申请上海交通大学硕士学位论文

抗搜索模式信息泄漏的同义词对称可搜索加密技术的研究

论文作者 柳祚鹏

学　　号 1120339063

指导教师 谷教授

专　　业 计算机科学与技术

答辩日期 2010 年 1 月 16 日

Submitted in total fulfilment of the requirements for the degree of Master in Physics

XeLaTeX/LaTeX Template for SJTU Master Degree Thesis v0.5.2

Si Li

Supervisor

Prof. San Zhang

Depart of XXX, School of XXX

Shanghai Jiao Tong University

Shanghai, P.R.China

Jan. 16th, 2010

上海交通大学
学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：_____

日　期：_____年 ____月 ____日

上海交通大学
学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保　密 □，在 _____ 年解密后适用本授权书。

本学位论文属于

不保密 □。

（请在以上方框内打“✓”）

学位论文作者签名：_____　　指导教师签名：_____

日　期：_____年 ___月 ___日　　日　期：_____年 ___月 ___日

抗搜索模式信息泄漏的同义词对称可搜索加密技术的研究

摘　　要

随着云计算的飞速发展，越来越多的用户将敏感数据集中存放到云服务端，以避免繁琐的本地管理并获取更多便捷的服务。与此同时由于云服务提供商并非完全可信，如何保证云端数据的隐私—即如何避免云端数据被未经授权的人所访问已成为云计算环境下安全隐私问题的研究重点。为此，一些学者率先提出了“云计算环境下的可搜索加密技术”的研究课题。可搜索对称加密即用户将数据加密存储到服务端，同时服务端维持用户对加密数据进行有效搜索的能力。

本课题研究重点是：对称可搜索加密技术（可搜索加密技术主要包括：对称可搜索加密技术、公钥可搜索加密技术和隐私信息检索）中模糊对称可搜索加密和动态可搜索加密技术。文中首先分析了现有方案的研究状况；然后找出现有方案中存在的不足和被忽略的问题；最后通过解决原有方案中存在的潜在问题和扩充原有模糊方案的功能—支持语义搜索，提出一种更实用、更高效、功能更强大并且没有降低安全性能的新方案，使之尽可能支持动态和模糊功能并且没有增加信息泄露。

关键词：可搜索加密　同义词搜索　相似搜索　搜索模式

XeTeX/LaTeX Template for SJTU Master Degree Thesis v0.5.2

ABSTRACT

An imperial edict issued in 1896 by Emperor Guangxu, established Nanyang Public School in Shanghai. The normal school, school of foreign studies, middle school and a high school were established. Sheng Xuanhuai, the person responsible for proposing the idea to the emperor, became the first president and is regarded as the founder of the university.

During the 1930s, the university gained a reputation of nurturing top engineers. After the foundation of People's Republic, some faculties were transferred to other universities. A significant amount of its faculty were sent in 1956, by the national government, to Xi'an to help build up Xi'an Jiao Tong University in western China. Afterwards, the school was officially renamed Shanghai Jiao Tong University.

Since the reform and opening up policy in China, SJTU has taken the lead in management reform of institutions for higher education, regaining its vigor and vitality with an unprecedented momentum of growth. SJTU includes five beautiful campuses, Xuhui, Minhang, Luwan Qibao, and Fahua, taking up an area of about 3,225,833 m2. A number of disciplines have been advancing towards the top echelon internationally, and a batch of burgeoning branches of learning have taken an important position domestically.

Today SJTU has 31 schools (departments), 63 undergraduate programs, 250 masters-degree programs, 203 Ph.D. programs, 28 post-doctorate programs, and 11 state key laboratories and national engineering research centers.

SJTU boasts a large number of famous scientists and professors, including 35 academics of the Academy of Sciences and Academy of Engineering, 95 accredited professors and chair professors of the "Cheung Kong Scholars Program" and more

than 2,000 professors and associate professors.

Its total enrollment of students amounts to 35,929, of which 1,564 are international students. There are 16,802 undergraduates, and 17,563 masters and Ph.D. candidates. After more than a century of operation, Jiao Tong University has inherited the old tradition of ”high starting points, solid foundation, strict requirements and extensive practice.” Students from SJTU have won top prizes in various competitions, including ACM International Collegiate Programming Contest, International Mathematical Contest in Modeling and Electronics Design Contests. Famous alumni include Jiang Zemin, Lu Dingyi, Ding Guangen, Wang Daohan, Qian Xuesen, Wu Wenjun, Zou Taofen, Mao Yisheng, Cai Er, Huang Yanpei, Shao Lizi, Wang An and many more. More than 200 of the academics of the Chinese Academy of Sciences and Chinese Academy of Engineering are alumni of Jiao Tong University.

KEY WORDS: SJTU, master thesis, XeTeX/LaTeX template

目　　录

表格索引

插图索引

主要符号对照表

ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率
ϵ	介电常数
μ	磁导率

第一章 绪论

1.1 研究背景及意义

随着网络技术的飞速发展，人们生活中的点点滴滴（从办公、娱乐到洗衣、做饭）逐渐网络化，致使网络中数据的量级呈指数增长。在这些数据的面前，先前的计算技术和存储能力显得有些力不从心。在这样的背景下，一种全新的网络技术油然而生 — 即“云存储”和“云计算”技术。他们的出现分别从理论上解决了大数据的存储和计算问题。为此，实际中各大企业纷纷提出不同场景下的“云”解决方案以满足各自的需求。亚马逊（Amazon）率先推出弹性计算云（Elastic Compute Cloud — EC2）服务 [1]；Google 首席执行官埃里克 · 施密特（Eric Schmidt）在搜索引擎大会首次提出“云计算”（Cloud Computing）[2] 的概念；随后包括 IBM、Microsoft、Intel、Apple 和 YAHOO 等一些大型企业都开始部署自己的云存储和计算平台。由此，网络计算和存储的发展正式进入“云”的世界。

“云计算”是一种基于互联网的计算方式，按照这种方式，云终端将已部署的资源 (如网络、服务器、存储、应用及服务) 按需提供给云用户, 并最大程度地减少用户对资源的管理和配置 [3]。通过互联网，云计算为使用者提供强大的可伸缩和廉价的分布式计算能力，并且在使用过程中，使用者不需要了解云端基础技术的细节和具备相关的专业知识，就能对云端资源进行合理的控制和使用。根据美国国家标准与技术研究院 (NIST) 的定义 [4]，云计算提供三种不同层次的服务：

1. 软件即服务（SaaS）：用户可以通过租借云平台上的软件来为自己提供服务或无偿使用一些基础服务软件；

2. 平台即服务（PaaS）：用户利用云计算服务提供商的平台，通过免费或低价租借的方式来部署自己的软件；

3. 基础设施即服务（IaaS）：云用户可以利用云平台基础设施来获得所需服务。

根据美国国家标准和技术研究院的定义，云计算模型按照部署方式主要包括公有云、私有云、社群云和混合云。公有云通过网络及第三方服务提供给用户使用；私有云具有许多公有云环境下的优点（如弹性，实时提供服务），两者差别主要在于，私有云资源仅需在组织内部管理和使用，不会受到网络带宽和安全疑虑的影响；社群云主要由许多利益相仿的组织掌握和使用，社群内成员可以使用共有的资源，避免了公有云开放环境的安全问题；混合云则基于经济性、可用性等的考虑，是公有云和私有云的结合。云计算部署模型从数据隐私方面的角度来考虑，具备以下特点：

- 在公有云环境下，由于数据拥有者与服务提供商站在不同的立场上并且拥有不同的利益，使得服务提供商并不被完全信任（semi-trust）；
- 在私有云环境下，虽然云资源仅在组织内部使用，但由于云计算普遍采用虚拟化技术 [5] 来提高系统的资源利用率来降低运营成本，不同用户的数据可以同时在同一物理服务器上进行计算和存储。跨虚拟机攻击 [6] 使得用户数据有可能被同一物理服务器上的其他用户非授权访问。

由于“云”计算提供廉价的计算和强大的可伸缩，越来越多的用户将本地的数据移植到“云”平台。与此同时，一些云安全事故也频频发生（e.g.2011 年谷歌邮箱爆发大规模的用户数据泄漏事件，2014 年 Apple 公司 icloud 帐号泄漏事件）。在透明的云环境下，用户简单地将数据以明文形式存储在云服务端存在明显的缺陷 — 数据安全隐患。因此，当数据拥有者将数据存储到云服务端后，如何保护云端数据隐私，避免云端数据被未经授权用户所访问成为云端数据隐私安全研究的焦点。

为了解决上述安全问题，最简单的做法就是用户在将其数据上传至非可信服务器之前，先进行数据加密；当需要使用数据时，从服务器下载密文数据并进行解密。这种做法在解决数据隐私性问题的同时，带来了新的数据可用性问题 — 如何对服务器上的数据进行有效地搜索。由于数据处于加密状态，服务器无法解密，在传统安全机制下，云用户只能下载所有所需的数据，解密之后进行搜索。这种做法将耗费大量的网络资源，且并没有充分利用云端强大的计算能力。

为了解决这样的安全隐患，一些学者提出使用传统的安全机制解决了数据隐私泄漏问题，具体过程如下：数据拥有者在将数据存储到并非完全可信的服

务端之前，首先将数据加密 [7], 然后将数据以密文的形式存储到云服务端；当用户需要使用数据时，首先将存储在服务器端的加密数据全部下载下来，然后进行解密并查找获得所需的文件。该机制虽然解决了数据的隐私问题，同时带来了新的问题 — 如何保证云存储和计算资源的合理利用。这种做法将不仅耗费大量的网络资源，同时也增加了用户的计算开销，这些都与云计算服务的设计理念和用户需求相违背。因此，如何高效利用云端的存储和计算能力是当前关注的重点。

为此，在“云”计算平台下一种新的确保数据隐私和具备高效计算和存储技术被提出 — 可搜索加密技术。可搜索加密技术解决了上述难点，其过程描述如下：数据拥有者将文档进行可搜索加密并存储到云服务平台；当授权的用户需要查找文档时，使用搜索条件陷门（Trapdoor）生成算法将单词的陷门提交至云服务端；一旦云服务端收到搜索陷门后，便直接在用户的密文数据上进行搜索操作，并将搜索结果返回给用户。在方案整个过程中，云服务器无法知晓用户数据以及搜索条件的任何信息，保证了数据隐私性；仅符合搜索条件的用户数据会被发送，而不是所有的用户数据，大大降低了网络资源的消耗。

可搜索加密技术包括对称可搜索加密 (Symmetric searchable encryption)、公钥可搜索加密 (Public key encryption with keyword search) 和多用户 (Multi-User) 可搜索加密技术。近年来，这些技术都得到的广泛的关注。它们的研究领域极其广泛，包括：单关键字搜索、多关键字搜索、模糊搜索、范围搜索、布尔搜索搜索和动态搜索等。这些方案试图提供一个完备的 —高性能、强安全解决方案。

相似搜索（包括模糊搜索和同义词搜索 [8]）在明文场景下具有广泛的应用，虽然这些技术在加密数据上已得到广泛的关注，但是其研究成果尚有不足。目前，人们仅仅关注模糊搜索却忽略了同义词搜索的情形，其实不然它在实际应用中有着极大的需求，因而提出具备强安全性的相似搜索技术对理论和实际具有远大的意义。同样目前可搜索环境下的诸多方案都存在信息泄漏 — 包括大小模式（size pattern）、访问模式（access pattern）和搜索模式（search pattern），这些信息的泄漏将大大降低数据的安全，因而如何避免或降低这些信息的泄漏也具有强大的显示意义。因此，探究实用的相似可搜索方案和减小数据在云计算可搜索场景下的泄漏，对安全云计算的普及具有重要的意义。

1.2 研究现状及相关问题

D.Song 在 [9] 中最早给出了一种对称可搜索加密方案，方案使用传统的加密算法对文档中的每个单词单独进行加密来对文档进行数据保护。在该方案中，主要缺点是搜索效率低下，同时也泄漏了文章中单词的位置和出现的频率等信息。为了加快检索，此后所有的可搜索加密方案都是基于索引的技术。Goh[10] 是第一篇利用安全索引技术解决可搜索加密问题的文档。该方案将每篇文档中的单词映射到一个 Bloom Filter，以此作为检索判断条件，这样大大提高了方案的索引建立和搜索时间，但是该方案却引进了误报率，且误报率和安全索引的大小成反比关系。文章 [11] 对有误报率的方案提出了改进技术。其主要通过对每个单词建立一个映射位，以此来判断文档是否包含某单词。但方案因引入了词典的存储而大大增加了用户或服务端的存储开销。论文 [12] 第一次基于 Inverted-index 提出了具有强安全性的可搜索加密方案，并给出了具体的安全性定义和严格的安全证明过程。Curtmola 方案中的主要优势是索引结构基于反向索引，极大地减少了服务器端的搜索时间（O(1) 的查找时间），同时提出了具有 Adaptive 安全性的方案，但是方案不支持复杂的条件搜索和文章的动态修改。与此同时，Van 基于 inverted-index 提出了另一种形式的安全索引方案，方案使用一个加密的二元数组维护安全索引信息，数组两维分别表示单词和文档 ID，数组元素为 1 表示对应的文档包含对应的单词。搜索时解密二维数组中对应单词所在的行，即可知晓包含该单词的文档 ID。

当前绝大部分可搜索加密方案假定服务器是诚实但好奇 (honest-but-curious) 的，即服务器严格遵守方案的算法和流程，但是会出于好奇，在用户搜索过程中偷偷分析文档的信息，希望了解用户的隐私信息；并基于这种假设，证明方案是有效的和安全的。而实际中，若服务器不遵守规定，如将篡改搜索结果，把某些文档从搜索结果中删去，而客户端却无法知晓这些篡改情况。Chai 在 [13] 中给出了一个支持搜索结果验证的对称可搜索加密方案，可以知晓服务器是否遵守方案协定。K. Kurosawa 在 [14] 中也给出了一个支持搜索结果验证的方案，该方案可以让客户端精确知晓服务器返回的搜索结果文档中，是否有误报或者漏报的情况，如果有漏报，具体是遗漏了哪些文档。但该方案的安全索引增加了大量存储空间。M. Chase 在 [15] 中提出了一种可控信息揭露 (Controlled Disclosure) 的结构数据加密思想，这种数据密文可以在保持一定机密性的情况下实现快速查询。作者 Mohamad 则在 [16] 中讨论了可控信息

揭露的结构数据加密应如何支持验证，以确保查询结果是正确的。

上述方案都仅仅支持单关键词的精确搜索，不支持复杂条件搜索 — 模糊搜索、布尔搜索、优先级搜索等。文章 [17] 最早提出了的支持布尔与功能 (conjunctive search) 的对称可搜索加密方案。该方案为每个文档创建一个安全索引，索引中包含文档单词及其位置信息，搜索时需提供所有参与逻辑与操作的单词陷门及其对应的位置，只有单词和位置都符合的文档才会作为搜索结果。上述布尔方案性能较低，L. Ballard 在 [18] 中提出了两个支持逻辑与搜索的改进方案，分别基于 Shamir 门限秘密分享 [19] 方案和双线性映射，效率上有所提高。以上方案均仅支持逻辑与的搜索操作，T. Moataz 在 [20] 第一次提出支持逻辑与或非布尔搜索的方案。该方案将每个单词转换成相互独立的矢量，然后通过 Gram-Schmidt 过程正交化，以此建立安全索引。并且 T. Moataz 还进一步讨论了支持逻辑或和逻辑非的方案，通过严格证明该方案具有 Adaptive 的语义安全，但是仍存在性能和客户端存储量过大的问题。

上述方案都仅仅支持精确搜索，当在输入包含小的错误或单词不一致时，显得无能为力。[21] 中提出了第一个支持模糊搜索的对称可搜索加密方案。方案中使用编辑距离来度量单词的相似性，以通配符来减少单词的模糊集大小，但缺乏详细的方案描述和严格的安全证明。C. Wang 在 [22] 中给出了利用通配符构造相似集和前缀查找树 (trie-traverse tree) 有效缩减安全索引的大小的方案。上述方案由于使用了通配符来衡量单词的相似度，故在相似单词的比较时，只能使用编辑距离作文单位进行度量。M. Kuzu[23] 中给出采用 locality sensitive hashing(LSH)[24] 实现更宽泛的快速相似性评价方案。M. Chuah 在 [25] 中将模糊搜索从单个关键词拓展到多个关键词的情况，并基于 Bed-Tree 设计了具体的可搜索加密方案。

上述方案仅仅能将结果返回给用户，不能对结果进行过滤，例如搜索结果按优先级排序。C. Wang 在 [26][27] 中最先考虑了支持单个单词的优先级对称可搜索加密技术，给出了这类方案的具体定义，并设计了一个具体的方案。方案中对文档的优先级使用保序加密 (Order Preserving Encryption)[28] 技术进行加密；对于不同的单词，使用不同的密钥加密优先级。使得服务器无法得到优先级的具体值，甚至无法评估不同关键词对应搜索结果的优先级差异，从而降低了信息泄漏，使其安全性能与之前的对称可搜索加密方案相当。N. Cao 在 [29] 中将优先级搜索问题扩展到多个单词的情形，通过 coordinate matching[30]

综合考虑一篇文档相对于多个搜索单词的综合优先级，并给出了具体的支持多个搜索单词的优先级搜索方案。该方案的问题在于单词词典是固定的，如果需要增加新的单词的话，需要进行重构操作。Z. Xu 在 [31] 中针对这个问题作出了改进，使得新增单词时，只需要少量调整操作；方案中还考虑了单词访问频率对优先级的影响。J. Yu 则在 [29] 中做了进一步的研究，给出了安全性更强的方案。

1.3 研究内容及成果

1.3.1 研究内容

本课题所研究的内容都基于对称环境下的可搜索加密方案。我们的研究内容主要集中在相似可搜索加密机试（FSSE)、动态对称可搜索加密技术（DSSE）及可搜索加密方案上安全的改进。将本课题的研究内容描述如下：

1. 详细分析现有模糊可搜索加密方案和动态可搜索加密方案中存在，指出其中存在的不足；如当前模糊可搜索加密方案中普遍存在一个被忽略的问题：不同单词的模糊集可能存在交集，这个小的问题的存在会导致敌手对用户多次的查询信息进行分析，这大大降低了方案的安全性，泄漏了数据的更多隐私；动态可搜索加密方案中并没有提出一个完整的安全模型，分析可以了解到方案在该安全模型下的不足。从这些问题出发，我们对这两个问题进行研究。

2. 研究现有对称可搜索加密方案中的信息泄漏 — 搜索模式、访问模式和大小模式；其次探讨当前方案普通忽略的一个问题 — 密文文档在查找时的信息泄漏（文档与关键字的关联)。在理想情况下，服务器不应该从查找过程及结果中了解到文档的大小、文档的个数以及单词与文档之间的关系。

3. 研究相似可搜索加密技术中的另一个未被探究的场景 — 同义词搜索技术，同义词指不同的两个单词在同一个场景下具有不相同的含义。最终设计出具有强安全性、高效性和适用性的同义词对称可搜索加密方案。

1.3.2 研究成果

在对课题的内容进行深入的研究后，我们取得了如下成果：

1. 在 semi-trust 的云环境下，设计出了一个低存储开销和通讯开销并支持同义词搜索的可搜索加密方案。方案在提升功能的同时，并没有降低安全性和增加客户的计算和存储开销。在该方案中，我们详细地证明的该方案具备 Non-adpative 语义安全，并分析具有有效的查找时间和通讯开销。

2. 在单关键字的可搜索加密可搜索加密场景中，我们增强了其安全性，改进了方案在搜索时的信息泄漏。基于史密斯正交化理论，方案中避免了查找时搜索模式的信息泄漏，同时在通过增加少许额外存储开销，降低了访问模式的泄漏 — 仅仅以小概率泄漏。并通过严格的安全分析，证明了我们的方案具备 Non-adpative 的语义安全性。

1.4 论文结构

本文的内容分为五章，其结构安排如下:

- 第一章，绪论，首先介绍了课题的研究背景，然后介绍当前国内外在该领域的研究现状，之后介绍了本文的研究内容及所取得的成果，并对本文的结构进行了总结。

- 第二章，对称可搜索加密技术，先介绍了本课题在对称可搜索加密领域的相关知识，分类介绍了对称可搜索加密领域具有代表性的一些成果，并对其进行了性能分析和比较，提出当前研究上的不足，并描述可能继续研究的方向。

- 第三章，同义词对称可搜索加密方案，首先给出同义词对称可搜索加密方案中的若干定义、安全模型与攻击模型，然后提出一个具有实践的安全的同义词可搜索加密方案并给出了优化方案，并对方案进行了完整安全性证明和性能分析。

- 第四章，抗搜索模式信息泄漏的可搜索加密方案，首先描述了目前单关键词可搜索方案的信息泄漏，提出了解决该方案需要的定义及安全模型，然后提出一个首先给出同义词对称可搜索加密方案的若干定义和安全模型，并给出了方案的安全性证明。

- 第五章，总结和展望，系统地概述了我们在该课题中的工作与成果，并根据目前的研究现状对将来的工作作出了规划与展望。

第二章 对称可搜索加密技术

2.1 基础知识

2.1.1 数学基础知识

定义 2.1 (可忽略函数)**.** 对于一个给定的函数 $f: N^* \to N^*$，如果对任意给定的正多项式 $p(.)$，总存在一个足够大的数 k，有 $f(k) < \frac{1}{p(k)}$，则称函数 f 在 k 上是可忽略的。

定义 2.2 (伪随机函数)**.** [32] 对任意的函数 $\mathcal{F}: \{0,1\}^n * \{0,1\}^k \to \{0,1\}^m$，如果满足:

1. 对于任意给定的密钥 $K \in \{0,1\}^k$ 和输入 $x \in \{0,1\}^n$，总存在有效的（多项式时间）算法: $F_K(x) = F(x, K)$;
2. 对所有的多项式时间 oracle 算法 $\mathcal{A}$，有: $|Pr[\mathcal{A}^{\mathcal{F}_K(.)} = 1 : k \xleftarrow{\$} 0,1^K] - Pr[\mathcal{A}^{\mathcal{G}(.)} = 1 : \mathcal{G} \xleftarrow{\$} Func[n,m]]| \leq negl(k)$ ($Func[n,m]$ 表示所有的函数集合: $\{0,1\}^n \to \{0,1\}^m$，$negl$ 是在 k 上的可忽略函数)。

称函数 $\mathcal{F}$ 是伪随机函数。如果函数 $\mathcal{F}$ 是双射，则称为伪随机置换。

定义 2.3 (多项式时间算法)**.** 对于输入规模为 n 的算法，如果在最差情况下的运行时间为 $O(n^l)$ （l 为常数），则称其为多项式时间算法。

定义 2.4 (向量空间)**.** 给定域 F, 数域 F 上一个有序的 n 元数组称为域 F 上的一个 n 维向量，该向量表示为: $\begin{pmatrix} a_1 \\ a_2 \\ \cdot \\ a_n \end{pmatrix}$，$a_i$ 表示向量上的第 i 个元素。

定义 2.5 (施密特正交化)**.** 设 $\boldsymbol{v} \in \boldsymbol{V^n}$，$\boldsymbol{V^k}$ 是 $\boldsymbol{V^n}$ 上的 k 维子空间，$\boldsymbol{v}$ 上标准正交基为 $\{\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_k\}$，且 $\boldsymbol{v}$ 不在 $\boldsymbol{V^k}$ 上。由投影原理知，$\boldsymbol{v}$ 与其在 $\boldsymbol{V^k}$ 上的投影 $\text{proj}_{\boldsymbol{V^k}}\boldsymbol{v}$ 之差 $\boldsymbol{\beta} = \boldsymbol{v} - \sum_{i=1}^{k} \text{proj}_{\boldsymbol{\eta}_i} \boldsymbol{v} = \boldsymbol{v} - \sum_{i=1}^{k} \langle \boldsymbol{v}, \boldsymbol{\eta}_i \rangle \boldsymbol{\eta}_i$ 正交于子空间

$\boldsymbol{V}^k$，即 $\boldsymbol{\beta}$ 正交于 $\boldsymbol{V}^k$ 的正交基 $\boldsymbol{\eta}_i$。将 $\boldsymbol{\beta}$ 单位化：$\boldsymbol{\eta}_{k+1} = \frac{\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|} = \frac{\boldsymbol{\beta}}{\sqrt{\langle\boldsymbol{\beta},\boldsymbol{\beta}\rangle}}$，那么 $\{\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_k, \boldsymbol{\eta}_{k+1}\}$ 就是 $\boldsymbol{V}^k$ 在 $\boldsymbol{v}$ 上扩展的子空间 span$\{\boldsymbol{v}, \boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_k\}$ 的标准正交基。

对于向量组 $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ 张成的空间 $boldsymbolV^m(m < n)$，只要从其中一个向量 $\boldsymbol{v}_1$ 所张成的一维子空间 span$\{\boldsymbol{v}_1\}$ 开始，重复上述扩展构造正交基的过程，就能够得到在 $boldsymbolV^n$ 的一组正交基，则称该向量正交化的过程为 Gram-Schmidt 正交化。

2.1.2 密码学基础知识

定义 2.6 (随机预言机)**.** 预言机 [33] 是一个具有预言的图灵机，由四元组 $M = \langle Q, \delta, q_0, F\rangle$ 构成：

- Q 是有限多个状态的集合；
- $Q \times \{B, 1\}^2 \to Q \times \{B, 1\} \times \{L, R\}^2$ 是转移函数，L 和 R 分别代表左移和右移；
- $q_0 \in Q$ 代表起始状态；
- $F \subseteq Q$ 是终止状态的集合。

如果预言机满足以下性质：

- 确定性：对于相同的输入，预言机产生相同的输出；
- 均匀性：预言机的取值在输出空间中均匀分布；
- 有效性：对于任意给定的输入，预言机都能在低阶多项式内完成。

则称预言机为随机预言机。

定义 2.7 (对称加密方案)**.** 一个对称加密方案 SKE 是由三个多项式时间算法组成的集合，有：$SKE = (Gen, Enc, Dec)$.

- **Gen:**输入安全参数 k，返回私钥 K；
- **Enc:**输入密钥 K 和明文消息 m，并返回密文 c；

- **Dec:**输入密钥 K 和密文 c（密钥 K 与密文 c 生成时的密钥相同），解密得到消息 m。

如果密文没有泄漏明文的任何信息给敌手 $\mathcal{A}$，则我们称 SSE 在选择明文攻击下（CPA）是安全的。

定义 2.8 (自信息 I). 对于一离散性随机实验，其各实验结果为独立的随机分布：$S \in s_1, s_2, s_3 ... \ p(S = s) = p_k$，$\sum_k p_k = 1$，故事件 $S = s_k$, 出现概率为 p_k 的消息量为：

$$I(S_k) = log_2(\frac{1}{p_k}) = -log_2(p_k)$$

定义 2.9 (信息熵). 对一个值域为 $\{x_1, x_2, ..., x_n\}$ 的随机变量 X 的熵值 H 定义为：

$$H(X) = E(I(X))$$

其中，E 代表了期望函数，而 $I(X)$ 是 X 的信息量（又称为信息本）。$I(X)$ 本身是个随机变量。如果 p 代表了 X 的随机概率分布函数（probability function），则熵的公式可以表示为：

$$H(X) = \sum_{i=1}^{n} p(x_i)\, I(x_i) = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

2.1.3 安全索引

1. **正向安全索引（Forward-Index）**

   Goh. 等在 [10] 中提出了一种高效的可搜索加密方案。该方案的高效来源于使用 Bloom Filter[34] 来建立正向的索引结构。正向的索引结构是一种基于用文档来对应单词集的结构，即针对每个文档保存一份包含所有单词的列表，例如表2–1所示：

   从表2–1的结构中可以知道，在 Forward-Index 中搜索单词时，需要遍历索引中的每个条目，即对每个文档扫描一遍，逐个检查每个单词是否是否为所搜索的单词，效率相当低下 — 与文档数目和文档中单词的数目的乘积正相关。但是对于文档的增删改操作，索引的调整非常简单，增加，修改或者删除相应的一条索引条目即可。

   基于 Forward-Index 的安全索引会为每个文档建立一个安全索引条目，加密存放该文档包含的单词信息，搜索时需要遍历每个文档的索引条目。

表 2–1 正向索引示例。

Table 2–1 An Example Of Forward-Index.

文档	单词
$D1$	hello, word, welcome, document
$D2$	hello, paper, good, well
$D3$	world, paper, welcome, good
$D4$	document, welcome, well

2. **反向的安全索引（Inverted-Index）**

2006 年 Curtmola 等在 [12] 中第一次提出了使用 Inverted-Index 来建立高效安全的可搜索加密方案。该方案的查找非常高效，不需要查找每个文档，仅需要通过单词就可以找到所需要的文档信息。反向索引结构是一种基于用单词来确定对应文档的映射结构，即针对每个单词，其对应与包含该单词的所有文档的集合，例如表2–2 所示：

表 2–2 反向索引示例。

Table 2–2 An Example Of Inverted-Index.

单词	文档
$hello$	$D1$, $D3$, $D5$
$world$	$D2$, $D3$, $D5$
$paper$	$D1$, $D4$, $D5$
$document$	$D2$, $D4$, $D5$

从表2–2的结构中可以知道，在 Inverted-Index 中搜索单词时，只需要找到对应的条目，输出条目部分的文档集合即可，对于索引单词部分按照哈希建表的方式存储的情况，效率相当高（$O(1)$ 的查找时间）。但是对于文档的增删改操作，索引的调整则相当麻烦，涉及到修改包含所有该单词所定义文档集合的条目。

基于 inverted-Index 的安全索引将建立一个统一的索引，并进行加密，搜索时通过一次查找就能获得所有符合搜索条件的文档 — 效率相当高。

2.2 安全模型

对称可搜索加密技术（SSE）中信息泄漏主要发生在搜索阶段，以单词所携带的信息量为单位，我们以选择关键字攻击（CKA）（不同于 CPA 和 CCA）为攻击模型进行安全分析。为此，我们描述了可搜索加密环境下的具有代表性的两个攻击模型 — Non-Adaptive 安全和 Adaptive 安全，并分别从语义安全（semantic security）和不可区分安全（indistinguishability security）进行描述。正式的定义安全模型之前，首先定义一些基本概念。

定义 2.8（History）：假定 Δ 代表单词词典，D 表示在字典 Δ 上的一组文档集合。D 上 q 次查询的历史 (History) 定义为一个元组 $H = (D, w)$，其中 w 是包含 q 个单词的集合：$w = (w_1, ..., w_q)$。

定义 2.9（Access Patter）：假定 Δ 代表单词词典，D 表示在字典 Δ 上的一组文档集合。D 上 q 次查询历史 $H = (D, w)$ 的访问模式 (Access Pattern) 定义为一个元组 $\partial(H) = (D(w_1), ..., D(w_q))$。

定义 2.10（Search Pattern）：假定 Δ 代表单词词典，D 表示在字典 Δ 上的一组文档集合。D 上 q 次查询历史 $H = (D, w)$ 的搜索模式 (Search Pattern) 定义为对称二元矩阵 $\sigma(H) = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,q} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ x_{q,1} & x_{q,1} & \cdots & x_{q,q} \end{pmatrix}$。对于任意 $1 \le i, j \le q$，如果 $w_i = w_j$，则 $x_{i,j}$ 为 1，否则为 0。

定义 2.11（Trace）：假定 Δ 代表单词词典，D 表示在字典 Δ 上的一组文档集合。D 上 q 次查询历史 $H = (D, w)$ 的 Trace 定义为：$(H) = (|D_1|, ..., |D_n|, \partial(H), \sigma(H))$，其中 $|D_i| (1 \le i \le n)$ 是代表第 i 个文档的长度。

定义 2.12（Non-singular history）：如果对任意的历史 H 满足：(1) 至少存在一个历史 $H' \neq H$，使 $(H') = (H)$；(2) 在给定的 (H) 下，历史 H' 可以在多项式时间内被发现；则称历史 H 是 Non-singular。

下面我们分别给出了在对称可搜索加密环境下的 Non-adaptive 和 adaptive 的安全模型，如图2–1 所示，然后给出相关的具体定义。

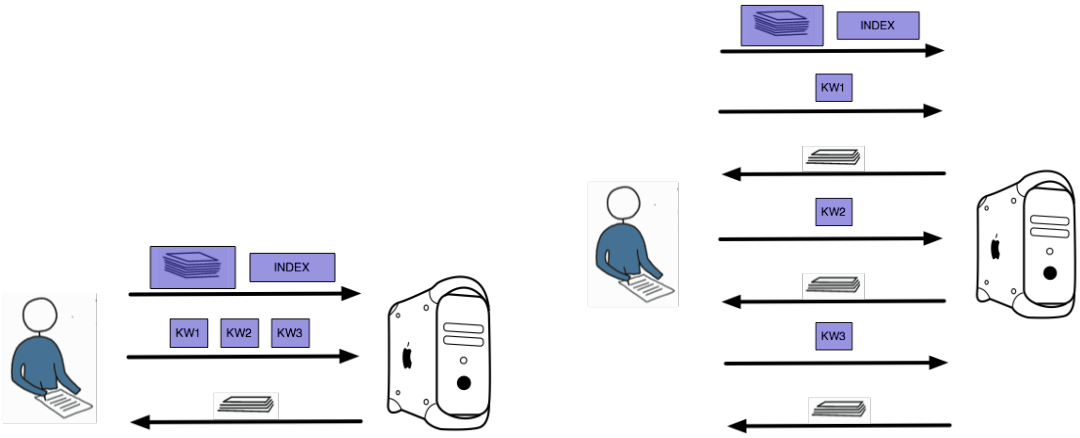

图 2–1 SSE 上 Non-adaptive 和 adaptive 安全模型

Fig 2–1 The Security Model of Nonadaptive and adaptive in SSE

2.2.1 Non-Adaptive 安全模型

定义 2.13 Non-adaptive Indistinguishability：在词典为 Δ，安全参数为 $k \in \mathbb{N}$ 的基础上，假设 $SSE = (Gen, Enc, Trpdr, Search, Decrypt)$ 是一个基于安全索引的对称可搜索加密方案，且 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ 为 $non - uniform$ 的敌手，考虑下面的概率试验 $\mathbf{Ind}_{(SSE,\mathcal{A})}(k)$:

$\mathbf{Ind}_{SSE,\mathcal{A}}(k)$
$K \leftarrow Gen(1^k)$
$(st_{\mathcal{A}}, H_0, H_1) \leftarrow \mathcal{A}_1(1^k)$
$b \xleftarrow{\$} 0, 1$
parse H_b as (D_b, w_b)
$(I_b, c_b) \leftarrow Enc_K(D_b)$
$for 1 \leq i \leq q$
$t_{b,i} \leftarrow Trpdr_K(w_{b,i})$
let $t_b = (t_{b,1}, ..., t_{b,q})$
$b' \leftarrow \mathcal{A}_2(st_{\mathcal{A}}, I_b, c_b, t_b)$
if $b' = b$, output 1
otherwise output 0

假设在 $\tau(H_0) = \tau(H_1)$，$st_{\mathcal{A}}$ — 表示敌手 $\mathcal{A}_1$ 的状态信息的前提下，如果对于任意多项式的敌手 $\mathcal{A}_1$，有:

$$Pr[\mathbf{Ind}_{SSE,\mathcal{A}}(k)=1] \leq \tfrac{1}{2} + negl(k),$$

则称该 SSE 方案在 Non-adaptive 不可区分的条件下是安全的。

定义 2.14（Non-adaptive Semantic Security）：在词典为 Δ，安全参数为 $k \in \mathbb{N}$ 基础上，假设 $SSE = (Gen, Enc, Trpdr, Search, Decrypt)$ 是一个基于安全索引的对称可搜索加密方案，$\mathcal{A}$ 为一个敌手，$\mathcal{S}$ 是一个模拟器（Simulator），考虑下面概率过程：

$\mathbf{Real}_{SSE,\mathcal{A}}(k)$
- $K \leftarrow Gen(1^k)$
- $(st_{\mathcal{A}}, H) \leftarrow \mathcal{A}(1^k)$
- parse H as (D, w)
- $(I, c) \leftarrow Enc_K(D)$
- $for 1 \leq i \leq q$
  - $t_i \leftarrow Trpdr_K(w_i)$
- let $t = (t_i, ..., t_q)$
- output $V = (I, c, t)$ and $st_{\mathcal{A}}$

$\mathbf{Sim}_{SSE,\mathcal{A},\mathcal{S}}$
- $(H, st_{\mathcal{A}}) \leftarrow \mathcal{A}(1^k)$
- $V \leftarrow S(\tau(H))$
- output V and $st_{\mathcal{A}}$

如果对于任何多项式规模敌手 $\mathcal{A}$，都存在一个模拟器 $\mathcal{S}$，使得对于任意多项式规模的区分器 $\mathcal{D}$，有：

$$|Pr[\mathcal{D}(V, st_{\mathcal{A}})] = 1 : (V, st_{\mathcal{A}}) \leftarrow \mathbf{Real}_{SSE,A}(k)] - Pr[\mathcal{D}(v, st_{\mathcal{A}}) = 1 : (V, st_{\mathcal{A}} \leftarrow \mathbf{Sim}_{SSE,\mathcal{A},\mathcal{S}}(k)]| \leq negl(k),$$

则称该 SSE 在 Non-adaptive 的条件下是语义安全的。

2.2.2 Adaptive 安全模型

定义 2.15 (Adaptive Indistinguishability)：在词典为 Δ，安全参数为 $k \in \mathbb{N}$ 的基础上，假设 $SSE = (Gen, Enc, Trpdr, Search, Decrypt)$ 是一个基于安全索引的对称可搜索加密方案，且 $\mathcal{A} = (\mathcal{A}_1, ..., \mathcal{A}_{q+1})$ 为 $non-uniform$ 的敌手，考虑下面的概率试验 $\mathbf{Ind}^*_{(SSE,\mathcal{A})}(k)$：

$\mathbf{Ind}^{*}_{SSE,\mathcal{A}}(k)$

$K \leftarrow Gen(1^k)$

$b \xleftarrow{\$} 0,1$

$(st_{\mathcal{A}}, D_0, D_1) \leftarrow \mathcal{A}_0(1^k)$

$(I_b, c_b) \leftarrow Enc_K(D_b)$

$(st_A, w_{0,1}, w_{1,1}) \leftarrow \mathcal{A}_1(st_{\mathcal{A}}, I_b)$

$t_{b,1} \leftarrow Trpdr_K(w_{b,1})$

for $2 \leq i \leq q$,

$(st_{\mathcal{A}}, w_{0,i}, w_{1,i} \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, I_b, c_b, t_{b,1}, ..., t_{b,i-1})$

$t_{b,i} \leftarrow Trpdr_K(W_{b,i})$

let $t_b = (t_{b,1}, ..., t_{b,q})$

$b' \leftarrow \mathcal{A}_{q+1}(st_{\mathcal{A}}, I_b, c_b, t_b)$

if $b' = b$, output 1

otherwise output 0

在 $\tau(D_0, w_{0,1}, ..., w_{0,q}) = \tau(D_1, w_{1,1}, ..., w_{1,q})$ 的前提下。如果对于所有多项式规模的敌手 $\mathcal{A} = (\mathcal{A}_0, ..., \mathcal{A}_{q+1})$，均有有:

$$Pr[\mathbf{Ind}^{*}_{SSE,\mathcal{A}}(k) = 1] \leq \tfrac{1}{2} + negl(k),$$

则称该 SSE 方案在 adaptive 不可区分的条件下是安全的。

定义 2.6 (Adaptive Semantic Security)：在词典为 Δ，安全参数为 $k \in \mathbb{N}$ 基础上，假设 $SSE = (Gen, Enc, Trpdr, Search, Decrypt)$ 是一个基于安全索引的对称可搜索加密方案，$\mathcal{A} = (\mathcal{A}_0, ..., \mathcal{A}_q)$ 为敌手，$\mathcal{S} = (\mathcal{S}, ..., \mathcal{S}_q)$ 是模拟器 (Simulator)，考虑下面概率过程:

$\mathbf{Real}^*_{SSE,\mathcal{A}}(k)$
$K \leftarrow Gen(1^k)$
$(D, st_{\mathcal{A}}) \leftarrow \mathcal{A}_0(1^k)$
$(I, c) \leftarrow Enc_K(D)$
$(w_1, st_{\mathcal{A}}) \leftarrow \mathcal{A}_1(st_{\mathcal{A}}, I, c)$
$t_1 \leftarrow Trpdr_K(w_1)$
for $2 \leq i \leq q$
$(w_i, st_{\mathcal{A}}) \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, I, c, t_1, ..., t_{i-1})$
$t_i \leftarrow Trpdr_K(w_i)$
let $t = (t_1, ..., t_q)$
output $V = (I, c, t)$ and $st_{\mathcal{A}}$

$\mathbf{Sim}^*_{SSE,\mathcal{A},\mathcal{S}}$
$(D, st_{\mathcal{A}}) \leftarrow \mathcal{A}_0(1^k)$
$(I, c, st_{\mathcal{S}}) \leftarrow \mathcal{S}_0(\tau(D))$
$(w_1, st_{\mathcal{A}} \leftarrow \mathcal{A}_1(st_{\mathcal{A}}, I, c)$
$(t_1, st_{\mathcal{S}}) \leftarrow \mathcal{S}_1(st_{\mathcal{S}}, \tau(D, w_1))$
for $2 \leq i \leq q$
$(w_i, st_{\mathcal{A}}) \leftarrow \mathcal{A}_i(st_{\mathcal{A}}, I, c, t_1, ..., t_{i-1})$
$(t_i, st_{\mathcal{S}}) \leftarrow (S)_i(st_{\mathcal{S}}, \tau(D, w_1, ..., w_i))$
let $t = (t_1, ..., t_q)$
output $V = (I, c, t)$ and $st_{\mathcal{A}}$

如果对于任何多项式规模敌手 $\mathcal{A} = (\mathcal{A}_0, ..., \mathcal{A}_q)$，都存在模拟器 $\mathcal{S} = (\mathcal{S}, ..., \mathcal{S}_q)$，使得对于任意多项式规模的区分器 $\mathcal{D}$，有：

$$|Pr[\mathcal{D}(V, st_{\mathcal{A}})] = 1 : (V, st_{\mathcal{A}}) \leftarrow \mathbf{Real}^*_{SSE,A}(k)] - Pr[\mathcal{D}(v, st_{\mathcal{A}}) = 1 : (V, st_{\mathcal{A}}) \leftarrow \mathbf{Sim}^*_{SSE,\mathcal{A},\mathcal{S}}(k)]| \leq negl(k),$$

则称该 SSE 在 adaptive 的条件下是语义安全的。

Curtmola 在论文 [12] 中分别证明 Non-adaptive 不可区分安全性与 Non-adaptive 语义安全和 adaptive 不可区分安全性与 adaptive 语义安全是等价的。

2.3　对称可搜索加密技术

2.3.1　单关键字搜索

Goh 在文章 [10] 中的提出了第一个基于正向索引的解决方案。方案使用一个映射文档到文档中所有单词的 Bloom Filter[34] 作为该方案的安全索引。在搜索时，授权用户发送待查单词的多个哈希值作为陷门；一旦服务器收到陷门后，对发送过来的每个元素，服务器检查在安全索引 Bloom Filter 中的对应位是否是“1”，如果全部都存在，则输出 1，否则输出 0。该方案突出的优势体现在性能上：Bloom Filter 的映射过程只需要计算若干 Hash 函数，故索引创建过程性能较高。在搜索阶段，服务器需要对每个文档调用一次 SearchIndex 算

法，而在该算法中对陷门中每个元素只需进行 Bloom Filter 数据位的比较操作，时间复杂度仅为 O(1)，故整体效率仍然是比较高的。但是该方案却引进了误报率，且误报率和安全索引的大小成反比关系。

Y. Chang 在 [11] 中则基于 forward-index 提出了改进的没有误报率的的方案。其基本思想如下：将待上传文档中所有出现的单词构成一个词典，安全索引中用 1 位代表每个单词是否存在 — 1 表示存在该单词，0 表示不存在。方案同样达到了相当高的效率，但问题在于词典保存将大大增加了存储开销。在文中，Y. Chang 给出了两个解决方案，第一个方案是将词典保存在客户端，这样用户每次查询则需要本地词典，这样增加了本地的存储和计算开销，并且在多用户环境下还需要同步词典；第二个方案则将词典加密后存放在远程服务器上，但查询过程则需要通讯两轮 — 一轮获得词典里索引信息，另一轮查询获得文档。

R.Curtmola 在文章 [12] 中第一次基于 Inverted-index 对对称可搜索加密提出了两个解决方案，并给出了详细的安全性定义和证明。第一个方案达到了 Non-Adaptive 安全性，其基本思想描述如下：首先构建待上传文档的 inverted-index；然后将索引表项中的文档 ID 部分加密并随机分散到一个数组中，如下图**??**所示：

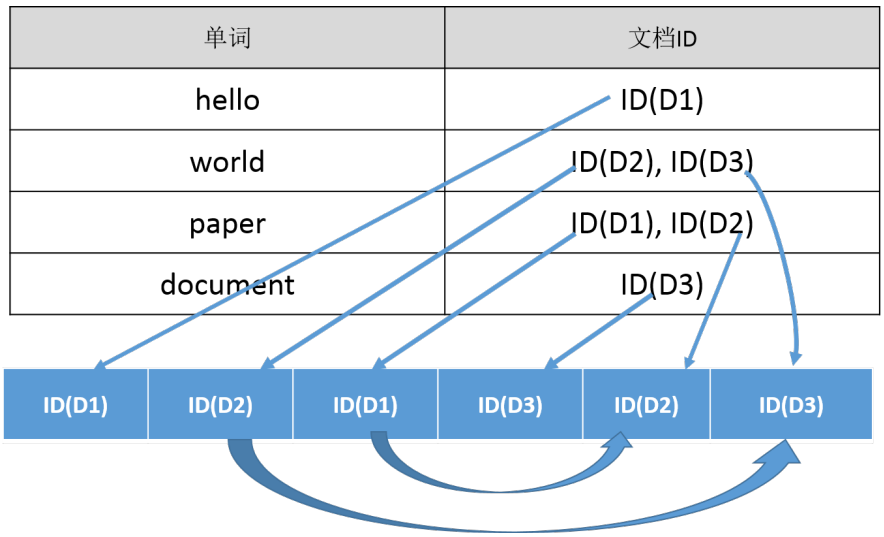

图 2–2 文档 ID 数组

从上表可知，每个单词所对应文档的 ID 形成一个链表。数组中的每个

元素使用不同的密钥调用对称分组密码算法加密。而文档中所有的单词在 Inverted-index 则以查找表的形式存储，如图**??** 所示：

单词陷门	文档ID在数组中起始下标
T(hello)	1
T(world)	2, 6
T(paper)	3, 5
T(document)	4

图 2–3 安全索引中的查找表

查找表的条目由键值对的形式表示，其键存储的是通过伪随机函数对单词置换后的结果 — 单词的陷门，其值所存储的部分为对应文档 ID 的链表在数组中的起始位置。查找的时候，通过单词的陷门在查找表处找到其 ID 链表在数组中的起始位置，然后根据起始 ID 位置逐个解密这个链表获得所有包含该单词的文档 ID 集。

Curtmola 方案一的主要优势是服务器端的搜索效率高，得益于 inverted-index，其服务器端搜索时间复杂度为 O(1)，而之前的方案都只能达到 O(n)（n 表示文档的数目）。同时 R. Curtmola 在文中给出了另一个具有 Adaptive 安全性的方案，服务器端搜索时间复杂度仍能保持 O(1)，但是服务器端的存储量和单词陷门的大小均有所增加。

为了确保方案足够安全 — 不能因 ID 链表元素个数不同而导致信息泄漏，Curtmola 建议数组元素的个数应与所有文档的总长度所能容纳的最大单词数相当。这将导致在安全索引中，数组部分所占用的存储空间将会远大于所有文档的总长度。H. Lu 在文章 [35] 中基于 inverted-index 提出了一种降低安全索引结构所占存储空间的方案。方案与 Curtmola 在方案 1 的最大不同在于合并了安全索引中 ID 链表结构中相同的元素，从而极大减少了数组的长度，如图**??**所示：基于这样的结构使得合并后每个文档 ID 在数组中只出现一次，从而减少数组元素的总数。经实际测试，方案中数组的元素个数可降至原来的 5% 以下。

作者 Van 在 [36] 中基于 inverted-index 提出了另一种形式的安全索引方案，方案使用一个加密的二元数组维护安全索引信息，数组两维分别表示单词和文

单词	文档ID
hello	ID(D1), ID(2), ID(4)
world	ID(1), ID(D2), ID(D3)
paper	ID(D1), ID(D2)
document	ID(D3)

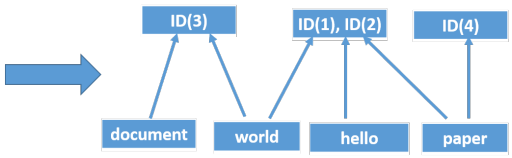

图 2–4 改进的文档 ID 链表

档 ID，数组元素为 1 表示对应的文档包含对应的单词。搜索时解密二维数组中对应单词所在的行，即可知晓包含该单词的文档 ID。

2.3.2 模糊搜索

可搜索加密环境中，当前的相似搜索集中在模糊搜索上。基于明文模糊搜索 [37] [38] [39] 和精确关键字搜索 [12] [11] [35] 的基础上，Jin Li 等人于 2010 年在 [21] 中率先提出了模糊对称可搜索加密的问题，并提供了两种解决模糊加密搜索的方案：直接模糊搜索方案和基于通配符的模糊搜索方案。两个方案都以编辑距离（edit distance[40]）来衡量单词之间的相似性，即若给定单词 s1 和 s2，s1 和 s2 之间的编辑距离为：将 s1 变换成 s2 所用的最小的变换操作数，变换操作包括：（1）插入：在单词的某个位置上插入一个字母；（2）删除：删除单词中某个字符；（3）替换：将单词中某个字符替换为另一个字符。该方案的基本思想如下：（1）加密阶段：对文档集中每个单词构建单词模糊集，使用 [12] 中方案构建单词令牌同样方式对模糊集中每个子元素作为一个新的单词来构建反向索引，然后将反向安全索引和加密文档外包；（2）令牌构建阶段：当搜索某单词 w 时，首先对单词 w 构建模糊集（模糊集的构建设计到可容忍的误差—例如单词的编辑距离 d），然后通过计算单词 w 和其模糊集的对应令牌并发送给服务器；（3）搜索阶段：当服务器收到用户发送过来的令牌后，服务器首先在安全索引中查询是否存在和单词 w 令牌精确匹配的索引，若存在则返回相应结果，否则返回单词 w 模糊集对应令牌的所有结果，查找过程与单关键字搜索方案类似。

在直接的模糊搜索方案中，对于单词 w 和单词之间可容忍的误差为 d（即编辑距离为 d）的情形，模糊集构建如下：枚举所有的单词 w' 使得 $ed(w, w') \leq d$，即模糊集中包含所有与单词 w 编辑距离小于等于 d 的单词。

而在改进的模糊搜索方案中，用通配符来替代同一位置上所有不同的单词，这样大大降低了模糊集的大小，致使网络传输开销和服务端的存储开销大大降低。本文做出了如下贡献：（1）第一次提出了模糊可搜索加密方案；（2）使用编辑距离来衡量关键字之间的相似性和通配符来减少通信和存储开销；3）证明了该方案具有 Non-adaptive 的安全性。同时该方案也存在以下不足：（1）每次在构建某个单词的索引时，将该单词对应模糊集中每个元素作为一个新的单词构建索引并插入到索引表中，这样大大增加了服务端的存储量；（2）该方案不支持多关键字搜索；3）不同单词的模糊集之间存在碰撞 — 降低数据的隐私，例如对于单词 "at" 和 "it" 在编辑距离为 1 时，$S_{at,1} = \{*at, *t, a*t, a*, at*\}, S_{it,1} = \{*it, *t, i*t, i*, it*\}$（$S_{w,d}$ 表示单词 w 在编辑距离为 d 时的模糊集，有 $S_{at,1} \cap S_{it,1} \neq \varnothing$。

为了解决在上述方案中服务端存储量过大的问题，M. Chuah 等人在 [25] 中提出了基于 BedTree[41] 的模糊可搜索加密方案 — 减少服务端存储开销和索引构建时间。方案的具体过程如下：（1）加密阶段：对于单词 w 在编辑距离为 d 时，首先将单词 w 的模糊集对应的令牌映射到 Bloom Filter 中，如将 $S_{w,i}$ 映射到 B_i 中（$S_{w,i}$ 表示编辑距离为 i 的单词的模糊集，B_i 表示编辑距离为 i 时模糊集所生成的 Bloom Filter），然后将单词的 $\{B_i, i \leq d\}$、单词所对应文档的 ID 信息以及单词的陷门作为一个叶子节点通过单词的数据向量 [17] 插入到 BedTree 树中；（2）令牌生成阶段：对待搜索单词 w 计算其数据向量和模糊集中每个元素的令牌—hash 函数计算，然后将其发送给服务端；（3）搜索阶段：当服务器收到用户发送过来的查询信息后，服务器首先通过待搜索单词的数据向量在基于 BedTree 树的安全索引中找到叶子节点，再通过单词模糊集所对应的令牌在叶子节点的 B_i 中进行查找，最终返回搜索到的结果集。该方案主要对 [9] 方案中的安全索引结构进行了改进，提出了基于 BedTree 的索引结构，使得方案比上述方案具有更小的存储量，同时方案具有很好的支持多关键词搜索和增量更新的扩展性，然而方案也引入了新的的问题 — 在索引构建过程中，由于对单词 w 和编辑距离 d 构建索引结构是基于 Bloom Filter，而 Bloom Filter 的构建最终依赖于哈希函数，但是哈希函数中存在碰撞，这样导致了搜索结果的误报。

Cong Wang 等人在 [22] 中基于 Trie 树的结构提出了搜索效率更高同时具有同样隐私安全的模糊可搜索加密方案。他们首先对方案 [21] 提出了改进措施：对每个单词 w，根据容许的最大编辑距离 d 计算出单词的模糊集 $S_{w,d}$，然后根据 Bloom Filter 技术将每个单词的模糊集计算出的令牌映射到一个 Bloom

Filter，即对每个关键字的模糊集构建一个 Bloom Filter；在搜索关键字 w 时，服务端只需搜索待搜索单词模糊集对应令牌是否存在某个 Bloom Filter 中，即可查询得到所需的结果。基于 [21] 的改进方案虽然减少服务器的存储量，但是却增加了服务端的搜索时间—– 对所有关键字构建的 Bloom Filter 索引都要搜索一次，并且由 Bloom Filter 结构带来了一定的误报率。方案的构建过程如下：（1）加密阶段：首先计算文档中每个单词的模糊集和对应令牌信息。然后对于每个令牌将其分成 N 块，并将 N 块的子结构插入到 Trie 树中，叶子节点即单词令牌的最后一块，并将单词对应文档信息放在叶子节点中或者在当前叶子节点下插入一个新的节点存放其他信息，最后将安全索引和文档密文外包到服务端；（2）令牌生成阶段：对单词 w，计算模糊集及对应令牌，并发送给服务端；（3）搜索阶段：当服务端收到用户发来的令牌集后，首先对单词 w 的令牌按照构建索引过程分成 N 块，然后在基于 Trie 树的索引中进行精确查找，若找到返回精确查询的结果，否则对其模糊集令牌按照同样方式搜索并返回结果集。该方案主要基于不同单词之间存在相同的部分，通过相同的部分来减少存储量。该方案与方案 [21] 相比，明显减少了服务端的存储开销但是增加了常量的搜索时间。与方案 [25] 相比，该方案减少了搜索过程中信息的泄漏增强了方案的安全性，但是从数据的时间局部性和空间局部性的角度来看，由于 BedTree 是一颗 B+ 树，而 Trie 是一棵多叉树，因而 B+ 树可以利用数据局部性原理来通过减少内存在访问过程中搜索失效时数据内存时间换入换出来提高单词的搜索时间。

基于以上模糊搜索方案，Deshpande 等人在 [42] 中对目前所有模糊可搜索加密方案进行了总结 — 包括一般的基于通配符构建模糊集的可搜索加密方案、基于 BedTree 树构建索引的模糊可搜索加密方案和基于 Trie 树构建索引的模糊可搜索加密方案；并提出了两种高效的模糊单词集构建方案 — 基于通配符和基于 Gram；最终通过实验衡量了各方案在不同编辑距离时索引构建的时间和搜索的时间，分析和比较了各方案的优缺点。

2.3.3　动态搜索

Goh 于 2003 年在 [10] 中第一次提出基于安全索引的可搜索加密技术的解决方案。该方案在整个过程中使用的是正向索引，方便建立索引和查找工作。并且该方案对文档的增删改操作，索引结构的调整非常简单，删除文档时仅需

删除该文档在正向索引中的索引项和多对应的文档；添加时，只需添加文档和在安全索引中插入一个文档及其单词集合的索引项。由于该方案的安全索引构建是基于 Bloom Filter，导致不同关键字的令牌可能会发生碰撞 — 不同单词将映射到 Bloom Filter 索引中相同的位置，从而导致返回的结果存在误报率。为了解决上述方案中的不使用问题，Kamara 等人于 2012 年在 [43] 中第一次提出了动态可搜索加密技术的问题及相应的解决方案。在文中作者主要结合了正向索引具有的动态修改特性和反向索引的强安全性和无误报率等特征，在反向索引方案的基础上通过增加另一个正向索引，实现了具有动态修改的可搜索加密方案。该方案不仅获得了有效的搜索时间 — 线性搜索时间（与文档数目成正比），而且确保了搜索过程中的信息泄漏，具有 Non-adaptive 的安全性。该方案的主要思想如下：在索引构建阶段，不但对每个单词构建了一个反向索引，而且对每篇文档构建了一个正向索引，正向索引和反向索引通过对偶节点（dual node）联系起来（所谓对偶节点即将正向索引结构中的 < 文档，单词 > 条目和反向索引结构中的 < 单词，文档 > 条目具有相同值的那对节点称为对偶节点）；在搜索时，只需要查询反向安全索引即可；而在添加或删除文档时，需通过一系列复杂的位操作对正向索引和方向索引进行更新，同时维护正向索引与反向索引之间的对应关系。随后，Kamara 在文章 [44] 中对 [43] 方案进行扩展，提出具有并行特征的动态可搜索加密方案，这样可充分利用基于多核和分布式的计算机来提升方案的整体性能。但是服务器端额外存储量增加了 1 倍以上。

Stefanov 等人在 [45] 中提出了具有更少信息泄漏和更高效的动态可搜索加密方案。他们指出 Kamara 方案中泄漏的信息并不仅仅限于他们方案中所定义的信息 — 泄漏搜索模式（search pattern）[46]、访问模式（access pattern）[47] 和大小模式（size pattern），同时也存在 forward privacy（即在搜索单词 w 后，然后立即添加一个包含 w 的文档，服务器不应该了解到新添加的文档包含用户刚搜索过的单词 w）和 backward privacy（即当删除包含某单词 w 的文档后，随之搜索单词 w，服务器不应该了解到刚被删除的文档包含单词 w）的信息泄漏。作者在文中第一次提出具有 forward privacy 安全的动态可搜索加密方案，但是 backward privacy 的信息泄露问题仍然没有得到解决。

第三章　抗搜索模式泄露的可搜索加密方案

到目前为止，可搜索加密方案大多从实用性、效率与存储开销入手，方案的安全性证明都基于 trace 信息 — size pattern、search pattern 和 access pattern 泄漏的前提下。没有一个方案能从本质上做到抵御 search pattern 的信息泄漏问题，为此从方案的安全性考虑，必须提供一种能解决或者尽量减少 search pattern 和 access pattern 信息泄漏的问题。

在这一节，我们将提出一种解决 access pattern 和 search pattern 信息泄漏的方案。首先我们指出当前方案信息泄漏的问题；然后建立一种更强的系统模型，并简述方案的框架模块；随后详细地描述整个方案的细节；最后给出方案的安全性证明、性能分析以及应用场景。

3.1　问题定义

在对称可搜索加密环境下，一个通用的系统模型如图3–1所示：

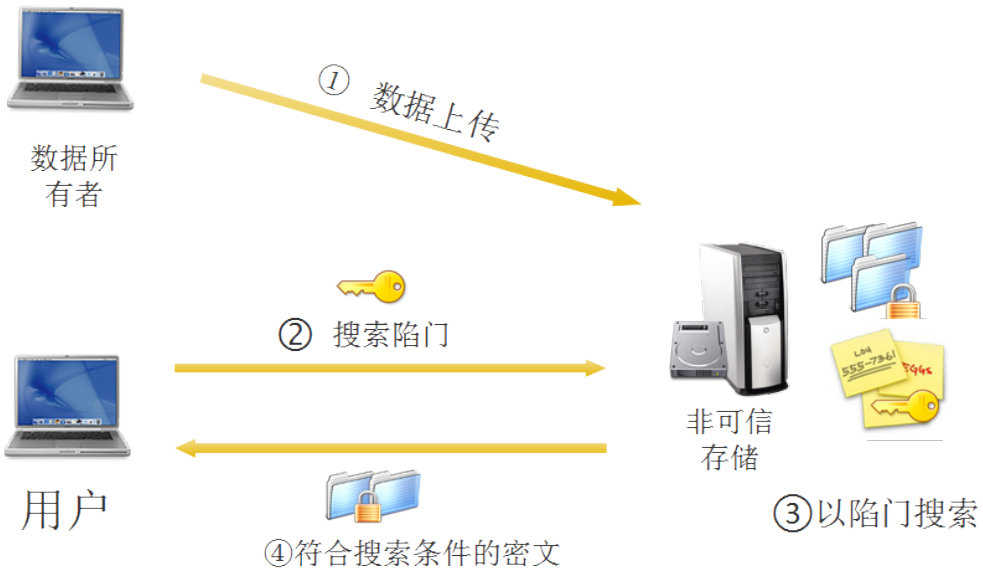

图 3–1 SSE 上通用系统模型

Fig 3–1 A General System Model in SSE

基于通用的可搜索加密模型，我们分别从搜索模式和访问模式分析当前方案中的信息泄漏。

3.1.1 搜索模式泄漏

3.1.2 访问模式泄漏

3.2 系统模型

3.3 算法框架

3.4 方案细节

3.5 安全性证明

3.6 性能分析

3.7 应用

第四章　同义词对称可搜索加密

当前对称可搜索加密技术得到了广泛的研究，各种关于复杂条件的可搜索加密技术也得到了各种深入的研究，但是处于复杂多变的云计算环境下，我们需要研究的知识点和面对的用户群里基数大。为此，我们不得不进一步挖掘出一些尚未提出并具有实际意义的问题和知识点。在前人工作的指导下，本文提出了一个类似模糊搜索的同义词搜索技术。

在本节，我们提出了一个支持同义词搜索并且具有强安全性的的对称可搜索加密方案。在这个方案中，我们首先定义了方案的系统模型和攻击模型；然后定义算法框架；针对方案的框架的各个算法，紧接着我们定义了它们的详细实现细节；最后我们对此方案进行了安全性证明和性能分析。

4.1　方案模型

在该小节，我们首先分析了提出我们方案的场景；然后在场景下定义了我们方案所应用的系统模型；并针对该系统模型，我们提出了有效的攻击模型；为了实现方案的详细细节，我们引入一些相关的符号定义和相关概念，同时我们在此定义了我们方案中的信息泄漏；最后，我们简略地描述了我们方案的基本流程。

4.1.1　问题提出

生活在网络高速发达的时代，数据的产量以指数的量级增长，大数据的时代已到来，为解决人们难以应付的难题，可搜索加密方案被提出并解决了大数据中存储和计算的问题。然而，在网络竞争时代，人们在工作中的强度日益剧增，致使人们的记忆力也随着超载工作而过度消耗而呈现下降趋势。人们记忆力的周期将大大缩短，即一段时间之后，对之前使用过的文档及其内容记忆模糊甚至遗忘。一个典型的场景如图4–1所示。首先，使用云计算平台的数据所有者编辑文档并随后将其存储到不可信的云端；一段时间后，数据所有者想要查找包含“文件”含义的关键字的文档时，其可能忘记不知道文档中到底包含“paper”还是“document”，为次，用户不得不逐个尝试有该含义的所有单词，

直至找到所需的答案。为解决这样的问题，该节提出了一个解决此场景的方案 — 同义词对称可搜索加密。

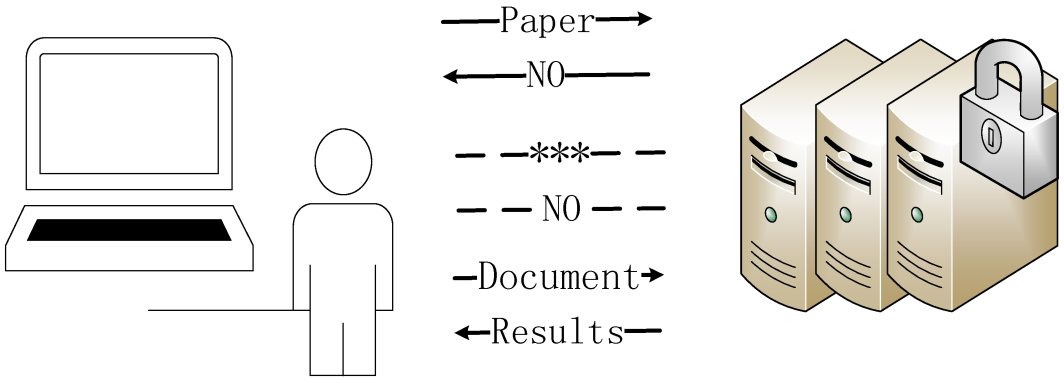

图 4–1 同义词可搜索加密系统问题示例

Fig 4–1 An Example of Problem in the Synonym Search

4.1.2 系统模型

在我们的方案中，一个常见的系统架构模型（如下图4–2所示）包括三部分 — 数据所有者（Owner），用户（Users）和云服务提供商（Provider）。这三部分在系统中分别承担不同的作用。数据拥有者拥有资源和系统的选举权，用户是系统中最频繁的访问者，而云服务提供商是系统的承担者，默默为用户提供这种功能。

1. **数据所有者：** 数据拥有者是系统的主体，数据拥有者拥有系统中最核心的部分 —数据。数据拥有者往往由一个公司、组织或团体组成。一般情况下，数据拥有者拥有数据，但是他想利用云外包带来的便利。同时，数据拥有者需要考虑系统的性能、功能和其它各种因素。当数据拥有者想外包数据时，他便购买云服务，将所有的数据首先加密，然后存储到云服务提供商。

2. **用户：** 用户是系统的使用者，往往系统是系统中最大的人群（包括数据所有者)。用户往往不考虑系统的任何细节，只希望使用系统的便利性

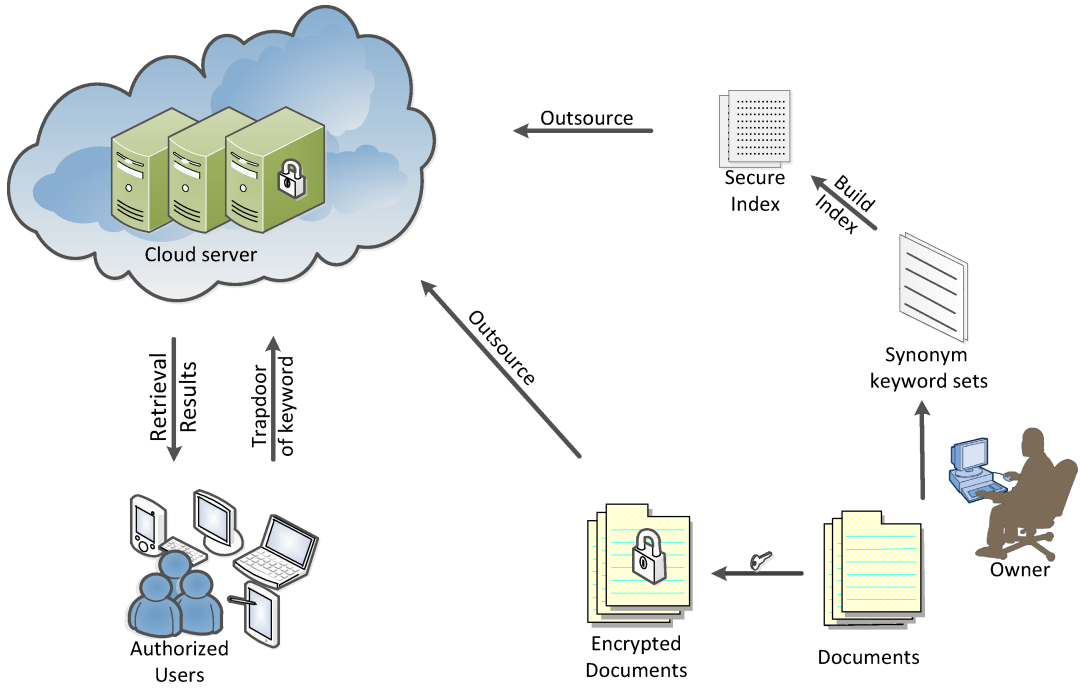

图 4–2 同义词可搜索加密系统模型

Fig 4–2 The System Model of Synonym Search

和安全性。通常情况下，如果一个用户想要使用系统的查询功能，首先他必须征求数据拥有者的允许 —获得请求数据的安全密钥（即被认证合法)；然后利用安全密钥，对远程外包数据进行搜索；最后在收到查询结构后，将所得的结果进行解密，还原所得的数据。在我们系统中，返回的结果包括所有包括该单词或和它有相同含义的文档。

3. **云服务提供商：** 云服务提供商是系统的客体 — 系统功能服务者，在系统中承担所有的服务，在系统中是必不可少的一部分。通常情况下，首先数据所有者将数据加密存储到云端，而云提供者提供对数据的计算和备份工作。当用户需要检索数据时，他提交查询信息给服务器，然后提供计算的服务器对数据进行查询操作，并将结果信息返回给用户，如果不存在则返回空。

在我们的系统中，如果假设云服务提供商是安全并且诚实的，因而我们系统中外包的数据是安全的 — 数据被加密。但是，云服务提供商通常情况下来说，虽然明着提供安全的云服务，而本身又是“honest-but-curious”—- 暗地里偷偷地分析用户查询的数据，然后根据一些其他知识（例如统计信息）来推断我们查询的信息，这置使我们的数据仍处理一定的风险。为此，我们必须清楚地了解到云服务到底能分析出我们系统的多少信息。为了分析我们系统泄漏信息量的大小，我们通常将云服务作为敌手，在他们有最强的计算能力的亲情况下，分析出的数据即为我们系统的信息泄漏量。

4.1.3 攻击模型

在系统模型中，我们指出了云服务提供者是“honst-but-curious”，使的我们的系统存在一定的信息泄漏。为了衡量我们信息泄漏量的正确性，我们在此定义攻击模型，才证明我们系统所定义的信息泄漏，并证明我们除了泄漏我们所定义的信息之外，不泄露任何其他的信息。我们定义我们的方案具有 Adaptive 安全性，我们定义了我们的攻击模型如下图。

4.1.4 相关定义

在我们的系统中，我们使用了一些符号以及定义，我们定义他们如下 (Note：X — 代表是一个数据结构，包括链表、数组或文档等等；n — 任意大小)：

- $|X|$ — 结构 X 的长度，即 X 中单词的个数。
- $[n]$ — 表示 n 个元素的集合，等价于 { 1, ..., n }。
- $max(|S|)$ — 如果 S 是集合，则它代表集合 S 的元素数目 |S|；若 S 是集合的集合，则它表示集合 S 中所有元素的最大数目，即 $max\{|S_i| \mid S_i \in S\}$。
- D — n 个明文文档的集合，即 D = $(D_1, D_2, ..., D_n)$。
- ED — n 个密文文档的集合即 ED = $(ED_1, ED_2, ..., ED_n)$。
- KED — 所有键值对$< ID(D_i), ED_i >$ 的集合（$ED_i \in ED$）。

- D_w SD_w — D_w 是仅包括单词 w 的文档的集合；SD_w 是包括单词 w 和其同义词的文档的集合。
- $ID(D)$ — 文档 D 的 ID 信息，可以用数字表示也哈希值来唯一表示，这里文档 D 可能是明文或加密密文。
- $W(X)$ —结构 X 中不同单词所组成的集合，如其有 p 个元素，则表示成: W(X) = $(w_1, w_2, ..., w_p)$。
- SW— 同义词字典的集合，它必须包含结构 $W(D)$ 中的所有单词，根据所定义环境的不同，可能包含其它的单词。m 个元素的集合 SW，表示为: SW = $(SW_1, ..., SW_m)$。
- S_w — 单词 w 的同义词的集合，包含 p 个单词的集合表示为: ($S_w = (S_{w1}, S_{w2}, ..., S_{wq})$。
- SI — 安全索引结构，在我们系统中使用数组来存储。
- T_w — 单词 w 的陷门信息；它使用伪随机函数来确保安全，用于在搜索过程中作文查询口令。
- ESR_w — 用户在查询时，服务器返回的加密的文件集合。
- SID_w — 包含单词 w 的文档的 ID 集合，定义 $SID_w = \{ID(D_i) \mid w \in D_i\}$, $SIDS = \{SID_w \mid w_i \in W(D)\}$。
- AL_i — 结构 AL（包括数组、链表等）中的第 i 个元素。
- $F(K, *)$ — 如果我们定义函数 F 为: $\{0,1\}^k * \{0,1\}^n \rightarrow \{0,1\}^m$，且在多项式时间内，函数 F 是可计算的和对于一个具有多项式访问 oracle 的敌手 A，有: $|Pr[A^{f_k(.)} = 1 : K \leftarrow \{0,1\}] - Pr[A^{g(.)} = 1 : g \leftarrow Func[n,m]]| \leq negl(k)$，则称函数 F 为伪随机函数（PRF），若函数 F 是双射，我们称它为伪随机置换（PRP）
- $SKE = (Gen, Enc, Dec)$ — 定义 SKE 是私钥加密函数。在标准 SKE 中，加密函数是伪随机函数，Gen 用于生成密钥，Enc 用于将给定的值加密，而 Dec 则用于解密。对于任意给定的两个密文，我们不能判断是否被同一个密钥加密。

同义词函数（Synonym Function） 对于任意给定的两个单词 w_1 和 w_2, 定义:

$$SF(w_1, w_2) = \begin{cases} 1 & \text{if } w_1 \text{ and } w_2 \text{ is synonym} \\ 0 & \text{otherwise} \end{cases} \tag{4–1}$$

从公式 1 中，如果输出结果为 1，则称单词 W_1 和 W_2 相似，称函数 SF 为相似判断函数（简称相似函数)。

同义词集合（Synonym Sets） 对于给定的单词 w （$w \in SW$），定义同义词集合为：$SWS = \{S_{w_i} \mid w_i \in S_w\}$。在我们的系统中对于 SW，我们取英文字典中的所有单词，其原因是：如仅包含单词 W(D)，若我们查询单词“document”时，而文中不包括“document”仅包含“paper”，则输入则无效，这将大大降低方案的可用性。在实际中，我们可以根据环境的不同来定义 SW，甚至我们能动态地根据环境信息搜索建立 SW 集合。

信息泄漏 在我们的方案中，我们主要分析我们场景下的信息泄漏情况，我们定义我们的信息泄漏包括文档大小、访问模式和搜索模式。这里我们不考虑文档大小，仅从搜索模式和访问模式信息泄露进行分析。

查询历史： 给定文档集 D，定义 q 次查询的历史 $H = (D, w)$，w 是 q 个单词的向量 $w = (w_1, w_2, ..., w_q)$。

搜索模式： 给定文档集 D 和 q 次查询历史 H，定义搜索模式:

$S(H) = \begin{bmatrix} x_{1,1} & x_{1,2} & ... & x_{1,q} \\ x_{2,1} & x_{2,2} & ... & x_{2,q} \\ . & . & ... & . \\ . & . & ... & . \\ . & . & ... & . \\ x_{q,1} & x_{q,2} & ... & x_{q,q} \end{bmatrix}$，其中 $x_{i,j}$ 可能取值 0，1，2。若 $x_{i,j}$ 为 0 — 表示 w_i 和 w_j 为同义词；取值 1 — 表示 w_i 和 w_j 不为同义词但是 $SD(w_i)$ 和 $SD(w_j)$ 相等；取值 2 — 表示 w_i 和 w_j 不为同义词且 $SD(w_i)$ 和 $SD(w_j)$ 不相等。

访问模式： 对于给定文档集 D 和 q 次查询历史 H，定义访问模式 $A(H) = (SD(w_1), SD(w_2), ...SD(_q))$，$SD(w_i) = \{D(w_j) \mid w_j \in S_w\}$。

4.1.5 方案描述

在我们的方案中，主要解决对称可搜索加密环境下同义词搜索（SSSE）的场景，即输入单词 w，即返回包含单词 w 的文档，同时返回包括和单词 w 有相同函数的文档的问题。下面简单描述我们的方案。

Synonym Searchable Encryption（SSSE） 方案包括 5 个基本的模块：SSSE = （SSKeyGen, SSEnc, SSTrapdoor, SSSearch, SSDec）。

SSKeyGen: 是一个概率算法，输入安全参数 sk 和输入私钥密钥 K。

SSEnc: 是方案中主要的加密模块。包括对文档进行加密（SSDEnc）和建立安全索引（SSBuildSI）。

SSTrapdoor: 是一个确定性算法（可能是概率算法）。对于一个给定的单词 w，在密钥 K 下，生成陷门 T_w。

SSSearch: 是一个确定性模块。对于一个给定的陷门 T_w 和安全索引，查询并输出同义词结果集 ESR_w。

SSDec: 是一个确定性的算法。输入返回的结果 ESR_w 和加密密钥 K，并输入解密的文档 PD_w。

4.2 算法框架及细节

在我们方案设计描述之中，我们仅仅考虑一个简单的情况 — SW 中每个单词仅存在一个含义。基于这样的假设，我们描述方案中各模块的详细实现流程，和方案中所涉及到的算法。最后我们阐述了如何将我们方案应用于一词多义的情形并分析我们方案的优势。

4.2.1 框架详细描述

同义词对称可搜索加密： 同义词对称可搜索加密方案由五个模块组成 — （SSKeyGen, SSEnc, SSTrapdoor, SSSearch, SSDec）。SSKeyGen 用于生成密钥；SSEnc 模块由（SSInitSets — 用于初始化文档, SSDEnc — 加密文档, SSBuildSI — 对文档建立安全索引）三部分组成；SSTrapdoor 生成单词的陷门；SSSearch 用户查找陷门；SSDec 则是简单的解密过程。

在详细地的分析我们方案之前，我们首先定义函数 F, G 和 Q 表示伪随机函数，函数 H 表示伪随机置换。我们使用数组 A_t 和 A_s 来存储安全索引信息，其中 A_t — 单词的安全索引信息，A_s — 文档 ID 的安全索引信息)。

- $\{K\} \leftarrow SSKeyGen(1^{sk})$: outputs K = (K_1,K_2,K_3,K_4,SK), where K_1, K_2, K_3 and K_2 take uniformly samples from $\{0,1\}^{sk}$ and $SK \leftarrow SKE.Gen(1^{sk})$.

- $\{SE, SWS, ED\} \leftarrow SSEnc(D, K)$: consists of these algorithms (SSInitSets, SSDEnc, SSBuildSI). We respectively run them in certain sort.

  – $(SWS, SID) \leftarrow SSInitSets(D, WA)$.

    1. $W(D) \leftarrow D$, browse the whole D and generate the all different keyword sets.
    2. forms the $SID_w (SID_w = \{ID(w_i) | w_i \in D_i\})$ by scanning the documents D, for each keyword w ($w \in W(D)$). We define: $SID = \{SID_{w_i} | w_i \in W(D)\}$
    3. for each keyword w in the W(D), we check if $SF(w, w_i)$ is 1, where $w_i \in WA$, and insert all keywords of the result 1 into S_w. At last, we define: $SWS = \{S_{w_i} | w_i \in WA\}$, if $w_i \notin W(D)$, and set only single element as set.

  – $(KED) \leftarrow SSDEnc(SK, D)$. For every D_i that belong to set D, we encrypt it by $ED_i \leftarrow SKE.Enc(SK, D_i)$. Then we define: ED = $\{ED_i\}$, and for ED_i, we set: $KED_i =< ID(ED_i), ED_i >$ and $KED = KED_i$.

  – $(SI) \leftarrow SSBuildSI(K, SWS, SID)$. In this part, we build the secure index in algorithm 1. (Note: in algorithm 1, when $|S_w| < max|SWS|$ $(max|SWS| = max\{S_{w_i} | w_i \in W(D)\})$, then insert the remaining differences of $max|SWS| - |S_w|$ to random keywords while maintaining its unique.

  At last, outsource OD ($OD = (SI, KED)$) to the remote server.

- $\{T_w\} \leftarrow SSTrapdoor(w, K)$: outputs T_w = ($F_{K_1}(w)$, $G_{K_2}(w)$, $P_{K_3}(w)$, $Q_{K_4}(w)$), and commits it to the server.

- $\{ESR_w\} \leftarrow SSSearch(T_w, SI, ED)$: upon receiving the the trapdoor of the remotely server, then search on the secure index and return the answer to the user. It mainly is implemented by algorithm SSSearchESR, we will describe it in algorithm 2.

- $\{PSD_w\} \leftarrow SSDec(SK, ESR_w)$: decrypts ESR_w and outputs PSD_w, where $PSD_w = \{SKE.Dec_{SK}(R_i) \mid R_i \in ESR_w\}$.

4.2.2 算法描述

安全索引建立算法（SSBuildSI）：安全索引建立算法是一个确定性的算法，在数据拥有者外包数据之前，由客户所完成。当数据所有者想要外包数据时，首先他必须通过将外包的数据来建立索引，在建立索引过程中，使用文档和同义词字典，最终以数组 A_s 和 A_t 的形式输入，同时建立了文档 ID 与密文文档的联系。具体的描述细节见 Algorithm1.

同义词查找算法（SSSearchESR）：同义词查找算法是确定性算法，处在查找阶段。当服务器收到用户提交的单词陷门时，在安全索引中进行查找文档 ID，最后通过文档 ID，得到密文的文档并将结果返回给用户。详细的算法系统细节如 Algorithm2。

4.3 安全性证明

4.4 性能分析

为此，我们构建的同义词可搜索加密方案已实现。首先，我们能实现了本方案的基本功能 —支持同义词搜索；接着我们构建了一个可扩展的安全性更强的方案同时定义了一些该方案所用到的基本知识；最后，最后我们对本方案进行了严格的安全性分析和详细的证明过程。最终，我们证明我们的方案达到我们的预期要求。但是，在我们上述的方案中，我们缺乏对方案的详细性能分析模块。考虑到在对称可搜索加密环境下的模型 —用户、数据拥有者和远程云服务器，下面我们分别从三个方面 —存储开销、计算开销和通讯开销，对我们的方案进行详细的性能分析。

4.4.1 存储开销

在明文可搜索方案中，为了支持同义词搜索，搜索引擎不得不额外承担计算并存储单词的同义词集合，这样显然在原有的可搜索环境下大大地增加了一定的存储开销。为此，根据明文同义词搜索环境的特征，我们总结：为增加同义词搜索功能，不得不付出额外的存储代价（在客户或者服务端），这也正好与我们在第二部分描述的香农信息论定理相一致 —为得到额外的功能，必须要在存储性能方面有所折中。

在我们的方案中，由于我们要增加同义词功能并且要保证数据的隐私，我们清楚地了解到支付额外的存储开销是使我们方案支持同义词搜索的必要条件，这也是我们方案所采用的策略。我们的方案主要由（SSKeyGen, SSEnc, SSTrapdoor, SSSearch, SSDec）五部分组成。在 SSKeyGen、SSTrapdoor、SSearch 和 SSDec 这些模块中，仅仅用于辅助作用（与方案的存储开销没有任何关联）；SSKeyGen—仅仅用于生成密钥；SSearch—用于搜索并返回结果集；而 SSDec 则仅用于加密返回的结果集。因而，在我们的方案中，仅仅 SSDec 与系统的存储性能息息相关。在 SSDec 模块中，我们包括了三个算法（SSInitSets, SSDEnc, SSBuildSI），在算法 SSDEnc 中，我们仅仅加密了明文文档，与前人的可搜索加密方案一致，下面我们详细地分析算法 SSInitSets 和 SSBuildSI.

在 SSInitSets 中，我们需要对每个单词建立同义字集合（即对每个单词 w，我们需要构造它们的同义字集 $S_w = \{w_i | SF(w_i, w) = 1\}$），我们知道在原有的可搜索加密方案中，并不需要建立同义字集合，因而我们方案与基本方案相比，同义字集合的增加是必不可少的。对于每个单词 w，我们知道产生同义字集合的大小为 $|S_w|$, 所有单词所构成的同义字总大小为 $SO(SSSE) = \sum_{i=1}^{|W(D)|}(|S_{w_i}|)$（其中，|W(D)|—指文档中所有不同单词集合的大小）。我们清楚地知道同义字的总大小随着文档的动态增加而增大，并且在不同领域或环境下，我们为了增进方案的可用性和可扩展性，我们所定义的同义词集合大小还会有所增加。这仅仅是没有考虑服务器可能会暗地里偷偷的查看我们的信息—这样服务器在查询过程中，根据长度来判断同义词的语义而导致信息泄漏。为了保证服务器不能通过同义词集合的长度来推断我们的信息，我们必须对每个同义词集合插入一些冗余信息以保证所有同义词集合的大小相同，因为我们得到同义词集合的新的大小为 $MAXSO(SSSE) = max(|SWS|) * |W(D)|)$，冗余信息的存储大小为 $MAXSO(SSSE) - SO(SSSE)$. 另外，在算法 SSBuildSI 过程中，为了增加同义词搜索功能，我们在索引数组中的每个节点增加了信息 $addr(w_{i+1}), G_{K_2}(w_{i+1}) \oplus Q_{K_4}(w_i)$，而这些信息与在 SSInitSets 过程中生成的同义词集合大小相关，因此，我们方案增加的存储开销仅与同义词集合的大小相关。从上面分析中，简单看似我们在索引数组中每个条目的开销是原有方案的三倍并且要增加每个单词的同义词集合的大小，但是对于大量文档的存储语境来说可以忽略不计，下面我们用一个实例来进行推断。

在这个例子中，我们假设用户存储文档的数目为 1000，每个文档的大小为

500KB。在这些文档中，假设不同的单词的数目是 1000，而对于每个单词 w，他们的同义词集合大小为 10，并且在存储同义词索引信息时，对于每个单词计算他们的哈希值的大小为 128bits(16bytes)，而存储 $addr(w_{i+1}), G_{K_2}(w_{i+1}) \oplus Q_{K_4}(w_i)$ 占用的存储开销为 32bytes。因此，我们的总的存储信息大约为大小大约为 6MB，而在我们不支持模糊搜索方案中大小仅仅是 2MB，文档的信息大小为 500MB。在不考虑文档大小和文档 ID 存储的情况下，因为我们在方案中增加的存储开销是很大的（大概是原有方案的三倍)。但是当我们将放入整个方案中时，模糊单词集合的大小显得忽略不计。因此从存储开销着手，我们方案与原有方案 —不支持模糊搜索的方案性能相当。

综上，我们的方案与不支持同义词搜索的方案在存储性能上基本持平，仅仅在索引数组的构建存储上有所增加，大概为原有的 3 倍以上，而整个方案在文档不太小的情况下，基本与基本可搜索方面为 1：1；显然，这样的存储性能增加，对于增加同义词搜索功能的方案来说，我们是可以接收并得到应用的。

4.4.2　计算开销

同义词可搜搜加密方案的计算过程贯穿于整个方案中，而 SSKeyGen、SSDEnc 和 SSDec 都是基本的对称可搜索加密算法，在此我们不作讨论，我们主要分析方案中 SSInitSets、SSBuildSI、SSSearchESR 的性能计算开销。

- **SSInitSets 过程。** 在 $(SWS, SID) \leftarrow SSInitSets(D, WA)$. 过程中，我们逐个遍历文档，并且对每个中每个已出现的单词维持一个数据，在这个数据中元素是包含这个单词的文档的 ID。与此同时，对每个单词 w，遍历同义词字典，找出所有相同含义的单词形成同义词集合 S_w。在算法描述中，我们知道，其计算时间复杂度为：$\sum(|D_i|) + W(D) * |WA|$ $(D_i \in D)$.

- **SSBuildSI 过程。** 客户端的计算时间主要集中的 SSBuildSI 阶段，在这个阶段的输出结果为安全索引 $SI = (A_t, A_s)$，因而我们可以简单地分析为主要的计算时间花费在对这两个数组的填充过程中，大约为：$|A_t| + |A_s|$. 从算法 1 中，我们可以清楚地评估出，构建 A_s 所用的时间复杂度为 $|W(SWS)| * |S_w| * max(|SIDS|)$，而构建数据结构 A_t 所用的时间复杂度为：$|W(SWS)| * |S_w|$.

- **SSSearchESR 过程。** 在构建好安全索引和数据被外包后，数据的交互在于 SSSearchESR 部分，因而这部分的算法性能指标对整个系统非常重要。从算法 2 中，我们算法主要集中在如何解析出待查单词的所有的同义词集并对其中每个单词查找所包含的文章，我们评估出我们在查找这部分的时间复杂度为：$|S_w| * |SID_w|$.

在我们的方案中，SSInitSets 的计算开销主要花在建立同义词集合阶段，并且仅仅在初始时才计算一次，以后保持不变；SSBuildSI 的计算开销主要是通过同义词集合建立安全索索引，也仅仅是一次性计算；而 SSSearchESR 过程中，每次我们查询都需要查询，并且查询主要消耗在遍历索引数组和单词对应文档 ID 的数组过程中。综上，我们方案的主要计算开销是 SSSearchESR 过程，我们知道这个过程的计算开销为 $|S_w| * |SID_w|$ （$|SID_w|$ — 指单词 w 所在文档的数目），与原有的非同义词可搜索方案相比，仅仅增加了对同义词集合的访问，而每个单词同义词的数目也不大，并且在不同场景下，可一次的书目会更小，因而我们是完全可以接受的；除此之外，我们还在我们的补充方案中提出了基于 trie 树结构的方案，这样讲讲进一步减小我们的查找开销。

4.4.3 传输开销

在对称可搜索加密场景下，为了进行在远程的服务器之间进行存储和搜索，我们不得不在在客户和服务器之间进行通讯。同理在我们的对称可搜索加密中，方案的传输开销主要来源于与服务器之间的通信。下面我们从三个方面的方案的通信开销进行阐述 — 客户外包数据、授权用户提交的搜索陷门和服务器的返回结果。

1. **外包的数据。**数据拥有者完成对待外包的数据进行安全操作后 — 主要是建立安全索引和加密文档数据过程。外包数据的通信开销主要发生在，数据拥有者将加密文档和安全索引提交给服务器的过程中。外包的数据主要包括加密文档和安全索引，我们知道要想外包数据，对文档的提交是必不可少的，因此这里我们主要分析外包的安全索引的大小。在 SSBuildSI 过程中，安全索引包括两部分信息（索引数组和每个单词对应的文档 ID 的数组）；对于索引数组，根据在存储开销部分的分析，我们知道它的大小为：$|W(D)| * max(|S_w|)$ — 即所有不同单词的大小和单词

同义词集最大长度的乘积。而对于文档 ID 链表所形成数组的大小主要为取决于每次单词被文档包含的次数，表示为：$max(|SID_{w_i}|) * |W(D)|$ $(w_i \in W(D))$ — 即单词的数目与所有单词被文档包含最多次数的乘积。这些信息会因支持同义词搜索而比原有方案有所增加。

2. **搜索陷门信息。** 当授权用户对外包数据进行搜索时，他们将提交单词的陷门给服务器。在我们方案中，提交单词 w 的陷门信息是：T_w = ($F_{K_1}(w), G_{K_2}(w), P_{K_3}(w), Q_{K_4}(w)$)。我们显然知道搜索的陷门是固定的四个哈希值，在任何时候保持不变，因而这样的信心是合理并且高效的。与非同义词可搜索加密技术相比，我们的方案传输的信息会增加一些，但是这对于一个功能增加的方案来说显得微不足道，我们认为这样的设计是非常合理并且有效的。

3. **待查单词的响应结果集。** 在算法 SSSearchESR 中，服务器得到结果后，将其返回给请求的用户。在这个返回过程中，服务器响应的结果信息包括所有包含待查单词同义词的文档，其最大开销是与在存储时每个单词的最大存储开销一致，即 $max(|SWS|) * |W(D)|$)。这个结果与简单的可搜索加密方案相比，显然会大得多，由于我们的方案是实现同义词查找的情形，因而其增加的开销也是必要的。

综上，我们方案中各个阶段的传输开销都是必要的，并且不存在冗余信息，是实现同义词搜索必不可少的。并且外包的数据通讯虽然较简单方案有所增加，但是这样的通讯是一次性的，仅在初始外包数据时才传输，并不会影响用户使用性能和增加网络的有用流量。而在陷门提交阶段，传送的仅仅是哈希信息值，对网络负荷毫无影响。而在响应结果集中，这样的开销增加是必须的，并且是有效的增加，返回的信息都是用户所需要的。

Algorithm 1 SSBuildSI

Require:

K **:** the secret key to encrypt the synonym sets.

SWS **:** the set of the synonym set.

SID **:** the set of the pair, that is make up of keyword and document identity sets where corresponding document of each one contains the given keyword.

Ensure:

while $w \in W(SWS)$ **do**
  search in SWS and get S_w
  while $w_i \in S_w$ **do**
    check the pair of <key,value> in SWS, and get SID_{w_i} that is abbreviated as set $WISID$.
    create a list L_{w_i}, for each SID_{w_i}.
    while $ID(D_j) \in WISID$ **do**
      construct it as the following form: N_i = (<$ID(D_i)$, $addr(N_{i+1})$> $\oplus$ $H1(P_{K_3}(w_i), r_i), r_i)$; Put it in array A_s by randomly selection but assuring the blank unique, and these nodes form a list L_{w_i}.
      fill the remaining elements with the random strings for WISID, whose size is the difference max(|SID|) and |WISID|.
    end while
    for each given keyword w_i, form its construction into the look-up table A_t, as follows: $A_t[F_{K_1}(w_i)]$ = ($<$ $addr(N_1)$,$addr(w_{i+1})$,$G_{K_2}(w_{i+1}) \oplus Q_{K_4}(w_i)$>$\oplus$ $G_{K_2}(w_i)$), where $addr_{N_1}$ is the first node address of the list generated by keyword w_i in A_s and $addr_{w_{i+1}}$ is the next address of the synonym of w_i in A_t, and $A_t[F_{K_1}(w_{max(|SWS|)})]$ = ($<$ $addr(N_1)$, $addr(w_1)$, $G_{K_2}(w_1) \oplus Q_{K_4}(w_{max(|SWS|)}) > \oplus G_{K_2}(w_{max(|SWS|)})$) ; The retaining elements are handled in the same manner. That is, each synonym set forms a circled list.
  end while
end while
set: $SI = (A_t, A_s)$
return SI;

Algorithm 2 SSSearchESR

Require:

T_w **:** the trapdoor of the keyword w, that is generated by secure encryption function.

SI **:** the secure index of the solution, that is builded in the phrase of SSBuildSI.

KED **:** the pair of encrypted document sets and its ID that are formed in the algorithm SSDEnc.

Ensure:

1: parse T_w as (T_1, T_2, T_3, T_4), and let $M1 = A_t[T_1]$.

2: compute and set $A_{M1} = M1 \oplus T_2$, and parse $A_{M1} = (M1_1, M1_2, M1_3)$.

3: get all document ID by parsing T_3 and $M1_2$, we set: $ID(M1) = \{ID_i \mid \text{ith ID in parsing } L_{M1}\}$

4: set $M2 = A_t[M1_3]$.

5: compute $A_{M2} = T_4 \oplus M1_3 \oplus M2$, and parse $A_{M2} = (M2_1, M2_2, M2_3)$.

6: Loop in steps [2-5] until finishing the search

7: parse $ED_j = \{KED[ID_i] \mid ID_i \in ID(M_i)\}$

8: set $ESR_w = \{ED_j, j \in [max(|SWS|)]\}$

9: **return** ESR_w;

第五章 总结与展望

5.1 全文总结

本文首先总结和分析了云计算环境下的对称可搜索加密技术，包括对称可搜索加密中的各个子问题（精确搜索、模糊搜索、范围搜索、布尔搜索以及动态搜索)；然后详细地分析了对称可搜索加密环境下的模糊搜索加密技术的研究现状和主要的研究内容，以及对该方案中的不足进行了阐述。通过对对称搜索环境下已有方案的研究，我们总结了该环境下普遍存在的一个安全缺陷 —关于搜索模式的信息泄漏的问题，并针对该问题提出了一个能避免在查找过程中搜索模式信息泄漏的方案 —即抗信息泄漏的可搜索加密技术（基于史密斯正交化的原理)。在现有对称可搜索机密技术方案之外，我们挖掘出了该语境下一个新的复杂对称可搜索技术问题 — 同义词对称可搜索加密技术；在文中我们阐述了同义词搜索月模糊搜索的联系与区别。随之通过对该问题的详细分析和设计，我们提出了一个确保低通讯开销、少信息泄漏兼高搜索性能的同义词搜索方案；在方案中，我们引入了同义词集合和同义词判断函数，并且详细地描述了方案的算法实现和证明了方案的安全性。为了验证我们同义词搜索方案中的返回结果的正确性，我们提出了结果可验证的同义词搜索加密方案。在该方案中，我们对方案中的返回结果进行了正确性验证，证明我们方案的可靠性。

综合上述问题，在文章中我们详细地介绍了如何简单地将抗搜索模式的信息泄漏方案应用于同义词可搜索方案，进一步使我们的方案更加健壮和安全。这使得我们的方案在云计算环境领域下，不仅支持了原有的理论基础并且有着远大的现实意义，甚至这将使得原有方案更好完善，是理论向显示又迈进了伟大的一篇。

5.2 未来展望

在大数据时代的背景下，云计算技术正以着飞速的速度发展和应用。但是安全的云计算可搜索加密方案仍处于理论的阶段，虽然有些安全的云计算系统

已被开发，但是不足以面对现实的需求，主要由于这些系统难以在可应用和强安全性之间达到平衡 — 可用性好不够安全，而足够安全则功能太过于单一而不被使用，并且这些系统功能也非常单一。为了开发出兼容诸多优势的方案，我们还有很大一段距离要走 — 主要是还有很多难题需要被提出和进一步研究。从本文的研究方向上来看，主要可以从如下几个方向进行进一步研究：

1. 在我们的同义词可搜索加密方案中，并不能实现同义词随着上下文环境的变化而变化 — 即同一份文档集在不同环境下有不同的同义词集，并且同义词随着环境的变化和改变。即我们需要实现一个方案。最终我们希望在将来，能构建具有这样功能的同义词可搜索加密方案。

2. 通过我们的方案的详细描述，我们了解了同义词和模糊搜索之间的紧密联系。接下来我们希望将同义词搜索和模糊搜索结合起来，实现一个确保安全的相似可搜索加密方案 — 同时兼容模糊和同义词搜索功能。

3. 由于现有的模糊搜索方案都是基于通配符的解决方案，这些方案有一个明显的缺陷 — 高存储开销。接下俩，我们希望能提出一个低存储的模糊搜索方案。

4. 我们希望将我们的方案扩展到动态的场景下 — 同时支持文档的动态的修改和同义词搜索，使方案更具可实现性和应用性。

附录 A 模板更新记录

2012 年 12 月 27 日 v0.5.2 发布，更正拼写错误：从“个人建立”更正为“个人简历”。在 diss.tex 加入 ack.tex，更名后忘了引用。

2012 年 12 月 21 日 v0.5.1 发布，在 LaTeX 命令和中文字符之间留了空格，在 Makefile 中增加 release 功能。

2012 年 12 月 5 日 v0.5 发布，修改说明文件的措辞，更正 Makefile 文件，使用 metalog 宏包替换 xltxtra 宏包，使用 mathtools 宏包替换 amsmath 宏包，移除了所有 CJKtilde(~) 符号。

2012 年 5 月 30 日 v0.4 发布，包含交大学士、硕士、博士学位论文模板。模板在github上管理和更新。

2010 年 12 月 5 日 v0.3a 发布，移植到 XeLaTeX/LaTeX 上。

2009 年 12 月 25 日 v0.2a 发布，模板由 CASthesis 改名为 sjtumaster。在 diss.tex 中可以方便地改变正文字号、切换但双面打印。增加了不编号的一章“全文总结”。添加了可伸缩符号 (等号、箭头) 的例子，增加了长标题换行的例子。

2009 年 11 月 20 日 v0.1c 发布，增加了 Linux 下使用 ctex 宏包的注意事项、.bib 条目的规范要求，修正了 ctexbook 与 listings 共同使用时的断页错误。

2009 年 11 月 13 日 v0.1b 发布，完善了模板使用说明，增加了定理环境、并列子图、三线表格的例子。

2009 年 11 月 12 日 上海交通大学硕士学位论文 LaTeX 模板发布，版本 0.1a。

附录 B Maxwell Equations

选择二维情况，有如下的偏振矢量

$$\mathbf{E} = E_z(r,\theta)\hat{\mathbf{z}} \tag{B–1a}$$
$$\mathbf{H} = H_r(r,\theta))\hat{\mathbf{r}} + H_\theta(r,\theta)\hat{\boldsymbol{\theta}} \tag{B–1b}$$

对上式求旋度

$$\nabla\times\mathbf{E} = \frac{1}{r}\frac{\partial E_z}{\partial\theta}\hat{\mathbf{r}} - \frac{\partial E_z}{\partial r}\hat{\boldsymbol{\theta}} \tag{B–2a}$$
$$\nabla\times\mathbf{H} = \left[\frac{1}{r}\frac{\partial}{\partial r}(rH_\theta) - \frac{1}{r}\frac{\partial H_r}{\partial\theta}\right]\hat{\mathbf{z}} \tag{B–2b}$$

因为在柱坐标系下，$\bar{\bar{\mu}}$ 是对角的，所以 Maxwell 方程组中电场 $\mathbf{E}$ 的旋度

$$\nabla\times\mathbf{E} = \mathbf{i}\omega\mathbf{B} \tag{B–3a}$$
$$\frac{1}{r}\frac{\partial E_z}{\partial\theta}\hat{\mathbf{r}} - \frac{\partial E_z}{\partial r}\hat{\boldsymbol{\theta}} = \mathbf{i}\omega\mu_r H_r\hat{\mathbf{r}} + \mathbf{i}\omega\mu_\theta H_\theta\hat{\boldsymbol{\theta}} \tag{B–3b}$$

所以 $\mathbf{H}$ 的各个分量可以写为:

$$H_r = \frac{1}{\mathbf{i}\omega\mu_r}\frac{1}{r}\frac{\partial E_z}{\partial\theta} \tag{B–4a}$$
$$H_\theta = -\frac{1}{\mathbf{i}\omega\mu_\theta}\frac{\partial E_z}{\partial r} \tag{B–4b}$$

同样地，在柱坐标系下，$\bar{\bar{\epsilon}}$ 是对角的，所以 Maxwell 方程组中磁场 $\mathbf{H}$ 的旋度

$$\nabla\times\mathbf{H} = -\mathbf{i}\omega\mathbf{D} \tag{B–5a}$$
$$\left[\frac{1}{r}\frac{\partial}{\partial r}(rH_\theta) - \frac{1}{r}\frac{\partial H_r}{\partial\theta}\right]\hat{\mathbf{z}} = -\mathbf{i}\omega\bar{\bar{\epsilon}}\mathbf{E} = -\mathbf{i}\omega\epsilon_z E_z\hat{\mathbf{z}} \tag{B–5b}$$
$$\frac{1}{r}\frac{\partial}{\partial r}(rH_\theta) - \frac{1}{r}\frac{\partial H_r}{\partial\theta} = -\mathbf{i}\omega\epsilon_z E_z \tag{B–5c}$$

由此我们可以得到关于 E_z 的波函数方程:

$$\frac{1}{\mu_\theta\epsilon_z}\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial E_z}{\partial r}\right) + \frac{1}{\mu_r\epsilon_z}\frac{1}{r^2}\frac{\partial^2 E_z}{\partial\theta^2} + \omega^2 E_z = 0 \tag{B–6}$$

参考文献

[1] Walker E. Benchmarking Amazon EC2 for high-performance scientific computing[J]. Usenix Login, 2008, 33(5):18–23.

[2] Bogatin D. Google CEO's new paradigm:'cloud computing and advertising go hand-in-hand'[J]. ZDNet.[Online]. Available: http://blogs. zdnet. com/micro-markets, 2006.

[3] Mell P, Grance T. The NIST definition of cloud computing[J]. National Institute of Standards and Technology, 2009, 53(6):50.

[4] Fox A, Griffith R, Katz R. Above the clouds: A Berkeley view of cloud computing[J]. Dept. Electrical Eng. and Comput. Sciences, 2009, 28:13.

[5] Chiueh S N T c, Brook S. A survey on virtualization technologies[J]. RPE Report, 2005:1–42.

[6] Ristenpart T, Tromer E, Shacham H, et al. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds[J]. 2009:199–212.

[7] Biham E, Shamir A. Differential cryptanalysis of the data encryption standard[M], Vol. 28.[S.l.]: [s.n.] , 1993.

[8] Grefenstette G. Explorations in automatic thesaurus discovery[M].[S.l.]: [s.n.] , 1994.

[9] Song D X, Wagner D, Perrig A. Practical techniques for searches on encrypted data[J]. 2000:44–55.

[10] Goh E J, et al. Secure Indexes.[J]. IACR Cryptology ePrint Archive, 2003, 2003:216.

[11] Chang Y C, Mitzenmacher M. Privacy preserving keyword searches on remote encrypted data[J]. 2005:442–455.

[12] Curtmola R, Garay J, Kamara. Searchable symmetric encryption: improved definitions and efficient constructions[J]. 2006:79–88.

[13] Chai Q, Gong G. Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers[J]. Communications ICC, 2012 IEEE International Conference, 2012:917–922.

[14] Kurosawa K, Ohtaki Y. UC-secure searchable symmetric encryption[J]. Financial Cryptography and Data Security, 2012:285–298.

[15] Chase M, Kamara S. Structured encryption and controlled disclosure[J]. Advances in Cryptology-ASIACRYPT 2010, 2010:577–594.

[16] Boneh D, Franklin M. Identity-based encryption from the Weil pairing[J]. SIAM Journal on Computing, 2003, 32(3):586–615.

[17] Golle P, Staddon J, Waters B. Secure conjunctive keyword search over encrypted data[J]. 2004:31–45.

[18] Ballard L, Kamara S, Monrose F. Achieving efficient conjunctive keyword searches over encrypted data[J]. 2005:414–426.

[19] Shamir A. How to share a secret[J]. Communications of the ACM, 1979, 22(11):612–613.

[20] Moataz T, Shikfa A. Boolean symmetric searchable encryption[J]. 2013:265–276.

[21] Li J, Wang Q, Wang C, et al. Fuzzy keyword search over encrypted data in cloud computing[J]. INFOCOM, 2010 Proceedings IEEE, 2010:1–5.

[22] Wang C, Ren K, Yu S, et al. Achieving usable and privacy-assured similarity search over outsourced cloud data[J]. INFOCOM, 2012 Proceedings IEEE, 2012:451–459.

[23] Kuzu M, Islam M S, Kantarcioglu M. Efficient similarity search over encrypted data[J]. 2012:1156–1167.

[24] Indyk P, Motwani R. Approximate nearest neighbors: towards removing the curse of dimensionality[J]. 1998:604–613.

[25] Chuah M, Hu W. Privacy-aware bedtree based solution for fuzzy multi-keyword search over encrypted data[J]. 2011:273–281.

[26] Wang C, Cao N, Li J, et al. Secure ranked keyword search over encrypted cloud data[J]. 2010:253–262.

[27] Wang C, Cao N, Ren K, et al. Enabling secure and efficient ranked keyword search over outsourced cloud data[J]. Parallel and Distributed Systems, IEEE Transactions on, 2012, 23(8):1467–1479.

[28] Boldyreva A, Chenette N, O'Neill A. Order-preserving encryption revisited: Improved security analysis and alternative solutions[J]. 2011:578–595.

[29] Cao N, Wang C, Li M, et al. Privacy-preserving multi-keyword ranked search over encrypted cloud data[J]. Parallel and Distributed Systems, IEEE Transactions on, 2014, 25(1):222–233.

[30] Witten I H, Moffat A, Bell T C. Managing gigabytes: compressing and indexing documents and images[M].[S.l.]: [s.n.] , 1999.

[31] Xu Z, Kang W, Li R, et al. Efficient Multi-Keyword Ranked Query on Encrypted Data in the Cloud.[J]. 2012:244–251.

[32] Bellare M, Canetti R, Krawczyk H. Pseudorandom functions revisited: The cascade construction and its concrete security[J]. 1996:514–523.

[33] Bellare M, Rogaway P. Random oracles are practical: A paradigm for designing efficient protocols[J]. 1993:62–73.

[34] Gremillion L L. Designing a Bloom filter for differential file access[J]. Communications of the ACM, 1982, 25(9):600–604.

[35] Jin C, Gu D, Tang Y, et al. Reducing extra storage in searchable symmetric encryption scheme[J]. 2012:255–262.

[36] Van Liesdonk P, Sedghi S, Doumen J, et al. Computationally efficient searchable symmetric encryption[J]. 2010:87–100.

[37] Ji S, Li G, Li C, et al. Efficient interactive fuzzy keyword search[J]. Proceedings of the 18th international conference on World wide web, 2009:371–380.

[38] Li C, Lu J, Lu Y. Efficient merging and filtering algorithms for approximate string searches[J]. Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on, 2008:257–266.

[39] Behm A, Ji S, Li C, et al. Space-constrained gram-based indexing for efficient approximate string search[J]. Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on, 2009:604–615.

[40] Ristad E S, Yianilos P N. Learning string-edit distance[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1998, 20(5):522–532.

[41] Zhang Z, Hadjieleftheriou M, Ooi B C, et al. Bed-tree: an all-purpose index structure for string similarity search based on edit distance[J]. Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, 2010:915–926.

[42] Balamuralikrishna T, Anuradha C, Raghavendrasai N. FUZZY KEYWORD SEARCH OVER ENCRYPTED DATA IN CLOUD COMPUTING[J]. ASIAN JOURNAL OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY, 2013, 1(3).

[43] Kamara S, Papamanthou C, Roeder T. Dynamic searchable symmetric encryption[J]. 2012:965–976.

[44] Kamara S, Papamanthou C. Parallel and dynamic searchable symmetric encryption[J]. 2013:258–274.

[45] Stefanov E, Papamanthou C, Shi E. Practical Dynamic Searchable Encryption with Small Leakage.[J]. IACR Cryptology ePrint Archive, 2013, 2013:832.

[46] Liu C, Zhu L, Wang M, et al. Search pattern leakage in searchable encryption: Attacks and new construction[J]. Information Sciences, 2014, 265:176–188.

[47] Islam M S, Kuzu M, Kantarcioglu M. Access Pattern disclosure on Searchable Encryption: Ramification, Attack and Mitigation.[J]. 2012.

致 谢

在本文中，涉及的研究工作以及论文的撰写过程得到了我的导师丁老师和指导老师陆老师的悉心教导和大力支持。其中丁老师对密码学研究有这身后的功底，而陆老师对可搜索加密的研究方案有的深入和透彻的理解，在这两文老师的指导很帮助下，使得我在密码学理论和可搜索加密的实践都有着和别人不一帮的认识，并且在两位老师的培训下，使得我的这方面的能力有着质的提高。其中最感谢的是谷教授，是我们实验室的顶梁柱，在谷老师的的管理和运作下，使得我们实验室比别人实验室有着天然的优势，有着一群好的老师和同学，并在谷老师的安排下，使得我们 LoCCS 的每个成员保持着良好的关系。另外，谷老师对密码学和信息安全领域有深刻的理解和洞察, 他严谨治学的作风和工作热情, 是我今后学习的导航和奋斗的目标。同时，在这两年多的硕士生涯中, 能够方方面面都有所成长, 得益于谷老师带领下营造了实验室良好的学术氛围、舒适的环境、融洽的人际关系。在每次的学术交流和讨论中总能得到启发, 同时生活上的很多方面, 都得到了谷老师的大力帮助。再次对谷老师表示中心的感谢和敬意。

感谢陆海宁老师在论文写作、项目研发等方面给予的指导, 陆老师对待工作严谨细致的态度、很强的执行力都让我很震撼。您的这些优良品质，是我一直追寻而尚未达到的，您就是我的标杆。

感谢丁宁老师、李晓辉、汤殷琦、熊冶等 LOCCS 成员在科研和生活上对我提供过的帮助。

感谢熊冶师兄，在我最无助是，在论文上对我。

感谢 LOCCS 所有成员的陪伴, 你们每个人都个性鲜明, 从你们每个人身上都学习到了难能可贵的经验。在我的生活和学习中，也正因为有你们参与，才变得更加丰富多彩。

感谢父母对我无私奉献的付出和关爱。感谢各位亲朋好友对我业余生活的充实，有你们未来才更美好，我的生活才更加绚丽多彩。你们的呵护与关爱，是我人生最宝贵的财富；你们一直以来的支持和鼓励，是我前进的最大动力。

其次感谢实验室的各位同学，在日常实验室的学习生活中，与他们的交流

使我受益匪浅。特别感谢赵康同学，在学术上的指导与合作使得我得到了很大的进步。与各位同学度过的两年时光在我看来是非常美好的。

最后感谢父母对我无私奉献的付出和关爱，他们在物质和精神上给我的支持使得我能够更加专心地完成学业。感谢各位亲朋好友对我业余生活的充实，是你们才使我的生活更加美好和绚丽多彩。你们的呵护与关爱，是我人生最宝贵的财富；你们一直以来的支持和鼓励，是我前进的最大动力。同时也祝福你们越活越年轻，在今后的生活和工作中，心想事成、安安康康。

攻读学位期间发表的学术论文目录

[1] CHEN H, CHAN C T. Acoustic cloaking in three dimensions using acoustic metamaterials[J]. Applied Physics Letters, 2007, 91:183518.

[2] CHEN H, WU B I, ZHANG B, et al. Electromagnetic Wave Interactions with a Metamaterial Cloak[J]. Physical Review Letters, 2007, 99(6):63903.

攻读学位期间参与的项目

[1] 973 项目“XXX”

[2] 自然基金项目“XXX”

[3] 国防项目“XXX”