(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 123 443**

(21) Número de solicitud: 9602747

(51) Int. Cl.[6]: H04L 9/06

(12) PATENTE DE INVENCION B1

(22) Fecha de presentación: **27.12.96**

(43) Fecha de publicación de la solicitud: **01.01.99**

Fecha de concesión: **30.06.99**

(45) Fecha de anuncio de la concesión: **16.09.99**

(45) Fecha de publicación del folleto de patente: **16.09.99**

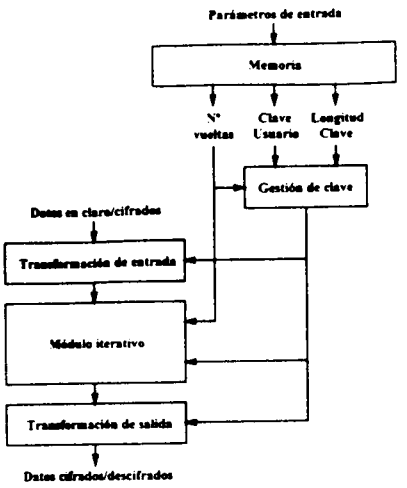
(73) Titular/es:
Consejo Superior Investigaciones Científicas
Serrano 117
28006 Madrid, ES

(72) Inventor/es: **Alvarez Marañón, Gonzalo;**
Fúster Sabater, Amparo;
Guía Martínez, Dolores de la;
Montoya Vitini, Fausto y
Peinado Domínguez, Alberto

(74) Agente: **No consta**

(54) Título: **Método y aparato para cifrado en bloque de datos.**

(57) Resumen:
Método y aparato para cifrado en bloque de datos.
Se describe un método y aparato para el cifrado de datos electrónicos que permite realizar de forma segura la transmisión, el almacenamiento y el procesado de los mismos. El esquema general de un cifrador en bloque está compuesto por tres elementos: transformación de entrada, módulo iterativo integrado por distintas operaciones y transformación de salida. Las aplicaciones de la invención como criptografía, tecnología de las comunicaciones, generación de secuencias pseudo-aleatorias, seguridad informática.



ES 2 123 443 B1

Aviso: Se puede realizar consulta prevista por el artº 37.3.8 LP.

Venta de fascículos: Oficina Española de Patentes y Marcas. C/Panamá, 1 – 28036 Madrid

DESCRIPCION

Método y aparato para cifrado en bloque de datos.

Sectores de la técnica en los que tiene aplicación

. Criptografía

. Tecnología de las comunicaciones

. Generación de secuencias pseudo-aleatorias

. Seguridad informática

Estado de la técnica anterior

El aumento en la transmisión, almacenamiento y manipulación de información por medios electrónicos exige elementos de protección que eviten la interceptación y/o la modificación de los datos por parte de personas ajenas a ellos.

Entre los distintos tipos de algoritmos criptográficos destacan los cifradores en bloque que operan sobre bits agrupados en bloques de tamaño variable según el diseño. Pertenecen a los algoritmos llamados simétricos, en los que se usa la misma clave tanto para el cifrado como para el descifrado. Emisor y receptor deben ponerse de acuerdo en cuanto a la clave que va a ser utilizada antes de iniciar la comunicación.

Los cifradores en bloque se implentan fácilmente tanto en forma de programa informático como de circuito electrónico, por lo que son aconsejables tanto en aplicaciones software como en aplicaciones hardware.

Existen numerosos algoritmos de cifrado en bloque. En el apartado de Referencias aparecen citados los más populares aunque hay otros menos conocidos de casas comerciales o de uso privado para organizaciones gubernamentales o militares. Muchos de ellos han sido ya criptoanalizados con éxito, mientras que otros son demasiado recientes y no han sido suficientemente analizados hasta el momento.

Los algoritmos de cifrado en bloque más difundidos son DES [DES 77], IDEA [LAI 90] y RC5 [RIV 94]. El DES, aunque todavía seguro, tiene sus días contados como estándar internacional debido a la corta longitud de su clave, tan sólo de 56 bits, que permite un ataque de fuerza bruta para encontrar la clave en, aproximadamente, tres horas y media con una máquina de un coste de un millón de dólares [WIE 93]. Además ha sido criptoanalizado por Shamir y Biham [BIH 93] para un reducido número de vueltas y para diversas variaciones del algoritmo y modos de implementación.

RC5 e IDEA se cuentan entre los posibles candidatos para convertirse en estándar del cifrado de datos.

Referencias

[DES 77] *Data Etictypliozi Standard.* National Bureau of Standards, FIPS PUB 46, Washington, DC, January 1977.

[SOR 84] A. SORKIN, *Lucifer: a Cryptographic Algorithm.* Cryptologia, 8(1), 22–35, 1984.

[MAD 84] W. E. MADRYGA,*A High Performance Ecryption Algorithm.* Computer Security: A global Challenge, North Holland: Elsevier Science Publishers, 557–570, 1984.

[SHI 88] A. SHIMIZU, S. MIYAGUCHI, *Fast Data Encipherment Algorithm FEAL.* Proceedings Eurocrypt'87, 267–278, Springer-Verlag, 1988.

[BRO 90] L. BROWN, J. PIEPRZYK, J. SEBERRY, *LOKI - a Cryptographic Primitive for Authenfication and Secrecy Applicaliotis.* Proceedings of Auscrypt 90, 229-236, Springer-Veriag, 1990.

[MER 90] R. MERKLE, *Fast Software Encryption Functions.* Proceedings of Crypto '90, Menezes and Vanstone ed., 476-501, Springer-Verlag,1991.

[LAI 90]. X. LAI, J. MASSEY, *A Proposal for a New Block Encryption Standard.* Proceeding of Eurocrypt 90, 389–404, Springer-Verlag, 1990.

[LAI 91] X. LAI, J. MASSEY, *Markov and Ciphers and Differential Cryptatialysis.* Proceedings of Eurocrypt 91, 17-38, Springer-Verlag, 1991.

[LAI 92] X. LAI, *On the Design and Security of Block Ciphers.* ETH Series in Information Processing, v. 1, Konstanz: Hartung-Gorre Verlang, 1992.

[WIE 93] M. J. WIENER, *Efficient DES Key Search.* Proceedings of Crypto'93, Springer-Verlag, 1993.

[BIH 93] E. BIHAM, A. SHAMIR: *Differential Cryptanalysis of the Data Encription Standard.* Springer-Verlag, 1993.

[RIV 94] R. L. RIVEST, *The RC5 Encryption Algorithm.* Proceedings of the Workshop on Cryptographic Algorithms, K. U. Leuven, Springer-Verlag, 1994.

[FUS 93] A. FUSTER, D. de la GUIA, J. NEGRILLO, F. MONTOYA, *Estructura no lineal para la generación de secuencias pseudoaleatorias.* N° de Solicitud: 9300561. Año: 1993. Entidad Titular: C.S.I.C. Estado: EN EXPLOTACION.

[GUI 96] D. de la GUIA, A. FUSTER. *Estudio de autómatas celulares para criptografía.* A publicar en Actas de la IV Reunión Española sobre Criptografía. Valladolid, 1996.

Explicación de la invención

Breve descripción de la invención

El esquema general de un cifrador en bloque (Fig. 1) está compuesto por tres elementos:

· transformación de entrada

· módulo iterativo integrado por distintas operaciones

· transformación de salida

Las entradas en un cifrador de este tipo son los datos en claro que el usuario quiere protege y las claves de cifrado, que se habrán calculado por algún método de expansión a partir de la clave de usuario. A la transformación de entrada llegan los datos en claro, divididos en bloques de un cierto tamaño, que son combinados con algunas de las claves para formar los bloques de datos de entrada al módulo iterativo.

Este módulo es una función, compuesta por distintas operaciones, que combina los datos de entrada con varias de las claves de cifrado y que se puede repetir tantas veces como quiera el usuario. Por este motivo, los datos de entrada proceden de la transformación de entrada en la primera iteración y de los datos de salida de la iteración previa en el resto de vueltas. La salida de esta función es un bloque de datos de igual tamaño que el bloque de entrada.

Una vez realizado el número de vueltas previsto, los datos de salida del módulo iterativo pasan a la transformación de salida que, combinándolos con las últimas claves, proporciona el bloque de datos cifrado.

El algoritmo propuesto objeto de esta patente sigue el esquema general descrito utilizando funciones que, aunque no son novedosas, si lo es la combinación de ellas que se ha realizado. Comparando con otros algoritmos similares, esta combinación proporciona una mejora de la seguridad de alguno de los casos (RC5, [RIV 94]) o mayor facilidad de implementación y aumento de la flexibilidad en otros ejemplos (IDEA, [LAI 90, 91]).

Una característica relevante es el uso de rotadores con un número variable de rotaciones, dependiente tanto de los datos de entrada como de la clave, modelo ya seguido por [FUS 93]. Las palabras de resultados intermedios son rotadas un número de posiciones determinado por otras palabras de resultados intermedios y por la clave, rasgo que refuerza su seguridad criptográfica, puesto que los bits rotan a posiciones aleatorias, que no están predeterminadas.

Una segunda característica es la mezcla de dos operaciones de grupos algebraicos diferentes, combinadas con el Grupo cíclico de rotaciones dependientes de los datos, de tal forma que ningún par de estas operaciones satisfaga una ley distributiva o conmutativa. Así se consigue ocultar cualquier relación entre los datos en claro, los datos cifrados y la clave, de modo que la estadística de relaciones entre ellos sea tan complicada que fracasen los criptoanálisis línea y diferencial. Las operaciones utilizadas, además de las rotaciones ya citadas, son la suma en complemento a 2 de palabras, correspondiente al grupo aditivo en Z_{2^w} y la disyunción exclusiva bit a bit de palabras, correspondiente al grupo aditivo en Z_2.

El algoritmo de cifrado/descifrado actúa sucesivamente sobre bloques de datos de una longitud predeterminada por el usuario. Ofrece una gran flexibilidad, ya que es posible seleccionar libremente parámetros como la longitud de la clave l, el tamaño de palabra w y el número de vueltas v, resultando así posible modificar a voluntad el compromiso entre velocidad de cálculo y nivel de seguridad requerido. Resulta adecuado tanto para hardware como para software, ya que las operaciones de cálculo son sencillas, están disponibles en todos los microprocesadores y su implementación en hardware resulta aún más fácil y eficiente.

Descripción detallada de la invención

En la figura 2 se muestra el diseño general del algoritmo. Para cifrar o descifrar cada bloque de datos son necesarias palabras de clave $Z_j^{(k)}$ que se habrán calculado previamente. Las claves $Z_1^{(k)}$ (k>0) tendrán longitud $\log_2$ 4w y cada bloque de datos necesitará $\theta_1 = (t.v) + 1$, donde v el número de iteraciones que se quiere realizar y t el número de palabras de clave utilizadas en el módulo iterativo. El resto de claves tendrán longitud w y el número necesario por cada bloque de datos será $\theta_2 = 4 + (t.v) + 4$. La distribución de su uso está reflejada en la tabla adjunta.

TABLA 1

Distribución de las palabras de clave $Z_j^{(K)}$ en los distintos elementos del cifrador/descifrador

Elemento	Sub-bloques de cifrado	Sub-bloques de descrifrado
Transformación de entrada	$Z_1^{(0)}$, $Z_2^{(0)}$, $Z^{(0)}{}_3$, $Z_4^{(0)}$	$-Z_2^{(v+1)}$, $Z_3^{(v+1)}$, $Z_4^{(v+1)}$,$- Z_5^{(v+1)}$
vuelta 1	$Z_1^{(1)}$, $Z_2^{(1)}$, K, $Z^{(1)}{}_t$	$(Z_1^{(v+1)})^{-1}$, $Z_2^{(v)}$,K, $Z_t^{(v)}$
vuelta 2	$Z_1^{(2)}$, $Z_2^{(2)}$, K, $Z^{(2)}{}_t$	$(Z_1^{(v)})^{-1}$, $Z_2^{(v-1)}$,K, $Z_t^{(v-1)}$
vuelta i	$Z_1^{(i)}$, $Z_2^{(i)}$, K, $Z^{(1)}{}_t$	$(Z_1^{(v+2-i)})^{-1}$, $Z_2^{(v+1-i)}$,K, $Z_t^{(v+1-i)}$
vuelta v	$Z_1^{(v)}$, $Z_2^{(v)}$, K, $Z^{(v)}{}_t$	$(Z_1^{(2)})^{-1}$, $Z_2^{(1)}$,K, $Z_t^{(1)}$
Transformación de salida	$Z_1^{(v+1)}$, $Z_2^{(v+1)}$, K, $Z_5^{(v+1)}$	$(Z_1^{(1)})^{-1}$,$-Z_1^{(0)}$,$Z_2^{(0)}$,$Z_3^{(0)}$,$-Z_4^{(0)}$

Los datos en claro se someten a un proceso de reorganización, previo a su entrada en el algoritmo, que consiste en formar bloques de longitud 4w bits de la forma X = (X1, X2, X3, X4) y que constituyen la entrada a la transformación de entrada. Cada uno de los sub bloques Xj se mezcla con una palabra de clave $Z_j^{(0)}$ mediante la suma en Z_{2^w} y la suma en Z_2 según queda indicado en la figura 2. El sentido de estas operaciones es evitar que al módulo iterativo lleguen bloques con todos los datos idénticos e iguales a 000...0 o iguales a 111...1 y entorpecer así el criptoanálisis diferencial.

A continuación se ejecuta v veces el módulo iterativo. La primera operación que se realiza en cada vuelta i es una rotación conjunta de todos los bits de entrada controlada por la palabra de clave $Z_1^{(i)}$. La razón de esta operación es mejorar el efecto de difusión cuando se realizan varias vueltas del módulo iterativo evitando que los datos que lleguen a la transformación de salida sean los mismos sub-bloques de entrada a los que se han ido sumando en Z_2 los datos Q1 y Q2 generados en las estructuras suma-rotación (figura 3) del módulo iterativo según se ve en la figura 2. Si esta operación no estuviera presente se podría intentar un ataque por búsqueda exhaustiva que permitiría encontrar los datos de partida de cada sub-bloque Xj con solo 2^{w+1} intentos.

En las figuras 2 y 3 se aprecian dos columnas, cada una compuesta por sumadores y rotadores, estando el número de posiciones rotadas determinado por la otra columna. El número de elementos que

compongan estas columnas debe estar relacionado con el tamaño de palabra elegido w de manera que se asegure con un 100% de probabilidad que todos los bits de los datos en claro y de la clave determinen al menos una rotación. Cada rotación va seguida de una suma en Z_{2^w} con una palabra de clave $Z_j^{(i)}$ excepto la última rotación de cada columna. Estas dos operaciones realizadas de forma consecutiva no tienen las propiedades asociativa y distributiva, lo que contribuye a dificultar el criptoanálisis.

Los registros de rotación puede tener, en principio, cualquier longitud p menor o igual a w aunque se aconseja que p sea un número primo. La razón de esta preferencia es que cuando se rota un número primo p de bits se consigue un vector de salida del rotador distinto al vector de entrada cualquiera que sea éste, excepto, por supuesto, cuando el número de rotaciones es igual a cero. Al usar un número compuesto y recibir el rotador como vector de entrada un patrón repetitivo, es decir, un patrón con períodos de repetición igual a divisores de la longitud del rotador, se pueden producir rotaciones invariantes cuya salida es idéntica a la entrada. Sin embargo, con un número primo, al no tener más divisores que él mismo y la unidad, cualquier patrón de entrada producirá salidas diferentes para cualquier rotación distinta de cero. La conveniencia de esta operación ha sido estudiada en [GUI 96].

Veamos un ejemplo aclaratorio: supongamos un rotador de longitud p = 4 al que llega un vector de entrada repetitivo 1010. El vector de salida será el mismo cuando reciba la instrucción de rotar cero posiciones o la instrucción de rotar dos posiciones. Sin embargo, si p = 5 con cualquier vector de entrada se obtendrá un vector de salida diferente, como puede verse en la siguiente tabla.

TABLA 2
Comparación de los vectores de salida de rotadores con distintas longitudes

Longitud del rotador	Vector inicial	Número de rotaciones				
		0	1	2	3	4
p = 4	1010	1010	0101	1010	0101	
p = 5	10101	10101	01011	10110	01101	11010

Los registros de rotación pueden ser tratados como un registro único o como dos o mas registros. En la figura se han dividido en dos registros de longitudes p y w-p para simplificar el dibujo. En esta figura se observa cómo se ejecutan las rotaciones de p bits, tomándose los p bits más y menos significativos de la palabra alternativamente. En cuanto a los bits restantes w-p se pueden rotar o no a voluntad del usuario. Se puede utilizar cualquier combinación de las anteriores en la medida en que todos los w bits de las palabras P1 y P2 estén implicados en el control de al menos una rotación. Así, para los parámetros que aparecen en la figura 3 - p, w, m, n, r, s, t - son válidos todos los valores que permitan cumplir esta condición. La figura 3 tal y como está dibujada tendría un número impar de rotadores en cada columna.

Después de la última vuelta, se efectúa una transformación de salida y se enlazan los cuatro subbloques a la salida del algoritmo, Y1, Y2, Y3, Y4, para conformar el bloque de salida cifrado Y = (Y1, Y2, Y3, Y4). Las operaciones que se realizan aquí, rotación de un registro de longitud 4w bits, y las sumas en Z_{2^w} y en Z_2, no buscan tanto aumentar la seguridad del algoritmo como conseguir que la transformación total de la operación de cifrado sea una involución. Esto significa que para descifrar basta con repetir las transformaciones realizadas en el proceso de cifrado. El único cambio necesario entre ambos procesos, cifrado y descifrado, es la asignación de las palabras de clave $Z^{(i)}{}_j$ a los distintos elementos como se aprecia en la Tabla 1.

Explicación detallada de las figuras

Figura 1

Esquema general de un cifrador en bloque.

Figura 2

Esquema general del cifrador en bloque propuesto

· *Símbolos generales*

+: Suma módulo 2^w de enteros de w bits

$\oplus$: OR exclusivo bit a bit de sub-bloques de w bits

$\lll$: Rotación

· *Transformación de entrada:*

Xj: Sub-bloque de datos en claro (cifrados) de w bits

$Z_j^{(0)}$: Palabra de clave de w bits

$1 \leq j \leq 4$

· *Módulo iterativo:*

$Z_j^{(i)}$: Palabra de clave de w bits para el j-ésimo elemento en la i-ésima iteración

$1 \leq i \leq v$

$1 \leq j \leq t$

v : Número de vueltas o iteraciones

t : Número de elementos del módulo iterativo

P1,P2: Palabras de entrada de w bits a las estructuras suma-rotación

Q1,Q2:Palabras de salida de w bits de las estructuras suma-rotación

· *Transformación de salida:*

Yj: Sub-bloque de datos cifrados (en claro) de w bits

$Z_k^{(v+1)}$: Palabra de clave de w bits

$1 \leq j \leq 4$

$1 \leq k \leq 5$

Figura 3

Estructura de suma-rotación.

+: Suma módulo 2^w de enteros de ^{w} bits

$\lll$: Rotación

P1,P2: Palabras de entrada de w bits a las estructuras suma-rotación

Q1,Q2: Palabras de salida de w bits de las estructuras suma-rotación

Z_j: Palabra de clave de w bits para el j-ésimo elemento de la estructura

$2 \leq j \leq t$

t : Número de elementos del módulo iterativo

w : Número de bits de la palabra básica utilizada

p : Número de bits, ≤ w, en que se puede dividir la palabra básica

m, n, r, s: parámetros para seleccionar la longitud de los indicadores del número de rotaciones. Su valor será elegido de acuerdo con las normas exigidas al diseño.

Figura 4

Estructura de suma-rotación en un diseño concreto para w = 32 bits.

+ : Suma módulo 2^{32} de enteros de 32 bits

 : Rotación

C1,C2: Columnas de la estructura suma-rotación

P1,P2: Palabras de entrada de 32 bits a las estructuras suma-rotación

Q1,Q2: Palabras de salida de 32 bits de las estructuras suma-rotación

Z_j: Palabra de clave de 32 bits para el j-ésimo elemento de la estructura

$2 \leq j \leq 13$

p1[..]: bits procedentes de P1 para control de rotaciones en C2

q2[..]: bits procedentes de Q2 para control de rotaciones en C1

Figura 5a

Resultado del test de correlación cruzada entre los ficheros de texto en claro y texto cifrado.

Figura 5b

Resultado del test de rachas en el fichero de texto cifrado

Exposición de un modo de la realización de la invención

Descripción

A continuación se describe una posible implementación para w = 32 bits.

El texto en claro se reorganiza en bloques, X, de 128 bits que entran en la transformación de entrada del cifrador divididos en cuatro sub-bloques, Xj, de 32 bits cada uno.

En el módulo iterativo, la rotación conjunta de todos los bits, la primera operación, se realiza sobre un registro de longitud igual a 128 bits siendo sus bits de control los 7 bits menos significativos de la palabra de clave $Z_1^{(i)}$,

La figura 4 corresponde a la estructura suma-rotación de esta implementación. Está compuesta por dos columnas, C1 y C2, que contienen cada una de ellas siete registros rotadores y seis sumadores. Los registros rotadores, de 32 bits, han sido divididos en dos partes de la forma p = 31 y w - p = 1. La parte del registro en la que queda un único bit no sufre ninguna modificación mientras que los 31 bits restantes rotan según el contenido de los bits procedentes de P1 o Q2 según se ve en la figura 4. De los siete registros rotadores, los cuatro primeros reciben 5 bits para controlar su rotación mientras que a los tres últimos sólo llegan 4 bits; en total, 5 . 4 + 4 . 3 = 32 bits de control, es decir, la totalidad de los bits de cada palabra interviene en el control de rotaciones.

El número de palabras de clave $Z_t^{(i)}$, de 32 bits, que son necesarias en cada iteración es igual a t = 6.2 + 1.

Finalmente, se realiza la transformación de salida que comienza con una rotación de 128 bits, controlada por los 7 bits menos significativos de la palabra de clave $Z_1^{(v+1)}$. Una vez realizadas las operaciones restantes, se obtiene un bloque cifrado, Y, de 128 bits.

Para el descifrado basta con realizar las mismas operaciones teniendo en cuenta la permutación de algunas palabras de clave según se indica en la Tabla 1.

Funcionamiento y seguridad

El esquema anterior del algoritmo de cifrado más un algoritmo de expansión de clave a partir de la clave de usuario han sido implementados como programa informático en un ordenado con microprocesador Pentium$_{TM}$-130 Mhz. de Intel, trabajando bajo el sistema operativo Windows® 95 y compilado con Visual C++$_{TM}$ 4.0 de Microsoft. Bajo este entorno, ejecutando el algoritmo de cifrado con v = 4 iteraciones y con una clave de usuario de 128 bits, se obtiene una velocidad de cifrado de 3,22 Mbits/s. aproximadamente. Obviamente, esta velocidad cambiará según varíe el número de iteraciones y mejorará drásticamente en una implementación hardware.

Los resultados obtenidos en los tests estadísticos realizados muestran un comportamiento prácticamente ideal tanto en la distribución de unos y ceros como en las correlaciones. En las figuras 5a y 5b aparecen los resultados de estos tests aplicados a un fichero de texto en claro de longitud igual a 512 bytes y el valor de todos sus datos igual a 000...0 cifrado con una clave de usuario de 128 bits todos iguales a 000...0 y para una única iteración.

En la siguiente tabla aparece un ejemplo que corrobora el efecto de difusión conseguido con el algoritmo propuesto cuando sólo varía un bit del texto en claro o un bit de la clave de usuario. Se ha realizado la simulación para una clave de usuario de 64 bits y una única iteración. En ambos casos, el texto cifrado presenta una variación de sus bits muy cercana al 50% respecto al primer texto cifrado.

TABLA 3

Estudio de la variación de bits entre textos cifrados cuando se cambia un bit del texto en claro o un bit de la clave de usuario

Texto en Claro	Clave de Usuario	Texto Cifrado	Diferencia
0000 0000 0000 0000 0000 0000 0000 0000	0000 0000 0000 0000	6F06 02DF 74B0 117F 8372 49E9 72C2 F0CE	
0000 0000 0000 0000 0000 0000 1000 0000	0000 0000 0000 0000	759A D74E 7166 4EA2 9D9F 145A 7698 B79B	61 bits 47.65%
0000 0000 0000 0000 0000 0000 0000 0000	0000 0000 0001 0000	B915 06A7 C751 6324 D905 4DB1 2061 A315	66 bits 51.5%

Aplicaciones de la invención

El cifrador en bloque transforma unos datos de entrada según una clave dada en otros datos de salida cuya relación con los datos originales y la clave ha sido totalmente oscurecida, de manera que sin la clave resulta totalmente imposible recuperar los datos de entrada. Los datos de salida presentan una distribución estadística que los convierte en una secuencia pseudo-aleatoria. Por todo ello, este algoritmo puede usarse como:

· Algoritmo simétrico de cifrado en bloque de datos para criptografía de clave privada: permite

cifrar/descifrar datos con una clave secreta, de manera que sea imposible recuperar el mensaje original si no se posee la clave secreta.

- Código de autenticación de mensajes: puede ser empleado como código de autenticación de mensajes, o CAM, operando en cualquiera de los modos de libro electrónico de códigos, realimentación de la salida o realimentación del texto cifrado.

- Resumen de mensajes: puede ser empleado como función de hash para resumir mensajes, sin más que hacer pública la clave secreta cuando opera como CAM.

- Firma digital: puede ser empleado como algoritmo de cifrado para firmar documentos.

- Generador de secuencias pseudo-aleatorias: en los modos de operación del algoritmo como realimentación de la salida o realimentación del texto cifrado, puede ser también utilizado como generador de secuencias pseudo-aleatorias, ya que la distribución de unos y ceros obtenida a la salida satisface los requisitos que deben verificar estos generadores.

REIVINDICACIONES

1. Método y aparato para cifrado en bloque de datos, constituido por un sistema electrónico que implementa en hardware o software un algoritmo de cifrado con clave secreta que actúa sobre bloques de datos, de una longitud predeterminada por el usuario por medio de una función iterada el número de veces que el usuario haya seleccionado y siendo la salida de cada iteración, o vuelta de cifrado, función de la salida de la vuelta anterior y de varios sub-bloques derivados de la clave secreta del usuario por medio de un algoritmo de expansión de clave.

2. Un algoritmo de cifrado/descifrado, según reivindicación 1, **caracterizado** por el uso de tres operaciones: la primera es una rotación cíclica sobre n bits, la segunda es la suma del grupo cíclico de los enteros módulo 2^w,(Z_{2^w},+), interpretada en términos booleanos, y la tercera es la suma del grupo cíclico de los enteros módulo 2, (Z_2,+).

3. Un algoritmo de cifrado/descifrado, según reivindicaciones 1 y 2, **caracterizado** porque la combinación de dos operaciones diferentes de cualesquiera de las tres operaciones definidas en la reivindicación 2 no satisfaga una ley distributiva o conmutativa.

4. Un algoritmo de cifrado/descifrado, según reivindicaciones 1, 2 y 3, **caracterizado** porque se usan rotadores cuyo número de rotaciones es variable y dependiente tanto de los datos en claro como de la clave.

5. Un algoritmo de cifrado/descifrado, según reivindicaciones 1, 2, 3 y 4, **caracterizado** por el uso de una estructura de suma-rotación integrada por dos columnas, que contienen cada una de ellas varios sumadores y rotadores, en los que el número de posiciones rotadas está determinado por la otra columna.

6. Un algoritmo de cifrado/descifrado, según reivindicaciones 1, 2, 3, 4 y 5, **caracterizado** porque su estructura de suma-rotación asegura con un 100% de probabilidad que *todos* los bits de los datos en claro y de la clave determinarán *al menos una* rotación, lográndose así una difusión completa con una sola vuelta de cifrado.

7. Un algoritmo de cifrado/descifrado, según reivindicaciones 1, 2, 3, 4, 5 y 6, **caracterizado** porque los rotadores utilizados en la estructura suma-rotación tienen una longitud igual a un número primo.

8. Un algoritmo de cifrado/descifrado, según reivindicaciones 1, 2, 3, 4, 5, 6 y 7, **caracterizado** porque la palabra de w bits a rotar se puede dividir en dos o más partes de longitudes w_1, w_2, ..., w_r bits, siendo w_i, $1 \leq i \leq r$, números enteros primos.

9. Un algoritmo de cifrado/descifrado, según reivindicaciones 1, 2, 3, 4, 5, 6, 7 y 8, **caracterizado** porque se pueden rotar a voluntad del usuario cada una de las partes w_i, $1 \leq i \leq r$, en que se haya dividido la palabra total de w bits.

10. Un algoritmo de cifrado/descifrado, según reivindicaciones 1, 2, 3, 4, 5, 6, 7, 8 y 9, **caracterizado** por el uso de un rotador de longitud 4w como primera operación de cada iteración y también como primera operación de la transformación de salida.

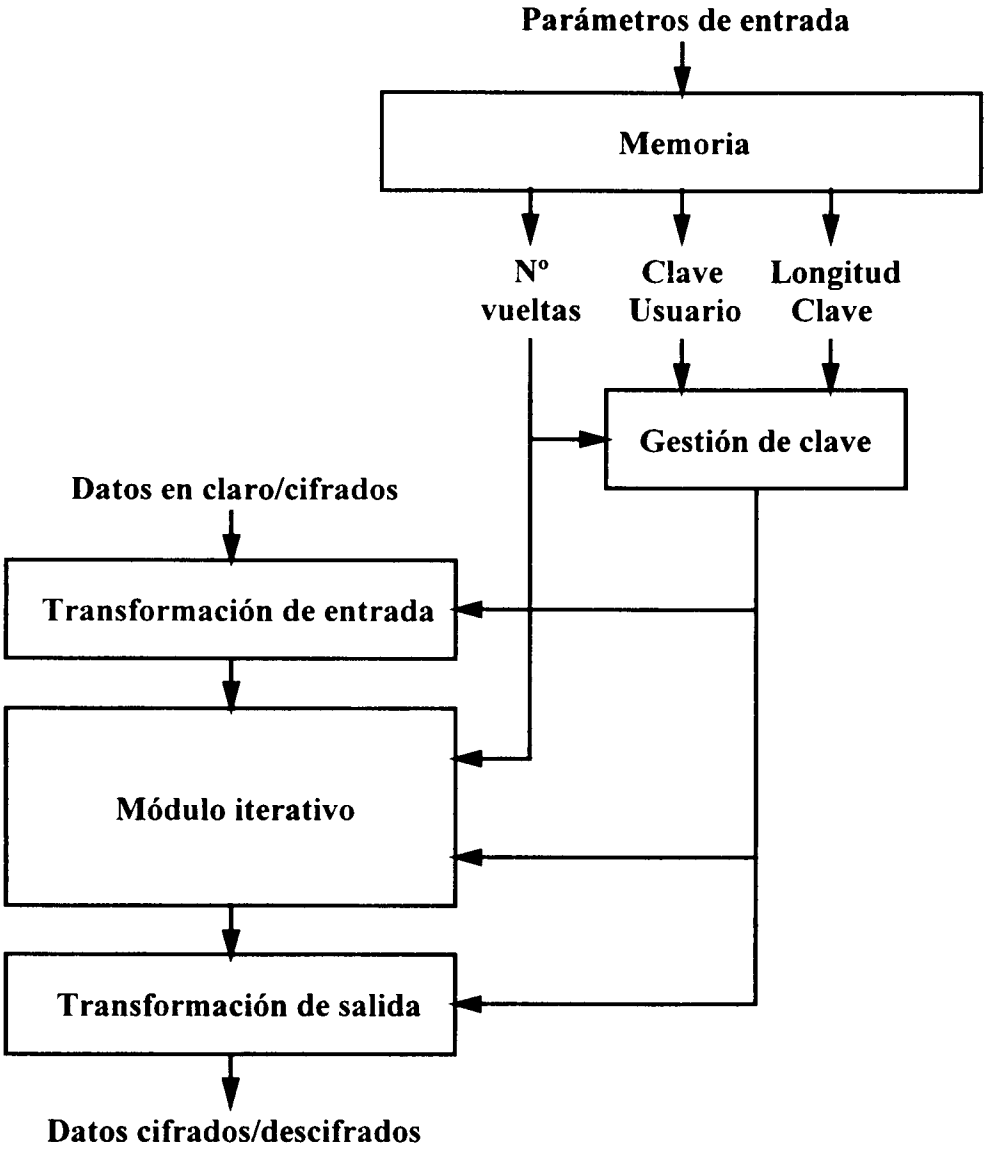

Parámetros de entrada
Memoria
Nº vueltas
Clave Usuario
Longitud Clave
Gestión de clave
Datos en claro/cifrados
Transformación de entrada
Módulo iterativo
Transformación de salida
Datos cifrados/descifrados

Figura 1

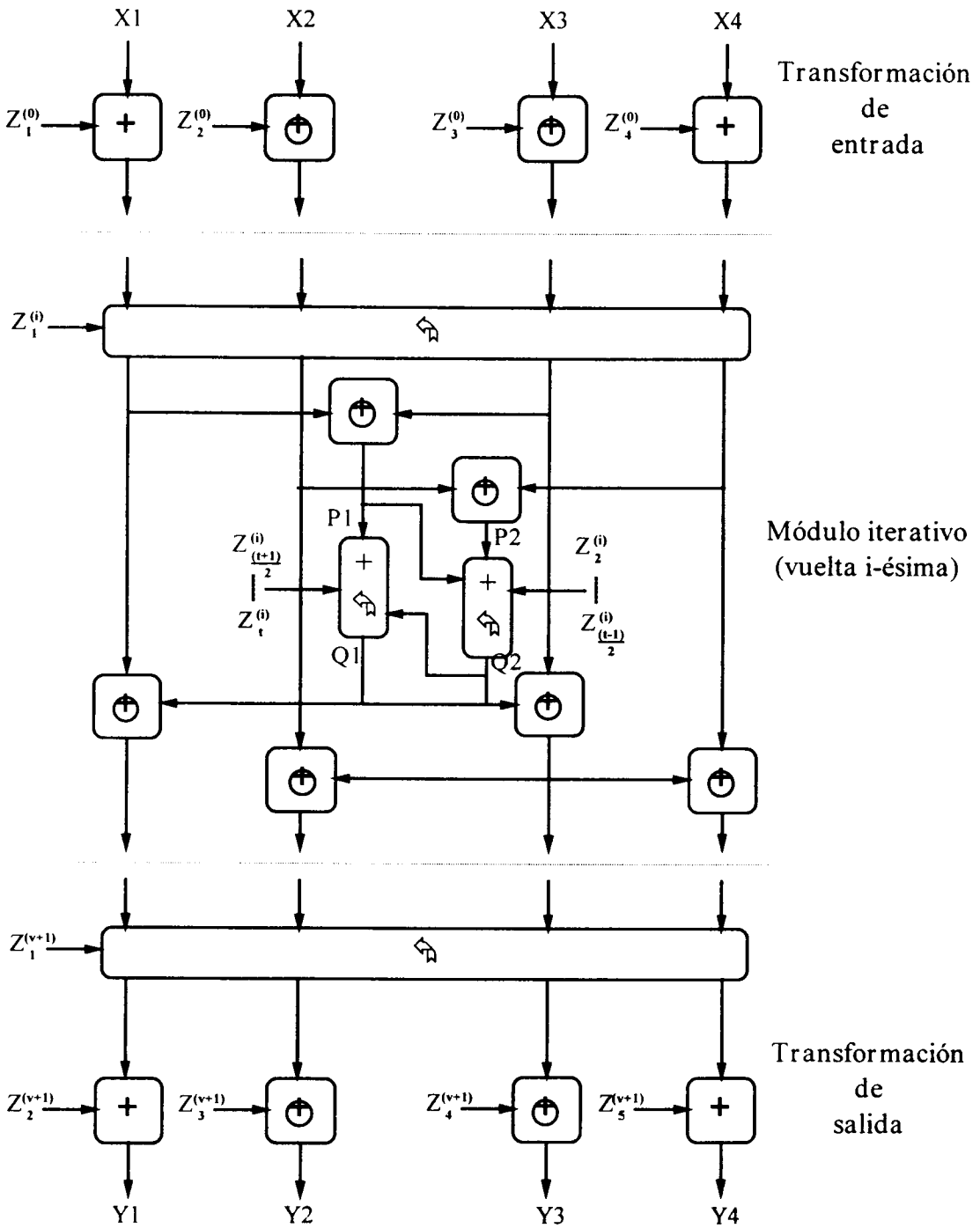

X1
X2
X3
X4
Transformación de entrada
Módulo iterativo (vuelta i-ésima)
P1
P2
Q1
Q2
Transformación de salida
Y1
Y2
Y3
Y4

Figura 2

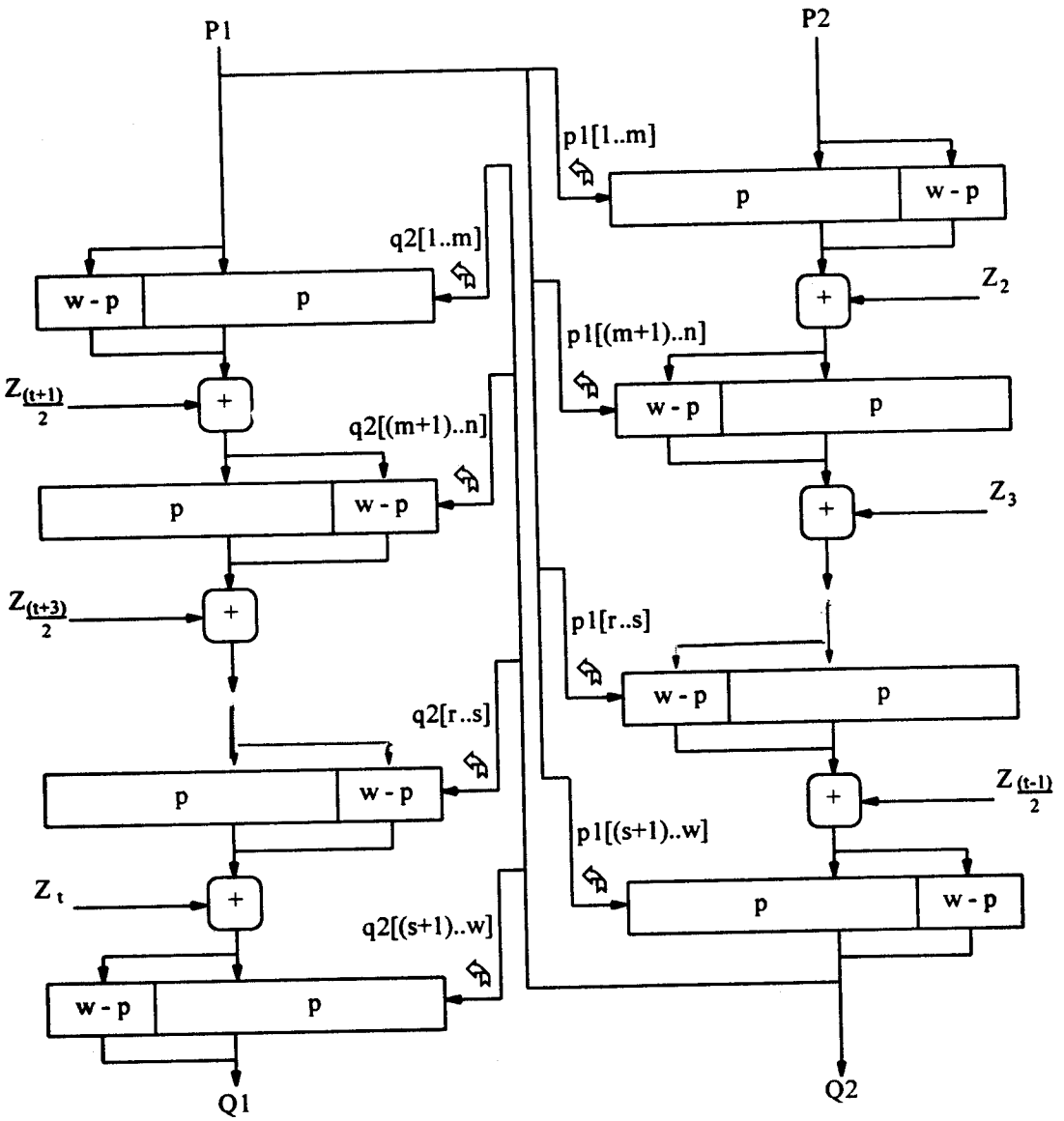

P1
P2
p1[1..m]
q2[1..m]
w - p
p
Z2
p1[(m+1)..n]
Z(t+1)/2
q2[(m+1)..n]
Z3
Z(t+3)/2
p1[r..s]
q2[r..s]
Z(t-1)/2
p1[(s+1)..w]
Zt
q2[(s+1)..w]
Q1
Q2

Figura 3

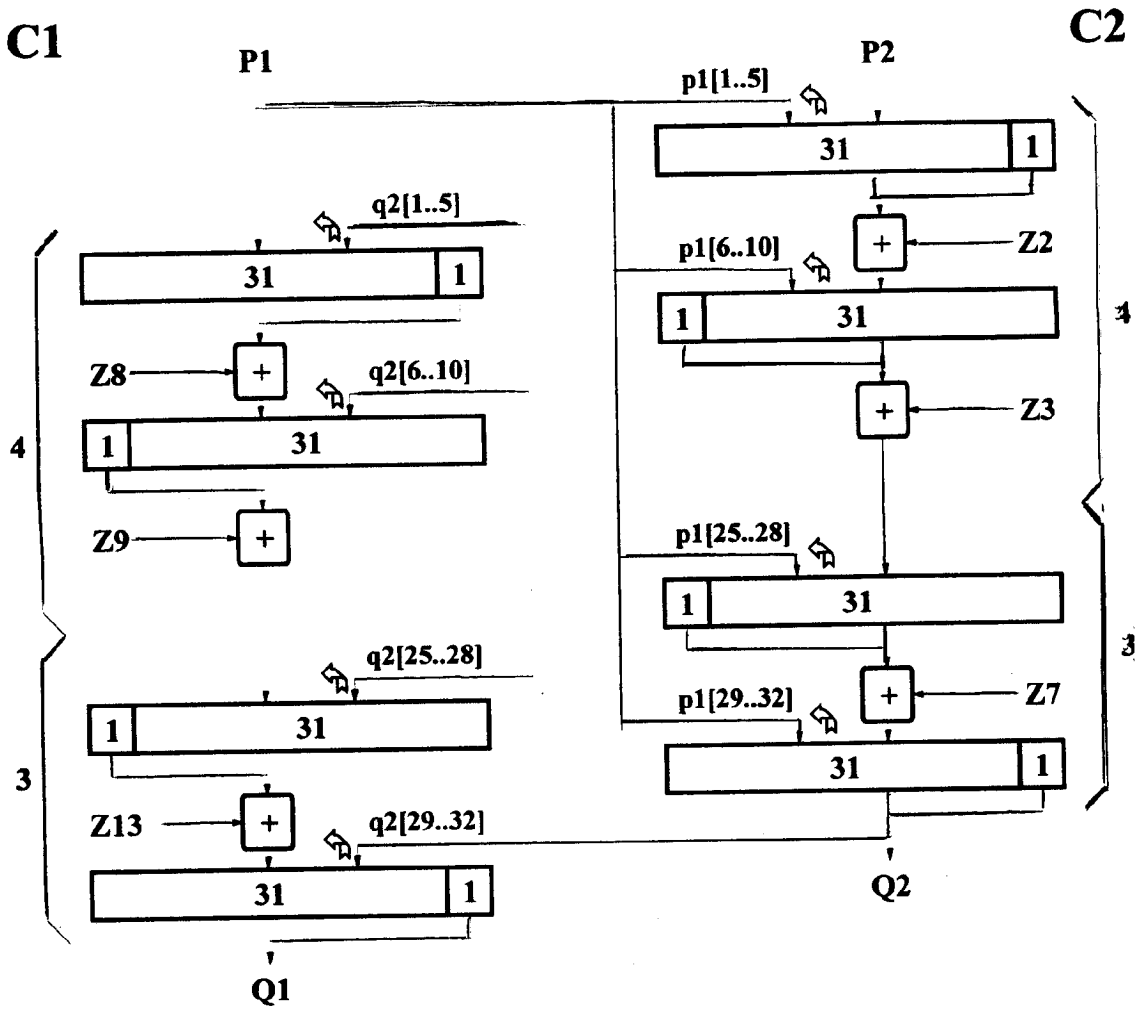

C1
C2
P1
P2
p1[1..5]
31
1
q2[1..5]
Z2
p1[6..10]
Z8
q2[6..10]
Z3
4
Z9
p1[25..28]
q2[25..28]
Z7
p1[29..32]
3
Z13
q2[29..32]
Q2
Q1

Figura 4

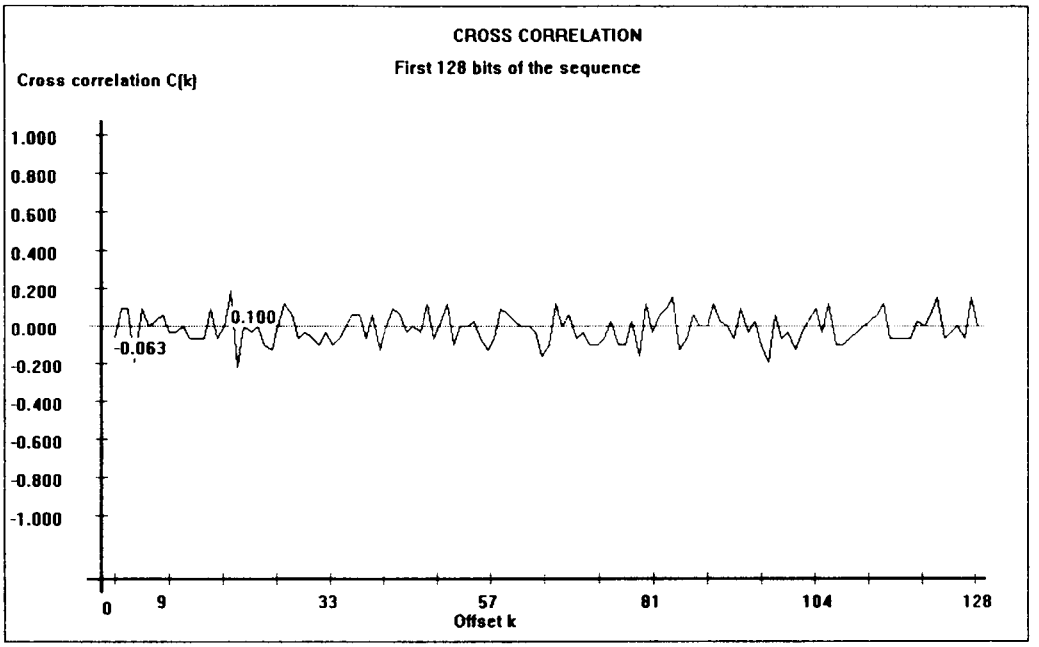


Figura 5a.

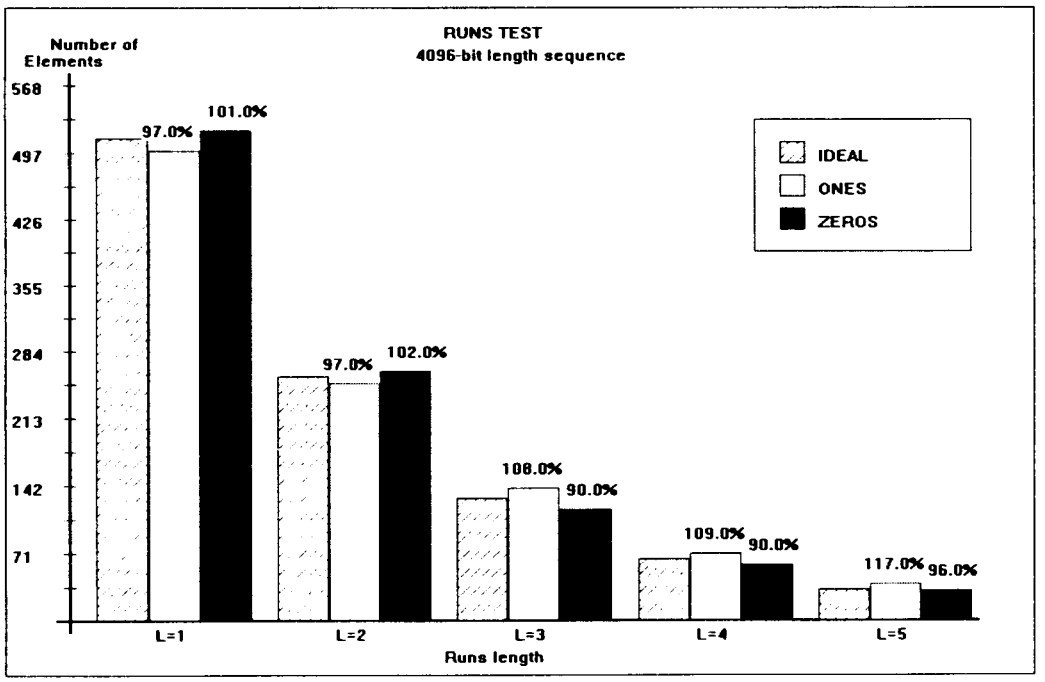


Figura 5b.

OFICINA ESPAÑOLA
DE PATENTES Y MARCAS

ESPAÑA

(11) ES 2 123 443

(21) N.° solicitud: 9602747

(22) Fecha de presentación de la solicitud: **27.12.96**

(32) Fecha de prioridad:

INFORME SOBRE EL ESTADO DE LA TECNICA

(51) **Int. Cl.**[6]**:** H04L 9/06

DOCUMENTOS RELEVANTES

Categoría	Documentos citados	Reivindicaciones afectadas
E	US 5724428 A (RIVEST) 03.03.1998, columna 3, línea 13 - columna 6, línea 49.	1,2
A	EP 0443752 A2 (GENERAL INSTRUMENT CORPORATION) 28.08.1991, todo el documento.	1,2
A	US 4078153 A (MOREAU) 07.03.1978, columna 5, línea 25 - columna 6, línea 22.	1,2

Categoría de los documentos citados

X: de particular relevancia

Y: de particular relevancia combinado con otro/s de la misma categoría

A: refleja el estado de la técnica

O: referido a divulgación no escrita

P: publicado entre la fecha de prioridad y la de presentación de la solicitud

E: documento anterior, pero publicado después de la fecha de presentación de la solicitud

El presente informe ha sido realizado

[×] para todas las reivindicaciones

[ ] para las reivindicaciones n°:

Fecha de realización del informe	Examinador	Página
18.11.98	M. Alvarez Moreno	1/1