第六届（2021 年）
全国高校密码数学挑战赛
参赛论文

赛题一：环上序列的截位还原

战队名称 Se_P3t

学　　校 杭州电子科技大学

学　　院 通信工程学院

专　　业 通信工程

参赛选手 方佳乐

指导教师 钟华

二零二一年六月

摘　要

序列密码由于其高效性在密码学中被广泛研究和应用. 对于一个密码学安全的伪随机序列而言, 除了需要满足统计学上的伪随机外, 它还需要具备一定的不可预测性, 即给出一段输出和必要参数, 还原生成器的初态是一个 NP 问题.

在本次密码学数学挑战赛中, 我们研究了环上序列的截位还原问题. 基于格基约化算法, 使用 python 语言 / SageMath 软件调用开源库 fpylll [3] 与 g6k [1], 我们取得了以下成果:

对于第一类挑战, 在已知模数和本原多项式的条件下, 我们使用 BKZ 算法和 G6K 求解 approx-SVP, 在数分钟内解决第 1-6 级挑战; 对于第 7 级挑战, 我们使用 G6K - GPU Tensor [4] 中实现的 gpu_sieve 也在两小时以内还原了初态.

对于第二类挑战, 给出模数和级数, 受 Sun. 等人[14]的研究启发, 我们提出一种基于零空间的新解法, 能够通过几次 BKZ 调用并在数秒内分别完成所有 9 级挑战.

对于第三类挑战, 只给出了模数的比特长度以及级数, 我们采用第二类挑战中的算法直接求出初态, 进而求出模数和本原多项式. 但是由于格的维数过大, 对于一台普通个人笔记本, 只能在数分钟内完成前 3 级挑战.

关键词：线性递归序列，整数剩余类环，截位序列，序列还原，格基约化算法，最短向量问题

目 录

第一章　引言

1.1 背景

伪随机序列生成器为序列密码提供了随机性，需要具备一定的不可预测性. 其中，线性同余生成器是一个典例. 基于线性同余的截位序列早在上世纪被 Stern[13]证明不是密码学安全的.

随后，线性同余生成器的一般化类型，线性递归生成器，也被广泛研究.

定义 1 (环上序列的截位还原问题). 对于一个在整数剩余类环 $Z/(m)$ 上的 n 阶本原多项式 $f(x) = x^n - c_{n-1}x^{n-1} - \cdots - c_0$，其对应的线性递归生成器定义为

$$a_{i+n} = c_{n-1}a_{i+n-1} + c_{n-2}a_{i+n-2} + \cdots + c_0 a_i \ mod \ m$$

其中 a_i 为生成器的内部状态. 令 $a_i = 2^k y_i + z_i$ 且 $z_i = LSB_k(a_i) = a_i \ mod \ 2^k$，那么该生成器的截位高 $bitlength(m) - k$ 比特的输出为 y_i. 环上序列的截位还原问题定义为给出生成器的一段长为 d 的连续的截位输出 $y_0, y_1, \ldots, y_{d-1}$，求出初态 $a_0, \ldots, a_{n-1}$.

Sun 等人 [14] 基于 Stern [13] 对于线性同余生成器的攻击方法，将其拓展至高阶本原多项式，给出了环上序列截位还原问题的解决方法.

1.2 赛题分析

本届挑战赛赛题一主要围绕环上序列的截位还原问题，其中包含三类挑战，每类挑战分为 9 级. 三类挑战类似 Sun 等人 [14] 的主要研究结果: 未知初态，未知本原多项式和未知模数.

对于第一类挑战, Sun 等人 [14, Section 3.5] 将生成器内部状态之间的关系转换成一组线性同余方程组，并考虑使用 Frieze 等人 [5] 的理论求解初态. 虽然题目不全满足他们给出的求解条件，但是我们可以通过使用其他格基约化算法 (如 BKZ 算法) 来优化条件. 另外，这组同于方程组可以看作一个多元隐藏数问题 [7]，我们还考虑了基于傅里叶分析的方法求解. 对于一元的情形，即最初的截位线性同余生成器，使用第 9 级的参数(截取位数，模数及输出个数)，我们确实能在很短的时间内一次还原部分高位比特；但是对于多元的情形，由于隐藏数的乘数的选取属于一特定子群，我们无法使用此类方法还原[6, Section 6.2].

对于第二类挑战，题目还未给定内部状态之间的线性关系. 根据前三级问题给出参考参数取值，我们很容易就可以发现截为输出个数恰等于参数 r，t 之和，由此可以推出给定条

件无法满足 Sun 等人提出的方法的基本条件: $L(g_i)^* = L(g_i)$. 这也意味着我们需要尝试优化他们的解法或者提出新的解法.

对于第三类挑战, 题目未明确给出模数, 而只给出了模数的比特长度. 仅仅观察给出的输出个数也能够感受到格基维数的巨大, 这对算力有很大的要求. 若非提出足够大的优化方法, 对于一台普通的个人笔记本电脑而言, 就只能经历而为了.

1.3 解题方法与结果

对于第一类挑战, 我们将其转换为 SVP 问题,使用 BKZ 算法和 G6K 求解 approx-SVP, 在数分钟内解决第 1-6 级挑战; 对于第 7 级挑战, 我们使用 gpu_sieve 也在两小时以内还原了初态.

对于第二类挑战, 我们提出了一种基于零空间的新解法, 较 Sun. 等人[<sun2020>]的解法快, 能够通过几次 BKZ 调用并在数秒内分别完成所有 9 级挑战.

对于第三类挑战,我们采用第二类挑战中的算法直接求出初态, 进而求出模数和本原多项式. 但是由于格的维数过大, 对于一台普通个人笔记本, 只能在数分钟内完成前 3 级挑战.

第二章　预备知识

所有的向量均用小写粗体字母表示，并且默认为行向量. 矩阵使用大写粗体字母表示. $||\boldsymbol{v}||$ 表示向量 $\boldsymbol{v}$ 的欧几里得范数.

2.1 格的基本概念和性质

定义 2 (格). 令 $\{\boldsymbol{b_1}, \boldsymbol{b_2}, \dots, \boldsymbol{b_n}\}$ 为 R^m 内的一组线性不相关(行)向量($m \geq n$). 那么由 $\{\boldsymbol{b_1}, \boldsymbol{b_2}, \dots, \boldsymbol{b_n}\}$ 生成的格为 $\boldsymbol{b_i}$ 的整系数线性组合，即

$$L = \{\sum_{i=1}^{n} l_i \boldsymbol{b}_i \mid l_i \in Z\}.$$

其中向量 $\boldsymbol{b_1}, \boldsymbol{b_2}, \dots, \boldsymbol{b_n}$ 称为格的基，格的秩为 n，且格的维数为 m. 如果 $n = m$，那么 L 是一个满秩格.

格 L 的基矩阵 $\boldsymbol{B}$ 是将行作为基向量 $\boldsymbol{b_i}$ 形成的 $n \times m$ 矩阵. 因此 $\boldsymbol{B}_{i,j}$ 为行向量 $\boldsymbol{b_i}$ 的第 j 个元素，且

$$L = \{\boldsymbol{xB} \mid \boldsymbol{x} \in Z^n\}.$$

格 L 的行列式是 L 的任一基矩阵 $\boldsymbol{B}$ 的基本平行六面体的体积，即

$$\det(L) = \mathrm{vol}(L) = \sqrt{\det(\boldsymbol{BB^T})}.$$

当格 L 满秩时，其行列式 $\det(L) = |\det(\boldsymbol{B})|$.

定义 3 (逐次最小值). 令 $L \subset R^m$ 是一个秩为 n 的格. L 的逐次最小值为 $\lambda_1, \dots, \lambda_n \in R$ 使得，对于 $1 \leq i \leq n$，λ_i 是满足存在 i 个线性无关的向量 $v_1, \dots, v_i \in L$ 且 $||v_j|| \leq \lambda_i, 1 \leq j \leq i$ 的最小值. 特别的，λ_1 是 L 的最短非零向量长度.

定义 4 (Minkowski 定理). 令 $L \subset R^n$ 是一个秩为 n 的格，则其逐次最小项 $\lambda_1, \lambda_2, \dots, \lambda_n$ 满足

$$\left(\prod_{i=1}^{n} \lambda_i\right)^{1/n} < \sqrt{n}\, det(L)^{1/n}.$$

推论 4.1. $\lambda_1 < \sqrt{n}\, det(L)^{1/n}$.

定义 5 (高斯启发式 [10]). 高斯启发式指出，在 R^n 中维数为 n 的一个随机格的最短非零向量长度期望为

$$\mathrm{gh}(L)=\left(\frac{\mathrm{vol}(L)}{\mathrm{vol}\left(\mathcal{B}_n(1)\right)}\right)^{1/n}=\frac{\Gamma\left(1+\frac{n}{2}\right)^{1/n}}{\sqrt{\pi}}\cdot\mathrm{vol}(L)^{1/n}\approx\sqrt{\frac{n}{2\pi e}}\cdot\det(L)^{1/n},$$

其中, $\mathcal{B}_n(r)$ 表示半径为 r 的 n-维欧几里得球.

定义 6 (Hermite 根因子). 对于维数为 n 的格的一个基矩阵 $\boldsymbol{B}$, Hermite 根因子定义为

$$\delta=\left(||\boldsymbol{b}_1||/\mathrm{vol}(\boldsymbol{B})^{1/n}\right)^{1/n}.$$

2.2 格中困难问题

令 L 是 Z^m 中的一个格.

定义 7 (最短向量问题). 给出格 L 的一个基矩阵 $\boldsymbol{B}$, 求出一个非零向量 $\boldsymbol{v}\in L$ 使得它的欧几里得范数 $||\boldsymbol{v}||$ 是最小的, 即 $\left|\left|\boldsymbol{v}\right|\right|=\lambda_1$.

定义 8 (最近向量问题). 给出格 L 的一个基矩阵 $\boldsymbol{B}$ 和一个向量 $\boldsymbol{w}\in Q^m$, 求出一个非零向量 $\boldsymbol{v}\in L$ 使得 $||\boldsymbol{w}-\boldsymbol{v}||$ 是最小的.

定义 9 (近似最短向量问题). 给定 $\gamma>1$, 给出格 L 的一个基矩阵 $\boldsymbol{B}$, 求出一个非零向量 $\boldsymbol{v}\in L$ 使得 $||\boldsymbol{v}||\leq\gamma\lambda_1$.

2.3 格基约化算法

格基约化的目的是将给定的格基转换为由短且接近正交的向量组成的"优质"基. 要实现这一点, 既需要一个合适的"优质"基的定义, 也需要一种计算满足该定义的基的有效算法.

2.3.1 LLL 算法

1982 年 Lenstra 等人提出了 LLL 算法 [9], 这在实际应用中是一个非常重要的算法.

定义 10 (LLL-规约基). 令 $\{\boldsymbol{b}_1,\dots,\boldsymbol{b}_n\}$ 是格的一组基. 设其对应的 Gram-Schmidt 正交基为 $\{\boldsymbol{b}_1^*,\dots,\boldsymbol{b}_n^*\}$, 并记 $B_i=\left|\left|\boldsymbol{b}_i^*\right|\right|^2=\langle\boldsymbol{b}_i^*,\boldsymbol{b}_i^*\rangle$. 对于 $1\leq j<i\leq n$ 令

$$\mu_{i,j}=\langle\boldsymbol{b}_i,\boldsymbol{b}_j^*\rangle/\langle\boldsymbol{b}_j^*,\boldsymbol{b}_j^*\rangle$$

为 Gram-Schmidt 过程中的系数. 给定约化参数 $1/4<\delta<1$. 我们称 $\{\boldsymbol{b}_1,\dots,\boldsymbol{b}_n\}$ 为 LLL-规约基当且仅当下列条件满足:

- (size 条件) 对于 $1\leq j<i\leq n$ 有 $\left|\mu_{i,j}\right|\leq 1/2$.
- (Lovász 条件) 对于 $2\leq i\leq n$ 有 $B_i\geq\left(\delta-\mu_{i,i-1}^2\right)B_{i-1}$.

对于一组 LLL-规约基, 它满足一下性质.

定理 1. 设 $1/4 < \delta < 1$, $\alpha = \frac{4}{4\delta-1}$,将格 $L \in R^m$ 的任一基矩阵 $\boldsymbol{B}$ 作为 LLL 算法的输入, 则其输出的格基 $\{\boldsymbol{b}_1, \dots, \boldsymbol{b}_n\}$ 满足:

1. $||\boldsymbol{b}_1|| \leq \alpha^{(n-1)/2}\lambda_1$.
2. 对于 $1 \leq j \leq i \leq n$, 有$||\boldsymbol{b}_j|| \leq \alpha^{(n-1)/2}\lambda_i$.
3. $\alpha^{(1-i)/2}\lambda_i \leq ||\boldsymbol{b}_i|| \leq \alpha^{(n-1)/2}\lambda_i$.
4. $det(L) \leq \prod_{i=1}^{n}||\boldsymbol{b}_i|| \leq \alpha^{n(n-1)/4}\, det(L)$.
5. $||\boldsymbol{b}_1|| \leq \alpha^{(n-1)/4}\, det(L)^{1/n}$.

2.3.2 BKZ 算法

1994 年, Schnorr 和 Euchner [11] 提出了一个更加实用的算法 BKZ. 该算法基于 KZ 基, 具体定义参考 [12] ,这里我们不再赘述. 给定一个维数为 n 的随机格, 对于 block-size 为 β 的 BKZ 算法输出的最短向量 $\boldsymbol{b}$ 的欧几里得范数满足

$$||\boldsymbol{b}|| \approx \delta_\beta^{n-1}\, det(L)^{1/n},$$

其中 $\delta_\beta \in \mathcal{O}\left(\beta^{1/(2\beta)}\right)$.

第三章　主要结果

已知在整数剩余类环 $Z/(m)$ 上的 n 阶本原多项式 $f(x)=x^n-c_{n-1}x^{n-1}-\cdots-c_0$，给出其对应的线性递归生成器的一段长为 d 的连续的截位高 $bitlength(m)-k$ 比特输出 $y_0,y_1,\ldots,y_{d-1}$，我们讨论在三种不同已知条件下还原初态 $a_0,\ldots,a_{n-1}$：未知初态，未知本原多项式和未知模数.

3.1 初步分析

已知

$$a_{i+n}=c_{n-1}a_{i+n-1}+c_{n-2}a_{i+n-2}+\cdots+c_0a_i\ mod\ m,$$

令

$$Q=\begin{pmatrix}0&\cdots&0&c_0\\1&\cdots&0&c_1\\\vdots&\ddots&\vdots&\vdots\\0&\cdots&1&c_{n-1}\end{pmatrix}$$

则有

$$(a_{i+j},a_{i+j+1},\ldots,a_{i+j+n-1})=(a_i,a_{i+1},\ldots,a_{i+n-1})\boldsymbol{Q}^j\ mod\ m$$

那么设 $\boldsymbol{Q}^j$ 的第一列为 $(q_{j,0},q_{j,1},\ldots,q_{j,n-1})^T$，其中 $q_{j,i}$ 是关于 $c_0,c_1,\ldots,c_{n-1}$ 的整系数多项式，我们得到内部状态的线性关系

$$a_{i+j}=\sum_{l=0}^{n-1}q_{j,l}a_{i+l}\ \mathrm{mod\ m}.$$

令上式中 $i=0$，有任一状态 a_j 关于初态 $a_0,a_1,\ldots,a_{n-1}$ 的线性同余关系式:

$$a_j=\sum_{l=0}^{n-1}q_{j,l}\,a_l\ mod\ m.$$

现给出截位高 $bitlength(m)-k$ 比特输出 y_j，即有 $a_j=2^ky_j+z_j$，代入并整理得

$$c_j\coloneqq\sum_{l=0}^{n-1}q_{j,l}\,z_l-z_j\ \mathrm{mod\ m}=2^k\left(y_j-\sum_{l=0}^{n-1}q_{j,l}\,y_l\right)mod\ m,(n\le j<d),$$

其中 $0\le z_j<2^k$.

3.2 第一类问题

对于第一类环上序列截位还原问题，我们除了生成器的内部状态未知，其他参数均给出，也就是说我们只需求出初始内部状态的低 k 比特 $z_0, z_1, \dots, z_{n-1}$. 所有九级挑战的生成器的本原多项式以及模数相同 $(m = 2^{31} - 1, n = deg(f) = 16)$.

显然，z_j $(n \le j < n)$ 关于 $z_0, z_1, \dots, z_{n-1}$ 的线性同余关系能够转换成 CVP 问题，我们可以应用 Kannan 的嵌入技术 [8] 将其转换成 SVP 问题. 整理得:

$$\begin{pmatrix} -1 \\ z_0 \\ \vdots \\ z_{n-1} \\ k_n \\ \vdots \\ k_{d-1} \end{pmatrix}^T \cdot \begin{pmatrix} 1 & & & c_n & \cdots & c_{d-1} \\ & 1 & & q_{n,0} & \cdots & q_{d-1,0} \\ & & \ddots & \vdots & \cdots & \vdots \\ & & & 1 & q_{n,n-1} & \cdots & q_{d-1,n-1} \\ & & & & m & & \\ & & & & & \ddots & \\ & & & & & & m \end{pmatrix} = \begin{pmatrix} -1 & z_0 & \dots & z_{n-1} & z_n & \dots & z_{d-1} \end{pmatrix}$$

在讨论格基的放缩因子之前，我们注意到未知变量 z_j 均大于等于零，因此我们可以先进行换元以减少目标向量的长度一比特. 令 $\hat{z_j} = z_j - 2^{k-1}$, 则有 $\hat{z_j} \in [-2^{k-1}, 2^{k-1})$.

$$\begin{pmatrix} -1 \\ z_0 \\ \vdots \\ z_{n-1} \\ k_n \\ \vdots \\ k_{d-1} \end{pmatrix}^T \cdot \begin{pmatrix} 1 & 2^{k-1} & \cdots & 2^{k-1} & c_n - 2^{k-1} & \cdots & c_{d-1} - 2^{k-1} \\ & 1 & & & q_{n,0} & \cdots & q_{d-1,0} \\ & & \ddots & & \vdots & \cdots & \vdots \\ & & & 1 & q_{n,n-1} & \cdots & q_{d-1,n-1} \\ & & & & m & & \\ & & & & & \ddots & \\ & & & & & & m \end{pmatrix} = \begin{pmatrix} -1 & \hat{z}_0 & \dots & \hat{z}_{n-1} & \hat{z}_n & \dots & \hat{z}_{d-1} \end{pmatrix}$$

随后，根据目标向量预期大小，我们可以取放缩因子为 $(m, \alpha, \dots, \alpha)$，其中 $\alpha = m/2^{k-1}$. 最终我们得出目标向量

$$\boldsymbol{v} = (-m, \alpha\hat{z_0}, \dots, \alpha\widehat{z_{d-1}})$$

以及基矩阵

$$\boldsymbol{B}=\begin{pmatrix} m & 2^{k-1}\alpha & \cdots & 2^{k-1}\alpha & \alpha(c_n-2^{k-1}) & \cdots & \alpha(c_{d-1}-2^{k-1}) \\ & \alpha & & & \alpha q_{n,0} & \cdots & \alpha q_{d-1,0} \\ & & \ddots & & \vdots & \cdots & \vdots \\ & & & \alpha & \alpha q_{n,n-1} & \cdots & \alpha q_{d-1,n-1} \\ & & & & \alpha m & & \\ & & & & & \ddots & \\ & & & & & & \alpha m \end{pmatrix}$$

设基矩阵 $\boldsymbol{B}$ 定义的格为 L，由于其行列式

$$det(L)=|\,det(\boldsymbol{B})\,|=\alpha^{d}\cdot m^{d-n+1}=m^{2d-n+1}/2^{(k-1)d},$$

根据推论 4.1，格 L 中的最短向量的欧几里得范数至多为

$$\sqrt{d+1}\,\det(L)^{1/(d+1)}=\frac{\sqrt{d+1}}{2^{(k-1)d/(d+1)}}m^{1+\frac{d-n}{d+1}}.$$

为了能够使用格基约化算法快速发现目标向量 $\boldsymbol{v}$，一个必要条件为 $\boldsymbol{v}$ 的长度需要显著小于格中其他向量. 因此，当 $\left|\left|\boldsymbol{v}\right|\right|\geq gh(L)$，即

$$d<\frac{n\,log_2 m+log_2\sqrt{2\pi e}}{log_2 m+1-k-\log_2\sqrt{2\pi e}}$$

时，理论上能够发现目标向量的概率不大 (注意到 Martin 等人 [2] 利用额外的 predicate 检验来突破这个界限，但是由于算力限制，这里我们不考虑). 在实际应用中，这个界限要宽松些.

最终，我们只需要求解一个 SVP 问题. 首先，我们对基矩阵进行 LLL 规约. 然后对于较小维数的基，我们直接完成 HKZ；如果维数过大的话，我们先完成一个小 block-size 的 BKZ 过程，然后使用 G6K [1] (或其 GPU 实现 [4]) 筛出目标向量.

3.3 第二类问题

对于第二类问题，给定某一阶数的本源多项式，我们仅知道其模数而不知道具体系数. 在本节中，我们首先回顾 Sun 等人 [14, Section 3.4] 的解法并分析其在此类问题中的限制，随后我们提出一种新的解法，能够在更短的时间内求出结果.

3.3.1 Sun 等人的解法

由于本原多项式系数未知，此解法的关键在于利用生成器的输出导出关于内部状态的新的线性关系. 令

$$\boldsymbol{y}_i = (y_i, y_{i+1}, \dots, y_{i+t-1}), i = 0,1, \dots, r-1, (n < t < r),$$

其中参数 r，t 的取值待定. 由这些向量构成的 $r \times t$ 矩阵 $\boldsymbol{Y} = \{\boldsymbol{y}_0, \boldsymbol{y}_1, \dots, \boldsymbol{y}_{r-1}\}$ 的维数至多为 t，当 r 远大于 t 时，一般来说，其维数恰为其最大值. 那么 $\boldsymbol{Y}$ 的核 $ker(\boldsymbol{Y})$ 的维数应为 $r-t$，即存在向量 $\boldsymbol{\eta} = (\eta_0, \eta_1, \dots, \eta_{r-1})$ 使得 $\sum_{i=0}^{r-1} \eta_i \boldsymbol{y}_i = \vec{0}$.

令

$$\boldsymbol{a}_i = (a_i, a_{i+1}, \dots, a_{i+t-1}), i = 0,1, \dots, r-1,$$

则有 $\boldsymbol{u} = \sum_{i=0}^{r-1} \eta_i \boldsymbol{a}_i = \sum_{i=0}^{r-1} \eta_i \boldsymbol{a}_i - 2^k \sum_{i=0}^{r-1} \eta_i \boldsymbol{y}_i = \sum_{i=0}^{r-1} \eta_i \boldsymbol{z}_i$，其中 $\boldsymbol{z}_i = (z_i, \dots, z_{i+t-1})$. 注意到任意 $\boldsymbol{a}_i$ 均属于同一个格，记为 L，则

$$L = \begin{pmatrix} 1 & & & q_{n,0} & \cdots & q_{t-1,0} \\ & \ddots & & \vdots & \cdots & \vdots \\ & & 1 & q_{n,n-1} & \cdots & q_{t-1,n-1} \\ & & & m & & \\ & & & & \ddots & \\ & & & & & m \end{pmatrix}$$

显然，$\boldsymbol{u} \in L$. 如果 $||\boldsymbol{\eta}||$ 足够小使得 $||\boldsymbol{u}|| < \lambda_1(L)$，那么 $\boldsymbol{u} = \vec{0}$.

假设现已求出 $\boldsymbol{\eta} = (\eta_0, \eta_1, \dots, \eta_{r-1})$ 满足 $\sum_{i=0}^{r-1} \eta_i a_i = 0$，即对于 $j = 0,1, \dots, t-1$，有

$$\sum_{i=0}^{r-1} \eta_i a_{j+i} = 0.$$

结合内部状态的线性关系，可以推出

$$\begin{pmatrix} a_0 & a_1 & \cdots & a_{n-1} \\ a_1 & a_2 & \cdots & a_n \\ \vdots & \vdots & \vdots & \vdots \\ a_{t-1} & a_t & \cdots & a_{t+n-2} \end{pmatrix} \cdot \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{n-1} \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \pmod{m}$$

其中 $g_i = \eta_i + \sum_{j=n}^{r-1} \eta_j q_{j,i}$. 当等式左边关于 a_i 的 $t \times n$ 矩阵中 t 取值合适时，它的零空间为空，故有 $g_i \bmod m = 0$. 也就是说，对于任一 g_i，存在一个整数 k 使得

$$\eta_i = km - \sum_{j=n}^{r-1} \eta_j q_{j,i},$$

亦即 $\boldsymbol{\eta}(i) = (\eta_i, \eta_n, \eta_{n+1}, \dots, \eta_{r-1}) \in L(g_i)$，其中维数为 $r-n+1$ 的格

$$L(g_i) = \begin{pmatrix} m & 0 & 0 & \cdots & 0 \\ -q_{n,i} & 1 & 0 & \cdots & 0 \\ -q_{n+1,i} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -q_{r-1,i} & 0 & 0 & \cdots & 1 \end{pmatrix}$$

这个解法的最后一个步骤为使用不同偏移的连续输出，找出 $r-n+1$ 个不相关的满足条件的 $\boldsymbol{\eta}(i)$，即还原出 $L(g_i)$，最终求其子格获得每个 $q_{n,i} = c_i$.

接下来我们讨论本解法在解题中的限制. 根据挑战数据前三级给出的参考参数取值，我们发现给出的输出个数 N 恰等于 $r+t-1$. 那么使用参考值仅能构成一组核 $ker(\boldsymbol{Y})$，其中至多含有 $r-t$ 个向量 $\boldsymbol{\eta}$ 满足 $\boldsymbol{u} = \vec{0}$. 由于 $r-t < r-n+1$，所以我们无法获得足够多的向量生成格 $L(g_i)$.

在提出新解法之前，我们尝试了去优化该解法. 假设我们能够获得两个满足条件的 η，我们可以求出 $2n$ 个关于 $c_0, c_1, \dots, c_{n-1}$ 的模多项式 $g_i \equiv 0\ (mod\ m)$. 随后，我们在整数上对 $n+1$ 个多项式两两计算结式，最终获得两个关于某一 c_i 的单变量多项式 f_1，f_2. 根据结式的计算，易知 $f_1(c_i) \equiv f_2(c_i) \equiv 0\ (mod\ m)$，根据实验我们发现，在模 m 的剩余环上，大概率有 $gcd(f_1, f_2) = x - c_i$. 对于其他系数 $c_j (j \neq i)$，我们可以先将 c_i 的值代入，然后类似之前解法迭代求出剩余系数. 但是，当本原多项式的阶数以及参数的增大，多项式 g_i 的项数也不断增多，加大了求解结式的难度. 经过测试，我们无法在几小时内解出第三级挑战.

3.3.2 基于零空间的新解法

假设我们已经获得了关于内部状态的新的线性关系，即对于 $j = 0,1, \dots, t-1$，有

$$\sum_{i=0}^{r-1} \eta_i a_{j+i} = 0.$$

对于 d 组符合条件的 $\boldsymbol{\eta}$，我们构造如下 $(r+t-1) \times td$ 的矩阵

$$\boldsymbol{M} = \begin{pmatrix} \eta_0^{(0)} & \cdots & \eta_0^{(d-1)} & & & & \\ \vdots & \cdots & \vdots & \ddots & & & \\ \eta_{r-1}^{(0)} & \cdots & \eta_{r-1}^{(d-1)} & \ddots & \eta_0^{(0)} & \cdots & \eta_0^{(d-1)} \\ & & & \ddots & \vdots & \cdots & \vdots \\ & & & & \eta_{r-1}^{(0)} & \cdots & \eta_{r-1}^{(d-1)} \end{pmatrix}$$

注意到向量 $\boldsymbol{y}=(y_0,y_1,\dots,y_{t+t-2})$ 和向量 $\boldsymbol{z}=(z_0,z_1,\dots,z_{r+t-2})$ 为两个共同属于矩阵 $\boldsymbol{M}$ 的左核 $ker_{\text{left}}(\boldsymbol{M})$ 的线性不相关的向量. 一般的, 当 $d>(r+t-1)/t$ 时, $\boldsymbol{M}$ 的零空间维数恰为 2.

记 $ker_{\text{left}}(\boldsymbol{M})$ 生成的格为 L, 并设格 L 经过 LLL 规约后的维数为 2 的格基为 $\boldsymbol{B}$. 设两向量最大元素的比特长度分别为 zbits 和 ybits, 我们分如下四种情形讨论:

如果 $||\boldsymbol{z}||<||\boldsymbol{y}||$, 一般的, 即当 zbits < ybits 时, $\boldsymbol{B}$ 的最短向量即为所求的 $\boldsymbol{z}$;

当 ybits ≤ zbits < ybits + 2 时, 我们对两向量进行代换(保证长度短且不相关), 取 $\boldsymbol{y}'=(2y_0+1,\dots,2y_{r+t-2}+1)$, $\boldsymbol{z}'=\left(z_0-2^{\text{zbits}-1},\dots,z_{r+t-1}-2^{\text{zbits}-1}\right)$, 我们有$||\boldsymbol{y}'||_\infty=2^{\text{ybits}+1}$, $||\boldsymbol{z}'||_\infty=2^{\text{zbits}-1}$, 故 $||\boldsymbol{z}'||<||\boldsymbol{y}'||$, 我们仍可以直接获得初始状态;

当 zbits > ybits + 2 时, 有 $||\boldsymbol{y}||<||\boldsymbol{z}||$, $||\boldsymbol{y}'||<||\boldsymbol{z}'||$, 我们分别对代换前后进行 LLL 规约, 由于 LLL 输出接近正交的向量, 我们可以得到 $\boldsymbol{y}$, $\boldsymbol{z}+k_1\boldsymbol{y}$ 和 $\boldsymbol{y}'$, $\boldsymbol{z}'+k_2\boldsymbol{y}'$. 给最后一个向量加上 $\left(2^{\text{zbits}-1},\dots,2^{\text{zbits}-1}\right)$ 得到 $\boldsymbol{z}+k_2\boldsymbol{y}'$, 注意到

$$\boldsymbol{z}+k_2\boldsymbol{y}'=(\boldsymbol{z}+k_1\boldsymbol{y})-k_1\boldsymbol{y}-k_2\left(2\boldsymbol{y}+\vec{1}\right),$$

所以我们可以求出 k_1,k_2, 最终获得 $\boldsymbol{z}$;

最后, 当 zbits 恰等于 ybits + 2 时, 我们可以尝试其他代换或者试试运气. 由于挑战中并未出现该情形, 这里不做进一步讨论.

已知一定长度($\geq 2n-1$)的内部状态后, 我们可以在模 m 的剩余环上构建 n 个等式, 并求解出 $\boldsymbol{Q}$ 以及对应的本原多项式的系数.

3.4 第三类问题

在第三类问题中, 关于模数, 我们只知道其比特长度. 注意到第二类问题中提出的新解法在求解初始状态并没有使用到模数, 因此我们可以搬用之前的解法. 多次计算 GCD 直到达到给定的比特长度得到模数, 然后求解系数即可.

第四章　实验结果

本章节我们给出挑战赛的解题所需时间 (详细数据参见附录). 如无特别说明, 所有的实验均在一台个人笔记本电脑上完成 (Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz). 我们主要使用 python 语言 / SageMath 软件调用开源库 fpylll [3] 与 g6k [1] 进行求解 (具体代码已在 GitHub 上开源).

4.1 第一类挑战

对于第一级和第二级挑战, 我们直接完成 HKZ 规约, 在一秒内求得结果. 对于第三级挑战, 我们使用 30 条输出, 并进行 BKZ-20 规约, 另外对于第四级和第五级, 我们取 block-size 为 40, 均在数秒内完成. 对于第六级, 我们先进行 BKZ-30 规约, 随后使用 G6K 在三分钟内筛出目标向量. 对于第七级挑战, 利用 10 线程 CPU 筛法, 连续运行十小时我们也无法获得正确的向量, 最终利用 GPU 筛法, 在一块 Tesla T4 上运行两小时左右, 我们求得了结果.

具体解题时间如下:

级数	求解方法	格基维数	解题时间
1	HKZ	19	<1s
2	HKZ	20	<1s
3	BKZ-20	31	≈1s
4	BKZ-40	52	≈1s
5	BKZ-40	74	5.39s
6	BKZ-30 + Sieve	121	2m13s
7	BKZ-20 + GPU Sieve	151	99m4s

4.2 第二类挑战

在第二类挑战中, 我们提出了新的解法, 能够利用一两次 BKZ 过程在数秒内求解出所有挑战, 具体解题时间如下:

级数	阶数	zbits (mbits=31)	r	t	BKZ: block-size	解题时间
1	2	17	30	8	20	2.35s
2	2	23	60	15	20	2.9s
3	3	21	68	17	20	3.2s
4	4	21	95	25	30	7.8s
5	5	18	85	23	30	6.7s
6	8	11	90	20	20	4.4s

级数	阶数	zbits (mbits=31)	r	t	BKZ: block-size	解题时间
7	10	11	110	26	20	9.3s
8	12	8	110	28	20	4.65s
9	14	8	128	32	32	10.0s

4.3 第三类挑战

对于三类挑战，由于维数过大和时间限制，我们只能解出前三级挑战:

级数	阶数	mbits	zbits	r	t	解题方法	解题时间
1	16	31	5	140	30	BKZ-20	30.8s
2	16	31	10	190	40	BKZ-30	1m14.8s
3	16	31	14	265	70	BKZ-30 + sieve-10-threads	5m24.0s

第五章　参考文献

[1] Martin R. Albrecht, Léo Ducas, Gottfried Herold, Elena Kirshanova, Eamonn W. Postlethwaite, and Marc Stevens. The general sieve kernel and new records in lattice reduction. *IACR Cryptol. ePrint Arch.*, 2019:89, 2019.

[2] Martin R. Albrecht and Nadia Heninger. On bounded distance decoding with predicate: Breaking the "lattice barrier" for the hidden number problem. *IACR Cryptol. ePrint Arch.*, 2020:1540, 2020.

[3] The FPLLL development team. fpylll, a Python wraper for the fplll lattice reduction library. Available at https://github.com/fplll/fpylll, 2021.

[4] Léo Ducas, Marc Stevens, and Wessel P. J. van Woerden. Advanced lattice sieving on gpus, with tensor cores. *IACR Cryptol. ePrint Arch.*, 2021:141, 2021.

[5] Alan M. Frieze, Johan Håstad, Ravi Kannan, J. C. Lagarias, and Adi Shamir. Reconstructing truncated integer variables satisfying linear congruences. *SIAM J. Comput.*, 17(2):262–280, 1988.

[6] Steven D. Galbraith, Joel Laity, and Barak Shani. Finding significant fourier coefficients: Clarifications, simplifications, applications and limitations. *IACR Cryptol. ePrint Arch.*, 2016:682, 2016.

[7] Steven D. Galbraith and Barak Shani. The multivariate hidden number problem. *IACR Cryptol. ePrint Arch.*, 2015:111, 2015.

[8] Ravi Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987.

[9] A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. 261:515 – 534, 1982.

[10] Phong Q. Nguyen and Damien Stehlé. LLL on the average. In Florian Hess, Sebastian Pauli, and Michael E. Pohst, editors, *Algorithmic Number Theory, 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006, Proceedings*, volume 4076 of *Lecture Notes in Computer Science*, pages 238–256. Springer, 2006.

[11] Claus-Peter Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66:181–199, 1994.

[12] C.P. Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2):201–224, 1987.

[13] Jacques Stern. Secret linear congruential generators are not cryptographically secure. In *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*, pages 421–426. IEEE Computer Society, 1987.

[14] Hong-Yu Sun, Xuan-Yong Zhu, and Qun-Xiong Zheng. Predicting truncated multiple recursive generators with unknown parameters. *Des. Codes Cryptogr.*, 88(6):1083–1102, 2020.

第六章　谢辞

感谢方佳乐组员对本组 Se_P3t 所做的突出贡献；感谢其室友黄安等人对我的陪伴，在这期末考试的关键时刻，是他们与我一切熬夜，一起复习；最后，感谢杭电通信信息安全社团协会 Vidar-Team 以及指导老师钟华在我学习密码学初期起着关键作用的引导和支持.

第七章　附录

7.1　各类挑战使用的运行参数及输出

7.1.1　第一类挑战

7.1.1.1　第一级

```
 % time python recover_initial_state__embedding.py 2147483647 "257,0,0,
0,1048576,0,0,0,0,0,2097152,0,0,131072,0,32768" 18 2 --category 1 --
level 1 --verbose 1 --block-size 19
SEED: 3409482587277765135

solution: (257122259, 561754033, 340567466, 370127561, 1603890151, 7897
10361, 438276282, 1205614745, 929435387, 273101854, 1330188497, 1927005
651, 70974738, 512638222, 655376420, 1799812840)

================
CPU     371%
user    2.130
system  1.530
total   0.986
```

7.1.1.2　第二级

```
 % time python recover_initial_state__embedding.py 2147483647 "257,0,0,
0,1048576,0,0,0,0,0,2097152,0,0,131072,0,32768" 19 3 --category 1 --
level 2 --verbose 1 --block-size 20
SEED: 741431373315218258

solution: (720759561, 571519362, 1471239584, 2146374199, 1943406434, 10
21387208, 911847040, 972284090, 1269846527, 2125700056, 582856260, 1326
807684, 820744516, 83875554, 1033926054, 974403091)

================
CPU     337%
user    1.903
system  1.347
total   0.963
```

7.1.1.3 第三级

```
 % time python recover_initial_state__embedding.py 2147483647 "257,0,0,
0,1048576,0,0,0,0,0,2097152,0,0,131072,0,32768" 30 14 --category 1 --
level 3 --verbose 1 --block-size 20
SEED: 6296888057838770351

solution: (2056319062, 1175325906, 34344908, 300877541, 871395921, 9530
51611, 1276817066, 429446330, 716236050, 1498148665, 419137803, 9835138
00, 877501144, 162784581, 615817441, 1532450981)

================
CPU     362%
user    2.133
system  1.565
total   1.020
```

7.1.1.4 第四级

```
 % time python recover_initial_state__embedding.py 2147483647 "257,0,0,
0,1048576,0,0,0,0,0,2097152,0,0,131072,0,32768" 51 21 --category 1 --
level 4 --verbose 1 --block-size 40
SEED: 14464219551167228876

solution: (146123803, 1660954690, 553686861, 1592631770, 2039784960, 87
4444650, 1462760700, 1629573947, 927239148, 2020986341, 2134682761, 144
0980008, 214415113, 823589071, 1178840115, 237668181)

================
CPU     311%
user    2.567
system  1.425
total   1.284
```

7.1.1.5 第五级

```
 % time python recover_initial_state__embedding.py 2147483647 "257,0,0,
0,1048576,0,0,0,0,0,2097152,0,0,131072,0,32768" 75 24 --category 1 --
level 5 --verbose 1 --block-size 40
SEED: 2380040916040384452

solution: (1683538635, 247852287, 1110454234, 2134379965, 524624671, 11
35659173, 611030817, 917303344, 67431860, 844532212, 2121384016, 163002
9172, 1311537205, 1289944654, 358213366, 2024311471)

================
CPU     194%
user    8.672
system  1.823
total   5.389
```

7.1.1.6 第六级

```
 % time python recover_initial_state__embedding.py 2147483647 "257,0,0,
0,1048576,0,0,0,0,0,2097152,0,0,131072,0,32768" 120 26 --category 1 --
level 6 --verbose 2 --block-size 30 --threads 10 --sieve
SEED: 17559565087081154795

E|v|          = 23619293351.5
E|b[0]|       = 20806089090.1
E|v|/E|b[0]| = 1.135

gh = 4328933432254747770 88.000000, goal_r0/gh = 1.289033, r0/gh = 3.069
376
  31: ↑ 31 ↓  1  T:    0.61309s, TT:    0.61312s, r0:1.03855e+21 r0/gh:
 2.39910
  34: ↑ 34 ↓  1  T:    0.67278s, TT:    1.29709s, r0:1.03855e+21 r0/gh:
 2.39910
  37: ↑ 37 ↓  1  T:    1.03919s, TT:    2.34736s, r0:1.03855e+21 r0/gh:
 2.39910
  40: ↑ 40 ↓  6  T:    1.20862s, TT:    3.56684s, r0:1.03855e+21 r0/gh:
 2.39910
  43: ↑ 43 ↓  5  T:    1.70155s, TT:    5.27941s, r0:1.03855e+21 r0/gh:
 2.39910
  46: ↑ 46 ↓  8  T:    2.22245s, TT:    7.51265s, r0:1.03855e+21 r0/gh:
 2.39910
  49: ↑ 49 ↓ 18  T:    2.69158s, TT:   10.21570s, r0:1.03855e+21 r0/gh:
 2.39910
  52: ↑ 52 ↓ 12  T:    4.12785s, TT:   14.35474s, r0:1.03855e+21 r0/gh:
 2.39910
  55: ↑ 55 ↓ 14  T:    4.34259s, TT:   18.70839s, r0:9.68850e+20 r0/gh:
 2.23808
  58: ↑ 58 ↓ 22  T:    4.46029s, TT:   23.18006s, r0:9.37393e+20 r0/gh:
 2.16541
  61: ↑ 61 ↓ 29  T:    4.61750s, TT:   27.80868s, r0:9.37393e+20 r0/gh:
 2.16541
  64: ↑ 64 ↓ 35  T:    5.35344s, TT:   33.17343s, r0:8.38208e+20 r0/gh:
 1.93629
  67: ↑ 67 ↓ 28  T:    8.15652s, TT:   41.34195s, r0:8.38208e+20 r0/gh:
 1.93629
  70: ↑ 70 ↓ 43  T:   10.30037s, TT:   51.65352s, r0:8.03424e+20 r0/gh:
 1.85594
  73: ↑ 73 ↓ 45  T:   16.68387s, TT:   68.34896s, r0:6.19472e+20 r0/gh:
 1.43100
  76: ↑ 76 ↓ 50  T:   28.59283s, TT:   96.95415s, r0:6.19472e+20 r0/gh:
 1.43100
```

7.1.1.7 第七级

```
time python recover_initial_state__embedding.py \
2147483647 \
"257,0,0,0,1048576,0,0,0,0,0,2097152,0,0,131072,0,32768" \
150 \
27 \
--category 1 \
--level 7 \
--verbose 2 \
--block-size 20 \
--threads 1 \
--sieve \
--workout/dim4free-dec 2
```

日志文件:

https://drive.google.com/file/d/1RffrzXly7PzOhaBrUOkM8JUoy2RjNe4D/view?usp=sharing

7.1.2 第二类挑战

7.1.2.1 第一级

```
 % time sage recover_coefficients__kernel.sage 2147483647 2 30 8 17 --
category 2 --level 1 --verbose 1 --block-size 20 --check
SEED: 12928939533349146289

expect_vectors: 5

kernel rank: 2
kernel' rank: 2

find k: 7 0
checking z_i: maybe
checking c_i: True

================
CPU     102%
user    2.144
system  0.261
total   2.354
```

7.1.2.2 第二级

```
 % time sage recover_coefficients__kernel.sage 2147483647 2 60 15 23 --
category 2 --level 2 --verbose 1 --block-size 20 --check
SEED: 608460742630951138

expect_vectors: 5

kernel rank: 2
kernel' rank: 2

find k: -26204 -811
checking z_i: maybe
checking c_i: True

================
CPU     101%
user    2.714
system  0.237
total   2.902
```

7.1.2.3 第三级

```
 % time sage recover_coefficients__kernel.sage 2147483647 3 68 17 21 --
category 2 --level 3 --verbose 1 --block-size 20 --check
SEED: 15922487348339387766

expect_vectors: 5

kernel rank: 2
kernel' rank: 2

find k: -1610 -50
checking z_i: maybe
checking c_i: True

================
CPU     101%
user    2.914
system  0.296
total   3.160
```

7.1.2.4 第四级

```
 % time sage recover_coefficients__kernel.sage 2147483647 4 95 25 21 --
category 2 --level 4 --verbose 1 --block-size 30 --check
SEED: 18135446848677888257

expect_vectors: 5

kernel rank: 2
kernel' rank: 2

find k: 1540 9
checking z_i: maybe
checking c_i: True

================
CPU     100%
user    7.603
system  0.266
total   7.828
```

7.1.2.5 第五级

```
 % time sage recover_coefficients__kernel.sage 2113941029 5 85 23 18 --
category 2 --level 5 --verbose 1 --block-size 30 --check
SEED: 13512413672535046818

expect_vectors: 5

kernel rank: 2
kernel' rank: 2

find k: -24 0
checking z_i: maybe
checking c_i: True

================
CPU     100%
user    6.491
system  0.256
total   6.697
```

7.1.2.6 第六级

```
 % time sage recover_coefficients__kernel.sage 2140900439 8 90 20 11 --
category 2 --level 6 --verbose 1 --block-size 20 --check
SEED: 2774189050861033585

expect_vectors: 6

kernel rank: 2

checking z_i: maybe
checking c_i: True

================
CPU     100%
user    4.257
system  0.153
total   4.403
```

7.1.2.7 第七级

```
 % time sage recover_coefficients__kernel.sage 2086596509 10 110 26 11
--category 2 --level 7 --verbose 1 --block-size 20 --check
SEED: 4768417682194502211

expect_vectors: 6

kernel rank: 2

checking z_i: maybe
checking c_i: True

================
CPU     100%
user    9.136
system  0.159
total   9.289
```

7.1.2.8 第八级

```
 % time sage recover_coefficients__kernel.sage 2123058169 12 110 28 8 -
-category 2 --level 8 --verbose 1 --block-size 20 --check
SEED: 12270745752118493764

expect_vectors: 5

kernel rank: 2

checking z_i: maybe
checking c_i: True

================
CPU     100%
user    4.516
system  0.145
total   4.652
```

7.1.2.9 第九级

```
 % time sage recover_coefficients__kernel.sage 2147483647 14 128 32 8 -
-category 2 --level 9 --verbose 1 --block-size 32 --check
SEED: 8421578719646212792

expect_vectors: 5

kernel rank: 2

checking z_i: maybe
checking c_i: True

================
CPU     100%
user    9.820
system  0.190
total   10.004
```

7.1.3 第三类挑战

7.1.3.1 第一级

```
 % time sage recover_modulus__kernel.sage 31 16 140 30 5 --category 3 -
-level 1 --verbose 1 --block-size 20 --check
SEED: 10019631545210073147
FLAGS = 01

find the kernel (rank 110)

expect_vectors: 6

find the kernel (rank 2)

kernel rank: 2

checking z_i: maybe
finding modulus
 31 bits maybe
checking c_i: True

================
CPU     278%
user    1:25.33
system  0.483
total   30.772
```

7.1.3.2 第二级

```
 % time sage recover_modulus__kernel.sage 31 16 190 40 10 --
category 3 --level 2 --verbose 1 --block-size 30 --check
SEED: 2732523400034528389
FLAGS = 01

find the kernel (rank 150)

expect_vectors: 6

find the kernel (rank 2)

kernel rank: 2

checking z_i: maybe
finding modulus
 31 bits maybe
checking c_i: True

================
CPU     318%
user    3:57.23
system  0.699
total   1:14.81
```

7.1.3.3 第三级

```
 % time sage recover_modulus__kernel.sage 31 16 265 70 14 --
category 3 --level 3 --verbose 2 --block-size 30 --threads 10 --sieve -
-timeout 300 --max-dim 80 --check
SEED: 11316827758390952469
FLAGS = 01

find the kernel (rank 195)

Loaded file 'svpchallenge-195.txt'
gh = 165086.056394, goal_r0/gh = 0.000000, r0/gh = 0.909447
'threads': 10,        :: n: 195, cputime  0.0090s, walltime:  0.0090s, f
last: -1.00, |db|: 2^0.00
expect_vectors: 5

find the kernel (rank 2)

kernel rank: 2

checking z_i: maybe
finding modulus
 31 bits maybe
checking c_i: True

================
CPU     815%
user    43:59.48
system  3.875
total   5:23.99
```

7.2 求解结果

7.2.1 第一类挑战

7.2.1.1 第一级

```
{"modulus": 2147483647, "zbits": 2, "coefficients": [257, 0, 0, 0, 1048
576, 0, 0, 0, 0, 0, 2097152, 0, 0, 131072, 0, 32768], "initial_state":
[257122259, 561754033, 340567466, 370127561, 1603890151, 789710361, 438
276282, 1205614745, 929435387, 273101854, 1330188497, 1927005651, 70974
738, 512638222, 655376420, 1799812840]}
```

7.2.1.2 第二级


```
{"modulus": 2147483647, "zbits": 3, "coefficients": [257, 0, 0, 0, 1048
576, 0, 0, 0, 0, 0, 2097152, 0, 0, 131072, 0, 32768], "initial_state":
[720759561, 571519362, 1471239584, 2146374199, 1943406434, 1021387208,
911847040, 972284090, 1269846527, 2125700056, 582856260, 1326807684, 82
0744516, 83875554, 1033926054, 974403091]}
```


7.2.1.3 第三级


```
{"modulus": 2147483647, "zbits": 14, "coefficients": [257, 0, 0, 0, 104
8576, 0, 0, 0, 0, 0, 2097152, 0, 0, 131072, 0, 32768], "initial_state":
 [2056319062, 1175325906, 34344908, 300877541, 871395921, 953051611, 12
76817066, 429446330, 716236050, 1498148665, 419137803, 983513800, 87750
1144, 162784581, 615817441, 1532450981]}
```


7.2.1.4 第四级


```
{"modulus": 2147483647, "zbits": 21, "coefficients": [257, 0, 0, 0, 104
8576, 0, 0, 0, 0, 0, 2097152, 0, 0, 131072, 0, 32768], "initial_state":
 [146123803, 1660954690, 553686861, 1592631770, 2039784960, 874444650,
1462760700, 1629573947, 927239148, 2020986341, 2134682761, 1440980008,
214415113, 823589071, 1178840115, 237668181]}
```


7.2.1.5 第五级


```
{"modulus": 2147483647, "zbits": 24, "coefficients": [257, 0, 0, 0, 104
8576, 0, 0, 0, 0, 0, 2097152, 0, 0, 131072, 0, 32768], "initial_state":
 [1683538635, 247852287, 1110454234, 2134379965, 524624671, 1135659173,
 611030817, 917303344, 67431860, 844532212, 2121384016, 1630029172, 131
1537205, 1289944654, 358213366, 2024311471]}
```


7.2.1.6 第六级


```
{"modulus": 2147483647, "zbits": 26, "coefficients": [257, 0, 0, 0, 104
8576, 0, 0, 0, 0, 0, 2097152, 0, 0, 131072, 0, 32768], "initial_state":
 [512741665, 1096178369, 808807049, 608491186, 420891056, 1682771835, 3
58966452, 55989687, 890238631, 2137448551, 2058494244, 38743896, 961704
10, 49379854, 1551639435, 1878181314]}
```

7.2.1.7 第七级


```
{"modulus": 2147483647, "zbits": 27, "coefficients": [257, 0, 0, 0, 104
8576, 0, 0, 0, 0, 0, 2097152, 0, 0, 131072, 0, 32768], "initial_state":
 [1098437589, 1969030925, 863819139, 405615144, 1088995554, 377506318,
557587767, 333944786, 426254320, 2045562272, 1027636919, 1342582532, 15
30252162, 1254788865, 1481982652, 88523237]}
```


7.2.2 第二类挑战

7.2.2.1 第一级


```
{"modulus": 2147483647, "zbits": 17, "coefficients": [1596998372, 91367
4193], "initial_state": [25583676, 1935662022]}
```


7.2.2.2 第二级


```
{"modulus": 2147483647, "zbits": 23, "coefficients": [423368878, 137551
7413], "initial_state": [1968687461, 159779378]}
```


7.2.2.3 第三级


```
{"modulus": 2147483647, "zbits": 21, "coefficients": [233454232, 712694
596, 1250324919], "initial_state": [1968687461, 159779378, 933973255]}
```


7.2.2.4 第四级


```
{"modulus": 2147483647, "zbits": 21, "coefficients": [1724886998, 28776
4120, 669496309, 29431304], "initial_state": [23214526, 63888791, 63252
6187, 515165230]}
```


7.2.2.5 第五级


```
{"modulus": 2113941029, "zbits": 18, "coefficients": [241592126, 122570
0761, 270381722, 1809937814, 545364186], "initial_state": [1265879737,
1938499214, 100295411, 164486483, 782938]}
```

7.2.2.6 第六级


```
{"modulus": 2140900439, "zbits": 11, "coefficients": [1468898684, 42920
1201, 1438911747, 1343646518, 197478154, 1760674261, 1954064960, 152159
6057], "initial_state": [2100894922, 1278709211, 1882769045, 822236456,
 503871792, 639865048, 740172666, 1225091197]}
```


7.2.2.7 第七级


```
{"modulus": 2086596509, "zbits": 11, "coefficients": [1111873952, 12101
56476, 822665602, 1221543376, 50425289, 1211476215, 1888560914, 1679919
063, 1715756131, 141246785], "initial_state": [1800774673, 1719445571,
1277627869, 633482595, 1079260842, 2060264416, 573852671, 1245793554, 1
41816054, 891093089]}
```


7.2.2.8 第八级


```
{"modulus": 2123058169, "zbits": 8, "coefficients": [1380518532, 572802
739, 397998604, 1517367287, 1838517468, 1894991963, 186761507, 16919261
63, 917271042, 1840254211, 1520485994, 1544456793], "initial_state": [1
407121795, 1964561578, 1522896740, 706799210, 1022864344, 1609032902, 9
45477963, 1469966024, 232347115, 2025154173, 653675533, 603475034]}
```


7.2.2.9 第九级


```
{"modulus": 2147483647, "zbits": 8, "coefficients": [755735009, 4351053
67, 1987422269, 141113323, 1831273687, 150474978, 1521781010, 88703098,
 2128502238, 1314935750, 1897202874, 765777736, 1257457888, 851182418],
 "initial_state": [2043273873, 1844618476, 93566804, 974010131, 1813982
016, 210537594, 1382311654, 397516522, 1569721206, 1648699957, 97174876
9, 1513210834, 1522349751, 1075882738]}
```


7.2.3 第三类挑战

7.2.3.1 第一级


```
{"modulus": 2123847813, "zbits": 5, "coefficients": [2018484748, 184518
9071, 1463275556, 1602150465, 194422107, 1025586312, 1724843525, 410393
693, 106087782, 229852172, 1293380914, 235543842, 1642599451, 250756393
, 239416449, 1593118903], "initial_state": [1422595032, 698794309, 2264
898, 2123217555, 1515919515, 2048279701, 1227019859, 1625549320, 475257
352, 1682624639, 1669847210, 510649846, 272012336, 1608958057, 13183174
28, 1804116296]}
```

7.2.3.2 第二级

{"modulus": 2146390813, "zbits": 10, "coefficients": [1709517653, 1473447434, 146866621, 1301261246, 1098198483, 1586144939, 880631859, 1190804449, 419206704, 377180855, 997067781, 668707083, 1991423106, 1200018419, 1124879071, 2081342702], "initial_state": [963105734, 789585151, 1238195227, 2028522939, 1124205863, 1618865668, 452174891, 77673612, 46901163, 1112184517, 2114462053, 259959215, 1976235589, 1517149832, 147549104, 73665604]}

7.2.3.3 第三级

{"modulus": 2147385873, "zbits": 14, "coefficients": [1433408125, 1187896517, 1493677871, 674581842, 873135996, 1326711093, 1726922573, 833159114, 897246300, 1821464147, 1671306427, 570111420, 268554940, 1034987293, 1263557877, 894951295], "initial_state": [1976346479, 1550017240, 380413913, 160326487, 379904755, 622685282, 377075770, 1527069890, 946605673, 2096385369, 1160042120, 619378670, 50681874, 1753705306, 1683069046, 1512092243]}