计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.201200124

基于同态加密的线性系统求解方案

吕　由[1,2]　吴文渊[1]
1 中国科学院重庆绿色智能技术研究院　重庆 400714
2 中国科学院大学　北京 100049
(lvyou@cigit.ac.cn)

摘　要　在科学计算、统计分析以及机器学习领域,许多实际问题都可以归结到线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 的求解,如最小二乘估计和机器学习中的回归分析等。而实际中用于计算的数据往往由不同用户拥有且包含用户的敏感信息。当不同的数据拥有者想在合作求解一个模型的同时保护数据的隐私,同态加密可以作为解决方法之一。针对两个用户参与的场景,基于 Cheon 等提出的 HEAAN 同态加密技术,设计了一种两方参与、利用 Gram-Schmidt 正交化方法安全求解线性系统 $Ax=b$ 的新方案;提出了一种适用于该场景的交互式安全乘法逆协议,解决了同态加密无法高效计算除法的问题,保证在高效计算的同时保护数据的隐私信息;分析了方案的安全性、通信损耗以及计算复杂度;基于 HEAAN 同态加密库,利用 C++ 实现了该方案;最后通过大量的实验证明,该方案可以安全高效地求解维度不超过 17 的线性系统,与在明文数据上的计算结果相比,相对误差不超过 0.0001;针对该方案设计的平行编码方法,可以通过 SIMD 技术并行求解多个线性系统,拓宽了方案的可用性,基本满足特定场景下的实际应用需求,可进一步用于隐私保护数据挖掘算法的设计。

关键词:线性系统;Gram-Schmidt 正交化;隐私保护;同态加密;HEAAN
中图法分类号　TP309.7

Linear System Solving Scheme Based on Homomorphic Encryption

LYU You[1,2] and WU Wen-yuan[1]
1 Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China
2 University of Chinese Academy Sciences, Beijing 100049, China

Abstract　In the fields of scientific computing, statistical analysis and machine learning, many practical problems can be reduced to solving linear system $Ax=b$, such as least square estimation and regression analysis in machine learning. In practice, the data used for calculation often belong to different users, containing their sensitive information. When different data owners want to collaboratively solve a model, homomorphic encryption can be one of the methods to deal with the privacy leakage in computing. In a scenario with only two parties, based on the HEAAN scheme proposed by Cheon et al, we propose a new scheme involving two-party to securely solve the linear system through Gram-Schmidt orthogonalization, and design an interactive secure multiplicative inverse protocol to solve a problem that they cannot do efficient division. We analyze the security, communication cost and computational complexity, and also implement our scheme based on HEAAN library using C++ language. Through a large number of experiments, it shows that our scheme can solve a linear system with dimension up to 17 safely and efficiently. Compared with the results on unencrypted data, the relative error is no more than 0.0001. By the proposed parallel encoding method, our scheme can process multiple linear systems simultaneously in SIMD mode, which expands the availability of the scheme. Our scheme can be practically applied in some specific scenarios, and can be further used for the design of privacy-preserving data mining algorithms.

Keywords　Linear system, Gram-Schmidt orthogonalization, Privacy preserving, Homomorphic encryption, HEAAN

1　引言

1.1　研究背景

线性方程组 $\boldsymbol{Ax}=\boldsymbol{b}$ 的求解在实际中有着广泛的用途,科学计算、统计分析以及机器学习领域中的许多问题都可以归结到该线性系统的求解。机器学习依赖于数据的获取,在实际场景中用于计算和分析建模的数据可能由多个用户拥有[1]且包含不同用户的敏感信息。当多个数据持有者想要集中各自的数据以合作求解某个问题或者模型时,数据的隐私泄露就成为了重大的安全问题,尤其是在医药、金融等行业中。在保证高效数据处理的同时防止数据中敏感信息的泄露逐渐成为一个重要的研究方向。随着同态加密[2]、安全多方计算[3]

到稿日期:2020-12-14　返修日期:2021-04-23　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:重庆市科委项目(cstc2018jcyj-yszxX0002);贵州省科技计划项目([2020]4Y056);国家重点研发计划(2020YFA0712300)
This work was supported by the Chongqing Science and Technology Program(cstc2018jcyj-yszxX0002), Guizhou Science and Technology Program([2020]4Y056) and National Key Research and Development Project(2020YFA0712300).
通信作者:吴文渊(wuwenyuan@cigit.ac.cn)

以及差分隐私[4]等技术的出现，各种隐私保护数据处理算法被广泛用于实际场景。其中，同态加密由于自身的技术特性，在防止数据的隐私泄露方面有着天然的优势。

同态加密作为一种特殊的公钥加密技术，允许我们直接在密文上进行运算，得到的结果在解密后与在未加密数据上执行相同的运算得到的结果一致。这种特性让我们可以直接在密文域上进行数据处理。由于文献[2]中 Gentry 的突破性贡献，基于格理论的同态加密技术引起了研究人员的关注，利用 bootstrapping 技术可以实现全同态加密(FHE)方案，支持密文加法以及乘法运算，且可以计算任意深度的函数。但是，由于 bootstrapping 代价过高，在实际中应用广泛的是效率更高的层次型同态加密技术(LHE[5])，其只能计算有限深度的函数。已有的比较成熟的方案包括 BGV[6]，BFV[7]，HEAAN[8]等，可以抵抗量子攻击，实现复杂的密文计算。目前大多数同态加密技术的瓶颈是不支持高效的密文除法运算，文献[9]中的研究实现了整数环上的同态密文除法，但是运算代价很高，在实际应用中，这也是我们需要解决的主要问题之一。

在数据被不同用户持有的场景下，我们将实际问题转化为求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 后，矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 则可能包含不同用户的敏感数据，如在机器学习中，常用最小二乘拟合方法将训练线性回归模型转化为求解 $\boldsymbol{Ax}=\boldsymbol{b}$，预处理所有用户的数据集之后得到矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$。如果在明文状态下进行处理，用户的隐私信息则会暴露给负责集中数据进行计算的一方。因此，在不泄露用户隐私的前提下，寻求一种安全求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 的方法有着很大的实际应用价值，基于同态加密研究密文域上线性系统的求解方案，也可以作为设计其他复杂隐私保护数据挖掘算法的基础。

1.2　相关工作

已经有很多关于隐私保护场景下求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 的研究，文献[10-14]都是基于求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$，在多方分布的数据上训练隐私保护线性回归模型，但研究的应用场景与本文不同。文献[10-14]研究的场景包含多个客户端和两个服务器(分别为提供加密服务的云服务端 CSP 和负责计算的服务端 Evaluator)，各客户端将自己的数据利用 CSP 提供的公钥加密后上传给 Evaluator 来求解线性系统。基于这样的场景，Nikolaenko 等[10]在 2013 年利用 Cholesky 分解求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$，结合 Paillier[15]加法同态加密方案，利用姚氏混淆电路来完成求解过程中的一些复杂运算。由于需要设计复杂的电路，该方案的效率很低，需要高昂的通信代价。Hu 等[11]于 2017 年分别利用密文域上的高斯消元法和雅可比迭代法求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$，基于 Paillier 同态加密方案设计了一种新的支持打包的安全乘法协议，相比文献[10]的方案，文献[11]的方案提高了模型的计算效率，减少了通信损耗。

Chen 等[12]于 2018 年设计的隐私保护线性回归分析协议同时利用乘法同态和加法同态操作来实现数据整合和求解线性方程组。利用同态加密，服务器通过交互协议来实现安全的 LDLT 分解，以高效地求解线性系统。他们单纯地基于同态加密技术构造的协议，比一些混合利用混淆电路的方案更加高效。

Giacomelli 等[13]于 2018 年提出了一种基于 Paillier 同态加密和数据脱敏技术的协议，在加密后的数据上实现了脱敏技术，掩盖了原始数据信息，解密后再在包含噪声的数据上实现线性系统的安全求解。Qiu 等[14]于 2020 年使用不同的数据脱敏方法设计了高效的解决方案，且保证结果有足够的准确度，证明了数据脱敏技术可以保证足够的安全性并且能有效地减少计算代价。

上述方案虽然都是基于数据分布在多用户场景下线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 的安全求解，但所有用户的数据都需要外包给两个非共谋的服务器来完成计算，且使用的 Paillier 加法同态加密方案只能进行简单的密文计算，灵活性较差。为了实现复杂的计算，需要结合安全多方计算、数据脱敏等其他技术来设计复杂的交互协议。

与本文工作最相近的是文献[16]中的研究，Zhang 等于 2019 年基于 BFV 同态加密方案，利用施密特正交化技术实现了矩阵 QR 分解的安全外包计算。客户端将数据加密后外包给服务器，由服务器在加密数据上求解矩阵的 QR 分解。两者的相似点是都实现了加密数据上的 Gram-Schmidt 正交化过程，可以进一步用于线性系统的求解。但文献[16]中的方案存在的主要问题在于，其利用 Newton 迭代法[17]实现加密数据上的除法运算，在加密状态下由于缺乏足够的明文信息指导迭代初值以及迭代次数的选择，Newton 迭代法的收敛性和精度得不到保证；且在迭代过程中需要额外消耗大量的同态乘法层数，增加计算代价，导致噪声增长过快，可以求解的问题规模也受到了极大限制；他们仅提到对一些计算的中间值进行重加密来解决计算过程中噪声膨胀的问题，这需要通过用户与服务器的交互，但他们没有分析具体的方法。

1.3　本文的贡献

本文的主要贡献是：1)针对两个用户的场景，基于 HEAAN 同态加密技术，提出了一种两方参与、利用 Gram-Schmidt 正交化方法安全求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 的新方案，分析了方案的安全性、计算复杂度以及通信损耗；2)结合同态加密和添加随机扰动的技术，设计了一种适用于所给场景下的交互式安全乘法逆协议，用于弥补同态加密不支持高效密文除法的缺点；3)通过本文提出的平行编码方式，可以利用 SIMD 技术[18]在相同的时间损耗下并行求解多个不同的线性系统，拓展了方案的可用性；4)基于 HEAAN 同态加密库[19]，用 C++语言实现了本文方案。最后，大量的实验结果表明，在加密数据上求得的结果与在明文状态下执行相同的算法得到的结果相比，相对误差只有 10^{-4}，矩阵 $\boldsymbol{A}$ 的维度不超过 17 时，方案的效果很好，基本满足所给场景下的实际应用需求。

2　预备知识

2.1　Gram-Schmidt 正交化

本文的目的是在密文域上求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$，假设 $\boldsymbol{A}$ 始终为满秩矩阵。受限于密文域上除法运算代价过高的问题，我们要合理地设计求解算法以适用于同态加密技术，减少线性系统求解过程中需要的除法运算。本文主要利用 Gram-Schmidt 正交化方法，通过矩阵分解求解线性系统，使用该方法的好处是不需要矩阵求逆，对于 n 维线性系统，所需的除法次数仅为 n。

令 $\boldsymbol{A}=(\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n)$，$\boldsymbol{a}_i$ 为 $\boldsymbol{A}$ 矩阵的列向量，首先对 $\boldsymbol{A}$ 矩阵的列向量执行 Gram-Schmidt 正交化过程，得到正交矩阵 $\boldsymbol{P}=(\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_n)$，推导过程如下：

$$\begin{cases}\boldsymbol{p}_1=\boldsymbol{a}_1\\ \boldsymbol{p}_2=\boldsymbol{a}_2-C_{12}\cdot\boldsymbol{p}_1=R_{12}\cdot\boldsymbol{a}_1+\boldsymbol{a}_2\\ \boldsymbol{p}_3=\boldsymbol{a}_3-C_{13}\cdot\boldsymbol{p}_1-C_{23}\cdot\boldsymbol{p}_2\\ \quad=R_{13}\cdot\boldsymbol{a}_1+R_{23}\cdot\boldsymbol{a}_2+\boldsymbol{a}_3\\ \quad\cdots\\ \boldsymbol{p}_i=\boldsymbol{a}_i-C_{1,i}\cdot\boldsymbol{p}_1-C_{2,i}\cdot\boldsymbol{p}_2-\cdots-C_{i-1,i}\cdot\boldsymbol{p}_{i-1}\\ \quad=R_{1,i}\cdot\boldsymbol{a}_1+R_{2,i}\cdot\boldsymbol{a}_2+\cdots+R_{i-1,i}\cdot\boldsymbol{a}_{i-1}+\boldsymbol{a}_i\end{cases}\tag{1}$$

其中，$C_{i,j}=\dfrac{\langle\boldsymbol{p}_i,\boldsymbol{a}_j\rangle}{\langle\boldsymbol{p}_i,\boldsymbol{p}_i\rangle}_{1\leqslant i<j\leqslant n}$，$\langle\cdot,\cdot\rangle$ 表示向量的内积，得到上三角矩阵：

$$\boldsymbol{C}_{n\times n}=\begin{bmatrix}1&C_{12}&C_{13}&\cdots&C_{1n}\\0&1&C_{23}&\cdots&C_{2n}\\0&0&1&\cdots&C_{3n}\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&1\end{bmatrix}$$

上述 Gram-Schmidt 正交化过程可表示为矩阵形式：

$$\boldsymbol{P}=\boldsymbol{A}\cdot\boldsymbol{R}\tag{2}$$

其中，

$$\boldsymbol{R}_{n\times n}=\begin{bmatrix}1&R_{12}&R_{13}&\cdots&R_{1n}\\0&1&R_{23}&\cdots&R_{2n}\\0&0&1&\cdots&R_{3n}\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&0&0&\cdots&1\end{bmatrix}$$

是上三角矩阵，其主对角线元素都为 1，容易得到递推式：

$$R_{i,j}=-C_{i,j}-\sum_{k=i+1}^{j-1}R_{i,k}\cdot C_{k,j\,(1\leqslant i<j\leqslant n)}\tag{3}$$

然后，将 $\boldsymbol{P}$ 矩阵的列向量单位化得到标准正交矩阵 $\boldsymbol{Q}$，有：

$$\boldsymbol{P}\cdot\boldsymbol{\Sigma}=\boldsymbol{Q}\tag{4}$$

其中，

$$\boldsymbol{\Sigma}=\begin{bmatrix}\frac{1}{\|\boldsymbol{p}_1\|}&&&\\&\frac{1}{\|\boldsymbol{p}_2\|}&&\\&&\ddots&\\&&&\frac{1}{\|\boldsymbol{p}_n\|}\end{bmatrix}$$

是对角矩阵，其主对角线元素为 $\Sigma_{ii}=\dfrac{1}{\|\boldsymbol{p}_i\|}(0<i\leqslant n)$，$\|\cdot\|$ 表示向量的 l_2 范数。

将式(1)代入式(3)得到 $\boldsymbol{A}\cdot\boldsymbol{R}\cdot\boldsymbol{\Sigma}=\boldsymbol{Q}$，则 $\boldsymbol{A}^{-1}=\boldsymbol{R}\cdot\boldsymbol{\Sigma}\cdot\boldsymbol{Q}^{-1}$，根据标准正交矩阵的性质有 $\boldsymbol{Q}^{-1}=\boldsymbol{Q}^{\mathrm{T}}=\boldsymbol{\Sigma}\cdot\boldsymbol{P}^{\mathrm{T}}$，因此可以得到 $\boldsymbol{A}^{-1}=\boldsymbol{R}\cdot\boldsymbol{\Sigma}^2\cdot\boldsymbol{P}^{\mathrm{T}}$。在 Gram-Schmidt 正交化过程的每一轮迭代中可分别求出矩阵 $\boldsymbol{R}$，$\boldsymbol{\Sigma}^2$，$\boldsymbol{P}$ 的一个列向量，经过 n 轮迭代即可得到矩阵 $\boldsymbol{R}$，$\boldsymbol{\Sigma}^2$，$\boldsymbol{P}$，进而可以求解 $\boldsymbol{x}=\boldsymbol{R}\cdot\boldsymbol{\Sigma}^2\cdot\boldsymbol{P}^{\mathrm{T}}\cdot\boldsymbol{b}$。

2.2 HEAAN 同态加密

Cheon 等基于 Ring-LWE[20] 问题，提出了一种支持近似数算术的层次型同态加密技术，我们简称其为 HEAAN。其核心思想是将同态加密中引入的噪声看作近似计算过程中误差的一部分，对于明文消息 m，其利用私钥 sk 加密得到的密文 ct 拥有解密结构 $\langle ct,sk\rangle=m+e\approx m(\mathrm{mod}\ q)$。

下文是对 HEAAN 的简要介绍。对于 2 的整数幂 N，我们把 N 维分圆多项式环记作 $R=\mathbb{Z}[X]/(X^N+1)$。对于一个正整数 q，把 R 模 q 上的剩余环记作 $R_q=\mathbb{Z}[X]_q/(X^N+1)$。

(1)密钥生成 KeyGen(1^λ)：指定最大密文模数 Q，选择安全参数 λ，输出环的维数 N。设置私钥 sk 和公钥 pk，计算密钥 evk。

(2)加密 $\mathrm{Enc}_{pk}(m)$：输入明文多项式 $m\in R$，输出对应的密文多项式 ct_m。

(3)解密 $\mathrm{Dec}_{sk}(ct_m)$：输入密文 $ct_m\in R_q^2$，输出解密后的明文多项式 m。

(4)乘法 $\mathrm{Mult}_{evk}(ct_a,ct_b)$：输入 $a,b\in R$ 的密文 $ct_a,ct_b\in R_q^2$，输出 $a\cdot b$ 的密文 $ct_{a\cdot b}$。

(5)加法 $\mathrm{Add}(ct_a,ct_b)$：输入 $a,b\in R$ 的密文 $ct_a,ct_b\in R_q^2$，输出 $a+b$ 的密文 ct_{a+b}。

(6)取负 $\mathrm{Negate}(ct_m)$：输入 m 对应的密文 ct_m，输出 $-m$ 对应的密文 ct_{-m}。

(7)重缩放 $\mathrm{ReScale}(ct_m;p)$：输入 m 对应的密文 ct_m 和 2 的整数幂 p，输出 $\lfloor p^{-1}\cdot m\rceil$ 的密文 $ct_{\lfloor p^{-1}\cdot m\rceil}$。

(8)旋转 $\mathrm{Rotate}_{rk}(ct;r)$：输入密文 $ct\in R_q^2$ 和旋转密钥 rk，输出 ct 的明文向量旋转 r 个位置后对应的密文 ct'。

HEAAN 支持编码向量，对于 2 的整数幂 $k\leqslant N/2$，基于 cannonical embedding[20]，用映射 $\phi:\mathbb{C}^k\to R$，将 k 个复数打包编码成单个明文多项式，每个复数对应一个明文槽，密文运算对应明文中相应槽位的运算。HEAAN 要求明文插槽的个数为 2 的整数幂，因此对于 n 维复数向量，编码后的明文槽数为 $k=2^{\lceil\log n\rceil}$，多出来的插槽填充 0 即可。对于单个复数，编码时将该数复制到所有的明文插槽中。

本文限制明文空间在实数域中。利用插值点的共轭性，实数向量编码后对应多项式的系数不再是复数，而是实数。由于同态加密只支持整数运算，HEAAN 通过给明文多项式乘上一个缩放因子 p(p 是 2 的整数幂)，再对多项式的系数取整转化为整数多项式，通过调整 p 的大小来保持原始数据的精度。例如，选取分圆多项式 $\phi_8(X)=X^4+1$，$p=2^7$，两个明文槽对应的本原单位根分别设为 $\zeta_8^1=\exp\left(\frac{2\pi i}{8}\right)$，$\zeta_8^5=\exp\left(\frac{10\pi i}{8}\right)$，对于实数向量 $\boldsymbol{z}=(2.3,5.6)$，其对应的实数多项式为 $1.1667\cdot X^3-1.1667\cdot X+3.950$，此时如果直接对多项式系数取整，则会导致解码后结果误差很大，因此先给多项式乘上缩放因子 2^7，则编码算法输出整数明文多项式 $m(X)=149\cdot X^3-149\cdot X+506$，满足 $2^{-7}\cdot(m(\zeta_8^1),m(\zeta_8^5))\approx(2.307,5.599)$，非常接近原始向量。$p$ 越大，解码后的结果精度就越高。

有关 HEAAN 同态加密的技术细节和噪声分析可以参考文献[8]，文献[21-22]实现了支持 HEAAN 的 bootstrapping 技术。文献[23]利用“FullRNS”优化技术，显著提高了 HEAAN 的效率。

3　方案设计

3.1　场景描述

在我们设计的场景中存在两个用户 Owner1 和 Owner2，线性系统的原始数据$\{\boldsymbol{A},\boldsymbol{b}\}$以一定规则分割存储在两个用户上，分别记作$\{\boldsymbol{A}_1,\boldsymbol{b}_1\}$和$\{\boldsymbol{A}_2,\boldsymbol{b}_2\}$，包含各自的敏感信息。

Owner1 和 Owner2 想要集中双方的数据合作求解 $\boldsymbol{Ax}=\boldsymbol{b}$，同时都不想将自己的数据泄露给对方，即我们要求在整个计算过程中，除了最终的目标向量 $\boldsymbol{x}$，双方不能得到任何关于彼此的原始数据以及其他中间结果的有效信息。

我们假设 Owner1 和 Owner2 都是诚实且好奇的实体，他们诚实地履行协议，但是在协议执行过程中尝试观察并获取彼此的数据以及中间结果的有效信息。

3.2　解决方案

本文主要利用同态加密技术保护计算中的数据隐私。Owner1 生成一对用于同态加密的公私钥$\{sk,pk\}$，并且公开自己的公钥 pk。Owner1 和 Owner2 分别将自己的数据$\{\boldsymbol{A}_1,\boldsymbol{b}_1\}$和$\{\boldsymbol{A}_2,\boldsymbol{b}_2\}$用相同的公钥 pk 加密，然后由 Owner2 负责集中双方的加密数据进行整合（具体的数据整合方法要依据实际问题中数据的分布规则来设计，如一种可能的情况为 $\boldsymbol{A}=\boldsymbol{A}_1+\boldsymbol{A}_2,\boldsymbol{b}=\boldsymbol{b}_1+\boldsymbol{b}_2$），处理后得到求解线性系统所需要的数据$\{\boldsymbol{A},\boldsymbol{b}\}$，Owner2 再利用 Gram-Schmidt 正交化技术求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$，得到目标向量 $\boldsymbol{x}$ 的密文，整个过程中 Owner2 负责的运算全部在密文上完成。

对于计算中涉及的除法运算，我们结合同态加密和添加随机扰动的方法设计了一种交互式的安全乘法逆协议，用于打破同态加密方案不支持高效密文除法的瓶颈，同时可以保护计算中的数据隐私。

该方案的实现细节将在第 4 节详细阐述。我们设计的特定场景下的解决方案很容易被扩展到实际应用场景中，典型的有机器学习算法中的回归分析。当数据集不规则（水平或者垂直）分布在两个用户上时，双方想要协作训练一个线性回归模型用于预测，但是都不想将自己数据集的信息暴露给对方，此时就可以将训练任务转化成一个两方参与的安全求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 的问题，利用本文的方案来提供隐私保护。

4　隐私保护线性系统求解方案

本节将阐述方案的实现细节，基于第 3 节中设计的场景以及提出的解决方案，借助同态加密技术设计密文域上的 Gram-Schmidt 正交化算法，通过矩阵分解实现单个以及多个线性系统的安全求解。

4.1　数据打包与加密方式

对于原始数据矩阵$\boldsymbol{A}_{n\times n}$和向量$\boldsymbol{b}_{n\times 1}$，我们基于 HEAAN 的编码特性，为隐私保护线性系统求解方案设计了独特的打包方式，可以在相同的时间损耗下求解单个线性系统和多个线性系统。

4.1.1　单个线性系统

对于单个线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$，我们将原始矩阵$\boldsymbol{A}_{n\times n}=(\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_n)$逐列向量编码，对于列向量$\boldsymbol{a}_i$，首先通过填充 0 将其维度扩充为$2^{\lceil\log n\rceil}$，即：

$$\boldsymbol{a}_i^{\mathrm{T}}=[A_{1i},A_{2i},\cdots,A_{ni},0,\cdots,0]$$

将其逐列向量打包成 n 条密文，定义为：

$$ct_{\boldsymbol{A}}\leftarrow[Enc(\boldsymbol{a}_1),Enc(\boldsymbol{a}_2),\cdots,Enc(\boldsymbol{a}_n)]$$

其表示为$ct_{\boldsymbol{A}}[i]\leftarrow Enc(\boldsymbol{a}_i)_{1\leqslant i\leqslant n}$。向量 $\boldsymbol{b}$ 的编码方式与$\boldsymbol{a}_i$相同，打包成单个密文，定义为$ct_{\boldsymbol{b}}\leftarrow Enc(\boldsymbol{b})$。计算得到的目标向量 $\boldsymbol{x}$ 的每个元素分别存储在单独的密文中，即存储 $\boldsymbol{x}$ 需要 n 条密文，定义为：

$$ct_{\boldsymbol{x}}\leftarrow[Enc(x_1),Enc(x_2),\cdots,Enc(x_n)]$$

其表示为$ct_x[i]\leftarrow Enc(x_i)_{1\leqslant i\leqslant n}$。

由于方案涉及的运算基本全部在密文状态下进行，因此存储整个线性系统需要 $2n+1$ 条密文。我们还需要额外的密文来存储中间结果，即矩阵 $\boldsymbol{C},\boldsymbol{R},\boldsymbol{\Sigma}^2,\boldsymbol{P}$，其中 $\boldsymbol{C}$ 和 $\boldsymbol{R}$ 分别为主对角线元素全部为 1 的上三角矩阵，只需要存储主对角线以上部分的元素，采用逐元素编码，分别需要加密成$\left(\frac{n^2}{2}-n\right)$条密文，定义为：

$$ct_{\boldsymbol{R}}=\begin{bmatrix}1 & Enc(R_{12}) & Enc(R_{13}) & \cdots & Enc(R_{1n})\\ 0 & 1 & Enc(R_{23}) & \cdots & Enc(R_{2n})\\ 0 & 0 & 1 & \cdots & Enc(R_{3n})\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \cdots & 1\end{bmatrix}$$

则 $ct_{\boldsymbol{R}}[i,j]\leftarrow Enc(R_{ij})_{1\leqslant i<j\leqslant n}$，类似地有 $ct_{\boldsymbol{C}}[i,j]\leftarrow Enc(C_{ij})_{1\leqslant i<j\leqslant n}$。$\boldsymbol{\Sigma}^2$为对角矩阵，存储其主对角线的元素也需要 n 条密文，定义为$ct_{\boldsymbol{\Sigma}^2}\leftarrow\left[Enc\left(\frac{1}{\langle\boldsymbol{p}_1,\boldsymbol{p}_1\rangle}\right),Enc\left(\frac{1}{\langle\boldsymbol{p}_2,\boldsymbol{p}_2\rangle}\right),\cdots,Enc\left(\frac{1}{\langle\boldsymbol{p}_n,\boldsymbol{p}_n\rangle}\right)\right]$，表示为$ct_{\boldsymbol{\Sigma}^2}[i]\leftarrow Enc(\Sigma_{ii}^2)_{1\leqslant i\leqslant n}$。$\boldsymbol{P}$ 是 $\boldsymbol{A}$ 正交化后的矩阵，存储 $\boldsymbol{P}$ 需消耗 n 条密文，定义为：

$$ct_{\boldsymbol{P}}\leftarrow[Enc(\boldsymbol{p}_1),Enc(\boldsymbol{p}_2),\cdots,Enc(\boldsymbol{p}_n)]$$

因此共消耗(n^2+3n+1)条密文。通过优化存储空间，具体实现方案时 $\boldsymbol{A},\boldsymbol{P}$ 和 $\boldsymbol{x}$ 的密文共享存储空间，$\boldsymbol{C}$ 和 $\boldsymbol{R}$ 共享存储空间，共需要存储$\left(\frac{n^2}{2}+n+2\right)$条密文。

4.1.2　多个线性系统

通过优化明文打包方式，本文方案可以利用 SIMD 技术，在相同的计算复杂度以及不影响计算结果精度的情况下，并行求解多个不同的线性系统，几乎不需要额外的时间损耗。

对于 HEAAN，我们可以利用的明文插槽数最多为 $N/2$，求解单线性系统时只用到了$2^{\lceil\log n\rceil}$个明文插槽，一般我们选取 $N\gg 2^n$，导致了大量明文槽的浪费，因此我们通过利用更多的明文插槽来实现并行求解多个不同的线性系统。

对于 m 个不同的线性系统：

$$\begin{cases}\boldsymbol{A}^{(1)}\boldsymbol{x}^{(1)}=\boldsymbol{b}^{(1)}\\ \boldsymbol{A}^{(2)}\boldsymbol{x}^{(2)}=\boldsymbol{b}^{(2)}\\ \quad\vdots\\ \boldsymbol{A}^{(m)}\boldsymbol{x}^{(m)}=\boldsymbol{b}^{(m)}\end{cases}\tag{5}$$

本文采用平行编码的方式，在单个密文中平行打包了 m 个实数向量。以 m 个 $\boldsymbol{A}$ 矩阵的编码为例（为了便于表示，假设 m 是 2 的整数幂），用填充 0 的方式将每个 $\boldsymbol{A}$ 矩阵的第 i 列进行扩充，组成$2^{\lceil\log n\rceil}\times m$ 维矩阵：

$$\boldsymbol{\alpha}_i=\begin{bmatrix} A_{1i}^{(1)} & A_{1i}^{(2)} & \cdots & A_{1i}^{(m)} \\ A_{2i}^{(1)} & A_{2i}^{(2)} & \cdots & A_{2i}^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ A_{ni}^{(1)} & A_{ni}^{(2)} & \cdots & A_{ni}^{(m)} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

矩阵$\boldsymbol{\alpha}_i$满足$2^{\lceil \log n \rceil} \times m \leqslant N/2$，然后逐行将矩阵$\boldsymbol{\alpha}_i$打包成单个密文：

$$Enc(\boldsymbol{\alpha}_i) \leftarrow Enc(A_{1i}^{(1)}, \cdots, A_{1i}^{(m)}, A_{2i}^{(1)}, \cdots, A_{2i}^{(m)}, \cdots, A_{ni}^{(1)}, \cdots, A_{ni}^{(m)}, 0, \cdots, 0)$$

因此，对于存储 m 个线性系统的矩阵$\boldsymbol{A}$，仍然只需要 n 条密文。对于 m 个 $\boldsymbol{b}$ 向量，采用与 $\boldsymbol{A}$ 中列向量相同的平行编码，表示成矩阵形式为：

$$\boldsymbol{\beta}=\begin{bmatrix} b_1^{(1)} & b_1^{(2)} & \cdots & b_1^{(m)} \\ b_2^{(1)} & b_2^{(2)} & \cdots & b_2^{(m)} \\ \vdots & \vdots & \ddots & \vdots \\ b_n^{(1)} & b_n^{(2)} & \cdots & b_n^{(m)} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

逐行将矩阵 $\boldsymbol{\beta}$ 打包成单个密文：

$$ct_{\boldsymbol{\beta}} \leftarrow Enc(b_1^{(1)}, \cdots, b_1^{(m)}, b_2^{(1)}, \cdots, b_2^{(m)}, \cdots, b_n^{(1)}, \cdots, b_n^{(m)}, 0, \cdots, 0)$$

与单线性系统的编码方式相比，这种平行编码方式并没有增加额外的密文，只是利用了每条密文中更多的明文插槽，同时打包了多个线性系统的数据。多线性系统编码方式中每条密文中的明文插槽数为$2^{\lceil \log n \rceil} \times m$。

4.2 安全乘法逆协议

由于 Gram-Schmidt 正交化过程涉及 n 次除法运算，而 HEAAN 同态加密技术仅提供了受限的同态乘逆运算，即对于一个实数 a，限制$|1-a| \leqslant 1/2$，才可以利用多项式逼近的方法近似地求其乘逆 $1/a$。该方法的代价较高，且由于本文方案涉及的计算基本全部在密文下完成，明文数据不一定满足其限制条件，因此不能满足本文的应用需求。文献[16]利用 Newton 迭代法在密文上求解 a 的乘逆 $1/a$，其迭代公式为 $x_{n+1}=(2-a\,x_n)x_n$。由于在密文状态下我们无法得知 a 的信息，选择合适的迭代初值x_0以及迭代次数 n 较为困难，该方法的结果的正确性得不到保证。例如，如果 a 是负数，而迭代初值选择正数，则始终得不到正确结果，且迭代过程会导致密文噪声快速膨胀，因此在实际使用中有很大的限制。

为了以尽可能小的代价实现密文域上的除法运算，并且防止原始数据以及中间结果的有效信息被泄露给任意一方，我们结合同态加密和添加随机乘法扰动的技术，设计了一种如算法 1 中描述的交互式安全乘法逆协议。该协议由双方通过交互协同计算乘法逆。首先利用 HEAAN 的特性——支持明文常量直接与密文多项式做乘法，由 Owner2 给原密文添加随机扰动 r，掩盖原始数据，然后将得到的新密文发送给 Owner1 进行解密，Owner1 在明文状态下计算乘逆，再用公钥将结果加密后发送给 Owner2，最后 Owner2 去除结果中的扰动，获得加密状态下的乘法逆。由于 Owner2 始终看到的是密文，而 Owner1 虽然拥有私钥，但其看到的明文含有噪声，因此双方均不会得到原始明文的有效信息，从而防止了计算过程中源数据的隐私泄露。本文将在 5.1 节中分析随机扰动的选取以及协议的安全性。

算法 1 安全乘法逆协议 MultInverse(ct)

输入：$ct \leftarrow Enc(x_1, x_2, \cdots, x_n)$

输出：$ct \leftarrow Enc\left(\frac{1}{x_1}, \frac{1}{x_2}, \cdots, \frac{1}{x_n}\right)$

1. Owner2：
2. 在相邻点间隔为10^{-4}的离散分布$\{1, 1.0001, \cdots, 2^4\}$上均匀随机选取实数 r
3. $ct \leftarrow ReScale(CMult(r \cdot 2^{\log p}, ct))$
4. 将 ct 发送给 Owner1
5. Owner1：
6. 接收来自 Owner2 的密文 ct
7. $(rx_1, rx_2, \cdots, rx_n) \leftarrow Dec(ct)$
8. $ct \leftarrow Enc\left(\frac{1}{rx_1}, \frac{1}{rx_2}, \cdots, \frac{1}{rx_n}\right)$
9. 将 ct 发送给 Owner2
10. Owner2：
11. 接收来自 Owner1 的密文 ct
12. $ct \leftarrow ReScale(CMult(r \cdot 2^{\log p}, ct))$
13. return ct

4.3 密文域上求解线性系统

本节介绍如何基于设计的数据打包方法和安全乘法逆协议来有效求解规模为 n 的线性系统$\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$。

首先本文定义了一种密文域上的向量内积运算，利用 HEAAN 提供的密文旋转操作，通过文献[24-25]中的方法实现了同态内积运算，将两个 n 维明文向量对应的密文相乘后旋转相加，通过 $\log n$ 次迭代得到其内积所对应的密文。我们将该同态内积操作扩展成算法 2，以适用于多个线性系统的求解，对于分别求解单个线性系统和多个线性系统，每次密文旋转的位数不同。

算法 2 同态内积 InnerProd(ct_a, ct_b, m)

输入：向量 $\mathbf{a}, \mathbf{b}$ 的密文$ct_{\mathbf{a}}, ct_{\mathbf{b}}$，线性系统个数 m

输出：内积 $ct_{dot} \leftarrow Enc(\langle \mathbf{a}, \mathbf{b} \rangle)$

1. $ct_{dot} \leftarrow ReScale(Mult(ct_{\mathbf{a}}, ct_{\mathbf{b}}))$
2. $ct_{dot} \leftarrow Add(ct_{dot}, Rotate(ct_{dot}, -2^j \cdot m))$ for $j \leftarrow 0, 1, \cdots, \lceil \log n \rceil - 1$
3. return ct_{dot}

本文利用 HEAAN 提供的同态操作函数来实现本文的方案。首先 Owner1 和 Owner2 各自在本地打包自己的数据，得到$\{\boldsymbol{A}_1, \boldsymbol{b}_1\}$和$\{\boldsymbol{A}_2, \boldsymbol{b}_2\}$，然后通过算法 3 实现密文状态下的 Gram-Schmidt 正交化过程，最后双方利用算法 4 合作完成线性系统的安全求解。

Owner1 生成自己的同态加密公私钥对$\{pk, sk\}$，将公钥 pk 分享给 Owner2。双方分别将自己的数据利用 pk 加密后，Owner2 集中两部分数据进行整合（实际中按照数据的不同分布规则有不同的整合方式，为便于表述，算法中定义为 Gather 操作），得到矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 的密文。在 Owner1 的协助下，Owner2 开始在密文状态下求解目标向量 $\boldsymbol{x}=\boldsymbol{R} \cdot \boldsymbol{\Sigma}^2 \cdot \boldsymbol{P}^{\mathrm{T}} \cdot \boldsymbol{b}$。Owner2 先调用算法 3 实现密文状态下的 Gram-Schmidt 正交化过程，获得矩阵 $\boldsymbol{R}, \boldsymbol{\Sigma}^2, \boldsymbol{P}$ 的密文，该过程需要

执行 n 次安全乘法逆协议，即 Owner1 和 Owner2 交互 n 次，单次通信量为 2 条密文；之后 Owner2 进行同态计算得到目标向量 $\boldsymbol{x}$ 的密文，为了消耗尽可能少的乘法层数，我们的运算顺序为 $\boldsymbol{R}\cdot(\boldsymbol{\Sigma}^2\cdot(\boldsymbol{P}^{\mathrm{T}}\cdot\boldsymbol{b}))$。最后 Owner2 将目标密文发送给 Owner1 进行解密，Owner1 再把解密结果共享给 Owner2，以完成整个线性系统的求解。

本文构造的求解方案同时适用于单个线性系统和多个线性系统。对于多个线性系统，不需要改变算法的结构，仅需要在数据打包阶段利用平行打包方式打包原始数据即可。

算法 3　同态 Gram-Schmidt 正交化 GSO(ct_A, m)

输入：密文$\mathrm{ct}_{\mathbf{A}}$，线性系统个数 m

输出：$\mathrm{ct}_{\mathbf{R}}$，$\mathrm{ct}_{\boldsymbol{\Sigma}^2}$，$\mathrm{ct}_{\mathbf{P}}$

1. for i←1 to n do
2.　$\mathrm{ct}_{\mathbf{P}}[i]\leftarrow\mathrm{ct}_{\mathbf{A}}[i]$
3.　for j←1 to i−1 do
4.　　$\mathrm{ct}_{\mathbf{C}}[j,i]\leftarrow\mathrm{InnerProd}(\mathrm{ct}_{\mathbf{P}}[j],\mathrm{ct}_{\mathbf{A}}[i],m)$
5.　　$\mathrm{ct}_{\mathrm{temp}}\leftarrow\mathrm{ReScale}(\mathrm{Mult}(\mathrm{ct}_{\mathbf{C}}[j,i],\mathrm{ct}_{\mathbf{P}}[j]))$
6.　　$\mathrm{ct}_{\mathbf{P}}[i]\leftarrow\mathrm{Add}(\mathrm{ct}_{\mathbf{P}}[i],\mathrm{Negate}(\mathrm{ct}_{\mathrm{temp}}))$
7.　end
8.　$\mathrm{ct}_{\boldsymbol{\Sigma}^2}[i]\leftarrow\mathrm{InnerProd}(\mathrm{ct}_{\mathbf{P}}[i],\mathrm{ct}_{\mathbf{P}}[i],m)$
9.　执行算法 1 计算乘法逆：
　$\mathrm{ct}_{\boldsymbol{\Sigma}^2}[i]\leftarrow\mathrm{MultInverse}(\mathrm{ct}_{\boldsymbol{\Sigma}^2}[i])$
10.　for j←1 to i−1 do
11.　　$\mathrm{ct}_{\mathbf{R}}[j,i]\leftarrow\mathrm{Negate}(\mathrm{ct}_{\mathbf{C}}[j,i])$
12.　　for k←j+1 to i−1 do
13.　　　$\mathrm{ct}_{\mathrm{te}}\leftarrow\mathrm{ReScale}(\mathrm{Mult}(\mathrm{ct}_{\mathbf{R}}[j,k],\mathrm{ct}_{\mathbf{C}}[k,i]))$
14.　　　$\mathrm{ct}_{\mathbf{R}}[j,i]\leftarrow\mathrm{Add}(\mathrm{ct}_{\mathbf{R}}[j,i],\mathrm{Negate}(\mathrm{ct}_{\mathrm{te}}))$
15.　　end
16.　end
17. end

算法 4　线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 安全求解

输入：$\{\mathbf{A}_1,\mathbf{b}_1\}$，$\{\mathbf{A}_2,\mathbf{b}_2\}$，线性系统个数 m

输出：目标向量 **x**

1. Owner1：
2. $\{\mathrm{pk},\mathrm{sk}\}\leftarrow\mathrm{KeyGen}(1^{80})$
3. $\mathrm{ct}_{\mathbf{A}_1}$，$\mathrm{ct}_{\mathbf{b}_1}\leftarrow\mathrm{Enc}_{\mathrm{pk}}(\{\mathbf{A}_1,\mathbf{b}_1\})$
4. send $\mathrm{ct}_{\mathbf{A}_1}$，$\mathrm{ct}_{\mathbf{b}_1}$，pk to Owner2
5. Owner2：
6. $\mathrm{ct}_{\mathbf{A}_2}$，$\mathrm{ct}_{\mathbf{b}_2}\leftarrow\mathrm{Enc}_{\mathrm{pk}}(\{\mathbf{A}_2,\mathbf{b}_2\})$
7. $\mathrm{ct}_{\mathbf{A}}$，$\mathrm{ct}_{\mathbf{b}}\leftarrow\mathrm{Gather}(\mathrm{ct}_{\mathbf{A}_1},\mathrm{ct}_{\mathbf{A}_2},\mathrm{ct}_{\mathbf{b}_1},\mathrm{ct}_{\mathbf{b}_2})$
8. 调用算法 3：$\mathrm{ct}_{\mathbf{R}}$，$\mathrm{ct}_{\boldsymbol{\Sigma}^2}$，$\mathrm{ct}_{\mathbf{P}}\leftarrow\mathrm{GSO}(\mathrm{ct}_{\mathbf{A}},m)$
9. for i←1 to n do
10.　$\mathrm{ct}_{\mathbf{x}}[i]\leftarrow\mathrm{InnerProd}(\mathrm{ct}_{\mathbf{P}}[i],\mathrm{ct}_{\mathbf{b}},m)$
11.　$\mathrm{ct}_{\mathbf{x}}[i]\leftarrow\mathrm{ReScale}(\mathrm{Mult}(\mathrm{ct}_{\mathbf{x}}[i],\mathrm{ct}_{\boldsymbol{\Sigma}^2}[i]))$
12. end
13. for i←1 to n do
14.　for j←1 to i−1 do
15.　　$\mathrm{ct}_{\mathrm{temp}}\leftarrow\mathrm{ReScale}(\mathrm{Mult}(\mathrm{ct}_{\mathbf{R}}[i,j],\mathrm{ct}_{\mathbf{P}}[j]))$
16.　　$\mathrm{ct}_{\mathbf{x}}[i]\leftarrow\mathrm{Add}(\mathrm{ct}_{\mathbf{x}}[i],\mathrm{ct}_{\mathrm{temp}})$
17.　end
18. end
19. send $\mathrm{ct}_{\mathbf{x}}$ to Owner1
20. Owner1：
21. $\mathbf{x}\leftarrow\mathrm{Dec}_{\mathrm{sk}}(\mathrm{ct}_{\mathbf{x}})$
22. send x to Owner2
23. return **x**

5　方案分析

5.1　安全性分析

在本文的方案中，两个用户 Owner1 和 Owner2 都是半诚实的，因为双方的目的都是计算获得正确的目标向量 $\boldsymbol{x}$，所以不存在恶意的 Owner1 和 Owner2。

在整个方案中，Owner2 的计算任务全部在密文状态下进行，其看到的始终是密文，原始数据的隐私性由同态加密技术的安全性保证。HEAAN 的安全性基于 RLWE 问题的困难性，其安全性分析可以参考文献[20]，本文方案中选择安全参数 $\lambda=80$ 来保证 80 bit 的安全强度。

算法 2 的安全乘法逆协议中，计算的中间结果对于 Owner1 的隐私性由随机选取的乘法扰动保证。本文假设整个方案计算中涉及的矩阵和向量中的元素 $h\in[-2^{16},2^{16}]$，基本可以满足一般的矩阵运算。在相邻点间隔为10^{-4}的离散分布$\{1,1.0001,\cdots,2^4\}$上均匀选取随机扰动 r，则明文消息空间为 $r\cdot h\in[-2^{20},2^{20}]$。在安全乘法逆协议中，Owner1 猜出随机扰动 r 的概率为 1/150000。由于算法 3 每轮迭代中随机扰动 r 掩盖的是 $\boldsymbol{\Sigma}$ 矩阵的信息，即使 Owner1 以$(1/150000)^n$的概率猜对所有随机扰动 r，也只能得到 $\boldsymbol{\Sigma}$ 矩阵，其中包含 $\boldsymbol{P}$ 矩阵列向量的长度信息。仅通过正交化后 $\boldsymbol{P}$ 矩阵列向量的长度信息，很难得到其他关于原始矩阵 $\boldsymbol{A}$ 的有效信息。因此，从实际应用的角度来看，本文设计的乘法逆协议的安全性是足够的。

5.2　参数分析

HEAAN 提供的层次型同态加密技术，在每次同态乘法后伴随一次重缩放(ReScale)操作，密文模会随之减少 $\log p$ 比特。我们用参数 Q 表示一条新鲜密文的模数，则 $Q=p^L\cdot Q_0$，L 为 HEAAN 支持的最大乘法层数，Q_0 表示输出密文的模数。在整个求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 的方案中，对于 n 维矩阵 $\boldsymbol{A}$，总共消耗 $2n+3$ 层同态乘法，密文模减少了 $(2n+3)\cdot\log p$ bit，因此参数 Q 满足：

$$\log Q=(2n+3)\cdot\log p+\log Q_0 \tag{6}$$

为了保证输出结果的精度，必须满足$Q_0\gg p$。

根据文献[8,26-27]得到的安全条件：

$$N\geqslant\frac{\lambda+110}{7.2}\log(P\cdot Q) \tag{7}$$

选取分圆多项式的维度 N 来保证 λ bit 的安全强度，其中 $P\approx Q$。由于随着初始矩阵维度的增大，施密特正交化的迭代次数增多，计算消耗的乘法层数也会越来越多，导致密文中噪声不断增大，使得输出结果的精度损失也越多。由文献[8]中 HEAAN 的噪声分析可知，每次乘法至多损失 1 bit 的精度，同态加密方案中的参数 p 可以看作是输入精度，p 越大，输入的精度就越高，因此在实验过程中根据经验以及实际需求选取合适的参数 p 来保证输出结果的精度。

5.3　复杂度分析

对于同态加密技术 HEANN，算法 4 中同态求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 共消耗了 $2n+3$ 层同态乘法，需要 $O(n^3)$次密文乘法，单次密文乘法对应次数为 N 的多项式乘法，基于快速

傅里叶变换(FFT)的多项式乘法的复杂度为 $N\log N$,最大的密文模为 Q,即多项式系数规模最大为 $\log Q$ bit,则整个求解过程中的计算复杂度为 $O(n^5 \cdot \lambda \cdot \log\lambda \cdot p^{2n+3} \cdot \log^2 p)$。双方运行安全乘法逆协议,在交互中需要发送 $2n$ 条密文,且密文模数会随着 Gram-Schmidt 正交化的迭代过程不断约减,理论通信量为 $nN(2\log Q-(n+1)\log p)$ bit。

6 实验结果与评价

本文方案基于 HEAAN 开源库和 NTL 库[28]使用 C++ 实现。实验环境如下:CPU 为 Intel Core i9-9900K,主频为 3.60 GHz,8 核 16 线程,内存为 32 GB 的计算机;操作系统为 Ubuntu 18.04。

本文定义同态运算下得到的目标向量 $\boldsymbol{x}$ 相对于明文状态下求解的相对误差为:

$$error=\frac{\|\boldsymbol{x}'-\boldsymbol{x}\|_2}{\|\boldsymbol{x}'\|_2} \tag{8}$$

其中,$\boldsymbol{x}'$表示明文状态下线性系统的解。实验中我们随机生成不同维度的矩阵 $\boldsymbol{A}$ 和向量 $\boldsymbol{b}$ 作为输入数据,其元素都是从区间[0,1]中均匀随机选取的,设置 $\log Q_0=\log p+10$。实验过程中选取不同的 $\log p$ 来保证输出结果的精度。

6.1 结果分析

表 1 列出了求解不同规模单线性系统时的参数设置以及实验结果。随着原始数据维度的增加,同态计算中消耗的乘法层数线性增长,导致密文模不断增大。为了解决噪声累积对计算结果的影响,实验中根据经验适当增大参数 $\log p$ 来保持计算结果的精度。实验结果表明,线性系统的维度不超过一定规模($n\leqslant 17$)时,利用本文的隐私保护方案可以高效地求解该系统,平均时间损耗不超过 6.31 min,与明文状态下求得的结果相比,相对误差仅10^{-4},这样的误差对于实际中的绝大多数场景来说是完全可以接受的;在计算机中实现本文方案时,由于 NTL 数论库底层数据结构的设计特性,除了计算数值外,还需要额外存储相应的数据结构,导致实验中通信损耗的实际值高于理论值,如 $n=9$ 时,实际通信量为 175.1 MB,而理论通信量仅为 34.1 MB,实际中可以通过仅传输必要的二进制计算数值来降低实际通信量以达到理论值;且在网络畅通、带宽和内存充足的环境下,这样的通信代价远小于 bootstrapping 以及文献[9,16]中实现密文除法付出的代价。因此,可以验证在本文设计的场景下,两方安全高效地求解一定规模的线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 是可行的。

表 1 单个线性系统的实验结果

Table 1 Experiment results of single linear system

security parameter $\lambda=80$						
Dimension of matrix	N	Consumed levels	$\log Q$	$\log p$	Total time/min	Relative error
2	2^{14}	7	217	25	0.004	10^{-6}
5	2^{15}	13	373	25	0.09	10^{-4}
9		21	620	27	0.35	10^{-4}
13	2^{16}	29	939	30	2.78	10^{-4}
15		33	1063	30	4.05	10^{-4}
17		37	1240	32	6.31	10^{-4}
18	2^{17}	39	1329	32	17.73	10^{-5}
21		45	1665	35	31.88	10^{-6}
26		55	1985	35	71.83	10^{-5}

表 2 列出了求解维度为 16 的多个线性系统时的时间损耗,可以看到,与求解单个线性系统相比,采用平行编码的方式同时打包多个线性系统求解的时间损耗几乎相同,本文方案求解 1024 个线性系统耗时 289.984 s,平均每个系统仅分摊 0.283 s,如此来看效率是非常可观的,可以很好地满足用户想要同时解决多个问题的需求,进一步扩展了方案的可用性。

表 2 $n=16$, $\log p=30$, $N=16$, $\log Q=1100$ 时多线性系统的实验结果

Table 2 Experiment results of multiple linear systems when $n=16$, $\log p=30$, $N=16$, $\log Q=1100$

Number of linear systems	Total time/s
1	286.007
2	286.967
16	287.020
128	288.671
512	288.800
1024	289.984

6.2 方案比较

图 1 给出了本文方案与文献[16]中 Zhang 等提出的方案的效率对比结果。本文方案在矩阵维度小于 11 时效率更好,在矩阵维度超过 11 时,文献[16]中的方案更高效,两种方案都可以高效地解决规模不超过 17 的该类问题;但是文献[16]中的问题在于,如果完全在密文状态下使用 Newton 迭代法计算除法,需要消耗大量同态乘法层数,理论上可以求解的问题规模远达不到 17,要解决这个问题,他们需要增加额外的交互,而这违背了使用 Newton 迭代法的初衷,使用 Newton 迭代法计算除法正是为了避免交互。本文方案使用的安全乘法逆协议则不存在这样的问题。

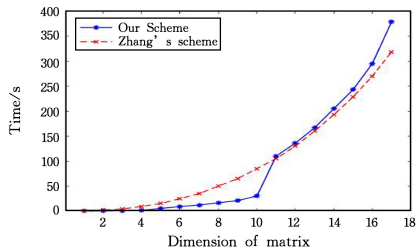

图 1 方案运行时间的比较

Fig. 1 Comparison of running time between schemes

使用 Newton 迭代法的另一个缺点是,在加密状态下由于缺乏足够的明文信息来指导迭代初值以及迭代次数的选择,Newton 迭代法的收敛性和精度得不到保证。通过对文献[16]中的结果的损失函数进行转换,评估得到其计算结果的相对误差不大于10^{-2},而本文使用安全乘法逆协议实现的除法运算几乎没有精度损失,最终结果的相对误差不超过10^{-4}。因此,本文方案的结果准确度更好,并且支持并行求解多个该类问题,在设计隐私保护数据挖掘算法时更具优势。

结束语 本文基于同态加密技术,为两方参与安全求解线性系统 $\boldsymbol{Ax}=\boldsymbol{b}$ 提出了一种可行的解决方案,并对该方案的效率、安全性以及准确性进行了评估。在求解小规模线性系统时,本文方案表现出了良好的性能,以可接受的通信代价实现了安全乘法逆运算,有效地保护了数据的隐私,且有效利用 SIMD 技术并行多个线性系统,基本满足所给场景下的实际应用。

在未来的研究工作中，我们将进一步优化方案，可以通过更多的交互来提高方案在求解更大规模线性系统时的表现力，同时权衡通信代价与同态密文计算的时间损耗，使方案的整体效率达到最优；寻求更好的数据打包方式以优化存储空间，进一步尝试将该方案应用到实际的机器学习算法中，同时解决双方在共享计算结果时的零知识证明问题。

参 考 文 献

[1] YANG Q，LIU Y，CHEN T J，et al. Federated Machine Learning：Concept and Applications[J]. ACM Transaction on Intelligent Systems and Technology，2019，10(2)：19.

[2] GENTRY C. Fully Homomorphic Encryption Using Ideal Lattices[C]// Proceedings of the 2009 Acm Symposium on Theory of Computing. 2009：169-178.

[3] YAO A C C. How to generate and exchange secrets[C]// Proceedings of the 27th Annual Symposium on Foundations of Computer Science. IEEE Computer Society，1986：162-167.

[4] DWORK C. Differential privacy[M]// Automata，Languages and Programming，Pt 2. Berlin：Springer，2006：1-12.

[5] BRAKERSKI Z，VAIKUNTANATHAN V. Efficient Fully Homomorphic Encryption from (Standard) LWE[C]// 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science. 2011：97-106.

[6] BRAKERSKI Z，GENTRY C，VAIKUNTANATHAN V. (Leveled) Fully Homomorphic Encryption without Bootstrapping [J]. ACM Transactions on Computation Theory，2014，6(3)：1-36.

[7] FAN J，VERCAUTEREN F. Somewhat practical fully homomorphic encryption[J]. IACR Cryptology ePrint Archive，2012 (144)：1-19.

[8] CHEON J H，KIM A，KIM M，et al. Homomorphic Encryption for Arithmetic of Approximate Numbers[M]// Advances in Cryptology-Asiacrypt 2017，Pt I. Cham：Springer. 2017：409-437.

[9] XU C，CHEN J，WU W，et al. Homomorphically Encrypted Arithmetic Operations Over the Integer Ring[M]// Information Security Practice and Experience (ISPEC 2016). Cham：Springer，2016：167-181.

[10] NIKOLAENKO V，WEINSBERG U，IOANNIDIS S，et al. Privacy-Preserving Ridge Regression on Hundreds of Millions of Records. [C]// 2013 IEEE Symposium on Security and Privacy. 2013：334-348.

[11] HU S，WANG Q，WANG J，et al. Securing Fast Learning! Ridge Regression over Encrypted Big Data[C]// 2016 IEEE Trustcom/BigDataSE/ISPA. 2016：19-26.

[12] CHEN Y R，REZAPOUR A，TZENG W G. Privacy-preserving ridge regression on distributed data[J]. Information Sciences，2018，451：34-49.

[13] GIACOMELLI I，JHA S，JOYE M，et al. Privacy-Preserving Ridge Regression with only Linearly-Homomorphic Encryption [M]// Applied Cryptography and Network Security. Cham：Springer，2018：243-261.

[14] QIU G，GUI X，ZHAO Y. Privacy-Preserving Linear Regression on Distributed Data by Homomorphic Encryption and Data Masking[J]. IEEE Access，2020，8：107601-107613.

[15] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes [C]// Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques. Springer-Verlag，1999：223-238.

[16] ZHANG Y，ZHENG P，LUO W，et al. Privacy-Preserving Outsourcing Computation of QR Decomposition in the Encrypted Domain[C]// 2019 18th Ieee International Conference on Trust，Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering. 2019：389-396.

[17] STEWART G，MAHAJAN A. Matrix Algorithms，Volume II：Eigensystems[J]. Applied Mechanics Reviews，2003，56(1)：B2.

[18] SMART N P，VERCAUTEREN F. Fully homomorphic SIMD operations[J]. Designs Codes and Cryptography，2014，71(1)：57-81.

[19] CHEON J H，KIM A，KIM M，et al. Implementation of HEAAN [OL]. https：//github. com/kimandrik/HEAAN.

[20] LYNBASHEVSKY V，PEIKERT C，REGEV O. On Ideal Lattices and Learning with Errors over Rings[M]// Advances in Cryptology—Eurocrypt 2010. Berlin：Springer，2010：1-23.

[21] CHEON J H，HAN K，KIM A，et al. Bootstrapping for Approximate Homomorphic Encryption[M]// Advances in Cryptology—Eurocrypt 2018，Pt I. Cham：Springer，2018：360-384.

[22] CHEN H，CHILLOTTI I，SONG Y. Improved Bootstrapping for Approximate Homomorphic Encryption [M] // Advances in Cryptology — Eurocrypt 2019，Pt Ii. Cham：Springer，2019：34-54.

[23] CHEON J H，HAN K，KIM A，et al. A Full RNS Variant of Approximate Homomorphic Encryption[C]// Selected Areas in Cryptography—SAC 2018. Cham：Springer International Publishing，2019：347-368.

[24] KIM A，SONG Y，KIM M，et al. Logistic regression model training based on the approximate homomorphic encryption[J]. BMC Medical Genomics，2018，11(4)：23-31.

[25] KIM M，SONG Y，WANG S，et al. Secure Logistic Regression Based on Homomorphic Encryption：Design and Evaluation[J]. Jmir Medical Informatics，2018，6(2)：245-255.

[26] GENTRY C，HALEVI S，SMART N P. Homomorphic Evaluation of the AES Circuit[M]// Advances in Cryptology—Crypto 2012. Berlin：Springer，2012：850-867.

[27] LINDNER R，PEIKERT C. Better Key Sizes (and Attacks) for LWE-Based Encryption[M]// Topics in Cryptology—CT-RSA 2011. Berlin：Springer，2011：319-339.

[28] VICTOR S. NTL：A Library for doing Number Theory[OL]. https：//www. shoup. net/ntl.

LYU You，born in 1996，postgraduate. His main research interests include homomorphic encryption and information security.

WU Wen-yuan，born in 1976，Ph.D，professor. His main research interests include lattice based cryptography，automated reasoning and symbolic computation.

（责任编辑：柯颖）