基于 NTRU 的多密钥同态代理重加密方案及其应用

李瑞琪[1,2]，贾春福[1,2]，王雅飞[1,2]
（1. 南开大学网络空间安全学院，天津 300350；2. 天津市网络与数据安全技术重点实验室，天津 300350）

摘 要：为了提高同态加密算法在多用户云计算场景下的实用性，构造了一个基于 NTRU 的多密钥同态代理重加密方案。首先利用密文扩张思想提出了一种新的 NTRU 型多密钥同态密文形式，并基于此设计了相应的同态运算和重线性化过程，从而形成一个支持分布式解密的 NTRU 型多密钥同态加密方案；然后借助于密钥交换思想设计了重加密密钥和重加密过程，将代理重加密功能集成到该 NTRU 型多密钥同态加密方案中。所提方案保留了多密钥同态加密和代理重加密的特性，而且在用户端的计算开销较低。将所提方案应用于联邦学习中的隐私保护问题并进行了实验，结果表明，所提方案基本不影响联邦训练的准确率，加解密、同态运算和重加密等过程的计算开销也可接受。

关键词：同态加密；代理重加密；多密钥；云计算；联邦学习
中图分类号：TP309.7
文献标识码：A
DOI: 10.11959/j.issn.1000−436x.2021023

Multi-key homomorphic proxy re-encryption scheme based on NTRU and its application

LI Ruiqi[1,2], JIA Chunfu[1,2], WANG Yafei[1,2]
1. College of Cyber Science, Nankai University, Tianjin 300350, China
2. Tianjin Key Laboratory of Network and Data Security Technology, Tianjin 300350, China

Abstract: To improve the practicability of homomorphic encryption in the application of multi-user cloud computing, a NTRU-based multi-key homomorphic proxy re-encryption (MKH-PRE) scheme was constructed. Firstly, a new form of NTRU-based multi-key ciphertext was proposed based on the idea of ciphertext extension, and the corresponding homomorphic operations and relinearization procedure were designed on the basis of this new ciphertext form, so that a NTRU-based multi-key homomorphic encryption (MKHE) scheme which supported distributed decryption was constructed. Then, resorting to the idea of key switching, the re-encryption key and re-encryption procedure were put forward such that the functionality of proxy re-encryption (PRE) was integrated to this new NTRU-based MKHE scheme. The MKH-PRE scheme preserved the properties of MKHE and PRE, and had a better performance on the client side. The scheme was applied to the privacy-preserving problems in federated learning and an experiment of the application was carried out. The results demonstrate that the accuracy of learning is scarcely affected by the encryption procedure and the computational overhead of this MKH-PRE scheme is acceptable.

Keywords: homomorphic encryption, proxy re-encryption, multi-key, cloud computing, federated learning

收稿日期：2020−10−14；**修回日期**：2020−12−16
通信作者：贾春福，cfjia@nankai.edu.cn
基金项目：国家重点研发计划基金资助项目（No.2018YFA0704703）；国家自然科学基金资助项目（No.61972215, No.61702399, No.61972073）；天津市自然科学基金资助项目（No.20JCZDJC00640）
Foundation Items: The National Key Research and Development Program of China (No.2018YFA0704703), The National Natural Science Foundation of China (No.61972215, No.61702399, No.61972073), The Natural Science Foundation of Tianjin (No.20JCZDJC00640）

1 引言

全同态加密（FHE, fully homomorphic encryption）是一种新型密码学原语，其支持在加密信息上进行任意函数运算，并且解密后得到的结果与在明文上执行相应运算的结果一致。同态加密的思想最初由 Rivest 等[1]在 1978 年提出（最初的概念被称作 privacy homomorphism）。2009 年，Gentry[2-3]构造出了第一个全同态加密方案。在 Gentry 的开创性工作之后，一系列关于同态加密的研究成果相继出现[4-10]。同态加密的特性使其能够广泛应用于多种云计算场景，如外包计算。

考虑如下的外包计算场景。多个数据拥有者想要联合进行某种计算（例如机器学习或数据挖掘等），于是各自将数据加密后发送到云端。云服务器在密文上进行指定的计算，然后将结果发送给接收者。接收者既可以是数据提供者，也可以是数据提供者之外的第三方。大多数情况下，数据提供者希望自己的数据对其他提供者和接收者是保密的，显然单密钥同态加密方案无法满足此要求。因此，需要一个具有如下性质的同态加密方案。1) 每个数据提供者能够用自己的公钥进行加密，并且这些密文能同时参加同态计算；2) 计算结果的接收者即使获得了（其他）数据提供者的原始密文也无法对其进行解密。

多密钥同态加密（MKHE, multi-key homomorphic encryption）支持多方用户以各自的密钥对消息进行加密，得到的密文可以一起参与同态运算。López-Alt 等[8]首次提出了 MKHE 的概念并构造了第一个 MKHE 方案，此后一系列 MKHE 方案被提出[11-18]。MKHE 方案能够满足性质 1)，并且当接收者是数据提供者时，借助于分布式解密，MKHE 方案也能够满足性质 2)。然而当接收者是数据提供者之外的第三方时，只有 2 种解密方式：一是把所有数据提供者的私钥收集给接收者；二是数据提供者和接收者之间进行交互。前一种情况在现实中不太可能发生，后一种情况则需要数据提供者在线才能进行解密，因此在这种场景下 MKHE 的使用仍存在问题。

为了解决上述问题，Yasuda 等[19]提出了一个新概念——多密钥同态代理重加密（MKH-PRE, multi-key homomorphic proxy re-encryption）。MKH-PRE 方案既有 MKHE 的特性，也具有 PRE 的性质。此类方案允许数据提供者用自己的公钥加密数据来进行多密钥同态计算，同时也允许对同态计算得到的密文进行代理重加密，将结果密文转换为只能由接收者解密的新密文。重加密功能的加入使当接收者是数据提供者之外的第三方时能够满足性质 2)。由于数据提供者预先生成了重加密密钥并发送到云端，因此接收者在解密时不需要与提供者进行交互，即提供者不需要实时在线。Yasuda 等[19]也给出了一个基于 PS16 方案[16]的 MKH-PRE 实例（简称 YKHK18 方案），但该实例不适用于实际的外包计算场景，这是因为在 PS16 多密钥同态加密方案中，明文空间只有 1 bit，大大降低了实用性；同时，PS16 方案需要在加密时生成冗余密文，给用户带来了额外的开销，并且整个方案难以实现。

NTRU（number theory research unit）是由 Hoffstein、Pipher 和 Silverman[20]于 1998 年提出的一种基于格的公钥加密体制，此后的研究工作[21-23]构造了可证明安全的 NTRU 加密方案，这些方案的安全性能够规约于标准的密码学假设 RLWE（ring learning with error）。相较于传统公钥密码，NTRU 具有显著的计算性能优势，而且被认为能够抵抗量子计算攻击。除此之外，NTRU 加密体制自提出后一直备受关注的原因还包括其能够用于构造具有功能性的密码学原语。近年来，一些研究关注于利用 NTRU 算法构造多密钥同态加密方案[8,24-25]，但现有的 NTRU 型 MKHE 方案在实际应用时仍存在一定的问题。此类 MKHE 方案需要将所有数据提供者的私钥收集给接收者才能解密，这意味着当接收者是数据提供者时，需要多轮协议才能完成解密；当接收者是第三方时，需要将所有私钥提供给第三方。NTRU 算法也能够用于构造代理重加密方案[26-27]，这些方案可以将密文转换成接收者能够解密的密文，因而适用于接收者为提供者之外的第三方的情况，但 PRE 并不支持密文计算。综上所述，鉴于 NTRU 能构造多种功能性密码学原语以及上述方案在应用中存在局限性，可以利用 NTRU 算法构造适合多种外包计算场景的多密钥同态代理重加密方案。

本文构造了一个基于 NTRU 的多密钥同态代理重加密方案，并给出了该方案的一个实际应用。本文主要的研究工作如下。

1) 利用密文扩张的思想设计了一种新的

NTRU 型多密钥同态密文形式，并以此为基础设计了相应的同态运算和重线性化过程，从而得到了一个支持分布式解密的 NTRU 型 MKHE 方案。

2) 利用密钥交换的思想，在此NTRU型MKHE方案中加入了代理重加密的功能，得到了一种新的基于 NTRU 的 MKH-PRE 方案。新方案同时保留了 MKHE 和 PRE 的特性，可根据使用场景选取功能。

3) 本文方案在用户端的开销相对较低，并且支持加密多项式而不是 1 bit，从而支持并行化操作，因此相较此前的 MKH-PRE 方案更具实用性。

4) 将本文方案进行了实现，并应用于一个实际场景——联邦学习。实验结果表明，本文方案的使用基本不会影响联邦学习的准确性，加解密、同态运算、重加密等过程的计算开销也是可接受的。

2　基础知识

本文使用加粗大写字母表示矩阵，例如 $\boldsymbol{M}$；使用加粗小写字母表示（行）向量，例如 $\boldsymbol{v}$；$\boldsymbol{v}[i]$ 表示向量 $\boldsymbol{v}$ 的第 i 个分量。对于一个实数 r，$\lfloor r \rceil$ 表示四舍五入取整。$x \leftarrow D$ 表示从分布 D 中随机抽取样本 x。对于有限集 S，$U(S)$ 表示 S 上的均匀分布。若一簇分布 $\{\chi_n\}_{n\in\mathbb{N}}$ 满足 $\Pr\limits_{e\leftarrow\chi_n}[\|e\|_\infty > B] \leqslant \mathrm{negl}(n)$，则称其为 B 界分布。

本文方案中涉及的所有运算均在环 $R=\mathbb{Z}[X]/\Phi_m(x)$ 中进行，其中 $\Phi_m(x)$ 是分圆多项式，次数为 $n=\varphi(m)$。令 $R_q = R/qR$，其中元素的系数都在 $\left\{-\left\lfloor\frac{q}{2}\right\rfloor,\cdots,\left\lfloor\frac{q}{2}\right\rfloor\right\}$ 内（R_2 中元素的系数都在 $\{0,1\}$ 中）。定义环 R 中元素 $a=a_{n-1}x^{n-1}+a_{n-2}x^{n-2}+\cdots+a_1x+a_0$ 的无穷范数为 $\|a\|_\infty=\max|a_i|$。对于任意的 $a,b\in R$，有 $\|a+b\|_\infty \leqslant \|a\|_\infty+\|b\|_\infty$，$\|ab\|_\infty \leqslant \delta\|a\|_\infty\|b\|_\infty$，其中 δ 被称作环常数。

2.1　RLWE 问题与 DSPR 假设

环带错学习问题（RLWE, ring learning with error）是由 Lyubashevsky 等[28]提出的。参数为 n,q,χ,ς 的 RLWE 分布定义为：随机选取 R_q 中的多项式 $s\leftarrow\varsigma$，$a\leftarrow U(R_q)$，$e\leftarrow\chi$，输出 $(a,b=as+e)$。判别版本的 RLWE 假设（记作 $\mathrm{DRLWE}_{n,q,\chi,\varsigma}$）定义为：从 $\mathrm{RLWE}_{n,q,\chi,\varsigma}$ 分布中抽取的多项式个样本 (a_i,b_i) 与从 $U(R_q)$ 中抽取的相同个数的样本之间是统计上不可区分的。

López-Alt 等[8]介绍了判别小多项式比（DSPR, decisional small polynomial ratio）问题，其定义如下。令 q 为素数，ϕ 是环 R 上的分布。DSPR 问题（记作 $\mathrm{DSPR}_{q,\phi}$）是区分如下 2 个分布：1) $h=gf^{-1}$ 的分布，其中 f 和 g 是从分布 ϕ 中随机选取的（要求 f 在 R_q 上可逆）；2) R_q 上的均匀分布。

此前有关 NTRU 的研究[21-23]已经证明，当分布 ϕ 为离散高斯分布 $D_{\mathbb{Z}^n,r}$ 且 $r>\sqrt{q}\mathrm{poly}(n)$ 时，$\mathrm{DSPR}_{q,\phi}$ 问题是困难的（即分布 1)和 2)统计上不可区分），然而为了能进行多密钥同态计算，需假设当 ϕ 的标准差较小时，$\mathrm{DSPR}_{q,\phi}$ 问题仍是困难的。

2.2　工具向量

文献[29]中介绍了工具矩阵（Gadget）$\boldsymbol{G}$ 及其对应的逆函数 $\boldsymbol{G}^{-1}(\cdot)$。本文中，令 $\boldsymbol{g}:=(g_i)\in\mathbb{Z}^d$ 为工具向量，对应的分解函数为 $\boldsymbol{g}^{-1}:R_q\to R^d$。该函数将 R_q 中的一个元素 a 分解为一个向量 $\boldsymbol{u}=(u_0,\cdots,u_{d-1})\in R^d$，满足 $a=\sum\limits_{i=0}^{d-1}g_iu_i \pmod q$，并且每个 u_i 都是小多项式。对于向量 $\boldsymbol{v}\in R_q^d$，$\boldsymbol{g}^{-1}(\boldsymbol{v})$ 表示将 $\boldsymbol{g}^{-1}$ 应用到 $\boldsymbol{v}$ 的每一个分量 $\boldsymbol{v}[i]$ 上，从而得到矩阵 $\boldsymbol{V}\in R^{d\times d}$，满足 $\boldsymbol{gV}=\boldsymbol{v} \pmod q$。

3　多密钥同态代理重加密

本节给出多密钥同态代理重加密的形式化定义及其 IND-CPA 安全性的定义。

3.1　MKH-PRE 的定义

令 $\mathcal{M}$ 为明文空间，设每个参与同态运算的用户都有一个 id，每个同态密文都伴随着一个用户 id 集合，用来记录计算过程中涉及的所有用户。“新鲜的”密文所对应的集合 T 中只包含一个元素，即 $T=\{\mathrm{id}\}$，而经过多密钥同态运算后的密文对应的集合会变为 $T=\{\mathrm{id}_1,\mathrm{id}_2,\cdots,\mathrm{id}_k\}$。一个多密钥同态代理重加密方案包括如下算法。

$\mathrm{pp}\leftarrow\mathrm{Setup}(1^\lambda)$：输入安全参数 λ，输出公共参数 pp。

$(\mathrm{pk},\mathrm{sk})\leftarrow\mathrm{KeyGen}(\mathrm{pp})$：输入公共参数 pp，输出公钥 pk 和私钥 sk。

$c\leftarrow\mathrm{Enc}(\mathrm{pk},m)$：输入消息明文 $m\in\mathcal{M}$ 和公钥 pk，输出密文 c。

$m \leftarrow \mathrm{Dec}(\{\mathrm{sk}_{\mathrm{id}}\}_{\mathrm{id}\in T}, c)$：输入密文 c 及其对应的运算参与方的私钥 $\{\mathrm{sk}_{\mathrm{id}}\}_{\mathrm{id}\in T}$，输出明文 m。

$c_{\mathrm{Eval}} \leftarrow \mathrm{Eval}(\mathcal{C}, \{c_1,\cdots,c_k\}, \{\mathrm{pk}_{\mathrm{id}}\}_{\mathrm{id}\in T})$：输入电路 $\mathcal{C}$、密文 $c_1,\cdots,c_k$（对应的用户 id 集合分别为 $T_1,\cdots,T_k$）和公钥 $\{\mathrm{pk}_{\mathrm{id}}\}_{\mathrm{id}\in T}$（其中 $T = T_1 \cup T_2 \cup \cdots \cup T_k$），输出密文 c_{Eval}（对应的用户 id 集合为 T）。

$\mathrm{rk}_{i\to j} \leftarrow \mathrm{RKGen}(\mathrm{sk}_i, \mathrm{pk}_j)$：输入私钥 sk_i 和公钥 pk_j，输出重加密密钥 $\mathrm{rk}_{i\to j}$。

$c_{\mathrm{ReEnc}} \leftarrow \mathrm{ReEnc}(\{\mathrm{rk}_{\ell\to k}\}_{\ell\in T}, c)$：输入密文 c 和重加密密钥 $\{\mathrm{rk}_{\ell\to k}\}_{\ell\in T}$（其中 T 是密文 c 对应的用户 id 集合），输出密文 c_{ReEnc}（对应的用户 id 为 k）。

一个 MKH-PRE 方案应当满足以下性质。

正确性。令 $c_1,\cdots,c_s$ 为密文，m_i 为 c_i 对应的明文，$\{\mathrm{sk}_{\mathrm{id}}\}_{\mathrm{id}\in T_i}$ 为 c_i 对应的私钥。令 $\mathcal{C}:\mathcal{M}^s \to \mathcal{M}$ 为作用于 $m_1,\cdots,m_s$ 的电路，c_{Eval} 为同态计算电路 $\mathcal{C}$ 后得到的密文。令 rk 为重加密密钥，c_{ReEnc} 为对 c_{Eval} 重加密后得到的密文，其对应的私钥为 sk_j。若

$$\Pr[\mathrm{Dec}(\{\mathrm{sk}_{\mathrm{id}}\}_{\mathrm{id}\in\bigcup_{i=1}^{s} T_i}, c_{\mathrm{Eval}}) \neq \mathcal{C}(m_1,\cdots,m_s)] = \mathrm{negl}(\lambda)$$

$$\Pr[\mathrm{Dec}(\mathrm{sk}_j, c_{\mathrm{ReEnc}}) \neq \mathcal{C}(m_1,\cdots,m_s)] = \mathrm{negl}(\lambda) \quad (1)$$

成立，则称该 MKH-PRE 方案是正确的。

紧致性。若存在一个多项式 $\mathrm{poly}(\cdot,\cdot)$ 使一个对应 N 个用户的密文 c_{Eval} 及其重加密后得到的密文 c_{ReEnc} 的尺寸满足 $|c_{\mathrm{Eval}}|$ 和 $|c_{\mathrm{ReEnc}}|$ 都小于 $\mathrm{poly}(\lambda, N)$，则称该 MKH-PRE 方案是满足紧致性的。

3.2 安全性定义

本文采用的是文献[21]中关于 MKH-PRE 方案的安全性定义。该定义中设计了一个敌手 $\mathcal{A}$ 与挑战者之间的 IND-CPA 安全游戏，使用一个有向非循环图来记录重加密过程，图中的顶点表示诚实用户，边表示可能的重加密方向。在此游戏中，敌手能够根据重加密图的情况来发起重加密密钥生成的询问。敌手和挑战者之间的 IND-CPA 安全游戏的形式化定义如下。

阶段 1

准备 挑战者将 $\mathrm{pp} \leftarrow \mathrm{Setup}(1^\lambda)$ 发送给 $\mathcal{A}$。令 $E = \varnothing$ 为重加密图的边的集合。

诚实密钥生成 $\mathcal{A}$ 将诚实密钥的数量 N_U 发送给挑战者，挑战者生成 $(\mathrm{pk}_i, \mathrm{sk}_i),\ i = 1,\cdots,N_U$ 并将 pk_i 发送给 $\mathcal{A}$。令 Γ_H 为诚实公钥集合。

非诚实密钥生成 $\mathcal{A}$ 将非诚实密钥的数量 N_C 发送给挑战者，挑战者生成 $(\mathrm{pk}_i, \mathrm{sk}_i), i = 1,\cdots,N_C$ 并将其发送给 $\mathcal{A}$。令 Γ_C 为非诚实公钥集合。

阶段 2

$\mathcal{A}$ 可以以任意顺序发起多项式个询问。

重加密密钥生成 $\mathcal{A}$ 将 (i,j) 发送给挑战者。若 $i,j \in \Gamma_H$ 且 $G = (\Gamma_H, E \cup (i,j))$ 是有向非循环图，挑战者将 (i,j) 添加到 E 中并将 $\mathrm{rk}_{i\to j} \leftarrow \mathrm{RKGen}(\mathrm{sk}_i, \mathrm{pk}_j)$ 发给 $\mathcal{A}$；否则返回 $\perp$。

重加密 $\mathcal{A}$ 将 $(i_1,\cdots,i_s,j,c)$ 发送给挑战者。若 $j \in \Gamma_C$ 且 c=c^*，挑战者返回 $\perp$；否则挑战者将 $c_j \leftarrow \mathrm{ReEnc}(\mathrm{rk}_{i_1\to j},\cdots,\mathrm{rk}_{i_s\to j}, c)$ 发送给 $\mathcal{A}$，其中 $\mathrm{rk}_{i_k\to j} \leftarrow \mathrm{RKGen}(\mathrm{sk}_{i_k}, \mathrm{pk}_j),\ k = 1,\cdots,s$。

挑战 令 $m_0, m_1 \in \mathcal{M},\ i^* \in \Gamma_H$。$\mathcal{A}$ 将 (i^*, m_0, m_1) 发送给挑战者，挑战者随机选取一个比特 $b \in \{0,1\}$ 并将 $c^* \leftarrow \mathrm{Enc}(\mathrm{pk}_{i^*}, m_b)$ 返回给 $\mathcal{A}$。$\mathcal{A}$ 只能发起一次挑战询问。

阶段 3

$\mathcal{A}$ 输出一个比特 $b' \in \{0,1\}$。

在此游戏中，敌手 $\mathcal{A}$ 的优势定义为

$$\mathrm{Adv}^{\text{ind-cpa}}_{\text{MKH-PRE},\mathcal{A}}(\lambda) = \left|\Pr[b' = b] - \frac{1}{2}\right| \quad (2)$$

若对于任意概率多项式时间的敌手 $\mathcal{A}$ 有

$$\mathrm{Adv}^{\text{ind-cpa}}_{\text{MKH-PRE},\mathcal{A}}(\lambda) = \mathrm{negl}(\lambda) \quad (3)$$

成立，则称该 MKH-PRE 方案是 IND-CPA 安全的。

4 方案构造

本节介绍基于 NTRU 的多密钥同态代理重加密方案的构造。首先介绍 Setup 和 KeyGen 这 2 个算法，这是整个方案的基础。

$\mathrm{Setup}(1^\lambda)$。输入安全参数 λ，生成 RLWE 的维数 n、密文模数 q、明文模数 t 和 R 上的分布 ψ, χ；随机抽取向量 $\boldsymbol{a} \leftarrow U(R_q^d)$ 和矩阵 $\boldsymbol{A} \leftarrow U(R_q^{2\times m})$；输出公共参数 $\mathrm{pp} = (n, q, t, \psi, \chi, \boldsymbol{a}, \boldsymbol{A})$。

$\mathrm{KeyGen}(\mathrm{pp})$。随机抽取 $f', g \leftarrow \psi$，计算 $f = tf' + 1 (\mathrm{mod}\, q)$，若 f 在 R_q 中不可逆，则重新抽取 f'。计算 $f^{-1} \in R_q$ 并设 $h = tgf^{-1} (\mathrm{mod}\, q)$。随机抽取误差向量 $\boldsymbol{e} \leftarrow \chi^d$，计算 $\boldsymbol{b} = -\boldsymbol{af} + \boldsymbol{e} (\mathrm{mod}\, q)$。令 $\boldsymbol{f} = (f, 1) \in R_q^{1\times 2}$，随机抽取误差向量 $\boldsymbol{e}' \leftarrow \chi^{n'}$，计算 $\boldsymbol{d} = -\boldsymbol{fA} + \boldsymbol{e}' (\mathrm{mod}\, q) \in R_q^{n'}$。输出公钥 $\mathrm{pk} = (h, \boldsymbol{b}, \boldsymbol{d})$ 和

私钥 $\mathrm{sk}=f$。

4.1　多密钥同态加密的构造

本节介绍多密钥同态加密部分的算法。类似于文献[30-31]，本文使用的是 Regev 型的 NTRU 算法。本节除了给出前文所介绍的 Enc、Dec 和 Eval 算法之外，还将介绍 EvkGen、CtxtExtend 和 Relin 算法，这些算法是多密钥同态加密的重要组成部分。

EvkGen(pp,pk,sk)。随机抽取 $r\leftarrow\chi$ 和 $\boldsymbol{s}_0,\boldsymbol{e}_0,\boldsymbol{e}_1\leftarrow\chi^d$，计算 $\mathbf{evk}_0=h\boldsymbol{s}_0+\boldsymbol{e}_0+f^{-1}r\boldsymbol{g}\left(\mathrm{mod}\,q\right)$ 和 $\mathbf{evk}_1=r\boldsymbol{a}+\boldsymbol{e}_1+f\boldsymbol{g}\left(\mathrm{mod}\,q\right)$，输出计算密钥 $\mathbf{Evk}=[\mathbf{evk}_0\,|\,\mathbf{evk}_1]$。

Enc(m,pk)。随机抽取 $s,e\leftarrow\chi$，设 $\Delta=\lfloor q/t\rfloor$，计算并输出密文 $c=hs+e+\Delta m\ (\mathrm{mod}\,q)$。

MKDec($\mathbf{ct},\mathrm{sk}_1,\cdots,\mathrm{sk}_k$)：设 $\boldsymbol{f}=(f_1,\cdots,f_k)$，输出明文 $m=\lfloor(t/q)\langle\mathbf{ct},\boldsymbol{f}\rangle\rceil(\mathrm{mod}\,t)$。

CtxtExtend($\mathbf{ct}_1,\mathbf{ct}_2,\cdots,\mathbf{ct}_s$)。设此时有 s 个密文参与运算，并设 $\mathbf{ct}_i=(c_1,c_2,\cdots,c_{k_i})\in R_q^{k_i}$，其对应的用户 id 集合为 $\{\mathrm{id}_1,\mathrm{id}_2,\cdots,\mathrm{id}_{k_i}\}$，$i=1,2,\cdots,s$。令 $k=\max\{k_1,k_2,\cdots,k_s\}$，输出 k 维密文 $\mathbf{ct}_i^*=(c_1^*,c_2^*,\cdots,c_k^*)\in R_q^k$，其中

$$c_i^*=\begin{cases}c_j, & i=\mathrm{id}_j,\ 1\leqslant j\leqslant k_i\\ 0, & \text{其他}\end{cases}\tag{4}$$

Eval($\mathbf{ct}_1,\mathbf{ct}_2,\mathbf{Evk}$)。首先调用 CtxtExtend($\mathbf{ct}_1,\mathbf{ct}_2$)，得到 $\mathbf{ct}_1^*,\mathbf{ct}_2^*\in R_q^k$；然后进行同态加法或乘法运算，过程如下。

① HomAdd($\mathbf{ct}_1^*,\mathbf{ct}_2^*$)。计算并输出密文 $\mathbf{ct}=\mathbf{ct}_1^*+\mathbf{ct}_2^*\,(\mathrm{mod}\,q)$。

② HomMult($\mathbf{ct}_1^*,\mathbf{ct}_2^*$,evk)。计算 $\mathbf{ct}'=\lfloor\frac{t}{q}\mathbf{ct}_1^*\otimes\mathbf{ct}_2^*\rceil(\mathrm{mod}\,q)\in R_q^{k^2}$，输出密文 $\mathbf{ct}=\mathrm{Relin}(\mathbf{ct}',\mathrm{evk})\in R_q^k$（下文介绍 Relin 算法的细节）。

重线性化。在同态乘法运算的过程中，需要对密文进行重线性化。算法过程如下。

算法 1　重线性化算法 Relin

1) 输入密文 $\mathbf{ct}'=(c'_{i,j})_{1\leqslant i,j\leqslant k}\in R_q^{k^2}$，计算密钥和公钥 $\{\mathbf{Evk}_i,\boldsymbol{b}_i\}_{1\leqslant i\leqslant k}$。

2) for i=1:1:k

3)　　for j=1:1:k

4)　　$c''_{i,j}\leftarrow\langle\boldsymbol{g}^{-1}(c'_{i,j}),\boldsymbol{b}_j\rangle$

5)　　$c_i\leftarrow c_i+\langle\boldsymbol{g}^{-1}(c''_{i,j}),\mathbf{evk}_{i,0}\rangle\left(\mathrm{mod}\,q\right)$

6)　　$c_j\leftarrow c_j+\langle\boldsymbol{g}^{-1}(c'_{i,j}),\mathbf{evk}_{i,1}\rangle\left(\mathrm{mod}\,q\right)$

7)　　end for

8) end for

9) 输出密文 $\mathbf{ct}=(c_i)_{1\leqslant i\leqslant k}\in R_q^k$

下面，说明重线性化算法的正确性。已知用户 i 生成的计算密钥 $\mathbf{Evk}_i$ 为

$$\begin{aligned}\mathbf{evk}_{i,0}&=h_i\boldsymbol{s}_i+\boldsymbol{e}_{i,0}+f_i^{-1}r_i\boldsymbol{g}\left(\mathrm{mod}\,q\right)\\ \mathbf{evk}_{i,1}&=r_i\boldsymbol{a}+\boldsymbol{e}_{i,1}+f_i\boldsymbol{g}\left(\mathrm{mod}\,q\right)\end{aligned}\tag{5}$$

根据式(5)可得

$$\begin{aligned}&f_i\langle\boldsymbol{g}^{-1}(c''_{i,j}),\mathbf{evk}_{i,0}\rangle\approx r_ic''_{i,j}\ (\mathrm{mod}\,q)\\ &f_j\langle\boldsymbol{g}^{-1}(c'_{i,j}),\mathbf{evk}_{i,1}\rangle\approx-r_ic''_{i,j}+f_if_jc'_{i,j}\ (\mathrm{mod}\,q)\end{aligned}\tag{6}$$

所以有

$$\begin{aligned}&\left\langle\mathbf{ct},\boldsymbol{f}\right\rangle=\sum_{i=1}^{k}c_if_i=\sum_{i,j=1}^{k}f_i\left\langle\boldsymbol{g}^{-1}(c''_{i,j}),\mathbf{evk}_{i,0}\right\rangle+\\ &f_j\left\langle\boldsymbol{g}^{-1}(c'_{i,j}),\mathbf{evk}_{i,1}\right\rangle(\mathrm{mod}\,q)\approx\\ &\sum_{i,j=1}^{k}f_if_jc'_{i,j}=\left\langle\mathbf{ct}',\boldsymbol{f}\otimes\boldsymbol{f}\right\rangle(\mathrm{mod}\,q)\end{aligned}\tag{7}$$

因此，重线性化过程是正确的。注意到，重线性化过程中需要用公钥加密其对应的私钥，因此需假设本文方案具有循环安全性。

4.2　分布式解密

在 4.1 节介绍的多密钥同态加密方案中，解密多密钥密文需要所有参与运算的用户的私钥作为解密算法的输入。然而在实际应用中，某一方（运算参与方或第三方接收者）拥有所有私钥是不合理的。因此，可以设计一个协议进行分布式解密。本文方案利用一个简单且直接的方式——噪声泛洪[32-33]，来实现分布式解密。

分布式解密由 2 个算法组成：PartDec 和 FinDec。在 PartDec 算法中，用户 id 为 i 的运算参与方接收到多密钥密文 $\mathbf{ct}$ 的第 i 个分量 c_i 后，用自己的私钥 f_i 将其解密并加上随机抽取的噪声 $e_i^{\mathrm{flood}}\leftarrow\phi$（噪声分布 ϕ 的标准差远大于噪声分布 χ 的标准差）。在 FinDec 算法中，所有用户将自己的部分解密结果广播给其他用户，每个用户接收到所有其他用户的部分解密结果后，将其合并起来以恢复明文。

PartDec(c_i,f_i)。输入分量 c_i 和密钥 f_i，随机抽

取 e_i^{flood}，计算并输出 $\rho_i = f_i c_i + e_i^{\text{flood}}$。

FinDec($\{\rho_i\}_{1\leqslant i\leqslant k}$)。计算 $\rho = \sum \rho_i \pmod q$，输出 $m = \lfloor \frac{t}{q}\rho \rceil \pmod t$。

4.3 代理重加密

本节介绍重加密密钥生成算法 RKGen、重加密算法 ReEnc 及其对应的解密算法 PRDec。

RKGen(sk_i, pk_j)。从 pk_j 中提取出 $\boldsymbol{d}_j$，令 $\boldsymbol{D}_j = \boldsymbol{A} + \boldsymbol{e}_2^{\mathrm{T}} \otimes \boldsymbol{d}_j$，其中 $\boldsymbol{e}_2 = (0 \| 1) \in \mathbb{Z}^2$。抽取 $\boldsymbol{X}_i \leftarrow \{0,1\}^{n' \times d}$, $n' \geqslant 2\log q + 2\lambda$，计算并输出 $\mathbf{RK}_{i\to j} = \boldsymbol{D}_j \boldsymbol{X}_i + \begin{pmatrix} \mathbf{0} \\ f_i \boldsymbol{g} \end{pmatrix} (\bmod q) \in R_q^{2\times d}$。

ReEnc($\mathbf{rk}_{1\to j}, \cdots, \mathbf{rk}_{k\to j}, \mathbf{ct}$)。设 $\mathbf{ct} = (c_1, \cdots, c_k)$，计算并输出 $\boldsymbol{c}' = \sum_{i=1}^{k} \mathbf{RK}_{i\to j} \boldsymbol{g}^{-1}(c_i) (\bmod q) \in R_q^2$。

PRDec($\text{sk}_j, \boldsymbol{c}'$)。令 $\boldsymbol{f}_j = (f_j, 1)$，计算并输出 $m = \lfloor (t/q) < \boldsymbol{f}_j, \boldsymbol{c}' > \rceil \pmod t$。

下面，说明重加密过程的正确性。设经过多密钥同态运算后的密文为 $\mathbf{ct} = (c_1, \cdots, c_k)$，对应的私钥为 $\boldsymbol{f} = (f_1, \cdots, f_k)$。为了使用户 j 能够解密密文 $\mathbf{ct}$，需要重加密密钥 $\mathbf{RK}_{i\to j} \leftarrow \text{RKGen}(f_i, \boldsymbol{d}_j),\ i = 1, \cdots, k$。令 $\boldsymbol{f}_j = (f_j, 1)$，每个 $\boldsymbol{D}_j$ 满足

$$\begin{aligned} &\boldsymbol{f}_j \boldsymbol{D}_j = \boldsymbol{f}_j \boldsymbol{A} + \boldsymbol{f}_j (\boldsymbol{e}_2^{\mathrm{T}} \otimes \boldsymbol{d}_j) = \\ &\boldsymbol{f}_j \boldsymbol{A} + \boldsymbol{d}_j = \boldsymbol{f}_j \boldsymbol{A} + (-\boldsymbol{f}_j \boldsymbol{A} + \boldsymbol{e}') \approx \mathbf{0} \end{aligned} \tag{8}$$

于是，每个重加密密钥 $\mathbf{RK}_{i\to j}$ 满足

$$\boldsymbol{f}_j \mathbf{RK}_{i\to j} = \boldsymbol{f}_j \boldsymbol{D}_j \boldsymbol{X}_i + f_i \boldsymbol{g} \approx f_i \boldsymbol{g} (\bmod q) \tag{9}$$

因此，调用 PRDec($\text{sk}_j, \boldsymbol{c}'$) 的计算结果满足

$$\begin{aligned} \left\langle \boldsymbol{f}_j, \boldsymbol{c}' \right\rangle &= \boldsymbol{f}_j \sum_{i=1}^{k} \mathbf{RK}_{i\to j} \boldsymbol{g}^{-1}(c_i) = \sum_{i=1}^{k} \boldsymbol{f}_j \mathbf{RK}_{i\to j} \boldsymbol{g}^{-1}(c_i) \approx \\ &\sum_{i=1}^{k} f_i \boldsymbol{g} \boldsymbol{g}^{-1}(c_i) = < \boldsymbol{f}, \mathbf{ct} > (\bmod q) \end{aligned} \tag{10}$$

因此，重加密过程是正确的。

5 方案分析

5.1 噪声分析

本节对同态运算过程和代理重加密过程中的噪声变化问题进行分析。首先，对密文的噪声进行定义。令 $\mathbf{ct} \in R_q^k$ 为一个多密钥密文，其对应的明文为 $m \in R_t$，对应的私钥为 $\boldsymbol{f} \in R_q^k$。若存在 $e \in R$ 使 $\langle \mathbf{ct}, \boldsymbol{f} \rangle = \Delta m + e \pmod q$，则称 e 是密文 $\mathbf{ct}$ 的噪声。为了保证密文能够被正确解密，该密文的噪声需要满足 $\|e\|_\infty < \frac{q}{2t}$。设方案中使用的分布 ψ 是 B_{key} 界分布，χ 是 B_χ 界分布，函数 $\boldsymbol{g}^{-1}(\cdot)$ 输出向量的无穷范数的上界为 B_g。

显而易见，同态加法使噪声线性增长，因此重点关注同态乘法和代理重加密对噪声的影响。

同态乘法。设 $\mathbf{ct}_1, \mathbf{ct}_2$ 分别是 $m_1, m_2 \in R_t$ 的密文，于是有 $\langle \mathbf{ct}_i, \boldsymbol{f} \rangle = \Delta m_i + e_i + q I_i$，其中 $e_i, I_i \in R$，且 $\|e_i\|_\infty \leqslant E < \frac{q}{2t}$。令 $\mathbf{ct}' = \lfloor (t/q) \mathbf{ct}_1 \otimes \mathbf{ct}_2 \rceil (\bmod q)$，$\mathbf{ct} \leftarrow \text{Relin}(\mathbf{ct}', \text{evk})$。下面，评估结果密文 $\mathbf{ct}$ 中的噪声 e_{mult} 的大小。

由 $\|e_i\|_\infty < \frac{q}{2t}$，$\|\mathbf{ct}_i\|_\infty \leqslant \frac{q}{2}$，$\|\boldsymbol{f}\|_\infty \leqslant t B_{\text{key}}$ 可知

$$\begin{aligned} &\|I_i\|_\infty \leqslant \frac{1}{q} \|\langle \boldsymbol{c}_i, \boldsymbol{f} \rangle - \Delta m_i - e_i\|_\infty < \\ &\frac{1}{q}\left(k\delta t B_{\text{key}} \frac{q}{2} + \Delta \frac{t}{2} + \|e_i\|_\infty \right) < tk\delta B_{\text{key}} \end{aligned} \tag{11}$$

已知

$$\begin{aligned} &\langle \boldsymbol{c}_1 \otimes \boldsymbol{c}_2, \boldsymbol{f} \otimes \boldsymbol{f} \rangle = \langle \boldsymbol{c}_1, \boldsymbol{f} \rangle \langle \boldsymbol{c}_2, \boldsymbol{f} \rangle = \\ &\Delta^2 m_1 m_2 + \Delta(m_1 e_2 + m_2 e_1) + \\ &q(I_1 e_2 + I_2 e_1) + e_1 e_2 \pmod q \end{aligned} \tag{12}$$

从而 $\mathbf{ct}'$ 满足

$$\begin{aligned} &\langle \mathbf{ct}', \boldsymbol{f} \otimes \boldsymbol{f} \rangle = \Delta m_1 m_2 + (t(I_1 e_2 + I_2 e_1) + \\ &(m_1 e_2 + m_2 e_1) + \Delta^{-1} e_1 e_2) + e_{\text{rd}} \pmod q \end{aligned} \tag{13}$$

其中，$e_{\text{rd}} = \left\langle \left(\frac{t}{q} \right) \boldsymbol{c}_1 \otimes \boldsymbol{c}_2 - \mathbf{ct}', \boldsymbol{f} \otimes \boldsymbol{f} \right\rangle$。在式(13)中，内积左侧向量中的多项式的无穷范数是小于 1/2 的，内积右侧向量中的多项式的无穷范数是小于 $\delta(tB_{\text{key}})^2$ 的，于是可以得到

$$\|e_{\text{rd}}\|_\infty \leqslant k^2 \delta \frac{1}{2} \delta (t B_{\text{key}})^2 = \frac{1}{2} (tk\delta B_{\text{key}})^2 \tag{14}$$

因此，$\mathbf{ct}'$ 的噪声上界为

$$\begin{aligned} \|e_{\text{ten}}\| &= \|t(I_1 e_2 + I_2 e_1) + (m_1 e_2 + m_2 e_1) + \Delta^{-1} e_1 e_2 + e_{\text{rd}}\|_\infty \leqslant \\ &\left(2tk\delta^2 B_{\text{key}} + t\delta + \frac{1}{2}\delta \right) E + \frac{1}{2} (tk\delta B_{\text{key}})^2 \end{aligned} \tag{15}$$

下面，评估 $\mathbf{ct} \leftarrow \mathrm{Relin}(\mathbf{ct}', \mathrm{evk})$ 过程中密文噪声的变化情况。根据算法 1 可知

$$f_i\langle \boldsymbol{g}^{-1}(c''_{i,j}), \mathbf{evk}_{i,0}\rangle + f_j\langle \boldsymbol{g}^{-1}(c'_{i,j}), \mathbf{evk}_{i,1}\rangle = (r_i c''_{i,j} + e''_{i,j}) + (-r_i c''_{i,j} + f_i f_j c'_{i,j} + e'_{i,j}) = f_i f_j c'_{i,j} + e'_{i,j} + e''_{i,j} \pmod q \tag{16}$$

其中，$e'_{i,j} = \left\langle \boldsymbol{g}^{-1}(c'_{i,j}), r_i \boldsymbol{e}_j + f_j \boldsymbol{e}_{i,1} \right\rangle$，$e''_{i,j} = \left\langle g^{-1}(c''_{i,j}), tg_i \boldsymbol{s}_i + f_i \boldsymbol{e}_{i,0} \right\rangle$，根据等式 $\langle \mathbf{ct}, \boldsymbol{f}\rangle = \langle \mathbf{ct}', \boldsymbol{f} \otimes \boldsymbol{f}\rangle + \sum_{i,j=1}^{k} e_{i,j} \pmod q$ 可知，重线性化过程中增长的噪声的上界为

$$e_{\mathrm{relin}} = \sum_{i,j=1}^{k} e_{i,j} \leqslant 3tdk^2\delta^2 B_g B_\chi B_{\mathrm{key}} + dk^2\delta^2 B_\chi^2 B_g \tag{17}$$

综上所述，进行一次同态乘法后，密文噪声的上界为

$$\|e_{\mathrm{mult}}\|_\infty \leqslant \|e_{\mathrm{ten}}\|_\infty + \|e_{\mathrm{relin}}\|_\infty < 4tkn^2 B_{\mathrm{key}} E + 8t^2 dk^2\delta^2 B_\chi^2 B_g B_{\mathrm{key}}^2 \tag{18}$$

从此结论中可以看出，同态乘法使密文噪声线性增长。然而由于 B_{key}^2 项的存在，B_{key} 不可能大于 $\sqrt{q}\mathrm{poly}(n)$（为了保证解密正确），因此需要将 B_{key} 设为较小的值，例如 $\mathrm{poly}(n)$。

代理重加密。设 $\mathbf{ct} = (c_1, \cdots, c_k)$ 为多密钥同态运算的结果密文，其对应的私钥为 $\boldsymbol{f} = (f_1, \cdots, f_k)$，噪声为 e。设 $\boldsymbol{c}'$ 为对 $\mathbf{ct}$ 进行重加密后得到的密文，其对应的密钥为 $\boldsymbol{f}_j = (f_j, 1)$，噪声为 e_{re}。

根据 $\mathrm{PRDec}(\mathrm{sk}_j, \boldsymbol{c}')$ 的计算过程可知

$$\left\langle \boldsymbol{f}_j, \boldsymbol{c}' \right\rangle = \boldsymbol{f}_j \sum_{i=1}^{k} \mathbf{RK}_{i\to j} \boldsymbol{g}^{-1}(c_i) = \sum_{i=1}^{k} \boldsymbol{f}_j \mathbf{RK}_{i\to j} \boldsymbol{g}^{-1}(c_i) = \sum_{i=1}^{k} \boldsymbol{e}_j \boldsymbol{X}_i \boldsymbol{g}^{-1}(c_i) + < \boldsymbol{f}, \mathbf{ct} > \pmod q \tag{19}$$

重加密过程新增的噪声为 $\sum_{i=1}^{k} \boldsymbol{e}_j \boldsymbol{X}_i \boldsymbol{g}^{-1}(c_i)$，而 $\|\boldsymbol{e}_j\|_\infty \leqslant B_\chi$，$\|\boldsymbol{g}^{-1}(c_i)\|_\infty \leqslant B_g$，由此可得 $\left\| \sum_{i=1}^{k} \boldsymbol{e}_j \boldsymbol{X}_i \boldsymbol{g}^{-1}(c_i) \right\|_\infty \leqslant \delta^2 dn' B_\chi B_g$。因此，$\|e_{\mathrm{re}}\|_\infty \leqslant \|e\|_\infty + \delta^2 dn' B_\chi B_g$。由此可见，重加密过程引入的额外噪声较小。

5.2　安全性分析

首先，说明 MKHE 方案的安全性。考虑由公钥和密文向量中的一个元素组成的二元组 $(h_i, c_i = h_i s + e + \Delta m)$。若 h_i 是从 R_q 上的均匀分布中选取的，并且 $s \leftarrow \varsigma$, $e \leftarrow \chi$，那么此时 (h_i, c_i) 可以看成是从 RLWE 分布中选取的。因此需要保证选取的 g 和 f 能够使 $h = pgf^{-1}$ 的分布与 $U(R_q)$ 是统计意义上接近的。此前相关研究[11-12,27-28]已经证明，当选取 g 和 f 的分布为离散高斯分布且标准差大于 $\sqrt{q}\mathrm{poly}(n)$ 时，$\mathrm{DSPR}_{q,\phi}$ 问题是困难的。然而为了保证解密正确性，需要将标准差设为较小的值（即需要 DSPR 假设）。尽管 DSPR 假设未被证明是困难的，但是当 $q = 2^{n^\varepsilon}, \varepsilon \in (0, 0.5)$ 时，仍然没有高效的方法去攻破 DSPR 假设[34]。因此，若 RLWE 假设、DSPR 假设和循环安全假设是困难的，则本文的 MKHE 方案是 IND-CPA 安全的。

然后，考虑 PRE 过程的安全性。令 $\mathcal{A}$ 为任意一个概率多项式时间的敌手，其能够访问重加密密钥生成预言机 RKGen 和重加密预言机 ReEnc，但只能够根据重加密图来发起重加密密钥生成的询问。考虑如下的一系列安全游戏。

Game 0。此游戏就是 3.2 节中定义的 IND-CPA 安全游戏。假设 $\Gamma_H = \{1, \cdots, N\}$，$\Gamma_C = \{N+1, \cdots, M\}$。令 $1, \cdots, N$ 是由重加密图确定的拓扑顺序，即若 $i < j$ 则不存在从 i 到 j 的边。也就是说，$\mathcal{A}$ 只能发起生成满足 $i > j$ 的重加密密钥 $\mathrm{rk}_{i\to j}$ 的询问。

将 **Game** $\boldsymbol{k}, k = 1, \cdots, N$ 分为 2 类，**Game** $\boldsymbol{k}$**.1** 和 **Game** $\boldsymbol{k}$**.2**，并且令 **Game 0.2** 即为 **Game 0**。

Game $\boldsymbol{k}$**.1**。除了当 $\mathcal{A}$ 发起生成诚实密钥的询问时，对于所有的 $i < k$，挑战者通过随机抽取 $h_i \leftarrow U(R_q)$ 和 $\boldsymbol{d}_i \leftarrow U(R_q^{n'})$ 来生成公钥；对于所有的 $k < i \leqslant N$，挑战者计算公钥 $(h_i, \boldsymbol{d}_i) \leftarrow \mathrm{KeyGen}(\mathrm{pp})$，其余操作与 **Game** $\boldsymbol{k}$**-1.2** 相同。

Game $\boldsymbol{k}$**.2**。除了当 $\mathcal{A}$ 发起生成重加密密钥的询问时，对于所有的 $j < i \leqslant k$，挑战者通过从 $R_q^{2\times n'}$ 中随机抽取矩阵来生成重加密密钥 $\mathrm{rk}_{i\to j}$；对于所有的 $k < i, j \leqslant N$，挑战者计算 $\mathrm{rk}_{i\to j} \leftarrow \mathrm{RKGen}(\mathrm{sk}_i, \mathrm{pk}_j)$，其余操作与 **Game** $\boldsymbol{k}$**.2** 相同。

Game final。除了当 $\mathcal{A}$ 发起挑战询问时，挑战者通过随机抽取来生成密文 c，其余操作与 **Game** $\boldsymbol{N}$**.2** 相同。

敌手 $\mathcal{A}$ 在 **Game** $\boldsymbol{i}$ 中的优势记作 $\mathrm{Adv}_{\mathcal{A}}^{\mathrm{Game}\ i}(\lambda)$。下面，评估 $\mathcal{A}$ 在每个游戏中的优势。

由于 **Game 0** 是原始的 MKH-PRE 方案的 IND-CPA 游戏，因此有

$$\mathrm{Adv}^{\text{ind-cpa}}_{\text{MH-PRE},\mathcal{A}}(\lambda)=\mathrm{Adv}^{\text{Game 0}}_{\mathcal{A}}(\lambda) \tag{20}$$

在 **Game *k*.1** 中，挑战者生成的重加密密钥 $\mathrm{rk}_{k\to i}$，$i<k$ 满足 $\mathrm{rk}_{k\to i}=\mathbf{RK}_{k\to i}=\boldsymbol{D}_i\boldsymbol{X}_k+\begin{pmatrix}\mathbf{0}\\ f_k\boldsymbol{g}\end{pmatrix}(\mathrm{mod}\,q)$，其中 $\boldsymbol{X}_k\leftarrow U(\{0,1\}^{n'\times d})$，$n'\geqslant 2n\log q+2\lambda$。由于对于所有的 $i<k$，$\boldsymbol{d}_i$ 和 $\boldsymbol{A}$ 都是从均匀分布中随机抽取的，因此 $\boldsymbol{D}_i=\boldsymbol{A}+\boldsymbol{e}_2^{\mathrm{T}}\otimes\boldsymbol{d}_i$ 同样是服从均匀分布的。另外，因为 $\boldsymbol{D}_i$ 和 $\boldsymbol{X}_k$ 是服从均匀分布的，所以根据剩余哈希引理，$\boldsymbol{D}_i\boldsymbol{X}_k$ 与一个从均匀分布中随机抽取的矩阵是统计不可区分的。综上，$\mathrm{rk}_{k\to i}$ 与从均匀分布中随机抽取的矩阵在统计上是不可区分的，从而 **Game *k*.1** 与 **Game *k*.2** 是统计不可区分的。因此有

$$\left|\mathrm{Adv}^{\text{Game }k.1}_{\mathcal{A}}(\lambda)-\mathrm{Adv}^{\text{Game }k.1}_{\mathcal{A}}(\lambda)\right|=\mathrm{negl}(\lambda) \tag{21}$$

利用敌手 $\mathcal{A}$ 构造一个 PPT 算法 $\mathcal{B}$，用来区分 RLWE 分布与均匀分布。向 $\mathcal{B}$ 输入的样本 $\boldsymbol{x}\in R^2$ 要么来自 RLWE 分布（其秘密多项式为 f），要么来自均匀分布。

阶段 1

准备 $\mathcal{B}$ 计算 $\overline{\boldsymbol{A}}=\left(\boldsymbol{x}_1^{\mathrm{T}}\|\cdots\|\boldsymbol{x}_{n'}^{\mathrm{T}}\right)$，随机选取 $h_k\leftarrow U(R_q)$ 和 $\boldsymbol{d}_k\leftarrow U(R_q^{n'})$。$\mathcal{B}$ 将 $\boldsymbol{A}=\overline{\boldsymbol{A}}+\begin{pmatrix}\mathbf{0}\\ \boldsymbol{d}_k\end{pmatrix}$ 发送给 $\mathcal{A}$。

诚实密钥生成 当 $\mathcal{A}$ 发起一个生成诚实密钥的询问时，$\mathcal{B}$ 的应答如下。

① $i<k$ 时，$\mathcal{B}$ 随机选取 $h_i\leftarrow U(R_q)$ 和 $\boldsymbol{d}_i\leftarrow U(R_q^{n'})$ 并设 $\mathrm{pk}_i=(h_i,\boldsymbol{d}_i)$。

② $i=k$ 时，$\mathcal{B}$ 设 $\mathrm{pk}_i=(h_k,\boldsymbol{d}_k)$。

③ $i>k$ 时，$\mathcal{B}$ 计算 $(\mathrm{pk}_i,\mathrm{sk}_i)\leftarrow\mathrm{KeyGen}(\mathrm{pp})$。

最后，$\mathcal{B}$ 将 pk_i，$i\in\{1,\cdots,k\}$ 发送给 $\mathcal{A}$。

非诚实密钥生成 当 $\mathcal{A}$ 发起一个生成非诚实密钥的询问时，$\mathcal{B}$ 计算 $(\mathrm{pk}_i,\mathrm{sk}_i)\leftarrow\mathrm{KeyGen}(\boldsymbol{A})$ 并将 $(\mathrm{pk}_i,\mathrm{sk}_i)$ 发送给 $\mathcal{A}$。

阶段 2

重加密密钥生成 当 $\mathcal{A}$ 发起一个生成重加密密钥的询问 (i,j) 时，若 $i,j<k$，则 $\mathcal{B}$ 返回一个从均匀分布中随机抽取的 $\mathbf{RK}_{i\to j}\in R_q^{2\times d}$；若 $i,j>k$，则 $\mathcal{B}$ 返回 $\mathbf{RK}_{i\to j}\leftarrow\mathrm{RKGen}(\mathrm{sk}_i,\mathrm{pk}_j)$。

重加密 当 $\mathcal{A}$ 发起了重加密询问 $(i_1,\cdots,i_s,j,c)$ 时，$\mathcal{B}$ 返回 $c_j\leftarrow\mathrm{ReEnc}(\mathbf{RK}_{i_1\to j},\cdots,\mathbf{RK}_{i_s\to j},c)$ 并将 c_j 发送给 $\mathcal{A}$。

挑战 令 $m_0,m_1\in\mathcal{M}$，$i^*\in\Gamma_H$。$\mathcal{A}$ 发起一个挑战询问 (i^*,m_0,m_1)，$\mathcal{B}$ 随机选取一个比特 $b\in\{0,1\}$ 并返回 $c^*\leftarrow\mathrm{Enc}(\mathrm{pk}_{i^*},m_b)$。

阶段 3

当 $\mathcal{A}$ 停止询问并输出比特 $b'\in\{0,1\}$ 时，若 $b=b'$，则 $\mathcal{B}$ 输出 1；否则输出 0。

如果向 $\mathcal{B}$ 输入的分布是 RLWE 分布，那么可以发现 $\mathcal{B}$ 模拟了 **Game *k*-1.2**。$\overline{\boldsymbol{A}}$ 的第一行和 $\boldsymbol{d}_k$ 都是服从均匀分布的随机量，因此 $\boldsymbol{A}$ 也是服从均匀分布的。另外，$\boldsymbol{d}_k\approx-(f\|1)\boldsymbol{A}$ 的分布与实际游戏中的相同。由于在先前的游戏中 $rk_{i\to j}$ 被替换成了从均匀分布中随机抽取的矩阵，因此 $\mathcal{B}$ 完全模拟了 **Game *k*-1.2**。相反地，如果向 $\mathcal{B}$ 输入的是均匀分布，显然 $\mathcal{B}$ 模拟了 **Game *k*.1**。因此根据上述讨论、RLWE 假设和 DSPR 假设可知

$$\left|\mathrm{Adv}^{\text{Game }k\text{-}1.2}_{\mathcal{A}}(\lambda)-\mathrm{Adv}^{\text{Game }k.1}_{\mathcal{A}}(\lambda)\right|=\mathrm{negl}(\lambda) \tag{22}$$

在 **Game final** 中，密文是从均匀分布中随机抽取的；在 **Game *N*.2** 中，所有的公钥在前面的游戏中都被替换成了从均匀分布中随机抽取的向量，从而根据 RLWE 假设，加密算法输出的密文与从均匀分布中随机抽取的多项式是统计不可区分的。因此 **Game *N*.2** 与 **Game final** 是统计上不可区分的，即

$$\left|\mathrm{Adv}^{\text{Game }N.2}_{\mathcal{A}}(\lambda)-\mathrm{Adv}^{\text{Game final}}_{\mathrm{A}}(\lambda)\right|=\mathrm{negl}(\lambda) \tag{23}$$

$\mathcal{B}$ 在 **Game final** 中的优势为

$$\mathrm{Adv}^{\text{Game final}}_{\mathcal{A}}(\lambda)=\mathrm{negl}(\lambda) \tag{24}$$

因此，根据上述分析可以得出

$$\mathrm{Adv}^{\text{ind-cpa}}_{\text{MH-PRE},\mathcal{A}}(\lambda)\leqslant\sum_{k=1}^{N}\left|\mathrm{Adv}^{\text{Game }k\text{-}1.2}_{\mathcal{A}}(\lambda)-\mathrm{Adv}^{\text{Game }k.1}_{\mathcal{A}}(\lambda)\right|+\left|\mathrm{Adv}^{\text{Game }N.2}_{\mathcal{A}}(\lambda)-\mathrm{Adv}^{\text{Game final}}_{\mathcal{A}}(\lambda)\right|+\mathrm{Adv}^{\text{Game final}}_{\mathcal{A}}(\lambda)=\mathrm{negl}(\lambda) \tag{25}$$

因为敌手 $\mathcal{A}$ 的选取是任意的，所以本文方案是 IND-CPA 安全的。

5.3 性能分析

本节讨论 MKH-PRE 方案的计算开销和空间开销，并假设某次同态运算中有 k 个用户参与。

首先，考虑空间开销。一个多密钥密文由 k 个多项式组成；进行同态乘法共需 k 个计算密钥，每个计算密钥由 $2d$ 个多项式组成；进行代理重加密时同样需要 k 个重加密密钥，每个重加密密钥由 $2d$ 个多项式组成。因此，同态计算中的密文尺寸为 $O(kn\log q)$，单个用户生成的计算密钥尺寸为 $O(dn\log q)$，重加密密钥尺寸为 $O(dn\log q)$。

然后，考虑计算开销，主要考虑同态乘法和重加密过程的开销。计算过程中多项式乘法在总开销中占主导地位，因此主要统计每个算法需要进行的多项式乘法的次数。在同态乘法中，计算张量积需要 k^2 次多项式乘法。算法 1 中的一次迭代则需要 $3d$ 次多项式乘法，因此运行一次算法 1 需要 $3kd$ 次多项式乘法。重加密过程需要 $2kd$ 次多项式乘法。

5.4　方案对比

本节将给出本文方案与相关方案的对比。

1) 与 LTV12 方案的对比

表 1 中列出了本文方案与 LTV12 方案[8]的一些基本性质之间的对比，其中 n 表示分圆多项式的次数，q 表示模数，d 表示工具向量的维数。第二行中的“(用户)”是指单个用户所需生成的计算密钥的尺寸。

从表 1 中可以看出，本文方案中用户所需生成的计算密钥相对较小，因此用户端的开销也会相对较低。同时，本文方案中的 MKHE 部分支持分布式解密，从而可以构造一个 2 轮的安全多方计算协议，而 LTV12 方案并不支持；另外，本文方案还具有代理重加密功能。

2) 与 YKHK18 方案的对比

表 2 中列出了本文方案与 YKHK18 方案[19]的一些基本性质之间的对比，其中 λ 表示安全参数，L 表示方案能够同态运行的电路的最大层数，K 表示预先设定的最大用户数上限，k 表示当前参与运算的用户数量，t 表示当前参与运算的密文数量，$\tilde{O}(\cdot)$ 表示省略对数项。文献[35-36]提出的对于 NTRU 型方案的攻击手段的复杂度为 $2^{\tilde{O}(\sqrt{n}/\log q)}$，其中 n 为多项式次数，q 为模数。为了保证安全性，攻击成功的时间需要大于 2^{λ}，于是有 $n>\tilde{O}(\lambda^2\log^2 q)$。另外，为了能够进行 L 层同态运算，需满足 $\log q>\tilde{O}(L)$。

从表 2 中可以看出，本文方案的明文空间更大，能够支持批处理操作，因而更适合于实际应用。同时，虽然 YKHK18 方案不需要生成计算密钥，但是其需要生成冗余密文来辅助多密钥同态运算。因此，综合用户端通过计算生成的公钥、计算密钥、重加密密钥的尺寸以及明密文的尺寸比例来看，相较于 YKHK18 方案，本文方案用户端开销相对较低。

6　在联邦学习隐私保护中的应用

本节将给出本文方案在一类典型的多方云计算场景——联邦学习中的应用。

6.1　联邦学习及其隐私保护

联邦学习是一种机器学习框架，由多个数据所有者和云服务器组成，拥有不同训练数据集的多个训练参与者共同执行一个学习任务。每个数据所有者事先约定好模型初始参数，首先在本地训练模型，然后将模型参数上传至云服务器。云服务器在收集所有参与训练方的模型参数后对其进行处理，更新全局模型参数并发送给训练参与方。不断重复上述过程，直至模型收敛。联邦学习环境可以实现数据所有者在不需要给出己方数据的情况下，能够协同其他数据所有者共同训练模型。

联邦学习的训练过程中权重参数的传输会存在隐私安全问题，因此需要建立隐私保护机制。将

表 1　**本文方案与 LTV12 方案的对比**

方案	公钥尺寸	计算密钥尺寸（用户）	密文尺寸	分布式解密	代理重加密
本文方案	$O(dn\log q)$	$O(dn\log q)$	$O(kn\log q)$	支持	支持
LTV12 方案	$O(n\log q)$	$O(n\log^3 q)$	$O(n\log q)$	不支持	不支持

表 2　**本文方案与 YKHK18 方案的对比**

方案	密码学假设	明文空间	公钥尺寸	计算密钥尺寸（用户端）	重加密密钥尺寸	密文扩张	同态密文/明文尺寸比例（用户端）	同态密文/明文尺寸比例（计算过程中）	重加密后密文/明文尺寸比例	批处理
本文方案	RLWE、DPSR	多项式环	$\tilde{O}(\lambda^2L^4)$	$\tilde{O}(\lambda^2L^4)$	$\tilde{O}(\lambda^2L^4)$	$\tilde{O}(1)$	$\tilde{O}(\lambda L^2)$	$\tilde{O}(\lambda kL^2)$	$\tilde{O}(\lambda L^2)$	支持
YKHK18 方案	LWE	{0,1}	$\tilde{O}(\lambda(N+L)^2)$	—	$\tilde{O}(\lambda^2(N+L)^2)$	$t\tilde{O}(k^2\lambda^4(N+L)^4)$	$\tilde{O}(\lambda^3(N+L)^4)$	$\tilde{O}(k^2\lambda^3(N+L)^4)$	$\tilde{O}(\lambda(N+L))$	不支持

权重参数进行加密处理能够实现权重参数的保密，但各个训练参与方若使用同一对公私钥，则需要共享密钥。如果密钥被窃取，整个系统的安全性将遭到破坏。在这一场景下，多密钥同态加密方案能够提供更好的安全性，每个参与者可以直接使用自己的公钥进行加密操作，不需要执行共享密钥的协议，进而降低了密钥被窃取的风险。

6.2 模型设计

本文设计了 2 种联邦训练卷积神经网络的场景，分别是联邦卷积神经网络场景和代理重加密−联邦卷积神经网络场景。在这 2 种场景中，云服务器都是诚实且好奇的。联邦卷积神经网络场景如图 1 所示。每个训练参与方使用自己的公钥将自己的模型参数 w_i 加密后上传到云服务器；云服务器对参数进行处理，然后将更新后的全局参数发送给参与者；参与者解密参数后更新自己的模型并进行下一轮迭代。重复上述步骤，可以实现多方共同训练一个神经网络模型，同时训练者无法获知其他训练者的模型参数。

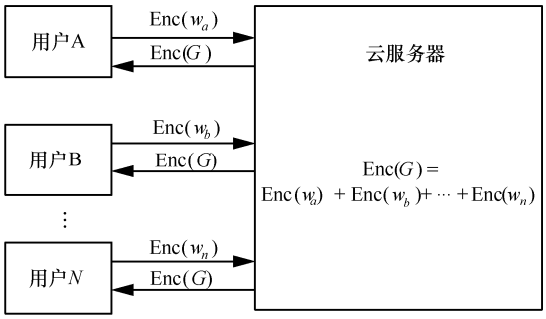


图 1 联邦卷积神经网络场景

代理重加密−联邦卷积神经网络场景如图 2 所示。与联邦卷积神经网络场景的不同之处在于，模型接收者并不存在于模型训练者之中。此时，需要重加密机制将密文转换为接收者可以解密的密文。在此场景中，需假设云服务器不能与用户 K 共谋。

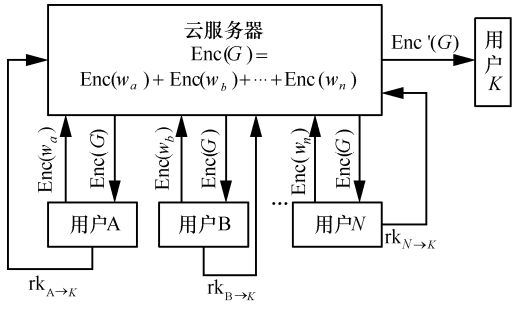


图 2 代理重加密−联邦卷积神经网络场景

6.3 实验与分析

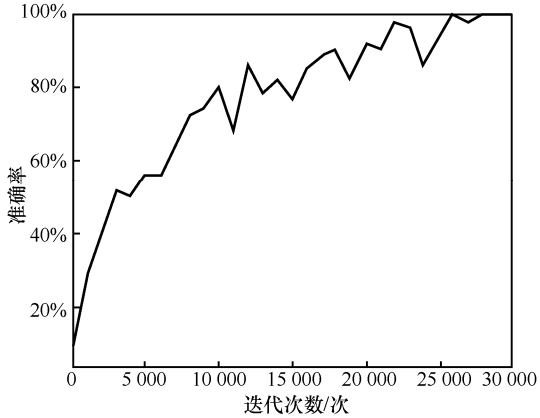
本实验使用的配置为：Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz 3.41 GHz，16 GB 内存。在虚拟机中使用 Ubuntu16.04 系统和 C++语言，借助 GMP 库和 NTL 库实现 MKH-PRE 方案；使用 Python 语言实现联邦学习过程。实验选取的具体场景为 3 个参与方在联邦学习框架下通过一个云服务器共同训练卷积神经网络。选取的卷积神经网络包含四层结构，分别是卷积层、非线性层、池化层和全连接层，非线性层选取 ReLU 函数，全连接层选取 softmax 激活函数。实验使用的数据集为 cifar-10，训练集包含 55 000 个样本，其中参与者 A 有 20 000 个样本，参与者 B 有 20 000 个样本，参与者 C 有 15 000 个样本；测试集包含 5 000 个样本。

本文采用的对照实验如下。1) 使用明文参数直接进行联邦学习，迭代 30 000 次。2) 参与者使用自己的公钥对权重参数进行加密，云服务器每一轮收集 3 个密文参数并进行聚合，将计算得到的全局参数密文发送给训练参与者，参与者解密后进行下一轮训练；重复此过程进行联邦学习，其余参数设定与 1)相同。两组实验得到的训练集准确率如图 3 所示，其中横坐标表示迭代次数（坐标间隔为 1 000），纵坐标表示准确率。图 3(a)为明文参数下的训练集准确率随迭代次数变化的曲线，图 3(b)为密文参数下训练集准确率随迭代次数变化的曲线，两组实验得到的准确率变化趋势基本一致，说明同态加密过程几乎不影响训练过程。

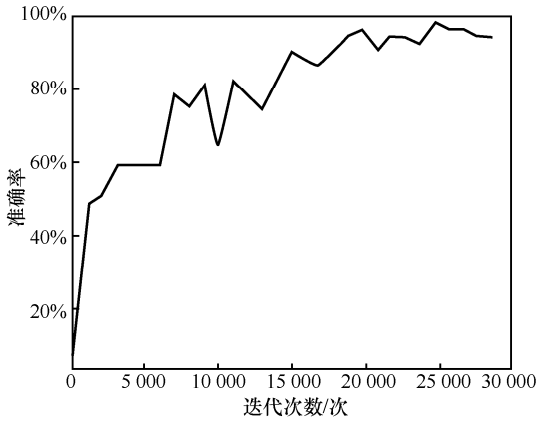
两组实验的测试集准确率如表 3 所示，表 3 中的准确率为多次实验后的平均值。从表 3 中可以看出，明密文参数模型下的测试集准确率几乎一致，说明同态加密过程基本不影响模型的准确性。

本文也进行了代理重加密过程的实验。云服务器使用训练参与方生成的重加密密钥将多密钥同态计算得到的模型参数密文进行重加密，所得结果发送给用户 K。用户 K 使用自己的私钥将模型参数解密，得到训练好的模型。实验结果表明用户 K 的测试集准确率约为 70.83%，与表 3 中准确率几乎一致，这说明重加密过程基本不影响模型的准确性。上述结果表明本文所提的 MKH-PRE 方案并没有使训练过程和训练结果（获得的模型）产生偏差。

本文也对 MKH-PRE 方案的各个算法的运行时间进行了测试。多密钥同态加密过程中各个算法的运行时间如表 4 所示，代理重加密过程中各个算法的运行时间如表 5 所示。由表 4 和表 5 的实验结果可知，本文方案具有良好的时间效率。



(a) 明文参数模型训练集准确率变化



(b) 密文参数模型训练集准确率变化

图 3　训练集准确率变化

表 3　明密文参数模型下的测试集准确率

数据集	准确率
明文参数模型测试集	71.08%
密文参数模型测试集	71.19%

表 4　多密钥同态加密过程中各个算法的时间开销

算法	运行时间/ms
Setup	1 450
KeyGen	0.549
Enc	8.972
Eval	0.158
PartDec	10.920
PartDec+FinDec	33.747

表 5　代理重加密过程中各个算法时间开销

算法	运行时间/ms
RKGen	3.015
ReEnc	276.613
PRDec	280.800

7　结束语

本文提出了一种基于 NTRU 的多密钥同态代理重加密方案。该方案同时保留了多密钥同态加密和代理重加密的特性：多密钥同态加密功能支持分布式解密，适用于数据提供者是接收者的安全多方计算场景；代理重加密功能适用于接收者为数据提供者之外的第三方的外包计算场景，使用者可以根据具体场景需求选择合适的功能。另外，本文方案支持加密多项式环中的元素，用户端的开销也相对较小，因而相较于此前的 MKH-PRE 方案更具实用性。最后，将本文方案进行了实现并应用于联邦学习场景以解决其中的隐私保护问题。实验结果表明，本文方案几乎不影响联邦学习的准确率，也不会带来较大的计算开销。

参考文献：

[1] RIVEST R, ADLEMAN L, DERTOUZOS M. On data banks and privacy homomorphisms[J]. Foundations of Secure Computation, 1978, 4(11): 169-177.

[2] GENTRY C. A fully homomorphic encryption scheme[D]. Palo Alto: Stanford University, 2009.

[3] GENTRY C. Fully homomorphic encryption using ideal lattices[C]//Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC). New York: ACM Press, 2009: 169-178.

[4] BRAKERSKI Z, VAIKUUNTANATHAN V. Efficient fully homomorphic encryption from (standard) LWE[C]//Proceedings of the 52nd IEEE Annual Symposium on Foundations of Computer Science. Piscataway: IEEE Press, 2011: 97-106.

[5] BRAKERSKI Z, GENTRY C, VAIKNTANATHAN V. (Leveled) Fully homomorphic encryption without bootstrapping[C]//Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. New York: ACM Press, 2012: 309-325.

[6] DIJK V M, GENTRY C, HALEVI S, et al. Fully homomorphic encryption over the integers[C]//29th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2010: 24-43.

[7] GENTRY C, SAHAI A, WATERS B. Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based[C]//33rd Annual Cryptology Conference. Berlin: Springer, 2013: 75-92.

[8] LÓPEZ-ALT A, TROMER E, VAIKUNTANATHAN V. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption[C]//Proceedings of the 44th Annual ACM Symposium on Theory of Computing. New York: ACM Press, 2012: 1219-1234.

[9] CHILLOTTI I, GAMA N, GEORGIEVA M, et al. Faster fully homomorphic encryption: bootstrapping in less than 0.1 seconds[C]//International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2016: 3-33.

[10] 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. Berlin: Springer, 2017: 409-437.

[11] CLEAR M, MCGOLDRICK C. Multi-identity and multi-key leveled FHE from learning with errors[C]//35th Annual International Cryptology Conference. Berlin: Springer, 2016: 630-656.
[12] MUKHERJEE P, WICHS D. Two round multiparty computation via multi-key FHE[C]//35th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2016: 735-763.
[13] PEIKERT C, SHIEHIAN S. Multi-key FHE from LWE, revisited[C]//14th Theory of Cryptography Conference. Berlin: Springer, 2016: 217-238.
[14] BRAKERSKI Z, PERLMAN R. Lattice-based fully dynamic multi-key FHE with short ciphertexts[C]//36th Annual International Cryptology Conference. Berlin: Springer, 2016: 190-213.
[15] CHEN L, ZHANG Z F, WANG X Q. Batched multi-hop multi-key FHE from ring-LWE with compact ciphertext extension[C]//15th Theory of Cryptography Conference. Berlin: Springer, 2017: 597-627.
[16] LI N, ZHOU T, YANG X, et al. Efficient multi-key FHE with short extended ciphertexts and directed decryption protocol[J]. IEEE Access, 2019, 7: 56724-56732.
[17] CHEN H, CHILLOTTI I, SONG Y. Multi-key homomorphic encryption from TFHE[C]//25th International Conference on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2019: 446-472.
[18] CHEN H, DAI W, KIM M, et al. Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference[C]//2019 Conference on Computer and Communications Security. New York: ACM Press, 2019: 395-412.
[19] YASUDA S, KOSEKI Y, HIROMASA R, et al. Multi-key homomorphic proxy re-encryption[C]//2018 International Conference on Information Security. Berlin: Springer, 2018: 328-346.
[20] HOFFSTEIN J, PIPHER J, SILVERMAN J H. NTRU: a ring-based public key cryptosystem[C]//1998 International Algorithmic Number Theory Symposium. Berlin: Springer, 1998: 267-288.
[21] STEHLÉ D, STEINFELD R. Making NTRU as secure as worst-case problems over ideal lattices[C]//30th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2011: 27-47.
[22] YU Y, XU G W, WANG X Y. Provably secure NTRU instances over prime cyclotomic rings[C]//20th IACR International Conference on Practice and Theory in Public-Key Cryptography. Berlin: Springer, 2017: 409-434.
[23] WANG Y, WANG M Q. Provably secure NTRUEncrypt over any cyclotomic field[C]//25th Selected Areas in Cryptography. Berlin: Springer, 2018: 391-417.
[24] 李瑞琪, 贾春福. 一个基于 NTRU 的多密钥同态加密方案[J]. 密码学报, 2020, 7(5): 683-697.
LI R Q, JIA C F. A multi-key homomorphic encryption scheme based on NTRU[J]. Journal of Cryptologic Research, 2020, 7(5): 683-697.
[25] 车小亮, 周潭平, 李宁波, 等. NTRU 型多密钥全同态加密方案的优化[J]. 工程科学与技术, 2020, 52(5): 186-193.
CHE X L, ZHOU T P, LI N B, et al. Optimization of NTRU-type multi-key fully homomorphic encryption scheme[J]. Advanced Engineering Sciences, 2020, 52(5):186-193.
[26] NUÑEZ D, AGUDO I, LOPEZ J. NTRUReEncrypt: an efficient proxy re-encryption scheme based on NTRU[C]//Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security. New York: ACM Press, 2015: 179-189.
[27] 张明武, 杜林. 基于 NTRU 的单向抗合谋代理重加密方案[J]. 密码学报, 2020, 7(2): 187-196.
ZHANG M W, DU L. A collusion-resistant and uni-directional proxy re-encryption scheme based on NTRU[J]. Journal of Cryptologic Research, 2020, 7(2): 187-196.
[28] LYUBASHEVSKY V, PEIKERT C, REGEV O. On ideal lattices and learning with errors over rings[C]//29th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2010: 1-23.
[29] MICCIANCIO D, PEIKERT C. Trapdoors for lattices: simpler, tighter, faster, smaller[C]//31st Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2012: 700-718.
[30] BOS J W, LAUTER K, LOFTUS J, et al. Improved security for a ring-based fully homomorphic encryption scheme[C]//2013 IMA International Conference on Cryptography and Coding. Berlin: Springer, 2013: 45-64.
[31] BRAKERSKI Z. Fully homomorphic encryption without modulus switching from classical GapSVP[C]//31st Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2012: 868-886.
[32] MUKHERJEE P, WICHS D. Two round multiparty computation via multi-key FHE[C]//35th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2016: 735-763.
[33] ASHAROV G, JAIN A, LÓPEZ-ALT A, et al. Multiparty computation with low communication, computation and interaction via threshold FHE[C]//31st Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Springer, 2012: 483-501.
[34] LÓPEZ-ALT A, TROMER E, VAIKUNTANATHAN V. Multikey fully homomorphic encryption and applications[J]. SIAM Journal on Computing, 2017, 46(6): 1827-1892.
[35] ALBRECHT M, BAI S, DUCAS L. A subfield lattice attack on overstretched NTRU assumptions[C]//36th Annual International Cryptology Conference. Berlin: Springer, 2016: 153-178.
[36] KIRCHNER P, FOUQUE P A. Revisiting lattice attacks on overstretched NTRU parameters[C]//36th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2017: 3-26.

[作者简介]

李瑞琪（1993− ），男，黑龙江尚志人，南开大学博士生，主要研究方向为同态加密、格密码学等。

贾春福（1967− ），男，河北文安人，博士，南开大学教授、博士生导师，主要研究方向为网络与信息安全、可信计算、恶意代码分析、密码技术应用等。

王雅飞（1995− ），女，天津人，南开大学硕士生，主要研究方向为同态加密应用、隐私保护等。