Name: Ayush Patel   Enrolment 22162171038   Class B Batch 55

Practical – 6

Subject – Crypto

Aim

Alice wants to send some confidential information to Bob over a secure network.

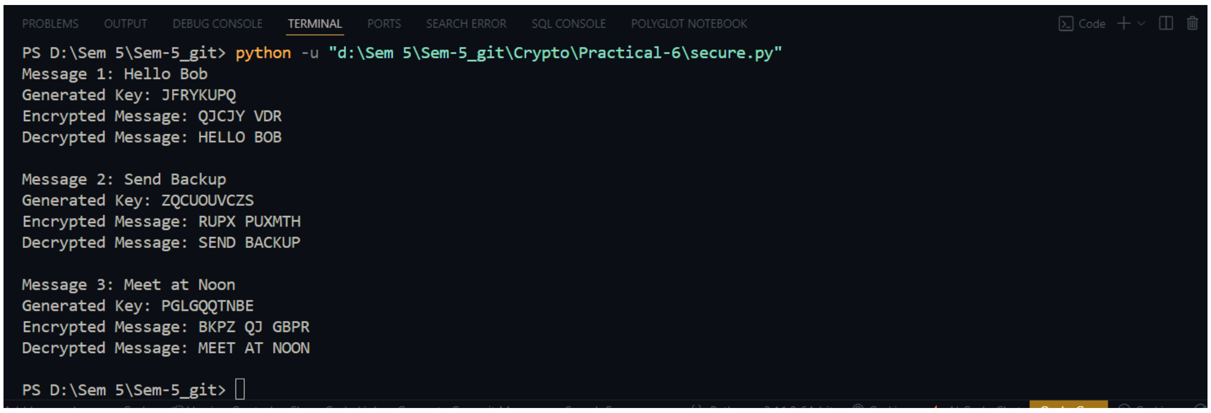
i) Create a system where the key will be generated randomly for encryption, and it will be changed with every message. Send three messages from sender to receiver and also decrypt the message at receiver end.

Code:

```
import random

import string


# Function to generate a random key for each message

def generate_random_key(length):

    return ''.join(random.choices(string.ascii_uppercase, k=length))


# Function to encrypt message using a random key

def encrypt_message(message, key):

    encrypted_message = ""

    key_index = 0  # Separate index for key to handle spaces in
message

    for char in message:
        if char.isalpha():
            shift = ord(key[key_index]) - ord('A')
            encrypted_char = chr((ord(char.upper()) - ord('A') + shift) %
26 + ord('A'))
            encrypted_message += encrypted_char
            key_index += 1  # Only move to next key character if message
character is a letter
        else:
            encrypted_message += char
    return encrypted_message


# Function to decrypt message using the same key
def decrypt_message(encrypted_message, key):
    decrypted_message = ""
    key_index = 0  # Separate index for key to handle spaces in
message
    for char in encrypted_message:
        if char.isalpha():
            shift = ord(key[key_index]) - ord('A')
            decrypted_char = chr((ord(char) - ord('A') - shift) % 26 +
ord('A'))
            decrypted_message += decrypted_char
            key_index += 1  # Only move to next key character if message
character is a letter
        else:
            decrypted_message += char
    return decrypted_message


# Example usage for 3 messages
messages = ["Hello Bob", "Send Backup", "Meet at Noon"]
# Sending 3 messages
for i, message in enumerate(messages):
    key = generate_random_key(len(message.replace(" ", "")))  # Generate random key ignoring spaces
    encrypted_message = encrypt_message(message, key)  # Encrypt the message
    decrypted_message = decrypt_message(encrypted_message, key)  # Decrypt the message

    print(f"Message {i+1}: {message}")
    print(f"Generated Key: {key}")
    print(f"Encrypted Message: {encrypted_message}")
    print(f"Decrypted Message: {decrypted_message}\n")
```

Name: Ayush Patel   Enrolment 22162171038   Class B Batch 55

Output:



```
PS D:\Sem 5\Sem-5_git> python -u "d:\Sem 5\Sem-5_git\Crypto\Practical-6\secure.py"
Message 1: Hello Bob
Generated Key: JFRYKUPQ
Encrypted Message: QJCJY VDR
Decrypted Message: HELLO BOB

Message 2: Send Backup
Generated Key: ZQCUOUVCZS
Encrypted Message: RUPX PUXMTH
Decrypted Message: SEND BACKUP

Message 3: Meet at Noon
Generated Key: PGLGQQTNBE
Encrypted Message: BKPZ QJ GBPR
Decrypted Message: MEET AT NOON

PS D:\Sem 5\Sem-5_git> 
```

ii) Provide encryption through vigener table as well. (Use Second Method)

code:

```
# Function for Vigenère Cipher Encryption
def vigenere_encrypt(plain_text, key):
    key = key.upper()
    encrypted_text = []
    for i in range(len(plain_text)):
        if plain_text[i].isalpha():
            shift = ord(key[i % len(key)]) - ord('A')
            encrypted_char = chr((ord(plain_text[i].upper())
- ord('A') + shift) % 26 + ord('A'))
            encrypted_text.append(encrypted_char)
        else:
            encrypted_text.append(plain_text[i])
    return ''.join(encrypted_text)

# Function for Vigenère Cipher Decryption
def vigenere_decrypt(encrypted_text, key):
    key = key.upper()
    decrypted_text = []
    for i in range(len(encrypted_text)):
        if encrypted_text[i].isalpha():
            shift = ord(key[i % len(key)]) - ord('A')
            decrypted_char = chr((ord(encrypted_text[i]) -
ord('A') - shift) % 26 + ord('A'))
            decrypted_text.append(decrypted_char)
        else:
            decrypted_text.append(encrypted_text[i])
    return ''.join(decrypted_text)

# Example usage with a specific key for Vigenère Cipher
vigenere_key = "SECRET"
message = "Palladium Mall"
```

encrypted_vigenere_message = vigenere_encrypt(message, vigenere_key)

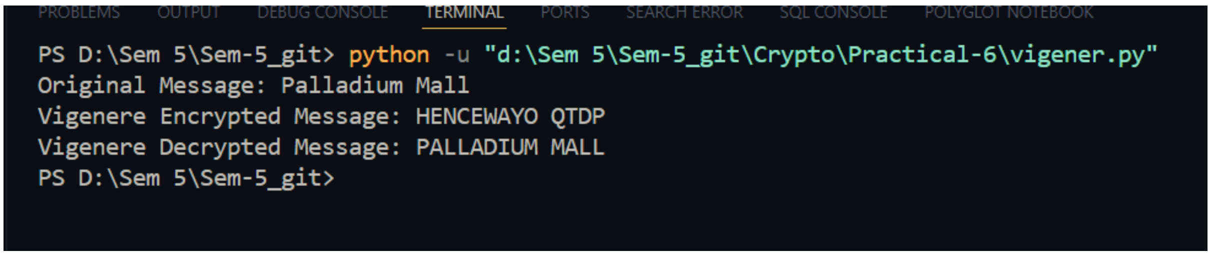
decrypted_vigenere_message = vigenere_decrypt(encrypted_vigenere_message, vigenere_key)

print("Original Message:", message)

print("Vigenere Encrypted Message:", encrypted_vigenere_message)

print("Vigenere Decrypted Message:", decrypted_vigenere_message)

Output:



```
PS D:\Sem 5\Sem-5_git> python -u "d:\Sem 5\Sem-5_git\Crypto\Practical-6\vigener.py"
Original Message: Palladium Mall
Vigenere Encrypted Message: HENCEWAYO QTDP
Vigenere Decrypted Message: PALLADIUM MALL
PS D:\Sem 5\Sem-5_git>
```