数字图像处理技术在图像加密的应用

摘要：
随着网络技术和多媒体技术的飞速发展，多媒体安全问题日益突显出来。图像加密技术是多媒体信息安全的核心技术，由于图像数据具有编码结构特殊、数据量大、实时性要求高等特点，而传统的数据加密算法直接用于图像数据加密，很难满足实时性要求，且会改变数据格式，这就要求对图像数据采用特殊的加密算法。

关键字：图像加密；计算机安全

数字图像处理技术在图像加密的应用

1. 图片加密算法的实现

1.1. 随机打乱各行或列进行数字图像加密

对于这种加密算法，在加密后需要立即解密，否则无法解密成功，因此，在实际的应用中并无什么作用。

下为代码：

```
%随机打乱各行进行数字图像加密
clear
RGB = imread('test.jpg');
s = size(RGB);
%不放回的均匀分布的从 1 到 s(1)到整数，个数为 s(1)抽样
r = randsample(s(1), s(1));
RGBS = RGB(r, :, :);
j = 1; f = 1:length(r);
while j <= length(r)
    f(j) = find(r == j);
    j = j + 1;
end
RGBE = RGBS(f, :, :);
subplot(1, 2, 1); imshow(RGBS); title('加密后');
subplot(1, 2, 2); imshow(RGBE); title('解密后');
```

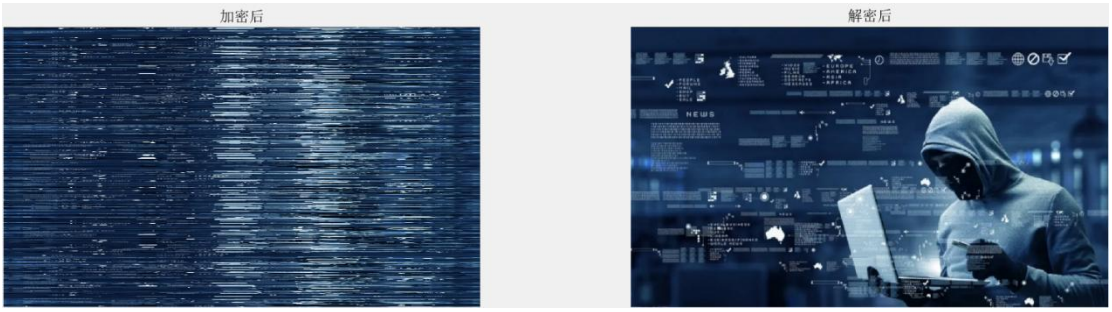

图 1.1.1 随机打乱各行或列进行加密

1.2. 利用混沌序列进行加密解密

```
%% 混沌序列数字图像加密程序，输入的加密密钥在 0～1 范围的数
clear;
RGB = imread('test.jpg', 'jpg');
s = size(RGB);
N = s(1) * s(2) * s(3);
%m = 1:N;
m(1) = input('输入加密密钥：  ');
disp('加密中...');
for i = 1:N-1
    m(i+1) = 4 * m(i) - 4 * m(i)^2;
```

```
end
m = mod(m * 1000, 256);
m = uint8(m);
n = 1;
%RGBS = zeros(s(1), s(2), s(3));
for i = 1:s(3)
    for j = 1:s(2)
        for k = 1:s(1)
            RGBS(k, j, i) = bitxor(m(n), RGB(k ,j, i));
            n = n + 1;
        end
    end
end
disp('加密成功!');
imwrite(RGBS, 'test1.jpg','jpg');
subplot(1,2 ,1);imshow(RGBS);title('加密后的图片');

%% 混沌加密的解密程序，输入的解密密钥即是加密密钥
%clear;
RGBS1 = imread('test1.jpg', 'jpg');
s1 = size(RGBS1);
N1 = s1(1) * s1(2) * s1(3);
%m = 1:N;
m1(1) = input('输入解密密钥:  ');
disp('解密中...');
for i1 = 1:N1-1
    m1(i1+1) = 4 * m1(i1) - 4 * m1(i1)^2;
end
%RGBE = zeros(s(1), s(2), s(3));
m1 = mod(m1 * 1000, 256);
m1 = uint8(m1);
n1 = 1;
for i1 = 1:s1(3)
    for j1 = 1:s1(2)
        for k1 = 1:s1(1)
            %这里不能使用 RGBS1，由于 jpg 是有损压缩格式，读入和读出存在误差；
            %故使用加密后没写入 jpg 格式时的 RGB;
            %RGBE(k1, j1, i1) = bitxor(m1(n1), RGBS1(k1, j1, i1));
            RGBE(k1, j1, i1) = bitxor(m1(n1), RGBS(k1, j1, i1));
            n1 = n1 + 1;
        end
    end
end
disp('解密成功!');
```

```
imwrite(RGBE, 'test2.jpg', 'jpg');
subplot(1, 2, 2); imshow(RGBE); title('解密后的图片');
```

对于我们而言，组成图最基本的要素是像素值的大小和像素值的位置。图片中的像素值或者位置的改变都可能引发我们对图片的理解出现偏差。

如果有这么一个”图片破坏者“，其工作就可以是改变图像像素的位置或者改变图片像素值。他通过成程序随机数利用随机数改变像素值或者像素位置那么轻而易举就可以使得这个图片面目全非。但是这个”图片破化者“只能破坏图片却不能还原图片。因为任何程序都没有机会再次生成同样的随机数。

但是对于混沌系统而言，混沌系统也具有十分优越的伪随机性，这样的随机性能够有效的对这张图片进行破坏，使得通过非法手段获取图片的无法理解这张图。而这样的”随机性“对于图片的加密者是完全可再恢复的。同时出色的初值敏感性能防止绝大数的暴力破解，遍历性和随机性能有效的预防差分攻击和统计学攻击。

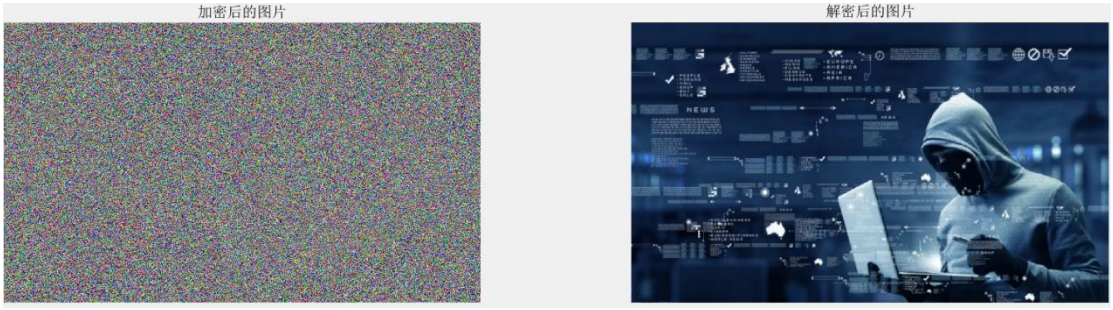

图 1.2.1 利用混沌序列进行加密

我们对于上述理论进行验证，验证图片加密像素值的完全改变。

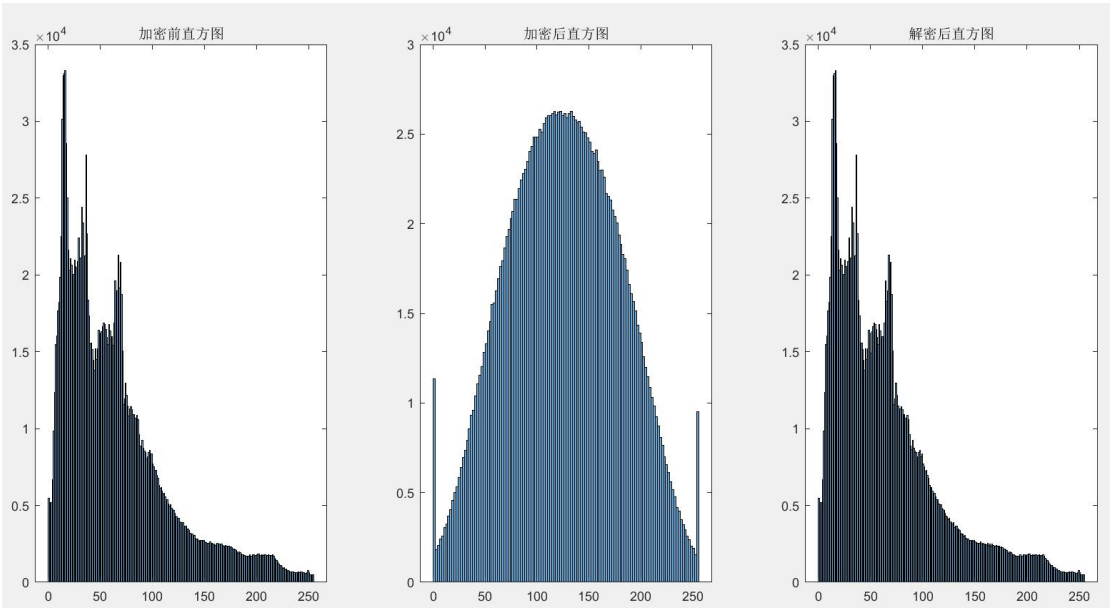

图 1.2.2 像素直方图验证

由上图验证可得，混沌加密图像的直方图的像素点分布相对均匀，很好地隐藏了原始图像的统计特性，达到了图像加密的效果。并且，在加密前后，图片像素完全没有发生改变。对于非立即解密的方法中，由于密钥随机生成，电脑生成的随机数不可能出现相同的可能，而且无法通过加密后的图像推导出原图像，因此，在不泄露密钥的情况下，该算法保持了非常良好的加密效应，也广泛使用于众多方面。

2. 图像加密在实际生活中的应用

2.1. 数字图像置乱及安全性

数字图像置乱可分为空间域置乱与频域置乱两种，主要是指通过适当修改置换数字图像空间域或者数字图像变换域参数，以杂乱图像的方式作为传输数字图像的表象，实现数字图像资料的安全传输过程。结合当前应用研究情况来看，研究人员对数字图像的变换算法进行了深入研究与分析。并相继应用 FASS 曲线、生命模型以及混沌置乱等算法方式，初步实现了对数字图像信息的安全处理。最重要的是，这些算法方式可用于数字图像的预处理以及后处理工作当中，不仅可以掩盖真实的数字图像信息，同时还可以最大限度确保数字图像信息传输安全[1]。

2.2. 数字图像信息隐藏及安全性

数字图像信息隐藏基本上可以视为数字图像加密技术安全性的重要表现。作用原理上来看，数字图像信息隐藏在一定程度上与经典密码学中的信息隐藏相似。都是通过将公开的图像以载体形式嵌入到一些需要隐藏的信息当中。其中，这些公开的图像通常会表现为视觉不敏感的图像像素。

而经过保密处理的数字图像信息可以隐藏于这些视觉不敏感的图像像素当中，完成加密传输过程。一般来说，处于公开状态的图像因其本身具有迷惑性，可以促使对方放松警惕且不易被察觉，而会降低数字图像信息遭受攻击的概率。由此不难看出，数字图像信息隐藏在一定程度上可以实现对数字图像加密过程的安全管理。

对于信息隐藏来说，和如今非常热门的区块链中有广泛的应用。在区块链中 NFT 中有较为广泛的运用。

非同质化代币（英语：Non-Fungible Token，简称：NFT），是一种众筹扶持项目方的方式，也是一种被称为区块链数位账本上的数据单位，每个代币可以代表一个独特的数码资料，作为虚拟商品所有权的电子认证或凭证。由于其不能互换的特性，非同质化代币可以代表数位资产，如画作、艺术品、声音、影片、游戏中的项目或其他形式的创意作品。虽然作品本身是可以无限复制的，但这些代表它们的代币在其底层区块链上能被完整追踪，故能为买家提供所有权证明。诸如以太币、比特币等加密货币都有自己的代币标准以定义对 NFT 的使用。

卖家常常会运用加密的手段，在 NFT 上加入一些信息，公司可以投资 NFT，以此获得其中所需求的信息，这样的传递信息非常的隐蔽安全便捷，获得了众多投资者的青睐。

如下介绍最简单的加密技术，隐写术。

代码如下：

```
from PIL import Image #从 pillow 库（即 PIL）中导入 Image 类
img = Image.open('test.jpg') #读取图片存入变量 img 中
print(img.format) #输出图片格式(str)
print(img.size) #输出图片大小信息 （宽度 w，高度 h）tuple = (int,int)

#像素载入
pix = img.load()
width = img.size[0] #.size 方法返回的是一个元组 tuple =(int,int)
height = img.size[1]
#获取像素点的 RGB 值
rgb_list = [] #创建一个数组存储 RGB 值
for y in range(height):#遍历每一个像素点，将图像看作是一个二维数组，
    for x in range(width): #如果 x 循环在外层输出的图像会发生一个九十度的翻转
```

```
            r,g,b =pix[x,y] #此处的 r,g,b 是像素点 pix[x,y]的 RGB 值
            rgb_list.append(r)
            rgb_list.append(g)
            rgb_list.append(b)

#输出图像
j = 0
pixels = [] #以[(r1,g1,b1),(r2,g2,b2)]形式存放每个像素点的 RGB 值，于绘制图像
img_out = Image.new(img.mode,img.size) #生成新图像，以原图的格式和大小
#img_out 此时还是一张白纸，下面的代码旨在更新 img_out 的像素信息
while j<len(rgb_list):      #循环次数高达 786432 次
    pixels.append((rgb_list[j],rgb_list[j+1],rgb_list[j+2])) #以元组的形式
    j += 3
img_out.putdata(pixels)#放置像素信息
img_out.save("test3.jpg")#将图像保存为，程序运行后会出现在根目录

def bina_to_txt(bina):
    #只要传入一个二进制数组成的序列即可翻译成文本
    tex = []
    for i in bina:
        tex.append(chr(int(i,2)))
      return tex #返回一个单字符序列

#要求 bina 的格式为['01010101','11111111']

def txt_to_bina(txt):
    c=[]
    for a in txt:
        c.append("{:0>8}".format(bin(ord(a)).lstrip('0b')))
        #格式化将二进制码保存起来
        #注意要在右侧补齐八位，否则信息会错位
    resultlist = []
    for i in c:
        for j in range(8):
            resultlist.append(i[j])
    return resultlist

txt = "Hello World!"

#替换信息位的信息
i = 0
while i < len(txt_to_bina(txt)):
    temp =list(bin(rgb_list[i])) #用  bin()强制转换，bin()返回一个字符串类型
    temp[-1]=txt_to_bina(txt)[i] #将二进制型的 RGB 信息的最后一位转换成文本二进制码
```

```
    rgb_list[i] = int(''.join(temp),2)
      i += 1


s = 'abcde'
temp = list(s)
temp[-1] = 'f'
s = ''.join(temp)

#这里直接用的是“hello world！”的长度，后期优化可以加个旗帜识别
c = ''
for i in range(96):
    c += bin(rgb_list[i])[-1]
out_list_bin =[]
for i in range(12):
    out_list_bin.append(c[i*8:(i+1)*8])
print(''.join(bina_to_txt(out_list_bin)))
```

图片 2.2.1 未加密的图片

图片 2.2.2 已加密的图片

在上述看似相同的图片中，后者实则隐藏了”Hello World!”的文字在其中。上述代码即为对左图的加密以及右图的解密。

```
st2.py
JPEG
(1066, 623)
图片中隐藏的文字:Hello World!
```

图片 2.2.3 代码运行结果

2.3. 数字图像水印技术及安全性

所谓的数字图像水印技术就是常说的数字水印，通过按照某种方式将著作权人身份信息植入到出版物当中，并完成数字图像加密传输过程的一种技术内容。一般来说，数字图像水印技术可根据作用机理的不同细化分为鲁棒性认证水印与脆弱性认证水印两种技术类型。结合当前应用反馈情况来看，数字图像水印技术在图书以及期刊论文发表中得到了良好的应用实践效果[2]。

不仅可以有效解决版权纠纷问题，同时还可以借助数字图像水印技术对出版物中的关键信息进行提取分析，保障著作权人权利。举例而言，数字图像水印技术在提取完出版物关键信息之后，可显示之前所植入的数字水印信息。这样一来，操作人员可以直接对版权的归属问题进行验证分析，规避非法盗版问题。

2.4. 数字图像分存及安全性

所谓的数字图像分存主要是指在数字图像信息隐藏的前提条件下，对数字图像信息进行分散处理所运用到的一项技术内容。结合以往的应用经验来看，科学利用数字图像分存技术不仅可以实现对数字图像信息的保密与隐藏处理，同时还可以实现对数字图像信息的分散存储处理，可保障数字图像加密技术应用的安全性与高效性。

参考文献

[1]王雪荣，刘浩. 5G 框架下数字图像加密技术的发展趋势[J]. 广东通信技术，2019，39（08）：2-4.
[2]赵凤怡. 数字图像加密技术的研究[J]. 信息安全与技术，2018，5（09）：7-8+13.