Experiment No. 1: Design and Implementation of a Substitution and Transposition ciphers

Aim: To design and Implementation of a product cipher using Substitution and Transposition ciphers

Theory:

Substitution cipher is a method of encryption by which units of plaintext are replaced with ciphertext according to a regular system; the "units" may be single letters (the most common), pairs of letters, triplets of letters, mixtures of the above, and so forth. The receiver deciphers the text by performing an inverse substitution.

Transposition cipher is a method of encryption by which the positions held by units of plaintext (which are commonly characters or groups of characters) are shifted according to a regular system, so that the ciphertext constitutes a permutation of the plaintext. That is, the order of the units is changed.

Substitution ciphers can be compared with Transposition ciphers. In a transposition cipher, the units of the plaintext are rearranged in a different and usually quite complex order, but the units themselves are left unchanged. By contrast, in a substitution cipher, the units of the plaintext are retained in the same sequence in the ciphertext, but the units themselves are altered.

- Caesar Cipher: In cryptography, a Caesar cipher, also known as a Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet. For example, with a shift of 3, A would be replaced by D, B would become E, and so on. The method is named after Julius Caesar, who used it to communicate with his generals.

Example:

The transformation can be represented by aligning two alphabets; the cipher alphabet is the plain alphabet rotated left or right by some number of positions. For instance, here is a Caesar cipher using a left rotation of three places (the shift parameter, here 3, is used as the key):

Plain: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Cipher: DEFGHIJKLMNOPQRSTUVWXYZABC

• Columnar Transposition: In a columnar transposition, the message is written out in rows of a fixed length, and then read out again column by column, and the columns are chosen in some scrambled order. Both the width of the rows and the permutation of the columns are usually defined by a keyword.

In a regular columnar transposition cipher, any spare spaces are filled with nulls; in an irregular columnar transposition cipher, the spaces are left blank. Finally, the message is read off in columns, in the order specified by the keyword.

Algorithm/Procedure:

- **Substitution**
    - Display menu of operation – e for encryption and d for decryption.
    - Accept choice from user
    - If choice is encryption-
        - Accept plaintext from user
        - Accept key from user.

- Take k = 0.
- Extract k^{th} character from string.
- Add key to it and get new value.
- If new value > 26

  New value = New value % 26.
- Add as k^{th} character of ciphertext.
- Increment k.
- If(k < length(plaintext)) goto step 'd'.
- Display plaintext and ciphertext(output).

- If choice is decryption-
  - Accept cipher text from user
  - Accept key from user.
  - Take k = 0.
  - Extract k^{th}character from string.

- Subtract key from it and get new value.
- If new value > 26

  New value = New value % 26.
- Add as k^{th}character of plaintext.
- Increment k.
- If(k < length(ciphertext)) goto step 'd'.
- Display ciphertext and plaintext(output).

- Ask user want to continue or not
- If yes , go to step 2;else stop.

- **Transposition**
  - Count how many letters are in your ciphertext (for example, 75) and factor that number (75 =5*5*3).
  - Create all of the possible matrices to fit this ciphertext (in our case, 3x25, 5x15, 15x5, 25x3).
  - Write the ciphertext into these matrices down the columns.

- For each of your matrices, consider all of the possible permutations of the columns (for n columns, there are n! possible rearrangements). In our case, we hope that the message was enciphered using one of the last two matrices (the 15x5 and the 25x3), since in those cases, we have only 6 and 120 possibilities to check (3! = 6, 5! = 120, 15! ~ 1.31x10^12, 25! ~ 1.55x10^25).

- Rearrange each matrix to see if you get anything intelligible. Read the message off row-by-row. Note that this is much more easily done by a computer than by hand, but it is doable (for small matrices).

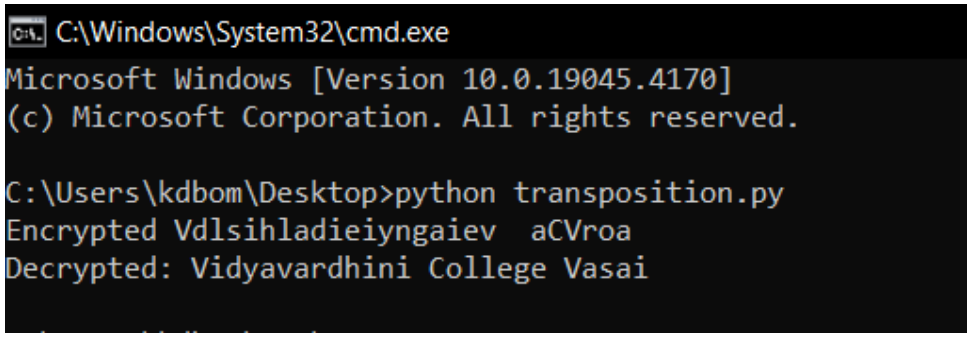
Source Code:

transposition.py

```
def check():
    msg="Vidyavardhini College Vasai"
    key=8
    ct=encrypt(key,msg)
    print("Encrypted",ct)
    dt = decrypt(key, ct)
    print("Decrypted:", dt)
def encrypt(key,msg):
    ct=[""]*key
    for col in range(key):
        colI=col
        while colI<len(msg):
            ct[col]+=msg[colI]
            colI+=key
    return "".join(ct)
```

```
def decrypt(key, msg):
    pt = [''] * key
    i = 0
    for col in range(key):
        colI = col
        while colI < len(msg):
            pt[colI // key] += msg[i]
            colI += key
            i += 1
    return "".join(pt)
check()
```

Output:



substitution.py

```
def decrypt(a,b):
        a = a % b
        for x in range(a, b):
```

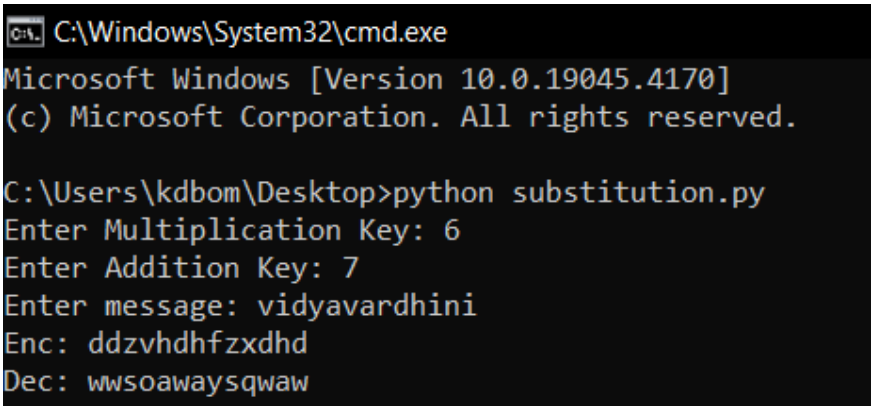
```
            if((a * x) % b == 1):
                return x
    return 1

alpha = "abcdefghijklmnopqrstuvwxyz"
k1 = int(input("Enter Multiplication Key: "))
k2 = int(input("Enter Addition Key: "))
kd = decrypt(k1, 26)
msg = input("Enter message: ")

enc = ""
for i in msg:
    if i == "":
        enc = enc +i
    enc = enc + alpha[(alpha.find(i)*k1+k2) % 26]
print("Enc: "+enc)

dec = ""
for i in enc:
    if i == "":
        dec = dec +i
    dec = dec + alpha[((alpha.find(i) - k2)*kd)%26]
print("Dec: "+dec)
```

Output:



Conclusion: In conclusion**,** Transposition rearranges characters based on a key, while substitution replaces characters based on a predetermined mapping. Both ciphers provide basic encryption, but substitution is more vulnerable to frequency analysis due to fixed character mappings. Transposition can be more resistant to such attacks. However, both ciphers are relatively simple and can be easily broken with modern cryptographic techniques. They are mainly educational or used in combination with stronger encryption methods for added security.