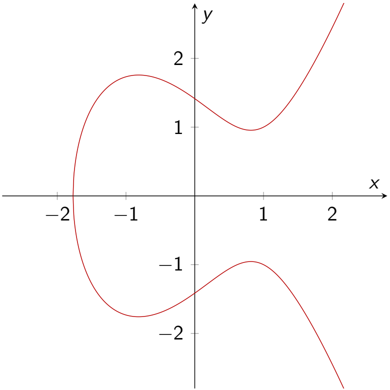
Bitcoin, Blockchain and Cryptoassets
Elliptic Curves and ECDSA

Prof. Dr. Fabian Schär
University of Basel

Release Ver.: (Local Release)
Version Hash: (None)
Version Date: (None)

License: Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International

Example of an Elliptic Curve



Elliptic curve with $a = -2$, $b = 2$

Weierstrass equation:

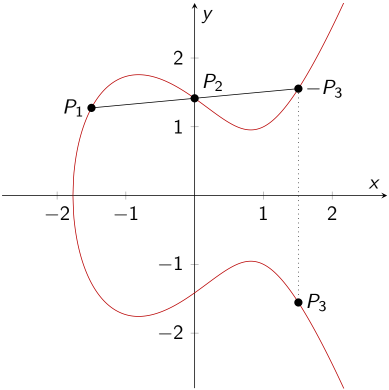
$$y^2 = x^3 + ax + b$$

Non-singularity condition:

$$4a^3 + 27b^2 \neq 0$$

Addition of Two Points

$$P_1 = (-1.5, \sqrt{1.625})$$
$$P_2 = (0, \sqrt{2})$$
$$P_3 = P_1 + P_2$$



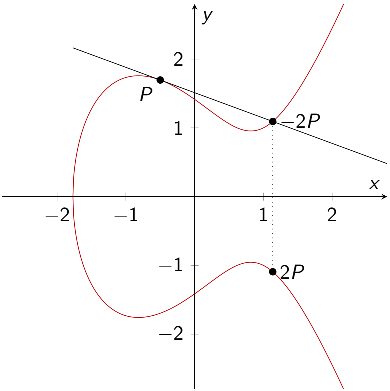
$$s = \frac{y_{P_1} - y_{P_2}}{x_{P_1} - x_{P_2}} = 0.0930$$

$$x_{P_3} = s^2 - (x_{P_1} + x_{P_2}) = 1.5086$$

$$y_{P_3} = s(x_{P_1} - x_{P_3}) - y_{P_1} = -1.5545$$

Point Doubling

$$P = (-0.5, \sqrt{2.875})$$
$$P + P = 2P$$



$$s = \frac{3x_P^2 + a}{2y_P} = -0.3686$$

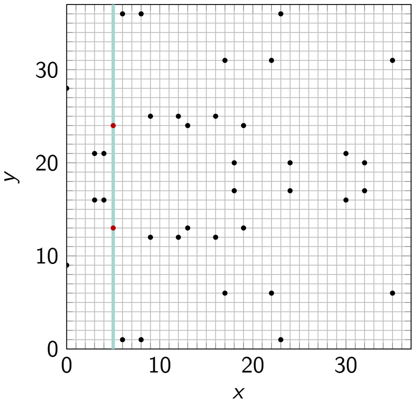
$$x_{2P} = s^2 - 2x_P = 1.1359$$

$$y_{2P} = s(x_P - x_{2P}) - y_P = -1.0926$$

Elliptic Curves over Finite Fields

Bitcoin uses `secp256k1`: $y^2 = x^3 + 7 \ (mod\ p)$ over $\mathbb{F}_p$ where $p = 2^{256} - 2^{32} - 2^9 - 2^6 - 2^4 - 1$.

Simplified example: $y^2 = x^3 + 7 \ (mod\ 37)$ in $\mathbb{F}_{37}$ with $x = 5$



$$
\begin{aligned}
y^2 \ (mod\ 37) &\equiv x^3 + 7 \ (mod\ 37) \\
&\dots \\
5^3 + 7 \ (mod\ 37) &= 132 \ (mod\ 37) \\
&= 21 \\
&\dots \\
y^2 \ (mod\ 37) &\equiv 21 \\
13^2 \ (mod\ 37) &\equiv 21 \\
24^2 \ (mod\ 37) &\equiv 21
\end{aligned}
$$

Modular Multiplicative Inverse

For our computations we often need the so-called modular multiplicative inverse.

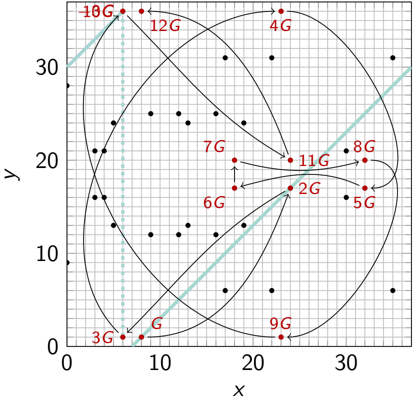
Regular division:
$10/4 = 2.5$

Multiplicative inverse:
$4/4 = 4 \cdot 4^{-1} = 1$
$10 \cdot 4^{-1} = 2.5$

Modular multiplicative inverse:
$\{4 \cdot x\} \ (mod\ 3) = 1$
for $x = 1$
because $4 \ (mod\ 3) = 1$

ECDSA

Simplified example: Elliptic curve of order $N = 39$ with subgroups of the order $n = 13$:

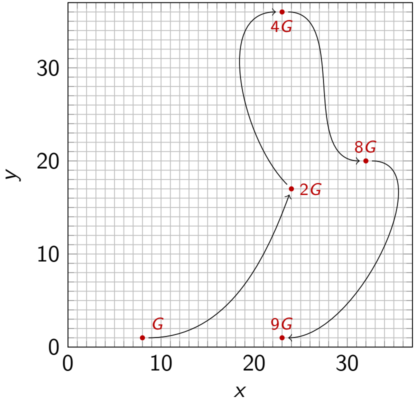


e.g. $G = (8, 1)$**:**
The cyclic subgroup consists of
$\{0, G, 2G, ..., (n-1)G\}$

$$
\begin{aligned}
s &= \{(1-17)\cdot(8-24)^{-1}\}\ (mod\ 37) \\
&= \{(-16)\cdot(30)\}\ (mod\ 37) \\
&= 1 \\
x_{3G} &= \{1-(8+24)\}\ (mod\ 37) \\
&= 6 \\
y_{3G} &= \{1\cdot(8-6)-1\}\ (mod\ 37) \\
&= 1
\end{aligned}
$$

Key Generation

Simplified example: $k_{prv} = 9$ **and** $G = (8, 1)$

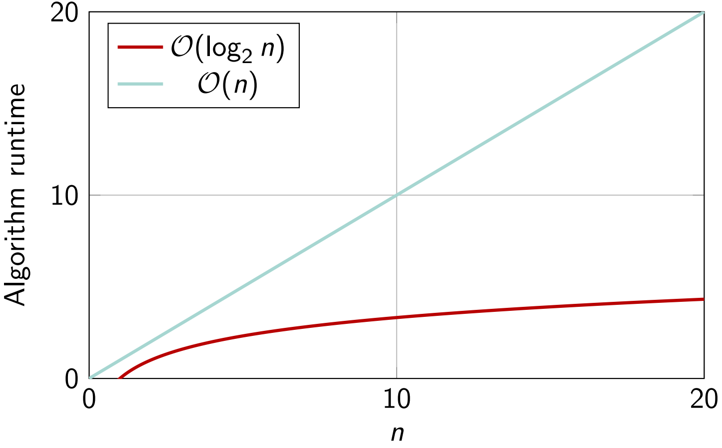


From k_{prv} to K_{pub} using the double and add algorithm:

1. Double: $2 \circ G = 2G$
2. Double: $2 \circ 2G = 4G$
3. Double: $2 \circ 4G = 8G$
4. Add: $8G + G = 9G$

$\rightarrow$ Four steps only! Algorithm runtime: $\mathcal{O}(\log_2 n)$

Elliptic Curve Discrete Logarithm Problem



Example: Signature

Simplified example:

- $y^2 = x^3 + 7$
- $G = (8, 1)$
- $k_{prv} = 9$
- $K_{pub} = (23, 1)$
- $t = 4$

1. Choose random number, e.g. $i = 7$
2. Compute
   a. $P = i \cdot G = 7G = (18, 20)$
   b. $r = x_P \ (mod\ n) = 18 \ (mod\ 13) = 5$
   c. $s = \{i^{-1}(t + r \cdot k_{prv})\} \ (mod\ n)$
      $= \{2 \cdot (4 + 5 \cdot 9)\} \ (mod\ 13)$
      $= 7$
3. Send
   a. $(r, s) = (5, 7)$
   b. $t = 4$
   c. $K_{pub} = (23, 1)$

Example: Verification

Simplified example:

- $y^2 = x^3 + 7$
- $G = (8, 1)$
- $K_{pub} = (23, 1)$
- $t = 4$
- $(r, s) = (5, 7)$

1. Compute
   a. $\{u_1 = (s^{-1}t)\}\ (mod\ n) = 8\ mod\ 13 = 8$
   b. $\{u_2 = (s^{-1}r)\}\ (mod\ n) = 10\ mod\ 13 = 10$
   c. $$\begin{aligned} P &= u_1 \circ G + u_2 \circ K_{pub} \\ &= 8G + 10 \circ (23, 1) \\ &= (32, 20) + (8, 36) \\ &= (18, 20) \end{aligned}$$
2. Check authenticity: $x_P\ mod\ n = r$
   Here: $18\ (mod\ 13) = 5$

→ The private key is never revealed.
→ Python script to signature and verification examples