华通闪存存储加密软件 V3.0

详细设计说明书

北京华通伟业科技发展有限公司

二〇一七年五月

目　　录

1 项目概述

华通闪存存储加密软件 V3.0 是基于 Windows 操作系统，能够在闪迪存储卡设备上运行的系统安全信息方面的管理软件。本产品在原华通伟业闪迪存储加密软件 V2.0 的基础上进行了技术升级和功能的优化，扩展了底层核心服务组件的设计和代码策略，对于性能提升和存取速率进行的大幅调节。

依然结合存储卡硬件技术实现了对文件的加密、解密。使用强大的加密算法，系统底层编程技术为客户提供了完美的加密、解密方案。

即使存储卡丢失也不用担心里面的数据被泄漏，提供一种绝对性的保密文件方法。该系统闪迪存储加密是一款支持 Windows Vista/XP/2000/Linux 的闪迪存储卡加密软件，能够创建和设置加密的虚拟磁盘镜像，虚拟磁盘可以与其它磁盘一样正常访问，内部所有文件都会自动加密，需要通过密码来进行访问，加密和解密都是实时的。

本软件产品能够基于 filedisk 驱动技术实现的虚拟磁盘，将我们所使用的文件拷贝进去。然后使用文件加密技术加密。使用 UltraISO 制作出将要拷贝到存储卡上的内容的 ISO 文件。使用 SD 存储卡量产的工具将这个 ISO 制作出来。这样 U 盘插入可以出现 2 个盘符，一个就是我们的这个 ISO 文件，此时使用我们的上层软件输入正确密码就可以将我们加密后的文件显示在一个虚拟磁盘中，这样我们就可以任意访问，甚至删除，但是再次插入存储卡，里面的内容依然还是在的。此软件满足了我们的实际需要，是一个专业、安全、实用的存储卡加密系统。

该系统是集体智慧的结晶。系统开发自始至终采取“安全至上、简单实用”的开发原则；自始至终采取一线业务人员直接参与，技术专家深入一线的开发路线；自始至终采用优化管理，着眼未来的开发思路。再加上业务人员对功能不断追求，技术人员对品质精益求精，双方团结协作、紧密配合、相互补充，成功地实现了管理与技术的完美结合

1.1 编写目的

本说明书在《华通闪存存储加密软件》概要设计说明书的基础上，进一步对闪迪存储加密存储卡系统加密各模块、程序分别进行了详细的要求和说明，并作为程序开发者的主要依据。

1.2 项目背景

a．项目名称：华通闪存存储加密软件 V3.0；

b．项目功能：实现闪迪存储卡系统加密、解密，加密文件挂载和卸载；

c．项目任务提出者：北京华通伟业科技发展公司

d．项目开发者：

e．该项目的用户：闪迪 U 盘和存储卡的使用人

f．有联系的软件：VS 和 WDK。

1.3 定义

本文涉及到的专业术语有：Microsoft Visual 开发工具，Microsoft Windows Driver Kit，Windows7 运行环境，NASM，gzip。

2 总体设计

2.1 需求概述

程序分为用户操作层，以及驱动实施层。驱动实施层又分为磁盘虚拟模块，读写请求和加解密模块。程序实现功能，可概述为：1.创建功能，用户通过指定文件路径，大小，密码创建用于虚拟磁盘的存储空间。2，挂载磁盘，用户通过输入磁盘存储空间来源的文件路径，密码和盘符等参数，来挂载虚拟的磁盘。3，卸载磁盘，用户指定虚拟磁盘的盘符之后，即可通知驱动卸载磁盘。

系统在处理设备请求（例如，密码初始化、挂载/卸载）和读写请求时，主要是采用串行的结构。对于同一设备上述两种请求，皆调用同一线程来处理。线程在驱动设备创建时，就开始运行，接受来自上层的请求。

来自上层的请求皆加入到同一任务队列中，等待处理线程一一处理。在任务队列为空的情况下，线程处于空循环状态，等待请求的到来，然后判断请求类型，调用相应的函数进行处理。线程在驱动启动时被创建，在驱动结束时，通过改变结束符为真，使得线程进行自我终止。

虽然程序在处理上采用了单线程的结构，在磁盘有多个文件需要进行读写时会对效率有一定的影响，但是这样单线程的结构，更易于系统的实现，结构更为

简单，不容易出现错误，导致系统的不稳定。同是也避免了同步模式下的死锁的情况，即用损失效率的代价换得了系统的稳定。

系统对于硬件性能，没有很高的要求。从系统上来讲，由于采用的是 Windows 的驱动所以必须为 Windows 的系统，从版本上来讲，win7 与 XP 皆可运行，不同的是编译的版本是不一样的，即两个系统下代码是相同的，只不过编译器版本不同。对于存储要求，即用户磁盘的可用空间必须满足用户所需虚拟挂载磁盘的大小。由于本系统采用 DES 算法的真加密，所以系统的处理速度将影响磁盘的读写速度，对于具体的处理能力没有具体要求。

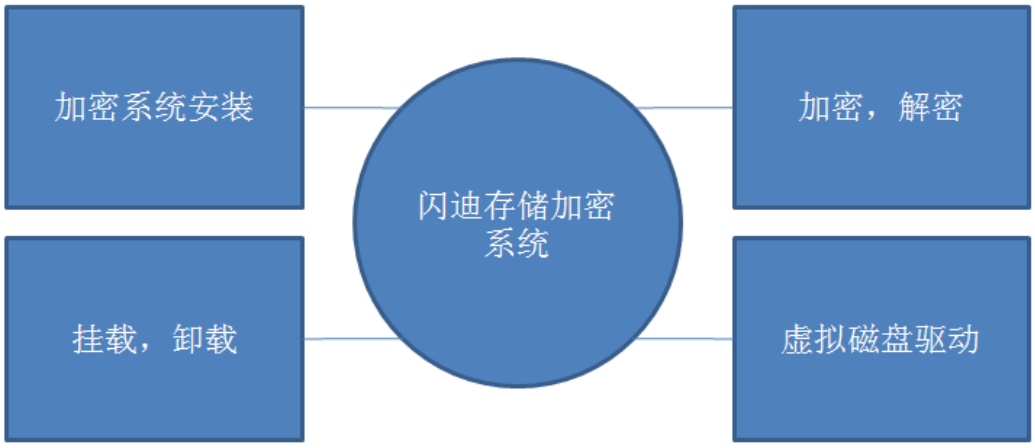
从精度上讲，根据在 Win7 和 XP 上运行的结果，读写请求的数据块大小，总是 8byte 的倍数，这刚好满足 DES 分组加密的大小规格，即不需要进行补位，从而，节省了系统资源。

灵活性上，程序对磁盘文件的大小一旦指定就不可改变，不可以再扩展大小。而对于密码的更改，可直接调用 DES 加解密程序，对文件进行密码更改即可。

时间特性上讲，对于新建的磁盘，需要格式化，这是需要做大量的加解密工作，所以耗的时间相对正常的格式化，将会较慢。而对于已格式化的磁盘文件，在挂载时，耗时是较短的。在读写时，也是耗时较短的，用户感受不到加解密操作的存在。

华通伟业U盘和内存卡等的移动存储设备已经成为了信息交流中不可或缺的介质之一。日常工作生活和学习以及商务方面也都需要通过移动硬盘和 U 盘来传递共享数据信息，数据安全性这个问题由于先前没有造成太大的影响，所以一直被人们所忽略。但其实这些设备都有潜在的遗失或被盗的风险，一旦设备丢失或数据被盗，其保存信息的泄露就可能造成难以预计的损失。随着近年来各式各类的典型门事件如雨后春笋般的频发，数据的安全性一时间也被推向了风口浪尖，消费者对数据信息的保护意识迅速提高。随着个人隐私日益受到用户关注，那么如何保障信息安全有效的流通，成为了大众关注的焦点。而我们的华通闪迪存储加密软件 V3.0 是在闪迪存储卡硬件的基础上，使用底层加密的软件加密方式，力争做到数据安全，使用简单，费用低廉。存储卡的存储容量基本可以满足现在一般的需要，有 1G，4G，8G，16G，128G 等等。软件加密采用的是国际上著名的开源的驱动代码为基础，结合我们的实际进一步开发的软件。具有安全可靠，简单实用等等优点。

2.2 存储加密软件架构



图表 2-1 存储加密软件架构图

1. 加密系统安装：将要使用的应用程序和驱动安装到 window 系统中。
2. 加密，解密：各种加密算的代码实现。
3. 挂载，卸载：应用程序发出挂载，卸载命令将加密文件挂载成一个磁盘，或者卸载此磁盘。
4. 虚拟磁盘驱动：是 windows 上的驱动代码部分，是在 FileDisk 的基础开发出来的。

2.3 存储卡编译工具安装

Visual Studio 2008 默认安装。

Visual Studio 2008 SP1 默认安装。

VC1.5 ++解压到某个目录。

Windows Driver Kit Version 7.1.0 默认安装。

将 gzip.exe 文件解压出来，放到 system32 目录里。

将 nasm.exe 文件解压出来，放到 system32 目录里。

PKCS#11 解压到某个目录。

2.4 存储卡编译环境变量设置

创建一个环境变量 MSVC16_ROOT 指向 VC1.5 解压目录。

创建环境变量 WINDDK_ROOT 指向 WDK 目录，如我的 WDK 目录是 C:\WinDDK\7600.16385.1。

创建环境变量 PKCS11_INC 指向 PKCS#11 的头文件解压目录。

3 闪迪存储加密安装程序描述

3.1 功能

闪迪存储加密安装程序的功能就是将闪迪存储加密需要的驱动和应用程序安装或者解压到 windows 系统中。

要支持多种系统：windows2000 windowsXP windows7 Vista。

3.2 性能

达到反复安装，卸载 1000 次无问题。

3.3 输入项目

这个程序会要求接受安装协议，和选择安装或者解压的目录，和另外一些信息设置，例如创建桌面快捷方式等等。

3.4 输出项目

输出就是安装后的应用和驱动程序。

3.5 算法

本程序是存储卡安装程序通过界面的操作流程进行一些信息设置，从而完成程序安装。

If（是否接受安装协议）

无法安装；

else

if（选择 Install 方式）

if（选择 Install 目录）

选择安装使用者；

选择添加安装程序到 StartMenu；

选择添加快捷方式；

选择备份系统还原点；

Else

无法安装成功；

else if（选择 Extract 方式）

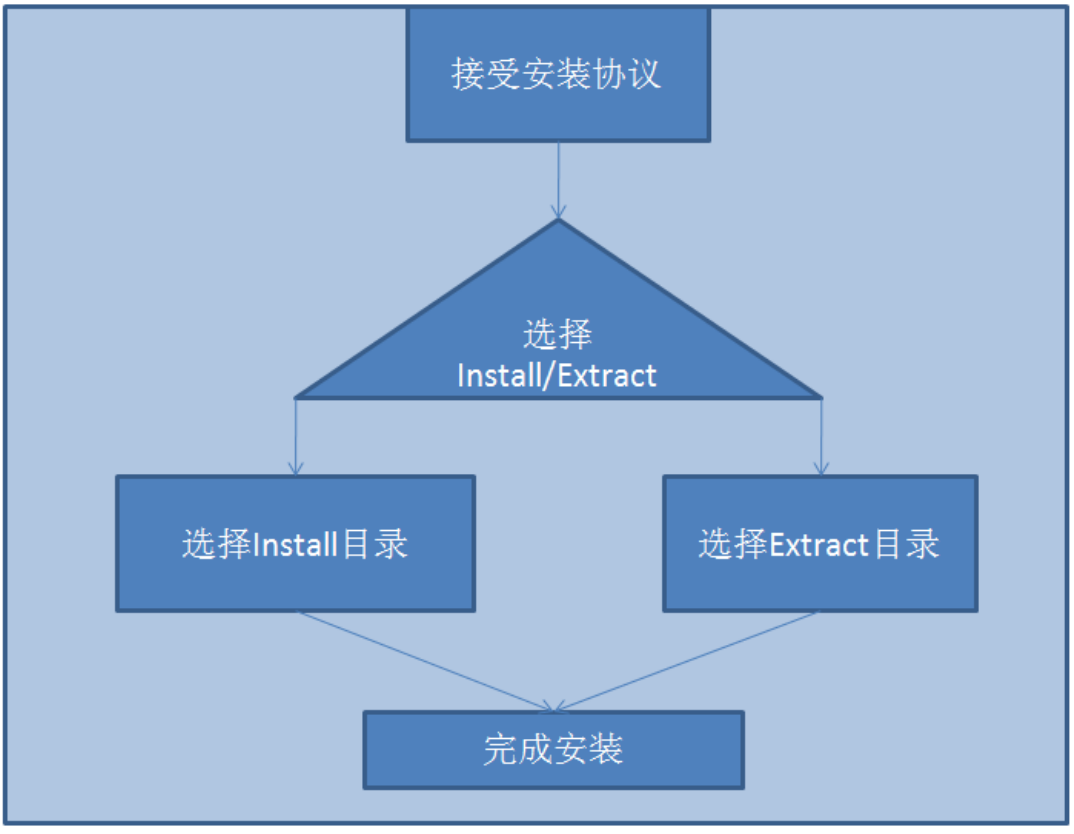
if（选择 Extract 目录）

选择打开 Extract 后的目录；

else

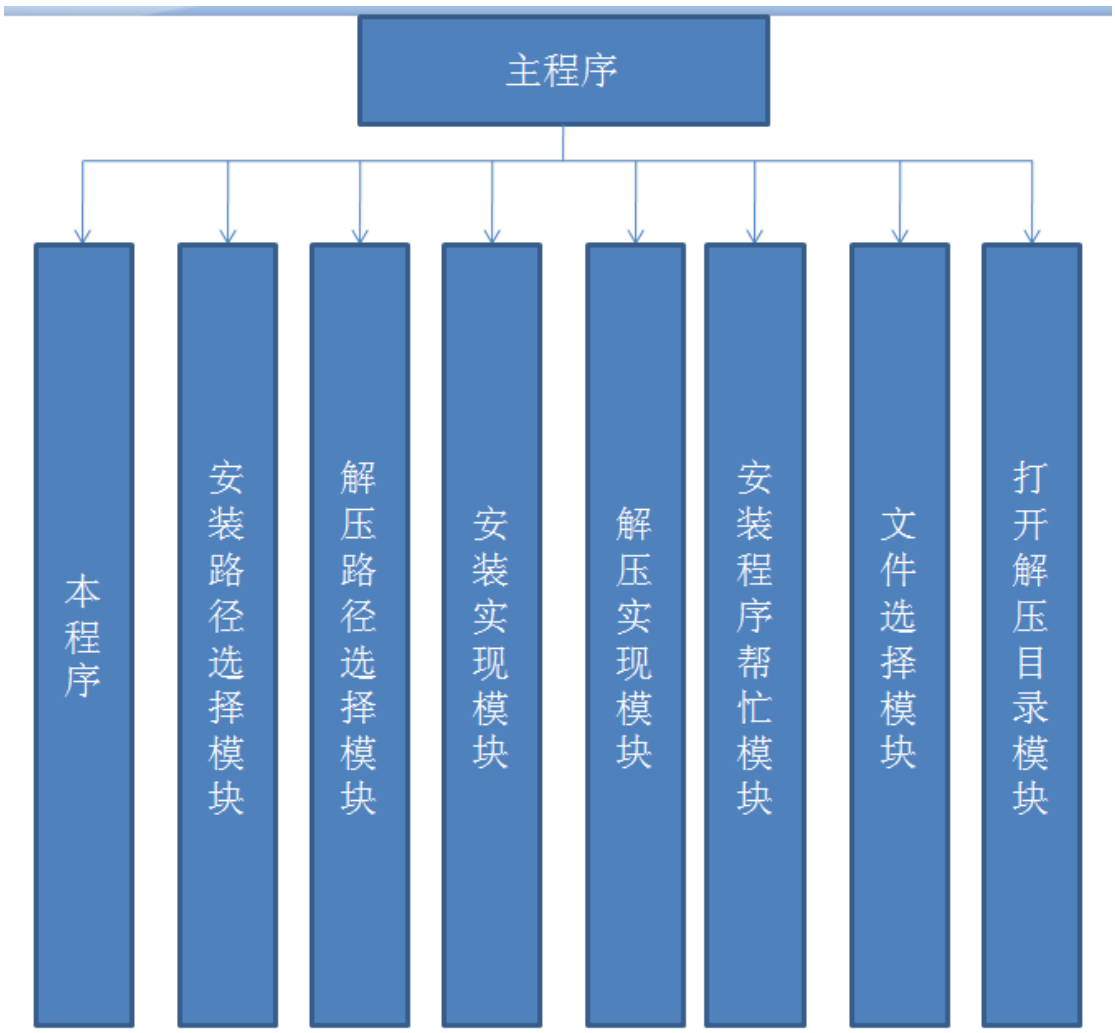
Extract 不成功；

3.6 程序流程图



这个模块的流程逻辑是很简单的，简单的将就是一个程序拷贝的过程而已。

3.7 接口



3.8 存储区域

会将驱动名称和驱动安装目录保存到注册表中。

3.9 限制条件

只适用在 windows 系统上。

3.10 测试要点

是否支持 windows2000 windowsXP windows7 Vista 等等系统安装。安装后的应用程序和驱动是否正常使用，桌面快捷方式等等是否可用。

3.11 关键函数代码介绍

int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance, char *lpszCommandLine, int nCmdShow)

WinMain 函数是闪迪存储加密安装程序的代码入口函数。根据 win32 工程模板原则开始代码执行。

参数是 win32 工程模板传入的。不需要关心。

返回值也是系统调用不需要关心。

这里面的代码回设置安装目录，调用安装，或者解压的代码完成功能。

static void SetSystemRestorePoint (HWND hwndDlg, BOOL finalize)

SetSystemRestorePoint 函数比较特殊，是完成系统还原点的函数。

下面我们来看下每个参数具体的含义

参数	类型	含义	示例
hwndDlg	HWND	传入的窗口句柄	NULL
finalize	BOOL	是否设置还原点	FALSE

函数无返回值。

函数根据参数通过调用_SRSetRestorePoint 函数来完成还原点设置。

4 加密程序描述

4.1 功能

此模块是本系统的关键部分，他通过界面一步一步将加密需要的所有信息设置出来，其中包括要加密的文件，设置加密模式，加密文件显示方式，加密算法，哈希表，加密后文件的大小，加密密码设置，加密后盘符的文件格式等等完成加密。

4.2 性能

a) 加密算法的性能：由于我们使用的是国际上最常用的加密算法，所以性能是有保证的。
b) 格式化过程性能：格式化的速度跟要格式化的大小有很大的关系。

4.3 输入项目

字段名称	说明
加密的文件名	指明要加密的文件
闪迪存储加密加密模式	有文件，非系统盘加密，系统盘加密
闪迪存储加密加密文件显示方式	普通模式和隐藏模式
加密算法	AES Twofish 等等
哈希表模式	SHA-512 等等
加密后文件大小	296K 以上到几 G。
密码设置	可以设置密码，或者使用密码文件
加密盘的文件格式	FAT NTFS 等等

4.4 输出项目

加密后的文件

4.5 算法

本程序是加密程序通过界面的操作流程进行一些信息设置，从而实现文件加密。

If（NO 选择加密文件）

无法安装；

else

if（选择文件加密模式）

if（选择文件加密后显示模式）

if（选择文件加密后保存目录）

if（选择是否保存历史记录）

会保存历史记录，以便后面使用

Else

无法保存历史记录；

if（选择文件加密算法）

switch(加密算法)

case AES：

选择 AES 加密算法

case Serpent-Twofish：

选择 Serpent-Twofish 加密算法

case Twofish：

选择 Twofish 加密算法

case AE-Twofish：

选择 AE-Twofish 加密算法

case Serpent-AES：

选择 Serpent-AES 加密算法

case Twofish-Serpent：

选择 Twofish- Serpent 加密算法

case AES-Twofish-Serpent：

选择 AES-Twofish-Serpent 加密算法

测试加密算法是否正确

设置要加密的字符；

设置 AES 的 XTS 模式；

测试加密是否正确

选择哈希表模式

switch(哈希表模式)

case Whirlpool：

选择 Whirlpool 哈希表模式

case SHA-512：

选择 SHA-512 哈希表模式

case RIPEMD-160：

选择 RIPEMD-160 哈希表模式

if（设置加密文件的大小）

if（设置加密密码）

if（格式化）

if（选择文件系统格式）

Else

无法安装成功；

Else

无法安装成功；

Else

无法安装成功；

if（是否使用密钥文件）

选择一个密钥文件。

生成随机密钥文件

Else

无法安装成功；

Else

无法安装成功；

Else

无法安装成功；

Else

无法安装成功；

else if（选择非系统盘加密）

if（选择文件加密后显示模式）

if（选择要加密盘）

if（选择是否保存历史记录）

会保存历史记录，以便后面使用

Else

无法保存历史记录；

if（选择文件加密算法）

switch(加密算法)

case AES：

选择 AES 加密算法

case Serpent-Twofish：

选择 Serpent-Twofish 加密算法

case Twofish：

选择 Twofish 加密算法

case AE-Twofish：

选择 AE-Twofish 加密算法

case Serpent-AES：

选择 Serpent-AES 加密算法

case Twofish-Serpent：

选择 Twofish- Serpent 加密算法

case AES-Twofish-Serpent：

选择 AES-Twofish-Serpent 加密算法

测试加密算法是否正确

设置要加密的字符；

设置 AES 的 XTS 模式；

测试加密是否正确

选择哈希表模式

switch(哈希表模式)

case Whirlpool：

选择 Whirlpool 哈希表模式

case SHA-512：

选择 SHA-512 哈希表模式

case RIPEMD-160：

选择 RIPEMD-160 哈希表模式

if（设置加密文件的大小）

if（设置加密密码）

if（格式化）

if（选择文件系统格式）

Else

无法安装成功；

Else

无法安装成功；

Else

无法安装成功；

if（是否使用密钥文件）

选择一个密钥文件。

生成随机密钥文件

Else

无法安装成功；

Else

无法安装成功；

Else

无法安装成功；

Else

无法安装成功；

else if（选择系统盘加密）

if（选择文件加密后显示模式）

if（选择系统目录）

if（选择是否保存历史记录）

会保存历史记录，以便后面使用

Else

无法保存历史记录；

if（选择文件加密算法）

switch(加密算法)

case AES：

选择 AES 加密算法

case Serpent-Twofish：

选择 Serpent-Twofish 加密算法

case Twofish：

选择 Twofish 加密算法

case AE-Twofish：

选择 AE-Twofish 加密算法

case Serpent-AES：

选择 Serpent-AES 加密算法

case Twofish-Serpent：

选择 Twofish- Serpent 加密算法

case AES-Twofish-Serpent：

选择 AES-Twofish-Serpent 加密算法

测试加密算法是否正确

设置要加密的字符；

设置 AES 的 XTS 模式；

测试加密是否正确

选择哈希表模式

switch(哈希表模式)

case Whirlpool：

选择 Whirlpool 哈希表模式

case SHA-512：

选择 SHA-512 哈希表模式

case RIPEMD-160：

选择 RIPEMD-160 哈希表模式

if（设置加密文件的大小）

if（设置加密密码）

if（格式化）

if（选择文件系统格式）

Else

无法安装成功；

Else

无法安装成功；

Else

无法安装成功；

if（是否使用密钥文件）

选择一个密钥文件。

生成随机密钥文件

Else

无法安装成功；

Else

无法安装成功；

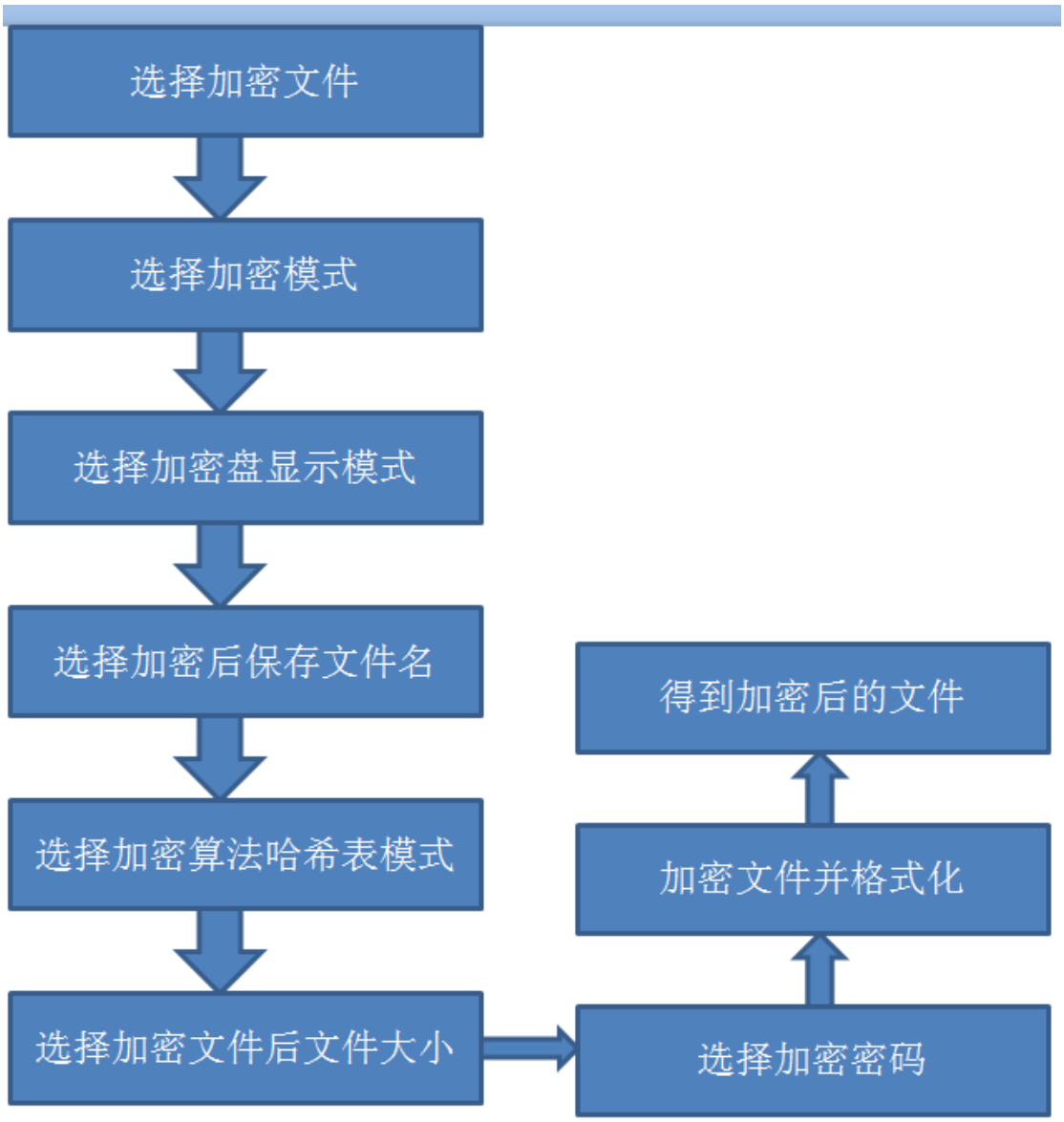
Else

无法安装成功；

Else

无法安装成功；

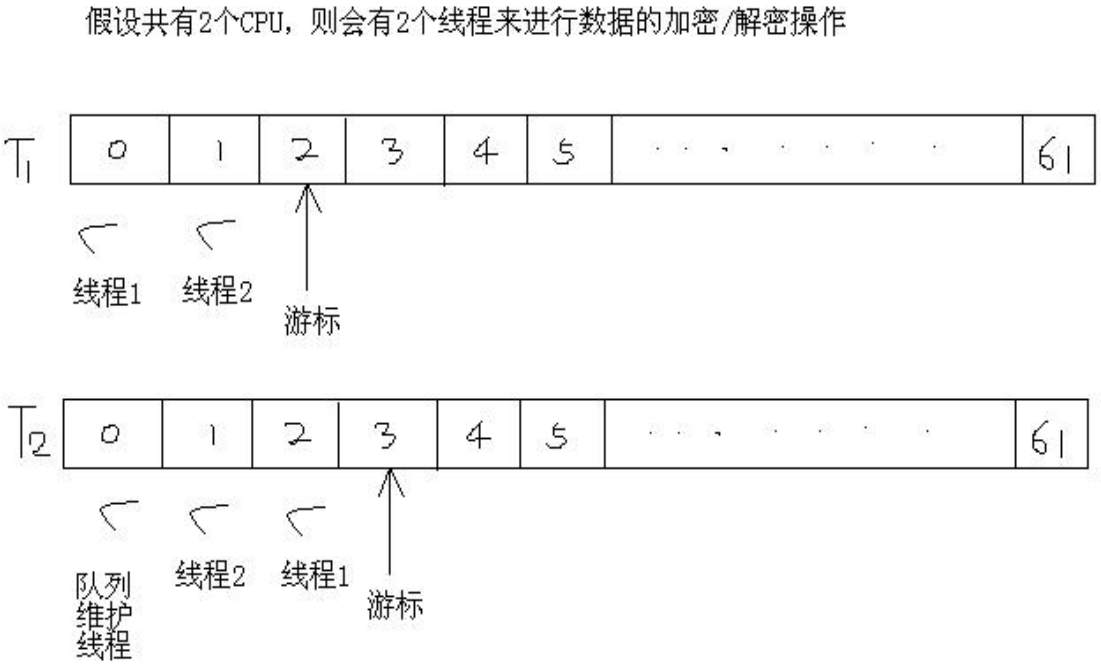
4.6 程序流程图



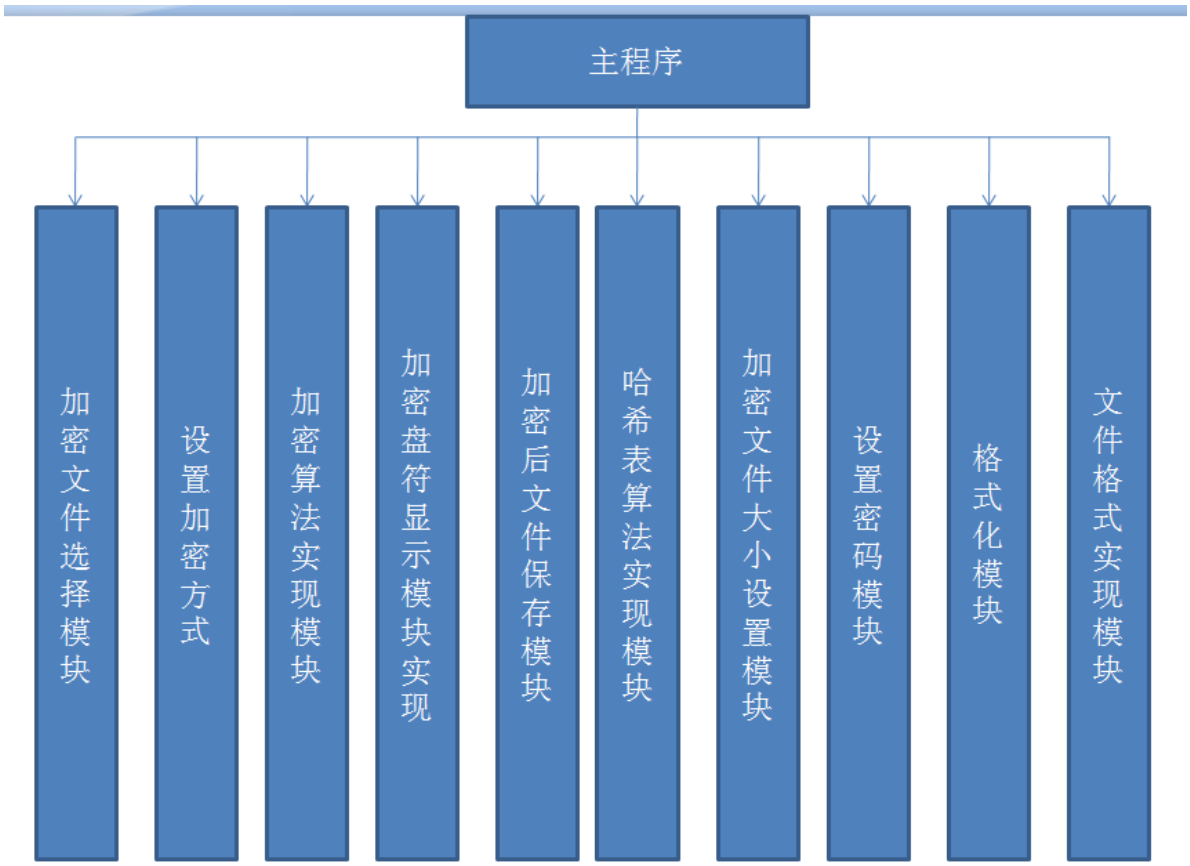
闪迪存储加密系统通过 Windows 操作系统加载，模块驱动在入口接口中作了初始化：初始化了密码算法、加载了启动参数并创建了业务框架。模块驱动初始化完成后，整个闪迪存储加密模块分为了两部分：与 Windows 操作系统交互部分和业务流程部分。与 Windows 操作系统交互部分通过创建设备对象来与 Windows 操作系统的 I/O 管理器进行数据交互；业务流程部分通过业务架构进行工作；他们之间的交互通过队列来传递数据流。这样基本形成了从上到下和从下到上两种数据流，共 cpu 个数 * 2 条数据流，每条数据流通过队列和线程池实现了半异步和半同步的设计模式。闪迪存储加密模块启动后，创建了业务框架，其中最重

要的数据处理流程(数据的加密/解密/密钥生成)。闪迪存储加密模块这部分设计很巧妙，使用了半同步/半异步的设计模式，并结合了多核条件下数据同步技术。

闪迪存储加密根据 CPU 个数，创建了相应的线程(我们这里给叫做数据处理线程)。这些线程开始时分别选择一个工作队列，等待数据的到来。一旦有数据进入队列，根据数据的类型，进行相应的处理；处理完毕后，通知队列的状态维护线程数据处理完毕，请维护队列状态；然后自己根据游标选择下一个要处理的队列进行处理；为什么不直接选择刚处理完的队列呢？因为此时刚处理完的队列状态还没有恢复到准备阶段，为了提高处理速度，选择下一个准备好的队列，进行处理。具体请看草图：



4.7 接口



4.8 存储分配

a.保存加密后的文件

b.注册表保存驱动，加密密码等等信息。

4.9 限制条件

必须将设置的密码记住，或者密匙文件不能丢失。

4.10 测试要点

1）加密方式测试

2）各种加密算法测试

3）密匙文件测试

4）哈希表选择测试

5）加密文件大小测试

6）加密后的显示模式测试

7）加密后文件系统测试

8）加密文件测试

4.11 关键函数代码介绍

闪迪存储加密的核心就是创建加密卷函数了，该函数的作用就是接受上述的加密卷位置等几个参数：密卷位置，加密算法、加密卷大小、加密卷密码、加密卷格式化，完成创建加密卷的过程。

在闪迪存储加密里的原型是：

int

TCFormatVolume

(volatile FORMAT_VOL_PARAMETERS *volParams)

volParams 参数其实是一个关键的结构体。其结构体变量入下表:

参数	类型	含义	示例
volumePath	char *	用户选择的加密卷文件	C://1.txt
bDevice	BOOL	是否是一个分区卷	FALSE
size	unsigned __int64	加密卷大小（以 byte 为单位）	10*1024*1024
hiddenVolHostSize	unsigned __int64	隐藏加密卷大小（以 byte 为单位）	0
password	Password	用户输入的密码	123456
cipher	int	默认加密的算法	1
pkcs5	int	哈希算法	DEFAULT_HASH_ALGORITHM
quickFormat	BOOL	是否是快速格式化	FALSE
sparseFileSwitch	BOOL		TRUE
fileSystem	int	文件系统	FILESYS_FAT
clusterSize	int		0

hwndDlg	HWND	传入的窗口句柄	NULL
hiddenVol	BOOL	是否是隐藏卷	FALSE
realClusterSize	int		512
uac	BOOL		FALSE

TCFormatVolume 函数返回结果为

0 加载成功

1 加载失败

2 在共享模式加载成功

static TC_THREAD_PROC EncryptionThreadProc (void *threadArg)

EncryptionThreadProc 是加密调度实现的关键函数。

1.设置队列保护机制，保护珍贵的队列资源。

2.根据 CPU 个数和队列机制完成加密，解密算法的合理调度。

void DecryptDataUnitsCurrentThread (unsigned __int8 *buf, const UINT64_STRUCT *structUnitNo, TC_LARGEST_COMPILER_UINT nbrUnits, PCRYPTO_INFO ci)

DecryptDataUnitsCurrentThread 函数是实现加密算法的函数，此时与其他模块无关。

下面我们来看下每个参数具体的含义

参数	类型	含义	示例
Buf	unsigned __int8	要加密的字符串	“123”
structUnitNo	UINT64_STRUCT	加密密匙结构体	
nbrUnits	TC_LARGEST_COMPILER_UINT	Nbr 结构体	
ci	PCRYPTO_INFO	加密算法信息	AES

DecryptDataUnitsCurrentThread 函数无返回值。

5 挂载程序描述

5.1 功能

将加密后的文件挂载为一个盘符，这样就可以像普通盘一样的操作使用。可以将我们需要的一些要保存的文件放入到这个盘符中。当然拷贝进去的文件大小不能超过我们设置的加密文件的大小。卸载加密文件后就将拷贝进去的文件保存下来，以便后面使用。

5.2 性能

挂载，卸载的速度很快。而且反复挂载，卸载 1000 次无问题出现。

5.3 输入项目

加密后的文件。

5.4 输出项目

显示挂载盘符

5.5 算法

本程序是存储卡加密文件挂载，卸载程序，在输入正确密码以后就可以挂载加密文件为一个盘，可以正常操作这个磁盘。

If（NO 点击 Mount 按钮）

无法安装；

else

If（选择显示密码）

输入的密码将显示出来；

else

输入的密码不能显示出来；

If（选择密匙文件）

If（输入正确的密匙文件）

挂载成磁盘；

else

不能挂载出磁盘；

else

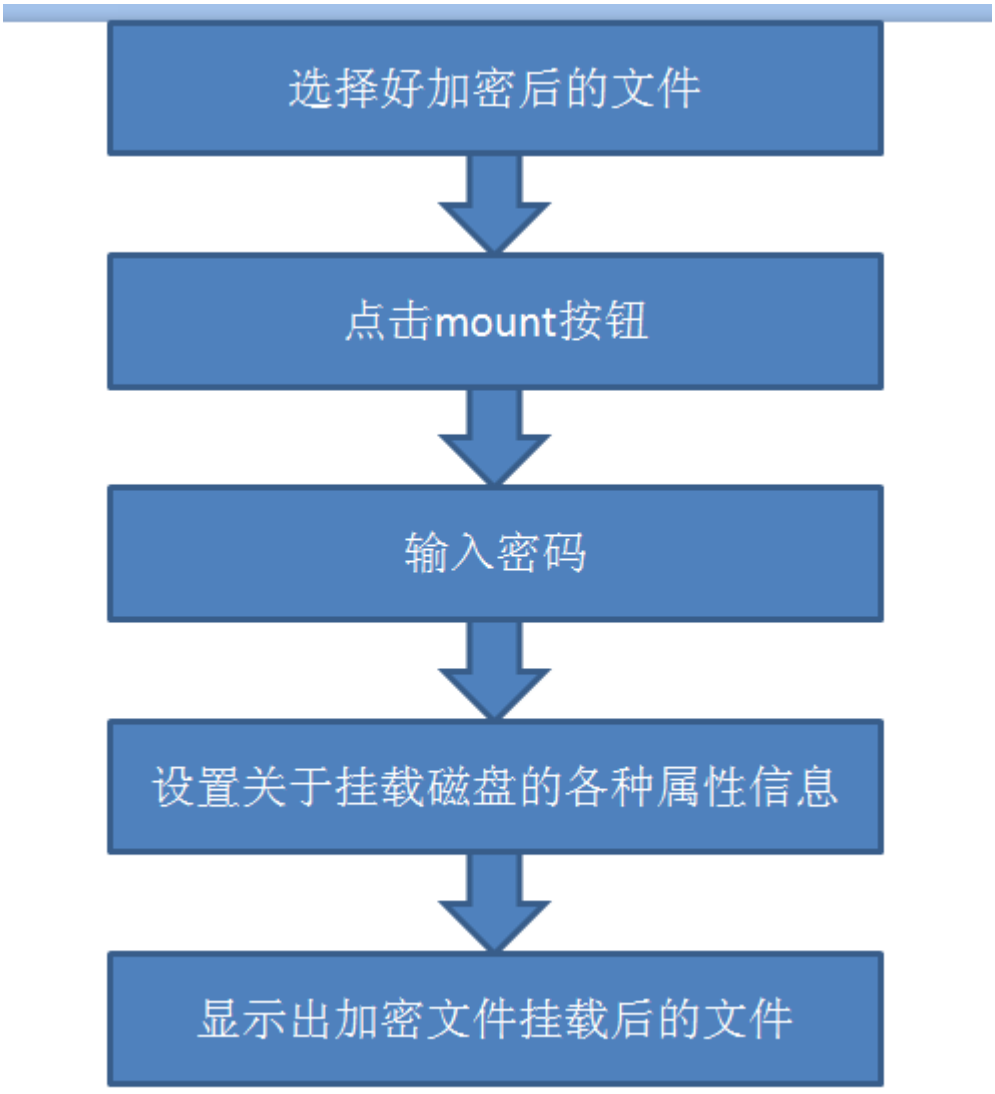
输入的密码不能显示出来；

设置挂载后的属性为隐藏，移动磁盘属性。

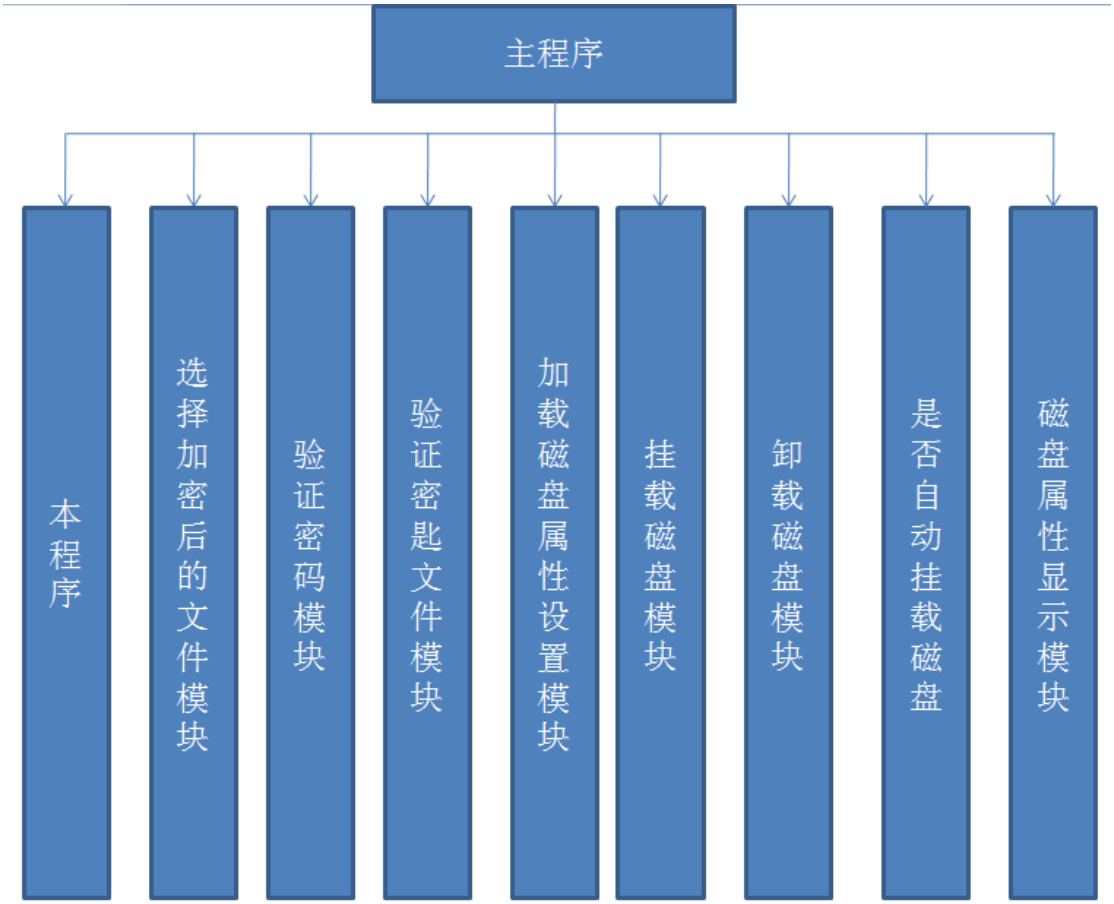
if（输入正确的密码）

显示出正确的磁盘

5.6 程序逻辑



5.7 接口



5.8 存储分配

a.保存加密后的文件

b.注册表保存驱动，加密密码等等信息。

5.9 限制条件

必须输入正确的密码或者加密后的密匙文件。

5.10 测试要点

1）选择加密文件测试

2）验证输入密码是否正确测试

3）测试价加载磁盘的属性设置是否正确

4）挂载，卸载是否正确

5.11 关键函数代码介绍

在创建完加密卷之后，怎么使用虚拟磁盘呢？这个时候需要加载虚拟磁盘，加载完虚拟磁盘后，就会在磁盘分区里多出一个分区，比如在我的电脑里可以看到一个新的本地磁盘 (L:)。那么，闪迪存储加密里具体是怎么实现的？首先需要选择我们在上一步创建的加密卷，选择一个需要加载的分区，这其实就是在还没用到的分区卷标里选择一个，之后点击载入载入即可，这个时候就会找到加密卷进行加载，这时会需要用户输入创建加密卷时的设定的密码，加载成功后，就能看到新虚拟出来的磁盘了。因此，闪迪存储加密另一个核心函数加载虚拟磁盘函数应该是接受加密卷、磁盘分区、用户密码等参数进行加载，该函数在闪迪存储加密的原型是

```
int MountVolume (HWND hwndDlg,
                 int driveNo,
                 char *volumePath,
                 Password *password,
                 BOOL cachePassword,
                 BOOL sharedAccess,
                 const MountOptions* const mountOptions,
                 BOOL quiet,
                 BOOL bReportWrongPassword);
```

下面我们来看下每个参数具体的含义

参数	类型	含义	示例
hwndDlg	HWND	传入的窗口句柄	NULL
driveNo	int	加载的磁盘分区序号	8
volumePath	char *	用户选择的加密卷文件	C://1.txt
password	Password	用户输入的	123456

		密码	
cachePassword	BOOL	是否采用缓存密码	FALSE
sharedAccess	BOOL		FALSE
mountOptions	MountOptions	加载选项	
quiet	BOOL		FALSE
bReportWrongPassword	BOOL	是否报告错误的密码	TRUE

MountVolume 函数返回结果为

-1 用户终止加载

0 加载失败

1 加载成功

2 在共享模式加载成功

成功虚拟磁盘后，即会弹出虚拟出的磁盘，可以在虚拟磁盘里创建文件，在虚拟磁盘里的文件都会加密到用户指定的加密卷文件中，任何人想要看到虚拟磁盘里的文件，需要加载该虚拟磁盘并正确输入密码。

加载虚拟磁盘之后，可以创建受虚拟磁盘加密保护的文档，之后卸载虚拟磁盘，创建的文档则加密保存在加密卷中，这步操作起来比较简单，那么，在闪迪存储加密实际是如何实现的呢？DCrypt 另一个核心函数卸载虚拟磁盘函数 UnmountVolume，接受三个函数，分别是传入的窗口句柄、磁盘分区的序号和是否强制卸载的标识，其在闪迪存储加密中的原型是

BOOL UnmountVolume (HWND hwndDlg, int nDosDriveNo, BOOL forceUnmount)

下面我们来看下每个参数具体的含义

参数	类型	含义	示例
hwndDlg	HWND	传入的窗口句柄	NULL
driveNo	int	加载的磁盘分区序	8

		号	
forceUnmount	BOOL	是否强制卸载	FALSE

当 unmounted 为 1 的时候卸载虚拟磁盘成功，为 0 的时候卸载失败，那么失败的原因有哪些呢？

1、UNMOUNT_FAILED 卸载出错

2、UNMOUNT_LOCK_FAILED 加载的虚拟磁盘打开或者虚拟磁盘内的文件还在打开等

当为第二种情况时，闪迪存储加密的处理情形是提出提示，虚拟磁盘还打开着，提示用户是否强制卸载，如果强制卸载的话，创建的文件将不会保存，虚拟磁盘强制退出，如果不强制卸载的话，重复执行卸载虚拟磁盘的操作。

6 磁盘属性设置显示程序描述

6.1 功能

这个部分就是设置，显示磁盘属性的程序。

6.2 性能

反复使用 1000 次无问题出现。

6.3 输入项目

磁盘的属性如下：

1. 磁盘的大小。

2. 磁盘的盘符。

3. 磁盘可写，只读属性。

4. 磁盘的移动，飞移动介质属性。

6.4 输出项目

可以修改的磁盘属性：

1. 磁盘可写，只读属性。

2. 磁盘的移动，飞移动介质属性。

6.5 算法

本程序是存储卡加密文件挂载后的磁盘属性显示，修改。

If（NO 点击 MountProperties 按钮）

无法显示磁盘属性；

else

显示挂载磁盘的属性

If（NO 点击 Mount Tools 按钮）

无法显示磁盘工具的属性；

else

显示可以修改的磁盘工具的属性

If（选择修改密码选项）

显示修改面膜修改选项；

If（选择正确的旧密码 或者输入正确的密匙文件）

If（输入新密码）

显示修改面膜修改选项；

If（选择正确的密匙文件）

修改密码成功；

else

修改密码不能成功；

Else

密码修改不成功

else

输入的密码不能显示出来；

else

输入的密码不能显示出来；

If（选择将密匙文件到加密磁盘）

If（选择正确的密匙文件）

将密匙文件放入到加密文件中；

else

不能将密匙文件放入到加密文件中；

else

输入的密码不能显示出来；

If（备份加密磁盘文件头）

If（输入正确的密码）

If（磁盘是否含有隐藏磁盘属性）

If（输入正确密码）

备份加密磁盘文件头；

else

不能备份加密磁盘文件头；

else

不能备份加密磁盘文件头；

else

不能备份加密磁盘文件头；

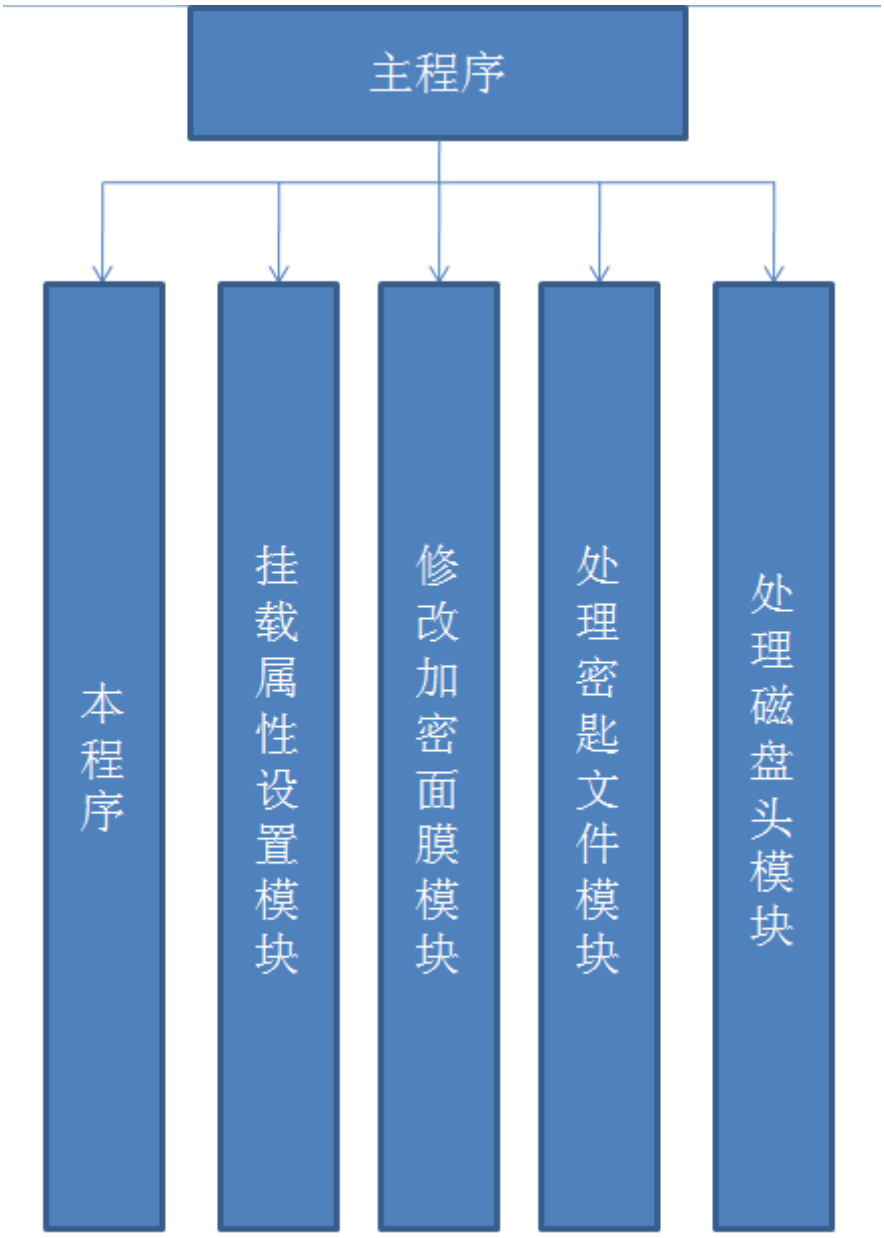
else

不能备份加密磁盘文件头；

6.6 程序逻辑

程序逻辑比较简单，只是显示和设置磁盘的属性。

6.7 接口



6.8 存储分配

存储密匙文件名到注册表。

密码修改信息保存到注册表

6.9 限制条件

牢记加密密码，

保存好密匙文件。

6.10 测试要点

1）测试是否可以将密匙文件拷贝到加密文件，或者密匙文件剪切出来。

2）测试是否备份，加载磁盘头。

3）测试修改密码是否正确。

6.11 关键函数代码介绍

在实际的应用中，闪迪存储加密可以很好地与业务系统相结合，比如，为不同的用户创建各自的加密卷，用户登录业务系统即加载各自的虚拟磁盘，那么面对业务系统中平常的修改密码操作，闪迪存储加密也能够应付自如。在闪迪存储加密中，提供了修改加密卷密码的接口，该函数在闪迪存储加密中的原型是

```
int
ChangePwd (char *lpszVolume,
Password *oldPassword,
Password *newPassword,
int pkcs5, HWND hwndDlg
);
```

下面我们来看下每个参数具体的含义

参数	类型	含义	示例
lpszVolume	char *	加密卷文件	NULL
oldPassword	Password	原密码	
newPassword	Password	新密码	
pkcs5	int	哈希算法	0
hwndDlg	HWND	传入的窗口句柄	NULL

ChangePwd 函数返回结果为

0　加载成功　　密码修改成功，或者密匙文件更改成功。

1　加载失败

7 加密文件属性显示修改程序描述

7.1 功能

此模块是加密文件属性的显示，修改程序，显示性能等等属性。修改程序文件支持语言属性。

7.2 性能

反复使用 1000 次无问题出现。

7.3 输入项目

加密文件的显示属性如下：

1. 显示支持语言

2. 显示所支持的 HotKey。

3. 加密磁盘保存到收藏夹中。

7.4 输出项目

加密文件的修改属性如下：

1. 显示支持语言

2. 显示所支持的 HotKey。

3. 加密磁盘是否保存到收藏夹中。

7.5 算法

```
If（点击 language 按钮）
    If（选择支持的语言的种类）
        If（点击确定按钮）
            变换支持语言；
        else
            不能变换支持语言；
```

else

不能变换支持语言；

else

不能变换支持语言；

If（点击 Hotkey 按钮）

If（Hotkey 设置属性设置）

If（Hotkey 快捷键选择）

变换快捷键成功；

else

不能变换快捷键成功；

else

不能变换快捷键成功；

else

不能变换快捷键成功；

If（加密磁盘设置到收藏文件夹）

If（选择合适的收藏文件夹属性）

加密磁盘设置到收藏文件夹；

else

不能变换快捷键成功；

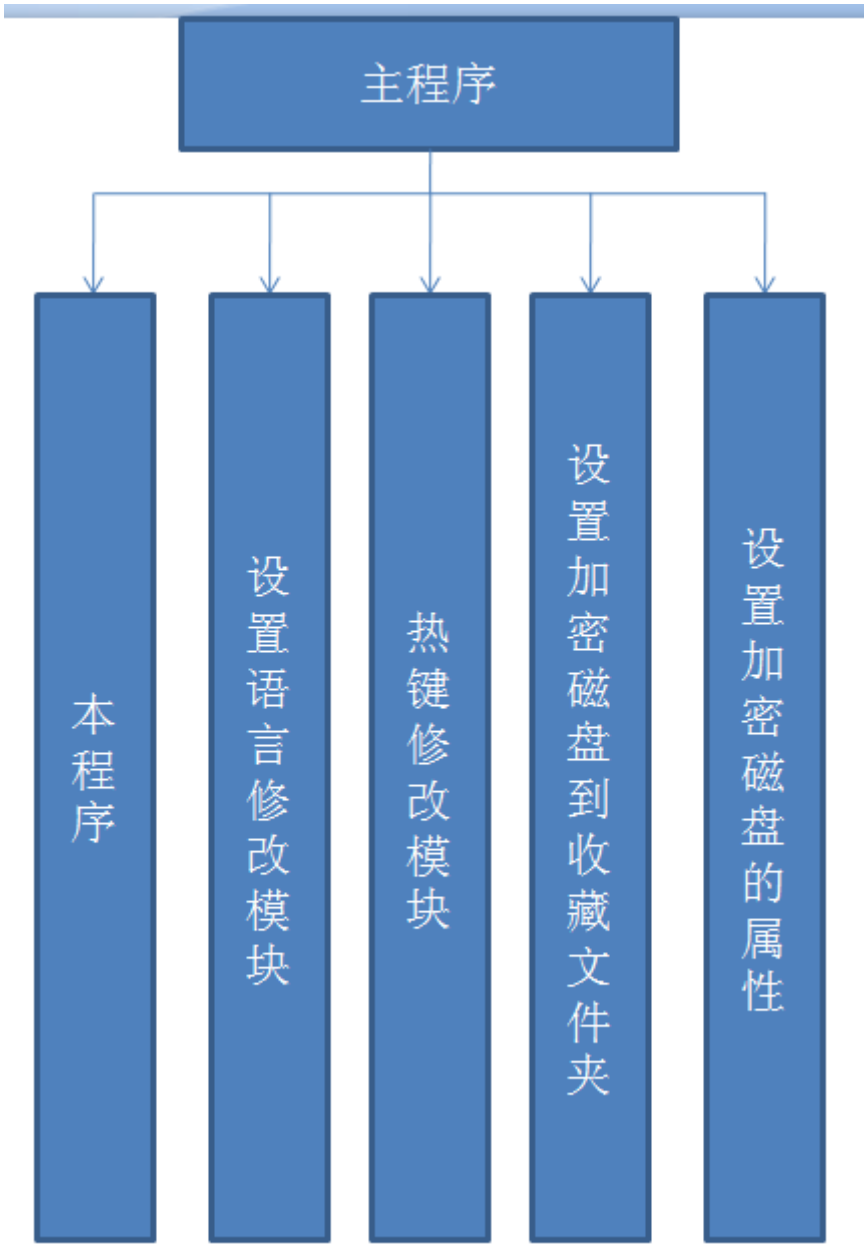
else

不能变换快捷键成功；

7.6 程序逻辑

程序逻辑比较简单，就是一些信息的显示和简单修改。

7.7 接口



7.8 限制条件

输入正确密码或者输入正确的密匙文件。

7.9 测试要点

1）测试设置支持语言修改是否正确

2）测试热键修改是否正确

3）测试添加收藏文件夹是否正确。

7.10 关键函数代码介绍

BOOL CALLBACK LanguageDlgProc (HWND hwndDlg, UINT msg, WPARAM wParam, LPARAM lParam)

LanguageDlgProc 函数是实现语言转换的实现函数。

下面我们来看下每个参数具体的含义

参数	类型	含义	示例
hwndDlg	HWND	传入的窗口句柄	NULL
msg	UINT	对话框句柄 ID	WM_INITDIALOG
wParam	WPARAM	对话框的高位参数	
lParam	LPARAM	对话框的低位参数	

LanguageDlgProc 函数返回结果为

0 加载成功

其他数值 加载失败

8 驱动程序描述

8.1 功能

此模块是驱动代码的实现，将文件加密为，虚拟为一个磁盘的代码实现。

8.2 性能

对于大文件虚拟有很高的性能要求

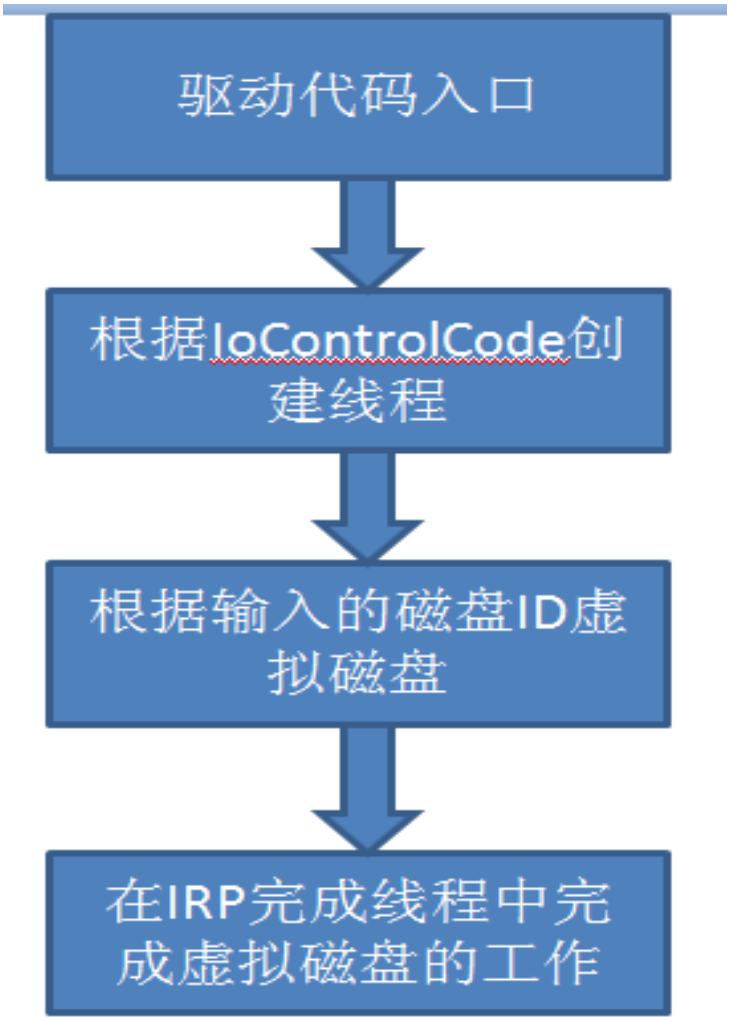
8.3 输入项目

输入的文件名字

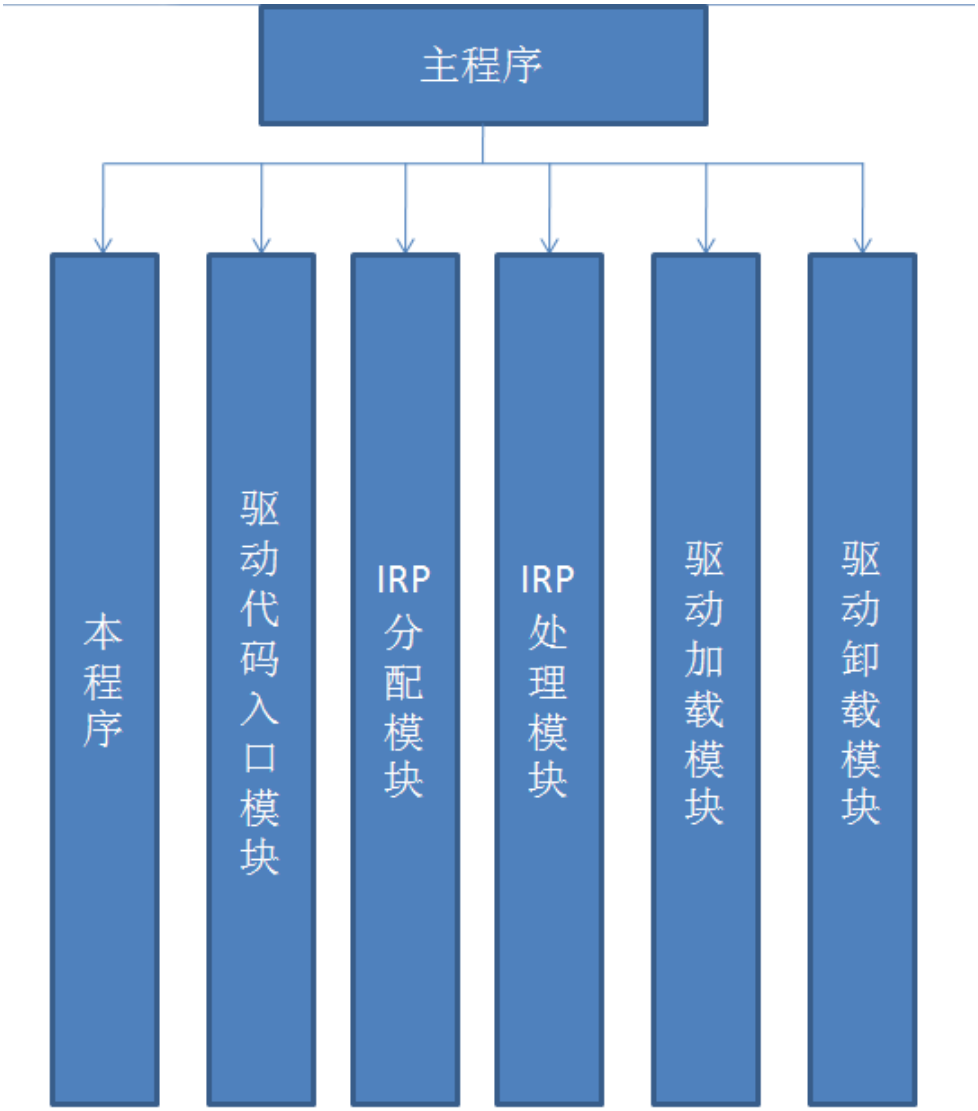
8.4 输出项目

得到虚拟后的磁盘

8.5 程序逻辑



8.6 接口



8.7 存储分配

将驱动的名称保存到注册表。

8.8 限制条件

输入要虚拟的磁盘 ID。

8.9 测试要点

1. 测试不同的磁盘 ID 是否正确。
2. 反复加载，卸载驱动是否正确。

8.10 关键函数代码分析

NTSTATUS DriverEntry (PDRIVER_OBJECT DriverObject, PUNICODE_STRING RegistryPath)

DriverEntry 函数是 windows 驱动开发标准的入口函数。

他的参数是系统分配，调用的，不需要关心。

主要内容是：

1.读取注册表中驱动有用的信息。

2.启动加密线程，以便后面调用。

3.按照 windows 驱动的原则，创建各种 IRP 的执行函数。