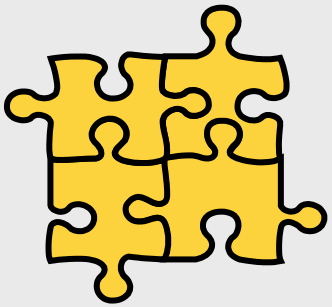
# Procedimientos de Cifrado en flujo

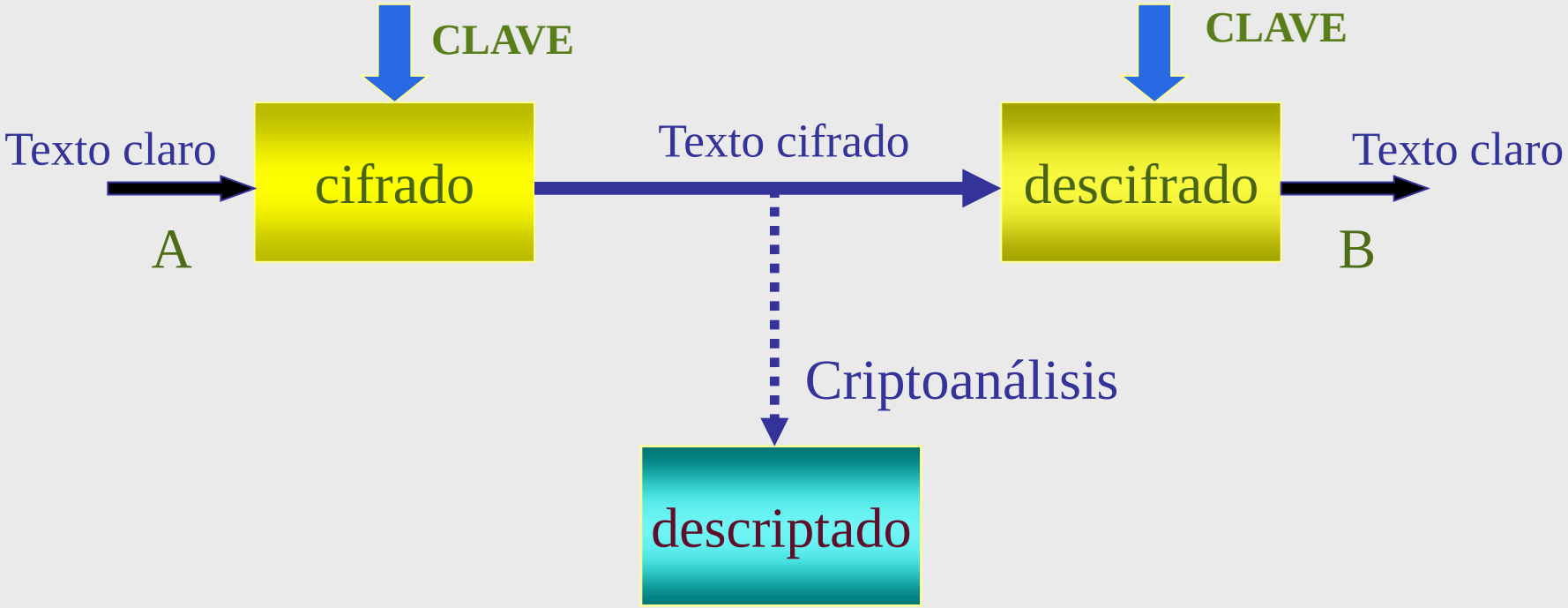
Amparo Fúster Sabater
Instituto de Física Aplicada C.S.I.C.
amparo@iec.csic.es

## Contenido

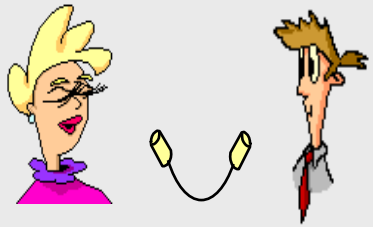


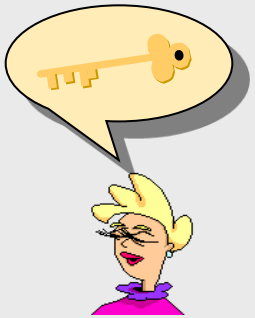
- Operaciones binarias
- Del cifrado Vernam al cifrado en flujo
- Secuencias cifrantes
- Generadores de secuencias cifrantes

# Procedimiento Criptográfico: Esquema General



# Criptografía Actual





# Operaciones binarias

- Suma módulo 2 → $\oplus$

| $x_1$ | $x_2$ | $\oplus$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Operación **XOR**

$$F(x_1, x_2) = x_1 \oplus x_2$$

- Producto lógico → $\cdot$

| $x_1$ | $x_2$ | $\cdot$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Operación **AND**

$$F(x_1, x_2) = x_1 x_2$$

# Fundamentos matemáticos

- Función Booleana:

$$F(x_1, x_2, \ldots, x_n) \longrightarrow \text{variables binarias}$$

- Función lineal: **no** adecuadas para criptografía

$$F(x_1, x_2, \ldots, x_n) = x_1 \oplus x_2 \oplus x_n$$

- Función no lineal: Forma algebraica normal (A.N.F.)

$$F(x_1, x_2, \ldots, x_n) = x_1 x_2 \oplus x_3 x_{n-1} x_n$$

Orden no lineal: producto con mayor número de variables

# Cifrado Vernam (1917)

- **One-time Pad** (Cifrado con cinta aleatoria)

  Clave: ***Secuencia binaria aleatoria de uso único***

- **Cifrado:**

$$Y_i = X_i \oplus Z_i \mod 2$$

- **Descifrado:**

$$X_i = Y_i \oplus Z_i = (X_i \oplus Z_i) \oplus Z_i \mod 2$$

Mensaje: come soon (Código ITA-2)

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Mensaje X** | 00011 | 01111 | 01101 | 00101 | 10011 | 01111 | 01111 | 01110 |
| **Clave Z** | 11011 | 00101 | 01011 | 00110 | 10110 | 10101 | 01100 | 10010 |
| **Criptograma Y** | 11000 | 01010 | 00110 | 00011 | 00101 | 11010 | 00011 | 11100 |

# Condiciones de secreto perfecto (Shannon)

- Condiciones de aplicación:
  - La clave se utiliza sólo una vez
  - El criptoanalista tiene acceso sólo al criptograma

- **Secreto perfecto:**

  "El texto claro X es estadísticamente independiente del criptograma Y para todos los posibles textos claros y todos los posibles criptogramas"

$$P(X = x \mid Y = y) = P(X = x)$$

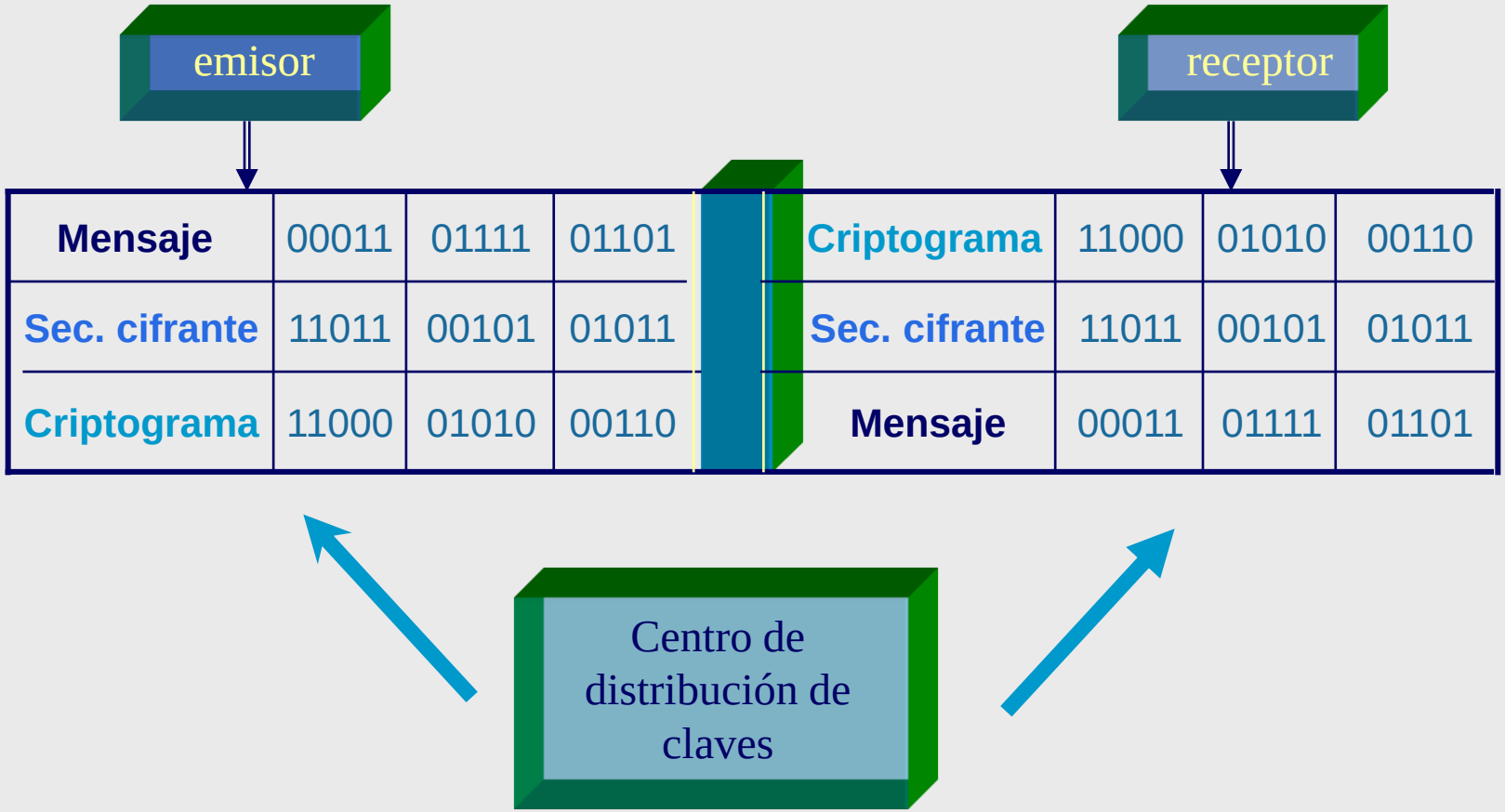
C.E. Shannon, "Communication Theory of Secrecy Systems", Bell. Syst. Tech. J., 28 (1949), 656-715.

- Existen cifradores perfectos?
  - Cifrador con X, Y, Z $\in\{0,1,\ldots,L-1\}$
  - La clave se elige de forma aleatoria
  - Transformación de cifrado: $y_i = x_i \oplus z_i \bmod L$

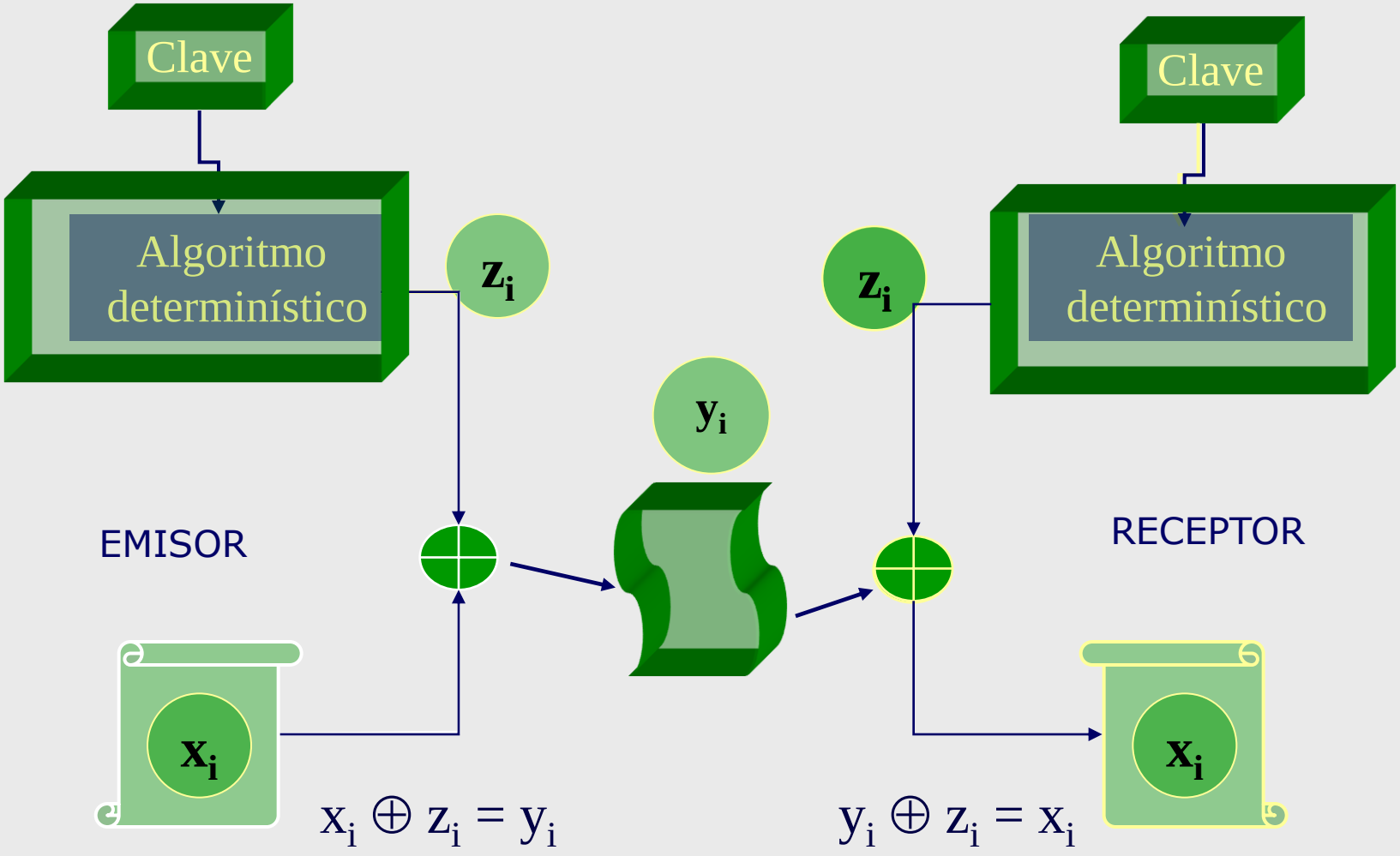
- Los cifradores aditivos módulo L verifican las condiciones de secreto perfecto

- **Conclusión:** La clave debe ser aleatoria, usarse una única vez y tener al menos igual longitud que el mensaje original

# Cifrado Vernam: es viable?

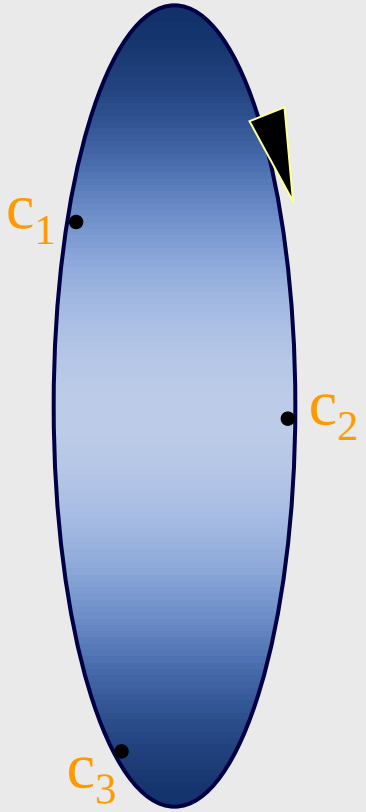


# Cifrado en Flujo



# Cifrado en flujo

- Cifrado bit a bit
- Sec. cifrante = Algoritmo + clave
- Generación de secuencias cifrantes
  - Periodo largo
  - Propiedades de **pseudoaleatoriedad**
  - Imprevisibilidad
  - Espacio claves
  - Facilidad de implementación
- No estamos propiamente en condiciones de secreto perfecto
  (Aproximación al cifrado Vernam)

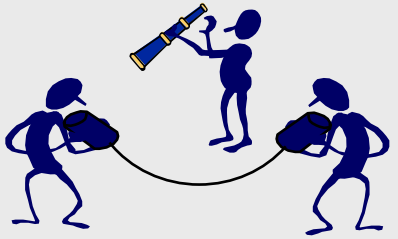


# Secuencia Cifrante:

1. ¿Qué características generales presentan estas secuencias?

2. ¿Cómo son los generadores que las producen?

# Secuencias Cifrantes: Características Generales (I)



- Periodo ($\approx 10^{38}$ bits)

  $T = 2^{128} - 1 \approx 3{,}40 \times 10^{38}$ bits

  $V_c = 120 \times 10^6$ bits/seg $\Rightarrow 9.452 \times 10^{22}$ años

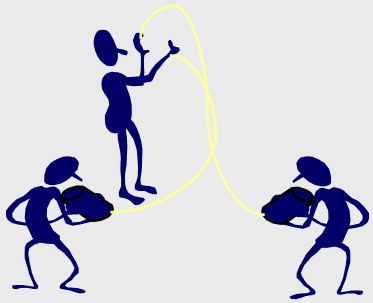
22.200 veces la edad del universo

# Secuencias Cifrantes: Características Generales (II)

- Distribución de ceros y unos

  01001101**00**1110**1**0110010010

  - rachas de ceros (gaps)
  - rachas de unos (blocks)



# Secuencias Cifrantes: Características Generales (III)

- Autocorrelación

$$AC(k) = (A - D)/T$$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sec.original | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| Sec. desplazada | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

- *A* = Número de coincidencias
- *D* = Número de no coincidencias
- *T* = periodo de la secuencia
- *k* = desplazamiento

$$AC(4) = (6 - 10)/16$$

# Secuencias Cifrantes: Características Generales (III)

- Autocorrelación

$$AC(k) = (A - D)/T$$

| Sec.original | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sec. desplazada | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |

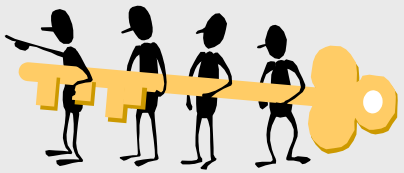
  - Autocorrelación en fase: $AC(k) = 1$
  - Autocorrelación fuera de fase: $AC(k) \in [-1, 1]$

# Postulados de pseudoaleatoriedad de Golomb

- G1: En cada periodo de la secuencia considerada el número de 1's tiene que ser aproximadamente igual al número de 0's.

- G2: En cada periodo de la secuencia considerada la mitad de las rachas, del número total de rachas observadas, tiene longitud 1, la cuarta parte longitud 2, la octava parte longitud 3 … etc. Para cada longitud habrá el mismo número de blocks que de gaps.

- G3: La autocorrelación *AC(k)* fuera de fase tiene que ser constante para todo valor de *k*.

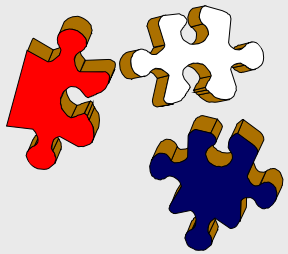


# Interpretación de los Postulados de Golomb

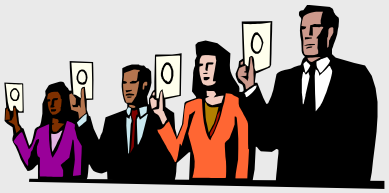
- G1: El número de 1's y 0's tiene que aparecer a lo largo de la secuencia con la misma probabilidad.

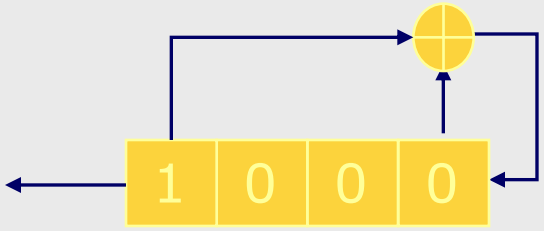
- G2: En cada periodo de la secuencia diferentes *n*-gramas (muestras de *n* dígitos consecutivos) deben aparecer con la probabilidad correcta .

- G3: El cálculo de coincidencias entre una secuencia y su versión desplazada no debe aportar información sobre el periodo de la secuencia.



# Registros de desplazamiento realimentados linealmente (LFSR)



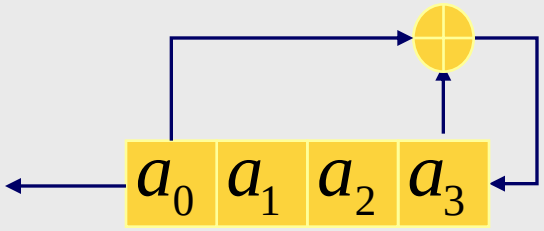



$\leftrightarrow$ $x^4 + x + 1$

- Parámetros del registro
  - Longitud ***L***
  - Pol. de realimentación ó característico
- Funcionamiento
  - Desplazamiento del contenido binario
  - Entra bit realimentado
- Secuencia generada: **1 0 0 0 1 1 1 1 .....**

| | | | |
|---|---|---|---|
| • | | | • |
| 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| • | • | •• | • |

# Linear Feedback Shift Registers




$$\leftrightarrow \quad x^4 + x + 1$$

- Relación de recurrencia **lineal** $P_c(x) = x^4 + x^3 + 1$

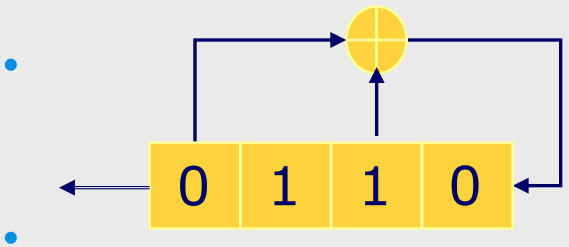
$$a_{n+4} = a_{n+3} \oplus a_n \qquad n \geq 0$$

$$a_4 = a_3 \oplus a_0$$
$$a_5 = a_4 \oplus a_1$$
$$\vdots \qquad \vdots$$

- Cada elemento se expresa en función de los anteriores (relación **lineal**)

| | | | |
|---|---|---|---|
| • | | | • |
| 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| • | • | • | • |

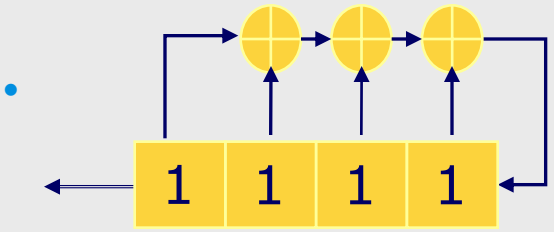
# Tipos de LFSRs




$\leftrightarrow$ $x^4 + x^2 + 1$

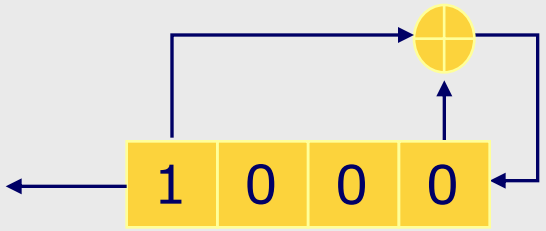
Estado inicial

- 0110
- 0001
- 0011




$\leftrightarrow$ $x^4 + x^3 + x^2 + x + 1$

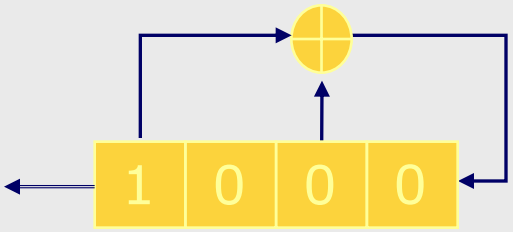
- 1111
- 0001
- 0010



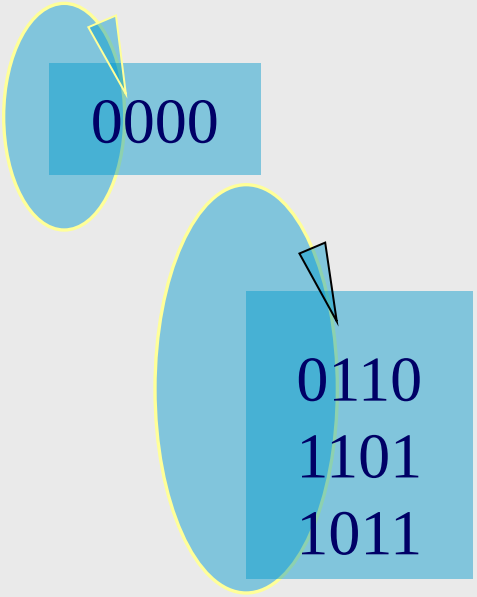

$\leftrightarrow$ $x^4 + x + 1$

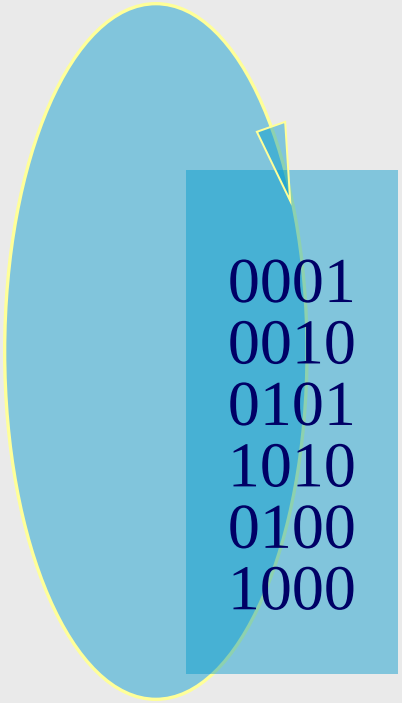
- 1000

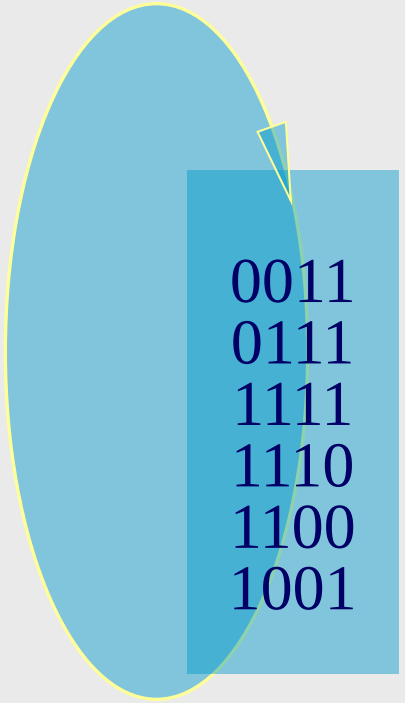
# Periodo del LFSR (factorizable)




$$x^4 + x^2 + 1 = (x^2 + x + 1)(x^2 + x + 1)$$



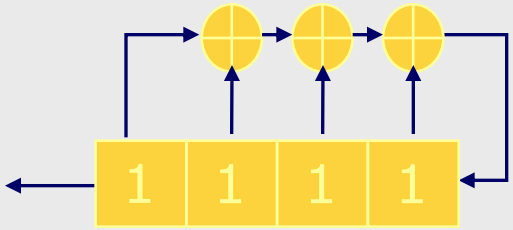



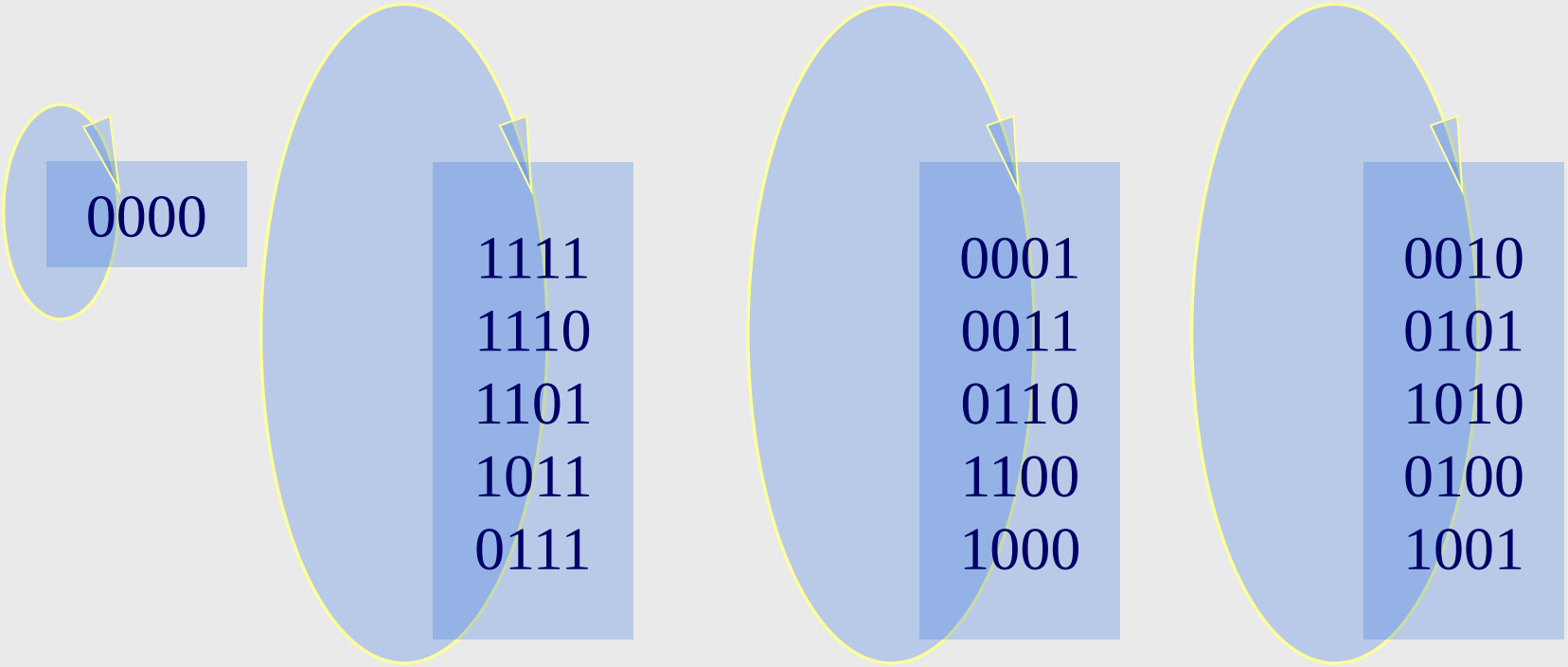
# Generadores con polinomio de realimentación **factorizable**

- La longitud de la secuencia depende del estado inicial
- El periodo *T* verifica $L \leq T < 2^L - 1$ pudiendo aparecer periodos secundarios que son divisores del periodo *T*
- No adecuados para usos criptográficos
- Golomb, S. W., *Shift Register Sequences,* Holden Day, San Francisco, 1967

# Periodo del LFSR (irreducible)



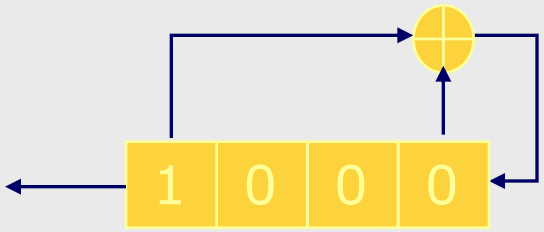

$$x^4 + x^3 + x^2 + x + 1$$



# Generadores con polinomio de realimentación **irreducible**

- La longitud de la secuencia no depende del estado inicial
- El periodo *T* es un factor de $2^L - 1$
- No adecuados para usos criptográficos

# Periodo del LFSR (primitivo)




$x^4 + x + 1$

0000

1000
0001
0011
0111
1111
1110
1101
1010
0101
1011
0110
1100
1001
0010
0100

- ***PN-secuencia (m-secuencia)***
- Máximo periodo posible para este tipo de generador

100011110101100 …..

# Generadores con polinomio de realimentación **primitivo**

- La longitud de la secuencia **no** depende del estado inicial
- El período es $T = 2^L - 1$
- Adecuados para usos criptográficos, su secuencia satisface los postulados de Golomb

- ¿Cuántos polinomios primitivos de grado ***L*** hay?

$$\Phi(2^L - 1)/L$$

$L = 11 \qquad No. = 176$

$L = 24 \qquad No. = 276480$

- Son todos buenos?
  - $x^{15} + x + 1 \Rightarrow a_{n+15} = a_{n+1} \oplus a_n$

$$x^{15} + x^{14} + x^{13} + x^{12} + x^{11} + x^5 + x^4 + x^3 + 1 \Rightarrow$$

$$a_{n+15} = a_{n+14} \oplus a_{n+13} \oplus a_{n+12} \oplus a_{n+11} \oplus a_{n+5} \oplus a_{n+4} \oplus a_{n+3} \oplus a_n$$

  - Distribución uniforme de etapas

# PN-secuencias y postulados de Golomb

- G1:

$$T = 2^L - 1$$

$$\begin{cases} No.1's = 2^{L-1} \\ No.0's = 2^{L-1} - 1 \end{cases}$$

- G2:

| Long. | Gaps | Blocks |
|---|---|---|
| 1 | $2^{L-3}$ | $2^{L-3}$ |
| 2 | $2^{L-4}$ | $2^{L-4}$ |
| : | : | : |
| r | $2^{L-r-2}$ | $2^{L-r-2}$ |
| : | : | : |
| *L-2* | 1 | 1 |
| *L-1* | 1 | 0 |
| *L* | 0 | 1 |
| Total | $2^{L-2}$ | $2^{L-2}$ |

PN-secuencia

100111110101100

# PN-secuencias y postulados de Golomb

- G3:

$$\begin{array}{l} 1\,0\,0\,0\,1\,1\,1\,1\,0\,1\,0\,1\,1\,0\,0 \\ \oplus\ 0\,0\,0\,1\,1\,1\,1\,0\,1\,0\,1\,1\,0\,0\,1 \\ \hline 1\,0\,0\,1\,0\,0\,0\,1\,1\,1\,1\,0\,1\,0\,1 \end{array} \qquad AC(1) = (7-8)/15$$

$$\begin{array}{l} 1\,0\,0\,0\,1\,1\,1\,1\,0\,1\,0\,1\,1\,0\,0 \\ \oplus\ 0\,0\,1\,1\,1\,1\,0\,1\,0\,1\,1\,0\,0\,1\,0 \\ \hline 1\,0\,1\,1\,0\,0\,1\,0\,0\,0\,1\,1\,1\,1\,0 \end{array} \qquad AC(2) = (7-8)/15$$

Las PN-secuencias verifican los postulados de Golomb

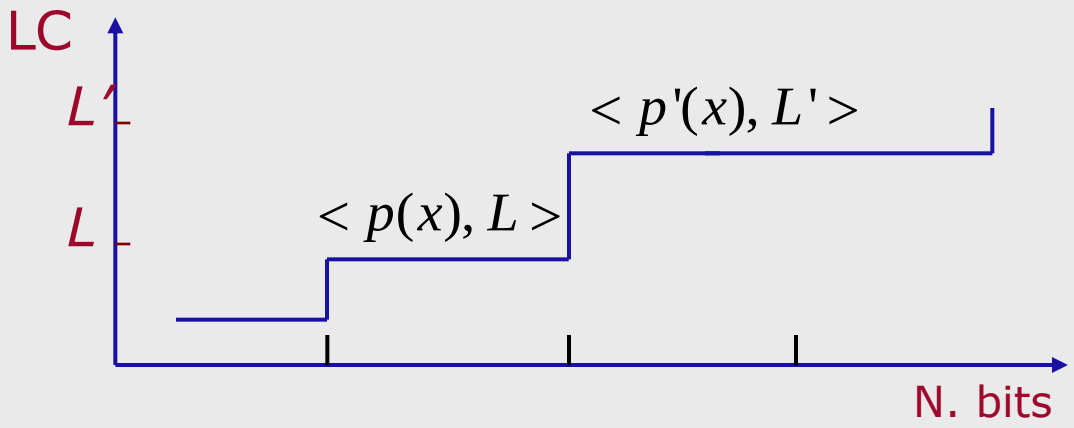
# Complejidad Lineal: imprevisibilidad

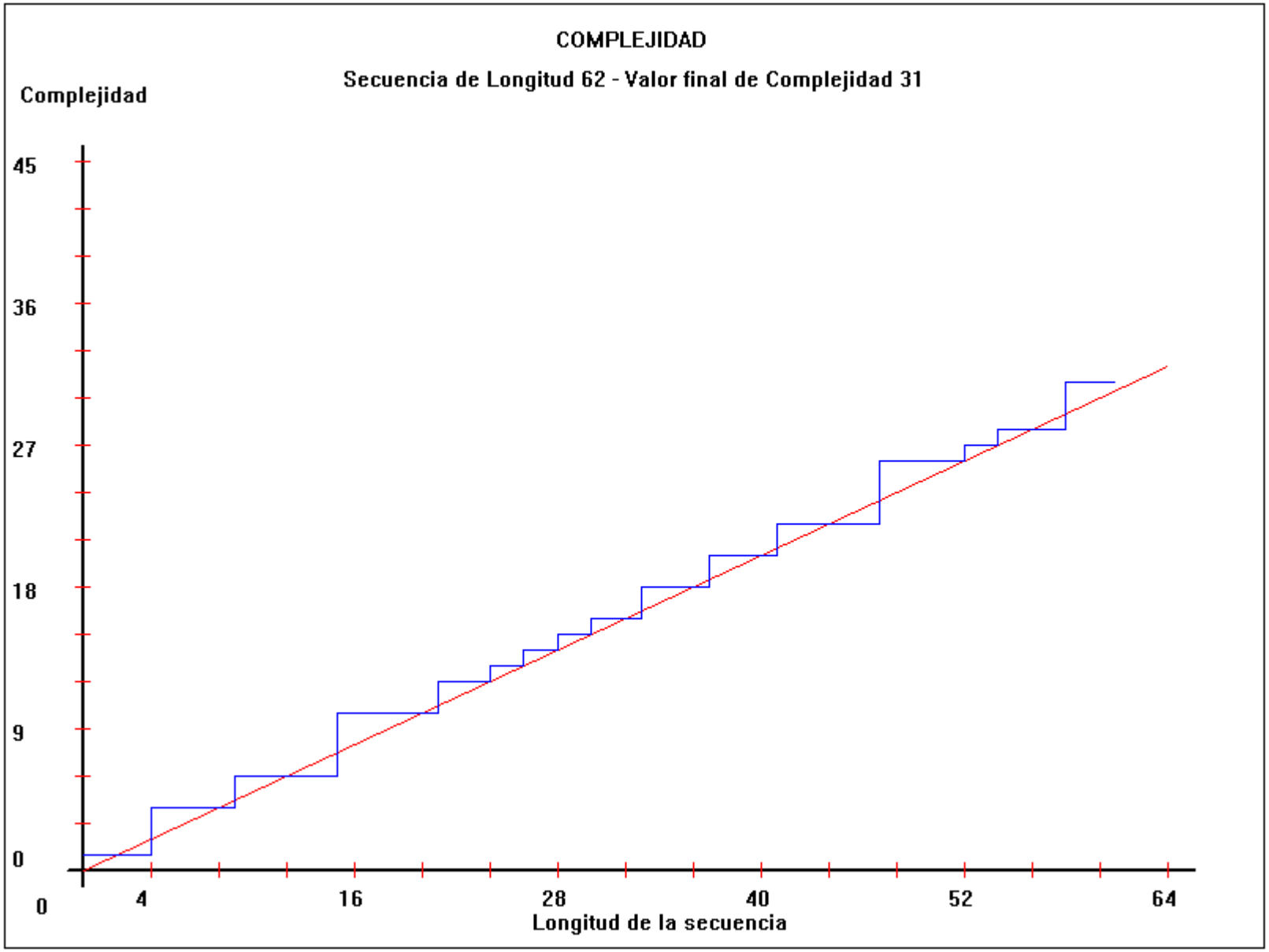
- Concepto básico: cantidad de secuencia que hay que conocer para determinar el resto de la misma
- Idea general: Asociar a cada secuencia un LFSR
- Complejidad Lineal = La longitud del menor LFSR capaz de generarla
- Algoritmo de Massey-Berlekamp (1969)
  - Input: La secuencia binaria considerada
  - Output: $< P(x),\ LC >$

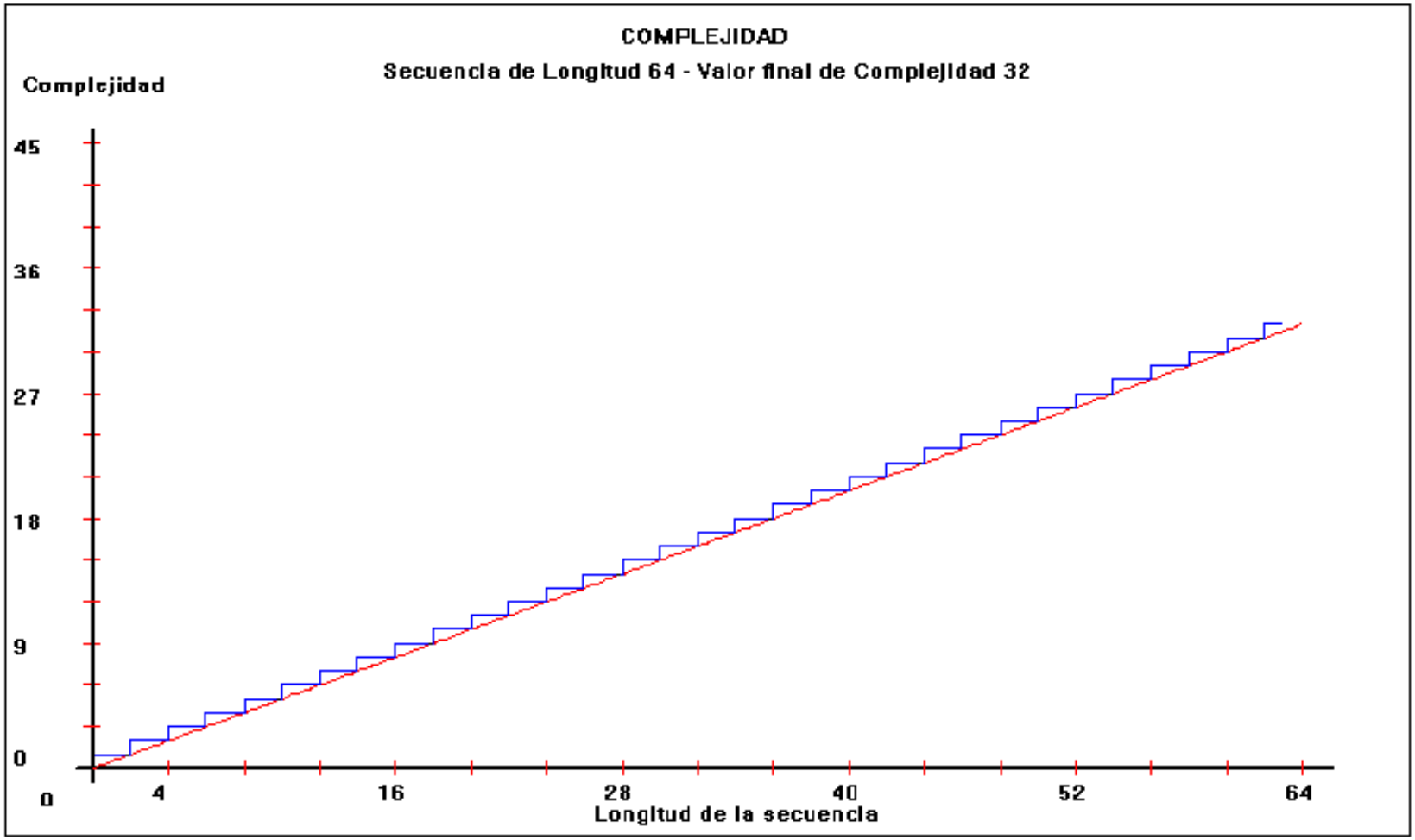
# Algoritmo de Berlekamp-Massey

- Idea general:
  - $S^n \equiv a_0, a_1, \ldots, a_{n-1} \rightarrow < p(x), L >$
  - $S^{n+1} \equiv a_0, a_1, \ldots, a_{n-1}, a_n \rightarrow < p'(x), L' >$

- Curva de complejidad




COMPLEJIDAD
Secuencia de Longitud 62 - Valor final de Complejidad 31
Complejidad
45
36
27
18
9
0
0
4
16
28
40
52
64
Longitud de la secuencia


COMPLEJIDAD
Secuencia de Longitud 64 - Valor final de Complejidad 32
Complejidad
45
36
27
18
9
0
0
4
16
28
40
52
64
Longitud de la secuencia

# Complejidad Lineal (II)

- Secuencia 1: $T = 2^{127} - 1 \approx 10^{38}\ bits$

Sec. generada por un LFSR (pol. Primitivo)

**MUY PREVISIBLE**

$< P(x), LC >$ $\quad LC = 127, \quad 2LC = 256\ bits$

- Secuencia 2:

10001111010000110111110100010100

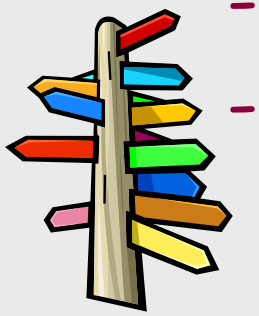
$T = 31\ bits$ **MUY IMPREVISIBLE**

$< P(x), LC >$ $\quad LC = 15, \quad 2LC = 30\ bits$

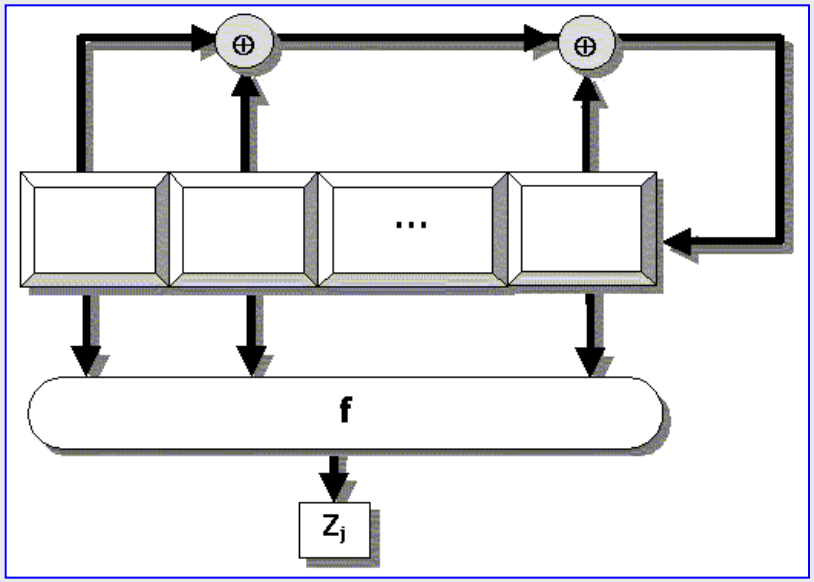
- **Relación entre *T y 2·LC***

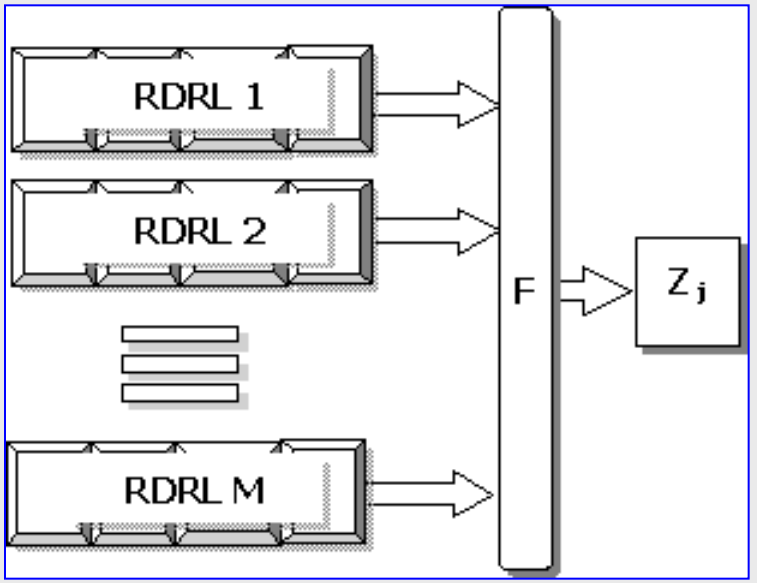
# Criptosistemas de cifrado en flujo

- Generadores de secuencia binaria
- Basados en Registros (LFSR)
  - Preservar las características de las PN-sec.
  - Incrementar su complejidad
- Clave: contenido inicial de estos registros
- Hipótesis de partida:
  - Conocimiento del esquema de generación
  - Conocimiento de una cantidad de la secuencia de salida



# Generadores combinacionales

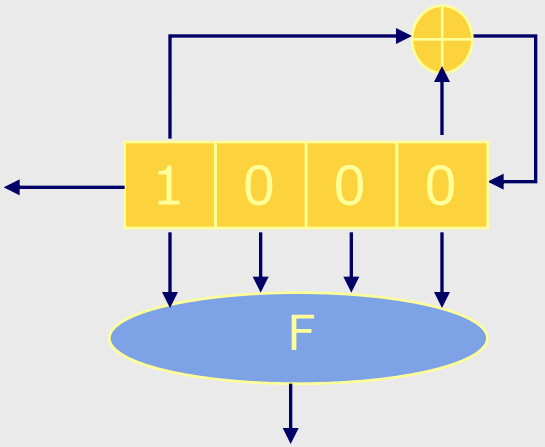


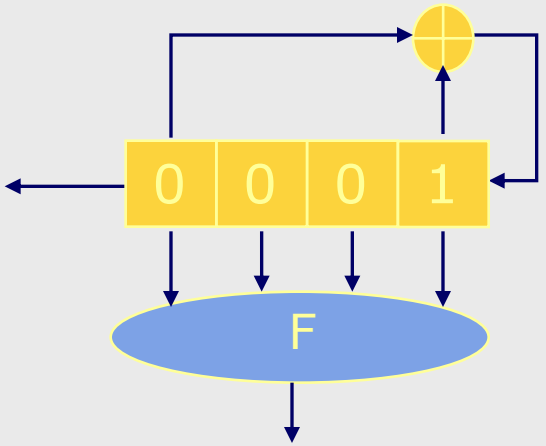




- Filtro no lineal
- Combinador no lineal

# Filtros no lineales

$$F(a_0, a_1, a_2, a_3) = a_0 a_1 \oplus a_1 a_2 \oplus a_3$$



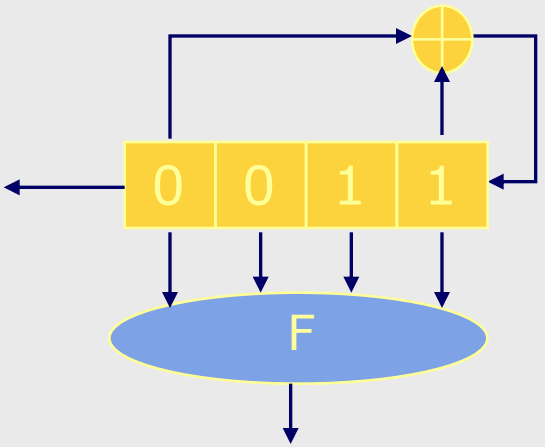

$$F(1,0,0,0) = 0 \oplus 0 \oplus 0 = 0$$

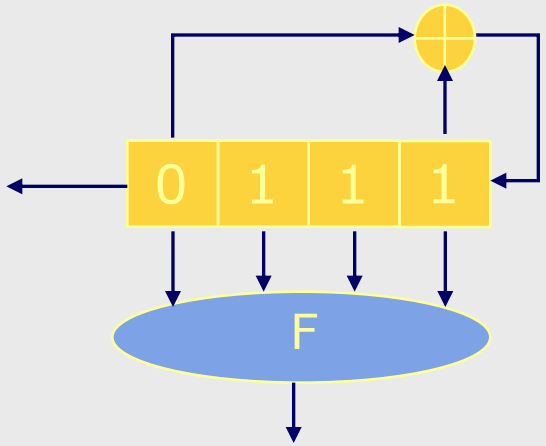
$$F(0,0,0,1) = 0 \oplus 0 \oplus 1 = 1$$

Sec. Generada = 0 1

# Filtros no lineales

$$F(a_0, a_1, a_2, a_3) = a_0 a_1 \oplus a_1 a_2 \oplus a_3$$



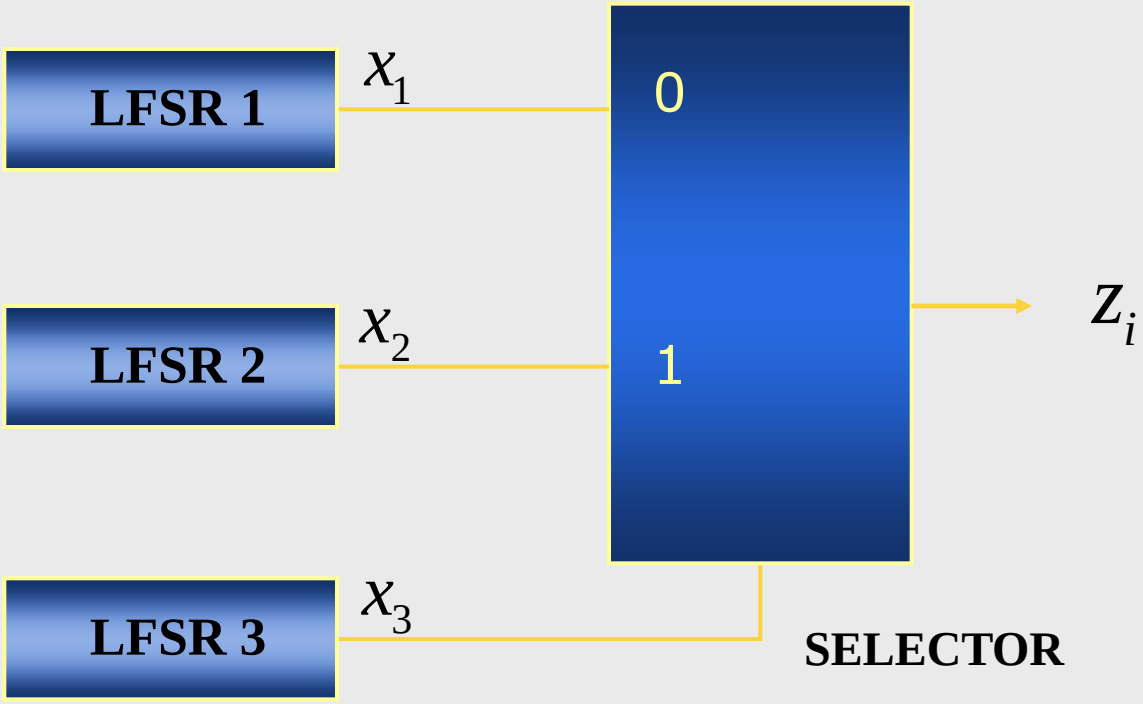




$$F(0,0,1,1) = 0 \oplus 0 \oplus 1 = 1$$

$$F(0,1,1,1) = 0 \oplus 1 \oplus 1 = 0$$

Sec. Generada = 0 1 1 0

# Generador de Geffe (1973)



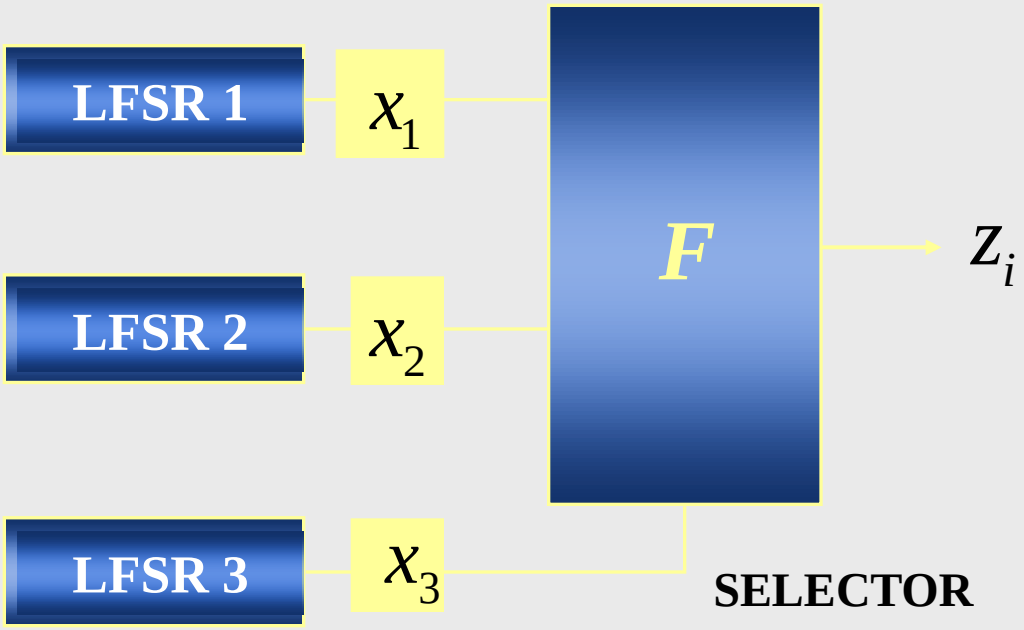

$$F(x_1, x_2, x_3) = x_1 \oplus x_1 x_3 \oplus x_2 x_3$$

# Generador de Geffe (II)

- $F(x_1, x_2, x_3) = x_1 \oplus x_1 x_3 \oplus x_2 x_3$
- $T = m.c.m.(2^{L_1} - 1, 2^{L_2} - 1, 2^{L_3} - 1)$
- $LC = L_2 + L_1 L_2 + L_1 L_3$
- Correlaciones!!!

$$L_1 = 61,\ L_2 = 62,\ L_3 = 63, \quad LC = 7687$$

# Ataques por correlación (Generador de Geffe 1973)




Leakage!!!!

$$z_i = F(x_1, x_2, x_3) = x_1 \oplus x_1 x_3 \oplus x_2 x_3$$

# Ataques por correlación

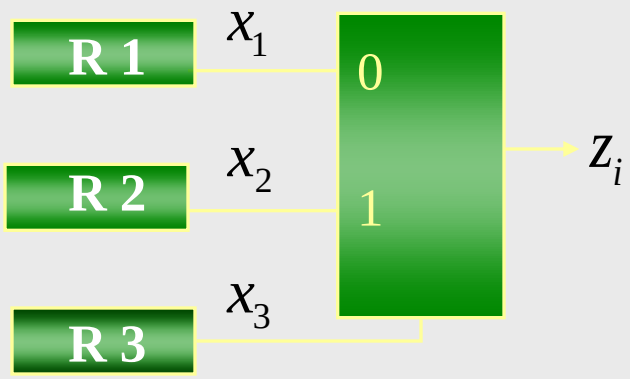



**Tabla de verdad**

| $X_1$ | $X_2$ | $X_3$ | $z_i$ |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

- Estrategia: “Divide and conquer”
- $2^{L_1+L_2+L_3}$ *e. i.* → $2^{L_1}, 2^{L_2}, 2^{L_3}$ por separado
  - $2^{41+43+44}$ → $2^{41}, 2^{43}, 2^{44}$ “ “
  - T. Siegenthaler, “Correlation-Immunity of Nonlinear Combining Functions”, IEEE Trans. on IT., 30 (1984), 312-317.

# Ataques por correlación (Geffe)

- Objetivo:
  - Determinar el estado inicial de R1, R2 y R3
- Requerimientos:
  - Conocimiento de una cierta cantidad de sec. cifrante $\{z_i\}$
- Idea general:
  - Identificar un subconjunto de posibles estados iniciales de R1 y R2.
  - Comparar cada par con todos los posibles estados iniciales de R3.

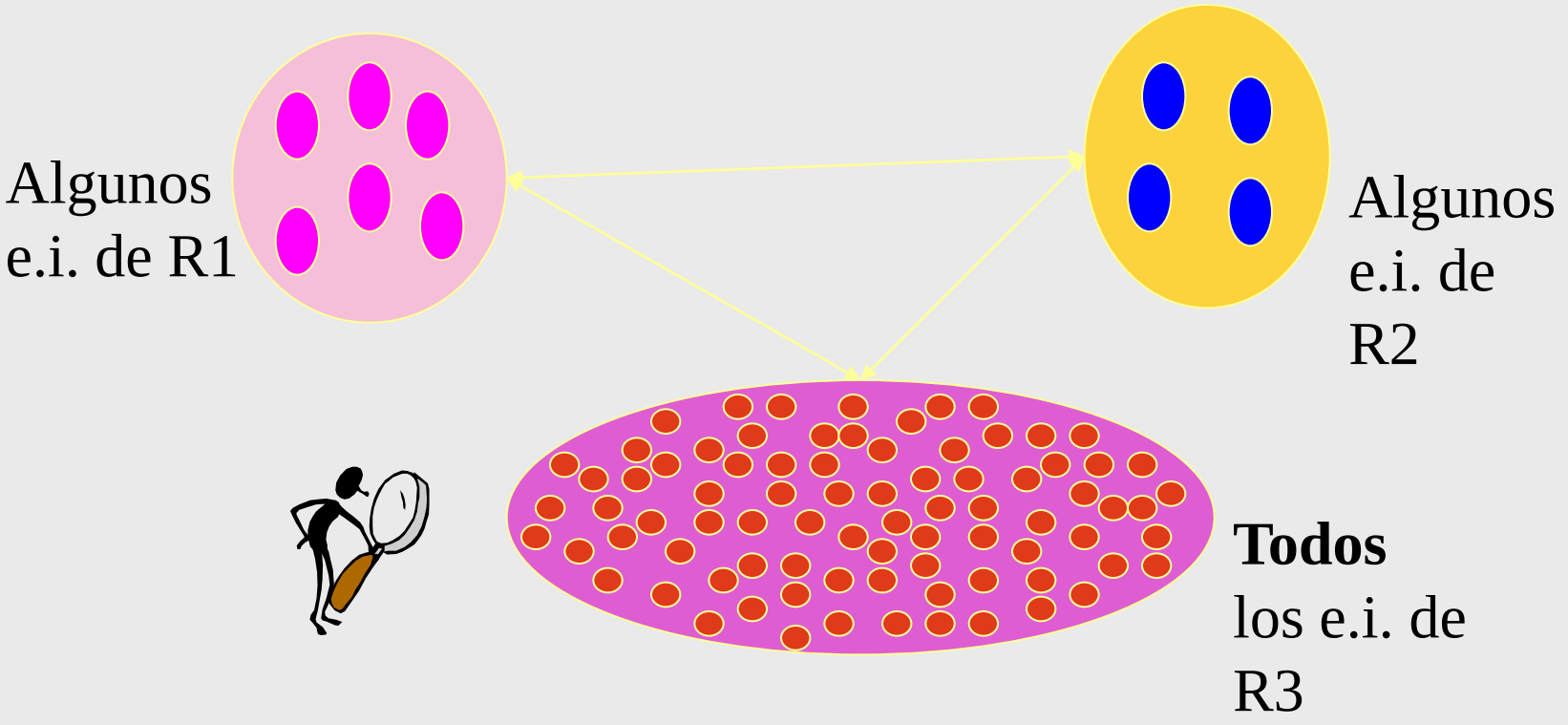
# Para un registro (no selector), p.e. R1

$2^{L_1}$ *e. i.*

| *e. i.* | Sec. gener. | Indice coinc. |
|---|---|---|
| 00…001 | 101000..001 | ≈ 50% |
| 00…010 | 001000..010 | ≈ 50% |
| ⋮ | ⋮ | ⋮ |
| 01…101 | 001001..101 | ≈ 75% |
| ⋮ | ⋮ | ⋮ |

- Medida: Distancia de Hamming (secuencias de igual longitud)
- Comparar: Sec. generada vs Sec.interceptada $\{z_i\}$
- Obtener un subconjunto de posibles *e. i.* para R1
- Actuar en paralelo para R2

# Ataques por correlación

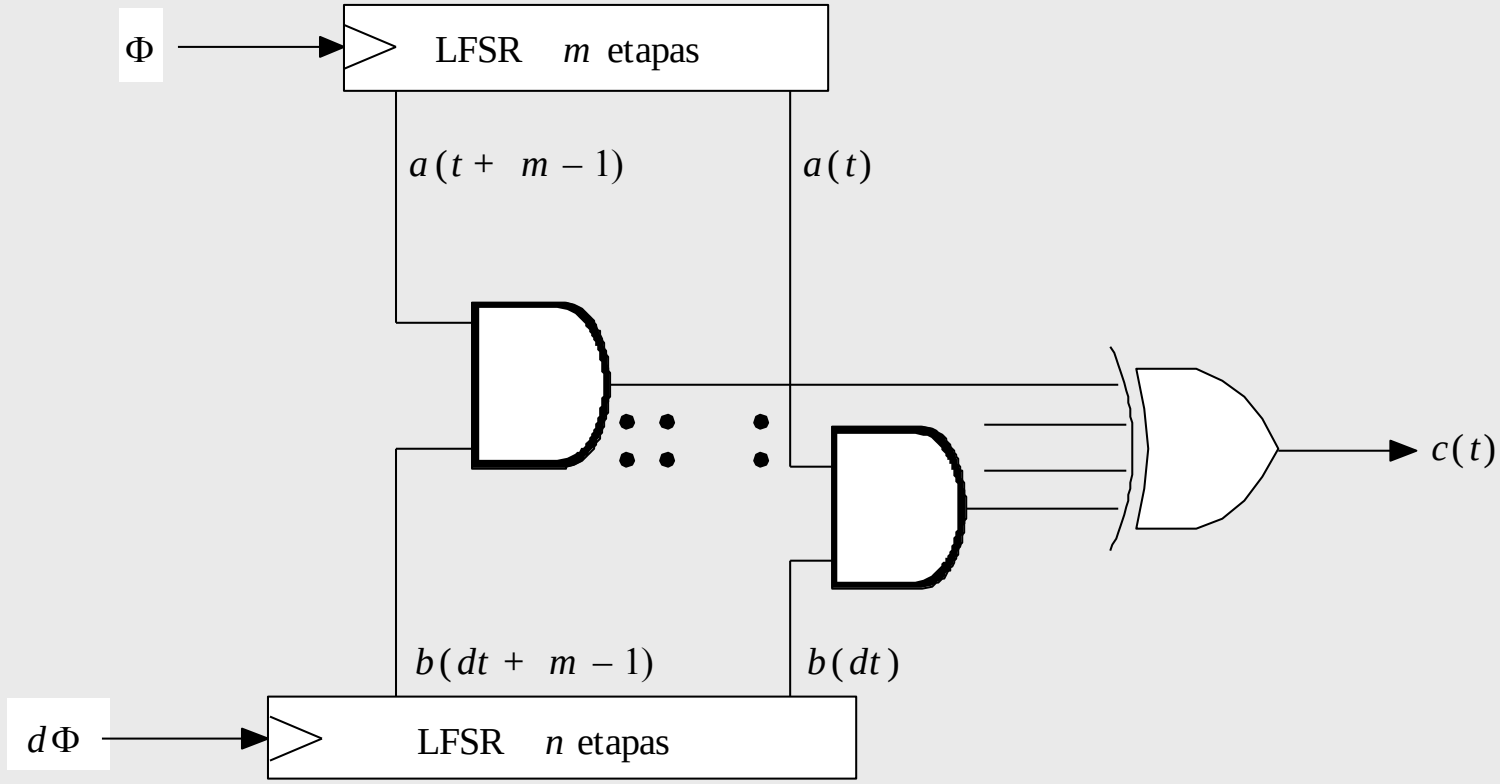



- Comprobar cada ( , , todos los ) con la sec. interceptada hasta descubrir el estado inicial del generador

# Referencias sobre correlación


- W. Meier & O. Staffelbach, *Fast correlation attacks on certain stream ciphers,* J. Cryptology, 1989, pp.159-176.
- R. Anderson, *Searching for the optimum correlation attack,* Fast Software Encryption 1994, 1008, pp. 137-143.
- R. Anderson, *Faster Attack on certain stream ciphers,* Electro. Letters, 29, 1993.
- J. Golic et al. , *Fast correlation attacks on nonlinear filter generators,* Information Processing Letters, 64 (1), 1997.
- J. Golic, *Fast Correlation Attacks on the summation generator,* J. Cryptology, (13) 2000, pp.245-262.
- M. Mihaljevic et al., *A low-complexity algorithm for the fast correlation attack,* FSE 2000, LNCS, 1978 (2000), pp. 196-212.
- http://www.cryptosystem.net/stream/
- J. Golic, *Correlation Analysis on the Alternating Step Generator,* Designs, Codes & Crypto, 31, pp. 51-74, 2004.

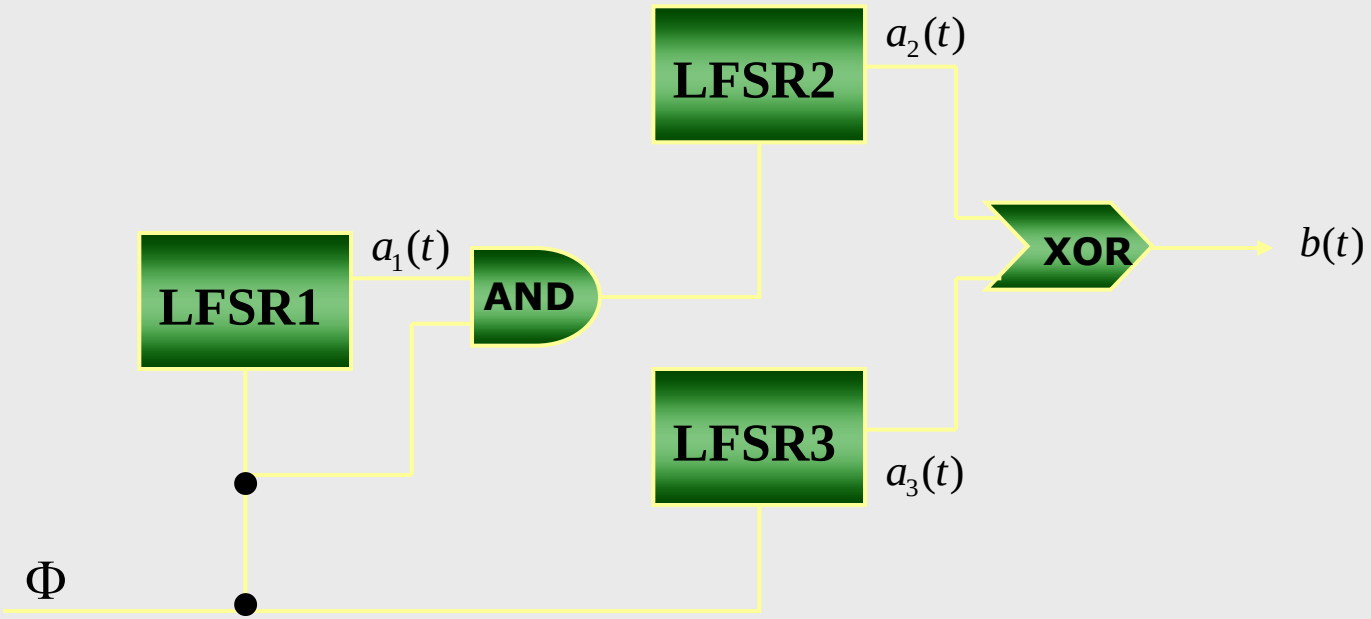
# Generador de Massey-Rueppel



# Generador de Massey-Rueppel (II)

- Los LFSRs desplazan a distinta velocidad
- Muy buena distribución de 0's y 1's
- $T = (2^{L_1} - 1)\,(2^{L_2} - 1)$
- $LC = L_1\ L_2$ lineal en las longitudes de los LFSRs
- Criptoanálisis por baja LC

# Generador de Beth-Piper




Φ Función pulsos de reloj

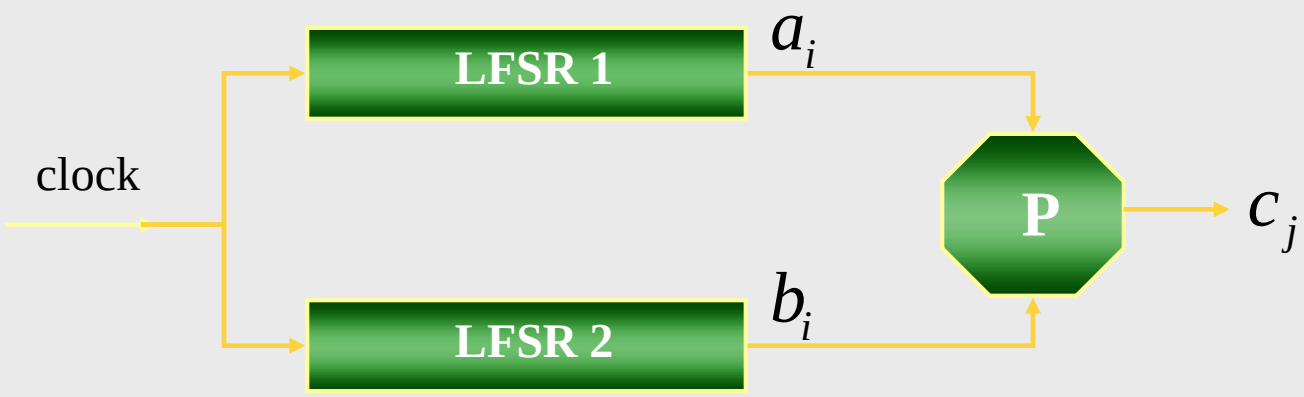
# Generador de Beth-Piper (II)

- Modificación en el reloj de algunos LFSRs
- $T = (2^{L_1} - 1)\,(2^{L_2} - 1)\,(2^{L_3} - 1)$
- $LC = (2^{L_1} - 1)\,L_2 + L_3$
- Posible criptoanálisis por correlación

  J. Golic, *Correlation Analysis on the Alternating Step Generator,* Designs, Codes & Crypto, 31, pp. 51-74, 2004.

# El Generador Shrinking (1993)

- Generador binario muy sencillo (Crypto'93)
- Compuesto de dos LFSRs: LFSR1 y LFSR2




- Según P, LFSR1 (registro de control) decima la secuencia producida por LFSR2

# El Generador Shrinking (Un ejemplo)

LFSRs:

- LFSR1: $L_1=3,\ P_1(D)=1+D^2+D^3,\ IS_1=(1,0,0)$
- LFSR2: $L_2=4,\ P_2(D)=1+D+D^4,\ IS_2=(1,0,0,0)$

Ley de decimación P:

- $\{a_i\}$= 1 0 0 1 0 1 1 1 0 0 1 0 1 1 1 …
- $\{b_i\}$= 1 <u>0</u> <u>0</u> 0 <u>1</u> 1 1 1 <u>0</u> <u>1</u> 0 <u>1</u> 1 0 0 …
- $\{c_j\}$= 1 0 1 1 1 0 1 0 0 …

Los bits subrayados (<u>1</u> y <u>0</u>) se eliminan

# Características criptográficas de la secuencia generada

- Periodo:

$$T = (2^{L_2} - 1)\, 2^{(L_1 - 1)}$$

- Complejidad lineal:

$$L_2\, 2^{(L_1 - 2)} < LC \leq L_2\, 2^{(L_1 - 1)}$$

- Número de 1's:

$$No.1's = 2^{(L_2 - 1)}\, 2^{(L_1 - 1)}$$

secuencia equilibrada

# Características criptográficas de la secuencia generada

- Periodo:

$$T = (2^{L_2} - 1)\, 2^{(L_1 - 1)}$$

- Complejidad lineal:

$$L_2\, 2^{(L_1 - 2)} < LC \leq L_2\, 2^{(L_1 - 1)}$$

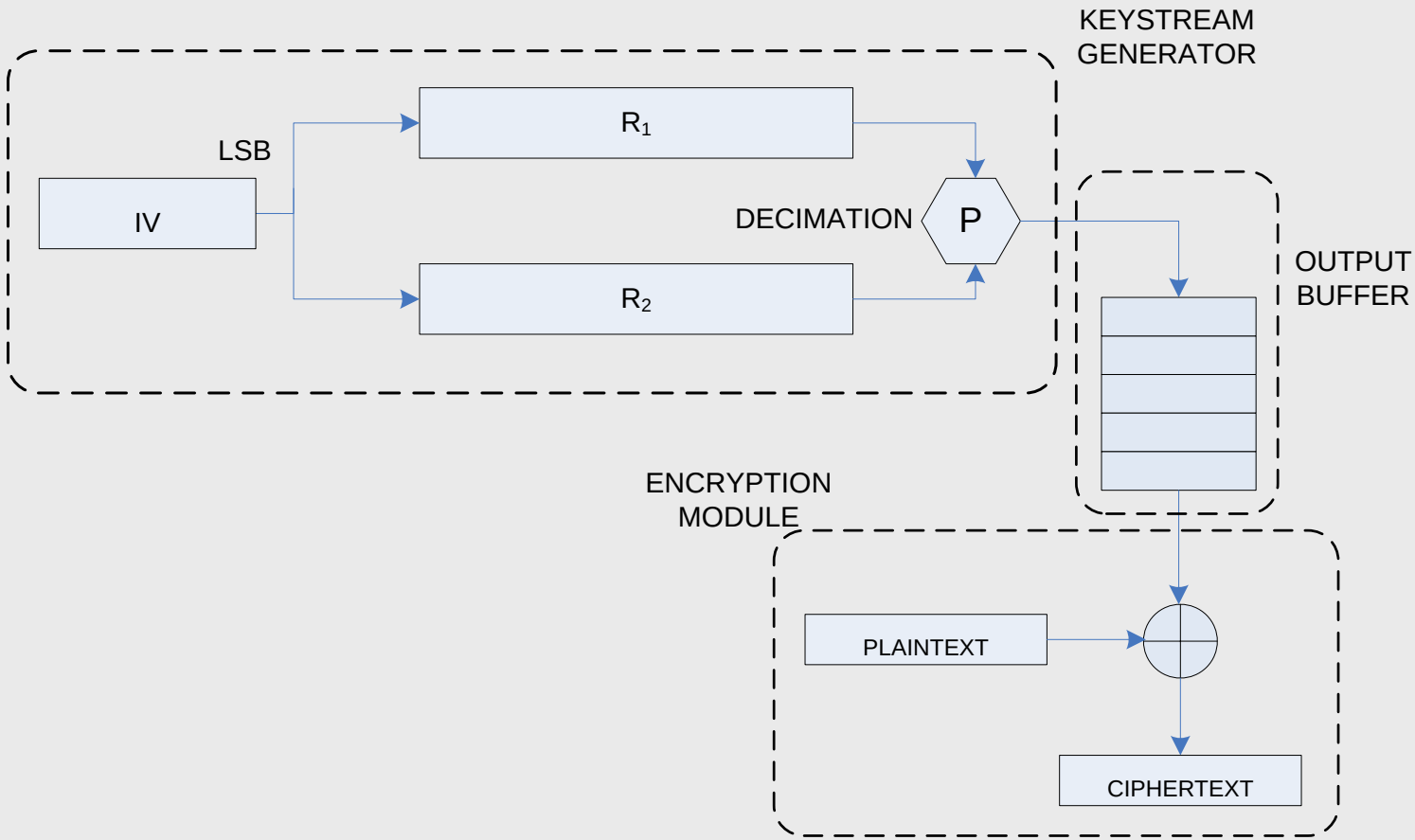
- Número de 1's:

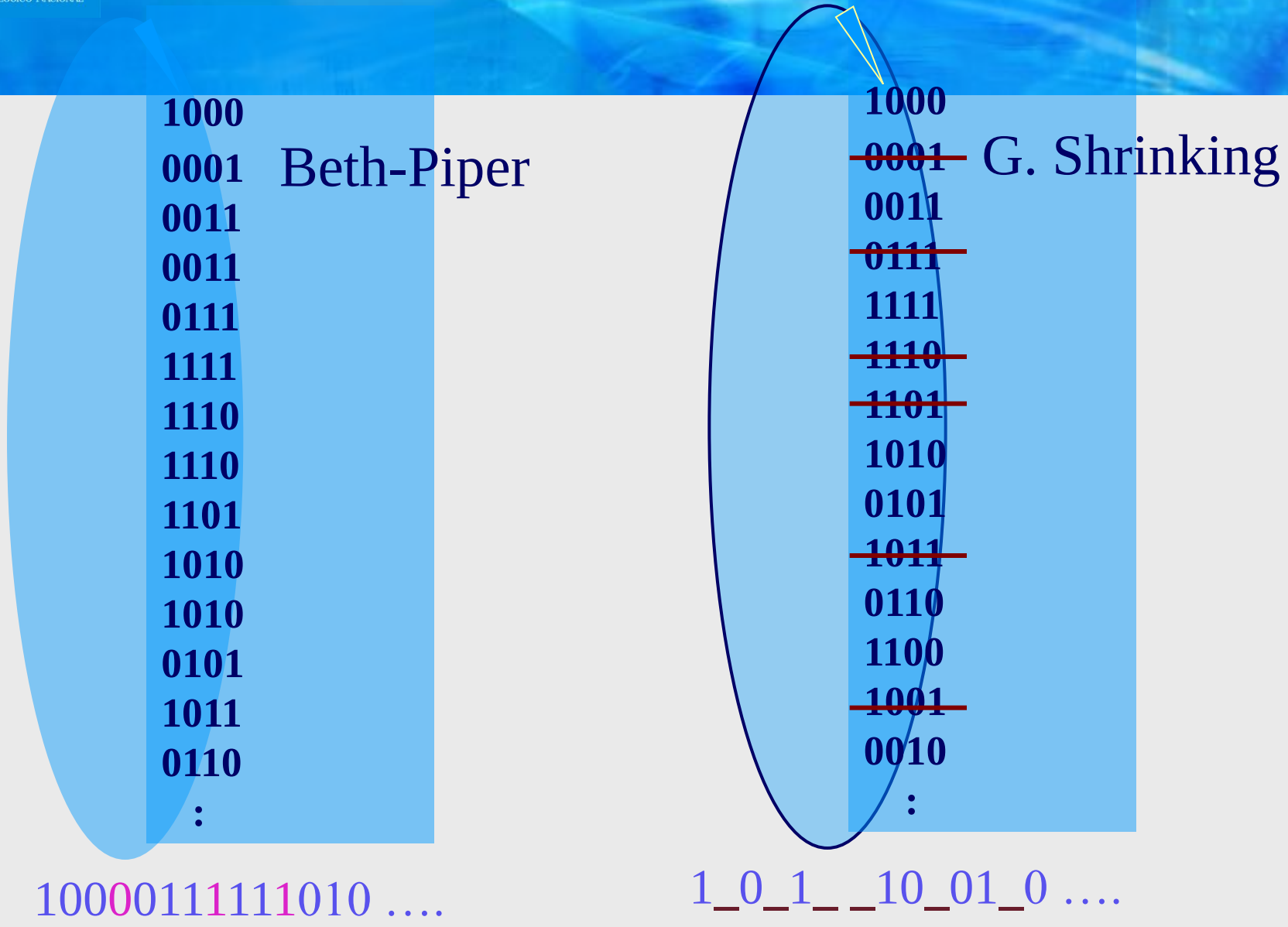
$$No.1's = 2^{(L_2 - 1)}\, 2^{(L_1 - 1)}$$

secuencia equilibrada

# El Generador Shrinking (1993)

- **Implementación**




1000
0001
0011
0011
0111
1111
1110
1110
1101
1010
1010
0101
1011
0110
:
Beth-Piper
1000
0001
0011
0111
1111
1110
1101
1010
0101
1011
0110
1100
1001
0010
:
G. Shrinking
10000111111010 ….
1_0_1_ _10_01_0 ….