面向特征的信息隐藏检测研究①

Study on Stegnodetect Face to Trait

陈 伟　孙 勇　杨义先　(北京邮电大学信息安全中心 北京 100876)
钮心忻　(北京邮电大学数字内容研究中心 北京 100876)

摘要:本文研究了图像信息隐藏中针对结构隐藏和对软件特征码的特征识别技术,根据特征的多段、多选项性质提出特征分层思想,在此基础上建立了特征识别算法。它用特征搜索引擎解析特征数据库,扩大了能处理的特征范围,简便快速高效,避免了为每个复杂特征分别编写代码造成的问题。

关键词:信息隐藏检测 特征 多段 多选项

1 前言

保护通信信息主要应用密码术和隐写术。前者隐藏通信内容,而后者则隐藏通信本身。从编码角度而言,前者将通信内容编码成伪随机序列在常规信道上传输,而后者则将通信内容加载在某种常规通信内容(载体)上,用隐藏算法编码后在常规信道上传输[1]。

在应对隐秘通信的工程实践中,往往要求检测并提取隐藏信息,而提取相对于检测是一个更具有挑战意义的工作。常用隐藏方法有最低比特位嵌入(LSB)、DCT 域嵌入等[2]。在关注这些现代隐藏技术同时也必须注意到,在合法图像结构中存在很多可以隐藏任意长度信息的区域,如 jpeg 和 gif 图像中应用程序扩展区、注释区和图像结束符之后,tiff 图像的数据块间隙等。由于图像处理软件对这部分数据不进行解析,因此可以将秘密信息隐藏在图像的这些位置[3]。网上目前有不少工具软件针对这些区域隐藏数据,如果单纯考虑 LSB 隐藏等现代技术,则对信息隐藏检测的实际应用来说是不完整的。

作为信息隐藏检测技术的有益补充,也可以通过研究各种隐藏软件对隐藏载体的处理过程中留下的痕迹来判定载体是否经过隐藏软件处理,或载体中是否包含某种外来格式化数据,进而,如果能够判断该图像使用了什么软件来隐藏信息的话,将有助于判别可疑数据的格式和可能使用的加密算法,从而增大成功恢复数据的可能性。

目前,国内外针对以上两方面进行的研究较少,普遍关注的是对 LSB、DCT 域嵌入等的检测。但它作为隐藏检测技术的一部分,是现代隐藏检测在实际应用中的有益补充,并且提取隐藏信息相对容易,具有很强的工程实用性。对软件处理痕迹的提取和对外来格式数据的检测都可以归结到特征检测,以下的研究工作将紧紧围绕它来展开。

2 特征的分层

在研究搜集了几十种信息隐藏软件处理特征和一些常用格式数据后发现,很多都有明显特征。比如,P. Satya. Kiran 发布的 THE THIRD EYE 在隐写对象中明显的存在"www. binary – techNologies. com"的标记,如图 1 所示。

然而不同软件的特征差别很大,表现异常复杂,有以下几种情况:

(1) 特征有一到多个不等长字节段,段与段之间的间距不固定。这是最简单常见的情况。

(2) 特征分为多个不等长字节段,段与段之间的间距固定。例如,隐藏软件 Hide and Encrypt 在隐藏信息尾部使用 24 字节做为软件数据结束符,特征为 3 字节数据长度和 5 个 0,然后是 3 字节数据长度和 5

① 国家重点基础研究发展规划项目(TG1999035804)、国家自然科学基金(60473016)项目

个0，后跟数据0XA2E601和5个0。某国产软件的隐藏信息尾部为0X1234和三字节任意数据，后跟一字节0。

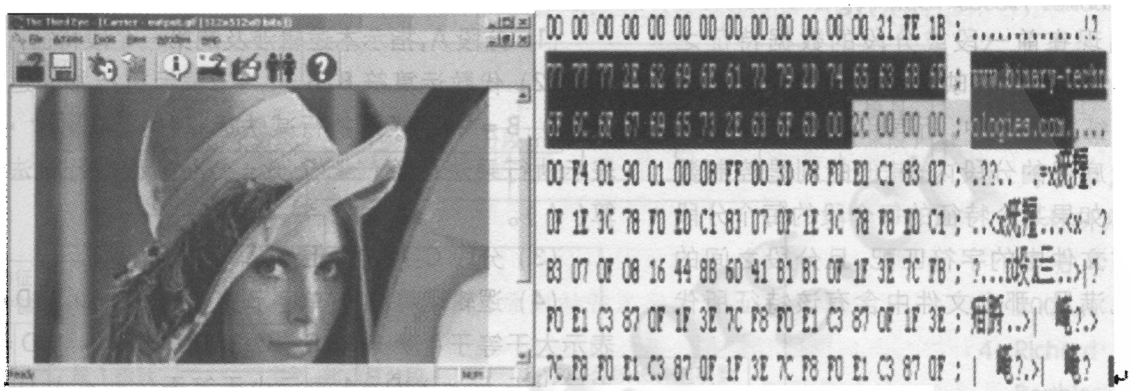

图 1

（3）一些特征段可以有多种取值。例如，GIF文件头标签有“GIF89a”和“GIF87a”两种，TIFF文件头标签有0x4D4D002A和0x49492A00两种，有的格式字段的选项就更多。

（4）一些特征段的取值仅限于某些范围之内，而并非取遍所有可能的值。

（5）一些隐写软件特征或格式化数据位于载体头部，而另一些则位于载体结束处。

（6）一些隐写软件处理过的图像出现了类型/标签－长度－值（T－L－V）型外来数据，可以将之作为一种格式化数据独有的特征。例如，隐藏软件DataStealth的隐藏特征是在文件尾部出现如下内容：第1段格式头：0X200000和某种数据，后跟4字节数据段长度（一般后3字节为0）和数据，然后是0x8c04和6字节0；第2段格式头：0X200000和某种数据，后跟0x06000000（或者0x07000000）和隐藏数据；隐藏信息尾部为0X0079000000和四字节任意数据，然后是四字节0。

（7）特征字段彼此之间存在某种代数关系，但字段本身无固定值。例如tga图像的特征之一为0X0001－X－A－0－B－0－0X0F/0X10/0X18/0X20－0000，其中A＋B＜256，X表示任意字符。

段间距反映了段之间的联系。情况1和情况2的处理过程有本质的不同，段间距固定的系列段只能作为一个刚性整体进行匹配，而段间距不固定的几个段则不然。因此，特征扫描必须以段间距固定的系列段作为基本匹配单位。在匹配发生错误时，不是退回特征开始处重新匹配，或者退回段开始处重新匹配，而是找到当前段所属的系列段中的第一个，从其起始位置前进一字节，开始下一次匹配。如果本系列段已经匹配到扫描目标允许匹配部分的末尾，则对前一个系列段重新匹配，以此类推。

可以借鉴数据挖掘中“聚类搜索”概念处理情况3，将字段的多个选项聚为一类，而不是作为多个特征进行重复扫描。对于情况4到7，也必须在扫描时考虑特征的相应方面。

对于特征的这些复杂情况，特别是后面几种情况，简单的字符串搜索算法无能为力。如果对每个复杂特征都用代码模块实现，不仅代码冗余大，而且不利于用户自主扩展。综合以上所有复杂特征情况，可以将特征的概念加以扩大，不仅包括一些字符串，也包括它们之间的逻辑和距离关系，即特征包括字节流和逻辑两个方面，而不是单纯的字符串匹配。

在针对特征进行信息隐藏检测上，可以用文本数据库存放各类特征数据，使用特征搜索引擎驱动该数据库，执行特征搜索任务，从而判别出载体是否被隐写过。这样可以保持特征搜索引擎的稳定性，今后用户自主扩展时只需要按照特征数据库格式编写特征条目就可以了。而该文本数据库的结构设计应该充分反映出特征的字节流和逻辑两个方面。

综合以上情况，特征数据库在设计上的三层结构就一目了然了，它们分别是：特征、段（即上述的“系列段”）、分段（即上述的“段”）和逻辑，特征由一到多个段组成，段与段之间的间距不固定，特征对多

个段(基本匹配单位)进行管理。段则由一到多个分段信息组成。每个分段为长度为 1 - n 字节的连续特征字符串,分段之间的间距固定。段和分段都遵循严格的先后次序,即待扫描文件的后一段或分段数据特征必须出现在前一段或分段的数据特征之后。段还包含任意数量的反映分段之间联系的逻辑信息。在整个特征数据库中,所有用于进行匹配的字节流都位于最底层的分段内,其他的则是控制数据(包括逻辑)。如果某个特征的每个段的每个分段的某一选项都与文件中的字符匹配,且分段之间的逻辑匹配关系也满足,那么文件中含有该特征所代表的隐藏信息。

3 特征数据库和扫描引擎设计

特征数据库支持多段多选项复杂特征,内嵌简单逻辑运算功能。它分为三层:特征头、段信息、分段信息和逻辑信息。祥述如下:

3.1 特征头

包括类型名字符串、段数和段指针列表。

3.2 段信息

包括段位序标识、段长、分段数、逻辑数、分段指针列表和逻辑指针列表。段位序标识为 0 指示从待扫描对象的头部开始扫描,非 0 则指示从待扫描对象尾部开始扫描,用户可以使用这一字段方便地指定某段特征的扫描位置和方向。段长指示从第一个分段的第一个字符到最后一个分段的最后一个字符之间的长度,包括中间的固定间隔。

3.3 分段信息

包括分段长、选项数、分段间距和分段数据。分段间距用 0XFE 表示最后一段。选项数取值不为 0X00 和 0XFD 、0XFE 、0XFF 时,用于指示本分段特征数据可能有多少种取值;值为 0X00 时,用于指示扫描到的目标数据应大于特征数据;取值为 0XFE 时,用于指示扫描到的目标数据应小于特征数据;取值为 0XFF 时,用于指示扫描到的目标数据应不等于特征数据。特别针对情况 7,在特征数据库的设计上专门增加了"虚分段"的概念。当选项数取值为 0XFD 时,用于指示该段是"虚分段",扫描到的目标数据仅应用于逻辑,而不和分段数据进行字节匹配。分段数据中依次排列各种特征字节流选项。

3.4 逻辑信息

包括逻辑类型等控制数据。逻辑类型 T 取值为 0,表示可以建立代数运算:(分段 A)代数运算符 B(分段 C)逻辑运算符 D(操作数 E)。后续数据项依次为:

(1) 分段 A,指示本逻辑涉及的第一个段特征;

(2) 代数运算符 B:B = ' + ' 表示执行加法运算(+);B = ' - ' 表示执行减法运算(-);B = ' * ' 表示执行乘法运算(*);B = '/' 表示执行除法运算(/)。

(3) 分段 C:规定同上。

(4) 逻辑运算符 D:单字节整数。取值如下:D = 1 表示大于等于(> =);D = 2 表示大于(>);D = 3 表示小于 (<);D = 4 表示小于等于(< =);D = 5 表示不等于。

(5) 操作数 E。

逻辑类型取值为 1,表示分段为 T - L - V 型(类型 - 长度 - 值)数据,可以建立双字段(分段 A,回绕字节数 B),其中字段 B 指示读写指针向后跳跃字节数。读写指针跳跃到指定位置后,再将段 A 中的数据作为 T - L - V 型数据中的 L(长度),前向跳跃[A 中数据]个字节。

特征数据库分级结构中的各项和待扫描对象特征之间的关系可以简图 2 表示。

特征扫描引擎的主要任务是解析特征数据库中的各个数据项,以段为基本匹配单位从头尾两个方向相向扫描,类似特征这个多节段"蠕虫"在待扫描目标上爬行,因此称之为"双向蠕行"技术。

4 实验结论

实验选取了 Masker、JPEGX、Hide and Encrypt、Cloak、Invisible Secrets、DataStealth、Safe & Quick Hide Files and Folders、渗透、Steganography、Data Stash 等十余种信息隐藏软件进行测试,提取了它们的特征码编写成特征数据库中的条目后,能达到 100% 的检出率。

同时,实验提取了数据库类、文档类、图像类、音频类、压缩类五种类型各种格式的文件特征进行研究,如果在载体文件中发现不属于原文件的外来格式化数据,则可以认为该文件被改写过,隐藏了某类信息。该项检出率也可以达到 100% 。

基于本文开发的面向特征的检测算法系统不光可以应用于信息隐藏检测，在病毒扫描、未知协议识别和阻断等许多方面都具有一定的应用前景。随着应用的扩大，算法也将从执行效率和功能两方面不断完善。

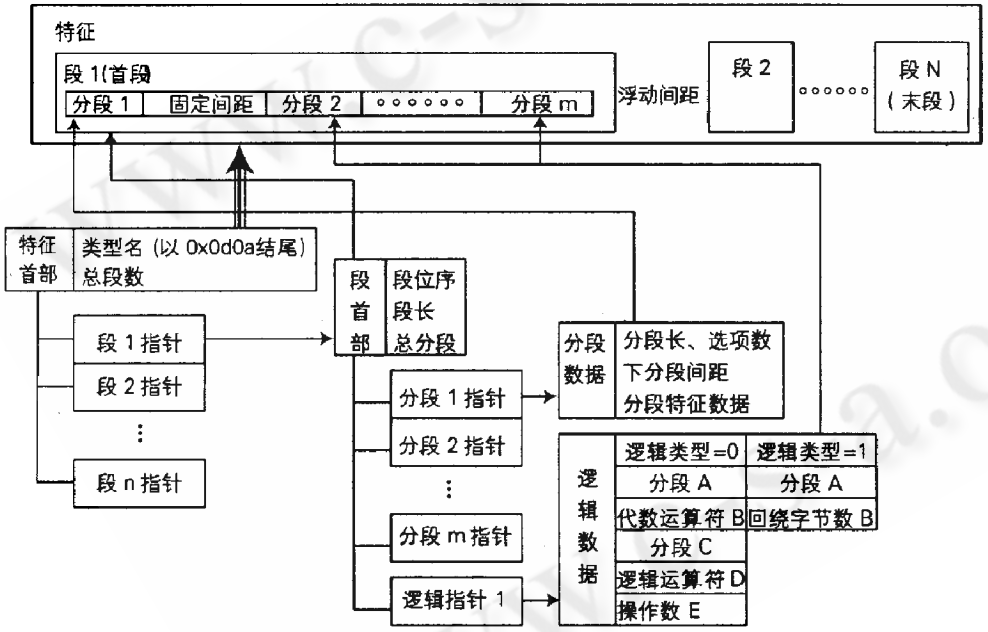

图 2

参考文献

1 Stefan Katzenbeisser, Fabien A. P, Petitcolas. Information Hiding Techniques for Stegnography and Digital Watermarking, Artech House, London, 2000.

2 刘振华、尹萍，信息隐藏技术及其应用，科学出版社，北京，2002。

3 周继军，信息隐藏逆向分析研究，北京邮电大学博士学位论文，北京，2005。

4 Richard Baeza – Yates, Berthier Ribeiro – Neto. Modern Information Retrieval, Addison Wesley, 2004.

5 Michael Steinbach, Geoge Karypis, Vipin Kumar, A Comparison of Document Clustering Techniques, 2000.