(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 101729550 A
(43)申请公布日 2010.06.09

(21)申请号 200910218880.9

(22)申请日 2009.11.09

(71)申请人 西北大学
地址 710127 陕西省西安市长安区学府大道1号

(72)发明人 房鼎益 张汉宁 高丽 汤战勇 陈晓江 杭继春 高沛 苏琳 章哲 安娜 李磊 赵玉洁 杨朕 何路 陈峰 王妮 胡伟 杨红

(74)专利代理机构 西安恒泰知识产权代理事务所 61216
代理人 李郑建

(51)Int.Cl.
H04L 29/06(2006.01)
H04L 9/08(2006.01)
G06F 21/00(2006.01)

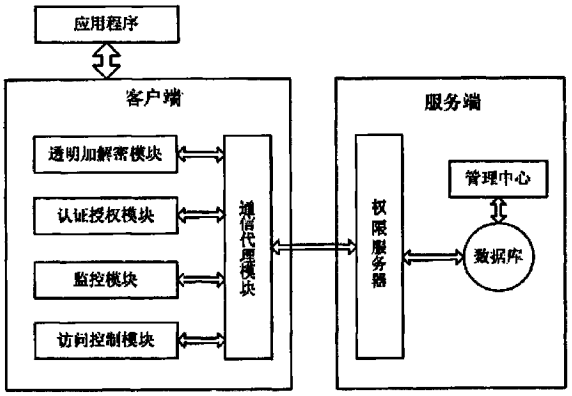
权利要求书 4 页 说明书 10 页 附图 6 页

(54)发明名称

基于透明加解密的数字内容安全防护系统及加解密方法

(57)摘要

本发明属于信息安全领域，提供了一种基于透明加解密的数字内容安全防护系统，该系统包括客户端的透明加解密模块、访问控制模块、监控模块、认证授权模块和通信代理模块，服务端的管理中心和权限服务器模块，客户端和服务端由通信代理模块和权限服务器模块连接；针对该安全防护系统提供了对数字内容加密、访问控制以及对密文进行打开、读、写操作的动态加解密方法，通过在操作系统底层实现过滤驱动实现对数字内容的透明加解密，并对用户的所有操作记录完整的日志，不但提高了系统安全性，而且加解密速度有了很大的提升，与现有同类产品相比，具有加密方式安全高效、细粒度权限控制、日志审计功能完善、管理方式便捷高效的优点。



CN 101729550 A

1. 一种基于透明加解密的数字内容安全防护系统，由客户端和服务端组成，其特征在于：

所述的客户端包括：

透明加解密模块，与通信代理模块交互，用于接收应用程序通过通信代理模块发来的数字内容加密请求，并根据请求对数字内容加密；在打开、读、写操作过程中，通过通信代理模块从服务端动态获取所需的密钥、权限信息，并根据这些信息对被访问的数字内容进行动态加解密；

认证授权模块，与通信代理模块交互，向服务端权限服务器发送身份认证信息请求，根据权限服务器返回身份信息对登陆用户进行身份认证，同时从服务端权限服务器获得权限信息，根据身份信息和权限信息对用户进行控制；用户能够通过认证授权模块为其他用户进行密文授权分发；

监控模块，与通信代理模块交互，记录用户对系统的使用、对数字内容的操作；通过通信代理模块将记录的操作日志传入服务端的权限服务器并保存在数据库中，以便对数字内容的使用进行审计与追踪；

访问控制模块，与通信代理模块交互，用于在用户对数字内容进行访问过程中，截获应用程序对数字内容的打开操作，通过透明加解密模块构造的数据结构获取数字内容的全路径；根据数字内容的全路径从服务端的权限服务器获得数字内容的内容 ID 及相应权限信息，根据权限信息控制用户对密文的使用；

通信代理模块，用以客户端其他各模块与服务端各模块之间的通信连接，发送各种请求或接收请求返回信息，传递客户端与服务端所需数据，屏蔽服务器的异构，支持离线方式使用该系统；

所述的服务端包括：

管理中心，为系统管理员提供对系统用户管理的统一的接口界面，包括添加新用户、添加用户分组，用户注册时对用户身份进行验证，查看用户对数字内容的操作日志；

权限服务器，通过通信代理同客户端各模块交换信息，接收客户端各模块发出的身份认证请求、权限信息请求或密钥信息请求，根据相应请求从数据库中获得数据，返回给客户端各模块的所需信息；

数据库，用以保存客户的身份信息，数字内容的权限信息、密钥信息、用户操作日志；

服务端的管理中心和权限服务器分别与数据库连接，服务端和客户端通过通信代理模块和权限服务器连接。

2. 权利要求 1 所述的基于透明加解密的数字内容安全防护系统对数字内容的加密保护方法，其特征在于，该方法包括以下步骤：

步骤 201：用户通过应用程序选择需要加密保护的数字内容，包括选择一个文件，一次性选择多个文件或者选择整个文件夹；

步骤 202：应用程序向通信代理模块发送加密请求；

步骤 203：通信代理模块收到加密请求后，转发给透明加解密模块；

步骤 204：透明加解密模块收到请求后，将请求保存在自身维护的请求链表中；

步骤 205：当应用程序关闭时，透明加解密模块对用户选择的数字内容加密，并在数字内容的尾部添加加密标识，用来区分明文和密文，同时将加密密钥通过通信代理模块传送

给权限服务器存储；

步骤 206：加密结束后，透明加解密模块把密文写入磁盘保存。

3. 如权利要求 2 所述的方法，其特征在于，所述的加密标识组成部分如下：

301. 标志位，标志该内容是否是受保护内容，占用 128 个字节；

302. 内容 ID，唯一标识一个数字内容，由当前时间（精确到秒）、MAC 地址和 16 位随机字符序列三部分组成，占用 256 个字节；

303. 内容类型，用来存储数字内容的原始类型信息，如定义 Office 文档中的 Word 文档为 MS001，Excel 文档类型为 MS002 等，占用 256 个字节；

304. 加密算法，用来存储该数字内容采用的加密算法类型，以便在后续的加解密操作时采用相同的算法，占用 256 个字节；

305. 预留字节，为后续的扩展提供预留空间，占用 128 个字节。

4. 权利要求 1 所述的基于透明加解密的数字内容安全防护系统的密文授权分发的方法，其特征在于，包括以下步骤：

步骤 401：用户通过应用程序选择受保护内容；

步骤 402：用户通过应用程序选择需授权的用户和权限信息，向认证授权模块发送授权请求；

步骤 403：认证授权模块接收授权请求，通过通信代理模块向服务端权限服务器发出更新权限请求，包括将原权限取交集或并集；权限服务器更新用户的权限信息并返回结果；

步骤 404：认证授权模块收到请求返回信息，将受保护数字内容通过 U 盘、email、网络共享等方式分发给授权用户，用户收到数字内容后根据授予的权限进行使用；

5. 权利要求 1 所述的基于透明加解密的数字内容安全防护系统的动态加解密方法，其特征在于，所述的动态加解密方法在数字内容打开、读、写操作中进行，其中：

所述的数字内容打开过程包括以下步骤：

步骤 501：用户通过应用程序选择需要打开的受保护数字内容；

步骤 502：应用程序向透明加解密模块发送 IRP_MJ_CREATE 请求；

步骤 503：透明加解密模块截获 IRP_MJ_CREATE 请求后，构造 IRP 查询该数字内容的尾部是否有加密标志，如有，表明此数字内容是密文，则构造数据结构记录该文件相关信息，以便在对所有打开数字内容的后续操作中区分明文和密文，然后清空系统缓存，跳到步骤 504；如果没有加密标识，则表明不是密文，跳到步骤 506；该透明加解密模块构造的数据结构包括以下部分：

1)ListEntry，为 Windows 内核链表结构；

2)FsContext，实际为数字内容控制块 FCB 的指针，唯一标志该数字内容；

3)Pid，为访问该数字内容的进程 ID；

4)FilePath，存储数字内容全路径；

步骤 504：透明加解密模块从密文的加密标识中获取内容 ID，根据该内容 ID，通过通信代理从权限服务器获取用户对该密文的权限信息和密钥信息，根据用户的权限信息判断用户是否有权限打开该内容，若有，则用相应的密钥解密该内容，然后执行步骤 505；否则，不予解密，应用程序提示用户无权打开；

步骤 505 :访问控制模块通过通信代理从权限服务器获得数字内容权限信息,根据权限信息执行细粒度的权限控制,包括菜单、按钮的可用性,剪贴板的复制和粘贴、程序之间的拖拽、OLE 数据交换、截屏的控制 ;

步骤 506 :把数字内容显示给用户 ;

所述的对密文进行读操作包括以下步骤 :

步骤 601 :应用程序向底层过滤驱动程序发送 IRP_MJ_READ 请求 ;

步骤 602 :透明加解密模块收到 IRP_MJ_READ 请求后,判断 Irp- > Flags 是否为 IRP_NOCACH 或 IRP_PAGING_IO,是则执行步骤 603,否则,透明加解密模块不做处理,而是调用操作系统的默认处理函数 PassThroughLowerDriver ;

步骤 603 :保存 Read Irp 所带 Buffer 指针,申请与 Buffer 同样大小的 SwapBuffer ;

步骤 604 :将原 Buffer 替换为 SwapBuffer,设置完成例程 ReadProcCompletion,然后等待过滤驱动程序处理的返回结果 ;

步骤 605 :完成例程被激活,透明加解密模块将 SwapBuffer 中的数据用密钥进行解密,并将解密后数据拷贝到原 Buffer 中 ;

步骤 606 :还原 Irp Buffer 指针 Irp- > MdlAddress 和 Irp- > UserBuffer ;

步骤 607 :把解密后的数字内容显示给用户 ;

所述的对密文进行写操作包括以下步骤 :

步骤 701 :应用程序发送 IRP_MJ_WRITE 请求 ;

步骤 702 :透明加解密模块截获 IRP_MJ_WRITE 请求,判断 Irp- > Flags 是否为 IRP_NOCACHE 或 IRP_PAGING_IO,是则执行步骤 703,否则 PassThroughLowerDriver(Irp),透明加解密模块不做处理,直接返回 ;

步骤 703 :保存 Write Irp 所带 Buffer 指针,申请同样大小的 SwapBuffer ;

步骤 704 :将 Buffer 中数据进行加密并将加密后的数据拷贝到 SwapBuffer 中 ;

步骤 705 :将原 Buffer 替换为 SwapBuffer,设置完成例程 (WriteProcCompletion),等待底层过滤驱动程序处理的返回结果 ;

步骤 706 :完成例程被激活,还原 Irp Buffer 指针 Irp- > MdlAddress 和 Irp- > UserBuffer ;

步骤 707 :系统将加密后的数字内容保存到计算机磁盘上。

6. 如权利要求 5 所述的方法,其特征在于,当打开的多个数字内容中有密文时,步骤 503 还包括以下步骤 :透明加解密模块在密文打开时对其创建一个新的文件节点,数据结构中的内核链表结构 (ListEntry) 将所有打开的密文的文件节点串联为链表,以区分打开的数字内容中的明文和密文,当密文关闭时,其节点被删除。

7. 如权利要求 5 所述的方法,其特征在于,在步骤 505 中,访问控制模块通过通信代理从权限服务器获得相应的权限信息,并根据权限信息执行细粒度的权限控制的过程包括以下步骤 :

步骤 801 :用户通过应用程序打开受保护的数字内容,应用程序发送内容的打开操作请求 ;

步骤 802 :访问控制模块截获应用程序的打开操作请求,通过透明加解密模块构造的数据结构获取数字内容的全路径 ;

步骤 803：访问控制模块根据数字内容的全路径，通过通信代理向权限服务器发送请求，权限服务器返回数字内容的内容 ID 及相应的权限信息；

步骤 804：访问控制模块根据获得的权限信息执行细粒度的权限控制，包括菜单、按钮的可用性，剪贴板的复制和粘贴、程序之间的拖拽、OLE 数据交换、截屏等方式的控制。

基于透明加解密的数字内容安全防护系统及加解密方法

技术领域

[0001] 本发明属于信息安全领域，具体涉及一种基于透明加解密的数字内容安全防护系统及加解密方法。

背景技术

[0002] 随着计算机的普遍应用和 Internet 的飞速发展，越来越多技术发明、创新等依赖计算机技术，因此，很多核心的机密文档以电子化形式存储在计算机上，甚至绝大多数的企业核心技术文档本身就是设计图纸、程序源代码等的电子文档。因此，技术进步给信息安全带来了新的挑战，网络技术的普及和移动办公设备、移动存储设备、笔记本电脑的广泛使用等，在给人们带来高效和方便的同时又增加了信息被侦听、截获及非法拷贝的危险。据调研机构调查结果显示，每年都会发生大量的企业敏感数据丢失事件，电子文件泄露对企业所造成的损失是极其惨重的。而当这种情况涉及国家机密方面，所造成的损失更是不可估量的。为了防止机密泄漏，企业采取了各种各样的文件加密措施，同时也出现了很多对文件进行加密的技术出现。

[0003] 加解密技术分为静态加解密和动态加解密，静态加解密是指在加密期间，待加密的数据处于未使用状态，这些数据一旦加密，用户在使用前需首先通过静态解密得到明文，然后才能使用；动态加密即透明加解密技术，是指数据在使用过程中，系统自动对数据进行加解密操作，不改变用户对文件的访问（打开、读、写等）习惯，无需用户的干预，表面看来，访问加密的文件和访问未加密的文件基本相同，因此对合法用户来说这些加密文件是“透明的”，即好像没有加密一样，但对于没有访问权限的用户，即使通过其它非常规手段得到了加密文件也无法使用。由于透明加解密技术不改变用户的使用习惯，而且无需用户太多干预操作即可实现文件的安全，因而近年来得到了广泛的应用。

[0004] 目前市场上已经有很多透明加解密的安全产品实现了对数字内容的保护，但存在着各种各样的不足和缺陷：

[0005] 1、安全性低。大数产品采用在操作系统用户态完成加解密操作，这种方式安全性低，会造成数字内容在使用过程中“明文落地”，即明文内容存储在磁盘上的情况，易造成机密信息的失密和泄露；

[0006] 2、速度低。由于在操作系统用户态完成的加解密操作，其速度比较低，导致处理文件效率不够高；例如上海索远 Docsecurity 系统，未采用过滤驱动程序且改变文档格式，使得加密文件必须用限定的应用程序操作，速度较低，并影响了用户的使用习惯；

[0007] 3、权限控制细化不够。虽然大多安全产品能够允许或拒绝用户访问受保护的数字内容，但对不能提供更加细分化的权限控制，这种静态的提供“全部或零”权限的安全产品不能满足当今动态的业务需求。例如铁卷电子文档安全系统，虽然引入了过滤驱动技术，但不支持细粒度的权限控制，无法满足用户的动态需求；

[0008] 4、监控机制欠缺。大多同类产品设计较简单，没有实现对数字内容使用行为进行完善的跟踪记录。

发明内容

[0009] 为了克服上述现有加解密技术同类产品的不足和缺陷，本发明的目的在于，提供一种基于透明加解密的数字内容安全防护系统以及加解密方法，本发明通过在操作系统底层实现过滤驱动，从而实现对数字内容的透明加解密，本发明结合透明加解密技术、访问控制技术和数字权限管理技术，不但提高了系统的安全性，而且加解密速度有了很大的提升。

[0010] 为了实现上述任务，本发明采用的技术方案如下：

[0011] 一种基于透明加解密的数字内容安全防护系统，由客户端和服务端组成，客户端包括：

[0012] 透明加解密模块，与通信代理模块交互，用于接收应用程序通过通信代理模块发来的数字内容加密请求，并根据请求对数字内容加密；在打开、读、写操作过程中，通过通信代理模块从服务端动态获取所需的密钥、权限信息，并根据这些信息对被访问的数字内容进行动态加解密；

[0013] 认证授权模块，与通信代理模块交互，向服务端权限服务器发送身份认证信息请求，根据权限服务器返回身份信息对登陆用户进行身份认证，同时从服务端权限服务器获得权限信息，根据身份信息和权限信息对用户进行控制；用户能够通过认证授权模块为其他用户进行密文授权分发；

[0014] 监控模块，与通信代理模块交互，记录用户对系统的使用、对数字内容的操作；通过通信代理模块将记录的操作日志传入服务端的权限服务器并保存在数据库中，以便对数字内容的使用进行审计与追踪；

[0015] 访问控制模块，与通信代理模块交互，用于在用户对数字内容进行访问过程中，截获应用程序对数字内容的打开操作，通过透明加解密模块构造的数据结构获取数字内容的全路径；根据数字内容的全路径从服务端的权限服务器获得数字内容的内容 ID 及相应权限信息，根据权限信息控制用户对密文的使用；

[0016] 通信代理模块，用以客户端其他各模块与服务端各模块之间的通信连接，发送各种请求或接收请求返回信息，传递客户端与服务端所需数据，屏蔽服务器的异构；

[0017] 服务端包括：

[0018] 管理中心，为系统管理员提供对系统用户管理的统一的接口界面，包括添加新用户、添加用户分组，用户注册时对用户身份进行验证，查看用户对数字内容的操作日志；

[0019] 权限服务器，通过通信代理同客户端各模块交换信息，接收客户端各模块发出的身份认证请求、权限信息请求或密钥信息请求，根据相应请求从数据库中获得数据，返回给客户端各模块的所需信息；

[0020] 数据库，用以保存客户的身份信息，数字内容的权限信息、密钥信息、用户操作日志；

[0021] 服务端的管理中心和权限服务器分别与数据库连接，服务端和客户端通过通信代理模块和权限服务器连接。

[0022] 基于透明加解密的数字内容安全防护系统对数字内容的加密保护方法，包括以下步骤：

[0023] 步骤 201：用户通过应用程序选择需要加密保护的数字内容，包括选择一个文件，

一次性选择多个文件或者选择整个文件夹；

[0024] 步骤 202：应用程序向通信代理模块发送加密请求；

[0025] 步骤 203：通信代理模块收到加密请求后，转发给透明加解密模块；

[0026] 步骤 204：透明加解密模块收到请求后，将请求保存在自身维护的请求链表中；

[0027] 步骤 205：当应用程序关闭时，透明加解密模块对用户选择的数字内容加密，并在数字内容的尾部添加加密标识，用来区分明文和密文，同时将加密密钥通过通信代理模块传送给权限服务器存储；

[0028] 步骤 206：加密结束后，透明加解密模块把密文写入磁盘保存。

[0029] 上述加密标识组成部分如下：

[0030] 301：标志位，标志该内容是否是受保护内容，占用 128 个字节；

[0031] 302：内容 ID，唯一标识一个数字内容，由当前时间（精确到秒）、MAC 地址和 16 位随机字符序列三部分组成，占用 256 个字节。

[0032] 303：内容类型，用来存储数字内容的原始类型信息，如定义 Office 文档中的 Word 文档为 MS001，Excel 文档类型为 MS002 等，占用 256 个字节。

[0033] 304：加密算法，用来存储该数字内容采用的加密算法类型，以便在后续的加解密操作时采用相同的算法，占用 256 个字节。

[0034] 305：预留字节，为后续的扩展提供预留空间，占用 128 个字节。

[0035] 基于透明加解密的数字内容安全防护系统的密文授权分发的方法包括以下步骤：

[0036] 步骤 401：用户通过应用程序选择受保护内容；

[0037] 步骤 402：用户通过应用程序选择需授权的用户和权限信息，向认证授权模块发送授权请求；

[0038] 步骤 403：认证授权模块接收授权请求，通过通信代理模块向权限服务器发出更新权限请求，包括将原权限取交集或并集；权限服务器更新用户的权限信息并返回结果；

[0039] 步骤 404：认证授权模块收到请求返回信息，将受保护数字内容通过 U 盘、email、网络共享等方式分发给授权用户，用户收到数字内容后根据授予的权限进行使用；

[0040] 基于透明加解密的数字内容安全防护系统的动态加解密方法，动态加解密在数字内容打开、读、写操作中进行，其中：

[0041] 数字内容打开过程包括以下步骤：

[0042] 步骤 501：用户通过应用程序选择需要打开的受保护数字内容；

[0043] 步骤 502：应用程序向透明加解密模块发送 IRP_MJ_CREATE 请求；

[0044] 步骤 503：透明加解密模块截获 IRP_MJ_CREATE 请求后，构造 IRP 查询该数字内容的尾部是否有加密标志，如有，表明此数字内容是密文，则构造数据结构记录该文件相关信息，以便在对所有打开数字内容的后续操作中区分明文和密文，然后清空系统缓存，跳到步骤 504；如果没有加密标识，则表明不是密文，跳到步骤 506；该透明加解密模块构造的数据结构包括以下部分：

[0045] 1)ListEntry，为 Windows 内核链表结构；

[0046] 2)FsContext，实际为数字内容控制块 FCB 的指针，唯一标志该数字内容；

[0047] 3)Pid，为访问该数字内容的进程 ID；

[0048] 4)FilePath,存储数字内容全路径；

[0049] 步骤 504：透明加解密模块从密文的加密标识中获取内容 ID,根据该内容 ID,通过通信代理从权限服务器获取用户对该密文的权限信息和密钥信息,根据用户的权限信息判断用户是否有权限打开该内容,若有,则用相应的密钥解密该内容,然后执行步骤 505；否则,不予解密,应用程序提示用户无权打开；

[0050] 步骤 505：访问控制模块通过通信代理从权限服务器获得数字内容权限信息,根据权限信息执行细粒度的权限控制,包括菜单、按钮的可用性,剪贴板的复制和粘贴、程序之间的拖拽、OLE 数据交换、截屏的控制；

[0051] 步骤 506：把数字内容显示给用户；

[0052] 对密文进行读操作包括以下步骤：

[0053] 步骤 601：应用程序向底层过滤驱动程序发送 IRP_MJ_READ 请求；

[0054] 步骤 602：透明加解密模块收到 IRP_MJ_READ 请求后,判断 Irp- ＞ Flags 是否为 IRP_NOCACH 或 IRP_PAGING_IO,是则执行步骤 603,否则,透明加解密模块不做处理,而是调用操作系统的默认处理函数 PassThroughLowerDriver；

[0055] 步骤 603：保存 Read Irp 所带 Buffer 指针,申请与 Buffer 同样大小的 SwapBuffer；

[0056] 步骤 604：将原 Buffer 替换为 SwapBuffer,设置完成例程 ReadProcCompletion,然后等待过滤驱动程序处理的返回结果；

[0057] 步骤 605：完成例程被激活,透明加解密模块将 SwapBuffer 中的数据用密钥进行解密,并将解密后数据拷贝到原 Buffer 中；

[0058] 步骤 606：还原 Irp Buffer 指针 Irp- ＞ MdlAddress 和 Irp- ＞ UserBuffer；

[0059] 步骤 607：把解密后的数字内容显示给用户；

[0060] 对密文进行写操作包括以下步骤：

[0061] 步骤 701：应用程序发送 IRP_MJ_WRITE 请求；

[0062] 步骤 702：透明加解密模块截获 IRP_MJ_WRITE 请求,判断 Irp- ＞ Flags 是否为 IRP_NOCACHE 或 IRP_PAGING_IO,是则执行步骤 703,否则 PassThroughLowerDriver(Irp),透明加解密模块不做处理,直接返回；

[0063] 步骤 703：保存 Write Irp 所带 Buffer 指针,申请同样大小的 SwapBuffer；

[0064] 步骤 704：将 Buffer 中数据进行加密并将加密后的数据拷贝到 SwapBuffer 中；

[0065] 步骤 705：将原 Buffer 替换为 SwapBuffer,设置完成例程 (WriteProcCompletion),等待底层过滤驱动程序处理的返回结果；

[0066] 步骤 706：完成例程被激活,还原 Irp Buffer 指针 Irp- ＞ MdlAddress 和 Irp- ＞ UserBuffer；

[0067] 步骤 707：系统将加密后的数字内容保存到计算机磁盘上。

[0068] 当打开的多个数字内容中有密文时,步骤 503 还包括以下步骤：透明加解密模块在密文打开时对其创建一个新的文件节点,数据结构中的内核链表结构 (ListEntry) 将所有打开的密文的文件节点串联为链表,以区分打开的数字内容中的明文和密文,当密文关闭时,其节点被删除。

[0069] 在步骤 505 中,访问控制模块通过通信代理从权限服务器获得相应的权限信息,

并根据权限信息执行细粒度的权限控制的过程包括以下步骤：

[0070] 步骤 801：用户通过应用程序打开受保护的数字内容，应用程序发送内容的打开操作请求；

[0071] 步骤 802：访问控制模块截获应用程序的打开操作请求，通过透明加解密模块构造的数据结构获取数字内容的全路径。

[0072] 步骤 803：访问控制模块根据数字内容的全路径，通过通信代理向权限服务器发送请求，权限服务器返回数字内容的内容 ID 及相应的权限信息。

[0073] 步骤 804：访问控制模块根据获得的权限信息执行细粒度的权限控制，包括菜单、按钮的可用性，剪贴板的复制和粘贴、程序之间的拖拽、OLE 数据交换、截屏等方式的控制。

[0074] 与现有技术相比，本发明的有益效果如下：

[0075] 1. 加密方式安全高效。由于本发明采用基于底层过滤驱动实现的透明加解密，与传统的在应用层实现加解密方式相比，此方式提高了系统的安全性，同时加解密速度有了很大提升，经过测试：对于 35M 的文件，传统的应用层实现加解密需 2 分钟，而本发明基于底层过滤驱动实现加解密仅需 6 秒钟。

[0076] 2. 细粒度权限控制。本发明根据数字内容拥有者的不同需求，编写 COM 插件实现对重要应用软件（如 Word、Excel、AutoCad）的控制，对其它不支持插件开发的软件采用 Hook 技术，灵活设置权限，从而满足了用户不断增长的需求。个人或组均可赋权，权限具体包括完全控制、只读几次，打印几次、可复制、可另存、可编辑、失效日期、有效时间等，这对于传统的静态提供“全部或零”权限的安全产品是一个很大的突破。

[0077] 3. 日志审计功能完善。本发明对用户对受保护内容的所有操作（如打开、保存、另存、打印等）都做详细的日志记录，提供全面的日志审计功能，对涉密数字内容外泄的事后追查取证提供有力支持。

[0078] 4. 管理方式便捷高效。本发明的数字安全防护系统的管理中心采用 B/S 结构，Web 管理灵活方便，适合在使用环境内任何主机对管理中心的访问，为管理员提供统一的接口界面对系统进行配置管理；用户注册时对用户身份进行验证；查询用户详细的操作日志。

附图说明

[0079] 图 1 为本发明的数字内容安全防护系统结构图；

[0080] 图 2 为对需要保护的数字内容的加密保护过程图；

[0081] 图 3 为加密标识结构图；

[0082] 图 4 为用户将密文进行授权分发过程图；

[0083] 图 5 为数字内容的打开过程流程图；

[0084] 图 6 为对密文进行读操作的流程图；

[0085] 图 7 为对密文进行写操作的流程图；

[0086] 图 8 为访问控制模块对密文进行访问控制的过程图；

[0087] 以下结合附图对本发明作进一步详细说明。

具体实施方式

[0088] 本发明适用的操作系统有：Microsoft Windows XP，Microsoft Windows2000，

Microsoft Windows 2003,Microsoft Windows vista等;硬件环境:Pentium(R)3CPU,256M内存以上;应用软件:Microsoft Office2000/XP/2003/2007, Adobe Reader, AutoCAD等;适用的开发语言:C++,C,C#。

[0089] 参见图1,一种基于透明加解密的数字内容安全防护系统,包括客户端和服务端,其中,

[0090] 客户端包括以下各单元:

[0091] 透明加解密模块,与通信代理模块交互,用于接收应用程序通过通信代理模块发来的数字内容加密请求,并根据请求对数字内容加密;在打开、读、写操作过程中,通过通信代理模块从服务端动态获取所需的密钥、权限信息,并根据这些信息对被访问的数字内容进行动态加解密;

[0092] 认证授权模块,与通信代理模块交互,向服务端权限服务器发送身份认证信息请求,根据权限服务器返回身份信息对登陆用户进行身份认证,同时从服务端权限服务器获得权限信息,根据身份信息和权限信息对用户进行控制;用户能够通过认证授权模块为其他用户进行密文授权分发;

[0093] 监控模块,与通信代理模块交互,记录用户对系统的使用、对数字内容的操作;通过通信代理模块将记录的操作日志传入服务端的权限服务器并保存在数据库中,以便对数字内容的使用进行审计与追踪;

[0094] 访问控制模块,与通信代理模块交互,用于在用户对数字内容进行访问过程中,截获应用程序对数字内容的打开操作,通过透明加解密模块构造的数据结构获取数字内容的全路径;根据数字内容的全路径从服务端的权限服务器获得数字内容的内容ID及相应权限信息,根据权限信息控制用户对密文的使用;

[0095] 通信代理模块,用以客户端其他各模块与服务端各模块之间的通信连接,发送各种请求或接收请求返回信息,传递客户端与服务端所需数据,屏蔽服务器的异构,即服务器如果有变动,不用修改其它模块,只需修改通信代理模块。

[0096] 服务端为系统管理员提供一个方便快捷安全有效的管理控制中心,所有的客户端请求都通过服务端权限服务器得到响应,服务端包括以下各单元:

[0097] 管理中心,为系统管理员提供对系统用户管理的统一的接口界面,包括添加新用户、添加用户分组,用户注册时对用户身份进行验证,查看用户对数字内容的操作日志;

[0098] 权限服务器,通过通信代理同客户端各模块交换信息,接收客户端各模块发出的身份认证请求、权限信息请求或密钥信息请求,根据相应请求从数据库中获得数据,返回给客户端各模块的所需信息;

[0099] 数据库,用以保存客户的身份信息,数字内容的权限信息、密钥信息、用户操作日志;

[0100] 服务端的管理中心和权限服务器模块分别与数据库连接,服务端和客户端通过通信代理模块和权限服务器模块的连接交换信息。

[0101] 以上各模块之间的主要接口如下:

[0102] 透明加解密-通信代理接口:用于透明加解密模块向通信代理模块发送获取数字内容的权限和密钥等信息的请求。通过DeviceIoControl实现。

[0103] 认证授权-通信代理接口:用于认证授权模块向通信代理模块发送身份认证消

息，获取文件权限信息等。通过管道通信机制实现。

[0104] 监控－通信代理接口：用于监控模块向通信代理模块发送用户操作日志操作信息，通过 COM 接口实现通信。

[0105] 访问控制－通信代理接口：用于访问控制模块向通信代理模块发送请求，获得文件的权限、客户端认证等信息。通过 Windows 管道机制实现通信。

[0106] 通信代理－权限服务器接口：用于通信代理模块转发来自客户端其他模块的请求，如获得文件的权限信息、加密密钥、用户操作日志等信息。通过 SSL 加密信道实现通信。

[0107] 权限服务器和管理中心之间没有直接通信，各自与服务端数据库直接通信。

[0108] 客户端与应用程序连接，底层过滤驱动程序与应用程序的通信通过 DeviceIoControl 实现。

[0109] 在该系统中，数字内容的访问控制通过客户端的访问控制模块完成，通过编写 COM 插件实现对重要应用软件（如 Word、Excel、AutoCad）的控制，对不支持插件开发的软件采用 Hook 技术拦截信息，并对剪贴板的复制和粘贴、程序之间的拖拽、OLE 数据交换、截屏等方式均进行控制，实现两个目的：一是保证的应用程序与非涉密的应用程序之间的数据交换只进不出，例如，若 Word 文档被加密后，其内容就不能被粘贴到非涉密的 OutLook 中，或者粘贴出的内容是乱码；二是加密软件之间能够进行正常的数据交换，例如，过 Word 和 Excel 均为受保护进程，则数据可以从 Word 复制粘贴到 Excel 中。通过上述方式实现细粒度的访问控制，并且 COM 插件和系统 Hook 与底层过滤驱动程序有工作状态验证机制，一旦上层的访问控制与监控模块被恶意修改或破坏，透明加解密服务将自动停止。

[0110] 参见图 2，在每个数字内容被使用之前，需要根据其重要程度对其进行加密保护，不需要加密保护的数字内容是明文，需要加密保护并被透明加解密模块加密后的数字内容为密文，基于透明加解密的数字内容安全防护系统对数字内容的加密保护方法包括以下步骤：

[0111] 步骤 201：用户通过应用程序选择需要加密保护的数字内容，包括选择一个文件、一次性选择多个文件、选择整个文件夹，该选择的操作方式支持多种方式，如右键、拖拽、属性页等；

[0112] 步骤 202：应用程序向通信代理模块发送加密请求；

[0113] 步骤 203：通信代理模块收到加密请求后，转发给透明加解密模块；

[0114] 步骤 204：透明加解密模块收到请求后，将请求保存在自身维护的请求链表中；

[0115] 步骤 205：当应用程序关闭时，透明加解密模块对用户选择的数字内容加密，并在数字内容的尾部添加加密标识，用来区分明文和密文，同时将加密密钥通过通信代理模块传送给权限服务器模块存储；

[0116] 步骤 206：加密结束后，透明加解密模块把密文写入磁盘保存；

[0117] 参见图 3，透明加解密模块加密文件时，在文件尾部添加的加密标识包括以下部分：

[0118] 301：标志位，标志该内容是否是受保护内容，占用 128 个字节；

[0119] 302：内容 ID，唯一标识一个数字内容，由当前时间（精确到秒）、MAC 地址和 16 位随机字符序列三部分组成，占用 256 个字节；

[0120] 303：内容类型，用来存储数字内容的原始类型信息，如定义 Office 文档中的 Word

文档为 MS001，Excel 文档类型为 MS002 等，占用 256 个字节；

[0121] 304：加密算法，用来存储该数字内容采用的加密算法类型，以便在后续的加解密操作时采用相同的算法，占用 256 个字节；

[0122] 305：预留字节，为后续的扩展提供预留空间，占用 128 个字节。

[0123] 参见图 4，一种基于透明加解密的数字内容安全防护系统的密文授权分发的方法，包括以下步骤：

[0124] 步骤 401：用户通过应用程序选择受保护内容；

[0125] 步骤 402：用户通过应用程序选择需授权的用户并选择权限信息，向认证授权模块发送授权请求；

[0126] 步骤 403：认证授权模块接收授权请求，通过通信代理模块向服务端权限服务器发出更新权限请求，包括将原权限取交集或并集；权限服务器更新用户的权限信息并返回结果；

[0127] 步骤 404：认证授权模块收到请求返回信息，将受保护文件通过 U 盘、email、网络共享等方式分发给授权用户，用户收到文件后根据授予的权限进行使用；

[0128] Windows NT 系统对数字内容和设备的访问过程首先对应驱动层为 IRP_MJ_CREATE，最后操作对应驱动层为 IRP_MJ_CLOSE，为避免系统缓存造成的数据泄露，在 IRP_MJ_CREATE 和 IRP_MJ_CLOSE 操作中均对缓存进行清空处理，IRP_MJ_READ 和 IRP_MJ_WRITE 对应应用程序的读写请求，读写所操作的数据存于 IRP(I/O Request Packet，是 I/O 管理器根据应用程序发出的请求构造的固定数据格式)。透明加解密是在系统对数据的打开、读、写操作中完成的，在上述操作中，上层应用程序向透明加解密模块发出相应的读写请求，透明加解密模块过滤掉应用程序对缓存的读写请求，只对非缓存读写请求进行相应操作，系统通过判断读写请求中标志是否为 IRP_NOCACH 和 IRP_PAGING_IO 来判断是否为非缓存读写。

[0129] 基于透明加解密的数字内容安全防护系统的动态加解密方法，该方法在数字内容打开、读、写操作中进行，其中：

[0130] 参见图 5，数字内容打开过程包括以下步骤：

[0131] 步骤 501：用户通过应用程序选择需要打开的受保护数字内容；

[0132] 步骤 502：应用程序向透明加解密模块发送 IRP_MJ_CREATE 请求；

[0133] 步骤 503：透明加解密模块截获 IRP_MJ_CREATE 请求后，构造 IRP 查询该数字内容的尾部是否有加密标志，如有，表明此文件是密文，则构造数据结构记录该文件相关信息，以便在对所有打开数字内容的后续操作中区分明文和密文，然后清空系统缓存，跳到步骤 504；如果没有加密标识，则表明不是密文，跳到步骤 506；该透明加解密模块构造的数据结构包括以下部分：

[0134] 1)ListEntry，为 Windows 内核链表结构；

[0135] 2)FsContext，实际为数字内容控制块 FCB 的指针，唯一标志该数字内容；

[0136] 3)Pid，为访问该数字内容的进程 ID；

[0137] 4)FilePath，存储数字内容全路径；

[0138] 步骤 504：透明加解密模块从密文的加密标识中获取内容 ID，根据该内容 ID 通过通信代理从权限服务器获取用户对该密文的权限信息和密钥信息，根据用户的权限信息判

断用户是否有权限打开该内容，如果有，则用相应的密钥解密该内容，然后执行步骤 505；否则，不予解密，应用程序提示用户无权打开；

[0139] 步骤 505：访问控制模块通过通信代理从权限服务器获得数字内容权限信息，根据权限信息执行细粒度的权限控制，包括菜单、按钮的可用性，剪贴板的复制和粘贴、程序之间的拖拽、OLE 数据交换、截屏的控制；

[0140] 步骤 506：把数字内容显示给用户。

[0141] 参见图 6，对密文进行读操作包括以下步骤：

[0142] 步骤 601：应用程序向底层过滤驱动程序发送 IRP_MJ_READ 请求；

[0143] 步骤 602：透明加解密模块收到 IRP_MJ_READ 请求后，判断 Irp- ＞ Flags 是否为 IRP_NOCACH 或 IRP_PAGING_IO，是则执行步骤 603，否则，透明加解密模块不做处理，而是调用操作系统的默认处理函数 PassThroughLowerDriver；

[0144] 步骤 603：保存 Read Irp 所带 Buffer 指针，申请与 Buffer 同样大小的 SwapBuffer；

[0145] 步骤 604：将原 Buffer 替换为 SwapBuffer，设置完成例程 ReadProcCompletion，然后等待底层过滤驱动程序处理的返回结果；

[0146] 步骤 605：完成例程被激活，透明加解密模块将 SwapBuffer 中的数据用密钥进行解密，并将解密后数据拷贝到原 Buffer 中；

[0147] 步骤 606：还原 Irp Buffer 指针 Irp- ＞ MdlAddress 和 Irp- ＞ UserBuffer；

[0148] 步骤 607：把解密后的数字内容显示给用户。

[0149] 参见图 7，对密文进行写操作包括以下步骤：

[0150] 步骤 701：应用程序发送 IRP_MJ_WRITE 请求；

[0151] 步骤 702：透明加解密模块截获 IRP_MJ_WRITE 请求，判断 Irp- ＞ Flags 是否为 IRP_NOCACHE 或 IRP_PAGING_IO，是则执行步骤 703，否则 PassThroughLowerDriver(Irp)，透明加解密模块不做处理，直接返回；

[0152] 步骤 703：保存 Write Irp 所带 Buffer 指针，申请同样大小的 SwapBuffer；

[0153] 步骤 704：将 Buffer 中数据进行加密并将加密后的数据拷贝到 SwapBuffer 中；

[0154] 步骤 705：将原 Buffer 替换为 SwapBuffer，设置完成例程 (WriteProcCompletion)，等待底层过滤驱动程序处理的返回结果，如写操作是否成功，写了多少字节；

[0155] 步骤 706：完成例程被激活，还原 Irp Buffer 指针 Irp- ＞ MdlAddress 和 Irp- ＞ UserBuffer；

[0156] 步骤 707：系统将加密后的数字内容保存到计算机磁盘上。

[0157] 在上述对数字内容的读写过程中，加解密操作均在 SwapBuffer 中进行，原始 Irp 所带的数据缓冲区为明文，而对磁盘的读写均为密文，这样既保证了明文数据不落地又避免了与应用程序的数据操作可能产生的冲突。

[0158] 另外，当打开的多个数字内容中有密文时，步骤 503 还包括以下步骤：透明加解密模块在密文打开时对其创建一个新的文件节点，数据结构中的内核链表结构 (ListEntry) 将所有打开的密文的文件节点串联为链表，以区分打开的数字内容中的明文和密文，当密文关闭时，其节点被删除。

[0159] 参见图 8，步骤 505 中访问控制模块通过通信代理从权限服务器获得相应的权限信息，并根据权限信息执行细粒度的权限控制的过程包括以下步骤：

[0160] 步骤 801：用户通过应用程序打开受保护的数字内容，应用程序发送内容的打开操作请求；

[0161] 步骤 802：访问控制模块截获应用程序的打开操作请求，通过透明加解密模块构造的数据结构获取数字内容的全路径；

[0162] 步骤 803：访问控制模块根据数字内容的全路径，通过通信代理向权限服务器发送请求，权限服务器返回数字内容的内容 ID 及相应的权限信息；

[0163] 步骤 804：访问控制模块根据获得的权限信息执行细粒度的权限控制，包括菜单、按钮的可用性，剪贴板的复制和粘贴、程序之间的拖拽、OLE 数据交换、截屏等方式的控制。

[0164] 为了达到事前防御事后追踪的目的，客户端监控模块对用户进行的所有关键操作（如打开、保存、另存、打印等）记录了详细的操作日志，所有的日志操作信息都可以通过服务端的管理中心查询。

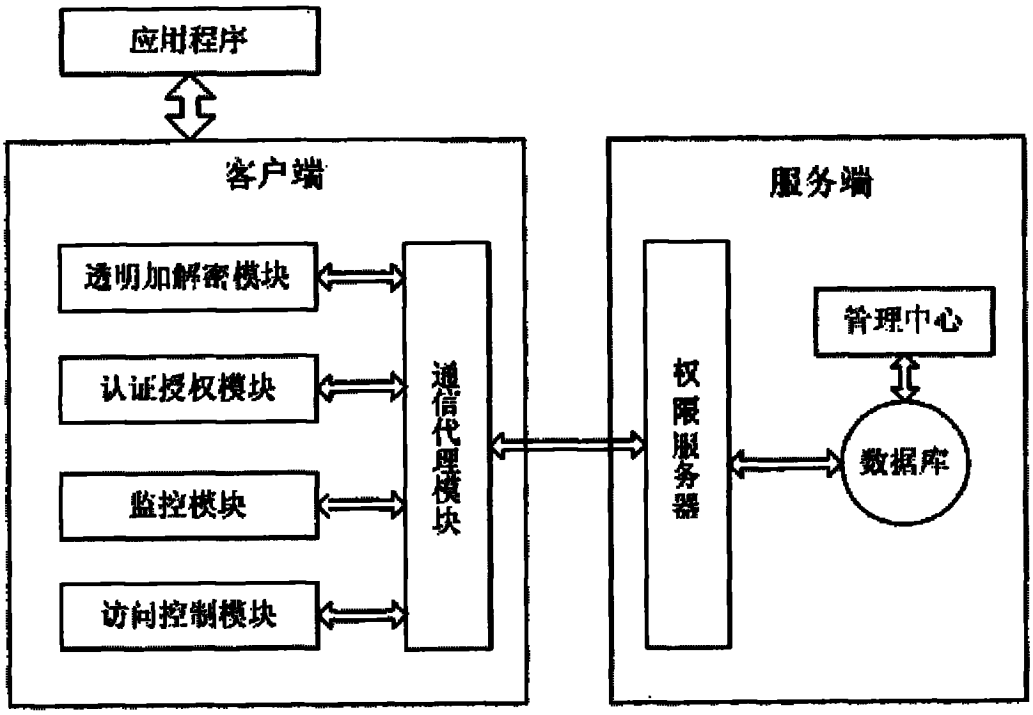

应用程序
客户端
透明加解密模块
认证授权模块
监控模块
访问控制模块
通信代理模块
服务端
权限服务器
管理中心
数据库

图 1

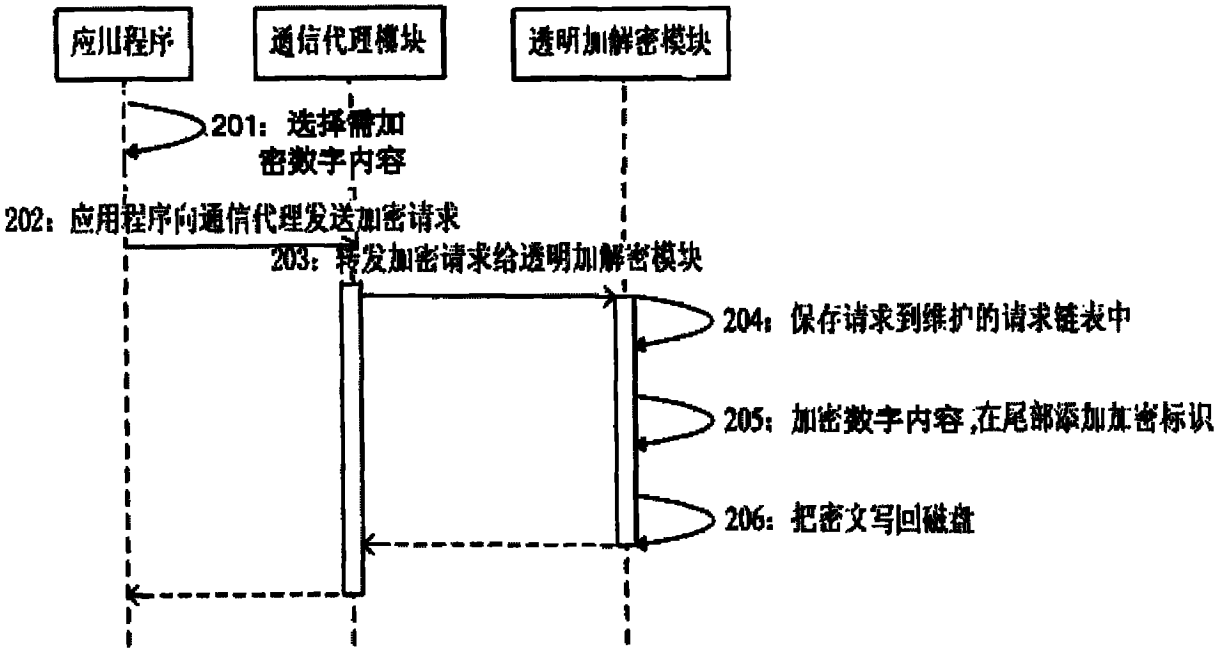

应用程序
通信代理模块
透明加解密模块
201：选择需加密数字内容
202：应用程序向通信代理发送加密请求
203：转发加密请求给透明加解密模块
204：保存请求到维护的请求链表中
205：加密数字内容，在尾部添加加密标识
206：把密文写回磁盘

图 2

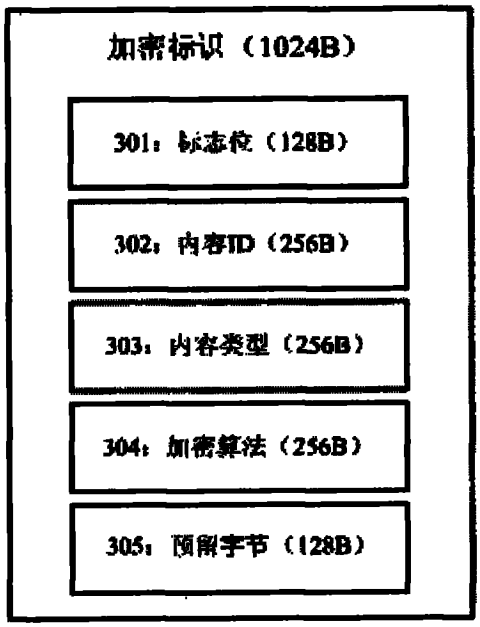

加密标识（1024B）
301：标志位（128B）
302：内容ID（256B）
303：内容类型（256B）
304：加密算法（256B）
305：预留字节（128B）

图 3

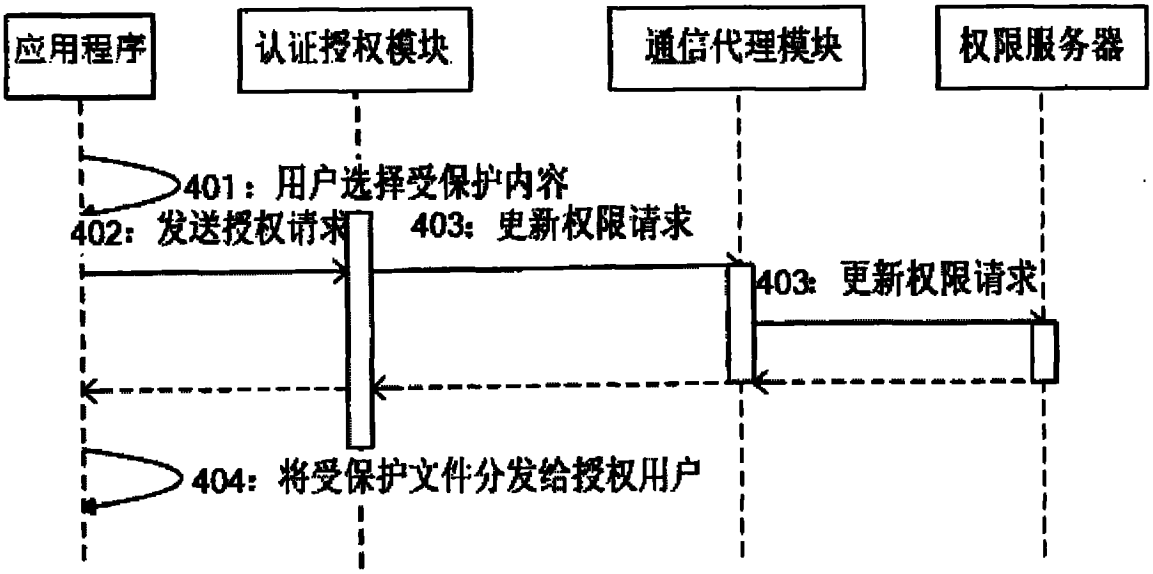

应用程序
认证授权模块
通信代理模块
权限服务器
401：用户选择受保护内容
402：发送授权请求
403：更新权限请求
403：更新权限请求
404：将受保护文件分发给授权用户

图 4

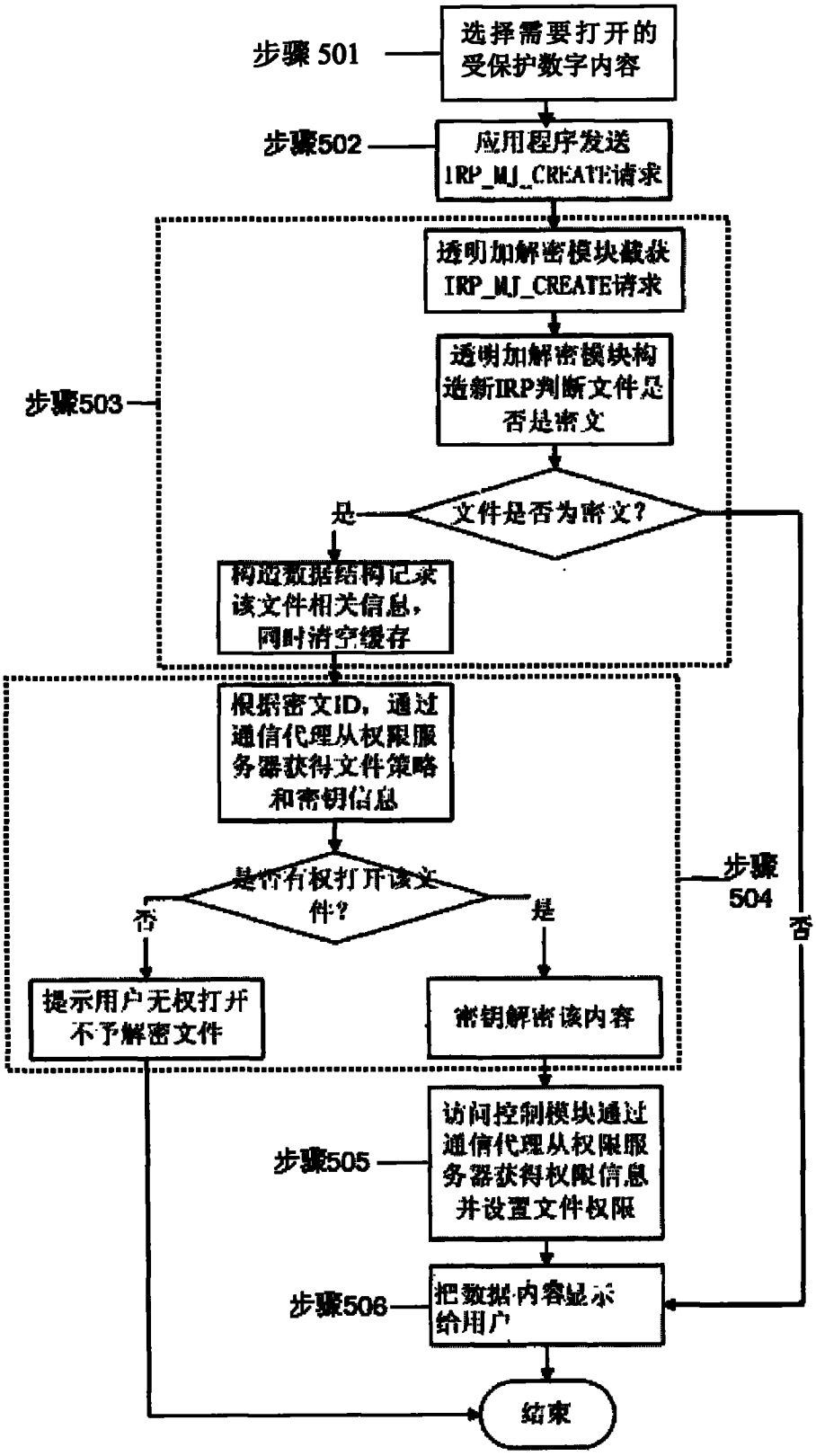

步骤 501
选择需要打开的受保护数字内容
步骤502
应用程序发送IRP_MJ_CREATE请求
步骤503
透明加解密模块截获IRP_MJ_CREATE请求
透明加解密模块构造新IRP判断文件是否是密文
是
文件是否为密文?
构造数据结构记录该文件相关信息，同时清空缓存
根据密文ID，通过通信代理从权限服务器获得文件策略和密钥信息
是否有权打开该文件?
否
是
步骤504
否
提示用户无权打开不予解密文件
密钥解密该内容
步骤505
访问控制模块通过通信代理从权限服务器获得权限信息并设置文件权限
步骤506
把数据内容显示给用户
结束

图 5

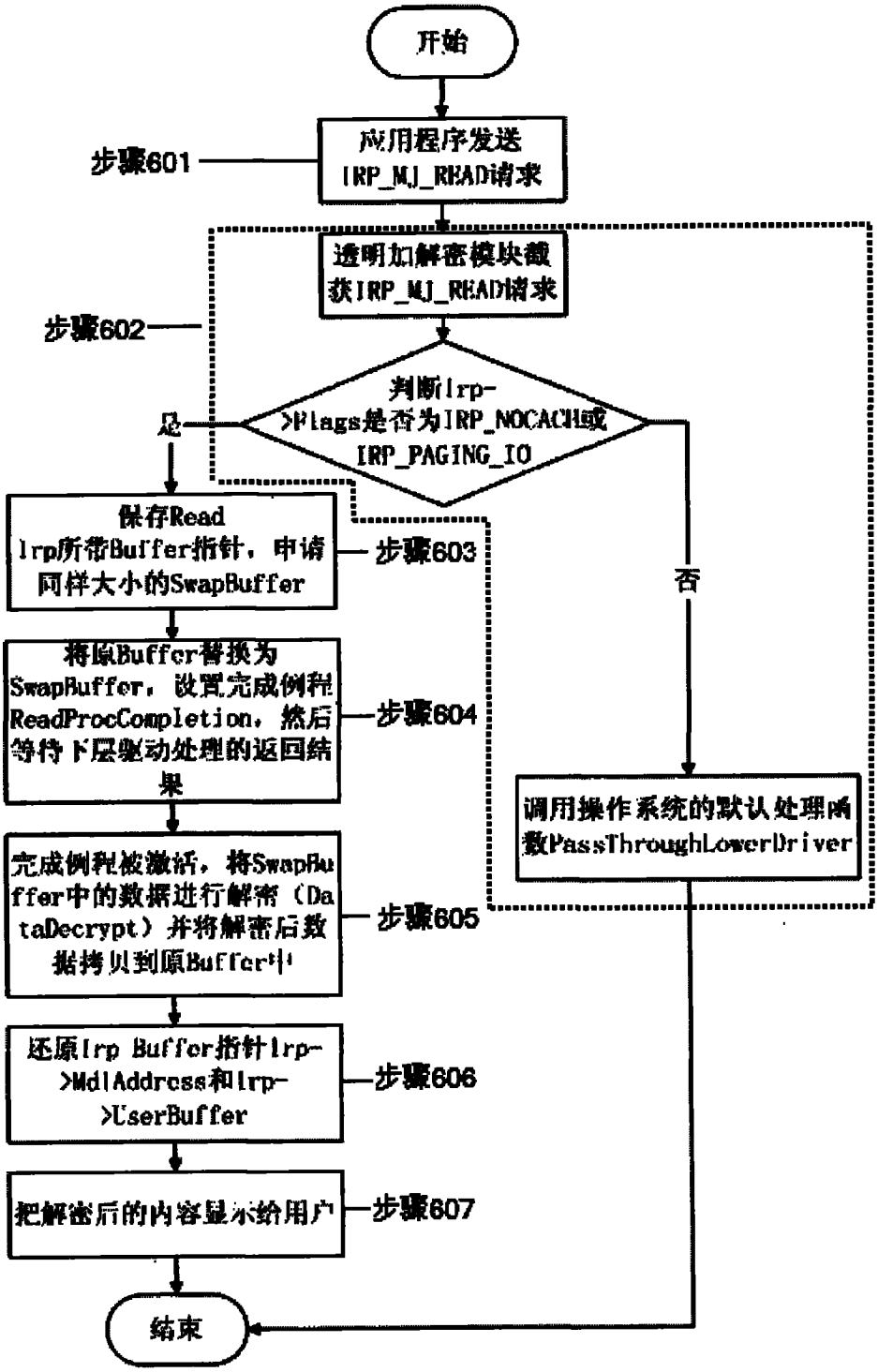

开始
步骤601
应用程序发送IRP_MJ_READ请求
透明加解密模块截获IRP_MJ_READ请求
步骤602
判断Irp->Flags是否为IRP_NOCACH或IRP_PAGING_IO
是
否
保存Read Irp所带Buffer指针，申请同样大小的SwapBuffer
步骤603
将原Buffer替换为SwapBuffer，设置完成例程ReadProcCompletion，然后等待下层驱动处理的返回结果
步骤604
调用操作系统的默认处理函数PassThroughLowerDriver
完成例程被激活，将SwapBuffer中的数据进行解密（DataDecrypt）并将解密后数据拷贝到原Buffer中
步骤605
还原Irp Buffer指针Irp->MdlAddress和Irp->UserBuffer
步骤606
把解密后的内容显示给用户
步骤607
结束

图 6

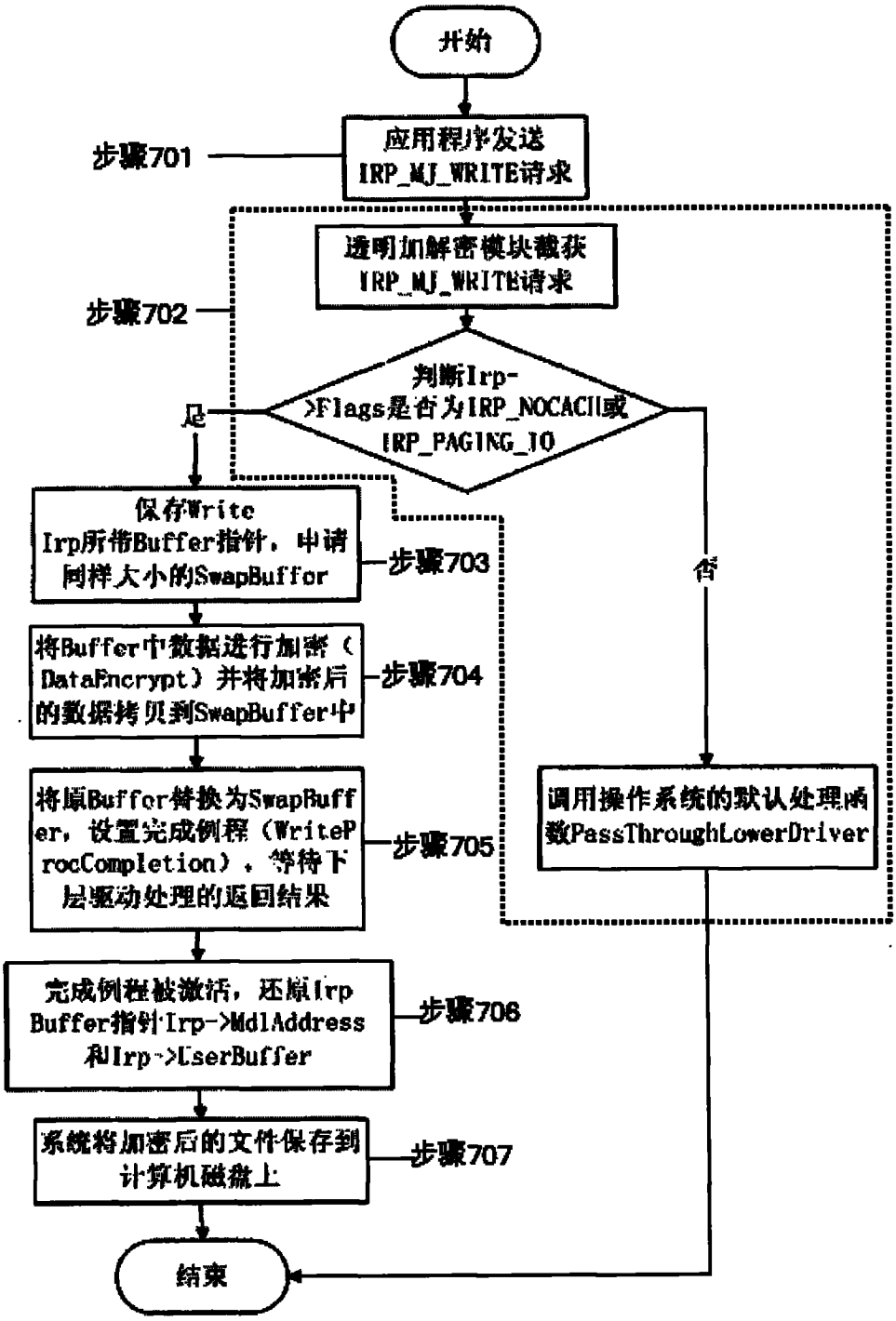

开始
步骤701
应用程序发送IRP_MJ_WRITE请求
透明加解密模块截获IRP_MJ_WRITE请求
步骤702
判断Irp->Flags是否为IRP_NOCACH或IRP_PAGING_IO
是
否
保存Write Irp所带Buffer指针，申请同样大小的SwapBuffer
步骤703
将Buffer中数据进行加密（DataEncrypt）并将加密后的数据拷贝到SwapBuffer中
步骤704
将原Buffer替换为SwapBuffer，设置完成例程（WriteProcCompletion），等待下层驱动处理的返回结果
步骤705
调用操作系统的默认处理函数PassThroughLowerDriver
完成例程被激活，还原Irp Buffer指针Irp->MdlAddress和Irp->UserBuffer
步骤706
系统将加密后的文件保存到计算机磁盘上
步骤707
结束

图 7

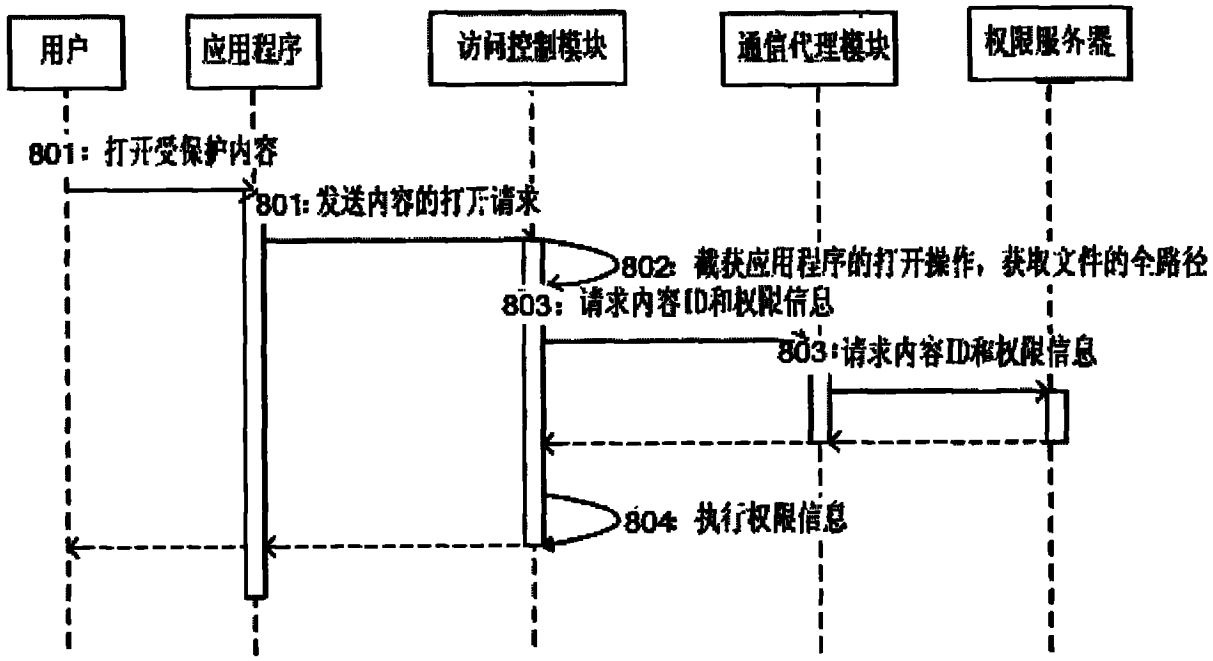

用户
应用程序
访问控制模块
通信代理模块
权限服务器
801：打开受保护内容
801：发送内容的打开请求
802：截获应用程序的打开操作，获取文件的全路径
803：请求内容ID和权限信息
803：请求内容ID和权限信息
804：执行权限信息

图 8