选题报告

一、选题的来源、研究的目的意义（包括在我国应用的前景）、学术和应用价值、创新点以及国内外研究现状及水平：

**1.研究背景意义**

在当今信息时代，互联网已经渗入到人们生活的各个领域。随着互联网技术不断的升级，人们工作与生活产生的数据也呈现指数形式的增长，海量数据存储与处理的需求日益增长，云计算技术应运而生。云计算，根据美国国家标准与技术研究院定义[1]：云计算是一种按使用量付费的模式，这种模式提供可用的、便捷的、按需的网络访问，进入可配置的计算资源共享池（资源包括[网络](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C/143243)，服务器，存储，[应用软件](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E8%BD%AF%E4%BB%B6/216367)，服务），这些资源能够被快速提供，只需投入很少的管理工作，或与服务供应商进行很少的交互。云存储，作为云计算的一种典型服务类型[2]，可以很好地解决用户对海量数据或碎片化数据的存储和处理需求。云存储是指将接入互联网的大量终端通过集群应用、网络技术或分布式文件系统集合起来，借助应用软件虚拟化后对外提供数据存储和访问功能。

伴随移动计算需求的持续增长，个人拥有多种智能设备已成为普遍现象，用户可以通过多种设备在不同的场景进行工作或访问网络资源，极大地丰富了互联网的使用场景，带来无与伦比的便利性。为了实现不同终端都能访问到网络资源并保持数据同步，以Dropbox[3]为代表的网盘类应用服务在过去的数年里发展迅猛。网盘，又称作网络U盘、网络硬盘，是由互联网公司推出的在线存储服务，服务器商为用户分配一定的磁盘空间，为用户提供免费或收费的文件的存储、访问、下载等文件管理等功能。网盘应用服务本质上是AaaS模式的云计算，通过提供客户端或API的方式，使用户能够从兼容的终端上传文件到云端，并支持用户通过客户端程序或API从任意终端访问。

网盘应用服务让用户像在操作本地文件一样方便快捷的存取云端数据，并保持数据在多个设备间的同步。但我们注意到，在享受网盘服务带来的便捷性的同时，用户的数据安全和隐私正在面临风险。

首先，数据安全在网盘应用服务中具体表现为数据在云端存储的安全与数据传输过程中的安全。从近年来频繁发生的云存储数据泄露事故[4]的事实中不难看出，网盘应用服务并没有给用户提供有效的存储保护措施或仅提供了非常简单的加密手段，因此用户保存在云端的数据面临巨大风险，一旦云存储服务提供商的系统遭到渗透，用户未经妥善保密的数据将会轻易的被他人获取。近年来已经发生多起重大云数据泄露事件，给用户、云存储服务提供商造成了巨大的损失，甚至引发舆论危机。例如苹果公司的iCloud数据被黑客攻击[5]，大量用户数据泄露，包括一些极度私密的数据，给用户带来精神上的巨大损失。另一方面，很多网盘应用服务在数据传输时并没有引入有效的加密措施，在无法确保信道可靠的情况下使用明文传输，使得黑客能够轻易地在信道中截获用户数据，甚至发动中间人攻击进行进一步渗透。

其次，我们认为当前的网盘应用服务在隐私保护方面不够完善，通常情况下，用户数据在云端以明文形式存储，即便部分网盘应用服务提供了加密存储功能，用户的数据的加密过程通常在云端完成，密钥亦保存在中心化云服务提供商处。此外，为了提升存储效率，多数云服务提供商都采用了去冗余技术，该技术的基本原理是对用户上传的文件进行哈希，哈希值相同的文件即被认为是同样的文件，可以指向同一物理存储位置。因此云服务提供商很容易窥探用户所存储的数据或对用户的数据进行解密。同时，用户对数据的存取操作亦完全被云服务提供商所知，因为用户对云端数据进行检索、查询的操作是完全暴露给云服务提供商的。在大数据分析方法趋于完善的今天，云服务提供商会有意识地搜集用户的各种数据并进行分析，进而实现其商业目的，或协助政府进行内容审查。Facebook公司在2018年被爆出泄露用户隐私数据给其他公司分析使用[6]，Facebook公司因此被推上舆论的风口浪尖。

综上所述，我们认为现有的网盘应用服务主要存在两方面风险，即数据安全风险和用户隐私风险。在本项目的研究中，我们希望解决这两项痛点问题，设计出安全的隐私可控的新型网盘系统。在数据安全方面，我们计划引入加密存储技术并设计新型数据可靠传输协议，从而分别应对数据在云端和信道传输过程中的安全问题。在隐私保护方面，我们计划引入可搜索加密技术以保证用户在数据检索过程中不会泄露行为数据，同时，我们还计划使用去中心化的密钥管理机制，使得数据访问权限完全控制在用户手中。最后，为了使得系统具有更好的现实意义，我们将会实现原型系统，并对相关算法进行优化，使整个系统的效能达到实用水平。

# 2国内外现状

**（1）加密搜索**

加密搜索，又称可搜索加密[7]（searchable encryption, SE），可搜索加密主要解决在服务端不完全可信的情况下对加密数据的关键字安全搜索。2000年，Song等人首次提出可搜索加密的概念。在借助这一技术下，用户借用服务器强大的计算资源进行关键字查询同时不会向服务器泄露任何隐私信息。这种模式下，不仅仅保护了用户数据的隐私，也防止他人窃取用户的信息，最后还能极大的提高对加密数据的查询效率。目前，可搜索加密在云存储技术上也有很大进展。

**（2）网盘现状**

当今社会也不乏一些主流的网盘服务产品，例如百度公司的百度云、115网盘和国外Dropbox公司的Dropbox。这些网盘应用服务确实给用户提供了很好的服务体验，但是如果从数据的安全性和用户的隐私性来看，这些网盘应用服务可能存在一些安全漏洞和隐私问题。

当今社会主流网盘应用服务，按照对用户安全和隐私的保护程度可以大致划分为三大类。

第一类网盘的设计是服务商对用户的数据没有任何加密和保护，即用户数据明文存储和明文传输，这类网盘产品代表有115网盘和百度网盘。首先，明文存储不仅让用户的数据面临被非法获取的风险，而且给与了服务商收集用户数据的能力。其次，在网络信道不可信的情况下进行明文传输，用户数据亦有被篡改和泄漏的可能。结合上述讨论可以看出这种类型的网盘软件在使用过程中既不能保护用户的隐私也不能确保用户数据的安全。

第二类网盘的设计是软件服务提供商对用户保存的文件进行加密和解密。这类网盘的代表有Dropbox公司推出的Dropbox网盘。服务商对文件进行加密解决了用户的数据存储安全问题，只要服务商加密解密的密钥不丢失，那么服务商可以防止他人在服务端盗取用户的数据内容。由于这种加密手段是由服务商提供，密钥也是由运营商生成和保管，因而服务商也具有检索与收集用户数据的能力。总体来看，这种加密设计虽然解决了数据存储时候的安全问题，可是却没有很好地保护用户的隐私。此外，用户数据在传输过程中仍面临安全威胁。

第三类网盘的设计是由客户端在本地加密，加密完成后上传到服务端，密钥由用户进行管理。这类网盘服务的代表是一款名为“隐形云”的产品，由杭州奕锐电子有限公司和阿里云合作推出的一个在线云盘。在安全方面，对数据的加密可以防止他人非法获取数据内容，从而保证了用户数据的存储安全。针对用户存储数据的隐私问题，网盘设计中采用让用户保管密钥，服务商无法查看用户的数据内容，从而保证了用户存储数据的隐私不被泄露。但是“隐形云”缺点在于两点。第一点是使用时必须要先下载客户端才能使用这个网盘。第二点是用户使用网盘软件的过程中，网盘软件并没有对用户交互的信息进行加密。例如用户在使用搜索过程中的输入的隐私信息可能会泄露给服务端，服务端进而可以收集用户信息用作数据分析。

通过分析这三种主流网盘服务软件，我们可以发现主流网盘应用服务在用户数据的隐私和安全上还存在很多潜在的问题。从用户角度考虑，用户需要一款既能安全存储数据又能保护自身隐私的网盘应用软件。

**（3）新型安全网盘系统模型**

基于对主流网盘应用服务的分析与总结，我们提出了一个新型安全网盘系统模型。相比较之前主流的网盘系统，新的系统具有以下的优点。

一、数据上云的全生命周期均处于加密状态，且加密、解密过程不受服务商干预。文件上传之前会先进行加密，文件下载后进行解密，文件加解密操作在用户侧完成，文件加密解密的密钥由用户自身保管，确保了用户数据在上传与存储的安全。

二、新的网盘系统在传输数据过程中采用了基于UDP传输协议设计的可靠文件传输协议。使用UDP文件传输方式，将大文件分片传输，以及后续对文件片的校验更是确保文件传输过程中的安全性与完整性。采用新型传输协议，传输文件的速度在控制上更具有灵活性。

三、保护用户隐私，我们引入可搜索加密技术,实现用户对加密数据的检索，进而保护用户隐私。可搜索加密允许用户在密文上进行关键字查找，在服务商不可信的情况下，使用这种方式来处理用户的检索信息可以保证用户搜索时的隐私。

四、针对用户的特定文件,我们提供版本控制的功能。对一些重要的文件可以追踪其变化，如果重要的文件不小心被删除或者修改就可以借助此功能回溯到指定的版本。

**（3）创新点分析**

1、基于UDP的可靠传输协议设计

2、将加密搜索概念引入到新型安全网盘系统中，并实现该系统

3、对加密搜索过程中搜索算法进行改进，优化加密搜索的性能。

二、论文研究的主要内容，方案和拟采用的研究方法、手段。已进行的科研工作基础和已具备的科学研究条件（包括文献资料及主要实验仪器设备准备情况等），对其它单位的协作要求。论文总工作量（估计），论文初稿的进度以及预期结果：

**1、论文研究的主要内容**

在本论文中，作者阐述了一种新型的基于加密搜索技术的安全网盘系统的设计与实现。论文工作主要从两方面展开

**1.1可搜索加密文件管理系统设计及实现**

**1.1.1传统网盘系统架构设计**

传统网盘系统设计架构图如下



图1.1

传统网盘系统架构概念如图1.1所示。网盘的设计分为服务端和客户端。在客户端方面，用户通过无线/有线方式连接到互联网，然后通过相应的客户端或者浏览器方式登录系统。客户端/浏览器会提供一系列交互界面，用户通过客户端交互界面可以对自己网盘内的文件进行一系列操作。至于服务器方面，服务商通常会使用数据库进行身份认证，然后通过接口服务器用于接受来自客户端的请求。根据不同请求，服务端会调用不同的服务器完成各项操作然后将处理结果返回给客户端。

**1.1.2威胁模型**

通过对传统的网盘系统架构进行分析，结合我们之前对于网盘应用服务中安全和隐私问题的讨论，我们可以总结出传统网盘的安全和隐私漏洞。为了更清楚地阐述情形，这里采用C/S架构进行描述（B/S情况和C/S架构下是完全相同的）。首先，我们假设服务器与网络环境是不可信的，接下来我们针对网盘客户端的一系列操作进行安全分析：（1）文件在上传与下载时可能他人会通过网络信道窃取数据信息，因此数据在网络环境中的传输需要进行加密处理。（2）文件存储在服务端应该以密文进行存储，基于服务端不可信的前提下，文件加密与解密的密钥应该由客户端进行保存。（3）用户可能对自身文件进行查询，这种查询一般是基于关键字查询的搜索，在服务端与通信链路都不可靠的环境下为了保护用户的隐私与安全，我们需要引入加密搜索技术。

**1.1.3网盘系统设计**

**1.1.3.1服务端设计**

服务端主要功能有存储、交互、传输、加(解)密和加密搜索，因此按照功能的分类，我们设计三个模块负责上述功能的实现。三个模块分别为存储模块、传输模块和加密模块。其中存储模块主要负责文件的存取、文件版本控制和访问控制，传输模块主要负责数据的可靠传输，加密模块主要负责对信息的加（解）密、密钥的生成和提供加密搜索功能。

**1.1.3.2客户端设计**

客户端按照功能设计，也可以将不同的功能按照模块化设计分为三大模块。分别是前台模块，加密模块和传输模块。前台模块主要是给用户提供界面展示以及各种文件管理的操作，加密模块主要负责文件加（解）密、信息的加（解）密和密钥的生成。传输模块主要负责数据的可靠传输。

**1.1.4版本控制**

在使用网盘应用服务的过程中，用户可能会对网盘里的文件做多次修改。网盘中文件的修改的形式主要分为两种，一种是文件本身内容的增加或者减少，常见为文件的重命名和文本类文件的编辑。另外一种是文件的新增和删除。这两种形式的修改都是用户基于当前情况做出的决定，但很有可能的是用户在修改文件一段时间后希望能够将某个文件或者整个网盘回溯到之前的状态，而这种回溯的需求在文本类文件中尤为常见。考虑到这种需求，我们为网盘添加了基于时间点的版本控制功能，用户可以通过备份的时间点来将网盘文件重置到特定时间下的状态。

**1.1.5可靠传输协议设计**

日常所用软件在网络中传输数据时都需要借助网络协议才能工作，常用的传输协议有UDP（User Datagram Protocol）和TCP（Transmission Control Protocol ）。UDP又名数据包传输协议，在OSI七层网络结构模型中位于第四层-传输层，是一种无连接，面向事务的简单不可靠信息传送服务。同样位于传输层的另外一个传输协议是TCP文件传输协议。UDP相比较TCP传输协议有以下特点：（1）UDP是无连接不可靠的传输协议，TCP是连接可靠的传输协议。（2）UDP比TCP更能节省处理器资源，由于无连接，UDP在发送时候更具有灵活性。（3）UDP是面向数据包协议，而TCP是面向字节流协议。所以我们在UDP传输数据过程中对于传输内容可以更进一层封装上我们自定义的协议信息。（4）传输速度上面，TCP发送速度是基于当前硬件和网络环境自适应的，UDP发送速度是可以由我们指定。在传输速度的控制上，UDP相比较TCP更具有灵活性。

结合以上关于TCP与UDP之间的比较，新型安全网盘系统采用了UDP协议作为文件传输的基础协议。在UDP基础之上我们设计了新型可靠传输协议。借助新型传输协议，新的安全网盘系统可以实现文件在不可靠信道下的安全可靠传输。

**1.1.5 加密算法**

加密算法通常分为两大类，即对称加密[8]和非对称加密[9]。

**1.1.5.1对称加密**

对称加密算法是应用较早的加密算法，技术成熟。在对称加密算法中，数据发信方将明文（原始数据）和加密密钥一起经过特殊加密算法处理后，使其变成复杂的加密密文发送出去。收信方收到密文后，若想解读原文，则需要使用加密用过的密钥及相同算法的逆算法对密文进行解密，才能使其恢复成可读明文。在对称加密算法中，使用的密钥只有一个，发收信双方都使用这个密钥对数据进行加密和解密，这就要求解密方事先必须知道加密密钥。对称加密算法的特点是算法公开、计算量小、加密速度快、加密效率高。不足之处是，交易双方都使用同样钥匙，安全性得不到保证。此外，每对用户每次使用对称加密算法时，都需要使用其他人不知道的惟一钥匙，这会使得发收信双方所拥有的钥匙数量成几何级数增长，密钥管理成为用户的负担。对称加密算法在分布式网络系统上使用较为困难，主要是因为密钥管理困难，使用成本较高。在计算机专网系统中广泛使用的对称加密算法有DES和AES。DES全称为Data Encryption Standard，即数据加密标准，是一种使用[密钥加密](https://baike.baidu.com/item/%E5%AF%86%E9%92%A5%E5%8A%A0%E5%AF%86/5928903" \t "_blank)的块算法，1977年被[美国联邦政府](https://baike.baidu.com/item/%E7%BE%8E%E5%9B%BD%E8%81%94%E9%82%A6%E6%94%BF%E5%BA%9C/8370227" \t "_blank)的国家标准局确定为[联邦资料处理标准](https://baike.baidu.com/item/%E8%81%94%E9%82%A6%E8%B5%84%E6%96%99%E5%A4%84%E7%90%86%E6%A0%87%E5%87%86/3940777)（FIPS），并授权在非密级政府通信中使用，随后该算法在国际上广泛流传开来。传统的DES加密只有56位密钥，在1997年RSA公司发起对DES加密挑战时，DES很快被破解，因此DES在计算机运算速度提升后的今天被认为是不安全的。高级加密标准（英语：Advanced Encryption Standard，缩写：AES），在[密码学](https://baike.baidu.com/item/%E5%AF%86%E7%A0%81%E5%AD%A6/480001" \t "_blank)中又称Rijndael加密法，是[美国联邦政府](https://baike.baidu.com/item/%E7%BE%8E%E5%9B%BD%E8%81%94%E9%82%A6%E6%94%BF%E5%BA%9C/8370227" \t "_blank)采用的一种区块加密标准。这个标准用来替代原先的[DES](https://baike.baidu.com/item/DES)，已经被多方分析且广为全世界所使用。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](https://baike.baidu.com/item/%E7%BE%8E%E5%9B%BD%E5%9B%BD%E5%AE%B6%E6%A0%87%E5%87%86%E4%B8%8E%E6%8A%80%E6%9C%AF%E7%A0%94%E7%A9%B6%E9%99%A2/3931459" \t "_blank)（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。2006年，高级加密标准已然成为[对称密钥加密](https://baike.baidu.com/item/%E5%AF%B9%E7%A7%B0%E5%AF%86%E9%92%A5%E5%8A%A0%E5%AF%86" \t "_blank)中最流行的算法之一。

相比较于DES，AES加密算法会更加安全，因此我们在本论文的工作中也是采用了AES加密技术，确保加密的可靠性。

**1.1.5.2 非对称加密**

不对称加密算法使用两把完全不同但又是完全匹配的一对钥匙—公钥和私钥。在使用不对称加密算法加密文件时，只有使用匹配的一对公钥和私钥，才能完成对明文的加密和解密过程。加密明文时采用公钥加密，解密密文时使用私钥才能完成，而且发信方（加密者）知道收信方的公钥，只有收信方（解密者）才是唯一知道自己私钥的人。不对称加密算法的基本原理是，如果发信方想发送只有收信方才能解读的加密信息，发信方必须首先知道收信方的公钥，然后利用收信方的公钥来加密原文；收信方收到加密密文后，使用自己的私钥才能解密密文。显然，采用不对称加密算法，收发信双方在通信之前，收信方必须将自己早已随机生成的公钥送给发信方，而自己保留私钥。由于不对称算法拥有两个密钥，因而特别适用于分布式系统中的数据加密。广泛应用的不对称加密算法有RSA算法和美国国家标准局提出的DSA。以不对称加密算法为基础的加密技术应用非常广泛。

DSA（Digital Signature Algorithm）是Schnorr和ElGamal签名算法的变种，被[美国](https://www.baidu.com/s?wd=%E7%BE%8E%E5%9B%BD&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "_blank)NIST作为DSS(DigitalSignature Standard)。 DSA是基于整数有限域离散对数难题的。DSA被常用作数字签名，并且DSA不具有加密解密功能。相比较之下，RSA具有加密和解密功能，同时也能用作数字签名。RSA是一种基于大数做因数分解的加密算法，RSA是1977年由[罗纳德·李维斯特](https://baike.baidu.com/item/%E7%BD%97%E7%BA%B3%E5%BE%B7%C2%B7%E6%9D%8E%E7%BB%B4%E6%96%AF%E7%89%B9/700199" \t "_blank)（Ron Rivest）、[阿迪·萨莫尔](https://baike.baidu.com/item/%E9%98%BF%E8%BF%AA%C2%B7%E8%90%A8%E8%8E%AB%E5%B0%94" \t "_blank)（Adi Shamir）和[伦纳德·阿德曼](https://baike.baidu.com/item/%E4%BC%A6%E7%BA%B3%E5%BE%B7%C2%B7%E9%98%BF%E5%BE%B7%E6%9B%BC/12575612" \t "_blank)（Leonard Adleman）一起提出的。RSA就是他们三人姓氏开头字母拼在一起组成的。

在论文中，我们采用RSA加密算法，这样既可以对信息进行加密也能用作签名认证。

**1.2加密搜索算法优化**

**1.2.1可搜索加密实现**

可搜索加密的介绍上文已经阐述，接下内容我们聚焦于加密搜索的具体实现方式[10]。

SE（加密搜索）的实现方法较多，但是基础的SE的机制主要包括4种算法[8]，分别是密钥产生（Gen）、搜索凭证的生成（GenToken）、索引建立（BuildIndex）和搜索执行（Search）。

（1）Gen：该算法主要由数据所有者或者权威的公信机构运行，主要用来产生密钥，算法会根据输入的安全系数生成相应的密钥。

（2）GenToken：该算法以用户的输入的搜索关键字为输入，产应相应的搜索凭证，主要用于后续搜索执行。

（3）BuildIndex：该算法是由数据拥有者执行，数据拥有者基于数据选出关键字，并在可搜索加密机制上建立索引表。在基于公钥加密的SE机制中，数据拥有者会使用公钥对关键字进行加密，在对称加密的SE机制中，数据拥有者使用对称密钥或者使用哈希算法对关键字集体进行加密。

（4）Search：该算法是由服务器端进行，服务器将得到的搜索凭证和索引表作为输入，服务器依靠自身进行计算最后输出结果判断该文件是否满足搜索请求。

本论文中的加密搜索模型也是基于以上四个基本算法展开，在此基础上我们补充一些新的算法用作改进我们加密搜索模型的性能。

**1.2.2 可搜索加密模型**

可搜索加密按照构造算法的不同可以分为两大类，第一种是基于对称加密的可搜索加密模型，第二种是基于公钥加密的搜索加密模型。两者的区别在于：一、基于对称加密的可搜索加密模型主要使用一些伪随机函数生成器、哈希算法和对称加密算法等构成，而基于公钥加密的可搜索加密模型主要通过一些复杂问题，将安全性建立在复杂性问题的求解难度上。因此相比较于对称可搜索加密模型，公钥加密搜索模型计算开销更大。二、基于对称加密的可搜索加密模型适合单用户创建文件多用户分享的情况，而公钥加密可搜索模型允许文件拥有者之外的用户使用可搜索加密技术生成数据密文并产生新的加密索引表。结合上述分析，新的网盘系统中选用了基于对称加密的可搜索模型。

**1.2.3可搜索加密机制分析**

可搜索加密一个关键用处就是给用户提供安全，隐私的搜索功能。实验中，我们采用了支持单词字的搜索模型。

单词字搜索模型是指对客户端将用户输入的特定关键字转化为搜索凭证并将搜索凭证发送到服务器，服务器用自身的计算能力通过该搜索凭证搜索包含该关键字的文件并将符合条件的结果返回给客户端。之前内容讲到，可搜索加密模型分为四个步骤，即密钥生成、搜索凭证生成、建立索引和搜索执行。加密搜索很好的保护了数据使用者和拥有者的隐私，这是加密搜索的优势所在。因此在搜素执行这一步骤的性能是作为评价加密搜索模型整体性能的重要指标。

搜索执行的过程就是服务器通过将搜索凭证与索引表进行比较来判断当前文件是否符合搜索条件。一些文章在应用加密搜索相关技术在搜索执行这一步用的方法是将搜索凭证与索引表中的关键字进行一一比对。假设当前有m个文件，每个文件平均有n个关键字。那么原始算法的最坏时间复杂度为O（m\*n），即搜索算法的时间复杂度与关键字总量呈线性关系。也就是说每次查询都需要浏览一遍所有的关键字，这种方法在关键字数量较少时候对系统性能不会有影响，而当关键字数量达到一个比较大的数量级后这种方法会严重的影响系统的性能。因此对搜索执行过程的优化是非常有必要的。

**1.2.3可搜索加密算法的优化**

考虑到关键字的数量会随着大量文件的添加而急速增加，单纯的线性搜索时间不能够适用于大量关键字的情况。首先，单纯的关键字搜索类似于对整个关键字集合进行逐一遍历，这种方法是不会发生查找遗漏情况，但是效率较低。其次，搜索执行的过程涉及到关键字的比对，从关键字分布情况分析，关键字并不是孤立存在，可能好几个关键字都是相近的或者是描述同一文件的。根据这一结论，我们可以先将所有关键字这个大集合按照某种方法划分成小的集合，当我们进行搜索时可以先判断关键字是否属于某一个小集合，然后在小集合里进行精确查找。按照这种方法，我们可以将新的搜索执行过程划分为三个步骤：（1）先将关键字按照不同属性划分成不同的小集合（2）每个小集合生成一个特征值（3）查询时，先将由关键字生成的token与小集合的特征值进行比对，找到匹配的特征值后再进入相应的小集合进行精确查找。如果所有特征值都不匹配则返回空值。

在上述搜索步骤中，我们用一个特征值来代替一个小集合所有的关键字进行搜索比对，所以特征值的选取需要能代表整个集合的特征，选取过程中我们引入密码学中累加器概念。一个密码学上的累加器[11]是一个[单向的](https://zh.wikipedia.org/wiki/%E5%96%AE%E5%90%91%E5%87%BD%E6%95%B8" \o "单向函数)[隶属函数](https://zh.wikipedia.org/wiki/%E9%9A%B8%E5%B1%AC%E5%87%BD%E6%95%B8)。它可用于识别一个候选是否为一个[集合](https://zh.wikipedia.org/wiki/%E9%9B%86%E5%90%88)的成员，且不会在过程中暴露[集合](https://zh.wikipedia.org/wiki/%E9%9B%86%E5%90%88)中的成员。在此基础借助密码上累加器这一特点，我们可以将一组集合中toekn值累加到一个值，并且对于任意搜索的一个token值判断是否已经在这个集合内。

假设当前有m个文件，每个文件平均有n个关键字。那么原始算法的最坏时间复杂度为O（m\*n）,而按照我们新提出的算法最坏情况下的时间复杂度为O（m+n）。可以看出新的算法比原始算法在查找关键字的效率上要高很多。

**2、已进行的科研工作基础**

1）UDP可靠传输协议设计与实现

1、数据格式设计

UDP传输协议是无连接不可靠，因此如果想可靠传输我们需要另外添加上序号信息。具体设计如下

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A | B | C | D | E | F |

其中A为传输时的序号，通过文件进行hash取值得到，B是当前数据片所在的序号，C是表示数据片总数，D代表文件此数据片所含数据信息的长度，E代表当前数据片的校验信息，F是数据片中的信息长度。A处主要用于标识文件，后续进行多文件传输会用来区分不同的文件。B处表示在当前数据片在所属文件的所有数据片中的位置。C标记为同一文件所分成数据片的总数，后续在进行传输速度控制时需要记录此信息。D标记功能在于说明该数据片所包含有效数据的长度，防止在文件重组时发生越界数据写入。E标记用作当前数据片的校验，防止数据片在传输过程中被篡改。F标记携带了原文件的内容信息。

2、指令控制

数据传输过程中需要进行指令控制，指令控制的意义在于数据接收方对于接收到的数据信息作何处理以及发送方和接收方如何交互。指令需要再TCP情况下进行，其具体内容代表当前的状态，基础的指令分为5个部分：（1）发送（2）完成（3）同意（4）结束（5）校验。通过这基础5个指令可以识别当前传输状态。具体交互情况如图1.2



图1.2

2）新型网盘系统设计以及组件设计

新型安全网盘系统设计如下图1.3



图1.3

3）可搜索加密算法的优化

根据阅读的论文以及对可搜索加密机制的分析，我们提出了基于动态累加器的优化算法，算法过程如下：（1）将关键字按照不同属性划分成不同的小集合（2）每个小集合生成一个特征值（3）查询时，先将由关键字生成的token与小集合的特征值进行比对，找到匹配的特征值后再进入相应的小集合进行精确查找。如果所有特征值都不匹配则返回空值。

假设当前系统中有m个文件，每个文件平均有n个关键字。那么原始加密搜索算法的最坏时间复杂度为O（m\*n）。而我们新提出的算法在最坏情况下的时间复杂度为O（m+n），显著降低了搜索次数提升了搜索性能。

**3、实验环境：**

机器配置如图1.4

|  |  |  |  |
| --- | --- | --- | --- |
| 配置信息 | 操作系统 | 数量 | 服务器还是客户端 |
| CPU：英特尔i7-8700  内存：16GB  显卡：GTX-1060-3G | Win10 | 2 | 服务器 |
| CPU：英特尔i5-8250U  内存：8GB  显卡：MX150 | Win10 | 1 | 服务器 |
| CPU：AMD 1700  内存：16GB  显卡：1050ti | Win10 | 1 | 客户端 |

图1.4

实验环境拓扑图如图1.5



图1.5

**4、开发工具及开源框架：** visual studio 2017、Hadoop、SQL Server、Redis、Docker、Wireshark、Chrome。

**5、工作安排**

本课题研究时间为2019年3月到2020年3月。工作进程安排为三个阶段：

前期2019 年3月-4月：

文献阅读、文献整理及开题报告撰写

中期2019年4月—2019年12月：

5月-6月，实现加密搜索模块，完成单元测试，改进搜索算法；

6月－9月，完成加密搜索网盘原型设计，进行集成测试；

9月－11月，对系统进行优化和改进；

11月－12月，开展实验，收集数据并与文献中的相类似的工作进行比较，总结新型系统的优缺点。

后期2019年12月－2020年4月：

整理实验并完成论文初稿；

论文修改；

论文定稿。

参考文献

[1] Peter Mell, Tim Grance. [The NIST definition of cloud computing](http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf)[R].National Institute of Standards and Technology,2011.

[2] Cloud storage[EB/OL].[2019-03-26]. <https://en.wikipedia.org/wiki/Cloud_storage>

[3] Drago I, Mellia M, Munafo MM, Sperotto A, Sadre R, Pras A (2012) Inside Dropbox: understanding personal cloud storage services. In: Proceedings of the 2012 ACM Conference on Internet Measurement Conference, IMC ’12, 481–494, Boston, MA, USA.

[4] Asaf Shabtai, Yuval Elovici, and Lior Rokach. 2012. A Survey of Data Leakage Detection and Prevention Solutions. Springer Science & Business Media.

[5] iCloud leaks of celebrity photos[EB/OL]. [2019-03-26]. <https://en.wikipedia.org/wiki/ICloud_leaks_of_celebrity_photos>

[6] Facebook–Cambridge Analytica data scandal[EB/OL].[2019-04-14]. <https://en.wikipedia.org/wiki/Facebook%E2%80%93Cambridge_Analytica_data_scandal>

[7] Song, Dawn Xiaodong;Wagner, David;Perrig, Adrian. Practical techniques for searches on encrypted data[C]. Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy, p 44-55, 2000;

[8] Delfs, Hans & Knebl, Helmut. Symmetric-key encryption. Introduction to cryptography: principles and applications. Springer. 2007. ISBN 9783540492436.

[9] Public-key cryptography[EB/OL].[2019-04-03]. <https://en.wikipedia.org/wiki/Public-key_cryptography>

[10] Reza Curtmola, Juan Garay, Seny Kamara, Rafail Ostrovsky. Searchable symmetric encryption: Improved definitions and efficient constructions[C]. Journal of Computer Security, 2011, Vol.19 (5)

[11] J. Benaloh and M. de Mare, One-way accumulators: a decentralized alternative to digital signatures, Advances in Cryptology—Eurocrypt’93, LNCS, vol. 765, Springer-Verlag, 1993, pp. 274–285.