program for ECB mode, if there is an error in a block of the transmitted ciphertext, only the corresponding plaintext block is affected. However, in the CBC mode, this error propagates. For example, an error in the transmitted C1 obviously corrupts P1 and P2. a. Are any blocks beyond P2 affected? b. Suppose that there is a bit error in the source version of P1. Through how many ciphertext blocks is this error propagated? What is the effect at the receiver?

# Simple ECB vs CBC error propagation demo (no external libraries)

def xor\_bytes(a, b):

return bytes([x ^ y for x, y in zip(a, b)])

def simple\_encrypt\_block(block, key):

"""Tiny fake block cipher (for demo only)"""

return bytes([(b + key[i % len(key)]) % 256 for i, b in enumerate(block)])

def simple\_decrypt\_block(block, key):

"""Inverse of simple\_encrypt\_block"""

return bytes([(b - key[i % len(key)]) % 256 for i, b in enumerate(block)])

def pad(data):

pad\_len = 8 - (len(data) % 8)

return data + bytes([pad\_len]) \* pad\_len

def unpad(data):

pad\_len = data[-1]

return data[:-pad\_len]

def ecb\_encrypt(data, key):

data = pad(data)

ciphertext = b""

for i in range(0, len(data), 8):

block = data[i:i+8]

ciphertext += simple\_encrypt\_block(block, key)

return ciphertext

def ecb\_decrypt(data, key):

plaintext = b""

for i in range(0, len(data), 8):

block = data[i:i+8]

plaintext += simple\_decrypt\_block(block, key)

return unpad(plaintext)

def cbc\_encrypt(data, key, iv):

data = pad(data)

ciphertext = b""

prev = iv

for i in range(0, len(data), 8):

block = xor\_bytes(data[i:i+8], prev)

enc = simple\_encrypt\_block(block, key)

ciphertext += enc

prev = enc

return ciphertext

def cbc\_decrypt(data, key, iv):

plaintext = b""

prev = iv

for i in range(0, len(data), 8):

block = data[i:i+8]

dec = simple\_decrypt\_block(block, key)

plaintext += xor\_bytes(dec, prev)

prev = block

return unpad(plaintext)

# --- Main Program ---

key = b"key12345"

iv = b"initvect"

plaintext = input("Enter plaintext: ").encode()

# Encrypt in ECB and CBC modes

ecb\_cipher = ecb\_encrypt(plaintext, key)

cbc\_cipher = cbc\_encrypt(plaintext, key, iv)

# Introduce an error in 1st ciphertext block

ecb\_error = bytearray(ecb\_cipher)

cbc\_error = bytearray(cbc\_cipher)

ecb\_error[0] ^= 1 # flip 1 bit

cbc\_error[0] ^= 1 # flip 1 bit

# Decrypt both

ecb\_decrypted = ecb\_decrypt(bytes(ecb\_error), key)

cbc\_decrypted = cbc\_decrypt(bytes(cbc\_error), key, iv)

print("\n=== ECB Mode ===")

print("Decrypted text:", ecb\_decrypted.decode(errors='ignore'))

print("Only the first block affected.")

print("\n=== CBC Mode ===")

print("Decrypted text:", cbc\_decrypted.decode(errors='ignore'))

print("Error propagates: first TWO blocks affected.")

# --- Explanations ---

print("\n(a) Beyond P2? → No. Only P1 and P2 are affected.")

print("(b) Bit error in source P1 affects → Only ciphertext block C1.")

print(" Receiver sees corrupted P1 and P2 when decrypting CBC.")

