Real world ctf密码学题目讲解

```python
#coding:gbk
from random import randrange
from Crypto.Cipher import AES
from Crypto.Util.number import *
p = 1933879442025658861982562605919097560 41
i = lambda x: pow(x, p-2, p)

def add(A, B):
    (u, v), (w, x) = A, B
    assert u != w or v == x
    if u == w: m = (3*u*w + 4*u + 1) * i(v+x)
    else: m = (x-v) * i(w-u)
    y = m*m - u - w - 2
    z = m*(u-y) - v
    return y % p, z % p


def mul(t, A, B=0):
    if not t:
        return B
    if t%2==0:
        return mul(t//2, add(A,A), B)
    elif B!=0:
        return mul(t//2, add(A,A), add(B,A))
    else:
        return mul(t//2, add(A,A),A)
x=randrange(p)
print(mul(x, (4,10)))
#(6563950458720970587281154211125696405,125330437930804525313
```

$$y^2 \equiv x^3 + 2x^2 + x (mod p)$$

椭圆曲线的阶

(4, 10) 1
(16, 8) 2
(6, 3) 3
(5, 3) 4
(0, 0) 5
(5, 16) 6
(6, 16) 7
(16, 11) 8
(4, 9) 9
(4, 11) 10
(4, 10) 11
(16, 8) 12
(6, 3) 13
(5, 3) 14
(0, 0) 15

$$\boldsymbol{y^2 \equiv x^3 + 2x^2 + x(mod19)}$$

阶为10

(4, 10) 1
(3, 3) 2
(1, 11) 3
(9, 9) 4
(10, 12) 5
(0, 0) 6
(10, 1) 7
(9, 4) 8
(1, 2) 9
(3, 10) 10
(4, 3) 11
(6, 4) 12
(4, 10) 13
(3, 3) 14
(1, 11) 15

$$\boldsymbol{y^2 \equiv x^3 + 2x^2 + x(mod13)}$$

阶为12

对于椭圆曲线

$$\boldsymbol{y^2 \equiv x^3 + ax + b(modp)}$$

若椭圆曲线的阶等于p-1，那么就

称这类椭圆曲线为超奇异椭圆曲线

椭圆曲线问题和离散对数问题的转化（mov攻击）

若$\boldsymbol{y^2 \equiv x^2(x+m)(modp)}$，则先寻找一数n，令$\boldsymbol{n^2 modp = m}$，假设基点P的坐标为$(x_0，y_0)$，

设公钥Q=kP的坐标为$(x_w，y_w)$，那么设u=$\frac{y_w+nx_w}{y_w-nx_w}$，设v= $\frac{y_0+nx_0}{y_0-nx_0}$，则k满足下列等式：

$$\boldsymbol{v}^k \equiv \boldsymbol{u}(modp)$$

本例中，由于$\boldsymbol{y^2 \equiv x^3 + 2x^2 + x(mod p) = x(x+1)^2}$，设$\boldsymbol{x+1=w}$，则$\boldsymbol{y^2 \equiv w^2(w-1)(mod p)}$，便可以按照上述的方法进行转化。

由于本题中基点P=(4,10)

公钥Q=(65639504587209705872811542111125696405,125330437930804525313353306745824609665)

由于$\boldsymbol{w = x + 1}$，所以基点P需要变形为P=(5,10)

Q变为(65639504587209705872811542111125696406,125330437930804525313353306745824609665)

将公式代入本题，则需要先找到一个数n，令$\boldsymbol{n^2 \equiv -1(modp)}$，寻找的方法就需要用到奇波拉算法

奇波拉算法：

形如$\boldsymbol{x^2 \equiv m(modp)}$的方程，解法如下：

（1）判断m是否为模p的二次剩余，也就是$m^{\frac{p-1}{2}} modp$ 的值是否为1，如果不是则结束。

（2）通过随机试错的方法从集合{ 0 , 1 , 2 , . . . , p − 1 } 中找到一个a，使$a^{\frac{p-1}{2}} \equiv -1(modp)$

（3）$x = (a + \sqrt{a^2 - m})^{\frac{p+1}{2}}$就是方程的一个解

举个例子：比如求解方程$x^2 \equiv 4(mod7)$

根据奇波拉算法，（1）式中满足$4^{\frac{7-1}{2}} mod7 = 1$

（2）式中，不妨令a=3

代入（3）中，那么$x = (3+\sqrt{3^2-4})^{\frac{7+1}{2}} = 376 + 168\sqrt{5}$（只取有理数部分）=376就是方程的一个解

```python
def solve(p,a):#求解x**2%p=a的方程
    k=0
    P=(p-1)
    if p%4==3:
        return(gmpy2.powmod(a,int((p+1)//4),p))
    else:
        while P%2==0:
            P=P//2
            k=k+1
        q=2
        while q:
            l=gmpy2.powmod(q,int((p-1)//2),p)
            if l==p-1:
                break
            q=sympy.nextprime(q)
        b=gmpy2.powmod(q,P,p)
        x=[0 for i in range(k)]
        re_a=gmpy2.invert(a,p)
        x[k-1]=gmpy2.powmod(a,int((P+1)//2),p)
        for i in range(1,k):
            m=re_a*pow(x[k-i],2)
            n=pow(2,(k-i-1))
            if gmpy2.powmod(m,n,p)==p-1:
                j0=1
                x[k-i-1]=x[k-i]*pow(b,j0*(2**(i-1)))%p
            else:
                j1=0
                x[k-i-1]=x[k-i]%p
        return x[0]
```

本题中

p = 193387944202565886198256260591909756041

a = p−1

print(solve(p,a))#89654903351345918131227153390056628523

于是可以算出

u = 71723922681076734504981722712045302819

v = 85023335108686885465959029191933522385

那么只需要算出满足方程$\boldsymbol{v}^{k} \equiv \boldsymbol{u}(mod p)$的k即可，具体的算法为BSGS暴力破解算法

BSGS算法主要解决 ：求a^x ≡b(modp)的最小满足题意的正整数 x 的问题。

算法如下:

（1）计算m=[$\sqrt{p}$]，[]表示取整符号。

（2）设x=mi-j，i，j为未知数。

（3）变形： a^x ≡b(modp)推出 a^{mi-j} ≡b(modp)推出a^{mi} ≡ b × a^j(modp)

（4）将j遍历（0-m）的所有值，保存b × $a^j modp$的值，组成表

(5) 将i遍历（1-m）的所有值，查看$a^{mi} modp$的值是否在表中，如果在，那么根据i和j算出x

此算法的时间复杂度为$O(\sqrt{p})$

最后算出k的值为4470735776084208177429085432176719338

答案为rwctf{Singular_Elliptic_Curve_is_easy_to_break}