

基于余数系统的小间隔插值拟合自举方法

李慧贤¹, 王富磊¹, 沈春², 刘诗源¹, 庞辽军³

(1.西北工业大学 计算机学院, 陕西 西安 710072; 2.西北工业大学 软件学院, 陕西 西安 710072;
3.西安电子科技大学 生命科学技术学院, 陕西 西安 710071)

摘要:针对近似同态加密方案自举耗时过大的问题提出了一种基于余数系统的小区间插值拟合自举方法。通过在多个小区间内对模函数进行插值拟合避免因拟合多项式次数过高产生自举时间过长或计算精度降低的问题,并通过结合余数系统提高计算过程中模乘运算的模逆运算效率。选用拉格朗日插值多项式对小区间内的正弦函数进行插值拟合。通过多个低次多项式复合计算实现比较函数,并提出了一种区间判断算法来识别密文所在区间。最终在24比特精度下,同态计算过程中模运算耗时下降到 HEAAN 库的8%,在计算槽的数量为65 536时,平均每槽的模运算时间为0.028 ms。

关键词:全同态加密;近似计算;自举;余数系统;插值拟合;拉格朗日插值

中图分类号:TN918.1

文献标志码:A

文章编号:1000-2758(2024)05-0969-10

随着大数据、云计算等技术的发展与应用,数据隐私保护问题成为目前隐私计算领域研究的重点问题。同态加密方案可以直接对加密数据进行操作,达到与对明文进行同样操作的结果,可以有效完成隐私计算任务需求,已成为目前的研究热点。

2017年,Cheon等^[1]首次提出了可以支持浮点数运算的层次同态加密方案 CKKS17,该方案基于单指令多数据操作(single instruction multiple data, SIMD),可以高效地进行算术运算。因此,该方案应用价值较高,已经应用于全基因组关联分析^[2]、聚类分析^[3]以及神经网络^[4]等领域。

但 CKKS17 方案为层次同态加密方案,其仅能支持有限次同态加法和同态乘法运算。可以进行任意次同态加法和同态乘法运算的加密方案被称为全同态加密方案,需要使用自举技术才能将层次同态加密方案转换为全同态加密方案。Cheon等^[5]为 CKKS17 设计了近似同态加密的自举方案,从而实现全同态加密方案。该方案的4个主要步骤是模数提升、槽转换到系数、模函数和系数转换到槽数。模

数提升是将位于小模数 q 上的密文 c_i 提升到大模数 Q 上,其中 $Q \gg q, s_k, m$ 分别表示密文 c_i 的私钥和明文,即 $\langle c_i, s_k \rangle \bmod q$,那么就有 $\langle c_i, s_k \rangle \bmod Q = ql + m, l$ 是一个整数。槽转换到系数和系数转换到槽数是2个线性转换过程。模函数为 $(ql + m) \bmod q = m$,但 CKKS 方案只支持同态加法和同态乘法,因此无法直接计算非多项式函数的模函数,而需要将其转换成三角函数 $q/2\pi \cdot \sin(2\pi(ql + m)/q)$,再利用泰勒展开公式计算三角函数。

但自举技术的运算耗时过大,不利于结合现实应用。为提高近似同态加密方案的自举效率,本文提出一种基于余数系统的小间隔插值拟合自举方法,通过在多个小间隔内对模函数进行近似拟合完成自举过程中模运算近似拟合,并通过余数系统提高方案的模乘运算和模逆运算计算速度,最终使模运算计算耗时降为目前最新公布的近似同态加密库 HEAAN^[6]的8%。

1 相关工作

CHK+18a 方案^[5]利用等价无穷小性质,当 $x \rightarrow 0$ 时, x 与 $\sin x$ 为等价无穷小,并通过模函数曲线发现,可以通过 $[\langle c_i, s_k \rangle]_q \approx q/2\pi \cdot \sin(2\pi/q \cdot \langle c_i, s_k \rangle)$ 来对模函数进行近似计算。同时在计算的过

收稿日期:2023-09-25

基金项目:陕西省自然科学基金基础研究计划(2023-JC-YB-546, 2024JC-YBMS-471)资助

作者简介:李慧贤(1977—),副教授

通信作者:李慧贤(1977—) e-mail:lihuixian@nwpu.edu.cn

程中,CHK+18a 方案使用欧拉公式 $e^{ix} = \cos x + i \sin x$ 以及幂运算 $e^{i \cdot 2\theta} = (e^{i \cdot \theta})^2$ 对三角函数的计算进行简化处理,最终通过泰勒展开多项式进行计算,从而完成了模运算,最终实现了自举技术。但该方案在使用泰勒展开多项式对模函数进行近似计算过程中,为保证精度要求,所需多项式的次数过高,从而导致方案计算效率下降。

针对于此,CCS19 方案^[7]使用切比雪夫插值方法取代泰勒展开方法,并通过优化 Paterson-Stockmeyer 算法来快速计算密文上的切比雪夫插值,将每个明文槽的自举时间提高 2 个数量级,即从 1 s 减少至 0.01 s,但计算效率依然过低。HK20 方案^[8]通过考虑密文尺寸和密文模尺寸之间的比例对计算正弦函数过程进行优化,并最终用余弦函数替代正弦函数,最终使得方案的标量乘法运算量下降到 CCS19 方案的 50%,然而该方案得到的近似多项式系数过大,从而导致计算结果精度较差,无法满足实际应用中的高精度需求。JKA+21 方案^[9]通过结合 GPU 对近似同态加密的运行效率进行了极大提升,该方案发现 GPU 并行化难点在于高主存带宽需求,利用内存中心化方法进行处理,JKA+21 方案单个同态乘法操作与此前 GPU 实现相比加速了 7.02 倍,在 GPU 上每比特自举平均用时 0.423 μs ,相较于此前单线程 CPU 加速了 257 倍。CN22 方案^[10]使用正弦级数替代 CHK+18a 方案中的正弦函数,大幅度提升了方案的计算精度,自举精度最高可以达到 100 多位。但由于存在从正弦级数 $\{\sin(kx)\}$ 到正弦函数 $\{\sin^k(x)\}$ 幂次的线性变换,与先前文献中的 $\sin x$ 相比,其计算效率基本不变。

方案 CHK+18b^[11]将余数系统与近似同态加密相结合提升方案的模乘、模逆运算效率,但 CHK+18b 没有提供余数系统下的自举方案。BMTH21^[12]方案是第一个余数系统下的自举方案,通过采用优化密钥交换技术和矩阵向量乘法对方案效率进行提升。LLL+21^[13]提出了一种求模函数和三角函数的 minimax 近似多项式的快速算法——改进的多区间雷米兹(Remez)算法,减少了自举过程中的误差,精度比 HK20 方案提高了 5.4~10.2 比特,但该方案的计算时间是 BMTH21 方案的 20 倍左右。本文为基于余数系统的近似加密同态方案提出一种小间隔插值拟合自举方法。对于模函数 $(ql+m) \bmod q = m$, $l \in (-K, K)$,且 l 为整数,结合模函数的周期性,可以将其分割成 $2K$ 小区间,在每个小区间内使用拉

格朗日插值法得到高精度的小次数拟合多项式, $2K$ 小区间对应 $2K$ 个拟合多项式。使用同态加密比较算法算出密文所处的区间,在小区间内进行模函数计算。与 CHK+18a、HK20 等相比较,该方法可大幅度提高自举方案的计算效率和计算精度,满足了隐私计算、机器学习等领域的应用需求。

2 符号定义

本文使用加粗小写字母表示向量,如 \mathbf{a}, a_i 表示向量 \mathbf{a} 的第 i 个元素,无特殊说明下标从 0 开始,对数的底数为 2。 \mathbf{Z} 代表整数, \mathbf{R} 代表实数, \mathbf{Q} 代表有理数, \mathcal{R} 代表环, \mathbf{Z}^+ 代表正整数, \mathbf{Z}^n 代表整数上的 n 维向量, \mathbf{Z}_q^n 代表模 q 整数上的 n 维向量, $\mathbf{Z}^{n \times m}$ 代表整数上 n 行 m 列的矩阵, $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ 代表模为 q 的剩余环。对于实数 $r \in \mathbf{R}$, $\lfloor r \rfloor$, $\lceil r \rceil$ 和 $\lceil r \rceil$ 分别表示对 r 进行向下取整、近似取整和向上取整。对于长度为 n 的向量 \mathbf{v} , $\|\mathbf{v}\|_0$ 表示 0 范数,为向量 \mathbf{v} 中非零元素的个数; $\|\mathbf{v}\|_1$ 表示一范数,表示向量 \mathbf{v} 中所有元素绝对值之和,即 $\|\mathbf{v}\|_1 = |x_0| + |x_1| + |x_2| + \dots + |x_{n-1}|$; $\|\mathbf{v}\|_2$ 表示二范数,表示向量 \mathbf{v} 中所有元素平方和的开方,即 $\|\mathbf{v}\|_2 = \sqrt{|x_0|^2 + |x_1|^2 + |x_2|^2 + \dots + |x_{n-1}|^2}$; $\|\mathbf{v}\|_\infty$ 表示无穷范数,表示向量中所有元素绝对值的最大值,即 $\|\mathbf{v}\|_\infty = \max(|x_0|, |x_1|, |x_2|, \dots, |x_{n-1}|)$ 。 $x \leftarrow D$ 表示从分布 D 中随机采样得到样本 x , U 代表均匀分布, $\text{DG}(\sigma^2)$ 表示 \mathbf{Z}^N 上的方差为 σ^2 的离散高斯分布;规定均匀分布 $\text{HWT}(h)$ 产生一个汉明权重为 h 、长度为 N 的向量 $\{\pm 1\}^N$;规定对于给定的概率 ρ ,分布 $\text{ZO}(\rho)$ 以 $\rho/2$ 的概率取 $+1$ 或 -1 ,以 $1-\rho$ 的概率取 0,其中 $0 \leq \rho \leq 1$, x_B 表示最大值上限为 B 的随机分布。使用 $\langle \mathbf{a}, \mathbf{b} \rangle$ 表示 2 个向量的内积。复合函数 $f[g(x)]$ 记作 $g(x) \circ f(x)$,函数 $f(x)$ 的导数记作 $f'(x)$ 。

λ 为安全参数。通过安全参数 λ 对参数 P 进行选择,记作 $P = P(\lambda)$ 。

3 基于余数系统的小间隔插值拟合自举方法

本文将余数系统与近似同态加密结合起来提高运算效率,并通过采用小间隔插值拟合方法对自举过

程中的模运算进行优化。

3.1 基于余数系统的小区间插值拟合自举算法

基于余数系统的小间隔插值算法包含余数系统下的小区间插值算法、余数系统下的区间判断算法和余数系统下的新鲜密文计算方法。

3.1.1 基于余数系统的小间隔插值算法

使用正弦函数 $S(t) = q/2\pi \sin(2\pi/q \cdot t)$ 对模函数 $F(t) = [t]_q$ 进行近似计算时,对周期为 q 的 $S(t)$, 可以将 $S(t)$ 分割为 $2K + 1$ 个小区间, 分别为 $[-q/2, -Kq, q/2 - Kq], (-q/2 - (K-1)q, q/2 - (K-1)q), \dots, (-3q/2, -q/2], (-q/2, q/2], (q/2, 3q/2], \dots, (-q/2 + Kq, q/2 + Kq]$ 在每个小区间内对 $S(t)$ 进行拉格朗日插值拟合。

对于上述 $2K + 1$ 个小区间内的函数,使用拉格朗日插值方法对函数进行多项式拟合。仍然选取函数值趋近于 x 轴附近的曲线进行拟合,用 x_{left} 表示插值左端点, x_{right} 表示插值右端点。

由正弦函数周期性可知, 区间 $[x_{\text{left}}, x_{\text{right}}]$ 一定包含在某一个完整的区间 $(-q/2 + iq, q/2 + iq]$ 中, 其中 $i \in \mathbf{Z} \cap [-2K, 2K]$ 。

对于函数 $y = f(x)$ 在区间 $[x_{\text{left}}, x_{\text{right}}]$ 上的 $n + 1$ 个节点 $x_i, i = 0, 1, \dots, n, x_0 < x_1 < \dots < x_n$, 有函数值 $y_i = f(x_i)$, 按照 $L(x) = \sum_{k=0}^n \left(\prod_{\substack{i=0 \\ i \neq k}}^n \frac{x - x_i}{x_k - x_i} \cdot y_k \right)$ 形式计算拉格朗日插值基函数, 同时选择满足 $f(x) - L(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i)$ 的最小 n 值作为拉格朗日插值多项式的次数 n_{ploy} 。

在进行拉格朗日插值过程中, 拉格朗日插值多项式的次数为 n_{ploy} 时需要选取 $n_{\text{ploy}} + 1$ 个插值点, 记第 i 个区间的 $n_{\text{ploy}} + 1$ 个插值点分别为 $x_{i,0}, x_{i,1}, \dots, x_{i,n_{\text{ploy}}-1}, x_{i,n_{\text{ploy}}}$ 。规定选择区间左右端点作为插值点, 则 $x_{i,0} = x_{\text{left}}, x_{i,n_{\text{ploy}}} = x_{\text{right}}$ 。

对于函数 $y = f(x) = S(t) = q/2\pi \cdot \sin 2\pi/q \cdot t$, 计

$$G = \left[E_{\text{ncpk}} \left(x_{0,0} - \frac{q}{2} \right), E_{\text{ncpk}} \left(x_{1,0} - \frac{q}{2} \right), \dots, E_{\text{ncpk}} \left(x_{2K,0} - \frac{q}{2} \right) \right] \quad (1)$$

实际使用的插值区间右端点密文值向量为

$$E = \left[E_{\text{ncpk}} \left(x_{0,n_{\text{ploy}}} + \frac{q}{2} \right), E_{\text{ncpk}} \left(x_{1,n_{\text{ploy}}} + \frac{q}{2} \right), \dots, E_{\text{ncpk}} \left(x_{2K,u} + \frac{q}{2} \right) \right] \quad (2)$$

比较函数 $C_{\text{mpn}}(a, b)$ 的复合多项式次数选择为 $2n + 1$, 当 $2n + 1 = 15$ 时, 基于余数系统的同态加密

算 2 个端点 $x_{i,0}$ 和 $x_{i,n_{\text{ploy}}}$ 处的函数值, 分别为 $y_{x_{i,0}} = f(x_{i,0})$ 和 $y_{x_{i,n_{\text{ploy}}}} = f(x_{i,n_{\text{ploy}}})$ 。按照 y 值等间距选取插值点 $x_{i,j}, j = 0, 1, \dots, n_{\text{ploy}}, y$ 值间隔 $\Delta y = (y_{x_{i,n_{\text{ploy}}}} - y_{x_{i,0}}) / n_{\text{ploy}}$, 则 $y_{x_{i,j}} = y_{x_{i,0}} + j \times \Delta y$ 。通过 $y = f(x)$ 的反函数 $x = f^{-1}(y)$ 计算出 $n_{\text{ploy}} - 1$ 个插值点: $x_{i,j} = f^{-1}(y_{x_{i,j}}), j = 1, 2, \dots, n_{\text{ploy}} - 1$ 。

将上述 $n_{\text{ploy}} + 1$ 个插值点 $(x_{i,j}, y_{x_{i,j}})$ 进行拉格朗日插值拟合, 得到多项式函数 $L(x) = a_0 + a_1x + a_2x^2 + \dots + a_{n_{\text{ploy}}}x^{n_{\text{ploy}}}$ 。

对于区间 $2K + 1$ 个小插值区间做如上操作, 得到包含 $2K + 1$ 个插值多项式的拉格朗日插值多项式族, 这 $2K + 1$ 个插值多项式构成了对原始函数 $S(t) = q/2\pi \cdot \sin 2\pi/q \cdot t$ 选定小区间内的近似拟合。

3.1.2 基于余数系统的区间判断算法

用 $E_{\text{ncpk}}(\mu)$ 表示使用公钥 pk 对明文 μ 加密后的结果。插值区间左端点密文值向量为 $(E_{\text{ncpk}}(x_{0,0}), E_{\text{ncpk}}(x_{1,0}), \dots, E_{\text{ncpk}}(x_{2K,0}))$, 插值区间右端点密文值向量为 $(E_{\text{ncpk}}(x_{0,u}), E_{\text{ncpk}}(x_{1,u}), \dots, E_{\text{ncpk}}(x_{2K,u}))$ 。为避免比较运算过程中密文值恰好等于左右端点值而使比较结果为 $1/2$, 而非期待的 0 或 1 , 将左端点由 $-q/2 + iq$ 移到 $-q/2 + iq - q/2 = (i-1)q$, 右端点由 $q/2 + iq$ 移动到 $q/2 + iq + q/2 = (i+1)q$ 。此时端点区间为 $[(i-1)q, (i+1)q]$, 每一个端点区间长度由 q 增长为 $2q$, 而实际函数所在区间仍为 $[-q/2 + iq, q/2 + iq]$ 。此时再进行 c_i 与比较区间左端点或右端点的计算, 就不会出现插值绝对值过小问题。

端点区间长度由 q 增长为 $2q$, 计算后插值的取值范围也从 $(0, q)$ 增长到了 $(0, 2q)$, 需要对插值结果乘 $1/2$ 进行归一化操作保证结果的准确性。

则可以计算得到实际使用的插值区间左端点密文值向量为

方案多项式计算流程如算法 1 所示。

算法 1 $E_{\text{valPoly}}(x, 2n + 1, a_1, \dots, a_{2n+1})$

输入:一次项 x , 多项式次数 $2n + 1$ 以及多项式系数

$$a_{2i+1}, 0 \leq i \leq n$$

输出:多项式计算结果

$$1: x^2 = x * x$$

$$2: S_{\text{um}} += a_1 * x$$

$$3: x = R_S(x)$$

$$4: x^3 = x * x^2, x^4 = x^2 * x^2$$

$$5: S_{\text{um}} += a_3 * x^3$$

$$6: x = R_S(x)$$

$$7: x^5 = x^3 * x^2, x^7 = x^3 * x^4, x^8 = x^4 * x^4$$

$$8: S_{\text{um}} += a_5 * x^5 + a_7 * x^7$$

$$9: x = R_S(x), x^3 = R_S(x^3)$$

$$10: x^9 = x * x^8, x^{11} = x^3 * x^8, x^{13} = x^5 * x^8, x^{15} = x^7 * x^8$$

$$11: S_{\text{um}} += a_9 * x^9 + a_{11} * x^{11} + a_{13} * x^{13} + a_{15} * x^{15}$$

$$12: \text{return } S_{\text{um}}$$

可以看到,在步骤4、步骤7和步骤10的计算过程中,每一步骤各个计算均独立,故而可以对每一步骤的计算进行并行化处理。同时可以看到,步骤5和6、步骤8和9间是独立的,同样可以并行化处理进行计算。

规定归一化向量 $T = (Kq, (K-1)q, \dots, 2q, q, 1, q, 2q, \dots, (K-1)q, Kq), \varepsilon$ 表示插值小区间长度,即 $\varepsilon = x_{\text{right}} - x_{\text{left}}$ 。将待自举密文 c_1 分别与(1)和(2)式中插值区间左端点密文值和插值区间右端点密文值进行比较。计算 c_1 与 G 中各元素的比较值:

$$v_i = C_{\text{mpn}} \left(\frac{c_1}{\varepsilon \cdot T_i}, \frac{G_i}{\varepsilon \cdot T_i} \right) \quad (3)$$

式中, $0 \leq i \leq 2K$ 。 v_i 表示待自举密文 c_1 与 G_i 之间的明文大小关系, $D_{\text{ec}_{\text{sk}}}(c_1) > D_{\text{ec}_{\text{sk}}}(G_i)$ 时, $v_i = E_{\text{nc}_{\text{pk}}}(1)$, 否则 $v_i = E_{\text{nc}_{\text{pk}}}(0)$ 。同样地,计算 c_1 与 E 中各元素的比较值

$$w_i = C_{\text{mpn}} \left(\frac{E_i}{\varepsilon \cdot T_i}, \frac{c_1}{\varepsilon \cdot T_i} \right) \quad (4)$$

式中, $0 \leq i \leq 2K$ 。 w_i 表示待自举密文 c_1 与 E_i 之间的明文大小关系, $D_{\text{ec}_{\text{sk}}}(c_1) < D_{\text{ec}_{\text{sk}}}(E_i)$ 时, $w_i = E_{\text{nc}_{\text{pk}}}(1)$, 否则 $w_i = E_{\text{nc}_{\text{pk}}}(0)$ 。

计算区间判断算法的结果

$$c_{\text{mpres}} = \langle v, u \rangle \quad (5)$$

若 $c_{\text{mpres}_i} = E_{\text{nc}_{\text{pk}}}(1)$, 则代表待自举密文属于第 i 个区间;若 $c_{\text{mpres}_i} = E_{\text{nc}_{\text{pk}}}(0)$, 则代表待自举密文不属于第 i 个区间。

3.1.3 基于余数系统的新鲜密文计算

将待自举密文 c_1 分别代入插值多项式 $L_i(x)$, $i = 0, 1, \dots, 2K$, 得到插值结果向量 $L_{\text{res}} = (L_0(c_1), L_1(c_1), \dots, L_{2K}(c_1))$ 。

对于较小的模 q , 将其替换为新的模 Q , 满足 $Q \gg q$, 并且满足

$$[\langle c_1, s_k \rangle]_Q = \langle c_1, s_k \rangle \pmod{x^N + 1} \quad (6)$$

计算自举后的新鲜密文

$$c_{\text{t, fresh}} = \langle L_{\text{res}}, c_{\text{mpres}} \rangle \quad (7)$$

3.2 模乘模逆运算实现

同态加密方案在实现过程中模乘模逆运算占据了大量的运算时间。余数系统可以高效地将大数运算转化为多个互相独立的小数进行等效运算, 可以为基于近似同态加密方案的实现提供良好效率支持。但因在使用余数系统进行运算过程中, 需要尽可能地避免进行除法运算, 这就使得在余数系统下进行取模操作变得困难。可以使用蒙哥马利模乘算法来实现余数系统中的除法运算。

蒙哥马利模乘算法可以不使用除法运算就实现模约减运算。需要将模乘算法改进成余数系统下的蒙哥马利模乘算法。对于基 $A = (a_1, a_2, \dots, a_i, \dots, a_l)$ 和基 $B = (b_1, b_2, \dots, b_i, \dots, b_l)$, 有运算元素 X 和 Y , $[X]_{A \cup B}$ 和 $[Y]_{A \cup B}$ 为 X 和 Y 在基 A 和基 B 下的表示, x_k 和 y_k 为 X 和 Y 在基 A 和基 B 下的第 k 个元素。 $|-N_i^{-1}| = (-N_i^{-1}) \pmod{a_i}$ 为 N_i 在模 a_i 下的模逆, $|-H_j^{-1}| = (-H_j^{-1}) \pmod{b_j}$ 为 H_j 在模 b_j 下的模逆。 N 为大素数 X 和 Y 的模数, $X, Y < 2N$, $H = \prod_{i=1}^l a_i$ 为蒙哥马利模乘因子。余数系统下的蒙哥马利模乘运算的具体步骤见算法2。

算法2 RNSMonModMult(X, Y, N)

输入: $[X]_{A \cup B}$, $[Y]_{A \cup B}$ 和 N

输出: $r = XYH^{-1} \pmod{N}$

1: for ($i = 1; i \leq l; ++i$)

2: $w_i = (x_i \times y_i) \pmod{a_i}$

3: $z_i = (w_i \times |-N_i^{-1}|_{a_i}) \pmod{a_i}$

4: $z_i = \text{Conv}(z_i)$

5: for ($j = 0; j \leq l; ++j$)

6: $w_j = (x_j \times y_j) \pmod{b_j}$

7: $f_j = (w_j + z_j \times N_j) \pmod{b_j}$

8: $r_j = (f_j \times |-H_i^{-1}|_{b_j}) \pmod{b_j}$

9: $r_i = \text{Conv}(r_i)$

10: return (r_i, r_j)

可以通过费马小定理在素域上完成模逆运算的高效实现。算法3为使用费马小定理进行模逆运算。

算法3 FermModInv(g)

输入:非零元素 g

输出:模逆元素 g^{-1}

1: $c = g^{q-2}$

2: return c

相较于使用费马小定理来实现模逆运算,在普遍情况下,扩展欧几里得算法实现模逆运算更为高效。但扩展欧几里得算法需要比较运算以及模运算,这些运算在基于近似计算的同态加密方案中是较难计算的,实现复杂耗时高。因此在实际计算模逆运算时,使用费马小定理来实现模逆运算是基于近似同态加密方案中更为高效实际的计算方法。

4 噪声与安全性分析

4.1 噪声分析

对于第 l 层的加密后的密文 c_l ,其对应明文为 m ,重缩放算法返回一个第 $l-1$ 层的密文,且该密文对应明文为 $q_l^{-1} \cdot m$,并且满足 $q_l^{-1} \cdot m \approx q^{-1} \cdot m$ 。这一约等于过程相当于引入了新的计算误差。现对基于余数系统的近似同态加密方案噪声进行分析。

记 $\zeta = e^{-\pi i/N}$, $K = Q[X]/(X^N + 1)$ 上的正规嵌入映射定义为 $\mathbf{a}(X) \rightarrow (\mathbf{a}(\zeta), \mathbf{a}(\zeta^3), \dots, \mathbf{a}(\zeta^{2N-1}))$,其无穷范数被称为正规嵌入范数, $\|\mathbf{a}\|_\infty^{\text{can}} = \|\sigma(\mathbf{a})\|_\infty$ 。对于解码映射 τ 和任意 $\mathbf{a} \in K$,有 $\|\mathbf{a}\|_\infty^{\text{can}} = \|\tau(\mathbf{a})\|_\infty$ 。

对于正整数 h ,私钥的随机分布 χ_{key} 为 $\text{HWT}(h)$ 。噪声的随机分布 χ_{err} 为 $\text{DG}(\sigma^2)$,其中 σ^2 表示 \mathbf{Z}^N 上的方差。加密密钥的随机分布 χ_{enc} 为 $\text{ZO}(\rho)$ 。

$\mathbf{a}(X)$ 为经过随机采样得到的多项式, $\mathbf{a}(\zeta)$ 为系数向量 \mathbf{a} 的内积,向量 $(1, \zeta, \dots, \zeta^{N-1})$ 的欧拉范数为 \sqrt{N} ,对于 \mathbf{a} 中的每一个系数,其方差为 σ^2 ,随机变量 $\mathbf{a}(\zeta)$ 的方差为 $V_{\text{err}} = \sigma^2$,当 \mathbf{a} 随机采样自 $U(R_q)$ 时, $\mathbf{a}(\zeta)$ 的方差为 $V_{\text{enc}} = N/2$,当 \mathbf{a} 随机采样自 χ_{enc} 时, $\mathbf{a}(\zeta)$ 的方差为 $V_q = q^2 N/12$,当 \mathbf{a} 随机采样自 χ_{key} 时, $\mathbf{a}(\zeta)$ 的方差为 $V_{\text{key}} = h$ 。在复平面上看,因 $\mathbf{a}(\zeta)$ 为多个独立同分布随机变量的集合,可以将其看作高斯随机变量。对于单位根 ζ^j 的任意计算都有相同

的方差,因此当 \mathbf{a} 中的每一个系数的方差为 V 时,可以用 $6 \cdot \sqrt{V}$ 作为 \mathbf{a} 的规范嵌入范数的高置信上界。对于2个服从高斯分布的独立随机变量,其方差分别为 V_1 和 V_2 ,则这2个变量的乘积的高置信上界为 $16 \cdot \sqrt{V_1 V_2}$ 。

对于基于余数系统的近似同态加密方案,其加密阶段并没有使用到近似模交换算法,故而加密过程的噪声上界与原始近似同态加密方案一样,加密过程的噪声上界为 $B_{\text{Enc}} = 8\sqrt{2}\sigma N + 6\sigma\sqrt{N} + 16\sigma\sqrt{hN}$ 。

同态加法操作也没用使用到近似模交换算法,故而同态加法过程的噪声上界与原始近似同态加密方案一样,对于处于同层的密文 (c_1, l, v_1, B_1) 和 (c_2, l, v_2, B_2) ,对应明文分别为 m_1 和 m_2 ,有同态加法运算 $c_{\text{add}} = \text{Add}(c_1, c_2)$,则同态加法后的密文形式为 $(c_{\text{add}}, l, v_1 + v_2, B_1 + B_2)$ 。

对于重缩放操作, $c_l = (c_l^{(j)} = (c_0^{(j)}, c_1^{(j)}))_{0 \leq j \leq l} \in \prod_{j=0}^l R_{q_j}^2$ 为第 l 层的待重缩放密文, $c'_l = (c_l'^{(j)} = (c_0'^{(j)}, c_1'^{(j)}))_{0 \leq j \leq l-1} = \text{RS}(c_l)$ 为重缩放过程的输出结果,其中 $c_i^{(j)} = q_l^{-1} \cdot (c_i^{(j)} - c_i^{(l)})$, $i = 0, 1, 0 \leq j \leq l-1$ 。多项式 $c_i \in R_{Q_L}$ 满足 $[c_i]_{c_l} = (c_i^{(0)}, \dots, c_i^{(l)})$,那么可以计算 $c'_i = q_l^{-1} \cdot (c_i - [c_i]_{q_l}) = \lfloor q_l^{-1} \cdot c_i \rfloor$,则 $[c_i]_{c_{l-1}} = (c_i'^{(0)}, \dots, c_i'^{(l-1)})$ 。因此可以得到 $[\langle c'_l, s_k \rangle]_{Q_{l-1}} = q_l^{-1} \cdot [\langle c_l, s_k \rangle]_{Q_l} + e_{R_S}$,其中 e_{R_S} 满足 $\|e\|_\infty^{\text{can}} \leq B_{R_S} = \sqrt{N/3} \cdot (3 + 8\sqrt{h})$ 。

对于同态乘法运算,2个同层的密文 c_l 和 c'_l ,记层数为 l 。同态乘法计算先得到的结果为 $(d_0, d_1, d_2) \in R_{Q_l}^3$,满足 $d_0 + d_1 \cdot s + d_2 \cdot s^2 \equiv \langle c_l, s_k \rangle$ 。接着计算 $\tilde{d}_2 = d_2 + Q_l \cdot e$,满足 $\|\tilde{d}_2\|_\infty = (1/2) \cdot (l+1) \cdot Q_l$ 。可以假设整数多项式 \tilde{d}_2 为 R_{Q_l} 上的 $l+1$ 个独立的均匀分布随机变量的和,则 \tilde{d}_2 的方差为 $V = (1/2) \cdot (l+1) \cdot (Q_l^2 \cdot (N/12))$ 。因此,同态计算密钥 evk 的前 $k+l+1$ 个组成元素可以看作是在模数为 $P \cdot Q_l$ 下对 $P \cdot s^2$ 的加密。最终的输出结果 \tilde{c}'_l 为 $P \cdot \tilde{d}_2 \cdot s^2 \equiv P \cdot d_2 \cdot s^2 \pmod{P \cdot Q_l}$,因此密文噪声上界为 $16 \cdot \sqrt{V} \cdot \sqrt{N\sigma^2} = 8\sqrt{(l+1)/6} \cdot Q_l \cdot \sigma N = \sqrt{(l+1)/2} \cdot B_{K_S} \cdot Q_l$ 。通过模约减算法进行模约

减后,返回 $\hat{c}_1 \in R_{Q_l}^2$, 满足 $P \cdot \hat{c}_1 \approx \tilde{c}_1$ 。对于误差 $P \cdot \hat{c}_1 - \tilde{c}_1$, 可以将其看作是 R_p 上的 k 个独立均匀分布随机变量的和, 所以其方差为 $k \cdot V_p = k \cdot P^2 \cdot (N/12)$ 。最后, 通过除以 P , 得到模约减噪声 $k \cdot P \cdot (N/12)$ 。所以, 结果 \hat{c}_1 为一个噪声上界为 $\sqrt{(l+1)/2} \cdot P^{-1} \cdot B_{K_S} \cdot Q_l + \sqrt{k} \cdot B_{R_S}$, 对 $d_2 \cdot s^2$ 的加密结果。

4.2 安全性分析

CKKS 一系列加密方案都是基于错误学习问题 (LWE 问题) 设计的同态加密方案, 即其安全性可以归约到 LWE 困难问题。下面先给出同态加密方案的 IND-CPA 安全性定义和 IND-CPA^D 安全性定义, 在此基础上对本文方案的安全性进行分析。

对于公钥加密方案, 语义安全等同于选择明文攻击下的不可区分性 (indistinguishability under chosen plaintext attacks, IND-CPA) 安全。定义 1 通过安全性验证方式给出了 IND-CPA 安全的定义。在安全性验证试验中称同态加密方案拥有者为挑战者, 对同态加密方案进行攻击的行为方为攻击者 E_{ve} 。

定义 1 IND-CPA 安全

$E = (K_{eyGen}, E_{nc}, D_{ec}, E_{val})$ 为一个同态加密方案。定义安全性验证试验过程 $\text{expr}_b^{\text{IND-CPA}}$, 其中 b 可取 0 或 1, 安全性验证试验过程如下

$$\begin{aligned} \text{expr}_b^{\text{IND-CPA}}(1^\kappa) : & (s_k, p_k, e_k) \leftarrow K_{eyGen}(1^\kappa) \\ & (x_0, x_1) \leftarrow E_{ve}(1^\kappa, p_k, e_k) \\ & c_t \leftarrow E_{ncpk}(x_b) \\ & b' \leftarrow E_{ve}(c_t) \\ & \text{return } b' \end{aligned}$$

那么定义 IND-CPA 安全性验证试验的优势为 $Adv_{\text{IND-CPA}} = |\Pr\{\text{expr}_0^{\text{IND-CPA}}(1^\kappa) = 1\} - \Pr\{\text{expr}_1^{\text{IND-CPA}}(1^\kappa) = 1\}|$, 则当且仅当 $Adv_{\text{IND-CPA}}$ 的值关于 κ 可忽略时, 该同态加密方案为 IND-CPA 安全的。

同态加密方案因可以进行密文同态运算而不同于其他类型的加密方案。在安全性度量时这一特性也需要被考虑到。Li 等^[14]发现, 针对基于近似计算的同态加密方案 CKKS17, 该方案即便是满足 IND-CPA 安全下, 仍然无法完全防御被动攻击。对此, Li 等提出了带解密预言机的选择明文攻击下的不可区分性 (indistinguishability under chosen plaintext attacks with decryption oracles, IND-CPA^D) 安全的

概念。

定义 2 IND-CPA^D 安全

$E = (K_{eyGen}, E_{nc}, D_{ec}, E_{val})$ 为一个明文空间为 M , 密文空间为 C 的同态加密方案。定义安全性验证试验过程 $\text{expr}_b^{\text{IND-CPA}^D}$, 其中 b 可取 0 或 1。用三元组 $(M \times M \times C)$ 来表示状态, $*$ 代表可以不出现, 也可以出现一次或者多次。在该安全性验证试验中, 攻击者具有记录状态三元组 $S \in (M \times M \times C)^*$ 的能力, $|S|$ 表示 S 中状态的个数, $S[j] \cdot m_0, S[j] \cdot m_1$ 和 $S[j] \cdot m_c$ 分别代表 S 中下标为 j 的相应元素。攻击者可以使用如下预言机:

加密预言机 $E_{pk}(m_0, m_1)$: 向加密预言机输入明文消息 m_0 和 m_1 , 计算 $c \leftarrow E_{ncpk}(m_b)$, 将状态三元组 (m_0, m_1, c) 加入到 S 中, 并返回密文 c 给攻击者 E_{ve} 。

同态运算预言机 $H_{ek}(g, J)$: g 为映射关系 $M^h \rightarrow M, J$ 为下标序列集合, $J = (j_1, j_2, \dots, j_k) \in 1, 2, \dots, |S|!^k$ 。进行与映射关系 g 对应的同态运算得到密文 $c \leftarrow E_{valpk}(g, S[j_1] \cdot c, S[j_2] \cdot c, \dots, S[j_k] \cdot c)$, 并将三元组 $(g(S[j_1] \cdot m_0, S[j_2] \cdot m_0, \dots, S[j_k] \cdot m_0), g(S[j_1] \cdot m_1, S[j_2] \cdot m_1, \dots, S[j_k] \cdot m_1), c)$ 加入到 S 中。返回密文 c 给攻击者 E_{ve} 。

解密预言机 $D_{sk}(j)$: 对于下标 $j \leq |S|$, 判断 $S[j] \cdot m_0 = S[j] \cdot m_1$ 的真伪, 若为真, 返回 $D_{eck}(S[j] \cdot c)$ 给攻击者 E_{ve} , 若为假, 则返回错误信息给攻击者。

安全性验证试验过程如下

$$\begin{aligned} \text{expr}_b^{\text{IND-CPA}^D}(1^\kappa) : & (s_k, p_k, e_k) \leftarrow K_{eyGen}(1^\kappa) \\ & S = [] \\ & b' \leftarrow E_{ve}^{E_{pk}, H_{ek}, D_{sk}}(1^\kappa, p_k, e_k) \\ & \text{return } b' \end{aligned}$$

即定义 IND-CPA^D 安全性验证试验的优势为 $|\Pr\{\text{expr}_0^{\text{IND-CPA}^D}(1^\kappa) = 1\} - \Pr\{\text{expr}_1^{\text{IND-CPA}^D}(1^\kappa) = 1\}|$ 则当且仅当 $Adv_{\text{IND-CPA}^D}$ 的值可以忽略时, 该同态加密方案为 IND-CPA^D 安全的。

可以观察到, 本文方案的加密过程为: 随机采样得到 $s \leftarrow \text{HWT}(h), a \leftarrow R_{q_l}$ 和 $e \leftarrow \text{DG}(\sigma^2)$, 计算 $b = -as + e \pmod{q_l}$, 公钥 $pk = (b, a)$, 随机采样得到 $v \leftarrow \text{ZO}(0.5)$ 以及 $e_0, e_1 \leftarrow \text{DG}(\sigma^2)$, 对于明文 m , 加密后的密文为 $c = v \cdot pk + (m + e_0, e_1) \pmod{q_l}$ 。加密过程就是 LWE 问题中向量的构造过程, 因为 LWE 问题是困难的, 所以方案的安全性受到了

保障。

由定义1可知,对于 $(s_k, p_k, e_k) \leftarrow K_{\text{eyGen}}(1^\lambda)$,攻击者 E_{ve} 可以获得已经公布的公钥和计算密钥以及安全参数,那么攻击者可以选定 $(x_0, x_1) \leftarrow E_{\text{ve}}(1^\lambda, p_k, e_k)$ 。对于 $b \in \{0, 1\}$,明文 x_b 经过加密后的结果为 c_i ,由判定LWE问题是困难问题可知,攻击者无法以明显的优势进行判定,即对于 $b' \leftarrow E_{\text{ve}}(c_i)$, $\text{Adv}_{\text{IND-CPA}} = |\Pr\{\text{expr}_0^{\text{IND-CPA}}(1^\lambda) = 1\} - \Pr\{\text{expr}_1^{\text{IND-CPA}}(1^\lambda) = 1\}| = 0$,该方案是满足IND-CPA安全的。

但是对于IND-CPA^D安全的安全,选择加密明文为0,并且加密后不执行同态运算而是直接解密,多次重复该操作,就可以得到关于私钥 s_k 的线性方程组,通过高斯消元法即可结束 s_k 。上述过程即为对近似计算同态加密的被动恢复密钥攻击。所以CKKS17方案无法满足IND-CPA^D安全。对此本文方案在解密过程中先进行随机采样得到 $e_2 \leftarrow DG(\sigma^2)$,对于密文 $c = (b, a)$,解密形式变换为 $m = b + a \cdot s + e_2 \pmod{q_l}$,而非CKKS17方案中的 $m = b + a \cdot s \pmod{q_l}$ 。由于近似计算特性,解密结果中增加噪声不会影响方案正确性,同时增加噪声后,解密后的明文符合LWE问题形式,无法通过高斯消元法直接获得加密方案的私钥,所以本文方案可以成功防御针对近似计算同态加密方案的被动恢复密钥攻击。根据定义2可知,该方案满足IND-CPA^D安全要求。

本文方案仅是对自举过程进行优化改进,自举过程可以看作是同态加法运算和同态乘法运算的组合,对自举运算的优化并不影响方案的安全性。

5 实验结果和分析

本文实验是基于余数系统的HEAAN库^[6]实现的,运行设备CPU型号为Intel(R) Core(TM) i9-9900K CPU @ 3.60 GHz, 64 GB运行内存,操作系统为Ubuntu 18.04.4 LTS。

表1给出了参数集选择,其中“-”代表了该方案未使用到该参数。表中参数均遵循CKKS推荐参数^[15]选择,可以有效防止格攻击。其中参数 $\log_2 N$ 表示一个CKKS密文能打包 $N/2$ 个明文; $\log_2 p$ 和 $\log_2 q$ 表示自举过程模约减过程使用的参数; $\log_2 Q$ 表示模升操作后密文的模大小; r 是CHK+18a使用

的倍角公式; n 表示比较函数中多项式函数 $f(x)$ 和 $g(x)$ 的次数, d_f 和 d_g 分别表示多项式函数 $f(x)$ 和 $g(x)$ 的迭代次数。

CKKS密文支持SIMD,一个密文打包了 $N/2$ 个明文数据,因此,本文的对比实验中主要采取了耗时和平均耗时2个对比标准来进行自举方案对比,耗时表示的是对 $N/2$ 个明文数据进行自举计算所需的整体时间,而平均耗时表示对每一个明文数据进行自举计算所需的时间。下面将本文提出的优化方案与CHK+18a方案^[5]、HEAAN库^[6]以及方案^[16]进行对比。参数集I~IV为CHK+18a方案中设定的参数,HEAAN库同样按照CHK+18a方案中的参数选择进行设定,参数集V~VIII为应用小间隔差值拟合算法^[16]计算时所需参数,本文基于余数系统的小间隔差值拟合算法所使用的参数集是IX~XII。

CHK+18a、HEAAN、文献[16]方案和本文方案的自举耗时对比记录在表2,其中 Q_1 表示自举后密文模的大小; l_{left} 表示自举后还剩下的乘法次数;图1展示了HEAAN库,文献[16]方案和余数系统下的小间隔差值拟合方案自举实验。

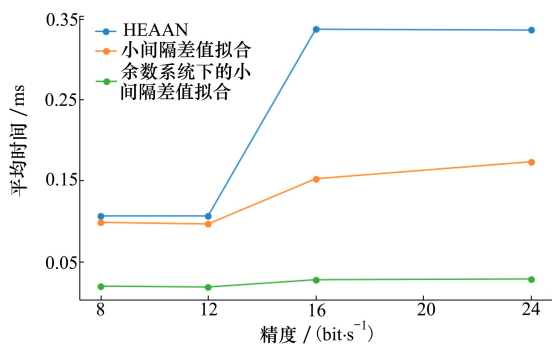


图1 平均耗时对比

I, V和IX这3组参数集有着相同的精度设置和基本相同的模大小,在这3组参数集下,本文提出的余数系统下小间隔差值拟合自举方案运行耗时仅为1 743 ms,平均到每一槽计算时间仅需0.027 ms,是CHK18+a的1/20,约为HEAAN 2.1版本和文献[16]方案的1/5;在更大的参数设置下,即参数集IV, VIII和XII,本文提出的余数系统下的小间隔差值拟合自举方案的运行耗时和平均耗时是CHK+18a方案的1/37,约为HEAAN 2.1版本的1/6,约为文献[16]方案运行耗时的1/6。在各组参数集对比下,本文自举方案比其他3个方案耗时少,并且在更大的参数集设置下,本文方案优势更加明显。

结合图 1 中的曲线变化,当自举精度从 12 比特上升到 16 比特时,HEEAN1.2 和文献[16]方案的耗时有大幅度上升,而基于余数系统的小间隔差值拟合自举方法运行时间虽有上升,但上升比例不大。分析其原因,当计算精度提升时,选取的模数变大,密文位数也随着变多,导致运算量增大,计算耗时上升。余数系统能将大整数拆分为多个小整数进行运

算,这样就避免了大整数乘法因位数上升而导致的运算效率下降这一问题。从实验数据中也可以得到相同的结论,HEEAN1.2 和文献[16]方案在精度上升时,运算耗时有显著的上升,但通过本文方案将大数转化为多个小整数后,精度上升时的运算耗时上升不明显。因此,本文方案能够在合理的耗时下满足高精度的自举计算需求。

表 1 参数集

参数集编号	$\log_2 N$	$\log_2 p$	$\log_2 q$	$\log_2 Q$	精度/比特	r	n	d_f	d_g
I	15	23	-	620	8	6	-	-	-
II	15	27	-	620	12	7	-	-	-
III	16	31	-	1 240	16	7	-	-	-
IV	16	39	-	1 240	24	9	-	-	-
V	15	23	-	620	8	-	4	2	2
VI	15	27	-	620	12	-	4	2	2
VII	16	31	-	1 240	16	-	4	2	2
VIII	16	39	-	1 240	24	-	4	2	3
IX	15	-	55	611	8	-	4	2	2
X	15	-	55	611	12	-	4	2	2
XI	16	-	55	886	16	-	4	2	2
XII	16	-	55	886	24	-	4	2	3

表 2 实验结果对比

方案	参数集	Q_1	l_{left}	耗时/ms	平均耗时/ms
CHK+18a ^[5]	I	202	8	12 300	0.375
	II	64	2	12 500	0.381
	III	631	20	63 000	0.961
	IV	344	8	68 000	1.038
HEEAN ^[6]	I	203	8	3 475	0.106
	II	65	2	3 459	0.106
	III	586	18	22 085	0.337
	IV	345	8	21 860	0.336
文献[16]	V	160	6	3 217	0.098
	VI	80	2	3 157	0.096
	VII	620	20	9 989	0.152
	VIII	165	4	11 308	0.173
本文	IX	-	6	625	0.019
	X	-	2	589	0.018
	XI	-	20	1 743	0.027
	XII	-	4	1 848	0.028

6 结 论

针对基于小间隔插值拟合的近似同态加密自举方法,本文继续探索优化,将其余数系统相结合,提出了一种基于余数系统的小间隔插值拟合自举方法。余数系统可以提高密码方案中模乘运算和模逆运算的计算速度,但应用在近似计算同态加密方案中会遇到模基选择问题。本文根据近似计算同态加密方案的特性选择了更高效模乘运算和模逆运算的实现算法。最终在 16 比特精度, $N=2^{16}$ 时,平均到每一槽的分时计算时间仅需 0.027 ms,相较于原始方案计算效率有显著提升。

在研究过程中发现,当设定较大 N 值时,近似计算同态加密方案所产生的密钥尺寸和密文尺寸较大,对内存的占用极为明显。这就导致了实验过程中对实验设备硬件要求较高,影响了方案的实用性,对此可以进一步探讨近似同态加密过程中的密文压缩技术,通过压缩手段来降低密文尺寸,进而提高方案的实用性。

参考文献:

- [1] CHEON J H, KIM A, KIM M, et al. Homomorphic encryption for arithmetic of approximate numbers[C] // 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, 2017: 409-437
- [2] KIM M, SONG Y, LI B, et al. Semi-parallel logistic regression for GWAS on encrypted data[J]. BMC Medical Genomics, 2020, 13: 1-13
- [3] CHEON J H, KIM D, PARK J H. Towards a practical cluster analysis over encrypted data[C] // International Conference on Selected Areas in Cryptography, Cham, 2019: 227-249
- [4] BOURSE F, MINELLI M, MINIHOLD M, et al. Fast homomorphic evaluation of deep discretized neural networks[C] // 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, 2018: 483-512
- [5] CHEON J H, HAN K, KIM A, et al. Bootstrapping for approximate homomorphic encryption[C] // 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, 2018: 360-384
- [6] CHEON J, HAN K, KIM A, et al. Implementation of bootstrapping for HEAAN[J/OL]. (2022-01-12)[2023-08-09]. <https://github.com/snucrypto/HEAAN>
- [7] CHEN H, CHILLOTTI I, SONG Y. Improved bootstrapping for approximate homomorphic encryption[C] // Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cham, 2019: 34-54
- [8] HAN K, KI D. Better bootstrapping for approximate homomorphic encryption[C] // Cryptographers' Track at the RSA Conference, Cham, 2020: 364-390
- [9] JUNG W, KIM S, AHN J, et al. Over 100x faster bootstrapping in fully homomorphic encryption through memory-centric optimization with GPUs[J]. IACR Trans on Cryptographic Hardware and Embedded Systems, 2021(4): 114-148
- [10] JUTLA C S, MANOHAR N. Sine series approximation of the mod function for bootstrapping of approximate HE[C] // Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cham, 2022: 491-520
- [11] CHEON J H, HAN K, KIM A, et al. A full RNS variant of approximate homomorphic encryption[C] // 25th International Conference, Calgary, AB, Canada, 2019: 347-368
- [12] BOSSUAT J P, MOUCHET C, TRONCOSO-PASTORIZA J, et al. Efficient bootstrapping for approximate homomorphic encryption with non-sparse keys[C] // Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cham, 2021: 587-617
- [13] LEE J W, LEE E, LEE Y, et al. High-precision bootstrapping of RNS-CKKS homomorphic encryption using optimal minimax polynomial approximation and inverse sine function[C] // 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, 2021: 618-647
- [14] LI B, MICCIANCIO D. On the security of homomorphic encryption on approximate numbers[C] // Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cham, 2021: 648-677
- [15] CHEON J, HONG S, KIM D. Remark on the security of CKKS scheme in practice[J/OL]. (2020-12-21)[2023-08-09]. <https://ia.cr/2020/1581>
- [16] 李慧贤, 刘诗源, 沈春. 基于小间隔插值拟合的近似同态加密自举方法: 中国, 2022110358775[P]. 2023-04-07

Small interval interpolation fitting bootstrapping method based on residue number system

LI Huixian¹, WANG Fulei¹, SHEN Chun², LIU Shiyuan¹, PANG Liaojun³

(1.School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China;
2.School of Software, Northwestern Polytechnical University, Xi'an 710072, China;
3.School of Life Science and Technology, Xidian University, Xi'an 710071, China)

Abstract: Aiming at the problem that the bootstrapping time of approximate homomorphic encryption scheme is too long, a small interval interpolation fitting method based on residue system is proposed. In this paper, the sinusoidal function by using interpolating and fitting method between the multiple cells to avoid the increase in bootstrapping time or decrease in calculation accuracy caused by the high degree of fitting polynomial is calculated. And the efficiency of modular multiplication and modular inversion in the calculation process is improved by combining the residual system. Lagrange interpolation polynomial is used to interpolate and fit the sine function among different intervals. The comparison function is implemented by the compound implementation of low-degree polynomials, and an interval judgment algorithm is proposed to identify the interval of the ciphertext. Finally, under the precision of 24 bits, the modular operation time in the bootstrapping process decreased to 8% of the HEAAN. When the number of slots is 65 536, the average module operation time per slot is 0.028 ms.

Keywords: fully homomorphic encryption; approximate arithmetic; bootstrapping; residue number system; Lagrange interpolation

引用格式: 李慧贤, 王富磊, 沈春, 等. 基于余数系统的小间隔插值拟合自举方法[J]. 西北工业大学学报, 2024, 42(5): 969-978

LI Huixian, WANG Fulei, SHEN Chun, et al. Small interval interpolation fitting bootstrapping method based on residue number system[J]. *Journal of Northwestern Polytechnical University*, 2024, 42(5): 969-978 (in Chinese)