Scalar Multiplication and Weierstrass Curves

Tanja Lange

EIPSI

Department of Mathematics and Computer Science

Technische Universiteit Eindhoven

tanja@hyperelliptic.org

12.05.2008

Notation

If

$$n = \sum_{i=0}^{l-1} n_i 2^i$$

we write n in binary representation

$$n = (n_{l-1} \dots n_0)_2.$$

E.g.
$n = 35 = 32 + 2 + 1 = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$,
then $35 = (100011)_2$.

The following algorithms are stated in some group $(G, \oplus)$ with neutral element O. Scalar multiplication is denoted by $[n]P = P \oplus P \oplus \cdots \oplus P$ (n terms).

Right–to–Left Binary

`IN:` An element $P \in G$ and a positive integer $n = (n_{l-1} \ldots n_0)_2$.
OUT: The element $[n]P \in G$.

1. $R \leftarrow O$, $Q \leftarrow P$,
2. `for` $i = 0$ `to` $l - 2$ `do`
   (a) `if` $n_i = 1$ `then` $R \leftarrow P \oplus Q$
   (b) $Q \leftarrow [2]Q$
3. `if` $n_{l-1} = 1$ `then` $R \leftarrow P \oplus Q$
4. `return` R

This algorithm computes $[35]P = [2^5]P \oplus [2^1]P \oplus P$.
For $i = j$, at the end of step 2, Q holds $[2^{j+1}]P$ and R holds $[(n_j \ldots n_0)_2]P$.

Left–to–Right Binary

`IN:` An element $P \in G$ and a positive integer $n = (n_{l-1} \dots n_0)_2, n_{l-1} = 1$.
OUT: The element $[n]P \in G$.

1. $R \leftarrow P$
2. $\texttt{for } i = l-2 \texttt{ to } 0 \texttt{ do}$
   (a) $R \leftarrow [2]R$
   (b) $\texttt{if } n_i = 1 \texttt{ then } R \leftarrow P \oplus R$
3. $\texttt{return } R$

This algorithm computes
$[35]P = [2]([2]([2]([2]([2]P))) \oplus P) \oplus P$.
For $i = j$ the intermediate variable R holds $[(n_{l-1} \dots n_j)_2]P$.

Number of additions

- For each 1 in the binary representation of n we compute an addition. On average there are $l/2$ non-zero coefficients.
- In some groups (e.g. elliptic curves) $P \oplus Q$ has the same cost as $P \ominus Q$), so it makes sense to use negative coefficients. This gives signed binary expansions.
- Note that $31 = 2^4 + 2^3 + 2^2 + 2 + 1 = 2^5 - 1$ and so

$$\begin{aligned}
[31]P &= [2]([2]([2]([2]P \oplus P) \oplus P) \oplus P) \oplus P) \\
&= [2]([2]([2]([2]([2]P))))) \ominus P
\end{aligned}$$

- Can always replace two adjacent 1's in the binary expansion by $10\bar{1}$ since $(11)_2 = (10\bar{1})_s$. ($\bar{1}$ denotes -1).

Examples

- By systematically replacing runs of 1's we can achieve that there are no two adjacent bits that are non-zero.
- A representation fulfilling this is called a “non-adjacent form” (NAF).
- NAF's have the lowest density among all signed binary expansions (with coefficients in $\{0, 1, -1\}$).
- $(10010100110111010\underline{11}0)_2 =$
  $(100101001101110\underline{11}0\bar{1}0)_2 =$
  $(10010100110\underline{1111}0\bar{1}0\bar{1}0)_2 =$
  $(10010100\underline{111}000\bar{1}0\bar{1}0\bar{1}0)_2 =$
  $(1001010100\bar{1}000\bar{1}0\bar{1}0\bar{1}0)_2$
- Results no worse, but not necessarily better
  $35 = (100011)_2 = (10010\bar{1})_s$.

Non-Adjacent Form

IN: Positive integer $n = (n_l n_{l-1} \dots n_0)_2$, $n_l = n_{l-1} = 0$.
OUT: NAF of n, $(n'_{l-1} \dots n'_0)_s$.

1. $c_0 \leftarrow 0$
2. `for` $i = 0$ `to` $l - 1$ `do`
   (a) $c_{i+1} \leftarrow \lfloor (c_i + n_i + n_{i+1})/2 \rfloor$
   (b) $n'_i \leftarrow c_i + n_i - 2c_{i+1}$
3. `return` $(n'_{l-1} \dots n'_0)_s$

Resulting signed binary expansion has length at most $l + 1$, so longer by at most 1 bit. On average there are $l/3$ non-zero coefficients.

Also possible to get a representation with the same density from left to right.

NAF – example

1. $c_0 \leftarrow 0$
2. `for` $i = 0$ `to` $l-1$ `do`
   (a) $c_{i+1} \leftarrow \lfloor (c_i + n_i + n_{i+1})/2 \rfloor$
   (b) $n_i' \leftarrow c_i + n_i - 2c_{i+1}$
3. `return` $(n_{l-1}' \dots n_0')_s$

$35 = (00100011)_2$, $c_0 = 0$
$c_1 = \lfloor (0+1+1)/2 \rfloor = 1$, $n_0 = 0+1-2 = -1$
$c_2 = \lfloor (1+1+0)/2 \rfloor = 1$, $n_1 = 1+1-2 = 0$
$c_3 = \lfloor (1+0+0)/2 \rfloor = 0$, $n_2 = 1+0-0 = 1$
$c_4 = \lfloor (0+0+0)/2 \rfloor = 0$, $n_3 = 0+0-0 = 0$
$c_5 = \lfloor (0+0+1)/2 \rfloor = 0$, $n_4 = 0+0-0 = 0$
$c_6 = \lfloor (0+1+0)/2 \rfloor = 0$, $n_5 = 0+1-0 = 1$
$c_7 = \lfloor (0+0+0)/2 \rfloor = 0$, $n_6 = 0+0-0 = 0 \Rightarrow 35 = (10010\bar{1})_s$

Generalizations

- So far all expansions in base 2 (signed or unsigned).
- Generalize to larger base; often 2^w $(w > 1)$. Then the coefficients are in $[0, 2^w - 1]$. Also fractional windows have been suggested.
- w is called window width.
- Assume that $[m]P$ for $m \in [0, 2^w - 1]$ are precomputed.
- Easiest way: just group w bits.
- Sliding windows: Group w bits and skip forward if LSB is 0 (requires only odd integers in $[0, 2^w - 1]$) as coefficients and leads to $l/(w + 1)$ additions).
- If $\ominus$ is cheap, use signed sliding windows; this leads to $l/(w + 2)$ additions.

Sliding windows

- $(\underline{10}\,\underline{01}\,\underline{01}\,\underline{00}\,\underline{11}\,\underline{01}\,\underline{11}\,\underline{01}\,\underline{01}\,\underline{10})_2 =$
  $(02\,01\,01\,00\,03\,01\,03\,01\,01\,02)_2 = (2110313112)_4$,
  needs 8 additions and precomputed $[2]P$ and $[3]P$
- $(10010100\underline{11}01\,\underline{11}010\underline{11}0)_2 =$
  $(1001010003\,0103010030)_2$,
  needs 7 additions and only precomputed $[3]P$
- $(1001010011011101\underline{011}0)_2 =$
  $(10010100\underline{111}0000\bar{3}0030)_2 =$
  $(10010\underline{101}00\bar{1}0000\bar{3}0030)_2 =$
  $(10\underline{011}00\bar{3}00\bar{1}0000\bar{3}0030)_2 =$
  $(1000300\bar{3}00\bar{1}0000\bar{3}0030)_2$
  needs 5 additions and precomputed $[3]P$, assuming that $\ominus$ is available.

Montgomery Ladder

Idea: Compute fixed difference additions.
Consider the intermediate results (i is decreasing).
$Q_i = \sum_{j=i}^{l} [n_j 2^{j-i}]P$, put $R_i = Q_i \oplus P$, then

$$Q_i = [2]Q_{i+1} \oplus [n_i]P = Q_{i+1} \oplus R_{i+1} \oplus n_i P \ominus P = [2]R_{i+1} \oplus n_i P \ominus [2]P.$$

This implies

$$(Q_i, R_i) = \begin{cases} ([2]Q_{i+1}, Q_{i+1} \oplus R_{i+1}) & \text{if } n_i = 0 \\ (Q_{i+1} \oplus R_{i+1}, [2]R_{i+1}) & \text{if } n_i = 1 \end{cases}$$

$$13 = (1101)_2 \sim \begin{array}{rcl} (Q_3, R_3) & = & (P, [2]P) \\ (Q_2, R_2) & = & ([3]P, [4]P) \\ (Q_1, R_1) & = & ([6]P, [7]P) \\ (Q_0, R_0) & = & ([13]P, [14]P) \end{array}$$

Multi-scalar multiplication

Idea of joint doublings

- To compute $[n_1]P_1 \oplus [n_2]P_2 \oplus \cdots \oplus [n_m]P_m$ compute the doublings together, i.e. write scalars n_i in binary:

$$\begin{array}{ccccccccc} n_1 & = & n_{1,l-1}2^{l-1} & +n_{1,l-2}2^{l-2} & +n_{1,l-3}2^{l-3} & \ldots & +n_{1,1}2 & +n_{1,0} \\ n_2 & = & n_{2,l-1}2^{l-1} & +n_{2,l-2}2^{l-2} & +n_{2,l-3}2^{l-3} & \ldots & +n_{2,1}2 & +n_{2,0} \\ \vdots & = & \vdots & \vdots & \vdots & & \vdots & \vdots \\ n_m & = & n_{m,l-1}2^{l-1} & +n_{m,l-2}2^{l-2} & +n_{m,l-3}2^{l-3} & \ldots & +n_{m,1}2 & +n_{m,0} \end{array}$$

Idea of joint doublings

- To compute $[n_1]P_1 \oplus [n_2]P_2 \oplus \cdots \oplus [n_m]P_m$ compute the doublings together, i.e. write scalars n_i in binary:

$$\begin{array}{ccccccccc} n_1 & = & n_{1,l-1}2^{l-1} & +n_{1,l-2}2^{l-2} & +n_{1,l-3}2^{l-3} & \ldots & +n_{1,1}2 & +n_{1,0} \\ n_2 & = & n_{2,l-1}2^{l-1} & +n_{2,l-2}2^{l-2} & +n_{2,l-3}2^{l-3} & \ldots & +n_{2,1}2 & +n_{2,0} \\ \vdots & = & \vdots & \vdots & \vdots & & \vdots & \vdots \\ n_m & = & n_{m,l-1}2^{l-1} & +n_{m,l-2}2^{l-2} & +n_{m,l-3}2^{l-3} & \ldots & +n_{m,1}2 & +n_{m,0} \end{array}$$

- Compute as
$$[2](\underbrace{[n_{1,l-1}]P_1 \oplus [n_{2,l-1}]P_2 \oplus [n_{3,l-1}]P_3 \oplus \cdots \oplus [n_{m,l-1}]P_m}_{\text{first column}})$$

Idea of joint doublings

- To compute $[n_1]P_1 \oplus [n_2]P_2 \oplus \cdots \oplus [n_m]P_m$ compute the doublings together, i.e. write scalars n_i in binary:

$$\begin{array}{cccccccc} n_1 & = & n_{1,l-1}2^{l-1} & +n_{1,l-2}2^{l-2} & +n_{1,l-3}2^{l-3} & \dots & +n_{1,1}2 & +n_{1,0} \\ n_2 & = & n_{2,l-1}2^{l-1} & +n_{2,l-2}2^{l-2} & +n_{2,l-3}2^{l-3} & \dots & +n_{2,1}2 & +n_{2,0} \\ \vdots & = & \vdots & \vdots & \vdots & & \vdots & \vdots \\ n_m & = & n_{m,l-1}2^{l-1} & +n_{m,l-2}2^{l-2} & +n_{m,l-3}2^{l-3} & \dots & +n_{m,1}2 & +n_{m,0} \end{array}$$

- Compute as

$[2]\left([2]([n_{1,l-1}]P_1 \oplus [n_{2,l-1}]P_2 \oplus [n_{3,l-1}]P_3 \oplus \cdots \oplus [n_{m,l-1}]P_m)\oplus\right.$
$\left.([n_{1,l-2}]P_1 \oplus [n_{2,l-2}]P_2 \oplus [n_{3,l-2}]P_3 \oplus \cdots \oplus [n_{m,l-2}]P_m\right) \oplus$
$\cdots$ etc.

- Needs many more additions than doublings, even with precomputations.

Applications

- ECDSA verification uses 2 scalar multiplications ... just to add the results.
- If base point P is fixed, precompute $R = [2^{l/2}]P$ and include in the curve parameters. Split scalar $n = n_1 2^{l/2} + n_0$ and compute

$$[n_1]R \oplus [n_0]P.$$

- GLV curves split scalar in two halves to get faster scalar multiplication.
- Verification in accelerated ECDSA can be extended to use 4 or even 6 scalars. Splitting of the scalar is done by LLL techniques.
- Further applications in batch verification of signatures – many scalars – by taking random linear combinations.

Examples

$[1338]P \oplus [1715]Q = [(10100111010)_2]P \oplus [(11010110011)_2]Q$
takes 20 doublings and 12 additions.

$[(10100111010)_2]P \oplus$
$[(11010110011)_2]Q$
Given precomputed $P \oplus Q$, we can compute the same in 10 doublings and 8 additions.

If $\ominus$ is efficient $[(01010100\bar{1}010)_2]P \oplus [(100\bar{1}0\bar{1}0\bar{1}010\bar{1})_2]Q$ reduces (compared to first line) the number of additions to 10, but needs 21 doublings.

Given precomputed $P \oplus Q$ and $P \ominus Q$
$[(01010100\bar{1}010)_2]P \oplus$
$[(100\bar{1}0\bar{1}0\bar{1}010\bar{1})_2]Q$
needs 11 doublings and 8 additions. So the joint Hamming weight has not decreased & length has increased.

Joint-Sparse-Form

Solinas introduced generalization of NAF to two scalars.

1. of any three consecutive columns, at least one is a zero column, that is for all i and all positive $j > 1$ one has $n_{i,j+k} = n_{1-i,j+k} = 0$ for at least one k in $\{0, \pm 1\}$
2. adjacent terms do not have opposite signs, i. e. it is never the case that $n_{i,j} n_{i,j+1} = -1$
3. if $n_{i,j+1} n_{i,j} \neq 0$ then one has $n_{1-i,j+1} = \pm 1$ and $n_{1-i,j} = 0$.

$1338 = (10100111010)_2 = (0101010\bar{0}\bar{1}010)_2$
$1715 = (11010110011)_2 = (100\bar{1}0\bar{1}0\bar{1}0011)_2$
So $[1338]P \oplus [1715]Q$ needs 7 additions and 11 doublings given precomputed $P \oplus Q$ and $P \ominus Q$.
Average joint density is 1/2, i.e. for two integers of length l (each) we need on average l doublings and $l/2$ additions

Joint sparse form recoding

`IN`: Nonnegative l-bit integers n_0 and n_1 not both zero.
`OUT`: The joint sparse form of n_0 and n_1.

1. $j \leftarrow 0$, $d_0 \leftarrow 0$ and $d_1 \leftarrow 0$
2. `while` $n_0 + d_0 > 0$ `OR` $n_1 + d_1 > 0$ `do`
   (a) $l_0 \leftarrow d_0 + n_0$ and $l_1 \leftarrow d_1 + n_1$
   (b) `for` $i = 0$ `to` 1 `do`
      i. `if` l_i is even `then` $n_{i,j} \leftarrow 0$
      ii. `else` $n_{i,j} \leftarrow l_i \bmod s\ 4$
      iii. `if` $l_i \equiv \pm 3 \pmod 8$ `AND` $l_{1-i} \equiv 2 \pmod 4$
      `then` $n_{i,j} \leftarrow -n_{i,j}$
   (c) `for` $i = 0$ `to` 1 `do`
      i. `if` $2d_i = 1 + n_{i,j}$ `then` $d_i \leftarrow 1 - d_i$
      ii. $n_i \leftarrow \lfloor n_i/2 \rfloor$
   (d) $j \leftarrow j + 1$
3. `return` $(n_{0,l} \ldots n_{0,0})_s$ and $(n_{1,l} \ldots n_{1,0})_s$

. . . well, somewhat related though . . .

Weierstrass curves

$$E : y^2 + \underbrace{(a_1 x + a_3)}_{h(x)} y = \underbrace{x^3 + a_2 x^2 + a_4 x + a_6}_{f(x)}, \;\; h, f \in \mathbb{F}_q[x].$$

Group: $E(\mathbb{F}_q) = \{ (x, y) \in \mathbb{F}_q^2 \, : \, y^2 + h(x)y = f(x) \} \cup \{ P_\infty \}$

Often $q = 2^r$ or $q = p$, prime. Isomorphic transformations lead to

$$y^2 = f(x) \qquad q \text{ odd},$$

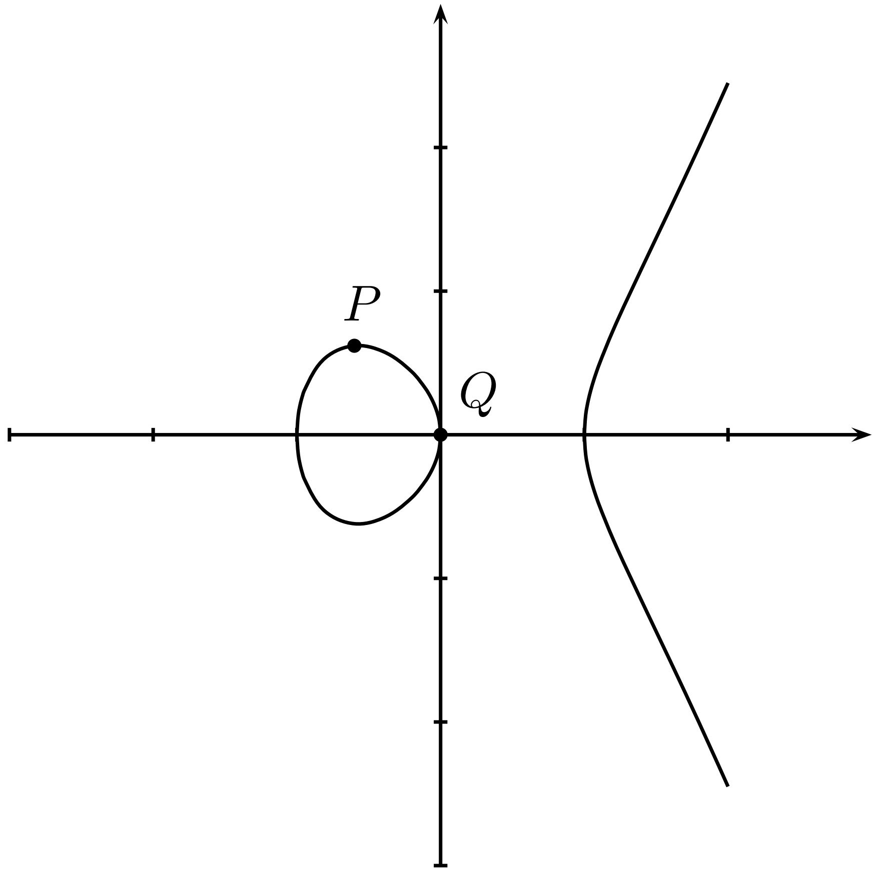
for

$$\begin{aligned} y^2 + xy &= x^3 + a_2 x^2 + a_6 \\ y^2 + y &= x^3 + a_4 x + a_6 \end{aligned} \qquad q = 2^r, \quad \begin{aligned} &\text{curve non-supersingular} \\ &\text{curve supersingular} \end{aligned}$$

In this talk we consider only fields of odd characteristic and $\mathbb{R}$.

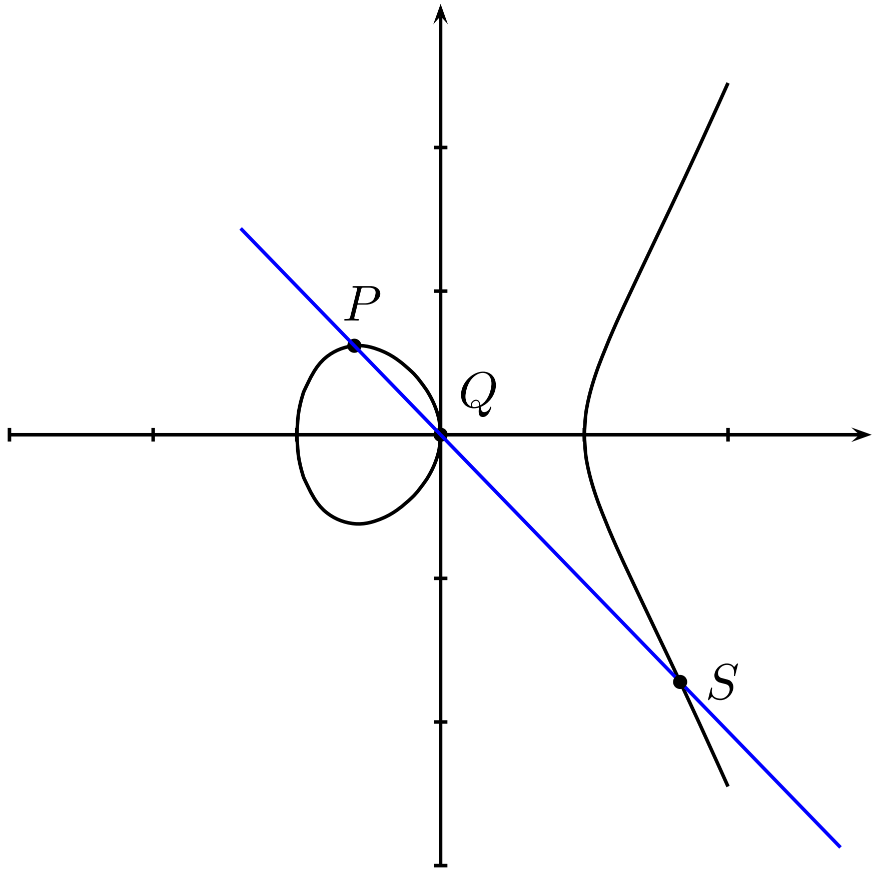
Group Law in $E(\mathbb{R}), h = 0$

$y^2 = x^3 - x$



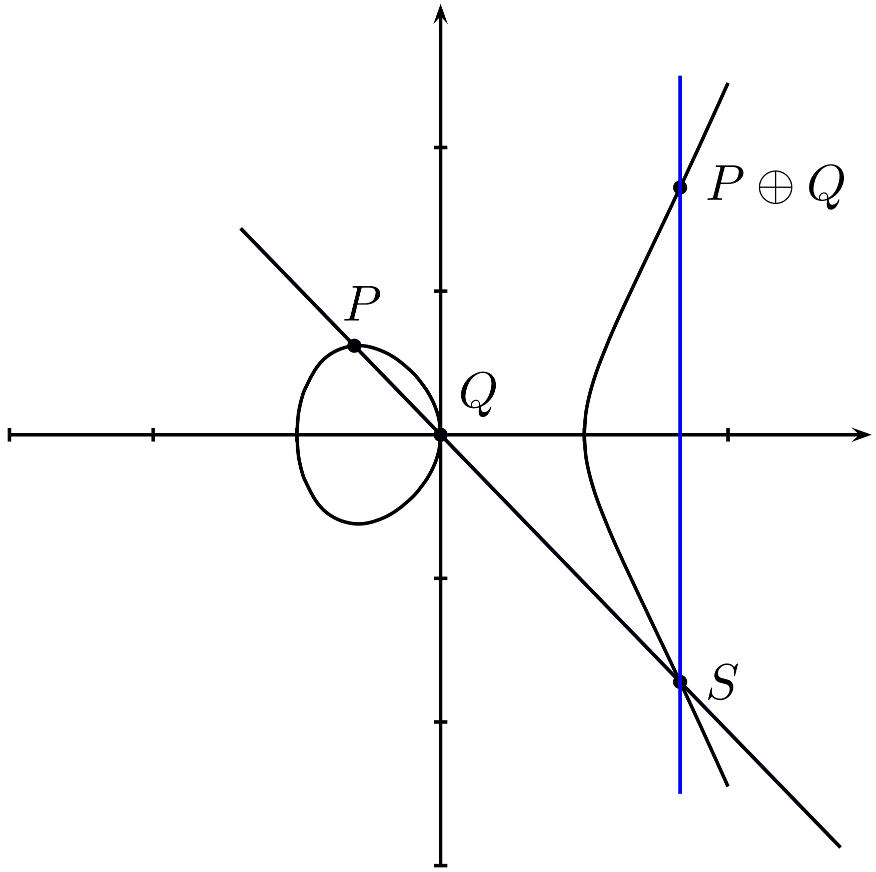
Group Law in $E(\mathbb{R}), h = 0$

$y^2 = x^3 - x$

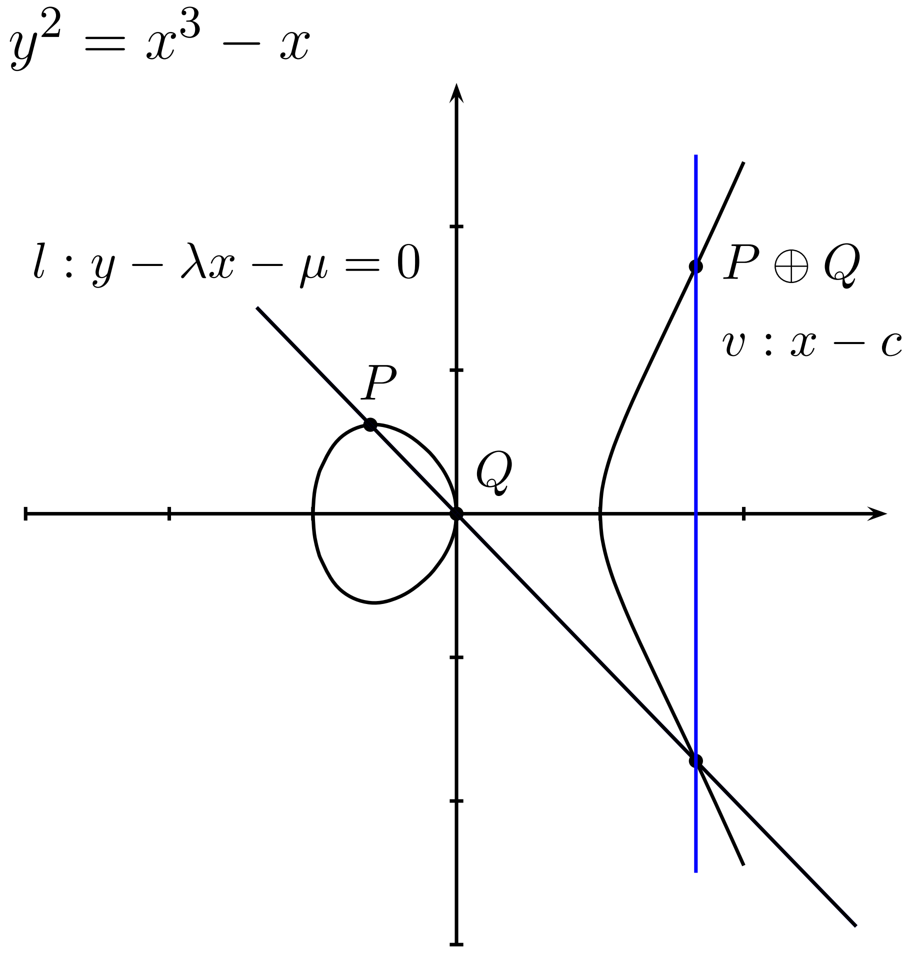


Group Law in $E(\mathbb{R}), h = 0$

$y^2 = x^3 - x$

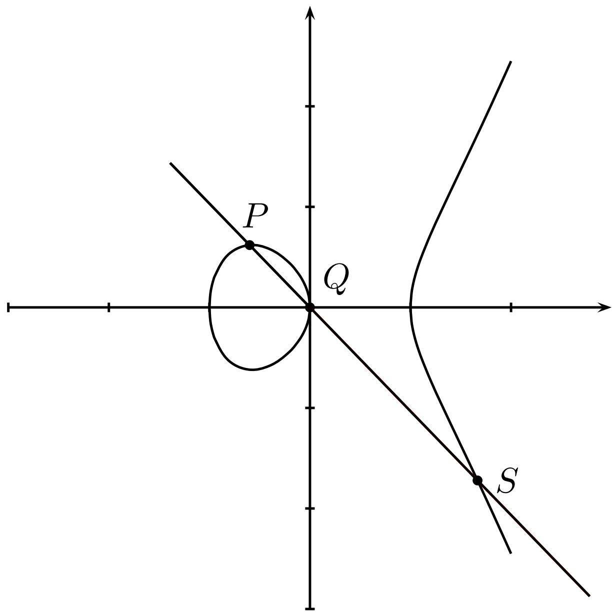


Notation (for Ben and for next slide)



Group Law (q odd)

$$E : y^2 = x^3 + a_4x + a_6,\ a_i \in \mathbb{F}_q$$



Line $y = \lambda x + \mu$ has slope

$$\lambda = \frac{y_Q - y_P}{x_Q - x_P}.$$

Equating gives

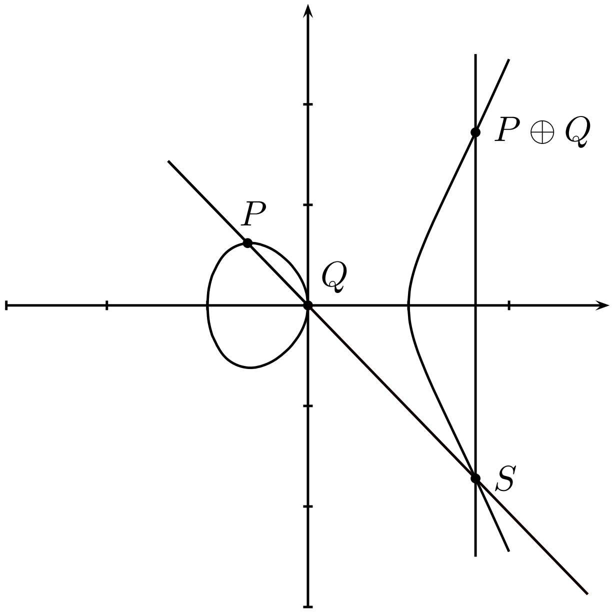
$$(\lambda x + \mu)^2 = x^3 + a_4x + a_6.$$

This equation has 3 solutions, the x-coordinates of P, Q and S, thus

$$\begin{aligned}(x - x_P)(x - x_Q)(x - x_S) &= x^3 - \lambda^2x^2 + (a_4 - 2\lambda\mu)x + a_6 - \mu^2 \\ x_S &= \lambda^2 - x_P - x_Q\end{aligned}$$

Group Law (q odd)

$$E : y^2 = x^3 + a_4x + a_6,\ a_i \in \mathbb{F}_q$$



Point P is on line, thus

$y_P = \lambda x_P + \mu$, i.e.
$\mu = y_P - \lambda x_P$,

and

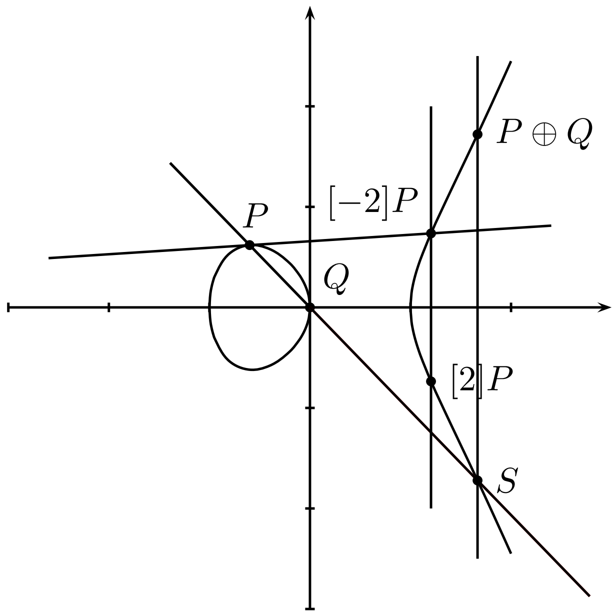
$$\begin{aligned} y_S &= \lambda x_S + \mu \\ &= \lambda x_S + y_P - \lambda x_P \\ &= \lambda(x_S - x_P) + y_P \end{aligned}$$

Point $P \oplus Q$ has the same x-coordinate as S but negative y-coordinate:

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

Group Law (q odd)

$$E : y^2 = x^3 + a_4 x + a_6, \; a_i \in \mathbb{F}_q$$



When doubling, use tangent at P. Compute slope λ via partial derivatives of curve equation:

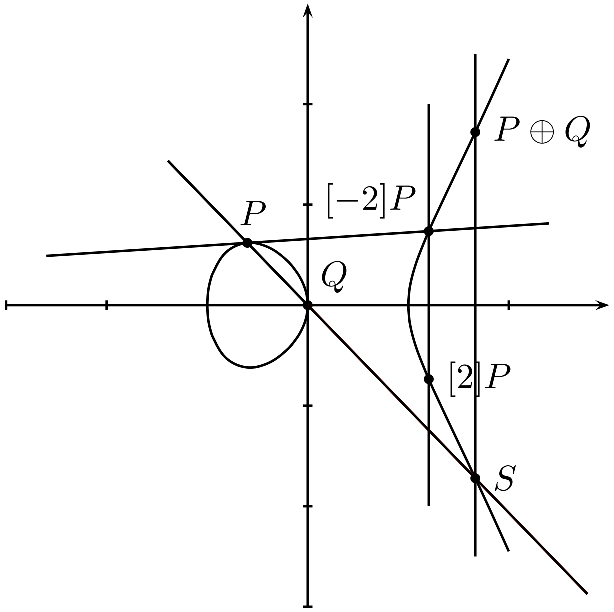
$$\lambda = \frac{3x_P^2 + a_4}{2y_P}.$$

Remaining computation identical to addition.

$$x_{[2]P} = \lambda^2 - 2x_P, \quad y_{[2]P} = \lambda(x_P - x_{[2]P}) - y_P$$

Group Law (q odd)

$$E : y^2 = x^3 + a_4 x + a_6,\ a_i \in \mathbb{F}_q$$



In general, for $(x_P, y_P) \neq (x_Q, -y_Q)$:

$$\begin{aligned}(x_P, y_P) \oplus (x_Q, y_Q) &= \\ &= (x_{P\oplus Q}, y_{P\oplus Q}) = \\ &= (\lambda^2 - x_P - x_Q, \lambda(x_P - x_{P\oplus Q}) - y_P),\end{aligned}$$

where

$$\lambda = \begin{cases} (y_Q - y_P)/(x_Q - x_P) & \text{if } x_P \neq x_Q, \\ (3x_P^2 + a_4)/(2y_P) & \text{if } P = Q \end{cases}$$

$\Rightarrow$ Addition and Doubling need

1 I, 2M, 1S and 1 I, 2M, 2S, respectively.

Note that $-(x, y) = (x, -y)$.

Long Weierstrass equation

$$E : y^2 + \underbrace{(a_1x + a_3)}_{h(x)}\, y = \underbrace{x^3 + a_2x^2 + a_4x + a_6}_{f(x)},\ \ h, f \in \mathbb{F}_q[x].$$

- Negative of $P = (x_P, y_P)$ is given by $-P = (x_P, -y_P - h(x_P))$.
- $(x_P, y_P) \oplus (x_Q, y_Q) = (x_R, y_R) =$ $= (\lambda^2 + a_1\lambda - a_2 - x_P - x_Q, \lambda(x_P - x_R) - y_P - a_1x_R - a_3)$, where

$$\lambda = \begin{cases} (y_Q - y_P)/(x_Q - x_P) & \text{if } x_P \neq x_Q, \\ \frac{3x_P^2 + 2a_2x_P + a_4 - a_1y_P}{2y_P + a_1x_P + a_3} & \text{if } P = Q \text{ and } P \neq -Q \end{cases}$$

- $P \oplus (-P) = P_\infty$.
- $P \oplus P_\infty = P_\infty \oplus P = P$.

Relationship between Weierstrass and Edwards

- Every elliptic curve with point of order 4 is birationally equivalent to an Edwards curve.
- Let $P_4 = (u_4, v_4)$ have order 4 and shift u s.t. $2P_4 = (0,0)$. Then Weierstrass form:
$$v^2 = u^3 + (v_4^2/u_4^2 - 2u_4)u^2 + u_4^2 u.$$
- Define $d = 1 - (4u_4^3/v_4^2)$.
- The coordinates $x = v_4 u/(u_4 v),\ y = (u - u_4)/(u + u_4)$ satisfy
$$x^2 + y^2 = 1 + dx^2y^2.$$
- Inverse map $u = u_4(1 + y)/(1 - y),\ v = v_4 u/(u_4 x)$.
- Finitely many exceptional points. Exceptional points have $v(u + u_4) = 0$.

Exceptional points of the map

- Points with $v(u+u_4)=0$ on Weierstrass curve map to points at infinity on desingularization of Edwards curve.
- Reminder: $d=1-(4u_4^3/v_4^2)$.
- $u=-u_4$ is u-coordinate of a point iff

$$\begin{aligned} & (-u_4)^3+(v_4^2/u_4^2-2u_4)(u_4)^2+u_4^2(u_5) \\ = \quad & v_4^2-4u_4^3=v_4^2 d \end{aligned}$$

  is a square, i. e., iff d is a square.
- $v=0$ corresponds to $(0,0)$ which maps to $(0,-1)$ on Edwards curve and to solutions of $u^2+(v_4^2/u_4^2-2u_4)u+u_4^2=0$. Discriminant is

$$(v_4^2/u_4^2-2u_4)^2-4u_4^2=v_4^4 d,$$

  i. e., points defined over k iff d is a square.

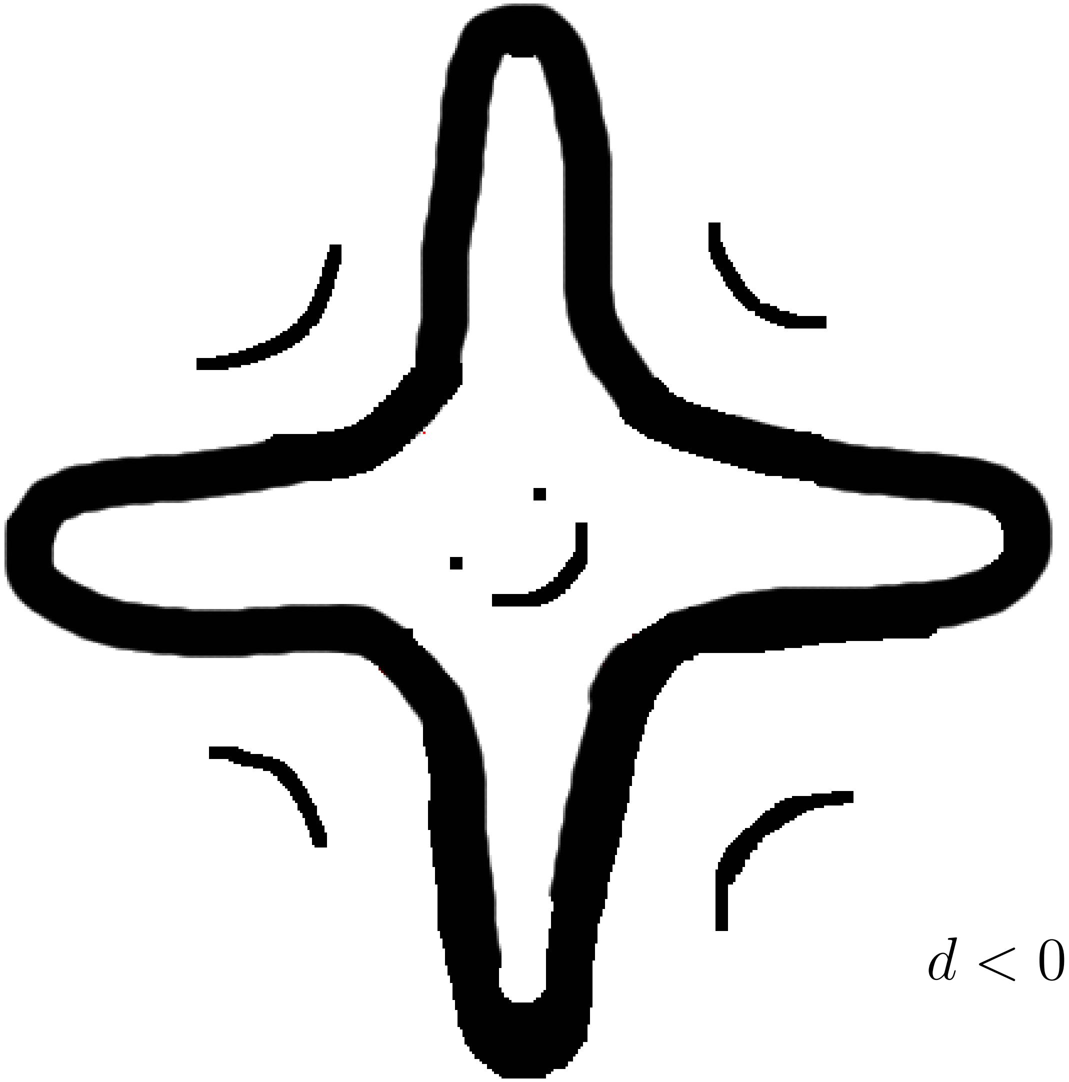
Complete addition law

- Previous slide shows that for $d \neq \square$ in $\mathbb{F}_q$ all points of the Weierstrass curve map to the affine part of the Edwards curve; where we extend the map by $P_\infty \mapsto (0,1)$ and $(0,0) \mapsto (0,-1)$.
- Geometric description: The points $(1:0:0)$ and $(0:1:0)$ at infinity on the Edwards curve are singular. They blow up to two points each on the desingularization of the curve; these points are defined over $\mathbb{F}_q(\sqrt{d})$.
- Attention: Having no $\mathbb{F}_q$-rational points at infinity does not guarantee that the formulas are complete:

$$(x_3, y_3) = ((x_1y_1 + x_2y_2)/(x_1x_2 + y_1y_2), (x_1y_1 - x_2y_2)/(x_1y_2 - y_1x_2))$$

is addition on Edwards curve . . . and fails for doublings.

Thank you for your attention!



$d < 0$, non-square!