
ET199/ET 金刚锁

智能虚拟化高强度加密工具

使用手册

V1.0



版权所有©2010 北京坚石诚信科技股份有限公司

<http://www.jansh.com.cn>

北京坚石诚信科技股份有限公司

软件开发协议

北京坚石诚信科技股份有限公司（以下简称坚石）的所有产品，包括但不限于：开发工具包，磁盘，光盘，硬件设备和文档，以及未来的所有定单都受本协议的制约。如果您不愿接受这些条款，请在收到后的 7 天内将开发工具包寄回坚石，预付邮资和保险。我们会把货款退还给您，但要扣除运费和适当的手续费。

1. 许可使用

您可以将本软件合并、连接到您的计算机程序中，但其目的只是如开发指南中描述的那样保护该程序。您可以以存档为目的复制合理数量的拷贝。

2. 禁止使用

除在条款 1 中特别允许的之外，不得复制、反向工程、反汇编、反编译、修改、增加、改进软件、硬件和产品的其它部分。禁止对软件 and 产品的任何部分进行反向工程，或企图推导软件的源代码。禁止使用产品中的磁性或光学介质来传递、存储非本产品的原始程序或由坚石提供的产品升级的任何数据。禁止将软件放在服务器上传播。

3. 有限担保

坚石保证在自产品交给您之日起的 12 个月内，在正常的使用情况下，硬件和软件存储介质没有重大的工艺和材料上的缺陷。

4. 修理限度

当根据本协议提出索赔时，坚石唯一的责任就是根据坚石的选择，免费进行替换或维修。坚石对更换后的任何产品部件都享有所有权。

保修索赔单必须在担保期内写好，在发生故障 14 天内连同令人信服的证据交给坚石。当将产品返还给坚石或坚石的授权代理商时，须预付运费和保险。

除了在本协议中保证的担保之外，坚石不再提供特别的或隐含的担保，也不再对本协议中所描述的产品负责，包括它们的质量，性能和对某一特定目的的适应性。

5. 责任限度

不管因为什么原因，不管是因合同中的规定还是由于刑事的原因，包括疏忽的原因，而使您及任何一方受到了损失，由我方产品所造成的损失或该产品是起诉的原因或与起诉有间接关系，坚石对您及任何一方所承担的全部责任不超出您购买该产品所支付的货款。在任何情况下，坚石对于由于您不履行责任所导致的损失，或对于数据、利润、储蓄或其它的后续的和偶然的损失，即使坚石被建议有这种损失的可能性，或您根据第 3 方的索赔而提出的任何索赔均不負責任。

6. 协议终止

当您不能遵守本协议所规定的条款时，将终止您的许可和本协议。但条款 2，3，4，5 将继续有效。

北京坚石诚信科技股份有限公司

地址：北京市海淀区学清路 9 号汇智大厦 B 座 2 层

邮编：100192

电话：010—82730011

传真：010—82737938

网址：<http://www.jansh.com.cn>

目 录

第一章 引言	1
第二章 产品介绍	2
2.1 产品功能	2
2.2 产品特点	2
2.2.1 丰富的文件类型支持	2
2.2.2 保护过程对用户透明	2
2.2.3 高加密强度	2
2.2.4 系统支持全面	2
2.3 工作原理	2
2.3.1 加密文件流程	3
2.3.2 使用加密后的文件流程	3
第三章 软件使用说明	4
3.1 软件界面	4
3.2 一个加密实例	6
3.3 使用加密后的文件	15
第四章 自定义保护段	16

第一章 引言

首先非常感谢您选择我们的 ET199 智能虚拟化高强度加密工具保护您的程序。ET199 智能虚拟化高强度加密工具是一款先进高效的软件保护产品。它可以保护您的程序不被非法复制，非授权访问和使用。

ET199 智能虚拟化高强度加密工具的设计在硬件上基于 ET199、ET 金刚锁这 2 种类型的加密锁，软件上基于虚拟机技术，这两方面技术的结合使其具有操作简单，保护强度高，使用透明等特点，该工具支持各种常见的 PE 可执行文件。

通过阅读本手册，您可以快速了解该产品的特性，掌握如何使用它来保护您的文件。

第二章 产品介绍

2.1 产品功能

ET199 智能虚拟化高强度加密工具是一款透明的软件保护产品，使用 ET199 智能虚拟化高强度加密工具加密软件，用户无需具有专业的软件加密知识就可以非常便捷的将自己各种类型的 PE 文件进行加密保护，受保护的程序只有通过指定的加密锁硬件存在的条件下才可以被运行。加密后的程序即使被别人非法获取，由于没有所依赖的硬件，程序依然无法运行。

2.2 产品特点

2.2.1 丰富的文件类型支持

ET199 智能虚拟化高强度加密工具能够对 Windows 平台下的各种常见类型的 32 位 PE 文件进行保护。这些文件包括 .EXE，.dll，.ocx 等文件格式，及各种语言编写的 PE 文件。

2.2.2 保护过程对用户透明

用户只需简单的用鼠标选择要保护的 PE 文件，即可完成对 PE 文件的保护。所有的安全检测、PE 文件的加解密等操作都是由 ET199 智能虚拟化高强度加密工具在后台自动处理完成，从用户角度看对文件的保护是完全透明的。

2.2.3 高加密强度

ET199 智能虚拟化高强度加密工具在硬件上基于 ET199 和 ET 金刚锁这 2 种加密锁实现。该锁使用虚拟机技术进行以函数为单位的加密，从而提高加密锁的加密强度。基于此硬件实现的 ET199 智能虚拟化高强度加密工具使用户数据的安全性提高到一个新的高度。

2.2.4 系统支持全面

文件保护中心可以运行于多种版本的 Windows 操作系统之上。加密的 PE 文件可以支持的操作系统有：Windows 2000/XP/Sever 2003/Vista/ Server 2008/7，为用户提供更多的平台选择。

2.3 工作原理

要让 ET199 智能虚拟化高强度加密工具正确保护你的文件，首先要执行将 PE 文件（Windows 平台下的 .exe、.dll、.ocx 等文件）进行加密操作。例如我们要

加密一个“记事本”程序 Notepad.exe，加密后的 Notepad.exe 文件只有使用加密时设置的对应加密锁才能打开。若未插入对应的加密锁，那么加密后的 PE 文件是不能被打开的，默认会提示“找不到指定的加密锁”，从而保证了 PE 文件的安全性。

2.3.1 加密文件流程

ET199 智能虚拟化高强度加密工具加密文件流程如

所示。

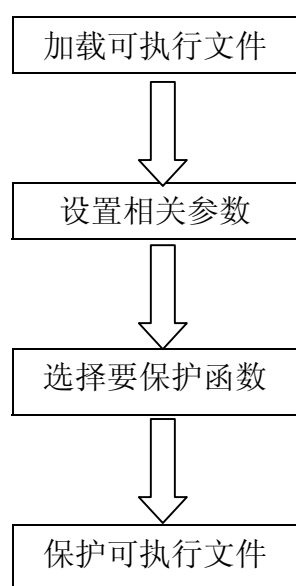


图 2.3-1 文件加密流程图

其中，“可执行文件”是指具有 PE 文件格式的.dll、.exe 和.ocx 等文件。执行完成上图流程后，PE 文件已经被加密保护。

2.3.2 使用加密后的文件流程

加密后的可执行文件在执行到被保护的函数时会检测加密锁硬件是否存在，相应的参数是否正确，若有一项条件未满足，应用程序将不能正常运行。

检测加密锁硬件是否存在不仅在函数被执行时检测，同时也会在程序运行过程中检测，如用户在运行中有拔除锁的操作，此时加密后的程序会默认提示找不到指定的加密锁，然后退出。

第三章 软件使用说明

3.1 软件界面

左键双击“VMProtect.exe”启动ET199 智能虚拟化高强度加密工具，显示如图 3.1-1所示的主界面。



图 3.1-1 加密工具主界面

其中加密锁显示项会显示设置加密时的相关参数，点击“加载文件”按钮会添加“待加密的可执行文件”，此时工具的左上方会显示待加密可执行文件的输入路径和输出路径，而且界面会从“加密锁”界面变换到“函数块”的界面（图 3.1-2）。用户可以在“参数配置栏”里设置加密相关参数，当插入加密锁时，工具会自动获取锁的客户号和 HID，进行加密时可选择绑定 HID 功能（**如果选定了与 HID 绑定，那么加密后的程序只能被加密时的锁打开**），用户也可以设置找不到锁时的提示信息标题和提示内容；如果是 ET 金刚锁，参数配置栏里还会有时间限制和锁过期信息的设置，时间限制可以设置加密后文件的使用期限，注意：

时间限制设置时不能小于当前时间；“信息栏”会实时显示各种操作的相关信息。



图 3.1-2 加密工具函数块界面

图 3.1-2 为没有显示要保护的函数界面，函数定位栏显示待加密的可执行文件要保护的函数，函数显示栏显示要保护的每个函数的具体信息，点击不同的函数会显示不同的内容，信息栏同 3.1-1 所述。

函数的定位分为 4 种方式，分别为标志定位，输入表定位，自动搜索定位和手工输入定位。标志定位方式和输入表定位方式适用于待加密可执行文件在有源代码的情况下进行加密，在源代码里可以添加自己想要保护的函数标志进行保护，如果待加密的可执行文件已经带有标志定位或输入表定位标志，在加载此文件时就可自动获取要保护的函数，需要注意的是：被保护函数需在函数返回前加以保护，否则会找不到标志。自动搜索定位方式适用于没有源代码的情况下模糊查找函数，当然标志定位或输入表定位可以和自动搜索定位同时使用，此时自动

搜索方式不会重复找出标志定位或输入表定位已找的函数。手工输入定位适用于在知道具体要保护哪个函数的情况下进行保护。

3.2 一个加密实例

这里我们将通过一个具体的例子来演示如何使用 ET199 智能虚拟化高强度加密工具来完成对文件的保护。

(1) 将 ET199 加密锁连接到执行加密操作的计算机上。

(2) 运行“VMProtect.exe”启动加密工具，工具会自动获取客户号和 HID，如图 3.2-1。

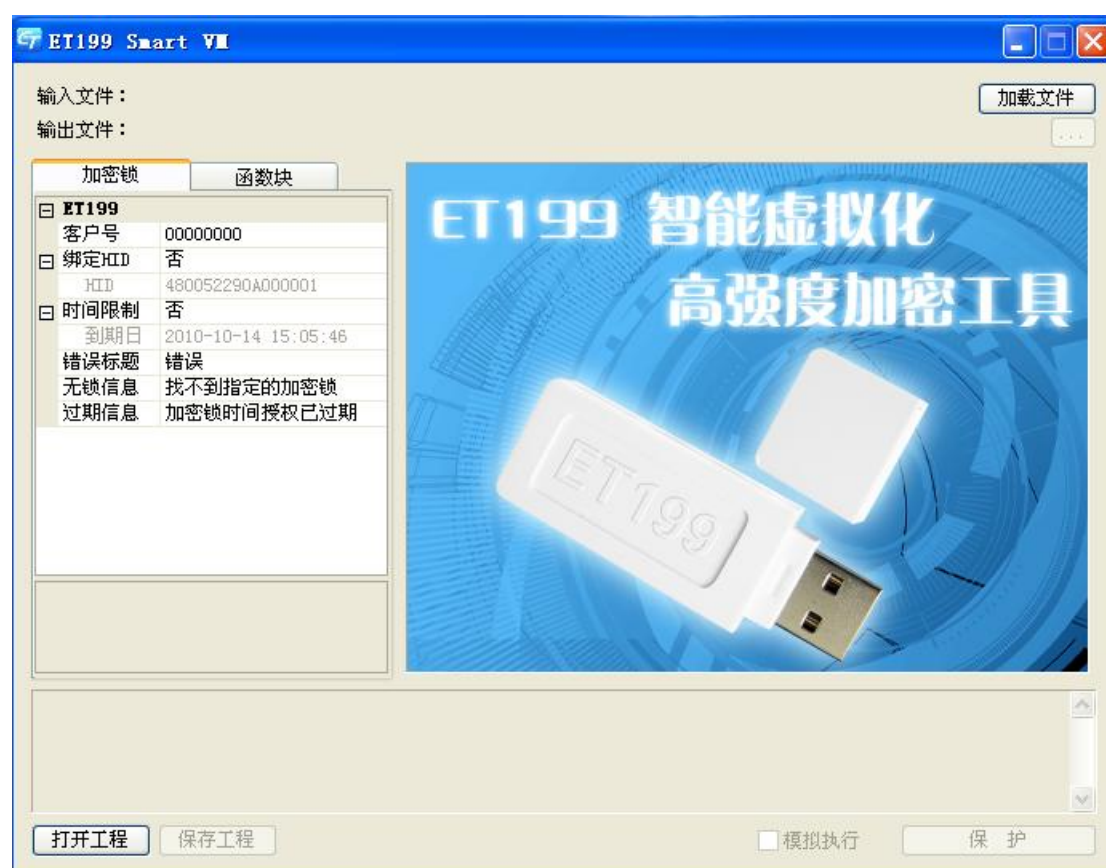


图 3.2-1 加密工具界面

(3) 添加可执行文件。点击工具右上角的“加载文件”按钮，弹出如图 3.2-2 文件选择对话框所示。

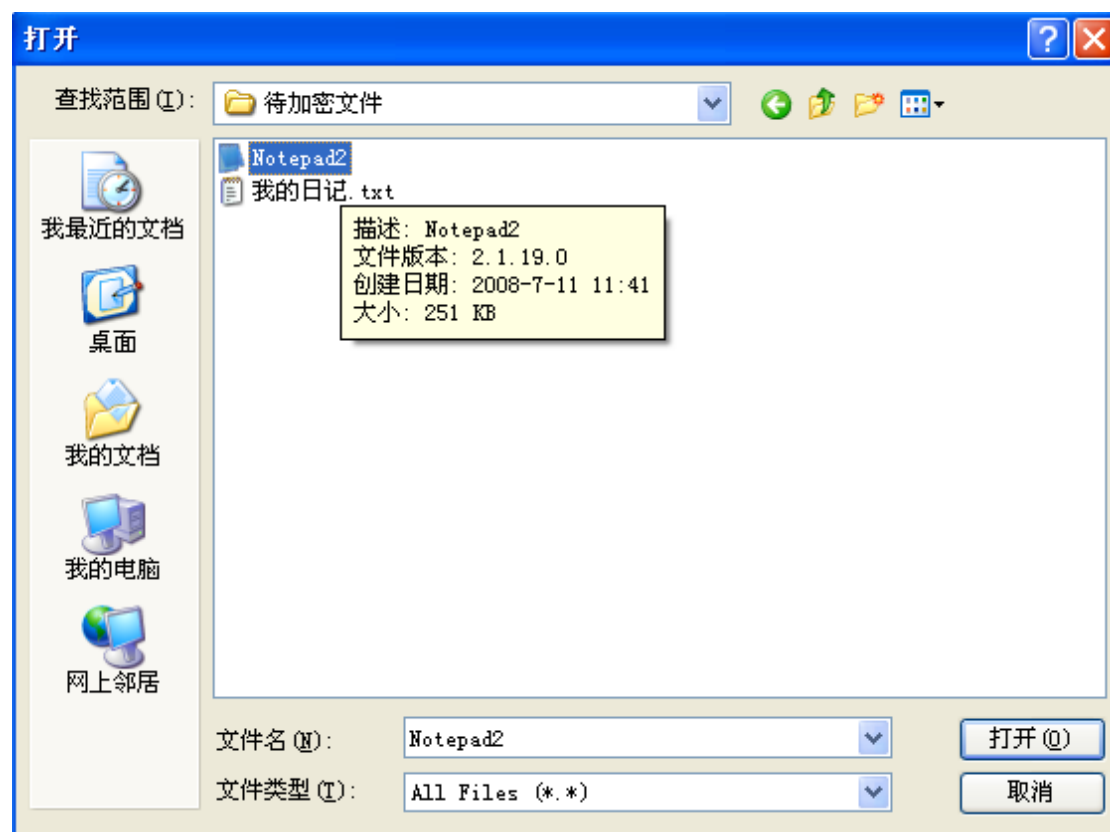


图 3.2-2 文件选择对话框

添加“Notepad2.exe”可执行文件，“Notepad2.exe”文件将显示在左上方输入文件的路径中，同时也显示了加密后的输出路径及加密后的文件名字，默认在其文件名前添加“Enc_”，工具界面也会从加密锁界面变换到函数块界面，如图 3.2-3。

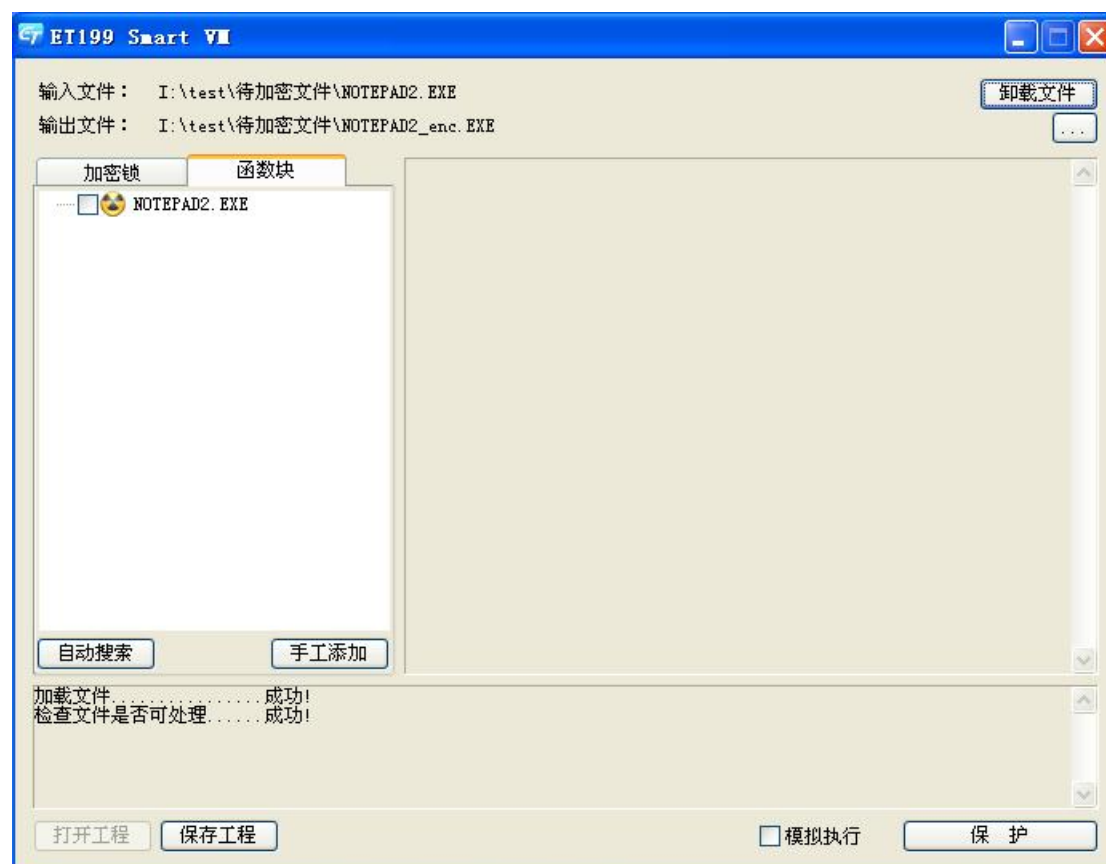



图 3.2-3 加载文件界面

用户也可更改加密后的文件输出路径及加密后的文件名，点击输出文件最右边的  按钮即可完成。注意，如果想加密其他的 PE 文件，必须卸载先前添加的加密文件后才能重新添加其他 PE 文件进行加密。

(4) 配置加密相关参数，设置绑定 HID，因为每个加密锁的 HID 各不相同，所以绑定加密锁的 HID 加密后，被加密软件只有在这把锁连接到主机的条件下才能运行，该加密方式加密后的可执行文件与加密锁是一一对应的关系。设置时间限制为 2 天后过期（当天时间为 2010-10-13），其他提示信息为默认，如图 3.2-2。

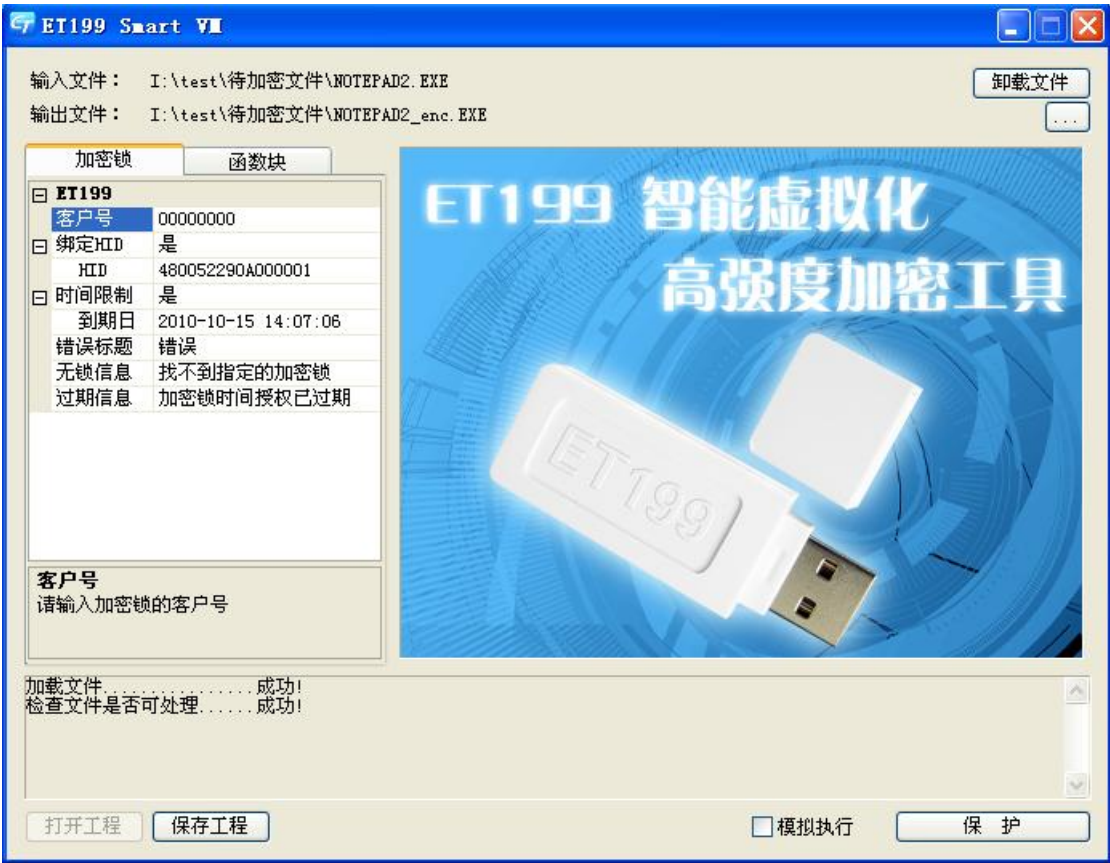


图 3.2-4 设置参数界面

(5) 配置加密函数块定位功能。选择自动搜索方式，搜索数量为 10，如图 3.2-5。

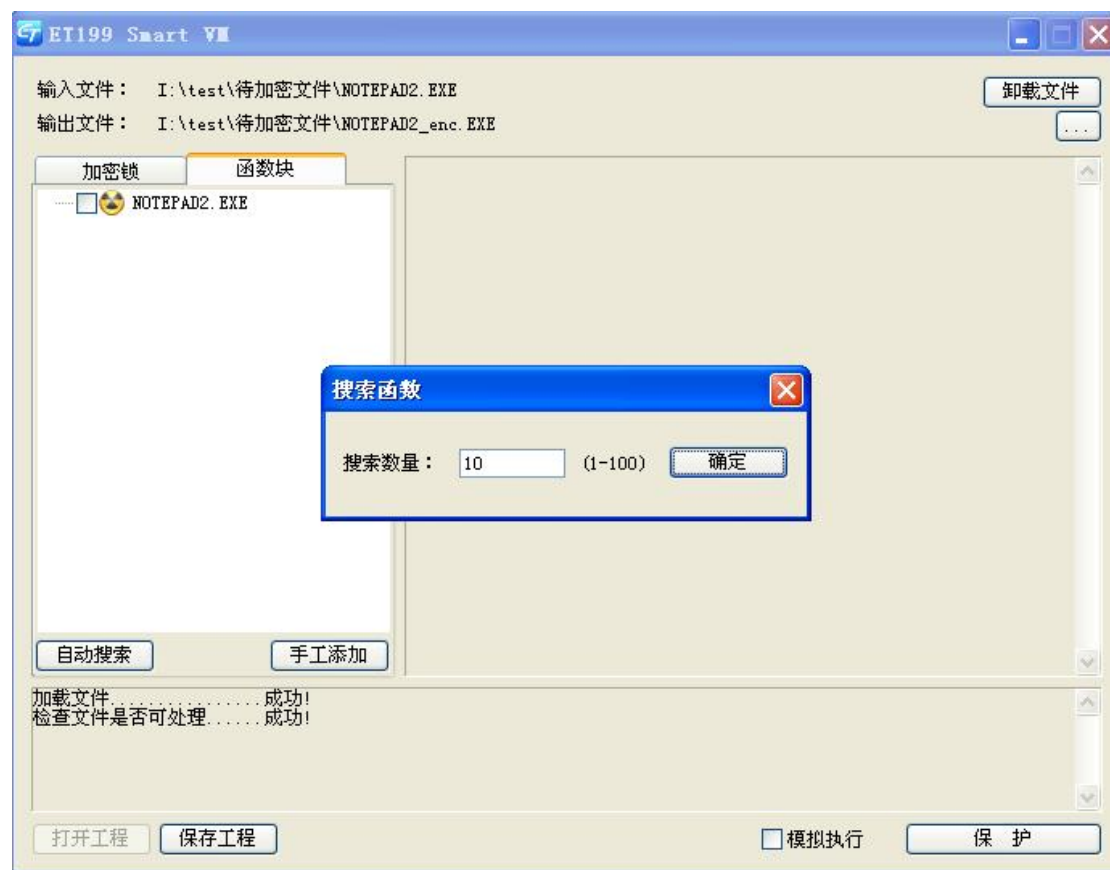


图 3.2-5 函数定位

点击确定按钮后会显示 10 个自动搜索到的函数，且界面右边区域会显示函数的具体内容，默认显示为第一个函数的具体内容，点击其他的函数可以查看相应函数的具体内容，如图 3.2-6。

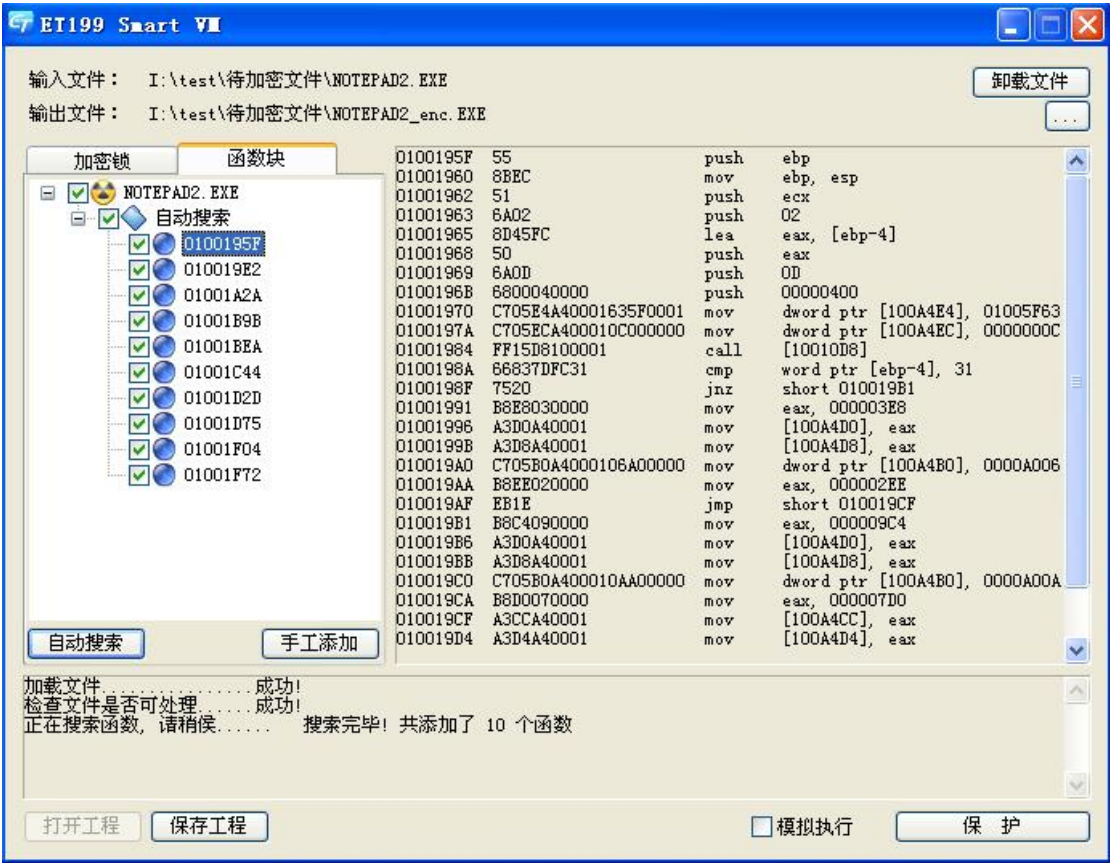


图 3.2-6 搜索到的函数具体信息

(6) 开始加密。点击右下角“保护”按钮开始加密。加密成功的提示信息将显示在“信息栏”中，如图 3.2-7，此时会在设置的输出路径目录中产生加密后的文件。

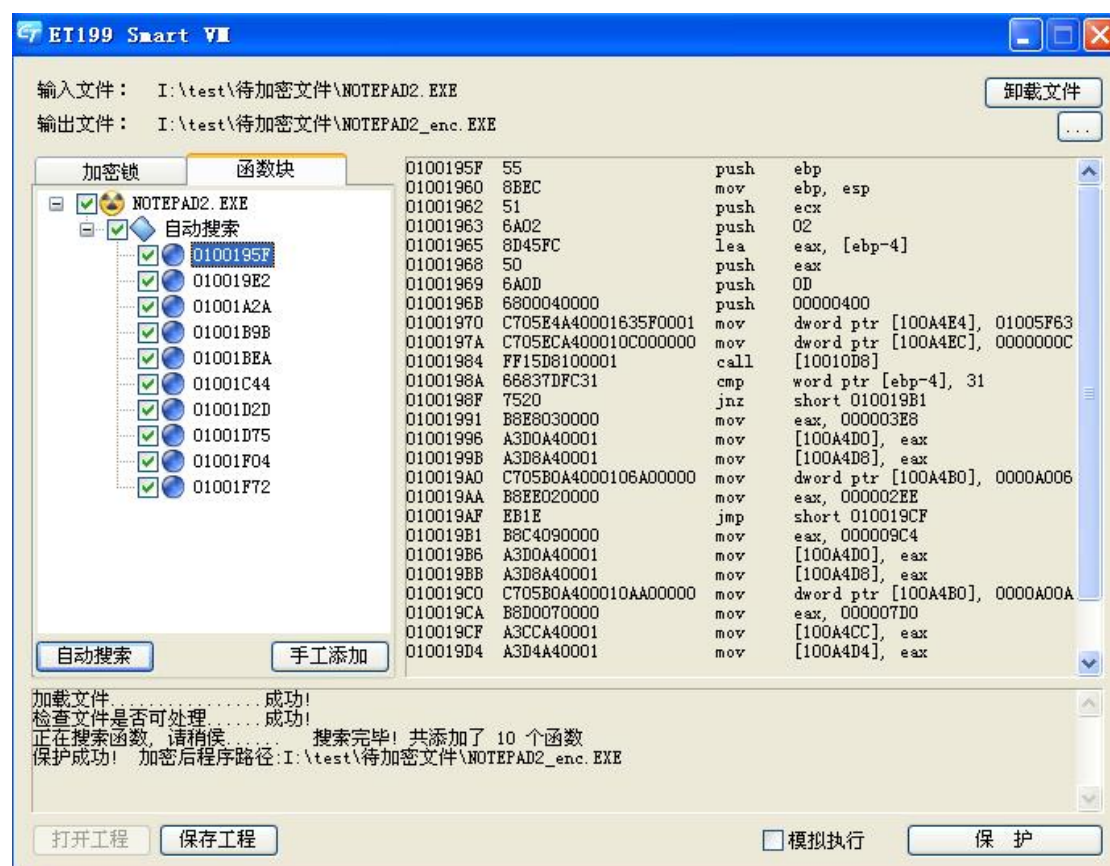


图 3.2-7 加密完成界面

加密注意事项:

1. 应该先选中“模拟执行”进行加密，加密后试运行，以便得知哪些函数被执行到了，避免因搜索到的函数根本没有被执行到，导致加密后的文件在不插锁的情况下也能运行。

选上“模拟运行”，然后按“保护”按钮完成加密。



这时运行加密后的文件，如：NOTEPAD_enc.EXE，会看到臂调用到的模块后面有数字，这个就是这段代码被调用的次数。同时在信息栏中会有执行的过程。如下图：



2. 在模拟执行的情况下，对于运行次数太多的函数建议取消这些函数的保护，以改善软件的性能和执行速度，这个完全可由用户在模拟执行时根据显示的函数执行次数自行体会决定。

3. 如果模拟执行时发现软件崩溃，可查看运行信息获知最后执行到哪个函数，取消对此函数的保护，重新加密，看是否能正常运行，依此类推，可快速找出虚拟化出错的函数并予以剔除。此时建议用户在事后把程序样本和出错的函数地址反馈给我们，以便我们查找原因，进行改进。

4. 经过模拟执行后，筛选出需要保护的函数，进行正式加密，这时加密后的程序才具有可靠性、稳定性，才能达到实际的加密效果。

5. 加密时如果选择了与 HID 绑定，那么加密后的程序只能被加密时的锁打开。

6. 使用时间加密后，当时间到期后，可以再次使用 VMProtect.exe 工具设置没有时间保护，或者新的到期时间，加密后，将加密后的新程序发给客户。

3.3 使用加密后的文件

要使用加密后的文件，首先应确保 ET199 加密锁已经连接在当前计算机上，然后启动加密后的可执行程序。如果没有插入加密锁或使用不对应的加密锁，会提示找不到指定的加密锁。

第四章 自定义保护段

可以在您的程序中使用“输入定位”和“宏定位”两种方式加上要保护代码的标识，这样 VMProtect.exe 工具中就能按照自定义标识将标识之间的代码进行虚拟保护了。如：

- 输入定位：

工程中需要加入 FTVMPSDK32.dll，该 DLL 文件的导出函数声明在 FTVMPSDK32.h 头文件中。即需要 FTVMPSDK32.h 头文件，FTVMPSDK32.dll 库文件和 FTVMPSDK32.lib 动态 lib 文件。

代码中：

```
//加密起始
VMProtectBegin();
//要加密的代码
tmp = atoi(pstr) + 100;
.....
//加密结束
VMProtectEnd();
```

- 宏定位：

需要加入宏定义，如下：

```
#define VMP_START \
__asm __emit 0xEB \
__asm __emit 0x10 \
__asm __emit 0x56 \
__asm __emit 0x4D \
__asm __emit 0x50 \
__asm __emit 0x72 \
__asm __emit 0x6F \
__asm __emit 0x74 \
__asm __emit 0x65 \
__asm __emit 0x63 \
__asm __emit 0x74 \
```

```
__asm __emit 0x20 \  
__asm __emit 0x62 \  
__asm __emit 0x65 \  
__asm __emit 0x67 \  
__asm __emit 0x69 \  
__asm __emit 0x6E \  
__asm __emit 0x00 \  
  
#define VMP_END \  
__asm __emit 0xEB \  
__asm __emit 0x0E \  
__asm __emit 0x56 \  
__asm __emit 0x4D \  
__asm __emit 0x50 \  
__asm __emit 0x72 \  
__asm __emit 0x6F \  
__asm __emit 0x74 \  
__asm __emit 0x65 \  
__asm __emit 0x63 \  
__asm __emit 0x74 \  
__asm __emit 0x20 \  
__asm __emit 0x65 \  
__asm __emit 0x6E \  
__asm __emit 0x64 \  
__asm __emit 0x00 \  

```

代码中：

```
//加密起始  
VMP_START  
//要加密的代码  
tmp = a * 5 + b;  
tmp = tmp * 12;  
.....  
//加密结束  
VMP_END|
```