Discrete Log Problem on Elliptic Curves

Tengyi Xu

April 9 2024

Groups: Definition

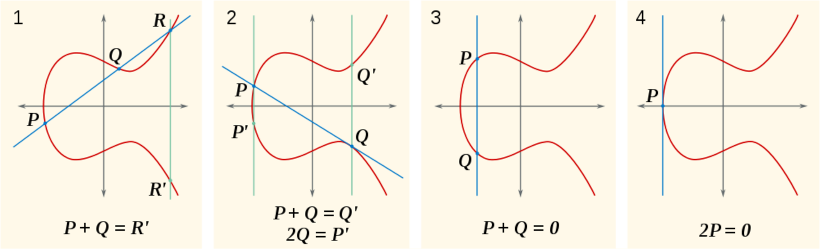
An abelian group is a set G with a binary operation $+$ such that all of the following hold:

- Associativity: $\forall a, b \in G, (a+b)+c = a+(b+c)$
- Identity: $\exists e \in G$ such that $\forall a \in G$, $a+e = e+a = a$
- Inverses: $\forall a \in G, \exists -a \in G$ such that $a+(-a) = (-a)+a = e$
- Commutativity: $\forall a, b \in G, a+b = b+a$

Elliptic Curves: Definitions

- An elliptic curve can be described as the set of solutions (x, y) to an equation of the form $y^2 = x^3 + ax + b$
(over a field K of characteristic $\neq 2, 3$)
- The set of points on an elliptic curve, along with one additional "point at infinity" $\mathcal{O}$, form an abelian group $E(K)$.

Elliptic Curves: Abelian Group



https://upload.wikimedia.org/wikipedia/commons/thumb/a/ae/ECClines-2.svg/1000px-ECClines-2.svg.png

Elliptic Curves: Discrete Log Problem

Given $P, Q \in E(K)$ such that $Q = P + P + \cdots + P = kP$, the **discrete logarithm problem for elliptic curves** describes calculating k.

This is useful for elliptic curve cryptography, since k can be a secret key. However, k can be found using Pollard's Rho Algorithm!

Pollard's Rho Algorithm for Elliptic Curves: Set up

Goal: Find k in $kP = Q$, in an $E(K)$, where K is a finite field.

Partition $E(K)$ into three subsets S_1, S_2, S_3 of around the same size. Define a function

$$A_{i+1} = g(A_i) = \begin{cases} A_i + P, & \text{if } A_i \in S_1 \\ 2A_i, & \text{if } A_i \in S_2. \\ A_i + Q, & \text{if } A_i \in S_3. \end{cases}$$

Like the function used for Pollard's Rho Algorithm for factorization, this is used to generate a pseudorandom sequence which gives us a higher chance of finding a collision.
We can pick A_0 to be some multiple of P.
Notice that this means all points are in the form $A_i = a_i P + b_i Q$.

Pollard's Rho Algorithm for Elliptic Curves: Result

Goal: Find k in $kP = Q$, in an $E(K)$, where K is a finite field.

Recurse until we find some $A_i = A_j$
This will happen since $E(K)$ has only finitely many points, so the sequence ultimately repeat after enough iterations.

$$a_i P + b_i Q = a_j P + b_j Q$$
$$(a_i - a_j) P = (b_j - b_i) Q$$

Let n be the smallest number such that $nP = \mathcal{O}$.
If $b_j - b_i$ is invertible $\pmod n$, then we get $\frac{a_i - a_j}{b_j - b_i} P = Q$.

Like, the factoring algorithm, we'll need $O(\sqrt{n})$ iterations to make the probability for a match around 50%.