**南京航空航天大学**

**信息安全综合实验报告**



**课 题 名 称 Office XML 文档信息隐藏方法**

**姓 名 廖俊轩 学 号 161820127**

**专 业 信息安全 班 级 1618204**

**同 组 成 员 161820218胡淼 161820219李云青**

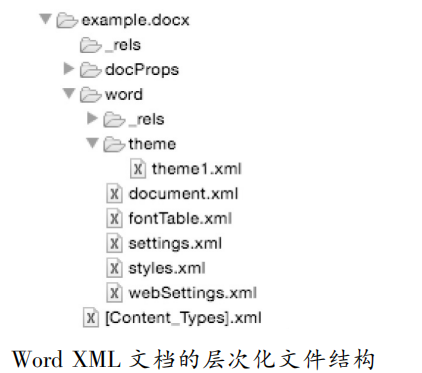
|  |  |
| --- | --- |
| **总 成 绩** |  |
| **指导老师** | **许 峰** |

一、个人工作

本次的项目中，我负责的部分是资料搜集与论文梳理。

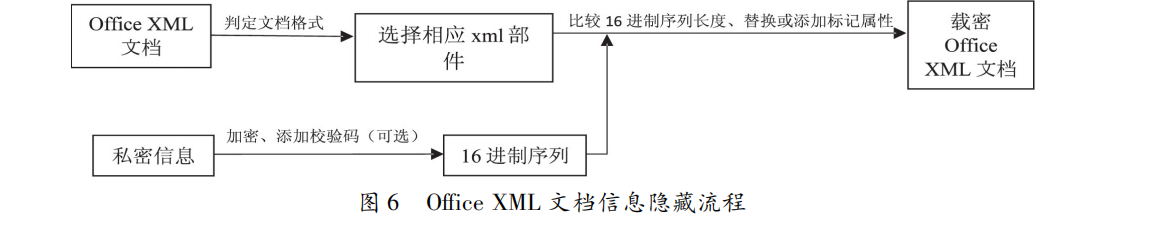
1. 在本次项目展开前，我先从bilibili上，查询了对应的word加密（ctf -Misc）视频，以及从csdn上找寻相关的算法实现。
2. 经过发现，目前的主流信息隐藏手段，主要是通过修改 Word 中文本的字符大小或文本颜色来隐藏信息。这些方法主要是针对 Word 文档提出的。在《基于word文本文档的信息隐藏方法研究》一文中，作者对常用的隐藏方法进行了说明，例如：格式文本文件、替换隐藏方法、基于word文本缩放、本身的隐藏文字功能、基于文本框和图片框等方式。这些方式都是利用视觉隐藏的方式，人工操作，具有很大的局限性。
3. 在经过筛选后，找到了一种基于 Office Open XML 的文档格式的隐藏方法，因其可以通过多种方式访问、降低文件损坏的风险等特点，已被越来越多的用户使用。
4. 之后，对该论文进行了梳理，如下：

1）Office Open XML 文件格式下的创建、编辑和保存均符合要求。



上图为Word XML 文档的层次化文件结构，不同的xml文件对应了不同的word 属性。其中我们需要注意的是document.xml 文件，这是文件包的主文档，其记载了 Word XML 文件的文字内容及其他相关属性;

1. 基于 XML 的 Office Open XML 中，最基本的单位是元素，元素可以带有若干个属性及属性值作为附加信息。无论是． docx、． pptx 还是． xlsx 格式的文件，元素的命名及其属性值的定义都应遵循 Open XML 规范，同时每个元素也有其特定的意义及作用。Ps：我们正是基于这种属性进行加密解密，“标识属性”一般用于区分文本、表格等数据或属性，其特点是拥有独一无二且由系统随机生成的属性值，并且该属性值与用户及修改时间等无任何关系。
2. 对于标识属性值的修改不会对文本内容造成影响。因此通过将待隐藏信息写入其属性值中，可以实现在 Office XML 文档中隐藏信息的目的。
3. 具体解决方案：将待隐藏信息转化为十六进制的 Unicode 码，然后选择加密算法将其进行加密，并添加校验位以便于之后的隐藏信息提取工作。
4. 对于 Word XML 中的 w: rsidR 属性而言，其出现的次数及频率较多，因此上述基于替换原有标识属性属性值的隐藏方案在 Word XML 中较易实现。而对于Excel XML 文档中不存在标识属性 x14: id，同样可通过在 slide． xml 部件中对其进行添加，从而实现信息隐藏的目的。
5. 整体流程



## 二、源代码

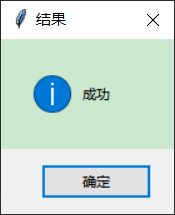
看附件

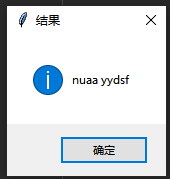
## 执行过程与结果

**word加解密**

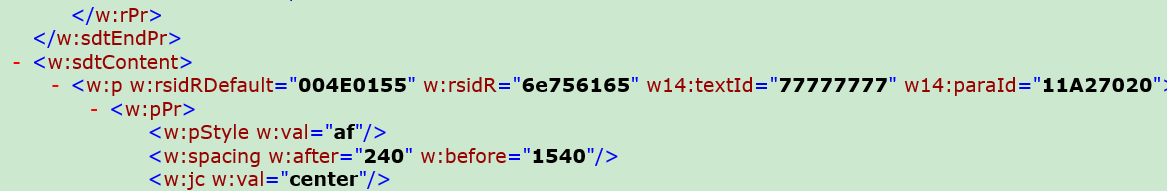
加密前xml信息：



加密中：

解密：

注，这里nuaa后面的乱码是因为水印没有特殊的结束符，故提取时按定长提取，所以出现了nuaa后面的乱码。

加密后：

算法分析

实验环境是 Windows 10，pycharm。所使用的实验文件是从自己制作的一些 Office XML 文档。下面从鲁棒性、信息隐藏容量和隐蔽性方面进行分析。

( 1) 鲁棒性: 在各式文档中，隐秘信息嵌入到标记属性值中。经试验测试，在 Office 系统软件中，对加密文档进行常用的各种格式设置和内容的添加删减均不会造成隐秘信息的遗失，因此该算法鲁棒性较强。但是若将含有隐密信息的 Word 文本段落整段删除，或将含有隐秘信息的 Excel 整页表格完全删除，隐秘信息均会丢失。

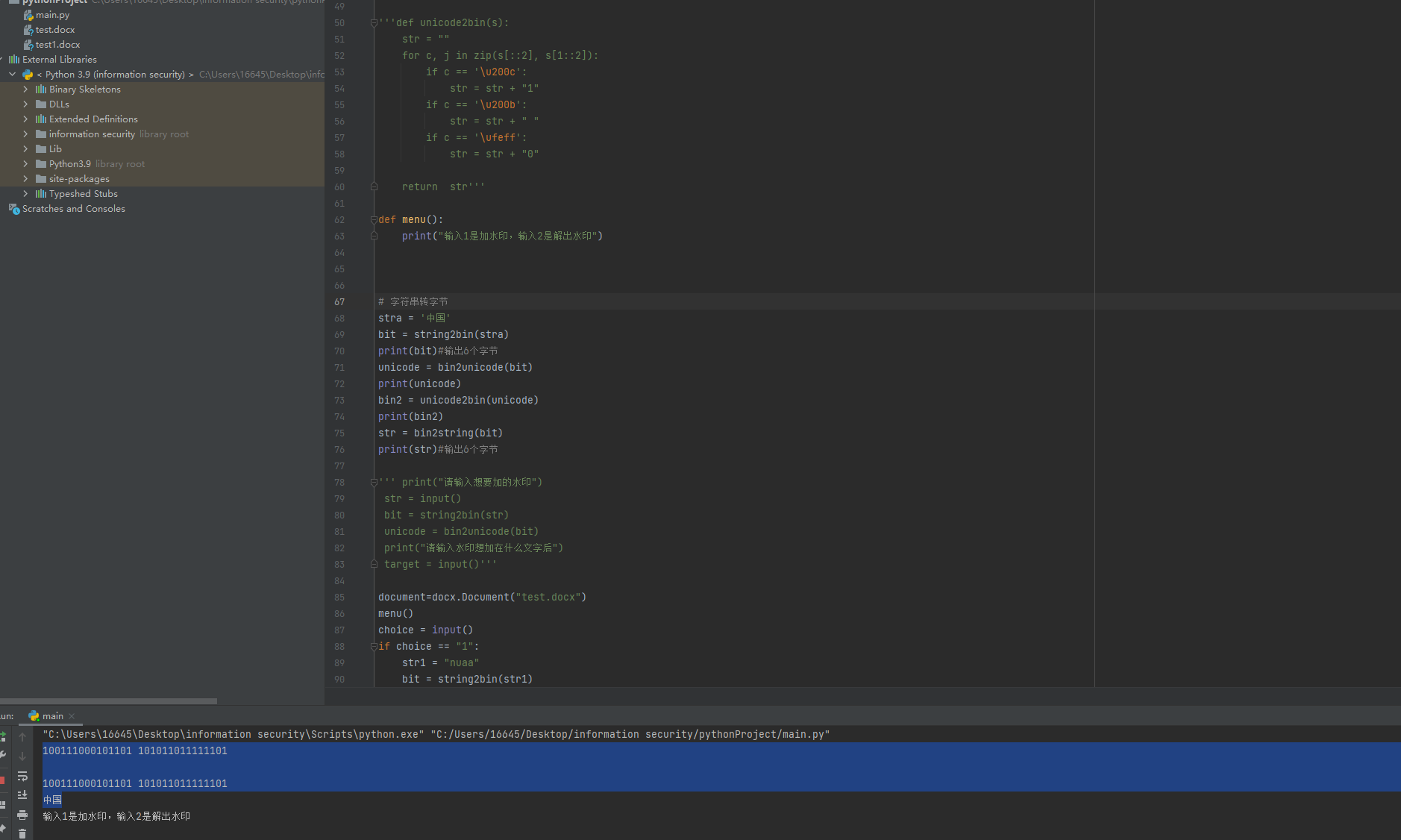
( 2) 信息隐藏容量: 针对 PowerPoint XML 及 Excel XML 格式文档，一个标记属性值允许隐藏长度为 16字节即 128 比特的加密信息。而对于 Word XML 文 档，一个标记属性值允许隐藏长度为 4 字节即 32 比特的加密信息。而对于一个大小 3 MB 左右的． docx 格

式文档，其存在着大约 300 个 w: rsidR 属性，即约 1 200字节。由此可见，该算法拥有较大的信息隐藏容量。

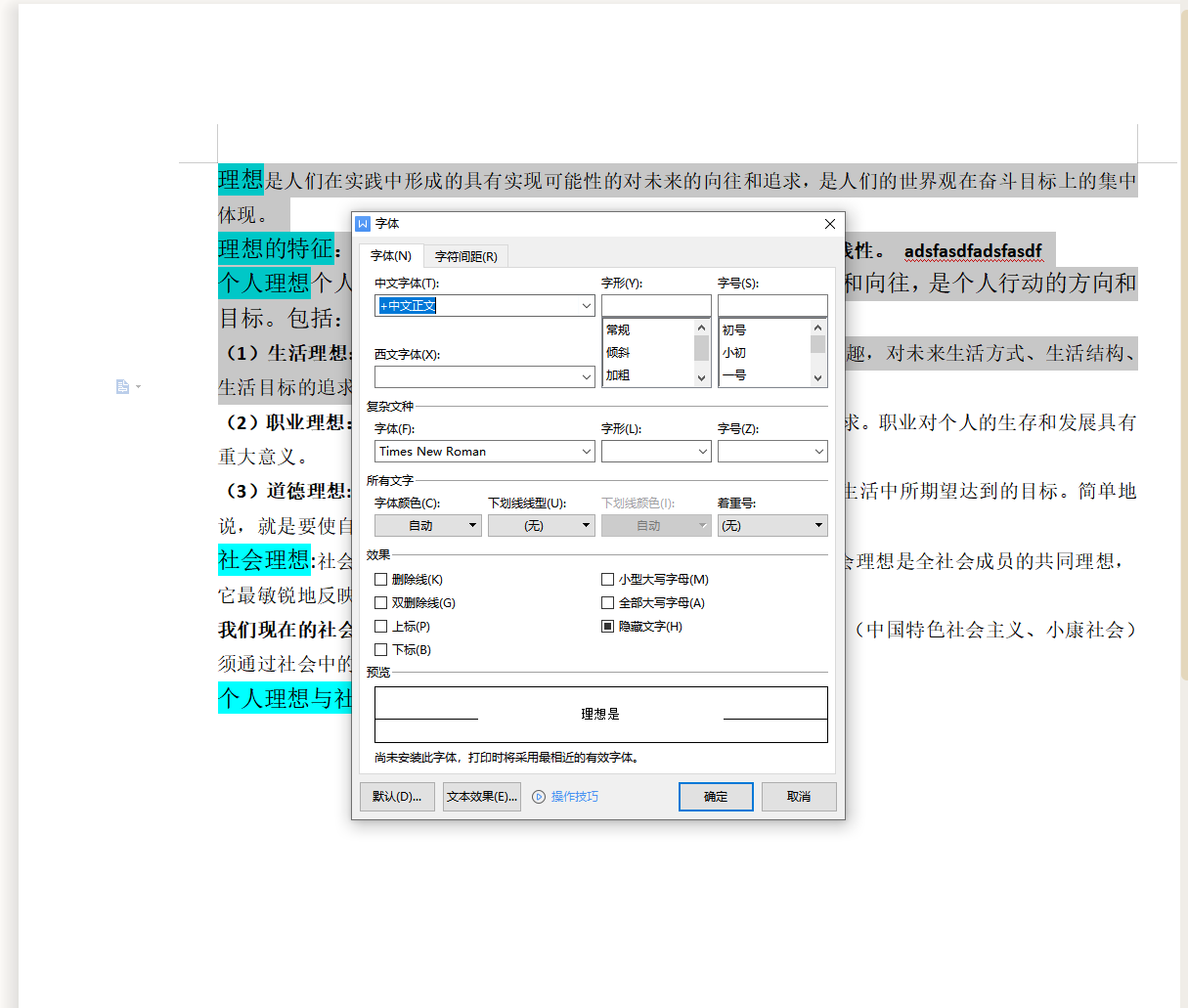
( 3) 隐蔽性: 通过实验证明，在 Office XML 文档中应用该算法，不会引起文档显示的任何改变，也不会影响文档的正常使用。若只采用算法中的替换属性值方案，则对文本大小不会产生任何改变，因此算法的隐蔽性较好。

# 设计过程

一开始，是想实现类似于js里面用零宽字符作为信息载体，用三个零宽字符进行对 0,1进行编码，如下图



结果发现在word当中没有对零宽字符的支持。所有js0宽字符会以空格显示。只能用word的隐藏实现隐写。效果很差，也容易发现。



故改变策略。联想到ctf-misc分支里面有对word解压后进行操作的题目。所以就上网确定这个操作的学名，发现他们是使用了docx的新特性即基于xml的结构，就上cnki找了相关论文。最后找到了那篇。进行复现。因为对于知识了解不够，一开始是以构建水印主（正确的方式是以修改原文件的水印位），一开始做的docx，pptx和xlx的相当于全部错误。因为时间来不及就对docx进行了重写。

初版xml隐写代码（添加表示位构造水印如下）

Watermark = "<a: tblStylestyleId =\"{" + "\"" + watermark + "}\"" + "styleName =\"" + watermark + "\"/>"  
strs = string.split("</a:tblStyleLst>");  
text = strs[0] + Watermark + "</a:tblStyleLst>" + strs[1]

Watermark = "< x14 : id> {" + watermark + "}" + "</ x14:id>"

strs = string.split("<extLst>");  
text = strs[0] + "<extLst>" + Watermark + strs[1]

可以看出这是直接添加，这种方法有很大的问题，最直接的xml格式校验都过不了。

最后改成了提交版的做法（提交版只实现了docx的隐写）