1. 明文 加密 密文

N->13 (13x06+3) mod 26 03->D

C->02 (02x06+3) mod 26 15->P

H->07 (07x06+3) mod 26 19->T

U->20 (20x06+3) mod 26 19->T

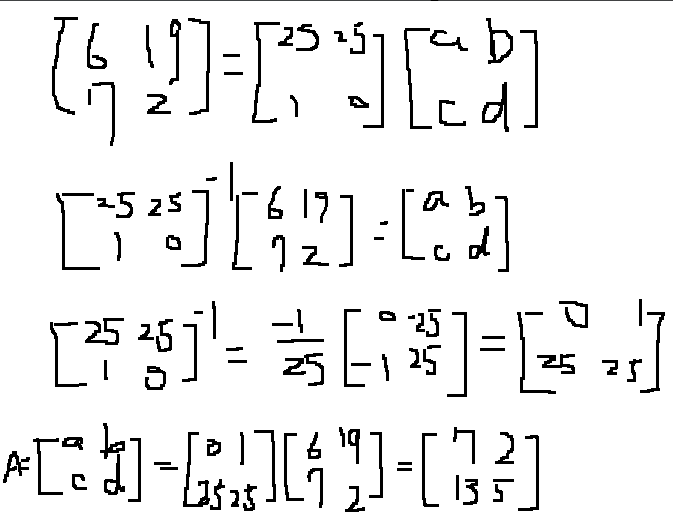
密文 解密 明文

H->07 ((07-2)x3-1) mod 26 19->T

S->18 ((18-2)x3-1) mod 26 14->O

W->22 ((22-2)x3-1) mod 26 24->Y

Z->25 ((25-2)x3-1) mod 26 25->Z



Example: A send to B

(i): A send HELLO with key “XYKLP”

(ii):XOR : XYKLP-> LVVND

(iii):Send to B

(iv):LVVND ->use key “XYKLP” -> HELLO

For sender and receiver, they both have a pre-shared pad of random keys. The pad consists of a series of random characters, with each character being used only once. Key is used to encrypt and decrypt a single message and then is discarded.

advantages : Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code

disadvantages: Problem of making large quantities of random keys, heavily used system might require millions of random characters on a regular basis, Mammoth key distribution problem.



1 2 3 4 5 6 7 8

C O M P U T E R

Encrypt

P R E T C U M O

* PRETCUMO

1 2 3 4 5 6 7 8

V P Y R I C U L

R Y H P T A G O

Decrypt

I L U V C R Y P

T O G R A P H Y

* I LUV CRYPTOGRAPHY

def reverse\_string(input\_str):

    if not isinstance(input\_str, str):

        raise TypeError("String!!!!")

    reversed\_str = ""

    for i in range(len(input\_str) - 1, -1, -1):

        reversed\_str += input\_str[i]

    return reversed\_str

user\_input = input("input: ")

try:

    reversed\_input = reverse\_string(user\_input)

    print("result:", reversed\_input)

except TypeError as e:

    print("錯誤:", e)

執行的方式: 使用python,

直接執行,終端機會出現”input:”字樣,

在後面輸入想轉換的字串,

會印出:”result: 反轉後的字串”



def encryption(text, key):

    input\_text = ""

    for char in text:

        if char.isalpha():

            shift = key % 26

            if char.islower():

                input\_text += chr((ord(char) - ord('a') + shift) % 26 + ord('a'))

            elif char.isupper():

                input\_text += chr((ord(char) - ord('A') + shift) % 26 + ord('A'))

        else:

            input\_text += char

    return input\_text.upper()

def decryption(text, key):

    ciphertext = ""

    for char in text:

        if char.isalpha():

            shift = key % 26

            if char.islower():

                ciphertext += chr((ord(char) - ord('a') - shift) % 26 + ord('a'))

            elif char.isupper():

                ciphertext += chr((ord(char) - ord('A') - shift) % 26 + ord('A'))

        else:

            ciphertext += char

    return ciphertext.lower()

text = input("input: ")

while True:

    try:

        key = int(input("input your key(int): "))

        break

    except ValueError:

        print("The key must be an integer!!!")

encrypted\_text = encryption(text, key)

print("encrypt:", encrypted\_text)

decrypted\_text = decryption(encrypted\_text, key)

print("decrypt:", decrypted\_text)

執行的方式: 使用python,

直接執行,會在terminal出現”input:”,然後輸入想傳遞的訊息,enter,再輸入鑰匙(需是整數),enter,會出現加密後的結果以及解密後的結果:

