**Pratical 4: GSM(A3,A5,A8)**

**Program:**

import random

# Constants

KEY\_SIZE = 16 # 128-bit key (Ki)

RAND\_SIZE = 16 # 128-bit RAND challenge

Kc\_SIZE = 8 # 64-bit cipher key for A5

# Generate a random 128-bit challenge

def generate\_rand():

return [random.randint(0, 255) for \_ in range(RAND\_SIZE)]

# A3 Algorithm - Generate SRES (simplified)

def A3(Ki, RAND):

# XOR Ki and RAND byte-wise then sum modulo 2^32 to simulate a signature

xor\_result = [Ki[i] ^ RAND[i] for i in range(KEY\_SIZE)]

sres = sum(xor\_result) % (2\*\*32)

return sres

# A8 Algorithm - Generate cipher key Kc (simplified)

def A8(Ki, RAND):

# Rotate Ki bytes and XOR with RAND to generate Kc

rotated\_ki = Ki[1:] + Ki[:1]

Kc = [rotated\_ki[i] ^ RAND[i] for i in range(Kc\_SIZE)]

return Kc

# A5 Algorithm - Stream cipher (XOR with key stream)

def A5\_encrypt(plaintext\_bytes, Kc):

ciphertext = []

key\_len = len(Kc)

for i, byte in enumerate(plaintext\_bytes):

ciphertext.append(byte ^ Kc[i % key\_len])

return ciphertext

def A5\_decrypt(ciphertext\_bytes, Kc):

# Symmetric XOR decryption

return A5\_encrypt(ciphertext\_bytes, Kc)

# Helpers to convert between string and bytes

def str\_to\_bytes(s):

return [ord(c) for c in s]

def bytes\_to\_str(b):

return ''.join(chr(byte) for byte in b)

# Main demonstration

if \_\_name\_\_ == "\_\_main\_\_":

# Simulated secret key Ki

Ki = [random.randint(0, 255) for \_ in range(KEY\_SIZE)]

print(f"Secret Key (Ki): {Ki}")

# Network generates RAND challenge

RAND = generate\_rand()

print(f"Random Challenge (RAND): {RAND}")

# Mobile Station computes SRES using A3

SRES\_MS = A3(Ki, RAND)

print(f"Mobile Station SRES: {SRES\_MS}")

# Network computes SRES to verify MS

SRES\_NW = A3(Ki, RAND)

print(f"Network SRES: {SRES\_NW}")

if SRES\_MS == SRES\_NW:

print("Authentication Successful.")

else:

print("Authentication Failed.")

# Both compute cipher key Kc using A8

Kc = A8(Ki, RAND)

print(f"Cipher Key (Kc): {Kc}")

# Sample plaintext message

plaintext = "HELLO GSM"

plaintext\_bytes = str\_to\_bytes(plaintext)

print(f"Plaintext: {plaintext}")

# Encrypt message using A5

ciphertext = A5\_encrypt(plaintext\_bytes, Kc)

print(f"Ciphertext bytes: {ciphertext}")

# Decrypt message using A5

decrypted\_bytes = A5\_decrypt(ciphertext, Kc)

decrypted\_text = bytes\_to\_str(decrypted\_bytes)

print(f"Decrypted Text: {decrypted\_text}")

**Output:**

Secret Key (Ki): [212, 178, 134, 37, 96, 180, 110, 201, 14, 218, 72, 211, 35, 62, 53, 231]

Random Challenge (RAND): [175, 131, 241, 198, 60, 59, 111, 13, 109, 19, 145, 2, 47, 63, 32, 5]

Mobile Station SRES: 1936

Network SRES: 1936

Authentication Successful.

Cipher Key (Kc): [29, 5, 212, 166, 136, 85, 166, 3]

Plaintext: HELLO GSM

Ciphertext bytes: [85, 64, 152, 234, 199, 117, 225, 80, 80]

Decrypted Text: HELLO GSM