第十三届“挑战杯”全国大学生课外学术科技作品竞赛

基于 TPM 的便携式跨平台网络安全云盘

研 究 报 告

项目作者：何兴平(2011 级本科)

合作者：郭一新（2012 级本科）刘蓬博（2012 级本科）

中国 CHINA

2013 年 6 月 1 日

摘要

随着信息时代的到来，日常办公自动化 OA 系统已在各行业各领域占据越发重要的地位，人们在享受办公智能化所带来的巨大便利的同时，也面临着信息安全问题的严峻挑战。政府机关、企业及个人每天都在或多或少的与重要信息相联系，如何提高涉密信息的存储、传输安全性，防止内部涉密信息泄漏是当前信息安全领域的重要课题。

本项目研究关于涉密文件管理的解决方案，提出了基于 TPM 可信计算技术的硬件动态密钥联网验证方案及基于云计算技术的文件安全存储与抗攻击方案。通过自主研发低成本硬件级(<10 元)的 USB 云盘，实现多系统全平台(手机/平板/笔记本/台式机)的无限容量硬件级文件加密与管理，本系统将非对称式加密技术与对称式加密技术有机结合，有效实现研发资料、财务报表等重要数据的随时随地安全存取与分发共享。

研发过程中首先对现有的加密方式进行深入调研，详细分析了现有数据加密方式在使用环境、安全强度、易用性方面的优缺点，并对研究现状作分析评估，提出改进方案。其次，从整体出发对本系统的硬件与软件解决方案进行阐述，分析了基于 AVR 单片机的完全自主知识产权 USB-TPM 可信计算动态密钥验证装置的原理与制作过程、基于 ASP.NET 服务器与 NGINX 负载均衡服务器的云存储架构设计与业务流处理程序解决方案、基于 Visual Basic.NET 的云计算文件管理客户端研发方案及基于 LibUSB32 的底层驱动实现原理、基于 PhoneGap 的移动平台客户端研发方案及 LINUX-OTG 的底层驱动实现原理。之后，分别从系统硬件部分、系统客户端部分、系统服务器端部分三个方面详细阐述了具体技术实现步骤。最后对本系统从加密安全度、数据加密速率、解密速率、系统兼容性、系统稳定性等多个方面给出测试报告，充分证明其可靠性、易用性、安全性、稳定性均达到预期要求。

本系统的创新点在于自主研发的低成本免驱动 TPM 可信计算硬件动态密钥联网验证方案，对称式加密与非对称式加密相结合的文件加密方案，云存储技术在信息安全领域应用的高效管理解决方案。有效解决了高数量、大容量数据的安全存储与分发报送问题。

关键词：TPM；OTG；LIBUSB；可信计算；云计算

目 录

1 绪论

1.1 研究背景和意义

由于学校在日常办公活动中经常需要在各个部门与个人之间传输涉密信息，如考试试题、部门工作报告、财务审计表、项目研发资料等。不仅在学校内部如此，通过市场调查还了解到在诸多政府机关及企事业单位都有对涉密信息报送方面的苦恼，这就带来了涉密文件安全共享与管理的巨大市场需求。而现有的文件加密系统及办公自动化系统通常对机密信息的存储与传输未作严格控制，不论从文件加密强度及系统运行效率都无法满足现有市场需求。现有文件管理系统普遍存在着加密速率低，反应迟钝，易被暴力破解、加密文件易丢失或无故损坏等情况，这就带来了涉密文件管理系统的巨大技术需求。而本系统正是在综合考虑以上问题的基础之上开发的高效率、易操作基于 TPM 硬件验证与云计算存储技术的文件管理系统。

1.2 研究历史和现状

随着 Internet 的不断发展，新的网络应用层出不穷，而且这些应用越来越贴近人们的经济利益和个人隐私，因此人们意识到要保证通信的高度安全必须在通信过程中利用密码技术。在传输线路缺乏物理保护的情况下，为了防止口令被窃听，人们开始利用加密技术和高级数学函数对基于口令的认证方案进行了改进。但是，用户为了方便记忆，口令通常较短，并且具有一定的规律性，甚至带有用户的一些个人信息，这样的口令很难对抗猜测攻击。将密码学引入身份认证技术后，口令认证方案就逐渐演变成基于密钥的认证方案。

目前国内外身份认证的研究主要集中在口令认证、动态口令认证、基于智能卡的认证、基于生物特征识别的认证、基于 PKI 的认证和基于 Kerberos 的认证几种方式上。

(1)口令认证是最传统的认证方式，系统将用户输入的 ID 和口令与数据库中的用户信息进行比对，从而确定用户身份。用户口令一般较短且容易猜测，因此这种方案不能抵御口令猜测攻击;另外，攻击者可能窃听通信信道或者进行网络窥探，口令的明文传输使得攻击者只要能在口令传输过程中获得用户口令，系统

就会被攻破。通过一些措施可以有效地改进口令认证的安全性，如通过增加口令的强度，提高抗穷举攻击和字典攻击的能力，将口令加密防止在传输中被窃听等，但增加了用户的负担。

(2)动态口令认证是指在用户登录系统、验证身份过程中，送入系统的验证数据是动态变化的。在登录系统时，用户只需输入 ID 和当前的动态口令，系统根据用户的标识符计算出该用户的当前口令，与用户输入的口令比较，对用户的身份进行认证。目前主要有两种技术：基于时间同步认证技术和基于挑战/应答方式的非同步认证技术。

①基于时间同步技术是以用户登陆时间作为随机因素，流逝时间作为变动因子进行认证。它要求用户密码卡和认证服务器所产生的密码在时间上必须同步。

②基于挑战/应答方式的非同步认证技术是在用户登陆时，系统随机生成一个信息作为挑战值发给用户。用户利用事先约定好的单向哈希函数对自己的秘密信息和随机信息进行运算，并把计算结果返回给系统，系统用同样的方法做验算即可验证用户身份。

动态口令技术采用一次一密的方法，有效保证了用户身份的安全性。但是用户每次登录时需要通过键盘输入一长串无规律的密码，一旦输错就要重新操作，使用起来不是很方便，而且，没法解决设备和软件认证的问题，无法有效防止木马程序的攻击。

(3)基于智能卡的认证技术是最为安全可靠的认证手段之一，其原理是在智能卡上存储着用户的个人的秘密信息。在进行鉴别时，用户首先输入用户的身份识别码 PIN，智能卡鉴别 PIN，确认后，计算机就可以从智能卡中读取用户的秘密信息并传递到远程服务器。服务器根据收到的秘密信息对用户的身份进行认证。由于基于智能卡的识别方法是一种双因素的识别方式(PIN+智能卡)，PIN 或者智能卡二者缺一不可，无论是盗用用户的 PIN，或者盗用用户的智能卡，入侵者都不能通过该系统的认证。但对于智能卡认证，需要在每个认证端添加读卡设备，增加了硬件成本，不如口令认证方便和易行。

(4)基于生物特征识别的认证是指利用人体本身的生物特征进行认证，如人脸、指纹，因为这些特征的唯一性，使其具有非常高的安全性，最适合于面对面的识别与认证。

(5)基于 PKI(Public Key Infrastructure)的认证技术是近几年发展起来的一种方便、安全的身份认证技术。PKI 不仅是一种技术，包括硬件和软件，还包括服务、策略、规程、协议等多方面的内容。PKI 中基本的元素就是数字证书，所有安全的操作主要通过证书来实现。证书由一个公正可信的权威机构(Certification Authority，CA)负责签发。CA 为某一用户 A 颁发证书，并用自己的私钥对证书签名，另一用户想验证 A 的身份时，利用以的公钥验证 A 的数字证书的完整性，从而判断 A 是否是其所声称的用户。PKI 提供的认证机制安全性较好，比较适合网上的安全认证，也是目前应用较多的身份认证方法。但它不可避免地存在着某些缺陷，如:在发布最初的证书时，如何验证一个远程用户提供的信息的真实性问题，用户私有密钥保存的安全性问题，用户用于取出私钥的通行字的质量问题，证书废止与证书废止列表刷新的时间差问题等。此类问题在理论上虽不然解决但在具体实施中却很困难。

(6)Kerberos 的基本原理是利用对称密码技术，通过可信的第三方(密钥分发中心，KDC)来提供身份认证服务，并在用户和服务器之间建立安全信道，可以提供安全的网络鉴别，允许个人访问网络中不同的机器。Kerberos 模型涉及客户端、应用服务器、中央数据库、认证服务器(Certification Server，简称 CS)和票据授予服务器(Ticket Granting Server，TGS)。其中中央数据库是安全服务的关键部分，在库中存有安全系统的安全信息，包括用户注册名及相关口令、工作站和服务器的网络地址、服务器的密钥以及访问控制列表等。

1.3 市场分析

目前，随着计算机网络技术、信息技术的快速发展，涉及信息安全的数字化产业正在世界范围内迅速崛起。无论是政府、企业还是个人都在依赖计算机来存储重要信息，并逐渐向网络传输与网络存储发展。这些信息无论是私人信息还是部门信息，无论是军工信息还是民用信息，在进行处理和传递前都要以数字文件形式存储在计算机、服务器或其他数码介质上，因此数字文档的安全存储成为实现信息安全的首要条件。

具体说来，数字文档的安全存储即是确保存储在计算机系统中的重要信息和数据免受意外的或恶意的破坏、更改、泄漏，也就是确保数字文档的保密性、完整性、可用性、真实性。但是由于使用数字文档的存储最终目的是实现信息的使

用，因此在保证安全存储的同时，还应确保电子文档的可操作性。

在安全产品方面，我国同国外依然有一定差距。主要表现在：一我国总体研发技术落后于国外，二我国信息安全认可度与普及度不高。我国信息安全技术虽然起步较晚，但发展迅速，与国际先进国家的差距正在逐步缩小。我国从 80 年代中期开始研究计算机网络文件信息安全系统，并在各种信息系统中陆续推广应用，其中有些技术已赶超国际水平，从而我国把信息安全保密技术推进到新的水平。从 90 年代中期开始，我国进入互联网发展时期，其发展势头迅猛，更加速了信息安全网络化的发展。

通过市场调研，我们可以发现现在市场上具有诸多类似的网络硬盘服务，如 115 网盘、百度云网盘、360 云盘、QQ 网络硬盘、华为网盘、金山网盘等，但通过 WINSOCK 抓包，我发现以上公共性网盘均未对用户数据进行加密操作，如果绕开防盗链机制的干扰，用户文件下载地址将完全暴露，这是非常可怕的！而根据艾瑞咨询的一份权威调查表明，全国近 80%使用网络硬盘的网民，不会选择将重要资料存放在网络中，原因皆是出于安全性的担忧，而本项目正有效解决这一难题。

基于用户体验及使用成本方面的考虑，诸多网络硬盘服务商均未支持数据文件硬件加密功能。而本项目就以上问题进行重点攻关，通过对比近十套方案后，才最终确定了该高安全性、低成本、TPM 可信计算联网验证、云存储方案。由于量产成本低（<10 元），应而具有极高的市场价值。由于充分考虑用户体验，用户几乎不会因为增加硬件验证环节而占用操作时间，并且可以支持多终端（手机/平板/笔记本/台式机）的文件操作，应而具有极高的实用价值。

本系统客户端在设计中，充分照顾用户的使用习惯，操作透明化将客户端主界面嵌入与普通资源管理器极其相似的文件资源管理器。通过该资源管理器，可直接浏览文件名称，文件类型，文件图标，就像操作本地文件一样轻松。客户端主界面支持文件拖拽，用户可将欲发送的文件直接拖动到主界面，系统将自动完成文件的校验、加密、上传，提高办公效率。

2 系统总体设计

2.1 基于 TPM 可信计算技术的身份认证技术

基于 TPM(Trusted Platform Module)的身份认证方式是近几年发展起来的一种方便、安全的身份认证技术。它采用软硬件相结合的强双因子认证模式，很好地解决了安全性与易用性之间的矛盾。TPM-USB 是一种 USB 接口的硬件设备，它内置单片机或智能卡芯片，可以存储用户的密钥或数字证书，利用 TPM 内置的密码算法实现对用户身份的认证。

在系统结构方面，TPM 是一个典型的嵌入式系统，由微处理器 CPU、存储单元，包括随机存储器 RAM、程序存储器 ROM(FLASH)、用户数据存储器(EEPROM)、芯片操作系统 Cos(Chip Operating System)以及 USB 接口等部分组成，装有 COS 的 TPM-USB 相当于一台微型计算机，不仅具有数据存储功能，同时具有命令处理和数据安全保护等功能。

TPM 的核心部分是其芯片上的卡操作系统 COS。TPM 安全性的关键就在于其操作系统的安全机制。COS 的主要功能是：控制 TPM 与外界信息进行交换，管理 TPM 存储器中的文件系统，并在 TPM 内完成各种命令的处理。COS 系统是由传输控制、文件管理、安全体系、命令解释四个功能模块组成的，通过安全体系和传输控制，使得 TPM 有很高的安全性。

TPM 安全体系对所传送的信息进行安全性的检查和处理，防止非法的窃听或侵入。安全体系包括三部分：安全状态、安全属性和安全机制。安全状态是指当前 TPM 所处的一种系统状态，这种安全状态是在 TPM 完成复位应答或完成某个命令后得到的，安全属性是 TPM 执行特定的命令所需要的安全条件，只有满足了这个安全条件，命令才能执行，安全机制是指安全状态实现转移所采取的方法和手段，它是与安全状态和安全属性相联系的。

2.1.1 基于 TPM 身份认证技术的优势

与基于口令的认证方式相比，在安全性方面，TPM 身份认证的优势具体有：

(l)双因子认证

每一个 TPM-USB 硬件都具有硬件 PIN 码保护，PIN 保护和 TPM 硬件构成了用户使用 TPM 的两个必要因素，即所谓“双因子认证”。用户只有同时取得了 TPM-USB 和用户 PIN 码，才可以登录系统。即使用户的 PIN 码被泄露，只要用户持有的 TPM-USB 不被盗取，合法用户的身份就不会被仿冒；如果用户的 TPM-USB 遗失，拾到者由于不知道用户 PIN 码，也无法仿冒合法用户的身份。

(2)带有安全存储空间

TPM-USB 具有一定容量的安全数据存储空间，可以存储数字证书、用户密钥等秘密数据，对该存储空间的读写操作必须通过程序实现，用户无法直接读取，其中用户私钥是不可导出的，杜绝了复制用户数字证书或身份信息的可能性。

(3)硬件实现加密算法

TPM-USB 内置 CPU 或智能卡芯片，可以实现 PKI 体系中使用的数据摘要、数据加解密和签名的各种算法，加解密及签名运算在 TPM-USB 内进行，保证用户密钥不会出现在计算机内存中，杜绝用户密钥被黑客截取的可能性。

(4)便于携带，安全可靠

TPM-USB 体积小，拇指大小，非常方便携带，方便用户在不同的平台进行登陆、签名，并且密钥和证书不可导出。TPM-USB 的硬件不可复制，安全性高。

认证技术	特点	缺点	主要产品
ID 密码认证	简单易行	易被暴力破解	普通软件
智能卡认证	成本低廉	易被内存扫描	智能加密卡
动态口令认证	安全性高	操作繁琐	动态令牌
生物特征认证	安全性高	稳定性低	指纹识别系统
可信计算认证	安全性高 成本低廉	无	TPM 芯片

表 2.1

2.1.2 基于 TPM 身份认证技术的实现原理

AVRUSB 技术是利用高性能的 8 位 RISC 架构的 AVR 单片机，使用单片机的 IO 口来模拟 USB 的通信端口，由软件来实现 USB 通信协议，将普通的 AVR

单片机模拟成一个 USB 低速设备，实现 AVR 单片机与计算机之间的通信和控制。

AVRUSB 技术的基本原理就是利用 AVR 单片机的普通 IO 端口来模拟 USB 的硬件端口进行通信。因为低速 USB 设备的速度是 1.5Mbps，而 AVR 单片机是单指令周期的，在单片机使用 12MHz 的时钟频率时，正好是 1.5MHz 的 8 倍。也就是说，单片机每 8 条指令就精确完成一个数据位的采集。采用这种方法时，对单片机的时序要求非常严格，所以软件的核心部分代码完全由汇编语言实现。

AVRUSB 技术具有如下特点：

低成本：传统的单片机与计算机进行 USB 通信，需要使用专用的接口芯片进行 USB 协议转换，如 CP2101、FT232、CH342、PDIUSBD12、SL811 等。像 CP2101、FT232 这样的芯片使用起来虽然简单，但是功能比较单一；而 PDIUSBD12、SL811 功能较强，但是使用复杂。并且这些专用芯片的价格都相对较高，增加了系统的成本。而 AVRUSB 简单易用，成本低廉，只需要一个普通的低成本 AVR 单片机以及很少的几个外部元件，就可以组成一个 USB 系统。

硬件要求低：AVRUSB 的代码为 AVR GCC 编译器做了高度优化，同时也完全兼容于更专业的 IAR C 编译器。程序编译后在最小情况下还不到 2KB，因此绝大部分的 AVR 单片机都可以使用 AVRUSB（只要支持外部中断 INT0，Flash 容量不小于 2KB 就可以实现 AVRUSB 的功能）。这样在很多低成本的小容量 AVR 单片机上也可以使用 AVRUSB，如 ATtiny2313、ATmega45、ATmega48 等，因此 AVRUSB 技术具有很高的实用价值。

容易开发：AVRUSB 提供了一个完整而又简单易用、成熟稳定的应用程序框架。这个框架包括了底层（单片机部分）和上层（PC 部分），单片机可以使用 GCC（或者 IAR）编程；PC 上则可以使用各种通用编程软件，如 Windows 下使用 VC、VB、Delphi，Linux 下使用 GCC 等等。用户可以在这个框架基础上添加和扩展各种功能，快速开发出适合于各种需求的单片机控制系统，而且 AVRUSB 支持 Windows、Linux、Mac OS 等多种操作系统，具有很好的跨平台特性。

本系统的中使用到的 TPM-USB 即是基于 AVRUSB 技术的应用实现。通过 AVRUSB 技术模拟 USB1.1 低速接口，片内编写了 HID 设备驱动描述符，动态口令自动更新程序，用户密钥全程加密程序，充分利用 ATmega8 上的 EEPROM 及寄存器资源，以较低的成本实现了 TPM-USB 的基本功能。

2.2 系统客户端设计方案

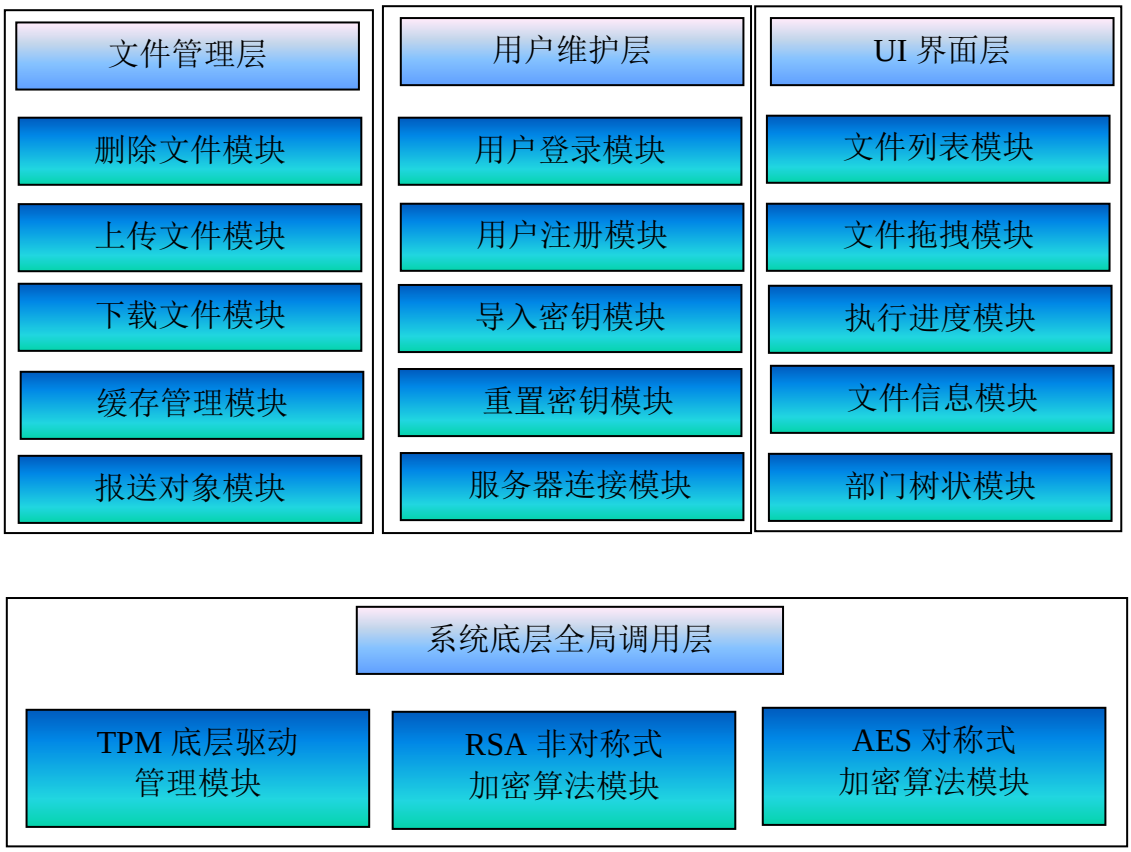


图 2.1

文件管理功能：透明化浏览及管理云服务器端的的文件，实现远程删除文件、上传文件、下载文件的功能，并扩展文件报送功能，可指定文件报送部门和报送对象。通过在客户端加入缓存管理，大幅度提高文件加密解密及实时同步传输的速率。

用户维护功能：实现基本的新用户注册并绑定 TPM-USB，普通用户登录并验证 TPM-USB，TPM-USB 丢失时重置密钥的完整用户安全管理功能。此外用户还可对系统进行设置，管理云服务器地址端口数据。

UI 界面模块：在保证界面简洁美观的同时，构建了一个文件管理器，可直接查看文件名称、文件类型、文件图标，透明直观。实现文件的即时拖拽，用户可将客户端内文件直接拖动到本地磁盘，亦可从本地磁盘直接拖动文件到客户端。执行进度精确显示，可看到当前的云同步进度，加密解密进度。部门树状管理，通过树状结构直接寻找对应部门的报送对象。

系统底层全局调用模块：系统核心模块，TPM-USB 驱动对接及密钥识别、修改、动态更新功能。基于 RSA 算法，将公钥存储在云端数据库，密钥存储在 TPM-USB，有效提高数据安全性。基于 AES 算法，将大容量文件快速加密，并生成最高 256 位的密钥，密钥经公钥加密后存储在云端服务器。

2.3 系统服务器端设计方案

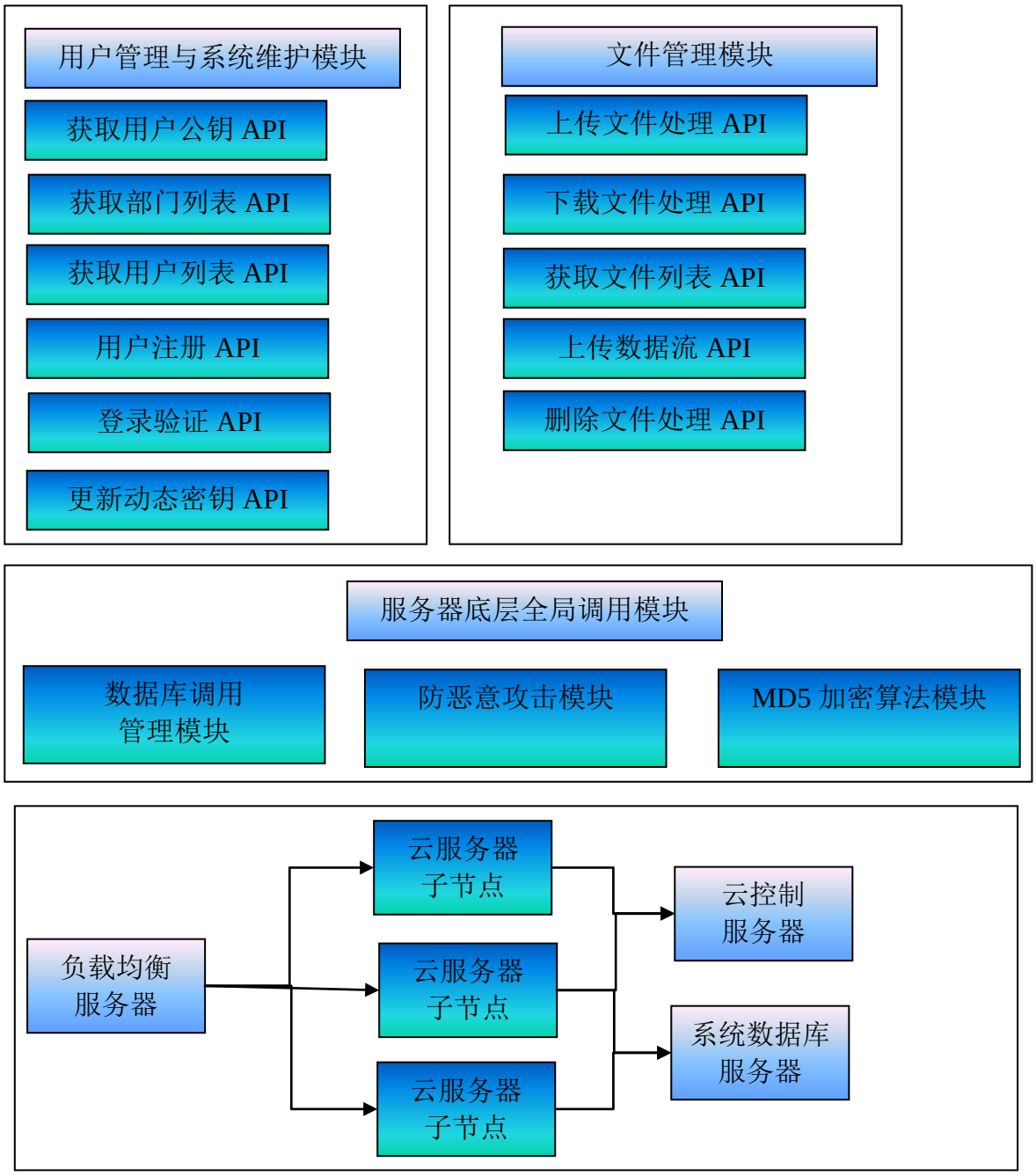


图 2.2

负载均衡服务器：系统服务器架构中位于最前端的服务器，承担着外界客户端与服务器连接的重要任务，不论后台云服务器配置多台，在客户端连接时统一将负载均衡服务器作为唯一接口。

云服务器子节点：当客户端与负载均衡服务器连接后，负载均衡服务器会根据后端云计算子节点的资源使用量，优化分配客户端到相应子节点，实现客户端与服务器的第二次握手。由子节点负责客户端的各项数据交换业务及云计算任务。随着需要承担的文件管理任务量的增加，云服务器子节点可以在现有资源基础上进行扩展，达到“永不饱和”的云存储效果。

云服务器文件系统：机密文件经客户端与云计算子节点处理完成后，加密后的文件数据被存储到云服务器文件系统，该文件系统通过 RAID 阵列技术，实现数据的实时备份，避免数据在服务器丢失。

系统数据服务器：专用于管理服务器上的所有用户信息、文件信息，所有云计算子节点均需在后台连接到该数据服务器完成数据更新与维护。

2.4 系统加密算法设计方案

2.4.1 AES 对称式加密算法介绍

自从 20 世纪 70 年代以来，计算机技术飞速发展。随之而来的是计算机网络通信量大幅增加，以及人们对计算机网络中的信息安全保密要求日益增长。大量的敏感数据与机密信息的传输都需要有密码进行保护。为了保护商业活动中至关重要的敏感信息，美国国家标准局在 1977 年正式颁布了第一个数据加密标准算法(Data Encryption Standard，DES)。这极大的促进了分组密码的研究与发展，并使分组密码第一次广泛应用到商业应用之中。但由于 DES 算法的密钥空间较小，随着计算机硬件计算能力的飞速提高，仅 20 年之后，DES 密钥的穷举破解就已成为可能。于是，寻找一个安全高效的 DES 替代算法就成了密码学界的重要工作。

为了应对这种挑战，1994 年美国颁布了密钥托管加密标准 EES，计划用 EES 取代 DES。EES 密码算法被设计为允许法律监听的保密通信方式。即如果法律部门不进行监听，则加密的密文对其他人来说是不可破译的，但是经过法律部分允许可以进行监听。如此设计的目的在于既要保护正常的商业通信秘密，又要阻

止不法份子利用保密通信进行犯罪活动。而且 EES 只提供加密芯片而不提供密码算法的做法标志着美国密码标准制定政策由公开转征集向秘密设计。EES 颁布以后再美国社会引起了激烈的争论。商业界和学术界对不公布算法只承诺安全的做法表示出了极大的不信任，强烈要求公开算法并取消其中的法律监督部分。迫于社会的压力，美国政府邀请了少数的密码专家介绍该算法以取得公众理解，但收效不大。1995 年美国贝尔实验室的 M.Blaze 博士通过攻击 EES 的法律监督字段成功伪造出一个合法的 ID。于是，EES 被迫退出了美国国家标准的舞台。

在这种背景下，1997 年由原美国国家标准局改组而成的美国国家标准和技术研究所发起了征集替代 DES 算法的高级加密标准(Advanced Encryption Standard，AES)工作。AES 的基本要求是比三重 DES 的加密速度快，安全性不低于三重 DES，分组长度为 128 比特，密钥长度为可变的 128/192/256 比特。

初步的征集活动于 1998 年结束，第一届 AES 候选算法会议宣布了来自全世界的 15 个候选算法。在综合了对安全性和可行性等各种意见后，MARS，RC6，Rijindael，Serpent 和 Twofish，5 个候选算法进入下一轮评估。

Rijindael 的特点:优秀的跨平台性能，快速的密钥设置，存储要求低，适合于资源有限的设计、保守的算子运用有利于进一步分析，算子的选择易于抵抗某些物有利于结构化设计。

Rijindael 优点如下：

1. 跨平台性能好。
2. ROM 和 RAM 要求低，适于 Smart 卡。
3. 运算的设置使得易于抵抗对 Smart 卡的攻击。
4. 快速的密钥设置。
5. 比较好的支持结构化平行设计。
6. 支持其他的 32bit 倍数的密钥长和分组长。

经过五年的甄选流程，高级加密标准由美国国家标准与技术研究院 （NIST）于 2001 年 11 月 26 日发布于 FIPS PUB 197，并在 2002 年 5 月 26 日成为有效的标准。

2.4.2 RSA 非对称式加密算法介绍

RSA 公钥加密算法是 1977 年由 Ron Rivest、Adi Shamirh 和 Len Adleman 在（美国麻省理工学院）开发的。RSA 取名来自开发他们三者的名字。RSA 是目前最有影响力的公钥加密算法，它能够抵抗到目前为止已知的所有密码攻击，已被 ISO 推荐为公钥数据加密标准。RSA 算法基于一个十分简单的数论事实：将两个大素数相乘十分容易，但那时想要对其乘积进行因式分解却极其困难，因此可以将乘积公开作为加密密钥。

RSA 公开密钥密码体制。所谓的公开密钥密码体制就是使用不同的加密密钥与解密密钥，是一种“由已知加密密钥推导出解密密钥在计算上是不可行的”密码体制。在公开密钥密码体制中，加密密钥（即公开密钥）PK 是公开信息，而解密密钥（即秘密密钥）SK 是需要保密的。加密算法 E 和解密算法 D 也都是公开的。虽然秘密密钥 SK 是由公开密钥 PK 决定的，但却不能根据 PK 计算出 SK。正是基于这种理论，1978 年出现了著名的 RSA 算法，它通常是先生成一对 RSA 密钥，其中之一是保密密钥，由用户保存；另一个为公开密钥，可对外公开，甚至可在网络服务器中注册。为提高保密强度，RSA 密钥至少为 500 位长，一般推荐使用 1024 位。这就使加密的计算量很大。为减少计算量，在传送信息时，常采用传统加密方法与公开密钥加密方法相结合的方式，即信息采用改进的 DES 或 AES 对话密钥加密，然后使用 RSA 密钥加密对话密钥和信息摘要。对方收到信息后，用不同的密钥解密并可核对信息摘要。

RSA 算法是第一个能同时用于加密和数字签名的算法，也易于理解和操作。RSA 是被研究得最广泛的公钥算法，从提出到现在的三十多年里，经历了各种攻击的考验，逐渐为人们接受，普遍认为是目前最优秀的公钥方案之一。

本系统正是基于 RSA 的非对称式加密算法对用户信息进行加密，将公钥与通钥存放于云数据库服务器，而由用户 A 通过 TPM-USB 保存私钥。使其他用户 B 在需要报送文件时可直接从数据库调用用户 A 的公钥完成文件加密与数字签名工作。而只有持有私钥的用户 A 才能解密报送的文件，确保了文件报送过程的稳定性与安全性。

2.4.3 MD5 信息摘要加密算法介绍

MD5 即 Message-Digest Algorithm 5（信息-摘要算法 5），用于确保信息传输完整一致。是计算机广泛使用的杂凑算法之一（又译摘要算法、哈希算法），主流编程语言普遍已有 MD5 实现。将数据（如汉字）运算为另一固定长度值，是杂凑算法的基础原理，MD5 的前身有 MD2、MD3 和 MD4。MD5 的作用是让大容量信息在用数字签名软件签署私人密钥前被"压缩"成一种保密的格式（就是把一个任意长度的字节串变换成一定长的十六进制数字串）。

1991 年，Rivest 开发出技术上更为趋近成熟的 MD5 算法。它在 MD4 的基础上增加了"安全-带子"（safety-belts）的概念。虽然 MD5 比 MD4 复杂度大一些，但却更为安全。这个算法很明显的由四个和 MD4 设计有少许不同的步骤组成。在 MD5 算法中，信息-摘要的大小和填充的必要条件与 MD4 完全相同。Den boer 和 Bosselaers 曾发现 MD5 算法中的假冲突（pseudo-collisions），但除此之外就没有其他被发现的加密后结果了。

MD5 的典型应用是对一段 Message(字节串)产生 fingerprint(指纹)，以防止被“篡改”。举个例子，你将一段话写在一个叫 readme.txt 文件中，并对这个 readme.txt 产生一个 MD5 的值并记录在案，然后你可以传播这个文件给别人，别人如果修改了文件中的任何内容，你对这个文件重新计算 MD5 时就会发现（两个 MD5 值不相同）。如果再有一个第三方的认证机构，用 MD5 还可以防止文件作者的“抵赖”，这就是所谓的数字签名应用。大家都知道，地球上任何人都有自己独一无二的指纹，这常常成为公安机关鉴别罪犯身份最值得信赖的方法；与之类似，MD5 就可以为任何文件（不管其大小、格式、数量）产生一个同样独一无二的“数字指纹”，如果任何人对文件做了任何改动，其 MD5 值也就是对应的“数字指纹”都会发生变化。

MD5 还广泛用于操作系统的登陆认证上，如 Unix、各类 BSD 系统登录密码、数字签名等诸多方面。如在 UNIX 系统中用户的密码是以 MD5（或其它类似的算法）经 Hash 运算后存储在文件系统中。当用户登录的时候，系统把用户输入的密码进行 MD5 Hash 运算，然后再去和保存在文件系统中的 MD5 值进行比较，进而确定输入的密码是否正确。通过这样的步骤，系统在并不知道用户密码的明码的情况下就可以确定用户登录系统的合法性。这可以避免用户的密码被具有系

统管理员权限的用户知道。MD5 将任意长度的“字节串”映射为一个 128bit 的大整数，并且是通过该 128bit 反推原始字符串是困难的，换句话说就是，即使你看到源程序和算法描述，也无法将一个 MD5 的值变换回原始的字符串，从数学原理上说，是因为原始的字符串有无穷多个，这有点象不存在反函数的数学函数。

本系统数据库中有关用户登录密码及加密文件表中的密钥数据即是基于 MD5 进行加密保存的，MD5 算法有效避免了用户在登录验证过程中明文密钥暴露，保证了整个验证流程的安全性。

2.5 系统关键技术与创新分析

以下重点分析本系统应用几大核心技术的设计方案与实现过程，首先是非对称式与对称式加密技术的整合。

（1）非对称式与对称式算法联合加密技术

我们都知道在信息安全领域，数据加解密算法可分类以下两类，非对称式算法特点为验证机制复杂，防破解能力高，运算效率低，而对称式加密特点为验证机制简单，密钥不易保管，运算效率高。本系统为了获得最高的安全性及最佳的运算效率，将两种算法结合设计。在云存储集群中的所有文件均采用对称式 AES-256 位加密算法存储，以最佳的运算效率完美完成 G 级大容量文件加密。

对称式加密算法的密钥管理又成为本系统需要解决的下一个问题，为保证安全性，所有文件 AES 密钥均由服务器随机生成，这些密钥在完成调用后将被通过非对称式 RSA 加密算法进行二次加密并被存储到服务器数据库，由于加密对象为长度有限的密钥，所以 RSA 算法可以保证加密效率。

本系统需解决的第三个问题便是非对称式加密算法的密钥管理，我们知道 RSA 算法具有两个验证因子：公钥和私钥，使用公钥可以加密却不能解密，而私钥则相反。本系统正是利用这一特性，将公钥与私钥分开保管来确保数据安全性。RSA 公钥对于用户数据不构成威胁，因此将其直接保存在服务器数据库，任何人（包括用户自己或第三方）任何时候需要加密文件，都可方便的提取以实现加密过程。而相应密钥对的 RSA 私钥属于高危因子，本系统将其设计为离线存储由用户自己保管。因此，即使服务器被攻击或入侵，只要黑客无法得到用户私钥，就可确保数据的绝对安全。

（2）基于 TPM 可信计算架构的 USB 硬件动态密钥验证装置

为确保安全性，RSA 加密算法的密钥对，也是在服务器随机生成的，因此由用户通过传统方法记忆或保管高危私钥是不明智的。本系统因此提出 TPM 可信计算硬件私钥验证装置，将冗长的高危私钥存储在硬件中，保证私钥安全。

本系统需要解决的下一个问题就是，如何确保私钥存储的安全性，首先我提出了开源的思想，在本系统整个硬件验证机制设计初期，就确立了设计电路及内部源程序完全公开的理念。对于信息安全类产品，无法获知内部工作流程是极其可怕的事情，因为其中的设计漏洞或恶意后门存在不确定性。

以下将具体陈述 TPM 硬件验证设计方案，在硬件中体现了三大核心功能：对称式加密，动态密钥更新，熔丝自毁保护。本系统硬件基于 AVR 单片机实现，考虑到硬件资源有限，采用经过嵌入式优化的对称式加密算法，对高危私钥进行加密后存储到硬件内部 ROM 中，因该高危私钥经过对称式算法加密存储，即使本硬件被破解，依然不会对该私钥造成威胁。

在硬件设计中需要解决的第二个问题就是在硬件中调用对称式加密算法的安全性，本系统依靠硬件的动态密钥更新功能来解决。每次调用硬件时，硬件内部将动态生成一条新的密钥，在芯片内部完成对用户高危私钥的解密及使用该新密钥的重新加密。通过动态更新用户加密高危私钥时使用的密钥以确保加密的时效性及安全性。

在硬件设计中需要解决的第三个问题就是每轮动态更新密钥时的密钥保管问题。如何确保旧密钥在执行解密时的可行性及新密钥在执行重新加密时的安全性呢？本系统给出的解决方案是通过云端服务器数据库来实现硬件密钥的保管。每轮动态密钥更新前，系统都将提取服务器数据库中的存储的旧密钥回传给硬件，硬件在芯片内部完成密钥处理工作后，返回新密钥给服务器作更新。通过这样的设计，即使密钥在传输过程中被监听，由于每轮密钥都不同，故可视为安全。

（3）多系统跨平台全硬件 TPM 验证的文件管理

2013 年，物联网与云计算技术正在逐渐普及，我们清晰的发现 IT 业发展进入后 PC 时代，笔记本电脑逐渐取代台式机，平板电脑逐渐取代笔记本，手机逐渐取代平板电脑。信息流的处理渠道更为复杂多样。如何让用户能随时随地的处理信息呢？本系统 TPM 硬件验证模块的跨平台多终端设计就有效解决了该问题。在 WINDOWS 操作系统，本项目基于 LIBUSB 函数库开发了 USB 驱动通信

层，并使用 VISUAL BASIC.NET 语言开发了云计算文件管理客户端。在 LINUX/ANDROID 操作系统，本项目基于 LINUX-OTG 技术开发了 USB 驱动通信层，并使用 PHONEGap 框架开发了云计算文件管理客户端。

用户可以通过 OTG 线将本 TPM 可信计算硬件模块直接接驳到自己的智能手机及平板电脑，也可以通过 USB 接口接驳到普通台式机。由此实现了用户多终端硬件级的文件安全管理。

（4）云计算架构前端防护，永不停机，无限扩容

传统离线存储方式（U 盘、硬盘）容量有限安全性差，传统在线存储方式（B/S，C/S，邮件，网络硬盘）可靠性低安全性差，本系统通过云计算架构的科学布局，可有效解决以上问题。首先是服务器架构的多层布局，最上层为前端服务器，其承担着抗 DDOS 流量攻击，业务交接的功能，前端服务器本身不存储任何用户数据及网站信息，负责业务分发，因此即使被攻击也不会造成任何威胁。第二层为业务处理服务器，前端通过判断业务请求合法性并对请求分类后转交给业务处理服务器处理，业务处理服务器本身不存储任何用户数据，负责业务处理，因此即使被攻击也不会对数据造成威胁。第三层为数据存储服务器，分别布局用户信息数据库及多地用户文件存储节点，用户文件备份存储节点。任何层级监测到攻击行为，都将立即通知管理员并断开与下层的数据传输，确保数据安全。

云计算架构的又一优势在于在线容错，无限扩容。当任何业务处理服务器出现问题时，前端服务器将自动把业务转交到其他服务器处理，该问题服务器自动离线挂起并通知管理员检查，以此实现云端永不停机 7X24 正常运行。当业务处理或数据存储出现瓶颈时，管理员可通过简单配置直接增加相应层级的服务器，确保云端资源的永不饱和，存储空间的无限扩容。

（5）基于非对称式授权的文件共享与分发

由于云端用户加密业务核心基于非对称式算法，因此可满足文件安全共享与定向分发的需求。如果用户 A 需要发送文件给用户 B，系统将从服务器数据库中提取用户 B 的公钥，并使用此公钥加密需要共享分发的文件，存放在用户 B 的云盘中。当用户 B 连接系统后，即可浏览到用户 A 发送的文件，并使用自己的私钥解密该文件后读取。该机制充分解决了加密文件传递困难或密钥暴露的问题，实现了涉密文件的安全授权共享。

3 系统基础硬件实施方案

3.1 TPM-USB 总体功能设计规划

3.1.1 新用户绑定

新用户注册时，正确填写完用户名和密码后，客户端自动生成一组 RSA 非对称私钥对，公钥与通钥随用户注册信息经加密后发送至云端数据库，而私钥将传输到 TPM-USB 内，并在片内进行硬件加密。TPM-USB 成功绑定后，TPM-USB 片内生成一组动态密码，该密码用于加密 TPM-USB 内保存的私钥，动态密码回传到客户端，并经客户端发送至云端数据库。至此，TPM-USB 绑定完成。

3.1.2 用户验证

客户端从云端服务器读取 TPM-USB 动态密码，发送到 TPM-USB 装置，TPM-USB 装置将用此动态密码对片内存储的私钥进行解密并回传到客户端。客户端使用此私钥进行登录验证，验证通过后，客户端向 TPM-USB 发送验证成功标识，TPM-USB 将在片内重新对私钥进行加密，并回传新的动态密码到客户端，客户端将此密码发送到云端数据库，至此用户验证完成。

3.1.3 私钥重置

当用户 TPM-USB 因某些原因丢失或损坏时，需要重新绑定 TPM-USB。用户将新的 TPM-USB 连接到客户端，客户端向云端服务器发送私钥重置请求，此时用户可手动录入在注册时记录的私钥，而不必更换私钥对，录入私钥传输到 TPM-USB 绑定后，TPM-USB 将生成新的动态密码并回传到客户端，客户端将此动态密码发送到云端数据库，完成私钥重置。

3.1.4 熔丝自毁保护

当 TPM-USB 监测到非法操作，如密钥穷举或多次验证失败时，芯片内部的熔丝自毁保护程序将启动，自动清空片内 ROM 中的用户私钥并将自身锁死。防止用户数据的外泄。

3.1.5 创新点与特色

自由开放：因为整个 TPM-USB 的固件程序以及底层驱动都是基于单片机独立开发完成，而未产用市面上现有的 TPM-USB 解决方案，所以可以避免部分商业产品因源程序封闭而造成的后门漏洞风险。

联网验证：不同于市面上 TPM-USB 解决方案仅依靠单机 API 完成验证的传统验证方式，从系统的全局角度出发，设计了联网验证方式。有效避免单机系统验证因嗅探工具或病毒入侵造成的 TPM-USB 被破解的风险。

动态密码：一次一密的验证方式，保证每次调用 TPM-USB 时都能在片内硬件更换动态密码，对最关键的私钥数据重新加密。TPM 动态密钥同步云端服务器，建立由服务器与用户 TPM 模块组成的双因子验证，避免私钥被暴力破解。

成本极低：因为采用被市面上广泛使用的 AVR 8 位单片机进行开发，使得整套 TPM-USB 的量产成本得以控制在 10 元以内。极具市场价值与商业前景。

3.2 TPM-USB 电路原理图设计

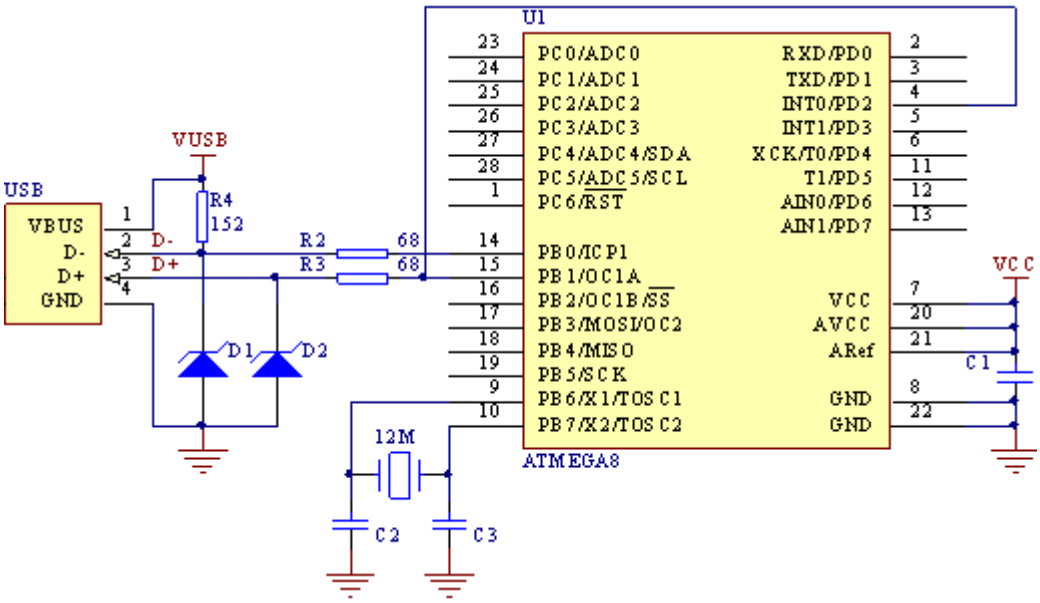


图 3.1

依据 Atmel 公司发布的关于低速 USB 设备的 AVR309 技术文档，使用 Protel 设计了 TPM-USB 的电路原理图，依据 USB 规范定义，数据线 D-上的上拉电阻 R4 用于通知计算机这是一个低速 USB 设备。12MHz 晶体和两个 22p 的电容 C2

和 C3 组成单片机运行所必须的时钟。D+ 和 D- 数据线可使用单片机的任意 IO 端口，但是必须使用相同的 IO 端口。在这里 D+ 连接到 PB1，D- 连接到 PB0。此外数据线 D+还需要连接到 INT0 上，这是为了在不同的 AVR 单片机中使用 AVRUSB 时有更好的适应性和兼容性，无需修改底层核心部分程序的代码。如果 D-连接到端口 D 上（就是和 INT0 同一端口中），同时 D+只连接到 INT0，还可以节省出一个端口来。此方案使用了两只整流二极管来平衡 USB 接口电平。

电阻 R2、R3 起到限流和保护作用，防止在意外情况下损坏计算机的 USB 端口或单片机的端口。单片机所需的电源 Vcc 可由 USB 的 5V 输出电源直接提供，或者由 USB 的 5V 电源转换得到（如 LDO、稳压二极管等），或者通过电池等其他外部电源来供电。

D+和 D-上的 3.6V 稳压二极管 D1 和 D2 起到限制数据线上的电平的作用。因为在 USB 规范中规定数据线 D+和 D-上的电平范围是 3.0V 至 3.6V，而 AVR 单片机的输出电平是 Vcc。如果单片机的 Vcc 是 5V，在没有 D1 和 D2 的情况下将造成电平不匹配，会造成在很多计算机中无法正确识别出 USB 设备。如果用户系统的 Vcc 在 3.0V 至 3.6V 之间，就可以省略这两个稳压二极管。从这里也可以看出用户系统的 Vcc 必须高于 3V。

为了节约成本，充分利用资源，使用以下电路原理图，可在保证原有功能不变的基础上，将成本降低至 6 元左右，更适合量产阶段使用。

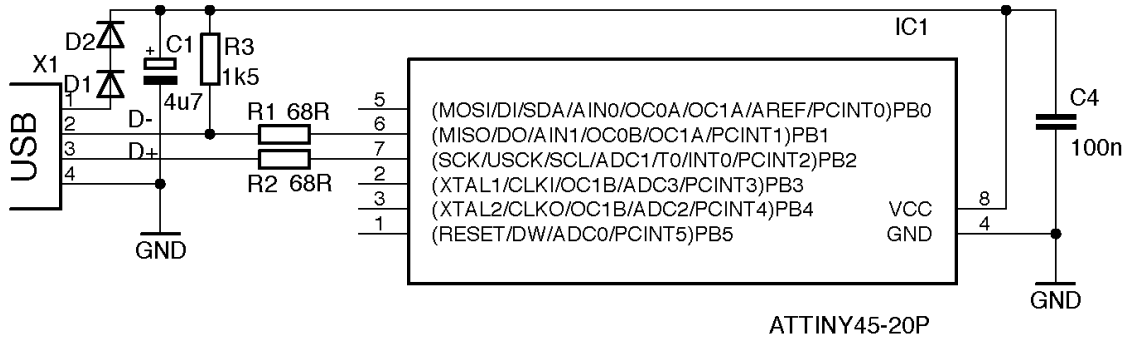


图 3.2

3.3 TPM-USB 固件程序设计

本项目 TPM-USB 源代码的开发，使用 Dev-C++作为编辑器，WinAVR 作为编译器，USBASP 作为烧录器，经过多次反复调试后，成功实现了所有预期功能。以下对部分关键程序进行分析。

3.3.1 USB 设备报告描述符定义

```
PROGMEM char usbHidReportDescriptor[22] = {       /* USB report descriptor */
    0x06, 0x00, 0xff,                  // USAGE_PAGE (Generic Desktop)
    0x09, 0x01,                        // USAGE (Vendor Usage 1)
    0xa1, 0x01,                        // COLLECTION (Application)
    0x15, 0x00,                        //   LOGICAL_MINIMUM (0)
    0x26, 0xff, 0x00,                  //   LOGICAL_MAXIMUM (255)
    0x75, 0x08,                        //   REPORT_SIZE (8)
    0x95, 0x02,                        //   REPORT_COUNT (1)
    0x09, 0x00,                        //   USAGE (Undefined)
    0xb2, 0x02, 0x01,                  //   FEATURE (Data,Var,Abs,Buf)
    0xc0                               // END_COLLECTION
};
```

报告在这里意思是数据传输（data transfer），而报告描述符是对这些传输的数据作用途（usage）上的说明。USB 通讯协议的规范是以 1ms 产生一个 USB 帧（frame），USB 设备可以每一个帧中发送和接收一个交换（transaction）。交换是由几个封包(packet)组成，而传输是由一个或几个交换来完成传送一口中有效的数据。在这里，传输和报告的意思相类似。传输方式有四种，在此我们重点讨论控制型传输(control transfer)和中断型传输(interrupt transfer)。控制型传输是当需要时才执行传输要求，是最一般的传输方式，组态、命令和状态的通讯都可以使用控制型传输；控制型传输主要用于消息型数据（message-type data）。中断型传输目的在做重复的数据更新（recurring data）传输，精确一点而言，即是在每个有限有周期内(bounded period)作至少一次的小量数据发送或接收；所以适用于流动型数据（stream-type data）。

数据本身没有任何意义，要赋予用途才能明确其为控制什么（control）。为了这个目的应运而生报告描述符，其将数据的操控与它的用途作一对一的对应，所以解读报告后就可以知道每个数据作何种操作。所以“传输的数据”和“操作”只是一事件的两种描述方式。用途是以一个 32 位卷标（称作 usage tag）来表示，高 16 位称作 usage page(用途类页)，低 16 位称为 usage ID(用途识别名)。

每种 USB 设备都有一个 PID 和 VID。VID 是生产商的代号，PID 是产品的代号，每个代号都是一个双字节的整数。PID 和 VID 不能随意设置，它是由 USB 标准协会进行分配的。针对这种情况，AVRUSB 提供了 3 个免费的 PID/VID 对，

分别适用于 HID 类、CDC 类和通用类设备，使用 AVRUSB 的用户可以免费使用它们，这样对于大多数应用来说就不用再自己去申请 PID/VID 了。

	VID	PID
控制类	0x16C0	0x05DC
CDC 类	0x16C0	0x05DF
HID 类	0x16C0	0x05E1

表 3.1

3.3.2 TPM-USB 绑定功能流程图

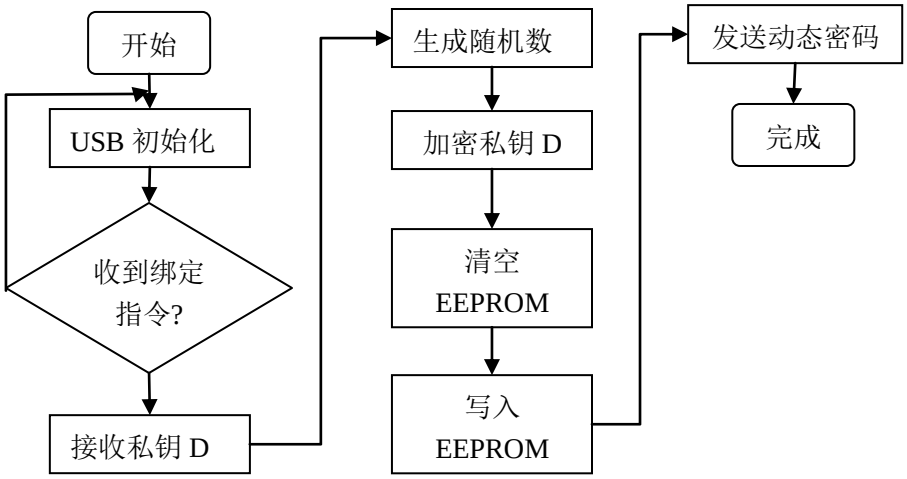


图 3.3

3.3.3 TPM-USB 验证功能流程图

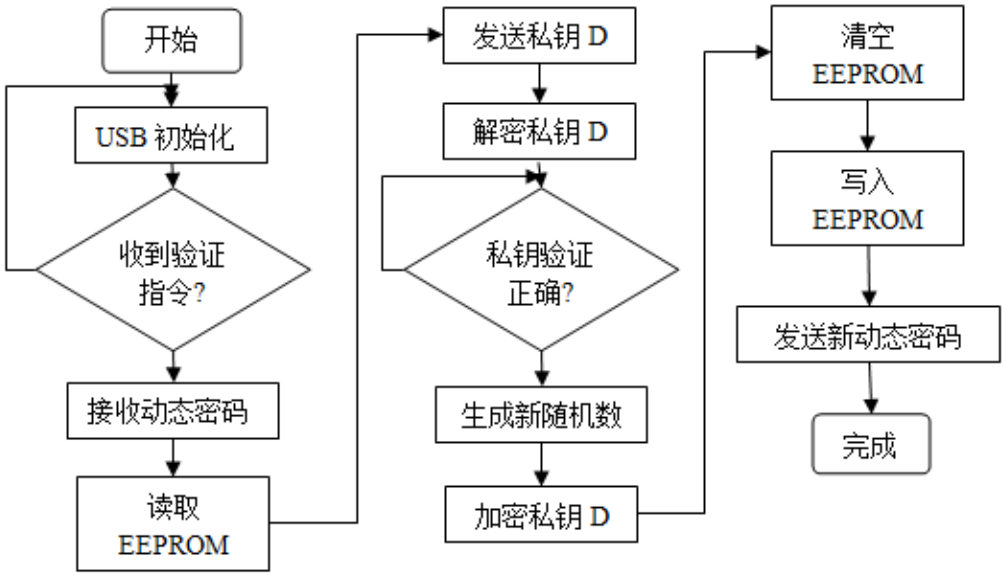


图 3.4

3.3.4 TPM-USB 轮询机制实现

在用户程序的主循环中需要定期调用 USB 事件处理函数 usbPoll()。USB 事件处理函数 usbPoll()在没有 USB 事件需要处理时将直接返回，否则将调用内部函数进行相应的事件处理，最后再将数据通过传递到后面介绍的用户接口函数中。一次 USB 通信的超时时间是 50ms。所以在编程时注意其他事件不要占用太长的时间，使得 usbPoll()函数不能及时执行。

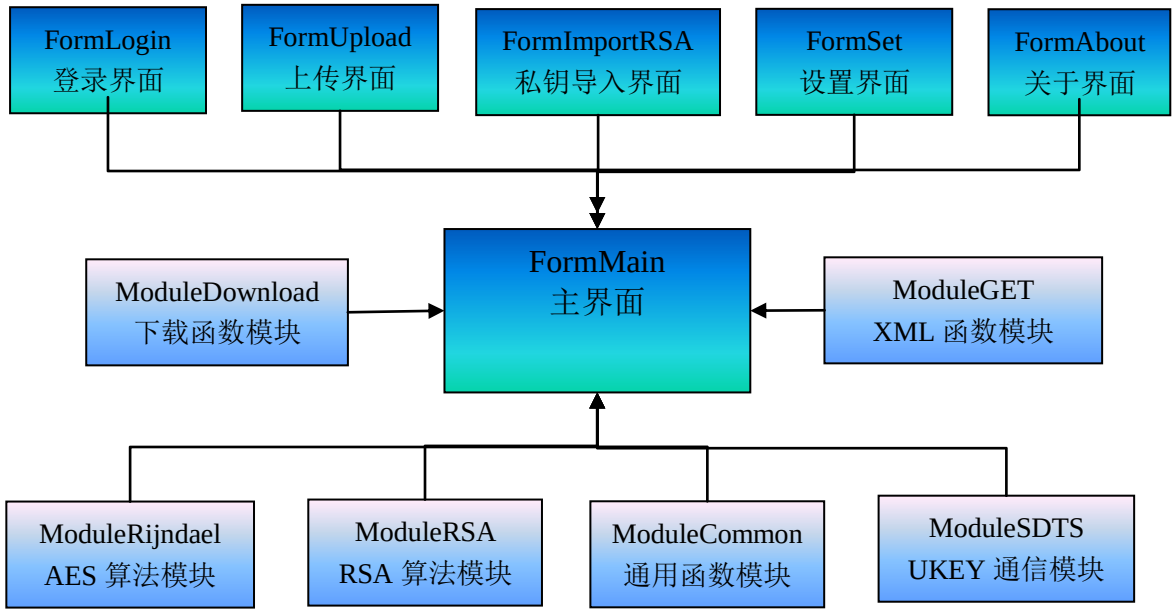
```
USB_PUBLIC void usbPoll(void)
{
schar    len; uchar    i;
len = usbRxLen - 3;
if(len >= 0){
unsigned crc = usbCrc16(buffer + 1, usbRxLen - 3);    /* CRC16 数据完整性验证*/
        usbProcessRx(usbRxBuf + USB_BUFSIZE + 1 - usbInputBufOffset, len);
#if USB_CFG_HAVE_FLOWCONTROL
        if(usbRxLen > 0) usbRxLen = 0;      /*  标记为活动状态*/
#else
        usbRxLen = 0;          /*  标记 RX 缓冲区为活动状态  */
#endif
}
    if(usbTxLen & 0x10){      /*  传输系统  idle */
        if(usbMsgLen != USB_NO_MSG) usbBuildTxBlock();      /*  传输数据等待中? */
    }
    for(i = 20; i > 0; i--){
        uchar usbLineStatus = USBIN & USBMASK;
        if(usbLineStatus != 0)   /* SE0  已经结束  */
            goto isNotReset;}
    /*  重置环境*/
    usbNewDeviceAddr = 0; usbDeviceAddr = 0;   usbResetStall();
    DBG1(0xff, 0, 0);   isNotReset:   usbHandleResetHook(i);}
```

3.3.5 TPM-USB 硬件设计总结

AVUSB 只有在正确设置设备描述符后才能让主机识别，并且要在 MAIN 函数中调用 USBINIT 函数初始化设备，调用 USBPOLL 函数执行轮询，而具体的指令接收与发送，则需要 usbFunctionWrite，usbFnctionRead，usbFunctionSetup 这三个函数来完成数据传输与处理。本项目中由于传输数据量不大，故只采用 usbFunctionWrite 函数。

4 系统客户端设计与实现

4.1.1 客户端总体方案设计



登录界面：获取用户名、密码，验证 TPM-USB，更新 TPMUSB，连接云服务器。

上传界面：获取预上传文件、报送对象，加密上传文件，上传文件到服务器。

私钥导入界面：验证 TPM-USB，重置 TPM-USB，更新 TPM-USB。

设置界面：系统设置，服务器 IP 与端口设置，加密算法设置。

关于界面：系统版本信息、帮助信息。

主界面：透明化文件资源管理器，上传文件，下载文件，删除文件等操作。

4.1.2 登录界面程序设计

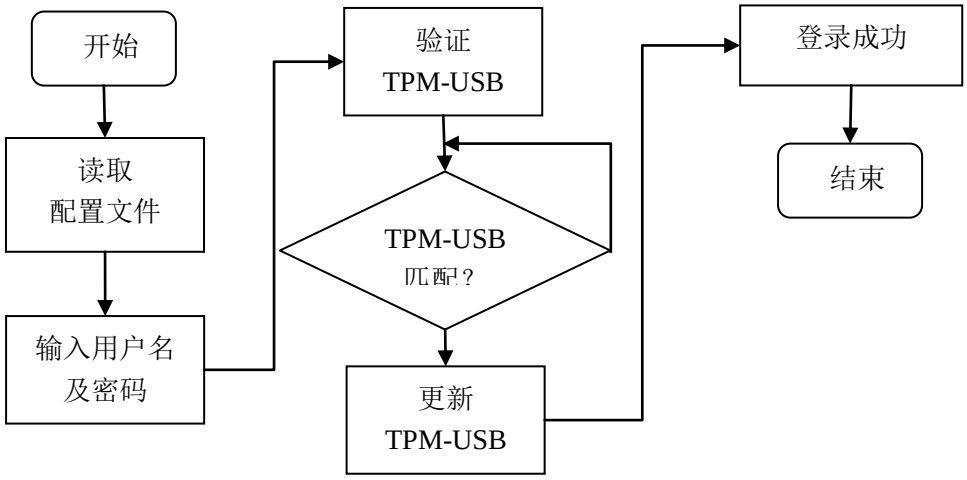


图 4.1

4.1.3 上传文件程序设计

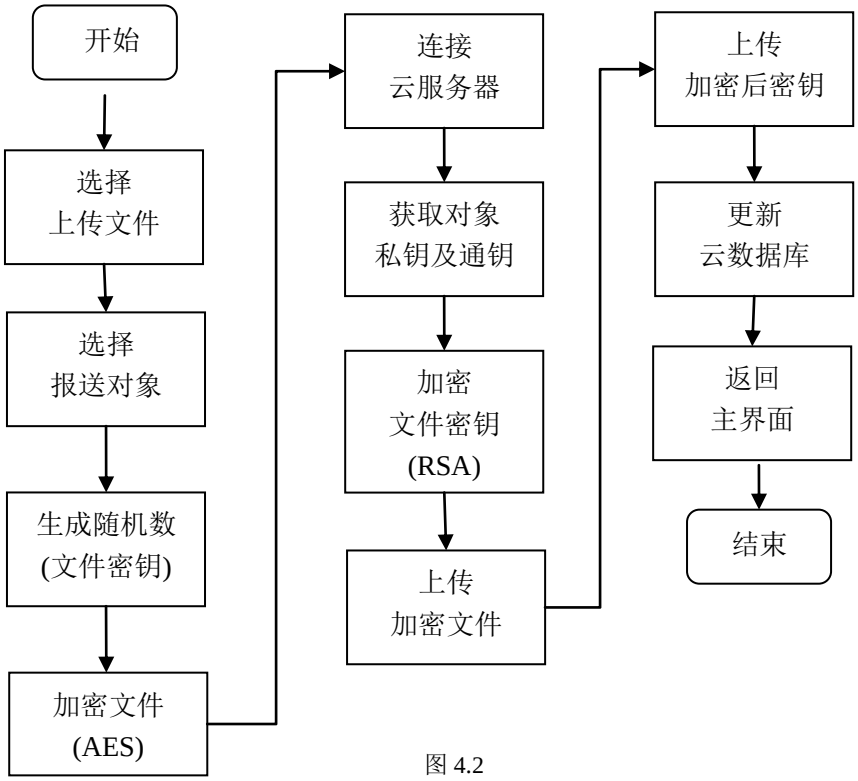


图 4.2

4.1.4 私钥导入程序设计

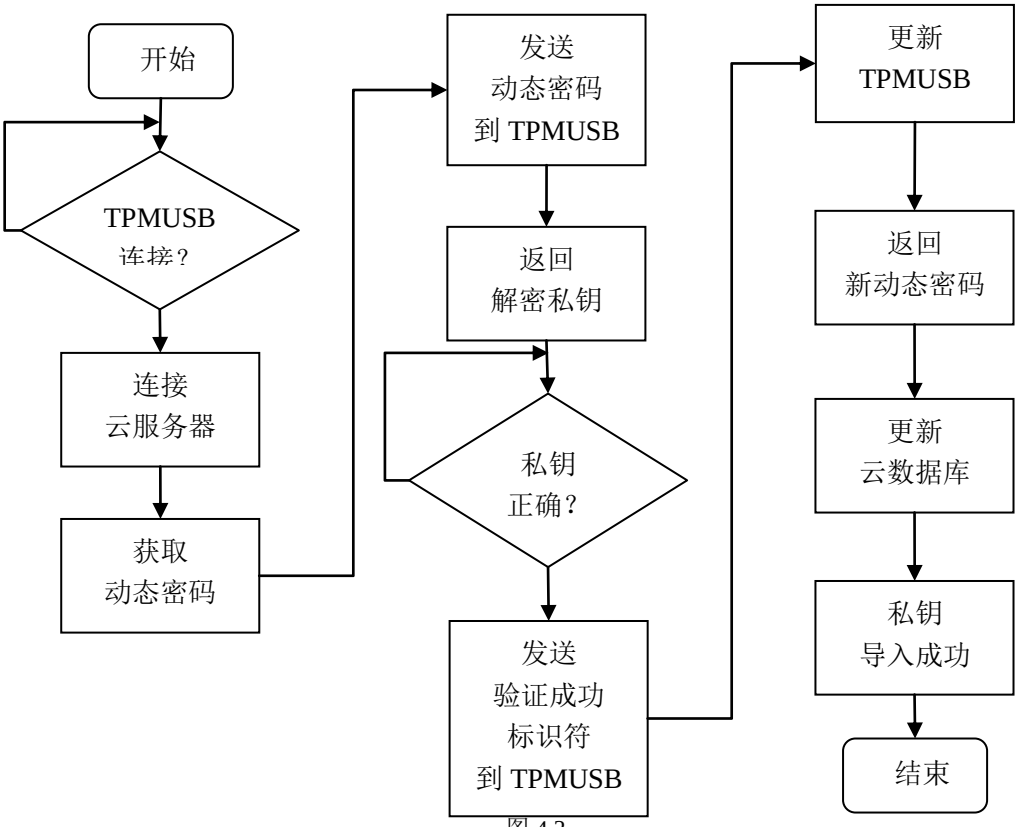


图 4.3

4.2 主界面程序设计

4.2.1 初始化程序设计

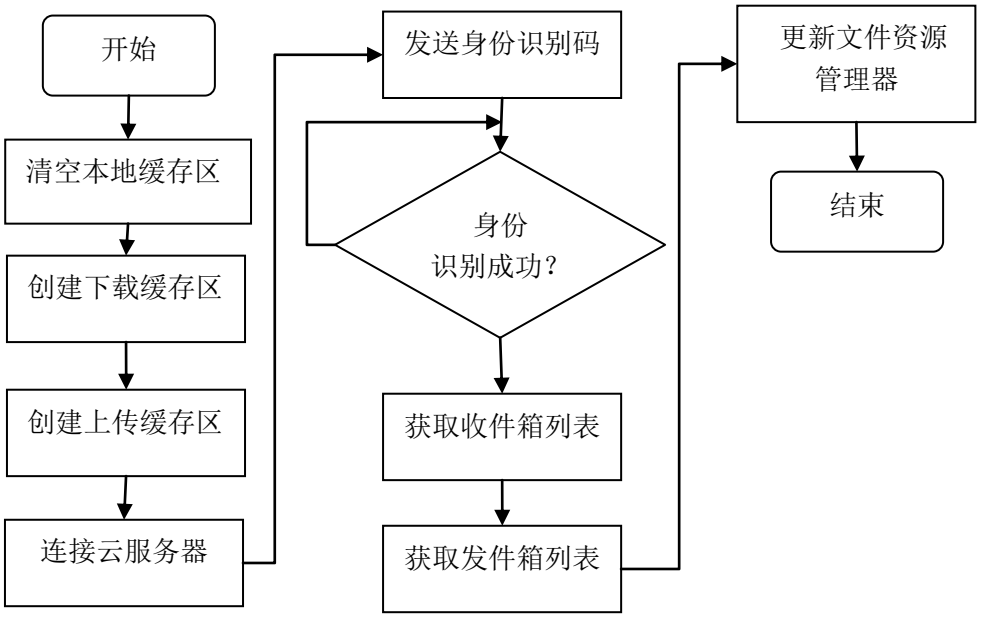


图 4.4

4.2.2 下载文件程序设计

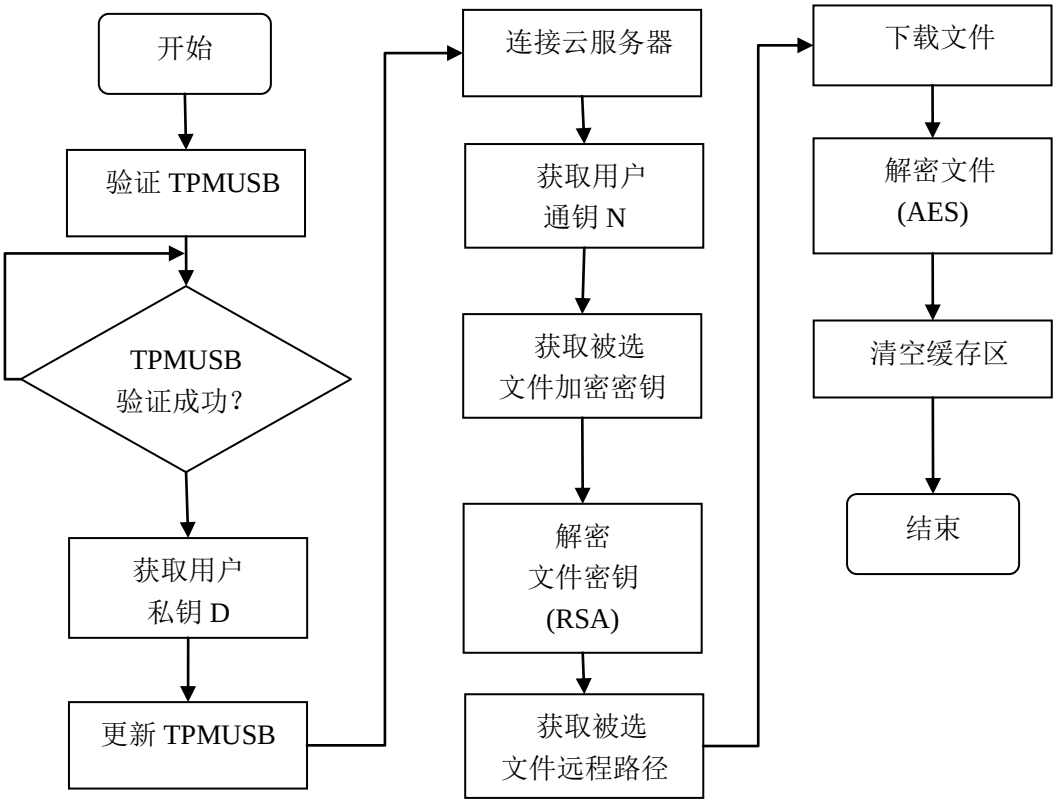


图 4.5

4.2.3 删除文件程序设计

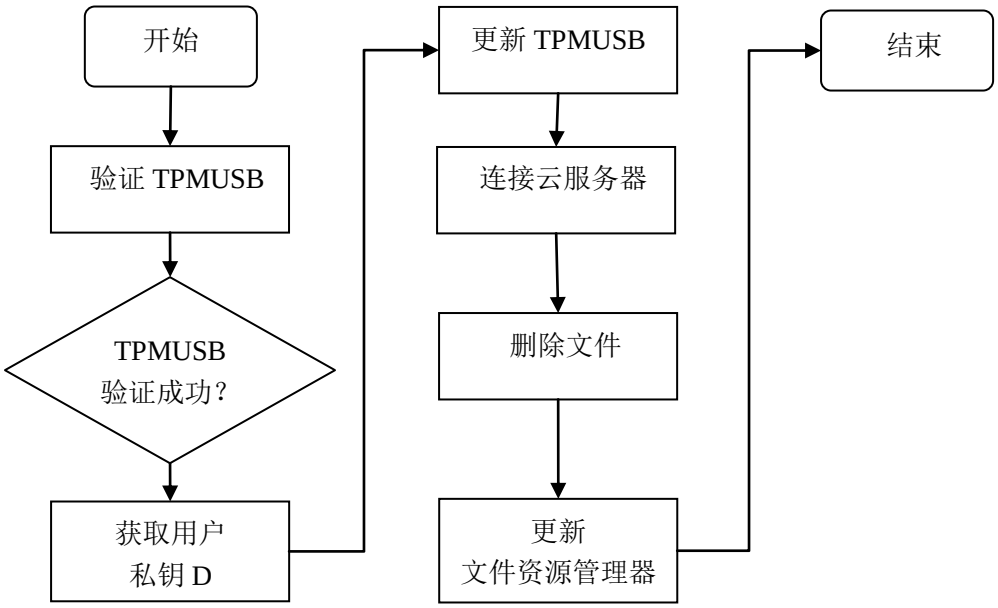


图 4.6

4.2.4 主界面测试版布局设计

图 4.7

4.3 跨平台移动应用设计方案

4.3.1 USB OTG 规范介绍

2001 年底，USB 开发者论坛(USB Implementers Forum，USB IF)发布了专门用于 USB 外设间“可移动互连”的 USB2.0 补充规范 USB On-The-Go，其目的是使外围设备以主机的身份和另外特定的一组外设直接通。OTG 规范是对于 USB2.0 规范的补充，没有指明要修改的任何方面遵守 USB2.0 规范。具有 OTG 特性的所有设备必须是遵守 USB2.0 规范的外围设备。它还要求这些外围设备具备双重角色(DRD dual role device)——既具有主机的功能又具有设备的功能。

在 USB 系统中，无论总线是否空闲，主机向外设提供 5V 电压和 500mA 总线驱动电流。这对于有固定电源的 PC 机来说不是问题，但便携式设备，如数码相机、手机等受电源限制，不能像 PC 机那样为外设供电。所以规定作为主机的双重角色设备只需提供 8mA 的总线驱动电流。为了节省能源，允许主机设备在总线空闲时关闭 VBUS，并提供从机设备启动总线活动的方法。任何一个设备都可以发起会话请求协议 SRP，任何一个设备都可以响应会话请求协议 SRP。该协议要求每个双重角色设备既能发起会话请求协议 SRP，也能响应会话请求协议 SRP。

OTG 规范定义了两种方法：数据线脉冲(data line pulsing)和 VBUS 脉冲(VBUS pulsing)。这两种方法组成了会话请求协议 SRP，实现从机设备对主机设备发出会话请求。从机设备在会话期间 VBUS 应该一直大于主机设备的会话有效阈值。当 VBUS 的值低于设备的阈值时，会话结束。从机设备只有在发现 VBUS 高于它的会话有效阈值且前一次会话结束 2 ms 之后，才能要求新的会话。发现 VBUS 小于它的会话阈值，主机设备认为会话结束。从机设备在请求新的会话前必须确保 VBUS 一直低于这个阈值。在会话起始条件满足后，从机设备可以使用数据线脉冲或 VBUS 发出一个新的会话请求。从机设备通过打开它的数据线上的上拉电阻 5~10ms 来实现前者。通过驱动 VBUS 实现后者。主机设备一直监视着 VBUS，当检测到数据线脉冲或 VBUS 脉冲后就打开 VBUS 开始一次新的会话。

当双重角色设备以主机方式工作时，USB 主机软件包工作主机控制器驱动程序完成 USB 主机软件包与 USB OTG 硬件层的数据交换，USB 总线驱动程序保存设备的信息，主机端设备驱动程序支持目标设备列表里的设备。OTG 提供通用的主机端设备驱动程序。当 OTG 双重角色设备以从机方式工作时，USB 设备软件包工作。设备控制器驱动程序完成 USB 设备软件包与硬件层的数据交换，协议层完成协议规范。USB 设备的功能取决于该双重角色 OTG 设备的功能。

4.3.2 Android 嵌入式操作系统介绍

Android 是一种基于 Linux 的自由及开放源代码的操作系统，主要使用于移动设备，如智能手机和平板电脑，由 Google 公司和开放手机联盟领导及开发。尚未有统一中文名称，中国大陆地区较多人使用“安卓”或“安致”。Android 操作系统最初由 Andy Rubin 开发，主要支持手机。2005 年 8 月由 Google 收购注资。2007 年 11 月，Google 与 84 家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发改良 Android 系统。随后 Google 以 Apache 开源许可证的授权方式，发布了 Android 的源代码。第一部 Android 智能手机发布于 2008 年 10 月。Android 逐渐扩展到平板电脑及其他领域上，如电视、数码相机、游戏机等。2011 年第一季度，Android 在全球的市场份额首次超过塞班系统，跃居全球第一。2012 年 11 月数据显示，Android 占据全球智能手机操作系统市场 76%的份额，中国市场占有率为 90%。

随着安卓手机技术日益成熟，USB OTG 技术正在逐渐成为安卓手机的标准配置。例如平板电脑直接连接到打印机上，通过 OTG 技术，连接两台设备间的 USB 口，将拍出的相片立即打印出来；也可以将平板电脑中的数据，通过 OTG 发送到 USB 接口的移动硬盘上，野外操作就不需携带价格昂贵的存储卡，或者背一个便携电脑。

4.3.3 Android 嵌入式操作系统内核 OTG 原理

通过分析 Android 源代码/android-4.0.4/kernel/omap/drivers/usb/otg/中的文件，我们可以找到 ANDROID 系统中支持的几种 OTG 接收器驱动程序模块，根据驱动程序源代码，可以了解 ANDROID 系统 OTG 功能的实现原理。

编译完成后的 ANDROID 必须获得 ROOT 权限才能使用 OTG 功能，移动应用客户端通过编辑/etc/udev/rules.d 中的规则，可以实现 OTG 与 TPM 的绑定。由于 TPM 模块兼容 HID 通信协议，不论在 PC 平台还是 LINUX 平台皆免驱动。ANDROID 2.3 以后的版本默认包含 LIBUSB 驱动，因此本项目不再需要编写新的驱动程序，即可在 ANDROID 系统与 TPMUSB 模块通信。

4.3.4 Libusb 函数库在 ANDROID 上的移植

由于移动客户端应用在初期研发阶段是在 ARM 开发板上进行调试的，需要手动移植 LIBUSB 驱动，故在此也将 LIBUSB 移植 ANDROID 的方法给出。

首先，我们获取 libusb-1.0 源代码，并将源代码解压到 ANDROID 源代码的 external 目录下，在 libusb 目录中的每个文件夹下创建 Android.mk 文件，顶级目录中的 Android.mk 内容如下：

```
LOCAL_PATH := $(call my-dir)
subdirs := $(addprefix $(LOCAL_PATH)/,$(addsuffix /Android.mk, libusb))
include $(subdirs)
```

LIBUSB 目录中的 Android.mk 内容如下：

```
LOCAL_PATH:= $(call my-dir)
  include $(CLEAR_VARS)
  LOCAL_SRC_FILES:= core.c descriptor.c io.c sync.c linux_usbfs.c
  LOCAL_C_INCLUDES += external/libusb-1.0.9/ \
                      external/libusb-1.0.9/libusb/ \
                      external/libusb-1.0.9/libusb/os
  LOCAL_MODULE:= libusb
  include $(BUILD_SHARED_LIBRARY)
```

接下来使用 ndk-build 编译后，手动挂载驱动模块即可调试 TPM 硬件数据。

4.3.5 移动应用客户端的程序实现

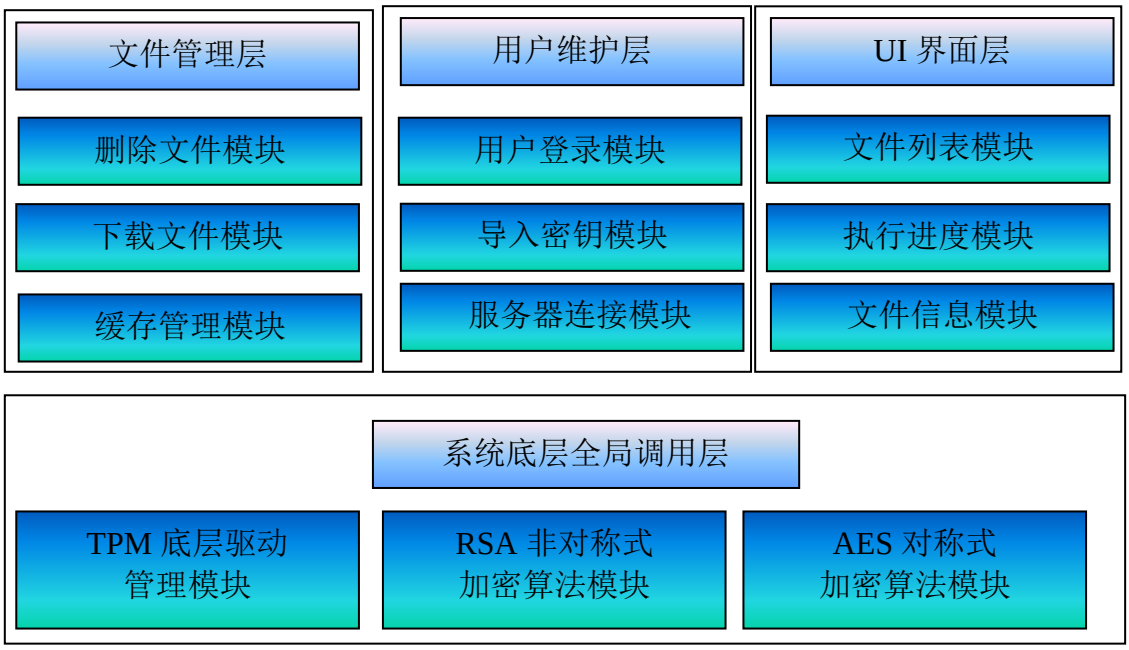


图 4.8

移动应用客户端的表现层与行为层通过 ADOBE 公司的开源移动应用开发框架 PHONEGap 实现，PhoneGap 是一个用基于 HTML，CSS 和 JavaScript 的，创建移动跨平台移动应用程序的快速开发平台。它使开发者能够利用 iPhone、Android、Palm、Symbian、WP7、Bada 和 Blackberry 智能手机的核心功能——包括地理定位，加速器，联系人，声音和振动等，此外 PhoneGap 拥有丰富的插件，可以以此扩展无限的功能。

TPM 数据通信底层通过调用 ANDROID USB Host API 实现，USB HOST API 支持 ANDROID 3.1 及以后的版本，2011 年后上市的手机及平板电脑几乎都可支持该组 API。USB HOST API 位于 android.hardware.usb 中，使用 UsbManager 获取 USB 设备的状态并建立连接，使用 UsbDevice 与硬件进行交流，本项目中的应用与 TPM 的数据通信，主要依靠调用 UsbDeviceConnection 实现。

4.4 客户端设计创新点

TPMUSB 验证机制：不同于普通 TPMUSB 基于固定接口的验证，本系统使用了动态口令反复硬件加密解密用户私钥的方法，实现了真正的“一次一密”。用户将 TPMUSB 插入电脑后，客户端先从云端服务器获取 TPMUSB 的动态密码，

然后使用此动态密码发送到 TPMUSB，TPMUSB 使用此动态密码完成片内私钥的解密，并将私钥返回给客户端，客户端通过云服务器确认私钥正确后，通知 TPMUSB 更新动态密码，TPMUSB 片内随机生成新的动态密码，并对私钥重新加密，返回新的动态密码给客户端，客户端将此动态密码发送到云端数据库，完成一轮 TPMUSB 的验证与动态密码更换。

非对称式加密机制：因为采用 RSA 算法来实现用户数据的加密，所以文件报送者不需要知道报送对象的私钥或 TPMUSB 信息就可直接发送加密文件给对方。而只有拥有对应私钥（TPMUSB）的对象，才能解密该文件。确保了文件传输及存储过程中的高安全性。

透明化：客户端主界面嵌入与普通资源管理器极其相似的文件资源管理器，通过该资源管理器，可直接浏览文件名称，文件类型，文件图标，就像操作本地文件一样轻松，使得用户操作简便易行。

拖拽功能：客户端主界面支持文件拖拽，用户可将欲发送的文件直接拖动到主界面，系统将自动完成文件的校验、加密、上传，提高了用户的办公效率。

4.5 客户端设计总结

本项目的客户端使用 Visual Basic.NET 开发，调用了 CMDialog 与 Windows Common Controls 控件完成界面设计，Explorer ActiveX 控件完成透明化文件资源管理器设计，WIN32 API 中 URLDownloadToFileA 函数完成文件下载程序设计，XMLHTTP 技术完成服务器连接程序设计，开源 USB 函数库 LibUSB 完成 TPM 验证模块与客户端通信程序设计。

经过多次源代码修改与程序调试，目前项目规划中提及的所有功能已全部实现，客户端界面简洁，稳定性高，安全性强，达到预期要求。

5 系统服务器端设计与实现

5.1 云存储技术研究分析

5.1.1 云存储的技术概念

云存储是在云计算概念上延伸和发展出来的一个新的概念，是指通过集群应用、网格技术或分布式文件系统等功能，将网络中大量各种不同类型的存储设备通过应用软件集合起来协同工作，共同对外提供数据存储和业务访问功能的一个系统。云存储是一个以数据存储和管理为核心的云计算系统，是对现有存储方式的一种变革，也就是“存储即服务”。与云计算系统相比，云存储可以认为是配置了大容量存储空间的一个云计算系统。从架构模型来看，云存储系统系统比云计算系统多了一个存储层，同时，在基础管理也多了很多与数据管理和数据安全有关的功能，在两者在访问层和应用接口层则是完全相同的。

5.1.2 云存储的优点

1、具备海量扩展能力

云存储采取的是并行架构，相对于传统的串行架构来说，存储容量分配不受物理硬盘限制，部署新的存储设备即可增加容量。对云存储来说，不论多少存储设备都只看做一台存储设备，整体硬盘容量将尽时，部署新磁盘阵列即可实现存储总容量线性增长。

2、实现负载均衡

传统存储模式下部署多台存储设备时，会出现工作量分配不均的现象，形成存储效能瓶颈。而云存储系统对外提供统一名称，使用这个名称可存取整个存储池的数据，便于应用开发；对内将工作量均匀分配，实现负载均衡，避免单点瓶颈，发挥系统最大效能。

3、便于管理

传统的存储对硬盘的一致性要求近乎苛刻，必须同厂牌、同容量、同型号，否则系统运行中容易出问题。使用云存储则没有这个问题，云存储的设计原理对硬盘一致性没有要求，跨平台的设备都可以一起工作，这样可以保护用户先前硬件的投入。

4、实现故障自动切换及无缝升级

云存储由多台存储设备构成，因此单台存储设备发生故障及硬件需要升级并不会影响整个系统停机，故障时系统会将问题设备上的文件迁移到其它的存储设备上，等新的存储设备上线后，文件再迁移回来，这些操作对于使用者来说都是透明不可见的，实现了各种故障下的零停机。

5.1.3 云存储简易结构图

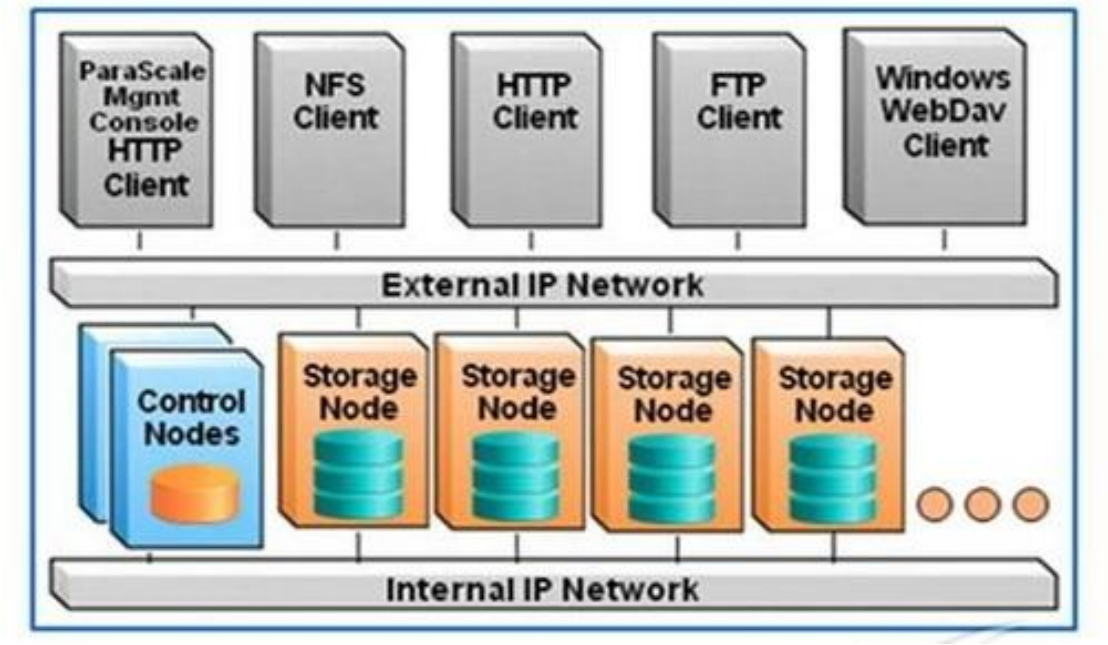


图 5.1

橘色的存储节点(storage node)负责存放文件，蓝色的控制节点(control node)则是作为文件索引，并负责监控存储节点间容量及负载的均衡，这 2 个部分合起来便组成一个云存储。存储节点与控制节点都是单纯的服务器，只是存储节点的硬盘多一些，存储节点服务器不需要具备 RAID 的功能，只要能安装 Linux 即可，控制节点为了保护数据，需要有简单的 RAID level 01 的功能。每个存储节点与控制节点至少有 2 片网卡(千兆、万兆卡都可以，有些也支持 infiniband)，一片网卡 Internal 负责内部存储节点与控制节点的沟通、数据迁移，另一片 external 负责对外应用端的数据读写。

上面灰色的方块(NFS、HTTP、FTP、WebDav)是应用端，左上角的灰色方块(mgmt console)是一台 PC，负责云存储中存储节点的管理。对应用端看来，云

存储只是个文件系统，而且一般来说支持标准的协议，例如 NFS、HTTP、FTP、WebDav 等等，所以很容易把旧有的系统与云存储结合，应用端不需要作改变。

5.1.4 云存储应用条件

云存储不是要取代现有的盘阵，而是为了应付高速成长的数据量与带宽而产生的新形态存储系统，因此云存储在设计时通常会考虑以下三点：

1. 容量、带宽的扩容是否简便

扩容是不能停机，会自动将新的存储节点容量纳入原来的存储池，不需要做繁复的设定。

2. 带宽是否线形增长

使用云存储的客户，很多是考虑未来带宽的增长，因此云存储产品设计的好坏会产生很大的差异，有些十几个节点便达到饱和，这样对未来带宽的扩容就有不利的影响。

3. 管理是否容易

国内有很多用户超过 500 台存储服务器，若不使用云存储来统一管理，管理 500 台存储是一个巨大的工作，一不小心就可能导致某些应用的崩溃，因此云存储的应用是一个必然的趋势。

上面我介绍的是一个纯软件的云存储解决方案，有的产品是硬件的解决方案，他们把橘色的存储节点和蓝色的控制节点，放在一台设备上，这样做的缺点是成本比较高，客户也不能够按照自己的需求，任意选择适合自己规格的硬件，例如读写性能、网卡、硬盘容量等等。

5.2 服务器端网络拓扑结构

本系统服务器端，从以太网接入到负载均衡器，负载均衡器通过判断客户端行为及当前各个云业务节点的资源占用状况，动态分配客户端到相应的云节点，完成业务握手。具体业务可分为文件上传、文件下载、用户数据验证（登录、登出、TPMUSB 验证）等，云业务节点将通过连接下一层的数据库服务器和云存储服务器完成业务处理。经云业务节点集群处理后的数据将继续传输到云存储节点集群，最终保存到云存储集群。

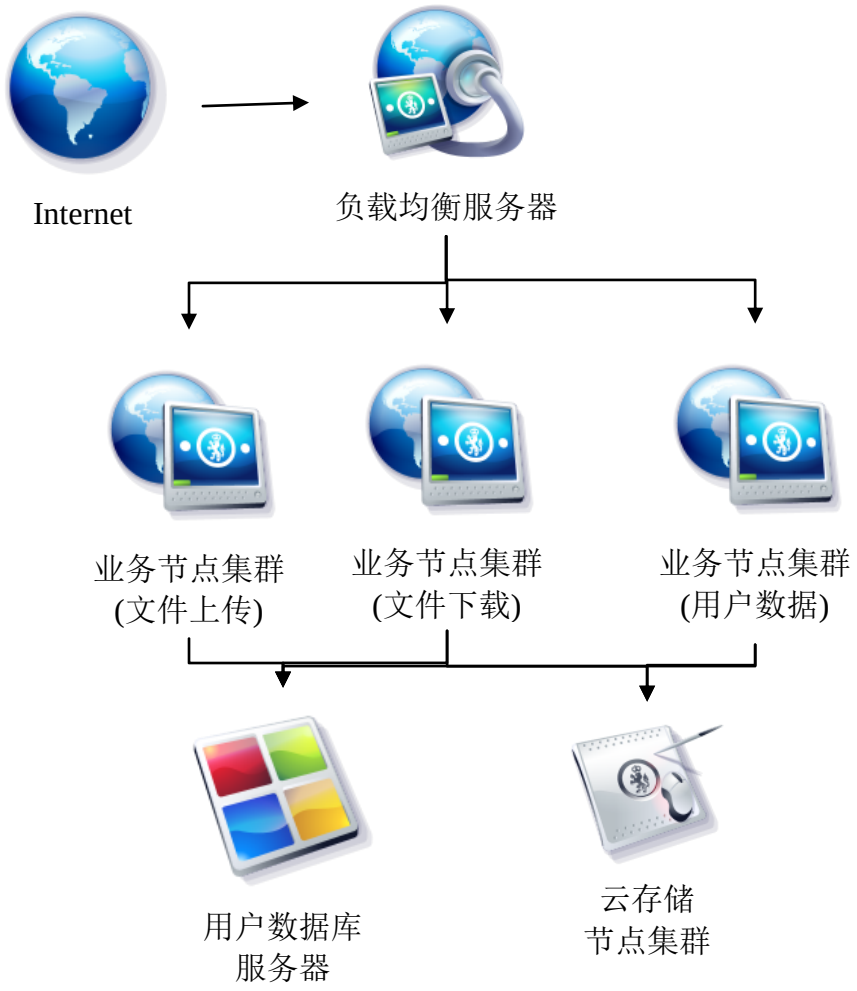


图 5.2

5.3 系统云存储架构模型

与传统的存储设备相比，云存储不仅仅是一个硬件，而是一个网络设备、存储设备、服务器、应用软件、公用访问接口、接入网、和客户端程序等多个部分组成的复杂系统。各部分以存储设备为核心，通过应用软件来对外提供数据存储和业务访问服务。

终端会话层	PC 客户端 Client、WEB 终端、智能手机、移动终端
业务应用层	TPMUSB 验证，上传文件、下载文件、删除文件、发送文件
处理接口层	用户认证接口、文件管理接口、数据更新接口
基础管理层	分布式集群存储管理、并行计算、Session 服务、远程升级
数据存储层	用户数据库、文件虚拟化存储集群、文件备份、NAS 存储管理

5.3.1 数据存储层

存储层是云存储最基础的部分。存储设备可以是 FC 光纤通道存储设备，可以是 NAS 和 iSCSI 等 IP 存储设备，也可以是 SCSI 或 SAS 等 DAS 存储设备。云存储中的存储设备往往数量庞大且分布多不同地域，彼此之间通过广域网、互联网或者 FC 光纤通道网络连接在一起。

存储设备之上是一个统一存储设备管理系统，可以实现存储设备的逻辑虚拟化管理、多链路冗余管理，以及硬件设备的状态监控和故障维护。

5.3.2 基础管理层

基础管理层是云存储最核心的部分，也是云存储中最难以实现的部分。基础管理层通过集群、分布式文件系统和网格计算等技术，实现云存储中多个存储设备之间的协同工作，使多个的存储设备可以对外提供同一种服务，并提供更大更强更好的数据访问性能。数据加密技术保证云存储中的数据不会被未授权的用户所访问，同时，通过各种数据备份和容灾技术和措施可以保证云存储中的数据不会丢失，保证云存储自身的安全和稳定。

5.3.3 处理接口层

处理接口层由多台业务处理服务器节点集群组成，实现对客户端不同业务请求的处理与计算，并为客户端连接提供统一标准 API 接口。

5.3.4 业务应用层

业务应用层是基于客户端得出的定义，主要由客户端同服务器进行数据交换与数据处理的相关源代码组成，由此实现与服务器 API 接口的通信。

5.3.5 终端会话层

终端会话层是整套系统的最上层，是直接与用户沟通，由用户操作执行的层，提供具备较高用户体验的界面及功能触发开关。

5.4 系统数据库服务器设计

本项目设计了基于 Access 与 MSSQL 架构的两种数据库，分别对应小型应用与大型项目应用。在最终测试阶段，使用 MSSQL 数据库，以获得较高的安全性及较快的存储速度，并实现远程数据库连接。

名称	作用	备注
ID	用户组 ID	整型
GroupName	用户组名	字符串

表 5.1 UserGroup(用户组表)

名称	作用	备注
ID	用户 ID	整型
Username	用户名	字符串
Password	用户密码	经 MD5 算法加密
N	用户通钥	字符串
E	用户公钥	字符串
LastKey	TPM 动态密码	字符串
USBFlag	TPM 使用标识符	布尔值
GroupID	用户组 ID	整型
RegTime	注册时间	字符串
ValidityCode	Session 验证码	字符串

表 5.2 User(用户表)

名称	作用	备注
ID	文件 ID	整型
Filename	文件名	字符串
FilePath	文件存储路径	字符串
FileKey	文件密钥	经 RSA 算法加密
SendID	报送者 ID	整型
ToID	接收者 ID	整型
FileSize	文件大小	长整型
FileTime	更新时间	字符串

表 5.3 File（文件数据表）

5.5 系统负载均衡服务器设计

5.5.1 负载均衡概述

由于目前现有网络的各个核心部分访问量和数据流量的快速增长，其处理能力和计算强度也相应地增大，使得单一的服务器设备根本无法承担。在此情况下，如果扔掉现有设备去做大量的硬件升级，这样将造成现有资源的浪费，而且如果再面临下一次业务量的提升时，这又将导致再一次硬件升级的高额成本投入，甚至性能再卓越的设备也不能满足当前业务量增长的需求。

负载均衡（又称为负载分担），英文名称为 Load Balance，其意思就是将负载（工作任务）进行平衡、分摊到多个操作单元上进行执行，例如 Web 服务器、企业应用服务器和其它关键任务服务器等，从而共同完成工作任务。

负载均衡建立在现有网络结构之上，它提供了一种廉价又有效的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性。需要说明的是：负载均衡设备不是基础网络设备，而是一种性能优化设备。对于网络应用而言，并不是一开始就需要负载均衡，当网络应用的访问量不断增长，单个处理单元无法满足负载需求时，网络应用流量将要出现瓶颈时，负载均衡才会起到作用。

5.5.2 四层与七层负载均衡设备

由于 DNS 轮询的缺点，一些对可靠性要求较高的服务器集群，通过采用专用负载均衡设备来实现服务器的负载均衡。

世界上第一个网络体系结构（SNA）由 IBM 公司提出，以后其他公司也相继推出自己的网络体系结构。为了促进计算机网络的发展，国际标准化组织 ISO 于 1977 年成立了一个委员会，在现有网络的基础上，提出了不基于具体机型、操作系统或公司的网络体系结构，成为开放系统互连模型（OSI）。

这个模型把网络通信的工作分为七层。一至四层被认为是低层，这些层与数据移动密切相关。五至七层是高层，包含应用程序级的数据。每一层负责一项具体的工作，然后把数据传送到下一层。由低到高具体分为：物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。

现在的负载均衡技术通常操作于 OSI 网络模型的第四层或低七层。第四层负载均衡将第一个 Internet 上合法注册的 IP 地址映射为多个内部服务器的 IP 地址，对每次 ICP 连接请求动态分配其中的一个内部 IP 地址，达到负载均衡的目的。在第四层交换机中，此种均衡技术得到广泛的应用，一个目标地址是服务器群虚拟 IP 连接请求的数据包流经交换机，交换机根据远端目的 IP 地址、TCP 或 UDP 端口号和负载均衡策略，在服务器 IP 和虚拟 IP 间进行映射，选取服务器集群中空闲的服务器处理连接请求。

第七层负载均衡控制应用层服务的内容，提供一种对访问流量的高层控制方式，适合对 HTTP 服务器集群的应用。

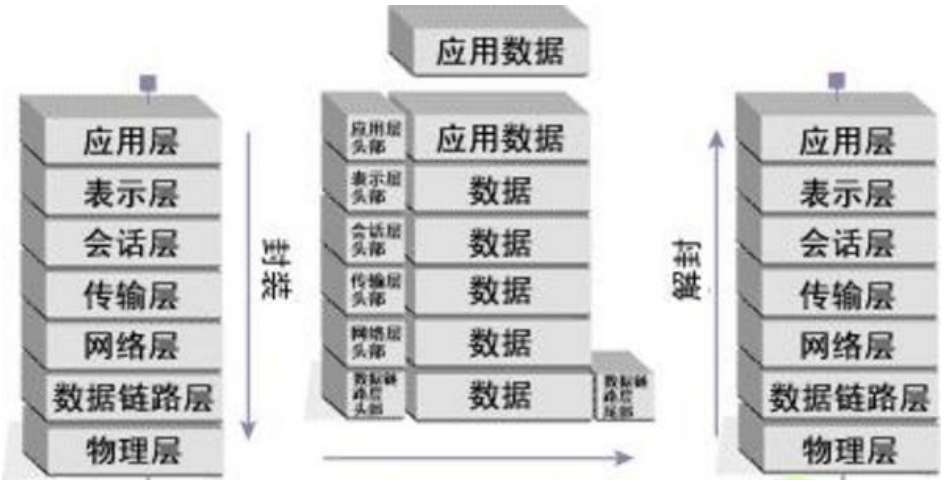


图 5.3

软件四层负载均衡的代表作品为 LVS(Linux Virtual Server)，作者为曾经在国家并行与分布式处理重点实验室工作的章文嵩博士。LVS 是一个开源的软件，可以实现 LINUX 平台下的简单负载均衡。LVS 集群采用 IP 负载均衡技术和基于内容请求的分发技术。调度器具用很好的吞吐率，将请求均衡地分配到不同的服务器上执行，且调度器自动屏蔽掉故障服务器，从而实现服务器集群的高可用性虚拟服务器。整个服务器集群的结构对客户是透明的，而且无需修改客户端和服务器端的程序。

软件七层负载均衡大多基于 HTTP 反向代理方式，代表软件有 Nginx 等，Nginx 的反向代理负载均衡能够很好的支持虚拟主机，可配置性很强，可以按轮询、IP Hash、URL Hash、权重等多种方式对后端服务器做负载均衡，同时还支持后端服务器的故障检查。

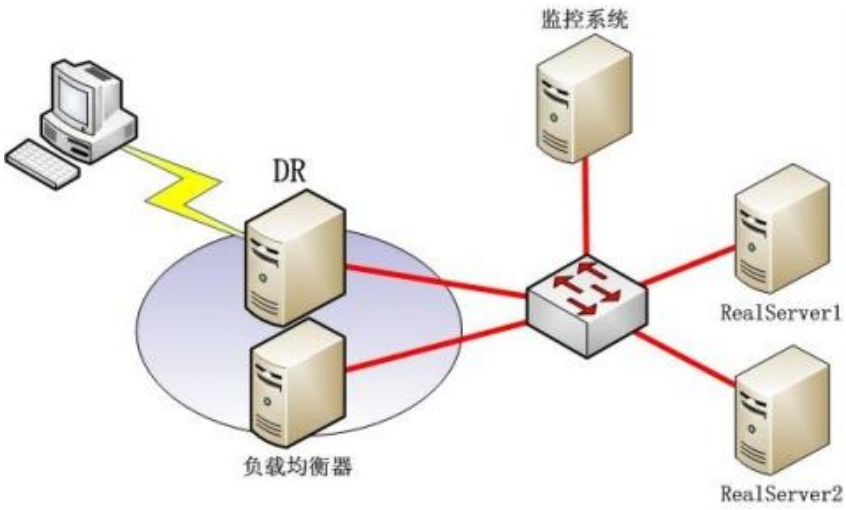


图 5.4

5.5.3 Nginx 负载均衡器的配置

Nginx ("engine x") 是一个高性能的 HTTP 和反向代理服务器，也是一个 IMAP/POP3/SMTP 代理服务器。 Nginx 是由 Igor Sysoev 为俄罗斯访问量第二的 Rambler.ru 站点开发的，它已经在该站点运行超过两年半了。

Nginx 的特点是：

1. 工作在网络的 7 层之上，可以针对 HTTP 应用做一些分流的策略，比如针对域名、目录结构，它的正则规则比 HAProxy 更为强大和灵活，这也是许多朋友喜欢它的原因之一；
2. Nginx 对网络的依赖非常小，理论上能 ping 通就就能进行负载功能；
3. Nginx 安装和配置比较简单，测试起来比较方便；
4. 也可以承担高的负载压力且稳定，一般能支撑超过几万次的并发量；
5. Nginx 可以通过端口检测到服务器内部的故障，比如根据服务器处理网页返回的状态码、超时等等，并且会把返回错误的请求重新提交到另一个节点，不过其中缺点就是不支持 url 来检测；
6. Nginx 仅能支持 http 和 Email，在适用范围上面小很多，这个它的弱势；
7. Nginx 不仅仅是一款优秀的负载均衡器/反向代理软件，它同时也是功能强大的 Web 应用服务器。LNMP 现在也是非常流行的 web 架构，大有和以前最流行的 LAMP 架构分庭抗争之势，在高流量的环境中也有很好的效果。

本项目即是采用 Nginx 实现了云服务器的负载均衡。

Nginx 的编译参数如下：

```
[root@localhost]#./configure --prefix=/usr/local/server/nginx --with-openssl=/usr/include \
--with-pcre=/usr/include/pcre/--with-http_stub_status_module
--without-http_memcached_module\
--without-http_fastcgi_module --without-http_rewrite_module
--without-http_map_module \
--without-http_geo_module --without-http_autoindex_module
```

修改配置文件/usr/local/server/nginx/conf/nginx.conf

```
#运行用户
usernobody nobody;
#启动进程
worker_processes 2;
#全局错误日志及 PID 文件
error_log logs/error.log notice;
pid logs/nginx.pid;
#工作模式及连接数上限
events {    use epoll;    worker_connections1024;}
#设定 http 服务器,利用它的反向代理功能提供负载均衡支持
http {
#设定 mime 类型
include conf/mime.types;
default_typeapplication/octet-stream;
#设定日志格式
log_format main '$remote_addr - $remote_user [$time_local] '
'"$request" $status $bytes_sent '
'"$http_referer" "$http_user_agent" '
'"$gzip_ratio"';
log_format download '$remote_addr - $remote_user [$time_local] '
'"$request" $status $bytes_sent '
'"$http_referer" "$http_user_agent" '
'"$http_range" "$sent_http_content_range"';
#设定请求缓冲
client_header_buffer_size1k;
large_client_header_buffers4 4k;
#开启 gzip 模块
gzip on;
gzip_min_length1100;
gzip_buffers 4 8k;
gzip_types text/plain;
output_buffers 1 32k;
postpone_output1460;
```

```
#设定 access log
access_loglogs/access.logmain;
client_header_timeout 3m;
client_body_timeout 3m;
send_timeout 3m;
sendfileon;
tcp_nopush on;
tcp_nodelay on;
keepalive_timeout 60;
#设定负载均衡的服务器列表
upstream mysvr {
#weigth 参数表示权值,权值越高被分配到的几率越大
#本机上的 Squid 开启 3128 端口
server 192.168.8.1:3128 weight=5;
server 192.168.8.2:80 weight=1;
server 192.168.8.3:80 weight=6;}
#设定虚拟主机
server {
listen80;
server_name 192.168.8.1;
charset gb2312;
#设定本虚拟主机的访问日志
access_log logs/log.access.logmain;
#对 "/" 启用负载均衡
location / {
proxy_passhttp://mysvr;
proxy_redirectoff;
proxy_set_headerHost $host;
proxy_set_headerX-Real-IP $remote_addr;
proxy_set_headerX-Forwarded-For $proxy_add_x_forwarded_for;
client_max_body_size 10m;
client_body_buffer_size 128k;
proxy_connect_timeout 90;
proxy_send_timeout 90;
proxy_read_timeout 90;
proxy_buffer_size 4k;
proxy_buffers 4 32k;
proxy_busy_buffers_size    64k;
proxy_temp_file_write_size    64k;
}
```

5.6 系统业务接口服务器设计

本项目业务处理服务器接口使用微软公司的 ASP.NET 脚本语言开发，使用数据库服务器存储 Session 验证符的方式，保证在客户端会话过程中临时切换业务接口服务器造成的验证信息丢失。具体功能模块分为：数据库调用模块、MD5 加密算法模块、防注入攻击模块、二进制上传类模块，文件管理 API（上传、下载、删除），获取文件列表 API，获取用户分组 API，获取用户 RSA 公钥 API，获取用户列表 API，更新动态密码，用户登录 API，用户注册 API。

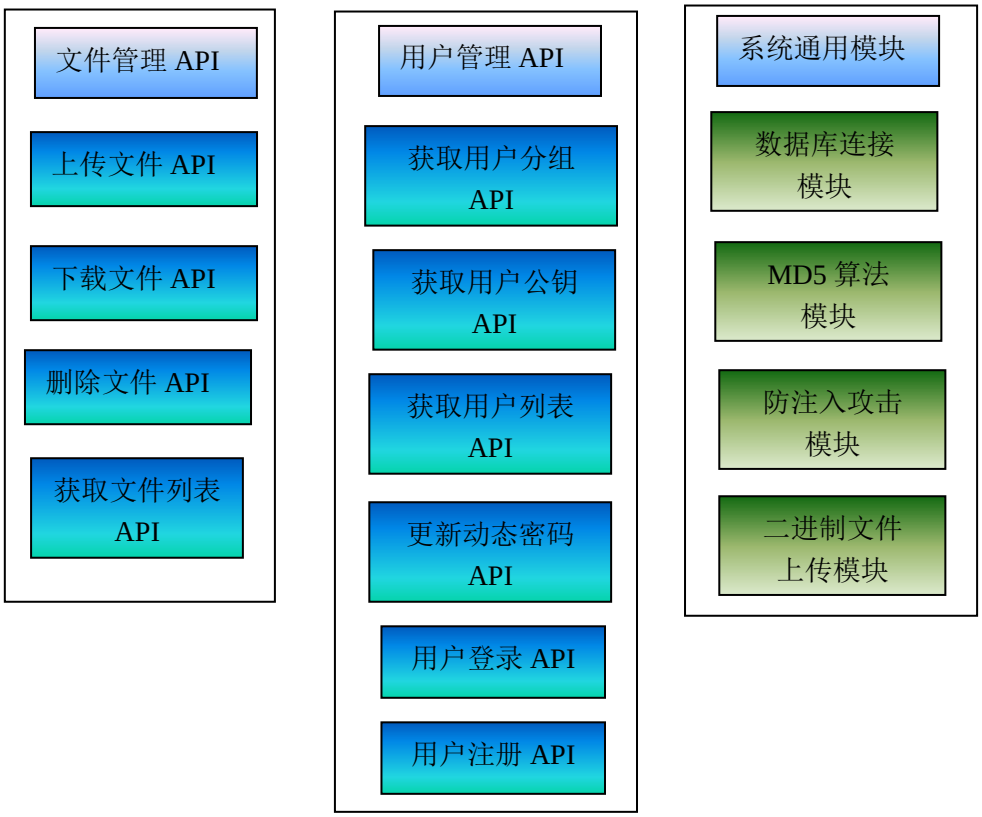


图 5.5

本项目在 WINDOWS SERVER 2008 系统的服务器上安装 IIS 信息服务，并配置了有关 ASP.NET 脚本的解释器，实现了业务接口各个子节点的无缝切换。

上传文件 SQL：

```
sql="insert into File(Filename,FilePath,FileKey,SendID,ToID,FileSize,FileTime) values('"&Filename&"','"&FilePath&"','"&FileKey&"',"&UserID&","&ToID&",'"&FileSize&"','"&FileTime&"')"
```

注册用户 SQL：

```
sql="insert into user (Username,Password,N,E,LastKey,USBFlag,GroupID,RegTime) values('"&UserName&"','"&Password&"',"&N&","&E&","&LastKey&","&USBFlag&","&GroupID&",'"&RegTime&"')"
```

获取用户公钥信息：

```
sql="select N,E from user where ID=" & ToID
rs.open sql,conn,1,1
response.write "RSAInfo"
if rs.recordcount<>0  then
        response.write "N(" & rs("N") & ")"
        response.write "E(" & rs("E") & ")"
        rs.MoveNext
end if
response.write "EndRSAInfo"
```

更新动态密码 SQL：

```
sql="update User set LastKey='"&LastKey&"' where ID=" & UserID
```

5.7 系统文件存储服务器设计

5.7.1 NAS 网络存储技术概述

网络存储技术（Network Storage Technologies） 网络存储技术是基于数据存储的一种通用网络术语。网络存储结构大致分为三种：直连式存储（DAS：Direct Attached Storage）、网络连接式存储（NAS：Network Attached Storage）和存储网络（SAN：Storage Area Network）。

NAS 是一种采用直接与网络介质相连的特殊设备实现数据存储的机制。由于这些设备都分配有 IP 地址，所以客户机通过充当数据网关的服务器可以对其进行存取访问，甚至在某些情况下，不需要任何中间介质客户机也可以直接访问这些设备。

5.7.2 NAS 网络存储器功能

文件共享(即文件服务器)是 NAS 最基本的应用。我们可以在“网上邻居”中找到 NAS 网络存储器设备，并在它的共享目录中存储公用文件。此外，部分 NAS 也内置了文件服务器功能，我们可以通过浏览器访问和管理 NAS 中的文件，并以 HTTP 方式上传和下载文件，就像访问软件下载网站一样方便。

NAS 网络存储器的另一项重要功能是备份/容灾。大多数 NAS 都具有多种备份功能，包括本地备份(将电脑上的数据通过局域网备份到 NAS 中)、异地备份(将异地电脑上的数据通过广域网备份到 NAS 中)和 NAS 间备份(NAS 与 NAS 之间复制数据)等等。部分 NAS 还具有一键备份功能，将 USB 存储设备(如闪盘和外置硬盘)插入 NAS 上特定 USB 接口，按一下备份按钮就能把 USB 存储设备上的文件备份到 NAS 中。此外，具有两个硬盘位的 NAS 可以组建 RAID 0 和 RAID 1 系统，其中 RAID 0 系统具有较好的磁盘性能，RAID 1 系统具有较好的安全性。具有 4 个硬盘位的 NAS 则可以组建更高级的 RAID 5 系统，在保障数据安全的同时还能提高磁盘性能。

5.7.3 NAS 网络存储器配置

网络附加存储（NAS）和 iSCSI 存储已成为虚拟存储的解决方案。相比传统的 FC 光纤通道 SAN 存储，iSCSI 具有成本低廉使用方便的优势但它却面临着其他同样的问题。NAS 具有同样的成本优势和使用简便的特点，但固有的可扩展性和共享功能定位的 NAS 无疑会成为云存储平台的首选。当前，数据中心被大幅引入虚拟化并采纳能显著提高 NAS 效率实现云存储的云模型。

衡量以上考虑后，本系统选用 NAS 存储设备，实验室环境为双节点，在实际运行项目时，可视使用情况在存储集群中增加更多的 NAS 存储服务器节点。

5.8 云存储服务器研究总结

云存储服务器的架构研究及设计验证是本项目相对耗时较长的开发过程，需要解决的几个问题，如：负载均衡配置、业务节点 API 开发、Session 动态交接，数据库规划、NAS 存储系统搭建。其中业务节点 API 是与客户端联合调试进行共同开发的，而云服务器架构搭建与测试则是在实验室使用数台服务器完成的。

6　系统应用测试及安全性析

6.1 系统应用实例

6.1.1　应用环境

本项目在学院科技创新基地完成了实验室测试工作，并于 2012 年 6 月在校网络中心机房开始中试构建，同时为校学生会开通了安全云盘测试帐号，并发连接数峰值达到数百人，目前系统已稳定运行数百天，无宕机故障发生。经过校学生会内部测试反馈的结果表明，系统各项功能达到预期要求，简易性与安全性均满足其日常办公需要。

2012 年首都大学生科技创新作品与专利成果博览会中，本项目获得推荐向相关领域专家进行现场展示，获得了诸多好评，并且同时申请了国家发明专利。在 2013 年本项目经过测评，得以在北京市政府某重点政务服务项目中进行试点推广，项目稳定性、安全性及易用性均获得好评。

6.1.2　用户注册

图 6.1

新用户注册只需填写用户名及密码并选择所属部门，插入 TPMUSB，点击提交后，客户端将自动完成 TPMUSB 的绑定与云端服务器注册工作。

6.1.3 用户登录

图 6.2

用户登录时只需输入用户名及密码即可完成登录，TPM 模块可在登录后再绑定。

6.1.4 文件浏览

图 6.3

用户可通过文件资源管理器上部的 TAB 控件，选择查看自己收到的报送文件（收件箱）或是自己发送给别人的文件（发件箱）。当用户点击“下载文件”或“删除文件”后可直接通过界面下部的进度条观察到下载进度。

6.1.5 发送文件

图 6.4

当用户在主界面点击“发送文件”后，系统会通过此界面提示用户选择欲发送的文件和发送对象。选择完成后，点击“发送”即可完成文件报送。

6.1.6 导入密钥

图 6.5

用户登录系统后，可点击“导入密钥”，插入 TPM 模块后，点击“导入”即可完成用户私钥的导入。若需要重置 TPM 模块或绑定新的 TPM 模块，只需将新的 TPM 模块插入电脑并点击“绑定”即可。

6.2 设计性能要求

6.2.1 性能指标

1. TPM-USB 验证准确率大于 99.99%，使用寿命大于 3 年。
2. 存储容量、处理能力可根据业务的要求平滑扩展。
3. 单节点并发连接数达 5000 个以上 IIS 并发连接。
4. 数据查询处理时间平均小于 5 ms，最大小于 20 ms；
5. 数据更新处理时间平均小于 10 ms，最大小于 30 ms。
6. 各个硬件服务器的 CPU 忙时利用率平均不超过 80%。
7. 负载均衡在切换过程中保证处理中业务正常执行。
8. 支持数据的分布式存储、异地备份和高速访问。

6.2.2 可靠性要求

1. 排除人为误操作因素，由于系统自身原因导致的系统崩溃故障，平均无故障时
2. 间（MTBF）应不小于 26280 小时（3 年），系统平均无故障率不低于 99.999%。
3. 系统应具备电信级可靠性、多种冗余、备份和集群处理的机制和功能，重要部
4. 件、数据库等采用双备份配置，具备冗余和负载分担机制，保证系统无单一故
5. 障点。主要模块冗余度为 1+1，易于扩容和维护。
6. 系统处理能力应采用相应的流量控制措施，满足对处理时延、CPU 使用率的要
7. 求，保证系统的稳定运行。
8. 系统应具有安全防御能力，防止非法入侵，保证系统的快速恢复性，并采用冗余技术（冗余设备、冗余通信链路、RAID 技术）保证数据可靠存储、网络系统可靠运行。

6.2.3 扩展性要求

1. TPM 模块支持二次使用开发，提供公共调用接口 API。
2. 云计算服务器集群可根据业务需要，增加业务处理节点，云存储节点。
3. 数据库设计便于二次开发及数据维护。

6.2.4 故障处理要求

1. 系统应采用多级负荷控制机制，实现负荷分担和节点级备份。
2. 系统应有良好的备份和恢复的日常工作计划，系统数据和业务数据可联机备份、联机恢复，恢复的数据应保持其完整性和一致性。
3. 系统出现故障后，应能够快速切换到其他同级别服务器（包括人工或自动切换），保证系统连续工作。
4. 系统应具备自动或手动恢复措施，在系统使用过程中，由于硬件出现故障或其它原因造成系统暂时性的中断后系统重新启动时，能够保证系统将原有的数据快速恢复。
5. 系统发生故障时，应该能够尽快维护和恢复。系统恢复时间应小于 30 分钟。

6.3 系统性能测试

6.3.1 系统测试环境

系统实验室测试工作在基础教育学院科技创新基地开展，使用了六台服务器，一台交换机完成本次测试工作。具体配置如下：

序号	名称	配置	数量
1	负载均衡器	Core2/4G RAM/SAS 15K 300G*2/RAID1/千兆网口*2/Linux 2.6/Nginx 1.0	1
2	业务子节点	Core2/4G RAM/SAS 15K 300G*2/RAID1/千兆网口*2/WINDOWS SERVER 2008/IIS	2
3	用户数据库	Core2/4G RAM/SAS 15K 300G*2/RAID1/千兆网口*2/WINDOWS SERVER 2008/MSSQL	1
4	云存储节点	Core2/4G RAM/iSCSI 2T*2/RAID1/千兆网口*2/WINDOWS SERVER 2008/NAS	2
5	交换机	24 千兆电口/4 光纤口	1
6	监控平台	个人笔记本电脑	1

表 6.1

6.3.2 文件加解密速度测试

系统加密速度是衡量文件加密系统性能优劣的一个重要指标。为了能够准确的测试加密速度，首先要精确记录下完成一段操作所用的时间差，在 Visual Basic 中，调用 GetTickCount()函数可以得到从系统启动到当前所用过的时间，单位为毫秒，只需分别记录完成每一段操作的时间差，即可求出加密速度值。本系统中的加密速度测试主要为文件的加密解密速度测试。

文件的加密、解密过程包括读取文件数据，获得、计算、写入相关内容，加密、解密文件数据，写入文件数据等所有操作。所有加密解密文件的速度相对于内存数据处理要慢一些。因为加密处理和解密处理的操作步骤有所区别，所以分别记录下这两个操作的测量结果。测试系统对文件的加密解密速度时，从用户选择目标文件并开始操作时记录下时间点 Time_Begin，直到将源文件所有数据操作完成并写入到缓存文件后记录下操作完成时间 Time_End。从而得到处理文件所用的时间，再根据测试文件数据的兆字节数 File_Size，通过公式

$$V=(File\_Size*8)/[(Time\_End-Time\_Begin)*10^{-3}] \quad Mbps$$

得出加密解密速度值。下表给出了系统对文件加密解密的速度测试记录。

文件大小(b)	分块(b)	加解密速度(Mbps)				平均速度(Mbps)
1073741824 (1G)	128	加密	24.20	25.30	21.60	23.70
		解密	25.50	26.90	27.30	26.50
	256	加密	24.60	26.20	28.30	26.30
		解密	19.90	18.90	17.80	18.80
1048576(1M)	128	加密	28.20	25.60	26.20	26.60
		解密	23.50	25.60	23.50	24.20
	256	加密	25.60	28.20	28.20	27.30
		解密	23.50	25.60	28.20	25.70

表 6.2

由表可知，加密与解密速度都比较平滑，稳定性强。数据分块较大时，加密速度有所提升。文件大小较大时，解密速度相对较慢。由此可得出使用 AES 对称式加密算法对文件进行加密的平均加密速度为 25.90 Mbps，平均解密速度为 23.80Mbps，达到计划性能参数的要求。

6.3.3 云服务器抗攻击测试

在排除各个服务器配置存在漏洞的前提下，可以认为服务器及业务处理是相对安全的，而在生产过程中，服务器容易遭受的攻击多以 DDOS 攻击最为常见。DDOS 攻击原理是通过使网络过载来干扰甚至阻断正常的网络通讯。通过向服务器提交大量请求，使服务器超负荷。阻断某一用户访问服务器阻断某服务与特定系统或个人的通讯。

所以，本项目分别采用传统 LINUX+APACHE 平台服务器，及本系统云存储结构服务器集群进行 DDOS 抗攻击对比测试，以此确定其所能承受的最大并发连接数。攻击工具采用 10 台普通 PC 机同时对攻击对象发起 DDOS 洪水攻击。

测试对象	负载并发连接数(request/s)				平均值(request/s)
LINUX+APACHE 服务器	780	799	760	721	765
云存储结构服务器集群	2000	1850	1780	1860	1872

表 6.3

经过对比，可发现采用云存储结构抗攻击性能明显，并且随着业务处理服务器集群的增加，可实现不中断服务的无缝扩容。云存储结构服务集群在两台业务处理服务器节点联合均衡的情况下，达到了 1872 请求数/秒，相当于每天可处理 161740800 个并发请求，远远高于系统设计指标的要求。

6.3.4 文件上传下载速度测试

由于实验环境限制，本系统未接入到以太网进行测试，而选择了相对应用范围较为常见的局域网进行测试。通过将本系统接入到校园网进行测试，文件传输速度均达到了校园网最高限制 10Mbps，所以本项目选择了通过组建小范围局域网的方式来测试本系统所能提供的最大文件传输速率。为了便于比较差异，在相同的硬件结构平台上分别搭建了普通的 FTP 文件管理服务及机密信息报送系统的业务处理服务架构。

测试对象	队列深度	传输速度(Mbps)			平均速度
普通 FTP 服务架构	64	58.80	60.50	59.20	59.50
	128	56.50	52.60	50.40	53.10
	256	25.10	38.90	26.50	30.10
	512	-	-	-	-
业务处理节点集群	64	75.20	76.80	72.50	74.80
	128	76.30	73.70	75.50	75.10
	256	72.90	75.10	73.50	73.60
	512	73.20	70.80	71.00	71.60

表 6.4

由表得出，本系统的云存储服务器架构具备明显的速度优势。普通 FTP 服务在队列深度达到 256 时，传输速度已经呈现出明显的不稳定性，不具备实用价值，而当将队列深度提升到 512 时，服务器出现了大量拒绝请求的情况，严重超出负载范围，无法记录实时传输速度。本系统设计的云存储服务器架构，在相同的测试条件下，队列深度即使达到 512 仍能轻松应对，服务器 CPU 使用量、内存使用量、硬件温度均在正常工作范围内，传输速度稳定性高，无明显波动，平均传输速度达 72.80Mbps，达到系统设计性能要求。

6.3.5 系统最大并发用户数测试

为了测试在生产环境下，普通用户正常访问时，系统所能并发处理的最大用户数，而忽略少量用户高速传输文件的情况，考虑网络带宽、服务器硬盘物理特性带来的速度限制，本测试通过硬件配置服务器架构，强制限制每用户文件传输速率最大值为 512kbps，依然采用普通 FTP 服务与本系统进行对比测试。

测试对象	并发用户数	传输结果
普通 FTP 服务架构	1000	正确
	2000	正确
	5000	拒绝
业务处理节点集群	1000	正确
	2000	正确
	5000	正确

表 6.5

根据测试结果，在相同硬件条件下，普通 FTP 服务架构当并发用户数达到 5000 时，出现了部分用户被拒绝，服务器 CPU 占用量 100%，硬盘读写曲线严重飘红的情况。而采用本系统设计的云存储结构处理文件，在并发用户数达到 5000 时，依然应对自如，服务器各项指标均在正常范围内。经过深度测试后，得出本系统测试平台可负载的峰值并发用户数接近 10000，完全满足设计要求。

6.4 系统安全性分析

(l)对拒绝服务攻击的防范

攻击者可以发送大量的有效或者无效的数据包，引起认证服务器大量的公钥密码运算，如果通过发送大量的认证请求攻击认证服务器，将占用相当的系统资源，本系统的负载均衡器就能有效处理此类攻击。

(2)重放攻击

整个身份认证系统的身份认证流程由服务器发起，通过服务器和客户端分别产生仅对一次会话有效的随机数，而且认证服务器每次选取的随机数不同，即每次的动态密码不同，每次都是对新的认证信息和随机数进行加密。

(3)网络监听

身份认证过程中，客户端发给服务器端的认证数据包，均经过加密后传输，所以网络监听攻击是无效的。同时由于认证信息是动态变化的，因此这对攻击者来说可利用的攻击时间很短，没有足够的时间进行攻击。

(4)中间人攻击

本系统应用 RSA 公钥认证机制结合 MD5 算法加密口令的方式，通过用户与认证服务器双方相互认证，真正解决了通信双方的身份认证问题，攻击者无法冒充或者进行中间修改然后转发认证信息，有效地避免中间人攻击。

(5)暴力攻击

本系统采用高强度非对称式 RSA 加密算法加密用户验证信息，MD5 算法加密用户登录密码，AES 算法加密用户文件。加密文件的密钥长度达到 256bits，安全性高于传统 3DES 算法 1024bits，因此可以很好的抵御暴力攻击。

(6)伪服务器攻击

验证 TPM 模块时，TPM 模块与认证服务器之间进行双向认证，服务器上记录着 TPM 模块所随机生成的动态密码，攻击者无法冒充或者进行中间修改然后转发认证信息，不会被伪服务器攻击。

(7)穷举攻击

MD5 算法的安全性建立在消息摘要值的长度上，如果采用穷举攻击的办法，产生任何一个报文使其摘要等于给定报文摘要，将需要较高数量级的操作。因此，MD5 能有效地抵抗穷举攻击。

7 总结与展望

7.1 科学性分析及成果体现

本项目目标用户定位明确，应用领域广泛，对于财务部门可有效管理相关财务报表数据，对于研发部门可有效管理相关项目研发数据，对于销售部门可有效管理相关客户资料，对于人事部门可有效管理相关员工绩效数据，对于政府部门可有效管理相关政务数据，对于税务部门可有效管理相关税务数据，对于军事部门可有效管理相关涉密数据。硬件与软件完全自主研发，作者从大学入学（2011 年 9 月)开始进行本项目的研发及相关信息安全资料查阅，更换了数套设计方案，最终研发出本项目，确保了其稳定性及安全性均达到设计要求。

以下重点分析本系统应用几大核心技术的设计方案与实现过程，首先是非对称式与对称式加密技术的整合。

（1）非对称式与对称式算法联合加密技术

我们都知道在信息安全领域，数据加解密算法可分类以下两类，非对称式算法特点为验证机制复杂，防破解能力高，运算效率低，而对称式加密特点为验证机制简单，密钥不易保管，运算效率高。本系统为了获得最高的安全性及最佳的运算效率，将两种算法结合设计。在云存储集群中的所有文件均采用对称式 AES-256 位加密算法存储，以最佳的运算效率完美完成 G 级大容量文件加密。

对称式加密算法的密钥管理又成为本系统需要解决的下一个问题，为保证安全性，所有文件 AES 密钥均由服务器随机生成，这些密钥在完成调用后将被通过非对称式 RSA 加密算法进行二次加密并被存储到服务器数据库，由于加密对象为长度有限的密钥，所以 RSA 算法可以保证加密效率。

本系统需解决的第三个问题便是非对称式加密算法的密钥管理，我们知道 RSA 算法具有两个验证因子：公钥和私钥，使用公钥可以加密却不能解密，而私钥则相反。本系统正是利用这一特性，将公钥与私钥分开保管来确保数据安全性。RSA 公钥对于用户数据不构成威胁，因此将其直接保存在服务器数据库，任何人（包括用户自己或第三方）任何时候需要加密文件，都可方便的提取以实现加密过程。而相应密钥对的 RSA 私钥属于高危因子，本系统将其设计为离线存储由用户自己保管。因此，即使服务器被攻击或入侵，只要黑客无法得到用户

私钥，就可确保数据的绝对安全。

（2）基于 TPM 可信计算架构的 USB 硬件动态密钥验证装置

为确保安全性，RSA 加密算法的密钥对，也是在服务器随机生成的，因此由用户通过传统方法记忆或保管高危私钥是不明智的。本系统因此提出 TPM 可信计算硬件私钥验证装置，将冗长的高危私钥存储在硬件中，保证私钥安全。

本系统需要解决的下一个问题就是，如何确保私钥存储的安全性，首先我提出了开源的思想，在本系统整个硬件验证机制设计初期，就确立了设计电路及内部源程序完全公开的理念。对于信息安全类产品，无法获知内部工作流程是极其可怕的事情，因为其中的设计漏洞或恶意后门存在不确定性。

以下将具体陈述 TPM 硬件验证设计方案，在硬件中体现了三大核心功能：对称式加密，动态密钥更新，熔丝自毁保护。本系统硬件基于 AVR 单片机实现，考虑到硬件资源有限，采用经过嵌入式优化的对称式加密算法，对高危私钥进行加密后存储到硬件内部 ROM 中，因该高危私钥经过对称式算法加密存储，即使本硬件被破解，依然不会对该私钥造成威胁。

在硬件设计中需要解决的第二个问题就是在硬件中调用对称式加密算法的安全性，本系统依靠硬件的动态密钥更新功能来解决。每次调用硬件时，硬件内部将动态生成一条新的密钥，在芯片内部完成对用户高危私钥的解密及使用该新密钥的重新加密。通过动态更新用户加密高危私钥时使用的密钥以确保加密的时效性及安全性。

在硬件设计中需要解决的第三个问题就是每轮动态更新密钥时的密钥保管问题。如何确保旧密钥在执行解密时的可行性及新密钥在执行重新加密时的安全性呢？本系统给出的解决方案是通过云端服务器数据库来实现硬件密钥的保管。每轮动态密钥更新前，系统都将提取服务器数据库中的存储的旧密钥回传给硬件，硬件在芯片内部完成密钥处理工作后，返回新密钥给服务器作更新。通过这样的设计，即使密钥在传输过程中被监听，由于每轮密钥都不同，故可视为安全。

（3）多系统跨平台全硬件 TPM 验证的文件管理

2013 年，物联网与云计算技术正在逐渐普及，我们清晰的发现 IT 业发展进入后 PC 时代，笔记本电脑逐渐取代台式机，平板电脑逐渐取代笔记本，手机逐渐取代平板电脑。信息流的处理渠道更为复杂多样。如何让用户能随时随地的处

理信息，本系统 TPM 硬件验证模块的跨平台多终端设计就有效解决了该问题。在 WINDOWS 操作系统，本项目基于 LIBUSB 函数库开发了 USB 驱动通信层，并使用 VISUAL BASIC.NET 语言开发了云计算文件管理客户端。在 LINUX/ANDROID 操作系统，本项目基于 LINUX-OTG 技术开发了 USB 驱动通信层，并使用 PHONEGap 框架开发了云计算文件管理客户端。

用户可以通过 OTG 线将本 TPM 可信计算硬件模块直接接驳到自己的智能手机及平板电脑，也可以通过 USB 接口接驳到普通台式机。由此实现了用户多终端硬件级的文件安全管理。

（4）云计算架构前端防护，永不停机，无限扩容

传统离线存储方式（U 盘、硬盘）容量有限安全性差，传统在线存储方式（B/S，C/S，邮件，网络硬盘）可靠性低安全性差，本系统通过云计算架构的科学布局，可有效解决以上问题。首先是服务器架构的多层布局，最上层为前端服务器，其承担着抗 DDOS 流量攻击，业务交接的功能，前端服务器本身不存储任何用户数据及网站信息，负责业务分发，因此即使被攻击也不会造成任何威胁。第二层为业务处理服务器，前端通过判断业务请求合法性并对请求分类后转交给业务处理服务器处理，业务处理服务器本身不存储任何用户数据，负责业务处理，因此即使被攻击也不会对数据造成威胁。第三层为数据存储服务器，分别布局用户信息数据库及多地用户文件存储节点，用户文件备份存储节点。任何层级监测到攻击行为，都将立即通知管理员并断开与下层的数据传输，确保数据安全。

云计算架构的又一优势在于在线容错，无限扩容。当任何业务处理服务器出现问题时，前端服务器将自动把业务转交到其他服务器处理，该问题服务器自动离线挂起并通知管理员检查，以此实现云端永不停机 7X24 正常运行。当业务处理或数据存储出现瓶颈时，管理员可通过简单配置直接增加相应层级的服务器，确保云端资源的永不饱和，存储空间的无限扩容。

（5）基于非对称式授权的文件共享与分发

由于云端用户加密业务核心基于非对称式算法，因此可满足文件安全共享与定向分发的需求。如果用户 A 需要发送文件给用户 B，系统将从服务器数据库中提取用户 B 的公钥，并使用此公钥加密需要共享分发的文件，存放在用户 B 的云盘中。当用户 B 连接系统后，即可浏览到用户 A 发送的文件，并使用自己

的私钥解密该文件后读取。该机制充分解决了加密文件传递困难或密钥暴露的问题，实现了涉密文件的安全授权共享。

7.2 特色与创新分析

1. 软件硬件自由开放，支持升级功能：由于整个 TPM-USB 可信计算硬件装置从电路设计，固件程序设计及底层驱动设计均是作者基于单片机独立开发完成，而未采用市面上的现有解决方案，所以可以确保验证机制的创新性，有效避免部分商业产品因源程序封闭而造成的恶意后门及漏洞风险。
2. TPM 动态密钥联网验证机制：在硬件设计中体现了三大核心功能：对称式加密，动态密钥更新，熔丝自毁保护。存储在硬件中的用户私钥经过加密，不可破解。每次使用都会用动态密钥更新，并且与服务器同步，确保一次一密。当硬件验证中监测到异常时自动启动熔丝自毁保护。
3. 非对称式与对称式算法联合加密技术：本系统通过对称式加密算法保证了大数据文件加密与解密的效率，并结合非对称式加密算法保证了用户数据验证的多因子，将公钥与私钥分开保管，让用户自己掌握私钥保管的的主动权，确保了用户资料的易用性及安全性。
4. 多系统跨平台全硬件 TPM 验证的文件管理：用户可以通过 OTG 线将本 TPM 可信计算验证模块直接接驳到自己的智能手机及平板电脑，也可以通过 USB 接口接驳到普通台式机。由此实现了用户多终端硬件级的文件安全管理。
5. 云计算架构的不停机在线容错：采用服务器分层架构，任何层级监测到攻击行为，都将立即通知管理员并断开与下层的数据传输。当任何业务处理服务器出现问题时，前端服务器自动把业务转交到其他服务器处理，实现云端永不停机。
6. 云计算架构的不饱和无缝扩容：本系统可在保持前端用户正常访问的情况下，实现不中断服务的无缝硬件扩容与硬件升级，达到云端服务器永不饱和、弹性扩展的特点，满足各种规模用户群的加密文件存储需求。
7. 基于非对称式授权的文件共享与分发：文件分发者不需要知道接收对象的私钥就可直接发送加密文件给对方。而只有拥有对应私钥（TPM-USB 可信计算验证装置）的文件接受对象，才能解密该文件。该机制充分解决了加密文件传递困难或密钥暴露的问题，实现了涉密文件的安全授权共享。

7.3 实用性分析及应用推广前景

本项目具有高安全性、低成本、TPM 硬件验证、云存储的特点。由于量产成本低（<10 元），应而具有极高市场价值。由于充分考虑用户体验，用户不会因增加硬件验证而占用操作时间，并且支持多终端（手机/平板/笔记本/台式机）文件操作，应而具有极高实用价值。

本项目目标用户定位明确，应用领域广泛，对于财务部门可有效管理相关财务报表数据，对于研发部门可有效管理相关项目研发数据，对于销售部门可有效管理相关客户资料，对于人事部门可有效管理相关员工绩效数据。硬件与软件完全自主研发，历时近两年，更换数套设计方案，最终研发出本项目，确保稳定性及安全性均达到设计要求。

在 2012 年首都大学生科技创新作品与专利成果博览会中，本项目获得推荐向相关领域专家进行现场展示，获得了诸多好评，并且同时申请了国家发明专利。在 2013 年本项目经过推荐，得以在北京市政府某重点政务服务项目中进行试点推广，项目稳定性、安全性及易用性均获得好评。

8 参考文献

[1] Lei Han, Jiqiang Liu, Zhen Han, Xueye Wei: ,Design and implementation of a portable TPM scheme for general-purpose trusted computing based on EFI,中国计算机科学前沿(英文版) 2011 年 第 2 期

[2] Peng Shuang he ,Design and implementation of portable TPM device driver based on extensible firmware interface,2009 International Conference on Multimedia Information Networking and Security (MINES 2009) [0-7695-3843-6; 1-4244-5068-3]

[3] Building Trust into Cloud Computing Using Virtualization of TPM,Multimedia Information Networking and Security (MINES), 2012 Fourth International Conference

[4] A Security Authentication System Based on Trusted Platform ,Information Science and Engineering (ICISE), 2009 1st International Conference

[5] Research and application of trusted computing platform based on portable TPM ,Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference

[6] LiuZhen-Peng ,Mutual authentication scheme based on the TPM cloud computing platform,通信学报 [1000-436X] 年:2012 卷:33 期:2 页:20 -24

[7] LiuYanfei ,A trusted network platform architecture scheme on clouding computing model,2012 International Conference on Computer Science and Information Processing, CSIP 2012 [1-4673-1411-0]

[8] KaiTang The secure boot of embedded system based on mobile trusted module,2012 International Conference on Intelligent Systems Design and Engineering Applications, ISDEA 2012 [0-7695-4608-0]

[9] Design Principles for Trusted Platform Modules Protected with Power Analysis ,Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference

[10] Tang, Jian ,Trusted network model based on trusted platform module,计算机工程 [1000-3428]年:2011 卷:37 期:11 页:117 -19

[11] 杨旭峰.基于 ARP 协议和 Agent 协作方法的动态入侵防护.电脑知识与技术 2009 年第 23 期

[12] 徐远航.USB Key 身份认证产品的产生与发展.计算机安全 2004 年第 8 期

[13] 邵子扬，王育强，吕益光.AVRUSB 技术探讨.电子产品世界 EEPW 网 2008

[14] 谢晓燕.网络安全与管理实验教程.西安电子科技大学出版社 2008

[15] 袁巍.AES 算法的设计原则与其密钥扩展算法的改进.吉林大学 2010 年

[16] 李隽.看图识云 全面解析云存储的网格架构.IT168 网 2008

[17] 张宴.实战 Nginx：取代 Apache 的高性能 Web 服务器.电子工业出版社 2009

[18] 张勤.开源 IT 系统及应用架构宝典--系统、工具、案例.人民邮电出版社 2010

[19] 肖斌.Visual Basic 6 网络编程实例教程.北京希望电子出版社 2002

[20] 阙喜戎.信息安全原理及应用.清华大学出版社 2003

[21] 三恒星科技.AVR 单片机原理与应用实例.电子工业出版社 2009

[22] 许永和.8051 单片机 USB 接口 Visual Basic 程序设计.北京航空航天大学出版社 2007

[23] 汤惟.密码学与网络安全技术基础.机械工业出版社 2004

[24] Andrew Nash.公钥基础设施 (PKI) 实现和管理电子安全.清华大学出版社 2003

[25] 林柏钢.网络与信息安全教程.机械工业出版社 2004

[26] 肖踞雄, 翁铁成, 宋中庆.USB 技术及应用设计.清华大学出版社 2003

[27] 张先红.数字签名原理及技术.机械工业出版社 2004

[28] 张军.AVR 单片机应用系统开发典型实例.中国电力出版社 2005

[29] 刘荣.圈圈教你玩 USB. 北京航空航天大学出版社 2009

[30] USB Host API ,Android API Guide,2012

9 附件目录

1. 专利受理：基于非对称式硬件密钥联网验证的数据安全传输方法
2. 软件著作权证书：基于 TPM 与云计算技术的跨平台安全云盘
3. 查新报告：中国科学技术信息研究所检索查新中心
4. 查新报告：中国科学院文献情报中心
5. 获奖证书：首都大学生科技创新及专利成果博览会推介作品证书
6. 获奖证书：第七届首都“挑战杯”大学生课外学术科技作品竞赛获奖证书
7. 推荐函：中国科学院与中国工程院信息安全研究领域，王越院士
8. 推荐函：国家计算机网络与信息安全管理中心，刘欣然处长
9. 期刊杂志：《信息网络安全》期刊发表学术论文 2 篇