《网络空间安全系统设计》
课程报告

基于文本的信息隐藏设计 实现

小组成员： 张立飞 余天航 李冲 李润锦 刘皓予

学号：

班级：

日期： 2024 年 6 月 14 日

西北工业大学网络空间安全学院

目录

一、研究简介

本次实验，以文本文件作为载体，对其进行隐藏信息嵌入与提取。可以根据文本信息隐藏常用改变字符缩放、字体颜色、字间距、行间距等手段，设计并实现了一种文本隐藏算法。

在设计并实现的过程中，我们使用了多种方法，如 docx 文件隐写的常规方法与改变字体 RGB 颜色、大小等，通过这些信息，合理推断出在 word 文档中隐藏的信息。

具体而言，由于 docx 文件的本质是一个压缩包，我们基于此构造出了一个冗余 xml 文件，藏入部分内容；基于 16 进制文件内容，以 ANSI ASCII 形式同样藏入部分内容；最后，在于基于文本藏进的信息对比拼接，得到最终隐藏的内容。

二、研究内容

2.1 研究方案

我们将明文转换成十六进制的格式隐藏在文档中。第一步，得先找到原文中字符大小被改变了的那些字符，就能得到一个 List 集合，里面的元素实质上为全体十六进制字符 1~F，只不过顺序是他们在文中出现的顺序；第二步，我们需要找到一个取字顺序，为了增加难度，我们用了三种方法将取字顺序分成 Begin、Middle、End 三块分别隐藏。破译者得到三块部分并拼接后，就能得到完整的取字顺序。最后一步，根据刚刚得到的取字顺序，从 List 一个一个取出十六进制字符并排列，就能得到被隐藏信息的十六进制文本。

2.2 相关方法

Word 文件隐写具有多种方法，以下是我们用到的部分相关方法。我们将逐一对这些方法进行介绍，在第三部分中更详细地介绍我们是如何利用这些方法进行信息隐藏的。

2.2.1 Word 文件本质

Docx 文件是 Microsoft Word 使用的一种文档格式，属于 Office Open XML（OOXML）文件格式标准。它本质上是一种文本文件，但其结构比简单的纯文本文件更复杂，包含了丰富的格式信息和各种嵌入对象（如图片、表格等）。

一个 DOCX 文件实际上是一个压缩包（ZIP 文件），其中包含了一系列 XML 文件和其他资源文件。可以通过将一个 DOCX 文件的扩展名改为“.zip”，然后使用解压缩软件打开这个文件，来查看其内部结构。

解压后的 DOCX 文件通常包含以下几个重要的文件和目录：

a. [Content_Types].xml：描述文档中所有内容类型的信息。

b. _rels 目录：包含关系文件（.rels），定义了文件之间的关系。

c. docProps 目录：包含文档属性文件（如 core.xml 和 app.xml），描述文档的基本信息（如标题、作者、修改日期等）。

d. word 目录：这是主要内容所在的目录，包含了多个 xml 文件。在 document.xml 里存放了文档的主要内容，包含了文本文字和基本的格式信息；在 styles.xml 中定义文档的样式；在 settings.xml 中保存了文档的配置信息；在 theme 目录中包含了主题信息，定义文档的整体视觉风格；在 media 目录中包含文档中嵌入的媒体文件。

通过在修改这些 xml 文件，可以达到信息隐藏的目的。

2.2.2 图片隐写术

图片隐写术具有多种方法，比如可以通过修改图片中文字的颜色来实现隐藏的目的。还可以通过改变图像中像素的最低有效位（LSB），调整图像的颜色分

量，或利用频域方法如离散傅里叶变换（DFT）和离散余弦变换（DCT）等技术，将隐秘信息嵌入到图像的视觉内容中，这些方法使得在不明显改变图像外观的情况下实现信息隐藏。

最简单的方法是将文字颜色变为和背景一样，再将图片位置隐藏在 word 文档中，从而实现图片隐写

2.2.3 文本信息隐藏

利用人类的视觉系统（HVS）对文档结构进行微调二使人眼无法察觉，从而嵌入隐藏信息。

常用基于文本的信息隐藏方法包括如下：

A. 通过放缩字符比例在文档中隐藏信息

在文本文档中，轻微缩放字符大小的比例，人的肉眼是不易察觉的。实现信息隐藏的方法是改变文本中字符大小的横向缩放比例，在一段字符在文本文档中字符的缩放比例通常是标准形，即 100%，对需要嵌入秘密信息的字符，可以采用缩放的比例分别设为 101%、102%、103%和 104%来进行编码，从而使得每个载体文本的字符可实现 2 位二进制码的隐藏而不易被发觉。假设待隐藏的信息为比特流，选择两种接近的字号(如 13 和 13.5),规定字号“13”代表“0”,“13.5”代表“1”,逐个读入载体文本中的字符(汉字或英文均可)，根据当前所要隐藏的比特为“1”或“0”，把字符的字号改为“13.5”或“13”。这种方法的隐藏效果比较好，英文字符的隐藏效果更好，隐藏信息后文本文件的大小没有改变。

B. 通过置换字符 RGB 颜色值在文档中隐藏信息

如果同时分别轻微改变字符 RGB 颜色值中的 R、G、B 值，那么既可提高信息隐藏量，又不易被发觉。该方法是对需要嵌入秘密信息的载体文本字符，同时置换字符 RGB 颜色值中 R、G、B 低 4 位的值，从而每个载体文本的字符可实现 12 位二进制数的隐藏，隐藏后文档和原文档在视觉上的差别很小。

C. 通过调整字间距与行间距进行信息隐藏

行移编码法是通过将文本的某一整行垂直移动来嵌入水印。通常，当一行被上移或下移时，与其相邻的两行或其中的一行保持不动，作为解码过程的参考位置。例如，在移动过的一行中编码 1bit 信息，如果这一行上移，则编码为“1”；如果这一行下移，则编码为“0”。根据经验，当垂直位移量等于或小于 1/300 英寸(1 英寸≈2.54 厘米)时，人眼将无法辨认。

字移编码法是在编码过程中将文本某行中的一个单词水平左移或右移来嵌入水印信息，而与其相邻的单词并不移动。这些不移动的单词作为解码过程的参考位置。根据经验人眼无法辨认 1/150 英寸以内的单词的水平位移量。

2.3 构造过程与解题思路

2.3.1 隐藏过程

A. 文本字号与颜色

原文档的标题字体大小为三号，副标题的字体大小为小四，正文的字体大小为五号，我们参照 list= [2 d 3 8 A f 6 9 b e 4 c 0 7 5 1]，选择正文中的对应的字符，将其字体大小更改为 11 号。

依照 Middle：10 14 3 7 4 7 15 7 10 15 6 14 16 7 8 7 的顺序在文本中任选字符，并在字体颜色中将蓝色一栏的数值从 0 更改为上述的数值。

图 1 RGB 的修改

B. XML 文件的添加

依据前文中对 docx 文件的介绍，我们将 docx 文件的后缀更改为.zip，打开压缩包后添加一个名为 something.xml 的文件，文本里面隐藏了有效信息 Begin 且。将报告的后缀重新更改为 docx 后，报告本身没有变动。

C. 图片隐藏

我们在标题《2.1Hash 函数的研究》这一行中的 Hash 后面添加了一张隐藏信息的图片，然后将让后续的字符覆盖该图片。图片的内容为 Tip:Middle in RGB 用于告知解题者 Middle 信息使用了 RGB 即字体的颜色来隐藏。

D. 16 进制文件隐藏

使用 16 进制编辑器打开文档，在不影响文件内容的情况下，将其中一段全 0 的数值更改为 End:11 7 6 7 10 7 14 14 2 中的数值。

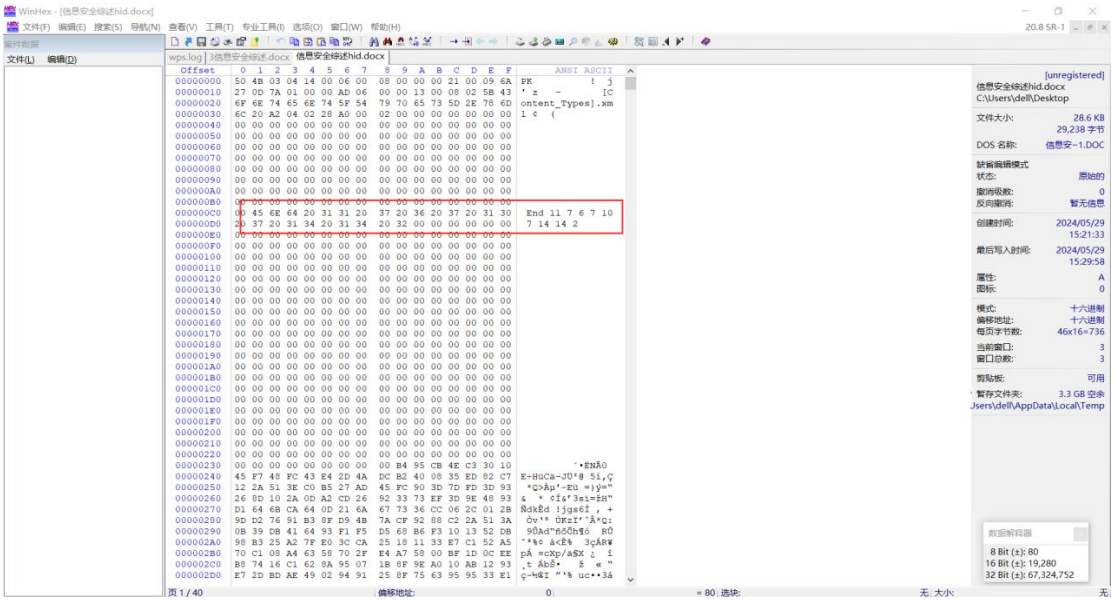

图 2 十六进制隐藏

2.3.2 解题过程

首先，隐藏的明文形式以 flag{}的形式给出

解题步骤：

①从文档里找到大小不一样的字符，形成列表 list=[2 d 3 8 A f 6 9 b e 4 c 0 7 5 1]，得到这些字符的顺序 1~16，对应十六进制中的字符 1~16，具体方法为：

a) 查找出文档中所有大小和所在部分字体不一样的字符（或编写能够识别和比对字符大小的 python 代码）

b) 最后将修改过大小的字符按照顺序输出即可得到 list。

②修改文件后缀为. zip，如图 3.

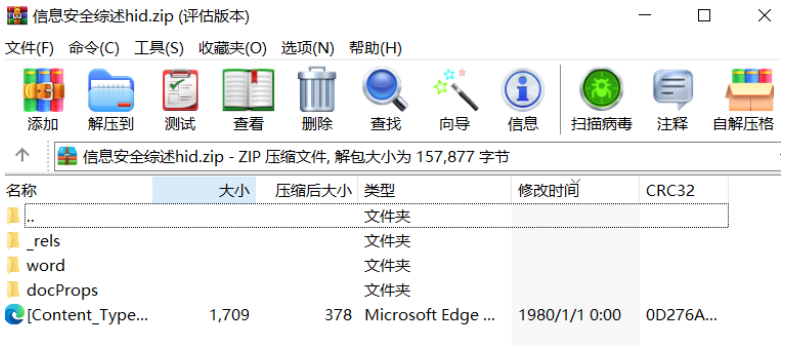

图 3 压缩包目录

找到以下文件并打开，如图 4.

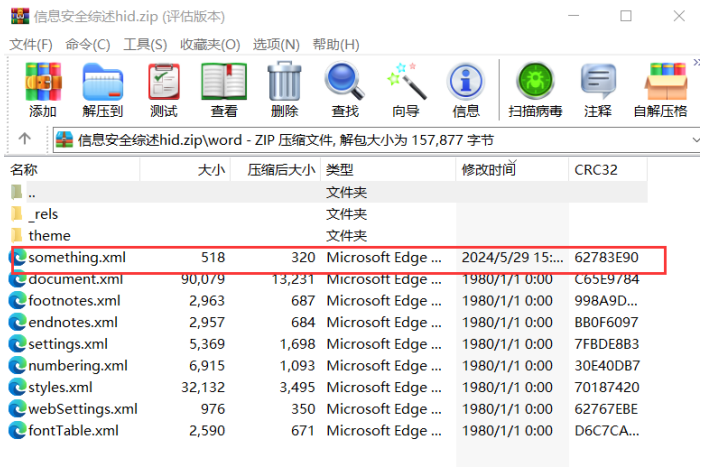

图 4 隐藏信息的文件

得到以下内容，见图 5：

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title>Blank::Error</title>
    <script type="text/javascript">

        var hiddenFlag = "Begin: 7 7 7 12 7 16 7 14 14 9 14 8 14 15 7 16 7";

    </script>
</head>
<body>
    <!-- The flag is hidden in the script above and nothing is output on the screen -->
</body>
</html>
```

图 5 隐藏的信息

有效信息为 Begin: 7 7 7 12 7 16 7 14 14 9 14 8 14 15 7 16 7

③得到 Middle 的过程（以下是提示，没找到无影响，降低难度用的）：

首先通过换行等方法找到被隐藏在标题 2.1 文字里的图片，通过调整图片的对比度、亮度和饱和度等得到图片上被隐藏的信息：Tip:Middle in RGB。（也可以 zip 格式中从 media 文件夹中找到该图片），图片在文档中隐藏的地方见图 6：

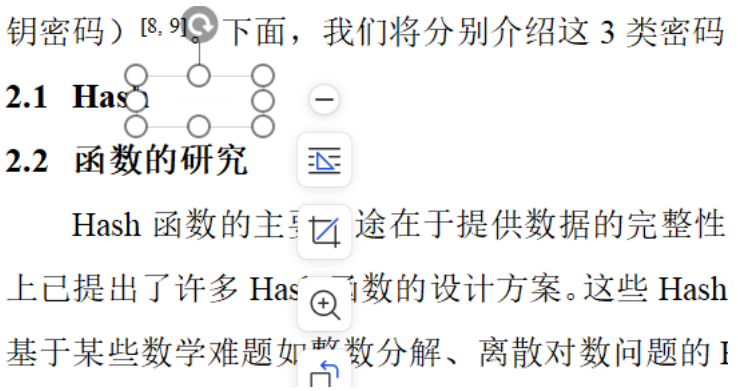

图 6 图片隐藏位置方法①

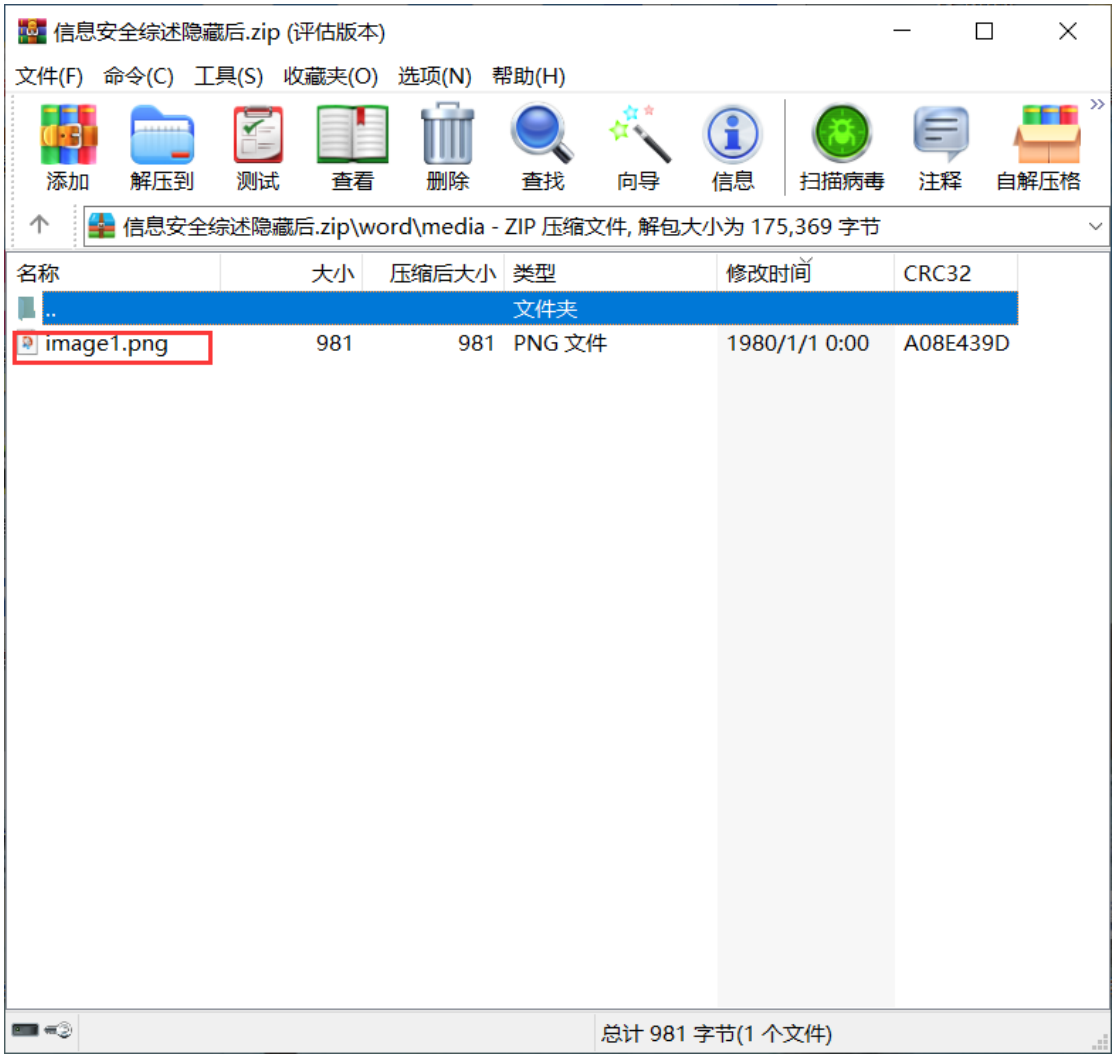

图 7 图片隐藏位置方法②

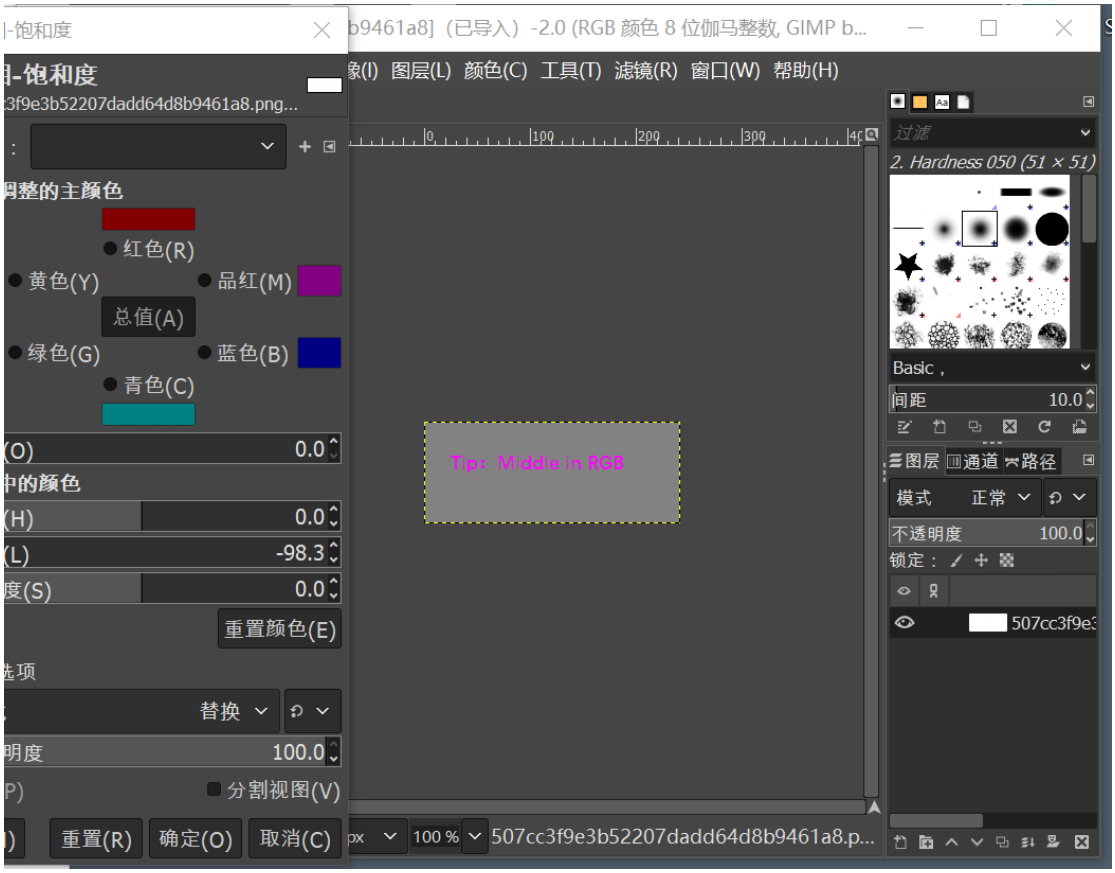

图 8a 调整 RGB，得到图片文字

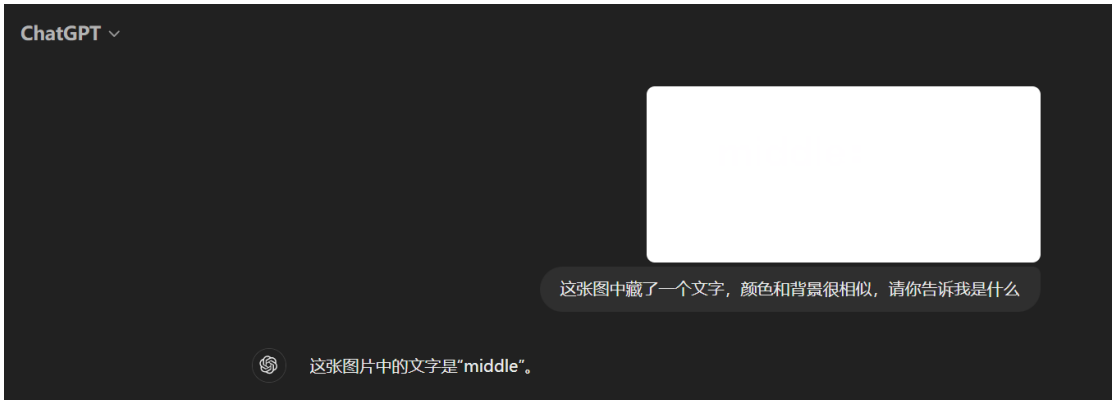

图 8b ChatGPT 得到内容

然后，Word 查找出所有黑色（RGB：255,255,255）字符删掉（也可编写能够识别和对比字符 RGB 值的 python 代码），得到被修改过 RGB 值的字符

最后，将被修改过的字符的 RGB 值的低八位输出，按顺序即得到 Middle:10 14 3 7 4 7 15 7 10 15 6 14 16 7 8 7

④以十六进制打开文件，得到以下内容：

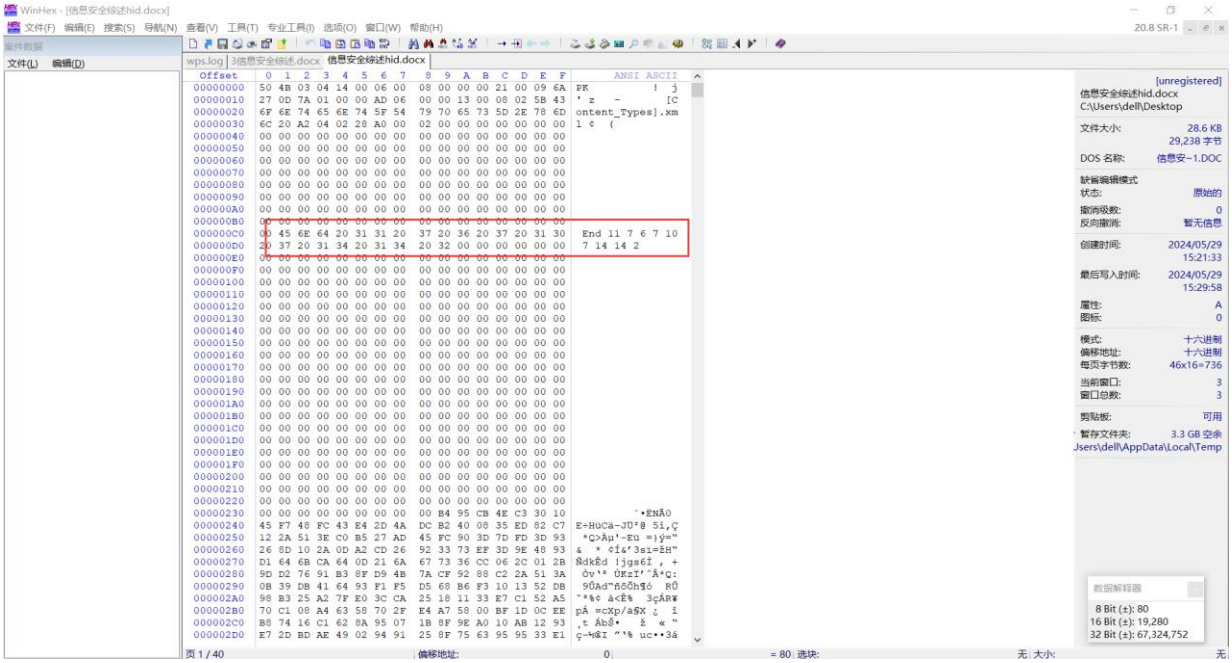

图 9 16 进制文件隐藏信息

得到：End:11 7 6 7 10 7 14 14 2

⑤在这里，把②③④中得到的内容按顺序拼在一起：

7 7 7 12 7 16 7 14 14 9 14 8 14 15 7 16 7 10 14 3 7 4 7 15 7 10 15 6 14 16 7 8 7 11 7 6 7 10 7 14 14 2

然后，根据①中得到的 list，按照元素顺序取出元素，得到：

66 6C 61 67 7B 79 75 61 6E 73 68 65 6E 5F 71 69 64 6F 6E 67 7D 就是明文中每个字符 ASCII 码的十六进制表示。所以二进制：01100110 01101100 01100001 01100111 01111011 01111001 01110101 01100001 01101110 01110011 01101000 01100101 01101110 01011111 01110001 01101001 01100100 01101111 01101110 01100111 01111101

检查方法：转成字符，有 flag 就是

明文：flag{yuanshen_qidong}

三、成员分工及个人课程目标达成情况

成员分工情况如下：

学号	负责部分
	项目整体构思，16 进制与 xml 隐藏，参与破解第 9 组，第 2 组
	图片隐藏与文本信息隐藏，参与破解第 9 组，第 2 组
	理论方法研究与选取，参与破解第 9 组，第 2 组
	理论方法研究与选取，参与破解第 9 组，尝试破解第 5 组
	初期规划，提出建议