

# 基于信息隐藏技术的 Outlook 邮件隐藏插件开发

余鹏飞<sup>1</sup>, 周浩<sup>1</sup>, 刘兵<sup>2</sup>

(1. 云南大学信息学院, 昆明 650091; 2. 云南大学软件学院, 昆明 650091)

**摘要:** 信息隐藏技术可以应用在隐秘通信和数字媒体版权保护领域中, COM Add-in 技术使用 COM 接口扩展 Office 等软件的功能。该文讨论了在 Microsoft Outlook 下利用 COM Add-in 技术, 实现邮件隐藏插件的方法。该插件被无缝地集成在 Outlook 中, 并将邮件内容隐藏到图像文件中, 为邮件信息安全提供了一种解决方案。

**关键词:** 信息隐藏; 邮件隐藏插件; COM add-in

## Development of E-mail Concealment Add-in for Outlook Based on Information Concealment Technology

YU Pengfei<sup>1</sup>, ZHOU Hao<sup>1</sup>, LIU Bing<sup>2</sup>

(1. College of Information, Yunnan University, Kunming 650091; 2. College of Software, Yunnan University, Kunming 650091)

**【Abstract】** Information concealment technology is used in secret communication and copyright protection of digital media. COM add-in uses COM interface to extend office software's function. This paper discusses the realization of E-mail hiding plug-ins under Outlook. The E-mail hiding plug-ins, which can be used to hide E-mail content into an image file, and it is integrated seamlessly in Outlook, provides a novel method for E-mail security.

**【Key words】** Information concealment; E-mail concealment plug-ins; COM add-in

网络信息安全是关系到个人隐私保护、企业信息安全乃至国家安全的关键问题。电子邮件作为企业和个人交流的常用方式, 其安全问题更是受到各方面的关注。为了保护邮件的内容不被窃取, 邮件用户可以使用密码学的方法对邮件的内容进行加密。加密的方法对于保护邮件信息安全来说是一种行之有效的方法, 但经过加密以后, 邮件内容变成了一些乱码, 攻击者很容易从形式上发现邮件是被加密处理过的, 如果加密的强度不够, 例如: 加密口令太短, 或者加密算法有缺陷, 很容易被攻击者所破解。

作为一门新兴的学科, 信息隐藏利用多媒体数据具有很大冗余性的特点, 将要保护的信息隐藏在多媒体载体文件中。由于掩蔽效应, 人眼对灰度和彩色图像的分辨率有限, 因此基于图像的信息隐藏技术是当前研究的热点。从嵌入信息的方式来看, 可以分为空域和变换域两大类。其中, 基于空域的 LSB 方法由于嵌入的信息量大, 恢复信息的时候不需要载体图像, 是一种有效而简便易行的信息隐藏方法。

### 1 邮件隐藏原理

LSB(Least Significant Bit)算法<sup>[1]</sup>是基于空域的信息隐藏方法的, 该方法具有速度快、容量大、不可见性好的特点。这种方法使用计算机系统中以无损格式存放的图像文件作为嵌入的载体, 例如BMP位图文件和TIFF格式的图像文件。本文采用该算法作为邮件隐藏的具体方法。Outlook邮件隐藏插件采用的是BMP位图文件格式, 能够将邮件的内容嵌入到 24 位或者 256 色的位图文件中。

#### 1.1 邮件隐藏方法

**Step1** 将标志“InfoHide”转化成连续的二进制串  $M = \{m_1, m_2, \dots, m_p\}$ ,  $p=64$ ; 该标志用于在提取邮件内容时, 判

断位图文件是否包含隐藏信息;

**Step2** 将邮件所占的字节数也转换成二进制串  $L = \{l_1, l_2, \dots, l_k\}$ ,  $k=48$ ; 以便于提取邮件内容时得到邮件的准确长度;

**Step3** 将邮件的内容转化成连续的二进制串  $C = \{c_1, c_2, \dots, c_n\}$ ;

**Step4** 将  $M$ 、 $L$ 、 $C$  依次顺序连接起来, 即构成了嵌入到位图文件中的二进制水印序列  $W = \{w_1, w_2, \dots, w_q\}$ ; 其中,  $q = n + k + p$ ;

**Step5** 读入位图文件, 依据图像文件的大小和  $W$  的长度  $q$ , 判断是否能完全嵌入到位图文件中; 如果可以嵌入则转到 Step 6, 否则退出程序;

**Step6** 取出位图文件数据区中一个字节的数  $B_i$ ,  $1 \leq i \leq q$ , 把它转化为二进制形式, 然后将  $B_i$  二进制最低位用水印序列中的一位  $w_i$  替换之;

**Step7** 重复上述操作, 直到水印序列  $W$  的所有数据嵌入到图像中;

**Step8** 将图像作为邮件的附件发送。

#### 1.2 邮件提取方法

**Step1** 取出邮件附件中的位图文件;

**Step2** 顺序取出位图文件数据区中  $p$  个字节的数  $B_i$ ,  $1 \leq i \leq p$ , 取出每个字节二进制形式的最低位, 组成一个二进制

**基金项目:** 云南省自然科学基金资助项目(2002D0003Q); 云南大理理工科校级科研基金资助项目(2003Q025C)

**作者简介:** 余鹏飞(1974-), 男, 讲师、硕士, 主研方向: 图像处理; 周浩、刘兵, 讲师、硕士

**收稿日期:** 2006-02-08 **E-mail:** pfyu@vip.km169.com

串 $M'$ ，如果 $M' = M$ ，则代表附件文件中的图像包含有隐藏的邮件，否则退出程序；

**Step3** 顺序取出位图文件数据区中 $k$ 个字节的数据 $B_i$ ， $p < i \leq p+k$ ，取出每个字节二进制形式的最低位，组成一个二进制串 $L'$ ，求出邮件内容的长度；

**Step4** 顺序取出位图文件数据区中一个字节的数据 $B_i$ ， $p+k < i \leq q$ ，取出 $B_i$ 的二进制形式的最低位 $b_{i0}$ ；

**Step5** 重复Step4，直到取出所有数据，组成一个二进制串 $\{b_{10}, b_{20}, \dots, b_{n0}\}$ ，令为 $C'$ ；

**Step6** 将提取出来的二进制串 $C'$ 转换成邮件内容。

为了保证邮件的安全性，还可以在邮件嵌入到图像之前用密码学的方法，例如 DES 算法或者 AES 算法对邮件进行加密。

## 2 插件工作原理及关键代码介绍

### 2.1 插件工作原理

Office 用户可以使用 VBA 语言来扩展 Office 软件的功能，但是，这种方式有很大的局限性。首先，宏附带在某一个 Office 文件当中的程序，如果要在其它文件中使用时较为麻烦；其次，受宏病毒问题的困扰，在打开带有宏的文件时，Office 程序都会给出警告信息，给用户带来不便和困扰。为此，从 Office 2000 以后，微软公司用一种全新的方式——COM Add-in 技术提供了 Office 开发的接口。该技术基于 COM(Component Object Model)技术，开发人员可以通过编写 COM 组件来扩展 Office 软件的功能。

Office 软件都提供了一个接口——IDTExtensibility2，该接口实际上是 IDTExtensibility 接口的扩展，Add-in 插件就是通过该接口连接到 Office 软件。开发者可以用支持 COM 开发的软件开发工具，例如：Visual Basic 导入该接口，然后编写扩展 Office 软件功能的 COM Add-in 应用程序。

Office 软件本质上是实现了 IDispatch 接口的一个自动化服务器(Automation Server)，Add-in 插件通过 IDispatch 接口提供的成员函数实现对 Office 软件自动化服务器内部函数的间接调用，是和 Office 软件进行通信的 COM 客户。通俗地说，IDispatch 接口主要用来处理和交互相关的部分。Add-in 插件实际上是一些 COM 对象，为了创建这些对象，要使用 IClassFactory 接口。

邮件隐藏插件实际上是一个 COM 客户程序，通过 IDTExtensibility 接口连接到 Outlook 自动化服务器，即 COM 服务器中，然后用 IDispatch 提供的接口函数和 Outlook 进行通信和交互。而用户模块层定义的信息隐藏模块和加密模块则可以根据用户的需求进行扩展，例如：采用不同的信息隐藏技术手段或者实现不同的加密算法。

图 1 显示了 Outlook 邮件信息隐藏插件的结构。图 2 显示了 Outlook 对象模型的一些主要对象，其中，Application 为应用程序对象，是 Outlook 对象模型的核心，通过该对象就可以访问到 Outlook 的其它子对象，还可以对 Outlook 的一些事件进行响应，例如收到新邮件、邮件发送等。Explorer 对象是浏览器对象，用来显示项目的集合，对应着用户启动 Outlook 以后看到的窗口界面。项目是 Outlook 中的基本信息单元，例如邮件、联系人。Inspectors 对象是 Inspector(检查器)对象的集合。Inspector 对象用来显示具体的项目，例如邮件项目。当用户新建邮件时，就会将邮件显示在 Inspector 对应的窗口对象中。CommandBars 对象对应着应用程序窗口的

工具栏。NameSpace 对象则是 Outlook 的数据源，当用户要访问 Outlook 的数据时，必须使用该对象。

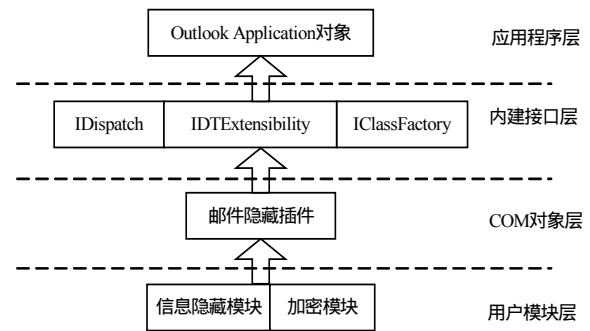


图 1 邮件隐藏插件结构

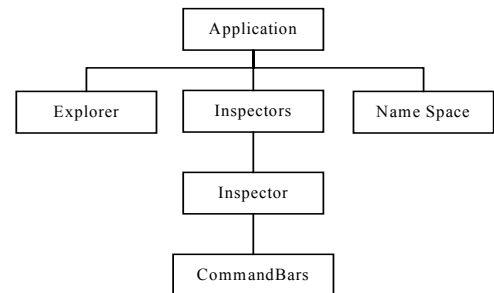


图 2 插件使用的 Outlook 对象

### 2.2 主要的实现步骤和关键代码

以 Visual Basic 为例来说明在 Outlook 中实现邮件隐藏插件的重要代码。首先必须创建一个 in-process COM server 工程，该 COM server 必须实现 IDTExtensibility2 接口。该接口包括 5 个重要的方法：

- (1)OnConnection, Outlook 应用程序启动后加载 COM Add-in 时就会调用该方法，因此该方法中主要包括了 COM Add-in 的初始化代码，例如对象变量赋初值，建立用户自己的菜单、工具条和按钮等；
- (2)OnStartupComplete Outlook 应用程序启动完毕后调用本方法；
- (3)OnAddinsUpdate 加载或者卸载 COM Add-in 时调用本方法；
- (4)OnBeginShutDown Outlook 的应用程序关闭前调用本方法；
- (5)OnDisconnection 卸载 COM Add-in 时调用本方法。

在进行 COM Add-in 开发的过程中，开发人员经常需要操作 Outlook 的各种对象。使用对象之前要事先定义一个对象变量，并在适当的时候对它初始化赋值，就可以在后续的操作中使用它。例如，开发者如果要使用 Outlook 的 Application 对象，可以在 Visual Basic 代码中定义：

```
Dim WithEvents objOutlook As Outlook.Application
```

在 IDTExtensibility2 接口的 OnConnection 事件中进行初始化赋值：

```
Set objOutlook = Application
```

做了上述定义和赋值以后，开发人员就可以通过 objOutlook 获得其它子对象，进而对 Outlook 的一些事件进行编程，比如为邮件发送、收到新邮件等事件编写响应代码。

为了方便用户的操作，邮件信息隐藏插件定义了 4 个工具条按钮，分别对应隐藏邮件到图片、从图片中提取邮件、邮件加密、邮件解密 4 个功能。以邮件隐藏功能为例，可以在 Visual Basic 代码中定义

```
Private WithEvents CBHideMail As Office.CommandBarButton
```

定义了上述的按钮以后，要在 IDTExtensibility2 接口的 OnConnection 事件中编写代码添加这些工具条按钮。然后，

就可以在 VB 中便写相应的按钮点击事件响应代码。以隐藏邮件代码为例：

```
Private Sub CBHideMail_Click(ByVal Ctrl As Office.CommandBarButton, CancelDefault As Boolean)
    Dim myItem As Outlook.MailItem
    Dim strMailBody As String
    Dim i As Integer
    Dim ofn As OPENFILENAME
    Dim rtn As String
    Dim SourceFileName As String
    '取得邮件的内容
    Set myItem = myInspector.CurrentItem
    strMailBody = myItem.Body
    '显示文件打开对话框,选择载体图像
    ofn.lStructSize = Len(ofn)
    ofn.hwndOwner = myApplication
    ofn.hInstance = App.hInstance
    ofn.lpstrFilter = "位图文件(*.bmp)" + vbNullChar + "*.bmp"
    ofn.lpstrFile = Space(254)
    ofn.nMaxFile = 255
    ofn.lpstrFileTitle = Space(254)
    ofn.nMaxFileTitle = 255
    ofn.lpstrInitialDir = App.Path
    ofn.lpstrTitle = "打开文件"
    ofn.flags = 6148
    rtn = GetOpenFileName(ofn)
    If rtn >= 1 Then
        SourceFileName = ofn.lpstrFile
    Else
        Exit Sub
    End If
    '调用 HideInfoToImage 函数执行嵌入邮件功能
    If HideInfoToImage(strMailBody, SourceFileName)=0 Then
        MsgBox "嵌入成功!", vbOKOnly, "邮件隐藏插件 1.00"
    End If
    '将包含隐藏信息的图像文件 info.bmp 加入到邮件附件中
    myItem.Attachments.Add (App.Path + "\info.bmp")
End Sub
```

(上接第 244 页)

boolean mo(Map params) 函数作为其处理函数。然后程序根据手机号和发送信息到数据库中验证该用户是否有权限访问此监控终端。如果有权限访问则启动一个进程去和目的监控终端通信,取回终端上发送回的视频文件,把这些视频文件按照一定格式连同其程序自动生成的彩信配置文件(smile 文件)一同通过 SendMMS 类发送给用户。

#### 4 实验结果与结论

目前该系统已经整体完成了从用户点播到最后用户收到数据的整体测试。实验结果表明系统操作方便灵活,采用点播方式,事件触发,反应迅速。系统功能齐备且扩充性好,管理方便。终端设备采用嵌入式系统构建,具有嵌入式系统的普遍优点,如成本低、体积小、重量轻、功耗低、稳定性好、网络功能强大等。此外,终端设备还有一个特点是采用

End Sub

### 3 结论

Outlook 邮件隐藏插件采用 Visual Basic6.0 来开发,可以运行在 Outlook 2000 及其后的版本下,具体的运行效果如图 3 所示,当用户收发邮件时,在操作界面上可以看到插件提供的按钮,并根据需要进行操作。

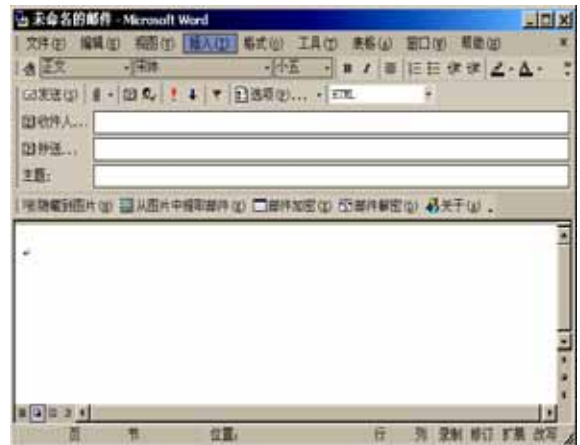


图 3 邮件隐藏插件运行

Outlook 邮件隐藏插件充分利用了 COM Add-in 技术,扩展了 Outlook 的功能,为邮件信息安全提供了一种新的解决方案。由邮件隐藏的方法可知,位图文件数据区的每个字节可以隐藏 1bit 的信息,一个 8kB 大小的位图文件大约可以隐藏 1kB 的邮件信息。这样的容量足以隐藏一般的邮件,同时也不会占用太多的网络带宽。

#### 参考文献

- 1 Tirkel A Z, Rankin G A, Schyndel R M V, et al. Electronic Watermark[C]. Proceedings of Digital Image Computing, Technology and Applications, London, 1993: 666-672.
- 2 Rogerson D. 杨秀章译. COM 技术内幕——微软组建对象模型[M]. 北京: 清华大学出版社, 1999.
- 3 Clarkson J. Creating a Microsoft Office 2000 COM Add-in[EB/OL]. <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dno2kta/html/mso2kaddin.asp>, 2005-05-15.

即插即用的 USB 摄像头,比起一般的监控系统在成本和易用性方面具有明显的优势。非常适用于专业化要求不高的领域的监护服务,为大众化的监护服务提供了一种可行的解决方案,具有一定的推广价值。

#### 参考文献

- 1 Hermans L. Linux Embedding Mini How to[EB/OL]. <http://www.dobit.com/emblin/embhowto.htm>, 2000.
- 2 雅 莫. 构建嵌入式 Linux 系统[M]. 北京: 中国电力出版社, 2005.
- 3 Mitchell M, Oldham J, Samuel A. Advanced Linux Programming[M]. New Riders Publishing, 2001.
- 4 Corbet J. Rubini A, Hartman G K. Linux Device Driver[M]. O'Reilly Publishing, 2000.