Experiment No. 2

Title: Transposition Cipher

Batch: A3 Roll No.: 16010421065 Name: Tanvi Natu Experiment No.: 2

Aim: To implement transposition cipher – Row transposition and column transposition cipher.

Resources needed: Windows/Linux

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

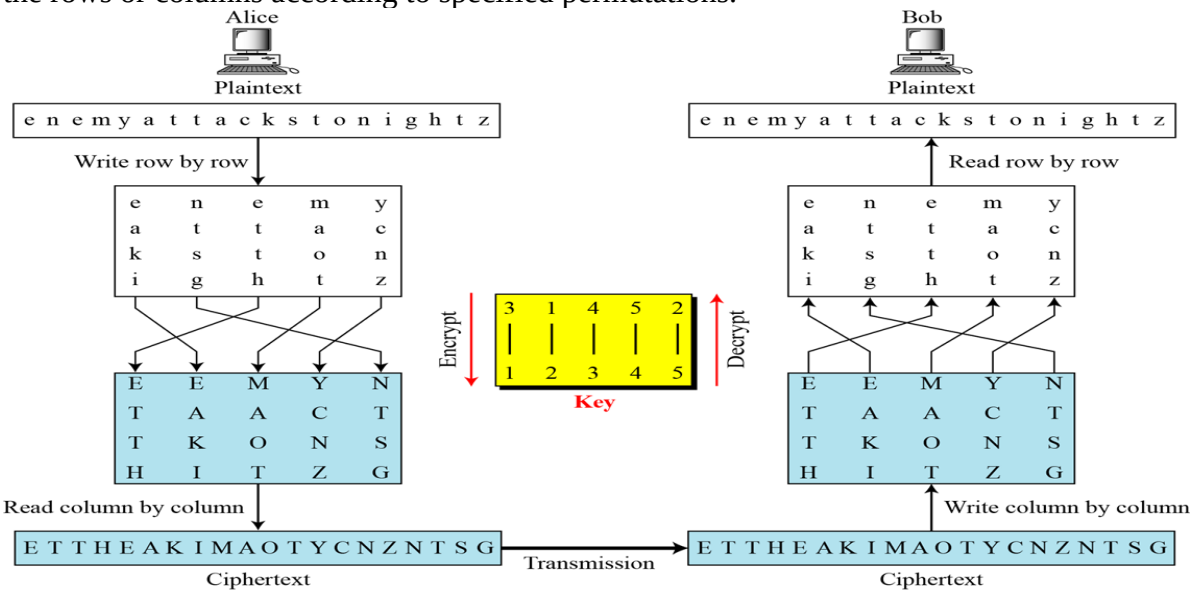
A transposition cipher rearranges (permutes) symbols in a block without altering actual values. It has the same frequency distribution as the original text .So it is easily recognizable.

EXAMPLE :

Plaintext: HELLO MY DEAR
Cipher text: ELHLMDOYAER

There are varieties of transposition ciphers like: keyless and keyed transposition ciphers.

Following figure shows the combination of both keyed and keyless. To encrypt with a transposition cipher, we first write the plaintext into a matrix of a given size and then permute the rows or columns according to specified permutations.



For the transposition, the key consists of the size of the matrix and the row or column permutations. The recipient who knows the key can simply put the cipher text into the appropriate sized matrix and undo the permutations to recover the plaintext.
Unlike a simple substitution, the transposition does nothing to disguise the letters that appear in the message But it does appear to thwart an attack that relies on the statistical information contained in the plaintext, since the plaintext statistics are disbursed throughout the cipher text. The double transposition is not a trivial cipher to break.

Activity:
Step 1 – Go through the theory explained and the instructions given by the instructor.

Step 2 – Derive/find encryption/decryption formula for each of the following transposition ciphers. Assume P as a plaintext square matrix, K as a row/column key and C as a ciphertext square matrix.

1) Row transposition cipher –

C[i][j] = P[k[i]][j]

P[i][j] = C[k_inverse[i]][j]

2) Column transposition cipher –

C[i][j] = P[i][k[j]]

P[i][j] = C[i][k_inverse[j]]

3) Double transposition cipher (row followed by column)

C[i][j] = P[k[i]][j]

P[i][j] = C[k_inverse[i]][k_inverse[j]]

Step 3 – Implement

1) Row transposition cipher
2) Column transposition cipher
3) Double transposition cipher (row followed by column)

Implementation:
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.

Results: (Program with output as per the format)

Code:

```
import numpy as py
def encrypt(text,l,k):
  m=0
  l=l+1
  c=[[[] for column in range(l)] for row in range(l)]
  p=[[[] for column in range(l)] for row in range(l)]
  for i in range(l):
    for j in range(l):
      if(m<len(text)):
        p[j][i]=text[m]
        m=m+1
      else:
        p[j][i]=" "
  print("\nThe plain text matrix is as follows:")
  for i in range(l):
      print(p[i])
  print(" ")
  for i in range(l):
    for j in range(l):
      c[i][j]=p[k[i]][j]
  return c


def decrypt(text,l,k):
  m=0
  l=l+1
  c=[[[] for column in range(l)] for row in range(l)]
  p=[[[] for column in range(l)] for row in range(l)]
  for i in range(l):
    for j in range(l):
      if(m<len(text)):
        c[i][j]=text[m]
        m=m+1
      else:
        c[i][j]=" "
  print("\nThe Cypher text matrix is as follows:")
  for i in range(l):
      print(c[i])
  print(" ")
  for i in range(l):
    for j in range(l):
      p[k[i]][j]=c[i][j]
  return p
```

```
def encrypt2(text,l,k):
  m=0
  l=l+1
  c=[[] for column in range(l)] for row in range(l)]
  p=[[] for column in range(l)] for row in range(l)]
  for i in range(l):
    for j in range(l):
      if(m<len(text)):
        p[i][j]=text[m]
        m=m+1
      else:
        p[i][j]=" "
  print("\nThe plain text matrix is as follows:")
  for i in range(l):
      print(p[i])
  print(" ")
  for i in range(l):
    for j in range(l):
      c[i][j]=p[i][k[j]]
  return c


def decrypt2(text,l,k):
  m=0
  l=l+1
  c=[[] for column in range(l)] for row in range(l)]
  p=[[] for column in range(l)] for row in range(l)]
  for i in range(l):
    for j in range(l):
      if(m<len(text)):
        c[j][i]=text[m]
        m=m+1
      else:
        c[i][j]=" "
  print("\nThe cypher text matrix is as follows:")
  for i in range(l):
      print(c[i])
  print(" ")
  for i in range(l):
    for j in range(l):
      p[i][k[j]]=c[i][j]
  return p


x=1
```

```
while x:
  print("Menu Driven Program")
  print("1.Encrypt (Row Transposition)")
  print("2.Decrypt (Row Transposition")
  print("3.Encrypt (Column Transposition)")
  print("4.Decrypt (Column Transposition")
  print("5.Encrypt (Double Transposition)")
  print("6.Decrypt (Double Transposition")
  print("7.Exit")
  choice=int(input("Enter your choice:"))
  if choice==1:
    text = input("Enter the message: ")
    length=len(text)
    l=0
    for i in range(length):
      if(i*i>=length):
        l=i-1
        break
    print("The range for keys is 0 to ",l)
    s=input("Enter the keys: ")
    k = list(s.split(" "))
    k = [int(i) for i in k]
    c=encrypt(text,l,k)
    l=l+1
    print ("Cipher Text:",end="")
    for i in range(l):
      for j in range(l):
        print(c[i][j],end="")
    print("\n")

  elif choice==2:
    text = input("Enter the message: ")
    length=len(text)
    l=0
    for i in range(length):
      if(i*i>=length):
        l=i-1
        break
    print("The range for keys is 0 to ",l)
    s=input("Enter the keys: ")
    k = list(s.split(" "))
    k = [int(i) for i in k]
    p=decrypt(text,l,k)
    l=l+1
    print ("Plain Text:",end="")
```

```
    for i in range(l):
      for j in range(l):
        print(p[j][i],end="")
    print("\n")

  elif choice==3:
    text = input("Enter the message: ")
    length=len(text)
    l=0
    for i in range(length):
      if(i*i>=length):
        l=i-1
        break
    print("The range for keys is 0 to ",l)
    s=input("Enter the keys: ")
    k = list(s.split(" "))
    k = [int(i) for i in k]
    c=encrypt2(text,l,k)
    l=l+1
    print ("Cipher Text:",end="")
    for i in range(l):
      for j in range(l):
        print(c[j][i],end="")
    print("\n")

  elif choice==4:
    text = input("Enter the message: ")
    length=len(text)
    l=0
    for i in range(length):
      if(i*i>=length):
        l=i-1
        break
    print("The range for keys is 0 to ",l)
    s=input("Enter the keys: ")
    k = list(s.split(" "))
    k = [int(i) for i in k]
    p=decrypt2(text,l,k)
    l=l+1
    print ("Plain Text:",end="")
    for i in range(l):
      for j in range(l):
        print(p[i][j],end="")
    print("\n")
```

```
elif choice==5:
  text = input("Enter the message: ")
  length=len(text)
  l=0
  for i in range(length):
    if(i*i>=length):
      l=i-1
      break
  print("For Row Transformation first-")
  print("The range for keys is 0 to ",l)
  s=input("Enter the keys: ")
  k = list(s.split(" "))
  k = [int(i) for i in k]
  c=encrypt(text,l,k)
  l=l+1
  text2=""
  for i in range(l):
    for j in range(l):
      text2=text2+c[i][j]
  length=len(text2)
  l=0
  for i in range(length):
    if(i*i>=length):
      l=i-1
      break
  print("For Column Transformation now-")
  print("The range for keys is 0 to ",l)
  s=input("Enter the keys: ")
  k = list(s.split(" "))
  k = [int(i) for i in k]
  c=encrypt2(text2,l,k)
  l=l+1
  print ("Cipher Text:",end="")
  for i in range(l):
    for j in range(l):
      print(c[j][i],end="")
  print("\n")

elif choice==6:
  print("For Column Transformation first-")
  text = input("Enter the message: ")
  length=len(text)
  l=0
  for i in range(length):
    if(i*i>=length):
```

```
        l=i-1
        break
    print("The range for keys is 0 to ",l)
    s=input("Enter the keys: ")
    k = list(s.split(" "))
    k = [int(i) for i in k]
    p=decrypt2(text,l,k)
    l=l+1
    text2=""
    for i in range(l):
      for j in range(l):
        text2=text2+p[i][j]
    print("For Row Transformation first-")
    length=len(text2)
    l=0
    for i in range(length):
      if(i*i>=length):
        l=i-1
        break
    print("The range for keys is 0 to ",l)
    s=input("Enter the keys: ")
    k = list(s.split(" "))
    k = [int(i) for i in k]
    p=decrypt(text2,l,k)
    l=l+1
    print ("Plain Text:",end="")
    for i in range(l):
      for j in range(l):
        print(p[j][i],end="")
    print("\n")

  elif choice==7:
    print("Thank you. Exiting.")
    x=0;
  else:
    print("Please enter the correct choice")
```

Output:

```
Menu Driven Program
1.Encrypt (Row Transposition)
2.Decrypt (Row Transposition
3.Encrypt (Column Transposition)
4.Decrypt (Column Transposition
5.Encrypt (Double Transposition)
6.Decrypt (Double Transposition
7.Exit
Enter your choice:1
Enter the message: hello world
The range for keys is 0 to  3
Enter the keys: 0 3 1 2

The plain text matrix is as follows:
['h', 'o', 'r', ' ']
['e', ' ', 'l', ' ']
['l', 'w', 'd', ' ']
['l', 'o', ' ', ' ']

Cipher Text:hor lo  e l lwd

Menu Driven Program
1.Encrypt (Row Transposition)
2.Decrypt (Row Transposition
3.Encrypt (Column Transposition)
4.Decrypt (Column Transposition
5.Encrypt (Double Transposition)
6.Decrypt (Double Transposition
7.Exit
Enter your choice:2
Enter the message: hor lo  e l lwd
The range for keys is 0 to  3
Enter the keys: 0 3 1 2

The Cypher text matrix is as follows:
['h', 'o', 'r', ' ']
['l', 'o', ' ', ' ']
['e', ' ', 'l', ' ']
['l', 'w', 'd', ' ']

Plain Text:hello world
```

```
Menu Driven Program
1.Encrypt (Row Transposition)
2.Decrypt (Row Transposition
3.Encrypt (Column Transposition)
4.Decrypt (Column Transposition
5.Encrypt (Double Transposition)
6.Decrypt (Double Transposition
7.Exit
Enter your choice:3
Enter the message: hello world
The range for keys is 0 to  3
Enter the keys: 3 1 0 2

The plain text matrix is as follows:
['h', 'e', 'l', 'l']
['o', ' ', 'w', 'o']
['r', 'l', 'd', ' ']
[' ', ' ', ' ', ' ']

Cipher Text:lo  e l hor lwd
```

```
Menu Driven Program
1.Encrypt (Row Transposition)
2.Decrypt (Row Transposition
3.Encrypt (Column Transposition)
4.Decrypt (Column Transposition
5.Encrypt (Double Transposition)
6.Decrypt (Double Transposition
7.Exit
Enter your choice:4
Enter the message: lo  e l hor lwd
The range for keys is 0 to  3
Enter the keys: 3 1 0 2

The cypher text matrix is as follows:
['l', 'e', 'h', 'l']
['o', ' ', 'o', 'w']
[' ', 'l', 'r', 'd']
[' ', ' ', ' ', ' ']

Plain Text:hello world
```

```
Menu Driven Program
1.Encrypt (Row Transposition)
2.Decrypt (Row Transposition
3.Encrypt (Column Transposition)
4.Decrypt (Column Transposition
5.Encrypt (Double Transposition)
6.Decrypt (Double Transposition
7.Exit
Enter your choice:5
Enter the message: hello world
For Row Transformation first-
The range for keys is 0 to  3
Enter the keys: 3 2 1 0

The plain text matrix is as follows:
['h', 'o', 'r', ' ']
['e', ' ', 'l', ' ']
['l', 'w', 'd', ' ']
['l', 'o', ' ', ' ']

For Column Transformation now-
The range for keys is 0 to  3
Enter the keys: 3 2 1 0
```

```
The plain text matrix is as follows:
['h', 'o', 'r', ' ']
['l', 'o', ' ', ' ']
['l', 'w', 'd', ' ']
['e', ' ', 'l', ' ']

Cipher Text:    r dloow hlle

Menu Driven Program
1.Encrypt (Row Transposition)
2.Decrypt (Row Transposition
3.Encrypt (Column Transposition)
4.Decrypt (Column Transposition
5.Encrypt (Double Transposition)
6.Decrypt (Double Transposition
7.Exit
Enter your choice:6
For Column Transformation first-
Enter the message:     r dloow hlle
The range for keys is 0 to  3
Enter the keys: 3 2 1 0
```

```
The cypher text matrix is as follows:
[' ', 'r', 'o', 'h']
[' ', ' ', 'o', 'l']
[' ', 'd', 'w', 'l']
[' ', 'l', ' ', 'e']

For Row Transformation first-
The range for keys is 0 to  3
Enter the keys: 0 3 2 1

The Cypher text matrix is as follows:
['h', 'o', 'r', ' ']
['l', 'o', ' ', ' ']
['l', 'w', 'd', ' ']
['e', ' ', 'l', ' ']

Plain Text:hello world
```

```
Menu Driven Program
1.Encrypt (Row Transposition)
2.Decrypt (Row Transposition
3.Encrypt (Column Transposition)
4.Decrypt (Column Transposition
5.Encrypt (Double Transposition)
6.Decrypt (Double Transposition
7.Exit
Enter your choice:8
Please enter the correct choice
Menu Driven Program
1.Encrypt (Row Transposition)
2.Decrypt (Row Transposition
3.Encrypt (Column Transposition)
4.Decrypt (Column Transposition
5.Encrypt (Double Transposition)
6.Decrypt (Double Transposition
7.Exit
Enter your choice:7
Thank you. Exiting.
```

Questions:

1) **Compare substitution ciphers and transposition/permutation ciphers.**

Answer:

Aspect	Substitution Ciphers	Transposition/Permutation Ciphers
Basic Principle	Replace elements with others based on a key or rule	Rearrange elements' order based on a key or algorithm
Security	Higher security due to complex substitution patterns	Generally lower security; patterns may still be visible

Aspect	Substitution Ciphers	Transposition/Permutation Ciphers
Complexity	Can create complex mappings, harder to break using frequency analysis	Simpler rearrangement, easier analysis of patterns
Ciphertext Length	Same length as plaintext	Same length or longer, depending on the algorithm
Encryption/Decryption	Relatively easier since it involves direct mapping	More complex due to rearrangement, requires algorithms
Key Usage	Key defines substitution pattern for characters	Key defines order of rearrangement for characters
Examples	Caesar, Monoalphabetic, Polyalphabetic ciphers	Columnar, Rail Fence, Route ciphers
Use in Modern Cryptography	Typically as components in more complex algorithms	Rarely used due to lower security; modern ciphers are more advanced
Strengths	Complex mapping, resistant to frequency analysis	Variation in order, limited use in modern cryptography
Weaknesses	Vulnerable to known-plaintext attacks, frequency analysis	Vulnerable to pattern analysis, limited security

2) Define confusion and diffusion properties. Comment on of both substitution and transposition ciphers w.r.t. confusion and diffusion properties.

Answer:
Confusion and Diffusion Properties:

Confusion and diffusion are two fundamental concepts in the design of secure encryption algorithms, particularly in the context of modern symmetric-key ciphers. These properties were introduced by Claude Shannon, a pioneer in the field of cryptography.

Confusion: Confusion aims to make the relationship between the plaintext, the ciphertext, and the encryption key as complex and confusing as possible. In other words, it ensures that a small change in the input (plaintext or key) results in a drastic change in the output (ciphertext). This property adds a level of randomness and non-linearity to the encryption process, making it difficult for an attacker to deduce any meaningful information about the key from the ciphertext.

Diffusion: Diffusion aims to spread the influence of each input element over the entire output, increasing the statistical independence of the ciphertext from the plaintext. This property ensures that any small change in the input leads to significant changes throughout the ciphertext. Diffusion prevents patterns in the plaintext from persisting in the ciphertext and helps in hiding statistical properties of the original data.

Confusion and Diffusion in Substitution Ciphers:

Confusion: Substitution ciphers, particularly modern ones like the Advanced Encryption Standard (AES), exhibit high confusion due to complex substitution operations involving multiple rounds and substitution boxes (S-boxes). The

relationship between the input data, the key, and the ciphertext is intricate and nonlinear.
Diffusion: Diffusion might be relatively weaker in simple substitution ciphers, as small changes in the plaintext might not lead to significant changes in the ciphertext. More advanced substitution ciphers, like AES, incorporate diffusion through multiple rounds and mixing operations.

Confusion and Diffusion in Transposition Ciphers:

Confusion: Transposition ciphers, especially those that involve complex rearrangements and permutations, can introduce confusion by making the relationship between plaintext elements and ciphertext elements more complex.
Diffusion: Diffusion can be less pronounced in some transposition ciphers, as the arrangement of elements doesn't necessarily ensure a spread-out influence of individual plaintext elements.

Outcomes: CO1: Describe the basics of Information Security
CO2: Illustrate different cryptographic algorithms for security.

Conclusion: Successfully implemented Row and Column Transposition ciphers, showcasing basic encryption through character rearrangement.

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

Signature of faculty in-charge with date

References: Books/ Journals/ Websites:

1. Behrouz A. Forouzan, “Cryptography and Network Security”, Tata McGraw Hill
2. Mark Stamp, “Information Security Principles and Practice”, Wiley.
3. William Stalling, “Cryptography and Network Security”, Prentice Hall