DLP-II

and curves with endomorphisms

Tanja Lange
Technische Universiteit Eindhoven

Additive walks

Generic rho method
$f(W_i) = a(W_i)P + b(W_i)Q$
requires two scalar multiplications
for each iteration.
Could replace by double-scalar
multiplication; could further
merge the 2-scalar multiplications
across several parallel iterations.

Additive walks

Generic rho method
$f(W_i) = a(W_i)P + b(W_i)Q$
requires two scalar multiplications for each iteration.
Could replace by double-scalar multiplication; could further merge the 2-scalar multiplications across several parallel iterations.

More efficient: use *additive walk*:
Start with $W_0 = a_0 P$ and put
$f(W_i) = W_i + c_j P + d_j Q$
where $j = h(W_i)$.

Pollard's initial proposal:

Use $x(W_i)$ mod 3 as h

and update:

$$W_{i+1} = \begin{cases} W_i + P & \text{for } x(W_i) \bmod 3 = 0 \\ 2W_i & \text{for } x(W_i) \bmod 3 = 1 \\ W_i + Q & \text{for } x(W_i) \bmod 3 = 2 \end{cases}$$

Easy to update a_i and b_i.

$$(a_{i+1}, b_{i+1}) = \begin{cases} (a_i + 1, b_i) & \text{for } x(W_i) \bmod 3 = 0 \\ (2a_i, 2b_i) & \text{for } x(W_i) \bmod 3 = 1 \\ (a_i, b_i + 1) & \text{for } x(W_i) \bmod 3 = 2 \end{cases}$$

Additive walk requires only one addition per iteration.
h maps from $\langle P\rangle$ to $\{0, 1, \ldots, r-1\}$, and $R_j = c_j P + d_j Q$ are precomputed for each $j \in \{0, 1, \ldots, r-1\}$.

Easy coefficient update:
$W_i = a_i P + b_i Q$,
where a_i and b_i are defined recursively as follows:
$a_{i+1} = a_i + c_{h(W_i)}$ and
$b_{i+1} = b_i + d_{h(W_i)}$.

Additive walks have disadvantages:
The walks are noticeably nonrandom; this means they need more iterations than the generic rho method to find a collision.

This effect disappears as r grows, but but then the precomputed table $R_0, \ldots, R_{r-1}$ does not fit into fast memory. This depends on the platform, e.g. trouble for GPUs.
More trouble with adding walks later.

Randomness of adding walks

Let $h(W) = i$ with probability p_i.
Fix a point T, and let W and W' be two independent uniform random points.
Let $W \neq W'$ both map to T.
This event occurs if there are $i \neq j$ such that simultaneously:
$T = W + R_i = W' + R_j$;
$h(W) = i$; $h(W') = j$.

These conditions have probability $1/\ell^2$, p_i, and p_j respectively.

Summing over all (i, j) gives the overall probability $\left(\sum_{i \neq j} p_i p_j\right) / \ell^2 = \left(\sum_{i,j} p_i p_j - \sum_i p_i^2\right) / \ell^2 = \left(1 - \sum_i p_i^2\right) / \ell^2$.

This means that the probability of an immediate collision from W and W' is $\left(1 - \sum_i p_i^2\right) / \ell$, where we added over the ℓ choices of T. In the simple case that all the p_i are $1/r$, the difference from the optimal $\sqrt{\pi \ell / 2}$ iterations is a factor of $1/\sqrt{1 - 1/r} \approx 1 + 1/(2r)$.

Various heuristics leading to standard $\sqrt{1-1/r}$ formula in different ways:

1981 Brent–Pollard;

2001 Teske;

2009 ECC2K-130 paper, eprint 2009/541.

Various heuristics leading to
standard $\sqrt{1-1/r}$ formula
in different ways:
1981 Brent–Pollard;
2001 Teske;
2009 ECC2K-130 paper,
eprint 2009/541.

2010–2012 Bernstein–Lange:
Standard formula is wrong!
There is a further slowdown
from higher-order anti-collisions:
e.g. $W+R_i+R_k \neq W'+R_j+R_l$
if $R_i+R_k = R_j+R_l$.
For details see
"Two grumpy giants and a baby".

Eliminating storage

Usual description: each walk keeps track of a_i and b_i with $W_i = a_i P + b_i Q$.

This requires each client to implement arithmetic modulo ℓ or at least keep track of how often each R_j is used.

For distinguished points these values are transmitted to server (bandwidth) which stores them as e.g. (W_i, a_i, b_i) (space).

2009 ECC2K-130 paper:
Remember where you started.
If $W_i = W_j$ is the collision of distinguished points,
can recompute these walks
with a_i, b_i, a_j, and b_j;
walk is deterministic!
Server stores 2^{45} distinguished points; only needs to know coefficients for 2 of them.

Our setup: Each walk remembers seed; server stores distinguished point and seed.
Saves time, bandwidth, space.

Negation and rho

$W = (x, y)$ and $-W = (x, -y)$
have same x-coordinate.
Search for x-coordinate collision.

Search space for collisions is
only $\lceil \ell/2 \rceil$; this gives factor $\sqrt{2}$
speedup . . . if $f(W_i) = f(-W_i)$.

To ensure $f(W_i) = f(-W_i)$:
Define $j = h(|W_i|)$ and
$f(W_i) = |W_i| + c_j P + d_j Q$.
Define $|W_i|$ as, e.g., lexicographic
minimum of $W_i, -W_i$.
This negation speedup
is textbook material.

Problem: this walk can run into fruitless cycles!

Example: If $|W_{i+1}| = -W_{i+1}$ and $h(|W_{i+1}|) = j = h(|W_i|)$ then $W_{i+2} = f(W_{i+1}) = -W_{i+1} + c_j P + d_j Q = -(|W_i| + c_j P + d_j Q) + c_j P + d_j Q = -|W_i|$ so $|W_{i+2}| = |W_i|$ so $W_{i+3} = W_{i+1}$ so $W_{i+4} = W_{i+2}$ etc.

If h maps to r different values then expect this example to occur with probability $1/(2r)$ at each step.

Known issue, not quite textbook.

1999 Gallant–Lambert–Vanstone "Improving the parallelized Pollard lambda search on anomalous binary curves":

"For example, the cycle could be traversed, the lexicographically least label identified, and a modified iteration taking us out of the cycle could be applied at the point or equivalence class corresponding to this identified label."

1999 Duursma–Gaudry–Morain "Speeding up the discrete log computation on curves with automorphisms":

"If the cycle is $R_1 \mapsto R_2 \mapsto \cdots \mapsto R_t$, we want to get out of it in a *symmetric* way . . . Our version is to sort the points R_i to obtain $S_1, S_2, \ldots, S_t$ and start again, say, from $R = \oplus_{i=1}^{t} [i^i + 1] S_i$. Anything that breaks linearity would be convenient."

e.g. Sort 2-cycle,
obtaining $S_1 \leq S_2$.
Duursma–Gaudry–Morain "start again, say, from" $2S_1 + 5S_2$.

Gallant–Lambert–Vanstone keep only S_1 and apply a "modified iteration" but are vague about the choice of modified iteration.
Maybe $2S_1$?

2009 Bos–Kaihara–Kleinjung–Lenstra–Montgomery use $2S_1$.

Current ECDL record:
2009.07 Bos–Kaihara–Kleinjung–Lenstra–Montgomery

Break DLP on
standard curve over $\mathbf{F}_p$
where $p = (2^{128} - 3)/(11 \cdot 6949)$.

Current ECDL record:
2009.07 Bos–Kaihara–Kleinjung–
Lenstra–Montgomery

Break DLP on
standard curve over $\mathbf{F}_p$
where $p = (2^{128} - 3)/(11 \cdot 6949)$.

Did not use negation map to
obtain $\sqrt{2}$ speedup.

Current ECDL record:
2009.07 Bos–Kaihara–Kleinjung–Lenstra–Montgomery

Break DLP on standard curve over $\mathbf{F}_p$ where $p = (2^{128} - 3)/(11 \cdot 6949)$.

Did not use negation map to obtain $\sqrt{2}$ speedup.

Some controversy about this.
Justification after the fact
2010.07 Bos–Kleinjung–Lenstra
"On the use of the negation map in the Pollard rho method"

Bernstein, Lange, Schwabe
(PKC 2011):

Our software solves
random ECDL on the same curve
(with no precomputation)
in 35.6 PS3 years on average.

For comparison:
Bos–Kaihara–Kleinjung–Lenstra–
Montgomery software
uses 65 PS3 years on average.

Bernstein, Lange, Schwabe
(PKC 2011):

Our software solves
random ECDL on the same curve
(with no precomputation)
in 35.6 PS3 years on average.

For comparison:
Bos–Kaihara–Kleinjung–Lenstra–
Montgomery software
uses 65 PS3 years on average.

First big speedup:
We use the negation map.
Second speedup: Fast arithmetic.

Bos–Kleinjung–Lenstra say that “on average more elliptic curve group operations are required per step of each walk. This is unavoidable” etc.

Specifically: If the precomputed additive-walk table has r points, need 1 extra doubling to escape a cycle after $\approx 2r$ additions. And more: “cycle reduction” etc.

Bos–Kleinjung–Lenstra say that the benefit of large r is “wiped out by cache inefficiencies.”

Eliminating fruitless cycles

Issue of fruitless cycles is known and several fixes are proposed. See appendix of full version ePrint 2011/003 for even more details and historical comments.

Summary: most of them got it wrong.

Eliminating fruitless cycles

Issue of fruitless cycles is known and several fixes are proposed. See appendix of full version ePrint 2011/003 for even more details and historical comments.

Summary: most of them got it wrong.

So what to do?
Choose a big r, e.g. $r = 2048$.
$1/(2r) = 1/4096$ small;
cycles infrequent.

Define $|(x, y)|$ to mean
(x, y) for $y \in \{0, 2, 4, \ldots, p - 1\}$
or
$(x, -y)$ for $y \in \{1, 3, 5, \ldots, p - 2\}$.

Precompute points
$R_0, R_1, \ldots, R_{r-1}$ as known
random multiples of P.

Define $|(x, y)|$ to mean
(x, y) for $y \in \{0, 2, 4, \ldots, p-1\}$
or
$(x, -y)$ for $y \in \{1, 3, 5, \ldots, p-2\}$.

Precompute points
$R_0, R_1, \ldots, R_{r-1}$ as known
random multiples of P. Here you
can do full scalar multiplication in
inversion-free coordinates!
Start each walk at a point
$W_0 = |b_0 Q|$,
b_0 is chosen randomly.
Compute $W_1, W_2, \ldots$ as
$W_{i+1} = |W_i + R_{h(W_i)}|$.

Occasionally, every w iterations, check for fruitless cycles of length 2.

For those cases change the definition of W_i as follows:

Compute W_{i-1} and check whether $W_{i-1} = W_{i-3}$.

If $W_{i-1} \neq W_{i-3}$, put $W_i = W_{i-1}$.

If $W_{i-1} = W_{i-3}$, put $W_i = |2 \min\{W_{i-1}, W_{i-2}\}|$, where min means lexicographic minimum.

Doubling the point makes it escape the cycle.

Cycles of length 4, 6, or 12
occur far less frequently.
Cycles of length 4, or 6
are detected when checking
for cycles of length 12;
so skip individual ones.

Same way of escape:
define $W_i =$
$$|2\min\{W_{i-1}, W_{i-2}, W_{i-3}, W_{i-4}, W_{i-5}, W_{i-6}, W_{i-7}, W_{i-8}, W_{i-9}, W_{i-10}, W_{i-11}, W_{i-12}\}|$$
if trapped
and $W_i = W_{i-1}$ otherwise.

Do not store all these points!

When checking for cycle,
store only potential entry point
W_{i-13} (one coordinate, for
comparison) and the
smallest point encountered since
(to escape).

For large DLP
look for larger cycles;
in general, look for
fruitless cycles of even lengths
up to $\approx (\log \ell)/(\log r)$.

How to choose w?

Fruitless cycles of length 2 appear
with probability $\approx 1/(2r)$.
These cycles persist
until detected.
After w iterations,
probability of cycle $\approx w/(2r)$,
wastes $\approx w/2$ iterations
(on average) if it does appear.

Do not choose w
as small as possible!
If a cycle has *not* appeared then
the check wastes an iteration.

The overall loss is approximately
$1 + w^2/(4r)$ iterations out of w.
To minimize the quotient
$1/w + w/(4r)$ we take $w \approx 2\sqrt{r}$.

Cycles of length $2c$ appear with
probability $\approx 1/r^c$,
optimal checking frequency is
$\approx 1/r^{c/2}$.
Loss rapidly disappears
as c increases.
Can use lcm of cycle lengths
to check.

Concrete example: 112-bit DLP

Use $r = 2048$. Check for 2-cycles every 48 iterations.
Check for larger cycles much less frequently.
Unify the checks for 4-cycles and 6-cycles into a check for 12-cycles every 49152 iterations.

Choice of r has big impact!
$r = 512$ calls for checking for 2-cycles every 24 iterations.
In general, negation overhead $\approx$ doubles when table size is reduced by factor of 4.

Why are we confident this works?

We only have one PlayStation 3, not the 200 that Lausanne has, and we want to wait for 36 years to show that we actually compute the right thing.

Why are we confident this works?

We only have one PlayStation 3, not the 200 that Lausanne has, and we want to wait for 36 years to show that we actually compute the right thing.

Can produced scaled versions:
Use *same* prime field
(so that we can compare the field arithmetic)
and same curve shape
$y^2 = x^3 - 3x + b$
but vary b to get curves with small subgroups.

This produces other curves, and many of those have smaller order subgroups.
Specify DLP in subgroup of size 2^{50}, or 2^{55}, or 2^{60} and show that the actual running time matches the expectation.
And that DLP is correct.

We used same property for a point to be distinguished as in big attack; probability is 2^{-20}.
Need to watch out that walks do not run into rho-type cycles (artefact of small group order).
We aborted overlong walks.

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 $(x, y) \in k \times k$ 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 $a_1y = 3x^2 + 2a_2x + a_4$.

Easy to check nonsingularity. Almost all curves are nonsingular when k is large.

An example over **R**

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

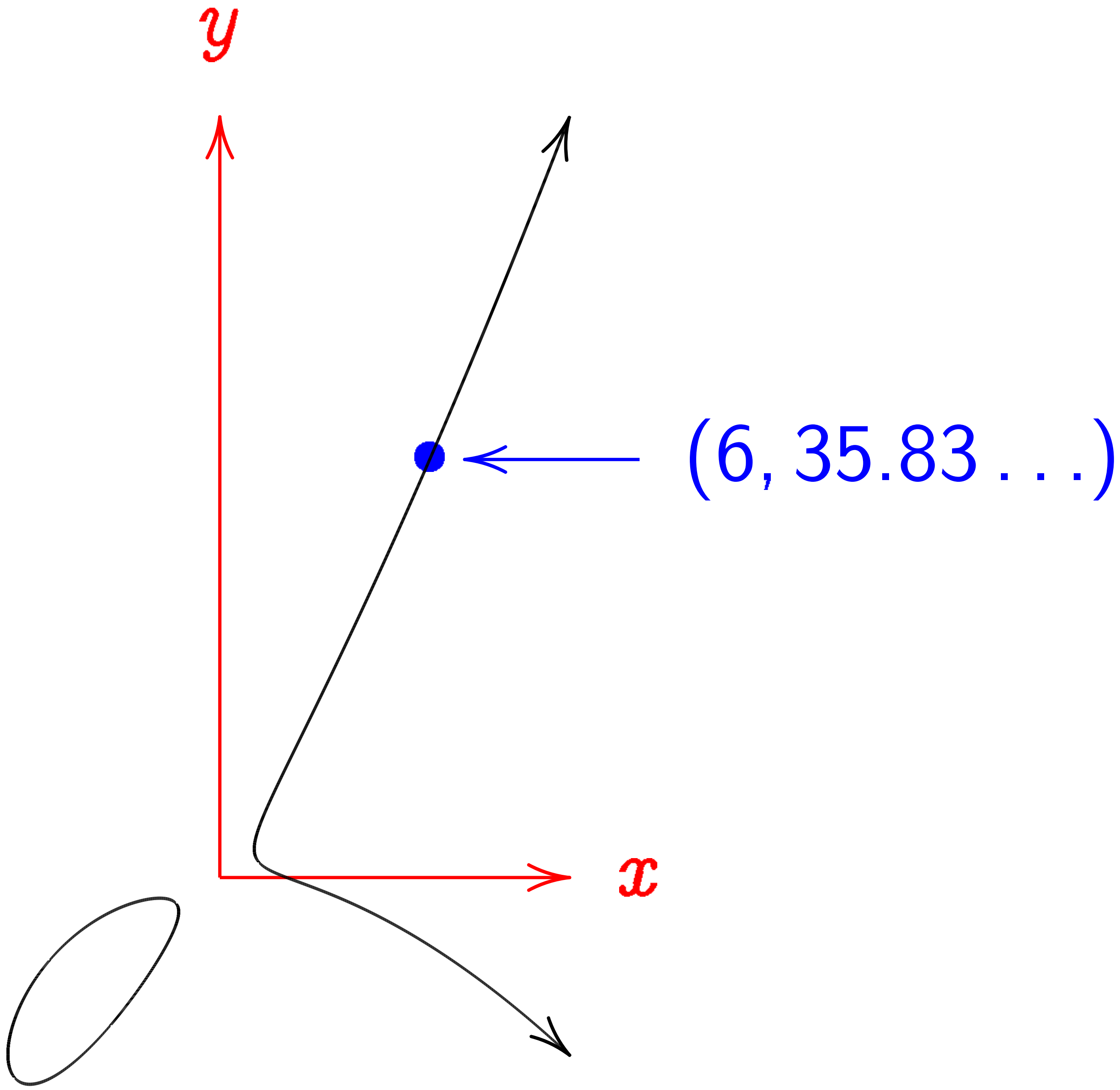
The "points on the elliptic curve
$y^2 - 5xy = x^3 - 7$ over **R**"
are those pairs and
one additional point, ∞.

i.e. The set of points is

$$\{(x, y) \in \mathbf{R} \times \mathbf{R} : \\ y^2 - 5xy = x^3 - 7\} \cup \{\infty\}.$$

(**R** is the set of real numbers.)

Graph of this set of points:



Don't forget ∞.

Visualize ∞ as top of y axis.

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

Consider the non-prime field

$$(\mathbf{Z}/2)[t]/(t^4 - t - 1) = \{$$

$$0t^3 + 0t^2 + 0t^1 + 0t^0,$$

$$0t^3 + 0t^2 + 0t^1 + 1t^0,$$

$$0t^3 + 0t^2 + 1t^1 + 0t^0,$$

$$0t^3 + 0t^2 + 1t^1 + 1t^0,$$

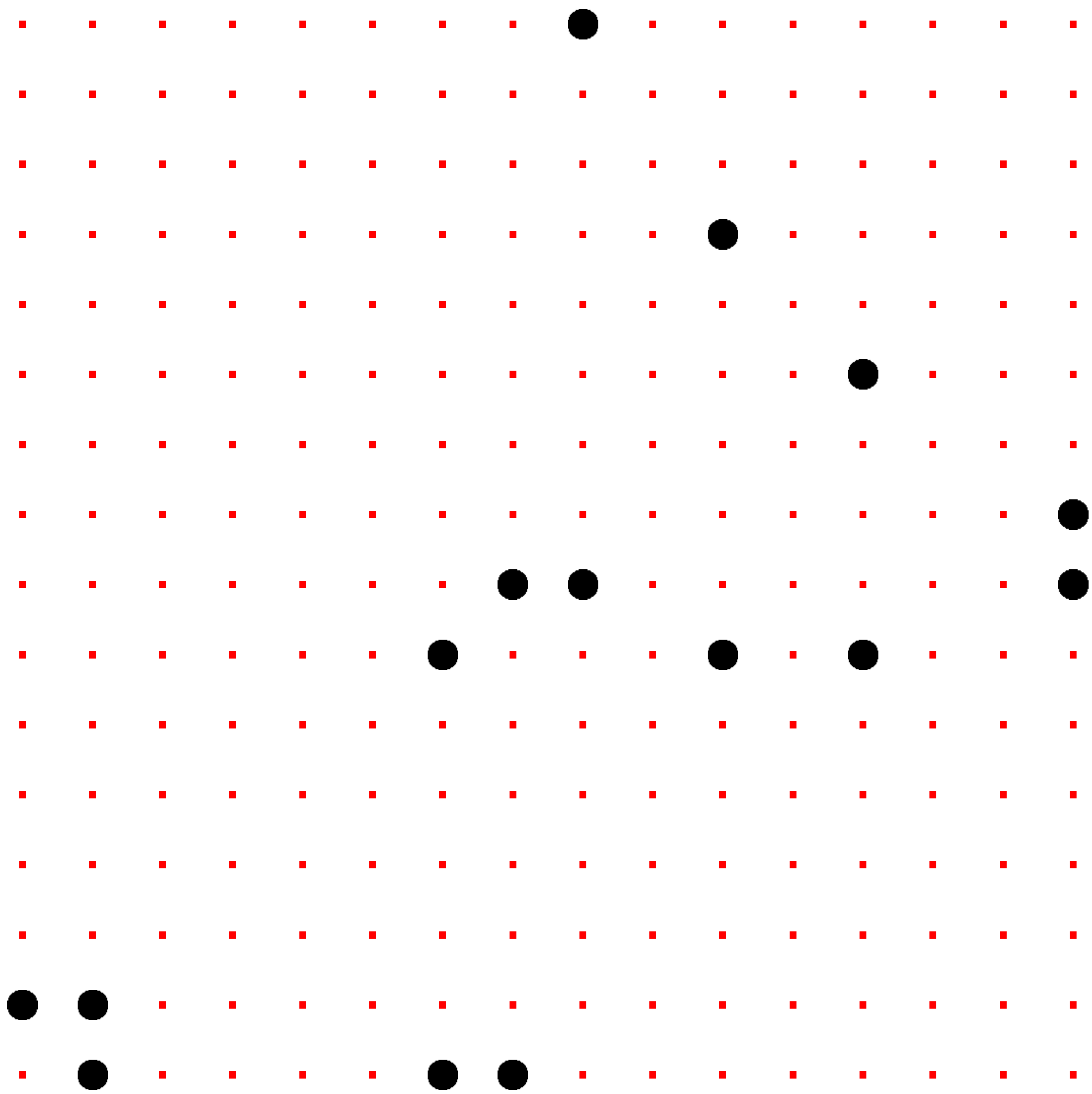
$$0t^3 + 1t^2 + 0t^1 + 0t^0,$$

$$\vdots$$

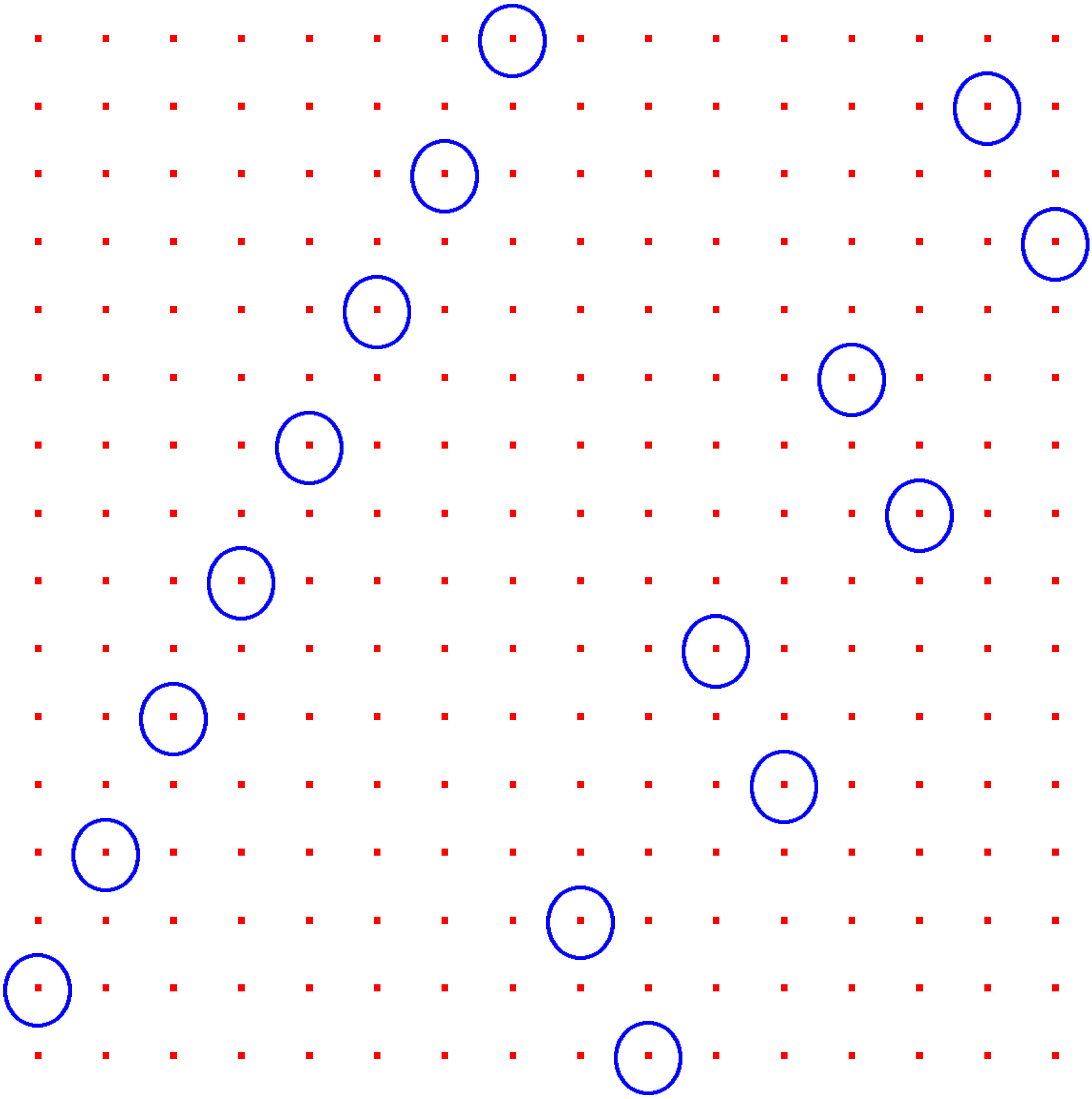
$$1t^3 + 1t^2 + 1t^1 + 1t^0\}$$

of size $2^4 = 16$.

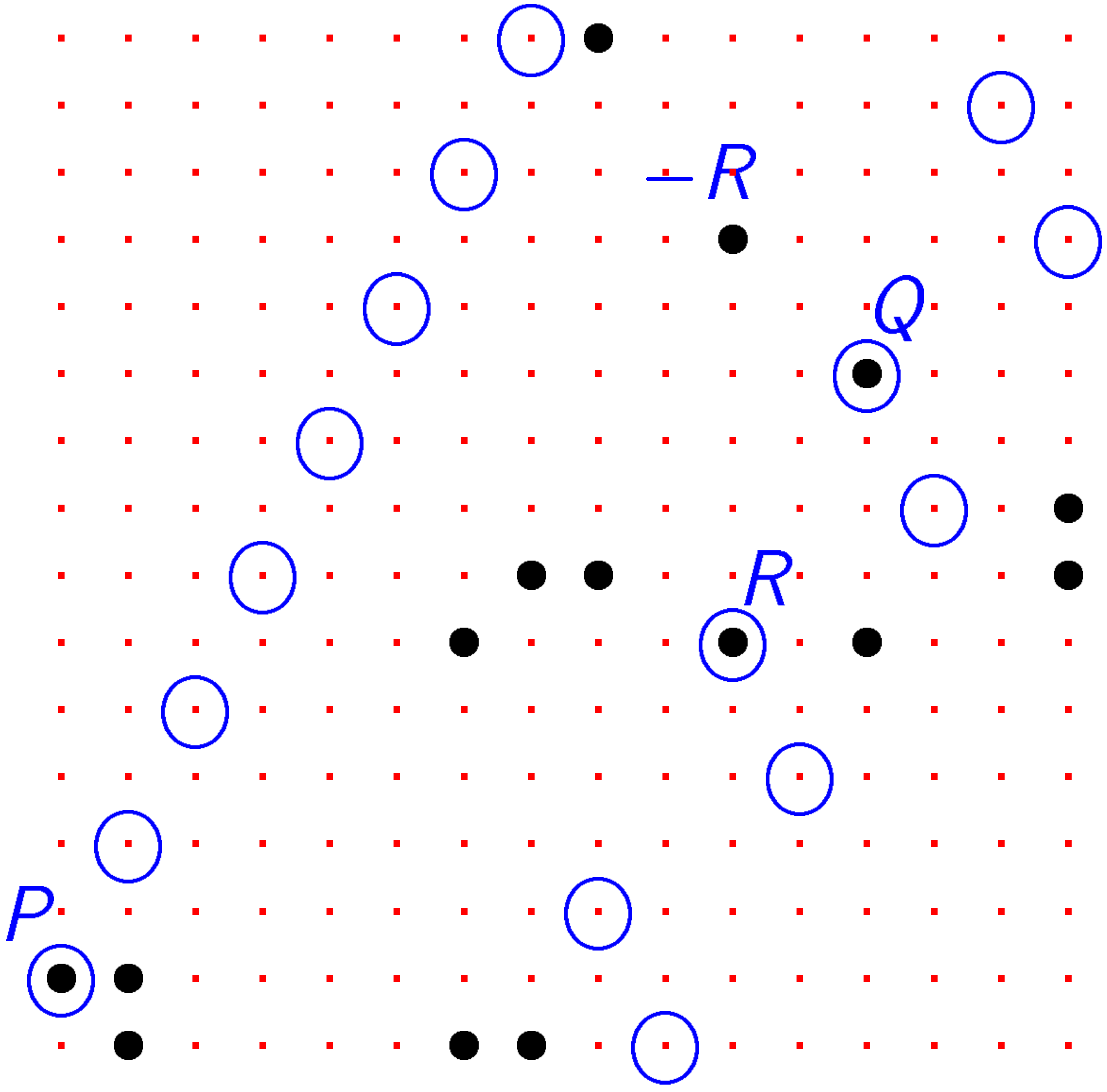
Graph of the "set of points on the elliptic curve $y^2 - 5xy = x^3 - 7$ over $(\mathbf{Z}/2)[t]/(t^4 - t - 1)$":



Line $y = tx + 1$:



$P + Q = -R$:



General addition law

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

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

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

where $\lambda =$
$$\begin{cases} (y_R - y_P)/(x_R - x_P) & x_P \neq x_R, \\ \frac{3x_P^2 + 2a_2 x_P + a_4 - a_1 y_P}{2y_P + a_1 x_P + a_3} & P = R \neq -R \end{cases}$$

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

$a = (\xi - 3)/4$.

ϕ: $E \to E$,

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

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 Sparse Form to quickly evaluate double scalar multiplication.

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
$E : y^2 = x^3 + a_4 x + a_6$ is
$\tilde{E} : y^2 = x^3 + a_4/c^2 x + a_6/c^3$,
$c \in \mathbf{F}_{p^2}$ 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.