# babymath

一道披着逆向外壳的密码学题- -

这题逻辑其实不难。给出一个递推式：

$$a_{n+3} \equiv a_{n+2} + a_{n+1} + a_n \pmod p$$

其中，$p = 25221961025508539$，$a_0 = 0$，$a_1 = 0$，$a_2 = 1$，$a_n = 1914483117172565$，$a_{n+1} = 15349972598427081$，$a_{n+2} = 2510729161496127$，求$n$。

$p$很大，$n$可能也很大，直接算是不可能的。后来想过转化为矩阵乘法，用大步-小步法求离散对数，感觉很麻烦（其实是coding太菜），也不知道要算多久，遂放弃。

太困了，实在是hold不住，就睡觉了。。

第二天想，还是换个思路吧，就尝试求$\{a_n\}$的通项公式。

由高中所学知识，$\{a_n\}$通项公式的形式是这个样子的：

$$a_n \equiv c_1\lambda_1^n + c_2\lambda_2^n + c_3\lambda_3^n \pmod p$$

其中，$\lambda_1$，$\lambda_2$，$\lambda_3$分别为数列$\{a_n\}$的特征多项式

$$\lambda^3 - \lambda^2 - \lambda - 1 \equiv 0 \pmod p$$

的三个根。如果可以解出$\lambda_1^n$，$\lambda_2^n$，$\lambda_3^n$的其中一个，那么问题就可以转化为有限域上的离散对数问题，有现成的工具求解，不用自己造轮子了。

先用SageMath尝试在有限域$GF(p)$上分解$f(\lambda) = \lambda^3 - \lambda^2 - \lambda - 1$：

```
sage: F.<x>=GF(25221961025508539)[]
sage: (x^3-x^2-x-1).factor()
(x + 9362504514365522) * (x^2 + 15859456511143016*x + 12833828353685245)
```

emmm。。不行，中间有个$\lambda^2 + 15859456511143016 * \lambda + 12833828353685245$不可约。。

那就扩域，在$GF(p^2)$上分解（肯定能成功，因为上面不可约多项式的最高次数是2）：

```
sage: F.<x>=GF(25221961025508539^2, modulus=x^2 + 15859456511143016*x +
12833828353685245)
sage: R.<y>=PolynomialRing(F)
sage: (y^3-y^2-y-1).factor()
(y + 9362504514365522) * (y + 25221961025508538*x) * (y + x + 15859456511143016)
```

顺利得到$\lambda_1$，$\lambda_2$，$\lambda_3$。然后设法解出$c_1$，$c_2$，$c_3$：

因为

$$a_0 \equiv c_1\lambda_1^0 + c_2\lambda_2^0 + c_3\lambda_3^0 \equiv 0 \pmod p$$

$$a_1 \equiv c_1\lambda_1^1 + c_2\lambda_2^1 + c_3\lambda_3^1 \equiv 0 \pmod p$$

$$a_2 \equiv c_1\lambda_1^2 + c_2\lambda_2^2 + c_3\lambda_3^2 \equiv 1 \pmod p$$

所以有

$$\begin{bmatrix} \lambda_1^0 & \lambda_2^0 & \lambda_3^0 \\ \lambda_1^1 & \lambda_2^1 & \lambda_3^1 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \equiv \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \pmod p$$

有了$c_1$，$c_2$，$c_3$，现在可以解出$\lambda_1^n$，$\lambda_2^n$，$\lambda_3^n$了：

因为

$$a_n \equiv c_1\lambda_1^n + c_2\lambda_2^n + c_3\lambda_3^n \pmod p$$

$$a_{n+1} \equiv c_1\lambda_1^{n+1} + c_2\lambda_2^{n+1} + c_3\lambda_3^{n+1} \pmod p$$

$$a_{n+2} \equiv c_1\lambda_1^{n+2} + c_2\lambda_2^{n+2} + c_3\lambda_3^{n+2} \pmod p$$

所以

$$\begin{bmatrix} c_1\lambda_1^0 & c_2\lambda_2^0 & c_3\lambda_3^0 \\ c_1\lambda_1^1 & c_2\lambda_2^1 & c_3\lambda_3^1 \\ c_1\lambda_1^2 & c_2\lambda_2^2 & c_3\lambda_3^2 \end{bmatrix} \begin{bmatrix} \lambda_1^n \\ \lambda_2^n \\ \lambda_3^n \end{bmatrix} \equiv \begin{bmatrix} a_n \\ a_{n+1} \\ a_{n+2} \end{bmatrix} \pmod p$$

得到$\lambda_1^n$，$\lambda_2^n$，$\lambda_3^n$后，再用SageMath求离散对数$n = \log_{\lambda_1}(\lambda_1^n)$，大功告成。

---

flag{1807948441590188}