

Cryptography and Network Security Lab

Assignment 2

Student Details

Name : Krunal Rank
Adm No. : U18C0081

1

```
# sample text : once upon a time there was a little girl named goldilocks she went
for a walk in the forest pretty soon she came upon a house she knocked and when no
one answered she walked right in at the table in the kitchen there were three bowls
of porridge goldilocks was hungry she tasted the porridge from the first bowl

from typing import OrderedDict
import math
from itertools import chain

PLACEHOLDER = "#"

class ERRORS:
    KEY_LENGTH = "Invalid Key! Key Length should be more than 1!"
    KEY_UNIQUE_CHARACTERS = "Invalid Key! Key characters should be unique!"
    CHOICE = "Please enter a valid choice!"

def columnar_transposition_cipher_encrypt(text:str,key:str)->str:
    """
    Encrypts the given text using Key using Columnar Transposition Cipher Encryption
    """
    if len(key) <=1:
        raise Exception(ERRORS.KEY_LENGTH)
    key_mapping = OrderedDict()
    for i in range(len(key)):
        if key[i] in key_mapping:
            raise Exception(ERRORS.KEY_UNIQUE_CHARACTERS)
        key_mapping[key[i]] = i

    final_row = int(math.ceil(len(text)/len(key)))
    character_matrix = [[PLACEHOLDER for _ in range(len(key))] for _ in
range(final_row)]

    for i in range(len(text)):
```

```

        row = i//len(key)
        col = i%len(key)
        character_matrix[row][col] = text[i]

    encrypted_char_list = []
    for key in sorted(key_mapping):
        col = key_mapping[key]
        for row in range(final_row):
            encrypted_char_list.append(character_matrix[row][col])

    return "".join(encrypted_char_list)

def columnar_transposition_cipher_decrypt(text:str,key:str)->str:
    """
    Decrypts the given text using Key using Columnar Transposition Cipher Decryption
    """
    if len(key) <=1:
        raise Exception(ERRORS.KEY_LENGTH)
    key_mapping = OrderedDict()
    for i in range(len(key)):
        if key[i] in key_mapping:
            raise Exception(ERRORS.KEY_UNIQUE_CHARACTERS)
        key_mapping[key[i]] = i

    final_row = int(math.ceil(len(text)/len(key)))
    character_matrix = [[PLACEHOLDER for _ in range(len(key))] for _ in
range(final_row)]

    text_char_pos = 0
    for key in sorted(key_mapping):
        col = key_mapping[key]
        for row in range(final_row):
            character_matrix[row][col] = text[text_char_pos]
            text_char_pos += 1

    return
"".join(list(chain.from_iterable(character_matrix))).replace(PLACEHOLDER,"")

def run_columnar_transposition_cipher_encrypt_dialog():
    """
    Runs dialog to facilitate Columnar Transposition Cipher Encryption
    """
    text = input("Please enter Text : ")
    key = input("Please enter Key : ")

```

```

encrypted_text = columnar_transposition_cipher_encrypt(text,key)
print(f"Encrypted Text : {encrypted_text}")
pass

def run_columnar_transposition_cipher_decrypt_dialog():
    """
    Runs dialog to facilitate Columnar Transposition Cipher Decryption
    """
    text = input("Please enter Text : ")
    key = input("Please enter Key : ")
    decrypted_text = columnar_transposition_cipher_decrypt(text,key)
    print(f"Decrypted Text : {decrypted_text}")
    pass

def run_menu():
    """
    Runs menu
    """
    try:
        choice = int(input("Columnar Transposition Cipher\n1. Encrypt Text\n2. Decrypt Text\nEnter your choice : "))
        if choice not in (0,1,2):
            raise Exception(ERRORS.CHOICE)
    except Exception as e:
        print(f"Failed to run Program. Error : {e}")
        return

    try:
        if choice == 1:
            run_columnar_transposition_cipher_encrypt_dialog()
        elif choice == 2:
            run_columnar_transposition_cipher_decrypt_dialog()
    except Exception as e:
        print(f"Failed to run Program. Error : {e}")
        return

if __name__ == "__main__":
    run_menu()

```

```

krhero@arc-warden:/media/krhero/0FB812900FB81290/BTech/Assignments/4th_Year/CNS/Assignment_2$ python3 1.py
Columnar Transposition Cipher
1. Encrypt Text
2. Decrypt Text
Enter your choice : 1
Please enter Text : once upon a time there was a little girl named goldilocks she went for a walk in the forest pretty soon she came upon
a house she knocked and when no one answered she walked right in at the table in the kitchen there were three bowls of porridge goldilocks
was hungry she tasted the porridge from the first bowl
Please enter Key : KRUNAL
Encrypted Text : a e lsn kheeoe aekdwoae dttti ereesoei usshrfhslopthaigaoce i ttncpshsoaeos aritb i tborgoagee do u twl ngohta esto
u n h ndw ankne r lwnhteiret#e eealldi erltrrshe s e n reeha eherelpgdsh atr trwnoiestimlk fwnf y aohcnnnwslinhltttwhofiocsr dpgmf
bcnmr tredswoa op smnuekd eehkg eehcherw dlk yt oe io
krhero@arc-warden:/media/krhero/0FB812900FB81290/BTech/Assignments/4th_Year/CNS/Assignment_2$ python3 1.py
Columnar Transposition Cipher
1. Encrypt Text
2. Decrypt Text
Enter your choice : 2
Please enter Text : a e lsn kheeoe aekdwoae dttti ereesoei usshrfhslopthaigaoce i ttncpshsoaeos aritb i tborgoagee do u twl ngohta e
sto u n h ndw ankne r lwnhteiret#e eealldi erltrrshe s e n reeha eherelpgdsh atr trwnoiestimlk fwnf y aohcnnnwslinhltttwhofiocsr dp
gmfbcnmr tredswoa op smnuekd eehkg eehcherw dlk yt oe io
Please enter Key : KRUNAL
Decrypted Text : once upon a time there was a little girl named goldilocks she went for a walk in the forest pretty soon she came upon a h
ouse she knocked and when no one answered she walked right in at the table in the kitchen there were three bowls of porridge goldilocks wa
s hungry she tasted the porridge from the first bowl

```

2

```

# sample text : once upon a time there was a little girl named goldilocks she went
for a walk in the forest pretty soon she came upon a house she knocked and when no
one answered she walked right in at the table in the kitchen there were three bowls
of porridge goldilocks was hungry she tasted the porridge from the first bowl

from typing import OrderedDict

PLACEHOLDER = "#"

class ERRORS:
    KEY_LENGTH = "Invalid Key! Key Length should be more than 1!"
    KEY_UNIQUE_CHARACTERS = "Invalid Key! Key characters should be unique!"
    CHOICE = "Please enter a valid choice!"

def rail_fence_transposition_cipher_encrypt(text: str, key: str) -> str:
    """
    Encrypts the given text using Key using Rail Fence Transposition Cipher Encryption
    """
    if len(key) <= 1:
        raise Exception(ERRORS.KEY_LENGTH)
    key_mapping = OrderedDict()
    for i in range(len(key)):
        if key[i] in key_mapping:
            raise Exception(ERRORS.KEY_UNIQUE_CHARACTERS)
        key_mapping[key[i]] = i

    final_row = len(key)
    character_matrix = []

```

```

gap1 = 2*final_row - 2
gap2 = 0
for i in range(final_row):
    text_pos = i
    check = 0
    character_row = []
    while text_pos < len(text):
        character_row.append(text[text_pos])
        if check:
            text_pos += gap2 if gap2 != 0 else 2*final_row - 2
        else:
            text_pos += gap1 if gap1 != 0 else 2*final_row - 2
        check ^= 1
    character_matrix.append(character_row)
    gap1 -= 2
    gap2 += 2

encrypted_char_list = []
for key in sorted(key_mapping):
    row = key_mapping[key]
    for value in character_matrix[row]:
        encrypted_char_list.append(value)

return "".join(encrypted_char_list)

```

```

def rail_fence_transposition_cipher_decrypt(text: str, key: str) -> str:
    """
    Decrypts the given text using Key using Rail Fence Transposition Cipher Decryption
    """
    if len(key) <= 1:
        raise Exception(ERRORS.KEY_LENGTH)
    key_mapping = OrderedDict()
    for i in range(len(key)):
        if key[i] in key_mapping:
            raise Exception(ERRORS.KEY_UNIQUE_CHARACTERS)
        key_mapping[key[i]] = i

    final_row = len(key)
    decrypted_text = [PLACEHOLDER for _ in range(len(text))]
    row_order = [key_mapping[key] for key in sorted(key_mapping)]
    pos = 0
    for i in row_order:
        gap1 = 2*final_row - 2*(i+1)
        gap2 = 2*i
        text_pos = i

```

```

        check = 0
        while text_pos < len(text):
            decrypted_text[text_pos] = text[pos]
            if check:
                text_pos += gap2 if gap2 != 0 else 2*final_row - 2
            else:
                text_pos += gap1 if gap1 != 0 else 2*final_row - 2
            check ^= 1
            pos += 1

    return "".join(decrypted_text)

def run_rail_fence_transposition_cipher_encrypt_dialog():
    """
    Runs dialog to facilitate Rail Fence Transposition Cipher Encryption
    """
    text = input("Please enter Text : ")
    key = input("Please enter Key : ")
    encrypted_text = rail_fence_transposition_cipher_encrypt(text, key)
    print(f"Encrypted Text : {encrypted_text}")
    pass

def run_rail_fence_transposition_cipher_decrypt_dialog():
    """
    Runs dialog to facilitate Rail Fence Transposition Cipher Decryption
    """
    text = input("Please enter Text : ")
    key = input("Please enter Key : ")
    decrypted_text = rail_fence_transposition_cipher_decrypt(text, key)
    print(f"Decrypted Text : {decrypted_text}")
    pass

def run_menu():
    """
    Runs menu
    """
    try:
        choice = int(input(
            "Rail Fence Transposition Cipher\n1. Encrypt Text\n2. Decrypt Text\nEnter
your choice : "))
        if choice not in (0, 1, 2):

```

```

        raise Exception(ERRORS.CHOICE)
    except Exception as e:
        print(f"Failed to run Program. Error : {e}")
        return

    try:
        if choice == 1:
            run_rail_fence_transposition_cipher_encrypt_dialog()
        elif choice == 2:
            run_rail_fence_transposition_cipher_decrypt_dialog()
    except Exception as e:
        print(f"Failed to run Program. Error : {e}")
        return

if __name__ == "__main__":
    run_menu()

```

```

krhero@arc-warden:/media/krhero/0FB812900FB81290/BTech/Assignments/4th_Year/CNS/Assignment_2$ python3 2.py
Rail Fence Transposition Cipher
1. Encrypt Text
2. Decrypt Text
Enter your choice : 1
Please enter Text : once upon a time there was a little girl named goldilocks she went for a walk in the forest pretty soon she came upon
a house she knocked and when no one answered she walked right in at the table in the kitchen there were three bowls of porridge goldilocks
was hungry she tasted the porridge from the first bowl
Please enter Key : SVNIT
Encrypted Text : eu ieeal imdloh frl ef rso ao s ncn n nwsee nh nktter ofpg lcsh htdpr refbwcpamh iera dcs w aihote nempauskkaw oae k
rt tte c retewoodgikauyes reohi lon at osn ets hedeeagtbhe rsrl gthd s tr geleok pocneodoshdieiiehb eo seof ono etwstllnglk etawntrsty
heu oh e hnn rdwliha altehnewhel riodeswnr ateigmtrt
krhero@arc-warden:/media/krhero/0FB812900FB81290/BTech/Assignments/4th_Year/CNS/Assignment_2$ python3 2.py
Rail Fence Transposition Cipher
1. Encrypt Text
2. Decrypt Text
Enter your choice : 2
Please enter Text : eu ieeal imdloh frl ef rso ao s ncn n nwsee nh nktter ofpg lcsh htdpr refbwcpamh iera dcs w aihote nempauskkaw oae
sty heu oh e hnn rdwliha altehnewhel riodeswnr ateigmtrt
Please enter Key : SVNIT
Decrypted Text : once upon a time there was a little girl named goldilocks she went for a walk in the forest pretty soon she came upon a h
ouse she knocked and when no one answered she walked right in at the table in the kitchen there were three bowls of porridge goldilocks wa
s hungry she tasted the porridge from the first bowl

```