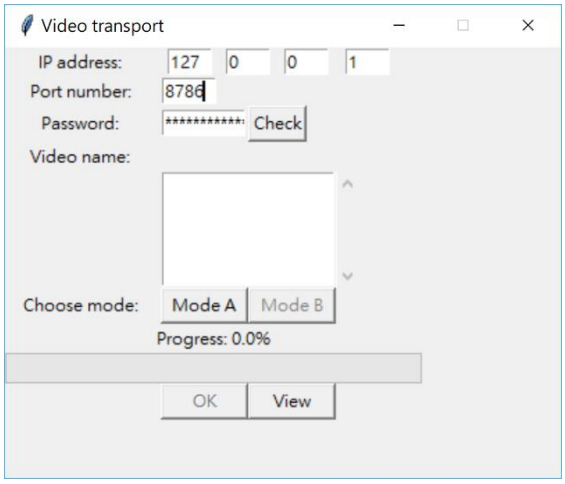
Computer Network Final Project

目錄

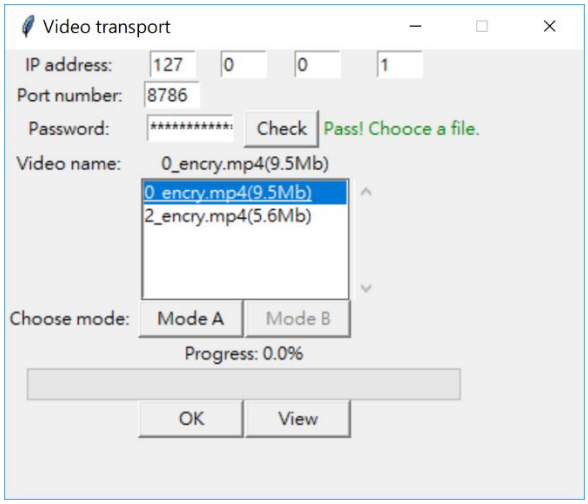
1. 動機

隨著隱私的日加注重，加密傳輸影片便顯得更加重要。然而，若是單純的以亂數加碼呈現，欲攔截的第三方便能輕易辨識加密後的影片。因此，我們想到在網路傳輸影片時，將欲加密的 A 影片藏在另外一個 B 影片下面，使得在傳輸中若是被第三方攔截，檔案看起來就是 B 影片而不被察覺其背後的 A 影片。當使用者拿到影片時，再加以解密，去掉 B 影片而還原成 A 影片。

2. 使用者介面操作

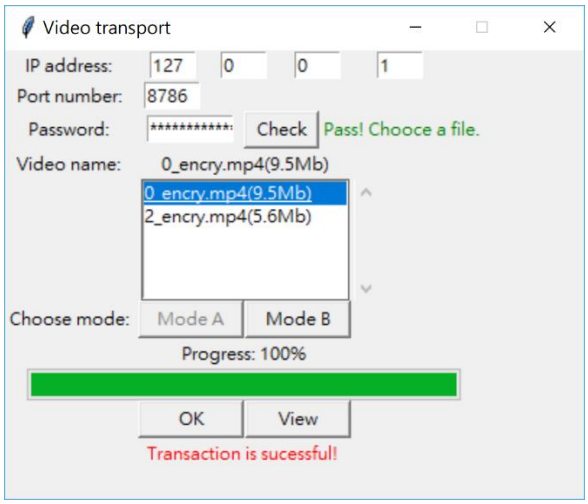


當 client 端要求接收影片時，呈現的介面如上圖所示。首先輸入 server 端的 IP Address，假設 server 的 IP Address 是 140.112.253.33，則以相同的格式分別在四格內輸入 140、112、253、33。接下來輸入對應的 Port Number 以及密碼，而這些資訊將會由 server 連同 IP Address 事先一同告知。

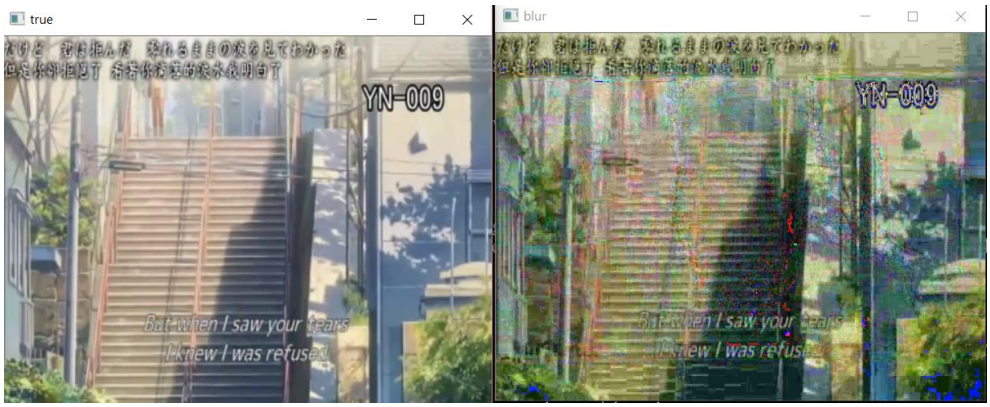


輸入完成後點選"檢查(Check)"驗證密碼。 成功驗證密碼以後，將會顯示 server 端可以傳送的影片清單。選完欲傳送的影片後，即可選擇以 Mode A，也就是

影片加密的方法傳送影片。



傳送完成後，點選 Mode B 進行影片解密。



解密完成後即可點選"View" 觀看影片。

3. 程式內容

使用程式語言:python(版本 3.6)

使用套件:Server 和 Client 端相同
opencv
numpy
opencv
pillow
tkinter
pycrypto

README:

Server:

(1) 加密影片:安裝 python 及所需套件後，確認好影片及覆蓋影片都在 network 資料夾底下，修改 server_process.py 影片的路徑，在執行即可

(2) 連線:安裝 python 及所需套件後，在 CMD 執行 server.py 即可。

Client:安裝所需套件後，執行 User_interface.py 即可

code:

server_process.py

影片預處理部分(Server 端):

resize 函式修改輸入圖片的矩陣，修改成需要的大小，再輸出矩陣，確保被蓋與覆蓋的圖片大小一樣。

```
def resize(image,width,height):
    try:
        im = Image.fromarray(image)
        new = im.resize((width,height),Image.BILINEAR)
        return np.array(new)
    except:
        return 0
```

先輸入影片(路徑)、淡化倍率、

```
def encry(video_path, cover_path, transform_number, method = 0):
    out_name = video_path.split('/')[-1][:-4]+"_"+cover_path.split('/')[-1][:-4]
    count = 0


    origin_cap = cv2.VideoCapture(video_path)
    cover_cap = cv2.VideoCapture(cover_path)
    # fps = origin_cap.
```

準備輸出覆蓋好的影片

```
    out = cv2.VideoWriter("../network/"+video_path.split('/')[-1][:-4]+"_encry.mp4",cv2.VideoWriter_fourcc(*'avc1'),24.0,(480,360))
    #out = cv2.VideoWriter("../mp4/"+out_name+"_encry1.avi",cv2.VideoWriter_fourcc('M','J','P','G'),20.0,(480,360))
```

將兩個影片全部讀取成一張張的圖片

```
    while(origin_cap.isOpened()):
        origin_ret, origin_frame = origin_cap.read()
        cover_ret, cover_frame = cover_cap.read()
```

為了避免覆蓋影片比較短，所以當覆蓋圖片用完時，再讀取一次覆蓋影片，讀取成一張張的圖片。

```
        if (cover_ret == False):
            cover_cap.release()
            cover_cap = cv2.VideoCapture(str(cover_path))
            cover_ret, cover_frame = cover_cap.read()
```

將兩種圖片改成一樣大小

```
origin_frame = resize(origin_frame,480,360)
cover_frame = resize(cover_frame,480,360)
```

將原始圖片的 RGB 除以淡化的倍率，覆蓋影片的 RGB 除以二，再加在一起。
覆蓋影片除以二是避免 RGB 相加後會超過 255。

```
blur = np.uint8(np.floor(origin_frame / transform_number))

blur = blur+np.uint8(np.ceil(cover_frame / 2))
```

進行到原始圖片用完為止。

```
if(origin_ret == True):
    out.write(blur)
else:
    break
#cv2.imshow("blur",blur)
#if cv2.waitKey(1) & 0xFF == ord('q'):
#    count+=1
#    break

origin_cap.release()
cover_cap.release()
out.release()
cv2.destroyAllWindows()
```

Client 端
一樣會用到 resize

```
def resize(image,width,height):
    try:
        im = Image.fromarray(image)
        new = im.resize((width,height),Image.BILINEAR)
        return np.array(new)
    except:
        return 0
```

將接收到的影片跟覆蓋影片讀取成一張張的圖片

```
def decry(video_path, cover_path,transform_number):
    out_name = video_path.split('/')[-1][:-4]+"_"+cover_path.split('/')[-1][:-4]
    #out = cv2.VideoWriter(out_name+"_decry.avi",cv2.VideoWriter_fourcc('M','J','P','G'),20.0,(480,360))
    #out = cv2.VideoWriter("../mp4/"+out_name+"_decry.mp4",cv2.VideoWriter_fourcc(*'mp4v'),25.0,(480,360))
    origin_cap = cv2.VideoCapture(video_path)
    cover_cap = cv2.VideoCapture(cover_path)
```

覆蓋圖片用完時，再讀取一次覆蓋影片，讀取成一張張的圖片。

```
while(origin_cap.isOpened()):
    origin_ret, origin_frame = origin_cap.read()
    cover_ret, cover_frame = cover_cap.read()
    if (cover_ret == False):
        cover_cap.release()
        cover_cap = cv2.VideoCapture(str(cover_path))
        cover_ret, cover_frame = cover_cap.read()
```

覆蓋圖片改成跟接收到的一樣大小

```
        cover_frame = resize(cover_frame,479,359)
```

扣掉覆蓋圖片，乘以淡化倍率還原成原本的圖片

```
        blur = origin_frame - np.uint8(np.ceil(cover_frame / 2))
        blur*=transform_number
```

進行到接收到的圖片用完為止

```
        if(origin_ret == True):
            pass
            #out.write(blur)
        else:
            break
```

設定按空白鍵暫停播放，q 跟 esc 結束影片

```
        cv2.imshow("blur",blur)
        if cv2.waitKey(15) & 0xFF == ord(' '):
            while 1:
                #cv2.imshow("blur",blur)
                if cv2.waitKey(25) & 0xFF == ord(' '):
                    break
        if cv2.waitKey(15) & 0xFF == ord('q'):
            break
        if cv2.waitKey(15) == 27:
            break

    origin_cap.release()
    cover_cap.release()
    #out.release()
    cv2.destroyAllWindows()
```

連線部分

Server.py

引入所需模組

```
from Crypto.Cipher import AES
from socket import *
import sys
import os
from Crypto.PublicKey import RSA
```

Server 要先有 User 的 RSA 公鑰，從本地端讀取檔案(mykey.pem)

```
with open("mykey.pem",'r') as f:
    publickey = RSA.importKey(f.read())
```

建立一個 tcp socket 並指定 port number 為 8786

```
tcpSerSock = socket(AF_INET, SOCK_STREAM)
tcpSerSock.bind(('', 8786))
tcpSerSock.listen(1)
```

Server 接受連線，但是只接受一開始的 request code，request code 共有三種模式

(1)101:準備接收檔案名，然後傳送指定檔案

(2)200:準備接收公鑰，公鑰符合則傳送檔案列表，不符合則傳回"ERROR"

```
while 1:
    print('Ready to serve...')
    tcpCliSock, addr = tcpSerSock.accept()
    print('Received a connection from:', addr)
    try:
        request_code = tcpCliSock.recv(1024).decode('utf-8')
        if request_code in ["101","200"]:
            print(request_code)
            tcpCliSock.send(request_code.encode())
            message = tcpCliSock.recv(1024).decode('utf-8')

            if request_code == "101":
                f = open("./"+message,'rb')
                data = f.read(1024)
                while(data):
                    print("Sending")
                    tcpCliSock.send(data)
                    data = f.read(1024)
                print("transaction sucessful")
                tcpCliSock.shutdown(SHUT_WR)
            elif request_code == "200":
                message = RSA.importKey(message, passphrase=None)
                if message == publickey:
                    file_list = os.listdir('.')
                    file_list = [file for file in file_list if file.endswith("encry.mp4")]
                    file_list = ([file+"("+str(float(os.path.getsize("./"+file))/1024/1024)[:3]
                        +"Mb)" for file in file_list])
                    file_list.sort()
                    file_list = ','.join(file_list).encode()
                    print(message)
                    print(file_list)
                    tcpCliSock.send(file_list)
                    tcpCliSock.shutdown(SHUT_WR)
                else:
                    print(message)
                    data = "ERROR"
                    tcpCliSock.send(data.encode())
                    tcpCliSock.shutdown(SHUT_WR)
        else:
            tcpCliSock.shutdown(SHUT_WR)
    except:
        pass
```

User_interface(使用者介面)

引入所需模組

```
from tkinter import Tk, Label, Entry, Button, StringVar, IntVar, DoubleVar, messagebox
from tkinter import DISABLED, NORMAL, filedialog, Listbox, END, ANCHOR, Scrollbar
from tkinter import N, S, W
from tkinter.ttk import Progressbar
import os
from socket import *
from PIL import Image
import numpy as np
import cv2
```

建立一個應用程式介面

```
root = Tk()
root.title("Video transport")
root.geometry("400x310")
root.resizable(0,0)
```

建立輸入 IP 欄位的 Entry

```
## IP field  row = 0

Label(root, text="IP address: ").grid(row=0,column=0)

def limitSizeip(*argv):
    for i in range(4):
        value = ip[i].get()
        if len(value) > 3: ip[i].set(value[:3])
def Jumpip(*argv):
    for i in range(4):
        value = ip[i].get()
        if len(value) == 3:
            if i <= 3:
                try:
                    ip[i].trace_vdelete("w",ip[i].trace_id)
                    ip[i].trace('w',limitSizeip)
                    entry_list[i+1].focus()
                except:
                    pass

ip = []
ip_trace_id_list=[]
for i in range(4):
    ip.append(StringVar())
    ip[i].trace_id = ip[i].trace('w',Jumpip)

column = 1
entry_list=[]
for i in range(4):
    dot = Label(root,text=".")
    entry = Entry(root, textvariable=ip[i], width=4)
    entry_list.append(entry)
    entry.grid(row=0,column=column,columnspan=2)
    column+=2
```

新增 Port 欄位

```
## port number row = 1
row = 1
port = StringVar()

Label(root, text="Port number: ").grid(row=row, column=0)
port_ent = Entry(root, textvariable=port, width=5).grid(row=row, column=1,columnspan=2)
```

密碼欄位:如果路徑下有偵測到公鑰文件，則自動輸入密碼。然後傳送 request code “200”取回 Server 端的影片目錄並顯示在介面上。

```
## password   row = 2
row = 2
password_str = StringVar()

def hello():
    mode.set(2)
    try:
        if mode.get() == 2:
            get_file('.'.join(str(e.get()) for e in ip), port.get(), password_str.get(),"200")
    except:
        messagebox.showinfo("Error!", "Network Error!")
try:
    with open("mykey.pem",'r') as f:
        password_str.set(f.read())
except:
    pass

password_lab = Label(root, text="Password: ")
password = Entry(root, textvariable=password_str, show="*",width=8)
password_lab.grid(row=row, column=0)
password.grid(row=row, column=1, columnspan=3)
Button(root, text='Check', width=6, command=hello).grid(row=row, column=4,columnspan=2)
```

其中的 get_file 如下，輸入為 IP 位置--Host(string)、連接埠--PORT(string)、檔案路徑--file_name(string)、request_code(string)。

(1) request code 為 200 時，接收訊息為一個影片列表，並顯示在底下

(2) request code 為 101 時，接收訊息為一個影片檔案，隨著 mode 模式不同而接收不同檔案。mode 為 0 的時候，是接收加密過後的影片；mode 為 1 的時候，接收覆蓋的影片。

```
def get_file(HOST, PORT, file_name, request_code):
    PORT = int(PORT)
    tcpCliSock = socket(AF_INET, SOCK_STREAM)
    tcpCliSock.connect((HOST,PORT))
    if mode.get() == 0:
        data = file_name[:-7]
    elif mode.get() == 1:
        data = "cover.mp4"
    elif mode.get() == 2:
        data = file_name
    tcpCliSock.send(request_code.encode())
    response = tcpCliSock.recv(1000).decode('utf-8')
    if response == request_code:
        if len(data) != 0:
            tcpCliSock.send(data.encode())
            if request_code == "100":
                file_size = tcpCliSock.recv(1000).decode('utf-8')
                try:
                    file_size_int.set(int(file_size))
                    file_size_str = str(int(file_size)/1024/1024)[:3]+"Mb"
                    Label(root, text="OK, "+file_size_str,width=8, fg="red").grid(row=1, column=5,columnspan=3)
                except ValueError:
                    Label(root, text=file_size,width=8, fg="red").grid(row=1, column=5,columnspan=3)
            elif request_code == "200":
                file_list = tcpCliSock.recv(1024).decode('utf-8')
                if file_list != "ERROR":
                    file_list.split(',')
                    #print(file_list)
                    file_list = file_list.split(',')
                    Label(root, text="Pass! Chooce a file.", fg='green').grid(row=2, column=6,columnspan=4)
                    var_file_list.set(file_list)
                else:
                    Label(root, text="Wrong password!", fg='red').grid(row=2, column=6,columnspan=4)
        else:
            f = open("./"+data,'wb')
            stream = tcpCliSock.recv(1024)
            count = 1
            while(stream):
                f.write(stream)
                stream = tcpCliSock.recv(1024)
                #print(video_name.get()[-6:-3])
                ratio.set(ratio.get()+300/1024/float(video_name.get()[-6:-3]))
                print(ratio.get())
                if ratio.get() >= 300:
                    ratio_per.set("Progress: 100%")
                else:
                    ratio_per.set("Progress: "+str(ratio.get()/3.0)[:3]+"%")
                if count % 100 == 0:
                    root.update()
                    count = 1
                else:
                    count+=1
            f.close()
            print("Transaction sucessful")
            succ.grid(row=9, column=0, columnspan=10)
    else:
        Label(root, text="Wrong password!", fg="red").grid(row=2, column=6,columnspan=4)
    tcpCliSock.close()
```

接受 Server 端傳來的影片列表後，讓使用者選擇一個檔案準備下載

```
## video name row = 3
row = 3
video_name = StringVar()
Label(root, text="Video name: ").grid(row=row, column=0)
Label(root, textvariable=video_name, width=20).grid(row=row, column=1, columnspan=6)


## video list row = 4
row = 4
var_file_list = StringVar()
fb = Listbox(root, listvariable=var_file_list)
fb.config(height=5)
def fileSelection(self):
    name = fb.get(fb.curselection())
    video_name.set(name)

fb.bind("<ButtonRelease-1>", fileSelection)
fb.grid(row=row, column=1, columnspan=6)
scrollbar = Scrollbar(root, orient="vertical")
scrollbar.config(command=fb.yview)
scrollbar.grid(row=row, column=7, sticky=N+S+W)

fb.config(yscrollcommand=scrollbar.set)
```

選擇 mode 模式。模式 A 代表是加密模式，按下 OK 準備下載在 Server 端的影片檔案，下載完後按下 View 後會看到加密後的影片；模式 B 代表解密模式，按下 OK 準備下載在 Server 端作加密的覆蓋影片，下載完按下 View 後看到解密後的影片。

```
## mode button row = 5
row = 5
mode = IntVar()
mode.set(2)
def button_1():
    mode.set(0)
    button1['fg']="black"
    button2['fg']="gray"
    OK_but['state']=NORMAL
    print(mode)
def button_2():
    mode.set(1)
    button1['fg']="gray"
    button2['fg']="black"
    OK_but['state']=NORMAL
    print(mode)

Label(root, text="Choose mode: ").grid(row=row, column=0)
button1 = Button(root, text='Mode A', width=9, command=button_1, fg="black")
button2 = Button(root, text='Mode B', width=9, command=button_2, fg="gray")
button1.grid(row=row, column=1, columnspan=3)
button2.grid(row=row, column=4, columnspan=3)
button1.bind('<Button-1>',button1)
button2.bind('<Button-1>',button2)
```

下載進度條

```
## progress bar  row = 6
row = 6
ratio = DoubleVar()
ratio_per = StringVar()
ratio_per.set("Progress: "+str(ratio.get()/3.0)[:3]+"%")
ratio_per_label = Label(root, textvariable=ratio_per)

ratio_per_label.grid(row=row, column=0, columnspan=10)

row = 7

file_size_int = IntVar()
progress = Progressbar(orient = 'horizontal', length=300, mode = 'determinate', variable=ratio, maximum=300)
progress.grid(row = row, column=0, columnspan=10)
```

OK 按鈕和 View 按鈕

```
def press_button3():
    ratio.set(0)
    ratio_per.set("Progress: "+str(ratio.get()/3.0)[:3]+"%")
    try:
        if mode.get() == 0:
            get_file('.'.join(str(e.get()) for e in ip), port.get(), video_name.get(),"101")
        elif mode.get() == 1:
            get_file('.'.join(str(e.get()) for e in ip), port.get(), "cover.mp4","101")
            ratio.set(300)
            ratio_per.set("Progress: 100%")
            root.update()
        else:
            messagebox.showinfo("Error", "Choose a mode")
    except:
        messagebox.showinfo("Error", "Unknown Error")

def press_button4():
    if mode.get() == 0:
        os.system("open "+video_name.get())
    if mode.get() == 1:
        decry(video_name.get()[:-7],"cover.mp4",17)

OK_but = Button(root, text='OK', width=9, command=press_button3, state=DISABLED)
view_but = Button(root, text='View', width=9, command=press_button4)

OK_but.grid(row=row, column=1, columnspan=3)
view_but.grid(row=row, column=4, columnspan=3)
```

最後的 decry 函數，對於使用者請求的影片作解密並另開一個視窗播放。

```
def decry(video_path, cover_path,transform_number):
    out_name = video_path.split('/')[-1][:-4]+"_"+cover_path.split('/')[-1][:-4]
    origin_cap = cv2.VideoCapture(video_path)
    cover_cap = cv2.VideoCapture(cover_path)
    while(origin_cap.isOpened()):
        origin_ret, origin_frame = origin_cap.read()
        cover_ret, cover_frame = cover_cap.read()
        if (cover_ret == False):
            cover_cap.release()
            cover_cap = cv2.VideoCapture(str(cover_path))
            cover_ret, cover_frame = cover_cap.read()
        cover_frame = resize(cover_frame,479,359)
        blur = origin_frame.astype(np.int16) - np.uint16(np.floor(cover_frame / 2))
        blur[blur<0]=0
        blur = np.uint8(blur)
        blur*=transform_number
        cv2.imshow("blur",blur)
        if cv2.waitKey(15) & 0xFF == ord(' '):
            while 1:
                if cv2.waitKey(25) & 0xFF == ord(' '):
                    break
        if cv2.waitKey(15) & 0xFF == ord('q'):
            break
        if cv2.waitKey(15) == 27:
            break
    origin_cap.release()
    cover_cap.release()
    cv2.destroyAllWindows()
```

其中 resize 函式修改輸入圖片的矩陣，修改成需要的大小，再輸出矩陣，確保被蓋與覆蓋的圖片大小一樣。

```
def resize(image,width,height):
    try:
        im = Image.fromarray(image)
        new = im.resize((width,height),Image.BILINEAR)
        return np.array(new)
    except:
        return 0
```

4. 困難挑戰

在加密輸出影片的時候，會使用到製作影片壓縮的技術，但這會讓影片失真，再減掉的時候會有些微的誤差，導致無法完全去除覆蓋的影片。但是如果不用影片壓縮的技術，會讓輸出的影片檔案變得是原來的好幾十倍，而且也不能保證能完全去除，這會導致需要傳輸高精細的影片時會出問題。

5. 總結、未來展望

在本次報告呈現中，我們將 A 影片藏在 B 影片中傳給客戶，取代複雜的加密方式，用別的影片以假亂真，混淆視聽。然而，加密影片的壓縮方法對影片最後解壓縮的品質影響甚鉅。因此，還需要尋找更良好的壓縮倍率。