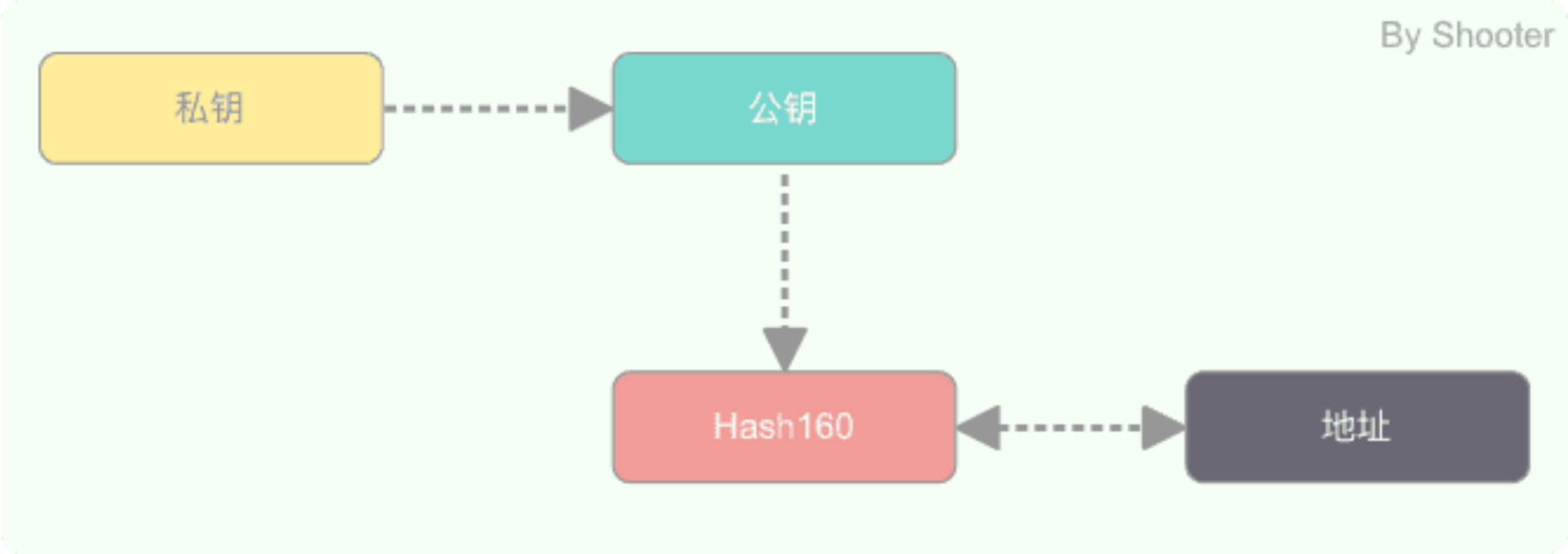
NERVOS

从椭圆曲线到一个地址

刘宁@NERVOS BJ

01 椭圆曲线

总体过程



ECDSA

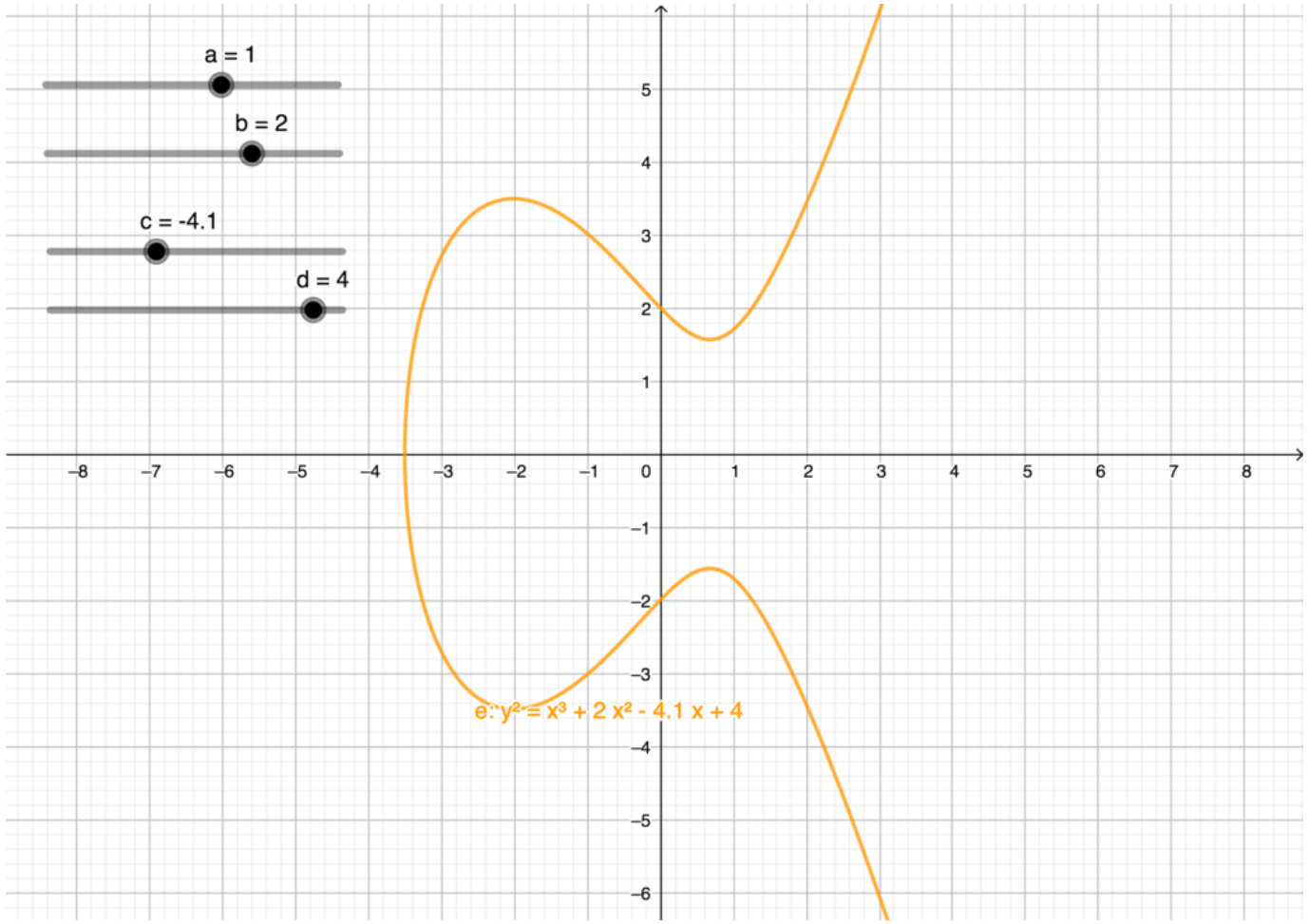
生成私钥，再得到公钥，是由ECDSA实现的。

ECDSA就是

`Elliptic Curve Digital Signature Algorithm`,

即**椭圆曲线数字签名算法**。

ECC



$$y^2 = ax^3 + b$$

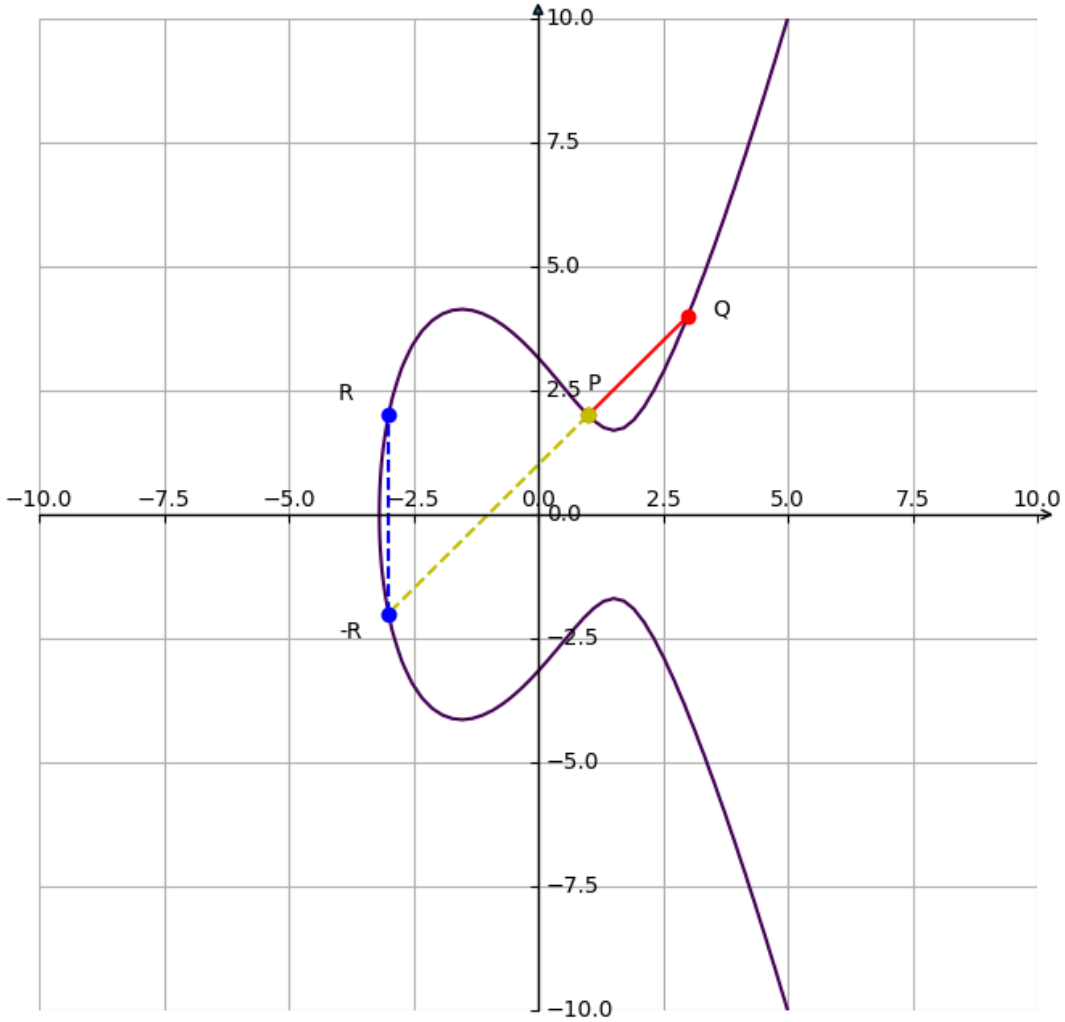
ECC

secp256k1

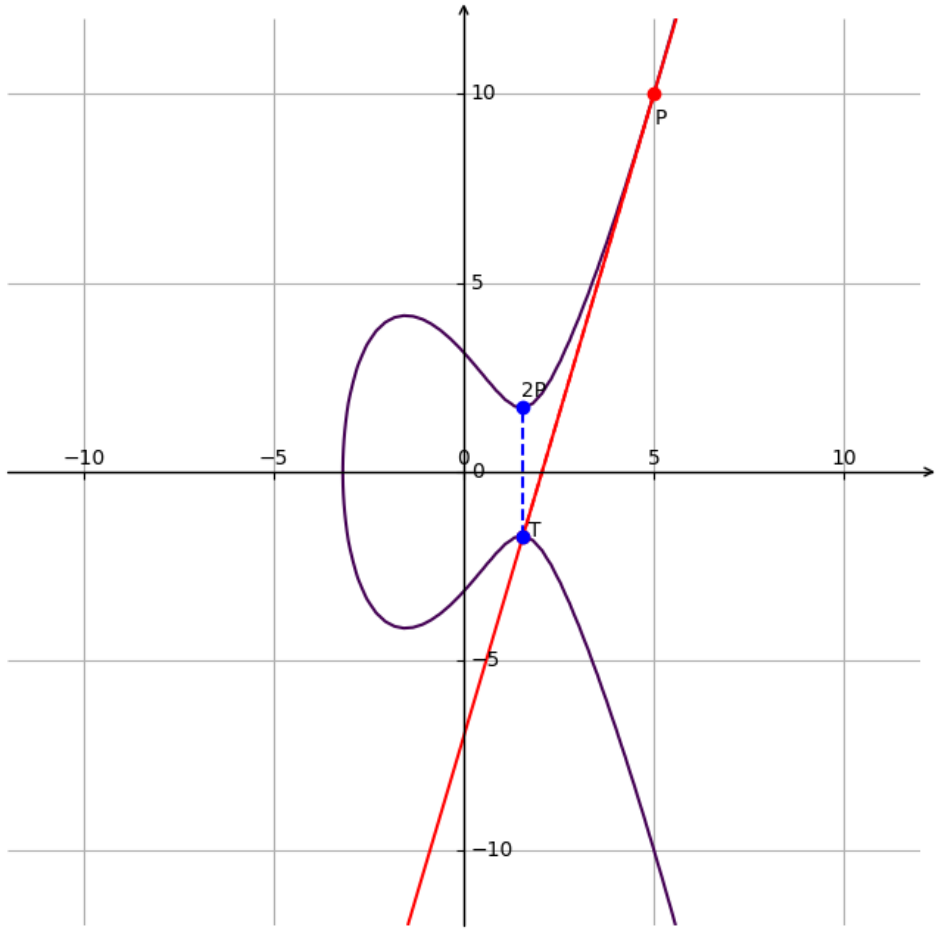
$$a = 1 \; b = 7$$

$$y^2 = x^3 + 7$$

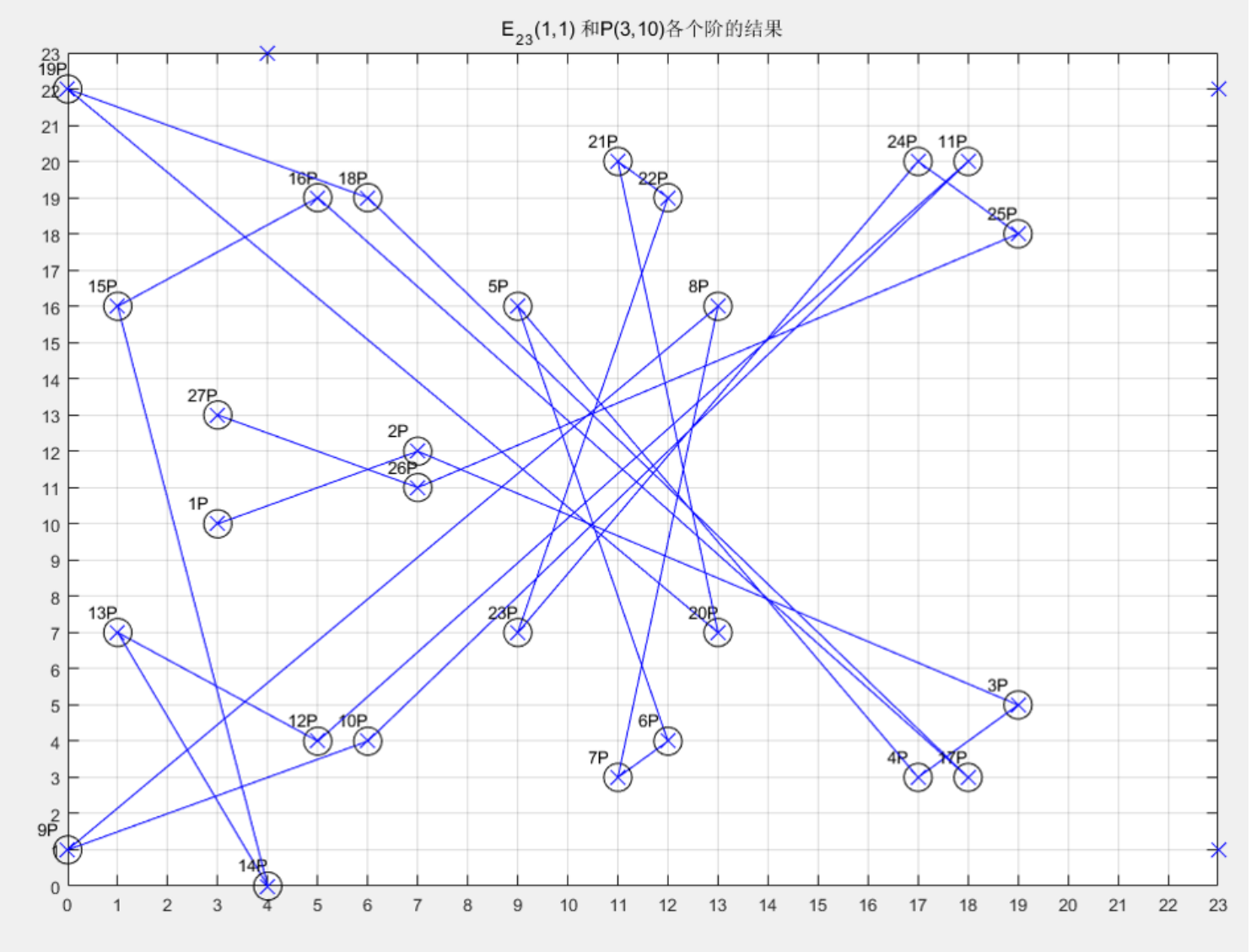
ECDSA加法



ECDSA倍乘



ECDSA 多倍乘法


$E_{23}(1,1)$ 和P(3,10)各个阶的结果
19P
21P
24P
11P
16P
18P
22P
25P
15P
5P
8P
27P
2P
26P
1P
13P
23P
20P
3P
12P
10P
6P
7P
4P
17P
9P
14P

ECDSA

椭圆曲线是连续的，并不适合用于加密；所以，我们必须把椭圆曲线变成离散的点，要把椭圆曲线定义在有限域上。

有限域表示：

7，11，13，29，233 都是质数，$\mathbb{F}_{233}$

求余

$\mathbb{F}_{233}$

$$y^2 \ mod \ 233 = x^3 + 7 \ mod \ 233$$

secp256k1

有限域为：

0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffefffffc2f

$$2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1$$

secp256k1

群的阶：

N=0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF4
8A03BBFD25E8CD0364141

从1~N的范围内其实都能当做私钥，
为了安全，要有64位长度，私钥要小于N

secp256k1生成点

生成点：G Point

Gx = 0x79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798

Gy = 0x483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8

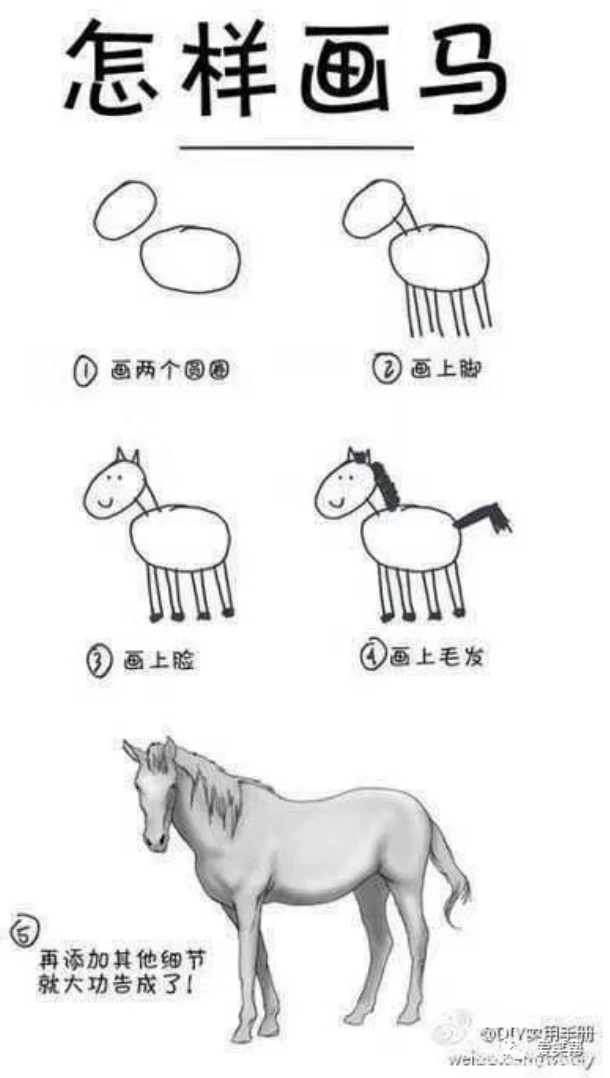
从这个点开始， 运行k*G，k是为私钥，k*G的结果就是公钥

如何判断质数

```
openssl prime
115792089237316195423570985008687907853269984665640564039457584007908834671663
```

0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEFFFFFC2F is prime

理论完成，画马



secp256k1

随机得到一个私钥：

0xccea9c5a20e2b78c2e0fbdd8ae2d2b67e6b1894ccb7a55fc1de08bd53994ea64

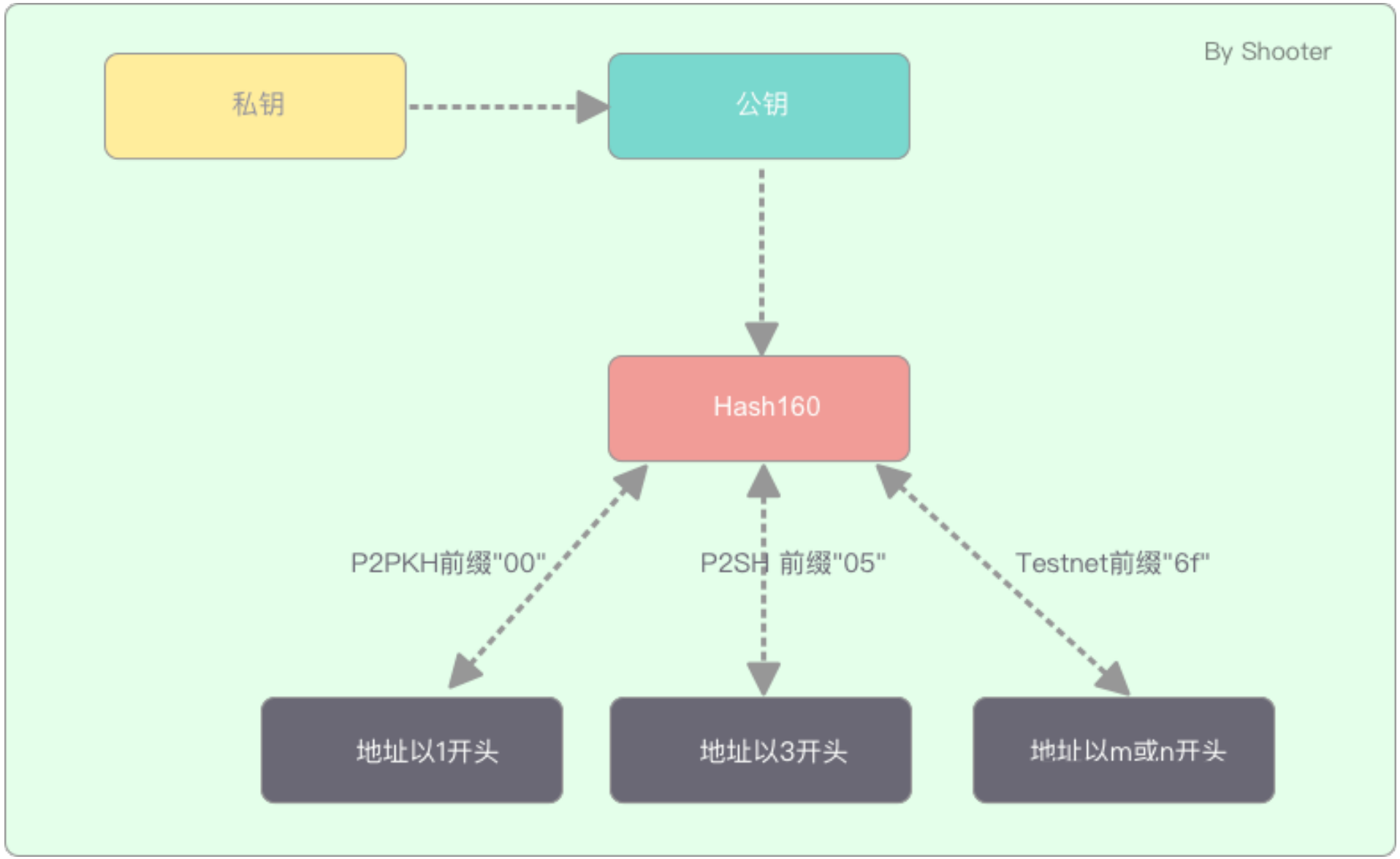
Ruby Code

```ruby
# Bitcoin's curve - the secp256k1

Pcurve = 2**256 - 2**32 - 2**9 - 2**8 - 2**7 - 2**6 - 2**4 -1 # Finite field, 有限域
# 0xfffffffffffffffffffffffffffffffffffffffffffffffffffffffefffffc2f

N = 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFEBAAEDCE6AF48A03BBFD25E8CD0364141 # 群的阶
Acurve = 0; Bcurve = 7 # 椭圆曲线的参数式. y^2 = x^3 + Acurve * x + Bcurve

Gx = 0x79be667ef9dcbbac55a06295ce870b07029bfcdb2dce28d959f2815b16f81798
Gy = 0x483ada7726a3c4655da4fbfc0e1108a8fd17b448a68554199c47d08ffb10d4b8

GPoint = [Gx, Gy] # 椭圆曲线生成点, Base point.
#(Gx**3+7) % Pcurve == (Gy**2) % Pcurve, GPoint在椭圆曲线上, x/y坐标符合椭圆曲线方程

h = 1 # Subgroup cofactor, 子群辅因子为1, 就不参与运算了

# Pcurve, N, GPoint, secp256k1的函数式, 都是严格规定的, 严禁修改!!!
```

https://github.com/NervosBeijingCommunity/jupyter-ppt/blob/master/RubySaturday20190309/code/ecc-secp256k1.rb

02 生成地址

前缀


By Shooter
私钥
公钥
Hash160
P2PKH前缀"00"
P2SH 前缀"05"
Testnet前缀"6f"
地址以1开头
地址以3开头
地址以m或n开头

hash160运算

```
bytes = [pub_key].pack("H*") # 转为16进制

hash160_val =
Digest::RMD160.hexdigest(Digest::SHA256.digest(bytes) )
```

前缀符

```
'00'+ '2b6f3b9e337cedbb7c40839523fb1100709c12f7'
```

前缀有很多种 00, 05, 6f

校验和

执行2次SHA256，取前8位作为校验和

```
hex_str = [step_04].pack(“H*")
```

```
checksum =
Digest::SHA256.hexdigest(Digest::SHA256.digest(h
ex_str) )[0...8]
```

第4步结果跟第5步结果合并

```
'002b6f3b9e337cedbb7c40839523fb1100709c12f7' +
'86b2e90c'

# step_04 + checksum
```

Base58编码

```
def encode_base58(int_val, leading_zero_bytes=0)
  alpha =
"123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwx
yz"
  base58_val, base = '', alpha.size

  while int_val > 0
    int_val, remainder = int_val.divmod(base)
    base58_val = alpha[remainder] + base58_val
  end

  base58_val
end
```

进行base58编码

```
step_06 =
"002b6f3b9e337cedbb7c40839523fb1100709c12f786b2e
90c"
leading_zero_bytes = (step_06.match(/^([0]+)/) ?
$1 : '').size / 2
# leading_zero_bytes的作用是字母填充
address = ("1" * leading_zero_bytes) +
encode_base58(step_06.to_i(16) )
```

14xfJr1DArtYR156XBs28FoYk6sQqirT2s

生成地址

更多详细介绍 https://www.jianshu.com/p/954e143e97d2

03 燃烧地址

燃烧地址

```
11111111111111111111114oLvT2
```

```
12ZEw5Hcv1hTb6YUQJ69y1V7uhcoDz92PH
```

更多介绍：https://www.jianshu.com/p/cc9e651faca4

03 靓号地址

靓号地址

```
1SNiPeRGTJuELpHRDTRzg7A64weBCWpNq => sniper

1LoveYoURwCeQu6dURqTQ7hrhYXDA4eJyn => 1LoveYoU
```

Hash后的数据是无规律的，越是有规律的数据，动用的计量越大，这是一种标示，也是实力的象征

04 来个骚操作

骚操作

以太坊也是使用Secp256k1椭圆曲线得到私钥、公钥，跟比特币使用椭圆曲线算法相同。

骚操作

用之前得到同一份公钥生成eth地址

```
04d061e9c5891f579fd548cfd22ff29f5c64
2714cc7e7a9215f0071ef5a5723f691757b2
8e31be71f09f24673eed52348e58d53bcfd2
6f4d96ec6bf1489eab429d
```

骚操作

0x9156a7cdab767ffe161ed21a0cb0b688b545b01f
14xfJr1DArtYR156XBs28FoYk6sQqirT2s
不同的地址是同一份私钥

给资产管理带来了一些方便，但如果一份私钥泄露，2份资产都有可能被盗。
管理一时爽，泄露竹篮打水一场空

更多资料：https://www.jianshu.com/p/d3837398f69c

NERVOS

In Math We Trust

NERVOS BJ