Name: Ayush Patel Enrolment: 22162171038 Class B Batch 55

Practical – 3

Subject – Cryptography

Aim

Alice wants to send some confidential information to Bob over a secure network, you have to create perform following task :

Task-1:

Provide Security using Caesar Cipher Algorithm. Also Decrypt on Receiver Side.

Code:

```
def caesar_cipher(text, shift, mode='encrypt'):
    if mode == 'decrypt':
        shift = -shift  # Reverse the shift for decryption

    processed_text = ""
    for char in text:
        if char.isalpha():
            shift_amount = shift % 26
            if char.islower():
                start = ord('a')
                processed_char = chr(start + (ord(char) - start +
shift_amount) % 26)
            else:
                start = ord('A')
                processed_char = chr(start + (ord(char) - start +
shift_amount) % 26)
            processed_text += processed_char
        else:
            processed_text += char

    return processed_text
```

```python
# User input
text = input("Enter the text: ")
shift = int(input("Enter the shift value: "))
mode = input("Enter 'encrypt' to encrypt or 'decrypt' to decrypt: ")

# Process
result = caesar_cipher(text, shift, mode)
print(f"Result: {result}")
```

Output:

```
PS E:\Sem 5\Sem-5_git> python -u "e:\Sem 5\Sem-5_git\Crypto\Practical-3\Task_1"
Enter the text: my name is ayush
Enter the shift value: 3
Enter 'encrypt' to encrypt or 'decrypt' to decrypt: encrypt
Result: pb qdph lv dbxvk
PS E:\Sem 5\Sem-5_git> python -u "e:\Sem 5\Sem-5_git\Crypto\Practical-3\Task_1"
Enter the text: pb qdph lv dbxvk
Enter the shift value: 3
Enter 'encrypt' to encrypt or 'decrypt' to decrypt: decrypt
Result: my name is ayush
PS E:\Sem 5\Sem-5_git>
```

Task-2:

Find the all possible Cipher Text & Plaintext pairs (Brute Force Attack)

Code:

```python
def caesar_cipher(text, shift, mode='encrypt'):
    if mode == 'decrypt':
        shift = -shift  # Reverse the shift for decryption

    processed_text = ""
    for char in text:
        if char.isalpha():
            shift_amount = shift % 26
            if char.islower():
                start = ord('a')
                processed_char = chr(start + (ord(char) - start +
shift_amount) % 26)
```

```
            else:
                start = ord('A')
                processed_char = chr(start + (ord(char) - start +
shift_amount) % 26)
            processed_text += processed_char
        else:
            if mode == 'decrypt':
                processed_text += char  # Add space back during
decryption
            # Skip spaces during encryption

    return processed_text

def brute_force_caesar_cipher(ciphertext):
    possible_plaintexts = []
    for shift in range(26):
        decrypted_text = caesar_cipher(ciphertext, shift,
mode='decrypt')
        possible_plaintexts.append((shift, decrypted_text))
    return possible_plaintexts

# User input
ciphertext = input("Enter the ciphertext: ")

# Brute force attack
all_possible_pairs = brute_force_caesar_cipher(ciphertext)
print("Possible plaintexts by trying all shifts:")
for shift, plaintext in all_possible_pairs:
    print(f"Shift {shift}: {plaintext}")
```

Output:

```
PS E:\Sem 5\Sem-5_git> python -u "e:\Sem 5\Sem-5_git\Crypto\Practical-3\Task_2"
Enter the ciphertext: pb qdph lv dbxvk
Possible plaintexts by trying all shifts:
Shift 0: pb qdph lv dbxvk
Shift 1: oa pcog ku cawuj
Shift 2: nz obnf jt bzvti
Shift 3: my name is ayush
Shift 4: lx mzld hr zxtrg
Shift 5: kw lykc gq ywsqf
Shift 6: jv kxjb fp xvrpe
Shift 7: iu jwia eo wuqod
Shift 8: ht ivhz dn vtpnc
Shift 9: gs hugy cm usomb
Shift 10: fr gtfx bl trnla
Shift 11: eq fsew ak sqmkz
```

Task-3:

Provide Security Mono-alphabetic Cipher Algorithm

Code:

```python
# Monoalphabetic Cipher Program

def generate_fixed_mono_alphabetic_key():
    # Define a fixed key mapping for the alphabet
    alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
    shuffled_alphabet = 'QWERTYUIOPASDFGHJKLZXCVBNM'  # Example
of a fixed key
    return dict(zip(alphabet, shuffled_alphabet))

def mono_alphabetic_cipher(text, key, mode='encrypt'):
    if mode == 'decrypt':
        key = {v: k for k, v in key.items()}  # Invert the key
for decryption

    processed_text = ""
    for char in text:
        if char.isalpha():
            if char.islower():
                processed_text += key[char.upper()].lower()
            else:
                processed_text += key[char]
        else:
            processed_text += char
    return processed_text
```

Name: Ayush Patel Enrolment: 22162171038 Class B Batch 55

```python
# Use a fixed key instead of randomizing
key = generate_fixed_mono_alphabetic_key()
print("Mono-alphabetic Cipher Algorithm")
# Ask user whether to encrypt or decrypt
operation = input("Do you want to (1) Encrypt or (2) Decrypt?
Enter 1 or 2: ")

if operation == '1':
    # User input for encryption
    plaintext = input("Enter the text to encrypt: ")
    ciphertext = mono_alphabetic_cipher(plaintext, key,
mode='encrypt')
    print(f"Key: {key}")
    print(f"Encrypted Text: {ciphertext}")

elif operation == '2':
    # User input for decryption
    ciphertext = input("Enter the text to decrypt: ")
    decrypted_text = mono_alphabetic_cipher(ciphertext, key,
mode='decrypt')
    print(f"Decrypted Text: {decrypted_text}")
else:
    print("Invalid option selected.")
```

Output:

```
41    print(f"Decrypted Text: {decrypted_text}")

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   SEARCH ERROR   SQL CONSOLE   POLYGLOT NOTEBOOK

PS E:\Sem 5\Sem-5_git> python -u "e:\Sem 5\Sem-5_git\Crypto\Practical-3\Task_3"
Mono-alphabetic Cipher Algorithm
Do you want to (1) Encrypt or (2) Decrypt? Enter 1 or 2: 1
Enter the text to encrypt: hey there ayush this side
Key: {'A': 'Q', 'B': 'W', 'C': 'E', 'D': 'R', 'E': 'T', 'F': 'Y', 'G': 'U', 'H': 'I', 'I': 'O', 'J': 'P', 'K': 'A', 'L': 'S', 'M': '
D', 'N': 'F', 'O': 'G', 'P': 'H', 'Q': 'J', 'R': 'K', 'S': 'L', 'T': 'Z', 'U': 'X', 'V': 'C', 'W': 'V', 'X': 'B', 'Y': 'N', 'Z': 'M'
}
Encrypted Text: itn zitkt qnxli ziol lort
PS E:\Sem 5\Sem-5_git> python -u "e:\Sem 5\Sem-5_git\Crypto\Practical-3\Task_3"
Mono-alphabetic Cipher Algorithm
Do you want to (1) Encrypt or (2) Decrypt? Enter 1 or 2: 2
Enter the text to decrypt: itn zitkt qnxli ziol lort
Decrypted Text: hey there ayush this side
PS E:\Sem 5\Sem-5_git>
```