# 第六届 (2021 年) 全国高校密码数学挑战赛

## 赛题一： 环上序列的截位还原

方佳乐 *@Se_P3t*
指导老师: 钟华

2021 年 8 月 23 日

杭州电子科技大学 通信工程学院
信息安全协会 @Vidar-Team

# 目录

# 目录

# 基本概念

**环上序列的截位还原问题**
对于一个在整数剩余类环 $\mathbb{Z}/(m)$ 上的 $n$ 阶本原多项式 $f(x) = x^n - c_{n-1}x^{n-1} - \cdots - c_1x - c_0$,
其对应的线性递归生成器定义为

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

其中 $a_i$ 为生成器的内部状态. 令 $a_i = 2^k y_i + z_i$ 且 $z_i = \mathrm{LSB}_k(a_i) = a_i \bmod 2^k$, 那么该生成器的截位高 $\mathrm{bitlength}(m) - k$ 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 $a_0, a_1, \ldots, a_{n-1}$.

# 赛题描述

**第一类挑战** 在 模数 $m$, 本原多项式 $f(x)$ 均已知的条件下, 求解环上序列的截位还原问题.

**第二类挑战** 在 级数 $n$, 模数 $m$ 均已知的条件下, 求解环上序列的截位还原问题.

**第三类挑战** 在 级数 $n$ 和 $m$ 的比特数 均已知的条件下, 求解环上序列的截位还原问题.

# 目录

# 赛题分析: 第一类挑战

Sun 等人在 [SZZ20, Section 3.5] 中将生成器内部状态之间的关系转换成一组<u>线性同余方程组</u>, 并考虑使用 Frieze 等人 [Fri+88] 的理论求解初态.

令

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

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

# 赛题分析: 第一类挑战

Sun 等人在 [SZZ20, Section 3.5] 中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 [Fri+88] 的理论求解初态.

令

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

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

# 赛题分析: 第一类挑战

Sun 等人在 [SZZ20, Section 3.5] 中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 [Fri+88] 的理论求解初态.

令

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

则有 $\left(a_{i+j}, a_{i+j+1}, \ldots, a_{i+j+n-1}\right) = \left(a_i, a_{i+1}, \ldots, a_{i+n-1}\right) \mathbf{Q}^j \bmod m$

# 赛题分析: 第一类挑战

设 $Q^j$ 的第一列为 $(q_{j,0}, q_{j,1}, \ldots, q_{j,n-1})^T$, 有任一状态 $a_j$ 关于初态的线性同余关系式:

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

现给出截位高 bitlength$(m) - k$ 比特输出 $y_i$, 代入得:

$$c_j := 2^k \left( y_j - \sum_{l=0}^{n-1} q_{j,l} y_l \right) \bmod m \equiv \sum_{l=0}^{n-1} q_{j,l} z_l - z_j \pmod{m},$$

其中 $0 \leq z_i < 2^k$.

# 赛题分析: 第一类挑战

设 $Q^j$ 的第一列为 $(q_{j,0}, q_{j,1}, \ldots, q_{j,n-1})^T$, 有任一状态 $a_j$ 关于初态的线性同余关系式:

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

现给出截位高 bitlength $(m) - k$ 比特输出 $y_i$, 代入得:

$$c_j := 2^k \left( y_j - \sum_{l=0}^{n-1} q_{j,l} y_l \right) \bmod m \equiv \sum_{l=0}^{n-1} q_{j,l} z_l - z_j \pmod{m},$$

其中 $0 \leq z_i < 2^k$.

# 赛题分析: 第一类挑战

我们基于该线性同余方程组, 考虑了以下优化:

- 利用嵌入技术转化成最短向量问题
- 使用更高级的格基约化算法, 如 BKZ 算法
- 调用筛法 G6K 与 G6K-GPU

# 赛题分析: 第一类挑战

我们基于该线性同余方程组, 考虑了以下优化:

- 利用嵌入技术转化成最短向量问题
- 使用更高级的格基约化算法, 如 BKZ 算法
- 调用筛法 G6K 与 G6K-GPU

# 赛题分析: 第一类挑战

我们基于该线性同余方程组, 考虑了以下优化:

- 利用嵌入技术转化成最短向量问题
- 使用更高级的格基约化算法, 如 BKZ 算法
- 调用筛法 G6K 与 G6K-GPU

# 赛题分析: 第一类挑战

我们基于该线性同余方程组, 考虑了以下优化:

- 利用嵌入技术转化成最短向量问题
- 使用更高级的格基约化算法, 如 BKZ 算法
- 调用筛法 G6K 与 G6K-GPU

# 赛题分析: 第二, 三类挑战

Sun 等人的解法不适用于第二类挑战, 在其研究基础上, 我们发现了关于初态的新的线性关系, 并由此提出一种基于零空间的解法能够直接求出初态.

对于第三类挑战, 由于格基维数过大, 我们只能求出前三级挑战.

# 赛题分析: 第二, 三类挑战

Sun 等人的解法不适用于第二类挑战, 在其研究基础上, 我们发现了关于初态的新的线性关系, 并由此提出一种基于零空间的解法能够直接求出初态.

对于第三类挑战, 由于格基维数过大, 我们只能求出前三级挑战.

# 目录

# 解题思路: 第一类挑战

首先, 我们应用 Kannan 的嵌入技术 [Kan87] 将其转换成 SVP 问题:

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

$$
= \begin{pmatrix} -1 & z_0 & \ldots & z_{n-1} & z_n & \ldots & z_{d-1} \end{pmatrix}
$$

# 解题思路: 第一类挑战

首先, 我们应用 Kannan 的嵌入技术 [Kan87] 将其转换成 SVP 问题:

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

$$
= \begin{pmatrix} -1 & z_0 & \ldots & z_{n-1} & z_n & \ldots & z_{d-1} \end{pmatrix}
$$

# 解题思路: 第一类挑战

注意到目标向量中 $z_j$ 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 $\hat{z}_j \in [-2^{k-1}, 2^{k-1})$.

$$
\begin{pmatrix} -1 \\ z_0 \\ \vdots \\ z_{n-1} \\ k_n \\ \vdots \\ k_{d-1} \end{pmatrix}^T \cdot
\begin{pmatrix}
1 & 2^{k-1} & \cdots & 2^{k-1} & c_n + 2^{k-1} & \cdots & c_{d-1} + 2^{k-1} \\
 & 1 & & & q_{n,0} & \cdots & q_{d-1,0} \\
 & & \ddots & & \vdots & \cdots & \vdots \\
 & & & 1 & q_{n,n-1} & \cdots & q_{d-1,n-1} \\
 & & & & m & & \\
 & & & & & \ddots & \\
 & & & & & & m
\end{pmatrix}
$$

$$
= \begin{pmatrix} -1 & \hat{z}_0 & \ldots & \widehat{z_{n-1}} & \hat{z}_n & \ldots & \widehat{z_{d-1}} \end{pmatrix}
$$

# 解题思路: 第一类挑战

注意到目标向量中 $z_j$ 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 $\hat{z}_j \in \left[-2^{k-1}, 2^{k-1}\right)$.

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

# 解题思路: 第一类挑战

注意到目标向量中 $z_j$ 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 $\hat{z}_j \in \left[-2^{k-1}, 2^{k-1}\right)$.

$$\begin{pmatrix} -1 \\ z_0 \\ \vdots \\ z_{n-1} \\ k_n \\ \vdots \\ k_{d-1} \end{pmatrix}^T \cdot \begin{pmatrix} 1 & 2^{k-1} & \cdots & 2^{k-1} & c_n + 2^{k-1} & \cdots & c_{d-1} + 2^{k-1} \\ & 1 & & & q_{n,0} & \cdots & q_{d-1,0} \\ & & \ddots & & \vdots & \cdots & \vdots \\ & & & 1 & q_{n,n-1} & \cdots & q_{d-1,n-1} \\ & & & & m & & \\ & & & & & \ddots & \\ & & & & & & m \end{pmatrix}$$

$$= \begin{pmatrix} -1 & \hat{z}_0 & \dots & \hat{z_{n-1}} & \hat{z}_n & \dots & \hat{z_{d-1}} \end{pmatrix}$$

# 解题思路: 第一类挑战

最后, 根据目标向量预期大小, 我们可以取放缩因子为 $(m, \alpha, \ldots, \alpha)$, 其中 $\alpha = \lfloor m/2^{k-1} \rfloor$.

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

# 解题思路: 第一类挑战

最后, 根据目标向量预期大小, 我们可以取放缩因子为 $(m, \alpha, \ldots, \alpha)$, 其中 $\alpha = \lfloor m/2^{k-1} \rfloor$.

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

## 解题思路: 第二, 三类挑战

对于第二, 三类挑战, 本原多项式未被给出, 因此我们无法利用之前的线性方程组.

Sun 等人考虑通过已知的截位输出 $y_i$ 获得关于内部状态 $a_i$ 的新的线性关系 [SZZ20, Section 3.1]:

$$\sum_{i=0}^{r-1} \eta_i a_{j+i} = \sum_{i=0}^{r-1} \eta_i y_{j+i} = 0 \quad (j = 0, 1, \ldots, t-1)$$

# 解题思路: 第二, 三类挑战

对于第二, 三类挑战, 本原多项式未被给出, 因此我们无法利用之前的线性方程组.

Sun 等人考虑通过已知的截位输出 $y_i$ 获得关于内部状态 $a_i$ 的新的线性关系 [SZZ20, Section 3.1]:

$$\sum_{i=0}^{r-1} \eta_i a_{j+i} = \sum_{i=0}^{r-1} \eta_i y_{j+i} = 0 \quad (j = 0, 1, \ldots, t-1)$$

# 解题思路: 第二, 三类挑战

对于 $d$ 组符合上述条件的向量 $\overrightarrow{\eta} = \eta_0, \eta_1, \ldots, \eta_{r-1}$, 我们构造如下 $(r+t-1) \times td$ 的矩阵:

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

显然, 向量 $\overrightarrow{\mathsf{y}} = y_0, y_1, \ldots, y_{r+t-2}$ 与向量 $\overrightarrow{\mathsf{z}} = z_0, z_1, \ldots, z_{r+t-2}$ 满足

$$\overrightarrow{\mathsf{y}}\mathsf{M} = \overrightarrow{\mathsf{z}}\mathsf{M} = \overrightarrow{0}$$

即, $\overrightarrow{\mathsf{y}}, \overrightarrow{\mathsf{z}} \in \mathrm{Ker}_{\mathrm{left}}(\mathsf{M})$.

# 解题思路: 第二, 三类挑战

对于 $d$ 组符合上述条件的向量 $\overrightarrow{\eta}=\eta_0,\eta_1,\ldots,\eta_{r-1}$, 我们构造如下 $(r+t-1)\times td$ 的矩阵:

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

显然, 向量 $\overrightarrow{\mathsf{y}}=y_0,y_1,\ldots,y_{r+t-2}$ 与向量 $\overrightarrow{\mathsf{z}}=z_0,z_1,\ldots,z_{r+t-2}$ 满足

$$\overrightarrow{\mathsf{y}}\mathsf{M}=\overrightarrow{\mathsf{z}}\mathsf{M}=\overrightarrow{0}$$

即, $\overrightarrow{\mathsf{y}},\overrightarrow{\mathsf{z}}\in\mathrm{Ker}_{\mathrm{left}}(\mathsf{M})$.

## 解题思路: 第二, 三类挑战

现在我们考虑由矩阵 M 的左核 $\mathrm{Ker}_{\mathrm{left}}(M)$ 生成的格 $L$. 一般的, 当 $d > (r+t-1)/t$ 时, M 的零空间维数恰为 2, 即格 $L$ 的一组基为 $\{\overrightarrow{y}, \overrightarrow{z}\}$.

设 B 为格 $L$ 的 LLL 规约基, $\mathrm{ybit} = \max \mathrm{bitlength}(y_i)$, $\mathrm{zbit} = \max \mathrm{bitlength}(z_i)$. 我们分以下四种情况讨论:

- zbit < ybit
- ybit $\leq$ zbit < ybit + 2
- zbit > ybit + 2
- zbit = ybit + 2

## 解题思路: 第二, 三类挑战

现在我们考虑由矩阵 M 的左核 $\mathrm{Ker}_{\mathrm{left}}(\mathrm{M})$ 生成的格 $L$. 一般的, 当 $d > (r + t - 1)/t$ 时, M 的零空间维数恰为 2, 即格 $L$ 的一组基为 $\{\overrightarrow{y}, \overrightarrow{z}\}$.
设 B 为格 $L$ 的 LLL 规约基, $\mathrm{ybit} = \max \mathrm{bitlength}\,(y_i)$, $\mathrm{zbit} = \max \mathrm{bitlength}\,(z_i)$. 我们分以下四种情况讨论:

- zbit < ybit
- ybit $\leq$ zbit < ybit + 2
- zbit > ybit + 2
- zbit = ybit + 2

## 解题思路: 第二, 三类挑战

zbit < ybit

格维数为 2, 且 $\|\overrightarrow{z}\| \ll \|\overrightarrow{y}\|$, 那么经过 LLL 规约后, 有

$$B[0] = \overrightarrow{z}$$

# 解题思路: 第二, 三类挑战

ybit $\leq$ zbit < ybit + 2

对输出进行代换: 取 $\overrightarrow{y'} = 2\overrightarrow{y} + 1$, 则有 $\overrightarrow{z'} = \left(z_0 - 2^{\mathrm{zbit}-1}, \ldots, z_{r+t-1} - 2^{\mathrm{zbit}-1}\right)$, 同上求出多组 $\overrightarrow{\eta'}$ 构造矩阵 $\mathrm{M}'$ 并设 $\mathrm{Ker}_{\mathrm{left}}(\mathrm{M}')$ 生成的格 $L'$ 的 LLL 规约基为 $\mathrm{B}'$.

相应地, 我们有 $\mathrm{bitlength}\left(\left\|\overrightarrow{y'}\right\|_\infty\right) = \mathrm{ybit} + 1$, $\mathrm{bitlength}\left(\left\|\overrightarrow{z'}\right\|_\infty\right) = \mathrm{zbit} - 1$,

即 $\left\|\overrightarrow{z'}\right\| \ll \left\|\overrightarrow{y'}\right\|$, 亦即

$$\mathrm{B}'[0] = \overrightarrow{z'}$$

# 解题思路: 第二, 三类挑战

ybit $\leq$ zbit < ybit + 2

对输出进行代换: 取 $\overrightarrow{y'} = 2\overrightarrow{y} + 1$, 则有 $\overrightarrow{z'} = \left(z_0 - 2^{\mathrm{zbit}-1}, \ldots, z_{r+t-1} - 2^{\mathrm{zbit}-1}\right)$, 同上求出多组 $\overrightarrow{\eta'}$ 构造矩阵 M′ 并设 $\mathrm{Ker}_{\mathrm{left}}\left(M'\right)$ 生成的格 $L'$ 的 LLL 规约基为 B′.

相应地, 我们有 $\mathrm{bitlength}\left(\left\|\overrightarrow{y'}\right\|_\infty\right) = \mathrm{ybit} + 1$, $\mathrm{bitlength}\left(\left\|\overrightarrow{z'}\right\|_\infty\right) = \mathrm{zbit} - 1$,

即 $\left\|\overrightarrow{z'}\right\| \ll \left\|\overrightarrow{y'}\right\|$, 亦即

$$B'[0] = \overrightarrow{z'}$$

## 解题思路: 第二, 三类挑战

zbit > ybit + 2

此时 $\|\overrightarrow{y}\| < \|\overrightarrow{z}\|$, $\|\overrightarrow{y'}\| < \|\overrightarrow{z'}\|$
存在一幺模矩阵 U 使得

$$B = U \cdot \begin{pmatrix} \overrightarrow{y} \\ \overrightarrow{z} \end{pmatrix}$$

不失一般性, 我们有

$$U = \begin{pmatrix} 1 & \\ k & \pm 1 \end{pmatrix},$$

其中 $k \in \mathbb{Z}$.
综上, 我们有 $\{\overrightarrow{y}, \overrightarrow{z} + k_1\overrightarrow{y}, \overrightarrow{y'}, \overrightarrow{z'} + k_2\overrightarrow{y'}\}$

# 解题思路: 第二, 三类挑战

## zbit > ybit + 2

此时 $\|\overrightarrow{y}\| < \|\overrightarrow{z}\|, \left\|\overrightarrow{y'}\right\| < \left\|\overrightarrow{z'}\right\|$
存在一么模矩阵 U 使得

$$B = U \cdot \begin{pmatrix} \overrightarrow{y} \\ \overrightarrow{z} \end{pmatrix}$$

不失一般性, 我们有

$$U = \begin{pmatrix} 1 & \\ k & \pm 1 \end{pmatrix},$$

其中 $k \in \mathbb{Z}$.
综上, 我们有 $\{\overrightarrow{y}, \overrightarrow{z} + k_1\overrightarrow{y}, \overrightarrow{y'}, \overrightarrow{z'} + k_2\overrightarrow{y'}\}$

# 解题思路: 第二, 三类挑战

zbit > ybit + 2

此时 $\|\overrightarrow{y}\| < \|\overrightarrow{z}\|, \left\|\overrightarrow{y'}\right\| < \left\|\overrightarrow{z'}\right\|$
存在一幺模矩阵 U 使得

$$B = U \cdot \begin{pmatrix} \overrightarrow{y} \\ \overrightarrow{z} \end{pmatrix}$$

不失一般性, 我们有

$$U = \begin{pmatrix} 1 & \\ k & \pm 1 \end{pmatrix},$$

其中 $k \in \mathbb{Z}$.
综上, 我们有 $\{\overrightarrow{y}, \overrightarrow{z} + k_1\overrightarrow{y}, \overrightarrow{y'}, \overrightarrow{z'} + k_2\overrightarrow{y'}\}$

# 解题思路: 第二, 三类挑战

zbit > ybit + 2

我们有 $\{\overrightarrow{y}, \overrightarrow{z} + k_1\overrightarrow{y}, \overrightarrow{y'}, \overrightarrow{z'} + k_2\overrightarrow{y'}\}$

给最后一个向量加上 $(2^{zbit-1}, \ldots, 2^{zbit-1})$ 得到 $\overrightarrow{z} + k_2\overrightarrow{y'}$.
注意到

$$\overrightarrow{z} + k_2\overrightarrow{y'} = (\overrightarrow{z} + k_1\overrightarrow{y}) - k_1(\overrightarrow{y}) + k2(\overrightarrow{y'})$$

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

# 解题思路: 第二, 三类挑战

zbit > ybit + 2

我们有 $\{\overrightarrow{y}, \overrightarrow{z} + k_1\overrightarrow{y}, \overrightarrow{y'}, \overrightarrow{z'} + k_2\overrightarrow{y'}\}$

给最后一个向量加上 $(2^{\text{zbit}-1}, \ldots, 2^{\text{zbit}-1})$ 得到 $\overrightarrow{z} + k_2\overrightarrow{y'}$.

注意到

$$\overrightarrow{z} + k_2\overrightarrow{y'} = (\overrightarrow{z} + k_1\overrightarrow{y}) - k_1(\overrightarrow{y}) + k2(\overrightarrow{y'})$$

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

# 解题思路: 第二, 三类挑战

zbit > ybit + 2

我们有 $\{\overrightarrow{y}, \overrightarrow{z} + k_1\overrightarrow{y}, \overrightarrow{y'}, \overrightarrow{z'} + k_2\overrightarrow{y'}\}$

给最后一个向量加上 $(2^{\text{zbit}-1}, \ldots, 2^{\text{zbit}-1})$ 得到 $\overrightarrow{z} + k_2\overrightarrow{y'}$.
注意到

$$\overrightarrow{z} + k_2\overrightarrow{y'} = (\overrightarrow{z} + k_1\overrightarrow{y}) - k_1(\overrightarrow{y}) + k2(\overrightarrow{y'})$$

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

# 解题思路: 第二, 三类挑战

zbit = ybit + 2

挑战中并未出现该情形, 这里不做进一步讨论.

- 试试运气
- 尝试其他代换

# 解题思路: 第二, 三类挑战

zbit = ybit + 2

挑战中并未出现该情形, 这里不做进一步讨论.

- 试试运气
- 尝试其他代换

# 解题思路: 第二, 三类挑战

在求出初态后

- **求模数** $m$ 计算 GCD 直到比特数一致
- **求系数** $c_i$ 获得 $n$ 组关于矩阵 Q 的等式, 在 $\mathbb{Z}/(m)$ 上求解

# 解题思路: 第二, 三类挑战

在求出初态后

- **求模数** $m$ 计算 GCD 直到比特数一致
- **求系数** $c_i$ 获得 $n$ 组关于矩阵 Q 的等式, 在 $\mathbb{Z}/(m)$ 上求解

# 解题思路: 第二, 三类挑战

在求出初态后

**求模数** $m$ 计算 GCD 直到比特数一致

**求系数** $c_i$ 获得 $n$ 组关于矩阵 Q 的等式, 在 $\mathbb{Z}/(m)$ 上求解

# 目录

# 方案亮点

- 第一类挑战转换成 SVP 问题，简化了过程
- 第二，三类挑战直接求出初态，只需要一两次 BKZ，缩短了时间
- 使用了 GPU-G6K，与时俱进，大幅缩短了时间
- 代码完善，提供了测试与验证代码，方便进一步的研究
- 封装了 lattice 与 meg 类，简化了重复代码，调用方便

# 方案亮点

- 第一类挑战转换成 SVP 问题, 简化了过程
- 第二, 三类挑战直接求出初态, 只需要一两次 BKZ, 缩短了时间
- 使用了 GPU-G6K, 与时俱进, 大幅缩短了时间
- 代码完善, 提供了测试与验证代码, 方便进一步的研究
- 封装了 lattice 与 meg 类, 简化了重复代码, 调用方便

# 方案亮点

- 第一类挑战转换成 SVP 问题, 简化了过程
- 第二, 三类挑战直接求出初态, 只需要一两次 BKZ, 缩短了时间
- 使用了 GPU-G6K, 与时俱进, 大幅缩短了时间
- 代码完善, 提供了测试与验证代码, 方便进一步的研究
- 封装了 lattice 与 meg 类, 简化了重复代码, 调用方便

# 方案亮点

- 第一类挑战转换成 SVP 问题, 简化了过程
- 第二, 三类挑战直接求出初态, 只需要一两次 BKZ, 缩短了时间
- 使用了 GPU-G6K, 与时俱进, 大幅缩短了时间
- 代码完善, 提供了测试与验证代码, 方便进一步的研究
- 封装了 lattice 与 meg 类, 简化了重复代码, 调用方便

# 方案亮点

- 第一类挑战转换成 SVP 问题, 简化了过程
- 第二, 三类挑战直接求出初态, 只需要一两次 BKZ, 缩短了时间
- 使用了 GPU-G6K, 与时俱进, 大幅缩短了时间
- 代码完善, 提供了测试与验证代码, 方便进一步的研究
- 封装了 lattice 与 meg 类, 简化了重复代码, 调用方便

# 方案亮点

- 第一类挑战转换成 SVP 问题, 简化了过程
- 第二, 三类挑战直接求出初态, 只需要一两次 BKZ, 缩短了时间
- 使用了 GPU-G6K, 与时俱进, 大幅缩短了时间
- 代码完善, 提供了测试与验证代码, 方便进一步的研究
- 封装了 lattice 与 meg 类, 简化了重复代码, 调用方便

# 目录

## 第一类挑战

**第一类挑战我们主要使用** FPyLLL [tea21] **中** BKZ **算法实现以及** G6K [Alb+19] **多核** CPU **筛法**;
其中, 对于第七级挑战, 我们使用 GPU 筛法 [DSW21], 并在 Google Colab 上于两小时内求出结果.
具体解题时间如下:

| 级数 | 求解算法 | 格基维数 | 耗费时间 |
|---|---|---|---|
| 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 |

表 1: 第一类挑战解题时间

## 第一类挑战

第一类挑战我们主要使用 FPyLLL [tea21] 中 BKZ 算法实现以及 G6K [Alb+19] 多核 CPU 筛法; 其中, 对于第七级挑战, 我们使用 GPU 筛法 [DSW21], 并在 Google Colab 上于两小时内求出结果.

具体解题时间如下:

| 级数 | 求解算法 | 格基维数 | 耗费时间 |
|---|---|---|---|
| 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 |



表 1: 第一类挑战解题时间

## 第一类挑战

第一类挑战我们主要使用 FPyLLL [tea21] 中 BKZ 算法实现以及 G6K [Alb+19] 多核 CPU 筛法; 其中, 对于第七级挑战, 我们使用 GPU 筛法 [DSW21], 并在 Google Colab 上于两小时内求出结果.

具体解题时间如下:

| 级数 | 求解算法 | 格基维数 | 耗费时间 |
|---|---|---|---|
| 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 |

**表 1:** 第一类挑战解题时间

## 第二类挑战

在第二类挑战中, 我们提出了基于零空间的解法, 能够在数秒内求解出所有九级挑战.

| 级数 | 阶数 | zbits | r | t | BKZ block size | 耗费时间 |
|---|---|---|---|---|---|---|
| 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 |
| 7 | 10 | 11 | 110 | 26 | 20 | 9.3s |
| 8 | 12 | 8 | 110 | 28 | 20 | 4.65s |
| 9 | 14 | 8 | 128 | 32 | 32 | 10.0s |

表 2: 第二类挑战解题时间

## 第二类挑战

在第二类挑战中, 我们提出了基于零空间的解法, 能够在数秒内求解出所有九级挑战.

| 级数 | 阶数 | zbits | r | t | BKZ block size | 耗费时间 |
|---|---|---|---|---|---|---|
| 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 |
| 7 | 10 | 11 | 110 | 26 | 20 | 9.3s |
| 8 | 12 | 8 | 110 | 28 | 20 | 4.65s |
| 9 | 14 | 8 | 128 | 32 | 32 | 10.0s |

**表 2:** 第二类挑战解题时间

## 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 | 求解算法 | 耗费时间 |
|---|---|---|---|---|---|---|---|
| 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 | 5m24.0s |

表 3: 第三类挑战解题时间

## 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 | 求解算法 | 耗费时间 |
|---|---|---|---|---|---|---|---|
| 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 | 5m24.0s |

**表 3:** 第三类挑战解题时间

# 致谢

谢 谢!

# 目录

# 参考文献 i


[AH20] Martin R. Albrecht and Nadia Heninger. "On Bounded Distance Decoding with Predicate: Breaking the "Lattice Barrier" for the Hidden Number Problem". In: *IACR Cryptol. ePrint Arch.* 2020 (2020), p. 1540. URL: `https://eprint.iacr.org/2020/1540`.

[Alb+19] Martin R. Albrecht et al. "The General Sieve Kernel and New Records in Lattice Reduction". In: *IACR Cryptol. ePrint Arch.* 2019 (2019), p. 89. URL: `https://eprint.iacr.org/2019/089`.

[DSW21] Léo Ducas, Marc Stevens, and Wessel P. J. van Woerden. "Advanced Lattice Sieving on GPUs, with Tensor Cores". In: *IACR Cryptol. ePrint Arch.* 2021 (2021), p. 141. URL: `https://eprint.iacr.org/2021/141`.

# 参考文献 ii


[Fri+88] Alan M. Frieze et al. “Reconstructing Truncated Integer Variables Satisfying Linear Congruences”. In: *SIAM J. Comput.* 17.2 (1988), pp. 262–280. DOI: `10.1137/0217016`.

[GLS16] Steven D. Galbraith, Joel Laity, and Barak Shani. “Finding Significant Fourier Coefficients: Clarifications, Simplifications, Applications and Limitations”. In: *IACR Cryptol. ePrint Arch.* 2016 (2016), p. 682. URL: `http://eprint.iacr.org/2016/682`.

[GS15] Steven D. Galbraith and Barak Shani. “The Multivariate Hidden Number Problem”. In: *IACR Cryptol. ePrint Arch.* 2015 (2015), p. 111. URL: `http://eprint.iacr.org/2015/111`.

[Kan87] Ravi Kannan. “Minkowski’s Convex Body Theorem and Integer Programming”. In: *Math. Oper. Res.* 12.3 (1987), pp. 415–440. DOI: `10.1287/moor.12.3.415`.

# 参考文献 iii


[LLL82] A. K. Lenstra, H. W. Lenstra, and L. Lovász. "Factoring polynomials with rational coefficients". In: 261 (1982), pp. 515–534. ISSN: 0025-5831. DOI: 10.1007/bf01457454.

[NS06] Phong Q. Nguyen and Damien Stehlé. "LLL on the Average". In: *Algorithmic Number Theory, 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006, Proceedings*. Ed. by Florian Hess, Sebastian Pauli, and Michael E. Pohst. Vol. 4076. Lecture Notes in Computer Science. Springer, 2006, pp. 238–256. DOI: 10.1007/11792086_18. URL: https://doi.org/10.1007/11792086_18.

[Sch87] C.P. Schnorr. "A hierarchy of polynomial time lattice basis reduction algorithms". In: *Theoretical Computer Science* 53.2 (1987), pp. 201–224. ISSN: 0304-3975. DOI: https://doi.org/10.1016/0304-3975(87)90064-8. URL: https://www.sciencedirect.com/science/article/pii/0304397587900648.

[SE94] Claus-Peter Schnorr and M. Euchner. "Lattice basis reduction: Improved practical algorithms and solving subset sum problems". In: *Math. Program.* 66 (1994), pp. 181–199. DOI: `10.1007/BF01581144`.

[Ste87] 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*. IEEE Computer Society, 1987, pp. 421–426. DOI: `10.1109/SFCS.1987.51`.

[SZZ20] Hong-Yu Sun, Xuan-Yong Zhu, and Qun-Xiong Zheng. "Predicting truncated multiple recursive generators with unknown parameters". In: *Des. Codes Cryptogr.* 88.6 (2020), pp. 1083–1102. DOI: `10.1007/s10623-020-00729-8`.

[tea21] The FPLLL development team. "fpylll, a Python wraper for the fplll lattice reduction library". Available at `https://github.com/fplll/fpylll`. 2021. URL: `https://github.com/fplll/fpylll`.