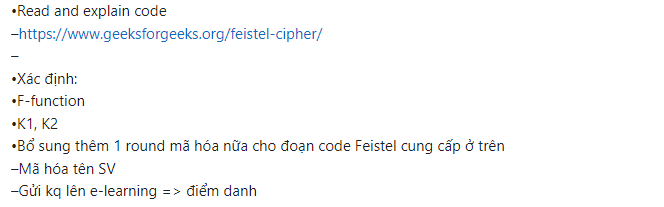
Họ và Tên : Nguyễn Ái Phát

Mã Sinh Viên : 21IT221

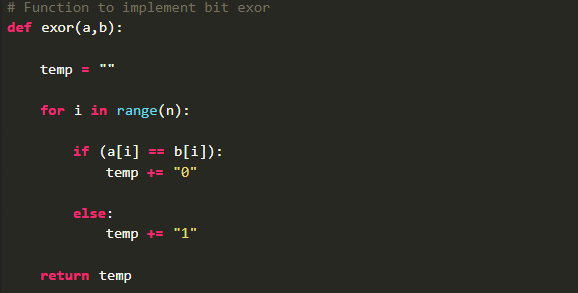
Lớp : 21NS

Môn Học : Mật Mã Học (1)

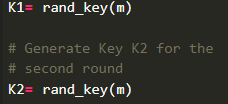
**Bài Tập**



F-function



Khóa K1 và K2



# Python program to demonstrate

# Feistel Cipher Algorithm

import binascii

# Random bits key generation

def rand\_key(p):

    import random

    key1 = ""

    p = int(p)

    for i in range(p):

        temp = random.randint(0, 1)

        temp = str(temp)

        key1 = key1 + temp

    return key1

# Function to implement bit exor

def exor(a, b):

    temp = ""

    for i in range(len(a)):  # Sửa từ n thành len(a) để sử dụng đúng giá trị

        if a[i] == b[i]:

            temp += "0"

        else:

            temp += "1"

    return temp

# Defining BinarytoDecimal() function

def BinaryToDecimal(binary):

    # Using int function to convert to string

    string = int(binary, 2)

    return string

# Feistel Cipher

PT = "PHAT"

print("Plain Text is:", PT)

# Converting the plain text to ASCII

PT\_Ascii = [ord(x) for x in PT]

# Converting the ASCII to 8-bit binary format

PT\_Bin = [format(y, '08b') for y in PT\_Ascii]

PT\_Bin = "".join(PT\_Bin)

n = int(len(PT\_Bin) // 2)

L1 = PT\_Bin[0:n]

R1 = PT\_Bin[n::]

m = len(R1)

# Generate Key K1 for the first round

K1 = rand\_key(m)

# Generate Key K2 for the second round

K2 = rand\_key(m)

# Generate Key K3 for the third round

K3 = rand\_key(m)

# First round of Feistel

f1 = exor(R1, K1)

R2 = exor(f1, L1)

L2 = R1

# Second round of Feistel

f2 = exor(R2, K2)

R3 = exor(f2, L2)

L3 = R2

# Third round of Feistel

f3 = exor(R3, K3)

R4 = exor(f3, L3)

L4 = R3

# Cipher text

bin\_data = L4 + R4

str\_data = ''

for i in range(0, len(bin\_data), 7):

    # Slicing the bin\_data from index range [0, 6]

    temp\_data = bin\_data[i:i + 7]

    # Passing temp\_data in BinarytoDecimal() function to get decimal value

    decimal\_data = BinaryToDecimal(temp\_data)

    # Decoding the decimal value returned by BinarytoDecimal() function

    # Using chr() function which returns the string corresponding

    # character for given ASCII value, and store it in str\_data

    str\_data = str\_data + chr(decimal\_data)

print("Cipher Text:", str\_data)

# Decryption

L5 = L4

R5 = R4

# Third round of Decryption

f4 = exor(L5, K3)

L6 = exor(R5, f4)

R6 = L5

# Second round of Decryption

f5 = exor(L6, K2)

L7 = exor(R6, f5)

R7 = L6

# First round of Decryption

f6 = exor(L7, K1)

L8 = exor(R7, f6)

R8 = L7

# Combine decrypted binary data

PT1 = L8 + R8

# Convert binary data to plain text

PT1 = int(PT1, 2)

RPT = binascii.unhexlify('%x' % PT1)

print("Retrieved Plain Text is:", RPT)

kết quả : 