In [1]:
```
p = 2^224 - 2^96 + 1 # NIST P-224
```

In [2]:
```
p
```

Out[2]:
```
26959946667150639794667015087019630673557916260026308143510066298881
```

In [3]:
```
Zp = IntegerModRing(p)
```

In [4]:
```
a, b = -3, 18958286285566608000408668544493926415504680968679321075787234672564
```

In [5]:
```
E = EllipticCurve(Zp, [a, b])
E
```

Out[5]:
```
Elliptic Curve defined by y^2 = x^3 + 26959946667150639794667015087019630673557916
260026308143510066298878*x + 18958286285566608000408668544493926415504680968679321
075787234672564 over Ring of integers modulo 26959946667150639794667015087019630673
557916260026308143510066298881
```

In [6]:
```
P = E(19277929113566293071110308034699488026831934219452440156649784352033, 1992680
```

In [7]:
```
P
```

Out[7]:
```
(19277929113566293071110308034699488026831934219452440156649784352033 : 1992680875
803447097019797437088874918420599199060394953763734319877 2 : 1)
```

In [8]:
```
E.order()
```

Out[8]:
```
26959946667150639794667015087019625940457807714424391721682722368061
```

In [9]:
```
P.order()
N = P.order()
N
```

Out[9]:
```
26959946667150639794667015087019625940457807714424391721682722368061
```

In [10]:
```
e_A = randint(2, N-1)
e_B = randint(2, N-1)
```

In [11]:
```
#Alice calcula e_A*P
#Bob calcula e_B*P
P1 = e_A*P
P2 = e_B*P
```

In [12]:
```
#Alice calcula e_A*P2
#Bob calcula e_B*P1
P11 = e_A*P2
P22 = e_B*P1
```

In [13]:
```
P11, P22
```

Out[13]:
```
((25055568003832945606535096908213452461808172344803469183224470995587 : 182885465
65377080497437026218107513170176841519117555025822354846857 : 1),
 (25055568003832945606535096908213452461808172344803469183224470995587 : 182885465
65377080497437026218107513170176841519117555025822354846857 : 1))
```

In [14]:
```
P11 == P22
```

Out[14]: True

In [ ]:
```

```