




python3实现简单图片隐写

原创

守护者  于 2019-05-04 21:52:05 发布  3157  收藏 28

分类专栏: [杂七杂八](#)

守护者安全 (www.zhaosimeng.cn)

本文链接: https://blog.csdn.net/weixin_42721957/article/details/89818431

版权



[杂七杂八](#) 专栏收录该内容

5 篇文章 0 订阅

订阅专栏

LSB算法

LSB(英文 least significant bit)即最低有效位。LSB加密是信息隐藏中最基本的方法。由于人们识别声音或图片的能力有限, 因此我们稍微改动信息的某一位是不会影响我们识别声音或图片的。一般来说, MSB(最高有效位)位于二进制数的最左侧, LSB位于二进制数的最右侧。

由于图像的每一个像素点都是由RGB(红、绿、蓝)三原色组成, 而这三种颜色又可以组合成各种其它颜色, 每个颜色占8位(如#FFFFFF), LSB隐写即是修改每个颜色值的最低一位, 将其替换为我们想要嵌入的信息中的内容, 以此来实现数据隐藏。

一个像素点包含三种颜色, 每个颜色修改最后1位, 这样一个像素点就可以携带3位信息。

应用LSB算法的图像格式需为位图形式, 即图像不能经过压缩, 如LSB算法多应用于png、bmp等格式, 而jpg格式较少。由于图像是由像素构成的, 每个像素最后一位的变化, 通过肉眼是无法察觉的

Python Imaging Library

简称PIL, 为Python解释器提供了图像处理的功能, 结合PIL可以方便的编写Python脚本处理图片隐写问题。

StegSolve

StegSolve是一款基于Java开发的流行图片隐写分析软件, 其支持常见的图片文件格式, 可以对不同的文件进行结合(包括XOR、ADD、SUB等操作), 可以对图片文件格式进行分析, 可以提取GIF文件中的帧等, 覆盖了基本的图片隐写分析需求。

隐写信息代码:

```
# -*- coding: UTF-8 -*-  
  
from PIL import Image  
  
# PIL, Python Imaging Library. 图像处理功能
```

```

def plus(str):
    # Python zfill() 方法返回指定长度的字符串，原字符串右对齐，前面填充0。

    return str.zfill(8)

def get_key(strr):
    # 获取要隐藏的文件内容

    tmp = strr

    f = open(tmp, "rb")          # 要读取二进制文件，比如图片、视频等等，用'rb'模式打开文件

    s = f.read()
    str = ""

    for i in range(len(s)):
        # 逐个字节将要隐藏的文件内容转换为二进制，并拼接起来
        # 1.使用bin()函数将十进制的ascii码转换为二进制

        # 2.由于bin()函数转换二进制后，二进制字符串的前面会有"0b"来表示这个字符串是二进制形式，所以用replace()替换为空

        # 3.又由于这样替换之后是七位，而正常情况下每个字符由8位二进制组成，所以使用自定义函数plus将其填充为8位

        str = str + plus(bin(s[i]).replace('0b', ''))    # s[i]为该位置字符的十进制表示的ascii码

        # print str

    f.closed

    return str

def mod(x, y):
    return x % y

# str1为载体图片路径，str2为隐写文件，str3为加密图片保存的路径

def func(str1, str2, str3):
    im = Image.open(str1)

    # 获取图片的宽和高

    width = im.size[0]

    print( "width:" + str(width) + "\n")

    height = im.size[1]

    print ("height:" + str(height) + "\n")

    count = 0

    # 获取需要隐藏的信息

    key = get_key(str2)

    keylen = len(key)

```

```

keylen = len(key)

for h in range(0, height):

    for w in range(0, width):

        pixel = im.getpixel((w, h))    # 返回该像素点三原色的二进制信息，形成一个数组

        a = pixel[0]    # R

        b = pixel[1]    # G

        c = pixel[2]    # B

        if count == keylen:
            break

        # 下面的操作是将信息隐藏进去

        # 分别将每个像素点的RGB值余2，这样可以去掉最低位的值

        # 再从需要隐藏的信息中取出一位，转换为整型

        # 两值相加，就把信息隐藏起来了

        a = a - mod(a, 2) + int(key[count])

        count += 1

        if count == keylen:
            im.putpixel((w, h), (a, b, c))

            break

        b = b - mod(b, 2) + int(key[count])

        count += 1

        if count == keylen:
            im.putpixel((w, h), (a, b, c))

            break

        c = c - mod(c, 2) + int(key[count])

        count += 1

        if count == keylen:
            im.putpixel((w, h), (a, b, c))

            break

        if count % 3 == 0:
            im.putpixel((w, h), (a, b, c))

    im.save(str3)

```

原图

```

old = r"E:\Python\python programs\splider\coffee.png"

# 处理后输出的图片路径

new = r"E:\Python\python programs\splider\yinxie.png"

# 需要隐藏的信息

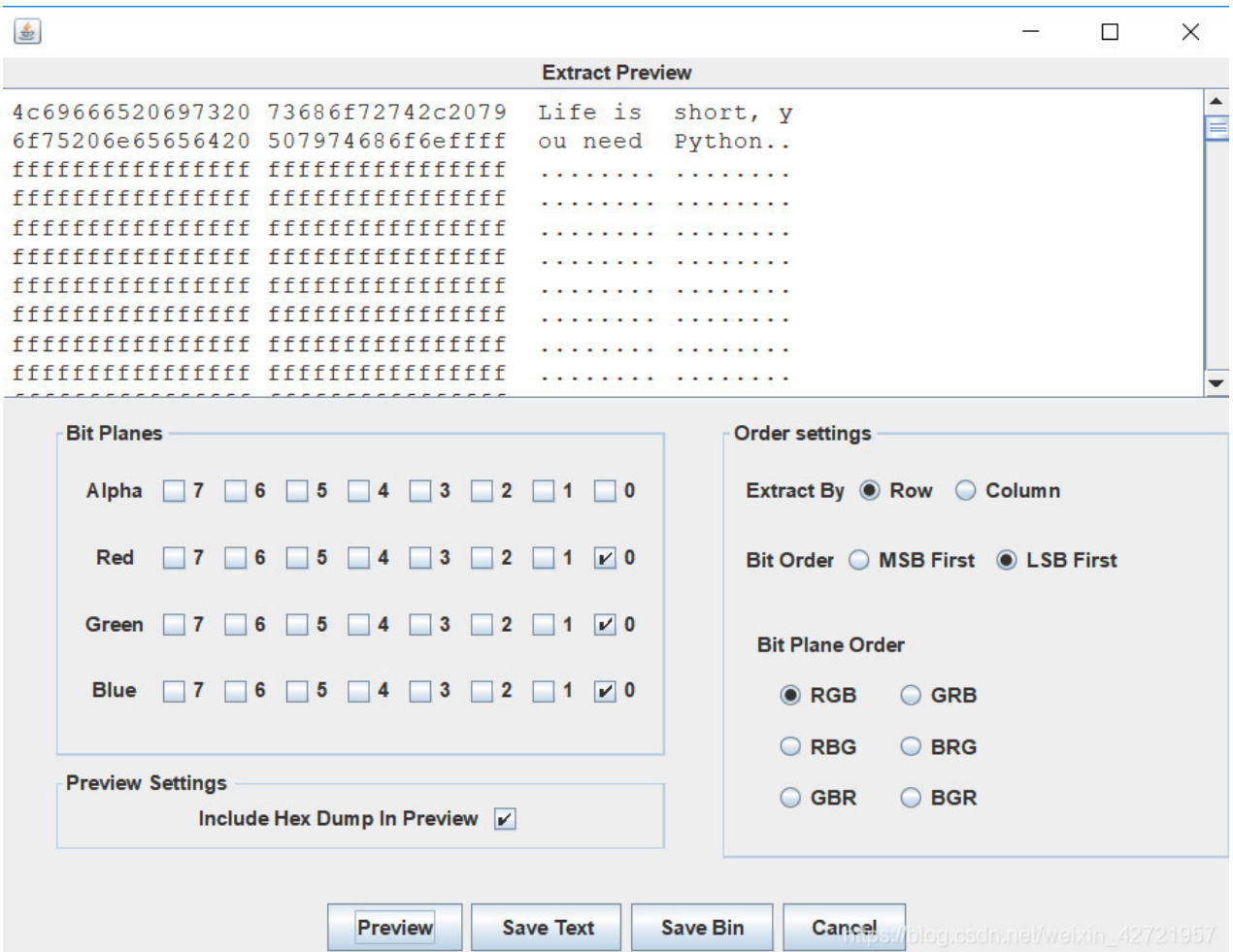
enc = r"E:\Python\python programs\splider\flag.txt"

func(old,enc,new)

```

效果如下:

on programs > splider



解密隐写图片代码:

```
# -*- coding:UTF-8 -*-

from PIL import Image

def mod(x, y):
    return x % y;

# le为所要提取的信息的长度，str1为加密载体图片的路径，str2为提取文件的保存路径

def func(le,str1,str2):

    b = ""

    im = Image.open(str1)

    lenth = le * 8

    width = im.size[0]

    height = im.size[1]

    count = 0

    for h in range(0, height):

        for w in range(0, width):

            # 获得(w,h)点像素的值

            pixel = im.getpixel((w, h))

            # 此处余3，依次从R、G、B三个颜色通道获得最低位的隐藏信息

            if count % 3 == 0:

                count += 1

                b = b + str((mod(pixel[0], 2)))

            if count == lenth:

                break

            if count % 3 == 1:

                count += 1

                b = b + str((mod(pixel[1], 2)))

            if count == lenth:

                break

            if count % 3 == 2:

                count += 1

                b = b + str((mod(pixel[2], 2)))
```

```

        if count == lenth:
            break

    if count == lenth:
        break

with open(str2, "w",encoding='utf-8') as f:

    for i in range(0, len(b), 8):
        # 以每8位为一组二进制，转换为十进制

        stra = int(b[i:i + 8],2)

        # 将转换后的十进制数视为ascii码，再转换为字符串写入到文件中

        f.write(chr(stra))

        stra = ""

    f.closed

# 文件长度

le = 30

# 含有隐藏信息的图片

new = "E:\Python\python programs\splider\yinxie.png"

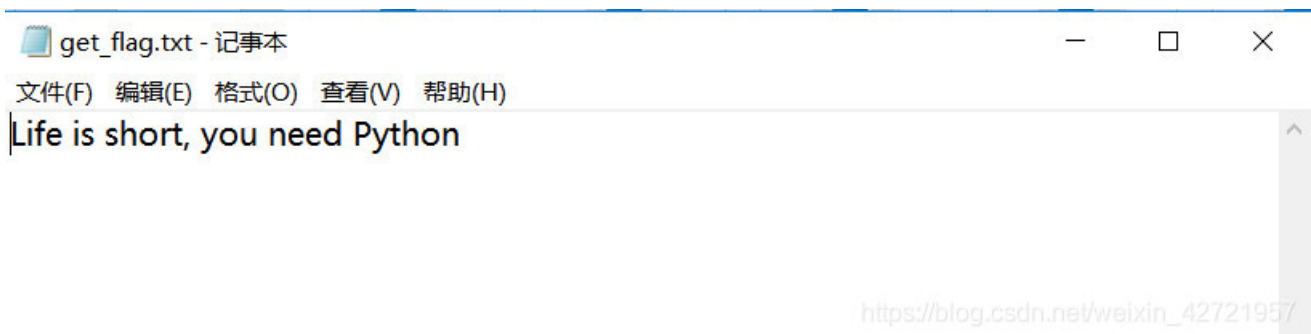
# 信息提取出后所存放的文件

tiqu = "E:\Python\python programs\splider\get_flag.txt"

func(le,new,tiqu)

```

效果如下：



过程中遇到了PIL无法安装的问题，参考链接：https://blog.csdn.net/qq_38906523/article/details/79723969

最后就是更新pip然后安装成功的。