SUB: Information Security

AY 2023-24 (Semester-V)

Experiment No: 5

Name:Bhuvi Ghosh
SAPID: 60009210191

Aim: Design and implement Encryption and Decryption Algorithm using Play fair Cipher.

Theory:

1. Playfair Cipher:

Introduction: Playfair Cipher is a cryptographic technique that encrypts pairs of letters (digraphs) instead of individual letters as in the case of traditional substitution ciphers. It employs a key square, typically a 5x5 grid, to perform the encryption and decryption. The key square is generated using a keyword.

Key Square Generation:

Start with a keyword (e.g., MONARCHY).
Remove duplicate letters, and combine with the remaining letters of the alphabet to form a 5x5 matrix (key square).
Fill the matrix row-wise with the letters of the keyword and remaining letters of the alphabet.
Encryption Algorithm:

Break the plaintext into digraphs.
If the letters of a digraph are in the same row of the key square, replace each letter with the letter to its right (circularly).
If the letters of a digraph are in the same column of the key square, replace each letter with the letter below it (circularly).
If the letters of a digraph form a rectangle, replace each letter with the letter in the same row but at the other corner of the rectangle.
The resulting digraphs are the ciphertext.
Decryption Algorithm:

Follow the same steps as the encryption algorithm, but replace the right and below movements with left and above movements.

Example:

1) **Plaintext: ATTACK**
Keyword: MONARCHY

Key Square:

M O N A R
C H YB D E
F G I J K
L P Q S T
U V W X Z
Encryption:

Break plaintext into digraphs: AT, TA, CK
Encrypt each digraph using the key square rules.
AT becomes TP
TA becomes MA
CK becomes KK
Concatenate the encrypted digraphs: TPMAKK
Decryption:

Break ciphertext into digraphs: TP, MA, KK
Decrypt each digraph using the key square rules.
TP becomes AT
MA becomes TA
KK becomes CK
Concatenate the decrypted digraphs: ATTAACK
Conclusion: Playfair Cipher provides a more secure alternative to simple substitution ciphers. Its use of digraphs and the key square introduces complexity that makes it resistant to frequency analysis. However, it is important to note that it is vulnerable to known-plaintext attacks and may not be suitable for all cryptographic purposes. The choice of a strong keyword and understanding the rules for handling various cases in the key square is crucial for the effectiveness of Playfair Cipher.

SUB: Information Security

```python
def clean_text(text):
    text = ''.join(filter(str.isalpha, text))
    text = text.upper()
    text = text.replace('J', 'I')
    return text
```

```python
def generate_key_square(key):
    key = key.replace('J', 'I')
    key = key.upper()
    key = ''.join(dict.fromkeys(key))
    alphabet = "ABCDEFGHIKLMNOPQRSTUVWXYZ"
    key_square = list(key)

    for letter in alphabet:
        if letter not in key_square:
            key_square.append(letter)

    key_square_matrix = [key_square[i:i+5] for i in range(0, 25, 5)]
    return key_square_matrix
```

```python
def find_position(key_square, letter):
    for i in range(5):
        for j in range(5):
            if key_square[i][j] == letter:
                return i, j
```

```python
def encrypt(text, key):
    cleaned_text = clean_text(text)
    key_square = generate_key_square(key)
    encrypted_text = []
    for i in range(0, len(cleaned_text), 2):
        letter1 = cleaned_text[i]
        letter2 = cleaned_text[i + 1] if i + 1 < len(cleaned_text) else 'X'

        row1, col1 = find_position(key_square, letter1)
        row2, col2 = find_position(key_square, letter2)

        if col1 == col2:
            encrypted_text.append(key_square[(row1 + 1) % 5][col1])
            encrypted_text.append(key_square[(row2 + 1) % 5][col2])
        elif row1 == row2:
            encrypted_text.append(key_square[row1][(col1 + 1) % 5])
            encrypted_text.append(key_square[row2][(col2 + 1) % 5])
        else:
            encrypted_text.append(key_square[row1][col2])
            encrypted_text.append(key_square[row2][col1])

    return ''.join(encrypted_text)
```

```python
def print_matrix(matrix):
    for row in matrix:
        print(' '.join(row))

def main():
    key = "KEYWORD"
    key_square = generate_key_square(key)

    print("Key Square Matrix:")
    print_matrix(key_square)
```

```
[ ] if __name__ == '__main__':
        main()
```

```
Key Square Matrix:
K E Y W O
R D A B C
F G H I L
M N P Q S
T U V X Z
```

```
[ ] key = "Keyword"
    plaintext = "Hell"

    encrypted = encrypt(plaintext, key)
    print("Plaintext:", plaintext)
    print("Encrypted:", encrypted)
```

```
Plaintext: Hell
Encrypted: GYSS
```

Conclusion: Encryption & decryption has been achieved using Playfair cipher in the given experiment.