目录

[1. 椭圆曲线理论基础 2](#_Toc707896227)

[1.1. 定义 2](#_Toc552482809)

[1.2. 四则运算 3](#_Toc2006764882)

[1.3. 有限域GF(p) 5](#_Toc1466695639)

[1.4. 有限域椭圆曲线 6](#_Toc1936304407)

[1.5. 有限域椭圆曲线点的阶 7](#_Toc500981811)

[2. ECC椭圆曲线加密算法 7](#_Toc1865401237)

[2.1. 定义 7](#_Toc684827706)

[2.2. 安全性 8](#_Toc1534390469)

[2.3. 加密解密原理与演示 9](#_Toc1516979307)

[2.4. ECDSA数字签名原理 10](#_Toc945355565)

[2.5. 签名演示 12](#_Toc1506960449)

[2.6. secp256k1 12](#_Toc62133625)

[3. 结语 13](#_Toc602782933)

[4. 关于作者 13](#_Toc1292392032)

[5. 参考资料 14](#_Toc1583276066)

**大白话椭圆曲线加密算法**

**作者： 虞双齐**

**版本记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 作者 | 内容 |
| 2020年04月03日 | V1 | 虞双齐 | 初版 |
| **云共享版：**<https://kdocs.cn/l/spNSpCEr5?f=101> | | | |

你应该听过 ECC，ECDH或者ECDSA。ECC是椭圆曲线加密算法（Elliptic curve cryptography）的简称，后面两个是基于它的算法实现。

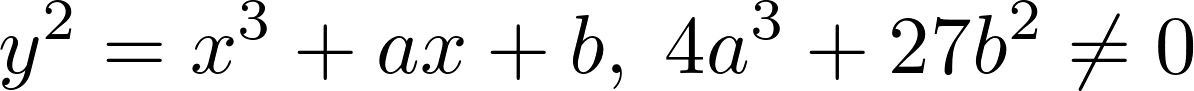
在数字货币加密技术中，不得不谈ECC，它是数字货币的安全基石。本文不涉及ECC中复杂的数学知识，笔者将努力使用简单通俗的语言来解释ECC是如何提供与保障加密安全的。

整篇文章，先讲解所涉及的理论基础知识，然后讲解ECC的定义，再通过实例来讲解ECC的加密解密原理和ECDSA的签名原理。对理论基础不了解的读者，请务必掌握理论基础后再继续往下看，不推荐跳读。

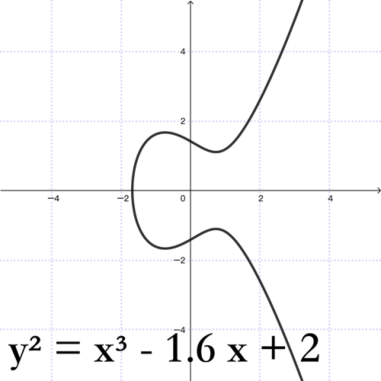
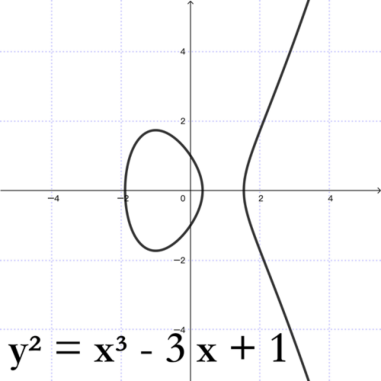
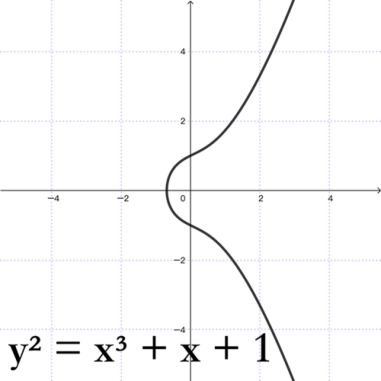
# 椭圆曲线理论基础

## 定义

什么是椭圆曲线？在数学上，它是下方方程所有点的集合。

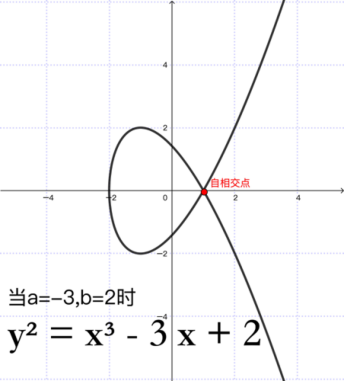
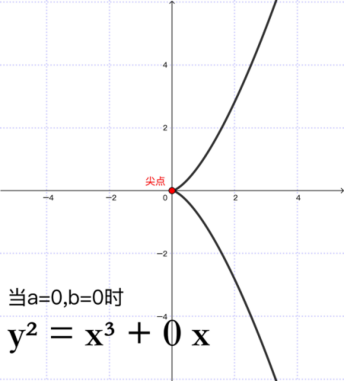


下图是不同a,b得到的不同图形的椭圆曲线。可以看到椭圆曲线的形状，并非椭圆的。只是因为椭圆曲线的描述方程，类似于计算一个椭圆周长的方程，故得此名。



随着a,b的不同([点击查看](https://img.learnblockchain.cn/book_geth/椭圆曲线.gif)演示动图)，椭圆曲线也会在平面上呈现不同的形状。辨识度很高，可以看到椭圆曲线始终是关于x轴对称的。

方程后面的判别式 wpsoffice 是来保障曲线的光滑（非奇异），也就是说在曲线里面不能有尖点、自相交点和孤立点。从而保证椭圆曲线上所有的点都有切线。



上方曲线中他们的判别式均为0。左曲线带尖点；右曲线曲线自相交。他们都不是有效的椭圆曲线。

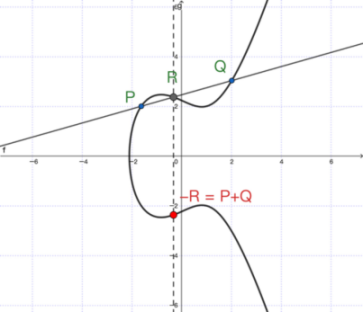
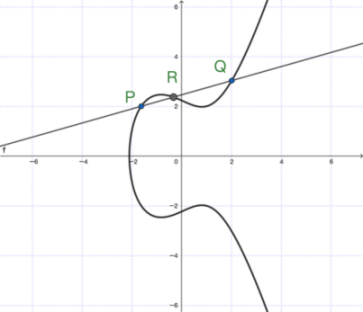
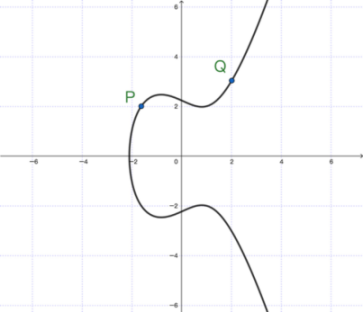
## 四则运算

不同于代数几何中的四则运算，在椭圆曲线中有定义自有四则运算。

#### 加法运算

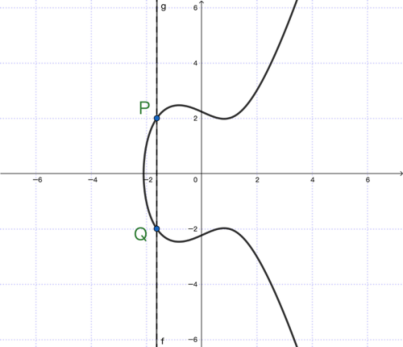
已知椭圆曲线上的点P和Q，求P+Q。

首先在曲线上标出点P和点Q。再做经过点P和Q的直线，与曲线相交于第三点R。继续做点R关于X轴的对称点-R。这个 -R 点就是点P和Q相加得到的点，记做 P+Q= -R。

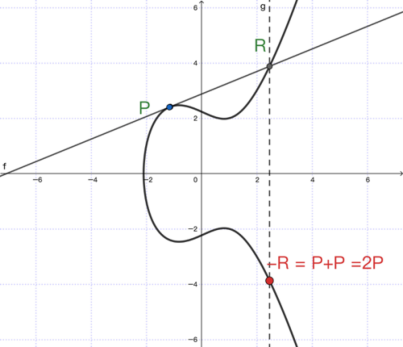


因椭圆曲线是关于X轴对称的，-R是点R的对称点， 点 -R 必然也在椭圆曲线上，换句话说 P+Q 也在椭圆曲线上。

点R的存在对于 -R 至关重要。在大多数情况下直线一定将和椭圆曲线相交于第三点。如果直线和椭圆曲线没有第三个交叉点时，即点R不存在时，就无法按上面方法求P+Q。以下情况中，点R不存在。

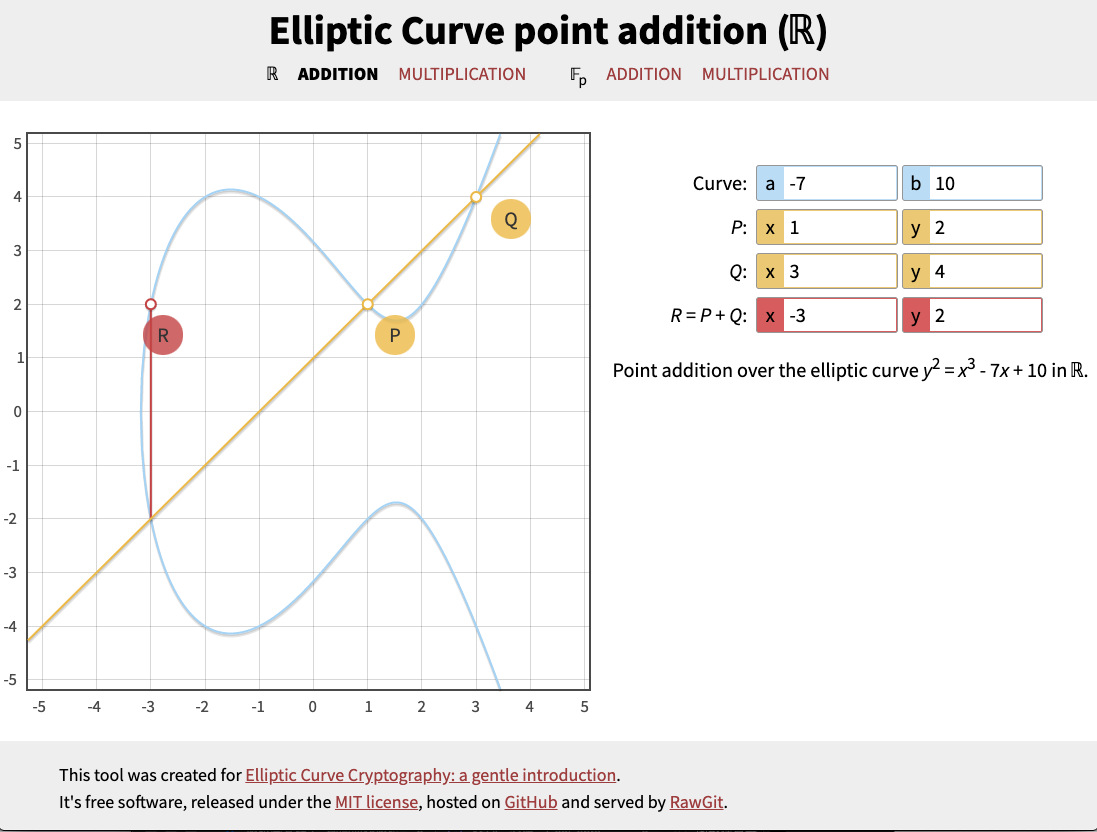
当 P = -Q 时，直线将于Y轴平行，没有第三个交点。假设 P+Q=O，此时P +Q =P+(-P)=O。移项得到 P + O = P。

这意味着点P加上一个点O后还是自身点P。我们把这个点O成为加法的单位元，曲线上的任意点与单位元相加不会改变其值。实际上点O表示平行线的相交点，称为无穷远点 O∞。可以理解为所有平行于y轴的直线交于O点。



当P=Q时，只能以点P作为切点做一条关于椭圆曲线的切线，切线将与椭圆曲线相加与另一点，记做点R。同样，做点R的关于X轴的对称点-R。这个 -R 点就是点P和P相加得到的点，记做 P+P=2P=-R。

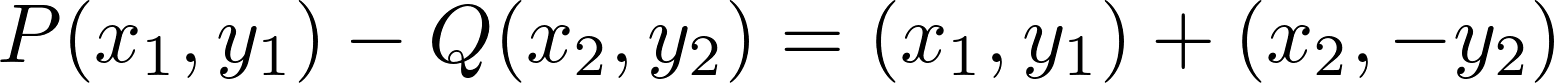
在图像上，我们可以非常便捷的找到 P+Q点。而在数学上，该如何计算呢？可以通过代数算法来计算P+Q，但这里不展开讲解，本文将借助网页工具[《Elliptic Curve point addition (ℝ)》](https://cdn.rawgit.com/andreacorbellini/ecc/920b29a/interactive/reals-add.html)来进行求解。



#### 减法运算

已知椭圆曲线上的点P和Q，求P-Q。

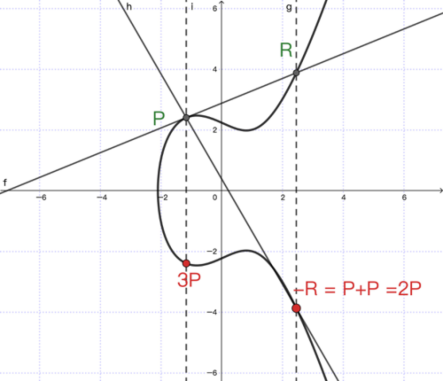
因 P -Q = P +(-Q)，求 P-Q就是点P与点-Q的加法运算。而点-Q就是点Q对称点，在运算中称-Q是Q的负元，Q(x,y)的负元是(x,-y)。综上，可得P-Q的计算公式如下：



#### 乘法运算

已知椭圆曲线上的点P和数值n，求nP。

前面已计算出P+P，同理可以继续计算出 P+P+P=2P+P。



重复 n 次加法运算就可以得到 n次点P相加的结果，记为 nP 。我们把这个过程叫做P的自累。

7A235AAB-EB49-443A-973F-04F9043DDEE0

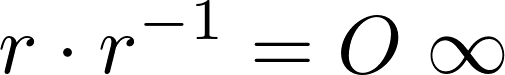
这就是乘法运算，比如 3P=P+P+P，通过进行3 次加法运算就可以得到3P的值。

**思考：**如果 n 相当大，如何提高加法运算效率？

#### 除法运算

已知椭圆曲线上的点P和数值r，求 r-1P。

注意，这里的r-1并不是表示r的负一次方。在椭圆曲线中有如下定义。



当已知r和单位元O的值即可求解出r-1，从而使用乘法求解r-1P。

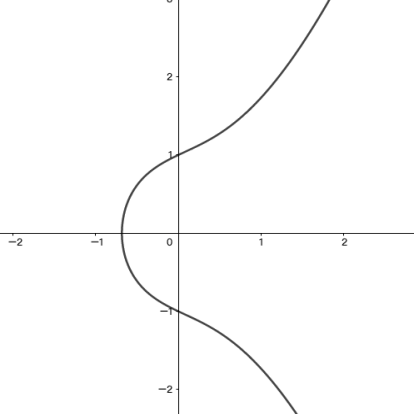
假设已知椭圆曲线上的点P(1,4)和r=4,椭圆曲线单位元为8。则有 4•r-1=8,得r-1=2,r-1P=2P。

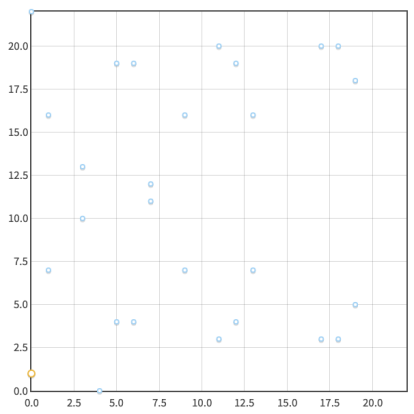
## 有限域GF(p)

有限域是指有限个元素的域,记做 GF(p),其中 p 是质数，有{0,1,2,3...,p-1}共p个元素。例如GF(7)={0,1,2,3,4,5,6}。

GF(p)中的四则运算都是基于取模运算的，就是小学学过的带余除法的余数。例如：12 除以7，商为1，余数为5，记做 12 mod 7 =5 。

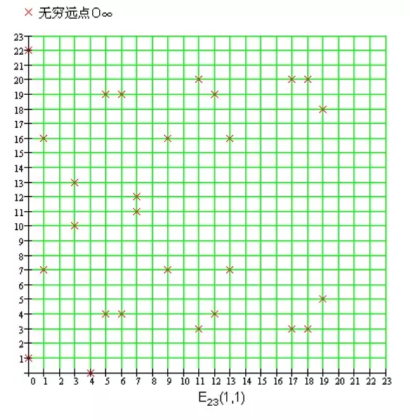
如果两个不同的数a,b除以同一个数p之后余数相同，称之为a,b模 p 同余，记 a mod p = b mod p。为了简便，我们把“mod p”统一写到最后面，并且为了防止与 “a=b”混淆，使用恒等符号“≡”，记 a ≡ b (mod p)。

而椭圆曲线（如 y2=x3+x+1）是定义在实数域上的，实数是连续的，导致了椭圆曲线的连续（右图）。这种连续性，使它并不合适用于安全加密。所以，可以把椭圆曲线定义在有限域上，使得曲线变成离散的点。下图是椭圆曲线取p为2所形成的离散的点。

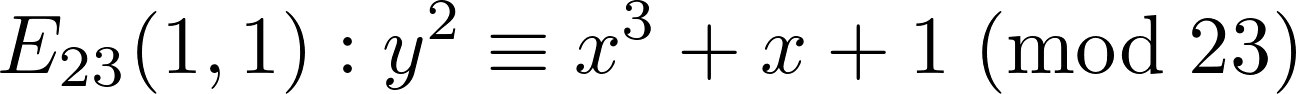


## 有限域椭圆曲线

如前面所述，在有限域 Fp上的椭圆曲线，可用于加密。按下定义有限域椭圆曲线：

1. 选择满足条件4a3+27b2 ≠ 0 (mod p)，且小于p的两个非负整数a、b。
2. 满足方程y2=x3+ax+b (mod p) 的所有点(x,y)，再加上无穷远点O∞，构成一条椭圆曲线。其中x,y 属于 0 到 p-1 间的整数。

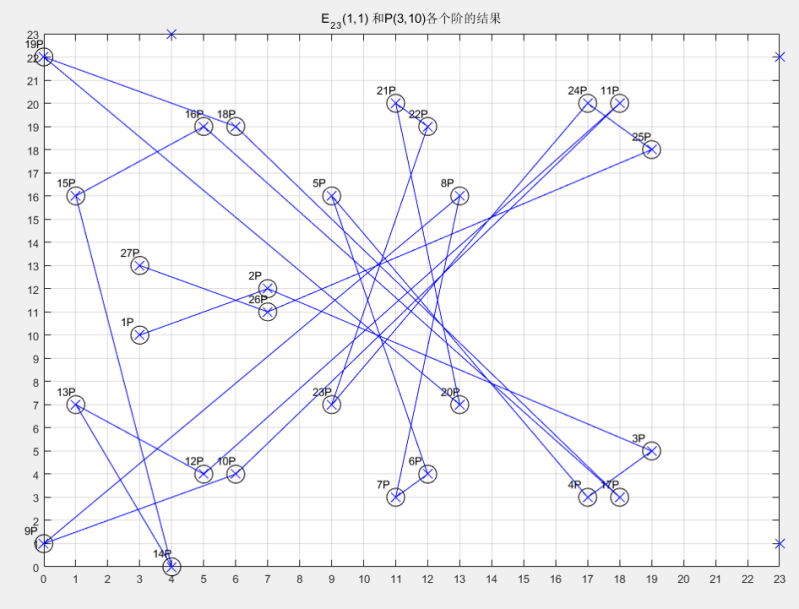
我们把这条椭圆曲线记为 Ep(a,b),比如E23(1,1)方程如下，它的离散点如右图所示。



## 有限域椭圆曲线点的阶

如果椭圆曲线Ep(a,b)上一点P，存在最小的正整数n使得 nP≡O∞,则将n 称为 P的阶。

若 n 不存在，则P是无限阶的。下图是E23(1,1)和P(3,10)各个阶的结果。从图可以明显看出，点的分布和顺序都是杂乱无章的。



计算可得 27P=-P=(3,13)，所以 28P=27P+P=-P+P=O∞，因此E23(1,1)上点P(3,10)的阶为29。

# ECC椭圆曲线加密算法

## 定义

椭圆曲线加密算法，描述的是一条Fp的椭圆曲线，记为：T =(p,a,b,G,n,h)。

其中：

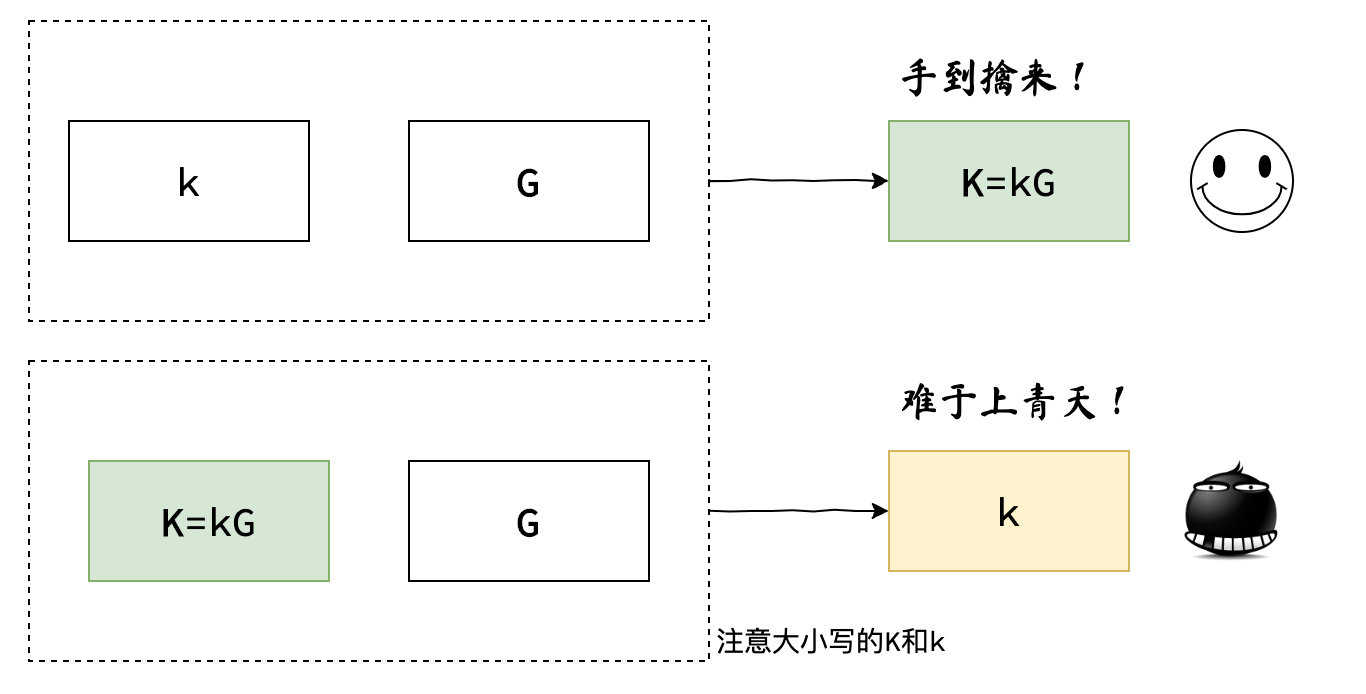
1. p,a,b 用于确定一条有限域椭圆曲线Ep(a,b)；
2. G为基点；
3. n为G点的阶数；
4. h是椭圆曲线上所有点的个数m与n相除的整数部分。

这几个参数取值，直接影响到加密的安全性。参数值一般满足以下条件：

1. p 当然越大越安全。但是越大，计算速度会变慢，200位左右可以满足一般安全要求;
2. p ≠ nh ;
3. pt≠ 1 (mod n), t∈ [1,20)；
4. 4a3+27b2≠0(mod p);
5. n为素数；
6. h≤4。

## 安全性

如果已知k和G，根据乘法法则求K是很容易的，但反过来，仅已知 K和G，求k是非常困难的。

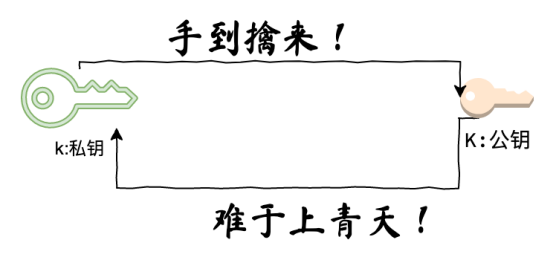


在普通代数里，比如说你有一个数10，然后有人告诉你，他是用5做的乘法得到的10，你就知道另一个乘数肯定是2。 但是在ECC里中，想求k 是非常困难的。

为了寻找 k , 你需要暴力枚举，直到找到了符合条件的点。以目前人类文明的计算能力这是不可能的，尤其当你使用了非常大的数值n。

实际应用中的ECC原则上会把 k 取得相当大，n 也相当大，要把 n 个解点逐个计算出来是不可能的。这个逆向寻找k的过程的困难就是离散对数问题。 这个问题的难度保证了ECC的安全性。

这里的 k就是私钥(private key),K是公钥(public key)。通过私钥计算公钥非常容易，但是希望通过公钥逆向推导私钥是非常困难的。



非常困难，不代表比可能。科学家们已经尝试通过量子计算机暴力枚举，寻找k值。但不需要过于担心，现有的量子计算机还是一个婴儿，科学们也在研究抗量子计算机破解的加密算法。

## 加密解密原理与演示

在ECC之前，已经有诸如 RSA等非对称加密算法，但是ECC是一种更加优越的非对称加密算法。

假设情报员葛二蛋需要加密传递一份情报给延安总部。如果葛二蛋掌握了椭圆曲线加密算法，则他可以这样加密传递情报。

**第一部分：**延安总部告诉葛二蛋使用哪个公钥加密情报。

1. 延安总部选定一条椭圆曲线Ep(a,b)和一个基点G。假设选定E23(4,20),基点G(13,23),G的阶数n=37。
2. 使用私钥 k，生成公钥 K=kG。假设k=25,则K=25G=(14,6)。
3. 将椭圆曲线E23(4,20)、n=37、K(14,6)、G(4,20)传递给葛二蛋。

**第二部分：**葛二蛋使用公钥K加密情报。

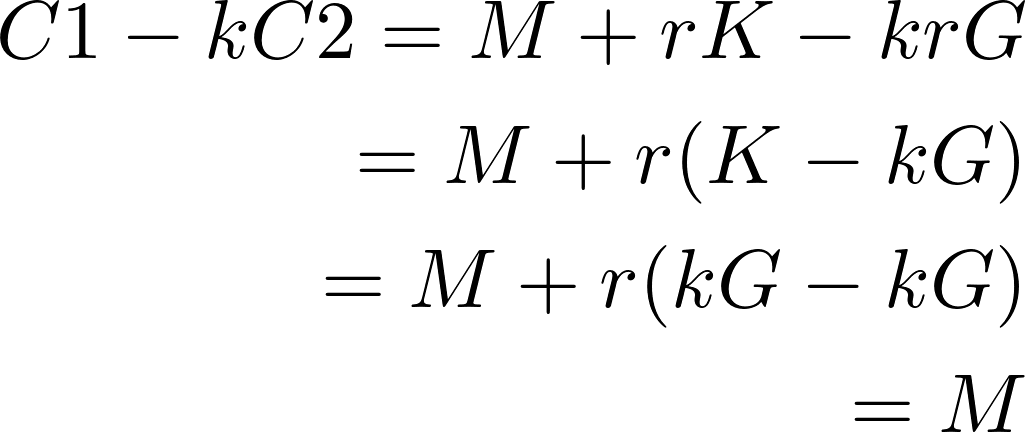
1. 葛二蛋将待传递的情报进行编码。假设编码结果为3;
2. 在椭圆曲线E23(4,20)上，当x=3时到，y=28。把这点M（3，28）当做情报编码在曲线上的映射;
3. 生成一个随机数 r （r<n），假设 r=6;
4. 计算点C1= M + rK 和C2=rG;
5. C1= M+6K= M+6\*25G=M+150G=（3,28）+(27,27)=(6,12);
6. C2=rG=6G=(5,7);
7. 将点C1(6,12)和C2(5,7)传递给延安总部;

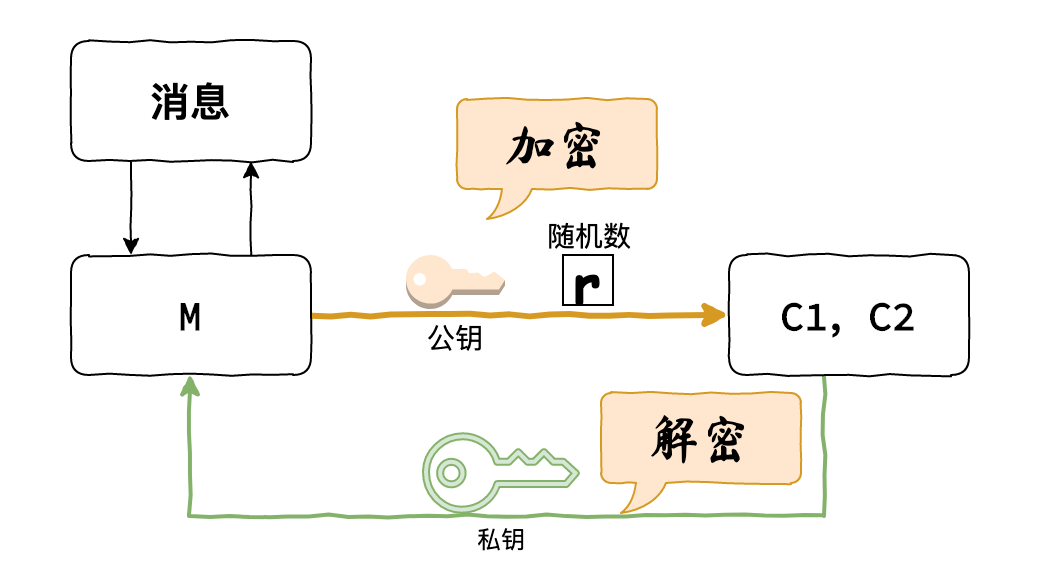
**第三部分：**延安总部解密情报。

延安总部收到C1和C2后，计算 C1-kC2，计算结果应该就是情报在曲线上的映射点M。

C1 - kC2= C1- 25C2= (6,12) - 25(5,7)=(6,12) - (27,27)= (6,12) + (27,2)=（3，28）。

延安总部能解密出情报的原因是因为：





在这个加解密过程中，如果有一个偷窥者H，他只能看到 Ep(a,b)、K、G、 C1、C2 。

通过 K= kG 求k 或者通过 C2=rG求 r 是相对困难的。因此 H 无法得到A、B件传送的明文消息。

需要强调的是，如果延安总部需要加密发送指令给葛二蛋，也可以采用同样的方式进行，唯一不同的是延安总部需要使用葛二蛋提供的公钥进行加密。只要双方各自的私钥不泄露，那么他们间的情报交流是安全的，即使落在别人手里，也无法解密。

## ECDSA数字签名原理

有一次，延安总部需要紧急通知所有情报员，“国共合作已宣告破裂，请务必保护和隐藏好自己”。情况紧急，延安总部如果逐个给所有情报员发送加密通知，则效率低下。延安总部可以将此通知刊登到报纸上，这样一早起来各地的情报员就可以看到。

可问题是，情报员怎么确信该通知真的出自延安总部呢？这里，我们就需要用到数字签名技术。

ECC是加密算法，无法直接用于数字签名。在ECC之前就已经有数字签名（digital signature）技术，如DSA。

椭圆曲线签名算法（ECDSA）就是使用ECC对DSA数字签名算法的模拟，最终签名可得到两个值r和s。验证签名只需要求解一个值，判断值是否与r相同，如果相同，则说明是有效签名。

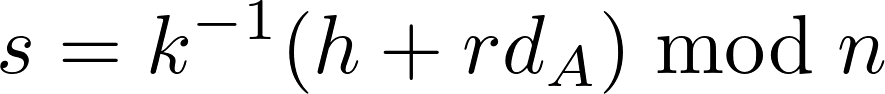
ECDSA签名时，已知Ep(a,b)、基点G、G的阶数n。

**第一部分：生成摘要**

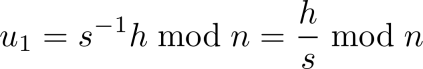
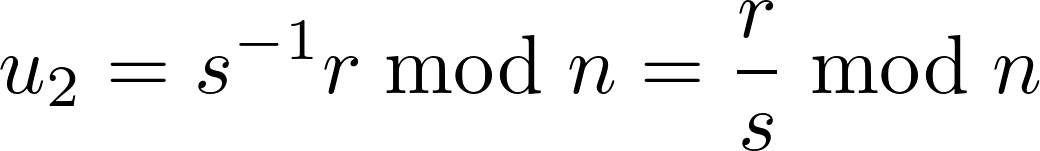
要签名的内容，实际上并非待签名数据的明文，而是对数据的哈希进行签名。

记做 /private/var/folders/_c/cxql7lhd7ss9p1zjm_xc20z40000gp/T/com.kingsoft.wpsoffice.mac/wpsoffice.YsdwOewpsoffice。

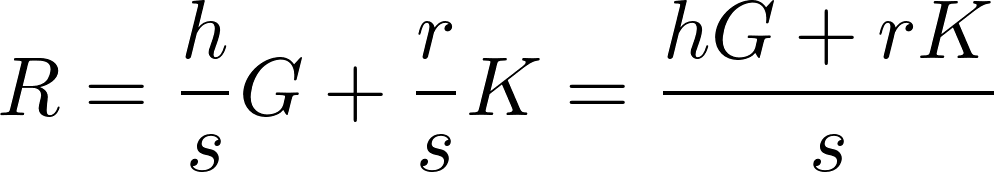
**第二部分：生成签名**

1. 生成随机数 k，k是小于n的正整数。
2. 计算 /private/var/folders/_c/cxql7lhd7ss9p1zjm_xc20z40000gp/T/com.kingsoft.wpsoffice.mac/wpsoffice.onmKCUwpsoffice
3. 计算 /private/var/folders/_c/cxql7lhd7ss9p1zjm_xc20z40000gp/T/com.kingsoft.wpsoffice.mac/wpsoffice.qSmnxhwpsoffice,如果 x mod n 为0，则回到步骤1重试随机数。
4. 计算  ,其中 dA为私钥。这里的k-1 并不是k的-1次方，而是表示k的逆元。在ECC中有 a•a-1 (mod n)=O∞=1
5. 签名结果就是r,s。公开签名数据message 和 r和s值。

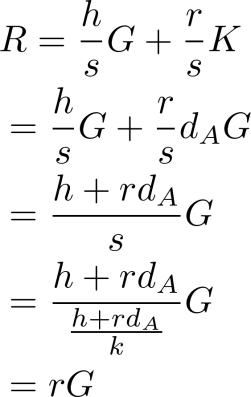
**第三部分：验证签名**

1. 计算 
2. 计算 
3. 计算点R，，其中K为签名者的公钥。
4. 判断点R 是否等于 rG。如果两者相同，则说明签名合法。

验证中求解的R的计算过程如下：



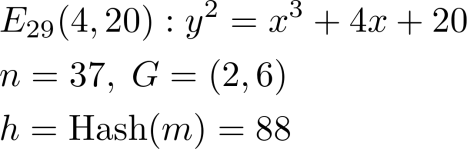
为什么说要求解R就可以呢？这是因为：



这里的关键是在于在签名时引入了随机数k，提供了安全性。即使对于同一消息，只要改变随机数k,所得到的签名也会随之变化。

## 签名演示

现在，延安总部将使用ECDSA签名通知刊登到报纸上。假设已知信息如下：

* **延安总部签名**

1. 假设延安总部的私钥是7，则K=kG=7(2,6)=(3,28);
2. 生成随机数k =11;
3. 计算 P=kG=11(2,6)=(16,2);
4. 计算 r= 16 mod 37 = 16;
5. 计算 s= k-1(h+rk) mod 37 = k-1 (88+16\*7) mod 37 = 200 k-1 mod 37。因k•k-1 (mod 37)=11•k-1 (mod 37)=1,得 k-1为27。因此 s= 5400 mod 37 =35;
6. 在报纸刊登通知和签名结果(r,s);

* **情报员验证签名**

1. 计算u1= s-1 \*88 mod 37 = 18\*88 mod 37 =30;
2. 计算u2 = s-1 \*16 mod 37 = 18 \*16 mod 37 =29;
3. 计算R= 30G +29K = (3,1)+(14,23)= (16,2);
4. 计算 16 mod 37 = 16 和 r相等，说明签名合法;

## secp256k1

secp256k1 是高效密码组标准([SECG](https://www.secg.org/)) 协会开发的一套高效的椭圆曲线签名算法标准。在比特币流行之前，secp256k1并未真正使用过。自从比特币之后，secp2256k1已成为数字货币中默认的数字签名算法。大多数常用的曲线具有一个随机的结构，但是secp256k1是为了更有效率的计算而构造了一个非随机结构。因此如果该算法的实现通过合理的优化，其计算效率可以比别的曲线快30%以上。同时，与常用的NIST曲线不同，secp256k1的常量是通过可预测的方式挑选的，这可以有效的减少防止曲线设计者安置后门的可能性。

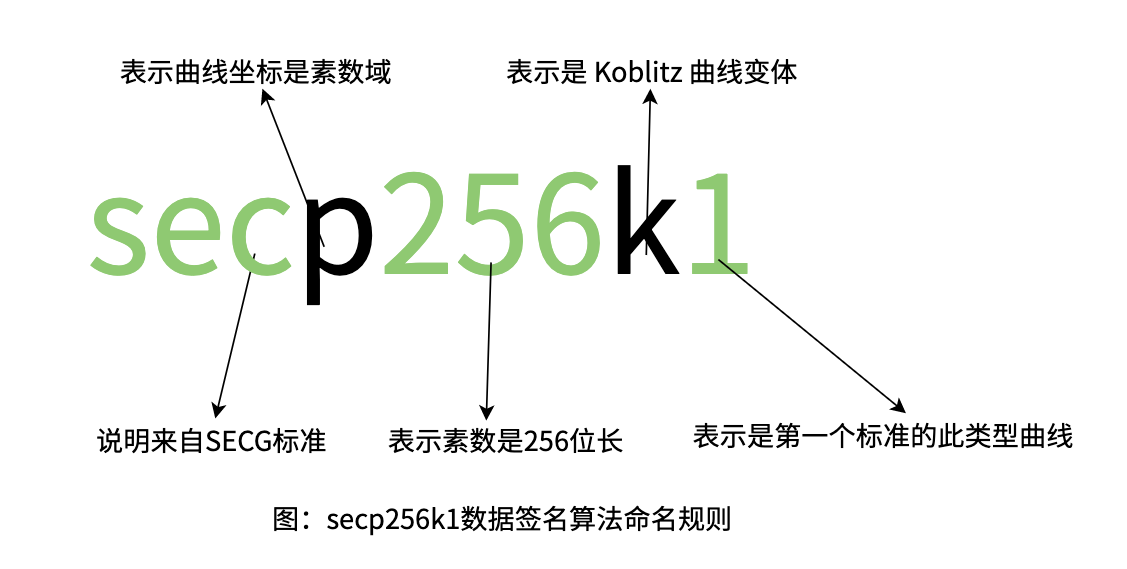
secp256k1 是一条基于有限域的Koblitz 椭圆曲线，T =(p,a,b,G,n,h)的各项参数定义如下：

1. p是一个无限接近2256 的大数;

p = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F

= 2256 - 232 - 29 - 28 - 27 - 26 - 24 - 1。

1. a=0,b=0;
2. G= 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8
3. n = FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE BAAEDCE6 AF48A03B BFD25E8C D0364141
4. h=01



为什么比特币要选择 secp256k1 签名算法而不是其他算法呢？比特币开发者社区曾讨论过 secp256k1 是否安全。中本聪没有明确解释，只是说道”有根据的推测”。

社区的讨论不外乎是在安全和效率上做权衡，选择一个不受任何政府控制、无后门的签名算法是比特币的首要考虑因素；其次，也需要提供计算速度，毕竟在比特币中签名与校验签名是不断在处理的事情（60%左右的CPU时间几乎全用在这上面），而具有可预测性、高计算效率特性的Koblitz曲线是不错的选择；

除椭圆曲线签名算法使用非常短的私钥和签名值外。基于安全第一，效率第二原则，secp256k1 就是一个优解。以太坊的签名算法也是采用 secp256k1。

# 结语

椭圆曲线加密算法是一种非对称加密算法，比RSA具有更高的效率和更短私钥。有效解决“提高安全强度必须增加密钥长度”的工程实现问题。且已得到广泛的支持和使用。使用ECC作为加密算法已成为首先项。

# 关于作者

虞双齐是区块链技术CTO，登录学院合伙人。



# 参考资料

1. 掘金[《椭圆曲线加密原理与应用》](https://juejin.im/post/5d3a68a96fb9a07ed91207e7)
2. 掘金[《椭圆曲线机密算法》](https://juejin.im/post/5a67f3836fb9a01c9b661bd3)
3. 博客园-[ECC椭圆曲线详解](https://www.cnblogs.com/Kalafinaian/p/7392505.html)