实验一报告

实验目的

- 对数字水印技术有所了解
- 理解LSB的基本原理
- 能将隐藏在图片中的文本信息通过LSB算法解码出来
- 能在LSB算法的基础上编码实现将自己的照片（例如证件照等）隐藏到所提供的校门照片中

设计说明

实验环境

- python3
  - Pillow：图像处理
  - PyCryptodome：加密算法

文件结构

`steganography.py`：隐写术基类

`lsb.py`：LSB隐写类

`degrade.py`：图像退化隐写类

`crypt.py`：加解密算法

`main.py`：主函数

`./src`：资源文件

运行方式

```
python main.py [option] ...
参数：
-M 使用LSB隐写
-H 隐写数据
-E 提取隐写数据
-C 载体图像路径
-t 待隐写文件路径
-o 输出文件路径
-K 加密隐写数据或提取加密数据
-pub 公钥文件
-pri 私钥文件
```

设计亮点

- LSB算法实现加密隐写信息和数字签名认证
  - 128位AES加密隐写信息
  - RSA非对称加密传输AES密钥
  - 数字签名认证
- LSB算法实现基于伪随机数的随机替换位置
- 实现图像退化隐写

实验过程

使用Pillow库中的Image类来处理图像

主要用于获取图像的信息包括：

- 图像大小
- 图像颜色格式（RGB或RGBA）
- 图像像素信息
- 生成隐写信息的图像

LSB隐写

隐写信息

函数`hide()`用与隐写信息

首先打开载体图片和待隐写文件，根据载体图像格式判断是否足够容纳嵌入信息

根据命令行参数判断是否进行加密隐写

- 直接隐写
  1. 直接隐写主要由子函数`set_data()`完成，该函数遍历载体图像的像素点，调用`set_lsb()`函数将信息比特数据依次嵌入载体每个像素点的RGB值的最低有效位
  2. 直到传入的比特数据全部被嵌入后保存隐写信息的图像

- 加密隐写
  1. 获取待隐写信息的byte数据后，调用`generate_encrypt_data()`生成加密数据和签名；该函数首先生成一个随机的AES密钥，然后利用AES密钥对隐写信息进行加密；AES密钥通过RSA加密并放在加密数据前；最后计算加密数据和加密后的AES密钥的hash形成签名并用发送方私钥加密；函数返回加密后的信息和加密信息的长度

     > 非对称加密的加密速度要比对称加密慢得多，因此利用对称加密来加密数据，并使用非对称加密来加密对称加密密钥的方式来提高效率
     >
     > 最后返回的加密信息长度能提高后续提取信息的效率，避免将图像的最低有效位全部提取出来；同时也作为后续生成随机替换位置的伪随机数种子

  2. 生成加密信息后先在前6个byte中设置加密信息的长度（伪随机数种子），在RGB模式下是前16个像素点，RGBA模式下是前12个像素点；然后调用`set_encrypt_data()`来嵌入加密信息

3. `set_encrypt_data()`函数中先调用`get_random_map()`来使用长度作为伪随机数种子生成随机替换位置，这里生成的随机替换位置不能包括前6byte的长度块，避免伪随机数种子被改变；然后遍历生成的替换位置来将加密信息嵌入到最低有效位
4. 最后保存隐写信息的图像

提取信息

函数`extract()`用与提取隐写信息

首先打开载体图片，然后根据命令行参数判断使用进行加密信息的提取

- 直接提取
  1. 直接提取信息由函数`get_data()`实现，该函数从第一像素点开始提取每个像素点的最低有效位，直到提取到参数指定长度的数据
  2. 直接提取就是将所有像素点的最低有效位取出即可

- 提取加密信息
  1. 首先要调用`get_data()`函数来取出前6byte数据，也就是加密数据的长度
  2. 然后调用`get_encrypt_data()`函数来提取加密信息，这个函数首先要利用上一步取出的长度作为伪随机数种子生成随机替换位置，然后再遍历替换位置提取出最低有效位信息
  3. 提取出加密信息后调用`decrypt_data()`函数对加密信息进行数字签名验证和解密，该函数首先取出加密信息的前128byte，也就是数字签名，然后利用发送方的公钥解密获取消息的hash值，然后计算加密信息的hash比较验证签名是否正确；验证签名成功之后，再使用接收方的私钥解密AES密钥（在签名之后的128byte）；最后利用AES密钥解密加密的信息

  提取出信息后根据命令行参数判断直接按照明文输出还是写入输出文件

图像退化隐写

隐写信息

同样是`hide()`函数进行信息隐写

1. 首先打开载体图像和待隐写图像，然后判断载体图像是否能够容纳待隐写图像，因为退化隐写要求载体图像和待隐写图像具有相同的维度，因此需要比较两个图像的维度是否相同
2. 同时遍历载体图像和待隐写图像的像素点，先将载体图像RGB低4位清零，然后再提取待隐写图像RGB的高4位，并写入载体图像RGB的低4位
3. 最后输出保存隐写图像

提取信息

提取信息是隐写信息的逆过程，由`extract()`函数实现

1. 首先打开载体图像，然后遍历载体图像的所有像素点，依次将像素的RGB值低4位左移到高4位
2. 最后保存输出图像

实验结果

1. 提取隐藏文字

`python .\main.py -E -M -T -c .\src\Figure2-encode.png`

```
[+]Steganography in LSB mode
[+]Extract data...
[+]Output plain text:
Nought may endure but Mutability.
[+]Extract complete
```

2. LSB隐写和解码图像

`python .\main.py -H -c .\src\Figure3-origin.png -t .\src\head.jpg -o .\src\Figure3-encode.png -K --pub .\public_recv.pem --pri .\private_send.pem`

```
PS D:\Pycharm projects\test\LSB> python .\main.py -H -c .\src\Figure3-origin.png -t .\src\head.jpg -o .\src\Figure3-encode.png -K --pub .\public_recv.pem --pri .\private_send.pem
[+]Steganography in LSB mode
[+]Hide data...
[+]Carrier capacity 4572288 byte
[+]Target capacity 1177726 byte
[+]Message encryption...
[+]Encrypt complete, payload size: 1178000 byte
[+]Generate random bit map...
[+]Set encrypt data...
[+]Hide completion
```

`python .\main.py -E -c .\src\Figure3-encode.png -o .\src\Figure3-decode.png -K --pub .\public_send.pem --pri .\private_recv.pem -M`

```
PS D:\Pycharm projects\test\LSB> python .\main.py -E -c .\src\Figure3-encode.png -o .\src\Figure3-decode.png -K --pub .\public_send.pem --pri .\private_recv.pem -M
[+]Steganography in LSB mode
[+]Extract data...
[+]Message decryption...
[+]Payload size: 1178000 byte
[+]Generate random bit map...
[+]Get encrypt data...
[+]Decrypt data...
[+]verify successfully!
[+]Extract complete
```

3. 图像退化隐写和解码

`python .\main.py -H -c .\src\Figure4-origin.png -t .\src\Figure5-origin.png -o .\src\Figure4-encode-degrade.png`

```
PS D:\Pycharm projects\test\LSB> python .\main.py -H -c .\src\Figure4-origin.png -t .\src\Figure5-origin.png -o .\src\Figure4-encode-degrade.png
[+]Steganography in Degrade mode
[+]Hide data...
[+]Hide completion
```

`python .\main.py -E -c .\src\Figure4-encode-degrade.png -o .\src\Figure4-decode-degrade.png`

```
PS D:\Pycharm projects\test\LSB> python .\main.py -E -c .\src\Figure4-encode-degrade.png -o .\src\Figure4-decode-degrade.png
[+]Steganography in Degrade mode
[+]Extract data...
[+]Extract complete
```

参考资料

- PyCryptodome — PyCryptodome 3.19.0 documentation
- cloacked-pixel/lsb.py at master · livz/cloacked-pixel (github.com)