《网络空间安全概论》实验报告

一、实验目的

1. 掌握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.

3. 用卡方检测对可疑图像进行 LSB 隐写检测.

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. 用卡方检测算法对可疑图像进行检测.

三、实验设计

1. LSB 算法在图片中隐藏信息

1.1 实验原理

首先读入图片，准备待隐藏的信息，将其转换为二进制（需要用加密算法进行加密)。遍历图像，对像素的最低 1bit 置 0，同时在该比特位写入 1 位二进制表示隐藏的信息

1.2 过程设计

对于 LSB 实验，采用 Matlab 语言进行编写，环境为 R2022b 版本.

首先将待加密的文本转化为二进制的字符串的格式，然后随机生成与二进制字符串等长度的密钥 key，然后调用实验中的 `encrypt_binary()`函数将二进制信息与 `key` 进行异或运算进行加密，然后读取准备好的图像转化为 `R,G,B` 二进制三元组，先将最低位置 `0`，然后再与加密后的信息逐个进行异或操作，最后输出含有隐藏信息的图片。

2. LSB 算法从图片中获取信息

1.1 实验原理

首先根据 key 预知隐藏信息量（等同于 key），然后提取出像素的最低 1bit，组合成连续 bit 数据，转换为 ASCII 码对比是否与隐藏信息一致。

1.2 过程设计

首先读取含有隐藏信息的图像，根据 key 的长度逐位获取像素最低位的值，然后将提取出来的信息与 key 进行异或操作得到最终二进制字符串，最后再将二进制字符串转化为 ASCII 与实际文本对比。

3. 用卡方检测算法对可疑图像进行检测

调用实验中的 detect_steganography()函数读取包含隐藏信息和未包含隐藏信息的图像，计算每个像素值的期望频率并计算其卡方值，设置自定义阈值后判断卡方值是否超过阈值。若超过，则可能包含隐藏信息；反之则可能不包含隐藏信息。

四、实验过程或算法

1. LSB 算法在图片中隐藏信息

• 首先，读取待隐藏的信息 encrypt_text，并将其转化为二进制字符串，同时编写二进制字符串转化为 ASCII 函数，检验其是否转换正确。

```
% 将待隐藏的信息转化为二进制字符串
encrypt_text = 'CQUWATERMASKEXP';
encrypt_text_binary = string_to_binary(encrypt_text);
fprintf("待隐藏信息二进制为:\n %s \n", encrypt_text_binary)

% 检测二进制转化为字符串是否正确
raw_text = binary_to_string(encrypt_text_binary);
fprintf("待隐藏信息为:\n %s \n", raw_text)
```

其中普通字符串转化为二进制字符串的实现为：

```
function binary = string_to_binary(text)
    binary = '';
    for i = 1:length(text)
        binary = strcat(binary, dec2bin(text(i), 8));
    end
end
```

二进制字符串转化为普通字符串的实现为：

```
function text = binary_to_string(binary)
    text = '';
    for i = 1:8:length(binary)
        tmp_byte = binary(i:i+7);
        text = strcat(text, char(bin2dec(tmp_byte)));
    end
end
```

• 然后随机生成与二进制字符串等长的密钥 key，传递给 encrypt_in_image() 函数进行隐藏，其中 raw2.png 是不包含隐藏信息的图像。

• encrypt_in_image() 函数的实现：

```
function encrypt_in_image(path, encrypt_text1, key)
    % 将字符串转为二进制
    binary_text = string_to_binary(encrypt_text1);
    % 读取图像
    img = imread(path);

    % 加密
    binary_text = encrypt_binary(binary_text, key);

    % 判断图像大小是否能容纳隐藏信息
    if length(binary_text) > size(img, 1) * size(img, 2) * size(img, 3)
        disp('图像无法隐藏下加密信息');
        return;
    end
```

采用 imread 读取图像，并对原始信息采用 key 进行加密，本文直接采用简单的异或运算进行加密得到加密文本，实现如下。并判断文本长度是否超过图片能够容纳加密信息。

```
% 加密二进制信息
function result_binary_str = encrypt_binary(binary_text_str, key_binary)
    % 将字符串形式的二进制文本转换为数字数组
    binary_text = double(binary_text_str) - 48; % 将字符转换为数字（'0' 对应 ASCII 码为 48

    % 逐位异或操作
    result_binary = xor(binary_text, key_binary);

    % 将结果转换为字符串形式的二进制
    result_binary_str = char(result_binary + 48); % 将数字转换为字符
end
```

然后先将最低像素置为 0，再将加密信息隐藏到最低位，最后输出隐藏后的图像 encrypted_image.png.

```
% 隐藏信息
idx = 1;
for x = 1:size(img, 1)
    for y = 1:size(img, 2)
        for z = 1:size(img, 3)
            if idx <= length(binary_text)
                p1 = bitand(img(x, y, z), hex2dec('FE'));    % 低位置为0
                p1 = bitor(p1, str2double(binary_text(idx))); % 隐藏信息
                img(x, y, z) = p1;
                idx = idx + 1;
            else
                break;
            end
        end
    end
end
% 输出包含隐藏信息的图像
imwrite(img, 'encrypted_image.png');
disp('加密完成，加密后的图像为encrypted_image.png');
```

2. 从图像中获取隐藏信息

• 读取 encrypted_image.png 图像，并传入密钥 key，调用 decrypt_from_image() 函数获取信息。

• decrypt_from_image() 函数的实现：

```
function decrypted_text = decrypt_from_image(path, key)
    % 获取隐藏有信息的图片
    img = imread(path);
    % 获取二进制信息
    binary = '';
    idx = 1;
    for x = 1:size(img, 1)
        for y = 1:size(img, 2)
            for z = 1:size(img, 3)
                if idx <= length(key)
                    pl = img(x, y, z);
                    binary = strcat(binary, num2str(bitget(pl, 1)));
                    idx = idx + 1;
                else
                    break
                end
            end
        end
    end
    binary_text = binary(1:length(key));
    binary_text = decrypt_binary(binary_text, key);
    decrypted_text = binary_to_string(binary_text);
end
```

通过 key 的长度预知隐藏信息的长度，逐位获取像素的最低位进行拼接，其次调用解密函数 decrypt_text() 函数进行解密，实现如下：

```
% 解密二进制信息
function result_binary_str = decrypt_binary(binary_text_str, key_binary)
    % 将字符串形式的二进制文本转换为数字数组
    binary_text = double(binary_text_str) - 48; % 将字符转换为数字（'0' 对应 ASCII 码

    % 逐位异或操作
    result_binary = xor(binary_text, key_binary);

    % 将结果转换为字符串形式的二进制
    result_binary_str = char(result_binary + 48); % 将数字转换为字符
end
```

最后再将二进制字符串转化为普通字符串，与实际字符串进行对比，判断是否一致，若一致则说明加解密过程正确；反之，则不正确。

```
% 进行信息解密
decrypted_text = decrypt_from_image('encrypted_image.png', key);

fprintf('解密后的信息为 %s\n', decrypted_text);
if strcmp(encrypt_text, decrypted_text)
    disp('加解密成功!');
end
```

3. 使用卡方检测算法对可以图像进行检测

将含有隐藏信息的图像读入 detect_steganography()函数，将图像转换为一维向量后计算每个像素值的期望频率，并计算卡方值。使用 Matlab 的 chi2inv 函数设置阈值，本实验中以 chi2inv(0.95, 1)模拟卡方分布并返回阈值，若计算的卡方值大于阈值，则判定图像中可能存在隐藏信息。

```
% 提取LSB
lsb_r = bitget(r, 1);
lsb_g = bitget(g, 1);
lsb_b = bitget(b, 1);

% 将所有通道的LSB合并为一个序列
lsb_all = [lsb_r(:); lsb_g(:); lsb_b(:)];

% 计算卡方统计量
% 假设没有隐藏信息时，LSB应该是随机的，即0和1出现的概率应该是相等的
observed_0 = sum(lsb_all == 0);
observed_1 = sum(lsb_all == 1);
expected_0 = length(lsb_all) / 2;
expected_1 = length(lsb_all) / 2;
chi_squared = ((observed_0 - expected_0)^2 / expected_0) + ((observed_1 - expected_1)^2 / expected_1);

% 使用95%的置信度（即5%的显著性水平）进行检验：
alpha = 0.05;
df = 1; % 自由度
critical_value = chi2inv(1-alpha, df);

disp("检测卡方值为:");
disp(chi_squared);
disp("临界值为:");
disp(critical_value);
% 比较卡方统计量和临界值
if chi_squared > critical_value
    disp('图像可能包含隐藏信息');
else
    disp('图像可能没有包含隐藏信息');
end
```

五、验过程中遇到的问题及解决情况

1. **问题：由 randi 生成的随机数无法直接与二进制字符串进行异或运算.**

解决：将二进制字符串通过强制转换转化为 double 型的数字，再与随机数 key 进行异或操作。最后再将结果强制转化为 char 字符串。

2. **问题：在从图像中提取信息时程序运行时间过长。**

解决：因为图像像素值密度过大，如果将所有信息提取后再截断会耗时很久甚至无法与运行完成，因此只需根据 key 的长度提取到信息的长度即可中断操作。

六、实验结果及分析和（或）源程序调试过程

1. LSB 加解密结果

- **待隐藏信息为：**CQUWATERMASKEXP
- **待隐藏信息的二进制为：**

01000011010100010101010101010111010000010101010001000101010101001001101010000010101001101001011010001010101100001010000

- **生成的密钥 key 为：**

11110100100110101101111000110111111100000001111100101110011100100100101011000000101010110011010110111110001110100010000

- **加密后的二进制为：**

10110111110010111000101101100000010110001010010110110101100100000000001111000000111111000011111101111101101100010011100000

• **不含加密信息的图像：**

• **含有加密信息的图像：**

可以看出两张图像几乎毫无差别，达到了 LSB 的效果。

• **解密后的信息为：** CQUWATERMASKEXP，与原始信息一致，说明加解密过程正确。

```
加密后的信息为:
10110111110010111000101101100000101100010100101101101011001000000000011110000001111110000111111
加密完成，加密后的图像为encrypted_image.png
解密后的信息为 CQUWATERMASKEXP
加解密成功!
```

2. 卡方检测

首先对比不含隐藏信息的图像和含有隐藏信息的图像的直方图，可以发现两张图几乎毫无差别，这是因为由于加密的信息转化为二进制形式仅有 120bit 大小，远小于给定的图像的像素值 bit，因此即使将信息隐藏到原始图像中，对整体像素的分布的影响也很小。

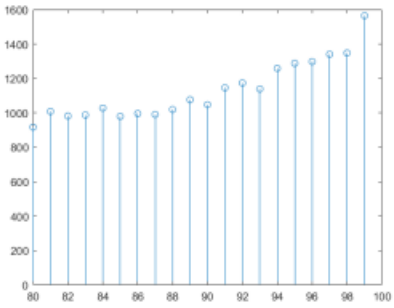

图 1 不含隐藏信息的图像直方图分布

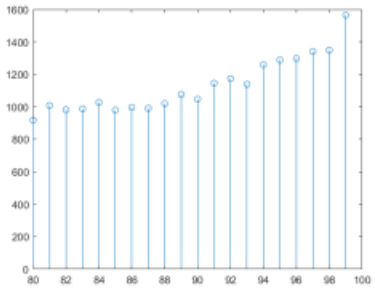

图 2 含隐藏信息图像直方图分布

同时对两张图进行检验，得到相同类似的结果，可以看出不含隐藏信息的卡方值要小于含隐藏信息的值，但两者均被判断为可能不含隐藏信息。这是因为在本实验中即使对图像信息进行了数据隐藏，但是因为隐藏的数据量过小，难以对原始像素分布产生较大的影响，导致隐藏了信息的图像经过卡方检测也被认定为不含隐藏信息。

```
--------对不含隐藏信息进行检测-------
检测卡方值为:
    1.3859

临界值为:
    3.8415

图像可能没有包含隐藏信息
--------对含隐藏信息进行检测-------
检测卡方值为:
    1.4072

临界值为:
    3.8415

图像可能没有包含隐藏信息
```