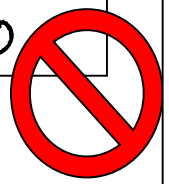
Clase 3

25 de marzo 2024

Secreto Perfecto



SEGURIDAD COMPUTACIONAL

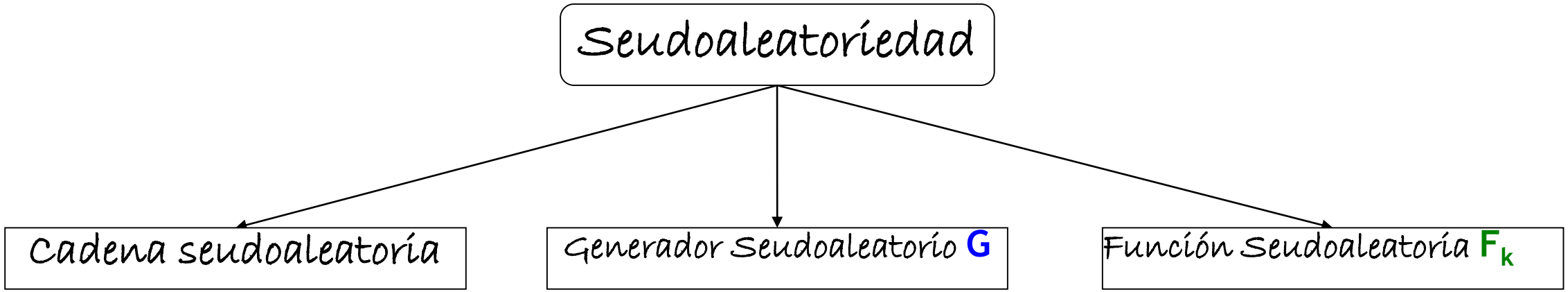
- Adversarios Eficientes
- Probabilidad éxito despreciable

Redefine Esquema π(Gen, Enc, Dec) priv.

- $k \leftarrow Gen(1^n)$
- $c \leftarrow Enc_k(m)$
- $m := Dec_k(c)$

Redefine el experimento:

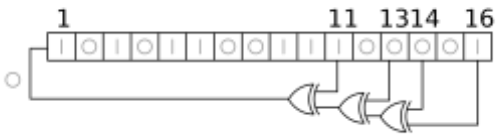
$$\Pr\left[\Pr ivK^{eav}_{A,\pi}(n) = 1\right] \leq \frac{1}{2} + negl(\ )$$


Seudoaleatoriedad
Cadena seudoaleatoria
Generador Seudoaleatorio G
Función Seudoaleatoria F_k

Cadena seudoaleatoria: parece una cadena con una distribución uniforme (aleatoria)

Generador Seudoaleatorio G:

Algoritmo determinístico que recibe una semilla s pequeña y aleatoria y la expande a una r de longitud mayor y seudoaleatoria = STREAM CIPHER.



Ej: **RC4, LFSR**

Es facil generar valores pequeños aleatorios, se complica para valores mas grandes

$r := G(s)$ con $|s| = n$ y $|r| = l(n) > n$

La semilla s es el secreto.

G es el generador que expande la semilla

Redefine OTP:

- $k \leftarrow Gen(1^n)$
- $c := G(k) \oplus m$
- $m := G(k) \oplus c$

- $|k| = n$
- $|m| = l(n) > n$

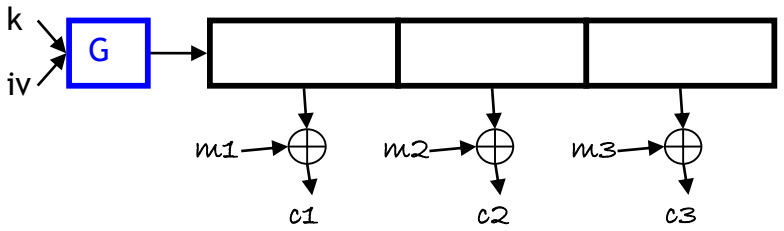
ESQUEMA SEGURO ante Eavesdropping PARA MENSAJES DE LONGITUD FIJA.

$$\Pr\left[\Pr iv K^{eav}_{A,OTP}(n) = 1\right] \leq \frac{1}{2} + negl(\ )$$

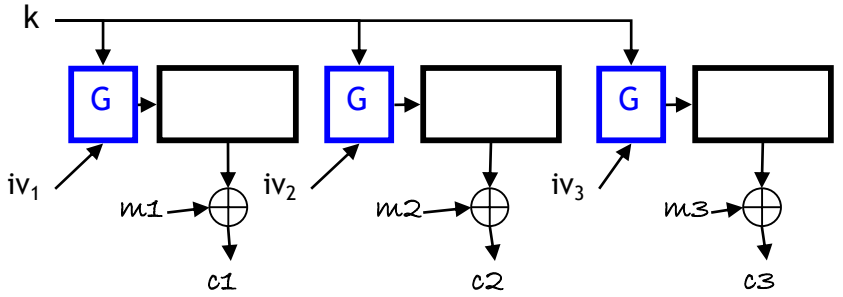
OTP NO ES SEGURO PARA MÚLTIPLES MENSAJES. (Vectores de mensajes)

Seguridad para múltiples mensajes

- Modo Síncrónico



- Modo Asíncrónico

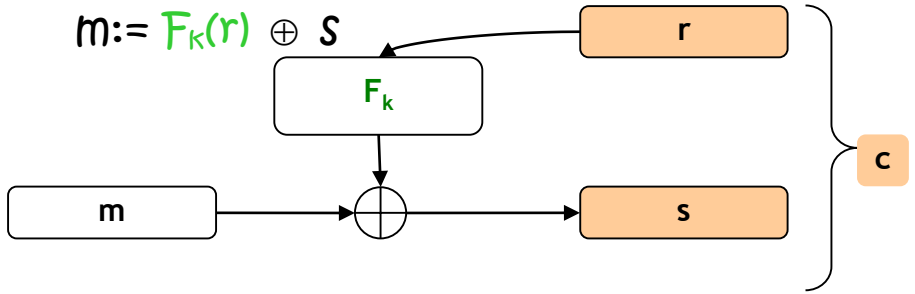


En el modo asincronico, yo tengo que enviar todos los IV para que el receptor lo pueda decodificar. El modo sincronico usa siempre el mismo IV, aunque el inconveniente es que si pierdo algun bit en el camino no voy a poder decodificar el resto de los mensajes.

Función Seudoaleatoria F_k: Función que es indistinguible de una función elegida en forma aleatoria del conjunto de $F : \{0,1\}^n \rightarrow \{0,1\}^n$ = BLOCK CIPHERS

Esquema π(Gen,Enc,Dec):

- Gen: $k \leftarrow \{0,1\}^n$ ($|k|=n$)
- Enc: Para m de $|m|=n$, se elige $r \leftarrow \{0,1\}^n$
  $c := \langle r, F_k(r) \oplus m \rangle$
- Dec: Dados $\langle r,s \rangle$
  $m := F_k(r) \oplus s$



La funcion toma un bloque de n bits y me da un bloque de n bits

Lo que me da la funcion de inicializacion tiene que ser de la misma longitud que el mensaje que quiero encriptar. Esto funciona para un solo mensaje, y cuando yo quiero mandar un mensaje ya tengo que poner algun tipo de cifrado de bloque.

ESQUEMA SEGURO ante CPA

Chosen Plaintext Attack

$$\Pr\left[\Pr ivK^{CPA}_{A,\pi}(n)=1\right] \leq \frac{1}{2} + negl(\ )$$

La diferencia entre CPA (chosen plaintext attack) y Eavesdropping, es que el primero es un ataque activo, ya que tengo un "oraculo" al cual le puedo consultar la encripcion de mensajes de mi eleccion (incluyendo el mismo mensaje, por lo que ya sabemos que tiene que ser no deterministico). Conclusion: ningun esquema deterministico es seguro ante CPA.

Seguridad ante CPA:

Experimento $\Pr ivK^{CPA}_{A,\pi}(n)$

1) $k \leftarrow Gen(n)$
2) El adversario A recibe 1^n y acceso al oráculo $Enc_k(.)$, y emite m_0 y m_1, de igual longitud
3) Se elige un bit aleatorio $b \leftarrow \{0,1\}$. Se calcula $c \leftarrow Enc_k(m_b)$ y se lo entrega a A. (c = challenge ciphertext).
4) A sigue teniendo acceso a través del oráculo a $Enc_k(.)$ y emite un bit b'
5) Si b'=b, la salida del experimento es 1 (ÉXITO). Sino, es 0.

$$\Pr\left[\Pr ivK^{CPA}_{A,\pi}(n)=1\right] \leq \frac{1}{2} + negl(\ )$$

Modos de Operación de BLOCK CIPHERS

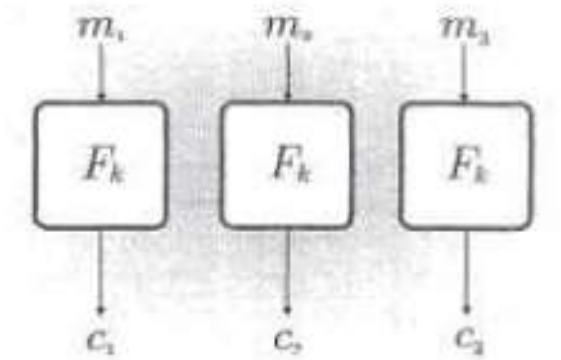
1) ECB – Electronic Code Book Mode

Enc: Se aplica permutación seudoaleatoria sobre cada bloque por separado.

$$m=m_1m_2...m_l$$

$$C=<F_k(m_1), F_k(m_2),... F_k(m_l)>$$

Dec: Se usa F_k^{-1}



ECB es el mas simple de todos. Aplica la permutacion en cada bloque por separado. El problema es que se estaria usando la misma clave para distintos bloques sin seguridad agregada.

➢ No es seguro frente a CPA

2) CBC – Cipher Block Chain Mode

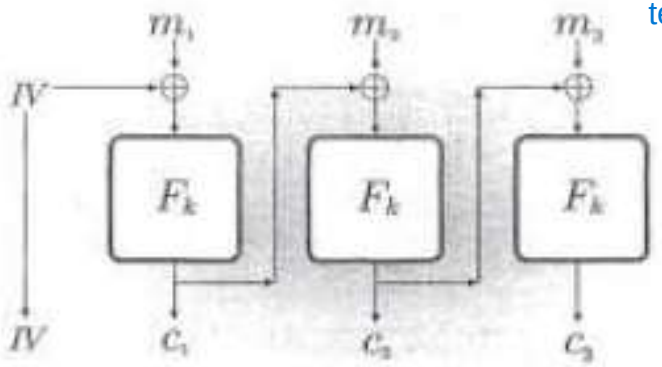
Enc:

$$m=m_1m_2...m_l$$

1. $C_0=IV$
2. $C_i= F_k(C_{i-1}\oplus m_i)$

Mi IV es una secuencia pseudoaleatoria

El IV va en plano.



En CBC, los bloques utilizan las mismas claves, y tambien se agrega una seguridad adicional (IV) que depende del bloque anterior (por lo cual es relativamente aleatorio). La ventaja es que es seguro ante CPA. La desventaja es que es una encripcion secuencial, por lo que tengo que recibir todos los bloques en orden y recibirlos todos.

➢ Es seguro frente a CPA

➢ Encripción Secuencial

3) OFB – Output Feedback Mode

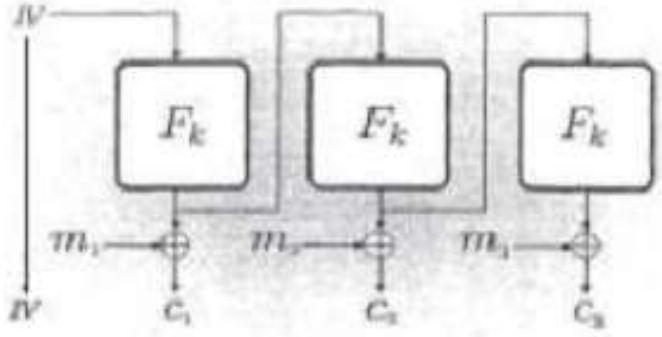
Enc:

$$m=m_1m_2...m_l$$

1. $r_0=IV$
2. $r_i= F_k(r_{i-1})$
3. $C_i= r_i \oplus m_i$

ri es la salida de la function para r(i-1)

El IV va en plano.



En OFB, se toma la salida de la funcion como entrada del siguiente. En el primer bloque, la entrada es el IV, y puedo precalcular m1,m2,m3... por lo que no es una encripcion secuencial y me ahorro esos problemas.

➢ Es seguro frente a CPA.

➢ No requiere F_k biyectiva.

➢ Los ri se pueden calcular antes.

➢ Encripción Secuencial

Modos de Operación de BLOCK CIPHERS (cont.)

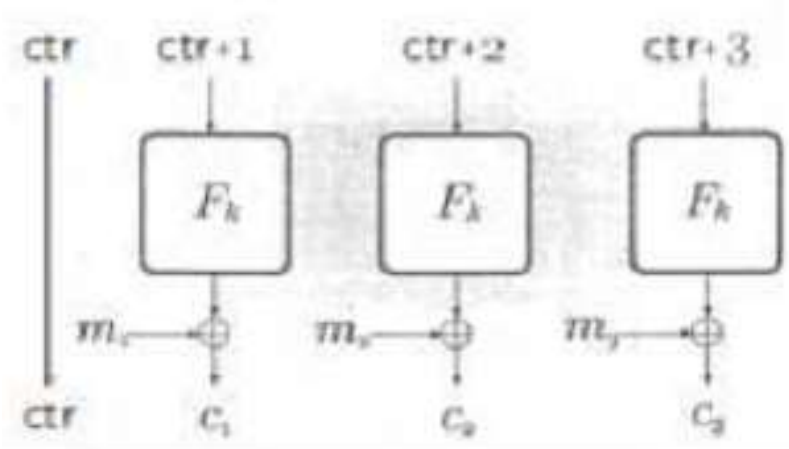
4) CTR – Counter Code Mode

Enc:

$m = m_1 m_2 \ldots m_l$

1. Ctr = IV (counter)
2. $r_i = F_k(Ctr+i)$ (suma modulo 2^n.)
3. $C_i = r_i \oplus m_i$

El counter se elige en forma aleatoria



- Es seguro frente a CPA.
- No requiere Fk biyectiva.
- El stream aleatorio se pueden calcular antes.
- Permite Encripción y desencripción en paralelo
- Permite desencriptar el bloque i ésimo por separado.

CTR es bastante parecido a OFB, pero en este caso los valores van aumentando en cada bloque. Por lo que tengo acceso aleatorio para desencriptar.

5) CFB – Cipher Feedback Mode

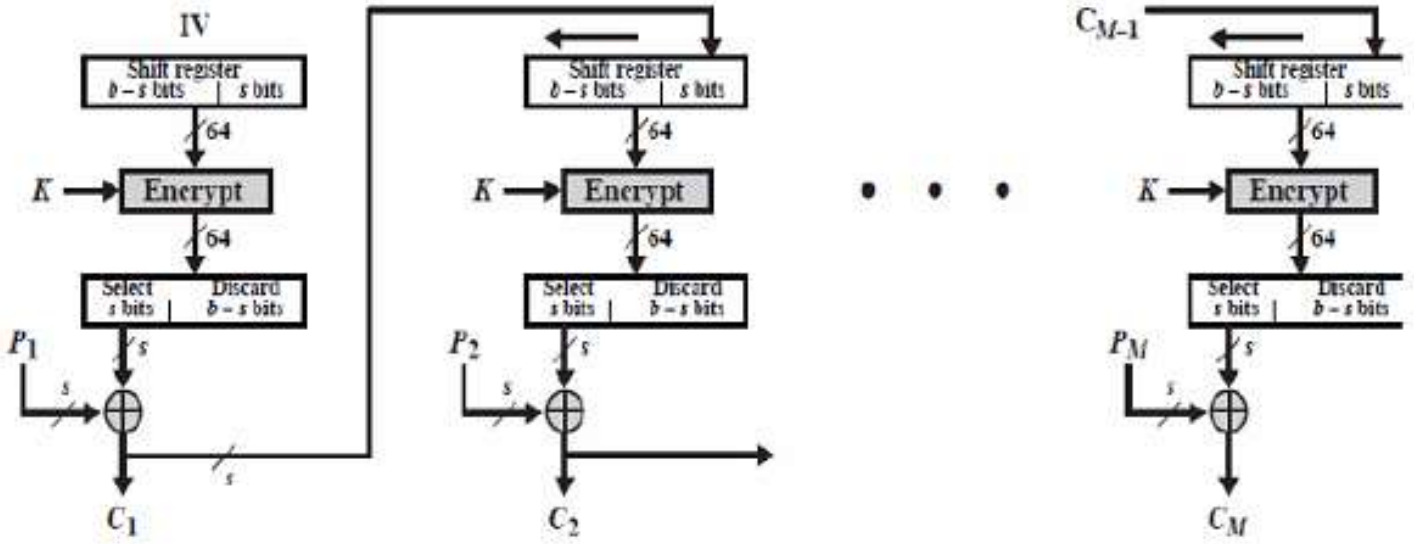
Enc:

$m = m_1 m_2 \ldots m_l$

1. $r_0 = IV$
2. $C_1 = F_k(r_0) >> s \oplus m_1$
3. $r_i = r_{i-1} << s \mid C_{i-1}$
4. $C_i = F_k(r_i) >> s \oplus m_i$

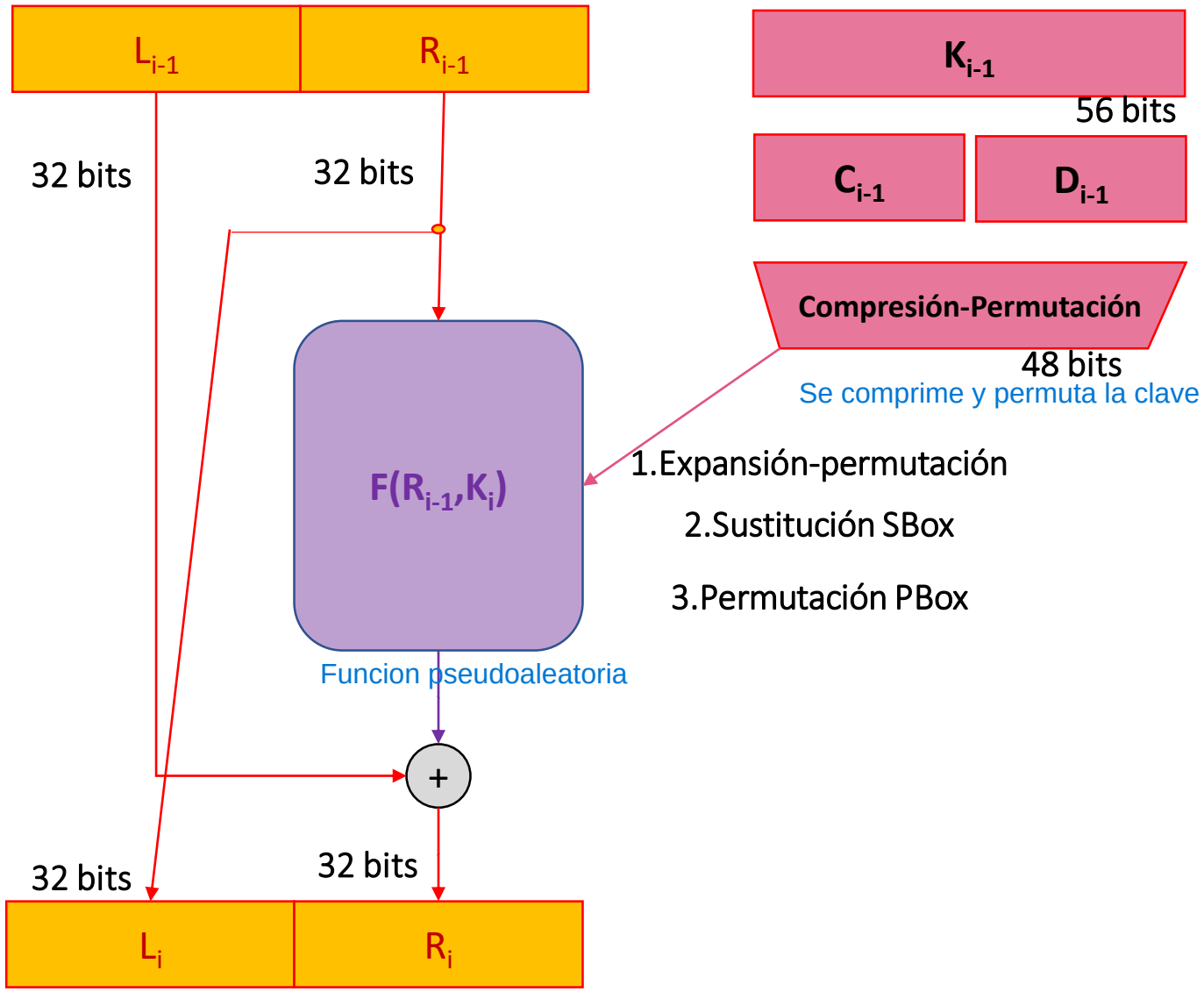
s es el número de bits.

Si s = 1, equivale a un stream cipher.

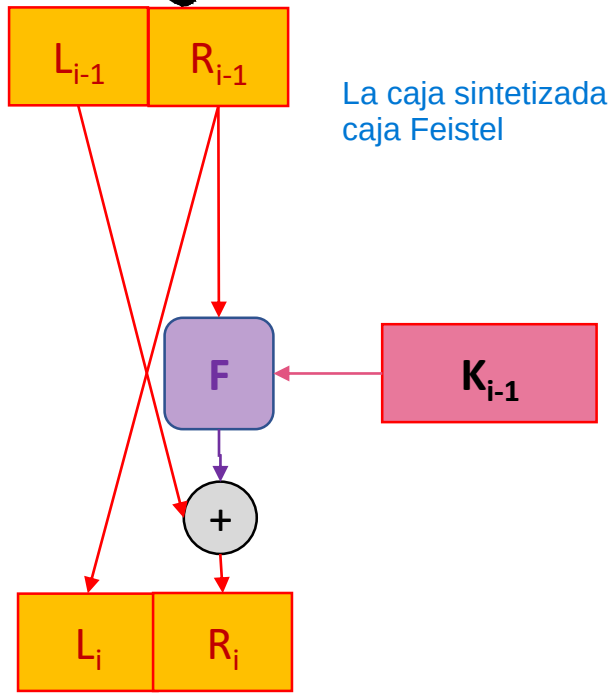


DES = Data encription standard

El mensaje se va a ir transformando en 16 rondas, y en cada ronda va cambiando tanto la clave como el mensaje.



Caja Feistel



La caja sintetizada se llama caja Feistel

Claves Débiles:

Las claves debiles encriptan las 16 rondas igual, por lo que es innecesario hacer 16 rondas.

$E_k(m) = D_k(m)$

o sea:

$E_k(E_k(m)) = m$

(en lugar de generar 16 subclaves distintas, generan 1)

Claves Semidébiles: (vienen de a pares)

$E_{kx}(E_{ky}(m)) = m$

Las claves semidebiles vienen de a pared (ejemplo, las rondas 2, 4, 6 tienen la misma encriptacion y las rondas 3, 5, 7 tambien)

(en lugar de generar 16 subclaves distintas, generan 2 o 4)

Vídeo sobre LFSR

https://www.youtube.com/watch?v=vfq3onw-uiM

Vídeo sobre RC4:

https://www.youtube.com/watch?v=G3HajuqYH2U

Vídeo sobre DES:

https://www.youtube.com/watch?v=XwUOwqSHzyo

Apunte e implementación DES:

En Material Didáctico/Extra/

(yTambién hay una Implementación AES)

Hacer ejercicios 5,6,7,8