Universidad Autónoma de Baja California Sur

Departamento Académico de Sistemas Computacionales

“Diseño e implementación de un algoritmo de encriptación propio”

Por

Olivas Flores Cristhian Iram.

Ochoa Beltran Maria Jose.

Victorio Coronado Jose Carlos.

Criptografía aplicada.

Ingeniería en Desarrollo de Software, séptimo semestre

Julián Ernesto Cadena Vázquez .

La Paz, B.C.S, a 13 de octubre del 2025

1. Resumen ejecutivo....3
2.Motivación y alcance del algoritmo....3
3.Descripción formal del algoritmo....3
Tipo de algoritmo....3
Justificación del diseño....4
4. Especificación matemática y estructuras de datos....4
4.1 Tamaños....4
4.2 S-box dependiente de clave....4
4.3 Key-Schedule....5
4.4 Funciones de ronda....5
4.5 Whitening....5
5. Especificación de la clave....5
Formato....5
Generación....5
Manejo....5
6. Protocolo de cifrado/descifrado....6
6.1 Flujo de cifrado de bloque (SPN)....6
6.2 Flujo de descifrado....6
6.3 Modo CBC....6
6.4 Diagrama de flujo del proceso de cifrado....7
6.5 Diagrama de flujo del proceso de descifrado....11
7. Análisis de seguridad....15
7.1 Objetivos alcanzados....15
7.2 Fortalezas....15
7.3 Debilidades (naturales del diseño académico)....15
8. Plan de pruebas y métricas....16
8.1 Pruebas funcionales....16
8.2 Métricas implementadas....16
8.3 Qué miden....16
9. Referencias (APA)....16

1. Resumen ejecutivo.

CIPHERX-128 es un cifrado simétrico académico implementado desde cero en Python. Trabaja con bloques de 128 bits, clave maestra de 256 bits, 4 rondas y una S-box generada dinámicamente a partir de la clave maestra.
El proyecto implementa:

- Cifrado por bloques CIPHERX-128
- Modos ECB (interno) y CBC (expuesto en CLI)
- Padding PKCS#7
- Escrow opcional mediante RSA OAEP (demo)
- Métricas de entropía, histogramas y avalancha

Su objetivo es educativo: permitir estudiar diseño de cifradores SPN simplificados, key-schedule básico y efectos en seguridad.

2.Motivación y alcance del algoritmo.

El proyecto busca:

- **Explorar la construcción de un cifrador tipo SPN minimalista** (4 rondas, S-box generada por clave).
- **Demostrar modos de operación**, especialmente CBC.
- **Ejercitar técnicas de sandboxing**, gestión de claves e interfaz CLI.
- **Evaluar propiedades estadísticas** (entropía, avalancha).

Alcance:

- Cifrado/descifrado de archivos dentro del sandbox
- CBC con IV aleatorio
  Generación de clave maestra y escrow de recuperación
- Métricas criptoestadísticas
- No influye la autenticación (MAC, AEAD).

3.Descripción formal del algoritmo.

Tipo de algoritmo.

CIPHERX-128 es un **cifrado simétrico por bloques** que combina:

- Esquema SPN (Substitución-Permutación Network)
- Clave dependiente de S-box
- Key-schedule determinístico mediante SHA-256
- Whitening pre y post-ronda

- Modo CBC académico

Justificación del diseño.

- SPN simplifica la implementación en Python y permite medir difusión/avalancha.
- La S-box dependiente de la clave evita patrones fijos.
- SHA-256 asegura key-schedule determinístico y uniforme.
- CBC permite manejar mensajes de longitud arbitraria.

4. Especificación matemática y estructuras de datos.

4.1 Tamaños.

Concepto.	**Valor.**
Tamaño de bloque.	128 bits = 16 bytes
Tamaño de clave.	256 bits = 32 bytes
Rondas.	4
S-box.	Permutación de 256 elementos

4.2 S-box dependiente de clave.

Se genera mediante:

1. `seed = SHA256(master_key)`
2. Uso de un **generador congruencial lineal (LCG)**:
   Xn+1=(aXn+c)mod232X_{n+1} = (a X_n + c) \bmod 2^{32}Xn+1=(aXn+c)mod232
   con a=1664525,c=1013904223a=1664525, c=1013904223a=1664525,c=1013904223
3. Fisher–Yates sobre `[0..255]`

Produce:

- `sbox[x]`: sustitución
- `inv_sbox[sbox[x]] = x`: inversa

4.3 Key-Schedule.

Para cada ronda `i`:

KS(i)=SHA256(master ∥ i)[0:16]KS(i) = \text{SHA256}(\text{master} \| i)[0:16]KS(i)=SHA256(master ∥ i)[0:16]

Se generan **R+2 subclaves** (pre y post-whitening + 4 rondas).

4.4 Funciones de ronda.

Dada una entrada de 16 bytes:

1. **SubBytes:**
   state[i]=sbox[state[i]]state[i] = sbox[state[i]]state[i]=sbox[state[i]]
2. **Permutación P (fija):**
   Permuta las 16 posiciones según la lista `P`.
3. **XOR con subclave:**
   state=state⊕KS(r)state = state \oplus KS(r)state=state⊕KS(r)

4.5 Whitening.

- **Pre-whitening:** `state ^= KS(0)`
- **Post-whitening:** `state ^= KS(R+1)`

5. Especificación de la clave.

Formato.

- 32 bytes generados con `secrets.token_bytes(32)`.

Generación.

- Se crea en `cipherx-cli init` y se guarda en `sandbox/keys/secret.key`.

Manejo.

- Permisos 600 cuando sea posible.
- Cifrado de respaldo en `escrow/recovery.enc` usando RSA-OAEP demo.

6. Protocolo de cifrado/descifrado.

6.1 Flujo de cifrado de bloque (SPN).

```
state = block XOR KS[0]
for r in 1..4:
    state = SubBytes(state)
    state = Permute(state)
    state = state XOR KS[r]
return state XOR KS[5]
```

6.2 Flujo de descifrado.

```
state = block XOR KS[5]
for r in 4..1:
    state = state XOR KS[r]
    state = PermuteInverse(state)
    state = InvSubBytes(state)
return state XOR KS[0]
```

6.3 Modo CBC.

Cifrado:

$C0=IV, Ci=Enc(Pi \oplus Ci-1)$

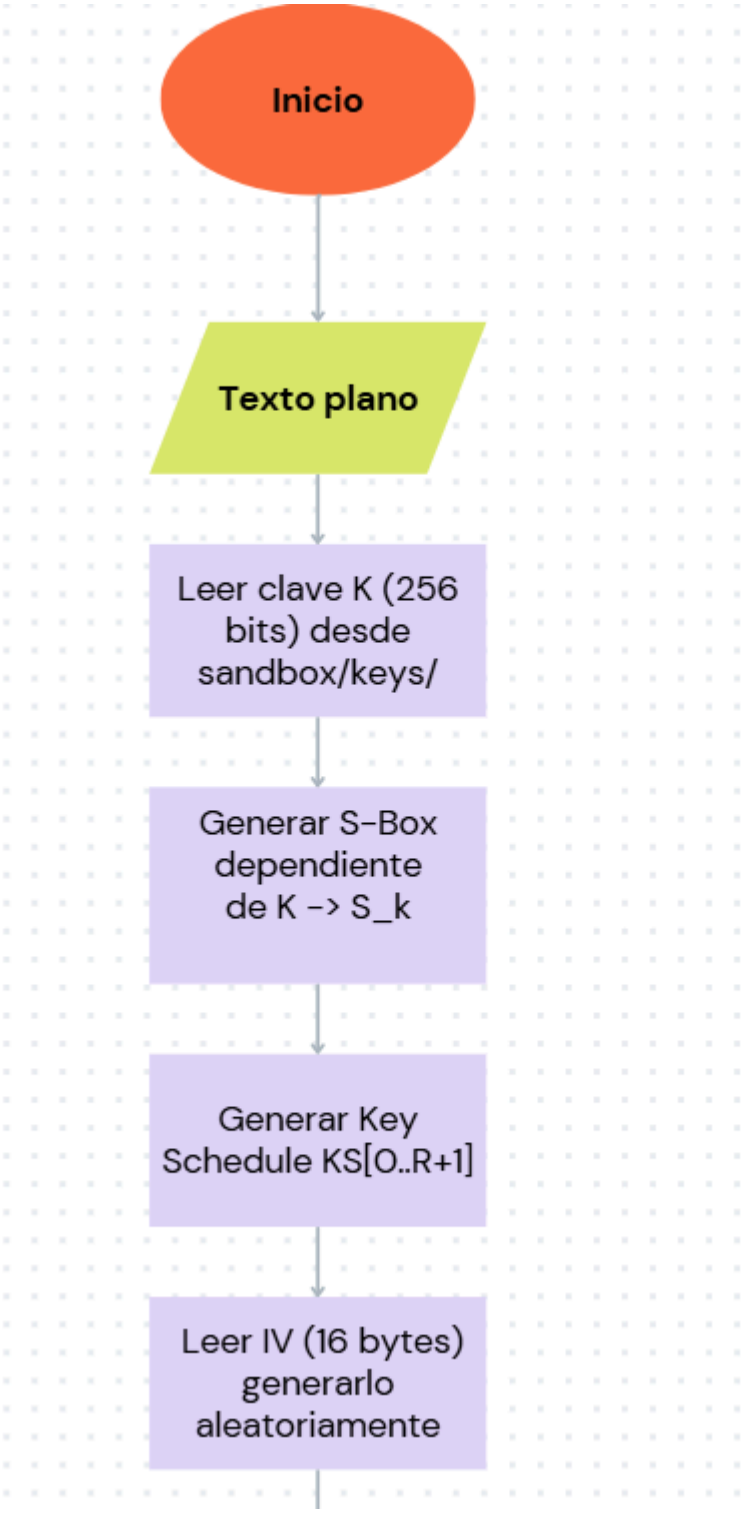
Salida del archivo:

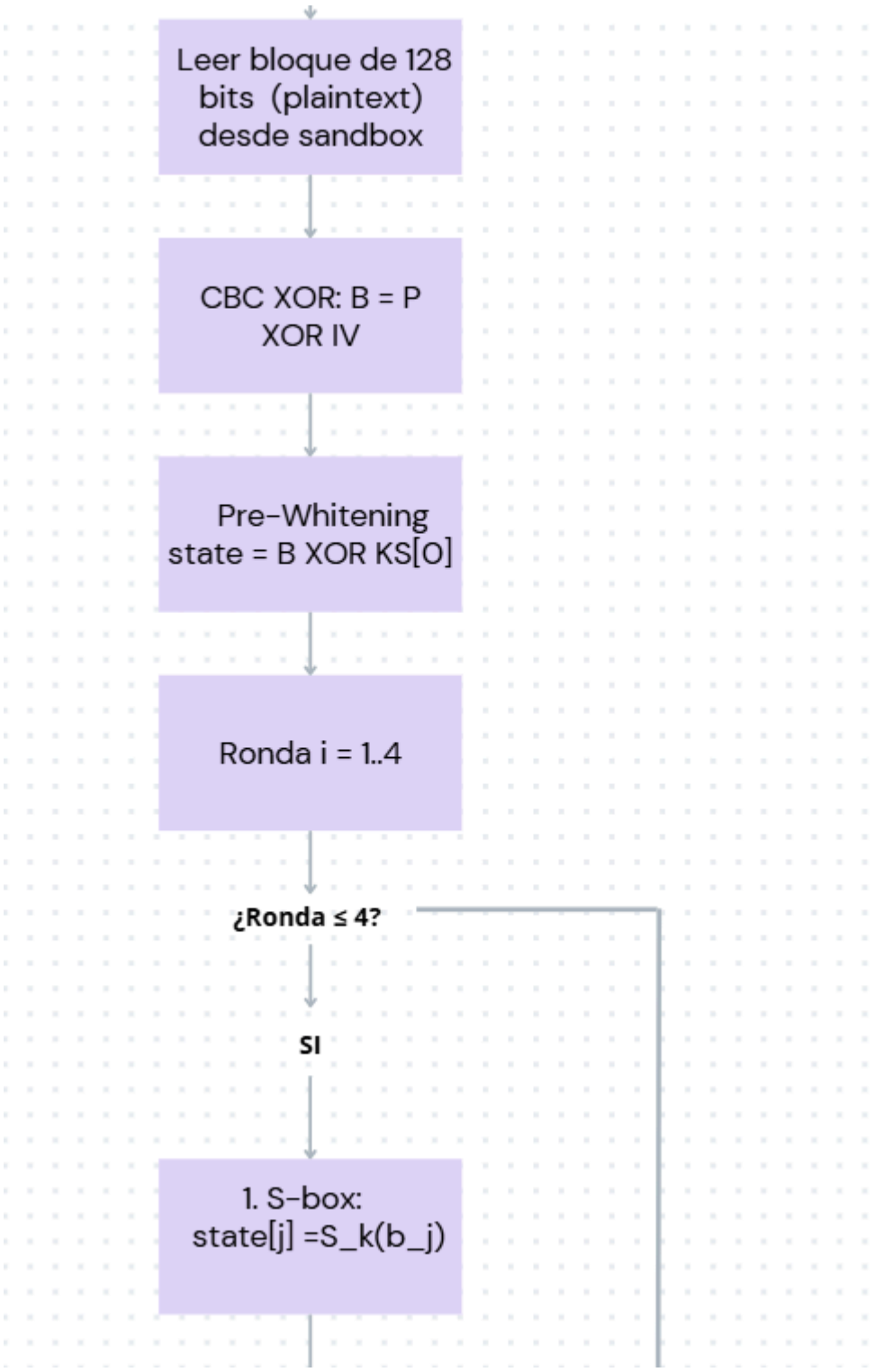
- `IV || C1 || C2 || ... || Cn`

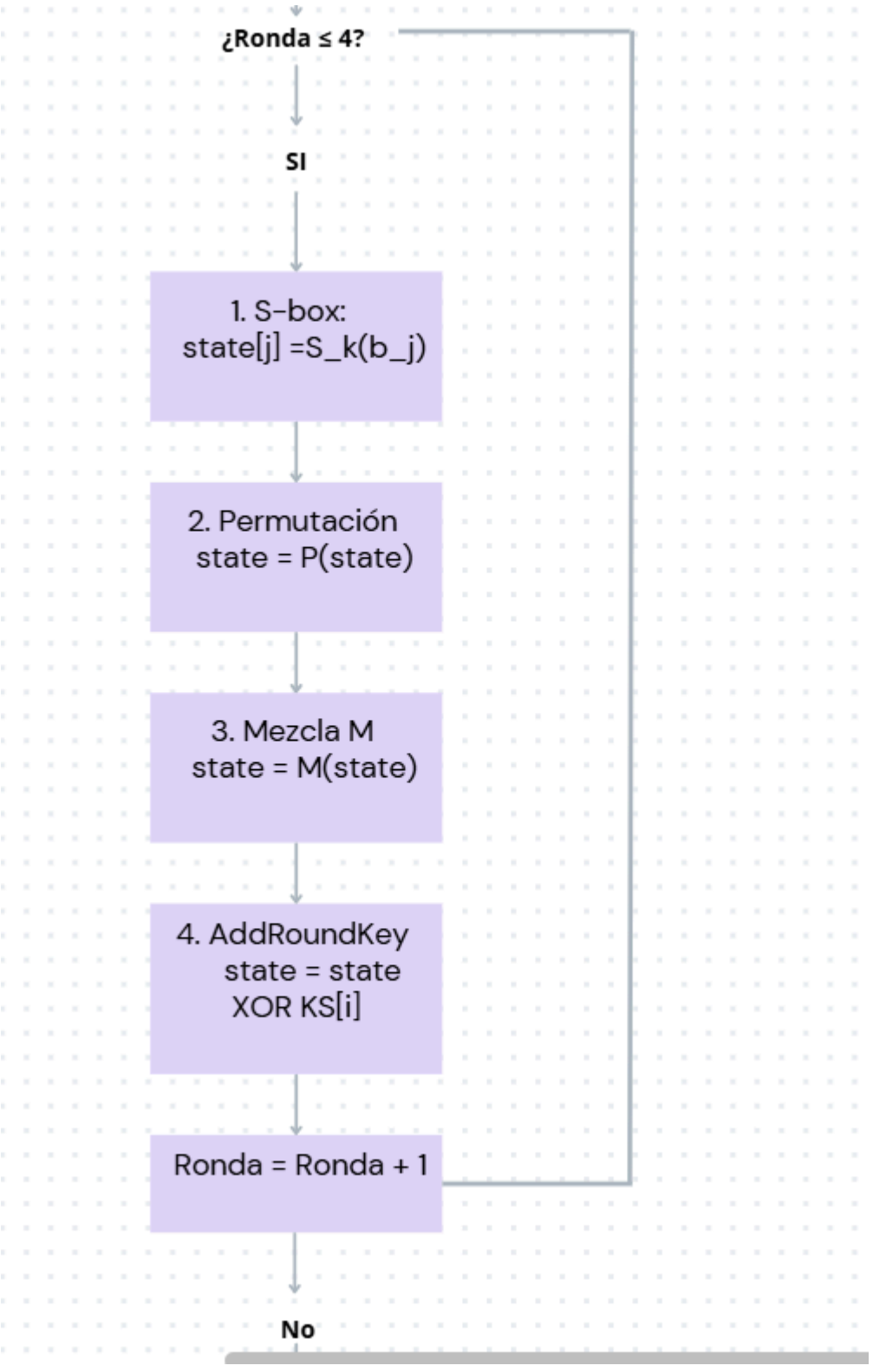
Descifrado:

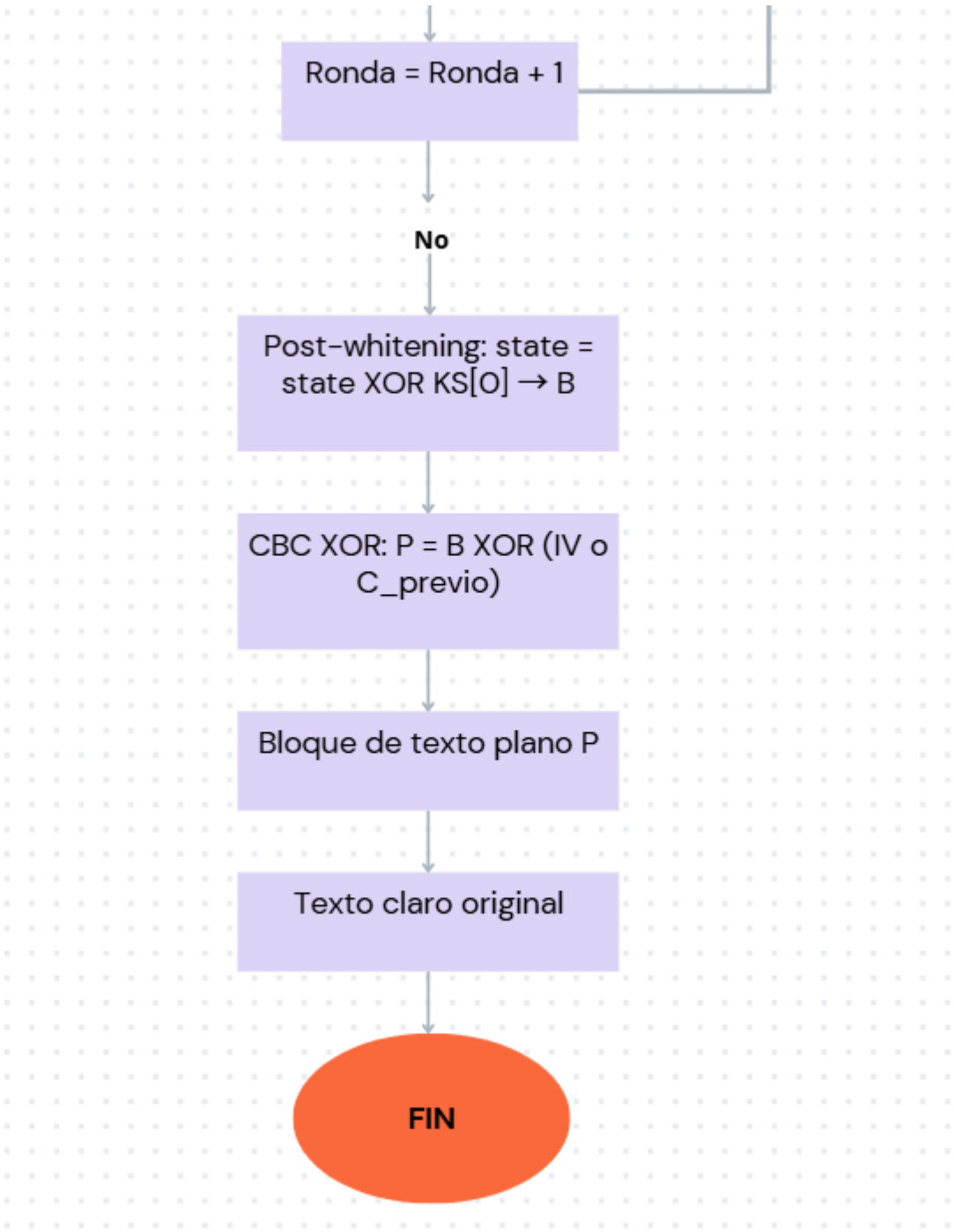
$Pi=Dec(Ci) \oplus Ci-1$

6.4 Diagrama de flujo del proceso de cifrado

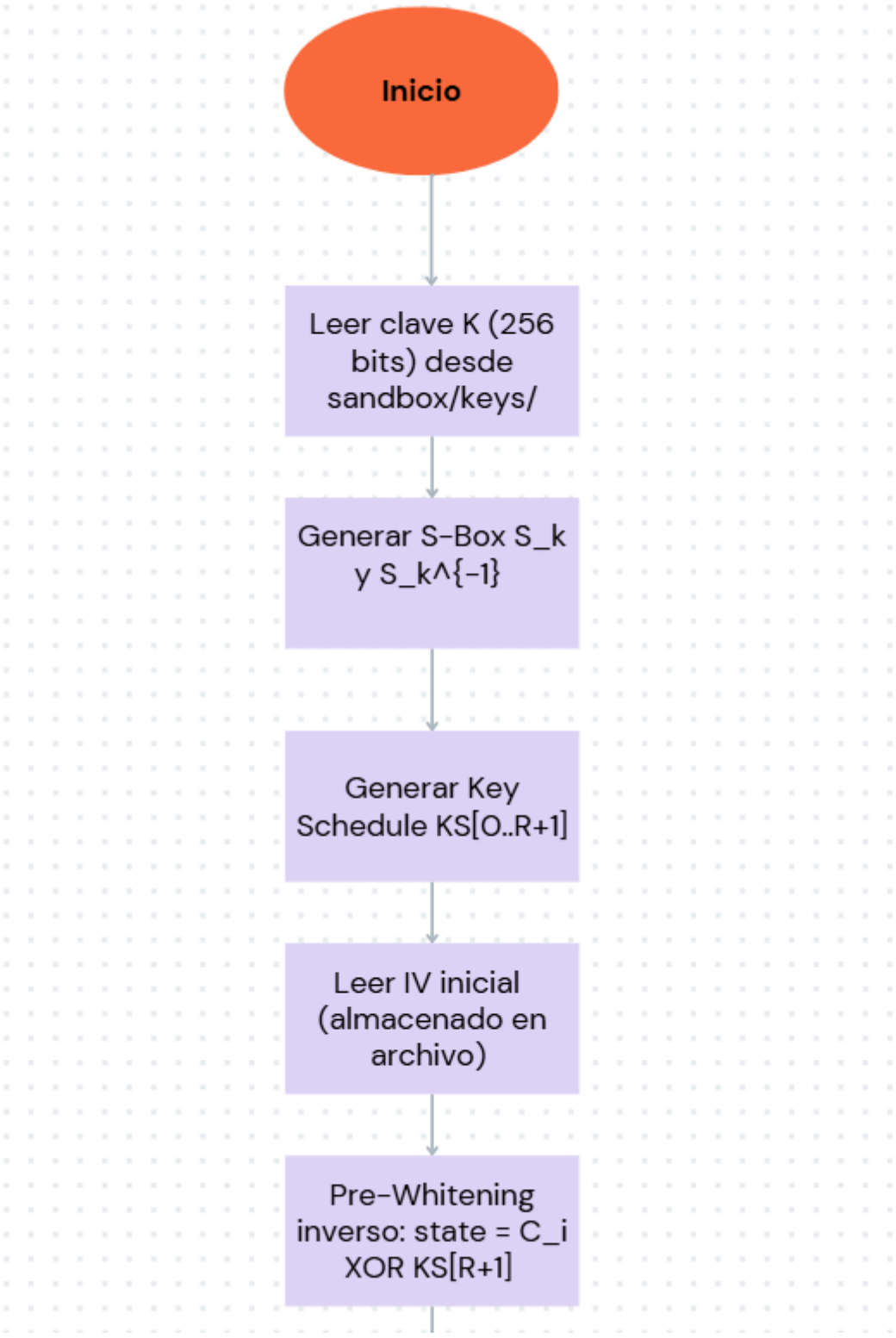


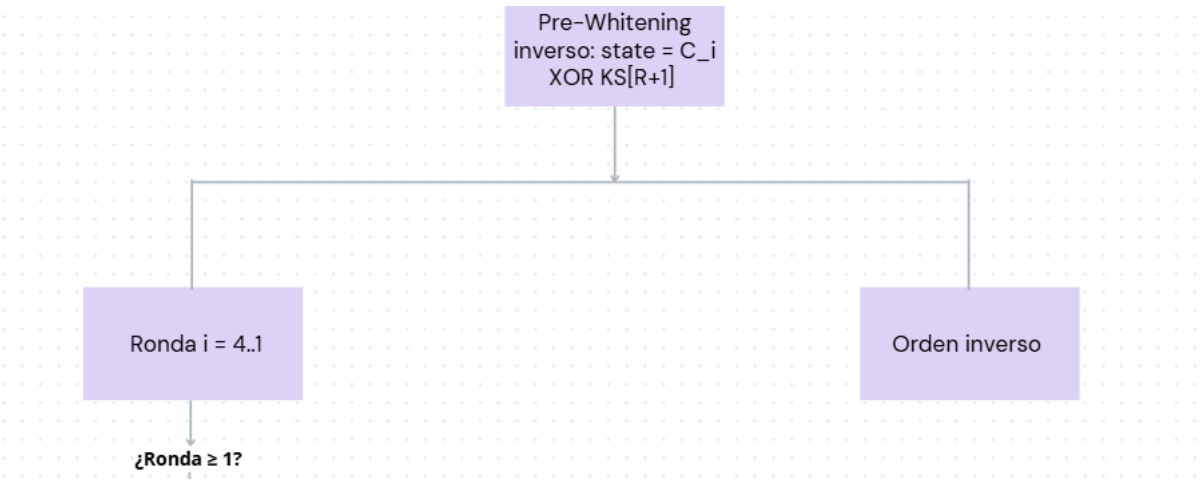

Leer bloque de 128 bits (plaintext) desde sandbox
CBC XOR: B = P XOR IV
Pre-Whitening state = B XOR KS[0]
Ronda i = 1..4
¿Ronda ≤ 4?
SI
1. S-box: state[j] =S_k(b_j)

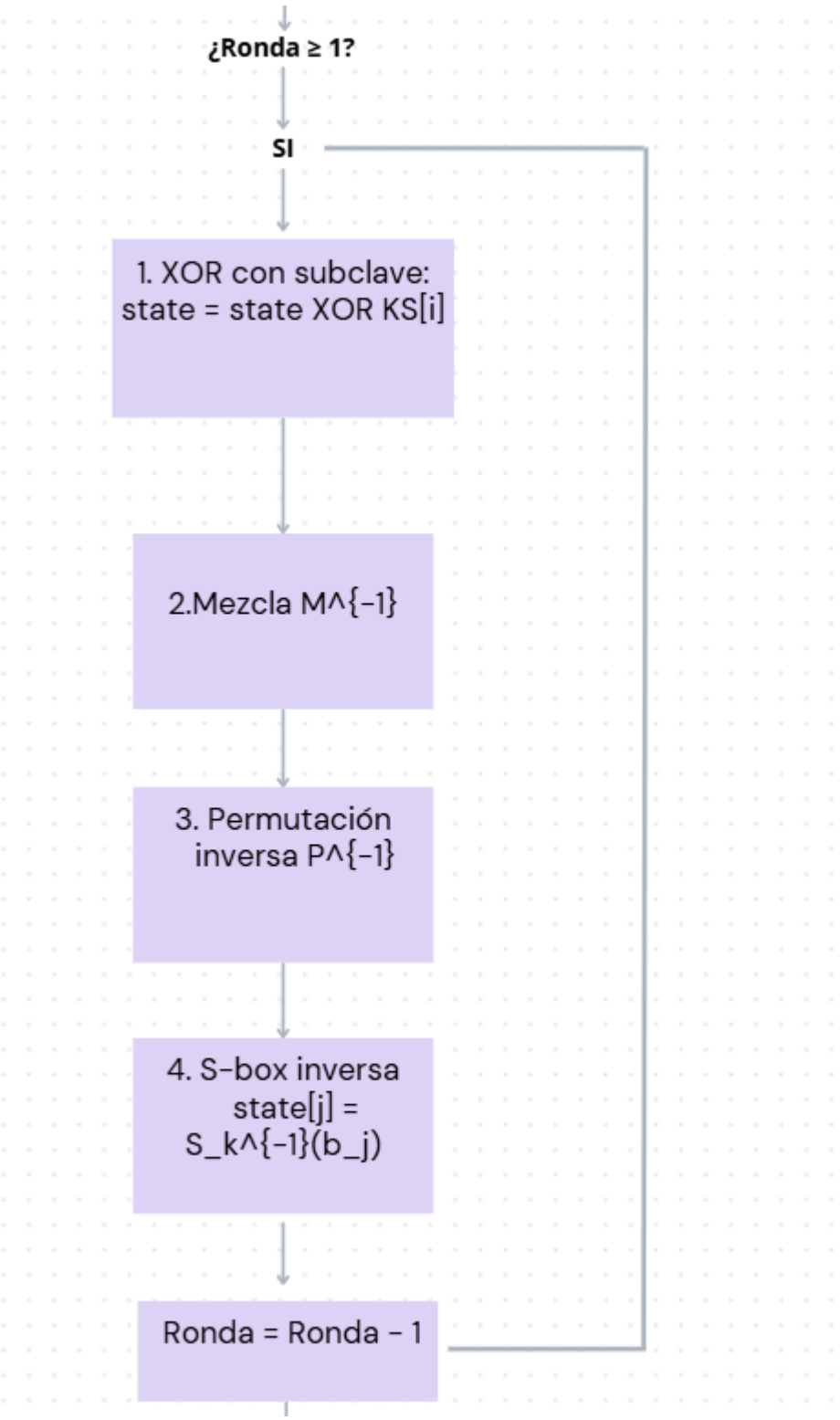

¿Ronda ≤ 4?
SI
1. S-box:
state[j] =S_k(b_j)
2. Permutación
state = P(state)
3. Mezcla M
state = M(state)
4. AddRoundKey
state = state
XOR KS[i]
Ronda = Ronda + 1
No

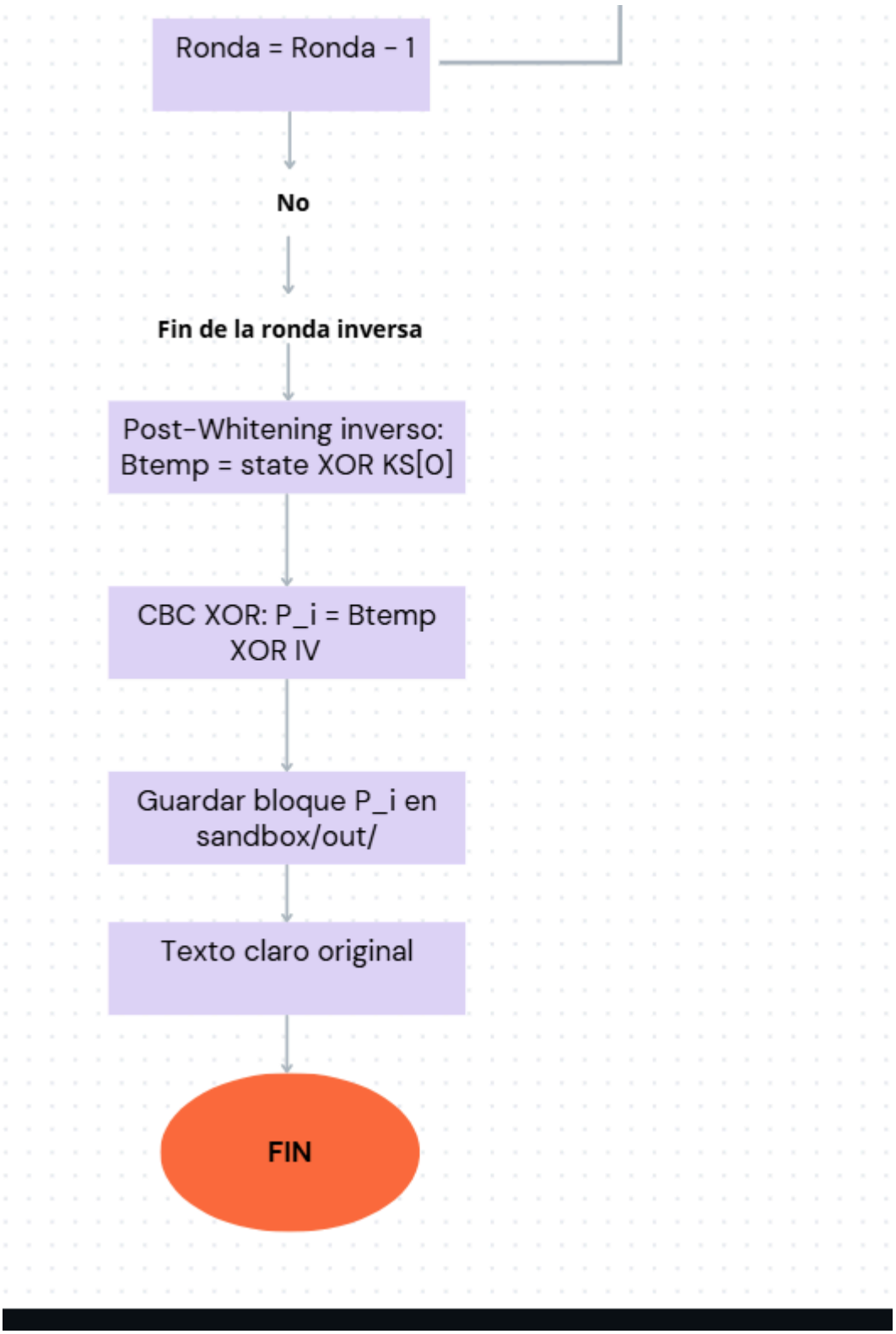

Ronda = Ronda + 1
No
Post-whitening: state = state XOR KS[0] → B
CBC XOR: P = B XOR (IV o C_previo)
Bloque de texto plano P
Texto claro original
FIN

6.5 Diagrama de flujo del proceso de descifrado




Pre-Whitening inverso: state = C_i XOR KS[R+1]
Ronda i = 4..1
Orden inverso
¿Ronda ≥ 1?


¿Ronda ≥ 1?
SI
1. XOR con subclave:
state = state XOR KS[i]
2.Mezcla M^{-1}
3. Permutación
inversa P^{-1}
4. S-box inversa
state[j] =
S_k^{-1}(b_j)
Ronda = Ronda − 1


Ronda = Ronda - 1
No
Fin de la ronda inversa
Post-Whitening inverso:
Btemp = state XOR KS[0]
CBC XOR: P_i = Btemp
XOR IV
Guardar bloque P_i en
sandbox/out/
Texto claro original
FIN

7. Análisis de seguridad.

7.1 Objetivos alcanzados.

Objetivo.	**Cumplimiento.**
Confidencialidad.	Sí (a nivel académico).
Integridad.	No (se sugiere HMAC/AEAD externo).
Autenticación.	No.

7.2 Fortalezas.

- S-box dependiente de clave → dificulta análisis estándar.
- Key-schedule SHA-256 → sin correlaciones triviales.
- Permutación fija invertible → difusión clara.
- Pruebas de avalancha soportadas en `metrics.py`.

7.3 Debilidades (naturales del diseño académico).

- Solo **4 rondas** → bloque SPN muy pequeño comparado con AES-128 (10 rondas).
- S-box generada por un LCG → no garantiza propiedades criptoanalíticas.
- No incluye protección contra ataques de tiempo/side-channels.
- CBC sin autenticación → vulnerable a padding oracle, bit-flipping, etc.
- Implementación en Python → lenta y sin endurecimiento de memoria.

8. Plan de pruebas y métricas.

8.1 Pruebas funcionales.

- `test_avalanche_basic`
- `test_entropy_minimum`
- Tests de sandboxing para rutas seguras.

8.2 Métricas implementadas.

Métrica.	**Módulo.**	**Descripción.**
Avalancha.	`metrics.avalanche_test`	Cambia bits del mensaje y compara salidas.
Entropía.	`metrics.shannon_entropy`	Bits de entropía por byte del ciphertext.
Histograma.	`metrics.byte_histogram`	Distribución estadística de bytes.
Tiempo.	`metrics.time_exec`	Benchmark del cifrado.

8.3 Qué miden.

- **Avalancha:** busca dispersión ideal (~50% bits cambiados).
- **Entropía:** evitar patrones visibles en el ciphertext.
- **Histograma:** uniformidad de distribución.
- **Tiempo:** coste temporal por operación.

9. Referencias (APA).

Daemen, J., & Rijmen, V. (2002). *The Design of Rijndael*. Springer.

Ferguson, N., Schneier, B., & Kohno, T. (2010). *Cryptography Engineering*. Wiley.

Katz, J., & Lindell, Y. (2020). *Introduction to Modern Cryptography* (3rd ed.). CRC Press.

Menezes, A., van Oorschot, P., & Vanstone, S. (1996). *Handbook of Applied Cryptography*. CRC Press.

NIST. (2012). *Recommendation for Block Cipher Modes of Operation* (SP 800-38 series).