組員與分工

學號	姓名	分工	比重
4108056052	徐幸妤	介面設計、說明文件撰寫	25%
4108056049	侯善融	程式撰寫、介面設計	25%
4108029027	陳宥沅	程式撰寫、說明文件撰寫	25%
4108033007	陳榆	介面設計、說明文件撰寫	25%

程式使用說明

請助教先將我們的 zip 檔解壓縮後，會看到以下內容

點擊紅框中的.exe 檔執行即可

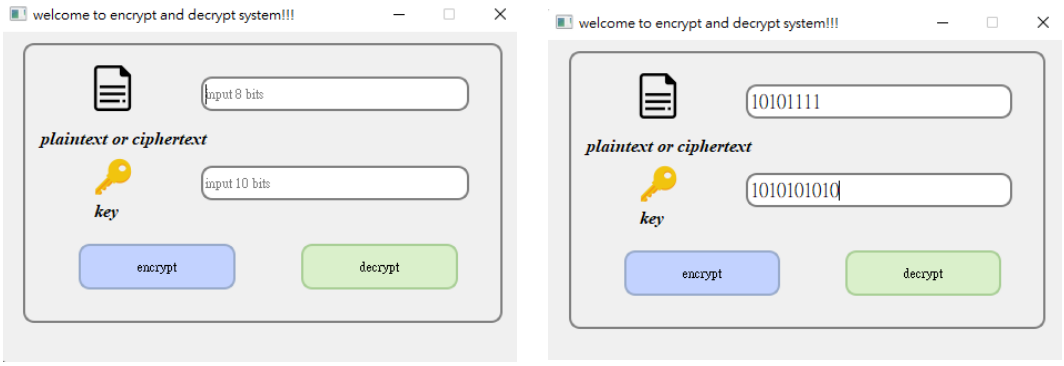
註 1: 有可能會跑出一些警告，繼續執行即可

註 2: 須將 figure 資料夾與執行檔放在同一資料夾

註 3: 如需直接執行 code，請將 figure 資料夾放入 code 資料夾中，並以終端機執行(python controller.py)

S-DES 程式介面操作介紹

點完執行檔後會先看到這個頁面，這裡會有提示告訴使用者輸入的 plaintext/ ciphertext 以及 key 的位元數應該為何



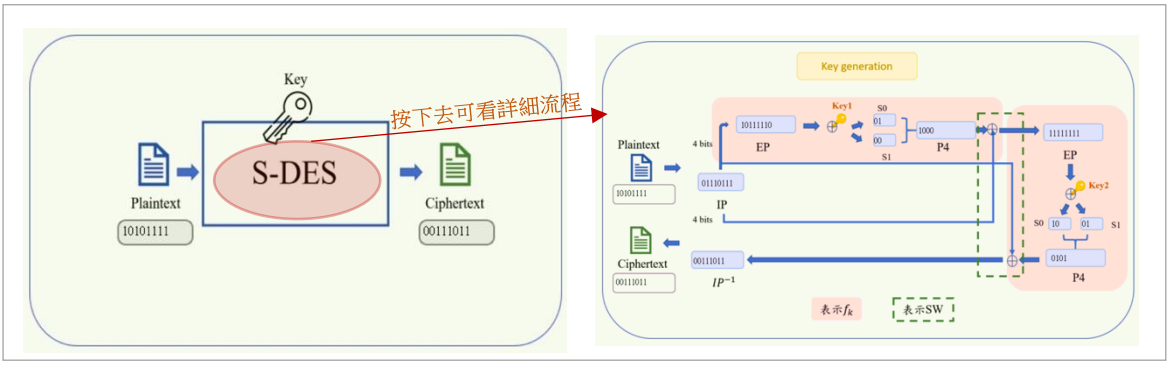
使用者可以自行輸入

1. Input：plaintext 或是 ciphertext (8bits)

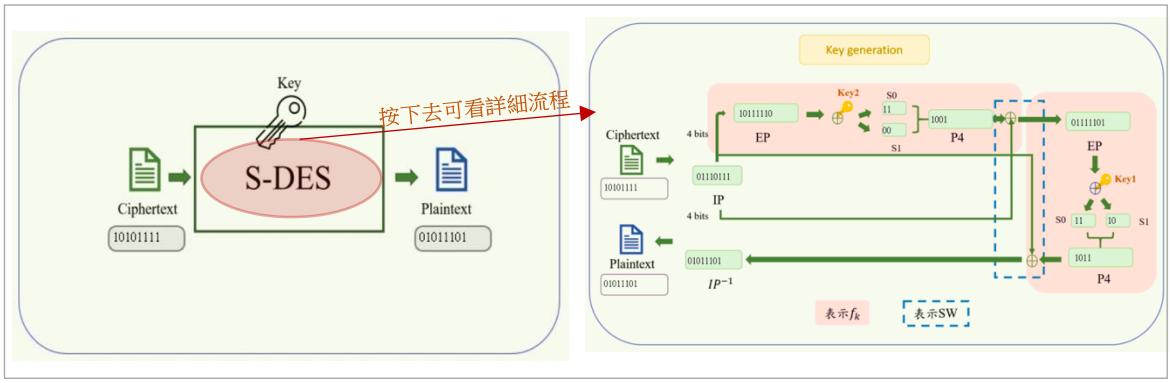
2.Input：key (10bits)

並且可以選擇要做加密或解密

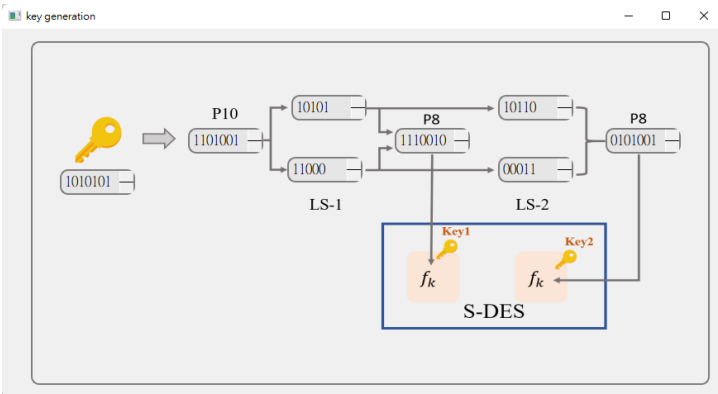
1. 選擇加密 encrypt



2. 選擇解密 decrypt



除此之外，在顯示詳細流程的介面中，可以點擊上方的 Key generation 來查看 Key 產生的過程 Key generation



S-DES 介面防呆機制介紹

在最初的輸入頁面時，我們有設定防呆提醒避免使用者輸入錯誤。

情況 1: 如果 plaintext 或是 ciphertext 的輸入長度不為 8bits 時，便會出現提示告知使用者輸入錯誤。

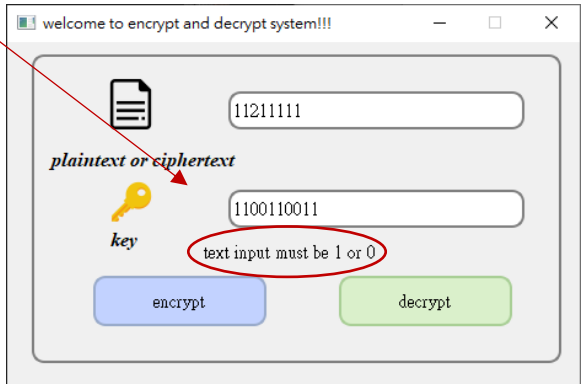
→出現提示

情況 2: 如果 key 輸入長度不為 10bits 時，同樣會出現提示告知使用者輸入錯誤。

→出現提示

情況 3: 如果 key 或 plaintext 或是 ciphertext 輸入不是 0 或 1 時，同樣會出現提示告知使用者輸入錯誤。

→出現提示



S-DES 程式碼說明

code 資料夾中有 6 支程式，分別為

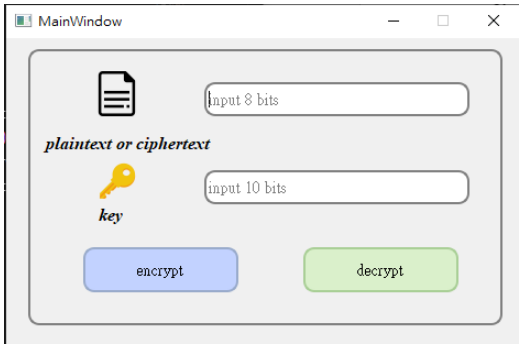
1. main2.py:首頁介面
2. des.py:內容為 S-DES 的程式
3. key_generation.py: 內容為產生 S-DES 中的 KEY 的介面
4. first_scene.py: 首頁點選 encrypt 或 decrypt 後的介面
5. process.py: encrypt 或 decrypt 的詳細流程介面
6. controller.py 為本次作業的主要程式，下面接詳細介紹此程式的程式碼內容

首先會先 import 相關 library 以及上述 1~5 的程式

```
from PyQt5 import QtWidgets, QtGui, QtCore
from main2 import Ui_MainWindow
from proccess import Ui_Form
from first_scene import init_scene
from key_generation import Ui_keygeneration
import des
```

1.主視窗介面設定: (1) encrypt bottom (2) decrypt bottom

```
class MainWindow(QtWidgets.QMainWindow):    #主視窗
    def __init__(self):
        # in python3, super(Class, self).xxx = super().xxx
        super(MainWindow, self).__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        self.setWindowTitle("welcome to encrypt and decrypt system!!!")     #主視窗標題
        self.setup_control()

    def setup_control(self):
        # TODO
        # self.ui.keyinput.setText('Happy World!')
        self.ui.EncryptButton.clicked.connect(self.press_en)    #click encrypt按鍵->執行press_en
        self.ui.DecryptButton.clicked.connect(self.press_de)    #click decrypt按鍵->執行press_de
```

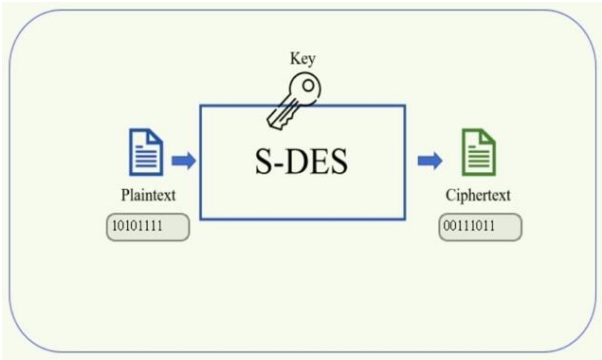


介面主視窗預覽圖

2. encrypt bottom 觸發動作:檢查 input 是否合法(有防呆提醒)

若 input 皆正確就開啟 encrypt 介面

```python
def press_en(self):
    error_flag = 0
    des.KEY = self.ui.keyinput.text()   #get key
    msg = self.ui.textinput.text()      #get plaintext
    if len(des.KEY) != 10:  #防呆
        self.ui.condition_label.setText("KEY input length error")
        error_flag = 1
    else:
        for ch in des.KEY:
            if ch != '1' and ch != '0':     #if key 含非0/1之數字->錯誤
                self.ui.condition_label.setText("key input must be 1 or 0")
                error_flag = 1
                break
    if len(msg) != 8:   #防呆
        self.ui.condition_label.setText("text input length error")
        error_flag = 1
    else:
        for ch in msg:
            if ch != '1' and ch != '0':     #if key 含非0/1之數字->錯誤
                error_flag = 1
                self.ui.condition_label.setText(
                    "text input must be 1 or 0")
                break
    if error_flag == 0:     #key,plaintext皆合法
        first_window.setWindowTitle("encryption")   #進入encrypt畫面
        scene.label.setPixmap(QtGui.QPixmap(
            "figure/encryption_first_sceen.png"))
        scene.textinput.setText(msg)    #print plaintext
        scene.textoutput.setText(des.encrypt(msg))  #print ciphertext
        scene.pushButton.clicked.connect(self.encrypt)  #點按鍵->開啟S-DES加密過程視窗
        first_window.show()
```

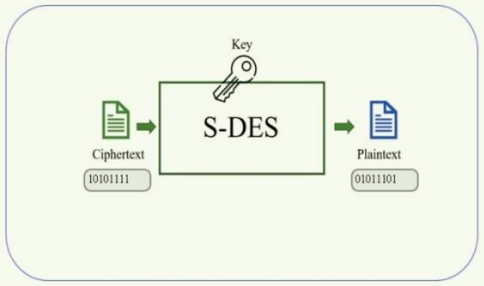


encrypt 介面

3. decrypt bottom 觸發動作:檢查 input 是否合法(有防呆提醒)

若 input 皆正確就開啟 decrypt 介面

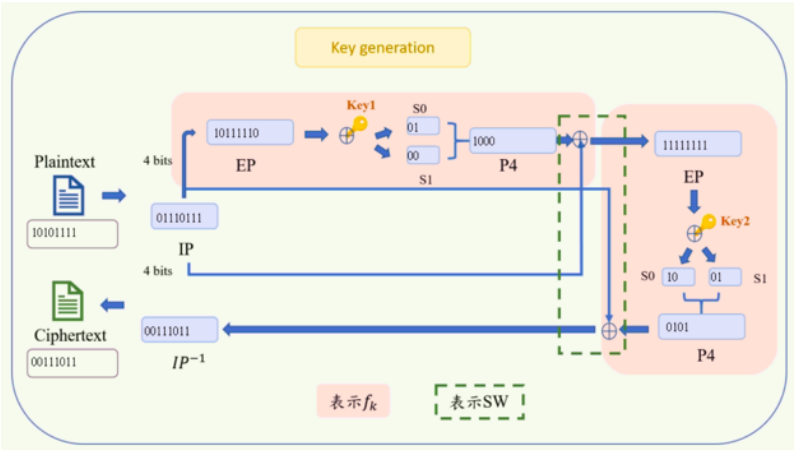
```python
def press_de(self):
    error_flag = 0
    des.KEY = self.ui.keyinput.text()   #get key
    msg = self.ui.textinput.text()      #get ciphertext
    if len(des.KEY) != 10:
        self.ui.condition_label.setText("KEY input length error")
        error_flag = 1
    else:
        for ch in des.KEY:  #防呆
            if ch != '1' and ch != '0':     #if key 含非0/1之數字->錯誤
                self.ui.condition_label.setText("key input must be 1 or 0")
                error_flag = 1
                break
    if len(msg) != 8:   #防呆
        self.ui.condition_label.setText("text input length error")
        error_flag = 1
    else:
        for ch in msg:
            if ch != '1' and ch != '0':     #if key 含非0/1之數字->錯誤
                error_flag = 1
                self.ui.condition_label.setText(
                    "text input must be 1 or 0")
                break
    if error_flag == 0:     #key,ciphertext皆合法
        first_window.setWindowTitle("decryption")   #進入decrypt畫面
        scene.label.setPixmap(QtGui.QPixmap(
            "figure/decryption_first_sceen.png"))
        scene.textinput.setText(msg)    #print ciphertext
        scene.textoutput.setText(des.decrypt(msg))  #print plaintext
        scene.pushButton.clicked.connect(self.decrypt)  #點按鍵->開啟S-DES解密過程視窗
        first_window.show()
```



decrypt 介面

4.S-DES encrypt 詳細流程介面，會分別寫出在每個步驟時的值，讓使用者可以更了解加密流程

```
def encrypt(self):  #S-DES加密過程視窗
    ui.label.setPixmap(QtGui.QPixmap("figure/encryption_structure.png"))
    proccess.setWindowTitle("encryption")
    des.KEY = self.ui.keyinput.text()
    msg = self.ui.textinput.text()
    ui.textinput.setText(msg)
    IP = des.permutate(msg, des.IP) #IP
    ui.IP.setText(IP)

    EP = des.permutate(des.right_half(IP), des.EP)  #EP
    ui.EP_1.setText(EP)
    bits = des.xor(EP, des.key1())  #XOR
    S0_1 = des.lookup_in_sbox(des.left_half(bits), des.S0)  #S0
    ui.S0_1.setText(S0_1)
    S1_1 = des.lookup_in_sbox(des.right_half(bits), des.S1) #S1
    ui.S1_1.setText(S1_1)
    bits = S0_1 + S1_1
    bits = des.permutate(bits, des.P4)  #P4
    ui.P4_1.setText(bits)

    temp = des.f_k(IP, des.key1()) #fk
    after_swap = des.right_half(IP)+temp    #SW

    EP = des.permutate(des.right_half(after_swap), des.EP)  #EP
    ui.EP_2.setText(EP)
    bits = des.xor(EP, des.key2())  #XOR
    S0_2 = des.lookup_in_sbox(des.left_half(bits), des.S0)  #S0
    ui.S0_2.setText(S0_2)
    S1_2 = des.lookup_in_sbox(des.right_half(bits), des.S1) #S1
    ui.S1_2.setText(S1_2)
    bits = S0_2 + S1_2
    bits = des.permutate(bits, des.P4)  #P4
    ui.P4_2.setText(bits)
    bits = des.f_k(after_swap, des.key2())
    ui.IP_inverse.setText(des.permutate(bits + temp, des.IP_INVERSE))   #IP-1

   ui.textoutput.setText(des.encrypt(msg))
   ui.key_button.clicked.connect(self.key_maker)
   proccess.show()
```

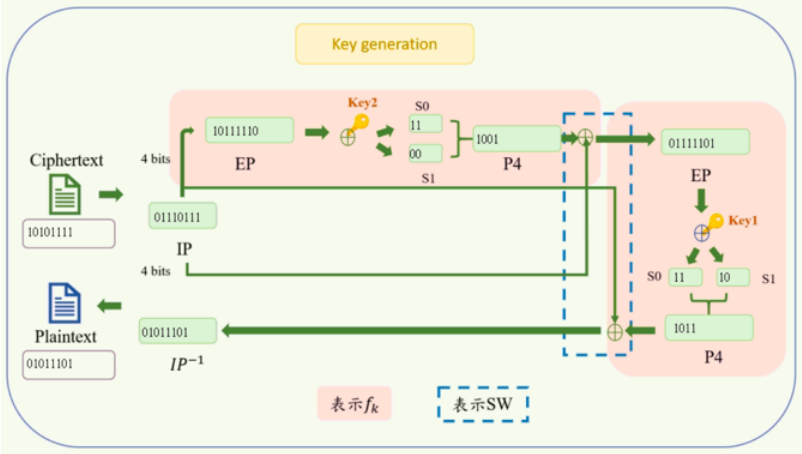


S-DES encrypt 詳細流程介面

5.S-DES decrypt 詳細流程介面，會分別寫出在每個步驟時的值，讓使用者可以細看解密流程

```
def decrypt(self):      ##S-DES解密過程視窗
    ui.label.setPixmap(QtGui.QPixmap("figure/decryption_structure.png"))
    proccess.setWindowTitle("decryption")
    des.KEY = self.ui.keyinput.text()
    msg = self.ui.textinput.text()
    ui.textinput.setText(msg)

    IP = des.permutate(msg, des.IP) #IP
    ui.IP.setText(IP)

    EP = des.permutate(des.right_half(IP), des.EP)  #EP
    ui.EP_1.setText(EP)
    bits = des.xor(EP, des.key2())  #XOR
    S0_1 = des.lookup_in_sbox(des.left_half(bits), des.S0)  #S0
    ui.S0_1.setText(S0_1)
    S1_1 = des.lookup_in_sbox(des.right_half(bits), des.S1) #S1
    ui.S1_1.setText(S1_1)
    bits = S0_1 + S1_1
    bits = des.permutate(bits, des.P4)  #P4
    ui.P4_1.setText(bits)

    temp = des.f_k(IP, des.key2())
    after_swap = des.right_half(IP)+temp    #SW

    EP = des.permutate(des.right_half(after_swap), des.EP)  #EP
    ui.EP_2.setText(EP)
    bits = des.xor(EP, des.key1())
    S0_2 = des.lookup_in_sbox(des.left_half(bits), des.S0)  #S0
    ui.S0_2.setText(S0_2)
    S1_2 = des.lookup_in_sbox(des.right_half(bits), des.S1) #S1
    ui.S1_2.setText(S1_2)
    bits = S0_2 + S1_2
    bits = des.permutate(bits, des.P4)  #P4
    ui.P4_2.setText(bits)
    bits = des.f_k(after_swap, des.key1())
    ui.IP_inverse.setText(des.permutate(bits + temp, des.IP_INVERSE))   #IP-1
```

```
    ui.textoutput.setText(des.decrypt(msg))
    ui.key_button.clicked.connect(self.key_maker)
    proccess.show()
```

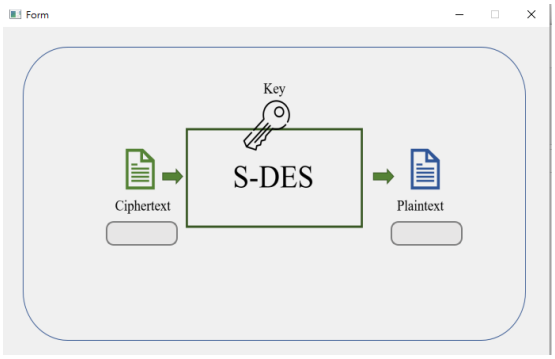


S-DES decrypt 詳細流程介面

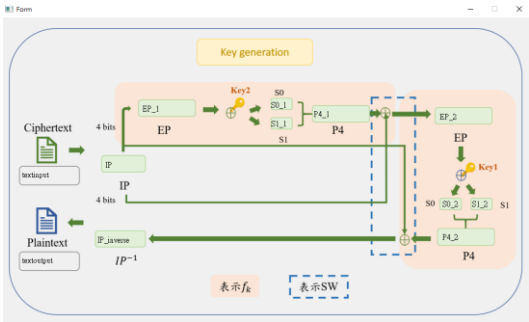
6.主程式

此處的內容主要為每個視窗的介面設定，下面以 1、2、3 的圖示說明分別為那些視窗

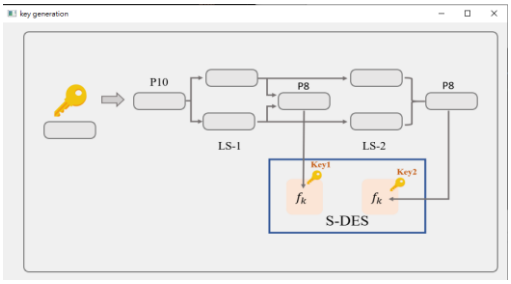
```python
if __name__ == '__main__':
    import sys
    app = QtWidgets.QApplication(sys.argv)

    first_window = QtWidgets.QWidget()  # press encryption or decryption
    scene = init_scene()
    scene.setupUi(first_window)          ①

    proccess = QtWidgets.QWidget()
    ui = Ui_Form()
    ui.setupUi(proccess)                 ②

    key_window = QtWidgets.QWidget()
    key_ui = Ui_keygeneration()
    key_ui.setupUi(key_window)           ③

    window = MainWindow()
    window.show()
    sys.exit(app.exec_())
```



Init scene:①



Proccess 視窗：②



Key generation: ③