i combinando las dos claves mediante la operación XOR, siendo cifrada una variable de entrada x mediante la clave enmascarada $k_c \oplus k_i$;
siendo creada la segunda clave k_i mediante una función físicamente inclonable o PUF.

2. Circuito según la reivindicación 1, **caracterizado porque** el enmascarado introducido por la segunda clave k_i se protege contra los ataques de orden elevado HO-DPA mediante un enmascarado constante.

3. Circuito según una cualquiera de las reivindicaciones anteriores, **caracterizado porque** dicho circuito se realiza en un circuito programable de tipo FPGA.

4. Circuito según la reivindicación 3, **caracterizado porque** incluye una tercera clave k_s para cifrar el archivo de programación (25) de dicho circuito FPGA, confiriendo a éste la confidencialidad de almacenamiento externo y de la transferencia de la clave k_i hacia la FPGA.

5. Circuito según una cualquiera de las reivindicaciones anteriores, **caracterizado porque** el cardinal de la segunda clave k_i es igual al cardinal de la clave funcional k_c.

6. Circuito según una cualquiera de las reivindicaciones 4 o 5, **caracterizado porque** el cardinal de la tercera clave k_s es superior o igual al cardinal de la clave funcional k_c.

7. Circuito según una cualquiera de las reivindicaciones anteriores, **caracterizado porque** el algoritmo de cifrado es el algoritmo DES.

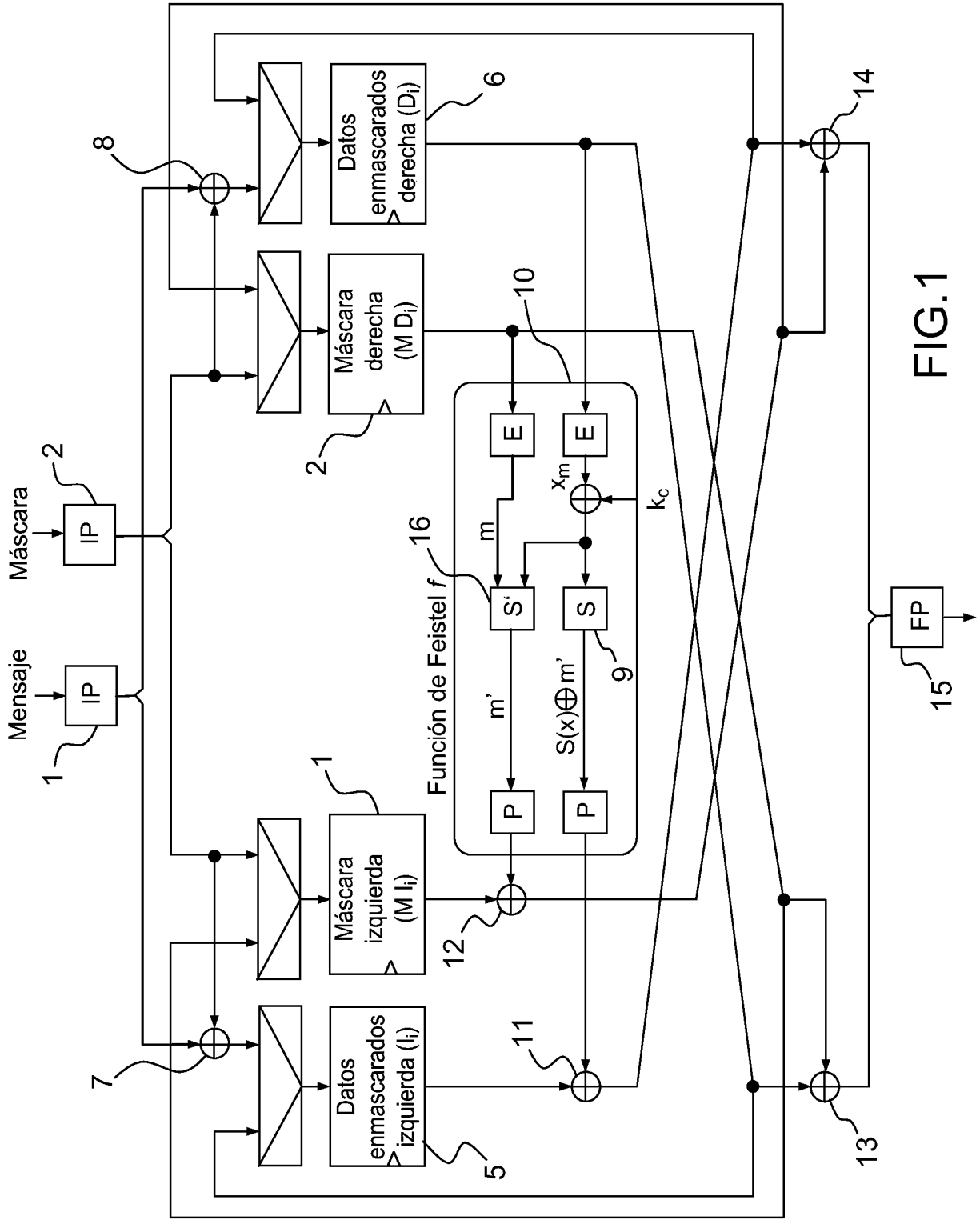
Máscara
Mensaje
IP
IP
1
2
Datos enmascarados izquierda (I_i)
Máscara izquierda ($M I_i$)
Datos enmascarados derecha (D_i)
Máscara derecha ($M D_i$)
5
6
7
8
9
10
11
12
13
14
15
16
Función de Feistel *f*
E
E
x_m
k_c
m
m'
S'
S
S(x)⊕m'
P
P
FP

FIG.1

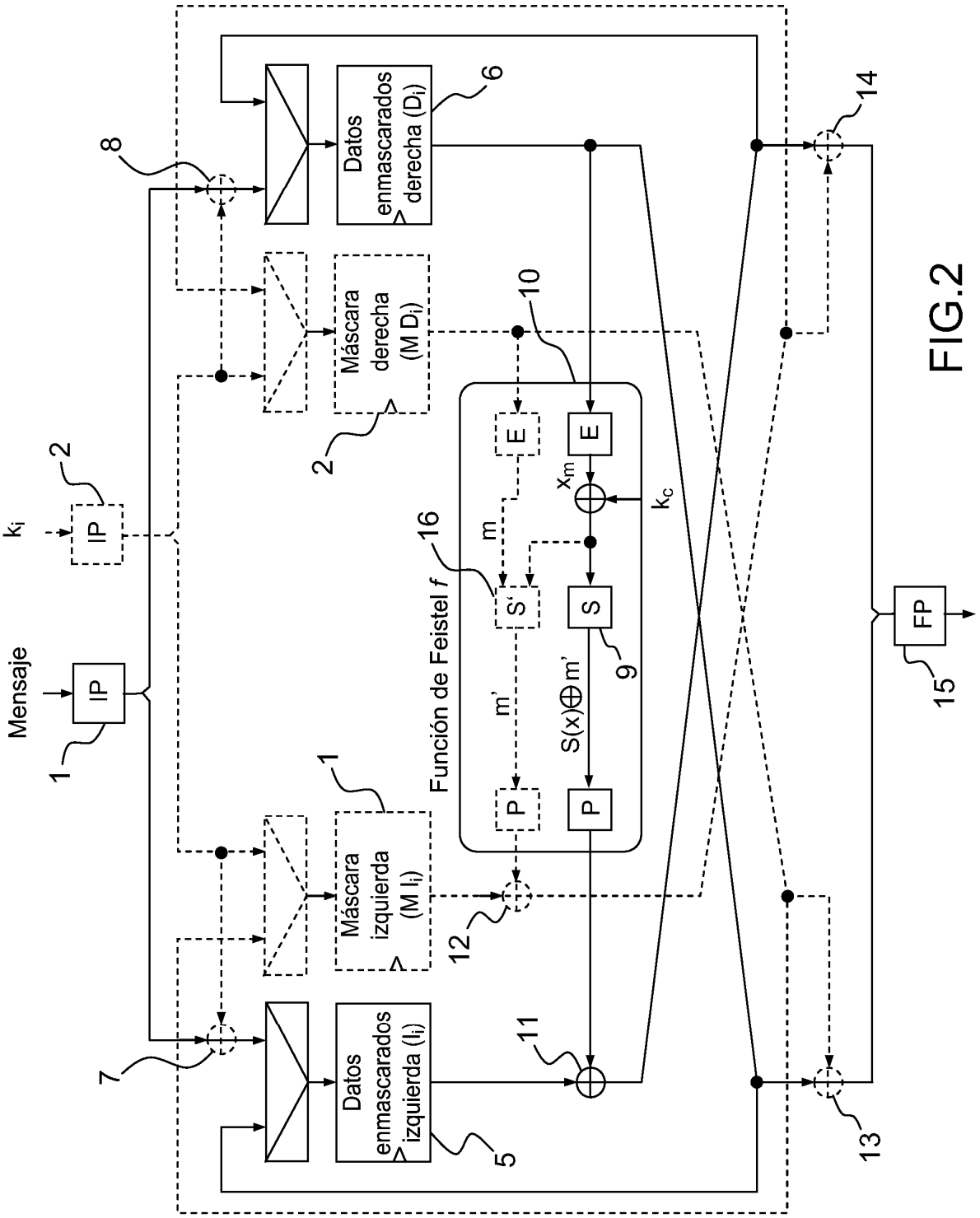
Mensaje
IP
1
k_i
IP
2
Datos enmascarados derecha (D_i)
6
Máscara derecha ($M D_i$)
2
Máscara izquierda ($M I_i$)
1
Datos enmascarados izquierda (I_i)
5
7
8
10
11
12
13
14
15
FP
Función de Feistel *f*
16
E
E
x_m
k_c
m
S'
S
9
m'
S(x)⊕ m'
P
P

FIG.2

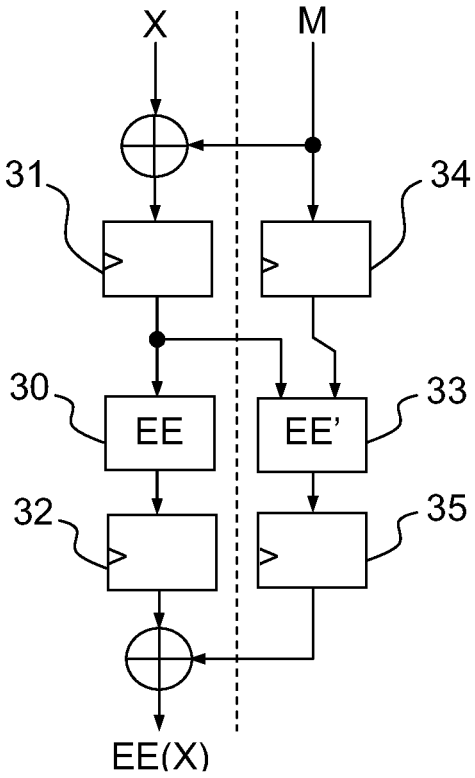

FIG.3

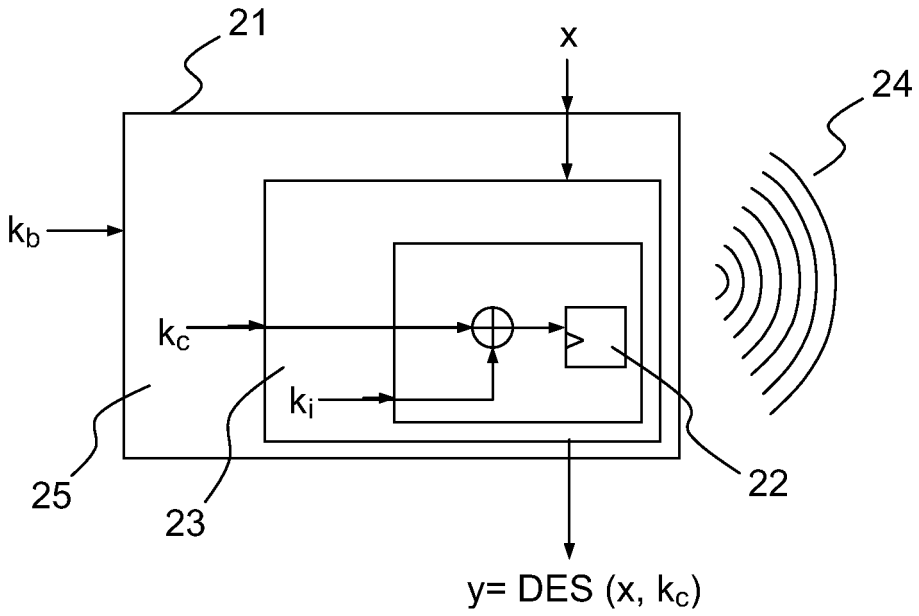

FIG.4