

灵锐 I

快速入门

2012年9月

Version 1.0.0.0

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北京深思洛克软件技术股份有限公司。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。

1. 授予您使用许可权。您可以为了备份的目的而复制磁盘中的软件，可以为了将本产品集成到您的软件的目的，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。
2. 除已按上述第一条被授权外，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。
3. 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材料上和生产制造上的缺陷。北京深思洛克软件技术股份有限公司的全部责任和您能获得的全部补救措施为：可选择尝试更换或修理或其他补救措施。
4. 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他的保证。对北京深思洛克软件技术股份有限公司的产品、性能或服务亦没有明示的或暗示的或其他任何形式的保证，包括但不限于商品的适销性和对特定用途的适用性。
5. 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：丢失数据、损失利润及其他特别的、偶然的、附随的、继发的或间接的损失。
6. 所有的产品，包括灵锐 I 设备、软件、文档、与本产品一并附送的其他材料及您制作的备份的所有权与版权均属于北京深思洛克软件技术股份有限公司。
7. 违反上述条款时，本协议的授权将自动终止。

“灵锐”是北京深思洛克软件技术股份有限公司的注册商标。

本文所涉及的其他产品和公司名称可能是各自相应所有者的商标。

联系深思洛克

公司名称：北京深思洛克软件技术股份有限公司

办公地点：北京市海淀区中关村南大街甲 6 号铸诚大厦 B 座 1201 室

邮 编：100086

电 话：+86-10-51581366

传 真：+86-10-51581365

电子邮件：sense@sense.com.cn

网 址：<http://www.sense.com.cn>

阅读指南

【手册目标】

本手册主要对北京深思洛克软件技术股份有限公司开发的灵锐 I 加密锁的使用进行说明，由于实际情况千变万化，本手册很难一次做到面面俱到，需要逐渐完善。

【手册约定】



手册中出现该标志的地方表示需要您引起高度重视，否则可能会引发严重的后果。



手册中出现该标志的地方表示需要您特别引起注意的内容。

目 录

第 1 章 产品简介.....	1
1.1 了解灵锐 I.....	1
1.2 灵锐 I 的特点.....	1
1.3 灵锐 I 的工作原理.....	1
1.4 灵锐 I 设备的驱动.....	1
1.5 灵锐 I 的开发包.....	1
1.6 灵锐 I 的工具.....	2
1.7 灵锐 I 的 API 函数.....	2
1.8 名词解释.....	2
第 2 章 灵锐 I 的开发包.....	4
2.1 灵锐 I 的硬件安装.....	4
2.2 安装和卸载开发包.....	5
2.2.1 安装开发包.....	5
2.2.2 开发包中的内容.....	8
2.3.3 卸载开发包.....	8
第 3 章 快速开发.....	11
3.1 使用灵锐 I 保护软件的流程.....	12
3.2 使用灵锐 I 保护软件的范例.....	13
附录 I.....	23

第 1 章 产品简介

1.1 了解灵锐 I

灵锐 I 是北京深思洛克软件技术股份有限公司开发的一款加密锁,属于我公司的低端产品。灵锐 I 是基于硬件(加密锁)的软件防盗版系统,采用了国际标准的 AES 加密算法技术,您可以使用它来保护自己的软件,从而远离盗版的困扰。此外,灵锐 I 还提供了标准的 HMAC-SHA1 算法功能,可用于认证网络客户端的身份,实现身份认证的功能。

1.2 灵锐 I 的特点

灵锐 I 不仅提供了大容量的非易失性数据存储空间,还支持 AES 高级加密算法,除此之外,还具有全球唯一序列号、安全隧道等特点。关于灵锐 I 的更多特点以及详细的介绍请参考《灵锐 I V1.0.0.0 开发指南》第 1.2 章节。

1.3 灵锐 I 的工作原理

灵锐 I 提供了大容量的数据区,可以将重要的数据存放到加密锁的数据区中,当软件中用到这些数据时,使用我们提供的 API 函数来读取加密锁中的数据。这样,就可以实现将软件与加密锁绑定在一起,达到保护软件的效果。此外,灵锐 I 还支持 AES 高级加密算法,利用该算法可以实现更加丰富的功能。[第 3 章](#)我们将通过一个完整的范例向您演示使用灵锐 I 保护软件的一般方式。关于灵锐 I 的更多应用请参考《灵锐 I V1.0.0.0 开发指南》。

1.4 灵锐 I 设备的驱动

灵锐 I 设备在 Windows 平台上以免驱动的方式工作,即只要在计算机的 USB 接口上插入灵锐 I,就能够在设备管理器中的人体学输入设备下面看到灵锐 I 设备,详细内容请参考第 [2.1 章节](#)。

1.5 灵锐 I 的开发包

可以在我们的网站下载灵锐 I 开发包的安装文件,详细的安装过程请参考第 [2.2 章节](#)。

1.6 灵锐 I 的工具

为了便于测试灵锐 I 硬件和开发应用程序，我们提供了几个常用的工具，所有工具都存放在开发包的 Tools 目录中，关于这些工具的详细说明请参考《灵锐 I V1.0.0.0 开发指南》第 2 章。

1.7 灵锐 I 的 API 函数

为了便于在软件中访问灵锐 I 设备，我们提供了丰富的接口函数，在软件中使用 API 函数访问灵锐 I 的方法请参考《灵锐 I V1.0.0.0 开发指南》第 3.3 章节，有关 API 函数的详细介绍请参考《灵锐 I V1.0.0.0 函数参考》第 1 章。

1.8 名词解释

- 全球唯一序列号

全球唯一序列号是灵锐 I 出厂时设定的硬件序列号，您可以利用它来实现软件与设备的绑定或者实现产品的跟踪与追溯。

- 开发商编号

当您初次订购灵锐 I 产品时，我们会为您分配一个唯一的开发商编号，在以后的订购中，所有提供给您的灵锐 I 设备都会设置成和初次订购相同的编号。

- 密码

灵锐 I 使用密码来管理不同的权限，密码验证通过后即可获得相应的权限，共设置了三种密码，即管理密码、普通用户密码和认证密码。

- 管理密码

管理密码是您管理灵锐 I 设备时所使用的密码，管理密码拥有操作灵锐 I 设备的最高权限，验证通过后可以完成设置数据区、设置密钥以及重置密码等管理级别的操作。

- 普通用户密码

普通用户密码验证通过后，可以用于调用 AES 算法和读写数据区，拥有此级密码的权限为对区块 0 可读可写，对区块 1~3 只读。

- 认证密码

认证密码验证通过后，可以用于调用 HMAC 算法进行身份认证操作和读写数据区，拥有此级密码的权限为对区块 0 可读可写，对区块 1~3 只读。

- 密钥

灵锐 I 的密钥有两种，即远程升级密钥和身份认证密钥。

- 远程升级密钥

远程升级密钥是验证远程升级包时使用的密钥，长度为 20 字节。

- 身份认证密钥

身份认证密钥用来鉴别终端用户的身份，长度为 20 字节，您需要为每把灵锐 I 设备设置不同的身份认证密钥，通过此密钥来鉴别终端用户的身份。

关于这些名词的更多介绍请参考《灵锐 I V1.0.0.0 开发指南》第 1.4 章节。

第 2 章 灵锐 I 的开发包

2.1 灵锐 I 的硬件安装

灵锐 I 设备遵循 USB 2.0 设备规范，并符合 HID 协议要求，在大多数常用操作系统上无需另外安装设备驱动程序，您只要在计算机的 USB 接口上插入灵锐 I 设备，就能够在设备管理器的人体学输入设备下面看到灵锐 I 设备是否可以正常使用，降低了部署和维护的难度。

灵锐 I 在设备管理器中的显示如图 2-1 所示。

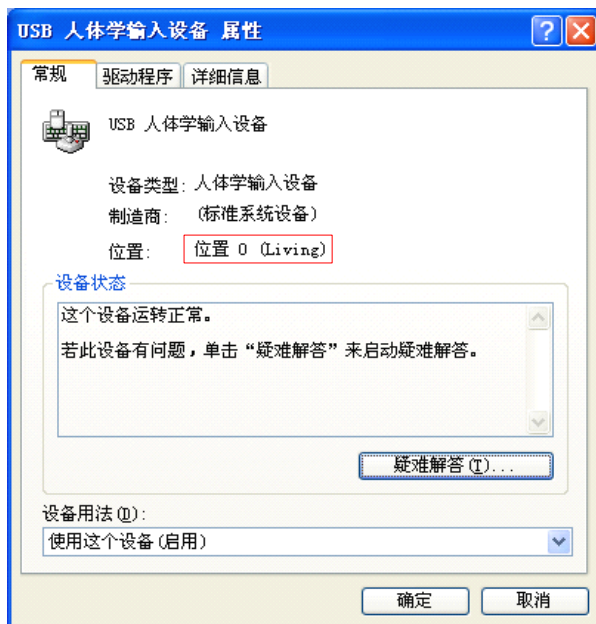
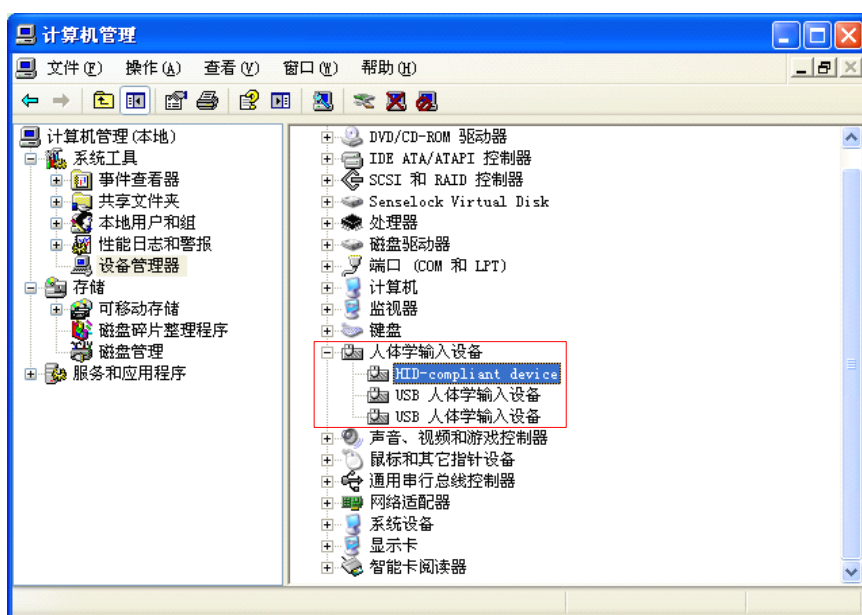
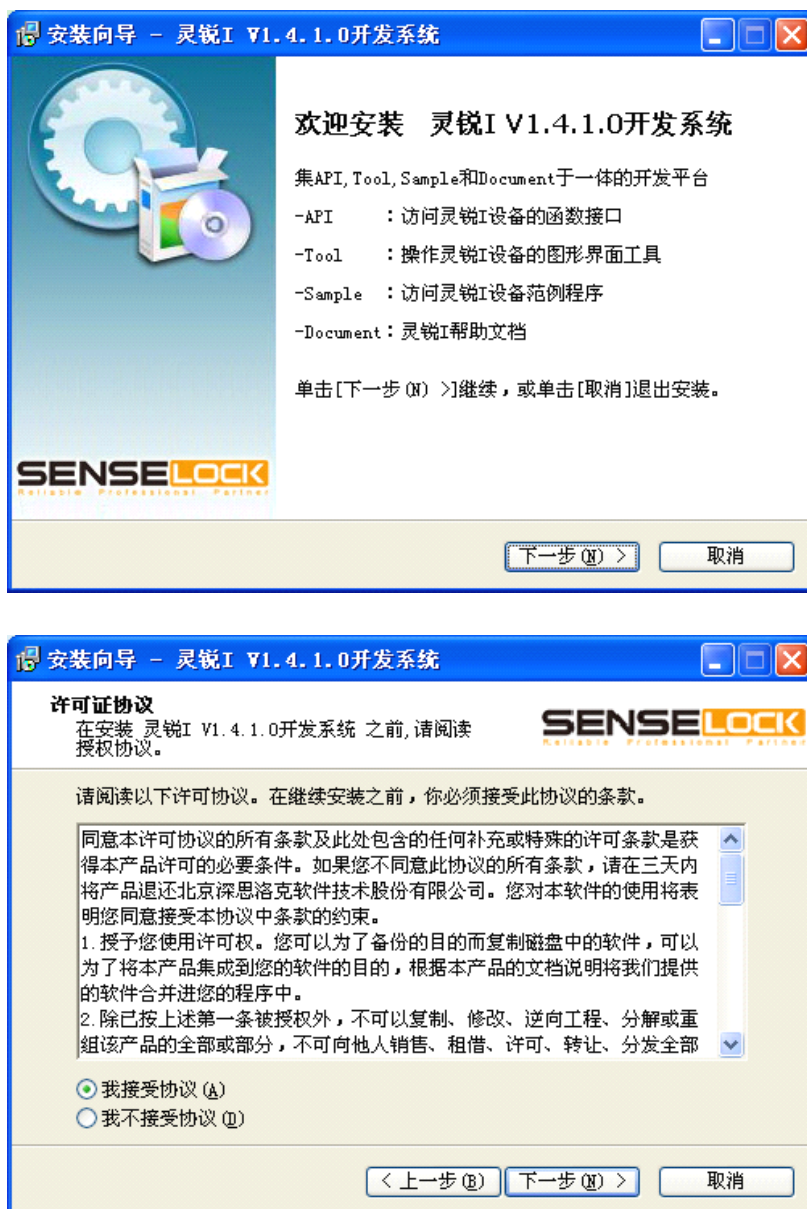


图 2-1 设备管理器中显示的灵锐 I 设备

2.2 安装和卸载开发包

2.2.1 安装开发包

在使用灵锐 I 开发套件之前，您需要先安装产品的开发包（SDK）。安装开发包的过程如图 2-2 中的几个图所示。



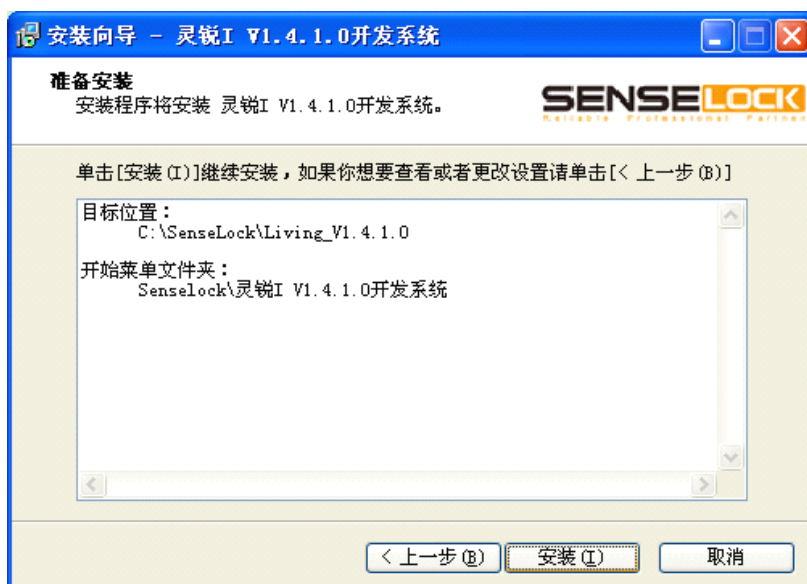
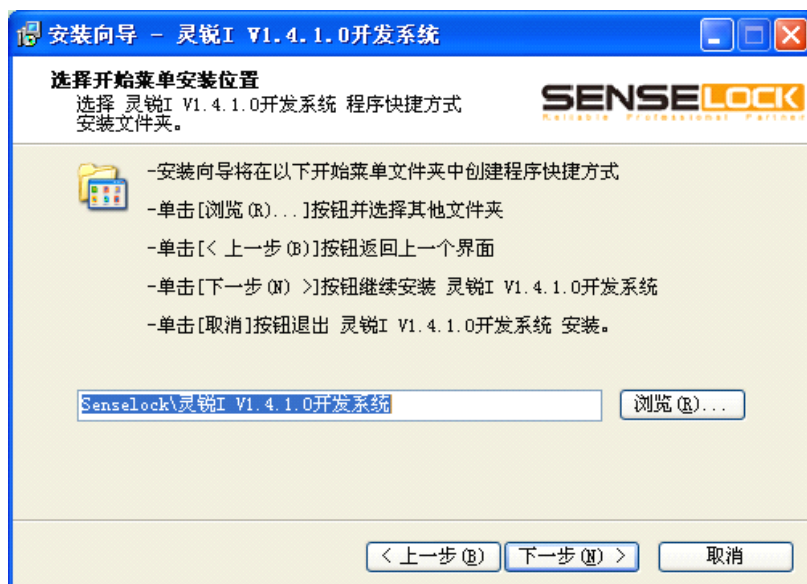
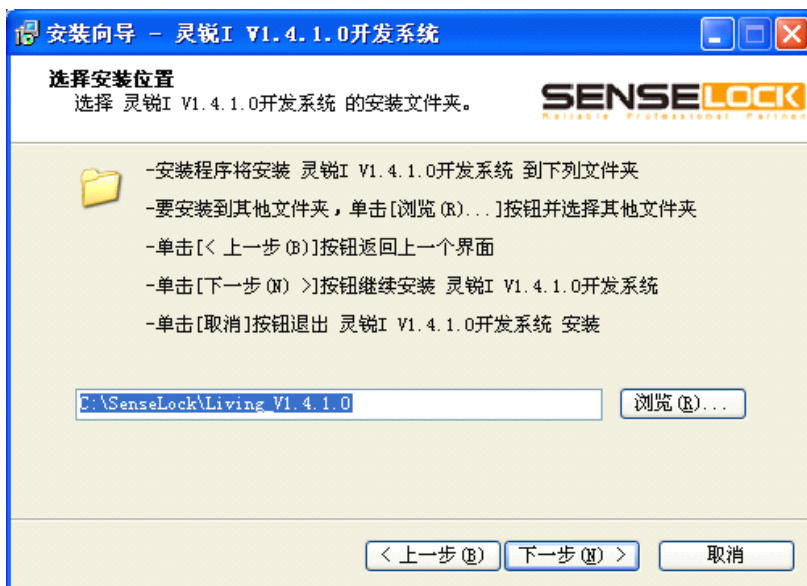




图 2-2 安装灵锐 I 开发包

2.2.2 开发包中的内容

开发包安装完成后，打开开发包的安装目录，目录中的内容如表 2-1 所示。

表 2-1 开发包中的内容

目录	内容
API	软件中使用 API 函数时需要使用的动态库、静态库和头文件
Docs	灵锐 I 的手册
Samples	各种范例
Tools	灵锐 I 的各种工具

2.3.3 卸载开发包

可以使用控制面板中的“添加/删除程序”卸载开发包，也可以通过“开始”菜单来完成开发包的卸载。

第 3 章 快速开发

灵敏 I 提供了大容量的数据区，可以将重要的数据存放到加密锁的数据区中，当软件中用到这些数据时，使用我们提供的 API 来读取加密锁中的数据。这就是使用灵敏 I 保护软件的一般方式。

本章将通过一个范例向您展示如何将软件中的数据存放到加密锁的数据区中以及如何在软件运行过程中读取这些数据。通过这个范例，您大致可以了解使用灵敏 I 保护软件的一般方式。

3.1 使用灵敏 I 保护软件的流程

使用灵敏 I 保护软件的流程如图 3-1 所示。

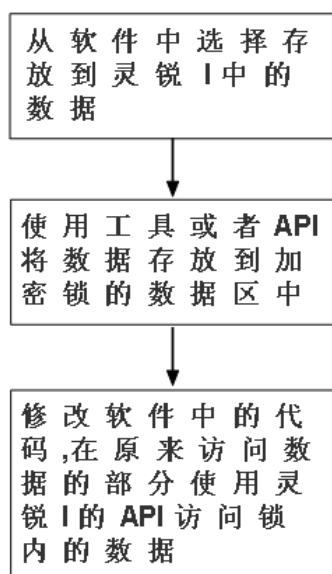


图 3-1 使用灵敏 I 保护软件的流程

3.2 使用敏锐 I 保护软件的范例

为了便于理解，我们选择一个比较简单的示例程序，程序的功能是将一组数据输出。此处我们只是为了举例说明，实际应用过程中请不要将从加密锁中读取的数据直接输出。

在 VC6 中新建一个 WIN32 控制台程序工程 (Win32 Console Application)，将图 3-2 所示代码添加到 VC6 工程中。在本例中，我们假定将工程保存为 C:\livedemo\Win32\demo1.dsp，图 3-2 所示的代码保存为 C:\livedemo\Win32\pc_demo1.c。

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
void main(int argc, char **argv)
{
    unsigned char data[] = {'s','e','n','s','e','l','o','c','k'};
    int len = sizeof(data);
    int i;
    printf("\n");
    for (i=0;i<len;i++)
    {
        printf("%c",data[i]);
    }
    printf("\n\n");
}
```

图 3-2 示例程序

编译并运行上面的代码会得到图 3-3 所示的结果。

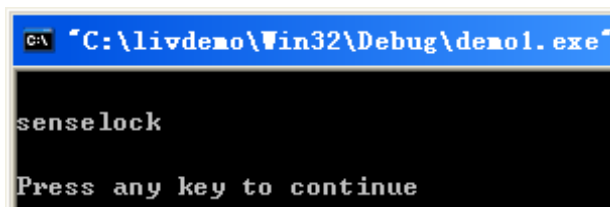


图 3-3 示例程序的运行结果

一、将软件中的数据存放到灵敏 I 设备内

假设范例中使用的数据非常重要，现在需要将这些数据存放到灵敏 I 的数据区中。选择好数据之后，可以使用我们提供的工具将数据存放到加密锁中，也可以使用 API 的方式来写入数据，此处我们选择工具的方式，有关 API 方式的详细介绍请参考《灵敏 I V1.0.0.0 开发指南》第 3.3 章节。

插入一把灵敏 I 设备，然后双击开发包 Tools 目录中的 DevelTool.exe 启动灵敏开发设置工具，如图 3-4 所示，此处我们将范例中的数据（“senselock”）存放到加密锁的区块 1 中。关于灵敏开发设置工具的详细介绍请参考《灵敏 I V1.0.0.0 开发指南》第 2.1 章节，关于数据区的详细介绍请参考《灵敏 I V1.0.0.0 开发指南》第 3.1 章节。

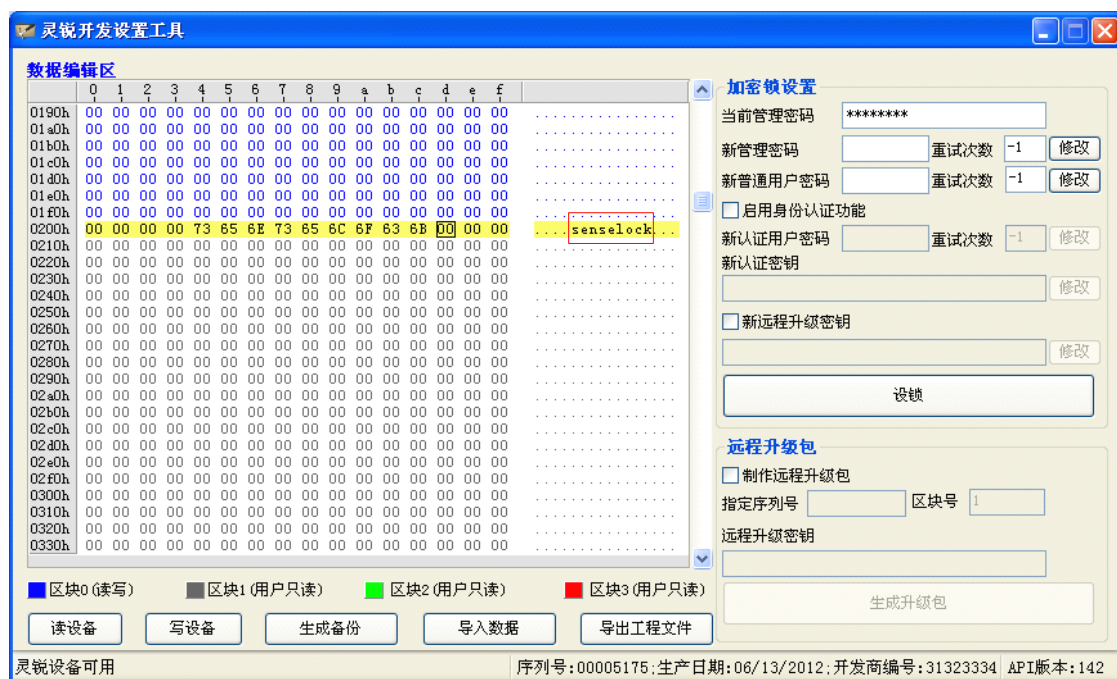


图 3-4 灵敏 I 开发工具

编辑好数据之后，需要填写工具右上侧的“当前管理密码”一项，工具会自动填入此项，但工具自动填入的是默认的管理密码，如果您更改过管理密码，需要重新输入新的值。填写完之后，点击“写设备”按钮，如果写锁成功，会在工具左下侧的状态信息栏显示“写设备成功”，否则将提示相应的错误信息。

二、软件中使用 API 访问灵锐 I

将重要数据存放到加密锁的数据区之后，可以点击开发设置工具中的“读设备”按钮来查看写入的数据，有关该功能的详细介绍请参考《灵锐 I V1.0.0.0 开发指南》第 2.1 章节，这里介绍在软件中通过 API 的方式访问灵锐 I 的方法。修改本章节之前介绍过的工程 demo1.dsp，首先将开发包 API\Win_X86\Visual Studio\Include 文件夹中的 living1.h 拷贝到工程目录中，将开发包 API\Win_X86\Visual Studio\DLL 文件夹中的动态库拷贝到工程目录下并加入到工程中，然后修改图 3-2 所示的 pc_demo1.c 的代码，修改后的所有代码如图 3-5 所示。关于代码中所使用 API 函数的详细介绍请参考《灵锐 I V1.0.0.0 函数参考》第 1 章。

```
#include <windows.h>
#include <stdio.h>
#include <string.h>
#include "living1.h"

int main(int argc, char* argv[])
{
    int handle;
    int res,i;
    unsigned char data[512];

    /*打开灵锐 I*/
    res=LIV_open(0x31323334,0,&handle);
    if(res!=LIV_SUCCESS)
    {
        printf("Open failed! Error code=%d\n",res);
        return 1;
    }

    /*校验普通用户密码*/
    res=LIV_passwd(handle,1,(unsigned char *)"12345678");
    if(res!=LIV_SUCCESS)
    {
        LIV_close(handle);
        printf("Verify user password failed! Error code=%d\n",res);
        return 1;
    }

    /*读取区块 1 的数据*/
    res=LIV_read(handle,1,data);
    if(res!=LIV_SUCCESS)
```

```
{
    LIV_close(handle);
    printf("Read failed! Error code=%d\n",res);
    return 1;
}

/*显示读取的数据，此处只显示之前写入的内容*/
printf("read data: ");
for (i=4;i<13;i++)
{
    printf("%c",data[i]);
}

/*关闭灵锐 I*/
res = LIV_close(handle);
if(res!=LIV_SUCCESS)
{
    printf("Close failed! Error code=%d\n",res);
    return 1;
}
printf("\n\n");
return 0;
}
```

图 3-5 pc_demo1.c 修改后的代码

程序的执行结果分为以下两种情况：

1. 插入灵锐 I 设备执行程序

结果如图 3-6 所示，成功输出了从加密锁中读取出的数据。

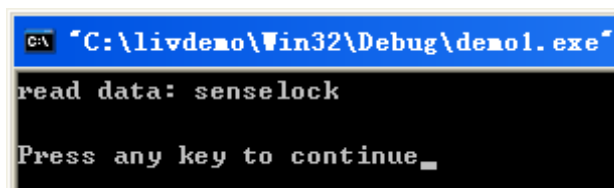


图 3-6 插入灵锐 I 运行程序的结果

2. 不插灵锐 I 设备执行程序

结果如图 3-7 所示，执行失败。



图 3-7 不插灵锐 I 运行程序的结果



- ◆ 在测试本范例时，需要将“LIV_open(0x31323334, 0, &handle);”的第一个参数替换成您自己锁的开发商编号。

三、总结

通过此范例可以看出，最初“senselock”是存放到了软件中，假设该数据非常重要，于是我们就把它存放到加密锁内，当软件中用到此部分数据时，再通过我们提供的 API 读取这些数据，这样就达到了不插灵敏 I 软件不能够正常运行的效果。图 3-8 说明了软件被保护前后的变化。

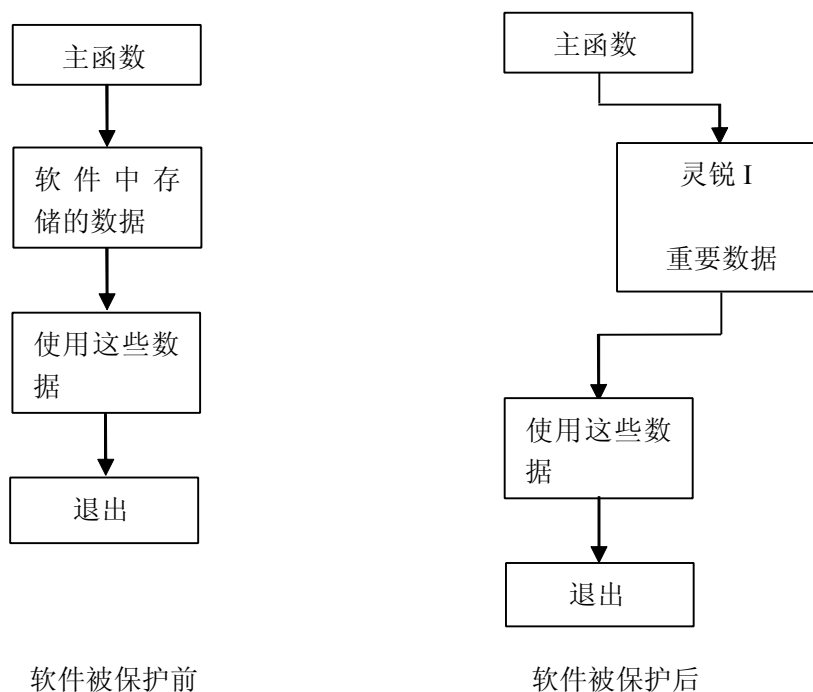


图 3-8 软件被保护前后的对比

附录 I

A. 支持的开发语言

VC++、C++ Builder、C#、Java、Delphi、VB、PB、AutoCAD。

B. 支持的操作系统

Windows 2000、Windows XP (32 位, 64 位)、Windows Server 2003 (32 位, 64 位)、Windows Vista (32 位, 64 位)、Windows Server 2008 (32 位, 64 位) 和 Windows7 (32 位, 64 位)。

C. 硬件技术规格

项目	值	备注
工作电压	DC 4.5~5.5V	无
最大功耗	200mW	无
工作温度	0~70℃	无
数据存储时间	10 年	典型值
擦除/写周期	1 万次	最低值
设备接口	USB 1.1	低速设备, 符合 HID 规范

D. 性能测试

项目	数值	备注
AES 加密时间	20ms	16 字节平均值
AES 解密时间	26ms	16 字节平均值
读取时间	179ms	512 字节平均值
写入时间	246ms	512 字节平均值
HMAC 运算时间	185ms	100 字节平均值

注：以上数据仅供参考，实际应用过程中可能存在误差。