巅峰极客tryecc

原创

宁嘉 于 2020-11-29 13:21:51 发布 31 收藏

分类专栏： 离散对数问题 文章标签： ECC

本文链接：https://blog.csdn.net/MikeCoke/article/details/110245462

离散对数问题 专栏收录该内容

3 篇文章 0 订阅

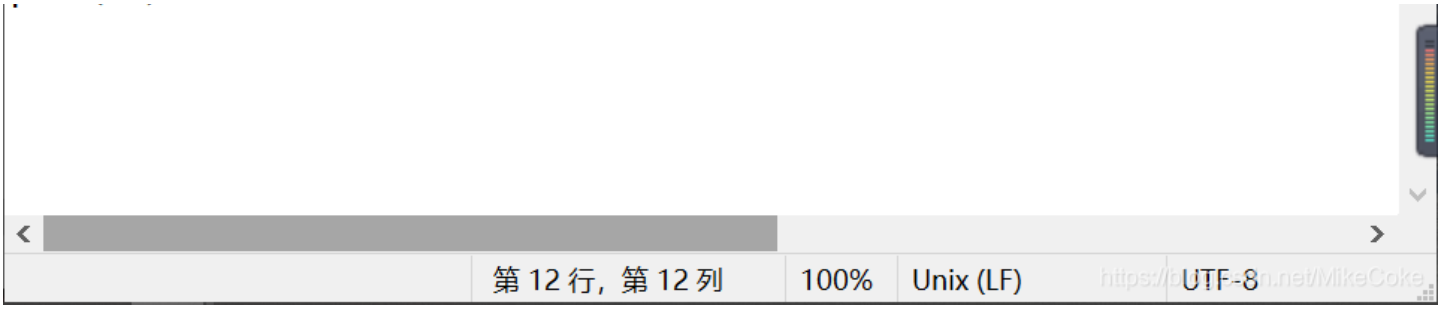
订阅专栏

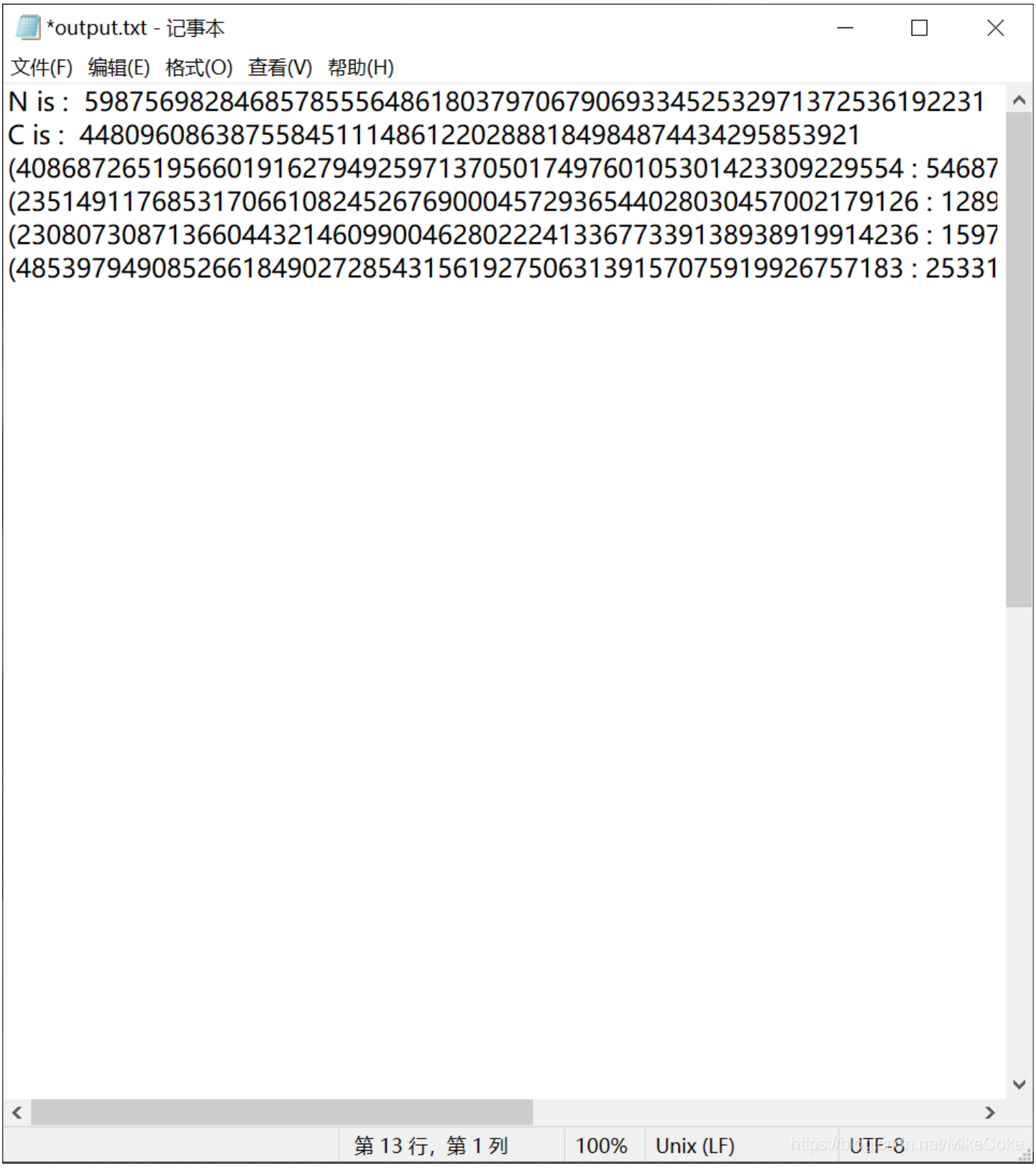
题目：（我注释后的）

tryecc.txt - 记事本

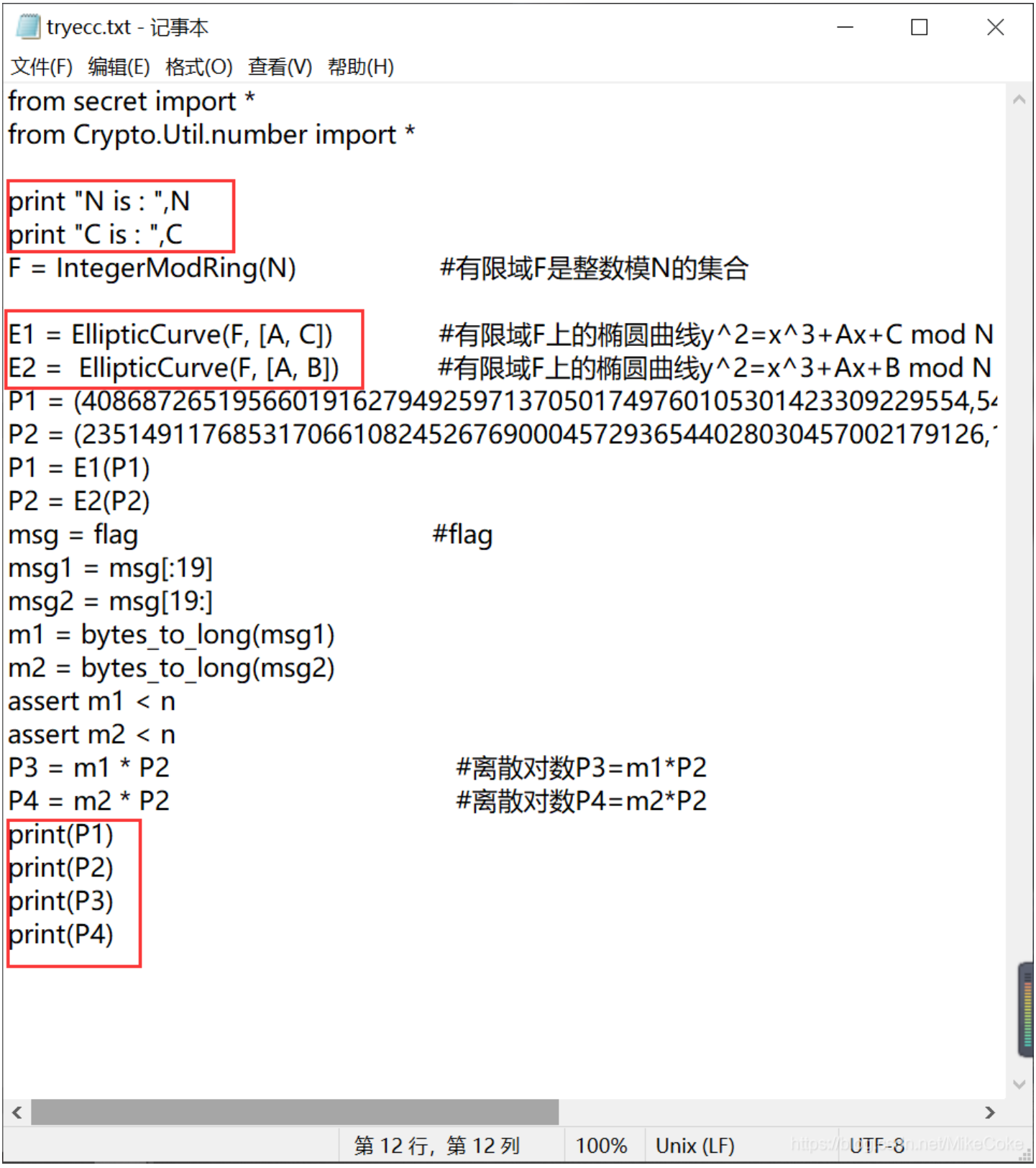
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
from secret import *
from Crypto.Util.number import *

print "N is : ",N
print "C is : ",C
F = IntegerModRing(N)                    #有限域F是整数模N的集合

E1 = EllipticCurve(F, [A, C])            #有限域F上的椭圆曲线y^2=x^3+Ax+C mod N
E2 =  EllipticCurve(F, [A, B])           #有限域F上的椭圆曲线y^2=x^3+Ax+B mod N
P1 = (40868726519566019162794925971370501749760105301423309229554,54
P2 = (23514911768531706610824526769000457293654402803045700217912 6,
P1 = E1(P1)
P2 = E2(P2)
msg = flag                               #flag
msg1 = msg[:19]
msg2 = msg[19:]
m1 = bytes_to_long(msg1)
m2 = bytes_to_long(msg2)
assert m1 < n
assert m2 < n
P3 = m1 * P2                               #离散对数P3=m1*P2
P4 = m2 * P2                               #离散对数P4=m2*P2
print(P1)
print(P2)
print(P3)
print(P4)
```


第 12 行，第 12 列
100%
Unix (LF)
UTF-8


*output.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
N is : 59875698284685785556486180379706790693345253297137253619223 1
C is : 448096086387558451114861220288818498487443429585392 1
(40868726519566019162794925971370501749760105301423309229554 : 54687
(23514911768531706610824526769000457293654402803045700217912 6 : 1289
(23080730871366044321460990046280222413367733913893891991423 6 : 1597
(48539794908526618490272854315619275063139157075919926757183 : 25331
第 13 行，第 1 列
100%
Unix (LF)
UTF-8

解题思路：



已知量：N,C, P1 , P2 , P3 , P4

未知量：A,B

通过有限域上的椭圆曲线方程：$P1(x_1,y_1),P2(x_2,y_2)$

可以解出：未知量（A和B）

$A=\frac{y_1^2-x_1^3-y_2^2+x_2^3}{x_1-x_2}$

```
#sagemath

N=59875698284685785556486180379706790693345253297137253619223 1
p= 773793889124783574343562335367
q=77379388912478357434361327939 3
C= 448096086387558451114861220288818498487443429585392 1
p1=(40868726519566019162794925971370501749760105301423309229554,546879808683716283109081231789789778648971238713 28723)
p2=(23514911768531706610824526769000457293654402803045700217912 6,128937123892129837123216378126129873181213762819 0)
p3=(23080730871366044321460990046280222413367733913893891991423 6,159792707831962038225238020158451508859287389605 40101206481)
p4=(48539794908526618490272854315619275063139157075919926757183,25331758758075812106106148031467253138305760304805478032678 1)

def cal_AB(N,C,p1,p2):
    x1,y1=p1
    x2,y2=p2
    A=Mod(inverse_mod(x1,N)*(power_mod(y1,2,N)-power_mod(x1,3,N)-C),N)
    B=Mod(power_mod(y2,2,N)-power_mod(x2,3,N)-A*x2,N)
    return A,B

A,B=cal_AB(N,C,p1,p2)
print(A)
print(B)
```

About SageMathCell

SageMath Cell

Type some Sage code below and press Evaluate.

```
8  p4=(48539794908526618490272854315619275063139157075919926757183,25331758758075812106106148031467253138305760304805478032678 1)
9
10 def cal_AB(N,C,p1,p2):
11     x1,y1=p1
12     x2,y2=p2
13     A=Mod(inverse_mod(x1,N)*(power_mod(y1,2,N)-power_mod(x1,3,N)-C),N)
14     B=Mod(power_mod(y2,2,N)-power_mod(x2,3,N)-A*x2,N)
15     return A,B
16
17 A,B=cal_AB(N,C,p1,p2)
18 print(A)
19 print(B)
```

Evaluate

Language: Sage

Share

```
693404150690001177841293118214270948684961862980427571
301433075006245665142066249069379803968009253292294929
```

Help | Powered by SageMath

得到椭圆曲线方程 E_2 后，我们就能解离散对数了。

tryecc.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
from secret import *
from Crypto.Util.number import *

print "N is : ",N
print "C is : ",C
F = IntegerModRing(N)                    #有限域F是整数模N的集合

E1 = EllipticCurve(F, [A, C])            #有限域F上的椭圆曲线y^2=x^3+Ax+C mod N
E2 =  EllipticCurve(F, [A, B])           #有限域F上的椭圆曲线y^2=x^3+Ax+B mod N
P1 = (40868726519566019162794925971370501749760105301423309229554,54
P2 = (23514911768531706610824526769000457293654402803045700217912 6,
P1 = E1(P1)
P2 = E2(P2)
msg = flag                               #flag
msg1 = msg[:19]
msg2 = msg[19:]
m1 = bytes_to_long(msg1)
m2 = bytes_to_long(msg2)
assert m1 < n
assert m2 < n
P3 = m1 * P2                             #离散对数P3=m1*P2
P4 = m2 * P2                             #离散对数P4=m2*P2
print(P1)
print(P2)
print(P3)
print(P4)
```


SageMath求离散对数的函数

求解以base为底，a的对数；ord为base的阶，可以缺省，operation可以是'+'与''，*默认为*'；

x=discrete_log(a,base,ord,operation)

这里直接解离散对数是解不出来的，因为模N不是素数，N可以分解为p,q。

所以椭圆曲线 E2：方程可以写为：

在椭圆曲线 $Ep(P2)$、 $Eq(P2)$ 上解离散对数 $P3=m1*P2$

求解出来的 $m1$ 并不是真正的 $m1$，因为分解后的椭圆曲线E2的阶已经改变了。所以这里求出的结果应该是 m1 mod order(Ep(P2)) 后的值。

即：$S = m1 \bmod order(E(P))$

通过 中国剩余定理 CRT，即可求出 m 的值。

```
N=59875698284685785556486180379706790693345253297137253619223 1
p=773793889124783574343562335367
q=77379388912478357434361327939 3
C= 448096086387558451114861220288818498487443429585392 1
p1=(40868726519566019162794925971370501749760105301423309229554,54687980868371628310908123178978977864897123871328723)
p2=(23514911768531706610824526769000457293654402803045700217912 6,128937123892129837123216378126129873181213762819 0)
p3=(23080730871366044321460990046280222413367733913893891991423 6,15979270783196203822523802015845150885928738960540101206481)
p4=(48539794908526618490272854315619275063139157075919926757183,25331758758075812106106148031467253138305760304805478032678 1)

def cal_AB(N,C,p1,p2):
    x1,y1=p1
    x2,y2=p2
    A=Mod(inverse_mod(x1,N)*(power_mod(y1,2,N)-power_mod(x1,3,N)-C),N)
    B=Mod(power_mod(y2,2,N)-power_mod(x2,3,N)-A*x2,N)
    return A,B

def dis_log(point,base):
    xp=discrete_log(Ep(point),Ep(base),operation='+')
    xq=discrete_log(Eq(point),Eq(base),operation='+')
    return crt([ZZ(xp),ZZ(xq)],[ZZ(Ep(base).order()),ZZ(Eq(base).order())])

A,B=cal_AB(N,C,p1,p2)

Ep=EllipticCurve(GF(p),[A,B])
Eq=EllipticCurve(GF(q),[A,B])

m1=dis_log(p3,p2)
m2=dis_log(p4,p2)
print(m1)
print(m2)
```

计算结果：

```
22841172820715173376955392631164734002319330 9
22791382009694924467299337996210448446366168 9
```

flag{de7a89ab1d074ef3930fb3054c0e3ac8}