**2023年全国大学生信息安全竞赛**

**作品报告**

**作品名称：**

**电子邮箱： marcus.yh.ma@foxmail.com**

**提交日期：**

填写说明

1. 所有参赛项目必须为一个基本完整的设计。作品报告书旨在能够清晰准确地阐述（或图示）该参赛队的参赛项目（或方案）。

2. 作品报告采用A4纸撰写。除标题外，所有内容必需为宋体、小四号字、1.5倍行距。

3. 作品报告中各项目说明文字部分仅供参考，作品报告书撰写完毕后，请删除所有说明文字。(本页不删除)

4. 作品报告模板里已经列的内容仅供参考，作者可以在此基础上增加内容或对文档结构进行微调。

5. 为保证网评的公平、公正，作品报告中应避免出现作者所在学校、院系和指导教师等泄露身份的信息。

**目 录**

[摘要 1](#_Toc33965493)

[第一章 作品概述 2](#_Toc33965494)

[第二章 作品设计与实现 3](#_Toc33965495)

[第三章 作品测试与分析 4](#_Toc33965496)

[第四章 创新性说明 5](#_Toc33965497)

[第五章 总结 6](#_Toc33965498)

[参考文献 7](#_Toc33965499)

# 摘要

（请简要说明创作本作品之动机、功能、特性、创新处、实用性）

# 第一章 作品概述

（建议包括：背景分析、相关工作、特色描述及应用前景分析等）

## 1.1 项目背景

2011年，Dropbox曾经因为安全漏洞事件而被黑客攻击，用户的电子邮件地址和密码哈希值被窃取；2015年，微软OneDrive在Windows 10系统中自动备份用户的图片和文档时，意外地备份了一些用户的私人文件，这个事件引起了很大的争议；2017年，亚马逊AWS S3存储服务的某个区域存在配置错误，导致数百万客户的数据被曝光……可以看到，随着近年来云计算的发展和普及，越来越多的数据被存储在云端。然而，数据隐私和安全性的问题也随之而来。云存储环境中如果采用最直接的明文存储和检索方式，容易被攻击者入侵云端后获取用户的数据，同时云存储服务商勾结他人导致的数据泄露风险也很高。为了解决这些问题，采用密文存储和检索已成为必然趋势。

上面提到的技术就是可搜索加密技术（Searchable Encryption，SE），它是一种结合密码学原语和信息检索的技术，可以对数据和数据的关键词索引进行加密，从而实现用户能够方便灵活且高效地搜索数据，同时云服务器又能对密文数据本身以及关键词相关信息一无所知。其中，单用户可搜索加密和多用户可搜索加密是两种常见的模型。单用户可搜索加密通常只能由一个用户上传和搜索加密文件。这种模型适用于个人用户或单个组织，需要保护自己数据隐私的场景。例如，一个人使用云存储来备份自己的文件，或者一个小型公司使用云存储来共享内部文件；多用户可搜索加密则允许多个用户上传和搜索加密文件。这种模型适用于多用户共享数据和搜索的场景，例如企业内部文件共享、医疗保健领域的数据共享等。多用户可搜索加密具有更大的使用范围和应用场景，因此在实用性上要优于单用户可搜索加密。

值得注意的是，现有的多用户可搜索加密方案往往存在授权粒度较粗的问题，即只能通过分发不同策略的密钥来实现对用户的访问控制。这种方法虽然能够实现基本的访问控制，但对于细粒度的授权控制则存在较大的局限性。文献[1]提出了一种多用户可搜索加密方案，将每个关键词的搜索和排名视为一个整体，但无法实现细粒度访问控制；文献[2]提出了一种基于密钥策略属性加密（Key Policy Attribute-Based Encryption，KPABE）的多用户可搜索加密方案，将文件和关键词分别加密，然后通过分发不同策略的密钥来实现对用户的访问控制，但无法进行对文件的细粒度授权。因此，基于多用户可搜索加密，实现细粒度的授权访问控制是当今研究的重要方向之一，我们也由此选择了这个题目，开发了云场景下细粒度授权的多用户可搜索加密系统。

（1）先说存储方面，例如百度网盘等，都是明文搜索的，它不安全，然后引出可搜索加密的概念、特点和大致的应用场景，这里可以排比列举一些简短的例子，这时候再说明为什么需要在云存储中使用可搜索加密技术来保护用户数据的隐私。

（2）这里首先要写一下单用户和多用户可搜索加密的区别，然后说明授权的意义在哪里，再写多用户授权的可搜索加密的需求和挑战。注意，要着重写一下安全方面的问题，初步打算分两块，一个是攻击者可能的手段和可能造成的威胁，一个是现有的方案有什么缺陷。攻击者方面最好对应着我们方案的优势来写，即我们的方案能抵御什么样的问题；现有方案的缺陷简单提一下就可以，篇幅要比攻击者方面短，因为后面马上就要写到。

## 1.2 相关工作

①动态层次化授权，有访问控制

②不需要DO时刻在线

③不需要辅助（第三方）授权

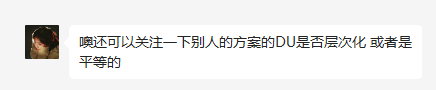
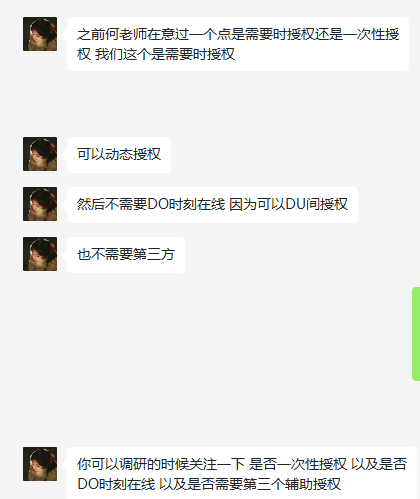
④文件，细粒度

上面是我们的方法的优点，围绕这四点对别人的文献展开全方位攻击。大概选三四篇即可，目前还在调研，很慢，会尽快写的。

此外，多用户也是我们的一个特点（硬说的话也算是吧），所以再列一个单用户的进来。

关注一下别人的方案DU授权是层次化的还是平级的。

最后归纳一下都有什么问题。



## 1.3 项目特色

针对上述问题，我们设计了云场景下细粒度授权的多用户可搜索加密系统。该系统基于可搜索加密技术，在具备了多用户功能的同时，利用***某数据结构（后面问问，忘了叫什么了，好像也没名字）***完成了数据拥有者对用户的、用户之间的以文件为粒度的授权，实现了用户仅能在由数据拥有者授权访问的加密数据中进行关键词搜索，无法获取未授权数据的任何信息，云服务器也无法获取外包数据明文以及查询关键词。综上，该系统（名字还没想好）具有以下特色：（每一条的描述还没写，因为万一被老师认为这几条不够恰当呢（狗头））

①多用户

②动态层次化授权，有访问控制

③不需要DO时刻在线

④不需要辅助（第三方）授权

⑤搜索文件，细粒度

## 1.4 应用前景

随着云计算技术的不断发展，本文提出的细粒度授权的多用户可搜索加密系统在很多领域具有广泛的应用前景。通过该系统，可以在保证数据安全和用户隐私的前提下，实现多用户之间的高效数据共享和搜索。

（1）企业数据共享和协作：在企业内部，员工之间需要共享大量的文件和信息。这些文件可能包含敏感的商业秘密、员工个人信息等，因此在共享过程中要求严格的访问控制。本系统可以为企业提供细粒度的访问控制，使得员工仅能访问到授权范围内的文件，同时避免了未经授权的数据泄露。此外，通过对关键词的访问控制，可以实现文件内容的局部共享，有助于提高企业内部的工作效率。

（2）医疗信息共享：医疗行业涉及大量患者的敏感信息，如病历、诊断报告等。在医生之间、医院之间甚至跨国共享这些信息时，必须保证患者隐私的安全。通过本系统，可以实现对医疗数据的细粒度授权和访问控制，确保只有获得授权的医生能够访问患者的相关信息。同时，可搜索加密技术使得医生可以在保持数据机密性的前提下，快速检索到所需的患者资料，提高医疗服务质量。

（3）在线教育平台：在线教育平台上，教师需要向不同学生共享课件、作业等教学资料。本系统可以实现对这些资料的细粒度授权，确保学生只能访问到自己所需的教学内容。此外，通过可搜索加密技术，学生可以方便地搜索到相关课件和资料，提高学习效率。同时，在线教育平台可以在不泄露教学内容的前提下，对平台内的数据进行统计分析，为教育改革和发展提供有力支持。

（4）智能家居数据共享：智能家居设备的普及使得家庭成员之间需要共享大量的设备数据，如监控视频、环境传感器数据等。本系统可以实现对这些数据的细粒度授权和访问控制，保证家庭成员在保持隐私的前提下，共享和使用这些数据。例如，父母可以授权子女查看家中的监控视频，而不允许查看其他与隐私相关的信息。同时，通过可搜索加密技术，家庭成员可以快速检索到所需的数据，提高生活便利性。

（5）法律行业信息共享：法律行业涉及大量敏感的案件文件和客户信息，律师在与同事或合作伙伴共享这些信息时，需要严格的访问控制。通过本系统，可以实现对案件文件和客户信息的细粒度授权，确保只有获得授权的律师能够访问相关信息。可搜索加密技术使律师能够在保持数据机密性的前提下，快速检索到所需的案件资料，提高工作效率。

（6）科研数据共享：在科学研究领域，研究者需要与合作者共享大量的实验数据、论文等资料。本系统可以实现对这些资料的细粒度授权，确保研究者只能访问到自己所需的数据，防止未经授权的数据泄露。此外，可搜索加密技术使研究者可以在保持数据机密性的前提下，快速检索到相关资料，促进科研合作与交流。

综上所述，本文提出的细粒度授权的多用户可搜索加密系统具有广泛的应用前景，可以为不同行业提供安全、高效的数据共享和搜索服务。在当前对数据安全和隐私保护日益重视的背景下，该系统将在实践中发挥重要作用，推动云计算技术在各领域的应用与发展。

# 第二章 作品设计与实现

（建议包括系统方案、实现原理、硬件框图、软件流程、功能、指标等）

2.1 系统设计

*（1）一小段，直接说清楚它是在什么场景下、有着什么样的功能、会有怎样的效果。*

*（2）一段，讲解一下设计思路，我认为应该包括授权与回收（可以搭配图片）、可搜索加密。*

*（3）一段，讲解一下工作流程，画个图，讲清楚过程。*

*（4）可以从****系统模型****开始写，包含哪些实体，实体之间如何交互？对每个实体有功能介绍？*

*核心设计思想：阐述系统的基本设计原则和关键技术。*

*系统架构：描述系统的主要组成部分和它们之间的关系。*

*功能模块：说明系统中各个功能模块的作用及其实现方法。*

*工作流程：描述系统在实际应用中的主要操作流程和步骤。*

*这里只是大概写了一下，后面会组内讨论后写得更加正确和具体*

（1）核心设计思想：本系统旨在实现云场景下的细粒度授权和多用户可搜索加密，保证用户数据的机密性和可用性，同时简化用户之间的数据共享和搜索过程。

（2）系统架构：本系统主要由三个组成部分构成，分别是数据拥有者、数据查询者和云服务器。数据拥有者将加密数据外包给云服务器，数据查询者在获得授权后，可以在云服务器上搜索加密数据。

（3）功能模块：

①加密模块：负责对数据拥有者的原始数据进行加密，生成加密数据以供外包至云服务器。

②授权模块：实现数据拥有者对数据查询者的细粒度授权，包括对文件和关键词的访问控制。

③搜索模块：负责在云服务器上对加密数据进行关键词搜索，使得数据查询者能够在保持数据机密性的前提下获取所需信息。

④解密模块：对搜索结果进行解密，供数据查询者查看。

（4）工作流程：

数据拥有者将原始数据通过加密模块进行加密，然后将加密数据外包给云服务器。

数据拥有者通过授权模块为数据查询者分发授权信息，实现对文件和关键词的细粒度授权。

数据查询者在获得授权后，通过搜索模块在云服务器上进行关键词搜索，云服务器返回搜索结果。

数据查询者使用解密模块对搜索结果进行解密，获取明文信息。

通过以上设计，本系统可以实现云场景中细粒度授权的多用户可搜索加密功能，满足用户对数据安全和隐私保护的需求，同时提高数据共享和搜索的效率。

2.2 形式化

蹲蹲学姐写的

2.3 实现原理

把算法写清楚，原理细节讲清楚

此外不知道用不用把整体架构写一写

|  |
| --- |
| Algorithm 1 |
| Data Owner  1. Randomly select  2. Randomly select three master keys  3. Initialize  4.  5.  6. Send  to the Server |

|  |
| --- |
| Algorithm 2 |
| Data Owner  1. Parse  2. Initialize  3. For every document  4. Set  5. For every keyword  6. Set  7. Set  8. Update  9. Randomly permute the tuple-entries of  10. Send  to the Server |

|  |
| --- |
| Algorithm 3 |
| Data Owner  1. Randomly select two user’s keys  2. Initialize  3.  4.  5. Send  to the Data User |

|  |
| --- |
| Algorithm 4 |
| Data Owner  1. Parse  2. Initialize  3. For every document Data Owner authorize Data User to access  4. Set  5. Set  6. Set  7. Set  8. Update  9. Send  to the Data User  10. Send  to the Server  Data User  1. Parse  2. Update  3.  Server  1. Parse  2. Update  3. |

|  |
| --- |
| Algorithm 5 |
| Data User A  1. Parse  2. Initialize  3. Set  4. Set  5. Set  6. For every document Data User A authorize Data User B to access  7. Initialize  8. If  then  9. Set  10. Set  11. Set  12. Else if  then  13. Set  14. Update  15. Set  16. Set  17. Update  18. Send  to the Data User B  19. Send  to the Server  Data User B  1. Parse  2. Update  3. Update  4.  Server  1. Parse  2. Update  3. |

|  |
| --- |
| Algorithm 6 |
| Data User  1. Parse  2. Initialize  3. For every document key information  4. If  then  5. Set  6. Set  7. Initialize  8. Else if  then  9. Set  10. Update  11. Randomly permute the tuple-entries of  12. Sent  to the Server  Server  1. Parse  2. Initialize  3. For every query tuple  4. Set  5. If  then  6. For every  7. Set  8. If  then  9.  10. Send  to the Data User  Data User: Final Output  1. Initialize  2. For every encrypted document identifier result  3. Recover  4. Update  5. Output |

2.3 硬件框图和软件流程

这里分模块或者分部来用图表示一下，并且在图后面加以详细的解释

2.4 功能与指标

着重把做出来的效果呈现出来

# 第三章 作品测试与分析

（建议包括测试方案、测试环境搭建、测试设备、测试数据、结果分析等）

测试内容？

# 第四章 创新性说明

（本部分内容主要说明作品的创新性）

这个很重要

# 第五章 总结

现有+规划

# 参考文献

[1] Gurjar S P S, Pasupuleti S K. A privacy-preserving multi-keyword ranked search scheme over encrypted cloud data using MIR-tree[C]//2016 International Conference on Computing, Analytics and Security Trends (CAST). IEEE, 2016: 533-538.