**《信息安全概论》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **姓名** | | 谭星 | | **年级** | | 2019级 |
| **学号** | | 20191584 | | **专业、班级** | | 计算机科学与技术卓越01班 |
| **实验名称** | **信息隐藏实验** | | | | | |
| **实验时间** | **2022.4.19** | | **实验地点** | | **DS3402** | |
| **实验成绩** |  | | **实验性质** | | **□验证性 □设计性 □综合性** | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  评语：  评价教师签名（电子签名）： | | | | | | |
| 一、实验目的   1. 学习并掌握图像信息隐藏的基本原理和方法 2. 实现基于LSB的信息隐藏和提取算法 | | | | | | |
| 二、实验项目内容   1. 使用LSB算法在图片中隐藏如下信息：CQUWATERMASKEXP 2. 从被隐藏数据的图片中解析出如上信息，建议使用Matlab 3. 在实际应用中，隐藏信息量通常是不可预知的，同时，攻击者也很容易从最低位像素提取到隐藏信息。另一方面，如何确保信息来源于正确的发送者？针对这些问题，请设计完整的保密通信方案。 | | | | | | |
| 三、实验设计  （1）LSB算法嵌入加密信息  A、实验原理  在图像中，像素的高位信息对图像感官质量其主要作用，更改图像像素最低位的信息，对图像感官质量的影响很低甚至没有影响。比如像素有124变为125，感官并不会感知到图像的变化。所以可以利用这一特性，用秘密信息替代图像像素低位的信息，从而实现信息嵌入。  B、模块设计  读入图像预处理模块：  将入读的图像的最低位置位为0；  秘密信息预处理模块：  将待嵌入的秘密信息转换为二进制形式；  嵌入信息：  遍历图像，依次将二进制秘密信息的一位存入像素的最低位处。  C、流程图  整体处理过程，如下图所示：    （2）LSB算法从加密图像中提取秘密信息  A、实验原理  按照LSB加密的原理，可以根据加密二进制信息的长度从携带秘密信息的图像的像素中提取对应长度的二进制秘密信息，进而解密秘密信息。  B、模块设计  提取二进制秘密信息模块：  读入图像，根据隐藏信息量，提取像素的最低1bit，组合成连续的二进制秘密信息。  将二进制秘密信息：  将二进制秘密信息转换为ASCII码。    C、流程图  整体处理流程，如下图所示：    （3）方案设计：  A、待解决的问题：  在实际应用中，上述的LSB算法存在一些关键问题，如下  a、隐藏信息量通常是不可预知的  b、攻击者也很容易从最低位像素提取到隐藏信息  c、不能确保信息来源于正确的发送者  本实验中提出响应的解决思路如下：  a、在图像固定长度的位置存储秘密信息量  b、将秘密信息加密后，再嵌入图像，这样就算被提取加密信息，也无法得到正确的信息。  c、可利用数字签名技术，再图像中加入数字签名验证信息是否来源于正确的发送者。 | | | | | | |
| 四、实验过程或算法  （1）LSB算法嵌入加密信息  A、关键步骤  a、将待隐藏信息转换为二进制。    b、遍历图像，对像素的最低1bit置0。    c、遍历图像，在像素最低比特位写入二进制表示隐藏的信息。    B、代码   |  | | --- | | from PIL import Image  import numpy as np  I = Image.open('swimRing.png')  img = np.array(I)  def str\_2\_bin(str):      """      字符串转换为二进制      """      return ''.join([bin(ord(c)).replace('0b', '') for c in str])  def bin\_2\_str(bin):      """      二进制转换为字符串      """      return ''.join([chr(i) for i in [int(b, 2) for b in bin.split(' ')]])  for i in range(img.shape[0]):      for j in range(img.shape[1]):          for w in range(img.shape[2]):              img[i,j,w] = img[i,j,w]&254  imgMask = str\_2\_bin("CQUWATERMASKEXP")  length = len(imgMask)  index = 0  for i in range(img.shape[0]):      for j in range(img.shape[1]):          for w in range(img.shape[2]):              if index < length:                  if imgMask[index]=='1':                      img[i,j,w] = img[i,j,w]|1                  index += 1  im = Image.fromarray(img)  im.save("swimRing1.png") |   （2）LSB算法从加密图像中提取加密信息  A、关键步骤  a、将提取出的二进制转换为ASCII字符    B、代码   |  | | --- | | from PIL import Image  import numpy as np  I = Image.open('swimRing1.png')  img = np.array(I)  def bin\_2\_str(bin):      """      二进制转换为字符串      """      return ''.join([chr(i) for i in [int(b, 2) for b in bin.split(' ')]])  length = len("CQUWATERMASKEXP")\*7  index = 0  res = ""  for i in range(img.shape[0]):      for j in range(img.shape[1]):          for w in range(img.shape[2]):              if index < length:                  if img[i,j,w]%2==1:                      if (index)%7!=0:                          res+='1'                      else:                          res+=' 1'                  else:                      if (index)%7!=0:                          res+='0'                      else:                          res+=' 0'                  index += 1  res = res[1:]  print("解密后：")  print(bin\_2\_str(res)) |   （3）方案设计  A、发送方方案  a、将待加密信息M通过Des加密成为密文C，并统计密文的信息量len（C）。  b、将密文C通过hash函数得到MD，并结合密钥K得到MAC，并统计MAC的信息量len（MAC）。  c、在图像的前0~31位像素最低位插入密文的信息量len（C），在32~31+len（C）的像素最低位插入密文C；  在图像的32+len（C）~64+len（C）的像素最低位位插入MAC的信息量len(MAC)；在图像的65+len(C)~65+len（C）+len（MAC）像素最低位插入MAC。完成加密信息的嵌入。  整体的处理流程如下图所示：    B、接收方接收方案  a、遍历接收的携带密文的图像，在图像的0~31位像素的最低位读取1bit组成连续的二进制信息，得到密文C的信息量len（C）。  在图像的32~31+len（C）位像素最低位读取1bit组成连续的二进制信息，得到密文C’。  在图像的32+len（C）~64+len（C）位像素最低位读取1bit组成连续的二进制信息，得到MAC的信息量len（MAC）。  在图像的65+len(C)~65+len（C）+len（MAC）位像素最低位读取1bit组成连续的二进制信息，得到MAC’。  b、将提取的加密密文C’通过hash函数得到MD’，并结合密钥K得到MAC’’。  c、比较MAC’ 和 MAC’’ 是否一致，一致则说明信息来源于正确的发送者。  d、MAC’ 和 MAC’’一致，则将提取到的加密密文通过Des解密得到明文信息M。  整体的接受流程方案如下图所示： | | | | | | |
| 五、实验过程中遇到的问题及解决情况  （1）设计方案  A、难点  在加密的过程中为什么不可以使用非对称加密。  B、答案  非对称算法的效率较低，而待嵌入的信息量一般十分巨大，所以不适合使用非对称加密的方式加密密文。 | | | | | | |
| 六、实验结果及分析和（或）源程序调试过程  （1）LSB算法嵌入信息  A、结果  控制台输出：  原图像和携带信息的图像：    图1 原图像 图2 携带信息图像  （2）LSB算法从携带信息图像中提取信息  A、结果  控制台输出    可以看到提取出的明文信息为CQUWATERMASKEXP，正是嵌入的信息，所以算法正确。 | | | | | | |