Universidad
Carlos III de Madrid

COSEC LAB· Dpto. Informática

T 2 CRIPTOGRAFÍA
T 2.3 CRIPTOSISTEMAS SIMÉTRICOS. CIFRADORES DE BLOQUE Y FLUJO (parte 3)

Criptografía y seguridad informática
Seguridad en las tecnologías de la información
@ COSEC

Curso 2016-2017

CONTENIDOS

- Criptosistemas simétricos y asimétricos
- Criptosistemas simétricos
  - Cifradores de bloque
  - **Cifradores de flujo**
    - **Introducción**
    - Tipos
    - Serie cifrante
    - PRNGs criptográficos
      - LFSRs
    - Cifrador de flujo. Ventajas y desventajas
    - RC4

Introducción

Características de los cifradores de flujo

- Descomponen el mensaje en bytes (o en bits):

$$M = m_1, m_2, \ldots m_n$$

- Cifran cada m_i con el correspondiente k_i de la serie cifrante
  - idealmente infinita y aleatoria
    - $K = k_1, k_2, \ldots k_n, k_{n+1}, \ldots$

- $E_K (M) = E_{k1} (m_1) E_{k2} (m_2) \ldots E_{kn} (m_n)$

Introducción

- **Cifrado de Vernam. One-time-pad**

  - Cifrado: $E(M) = M \oplus K = m_1+k_1, m_2+k_2, \ldots, m_n+k_n$

$$\begin{array}{cccccccccc} & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & \mathtt{M} \\ \oplus & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & \mathtt{K} \\ \hline & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & \mathtt{C} \end{array}$$

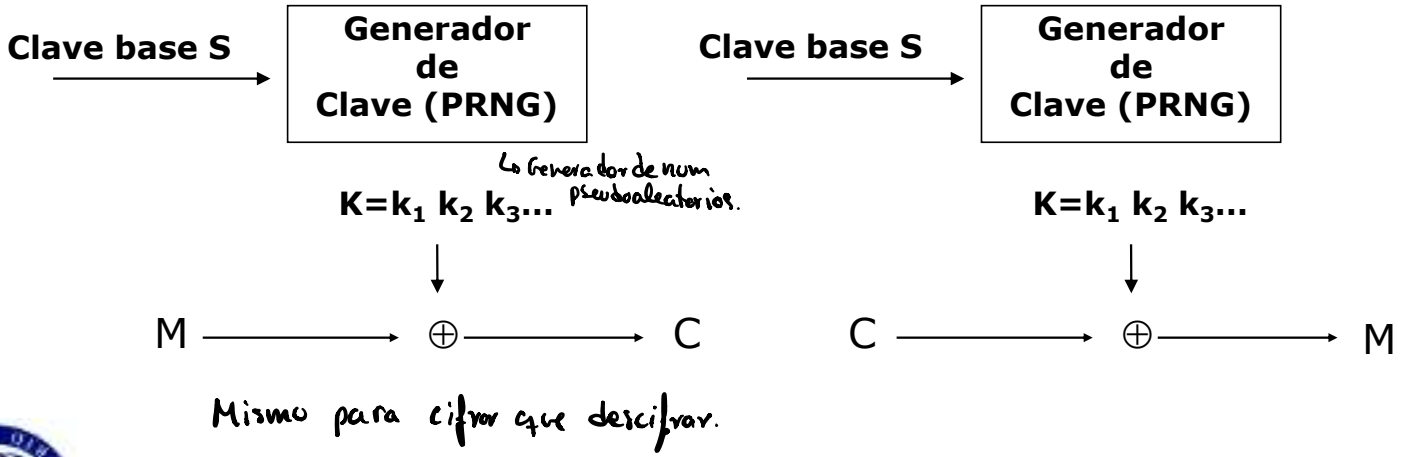
  - Descifrado: $M = E(M) \oplus K$
  - Shannon demostró que el cifrado de Vernam es incondicionalmente seguro si la clave K:
    - Es realmente aleatoria
    - Se usa una sola vez
    - Es de longitud igual o mayor que M

Introducción

- Vernam no es práctico

Cifrador de flujo práctico:

- K: serie cifrante obtenida a partir de clave base

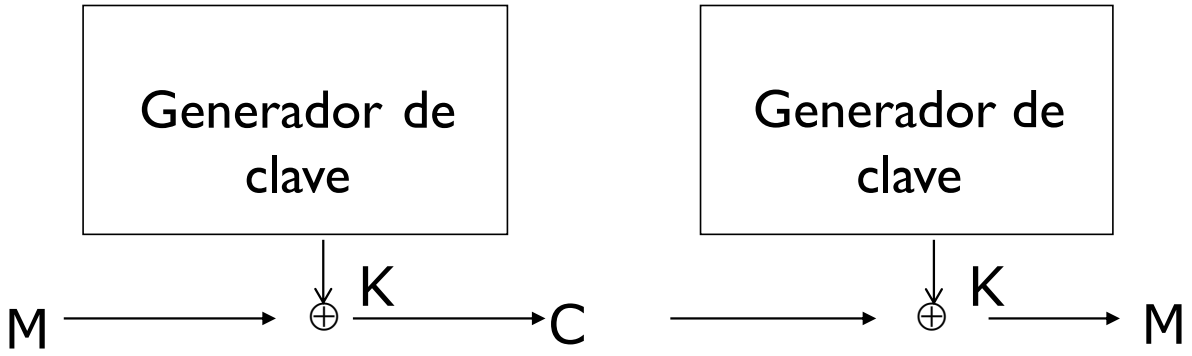


CONTENIDOS

- Criptosistemas simétricos y asimétricos
- Criptosistemas simétricos
  - Cifradores de bloque
  - Cifradores de flujo
    - Introducción
    - **Tipos**
    - Serie cifrante
    - PRNGs criptográficos
      - LFSRs
    - Cifrador de flujo. Ventajas y desventajas
    - RC4

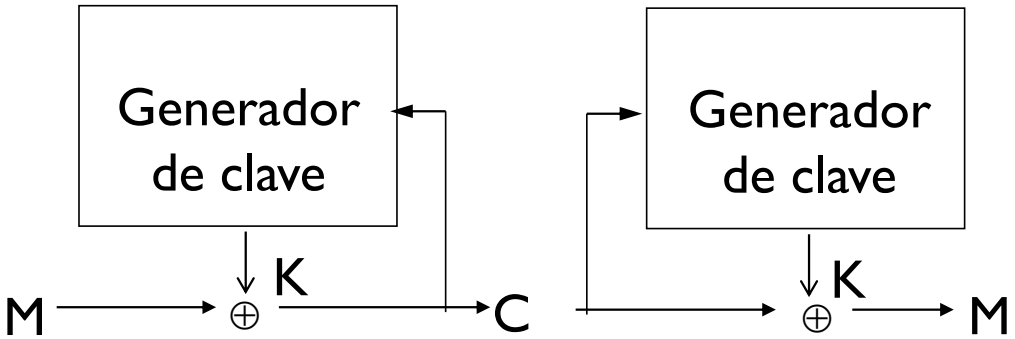
Tipos de cifradores de flujo

- Síncrono
  - Emisor y receptor se sincronizan externamente
  - Serie cifrante independiente del texto en claro y del criptograma



Tipos de cifrado de flujo

- Autosíncrono
  - Emisor y receptor se sincronizan automáticamente
  - La serie cifrante es una función de símbolos previamente cifrados



CONTENIDOS

- Criptosistemas simétricos y asimétricos
- Criptosistemas simétricos
  - Cifradores de bloque
  - Cifradores de flujo
    - Introducción
    - Tipos
    - **Serie cifrante**
    - PRNGs criptográficos
      - LFSRs
    - Cifrador de flujo. Ventajas y desventajas
    - RC4

Serie cifrante

- Aproximación para generar de la serie cifrante en emisor y receptor
  - Mediante un generador de números pseudoaleatorios
    - Generación determinista
  - A partir de una clave base (secreta e impredecible)
    - De centenas de bits (para evitar ataques de fuerza bruta)

Serie cifrante.

- **Propiedades deseables: Postulados de Golomb**
- Postulado G1:
  - Debe existir igual número de ceros que de unos. Se acepta como máximo una diferencia igual a la unidad.
- Postulado G2:
  - La mitad de las rachas (sucesión de dígitos iguales) tiene longitud 1, la cuarta parte tiene longitud 2, la octava longitud 3, etc. Que no haya demasiados iguales seguidos.
- Postulado G3:
  - Para todo k, la Autocorrelación fuera de fase AC(k) es igual a una constante.
- Función de Autocorrelación:
  - Desplazamiento de la secuencia S de período T de k bits hacia la izquierda:
  - AC(k) = (A - F) / T
  - Aciertos = bits iguales Fallos = bits diferentes

Serie cifrante

Postulados de Golomb. Ejemplo

```
k=1
1 1 1 1 0 1 0 1 1 0 0 1 0 0 0
1 1 1 0 1 0 1 1 0 0 1 0 0 0 1
^ ^ ^           ^   ^       ^ ^
```

A = 7, F = 8 => AC(1) = -1/15

Ejercicio:

Compruebe que para esta secuencia cifrante

s_i = 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0

la Autocorrelación fuera de fase AC(k) para todos los valores de k ($1 \leq k \leq 14$) es constante e igual a -1/15.

s_i = 1 1 1 1 0 1 0 1 1 0 0 1 0 0 0

} Desplatado 1 itg.

k=1 1 1 1 0 1 0 1 1 0 0 1 0 0 0 1 A=7

k=2 1 1 0 1 0 1 1 0 0 1 0 0 0 1 1 A=7

k=3 1 0 1 0 1 1 0 0 1 0 0 0 1 1 1 A=7

Los que coinciden con la anterior desplatada.

Rachas de 00 1 0 1 00 1 1 1 0 1 1 0

00 → 11
1 → 11
0 → 111
111 → 1
11 → 1

Rachas de 1 símbolo: 5 — 1/2 — Hay 14 por lo que para que cumple debe haber: 7

Rachas de 2 símb: 3 — 1/4 — 3-4

Rachas de 3 símb: 1 — 1/8 — 1-2

Se cumple el postulado $5 \in 7$, $3 \in (3,4)$ y $1 \in (1,2)$

0 0 1 0 1 0 0 1 1 1 0 1 1 0

k=1 0 1 0 1 0 0 1 1 1 0 1 1 0 0 A=6

k=2 1 0 1 0 0 1 1 1 0 1 1 0 0 0 A=6

k=3 0 1 0 0 1 1 1 0 1 1 0 0 0 1 A=6

k=4 1 0 0 1 1 1 0 1 1 0 0 0 1 0 A=6

⋮

k=15 Buscamos que falle. (A≠6)

Serie cifrante

- Propiedades deseables
  - Período muy grande
  - Aleatoriedad: Distribución uniforme, independencia
  - Impredecibilidad
    - Se puede medir por su complejidad lineal LC
      - número de bits necesarios para predecir el resto de la secuencia
      - viene dada por la longitud mínima del LFSR capaz de reproducirla
      - Se calcula L (número de celdas) y si se conocen 2L bits se puede predecir el resto de la serie
    - Meta: conseguir una complejidad lineal lo más alta posible

CONTENIDOS

- Criptosistemas simétricos y asimétricos
- Criptosistemas simétricos
  - Cifradores de bloque
  - Cifradores de flujo
    - Introducción
    - Tipos
    - Serie cifrante
    - **PRNGs criptográficos**
      - LFSRs
    - Cifrador de flujo. Ventajas y desventajas
    - RC4

PRNGs criptográficos

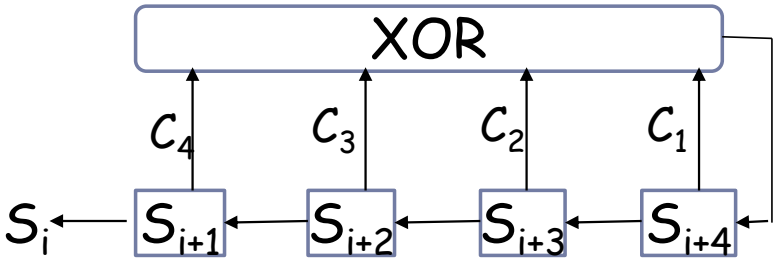
- Basados en algoritmos criptográficos existentes
  - Cifradores simétricos
  - Cifradores asimétricos
  - Funciones resumen
- Ad-hoc
  - Generador de registros de desplazamiento
  - **LFSR *(linear feed-back shift register)***
  - A5/1 (2000)
  - A5/2 (2001)
  - PRNG propio de RC4

CONTENIDOS

- Criptosistemas simétricos y asimétricos
- Criptosistemas simétricos
  - Cifradores de bloque
  - Cifradores de flujo
    - Introducción
    - Tipos
    - Serie cifrante
    - PRNGs criptográficos
      - **LFSRs**
    - Cifrador de flujo. Ventajas y desventajas
    - RC4

LFSR

Registro de desplazamiento con retroalimentación lineal [Linear Feedback Shift Register (LFSR)]



Polinomio de conexión asociado:

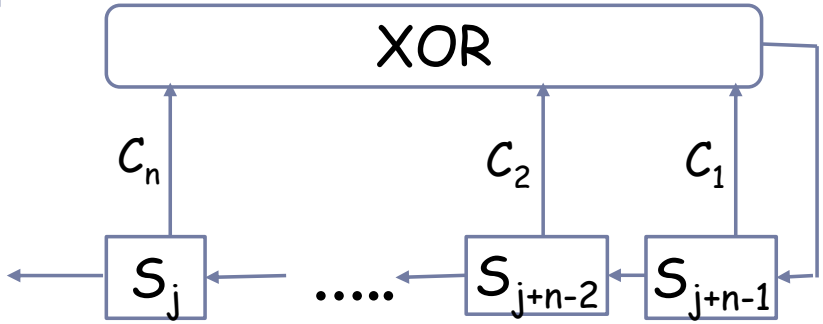
$f(x) = C_4x^4 + C_3x^3 + C_2x^2 + C_1x + 1$

Función única: XOR

$T_{máx} = 2^4 - 1$

Valores iniciales = "semilla",

prohibido cadena de ceros



$f(x) = C_nx^n + C_{n-1}x^{n-1} + ... + C_2x^2 + C_1x + 1$

$T_{máx} = 2^n - 1$

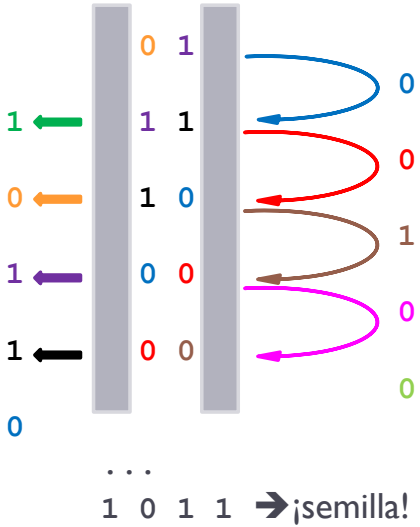
LFSR

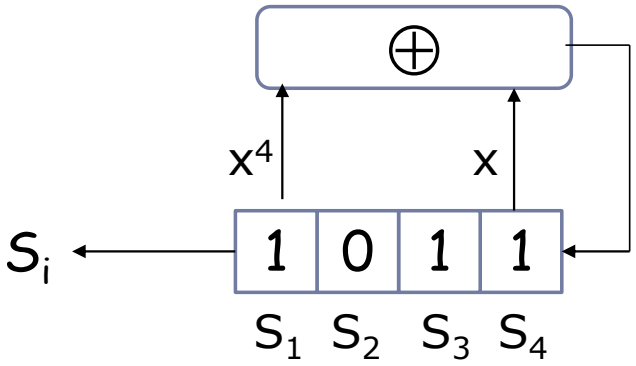
EJEMPLO -- Generador LFSR de cuatro celdas (n = 4)

Mirarlo está aparte.

- Clave base: $S_1S_2S_3S_4 = 1\ \ 0\ \ 1\ \ 1$
- Polinomio de conexión $f(x) = x^4 + x + 1$
- En este ejemplo el periodo es $T = T_{máx} = 2^n - 1$

Bit s_i Registro bit realim.





S_i = 101100100011110
T = 15

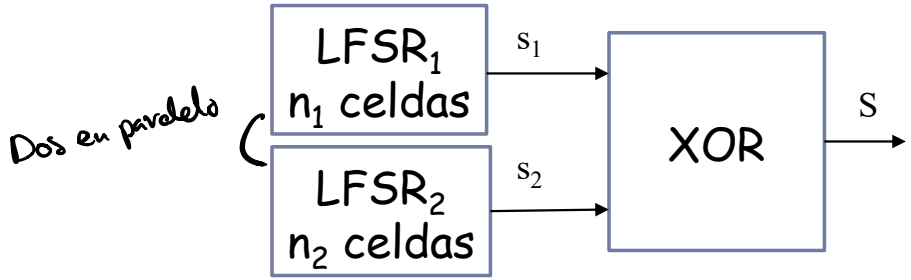
LFSR

- Periodos altos pero complejidad lineal muy baja
- Solución:
  - Aumentar la complejidad lineal del generador
  - Eg, empleando varios LFSRs
    - Operaciones lineales de secuencias seudoaleatorias
    - Operaciones no lineales de las secuencias seudoaleatorias
    - Filtrado no lineal de los estados de un LFSR
    - Otros

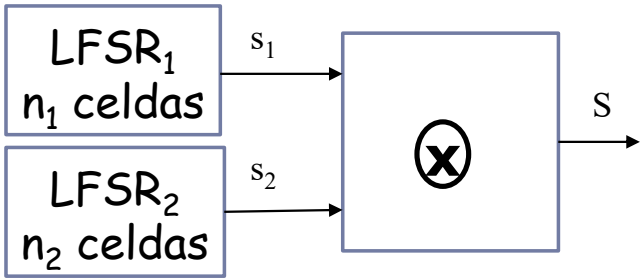
LFSR

Aumentando la complejidad lineal de los LFSRs

- Operaciones lineales de secuencias seudoaleatorias:



- ~~Operaciones no lineales~~ de las secuencias seudoaleatorias:



CONTENIDOS

- Criptosistemas simétricos y asimétricos
- Criptosistemas simétricos
  - Cifradores de bloque
  - Cifradores de flujo
    - Introducción
    - Tipos
    - Serie cifrante
    - LFSRs
    - PRNGs criptográficos
    - **Cifrador de flujo. Ventajas y desventajas**
    - RC4

Cifrado de flujo. Ventajas y desventajas

- **Ventajas:**
  - Transformación byte a byte, o bit a bit
    - Altas velocidades de cifrado
  - Los errores de transmisión no se propagan

- **Desventajas:**
  - Escasa difusión de la información
    - Cada símbolo de M se corresponde con uno de C
  - Las series cifrantes no son realmente aleatorias
    - Generación determinista
  - Problemas de reutilización de la clave →

Cifrado de flujo. Ventajas y desventajas

- Problemas de reutilización de la clave:
  - Ataque con texto original conocido

    Se puede obtener K, teniendo M y C:

    $$M \oplus C = M \oplus M \oplus K = K$$

  - Ataque sólo al criptograma

    M_i a partir de C_i y C_j escogidos si M_j predecible:

    $$C_i \oplus C_j = M_i \oplus K \oplus M_j \oplus K = M_i \oplus M_j$$

CONTENIDOS

- Criptosistemas simétricos y asimétricos
- Criptosistemas simétricos
  - Cifradores de bloque
  - Cifradores de flujo
    - Introducción
    - Tipos
    - Serie cifrante
    - LFSRs
    - PRNGs criptográficos
    - Cifrador de flujo. Ventajas y desventajas
    - **RC4**

RC4

- Algoritmo propietario de RSA
- Inicialmente secreto, luego desensamblado y publicado en sci.crypt
- Diseño de Ron Rivest, simple pero muy efectivo
- Tamaño variable de clave, trabaja sobre bytes
- Muy simple -> rápido en sw
- Muy usado (web SSL/TLS, wireless WEP, etc.)

RC4

1. **Fase de inicialización**
   - Clave base variable de 1 a 256 bytes
   - Vector de estados S={S[0],S[1],...,S[255]}
     - S es el estado interno del cifrador
   - Usa la clave para permutar el vector S
   - Dada una clave k de longitud l bytes

```
for i = 0 to 255 do
    S[i] = i
j = 0
for i = 0 to 255 do
    j = (j + S[i] + k[i mod l]) (mod 256)
    swap (S[i], S[j])
```

RC4

2. Serie cifrante y 3. cifrado

- En cada paso de cifrado se modifica S : En cada paso se vuelve a desordenar/permut
- La suma de un par de valores en S determina el byte de salida

Cogemos 2 (i,j) posiciones, sumamos sus valores y la clave es el elemento de la posición de la suma. Y cambiamos las 2 pos, i y j

```
i = j = 0
    for each message byte Mi
        i = (i + 1) (mod 256)  // contador simple
        j = (j + S[i]) (mod 256) // simula un random-walk
        swap(S[i], S[j])
        t = (S[i] + S[j]) (mod 256)
        Ci = Mi ⊕ S[t]
```

1.ª fase:

¿Que valor de la clave hace que el vector no se desordene?

$j = j + S(i) + k$

$j = 0$

$i = 0$

$j = 0 + 0 + k_0 \Rightarrow k_0 = 0$

$i = 1 \quad j = 0$

$j = 0 + 1 + k_1 = 1 \Rightarrow k_1 = 0$

$i = 2 \quad j = 1$ (2) Son igual a i para que no se desordenen, ya que después de esto hace swap(S[i], S[j])

$j = 1 + 2 + k_2 = 2 \Rightarrow k_2 = 255$ ($3 + k_2 = 2$; $-1 \bmod 256 = 255$)

$i = 3 \quad j = 2$

$j = 2 + 3 + k_3 = 3 \Rightarrow k_3 = 254$

$i = 4 \quad j = 3$ ($5 + k_3 = 3$; $k = -2 \bmod 256$)

$j = 3 + 4 + k_4 = 4 \Rightarrow k_4 = 253$

Hay 255 y como hay 2 caros llega

La clave será: $[k] = \{0, 0, 255, 254, 253, \ldots, 2\}$

2ª fase: $S = \{4, 7, 9, 1, 2, 6, 8, 3, 5, 0\}$ (posiciones 0 a 9)

$i = j = 0$

$i = 1 \quad i + 1$

$j = 0 + 7 = 7 \quad j + S[i]$ swap(S[i], S[j])

$\{4, 7, 9, 1, 2, 6, 8, 3, 5, 0\}$

$t = 3 + 7 = 10 \bmod 10 = 0 \quad S[i] + S[j]$

$k_0 = 4$

$i = 2$

$j = 7 + 9 = 6$

$\{4, 3, 8, 1, 2, 6, 9, 7, 5, 0\}$

$t = 17 \bmod 10 = 7$

$k_1 = 7$

$i = 3 \quad S[3]$ sería 1

$j = 6 + 3 = 9$

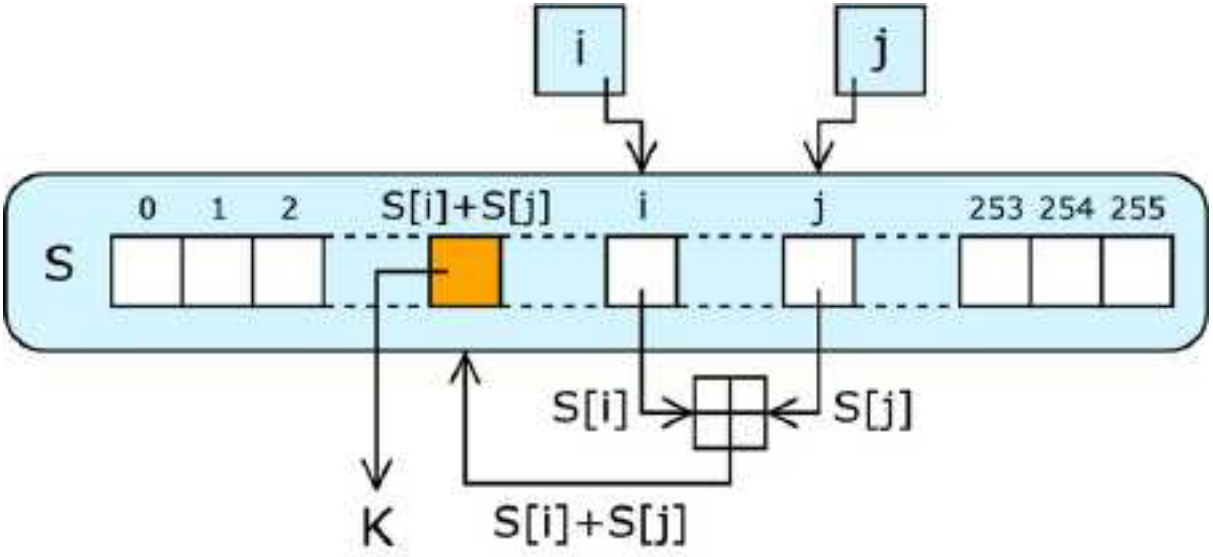
$\{4, 3, 8, 0, 2, 6, 9, 7, 5, 1\}$

$t = 0 + 1 = 1$

$k_3 = 3$

RC4

Serie cifrante



RC4

Seguridad

- El resultado es muy no-lineal
- Ningún ataque práctico con tamaño de clave base razonable (128 bits o más) **HASTA** el año 2015
  - había ataques contra malas implementaciones concretas
    - La serie cifrante sufre un bias…

RC4

Seguridad

- En 2015 investigadores de KU Leuven han demostrado ataques "prácticos"
  - Recuperar una cookie segura (enviada sobre HTTP con TLS) en 75 horas
  - Descifrar e inyectar paquetes arbitrarios en WPA-TKIP en 1 hora
- Se está prohibiendo poco a poco su uso en los protocolos que lo contemplan (eg, RFC 7465 lo prohibe para TLS)
- Se están buscando nuevos algoritmos para sustituir a RC4
- De momento:
  - AES-CTR (AES con Counter Mode) o AES-GCM
  - Salsa 20 (resultado del proyecto europeo eSTREAM)