[题目信息]:

出题人	出题时间	题目名字	题目类型	难度等级
Soreat_u	20200206	GGH	Crypto	6

[题目描述]:

```
Only LLL may not help. Nguyen said that there is a major flaw in the design of the scheme. Can you exploit it?
```

[题目考点]:

```
1. 格密码
2. 最近向量难题(CVP)、最短向量难题 (SVP)
3. LLL算法
4. Embedded Technique
5. Nguyen's Attack
```

[Flag]:

`flag{5cd9893d-2753-4e8a-a954-11de5b2d553b}`

[题目环境]:

```
SageMath 8.9
```

[题目制作过程]:

1. 在“源码”目录下，执行`sage task.py`。

[题目writeup]:

1997年，Goldreich、Goldwasser、Halevi三人受Ajtai在格难题上的研究所启发，提出了一个基于格中最近向量难题的非对称密码学算法：GGH Cryptosystem。

1999年，Nguyen发现在这个密码学算法设计中，有一个很大的缺陷，可以使攻击者从密文中获取到明文的部分信息，且可以将原来的最近向量难题转化为一个较为简单的最近向量难题。基于这个观察，Nguyen解出了设计者放在网上的5个challenge中的4个（其中有2个被设计者认为是不可能攻破的），足以证明该密码算法是broken的。

本题即基于`Nguyen's Attack`。

由于大部分CTF的crypto题中的非对称密码学算法都是围绕着RSA展开，对格密码涉及很少，因此想要解出本题则需要选手有较为丰富的格相关的数学基础。且有关于格密码的内容，网上几乎很少，只能通过阅读相关的paper来进行学习，因此本题也需要选手有相当优秀的自学能力。

> 后来在i春秋上发现了一个很不错的格相关教学视频：https://www.ichunqiu.com/course/50433
>
> 里面都是些密码学大牛的讲课，非常专业。不过可能只有数理基础及其扎实的人才能听得懂吧。:)

具体关于GGH密码算法可以参考如下内容，在此就不详细展开了：

- The GGH Cryptosystem
- Book: **An Introduction to Mathematical Cryptography**
- Book: **Mathematics of Public Key Cryptography**
- Paper: **Public-Key Cryptosystems from Reduction Problems**

下面简单介绍一下`Nguyen's Attack`：

GGH的加密过程如下：

$$\mathbf{c} = \mathbf{m}B + \mathbf{e}$$

其中，

- $\mathbf{m}$：由明文组成的一个`1×n`向量
- B：由公钥（bad basis）组成的一个`n×n`矩阵
- $\mathbf{e}$：一个`1×n`向量，其中每一项不是3就是-3
- $\mathbf{c}$：加密后的密文

我们现在已知的就只有$\mathbf{c}, B$，想要求的是这个m。

`Nguyen`观察到，如果对上式取模3，

$$\mathbf{c_3} = \mathbf{m_3}B_3 + \mathbf{e_3} \pmod 3$$

那么由于$\mathbf{e}$中每一项都是±3，所以取模3后就是$\mathbf{0}$：

$$\mathbf{c_3} = \mathbf{m_3}B_3 \pmod 3$$

因此可以求出$\mathbf{m_3}$，即明文`mod 3`后的内容。

但是`Nguyen`又观察到其中取模6是一个更好的选择，

我们先令

$$\mathbf{s} = (3, 3, \ldots, 3) \in \mathbb{Z}^n,$$

那么，$\mathbf{s} + \mathbf{e}$中每一项不是6就是0，取模6后也是$\mathbf{0}$。

所以，

$$\mathbf{c_6} = \mathbf{m_6}B_6 \pmod 6$$

这样就可以求出$\mathbf{m_6}$，即明文`mod 6`后的内容。

所以说，这个密码学算法是可以让攻击者从密文中得到部分明文的信息。

下面，我们再来推算一下，如何将这个最近向量难题（CVP）变成一个更简单的CVP。

有了$\mathbf{m_6}$之后，我们可以在等式

$$\mathbf{c} = \mathbf{m}B + \mathbf{e}$$

的两边同时减去$\mathbf{m_6}B$：

$$\mathbf{c} - \mathbf{m_6}B = (\mathbf{m} - \mathbf{m_6})B + \mathbf{e}$$

其中$\mathbf{m} - \mathbf{m_6}$中的每一项必定是6的倍数，可以写为$6 \cdot \mathbf{m}'$，且$\mathbf{m}' \in \mathbb{Z}^n$。

我们可以在上式两边同时除去6：

$$\begin{aligned} \frac{\mathbf{c} - \mathbf{m_6}B}{6} &= \frac{(\mathbf{m} - \mathbf{m_6})B}{6} + \frac{\mathbf{e}}{6}, \\ \frac{\mathbf{c} - \mathbf{m_6}B}{6} &= \mathbf{m}'B + \frac{\mathbf{e}}{6}, \\ \mathbf{c}' &= \mathbf{m}'B + \mathbf{e}' \end{aligned}$$

$\mathbf{c}'$我们可以算出来，$\mathbf{e}'$中的每一项不是$\frac{1}{2}$就是$-\frac{1}{2}$，$\mathbf{m}'$未知。

这样，我们就成功构建出了一个新的CVP，且偏差向量$\mathbf{e}'$比$\mathbf{e}$小得多，即构建出了一个更加简单的CVP。

可以利用`embedded technique`（篇幅有限，不深入，可以参考hxp的一篇wp）将这个CVP转化为SVP，再利用LLL算法求解最短向量，即可得到$\mathbf{e}'$，进而解出$\mathbf{m}'$，最后求得$\mathbf{m}$。

> 注：在式子
>
> $$\frac{\mathbf{c} - \mathbf{m_6}B}{6} = \mathbf{m}'B + \frac{\mathbf{e}}{6}$$
>
> 中可能会涉及到实数域上的运算，可以在两边同乘上2，转化为在整数域上的运算。
>
> 即，求
>
> $$\frac{\mathbf{c} - \mathbf{m_6}B}{3} = \mathbf{m}' \cdot (2B) + \frac{\mathbf{e}}{3}$$
>
> 上的CVP。

更多内容可以参考Nguyen的那篇paper：

- Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto ’97

根据这个思路，编写exp（见“解题”下的`exp.sage`），即可获取到flag。

```
$ sage exp.sage
flag{5cd9893d-2753-4e8a-a954-11de5b2d553b}
```