program for set of blocks encoded with the RSA algorithm and we don’t have the private key. Assume n = pq, e is the public key. Suppose also someone tells us they know one of the plaintext blocks has a common factor with n. Does this help us in any way?

# Simple RSA GCD attack demo

# If any plaintext block shares a factor with n, we can recover p and q.

import math

# Given RSA public values

n = 3599 # n = p \* q

e = 31 # public exponent

# Suppose we somehow know a plaintext block that shares a factor with n

plaintext\_block = 59 # deliberately shares factor with n (since n = 59 \* 61)

# Step 1: Compute GCD

p = math.gcd(plaintext\_block, n)

if p != 1 and p != n:

q = n // p

print(f"Common factor found! p = {p}, q = {q}")

# Step 2: Compute phi(n)

phi = (p - 1) \* (q - 1)

# Step 3: Compute private key 'd' using modular inverse

def mod\_inverse(a, m):

for x in range(1, m):

if (a \* x) % m == 1:

return x

return None

d = mod\_inverse(e, phi)

print(f"Private key d = {d}")

else:

print("No common factor found. Attack failed.")

