SUB: Information Security

AY 2023-24 (Semester-V)

Experiment No: 4
60009210105 Amitesh Sawarkar
D 12

Aim: To Implement Encryption and Decryption using Columnar Transposition Cipher.

Theory:

1. Transposition Cipher

   Transposition Cipher is a cryptographic algorithm where the order of alphabets in the plaintext is rearranged to form a cipher text. In this process, the actual plain text alphabets are not included.

2. Columnar Transposition Cipher/ Row Column Transposition Cipher.

   Given a plain-text message and a numeric key, cipher/de-cipher the given text using Columnar Transposition Cipher

   The Columnar Transposition Cipher is a form of transposition cipher just like Rail Fence Cipher. Columnar Transposition involves writing the plaintext out in rows, and then reading the ciphertext off in columns one by one.

SUB: Information Security

In a transposition cipher, the order of the alphabets is re-arranged to obtain the cipher-text.

1. 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.
2. Width of the rows and the permutation of the columns are usually defined by a keyword.
3. For example, the word HACK is of length 4 (so the rows are of length 4), and the permutation is defined by the alphabetical order of the letters in the keyword. In this case, the order would be "3 1 2 4".
4. Any spare spaces are filled with nulls or left blank or placed by a character (Example: _).
5. Finally, the message is read off in columns, in the order specified by the keyword.

1. To decipher it, the recipient has to work out the column lengths by dividing the message length by the key length.
2. Then, write the message out in columns again, then re-order the columns by reforming the key word.

Example:
Encryption and Decryption

1) **Plaintext : "Attack Postponed until two am"**
   Keyword: 4312567
   Ciphertext:

SUB: Information Security

```
import math

key = "4312567|"

def encryptMessage(msg):
    cipher = ""

    k_indx = 0

    msg_len = float(len(msg))
    msg_lst = list(msg)
    key_lst = sorted(list(key))

    col = len(key)

    row = int(math.ceil(msg_len / col))

    fill_null = int((row * col) - msg_len)
    msg_lst.extend('_' * fill_null)

    matrix = [msg_lst[i: i + col]
              for i in range(0, len(msg_lst), col)]
```

SUB: Information Security

```
    for _ in range(col):
        curr_idx = key.index(key_lst[k_indx])
        cipher += ''.join([row[curr_idx]
                           for row in matrix])
        k_indx += 1

    return cipher

def decryptMessage(cipher):
    msg = ""

    k_indx = 0

    msg_indx = 0
    msg_len = float(len(cipher))
    msg_lst = list(cipher)

    col = len(key)

    row = int(math.ceil(msg_len / col))
```

SUB: Information Security

```
    key_lst = sorted(list(key))

    dec_cipher = []
    for _ in range(row):
        dec_cipher += [[None] * col]

    for _ in range(col):
        curr_idx = key.index(key_lst[k_indx])

        for j in range(row):
            dec_cipher[j][curr_idx] = msg_lst[msg_indx]
            msg_indx += 1
        k_indx += 1

    try:
        msg = ''.join(sum(dec_cipher, []))
    except TypeError:
        raise TypeError("This program cannot",
                        "handle repeating words.")

    null_count = msg.count('_')

    if null_count > 0:
        return msg[: -null_count]
```

SUB: Information Security

```
    return msg

msg = "Attack Postponed until two am"

cipher = encryptMessage(msg)
print("Encrypted Message: {}".
                format(cipher))

print("Decryped Message: {}".
       format(decryptMessage(cipher)))
```

```
Encrypted Message: ttn aptatsuoAo wcoimknl_ e _Pdt_
Decryped Message: Attack Postponed until two am
```

Conclusion:

The experiment implementing the Columnar Transposition Cipher revealed its effectiveness in basic message encryption. It demonstrated the significance of a secret key, the importance of handling spaces, and the successful reversibility of the encryption process. However, it's essential to acknowledge its limited security for critical data and its suitability for educational or non-sensitive communication purposes.