Curves with endomorphisms and DLPs in intervals

Tanja Lange
Technische Universiteit Eindhoven

with some slides by
Daniel J. Bernstein

More elliptic curves

Can use any field k.

Can use any nonsingular curve $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$.

"Nonsingular": no $(x, y) \in \bar{k} \times \bar{k}$ simultaneously satisfies $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ and $2y + a_1x + a_3 = 0$ and $a_1y = 3x^2 + 2a_2x + a_4$.

Easy to check nonsingularity.

Almost all curves are nonsingular when k is large.

An example over $\mathbf{R}$

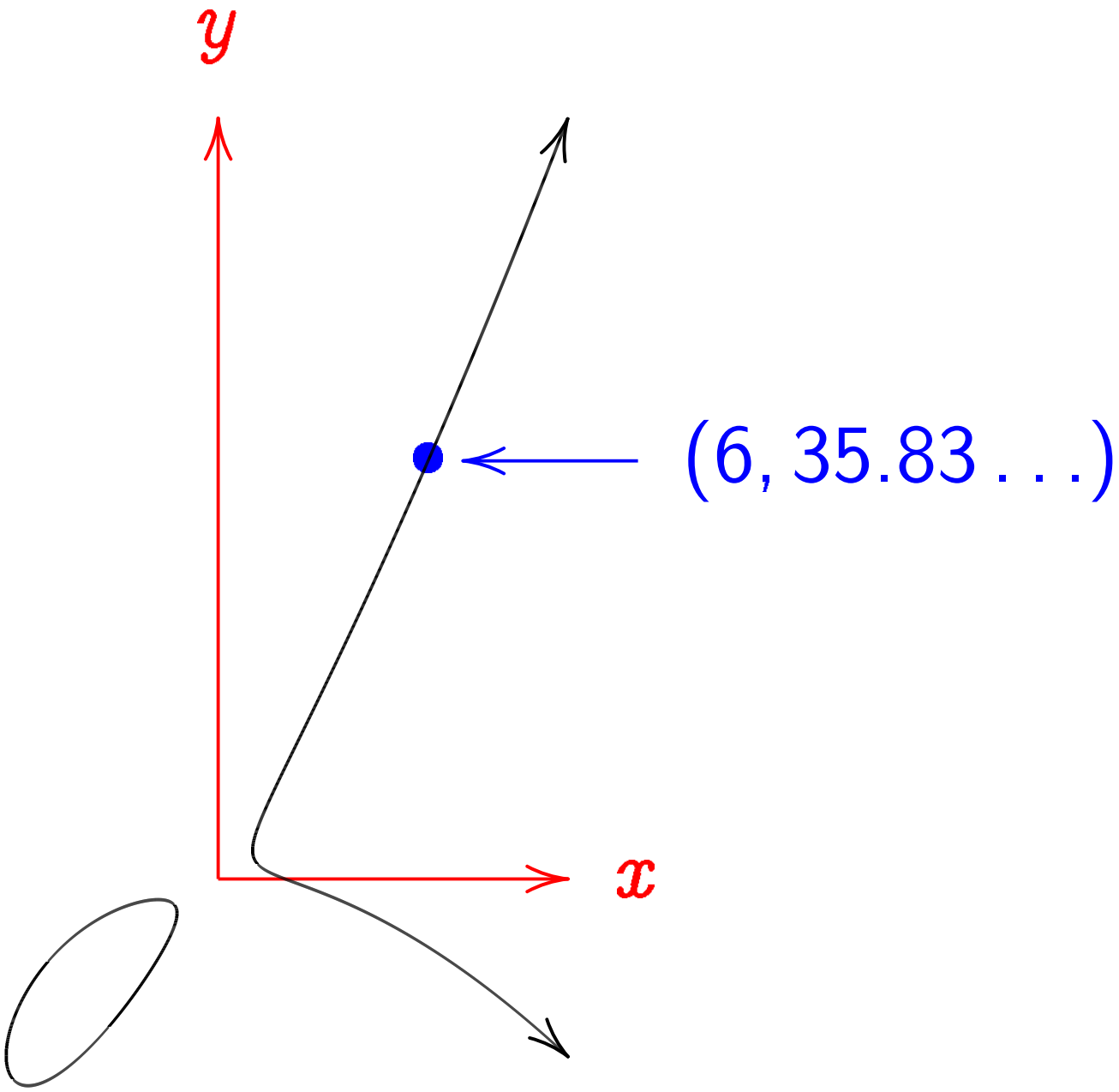
Consider all pairs
of real numbers x, y
such that $y^2 - 5xy = x^3 - 7$.

The "points on the elliptic curve
$y^2 - 5xy = x^3 - 7$ over $\mathbf{R}$"
are those pairs and
one additional point, ∞.

i.e. The set of points is
$\{(x, y) \in \mathbf{R} \times \mathbf{R} :$
$y^2 - 5xy = x^3 - 7\} \cup \{\infty\}$.

($\mathbf{R}$ is the set of real numbers.)

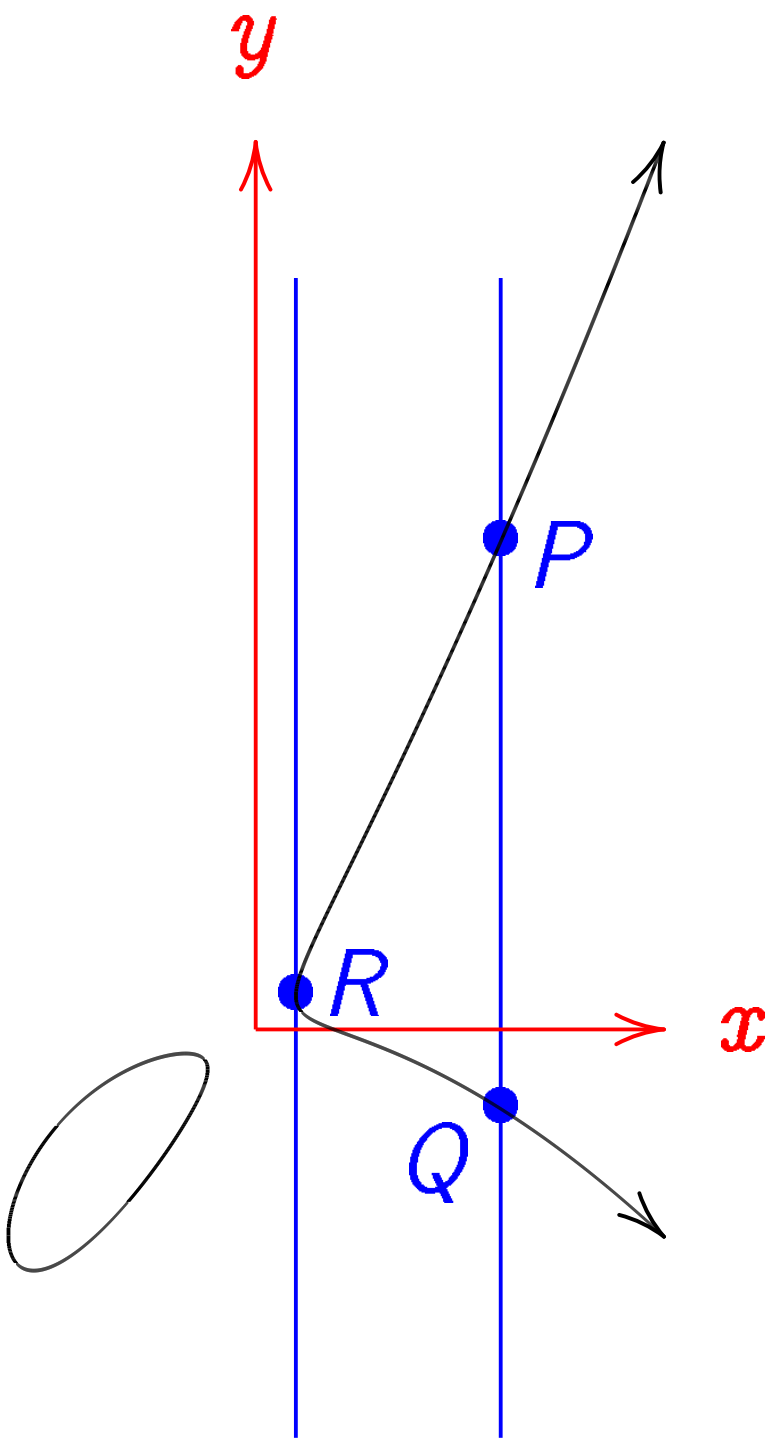
Graph of this set of points:



Don't forget ∞.

Visualize ∞ as top of y axis.

Here $-P = Q$, $-Q = P$, $-R = R$:

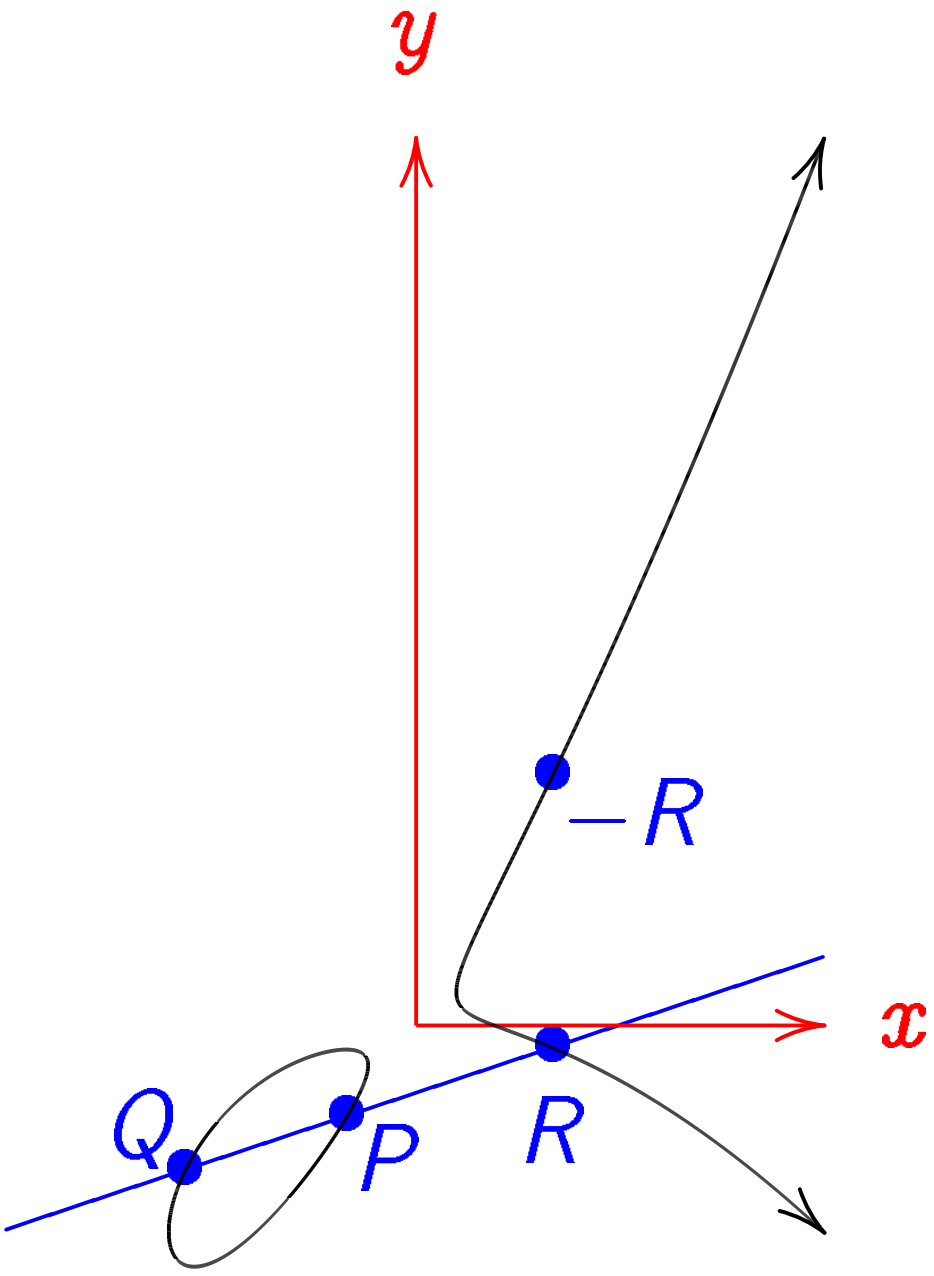


Distinct curve points P, Q, R
on a line
have $P + Q = -R$;
$P + Q + R = \infty$.

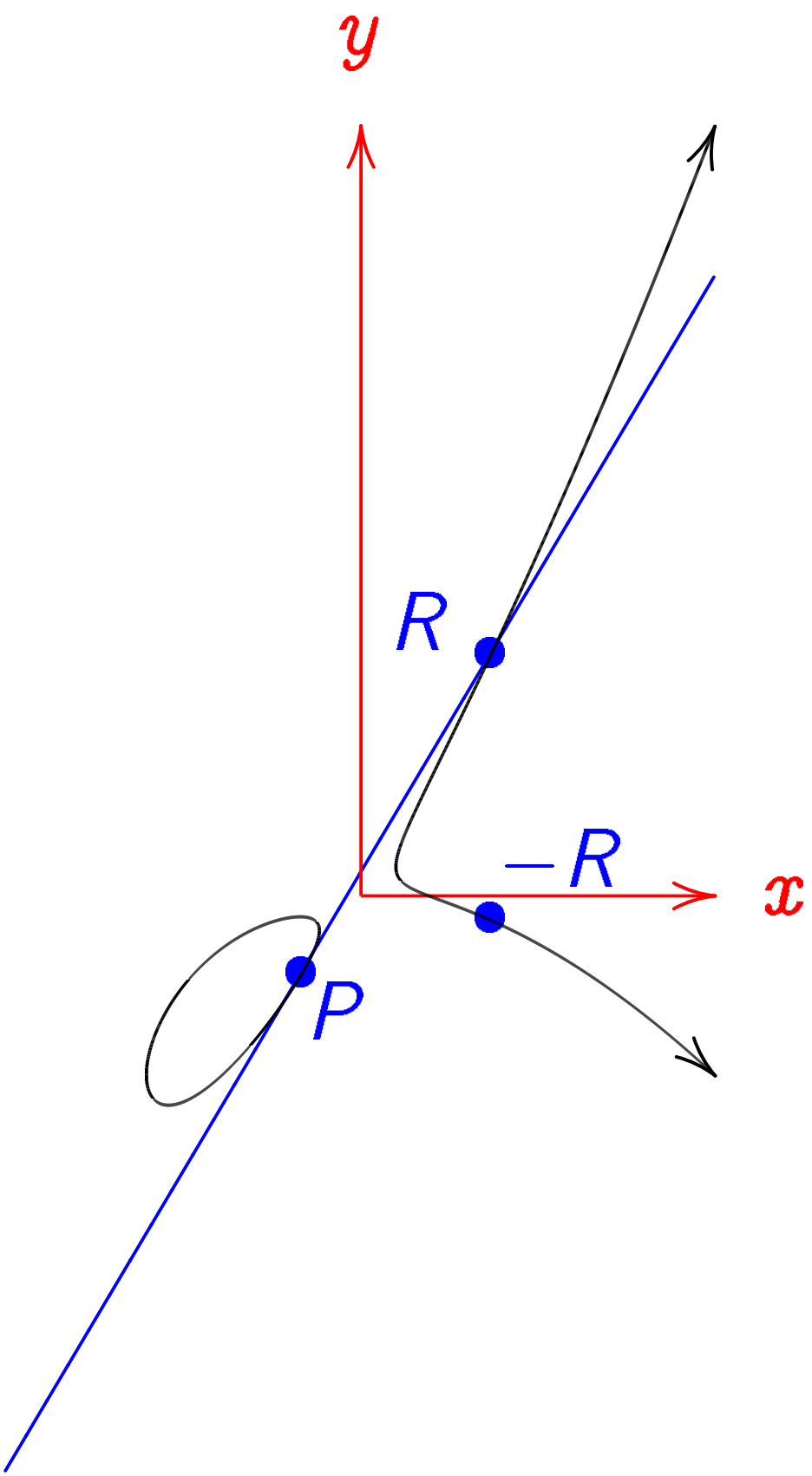
Distinct curve points P, R
on a line tangent at P
have $P + P = -R$;
$P + P + R = \infty$.

A non-vertical line
with only one curve point P
(a flex of the curve)
has $P + P = -P$;
$P + P + P = \infty$.

Here $P + Q = -R$:



Here $P + P = -R$:



Curve addition formulas

Easily find formulas for $+$
by finding formulas for lines
and for curve-line intersections.

$x \neq x'$: $(x, y) + (x', y') = (x'', y'')$
where $\lambda = (y' - y)/(x' - x)$,
$x'' = \lambda^2 - 5\lambda - x - x'$,
$y'' = 5x'' - (y + \lambda(x'' - x))$.

$2y \neq 5x$: $(x, y) + (x, y) = (x'', y'')$
where $\lambda = (5y + 3x^2)/(2y - 5x)$,
$x'' = \lambda^2 - 5\lambda - 2x$,
$y'' = 5x'' - (y + \lambda(x'' - x))$.

$(x, y) + (x, 5x - y) = \infty$.

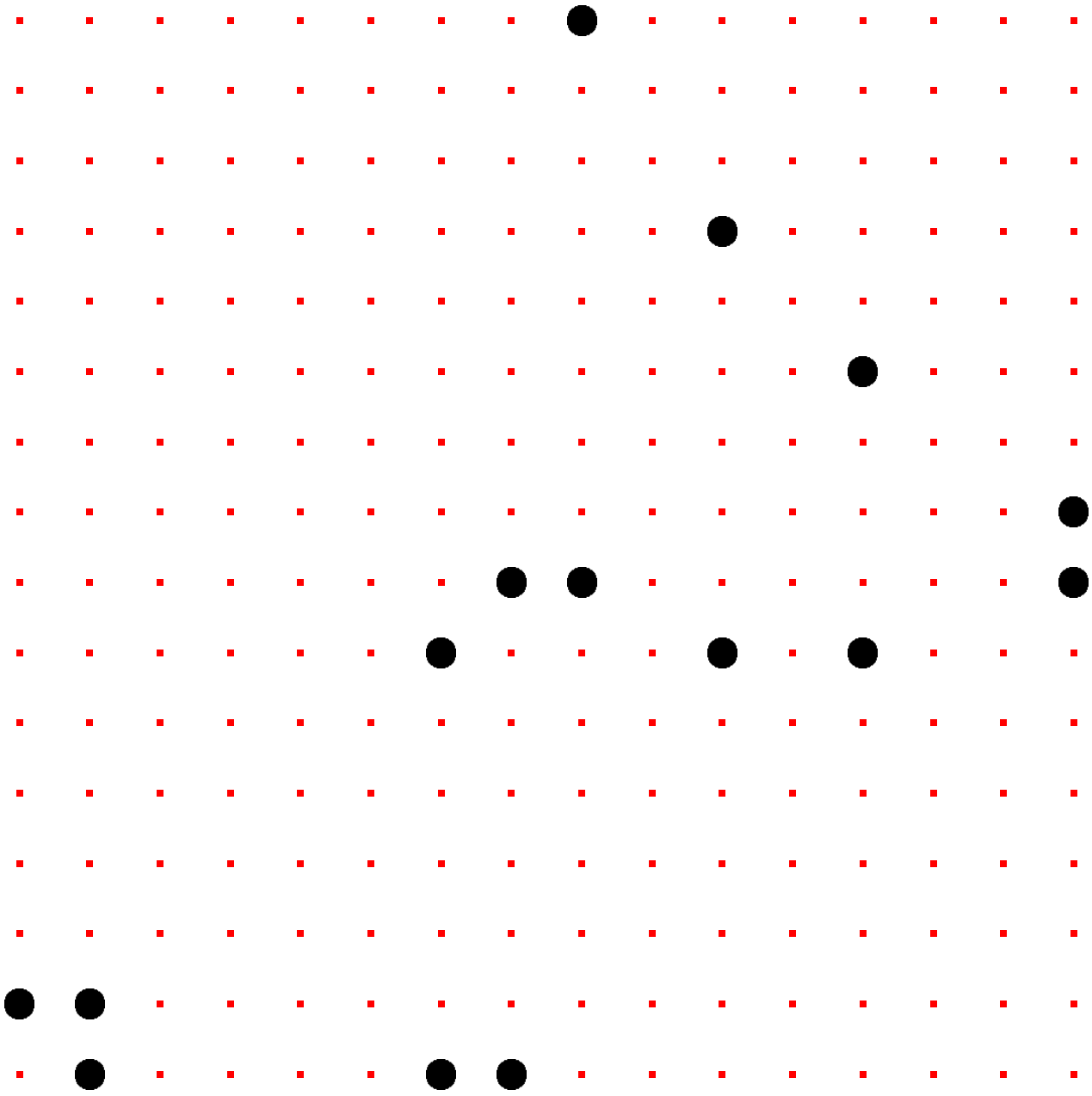
An elliptic curve over $\mathbf{F}_{16}$

Consider the non-prime field

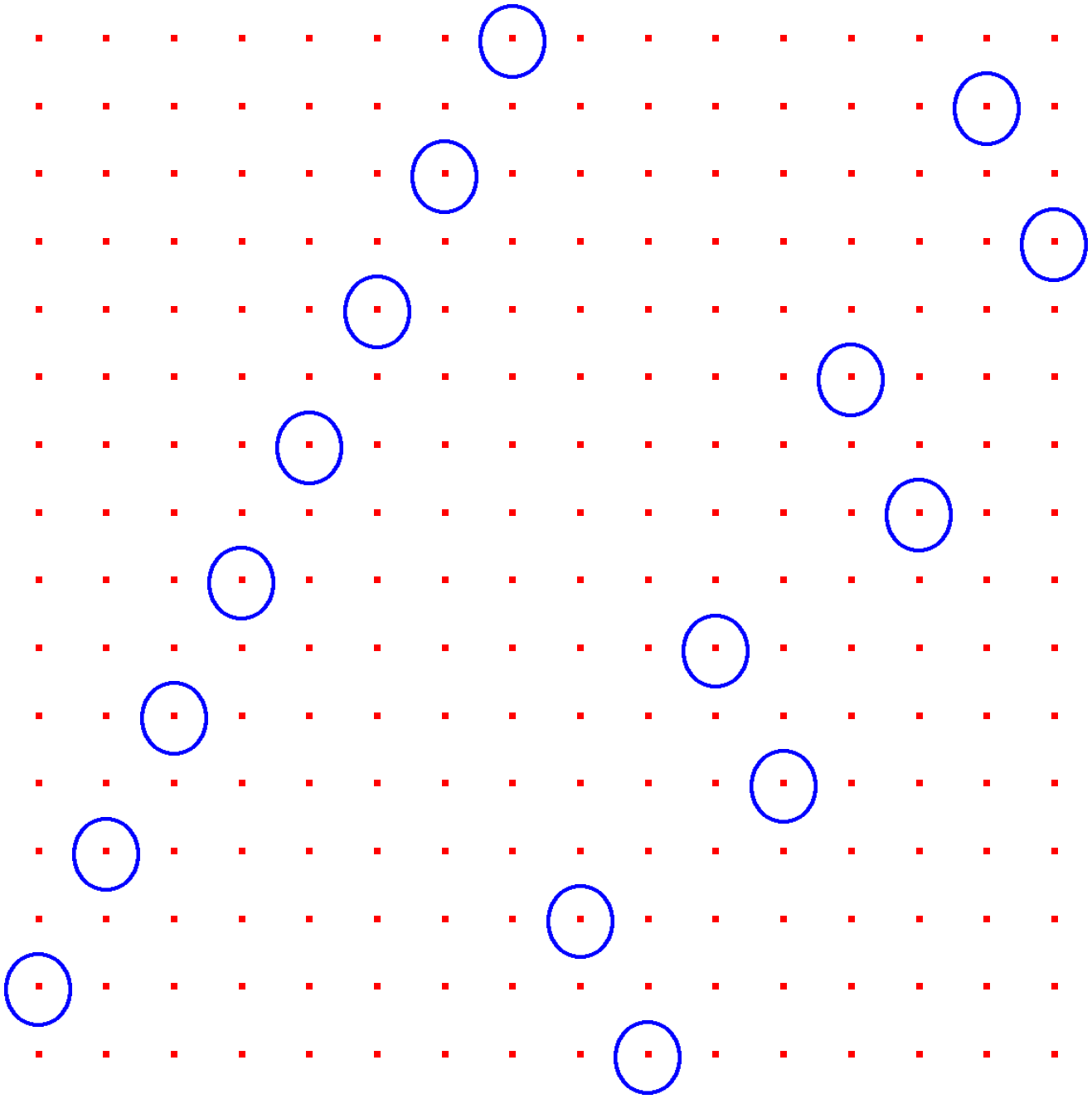
$$(\mathbf{Z}/2)[t]/(t^4 - t - 1) = \{$$
$$0t^3 + 0t^2 + 0t^1 + 0t^0,$$
$$0t^3 + 0t^2 + 0t^1 + 1t^0,$$
$$0t^3 + 0t^2 + 1t^1 + 0t^0,$$
$$0t^3 + 0t^2 + 1t^1 + 1t^0,$$
$$0t^3 + 1t^2 + 0t^1 + 0t^0,$$
$$\vdots$$
$$1t^3 + 1t^2 + 1t^1 + 1t^0\}$$

of size $2^4 = 16$.

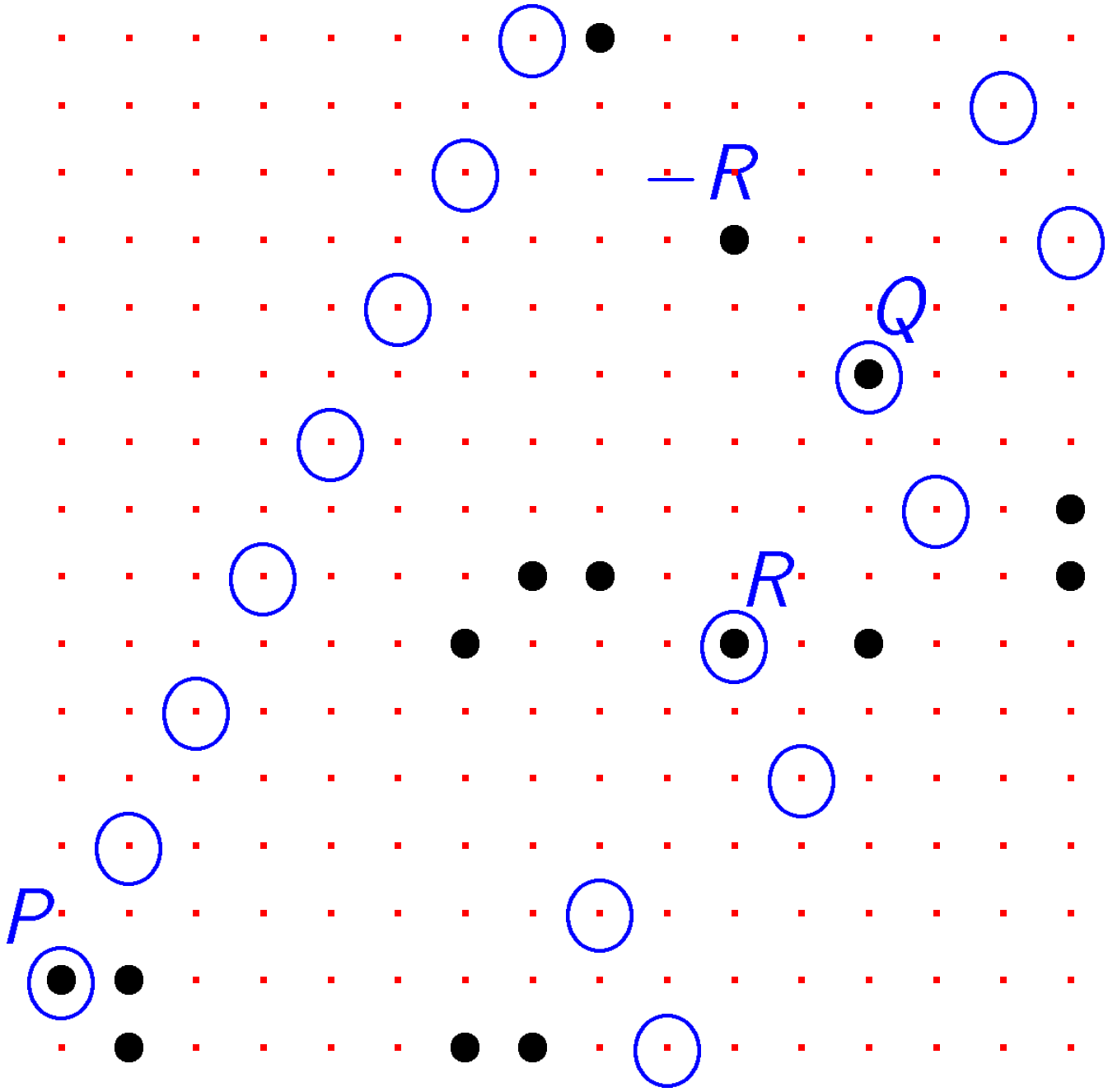
Graph of the "set of points on the elliptic curve $y^2 - 5xy = x^3 - 7$ over $(\mathbf{Z}/2)[t]/(t^4 - t - 1)$":



Line $y = tx + 1$:



$P + Q = -R$:



Why more coefficients?

$y^2 + a_1xy + a_3y =$
$x^3 + a_2x^2 + a_4x + a_6$.

"Nonsingular": no $(x, y) \in \bar{k} \times \bar{k}$ simultaneously satisfies $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ and $2y + a_1x + a_3 = 0$ and $a_1y = 3x^2 + 2a_2x + a_4$.

Easy to check nonsingularity.

Almost all curves are nonsingular when k is large.

Why more coefficients?

$y^2 + a_1xy + a_3y =$
$x^3 + a_2x^2 + a_4x + a_6.$

"Nonsingular": no $(x, y) \in k \times k$ simultaneously satisfies $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ and $2y + a_1x + a_3 = 0$ and $a_1y = 3x^2 + 2a_2x + a_4$.

$k = \mathbf{F}_{2^n}$, then partial derivatives become: $a_1x + a_3 = 0$ and $a_1y = x^2 + a_4$. Monday's curve shape had $a_1 = a_3 = 0$
$\Rightarrow$ only condition $x^2 = a_4$ and every element is a square in $\mathbf{F}_{2^n}$.

Isomorphic transformations

Elliptic curves over $\mathbf{F}_{2^n}$ need to have at least one of a_1 and a_3 non-zero.

Do *isomorphic transformations* linear transformations $y \mapsto a^3y + bx + c$, $x \mapsto a^2x + d$ to simplify curve equation.

If $a_1 \neq 0$ use a and d to map to $y^2 + xy = x^3 + a_2'x^2 + a_4'x + a_6'$ and c to achieve $a_4' = 0$.

b appears as $b^2 + b + a_2'$, can restrict coefficient of x^2 to two choices.

If $a_1 = 0$, put $b = 0$, $d = a_2$ to map to
$y^2 + a_3 y = x^3 + a_4' x + a_6'$
c appears as $c^2 + a_3 c + a_6'$, can restrict constant term; can use a to restrict choice of a_3; if n odd can get $a_3 = 1$.

If $\mathrm{char}(k) \neq 2$ put $b = -a_1/2$ and $c = -a_3/2$ to map to
$y^2 = x^3 + a_2' x^2 + a_4' x + a_6'$.
If $\mathrm{char}(k) \neq 3$ can additionally remove a_2' using d. Can use a to restrict a_4' or a_6'.

Short Weierstrass forms

Over $\mathbf{F}_{2^n}$ can map to one of
$y^2 + xy = x^3 + a_2x^2 + a_6$
$y^2 + a_3y = x^3 + a_4x + a_6$
with $a_2, a_4, a_6 \in \mathbf{F}_{2^n}$;
$a_3 = 1$ for n odd.

Over $\mathbf{F}_q$, $q = p^n$, $p > 3$ can map
to $y^2 = x^3 + a_4x + a_6$
with $a_4, a_6 \in \mathbf{F}_q$.

Nice for proofs but arithmetic
might prefer other choices,
e.g. Montgomery curves
$y^2 = x^3 + a_2x^2 + x$ over $\mathbf{F}_q$
are faster than above form.

Quadratic twists

Over $\mathbf{F}_q$, $q = p^n$, $p > 3$
still have freedom to map
$E : y^2 = x^3 + a_4x + a_6$ to
$E' : y^2 = x^3 + a_4/c^4x + a_6/c^6$
using $y \mapsto c^3y$, $x \mapsto c^2x$, $c \in \mathbf{F}_q$.

For $d \in \mathbf{F}_q$, curve
$\tilde{E} : y^2 = x^3 + a_4/d^2x + a_6/d^3$
is defined over $\mathbf{F}_q$ but
isomorphism is defined over $\mathbf{F}_q$
only if d is a square in $\mathbf{F}_q$.
$\tilde{E}$ is a *quadratic twist* of E. This
concept includes isomorphisms.
Only *one* non-isomorphic class.

General addition law

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

$$-(x_P, y_P) = (x_P, -y_P - h(x_P)).$$

$$(x_P, y_P) + (x_R, y_R) = (x_3, y_3) =$$
$$= (\lambda^2 + a_1\lambda - a_2 - x_P - x_R,$$
$$\lambda(x_P - x_3) - y_P - a_1x_3 - a_3),$$

where $\lambda =$
$$\begin{cases} (y_R - y_P)/(x_R - x_P) & x_P \neq x_R, \\ \frac{3x_P^2 + 2a_2x_P + a_4 - a_1y_P}{2y_P + a_1x_P + a_3} & P = R \neq -R \end{cases}$$

Koblitz curves

Let $q = p^n$ for small p and big n.
$y^2 + h(x)y = f(x)$
over $\mathbf{F}_q$ is called a *Koblitz curve*
if it is defined over $\mathbf{F}_p$, i.e., if
$h(x), f(x) \in \mathbf{F}_p[x]$.

p need not be prime; $p = 4$ is also small.

Typical case: $p = 2$. This is the case proposed by Koblitz; also called *anomalous binary curves*.

Take $E : y^2 + h(x)y = f(x)$,
with $h(x), f(x) \in \mathbf{F}_p[x]$ as curve
over $\mathbf{F}_{p^n}$
and let $P = (x_P, y_P) \in E(\mathbf{F}_{p^n})$.

Then $\sigma(P) = (x_P^p, y_P^p)$ is also a
point in $E_a(\mathbf{F}_{p^n})$:

Proof uses that Frobenius
automorphism is linear
$(a + b)^p = a^p + b^p$
and that $c^p = c$ for $c \in \mathbf{F}_p$.

The map σ is called the *Frobenius endomorphism* of E.

Properties of Koblitz curves

Let $\#E(\mathbf{F}_p) = p + 1 - t$ and let $T^2 - tT + p = (T - \tau)(T - \bar{\tau})$ then

$$\#E(\mathbf{F}_{p^n}) = (1 - \tau^n)(1 - \bar{\tau}^n).$$

Easy computation of number of points – but shows restriction: if $m|n$ then $\#E(\mathbf{F}_{p^m})|\#E(\mathbf{F}_{p^n})$, so require *prime* n to have large prime order subgroup.

$\chi(T) = T^2 - tT + p$ called *characteristic polynomial of the Frobenius endomorphism*.

Each $P \in E(\mathbf{F}_{p^n})$ satisfies
$\sigma^2(P) - t\sigma(P) + pP = \infty$.

Each $P \in E(\mathbf{F}_{p^n})$ satisfies
$\sigma^2(P) - t\sigma(P) + pP = \infty$.

This means
$pP = t\sigma(P) - \sigma^2(P)$
for $t \in [-2\sqrt{p}, 2\sqrt{p}]$.

Each $P \in E(\mathbf{F}_{p^n})$ satisfies
$\sigma^2(P) - t\sigma(P) + pP = \infty$.

This means
$pP = t\sigma(P) - \sigma^2(P)$
for $t \in [-2\sqrt{p}, 2\sqrt{p}]$.

Expand integer k in base τ
$k = \sum k_i \tau^i$, with
$k_i \in [-\lfloor (p-1)/2 \rfloor, \lceil (p-1)/2 \rceil]$
and compute
$kP = \sum k_i \sigma^i(P)$.

Each $P \in E(\mathbf{F}_{p^n})$ satisfies
$\sigma^2(P) - t\sigma(P) + pP = \infty$.

This means
$pP = t\sigma(P) - \sigma^2(P)$
for $t \in [-2\sqrt{p}, 2\sqrt{p}]$.

Expand integer k in base τ
$k = \sum k_i \tau^i$, with
$k_i \in [-\lfloor (p-1)/2 \rfloor, \lceil (p-1)/2 \rceil]$
and compute
$kP = \sum k_i \sigma^i(P)$.
Density of expansion similar to base p expansion, same set of coefficients – but computing $\sigma(P)$ is much cheaper than pP.

Case $p = 2$: $T^2 + (-1)^a T + 2 = 0$

DBL costs $1\mathbf{I} + 2\mathbf{M} + 1\mathbf{S}$.

σ costs $2\mathbf{S}$.

Few tricks (Meier-Staffelbach, Solinas)

$kP = \sum_{i=0}^{n} k_i \sigma^i(P)$,

$k_i \in \{0, 1\}$ for $P \in E(\mathbf{F}_{2^n})$ has average density $1/2$.

$kP = \sum_{i=0}^{n+1} k_i \sigma^i(P)$,

$k_i \in \{-1, 0, 1\}$ for $P \in E(\mathbf{F}_{2^n})$ has average density $1/3$.

Similar to binary and NAF expansion; generalizations of other methods exist.

General case:
Frobenius endomorphism makes
scalar multiplications faster.

Optimal extension fields –
medium size p and n –
get some benefit, too.
OEF assumes p fits word size.

Most extreme cases:
Prime order subgroup $\leq p^{n-1}$.
$n = 3$ or 5: *trace-zero varieties*
$n = 2$: not worthwhile.

Attacks get somewhat faster –
but not devastating, except for
some bad choices.

Other curves with endomorphisms

Gallant-Lambert-Vanstone:

When E has equation
$y^2 = x^3 + ax$ over $\mathbf{F}_p$
with $p \equiv 1 \pmod 4$.
$\phi\colon E \to E,\ (x, y) \mapsto (-x, \sqrt{-1}y)$
Note that $\phi^2 + 1 = 0$.

When E has equation
$y^2 = x^3 + b$ over $\mathbf{F}_p$
with $p \equiv 1 \pmod 3$.
Let $\xi_3 = (1 - \sqrt{-3})/2$.
$\phi\colon E \to E,\ (x, y) \mapsto (\xi_3 x, y)$
Note that $\phi^2 + \phi + 1 = 0$.

Bigger example of GLV method:

When E has equation $y^2 = x^3 - 3x^2/4 - 2x - 1$ over $\mathbf{F}_p$ with $p \equiv 1, 2$ or $4 \pmod 7$.

Denote $\xi = (1 + \sqrt{-7})/2$ and $a = (\xi - 3)/4$.

$\phi\colon E \to E$,

$$(x, y) \mapsto \left(\frac{x^2 - \xi}{\xi^2(x-a)}, \frac{y(x^2 - 2ax + \xi)}{\xi^3(x-a)^2} \right)$$

Note that $\phi^2 - \phi + 2 = 0$.

Computation of $Q = kP$

Gallant-Lambert-Vanstone method, where endomorphism ϕ is different from the Frobenius σ.

Write
$kP = k^{(0)}P + k^{(1)}\phi(P)$,
$\max\left\{|k^{(0)}|, |k^{(1)}|\right\} = O(\sqrt{\ell})$

Key points:
Each $k^{(i)}$ is half as long as $k \in [1, \ell]$.
Computing $\phi(P)$ is easy.
Use joint doublings to quickly evaluate double scalar multiplication.

Idea of joint doublings

To compute

$n_1 P_1 + n_2 P_2 + \cdots + n_m P_m$

compute the doublings together,

i.e. write scalars n_i in binary:

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

Idea of joint doublings

To compute

$n_1 P_1 + n_2 P_2 + \cdots + n_m P_m$

compute the doublings together,

i.e. write scalars n_i in binary:

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

Compute as

$$
2\Big(2(n_{1,l-1}P_1 + n_{2,l-1}P_2 + \cdots + n_{m,l-1}P_m) + (n_{1,l-2}P_1 + n_{2,l-2}P_2 + \cdots + n_{m,l-2}P_m\Big) + \cdots
$$

etc. Even with precomputations, many more adds than doublings.

Examples

$1338P + 1715Q =$
$(10100111010)_2P +$
$(11010110011)_2Q$ takes 20
doublings and 12 additions.

Given precomputed $P + Q$,
we can compute the same
in 10 doublings and 8 additions.
If $-$ is efficient
$(010101001\bar{0}10)_2P +$
$(100\bar{1}0\bar{1}0\bar{1}0101\bar{1})_2Q$ reduces
(compared to first line) the
number of additions to 10,
but needs 21 doublings.

Given precomputed
$P + Q$ and $P - Q$:
$(01010100\bar{1}010)_2 P +$
$(100\bar{1}0\bar{1}0\bar{1}010\bar{1})_2 Q$
needs 11 doublings
and 8 additions.
The joint Hamming weight
has not decreased,
and length has increased.

Combination

GLV curves are rare.

Galbraith-Lin-Scott (GLS)
use Frobenius σ with $n = 2$
– and avoids having big subgroup!

Let E be an elliptic curve defined over $\mathbf{F}_{p^2}$.
Quadratic twist of
$E : y^2 = x^3 + a_4x + a_6$ is
$\tilde{E} : y^2 = x^3 + a_4/c^2x + a_6/c^3$,
$c \in \mathbf{F}_{p^2}$ and $c \neq \blacksquare$ over $\mathbf{F}_{p^2}$.
Start with $\tilde{E}$ over $\mathbf{F}_p$.
(Aha, the subfield idea comes in!)
and pick nonsquare $c \in \mathbf{F}_{p^2}$.

$\tilde{E} : y^2 = x^3 + b_4 x + b_6$; $b_4, b_6 \in \mathbf{F}_p$.

Gets E over $\mathbf{F}_{p^2}$:

$E : y^2 = x^3 + b_4 c^2 x + b_6 c^3$,

$b_4 c^2, b_6 c^3 \in \mathbf{F}_{p^2}$.

No reason that E cannot have (almost) prime order.

Yet E closely related to curve with Frobenius endomorphism.

Define $\psi : E \to E$

as map from E to $\tilde{E}$, followed by p-th power Frobenius on $\tilde{E}$, followed by map back to E.

ψ satisfies $\psi^2 + 1 = 0$ on points of order $\geq 2p$ on E. Can use all GLV tricks; many more curves.

Endomorphisms speed up DLP

In general, an efficiently computable endomorphism ϕ of order r speeds up Pollard rho method by factor $\sqrt{r}$.

Can define walk on classes by inspecting all $2r$ points $\pm P, \pm\phi(P), \ldots, \pm\phi^{r-1}(P)$ to choose unique representative for class and then doing an adding walk.

So $y^2 = x^3 + ax$ and $y^2 = x^3 + b$ come at a security loss of $\sqrt{2}$.

GLS curves also have endomorphisms of order 2.
As in the case of GLV curves, loss of factor $\sqrt{2}$ is fully made up for by the faster arithmetic.

Security of DLP might not be sufficient for your protocol; some are based on hardness of static Diffie-Hellman problem.

Recent observation (Granger 2010): Oracle assisted DHP is easier on GLS curves than on curves over prime fields.

The target: ECC2K-130

The Koblitz curve
$y^2 + xy = x^3 + 1$
over $\mathbf{F}_{2^{131}}$ has 4ℓ points,
where ℓ is prime.
Field representation uses
irreducible polynomial
$f = z^{131} + z^{13} + z^2 + z + 1$.

Certicom generated their
challenge points as two random
points in order-ℓ subgroup by
taking two random points on the
curve and multiplying them by 4.

This produced the following points P, Q:

```
x(P) = 05 1C99BFA6 F18DE467 C80C23B9 8C7994AA
y(P) = 04 2EA2D112 ECEC71FC F7E000D7 EFC978BD
x(Q) = 06 C997F3E7 F2C66A4A 5D2FDA13 756A37B1
y(Q) = 04 A38D1182 9D32D347 BD0C0F58 4D546E9A
```

(unique encoding of $\mathbf{F}_{2^{131}}$ in hex).

The challenge:

Find an integer

$k \in \{0, 1, \ldots, \ell - 1\}$

such that $[k]P = Q$.

Bigger picture:

128-bit curves have been proposed for real (RFID, TinyTate).

Equivalence classes for Koblitz curves

P and $-P$ have same x-coordinate.

Search for x-coordinate collision.

Search space is only $\ell/2$; this gives factor $\sqrt{2}$ speedup ...

provided that $f(P_i) = f(-P_i)$.

Equivalence classes for Koblitz curves

P and $-P$ have same x-coordinate.
Search for x-coordinate collision.
Search space is only $\ell/2$; this
gives factor $\sqrt{2}$ speedup ...
provided that $f(P_i) = f(-P_i)$.

More savings: P and $\sigma^i(P)$ have
$x(\sigma^j(P)) = x(P)^{2^j}$.

Consider equivalence classes under
Frobenius and $\pm$; gain
additional factor $\sqrt{n} = \sqrt{131}$.

Need to ensure that the iteration
function satisfies
$f(P_i) = f(\pm\sigma^j(P_i))$ for any j.

Could again define adding walk starting from $|P_i|$.

Redefine $|P_i|$ as canonical representative of class containing P_i: e.g., lexicographic minimum of P_i, $-P_i$, $\sigma(P_i)$, etc.

Iterations now involve many squarings, but squarings are not so expensive in characteristic 2.

Iteration function for Koblitz curves

Normal basis of finite field

$\mathbf{F}_{2^n}$ has elements
$\{\zeta, \zeta^2, \zeta^{2^2}, \zeta^{2^3}, \ldots, \zeta^{2^{n-1}}\}$.

Representation for x and x^2
$\sum_{i=0}^{n-1} x_i \zeta^{2^i} = (x_0, x_1, x_2, \ldots, x_{n-1})$
$\sum_{i=1}^{n} x_i \zeta^{2^i} = (x_{n-1}, x_0, \ldots, x_{n-2})$
using $(\zeta^{2^{n-1}})^2 = \zeta^{2^n} = \zeta$.

Harley and Gallant-Lambert-Vanstone observe that in normal basis, $x(P)$ and $x(P)^{2^j}$ have same Hamming weight

$$\mathrm{HW}(x(P)) = \sum_{i=0}^{n-1} x_i$$

(addition over $\mathbf{Z}$).

Suggestion:

$$P_{i+1} = P_i + \sigma^j(P_i),$$

as iteration function.

Choice of j depends on HW($x(P)$).

This ensures that the walk is well defined on classes since

$$\begin{aligned} f(\pm\sigma^m(P_i)) = \\ \pm\sigma^m(P_i) + \sigma^j(\pm\sigma^m(P_i)) = \\ \pm(\sigma^m(P_i) + \sigma^m(\sigma^j(P_i))) = \\ \pm\sigma^m(P_i + \sigma^j(P_i)) = \\ \pm\sigma^m(P_{i+1}). \end{aligned}$$

GLV suggest using
$j = \mathrm{hash}(\mathrm{HW}(x(P)))$,
where the hash function
maps to $[1, n]$.

Harley uses a smaller set of
exponents; for his attack on
ECC2K-108 he takes
$j \in \{1, 2, 4, 5, 6, 7, 8\}$;
computed as
$j = (\mathrm{HW}(x(P)) \bmod 7) + 2$
and replacing 3 by 1.

Our choice of iteration function

Restricting size of j matters – squarings are cheap but:

- in bitslicing need to compute all powers (no branches allowed);
- code size matters (in particular for Cell CPU);
- logic costs area for FPGA;
- having a large set doesn't actually gain much randomness.

Analysis of the loss in randomness similar to Wednesday's.

Having few coefficients lets us exclude short fruitless cycles.

To do so, compute
the shortest vector in the lattice
$\left\{v : \prod_j (1+\sigma^j)^{v_j} = 1\right\}$.
Usually the shortest vector has negative coefficients (which cannot happen with the iteration); shortest vector with positive coefficients is somewhat longer.

For implementation it is better to have a continuous interval of exponents, so shift the interval if shortest vector is short.

Our iteration function:

$P_{i+1} = P_i + \sigma^j(P_i)$ where

$j = (\mathrm{HW}(x(P))/2 \bmod 8) + 3$,

so $j \in \{3, 4, 5, 6, 7, 8, 9, 10\}$.

Shortest combination of these powers is long.

Note that $\mathrm{HW}(x(P))$ is even.

Iteration consists of

- computing the Hamming weight $\mathrm{HW}(x(P))$ of the normal-basis representation of $x(P)$;
- checking for distinguished points (is $\mathrm{HW}(x(P)) \leq 34$?);
- computing j and $P + \sigma^j(P)$.

Analysis of our iteration function

For a perfectly random walk
$\approx \sqrt{\pi \ell / 2}$ iterations
are expected on average.
Have $\ell \approx 2^{131}/4$ for ECC2K-130.

A perfectly random walk
on classes under $\pm$ and Frobenius
would reduce number of iterations
by $\sqrt{2 \cdot 131}$.

Analysis of our iteration function

For a perfectly random walk
$\approx \sqrt{\pi \ell / 2}$ iterations
are expected on average.
Have $\ell \approx 2^{131}/4$ for ECC2K-130.

A perfectly random walk
on classes under $\pm$ and Frobenius
would reduce number of iterations
by $\sqrt{2 \cdot 131}$.

Loss of randomness
from having only 8 choices of j.
Further loss from non-randomness
of Hamming weights:

Hamming weights around 66
are much more likely than at the
edges; effect still noticeable
after reduction to 8 choices.

Hamming weights around 66
are much more likely than at the
edges; effect still noticeable
after reduction to 8 choices.

Our $\sqrt{1 - \sum_i p_i^2}$ heuristic says
that the total loss is 6.9993%.
(Higher-order anti-collision
analysis: actually above 7%.)
This loss is justified by
the very fast iteration function.

Hamming weights around 66
are much more likely than at the
edges; effect still noticeable
after reduction to 8 choices.

Our $\sqrt{1 - \sum_i p_i^2}$ heuristic says
that the total loss is 6.9993%.
(Higher-order anti-collision
analysis: actually above 7%.)
This loss is justified by
the very fast iteration function.

Average number of iterations for
our attack against ECC2K-130:
$\sqrt{\pi \ell / (2 \cdot 2 \cdot 131)} \cdot 1.069993$
$\approx 2^{60.9}$.

Some highlights:

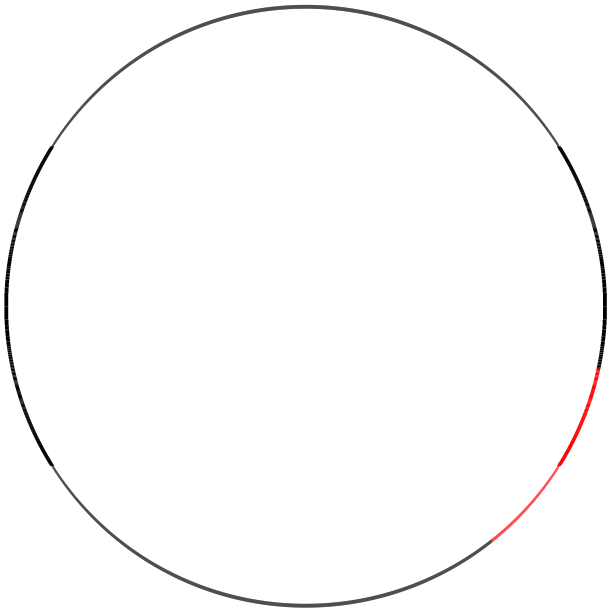
Detailed analysis of randomness of iteration function.

Could increase randomness of the walk but then iteration function gets slower. Optimized for time per iteration $\times$ # iterations.

Do not remember multiset of j's; instead recompute this from seed when collision is found (cheaper, less storage).

Comparative study of normal basis and polynomial basis representation;
new: optimal polynomial bases.

DLs in intervals



Want to use knowledge that DL is in a small interval $[a, b]$, much smaller than ℓ.

We can use this in baby-step giant-step algorithm.

How to use this in a memory-less algorithm?

Standard interval method:
Pollard's kangaroo method.

Pollard's kangaroos do small jumps around the interval.

Standard interval method:

Pollard's kangaroo method.

Pollard's kangaroos do small jumps around the interval.

Real kangaroos sleep

Standard interval method:
Pollard's kangaroo method.

Pollard's kangaroos do small jumps around the interval.

Real kangaroos sleep

(at least outside Australia).

Kangaroo method

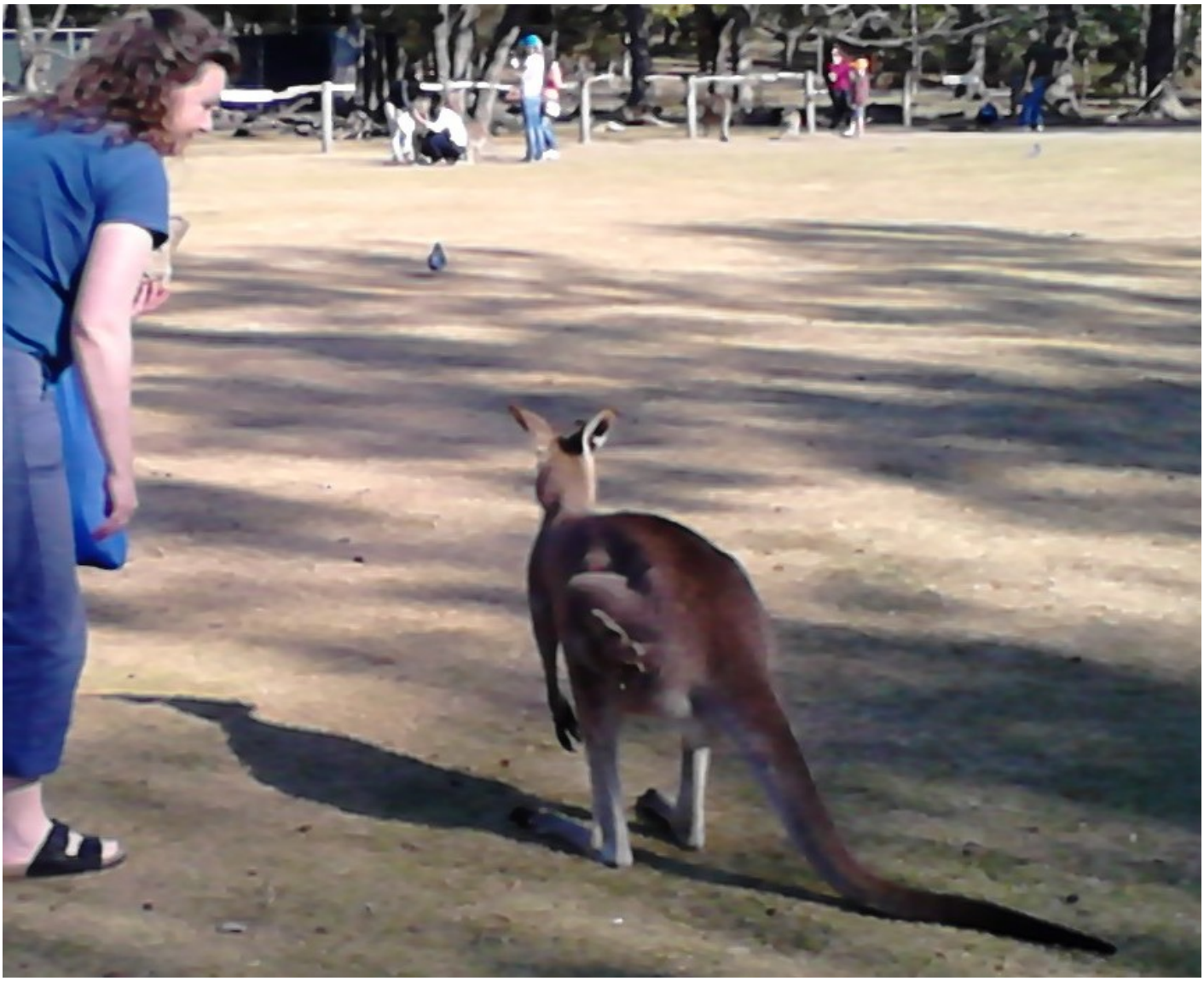
in Australia

Main actor:

The tame kangaroo

starts at a known
multiple of P, e.g. bP.

The tame kangaroo jumps.



Jumps are determined
by current position.

The tame kangaroo jumps.

Jumps are determined by current position.

Average jump distance is $\sqrt{b-a}$.

The tame kangaroo jumps.

Jumps are determined by current position.

Average jump distance is $\sqrt{b-a}$.

The tame kangaroo jumps.

Jumps are determined by current position.

Average jump distance is $\sqrt{b-a}$.

The tame kangaroo stops

after a fixed number of jumps (about $\sqrt{b-a}$ many).

Installs a trap and waits.

The wild kangaroo

starts at point Q.

Follows the same instructions for jumps.

But we don't know where
the starting point Q is.
Know $Q = nP$ with $n \in [a, b]$.

Hope that the paths of the tame
and wild kangaroo intersect.

Similar to the rho method the
kangaroos will hop on the same
path from that point onwards.

Eventually the wild kangaroo falls
into the trap.
(Or disappears in the distance if
paths have not intersected.
Start a fresh one
from $Q + P, Q + 2P, \ldots$)

Same story in math

Kangaroo = sequence $X_i \in \langle P \rangle$.

Starting point $X_0 = s_0 P$.

Distance $d_0 = 0$.

Step set: $S = \{s_1 P, \ldots, s_L P\}$,

with s_i on average

$s = \beta\sqrt{b-a}$.

Hash function

$H : \langle P \rangle \to \{1, 2, \ldots, L\}$.

Update function

$d_{i+1} = d_i + s_{H(X_i)}, \quad i = 0, 1, 2, \ldots,$

$X_{i+1} = X_i + s_{H(X_i)} P, \quad i = 0, 1, 2, \ldots$

Tame kangaroo starts at

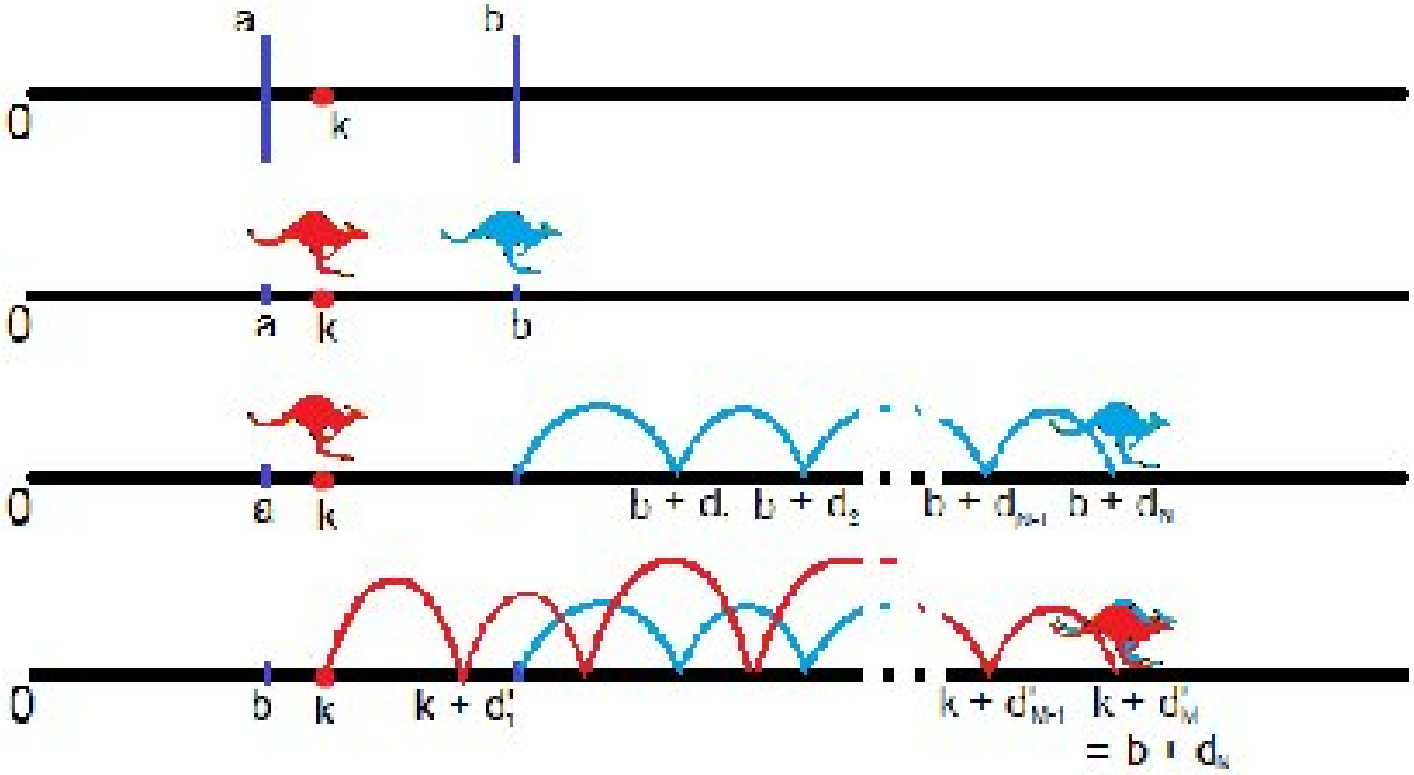
$X_0 = bP$,

wild kangaroo starts at

$X'_0 = Q = nP$.

Trap: distance d_N,

endpoint $X_N = (b + d_N)P$.



Picture credit:

Christine van Vredendaal.

Parallel kangaroo method

Use an entire herd

of tame kangaroos,

all starting

around $(b - a)/2P$ ...

. . . and define certain spots as distinguished points

Also start a herd of
wild kangaroos around Q.
Hope that one wild and
one tame kangaroo
meet at one distinguished point.