信息安全本科生 201*—201*学年 第（二）学期

《信息隐藏与数字水印》实验报告-范例

学号：14*****　姓名：* * *　成绩：_____

Project2 媒体格式分析与信息隐藏

一、 实验目的

1. 熟悉 wav 媒体文件格式
2. 掌握音频文件 wav 信息隐藏方法
3. 用 matlab 实现 wav 文件的信息隐藏过程

二、 实验原理

(一) 音频信息隐藏

由于人耳听觉系统（HAS）较之视觉系统（HVS）具有较宽的动态范围和较高的灵敏度，因此相对于图像和视频的信息隐藏而言，音频载体的信息隐藏技术更具有挑战性。目前主要的音频信息隐藏技术分为**时域和变换域**音频信息隐藏方法两类。

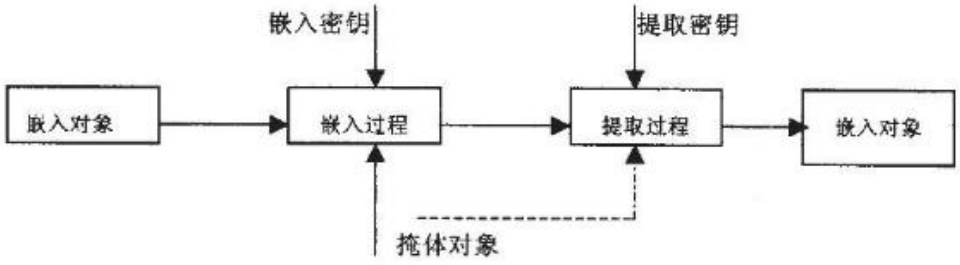
经典时域音频信息隐藏技术：到目前为止，公认比较成熟的时域音频信息隐藏技术有四种：最不重要位法、回声隐藏法、相位编码法、扩频法。

最不重要位（LSB）的隐藏算法是最早应用于音频信息隐藏领域的算法。它的基本思想是用秘密数据替换原始音频信号采样值的最低几个比特位，达到隐藏的目的。LSB 算法具有计算复杂度低、可实时实现及通用性等优点，但其抵抗攻击的能力较弱。

回声隐藏法是通过引入回声来将秘密信息嵌入到载体中。与其他方法不同，回声隐藏法对载体音频信号的改变，考虑的是环境条件而不是随机噪声的特性，因而具有较强的抵抗主动攻击的能力。但信道噪声、任何形式的篡改都会直接影响算法的正确提取率。

而相位编码则是利用 HAS 对人耳对绝对相位的不敏感性及对相对相位的敏感性，将代表秘密信息的参考相位替换原始音频段的绝对相位，并对其他音频段进行相应调整，以保持各段之间的相对相位不变。

扩频法的基本思想是利用扩频调制技术将秘密信息扩展到整个可听频谱范围内，再将扩频后的秘密信息叠加到原始的音频信号中完成隐藏。



(二)WAV 文件

WAV 文件是 Microsoft 公司的音频文件格式。文件头包含 40 字节,这些信息若被修改,则文件就不能被播放器识别为 WAV 格式而不能播放。文件正文来源于对声音模拟波形的采样。用不同的采样频率对声音的模拟波形进行采样可以得到一系列离散的采样点,以不同的量化位数(8 位或 16 位) 把这些采样点的值换成二进制数, 就产生了声音的 WAV 文件,即波形文件。WAV 文件是由采样数据组成的,所以它所需要的存储容量很大。用下列公式可以简单地推算出 WAV 文件所需的存储空间的大小。WAV 文件的字节数/ 秒= 采样频率×量化位数声道数/ 8WAV 文件所需的存储容量相当可观,是作为掩护媒体的良好材料。

1. RIFF 文件与 WAV 文件

在 Windows 环境下，大部分多媒体文件都依循着一种结构来存放信息，称为资源互换文件格式(Resources Interchange File Format)，简称 RIFF。比如声音的 WAV 文件，视频的 AVI 文件，动画的 MMM 文件等均是由此结构衍生出来的。所以，要掌握多媒体文件格式，首先得认识 RIFF 的结构。

RIFF 是一种树状结构，其基本组成单位是 chunk(即块)，每个 chunk 由辨识码，数据大小和数据组成，如下表↓ 可以看出，一个 chunk 的长度，就是数据的大小加上 8Byte。

辨识码	4Byte	4 个 ASCII 码
数据大小	4Byte	紧跟其后的数据长度
数据	……	……

一般而言,chunk 本身不允许内部再包含 chunk，但有两个例外:以"RIFF"和以"UST"为辨识码的 chunk。针对这两种 chunk，RIFF 又从原先的"裸数据"中切出 4Byte 作为"格式辨别码",如下图所示：

辨识码		4Byte	4 个 ASCII 码
数据大小		4Byte	紧跟其后的数据长度
数据	格式辨识码	4Byte	4 个 ASCII 码（例如 wave）
	数据	……	……

对 RIFF 的树状结构有所了解之后,可以知道它相当于一个根目录,而格式辨识码则相当于具体的盘符如 C:,D:等等.Windows 下的各种多媒体文件格式就如同在磁盘机下规定只能存放怎样的目录,而在该目录下仅能存放何种数据。

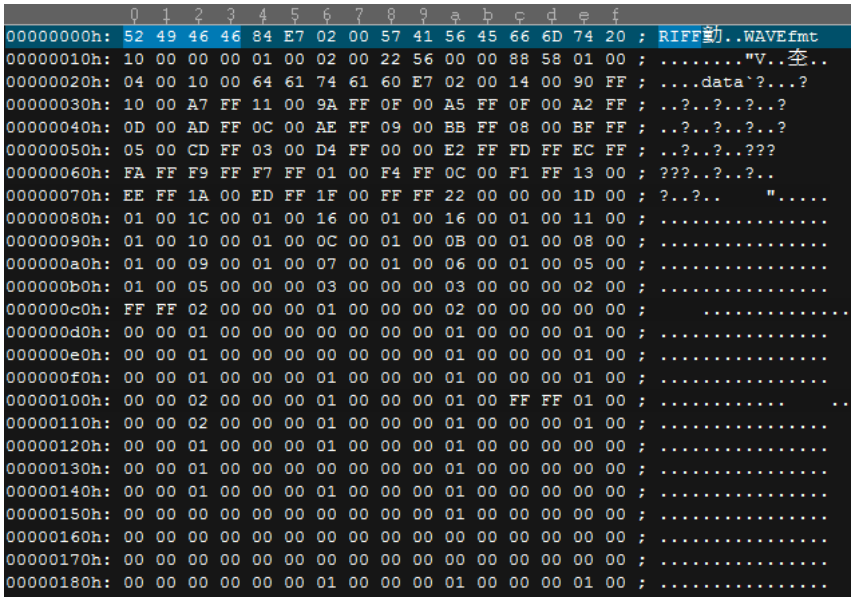
2. WAV 文件头

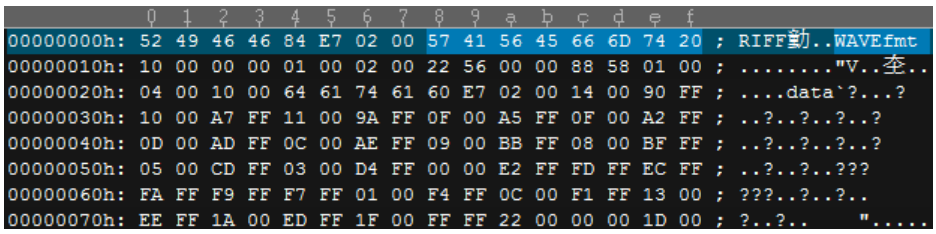
WAV 就是波形音频文件(Wave Audio)，是 Windows 中用来表示数字化声音的一种标准格式，其文件扩展名为.wav，是一种非常简单的 RIFF 文件，格式辨识码为"WAVE"。整个 WAV 文件分成两部分:文件头和数据块。WAV 格式文件主要有两种文件头。

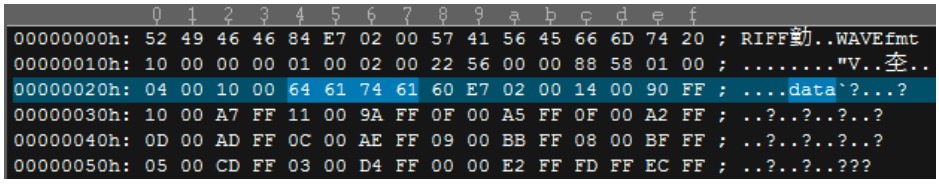
(1) 标准的 40 字节文件头

包含两个 chunk: <fmt-chunk>,<wave-data>，这两个子块都是一个 wav 文件必须包含的：

	标识符(Riff)	4Byte	52 49 46 46
	数据大小(Size)	4Byte	
	格式辨别码（”WAVE”）	4Byte	57 41 56 45
Fmt 子块	“fmt”	4Byte	66 6D 74 20
	Sizeof(PCMWAVEFORMAT)	16Byte	10 00 00 00
	PCMWAVEFORMAT	4Byte	
Data 子块	“data”	4Byte	64 61 74 61
	波形声音值字节数		
	波形声音值		







文件头以”RIFF”作为标示，然后紧跟着为 size 字段，该 size 是整个 wav 文件大小减去 ID 和 Size 所占用的字节数，即 fileLen - 8 =size。然后是 Type 字段,为'WAVE',表示是 wav 文件。

Format 子块以”fmt”作为标示，一般情况下 size 为 16，此时没有附加消息；如果为 18，则最后多了 2 字节的附加信息。

Data 子块中装的是真正的声音数据，除非安装其他特殊软件，否则只有

WAVE_FORMAT_PCM 一种数据格式，即脉冲编码调制（PCM）。针对此格式，windows 中 data 子块的数据存放形式如下：

单声道 8 位量化	取样 1	取样 2	取样 3	取样 4
	声道 0	声道 0	声道 0	声道 0
双声道 8 位量化	取样 1		取样 2	
	声道 0（左）	声道 1（右）	声道 0（左）	声道 1（右）
单声道 16 位量化	取样 1		取样 2	
	声道 0(低位组)	声道 0(高位组)	声道 0(低位组)	声道 0(高位组)
双声道 16 位量化	取样 1			
	声道 0（左）（低位元组）	声道 0（左）(高位元组)	声道 1（右）（低位元组）	声道 1(右)(高位元组）

(三)LSB（LeastSignificant Bits）算法

将秘密信息嵌入到载体图像像素值的最低有效位，也称最不显著位，改变这一位置对载体图像的品质影响最小。

1. 信息隐藏

间接嵌入是将隐藏媒体进行加工以反映要隐藏的信息。间接嵌入可以表示为 A'= Func (M ,A)，载体文件 A 可以是图像、声音等，只要 A '和 A 给人们的感觉差别不大,就可以认为 Func 是一个好的隐藏算法。

本次实验中，载体文件 A 取为 16 位 PCM 音频文件。对于普通音频文件,至少要达到 8 000/ s 的采样频率，因而每个采样点用 16 位表示编码达 128K/ s，因此 WAV 文件为待隐藏信息提供了广阔的隐藏空间。一段 16 位音频文件，由 40 字节的**文件头**和**音频数据**部分组成，其中**文件头不能隐藏信息**，从第 41 字节以后为音频数据部分，可以隐藏信息。音频数据部分是由一系列的 16 位二进制数所组成,由于每个 16 位二进制数中“1”的个数或者为奇数或者为偶数，约定:若一个字节中“1”的个数为奇数,则该字节为奇性字节，用“1”表示；若一个字节中“1”的个数为偶数,则称该字节为偶性字节，用“0”表示。用每个字节的奇偶性来表示隐藏的信息。

举例: 设一段 16 位 WAV 文件的数据为

1000100100000010 ,1001101100000010 ,0110001100000010 ,0101101000000010 ,则其字节的奇偶排序为:0 ,0 ,1 ,1 。现在需要隐藏信息 0xa ，由于 0xa 转化为二进制为 1010，将这 2 个数列相比较，发现第 1、4 位不一致，于是对这段 WAV 文件数据的 PCM 编码的较低位进行调整以使其奇偶性与要隐藏的对应位一致:第 1 位:将 1000100100000010 变为 1000100100000011 ，则该字节由偶变为奇;第 4 位:将 0101101000000010 变为 0101101000000011 ，则该字节由奇变为偶。经过这样的调整，此数据段双字节的奇偶性便与 0xa 转化的 4 位二进制数完全相同，这样，8 个字节便隐藏了半个字节的信息。注:若对 PCM 编码的较高位进行调整,则易使声音失真。

2. 信息提取

信息提取是把隐藏的信息从伪装媒体中读取出来,其过程和步骤正好与信息嵌入相反。其步骤如下：

(1) 判断 WAV 文件数据部分每个 PCM 编码的奇偶性,若编码中“1”的个数为偶

数,则输出“0”;若编码中“1”的个数为奇数,则输出“1"

(2) 每判断 8 个字节,便将输出的 8 位数组成一个二进制数(先输出的为高位)

(3) 经过上述处理,得到一系列 8 位二进制数,便是隐藏信息的代码,将代码转换成文本(或图像或声音) ,就是隐藏的信息。

三、 程序实现

隐藏核心代码

```
for k=1:audiolength
    if k>msglength
        break;
    end
    count=0;
    for i=1:16   %统计第 k 个字节中 1 的个数
        temp=bitget(new_audio(40+k),i);
        if temp==1
            count=count+1;
        end
    end
    if mod(count,2)==1   && msg(k)==0   %奇数变为偶数
        if bitget(new_audio(40+k),1)==1
            new_audio(40+k)=bitset(new_audio(40+k),1,0);
        else
            new_audio(40+k)=bitset(new_audio(40+k),1,1);
        end
    else if mod(count,2)==0 && msg(k)==1   %偶数变为奇数
            if bitget(new_audio(40+k),1)==1
                new_audio(40+k)=bitset(new_audio(40+k),1,0);
            else
                new_audio(40+k)=bitset(new_audio(40+k),1,1);
            end
        end
    end
end
fileId2=fopen('MarkedAudio.wav','wb');
fwrite(fileId2,new_audio,'uchar');
fclose(fileId2);
```

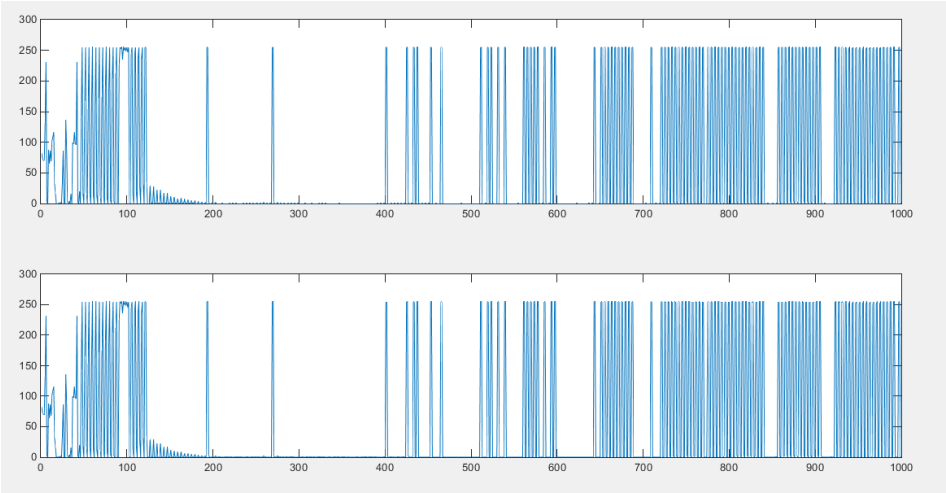
提取信息核心代码

```
for i=1:msglenght
    count=0;
    for j=1:16   %统计第 k 个字节中 1 的个数
        temp=bitget(marked(i),j);
        if temp==1
            count=count+1;
        end
```

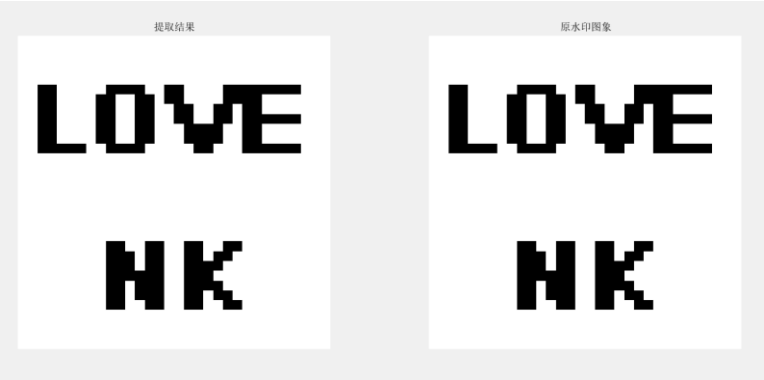
```
        end
        if mod(count,2)==1          %若含奇数个 1，则秘密信息位是 1
            extractmsg(i)=1;
        else if mod(count,2)==0    %若含偶数个 1，则秘密信息位是 0
                    extractmsg(i)=0;
            end
        end
end
```

四、 程序结果

音频文件嵌入秘密图片之后的波形图对比：



从直观上看，两个音频“并无区别”，另外通过聆听两个.wav 格式的音频文件，人耳也并不能听出有任何区别。



上面是嵌入的水印图片和从载体音频中提取出来的水印结果，由此可以还原出原始的密文。

五、 总结

本次实验中使用的 WAV 音频文件信息隐藏算法具有较好的透明性，利用人类视觉系统或人类听觉系统属性，经过一系列隐藏处理，使目标数据没有明显的降质现象，而隐藏的数据也无法人为地看见或听见。

目前的语音信息隐藏系统中，尚没有一种能够在各种攻击下都表现出良好的

健壮性，因此仍需要对现有的隐藏算法的鲁棒性、安全性等特性进行研究，结合数字信号处理技术以及音频信息的具体特点，提出更好的语音中信息隐藏的切入点以及相关算法，并进一步提高信息隐藏的容量，使其在更加广阔的范围内得到充分应用。