

OKCN-SEC：基于理想格 RLWE 的 通用和模块化的密钥协商协议

算法说明书

赵运磊（复旦大学）

刘哲（南京航空航天大学）

金正中（复旦大学）

巩博儒（复旦大学）

隋光烨（上海扈民区块链科技有限公司）

刘伟超（银联商务股份有限公司）

文黎明（银联商务股份有限公司）

2019 年 1 月

摘 要

基于我们所发展的性能几乎最优的 OKCN 机制，在本提案中我们设计与分析基于理想格 RLWE 的密钥协商协议。

在一般格、模格和理想格中，我们认为理想格 RLWE 的安全性是最弱的。另外，我们注意到在 128 比特后量子安全级别或会话密钥长度不低于 512 比特时，基于理想格 RLWE 的密码协议的综合性能逊色于基于模格 MLWE 的密码协议。因此，在本提案中我们坚持采用大约 256 比特后量子安全性的参数以及至少 512 比特的协商密钥长度。这种保守的参数取法使得我们相信我们所选取的 RLWE 的参数在较长的时间周期均可提供至少 128 比特的后量子安全。

当把 OKCN 直接应用到基于 RLWE 的密钥协商时，在我们保守选取的参数下错误率仍偏高。本提案的主要创新和贡献是为降低错误率所发展的新型技术。对于进一步降低错误率，有两种技术路线：一种是采用格编码，另一种是采用额外的纠错码。

对于基于纠错码的技术路线，在我们工作之前，业界普遍假设会话密钥不同位置出错的概率是独立的，但是缺少严格论证。我们首先严格证明了，当 n 足够大时会话密钥不同位置出错的确是趋于独立的。然后，基于扩展汉明码，我们设计了没一个 block 仅纠一位错误的纠错码 SEC (single-error correction, 单比特纠错码)。在已公开的基于纠错码的理想格密钥协商和封装机制中，SEC 是最为简单的并且非常便于常数实现。

目 录

1	引言.....	1
1.1	我们的贡献.....	2
2	预备知识.....	3
2.1	基础 LWE 和 LWR 问题.....	3
3	基本工具.....	4
3.1	Key Consensus with Noise.....	4
3.1.1	KC 的有效上界.....	6
3.1.2	OKCN 的构造和分析.....	7
3.2	Asymmetric Key Consensus with Noise.....	11
3.2.1	AKCN 的构造和分析.....	13
4	基于理想格 RLWE 的密钥协商方案.....	16
4.2	在不同位置错误的独立性.....	18
4.3	利用单比特纠错码降低错误率.....	22
4.3.1	单比特纠错码.....	22
4.3.2	带有 SEC 码的 KC 算法.....	23
4.3.3	在公钥密码设定中 OKCN-SEC 的 KEM 规范.....	25
4.6	扩展到基于 RLWR (Ring-LWR) 的 KE.....	26
5	优缺点声明.....	27
5.1	基于 KC 密钥协商优缺点的一般性讨论.....	27
5.2	OKCN/AKCN-SEC、OKCN/AKCN-E8 与 NewHope 比较.....	28
5.3	算法实现代码及性能测试.....	29
6.1.	基于理想格的 OKCN-SEC 性能分析.....	30
6.1.1	空间消耗.....	30
	OKCN-SEC-RLWE 算法的空间消耗.....	30
6.1.2	时间与时钟周期消耗.....	31
6	支持文档.....	31
7	参考文献.....	32
8	原创性声明.....	37
9	未来工作.....	38

1 引言

一旦实用化量子计算机出现，大多数基于普通离散对数、椭圆离散对数或者大数因素分解的公钥密码系统都将会被攻破，它们的安全性也就无从谈起。许多科学家认为，量子计算机目前面临的仅仅是工程实现上的挑战，并且 IBM 的工程师们预测在未来的二十年内必将被大规模应用。回顾公钥密码学的发展历史，现代密码学基础设施的部署几乎花费了二十年，因此无论我们是否能够准确预测量子计算时代的到来时间，我们都需要将目前的信息安全系统提升到抗量子级别。此外，如果想要让我们目前所有想要保密的文件等在 15 年或者更久之后依然具有很高的安全性，那么，我们就必须要从现在开始将所有的密码技术替换为抗量子版本，如文献所述[2,6]，在非对称密码学领域最关键的技术就是密钥协商。

格密码是目前对抗量子攻击的主要数学方法之一。在密码学的环境下，和其他古典的格困难问题（例如 SVP 和 CVP）相比，LWE（Learning With Error）问题已经被证明功能更加全面 [47]。在近些年，大多数研究 [19,41,12,2,11,47,25,34,6,37,46] 工作都集中在基于 LWE 及其变体的密钥协商和加密协议设计上，并且已经在实现一个实用的基于 LWE 的密钥协商协议方向取得了巨大的进展。

从技术的角度来说，最新的关于 LWE 及其变体的研究工作中的一个主要的贡献是改善了密钥协商机制。但是在之前的研究工作中，密钥协商仅在 KE 和 PKE 中使用和分析过，并且还是以一种非黑盒的形式进行分析的。这也就是说，对于未来用来构建格基密码系统的新的密钥协商机制，我们需要从头开始分析它们的安全性。此外，对于密钥协商中的不同的参数，我们仍不清楚这些参数之间需要满足什么样的上界条件。因此，对于如何去评估不同的密钥协商机制以及判断这些不同的机制是否能够进一步改善，我们依然缺乏一个基本的标准。

抽象化和一般化是自然科学（数学、物理）的基础，对于密码学来说尤其重要。例如，在数字签名领域中，Schnorr 签名就是首先抽象化 Σ 协议然后利用 Fiat-Shamir 转换进行一般化而得到的。类似的抽象化和一般化同样在 CCA 安全的 PKE 以及在现代密码学的很多领域扮演了重要的角色。抽象化和一般化在实际

中也是非常有用的，并且可能可以用于改进格基密码，因为格基密码更加难以理解和评估。

对于一些密码学应用而言，例如伪随机函数，我们需要一个没有随机性版本的 LWE 问题，这就驱动了 learning with error (LWR) 问题的产生。LWR 中确定的噪音也减轻了密码协议对随机数的依赖，然而，LWR 中确定的噪音和密钥之间具有相关性，又使对基于 LWR 的密钥协商的错误率的分析变得十分复杂。

1.1 我们的贡献

基于我们所发展的性能几乎最优的 OKCN 机制，在本提案中我们设计与分析基于理想格 RLWE 的密钥协商协议。

在一般格、模格和理想格中，我们认为理想格和 RLWE 的安全性是最弱的。另外，根据我们的研究观察，我们注意到在 128 比特后量子安全级别或会话密钥长度不低于 512 比特时，基于理想格 RLWE 的密码协议的综合性能逊色于基于模格 MLWE 的密码协议。因此，在本提案中我们坚持采用大约 256 比特后量子安全性的参数以及至少 512 比特的协商密钥长度。这种保守的参数取法使得我们相信我们所选取的 RLWE 的参数在较长的时间周期均可提供至少 128 比特的后量子安全。

当把 OKCN 直接应用到基于 RLWE 的密钥协商时，在我们保守选取的参数下错误率仍偏高。本提案的主要创新和贡献是为降低错误率所发展的新型技术。对于进一步降低错误率，有两种技术路线：一种是采用格编码，另一种是采用额外的纠错码。

对于基于纠错码的技术路线，在我们工作之前，业界普遍假设会话密钥不同位置出错的概率是独立的，但是缺乏严格论证。我们首先严格证明了，当 n 足够大时会话密钥不同位置出错的确是趋于独立的。然后，基于扩展汉明码，我们设计了没一个 block 仅纠一位错误的纠错码 SEC。在已公开的基于纠错码的理想格密钥协商和封装机制中，SEC 是最为简单的并且非常便于常数时间下的实现。

2 预备知识

在本文字符串或者值 α 都以二进制表示, $|\alpha|$ 表示 α 二进制的长度。对于任意实数 x , $\lfloor x \rfloor$ 表示小于等于 x 的最大整数, $\lfloor x \rfloor = \lfloor x + 1/2 \rfloor$ 。对于任意的正整数 a 和 b , 用 $\text{lcm}(a, b)$ 表示 a 和 b 的最小公倍数。对于任意的 $i, j \in \mathbb{Z}$, 并且 $i < j$, 用 $[i, j]$ 表示整数集合 $\{i, i+1, \dots, j-1, j\}$ 。对于任意的正整数 t , 令 \mathbb{Z}_t 表示 $\mathbb{Z}/t\mathbb{Z}$ 。 \mathbb{Z}_t 中的元素默认表示为 $[0, t-1]$, 但有时 \mathbb{Z}_t 会明确表示为 $[-\lfloor (t-1)/2 \rfloor, \lfloor t/2 \rfloor]$ 。

如果 S 是一个有限集合, 那么 $|S|$ 表示它的基数, 并且 $x \leftarrow S$ 表示均匀随机的从 S 中取一个元素。对于两个集合 $A, B \subseteq \mathbb{Z}_q$, 我们定义 $A + B \triangleq \{a + b \mid a \in A, b \in B\}$ 。对于一个加法群 $(G, +)$, 元素 $x \in G$ 并且子集 $S \subseteq G$, $x + S$ 表示将 S 中每一个元素都和 x 相加结果的集合。对于一个集合 S , 用 $\mathcal{U}(S)$ 表示 S 的一个均匀分布。对于任意的 \mathbb{R} 中的离散随机变量 X , $\text{Supp}(X) = \{x \in \mathbb{R} \mid \Pr[X = x] > 0\}$ 。

在后面的概率相关的算法、实验和交互协议当中, 我们使用传统的符号和概念。如果 \mathcal{D} 表示一个概率分布, 那么 $x \leftarrow \mathcal{D}$ 表示根据 \mathcal{D} 选择一个元素并赋值给 x 。如果 α 既不是一个算法也不是一个集合, 那么 $x \leftarrow \alpha$ 就表示简单的赋值操作。如果 A 是一个概率算法, 那么 $A(x_1, x_2, \dots; r)$ 表示将 x_1, x_2, \dots 作为输入, r 为随机种子 A 的运算结果。我们用 $y \leftarrow A(x_1, x_2, \dots)$ 表示随机选取 r 并令 y 为 $A(x_1, x_2, \dots; r)$ 的实验。用 $\Pr[R_1; \dots; R_n; E]$ 表示事件 E 在一连串有序的随机过程 R_1, \dots, R_n 之后发生的概率。

如果对于任意的 $c > 0$, 对于所有的 $\lambda > \lambda_c$, 都存在一个 λ_c 使得 $f(\lambda) < 1/\lambda^c$, 那么函数 $f(\lambda)$ 是可忽略的。在本文中, 当涉及到具体的参数时, 我们会说小于 2^{-60} 的数量级是可忽略的。

2.1 基础 LWE 和 LWR 问题

给定正连续数 $\sigma > 0$, 对 $x \in \mathbb{R}$, 定义高斯函数 $\rho_\sigma(x) \triangleq \exp(-x^2/2\sigma^2)/\sqrt{2\pi\sigma^2}$ 。令 $D_{\mathbb{Z}, \sigma}$ 表示在 \mathbb{Z} 上的一维离散高斯分布, 此由其概率密度函数 $D_{\mathbb{Z}, \sigma}(x) \triangleq \rho_\sigma(x)/\rho_\sigma(\mathbb{Z})$, $x \in \mathbb{Z}$ 决定。最后, 令 $D_{\mathbb{Z}^n, \sigma}$ 表示在 \mathbb{Z}^n 上的 n 维球面离散高斯分布, 其中每个坐标都独立于 $D_{\mathbb{Z}, \sigma}$ 。

给定正整数 n 和 q ，它们都是安全参数 λ 中的多项式，并给定整数向量 $\mathbf{s} \in \mathbb{Z}_q^n$ 和一个 \mathbb{Z}_q 上的概率分布 χ ，通过随机均匀选择 $\mathbf{a} \in \mathbb{Z}_q^n$ ，令 $A_{q,\mathbf{s},\chi}$ 是 $\mathbb{Z}_q^n \times \mathbb{Z}_q$ 上的分布，误差项 $e \leftarrow \chi$ ，并输出 $(\mathbf{a}, \mathbf{b} = \mathbf{a}^T \mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ 。误差分布 χ 通常被认为是离散高斯概率分布 $D_{\mathbb{Z},\sigma}$ ；但是，如文献[11]中所述，也可以采用其他的 χ 分布。简而言之，（决定性）Learning With Error (LWE) 假设认为，对足够大的安全参数 λ ，概率多项式时间算法无法以不可忽略的概率来区分 $A_{q,\mathbf{s},\chi}$ 和 $\mathbb{Z}_q^n \times \mathbb{Z}_q$ 上的均匀分布。即使 \mathcal{A} 看到多项式多个样本，并且即使秘密向量 \mathbf{s} 是从 χ^n 随机抽取的，这也是成立的。

LWR 问题是 LWE 问题的一个“随机”变体。令 \mathcal{D} 是在 \mathbb{Z}_q^n 上的一些分布且 $\mathbf{s} \leftarrow \mathcal{D}$ 。对整数 $q \geq p \geq 2$ 且任意 $x \in \mathbb{Z}_q$ ，表示

$$\lfloor x \rfloor_p = \left\lfloor \frac{p}{q} x \right\rfloor$$

之后，对任意正整数 n 和 $q \geq p \geq 2$ ，在 $\mathbb{Z}_q^n \times \mathbb{Z}_q$ 上的 LWR 分布 $A_{n,q,p}(\mathbf{s})$ 是由从 \mathbb{Z}_q^n 上均匀随机取样的 \mathbf{a} 得到的，且输出 $(\mathbf{a}, \lfloor \mathbf{a}^T \mathbf{s} \rfloor_p) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ 。简而言之，（决定性）LWR 假设认为，对足够大的安全参数，概率多项式时间算法无法以不可忽略的概率来区分 $A_{n,q,p}(\mathbf{s})$ 和 $u \leftarrow \mathbb{Z}_q$ 时的分布 $(\mathbf{a} \leftarrow \mathbb{Z}_q^n, \lfloor u \rfloor_p)$ 。即使 \mathcal{A} 看到多项式多个样本，这也是成立的。对于超多项式大 q ，[9]给出了一个从 LWE 问题到 LWR 问题的有效约简。令 B 表示秘密 \mathbf{s} 中任意组成部分的边界。当 $q \geq 2mBp$ ($m \leq q/2Bp$)，LWE 问题可以被约简为有 m 个独立随机样本的（决定性）LWR 假设。此外，从 LWE 到 LWR 的约简与秘密 \mathbf{s} 的分布无关。

3 基本工具

3.1 Key Consensus with Noise

在介绍密钥共识 (Key Consensus, KC) 的完整定义之前，我们首先引入一个函数 $|\cdot|_t$ ，其中 $t \geq 1$: $|x|_t = \min\{x \bmod t, t - x \bmod t\}, \forall x \in \mathbb{Z}$ ，并且模运算的

结果表示在 $\{0, \dots, (t-1)\}$ 中，例如 $|-1|_t = \min\{-1 \bmod t, (t+1) \bmod t\} = \min\{t-1, 1\} = 1$ 。在后面的描述中，我们使用 $|\sigma_1 - \sigma_2|_q$ 来表示两个元素 $\sigma_1, \sigma_2 \in \mathbb{Z}_q$ 之间的距离。

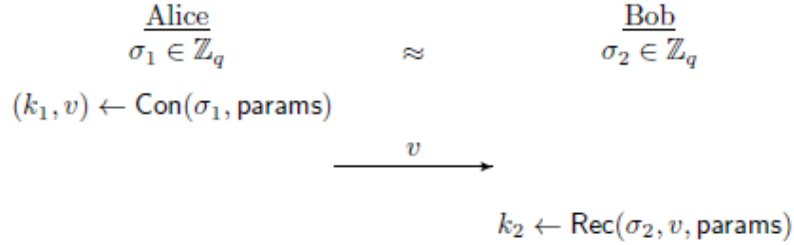


图 3.1 一个简单的密钥共识算法

定义 3.1 一个密钥共识算法 $KC = (\text{params}, \text{Con}, \text{Rec})$ ，如图 3.1 所示 ($k_1, k_2 \in \mathbb{Z}_m, v \in \mathbb{Z}_q$ 并且 $|\sigma_1 - \sigma_2|_q \leq d$)，其中参数的定义如下：

- $\text{params} = (q, m, g, d, \text{aux})$ 表示系统参数，其中 q, m, g, d 为正整数并且满足 $2 \leq m, g \leq q, 0 \leq d \leq \lfloor \frac{q}{2} \rfloor$ ， aux 表示辅助信息，通常由 (q, m, g, d) 确定，其值可以设定为 ϕ 表示值为空；
- $(k_1, v) \leftarrow \text{Con}(\sigma_1, \text{params})$ ：在输入为 $(\sigma_1 \in \mathbb{Z}_q, \text{params})$ 的条件下，概率多项式算法 Con 的输出为 (k_1, v) ，其中 $k_1 \in \mathbb{Z}_m$ 并且 k_1 为共享密钥， $v \in \mathbb{Z}_g$ 并且 v 为提示信号并在后续的过程中公开传输给另一个节点，从而使得双方能达到共识；
- $k_2 \leftarrow \text{Rec}(\sigma_2, v, \text{params})$ ：在输入为 $(\sigma_2 \in \mathbb{Z}_q, v, \text{params})$ 的情况下，确定的多项式时间一致算法 Rec 输出为 $k_2 \in \mathbb{Z}_m$ 。

正确性： 一个密钥共识算法满足正确性，如果对于任意的 $\sigma_1, \sigma_2 \in \mathbb{Z}_q$ 且 $|\sigma_1 - \sigma_2|_q \leq d$ ，都有 $k_1 = k_2$ 。

安全性： 一个密钥共识算法，满足安全性，如果 k_1 和 v 是相互独立的，并且无论 $\sigma_1 \in \mathbb{Z}_q$ 的值为多少， k_1 都均匀分布在 \mathbb{Z}_m 。算法中的随机性只来源于 σ_1 取样的概率和 Con 中使用的随机种子。

3.1.1 KC 的有效上界

下面的这个定理提出了一个上界，其中的参数为 q （控制安全性和效率）， m （共识密钥范围的参数）， g （带宽参数）和 d （错误率参数），有了这个上界，我们就可以根据不同的优先级在这些参数之间取得一个平衡。

定理 3.1 如果 $KC = (\text{params}, \text{Con}, \text{Rec})$ 满足正确性和安全性的密钥共识机制，并且 $\text{params} = (q, m, g, d, \text{aux})$ ，那么 $2md \leq q(1 - \frac{1}{g})$ 。

在证明定理 3.1 之前，我们首先证明下面的命题。

命题 3.1 给定一个参数为 $\text{params} = (q, m, g, d, \text{aux})$ 并且满足正确性和安全性的 KC 算法，对任意固定的 $\sigma_1 \in \mathbb{Z}_q$ ，如果 $\text{Con}(\sigma_1, \text{params})$ 以正数的概率输出 (k_1, v) ，那么对于某一组 (v, σ_1) ， k_1 也是确定的。也就是说，对于任意随机的 (r, r') ，如果 $\text{Con}(\sigma_1, \text{params}, r) = (k_1, v)$ 并且 $\text{Con}(\sigma_1, \text{params}, r') = (k'_1, v)$ ，那么 $k_1 = k'_1$ 。

证明-命题 3.1 令 $\sigma_2 = \sigma_1$ ，那么 $|\sigma_1 - \sigma_2|_q = 0 \leq d$ 。那么，根据 KC 的正确性，我们可以得到 $k_1 = k_2 = \text{Rec}(\sigma_2, v) = \text{Rec}(\sigma_1, v)$ 。然而， Rec 是一个确定的算法，对于一组特定的 (σ_1, v) ， k_2 也是固定的。因此，对于任何一组固定的 (σ_1, v) ，无论 Con 中使用了怎样的随机性， k_1 也是都固定的。

命题 3.2 给定一个参数为 $\text{params} = (q, m, g, d, \text{aux})$ 并且满足正确性和安全性的 KC 算法，对于任意的 $v \in \mathbb{Z}_g$ ，令 S_v 表示包含 σ_1 的集合，使得对于每一个 σ_1 都满足 $\text{Con}(\sigma_1, \text{params})$ 以正数概率输出 v 。具体来说，

$$S_v = \{\sigma_1 \in \mathbb{Z}_q \mid \Pr[(k_1, v') \leftarrow \text{Con}(\sigma_1, \text{params}) : v' = v] > 0\}.$$

那么，存在 $v_0 \in \mathbb{Z}_g$ 使得 $|S_{v_0}| \geq q/g$ 。

证明-命题 3.2 对于任何一个 $\sigma_1 \in \mathbb{Z}_q$ ，我们通过运行 $\text{Con}(\sigma_1, \text{params})$ 可以得到一组 $(k_1, v) \in \mathbb{Z}_m \times \mathbb{Z}_g$ 并且 $\sigma_1 \in S_v$ ，那么通过鸽巢原理我们就可以直接证明命题 3.2。

证明-定理 3.1 由命题 3.2 可知，存在一个 $v_0 \in \mathbb{Z}_g$ 使得 $|S_{v_0}| \geq q/g$ 。需要注意的是，对于任意的 $\sigma_1 \in S_{v_0}$ ， $\text{Con}(\sigma_1, \text{params})$ 输出 v_0 的概率大于 0。对于任意 $i \in \mathbb{Z}_m$ ，令 K_i 表示包含所有满足 $\text{Con}(\sigma_1, \text{params})$ 以不为零的概率输出 $(k_1 = i, v = v_0)$ 的 σ_1

的集合。由命题 3.1 可得, K_i 和 S_{v_0} 是相互分离的。因为 k_1 和 v 是相互独立的, 并且 k_1 是均匀分布的, 因为我们假设了底层的 KC 是安全的, 并且我们知道 $\Pr[k_1 = i | v = v_0] = \Pr[k_1 = i] > 0$, 因此对于每一个 $i \in \mathbb{Z}_m$, K_i 都是非空的。现在对于每一个 $i \in \mathbb{Z}_m$, 用 K_i' 表示包含所有 $\sigma_2 \in \mathbb{Z}_q$ 满足 $\text{Rec}(\sigma_2, v_0, \text{params}) = i$ 的集合, 因为 Rec 是确定的, 那么 K_i' 也是确定的并且是相互没有交集的。

因为 KC 满足正确性, 那么对于所有的 $\sigma_1 \in K_i, |\sigma_1 - \sigma_2|_q \leq d$, 我们都有 $\sigma_2 \in K_i'$, 也就是说 $K_i + [-d, d] \subseteq K_i'$ 。那么我们要证明的就是 $K_i + [-d, d]$ 至少包含 $|K_i| + 2d$ 个元素。如果 $K_i + [-d, d] = \mathbb{Z}_m$, 那么 $m = 1$, 这也就产生了矛盾 (我们在 KC 的定义中排除了 $m = 1$ 这种情况, 因为这种情况太简单了)。如果存在一个 $x \in \mathbb{Z}_m$ 使得 $x \notin K_i + [-d, d]$, 我们可以将 \mathbb{Z}_m 看成一条线段, 线段从 x 开始后面的每一个元素分别为 $(x+1) \bmod m, (x+2) \bmod m, \dots, (x+m-1) \bmod m$ 。令 l 表示线段在区间 $K_i + [-d, d]$ 的最左边的元素, 并且 r 为该区间的最后边的元素。那么 $K_i + [-d, d]$ 在区间 $[l, r]$ 内至少包含 $|K_i|$ 个元素。因为 $l + [-d, 0]$ 和 $r + [0, d]$ 都是 $K_i + [-d, d]$ 的子集, 并且互相没有交集 (因为 $x \notin K_i + [-d, d]$), 所以集合 $K_i + [-d, d]$ 至少包含 $|K_i| + 2d$ 个元素。

现在我们有 $|K_i + 2d| \leq |K_i'|$, 当我们在两边分别加上 $i \in \mathbb{Z}_m$ 时, 我们可以推出 $|S_{v_0}| + 2md \leq q$, 再结合 $|S_{v_0}| \geq q/g$, 就可以完成该定理的证明。

3.1.2 OKCN 的构造和分析

算法 1 OKCN: 带噪音的对称密钥共识算法

```

1:  $\text{params} = (q, m, g, d, \text{aux}), \text{aux} = \{q' = \text{lcm}(q, m), \alpha = q'/q, \beta = q'/m\}$ 
2: procedure CON( $(\sigma_1, \text{params})$ )  $\triangleright \sigma_1 \in [0, q -$ 
3:    $e \leftarrow [-(\alpha - 1)/2, [\alpha/2]]$ 
4:    $\sigma_A = (\alpha\sigma_1 + e) \bmod q'$ 
5:    $k_1 = \lfloor \sigma_A / \beta \rfloor \in \mathbb{Z}_m$ 
6:    $v' = \sigma_A \bmod \beta$ 
7:    $v = \lfloor v'g / \beta \rfloor$   $\triangleright v \in \mathbb{Z}$ 
8:   return  $(k_1, v)$ 
9: end procedure
10: procedure REC( $(\sigma_2, v, \text{params})$ )  $\triangleright \sigma_2 \in [0, q -$ 
11:    $k_2 = \lfloor \alpha\sigma_2 / \beta - (v + 1/2)/g \rfloor \bmod m$ 
12:   return  $k_2$ 
13: end procedure
```

算法 1 展示了一个密钥共识算法, 我们称之为 OKCN (Optimal Key Consensus

with Noise), 图 3.2 对 OKCN 进行了简单的描述, 下面我们将对算法 1 进行详细的解释。

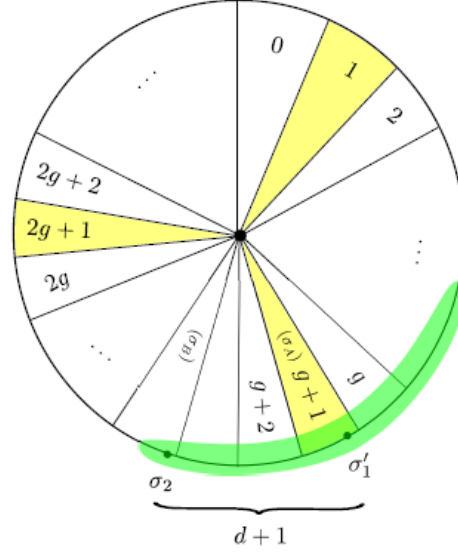


图 3.2 OKCN 算法的说明

我们定义 $\sigma'_A = \alpha\sigma_1 + e$, 则 $\sigma'_A < q'$ 始终成立。然而在某些特殊情况下 σ'_A 可能为负数, 例如, 当 $\sigma_1 = 0$ 并且 $e \in [-(\alpha - 1)/2, -1]$ 。在算法 1 的第 4 行中, 设置 $\sigma_A = \sigma'_A \bmod q'$ 是为了确保 σ_A 始终是 $\mathbb{Z}_{q'}$ 中的一个非负数, 这可以被简化表示为: 如果 $\sigma'_A < 0$, 那么令 $\sigma_A = \sigma'_A + q'$; 否则令 $\sigma_A = \sigma'_A$ 。考虑到敌手采用时间攻击的可能, 在判断 σ'_A 的正负性时可以对 σ'_A 的符号位进行取反操作来判断。具体来说, 假设 σ'_A 是一个 16 比特的有符号或者无符号整数, 那么我们在写代码时可以 $\sigma_A = \sigma'_A + ((\sigma'_A >> 15) \& 1) * q'$ 这样写来避免时间攻击。同样的技巧可以用于算法 1 中第 11 行的计算。

在第 5 行和第 6 行中, (k_1, v') 的计算在汇编语言中可以使用一个简单的 *div* 指令就可以算出两个变量的值。在第 11 行, 浮点数运算可以用整数运算代替, 如果 m 足够小, 例如 2 或者 3, 那么复数除法可以用相对较快的条件语句来替换。

第 11 行中的 $v + 1/2$ 估计了 $v'g/\beta$ 的准确值, 如果使用所有的 $v'g/\beta$ 的均值例如 $\lfloor v'g/\beta \rfloor = v$ 来进行估计, 那么这种估计可以更加准确一点。尽管这种估计能让正确性的边界提高一点, 但是却会使得 k_2 的计算变得更加复杂。

下面的这个事实可以直接从 $|\cdot|_t$ 的定义得到。

事实 3.1 对于任意的 $x, y, t, l \in \mathbb{Z}$, 其中 $t \geq 1, l \geq 0$, 如果 $|x - y|_q \leq l$, 那么存在 $\theta \in \mathbb{Z}, \delta \in [-l, l]$ 使得 $x = y + \theta t + \delta$ 。

定理 3.2 OKCN 算法满足安全性。具体来说, 当变量之间满足 $\sigma_1 \leftarrow \mathbb{Z}_q$, k_1 和 v 是相互独立的, 并且 k_1 均匀分布在 \mathbb{Z}_m 中, 其中概率取自 σ_1 取样的概率和 Con 中使用的随机性时, OKCN 算法是安全的。

证明-定理 3.2 在之前的条件中, 我们有 $q' = \text{lcm}(q, m), \alpha = q'/q, \beta = q'/m$ 。我们首先说明 σ_A 是服从 $\mathbb{Z}_{q'}$ 中的均匀分布。假设有映射 $f: \mathbb{Z}_q \times \mathbb{Z}_\alpha \rightarrow \mathbb{Z}_{q'}; f(\sigma, e) = (\alpha\sigma + e) \bmod q'$, 其中 \mathbb{Z}_q 和 \mathbb{Z}_α 中的元素以相同的方式进行表示, 因为 $\sigma_1 \leftarrow \mathbb{Z}_q$ 并且 $e \leftarrow \mathbb{Z}_\alpha$ 服从均匀分布, 并且它们之间相互独立, 那么 $\sigma_A = (\alpha\sigma_1 + e) \bmod q' = f(\sigma_1, e)$ 也同样服从 $\mathbb{Z}_{q'}$ 中的均匀分布。

同样, 定义 $f': \mathbb{Z}_m \times \mathbb{Z}_\beta \rightarrow \mathbb{Z}_{q'}$ 使得 $f'(k_1, v') = \beta k_1 + v'$, 那么 f' 显然是个一一映射。从算法 1 中的第 6 行我们有 $f'(k_1, v') = \sigma_A$ 。因为 σ_A 服从 $\mathbb{Z}_{q'}$ 中的均匀分布, (k_1, v') 服从 $\mathbb{Z}_m \times \mathbb{Z}_\beta$ 中的均匀分布, 因此 k_1 和 v' 是相互独立的。而 v 只依赖于 v' , 因此 k_1 和 v 是相互独立的。

定理 3.3 假设系统参数满足 $(2d + 1)m < q(1 - \frac{1}{g})$, 其中 $m \geq 2, g \geq 2$, 那么 OKCN 算法满足正确性。

证明-定理 3.3 假设 $|\sigma_1 - \sigma_2|_q \leq d$, 有事实 3.1 可知, 存在一个 $\theta \in \mathbb{Z}$ 并且 $\delta \in [-d, d]$, 使得 $\sigma_2 = \sigma_1 + \theta q + \delta$ 。从算法 1 的第 4 行和第 6 行可知, 存在 $\theta' \in \mathbb{Z}$, 使得 $\alpha\sigma_1 + e + \theta' q' = \sigma_A = k_1 \beta + v'$ 。从 α 和 β 的定义中, 我们有 $\alpha/\beta = m/q$ 。将这两个等式代入到 Rec (算法 1 的第十一行) 中 k_2 的等式中可得,

$$k_2 = \lfloor \alpha\sigma_2/\beta - (v + 1/2)/g \rfloor \bmod m \quad (2)$$

$$= \lfloor \alpha(\theta q + \sigma_1 + \delta)/\beta - (v + 1/2)/g \rfloor \bmod m \quad (3)$$

$$= \left\lfloor m(\theta - \theta') + \frac{1}{\beta}(k_1\beta + v' - e) + \frac{\alpha\delta}{\beta} - \frac{1}{g}(v + 1/2) \right\rfloor \bmod m \quad (4)$$

$$= \left\lfloor k_1 + \left(\frac{v'}{\beta} - \frac{v + 1/2}{g} \right) - \frac{e}{\beta} + \frac{\alpha\delta}{\beta} \right\rfloor \bmod m \quad (5)$$

注意到 $|v'/\beta - (v + 1/2)/g| = |v'g - \beta(v + 1/2)|/\beta g \leq 1/2g$ 。因此

$$\left| \left(\frac{v'}{\beta} - \frac{v + 1/2}{g} \right) - \frac{e}{\beta} + \frac{\alpha\delta}{\beta} \right| \leq \frac{1}{2g} + \frac{\alpha}{\beta}(d + 1/2)。$$

由假设的条件 $(2d + 1)m < q(1 - \frac{1}{g})$ 我们可以得到右边是严格小于 $1/2$ ，因此，在取整之后， $k_2 = k_1$ 。

另外，当算法 1 中的参数满足 $q = 2^{\bar{q}}, g = 2^{\bar{g}}, m = 2^{\bar{m}}, \bar{q}, \bar{g}, \bar{m} \in \mathbb{Z}$ 时，也就是 q, g, m 都是 2 的次方数时，这个变换就是没有必要的，并且，算法 1 中用来计算 σ_A 的随机噪音 e 也可以不用，因此在这种情况下，Con 和 Rec 可以被简化成算法 2 的形式。

算法 2 OKCN power of 2 (OKCN-2)

```

1: params :  $q = 2^{\bar{q}}, g = 2^{\bar{g}}, m = 2^{\bar{m}}, d, aux = \{(\beta = q/m = 2^{\bar{q}-\bar{m}}, \gamma = \beta/g = 2^{\bar{q}-\bar{m}-\bar{g}})\}$ 
2: procedure CON( $\sigma_1, params$ )
3:    $k_1 = \lfloor \sigma_1 / \beta \rfloor$ 
4:    $v = \lfloor (\sigma_1 \bmod \beta) / \gamma \rfloor$ 
5:   return ( $k_1, v$ )
6: end procedure
7: procedure REC( $\sigma_2, v, params$ )
8:    $k_2 = \lfloor \sigma_2 / \beta - (v + 1/2) / g \rfloor \bmod m$ 
9:   return  $k_2$ 
10: end procedure

```

而当 $\bar{g} + \bar{m} = \bar{q}$ 时，算法 2 可以进一步简化为算法 3 中描述的变量，这进一步放宽了对参数的约束。

算法 3 OKCN Simple

```

1: params :  $q = 2^{\bar{q}}, g = 2^{\bar{g}}, m = 2^{\bar{m}}, d$ , where  $\bar{g} + \bar{m} = \bar{q}$ 
2: procedure CON( $\sigma_1, params$ )
3:    $k_1 = \lfloor \frac{\sigma_1}{g} \rfloor$ 
4:    $v = \sigma_1 \bmod g$ 
5:   return ( $k_1, v$ )
6: end procedure
7: procedure REC( $\sigma_2, v, params$ )
8:    $k_2 = \lfloor \frac{\sigma_2 - v}{g} \rfloor \bmod m$ 
9:   return  $k_2$ 
10: end procedure

```

Con（算法 1 的第 3、4 行）中的第一行和第二行主要是将 \mathbb{Z}_q 中的均匀分布变换到 $\mathbb{Z}_{q'}$ 中的均匀分布中去了。如果将 q, g, m 选定为 2 的次方数，例如 $q = 2^{\bar{q}}, g = 2^{\bar{g}}, m = 2^{\bar{m}}$ 其中 $\bar{q}, \bar{g}, \bar{m} \in \mathbb{Z}$ ，那么这个变换是没有必要的，并且算法 1 中用来计算 σ_A 的随机噪音 e 也是可以不用。在这种情况下，Con 和 Rec 可以简化成算法 2。下面的推论是显而易见的。

推论 3.1 如果 q 和 m 是 2 的次方数，并且 d, g, m 满足 $2md < q(1 - \frac{1}{g})$ ，那么算法 2

中描述的 KC 机制既具有正确性，也具有安全性。

如果我们取 $\bar{g} + \bar{m} = \bar{q}$ ，那么算法 2 可以进一步简化成算法 3，参数的限制也会进一步放松。

推论 3.2 如果 g 和 m 是 2 的次方数，并且 $q = m \cdot g$, $2md < q$ ，那么算法 3 满足正确性和安全性。

证明-推论 3.2 为了正确性，假设 $|\sigma_1 - \sigma_2|_q \leq d$ ，有事实 3.1 可知，存在 $\theta \in \mathbb{Z}$ 并且 $\delta \in [-d, d]$ ，使得 $\sigma_2 = \sigma_1 + \theta q + \delta$ 。考虑到算法 3 的第 8 行，即计算 k_2 的公式，我们有

$$\begin{aligned} k_2 &= \lfloor (\sigma_1 - v + \theta q + \delta) / g \rfloor \bmod m \\ &= (k_1 + \theta m + \lfloor \delta / g \rfloor) \bmod m. \end{aligned}$$

如果 $2md < q$ ，那么 $|\delta/g| \leq d/g < 1/2$ ，所以 $k_2 = k_1 \bmod m = k_1$ 。

对于安全性，作为算法 1 中描述的一种特殊的通用方案，算法 3 的安全性直接来自于算法 1 的安全性。

3.2 Asymmetric Key Consensus with Noise

就像我们之前看到的，在基于 OKCN 的密钥协商中，发送者和响应者在产生最终的共享密钥过程中扮演的角色是对等的，也就是说没有一方可以在密钥协商（KE）协议运行之前预先确定会话密钥。尽管 OKCN 对于（身份认证）密钥协商来说已经足够了，但却并不适用于直接的密钥封装或者是公钥加密。这就驱使我们引入下面的非对称密钥共识（AKC）。

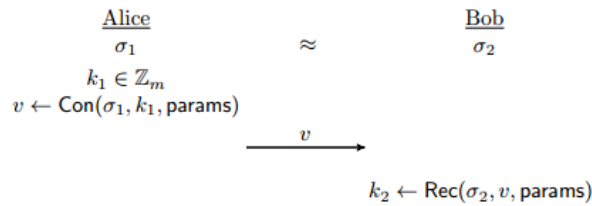


图 3.2 一个简单的非对称密钥共识算法，其中 $k_1, k_2 \in \mathbb{Z}_m, v \in \mathbb{Z}_q$ 并且 $|\sigma_1 - \sigma_2|_q \leq d$

定义 3.2 一个非对称密钥共识算法 $AKC = (\text{params}, \text{Con}, \text{Rec})$ ，如图 3.2 所示，其中参数定义如下：

- $\text{params} = (q, m, g, d, \text{aux})$ 表示系统参数，其中 $2 \leq m, g \leq q, 1 \leq d \leq \lfloor \frac{q}{2} \rfloor$ 并且都为正整数， aux 表示由 (q, m, g, d) 确定的辅助信息，其值可能为空；

- $v \leftarrow \text{Con}(\sigma_1, k_1, \text{params})$: 在输入为 $(\sigma_1 \in \mathbb{Z}_q, k_1 \in \mathbb{Z}_m, \text{params})$ 的条件下, 概率多项式算法 Con 输出公开提示信息 $v \in \mathbb{Z}_g$;
- $k_2 \leftarrow \text{Rec}(\sigma_2, v, \text{params})$: 在输入为 $(\sigma_2, v, \text{params})$ 的情况下, 确定的多项式时间一致算法 Rec 输出为 $k_2 \in \mathbb{Z}_m$ 。

正确性: 一个非对称密钥共识算法是正确的, 如果对于任意的 $\sigma_1, \sigma_2 \in \mathbb{Z}_q$ 且 $|\sigma_1 - \sigma_2|_q \leq d$, 都有 $k_1 = k_2$ 。

安全性: 一个非对称密钥共识算法满足安全性, 如果对于任意的均匀分布在 \mathbb{Z}_q 中的 σ_1 , v 和 k_1 的分布都是相互独立的。另外, 对于任意的 $\tilde{v} \in \mathbb{Z}_g$, 对任意的 $\widetilde{k_1}, \widetilde{k_1'} \in \mathbb{Z}_m$, 都有 $\Pr[v = \tilde{v} | k_1 = \widetilde{k_1}] = \Pr[v = \tilde{v} | k_1 = \widetilde{k_1'}]$, 其中概率取自 $\sigma_1 \leftarrow \mathbb{Z}_q$ 和 Con 中使用的随机性。

当把 AKC 用作密钥封装的底层工具时, k_1 是从 \mathbb{Z}_m 中均匀随机取值的, 然而, 当 AKC 被用作公钥加密时, k_1 可以从明文空间中任意取值。在任意情况下, k_1 都可以离线产生并且作为另一方的输入。

定理 3.4 令 AKC 表示一个参数为 $\text{params} = (q, m, d, g, \text{aux})$ 非对称密钥共识算法, 如果 AKC 满足正确性和安全性, 那么 $2md \leq q(1 - \frac{m}{g})$ 。

将定理 3.4 中的 $2md \leq q(1 - m/g)$ 和定理 4.1 中的 $2md \leq q(1 - 1/g)$ 相比较, 我们可以发现两者之间只相差了一个系数 m , 这表明对于相同的 (q, m, d) , AKC 机制相对于 KC 机制来讲需要使用更大的带宽。

在证明定理 3.4 之前, 我们首先将命题 3.2 变换到 AKC 环境下, 可以得到如下命题。

命题 3.3 给定一个参数为 $\text{params} = (q, m, g, d, \text{aux})$ 且满足正确性和安全性的 AKC 机制, 那么就存在一个 $v_0 \in \mathbb{Z}_g$ 使得 $|S_{v_0}| \geq mq/g$ 。

证明-命题 3.3 如果 k_1 是从 \mathbb{Z}_m 中随机均匀选取的, 通过将 $k_1 \leftarrow \mathbb{Z}_m$; $v \leftarrow \text{Con}(\sigma_1, k_1, \text{params})$ 处理为 $(k_1, v) \leftarrow \text{Con}(\sigma_1, \text{params})$, AKC 就可以被认作是一种特殊的 KC 机制。

令 $S'_v \triangleq \{(\sigma_1, k_1) \in \mathbb{Z}_q \times \mathbb{Z}_m \mid \Pr[v' \leftarrow \text{Con}(\sigma_1, k_1, \text{params}): v' = v] > 0\}$ 。另外,

定义在命题 3.2 中的 S_v 等于包含 $(\sigma_1, \cdot) \in S'_v$ 中出现的所有 σ_1 值的集合。我们对每对 $(\sigma_1, k_1) \in \mathbb{Z}_q \times \mathbb{Z}_m$ 运行 $\text{Con}(\sigma_1, k_1, \text{params})$ 。根据鸽笼原理，一定存在 $v_0 \in \mathbb{Z}_g$ 使 $|S'_{v_0}| \geq qm/g$ 。对 S'_{v_0} 中的任意两对 (σ_1, k_1) 和 (σ'_1, k'_1) ，如果 $\sigma_1 = \sigma'_1$ ，从命题 3.1 中，我们可以导出 $k_1 = k'_1$ ，那么 $(\sigma_1, k_1) = (\sigma'_1, k'_1)$ 。因此，如果 (σ_1, k_1) 和 (σ'_1, k'_1) 是不同的，则 $\sigma_1 \neq \sigma'_1$ ，所以 $|S_{v_0}| = |S'_{v_0}| \geq mq/g$ 。

证明-定理 3.4 从 AKC 的角度看，用 $k_1 \leftarrow \mathbb{Z}_q$ 作为一种特殊的 KC 方案，定理 3.1 证明中的所有推理现在都是正确的。在定理 3.1 证明的最后，我们得到 $|S_{v_0}| + 2md \leq q$ 。根据命题 3.3，取 $|S_{v_0}| \geq mq/g$ ，证明完毕。

3.2.1 AKCN 的构造和分析

算法 4 AKCN: 带噪音的非对称密钥共识算法

```

1: params =  $(q, m, g, d, \text{aux})$ , where  $\text{aux} = \emptyset$ .
2: procedure  $\text{CON}(\sigma_1, k_1, \text{params})$   $\triangleright \sigma_1 \in [0, q-1]$ 
3:    $v = \lfloor g(\sigma_1 + \lfloor k_1 q/m \rfloor) / q \rfloor \bmod g$ 
4:   return  $v$ 
5: end procedure
6: procedure  $\text{REC}(\sigma_2, v, \text{params})$   $\triangleright \sigma_2 \in [0, q-1]$ 
7:    $k_2 = \lfloor m(v/g - \sigma_2/q) \rfloor \bmod m$ 
8:   return  $k_2$ 
9: end procedure

```

算法 4 描述了带噪音的非对称密钥共识算法(Asymmetric Key Consensus with Noise, AKCN)。我们注意到，从某种意义讲 AKCN 可以看作是具有 CPA 安全的公钥加密的一种通用优化共识机制。对于 AKCN，如果想要加速在线计算的性能，我们可以离线计算并存储 k_1 和 $g\lfloor k_1 q/m \rfloor$ 。

定理 3.5 假设 AKCN 的参数满足 $(2d+1)m < q(1 - \frac{m}{g})$ ，那么算法 4 描述的 AKCN 算法满足正确性。

证明-定理 3.5 根据生成 v 的公式，我们知道存在 $\varepsilon_1, \varepsilon_2 \in \mathbb{R}$ 和 $\theta \in \mathbb{Z}$ ，其中 $|\varepsilon_1| \leq 1/2$ 且 $|\varepsilon_2| \leq 1/2$ ，使得

$$v = \frac{g}{q} \left(\sigma_1 + \left(\frac{k_1 q}{m} + \varepsilon_1 \right) \right) + \varepsilon_2 + \theta g$$

考虑到在 Rec 中计算 k_2 的公式，我们有

$$\begin{aligned} k_2 &= \lfloor m(v/g - \sigma_2/q) \rfloor \bmod m \\ &= \left\lfloor m \left(\frac{1}{q}(\sigma_1 + k_1 q/m + \varepsilon_1) + \frac{\varepsilon_2}{g} + \theta - \frac{\sigma_2}{q} \right) \right\rfloor \bmod m \\ &= \left\lfloor k_1 + \frac{m}{q}(\sigma_1 - \sigma_2) + \frac{m}{q}\varepsilon_1 + \frac{m}{g}\varepsilon_2 \right\rfloor \bmod m \end{aligned}$$

根据事实 3.1，存在 $\theta' \in \mathbb{Z}$ 和 $\delta \in [-d, d]$ 使 $\sigma_1 = \sigma_2 + \theta'q + \delta$ ，由于 $|m\delta/q + m\varepsilon_1/q + m\varepsilon_2/g| \leq md/q + m/2q + m/2g < 1/2$ ， $k_1 = k_2$ ，因此

$$k_2 = \left\lfloor k_1 + \frac{m}{q}\delta + \frac{m}{q}\varepsilon_1 + \frac{m}{g}\varepsilon_2 \right\rfloor \bmod m$$

定理 3.6 AKCN 算法满足安全性。具体来说，当变量之间满足 $\sigma_1 \leftarrow \mathbb{Z}_q$ ， k_1 和 v 是相互独立时，AKCN 算法是安全的。

证明 - 定理 3.6 对任意 $\tilde{v} \in \mathbb{Z}_g$ 和任意 $\tilde{k}_1, \tilde{k}'_1 \in \mathbb{Z}_m$ ，我们证明了当 $\sigma_1 \leftarrow \mathbb{Z}_q$ 时 $\Pr[v = \tilde{v} | k_1 = \tilde{k}_1] = \Pr[v = \tilde{v} | k_1 = \tilde{k}'_1]$ 。

对 $\mathbb{Z}_m \times \mathbb{Z}_g$ 中的任意 (\tilde{k}, \tilde{v}) ，事件 $(v = \tilde{v} | k_1 = \tilde{k})$ 等于存在 $\sigma_1 \leftarrow \mathbb{Z}_q$ 使 $\tilde{v} = \lfloor g(\sigma_1 + \lfloor \tilde{k}q/m \rfloor)/q \rfloor \bmod g$ 。注意 $\sigma_1 \leftarrow \mathbb{Z}_q$ 满足 $\tilde{v} = \lfloor g(\sigma_1 + \lfloor \tilde{k}q/m \rfloor)/q \rfloor \bmod g$ ，有且仅有存在 $\varepsilon \in (-1/2, 1/2]$ 且 $\theta \in \mathbb{Z}$ 时，可以使 $\tilde{v} = g(\sigma_1 + \lfloor \tilde{k}q/m \rfloor)/q + \varepsilon - \theta g$ 。也就是对于某些 $\varepsilon \in (-1/2, 1/2]$ ， $\sigma_1 = (q(\tilde{v} - \varepsilon)/g - \lfloor \tilde{k}q/m \rfloor) \bmod q$ 。令 $\Sigma(\tilde{v}, \tilde{k}) = \{\sigma_1 \in \mathbb{Z}_q | \exists \varepsilon \in (-1/2, 1/2] \text{ s.t. } \sigma_1 = (q(\tilde{v} - \varepsilon)/g - \lfloor \tilde{k}q/m \rfloor) \bmod q\}$ 。通过令 $\phi(x) = (x - \lfloor \tilde{k}q/m \rfloor) \bmod q$ ，定义映射 $\phi: \Sigma(\tilde{v}, 0) \rightarrow \Sigma(\tilde{v}, \tilde{k})$ 。那么， ϕ 显然是一个一一映射。因此， $\Sigma(\tilde{v}, \tilde{k})$ 的基数和 \tilde{k} 相关。具体来说，对于任意的 $\tilde{v} \in \mathbb{Z}_g$ 和任意的 $\tilde{k}_1, \tilde{k}'_1 \in \mathbb{Z}_m$ ，都有 $|\Sigma(\tilde{v}, \tilde{k}_1)| = |\Sigma(\tilde{v}, \tilde{k}'_1)| = |\Sigma(\tilde{v}, 0)|$ 。

现在，对于任意的 $\tilde{v} \in \mathbb{Z}_g$ 和任意的 $\tilde{k} \in \mathbb{Z}_m$ ，当 $\sigma_1 \leftarrow \mathbb{Z}_q$ 时，那么 $\Pr[v = \tilde{v} | k_1 = \tilde{k}] = \Pr[\sigma_1 \in \Sigma(\tilde{v}, \tilde{k}) | k_1 = \tilde{k}] = |\Sigma(\tilde{v}, \tilde{k})|/q = |\Sigma(\tilde{v}, 0)|/q$ 。等式右边只依赖于 \tilde{v} 的值，因此 \tilde{v} 和 k_1 是相互独立的。

另外，当算法中的参数满足 $q = g = 2^{\bar{q}}, m = 2^{\bar{m}}$ ，且 \bar{q}, \bar{m} 都是正整数时，因为计算将只涉及整数，我们可以直接消除算法 4 第 3 行 Con 中的两个舍入操作，从而将其简化为算法 5。注意，在算法 5 中，模块化和乘除法可以通过简单的位

运算来实现。

算法 5 AKCN power of 2

```

1: params :  $q = g = 2^{\bar{q}}, m = 2^{\bar{m}}, aux = \{G = q/m\}$ 
2: procedure CON( $\sigma_1, k_1, params$ )
3:    $v = (\sigma_1 + k_1 \cdot G) \bmod q$ , where  $k_1 \cdot G$  can be offline computed
4:   return  $v$ 
5: end procedure
6: procedure REC( $\sigma_2, v, params$ )
7:    $k_2 = \lfloor (v - \sigma_2)/G \rfloor \bmod m$ 
8:   return  $k_2$ 
9: end procedure

```

算法 6 AKCN simple

```

1: params =  $(q, m, g, d, aux)$ , where  $q = 2^{\bar{q}}, g = 2^{\bar{g}}, m = 2^{\bar{m}}$ , and  $q = gm$  (i.e.,  $\bar{g} + \bar{m} = \bar{q}$ )
2: procedure CON( $\sigma_1, k_1, params$ )  $\triangleright \sigma_1 \in [0, q-1]$ 
3:    $v = \lfloor (k_1 g + \sigma_1)/m \rfloor \bmod g$   $\triangleright k_1 g/m$  can be offline computed
4:   return  $v$ 
5: end procedure
6: procedure REC( $\sigma_2, v, params$ )  $\triangleright \sigma_2 \in [0, q-1]$ 
7:    $k_2 = \lfloor (mv - \sigma_2)/g \rfloor \bmod m$ 
8:   return  $k_2$ 
9: end procedure

```

对于算法 5 描述的协议变种，它的正确性和安全性可以通过对参数 (q, d, m) 的限制进行放松而得到证明，如下推论所示。

推论 3.3 如果 q 和 m 是 2 的次方数，并且 d, m, q 满足 $2md < q$ ，那么算法 5 描述的 AKCN-power-of-2 满足正确性和安全性。

证明-推论 3.3 对于正确性而言，假设 $|\sigma_1 - \sigma_2|_q \leq d$ ，那么存在一个 $\delta \in [-d, d]$ 和 $\theta \in \mathbb{Z}$ 使得 $\sigma_2 = \sigma_1 + \theta q + \delta$ 。由计算 v 的公式可知，存在一个 $\theta' \in \mathbb{Z}$ 使得 $v = \sigma_1 + k_1 2^{\bar{q}-\bar{m}} + \theta' q$ 。将这些代入到计算 k_2 的公式中，也就是算法 5 的第 7 行，可得，

$$\begin{aligned}
k_2 &= \lfloor (v - \sigma_1 - \delta - \theta q) / 2^{\bar{q}-\bar{m}} \rfloor \bmod m \\
&= \lfloor (k_1 2^{\bar{q}-\bar{m}} - \delta) / 2^{\bar{q}-\bar{m}} \rfloor \bmod m \\
&= (k_1 - \lfloor \delta / 2^{\bar{q}-\bar{m}} \rfloor) \bmod m
\end{aligned}$$

如果 $2md < q$ ，那么 $|\delta / 2^{\bar{q}-\bar{m}}| < 1/2$ ，因此 $k_1 = k_2$ 。

而对于安全性而言，算法 5 是算法 4 的一种特殊情况，因此安全性证明也可以直接根据算法 4 的安全证明推导而得到。

推论 3.4 如果 q, g, m 都是 2 的次方数且满足 $q = gm$ ，并且 d, m, q 满足 $m + 2d < g$ ，那么算法 6 描述的 AKCN-simple 满足正确性和安全性。

证明-推论 3.4 对于正确性而言，假设 $|\sigma_1 - \sigma_2|_q \leq d$ ，那么存在一个 $\delta \in [-d, d]$ 和 $\theta \in \mathbb{Z}$ 使得 $\sigma_2 = \sigma_1 + \theta q + \delta$ 。由计算 v 的公式可知，存在一个 $\theta' \in \mathbb{Z}$ 和 $\varepsilon \in (-1/2, 1/2]$ 使得 $v = \sigma_1 2^{-\bar{m}} + k_1 2^{\bar{g}-\bar{m}} + \varepsilon + \theta' g$ 。将这些代入到计算 k_2 的公式中，也就是算法 5 的第 7 行，可得，

$$k_2 = \lfloor k_1 + (m\varepsilon - \delta)/g \rfloor \bmod m$$

如果 $m + 2d < g$ ，那么 $\lfloor k_1 + (m\varepsilon - \delta)/g \rfloor < 1/2$ ，因此 $k_1 = k_2$ 。

算法 6 作为 AKCN 机制的一个特殊的情况，因此安全性也可以直接根据算法 4 的安全性证明推导而得到。

4 基于理想格 RLWE 的密钥协商方案

用 λ 表示安全参数， $q \geq 2$ 是正素数， σ 是离散高斯分布 $D_{\mathbb{Z}^n, \sigma, n}$ 的参数， n 表示 R_q 中多项式的度，Gen 表示从小的种子生成 $a \in R_q$ 的 PRG，那么系统参数是 $(\lambda, n, q, \sigma, KC)$ 。用 $KC = (\text{params}, \text{Con}, \text{Rec})$ 表示正确且安全的 KC 方案，其中 $\text{params} = (q, g, m, d)$ ，本节主要考虑 $m = 2$ 的情况。图 4.1 描述了从 RLWE 得到的基于 KC 的密钥协商协议，真实会话密钥是由 k_1 和 k_2 从密钥推导函数 KDF 导出的。基于 KC 的密钥协商协议可以比较容易地扩展成为 AKC 协议，并保证协议的正确性和安全性。具体转化过程参见我们提交的《AKCN-E8：基于理想格 RLWE 的通用和模块化的密钥封装机制》。

为简化协议描述，Con 和 Rec 函数适用于多项式，意味它们可以各自适用于每一个系数。为了简单和对称，在下面的分析中，我们用 $t = t_1 = t_2 \geq 0$ 表示在 y_1 和 y_2 中截去相同数目的比特。不失一般性，为优化公钥(或密文)的长度，可以设置 $t_1 > t_2$ (或 $t_1 < t_2$)。

参数和实现： 图 4.1 所示的协议可以应用于任何 RLWE 问题的实例化。但是如果 n 是 2 的幂，素数 q 满足 $q \bmod 2n = 1$ ，可以用数论变换(NTT)方法加速多

项式乘法以提升总体实现效率。同时使用蒙哥马利算法(Montgomery arithmetic)和 AVX2 指令集[2], 通过在 ARM 汇编[27,35]中仔细优化性能相关的关键程序(特别是 NTT), 可以使性能显著提升。在[2]中, 底层的噪声分布在二项分布 Ψ_η 的中心(而不是围绕具有标准差为 $\sigma = \sqrt{\eta/2}$ 的高斯分布)。二项分布中 η 的和独立中心化二项变量, 分布可以通过简单地对硬件或软件进行取样得到, 更好地抵抗时间攻击。我们强调真实的噪声分布是由 Ψ_η 和取决于 t 的截去比特合成。考虑到后量子安全层次, 我们经常假设 $t=0$ (例如, 在真实的噪声分布中不考虑 t 的影响); 有时我们也会考虑该值的影响, 观测到没有已知攻击可以利用不同噪声分布信息, 可以近似地把 $\sigma' = \sqrt{(2\sigma^2 + 2^{t-1})/2}$ 作为噪声的标准差。后量子安全的具体值可以通过运行

[2,13]提供的脚本获取。表 4.1 和 4.2 总结了 OKCN-RLWE 的参数和性能。

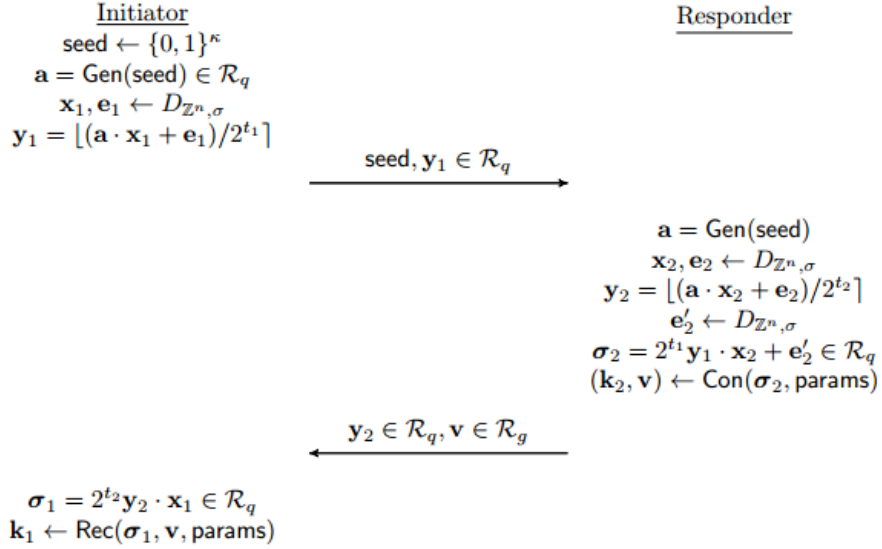


图 4.1: 从 KC 得到的 RLWE-based KE, $k_1, k_2 \in R_q$ 。使用算法 1 中所描述的 OKCN 实例化的协议称为 OKCN-RLWE

安全分析: 基于 RLWE 的密钥协商协议的安全定义和证明可以直接使用基于 LWE 和 LWR 的 KE 协议。[13]给出了相关分析。NewHope 达到了抗潜在格问题的 255 位后量子安全, 但是实际上使用的 256 位共享密钥本质上可能提供低级

别的安全保障(源自[32,29]的观点, 使用 Grover 搜索算法的平方加速和更加成熟的针对对称密码的量子攻击)。这种意义上, NewHope 的 255-bit 后量子安全实际上言过其实。对于 RLWE-based KE 协议, 我们力求达到可以同时抵抗潜在格问题和共享密钥的 256-bit 后量子安全。这意味着共享密钥的长度至少是 256 位。

误码率分析: 这里的误码率分析是一种在支持文档中第 9 节中描述的 MLWE-based 密钥协商协议的误码率分析的特殊情况。注意到 OKCN(或 ACKN)的正确性需要 $(2d + 1)m < q \left(1 - \frac{1}{g}\right)$ (或 $(2d + 1)m < q \left(1 - \frac{1}{g'}\right)$), 这意味着对应于相同参数 (q, m, d) , 如果 $g' = mg$, 使用参数 g 的 OKCN-RLWE 和使用参数 g' 的 AKCN-RLWE 具有相同的误码率。工作中, 我们设置 $m=2$, 具体的误码率可以通过运行[2,13]提供的脚本获取。

4.2 在不同位置错误的独立性

减少误码率的另外一种方法是使用纠错码(error correction code, ECC)。不幸的是, 在一般情况下, 纠错码技术比 NewHope 低效并且耗费资源。工作中, 我们有一个关于 RLWE-based KE 协议的关键发现: 当 n 足够大时, 共享密钥中在不同位置发生错误是独立的。基于这种发现, 我们提出一种超级简单快速的编码方案: SEC 编码, 至少可以修复一位的错误。通过在 OKCN/AKCN 中使用 SEC 编码, 我们提出了从 OKCN 和 AKCN 得到了迄今最简单的 RLWE-based 密钥协商, 该协议可用于 CCA 安全的公钥加密(例如: 在大约 250-bit 后量子安全中得到 3392 位带宽和 $2^{-73.2}$ 误码率的 765-bit 共享密钥)。

表 4.1: 表中所有方案使用 NewHope 提出的相同安全参数: $(q = 12289, n = 1024, m = 2^1, t = 0, \sigma = \sqrt{8}, k = 256, \Psi_{16})$ 。|K|代表共识位的二进制长度。bw.(B)代表带宽字节数。err.代表错误率。 n_H 代表 SEC 码使用的维度。per 代表在使用 SEC 码之前每位的错误率。err.代表全局错误率。pq-sec 代表针对底层格问题的已知最优抗量子攻击的安全强度。

	g	d	$ \mathbf{K} $	bw.(B)	per.	n_H	err.	pq-sec
OKCN-RLWE	2^4	2879	1024	4128	2^{-48}	-	2^{-38}	255
OKCN-RLWE	2^6	3023	1024	4384	2^{-52}	-	2^{-42}	255
AKCN-RLWE	2^4	2687	1024	4128	2^{-42}	-	2^{-32}	255
AKCN-RLWE	2^6	2975	1024	4384	2^{-51}	-	2^{-41}	255
OKCN-SEC	2^2	2303	765	3904	2^{-31}	4	$2^{-48.5}$	255
OKCN-SEC	2^3	2687	765	4032	2^{-42}	4	$2^{-70.5}$	255
OKCN-SEC	2^3	2687	837	4021	2^{-42}	5	$2^{-69.5}$	255
AKCN-SEC	2^4	2687	765	4128	2^{-42}	4	$2^{-70.5}$	255
AKCN-SEC	2^4	2687	837	4128	2^{-42}	5	$2^{-69.5}$	255
NewHope	2^2	-	256	3872	2^{-69}	-	2^{-61}	255
NewHope-Simple	2^2	-	256	4000	2^{-69}	-	2^{-61}	255
AKCN-4:1-RLWE	2^2	-	256	3904	2^{-69}	-	2^{-61}	255

由表 4.1 可以看出，（增加说明：目的是说明本算法的优势和价值所在！）。

假设 $f(x)$, $g(x)$ 是 n 阶的两个多项式，其系数独立于高斯分布。设 $h(x) = f(x) \cdot g(x) \in \mathbb{R}[x]/(x^n + 1)$ 。我们证明对于每两个不同的整数 $0 \leq c_1, c_2 < n$ ，当 n 趋于有限时， $(h[c_1], h[c_2])$ 的联合分布将接近二维高斯。因此，对于图 4.1 所示的 KC 和 AKC 基于 RLWE 的密钥协商的基本构造，可以合理地假设任意两个不同位置的错误率在 n 足够大时是独立的。

为了便于表示，对于任意多项式 f ，在下文中，令 $f[i]$ 表示 x_i 的系数。

表 4.2: 参数: $k = 256, q = 12289, n = 1024, m = 2, n_H = 4$ 。 $|\mathbf{K}|(\text{SEC})$ 代表密钥长度(或者使用 SEC 的密钥长度); $\text{bw.}(pk, cipher)$ 代表带宽字节数(包括 $pk=(y_1, seed)$ 和 $cipher=(y_2, v)$ 的长度); $\text{err.}(\text{SEC})$ 代表错误率(使用 SEC 的错误率); pq-sec (或 t-sec) 代表针对不考虑 t (或者试探性地把噪声标准差看做 $\sigma' = \sqrt{(2\sigma^2 + 2^{t-1})/2}$) 时底层格问题的已知最优后量子攻击的安全强度。

	g	t	σ (σ')	$ \mathbf{K} (\text{SEC})$	$\text{bw.}(pk, cipher)$	$\text{err.}(\text{SEC})$	pq-sec (t-sec)
OKCN-RLWE $\sigma = \sqrt{8}$	2^4	2	$\sqrt{8}$ ($\sqrt{9}$)	1024(765)	3392 (1440,1952)	$2^{-28.1}$ (2^{-61})	255 (258)
	2^3	2	$\sqrt{8}$ ($\sqrt{9}$)	1024(765)	3264 (1440,1824)	$2^{-24.8}$ ($2^{-54.4}$)	255 (258)
	2^3	1	$\sqrt{8}$ ($\sqrt{8.5}$)	1024(765)	3520 (1568,1952)	$2^{-33.4}$ ($2^{-71.6}$)	255 (257)
	2^4	1	$\sqrt{8}$ ($\sqrt{8.5}$)	1024(765)	3648 (1568,2080)	$2^{-37.8}$ ($2^{-80.4}$)	255 (257)
OKCN-RLWE $\sigma = \sqrt{6}$	2^2	2	$\sqrt{6}$ ($\sqrt{7}$)	1024(765)	3136 (1440,1696)	$2^{-31.8}$ ($2^{-68.4}$)	246 (250)
	2^3	2	$\sqrt{6}$ ($\sqrt{7}$)	1024(765)	3264 (1440,1824)	$2^{-43.2}$ ($2^{-91.2}$)	246 (250)
	2^4	2	$\sqrt{6}$ ($\sqrt{7}$)	1024(765)	3392 (1440,1952)	2^{-49} ($2^{-102.8}$)	246 (250)
	2^3	1	$\sqrt{6}$ ($\sqrt{6.5}$)	1024(765)	3520 (1568,1952)	$2^{-60.6}$ (2^{-126})	246 (248)
	2^4	1	$\sqrt{6}$ ($\sqrt{6.5}$)	1024(765)	3648 (1568,2080)	$2^{-68.9}$ ($2^{-142.6}$)	246 (248)

引理 4.1 假设 $f(x)$, $g(x) \in \mathbb{R}[x]/(x^n + 1)$ 是两个 n 阶多项式，其系数独立于 $\mathcal{N}(0, \sigma^2)$ 。设 $h(x) = f(x) \cdot g(x) \in \mathbb{R}[x]/(x^n + 1)$ ，其中 $h(x)$ 表示为 n 阶多项式。对于任意两个不同的整数 $0 \leq c_1, c_2 < n$ ，二维随机向量 $(h[c_1], h[c_2]) \in \mathbb{R}^2$ 的特征函数如下：

$$\phi_{c_1, c_2}(t_1, t_2) = \mathbb{E} \left[e^{i(t_1 h[c_1] + t_2 h[c_2])} \right] = t_1 \mathbf{f}^T \mathbf{A}_{c_1} \mathbf{g} + t_2 \mathbf{f}^T \mathbf{A}_{c_2} \mathbf{g} \quad (6)$$

$$= \prod_{k=0}^{n-1} \left(1 + \sigma^4 \left(t_1^2 + t_2^2 + 2t_1 t_2 \cos \left(\pi(c_1 - c_2) \frac{2k+1}{n} \right) \right) \right)^{-\frac{1}{2}} \quad (7)$$

证明。可以看出 $t_1 h[c_1] + t_2 h[c_2]$ 等于

$$\begin{aligned} & t_1 \left(\sum_{i+j=c_1} f[i]g[j] - \sum_{i+j=c_1+n} f[i]g[j] \right) + t_2 \left(\sum_{i+j=c_2} f[i]g[j] - \sum_{i+j=c_2+n} f[i]g[j] \right) \\ &= t_1 \mathbf{f}^T \mathbf{A}_{c_1} \mathbf{g} + t_2 \mathbf{f}^T \mathbf{A}_{c_2} \mathbf{g} = \mathbf{f}^T (t_1 \mathbf{A}_{c_1} + t_2 \mathbf{A}_{c_2}) \mathbf{g} \end{aligned}$$

其中 $\mathbf{f} = (f[0], f[1], \dots, f[n-1])^T, \mathbf{g} = (g[0], g[1], \dots, g[n-1])^T$ ，符号

$\mathbf{A}_{c_1}, \mathbf{A}_{c_2}$ 定义为：

$$\mathbf{A}_c = \begin{pmatrix} & & & 1 & & \\ & & \ddots & & & \\ & 1 & & & & \\ & & & & & -1 \\ & & & & \ddots & \\ & & -1 & & & \end{pmatrix}$$

第一行中的 1 在第 c 列中，最后一行的 -1 在第 $c+1$ 列中。

因为 $t_1 \mathbf{A}_{c_1} + t_2 \mathbf{A}_{c_2}$ 是对称的，它可以被正交对角化为 $\mathbf{P}^T \mathbf{\Lambda} \mathbf{P}$ ，其中 \mathbf{P} 是正交矩阵， $\mathbf{\Lambda}$ 是对角矩阵。因此， $\phi_{c_1, c_2}(t_1, t_2) = \mathbb{E}[\exp(i(\mathbf{P}\mathbf{f})^T \mathbf{\Lambda} (\mathbf{P}\mathbf{g}))]$ 。因为 \mathbf{P} 是正交的，它保持了均匀分布不变。因此， $(\mathbf{P}\mathbf{f})^T \mathbf{\Lambda} (\mathbf{P}\mathbf{g})$ 等于两个独立的一维高斯的 n 个标度的乘积之和。

假设 $\lambda_1, \lambda_2, \dots, \lambda_n$ 是 $t_1 \mathbf{A}_{c_1} + t_2 \mathbf{A}_{c_2}$ ，且 ϕ 是两个独立的一维标准高斯的乘积的特征函数。则我们得到等式（8）

$$\phi_{c_1, c_2}(t_1, t_2) = \prod_{k=0}^{n-1} \phi(\sigma^2 \lambda_k) \quad \dots\dots\dots (8)$$

从[48]， $\phi(t) = (1 + t^2)^{-1/2}$ 。对于 λ_k ，我们可进一步得到

$$\begin{aligned} (t_1 \mathbf{A}_{c_1} + t_2 \mathbf{A}_{c_2})^2 &= (t_1^2 + t_2^2) \mathbf{I} + t_1 t_2 (\mathbf{A}_{c_1} \mathbf{A}_{c_2} + \mathbf{A}_{c_2} \mathbf{A}_{c_1}) \\ &= (t_1^2 + t_2^2) \mathbf{I} + t_1 t_2 (\mathbf{G}^{c_2 - c_1} + \mathbf{G}^{c_1 - c_2}), \end{aligned}$$

其中：

$$\mathbf{G} = \begin{pmatrix} & & & 1 \\ & & 1 & \\ & & & \ddots \\ & & & & 1 \\ -1 & & & & \end{pmatrix}$$

\mathbf{G} 的特征多项式是 $x^n + 1$ 。因此， λ_k 满足

$$\lambda_k^2 = t_1^2 + t_2^2 + 2t_1t_2 \cos\left(\pi(c_1 - c_2)\frac{2k+1}{n}\right)$$

将上式代入等式 8，我们得到了等式 7。

对于任意固定整数 $0 \leq c_1, c_2 < n$, $c_1 \neq c_2$, 当 n 趋于无穷时, $\left(\frac{h[c_1]}{\sigma^2\sqrt{n}}, \frac{h[c_2]}{\sigma^2\sqrt{n}}\right)$ 的分布收敛到二维正态分布 $\mathcal{N}(0, \mathbf{I}_2)$ 。

证明 定理 4.1. 用 $\phi(t_1, t_2)$ 表示随机向量 $\left(\frac{h[c_1]}{\sigma^2\sqrt{n}}, \frac{h[c_2]}{\sigma^2\sqrt{n}}\right)$ 的特征函数。然后, 对于固定的 t_1, t_2 ,

$$\ln(\phi(t_1, t_2)) = -\frac{1}{2} \sum_{k=0}^{n-1} \ln\left(1 + \frac{1}{n} \left(t_1^2 + t_2^2 + 2t_1t_2 \cos\left(\pi(c_1 - c_2)\frac{2k+1}{n}\right)\right)\right) \quad (9)$$

$$= -\frac{1}{2} \sum_{k=0}^{n-1} \left[\frac{1}{n} \left(t_1^2 + t_2^2 + 2t_1t_2 \cos\left(\pi(c_1 - c_2)\frac{2k+1}{n}\right)\right) + r_k \right] \quad (10)$$

$$= -\frac{1}{2} (t_1^2 + t_2^2) - \frac{1}{2} \sum_{k=0}^{n-1} r_k, \quad (11)$$

其中 r_k 是拉格朗日余项。所以 $|r_k| \leq \lambda_k^4/2n^2$ 。因为 $\lambda_k^2 \leq (|t_1| + |t_2|)^2$, 我们能得出 $|r_k| \leq (|t_1| + |t_2|)^4/2n^2$ 。

当 n 趋向于无穷时, $\phi(t_1, t_2)$ 逐点收敛于 $\exp(-(t_1^2 + t_2^2)/2)$, 这是二维正态分布 $\mathcal{N}(0, \mathbf{I}_2)$ 的特征函数。由 Levy 的收敛理论, 我们可以得出随机向量 $\left(\frac{h[c_1]}{\sigma^2\sqrt{n}}, \frac{h[c_2]}{\sigma^2\sqrt{n}}\right)$ 在分布上收敛于正态分布 $\mathcal{N}(0, \mathbf{I}_2)$ 。

说明：最近, 在文献[50]中, 作者通过实验表明对于基于 RLWE 的密钥协商协议会话密钥在具体位置发生错误的概率不独立。特别地, 对 LAC 算法, 由于 LAC 使用很小的 $q=251$ 且使用强大的 BCH 纠错码, 文献[50]表明其会话密码不同位置的错误率不独立。在 $n=512$ 和 $n=1024$ 时错误率不独立。但是, 文献[10]

的结果并不和我们的结果矛盾。比如，对于 LAC 算法， $n=512$ 的错误率误差和 $n=1024$ 的错误率误差快速收敛。这和我们的证明，当 n 趋向无穷时，会话密钥不同位置错误率独立，实际上一致的。特别地，对于本提案而言，由于我们使用了一个足够大的 q 但使用了纠错能力非常弱的 SEC 码，文献[50]的结果对本提案工作的影响很小。

4.3 利用单比特纠错码降低错误率

注意，对于图 4.1 中展示的由 KC 得到的基于 RLWE 密钥协商的基础协议构造来说，每个比特的错误率已经能够达到 2^{-42} 。从这里我们能看到，根据定理 4.1，当 n 很大时，出现在不同位置的错误的具有独立性，如果我们能够纠正单个比特的错误，那么错误率将会显著降低。为了实现这一目标，我们提出了汉明码的一个变体，叫做单比特纠错码 (SEC)，它能以一种简单且快速的方式纠正单个比特的错误。

4.3.1 单比特纠错码

在本节中所有的算术运算都在 \mathbb{Z}_2 群上进行，对于一个正整数 n_H ， $N_H = 2^{n_H}$ ，并且按如下定义矩阵 H ，其中对于任意 i ， $1 \leq i \leq N_H - 1$ ， H 的第 i 列对应于 i 的二进制表示。

$$\mathbf{H}_{n_H \times (N_H - 1)} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & \cdots & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & \cdots & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 & 1 \\ \vdots & & & & & & & \ddots & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & 1 \end{pmatrix}$$

对于任意 $\mathbf{x} = (x_1, \dots, x_{N_H - 1}) \in \mathbb{Z}_2^{N_H - 1}$ ，让 $\mathbf{p}^T = \mathbf{H}\mathbf{x}^T$ 。容易验证， \mathbf{p} 的第 j 个元素是所有满足 i 的倒数第 j 个比特是 1 的 x_i 的异或，其中 $1 \leq j \leq n_H$ 并且 $1 \leq i \leq N_H - 1$ 。具体来说， \mathbf{p} 的第一个元素是所有满足 i 的最低有效位是 1 的 x_i 的异或， \mathbf{p} 的第二个元素是所有满足 i 的倒数第二有效位是 1 的 x_i 的异或，以此类推。表示为 $\mathbf{p} = (p_1, p_1, \dots, p_{n_H})$ 。我们可以将 \mathbf{p} 的所有比特结合成一个二进制数 $\bar{p} = 2^0 p_1 + 2^1 p_2 + \dots + 2^{n_H - 1} p_{n_H}$ 。从 H 的构造可以直接引出下面的命题。

命题 6.1. 如果 $\mathbf{p}^T = \mathbf{H}\mathbf{x}^T$, 且 \mathbf{x} 的汉明距离为 1, 那么 $\bar{\mathbf{p}}$ 是 \mathbf{x} 中唯一一个 1 的下标。

单比特纠错码 \mathcal{C} 定义为

$$\mathcal{C} = \left\{ (x_0, \mathbf{x}, \mathbf{p}) \in \mathbb{Z}_2 \times \mathbb{Z}_2^{N_H-1} \times \mathbb{Z}_2^{n_H} \mid x_0 = \bigoplus_{i=1}^{N_H-1} x_i, \mathbf{p}^T = \mathbf{H}\mathbf{x}^T \right\}$$

编码算法在算法 14 中给出了直接的描述。

Algorithm 14 $\text{Encode}_{\mathcal{C}}(\mathbf{x} = (x_1, \dots, x_{N_H-1}))$

```

1:  $x_0 = \bigoplus_{i=1}^{N_H-1} x_i$ 
2:  $\mathbf{p}^T = \mathbf{H}\mathbf{x}^T$ 
3:  $\mathbf{c} = (x_0, \mathbf{x}, \mathbf{p})$ 
4: return  $\mathbf{c}$ 

```

Algorithm 15 $\text{Decode}_{\mathcal{C}}(x_0, \mathbf{x}, \mathbf{p})$

```

1:  $p = \bigoplus_{i=0}^{N_H-1} x_i$ 
2: if  $p = 1$  then
3:    $i = \overline{\mathbf{H}\mathbf{x}^T \oplus \mathbf{p}}$      $\triangleright$  bitwise exclusive-or
4:    $x_i = x_i \oplus 1$ 
5: end if
6: return  $\mathbf{x} = (x_1, \dots, x_{N_H-1})$ 

```

我们现在展示 \mathcal{C} 可以纠正单个比特。假设 \mathbf{x} 被编码进 $\mathbf{c} = (x_0, \mathbf{x}, \mathbf{p})$ 。由于一些原因, 比如通信信道的噪声, 消息 \mathbf{c} 可能变成 $\mathbf{c}' = (x_0', \mathbf{x}', \mathbf{p}')$ 。我们只需要考虑最多只有一个比特出现错误的情况。如果 x_0' 等于奇偶校验比特 \mathbf{x}' , 那么 x_0 和 \mathbf{x} 没有发生错误。否则, 在 x_0' 或 \mathbf{x}' 中有一个比特的错误, 但是 $\mathbf{p}' = \mathbf{p}$ (正如我们所假设的存在最多一个比特的错误已经出现在 x_0' 或 \mathbf{x}' 中)。我们计算 $\mathbf{p}'' = \mathbf{H}\mathbf{x}'^T \oplus \mathbf{p}^T$ 。事实上, $\mathbf{p}'' = \mathbf{H}\mathbf{x}'^T \oplus \mathbf{p}^T = \mathbf{H}(\mathbf{x}'^T \oplus \mathbf{x}^T)$ 。如果 \mathbf{x}' 出现了单比特错误, 根据命题 8.1, $\bar{\mathbf{p}}''$ 是出错比特的下标。如果 x_0' 出现了单比特错误, 那么 $\mathbf{x}' = \mathbf{x}$, 并且 $\mathbf{p}'' = \mathbf{H}\mathbf{0} = \mathbf{0}$, 因此, $\bar{\mathbf{p}}''$ 总是等于出错比特的下标。

解码算法在算法 15 当中描述, 注意, 根据 \mathbf{H} 的特殊形式, 矩阵乘法 $\mathbf{H}\mathbf{x}^T$ 在编码和解码时都能通过像比特移动和逐个比特异或这样的单比特的运算 (这样的实现在附录 I 当中给出) 来完成。另外, 对于 AKCN-SEC 和 OKCN-SEC, 在算法 15 的 2-4 行中的计算只以 2^{-40} 的概率被执行, 所以解码算法会非常快速。

4.3.2 带有 SEC 码的 KC 算法

图 4.3 描述了采用了 SEC 码的 AKC 方案。注意 $\text{Encode}_{\mathcal{C}}$ 可以离线计算。

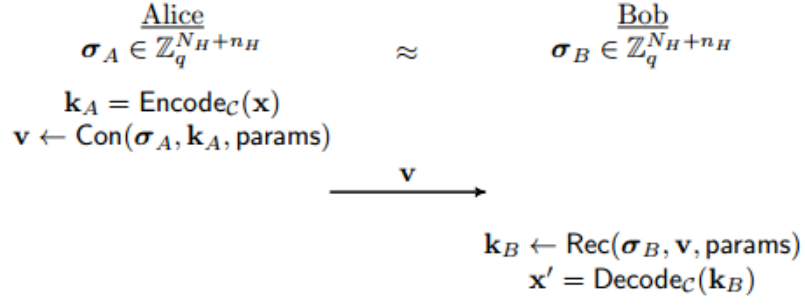


图 4.2: 带有 SEC 码的 AKC 方案

图中, $\mathbf{k}_A, \mathbf{k}_B \in \mathbb{Z}_2^{N_H+n_H}$, $|\mathbf{x}| = |\mathbf{x}'| = N_H - 1$ 。如果 \mathbf{k}_A 和 \mathbf{k}_B 的汉明距离至多为 1, 则 $\mathbf{x}' = \mathbf{x}$ 。

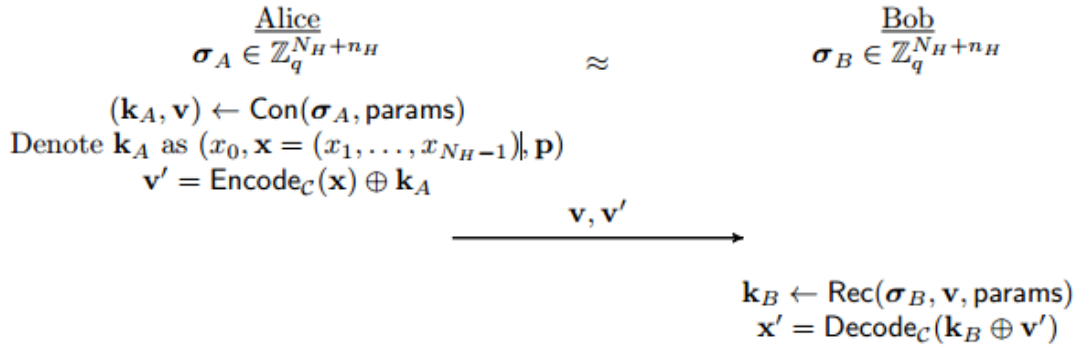


图 4.3: 应用 SEC 码到 KC 算法

图中, $\mathbf{k}_A, \mathbf{k}_B \in \mathbb{Z}_2^{N_H+n_H}$ 。如果 \mathbf{k}_A 和 \mathbf{k}_B 至多有一个比特不同, 则 $\mathbf{x}' = \mathbf{x}$ 。

对于采用了 SEC 码的 KC, 我们提出的算法在图 4.3 中描述。注意 Alice 只需要发送 \mathbf{v}' 的 $n_H + 1$ 比特, 因为从 \mathbf{v}' 的第 2 个元素到第 N_H 个元素全为 0。Bob 计算 $\mathbf{x}' = \text{Decode}_C(\mathbf{k}_B \oplus \mathbf{v}')$, 事实上, $\mathbf{k}_B \oplus \mathbf{v}' = \text{Encode}_C(\mathbf{x}) \oplus (\mathbf{k}_A \oplus \mathbf{k}_B)$ 。因此, 如果 \mathbf{k}_A 和 \mathbf{k}_B 的汉明距离为 1, 那么 $\mathbf{x}' = \mathbf{x}$ 。为了证明图 4.4 中算法的安全性, 我们需要下面的定理 4.2。

定理 4.2. 令 $\mathcal{V} = \mathbb{Z}_2 \times \{0 \in \mathbb{Z}_2^{N_H-1}\} \times \mathbb{Z}_2^{n_H}$, 则 $\mathbb{Z}_2^{N_H+n_H} = \mathcal{C} \oplus \mathcal{V}$, 其中 \oplus 表示直接相加。

证明. 对于任意 $\mathbf{k}_A = (x_0, \mathbf{x} = (x_1, \dots, x_{N_H-1}), \mathbf{p}) \in \mathbb{Z}_2^{N_H+n_H}$, 令 $\mathbf{c} = \text{Encode}_C(\mathbf{x})$ 并且 $\mathbf{v}' = \mathbf{c} \oplus \mathbf{k}_A$ 。我们可以得到 $\mathbf{k}_A = \mathbf{c} \oplus \mathbf{v}'$, 其中 $\mathbf{c} \in \mathcal{C}$, $\mathbf{v}' \in \mathcal{V}$ 。

接着, 我们证明 $\mathcal{V} \cap \mathcal{C} = \mathbf{0}$ 。如果 $\mathbf{k} = (x_0, \mathbf{x}, \mathbf{p}) \in \mathcal{V} \cap \mathcal{C}$, 那么 $\mathbf{x} = \mathbf{0}$, 这意味着 $x_0 = 0$ 且 $\mathbf{p}^T = \mathbf{H}\mathbf{0} = \mathbf{0}$ 。因此, $\mathbf{k} = \mathbf{0}$ 。

当 \mathbf{k}_A 服从均匀分布, 则根据定理 4.2, 在分解 $\mathbf{k}_A = \mathbf{c} \oplus \mathbf{v}'$, 其中 $\mathbf{c} \in \mathcal{C}$, $\mathbf{v}' \in \mathcal{V}$ 之后, \mathbf{c} 和 \mathbf{v}' 分别在 \mathcal{C} 和 \mathcal{V} 上服从均匀分布。并且 \mathbf{c} 和 \mathbf{v}' 是独立的。因为 $\mathbb{Z}_2^{N_H-1} \rightarrow \mathcal{C}$ 和 $\mathbf{x} \mapsto \text{Encode}_{\mathcal{C}}(\mathbf{x})$ 都是一一对应的, 我们得到 \mathbf{x} 和 \mathbf{v}' 是独立的, 并且 \mathbf{x} 均匀分布的。

我们在表 4.1 和表 4.2 中给出了 OKCN-SEC 和 AKCN-SEC 的参数以及性能。

4.3.3 在公钥密码设定中 OKCN-SEC 的 KEM 规范

Algorithm 19 $(\text{pk}, \text{sk}) \leftarrow \text{KeyGen}()$

```

1: seed  $\leftarrow \{0, 1\}^\kappa$ 
2:  $\mathbf{a} := \text{Gen}(\text{seed})$ 
3:  $\mathbf{x}_1, \mathbf{e}_1 \leftarrow D_{\mathbb{Z}^n, \sigma}$ 
4:  $\mathbf{y}_1 := \lfloor (\mathbf{a}\mathbf{x}_1 + \mathbf{e}_1) / 2^{t_1} \rfloor$ 
5: return  $(\text{pk} := (\text{seed}, \mathbf{y}_1), \text{sk} := \mathbf{x}_1)$ 

```

Algorithm 20 $(\text{ct}, \text{key}) \leftarrow \text{Encaps}(\text{pk})$

```

1:  $\mathbf{x}_2, \mathbf{e}_2, \mathbf{e}'_2 \leftarrow D_{\mathbb{Z}^n, \sigma}$ 
2:  $\mathbf{a} := \text{Gen}(\text{seed})$ 
3:  $\mathbf{y}_2 := \lfloor (\mathbf{a}\mathbf{x}_2 + \mathbf{e}_2) / 2^{t_2} \rfloor$ 
4:  $\sigma_2 := 2^{t_1} \mathbf{y}_1 \cdot \mathbf{x}_2 + \mathbf{e}'_2$ 
5:  $(\mathbf{k}_2, \mathbf{v}) \leftarrow \text{Con}(\sigma_2, \text{params})$ 
6: parse the vector  $\mathbf{k}_2$  into  $\Delta := \lfloor n / (N_H + n_H) \rfloor$  blocks, say  $\mathbf{k}_2^{(1)}, \dots, \mathbf{k}_2^{(\Delta)}$ , each of size  $N_H + n_H$ 
7: parse every  $\mathbf{k}_2^{(i)}$  into the form

```

$$\left(x_0^{(i)} \in \mathbb{Z}_2, \mathbf{x}^{(i)} \in \mathbb{Z}_2^{(N_H-1)}, \mathbf{p}^{(i)} \in \mathbb{Z}_2^{n_H} \right)$$

```

8:  $\mathbf{v}' := \left( \text{Encode}_{\mathcal{C}}(\mathbf{x}^{(i)}) \oplus \mathbf{k}_1^{(i)} \right)_{i \in [\Delta]}$ 
9: return  $(\text{ct} := (\mathbf{y}_2, \mathbf{v}, \mathbf{v}'), \text{key} := (\mathbf{x}^{(i)})_{i \in [\Delta]})$ 

```

Algorithm 21 $\text{key}' \leftarrow \text{Decaps}(\text{sk}, \text{ct})$

```

1:  $\sigma_1 := 2^{t_2} \mathbf{y}_2 \cdot \mathbf{x}_1$ 
2:  $\mathbf{k}_1 := \text{Rec}(\sigma_1, \mathbf{v}, \text{params})$ 
3: parse the vector  $\mathbf{k}_1$  into  $\Delta := \lfloor n / (N_H + n_H) \rfloor$  blocks, say  $\mathbf{k}_1^{(1)}, \dots, \mathbf{k}_1^{(\Delta)}$ , each of size  $N_H + n_H$ 
4: parse the vector  $\mathbf{v}'$  into  $\Delta$  blocks, say  $\mathbf{v}'_1, \dots, \mathbf{v}'_{\Delta}$ , each of size  $N_H + n_H$ 
5: return  $\text{key}' := \left( \text{Decode}_{\mathcal{C}}(\mathbf{k}_2^{(i)} \oplus \mathbf{v}'_i) \right)_{i \in [\Delta]}$ 

```

注: 上述伪代码 (算法 19, 20, 21) 描述了 OKCN-SEC 的 KEM 算法的一般工作原理。注意 $n / (N_H + n_H)$ 在实践中可能不是正整数; 特别是在我们的软件实现中, 它不是正整数。在这种情况下, $\mathbf{v}, \sigma_1, \sigma_2, \mathbf{k}_1, \mathbf{k}_2$ 中的一些系数不会有助于生成共享密钥。

- OKCN / AKCN-SEC 和 OKCN / AKCN-E8 比 NewHope 更通用，更灵活，允许在参数和性能之间进行更有用的权衡。

- OKCN / AKCN-SEC 与 OKCN / AKCN-E8。 OKCN / AKCN-SEC 具有更大的密钥大小并更简单。相比之下，在系统参数上，OKCN / AKCN-E8 同时具有更低的错误率，更小的带宽和更强的安全性，但是以更复杂的实现为代价。从系统简洁性和易于实现的观点看，我们更倾向于 OKCN / AKCN-SEC。

4.6 扩展到基于 RLWR (Ring-LWR) 的 KE

作为支持文档中第五部分中展示的基于 LWR 的 KE 的一个直接的拓展，基于 RLWR 的 KE 协议在图 4.4 中描述。简单起见，我们假设 p 和 q 是 2 的指数，而且 $p|q$ 。也可以实现 SEC 和 E8 格编码以进一步的降低错误概率。

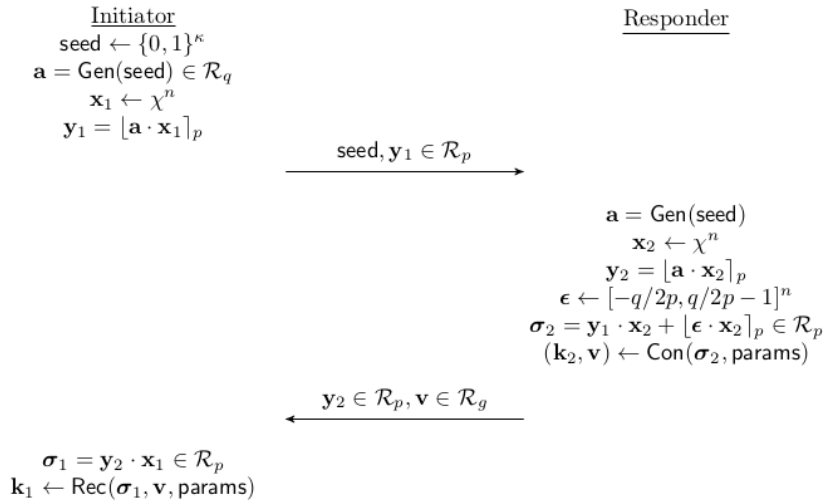


图 4.4 从 KC 得到的 RLWR-based 密钥协商

5 优缺点声明

5.1 基于 KC 密钥协商优缺点的一般性讨论

在我们提交的《AKCN-E8：基于理想格 RLWE 的通用和模块化的密钥封装机制》中，我们还给出了基于 AKCN 的密码机制。基于 OKCN 和 AKCN 的密码系统在不同的环境下会有不同的性能，这里我们针对他们的优缺点进行讨论：

- 基于 KC 的密钥协商对应于格世界中的 Diffie-Hellman 密钥协商，基于 AKC 的密钥封装对应于 EL Gamal 密钥封装；
- 当在实际中部署基于 AKC 的密钥封装时，如果响应者使用的随机性不够随机，那么就会完全破坏整个会话密钥的安全性。与之相比，在基于 KC 的密钥协商中，双方在整个协商的过程中扮演了同等重要的角色，那么由一方的随机性不够引起的问题可以得到解决。并且对称性通常也是密码学方案中一个必备的特点。
- 在某些应用中，比如之前曝光的 TLS 实现中，为了节省资源和管理简便，有些没有经验的用户会在多个会话中设置相同的会话密钥。显然，基于 AKC 的密钥封装不能有效避免这种情况，而对称的基于 KC 的密钥协商可以有效应对这种情况。
- 对于相同的参数 (q, m, g) （意味着相同的带宽），基于 OKCN 的密钥协商比基于 AKCN 的方案具有更低的错误率。这个比较我们在后面做详细介绍。
- 基于 KC 的密钥协商功能更加丰富，一方面它可以直接应用于密钥封装协议中或者 CPA 安全的 PKE 机制中；另一方面特定版本的 AKCN 还可以用于构造基于格密码的签名方案；
- 基于 KC 的密钥协商更加适用于目前的很多基于 Diffie-Hellman 的标准例如 IKE 和 TLS。我们注意到在 TLS1.3 中密钥封装已经明确被弃用了。
- 对于共享密钥为 256 比特的密钥协商，OKCN/AKCN-MLWE 是最高效的。但是对于共享密钥为 512 比特或者更大时，OKCN/AKCN-RLWE 更

加高效。

5.2 OKCN/AKCN-SEC、OKCN/AKCN-E8 与 NewHope 比较

和 NewHope 比较, OKCN/AKCN-SEC 和 OKCN/AKCN-E8 具有更大优势, 主要原因如下:

- 据我们所知, OKCN/AKCN-SEC 方案是最简单的基于 RLWE 的 KE 方案, 其具有错误概率在实践中可以视为可忽略, 相比于编码和解码四维格 \tilde{D}_4 , 更适用于硬件或软件实现。注意到 SEC 可以通过简单的位操作实现。此外, 以约为 1-2-40 的概率, 解码仅需涉及算法 14 第一行的 XOR 操作, 这极为简单和快速。

- AKCN-SEC 可以直接转换为一个 CPA 安全的加密 837 比特消息的 PKE 方案, 而 AKCN4:1-RLWE 和 NewHope-simple 用于加密 256 比特消息。

- 另一个优势来源于 OKCN/AKCN-SEC 和 OKCN/AKCN-E8 更有利于建立拥有直接共享性质或能够传输较大的密钥的 KE 协议。一方面, 人们普遍认为, 在后量子时代, 由于观察到通过 Grover 的搜索算法的二次加速, 以及更复杂的针对对称密钥加密的量子攻击的可能性[32, 29]。像 AES 这样的对称加密原语需要长度更长的密钥。实际上, 据我们所知, NewHope 的后量子安全性评估是作为独立协议进行的, 没有考虑可能的针对对称加密算法阶段共享密钥使用的量子攻击。另一方面, 在一些比公共商业用途更重要的应用领域, 更大的密钥长度实际上现在已经被强制要求执行了。注意到对于 NewHope, AKCN4:1-RLWE, 和 NewHope-simple, 如果想要得到一个 512 位的共享密钥 (以确保 256-位后量子安全性) 他们必须使用度为 2048 的多项式, 效率明显降低。

- 如上所述, SEC 方法仅在某些块中存在多个位错误时才会失败, 这使其在

某种意义上更加具有通用性：块大小 n_H 越小（相应地，越大），即错误概率（相应地，带宽扩展）将会更低。

5.3 算法实现代码及性能测试

算法实现代码在“参考实现文件夹”下。具体来说我们的软件实现如下。

1. 针对基于理想格的 OKCN-SEC 密钥封装算法给出了软件实现。

另外我们还在 测试实例文件夹下提供了测试报告作为本项目的支持文档。详细给出了程序的安装说明、测试环境、详细的测试用例、测试结果和基于测试结果的性能分析。

我们同时在“参考实现”文件夹下针对以上 三个部分算法分别给出了算法在 Linux 和 Windows 操作系统下的实现，这两份在不同操作系统下的代码在核心算法的实现上基本相同。但是，实验结果表明，两份基本相同的代码在两个不同的操作系统上的运行效率（无论是从运行时间，还是从时钟周期上）相差较大。我们猜测，这可能与以下因素有关：

1. 编译器的优化。对于我们的代码而言，GCC 的优化深度应该比 Visual Studio 要深。
2. 操作系统的调度问题。以 Ubuntu 系统为例，它的进程调度算法比 Windows 系统要更为出色。
3. 文件系统的管理。例如，Demand paging 和 zswap 技术的使用，都可以有效降低 I/O 次数。

一般情况下基于格密码的算法都是在 Linux 环境下进行实现，主要是因为 Linux 环境下有更好的随机库以及更高的计算性能，并且在安装编译上也更加简洁，所以我们的算法最开始只提供 Linux 下的实现，但是由于本次国家密码算法竞赛要求的测试环境为 Windows，所以我们同时提供了 Windows 版本的实现，但是从测试结果来看，Windows 下的性能的确是相对较差，更详细的测试数据以及测试环境，安装步骤，以及接口介绍参见“测试实例”文件夹下的测试报告（针对 Linux 和 Windows 实现 我们分别给出了测试报告）。

6.1. 基于理想格的 OKCN-SEC 性能分析

下面针对我们软件实现的基于理想格的 OKCN-SEC 给出性能分析总结，首先分析空间消耗，然后分析时间与时钟周期消耗。具体程序测试环境、测试详细数据、测试过程、测试用例参见测试报告。

6.1.1 空间消耗

下面我们将针对 OKCN-SEC-RLWE 算法实现给出空间性能分析。

OKCN-SEC-RLWE 算法的空间消耗

在表 6.1 中我们给出了 OKCN-SEC-RLWE 算法主要参数一览。

表 6.1 OKCN-SEC-RLWE 算法主要参数所耗空间一览表

名称	公钥 pk	私钥 sk	密文 ct	共享秘密 ss
长度 (byte)	1696	640	1955	95

对于具体 API 接口，针对大赛要求实现的 kem_api，一次密钥封装完整过程包括密钥生成函数(kem_keygen)，封装函数(kem_enc)和解封装函数(kem_dec)。表 6.2 给出了这三个函数输入输出参数所需要的空间 并对各个参数进行了简要的介绍。

表 6.2 OKCN-SEC-RLWE 算法各个接口参数空间性能

函数	输入	长度 (byte)	含义	输出	长度 (byte)	含义
kem_keygen				pk	1696	用户公钥
				sk	640	用户私钥
kem_enc	pk	1696	接收方公钥	ct	1955	密文
				ss	95	共享秘密
kem_dec	ct	1955	密文	ss	95	共享秘密
	sk	640	接收方私钥			

6.1.2 时间与时钟周期消耗

按照大赛要求的测试环境，我们针对算法实现的三个部分，进行不同组数据的测试，最终得到的可收敛的性能耗时统计。

OKCN-SEC-RLWE 算法的时间与时钟周期消耗:

表 6.3（ubuntu 18.04.1 操作系统 和 windows7 操作系统+visual studio 2010 中的数据）给出了 Higncryption 密钥封装算法的时间与时钟周期消耗总结。

表 6.3：OKCN-SEC-RLWE 算法测试收敛性能表

测试平台类型	函数	耗时（微秒 μs ）	消耗时钟周期（个）
ubuntu 18.04.1 操作系统	密钥生成 (kem_keygen)	96	307000
	封装函数 (kem_enc)	166	530000
	解封装函数 (kem_dec)	51	164600
windows 操作系统 +visual studio 2010	密钥生成 (kem_keygen)	195	666000
	封装函数 (kem_enc)	325	1106000
	解封装函数 (kem_dec)	86	294500

关于测试的详细数据请参见“测试实例”文件夹下的测试报告。

6 支持文档

A Modular and Systematic Approach to Key Establishment and Public-Key Encryption Based on LWE and Its Variants.pdf

7 参考文献

- 1 . Erdem Alkim, L  o Ducas, Thomas P  ppelmann, and Peter Schwabe. Newhope without reconciliation. Cryptology ePrint Archive, Report 2016/1157, 2016.
<https://eprint.iacr.org/2016/1157>.
- 2 . Erdem Alkim, L  o Ducas, Thomas P  ppelmann, and Peter Schwabe. Post-quantum key exchange - A new hope. In 25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016., pages 327–343, 2016.
- 3 . M. R. Albrecht, R. Player and S. Scott. On the Concrete Hardness of Learning with Errors. Journal of Mathematical Cryptology, Volume 9, Issue 3, pages 169-203, 2015.
- 4 . M. R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HELib and SEAL. EUROCRYPT 2017: 103-129.
- 5 . M. Abe, R. Gennaro, K. Kurosawa and V. Shoup. Tag-KEM/DEM: A New Framework for Hybrid Encryption and A New Analysis of Kurosawa-Desmedt KEM. EU-ROCRYPT 2005: 128-146.
- 6 . V. Lyubashevsky, C. Peikert, and O. Regev. On Ideal Lattices and Learning with Errors over Rings. EUROCRYPT 2010: 1-23.
- 7 . B. Applebaum, D. Cash, C. Peikert, and A. Sahai. Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems. CRYPTO 2009: 595-618.
- 8 . S. Bai, A. Langlois, T. Lepoint, D. Stehl_e, and R. Steinfeld. Improved Security Proofs in Lattice-Based Cryptography: Using the Renyi Divergence rather than the Statistical Distance. ASIACRYPT 2015: 3-24.
- 9 . A. Banerjee and C. Peikert and A. Rosen. Pseudorandom Functions and Lattices. EUROCRYPT 2012: 719-737.
- 10 . A. Bogdanov, S. Guo, D. Masny, S. Richelson, and A. Rosen. On the Hardness of Learning with Rounding over Small Modulus. TCC 2016: 209-224.

- 11 . J. Bos, C. Costello, L. Ducas, I. Mironov, M. Naehrig, V. Nikolaenko, A. Raghunathan, and D. Stebila. Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE. ACM CCS 2016: 1006-1018.
- 12 . J.W. Bos, C. Costello, M. Naehrig, and D. Stebila. Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem. IEEE Symposium on Security and Privacy 2015, pages 553-570.
- 13 . J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, D. Stehl_e. CRYSTALS-Kyber: a CCA-Secure Module-lattice-based KEM. Cryptology ePrint Archive, Report 2017/634.
- 14 . Y. Chen and P.Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. ASI-ACRYPT 2011: 1-20.
- 15 . J.H. Cheon, D. Kim, J. Lee, and Y. Song. Lizard: Cut O_the Tail! Practical Post-Quantum Public-Key Encryption from LWE and LWR. Cryptology ePrint Archive, Report 2016/1126, 2016.
- 16 . D. Coppersmith and S. Winograd. Matrix Multiplication via Arithmetic Progressions. Journal of Symbolic Computation, volume 9, issue 3, pages 251-280, 1990.
- 17 . R. Cramer and V. Shoup. Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. SIAM Journal on Computing, 33(1): 167226, 2003.
- 18 . A. W. Dent. A Designers Guide to KEMs. Cryptology ePrint Archive, Report 2002/174, 2002.
- 19 . J. Ding, X. Xie and X. Lin. A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem. Cryptology ePrint Archive, Report 2012/688, 2012.
- 20 . Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. SIAM Journal on Computing, volume 38, issue 1, pages 97-139, 2008.
- 21 . L. Ducas and A. Durmus. Ring-LWE in Polynomial Rings. PKC 2012: 34-51.

- 22 . A. Duc, F. Tram_er, and S. Vaudenay. Better Algorithms for LWE and LWR. EUROCRYPT 2015: 173-202.
- 23 . E. Fujisaki and T. Okamoto. How to Enhance the Security of Public-Key Encryption at Minimum Cost. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences Volume 83, Issue 1, pages 24-32, 1999.
- 24 . E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. Journal of Cryptology, Volume 26, Issue 1, pages 80-101, 2013.
- 25 . C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. ACM STOC 2008: 197-206.
- 26 . S. Gueron and F. Schlieker. Speeding Up R-LWE Post-Quantum Key Exchange. Cryptology ePrint Archive, Report 2016/467, 2016.
- 27 . D. Harvey. Faster Arithmetic for Number-Theoretic Transforms. Journal of Symbolic Computation, 60: 113-119, 2014.
- 28 . D. Hofheinz, K. Hovelmanns, and Eike Kiltz. A Modular Analysis of the Fujisaki-Okamoto Transformation. Cryptology ePrint Archive, Report 2017/604.
- 29 . M. Kaplan, G. Leurent, A. Leverrier and M. Naya-Plasencia. Quantum Differential and Linear Cryptanalysis. ArXiv Preprint: 1510.05836, 2015.
- 30 . H. Krawczyk. SIGMA: The 'SIGn-and-MAC' Approach to Authenticated Diffie-Hellman and Its Use in the IKE Protocols CRYPTO 2003: 400-425.
- 31 . H. Krawczyk, K.G. Paterson and H. Wee. On the Security of the TLS Protocol: A Systematic Analysis. CRYPTO 2013: 429-448.
- 32 . H. Kuwakado and M. Morii. Quantum Distinguisher between the 3-round Feistel Cipher and the Random Permutation. IEEE ISIT 2010: 2682-2685.
- 33 . A. Langlois and D. Stehl_e. Worst-case to Average-case Reductions for Module Lattices. Des. Codes Cryptography, 75(3): 565-599, 2015.
- 34 . R. Lindner and C. Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. CT-RSA 2011: 319-339.

- 35 . E. Alkim, P. Jakubeit, and P. Schwabe. A New Hope on ARM Cortex-M. Cryptology ePrint Archive, Report 2016/758, 2016.
- 36 . Thomas Pöppelmann, Erdem Alkim, Roberto Avanzi, Joppe Bos, Léo Ducas, Antonio de la Piedra, Peter Schwabe, and Douglas Stebila. Supporting documentation: Newhope. Technical report, National Institute of Standards and Technology, 2017. <https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/Round-1-Submissions>.
- 37 . V. Lyubashevsky, C. Peikert, and O. Regev. A Toolkit for Ring-LWE Cryptography. EUROCRYPT 2013: 35-54
- 38 . P. Montgomery. Modular Multiplication Without Trial Division. Mathematics of Computation, vol. 44, 519521, 1985.
- 39 . K. G. Paterson, T. Ristenpart, and T. Shrimpton. Tag Size Does Matter: Attacks and Proofs for the TLS Record Protocol. ASIACRYPT 2011: 372-389.
- 40 . C. Peikert. Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem. STOC 2009: 333-342.
- 41 . C. Peikert. Lattice Cryptography for the Internet. PQCrypto 2014: 197-219.
- 42 . C. Peikert. A Decade of Lattice Cryptography. In Foundations and Trends in Theoretical Computer Science, Volume 10, Issue 4, pages 283-424, 2016.
- 43 . C. Peikert, O. Regev and N. Stephens-Davidowitz. Pseudorandomness of Ring-LWE for Any Ring and Modulus. STOC 2017: 461-473.
- 44 . C. Peikert, V. Vaikuntanathan, and B. Waters. A Framework for Efficient and Composable Oblivious Transfer. CRYPTO 2008: 554-571.
- 45 . A.V. Poppelmann, Cryptographic Decoding of the Leech Lattice. Cryptology ePrint Archive, Report 2016/1050, 2016.
- 46 . T. Poppelmann and T. Güneysu. Towards Practical Lattice-Based Public-Key Encryption on Reconfigurable Hardware. SAC 2013: 68-85.
- 47 . O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. Journal of the ACM (JACM), Volume 56, Issue 6, pages 34, 2009.

- 48 . M.K. Simon. Probability Distributions Involving Gaussian Random Variables: A Handbook for Engineers and Scientists. Springer, 2012.
- 49 . Jan-Pieter D'Anvers, Frederik Vercauteren and Ingrid Verbauwhede . The Impact of Error Dependencies on Ring/Mod-LWE/LWR Based Schemes. PQCrypt 2019.

8 原创性声明

本报告所提交的研究工作，除已明确标注和致谢的地方外，所有的观点、文字、图表及数据等均为自己的研究成果。他人研究对本研究工作的启发和贡献均已作了明确的说明和致谢。除文中已经注明引用的内容外，本报告不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：

时间： 年 月 日

9 未来工作

本算法提案聚焦基于 LWE 和 LWR 的密钥协商和密钥封装。由于 CPA 和 CCA 安全的公钥加密可以从密钥协商和密钥封装基于公开的成熟的及时转换而得到，在本提案中我们没有提供相应的转换。在未来的版本中，我们计划增加相应的 CCA 安全的公钥加密及其实现。

这儿，特别值得一提的是，我们的 AKCN4:1 和性能更为优良的 AKCN-E8-RLWE 在相同参数下，性能实质优于进入 NIST 后量子密码竞赛第二轮的 NewHope-KEM 算法。未来根据 NIST 后量子密码竞赛的进展和我国对基于理想格 RLWE 密码标准的需求，我们将继续丰富和完善我们的提案。