信息隐藏技术第二次大作业展示

媒体文件格式剖析与信息隐藏实践

汇报人： 魏伯繁

目 录
ONTENTS

01

OGG文件格式简介

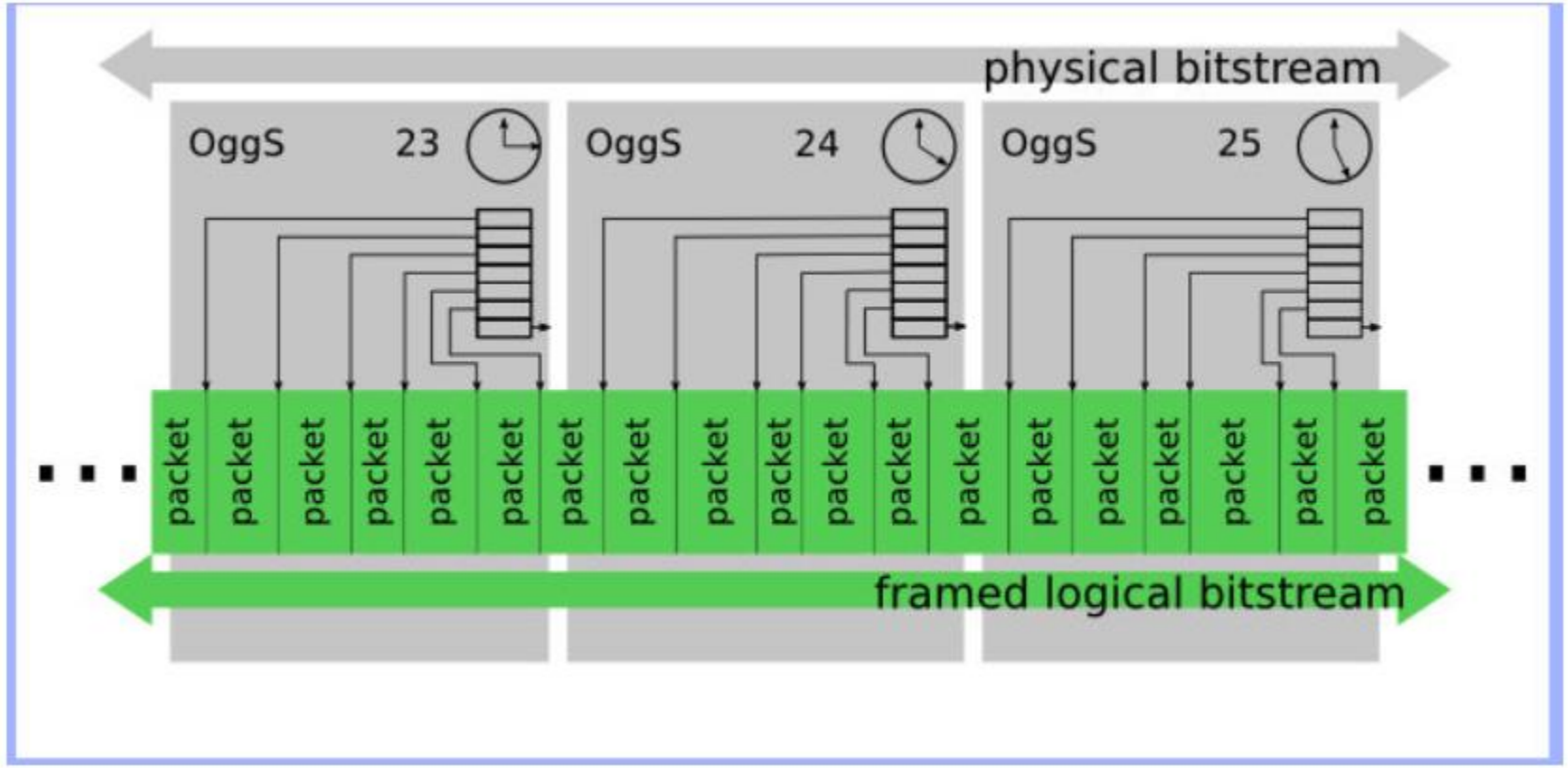
开源格式 | 内容丰富 | 多声道流媒体

- Ogg是一个自由且开放标准的容器格式，由Xiph.Org基金会所维护，Ogg格式并不受到软件专利的限制，并设计用于有效率地流媒体和处理高品质的数字多媒体。
- 可以纳入各式各样自由和开放源代码的编解码器，包含音效、视频、文字（像字幕）与元数据的处理
- OGGVobis(oggVorbis)是一种新的音频压缩格式，类似于MP3等的音乐格式。OggVobis是完全免费、开放和没有专利限制的。OggVorbis文件的扩展名是.OGG。Ogg文件格式可以不断地进行大小和音质的改良，而不影响旧有的编码器或播放器。OGG Vorbis有一个特点是支持多声道。

02

OGG文件格式剖析

Ogg文件格式由Ogg分装器和Ogg分离器组成，Ogg分装器用于将不同类型的数据流打包成一个文件，如音频、视频等。Ogg分离器用于从Ogg文件中提取单个数据流，并且可用于将不同类型的数据流复合在一起。这种方式与MPEG格式非常相似。Ogg文件以文件头开头，并且在文件头中包含元数据和分段表。



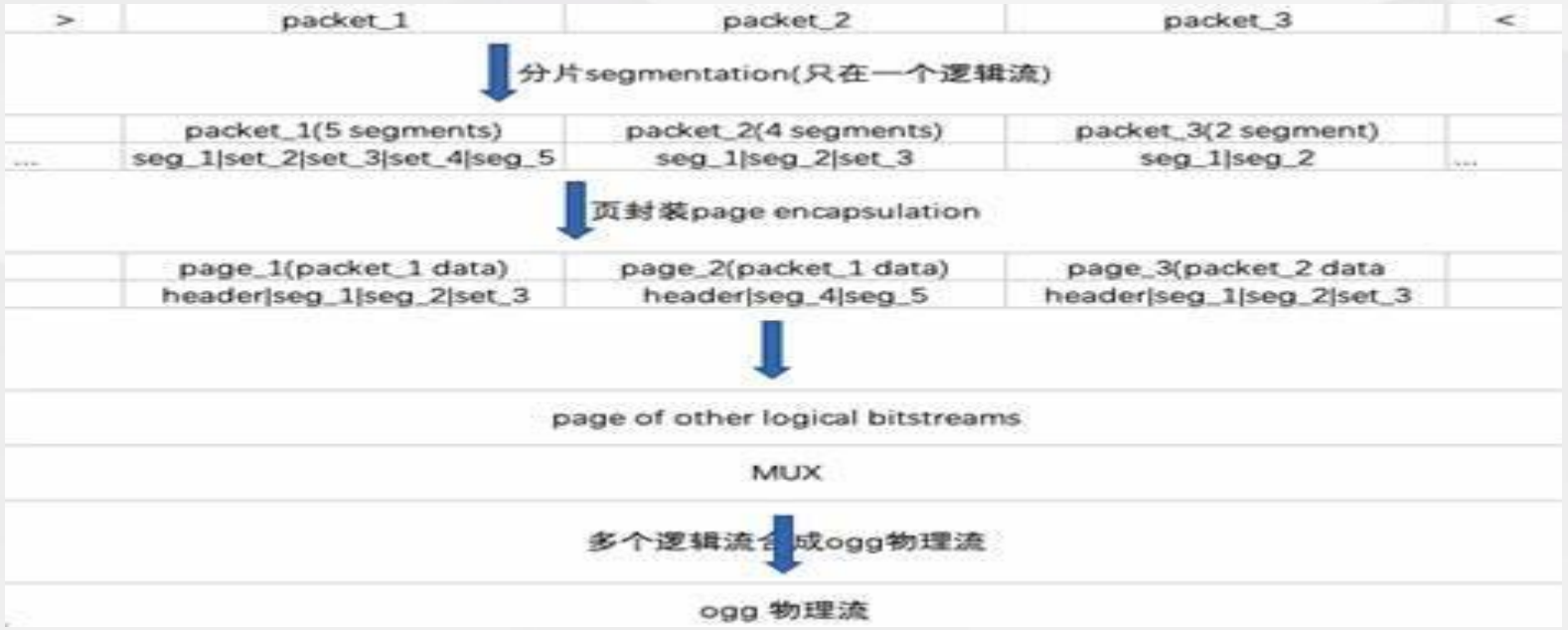
03

理论分析

文件头：Ogg文件的头码头**以字母"OggS"开头**，跟着一字节的**版本号**。接下来两个字节是数据流类型的**标识符**，然后是**标志**，标志位描述了流的具体性质，如数据的开头、中间或结尾。然后是一个8字节长的64bit全局唯一标识符，通常被称为magic，用于标识Ogg文件。接下来的四个字节代表Ogg文件的第几个数据段。

元数据：在Ogg文件头之后，是一些可选的**元数据信息**，这些信息通常包括作品的名称、作者和版权信息。

数据流：文件头和元数据信息之后，就是Ogg文件的**音频数据**。数据流被分成一个个数据块，称为**Ogg包**，每个包都包含一个数据段的完整数据，以及一个包头。包头中包含了一个长度字段，指示这个数据块包含的数据量。



04

编程实践

南開大學
Nankai University

页标识

信息隐藏示例音频.ogg

	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded Text
00000000	4F 67 67 53 00 02 00 00 00 00 00 00 00 00 AB 78	O g g S x
00000010	E5 4D 00 00 00 00 0D BB 9B 07 01 33 7F 46 4C 41	. M . . . . . 3 . F L A
00000020	43 01 00 00 01 66 4C 61 43 00 00 00 22 04 80 04	C . . . . f L a C . . . " . . .
00000030	80 00 00 00 00 09 15 03 E8 00 F0 00 00 00 00 00	. . . . .
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4F	. . . . . O
00000050	67 67 53 00 00 00 00 00 00 00 00 00 00 AB 78 E5	g g S . . . . . x .
00000060	4D 01 00 00 00 BC 75 57 A2 01 37 84 00 00 33 0D	M . . . . . u W . . 7 . . . 3 .
00000070	00 00 00 4C 61 76 66 35 38 2E 33 35 2E 31 30 30	. . . L a v f 5 8 . 3 5 . 1 0 0
00000080	01 00 00 00 1A 00 00 00 65 6E 63 6F 64 65 72 3D	. . . . . e n c o d e r =
00000090	4C 61 76 63 35 38 2E 36 32 2E 31 30 30 20 66 6C	L a v c 5 8 . 6 2 . 1 0 0 f l
000000A0	61 63 4F 67 67 53 00 00 00 3F 00 00 00 00 00 00	a c O g g S . . . ? . . . . .

版本号

信息隐藏示例音频.ogg

	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded Text
00000000	4F 67 67 53 00 02 00 00 00 00 00 00 00 00 AB 78	O g g S x
00000010	E5 4D 00 00 00 00 0D BB 9B 07 01 33 7F 46 4C 41	. M . . . . . 3 . F L A
00000020	43 01 00 00 01 66 4C 61 43 00 00 00 22 04 80 04	C . . . . f L a C . . . " . . .
00000030	80 00 00 00 00 09 15 03 E8 00 F0 00 00 00 00 00	. . . . .
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4F	. . . . . O
00000050	67 67 53 00 00 00 00 00 00 00 00 00 00 AB 78 E5	g g S . . . . . x .
00000060	4D 01 00 00 00 BC 75 57 A2 01 37 84 00 00 33 0D	M . . . . . u W . . 7 . . . 3 .
00000070	00 00 00 4C 61 76 66 35 38 2E 33 35 2E 31 30 30	. . . L a v f 5 8 . 3 5 . 1 0 0
00000080	01 00 00 00 1A 00 00 00 65 6E 63 6F 64 65 72 3D	. . . . . e n c o d e r =
00000090	4C 61 76 63 35 38 2E 36 32 2E 31 30 30 20 66 6C	L a v c 5 8 . 6 2 . 1 0 0 f l
000000A0	61 63 4F 67 67 53 00 00 00 3F 00 00 00 00 00 00	a c O g g S . . . ? . . . . .

南開大學
Nankai University

标识当前的页的类型

信息隐藏示例音频.ogg

	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded Text
00000000	4F 67 67 53 00 02 00 00 00 00 00 00 00 00 AB 78	O g g S x
00000010	E5 4D 00 00 00 00 0D BB 9B 07 01 33 7F 46 4C 41	. M . . . . . 3 . F L A
00000020	43 01 00 00 01 66 4C 61 43 00 00 00 22 04 80 04	C . . . . f L a C . . . " . . .
00000030	80 00 00 00 00 09 15 03 E8 00 F0 00 00 00 00 00	. . . . .
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4F	. . . . . O
00000050	67 67 53 00 00 00 00 00 00 00 00 00 00 AB 78 E5	g g S . . . . . x .
00000060	4D 01 00 00 00 BC 75 57 A2 01 37 84 00 00 33 0D	M . . . . . u W . . 7 . . . 3 .
00000070	00 00 00 4C 61 76 66 35 38 2E 33 35 2E 31 30 30	. . . L a v f 5 8 . 3 5 . 1 0 0
00000080	01 00 00 00 1A 00 00 00 65 6E 63 6F 64 65 72 3D	. . . . . e n c o d e r =
00000090	4C 61 76 63 35 38 2E 36 32 2E 31 30 30 20 66 6C	L a v c 5 8 . 6 2 . 1 0 0 f l
000000A0	61 63 4F 67 67 53 00 00 00 3F 00 00 00 00 00 00	a c O g g S . . . ? . . . . .

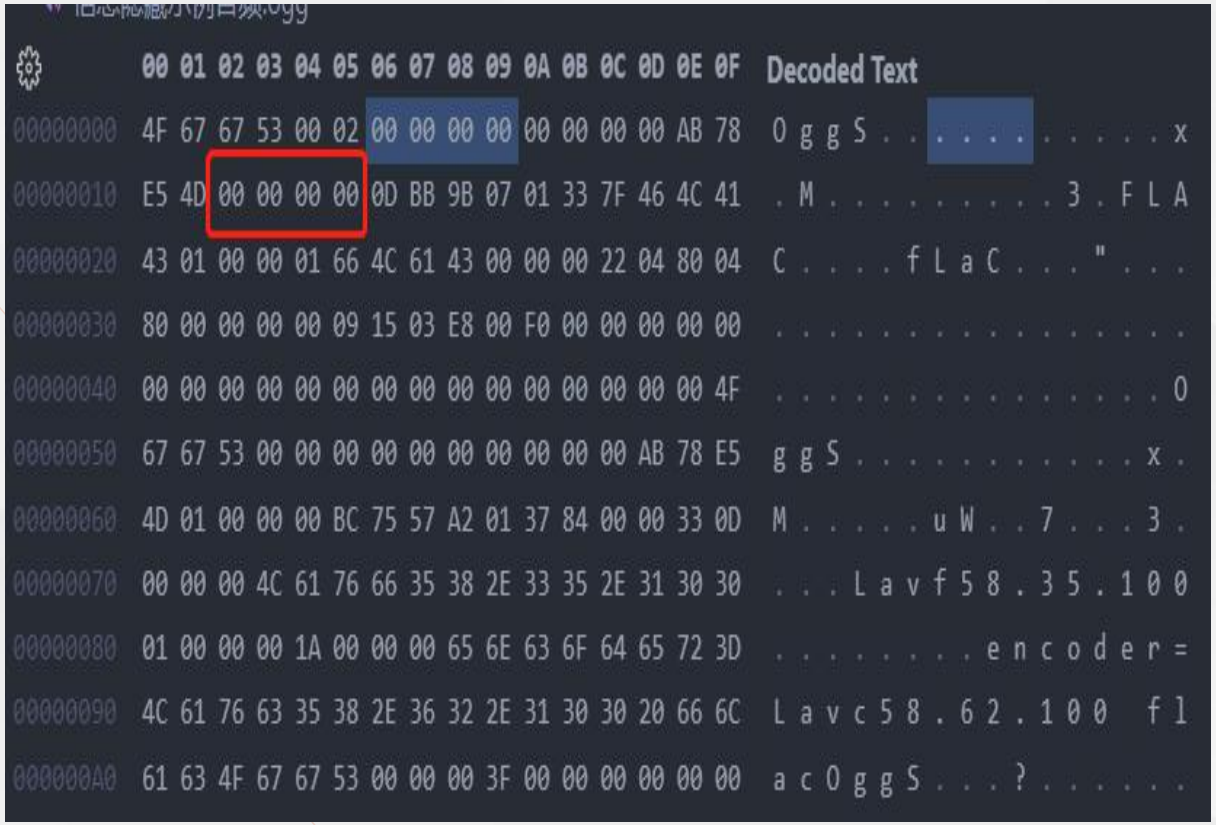
当前页中的流的 id

信息隐藏示例音频.ogg

	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded Text
00000000	4F 67 67 53 00 02 00 00 00 00 00 00 00 00 AB 78	O g g S x
00000010	E5 4D 00 00 00 00 0D BB 9B 07 01 33 7F 46 4C 41	. M . . . . . 3 . F L A
00000020	43 01 00 00 01 66 4C 61 43 00 00 00 22 04 80 04	C . . . . f L a C . . . " . . .
00000030	80 00 00 00 00 09 15 03 E8 00 F0 00 00 00 00 00	. . . . .
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4F	. . . . . O
00000050	67 67 53 00 00 00 00 00 00 00 00 00 00 AB 78 E5	g g S . . . . . x .
00000060	4D 01 00 00 00 BC 75 57 A2 01 37 84 00 00 33 0D	M . . . . . u W . . 7 . . . 3 .
00000070	00 00 00 4C 61 76 66 35 38 2E 33 35 2E 31 30 30	. . . L a v f 5 8 . 3 5 . 1 0 0
00000080	01 00 00 00 1A 00 00 00 65 6E 63 6F 64 65 72 3D	. . . . . e n c o d e r =
00000090	4C 61 76 63 35 38 2E 36 32 2E 31 30 30 20 66 6C	L a v c 5 8 . 6 2 . 1 0 0 f l
000000A0	61 63 4F 67 67 53 00 00 00 3F 00 00 00 00 00 00	a c O g g S . . . ? . . . . .
000000B0	AB 78 E5 4D 02 00 00 00 21 5A 09 87 34 0B 0B 0B	. x . M . . . . ! Z . . 4 . . .

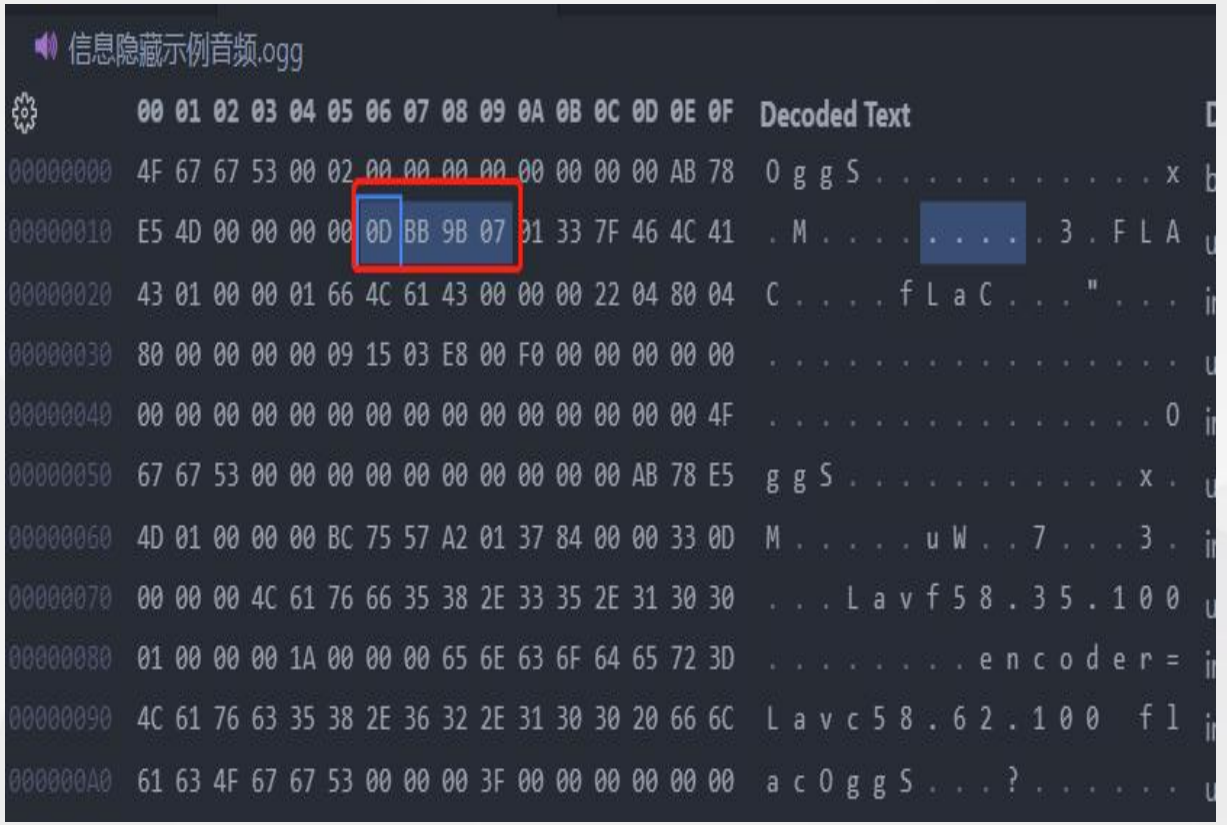
南開大學
Nankai University

本页在逻辑流的序号



	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded Text
00000000	4F 67 67 53 00 02 00 00 00 00 00 00 00 00 AB 78	O g g S x
00000010	E5 4D 00 00 00 00 0D BB 9B 07 01 33 7F 46 4C 41	. M . . . . . 3 . F L A
00000020	43 01 00 00 01 66 4C 61 43 00 00 00 22 04 80 04	C . . . . f L a C . . . " . . .
00000030	80 00 00 00 00 09 15 03 E8 00 F0 00 00 00 00 00	. . . . .
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4F	. . . . . O
00000050	67 67 53 00 00 00 00 00 00 00 00 00 00 AB 78 E5	g g S . . . . . x .
00000060	4D 01 00 00 00 BC 75 57 A2 01 37 84 00 00 33 0D	M . . . . . u W . . 7 . . . 3 .
00000070	00 00 00 4C 61 76 66 35 38 2E 33 35 2E 31 30 30	. . . L a v f 5 8 . 3 5 . 1 0 0
00000080	01 00 00 00 1A 00 00 00 65 6E 63 6F 64 65 72 3D	. . . . . e n c o d e r =
00000090	4C 61 76 63 35 38 2E 36 32 2E 31 30 30 20 66 6C	L a v c 5 8 . 6 2 . 1 0 0 f l
000000A0	61 63 4F 67 67 53 00 00 00 3F 00 00 00 00 00 00	a c O g g S . . . ? . . . . .

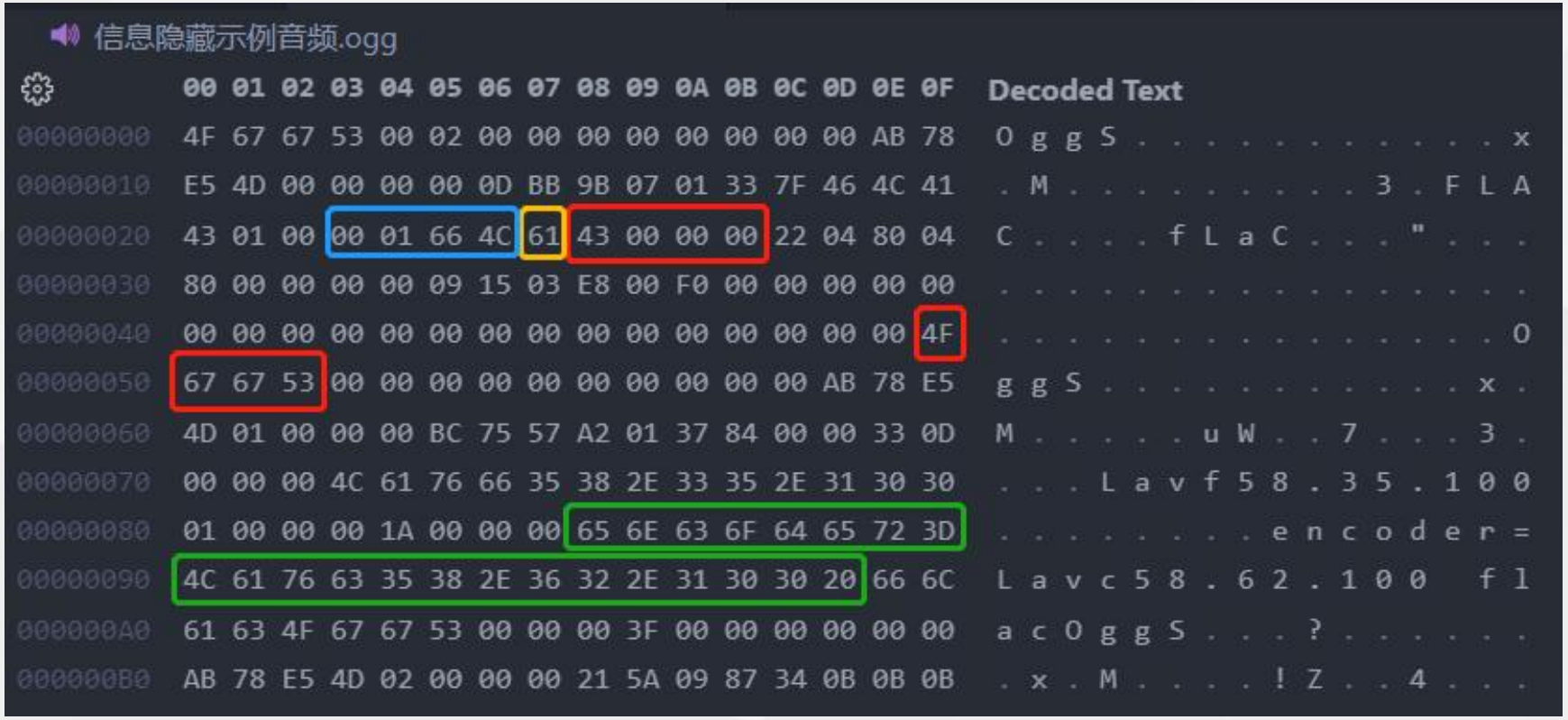
循环冗余校验码校验和



	00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F	Decoded Text
00000000	4F 67 67 53 00 02 00 00 00 00 00 00 00 00 AB 78	O g g S x
00000010	E5 4D 00 00 00 00 0D BB 9B 07 01 33 7F 46 4C 41	. M . . . . . 3 . F L A
00000020	43 01 00 00 01 66 4C 61 43 00 00 00 22 04 80 04	C . . . . f L a C . . . " . . .
00000030	80 00 00 00 00 09 15 03 E8 00 F0 00 00 00 00 00	. . . . .
00000040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 4F	. . . . . O
00000050	67 67 53 00 00 00 00 00 00 00 00 00 00 AB 78 E5	g g S . . . . . x .
00000060	4D 01 00 00 00 BC 75 57 A2 01 37 84 00 00 33 0D	M . . . . . u W . . 7 . . . 3 .
00000070	00 00 00 4C 61 76 66 35 38 2E 33 35 2E 31 30 30	. . . L a v f 5 8 . 3 5 . 1 0 0
00000080	01 00 00 00 1A 00 00 00 65 6E 63 6F 64 65 72 3D	. . . . . e n c o d e r =
00000090	4C 61 76 63 35 38 2E 36 32 2E 31 30 30 20 66 6C	L a v c 5 8 . 6 2 . 1 0 0 f l
000000A0	61 63 4F 67 67 53 00 00 00 3F 00 00 00 00 00 00	a c O g g S . . . ? . . . . .

南開大學
Nankai University

0x23-0x26 代表的是版本号，0x27 的位置代表通道数，后面紧跟着的四个比特是采样率
后面很大的一块是音频的注释包



很明显，ogg 文件的格式是以页为单位的，所以页首部我们肯定不能动，前面三个页通常都比较小，不便于我们隐藏数据，我们可以在第三个以后的页数中隐藏数据，那个时候每一个页的大小都会有接近 5000 个字节可以进行隐藏。于是我们根据前面给出的公式

$$header\_size = 27 + Num\_segments\ (byte)$$

基于流媒体的 LSB（最低有效位）隐藏法是一种数字水印技术，用于保护数字流的版权和完整性。该方法基于在音频、视频或图像流中嵌入水印信息，并用 LSB 算法将信息隐藏在最低有效位中

南開大學 Nankai University

完整代码及测试用例详见：
https://github.com/wbf1015/Information_hiding/tree/main/

```
'''
将hide_information隐藏到origin_data以index2hide开始的位置
'''
def HideInformation(origin_data,index2hide,hide_information):
    count = 0
    od = origin_data
    for char in hide_information:
        # print('end = ',char)
        # print('origin_data = ',od[index2hide+count])
        # if char == '1':
        #     print('2Replace = ',CalReplaceChar(od[index2hide+count],True))
        # if char == '0':
        #     print('2Replace = ',CalReplaceChar(od[index2hide+count],False))
        if char =='1':
            od = replaceHex(od,index2hide+count,CalReplaceChar(od[index2hide+count],True))
            pass
        if char =='0':
            od = replaceHex(od,index2hide+count,CalReplaceChar(od[index2hide+count],False))
        count += 1
    return od

'''
```

南開大學
Nankai University

完整代码及测试用例详见：
https://github.com/wbf1015/Information_hiding/tree/main/

```
'''
从data的begin位置按lsb方法提取l比特的信息
'''
def PatchInformation(data,begin,l):
    infor = ""
    zero = ['0','2','4','6','8','a','c','e']
    one = ['1','3','5','7','9','b','d','f']
    for i in range(0,l):
        if data[begin+i] in zero:
            infor = infor + '0'
        if data[begin+i] in one:
            infor = infor + '1'

    char_length = l//8
    ans = ""
    for i in range(0,char_length):
        temp = infor[i*8:(i+1)*8]
        num = int(temp,2)
        ch = chr(num)
        ans = ans + ch

    return ans
```

完整代码及测试用例详见：
https://github.com/wbf1015/Information_hiding/tree/main/

```
整个隐藏以及提取信息的pipeline
'''
def pipline():
    binary_data = None
    hex_data = None
    with open('信息隐藏示例音频.ogg', 'rb') as f:
        binary_data = f.read()
    hex_data = binary_data.hex()
    print('[PIPELINE LOG] SUCCESSFULLY READ OGG FILE')
    print('[PIPELINE LOG] DISPLAY HEAD OF THE OGG FILE:',hex_data[:20])
    index = FindStrIndex(hex_data)
    index = index+27+255
    print('[PIPELINE LOG] DISPLAY HIDING INDEX:',index)
    # 在这里更改你想要隐藏的信息，因为没有encode和decode所以只能隐藏英文、数字和一些特殊字符
    hide_infor = GenInformation2Hide('NKU')
    print('[PIPELINE LOG] INFORMATION 2 HIDE IS: ',hide_infor)
    hex_data2 = HideInformation(hex_data,index,hide_infor)
    byte = bytes.fromhex(hex_data2)
    # 打开文件，并以二进制写入模式写入二进制数据
    with open('隐藏后音频.ogg', 'wb') as f:
        f.write(byte)
    print('[PIPELINE LOG] SUCCESSFULLY DONE INFORMATION HIDING!')
    binary_data2 = None
    with open('隐藏后音频.ogg', 'rb') as f:
        binary_data2 = f.read()
    hex_data2 = binary_data2.hex()
    print('[PIPELINE LOG] START PATCHING INFORMATION FROM OGG FILE')
    print('PIPELINE LOG] SUCCESSFULLY DONE INFORMATION PATCHING! INFORMATION IS :',PatchInformation(hex_data2,index,len(hide_infor)))
```

南開大學
Nankai University

完整代码及测试用例详见：
https://github.com/wbf1015/Information_hiding/tree/main/

```
PS G:\code\information_hiding\Information_hiding\第二次大作业ogg> python ih.py
[PIPELINE LOG] SUCCESSFULLY READ OGG FILE
[PIPELINE LOG] DISPLAY HEAD OF THE OGG FILE: 4f676753000200000000
[PIPELINE LOG] DISPLAY HIDING INDEX: 22590
[PIPELINE LOG] INFORMATION 2 HIDE IS:  010011100100101101010101
[PIPELINE LOG] SUCCESSFULLY DONE INFORMATION HIDING!
[PIPELINE LOG] START PATCHING INFORMATION FROM OGG FILE
PIPELINE LOG] SUCCESSFULLY DONE INFORMATION PATCHING! INFORMATION IS : NKU
PS G:\code\information_hiding\Information_hiding\第二次大作业ogg>
```

谢谢大家