

**Experiment**

**No.**

**2**

**Title: Transposition Cipher**

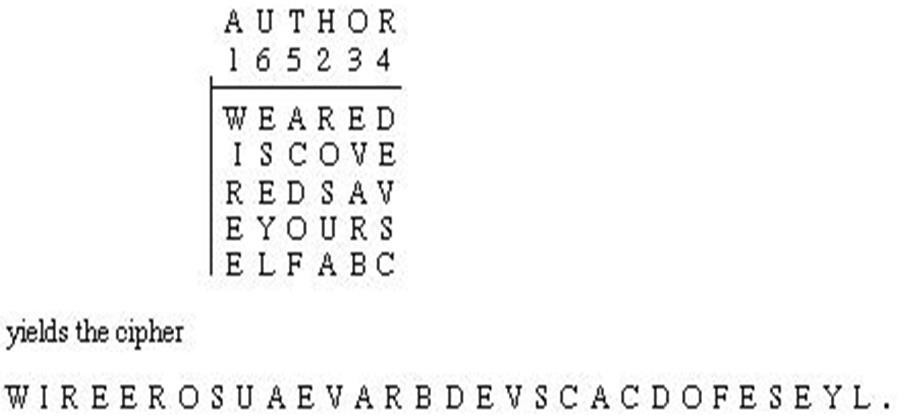
# Batch: B1 Roll No.: 1914078 Experiment No.: 3

**Title:** Transposition Cipher

**Resources needed:** Windows/Linux, IDE for Java, JRE.

# Theory

**Pre Lab/ Prior Concepts:**

**Symmetric-key algorithms** are a class of algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of cipher text. The keys may be identical or there may be a simple transformation to go between the two keys. The keys, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link. This requirement that both parties have access to the secret key is one of the main drawbacks of symmetric key encryption, in comparison to public-key encryption. Symmetric-key encryption can use either stream ciphers or block ciphers. Transposition Cipher is block cipher. Ancient cryptographic systems are classified as: Substitution and Permutation/Transposition Ciphers.

# Transposition Cipher/Permutation Cipher

A transposition cipher rearranges(permutes) symbols in a block without altering actual values. It has the same frequency distribution as the original text .So it is easily recognizable.

EXAMPLE :

Plaintext: HELLO MY DEAR

Cipher text: ELHLMDOYAER

There are varieties of transposition ciphers like: Rail Fence cipher, Column Transposition, Row Transposition

# Column Transposition Cipher

In Simple columnar transposition cipher plain text is written row by row in a predefined column size and cipher text message is Read column by column as per the key given. Example is demonstrated below.

# Row Transposition Cipher

In row transposition cipher letters of message are written out in rows over a specified number of columns and then r the columns are reordered according to some key before reading off the

rows

Key: 4 3 1 2 5 6 7 Rearranging 1 2 3 4 5 6 7

Plaintext: a t t a c k p t a t a c k p o s t p o n e t p s o o n e d u n t i l t n t u d i l t w o a m x y z a m o w x y z

Cipher text: TATACKPTPSOONENTUDILTAMOWXYZ

**Activity:**

Implement the following substitution ciphers:

1. Simple Columnar Transposition Cipher
2. Row Transposition Cipher

The program should have encryption function and decryption function for each cipher. Function should take message and a key as input from the user and display the expected output.

# SIMPLE COLUMNAR TRANSPOSITION CIPHER

**Code:**

def columnar\_encrypt(text,key):

    # Adding \_ to the difference place

    incompleteReplacedString = '\_'\*(len(key)-(len(text)%len(key)))

    updatedString = text + incompleteReplacedString

    sortedKeyWord = sorted(key)

    matrix = []

    while(len(updatedString)!=0):

        matrix.append(updatedString[:len(key)])

        updatedString=updatedString[len(key):]

    keyWord2 = list(key)

    cipher = ""

    for i in sortedKeyWord:

        index = keyWord2.index(i)

        for j in range(len(matrix)):

            cipher += matrix[j][index:index+1]

            matrix[j] = matrix[j][:index]+matrix[j][index+1:]

        keyWord2.remove(i)

    return cipher

def columnar\_decrypt(cipher,key):

    sortedKeyWord = sorted(key)

    matrix = []

    noOfRows = len(cipher)//len(key)

    while(len(cipher)!=0):

        matrix.append(cipher[:noOfRows])

        cipher=cipher[noOfRows:]

    decryptedText=""

    newMatrix = []

    for i in key:

        index = sortedKeyWord.index(i)

        newMatrix.append(matrix[index])

    for i in range(noOfRows):

        for j in range(len(newMatrix)):

            decryptedText = decryptedText + newMatrix[j][i]

    return decryptedText.replace("\_","")

text = input("Enter word to be encrypted: ").upper()

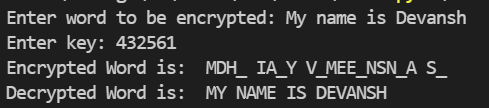
key = input("Enter key: ").upper()

cipher = columnar\_encrypt(text,key)

print("Encrypted Word is: ",cipher)

print("Decrypted Word is: ",columnar\_decrypt(cipher,key))

**Output:**

****

# ROW TRANSPOSITION CIPHER

**CODE:**

def row\_encrypt(text,key):

    incompleteReplacedString = '\_'\*(len(key)-(len(text)%len(key)))

    updatedString = text + incompleteReplacedString

    sortedKeyWord = sorted(key)

    matrix = []

    while(len(updatedString)!=0):

        matrix.append(updatedString[:len(key)])

        updatedString=updatedString[len(key):]

    key2 = list(key)

    encryptMatrix = []

    for i in matrix:

        encryptMatrix.append("")

    for i in range(len(sortedKeyWord)):

        index = key2.index(sortedKeyWord[i])

        for j in range(len(matrix)):

            letter = matrix[j][index:index+1]

            encryptMatrix[j] = encryptMatrix[j]+letter

    return "".join(encryptMatrix)

def row\_decrypt(cipher,key):

    matrix = []

    while(len(cipher)!=0):

        matrix.append(cipher[:len(key)])

        cipher=cipher[len(key):]

    cipher = []

    for i in matrix:

        cipher.append("")

    key2 = list(key)

    sortedKeyWord = sorted(key)

    for i in range(len(key2)):

        index = sortedKeyWord.index(key2[i])

        for j in range(len(matrix)):

            letter = matrix[j][index:index+1]

            cipher[j] = cipher[j]+letter

    return ("".join(cipher)).replace("\_","")

text = input("Enter word to be encrypted: ")

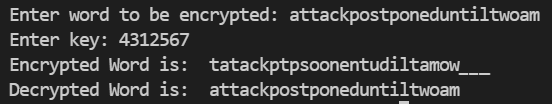
key = input("Enter key: ")

cipher = row\_encrypt(text,key)

print("Encrypted Word is: ",cipher)

print("Decrypted Word is: ",row\_decrypt(cipher,key))

**Output:**

****

**Questions:**

1. Write down the flaws of Simple Columnar Transposition cipher and Row Transposition Cipher:

Ans) In the case of small messages, the ciphertext is easily deciphered by anyone willing to try deciphering with different key values.

* A disadvantage is that such **ciphers** are considerably more laborious and error prone than simpler **ciphers**. ciphers.
* They have a Limited key space.
* Also, the shape choices of the key are tied to size of message trivially. Transposition ciphers are not highly secure because they do not change the letters in the plaintext or even cover up frequencies. Also if bogus characters not assigned or not given, it may be difficult to decrypt the message as it needs to be divided.

**Outcomes:**

**CO1:** Describe the basics of information security.

**Conclusion:** We learnt about transposition ciphers and implemented the code and saw the working of simple columnar and row transposition ciphers.

**Grade: AA / AB / BB / BC / CC / CD /DD**

## Signature of faculty in-charge with date

**References: Books/ Journals/ Websites:**

1. Charles P. Pfleeger, “Security in Computing”, Pearson Education
2. Behrouz A. Forouzan, “Cryptography and Network Security”, Tata McGraw Hill
3. William Stalling, “Cryptography and Network Security”, Prentice Hall