

1. 一种Android平台的文件透明加解密方法,其特征在于,包括以下步骤:

步骤一、选择需要保护的文件所在文件夹的路径并设置密码;

步骤二、根据用户输入的路径和密码,分别生成加密路径目录表和身份验证文件;

步骤三、扫描加密路径目录表,如果是第一次开启操作系统,按照表项将对受保护文件进行第一次初始化加密,所使用的密钥为第一密钥,第一密钥是将用户制定策略中的密码经sha1算法生成的,然后进行下一步骤;

步骤四、当用户触发解锁屏幕事件,则接受用户输入的密码短语,对密码短语进行哈希算法运算后与步骤二中产生的身份验证文件进行比对:如果不匹配,则解锁失败;如果匹配,则将密码短语采用sha1算法换算成密钥,将该密钥进行存储;

步骤五、当用户访问文件时,如果操作的文件或目录在加密路径目录表中,当用户发出写请求,调用步骤四所存储的密钥对文件进行加密;当用户发出读请求,调用密钥对文件进行解密;

步骤六、当用户触发锁定屏幕事件,则清除存放的密钥并锁屏。

2. 根据权利要求1所述的一种Android平台的文件透明加解密方法,其特征在于,所述哈希算法为MD5哈希算法。

3. 一种Android平台的文件透明加解密系统,包括用户模块和内核模块,其特征在于,用户模块包括配置单元和锁屏单元,内核模块包括密钥管理器单元和堆栈式文件系统单元;其中,

配置单元,用于接收用户制定策略,策略包括密码、身份验证文件、加密目录路径表,该密码经sha1算法生成第一密钥进行存储;身份验证文件输入至密钥管理器单元,加密目录路径表输入至堆栈式文件系统单元;

锁屏单元,将用户输入的密码短语输出至密钥管理器单元;

密钥管理器单元,用于将密码短语采用哈希算法运算后与身份验证文件进行匹配:若不匹配则解锁失败;若匹配一致则在解锁屏幕的同时,将密码短语采用sha1算法换算成第二密钥进行存储,当锁屏单元锁屏时,清除第二密钥;

堆栈式文件系统单元,当第一次开启操作系统,接收到加密目录路径表时调用第一密钥对加密目录路径表所对应的目录里的文件进行初始化加密;并覆盖在操作系统支持并挂载的所有文件系统之上,当用户发出读请求时调用密钥管理器中的第二密钥,对文件进行解密;当用户发出写请求时调用密钥管理器中的第二密钥,对文件进行加密。

4. 根据权利要求3所述的一种Android平台的文件透明加解密系统,其特征在于,所述用户制定的策略还包括是否启用加解密系统。

5. 根据权利要求3所述的一种Android平台的文件透明加解密系统,其特征在于,所述哈希算法为MD5哈希算法。

一种Android平台的文件透明加解密系统及方法

技术领域

[0001] 本发明涉及信息安全技术领域,特别是一种Android平台的文件透明加解密系统及方法。

背景技术

[0002] 随着信息时代的快速发展,互联网不断深入大众生活和工作的方方面面,成为不可或缺的一部分。计算机应用的普及和互联网及移动存储设备的发展,以前的纸质文件逐渐转变为电子,电子文件具有体积小,查看方便等诸多的优点,同时电子文件的易更改、易传播的特性,也会严重影响到了电子文件存储和交流的安全性。

[0003] 透明加密技术是近年来发展较为迅速的一种文件加密技术。所谓透明,是指对于授权用户,加解密过程是自动完成的,实现原理为在磁盘中以密文方式存放文件,读入时自动解密,保存到内存中,用户修改完内存中的副本后,再自动加密并写回磁盘。Windows中透明加密实现技术主要有两类:用户层的钩子透明加密技术和内核层的过滤驱动加密技术,且内核层的实现在性能,兼容性以及稳定性上优于用户层实现,同时技术难度也较用户层实现大。

[0004] 所基于的Android系统使用针对嵌入式设备所剪裁的Linux内核,它的设计体现了结构化设计的思想,具有很强的层次性,从底层到用户界面,层次主要包括:Linux内核,HAL(硬件抽象层),系统服务层,应用框架层和应用程序。Linux内核的文件系统的操作由具体文件系统维护几组操作表提供,其表项为函数指针,指向具体的操作代码。透明加密需要改变一些操作(如读写操作)的行为。实验证明,仅仅替换操作表技术上可行,但造成系统结构混乱,可维护性和可扩展性差。堆栈式文件系统是一种增量式开发模式,用于扩充原有文件系统的功能。该方式不用修改原文件系统的代码,而是覆盖在其上,过滤读写等操作,在过程中加入自己的处理方法,如加密和压缩等,以增强原有文件系统的功能。该模型由Erez Zadok提出,包括了FiST框架以方便构造该类文件系统。由于FiST框架构建于二〇〇〇年之前,缺少维护,且仅支持2.4到2.6的内核版本。

[0005] 现有的部分Android文件保护系统直接将个人电脑文件保护系统的思想应用在移动设备上,忽略了平台的差异:移动设备主要强调用户体验,不仅仅是功能实现。这些文件保护系统频繁要求用户输入密码,选择加解密文件,降低了设备的便利程度。另一方面,现有的透明加解密系统,减少了对用户操作习惯的影响,但是保护工作不全面:比如由于权限问题不能保护特定目录,不能保护SD卡上的文件(而SD卡正式用户数据的重要存放位置);或是只能保护某种指定格式的文件;或是对系统结合度低易受攻击;或是兼容性和扩展性低,比如只能支持部分特定版本的系统。

发明内容

[0006] 本发明所要解决的技术问题是克服现有技术的不足而提供一种Android平台的文件透明加解密系统及方法,本发明利用堆栈式文件系统实现技术,采用结合锁频界面尽可

能减少对用户操作的影响,紧密结合操作系统本身,实现对外受阻、对内无碍的加解密保护。

[0007] 本发明为解决上述技术问题采用以下技术方案:

[0008] 根据本发明提出的一种Android平台的文件透明加解密方法,包括以下步骤:

[0009] 步骤一、选择需要保护的文件所在文件夹的路径并设置密码;

[0010] 步骤二、根据用户输入的路径和密码,分别生成加密路径目录表和身份验证文件;

[0011] 步骤三、扫描加密路径目录表,如果是第一次开启操作系统,按照表项将对受保护文件进行第一次初始化加密,然后进行下一步骤;

[0012] 步骤四、当用户触发解锁屏幕事件,则接受用户输入的密码短语,对密码短语进行哈希算法运算后与步骤二中产生的身份验证文件进行比对:如果不匹配,则解锁失败;如果匹配,则将密码短语采用sha1算法换算成密钥,将该密钥进行存储;

[0013] 步骤五、当用户访问文件时,如果操作的文件或目录在加密路径目录表中,当用户发出写请求,调用步骤四所存储的密钥对文件进行加密;当用户发出读请求,调用密钥对文件进行解密;

[0014] 步骤六、当用户触发锁定屏幕事件,则清除存放的密钥并锁屏。

[0015] 作为本发明的一种Android平台的文件透明加解密方法的进一步优化的方案,所述哈希算法为MD5哈希算法。

[0016] 根据本发明提出的一种Android平台的文件透明加解密系统,包括用户模块和内核模块,用户模块包括配置单元和锁屏单元,内核模块包括密钥管理器单元和堆栈式文件系统单元;其中,

[0017] 配置单元,用于接收用户制定策略,策略包括密码、身份验证文件、加密目录路径表,该密码经sha1算法生成第一密钥进行存储;身份验证文件输入至密钥管理器单元,加密目录路径表输入至堆栈式文件系统单元;

[0018] 锁屏单元,将用户输入的密码短语输出至密钥管理器单元;

[0019] 密钥管理器单元,用于将密码短语采用哈希算法运算后与身份验证文件进行匹配:若不匹配则解锁失败;若匹配一致则在解锁屏幕的同时,将密码短语采用sha1算法换算成第二密钥进行存储,当锁屏单元锁屏时,清除第二密钥;

[0020] 堆栈式文件系统单元,当第一次开启操作系统,接收到加密目录路径表时调用第一密钥对加密目录路径表所对应的目录里的文件进行初始化加密;并覆盖在操作系统支持并挂载的所有文件系统之上,当用户发出读请求时调用密钥管理器中的第二密钥,对文件进行解密;当用户发出写请求时调用密钥管理器中的第二密钥,对文件进行加密。

[0021] 作为本发明的一种Android平台的文件透明加解密系统的进一步优化的方案,所述用户制定的策略还包括是否启用加解密系统。

[0022] 作为本发明的一种Android平台的文件透明加解密系统的进一步优化的方案,所述哈希算法为MD5哈希算法。

[0023] 本发明采用以上技术方案与现有技术相比,具有以下技术效果:(1)本发明利用堆栈式文件系统实现技术,采用结合锁屏界面尽可能减少对用户操作的影响,紧密结合操作系统本身,实现对外受阻、对内无碍的加解密保护;(2)保护Android终端用户存储在设备中的隐私数据,同时不改变用户操作习惯,不影响用户体验;(3)选用内核级加密方式,增强系

统安全性和加解密效率,与系统结合紧密,抗攻击能力强,具有较高的安全性;由于系统核心加解密模块工作在最底层,而与用户交互的界面为最顶层,系统设立中间件,辅助内核模块和上层应用的通信;通过JNI (Java本地访问接口) 与中间层通信,中间层再通过ioctl的方式完成与内核模块通信;(4) 本发明可以支持目前流行的Android4.x使用的3.x内核,本系统设计文件系统,在文件读取和写入操作上进行处理,并将其它文件操作直接定向到底层文件系统,灵活易移植;(5) 加解密对用户透明,对用户操作干扰小,用户体验好;便于部署和移植;具有较高的性能;不区分文件格式,用户可用来加密任意形式的文件;不区分存储位置,可加密手机应用程序信息和存储扩展卡中信息,实现对SD卡文件数据的保护。

附图说明

- [0024] 图1是本系统的各模块间的相互作用。
- [0025] 图2是本发明透明加解密文件系统工作原理图。
- [0026] 图3是上层栈式加解密文件系统与下层实际文件系统之间的示意图。
- [0027] 图4是锁屏单元工作流程图。

具体实施方式

- [0028] 下面结合附图对本发明的技术方案做进一步的详细说明:
- [0029] 一种Android平台的文件透明加解密方法,包括以下步骤:
- [0030] 步骤一、选择需要保护的文件所在文件夹的路径并设置密码;
- [0031] 步骤二、根据用户输入的路径和密码,分别生成加密路径目录表和身份验证文件;
- [0032] 步骤三、扫描加密路径目录表,如果是第一次开启操作系统,按照表项将对受保护文件进行第一次初始化加密,然后进行下一步骤;
- [0033] 步骤四、当用户触发解锁屏幕事件,则接受用户输入的密码短语,对密码短语进行哈希算法运算后与步骤二中产生的身份验证文件进行比对;如果不匹配,则解锁失败;如果匹配,则将密码短语采用sha1算法换算成密钥,将该密钥进行存储;
- [0034] 步骤五、当用户访问文件时,如果操作的文件或目录在加密路径目录表中,当用户发出写请求,调用步骤四所存储的密钥对文件进行加密;当用户发出读请求,调用密钥对文件进行解密;
- [0035] 步骤六、当用户触发锁定屏幕事件,则清除存放的密钥并锁屏。
- [0036] 所述哈希算法为MD5哈希算法。
- [0037] 如图1所示是本系统的各模块间的相互作用,一种Android平台的文件透明加解密系统,包括用户模块和内核模块,用户模块包括配置单元和锁屏单元,内核模块包括密钥管理器单元和堆栈式文件系统单元;其中,
- [0038] 配置单元,用于接收用户制定策略,策略包括密码、身份验证文件、加密目录路径表,该密码经sha1算法生成第一密钥进行存储;身份验证文件输入至密钥管理器单元,加密目录路径表输入至堆栈式文件系统单元;
- [0039] 锁屏单元,将用户输入的密码短语输出至密钥管理器单元;
- [0040] 密钥管理器单元,用于将密码短语采用哈希算法运算后与身份验证文件进行匹配;若不匹配则解锁失败;若匹配一致则在解锁屏幕的同时,将密码短语采用sha1算法换算

成第二密钥进行存储,当锁屏单元锁屏时,清除第二密钥;

[0041] 堆栈式文件系统单元,当第一次开启操作系统,接收到加密目录路径表时调用第一密钥对加密目录路径表所对应的目录里的文件进行初始化加密;并覆盖在操作系统支持并挂载的所有文件系统之上,当用户发出读请求时调用密钥管理器中的第二密钥,对文件进行解密;当用户发出写请求时调用密钥管理器中的第二密钥,对文件进行加密。

[0042] 所述用户制定的策略还包括是否启用加解密系统。所述哈希算法为MD5哈希算法。

[0043] 图2是本发明透明加解密文件系统工作原理图。读取受保护文件的过程:如果用户为非授权用户(未通过锁屏界面身份认证的用户),操作失败。对于授权用户,则传递读请求至下层文件系统,获得返回的文件内容,此时内容为密文。向密钥管理器单元请求密钥,利用该密钥对密文进行解密。将获得的明文从内核空间拷贝至用户空间。

[0044] 修改相应文件属性,完成读操作。

[0045] 写入受保护文件的过程:如果用户为非授权用户(未通过锁屏界面身份认证的用户),操作失败。

[0046] 对于授权用户,请求密钥,利用密钥将用户空间传递的数据所在缓冲区加密。

[0047] 将缓冲区内容传递到下层文件系统,由其写入到磁盘。

[0048] 修改相应文件属性,完成写操作。

[0049] 向内核注册文件系统模块,需要实现的文件系统操作方法有:超级块操作方法、节点操作方法、文件操作方法。

[0050] 本系统文件系统形成一个堆栈结构,上层文件系统就是开发的堆栈式加解密文件系统。下层文件系统是实际的文件系统(但也有可能是另一个堆栈式文件系统,如果该堆栈文件系统足够“透明”,也可以认为实际文件系统)。

[0051] 由于上层文件系统依赖于下层文件系统的操作方法和数据结构,所以第一步应构建上下层文件系统数据结构间的关系。

[0052] 如图3所示是上层堆栈式加解密文件系统与下层实际文件系统之间的示意图,以file结构为例:这里upper_file是本层文件系统的对象,lower_file是与之相对应的下层文件系统的对象。两者通过upper_file的private_data指针联系。文件操作时调用upper_file中的file_ops文件操作表中的函数。因为发给堆栈式文件系统上层的操作请求经过处理后,会传递给下层。具体说,上层file_ops操作表中的函数会调用下层file_ops表中对应函数。类似的,本层文件系统的dentry、inode以及address_space结构体,与下层文件系统的相应数据结构联系,传递各自的操作请求。

[0053] 完成上下层关键数据结构间关系的构建,为下面的操作奠定基础。

[0054] 除了文件读写操作外,其余文件系统操作的工作仅仅是调用下层文件系统对应函数,或是使用通用处理函数,以实现“通过”。如果需要,还要更新下层文件系统数据结构的相关域,如文件访问时间,文件当前读取位置等。以读取目录文件为例:此时上下层文件系统关系已构建完成,因为需要调用下层文件系统的操作,首先通过该关系寻找到与本层file对应的下层文件系的数据结构lower_file。通过VFS层通用函数读取lower_file的目录信息。这里的目录信息是下层文件系统的目录信息,但由于没有对该操作进行处理,可以直接作为本层的目录信息返回。VFS_readdir读取完信息后会更新lower_file的访问时间,但上层的file访问时间需要手动更新。这里通过拷贝下层文件访问信息来达到同步

两者的目的。

[0055] 其余需要"通过"的文件操作实现与此类似。

[0056] 对文件读写操作具体内容如下:在调用底层文件系统读入数据之后,将内容返回到用户空间之前,将缓冲区进行解密。同样,在写操作时,在调用底层文件系统写操作之前,加密缓冲区。

[0057] 其中加解密过程可以使用内核加解密框架实现,以节约时空开销,缩减开发成本。

[0058] 配置单元的实现:策略配置界面是本系统控制核心,其功能在上文发明内容一节已给出描述。它维护两个文件:用于身份验证时比对的密码md5文件和加密目录表文件。它作为普通的Android应用程序,但需要持有管理员权限,需要开机时被自动,然后扫描加密目录表,逐一为表中的目录挂载加解密文件系统。

[0059] 内核密钥管理器单元的实现:

[0060] 内核密钥管理器单元,作用类似与内核密钥环。不直接使用内核密钥环是因为其过于复杂,应尽量缩小内核占用的嵌入式设备有限的时间资源和空间资源。内核密管理器主要包括一个可以被其它模块访问的全局缓冲区,用来存放密钥。该模块直接与应用层锁屏界面通过ioctl方式通信。定义TRANSPARENT_IOCTLAUTHEN命令为接受来自用户输入的密码,经过md5运算后,与本地存放的身份验证文件进行比对,若一致则返回验证通过,并将密码经sha1运算转换成128bit密钥存放在全局缓冲区内。同时定义TRANSPARENT_IOCTLCLEARKEY命令为锁屏幕时需要完成的清除密钥管理器中密钥的任务。

[0061] 图4是锁屏单元工作流程图。锁屏单元的实现:锁屏应用接受用户输入的密码短语,并读取是否开启加解密服务的开关量。

[0062] 这些信息被拷贝到内核空间,在这里密码短语被哈希算法计算,与身份验证文件中存放的MD5值比对以确认用户身份。如不相符,则提示解锁失败,用户可进行有限次的尝试;如果相符,转下一步。

[0063] 如果身份验证成功,则判断是否开启加解密服务的开关量;如果服务不开启,则转下一步;如果服务开启,则将密钥进行变换存入密钥管理器,执行下一步。

[0064] 解锁屏幕。

[0065] 锁屏单元提供两个与用户交互的控件组:密码输入控件组和一个二值开关。前者用于记录用户输入的密钥,后者决定是仅解锁手机操作系统还是即解锁操作系统又解锁加密文件。锁屏单元本身只接受并缓存用户密码,不负责身份验证。处于安全考虑,身份验证由内核密钥管理单元完成。锁屏单元通过JNI(Java本地访问接口)与中间层动态库通信,中间层在通过ioctl的方式与内核通信,将密钥从用户空间传递到内核空间。考虑到试用本系统的用户分为手机操作系统定制商和个人用户两种,所以JNI部署既可以在应用框架层提供上层调用接口,作为系统API被应用程序调用,或者被第三方应用程序扩展,做到与系统紧密结合,又可以直接编译作为独立的动态库文件被应用程序加载,使得个人用户部署过程简单方便在Android系统中锁屏和解锁屏是以BroadCast形式传递的消息。本系统锁屏应用模块监听该消息,做出相应的处理。代码中的openScreen和closeScreen为解锁/开锁屏幕时执行的动作,主要是调用JNI接口,解锁是完成身份验证和密钥传递,锁屏时清除密钥。

[0066] 以上所述的具体实施方案,对本发明的目的、技术方案和有益效果进行了进一步的详细说明,所应理解的是,以上所述仅为本发明的具体实施方案而已,并非用以限定本发

明的范围,任何本领域的技术人员,在不脱离本发明的构思和原则的前提下所做出的等同变化与修改,均应属于本发明保护的范围。

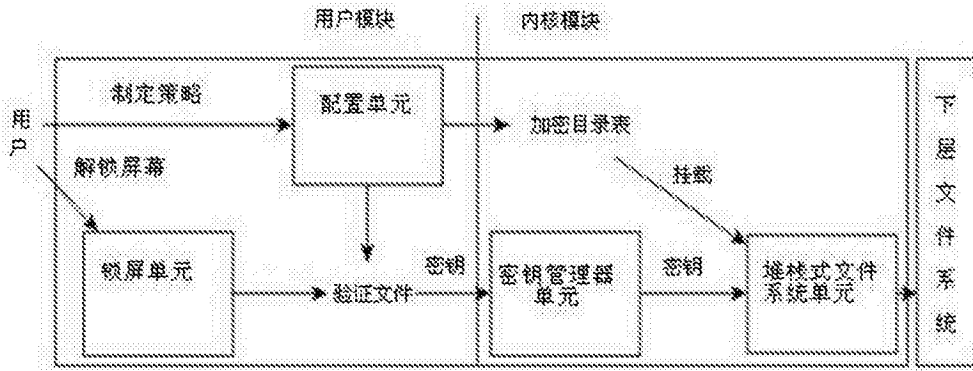


图1

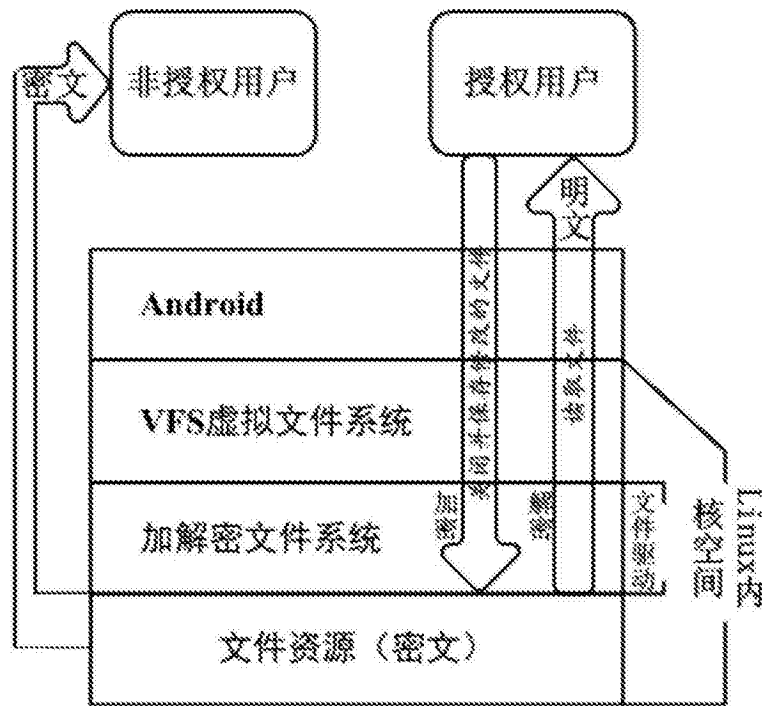


图2

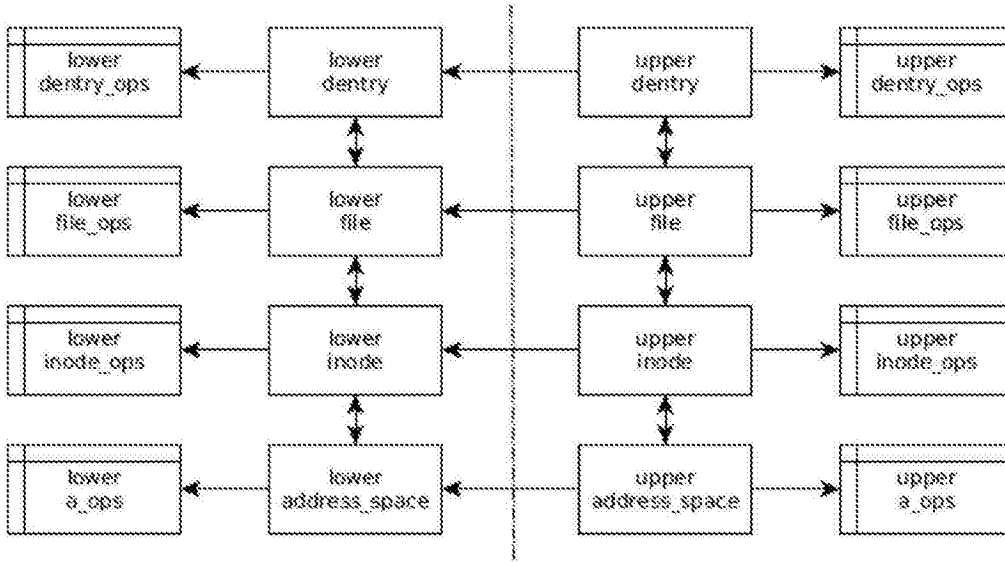


图3

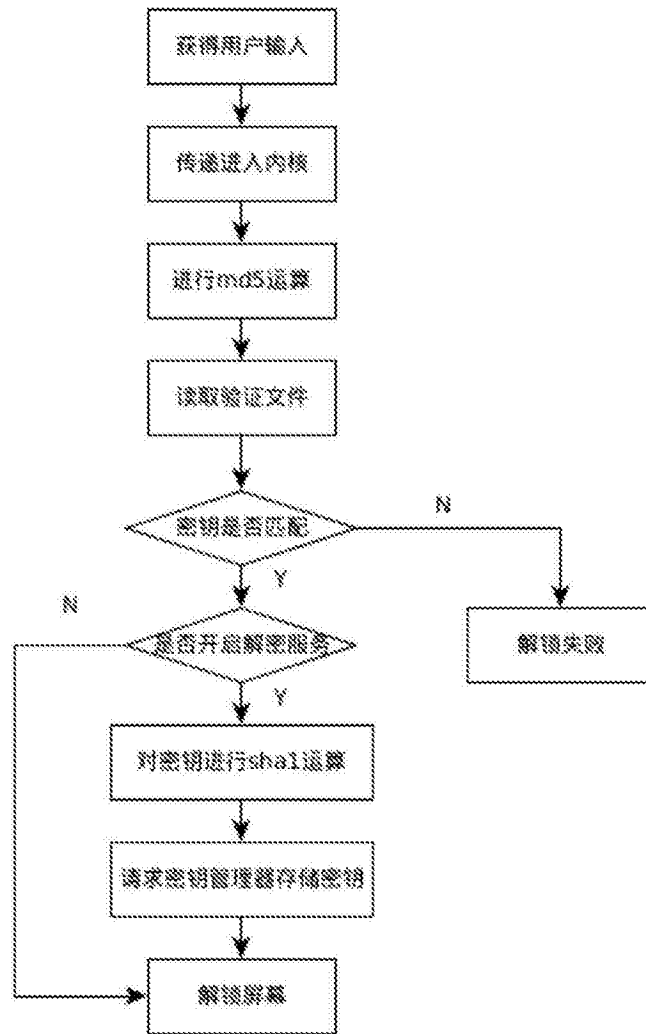


图4