Elliptic Curve 椭圆曲线

1. 困难问题
2. 椭圆曲线是什么
3. 加密步骤
4. 相关数学知识

What is Elliptic Curve?

椭圆曲线是一种在数学和密码学中广泛使用的几何对象。由于其描述方程和求椭圆周长的式子类似, 因此得名椭圆曲线

这个特定方程一般形式是：$y^2 = x^3 + ax + b$，其中 a 和 b 是常数。不同的参数 a 和 b 会导致不同形状的椭圆曲线。

椭圆曲线的一个重要性质是它们的点可以进行 "加法"运算。通过对两个点进行**特殊的加法运算**，可以得到一条新的曲线上的点。这种加法运算的定义是**特定的**，并且有一些规则。

椭圆曲线在密码学中的应用很多，其中最著名的是用于公钥密码学中的椭圆曲线加密（ECC）。
在 ECC 中，椭圆曲线的某个点 `G` 被用作公开的公钥，而 "加法"运算过程用于生成互相共享交换的 key，从而实现加密和解密。由于 ECC 需要比其他加密算法使用更短的密钥长度，因此它通常被认为是一种更加安全和高效的加密方法。

困难问题

首先, 任何加密算法可以实现加密的原因都是因为存在一个难解的困难问题, 这个问题的困难程度决定了这个算法的加密强度

对于 ECC 椭圆曲线上的两个点 `P` 和 `Q`， 任意整数 `k` ：

$$Q = kP$$

困难问题：

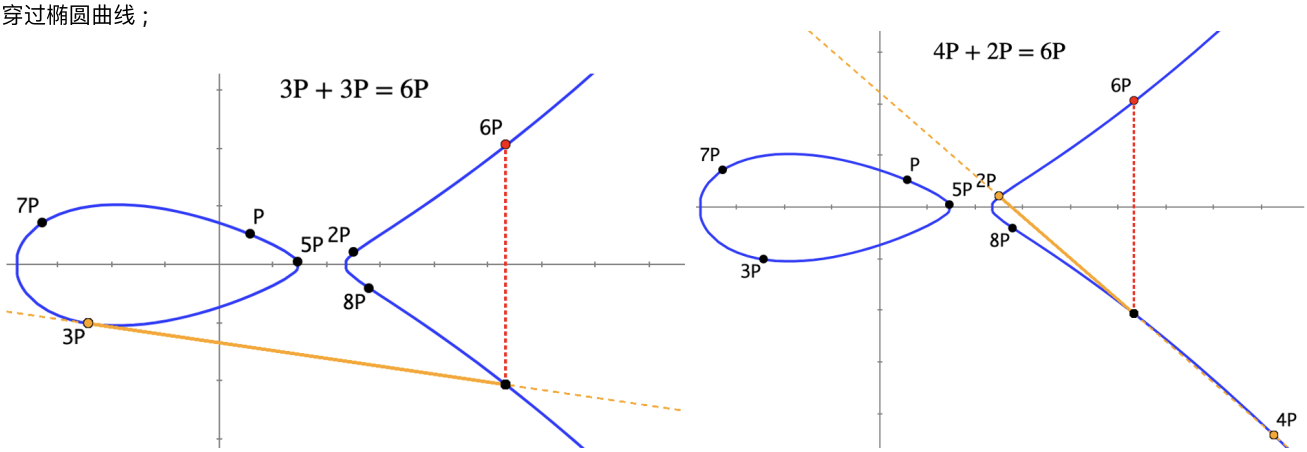
- 给定 `k` 和 `P`，根据加法法则，计算 `Q` 很容易 (对应验证过程)。
- 但给定 `P` 和 `Q` ，求 `k` 非常困难 (在 ECC 的实际应用中，质数 `p` 取的非常大，想穷举出 k 非常困难)。

运算

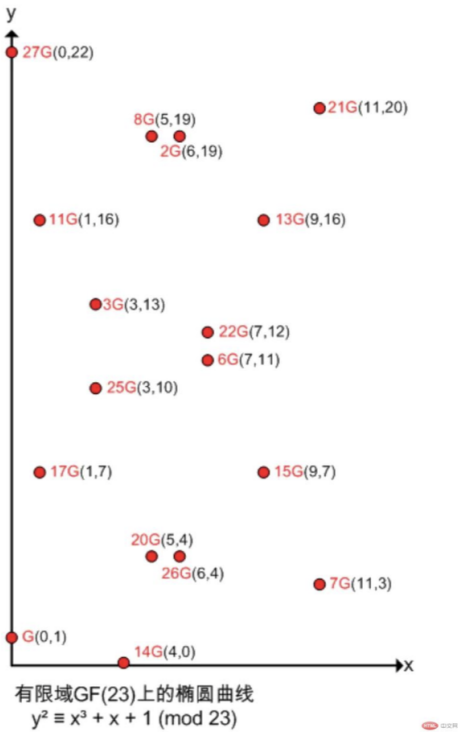
下面我们定义椭圆曲线上的 `加法运算` 为: 将两个点 `相加` 得到第三个点的操作

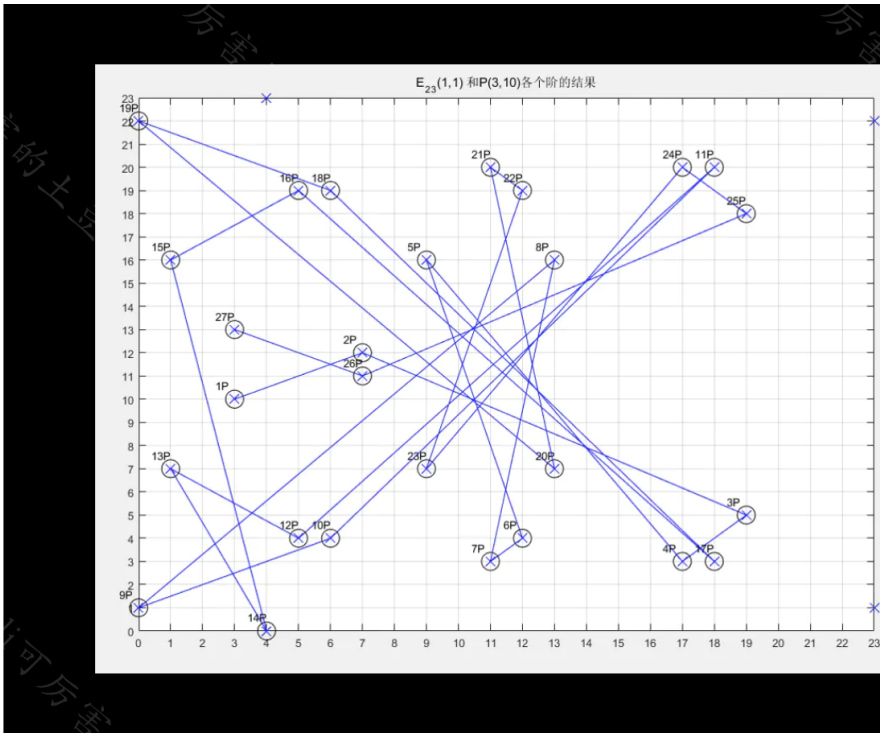
更具体地，要计算 A ✚ B , 过 A / B 做一条直线穿过椭圆曲线, 然后做一下对 `x 轴` 的映射：

- 右图: 2P + 4P = 6P ;
- 左图: 3P + 3P = 6P ; 3P+3P 可以理解为点 `3P` 和 `距离 3P 无限近的另一个点` 连线之后穿过椭圆曲线 , 即在 `3P` 这个点上做一条**切线**穿过椭圆曲线 ;



直观感受





观察上图, 是有限域 $GF(23)$ 上 $y^2 = x^3 + x + 1 \pmod{23}$ 的几个点 :

- $G(0,1)$ 是基点, 其余的点都可以在 G 上做相应的运算
- 观察右图, 可以发现 1P 2P 3P 都非常乱七八糟没有规律, 这就是这个算法困难的地方 , 私钥及其难求
- 有限域还没讲, 这里建立直观理解即可

离散化 -> 有限域

椭圆曲线是连续的，是在实数域上定义的曲线，并不适合加密。我们要把椭圆曲线变成离散的点。把椭圆曲线映射到一个有限域上! 这个过程就是椭圆曲线的离散化或者说在有限域上的定义。

> 为什么连续函数不适合加密 ; 因为实数是连续的，知道结果就可以使用逆运算求解。函数值随着自变量的变化而连续变化，这意味着它们在输入值的微小变化下，函数值也会随之微小地变化。这种连续性使得函数的变化可以被攻击者通过输入输出对（也称为“明文-密文对”）的分析所探测到。这种攻击技术称为“差分分析”
>
> 而且计算连续函数时, 计算机的浮点数会丢失部分精度
>
> 相比之下，离散函数在输入值发生微小变化时其函数值通常也会发生较大的变化，这使得攻击者难以进行差分分析，因此离散函数更适合用于加密算法中。常见的离散函数加密算法包括RSA、椭圆曲线密码学等。

为了在计算机上进行加密和其他操作，需要将其映射一个有限域上。

> 域是一个可以在其上进行 **加法、减法、乘法、和除法运算**，而结果不会超出域的集合
> 如: 有理数集合、实数集合、复数集合都是域，但整数集合不是。
> (很明显，使用除法得到的分数或者小数已超出整数集合)。

有限域

如果域 **F (Field)** 只包含有限个元素，则称其为有限域。

有限域中元素的个数称为有限域的**阶 (order)**。

每个有限域的阶必为素数的幂，即有限域的阶可表示为 p^n (p 是素数，n是正整数)， 该有限域通常称为 Galois 伽(ga)罗瓦域 (Galois Fields)，记为 $GF(pn)$。

椭圆曲线是在实数域上定义的曲线，但为了在计算机上进行加密和其他操作，需要将其映射到一个有限域上。这个过程就是椭圆曲线的离散化或者说在有限域上的定义。

具体地，设椭圆曲线为 E，其定义方程为 $y^2 = x^3 + ax + b$，其中 a 和 b 是实数。为了将其离散化，我们需要选择一个有限域 F_q，其中 q 是一个素数或者素数幂，表示有限域的大小。

举例 : 考虑 $GF(5)$，它包含 5 个元素 `{ 0, 1, 2, 3, 4}` 。在 $GF(5)$ 中，加法和乘法可以定义为 `模 5` 意义下的加法和乘法。
例如，2 + 3 == 0（因为2 + 3 = 5，而5 模 5等于0），4 × 3 == 2（因为4×3=12，而 12 mod 5 等于 2）, 这些计算结果都在 `Field` 里面。

然后，我们需要将实数域上的点 (x, y) 映射到 F_q 中的点 (x', y')。具体地，我们可以选择一个整数 p (如 `13`)，使得 p 能够整除 q，并将 (x, y) 映射为 (x', y')，其中

$$
\begin{aligned}
x' &= x \mod p \quad \text{如 } 2 = (28 \mod 13) \\
y' &= y \mod p \quad \text{如 } 10 = (36 \mod 13)
\end{aligned}
$$

这个过程称为模 p 运算。因为 p 能够整除 q，所以 (x', y') 仍然在 F_q 中。

现在，我们可以在 F_q 上定义离散化后的椭圆曲线 E'，其定义方程为 $y'^2 = x'^3 + ax' + b$。E' 上的点仍然满足椭圆曲线的性质

有限域上的椭圆曲线

在域的定义基础上，作如下修改:

1. 定义模 p 加法和模 p 乘法(加或乘的结果超过 p 时，模 p 去取余数，p 为素数)。
2. 集合内的元素经过加法和乘法计算，结果仍然在集合内。
3. 在有限域上，由于点的数量是有限的，所以我们可以定义椭圆曲线上的倍乘运算，即将一个点乘以一个整数 k，得到另一个点。这个运算可以通过重复进行加法来实现，即将一个点不断地加上自己，直到加了 k 次即得 kG。
4. 计算符合交换率、结合率、分配率。
5. 加法和乘法有单位元素(所有的集合内的值都有对应的负数，所有集合内非零值都有倒数)。
6. 有限域上的椭圆曲线上的点数量是有限的。这个数量由有限域的大小确定，通常表示为p。该数量包括一个特殊的“无穷远点”（Infinity Point），它可以视为一个虚拟的点，用于计算椭圆曲线上的加法和减法操作。

有限域上的椭圆曲线运算

$$
\begin{aligned}
x_3 &\equiv k^2 - x_1 - x_2 \pmod p \\
y_3 &\equiv k(x_1 - x_3) - y_1 \pmod p \\
&\text{若 } P = Q, \text{ 则 } k = \frac{3x_1^2 + a}{2y_1} \pmod p \\
&\text{若 } P \neq Q, \text{ 则 } k = \frac{y_2 - y_1}{x_2 - x_1} \pmod p
\end{aligned}
$$

假设 $y^2 \equiv x^3 + x + 1 \pmod{23}$, 设基点 A 为 $(0, 1)$

① 求 2A :

解 : 观察函数 , 根据椭圆曲线定义 $y^2 = x^3 + ax + b$, 此解析式 $a = b = 1$
求 $2A$ 即求 $A + A$
$\because P \equiv Q, \quad \therefore k = \frac{3 \cdot 0^2 + 1}{2 \cdot 1} == \frac{1}{2} \mod 23$

如何求 $\frac{1}{2} \mod 23$?

$$
\begin{aligned}
\text{设} \quad n &\equiv \frac{1}{2} \mod 23 \\
2n &\equiv 1 \mod 23 \quad \text{即} \\
2n \mod 23 &= 1 \\
&=> 2n = 24 \\
n &= 12
\end{aligned}
$$

或者这样: 在模 23 意义下，24 和 1 是等价的 :

$$
\frac{1}{2} \mod 23 = (24 \cdot \frac{1}{2}) \mod 23 = 12 \mod 23 = 12
$$

故 $k == 12$, 代入 (x_3, y_3) 的计算公式, 求出

$$
\begin{aligned}
&(x_3, y_3) = 2A \\
&= (12^2 - 0 - 0 \mod 23 = 6, \quad 12(0 - 6) - 1 \mod 23 = ?) \\
&= (6, \quad -73 \mod 23) \\
&= (6, \quad 19)
\end{aligned}
$$

$2A$ 计算出来之后, 就可以依次计算 $3A$, $4A$ .....

标量乘法

$2A$ 算得都费劲, $3A$, $4A$, $5A$ ..... 就更难算了, 可以使用标量乘法

先做倍数再做加法。假设 $n = 151$，其对应的二进制是 10010111。而二进制数字可以转化为:

$$\begin{aligned} 151 &= 10010111 \\ &= 1 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \\ &= 2^7 + 2^4 + 2^2 + 2^1 + 2^0 + \end{aligned}$$

$$\text{即} \quad 151P = 2^7P + 2^4P + 2^2P + 2^1P + 2^0P$$

先算 $2P$, 再算 $4P$ 即 2^2P ....

应用 DH 密钥交换

a(bG) = b(aG)