

# 基于 CPU 序列号和加密技术的软件注册机

何定华

武汉职业技术学院计算机学院, 武汉 430074

**摘要** 注册机是一种用于生成注册码的软件,本文采用 CPU 序列号来制作注册机。为了提高安全性,本文还对 CPU 序列号进行加密处理,CPU 序列号中不同位置的字符采用不同的加密算法。文中还列举了一基于 CPU 序列号和加密技术的注册机实例,实例证明这种基于 CPU 序列号和加密技术的注册机对软件加密是行之有效的。

**关键词** CPU 序列号;加密;解密;注册机;注册码

**中图分类号** TP311.52

**文献标识码** A

**doi** 10.3981/j.issn.1000-7857.2012.02.005

## Keygen Based on CPU Serial Number and Encryption Technique

HE Dinghua

Department of Computer, Wuhan Polytechnic, Wuhan 430074, China

**Abstract** The keygen is a software used to generate a registration code. In this paper, the serial number of CPU is used to construct the keygen. In order to improve the security level, in this paper, the serial number of CPU is encrypted, with its characters in different locations being encrypted by different algorithms. A keygen based on CPU serial number and encryption technique is given to show the effectiveness of the method.

**Keywords** CPU serial number; encryption; decryption; keygen; registration code

### 0 引言

计算机软件是一种特殊的产品,容易被复制,为了防止软件的非法复制、盗版,保护软件开发商的利益,就必须对软件进行加密保护。常用的软件加密方法为使用固定的注册号(又称序列号),但是这种方法的缺点是,一旦该序列号泄露并在网上传播,将导致该软件被无偿使用,势必大大损害软件开发商的利益。

一种比较好的软件加密方法是使用注册机。注册机是用于生成注册码的小型软件,为了保证不同机器的注册码与其他机器不相同,很多软件的注册码和计算机硬件信息(如硬件的序列号等独一无二的信息)相关,这样便可以做到一机一码,注册机在不同机器上生成的注册码拿到另外一台机器上是无法注册的。一般将注册机和应用程序分离保存。应用软件安装后,若软件未注册则在运行时自动出现注册窗口,并根据本机硬件特征信息产生注册申请码,把该注册申请码输入注册机,注册机自动算出所要破解的注册信息,即注册

序列号或注册码。

CPU 序列号是十六进制 24 位,网卡的物理地址是十六进制 12 位。将 CPU 序列号和网卡的物理地址加密后仍然为 24 位和 12 位,只是每位不再是十六进制的数字。若注册码的每位出现的字符有 100 种可能性(实际上不止 100 种可能性),则 24 位注册码的组合数是 12 位注册码组合数的  $100^{12}$  倍,显然 24 位的注册码比 12 位的注册码破解起来更困难。此外 CPU 是计算机必备的硬件设备,因此本文使用 CPU 序列号并进行加密来生成注册机,这种基于 CPU 序列号和加密技术的注册机更安全。

### 1 基于 CPU 序列号和加密技术的注册机

#### 1.1 设计思想

应用软件运行后自动检测注册表特定位置是否存在 CPU 序列号信息,如果存在则表示该应用软件已注册,是合法用户;若注册表特定位置中不存在 CPU 序列号信息,则表

收稿日期:2011-02-28;修回日期:2011-12-05

作者简介:何定华,副教授,研究方向为管理信息系统和数据安全,电子信箱:hedinghua@qq.com

示该应用软件尚未注册,如不注册,则用户不能正常使用应用软件,或者不能使用正版软件的全部功能(试用版)。注册时,应用软件首先获取 CPU 序列号,并对 CPU 序列号进行加密 1 生成注册申请码。将注册申请码输入注册机,通过解密还原成应用软件所在机器的 CPU 序列号,再把该 CPU 序列号加密 1 成注册码。在应用软件中输入注册码后,应用软件解密 1 注册码,并把解密得到的字符串和 CPU 序列号进行比较:如果相等,则注册码合法有效,应用软件将机器指纹写入注册表中特定位置,完成注册过程;如果不相等,则注册码无效,不将 CPU 序列号写入注册表,注册失败。

本文使用 Delphi 开发平台实现。利用 CPU 序列号和加密技术制作软件注册机的工作流程见图 1。需要说明的是,本文中的加密 1 和解密 1 是互逆的,加密 2 和解密 2 也是互逆的,使用两个加密过程和两个解密过程目的是为了保证注册申请码和注册号不相同,且在任何时候不明文显示 CPU 序列号。

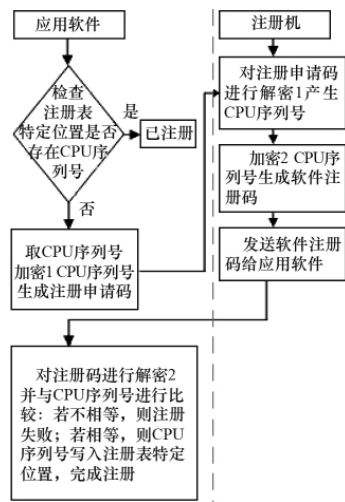


图 1 工作流程图  
Fig. 1 Work flowchart

## 1.2 实现过程

### 1.2.1 注册表操作

Delphi 中提供了 Registry 单元,该单元中的 TRegistry 对象提供一些属性和方法用于注册表操作,主要包括以下几步。

- (1) 创建注册表对象

Var

```
reg: TRegistry 类型;
reg ← TRegistry.Create;
```

- (2) 设置注册表根结点

```
reg.RootKey ← 注册表根键;
```

- (3) 打开注册键

Var

```
st:string;
reg.OpenKey(pathstring, true|false);
//打开注册表结点
```

```
//pathstring 是注册表路径
//true 表示若不存在则新建该结点
//false 表示若不存在则不新建该结点
```

- (4) 读注册表

```
st ← reg.ReadString('cpu_id');
```

```
//读注册表信息(cpu_id)赋值给字符串 st
```

- (5) 写注册表

```
reg.WriteString('cpu_id',hd_id);
```

```
//写注册表
```

```
//即写 cpu_id 到“根键 athstringpu_id”
```

### 1.2.2 获取 CPU 序列号

可以从“Delphi 园地”或“Delphi 盒子”等网站下载第三方控件 ComputerId 并安装该控件。在新建的应用软件注册窗口中添加该控件,通过下面算法获取 CPU 序列号:

Var

```
cpu_id:字符串变量;
```

```
cpu_id ← computerId1.cpuserialnumber;
```

或者使用包中的动态链接库文件 ComputerId.dll 提供的 GetCPUSerialNumber 方法函数获得 CPU 序列号。

### 1.2.3 加密原理

将一个字符 ch 加密的方法是:取该字符的序号,将该序号加某个常数之后,再通过 chr 函数返回该数字对应的字符,即完成字符的加密过程。例如将字符‘A’取序号之后再加 1,再通过 chr 函数可以将该数字转换为‘B’字符,因此字符‘A’加密之后变成了字符‘B’。解密过程与加密过程相反:取一个字符的序号,再将该序号减某个常数,然后再通过 chr 函数得到该数字对应的字符,即完成解密过程。例如取字符‘B’的序号,减 1 之后通过 chr 函数得到字符‘A’,即‘B’字符解密变成‘A’字符。

字符串的加密、解密,只需要对字符串中每个字符进行加密或者解密即可。为了加大解密的难度,可以将字符串中不同位置的字符加不同的常数。即:密文=f(明文,位置),明文=f<sup>-1</sup>(密文,位置),见文献[2]—[3]。

### 1.2.4 加密算法

下面给出加密 1 算法:

```
function Encrypt1(st:string):string;
```

var

```
i:整型变量;
s:字符串变量;
```

begin

```
s ← 空字符串;
```

```
l ← 字符串 st 的长度;
```

```
从 1 到 l 执行:
```

```
s ← s+chr(ord(st[i])+C-i);
```

```
Encrypt1 ← s;//得到密文
```

end;

解密 1 的算法只需要将加密 1 中的语句 s ← s+chr(ord(st

[i])+C-i);

修改成

$s \leftarrow s + \text{chr}(\text{ord}(\text{st}[i]) - C + i);$

即可。

加密 2 (Encrypt2)、解密 2 (Decrypt2) 的算法与加密 1 (Encrypt1)、解密 1 (Decrypt1) 的算法类似,只需要改变 C 的大小即可。加密 1 和解密 1 是一对的互逆过程,加密 2 和解密 2 是另一对互逆的过程。

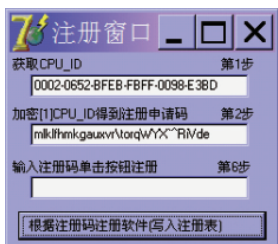


图2 注册窗口生成注册申请码  
Fig. 2 Generate the registration application code

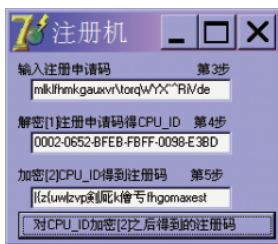


图3 输入注册申请码生成注册码  
Fig. 3 Input an application code to generate registration code

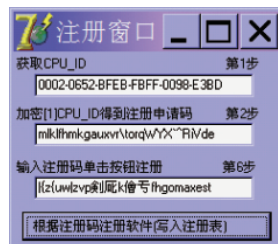


图4 输入注册码成功注册  
Fig. 4 Input registration code and register successfully

文中应用软件机器 CPU 序列号为“0002-0652-BFEB-FBFF-0098-E3BD”,可以不显示在注册窗口和注册机,本文把 CPU 序列号显示出来仅为演示方便和易于理解。

根据注册码注册软件的过程就是把注册码写入注册表特定位置的过程。注册表操作参照 1.2.1 小节,此处不再赘述。

应用软件成功注册之后,在注册表特定位置会保存注册码。应用软件在运行时取出注册表中的注册码,执行解密 1 过程,再取计算机 CPU 序列号,比较二者,若相同则说明此前已经正常注册,此时显示软件功能界面;否则,说明此前输入的注册码不正确,此时不能进入软件功能界面。

## 2 分析

使用注册机,将应用软件和注册机分离开。软件使用者仅拥有应用软件,软件开发商保存注册机,最大限度地保证了软件的安全性。文中采用的加密算法虽然不如 RSA 加密、MD5 加密、DES 加密复杂,但是由于每个字符的加密算法不相同,且与该字符在字符串中的位置有关,即:“密文=f(明文,位置)”和“明文=f<sup>-1</sup>(密文,位置)”比单一的“密文=f(明文)”和“明文=f<sup>-1</sup>(密文)”更科学,破解难度更大,且非常容易理解和实现。此外,本加密方法不会产生 ASCII 低于 32 的不显示字符,这些 ASCII 低于 32 的不显示字符,有的还是控制字符,例如响铃、回车、换行、退格等字符,不利于注册申请码和注册码的复制、粘贴、输入和邮件发送。

## 3 结论与改进思路

### 3.1 结论

笔者在自己开发的远程在线考试系统中使用基于 CPU 序列号和加密技术的注册机,效果非常好,根据前面的分析以及笔者的应用证明:这种基于 CPU 序列号和加密技术的注

### 1.3 验证实例

为了达到一定的演示效果,本文编写了一个应用软件和注册机。1.2 节中给出了关键代码和设计思想,此处省去实例代码。图 2 为注册前应用软件显示的注册窗口,并自动生成注册申请码的过程。图 3 是在注册机中输入注册申请码生成注册码的过程。图 4 为在应用软件注册窗口中输入注册码成功注册的过程。成功注册之后,运行应用软件不再显示注册窗口。

册机对软件保密是行之有效和难以破解的。

### 3.2 改进思路

软件使用者可以通过邮件等方式将注册申请码发送给软件开发者。软件开发者在注册机中输入根据注册申请码算出注册码,再通过邮件发送给软件使用者。当然,应用软件(使用者)和注册机(软件开发商)之间还可以使用分布式服务的方法来实现应用软件和注册机之间的通信:在应用软件端添加控件 SocketConnection 和 ClientDataSet;在注册机(注册服务器)上添加控件 Query 和 DataSetProvider,并建立与注册信息有关的表文件,科学地设置表的字段可以极大地增强注册机的功能。然后设置控件的相关属性,编写相应的代码,即可实现应用软件和注册机之间的网络通信<sup>[4]</sup>,具体操作过程可参阅与 Delphi 和多层分布式相关的参考资料。前一种方式一般应用于软件产品的小规模量发行,后一种方式通过客户端(应用软件)、服务器(注册机)的方式进行网络通信,更有利于软件产品的大规模量发行。

### 参考文献 (References)

[1] 何定华. Delphi 中利用 MAC 地址实现软件注册 [J]. 微计算机信息, 2011, 27(2): 246-248.  
He Dinghua. *Microcomputer Information*, 2011, 27(2): 246-248.

[2] 武新华, 张惠娟, 李秋菊. 加密解密全攻略[M]. 3 版. 北京: 中国铁道出版社, 2010.  
Wu Xinhua, Zhang Huijuan, Li Qiuju. *Encryption and decryption raiders* [M]. 3rd ed. Beijing: China Railway Publishing House, 2010.

[3] 段钢. 加密与解密[M]. 3 版. 北京: 电子工业出版社, 2008.  
Duan Gang. *Encryption and decryption* [M]. 3rd ed. Beijing: Publishing House of Electronics Industry, 2008.

[4] 何定华, 万彪. 基于 MIDAS 的软件远程注册原理及其实现[J]. 微计算机信息, 2011, 27(1): 301-303.  
He Dinghua, Wan Biao. *Microcomputer Information*, 2011, 27(1): 301-303.

(责任编辑 马宇红,代丽)