CNS ASSIGNMENT 2 - IMPLEMENTATION OF CLASSICAL CRYPTOGRAPHIC ALGORITHM : PLAYFAIR

MIS - 112003151

NAME - ADWAIT VIPRA

The Playfair cipher is a symmetric key encryption algorithm that is used for encrypting or encoding a message. It employs a table of letters to perform substitution. Here's a brief overview of the Playfair cryptographic algorithm:

1. **Key Table Generation:**
   - The Playfair cipher uses a key table, traditionally a 5x5 matrix of letters.
   - The key phrase (ignoring duplicate letters) is used to fill the matrix, and the remaining letters of the alphabet are added in order, excluding duplicates.
2. **Message Preparation:**
   - The message to be encrypted is first processed. Pairs of letters are created from the plaintext message, and any repeated letters in a pair are separated by an arbitrary letter (commonly 'X').
3. **Encryption:**
   - The Playfair algorithm operates on pairs of letters (digraphs).
   - For each digraph in the plaintext:
     - If the two letters are in the same row of the key table, replace them with the letters to their immediate right, wrapping around if necessary.
     - If the two letters are in the same column, replace them with the letters immediately below, wrapping around if necessary.
     - If the letters are not in the same row or column, form a rectangle with the two letters and replace them with the letters at the opposite corners of the rectangle.
4. **Decryption:**
   - Decryption in Playfair involves using the inverse operations of encryption.
   - For each digraph in the ciphertext:
     - If the two letters are in the same row, replace them with the letters to their immediate left.
     - If the two letters are in the same column, replace them with the letters immediately above.
     - If the letters are not in the same row or column, form a rectangle and replace them with the letters at the opposite corners of the rectangle.
5. **Handling Odd-Length Messages:**
   - If the message has an odd number of letters, a padding letter (commonly 'X') is added at the end.

The Playfair cipher is relatively easy to implement and was used historically for secure communication. However, it has been largely replaced by more secure algorithms in modern cryptography due to its vulnerability to certain types of attacks, especially those involving frequency analysis.

In [2]:
```
#!/bin/python

alph = "ABCDEFGHIKLMNOPQRSTUVWXYZ"
def genmat(key):
    key = key.upper().replace('J', 'I')
    matrix = [['' for _ in range(5)] for _ in range(5)]
    used_chars = set()
    row, col = 0, 0

    for char in key + alph:
        if char not in used_chars:
            matrix[row][col] = char
            used_chars.add(char)
            col += 1
            if col == 5:
                col = 0
                row += 1

    return matrix


def getpos(matrix, char):
    for row in range(5):
        for col in range(5):
            if matrix[row][col] == char:
                return row, col


def encrypt(plain_text, key):
    matrix = genmat(key)

    plain_text = plain_text.upper().replace('J', 'I').replace(' ', 
'')

    if len(plain_text) % 2 != 0:
        plain_text += 'X'

    cipher_text = ''
    for i in range(0, len(plain_text), 2):
        char1, char2 = plain_text[i], plain_text[i + 1]
        row1, col1 = getpos(matrix, char1)
        row2, col2 = getpos(matrix, char2)

        if row1 == row2:
            cipher_text += matrix[row1][(col1 + 1) % 5] + matrix[ro
w2][(col2 + 1) % 5]
        elif col1 == col2:
            cipher_text += matrix[(row1 + 1) % 5][col1] + matrix[(r
ow2 + 1) % 5][col2]
        else:
            cipher_text += matrix[row1][col2] + matrix[row2][col1]

    return cipher_text


def decrypt(cipher_text, key):
    matrix = genmat(key)

    plain_text = ''
    for i in range(0, len(cipher_text), 2):
        char1, char2 = cipher_text[i], cipher_text[i + 1]
        row1, col1 = getpos(matrix, char1)
        row2, col2 = getpos(matrix, char2)

        if row1 == row2:
            plain_text += matrix[row1][(col1 - 1) % 5] + matrix[row
2][(col2 - 1) % 5]
        elif col1 == col2:
            plain_text += matrix[(row1 - 1) % 5][col1] + matrix[(ro
w2 - 1) % 5][col2]
        else:
            plain_text += matrix[row1][col2] + matrix[row2][col1]

    return plain_text



key = "XHIDDEN"
plain_text = "SECRET"
print("plaintext :", plain_text)
cipher_text = encrypt(plain_text, key)
print("ciphertext :", cipher_text)

decrypted_text = decrypt(cipher_text, key)
print("plaintext :", decrypted_text)
```

```
plaintext : SECRET
ciphertext : TDBSFZ
plaintext : SECRET
```