

COMSATS University Islamabad Attock Campus

Assignment 2

SE-4

Submitted BY: Sheema Qazi

Registration No: SP24-BSE-012

Submitted To: Ma`am Ambreen gul

Course: IS

Submission Date: 12 October,2025.

Graded Lab Task 1:

You have implemented DES there is built in implemented DES in python in crypto cipher module use it for encryption/decryption and provide output sample.

**Output:**

****

Graded Task 2

Visualization of MITM Attack Flow:

Attacker intercepts plaintext (P) and ciphertext (C)1. Guess K1P --[Encrypt with K1]--> Intermediate Value 1 (I1)2. Guess K2C --[Decrypt with K2]--> Intermediate Value 2 (I2)3. If I1 == I2, then K1 and K2 are likely correct.Found the DES key: K = K1 + K2………………………………………………….

**Code:**

import random  
from collections import defaultdict  
  
MOD = 256  
KEY\_BITS = 5  
  
def E(k, m): return (m + k) % MOD  
def D(k, c): return (c - k) % MOD  
def double\_E(k1, k2, m): return E(k2, E(k1, m))  
  
def mitm(P, C, key\_bits=KEY\_BITS):  
 maxk = 1 << key\_bits  
 table = defaultdict(list)  
 for k1 in range(maxk):  
 table[E(k1, P)].append(k1)  
 candidates = []  
 for k2 in range(maxk):  
 i2 = D(k2, C)  
 if i2 in table:  
 for k1 in table[i2]:  
 candidates.append((k1, k2))  
 return candidates  
  
def demo(key\_bits=KEY\_BITS, seed=None):  
 if seed is not None:  
 random.seed(seed)  
 maxk = 1 << key\_bits  
 K1 = random.randrange(maxk)  
 K2 = random.randrange(maxk)  
 P = random.randrange(MOD)  
 C = double\_E(K1, K2, P)  
  
 print("=== MITM Demo ===")  
 print(f"(hidden) True keys: K1={K1}, K2={K2}")  
 print(f"Known pair: P={P}, C={C}\n")  
  
 candidates = mitm(P, C, key\_bits)  
 print(f"Found {len(candidates)} candidate pairs:")  
 for i, (k1, k2) in enumerate(sorted(candidates), 1):  
 print(f" {i:2d}. K1={k1}, K2={k2}, I={E(k1,P)}")  
  
 # verify with a second plaintext  
 P2 = random.randrange(MOD)  
 C2 = double\_E(K1, K2, P2)  
 verified = [(k1,k2) for (k1,k2) in candidates if double\_E(k1,k2,P2) == C2]  
 print(f"\nVerified candidates: {verified}")  
 if (K1,K2) in verified:  
 print("SUCCESS: true key found and verified.")  
 else:  
 print("True key not found among verified candidates.")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 demo(seed=42)

**Output:**

