PicPic

发表于 2021-05-25 更新于 2021-05-26 分类于 Challenge ， 2021 ， 第四届红帽杯网络安全大赛 ， Misc
Challenge | 2021 | 第四届红帽杯网络安全大赛 | Misc | PicPic

点击此处获得更好的阅读体验

WriteUp来源

来自c10udlnk的博客

题目描述

> *你想成为CV大师嘛？*

题目考点

解题思路

记一道折腾了我大半天的题，今天红帽杯Misc的PicPic，最后六点多一点做出来了（比赛六点结束□血亏323pt）。感觉做这个题的时候把所有关于fft代码实现的博客和加解密相关的论文都翻了个遍，短时间内应该不想再看到它了（

challenge 1

题目里面可以看到是一个challenge 1的文件夹，和next_challenge.rar的加密压缩包，合理猜测是用challenge 1解出压缩包的密码。

很容易就能发现题目给的create.py就是生成r、1.mkv和2.mkv的源码，逻辑也很容易走：

```
import os
import cv2
import struct
import numpy as np

def mapping(data, down=0, up=255, tp=np.uint8):
    data_max = data.max()
    data_min = data.min()
    interval = data_max - data_min
    new_interval = up - down
    new_data = (data - data_min) * new_interval / interval + down
    new_data = new_data.astype(tp)
    return new_data

def fft(img):
    fft = np.fft.fft2(img)
    fft = np.fft.fftshift(fft)
    m = np.log(np.abs(fft))
    p = np.angle(fft)
    return m, p

if __name__ == '__main__':
    os.mkdir('m')
    os.mkdir('p')
    os.mkdir('frame')
    os.system('ffmpeg -i secret.mp4 frame/%03d.png')

    files = os.listdir('frame')
    r_file = open('r', 'wb')

    for file in files:
        img = cv2.imread(f'frame/{file}', cv2.IMREAD_GRAYSCALE)

        m, p = fft(img)
        r_file.write(struct.pack('!ff', m.min(), m.max()))

        new_img1 = mapping(m)
        new_img2 = mapping(p)

        cv2.imwrite(f'm/{file}', new_img1)
        cv2.imwrite(f'p/{file}', new_img2)

    r_file.close()
    os.system('ffmpeg -i m/%03d.png -r 25 -vcodec png 1.mkv')
    os.system('ffmpeg -i p/%03d.png -r 25 -vcodec png 2.mkv')
```

好了，逆向手老本行，勤勤恳恳写逆算法.jpg（一些注释写代码里了）

```
import os
import cv2
import struct
import numpy as np

#事先创建m,p,frame三个文件夹，先按25帧生成这两个文件夹的图片
os.system('ffmpeg -i 1.mkv -r 25 m/%03d.png')
os.system('ffmpeg -i 2.mkv -r 25 p/%03d.png')

#用幅度谱和相位谱 双谱重构得到原图
def ifft(m,p,imgsize):
    s1=np.exp(m)
    s1_angle=p
    s1_real=s1*np.cos(s1_angle)
    s1_imag=s1*np.sin(s1_angle)
    s2=np.zeros(imgsize,dtype=complex)
    s2.real=np.array(s1_real)
    s2.imag=np.array(s1_imag)
    f2shift=np.fft.ifftshift(s2)
    img_back=np.fft.ifft2(f2shift)
    img_back=np.abs(img_back)
    return img_back

#mapping的逆映射
#源码中有保存原每一个m数组的最大最小值，unpack以后是元组，直接传入就可
#p的还原是根据相位角的范围为（0,2*pi）来定data_min、data_max
def rev_mapping(new_data,t=(),down=0,up=255):
    new_data=new_data.astype(np.float64)
    if t==():
        data_max=2*np.pi
        data_min=0
    else:
        data_max=t[1]
        data_min=t[0]
    interval=data_max-data_min
    new_interval=up-down
    data=(new_data-down)*interval/new_interval+data_min
    return data

file1=os.listdir('m')
file2=os.listdir('p')
unpack=[]
with open('r','rb') as f:
    for i in range(200):
        s=f.read(8)
        unpack.append(struct.unpack('!ff',s)) #unpack包装的r

for i in range(200):
    img1=cv2.imread(f'm/{file1[i]}', cv2.IMREAD_GRAYSCALE)
    img2=cv2.imread(f'p/{file2[i]}', cv2.IMREAD_GRAYSCALE)
    m=rev_mapping(img1,unpack[i])
    p=rev_mapping(img2)
    new_img=ifft(m,p,img1.shape)
    cv2.imwrite(f'frame/{file1[i]}',new_img)

os.system('ffmpeg -i frame/%03d.png -r 25 secret.mp4')
```

然后能得到一个secret.mp4，得到了压缩包密码。

rar密码为zs6hmdlq5ohav5ll

challenge 2

查看hint.txt可以看到：

```
<math><mrow><mo>{</mo><mtable><mtr><mtd><mi>A</mi><mi>cos</mi><mo></mo><mo>(</mo><mi>m</mi><mi>x</mi><mo>+</mo><mi>n</mi><mo>)</mo></mtd></mtr><mtr><mtd><mi>B</mi><mi>cos<
```

查了一下是Mathematical Markup Language (MathML)，在菜鸟教程在线编辑器跑一下可以看到：

然后写上一关的逆算法的时候看到过两幅图像幅度谱和相位谱替换_陨星落云的博客-CSDN博客，联想到这个式子有点幅度谱和相位谱交换的味道，所以直接把文章里的脚本拿来用，秒解：

```
import numpy as np
import cv2
from matplotlib import pyplot as plt

def magnitude_phase_split(img):
    # 分离幅度谱与相位谱
    dft = np.fft.fft2(img)
    dft_shift = np.fft.fftshift(dft)
    # 幅度谱
    magnitude_spectrum = np.abs(dft_shift)
    # 相位谱
    phase_spectrum = np.angle(dft_shift)
    return magnitude_spectrum,phase_spectrum

def magnitude_phase_combine(img_m,img_p):
    # 幅度谱与相位谱结合
    img_mandp = img_m*np.e**(1j*img_p)
    img_mandp = np.uint8(np.abs(np.fft.ifft2(img_mandp)))
    img_mandp =img_mandp/np.max(img_mandp)*255
    return img_mandp

# 读取图像
img1 = cv2.imread("mix1.png",0)
img2= cv2.imread("mix2.png",0)

# 分离幅度谱与相位谱
img1_m,img1_p = magnitude_phase_split(img1)
img2_m,img2_p = magnitude_phase_split(img2)

# 合并幅度谱与相位谱
# 将苹果图像的幅度谱与橘子图像的相位谱结合
img_1mAnd2p = magnitude_phase_combine(img1_m,img2_p)
# 将橘子图像的幅度谱与苹果图像的相位谱结合
img_2mAnd1p = magnitude_phase_combine(img2_m,img1_p)

plt.figure(figsize=(10,8))
plt.subplot(221)
plt.xlabel("1")
plt.imshow(img1,cmap="gray")
plt.subplot(222)
plt.imshow(img2,cmap="gray")
plt.xlabel("2")
plt.subplot(223)
plt.imshow(img_1mAnd2p,cmap="gray")
plt.xlabel("1m+2p")
plt.subplot(224)
plt.imshow(img_2mAnd1p,cmap="gray")
plt.xlabel("2m+1p")
plt.show()
```

就能看到

扫一下左下二维码能得到文本：`0f88b8529ab6c0dd2b5ceefaa1c5151aa207da114831b371ddcafc74cf8701c1d3318468d50e4b1725179d1bc04b251f`

final challenge

被上个challenge的那一串文本误导了，一直以为是用来解密图片的密钥啥的，结果白白卡了三个多小时（菜鸡落泪

中间查了n多资料就不细说了（

总之最后拿[相位掩膜+傅立叶变换进行图像加密_isyiming的博客-CSDN博客](#)的脚本改了一下：

```
I=imread('phase.png');% 载入图像
A=im2double(I);% 将图像转为double格式
figure,imshow(A);title('The original image');% 显示图像

%加密部分
Y=fftshift(A);% 傅立叶变换部分调整整幅图像，将零频点移到频谱的中间
figure,imshow(Y);title('shifted image');%显示
B=fft2(Y);% 二维傅立叶变换
figure,imshow(B);title('FFT imagea');%显示

%M1=rand(255/255);% 随机生成密钥
M1=0.814%这里你自己更改你想要的值，只要是0～1范围内就好

M11=exp(i*2*pi*M1);%M11为根据随机相位生成的图像掩膜
M111=B.*M11; %将要加密的图像和掩膜相乘
figure,imshow(M111);title('phase mask');%显示加密图像

D=fft2(M111); % 再次傅立叶变换
figure,imshow(D);title('FFT image b');

C=abs(D);%对经过两次傅立叶变化的图像像素灰度取绝对值

%解密部分
C1=ifft2(C); % 二维傅立叶逆变换
figure,imshow(C1);title('2-D IFFT b');%显示进行一次傅立叶逆变换的图像
C11=C1.*exp(-i*2*pi*M1); % 移除掩膜，这个M1就是信息发送方和接收方事先约定好的密钥，接收加密图像的人必须知道M1才能正确解密。
%这个程序只是演示加密解密过程，就随机生成的M1，不然应该是有一个密钥文件记录M1，双方保留。
figure,imshow(C11);title('remove mask');%显示移除掩膜后的图像
C111=ifft2(C11); %二维傅立叶逆变换
figure,imshow(C111);title('2-D IFFT a');%显示去除掩膜和进行两次傅立叶变换的图像
C1111=ifftshift(C111); %将零频点还原到原始位置
F=abs(C1111); %取绝对值
figure,imshow(F);title('The decrypted image');%显示，得到了最终解密后到图像
```

就能看到其中一张图（phase mask）是：

好像是没分离好（等其他大佬的wp），不过能大概辨认出key是`a8bms0v4qer3wgd67ofjhyxku5pi1cz1`

```
from Crypto.Cipher import AES
import binascii

key=b"a8bms0v4qer3wgd67ofjhyxku5pi1cz1"
aes=AES.new(key,AES.MODE_ECB)

cipher=binascii.unhexlify("0f88b8529ab6c0dd2b5ceefaa1c5151aa207da114831b371ddcafc74cf8701c1d3318468d50e4b1725179d1bc04b251f")
text=aes.decrypt(cipher)
print(text)
```

Flag

```
flag{1ba48c8b-4eca-46aa-8216-d164538af310}
```

- **本文作者：** CTFHub
- **本文链接：** https://writeup.ctfhub.com/Challenge/2021/第四届红帽杯网络安全大赛/Misc/xE1eE23b779te66e8f3Me.html
- **版权声明：** 本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！

#Challenge #2021 #Misc #第四届红帽杯网络安全大赛
hpcurve
ezlight