



软件狗

开发商手册

软件版本 4.1

部件号 007-011512-002

修订版B

感谢您购买并使用赛孚耐（北京）信息技术有限公司的软件保护产品。

在使用本产品前，请仔细阅读下面的许可协议。

同意本许可协议的所有条款及此处包含的任何补充或特殊的许可条款是获得本产品许可的必要条件。如果您不同意此协议的所有条款，请在三天内将产品退还赛孚耐（北京）信息技术有限公司或其代理商。您对本软件的使用将表明您同意接受本协议中条款的约束。

许可协议

授予使用许可权

您可以为了备份的目的而复制光盘中的软件，可以为了保护您的软件的目的而根据本产品的文档说明将我们提供的软件合并进您的程序中。

禁止使用

除按上述“**授予使用许可权**”条款被授权的行为外，不可以复制、修改、逆向工程、分解或重组该产品的全部或部分，不可向他人销售、租借、许可、转让、分发全部或部分本产品或本协议授予的权利。

有限担保

保证在自产品交给您之日起的 12 个月内，在正常使用情况下，产品不会出现实质性的质量问题和生产设计上的缺陷。自您购买产品之日起的 12 个月内，赛孚耐（北京）信息技术有限公司的全部责任和您能获得的全部补救措施为：可选择退货、更换或修理；购买产品之日起的 12 个月以后，您能获得的全部补救措施为：更换或修理。

除了上述对本产品的原始购买者所提供的有限保证之外，不向任何人作任何其它的保证。对赛孚耐（北京）信息技术有限公司的产品、性能或服务亦没有明示的或暗示的或其它任何形式的保证，包括但不限于商品的适销性和对特定用途的适用性。

任何情况下，无论如何引起及依据何种责任理论，均不负担任何因使用或不能使用本产品造成的损失责任，包括：由于使用本产品或本产品的功能失效导致丢失数据、损失利润及其它特别的、偶然的、附随的、继发的或间接的损失。

所有权

所有的产品，包括加密狗、软件、文档、与本产品一并附送的其它材料及您制作的备份的所有权与版权均属于赛孚耐（北京）信息技术有限公司。

协议的终止

违反上述条款时，本协议的授权将自动终止。

“狗”是北京金天地软件发展有限公司的注册商标，并已授权赛孚耐（北京）信息技术有限公司使用。

本文所涉及的其它产品和公司名称可能是各自相应所有者的商标。

赛孚耐（北京）信息技术有限公司

东城区北三环东路 36 号环球贸易中心 D 栋 15 楼 08-12 单元 (100013)

电话：86-10-57810666

传真：86-10-57810688

销售咨询电话：800-810-0804

网址：<http://cn.safenet-inc.com>

Email：support@safenet-inc.com

目 录

关于本手册	1
名词解释	3
第一章 产品简介	5
软件狗的基本原理.....	5
软件狗如何保护您的软件?	6
软件狗的特点及优势.....	6
如何订购开发套件、加密件?	7
第二章 软件狗的安装与卸载	8
产品清单	8
安装平台	8
开发套件的安装和卸载.....	8
驱动程序的安装和卸载.....	10
安装常见问题解答.....	10
第三章 开发商工具的使用	11
存储区编辑	11
批量导入	13
绑定升级	15
错误码查询	17

第四章 外壳加密工具.....	18
外壳加密工具的特点.....	18
应用示例	19
保护时的错误代码.....	32
支持的文件类型、编译程序.....	33
注意事项	35
第五章 使用API接口函数完成加密.....	39
全局变量说明	39
函数说明	40
如何保护WINDOWS 程序	40
例子程序	41
第六章 更好地使用软件狗开发套件	44
简单技巧	44
中级技巧	47
高级技巧	48
实现其他功能	49
第七章 封装发行您的软件	50
第八章 附录	51

关于本手册

本手册是软件狗开发套件产品的开发商手册。下面这部分内容将帮助您快速熟悉本手册的内容，更好地了解并使用软件狗开发套件。

- * 阅读一下“[名词解释](#)”，会使您更容易阅读本手册。
- * “[产品简介](#)”这一章概括地介绍了软件狗的原理、主要功能和特点。
- * 将安装光盘放入光驱后，安装程序会自动运行。按照安装程序界面上的提示信息完成安装，详细内容请参看“[软件狗的安装与卸载](#)”。
- * 使用开发商工具DogEdt32.EXE可完成对软件狗内信息的编辑修改、软件狗内信息的读出、写入等操作。开发商工具在软件狗开发套件安装目录\UTILITY\DogEdit\DogEdt32.EXE。详细内容请参看“[开发商工具的使用](#)”。
- * 使用外壳加密工具，您无需编写任何代码就可快捷地完成对EXE等文件的加密，并有着较高的加密强度。Windows程序的外壳加密工具为软件狗开发套件安装目录\UTILITY\SHELL下的WIN32SHL.EXE；详细内容请参看“[外壳加密工具](#)”。
- * 调用API函数进行加密，可以灵活地设置加密点，隐蔽性较好，具有很高的加密强度。对于外壳加密工具无法加密的特殊程序也可以通过这种内嵌方式完成加密。详细内容请参看“[使用API接口函数完成加密](#)”。
- * 在“[更好地使用软件狗开发套件](#)”一章中我们还为您提供简单级、中级和高级加密技巧，参看我们提供的例子程序您会很快学会使用API函数实现不同级别强度的加密。
- * 发行您的软件时，软件狗的驱动程序可以直接使用软件狗开发套件提供的安装程序。在软件狗开发套件安装目录\DRIVER目录下的SOFTDOGINSTDRV.EXE文件，可直接提供给最终客户；也可编程调用软件狗开发套件安装目录\DRIVER\SOFTDOGSETUP.DLL来安装软件狗的驱动程序。详细内容请参看“[封装发行您的软件](#)”。

手册图例

禁止提示信息



需要注意的重要信息



名词解释

加密件

加密件是指软件狗开发套件中的硬件，即 PDM 型并口软件狗或 UDA 型 USB 软件狗，我们亦将并口软件狗和 USB 软件狗统称为软件狗。

软件狗开发套件

软件狗开发套件包括一张安装光盘（所有加密工具及语言接口模块）、一个加密件、快速入门、装有 Dog Key 的信封。

开发商

使用软件狗开发套件对自己的软件产品进行保护的软件开发商。

Dog Key

Dog Key 是八组字符，每组包含 5 个。开发商在第一次安装开发套件时需要输入 Dog Key，对接口模块进行绑定。

在套装包装盒里有一个白色的信封，里面装有一张印有 Dog Key 的纸。利用开发商工具（DogEdt32.EXE）可以实现 Dog Key 和 Number.Dog 文件（此文件下面会有说明）之间的转换。在第二次安装开发套件时，可以使用 Number.Dog 文件，免去了输入 Dog Key 的麻烦。**请注意保管 Dog Key 文件。**

系列号

开发套件中的软件狗外壳上都印有一串数字（Serial No. xxxxxxxx），我们称之为系列号。一个软件狗开发套件对应一个唯一的系列号。购买开发套件后，开发商可以向我公司或代理商购买开发套件所对应的系列号的加密件（即同号狗）。

同号狗

系列号相同的软件狗。

绑定

在开发套件安装过程中，安装程序会对加密模块和相应的软件狗进行绑定操作。绑定过的模块只能操作对应系列号的软件狗，不能操作其他系列号的软件狗；未经绑定的模块不能使用。您可以使用开发商工具（DogEdt32.exe）对加密模块和相应的软件狗进行绑定。

级联

级联是指多个并口狗在并口上串联。软件狗与我公司的其他并口加密产品或其它公司开发的符合规范的并口加密产品可以无障碍级联。级联的个数不受限制，但受到并口工作电压及其他电气参数的影响（软件狗通过并口获取工作电压），一般计算机上可支持3个并口狗级联。

Number.Dog 文件

Number.Dog 文件包括软件狗开发套件系列号及订购加密件时要提供的信息。利用开发商工具 DogEdt32.EXE 可以将 Dog Key 文件转换成 Number.Dog 文件。开发商在订购加密件或同号狗时要提供 Number.Dog 文件。**请注意保管 Number.Dog 文件。**

存储区

软件狗开发套件为开发商提供了 100 字节的存储区。一般情况下，存储区 0-99 字节为读写区，如果开发商使用高强度动态库，则只能使用前 0-83 字节存储区作为读写区使用，最后 16 个字节为认证密钥。开发商可通过开发商工具（DogEdt32.EXE）或接口函数对存储区进行读写操作和认证密钥的设置。

第二代安全加密引擎

该引擎集成了最新的密码学理论和高强度加密算法，采用了动态认证和迷宫反跟踪技术，可以极大地提高受保护软件的防破解能力。

第一章 产品简介

软件狗是使用在计算机并行口或 USB 口上的用于软件保护的硬件产品，具有 100 个字节的数据存储区。软件狗开发套件包括一只并口软件狗或 USB 软件狗和用于加密的工具软件。软件开发商可以采用多种方法保护软件，防止软件被非法拷贝使用。



软件狗开发套件 4.1 由 4.0 升级而来，增强了安全性，并且只支持固件版本大于等于 5.0 的 USB 狗。该版本不支持固件版本小于 5.0 的 USB 狗。并口狗没有该限制。

软件狗开发套件由以下组件组成：

1. 软件狗

软件狗是指安装在并口上的 PDM 型并口软件狗或安装在 USB 口上的 UDA 型 USB 软件狗。软件狗具有 100 个字节的数据存储区。如果软件狗插在计算机上，您可以通过相应的加密接口函数或开发商工具对软件狗进行访问。

2. 加密接口

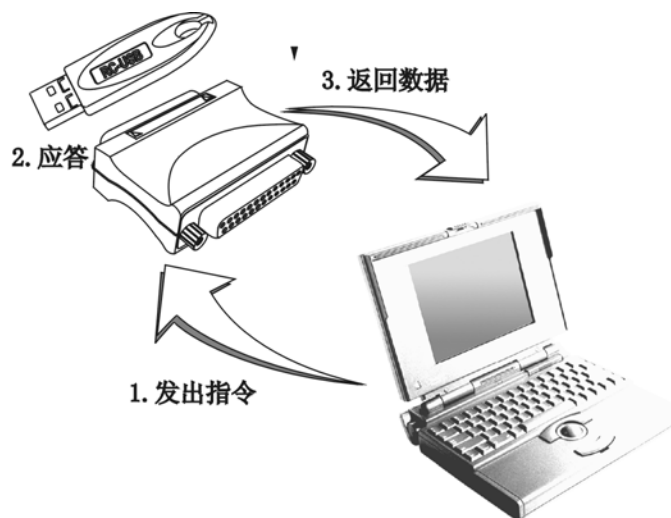
软件狗开发套件提供的加密接口是一套包含各种开发语言的程序接口模块，可以嵌在程序的源代码中。您可以在您的程序中使用软件狗开发套件提供的加密接口对软件狗进行操作，详细内容请参看“使用 API 接口函数完成加密”一章。

3. 开发商工具（DogEdt32.EXE）

软件狗开发商工具可以使开发商方便地对软件狗存储区进行编辑、读取序列号以及连续初始化等操作，详细内容请参看“开发商工具的使用”一章。

软件狗的基本原理

开发商的应用程序通过调用软件狗开发套件提供的接口模块对软件狗进行操作，软件狗响应该操作并通过接口模块将相应数据返回给应用程序。开发商程序可以对返回值进行判定并采取相应的动作。如果返回无效的响应，表明没有正确的软件狗，开发商可以终止运行应用程序。简要示意如下：



软件狗如何保护您的软件？

软件狗开发套件提供了两种加密方案：使用 API 或者直接使用外壳工具，请根据您的需求选择方案。

加密方案一：使用 API

软件狗 API 提供了操作软件狗的接口函数，您可以直接在应用程序的源代码里加入这些接口函数的调用，来保护您的一个或者多个软件，然后重新编译您的应用程序。

由您来设置应用程序中调用加密接口的次数以及没有发现软件狗采取的措施。原则上您调用、设置的加密接口越多将更有利于阻止潜在黑客破坏您的软件保护。

加密方案二：使用外壳工具

使用外壳工具对软件保护是一种快速、简单的保护方案。它不像使用接口函数需要对源代码进行改动，而是自动给您的可执行应用程序加了一个保护层。在程序开始运行和运行当中会自动地访问软件狗是否存在。假如软件狗不存在，应用程序将不能运行。

软件狗的特点及优势

软件狗开发套件适用于 Windows 2K（32 位）、Windows XP（32/64 位）、Windows Server 2003（32/64 位）、Windows Vista（32/64 位）、Windows Server 2008（32/64 位）、Windows Server 2008 R2（64 位）、Windows 7（32/64 位）操作系统，支持几乎所有主流编程语言、开发工具。软件狗的主要特点如下：

- **数据交换随机噪声技术：**有效地对抗逻辑分析仪及各种调试工具的攻击，完全禁止软件仿真程序模拟并口或 USB 口的数据。

- **迷宫技术：**在软件狗函数入口和出口之间包含大量复杂的判断跳转干扰代码，动态改变执行次序，提升软件狗的抗跟踪能力。
- **AS 技术：**API 函数调用与 SHELL 外壳加密相结合的方式，同时使用能够达到极高的加密强度。软件狗开发套件在外壳加密工具中与调用的 API 函数建立了对应关系。这样处理后，程序中调用的 API 函数只有在有外壳的情况下才能正确运行，而外壳本身隐蔽了对 API 函数的调用。
- **存储器：**提供 100 字节掉电保持数据存储区供开发商存放关键数据、配置参数等信息，可通过软件狗开发商工具（DogEdt32.EXE）或接口函数对存储区进行读写。
- **级联：**支持与我公司或其他公司开发的不同类型的并口加密狗的级联。

如何订购开发套件、加密件？

开发商可以选择订购软件狗开发套件或加密件。

加密件是指软件狗开发套件中的硬件，即 PDM 型并口软件狗、UDA 型 USB 软件狗。

在购买的开发套件包装中有一个信封，里面装有 Dog Key 文件。利用开发商工具 DogEdt32.EXE 可以将 Dog Key 转换成 Number.Dog 文件。订购加密件时需要提供 Number.Dog 文件。

一个开发套件只提供一个系列号。开发商在购买开发套件并完成加密工作后，要订购与开发套件相同系列号的软件狗，即订购加密件。如果要保护另一个单独发行的软件，建议开发商购买另一个开发套件，即订购开发套件。



软件开发商订购加密件时需提供特征码文件 Number.Dog。

特征码文件 Number.Dog 还可用于软件狗加密模块的升级。

请妥善保管此文件。

第二章 软件狗的安装与卸载

产品清单

软件狗开发套件分为并口型套件和 USB 型套件：

包含内容	产品型号	
	软件狗 并口型套件	软件狗 USB 型套件
并口狗（PDM 型）1 只	√	
USB 狗（UDA 型）1 只		√
安装光盘（所有加密工具及语言接口模块）	√	√
快速入门	√	√
装有 DogKey 文件的信封	√	√

安装平台

Windows 2K[®]（32 位）、Windows XP[®]（32/64 位）、Windows Server 2003[®]（32/64 位）、Windows Vista[®]（32/64 位）、Windows Server 2008[®]（32/64 位）、Windows Server 2008 R2[®]（64 位）及 Windows 7[®]（32/64 位）操作系统。

开发套件的安装和卸载

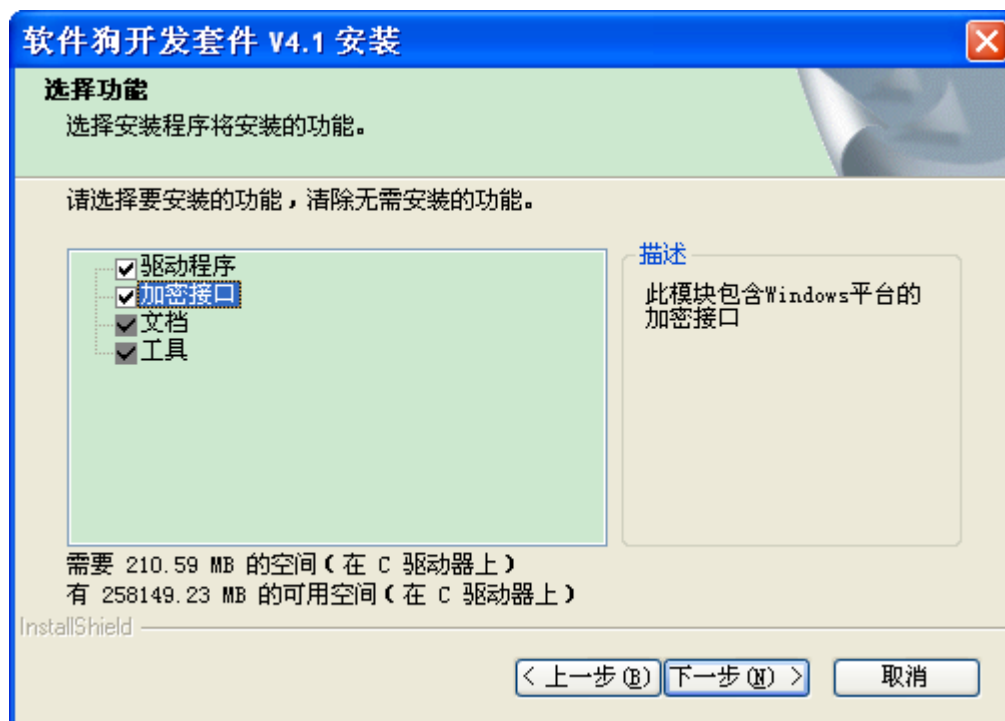
开发套件的安装

在安装软件狗开发套件前，请您退出所有 Windows 程序，特别是一些杀毒软件。请将安装光盘放入光盘驱动器，安装程序将自动运行。如果安装程序没有自动运行，请手工运行安装盘上的 SETUP.EXE，按照提示进行即可完成安装。软件狗开发套件的安装主要包括软件狗驱动程序的安装、各种语言模块的安装、模块与软件狗的绑定三部分：

- 软件狗驱动程序的安装

开发套件安装过程中会自动判断系统中是否安装了驱动程序或安装了旧版本的驱动程序，如果系统中没有安装驱动程序或安装了旧版本的驱动程序，将会自动安装驱动程序。

- 各种语言模块的安装



- 模块与软件狗的绑定



使用 Dog Key 或 Number.Dog 文件对软件狗进行绑定，未经绑定的模块不能使用。

开发套件的卸载

单击“开始”按钮，依次选择“设置”、“控制面板”，单击“添加/删除程序”图标。在已安装程序列表中选择“软件狗开发套件 V4.1”后单击“添加/删除”按钮。

驱动程序的安装和卸载

驱动程序的安装

开发商在安装开发套件的时候将会自动安装驱动程序，如果发现相同版本或者更新版本的驱动，将会提示用户是否继续安装驱动。同时我们在软件狗安装目录\DRIVER 下还提供了单独的驱动安装程序。

在 Windows 下的安装狗驱动程序

运行软件狗开发套件安装光盘中 Driver 目录下的安装程序 (SoftDogInstdrv.exe)，驱动安装程序会自动检测系统平台及是否已安装了最新的软件狗驱动程序。如果驱动已经安装可以不用再次安装，需要注意的是**如果使用 USB 软件狗，在安装驱动之前，最好不要插入 USB 软件狗，在驱动安装完毕以后，再插入 USB 软件狗，否则可能造成 USB 软件狗驱动的安装不正确。**对于并口软件狗不存在此问题。

驱动程序的卸载

在 Windows 下的卸载狗驱动程序

运行软件狗开发套件安装目录\Driver\SoftDogInstdrv.exe，在弹出的界面中单击“卸载”按钮。

安装常见问题解答

- 在 Windows 2000/XP/Server2003/Vista/2008 Server/ Windows 7/2008 Server R2 操作系统下安装时应以具有系统管理员权限的身份登录。
- 在开发套件安装过程中，如果驱动程序安装不正确，请您运行安装光盘中 Driver 目录下的 SoftDogInstdrv.exe，将驱动程序卸载后再重新安装。待驱动程序安装成功后再安装开发套件。

第三章 开发商工具的使用

单击工具栏上的“开始”按钮，依次选择“程序”、“软件狗开发套件 V4.1”、“开发商工具”，启动开发商工具。也可直接运行位于 UTILITY 子目录下的 DogEdt32.EXE 文件启动开发商工具。

开发商工具提供了以下四个功能：

存储区编辑

提供了对基本存储区数据的编辑功能、读狗、写狗、读取狗的系列号、设置认证密钥因子以供高强度动态库使用等功能。

批量导入

开发商可以使用功能来成批处理同系列号的软件狗。

绑定升级

提供了 Number.Dog 文件和 Dog Key 之间互相转换、绑定升级模块的功能。

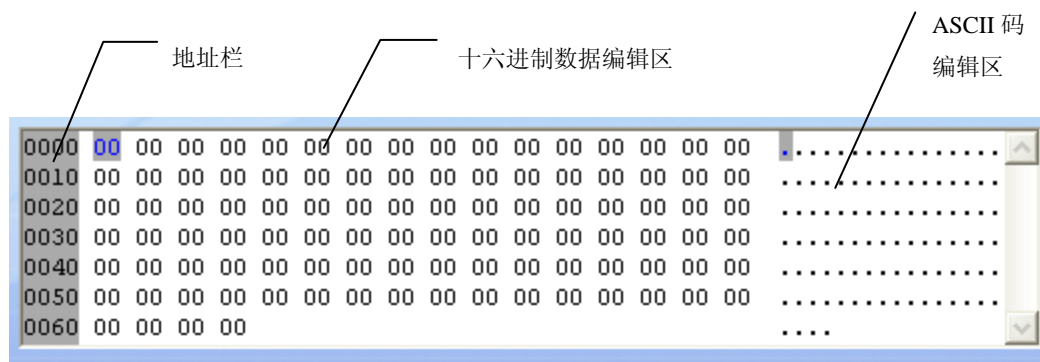
错误码查询

开发商可以使用此属性页的功能快速方便的查找错误码。

下面对每个功能进行详细地说明。

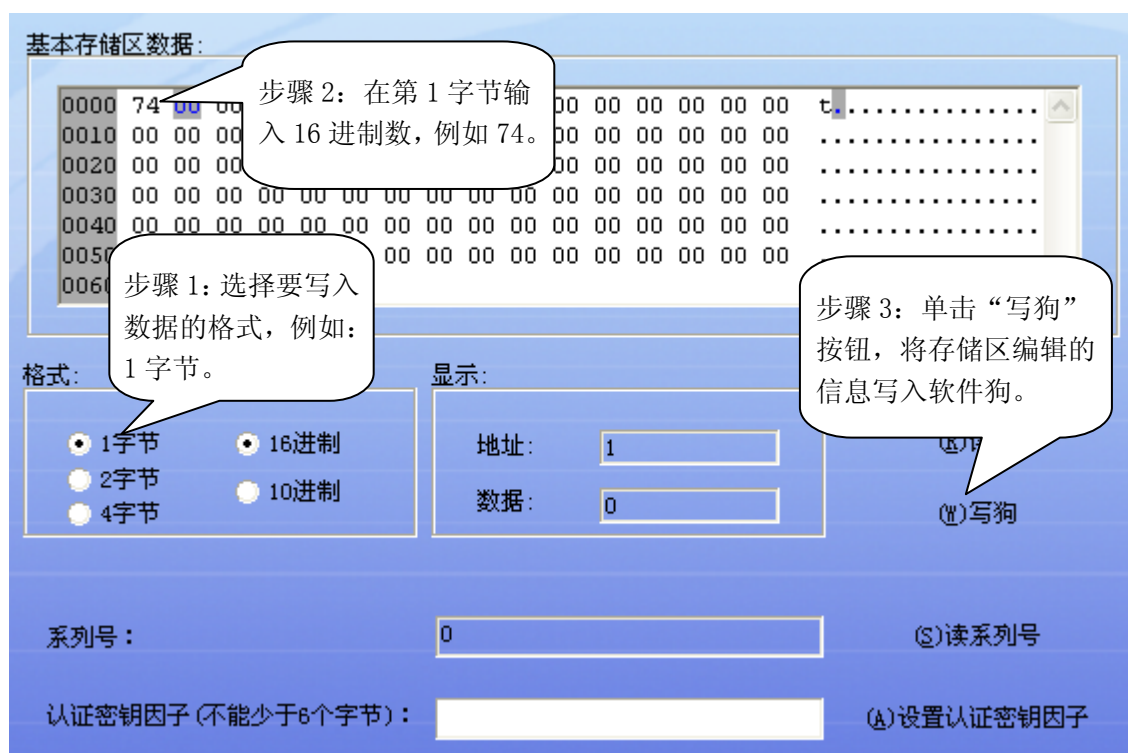
存储区编辑

在对存储区编辑之前，首先需要了解存储区的分布。



对软件狗内的 100 字节存储区进行读写

请按下图中的步骤 1，2，3 编辑存储区的信息。



取出软件狗的系列号

单击“读系列号”按钮，可以取出软件狗的系列号。

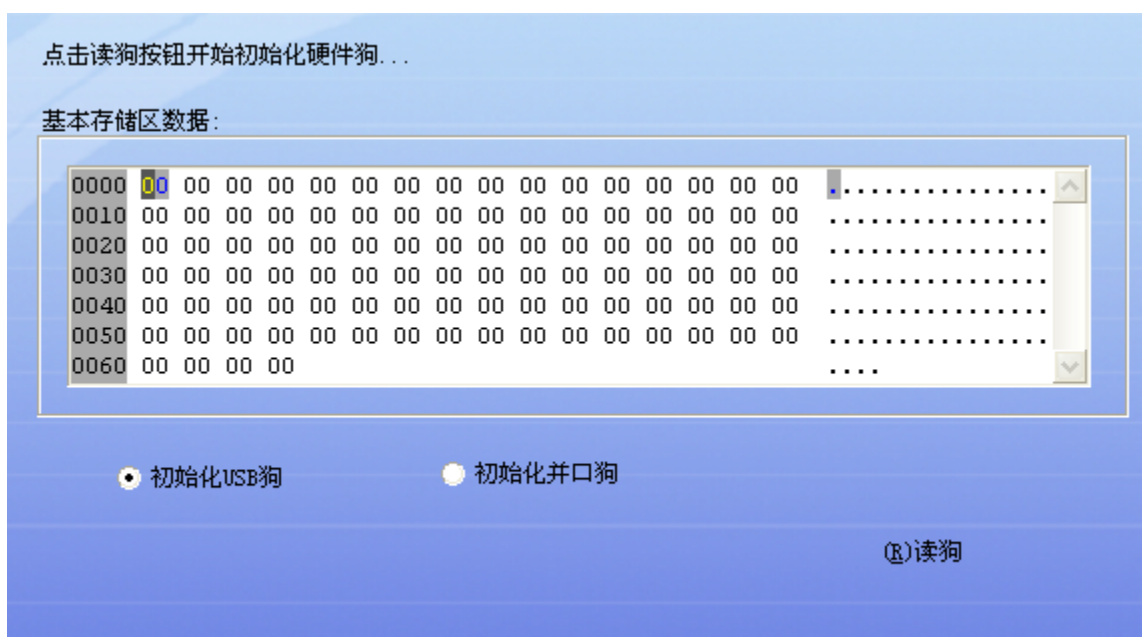
设置认证密钥因子

如果开发商使用高强度动态库，可在“认证密钥因子”文本框中输入认证密钥因子，单击“设置认证密钥因子”按钮，生成认证密钥。认证密钥占用存储区的最后 16 个字节，即存放在存储区的第 84 字节到第 99 字节。如果开发商没有使用高强度动态库可以不用设置此项。

配置文件

文件菜单中的**保存配置文件**和**打开配置文件**功能可以将存储区的内容保存到配置文件中，还可以在需要时读出，写入软件狗中。

批量导入



开发商可以使用此功能来成批地处理同系列号的软件狗。

开发商可以先对一只软件狗进行存储区设置，然后使用此软件狗对其它软件狗进行批量处理。

开发商插入已完成存储区设置的第一只软件狗，选择要初始化的软件狗的类型（并口软件狗或 USB 软件狗），然后点击“读狗”按钮。拔出第一只软件狗，出现如下图所示的对话框：



插入另一只要初始化的软件狗，单击“初始化”按钮，开始初始化操作。初始化完成时会出现提示信息，拔出软件狗，插入下一只要初始化的软件狗。重复以上操作来成批地处理同系列号的软件狗。

绑定升级



绑定升级模块

开发商如果需要升级自己的加密模块，可以到我们的网站下载最新的加密模块。由于网上的加密模块是被加密的，开发商下载以后必须使用此工具将加密模块和软件狗绑定在一起，即使加密模块和软件狗一一对应起来。

注意：待绑定的加密模块属性不能是只读的。

开发商可以使用 Number.dog 文件或者 Dog Key 来对升级模块进行升级绑定。

1. 转换狗信息

开发商可以任意从 Dog Key 转换成 Number.dog 文件，也可以从 Number.dog 文件转换成 Dog Key。

Number.dog 文件存储有软件狗的重要信息，开发商如果需要订购同号的软件狗，必须向我们的代理或者赛孚耐（北京）信息技术有限公司提供 Number.dog 文件。

将 Number.Dog 文件转换成 Dog Key

步骤一：选中“Number.Dog 转换成 Dog Key”单选按钮；

步骤二：在“Number.Dog 文件”文本框中输入 Number.Dog 文件所在的位置及文件名；也可单击“浏览”按钮指定 Number.Dog 文件所在位置；

步骤三：单击“操作”按钮，在“Dog Key”文本框中将显示 Dog Key。

将 Dog Key 转换成 Number.Dog 文件

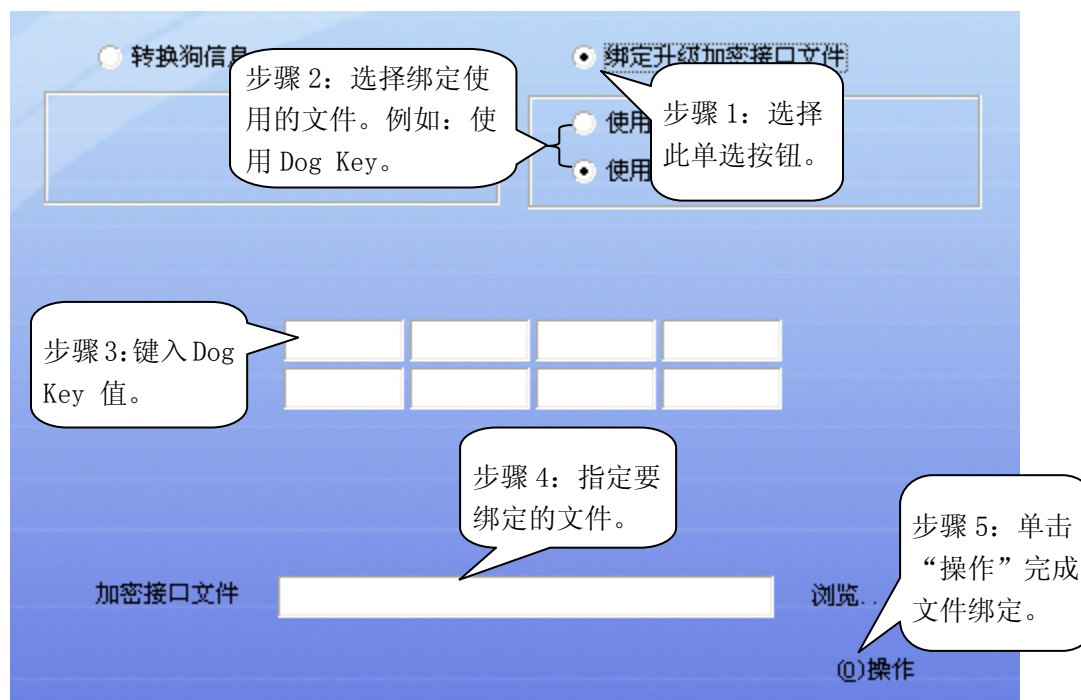
步骤一：选中“Dog Key 转换成 Number.Dog”单选按钮；

步骤二：在“Dog Key”文本框中输入 Dog Key；

步骤三：在“Number.Dog 文件”文本框中输入将生成的 Number.Dog 文件所保存的位置及文件名；也可单击“浏览”按钮指定 Number.Dog 文件所在位置。单击“操作”按钮，在指定位置生成 Number.Dog 文件。

2. 绑定升级加密接口文件

请按照下图中的步骤绑定升级加密接口文件：



错误码查询



请输入错误码：

20250 (S)查找

20250: 没有这个口;

开发商在文本框中输入接口函数返回的错误码，单击“查找”按钮即可获得错误码对应的错误信息，此功能便于开发商迅速解决问题。

第四章 外壳加密工具

使用外壳加密工具，开发商可以在没有程序源代码或不对程序源代码做任何改动的情况下快捷地完成加密工作，并保证有着较高的加密强度。若没有相应的硬件狗存在，加密后的程序将无法启动。

外壳加密工具的特点

- 1、使用三种方式加密：外壳、嵌入、内外结合。

外壳 所谓外壳，是加密工具加到被加密程序的一部分程序。加密程序启动时外壳首先被执行。外壳要检查硬件狗的存在性和一致性。如果对应的硬件狗存在，则进入原程序的入口，否则终止程序。外壳只在加密程序启动时执行一次。

嵌入 所谓嵌入，是加密工具在被加密程序中嵌入一个公共程序，加密程序在整个运行期内以一定的频率执行这个公共程序，该程序检查硬件狗的存在性和一致性。如果对应的硬件狗存在，则继续执行原程序，否则终止程序。本加密方式可选。

内外结合 如果被加密的程序已经调用了软件狗提供的接口函数，那么外壳程序会通知接口函数，外壳程序是否已经被执行，以防止外壳程序被跳过。如果发现外壳程序没有被执行，会终止程序。

- 2、可以对数据文件进行加密，这些数据文件由您的应用程序使用，或者由受保护的应用程序产生（输出文件）。这些文件只能由受保护的应用程序读取。
- 3、可以灵活地限制使用时间，一共有四种方式可以选择。如果选择了限制使用时间，程序将占用软件狗中 10 字节存储区，请指定存储区的首地址，并确保这部分存储区不被其它的应用程序改写。
 - A. 可以规定使用期限，如果使用期限已过，或者发现系统日期倒退，被加密的程序不能运行；
 - B. 可以规定使用次数，如果到达使用次数，被加密的程序不能运行；
 - C. 可以规定总的运行时间，如果程序的累计运行时间达到规定，被加密的程序不能运行；
 - D. 可以规定程序首次运行后的相对使用期限，如果使用期限已过，或者发现系统日期倒退，被加密的程序不能运行。
- 4、可以自定义查硬件狗错误时显示给用户的提示信息，也可以选择不显示提示信息。
- 5、外壳保护是一个多层的体系结构。只有前一层执行成功，才会解密紧相连的一层。这种多层体系结构可为您的应用程序提供额外的保护——这与用多道锁为您的门提供保护的功能相似。突破这些保护层需要更多资源、时间和技巧，可以有效的阻止黑客攻击。此外，由于采用了多层随机模式，所

以任何两次被外壳加密的应用程序均不相同。

外壳加密工具可以加密 32/64 位 Windows 程序，在软件狗安装目录下\Utility\Shell\Win32shl.exe。

应用示例

下面以使用一个软件狗保护 Notepad.exe 和 Calc.exe 为例说明外壳加密工具的使用。

1. 请先插入软件狗，执行步骤 1，如图 4-1 所示，指定要加密的 EXE 文件。



图 4-1

指定源文件后，系统会在“目标路径”文本框中自动添加加密后的 EXE 文件（即目标文件）存放的路径。系统默认将目标文件存放到 OUTPUT 路径下，我们建议用户采用这种默认方式。

如果目标文件存在，覆盖它

如果选中“如果目标文件已存在，覆盖它”复选框，当目标文件所在路径下有同名文件存在，加密过程

不再出现是否覆盖文件的提示信息。

2. 对加密狗的硬件参数进行设置。

校验数据

外壳工具可以选择是否校验存储区中的数据。如果选择校验数据，加密后程序运行时会从存储区中指定的位置读出数据，检查其是否正确。如果不正确，应用程序将不能运行。数据最长为 20 字节。

隐藏导入符号

选定此复选框隐藏您的程序中的导入符号。但是，请注意该选项不适用于此处所列出的文件类型：

- .NET
- Visual FoxPro
- Director
- Power Builder
- Adobe Acrobat Reader (PDF files)
- Windev
- Authorware
- Prolog
- GameMaker
- Excelsior JET
- Xenoage JESart
- MDM Zinc
- 使用 SmartHeap DLLs 的应用程序
- 当您同时加密数据文件或者使用数据文件加密选项时

注意 在加密过程中，当外壳加密工具检测到上述文件类型时，隐藏导入符号选项将会被忽略。

.NET 增强

.NET 增强功能为纯.NET 应用程序（EXE 和 DLL）提供了增强的安全性。此功能可执行以下操作：

- 隐藏原始入口点方法（仅用于.NET 可执行文件）
- 加密原始应用程序的字符串
- 加密原始应用程序的常量

如果选择了此功能，则需要将 SDNPro.dll（用于 32 位.NET 应用程序）和 SDNPro64.dll（用于 64 位.NET 应用程序）与受保护应用程序一起提供给最终用户。

注意 使用.NET 增强特征项（使用.Net 增强保护外壳.NET 应用程序）的前提条件是机器中应安装了同一版本的.NETFramework 和.NET SDK。.NET 增强选项不支持混合代码应用程序。

我使用了外壳 SDK

如果您已经使用外壳软件开发工具包（位于\Shell SDK 文件夹下）中的特性（代码混淆和/或运行时加密/解密）来保护您的重要代码段、常量和字符串数据，则选中此复选框。参阅该文件夹中的自述文件获得详细信息。

存在调试器时允许我的应用程序运行

您可以不选择该复选框，禁止在出现调试器时执行应用程序。如果在系统中探测到调试器，被保护的应用程序将退出。非恶意用户可以关闭调试器，重新启动应用程序。但是，如果出于某种原因，您需要在存在调试器的情况下允许您的应用程序运行，则选择该复选框。

嵌入

加密软件运行期间，可以不断地检查对应的硬件狗是否存在。如果清空“嵌入”复选框，则在程序运行期间不检查硬件狗。如果选中“嵌入”复选框，当程序运行期间检查不到对应的硬件狗时，程序会停止运行。您可以指定检查间隔，间隔以秒为单位。缺省值为 120 秒，即 2 分钟。

多层保护级别

外壳可提供多层保护。因为应用程序和外壳层之间的连接处非常容易受到攻击，所以您可以选择使用多层外壳（从第 1 级到第 5 级）来保护您的应用程序。第 1 级可提供合理的保护，而第 5 级可提供最强保护。但是，随着每一保护级别的增加，应用程序的大小和用于启动的时间也会相应增加。在缺省情况下，使用第 3 级保护方式。

3. 执行步骤 2、3、4、5、6，如图 4-2、4-3 所示，根据具体情况，选择不同方式设置使用期限。



图 4-2

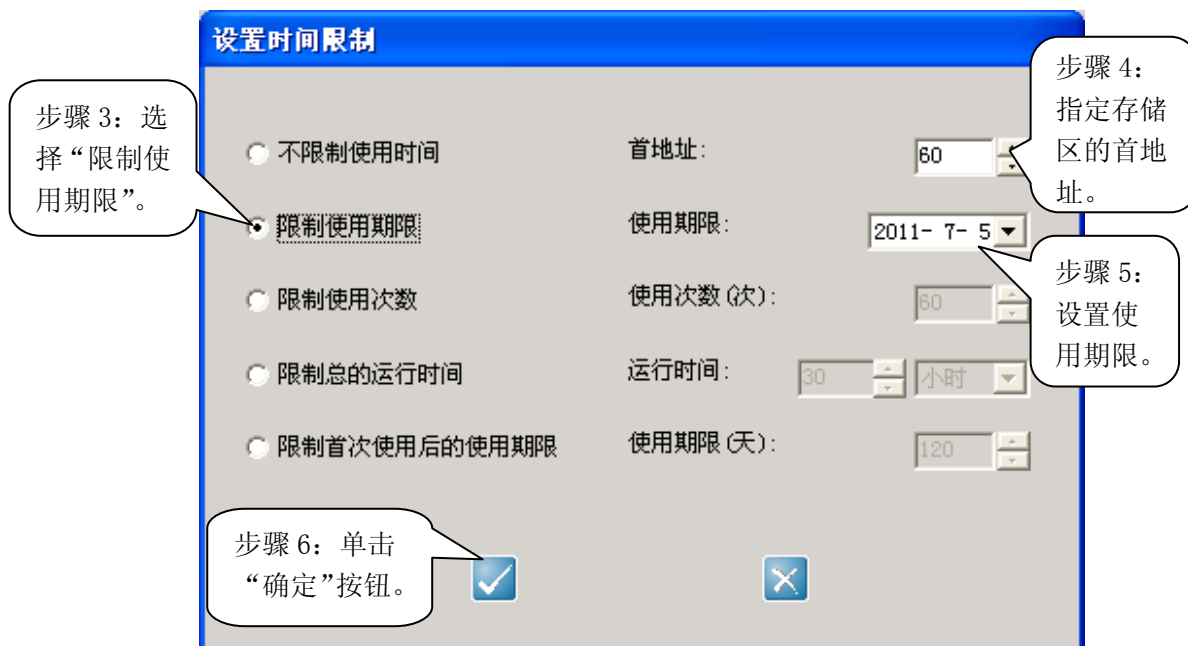


图 4-3

首地址

如果选择了限制使用时间，程序将占用软件狗中 10 字节存储区，请指定存储区的首地址，并确保这部分存储区不被其它的应用程序改写。

限制使用期限

可以设置使用期限，如果使用期限已过，或者发现系统日期倒退，被加密的程序不能运行。

限制使用次数

可以规定使用次数，如果到达使用次数，被加密的程序不能运行；使用次数最多为 60000 次。

限制总的运行时间

可以规定总的运行时间，如果程序的累计运行时间达到规定，被加密的程序不能运行；总的运行时间最多为 360 天。

限制首次使用后的使用期限

可以规定程序首次运行后的相对使用期限，如果使用期限已过，或者发现系统日期倒退，被加密的程序不能运行。此使用期限最多为 360 天。

注意

如果设置了“限制使用期限”，“限制首次使用后的使用期限”或者“限制总的运行时间”选项那么必须同时选定“嵌入”选择框。

加密完成后，您要对程序进行仔细测试，测试之前请先使用开发商工具（DogEdt32.EXE）初始化一只软件狗，因为运行被加密的程序可能会改变存储区指定区域的内容。

4. 我们提供了 18 种不同错误类型的错误信息，请根据具体情况分别指定其内容。标题限制在 60 字节内，每条消息限制在 100 字节内。执行步骤 7、8、9、10、11、12，如图 4-4、4-5 所示，设置提示消息。



图 4-4

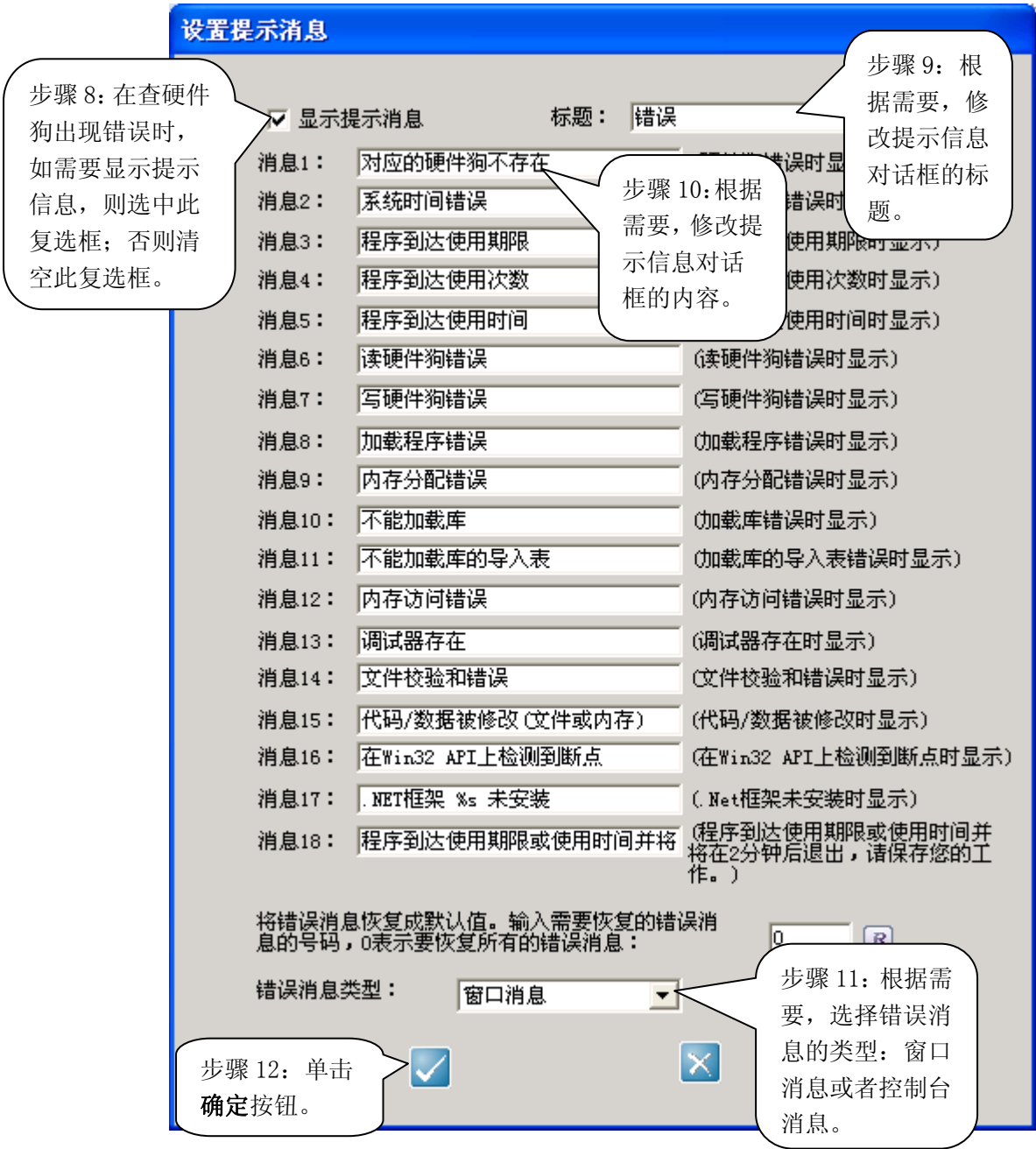


图 4-5

5. 如果在加密 EXE 文件的同时需要加密数据文件, 执行步骤 13, 如图 4-6 所示:



图 4-6

6. 指定要加密的数据文件（实际上 Notepad.exe 执行时不需要数据库文件，这里只是一个加密演示），执行步骤 14，如图 4-7 所示：



图4-7

7. 执行步骤 15，如图 4-8 所示，将数据文件加入文件列表。如需加入多个数据文件，请重复步骤 14 和 15。此时文件加密选项会根据文件列表进行自动设置，如果您要进一步对其进行设置，例如，当您加密新生成的数据文件（输出文件）时，执行步骤 16 更改文件加密选项。**注意：**在执行步骤 16 前，请插上狗。外壳工具会从狗里获得狗的 ID 作为默认的加密种子。

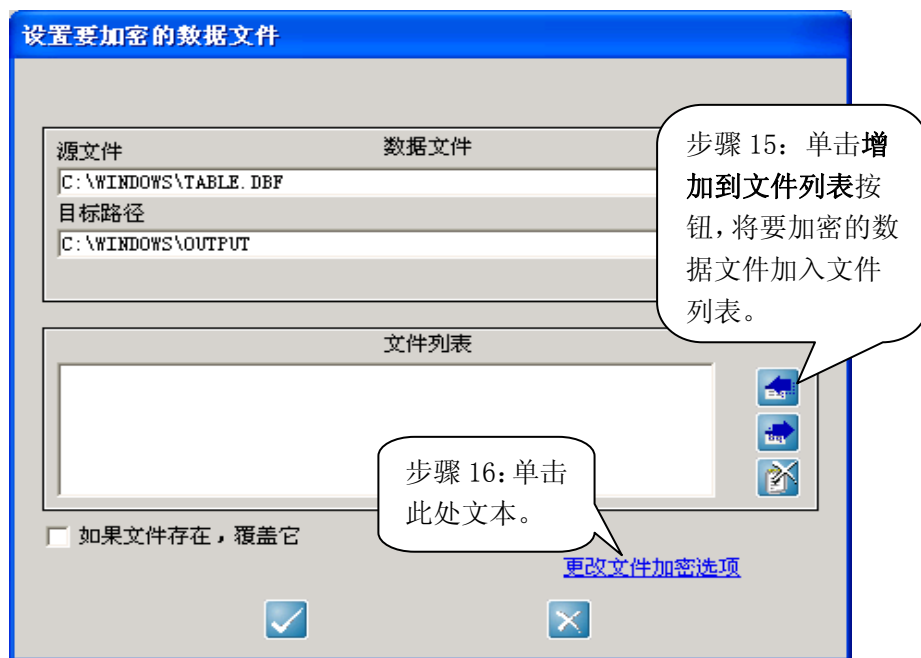


图4-8

8. 在文本编辑框中已经包含 “*.DBF;”，例如，为了加密受保护的程序在运行过程中生成的文本文件，

执行步骤 17，在文本编辑框中添加“*.TXT;”。如果想要指定自己的加密种子，执行步骤 18，选中“指定我自己的加密种子”复选框，并在文本编辑框中输入自定义的加密种子。另外，为了提高安全性，4.0 及更高版本的外壳工具默认会使用 AES 算法对数据文件进行加密，而以前的版本用的是私有算法。如果要保持跟以前版本的兼容性而使用私有算法，可以不选中“使用 AES 算法加密”复选框。执行步骤 20，返回“设置要加密的数据文件”对话框，单击确定按钮完成数据文件加密设置。

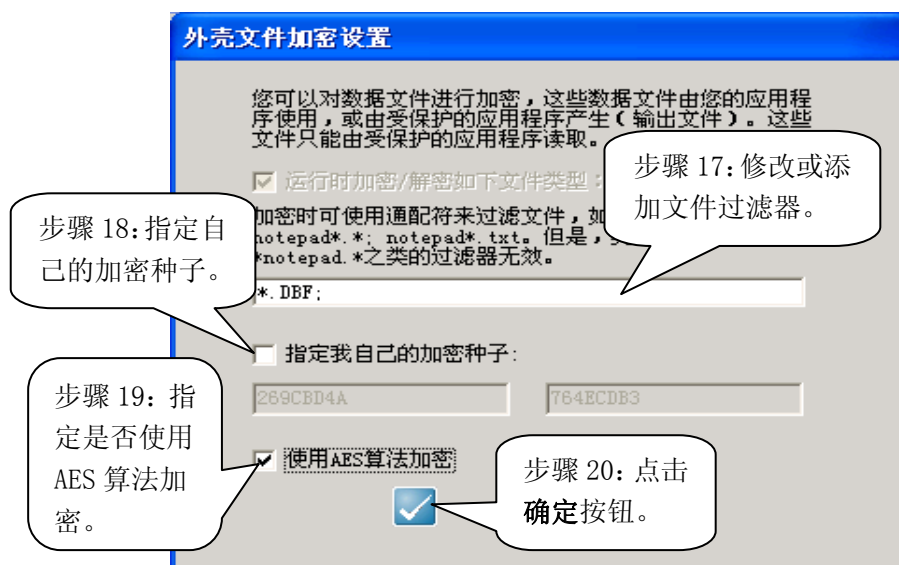


图4-9

9. 完成有关 Notepad.exe 的所有设置后执行步骤 21，如图 4-10 所示，将其加入加密中心，成为加密队列中的一个加密项。然后对 Calc.exe 重复步骤 1-21，如果需要限制使用时间，请修改“首地址”的值，如图 4-11 所示。使 Calc.exe 的首地址不同于 Notepad.exe 的设置。



图 4-10



图 4-11

10. 可以将加密中心所有加密项的设置存成一个配置文件，以供下一次使用。执行步骤 22-24，如图 4-12 所示。



图 4-12

11. 执行步骤 25，如图 4-13 所示，执行加密操作。



图 4-13

12. 加密过程中会出现如下图所示的对话框。

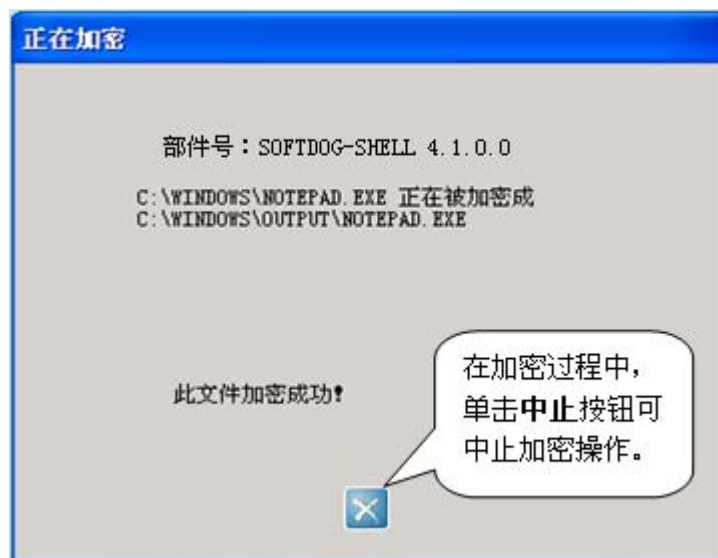


图 4-14

进程条指示加密正在进行。加密结束后会提示结果。如果加密成功，过 2 秒后，程序自动加密下一个文件。如果加密失败，会提示失败原因，并给出解决措施或错误码，关于错误码的定义请参考下面的**保护时的错误代码**。错误信息会左右移动，以引起您的注意。至此，完成了对 Notepad.exe 和 Calc.exe 的保护。

保护时的错误代码

下表列出了保护应用程序和数据文件时可能产生的错误，对于不在下表中的错误代码，请参见安装目录下的 errcode-chn.txt。

错误码 (十六进制)	错误描述
404	SP_ERR_NOT_ENOUGH_MEMORY 没有足够的资源来保护应用程序。尝试关闭您系统上运行的某些应用程序。
405	SP_ERR_CANNOT_OPEN 打开应用程序时出错。检查为您的应用程序所指定的路径。
406	SP_ERR_WRITE_ERROR 无法写入到应用程序。确保应用程序为非只读并且指定了正确的路径。
407	SP_ERR_CANNOT_OVERWRITE 无法写入到应用程序。可能没有选定覆盖标记。
409	SP_ERR_TMP_CREATE_ERROR 无法创建临时文件。
411	SP_ERR_BAD_FILE_INFO 文件没有进行数字签名。
413	SP_ERR_INVALID_MACHINE 该应用程序不是在 Intel x86 平台上开发的。
414	SP_ERR_INVALID_SECTION 文件头中指定的节在应用程序中无法找到。
415	SP_ERR_INVALID_RELOC 无法找到为您的应用程序重新指定的位置，或者重新指定的位置是一个无效地址。
416	SP_ERR_CRYPT_ERROR 无法生成随机数。
418	SP_ERR_IMPORT_OVERWRITE_ERROR 应用程序不支持导入重写。
420	SP_ERR_FRAMEWORK_REQUIRED 需要 .NET Framework 才能保护 .NET DLL。
421	SP_ERR_CANNOT_HANDLE_FILE 外壳引擎无法处理此文件。

错误码 (十六进制)	错误描述
422	SP_ERR_IMPORT_DLL_ERROR 导入的 DLL 的名称长于允许的最大长度（200 字节）。
425	SP_ERR_STRONG_NAME 无法保护使用强名称签名的 .NET 应用程序。
426	SP_ERR_FRAMEWORK_10 需要 .NET Framework 1.0 才能保护 .NET 1.0 应用程序。
427	SP_ERR_FRAMEWORK_SDK_10 需要 .NET Framework SDK 1.0 才能保护 .NET 1.0 应用程序。
428	SP_ERR_FRAMEWORK_11 需要 .NET Framework 1.1 才能保护 .NET 1.1 应用程序。
429	SP_ERR_FRAMEWORK_SDK_11 需要 .NET Framework SDK 1.1 才能保护 .NET 1.1 应用程序。
430	SP_ERR_FRAMEWORK_20 需要 .NET Framework 2.0 才能保护 .NET 2.0、3.0 或 3.5 应用程序。
431	SP_ERR_FRAMEWORK_SDK_20 需要 .NET Framework SDK 2.0 才能保护 .NET 2.0、3.0 或 3.5 应用程序。
432	SP_ERR_APP_NOT_SUPPORTED 不支持此版本的 .NET 应用程序。
433	SP_ERR_FILE_COPY 无法将相应的 DLL 文件复制到目标目录。
434	SP_ERR_HEADER_SIZE_EXCEED 由于文件头大小超出限制，加壳失败。
436	SP_ERR_CODE_MORPHING Code Morphing 不支持该应用程序。
438	SP_ERR_FRAMEWORK_40 需要 .NET Framework 4.0 才能保护 .NET 4.0 应用程序。
439	SP_ERR_FRAMEWORK_SDK_70 需要 Windows SDK 7.0 才能保护 .NET 4.0 应用程序。
440	SP_ERR_OPEN_DOG 外壳引擎不能打开狗。
441	SP_ERR_VERIFY_PASSWORD 外壳引擎校验用户口令失败。

支持的文件类型、编译程序

下表列出了对于 32 位应用程序外壳支持的文件类型/编译器：

编译程序、工具	版本	可执行文件	DLL 文件
---------	----	-------	--------

Visual C++	5.0, 6.0, 7.0, 7.1, 8.0, 9.0, 10.0	是	是
Visual Basic	5.0, 6.0	是	否
Visual FoxPro	5.0, 6.0, 7.0, 8.0, 9.0	是	NA
Borland C++ Builder*	6.0, v2006	是	是
Borland Delphi	7.0, v2006, v2007	是	是
Power Builder	6.0, 7.0, 8.0, 9.0, 10.0, 10.5, 11.0	是	否
Director	5.0, 6.0, 8.0, 8.5, 9.0, 10.1, MX 2004	是	NA
VB .NET**	7.0, 7.1, 8.0, 9.0, and 10.0 with .NET Framework version 1.1, 2.0, 3.0, 3.5, and 4.0	是	是***
C#**	7.0, 7.1, 8.0, 9.0, and 10.0 with .NET Framework version 1.1, 2.0, 3.0, 3.5, and 4.0	是	是***
Delphi .NET	v2006	是	是
Borland C#	v2006	是	是
MFC	6.0, 7.0, 7.1, 8.0, 9.0, and 10.0	是	是
Authorware	6.0, 7.0	是	否
Windev	11, 12	是	否
Labview	7.1	是	否
Prolog	7.1	是	否

GameMaker	5.0, 7.0	是	NA
Excelsior JET	6.0, 6.4	是	是
Xenoage JESart	2.0	是	NA
MDM Zinc****	2.5, 3.0	是	NA

* 也支持 Borland 包库 (BPLs) 。

**用 Any CPU 配置编译的程序在 64 位平台上会作为 32 位程序运行。

***在用来保护.NET DLL 的系统上必须存在.NET Framework。

**** 不支持 MDM Zinc 2.5.0.6 版本的应用程序。

下表列出了对于 64 位应用程序外壳支持的文件类型/编译器：

编译程序、工具	版本	可执行文件	DLL 文件
Visual C++	8.0, 9.0 和 10.0	是	是
VB .NET	8.0, 9.0, 10.0, 带有 .NET Framework 版本 2.0, 3.0, 3.5 和 4.0	是	是*
C#	8.0, 9.0, 10.0, 带有 .NET Framework 版本 2.0, 3.0, 3.5 和 4.0	是	是*

*在用来保护.NET DLL 的系统上必须存在.NET Framework。

注意事项

- 已知外壳加密程序不适合以下几种程序类型：
 - 1、曾经被加过密的程序，要求二次加密；
 - 2、曾经被解过密的程序，要求再加密；
 - 3、被 ASPACK 或 UPX 等压缩工具压缩过的程序；

4、各种开发工具编译出的 Debug 版本的程序。

- 数据文件输入文件名支持通配符 “*”，方便开发商一次处理大量文件。
- 如果选择了限制使用时间，程序将占用软件狗中 10 字节存储区，请指定存储区的首地址，并确保这部分存储区不被其它的应用程序改写。
- 进行加密时需要硬件狗。加密完成后，请您对程序进行仔细测试，测试之前请先初始化一只软件狗，用另一只软件狗进行测试，因为运行被加密的程序可能会改变存储区指定区域的内容。发布您的程序时，请用加密时使用的软件狗来初始化其它的软件狗。
- 加密后的程序在运行过程中，如果找不到硬件狗，可能会弹出对话框，插上狗后按‘重试’按钮，程序会继续执行；按‘取消’会终止程序。
- 当您使用此版本外壳加密工具打开使用软件狗外壳加密工具 3.1 版本保存的配置文件时，某些 3.1 版本不支持的加密选项会被设置为默认值。
- Windows 2003、XP（64 位）的缺省版本及更高版本的 iexplore.exe 无法打开加密的.htm 或.html 文件。新版本的 iexplore.exe 使用 urlmon.dll 打开数据文件，urlmon.dll 是一个系统 DLL 文件，不允许修改此文件。
- 外壳允许您保护多个可执行文件。但是，在保护某个由多个可执行文件，DLL 或者 BPL 组成的应用程序时，您必须牢记：
 - 如果某个单独应用程序中的多个可执行文件共享相同数据文件和相同的算法数据，则它们必须拥有相同的加密因子和加密设置。
- 保护多文件应用程序的一种特殊情况是保护解释程序，后者在运行时使用某个独立的解释程序来执行该应用程序。在这种情况下，您需要保护该解释程序以及您的程序，因为其中一个程序会调用另一个程序。您如何保护这两个可执行程序取决于您使用的编程系统。

在大多数情况下，您要使用相同的保护设置（加密因子和算法数据）将解释程序和您的程序作为两个单独的可执行文件进行保护。例如，要保护一个 FoxPro 3.0 应用程序，您需要保护运行时文件 vfp300.esl 以及您的 EXE 文件。如果您的应用程序是不需要一个独立的运行时程序来解释的解释程序（比如某些 Macromedia 应用程序，把解释程序扩展到可执行文件中），则您可以使用通常的方式保护此单一的可执行文件。在某些情况下，解释程序必须作为主要的可执行程序进行保护，而您的应用程序必须作为该解释程序所使用的数据文件来进行保护。

您可能需要进行试验，看看哪种方法对您有效。如果您需要帮助，请联系技术支持。

- 请注意，当保护一个以 .NET Framework 2.0, 3.0, 3.5 或 4.0 为目标的 DLL 后，程序可能发生异常。该异常发生在如下情况：Exe 和 DLL 使用 Any CPU 配置进行编译，然后只对 DLL 进行

加壳保护，并且 Exe 和 DLL 在 64 位系统上运行。为了避免这种异常，您可以对调用此 DLL 的 EXE 采用以下两种方法之一进行处理：

1. 在编译 EXE 时选择 x86 配置；
2. 使用 .NET Framework SDK 提供的 CorFlags 转换工具 (CorFlags.exe)：

CorFlags theApp.exe /32bit+

- 具有强名称的 .NET EXE 不能被保护；具有强名称的 .NET DLL 可以被保护，但是您必须使用 .NET Framework SDK 提供的强名称工具 (Sn.exe) 对保护后的 DLL 重新进行强名称签名：

Sn -R myAssembly.dll sgKey.snk

- Vista 上一些受保护的文件需要 .mui 文件才能执行。这些 .mui 文件存储在缺省的语言文件夹（例如，对于英文版本则为 en-US）中，该文件夹必须放置在受保护的可执行文件所在的位置。

例如，对于英文版本的 Vista，如果受保护的 Notepad.exe 位于 C:\protected 中，则 .mui 文件必须位于 en-US 文件夹内的相同位置中。如果被保护后应用程序运行失败，请在示例中建议的位置检查 .mui 文件。

- 通过 .NET 增强选项保护的应用程序，以下 DLL 文件应该分发给最终用户：

对于使用 .NET 增强选项保护的 32 位 .NET 应用程序：

SDNPro.dll（在保护过程中选择 .NET 增强选项时）：如果不存在 SDNPro.dll，则受保护的应用程序不能正常运行，并显示缺少 SDNPro.dll 的错误。

对于使用 .NET 增强选项保护的 64 位 .NET 应用程序：

SDNPro64.dll（在保护过程中选择 .NET 增强选项时）：如果不存在 SDNPro64.dll，则受保护的应用程序不能正常运行，并显示缺少 SDNPro64.dll 的错误。

- 有时候，静态链接到应用程序的 DLL 被保护后使得应用程序无法执行，而动态链接该 DLL 时，应用程序却能成功执行。这是在您在保护应用程序之后出现的静态链接 DLL 问题。在保护静态链接的 DLL 时按照以下提示来解决此问题。

- 同时保护静态链接的 DLL 和可执行文件，而不只是保护 DLL。
- 使用 LoadLibrary 加载动态库而不是静态链接 DLL。

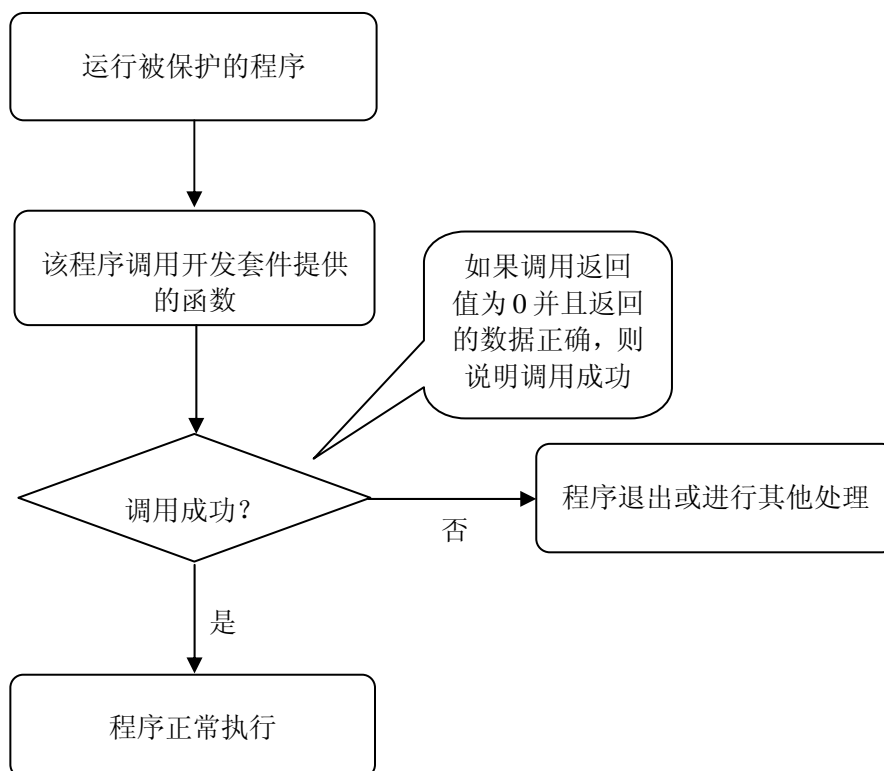
- 对于使用了 XML 序列化的 .Net 程序，需要注意以下事项：

.NET 框架版本	使用“.NET 增强”选项	指南
1.1	否	被保护的程序在运行时需要 .NET Framework 2.0。
1.1	是	被保护的程序在运行时需要 .NET Framework 2.0。
2.0, 3.0 和 3.5	否	.NET SDK 2.0 必须安装在系统上。 <应用程序名>.XMLSerializers.dll 会在保护的过程中自动生成。这个动态库必须和被保护的程序一起部署。
2.0, 3.0 和 3.5	是	.NET SDK 2.0 必须安装在系统上。 <应用程序名>.XMLSerializers.dll 会在保护的过程中自动生成。这个动态库必须和被保护的程序一起部署。
4.0	否	.NET SDK 7.0 必须安装在系统上。 <应用程序名>.XMLSerializers.dll 会在保护的过程中自动生成。这个动态库必须和被保护的程序一起部署。
4.0	是	.NET SDK 7.0 必须安装在系统上。 <应用程序名>.XMLSerializers.dll 会在保护的过程中自动生成。这个动态库必须和被保护的程序一起部署。

注意 使用 XML 序列化的 64 位 .NET 应用程序不能在 32 位操作系统上保护。

第五章 使用API接口函数完成加密

调用 API 函数进行加密，可以灵活的设置加密点，隐蔽性较好，可以具有很高的加密强度。调用 API 函数进行加密又称内嵌式加密。采用内嵌式加密还能够方便地做成限制使用次数、使用时间等功能的试用版软件。对于外壳工具无法加密的特殊程序也可以通过这种内嵌方式完成加密。



全局变量说明

short DogAddr

对软件狗读写操作时用户区中的首地址。取值范围为 0~99。

short DogBytes

对软件狗读写操作时的字节长度。读写时取值范围为 1~100，并且与 DogAddr 之和不能超过 100；

void * DogData

指针型变量。指向读、写操作的数据缓冲区。

函数说明

unsigned long WriteDog ()

输入参数： DogAddr, DogBytes, DogData。

输出参数： 无。

返回值： 0 表示操作成功。其他值是错误码。

功能： 函数把 DogData 指向的数据写入从 DogAddr 开始的存储区中，写入的字节数为 DogBytes。

unsigned long ReadDog ()

输入参数： DogAddr, DogBytes, DogData。

输出参数： DogData。

返回值： 0 表示操作成功。其他值是错误码。

功能： 函数从软件狗中的 DogAddr 开始的存储区读出数据，存入 DogData 指定的缓冲区，读出字节数为 DogBytes。缓冲区的大小要足够长。

如何保护Windows 程序

Windows\Readme-chn.txt 说明文件中列出了所支持的 Windows 2000/XP/Server 2003/Vista/Server 2008/Windows 7/Server 2008 R2 程序的语言模块及目录结构，包括 VC、C++Builder、DELPHI、JAVA、VB、VISUAL FOXPRO、C#、VB.Net、ASP.Net、AutoCAD 等语言。

在每个语言模块的目录中一般包括 Readme-chn.txt 说明文件（关于已通过的编译环境，变量及函数的说明），接口模块及调用接口模块的例子程序。如果采取 DLL 调用方式，运行例子前应将相应的 DLL 拷入系统的 SYSTEM 目录下。

例子程序

使用 API 调用的 C 程序例子

1) 定义 2 个函数和 3 个全局变量

需要在程序开始部分定义 2 个函数和 3 个全局变量

```
extern unsigned long WriteDog(void);
```

```
extern unsigned long ReadDog(void);
```

```
short int DogAddr;
```

```
short int DogBytes;
```

```
void * DogData;
```

如果您用的是 C++，函数声明应用 `extern "C"` 定义，如：

```
extern "C" unsigned long WriteDog(void);
```

2) 函数调用

`WriteDog()`、`ReadDog()` 2 个函数均返回一个长整型数。返回值为 0 表示函数调用成功，其它值是错误码。错误码的意义请查询错误码文件 `Errcode-chn.txt`。



以下部分为使用几个函数达到加密的最简单的例子，目的仅为说明函数的用法，在程序中用此方法只能起到简单的加密效果。如果您想达到良好的加密效果，请阅读“更好的使用软件狗开发套件”一章。

`WriteDog()` 函数可将字符串或其它数据写入软件狗中，`ReadDog()` 函数可将数据读出。但在读写操作时应注意地址分配问题，例如向地址 0 写了 10 字节数据，则再向软件狗中写数据时应写到地址 10 或之后，否则会破坏前面写的数据。读写字符串时将 `DogData` 指向字符串，而读写整型、长整型、浮点型数据时 `DogData` 应指向数据的地址。

读操作时应注意 `DogData` 指向的存储区应有足够的空间存放读出的字节，读操作时并不检查存储空间。有较多的开发人员在該问题上出错，请格外注意！

例 1：向地址 10 写入字符串“SoftDog”，然后读出并比较

.....

```
char WriteBuff[10]={"SoftDog"};
```

```
char ReadBuff[10];
.....

DogAddr = 10;
DogData = WriteBuff;
DogBytes = 7;
if (WriteDog ())
exit (2);
.....
.....

memset (ReadBuff, 0, 10);  /* 将 ReadBuff 清空 */
DogAddr = 10;
DogData = ReadBuff;
DogBytes = 7;
if (ReadDog ())
exit (2);
else
if (strncmp (ReadBuff, WriteBuff, 7))
exit (2);
.....
.....
```

例 2：向地址 50 写入整数 1234，然后读出并比较

```
.....
int wint, rint;
.....
wint = 1234;
DogAddr = 50;
DogData = &wint;
DogBytes = sizeof (int);
if (WriteDog ())
exit (2);
.....
.....
rint = 0;
DogAddr = 50;
DogData = &rint;
DogBytes = sizeof (int);
if (ReadDog ())
exit (2);
else
if (rint != 1234)
```

```
exit (2);  
.....
```

3) 工程文件

源程序修改后，将我们为您提供的 C 语言的 OBJ 文件加到您的工程文件中，编译链接后生成的 EXE 文件就是加密处理过的文件了。由于不同系列号的软件狗的 OBJ 文件也不同，通过 OBJ 文件只能对相对应的软件狗进行操作，所以您在设计加密时不必考虑区分不同号软件狗的工作。如果您购买的是同号狗，可以在软件狗中写入不同的数据来区分软件狗。

如果您使用其它语言，函数的用法差别不大，这里就不一一介绍了。详细情况请参考各个子目录下的 Readme-chn.txt 和例子程序。

第六章 更好地使用软件狗开发套件

简单技巧

1) . 赋值和判断分开

参考强度值:1

在赋值和对值的判断之间，可以插入很多的其他操作，让解密者难以猜到二者之间的联系。对于全局变量，可以在别的函数块中做判断。如下：

```
RetCode = ReadDog ();
// 加入多条语句，可以是与操作狗有关的，或者其他复杂的语句结构。
if ( RetCode != 0 )
{
...
}
```

2) . 多次判断值

参考强度值:2

对于一个值，可以在不同的地方做判断，可以做很多的判断，让解密者难以屏蔽所有的判断操作，而只要有一个地方判断结果不对就可以认为整个结果是不对的。例如：

```
RetCode = ReadDog();
If ( RetCode != 0 )
{
...
}
... // 加入多条语句，可以是与操作狗有关的，或者其他复杂的语句结构
If ( RetCode != 0 )
{
...
}
... // 加入多条语句，可以是与操作狗有关的，或者其他复杂的语句结构
If ( RetCode != 0 )
{
...
}
```

3) .多次赋值

参考强度值:3

对于一个返回值，可以多次赋值给它，可以用复杂循环来做，让解密者无法跟踪该值的变化。在下面的例子中是在使用 `RetCode` 之前多次赋一些无用的值：

```

for ( i = RetCode; i < RetCode + 1000; i++ )
{
    if ( random() % 3 )
        RetCode = i ^ i + 157;
}
RetCode = ReadDog ();

```

4) . 赋多个值

参考强度值:3

返回的结果可以赋给很多的值，让解密者难以跟踪所有的这些返回值。而任意一个返回值不正确，都可以认为结果不正确。

```

RetCode = ReadDog();
for ( i = 0; i < 100; i ++ )
{
    a [i] = RetCode ;
}
for ( i = 0; i < 100; i ++ )
{
    b [i] = a [i] ;
}

```

5) . 使用返回值

参考强度值:4

其实做判断是一种不好的方法，容易被识破和屏蔽。直接使用结果可以更好地让应用软件和加密软件结合。如下例中，如果返回值不正确，圆周率肯定也不正确，程序运行的结果也就会混乱了。

```

RetCode = ReadDog ();
PI = RetCode + 3.14159;

```

6) . 使用狗中常量

参考强度值:6

在软件狗中放置一些常量，要使用时从软件狗中读出使用。比如路径名，文件名，一些常用数据等。在下面的例子中，在软件狗中 10-14 单元放置了字符串“**AUTO**”，15 单元放置了数字 1。

```

DogBytes = 5;
DogAddr = 10;
RetCode = ReadDog();
Strecpy ( Path,  "C:\");
Strcat ( Path,  DogData );
Strcat ( Path,  "EXEC.BAT");
...
DogBytes = 1;
DogAddr = 15;
RetCode = ReadDog ();

```

```
for ( i = 0; i < 100; i = i + *DogData )  
{  
...  
}
```

中级技巧

7) . 随机执行

参考强度值:10

解密者往往是通过大工作量地跟踪程序以发现加密点的，这需要不停地运行程序，以发现规律。在程序中设置随机量可以让加密者糊涂。这种随机可以是随机对返回值进行判断，可以是随机的赋值，也可以是程序随机地执行。

比如下面的程序是随机执行：

```
if ( Random ( 10 ) == 5 )
{
    RetCode = WriteDog ();
    ...
}
if ( Random ( 10 ) == 7 )
{
    RetCode = ReadDog ();
    ...
}
```

一些解密的高手可能会注意到程序执行的随机性，从而屏蔽 **Random** 函数。下面几条技巧也用到随机量，如果可能，可以使用自己编写的随机数发生器。随机发生器的初始化可以取如系统时间这样的随机值。

8) . 随机读

参考强度值:15

可以随机地从软件狗中读取一些内容，但是除非需要，这些内容可以不使用。地址、长度都可以是随机的。

```
DogAddr = Random ( 100 );
DogBytes = Random ( 100 - DogAddr );
RetCode = ReadDog();
...
```

9) . 随机写

参考强度值:20

可以随机地向软件狗中写入一些随机内容，地址、长度都可以是随机的。但是要注意不要覆盖软件狗中有用的内容。较好的办法是在软件狗中规定一个区域用于随机写，并使随机写的地址和长度不会越界。在下面的例子中，单元 80-99 单元是不使用的。

```
DogAddr = 80 + Random(20);
DogBytes = 20 - Random ( 100 - DogAddr );
DogData[1] = DogAddr;
```

```
DogData[2] = DogBytes;  
RetCode = WriteDog();
```

10) . 随机写然后读出 参考强度值:35

这其实是上面两个技巧的配对使用。要注意的是读和写要从程序的运行逻辑上分离远一些，中间可以加入其他对软件狗的操作，这样解密者极难看出两者的联系。

11) . 使用软件狗做全局变量 参考强度值:40

可以把一些变量放在软件狗中，使用软件狗来做传递变量。而且这些变量还是全局的。甚至下一次运行程序时还在。

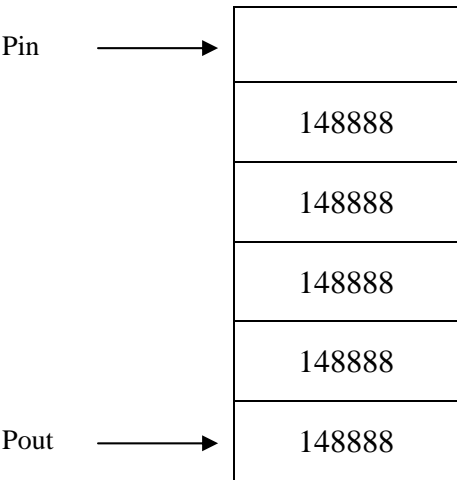
高级技巧

12) . 多线程 参考强度值:20-150

利用高级操作系统的多线程技术可以很好地迷惑解密者和解密工具。基本思路是这样的：在一个线程中对软件狗操作，在另外一个线程中判断和使用。

13) . 队列 参考强度值:40-60

可以建造一个队列以管理返回值和返回结果，管理的方法可以复杂一些，而程序的执行是依赖于这个队列的。当然不用队列而使用其他的数据结构也是一样的，甚至效果更好。一个简单的使用队列进行返回值管理的例子如下图所示：



在上图中，有一个初始化了的队列，共有 6 个单元，初始数据为 148888，当发生的返回值不为 0 时，将 205429 赋给 Pin，并使队列移动一个单元。在 Pout 处检测被推出的数据，当为 148888 时不做动作，当为 205429 时即认为非法。

总结:

在程序中综合使用以上 13 种加密方法可以很好地保护您的软件。参考强度值仅供参考，与使用者的具体使用方法有很大关系。事实上也并不一定要追求高强度值的技巧，低强度参考值的技巧只要使用巧妙，而且多次使用，也可以达到很好的效果。而高级的技巧一般难于实现，使用和调试都比较困难。开发商同时要注意，对软件狗操作是要耗时的，延时可能在 50 - 200 毫秒之间，与所读写字节的长度有关。此外，使用软件狗开发套件提供的 AS 技术，即程序中调用 API 函数和 SHELL 加密 EXE 同时使用，可以有更好的加密强度。

开发商在完成加密后可以使用软件狗加密评测工具（Utility\Evaluation\EvaluationTool-chn.htm）对加密强度进行评测，查看评测结果，建议开发商根据评测结果做相应的修改，以达到较高的加密强度。

实现其他功能

1) 多模块软件或多套软件共用一只软件狗

可以将软件狗提供的字节存储区合理的分配给不同的模块或软件使用，或者都使用相同的数据内容可以实现多模块软件或多套软件共用一只软件狗。

2) 限制软件使用次数及使用时间或使某些模块无法使用

软件开发商可以充分利用软件狗提供的字节存储区，将其中某些字节用来做为计数器使用，初始化为使用次数。程序中对该字节进行读操作，判断是否为 0（已累计到使用次数）来决定程序是否正常运行。然后（或程序退出时）对该字节进行写操作，将使用次数减一。限制使用时间与此相似。限制某些模块的使用同样可以通过判断某标志字节的值来实现。

3) 安装加密软件时检查软件狗（无软件狗不能安装）

软件开发商在加密软件的安装程序中可以通过调用软件狗的模块（如 WIN32DLL）进行查软件狗或使用外壳工具加密安装程序，使用方式与应用程序相同。注意这时应先安装软件狗的驱动程序，然后再对软件狗操作。

4) 使用高强度动态库加密

提供了动态库认证的安全机制，有效地保证了动态库调用的安全性。高强度动态库和一般动态库的主要区别是：高强度动态库提供了密钥认证功能。请参看\Windows\DLL\High DLL 目录下的例子程序。

第七章 封装发行您的软件

发行您的软件时，应提供给最终用户以下内容：

- 经过加密的程序，及软件狗的 DLL（如果调用了软件狗的 DLL）
- 相应的软件狗
- 软件狗的驱动程序
- 软件的说明文档，包括如何安装软件狗的驱动程序等等。

软件狗的驱动程序在软件狗开发套件安装目录\DRIVER 目录下。加密程序通过驱动程序申请资源对并口操作（而不是直接操作并口），可以很好地解决并口争用冲突，保证良好的兼容性。运行加密程序必须安装软件狗的驱动程序，所以要将软件狗开发套件安装目录\DRIVER 目录下的驱动安装程序提供给最终客户。

软件狗驱动程序可以通过以下方式安装

- **直接使用软件狗开发套件提供的驱动安装程序**

在软件狗开发套件安装目录\DRIVER 目录下包含驱动安装程序（SoftDogInstdrv.exe 和 SoftDogSetup.dll 文件）。最终客户须运行 SoftDogInstdrv.exe 安装软件狗驱动程序，运行时需要 SoftDogSetup.dll 文件。如果您不愿意在安装过程中显示我公司的标志，您可以使用 SoftDogInstdrv.exe 带命令行参数不显示安装界面。例如 SoftDogInstdrv /i/s，则安装过程中不显示任何信息。

- **自己编写安装程序安装软件狗的驱动程序**

您的软件安装程序可以调用软件狗开发套件安装目录\DRIVER 目录下的 SOFTDOGSETUP.DLL 来安装软件狗的驱动程序。在 DRIVER 下有各种语言调用 SOFTDOGSETUP.DLL 的例子，供您参考。更详细内容请参考该目录下的 Readme-chn.txt 文档。

关于驱动程序和硬件狗的安装，您可以参考第二章中的相关部分。

第八章 附录

并口硬件的产品规格如下：

尺寸：	55×45×15mm
EEPROM 存储容量：	100 字节
工作电压：	2.5V-6V
写操作平均工作电流：	小于 1mA
读操作平均工作电流：	小于 0.2mA
可写次数：	1, 000, 000
可读次数：	无限
掉电保持时间：	100 年

USB 口硬件产品遵循 USB2.0 规范，规格如下：

尺寸：	54×16×8mm
存储容量：	100 字节
环境温度：	0~55℃
可重写次数：	1, 000, 000
可读次数：	无限
掉电保持时间：	100 年
通讯速率：	1.5Mbps