计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2018,27(10):85−90 [doi: 10.15888/j.cnki.csa.006575]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 ObjectARX 的工程图纸安全管理系统①

陈玥秀, 张应中, 罗晓芳

(大连理工大学 机械工程学院, 大连 116024)
通讯作者: 张应中, E-mail: zhangyz@dlut.edu.cn

摘　要: 随着 CAD 图纸的广泛使用, 图纸安全受到广泛关注. 本文主要研究在 AutoCAD 环境下的工程图纸操作安全, 使用 AutoCAD 二次开发工具 ObjectARX 实现了本系统. 使用 ARX 反应器技术, 监视和控制 AutoCAD 的打印和保存命令. 使用 API HOOK 技术拦截剪切板函数, 改变其执行结果, 控制 AutoCAD 的复制粘贴操作. 同时实现了 PrtSc 键和软件截屏的控制. 测试结果表明, 该系统能够对用户操作进行监控和管理, 有效的保证电子图纸的完整性, 而且不影响用户使用习惯.

关键词: 工程图纸; 操作安全; ObjectARX; ARX 反应器; API HOOK

引用格式: 陈玥秀,张应中,罗晓芳.基于 ObjectARX 的工程图纸安全管理系统.计算机系统应用,2018,27(10):85–90. http://www.c-s-a.org.cn/1003-3254/6575.html

Engineering Drawing Safety Management System Based on ObjectARX

CHEN Yue-Xiu, ZHANG Ying-Zhong, LUO Xiao-Fang

(School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China)

Abstract: With the extensive use of CAD drawing, drawing safety is widely concerned. This study mainly focuses on the operation safety of engineering drawings under AutoCAD environment. A safety management system for engineering drawing is developed with ObjectARX technology. In this system, ARX reactor technology is employed to monitor and control AutoCAD printing and save operations. API HOOK functions are used to intercept the clipboard function in order to change its execution results, and to control the copy and paste operations. In addition, the PrtSc key and Screenshot operations are controlled. The test results show that the system can monitor and manage the user operation, and effectively ensure the integrity of the electronic drawings, at the same time, it does not affect the user's usage habits.

Key words: engineering drawings; operation safety; ObjectARX; ARX reactor; API HOOK

随着 AutoCAD 等制图软件的广泛使用, 传统的手工绘图已经被电子绘图所取代. 虽然电子工程图纸有绘制速度快, 传输便捷等优点, 但是易于泄露和损坏. 工程图纸是企业的知识财富, 是重要的商业机密, 如果泄露和损坏, 会给企业带来巨大的损失, 因此保证工程图纸安全有非常重要的意义.

现有的 PDM(Product Data Management) 系统有很强的图档保护功能, 包括防病毒侵入, 记录文档操作等, 但缺乏嵌入具体绘图软件的图纸安全管理. 目前的安全管理系统对于图纸加密算法的研究在不断深入, 但对于管理用户操作安全的技术研究却很少. 文献[1]通过研究加密算法和数字水印等技术, 保障图纸在流转过程中信息的机密性、完整性和真实性, 实现了图纸加密、图纸电子签章防篡改和篡改检测功能. 文献[2]主要针对 DWG 图纸格式特点, 提取图纸中各种实体信息, 然后对信息进行置乱处理达到图纸加密保护

① 基金项目: 国家自然科学基金 (51775081)
Foundation item: National Natural Science Foundation of China (51775081)
收稿时间: 2018-03-06; 修改时间: 2018-03-22; 采用时间: 2018-04-02; csa 在线出版时间: 2018-09-28

的目的. 文献[1,2]提出的图纸安全管理系统, 可以在图纸流转中保护图纸信息的安全, 但是对于企业内工作人员的泄密并没有加以防范. 工作人员与图纸的接触最为紧密频繁, 他们对图纸的越权访问和不当操作, 都会造成图纸信息的内部泄露, 而且是日常工作中图纸信息泄密的主要方式.

本文设计并实现了一个基于 ObjectARX 的工程图纸安全管理系统, 根据用户的操作权限, 能够有效的控制用户对 AutoCAD 某些功能 (如打印、保存和复制等) 的操作, 还能够有效防止用户使用工具 (PrtSc 和 QQ 等) 截屏, 从而能够保障工程图纸的安全, 对工程图纸在网络环境下的安全的管理有重要的作用.

1 相关技术

1.1 ObjectARX 二次开发技术

ObjectARX 是 AutoDesk 针对 AutoCAD 平台的二次开发而推出的一个开发软件包, 是一个以 C++语言为基础的面向对象的开发环境和应用程序接口. ObjectARX 应用程序本质上是一个 Windows 的 DLL 程序, 共享 AutoCAD 的地址空间, 使用 ObjectARX 可直接访问 AutoCAD 数据库的核心数据库结构、图形系统以及 AutoCAD 几何构造核心[3]. 同时可以在 ObjectARX 编程环境下添加新类, 并将其供其他程序使用, 简洁高效的实现多种复杂的功能.

1.2 ARX 反应器技术

当在 AutoCAD 发生各种事件时, 例如执行 AutoCAD 命令、修改系统变量、保存和退出图形编辑器等等, 事件通知者对象会自动将这个事件传递给事件接收者对象, 接收到消息后会对事件做出响应. 事件接收者对象称为反应器[4], 反应器的类型不同, 接收的事件也不相同. 事件的通知者对象根据需要可以有不同的反应器, 以便监视并响应不同的类型的事件. 当事件发生后, 通知者对象会调用反应器列表中对应的通知函数. 事件的处理过程如图 1 所示.

1.3 Windows 钩子技术

钩子是 Windows 操作系统中一种特殊的消息处理函数. 当某应用程序发出消息后, 在没有到达目的窗口前, 钩子过程会捕获感兴趣的消息. 这时钩子函数可以处理并改变消息, 或决定是否继续传递消息, 从而实现消息的拦截处理过程[5]. 钩子按照作用范围分为进程内钩子和全局钩子. 进程内钩子监视指定进程的主线程的事件消息, 全局钩子监视 Windows 操作系统中所有线程的事件消息. 使用钩子前用 SetWindowsHookEx() 函数来安装钩子过程, 不再使用钩子后用 UnHookWindowsHookEx() 函数进行卸载.

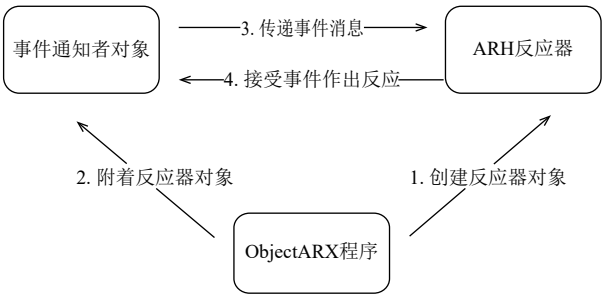

图 1 事件处理过程

API HOOK 是指截获特定系统或进程对某个 API 函数的调用, 使 API 函数转向自定义代码的技术. Windows 中的每个进程都有特定的地址空间, 进程在调用函数时, 只能调用自己地址空间中的函数. 为了改变目标进程对 API 函数的执行结果, 必须先自定义一个拦截函数来替换目标函数注入到目标进程中. 使用微软的 Detours 库可以拦截目标 API 函数, 拦截代码是在动态运行时加载的. Detours 技术将目标函数前几条指令替换为其无条件跳转到拦截函数的指令, 并将被替换的指令和目标函数其余部分的无条件分支保存到跳板函数中. 当执行到目标函数时, 将直接跳转到拦截函数, 拦截函数可以执行适当的拦截处理. 拦截函数可以直接返回对源函数的控制, 也可以调用跳板函数, 该函数调用目标函数而不被拦截. 当目标函数完成时, 返回到拦截函数, 拦截函数处理后, 将控制权返回给源函数[6]. 图 2 展示了 Detours 技术拦截前后函数调用过程.

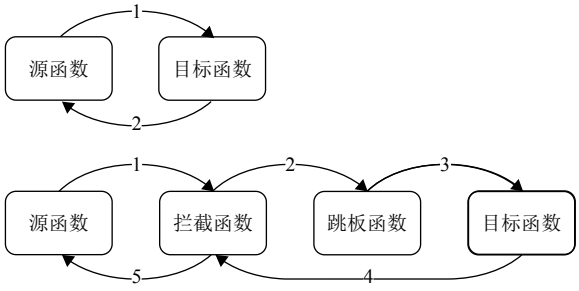

图 2 Detours 拦截前后函数调用过程

2 图纸安全管理系统分析及设计

图纸文档被打开后是明文, 此时用户的不当操作将会造成图纸信息的泄露和破坏. 例如当用户使用 AutoCAD 程序的另存为和打印操作, 图纸信息将会泄

露; 当不允许修改的图纸被打开后, 用户的保存操作会破坏图纸信息. 诸如此类的用户不当操作, 均危及到图纸信息的安全. 图纸操作安全管理系统以图纸管理系统为基础, 以用户的操作权限为基准限制不同权限用户对图纸的操作. 安全管理系统主要包括四大功能: 对图纸打印控制, 图纸内容修改控制, 图纸内容复制粘贴控制, 截屏控制, 安全控制流程如图 3 所示.

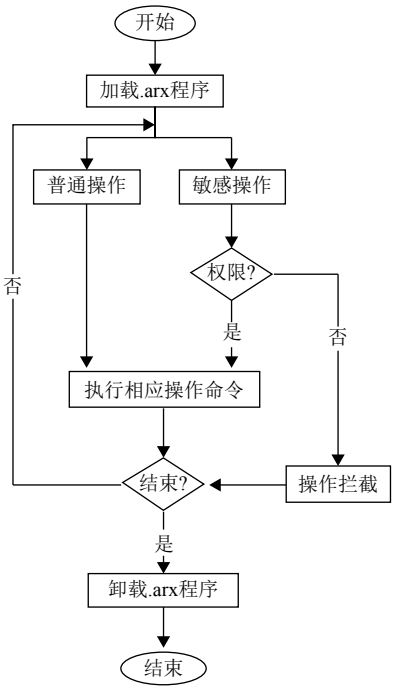

图 3 图纸安全控制流程

2.1 打印控制

打印控制的目的是, 禁止没有打印权限的用户将图纸以明文的方式打印成静态图纸. 通过对 AutoCAD 操作和 ObjectARX 技术的研究, 发现 AutoCAD 的每一个操作对应一个命令, 且每个命令对应一个字符串, 而 ARX 反应器恰恰是用来监视 AutoCAD 命令的. 通过使用 ARX 反应器来监视打印 (PLOT) 命令, 截获到打印命令后, 判断当前操作用户是否具有图档打印权限, 如果可以打印则将命令继续传递, 否则终止打印命令.

2.2 保存控制

保存操作主要有两大类, 一是保存操作, 将当前未命名的图纸命名保存或将已命名的图纸更新保存; 二是另存为操作, 将当前未命名的图纸命名保存或将已命名的图纸另外存储. 在 AutoCAD 中保存命令是 QSAVE, 另存为命令是 SAVEAS 和 SAVE. 通过控制保存 (QSAVE) 命令, 可以禁止无修改权限的用户修改图纸. 通过控制另存为 (SAVEAS 和 SAVE) 命令, 可以禁止用户将图纸保存到其他磁盘文件中, 有效防止图纸脱离管理系统造成泄露. 使用 ARX 反应器来监视保存命令, 截获 QSAVE 命令时, 通过判断用户的权限来决定是否继续传递 QSAVE 命令, 截获 SAVEAS 和 SAVE 命令后, 直接禁止命令继续传递.

通过控制保存操作可以使不具备修改权限的用户无法保存被修改的图纸, 但是被修改过的图纸在执行关闭图档 (CLOSE 和 CLOSEALL) 命令和关闭 AutoCAD 程序 (QUIT 和 EXIT) 命令时, 会弹出是否保存提示对话框. 若用户点击按钮“是”, 则当前图纸将被保存, 图纸内容被修改. 经研究发现, 未修改过的图纸将直接被关闭, 因此在关闭前告知应用程序图纸没有修改即可. 该何时告知应用程序? 一是在用户所有的修改操作结束之后, 二是在关闭命令执行前. AutoCAD 中图纸的修改命令繁多, 且难以预测用户何时停止修改图纸, 但是关闭命令可以预测和监视. 使用 ARX 反应器监控关闭命令, 在关闭命令执行前通知程序图纸未被修改, 然后执行关闭命令将图纸直接关闭.

2.3 复制粘贴控制

在 AutoCAD 的剪切板中有这样一组操作:粘贴、剪切和复制剪裁, 用来存储 (或提取) 选中的对象到剪切板中, 实现图形内部和图形之间对象的传输. 对图纸内容的复制粘贴操作, 实际上是使用 Windows 剪切板. 因为 Windows 的剪切板是进程间的通讯, 所有进程共享剪切板[7], 所以当用户执行粘贴操作时, 会将被复制或剪切的图纸信息从 Windows 剪切板中取出, 造成图纸信息泄露. 通过研究 Windows 剪切板的工作原理, 我们发现数据在剪切板中的存取是通过 API 函数来实现的. 复制剪切操作过程有四个步骤, 首先 OpenClipboard() 打开剪切板, 接着 EmptyClipboard() 清空剪切板内容, 然后 SetClipboardData() 设置剪切板内容, 最后 CloseClipboard() 关闭剪切板. 粘贴操作过程有三个步骤, 首先 OpenClipboard() 打开剪切板, 然后 GetClipboardData() 获取剪切板内容, 最后 CloseClipboard() 关闭剪切板.

复制粘贴的核心是使用 SetClipboardData() 和 GetClipboardData() 函数存取数据到剪切板中. 使用 API HOOK 技术, 自定义中间函数替换目标函数 SetClipboardData() 和 GetClipBoardData(), 实现复制粘贴控制.

2.4 截图控制

现在有许多工具都带有截图功能, 使用的方法更是多种多样, 从技术上讲, 要彻底防止其他应用程序进行拷屏是不可能的. 本文分析最常用的两种截图方法, 键盘上的“屏幕拷贝”功能键和软件截屏.

禁止“屏幕拷贝”功能键截屏, 即拦截键盘上的 PrtSc 键. 我们常用键盘钩子 (WH_KEYBOARD) 来截获键盘消息, 让它不再继续传递. 经研究发现 PrtSc 是系统热键, 键盘钩子不能获取到系统键的输入, 所以键盘钩子不能屏蔽 PrtSc 键. 但底层键盘钩子 (WH_KEYBOARD_LL) 可以截获系统键[8], 并根据用户操作权限决定是否对消息进行拦截. 当用户按下 PrtSc 或 Alt+PrtSc 时, 将消息拦截即可屏蔽“屏幕拷贝”功能键.

常用的截屏软件工具有 QQ、微信、360 浏览器等[9]. 软件截屏的一般流程是先获得屏幕设备对象句柄, 然后再根据句柄对屏幕数据进行获取, 生成了我们截取的图片, 截图保存到剪切板中等待粘贴到界面中. 这一系列的操作中使用到许多 API 函数, 但是我们不能使用 API HOOK 技术屏蔽软件截屏工具, 原因主要有以下两点: 一是在截图中所用到的一些 API 函数是图形化应用软件的公用函数, 阻止这类函数将影响 AutoCAD 的正常使用; 二是无法预知用户使用何种截图软件, 若使用 API HOOK 技术需要将钩子注入到所有可以截图的软件中, 这种做法是无法实现的. 本文通过分析软件截屏的工作原理, 将使用定时器对 Windows 剪切板进行清除, 从而达到拦截软件截屏的目的.

3 图纸安全管理系统实现

基于 VS2010, 使用 ObjectARX 技术二次开发 AutoCAD 程序, 实现了 AutoCAD 中打印、保存、复制粘贴和其他软件截图工具的控制. 当用户使用图纸管理系统查看图纸时, 图纸操作安全管理系统自动启用, 即将 ObjectARX 文件自动加载到 AutoCAD 程序中, 当用户执行敏感操作时, 通过判断用户权限来限制用户的操作, 最后关闭 AutoCAD 时自动卸载 ObjectARX 文件.

3.1 打印控制的实现

打印控制的目的是禁止不具备打印权限的用户打印图纸, 造成图纸信息泄露. 系统通过判断用户的打印权限, 控制用户的打印操作. 允许打印的用户正常使用打印功能, 禁止打印的用户其打印操作将会被拦截, 同时 AutoCAD 控制台提示禁止打印.

当用户输入命令执行前, 文档管理反应器中的 documentLockModeChanged() 函数将会被调用. 函数 documentLockModeChange() 是一个锁定请求的回调函数, 在函数 documentLockModeChanged() 执行期间调用 veto() 函数会使锁定请求被阻止, 意味着在开始执行命令之前取消命令. 要阻止打印 (PLOT) 命令执行, 即需要将命令在执行前取消掉. 具体实现代码如下:

```
Void CControl::documentLockModeChanged(
AcApDocument*,
AcAp:: DocLockMode myPreviousMode,
AcAp:: DocLockMode myCurrentMode,
AcAp::DocLockMode currentMode,
const ACHAR* pGlobalCmdName )
        {if (_tcscmp(pGlobalCmdName,
_T("PLOT"))==0)//输入打印命令 PLOT
          {if (没有打印权限)
            {acutPrintf(_T("\n 禁止打印操作！\n"));
            //在 AutoCAD 的控制台中输出提示
             MessageBox(NULL, TEXT("禁止打印操
作!"), NULL, MB_OK);//弹出提示对话框
             Acad::ErrorStatus es=veto();//禁止执行打印
命令
          }}}
```

3.2 保存控制的实现

保存控制的目的主要有两个, 一是防止不具备修改权限的用户在查阅图纸时修改图纸, 造成图纸信息损坏, 二是禁止用户使用另存为方法, 保障图纸存储位置不发生改变. 保存操作的控制方法分为两类, 第一类是对 AutoCAD 中 QSAVE、SAVEAS、SAVE 命令的拦截, 与打印命令的控制方法相同, 此处不再赘述. 第二类是执行关闭图档 (CLOSE 和 CLOSEALL) 命令和关闭 AutoCAD 程序 (QUIT 和 EXIT) 命令时, 拦截保存提示框.

由上文叙述可知, 为拦截保存提示框, 需要在关闭命令执行前, 通知 AutoCAD 程序图纸未修改. 编辑反应器用来监视 AutoCAD 命令, 当开始执行命令前会触发 commandWillStart() 函数. 通过研究发现, AutoCAD 是通过变量 DBMOD 的值来判断图纸是否被修改过, 从而来确定关闭时是否弹出保存提示框. 变量 DBMOD 是只读的, 因此不能直接修改变量的值, 需要通过接口函数 ACDBSETDBMOD() 来修改变量值.

因此在 COMMANDWILLSTART() 函数中, 使用 ACDBSETDBMOD() 接口函数使变量 DBMOD 的值为 0, 即告知 AUTOCAD 程序图纸未被修改, 随后继续执行关闭命令将程序关闭.

3.3 复制粘贴控制的实现

AutoCAD 程序通过 Windows 剪切板来实现图形对象的复制粘贴操作, 各图档之间可以快捷方便的传递信息. 但是使用复制粘贴功能, 可能会导致重要图纸信息的泄露. 本文使用 API HOOK 技术控制复制粘贴操作, 可以有效的防止用户泄露图纸信息. 自定义一个与目标函数结构相同的中间函数, 用来判断用户是否有复制粘贴权限, 如果有, 取消对操作的拦截, 执行原函数, 否则就什么也不做, 直接返回一个 NULL. 使用 Detours 库中 DetourAttach() 函数将目标函数用中间函数替换, 即可实现复制粘贴操作的控制. 但是在拦截开始之前还需要一些初始化工作, 首先初始化一个 detour 事物, 更新和事物相关的线程, 接下来开始执行目标 API 函数的拦截, 最后使 detour 生效. 复制粘贴操作监控流程如图 4 所示.

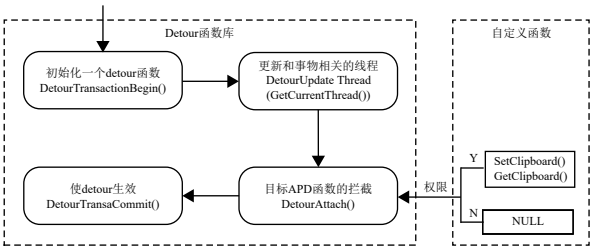

图 4 复制粘贴操作监控流程

自定义拦截函数代码如下:

```
//定义 SetClipboardData() 函数的替换函数
HANDLE WINAPI NEW_SetClipboardData(UINT uFormat, HANDLE hMem)
{ if(此用户有复制粘贴权限)
return SetClipboardData(uFormat, hMem);
//有权限用户不拦截, 将数据存入剪切板
return NULL;// 无权限用户, 将操作拦截
}
//定义 GetClipboardData() 函数的替换函数
HANDLE WINAPI NEW_GetClipboardData(UINT uFormat)
{ if(此用户有复制粘贴权限)
return GetClipboardData(uFormat);
//有权限用户不拦截, 将数据从剪切板中取出
return NULL; //无权限用户, 将操作拦截
}
```

3.4 截屏控制的实现

本文实现了对键盘的 PrtSc 键和常用软件的截屏控制, 当有图纸打开且用户不具备截屏权限时, 则截屏失败.

根据上文介绍的钩子函数使用方法, 首先建立键盘钩子过程函数, 用来拦截 PrtSc 键. 当有消息产生时, 先经过钩子函数, 判断是否是 PrtSc 和 Alt+PrtSc 键被按下, 如果是就截获消息不再向目标窗口传递, 其他消息将继续向下传递. 然后使用 SetWindowsHookEx() 函数将已定义的钩子过程安装到钩子链表中, 并指定钩子类型为底层键盘钩子, 用来拦截键盘截屏消息. 当用户打开图纸文件时, 通过判断用户的权限来决定是否安装钩子过程, 无权限的用户按下 PrtSc 或 Alt+PrtSc 键时, 按键失效. 在图纸被关闭前, 调用 UnhookWindowsHookEx() 函数将钩子卸载, 释放系统的资源.

关键实现部分的代码如下所示:

```
LRESULT CALLBACK KeyboardProc(int nCode, WPARAM wParam, LPARAM lParam)
{ BOOL fEatKeystroke=FALSE;
    if(nCode==HC_ACTION) //判断参数 wParam 和 lParam 中是否包含了键盘消息的信息
{switch (wParam)
  { case WM_KEYDOWN:
    case WM_SYSKEYDOWN:
    case WM_KEYUP:
    case WM_SYSKEYUP:
    //判断按下的键盘键是否为系统热键
PKBDLLHOOKSTRUCT p = (PKBDLLHOOKSTRUCT)lParam;
fEatKeystroke=((p->vkCode==VK_SNAPSHOT)||((p->vkCode== VK_SNAPSHOT)
&&((p->flags&LLKHF_ALTDOWN)!=0)));
//判断 PrtSc 键和 Alt+PrtSc 键是否被按下, 按下返回 TRUE, 否则返回 FALSE
        break;}}
    return (fEatKeystroke?1:CallNextHookEx(g_hKeyboard, nCode, wParam, lParam));
    //PrtSc 和 Alt+PrtSc 键被按下则消息不再传
```

递实现拦截, 否则将消息继续传递

}

通过对软件截屏的工作原理和实际应用条件的研究, 本文通过设置定时器来周期性清空剪切板来拦截常用软件的截屏操作. 使用定时器来周期性执行清空剪切板操作, 并根据需要自定义时间周期为 5 秒. 当不具备截图权限的用户打开图纸后, 使用 SetTimer() 函数安装定时器, 开始拦截截图软件, 关闭图纸后用 KillTimer() 函数取消定时器, 恢复软件的截图功能.

4 功能测试与分析

打开 AutoCAD 应用程序, 将所开发的.arx 程序加载到 AutoCAD 中, 开始测试所拦截的命令是否失效. 在加载.arx 程序前后, 使用打印命令时 AutoCAD 界面效果对比如图 5 所示.

图 5 打印 (PLOT) 命令拦截效果图

图 5 左侧是正常使用打印时弹出的打印对话框, 右侧是执行拦截后控制台输出提示并弹出提示框, 对比效果图可以证明打印命令被拦截而失效. 程序对于保存命令的拦截具有同样的效果, 拦截后将只弹出消息和提示框, 而不再有保存对话框. 使用键盘截屏键和日常工作中常用软件 (例如 QQ、微信和 360 浏览器) 的截图后, 右键选择粘贴命令时, 命令为灰色不可使用, 证明截图失效. 图 6 为对 AutoCAD 复制粘贴命令拦截前后的使用效果对比图. 图 6 左侧是复制圆形对象后执行粘贴操作的效果图, 右侧是执行拦截操作后粘贴时的效果图. 对比两幅图可以发现, 拦截后再执行粘贴操作, 圆形对象已被清空, 证明复制粘贴操作已失效.

5 结语

随着电子图纸的广泛使用, 工程图纸的安全问题受到更广泛的关注. 本文设计并实现了一个基于 ObjectARX 的图纸操作安全管理系统, 主要是保护已打开图纸的信息安全. 本系统利用 ObjectARX 技术对 AutoCAD 进行二次开发, 主要使用 ARX 反应器类拦截 AutoCAD 的打印和保存消息, 钩子技术实现复制粘贴和截屏控制. 实际运行结果表明, 本系统根据用户操作权限既可以限制用户对 AutoCAD 的操作, 又可以很好的控制 PrtSc 截屏键和常用截图软件工具的使用, 达到了图纸操作安全管理的目的.

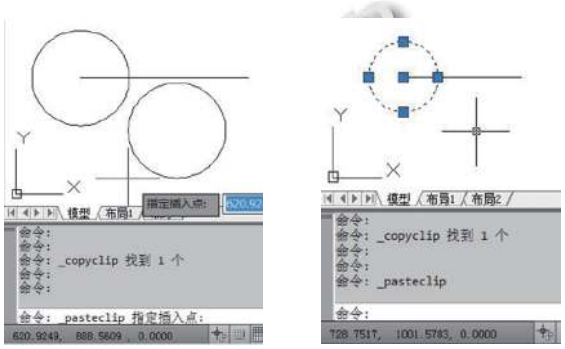

图 6 复制粘贴操作拦截效果图

参考文献

1 夏奎龙. 基于 ObjectARX 的工程图纸安全保护系统研究[硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2008.

2 王毅. 矢量 CAD 电子图纸保护系统研究[硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2008.

3 王永辉, 胡青泥, 李红彩. AutoCAD 二次开发方法的研究. 计算机系统应用, 2007, (3): 94–96, 100. [doi: 10.3969/j.issn.1003-3254.2007.03.024]

4 杨超琪, 刘静华, 李士才. ObjectARX 反应器在工厂设计中的应用. 科技风, 2011, (21): 79–80, 82. [doi: 10.3969/j.issn.1671-7341.2011.21.078]

5 Shen JF, Cheng LL, Fu XF. Implementation of program behavior anomaly detection and protection using hook technology. Proceedings of 2009 WRI International Conference on Communications and Mobile Computing. Yunnan, China. 2009. 338–342.

6 Hunt G, Brubacher D. Detours: Binary interception of Win 32 functions. Proceedings of the 3rd Conference on USENIX Windows NT Symposium. Berkeley, CA, USA. 1999.

7 Okolica J, Peterson GL. Extracting the windows clipboard from physical memory. Digital Investigation, 2011, 8(S1): S118–S124. [doi: 10.1016/j.diin.2011.05.014]

8 张云潮. 怎样屏蔽系统热键. 电脑编程技巧与维护, 2001, (8): 29–30. [doi: 10.3969/j.issn.1006-4052.2001.08.008]

9 刘胜达, 舒杰. 截图软件设计与实现的新方法. 哈尔滨理工大学学报, 2009, 14(05): 60-62,67.