COLA: 一种基于格的高效密钥封装机制

潘彦斌，李昊宇，谢天元，刘珍，杨照民，朱熠铭

中国科学院数学与系统科学研究院

1 简介

通过对经典LWE加密体制做更进一步的压缩, 我们提出一种新的基于格的高效密钥封装机制: COLA (Crazily cOmpressed Lwe-bAsed scheme). COLA 算法设计简洁, 易于实现. 相对于经典的LWE加密体制而言, COLA算法的密钥、密文规模等可以做得更小, 因此更加高效, 且其安全性可以基于NTRU 格上的CVP问题.

值得指出的是, COLA也可以看做是[14] 中体制的多项式版本, 或者高位版本的NTRU, 因此可以期望在密钥规模、安全性及应用场景等各方面具有与NTRU 加密体制相媲美的能力.

2 预备知识

2.1 记号

我们用小写黑体字母表示向量. 对于一个向量(多项式) $\boldsymbol{v}$, 我们用$\|\boldsymbol{v}\| := \sqrt{\sum_{i=0}^{n-1} v_i^2}$, $\|\boldsymbol{v}\|_\infty := \max_i(|v_i|)$ 分别表示$\boldsymbol{v}$ 的2− 范数和无穷范数.

2.2 格和格基约化算法

在数学中, 欧式空间上的一个离散加法子群称为一个格. 特别的, 设行向量$\boldsymbol{b}_1, \cdots, \boldsymbol{b}_m \in \mathbb{R}^n$ 为$\mathbb{R}^n$ 中的线性无关向量, 记其对应的矩阵为$\boldsymbol{B}$, 则$\mathcal{L}(\boldsymbol{B}) := \{\sum_{i=1}^m x_i \boldsymbol{b}_i \mid x_i \in \mathbb{Z}, i = 1, \cdots, m\}$ 构成一个$\mathbb{R}^n$ 上的格. 我们称$\boldsymbol{B}$ 为对应格的一组格基, 并且我们记$\det \mathcal{L}(\boldsymbol{B}) := \sqrt{\det(\boldsymbol{B}\boldsymbol{B}^T)}$ 为对应格的行列式.

格基约化算法在传统密码的分析中广泛应用, 同时也是分析格密码的重要手段. 格基约化算法, 包括LLL 算法[12], BKZ 算法[16, 4] 等, 其主要想法都是将原有格基做垂直投影转化为低维格, 在低维的投影格上利用SVP求解算法求低维格的最短向量, 再扩张到高维格上. 具体来说, 以块数b 为参数的BKZ 算法调用b 维SVP求解器来求解$\pi_{i+1}(\boldsymbol{b}_i), \cdots, \pi_i(\boldsymbol{b}_j)(i \le n, j = \min(n, i+b)$ 上的最短格向量(其中$\pi_i : \mathbb{R}^n \to \mathrm{span}(\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{i-1})^\perp$ 为投影映射), 并将其扩展为高维格的一组格基, 重复该过程, 可以得到一组较短的格基.

同时, b 的大小会影响格基的质量和运行时间, 不同模型下的SVP求解算法也会有不同运行时间估计. 一般而言, 运行时间与块数b 呈指数级别关系, 即运行时间$\approx 2^{c \cdot b}$. 关于常数c, 我们有如下结果:

- Classical: 目前已知最好的经典SVP 求解算法[2] 中, $c = \log_2 \sqrt{3/2} \approx 0.292$;
- Quantum: 目前已知最好的量子SVP 求解算法[11] 中, $c = \log_2 \sqrt{13/9} \approx 0.265$;
- Plausible: 密码学家猜测将来可能会达到[1], $c = \log_2 \sqrt{4/3} \approx 0.2075$.

GSA假设. 我们记格基约化算法输出的格基为$\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n$. 令$\|\boldsymbol{b}_1\| = \delta^n \det \mathcal{L}(\boldsymbol{B})^{\frac{1}{n}}$, 我们把$\delta$ 称为Hermite 根因子. 对于δ, [3] 给出了估计$\delta \approx (\frac{b}{2\pi e}(\pi b)^{\frac{1}{b}})^{\frac{1}{2(b-1)}}$. 约化后的格基对应的Gram-Schmidt 正交化$\boldsymbol{b}_1^\star, \cdots, \boldsymbol{b}_n^\star$ 的长度近似满足等比关系$\|\boldsymbol{b}_{i+1}^\star\| = q\|\boldsymbol{b}_i^\star\|$, 因此$q \approx \delta^{-\frac{2n}{n-1}} \approx \delta^{-2}$. GSA假设并不一定对整体格基是成立的, 它有可能局部成立, 这依赖于初始输入格基的形状. 对于局部成立的GSA 假设, $q \approx \delta^{-2}$.

2.3 独立同分布的和

假设$\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n$ 是n个独立服从$\mathcal{S}$的随机变量, 简记为$\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n \xleftarrow{i.i.d.} \mathcal{S}$.

Chernoff 界是一个非常好的估计尾部误差的办法, 其最一般的形式表述为

Theorem 1 (Chernoff 界). 假设$\boldsymbol{X}$ 是一个随机变量, 则

$$Pr[\boldsymbol{X} \geq a] \leq \inf_{t>0} \frac{E[e^{t\boldsymbol{X}}]}{e^{ta}},$$
$$Pr[\boldsymbol{X} \leq a] \leq \inf_{t<0} \frac{E[e^{t\boldsymbol{X}}]}{e^{ta}}.$$

因此当$a > 0$, $\boldsymbol{E}[\boldsymbol{X}] = 0$ 时, 我们有$\Pr[|\boldsymbol{X}| \geq a] \leq 2 \cdot \inf_{t>0} \frac{E[e^{t\boldsymbol{X}}]}{e^{ta}}$.

后面我们会用到如下关于Chernoff 界的应用.

Lemma 1. 假设$\boldsymbol{X}_1, \cdots, \boldsymbol{X}_n \xleftarrow{i.i.d.} \mathcal{C}_\rho$, $\boldsymbol{X} = \sum_{i=1}^n \boldsymbol{X}_i$, $a > 0$, 则

$$Pr[\boldsymbol{X} \geq a] \leq \frac{n^n(n(1-2\rho) + \sqrt{\Delta})^n \cdot 2^a(n-a)^a\rho^a}{(n+a)^n(n-a)^n(a(1-2\rho) + \sqrt{\Delta})^a},$$

其中$\Delta = 4n^2\rho^2 - 4\rho a^2 + a^2$. 特别的, 当$\rho = \frac{1}{4}$ 时, $Pr[\boldsymbol{X} \geq a] \leq \frac{n^{2n}}{(n+a)^{n+a}(n-a)^{n-a}}$.

Proof. 利用定理1 以及$\boldsymbol{X}_i$ 独立同分布, 我们有

$$\begin{aligned}\Pr[\boldsymbol{X} \geq a] &\leq \inf_{t>0} E[e^{t\boldsymbol{X}}] \cdot e^{-ta} \\ &\leq \inf_{t>0}(1 - 2\rho + \rho e^t + \rho e^{-t})^n \cdot e^{-ta},\end{aligned}$$

对$(1 - 2\rho + \rho e^t + \rho e^{-t})^n \cdot e^{-ta}$ 求关于$t > 0$ 的最小值, 取$e^t = \frac{(1-2\rho)a + \sqrt{\Delta}}{2(n-a)\rho}$, 带入式子即有上述表达式.

3 算法描述

在算法描述中, 我们使用如下记号:

- 多项式环$R = \mathbb{Z}[x]/(x^n - 1)$, 其中$n$ 为一个奇素数.
- 对正偶数q, 我们记$R_q = R/qR$. 对任意$\boldsymbol{a} \in R_q$, 我们可以将其写成$\boldsymbol{a} = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$, 其中$a_i \in [-\frac{q}{2}, \frac{q}{2}) \cap \mathbb{Z}$, $i = 0, 1, \ldots, n-1$.
- 对于一个$n-1$次多项式$x^n - 1 = f_0 + f_1x + \cdots + f_{n-1}x^{n-1}$, 我们将其与向量$\boldsymbol{f} := (f_0, \cdots, f_{n-1})$ 做一一对应. 因此, $\boldsymbol{f}$ 既可以表示一个向量, 也可以表示该向量对应的多项式, 视情况而定.

– 我们记$H(\cdot)$ 表示一个Hash函数: $\{0,1\}^* \to \{0,1\}^n$.
– $\mathcal{U}(S)$ 表示在一个有限集S 上的均匀分布;
– $\mathcal{C}_\rho(0 < \rho < \frac{1}{2})$ 为$\{0,1,-1\}$ 上的分布，且满足$\Pr[X=1] = \Pr[X=-1] = \rho$, $\Pr[X=0] = 1-2\rho$.
– 对于一维概率分布$\mathcal{D}$, 我们记$\mathcal{D}^n$ 为每个分量独立取自于$\mathcal{D}$ 的n维概率分布.

3.1 COLA密钥封装算法

COLA密钥封装体制的密钥生成算法, 密钥封装算法和解封装算法分别如算法1, 算法2和算法3 所示. 所有多项式运算均在R中进行.

算法 1 密钥生成算法COLA.KeyGen()

输入: n, q;
输出: 公钥$\boldsymbol{h}$, 私钥$\boldsymbol{f}$.
1: $\boldsymbol{f}, \boldsymbol{g} \leftarrow \mathcal{U}(S_1)$, 使得$\boldsymbol{f}$ 在R_q 中可逆, 其中S_1是R中某些小系数多项式的集合;
2: 计算$\boldsymbol{h} = \boldsymbol{f}^{-1} \cdot (\boldsymbol{g} + \frac{q}{2}) \bmod q$;
3: 输出$pk = \boldsymbol{h}$, $sk = \boldsymbol{f}$.

算法 2 密钥封装算法COLA.KEMEnc()

输入: 公钥$\boldsymbol{h}$;
输出: 密文$\boldsymbol{c}$, 共享密钥$\boldsymbol{K}$.
1: 随机选取多项式$\boldsymbol{r}, \boldsymbol{e} \leftarrow \mathcal{U}(S_2)$, 其中$S_2$是$R$中某些小系数多项式的集合;
2: 计算密文$\boldsymbol{c} = \boldsymbol{h}\boldsymbol{r} + \boldsymbol{e} \bmod q$, 共享密钥$\boldsymbol{K} = H(\boldsymbol{h} \| \boldsymbol{r} \| \boldsymbol{c})$.
3: 输出$(\boldsymbol{c}, \boldsymbol{K})$

算法 3 解封装算法COLA.KEMDec()

输入: 密文$\boldsymbol{c}$, 公钥$\boldsymbol{h}$, 私钥$\boldsymbol{f}$;
输出: 共享密钥$\boldsymbol{K}$.
1: 计算$\boldsymbol{d} = \boldsymbol{f}\boldsymbol{c} \bmod q$;
2: **for** $i \leftarrow 0$ to $n-1$ **do**
3: **if** $-\frac{q}{4} \le d_i < \frac{q}{4}$ **then**
4: $r_i = 0$;
5: **else**
6: $r_i = 1$.
7: **end if**
8: **end for**
9: 输出共享密钥$\boldsymbol{K} = H(\boldsymbol{h} \| \boldsymbol{r} \| \boldsymbol{c})$.

解密算法正确性. 注意到

$$\begin{aligned}\boldsymbol{d} &= \boldsymbol{fc} \bmod q \\ &= \boldsymbol{fhr} + \boldsymbol{fe} \bmod q \\ &= \frac{q}{2}\boldsymbol{r} + \boldsymbol{gr} + \boldsymbol{fe} \bmod q,\end{aligned}$$

易知, 当$\|\boldsymbol{gr} + \boldsymbol{fe}\|_\infty < \frac{q}{4}$ 时, 则解密算法成功.

3.2 COLA公钥加密算法

依据上面的体制, 当随机多项式$\boldsymbol{r}$ 在所有的二元多项式集合上均匀随机选取时, 我们可以构造如下加密方案, 其密钥生成算法仍为算法1: COLA.KeyGen(), 其加解密算法如算法4 和算法5 所述.

算法 4 加密算法COLA.PKEEnc()

输入: 公钥$\boldsymbol{h}$, 消息$\boldsymbol{m} \in \{0,1\}^n$;
输出: 密文$\boldsymbol{c}$.

1: 随机选取多项式$\boldsymbol{r} \xleftarrow{\$} \{0,1\}^n$, $\boldsymbol{e} \xleftarrow{\$} \mathcal{U}(S_2)$, 其中$S_2$是$R$中某些小系数多项式的集合;
2: 计算$\boldsymbol{c}_1 = \boldsymbol{hr} + \boldsymbol{e} \bmod q$, $\boldsymbol{c}_2 = \boldsymbol{r} \oplus \boldsymbol{m}$, 其中$\boldsymbol{r} \oplus \boldsymbol{m} = \boldsymbol{r} + \boldsymbol{m} \bmod 2$;
3: 输出密文$\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$.

算法 5 解密算法COLA.PKEDec()

输入: 密文$\boldsymbol{c} = (\boldsymbol{c}_1, \boldsymbol{c}_2)$, 私钥$\boldsymbol{f}$;
输出: 消息$\boldsymbol{m}$.

1: 计算$\boldsymbol{d} = \boldsymbol{f} \cdot \boldsymbol{c}_1 \bmod q \in [-\frac{q}{2}, \frac{q}{2})$;
2: **for** $i \leftarrow 0$ to $n-1$ **do**
3: **if** $-\frac{q}{4} \le d_i < \frac{q}{4}$ **then**
4: $r_i = 0$;
5: **else**
6: $r_i = 1$.
7: **end if**
8: **end for**
9: 输出明文$\boldsymbol{m} = \boldsymbol{r} \oplus \boldsymbol{c_2}$.

解密算法正确性. 类似于COLA.KEM的解密正确性分析, 当$\|\boldsymbol{gr} + \boldsymbol{fe}\|_\infty < \frac{q}{4}$ 时, 算法可以正确恢复出$\boldsymbol{r}$, 从而正确恢复出消息$\boldsymbol{m}$.

3.3 IND-CCA2安全的COLA体制

显然, 上述体制均不具有IND-CCA2的安全性. 我们可以通过经典的Fujisaki–Okamoto变换, 例如[5]和[8], 期望将上述体制变成具有IND-CCA2 安全性的体制.

以COLA.KEM为例, 我们可以采取下列封装算法6 和解封装算法7 来期望达到IND-CCA2的安全性:

算法 6 密钥封装算法COLA.KEMEnc()

输入: 公钥$\boldsymbol{h}$;
输出: 密文$\boldsymbol{c}$, 共享密钥$\boldsymbol{K}$.
1: 随机选取多项式$\boldsymbol{r} \leftarrow \{0,1\}^n$;
2: $(\bar{\boldsymbol{K}}||\boldsymbol{e}) = G(H(\boldsymbol{h})||\boldsymbol{r})$, 其中$G()$ 为某固定的随机数发生器, $\boldsymbol{K}$, $\boldsymbol{e}$均为n维向量;
3: 计算密文$\boldsymbol{c} = \boldsymbol{hr} + \boldsymbol{e} \bmod q$, 共享密钥$\boldsymbol{K} = H(\bar{\boldsymbol{K}}\|H(\boldsymbol{c}))$.
4: 输出$(\boldsymbol{c}, \boldsymbol{K})$

算法 7 解封装算法COLA.KEMDec()

输入: 密文$\boldsymbol{c}$, 公钥$\boldsymbol{h}$, 私钥$\boldsymbol{f}$, 预先固定的秘密随机多项式$\boldsymbol{z}$;
输出: 共享密钥$\boldsymbol{K}$.
1: 计算$\boldsymbol{r}' = \text{COLA.PKEDec}(\boldsymbol{c}, \boldsymbol{f})$;
2: 计算$(\bar{\boldsymbol{K}}'||\boldsymbol{e}') = G(H(\boldsymbol{h})||\boldsymbol{r}')$;
3: **if** $\boldsymbol{c} == \boldsymbol{hr}' + \boldsymbol{e}' \bmod q$ **then**
4: 输出共享密钥$\boldsymbol{K} = H(\bar{\boldsymbol{K}}'\|H(\boldsymbol{c}))$;
5: **else**
6: 输出$\boldsymbol{K} = H(\boldsymbol{z}\|\boldsymbol{c})$.
7: **end if**.

4 设计原理

4.1 设计思路

基于LWE的公钥加密体制，一般都具有如下结构[13]：

- **密钥生成:** 按照离散高斯分布抽取短多项式$\boldsymbol{s}$, $\boldsymbol{e}$, 计算公钥$(\boldsymbol{a}, \boldsymbol{b} = \boldsymbol{as} + \boldsymbol{e})$, 私钥$\boldsymbol{s}$.
- **加密:** 按照离散高斯分布抽取短多项式$\boldsymbol{r}$, $\boldsymbol{e_a}$, $\boldsymbol{e_b}$, 计算密文$(\boldsymbol{c_1} = \boldsymbol{ar} + \boldsymbol{e_a}$, $\boldsymbol{c_2} = \boldsymbol{br} + \boldsymbol{e_b} + \mathsf{Encode}(\boldsymbol{m}))\in R_q^2$, 其中$\mathsf{Encode}(\boldsymbol{m})$为消息$\boldsymbol{m}$的某种编码.
- **解密:** 计算$\mathsf{Decode}(\boldsymbol{c_2} - \boldsymbol{c_1 s})$来恢复消息, 其中$\mathsf{Decode}()$为解码函数.

对消息$\boldsymbol{m}$最常见的编码是$\mathsf{Encode}(\boldsymbol{m}) = \frac{q}{2}\boldsymbol{m}$ [15]. 而为了减少密文规模，通常会对密文$\boldsymbol{c_2}$进行压缩. 最常见的方法是, 将$\boldsymbol{c_2}$每个系数的低位比特舍弃, 仅保留2到3个高位比特. 注意到, 舍弃的比特实际上是$\boldsymbol{br} + \boldsymbol{e_b}$的低位比特, 而且我们需要选择足够大的$q$来兼容这种压缩, 从而保证解密正确率.

我们的主要想法是, 通过选择特殊的$\boldsymbol{b} = \frac{q}{2}$, 使得对任意$\boldsymbol{r}$, $\boldsymbol{br}$的低位比特都为0, 因此可以直接被舍弃, 而不需要很大的q.

4.2 困难性假设

COLA体制的安全性, 主要基于以下问题的困难性.

- **私钥安全性与NTRU 格上的近似CVP问题.** 与原始的NTRU体制基于的困难问题类似, 我们的加密体制需要求解$(\boldsymbol{f}, \boldsymbol{g})$, 使得$\boldsymbol{hf} = \boldsymbol{g} + \frac{q}{2} \bmod q$.

记$\mathcal{L}_{\boldsymbol{h},q} = \mathcal{L}\begin{pmatrix} \boldsymbol{I}_n & \mathcal{A}(\boldsymbol{h}) \\ \boldsymbol{0} & q\boldsymbol{I}_n \end{pmatrix}$, 其中$\mathcal{A}(\boldsymbol{h}) = \begin{bmatrix} \boldsymbol{h} \\ x\boldsymbol{h} \bmod x^n - 1 \\ \cdots \\ x^{n-1}\boldsymbol{h} \bmod x^n - 1 \end{bmatrix}$, 则$(\boldsymbol{f}, \boldsymbol{g} + \frac{q}{2}) \in \mathcal{L}_{\boldsymbol{h}}$. 显然我们可以通过在$\mathcal{L}_{\boldsymbol{h},q}$ 中求解距离$(0, \cdots, 0, \frac{q}{2}, 0, \cdots, 0)$ 的最近格向量来恢复$(\boldsymbol{f}, \boldsymbol{g})$. 我们假设求解该问题是困难的.

– **消息,密钥安全性与环上的LWE问题.** 注意到: 恢复共享密钥或消息主要需要恢复出$\boldsymbol{r}$, 而从形式上看，从$\boldsymbol{c}_1 = \boldsymbol{h}\boldsymbol{r} + \boldsymbol{e} \bmod q$ 中恢复消息$\boldsymbol{r}$ 类似于环上的LWE 问题. 对于密钥关系$\boldsymbol{h}\boldsymbol{f} - \boldsymbol{g} = \frac{q}{2}$, 也可以看成环上的LWE 问题. 因此, 我们假设这两种特殊类型的LWE 问题也是困难的.

5 参数选取及解密失败概率分析

为了便于分析，我们给出COLA密钥封装算法和加密算法的参数选取, 见表1. 其中

– $\mathcal{T}(d+1, d)$ 表示所有系数在$\{-1, 0, 1\}$中的$n-1$次三元多项式的集合, 且该集合中的每个多项式都满足系数中1的个数为$d+1$, -1 的个数为d, 其余均为0;
– $\mathcal{B}(0, 1)$表示所有系数在$\{0, 1\}$中的$n-1$ 次二元多项式的集合.

表 1. 参数选取

	n	q	S_1	S_2	解密失败概率
COLA-KEM(PKE)-587	587	1024	$\mathcal{T}(196, 195)$	$\mathcal{B}(0, 1)$	2^{-251}
COLA-KEM(PKE)-1117	1117	1024	$\mathcal{T}(374, 373)$	$\mathcal{B}(0, 1)$	2^{-118}

解密失败概率分析. 由COLA.KEM的解密正确性分析可知, 解密失败概率不会超过$\Pr[\|\boldsymbol{g}\boldsymbol{r} + \boldsymbol{f}\boldsymbol{e}\|_\infty \geq \frac{q}{4}]$. 易知, 设$a = \frac{q}{4}$, 则

$$\Pr[\|\boldsymbol{g}\boldsymbol{r} + \boldsymbol{f}\boldsymbol{e}\|_\infty \geq a] \leq \sum_{i=0}^{n-1} \Pr[|(\boldsymbol{g}\boldsymbol{r} + \boldsymbol{f}\boldsymbol{e})_i| \geq a] \leq n(\Pr[(\boldsymbol{g}\boldsymbol{r} + \boldsymbol{f}\boldsymbol{e})_i \geq a] + \Pr[(\boldsymbol{f}\boldsymbol{r} + \boldsymbol{g}\boldsymbol{e})_i \leq -a]).$$

考虑$\Pr[(\boldsymbol{g}\boldsymbol{r} + \boldsymbol{f}\boldsymbol{e})_i \geq a]$, 注意到$(\boldsymbol{g}\boldsymbol{r} + \boldsymbol{f}\boldsymbol{e})_i = \sum_{k=0}^{n-1} g_{i-k \bmod n} \cdot r_k + \sum_{k=0}^{n} f_{i-k \bmod n} \cdot e_k$, 则$(\boldsymbol{g}\boldsymbol{r} + \boldsymbol{f}\boldsymbol{e})_i$服从如下分布:

$$\sum_{i=1}^{2d+2} X_i - \sum_{i=1}^{2d} Y_i = \sum_{i=1}^{2d} (X_i - Y_i) + X_{2d+1} + X_{2d+2},$$

其中$X_i, Y_i \xleftarrow{i.i.d.} \mathcal{U}(\{0, 1\})$. 注意到$\boldsymbol{X}_i - \boldsymbol{Y}_i$ 服从分布$\mathcal{C}_{\frac{1}{4}}$, $i = 1, \cdots, 2d$, $\boldsymbol{X}_{2d+1} + \boldsymbol{X}_{2d+2} - 1$ 服从$\mathcal{C}_{\frac{1}{4}}$, 因此$\boldsymbol{X}_1 + \cdots + \boldsymbol{X}_{2d+2} - \boldsymbol{Y}_1 - \cdots - \boldsymbol{Y}_{2d} - 1$ 服从$2d+1$ 个$\mathcal{C}_{\frac{1}{4}}$ 的和分布. 因此, 利用引理1, 易知对任意整数a,

$$\Pr[(\boldsymbol{g}\boldsymbol{r} + \boldsymbol{f}\boldsymbol{e})_i \geq a] \leq \frac{(2d+1)^{4d+2}}{(2d+2-a)^{2d+2-a}(a+2d)^{a+2d}}.$$

由对称性, $\Pr[(\boldsymbol{f}\boldsymbol{e} + \boldsymbol{g}\boldsymbol{r})_i \leq -a] = \Pr[(\boldsymbol{f}\boldsymbol{e} + \boldsymbol{g}\boldsymbol{r})_i \geq a + 2]$. 这样我们可以得到表1中的失败概率估计.

6 具体安全性分析

6.1 恢复私钥 — 直接的格基约化攻击

由第4.2节所述, 我们可以通过在$\mathcal{L}_{\boldsymbol{h},q}$中求解CVP问题来求解私钥$(\boldsymbol{f},\boldsymbol{g})$, 而该CVP 问题可以通过嵌入技术转化为在格$\widetilde{\mathcal{L}}=\mathcal{L}\begin{pmatrix}\boldsymbol{I}_n & \mathcal{A}(\boldsymbol{h}) & \boldsymbol{0}\\ \boldsymbol{0} & q\boldsymbol{I}_n & \boldsymbol{0}\\ \boldsymbol{0} & \frac{q}{2}\boldsymbol{e_1} & 1\end{pmatrix}$中求解最短向量(因为$(\boldsymbol{f},\boldsymbol{g},-1)\in\widetilde{\mathcal{L}}$), 其中$\boldsymbol{e_1}$为$n$ 维向量$(1,0,0,\cdots,0)$.

由BKZ2.0 的分析, 利用BKZ算法输出的最短的向量长度约为$\delta^{2n+1}\widetilde{\mathcal{L}}^{\frac{1}{2n+1}}=\delta^{2n+1}q^{\frac{n}{2n+1}}$. 由[6]中的分析, 一旦我们找到了等价密钥，意味着我们能找到一个长度小于等于$\frac{q}{2}$的向量. 因此我们用$\frac{q}{2}$ 作为目标向量的长度来估计时间复杂度的下界. 此时, 我们需要求Hermite根因子δ, 满足$\delta^{2n+1}q^{\frac{n}{2n+1}}=\frac{q}{2}$. 进而，利用$\delta$ 与块数b 之间的关系$\delta\approx(\frac{b}{2\pi e}(\pi b)^{\frac{1}{b}})^{\frac{1}{2(b-1)}}$, 我们可知求解所需的块数及相应的比特安全性, 如表5 所示.

表 2. 直接格基约化攻击下的比特安全性

体制	δ	b	classical	quantum	plausible
COLA-KEM(PKE)-587	1.002365	824	241	218	171
COLA-KEM(PKE)-1117	1.001242	1901	555	504	394

6.2 恢复私钥 — 混合攻击

注意到$(\boldsymbol{f},\boldsymbol{g}+\frac{q}{2})$的任何循环移位$(x^i\cdot\boldsymbol{f}\mod x^n-1, x^i\cdot(\boldsymbol{g}+\frac{q}{2})\mod x^n-1)$, 都落在格$\mathcal{L}_{\boldsymbol{h},q}$中. 因此我们同样可以利用中间相遇攻击的思想来降低攻击的时间复杂度. 不过, 与对NTRU的中间相遇攻击不同, 此时我们应该猜测一个移位数i. 此时, 对COLA 的这种中间相遇攻击不会优于对NTRU的中间相遇攻击.

特别地, 我们考虑[10]中提出的更有效的混合攻击, 并采取[7]中的描述和分析.

考虑$\boldsymbol{B}=\begin{bmatrix}q\boldsymbol{I}_n & \boldsymbol{0} & \boldsymbol{0}\\ \mathcal{A}(\boldsymbol{h}) & \boldsymbol{I}_n & \boldsymbol{0}\\ \frac{q}{2}\boldsymbol{e}_1 & \boldsymbol{0} & 1\end{bmatrix}$, 混合攻击的主要想法是将$\boldsymbol{B}$ 分成三部分

$$\boldsymbol{B}=\begin{bmatrix}q\boldsymbol{I}_{r_1} & \boldsymbol{0} & \boldsymbol{0}\\ * & \boldsymbol{L}_1 & \boldsymbol{0}\\ * & * & \boldsymbol{I}_{r_2}\end{bmatrix},$$

其中$\boldsymbol{L}_1\in\mathbb{Z}_q^{2N\times 2N}$. 对$\boldsymbol{L}_1$ 的行做格基约化, 并对其列做正交化, 然后做旋转变换, 可得$\boldsymbol{U}'\boldsymbol{L}_1\boldsymbol{Y}'=\boldsymbol{T}'$, 其中$\boldsymbol{U}'$ 是幺模矩阵, $\boldsymbol{T}'$ 是下三角矩阵, $\boldsymbol{Y}'$是正交阵. 再将整个格基进行对应的变换$\boldsymbol{T}=\boldsymbol{U}\cdot\boldsymbol{B}\cdot\boldsymbol{Y}$, 即:

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{I}_{r_1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{U}' & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{r_2}\end{bmatrix}\cdot\begin{bmatrix}q\boldsymbol{I}_{r_1} & \boldsymbol{0} & \boldsymbol{0}\\ * & \boldsymbol{L}_1 & \boldsymbol{0}\\ * & * & \boldsymbol{I}_{r_2}\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{I}_{r_1} & \boldsymbol{0} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{Y}' & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_{r_2}\end{bmatrix}=\begin{bmatrix}q\boldsymbol{I}_{r_1} & \boldsymbol{0} & \boldsymbol{0}\\ * & \boldsymbol{T}' & \boldsymbol{0}\\ * & * & \boldsymbol{I}_{r_2}\end{bmatrix}.$$

注意到$(\boldsymbol{g},\boldsymbol{f},-1)\boldsymbol{Y}$ 是格$\mathcal{L}(\boldsymbol{T})$ 中的一个向量.

设$\boldsymbol{T}$ 对角线上的元素分别为$q^{\alpha_1},\cdots,q^{\alpha_{2n+1}}$, 则$\alpha_1+\cdots+\alpha_{2n+1}=n$. 假设当$i\le r_1$ 时, $\alpha_i=1$; 当$i>2n+1-r_2$ 时, $\alpha_i=0$. 由[7]中的分析, 可知

$$\alpha_{r_1+1}=\frac{1}{2}+\frac{n-r_1-N}{2N}+2N\log_q(\delta),$$
$$\alpha_{2n+1-r_2}=\frac{1}{2}+\frac{n-r_1-N}{2N}-2N\log_q(\delta).$$

由[10] 中的结论, 如果$\boldsymbol{y}=\boldsymbol{u}\boldsymbol{T}+\boldsymbol{x}$, 并且$-T_{ii}/2<x_i\le T_{ii}/2$, 则对$\boldsymbol{y}$ 利用$\boldsymbol{T}$ 上的Babai最近平面算法, 即可恢复$\boldsymbol{x}$. 由于此处我们仅考虑三元向量, 由[7]中的分析, 当$\alpha_{2n+1-r_2}>\log_q(2)$ 时, 我们可以通过枚举最后r_2 个分量, 然后利用Baibai 的最近平面算法来恢复私钥. 因此总的运行时间约为约化时间与遍历时间的和, 即$2^{c\cdot b}+s^{0.5}$, 其中s为遍历集合的大小，这里我们假设最后遍历的集合是所有r_2-1长的向量(最后一个分量不用遍历)，且包含1/3 个+1, 1/3个-1, 其余为0. 注意到我们同样利用Grover 算法来加速搜索.

通过平衡约化时间和搜索时间, 我们可得该模型下的比特安全性, 如表3 所示.

表 3. 混合攻击下的比特安全性

模型	体制	$2N$	b	r_2	比特安全性
classical	COLA-KEM(PKE)-587	878	485	184	142
quantum	COLA-KEM(PKE)-587	870	504	174	134
plausible	COLA-KEM(PKE)-587	950	547	148	114
classical	COLA-KEM(PKE)-1117	1566	1063	398	311
quantum	COLA-KEM(PKE)-1117	1610	1106	376	294
plausible	COLA-KEM(PKE)-1117	1732	1210	323	252

6.3 恢复私钥和消息 — 原始(Primal)攻击.

考虑LWE 实例$(\boldsymbol{A},\boldsymbol{b}=\boldsymbol{s}\boldsymbol{A}+\boldsymbol{e} \bmod q)\in\mathbb{Z}^{n\times(m+1)}$, 我们可以构造格

$$\Lambda=\{\boldsymbol{x}\in\mathbb{Z}^{n+m+1}\mid \boldsymbol{x}\cdot\begin{bmatrix}\boldsymbol{A}\\ -\boldsymbol{I}_m\\ -\boldsymbol{b}\end{bmatrix}=0 \bmod q\}.$$

这个格的行列式为q^m, 维数为$d=n+m+1$, 且这个格有一个短向量$(\boldsymbol{s},\boldsymbol{e},1)$.

利用NewHope 的分析模型[1]: 我们可以选择$m(0\le m\le n)$ 个样本, $(\boldsymbol{s},\boldsymbol{e},1)$ 长度近似为$\sigma\sqrt{m+n}$, 其中σ 表示$\boldsymbol{s}$, $\boldsymbol{e}$ 的标准差; 利用BKZ 算法关于选取块大小b, 当且仅当$\sigma\sqrt{b}\le\delta^{2b-d-1}q^{\frac{m}{d}}$, $\delta=((\pi b)^{\frac{1}{b}}\frac{b}{2\pi e})^{\frac{1}{2(b-1)}}$, 攻击可以生效. 此时运行时间$\approx 2^{c\cdot b}$. 因此我们寻找最优的$m,b$ 的组合使得运行时间最优来估计该模型下的具体复杂度.

在我们的体制中, $\boldsymbol{A}$ 为$\mathcal{A}(\boldsymbol{h})$ 中任意m 行, 这种模型下有如下两种方式:

1. 利用$\boldsymbol{h}\boldsymbol{f}-\boldsymbol{g}=\frac{q}{2} \bmod q$ 恢复私钥, 此时方差为$\sigma^2=\frac{2}{3}$;
2. 利用$\boldsymbol{h}\boldsymbol{r}+\boldsymbol{e}=\boldsymbol{c} \bmod q$恢复$\boldsymbol{r}$, 此时方差为$\sigma^2=\frac{1}{2}$.

我们可以估计出该模型下COLA体制的比特安全性如表4 所示.

6.4 恢复私钥和消息 — 对偶(Dual) 攻击.

在对偶攻击的模型下, 考虑LWE 实例$(\boldsymbol{A}, \boldsymbol{b} = \boldsymbol{s}\boldsymbol{A} + \boldsymbol{e} \bmod q) \in \mathbb{Z}^{n\times(m+1)}$. 可以构造$d = m + n$ 维格$\Lambda = \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{Z}^{m+n} | \boldsymbol{A}\boldsymbol{x}^T = \boldsymbol{y}^T \bmod q\}$, 由[1]中的结论, 利用BKZ 算法来寻找$\Lambda$ 中的短向量$\boldsymbol{v}$ 长度约为$l = \delta^{d-1}q^{n/d}$, 区分$\boldsymbol{b}\boldsymbol{x}^T$ 和均匀分布的统计距离$\epsilon = 4\exp(-2\pi^2 l^2\sigma^2/q^2)$. 因此对于固定的块长$b$, 运行筛法$R = \max(1, 1/(\gamma\epsilon^2))$ 次可以找到$1/\epsilon^2$ 个短向量, 这里$\gamma = \sqrt{4/3}^b$ 是筛法中b 维投影格最短向量个数的估计.

类似于原始攻击, 这种模型下的攻击也有两种方法, 其比特安全性如表4 所示.

表 4. NewHope攻击模型下的比特安全性

体制	攻击模型	攻击方式	m	b	classical	quantum	plausible
COLA-KEM(PKE)-587	primal	恢复密钥	476	503	147	133	104
COLA-KEM(PKE)-587	dual	恢复密钥	475	501	146	132	103
COLA-KEM(PKE)-587	primal	恢复消息	480	480	140	127	99
COLA-KEM(PKE)-587	dual	恢复消息	490	478	140	126	99
COLA-KEM(PKE)-1117	primal	恢复密钥	822	1057	309	280	219
COLA-KEM(PKE)-1117	dual	恢复密钥	846	1050	307	278	217
COLA-KEM(PKE)-1117	primal	恢复消息	795	1014	296	268	210
COLA-KEM(PKE)-1117	dual	恢复消息	793	1008	294	267	209

7 性能分析

参数规模. 显然, COLA体制具有如下规模:

表 5. 参数规模(单位: 字节)

体制	公钥	私钥	明文	KEM 密文	PKE密文
COLA-587	734	123	73.4	734	807
COLA-1117	1396	233	139.6	1396	1536

运行效率. 我们在软件平台(Intel酷睿i5处理器, 8GB内存, macOS High Sierra 64位操作系统) 上实现了COLA算法, 但由于算法实现的原因, 运行较慢. 显而易见, 我们的算法和NTRUEncrypt算法, 在同等参数下，运行效率应该相当. 因此我们提供NTRUEncrypt算法($n = 443$，$q = 1024$)在同等平台下的运行时间(KeyGen: 1.6毫秒, Enc 0.2毫秒, Dec 0.4毫秒), 为我们COLA-587体制的效率提供参考.

8 优缺点声明

优点.

- COLA算法设计简洁, 易于实现;
- COLA算法规模小, 效率高;
- COLA算法可以看做是NTRU算法的高位推广, 即原始NTRU体制将明文信息放置在多项式系数的低位, 而COLA 算法将明文信息放置在多项式系数的高位. 因此COLA 与NTRU具有平行平等的关系. NTRU加密算法的参数选取, 安全性分析, 抵抗量子攻击的能力, 应用场景等均可以轻易地移植到COLA算法. 换言之, COLA 算法具有与NTRU 算法相媲美的能力.
  关于将COLA算法视为高位版本的NTRU, 我们简要解释如下: 注意到, 在COLA算法中我们可以直接从密文$\boldsymbol{c}$中恢复出随机多项式$\boldsymbol{r}$. 显然，我们可以将KEM算法改造成类似NTRU的加密算法，同时也可以利用NAEP变换[9] 将其转换成具有IND-CCA2的加密算法:

算法 8 公钥加密算法COLA.PKE-HNTRU-Enc()

输入: 公钥$\boldsymbol{h}$, 消息$\boldsymbol{m} \in S_2$, 其中S_2是R中某些小系数多项式的集合;
输出: 密文$\boldsymbol{c}$.

1: 随机选取多项式$\boldsymbol{e} \leftarrow \mathcal{U}(S_2)$;
2: 输出密文$\boldsymbol{c} = \boldsymbol{h}\boldsymbol{m} + \boldsymbol{e} \bmod q$.

算法 9 公钥加密算法COLA.PKE-HNTRU-Dec()

输入: 密文$\boldsymbol{c}$, 公钥$\boldsymbol{h}$, 私钥$\boldsymbol{f}$;
输出: 消息$\boldsymbol{m}$.

1: 计算$\boldsymbol{d} = \boldsymbol{f}\boldsymbol{c} \bmod q$;
2: **for** $i \leftarrow 0$ to $n-1$ **do**
3: **if** $-\frac{q}{4} \le d_i < \frac{q}{4}$ **then**
4: $m_i = 0$;
5: **else**
6: $m_i = 1$.
7: **end if**
8: **end for**
9: 输出消息$\boldsymbol{m}$.

缺点.

- 缺少理论上的可证明安全, 一种可能的解决方案是仿照[17]中的做法, 给出可证明安全, 但这样做无疑会降低COLA算法的效率.

参考文献

1. Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange-a new hope. In *USENIX Security Symposium*, pages 327–343. USENIX Association, 2016.
2. Anja Becker, Léo Ducas, Nicolas Gama, and Thijs Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving. In Robert Krauthgamer, editor, *Proceedings of the Twenty-seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, pages 10–24, Philadelphia, PA, USA, 2016. Society for Industrial and Applied Mathematics.

3. Yuanmi Chen. *Réduction de réseau et sécuritéconcrète du chiffrement complètement homomorphe*. PhD thesis, 2013. Thèse de doctorat dirigée par Nguyen, Phong-Quang Informatique Paris 7 2013.
4. Yuanmi Chen and Phong Q. Nguyen. BKZ 2.0: Better lattice security estimates. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011: 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
5. Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Annual International Cryptology Conference*, pages 537–554. Springer, 1999.
6. Nicolas Gama and Phong Q. Nguyen. Predicting lattice reduction. In Nigel Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 31–51, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
7. Jeff Hoffstein, Jill Pipher, John M Schanck, Joseph H Silverman, William Whyte, and Zhenfei Zhang. Choosing parameters for ntruencrypt. In *Cryptographers' Track at the RSA Conference*, pages 3–18. Springer, 2017.
8. Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. A modular analysis of the fujisaki-okamoto transformation. In *Theory of Cryptography Conference*, pages 341–371. Springer, 2017.
9. N Howgrave-Graham, JH Silverman, A Singer, and W Whyte NAEP. Provable security in the presence of decryption failures. *IACR Cryptology ePrint Archive*, 2003:172, 2003.
10. Nick Howgrave-Graham. A hybrid lattice-reduction and meet-in-the-middle attack against ntru. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007*, pages 150–169, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
11. Thijs Laarhoven. *Search Problems in cryptography*. PhD thesis, Eindhoven University of Technology, 2015.
12. A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, Dec 1982.
13. Richard Lindner and Chris Peikert. Better key sizes (and attacks) for lwe-based encryption. In *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, pages 319–339, 2011.
14. Yanbin Pan and Yingpu Deng. A general ntru-like framework for constructing lattice-based public-key cryptosystems. In *International Workshop on Information Security Applications*, pages 109–120. Springer, 2011.
15. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93, New York, NY, USA, 2005. ACM.
16. C. P. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66(1-3):181–199, Aug 1994.
17. Damien Stehlé and Ron Steinfeld. Making ntru as secure as worst-case problems over ideal lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 27–47. Springer, 2011.