南京中医药大学

本科生毕业论文

人工智能与信息技术学院

计算机科学与技术专业 2016 年级

学　　号 084416261

学生姓名 翟 文 斌

论文题目 基于同态加密的中医药数据安全存储方法设计与实现

实习单位 南 京 中 医 药 大 学

指导教师 丁 有 伟

起止时间 2019.12-2020.5

2020 年 6 月 3 日

目 录

摘 要

随着中医药数据的爆炸式增长和云计算的广泛应用，医药机构逐渐将医药数据的存储和分析由本地迁移到云端进行。由于医药数据具有极高的经济价值和医疗价值，且包含了大量的患者隐私信息，所以医药机构大多将数据加密后再存储到云端。然而传统的加密方法无法对加密后的数据进行直接操作分析，必须先将数据解密得到明文后再分析处理，这就无法保证数据的安全性和隐私性。本文提出了一种基于同态加密的中医药数据存储方法，包括加密、解密和检索算法，利用明文槽对数据进行分组加密后合并、分组解密后合并以及对分组进行密文检索，基于客户端/服务器模式，可以直接对密文进行操作，在保障用户操作有效性的同时，有力的保护用户数据的安全。实验结果显示，该方案在中医药数据存储和操作方面具有不错的安全性和较好的实用性。

关键词：中医药数据；同态加密；明文槽；云计算

Abstract

With the explosive growth of traditional Chinese medicine data and the widespread use of cloud computing, medical institutions have gradually moved the storage and analysis of medical data from the local to the cloud. Because medical data has extremely high economic and medical value, and contains a large amount of patients' privacy information, most medical institutions must encrypt data before storing it in the cloud. However, we cannot directly operate the encrypted data by using the traditional encryption method, and the data must be decrypted to obtain the plain text before analysis and operation, which cannot guarantee the security and privacy of the data. This paper proposes a method for storing Chinese medicine data based on homomorphic encryption, including encryption, decryption and retrieval algorithms, using plaintext slots to merge after packet encryption, merge after packet decryption and perform ciphertext search on packets, based on the Client / Server mode, you can directly operate the ciphertext, while ensuring the effectiveness of users' operations, and the security of user data. Experimental results show that the scheme has good security and good practicability in the storage and operation of Chinese medicine data.

Key words: Chinese medicine data; Homomorphic encryption; Plaintext slot; Cloud computing

1. 绪论

1.1 研究背景

中医药大数据记录了疾病产生、发展、治疗的全过程，其数据具有极高的医疗价值和经济价值的同时，也容易泄露患者的个人隐私。传统的中医药数据通常采用集中式存储和处理分析。近年来，随着智慧医疗的不断推进、医疗数据的爆炸式增长以及中医药数据的流通和共享，传统的集中式存储方式已经不能满足需求，医疗机构逐步将医疗数据从机构内网移至云服务器上，云计算可以通过将医疗数据的存储和计算外包给云计算服务商来降低医疗保健成本。然而由于医药数据的复杂性、多维性、不规范性等性质以及医疗行业信息安全研究起步较晚，且医药数据价值很高，医药数据安全面临着巨大的挑战，医药数据泄露事故层出不穷，数据安全问题刻不容缓。

据 IBM 最新发布的《2019 数据泄露成本报告》[1]显示，医疗行业因其数据的复杂性、隐私性、多态性，平均每条记录的成本和数据泄露的平均总成本连续九年排在首列。医疗数据因为巨大的挖掘价值容易受到不法分子的攻击和窃取，恶意攻击、系统故障、人为失误、云迁移、IT 复杂性和第三方违规等各种因素严重威胁着医疗数据的安全。并且我国医疗行业的数据安全化还不到 50%，2017 年 9 月，暗网出现了我国大量的孕检信息交易，同时我国近 7 亿公民的个人隐私信息遭到泄露。2019 年 2 月，仅一个月美国报告了 32 起医疗数据泄露事件，超过 211 万医疗记录遭到泄露。

2018 年国务院发布了《国家健康医疗大数据标准、安全和服务管理办法（试行）》[2]，其中明确指出在数据存储方面，必须按照相关法律法规的要求进行安全评审；在服务提供方面，服务提供商必须具有确保数据安全的能力，并建有安全管理、隐私保护等制度；在数据利用方面，责任单位必须加强数据的规范应用和服务，不得泄露国家机密、商业机密和个人隐私；在数据共享方面，必须强化对医疗大数据全生命周期的服务与管理。

安全问题是医疗大数据和云计算发展的一个大问题，加密是确保云计算数据安全的核心技术，在医药领域也被广泛采用。为了保证数据的安全性，可以使用传统的数据加密技术将数据存储在云服务器中。传统的数据加密方法有对称数据加密算法和非对称数据加密算法。常用的对称数据加密算法有 DES、3DES、RC2、RC4 等等，常用的非对称数据加密算法有 RSA、DSA 等等。在实际应用中，经常采用 DES、3DES、MD5、数字签名等技术。尽管在私有云的情况下简单的加

密技术也许可以确保数据隐私，但是当使用公共云存储和处理数据时，确保数据隐私变得更有挑战性。同时，当数据以加密形式存储在云服务器时，没有人能对加密数据直接进行操作，无法充分发挥云计算的优势。当客户想要在云服务器上对这些数据执行某些计算时，必须与云服务器共享解密密钥，以明文的形式访问数据，这就带来了数据的不安全性，容易造成数据的泄露。那么，现在的问题是，如何才能对加密数据进行运算的同时确保其安全的？

同态加密可以在无需密钥的情况下直接对加密数据进行计算，同时保持数据的格式、特征和功能，即对加密后密文操作后解密得到的结果与直接对密文进行操作得到的结果是一致的。

1.2 研究现状

中医药数据存储由于起步较晚，发展很不完善。王丽等[3]提出采用 HDFS 实现数据加密，数据只能被客户端加密和解密，但是这种策略既不能保证数据的安全也不能灵活的应用数据；生慧等[4]提出一种基于区块链的数据安全共享方案，采用共识机制和访问控制来保证数据的安全，但是其需要三方的相互信任和协调，可用性不高。不难发现，中医药数据存储安全问题亟需解决。

传统的数据加密算法已经广泛应用于数据存储领域。2016 年，李莹[5]提出了一种基于 AES 算法和 CP-ABE 算法的混合加密算法，并且该算法结合了 Shamir 秘密共享算法来进行数据安全存储保护；2019 年，姜利娟[6]提出了将对称加密算法和非对称加密算法相结合的方案，如 AES-RSA 方案和 AES-ECC 方案；2019 年，赵嘉诚[7]提出了一种基于 MapReduce 的 RSA 和 AES 并行混合加密算法。以上几种数据存储方案虽然能够很好的保护云环境下数据存储时的安全，但是它们都有一个共同的问题，那就是无法直接对密文操作，如果要对数据进行操作，则必须先对密文进行解密，如果解密操作在云端进行则可能导致数据的泄露，如果将数据下载到本地进行解密则无法充分利用云计算的优势。为了解决这个难题，在保证数据安全的同时充分利用云计算的优势，同态加密应运而生。

自 2009 年 Gentry[8]提出第一个真正意义上的全同态加密算法，引发了国外学者对全同态加密方案的研究浪潮。2010 年，Dijk 等[9]提出了整数上的全同态加密方案，简称为 DGHV 方案，该方案主要应用中国剩余定理，一次可加密多位比特数据，但其公约尺寸太大从而导致其效率低下；2011 年 Vaikuntanathan 等提出了基于错误学习 LWE[10]（learning with errors）的全同态加密方案,之后又提出了在其基础上改进的基于环的 R-LWE[11]（ring-learning with errors）的全同态加密方案；2018 年，Acae 等[12]对现有的同态加密方案及其实现进行了系统的介

绍。

2015 年，Kocabas 等[13]使用全同态加密方案对医疗云中的心电图数据进行隐私保护，并进行了同态求平均、最大、最小心率计算，但其密钥尺寸过长，且时间效率低下，难以实际应用；2015 年，MAKKAOUI 等[14]提出了一种基于同态加密的新的 Client-Cloud 体系机构来保证云计算的安全，但其算法复杂度太高，且无法保证服务质量，可操作性较差。2017 年，Jayapandian 等[15]将概率加密和同态加密相结合，在保证安全的前提下，增加了吞吐量并改善了 QoS。

在实际应用方面，IBM 公司推出了 HELib 同态算法库[16-18]并将其开源，微软推出了 SEAL[19]同态加密算法库并将其开源，致力于推动全同态加密的发展。2018 年，安全公司 Enveil 发布其“零泄露计算架构”，实现了同态加密下的搜索，同年 6 月 4 日一家名为 Baffle 公司推出了其“高级数据防护服务”(Advanced Data Protection Service)，同样能让用户搜索和使用加密数据，但两者的数据运算速度很慢且对于应用的限制极大。同态加密具有广阔的应用前景，吸引了广大学者和企业的关注和研究，但由于目前存在着巨大的空间、时间性能瓶颈，极大地限制了同态加密在实际生活中的应用。

国内也对同态加密进行了一定的研究。杨竞[20]对整数上的全同态加密算法进行优化改进，同时将网络编码安全、数字图像信息隐藏以及机器学习的隐私保护等方面和同态加密相结合；何伟超[21]研究了适合机器学习的向量同态加密；吴卓伟[22]提出了一种保证安全性和效率性的整数上的全同态加密方案并提出一种矩阵上的整数全同态加密方案；梁雷元[23]构建了一个针对于大整数更高效便捷的全同态加密方案，并提出一种支持串行和并行浮点数运算的全同态加密算法。

2015 年，李帅[24]将同态加密应用于云安全存储模型中；2018 年，徐文玉[25]将同态加密应用于电子病历的隐私保护模型中；2018 年，郭俊彦[26]设计并实现了基于全同态加密的电子档案管理系统；2019 年，李雅囡[27]将同态加密应用于电子投票系统；2019 年，韩邦等[28]将同态加密应用于全文检索方案的设计和实现,基于 VHE 方案，根据向量空间模型余弦相似性计算，并将其运用到云端密文检索场景，实现了在服务器不可信的场景下对数据进行高效精确检索；2019 年，王利娟[29]基于全同态加密设计了云存储密文检索系统；2019 年，佘维等[30]提出一种基于同态加密和区块链的医疗大数据安全方法，但是其只能进行简单的运算，当涉及到复杂运算时，效率较低。

在同态加密应用方面，国内仍处于初步阶段，沙海信息科技有限公司在其数据存储中应用了同态加密技术，但其只应用了 RSA 乘法同态加密技术，提供的

操作极其受限。

综上所述，虽然全同态加密的应用前景十分广泛，吸引了广大密码学者的关注，但由于其性能瓶颈，全同态加密大多仍处于研究阶段，其应用场景十分受限[31]，而在医疗数据领域，尤其是中医药数据领域，由于数据的多维性、复杂性、不规范性，其应用成果更是少之又少。

1.3 主要研究工作及内容

本文首先对同态加密的概念进行了介绍，同时分别对几种常见的 PHE 方案和 FHE 方案从密钥生成、加密过程、解密过程和同态性质这四个方面进行了较为详细的描述，然后在对同态加密方案研究的基础上提出了改进的加密、解密和检索算法，对其同态性进行了分析。构建实验环境，针对不同的电路深度、不同的明文比特分组进行了详细的对比实验，获取实验结果并对其具体性能进行了分析，最后以浏览器/服务器模式将其集成到中医药数据存储系统中，该系统实现了数据的加密、解密和检索功能，并提供了上传、下载和检索模块。

1.4 论文结构

本文共分为五个章节，详细系统地介绍了基于同态加密的中医药数据安全存储的研究背景、相关知识、算法设计和系统实现等。本文的各个章节安排如下：

第一章绪论，主要介绍云计算的现状及医药数据安全存储的重要性和必要性，传统加密方法与云计算的不适用性，同态加密相比于其他加密方法的优势所在；然后介绍本文的研究重点即同态加密方案的研究现状；最后详细地介绍本文的主要研究工作及内容。

第二章同态加密相关知识，主要介绍同态加密的概念和同态加密的发展历程，以及从原理、特性、方法等方面对几种经典的部分同态加密和全同态加密方案进行系统的介绍，并对 HELib 同态加密算法库进行介绍。

第三章算法设计与实现，主要介绍基于同态加密的中医药数据的加密、解密和检索算法的设计与实现，并从各个不同的角度对算法进行分析与比较。

第四章同态加密技术在中医药数据存储中的应用，主要介绍同态加密算法在中医药数据存储系统中的实现并对实现结果进行分析。

第五章总结与展望，主要介绍本文所做的工作和根据本次的研究课题，对于经典同态加密方案以及研究过程中所学到的知识和感想的总结及对未来研究的展望。

1.5 本章小结

本文首先对研究背景进行了介绍，在云计算环境背景下，通过与传统加密方法的对比得出了同态加密方法的优势所在，接着对国内外同态加密方案的研究现状进行了介绍，最后对本文的主要研究内容和论文结构进行了概述。

2. 同态加密相关知识

2.1 同态加密概念

若一个加密方案，满足 $E(m_1)\Theta E(m_2)=E(m_1\Theta m_2),\forall m_1,m_2\in M$，即对于任意两个明文 m_1,m_2，加密后再进行操作与先进行操作再加密所得到的结果是一致的，则称其对于运算 M 是同态的，其中 $E(m_i)$ 是加密算法，M 是消息集合。同态加密是一种加密方案，它允许第三方（例如云服务提供商等）对加密后的密文数据进行某种运算，同时保留加密数据的格式、特征和功能。同态加密方案根据其特性可分为以下三类：

（1）部分同态加密（Fully Homomorphic Encryption，PHE）仅允许一种或不具备全功能集的几种运算，但操作次数不受限制。

（2）类同态加密（Somewhat Homomorphic Encryption，SWHE）只允许某些类型的运算，并且操作次数有限。

（3）全同态加密（Fully Homomorphic Encryption，FHE）允许无限次无限制的操作。

因为乘法和除法是有限域上的全功能集，所以创建允许进行任何操作的同态方案只需要满足加法和乘法同态。同理，对于布尔电路来说，只需要满足 *XOR* 和 *AND*。

同态加密方案主要包括 *KeyGen*、*Enc*、*Dec* 和 *Eval* 四个基本操作，其中 *KeyGen* 是生成密钥，密钥可以是对称的也可以是不对称的，*Enc* 是加密操作，负责将原始数据加密成密文形式，*Dec* 是解密操作，*Eval* 是同态操作。

2.2 同态加密发展历程

从密码学的历史角度来看，Rivest 等人首次使用了同态一词，并于 1978 年提出了一种同态加密的可能解决方案，但没有进行问题解密。Rivest 给定的同态基础引起了世界各地的研究人员对同态进行了无数次的尝试，并设计了大量的同态方案，可将其发展历程大致分为三个阶段，如图 2.1 所示：

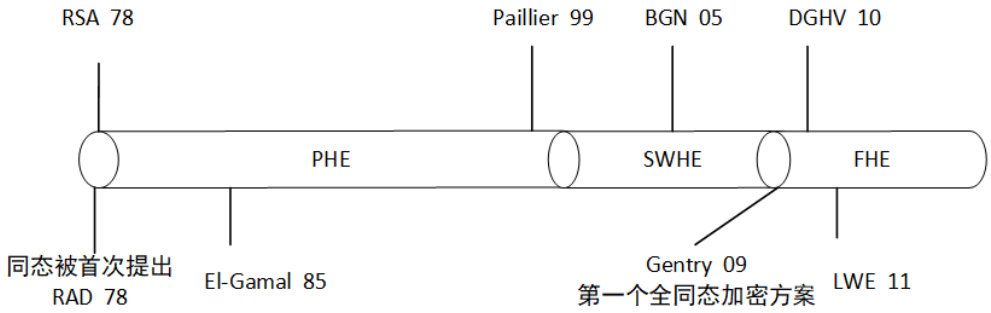

图 2.1 经典同态加密方案时间轴

第一阶段为 1978 年—2009 年：在这 31 年间提出的同态加密方案皆为部分同态加密方案（PHE）和类同态加密方案（SWHE）。PHE 方案已部署在某些应用中，但只能用于其算法仅包含加法或乘法运算的特定应用，例如 1978 年提出的传统的加密算法 RSA 方案具有乘法同态性，1985 年提出的 EIGamal 算法具有乘法同态性，1999 年提出的 Paillier 算法具有加法同态性。SWHE 支持加法和乘法运算，但是在第一个 FHE 方案之前提出的 SWHE 方案允许的同态运算最大数量受到限制，例如 2005 年提出的 BGN 方案支持任意数量的加法，但只支持一次乘法。

第二阶段为 2009 年 Gentry 提出的基于理想格的 FHE 方案[8]，该 HE 方案是真正意义上的第一个 FHE 方案，即支持任意数量的同态运算，此方案基于数学上的理想格，不仅是对方案的描述，而且还是实现 FHE 的强大框架。其提出的压缩和自举被用作构造 FHE 的广泛技术，但是，从概念和实践上来讲，它并不是一个现实的方案，其空间代价和时间代价太过昂贵。

第三阶段为 Brakerski 等于 2011 年提出了基于代数学群的错误学习 LWE（learning with errors）的全同态加密方案[10]，在此方案中，密文和密钥都是向量，每次密文运算后，不需要利用重加密技术进行降噪，而是通过密钥交换技术将密文约减回原来的维数，再通过模交换技术，降低每一次运算扩大的噪声，扩大同态加密的深度。

2.3 经典部分同态加密方案介绍

2.3.1 RSA 算法

RSA 是 PHE 的早期方案，RSA 密码系统的安全性基于两个大质数乘积的因式分解的难易问题。RSA 算法的主要思想如下：

① *KeyGen*：首先取两个大质数 p 和 q 并计算 $n=pq$ 和 $\phi=(p-1)(q-1)$，然后选择 e 以便计算 e 的乘法逆元 d 即 $ed\equiv 1 \bmod \phi$ 和 $\gcd(e,\phi)$。(e,n) 作为公钥，而

(d,n) 作为私钥。

② *Enc*：将消息转换为明文 m，$0<m<n$，采用公式(2.1)进行加密：

$$c=E(m)=m^e \pmod n, \forall m \in M \tag{2.1}$$

③ *Dec*：使用密钥可以将密文 c 转换为明文 m，如公式(2.2)所示：

$$m=D(c)=c^d \pmod n \tag{2.2}$$

④ *Eval*：RSA 算法具备乘法同态性，如公式(2.3)所示：对于 $\forall m_1, m_2 \in M$，

$$E(m_1)\bullet E(m_2)=\left(m_1^e \pmod n\right)\bullet\left(m_2^e \pmod n\right)=\left(m_1\bullet m_2\right)^e \pmod n=E(m_1\bullet m_2) \tag{2.3}$$

RAS 是乘法同态加密算法，使用密文 $E(m_1)$ 和 $E(m_2)$ 进行乘法操作后，其结果 $E(m_1)\bullet E(m_2)$ 与对明文求积再加密后的结果 $E(m_1\bullet m_2)$ 相同，即可以对密文数据直接进行任意乘法操作。

2.3.2 El-Gamal 算法

El-Gamal 算法[32]是原始 Dife-Hellman 密钥交换（Dife and Hellman）算法的改进版本，该算法基于有限域中离散对数的计算难题，其主要用于混合加密系统中，对对称加密系统的密钥进行加密。El-Gamal 算法的主要思想如下：

① *KeyGen*：使用生成器 g 生成阶数为 n 的循环群 G，在循环群中可以使用其中一个元素来生成所有元素。计算 $h=g^y$，其中 y 为随机数且 $y \in Z_n^*$。则公钥为 (G,n,g,h)，私钥为 x。

② *Enc*：使用 g 和 x 来加密明文 m，其中 x 为集合 $\{1,2,...,n-1\}$ 中的随机元素，加密后的密文为 $\left(c=(c_1,c_2)\right)$，如公式(2.4)所示：

$$c=E(m)=\left(g^x, mh^x\right)=\left(g^x, mg^{xy}\right)=(c_1,c_2) \tag{2.4}$$

③ *Dec*：首先计算 $s=c_1^{\,y}$，其中 y 是私钥，则解密如公式(2.5)所示：

$$c_2\bullet s^{-1}=mg^{xy}\bullet g^{-xy}=m \tag{2.5}$$

④ *Eval*：El-Gamal 与 RSA 算法相似，都具有乘法同态性，如公式(2.6)所示：

$$E(m_1)\bullet E(m_2)=\left(g^{x_1}, m_1h^{x_1}\right)\bullet\left(g^{x_2}, m_1h^{x_2}\right)=\left(g^{x_1+x_2}, m_1\bullet m_2h^{x_1+x_2}\right)=E(m_1\bullet m_2) \tag{2.6}$$

2.3.3 Paillier 算法

Paillier 算法[33]基于复合残差问题，算法主要思想如下：

① *KeyGen*：随机选取两个大素数 p 和 q 满足 $\gcd\left(pq,(p-1)(q-1)\right)=1$，计算

$n=pq$ 和 $\lambda=lcm(p-1,q-1)$，其中函数 L 定义为 $L(u)=(u-1)/n,\forall u\in Z_{n^2}^*$，其中 $Z_{n^2}^*$ 为整数模 n^2 的乘法子群。公钥为 (n,g)，私钥为 (p,q)。

② *Enc*：对于明文消息 m，随机选取一个数 r，并使用公式(2.7)进行加密：

$$c=E(m)=g^m r^n \left(\mathrm{mod}\, n^2\right) \tag{2.7}$$

③ *Dec*：对于密文 $c<n^2$，私钥为 (p,q)，使用公式(2.8)进行解密：

$$D(c)=\frac{L\left(c^{\lambda}\left(\mathrm{mod}\, n^2\right)\right)}{L\left(g^{\lambda}\left(\mathrm{mod}\, n^2\right)\right)}\,\mathrm{mod}\, n=m \tag{2.8}$$

④ *Eval*：Paillier 算法是加法同态的，如公式(2.9)所示：

$$\begin{aligned} E(m_1)\bullet E(m_2)&=\left(g^{m_1}r_1^{\,n}\left(\mathrm{mod}\, n^2\right)\right)\bullet\left(g^{m_2}r_2^{\,n}\left(\mathrm{mod}\, n^2\right)\right)\\ &=g^{m_1+m_2}\left(r_1\bullet r_2\right)^n\left(\mathrm{mod}\, n^2\right)\\ &=E(m_1+m_2)\end{aligned} \tag{2.9}$$

Paillier 算法除了在加法运算中具有同态性外，还有一些其他的同态性质，通过使用加密的明文 $E(m_1)$ 和 $E(m_2)$ 以及公钥 (p,q) 可以对明文 $m_1,m_2\in Z_{n^2}^*$ 进行额外的一些操作，如公式(2.10)~(2.12)所示。

$$E(m_1)\bullet E(m_2)\left(\mathrm{mod}\, n^2\right)=E\left(m_1+m_2\left(\mathrm{mod}\, n\right)\right) \tag{2.10}$$

$$E(m_1)\bullet g^{m_2}\left(\mathrm{mod}\, n^2\right)=E\left(m_1+m_2\left(\mathrm{mod}\, n\right)\right) \tag{2.11}$$

$$E(m_1)^{m_2}\left(\mathrm{mod}\, n^2\right)=E\left(m_1m_2\left(\mathrm{mod}\, n\right)\right) \tag{2.12}$$

以上这些附加的同态属性描述了对加密数据和明文操作的相互关系，即体现了对密文的计算是如何影响明文的。

虽然 PHE 方案在同态操作上有诸多限制，但是由于其理论相对简单，在实际生活中已经有诸多应用，表 2.1 列出了其中一些应用。

表 2.1 部分常见的部分加密算法同态性及其应用

加密算法	同态性	实际应用
RSA	乘法同态	确保互联网、银行和信用卡交易的安全
El-Gamal	乘法同态	混合动力系统
Paillier	加法同态	电子投票、门限方案

2.4 经典全同态加密方案介绍

如果一个加密方案允许在密文上进行无限次的操作仍能保持其数据的格式、

特性和功能，则称其为全同态加密方案，即 FHE。虽然同态加密在 1978 年就被提出，但直到 2009 年 Gentry 才在他的博士论文里面提出了第一个 FHE 方案。此方案不仅提供了 FHE 方案，还提供了构造 FHE 的一般框架，然而它一些先进的数学概念难以实现并且实现的时间代价和空间代价太过高昂，所以人们在此基础上又不断地进行了改进。2010 年引入了基于近似 GCD 问题的整数上的 FHE，2011 年提出了基于环上错误学习的 FHE，由于其可能的效率和标准化，2012 年提出了一种类似 NTRU（Number Theory Research Unit）的 FHE，是一种古老且高度标准化的基于格的加密方案。

2.4.1 基于理想格的 FHE 方案

Gentry 在其原有的基于理想格的 SWHE 方案中引入了压缩（squashing）和自举（bootstrapping）技术，从而可以重复进行对密文有限次数的操作[8]，换而言之，就是对密文进行无限次数的操作。Gentry 使用理想和没有格的环来设计同态加密，其中理想是环上保留属性的子集，如偶数。方案中每个理想都用格表示，如理想 $I \in Z[x]/f(x)$，$f(x)$ 为 n 阶，则 I 可以用一列长度为 n，基为 B_I 的理想格表示，基 B_I 将会生成一个 $n \times n$ 的矩阵。Gentry 基于理想环的 SWHE 方案描述如下：

① *KeyGen*：给定环 R 和理想 I 的基 B_I，使用 $IdealGen(R, B_I)$ 生成对 (B_J^{sk}, B_J^{pk})，其中 *IdealGen*() 生成基为 B_I 的理想格的公钥主要部分和密钥，使得 $I+J=R$。则公钥为 $(R, B_I, B_J^{pk}, Samp())$，私钥为 B_J^{sk}，其中 *Samp*() 从理想的陪集中进行采样，通过将理想值偏移一定程度可获得陪集。

② *Enc*：使用公钥 B_{pk} 从理想 L 的“坏”基底中随机选择向量 $\vec{r}$ 和 $\vec{g}$，则对于消息 $\vec{m} \in \{0,1\}^n$ 使用公式(2.13)进行加密：

$$\vec{c} = E(\vec{m}) = \vec{m} + \vec{r} \bullet B_I + \vec{g} \bullet B_J^{pk} \tag{2.13}$$

其中 B_I 为理想 L 的基底，$\vec{m} + \vec{r} \bullet B_I$ 为噪音。

③ *Dec*：使用私钥 B_J^{sk}，可以对密文进行解密，解密公式如(2.14)所示：

$$\vec{m} = \vec{c} - B_J^{sk} \bullet \left[\left(B_J^{sk} \right)^{-1} \bullet \vec{c} \right] \bmod B_I \tag{2.14}$$

其中 $[\bullet]$ 表示距离参数最近的整数

④ *Eval over Addition*：如公式(2.15)所示，对于明文 $\vec{m_1}, \vec{m_2} \in \{0,1\}^n$，

$$\begin{aligned}\vec{c_1}+\vec{c_2}&=E\left(\vec{m_1}\right)+E\left(\vec{m_2}\right)\\&=\vec{m_1}+\vec{m_2}+\left(\vec{r_1}+\vec{r_2}\right)\bullet B_I+\left(\vec{g_1}+\vec{g_2}\right)\bullet B_J^{pk}\end{aligned}\tag{2.15}$$

$\vec{c_1}+\vec{c_2}$ 在密文空间并保持形式不变，对密文和进行解密时，如公式(2.16)所示：

$$\begin{aligned}\vec{c_1}+\vec{c_2}&=\left(\vec{c_1}+\vec{c_2}\right)\bmod B_J^{pk}\\&=\left(\vec{m_1}+\vec{m_2}+\left(\vec{r_1}+\vec{r_2}\right)\bullet B_I+\left(\vec{g_1}+\vec{g_2}\right)\bullet B_J^{pk}\right)\bmod B_J^{pk}\\&=\vec{m_1}+\vec{m_2}+\left(\vec{r_1}+\vec{r_2}\right)\bullet B_I\\&=\left(\vec{m_1}+\vec{m_2}+\left(\vec{r_1}+\vec{r_2}\right)\bullet B_I\right)\bmod B_I\\&=\vec{m_1}+\vec{m_2}\end{aligned}\tag{2.16}$$

⑤ *Eval over Multiplication*：如公式(2.17)所示，设 $\vec{e}=\vec{m}+\vec{r}\bullet B_I$，则

$$\begin{aligned}\vec{c_1}\times\vec{c_2}&=E\left(\vec{m_1}\right)\times E\left(\vec{m_2}\right)\\&=\vec{e_1}\times\vec{e_2}+\left(\vec{e_1}\times\vec{g_2}+\vec{e_2}\times\vec{g_1}+\vec{g_1}\times\vec{g_2}\right)\bullet B_J^{pk}\end{aligned}\tag{2.17}$$

其中 $\vec{e_1}\times\vec{e_2}=\vec{m_1}\times\vec{m_2}+\left(\vec{m_1}\times\vec{r_2}+\vec{m_2}\times\vec{r_1}+\vec{r_1}\times\vec{r_2}\right)\bullet B_I$。

对于同态乘法来说，只要 $\left|\vec{e_1}\times\vec{e_2}\right|$ 足够小，那么密文 $\vec{c_1}\times\vec{c_2}$ 就可以正确恢复出明文 $\vec{m_1}\times\vec{m_2}$。只要将 $\left|\vec{e_1}\times\vec{e_2}\right|$ 控制在可解密范围内，通过压缩和自举技术，可以将 SWHE 方案升级为 FHE 方案，证明过程如下：

①压缩（Squashing）：Gentry 提出的自举技术仅适用于一些电路深度较小的解密方案，所以需要调整电路深度使其适用。首先，选择一组向量的集合，向量和为密钥的乘法逆元 $\left(B_J^{sk}\right)^{-1}$，则密文乘以此集合中的元素，电路深度将减小到可处理的。与此同时，密钥恢复的难度将基于稀疏子集和的问题。

②自举（bootstrapping）：自举实际上是一种加密过程，用于从明文相同的嘈杂密文中获取新的密文。首先生成两对不同的密钥对 $(pk1,sk1)$ 和 $(pk2,sk2)$，私钥由客户端保留，公钥与服务器共享，接着对私钥加密并将其传给服务器 $Enc_{pk1}(sk1)$，该服务器上已经有 $c=Enc_{pk1}(m)$。由于上述方案为 SWHE，所以可以使用加密后的私钥 $Enc_{pk1}(sk1)$ 进行同态解密，解密后的结果再使用另一对密钥进行加密，即 $Enc_{pk2}\left(Dec_{sk1}(c)\right)=Enc_{pk2}(m)$。此时密文是安全的并且可以由私钥进行解密，即 $Dec_{sk2}\left(Enc_{pk2}(m)\right)=m$。简而言之就是，首先对有噪声的密文进行同态解密，以消除噪声；然后，“新”的同态加密将新的小噪声引入密文中。此时可以对这个“新”的密文进行新一轮的同态运算，直到噪声再一次达到阈值，

循环往复。

2.4.2 基于整数的 FHE 方案

在 Gentry 方案推出后的一年，Dijk 等人提出了另一种 SWHE 方案[9]，该方案基于近似最大公约数（AGCD）问题，AGCD 问题试图从 $x_i = pq_i + r_i$ 中恢复出 p。该方案的经典版本可能是最简单的方案之一，算法思想如下：

① *KeyGen*：对于给定的安全参数 λ，随机生成位长为 η 的奇数 p。

② *Enc*：对于随机的大素数 p 和 q，选择一个较小的数 $r \ll p$，则消息 $m \in \{0,1\}$ 可由公式(2.18)进行加密：

$$c = E(m) = m + 2r + pq \tag{2.18}$$

其中 p 被隐藏作为私钥。

③ *Dec*：密文可以用公式(2.19)进行解密：

$$m = D(c) = (c \bmod p) \bmod 2 \tag{2.19}$$

当且仅当 $m + 2r < p/2$ 时，该方案才有效，这限制了同态操作的深度。Dijk 使用了 Gentry 的压缩和自举技术使得方案变为 FHE 方案。

④ *Eval over Addition*：如公式(2.20)所示：

$$\begin{aligned} E(m_1) + E(m_2) &= m_1 + 2r_1 + pq_1 + m_2 + 2r_2 + pq_2 \\ &= (m_1 + m_2) + 2(r_1 + r_2) + p(q_1 + q_2) \end{aligned} \tag{2.20}$$

当噪声 $|m_1 + 2r_1 + m_2 + 2r_2| < p/2$ 时，该密文可以正确解密，由于 $r_1, r_2 \ll p$，当噪声扩大到 $p/2$ 之前，仍可对其进行同态运算。

⑤ *Eval over Multiplication*：如公式(2.21)所示：

$$\begin{aligned} E(m_1)E(m_2) &= (m_1 + 2r_1 + pq_1)(m_2 + 2r_2 + pq_2) \\ &= m_1 m_2 + 2(m_1 r_2 + m_2 r_1 + 2r_1 r_2) + kp \end{aligned} \tag{2.21}$$

输出保留了原始密文的格式并保持了同态属性。如果噪声的大小不足私钥的一半，即 $|m_1 m_2 + 2(m_1 r_2 + m_2 r_1 + 2r_1 r_2)| < p/2$，则仍可以正确解密。噪声随着乘法呈指数级增长，这无疑对同态运算增加了很多限制。

2.4.3 基于 LWE 的 FHE 方案

错误学习（LWE）被认为是即使后量子算法在实际时间内也要解决的最困难

的问题之一。Ring-LWE（RLWE）问题是 LWE 的代数形式，可以归结为理想格上的最坏情况的问题，这对于多项式时间量子算法很难。尽管 LWE 和 RLWE 都可以作为 FHE 方案的假设，但是 RLWE 有更好的性能。2011 年 Brakersk 等提出了一个基于 RLWE 的 SWHE 方案[10]，该方案仍然利用 Gentry 提出的压缩和自举技术来实现 FHE。同样的，接下来介绍它的经典版本。

首先，对该方案中的一些常见符号表示进行解释。$\langle a,b\rangle$ 表示向量 a 和 b 的内积。$d \xleftarrow{\$} D$ 表示 d 从分布 D 中随机分配。$Z[x]/(f(x))$ 表示所有以 $f(x)$ 为模的多项式的环。$R_q = Z_q[x]/(f(x))$ 表示系数为 Z_q 的模 $f(x)$ 的多项式环。χ 表示环 R_q 上的误差分布。

① *KeyGen*：从误差分布中选择一个环上的元素作为密钥，记作 $s \xleftarrow{\$} \chi$，则密钥向量为 $\vec{s} = (1, s, s^2, ..., s^D)$，$D$ 为整数。

② *Enc*：选择一个随机向量 $a \xleftarrow{\$} R_q^{\,n}$，噪声为 $e \xleftarrow{\$} \chi$，则消息 m 可以通过式(2.22)加密，其中 $\vec{c} \in R_q^{\,2}$：

$$\vec{c} = (c_0, c_1) = (as + te + m, -a) \tag{2.22}$$

③ *Dec*：当 $\langle \vec{c}, \vec{s} \rangle$ 小于 $q/2$ 时，解密有效。解密如公式(2.23)所示：

$$m = \langle \vec{c}, \vec{s} \rangle (\mathrm{mod}\, t) \tag{2.23}$$

如果要使方案不对称，可生成随机对 $(a, as + te)$。

④ *Eval over Addition*：如公式(2.24)所示：

$$\begin{aligned} E(m) + E(m') &= (c_0 + c_0', c_1 + c_1') \\ &= ((a + a')s + t(e + e') + (m + m'), -(a + a')) \end{aligned} \tag{2.24}$$

与前面的方案相似，当噪声小时，同态运算可以正常进行。

⑤ *Eval over Multiplication*：如公式(2.25)所示：

$$\begin{aligned} E(m)E(m') &= (c_0 c_0', c_1 c_1') \\ &= (-a's^2 + (c_0'a + c_0 a')s + t(2ee' + em' + e'm) + mm') \end{aligned} \tag{2.25}$$

全同态加密方案的研究主要便集中在以上三个方向，如表 2.2 所示。三类方法基本思想和研究思路可以总结如下：

（1）基于 Gentry 提出的理想格上的全同态加密方案的改进和优化，研究重点在如何降低解密电路的深度。

（2）基于整数上的全同态加密方案的研究和扩展，主要是对 DGHV 算法的改进，改进方向集中在公钥和模数上。

（3）基于错误学习 LWE 和 R-LWE 问题的全同态加密算法的研究和改进，改进方向集中在密钥交换和模交换。

表 2.2 从安全性、有效性、实现方式将三种方案进行对比。

表 2.2 三种体制方案的比较

	Gentry 体制	**DGHV 体制**	**BGV 体制**
基于的数学难题	理想格	近似最大公约数	环上的错误学习问题
同态操作次数	$log_2 k$	$log_2 k$	k
算法运行时间	$O(n^{3.5})$	$O(n^{3.5})$	$O(n^{3.5})$
实现方式	电路压缩和自举技术	模数转换和自举技术	模数转换和自举技术

2.5 本章小结

在本章中，从 *KeyGen* 、 *Enc* 、 *Dec* 和 *Eval* 四个基本操作介绍了 HE 的理论基础。然后，介绍了一些比较经典具有代表性的 PHE 和 FHE 方案，对每种方案都从四种操作以及算法同态原理进行了简单的说明，在本章的所有方案中，PHE 方案仅支持加法同态或乘法同态，是 FHE 的基础；SWHE 仅支持有限数量的运算或某些特定电路，例如分支程序；FHE 支持任意操作，例如搜索、排序、最大、最小等。在 2009 年 Gentry 的工作后，人们对于 HE 领域的兴趣明显增加，这极大地促进了同态加密的发展。

3. 全同态加密算法设计与实现

3.1 BGV 同态加密方案

BGV 方案的核心思想就是在每轮同态加密运算噪声超过阈值前对其进行密钥切换与模数切换，将维数和模数较大的密文转换为维数和模数较小的密文，将噪声降低，以便进行下一轮的同态运算。

3.1.1 层次 FHE 方案

2009 年 Gentry 提出了第一个真正意义上的全同态加密方案，其允许对加密数据进行任意次的加法和乘法运算。在此方案中，对数据进行加密时会随机引入一个噪声，以确保数据的安全性。当进行加法时，噪声线性增长，实际操作中几乎可以忽略不计；当进行乘法操作时，噪声呈指数型增长，当超过一定程度时，数据失真，同态加密失败。因此为了降低噪声，引入了自举（Bootstrapping）技术，该方法可以重置噪声，但同时会导致严重的性能损失，影响 FHE 的实用性。

基于错误学习的同态加密采用了不同的降噪管理方法，例如线性变换、模数转换和密钥转换，以减轻自举带来的性能损失。基于 BGV 方案的一个变种就是层次 FHE 方案，其中预设一个参数 L 为电路深度（乘法深度）以标识一次可进行级联乘法的次数。在层次 FHE 方案下，数据在同态加密后的电路深度为 L，进行加法操作时 L 不变，每进行一次乘法操作，L 减 1。当 L>1 时，可进行密文操作；当 L=1 时，不可进行乘法操作，此后的任何乘法操作将导致数据失真，即解密失败。因此，为了保证 FHE 的顺利进行，L 的设置尤为重要。

3.1.2 明文空间

为了利用同态加密对加密的数据进行计算，首先需要根据 BGV 的特征将数据编码为明文空间上的信息。BGV 使用有限域上的多项式环 $GF\left(p^d\right)$ 来表示明文空间，因此同态加密的加法和乘法不同于整数域上的加法和乘法，为了将两者区别，在本文中，使用 $+_h$ 来表示同态加密，$\times_h$ 来表示同态乘法。为了简化对数据的在明文空间上的编码处理，将使用多项式环作为明文空间，其中 $+_h \Leftrightarrow XOR$，$\times_h \Leftrightarrow AND$，不难证明，XOR 和 AND 是明文空间上的全功能集，任何功能的函数均可通过这两种操作实现，因此，需要将函数转换为二进制电路，以便使用 FHE 进行计算。

3.1.3 密文打包

BGV 方案将明文划 $GF(2)$ 分为多个插槽，以便同时对多组明文进行加密计算。这些插槽被称为明文插槽，并可以并行执行同态运算，类似于我们所熟知的

单指令多数据（SIMD）方式。图 3.1 展示了将明文消息利用明文插槽打包成 N 个 4bit 消息进行加密。本文中，使用 $PTXT$ 来表示明文空间，$CTXT$ 来表示密文空间。如果 $ptxt = x_7x_6x_5\cdots x_0$，$x_7x_6x_5\cdots x_0$ 为各个明文插槽，则 $ctxt = c_7c_6c_5\cdots c_0$。

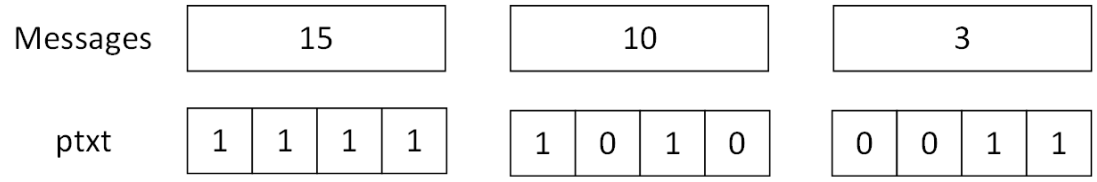

图 3.1 明文消息按 4bit 打包实例图

3.1.4 同态加密的基本操作

密文打包允许同态计算像 SIMD 方式一样并行执行。例如，密文 A 和 B 分为 n 个明文插槽，则 $A+_h B$ 分 n 个插槽并行进行同态加法。当然，同态操作的结果受明文空间 $GF\left(p^d\right)$ 选择的影响。这里为了方便解释同态加密的基本操作，选定明文空间为 $GF\left(2\right)$，这些操作在本文的中医药大数据中被广泛使用。

①同态加法 $+_h$：对相应的明文按明文插槽进行 XOR 操作。$+_h$ 并不会影响方案中 L 的深度。

②同态乘法 $\times_h$：对相应的明文按明文插槽进行 AND 操作。$\times_h$ 会使得电路深度 L 减 1，限制了同态运算。

③移位 $<<<_h$，$>>>_h$：以桶型移位器的方式进行插槽移位，移位可能会导致数据乱码，可通过选择操作进行纠正。

④选择掩码 $selmask$：类似于多路复用器电路，并在两个密文的插槽之间进行选择。未加密的二进制向量用作选择掩码。例如，如果第 i 个插槽的选择掩码为 1，则将从第一个密文中选择第 i 个插槽，否则从第二个密文中选择。在本文的实现过程中，将使用选择操作来屏蔽移位操作后的位。

3.1.5 同态加密方案性能分析

使用 BGV 方案执行 FHE 的性能将取决于参数 L。L 同时影响密文大小和同态运算的执行时间。将 L 选择得太高会增加密文大小和执行时间。但是，如果 L 太低，则无法计算所需的操作。与此同时，同态运算之间的也存在着性能差异。对于特定 L，加法几乎是免费的，而乘法和旋转运算的计算量很大。因此，在本文的实施过程中，将集中精力进行优化，以减少应用程序所需的 L，以及乘法和循环次数。

3.2 同态加密方案步骤

在本节中，以文献[13]为基础，通过一个简单的例子对同态加密方案进行一个较为详细的说明，其路线图如图 3.2 所示。消息数据先转换为明文消息，再转换为密文，函数经由电路转换为 SIMD，最后使用 FHE 进行同态操作。

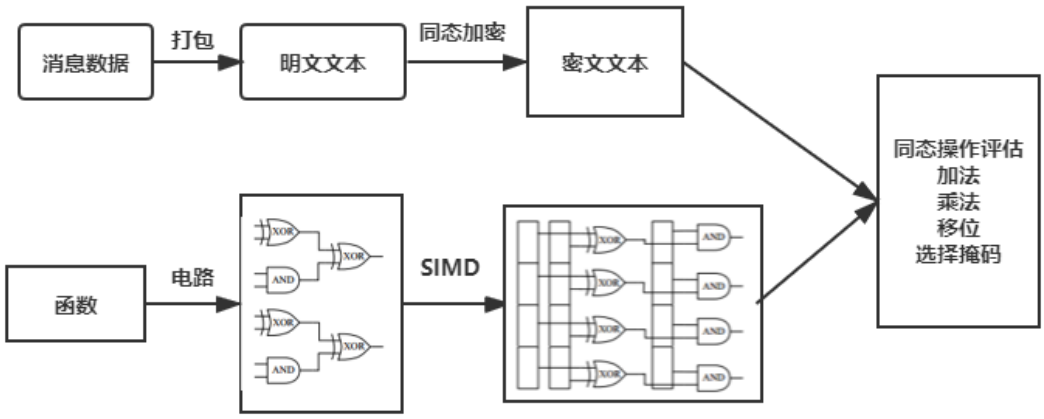

图 3.2 同态加密计算路线图

3.2.1 从函数到电路的转换

函数转换的第一步是将函数 $f(\bullet)$ 转换为二进制电路。不失一般性，可以通过 4-bit 的大于比较器进行说明 $(X > Y)$：

$$X > Y \Leftrightarrow \left(x_3\overline{y_3} \oplus x_2\overline{y_2}e_3 \oplus x_1\overline{y_1}e_3e_2 \oplus x_0\overline{y_0}e_3e_2e_1\right) \tag{3.1}$$

其中 x_i 是 X 的第 i 位的值，$\overline{y_i}$ 是 Y 的第 i 位的值的取反，e_i 是 X 和 Y 的按位相等，即 $x_i == y_i$。

3.2.2 电路到 SIMD 的转换

此阶段对于以 SIMD 方式进行同态运算是必须的。假设 X 和 Y 是明文空间 $GF(2)$ 上的 k-bit 明文，则按照明文插槽的定义 X 和 Y 可表示为 $X = x_k x_{k-1} x_{k-2} \cdots x_0$，$Y = y_k y_{k-1} y_{k-2} \cdots y_0$。则同态加法运算 $X +_h Y$ 对具有相同插槽索引的每个明文比特进行加法运算（ XOR 运算），这里有一个缺点，即不能对不同插槽索引的数据做任何操作，如上述公式，$x_3\overline{y_3}$、$x_2\overline{y_2}$、$x_1\overline{y_1}$、$x_0\overline{y_0}$ 等运算相当于执行 $\times_h$ 运算，即

$$X \times_h Y \Leftrightarrow \left(x_3\ x_2\ x_1\ x_0\right) \wedge \left(\overline{y_3}\ \overline{y_2}\ \overline{y_1}\ \overline{y_0}\right) \tag{3.2}$$

然而由于 e_3 与 $x_2\overline{y_2}$，e_3e_2 与 $x_1\overline{y_1}$，$e_3e_2e_1$ 与 $x_0\overline{y_0}$ 不在同一个插槽索引中，所以无法进行计算，后面将通过适当的向右移位和选择掩码操作解决这一问题。

3.2.3 SIMD 到 FHE 原语的转换

继续进行电路评估，可以将比较分解为两个$\times_h$操作，如公式(3.3)所示：

$$\begin{aligned} X >_h Y &\Leftrightarrow \left(X \times_h \overline{Y}\right) \times_h E' \\ &\Leftrightarrow \left(\left(x_3\ x_2\ x_1\ x_0\right) \wedge \left(\overline{y_3}\ \overline{y_2}\ \overline{y_1}\ \overline{y_0}\right)\right) \wedge \left(1 e_3\ e_3 e_2\ e_3 e_2 e_1\right) \end{aligned} \tag{3.3}$$

其中$E = \left(e_3\ e_2\ e_1\ e_0\right)$，$e_i = XNOR\left(x_i, y_i\right) = \overline{x_i \oplus y_i} = x_i \oplus y_i \oplus 1$。

通过E计算E'需要进行移位和选择掩码操作，将E进行移位并通过适当的选择掩码将其中不需要的位替换为“1”，如公式(3.4)所示：

$$E' = \left(1 e_3\ e_2\ e_1\right) \wedge \left(11 e_3\ e_2\right) \wedge \left(111 e_3\right) \tag{3.4}$$

由上述公式不难发现进行一次$>_h$同态操作，电路深度将会由 L 变为 L-4，即$>_h$的代价为 4。通常，对于 *k-bit* 的消息而言，进行一次$>_h$操作的代价为$\log_2 k + 2$，其中$\log_2 k$为计算E'的代价，2 为计算$\left(X \times_h \overline{Y}\right) \times_h E'$的代价。

3.3 基于同态加密的中医药数据安全存储算法设计

3.3.1 问题描述

传统的中医药数据存储通常是用传统的加密方法进行加密后然后上传，无法同时保证数据的安全性和可操作性。想要确保数据的安全性，就必须先将加密数据下载后进行解密操作，不能够充分利用云平台的优势来挖掘数据的价值；想要在云平台直接对数据进行操作就必须先对数据进行解密，这样就无法确保数据的安全性。其数据处理过程如图 3.3 所示。

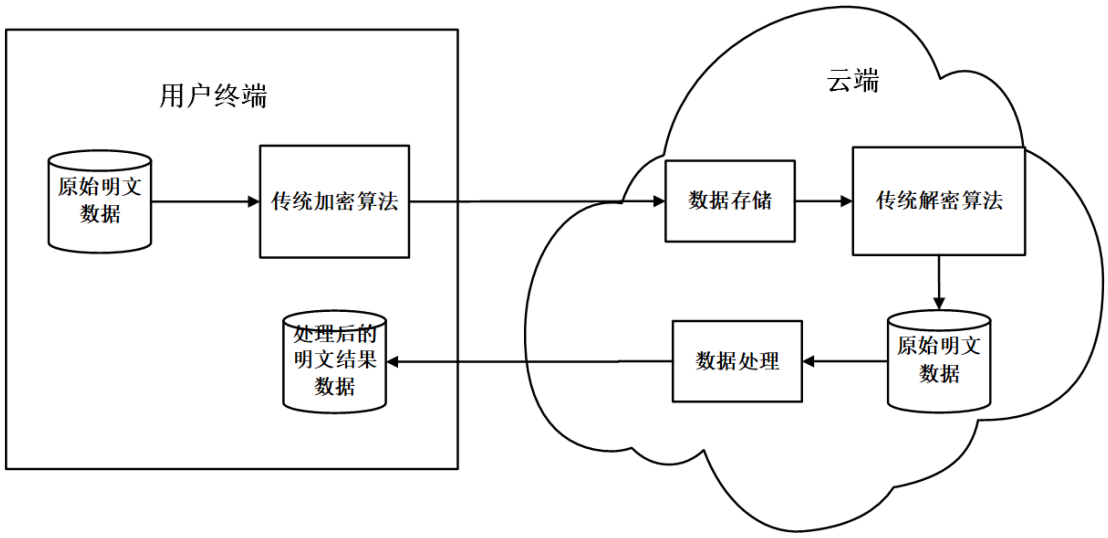

图 3.3 传统加密算法数据处理过程图

本文提供的算法主要应用于以下两个场景：

（1）在中医药数据安全存储中，$(M_1,M_2,\cdots,M_i,\cdots M_n)^T$ 表示未经算法处理的结构化的中医药数据中的一列属性，$(E(M_1),E(M_2),\cdots,E(M_i),\cdots E(M_n))^T$ 表示经过算法处理后的中医药数据。对于一个关键词 K，对其进行加密，得到 $E(K)$，则可以通过对 $E(K)$ 和 $E(M_i)$ 的比较得出 K 和 M_i 的比较结果，并将其返回。

（2）若数据文件为无结构文本文件，M 表示未经算法处理的流式文件，对文件进行明文分组，得到文件 $M_1M_2\cdots M_i\cdots M_n$，分别对每个分组进行加密，得到 $E(M_1)E(M_2)\cdots E(M_i)\cdots E(M_n)$，对于一个关键词 K，对其进行加密，得到 $E(K)$，则可以通过对 $E(K)$ 和 $E(M_i)$ 的比较得出 K 和 M_i 的比较结果，并将其返回。

3.3.2 加密算法

加密算法实现对明文数据的加密得到密文，流程如图 3.4 所示。

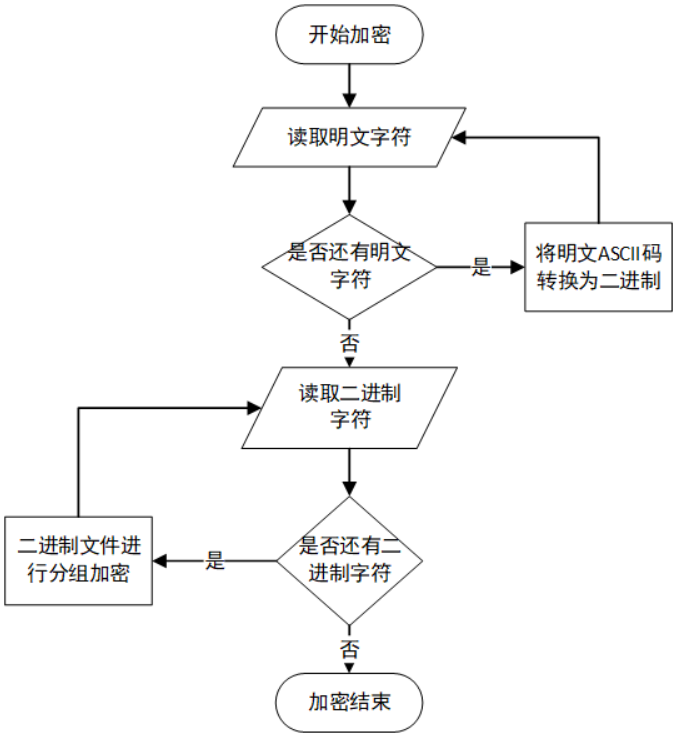

图 3.4 加密算法流程图

首先对明文 M 按比特进行分组（分组长度根据安全参数而定），其中中文明文为 16bit，英文明文为 8bit。本文中，分组长度可选 4bit、8bit、16bit、32bit 等。与 1bit 不同的是，多位算法文件末尾可能会出现不满足位数的情况，因此需要对其进行补充。分组分配采用类似边界对齐方式进行分配，然后对分组后的每个 m_i 进行加密，最后将加密后得到的密文分组依次合并，得到加密密文 C。算法具体

步骤如下：

（1）选取三个随机数：p 、q 和 r_1，其中 p 、q 为随机大素数，r_1 为随机数，同时计算 $n = pq$。

（2）将明文 M 按比特进行分组 $M = m_1 m_2 m_3 ... m_t$。

（3）对每一个明文分组 m_i 进行加密，加密算法如公式(3.5)所示：

$$c_i = \left(m_i + p \bullet r_1\right) \bmod n \tag{3.5}$$

（4）按照分组顺序，对加密后的密文进行合并操作，得到明文 M 的密文 $C = c_1 c_2 c_3 ... c_t$。

例如明文为五味子，将其按 8bit 分组，转换为二进制，结果为 01001110 10010100 01010100 01110011 01011011 01010000，相应的大数为 78 148 84 115 91 80，分别对每组用公式(3.5)进行加密，得到如图 3.5 所示结果，即为加密结果，转化为相应的大数为图 3.6 所示。

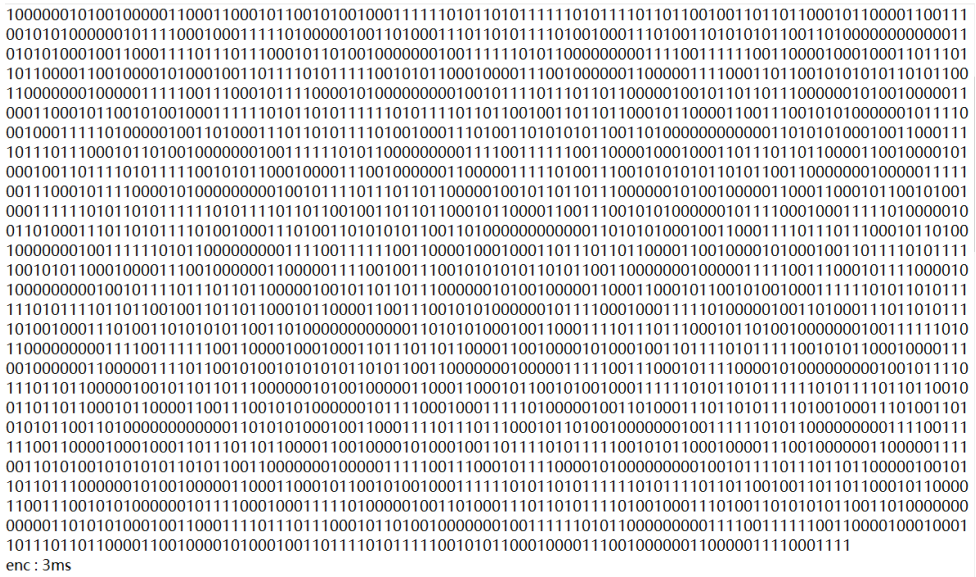

图 3.5 加密结果二进制示例图

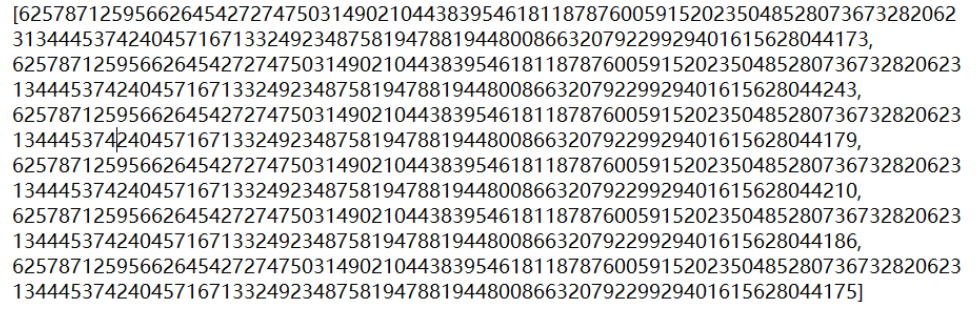

图 3.6 加密结果大数示例图

3.3.3 解密算法

解密算法用于将密文进行解密得到明文数据，流程如图 3.7 所示。

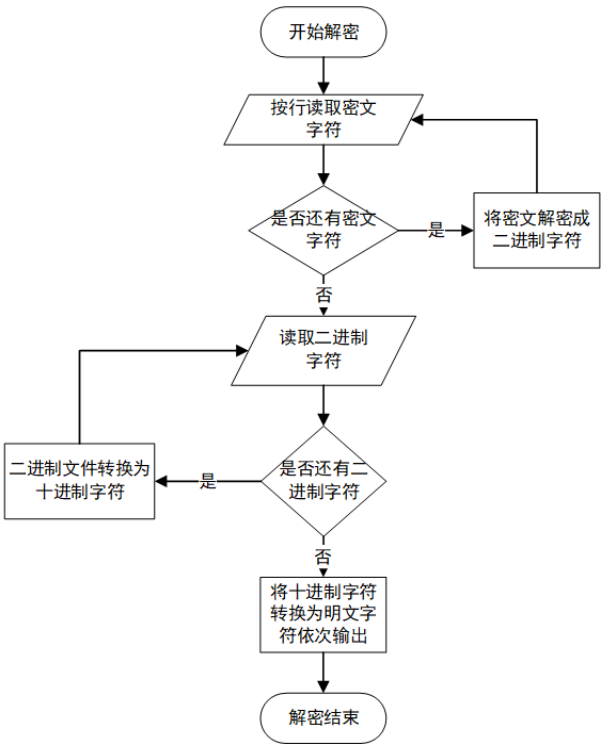

图 3.7 解密算法流程图

首先对密文 C 按比特进行分组，然后对分组后的每个 c_i 进行解密，最后将解密后得到的明文分组依次合并，得到明文 M 。算法具体步骤如下：

（1）将密文 C 按比特进行分组 $C = c_1 c_2 c_3 ... c_t$ 。

（2）使用公式(3.6)对每一个密文分组 c_i 进行解密：

$$m_i = c_i \bmod p \tag{3.6}$$

（3）对解密后的明文进行合并操作， $M = m_1 m_2 m_3 ... m_t$ 。

在上例中，对密文按分组进行如公式(3.6)所示运算，即可得到 78 148 84 115 91 80，将其转换为二进制后进行编码，得到明文“五味子”。

3.3.4 检索算法

检索算法用于在密文上直接进行数据的检索满足检索条件的数据，流程如图 3.8 所示。

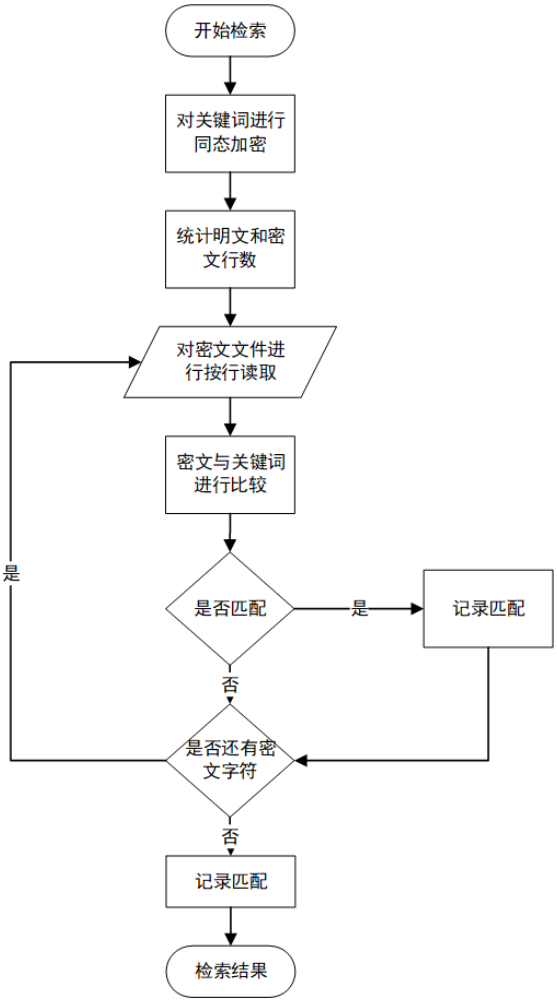

图 3.8 检索算法流程图

用户在本地对检索关键词进行加密后上传至服务器，接着进行密文检索匹配，若检索成功则返回。值得注意的是，如果匹配出现在文件末尾，需要判断文件末尾是否进行了补充，如果是，则不匹配；如果否，则匹配成功。算法具体步骤如下：

（1）将明文加密后上传到云端时，首先将 $q \times r_t$ 附加在明文数据的文件首部进行非对称加密，其中 q 为用户加密时所使用的大素数，r_t 为实时随机产生的随机大数。

（2）将用户检索关键词使用加密算法(3.7)加密后上传到服务器端，即

$$ckey = \left(key + p \bullet r_2\right) \bmod n \tag{3.7}$$

其中 key 为明文关键词，$ckey$ 为关键词明文。

（3）用私钥对明文消息前端的 $q \times r_t$ 部分进行解密。

（4）计算

$$\begin{aligned} res &= \left(\left(ckey - c_i\right) \times q \times r_t\right) \bmod n \\ &= \left(\left(\left(key + p \times r_2\right) - \left(m_i + p \times r_1\right)\right) \times q \times r_t\right) \bmod n \\ &= \left(\left(key - m_i\right) \times q \times r_t + \left(r_2 - r_1\right) \times p \times q \times r_t\right) \bmod n \\ &= \left(\left(key - m_i\right) \times q \times r_t\right) \bmod n \end{aligned} \tag{3.8}$$

（5）若 res 结果为 0，则检索成功，否则检索失败。

3.4 同态性分析

3.4.1 算法同态性分析

（1）加法同态性分析

对明文消息 m_1 和 m_2 分别进行加密得到密文 c_1 和 c_2，如公式(3.9)和(3.10)所示：

$$c_1 = E\left(m_1\right) = \left(m_1 + p \times r_1\right) \bmod n \tag{3.9}$$

$$c_2 = E\left(m_2\right) = \left(m_2 + p \times r_2\right) \bmod n \tag{3.10}$$

直接对密文进行加法操作，如公式(3.11)所示：

$$\begin{aligned} c_1 + c_2 &= E\left(m_1\right) + E\left(m_2\right) \\ &= \left(m_1 + m_2 + p \times \left(r_1 + r_2\right)\right) \bmod n \end{aligned} \tag{3.11}$$

对密文和进行解密，如公式(3.12)所示：

$$\begin{aligned} D\left(c_1 + c_2\right) &= \left(c_1 + c_2\right) \bmod p \\ &= \left(\left(m_1 + m_2 + p \times \left(r_1 + r_2\right)\right) \bmod n\right) \bmod p \\ &= m_1 + m_2 \end{aligned} \tag{3.12}$$

可见结果为明文和，加法同态性由此可证。

（2）乘法同态性分析

对明文消息 m_1 和 m_2 分别进行加密得到密文 c_1 和 c_2，，如公式(3.13)和(3.14)所示：

$$c_1 = E\left(m_1\right) = \left(m_1 + p \times r_1\right) \bmod n \tag{3.13}$$

$$c_2 = E\left(m_2\right) = \left(m_2 + p \times r_2\right) \bmod n \tag{3.14}$$

直接对密文进行乘法操作，如公式(3.15)所示：

$$\begin{aligned} c_1 \times c_2 &= E(m_1) \times E(m_2) \\ &= \left((m_1 + p \times r_1) \times (m_2 + p \times r_2)\right) \bmod n \\ &= \left(m_1 \times m_2 + m_1 \times p \times r_2 + m_2 \times p \times r_1 + p^2 \times r_1 \times r_2\right) \bmod n \\ &= \left(m_1 \times m_2 + p \times \left(m_1 \times r_2 + m_2 \times r_1 + p \times r_1 \times r_2\right)\right) \bmod n \end{aligned} \tag{3.15}$$

对密文乘积进行解密，如公式(3.16)所示：

$$\begin{aligned} D(c_1 \times c_2) &= (c_1 \times c_2) \bmod p \\ &= \left(\left(m_1 \times m_2 + p \times \left(m_1 \times r_2 + m_2 \times r_1 + p \times r_1 \times r_2\right)\right) \bmod n\right) \bmod p \\ &= m_1 \times m_2 \end{aligned} \tag{3.16}$$

可见结果乘积为明文乘积，乘法同态性由此可证。

3.4.2 检索算法分析

在公式(3.8)中，因为$n = pq$，所以如公式(3.17)所示，在检索算法中，随机数r_1和r_2在增加了密文安全性的同时，对密文检索的匹配结果没有影响：

$$\left((r_2 - r_1) \times p \times q \times r_t\right) \bmod n = 0 \tag{3.17}$$

若$key = m_i$，即关键词等于明文，则如公式(3.18)所示，此时检索成功；

$$\left((key - m_i) \times q \times r_t\right) \bmod n = 0 \tag{3.18}$$

否则如公式(3.19)所示，此时检索失败。

$$\left((key - m_i) \times q \times r_t\right) \bmod n > 0 \tag{3.19}$$

3.4.3 算法安全性分析

本文算法的安全性建立在大素数因式分解的困难问题。已知的证据都表明大整数因式分解问题是一个极其困难的问题。目前，没有已知算法可以在$O(n^k)$（k为常数）的时间内分解它，但是现有算法比$O(e^n)$快。换句话说，现在已知最好的算法比指数数量级时间要快，比多项式数量级要慢。已知最好的渐近线运行时间是普通数域筛选法（GNFS），其时间复杂度为$\Theta\left(\exp\left(\left(\frac{64}{9}n\right)^{\frac{1}{3}}(\log n)^{\frac{2}{3}}\right)\right)$。

本文算法的保密性取决于大素数因式分解的时间，假定用 10^6 次/秒的计算机进行运算，则分解 n=100 位的十进制数需要 74 年，分解 n=200 位的十进制数需要 $3.8*10^9$ 年。可见，当 n 足够大时（p 和 q 各为 100 位时，n 为 200 位），对其

进行分解是非常困难的。可以说，本文算法的保密强度等价于分解 n 的难易程度。而本文选取的 p 和 q 均接近 512 位，可充分保证算法的安全性和保密性。

3.5 算法性能分析

为了评价本文提出的同态加密算法的性能，将本文提出算法与 IBM 发布的 HELib 同态加密算法库进行比较，主要比较电路深度和明文槽大小对算法性能的影响。

HELib 库结构如图 3.9 所示[17]，包括数学结构层、格式转换层、数据加密层和数据移动层。数学结构层用于实现数学结构和各种其他实用程序的模块；格式转换层用于实现多项式的 Double-CRT 表示；数据加密层用于使用二进制多项式的明文空间来对实际 BGV 同态加密系统进行实现，包括密文、密钥、密钥交换矩阵等的实现，支持对本地明文空间的同态运算；数据移动层提供了一些接口，用于应用明文插槽使用密码系统对明文数组进行操作。

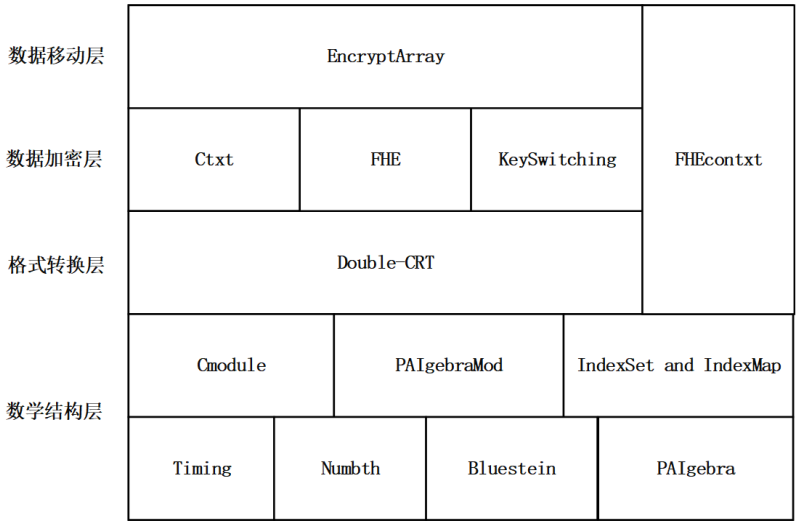

图 3.9 HELib 同态加密算法库结构图

在本文实验中，主要应用 HELib 库中的以下函数接口：

（1）$FHEcontext\ context(\lambda, m, p, r, w, L)$：初始化同态环境操作，并进行参数设置。

（2）$buildModChain(context, bits, c)$ ：构建密文刷新过程中的素数模数链。

（3）$FHE\,\mathrm{Sec}\,Key\,\mathrm{sec}\,ret\_key(context).Gen\,\mathrm{Sec}\,Key()$：根据同态环境生成密钥。

（4）$EncryptedArray.encrypt(ctxt, pk, ptxt)$：加密算法，其中，$ptxt$ 为明文，$ctxt$ 为密文，pk 为公钥。

（5）$EncryptedArray.decrypt(ctxt, sk, decrypted)$：解密算法，其中 $ctxt$ 为密文，sk 为私钥，$decrypted$ 为解密后的明文。

（6）$SwitchKey\left(\vec{c}, \vec{c_1}, s, s', q\right)$：对于相同的模数 q，将密钥为 s 的密文 $\vec{c}$ 转化为密钥为 s' 的密文 $\vec{c_1}$。通过密钥交换矩阵完成，主要用于降低密文的维数。

（7）$Ctxt.addCtxt\left(\vec{c_1}, \vec{c_2}, pk\right)$：同态加法，其中 $\vec{c_1}, \vec{c_2}$ 为密文，pk 为公钥，则 $\vec{c_3} = \left(\vec{c_1} + \vec{c_2}\right) \bmod q$。

（8）$Ctxt.multiplyBy\left(\vec{c_1}, \vec{c_2}, pk\right)$：同态乘法，$\vec{c_3} = \left(\vec{c_1} \otimes \vec{c_2}\right) \bmod q$。

3.5.1 电路深度 L 与检索时间的关系

对于同态操作来说，加法并不改变电路深度，其产生的噪声可以忽略不记，每进行一次乘法，会使电路深度减 1，是噪声的主要来源。而在进行密文检索时会不停地用到同态乘法运算，电路深度 L 的大小直接关系到密文检索的性能，因此需要对电路深度 L 与检索时间的关系进行研究。

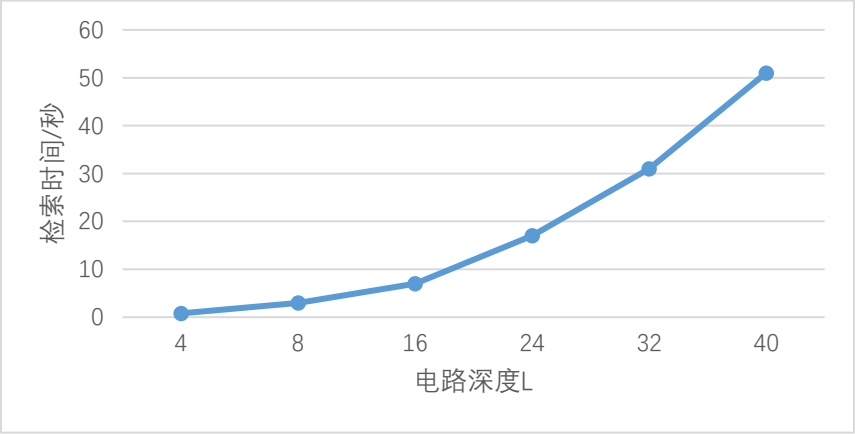

图 3.10　电路深度 L 与检索时间关系图

由图 3.10 可知，随着电路深度 L 的增加，检索时间也随之增加，且随着电路深度 L 的不断增加，检索时间的增长幅度也逐渐变大。

3.5.2 明文分组与加密时间的关系

在其他实验条件相同的情况下，研究在各种大小的明文下，不同大小的明文分组对加密时间的影响，如图 3.11 所示。

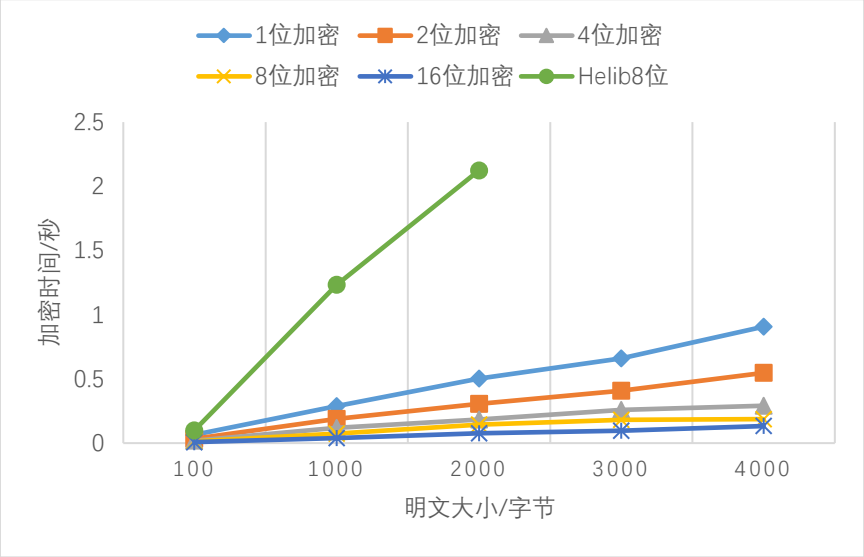

图 3.11 明文分组与加密时间关系图

实验结果表明，在其他实验条件相同的情况下，随着明文大小的不断增长，加密时间也不断增长。且对于相同的明文大小来说，随着分组大小的变大，加密时间减少。就总体而言，加密效率是非常可观的。相对于 Helib 的加密算法，在分组长度相同的情况下，本文的加密算法效率约是 Helib 的 10-40 倍，且明文数据字节越多，效率提升越明显。

3.5.3 明文分组与解密时间的关系

在其他实验条件相同的情况下，研究在各种大小的明文下，不同大小的明文分组对解密时间的影响，如图 3.12 所示。

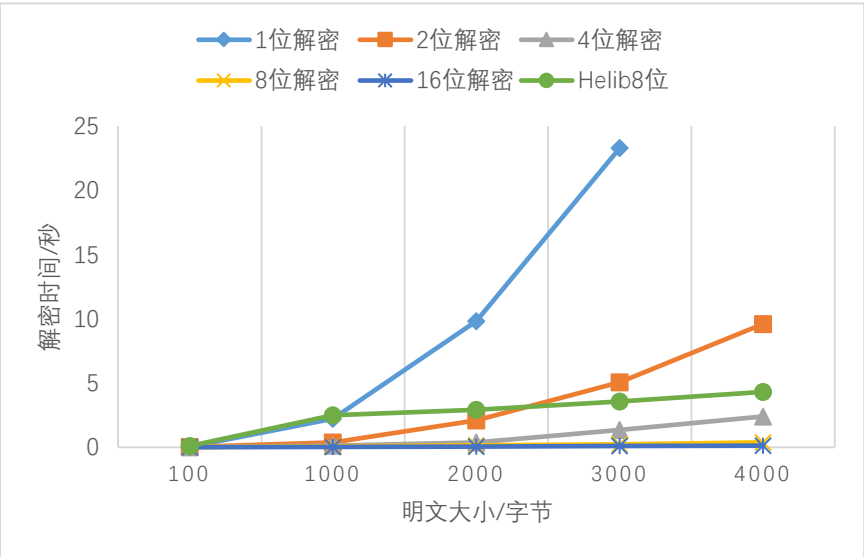

图 3.12 明文分组与解密时间关系图

实验结果表明，在其他实验条件相同的情况下，随着明文大小的不断增长，

解密时间也不断增长。且对于相同的明文大小来说，随着分组大小的变大，解密时间减少。就总体而言，解密算法的效率虽然没有加密算法高，但还在可接受范围内。相对于 Helib 的加密算法，在分组长度相同的情况下，本文的加密算法效率约是 Helib 的 10 倍左右。

3.5.4 明文分组与检索时间的关系

在其他实验条件相同的情况下，研究在各种大小的明文和各种大小的关键词下，不同大小的明文分组对检索时间的影响。

（1）当关键词为四字节时，明文分组与检索时间关系图如图 3.13 所示。

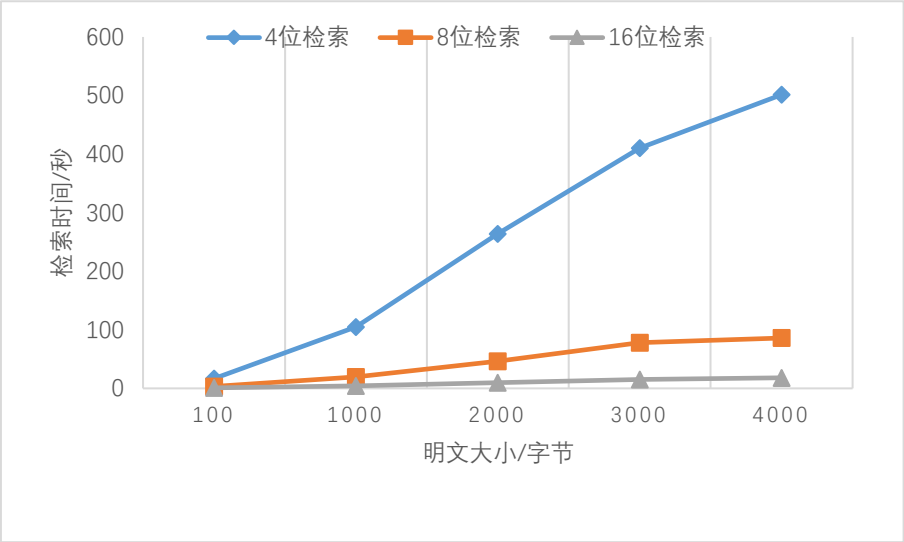

图 3.13 明文分组与检索时间关系图（key=4）

（2）当关键词为六字节时，明文分组与检索时间关系图如图 3.14 所示。

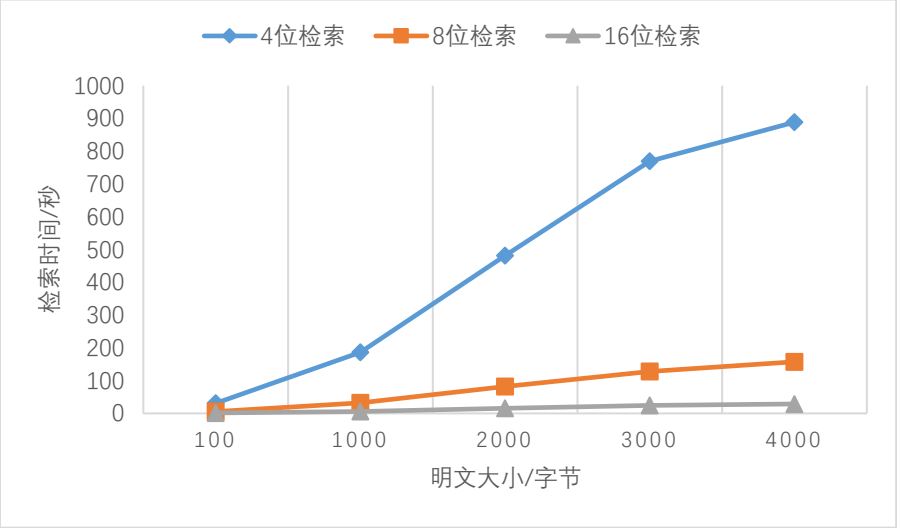

图 3.14 明文分组与检索时间关系图（key=6）

（3）当关键词为八字节时，明文分组与检索时间关系图如图 3.15 所示。

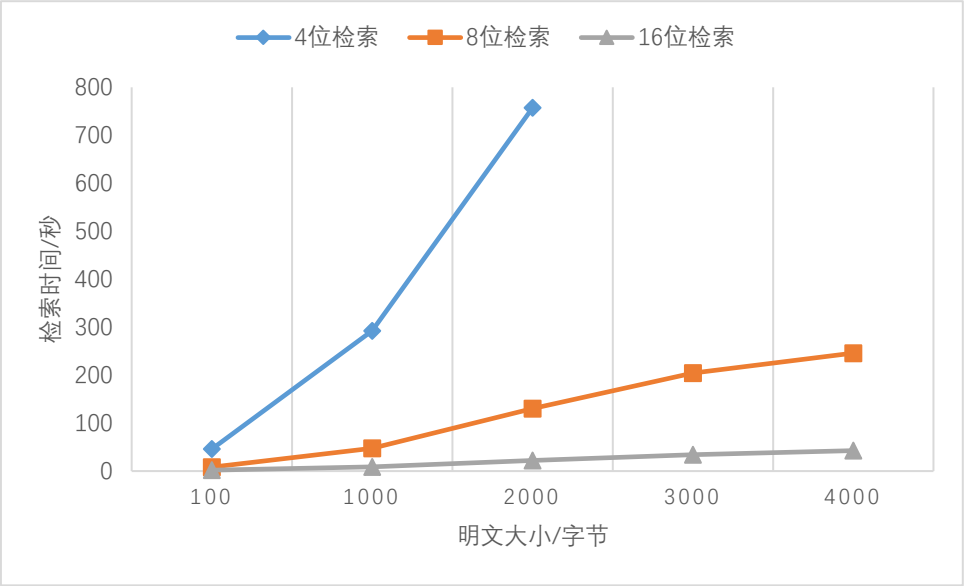

图 3.15 明文分组与检索时间关系图（key=8）

由图 3.13—图 3.15 可知，在其他实验条件相同的情况下，随着明文大小的不断增长，检索时间也不断增长。对于相同的明文大小来说，随着分组大小的变大，检索时间减少。且随着关键词的变长，检索时间也随之增长。但是可以发现，在明文分组较长的情况下，该检索算法的性能具有一定的实用性。

3.6 本章小结

本章首先对 BGV 同态加密方案的原理以及同态加密方案的步骤进行了一些介绍，然后提出了本文的加解密方案和检索算法，通过严密的数据逻辑证明，证明了加解密算法的全同态性和检索算法的可行性，并对算法的安全性和保密性进行了研究。通过实验对加解密算法和检索算法在不同电路深度 L、不同明文分组、不同明文大小以及不同长度的关键词检索的性能进行了分析，并和 HELib 算法进行了对比，实验证明，本文的加解密算法在兼顾数据安全的前提下提高了算法的效率。

4. 同态加密技术在中医药数据存储中的应用

4.1 应用背景

为了进一步检验本文提出的同态加密算法在中医药数据安全存储中的应用效果，本文在中医药数据安全系统中设计实现了该同态加密算法，用于对中医药数据文件的同态加密、同态解密和基于密文的同态检索，分别对应于系统的上传、下载和检索模块。在系统中，用户可以在客户端选择即将上传的文件并进行同态加密，之后通过网络传输到服务器端，服务器对加密后的数据文件进行接受存储，可以保证数据文件的安全。用户也可以选择已经存储在服务器端的文件进行下载，然后在本地进行同态解密，得到自己所需要的数据文件。用户还可以输入关键词直接在服务器端对加密的数据文件进行检索，以检索自己所需的数据进行下载。这样，即便是将数据存储在了如公有云等不可信第三方，依旧可以确保数据的安全。本系统利用了同态加密的性质，在保证数据安全性的同时充分利用云计算的强大功能。

全同态加密方案允许用户在无需对密文进行解密的情况下直接对密文进行操作，解密后依然可以得到自己想要的结果。在客户端/服务器模式下，用户将数据文件加密后上传到服务器端，服务器对密文文件进行存储，当用户有检索需求时，将检索关键词进行同态加密后上传到服务器端，服务器根据密文关键词对密文文件进行检索，用户得到服务器反馈后对密文文件进行下载到本地后进行解密，得到自己所需的明文文件，其流程如图 4.1 所示。

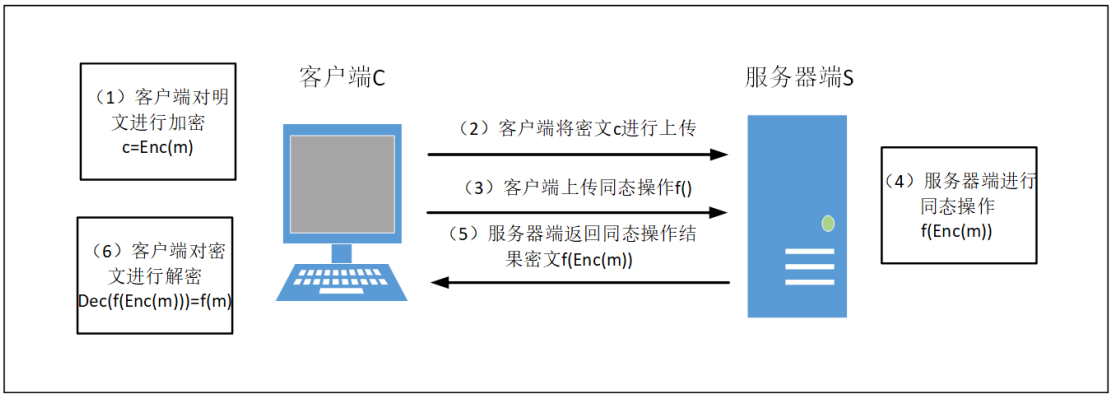

图 4.1 云环境中同态加密流程图

（1）客户端 C 首先对明文数据进行加密。

（2）客户端 C 将加密后的密文数据发送到云服务器 S 上。

（3）客户端 C 希望对其私有数据进行操作（即查询）。

（4）服务器端 S 使用同态方案对加密数据执行同态运算。

（5）服务器端 S 将加密结果返回给客户端 C。

（6）客户端 C 使用密钥恢复数据并得到自己想要的结果。

4.2 系统设计与实现

本文在中医药管理系统中实现了基于同态加密的中医药数据存储方法，分为上传、下载和检索模块。接下来分别对三个模块进行介绍。

4.2.1 上传模块

上传模块主要负责在客户端对数据文件进行加密并将其上传到服务器端。明文数据在客户端进行加密，加密密钥由客户端、服务器共同持有，解密密钥仅由客户端独有。此模块用例图如图 4.2 所示。

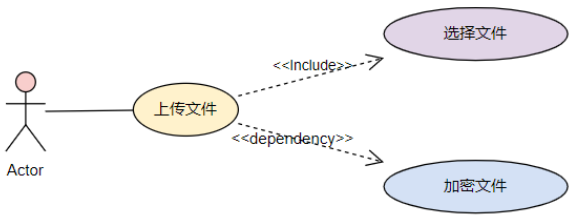

图 4.2 上传模块用例图

用户首先选择要进行加密上传的文件，然后对所选中文件进行加密，然后通过网络上传至服务器保存。活动图如图 4.3 所示。

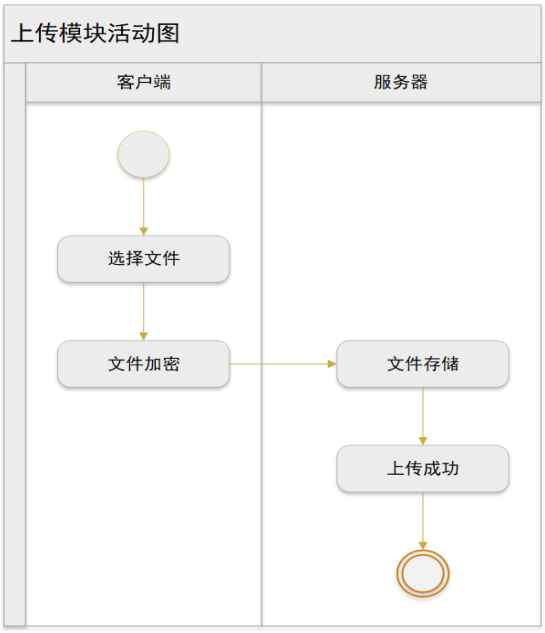

图 4.3 上传模块活动图

4.2.2 下载模块

下载模块主要负责根据用户需求在服务器中检索相应的文件，并将选中文件进行下载后，在本地对其进行解密，下载模块的用例图如图 4.4 所示。

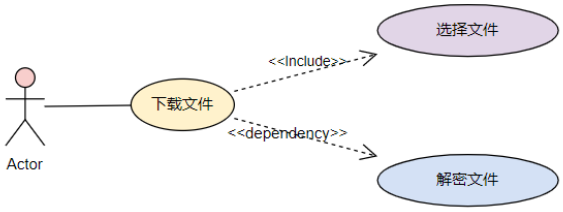

图 4.4 下载模块用例图

活动图如图 4.5 所示。

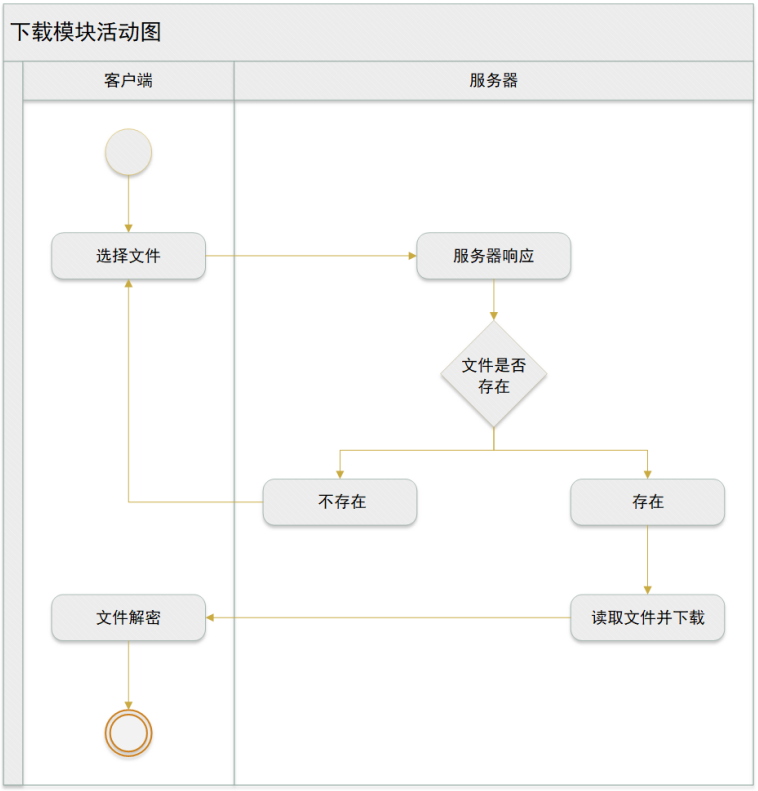

图 4.5 下载模块活动图

4.2.3 检索模块

检索模块主要负责数据文件的检索，首先将用户搜索的关键词在本地进行加密后上传到服务器端，在服务器端对关键词密文进行同态检索。若检索成功，则将检索到的密文文件下载到本地解密，若检索不成功则返回空。检索模块用例图

如图 4.6 所示。

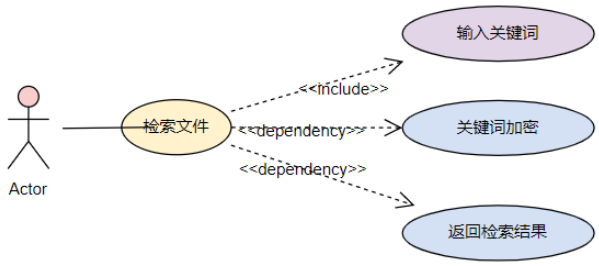

图 4.6 检索模块用例图

活动图如图 4.7 所示。

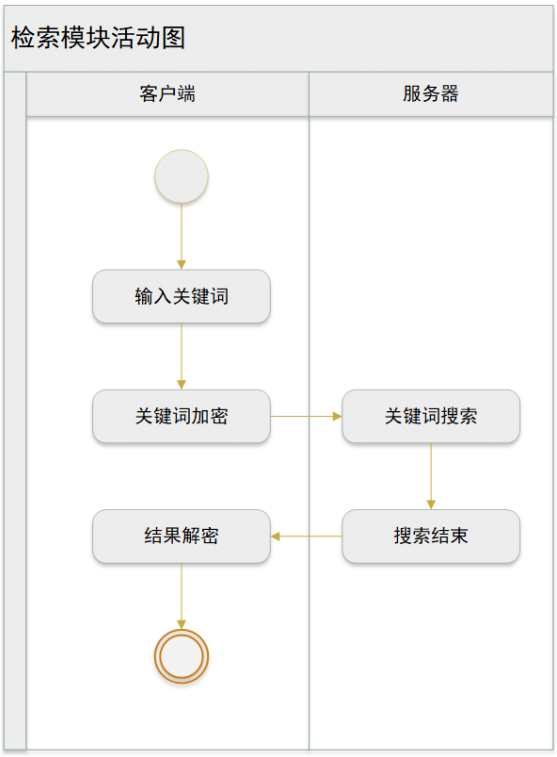

图 4.7 检索模块活动图

4.3 系统实验测试

4.3.1 系统运行环境

本文的中医药数据系统由客户端和服务器两部分组成，系统运行环境如表 4.1 所示。

表 4.1 系统运行环境表

运行环境	配置说明
系统开发环境	操作系统：OS X 10.13 编程语言：Java 集成开发环境：IDEA 和 WebStorm
服务器端配置	操作系统：OS X 10.13 CPU：Intel(R)Core(TM)i7-6700HQCPU@2.60GHZ 存储空间：16GB 内存、256GB 磁盘
客户端配置	操作系统：OS X 10.13 测试平台：Chrome 81.0.4044.138

4.3.2 实验结果分析

运行服务器程序，服务器的主要功能有：密文的存储，密文的传输，以及密文的操作。服务器将文件接受并存储在云端。

上传模块：用户选择本地中需要上传到云端的文件，选定之后系统自动进行全同态加密算法进行加密，加密完成后上传到服务端进行存储。用户上传系统模块如图 4.8 所示。

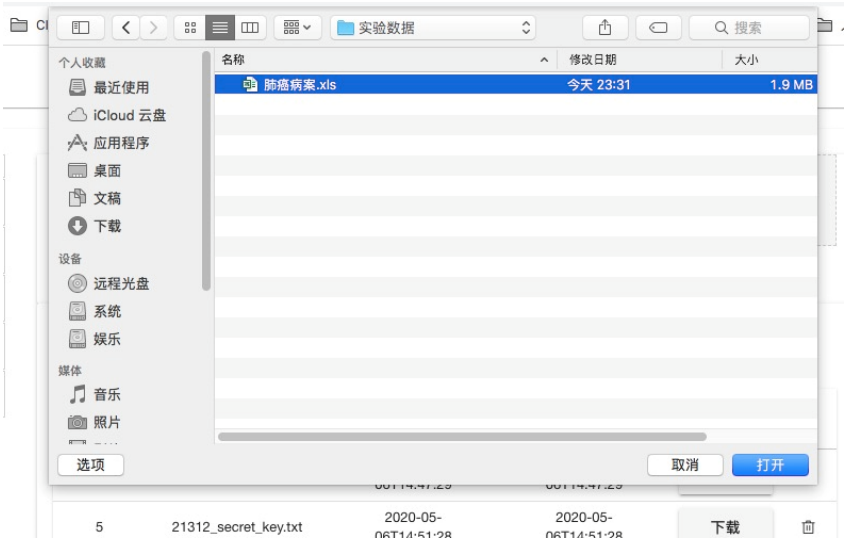

图 4.8 系统上传模块界面图

例如，从客户端选择文件“测试数据.txt”，文件内容如图 4.9 所示。

图 4.9 明文文件内容

对明文文件进行同态加密后上传到服务器端，服务器端存放的密文文件如图 4.10 所示。

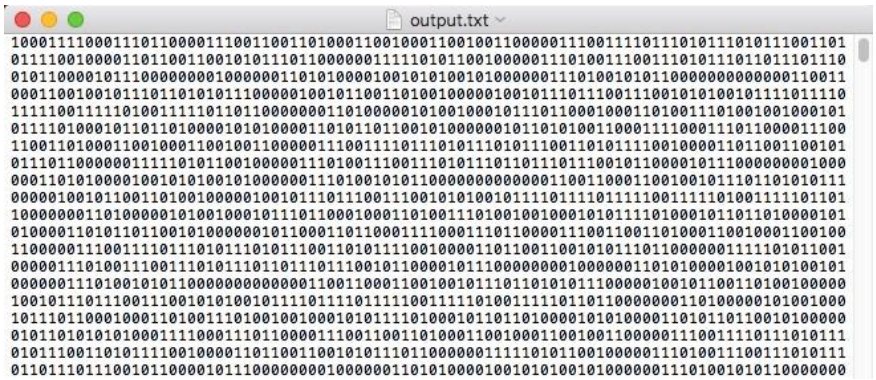

图 4.10 密文文件内容

下载模块：当用户需要下载存储在服务器端的文件时，可以直接在个人设置-我的数据中进行查看下载。服务器端将进行同态加密处理后的文件发送给客户端，然后客户端在本地进行同态加密后得到明文文件，用户即可访问文件。用户下载系统模块如图 4.11 所示。

图 4.11 系统下载模块界面

检索模块：用户在客户端的检索框中输入检索关键词后，系统将自动将关键词加密后上传到服务器端，服务器端根据检索算法将关键词和文件进行匹配，如果匹配成功，则返回文件名，如图 4.12 所示；若匹配失败，则返回空，如图 4.13 所示。

图 4.12 检索匹配成功图

图 4.13 检索匹配失败图

另外，因为中医药数据各个属性的隐私保护级别不同，对于用户的重要程度不同，以及每个属性的可操作性不同，在数据加密模块，还为用户提供了多种加密算法可以选择，用户可以根据自己的实际需求为自己数据的不同属性选择不同的加密算法，定制自己的个性化服务。用户首先选择需要进行加密上传到服务器的文件。接着如图 4.14 所示，系统对属性进行识别，用户可以对每个属性分别选择加密算法，然后用户对加密任务进行确认。最后，系统根据用户的要求对各个属性进行加密，并返回给用户结果，如图 4.15 所示。用户可以对结果进行导出到本地进行保存，如图 4.16 所示。

图 4.14 按需选择系统属性加密界面图

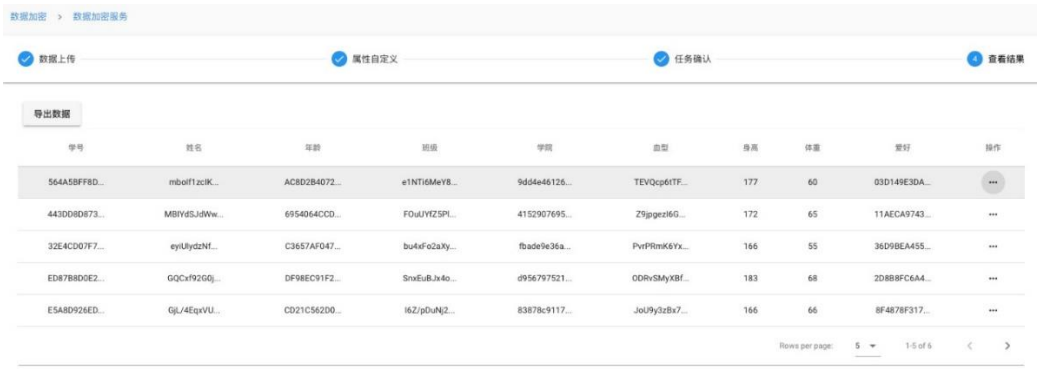

图 4.15 查看结果图

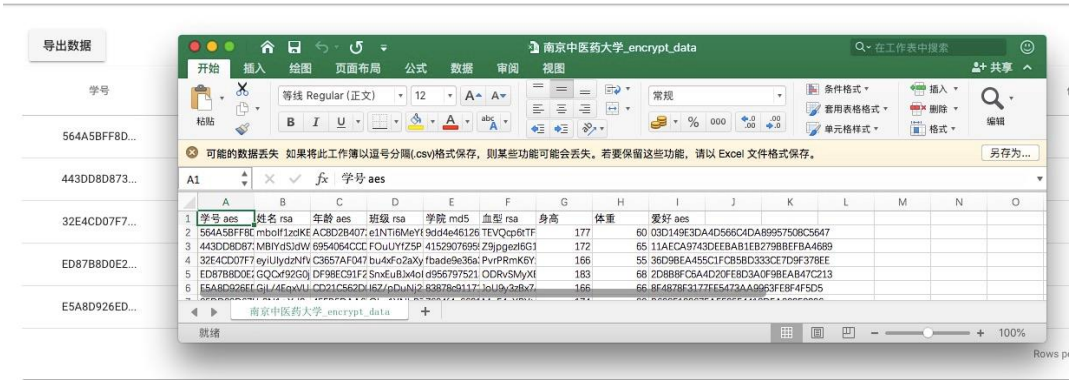

图 4.16 结果导出图

4.4 本章小结

本章主要介绍了本文提出的基于同态加密的算法在中医药数据系统中的详细设计与实现。首先对系统的应用背景进行了阐述，接着对本文所涉及的模块即上传、下载和检索模块的详细设计与处理流程进行了介绍，然后对系统模块进行了实验测试和结果分析。基于同态加密的中医药数据安全存储方法能够在保证数据安全的同时利用云计算服务器高性能的优点实现了数据的操作处理，解决了传统加密方法无法直接对密文进行检索的问题。

5. 总结与展望

5.1 论文工作总结

云计算云存储近年来被广泛应用于军事、工业、医疗等各个领域，各大医药机构也纷纷将医药数据放至存储。由于医药数据具有极高的经济价值和医疗价值，以及我国医疗安全研究起步较晚，医疗数据的安全性无法得到有效保障。同时，传统的加密手段虽然能一定程度上保证数据的安全，但是无法发挥云计算的优势，使用起来极其不方便，所以设计一个基于同态加密的中医药数据存储方案，将中医药和云计算充分结合起来，发挥云计算的优势来挖掘中医药数据中极大的医药价值和经济价值是中医药存储中的一个重要研究课题。同态加密相比于传统的加密算法具有两方面的优势：其一全同态加密方案可以直接在无需对密文进行解密的情况下直接对密文操作，保证了数据的方便性和可操作性，其二全同态加密过程中密钥随机生成且噪声极其复杂，能够防止数据泄露，保障了数据的安全性。本文在对全同态加密算法的研究基础上提出了比较高效的密文检索算法，在保证数据安全的情况，提高了密文检索效率。本文的工作如下：

（1）对同态加密的相关知识进行介绍。对同态加密的概念、发展历程以及几种经典的部分同态加密方案和全同态加密方案的原理、特点、算法进行了比较详细的介绍。从而提出了基于全同态加密的加密、解密和检索算法。

（2）在对同态加密方案研究的基础上提出了一种改进的基于同态加密的中医药数据存储方法，由加密、解密、检索三部分组成。对算法的有效性和同态性进行了严格的数学证明，对算法的安全性和保密性进行了研究，并通过实验对改进算法进行分析，通过实验结果论证了算法的正确性和实用性。

（3）设计并实现了基于全同态加密的中医药数据存储系统。本文主要设计实现了系统的上传、下载和检索模块。系统基于客户端/服务器模式，在保证数据安全性的时候提高了数据操作的便捷性。

5.2 研究展望

本文针对目前中医药数据存储中所面临的安全性和可操作性问题，在现有的同态加密方案的基础上，提出了新的加密、解密和检索算法，并设计实现了中医药数据系统。虽然该可以在保障数据安全性的同时，可以直接对密文进行检索，但是仍然存储一些问题需要研究。

（1）本文所提出的检索算法在效率方面表现并不如人意，在进行检索操作时，只能按位而无法按分组进行遍历，且由于为了使效率有所提升，导致公钥体

积太大，加密后得到的密文文件远远大于明文文件，所以还需要对同态加密方案进行进一步的研究，提出密文检索效率的同时，控制密文膨胀的规模。

（2）本文在实现数据同态加密解密的同时，仅仅提出了同态检索方案，并没有能够实现真正意义上的全同态，因此还需要对算法进行进一步的研究，丰富操作类型。

（3）虽然国内外学者对同态加密进行了漫长且丰富多彩的研究，但直到目前为止，还没有一个十分成熟的同态加密方案应用于实际需求，因此还需要对同态加密方案进行不断的研究和探索，使其尽早应用于实际需求，与云计算的强大优势相结合。

致谢

光阴似箭，转眼间大学生活即将过去，回首大学四年生涯，充满了收获和感动。在毕业论文即将完成之际，在大学即将毕业之际，我由衷的感谢那些曾经在学习上和生活上帮助过我的老师、同学和朋友们。

首先我要感谢我的导师丁有伟老师，感谢他大学四年来对我无微不至的照顾，无论是在学习上还是生活上都给予了我莫大的支持。感谢他在我低谷时给我的鼓励，感谢他在我成功时为我的欢喜，感谢他一直在背后默默地支持着我。并且在本次毕业设计过程中，从选择完课题，到完成课题所需要的论文资料，到课题的算法设计研究，到毕业论文的撰写，到论文的最终定稿，每一步都得到了导师细心的指导和帮助。在论文写作时，从文章结构的安排到内容的撰写，导师认真负责让我感动。很多细小的错误老师都一一指出加以改正。在论文一遍又一遍修改的过程中，我感受到了老师的爱，同时也学到了很多知识。在此向我的导师表示衷心的谢意。

在此，还要对大学四年来教导我的所有老师表示由衷的谢意，没有他们四年在知识上的教导，我也不会顺利地完成我的毕业设计和论文。在毕业设计过程中，我也遇到了很多问题，在同学们的帮助下，最终解决了这些难题，在此我也非常感谢他们。

最后感谢各位专家的批评指导！

参考文献

[1]https://www.all-about-security.de/fileadmin/micropages/Fachartikel_28/2019_Cost_of_a_Data_Breach_Report_final.pdf

[2]http://www.cac.gov.cn/2018-09/15/c_1123432498.htm

[3]王丽,王苹.中医药大数据基础平台安全性研究[J].电脑知识与技术,2017,13(35):17-18+20.

[4]生慧,周扬,马金刚,王振国. 一种基于联盟链的中医药海量异构数据安全共享解决方案[J]. 世界科学技术-中医药现代化,2019,21(08):1662-1669.

[5]李莹. 云环境下的数据存储安全技术研究[D].山东师范大学,2016.

[6]姜利娟. 云数据存储保护技术研究[D].扬州大学,2019.

[7]赵嘉诚. 基于云平台的数据存储安全技术与应用[D].南京邮电大学,2019.

[8]Craig Gentry. Fully Homomorphic Encryption Using Ideal Lattices. In the 41st ACM Symposium on Theory of Computing (STOC), 2009.

[9] Van Dijk M, Gentry C, Halevi S, et al. Fully homomorphic encryption over the integers[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 2010: 24-43.

[10]Brakerski Z, Vaikuntanathan V. Efficient fully homomorphic encryption from (standard) LWE[J]. SIAM Journal on Computing, 2014, 43(2): 831-871.

[11]Brakerski Z, Vaikuntanathan V. Fully homomorphic encryption from ring-LWE and security for key dependent messages[C]//Annual cryptology conference. Springer, Berlin, Heidelberg, 2011: 505-524.

[12]Acar A, Aksu H, Uluagac A S, et al. A survey on homomorphic encryption schemes: Theory and implementation[J]. ACM Computing Surveys (CSUR), 2018, 51(4): 1-35.

[13] Kocabas O, Soyata T. Utilizing homomorphic encryption to implement secure and private medical cloud computing[C]//2015 IEEE 8th International Conference on Cloud Computing. IEEE, 2015: 540-547.

[14] El Makkaoui K, Ezzati A, Hssane A B. Challenges of using homomorphic encryption to secure cloud computing[C]//2015 International Conference on Cloud Technologies and Applications (CloudTech). IEEE, 2015: 1-7.

[15] Jayapandian N, Rahman A M J M Z. Secure and efficient online data storage and sharing over cloud environment using probabilistic with homomorphic encryption[J]. Cluster Computing, 2017, 20(2): 1561-1573.

[16]Halevi S, Shoup V. Design and implementation of a homomorphic-encryption library[J]. IBM Research (Manuscript), 2013, 6: 12-15.

[17]Halevi S, Shoup V. Algorithms in helib[C]//Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2014: 554-571.

[18]Halevi S, Shoup V. Faster homomorphic linear transformations in HElib[C]//Annual International Cryptology Conference. Springer, Cham, 2018: 93-120.

[19] Chen H, Laine K, Player R. Simple encrypted arithmetic library-SEAL v2.1[C]//International Conference on Financial Cryptography and Data Security. Springer, Cham, 2017: 3-18.

[20]杨竞. 同态加密关键技术研究[D].电子科技大学,2019.

[21]何伟超. 基于同态加密隐私保护的机器学习关键技术研究[D].电子科技大学,2019.
[22]吴卓伟. 云环境下整数上的全同态加密算法研究[D].广西大学,2018.
[23]梁雷元. 基于云计算的全同态加密研究[D].新疆财经大学,2018.
[24]李帅. 基于同态加密技术的云安全存储模型研究[D].中国矿业大学,2015.
[25]徐文玉,吴磊,阎允雪.基于区块链和同态加密的电子健康记录隐私保护方案[J].计算机研究与发展,2018,55(10):2233-2243.
[26]郭俊彦.基于 LWE 全同态加密的电子健康档案系统的研究与实现[D].北京邮电大学,2018.
[27]李雅囡. 基于同态加密的电子评分系统的研究与实现[D].辽宁大学,2019.
[28]韩邦,李子臣,汤永利. 基于同态加密的全文检索方案设计与实现[J/OL]. 计算机工程与应用:1-8.http://kns.cnki.net/kcms/detail/11.2127.tp.20191115.1154.006.html.
[29]王利娟. 基于全同态加密算法的企业云存储密文检索研究[D].广西民族大学,2019.
[30]佘维,陈建森,刘琦,胡跃,顾志豪,田钊,刘炜. 一种面向医疗大数据安全共享的新型区块链技术[J]. 小型微型计算机系统,2019,40(07):1449-1454.
[31]李孟天.基于环 LWE 的全同态加密方案关键技术研究[D].战略支援部队信息工程大学,2018.
[32] Taher ElGamal. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms (PDF). IEEE Transactions on Information Theory. 1985, 31 (4): 469–472. doi:10.1109/TIT.1985.1057074. (conference version appeared in CRYPTO'84, pp. 10–18)
[33] Paillier, Pascal (1999). "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes". EUROCRYPT. Springer. pp. 223–238. doi:10.1007/3-540-48910-X_16.