[ECC]XCTF-easy_ECC(WP)

原创

_orangeP 于 2021-01-21 01:15:48 发布 222 收藏 2

分类专栏： CRYPTO 文章标签： python ecc

本文链接：https://blog.csdn.net/qq_43413736/article/details/112914256

从零开始的ECC学习□ easy_ECC

主要涉及到的椭圆曲线、费马小定理到分数取余

分数取余

计算（a/b）mod p时，可将其变为((a mod p)*(1/b mod p)) mod p

根据费马小定理$x \quad mod\ p = 1$，可得出$(1/b) mod\ p = b \quad mod\ p$

```
def mod(a,b,p):
  #a/b mod p
  if b<0:
     b=-b
     a=-a
  return (a%p*pow(b,p-2,p))%p
```

椭圆曲线

在学习ECC加密之前首先得知道什么是椭圆曲线（这个dalao写的很详细）

可以简单理解为一条曲线（射影平面上满足威尔斯特拉斯方程所有点的集合）。简化表示为$y = x + a_x + b$。考虑到要将曲线变为离散的点，于是就有$y = x + a_x + b \pmod p$

定义椭圆曲线上的点的加法运算（椭圆曲线阿贝尔群，详细看上面链接）：

$$P(x,y) + Q(x,y) =$$

得知ECC的基本原理后简单处理一下代码就得出了flag：

```
class point:
    def __init__(self,x,y):
        self.x=x
        self.y=y
class ell:
    def __init__(self,p,a,b):
        self.p=p
        self.a=a
        self.b=b
    def add(self,pA,pB):
        if pA.x==pB.x and pA.y==pB.y:
            k=mod((3*(pA.x*pA.x)+self.a),(2*pA.y),self.p)
        else:
            k=mod((pB.y-pA.y),(pB.x-pA.x),self.p)
        rx=k*k-pA.x-pB.x
        rx=rx%self.p
        ry=k*(pA.x-rx)-pA.y
        ry=ry%self.p
        R = point(rx,ry)
        return R
    def ne(self,n,G):
        s=str(bin(n)[::-1])
        sumG=G
        dict={}
        for i in range(len(s)):
            if s[i]=='1':
                dict[i]=0
                maxbin=i
        for i in range(0,maxbin+2):
            if i in dict:
                dict[i]=sumG
            sumG=self.add(sumG,sumG)
        flag=0
        for i in dict:
            if flag==0:
              sumG=dict[i]
              flag=1
            else:
                sumG=self.add(sumG,dict[i])
        return sumG

def mod(a,b,p):
    #a/b mod p
    if b<0:
        b=-b
        a=-a
    return (a%p*pow(b,p-2,p))%p


p = 15424654874903
a = 16546484
b = 4548674875
ep = ell(p,a,b)
G = point(6478678675,5636379357093)
k = 546768
flag=ep.ne(k,G)
print(flag.x+flag.y)
```