

基于 LabVIEW 的软件许可证系统设计

王会清^{1,2}, 程 勇^{1,2}

(1. 武汉工程大学智能机器人湖北省重点实验室, 湖北 武汉 430074;

2. 武汉工程大学计算机科学与工程学院, 湖北 武汉 430074)

摘 要:针对目前的软件许可保护技术还存在着一定的缺陷和不足, 本文设计了一种基于 LabVIEW 开发的软件许可保护系统, 通过将注册表、许可文件、使用天数和使用次数等紧密结合来实施软件许可认证管理. 实际运行表明, 该系统能够有效地提供软件版权保护.

关键词:LabVIEW; 软件保护; 加密算法

中图分类号:TP311.53

文献标识码:A

doi:10.3969/j.issn.1674-2869.2011.04.021

0 引 言

根据行业统计, 盗版软件的使用造成全球软件开发商每年收入的损失高达 290 亿美金, 仅美国就损失了 60 亿美金. 在东欧, 正在使用的软件有约 71% 是盗版软件. 在一些亚洲和东欧国家, 超过 90% 的软件都是盗版^[1]. 软件的破解技术和保护技术二者之间是矛与盾的关系^[2-3]. 当前软件保护技术根据是否需要依赖特定硬件分为硬加密技术和软加密技术^[4]. 硬件加密技术虽然可靠性高, 但需要硬件支持, 成本会有所提高. 目前常用的硬件加密技术有软盘加密、光盘加密、卡加密、加密狗加密; 软加密技术主要使用软件算法来验证使用者是否合法性, 如常见的序列号、许可文件, 这种软加密技术成本相对较低.

通过序列号和许可文件来保护软件使用的许可技术是目前共享软件和部分商业软件常用的保护技术. 用户通过运行软件时所提供的序列号, 来向软件开发商购买注册号, 但这种加密技术容易被破解, 甚至可以通过修改电脑的系统时间来延长提供给用户的免费使用期限. 本文提出了一种软件许可保护系统, 有效地解决了上述这些不足之处.

1 软件许可保护系统所用开发工具及算法

1.1 系统开发工具

该软件许可系统是利用 LabVIEW 开发平台开发. 对于采用 LabVIEW 语言开发的软件, 可以直接

将软件许可保护系统当成一个子 VI 来调用即可. 而对于利用文本语言开发的软件, 如 C, VC 等可以通过 .dll 或 .ocx 控件直接调用, 使用非常方便^[5].

LabVIEW 是一种用图标代替文本行创建应用程序的图形化编程语言. 它采用数据流编程方式, 程序框图中节点之间的数据流向决定了程序的执行顺序, 以图标表示函数, 用连线表示数据流^[6]. 对于开发非底层程序非常简捷, 同时还提供了调用第三方语言的接口, 如 CIN 节点、DLL 节点、ACTIVEX 控件、.net、MATLAB 脚本节点等技术.

1.2 软件许可保护系统算法

本系统在充分利用目前常用的软件许可保护系统的优点基础上, 针对其破解比较容易以及可以通过修改电脑系统时间来破解软件的试用期限等不足, 把注册表、许可文件、使用天数和使用次数几个方面结合起来加以改进, 避免了复杂的加密算法. 用户使用软件开发商的软件时, 首先调用本软件许可程序. 许可程序先检测注册表项和许可文件是否存在, 若存在, 再检测是否被非法修改, 如果一切正常就将运行的天数和使用次数写入注册表和许可文件, 再采用一般的加密算法, 从而达到对软件版权的保护目的. 这样即使通过更改电脑系统时间来蒙蔽许可系统也是不可能的, 因为许可系统会实时比较本次运行的时间是否大于上次已经写入注册表和许可文件的使用时间. 如果用户非法调整电脑系统时间, 许可系统会立即检测到本次使用时间小于上次使用时间, 软件系统立即退出. 其算法如图 1 所示.

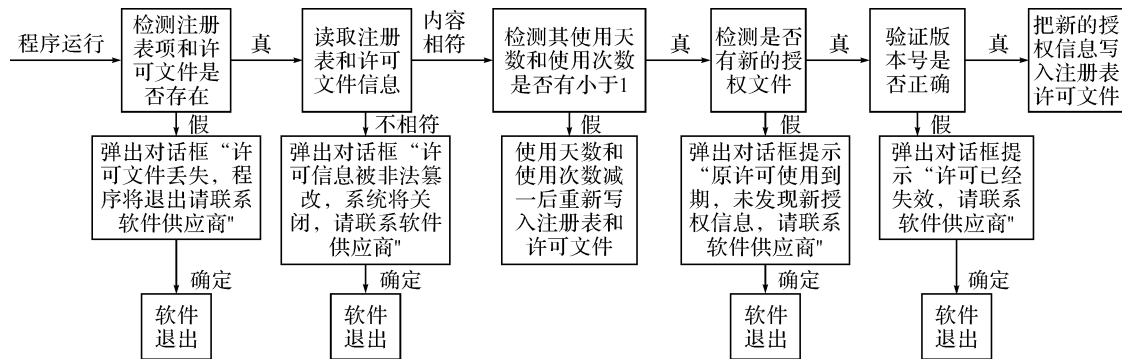


图 1 许可保护系统算法

Fig. 1 The algorithm of the license protection system

2 软件许可系统的程序设计

软件许可系统程序运行时, 首先检测许可文件、注册表项是否存在以及注册表项数是否符合。三个条件有一个不成立, 将弹出一个对话框, 提示“许可文件或注册表项存在问题, 请联系本软件供应商”, 点击“确定”后系统退出。只有在三个条件都为真的情况下, 才会读取注册表和许可文件信

息, 程序如图 2 所示。程序中的 DATA 路径子 VI 的程序如图 3 所示。这里之所以把路径独自写成一个 VI 是由于 LabVIEW 开发平台有一个支持文件路径的问题, 即打包成 EXE 文件后, 其路径会比原来的路径多一个以应用程序名为目录的路径, 为此需先写了一个路径 VI 程序, 这样在使用过程中如果用到支持文件, 需要使用相对路径时, 可直接调用此 VI, 这也为大的项目开发带来方便。

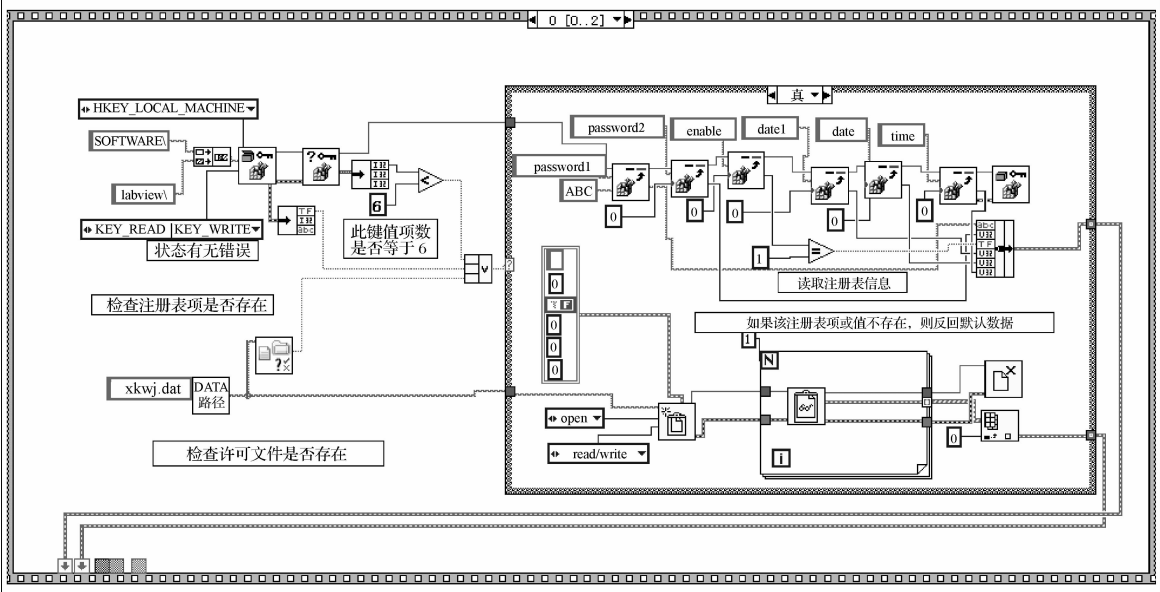


图 2 检测和读取许可文件与注册表项信息

Fig. 2 Detect and read the information of the license file and the registry

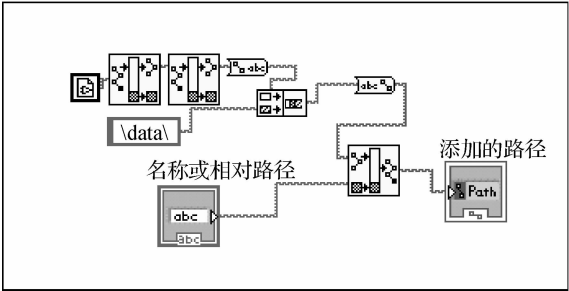


图 3 路径调用 VI

Fig. 3 The VI for calling path

经读取注册表和许可文件数据后进入图 4 所示程序, 检查是否满足正常运行的三个条件: 即注册表

内容和许可文件的内容是否一致; 注册表项中软件第一次运行时所写入的经加密算法得出的结果值与当前注册表所对应的值是否相等, 这里加密算法采用的是许可文件的版本号与 password1 密码字符串中相应字符值的乘积为结果值; 以及电脑上的系统时间是否大于上次使用时间。其中把注册表内容与许可文件内容相比较, 以及注册表中软件第一次运行时所写入的经加密算法得出的结果值与现在注册表里的对应的值是否相等, 其目的是确定注册表是否被更改; 核查电脑时间是否大于上次使用时间的作用是防止非法更改电脑系统时间来延长试用期限。

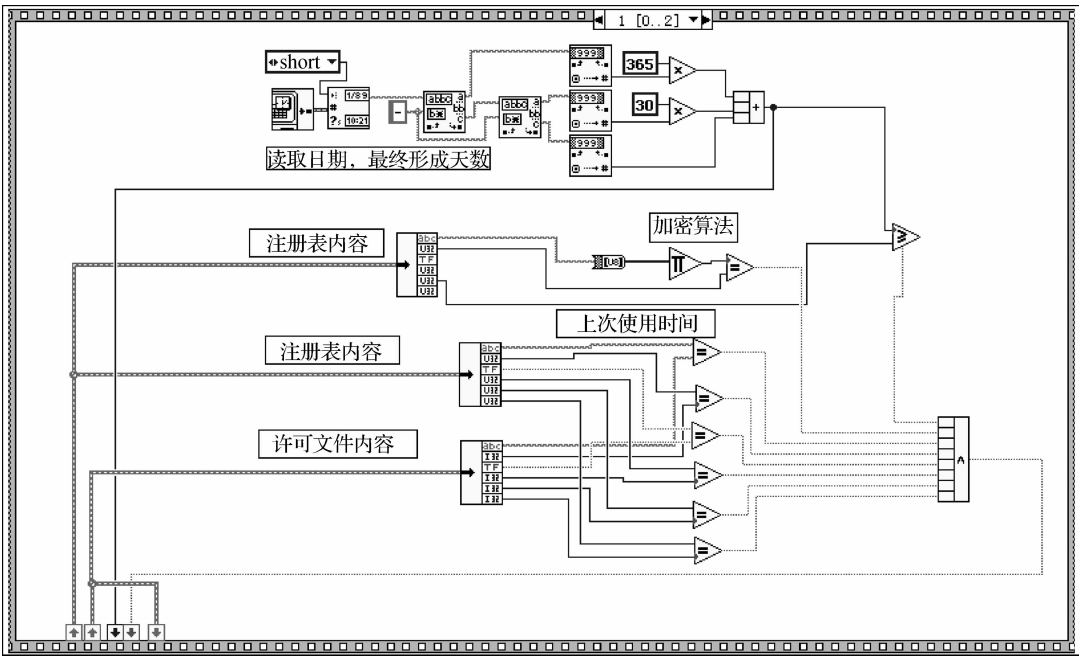


图 4 比较判断三个条件程序

Fig. 4 The program of comparing and determining three conditions

第一次运行时,注册表内容和许可文件内容是相符的,因为第一次安装时自动把许可文件信息动态的导入到注册表中,这样用户在以后的使用过程中,使用的天数和次数会动态地改变注册表和许可文件中相应的数据. 在线写入注册表,是为了防止非法用户将安装好的客户端系统备份后继续使用. 如果在写入注册表的同时还写入许可文件,这样即使备份了系统,恢复时,注册表信息和备份的许可文件内容也是不一致的. 只有所比较的信息都一一相对应后,才能进入条件结构为真的程序设计. 接着检查许可文件信息里的使用权限是否不可继续使用,所允许使用的天数和次数是否小于 1,这三个条件其中有一个成立,则检测有无新的授权文件存在. 若存

在,读取新的授权文件中新的授权信息,首先看其版本号是否大于上次的许可文件的版本号,大于就把新的授权信息写入注册表和许可文件,用户可继续使用;否则,自动弹出一个对话框,提示“许可文件已经失效,请联系系统软件供应商”. 当没有检测到新的授权文件时,也会自动弹出对话框,提示用户“原许可文件已经到期”,点击“确定”后,退出. 如果上述条件都符合要求,还可以继续使用,就进入图 5 程序,把原来的许可文件剩余次数和剩余天数都减 1 后,重新写入到注册表和许可文件. 若在图 4 的程序中比较的注册表项和许可文件内容所对应内容有一个不对应,会弹出一对话框,提示“注册表信息不对应或被修改”,点击“确定”按钮,退出.

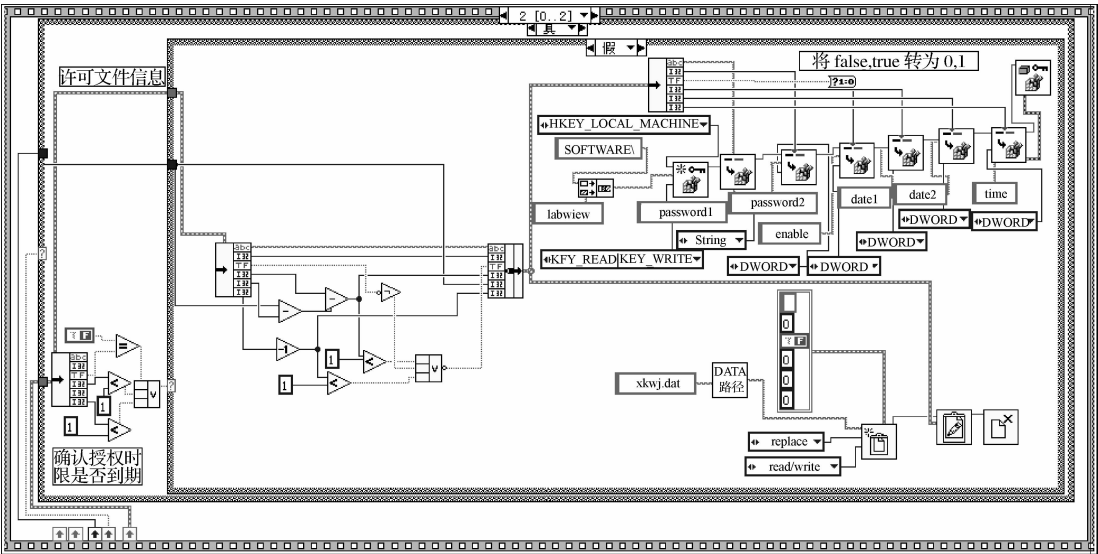


图 5 更改原来授权信息并写入许可文件与注册表

Fig. 5 Change the original authorization information and write in the license file and the registry

图 2 程序中所用到的许可文件 xkwj. dat 及新的授权文件里面的内容生成程序如图 6 所示. 程序采用的加密算法是所输入的版本号与 password1 密码字符串中相应字符值的乘积为结果值,同时把是否允许使用及允许使用的天数、次

数和当前的时间写入许可文件中. 如果为了更安全的许可保护,可以使用更复杂的加密算法. 考虑到注册表项、许可文件、使用天数和使用次数四方面的紧密结合,这种版本号与密码字符串中相应字符值的乘积方法足以保证安全.

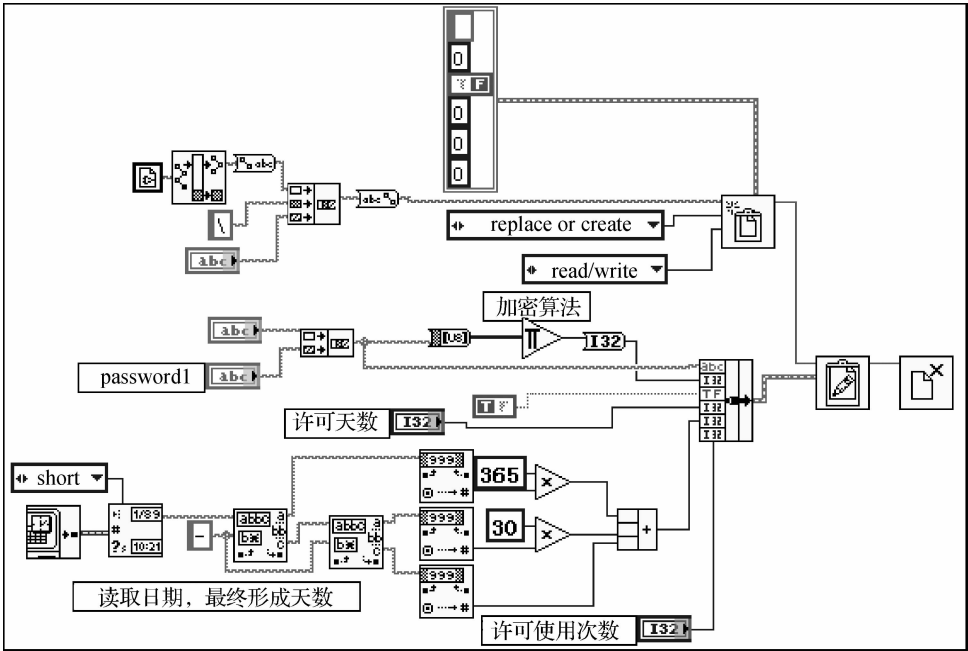


图 6 许可文件和新的授权文件数据生成程序

Fig. 6 The program of generating the data of the License file and the new authorization file

3 结 语

综上所述,解决了目前软件许可保护技术的一个常见的漏洞,即更改电脑系统时间来延长试用期限,包括 NI 公司开发的 LabVIEW 以及常用的数据库管理软件等. 本软件许可系统利用 LabVIEW 中获取时间函数来计算时间功能,然后转换成天数实时存入到注册表和许可文件,防止非法改变软件的时间限制;并针对目前常见的只有序列号或注册表配合加密保护,而容易被破解的问题,采用紧密结合注册表,许可文件,使用天数和使用次数四个方面来提高加密安全度,使得目前常用的破解方法对本软件许可保护系统进行破解都是无效的. 实际运行表明,该系统能够有效地保护软件提供商的利益,维护软件产业的正常发展.

参考文献:

[1] 陈泓. 破解软件保护五大误区[J]. 计算机安全, 2009 (11): 89 - 90.

[2] 刘佳娜, 张林龙, 钱松荣. 基于 ACE 的软件许可证系统设计和实现[D]. 上海: 复旦大学, 2006.

[3] 何成万, 焦素廷, 李健. 支持协同工作的加解密系统设计与实现[J]. 武汉工程大学学报, 2009, 31(3): 75 - 76.

[4] 陈向荣. 软件保护的软硬件方案研究与实现[D]. 武汉: 华中科技大学, 2002.

[5] 刘军, 陈冈强, 王会清. KVM 系统中远程虚拟存储的研究[J]. 武汉工程大学学报, 2010, 32(7): 85 - 88.

[6] 陈锡辉, 张银鸿. LabVIEW 程序设计从入门到精通 [M]. 北京: 清华大学出版社, 2007.

(下转第 93 页)