



南開大學
Nankai University

网络空间安全学院
信息隐藏实验报告

LSB 隐藏法实践

姓名：魏伯繁

学号：2011395

专业：信息安全

2023 年 4 月 10 日

目录

1 实验要求	2
1.1 实验目的	2
1.2 实验环境	2
1.3 实验要求	2
2 基于 LSB 的二值图像隐藏	2
2.1 实验原理	2
2.2 流载体上的 LSB	2
2.3 基于伪随机置换的 LSB	3
2.4 代码实现	3
3 基于 LSB 的字符串信息隐藏	5

1 实验要求

1.1 实验目的

- 实现将二值图像嵌入到位图中；
- 实现将学号（一个整数）嵌入到位图中。

1.2 实验环境

- Win10 操作系统
- matlab2018a
- bmp 文件格式

1.3 实验要求

- 写出实验报告，含程序代码和截图，word 或 pdf 格式。
- 将代码、实验报告和实验图片打包提交

2 基于 LSB 的二值图像隐藏

2.1 实验原理

LSB (Least Significant Bit) 方法是数字水印技术中的一种常用方法，其思想是将一张水印图像嵌入到载体图像中，具体实现过程是将水印图像的每个像素的二进制编码的最低位（即最不重要的一位）插入到载体图像中相应像素的二进制编码的最低位中。嵌入的时候，要避免修改像素的最高位，以免被人眼察觉。提取时，则需将含水印信息的载体图像的每个像素的二进制编码的最低位提取出来，组成一个新的图像，即原始水印图像。

LSB 方法的优点是简单、嵌入速度较快，不用强制要求载体图片与水印图片尺寸完全一致，且不容易被肉眼察觉。缺点是该方法的鲁棒性较差，当载体图像受到攻击（如进行滤波、几何变换等）时，提取水印的准确度会受到影响。

2.2 流载体上的 LSB

流媒体 (Streaming Media) 中的数字水印技术是为了提高信息攻击的鲁棒性和隐蔽性，解决流式多媒体内容的相关问题而发展的。在流媒体中，数字水印技术采用了一种称为流载体的 LSB (Least Significant Bit) 方法对流式多媒体进行加水印。与传统 LSB 方法不同，流载体 LSB 方法是将水印数据流直接嵌入到流式数据中，而不是将水印图像嵌入到静态图片中。

其流程如下：

1. 读取载体流媒体文件和水印文件；
2. 对于载体媒体流中的每个视频帧或音频采样，将水印数据流中的一部分嵌入到其二进制编码的最低位中，在多次嵌入后，每个视频帧或音频采样都含有的水印信息就可以完全应用到媒体流中；
3. 将嵌入水印的媒体流保存为新的流媒体文件。

提取流媒体 LSB 水印也类似于传统的 LSB 水印提取方法，主要包括以下步骤：

1. 读取含有嵌入水印的流媒体文件；
2. 对于流媒体文件中的每个视频帧或音频采样，提取其二进制编码中的最低位，组成新的水印数据流
3. 最终得到的水印数据流即为嵌入到流媒体中的原始水印数据流。

流载体 LSB 方法可以解决传统 LSB 方法在流式多媒体内容处理过程中产生的问题，如多次对一张图片进行 LSB 嵌入时必须对其所有像素进行遍历，难以应对实时性需求等问题。而且，流载体 LSB 方法的加水印和提取过程可在流式数据传输的过程中进行，既不需要先将所有数据存储下来再处理，又不会影响客户端的观看体验，而且不会影响数据源的原始信息。

总之，流载体 LSB 方法是数字水印技术中的一种有效方法，适用于流媒体中数字水印的加密和提取。

2.3 基于伪随机置换的 LSB

伪随机置换 (Pseudo Random Permutation, PRP) 是一种基于密钥的加密算法，通常用于保护数字水印的加密和提取。PRP 算法的本质是通过使用一个密钥和一个伪随机置换，将输入数据混淆为一个等长的输出序列，以达到加密的目的。

PRP 算法可以应用于 LSB 技术中，以增强加密和解密过程的隐蔽性和安全性。其基本思路是将 PRP 算法生成的伪随机置换应用于水印数据，然后在对载体图像进行 LSB 加密嵌入前，使用 PRP 嵌入水印图像，再将其插入到载体图像中的 LSB 位置上，从而实现水印的加密嵌入。

具体的 PRP LSB 嵌入流程如下：

1. 首先使用 PRP 加密算法生成一个随机的置换数列，作为 PRP 的密钥；
2. 将水印数据按照相应的 PRP 算法进行变换，得到 PRP 加密后的水印数据；
3. 将 PRP 加密后的水印数据嵌入到载体图像中，具体方法同普通的 LSB 算法一样。

PRP 算法可以大大提高 LSB 加密技术的安全性。由于 PRP 嵌入的水印数据已经被随机和混淆，攻击者在试图通过对 LSB 嵌入位置上的最低位进行修改或者分析时将变得异常困难。而且，PRP 加密算法还具有较强的可控性，使得加密后的水印数据与原始数据极为相似，有助于水印的隐蔽性。

PRP LSB 提取的过程就是 PRP 嵌入的逆操作。在 PRP LSB 提取的过程中，攻击者需要使用 PRP 算法解密加密的水印数据，再进行分析或者修改，因此 PRP LSB 技术可以大大增强数字水印的安全保障。

2.4 代码实现

在本次实验中，我们希望将南开的二值图像校徽嵌入到 lena 灰度图像中。

首先，我们将原图像读入内存，并对图像做必要处理，我们需要将二者的分辨率置为一致，然后将 lena 图像转换为灰度图像，再将南开的校徽转换成黑白二值图像。

```
1 x=imread('lena.bmp'); % 载体图像
2 m=imread('nk.bmp'); % 水印图像
```

```

3      y=imresize(x, [256, 256]);% 调整图像分辨率为 256x256
4      x=rgb2gray(y);% 转换成灰度图像
5
6      mm=imresize(m, [256, 256]);% 调整图像分辨率为 256x256
7      II=rgb2gray(mm);% 转换成灰度图像
8      m=imbinarize(II);% 转换成二值图像
9      figure;
10     imshow(x,[]);% 展示灰度图像 x
11     figure;
12     imshow(m,[]);% 展示二值图像 m
13
14     [index_row,index_col]=size(x);
15     AddPrint=uint8(zeros(size(x)));

```

我们使用的图片如下图所示：

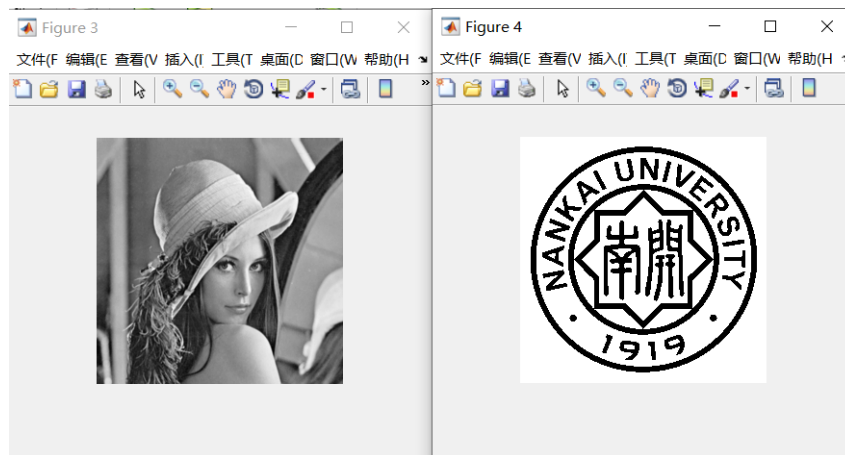


图 2.1: 样例图像展示

然后，该段代码的核心就在于这一段，通过遍历像素的方式，根据 LSB 技术将二值化的水印图像嵌入到灰度图像的最低位中，并将其保存为'lsb_watermarked.bmp'；其中 bitset() 函数将水印数据 m(i,j) 嵌入到灰度图像的最低位，而 bitget() 函数则从嵌入后的图像中提取出最低位，得到的结果就是二值化的水印图像。

```

1      [index_row,index_col]=size(x);
2      AddPrint=uint8(zeros(size(x)));
3
4      for i=1:index_row
5          for j=1:index_col
6              AddPrint(i,j)=bitset(x(i,j),1,m(i,j));
7          end
8      end
9

```

```

10     imwrite(AddPrint,'lsb_watermarked.bmp','bmp');
11     figure;
12     imshow(AddPrint,[]);
13     title("AddPrint Image");

```

最终提取嵌入在载体图像中的水印，即通过遍历像素的方式，获取 AddPrint 图像中每个像素的最低位，得到的结果就是 Print 图像；将 AddPrint 图像和 Print 图像展示出来，观察水印嵌入和提取的效果。

```

1     [index_row2,index_col2]=size(AddPrint);
2     Print=uint8(zeros(size(AddPrint)));
3
4     for i=1:index_row2
5         for j=1:index_col2
6             Print(i,j)=bitget(AddPrint(i,j),1);
7         end
8     end
9
10    figure;
11    imshow(Print,[]);
12    title("Print");

```

增加水印的图像以及通过增加水印后的图片如下图所示

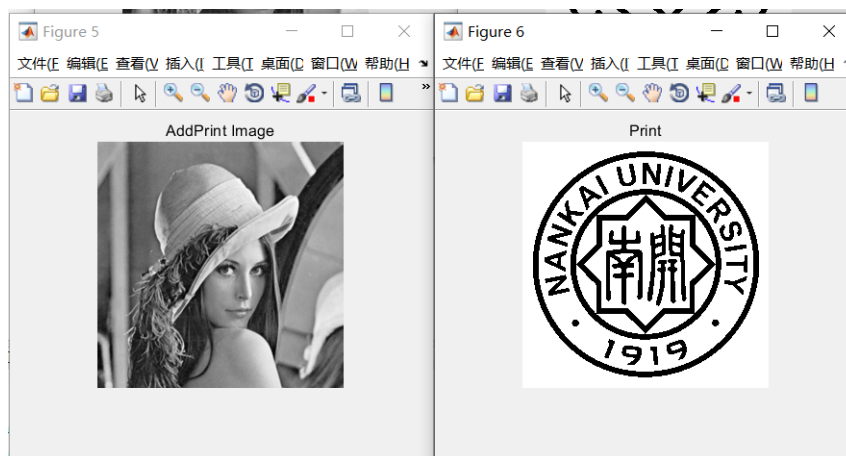


图 2.2: 实验效果展示

3 基于 LSB 的字符串信息隐藏

首先，我们对必要的信息读入内存并进行必要的处理，即读入载体图像'lenna.bmp'，并展示出来；设定要嵌入的信息为整数为 2011395

```
1 x=imread('lena.bmp'); % 载体图像
2 m=2011395; % 要嵌入的信息
3 imshow(x,[])
4 [index_row,index_col]=size(x);
5 WaterMarked=uint8(zeros(size(x)));
```

接下来，遍历灰度图像的每个像素，如果像素在第一行且列数小于等于 21，则将要嵌入的整数的每个二进制位依次嵌入到像素的最低位中，并保存到 WaterMarked 图像中；否则直接将像素保存到 WaterMarked 图像中；

```
1 for i=1:index_row
2     for j=1:index_col
3         if i==1 && j<=21
4             tem=bitget(m,j);
5             WaterMarked(i,j)=bitset(x(i,j),1,tem);
6         else
7             WaterMarked(i,j)=x(i,j);
8         end
9     end
10 end
```

最后，将 WaterMarked 图像保存到'lsb_int_watermarked.bmp'中，并展示出来，以观察整数嵌入的效果；遍历 WaterMarked 图像的第一行前 21 个像素，从中提取出嵌入的整数。具体实现是将每个像素的最低位依次提取出来，然后通过 bitset() 函数将这些位拼接成一个二进制数，并将其转换为十进制数，得到的结果即为嵌入的整数。

```
1 imwrite(WaterMarked,'lsb_int_watermarked.bmp','bmp');
2 figure;
3 imshow(WaterMarked,[]);
4 title("WaterMarked Image");
5 WaterMark=0;
6 for j=1:21
7     tem=bitget(WaterMarked(1,j),1);
8     WaterMark=bitset(WaterMark,j,tem);
9 end
```

实验效果如下图所示，可以发现原来的 lena 图没有明显的变化，但是却可以从图像中提取出对应的学号。

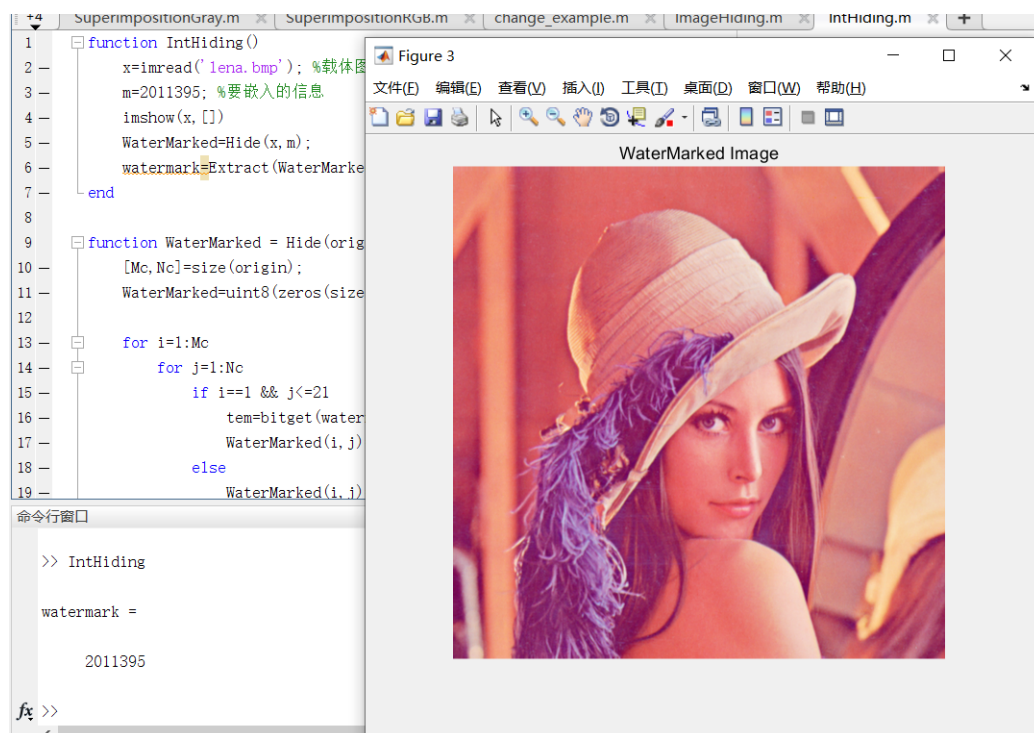


图 3.3: 实验效果展示

我们观察最后的控制台输出

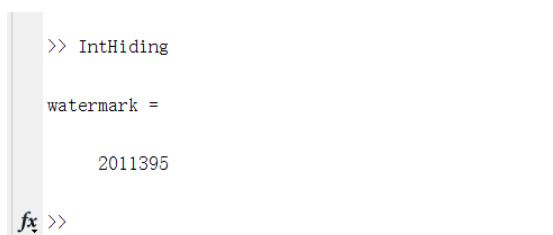


图 3.4: 可以看到控住台上的输出与加密前的一致