Reference

Friday, April 26, 2024 1:21 PM

1. https://andrea.corbellini.name/2015/05/17/elliptic-curve-cryptography-a-gentle-introduction/
2. https://andrea.corbellini.name/2015/05/23/elliptic-curve-cryptography-finite-fields-and-discrete-logarithms/
3. https://andrea.corbellini.name/2015/05/30/elliptic-curve-cryptography-ecdh-and-ecdsa/
4. https://www.rareskills.io/post/elliptic-curve-addition
5. https://www.rareskills.io/post/elliptic-curves-finite-fields
6. https://www.rareskills.io/post/bilinear-pairing

Elliptic curves

Tuesday, May 14, 2024 1:38 PM

Def: The set of points described by

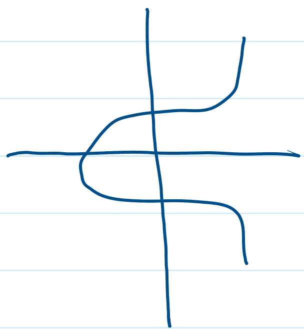
Δ : discriminant

$$y^2 = x^3 + ax + b \quad , \text{ where } \quad 4a^3 + 27b^2 \neq 0$$

Weierstrass Curves

exclude singular curves
(double or triple root x_0
and $(x_0, 0)$ is a singular
point)

hard to define addition

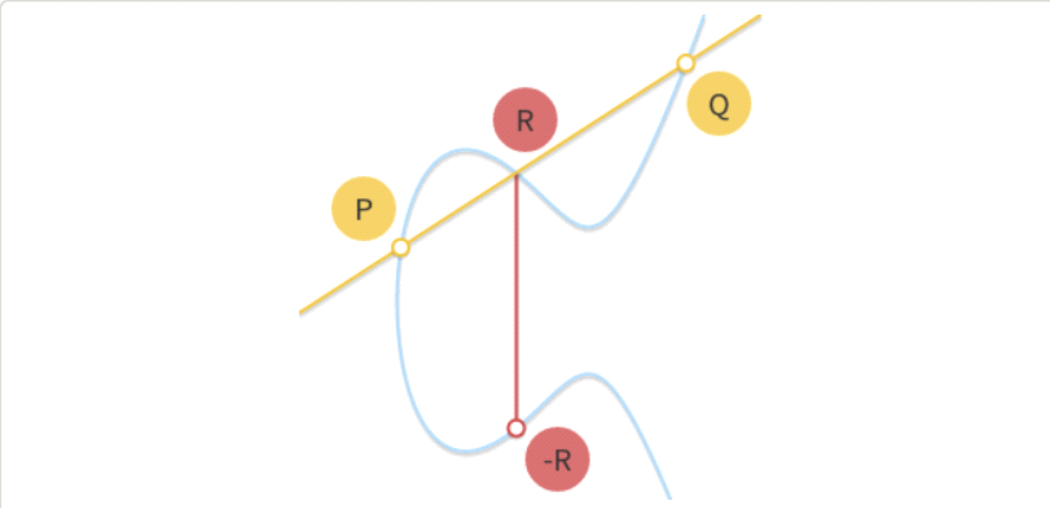


Symmetric about x-axis

Def: Point at infinity O (acts as identity in group structure)

Group Law for EC

- Elements in this group is just points on EC
- Identity is O
- Inverse of point P is $-P$ (symmetric about x-axis)
- Addition :



Draw the line through P and Q. The line intersects a third point R. The point symmetric to it, $-R$, is the result of $P+Q$.

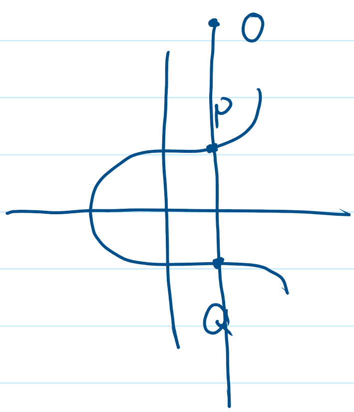
Claim: Group of EC is abelian group.

Intuition: $P+Q = Q+P$ cause they share the same straight line.

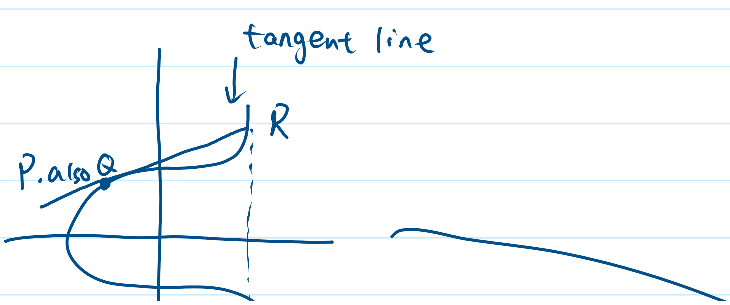
Special cases:

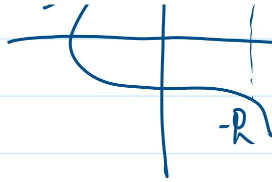
1. $P=0$ or $Q=0$ → identity

2. $P=-Q$



3. $P=Q$





- As long as we do not pick a perfectly vertical line, if we intersect two points in an elliptic curve, then we will also intersect a 3rd point on the elliptic curve.
- If a straight line crosses an elliptic curve at exactly two points, then it must be perfectly vertical.

$$P + P = \underbrace{2P}$$

Scalar multiplication

$$nP = \underbrace{P + P + P + \cdots + P}_{n \text{ times}}$$

nP can be computed efficiently using "double and add" algorithm:

Example: $n = 151$

$$151P = 2^7P + 2^4P + 2^2P + 2^1P + 2^0P$$

result = 0

P	add
$2P$	add
$4P$	add
$8P$	
$16P$	add
$32P$	
$64P$	
$128P$	add

$8 + 5 = 13$ computations $\quad O(\log n)$

$\uparrow$ double $\quad \uparrow$ add

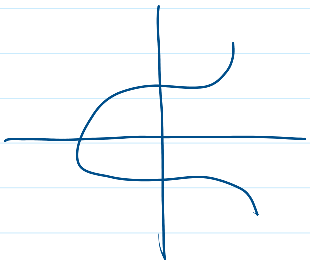
Remember, EC over R is just an abelian group. Group theory simplifies EC.

Elliptic curves over finite field

Tuesday, May 14, 2024 1:39 PM

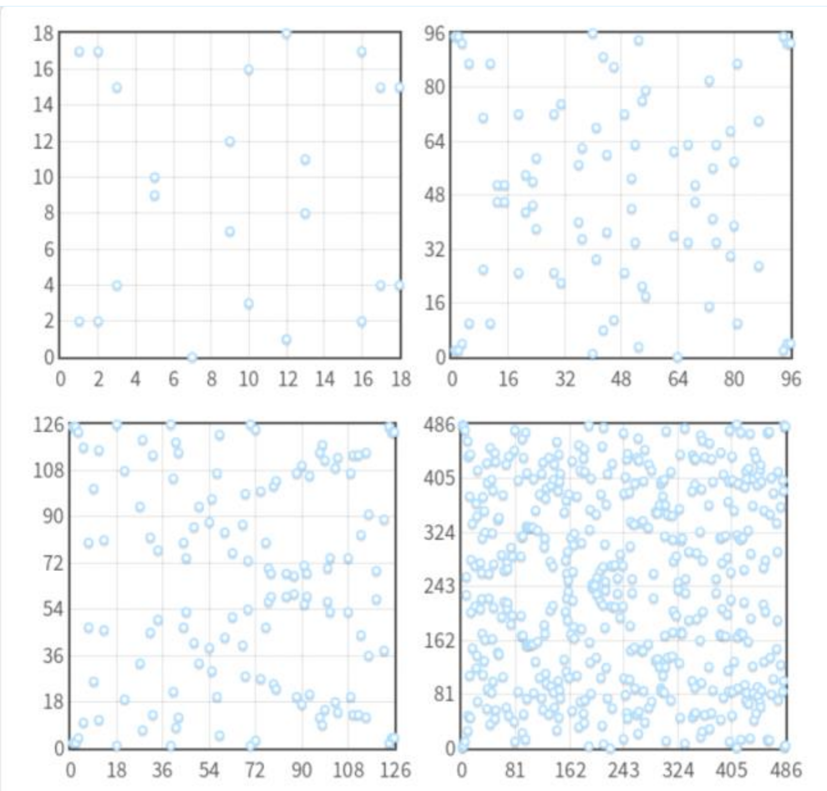
EC over $\mathbb{R}$:

$$\{(x,y) \in \mathbb{R}^2 \mid y^2 = x^3 + ax + b,\ 4a^3 + 27b^2 \neq 0\} \cup \{O\}$$



EC over $\mathbb{F}_p$:

$$\{(x,y) \in \mathbb{F}_p^2 \mid y^2 = x^3 + ax + b \pmod p,$$
$$4a^3 + 27b^2 \neq 0 \pmod p\} \cup \{O\}$$



The curve $y^2 \equiv x^3 - 7x + 10 \pmod p$ with $p = 19, 97, 127, 487$. Note that, for every x, there are at most two points. Also note the symmetry about $y = p/2$.

Claim: EC over $\mathbb{F}_p$ is abelian group

Even better, EC over $\mathbb{F}_p$ is cyclic group !!! ← recall that any G with $|G| = p$ is cyclic

Even better, EC over $\mathbb{F}_p$ is cyclic group !!! ← recall that any G with $|G| = p$ is cyclic

→ generator G, operator is $+$

→ every non-identity $(G \neq O)$ element is generator

→ points can be written as nG

→ $\{nG\}$ covers every points on EC

Corollary 6.12. *Let $|G| = p$ with p a prime number. Then G is cyclic and any $g \in G$ such that $g \neq e$ is a generator.*

Proof.

Corollary 6.12 suggests that groups of prime order p must somehow look like $\mathbb{Z}_p$.

source: https://abstract.ups.edu/abstract.ups.edu/aata/cosets-section-lagranges-theorem.html

Ethereum precompiles use bn128 (bits of security) (bn254 (# bits of prime p))   https://hackmd.io/@jpw/bn254

field_modulus =
21888242871839275222246405745257275088696311157297823662689037894645226208583

$y^2 = x^3 + 3 \pmod{\text{field\_modulus}}$ ← prime p ($y^2 = x^3 + b$ is BN-curve equation)

(curve_order can be computed by Schoof's algorithm: https://en.wikipedia.org/wiki/Schoof%27s_algorithm)

```
>>> from py_ecc.bn128 import G1, multiply, add, eq, neg
>>> G1
(1, 2)
>>> add(G1, G1)
(1368015179489954701390400359078579693043519447331113978918064868415326638035, 99181100
51302171585080402603319702774565515993150576347155970296011118125764)
>>> add(G1, G1) == multiply(G1, 2)
True
```

field_modulus != curve_order

```
>>> from py_ecc.bn128 import curve_order, field_modulus, G1, multiply, eq
>>> x = 5
>>> multiply(G1, x) == multiply(G1, x + curve_order)
True
>>> multiply(G1, x) == multiply(G1, x + field_modulus)
False
>>>
```

Can encode rational numbers : $\frac{2}{3} = 2 \cdot 3^{-1}$, $\frac{4}{3} = 4 \cdot 3^{-1}$, $\frac{2}{3} + \frac{4}{3} = 2$

always exists in a field

```
>>> two_over_three = (2 * pow(3, -1, curve_order)) % curve_order
>>> four_over_three = (4 * pow(3, -1, curve_order)) % curve_order
>>> add(multiply(G1, two_over_three), multiply(G1, four_over_three)) == multiply(G1, 2)
True
>>>
```

ECDLP and ECDH

Wednesday, May 15, 2024 9:32 PM

<u>ECDLP</u>: Given $Q = nP$, can't solve for n

Recall that DLP says given $h = g^x \bmod p$ it is hard to solve for x. This is the foundation for Diffie-Hellman. Similarly, ECDLP is the foundation for ECDH:

<u>Alice</u>	Generator G of EC	<u>Bob</u>
Generate random sk_A	---	sk_B
Compute $pk_A = \underline{sk_A \cdot G}$	---	$pk_B = \underline{sk_B \cdot G}$

$\xrightarrow{\qquad}$ exchange pk $\xleftarrow{\qquad}$

Compute $sk_A \cdot pk_B = sk_A \cdot sk_B \cdot G$ --- $sk_B \cdot pk_A = sk_B \cdot sk_A \cdot G$

<u>Correctness</u>: $(sk_A \cdot sk_B) \cdot G = (sk_B \cdot sk_A) \cdot G$

Attack: https://www.ret2basic.me/2024/04/12/elliptic-curve-attacks-small-subgroup.html

Basic ZK with EC

Thursday, May 16, 2024 6:37 PM

Example 1: Prover: I know x, y s.t. $x+y = 15$

Verifier: prove it, but don't send me x and y

Prover computes xG and yG, send to verifier

Verifier checks if $xG + yG = 15G$

Example: Prover: I know x s.t. $23x = 161$

Verifier: prove it, but don't send me x

Prover computes $161 \cdot 23^{-1} G$, send to verifier

Verifier checks if $23 \cdot (161 \cdot 23^{-1} G) = 23 \cdot 23^{-1} \cdot 161 G = 161 G$

In reality ZK property is achieved using bilinear pairing ("multiply 2 EC points from different dimensions") We will cover that next week.

Further study

Thursday, May 16, 2024 6:26 PM

Read py_ecc source code and play with it https://github.com/ethereum/py_ecc