本方案中用 n 表示安全参数，所有其余参数均为 n 的函数：

- $q \geq \beta \cdot n^{\delta}$，其中 $\delta \in (0,1)$ 是任意常数
- $k = \lceil \log q \rceil$
- $\beta = s \cdot \sqrt{m}$
- $s \geq 3 \cdot r \cdot s_1(\mathbf{T})$，其中 $s_1(\mathbf{T})$ 表示矩阵 $\mathbf{T} = [\mathbf{E}^t \ \mathbf{R}^t \ \mathbf{I}]^t$ 的最大奇异值，$\mathbf{E}, \mathbf{R}$ 的定义见算法 1
- $r = \sqrt{\ln\big(2m(1+1/\epsilon)\big)/\pi}$，其中 $\epsilon = 2^{-128}$ 或 $\epsilon = 2^{-256}$
- $m = n(2+k)$

$\mathbb{Z}_q$ 表示 $\mathbb{Z} \mod q$ 的绝对最小剩余。$\forall\ z \in \mathbb{Z}_q$，令 $[z]_2^k$ 表示唯一字符串 $\in \{0,1\}$ 或 $\in \{-1,0\}$ 使得 $z = \sum_i z_i 2^i$。

令 $\mathbf{g} = [1,\ 2,\ 4, \cdots,\ 2^{k-1}] \in \mathbb{Z}^{1\times k}$。矩阵 Σ_0 定义为

$$\Sigma_0 = \begin{bmatrix} 1 & & & & -(\frac{1}{2})^{k-1} \\ & 1 & & & -(\frac{1}{2})^{k-2} \\ & & \ddots & & \vdots \\ & & & 1 & -\frac{1}{2} \\ -(\frac{1}{2})^{k-1} & -(\frac{1}{2})^{k-2} & \cdots & -\frac{1}{2} & \sum_{i=1}^{k-1}(\frac{1}{4})^i + \frac{q^2}{4^k} \end{bmatrix}.$$

对任意实对称矩阵 Σ，若 $\Sigma = \mathbf{B}\mathbf{B}^t$，定义 $\sqrt{\Sigma} = \mathbf{B}$。$\mathcal{D}_{\Lambda,\sqrt{\Sigma},\mathbf{c}}$ 表示格 Λ 上以 $\mathbf{c}$ 为中心、高斯偏差为 $\sqrt{\Sigma}$ 的离散高斯分布（当 $\mathbf{c} = \mathbf{0}$ 时下标 $\mathbf{c}$ 省略）。

定义环 $\mathcal{R} = \mathbb{Z}[x]/(x^n+1)$，$\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n+1)$，$\mathcal{Q} = \mathbb{Q}[x]/(x^n+1)$。为方便描述，可将环 $\mathcal{R}$（或 $\mathcal{R}_q$，$\mathcal{Q}$）等同于 $\mathbb{Z}^n$（或 $\mathbb{Z}_q^n$，$\mathbb{Q}^n$），即将环中的元素 $u(x) = \sum_{i=0}^{n-1} u_i x^i$ 等同于其系数所构成的列向量 $\mathbf{u} = (u_0, \ldots, u_{n-1})$。$\mathrm{Rot}(\mathbf{u}) = (-u_{n-1}, u_1, \ldots, u_{n-2})$。$\langle \cdot, \cdot \rangle$ 表示向量的内积。令 $\hat{\mathbf{v}}$ 表示环上的向量。定义 $\hat{\mathbf{g}} = [\mathbf{g}_1,\ \mathbf{g}_2, \ldots, \mathbf{g}_{k-1}] = [\mathbf{1},\ \mathbf{2}, \cdots,\ \mathbf{2}^{k-1}] \in \mathcal{R}^{1\times k}$。$\mathcal{H}:\ \{0,1\}^* \to \mathbb{Z}_q^n$ 表示哈希函数。

1 算法描述

- KeyGen(1^n)：输入安全参数 n；输出公钥 pk，私钥 sk。
  - 选取均匀随机元素 $\mathbf{a} \leftarrow \mathcal{R}_q$
  - 选取 $\hat{\mathbf{r}} = [\mathbf{r}_1, \ldots, \mathbf{r}_k] \leftarrow \mathcal{D}_{\mathbb{Z},2\sqrt{n}}^{n\times k}$，$\hat{\mathbf{e}} = [\mathbf{e}_1, \ldots, \mathbf{e}_k] \leftarrow \mathcal{D}_{\mathbb{Z},2\sqrt{n}}^{n\times k}$，则 $\hat{\mathbf{r}},\ \hat{\mathbf{e}} \in \mathcal{R}_q^{1\times k}$
  - 令 $\hat{\mathbf{a}} = [\mathbf{1},\ \mathbf{a},\ \mathbf{g}_1 - (\mathbf{a}\mathbf{r}_1 + \mathbf{e}_1), \ldots,\ \mathbf{g}_k - (\mathbf{a}\mathbf{r}_k + \mathbf{e}_k)] \in \mathcal{R}_q^{1\times(2+k)}$
  - 令 $pk = \hat{\mathbf{a}},\ sk = (\hat{\mathbf{e}}, \hat{\mathbf{r}})$
- Sign(sk, μ)：输入私钥 sk 和消息 μ；输出签名 σ。
  - 令 $s = 2.4 \cdot r \cdot \sqrt{n} \cdot (\sqrt{nk} + \sqrt{2n} + 4.7)$

- $\hat{\mathbf{p}} = [\mathbf{p}_1, \ldots, \mathbf{p}_{k+2}] \leftarrow \mathsf{SampleP}\big(n, q, s, r, (\hat{\mathbf{e}}, \hat{\mathbf{r}})\big)$
- 令 $\mathbf{v} = \mathcal{H}(\mu \| str) - \langle \hat{\mathbf{a}}, \hat{\mathbf{p}} \rangle$
- 对 $\mathbf{v}$ 的每个分量 v_i, $\mathbf{z}_i \leftarrow \mathsf{SampleG}(q, v_i)$
- 将矩阵 $[\mathbf{z}_1, \ldots, \mathbf{z}_n] \in \mathbb{Z}^{k \times n}$ 按行转换为 $\hat{\mathbf{x}} = [\mathbf{x}_1, \ldots, \mathbf{x}_k] \in \mathcal{R}^{1 \times k}$
- 令 $\sigma = [\mathbf{p}_1 + \langle \hat{\mathbf{e}}, \hat{\mathbf{x}} \rangle,\ \mathbf{p}_2 + \langle \hat{\mathbf{r}}, \hat{\mathbf{x}} \rangle,\ \mathbf{p}_3 + \mathbf{x}_1, \ldots,\ \mathbf{p}_{k+2} + \mathbf{x}_k]$

- $\mathrm{Verf}(pk, \mu, \sigma)$：输入公钥 pk，消息 μ 和签名 σ；输出 0 或 1。
  - 若 $\langle \hat{\mathbf{a}}, \sigma \rangle = \mathcal{H}(\mu \| str)$ 且 $\|\sigma\| \le s \cdot \sqrt{m}$，输出 1；否则输出 0

Algorithm 1 $\mathsf{SampleP}\big(n, q, s, r, (\hat{\mathbf{e}}, \hat{\mathbf{r}})\big)$

Input: $n, q \in \mathbb{Z},\ s, r \in \mathbb{R},\ (\hat{\mathbf{e}}, \hat{\mathbf{r}}) \in \mathcal{R}^{2 \times k}$

Output: $\mathbf{p} \sim \mathcal{D}_{\mathbb{Z}^m, \sqrt{\Sigma_{\mathbf{p}}}}$，其中 $\Sigma_{\mathbf{p}} = s^2 \cdot \mathbf{I} - 9 \cdot r^2 \cdot \begin{bmatrix} \mathbf{E} \\ \mathbf{R} \\ \mathbf{I} \end{bmatrix} \cdot [\mathbf{E}^t\ \mathbf{R}^t\ \mathbf{I}]$, $\mathbf{E} = [\mathrm{Rot}(\mathbf{e}_1) | \cdots | \mathrm{Rot}(\mathbf{e}_k)]$, $\mathbf{R} = [\mathrm{Rot}(\mathbf{r}_1) | \cdots | \mathrm{Rot}(\mathbf{r}_k)]$

$\alpha = 3r$
$z = (\alpha^{-2} - s^{-2})^{-1}$
$\mathbf{a} = s^2 - z\langle \hat{\mathbf{r}}, \hat{\mathbf{r}} \rangle$
$\mathbf{b} = -z\langle \hat{\mathbf{e}}, \hat{\mathbf{r}} \rangle$
$\mathbf{d} = s^2 - z\langle \hat{\mathbf{e}}, \hat{\mathbf{e}} \rangle$
for $i = 0$ to $nk - 1$ **do**
    $q_i \leftarrow \mathsf{SampleZ}(\sqrt{s^2 - \alpha^2})$
end for
将 $\mathbf{Q} \in \mathbb{Z}^{k \times n}$ 转化为 $\hat{\mathbf{q}} \in \mathcal{R}_q^{k \times 1}$
$\hat{\mathbf{c}} = \frac{-\alpha^2}{s^2 - \alpha^2} \begin{bmatrix} \hat{\mathbf{r}} \\ \hat{\mathbf{e}} \end{bmatrix} \hat{\mathbf{q}}$
$\mathbf{p}' \leftarrow \mathsf{Sample2Z}(\mathbf{a}, \mathbf{b}, \mathbf{d}, \hat{\mathbf{c}})$
将 $\mathbf{p}' \in \mathbb{Z}^{2 \times n}$ 转化为 $\hat{\mathbf{p}}' \in \mathcal{R}_q^2$
将 $(\hat{\mathbf{p}}', \hat{\mathbf{q}}) \in \mathcal{R}^2 \times \mathcal{R}^k$ 转化为 $\mathbf{p} \in \mathbb{Z}^m$
return $\mathbf{p}$

Algorithm 2 $\mathsf{Sample2Z}(\mathbf{a}, \mathbf{b}, \mathbf{d}, \hat{\mathbf{c}})$

Input: $\mathbf{a}, \mathbf{b}, \mathbf{d} \in \mathcal{Q},\ \hat{\mathbf{c}} \in \mathcal{Q}^{2 \times 1}$

Output: $[\mathbf{q}_0, \mathbf{q}_1] \in \mathcal{R}_q^{1 \times 2}$

$\hat{\mathbf{c}} = (\mathbf{c}_0, \mathbf{c}_1)$
$\mathbf{q}_1 \leftarrow \mathsf{SampleFZ}(\mathbf{d}, \mathbf{c_1})$
$\mathbf{c}_0 = \mathbf{c}_0 + \mathbf{b}\mathbf{d}^{-1}(\mathbf{q}_1 - \mathbf{c}_1)$
$\mathbf{q}_0 \leftarrow \mathsf{SampleFZ}(\mathbf{a} - \mathbf{b}\mathbf{d}^{-1}\mathbf{b}^t, \mathbf{c}_0)$
return $(\mathbf{q}_0, \mathbf{q}_1)$

Algorithm 3 SampleFZ$(\mathbf{f}, \mathbf{c})$

Input: $\mathbf{f}, \mathbf{c} \in \mathcal{Q}$

Output: $\mathbf{q} \in \mathcal{R}_q$

if $\dim(\mathbf{f})$=1 **then**
  return SampleZ$(\sqrt{f}, c)$
else
  $f(x) = f_0(x^2) + x \cdot f_1(x^2)$
  $c(x) = c_0(x^2) + x \cdot c_1(x^2)$
  $\hat{\mathbf{c}} = (\mathbf{c}_0, \mathbf{c}_1)$
  $(\mathbf{q}_0, \mathbf{q}_1) \leftarrow$ Sample2Z$(\mathbf{f}_0, \mathbf{f}_1, \mathbf{f}_0, \hat{\mathbf{c}})$
  $q(x) = q_0(x^2) + x \cdot q_1(x^2)$
end if
return $\mathbf{q}$

Algorithm 4 SampleG(q, u)

Input: $q, u \in \mathbb{Z}_q$

Output: $\mathbf{x} \sim \mathcal{D}_{\Lambda_u^{\perp}(\mathbf{g}), 3r}$

$\mathbf{p} \leftarrow \mathcal{D}_{\mathbb{R}^k, 2r \cdot \sqrt{2 \cdot \mathbf{I} - \Sigma_0}}$
for $i = 0$ to $k-1$ **do**
  $p_i \leftarrow \mathcal{D}_{\mathbb{Z}, r, p_i}$
end for
$v = u - \langle \mathbf{g}, \mathbf{p} \rangle$
$\mathbf{v} = [v]_2^k,\ \mathbf{q} = [q]_2^k$
$c = -\frac{v}{q},\ y \leftarrow \mathcal{D}_{\mathbb{Z}, r, c}$
$\mathbf{w} = \mathbf{v} + y \cdot \mathbf{q}$
for $i = 0$ to $k-2$ **do**
  $x_i \leftarrow D_{2\mathbb{Z}+w_i, 2r}$
  $w_{i+1} = w_{i+1} + \frac{w_i - x_i}{2} \in \mathbb{Z}$
end for
$x_{k-1} = w_{k-1}$
$\mathbf{x} = (x_0, \ldots, x_{k-1})$
$\mathbf{x} \leftarrow \mathbf{x} + \mathbf{p}$
return $\mathbf{x}$

2 设计原理

2.1 策略和思路

签名方案使用了格上 “hash and sign” 签名策略 [7]，即对消息的哈希值进行签名。使用该策略有如下优点：

- 使得签名算法设计更加简洁
- 在经典/量子随机谕言机模型下均满足强 EUF-CMA 安全性 [3, 7]
- 容易扩展出更丰富的功能，如基于身份的加密 [7]、消息恢复模式 [5]等

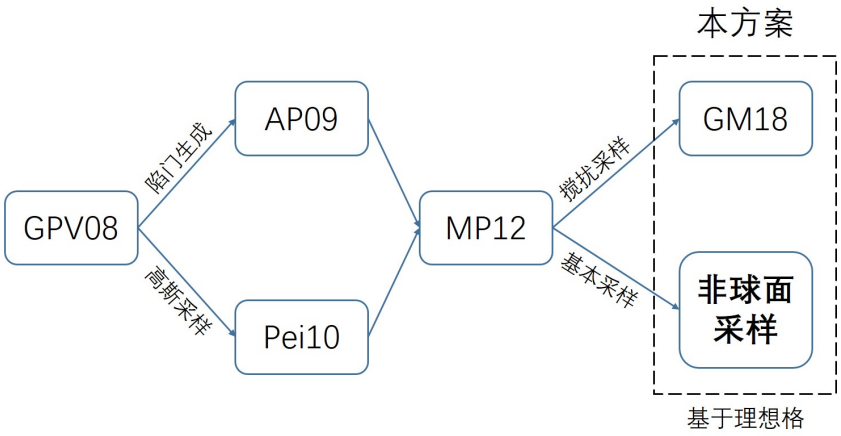

Figure 1: 技术路线图

出于安全考虑，消息 μ 被哈希前需要先串接一个 320 比特随机串 str。方案的公钥 $\hat{\mathbf{a}}$ 定义了一个抗碰撞原像可采样函数（PSF）$f_{\hat{\mathbf{a}}} : \mathcal{R}^{2+k} \to \mathcal{R}_q$。给定消息 μ 的哈希值 $\mathbf{u} = \mathcal{H}(\mu \| str)$，利用陷门 $(\hat{\mathbf{e}}, \hat{\mathbf{r}})$ 可以从其（指数多）原像 $\Lambda_{\mathbf{u}}^{\perp}(\hat{\mathbf{a}})$ 中按照离散高斯分布随机抽取一个原像作为签名 σ 而不泄露陷门的任何信息。

签名算法的核心是利用陷门进行原像高斯采样：

- 陷门：[7] 中的陷门是格的一组短基；[10] 中提出 q-ary 格上“更简单、更紧凑、更快速、更小巧”的陷门
- 高斯采样：[7] 中的采样算法针对的是任意格；[10] 基于 [12] 中的“ 卷积”技术，提出一种针对 q-ary 格的更高效的采样算法；该算法将高斯采样分为搅扰采样和基本采样两部分

本方案在 “hash and sign” 策略的基础上，使用了 [10] 中的陷门生成和离散高斯分布的“卷积”技术，即最终采样输出由搅扰向量和基本采样两部分“卷积”而成。算法 SampleP 用以搅扰向量的采样，算法 SampleG 用于基本采样。SampleP 与待签名的消息无关，可以在线下提前完成；SampleG 与待签名的消息相关，需要给定消息的哈希值后才能执行，因此需要在线上完成。本方案中的 SampleP 使用了[6] 中的采样方法；而 SampleG 使用了我们的非球面高斯采样技术。此外，为了克服格公钥密码庞大的密钥尺寸这一固有缺陷，本算法基于理想格，利用其特殊的代数结构来降低公钥和私钥的尺寸，进而减少应用中的通信带宽。综上，本方案所使用的技术发展路线如图 1 所示。

2.2 设计原则

算法的设计兼顾如下两方面的需求：

- 安全性
  - 抵抗量子攻击

– 由格上困难问题在最坏情况下的困难性保障

- 效率
  - 签名速度快
  - 存储空间小

2.3 参数选择依据

- 安全性

  签名算法本身基于 Ring-SIS 困难假设；而参数生成过程基于 Ring-LWE 困难假设。由于后者的假设更强，我们在参数选择时只考虑 Ring-LWE 问题的困难性。文献 [8, 9] 中的结果表明，Ring-LWE 问题至少和理想格上的近似最短向量问题（SVP_γ）一样困难。由于目前理想格上的环结构无法加速困难问题的求解，所以通常认为一般格和理想格上的困难问题具有相同的困难性。而求解格上 SVP_γ 最有效的算法是 BKZ 算法 [4, 13]。文献 [1, 2, 4] 中的结果表明，当前求解格上 SVP_γ 最高效算法的时间复杂度约为 $2^{0.292n}$。 因此，我们需要 $0.292n > 128$ 或 $0.292n > 256$，以期望达到 128 比特或 256 比特的安全性。

- 有效性

  本方案中的运算都定义在环 $\mathcal{R} = \mathbb{Z}[x]/f(x)$ 或 $\mathcal{R}_q = \mathbb{Z}_q[x]/f(x)$ 上。当 $f(x) = x^n + 1$ 且 n 为 2 的幂次时，环上的运算最高效。

综上，兼顾算法的安全性和有效性，本方案分别选择 $n = 512$ 和 $n = 1024$。

3 安全性分析

3.1 安全性证明

3.1 ([7] 命题 6.1)**.** 本方案基于 *Ring-SIS* 假设，在随机谕言机模型下满足强 *EUF-CMA*安全性。

Proof. 方案的公钥 $\hat{\mathbf{a}}$ 定义了一个抗碰撞 PSF $f_{\hat{\mathbf{a}}} : \mathcal{R}^{2+k} \to \mathcal{R}_q$。根据 [7] 中定理 5.9，该抗碰撞 PSF 基于 Ring-SIS 假设。假设存在 P.P.T. 的攻击者 $\mathcal{A}$ 能以不可忽略的概率 p 进行签名伪造，下面构造 P.P.T. 的攻击者 $\mathcal{S}$ 以接近 p 的概率攻破 $f_{\hat{\mathbf{a}}}$ 的抗碰撞性，进而攻破 Ring-SIS 假设。

给定公钥 $\hat{\mathbf{a}}$，$\mathcal{S}$ 按如下方式模拟哈希函数 $\mathcal{H}$ 及签名询问（不失一般性，假设 $\mathcal{A}$ 在询问消息的签名前已经进行了哈希询问）：

- 哈希询问：$\forall\ \mu \in \{0,1\}^*$，$\mathcal{S}$ 选取 $\sigma_\mu \leftarrow \mathcal{D}_{\mathbb{Z}^m,s}$，保存 (μ, σ_μ)，并将 $f_{\hat{\mathbf{a}}}(\sigma_\mu)$ 返回给 $\mathcal{A}$
- 签名询问：$\mathcal{S}$ 查找 (μ, σ_μ)，并将 σ_μ 返回给 $\mathcal{A}$

假设 $\mathcal{A}$ 输出消息 μ^* 的伪造签名 σ^* 前已进行哈希询问，即 $\mathcal{S}$ 已经保存 (μ^*, σ_{μ^*})。 当 $\mathcal{A}$ 输出伪造签名 σ^* 后，$\mathcal{S}$ 输出 $(\sigma_{\mu^*}, \sigma^*)$ 作为 $f_{\hat{\mathbf{a}}}$ 的碰撞。

根据 [7] 中推论 2.8，σ_μ 和 $\mathbb{Z}_q^n$ 上的均匀分布统计不可区分，因此 $\mathcal{A}$ 无法区分真实的签名伪造和 $\mathcal{S}$ 的模拟环境。此外，给定消息的哈希值 $\mathbf{u}$，签名询问的输出和真实签名的输出均近似服从 $\mathcal{D}_{\Lambda_{\mathbf{u}}^{\perp}(\hat{\mathbf{a}}),s}$。 因此，$\mathcal{A}$ 以接近 p 的概率输出有效的伪造 (μ^*, σ^*)。最后，由于 $\mathcal{A}$ 输出伪造签名前已进行哈希询问，根据 PSF 的原像最小熵性质（[7] 引理2.10），$\sigma_{\mu^*} \neq \sigma^*$ 以接近 1 的概率成立。 □

3.2 安全强度

针对不小于128比特、256比特的安全强度，本方案的两组参数集如下。

Table 1: 两组参数集

λ	n	m	q	k	r	s
128	512	13312	16770049	24	5.6	44873
256	1024	30720	268369921	28	7.8	131360

文献 [1, 2, 4] 中的结果表明，当前求解格上 SVP_γ 最高效算法的时间复杂度约为 $2^{0.292n}$；在量子计算环境下，时间复杂度约为 $2^{0.265n}$。 因此，我们的签名方案在经典计算环境下可以达到 149 比特或 299 比特的安全性；在量子计算环境下达到 135 比特或 271 比特的安全性。

3.3 失败率

本方案存在如下失败情况：

- 密钥生成阶段

  私钥 $\hat{\mathbf{e}}, \hat{\mathbf{r}} \sim \mathcal{D}_{\mathbb{Z},2\sqrt{n}}^{n\times k}$，参数 s 需满足 $s \geq 3\cdot r\cdot s_1(\mathbf{T})$，其中 $\mathbf{T} = [\mathbf{E}^t\ \mathbf{R}^t\ \mathbf{I}]^t$，根据文献 [10] 中引理 2.9，当 $s = 2.4\cdot r\cdot\sqrt{n}\cdot(\sqrt{nk}+\sqrt{2n}+4.7)$ 时，会有至多 2^{-100} 的概率使得 $s < 3\cdot r\cdot s_1(\mathbf{T})$，此时，可能的后果是方案无法达到所声称的 149 比特或 299 比特安全。

- 签名阶段

  - 签名失败

    签名 σ 需要满足 $\|\sigma\| \leq s\sqrt{m}$。本方案的签名算法输出 $\sigma \sim \mathcal{D}_{\Lambda_u^{\perp}(\hat{\mathbf{a}}),s}$。根据文献 [11] 中引理 4.3，签名失败的概率小于 $2^{1-m} \leq 2^{-13311}$。

  - 得到小向量

    假设攻击者进行 Q_s 次签名询问。根据生日悖论，攻击者以大约 $Q_s\cdot q^{-n/2}$ 的概率找到 $\mathcal{H}(\cdot)$ 上的一对碰撞，此时，签名算法会以接近 1 的概率得到格上同一陪集的两个短向量，它们之差即为格 $\Lambda^{\perp}(\hat{\mathbf{a}})$ 上的短向量，这攻破了 Ring-SIS 假设。

4 性能分析

本算法使用 C 语言，在软件平台（Intel 酷睿 $i3$ 处理器，4GB 内存，Windows 10 64位操作系统，DEV-C++）实现。计算量和存储需求估计分别如表 2 表 3 所示。

Table 2: 计算量需求（ms）

λ	keygen	sign	verf
128	12.15	99.05	10.25
256	31.7	271.4	23.4

Table 3: 存储需求（KB）

λ	pk	sk	σ
128	52	96	52
256	104	192	104

5 优缺点声明

- 优点
  - 本方案能够抵御量子攻击，在量子计算环境下，可分别达到 135 和 271 比特安全
  - 算法运行过程中需要较少的存储空间，并且通过 FFT 可以达到较高的运算的效率
  - 灵活的参数选择，当 n 不为 2 的幂次时，通过 [6, 9] 中的技术，算法依然保持高效
  - 算法可以通过修改算法提供额外功能，例如，消息恢复签名、基于身份的加密等
  - 签名过程可以分为 on-line/off-line 两个阶段，SampleP 以及 SampleG 中的步骤 1-4 都属于 off-line 阶段，可以提前执行并将结果保存，留作后用
  - 算法可并行实现
- 缺点
  - 算法中包含多精度浮点型运算
  - 算法中的高斯采样需要精心实现来抵抗侧信道攻击

References

[1] Alkim, E., Ducas, L., Poppelmann, T., and Schwabe, P.. Post-quantum Key Exchange: a New Hope. In USENIX Security 2016, https://eprint.iacr.org/2015/1092.

[2] Albrecht, M. R., Player, R., and Scott, S.. On the concrete hardness of Learning with Errors. In J. Mathematical Cryptology, vol. 9(3), 169–203, 2015.

[3] Boneh, D., Dagdelen, Ö., Fischlin, M., Lehmann, A., Schaffner, C., and Zhandry, M.. Random oracles in a quantum world. In ASIACRYPT 2011, 41–69, 2011.

[4] Chen, Y, Nguyen, P. Q.: BKZ 2.0: better lattice security estimates [C]. In: Proc. ASIACRYPT 2011, 1–20 2011.

[5] del Pino, R., Lyubashevsky, V., and Pointcheval, D.. The whole is less than the sum of its parts: Constructing more efficient lattice-based AKEs. In SCN 16, 273–291, 2016.

[6] Genise, N., Micciancio, D.: Faster gaussian sampling for trapdoor lattices with arbitrary modulus. In EUROCRYPT 2018, 174–203, 2018.

[7] Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for Hard Lattices and New Cryptographic Constructions. In STOC 2008, 197–206, 2008.

[8] Lyubashevsky, V., Peikert, C., and Regev, O.. On Ideal Lattices and Learning with Errors over Rings. In Eurocrypt 2010, 1–23, 2010.

[9] Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-lwe cryptography. In EUROCRYPT 2013, 35–54, 2013.

[10] Micciancio, D., Peikert, C.: Trapdoors for lattices: Simpler, tighter, faster, smaller. In EUROCRYPT 2012, 700–718, 2012.

[11] Micciancio, D., Regev, O.: Worst-case to average-case reductions based on Gaussian measure. SIAM Journal on Computing 37(1): 267–302, 2007. Preliminary version in FOCS 2004.

[12] Peikert, C.: An efficient and parallel gaussian sampler for lattices. In CRYPTO 2010, 80–97, 2010.

[13] Schnorr C.-P., Euchner M.: Lattice basis reduction: improved practical algorithms and solving subset sum problems [J]. Math. Programming 66, 181–199, 1994.