浙江理工大學
ZHEJIANG SCI-TECH UNIVERSITY

硕士学位论文

（专业学位）

基于全同态加密的安全矩阵运算研究与应用

Secure Matrix Computation Based on Full Homomorphic Encryption and Its Applications

作　　者：宗浩然
类别（领域）：工程硕士（计算机技术）
指 导 教 师：黄海　副教授
所 在 学 院：信息学院

完成日期：二〇二一年四月

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

浙江理工大学学位论文独创性声明

本人声明：所提交的学位论文是本人在导师指导下，独立进行的研究工作及取得的研究成果。除了文中已经标注和引用的内容外，本论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得浙江理工大学或其他教育机构的学位证书而使用过的材料。对本论文研究作出贡献的个人或集体，均已在论文中作了明确的说明并表示谢意。本人承担本声明的法律责任。

论文作者签名：宗浩然

签字日期：2021 年 6 月 9 日

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

浙江理工大学学位论文版权使用授权书

本学位论文作者完全了解浙江理工大学有权保留并向国家有关部门或机构送交本论文的复印件和磁盘，允许论文被查阅和借阅。本人授权浙江理工大学可以将学位论文的全部或部分内容编入有关数据库进行检索和传播，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

（涉密的学位论文在解密后适用本授权书）

论文作者签名：宗浩然

签字日期：2021 年 6 月 9 日

导师签名：黄海

签字日期：2021 年 6 月 9 日

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

摘 要

矩阵运算是数学领域最基本的问题之一，其在科学和工程领域都有着广泛的应用。矩阵运算在应用中往往需要大量的计算资源，因此计算资源有限的本地可能选择将计算交给云服务器处理以节省本地计算资源。但是对于本地的隐私数据，直接上传到云端显然是不安全的。全同态加密技术支持密文间的计算，是解决隐私计算的方法之一。本文基于全同态加密技术研究安全矩阵运算及其应用，主要工作如下：

1.提出了基于全同态加密的安全矩阵乘法方案。现有的加密矩阵乘法方案只支持两个方阵之间的运算。本文基于超立方结构编码方法，提出了支持任意维度的安全高效的矩阵乘法计算方法。本方法是第一个基于全同态加密的任意维度安全矩阵乘法方案。在选择明文攻击(IND-CPA)模型下，本方法具有足够的安全性，可以保证矩阵信息不泄露。实验结果表明本方法对不同维度的矩阵都具有良好的性能。为了证明该算法的实用性，本文将其应用到多矩阵连乘上。实验结果表明本方法明显优于现有的安全多矩阵连乘方法。

2.将安全矩阵运算应用到支持向量机加密图像分类中。现有的支持向量机加密图像分类方法将图像特征向量的每个元素单独加密成一个密文分类，具有很高的时间复杂度和空间复杂度。本文应用单指令多数据(SIMD)技术，将整个图像的特征向量加密为一个矩阵密文，节约了存储空间，并利用 SIMD 的并行技术，减少了分类过程中密文的运算次数，大大地提高了分类计算效率。另外，本文应用近似的同态比较方法处理分类中最后的非线性符号判断问题。本方法与最新的工作相比显著地降低了通信和计算开销，实验结果表明只需几秒钟便可实现加密图像分类，而现有的方法需要几百秒。

关键词：安全矩阵计算；任意矩阵乘法；加密图像分类；全同态加密

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

Abstract

Matrix computation is one of the most basic problems in the field of mathematics. It is widely used in science and engineering. Matrix computation often requires a large number of computing resources in applications, local users with limited computing resources may choose to hand over the calculation to the cloud server in order to save local computing resources. Uploading directly to the cloud is obviously unsafe for local privacy data. Full homomorphic encryption technology supports ciphertext computing, which is one of the methods to solve privacy computing. This paper investigates secure matrix computation and its application based on full homomorphic encryption. The main contributions of this paper are as follows:

1.We propose a secure matrix multiplication scheme based on fully homomorphic encryption. The existing work supports only the case of the square matrices. This paper investigates secure matrix multiplication for arbitrary matrix based on fully homomorphic encryption.Our proposal is the first secure matrix multiplication scheme for arbitrary matrix based on fully homomorphic encryption. Our method has sufficient security to ensure that matrix information is not leaked in the chosen-plaintext attack (IND-CPA) model. Experimental results show that our scheme has excellent performance for the matrices with different dimensions.To demonstrate the applicability of the proposed matrix multiplication, we further apply it to secure multiple matrices multiplication. Experimental result shows that our solution significantly outperforms the latest secure multiple matrices multiplication scheme.

2. Application of security matrix computation to support vector machine encrypted image classification. A naive SVM classification over encrypted data method is encrypting each element of image feature vector into one ciphertext. This method has high time and space complexity. We encrypt the whole feature vector into a single ciphertext based on SIMD technique and save the storage space. We also use SIMD parallel technology to reduce the number of operations between ciphertexts in the process of classification. Our method greatly improves the efficiency of classification calculation. What more, we use the approximate homomorphic comparison method to deal with the final nonlinear computation of classification. Our method significantly improves communication overhead and computational performance

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

compared to the existing work. Experimental results show that our method takes only seconds to classify an encrypted image while the state-of-art work takes several hundred seconds.

Keywords: Secure Matrix Computation; arbitrary matrix multiplication; encrypted image classification; fully homomorphic encryption

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

目 录

摘 要..... I
Abstract..... II
第一章 绪论..... 1
1.1 课题背景及意义..... 1
1.2 国内外研究现状..... 2
1.3 主要研究内容..... 4
1.4 论文结构安排..... 4
第二章 全同态加密相关技术..... 6
2.1 全同态加密基本概念..... 6
2.2 CKKS 方案..... 6
2.3 SIMD 技术..... 7
2.4 超立方结构..... 8
2.5 本章小结..... 8
第三章 基于全同态加密的矩阵乘法方案..... 9
3.1 引言..... 9
3.2 相关背景知识介绍..... 10
3.2.1 系统与安全模型..... 10
3.2.2 矩阵向量乘的对角方法..... 11
3.3 基于全同态加密的任意维度矩阵乘..... 11
3.3.1 同态矩阵乘情况一..... 12
3.3.2 同态矩阵乘情况二..... 17
3.4 实验结果与安全性..... 22
3.4.1 时间复杂度分析..... 22
3.4.2 实验效果比较..... 23
3.4.3 与其他相关工作的比较..... 24
3.4.4 安全性分析..... 25

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

3.5 安全多矩阵连乘中的应用 25
3.6 本章小结 26
第四章 基于支持向量机的加密图像分类 27
4.1 引言 27
4.2 相关背景知识介绍 28
4.2.1 系统与安全模型 28
4.2.2 支持向量机分类 29
4.2.3 HOG 特征提取技术 30
4.2.4 PCA 数据降维 30
4.3 基于支持向量机的加密图像分类 31
4.3.1 特征向量的加密 31
4.3.2 支持向量机分类的同态运算 32
4.4 同态计算非线性函数 36
4.4.1 求逆算法 36
4.4.2 比较算法 37
4.5 实验结果与安全性分析 38
4.5.1 时间复杂度对比 38
4.5.2 运行时间对比 39
4.5.3 同态近似比较实验结果 40
4.5.4 与其他类似工作的比较 41
4.5.5 安全性分析 42
4.6 本章小结 42
第五章 总结与展望 43
5.1 总结 43
5.2 展望 43
参考文献 45
攻读硕士期间的研究成果 50
致 谢 51

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

第一章 绪论

1.1 课题背景及意义

云计算是一种基于互联网的计算方式。通过这种方式，个人或组织可以获取各种 IT 资源（如服务器、存储和应用程序）的按需网络访问权。这些资源可以快速部署，以提高效率、减少管理开销。如今，云计算越来越受到关注，在各领域广泛应用。我们的日常生活也离不开云服务，如用来存储数据、编写文档、在线办公和线上游戏。云计算还为物联网、大数据和人工智能等技术发展提供了基础设施，加速了行业演化，并推动数字化转型。然而云计算带来大量好处和机会的同时也带来了一些安全问题，特别是当用户上传敏感的医疗、金融等隐私数据到云上时。

显而易见，云服务器会对用户所上传处理的数据感到好奇并很轻易的读取到。近年来的互联网隐私泄露问题不胜枚举，数据隐私问题也越来越受到重视。2017 年 11 月，美国五角大楼数据库配置错误导致 18 亿公民信息惨遭泄露。2018 年，用户数据泄漏事件多次发生在 Facebook，有超过 5000 万用户的姓名、联系方式、登录位置、私人照片等隐私数据被泄漏。2018 年 9 月，瑞士一家数据管理公司隐私泄露，客户姓名、居住地以及联系方式等个人信息均遭外泄。国内隐私泄漏事件更是层出不穷。由上述众多的隐私泄漏事件来看，随着云计算、大数据等科技兴起，数据泄漏频发，其引发的安全隐患受到社会的广泛关注。据云安全联盟(Cloud Security Alliance)[1]称，数据隐私问题仍然是云计算被广泛使用的主要障碍。

全同态加密(FHE)[2]支持在不解密的情况下对加密数据进行任意次的运算，这一特性使它成为云计算背景下安全计算中很有前景的工具。具体而言，计算资源受限的客户端向云服务器发送同态加密数据，请求云服务器对加密数据执行某种计算或处理，并将加密结果返回客户端。最后，客户端利用持有的私钥解密计算结果，整个过程保证无数据泄露。全同态加密是对加密数据进行计算的一种强有力的工具，在云计算安全领域日益流行。全同态加密的特性为敏感数据应用于云服务的隐私泄露问题提供一个解决方案，例如基因组序

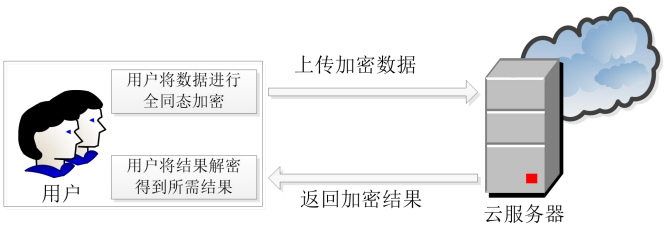

图 1.1 全同态加密在云计算中的应用

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

列分析、市场研究以及医学图像处理等外包计算中。同时其在机器学习[3]、图像分析与处理[4]等领域应用广泛。

矩阵是高等代数学中的常见工具，在许多学科领域中都有应用。矩阵运算即对矩阵中数据进行处理，例如矩阵乘法可以表示为对这个矩阵进行收缩、拉升、旋转、平移等操作。矩阵运算在图像分析、机器学习、数值分析等各个领域都有着广泛的应用。矩阵的奇异值分解可以用于图像、文件压缩以及数据降维。此外，矩阵的特征分解、谱分解可用于图傅里叶变换中，继而基于傅里叶变换衍生出卷积神经网络等一系列技术。因此矩阵运算的效率直接影响着许多高层应用的计算性能。

在云服务如此普及的今天，云端在给客户提供各种服务时避免不了用到矩阵运算。高效安全的矩阵运算可以在保护客户隐私的同时提高云服务的效率，为云计算的广泛使用提供前提条件。因此本课题基于全同态加密研究安全矩阵运算及其应用具有重要理论和实际意义。

1.2 国内外研究现状

近年来，随着同态加密技术的普及，有许多安全矩阵运算项目基于同态加密展开研究。

2008 年 Benjamin 等[5]基于同态加密在两个服务器不合谋的前提下提出了一个安全矩阵乘的方案。但他们的方案易受到服务器间的共谋攻击且计算效率低下，其时间复杂度为 $O(n^2)$，其中 n 为输入矩阵的维度。

2011 年 Wang 等[6]基于雅克比迭代法和加同态加密(AHE)[7]设计了一个解决线性方程 $Ax=b$ 的协议。他们的方案的主要缺点是需要客户端承担 $O(n)$ 的计算复杂度、输出的结果是近似值和只适用于特定的输入矩阵 A。

2011 年 Mohassel 等[8]基于加同态加密研究了一系列安全矩阵运算方案，其中包括矩阵求逆和矩阵行列式等。然而由于加同态加密只支持同态加法，因此他们的方案需要服务器和客户端之间进行多轮交互，导致了较高的通信量。

2015 年 Hiromasa 等[9]构造了一个名为 GSW 的全同态加密矩阵方案，该方案可以将矩阵表示为一个密文，并且完成两个矩阵相乘只需要一次同态乘法。然而他们的方案只适用于二进制方阵矩阵乘，如 $A_{n\times n}\times B_{n\times n}$ 且矩阵元素的范围为 $\{0,1\}$。

2016 年 Duong 等[10]提出了一种矩阵加密方法，该方法可以将一个矩阵加密为一个单独的密文，且只需要一次同态乘法即可实现矩阵乘。然而，其方案的一个缺点是，两个输

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

入矩阵的乘积与输出矩阵的形式不相同，因此不能与下一个矩阵相乘。因此，他们的方法不具备可组合性，该特性在一些上层应用中起着关键作用，例如矩阵连乘中。

2018 年 Rathee 等[11]基于 BGV 方案的全同态加密提出了一种平方矩阵乘方法。他们利用超立方结构把矩阵编码加密为一个密文，他们的方案实现两个矩阵乘的时间复杂度为 $O(n)$ 次同态乘。但是他们的方法只适用于方阵。

2018 年 Cheon 等[12]提出了一种多维的支持近似数的全同态加密方案名为 MHEAAN，该方案的特点是将原 HEAAN 方案扩展为支持超立方结构。然后他们基于 MHEAAN 提出了安全矩阵乘的方法，他们所用的技术类似于 Rathee 等[11]。同样的他们的方案也只考虑了方阵的情况。

2018 年 Jiang 等[13]基于支持近似数的 HEAAN 全同态加密方案提出了安全平方矩阵乘方案，他们的方法利用 SIMD 技术将矩阵打包加密为一个密文，两个矩阵乘需要 $O(n)$ 次同态乘法。此外，他们的方案将原有的只支持方阵的乘扩展到非方阵矩阵乘。然而他们的方案只考虑了特殊形式的非方阵，如 $A_{m\times n}\times B_{n\times n}$ 其中 $m<n$ 。

2020 年 Kim 等[14]研究了基因组分析的安全外包计算，提出了一种基于全同态加密的安全矩阵行列式计算方案。但他们的方案中使用矩阵行列式的标准定义计算行列式结果，该方法需要 $n!$ 次迭代，其中每次迭代需要 $O(n)$ 次同态乘，这导致了极高的计算量。

基于伪装的技术是另一种实现安全矩阵运算的方法。该技术利用矩阵的特定性质，设计安全矩阵运算的特殊协议。2002 年 Atallah 等[15]提出了许多适合于矩阵乘法、矩阵求逆和线性方程组求解等矩阵计算相关问题的伪装技术。此后，Lei 等[16]-[18]基于伪装技术提出了一系列与矩阵运算相关的方案，其中包括矩阵乘、矩阵行列式和矩阵求逆。随后的一些工作[19]-[24]又分别基于伪装技术提出了他们特定的安全矩阵运算方案。虽然这些自定义解决方案通常具有更好的性能，但基于伪装的技术安全性较弱，很难达到基于密码学技术方案的强度，易造成信息泄露[24]。此外，现有的基于伪装技术的方案都不具有可组合性，因此不适用于多矩阵乘法，也无法将其应用到上层应用中。

安全多方计算(MPC)[25]，例如混淆电路(Garbled circuits(GC))，也是一种实现矩阵安全运算的方法。安全多方计算是一种交互式计算模式，它允许双方在不泄露各自输入的情况下协作计算共同的输出。安全多方计算的运算效率更快，但缺点是需要客户端与服务器进

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

行多轮交互并导致较高的通信开销。相反，全同态加密除必要的上传和下载开销外不需要额外的通信开销。此外，安全多方计算的双方同时需要分担计算开销，此不利于计算资源受限的客户。

本研究基于全同态加密设计安全矩阵运算方案，与安全多方计算的方案相比通信开销更小，与基于伪装的技术相比更安全。

1.3 主要研究内容

针对上文中所介绍的安全矩阵运算国内外研究现状，本文围绕安全矩阵运算问题展开，研究了基于全同态加密的任意矩阵乘法计算和基于矩阵运算的加密图像分类应用，论文主要研究内容如下：

1.基于全同态加密的任意矩阵乘法计算方法。首先，本文利用超立方结构提出了一个高效的单密文安全矩阵乘法方案，形如$C_{m\times n}=A_{m\times l}\times B_{l\times n}$。本文的解决方案是第一种基于全同态加密的任意矩阵的单密文且具有可组合性的安全矩阵乘法方案。该方法只需l次同态乘即可实现安全矩阵乘。实验结果表明，该方案对不同维数的矩阵具有良好的性能。其次，该方法被应用于非方阵矩阵连乘中，实验结果表明本文的解决方案明显优于最新的安全多矩阵乘法方案。

2.基于全同态加密提出了一个高效的支持向量机分类算法。不同于将单个元素分别加密，本文利用 SIMD 技术将整个特征向量加密成一个密文。该方法将密文的数量从 t 降低到 1（t 是特征向量的维度）。此外，应用近似的同态比较算法处理分类最后的非线性运算。本文的方法与最近的其他工作相比显著地提高了计算效率，实验结果表明该方法仅需要几秒即可分类一个图像，而之前的方法通常需要几百秒。

1.4 论文结构安排

本文由五章构成，其具体内容如下：

第一章：阐述课题背景及研究意义，介绍安全矩阵运算国内外研究现状。最后描述本文主要研究内容和结构安排。

第二章：本章主要介绍了全同态加密的基本概念及相关技术。主要包括 CKKS 方案、SIMD 技术以及超立方结构相关技术，分析了上述技术及其相关技术的优缺点，并加以总结，为之后的研究提供技术支撑。

第三章：基于全同态加密的任意矩阵乘法计算方法。首先分析了最近的关于加密矩阵

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

乘法方案的优缺点，之后分析本文所用到的系统安全模型并详细地介绍了密文矩阵乘法的两种具体情况，最后进行了实验结果对比和安全性分析。

第四章：将安全矩阵运算应用到支持向量机加密图像分类中。首先介绍了关于支持向量机加密图像分类的相关工作以及系统安全模型，接着介绍了本文用到的图像分类的相关技术。随后介绍了基于矩阵运算技巧提出的加密图像分类方案，针对支持向量机最后的非线性计算问题提供了一个同态近似计算的解决方法，最后进行了实验对比和安全性分析。

第五章：总结本文提出的安全矩阵运算方法，描述未来的研究方向。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

第二章 全同态加密相关技术

2.1 全同态加密基本概念

全同态加密可以分为三代。第一代方案基于理想格[2]或整数[26]；第二代基于容错学习(LWE)[27]或环容错学习(RLWE)，例如 BGV[28]、HEAAN 方案[29]；第三代基于近似特征向量，例如 GSW 方案[30]。

全同态加密(FHE)允许在不知道私钥的情况下对 x 进行任意的函数 f 运算。由此可以得到函数结果 $f(x)$。全同态加密体系包括以下四个算法。

1. 密钥生成算法：$(pk,sk)\leftarrow KeyGen(1^{\lambda})$：给定安全参数 λ，输出公钥 pk 和私钥 sk；

2. 加密算法：$c\leftarrow Enc(pk,m)$：给定公钥 pk 和明文 m，输出密文 c；

3. 解密算法：$m\leftarrow Dec(sk,c)$：给定私钥 sk 和密文 c，输出解密后的明文 m；

4. 密文计算算法：$c_f\leftarrow Eval(pk,f,c_1,\cdots,c_n)$：输入公钥 pk、函数 $f:P^n\rightarrow P$ 和由明文集合 $(m_1,...,m_n)$ 加密组成的密文集合 $(c_1,...,c_n)$，输出 $f(m_1,...,m_n)$ 的加密 c_f。

全同态加密技术对任意的加法和乘法计算是同态的。本文中用 $\oplus$ 和 $\otimes$ 表示密文的加法和乘法。如下：

$$Dec(Enc(x_1)\oplus Enc(x_2))=x_1+x_2 \tag{2-(1)}$$

$$Dec(Enc(x_1)\otimes Enc(x_2))=x_1\times x_2 \tag{2-(2)}$$

2.2 CKKS 方案

实现全同态加密的方案有很多，比较典型的有 BGV 方案[28]和 CKKS 方案[31]。简单来说，BGV 方案只支持整数环上的同态方案，而 CKKS 方案支持浮点数的同态方案。给定两个密文 m_1 和 m_2，该方案支持在预定的精度下安全的计算 m_1+m_2 和 $m_1\times m_2$ 的近似值。该方案的核心思想是将 RLWE 问题[27]的插入噪声作为近似误差的一部分。该方案最重要的特征是明文的 rounding（舍入）操作。

下面简单介绍一下该方案的结构，方便起见，确定一个基数 $p>0$ 和模数 q_0，让 $q_l=p^l\cdot q_0$，其中 $0<l\le L$。整数 p 在近似计算中被用作 scaling 的基数。对于一个给定的安全参数 λ，选择参数 $M=M(\lambda,q_L)$ 作为循环多项式。对于一个给定层数 $0<l\le L$，层数为 l

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

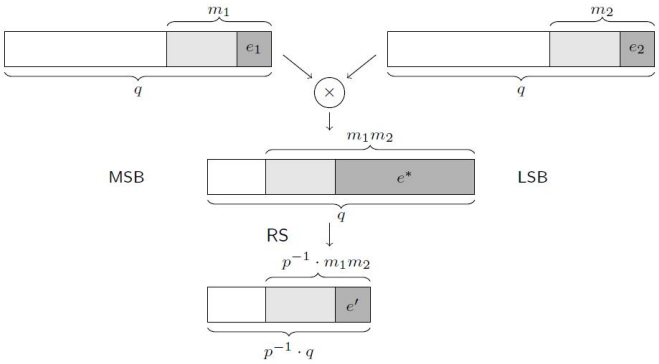

图 2.1 CKKS 方案的同态乘和 rescaling 操作

的密文是一个属于 $R_{q_l}^k$ 域的向量。该 HE 方案在多项式环 $R=\mathbb{Z}[X]/(\Phi_M(X))$ 上。

1. 密钥生成算法 $KeyGen(1^\lambda)$。输出密钥 sk，公钥 pk 和评估授权密钥 evk。

2. 加密算法 $Enc_{pk}(m)$。对于一个给定的多项式 $m\in R$，输出密文 $c\in R_{q_L}^k$。m 的加密 c 满足 $\langle c,sk\rangle = m+e(\mathrm{mod}\, q_L)$，$e$ 是一个小噪音。

3. 解密算法 $Dec_{sk}(c)$。给定一个层数为 l 的密文 c，利用私钥输出多项式 $m'\leftarrow\langle c,sk\rangle(\mathrm{mod}\, q_l)$。

与大多数现有的方案不同，该方案没有与噪音分开的明文空间。解密的结果 $m'=m+e$ 与明文 m 略有不同，但是在误差 $\|e\|_\infty^{can}$ 比 $\|m\|_\infty^{can}$ 足够小的情况下，可以看作是明文的近似值。

4. 加法算法 $Add(c_1,c_2)$。对于给定的两个密文 m_1 和 m_2，输出 m_1+m_2 的密文。输出密文的误差限制在两个密文误差的和之内。

5. 乘法算法 $Mult_{evk}(c_1,c_2)$。对于给定的两个密文 c_1 和 c_2，输出密文 $c_{mult}\in R_{q_l}^k$ 满足 $\langle c_{mult},sk\rangle=\langle c_1,sk\rangle\cdot\langle c_2,sk\rangle+e_{mult}(\mathrm{mod}\, q_l)$。其中 $e_{mult}\in R$ 且 $\|e\|_\infty^{can}\le B_{mult}(l)$。

2.3 SIMD 技术

基于环 LWE(RLWE)[27]问题的全同态加密的一个显著特征是 SIMD 技术[32]（单指令多数据），该技术支持把 l 条（明文槽数）信息编码到一个明文多项式中。然后，可以将明文多项式加密为密文，并在明文槽上执行 l 分量加法或乘法操作，而仅对密文进行一次操作。例如，给定两个密文 c_1 和 c_2 分别代表 $(x_1,\cdots,x_n)$ 和 $(y_1,\cdots,y_n)$。$c_1\oplus c_2$ 和 $c_1\otimes c_2$ 分别表

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

示$(x_1+y_1,\cdots,x_n+y_n)$和$(x_1\times y_1,\cdots,x_n\times y_n)$。本文充分利用 SIMD 技术支持密文并行运算的特性提出了性能良好的安全矩阵运算算法。此外，基于 RLWE 问题的全同态加密方案利用伽罗瓦(Galois)群的结构实现了对明文槽的旋转操作，该操作也为接下来的算法提供了技术支持。具体如下，给定$(x_1,\cdots,x_n)$的密文c，执行旋转操作$Rotate(c,l)$得到$(x_{l+1},\cdots,x_n,x_1,\cdots,x_l)$的密文。

2.4 超立方结构

超立方结构[33]是对 SIMD 技术[32]改进，其将线性数组排列扩展到二维的超立方结构上。超立方结构的这个特性在更大程度上利用了 SIMD 操作，支持在矩阵的维度上对数据进行旋转操作，同时也提高了空间和时间复杂度。超立方结构只支持 BGV 方案[28]。

简单来说，[33]中利用二维明文槽的超立方体排列，以优化加密矩阵和向量上的矩阵操作。与一般的 SIMD 技术不同，此时密文空间是二维的，超立方结构支持行旋转与列旋转。超立方结构支持以下操作：

$Rotate1D(ct,i,k)$：将超立方结构沿着第i维旋转k个位置，移动插槽中的内容$(e_0,...,e_i,...,e_{d-1})$为$(e_0,...,e_i+k \bmod m_i,...,e_{d-1})$。

二维的超立方结构将明文槽打造为矩阵形式且支持按行按列旋转，这对以下本文的安全矩阵乘法方案起到了很大的作用。

2.5 本章小结

本章主要介绍了全同态加密的相关技术，首先介绍了全同态加密和部分同态加密的区别，其次分别介绍了实现全同态加密的 CKKS 方案以及适用于 CKKS 方案和 BGV 方案的 SIMD 技术和适用于 BGV 方案的超立方结构。因为本论文中的方法都是基于密文间运算的，所以研究并了解全同态加密底层的相关技术对本文的研究至关重要。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

第三章 基于全同态加密的矩阵乘法方案

在数学领域，矩阵运算是最基本的问题之一，且矩阵运算在科学和工程领域都有着广泛的应用，包括数据分析、图像处理、机器学习等等。许多的应用问题都可以被表示为矩阵运算，例如深度学习模型的训练与预测可被简化为一系列的矩阵运算。在云时代到来的今天，用户要想在不泄露个人隐私的前提下利用云服务的计算力，那么研究如何高效且保证敏感数据安全的矩阵运算就至关重要了。本章基于全同态加密提出了支持任意维度的安全矩阵乘法方案。

3.1 引言

目前已经存在许多基于全同态加密的矩阵乘法研究。Halevi 等[34]提出了安全的矩阵-向量乘法例如 $w = A_{n\times n}v$，其中 w、$A_{n\times n}$ 和 v 都是被加密过的密文。他们提出了两种矩阵打包加密方式分别叫做行打包和列打包，即利用 SIMD 技术将一整行元素或一整列元素打包加密为一个简单的密文。Lu 等[35]和 Wang 等[36]分别根据矩阵-向量乘法利用行打包技术和列打包技术将该方法扩展到矩阵和矩阵之间的乘法。然而他们的方法都需要 n 个密文来表示一个矩阵，这导致了 $O(n)$ 的空间复杂度和 $O(n^2)$ 的同态计算复杂度。

Duong 等[10]基于全同态加密提出了一种全新的将矩阵打包加密为单一密文的方法，且只需一次乘法即可完成矩阵乘。但按照他们方案所生成的矩阵形式与输入矩阵形式不同，所以只能进行一次乘法，实际中很少出现这种情况，故他们的方案不实用。

Hiromasa 等[9]基于全同态加密设计了名为 GSW 的矩阵方案，该方案也可以将矩阵打包成一个单一密文且完成两个矩阵相乘只需要一次同态乘。但是他们的方案仅适用于特定的矩阵，即二进制方阵。

Rathee 等[11]基于超立方结构设计安全矩阵乘法方案。他们的方案利用超立方结构把矩阵编码加密为一个密文，该算法的时间复杂度为 $O(n)$ 次同态乘。但是他们的方法只适用于方阵。

Jiang 等[13]基于支持近似数的 HEAAN 全同态加密方案提出了安全平方矩阵乘法方案，该方案利用 SIMD 技术将矩阵打包加密为一个密文，两个矩阵相乘需要 $O(n)$ 次同态乘法。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

虽然他们的方案将原有的只支持方阵的乘法方案扩展到非方阵矩阵乘法，但他们的方案只考虑了特殊形式的非方阵，如 $A_{m\times n}\times B_{n\times n}$ 其中 $m<n$ 。

Cheon 等[12]提出了一种名为 MHEAAN 的全同态加密方案。该方案是 HEAAN 方案的扩展，将原 HEAAN 方案扩展为支持超立方结构，同时支持近似数。然后他们基于 MHEAAN 设计了安全矩阵乘法方案。同样他们的方案也只考虑了方阵的情况。

以上的[9]-[13]工作代表着最新的单密文、可组合性安全矩阵乘方案，但是它们只适用于平方矩阵或者特殊的非方阵，不支持一般的非方阵乘。本章基于全同态加密技术，提出了基于全同态加密的任意矩阵乘法方案。主要贡献如下：

1. 本文利用超立方结构提出了一个高效的单密文安全矩阵乘法方案，形如 $C_{m\times n}=A_{m\times l}\times B_{l\times n}$ 。该方法只需 l 次同态乘即可实现安全矩阵乘。实验结果表明该方案对不同维度的矩阵都具有良好的性能。例如，两个加密的平方矩阵 $A_{64\times 64},B_{64\times 64}$ 相乘仅需 1.297s，0.0202s 即可完成两个加密的非方阵 $A_{64\times 1},B_{1\times 64}$ 相乘。

2.将该方案应用到安全矩阵连乘上以证明其实用性。实验结果表明该方法明显优于最新的安全多矩阵乘法方案。对于维度为 40,30,35,15,60,5,70,10,50,20,25 的十个非方阵连乘，本文方法只需要 56s，而原来的方法需要 9710s。

3.2 相关背景知识介绍

3.2.1 系统与安全模型

如图 3.1 所示，本文的系统模型包含两个部分，具有有限计算资源的客户端和具有强大计算能力的云服务商提供计算服务。假设客户端一次性初始化公钥 pk 和私钥 sk 。每一次客户端将要执行矩阵乘，先将矩阵 A、B 用公钥 pk 打包，然后将加密好的密文 C_1、C_2 发送到云服务器。云服务器执行同态的安全矩阵乘运算，并将结果密文 C^* 返回客户端。最后客户端用私钥 sk 解密 C^* 得到 $A\times B$ 的结果。

假设客户端是诚实的，安全威胁主要来自服务器的行为。本文采用半诚实威胁模型[37]，该模型假设服务器端遵守协议但可能试图了解有关客户端数据的其他信息。本研究目的是提出一种安全的矩阵乘法协议，其中服务器提供计算服务，而不了解客户端的矩阵信息。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

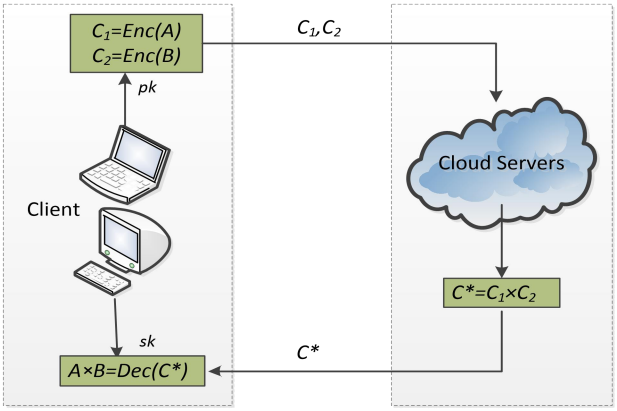

图 3.1 安全矩阵乘系统模型

3.2.2 矩阵向量乘的对角方法

上文中 Halevi 等[34]除了提出了行打包和列打包方法外，还提出了一种矩阵向量乘的对角方法。假设 $A_{n\times n}$ 是一个 $n\times n$ 明文方阵，向量 v 是加密过的密文，w 是一个向量。则矩阵向量乘的对角方法 $w=A_{n\times n}v$ 具体介绍如下。矩阵 $A_{n\times n}$ 的对角线元素定义为 $d_i=(a_{0,i},a_{1,i+1},...,a_{n-1,i-1})$，其中 $0\le i<n$ 且序号 i 在模 n 以下。$v<<i$ 表示向量 v 旋转 i 个位置。则两者的积 $w=Av$ 计算方式如下 $w=\sum_{i=0}^{n-1}d_i\times(v<<i)$。图 3.2 介绍了如何根据对角线方法计算矩阵向量乘。

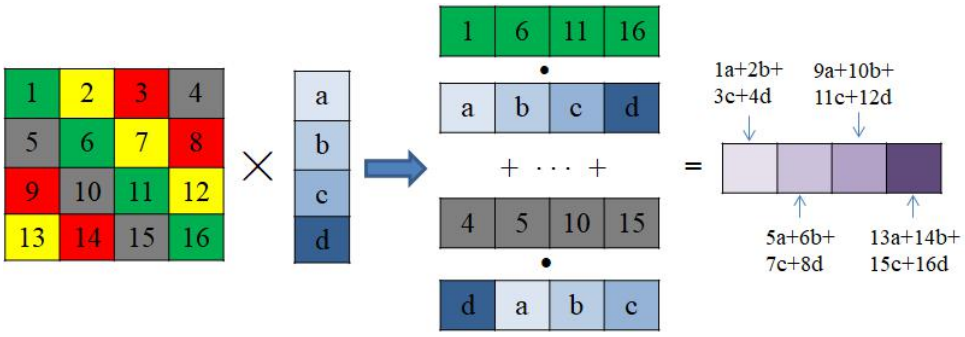

图 3.2 矩阵向量乘的对角线方法

3.3 基于全同态加密的任意维度矩阵乘

上文介绍过，传统的非方阵密文矩阵乘的方法是将矩阵填零扩充为方阵，之后利用安全平方矩阵乘的方法计算。如 $A_{m\times l},B_{l\times n}$ 扩充为 $A_{k\times k},B_{k\times k}$ 其中 $k=\max(m,l,n)$。同样的他们需要 $O(k)$ 次同态乘法。本小节详细地介绍一个高效的任意矩阵安全乘法算法，该方法基于对

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

角线方法并且只需 $O(l)$ 次同态矩阵乘。

上小节介绍的对角线方法只适用于矩阵向量的乘法，如 Av，且矩阵 A 是明文，向量 v 是密文。现在面临的问题是将明文矩阵与密文向量乘的方法扩展为密文矩阵之间的乘。对加密矩阵来说，要实现乘法，需要一个更有效的在明文槽之间数据移动的技术。超立方结构提供了这么一个技术，支持加密矩阵行与列之间数据的移动。本文的方法正是基于此。

本小节基于超立方结构和对角线方法提出了安全矩阵乘方法 $A_{m\times l}\times B_{l\times n}$，具体分以下两种情况讨论。

a. 矩阵 $A_{m\times l}$ 的行数小于等于列数，即 $m\le l$。

b. 矩阵 $A_{m\times l}$ 的行数大于列数，即 $m>l$。

3.3.1 同态矩阵乘情况一

假设有两个矩阵 $A_{m\times l}$ 和 $B_{l\times n}$ 其中 $m\le l$。假设 $t=\max(l,n)$。事先设置一个 $m_0\times m_1$ 的超立方结构，其中 $m_0=l$ 且 $m_1\ge t$。超立方结构可以看做是比之间两个矩阵更大的矩阵，如此矩阵 A 和 B 都可以放入超立方结构中，空余的位置填零。为了方便起见，假设 $m_0=l$ 且 $m_1=t$，易知本方法在 $m_0=l$ 且 $m_1\ge t$ 情况下同时有效。之后把矩阵 $A_{m\times l}$ 和 $B_{l\times n}$ 分别加密为 $ct_{A_{m\times l}}$ 和 $ct_{B_{l\times n}}$。加密过程是在客户端进行的。下面的例子介绍输入矩阵是如何加密的。

假设矩阵 $A_{2\times 3}$ 和 $B_{3\times 5}$ 定义如下：

$$A_{2\times 3}=\begin{bmatrix}1&2&3\\4&5&6\end{bmatrix},\quad B_{3\times 5}=\begin{bmatrix}a&d&g&k&n\\b&e&h&l&o\\c&f&i&m&p\end{bmatrix} \tag{3-(1)}$$

超立方结构的大小为 3×5。现将矩阵 $A_{2\times 3}$ 和 $B_{3\times 5}$ 分别加密为 $ct_{A_{2\times 3}}$ 和 $ct_{B_{3\times 5}}$。

$$ct_{A_{2\times 3}}=Enc\left(\begin{bmatrix}1&2&3&0&0\\4&5&6&0&0\\0&0&0&0&0\end{bmatrix}\right) \tag{3-(2)}$$

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$ct_{B_{3\times5}} = Enc\left(\begin{bmatrix} a & d & h & k & n \\ b & e & i & l & o \\ c & f & j & m & p \end{bmatrix}\right) \tag{3-(3)}$$

以下步骤 a1 到步骤 a4 介绍同态矩阵乘，该运算在服务器端。

步骤 a1：$A_{l\times l}$ 表示超立方结构 $m_0 \times m_1$ 中最左边的 $l \times l$ 平方矩阵。本步骤用乘掩码的操作将 $A_{l\times l}$ 中的对角线元素提取出来，如下：

$$ct_{d_i} = U_i \otimes ct_{A_{m\times l}} \tag{3-(4)}$$

其中 $0 \le i < l$，$U_i[I][J](0 \le I < l, 0 \le J < t)$ 定义如下：

$$U_i[I][J] = \begin{cases} 1 & If\ J = i + I \ (\mathrm{mod}\, l) \\ 0 & else \end{cases} \tag{3-(5)}$$

以下的例 a1 介绍如何提取 ct_A 的对角线元素。

例 a1：假设 $A_{3\times3}$ 是 3×5 超立方结构的最左边的 3×3 的平方矩阵。将对角线元素 $d_0 = (1,5,0)$，$d_1 = (2,6,0)$，$d_3 = (3,4,0)$ 分别提取出来，具体如下：

$$ct_{A_{2\times3}} = \left[\begin{array}{ccc|cc} 1 & 2 & 3 & 0 & 0 \\ 4 & 5 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right] \tag{3-(6)}$$

对于公式 3-(2)中给定的密文 $ct_{A_{2\times3}}$，对应的对角线元素提取后的密文 ct_{d_0}，ct_{d_1}，ct_{d_2} 如下：

$$\begin{aligned} ct_{d_0} &= Enc\left(\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right) \\ ct_{d_1} &= Enc\left(\begin{bmatrix} 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right) \\ ct_{d_2} &= Enc\left(\begin{bmatrix} 0 & 0 & 3 & 0 & 0 \\ 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right) \end{aligned} \tag{3-(7)}$$

步骤 a2：将 $ct_{d_i}(0 \le i < l)$ 分别沿行复制得到 $ct_{A_i}(0 \le i < l)$ 的密文。每行只有一个非零元素，

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

本步骤需要进行 $O(\log m_1)$ 次旋转和 $O(\log m_1)$ 次加法。具体如算法 1 所示，其中 $numBits(T)$ 表示 T 的二进制位数，$bit_j(T)$ 表示二进制 T 的第 j 位。

$$ct_{A_i} = \text{Replicate1D}(ct_{d_i}, 1, m, l, m_0, m_1) \tag{3-(8)}$$

以下的例 a2 展示了 $ct_{A_i}(0 \le i < l)$ 是如何得到的。

例 a2：根据公式 3-(7)，计算 $ct_{A_i}(0 \le i < l)$ 如下：

$$\begin{aligned} ct_{A_0} &= Enc\left(\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 5 & 5 & 5 & 5 & 5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right) \\ ct_{A_1} &= Enc\left(\begin{bmatrix} 2 & 2 & 2 & 2 & 2 \\ 6 & 6 & 6 & 6 & 6 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right) \\ ct_{A_2} &= Enc\left(\begin{bmatrix} 3 & 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right) \end{aligned} \tag{3-(9)}$$

Algorithm 1: Replicate1D($ct, i, p_0, p_1, m_0, m_1$)

Input: $ct: A_{p_0 \times p_1}$ 的密文

Input: i :第 i 维超立方结构

Input: p_0, p_1 :矩阵 $A_{p_0 \times p_1}$ 维度

Input: m_0, m_1 :超立方结构的维度

Output: $\overline{ct}$

1: $\overline{ct} = ct, e = 1$;

2: **if** $i = 0$ **then**

3: $T = m_0$;

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

4: **else**

5: $T = m_1$;

6: **end**

7: **for** $j = numBits(T) - 2$ to 0 **do**

8: $\overline{ct} = \overline{ct} \oplus Rotate1D(\overline{ct}, i, e)$;

9: $e = 2 \cdot e$;

10: **if** $bit_i(T) = 1$ **then**

11: $\overline{ct} = ct \oplus Rotate1D(\overline{ct}, i, 1)$;

12: $e = e + 1$;

13: **end**

14: **end**

15: return $\overline{ct}$.

步骤 a3：本步骤沿列将密文$ct_{B_{l\times n}}$旋转i位分别得到l个密文$ct_{B_i}(i = 0,1,...,l-1)$。

$$ct_{B_i} = Rotate1D(ct_B, 0, -i) \tag{3-(10)}$$

具体如例 a3。

例 a3：对于公式 3-(3)中给定的密文$ct_{B_{3\times 5}}$，本步骤得到的密文ct_{B_0}，ct_{B_1}，ct_{B_2}分别如下：

$$\begin{aligned} ct_{B_0} &= Enc\left(\begin{bmatrix} a & d & g & k & n \\ b & e & h & l & o \\ c & f & i & m & p \end{bmatrix}\right) \\ ct_{B_1} &= Enc\left(\begin{bmatrix} b & e & h & l & o \\ c & f & i & m & p \\ a & d & g & k & n \end{bmatrix}\right) \\ ct_{B_2} &= Enc\left(\begin{bmatrix} c & f & i & m & p \\ a & d & g & k & n \\ b & e & h & l & o \end{bmatrix}\right) \end{aligned} \tag{3-(11)}$$

步骤 a4：本步骤的目的是计算密文$ct_{C_{m\times n}} = \sum_{i=0}^{l-1} ct_{A_i} \otimes ct_{B_i}$。具体如例 a4。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

例 a4: 对于公式3-(9)、3-(11)中的密文 ct_{A_i} 和 ct_{B_i} $(i=0,1,2)$，以下为密文 $ct_{C_{2\times5}}=\sum_{i=0}^{3} ct_{A_i}\otimes ct_{B_i}$ 的具体例子，其中 $ct_{C_{2\times5}}$ 是 $C_{2\times5}=A_{2\times3}\times B_{3\times5}$ 加密前两行的密文。

$$
\begin{aligned}
&Enc\left(\begin{bmatrix}1&1&1&1&1\\5&5&5&5&5\\0&0&0&0&0\end{bmatrix}\right)\otimes Enc\left(\begin{bmatrix}a&d&g&k&n\\b&e&h&l&o\\c&f&i&m&p\end{bmatrix}\right)\\
&\oplus\\
&Enc\left(\begin{bmatrix}2&2&2&2&2\\6&6&6&6&6\\0&0&0&0&0\end{bmatrix}\right)\otimes Enc\left(\begin{bmatrix}b&e&h&l&o\\c&f&i&m&p\\a&d&g&k&n\end{bmatrix}\right)\\
&\oplus\\
&Enc\left(\begin{bmatrix}3&3&3&3&3\\4&4&4&4&4\\0&0&0&0&0\end{bmatrix}\right)\otimes Enc\left(\begin{bmatrix}c&f&i&m&p\\a&d&g&k&n\\b&e&h&l&o\end{bmatrix}\right)\\
&=\\
&Enc\left(\begin{bmatrix}1a+2b+3c&1d+2e+3f&1g+2h+3i&1k+2m+3l&1n+2o+3p\\4a+5b+6c&4d+5e+6f&4g+5h+6i&4k+5m+6l&4n+5o+6p\\0&0&0&0&0\end{bmatrix}\right)
\end{aligned}
\qquad 3\text{-}(12)
$$

如下算法 2 总结情况 1 的安全矩阵乘运算

Algorithm 2: SecMatMult1($ct_A, ct_B, p_0, p_1, p_2, m_0, m_1$)

Input: ct_A, ct_B: 矩阵 $A_{p_0\times p_1}, B_{p_1\times p_2}$ 的密文

Input: p_0, p_1, p_2: 矩阵 $A_{p_0\times p_1}, B_{p_1\times p_2}$ 的维度

Input: m_0, m_1 超立方结构的维度

Output: ct_C: $C_{p_0\times p_2}=A_{p_0\times p_1}\times B_{p_1\times p_2}$ 的密文

1: **for** $i=0$ to p_1-1 **do**

2: $ct_{d_i}=U_i\otimes ct_A$;

3: $ct_{A_i}=\text{Replicate1D}(ct_{d_i},1,p_0,p_1,m_0,m_1)$;

4: **end**

5: **for** $i=0$ to p_1-1 **do**

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

6: $ct_{B_i} = Rotate1D(ct_B, 0, -i)$;

7: $ct_C = ct_C \oplus (ct_{A_i} \otimes ct_{B_i})$;

8: **end**

9: return ct_C.

3.3.2 同态矩阵乘情况二

假设有两个矩阵 $A_{m\times l}$ 和 $B_{l\times n}$ 其中 $m \le l$。假设 k 是 l 的最小倍数且大于等于 m 的值。假设 $t = \max(l, n)$。事先设置一个 $m_0 \times m_1$ 的超立方结构，其中 $m_0 = k$ 且 $m_1 \ge t$。同样的，为了方便起见，以下假设 $m_0 = k$ 且 $m_1 = t$，易知本文的方法在 $m_0 = k$ 且 $m_1 \ge t$ 的情况下同样适用。首先把矩阵 $A_{m\times l}$ 和 $B_{l\times n}$ 分别加密为 $ct_{A_{m\times l}}$ 和 $ct_{B_{l\times n}}$。加密过程是在客户端进行的。下面的例子介绍了输入矩阵具体是如何加密的。

给定两个矩阵 $A_{3\times 2}$ 和 $B_{2\times 5}$ 如下：

$$A_{3\times 2} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}, \quad B_{2\times 5} = \begin{bmatrix} a & c & e & g & i \\ b & d & f & h & j \end{bmatrix} \tag{3-(13)}$$

设置一个 3×5 的二维超立方结构足够容纳 $A_{3\times 2}$ 和 $B_{2\times 5}$。将 $A_{3\times 2}$ 和 $B_{2\times 5}$ 分别加密为如下形式。

$$ct_{A_{3\times 2}} = Enc\left(\begin{bmatrix} 1 & 2 & 0 & 0 & 0 \\ 3 & 4 & 0 & 0 & 0 \\ 5 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right) \tag{3-(14)}$$

$$ct_{B_{2\times 5}} = Enc\left(\begin{bmatrix} a & c & e & g & i \\ b & d & f & h & j \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}\right) \tag{3-(15)}$$

以下步骤 b1 到步骤 b4 介绍同态矩阵乘，该运算在服务器端。

步骤 b1： $A_{k\times l}$ 表示 $m_0 \times m_1$ 的超立方结构最左边的 $k \times l$ 的矩阵。然而，$A_{k\times l}$ 并非平方矩阵，

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

非方阵的对角线元素概念未被定义。本文采用如下概念定义其对角线，对于矩阵 $A_{k\times l}$，$d_i=(A_{0,i},A_{1,i+1},\cdots,A_{k-1,i-1})$ 其中 $0\le i<l$ 且序号在模数 l 之下。本步骤利用乘掩码的操作提取矩阵 $A_{k\times l}$ 的对角线元素如下：

$$ct_{d_i}=V_i\otimes ct_{A_{m\times l}} \tag{3-(16)}$$

其中 $0\le i<l$ 且 $V_i[i][j](0\le I<k,0\le J<t)$ 定义如下：

$$V_i[I][J]=\begin{cases}1 & If\ J=i+I\ (\mathrm{mod}\ l)\\ 0 & else\end{cases} \tag{3-(17)}$$

以下例 b1 详细介绍了如何提取对角线元素。

例 b1：给定矩阵 $A_{4\times 2}$ 为 4×5 超立方结构最左边的非方阵。对角线元素分别为 $d_0=(1,4,5,0)$，$d_1=(2,3,6,0)$。

$$A_{4\times 5}=\left[\begin{array}{cc|ccc}1&2&0&0&0\\3&4&0&0&0\\5&6&0&0&0\\0&0&0&0&0\end{array}\right] \tag{3-(18)}$$

给定公式 3-(14)中的一个密文 $ct_{A_{3\times 2}}$，本步骤乘掩码得到的两个密文 ct_{d_0} 和 ct_{d_1} 如下：

$$ct_{d_0}=Enc\left(\begin{bmatrix}1&0&0&0&0\\0&4&0&0&0\\5&0&0&0&0\\0&0&0&0&0\end{bmatrix}\right),$$

$$ct_{d_1}=Enc\left(\begin{bmatrix}0&2&0&0&0\\3&0&0&0&0\\0&6&0&0&0\\0&0&0&0&0\end{bmatrix}\right) \tag{3-(19)}$$

步骤 b2：本步骤利用 Replicate 操作将 $ct_{d_i}(0\le i<l)$ 的每行复制，分别得到密文 $ct_{A_i}(0\le i<l)$。具体见算法 1。

$$ct_{A_i}=\mathrm{Replicate1D}(ct_{d_i},1,m,l,m_0,m_1) \tag{3-(20)}$$

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

如下例 b2 表示了一个具体的例子。

例 b2：对于公式 3-(19)中给定的密文ct_{d_0}和ct_{d_1}，经过本步骤得到的两个密文如下ct_{A_0}和ct_{A_1}如下。

$$ct_{A_0}=Enc\left(\begin{bmatrix}1&1&1&1&1\\4&4&4&4&4\\5&5&5&5&5\\0&0&0&0&0\end{bmatrix}\right),$$

$$ct_{A_1}=Enc\left(\begin{bmatrix}2&2&2&2&2\\3&3&3&3&3\\6&6&6&6&6\\0&0&0&0&0\end{bmatrix}\right) \quad 3\text{-}(21)$$

步骤 b3：本步骤将密文$ct_{B_{l\times n}}$沿着垂直方向按列复制k/l次得到密文$ct_{\overline{B}}$。本步骤是情况 1 所没有的操作。因为大部分情况下$ct_{B_{l\times n}}$不只有一行非零元素，故本文设计了与算法 1 略有不同的算法 3 来实现本操作，算法 3 可以看作是算法 1 的扩展，可以实现按行和按列复制操作。

$$ct_{\overline{B}}=\text{eReplicate1D}(ct_B,0,l,n,m_0,m_1) \quad 3\text{-}(22)$$

以下例 b3 是一个具体的实例。

例 b3：对于公式 3-(15)给定的密文$ct_{B_{2\times5}}$，经过本步骤后得到的密文$ct_{\overline{B}}$如下：

$$ct_{\overline{B}}=Enc\left(\begin{bmatrix}a&c&e&g&i\\b&d&f&h&j\\a&c&e&g&i\\b&d&f&h&j\end{bmatrix}\right) \quad 3\text{-}(23)$$

Algorithm 3: eReplicate1D(ct, i, p_0, p_1, m_0, m_1)

Input: ct： $A_{p_0\times p_1}$的密文

Input: i:第i维超立方结构

Input: p_0, p_1:矩阵$A_{p_0\times p_1}$维度

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

Input: m_0, m_1:超立方结构的维度

Output: $\overline{ct}$

1: $\overline{ct} = ct, e = 1$;

2: **if** $i = 0$ **then**

3: $f = p_0$;

4: $T = m_0 / p_0$;

5: **else**

6: $f = p_1$;

7: $T = m_1 / p_1$;

8: **end**

9: **for** $j = numBit(T) - 2$ to 0 **do**

10: $\overline{ct} = \overline{ct} \oplus Rotate1D(\overline{ct}, i, e \times f)$;

11: $e = 2 \cdot e$;

12: **if** $bit_j(T) = 1$ **then**

13: $\overline{ct} = ct \oplus Rotate1D(\overline{ct}, i, f)$;

14: $e = e + 1$;

15: **end**

16: **end**

17: return $\overline{ct}$.

步骤 b4：本步骤将密文$ct_{\overline{B}}$按列旋转i位分别得到l个密文$ct_{B_i}(i = 0,1,...,l-1)$。

$$ct_{B_i} = Rotate1D(ct_{\overline{B}}, 0, -i) \tag{3-(24)}$$

以下例 b4 是一个具体的实例。

例 b4：根据公式 3-(23)中给定的密文$ct_{\overline{B}}$，通过旋转操作得到密文ct_{B_0}和ct_{B_1}。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$ct_{B_0}=Enc\left(\begin{bmatrix}a&c&e&g&i\\b&d&f&h&j\\a&c&e&g&i\\b&d&f&h&j\end{bmatrix}\right),$$

$$ct_{B_1}=Enc\left(\begin{bmatrix}b&d&f&h&j\\a&c&e&g&i\\b&d&f&h&j\\a&c&e&g&i\end{bmatrix}\right) \quad 3\text{-}(25)$$

步骤 b5：本步骤计算密文$ct_{C_{m\times n}}=\sum_{i=0}^{l-1}ct_{A_i}\otimes ct_{B_i}$，具体如例 b5：

例 b5：对于公式 3-(21)、3-(25)中的密文ct_{A_i}和ct_{B_i}，本步骤计算$ct_{C_{3\times5}}=\sum_{i=0}^{2}ct_{A_i}\otimes ct_{B_i}$，其为$C_{3\times5}=A_{3\times2}\times B_{2\times5}$前三行的加密。

$$\begin{aligned}&Enc\left(\begin{bmatrix}1&1&1&1&1\\4&4&4&4&4\\5&5&5&5&5\\0&0&0&0&0\end{bmatrix}\right)\otimes Enc\left(\begin{bmatrix}a&c&e&g&i\\b&d&f&h&j\\a&c&e&g&i\\b&d&f&h&j\end{bmatrix}\right)\\&\oplus\\&Enc\left(\begin{bmatrix}2&2&2&2&2\\3&3&3&3&3\\6&6&6&6&6\\0&0&0&0&0\end{bmatrix}\right)\otimes Enc\left(\begin{bmatrix}b&d&f&h&j\\a&c&e&g&i\\b&d&f&h&j\\a&c&e&g&i\end{bmatrix}\right)\\&=\\&Enc\left(\begin{bmatrix}1a+2b&1c+2d&1e+2f&1g+2h&1i+2j\\3a+4b&3c+4d&3e+4f&3g+4h&3i+4j\\5a+6b&5c+6d&5e+6f&5g+6h&5i+6j\\0&0&0&0&0\end{bmatrix}\right)\end{aligned} \quad 3\text{-}(26)$$

如下算法 4 总结情况 2 的安全矩阵乘运算

Algorithm 4: SecMatMult2($ct_A, ct_B, p_0, p_1, p_2, m_0, m_1$)

Input: ct_A, ct_B: 矩阵$A_{p_0\times p_1}, B_{p_1\times p_2}$的密文

Input: p_0, p_1, p_2: 矩阵$A_{p_0\times p_1}, B_{p_1\times p_2}$的维度

Input: m_0, m_1超立方结构的维度

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

Output: ct_C： $C_{p_0\times p_2}=A_{p_0\times p_1}\times B_{p_1\times p_2}$ 的密文

1: **for** $i=0$ to p_1-1 **do**

2: $ct_{d_i}=V_i\otimes ct_A$;

3: $ct_{A_i}=\text{Replicate1D}(ct_{d_i},1,p_0,p_1,m_0,m_1)$;

4: **end**

5: $ct_{\bar{B}}=\text{eReplicate1D}(ct_B,0,p_1,p_2,m_0,m_1)$;

6: **for** $i=0$ to p_1-1 **do**

7: $ct_{B_i}=Rotate1D(ct_{\bar{B}},0,-i)$;

8: $ct_C=ct_C\oplus(ct_{A_i}\otimes ct_{B_i})$;

9: **end**

10: return ct_C.

3.4 实验结果与安全性

本小节进行安全性分析以及实验结果对比。本文的算法实现基于支持超立方结构的全同态加密，例如 BGV 方案，MHEAAN 方案。本章的实验基于 HElib 库，其支持 BGV 方案及其变体。本章实验运行在 Intel(R) Xeon(R) Gold 6148 CPU 2.40 GHz 503G RAM.

3.4.1 时间复杂度分析

本小节分析算法的时间复杂度。下面介绍表格中符号的含义，Add 和 Mult 分别表示密文加和密文乘。Cmult 表示明文与密文之间的乘法。Rot 表示密文沿行或沿列旋转操作，Depth 表示乘法电路的深度。M 和 C 分别表示明文密文乘和密文密文乘的电路深度。假设有两个矩阵 $A_{m\times l}$ 和 $B_{l\times n}$。其中 k 是 l 的最小倍数且大于等于 m 的值，$t=\max(l,n)$。表 3.1 和表 3.2 分别总结了算法 2、算法 4 每一步的时间复杂度和电路深度。与传统的方法相比，本文的算法无需将非方阵扩充为方阵，故本算法有更好的性能，表 3.3 为时间复杂度比较。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

表 3.1 算法 2 的时间复杂度及电路深度

Step	Add	Cmult	Mult	Rot	Depth
a1	-	l	-	-	1C
a2	$l\cdot\log t$	-	-	$l\cdot\log t$	-
a3	-	-	-	l	-
a4	l	-	l	-	1M
Total	$l\cdot\log t+l$	l	l	$l\cdot\log t+l$	1M+1C

表 3.2 算法 4 的时间复杂度及电路深度

Step	Add	Cmult	Mult	Rot	Depth
b1	-	l	-	-	1C
b2	$l\cdot\log t$	-	-	$l\cdot\log t$	-
b3	$\log(k/l)$	-	-	$\log(k/l)$	-
b4	-	-	-	l	-
b5	l	-	l	-	1M
Total	$l\cdot\log t+\log(k/l)+l$	l	l	$l\cdot\log t+\log(k/l)+l$	1M+1C

表 3.3 时间复杂度比较

Step	Add	Cmult	Mult	Rot	Depth
Native method	$k\cdot\log k+k$	k	k	$k\cdot\log k+k$	1M+1C
Ours(case a)	$l\cdot\log t$	l	l	$l\cdot\log t$	1M+1C
Ours(case b)	$l\cdot\log t+\log(k/l)+l$	l	l	$l\cdot\log t+\log(k/l)+l$	1M+1C

3.4.2 实验效果比较

本小节就不同维度的矩阵乘与传统方法做实验效果比较。首先选择 BGV 方案的参数，κ 表示安全级别，L 表示密文空间的模数，p 和 m 分别表示明文空间的模数和循环多项式的度数。

对于基于超立方结构的安全矩阵乘，本文依照 Rathee 等[11]的工作选择最优的参数，具体来说按照如下的规则选择参数 p 和 m 。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$m = h \cdot m_0 \cdot m_1 + 1 \quad \gcd(h, m_0) = \gcd(h, m_1) = \gcd(m_0, m_1) = 1 \quad ord(p) = h$

之后在 Z_m^* 中根据 m_0 和 m_1 选择参数 g_0 和 g_1，根据[11]中所述，可依据 m_0 和 m_1 的大小选择一个二维的超立方结构 $Z_m^* / < p >=< g_0, g_1 >$。选择一个足够大的 L 值以保证算法的正确性。所有的参数都保证至少 $\kappa = 80bit$ 的安全级别。

从表 3.4 可以看出本文的方法与传统的方法相比表现出更好的性能。例如非方阵乘 $A_{1\times5} \times B_{5\times128}$，传统的方法需要 7.80s，而本算法只需要 0.069s。

表 3.4 算法运行时间(s)比较

Dims	Algorithms	p	m	L	m_0	m_1	Enc	Dec	MatMult
$A_{64\times1} \times B_{1\times64}$	Naive method	8567417	7489	7	64	117	0.0215	0.667	1.168
	Ours	8567417	7489	7	64	117	0.0204	0.636	0.0202
$A_{5\times5} \times B_{5\times128}$	Naive method	8555227	18049	7	128	141	0.0874	3.935	6.4601
	Ours	8477303	7481	7	5	136	0.0199	0.114	0.069
$A_{1\times5} \times B_{5\times128}$	Naive method	8555227	18049	7	128	141	0.0399	4.91	7.80
	Ours	8477303	7481	7	5	136	0.0177	0.113	0.069
$A_{5\times10} \times B_{10\times64}$	Naive method	8567417	7489	7	64	117	0.052	0.6608	1.425
	Ours	8471819	7591	7	10	69	0.0382	0.123	0.196
$A_{64\times64} \times B_{64\times64}$	Naive method Ours	8567417	7489	7	64	117	0.0464	0.661	1.297
$A_{32\times32} \times B_{32\times32}$	Naive method Ours	8429731	7393	7	32	33	0.034	0.131	0.515

3.4.3 与其他相关工作的比较

表 3.5 列出了所提出的算法和现有的安全矩阵乘法方案的比较总结。Arbitrary 表示该方案是否支持任意维度的矩阵乘。Composability(可组合性)表示是否可以使用两个矩阵的乘积与下一个矩阵相乘，该特性在多矩阵乘法等应用中起着重要的作用。#Ciphertexts 表示加密一个矩阵的密文数。Security 表示该方案在 CPA 模型中被证明是安全的。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

表 3.5 与现有工作的比较

Methods Properties	Technique	Arbitrary	#Ciphertexts	Composability	Security
Jiang et al.[13]	FHE+SIMD	N	1	Y	CPA
Rathee et al.[11]	FHE+hypercube	N	1	Y	CPA
Hiromasa et al.[9]	FHE	N	1	Y	CPA
Cheon et al.[12]	FHE+hypercube	N	1	Y	CPA
Lu et al.[35]	FHE+SIMD	Y	n	Y	CPA
Wang et al.[36]	FHE+SIMD	Y	n	Y	CPA
Duong et al.[10]	FHE	N	1	N	CPA
Atallah et al.[15]	Disguise-based	N	-	N	non-CPA
Lei et al.[17]	Disguise-based	Y	-	N	non-CPA
Fu et al.[21]	Disguise-based	N	-	N	non-CPA
Zhao et al.[24]	Disguise-based	N	-	N	CPA
Zhang et al.[23]	Disguise-based	Y	-	N	non-CPA
Ours	FHE+hypercube	Y	1	Y	CPA

总的来说本文算法的特点是只需一个密文表示一个矩阵、算法具有可组合性、适用于非方阵乘，这是其他方案所不能同时具备的。

3.4.4 安全性分析

如在 3.2 小节中所描述的，本文的安全系统基于一个半诚实的模型并且威胁只来自于服务器。在选择明文攻击(IND-CPA)模型[38]下，底层 BGV 方案[28]已被证明是安全的。这意味着两个密文信息在运算中是不可被分辨的，也就是说密文的信息在服务器端不会被泄露；而在数据传输阶段，密文传输也不会造成信息泄露。因此本文提出的安全矩阵乘法方案在 IND-CPA 模型下是安全的。

3.5 安全多矩阵连乘中的应用

本小节将之前提出的安全矩阵乘法方案应用到多矩阵连乘中。与安全矩阵乘法相比，研究安全多矩阵乘法的工作较少。Mishra 等[39]首先研究了安全多矩阵乘法。他们的方案针

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

对不同的矩阵应用了不同的打包技术，且每个矩阵相较前一个都需要更大的参数。然而更大的参数会导致同态操作变得更慢。因此他们的方法随着矩阵数量的增多会变得不实用。

最近的工作中，Wang 等[36]提出了一种更有效的安全矩阵连乘方案。他们的方法基于 Halevi 等[34]的列打包技术，研究了矩阵向量乘和矩阵矩阵乘。他们的方法同样是具有可合成性的。

以下将安全矩阵乘法算法应用在非平方矩阵连乘上。表 3.6 为本文算法与 Wang[36]的方法在 10 个非方阵连乘时间的比较结果数据。为了实验公平两个实验同样选取了[36]中的矩阵维度 $p_0 = 40, p_1 = 30, ..., p_{10} = 25$，由表中可以看出，本文方法只需 56s，而她的方法需要 9710s。

表 3.6 10 个非方阵连乘的时间(s)结果比较

#Matrices	40,30,35,15,60,5,70,10,50,20,25		
Methods	Enc	MMatMult	Dec
Wang et al.[36]	8.36	9710.65	3.3
Ours	0.374	56.831	1.827

3.6 本章小结

本章基于超立方结构提出了一个安全高效的非平方矩阵连乘方案，形式为 $A_{m\times l} \times B_{l\times n}$。上述算法是第一种基于全同态加密的任意矩阵的安全矩阵乘法方案。该方案只需要一个简单的密文来表示一个矩阵且需 l 次同态乘法即可实现安全矩阵乘。实验结果表明，它对不同维数的矩阵都具有良好的性能。之后本算法被应用到安全矩阵连乘中。实验结果表明，该解决方案明显优于现有的最先进的安全多矩阵乘法方案。

由于矩阵乘法是科学计算中最基本的操作之一，本文提出的安全矩阵乘法有望在使用矩阵乘法作为构建块的高级安全外包计算方案中找到广泛的应用。超立方体结构将矩阵加密为单个密文，并支持沿行或列的灵活数据移动，从而为矩阵相关操作的安全外包计算提供了一个很有前途的工具。未来的工作是研究基于超立方体结构的其他与矩阵相关的操作，例如安全矩阵行列式、安全矩阵分解。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

第四章 基于支持向量机的加密图像分类

随着云服务的普及，许多机器学习算法可以外包给云平台。这就是所谓的“机器学习及服务”(MLaaS)，类似的例子有：谷歌的 Prediction API[40]、亚马逊的 AWS[41]和微软的 Azure[42]等等。然而，用户上传到云端的数据对服务器端透明，毫无隐私可言。数据隐私问题仍然是云计算被广泛使用的主要障碍之一。支持向量机是一种机器学习算法，为了保护用户隐私，本章基于安全矩阵运算和全同态加密，提出了一种保护隐私的支持向量机分类预测方法。该方法基于单指令多数据技术(SIMD)且与最新的工作相比显著地降低了通信开销，提高了计算性能。

4.1 引言

目前已经有许多工作研究支持向量机加密图像分类，他们的方法大部分用到了多方计算。多方计算更高效但通信开销大，且无法避免共谋攻击。另外有其他的自定义方法虽然性能很好，但未达到密码安全的标准。

Rahulamathavan 等[43]基于加同态加密提出了保护隐私的支持向量机分类算法。然而，由于 AHE 的局限性，Rahulamathavan 等人的方案需要客户端和服务器之间进行多轮交互来计算多项式核函数，该核函数也是支持向量机分类的重要步骤。

Barnett 等[44]首次提出了基于全同态加密的非交互式隐私保护 SVM 分类算法。但是他们的工作有几点缺陷。首先，他们把每一个特征向量的元素单个加密，这需要大量的密文表示向量，导致计算开销大。其次，服务器必须执行大量同态乘法操作，从而导致较高的计算开销。

其他的方法用到了安全多方计算[25][45]。Rizai[46]等人提出了一个名为 Chameleon 的保护隐私的支持向量机分类算法，他们的方法基于混淆电路(Garbled circuits)、GMW 和加法秘密共享(A-SS)。他们的方法需要客户端和服务器之间的多轮交互。相比之下，除了上传和结果返回，本文基于全同态加密的方法是非交互的。

Marki 等人[47]在假设存在多个服务器的前提下提出了另一种基于 MPC 的隐私保护 SVM 分类算法。尽管多服务器模型不需要客户端和服务器之间的交互，但是这极易遭受共谋攻击。相较而言，本文的方法工作在单一服务器模型中，从而避免了共谋攻击。

其他的工作是关于支持向量机训练的。Yu 等[48]和 Laur 等[49]分别提出了垂直和水平分

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

区训练数据集的隐私保护 SVM 训练算法。Vaidya 等[50]提出了一种针对任意分区数据集的隐私保护 SVM 训练算法。他们的方法假设训练数据集被分为两部分，双方都希望在保护隐私的前提下共同协作完成训练。

以下是一些在云环境背景下关于保护隐私的支持向量机分类的相关工作。Lin 等[51]基于随机线性变换技术提出了 SVM 训练算法。尽管他们的随机线性变换方法是有效的，但是他们的方案并未达到密码安全的标准，易受到攻击(IND-CPA[38])。Liu 等[52]基于分布式公钥系统[53]提出了 SVM 训练方法。但他们的方法也需要两个服务器协作，如此易受共谋攻击。

本章提出的加密图像分类方案主要贡献如下：

1.同态计算 $b+\sum_{i=1}^{n} a_i \cdot y_i \cdot (1+\boldsymbol{x_i} \cdot \mathbf{z})^d$。基于 SIMD 技术将整个特征向量加密成一个单独的密文，降低了算法空间复杂度；充分利用 SIMD 技术支持并行计算的特性，显著地提高了计算性能。

2.同态计算非线性函数 *sign*。应用同态近似比较方法处理分类最后的非线性计算。先将输入值映射到指定范围，然后同态地计算结果。

4.2 相关背景知识介绍

4.2.1 系统与安全模型

如图 4.1 所示，本文的支持向量机分类系统模型包含两个部分。其中客户端方持有图像 M，服务器端持有已训练好的支持向量机模型并提供分类服务。假设客户端已经生成了一对公钥 *pk* 和私钥 *sk*。每次客户端打算将图像的 SVM 分类外包给云服务器，便可用公钥 *pk* 将图像特征加密并上传密文 C 到服务器端。服务器接到请求后对密文执行同态的支持向量机算法，并将计算结果 $C^* = Enc(0/1)$ 返回给客户端。最后，客户端用私钥 *sk* 解密 C^* 并得到分类结果 0/1。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

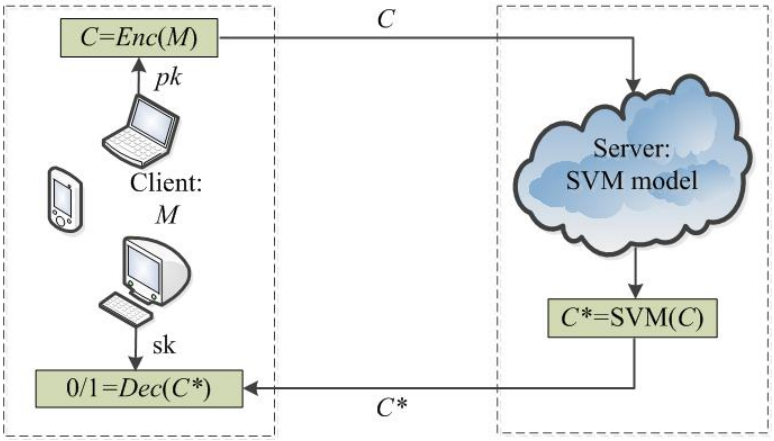

图 4.1 加密图像分类系统模型

4.2.2 支持向量机分类

支持向量机[54]是机器学习中的一个被广泛使用的分类方法，用于小样本分类效果良好。给定 n 个样本 $(\boldsymbol{x}_i, y)$，其中 $\boldsymbol{x}_i$ 为 t 维的特征向量，$y_i \in \{-1, 1\}$ 为分类的标签。支持向量机训练算法试图找到一个最佳的超平面，将训练样本划分为两个类，如图 4.2 所示。而对于线性不可分的数据集，一般的思想是先利用核函数将原始样本空间映射到训练集线性可分的高维空间，之后再进行分类。本文中采用的是多项式核函数，保证了较低的计算复杂

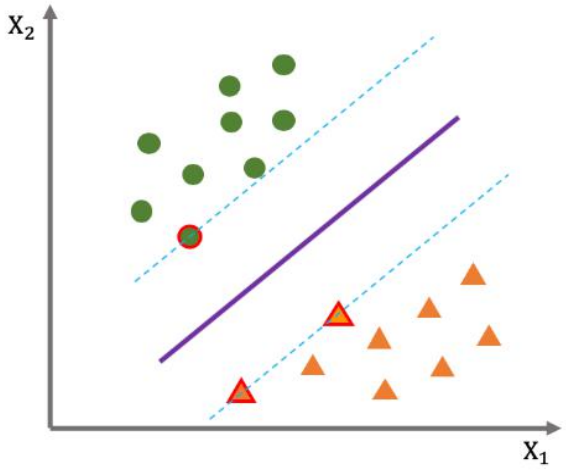

图 4.2 支持向量机分类超平面

度和较高的分类精确率。以下是一个典型的支持向量机多项式核函数分类器。

$$Class(z) = sign[b + \sum_{i=1}^{n} a_i \cdot y_i \cdot (1 + \boldsymbol{x}_i \cdot \mathbf{z})^d] \qquad 4\text{-}(1)$$

其中 n 代表支持向量机的个数，$\boldsymbol{x}_i$ 为支持向量，b 是模型的截距，a_i 是 SVM 分类优化问

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

题的拉格朗日乘子，y_i 是支持向量的标签，**z** 代表输入图像的 t 维特征向量，d 是多项式核函数的度数。符号函数 $sign[x]$ 的定义为：$x \geq 0$ 时 $sign[x]=1$；否则 $sign[x]=-1$。

4.2.3 HOG 特征提取技术

HOG 特征提取[55]是一种在图像处理中被用来提取物体识别特征的技术。本文所用到初始图像大小为 32×32 像素。虽然 32×32 像素的图像已经是一个很小的尺寸了，但即使是这样的小图像也能提供良好的分类结果，在实践中，更大的图像也会被压缩到这个尺寸，以提高运行时间。此外，从根本上影响性能的是提取的特征的数量和质量，而这不直接取决于图像大小。该图像被分为若干个名为 cell 的矩形区域，对于像素为 32×32 的图像，设置一个 cell 的大小为 8×8，如此一整个图像共有 16 个 cell。

为每个 cell 中的像素创建梯度直方图。像素处的梯度是对图像中该像素的强度变化的度量，并且具有大小（强度变化的水平）和方向（强度变化最大的方向）。梯度在检测图像中的线条和边缘方面是有效的，例如，垂直线将导致水平方向的大强度变化。cell 的梯度可以排序为直方图，其中每个直方图中的 bin 是梯度方向的范围，这些值总结了方向在各自 bin 中的梯度的大小。一个经典的方法是每个直方图使用 9 个 bin，因此 HOG 特征的总数（大致）是适合图像大小的 cell 数目的 9 倍。如此共有 16 个 cell，其中每个 cell 有 9 个 bin，最终得到了 144 个 HOG 特征。

4.2.4 PCA 数据降维

PCA 是一种被用在机器学习的标准降维技术[56]。它是一种将一组特征转化为一组较小的线性不相关特征的降维技术，降维后的特征被称为主成分，故该方法被称为主成分分析法。没有确定的规则来选择主成分的数量，因为它们是数据依赖的，但通常的方法是要么占总方差的最小百分比，要么删除贡献小于总方差或最大方差的给定百分比的成分。PCA 经常与 HOG 一起使用，以进一步降低特征向量的维数。PCA 降维的数量取决于将要降维的问题的实际情况。在本文所要处理的问题中，矩阵 A 由训练样本所决定，则 PCA 降维的公式为 $z = A^T \cdot (h - m_h)$，其中 h 是待降维的特征向量，m_h 是训练数据的中值向量。在本文中，假设服务器端提供 m_h 和 A^T 给客户端，然后在客户端完成 PCA 的降维，最后客户端

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

加密上传待分类图像。

4.3 基于支持向量机的加密图像分类

SVM 分类的对象是代表原始图像的特征向量而不是图像本身。第一步用 HOG 特征提取技术处理待分类图像，得到原图像的特征向量。然后根据需要对特征向量进行 PCA 降维。本文假设特征向量已经被客户端预先提取完成，并把重点放在保护隐私的分类算法上。

假设服务器端和客户端分别持有 SVM 模型和 t 维的特征向量 $\mathbf{z}=(z_1,\cdots,z_t)$。为了能更安全的进行分类预测，需要将待分类支持向量加密上传。

一个传统的安全分类方法例如 Barnett 等[44]和 Rahulamathavan 等[43]，他们的方法是客户端将特征向量的每一个元素分别加密，然后服务器端通过公式 4-(1)安全的计算分类结果。该方法正如前文中提到的，需要 t 个密文来表示一个图像并导致在服务器端的运算量加大。

本文采用 SIMD 技术将一整个向量加密成为一个密文，密文的数量由此从 t 降低到 1。另外，SIMD 技术还支持并行运算，如此显著地提高了运算的效率。SIMD 技术是基于 RLWE 问题的全同态加密方案最具代表性的特征。尽管 SIMD 技术已经在其他应用中被广泛使用，但本文第一个将该技术应用于保护隐私支持向量机图像分类中。支持向量机分类算法包含 n 对向量之间的内积计算、求 e 的指数和求和计算。因此将 SIMD 技术应用于此类较复杂计算问题中并不容易。

接下来的部分本文基于 SIMD 技术介绍一个高效的加密数据 SVM 分类算法，该算法的特点是只需一个密文即可表示图像且运算效率更高效。

4.3.1 特征向量的加密

客户端将待分类的特征向量（已将图像特征提取）$\mathbf{z}=(z_1,z_2,\cdots,z_t)$ 复制 n 次充满整个密文槽，如有余下的槽，一律补零，其中 n 代表支持向量的个数。接下来客户端把明文槽加密成一个密文并上传到客户端。如公式 4-(2)所示，假设 t 和 n 为 2 的幂，如果特征向量个数不足的话，客户端将不足的部分补零以便进行接下来的操作。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$ct_1 = Enc\begin{bmatrix} z_1 & \cdots & z_t \\ \vdots & \vdots & \vdots \\ z_1 & \cdots & z_t \end{bmatrix} \tag{4-(2)}$$

4.3.2 支持向量机分类的同态运算

云端持有 SVM 模型包含参数$(\boldsymbol{x_i}, b, a_i, y_i)$。在接下来的步骤中服务器端根据公式 4-(1)对输入的密文ct_1执行 SVM 同态运算。

步骤一：服务器端根据模型参数初始化三个明文矩阵E, X, B。按如下方式打包：

$$E = \begin{bmatrix} e_1 & \cdots & e_1 \\ \vdots & \vdots & \vdots \\ e_t & \cdots & e_t \end{bmatrix} \tag{4-(3)}$$

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,t} \\ \vdots & \vdots & \vdots \\ x_{n,1} & \cdots & x_{n,t} \end{bmatrix} \tag{4-(4)}$$

$$B = \begin{bmatrix} b & \cdots & b \\ \vdots & \vdots & \vdots \\ b & \cdots & b \end{bmatrix} \tag{4-(5)}$$

其中$e_i = a_i \cdot y_i$，$\boldsymbol{x_i} = (x_{i,1}, \cdots, x_{i,t})$。参数$e_i = a_i \cdot y_i$中的每一个元素都充满一整行，如果实际的数量不足，一律补零。$\boldsymbol{x_i}$中的每个元素按序充满整个槽，如果实际数量不如，一律补零。截距b复制$n \times t$次充满整个槽空间。

步骤二：给定明文X和密文ct_1，本步骤计算两者的乘积。

$$ct_2 = Enc\begin{bmatrix} x_{1,1} \cdot z_1 & x_{1,2} \cdot z_2 & \cdots & x_{1,t} \cdot z_t \\ x_{2,1} \cdot z_1 & x_{2,2} \cdot z_2 & \cdots & x_{2,t} \cdot z_t \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} \cdot z_1 & x_{n,2} \cdot z_2 & \cdots & x_{n,t} \cdot z_t \end{bmatrix} \tag{4-(6)}$$

应用 SIMD 技术，在实际的操作中，只需将两个密文相乘（点对点乘），两个位置对应的槽空间相乘，即可得到密文ct_2。

步骤三：通过旋转(Rotate)操作和加法(Add)操作，服务器计算一整行$x_{ij} \cdot z_j$的和，也即两向

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

量 $\boldsymbol{x}_i \cdot \mathbf{z}$ 的内积，如下：

$$ct_2 \leftarrow ct_2 \oplus Rotate(ct_2, 2^j) \tag{4-(7)}$$

其中 $j = 0,1,\cdots,\log_2 t - 1$。迭代计算后内积的计算结果在每一行的第一列中，其它槽中的结果为无效值，统一用*表示。

$$ct_3 = Enc\begin{bmatrix} \boldsymbol{x}_1 \cdot \mathbf{z} & \cdots & * \\ \vdots & \vdots & * \\ \boldsymbol{x}_n \cdot \mathbf{z} & \cdots & * \end{bmatrix} \tag{4-(8)}$$

如例 a 所示，运用边旋转边求和的方法，可以用较小的运算量计算每一整行的和。

例 a：给定一个 4×4 的密文矩阵 ct_A，以下演示旋转加的过程。

$$ct_A = Enc\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix} \tag{4-(9)}$$

步骤 a1：密文内部依次旋转一位得到 ct_{A_1}。

$$ct_{A_1} = Enc\begin{bmatrix} 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 3 \\ 3 & 3 & 3 & 4 \\ 4 & 4 & 4 & 1 \end{bmatrix} \leftarrow Enc\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix} \tag{4-(10)}$$

步骤 a2：旋转后密文与原密文矩阵相加，$ct_{A_2} = ct_A + ct_{A_1}$。

$$ct_{A_2} = Enc\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix} + Enc\begin{bmatrix} 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 3 \\ 3 & 3 & 3 & 4 \\ 4 & 4 & 4 & 1 \end{bmatrix} = Enc\begin{bmatrix} 2 & 2 & 2 & 3 \\ 4 & 4 & 4 & 5 \\ 6 & 6 & 6 & 7 \\ 8 & 8 & 8 & 5 \end{bmatrix} \tag{4-(11)}$$

步骤 a3：密文内部依次旋转两位。

$$ct_{A_3} = Enc\begin{bmatrix} 2 & 3 & 4 & 4 \\ 4 & 5 & 6 & 6 \\ 6 & 7 & 8 & 8 \\ 8 & 5 & 2 & 2 \end{bmatrix} \leftarrow Enc\begin{bmatrix} 2 & 2 & 2 & 3 \\ 4 & 4 & 4 & 5 \\ 6 & 6 & 6 & 7 \\ 8 & 8 & 8 & 5 \end{bmatrix} \tag{4-(12)}$$

步骤 a4：旋转后密文与旋转前密文相加，$ct_{A_4} = ct_{A_2} + ct_{A_3}$。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$ct_{A_4} = Enc\begin{bmatrix}2&2&2&3\\4&4&4&5\\6&6&6&7\\8&8&8&5\end{bmatrix} + Enc\begin{bmatrix}2&3&4&4\\4&5&6&6\\6&7&8&8\\8&5&2&2\end{bmatrix} = Enc\begin{bmatrix}4&*&*&*\\8&*&*&*\\12&*&*&*\\16&*&*&*\end{bmatrix} \quad 4\text{-}(13)$$

如上所示，经过两次旋转两次加法之后，在矩阵的每一行的第一个元素即原矩阵每行的和。这样计算的好处有两点：1.时间复杂度小。2.空间复杂度小。但是要注意的是行列的数量应为 2 的幂，不足应补零。

步骤四：本步骤的目的是得到$(1+\boldsymbol{x_i}\cdot\mathbf{z})^d$的值，服务器端先生成一个全 1 的明文矩阵$I$，如下。

$$I = \begin{bmatrix}1&\cdots&1\\\vdots&\vdots&\vdots\\1&\cdots&1\end{bmatrix} \quad 4\text{-}(14)$$

将I与ct_3相加，然后求指数幂。结果如下：

$$ct_4 = Enc\begin{bmatrix}(1+\boldsymbol{x}_1\cdot\mathbf{z})^d&\cdots&*\\\vdots&\vdots&*\\(1+\boldsymbol{x_n}\cdot\mathbf{z})^d&\cdots&*\end{bmatrix} \quad 4\text{-}(15)$$

步骤五：服务器端计算$ct_5 = E\otimes ct_4$，如下：

$$ct_5 = Enc\begin{bmatrix}e_1\cdot(1+\boldsymbol{x}_1\cdot\mathbf{z})^d&\cdots&*\\\vdots&\vdots&*\\e_n\cdot(1+\boldsymbol{x_n}\cdot\mathbf{z})^d&\cdots&*\end{bmatrix} \quad 4\text{-}(16)$$

步骤六：本步骤的目的是为了得到$\sum_{i=1}^{n} a_i\cdot y_i\cdot(1+\boldsymbol{x_i}\cdot\mathbf{z})^d$的值。通过旋转(Rotate)操作和同态加法(Add)操作，服务器端计算第一列的和，方法如下：

$$ct_5 = ct_5 \oplus Rotate(ct_5, 2^j) \quad 4\text{-}(17)$$

其中$j = \log_2 t, \log_2 t+1, \cdots, \log_2 t+\log_2 n-1$。迭代计算后第一行第一列的元素即为所求结果的密文，其它槽中的结果为无效值，统一用*表示。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

$$ct_6 = Enc\begin{bmatrix} \sum_{i=1}^{n} e_i \cdot (1+\boldsymbol{x}_1 \cdot \mathbf{z})^d & \cdots & * \\ \vdots & \vdots & * \\ * & \cdots & * \end{bmatrix} \tag{4-(18)}$$

如例 b 所示，运用边旋转边求和的方法，可以用较小的运算量计算第一列的和。

例 b：给定一个4×4的密文矩阵ct_A，以下演示旋转加的过程。

$$ct_A = Enc\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix} \tag{4-(19)}$$

步骤 b1：密文内部依次旋转 4 位得到ct_{A_1}。

$$ct_{A_1} = Enc\begin{bmatrix} 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} \leftarrow Enc\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix} \tag{4-(20)}$$

步骤 b2：旋转后密文与原密文矩阵相加，$ct_{A_2} = ct_A + ct_{A_1}$。

$$ct_{A_2} = Enc\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \end{bmatrix} + Enc\begin{bmatrix} 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \\ 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \end{bmatrix} = Enc\begin{bmatrix} 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 \\ 7 & 7 & 7 & 7 \\ 5 & 5 & 5 & 5 \end{bmatrix} \tag{4-(21)}$$

步骤 b3：密文内部依次旋转 8 位。

$$ct_{A_3} = Enc\begin{bmatrix} 7 & 7 & 7 & 7 \\ 5 & 5 & 5 & 5 \\ 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 \end{bmatrix} \leftarrow Enc\begin{bmatrix} 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 \\ 7 & 7 & 7 & 7 \\ 5 & 5 & 5 & 5 \end{bmatrix} \tag{4-(22)}$$

步骤 b4：旋转后密文与旋转前密文相加，$ct_{A_4} = ct_{A_2} + ct_{A_3}$。

$$ct_{A_4} = Enc\begin{bmatrix} 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 \\ 7 & 7 & 7 & 7 \\ 5 & 5 & 5 & 5 \end{bmatrix} + Enc\begin{bmatrix} 7 & 7 & 7 & 7 \\ 5 & 5 & 5 & 5 \\ 3 & 3 & 3 & 3 \\ 5 & 5 & 5 & 5 \end{bmatrix} = Enc\begin{bmatrix} 10 & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{bmatrix} \tag{4-(23)}$$

与上面的求行和类似，仅通过几次旋转求和即可在空间复杂度和时间复杂度都较小的

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

情况下求第一列的和。同样的，本方法只支持矩阵行列的数量均为 2 的幂的情况，如果不足应补零。

步骤七：本步骤将密文 ct_6 加明文 B 得到最终的结果。

$$ct_7 = Enc\begin{bmatrix} b+\sum_{i=1}^{n} e_i \cdot (1+\boldsymbol{x}_1 \cdot \mathbf{z})^d & \cdots & * \\ \vdots & \vdots & * \\ * & \cdots & * \end{bmatrix} \tag{4-(24)}$$

至此计算完成了所有的线性操作步骤，利用 SIMD 技术和特殊的运算方式高效的完成了整个计算。至此已经基本完成分类，可将密文结果返回，并由客户端解密后判断符号，但缺点是可能会造成云端分类模型泄露。由于全同态加密不支持非线性运算，因此无法计算分类预测模型最后的 *sign* 函数，为进一步保护云端分类模型的安全，下小节介绍如何利用非线性近似同态计算判断分类结果的符号。

4.4 同态计算非线性函数

对全同态加密来说，非线性运算一直是个难题，至今也没有很好的解决。比较常用的两个方法是用电路门计算和近似计算。因为 *sign* 函数本质上是一个比较操作，即比较自变量和 0 之间的大小关系，所以本文采用基于近似计算的 Cheon 等[57]的比较方法处理 *sign* 函数。该算法基于近似求逆算法。

4.4.1 求逆算法

戈尔德施密特除法算法[58]是求正实数逆的一个著名算法。对于给定的 $x \in (0,2)$，戈尔德施密特除法算法 $Inv(x;d)$ 的主要思想是：

$$\frac{1}{x} = \frac{1}{1-(1-x)} = \prod_{i=0}^{\infty}(1+(1-x)^{2^i}) \approx \prod_{i=0}^{d}(1+(1-x)^{2^i}) \tag{4-(25)}$$

当 i 趋近于无穷大的时候，$1+(1-x)^{2^i}$ 趋近于 1，所以该算法当 d 足够大的时候同样适用，其中 $d>0$。以下 **Algorithm 5** 为基于公式 4-(25)的近似求逆算法。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

近似求逆算法：

Algorithm 5: *Inv(x; d)*

Input: 0<x<2，d∈N

Output: an approximate value of $1/x$

1: $a_0 \leftarrow 2 - x$

2: $b_0 \leftarrow 1 - x$

3：**for** $n \leftarrow 0$ to $d-1$ **do**

4: $b_{n+1} \leftarrow b_n^2$

5: $a_{n+1} \leftarrow a_n \cdot (1 + b_{n+1})$

6: **end for**

7: return a_d

4.4.2 比较算法

本小节介绍如何有效的计算两个数的近似比较结果。比较公式如 4-(26)，$a < b$ 时结果逼近 0，$a > b$ 时结果逼近 1。

$$\frac{a^k}{a^k + b^k} \approx comp(a,b) \tag{4-(26)}$$

虽然该公式输出结果与 *sign* 函数略有不同，但不影响最终分类结果的判断。以下应用 **Algorithm 5** 实现近似的比较算法。

近似比较算法：

Algorithm 6: $comp(a,b;d,d',t,m)$

Input: distinct numbers a,b∈$[\frac{1}{2},\frac{3}{2})$,d, d',t,m∈N

Output: an approximate value of comp(a,b)

1: $a_0 \leftarrow \frac{a}{2} \cdot Inv(\frac{a+b}{2}; d')$

2: $b_0 \leftarrow 1 - a_0$

3：**for** $n \leftarrow 0$ to $t-1$ **do**

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

4: $inv \leftarrow Inv(a_n^m + b_n^m; d)$

5: $a_{n+1} \leftarrow a_{\mathrm{n}}^m \cdot inv$

6: $b_{n+1} \leftarrow 1 - a_{n+1}$

7: **end for**

8: **return** a_t

以上的两个算法均对输入的自变量范围有明确要求，因此比较前应先将输入值映射至目标范围。如此假设输入值处于区间$[O_{\min}, O_{\max})$，映射目标区间为$[\frac{1}{2}, \frac{3}{2})$，则有如下映射公式：

$$y = \frac{1}{O_{\max} - O_{\min}} \cdot (x - O_{\min}) + \frac{1}{2} \tag{4-(27)}$$

4.5 实验结果与安全性分析

本小节，主要实现上文中所提出的算法，并将其与最近的类似工作进行比较。该实验的实验环境为 $Intel(R)\ Xeon(R)\ E5-2620\ v2\ CPU\ 2.10\ GHz$ Intel(R) ，$31G\ RAM$，基于 SEAL 库[58]实现代码（SEAL 实现了 CKKS 方案[31]及其变体）。

本实验采用 cifar-10[60]数据集。该数据集包含 10 类图片各 6000 张，像素为 32*32。考虑到这是一个二分类问题，选取任意一类图片（本实验选取的是卡车类），然后把该类的标签值设为 1，其他类的标签值设为-1，在此条件下进行实验。首先运用 HOG 特征提取[54]方法和 PCA 降维[56]技术将待分类图片特征提取并降维，操作过后每一张图片被转化为一个一维的向量。接下来运用上文的算法实现加密图像分类。

4.5.1 时间复杂度对比

表 4.1 是 4.3 小节中的算法和 Barnett 等[44]的方法时间复杂度比较结果，比较的变量为同态密文的操作次数。其中 Mult 和 CMult 分别表示密文-密文乘和密文-明文乘。Add 和 CAdd 分别表示密文-密文加和密文-明文加。Rot 表示密文内部的旋转操作。HM 表示汉明距离。在 Barnett 等[44]的方法中，服务器端把每个特征向量的元素分别加密，并把密文发送到服务器端，该方法需要 t 个密文表示一张图片，通信开销极大。此外，他们的方案没

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

有使用 SIMD 技术，当多项式核的度数为d时，需要n次同态指数操作和n次内积操作。如此总共需要$n\times(\lfloor\log_2 d\rfloor+HW(d)-1)$次密文乘和$n\times f$次密文明文乘。而本文的方法基于 SIMD 技术把一个向量打包加密为一个密文，大大地提高了通信和计算效率，仅需要$\lfloor\log_2 d\rfloor+HW(d)-1$次密文乘和两次密文明文乘法。尽管本方法需要额外的旋转操作来实现计算，但是实验结果表明，密文旋转的计算量和运行时间均低于密文乘法，且本算法只需要少量的旋转操作，故本方法远远的优于 Barnett 等[44]的方法。

表 4.1 时间复杂度对比

Methods operations	Mult	CMult	Add	CAdd	Rot
Barnett 等[44]	$n\times(\lfloor\log_2 d\rfloor+HW(d)-1)$	$n\times f$	$n\times f+1$	$n+1$	-
Ours	$\lfloor\log_2 d\rfloor+HW(d)-1$	2	$\lceil\log_2 t+\log_2 n\rceil$	2	$\lceil\log_2 t+\log_2 n\rceil$

4.5.2 运行时间对比

在相同的实验环境和同态加密库下，运行了 4.3 小节中的算法和 Barnett 等[9]的方法，进行了时间比较。本实验将 PCA 降维的维度设为 t=32，并分别根据多项式核的维度 d=1,2,3,4 训练四个支持向量机模型，分别对应的支持向量个数为 1785,1235,1898,2793。两个算法都用了相同的 SVM 模型，并且特征向量的维度均为 32，保证在相同环境下比较。

下面介绍一下如何选择底层全同态加密方案 CKKS 的参数来支持本文的算法。CKKS 方案基于环 LWE(RLWE)假设问题。其中环$\mathbb{R}=Z[X]/X^N+1$，N 为 2 的幂。明文空间定义为$R_q:=R/qR$在，其中 q 为素数。全同态加密的 CKKS 方案提供了一个技术可将一个 N/2 维的实数多项式打包，且支持在只进行一次操作的条件下同时进行 N/2 次加法或乘法。本实验选择的循环多项式的度$N=2^{15}$，选择的新鲜密文（未同其他密文计算过的初始密文）的模数为$\log_q=200,280,320,320$，分别对应多项式核函数度数为$d=1,2,3,4$，安全级别为 128。由于 CKKS 同态加密方案中的每个密文在$\mathbb{R}_q$上需要两个多项式，而多项式中的每个系数都需要$\log_q$比特，所以每个新鲜密文的大小为$2\times N\times\log_q$ bit。表 4.2 为本文方法与

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

Barnett 等[44]方法的实验结果比较。从表格中可得，本文方法密文所需内存更小且在没有处理分类最后的非线性操作情况下，本文的方法和 Barnett 等[44]相比计算效率更高。例如，他们的方法在 $d=1,2,3$ 的时候需要几百秒，$d=4$ 时超出内存无法运算。而本文的方法仅仅需要几秒钟的时间便可实现分类。

表 4.2 本文方法与 Barnett 的方法的时间比较

Methods	Parameters			SVs	Ciphertext size(MB)	Running Time(s)		
	d	N	logq			Enc	Dec	HE-SVM
Barnett et al.[44]	1	2^{13}	200	1785	11.25	7.798	0.001	179.177
	2	2^{14}	240	1235	30	16.216	0.003	417.634
	3	2^{14}	280	1898	35	43.644	0.005	916.656
	4	2^{14}	320	2793	40	-	-	-
Ours	1	2^{15}	200	1785	1.56	0.105	0.004	0.554
	2	2^{15}	280	1235	2.19	0.125	0.007	1.107
	3	2^{15}	320	1898	2.5	0.164	0.007	2.065
	4	2^{15}	320	2793	2.5	0.219	0.007	3.067

SVs：支持向量的个数。Ciphertext size：新鲜密文的大小。Enc：加密时间。Dec：解密时间。HE-SVM：同态运算 SVM 分类的时间。

4.5.3 同态近似比较实验结果

以下应用 4.4 小节中所提到的同态近似比较方法处理分类公式最后的非线性运算，实验结果如表 4.3 所示。对于比较公式 4-(26)，选取 $k=2$。尽管选取的 k 的值越大比较效果越好，但 $k=2$ 对本实验来说已经足够区分正负样本。为了同态计算函数 $sign(x)$，先将分类后的密文结果 x 与 0 分别根据公式 4-(27)映射，然后应用算法 6 比较得到最终结果。得到最终结果之后将其返回给客户端，客户端自行解密得到图片的分类结果，至此客户端委托服务端对图片的分类全部完成，且全程无隐私泄露。本实验选择的循环多项式的度 $N=2^{15}$，选择的新鲜密文的模数为 $\log_q=600,640,680,680$，分别对应多项式核函数度数为 $d=1,2,3,4$，安全级别为 128。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

表 4.3 同态支持向量机分类的实验结果

Methods	Parameters			SVs	Ciphertext size(MB)	Running Time(s)		
	d	N	logq			Enc	Dec	Running time
Ours	1	2^{15}	600	1785	4.69	0.314	0.005	7.972
	2	2^{15}	640	1235	5	0.294	0.005	8.391
	3	2^{15}	680	1898	5.31	0.353	0.005	12.282
	4	2^{15}	680	2793	5.31	0.456	0.005	17.039

4.5.4 与其他类似工作的比较

以下介绍一下本文的方法和其他基于加同态和安全多方计算的方案之间的比较。以下是表格中符号意义：Collusion Attack 指共谋攻击，如果一个方案中需要多个协作服务器共同运作，就易受到服务器之间合谋攻击。Interaction(C-S)表示在计算过程中，除了至少两轮上传和下载通信之外，客户端和服务器之间是否有通信。Ciphertexts 表示一个图像所需要的密文数量。如表 4.4 所示，Rizai 等[46]和 Marki 等[47]的基于安全多方计算的方法需要客户端和服务器端进行多轮的交互而本文的方法除了必要的上传下载之外，不需要额外的交互，这有效的防止了共谋攻击。另一方面，类似于 Barnett 等[44]的工作，Rahulamathavan 等[43]的方案还分别对特征向量的每个元素进行加密，从而导致大量的通信开销和相当高的计算开销。此外，由于 AHE 技术的限制，Rahulamathavan 等[43]的方案需要客户端和服务器之间的交互来计算多项式核。相较来说，本文的方法则不需要交互。

表 4.4 该方案与其他类似工作的比较

Schemes Properties	Technique	Collusion Attack	Interaction(C-S)	Ciphertexts
Rizai et al.[46]	MPC	No	Yes	-
Marki et al.[47]	MPC	Yes	No	-
Rahulamathavan et al.[43]	AHE	No	Yes	n
Barnett et al.[44]	FHE	No	No	n
Ours	FHE	No	No	1

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

4.5.5 安全性分析

本文假设客户端是诚实的并且安全威胁只来自云服务器。文中提出的方案的目的是保护客户端的图像隐私不被云服务器所窃取。本文希望设计一个客户端最终获得分类结果，但云服务器不获得有关客户端图像信息的协议。如果底层同态加密方案是安全的，则云服务器端只能看到加密的特征向量而不能获得任何有关图像的有用信息。同样的，在选择明文攻击(IND-CPA)模型[38]下，底层的 CKKS 方案已被证明是安全的，这意味着密文信息在运算中是不可被分辨的，因此客户端的隐私能够得到保护。

需要注意的是，虽然本文最后应用同态近似比较方法解决最后一步的密文非线性运算问题，但不论最后的分类结果是近似值还是准确值，仍然会面临云服务端模型泄露的危险。而如果引进多服务器解决这个问题，就会面临共谋攻击的风险。因为本文关注的重点是客户端隐私不泄露，所以这部分不做过多讨论。

4.6 本章小结

本章基于矩阵运算和全同态加密提出了一个保护隐私的支持向量机分类方法。基于 SIMD 技术和矩阵运算技巧，本章的方法与之前的工作相比显著地提高了通信效率和计算效率。实验结果显示本文的方法只需几秒即可分类加密图像，而同类的方法需要上百秒。另一方面，与其他的需要与多服务器端进行多轮交互的方案相比，本文的方案是非交互式的且只需一个服务器，能有效的避免服务器之间的共谋攻击。

本方案假设在 SVM 分类之前在客户端执行包括提取特征和降维在内的预处理步骤。未来可能的工作是将预处理步骤交给云服务器，以进一步减轻客户端的计算开销。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

第五章 总结与展望

5.1 总结

在数学领域，矩阵运算是最基本的问题之一，其在科学和工程领域都有着广泛的应用，各种算法中也少不了它的身影。对于计算资源有限的客户端，要将需要大量计算资源的矩阵运算及其应用交给云服务器计算就要考虑数据隐私泄露的问题。全同态加密技术支持密文间的计算，它为解决隐私计算问题提供了一个方法。本文基于全同态加密技术研究安全矩阵运算及其应用，主要工作如下：

1.提出了基于全同态加密的任意矩阵乘法计算方法。现有的加密矩阵乘法只支持两个方阵之间的运算，而非方阵的乘法只能将其扩充为方阵再计算，因而计算效率低效。本文基于超立方结构的编码方法，提出了支持任意维度的安全高效的矩阵乘法计算方法。该方法只需一个单独的密文即可表示一个矩阵且只需更少的乘法次数。与其他的基于伪装的技术相比，本文提出的算法具有可合成性，并可将该算法作为模块应用于其他高级程序中。最后本文将其应用到矩阵连乘上。实验结果表明该方法明显优于现有的最先进的安全矩阵乘以及连乘方法。

2.将安全矩阵运算应用到支持向量机加密图像分类中。本文运用矩阵计算技巧实现了加密图像支持向量机分类方案。针对现有的将加密图像的每个元素加密为一个密文的低效方法，本文利用 SIMD 技术将图像特征向量加密成一个矩阵，然后运用密文的旋转及同态加法乘法操作，对一整个矩阵运算，减少了密文运算次数，提高了加密图像分类效率。最后应用近似的同态比较方法处理分类最后的非线性计算，为同态加密解决非线性运算问题提供了一种方法。实验结果表明本方法只需几秒钟即可实现加密图像分类，而原来的方法需要几百秒。

5.2 展望

本文围绕基于全同态加密的安全矩阵运算展开相应的研究。在下一阶段的研究中，将在已有成果上进一步研究。

1、研究基于超立方体结构的其他与矩阵相关的运算，例如安全矩阵行列式、安全矩阵分解。超立方结构是一个很好的工具，其不仅可以将整个矩阵打包为一个密文，而且能

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

够实现按行按列的旋转操作，因此对安全矩阵运算研究非常有力。

2、研究安全矩阵运算的应用。机器学习中的许多算法都可以表示为矩阵之间的运算，之后的工作将进一步利用全同态加密技术，将安全矩阵运算扩展应用到其他机器学习算法中，解决隐私计算的相关问题。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

参考文献

[1] Alliance C S . Security Guidance for Critical Areas of Focus in Cloud Computing V4.0. 2017.

[2] Gentry C. Fully homomorphic encryption using ideal lattices[C]. Proceedings of the forty-first annual ACM symposium on Theory of computing. 2009: 169-178.

[3] Graepel T, Lauter K, Naehrig M. ML confidential: Machine learning on encrypted data[C]. International Conference on Information Security and Cryptology. Springer, Berlin, Heidelberg, 2012: 1-21.

[4] Ibtihal M, Hassan N. Homomorphic encryption as a service for outsourced images in mobile cloud computing environment[M]. Cryptography: Breakthroughs in Research and Practice. IGI Global, 2020: 316-330.

[5] Benjamin D, Atallah M J. Private and cheating-free outsourcing of algebraic computations[C]. 2008 Sixth Annual Conference on Privacy, Security and Trust. IEEE, 2008: 240-245.

[6] Wang C, Ren K, Wang J, et al. Harnessing the cloud for securely solving large-scale systems of linear equations[C]. 2011 31st International conference on distributed computing systems. IEEE, 2011: 549-558.

[7] Paillier P. Public-key cryptosystems based on composite degree residuosity classes[C]. International conference on the theory and applications of cryptographic techniques. Springer, Berlin, Heidelberg, 1999: 223-238.

[8] Mohassel P. Efficient and Secure Delegation of Linear Algebra[J]. IACR Cryptol. ePrint Arch., 2011, 2011: 605.

[9] Hiromasa R, Abe M, Okamoto T. Packing messages and optimizing bootstrapping in GSW-FHE[J]. IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, 2016, 99(1): 73-82.

[10] Duong D H, Mishra P K, Yasuda M. Efficient secure matrix multiplication over LWE-based homomorphic encryption[J]. 2016.

[11] Rathee D, Mishra P K, Yasuda M. Faster PCA and linear regression through hypercubes in HElib[C]. Proceedings of the 2018 Workshop on Privacy in the Electronic Society. 2018: 42-53.

[12] Cheon J H, Kim A, Yhee D. Multi-dimensional Packing for HEAAN for Approximate Matrix Arithmetics[J]. IACR Cryptol. ePrint Arch., 2018, 2018: 1245.

[13] Jiang X, Kim M, Lauter K, et al. Secure outsourced matrix computation and application to neural

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

networks[C]. Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. 2018: 1209-1222.

[14] Kim D, Son Y, Kim D, et al. Privacy-preserving approximate GWAS computation based on homomorphic encryption[J]. BMC Medical Genomics, 2020, 13(7): 1-12.

[15] Atallah M J, Pantazopoulos K N, Rice J R, et al. Secure outsourcing of scientific computations[M]. Advances in Computers. Elsevier, 2002, 54: 215-272.

[16] Lei X, Liao X, Huang T, et al. Outsourcing large matrix inversion computation to a public cloud[J]. IEEE Transactions on cloud computing, 2013, 1(1): 1-1.

[17] Lei X, Liao X, Huang T, et al. Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud[J]. Information Sciences, 2014, 280: 205-217.

[18] Lei X, Liao X, Huang T, et al. Cloud computing service: The caseof large matrix determinant computation[J]. IEEE Transactions on Services Computing, 2014, 8(5): 688-700.

[19] Yin-jie S. Verifiable and Secure Protocol for Outsourcing Matrix Determinant Computation to Malicious Cloud[J]. Computer and Modernization, 2015 (5): 103.

[20] Fu S, Yu Y, Xu M. Practical privacy-preserving outsourcing of large-scale matrix determinant computation in the cloud[C]. International Conference on Cloud Computing and Security. Springer, Cham, 2017: 3-15.

[21] Fu S, Yu Y, Xu M. A secure algorithm for outsourcing matrix multiplication computation in the cloud[C]. Proceedings of the Fifth ACM International Workshop on Security in Cloud Computing. 2017: 27-33.

[22] Liu J, Bi J, Li M. Secure outsourcing of large matrix determinant computation[J]. Frontiers of Computer Science, 2020, 14(6): 1-12.

[23] Zhang S, Tian C, Zhang H, et al. Practical and secure outsourcing algorithms of matrix operations based on a novel matrix encryption method[J]. IEEE Access, 2019, 7: 53823-53838.

[24] Zhao L, Chen L. Sparse matrix masking-based non-interactive verifiable (outsourced) computation, revisited[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 17(6): 1188-1206.

[25] Yao A C C. How to generate and exchange secrets[C]. 27th Annual Symposium on Foundations of Computer Science (sfcs 1986). IEEE, 1986: 162-167.

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

[26] 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.

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

[28] Brakerski Z, Gentry C, Vaikuntanathan V. (Leveled) fully homomorphic encryption without bootstrapping[J]. ACM Transactions on Computation Theory (TOCT), 2014, 6(3): 1-36.

[29] Cheon J H, Kim A, Kim M, et al. Homomorphic encryption for arithmetic of approximate numbers[C]. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Cham, 2017: 409-437.

[30] Gentry C, Sahai A, Waters B. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based[C]. Annual Cryptology Conference. Springer, Berlin, Heidelberg, 2013: 75-92.

[31] Cheon J H, Kim A, Kim M, et al. Homomorphic encryption for arithmetic of approximate numbers[C]. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Cham, 2017: 409-437.

[32] Smart N P, Vercauteren F. Fully homomorphic SIMD operations[J]. Designs, codes and cryptography, 2014, 71(1): 57-81.

[33] Rathee D, Mishra P K, Yasuda M. Faster PCA and linear regression through hypercubes in HElib[C]. Proceedings of the 2018 Workshop on Privacy in the Electronic Society. 2018: 42-53.

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

[35] Lu W, Kawasaki S, Sakuma J. Using Fully Homomorphic Encryption for Statistical Analysis of Categorical, Ordinal and Numerical Data[J]. IACR Cryptol. ePrint Arch., 2016, 2016: 1163.

[36] Wang S, Huang H. Secure Outsourced Computation of Multiple Matrix Multiplication Based on Fully Homomorphic Encryption[J]. KSII Transactions on Internet and Information Systems (TIIS), 2019, 13(11): 5616-5630.

[37] Goldreich O. Foundations of cryptography: volume 2, basic applications[M]. Cambridge university press,

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

2009.

[38] Katz J, Lindell Y. Introduction to modern cryptography[M]. CRC press, 2020.

[39] Mishra P K, Duong D H, Yasuda M. Enhancement for secure multiple matrix multiplications over ring-LWE homomorphic encryption[C]. International Conference on Information Security Practice and Experience. Springer, Cham, 2017: 320-330.

[40] Google Prediction API, Available at https://cloud.google.com/prediction/.

[41] Amazon AWS, Available at https://aws.amazon.com.

[42] Microsoft Azure Machine Learning, Available at https://azure.microsoft.com, 2020.

[43] Rahulamathavan Y, Phan R C W, Veluru S, et al. Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud[J]. IEEE Transactions on Dependable and Secure Computing, 2013, 11(5): 467-479.

[44] Barnett A, Santokhi J, Simpson M, et al. Image Classification using non-linear Support Vector Machines on Encrypted Data[J]. IACR Cryptol. ePrint Arch., 2017, 2017: 857.

[45] Goldreich O. Foundations of cryptography: volume 2, basic applications[M]. Cambridge university press, 2009.

[46] Riazi M S, Weinert C, Tkachenko O, et al. Chameleon: A hybrid secure computation framework for machine learning applications[C]. Proceedings of the 2018 on Asia Conference on Computer and Communications Security. 2018: 707-721.

[47] Makri E, Rotaru D, Smart N P, et al. EPIC: efficient private image classification (or: Learning from the masters)[C]. Cryptographers' Track at the RSA Conference. Springer, Cham, 2019: 473-492.

[48] Yu H, Jiang X, Vaidya J. Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data[C]. Proceedings of the 2006 ACM symposium on Applied computing. 2006: 603-610.

[49] Laur S, Lipmaa H, Mielikäinen T. Cryptographically private support vector machines[C]. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006: 618-624.

[50] Vaidya J, Yu H, Jiang X. Privacy-preserving SVM classification[J]. Knowledge and Information Systems, 2008, 14(2): 161-178.

[51] Lin K P, Chen M S. Privacy-preserving outsourcing support vector machines with random

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

transformation[C]. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. 2010: 363-372.

[52] Liu X, Deng R H, Choo K K R, et al. Privacy-preserving outsourced support vector machine design for secure drug discovery[J]. IEEE Transactions on Cloud Computing, 2018, 8(2): 610-622.

[53] Liu X, Choo K K R, Deng R H, et al. Efficient and privacy-preserving outsourced calculation of rational numbers[J]. IEEE Transactions on Dependable and Secure Computing, 2016, 15(1): 27-39.

[54] Cortes C, Vapnik V. Support-vector networks[J]. Machine learning, 1995, 20(3): 273-297.

[55] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]. 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Ieee, 2005, 1: 886-893.

[56] Alpaydin E. Introduction to machine learning[M]. MIT press, 2020.

[57] Cheon J H, Kim D, Kim D, et al. Numerical method for comparison on homomorphically encrypted numbers[C]. International Conference on the Theory and Application of Cryptology and Information Security. Springer, Cham, 2019: 415-445.

[58] Goldschmidt R E. Applications of division by convergence[D]. Massachusetts Institute of Technology, 1964.

[59] Microsoft SEAL release 3.4, https://github.com/Microsoft/SEAL, 2019.

[60] CIFAR-10, https://www.kaggle.com/c/cifar-10/data.

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

攻读硕士期间的研究成果

●论著

[1]1/3,Secure Outsourced Computation of Matrix Determinant Based on Fully Homomorphic Encryption,IEEE Access（录用）

[2]2/3,Secure Matrix Multiplication Based on Fully Homomorphic Encryption and Its Application,information sciences（审稿中）

[3]3/3,Support Vector Machine Classification over Encrypted Data,Peer-to-Peer Networking and Applications（审稿中）

●科研项目

[1]卷积神经网络加密图像分类算法研究，Y201942830，2019.11-2020.11，浙江省教育厅一般科研项目，主要负责人

●专利

[1]2/2,发明专利一项（受理）

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

致 谢

光阴似箭，岁月如梭，三年的研究生生涯已然接近尾声。在这过去的三年中，自己学到了很多，收获了很多，当然也有一些遗憾。在这学生生涯最后的时光里，充满不舍，展望未来步入社会的生活，也充满了期待。在理工学习的这三年，自己不仅在学习方面，在做人做事方面也学到了很多，今后不论走到哪，都不忘学校的栽培之情。在这临别之际，我想向所有关心我、支持我、帮助我的家人、老师、同学和朋友致以最诚挚的谢意。

首先感谢我的导师。他是一个学识渊博、为人亲和、做事认真的人。老师做事事必躬亲，细致入微，跟随老师学习的这三年，不仅科研能力有所提升，也学会了如何做人、做事，在为人处世方面有十足的进步。在此感谢老师对我的学业的指导、生活的关心。

其次我要感谢我的师兄师姐。不论学习还是生活上，每当我遇到困难，师兄师姐总能给我关怀和指导。这种既是兄长又是同学的情谊，我将永远珍惜。感谢实验室的小伙伴还有同寝室的兄弟们，研究生生涯能和你们这样一群可爱的人一起度过，我很感恩。

最后感谢我的家人，谢谢你们一直以来对我无私的关爱和牵挂。在我生活苦恼、心情烦闷的时候，是父母鼓励我一直前行。未来的路还很长，而你们永远是我勇往无前的动力。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net