**论文题目：基于AES的可搜索加密方案设计与应用**

**摘要**

随着数字化进程推进，数据安全与隐私保护至关重要。传统加密方式在数据检索时面临风险，云计算与大数据环境也对加密数据搜索提出新需求。本课程设计基于AES算法，利用OCB模式加密文件、用CBC模式和HMAC加密关键词生成陷门关键词以及用HMAC保障数据完整性，实现了上传与检索文件功能。系统测试表明，功能运行正常，生成陷门关键字与匹配算法性能良好。该方案有效解决加密数据搜索问题，提升数据安全性与可用性，具有一定实用价值与可扩展性。

**关键词**：可加密搜索；数据安全；AES；HMAC

**abstract**

With the advancement of digitalization, data security and privacy protection are crucial. Traditional encryption methods face risks in data retrieval, and cloud computing and big data environments also pose new demands for encrypted data search. This course design is based on the AES algorithm, using OCB mode to encrypt files, CBC mode and IASC to encrypt keywords to generate trapdoor keywords, and using IASC to ensure data integrity, achieving the functions of uploading and retrieving files. System testing shows that the function runs normally, and the performance of generating trapdoor keywords and matching algorithms is good. This solution effectively solves the problem of encrypted data search, improves data security and availability, and has certain practical value and scalability.

**Keywords:** encrypted search；Data security；AES；HMAC

**目录**

[1 课设背景 5](#_Toc25164)

[1.1 设计背景 5](#_Toc695)

[1.2 课设要求的实现 5](#_Toc28532)

[1.3 课设内容 6](#_Toc21491)

[1.4 课设功能 6](#_Toc10302)

[1.4.1 上传文件 6](#_Toc24628)

[1.4.2 检索文件 6](#_Toc11074)

[2 实现方案 7](#_Toc31994)

[2.1 可加密搜索流程 7](#_Toc3097)

[2.1.1 生成陷门关键词 7](#_Toc15123)

[2.1.2 对陷门关键词检索匹配 8](#_Toc7572)

[2.2 可加密搜索平台流程 9](#_Toc23161)

[2.2.1 上传文件 9](#_Toc7805)

[2.2.2 检索文件 10](#_Toc10886)

[3 系统测试 11](#_Toc26017)

[3.1 运行结果 11](#_Toc8283)

[3.1.1 功能选择 11](#_Toc20217)

[3.1.2 上传文件 11](#_Toc32237)

[3.1.3 检索文件 12](#_Toc31516)

[3.2 性能测试 13](#_Toc30233)

[3.2.1 对生成陷门关键字进行测速 13](#_Toc15218)

[3.2.2 对匹配算法进行测速 13](#_Toc24231)

[4 总结及展望 13](#_Toc6327)

[4.1 总结 13](#_Toc18222)

[4.2 展望 14](#_Toc3785)

[参考文献 15](#_Toc17448)

# 课设背景

## 设计背景

在数字化进程加速的当下，数据安全与隐私保护的重要性愈发凸显。一方面，随着各类敏感信息的电子存储与传输愈发频繁，数据泄露事件频发，人们对数据安全和隐私的关注度不断攀升。传统加密方式虽能保障数据存储和传输时的保密性，但在数据检索时需解密，这使数据面临风险，可加密搜索方案因此应运而生。

另一方面，云计算与大数据环境蓬勃发展，大量数据存储于云端服务器，数据所有者对数据物理存储介质失去直接控制，同时大数据需要高效检索机制。而传统加密技术在加密数据搜索方面存在局限，无法满足需求。基于AES的可加密搜索方案恰能解决这些问题，用户可在不信任云服务提供商的情况下对加密数据进行搜索，既利用了云计算资源，又保障了数据安全。此外，众多行业受严格法规和标准约束，如医疗保健、金融等行业。这些法规要求企业保障客户数据在处理和检索过程中的安全，基于AES的可加密搜索方案有助于企业满足合规性要求，避免违规带来的严重后果。综上，基于AES的可加密搜索方案的研究具有重要意义。

## 课设要求的实现

1. 在可搜索加密方案上，通过密钥生成的iv和key来使用AES的OCB模式加密存储文件，保障了数据的机密性；用加密后的信息通过陷门进行关键词检索保障搜索的隐私性；用HMAC保障了数据的完整性。
2. 在方案的正确性和安全性上，使用AES算法结合HMAC和伪随机函数构建了安全的陷门，实现了基于陷门的可搜索加密方案。确保了方案的正确性和安全性。
3. 该方案中，存储数据用AES的OBC模式加密保障了数据机密性；通过陷门函数对关键词的加密实现了CPA攻击下的语义安全（加密时参入了随机值，即使是相同密钥和关键词的情况下生成的陷门关键词也不相同），以此实现了查询隔离，服务器在执行搜索时无法得知被搜索的关键词以及对应文件的内容保障了搜索隐私性；通过HMAC保障了数据完整性。潜在漏洞与攻击方式上，如果私人密钥泄露，则会导致数据机密性完全丧失，还有侧信道攻击也有可能获取敏感信息或者密钥。在关键词提取上，除了人为添加关键词，还可以通过布隆过滤器或者机器学习来提取关键词。
4. 在搜索和用户体验上，通过建立关键词和加密后文件名的索引连接，实现通过关键词快速检索匹配加密后的文件名，再通过加密后的文件名匹配存储的加密文件。
5. 通过多次对文件的关键词进行检索验证了该方案检索的准确性并求出了其检索效率，实现了性能评估。在检索时无需明文关键词保证了其安全性。这个方案可以用在多种云存储的场景中，体现了其较高的可扩展性

## 课设内容

本课设是基于Practical Techniques for Searches on Encrypted Data论文[1]中的可加密搜索方案实现的，在这篇论文的基础上做出了修改和创新。本课设使用论文中的可加密搜索方案对上传文件的关键词进行处理，并将得到的陷门关键词和对应的文件建立索引以此来实现可加密搜索，同时对密文生成HMAC以保护数据的完整性。

## 课设功能

本课设实现了两个功能，分别是上传文件和检索文件。

### 上传文件

上传文件时，可以上传各种文件，然后输入个人密钥和该文件对应的关键词，关键词不超过五个，每个关键词的中文长度不超过15。上传文件功能实现了对各种文件的加密上传存储，并保留其陷门关键词和文件名建立的索引，方便后面的搜索

### 检索文件

检索文件时，输入关键词和个人密钥进行检索文件，如果该关键词有对应的文件时，则显示出检索的文件名，同时提供下载文件的链接，如果上传的文件被篡改过则显示文件被篡改

# 实现方案

## 可加密搜索流程

### 生成陷门关键词

首先对文章的关键字进行填充到48字节，不足48字节时填充0补齐。用AES的CBC模式对填充后的关键词进行加密得到（长度为48字节），加密密钥为hashkey（长度为16个字符，保密），将划分为左右两部分(长度为16个字符)和（长度为32个字符）。使用HMAC-SHA256以hashkey为固定密钥对进行加密得到新的密钥（长度为32个字符）。使用为随机函数Random，输入随机种子seed，得到伪随机序列（长度为16个字符）。使用HMAC-SHA256以作为密钥对进行加密得到（长度为32个字符）。将和拼接起来得到，最后用和异或得到陷门关键词，最后将上传到不可信服务器即可。关键代码，如图1所示

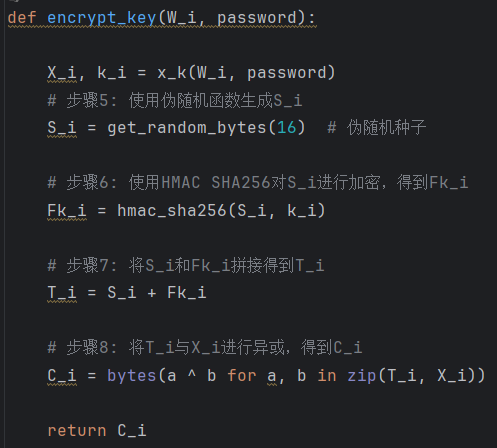


图 1

生成陷门关键词的流程如图2所示

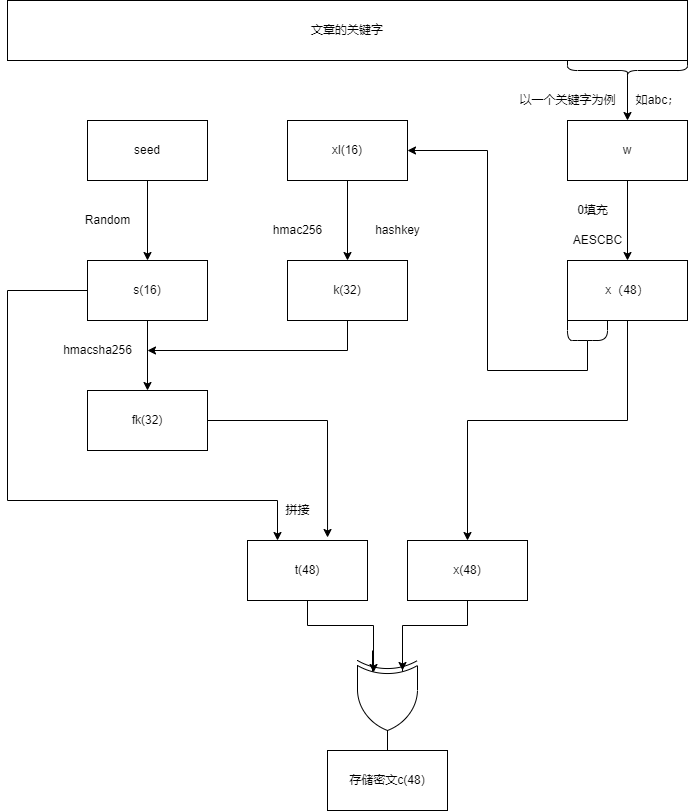


图 2

### 对陷门关键词检索匹配

在检索时，用户需将关键词对应的和上传给服务器进行检索，生成和的方式和生成陷门关键词的过程相同。服务器在得到和后，先计算和异或得到，再将划分成（16字节）和（32字节），最后用相同的HMAC-SHA256函数以为密钥加密，用加密结果和做对比，如果值相等，则检索成功，找到对应的关键词。关键代码如图3所示

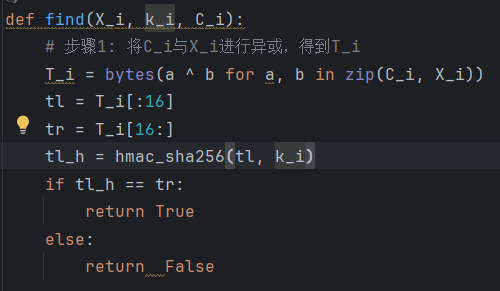


图 3

匹配关键词的流程如图4所示

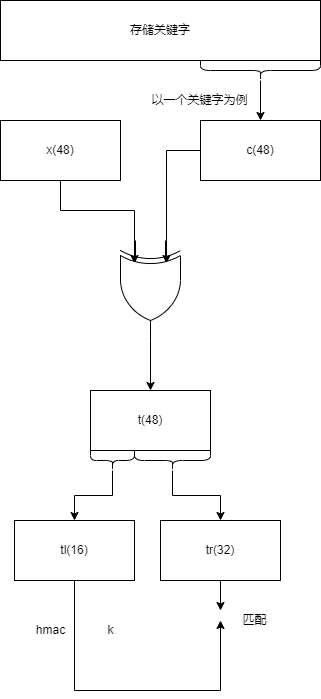


图 4

## 可加密搜索平台流程

### 上传文件

上传文件后输入个人密钥和关键词（添加的关键词不超过5个，每个关键词中文长度不超过15个字，每个关键词之间用空格间隔）。点击上传文件后，客户端用陷门函数处理关键词生成陷门关键词，将陷门关键词和对应文件加密后的文件名建立索引并将上传的原文件加密，同时对加密后的密文求HMAC（使用的密钥和前面生成陷门的密钥不同）拼接到密文后面，最后将建立的索引和生成的密文文件（密文文件的文件名是原文件加密后的文件名）上传到云端服务器。如图5所示。

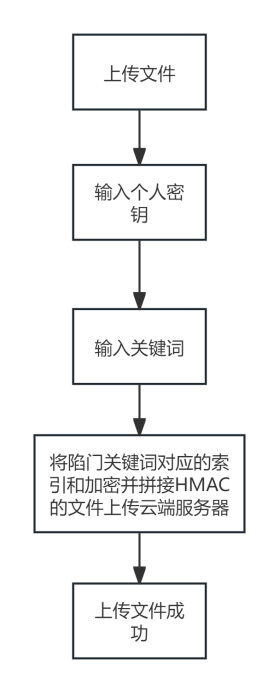


图 5

### 检索文件

输入检索文件对应的关键词和个人密钥进行检索文件，客户端会将关键词和个人密钥进行加密处理后再上传至云端服务器进行检索匹配，检索该关键词及密钥对应的所有文件并对检索到的所有文件进行数据完整性检验，通过对拼接在密文后的HMAC-SHA256值进行检验来验证数据的完整性。如果数据被篡改过，在显示检索结果时会将检索出的文件名用红色字体标出并在后面标注文件被篡改。如果文件正常，则可以通过点击检索出来的文件名下载原文件。如图6所示

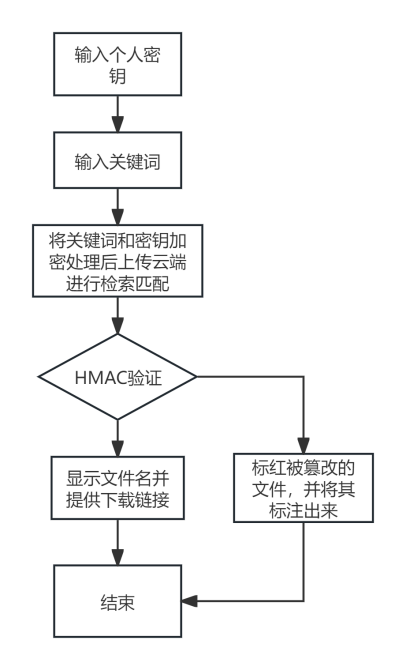


图 6

# 系统测试

## 运行结果

### 功能选择

在主页面选择功能：上传文件或检索文件，如图7所示



图 7

### 上传文件

上传文件后输入个人密钥和关键词，如图8所示



图 8

上传成功，如图9所示



图 9

### 检索文件

输入关键词和个人密钥，如图10所示



图 10

得到该关键词以及密钥对应的所有文件，并且如果文件被篡改过则会标红文件名并给出提示，同时可以下载正常的文件，如图11所示



图 11

## 性能测试

### 对生成陷门关键字进行测速

通过取随机长度的关键词和密码，对其生成陷门关键词的耗时进行重复1000次测速，取其平均值，得到其耗时总和0.132s和单次耗时0.00013s的结果，如图12所示



图 12

### 对匹配算法进行测速

建立1000对关键词索引，依次对每个关键词进行检索，得到其耗时的总和25.4s以及平均耗时0.0254s，如图13所示



图 13

# 总结及展望

## 总结

本课程设计成功实现了基于 AES 的可加密搜索方案，达到了预期目标。通过一系列技术手段，在保障数据机密性、搜索隐私性和结果完整性方面取得了良好效果。

在功能实现上，上传文件功能允许用户上传多种类型文件，并能准确提取不超过五个且中文长度不超过 15 个字的关键词，同时对文件加密存储并建立索引，为后续检索提供了便利。检索文件功能可根据用户输入的关键词和密钥，在加密数据集中进行高效检索，对检索到的文件进行完整性验证，并能准确提示文件是否被篡改，同时提供正常文件的下载链接。

在性能测试方面，对生成陷门关键字和匹配算法的测速结果显示，方案在处理效率上表现尚可。生成陷门关键字的平均耗时较短，表明系统在处理关键词加密时有一定的速度优势，能够满足一定规模的数据处理需求。匹配算法单次耗时同样相对较短，能够满足一定规模的数据处理需求

## 展望

在关键词生成方面，计划引入布隆过滤器或机器学习技术。布隆过滤器可以高效地判断一个元素是否在一个集合中，在关键词提取时可用于快速过滤掉重复或不相关的词汇，提高关键词的准确性和有效性。机器学习方法则可通过对大量文本数据的学习，自动识别出更具代表性和相关性的关键词，减少人工添加关键词的工作量，同时可能提高搜索的召回率和精确率。

对于索引表结构，将进一步优化。目前的索引结构在处理大规模数据和复杂查询时可能效率不高，计划采用更先进的数据结构，如基于哈希或树状结构的变种，以加快关键词匹配速度。例如，使用 B + 树或哈希索引结合的方式，既能利用哈希的快速查找特性，又能借助 B + 树的有序性和范围查询能力，提高索引的整体性能。

在安全性方面，除了关注私人密钥泄露和侧信道攻击问题，还将研究更高级的加密算法和密钥管理策略。探索量子加密等新兴技术在可加密搜索中的应用可能性，以应对未来可能出现的更强大的计算能力攻击。同时，加强对系统的访问控制和审计机制，确保数据在整个生命周期内的安全性。

未来还将考虑将该方案拓展到更多的应用场景，如分布式存储系统（如IPFS等）、移动设备数据加密搜索等，进一步提高方案的通用性和实用性，为数据安全领域提供更强大的解决方

**参考文献**

1. Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data[C]//Proceeding 2000 IEEE symposium on security and privacy. S&P 2000. IEEE, 2000: 44-55.
2. Bellare M. Practice-oriented provable-security[C]//International workshop on information security. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997: 221-231.
3. Bellare M, Desai A, Jokipii E, et al. A concrete security treatment of symmetric encryption[C]//Proceedings 38th Annual Symposium on Foundations of Computer Science. IEEE, 1997: 394-403.
4. Simmonds P. SSE: a nucleotide and amino acid sequence analysis platform[J]. BMC research notes, 2012, 5: 1-10.
5. Chen L, Xue Y, Mu Y, et al. Case-sse: Context-aware semantically extensible searchable symmetric encryption for encrypted cloud data[J]. IEEE Transactions on Services Computing, 2022, 16(2): 1011-1022.
6. Münch J P, Schneider T, Yalame H. Vasa: Vector aes instructions for security applications[C]//Proceedings of the 37th Annual Computer Security Applications Conference. 2021: 131-145.
7. Mendonca S N. Data security in cloud using AES[J]. Int. J. Eng. Res. Technol, 2018, 7.
8. Chu D, Zhu K, Cai Q, et al. Secure cryptography infrastructures in the cloud[C]//2019 IEEE Global Communications Conference (GLOBECOM). IEEE, 2019: 1-7.
9. Bost R, Minaud B, Ohrimenko O. Forward and backward private searchable encryption from constrained cryptographic primitives[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017: 1465-1482.
10. Kamara S, Lauter K. Cryptographic cloud storage[C]//International Conference on Financial Cryptography and Data Security. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010: 136-149.