实验三 实现LSB隐写算法

实验环境： os：Windows 10 arch：x86-64

实验思路

存入思路核心：将secret.zip文件以字符串读入再转换成bit串，存入pic.bmp的rgb段每个字节最低位
提取思路核心：将存入生成的secret.bmp图片解析，去掉信息头，拿到rgb段，根据输入的读取大小，提取对应字节的最低位组成字符串，写入ans.zip

隐写核心函数 embedMessage

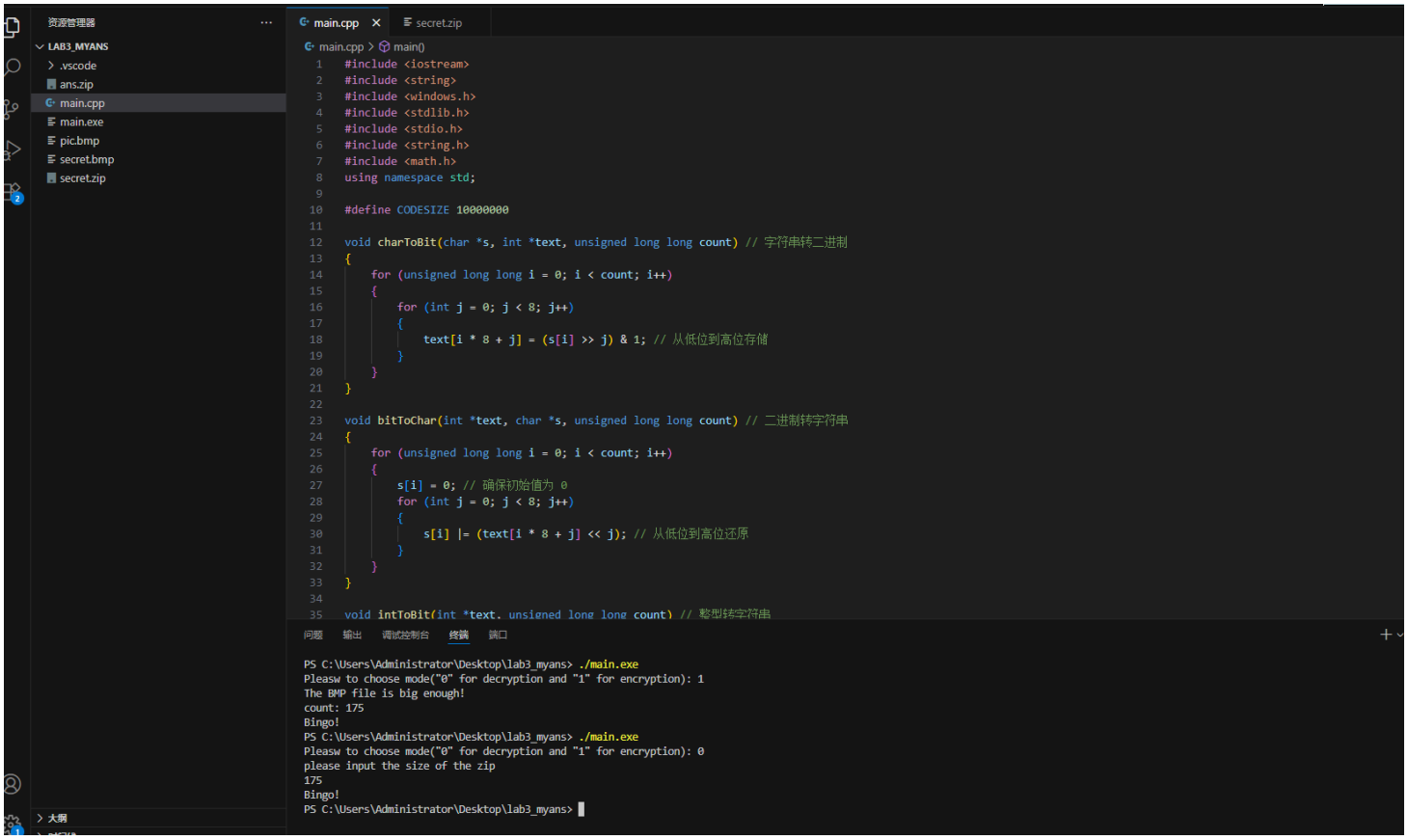
```
void embedMessage(int* l,int* f,int len)
{
    // 已实现
    for(int i=0;i<len;i++)
    {
        l[i]=f[i]|(l[i]&0xFE);
    }
}
```

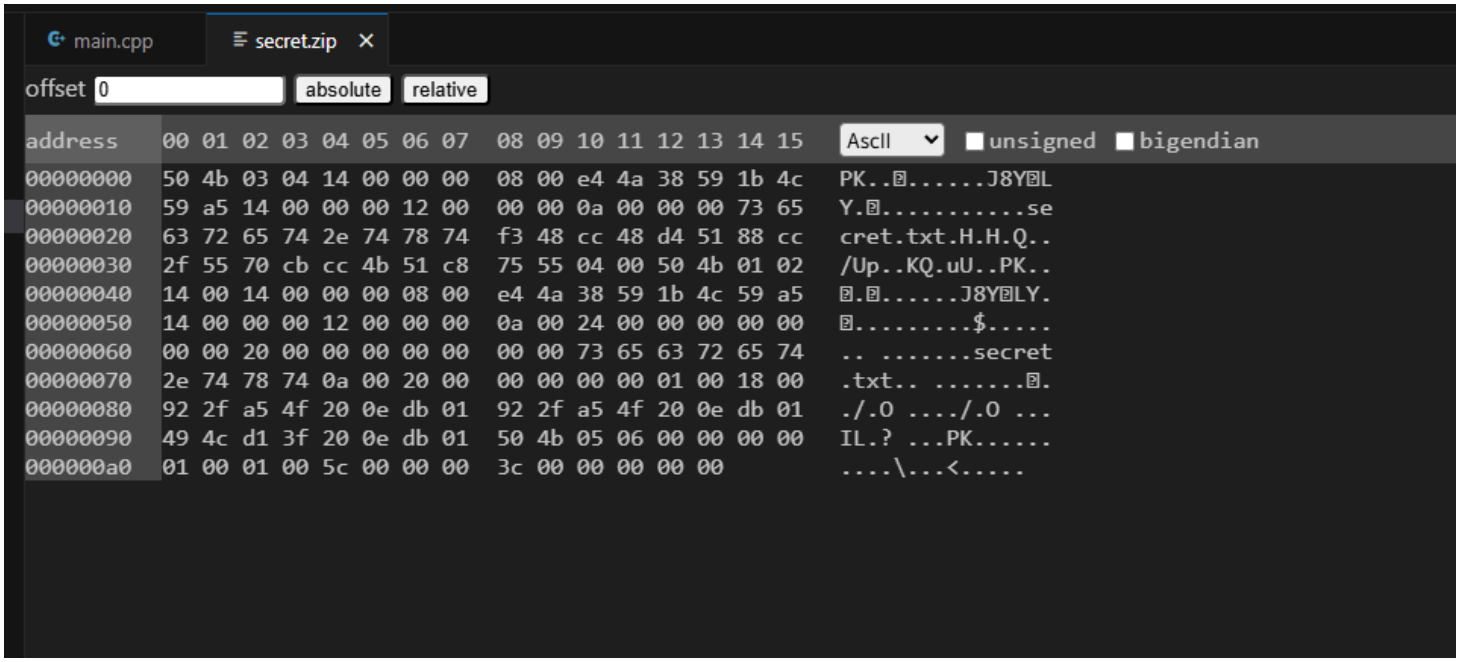
提取核心函数 extractMessage

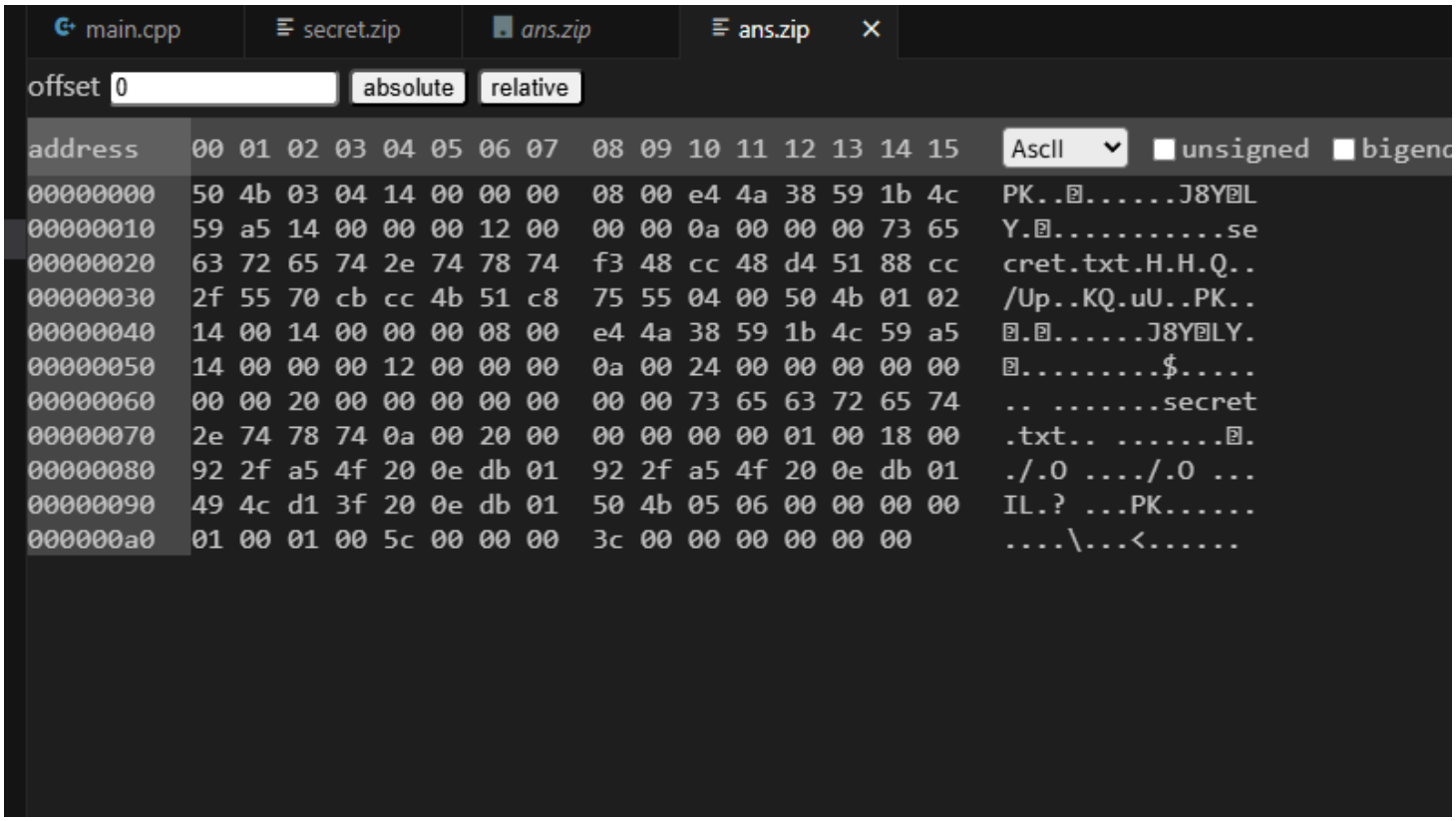
```
void extractMessage(int* r,int* f,int len)
{
    // 已实现
    for(int i=0;i<len;i++)
    {
        f[i]=r[i]&1;
    }
}
```

注：为了匹配日常使用习惯，我将模版里面bit2char和char2bit两个函数都进行了相应修改，让每个字节的最低位都出现在bit串相应索引小端

运行截图及结果展示


资源管理器
LAB3_MYANS
.vscode
ans.zip
main.cpp
main.exe
pic.bmp
secret.bmp
secret.zip
main.cpp
secret.zip
main.cpp > main()
#include <iostream>
#include <string>
#include <windows.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>
using namespace std;
#define CODESIZE 10000000
void charToBit(char *s, int *text, unsigned long long count) // 字符串转二进制
{
for (unsigned long long i = 0; i < count; i++)
{
for (int j = 0; j < 8; j++)
{
text[i * 8 + j] = (s[i] >> j) & 1; // 从低位到高位存储
}
}
}
void bitToChar(int *text, char *s, unsigned long long count) // 二进制转字符串
{
for (unsigned long long i = 0; i < count; i++)
{
s[i] = 0; // 确保初始值为 0
for (int j = 0; j < 8; j++)
{
s[i] |= (text[i * 8 + j] << j); // 从低位到高位还原
}
}
}
问题 输出 调试控制台 终端 端口
PS C:\Users\Administrator\Desktop\lab3_myans> ./main.exe
Pleasw to choose mode("0" for decryption and "1" for encryption): 1
The BMP file is big enough!
count: 175
Bingo!
PS C:\Users\Administrator\Desktop\lab3_myans> ./main.exe
Pleasw to choose mode("0" for decryption and "1" for encryption): 0
please input the size of the zip
175
Bingo!
PS C:\Users\Administrator\Desktop\lab3_myans>


main.cpp
secret.zip
offset 0
absolute
relative
address 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 Ascll unsigned bigendian
00000000 50 4b 03 04 14 00 00 00 08 00 e4 4a 38 59 1b 4c PK..□.....J8Y□L
00000010 59 a5 14 00 00 00 12 00 00 00 0a 00 00 00 73 65 Y.□.....se
00000020 63 72 65 74 2e 74 78 74 f3 48 cc 48 d4 51 88 cc cret.txt.H.H.Q..
00000030 2f 55 70 cb cc 4b 51 c8 75 55 04 00 50 4b 01 02 /Up..KQ.uU..PK..
00000040 14 00 14 00 00 00 08 00 e4 4a 38 59 1b 4c 59 a5 □.□.....J8Y□LY.
00000050 14 00 00 00 12 00 00 00 0a 00 24 00 00 00 00 00 □.....$.....
00000060 00 00 20 00 00 00 00 00 00 00 73 65 63 72 65 74 .. ..secret
00000070 2e 74 78 74 0a 00 20 00 00 00 00 00 01 00 18 00 .txt.. .....□.
00000080 92 2f a5 4f 20 0e db 01 92 2f a5 4f 20 0e db 01 ./.O ..../.O ...
00000090 49 4c d1 3f 20 0e db 01 50 4b 05 06 00 00 00 00 IL.? ...PK.....
000000a0 01 00 01 00 5c 00 00 00 3c 00 00 00 00 00 ....\...<.....


main.cpp
secret.zip
ans.zip
ans.zip
offset 0
absolute
relative
address 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 Ascll unsigned bigend
00000000 50 4b 03 04 14 00 00 00 08 00 e4 4a 38 59 1b 4c PK..□.....J8Y□L
00000010 59 a5 14 00 00 00 12 00 00 00 0a 00 00 00 73 65 Y.□.....se
00000020 63 72 65 74 2e 74 78 74 f3 48 cc 48 d4 51 88 cc cret.txt.H.H.Q..
00000030 2f 55 70 cb cc 4b 51 c8 75 55 04 00 50 4b 01 02 /Up..KQ.uU..PK..
00000040 14 00 14 00 00 00 08 00 e4 4a 38 59 1b 4c 59 a5 □.□.....J8Y□LY.
00000050 14 00 00 00 12 00 00 00 0a 00 24 00 00 00 00 00 □.....$.....
00000060 00 00 20 00 00 00 00 00 00 00 73 65 63 72 65 74 .. ..secret
00000070 2e 74 78 74 0a 00 20 00 00 00 00 00 01 00 18 00 .txt.. .....□.
00000080 92 2f a5 4f 20 0e db 01 92 2f a5 4f 20 0e db 01 ./.O ..../.O ...
00000090 49 4c d1 3f 20 0e db 01 50 4b 05 06 00 00 00 00 IL.? ...PK.....
000000a0 01 00 01 00 5c 00 00 00 3c 00 00 00 00 00 00 ....\...<.....