Ejemplos de Operaciones XOR, Desplazamiento y Rotación

A continuación unos ejemplos de operaciones a nivel de bits: XOR, desplazamiento hacia la izquierda (<<), desplazamiento hacia la derecha (>>), y rotaciones.

# Ejemplos de XOR

• A (65) XOR B (66): 65 (01000001) XOR 66 (01000010) = 3 (00000011)

• a (97) XOR A (65): 97 (01100001) XOR 65 (01000001) = 32 (00100000)

• M (77) XOR 5: 77 (01001101) XOR 5 (00000101) = 72 (01001000)

• 10101010 XOR 01010101: 10101010 XOR 01010101 = 11111111

• 11110000 XOR 00001111: 11110000 XOR 00001111 = 11111111

Cambia bits de manera "reversible". Si se conoce el valor usado para XOR, puedes recuperar el dato original aplicando XOR otra vez.

UTILIDAD

Excelente como método base: rápido, simple, pero débil si no cambias la clave o si el patrón es simple.

# Ejemplos de Desplazamiento a la Izquierda (<<)

• A (65) << 1: 01000001 << 1 = 10000010 (130)

• b (98) << 2: 01100010 << 2 = 10001000 (136)

• 5 << 3: 00000101 << 3 = 00101000 (40)

• 15 << 2: 00001111 << 2 = 00111100 (60)

• 00110011 << 1: 00110011 << 1 = 01100110

Pierdes bits (rellenas con ceros), lo que puede llevar a **pérdida irreversible** de información si no aplicas mecanismos para recuperar esos bits

UTILIDAD

No es ideal para encriptación sola porque puede destruir datos. Usada en conjunto puede complicar ataques.

# Ejemplos de Desplazamiento a la Derecha (>>)

• C (67) >> 1: 01000011 >> 1 = 00100001 (33)

• d (100) >> 2: 01100100 >> 2 = 00011001 (25)

• 240 >> 3: 11110000 >> 3 = 00011110 (30)

• 128 >> 1: 10000000 >> 1 = 01000000 (64)

• 60 >> 2: 00111100 >> 2 = 00001111 (15)

Pierdes bits (rellenas con ceros), lo que puede llevar a **pérdida irreversible** de información si no aplicas mecanismos para recuperar esos bits

UTILIDAD

No es ideal para encriptación sola porque puede destruir datos. Usada en conjunto puede complicar ataques.

# Ejemplos de Rotaciones

• Rotación izquierda de 1 bit: A (65): 01000001 -> 10000010 (130)

• Rotación derecha de 2 bits: B (66): 01000010 -> 10010000 (144)

• Rotación izquierda de 3 bits: 5: 00000101 -> 00101000 (40)

• Rotación derecha de 1 bit: 240: 11110000 -> 01111000 (120)

• Rotación izquierda de 2 bits: 60: 00111100 -> 11110000 (240)

No destruye información. Solo reordena los bits, por lo que **puede revertirse** si sabes cuánto se rotó y en qué dirección.

UTILIDAD

Añade confusión (confusión) al dato, útil para dificultar el análisis. Sola no es segura, pero combinada mejora la protección.