软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2021,32(11):3596−3605 [doi: 10.13328/j.cnki.jos.006032]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一种基于 MLWE 的同态内积方案*

柯程松[1,2], 吴文渊[1], 冯 勇[1]

[1](自动推理与认知重庆市重点实验室(中国科学院 重庆绿色智能技术研究院),重庆 400714)

[2](重庆邮电大学 计算机科学与技术学院,重庆 400065)

通讯作者: 吴文渊, E-mail: wuwenyuan@cigit.ac.cn

摘 要: 同态内积在安全多方几何计算、隐私数据挖掘、外包计算、可排序的密文检索等场景有广泛的应用.但现有的同态内积计算方案大多是基于 RLWE 的全同态加密方案,普遍存在效率不高的问题.在柯程松等人提出的基于 MLWE 的低膨胀率加密算法基础上,提出了一种同态内积方案.首先给出了密文空间上的张量积运算⊗,该密文空间上的运算对应明文空间上的整数向量内积运算;然后分析了方案的正确性与安全性;最后给出了两种优化的加密参数,对应计算两种不同大小的整数向量同态内积的应用场景.通过 C++与大整数计算库 NTL 实现了该方案.对比其他同态加密方案,该方案能够比较高效地计算整数向量的同态内积.

关键词: MLWE;同态内积;安全多方计算

中图法分类号: TP309

中文引用格式: 柯程松,吴文渊,冯勇.一种基于 MLWE 的同态内积方案.软件学报,2021,32(11):3596−3605. http://www.jos.org.cn/1000-9825/6032.htm

英文引用格式: Ke CS, Wu WY, Feng Y. MLWE-based homomorphic inner product scheme. Ruan Jian Xue Bao/Journal of Software, 2021,32(11):3596−3605 (in Chinese). http://www.jos.org.cn/1000-9825/6032.htm

MLWE-based Homomorphic Inner Product Scheme

KE Cheng-Song[1,2], WU Wen-Yuan[1], FENG Yong[1]

[1](Chongqing Key Laboratory of Automated Reasoning and Cognition (Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences), Chongqing 400714, China)

[2](College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China)

Abstract: The homomorphic inner product has a wide range of applications such as secure multi-geometry calculation, private data mining, outsourced computing, and sortable ciphertext retrieval. However, the existing schemes for calculating the homomorphism inner product are mostly based on FHE by RLWE with low efficiency. With MLWE, this study proposes a homomorphic inner product scheme by using a low expansion rate encryption algorithm proposed by Ke, *et al*. Firstly, the tensor product operation in the cipher space is given, which corresponds to the integer vector product operation in the plaintext space. Then, the correctness and security of the scheme are analyzed. At last, two sets of optimized encryption parameters are given, corresponding to the different application scenarios of homomorphic inner product. The scheme of this study is implemented by C++ and the large integer computation library NTL. Compared with other homomorphic encryption schemes, this scheme can efficiently calculate the homomorphism inner products of integer vectors.

Key words: MLWE; homomorphic inner product; secure multi-party computation

安全多方计算最早由 Yao[1]提出,指的是解决一组互不信任的参与方之间保护隐私的协同计算问题.随着云计算与大数据技术的广泛应用,越来越多的场景需要安全高效的计算两方所输入向量的内积,如安全多方几何

* 基金项目: 国家自然科学基金(11671377); 重庆市院士专项(cstc2017zdcy-yszxX0011, cstc2018jcyj-yszxX0002)

Foundation item: National Natural Science Foundation of China (11671377); Research Project of Chongqing Science and Technology Commission (cstc2017zdcy-yszxX0011, cstc2018jcyj-yszxX0002)

收稿时间: 2018-07-02; 修改时间: 2019-01-06, 2019-10-08; 采用时间: 2020-02-28

计算、隐私数据挖掘、外包计算、可排序的密文检索等场景.安全计算两个整数向量的内积,是安全多方计算的重要技术之一.

全同态加密允许对密文进行任意复杂的运算,而不需要解密密文,自然可以应用于安全计算两个整数向量的内积.设计全同态加密方案,一直是密码学界的研究热点.2009 年,Gentry[2]基于理想格构造了第一个全同态加密方案,由于需要使用压缩电路、自举等技术,该方案的效率很低,难以应用于实际.2012 年,Brakerski 等人[3]提出了一种基于 RLWE(ring learning with errors)问题[4]的层级全同态加密方案 BGV,即分成一层层的电路进行同态计算,计算效率有所提高. 2014 年,Halevi 等人根据 BGV 方案构造了第一个全同态计算库 HElib[5],该计算库是目前公认效率最高能够进行全同态加密的计算库.2016 年,Xu 等人[6]对 HElib 进行了优化,可完成两个整数的密文四则运算,使用 SIMD 技术将一个明文向量的所有坐标打包到一个密文中,尽管进行了优化,要计算两个 256 维 8bits 整数向量的同态内积,使用 SIMD 技术仍然需要计算一次乘法和 8 次加法共约 20s 左右的时间.这是由于基于 RLWE 构造同态加密方案,安全性要取较大的安全参数.文献[7]利用基于理想格的全同态加密来安全计算内积,并应用于身份认证.但是该方案的计算效率并不高,如对两个 2 048 维的 bit 向量,为了安全计算内积,其预计计算的时间需要 38s;并且该方案只能计算 0-1 向量,使用场景有局限性.除了使用全同态加密,也有学者通过其他方法计算整数向量的同态内积.2010 年,Dijk 等人[8]提出了基于整数的同态加密算法,虽然效率较高,但安全性是基于近似 GCD 问题,已被证明是不安全的[9].2014 年,Zhou 等人[10]提出了一种计算整数同态内积的方法,效率较高,但也存在安全性问题[11].总之,目前已有的方法都难以安全高效地计算两个整数向量的同态内积.

柯程松等人[12,13]于 2018 年对 Bos 等人的 Kyber 密钥封装算法[14]进行改进,提出了一种基于 MLWE(module learning with errors)问题[15]的低膨胀率加密算法,加密效率高于基于 RLWE 问题构造的加密算法.这是由于 RLWE 问题只能规约到理想格上的困难问题,加密参数会取得很大,所以加密效率较低.但 MLWE 问题能归约到模格上的困难问题,模格是一般格与理想格的推广,使能够在保证同等安全性的前提下选取更小的加密参数,所以柯程松等人提出的加密算法效率很高,并且该方案本身保证了具有能够加密多位的整数.在此基础上,本文构造一种基于 MLWE 的安全高效的同态内积方案.

1 本文贡献与文章结构

1.1 本文贡献

本文构造了一种基于 MLWE 的安全高效的同态内积方案,贡献如下:

(1) 基于 BGV 方案的思想,给出了基于 MLWE 问题构造的同态内积方案的密文空间上运算⊗,该密文空间上的运算对应明文空间上的整数向量内积运算.

(2) 分析了本文基于 MLWE 问题构造同态内积算法的正确性与安全性.

(3) 针对计算两种不同大小的整数向量同态内积的应用场景,给出了两种优化的加密参数.

(4) 通过 C++与大整数计算库 NTL 实现了本文方案.对比其他同态加密方案,该方案能够高效地计算整数向量的同态内积.

1.2 文章结构

本文第 2 节将介绍本文中出现的一些符号以及预备知识.第 3 节首先给出本文的同态内积方案,其中定义了密文空间上的张量运算⊗;接着描述如何通过多项式乘法计算向量内积.其次给出了通过密文空间上的张量运算⊗计算同态内积的过程;然后分析了方案的正确性.针对两种不同大小的整数向量同态内积的应用场景,给出了两种优化的加密参数.最后证明了本文方案的 IND-CPA 安全性.第 4 节分析本文同态内积方案的计算复杂度,并与其他几个方案做同态内积的效率进行对比.第 5 节总结全文,并对下一步值得关注的研究方向和应用场景进行简单讨论.

2 预备知识

本节将给出本文算法中会出现的符号以及一些相关概念.该部分内容会用到文献[12,14]中的相关技术,为了保证读者阅读的方便,在本文中采用的记号与文献[12]一致.

2.1 基础符号

令 $R=\mathbb{Z}[x]/f(x)$为整系数多项式环 $Z[x]$模 $f(x)$的商环,$R_q=\mathbb{Z}_q[x]/f(x)$表示 R 的系数再模 q.其中,如果未作特殊说明,$f(x)=x^n+1$,n 为 f 的次数.

当 q 为大于 2 的素数,令 mod q 表示模 q 的取值范围是$[-(q-1)/2,(q-1)/2]$,用 $\mathrm{mod}^+ q$ 表示模 q 的取值范围是$[0,q-1]$,类似地,$\mathrm{mod}^- q$ 的范围是$[-(q-1),0]$.

对 $x\in\mathbb{R}$,$round(x)$表示通常的四舍五入,$\lceil x\rceil$为向上取整,$\lfloor x\rfloor$为向下取整.

对于 R_q 上的多项式 $u(x)=u_0+u_1x+\ldots+u_{n-1}x^{n-1}$,简单记为 u,u 的无穷范数为$\|u\|_\infty=\max\{|u_j|\}$.对于 k 个多项式$(p_1,p_2,...,p_n)\in R_q^k$ 组成的向量 $\boldsymbol{p}$ 的无穷范数定义为所有分量无穷范数中最大的一个,即为$\|\boldsymbol{p}\|_\infty=\max\|p_i\|_\infty$.如未作特殊说明,通常使用$\|\cdot\|$简略表示无穷范数.

如果 $\boldsymbol{A}$ 为集合,令 $a\leftarrow\boldsymbol{A}$ 表示从集合 $\boldsymbol{A}$ 均匀选取元素 a;如果 $\boldsymbol{A}$ 为分布,则表示依分布 $\boldsymbol{A}$ 选取元素 a.

本文的 log 均表示以 2 为底的对数.

令 Pr[·]表示概率,$negl$ 表示概率可忽略不计.

令$\{a,b\}^n$ 表示一个 n 维整数向量,其中,向量元素大于等于 a 且小于等于 b.

令两个 n 维整数向量分别为 $\boldsymbol{a}=(a_0,\ldots,a_{n-1})$和 $\boldsymbol{b}=(b_0,\ldots,b_{n-1})$,令 $\boldsymbol{a}\cdot\boldsymbol{b}=\sum_{i=0}^{n-1}a_ib_i$,也就是向量 $\boldsymbol{a}$ 与 $\boldsymbol{b}$ 的内积.

2.2 同态内积

本节将给出同态内积的详细概念.同态内积指的是在加密的状态下计算两个整数向量的内积,即令两个 n 维整数向量 $\boldsymbol{a}=(a_0,\ldots,a_{n-1})$和 $\boldsymbol{b}=(b_0,\ldots,b_{n-1})$,将两个整数向量分别作为环 R_q 上多项式的系数得到明文 $m_1=(a_0+a_1x+\ldots+a_{n-1}x^{n-1})\in R_q$ 和 $m_2=(b_0+b_1x+\ldots+b_{n-1}x^{n-1})\in R_q$.设加密操作为 $Enc(\cdot)$,解密操作为 $Dec(\cdot)$,加密 m_1 与 m_2 分别得到密文 $c_1=Enc(m_1)$与 $c_2=Enc(m_2)$.令 $c_3=c_1\otimes c_2$,其中,$\otimes$是密文空间上的某种运算.如果能够通过 c_3 与私钥 sk 解密得到 $\boldsymbol{a}\cdot\boldsymbol{b}$,则我们将具有这种性质的加密方案称作同态内积方案.本文主要的贡献即基于文献[12]实现了一种高效的同态内积方案.

2.3 中心二项分布

本文采用与文献[12]中同样的中心二项分布β_η作为加密方案中的噪声分布,定义如下:

均匀随机选取$(a_1,\ldots,a_\eta,b_1,\ldots,b_\eta)\leftarrow\{0,1\}^{2\eta}$,计算并输出 $\sum_{j=1}^{\eta}(a_j-b_j)$.

我们用 $u\leftarrow\beta_\eta$表示各系数依中心二项分布β_η独立选取,得到 R_q 中的一个多项式 u.

从β_η取样 k 个独立并满足β_η分布的多项式构成向量 $\boldsymbol{v}$,则记作 $\boldsymbol{v}\leftarrow\beta_\eta^k$.

2.4 压缩技术(compress)与解压缩技术(decompress)

为了减小公钥与密文的大小,本文和文献[12]一样采用了文献[14]中的压缩技术(compress)与解压缩技术(decompress).相应地,这会在密文中引入误差.为了保证解密的正确性,将在后面章节分析如何选择合适的参数.

定义函数 $Compress_q(x,d)$:输入 $x\in Z_q$,$d<\lceil\log q\rceil$,输出 $y=round((2^d/q)\cdot x)\ \mathrm{mod}^+\ 2^d$.

定义函数 $Decompress_q(y,d)$:输入 $y=Compress_q(x,d)$,输出 $x'=round((q/2^d)\cdot y)$.

不难得到:对于整数 $x\in Z_q$,先压缩再解压后产生的误差满足:$|x-x'|\leqslant round(q/2^{d+1})$.

2.5 MLWE问题

和文献[12]一样,本文加密方案的安全性同样基于模容错学习问题 MLWE[15].设安全参数为λ,$f(x)=x^d+1$,其

次数 $d=d(\lambda)$为 2 的幂,模数 $q=q(\lambda)\geqslant 2$,通常为素数.与前面记号相同,定义多项式环 $R=\mathbb{Z}[x]/f(x)$,$R_q=\mathbb{Z}_q[x]/f(x)$.令 $\chi=\chi(\lambda)$是 R_q 上的一个分布.令 k 为模的维数,本文取 $k=2$ 为例,$MLWE_{n,q,\chi,k}$ 问题即区分以下两个分布:

- 第 1 个分布:$\left(\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix},\begin{bmatrix}b_1\\ b_2\end{bmatrix}\right)$.其中,均匀独立随机选取多项式 $a_{jr}\leftarrow R_q$(下标 $1\leqslant j\leqslant k,1\leqslant r\leqslant k$),$b_j\leftarrow R_q$ $(1\leqslant j\leqslant k)$.
- 第 2 个分布:$\left(\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix},\begin{bmatrix}b_1\\ b_2\end{bmatrix}=\begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\cdot\begin{bmatrix}s_1\\ s_2\end{bmatrix}+\begin{bmatrix}e_1\\ e_2\end{bmatrix}\right)$,简记为$(\boldsymbol{A},\boldsymbol{b}=\boldsymbol{A}\cdot\boldsymbol{s}+\boldsymbol{e})$.其中,均匀独立随机选取多项式 $a_{jr}\leftarrow R_q$(下标 $1\leqslant j\leqslant k,1\leqslant r\leqslant k$),$s_j\leftarrow R_q(1\leqslant j\leqslant k)$,$e_j\leftarrow\chi(1\leqslant j\leqslant k)$.注意:此处的向量 $\boldsymbol{s}$ 为加密算法中的私钥,$(\boldsymbol{A},\boldsymbol{b})$是对应的公钥.

3 基于 MLWE 的同态内积方案

本节将给出本文的同态内积方案.核心问题有两个:一是如何通过密文乘法计算得到内积;二是密文乘法计算后,对应的密钥需要做相应变化.所以,本文参考据 BGV 方案[3]的思路定义了类似于张量积的密文空间运算⊗(详见定义 1).由于 BGV 方案基于 RLWE 问题构造,而本文方案基于 MLWE 问题构造,本文的密文空间计算更为复杂,也会引入更大的解密噪声,所以本文参考 BGV 方案的思想对误差的积累与控制进行了分析,从而保证了本文方案的正确性.并给出了两种优化的加密参数,在两种加密参数下能够高效分别计算两种不同大小的整数向量同态内积.最后证明了本文方案是 IND-CPA 安全的.

3.1 同态内积方案

本节给出了本文的同态内积方案,总共分为 5 块——密钥生成:*keygen*(*parameters*);加密:*Enc*(*pk*,*m*);解密:*Dec*(*sk*,*c*);同态内积计算:*Evalutate*(c_1,c_2);同态运算后解密:*AfterDec*($c_1\otimes c_2$,*sk*).

首先给出方案中将出现的符号的定义,以帮助读者阅读,见表 1.

Table 1 Definitions of the notations in the scheme

表 1 方案记号的定义

符号	意义
λ	安全参数,方案可抗 2^{λ}次攻击
k	$k=k(\lambda)=2$,由安全参数λ决定的模的次数
n	R_q 上多项式的次数
q	$q>2^{dp}$ 的素数,决定有限域大小
η	噪声分布的参数
dp	明文压缩参数,决定明文的大小
dt	公钥压缩参数
du	密文压缩参数
dv	密文压缩参数

1. *keygen*(*parameters*)

1) 设安全参数为λ,根据 $MLWE_{n,q,\chi,k}$中的安全性规约,输入加密参数 $n=n(\lambda)=256$;$k=k(\lambda)=2$;模数 $q=q(\lambda)$;环 $R_q=\mathbb{Z}_q[x]/x^n+1$;噪声分布$\beta_\eta$,其中,$\eta=5$;公钥压缩参数 dt;
2) 均匀随机取样 $\boldsymbol{A}\leftarrow R_q^{k\times k}$;
3) 均匀取样私钥与噪声 $(\boldsymbol{s},\boldsymbol{e})\leftarrow\beta_\eta^k\times\beta_\eta^k$;
4) $\boldsymbol{t}:=CO\mathit{mpress}_q(\boldsymbol{A}\boldsymbol{s}+\boldsymbol{e},dt)$;
5) 输出公钥 $pk:=(\boldsymbol{t},\boldsymbol{A})$,私钥 $sk:=\boldsymbol{s}$.

2. *Enc*(*pk*,*m*)

1) 输入公钥 $pk:=(\boldsymbol{t},\boldsymbol{A})$、明文 $m\in\{0,\ldots,2^{dp}-1\}^n$、公钥压缩参数 dt、密文压缩参数 du 和 dv、加密参数 dp.其中,明文 m 是将 n 格规定范围的整数作为系数的多项式;

2) 对公钥进行解压缩操作 $\boldsymbol{t}'$:=$Decompress_q(\boldsymbol{t},dt)$;

3) 均匀随机取样随机向量及噪声 $(\boldsymbol{r},\boldsymbol{e}_1,e_2)\leftarrow\beta_\eta^k\times\beta_\eta^k\times\beta_\eta$;

4) 加密得到密文: $\boldsymbol{u}:=Compress_q(\boldsymbol{A}^T\boldsymbol{r}+\boldsymbol{e}_1,du)\in R_q^k$;

5) 加密得到密文: $v:=Compress_q\left(\boldsymbol{t}^T\boldsymbol{r}+e_2+round\left(\frac{q}{2^{dp}}\right)\cdot m,dv\right)\in R_q$;

6) 输出密文 $c:=(\boldsymbol{u},v)\in R_q^k\times R_q$.

3. *Dec*(sk,c)

1) 输入私钥 sk:=$\boldsymbol{s}$、密文 c:=($\boldsymbol{u}$,v)、密文解压缩参数 du 和 dv、加密参数 dp;

2) 解压缩得到 $\boldsymbol{u}'=Decompress_q(\boldsymbol{u},du)$;

3) 解压缩得到 $v'=Decompress_q(v,dv)$.

输出明文:

$$m':=Compress_q(v'-\boldsymbol{s}^T\boldsymbol{u}',dp) \tag{1}$$

4. *Evalutate*(c_1,c_2)

为了说明方案的同态内积计算 *Evalutate*(c_1,c_2),首先给出两种密文空间上的张量积运算的定义(具体分析见第 3.3 节).

定义 1. 分别定义两种密文空间上的张量积运算:

a) 对于密文 $c_1=(\boldsymbol{u}_1,v_1)\in R_q^k\times R_q$ 与 $c_2=(\boldsymbol{u}_2,v_2)\in R_q^k\times R_q$,其中,本文取 k=2,定义密文张量积为

$$c_1\otimes c_2=(v_1v_2,v_2u_{10},v_1u_{20},v_2u_{11},v_1u_{21},u_{10}u_{21},u_{11}u_{20},u_{10}u_{20},u_{11}u_{21}).$$

b) 对于密钥 $\boldsymbol{s}\in\beta_\eta^k$,定义密钥张量积 $\boldsymbol{s}\otimes\boldsymbol{s}$=(1,$-s_0$,$-s_0$,$-s_1$,$-s_1$,$s_0s_1$,$s_0s_1$,$s_0s_0$,$s_1s_1$).定义 $\boldsymbol{s}\otimes\boldsymbol{s}$ 为计算密钥.

Evalutate(c_1,c_2):

1) 输入密文 c_1,c_2,其中,

- $c_1=Enc(pk,m_1)=(\boldsymbol{u}_1,v_1)\in R_q^k\times R_q$ 是加密明文 m_1 得到的密文;
- $c_2=Enc(pk,m_2)=(\boldsymbol{u}_2,v_2)\in R_q^k\times R_q$ 是加密明文 m_2 得到的密文;

2) 求密文张量积 $c_1\otimes c_2$,其中,⊗见定义 1 中的种类 a),该运算对应明文的向量内积(将在第 3.3 节给出详细分析).

5. *AfterDec*($c_1\otimes c_2$,sk)

1) 输入私钥 sk:=$\boldsymbol{s}$ 并根据定义 1 中的种类 b)计算密文内积 $\boldsymbol{s}\otimes\boldsymbol{s}$.

2) 解密内积:$m_3=round(c_1\otimes c_2\cdot\boldsymbol{s}\otimes\boldsymbol{s}\cdot2^{2\cdot dp}/q^2)\ \bmod\ 2^{dp}$,多项式 m_3 的常数项即多项式 m_1 与 m_2 的系数组成的两个 n 维向量的内积(其中,m_2 的系数要以特殊的方式排列,将在下一节详细分析).

3) 存在计算的目标整数向量大于 n 维的情况,通过以下方案计算.

假设要计算两个 n+1 维的向量内积,则将向量分为两块加密:

- 第 1 块加密前 n 维向量得到密文 c_1=($\boldsymbol{u}_1$,v_1)与 c_2=($\boldsymbol{u}_2$,v_2).
- 第 2 块加密剩下的 1 维向量(用 0 填充明文多项式剩下的位置),得到密文 c_3=($\boldsymbol{u}_3$,v_3)与 c_4=($\boldsymbol{u}_4$,v_4).此时,计算同态内积即分别先计算 $c_1\otimes c_2$ 与 $c_3\otimes c_4$,然后进行密文相加(将向量 $c_1\otimes c_2$ 与 $c_3\otimes c_4$ 对应位置分别相加).解密即计算:

$$m_3=round((c_1\otimes c_2+c_3\otimes c_4)\cdot\boldsymbol{s}\otimes\boldsymbol{s}\cdot2^{2\cdot dp}/q^2)\bmod 2^{dp} \tag{2}$$

其他维数的情况以此类推.

3.2 通过多项式乘法计算向量内积

由于明文 m_1 与 m_2 是 R_q 上的多项式,可以通过如下多项式乘法运算来求得整数向量的内积:

若多项式 $m_1=w_0+w_1x+\ldots+w_{n-1}x^{n-1}\in R_q$,$m_2=u_0+u_1x+\ldots+u_{n-1}x^{n-1}\in R_q$,令多项式 $m_3=y_0+y_1x+\ldots+y_{n-1}x^{n-1}=m_1\cdot m_2$,

该乘法是 R_q 上的乘法,所以满足以下关系:

$$(y_0,...,y_{n-1})=(w_0,...,w_{n-1})\cdot\begin{bmatrix} u_0 & u_1 & u_2 & ... & u_{n-1} \\ -u_{n-1} & u_0 & u_1 & ... & u_{n-2} \\ -u_{n-2} & -u_{n-1} & u_0 & ... & u_{n-3} \\ \vdots & \vdots & \vdots & ... & \vdots \\ -u_1 & -u_2 & -u_3 & ... & u_0 \end{bmatrix},$$

即可得到 $y_0=(w_0,w_1,w_2,\ldots,w_{n-1})\cdot(u_0,-u_{n-1},-u_{n-2},\ldots,-u_1)$.所以在求两个即令两个 n 维整数向量 $\boldsymbol{a}=(a_0,\ldots,a_{n-1})$ 和 $\boldsymbol{b}=(b_1,\ldots,b_{n-1})$的同态内积时,需要按照以下排列将整数向量 $\boldsymbol{b}$ 作为多项式 m_2 的系数:

$$m_2=b_0-b_{n-1}x-b_{n-2}x^2-\ldots-b_1x^{n-1}\in R_q \tag{3}$$

3.3 通过密文空间张量运算计算同态内积

与 BGV[3]方案类似,可以采用密文作张量积的方式进行同态乘法运算.但是由于本文方案是基于 MLWE 问题构造的,与 BGV 方案的密文空间运算以及进行噪声的膨胀方式均不同.本节将具体分析第 3.1 节定义 1 中的密文空间运算⊗.

根据公式(1),通过解密算法得到两个明文 $m_1=round((v_1-\boldsymbol{s}^T\boldsymbol{u}_1)\cdot 2^{dp}) \bmod 2^{dp}=(a_0+a_1x+\ldots+a_{n-1}x^{n-1})\in R_q$,$m_2=round((v_2-\boldsymbol{s}^T\boldsymbol{u}_2)\cdot 2^{dp}/q) \bmod 2^{dp}=(b_0-b_{n-1}x-b_{n-2}x^2-\ldots-b_1x^{n-1})\in R_q$,根据第 3.2 节的分析,密文空间的运算只要对应明文做 R_q 上的多项式乘法即可,所以等式两边同时做乘法:

$$\begin{aligned} m_3 &= m_1\cdot m_2 \\ &= round((v_1-\boldsymbol{s}^T\boldsymbol{u}_1)\cdot(v_2-\boldsymbol{s}^T\boldsymbol{u}_2)\cdot 2^{2\cdot dp}/q^2)\bmod 2^{dp} \\ &= round(c_1\otimes c_2\cdot(1,-s_0,-s_0,-s_1,-s_1,s_0s_1,s_0s_1,s_0s_0,s_1s_1)\cdot 2^{2\cdot dp}/q^2)\bmod 2^{dp} \\ &= (z_0+z_1x+...+z_{n-1}x^{n-1})\in R_q \end{aligned} \tag{4}$$

其中,$c_1\otimes c_2=(v_1v_2,v_2u_{10},v_1u_{20},v_2u_{11},v_1u_{21},u_{10}u_{21},u_{11}u_{20},u_{10}u_{20},u_{11}u_{21})$即密文空间运算.那么进行同态运算之后,需要通过计算密钥 $\boldsymbol{s}\otimes\boldsymbol{s}$ 来解密.具体来讲就是第三方或者云服务器进行密文空间运算 $c_1\otimes c_2$ 后得到向量$(v_1v_2,v_2u_{10},v_1u_{20},v_2u_{11},v_1u_{21},u_{10}u_{21},u_{11}u_{20},u_{10}u_{20},u_{11}u_{21})$,将该向量传给用户,用户持有计算密钥,用户通过计算 $m_3=round(c_1\otimes c_2\cdot\boldsymbol{s}\otimes\boldsymbol{s}\cdot 2^{2\cdot dp}/q^2) \bmod 2^{dp}$ 进行同态运算后的解密操作,其中,$m_3=(z_0+z_1x+\ldots+z_{n-1}x^{n-1})\in R_q$,$z_0=\boldsymbol{a}\cdot\boldsymbol{b}$ 即目标整数向量的内积.但公式(4)成立,要保证将噪声控制在一定范围内.在第 3.4 节中具体分析.

3.4 方案正确性与优化参数

本节将分析本文同态内积方案的正确性,并针对不同的应用场景给出两种推荐的加密参数:在第 1 种参数下,能够计算两个$\{0,128\}^n$整数向量的同态内积;在第 2 种参数下,能够计算两个$\{0,1024\}^n$整数向量的内积.其中,向量的维数 n 根据第 3.1 节中加密算法的安全性取 256.在实际应用场景下(如逻辑回归分类、密文检索),需要计算的目标向量均是浮点数.应用本文的同态内积方案时,需要先将浮点数乘以 10 的幂次变为整数,一般情况下,两位有效数字就够用了(即使用本文推荐的第 1 种加密参数);如果需要更高的精度,那么推荐使用本文第 2 种加密参数,能够同态计算 3 位有效数字的向量内积,但使用第 2 种推荐参数时方案的效率会降低.

由于进行了同态运算,解密时的噪声必然会增大,所以必须调整加密参数才能保证方案的正确性.

根据文献[12],运行第 3.1 节中加解密算法解密后得到明文 $m'=round(m+\varepsilon\cdot 2^{dp}/q) \bmod 2^{dp}$,其中,噪声 $\|\varepsilon\|=\|(\boldsymbol{e}^T\boldsymbol{r}-\boldsymbol{s}^T\boldsymbol{e})+(\boldsymbol{c}_t^T\boldsymbol{r}+\boldsymbol{s}^T\boldsymbol{c}_u)+e_2+c_v\|$,且

$$\Pr\left(\|\varepsilon\|>12\cdot\sqrt{\frac{2\cdot 512\cdot\eta}{6}\cdot\left(\frac{q}{2^{du+1}}+\eta\right)^2+\frac{q}{2^{dv+1}}}\right)=negl \tag{5}$$

接下来分析同态内积之后解密得到正确结果的条件.

设 $m_1'=round(m_1+\varepsilon\cdot 2^{dp}/q) \bmod 2^{dp}$, $m_2'=round(m_2+\varepsilon\cdot 2^{dp}/q) \bmod 2^{dp}$,则

$$m_3'=m_1'\cdot m_2'=(m_1\cdot m_2+2^{2\cdot dp}/q^2\cdot\varepsilon_1\cdot\varepsilon_2+2^{dp}/q+2^{dp}/q\cdot\varepsilon_2\cdot m_1+2^{dp}/q\cdot\varepsilon_1\cdot m_2) \bmod 2^{dp} \tag{6}$$

由公式(6)可以推出 $m_1 \cdot m_2 \cdot n < 2^{dp}/2$.根据应用场景的需要,当计算两个$\{0,128\}^n$整数向量的同态内积时,$0 \leqslant m_1, m_2 \leqslant 2^8$;当计算两个$\{0,1024\}^n$整数向量的同态内积时,$0 \leqslant m_1, m_2 \leqslant 2^{10}$.所以在第 1 种推荐参数下 dp=23,第 2 种推荐参数下 dp=29.

由公式(6)推出同态内积操作后解密正确的条件如下式:

$$\Pr(\|2^{2\cdot dp}/q^2 \cdot \varepsilon_1 \cdot \varepsilon_2 + 2^{dp}/q + 2^{dp}/q \cdot \varepsilon_2 \cdot m_1 + 2^{dp}/q \cdot \varepsilon_1 \cdot m_2\| > 1/2) = negl \tag{7}$$

其中,令$\alpha_1 = 2^{2\cdot dp}/q^2 \cdot \varepsilon_1 \cdot \varepsilon_2$,$\alpha_2 = 2^{dp}/q$,$\alpha_3 = 2^{dp}/q \cdot \varepsilon_2 \cdot m_1 + 2^{dp}/q \cdot \varepsilon_1 \cdot m_2$.所以,同态内积运算后解密正确的条件为

$$\Pr(\|\alpha_1 + \alpha_2 + \alpha_3\| > 1/2) = negl.$$

将公式(5)带入公式(7),分别得到:

$$\alpha_1 = 2^{2\cdot dp}/q^2 \cdot \varepsilon_1 \cdot \varepsilon_2 = \frac{6144(2^{dp})^2\eta}{(2^{du})^2} + \frac{24576(2^{dp})^2\eta^2}{q \cdot 2^{du}} + \frac{24576(2^{dp})^2\eta^3}{q^2} + \frac{64(2^{dp})^2\sqrt{6}\sqrt{\frac{\eta q^2}{4(2^{du})^2} + \frac{\eta^2 q}{2^{du}} + \eta^3}}{q \cdot 2^{dv}} + \frac{(2^{dp})^2}{4(2^{dv})^2} \tag{8}$$

$$\alpha_2 = 2^{dp}/q \tag{9}$$

$$\alpha_3 = 2^{dp}/q \cdot \varepsilon_2 \cdot m_1 + 2^{dp}/q \cdot \varepsilon_1 \cdot m_2 = \frac{2\left(64\sqrt{6}\sqrt{\eta\left(\frac{q}{2^{du+1}} + \eta\right)^2} + \frac{q}{2^{dv+1}}\right)(2^{2\cdot dp})}{q} \tag{10}$$

保证$\|\alpha_1+\alpha_2+\alpha_3\|>1/2$的概率可以忽略不计,本文方案能够解密正确.根据第 3.1 节加密参数η=5,给出本文的第 1 种推荐参数(计算两位有效数字的整数向量内积),此时 dp=23,当模数 q=73786976294838206633、压缩参数 $du=dt=dv$=60 时,同态内积操作后能够正确解密;本文的第 2 种推荐参数(计算 3 位有效数字的整数向量内积),此时 dp=29,当模数 q=48357032784585516698824713、压缩参数 $du=dt=dv$=79 时,同态内积操作后能够正确解密.这两种加密参数均能够保证目标向量维数超过 n 维时,通过公式(2)多次进行密文加法时噪声保证能够正确解密的范围内.

3.5 方案安全性

定理 1. 在 $MLWE_{n,q,\chi,k}$ 问题假设的前提下,本文的同态内积方案是 IND-CPA 安全的.

证明:使用基于游戏的证明思路,用 $Adv_{Game1}[A]$表示敌手 A 在下列游戏中的优势.由于前面部分与文献[12]中定理 3 的证明类似,该处不再赘述.

Game 0: Game 0 为标准的 IND-CPA 游戏,按照前面给出的方案来生成公钥 pk:=(t,A),私钥 sk:=s.

Game 1: Game 1 改变 Game 0 中公钥 t:=As+e 的生成过程,采用均匀随机选取 $\boldsymbol{t}' \leftarrow R_q^k$.我们有:

$$Adv_{Game1}[A] - Adv_{CPA}[A] \leqslant Adv_{MLWE}[A].$$

Game 2: Game 2 改变 Game 1 中挑战密文的生成方式,不使用公钥加密得到挑战密文,而是均匀随机选取 $(\boldsymbol{u}', v') \leftarrow R_q^k \times R_q$.我们有:

$$Adv_{Game2}[A] - Adv_{Game1}[A] \leqslant Adv_{MLWE}[A].$$

Game 3: Game 3 使用 Game 2 中挑战密文的生成方式,均匀选取两段随机密文 $(u_1', v_1') \leftarrow R_q^k \times R_q$ 与 $(u_2', v_2') \leftarrow R_q^k \times R_q$.由于方案支持同态内积,必须还要考察密文的张量形式,敌手 A 区分 $\left(\begin{pmatrix}\boldsymbol{A}^T \\ \boldsymbol{t}^T\end{pmatrix}, \begin{pmatrix}\boldsymbol{u}_1' \\ v_1'\end{pmatrix} \otimes \begin{pmatrix}\boldsymbol{u}_2' \\ v_2'\end{pmatrix}\right)$和 $\left(\begin{pmatrix}\boldsymbol{A}^T \\ \boldsymbol{t}^T\end{pmatrix}, \begin{pmatrix}\boldsymbol{u}_1 \\ v_1\end{pmatrix} \otimes \begin{pmatrix}\boldsymbol{u}_2 \\ v_2\end{pmatrix}\right)$的分布与 $MLWE_{n,q,\chi,k}$ 问题一样难,其中,(u_1,v_1)和(u_2,v_2)分别为 m_1 和 m_2 的密文,所以,Game 3 与 Game2 中敌手 A 的优势差为

$$Adv_{Game3}[A] - Adv_{Game2}[A] \leqslant Adv_{MLWE}[A].$$

所以,总的来说,

$$Adv_{CPA}[A] \leqslant 3 \cdot Adv_{MLWE}[A].$$

在 $MLWE_{n,q,\chi,k}$ 问题是困难的假设成立的情况下,$Adv_{MLWE}[A]$可以忽略不计,本文的同态内积方案可证明是 IND-CPA 安全的. □

4 方案效率与计算复杂度

4.1 方案计算复杂度

本节给出了本文方案各个阶段计算复杂度.以下所有的 n,k,dp 均对应第 3 节中给出的加密参数.

本文只分析两个 n 维整数向量做内积时的计算复杂度.如果向量维数小于 n,则在多的位置补 0,计算复杂度则与 n 维整数向量内积时相同;维数大于 n 时则多次调用本文方案,时间复杂度相应增大.通过第 3.4 节中的方案正确性分析不难得出:在两种加密参数下,均能保证 n 小于 2 048 时,多次调用本文方案所累计的噪声不影响解密正确性.更大维数的向量内积则需具体分析,不作为本文的重点.

在生成公私钥对时,由于要调用中心二项分布函数生成满足算法要求分布的噪声,时间复杂度为 $O(n)$,空间复杂度为公钥大小 $O(k^2n)$.

在进行加密操作时,主要消耗的计算量是多项式乘法,其他计算(如压缩与解压缩操作)的计算复杂度可以忽略不计.加密时要进行 k^2+k 次多项式乘法,根据文献[16],加密时的时间复杂度是 $O((k^2+k)\cdot(3n\log n+n))$,空间复杂度为快速傅里叶变换所消耗的空间 $O(n)$.由于本文方案关注的是同态内积,所以不考虑直接解密加密后的明文的计算复杂度.

在进行同态内积操作时,主要消耗的计算量同样是多项式乘法,计算一次同态内积,要进行$(k+1)^2$ 次多项式乘法.加密时的时间复杂度是 $O((k+1)^2\cdot(3n\log n+n))$,空间复杂度为快速傅里叶变换所消耗的空间 $O(n)$.

在同态解密阶段,主要分为两步进行:首先,生成计算密钥,要进行 k^2 次多项式乘法,时间复杂度是 $O(k^2\cdot(3n\log n+n))$,空间复杂度为快速傅里叶变换所消耗的空间 $O(n)$;然后,解密时,主要求密文以及计算密钥的内积 $O((k^2-1)\cdot(3n\log n+n))$,空间复杂度为快速傅里叶变换所消耗的空间 $O(n)$.

表 2 给出了本文方案各个阶段的计算复杂度.

Table 2 Scheme computational complexity

表 2 方案计算复杂度

算法阶段	时间复杂度	空间复杂度
密钥生成	$O(n)$	$O(k^2n)$
加密操作	$O((k^2+k)\cdot(3n\log n+n))$	$O(n)$
同态内积	$O((k+1)^2\cdot(3n\log n+n))$	$O(n)$
同态解密	$O((2k^2-1)\cdot(3n\log n+n))$	$O(n)$

4.2 方案效率与实现

本文的实验环境采用 CPU 为 Intel Core i5-3470、频率:3.20GHz、内存 8GB 的计算机,操作系统为 CentOS7 64 位,算法使用 C++编程实现,大整数计算调用了 NTL 计算库.

本节给出了两种加密参数下,本文同态内积方案的效率,并与其他几种方案计算同态内积的效率与安全性进行了对比,见表 3.

Table 3 Scheme efficiency comparison

表 3 方案效率对比

方案	安全性	同态内积效率
Xu[6]	RLWE	256 维 8bits 向量内积大于 20s
Yasuda[7]	理想格	2048 维 bits 向量内积 38s
Dijk[8]	近似 GCD 问题	256 维 10bits 向量内积 0.5ms
Zhou[10]	LWE	40 维 10bits 向量内积 757.2ms
本文方案(参数 1)	MLWE	256 维 7bits 向量内积 10ms
本文方案(参数 2)	MLWE	256 维 10bits 向量内积 20ms

从表 3 中可以看出,计算同态内积最快的方案是 Dijk 等人[8]的 DGHV 方案.但该方案的安全性是基于近似 GCD 问题,已被证明有办法破解[9].而 Yasuda 等人的方案[7]基于理想格构造,引入了打包技术提升效率,但只能够计算 bit 向量的内积,是不实用的.Zhou 的计算整数内积的方案[10]在计算大于 40 维的同态内积时,公钥会非常大,计算内积的效率也十分低.Xu 等人[6]基于 HElib 优化的全同态加密方案,安全性基于 RLWE 问题,是目前最接近实用的方案,使用 SIMD 技术优化后计算 256 维 8bits 同态内积需要进行 1 次乘法与 8 次加法共 20s 左右.本文基于 Ke 等人[12]的基于 MLWE 的公钥加密方案构造了一种基于 MLWE 的同态内积方案,计算 256 维 7bits 向量内积需要 10ms,计算 256 维 10bits 向量内积需要 20ms,在效率与实用性上比其他方案有一定的提升.

5 结束语

本文基于格困难问题 MLWE 构造了一种安全计算整数向量内积的同态内积方案,由于基于的格问题是 MLWE,在保证了安全性的前提下能够取更小的加密参数,并通过计算多项式乘法的方式来计算整数向量内积,代替了传统同态加密方案中通过电路计算的方式,将时间复杂度降低到 O($n$log$n$).通过 C++与 NTL 库实现了该方案,计算 256 维 7bit 同态内积需要 10ms 左右,比起 Xu 等人[6]的全同态加密方案有很大的提升,有望应用于某些实际场景.下一步工作将研究把本文的同态内积方案应用于安全多方几何计算、隐私数据挖掘等实际场景.

References:

[1] Yao AC. Protocols for secure computations. In: Proc. of the 23rd Annual Symp. on Foundations of Computer Science. 1982. 160−164.

[2] Gentry C. A fully homomorphic encryption scheme [Ph.D. Thesis]. Stanford University, 2009.

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

[4] Lyubashevsky V, Peikert C, Regev O. On ideal lattices and learning with errors over rings. In: Proc. of the Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer-Verlag, 2010. 1−23.

[5] Halevi S, Shoup V. Algorithms in HElib. In: Proc. of the Int'l Cryptology Conf. Berlin, Heidelberg: Springer-Verlag, 2014. 554−571.

[6] Xu C, Chen J, Wu W, *et al*. Homomorphically encrypted arithmetic operations over the integer ring. In: Proc. of the Int'l Conf. on Information Security Practice and Experience. Cham: Springer-Verlag, 2016. 167−181.

[7] Yasuda M, Shimoyama T, Kogure J, Yokoyama K, Koshiba T. Packed homomorphic encryption based on ideal lattices and its application to biometrics. In: Proc. of the Int'l Conf. on Availability, Reliability, and Security. Berlin, Heidelberg: Springer-Verlag, 2013. 55−74.

[8] Van Dijk M, Gentry C, Halevi S, Vaikuntanathan V. Fully homomorphic encryption over the integers. In: Proc. of the Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer-Verlag, 2010. 24−43.

[9] Ding J, Tao C. A new algorithm for solving the general approximate common divisors problem and cryptanalysis of the FHE based on the gacd problem. 2014. http://eprint.iacr.org/2014/042

[10] Zhou H, Wornell G. Efficient homomorphic encryption on integer vectors and its applications. In: Proc. of the Information Theory and Applications Workshop (ITA 2014). IEEE, 2014. 1−9.

[11] Bogos S, Gaspoz J, Vaudenay S. Cryptanalysis of a homomorphic encryption scheme. Cryptography and Communications, 2016, 10(1):27−39.

[12] Ke CS, Wu WY, Feng Y. Low expansion rate encryption algorithm based on MLWE. Computer Science, 2019,46(4):144−151 (in Chinese with English abstract).

[13] Ke CS. Encryption algorithm based on module-LWE [MS. Thesis]. Chongqing: Chongqing University of Posts and Telecommunications, 2019 (in Chinese with English abstract).

[14] Bos J, Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schanck J, Schwabe P, Stehlé D. CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM. In: Proc. of the 2018 IEEE European Symp. on Security and Privacy (EuroS&P). 2018. 353−367. [doi: 10.1109/EuroSP.2018.00032]

[15] Langlois A, Stehlé D. Worst-Case to average-case reductions for module lattices. Designs, Codes and Cryptography, 2015,75(3): 565–599.

[16] Golub GH, Van Loan CF. Matrix Computations. Baltimore: The Johns Hopkins University Press, 2012. 219–222.

附中文参考文献:

[12] 柯程松,吴文渊,冯勇.基于 MLWE 的低膨胀率加密算法.计算机科学,2019,46(4):144–151.

[13] 柯程松.基于模容错学习问题的加密算法研究[硕士学位论文].重庆:重庆邮电大学,2019.

柯程松(1994－),男,硕士,主要研究领域为同态加密,格密码.

冯勇(1965－),男,博士,研究员,博士生导师,CCF 专业会员,主要研究领域为符号数值计算.

吴文渊(1976－),男,博士,研究员,主要研究领域为符号数值计算,格密码.