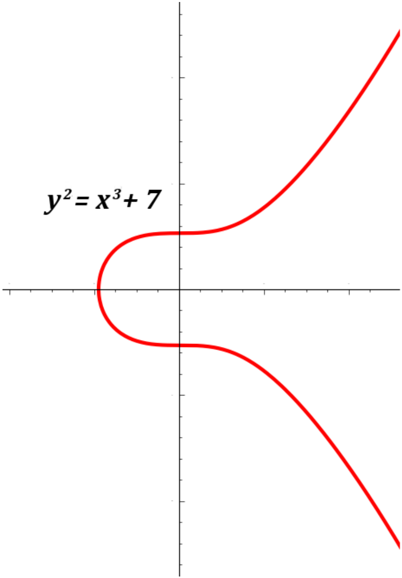
Elliptic Curves and ECDSA
Bonus Material!

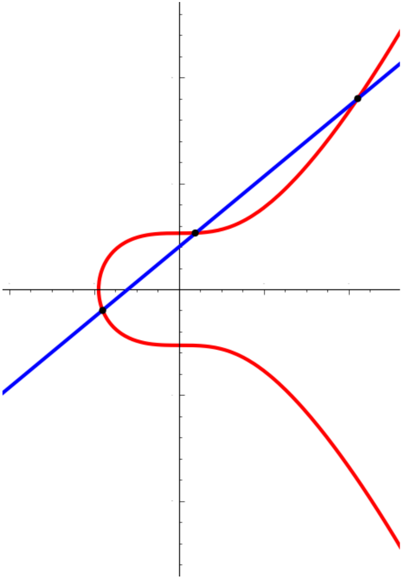
Brendan Cordy

!!Con 2016

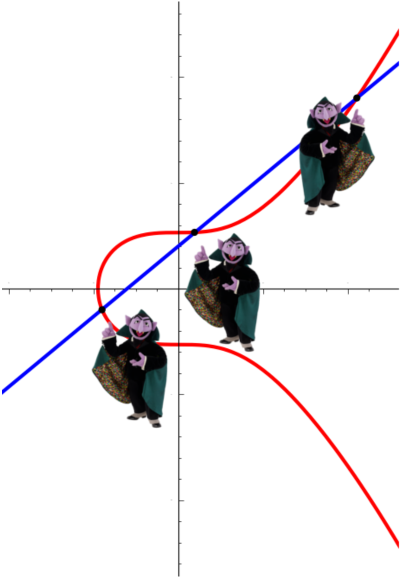
Elliptic Curves



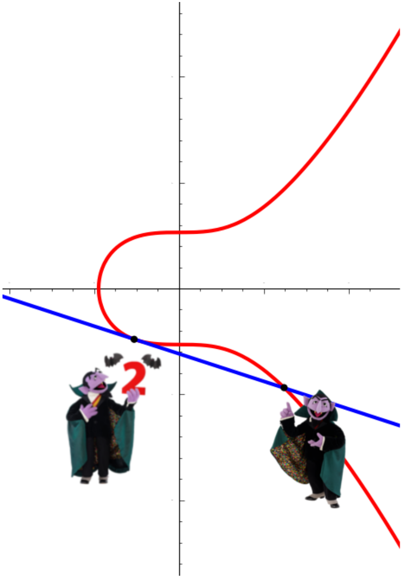
Elliptic Curves



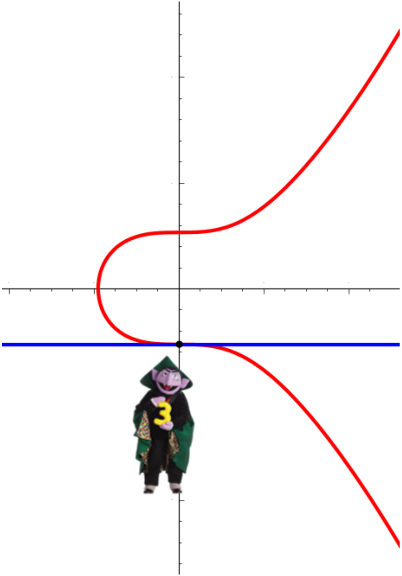
Counting Intersections



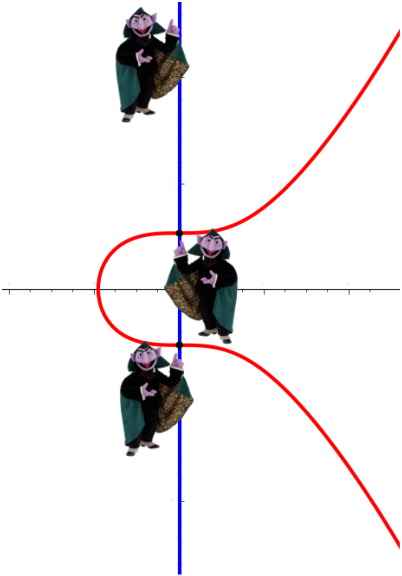
Double Intersections



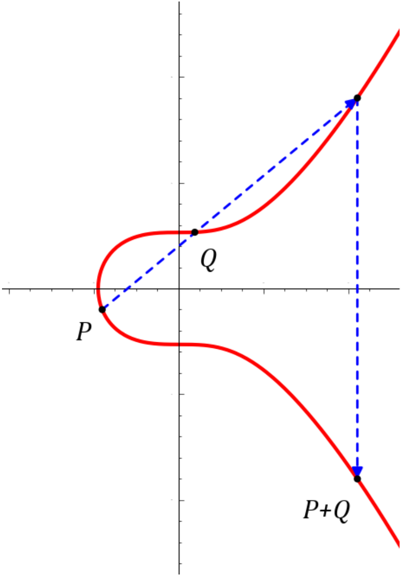
Triple Intersections



The Point at Infinity



Adding Points



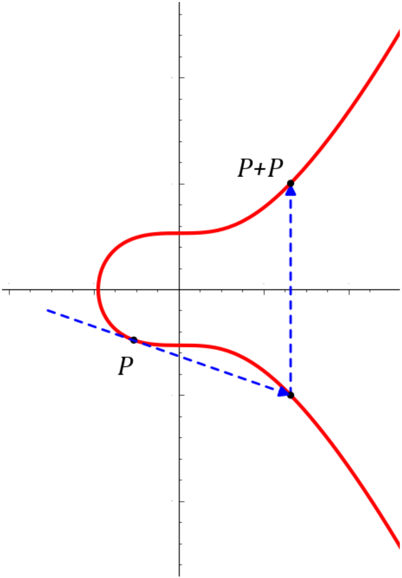
Adding Points

- Strange, but has all the nice properties that addition should. Explicitly computing the sum takes a handful of operations.

$$\text{Let } \lambda = \frac{y_Q - y_P}{x_Q - x_P}, \text{ then}$$

$$x_{P+Q} = \lambda^2 - x_P - x_Q$$
$$y_{P+Q} = \lambda(x_Q - x_{P+Q}) - y_P$$

Point Doubling



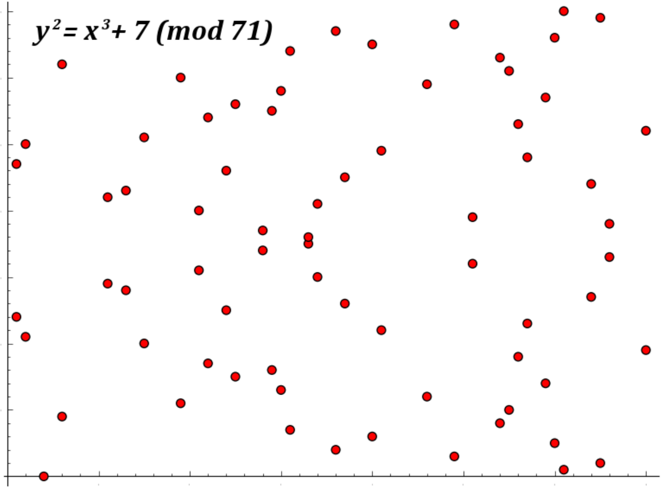
Point Doubling

- Again, explicitly computing the coordinates takes a handful of operations.

Point Doubling

- Again, explicitly computing the coordinates takes a handful of operations.
- In fact, we can use the same formulas, but with $\lambda = \frac{3{x_P}^2}{2y_P}$, so doubling takes about the same amount of time as adding distinct points.

Everything Works in $\mathbb{F}_p$



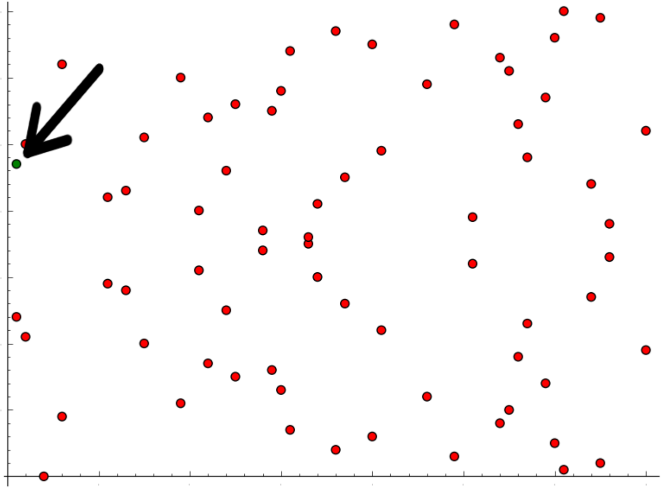
Everything Works in $\mathbb{F}_p$

- The graph is pretty nuts, but everything still works, even the explicit formulas for addition and doubling, if you do all the arithmetic mod p.

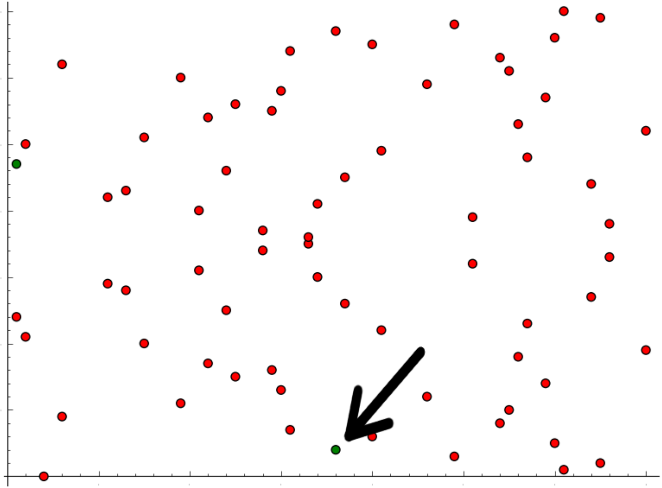
Everything Works in $\mathbb{F}_p$

- The graph is pretty nuts, but everything still works, even the explicit formulas for addition and doubling, if you do all the arithmetic mod p.
- $p = 2^{256} - 4294966319$ in the secp256k1 standard used in Bitcoin, and the curve has about that many points on it.

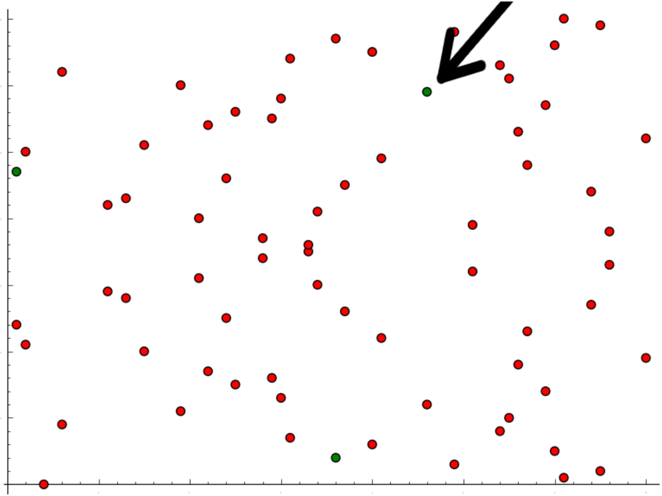
$P(4,47)$ on $y^2 = x^3 + 7\,(mod\,71)$



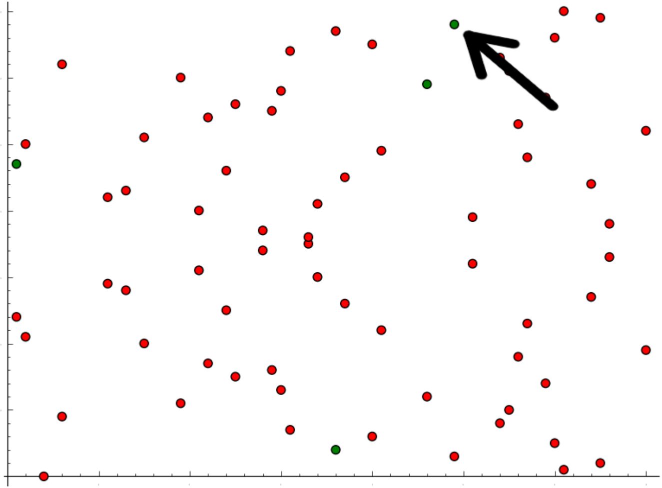
$P + P$



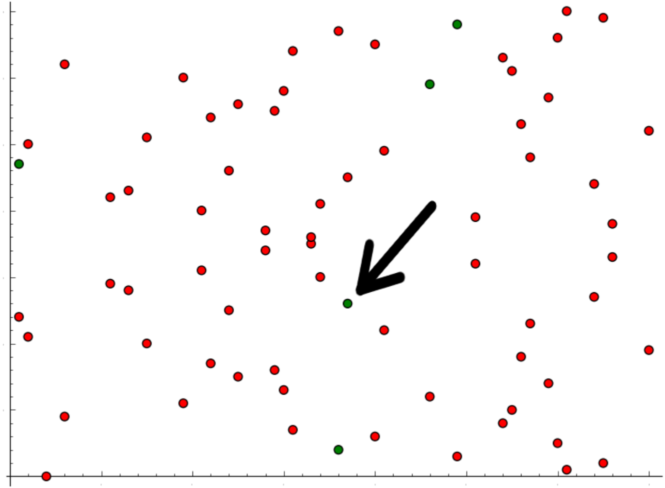
$P + P + P$



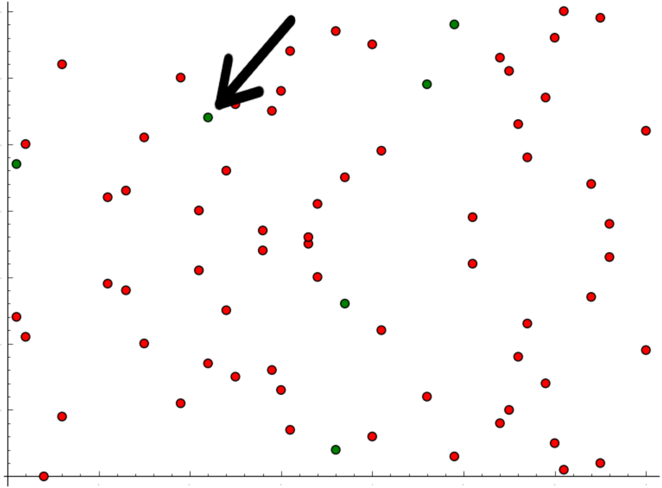
$$P + P + P + P$$



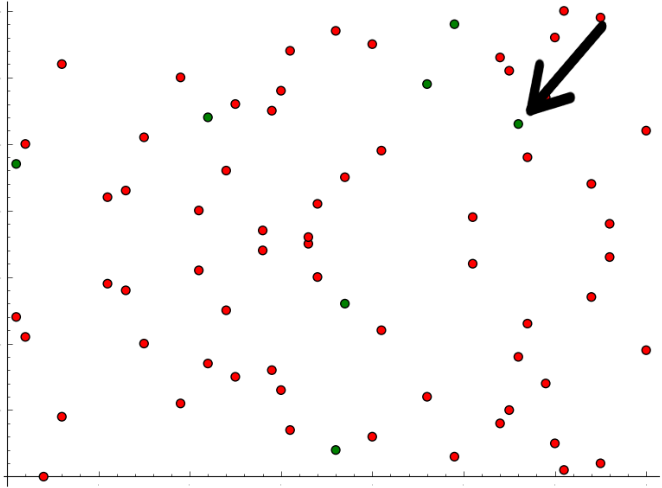
$$P + P + P + P + P$$



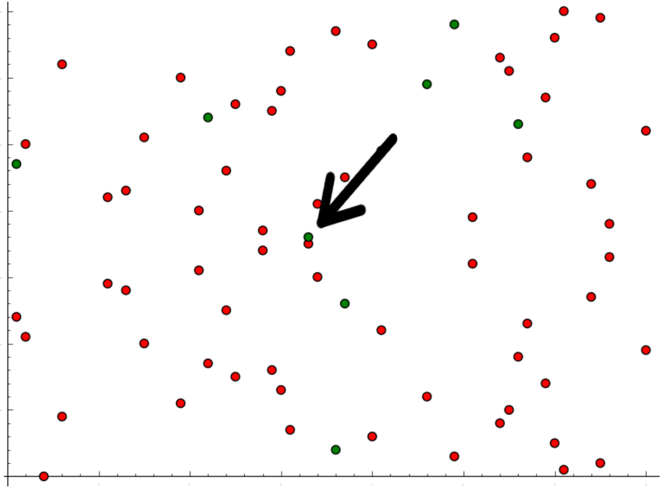
$$P + P + P + P + P + P$$



$P + P + P + P + P + P + P$



$P + P + P + P + P + P + P + P$



Point Multiplication

- Let $nP = \overbrace{P + P + \ldots + P}^{n \text{ times}}$.

Point Multiplication

- Let $nP = \overbrace{P + P + \ldots + P}^{n \text{ times}}$.
- There is a point P whose multiples cycle through all points on the curve (and it's given in the secp256k1 standard).

Point Multiplication

- Let $nP = \overbrace{P + P + \ldots + P}^{n \text{ times}}$.
- There is a point P whose multiples cycle through all points on the curve (and it's given in the secp256k1 standard).
- Finding nP appears to require n additions. However, there is a clever way to do it.

Peasant Multiplication

$$179P = 128P + 32P + 16P + 2P + P$$

Peasant Multiplication

$$179P = 128P + 32P + 16P + 2P + P$$

- How many doublings? $\lfloor \log_2(n) \rfloor$

Peasant Multiplication

$$179P = 128P + 32P + 16P + 2P + P$$

- How many doublings? $\lfloor \log_2(n) \rfloor$
- How many additions? $\leq \lfloor \log_2(n) \rfloor$

Peasant Multiplication

$$179P = 128P + 32P + 16P + 2P + P$$

- How many doublings? $\lfloor \log_2(n) \rfloor$
- How many additions? $\leq \lfloor \log_2(n) \rfloor$
- $n \to 2\log_2(n)$: Exponential speedup!

Generating Key-Pairs

- Take a random 256-bit integer d, and use peasant multiplication to compute $Q = dP$.

Generating Key-Pairs

- Take a random 256-bit integer d, and use peasant multiplication to compute $Q = dP$.
- What if instead we knew P and Q, but wanted to compute d? ¯_(ツ)_/¯

Generating Key-Pairs

- Take a random 256-bit integer d, and use peasant multiplication to compute $Q = dP$.
- What if instead we knew P and Q, but wanted to compute d? ¯_(ツ)_/¯
- The point Q is the public key, while the number d is kept secret.

Digital Signatures (ECDSA)

- If you know that $Q = dP$, you can solve...

$$
\begin{aligned}
aP + bQ &= xP \\
aP + bdP &= xP \\
(a + bd)P &= xP \\
x &= a + bd
\end{aligned}
$$

Digital Signatures (ECDSA)

- If you know that $Q = dP$, you can solve...

$$\begin{aligned} aP + bQ &= xP \\ aP + bdP &= xP \\ (a + bd)P &= xP \\ x &= a + bd \end{aligned}$$

- Anyone can check whether solution works, without knowledge of d.

Digital Signatures (ECDSA)

- To sign a transaction, let h be a hash of the transaction, k be a nonce, and r be the x-coordinate of kP. Solve $hP + rQ = xkP$.

Digital Signatures (ECDSA)

- To sign a transaction, let h be a hash of the transaction, k be a nonce, and r be the x-coordinate of kP. Solve $hP + rQ = xkP$.
- The transaction is sent out along with the triple (Q, r, x). Nodes in the network will hash the transaction to obtain h, and verify the equation holds.

Elliptic Curves Recap

- From $y^2 = x^3 + 7 \pmod{2^{256} - 4294966319}$ and P we can generate a key-pair (s, Q).

Elliptic Curves Recap

- From $y^2 = x^3 + 7 \pmod{2^{256} - 4294966319}$ and P we can generate a key-pair (s, Q).
- Using ECDSA, the network can verify that a transaction originated from an individual who knows the value s.

Elliptic Curves Recap

- From $y^2 = x^3 + 7 \pmod{2^{256} - 4294966319}$ and P we can generate a key-pair (s, Q).
- Using ECDSA, the network can verify that a transaction originated from an individual who knows the value s.
- The secret key s is a PIN used for spending.