(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 546 560**

(51) Int. CI.:

H04L 9/30 (2006.01)
H04L 9/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **30.04.2012 E 12717737 (6)**

(97) Fecha y número de publicación de la concesión europea: **10.06.2015 EP 2707989**

(54) Título: **Dispositivo y procedimiento de generación de claves con seguridad reforzada para algoritmo de cifrado plenamente homomórfico**

(30) Prioridad:

09.05.2011 FR 1153981

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
24.09.2015

(73) Titular/es:

COMPAGNIE INDUSTRIELLE ET FINANCIÈRE D'INGÉNIERIE "INGENICO" (100.0%)
28-32, boulevard de Grenelle
75015 Paris, FR

(72) Inventor/es:

NACCACHE, DAVID;
CORON, JEAN-SÉBASTIEN y
TIBOUCHI, MEHDI

(74) Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

DESCRIPCIÓN

Dispositivo y procedimiento de generación de claves con seguridad reforzada para algoritmo de cifrado plenamente homomórfico

1. Campo de la invención

El campo de la invención es el de los dispositivos de cifrado llamado plenamente homomórfico.

Más exactamente, la invención se refiere a la implementación de operaciones y de procesamientos numéricos de generación de claves destinadas a un algoritmo de cifrado homomórfico implementado en microprocesadores, y ello al objeto de proveer de un nivel de seguridad significativamente más elevado que la técnica anterior.

La invención se refiere, muy particularmente, a las infraestructuras y dispositivos de generación de claves.

2. Técnica anterior

2.1. Criptografía de clave pública

El procesamiento criptográfico de datos numéricos muchas veces requiere efectuar operaciones de cifrado de clave pública.

En un algoritmo de cifrado de clave pública, el cifrador cifra un mensaje *m* con el concurso de un algoritmo de cifrado *E* en un cifrado *c = E(PK,m)*, con el concurso de una clave pública, denotada por *PK*.

El destinatario del mensaje descifra el cifrado c aplicando una función de descifrado *D* tal que *m = D(SK,c)* donde *SK* es una clave secreta vinculada a la clave pública PK.

Las claves pública y secreta (respectivamente PK y SK) son generadas con el concurso de un algoritmo probabilístico llamado algoritmo de generación de claves.

Por ejemplo, son algoritmos célebres de cifrado de clave pública el algoritmo llamado RSA descrito en la patente estadounidense U.S. 4.405.829, o el intercambio de claves Diffie-Hellman descrito en la patente estadounidense U.S. 4.200.770.

2.1. Criptografía plenamente homomórfica de clave pública

Reviste un particular interés, para numerosas aplicaciones prácticas, disponer de un Algoritmo Plenamente Homomórfico de Clave Pública (APHCP).

Un APHCP incluye, aparte de los algoritmos *E* y *D*, otros dos algoritmos denotados por *ADD* y *MUL* que tienen, para todo mensaje *m[1]* y *m[2]*, las siguientes propiedades:

- $m[1] \times m[2] = D(SK, MUL(E(PK,m[1]), E(PK,m[2])))$
- $m[1] + m[2] = D(SK, ADD(E(PK,m[1]), E(PK,m[2])))$

Es posible demostrar que, aun si las operaciones $m[1] + m[2]$ y $m[1] \times m[2]$ se entienden como módulo 2 (a saber, "+" representa la operación lógica de "o-exclusivo" y "$\times$" representa la "y lógica"), se puede codificar cualquier procesamiento complejo de datos con el concurso de estas dos únicas operaciones.

Las aplicaciones de los APHCP son múltiples:

- Unos APHCP permiten, por ejemplo, efectuar cálculos sobre los datos médicos de pacientes que figuran en una base de datos sin tener que revelar por ello su identidad.
- Unos APHCP permiten conocer el número de votos obtenidos por los candidatos de una elección sin que se desvele la identidad de los votantes.
- Unos APHCP permiten la creación de protocolos de pago anónimos.
- Unos APHCP permiten la creación de un sistema de ventas donde la cuantía de las pujas se mantendría desconocida, con el fin de evitar que el vendedor tienda a la sobrepuja. Sólo se desvelaría, al final del procedimiento, la mayor cuantía.

Un primer APHCP fue publicado por Craig Gentry en el documento **D1** correspondiente al artículo titulado "Fully Homomorphic Encryption Using Ideal Lattices" publicado en las actas del coloquio 41st ACM Symposium on Theory of Computing (STOC), 2009. Por la gran complejidad de implementación de la que adolece este procedimiento, fue propuesto un segundo procedimiento de APHCP, basado en la aritmética sobre los enteros, por Marten van Dijk, Craig Gentry, Shai Halevi, y Vinod Vaikuntanathan (vDGHV) en el documento **D2** correspondiente al artículo titulado

"Fully Homomorphic Encryption over the Integers" publicado en las actas del coloquio EUROCRYPT'2010, en las páginas 24 a 43.

Los documentos **D1** y **D2** se incorporan a la presente memoria descriptiva como referencia.

2.2. Método vDGHV

En el método vDGHV, el procedimiento de generación G de claves secretas y públicas empieza por generar un número impar *p* correspondiente a una clave secreta *SK*, denominada clave secreta vDGHV, y una clave pública *PK*, denominada clave pública vDGHV correspondiente a una colección de números enteros $x[i] = q[i] \times p + r[i]$ para *i* desde 0 a k, siendo *q[i]* y *r[i]* números aleatorios que cumplen con las imposiciones especificadas en el documento **D2**.

Los números *x[i]* son tales que *r[i]* es de pequeño tamaño con relación a *x[i]* (por ejemplo, *r[i]* es un número de 80 ó 100 bits).

Uno de los elementos de la clave pública vDGHV, el elemento denotado por *x[0]*, presenta una particularidad: para el elemento *x[0]*, debe cumplirse la siguiente condición inicial: *r[0] = 0*.

Con objeto de cifrar (por intermedio del algoritmo *E*) un bit *m*, el originador calcula: $c = m + 2r + 2Z$ donde:

- *r* es un número aleatorio de tamaño más o menos similar al de los *r[i]* (pudiendo ser la diferencia, por ejemplo, de un bit o dos);
- $Z = x[1]\, e[1] + ... + x[k]\, e[k]$ donde los *e[i]* son bits aleatorios (es decir, *e[i]* = 0 ó 1 de manera aleatoria).

Con objeto de descifrar (por intermedio del algoritmo *D*) un cifrado *c*, el receptor calcula: $m = (c \bmod p) \bmod 2$.

La implementación de las operaciones *ADD* y *MUL* utiliza la técnica llamada de "*bootstrapping"* (correspondiente a una técnica de inferencia estadística), conocida para un experto en la materia y descrita en el documento **D2**.

2.3. Implementación del procedimiento de generación G de claves vDGHV mediante un microprocesador que se comunica con un soporte físico generador aleatorio.

El procedimiento de generación de la clave pública vDGHV, que se ha tratado anteriormente, se implementa en un dispositivo físico 10 cuya arquitectura física se ilustra mediante la figura 1.

Un microprocesador 11 se halla conectado a un medio de interfaz de entrada y de salida de datos 12, a un generador aleatorio 13 y a una memoria 14 de la que el microprocesador lee las instrucciones que codifican un programa *Pg* que implementa el procedimiento de generación G de claves vDGHV.

En el inicio, el microprocesador 11 empieza a leer el programa *Pg* de la memoria 14. En su ejecución en el microprocesador 11, el programa *Pg* genera la clave secreta *SK* correspondiente a un número impar *p*, y la clave pública $PK = x[0], ..., x[k]$.

Una vez obtenidos los elementos *x[i]*, el programa *Pg* da al microprocesador 11 la instrucción de comunicar los elementos *x[0], ..., x[k]*, por intermedio de la interfaz de entrada y de salida de datos 12, con destino a otro dispositivo.

El procedimiento de generación G de claves vDGHV, ilustrado por la figura 2, implementa las siguientes etapas (en cualquier orden):

- Definir *r[0] = 0*;
- generar un número aleatorio impar *p* (correspondiente a la clave secreta *SK*);
- generar *k* números aleatorios *r[i]* denotados por *r[1], ..., r[k]*;
- generar *k+1* números aleatorios *q[i]* denotados por *q[0], ..., q[k]*.

Seguidamente se implementa una etapa de obtención con el fin de determinar los elementos $x[i] = q[i]\, p + r[i]$ para *i* desde *0 a k*, definitorios de la clave pública *PK*.

2.4. Inconvenientes de la técnica anterior

El procedimiento de generación G de claves vDGHV anteriormente mencionado presenta una brecha de seguridad.

En efecto, en la medida en que la clave secreta SK se corresponde con el número *p* que es un número impar aleatorio, es perfectamente posible que este número *p* pueda escribirse como un producto de factores primos:

$$p = p[1]^{a[1]} \times \quad \times p[L]^{a[L]}.$$
...

En el presente caso, los números *p[i]* representan números primos y los enteros *a[i]* representan potencias, es decir, el número de veces que cada *p[i]* aparece en la clave secreta p.

Para un experto en la materia, es sabido que existen métodos que permiten descomponer por completo o parcialmente *p* en factores primos. Por ejemplo, un primer método, conocido con el nombre de factorización en curva elíptica de Lenstra, permite extraer ciertos factores primos de números enteros. Este primer método se encuentra descrito en el artículo de Lenstra Jr., H. W. “Factoring integers with elliptic curves” publicado en la revista Annals of Mathematics (2) 126 (1987) páginas 649 a 673 e incorporado como referencia. Un segundo método conocido con el nombre de algoritmo de factorización por criba sobre los cuerpos de números generalizado también permite obtener tal descomposición.

Aplicando un método de factorización de este tipo a la clave pública $x[0] = p \times q[0] = q[0] \times p[1]^{a[1]} \times ... \times p[L]^{a[L]}$, un ocasional atacante podría descubrir al menos un factor *p[j]* integrante de la composición de *p*.

El atacante puede calcular seguidamente la cantidad *t = x[1] mod p[j]*. En efecto, *t = x[1] mod p[j] = r[1] mod p[j]*.

A partir de ahí, pueden presentarse dos escenarios:

1. Si *p[j]* > *r[1]*, entonces t = r[1], y la clave secreta puede ser determinada directamente calculando *p = PGCD(x[0], x[1]−t)*.

2. Si *p[j]* < *r[1]*, entonces el atacante determina el valor *t = r[1] mod p[j]*, lo cual le permite buscar exhaustivamente el valor de *r[1]* de manera más rápida. En este caso, el atacante tratará de calcular la cantidad *PGCD(x[0],x[1]−t−p[j]×i)* para diferentes valores de *i* hasta que, para un cierto valor de *i*, la operación *PGCD(x[0],x[1]−t-p[j]×i)* revele la clave secreta *SK* correspondiente al número impar aleatorio *p*.

Así, para un experto en la materia no resultaba evidente detectar y formular este problema de seguridad inherente a la utilización del procedimiento de generación G de claves vDGHV. La invención es, pues, al menos en parte, una invención de problema, correspondiente a la detección de esta brecha de seguridad.

3. Objetivos de la invención

La invención tiene como objetivo general subsanar al menos algunos inconvenientes de la técnica conocida de vDGHV.

Más exactamente, es un primer objetivo de la invención proveer una técnica que permite generar claves secretas y públicas resistentes para el método de APHCP de vDGHV anteriormente descrito.

Es otro objetivo de al menos una forma de realización de la invención proveer una técnica que permite incrementar el nivel de seguridad de las claves utilizadas para el cifrado y el descifrado.

4. Explicación de la invención

Se propone un procedimiento de generación de claves secretas y públicas vDGHV con seguridad reforzada, implementado en un dispositivo que comprende al menos un microprocesador y una memoria, caracterizado por comprender una etapa de generación de una clave secreta SK correspondiente a la generación de un número aleatorio *p* difícil o imposible de factorizar.

Semejante procedimiento asegura, según una primera forma de realización, la generación de claves reforzada con el concurso del algoritmo de cifrado plenamente homomórfico de clave pública publicado en el documento **D2**, modificado al objeto de incluir las siguientes etapas:

(a) Definir *r[0] = 0*;

(b) generar un número primo aleatorio *p*, que es, por definición, imposible de factorizar;

(c) generar *k* números aleatorios *r[i]* denotados por *r[1], ..., r[k]*;

(d) generar *k+1* números aleatorios *q[i]* denotados por *q[0], ..., q[k]*;

(e) formar los elementos de la clave pública *x[i] = q[i] p + r[i]* para *i* desde *0* a *k*;

(f) devolver la clave pública *{x[0], ..., x[k]}* y la clave secreta *p*.

De este modo, este procedimiento permite un incremento de seguridad debido a la imposibilidad computacional incrementada para dar con el valor de *p*.

En una variante, se propone un procedimiento de generación de claves reforzada para el algoritmo de cifrado plenamente homomórfico de clave pública publicado en el documento **D2**, modificado al objeto de incluir las siguientes etapas:

(a) Definir *r[0] = 0*;

(b) generar un número aleatorio *p* difícil de factorizar;

(c) generar *k* números aleatorios *r[i]* denotados por *r[1], ..., r[k]*;

(d) generar *k+1* números aleatorios *q[i]* denotados por *q[0], ..., q[k]*;

(e) formar los elementos de la clave pública *x[i] = q[i] p + r[i]* para *i* desde *0* a *k*;

(f) devolver la clave pública *{x[0], ..., x[k]}* y la clave secreta *p*.

Un número aleatorio *p* difícil de factorizar es un número cuyos tamaño y composición se eligen de modo que la operación de factorización (que tiene una complejidad exponencial en cuanto a tiempo de cálculo y recursos de memoria) sea irrealizable por parte de un atacante.

En otra forma de realización, se propone un dispositivo de cálculo que incluye un microprocesador conectado a un medio de interfaz de entrada y de salida de datos, a un generador aleatorio y a una memoria de la cual dicho microprocesador lee las instrucciones que codifican un programa inventivo de generación de claves que funciona según uno cualquiera de los procedimientos anteriormente descritos.

5. Lista de figuras

En la figura 1 se describe el dispositivo físico de generación de claves del procedimiento vDGHV de la técnica anterior.

En la figura 2 se describen las principales etapas del procedimiento de generación *G* de claves vDGHV.

La figura 3 presenta etapas de un procedimiento de generación G′ de claves, según una forma de realización de la invención.

6. Descripción de la invención

La generación inventiva de los elementos *x[i]* de la clave pública *PK* con seguridad reforzada para un algoritmo de tipo vDGHV en una arquitectura física se efectúa como sigue.

La arquitectura física del dispositivo según la invención (no representado) rescata los elementos de la arquitectura física del dispositivo 10 de la técnica anterior descrito en la figura 1, a saber, un microprocesador 11 conectado a un medio de interfaz de entrada y de salida de datos 12, a un generador aleatorio 13 y a una memoria 14 de la que el microprocesador 11 lee las instrucciones que codifican e implementan el procedimiento de generación G′ de claves según una forma de realización de la invención.

El procedimiento de generación G′ de claves difiere del procedimiento de generación G de claves anteriormente descrito por la etapa de generación de la clave secreta.

En el inicio, el microprocesador 11 genera la clave secreta *p* según una forma de realización de la invención, y los elementos correspondientes *x[0], ..., x[k]* de la clave pública.

Una vez generados los elementos *x[i]*, el dispositivo según la invención transmite los elementos *x[0], ..., x[k]* con destino a otro dispositivo, por intermedio de la interfaz de entrada y de salida de datos 12.

La figura 3 presenta unas etapas de un procedimiento de generación G′ de claves, según una forma de realización de la invención:

- Definir *r[0] = 0*;
- generar un número aleatorio *p* difícil o imposible de factorizar;
- generar *k* números aleatorios *r[i]* denotados por *r[1], ..., r[k]*;
- generar *k+1* números aleatorios *q[i]* denotados por *q[0], ..., q[k]*.

Obsérvese que estas etapas pueden ser realizadas en cualquier orden.

Seguidamente, se implementa una etapa de obtención con el fin de determinar los elementos *x[i] = q[i] p + r[i]* para *i* desde *0* a *k*, definitorios de la clave pública *PK*.

De acuerdo con una primera forma de realización, la clave secreta *SK* correspondiente al número *p* es un número primo secreto. El modo de generación de tales números primos secretos *p* es conocido para un experto en la materia y se utiliza, por ejemplo, con el fin de generar claves secretas para el algoritmo RSA.

De acuerdo con una segunda forma de realización, la clave secreta *SK* correspondiente al número *p* es un producto de números primos que es tal que el producto es difícil de factorizar. El modo de generación de tales números *p* es conocido para un experto en la materia y se utiliza, por ejemplo, con el fin de generar claves públicas para el algoritmo RSA.

En ambos casos, los tamaños de los parámetros *p*, *q[i]* y *r[i]* siguen las mismas recomendaciones que las descritas en el documento **D2**.

Por otro lado, una cualquiera de las variantes del procedimiento según la invención, anteriormente descritas, puede asimismo llevarse a la práctica en forma de soporte físico en un componente programable de tipo FPGA (“Field Programmable Gate Array” en inglés) o de tipo ASIC (“Application-Specific Integrated Circuit” en inglés).

REIVINDICACIONES

1. Procedimiento de generación de claves secretas y públicas obtenidas por medio de un algoritmo plenamente homomórfico de clave pública basado en la aritmética sobre los enteros, denominadas claves secretas y claves públicas vDGHV, con seguridad reforzada, implementado en un dispositivo que comprende al menos un microprocesador (11) y una memoria (14), **caracterizado por** comprender una etapa de generación de una clave secreta *SK* correspondiente a un número aleatorio *p* primo o producto de números primos cuyos tamaño y composición se eligen

de modo que la operación de factorización de dicho número aleatorio *p* sea irrealizable por parte de un atacante.

2. Procedimiento de generación de claves según la reivindicación 1, caracterizado por comprender las siguientes etapas:

(a) Definir *r[0] = 0*;

(b) generar dicha clave secreta *SK* correspondiente a dicho número aleatorio *p*;

(c) generar *k* números aleatorios *r[i*] denotados por *r[1], ..., r[k]*;

(d) generar *k+1* números aleatorios *q[i]* denotados por *q[0], ..., q[k]*;

(e) formar elementos *x[i] = q[i] p + r[i]* para *i* desde *0* a *k*, definitorios de una clave pública *PK*;

(f) devolver dicha clave pública *PK = {x[0], ..., x[k]}* y la clave secreta *SK = p*.

3. Dispositivo que comprende al menos un microprocesador (11) conectado a un medio de interfaz de entrada y de salida de datos (12), a un generador aleatorio (13) de claves secretas y públicas obtenidas por medio de un algoritmo plenamente homomórfico de clave pública basado en la aritmética sobre los enteros, denominadas claves secretas y claves públicas vDGHV, con seguridad reforzada, y a una memoria (14) en la que dicho microprocesador implementa medios de generación de una clave secreta *SK* correspondiente a un número aleatorio *p* primo o producto de números primos cuyos tamaño y composición se eligen de modo que la operación de factorización de dicho número aleatorio *p* sea irrealizable por parte de un atacante.

4. Dispositivo según la reivindicación 3, **caracterizado por que** dicho microprocesador (11) implementa medios de generación de una clave secreta *SK* correspondiente a un número primo aleatorio *p*.

5. Producto programa de ordenador, que comprende instrucciones de código de programa para la implementación del procedimiento según al menos una de las reivindicaciones 1 a 2 cuando dicho programa se ejecuta en un ordenador.

6. Medio de almacenamiento legible por ordenador y no transitorio, que almacena un programa de ordenador que comprende un juego de instrucciones ejecutables por un ordenador o un procesador para llevar a la práctica el procedimiento según al menos una de las reivindicaciones 1 a 2.

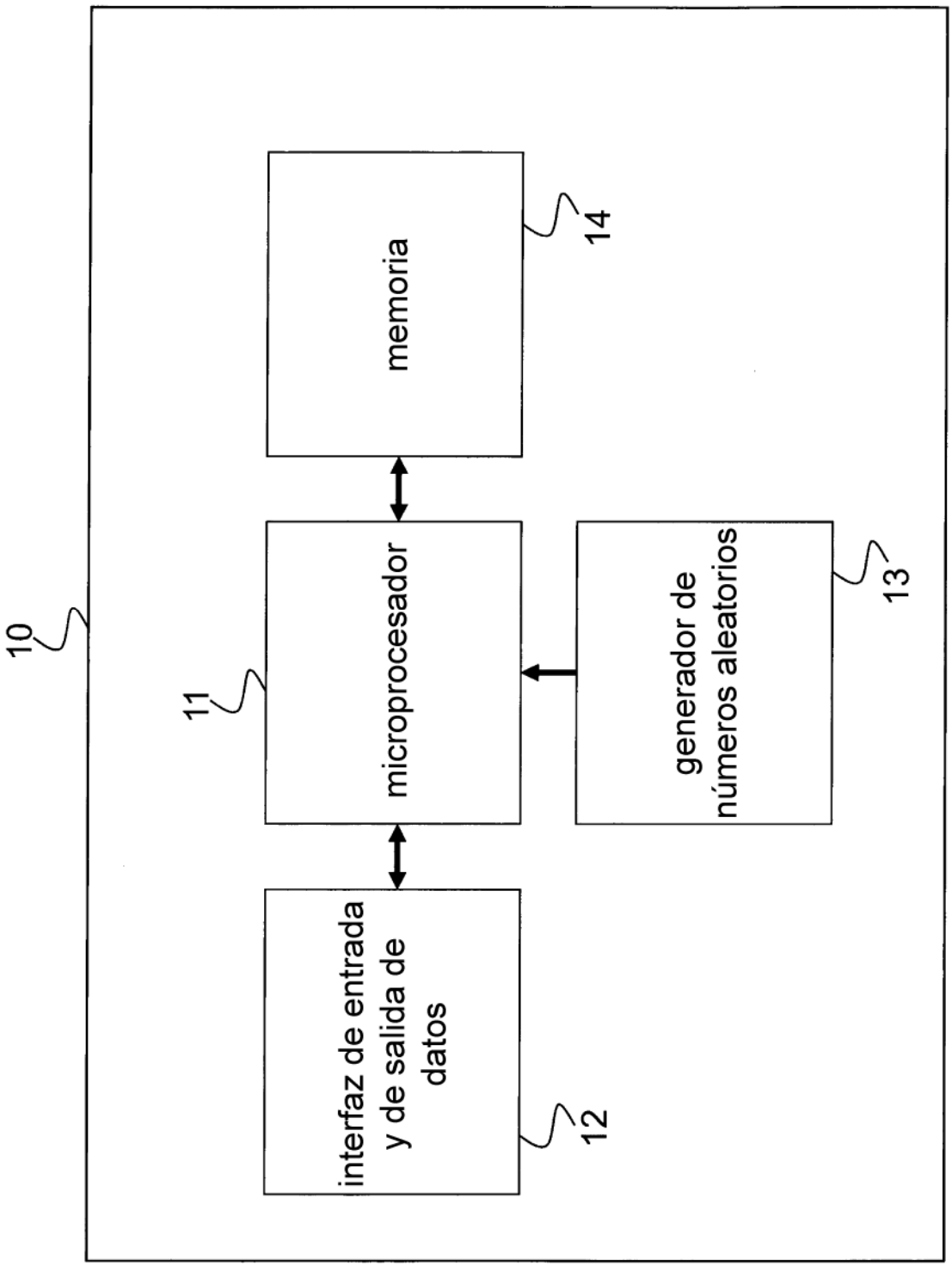
10
11
microprocesador
memoria
14
generador de números aleatorios
13
interfaz de entrada y de salida de datos
12

Figura 1

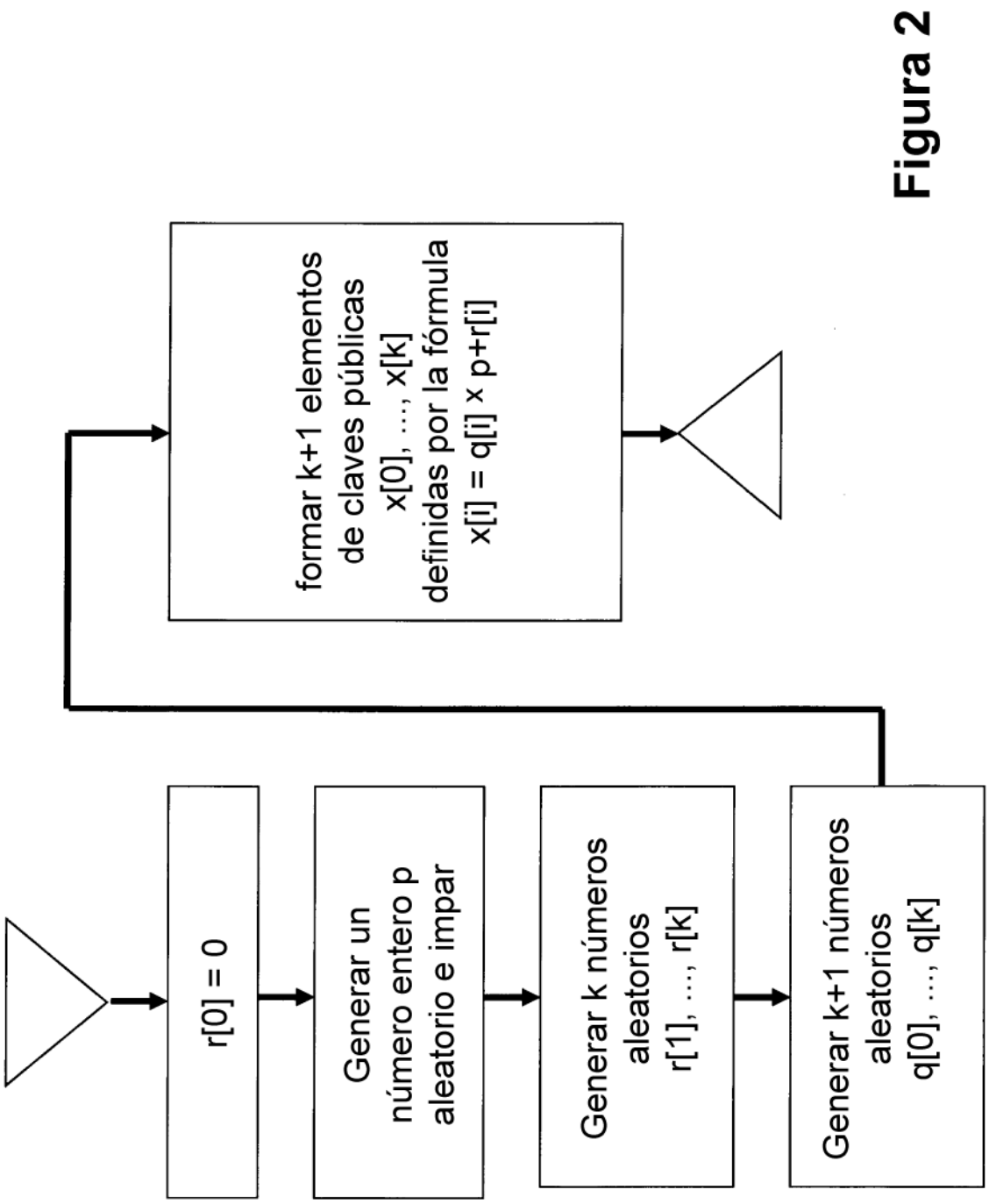
r[0] = 0
Generar un número entero p aleatorio e impar
Generar k números aleatorios r[1], ..., r[k]
Generar k+1 números aleatorios q[0], ..., q[k]
formar k+1 elementos de claves públicas x[0], ..., x[k] definidas por la fórmula x[i] = q[i] x p+r[i]

Figura 2

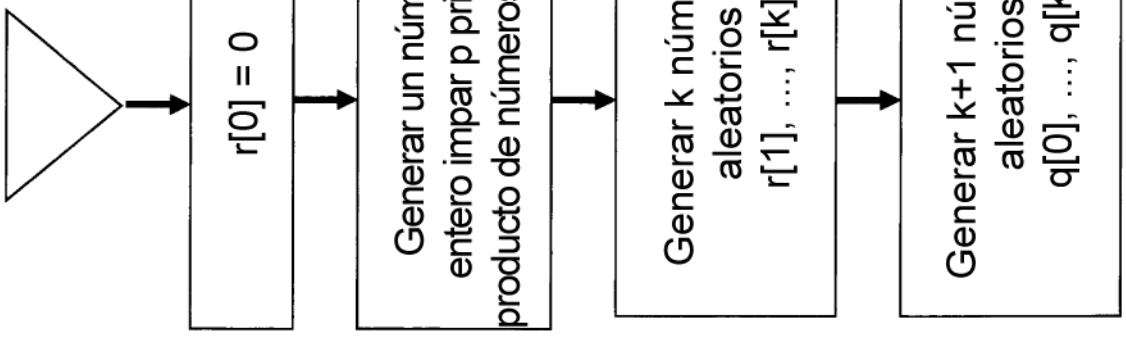
r[0] = 0
Generar un número entero impar p primo o producto de números primos
Generar k números aleatorios r[1], ..., r[k]
Generar k+1 números aleatorios q[0], ..., q[k]
formar k+1 elementos de claves públicas x[0], ..., x[k] definidas por la fórmula x[i] = q[i] x p+r[i]

Figura 3