King Saud University
College of Computer and Information Sciences
Department of Software Engineering

Assignment #2

Software Security Engineering

SWE314

Section: 54971

Lina Alkhodair	**437202486**
Rahaf Alzamil	**437202514**
Noura Alzamil	**437201763**

Cryptosystem:

Keyword Columnar Transposition Cipher.

Encryption description:

Firstly, a keyword is chosen, and based on the keyword length a matrix is created with columns that are equal to the keyword length. When the plain text is entered, the matrix is filled row-wise with plain text's characters, and if there are any extra positions will be replaced with a chosen character such as, 'x'. To produce the encrypted text, it will be read column-wise based on the alphabetical order of the letters in the keyword.

Decryption description:

The exact keyword that was used in the encryption method must be used while performing decryption. To get the column lengths we'd have to divide message length by the keyword length, then fill the message out in the columns again. And finally, reorder the columns by reforming the keyword.

Source code:

```
import math
 # Main
def main():
choice = 0
while choice != 3:
 choice = int(input('Choose \n 1- Encryption \n 2- Decryption \n 3- exit
\n'))

 if choice == 1:
    key = input("Enter key: ")
    message = input("Enter a message: ")
    cipher = encrypt(message, key)
    print("Encrypted message: ", cipher)

 elif choice == 2:
    key = input("Enter key: ")
    message = input("Enter a message: ")
    plainText = decrypt(message, key)
    print("Decrypted message: ",plainText)

 elif choice == 3:
  exit()
 else:
   print("Invalid entry, please try again.")
# Encryption
def encrypt(msg, key):

   msgLength = len(msg)
   msgList = list(msg)

   cols = len(key)
```

```
    rows = int(math.ceil(msgLength / cols))

    empty = int((rows * cols) - msgLength)
    msgList.extend('x' * empty)

    cipher = [''] * cols

    for col in range(cols):
      i = col
      while i < len(msgList):
        cipher[col] += msgList[i]
        i += cols
    #print(cipher)

    keyPointer = 0
    keyList = sorted(list(key))
    cipherText = ""

    for i in range(cols):
      curr = key.index(keyList[keyPointer])
      cipherText += ''.join(cipher[curr])
      keyPointer += 1


     return cipherText
  # Decryption
def decrypt(cipher, key):

    msgLength = len(cipher)
    msgList = list(cipher)
     cols = len(key)

    rows = int(math.ceil(msgLength / cols))
     decipher = []
```

```
  for i in range(rows):
    decipher += [[''] * cols]

  keyPointer = 0
  msgPointer = 0
  keyList = sorted(list(key))
   for i in range(cols):
    curr = key.index(keyList[keyPointer])
    for j in range(rows):
      decipher[j][curr] = msgList[msgPointer]
      msgPointer += 1
    keyPointer += 1

  msg = ""

  for i in range(rows):
    msg += ''.join(decipher[i])
   emptyCounter = msg.count('x')
   if emptyCounter > 0:
      return msg[: -emptyCounter]
   return msg
main()
```

Execution Results:

```
Choose
 1- Encryption
 2- Decryption
 3- exit
1
Enter key: swe
Enter a message: software security
Encrypted message:  fa cixstrsutoweery
Choose
 1- Encryption
 2- Decryption
 3- exit
2
Enter key: swe
Enter a message: fa cixstrsutoweery
Decrypted message:  software security
Choose
 1- Encryption
 2- Decryption
 3- exit
3
```

References:

[1] 2020. [Online]. Available: https://www.youtube.com/watch?v=TOl06u2UQl4&t=229s. [Accessed: 08- Mar-

[2] "Columnar Transposition Cipher", Kaidzohar.blogspot.com, 2020. [Online]. Available: https://kaidzohar.blogspot.com/2017/08/columnar-transposition-cipher-code-in.html. [Accessed: 08- Mar- 2020]. 2020].