DES

Author: Bill BAO

1. Background

The **DES** algorithm is known as the Data Encryption Algorithm (DEA). It is a bit-oriented and shared-key algorithm. They are based on 64 bits. In other words, each text—including plain text, encrypted text, and key—is split up into blocks, each of which is 64 bits. Although key should only be 8 bytes long, we could still obtain the key. The same key or password is used for both encryption and decryption. Encryption and decryption steps will be described briefly in the remaining paragraphs.

Since it uses a shared-key technique, both the encryption and decryption processes use the same secret key stages. 16 sub-keys will be computed for it. Every sub-key will be assigned to the appropriate wheel function.

On the one hand, encryption is composed of the initial permutation the wheel function with 16 times, as well as the inverse initial permutation. As I mentioned above, each wheel function receives a sub-key. It is important that the order of putting sub-key into the wheel function decide whether it is encryption or decryption.

The identical permutation and wheel functions are used for decryption instead, but the sub-keys are arranged in the opposite direction from how they were formed.

2. Requirements and features

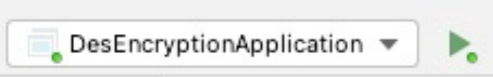
Requirements:

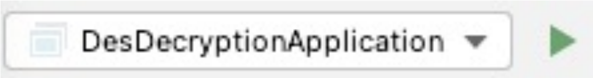
1. User could enter freely regardless of length limitation of plain/cipher text or key.
2. User could use the friendly interface to select file from file system or type the plain text to encrypt/decrypt.
3. There are two independent components that show the encryption and decryption, respectively.
4. DES algorithm is implemented correctly.

Features: Above requirements are all achieved.

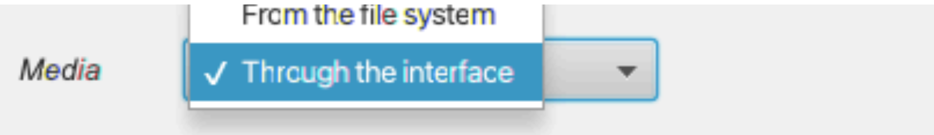
3. User instruction

Firstly, you run the *DesDecryptionApplication* and *DesDecryptionApplication* files to open two components. These two components individually work by accepting the key and plain/cipher text or file.





Next, you should choose which media (text or file) you would like to use DES algorithm.



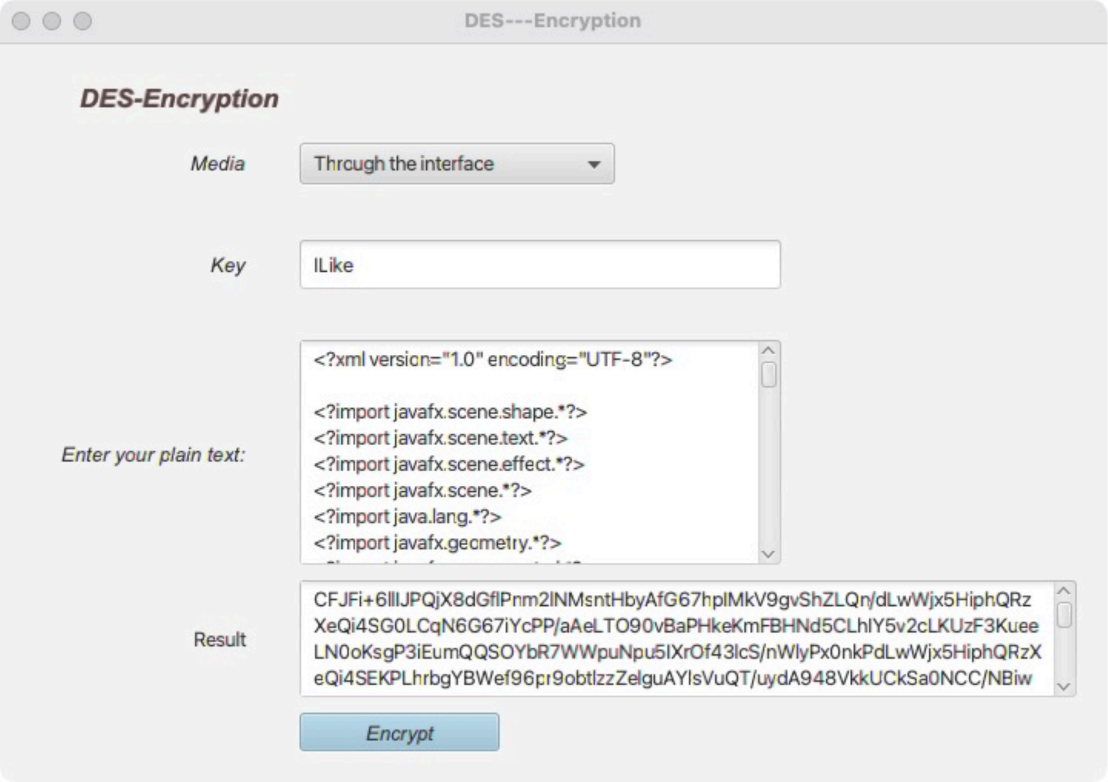
a. Text encryption & decryption:

Encryption

a) Enter the key

b) Enter the plain text you want to keep secret

c) Click 'Encrypt' button.



DES-Encryption

Media: Through the interface

Key: ILike

Enter your plain text:

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.shape.*?>
<?import javafx.scene.text.*?>
<?import javafx.scene.effect.*?>
<?import javafx.scene.*?>
<?import java.lang.*?>
<?import javafx.geometry.*?>
```

Result:

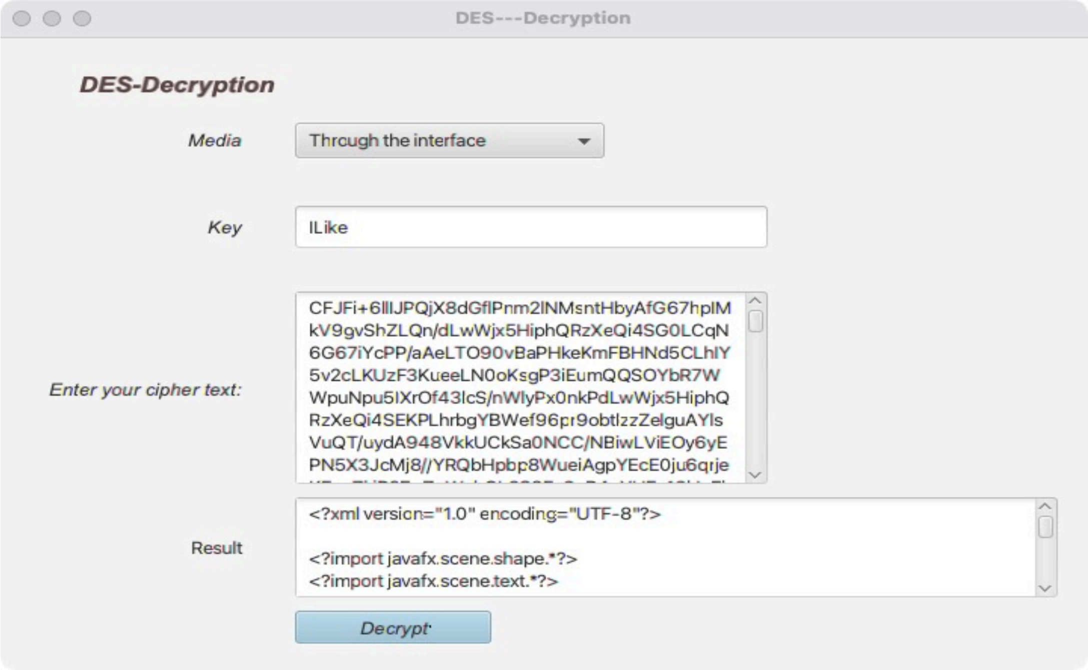
```
CFJFi+6lllJPQjX8dGflPnm2lNMsntHbyAfG67hplMkV9gvShZLQn/dLwWjx5HiphQRz
XeQi4SG0LCqN6G67iYcPP/aAeLTO90vBaPHkeKmFBHNd5CLhlY5v2cLKUzF3Kuee
LN0oKsgP3iEumQQSOYbR7WWpuNpu5lXrOf43lcS/nWlyPx0nkPdLwWjx5HiphQRzX
eQi4SEKPLhrbgYBWef96pr9obtlzzZelguAYlsVuQT/uydA948VkkUCkSa0NCC/NBiw
```

d) Then you get the Cipher text according to your key and plain text

Decryption

e) Use the decryption component, type the key that you have used, and the cipher text you have just got.

f) Click 'Decrypt' button



DES---Decryption

DES-Decryption

Media: Through the interface

Key: ILike

Enter your cipher text:

```
CFJFi+6lllJPQjX8dGflPnm2lNMsntHbyAfG67hplM
kV9gvShZLQn/dLwWjx5HiphQRzXeQi4SG0LCqN
6G67iYcPP/aAeLTO90vBaPHkeKmFBHNd5CLhlY
5v2cLKUzF3KueeLN0oKsgP3iEumQQSOYbR7W
WpuNpu5lXrOf43lcS/nWlyPx0nkPdLwWjx5HiphQ
RzXeQi4SEKPLhrbgYBWef96pr9obtlzzZelguAYls
VuQT/uydA948VkkUCkSa0NCC/NBiwLViEOy6yE
PN5X3JcMj8//YRQbHpbp8WueiAgpYEcE0ju6qrje
```

Result

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.scene.shape.*?>
<?import javafx.scene.text.*?>
```

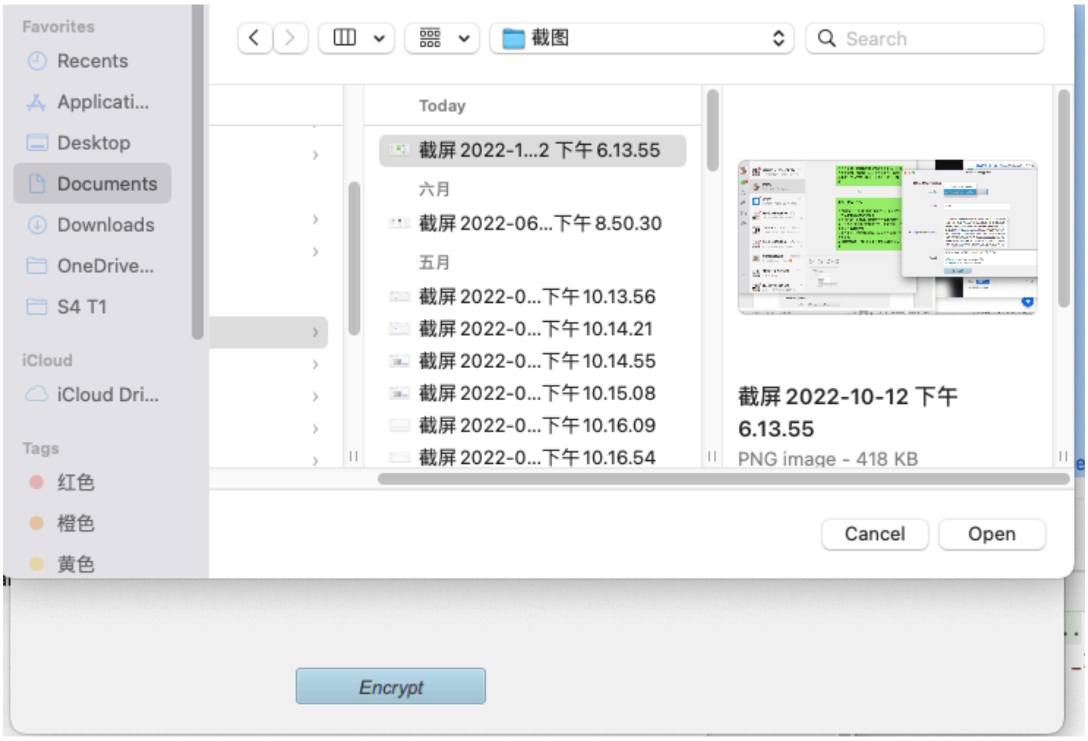
g) The origin text is returned as the result.

b. File encryption & decryption:

Encryption

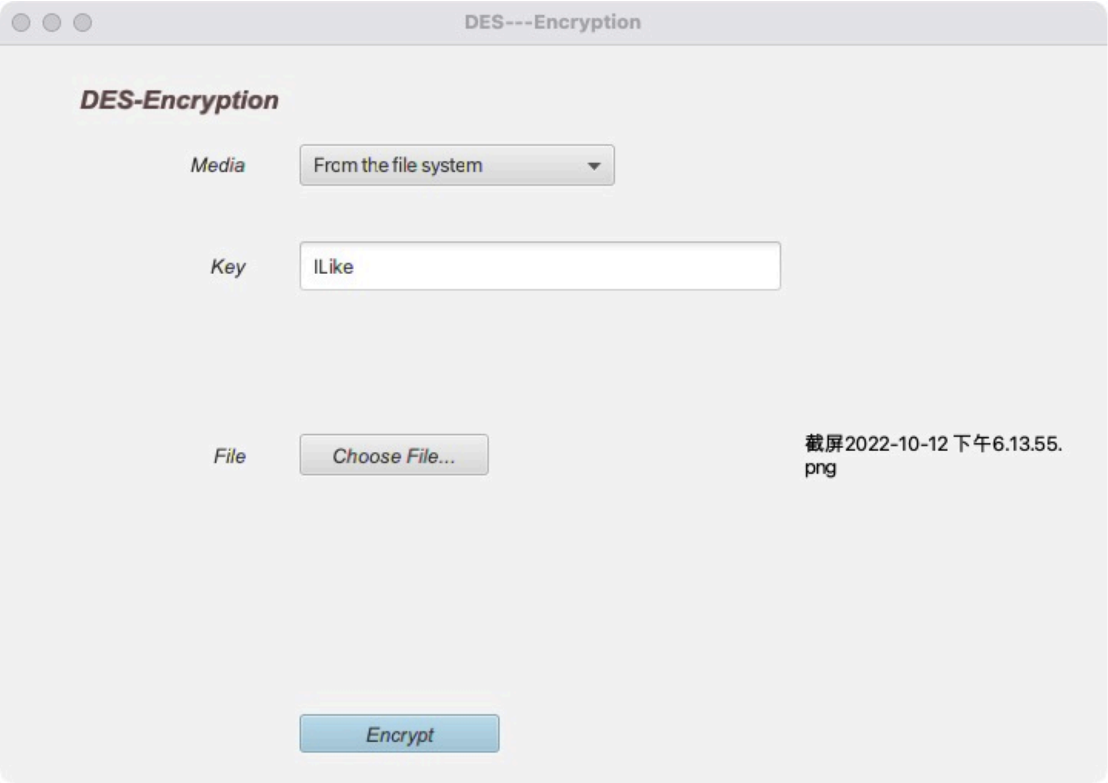
a) Enter the key

b) Click ‘Choose File…’ button, and a window will ask you to select a file (all format is acceptable) you want to encrypt.



Favorites
Recents
Applicati...
Desktop
Documents
Downloads
OneDrive...
S4 T1
iCloud
iCloud Dri...
Tags
红色
橙色
黄色
截图
Search
Today
截屏 2022-1...2 下午 6.13.55
六月
截屏 2022-06...下午 8.50.30
五月
截屏 2022-0...下午 10.13.56
截屏 2022-0...下午 10.14.21
截屏 2022-0...下午 10.14.55
截屏 2022-0...下午 10.15.08
截屏 2022-0...下午 10.16.09
截屏 2022-0...下午 10.16.54
截屏 2022-10-12 下午 6.13.55
PNG image - 418 KB
Cancel
Open
Encrypt

c) Click ‘Open’ button



DES---Encryption

DES-Encryption

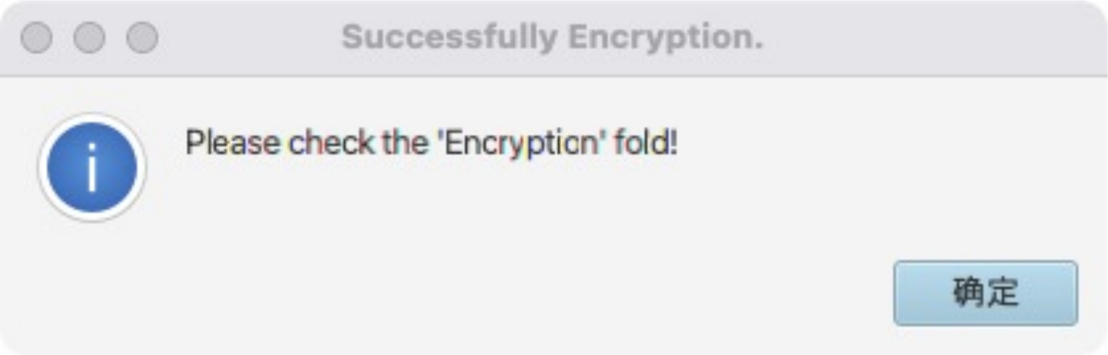
Media: From the file system

Key: ILike

File: 截屏2022-10-12 下午6.13.55.png

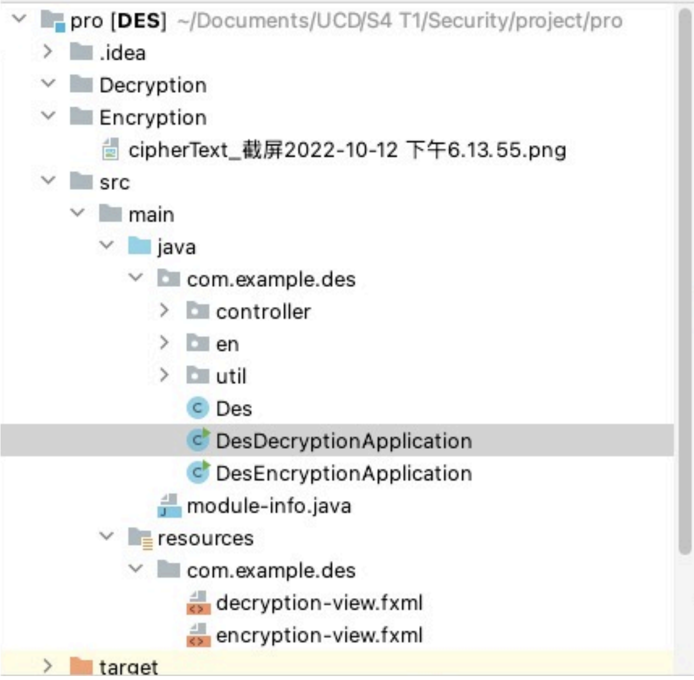
d) It tells you which file you have chosen, and you could click ‘Choose File…’ and repeat above steps to re-select the file.

e) Click ‘Encrypt’ button, and then the encryption will be successfully finished if the key has typed. The below picture tells you that the encrypted file with the prefix ‘cipherText_’ is stored in the ‘Encryption’ fold. The second one shows that.



Successfully Encryption.

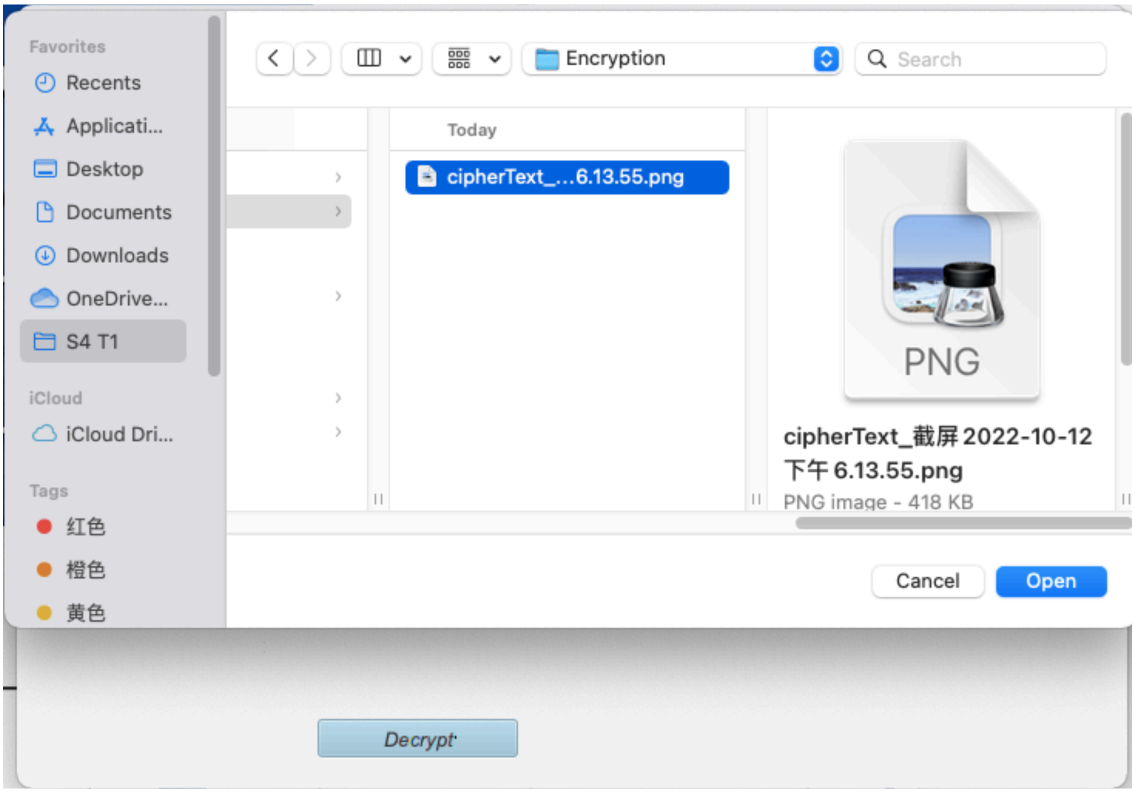
Please check the 'Encryption' fold!

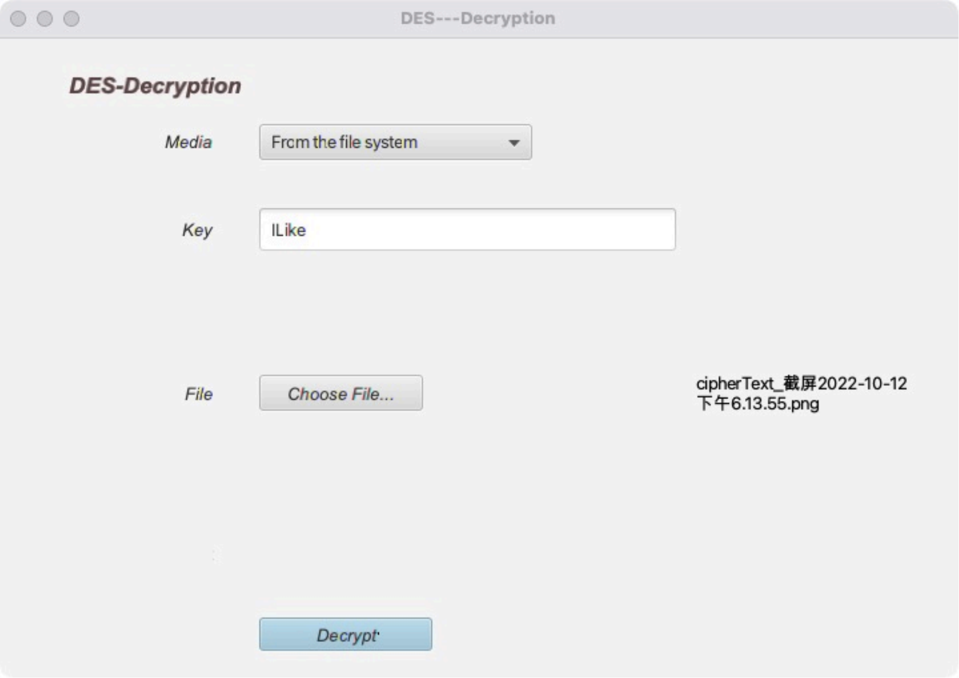


```
pro [DES] ~/Documents/UCD/S4 T1/Security/project/pro
  .idea
  Decryption
  Encryption
    cipherText_截屏2022-10-12 下午6.13.55.png
  src
    main
      java
        com.example.des
          controller
          en
          util
          Des
          DesDecryptionApplication
          DesEncryptionApplication
        module-info.java
      resources
        com.example.des
          decryption-view.fxml
          encryption-view.fxml
  target
```

Decryption

f) Use the decryption component, and repeat the above steps to enter the key and select the encrypted file on the 'Encryption' fold.





DES---Decryption

DES-Decryption

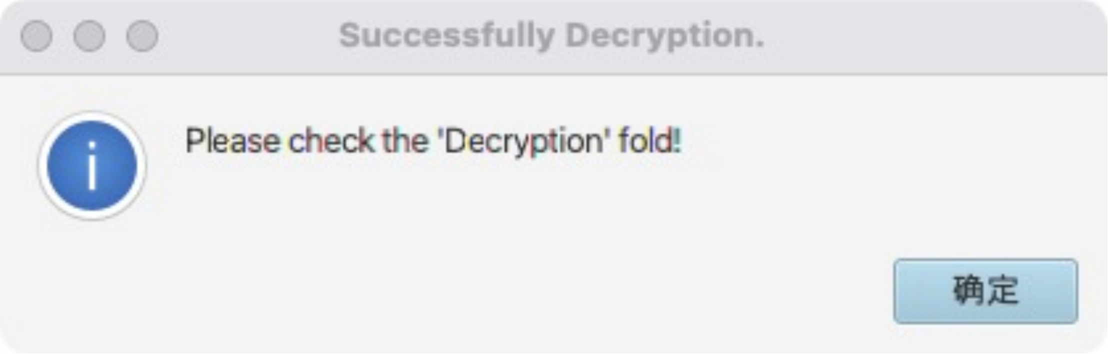
Media: From the file system

Key: ILike

File: Choose File...

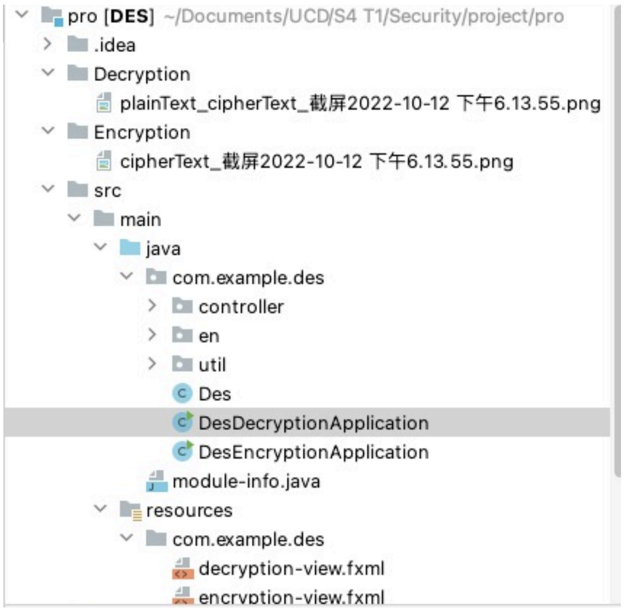
cipherText_截屏2022-10-12 下午6.13.55.png

g) Click ‘Decrypt’ button, and a window tells you the cipher file is decrypted successfully and stored under the ‘Decryption’ directory. Its filename is with the prefix ‘plainText_’.



Successfully Decryption.

Please check the 'Decryption' fold!



```
pro [DES] ~/Documents/UCD/S4 T1/Security/project/pro
  .idea
  Decryption
    plainText_cipherText_截屏2022-10-12 下午6.13.55.png
  Encryption
    cipherText_截屏2022-10-12 下午6.13.55.png
  src
    main
      java
        com.example.des
          controller
          en
          util
          Des
          DesDecryptionApplication
          DesEncryptionApplication
        module-info.java
      resources
        com.example.des
          decryption-view.fxml
          encryption-view.fxml
```