

AKCN-MLWE: 基于模格 MLWE 的 通用和模块化的密钥封装机制

算法说明书

赵运磊（复旦大学）

程蕾晓（复旦大学）

金正中（复旦大学）

巩博儒（复旦大学）

吴洪祥（银联商务股份有限公司）

隋光烨（上海扈民区块链科技有限公司）

张振峰（中国科学院软件研究所）

2019 年 1 月

摘要

格密码是目前对抗量子攻击的主要数学方法之一。模格介于一般格和理想格之间，其在灵活性、通用性和在 128 比特后量子安全级别的效率方面具有独特优势。在本提案中，我们给出了基于 MLWE (module learning-with-errors) 的通用和模块化的密钥封装协议。基于 MLWE 的密钥封装协议基于我们所发展的性能几乎达到最优界的 AKCN 非对称密钥共识机制为基础工具。为了适应不同的目标场景或者在安全性、性能和错误率之间取得更好的平衡，我们做了大量的测试去寻找最适合的具体参数设置。

格基密钥协商和格基密钥封装之间存在着紧密的关系，由其中一者不难构造得到另外一者。在本提案中，我们以介绍我们的基于 MLWE 和 AKCN 的密钥封装为主，以介绍与之相对应的基于 MLWE 和 SKCN 的密钥协商为辅。特别地，为了描述方便，本提案的部分内容以 MLWE-SKCN 密钥协商为例进行介绍和分析，由此得到的结论和量化的结果可以简单推广到 MLWE-AKCN 的密钥封装。类似的，我们的核心工具 AKCN 和用于密钥协商的 SKCN 之间也存在着紧密的关系或简单推广。

据我们所知，我们基于 MLWE 的密钥协商和密钥封装在 128 比特后量子安全级别具有优良的性能；其核心工具 SKCN 既用在密钥协商/公钥加密又是我们基于模格数字签名提案的核心工具，这也很好地体现了模块化和通用性。

特别地，与进入 NIST 后量子密码竞赛第二轮的明星提案 KYBER 相比，我们的提案同时提供了密钥协商和密钥封装，而 KYBER 仅提供了密钥封装；在 128 比特后量子安全级别，我们的算法在综合性能上优于 KYBER (更小的带宽、更低的错误率、削掉更多低位比特所带来的额外安全保障等)；并且 KYBER 的核心工具也是我们在 2016 年所公布的 AKCN 非对称密钥共识机制。

本提案的所有算法实现均是算法提交团队独立自主开发完成，没有使用国际开源代码 (比如 KYBER 的实现代码)，具有完全的自主知识产权，不存在代码应用的风险。

目录

1	引言.....	1
1.1	我们的贡献.....	2
2	预备知识.....	2
2.1	LWE 问题	3
2.2	环的选择	4
2.3	MLWE 问题	4
3	Key Consensus with Noise.....	5
3.1	KC 的有效上界	6
3.2	SKCN 的构造和分析	6
4	Asymmetric Key Consensus with Noise.....	9
4.1	AKCN 的构造和分析	10
4.2	KCvs.AKC 的讨论	12
5	基于 MLWE 的密钥协商和密钥封装	13
5.1	公钥密码体制中基于 SKCN-MLWE 的 KEX 算法描述.....	15
5.2	公钥密码体制中基于 AKCN-MLWE 的 KEM 算法描述.....	16
6	基于 MLWE 的密钥协商和封装分析	16
6.1	安全性分析.....	16
6.2	错误率分析.....	17
7	具体参数、和优缺点声明	18
8	算法实现代码及性能测试	20
8.1	基于模格的密钥封装性能分析.....	21
8.1.1	空间消耗.....	21
8.1.2	时间与时钟周期消耗	22
9	支持文档.....	23
10	参考文献.....	24
11	原创性声明	28
12	未来工作.....	29

1 引言

一旦大规模的量子计算机被制造出来,大多数基于普通离散对数、椭圆离散对数或者大整数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,这种大规模的计算机的到来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将被应用。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,就必须从现在开始将所有的密码技术替换为抗量子版本。在非对称密码学领域,最关键的技术就是密钥协商和密钥封装。

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[Reg09]。然而,基于 LWE 的密码系统通常都比较低效,因此学者提出了更高效的 ring-LWE (RLWE) 问题[LPR10]和 MLWE 问题[LS15]。在近些年,许多基于 LWE 及其变体的优秀工作涌现,其中大多数研究 [DXL12, Pei14, BCNS15, ADPS16, BCD+16, Reg09, GPV08, LP10, LPR10, LPR13, PG13, BDK+17, NIST] 集中在基于 LWE 及其变体的密钥协商和公钥加密协议的设计上。模格介于一般格和理想格之间,并且拥有灵活性、通用性和在构造密码系统时性能有竞争力等优良特性。在本提案中,我们设计了基于 MLWE 的密钥协商和密钥封装协议。

从技术的角度来说,最新的基于 LWE 及其变体的密钥协商和密钥封装协议(如 [ADPS16, BCD+16, PG13])的研究工作中的主要的贡献是改善了密钥协商机制(key reconciliation mechanisms) [DXL12, Pei14, LP10, NIST]。但是在之前的研究工作中,密钥协商机制仅以一种非黑盒的形式在 KE 和 PKE 中被使用和分析。这也就是说,对于未来用于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,对于密钥协商机制中的不同的参数,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[CDS94]然后利用 Fiat-Shamir 转换[FS86]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。抽象化和一般化在格基密码中也是非常有用的,因为格基密码通常更难以被理解和评估,并且也跟 NIST 后量子密码标准化[NIST]息息相关。

1.1 我们的贡献

在本提案中,通过引入和形式化被称为 KC (key consensus) 和 AKC (asymmetric key consensus) 的基础工具,我们提炼出已发表的基于 LWE 及其变体的密钥协商和 PKE 方案的关键成分。KC 和 AKC 允许通信双方首先通过某种安全的信息交换协议(比如交换 LWE 样本)得到两个比较接近的值,然后通过这两个接近的值来达到共识。对于任意的 KC 和 AKC,我们给出参数之间的上界。作为概念上的贡献,这大大的简化了未来密码系统的设计和分析。除此之外,我们分别设计和分析了一般化的是实用的 KC 和 AKC 方案,为了表述方便,它们分别被称为 SKCN (symmetric key consensus with noise) 和 AKCN (asymmetric key consensus with noise)。SKCN 和 AKCN 都是紧紧贴合被证明的参数上界的方案。特别地,我们的其中一个工作 [CGZ18] 说明了确定版本的 SKCN 可以作为构造基于格的签名方案的基石。借助 SKCN 和 AKCN,我们设计了基于 MLWE 的通用的模块化的密钥协商和密钥封装协议。最后,为了适应不同的目标场景或者在安全性、性能和错误率之间取得更好的平衡,我们做了大量的测试去寻找最合适的具体的参数设置。

2 预备知识

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, $x \leftarrow S$ 表示均匀随机的从 S 中取一个元素, $\mathcal{U}(S)$ 表示 S 上的一个均匀分布。

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 $x \leftarrow \mathcal{D}$ 表示根据分布 \mathcal{D} 选择一个元素 x 。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 和随机种子 r 作为输入的 A 的运算结果。 $y \leftarrow A(x_1, x_2, \dots)$ 表示随机选取 r 并将运行结果 $A(x_1, x_2, \dots; r)$ 赋值给 y 。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。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。

2.1 LWE 问题

给定正数 $\sigma > 0$, 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, 其概率密度函数为 $D_{\mathbb{Z}, \sigma}(x) \triangleq \rho_\sigma(x)/\rho_\sigma(\mathbb{Z})$, $x \in \mathbb{Z}$ 。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, 其中每个坐标都从 $D_{\mathbb{Z}, \sigma}$ 中独立选取。

给定正整数 n 和 q , 它们都是关于安全参数 λ 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 。令 $A_{q, \mathbf{s}, \chi}$ 是 $\mathbb{Z}_q^n \times \mathbb{Z}_q$ 上的分布: 随机均匀选择 $\mathbf{a} \in \mathbb{Z}_q^n$, 选取误差项 $e \leftarrow \chi$, 并输出 $(\mathbf{a}, \mathbf{b} = \mathbf{a}^T \mathbf{s} + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ 。这里误差分布 χ 通常被认为是离散高斯概率分布 $D_{\mathbb{Z}, \sigma}$; 但是, 如文献[BCD+16]中所述, 也可以采用其他的 χ 分布。简而言之, (判定版本的) LWE 假设[Reg09]认为, 对足够大的安全参数 λ , 没有概率多项式时间 (PT) 算法能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。即使敌手 \mathcal{A} 看到多项式个样本, 并且即使私密向量 \mathbf{s} 是从 χ^n 随机抽取的[ACPS09], 该结论仍成立。

2.2 环的选择

对于正整数 m , $\Phi_m(x) \in \mathbb{Z}[x]$ 表示 m 阶分圆多项式 (cyclotomic polynomial)。 $\varphi(\cdot)$ 表示欧拉函数 (Euler's phi function)。下面我们显式地罗列三种类型的环：

- **幂次环**： n 是2的幂次， q 是素数且 $q \equiv 1 \pmod{2n}$ 。故 $n = \varphi(m)$ 且 $q \equiv 1 \pmod{m}$ 。在这种情形中， $\Phi_m(X) = X^n + 1$ 。定义环

$$\mathcal{R} = \mathbb{Z}[X]/(X^n + 1), \mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1).$$

这是环 $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$ 的特殊情形。

- **安全素数环-1**： m' 是素数， m 是使得 $m = 2m' + 1$ 的安全素数。 e 是使得 $2^e > 2m$ 的最小的整数， q 是使得 $q \equiv 1 \pmod{2^e \cdot m}$ 的素数。在这种情形中， $\Phi_m(X) = X^{m-1} + X^{m-2} + \dots + 1 = X^n + X^{n-1} + \dots + 1$ ， 其中 $n = m - 1 = 2m'$ 。定义环

$$\mathcal{R} = \mathbb{Z}[X]/(X^n + X^{n-1} + \dots + 1), \mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + X^{n-1} + \dots + 1).$$

- **安全素数环-2**： $n + 1$ 是安全素数。在这种情形中， $\Phi_{n+1}(X) = X^n + X^{n-1} + \dots + 1$ ，模数 q 是2的幂次或者是使得 $q \equiv 1 \pmod{n+1}$ 的素数。定义环

$$\mathcal{R} = \mathbb{Z}[X]/(X^n + X^{n-1} + \dots + 1), \mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + X^{n-1} + \dots + 1).$$

在下文中，我们不加说明地将多项式 $a = \sum_{i=0}^{n-1} a_i x^i \in \mathcal{R} \ (\mathcal{R}_q)$ 跟向量 $\mathbf{a} = (a_0, a_1, \dots, a_{n-1}) \in \mathbb{Z}^n \ (\mathbb{Z}_q^n)$ 等同。因此，对于多项式 $a \in \mathcal{R} \ (\mathcal{R}_q)$ 和任意 $\mathbb{Z}^n \ (\mathbb{Z}_q^n)$ 上的分布 \mathcal{D} ， $a \leftarrow \mathcal{D}$ 指的是以分布 \mathcal{D} 取向量 \mathbf{a} 并将该向量对应为相应的多项式 a 。类似地，一个 k 维的多项式向量 $\mathbf{a} \in \mathcal{R}^k \ (\mathcal{R}_q^k)$ 可以按照分布 \mathcal{D}^k 生成。

2.3 MLWE 问题

令 $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$ 为上节三种环中的任一类环。在本提案中，为了更高效的实现，在实际应用中我们使用幂次环，即 $\mathcal{R} = \mathbb{Z}[X]/(X^n + 1), \mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$ 。

令 l 是一个正整数的参数，令 S_η 表示 $\mathbf{w} \in \mathcal{R}$ 中满足 $\|\mathbf{w}\|_\infty \leq \eta$ 的所有元素所构

成的集合¹。MLWE 问题[LS15]是 RLWE 问题的推广，我们使用文献[BDK+17]中的定义。

- **MLWE 分布**: 来自 MLWE 分布的样本 $(\mathbf{a}_i, b_i) \in \mathcal{R}_q^l \times \mathcal{R}_q$ ，其中 $\mathbf{a}_i \leftarrow \mathcal{R}_q^l$ 均匀， $b_i = \mathbf{a}_i^T \mathbf{s} + e_i$ ，所有样本使用相同的 $\mathbf{s} \leftarrow S_\eta^l$ 和不同的 $e_i \leftarrow S_\eta$ ， $1 \leq i \leq h$ 。
- **MLWE 假设**: MLWE 问题是从多项式个来自 MLWE 分布的样本中恢复 \mathbf{s} 。具体地，对于敌手 A ，定义

$$\text{Adv}_{h,l,\eta}^{\text{mlwe}}(A) = \left[\mathbf{x} = \mathbf{s}: \begin{array}{l} \mathbf{A} \leftarrow \mathcal{R}_q^{h \times l}; (\mathbf{s}, \mathbf{e}) \leftarrow S_\eta^l \times S_\eta^h; \\ \mathbf{b} \leftarrow \mathbf{A}\mathbf{s} + \mathbf{e}; \mathbf{x} \leftarrow A(\mathbf{A}, \mathbf{b}); \end{array} \right]$$

我们称 (t, ϵ) -MLWE $_{h,l,\eta}$ 困难性假设成立，如果没有最多运行 t 时间的算法 A 的优势大于 ϵ 。

3 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。

定义 3.1 一个密钥共识算法 $KC = (params, Con, Rec)$ 的定义如下：

- $params = (q, m, g, d, aux)$ 表示系统参数，其中 q, m, g, d 为正整数并且满足 $2 \leq m, g \leq q, 0 \leq d \leq \left\lfloor \frac{q}{2} \right\rfloor$ ， 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；
- $(k_1, v) \leftarrow Con(\sigma_1, params)$: 在输入为 $(\sigma_1 \in \mathbb{Z}_q, params)$ 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， $v \in$

¹ S_η 通常基于文献[BDK+17] 给出的中心二项分布: 选取 $(a_1, \dots, a_\eta, b_1, \dots, b_\eta) \leftarrow \{0, 1\}^{2\eta}$ ，再输出 $\sum_{i=1}^\eta (a_i - b_i)$ 。该分布的标准差是 $\sqrt{\eta/2}$ 。

\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，如果对任意 $\sigma_1 \leftarrow \mathbb{Z}_q$ ， k_1 和 v 是相互独立的，且 k_1 在 \mathbb{Z}_m 上均匀分布。算法中的概率来源于 σ_1 取样的和 Con 中使用的随机种子。

3.1 KC 的有效上界

下面的这个定理提出了关于参数 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）的上界，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 的详细证明参见支持文档所列论文附录 A。

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 $2md \leq q(1 - \frac{1}{g})$ 。

3.2 SKCN 的构造和分析

令 $r \bmod^\pm \alpha$ 表示唯一的 $r'' \in [-\lfloor \frac{\alpha-1}{2} \rfloor, \lfloor \frac{\alpha-1}{2} \rfloor]$ 使得 $\alpha \mid (r'' - r)$ 的整数。对于正整数 t 和 $x \in \mathbb{Z}_t$ ，有 $|x|_t = |x \bmod^\pm t|$ 。密钥协商方案 SKCN 在算法 1 中描述。

算法 1SKCN: 带噪音的对称且四舍五入的密钥共识算法

```

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

```

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 $|x - y|_t \leq l$, 那么存在 $\theta \in \mathbb{Z}, \delta \in [-l, l]$ 使得 $x = y + \theta t + \delta$ 。

定理 3.2 若系统参数满足 $(2d + 1)m < q(1 - \frac{1}{g})$, 其中 $m \geq 2, g \geq 2$, 那么 SKCN 算法满足正确性。

证明-定理 3.2 假设 $|\sigma_1 - \sigma_2| \leq d$, 由事实 3.1 可知, 存在一个 $\theta \in \mathbb{Z}$ 并且 $\theta \in [-d, d]$, 使得 $\sigma_2 = \sigma_1 + \theta q + \delta$ 。从算法 1 的第四行到第六行可知, 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,

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

注意到 $|v'/\beta - v/g| = |v'g - \beta v|/\beta g \leq 1/2g$, 故

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

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, 有 $k_2 = k_1$ 。

定理 3.3SKCN 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 取样和 Con 中使用的随机数。

证明-定理 3.3 在之前的条件中，我们有 $q' = \text{lcm}(q, m)$, $\alpha = \frac{q'}{q}$, $\beta = \frac{q'}{m}$ 。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 中元素的表达形式如算法 1 所示。因为 $\sigma_1 \leftarrow \mathbb{Z}_q$ 和 $e \leftarrow \mathbb{Z}_\alpha$ 服从均匀分布，并且它们之间相互独立，故 $\sigma_A = (\alpha\sigma_1 + e) \bmod q' = f(\sigma_1, e)$ 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，故 (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。

另外，当算法中的参数满足 $q = 2^{\bar{q}}, g = 2^{\bar{g}}, m = 2^{\bar{m}}$ ，也就是 q, g, m 都是 2 的幂次时，并且 $q = mg$ 时，SKCN 可简化为 SKCN Simple，如算法 2 所示。

算法 2 SKCN 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:    $k_1 = \left\lfloor \frac{\sigma_1}{g} \right\rfloor$ 
4:    $v = \sigma_1 \bmod g$ 
5:   return  $(k_1, v)$ 
6: end procedure
7: procedure REC( $\sigma_2, v$ , params)
8:    $k_2 = \left\lfloor \frac{\sigma_2 - v}{g} \right\rfloor \bmod m$ 
9:   return  $k_2$ 
10: end procedure

```

推论 3.1 如果 g 和 m 是 2 的幂次，并且 $q = mg, 2md < q$ ，那么算法 2 满足正确性和安全性。

证明-推论 3.1 假设 $|\sigma_1 - \sigma_2| \leq d$ ，由事实 3.1 可知，存在一个 $\theta \in \mathbb{Z}$ 并且 $\theta \in [-d, d]$ ，使得 $\sigma_2 = \sigma_1 + \theta q + \delta$ 。将等式代入到算法 2 的第八行中 k_2 的计算中可得，

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

由假设的条件 $2md < q$, 有 $|\delta/g| \leq d/g < 1/2$, 因此, $k_2 = k_1 \bmod m = k_1$ 。

关于安全性的证明, 因为算法 2 是算法 1 的特殊情形, 故安全性成立。

4 Asymmetric Key Consensus with Noise

定义 4.1 一个非对称密钥共识算法 $AKC = (params, Con, Rec)$ 定义如下:

- $params = (q, m, g, d, aux)$ 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;
- $v \leftarrow Con(\sigma_1, k_1, params)$: 在输入为 $(\sigma_1 \in \mathbb{Z}_q, k_1 \in \mathbb{Z}_m, params)$ 的条件下, 概率多项式算法 Con 输出公开提示信息 $v \in \mathbb{Z}_g$;
- $k_2 \leftarrow Rec(\sigma_2, v, params)$: 在输入为 $(\sigma_2, v, params)$ 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 的分布相互独立。也就是说, 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 $\Pr[v = \tilde{v} | k_1 = \tilde{k}_1] = \Pr[v = \tilde{v} | k_1 = \tilde{k}_1']$, 其中概率取自 $\sigma_1 \leftarrow \mathbb{Z}_q$ 和 Con 中使用的随机性。

定理 4.1 令 AKC 表示一个参数为 $params = (q, m, d, g, aux)$ 非对称密钥共识算法, 如果 AKC 满足正确性和安全性, 那么 $2md \leq q(1 - \frac{m}{g})$ 。

定理 4.1 的详细证明参见支持文档所列论文附录 B。将定理 4.1 中的 $2md \leq q(1 - \frac{m}{g})$ 和定理 3.1 中的 $2md \leq q(1 - \frac{1}{g})$ 相比较, 我们可以发现两者之间只相差了一个分子 m , 这表明对于相同的 (q, m, d) , AKC 机制相对于 KC 机制而言带宽

g 更大。KC/AKC 与模糊提取 (fuzzyextractor) [DORS08]之间的关系讨论参见支持文档所列论文附录 C。

4.1 AKCN 的构造和分析

算法 3 描述了带噪音的非对称密钥共识算法(Asymmetric Key Consensus with Noise, AKCN)。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 $g\lfloor k_1 q/m \rfloor$ 。

算法 3 AKCN: 带噪音的非对称密钥共识算法

```

1:  $\text{params} = (q, m, g, d, aux)$ , where  $aux = \emptyset$ .
2: procedure 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 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
```

本文所证明的 AKC 的参数间的上界是 AKCN 的设计动机。在 AKCN 的设计中，我们结合了现有文献的优化方法，只为了尽可能满足定理 4.1 所给出的上界。AKCN 是基础的调和机制 (reconciliation mechanisms) [LPR10, LP10]的一般化，也受到了 SKCN 和文献[BPR12, PG13]的启发²。特别地，文献[LPR10, LP10]中的调和机制对应着 $g = q$ 且 $m = 2$ 时特殊的 AKCN。

简而言之，AKCN 的创新性体现在两个方面：(1) Con 程序的设计结合了[LPR10, LP10]中的基础的调和机制以及[BPR12, Pei09]中的四舍五入技巧 (rounding technique)，而且 Rec 程序的设计也不是那么直接；(2) 参数 m 一般化后允许多比特的共识。据我们所知，现今的工作都不能单独地实例化出 AKCN。

定理 4.2 假设 AKCN 的参数满足 $(2d+1)m < q(1 - \frac{m}{g})$ ，那么算法 3 描述的 AKCN 算法满足正确性。

证明-定理 4.2 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

²AKCN 和[PG13]的底层调和机制大体上可以看成是不兼容的。

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

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 $\sigma_1 = \sigma_2 + \theta' q + \delta$ ，因此

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

由于 $|m\delta/q + m\varepsilon_1/q + m\varepsilon_2/g| \leq md/q + m/2q + m/2g < 1/2$ ，故 $k_1 = k_2$ 。

定理 4.3 AKCN 算法满足安全性。具体来说，当 $\sigma_1 \leftarrow \mathbb{Z}_q$ 时， k_1 和 v 相互独立。

证明-定理 4.3 对任意 $\tilde{v} \in \mathbb{Z}_g$ 和任意 $\tilde{k}_1, \tilde{k}'_1 \in \mathbb{Z}_m$ ，我们证明当 $\sigma_1 \leftarrow \mathbb{Z}_q$ 时，有 $\Pr[v = \tilde{v} | k_1 = \tilde{k}_1] = \Pr[v = \tilde{v} | k_1 = \tilde{k}'_1]$ 。对 $\mathbb{Z}_m \times \mathbb{Z}_g$ 中的任意 (\tilde{k}, \tilde{v}) ，事件 $(v = \tilde{v} | k_1 = \tilde{k})$ 等价于存在 $\sigma_1 \leftarrow \mathbb{Z}_q$ 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 $\tilde{v} = \lfloor g(\sigma_1 + \lfloor \tilde{k}q/m \rfloor)/q \rfloor \bmod g$ 当且仅当存在 $\varepsilon \in (-\frac{1}{2}, \frac{1}{2}]$ 和 $\theta \in \mathbb{Z}$ ，使 $\tilde{v} = g(\sigma_1 + \lfloor \tilde{k}q/m \rfloor)/q + \varepsilon - \theta g$ 。也就是说，存在 $\varepsilon \in (-1/2, 1/2]$ ，使得 $\sigma_1 = (q(\tilde{v} - \varepsilon)/g - \lfloor \tilde{k}q/m \rfloor) \bmod q$ 。令 $\Sigma(\tilde{v}, \tilde{k}) = \{\sigma_1 \in \mathbb{Z}_q | \exists \varepsilon \in (-\frac{1}{2}, \frac{1}{2}] \text{ 使得 } \sigma_1 = (q(\tilde{v} - \varepsilon)/g - \lfloor \tilde{k}q/m \rfloor) \bmod q\}$ 。定义映射 $\Phi: \Sigma(\tilde{v}, 0) \rightarrow \Sigma(\tilde{v}, \tilde{k}), \Phi(x) = (x - \lfloor \tilde{k}q/m \rfloor) \bmod q$ 。显然， Φ 是一个一一映射。因此， $\Sigma(\tilde{v}, \tilde{k})$ 的基数和 \tilde{k} 无关。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 $|\Sigma(\tilde{v}, \tilde{k})| = |\Sigma(\tilde{v}, \tilde{k}')| = |\Sigma(\tilde{v}, 0)|$ 。故对于任意的 $\tilde{v} \in \mathbb{Z}_g$ 和任意的 $\tilde{k}_1 \in \mathbb{Z}_m$ ，当 $\sigma_1 \leftarrow \mathbb{Z}_q$ 时，有 $\Pr[v = \tilde{v} | k_1 = \tilde{k}] = \Pr[\sigma_1 \in \Sigma(\tilde{v}, \tilde{k}) | k_1 = \tilde{k}] = |\Sigma(\tilde{v}, \tilde{k})|/q = |\Sigma(\tilde{v}, 0)|/q$ 。等式右边只依赖于 \tilde{v} 的值，因此 v 和 k_1 是相互独立的。

另外，当算法中的参数满足 $q = 2^{\bar{q}}, m = 2^{\bar{m}}$ ，也就是 q, g, m 都是 2 的幂次，且 $q = gm$ 时，AKCN 算法可以简化成 AKCN simple，如算法 4 所示。

算法 4 AKCN simple

```
1:  $\text{params} = (q, m, g, d, \text{aux})$ , 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( $\sigma_1, k_1, \text{params}$ )  $\triangleright \sigma_1 \in [0, q-1]$ 
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( $\sigma_2, v, \text{params}$ )  $\triangleright \sigma_2 \in [0, q-1]$ 
7:    $k_2 = \lfloor (mv - \sigma_2) / g \rfloor \bmod m$ 
8:   return  $k_2$ 
9: end procedure
```

推论 4.1 如果 q, g, m 都是 2 的幂次满足 $q = gm$, 并且 d, m, q 满足 $m + 2d < g$, 那么算法 4 描述的 AKCN-simple 满足正确性和安全性。

证明-推论 4.1 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 $v = \sigma_1 2^{-\bar{m}} + k_1 2^{\bar{q}-\bar{m}} + \varepsilon + \theta' q$ 。将这些代入到计算 k_2 的公式中, 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 4 作为 AKCN 机制的一个特殊的情况, 安全性可以直接根据算法 3 的安全性证明推导得到。

4.2 KCvs.AKC 的讨论

基于 KC 和 AKC 的密钥协商和密钥封装协议具有以下不同的性能和特点:

- 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 的密钥协商对应于 ELGamal 密钥传输或密钥封装;
- 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。
- 对于相同的参数 (q, m, g) (意味着相同的带宽), 基于 SKCN 的密钥协商比基于 AKCN 的方案具有更低的错误率。或者说, 对于相同的参数

(q, m, d) (意味着相同的错误率), 基于 SKCN 的密钥协商比基于 AKCN 的方案具有更小的带宽。这个比较是根据定理 3.1 和 4.1 中参数之间的上界不等式所得。

- 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传输协议中或者 CPA 安全的 PKE 机制中; 另一方面特定版本的 SKCN 还可以用于构造基于格密码的签名方案[CGZ18];
- 基于 KC 的密钥协商更加适用于目前的很多基于 Diffie-Hellman 和 SIGMA 机制[Kra03]的标准例如 IKE 和 TLS。我们注意到在 TLS1.3 中密钥传输已经明确被弃用了[Res]。

基于以上理由, 在本提案中我们更多的关注基于 KC (而非 AKC) 的密钥协商和密钥封装协议 (特别是密钥协商)。为了简化系统的复杂性, 我们仍然致力于寻找可以同时被 KC 或 AKC 实例化统一的协议结构。

5 基于 MLWE 的密钥协商和密钥封装

令 $(Con, Rec, params = (q, m, g, d))$ 是正确且安全的 KC 方案, 或者是 $m = 2$ 的 AKC 方案。令 Gen 为一个伪随机生成器 (PRG), 我们利用 Gen 输入小种子 $seed$ 生成矩阵 A , $seed$ 的长度是 κ 。

基于 SKCN 和 MLWE 的密钥协商协议如图 1 所示, 基于 AKCN 和 MLWE 的密钥传输协议如图 2 所示。第 5.1 节和 5.2 节则分别描述了如何在公钥密码体制中应用 SKCN 和 AKCN 构造基于 MLWE 的 KEM。为了可证明安全, 参数 t_1 会被设为 0。但是, 为了简单和对称性, 我们也会假设 Y_1 和 Y_2 裁剪了相同数目的比特, 即令 $t = t_1 = t_2 \geq 0$ 。

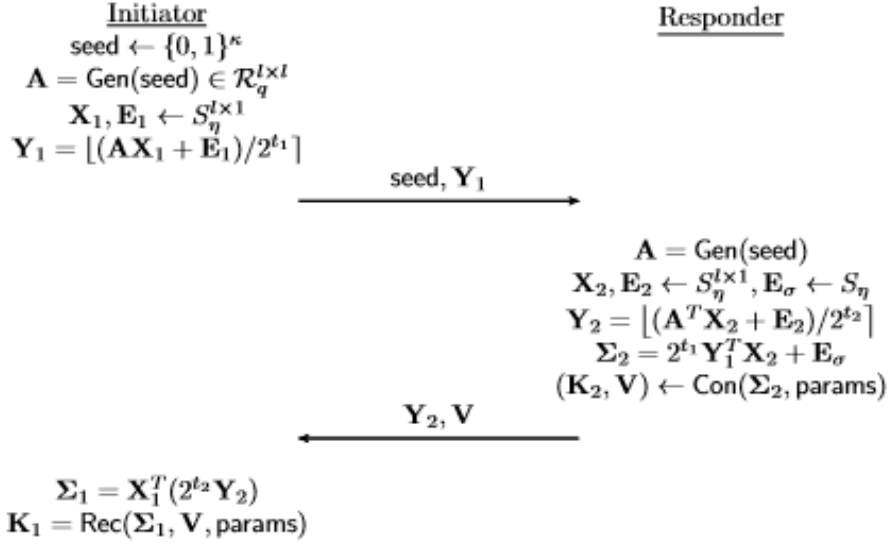


图 1 SKCN-MLWE 的通用构造

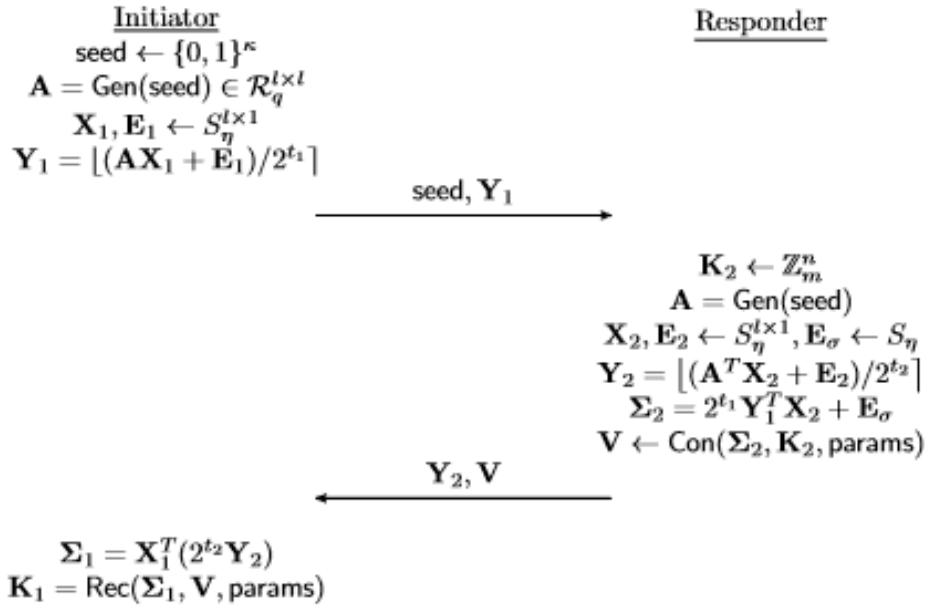


图 2 AKCN-MLWE 的通用构造，其中 $m = 2$

5.1 公钥密码体制中基于 SKCN-MLWE 的 KEX 算法描述

在公钥密码体制中，基于 SKCN-MLWE 的 KEX 描述如下。为了简单起见，令 $\mathbf{K} = \mathbf{K}_1 = \mathbf{K}_2$ 。在 KEX 的实际实现中，共享密钥是由 \mathbf{K} 导出的，交互则利用了 KDF (key derivation function) (如密码哈希函数或 HMAC 等) 实现。

到目前为止，所有基于格的密钥交换协议，都是非对称结构的。具体在我们的基于 SKCN-MLWE 的 KEX 算法中，

- Alice 和 Bob 在密钥生成的过程中，需要共享一个矩阵 \mathbf{A} 。在调用 KeyGen() 函数时，通过传递相同的 seed 参数，来确保两者共享矩阵 \mathbf{A} 。但 Alice 和 Bob 在计算各自的共享密钥过程中，分别使用矩阵 \mathbf{A} 和 \mathbf{A} 的转置矩阵，因此，我们通过两个不同的函数调用 $(pk_1, sk_1) := \text{KeyGen_Alice}(\text{seed})$, $(pk_2, sk_2) := \text{KeyGen_Bob}(\text{seed})$ 来体现它们的区别和相似性。

具体地说，Alice 通过调用 $\text{KeyGen_Alice}(\text{seed})$ 来生成自己的公钥/私钥对 (pk_1, sk_1) ，而 Bob 通过调用 $\text{KeyGen_Bob}(\text{seed})$ 来生成自己的公钥/私钥对 (pk_2, sk_2) 。

- Alice 和 Bob 需要用两个不同的密钥导出函数，来生成各自的共享密钥。具体地说，Bob 需要使用 Alice 的公钥 pk_1 和自己的私钥 sk_2 ，通过调用 $\text{KDF-Bob}(pk_1, sk_2)$ ，来生成自己的共享密钥 ss_2 以及发送给 Alice、帮助 Alice 生成共享密钥的启示信息 signal ；而只有在收到 Bob 发送的启示信息 signal 后，Alice 才能使用 Bob 的公钥 pk_2 、自己的私钥 sk_1 ，通过调用 $\text{KDF-Alice}(pk_2, sk_1, \text{signal})$ ，来生成自己的共享密钥 ss_1 。

Algorithm 1 $(pk_1, sk_1) \leftarrow \text{KeyGen-Alice}(\text{seed})$

```

1:  $\mathbf{A} := \text{Gen}(\text{seed})$ 
2:  $\mathbf{X}_1, \mathbf{E}_1 \leftarrow S_{\eta}^{l \times 1}$ 
3:  $\mathbf{Y}_1 := \lfloor (\mathbf{A}\mathbf{X}_1 + \mathbf{E}_1) / 2^{t_1} \rfloor$ 
4: return  $(pk_1 := \mathbf{Y}_1, sk_1 := \mathbf{X}_1)$ 

```

Algorithm 2 $(pk_2, sk_2) \leftarrow \text{KeyGen-Bob}(\text{seed})$

```

1:  $\mathbf{A} := \text{Gen}(\text{seed})$ 
2:  $\mathbf{X}_2, \mathbf{E}_2 \leftarrow S_{\eta}^{l \times 1}$ 
3:  $\mathbf{Y}_2 := \lfloor (\mathbf{A}^T \mathbf{X}_2 + \mathbf{E}_2) / 2^{t_1} \rfloor$ 
4: return  $(pk_2 := \mathbf{Y}_2, sk_2 := \mathbf{X}_2)$ 

```

Algorithm 3 $(ss_2, \text{signal}) \leftarrow \text{KDF-Bob}(pk_1, sk_2)$

```

1:  $\mathbf{E}_{\sigma} \leftarrow S_{\eta}^{l \times 1}$ 
2:  $\Sigma_2 := 2^{t_1} \mathbf{Y}_1^T \mathbf{X}_2 + \mathbf{E}_{\sigma}$ 
3:  $(\mathbf{K}_2, \mathbf{V}) \leftarrow \text{Con}(\Sigma_2, \text{params})$ 
4: return  $(ss_2 := \mathbf{K}_2, \text{signal} := \mathbf{V})$ 

```

Algorithm 4 $ss_1 \leftarrow \text{KDF-Alice}(pk_2, sk_1, \text{signal})$

```

1:  $\Sigma_1 := \mathbf{X}_1^T (2^{t_2} \mathbf{Y}_2)$ 
2:  $\mathbf{K}_1 := \text{Rec}(\Sigma_1, \mathbf{V}, \text{params})$ 
3: return  $ss_1 := \mathbf{K}_1$ 

```

5.2 公钥密码体制中基于 AKCN-MLWE 的 KEM 算法描述

在公钥密码体制中，基于 AKCN-MLWE 的 KEM 描述如下。

Algorithm 8 $(pk, sk) \leftarrow \text{KeyGen}()$

```

1:  $\text{seed} \leftarrow \{0, 1\}^{\kappa}$ 
2:  $\mathbf{A} := \text{Gen}(\text{seed})$ 
3:  $\mathbf{X}_1, \mathbf{E}_1 \leftarrow S_{\eta}^{l \times 1}$ 
4:  $\mathbf{Y}_1 := \lfloor (\mathbf{A}\mathbf{X}_1 + \mathbf{E}_1) / 2^{t_1} \rfloor$ 
5: return  $(pk := (\text{seed}, \mathbf{Y}_1), sk := \mathbf{X}_1)$ 

```

Algorithm 9 $(ct, \text{key}) \leftarrow \text{Encaps}(pk)$

```

1:  $\mathbf{K}_2 \leftarrow \mathbb{Z}_m^n$ 
2:  $\mathbf{X}_2, \mathbf{E}_2 \leftarrow S_{\eta}^{l \times 1}, \mathbf{E}_{\sigma} \leftarrow S_{\eta}$ 
3:  $\mathbf{A} := \text{Gen}(\text{seed})$ 
4:  $\mathbf{Y}_2 := \lfloor (\mathbf{A}^T \mathbf{X}_2 + \mathbf{E}_2) / 2^{t_2} \rfloor$ 
5:  $\Sigma_2 := 2^{t_1} \mathbf{Y}_1^T \mathbf{X}_2 + \mathbf{E}_{\sigma}$ 
6:  $\mathbf{V} \leftarrow \text{Con}(\Sigma_2, \mathbf{K}_2, \text{params})$ 
7: return  $(ct := (\mathbf{Y}_2, \mathbf{V}), \text{key} := \mathbf{K}_2)$ 

```

Algorithm 10 $\text{key}' \leftarrow \text{Decaps}(sk, ct)$

```

1:  $\Sigma_1 := \mathbf{X}_1^T (2^{t_2} \mathbf{Y}_2)$ 
2:  $\mathbf{K}_1 := \text{Rec}(\Sigma_1, \mathbf{V}, \text{params})$ 
3: return  $\text{key}' := \mathbf{K}_1$ 

```

6 基于 MLWE 的密钥协商和封装分析

6.1 安全性分析

定义 6.1 我们称一个基于 MLWE 以及 KC 或 AKC 的密钥协商协议是安全的，若对任意足够大的安全参数 λ 和任意 PT 敌手 \mathcal{A} ，在算法 11 定义的游戏 G_0 中，绝对值

$$\left| \Pr[b' = b] - \frac{1}{2} \right|$$

是可忽略的。

Algorithm 11 Game G_0

```

1:  $\mathbf{A} \leftarrow \mathcal{R}_q^{l \times l}$ 
2:  $\mathbf{X}_1, \mathbf{E}_1 \leftarrow S_\eta^{t \times 1}$ 
3:  $\mathbf{Y}_1 = \mathbf{A}\mathbf{X}_1 + \mathbf{E}_1$ 
4:  $\mathbf{X}_2, \mathbf{E}_2 \leftarrow S_\eta^{t \times 1}$ 
5:  $\mathbf{Y}_2 = \mathbf{A}^T \mathbf{X}_2 + \mathbf{E}_2$ 
6:  $\mathbf{E}_\sigma \leftarrow S_\eta$ 
7:  $\Sigma_2 = \mathbf{Y}_1^T \mathbf{X}_2 + \mathbf{E}_\sigma$ 
8:  $(\mathbf{K}_2^0, \mathbf{V}) \leftarrow \text{Con}(\Sigma_2, \text{params})$ 
9:  $\mathbf{K}_2^1 \leftarrow \mathbb{Z}_m^n$ 
10:  $b \leftarrow \{0, 1\}$ 
11:  $b' \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{Y}_1, \lfloor \mathbf{Y}_2 / 2^{t_2} \rfloor, \mathbf{K}_2^b, \mathbf{V})$ 

```

定理 6.1 若 $(\text{Con}, \text{Rec}, \text{params} = (q, m, g, d))$ 是一个正确且安全 KC 或 AKC 方案且 $t_1 = 0$ ，在 MLWE 假设下，图 1 和 2 中的密钥协商协议是安全的。

证明-定理 6.1 详细的安全证明参考支持文档所列的论文。

6.2 错误率分析

令 $t_1 = t_2 = t$ ，记 $\epsilon_2 = \mathbf{A}^T \mathbf{X}_2 + \mathbf{E}_2 - 2^t \lfloor (\mathbf{A}^T \mathbf{X}_2 + \mathbf{E}_2) / 2^t \rfloor$ ， $\epsilon_1 = \mathbf{A} \mathbf{X}_1 + \mathbf{E}_1 - 2^t \lfloor (\mathbf{A} \mathbf{X}_1 + \mathbf{E}_1) / 2^t \rfloor$ ，则

$$\begin{aligned}
\Sigma_1 - \Sigma_2 &= \mathbf{X}_1^T (2^t \mathbf{Y}_2) - (2^t \mathbf{Y}_1^T \mathbf{X}_2 + \mathbf{E}_\sigma) \\
&= 2^t \mathbf{X}_1^T \lfloor (\mathbf{A}^T \mathbf{X}_2 + \mathbf{E}_2) / 2^t \rfloor - ((2^t \lfloor (\mathbf{A} \mathbf{X}_1 + \mathbf{E}_1) / 2^t \rfloor)^T \mathbf{X}_2 + \mathbf{E}_\sigma) \\
&= \mathbf{X}_1^T (\mathbf{A}^T \mathbf{X}_2 + \mathbf{E}_2 - \epsilon_2) - ((\mathbf{A} \mathbf{X}_1 + \mathbf{E}_1 - \epsilon_1)^T \mathbf{X}_2 + \mathbf{E}_\sigma) \\
&= \mathbf{X}_1^T (\mathbf{E}_2 - \epsilon_2) - (\mathbf{E}_1 - \epsilon_1)^T \mathbf{X}_2 - \mathbf{E}_\sigma
\end{aligned}$$

根据 MLWE 假设可知， $(\mathbf{A}, \mathbf{A}^T \mathbf{X}_2 + \mathbf{E}_2)$ 与 (\mathbf{A}, \mathbf{U}) 不可区分，其中 \mathbf{U} 服从均匀分布，故 $\epsilon_i (i = 1, 2)$ 非常接近 $\mathbf{U} - 2^t \lfloor \mathbf{U} / 2^t \rfloor$ ，我们可以把多项式向量 $\mathbf{U} - 2^t \lfloor \mathbf{U} / 2^t \rfloor$ 中的每个分量看成 $[-2^{t-1}, 2^{t-1}]^n$ 上的均匀分布。

基于以上分析，对于多项式 $\Sigma_2 - \Sigma_1$ 中的每个系数，我们可以计算它的标准差 s ，即

$$s^2 = 2nl\sigma^2 \left(\sigma^2 + \frac{(1 + 2^t)^2 - 1}{12} \right) + \sigma^2,$$

其中 $\sigma = \sqrt{\eta/2}$ 。至此，我们可以通过运行程序脚本得到错误率。

7 具体参数、和优缺点声明

表格 7.1 给出了 AKCN-MLWE 的参数和性能。其中，“pq-sec”指的是 $t = 0$ 时对底层格问题的现有的最好的量子攻击，具体的安全性的值可通过运行文献 [BDK+17] 中的脚本得到。我们相信在实际中裁剪 Y_1 和 Y_2 最低的 t 个有效比特可以增强安全性（特别是对一次性的密钥协商和传输而言）。

注意到在安全性证明中我们设置 $t_1 = 0$ ，在实际的实现中却跟 Kyber [BDK+17] 一样取 $t_1 = t_2 = t$ ，理由如下：

- 设置 $t_1 = t_2 = t > 0$ 在减少了带宽的同时并没有牺牲实际的安全性（文献 [BDK+17] 中给出了详细的解释）。
- 在实际应用中对称性是一个优良的性质。

如果我们强调可证明安全，则在协议实现中令 $t_1 = 0$ 即可。在相同的参数下，若 $t_1 = 0$ ，则错误率减少，带宽（特别是 Y_1 的大小）会相应增加。

表 7.1 AKCN-MLWE 的参数选取以及跟 Kyber 的对比。KT 指一次性密钥传输的参数，PKE 是 CCA 安全的公钥加密的参数。“ $|K|$ ”指共识比特的长度，“pq-sec”指量子攻击下的比特安全性，“err”指错误率，“pk”指公钥大小（字节），“cipher”指密文大小（字节），“bw.”指带宽大小（字节）。

	$ K $	n	q	η	g	t	l	pq-sec	err	pk (B)	cipher (B)	bw. (B)
AKCN-MLWE-KT-light	256	256	7681	5	2^3	4	2	102	$2^{-36.2}$	608	704	1312
AKCN-MLWE-PKE-light	256	256	7681	5	2^3	3	2	102	$2^{-105.5}$	672	768	1440
Kyber-light	256	256	7681	5	2^3	2	2	102	2^{-145}	736	832	1568
AKCN-MLWE-KT-Recommended	256	256	7681	2	2^4	4	3	147	$2^{-67.1}$	896	992	1888
AKCN-MLWE-PKE-Recommended	256	256	7681	2	2^3	3	3	147	$2^{-166.4}$	992	1056	2048
Kyber-Recommended	256	256	7681	4	2^3	2	3	161	$2^{-142.7}$	1088	1152	2240
AKCN-MLWE-KT-Paranoid	512	512	7681	8	2^6	1	2	248	$2^{-64.1}$	1568	1920	3488
Kyber-Paranoid	256	256	7681	3	2^3	2	4	218	2^{-169}	1440	1536	2976

与 Kyber 的对比：

Kyber [BDK+17] 是基于 MLWE 和 AKC 的密钥传输协议，目前进入 NIST 后量子密码竞赛第二轮。在 Kyber 中， $Y_1 = \lfloor 2^{d_t}(AX_1 + E_1)/q \rfloor$ ， $Y_2 = \lfloor 2^{d_u}(A^T X_2 + E_2)/q \rfloor$ ， $\Sigma_2 = \lfloor qY_1/2^{d_t} \rfloor^T X_2 + E_\sigma$ ， $\Sigma_1 = X_1^T \lfloor qY_2/2^{d_u} \rfloor$ ，其中 d_t, d_u 是非负整数。Kyber 的底层 AKC 机制是 AKCN，一个细微的调整是 Kyber 中

$$\text{Rec}(\sigma_2, v, \text{params}) = \lfloor m \cdot (\lfloor qv/g \rfloor / q - \sigma_2/q) \rfloor \bmod m,$$

而 AKCN 中

$$\text{Rec}(\sigma_2, v, \text{params}) = \lfloor m \cdot (v/g - \sigma_2/q) \rfloor \bmod m.$$

注意我们的 AKCN 机制在 2016 年 11 月就已经在 arXiv 公布，并在这之前申请了专利保护。因此，Kyber 实际上存在侵权嫌疑。

本提案给出了基于 AKCN 的密钥传输协议。下面我们简单地对比 AKCN-MLWE 和 Kyber。

- AKCN-MLWE-PKE-light 和 Kyber-light 都达到 102 比特的后量子安全性，其中 AKCN-MLWE-PKE-light (Kyber-light) 的带宽是 1440 字节 (1568 字节)，错误率是 2^{-105} (2^{-145})。对于 102 比特的后量子安全性级别， 2^{-105} 的错误率已经足够小。AKCN-MLWE-PKE-light (Kyber-light) 裁剪了 3 (2) 个最低有效位，故在实际中我们的方案的安全性更高。
- AKCN-MLWE-PKE-Recommended 和 Kyber-Recommended 都超过了 128 比特的后量子安全性，其中 AKCN-MLWE-PKE (Kyber-Recommended) 在 $t = 0$ 时有 147 (161) 比特后量子安全性，错误率为 $2^{-166.4} \ll 2^{-147}$ ($2^{-142.7} \gg 2^{-161}$)。AKCN-MLWE-PKE-Recommended (Kyber-Recommended) 裁剪了 3 (2) 个最低有效位，故我们的方案的带宽更低，带宽为 2048 字节 (2240 字节)。
- AKCN-MLWE-KE-Paranoid 和 Kyber-Paranoid 则不容易比较。前者 AKCN-MLWE-KE-Paranoid 是一次性密钥协商的参数，达到了 512 比特的共识密钥，248 比特的后量子安全性，错误率为 2^{-64} ；Kyber-Paranoid 达到了 256 比特的共识密钥，218 比特的后量子安全性，错误率为 2^{-169} 。注意到经过 Grover 搜索算法 (Grover's search algorithm) 的平方加速，再加上近期对称密码学量子攻击的进展，256 比特的共识密钥理论上只

能保证大约 128 比特的量子安全性。

8 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分为两个部分。

1. 针对基于模格的 AKCN-MLWE-KT-Recommended 密钥封装算法给出了软件实现。
2. 针对基于模格的 AKCN-MLWE-PKE-Recommended 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。

另外我们同时在“参考实现”文件夹下给出了算法在 Linux 和 Windows 操作系统下的实现，这两份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。简单地说，代码在 Linux 系统下运行，比在 Windows 系统下运行快 5-14 倍左右。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）。

8.1. 基于模格的密钥封装性能分析

下面针对我们软件实现的 4 个部分，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。

8.1.1 空间消耗

下面我们将针对 AKCN-MLWE-KT-Recommended 、AKCN-MLWE-PKE-Recommended 两种算法实现给出空间性能分析。

AKCN-MLWE-KT-Recommended 密钥封装算法的空间消耗

在表 8.1 中我们给出了 AKCN-MLWE-KT-Recommended 算法主要参数一览。

表 8.1AKCN-MLWE-KT-Recommended 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)	992	288	1088	32

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 8.2 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 8.2 AKCN-MLWE-KT-Recommended 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	992	用户公钥
				sk	288	用户私钥
kem_enc	pk	992	接收方公钥	ct	1088	密文
				ss	32	共享秘密
kem_dec	ct	1088	密文	ss	32	共享秘密
	sk	288	接收方私钥			

AKCN-MLWE-PKE-Recommended 密钥封装算法的空间消耗

在表 8.3 中我们给出了 AKCN-MLWE-PKE-Recommended 算法主要参数一览。

表 8.3 AKCN-MLWE-PKE-Recommended 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)	1088	288	1152	32

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 8.4 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 8.4AKCN-MLWE-PKE-Recommended 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	1088	用户公钥
				sk	288	用户私钥
kem_enc	pk	1088	接收方公钥	ct	1152	密文
				ss	32	共享秘密
kem_dec	ct	1152	密文	ss	32	共享秘密
	sk	288	接收方私钥			

8.1.2 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两个部分，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。

Windows7 + VS2010 平台下的性能统计如表 8.5 所示。

表 8.5 Windows7 + VS2010 平台下 AKCN 密钥封装算法 API 实现的性能统计表

算法名	接口名	平均耗时（微秒）	平均 CPU 周期消耗(万)
AKCN-MLWE-KT-Recommended	密钥生成	370	125
	封装	510	176
	解封装	250	86

AKCN-MLWE-PKE-Recommended	密钥生成	367	125
	封装	520	177
	解封装	253	86

Linux 平台下的性能统计如表 8.6 所示。

表 8.6 Linux 平台下 AKCN 密钥封装算法 API 实现的性能统计表

算法名	接口名	平均耗时 (微秒)	平均 CPU 周期消 耗(万)
AKCN-MLWE-KT-Recommended	密钥生成	65	20.30
	封装	78	24.29
	解封装	19	5.89
AKCN-MLWE-PKE-Recommended	密钥生成	66	20.57
	封装	77	24.70
	解封装	18	5.89

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。

9 支持文档

Practical Key Establishment from Module Lattice

10 参考文献

- [AGKS05] 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. EUROCRYPT 2005: 128-146.
- [A17] M. R. Albrecht. On dual lattice attacks against small-secret LWE and parameter choices in HELib and SEAL. EUROCRYPT 2017: 103-129.
- [APS15] 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.
- [ADPS16] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. Post-quantum Key Exchange — A New Hope. 25th USENIX Security Symposium (USENIX Security 16), pages 327–343.
- [ADPS16b] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe. NewHope without Reconciliation. Cryptology ePrint Archive, Report 2016/1157, 2016.
- [AJS16] E. Alkim, P. Jakubeit, and P. Schwabe. A New Hope on ARM Cortex-M. Cryptology ePrint Archive, Report 2016/758, 2016.
- [ACPS09] 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.
- [BLL+15] S. Bai, A. Langlois, T. Lepoint, D. Stehlé, and R. Steinfeld. Improved Security Proofs in Lattice-Based Cryptography: Using the Rényi Divergence rather than the Statistical Distance. ASIACRYPT 2015: 3-24.
- [BPR12] A. Banerjee and C. Peikert and A. Rosen. Pseudorandom Functions and Lattices. EUROCRYPT 2012: 719-737.
- [BGL+18] S. Bhattacharya, O. Garcia-Morchon, T. Laarhoven, R. Rietman, M. Saarinen, L. Tolhuizen, and Z. Zhang. Round5: Compact and Fast Post-Quantum Public-Key Encryption. Cryptology ePrint Archive, 2018/725.
- [BBG+17] H. Baan, S. Bhattacharya, O. Garcia-Morchon, R. Rietman, L. Tolhuizen, J.L. Torre-Arce, and Z. Zhang. Round2: KEM and PKE based on GLWR. Cryptology ePrint Archive, 2017/1183.
- [BGM+16] 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.
- 19
- [BCD+16] 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.
- [BCNS15] 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, 553-570.
- [BDK+17] J. W. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, D. Stehlé. CRYSTALS-Kyber: a CCA-Secure Module-lattice-based

- KEM. Available from: <https://pq-crystals.org/> Preliminary version appears at Euro S&P 2018.
- [CZZ18] R. Chen, Z. Zhang and Z. Zhang. On the Hardness of the Computational Ring-LWR Problem and its Applications. ASIACRYPT 2018. Available from Cryptology ePrint Archive, 2018/536.
- [CN11] Y. Chen and P.Q. Nguyen. BKZ 2.0: Better Lattice Security Estimates. ASIACRYPT 2011: 1-20.
- [CGZ18] L. Cheng, B. Gong, and Y. Zhao. Lattice-Based Signature from Key Consensus. Cryptology ePrint Archive, Report 2018/1180. <https://eprint.iacr.org/2018/1180>
- [CKLS16] J.H. Cheon, D. Kim, J. Lee, and Y. Song. Lizard: Cut Off the Tail! Practical PostQuantum Public-Key Encryption from LWE and LWR. Cryptology ePrint Archive, Report 2016/1126, 2016.
- [CW90] D. Coppersmith and S. Winograd. Matrix Multiplication via Arithmetic Progressions. Journal of Symbolic Computation, volume 9, issue 3, pages 251-280, 1990.
- [CDS94] R. Cramer, I. Damgrd and B. Schoenmakers. Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols. CRYPTO 1994: 174187.
- [DKRV17] J. D’Anvers, A. Karmakar, S.S. Roy, and F. Vercauteren. SABER: Mod-LWR based KEM. Proposal to NIST PQC Standardization.
- [DXL12] 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.
- [DORS08] 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.
- [DD12] L. Ducas and A. Durmus. Ring-LWE in Polynomial Rings. PKC 2012: 34-51.
- [DTV15] A. Duc, F. Tramèr, and S. Vaudenay. Better Algorithms for LWE and LWR. EUROCRYPT 2015: 173-202.
- [FS86] A. Fiat and A. Shamir. How to Prove Yourself: Practical Solutions to Identification and Signature Problems. CRYPTO 1986: 186194.
- 20
- [FO99] 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.
- [FO13] E. Fujisaki and T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Schemes. Journal of Cryptology, Volume 26, Issue 1, pages 80-101, 2013.
- [GPV08] C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for Hard Lattices and New Cryptographic Constructions. ACM STOC 2008: 197-206.
- [GS16] S. Gueron and F. Schlieker. Speeding Up R-LWE Post-Quantum Key Exchange. Cryptology ePrint Archive, Report 2016/467, 2016.
- [KLL15] M. Kaplan, G. Leurent, A. Leverrier and M. Naya-Plasencia. Quantum Differential and Linear Cryptanalysis. ArXiv Preprint: 1510.05836, 2015.

- [Kra03] H. Krawczyk. SIGMA: The ‘SIGn-and-MAC’ Approach to Authenticated Diffie-Hellman and Its Use in the IKE Protocols CRYPTO 2003: 400-425.
- [KM10] H. Kuwakado and M. Morii. Quantum Distinguisher between the 3-round Feistel Cipher and the Random Permutation. IEEE ISIT 2010: 2682-2685.
- [LP10] R. Lindner and C. Peikert. Better Key Sizes (and Attacks) for LWE-Based Encryption. CT-RSA 2011: 319-339. Also available from <http://eprint.iacr.org/2010/613>.
- [LS15] A. Langlois and D. Stehlé. Worst-case to Average-case Reductions for Module Lattices. Des. Codes Cryptography, 75(3): 565-599, 2015.
- [LPR10] V. Lyubashevsky, C. Peikert, and O. Regev. On Ideal Lattices and Learning with Errors over Rings. EUROCRYPT 2010: 1-23.
- [LPR13] V. Lyubashevsky, C. Peikert, and O. Regev. A Toolkit for Ring-LWE Cryptography. EUROCRYPT 2013: 35-54.
- [FrodoKEM] Michael Naehrig, Erdem Alkim, Joppe Bos, Leo Ducas, Karen Easterbrook, Brian LaMacchia, Patrick Longa, Ilya Mironov, Valeria Nikolaenko, Christopher Peikert, Ananth Raghunathan, and Douglas Stebila. Supporting documentation: Frodokem. Technical report, National Institute of Standards and Technology, 2017. <https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/Round-1-Submissions>.
- [NIST] NIST. Post-Quantum Cryptography Standardization. <https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization>
- [Pei09] C. Peikert. Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem. STOC 2009: 333-342.
- [Pei14] C. Peikert. Lattice Cryptography for the Internet. PQCrypto 2014: 197-219.
- [Pei16] C. Peikert. A Decade of Lattice Cryptography. In Foundations and Trends in Theoretical Computer Science, Volume 10, Issue 4, pages 283-424, 2016.
- [PVW08] C. Peikert, V. Vaikuntanathan, and B. Waters. A Framework for Efficient and Composable Oblivious Transfer. CRYPTO 2008: 554-571.
- [Pop16] A.V. Poppelmann, Cryptographic Decoding of the Leech Lattice. Cryptology ePrint Archive, Report 2016/1050, 2016.
- [PG13] T. Pöppelmann and T. Güneysu. Towards Practical Lattice-Based Public-Key Encryption on Reconfigurable Hardware. SAC 2013: 68-85.
- [Reg09] O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. Journal of the ACM (JACM), Volume 56, Issue 6, pages 34, 2009.
- [Res] E. Rescorla. The Transport Layer Security (TLS) Protocol Version 1.3. IETF RFC 8446. <https://datatracker.ietf.org/doc/rfc8446/>.
- [SBG+18] M. Saarinen, S. Bhattacharya, O. Garcia-Morchon, R. Rietman, L. Tolhuizen, and Z. Zhang. Shorter Messages and Faster Post-Quantum Encryption with Round5 on Cortex M. Cryptology ePrint Archive, 2018/723.
- [SE94] C. P. Schnorr and M. Euchner. Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems. Mathematical Programming, Volume 66, Issue 2, pages 181-199, Springer, 1994.

- [Sim02] M.K. Simon. Probability Distributions Involving Gaussian Random Variables : A Handbook for Engineers and Scientists. Springer, 2012.
- [SM16] D. Stebila and M. Mosca. Post-Quantum Key Exchange for the Internet and the Open Quantum Safe Project. Cryptology ePrint Archive, Report 2016/1017, 2016.
- [Str69] V. Strassen. Gaussian Elimination is not Optimal. NumerischeMathematik, Volume 13, Issue 4, pages 354-356, Springer, 1969. Cryptology ePrint Archive: Report 2018/309
- [ZWXZ18] Z. Zheng, X. Wang, G. Xu and C. Zhao. Error Estimation of Practical Convolution Discrete Gaussian Sampling with Rejection Sampling. Cryptology ePrint Archive: Report 2018/309.

11 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

12 未来工作

本算法提案聚焦基于 MLWE 的 CPA-安全的密钥封装。由于 CCA 安全的密钥封装和公钥加密可以从 CCA-安全的密钥封装基于公开的成熟的技术转换而得到，在本提案中我们没有提供相应的转换。在未来的版本中，我们将增加相应的 CCA 安全的密钥封装及其实现。另外，我们注意到最新版本的 Kyber 使用了更小的模数 $q = 3329$ 和优化的 NTT 技巧；Kyber 的这些技巧也同样可以应用到 SKCN-MLWE。我们计划在未来版本中应用这些技巧。