Edwards coordinates for elliptic curves

D. J. Bernstein
University of Illinois at Chicago

Joint work with:
Tanja Lange
Technische Universiteit Eindhoven

The Domain Name System

Mail sender at `columbia.edu`

↑

DNS packet:
"The mail server for
`uottawa.ca`
has IP address
137.122.6.57."

Administrator at `uottawa.ca`

Now `columbia.edu`
sends mail to 137.122.6.57.

Is this system secure?

Many security holes
in DNS software:
BIND libresolv buffer overflow,
Microsoft cache promiscuity,
BIND 8 TSIG buffer overflow,
BIND 9 dig promiscuity, etc.

Fix: Use better DNS software.
`http://cr.yp.to/djbdns.html`

But what about protocol holes?

Stealing mail by attacking DNS

Mail sender at `columbia.edu`

DNS packet:
"The mail server for
`uottawa.ca`
has IP address
131.193.36.27."

Attacker anywhere on network

Now `columbia.edu`
sends mail to 131.193.36.27.
Real `uottawa.ca` never sees it.
No warning to `columbia.edu`.

Are attacks really so easy?

Can attacker guess
where mail is being sent?

Can attacker guess
time when mail is being sent?

Can attacker guess
UDP port for DNS packet?

Can attacker guess
the random 16-bit ID
that the mail sender
puts into its DNS request?

For sniffing attackers, yes; but
attackers anywhere on network?

Three weeks ago: Emergency security update for BIND to change ID generation.

Previous ID generator was cryptanalyzed by Amit Klein: "This is a weak version (since the output is 16 bits, as opposed to the traditional 1 bit) of the … mutually clock controlled (LFSR) generator …"

Attacker legitimately receives 13 successive IDs from sender, reconstructs stream-cipher state, predicts sender's subsequent IDs.

Add signatures to DNS?

Long IDs and strong generators don't stop sniffing attackers.

Obvious solution:
Public-key signatures in packets.

But many deployment obstacles:
many DNS implementations;
many different databases;
tiny packets, 512 bytes;
heavily loaded senders;
heavily loaded receivers.

Current Internet situation:
0% of DNS packets are signed.

Can change DNS-security protocol to minimize effects on implementations, databases.

But still need extremely small, extremely fast signatures with extremely fast verification.

For fastest verification: state-of-the-art Rabin-Williams. But that could be trouble for signature time, space.

Let's instead choose an elliptic-curve signature system.

Start by choosing
high-speed, high-security
elliptic curve: "Curve25519"
(Bernstein, PKC 2006).

This is the elliptic curve
$y^2 = x^3 + 486662x^2 + x$
modulo the prime $2^{255} - 19$.

Standard base point B with
known prime order $q \approx 2^{252}$:
$(9, \sqrt{39420360} \bmod 2^{255} - 19)$.

Also choose a high-speed,
high-security hash function H.

I offer US$1000 prize for
the public Rumba20 cryptanalysis
that I consider most interesting.
Awarded at the end of 2007.

Rumba20 is a function from
192 bytes to 64 bytes;
designed for collision-resistance.

`http://cr.yp.to`
`/rumba20.html`

A sensible ElGamal-type system (van Duin, `sci.crypt`, 2006):

Signer has 32-byte secret key k.

Everyone knows sender's 32-byte public key: compressed kB.
Here $kB = k$th multiple of B in the Curve25519 group.

To verify (m, compressed R, t): verify $tB = H(R, m)R + kB$.

To sign m: generate a secret s; $R = sB$; $t = H(R, m)s + k \bmod q$.

No tricky inversions mod q.
More advantages, as we'll see.

Elliptic-curve arithmetic

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

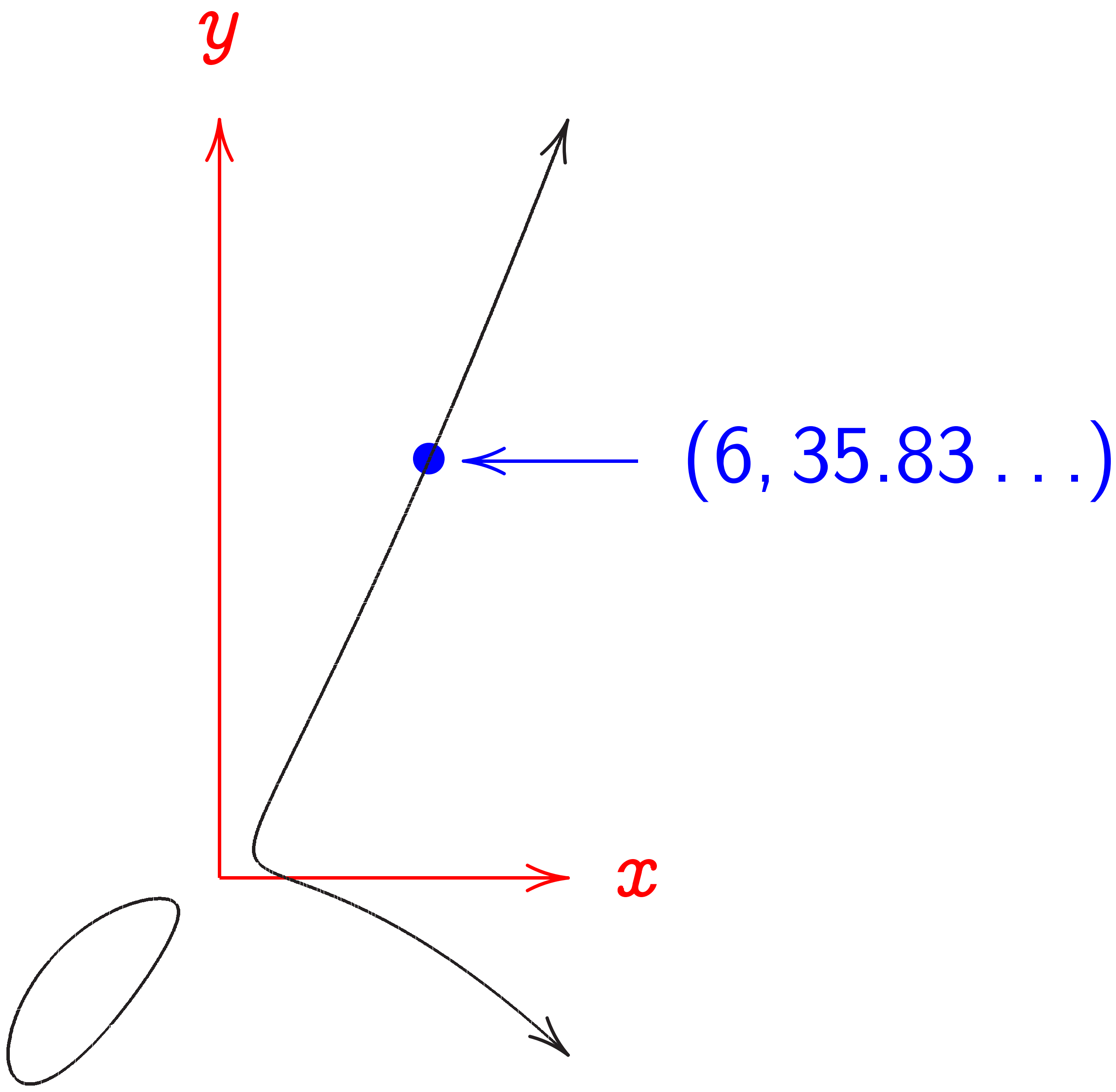
The "points on the elliptic curve
$y^2 - 5xy = x^3 - 7$ over **R**"
are those pairs and
one additional point, ∞.

i.e. The set of points is
$\{(x, y) \in \mathbf{R}^2 :$

$$y^2 - 5xy = x^3 - 7 \;\; \cup \{\infty \;\; .$$

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

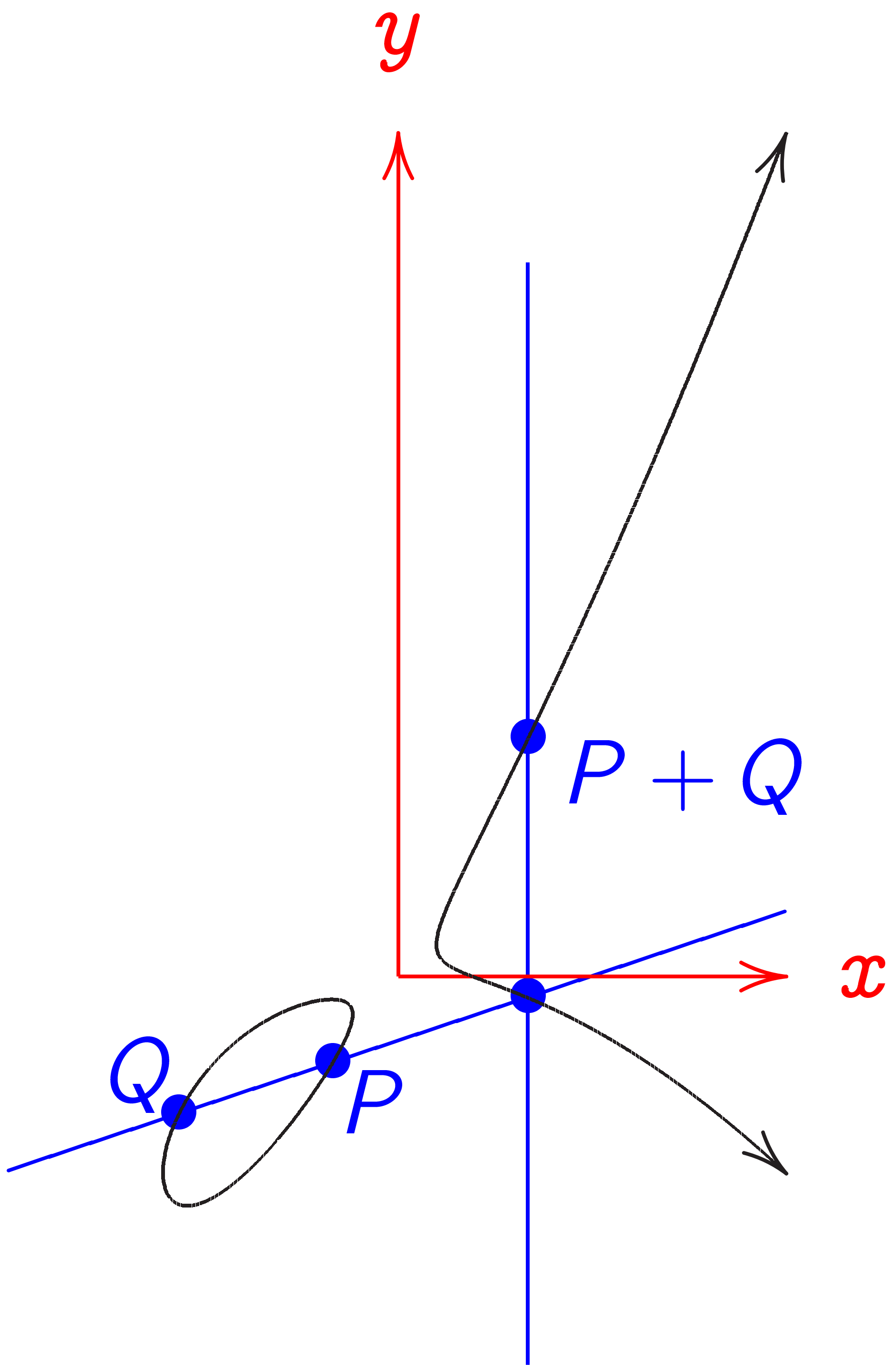
Graph of this set of points:



Don't forget ∞.

Visualize ∞ as top of y axis.

Elliptic-curve addition law:



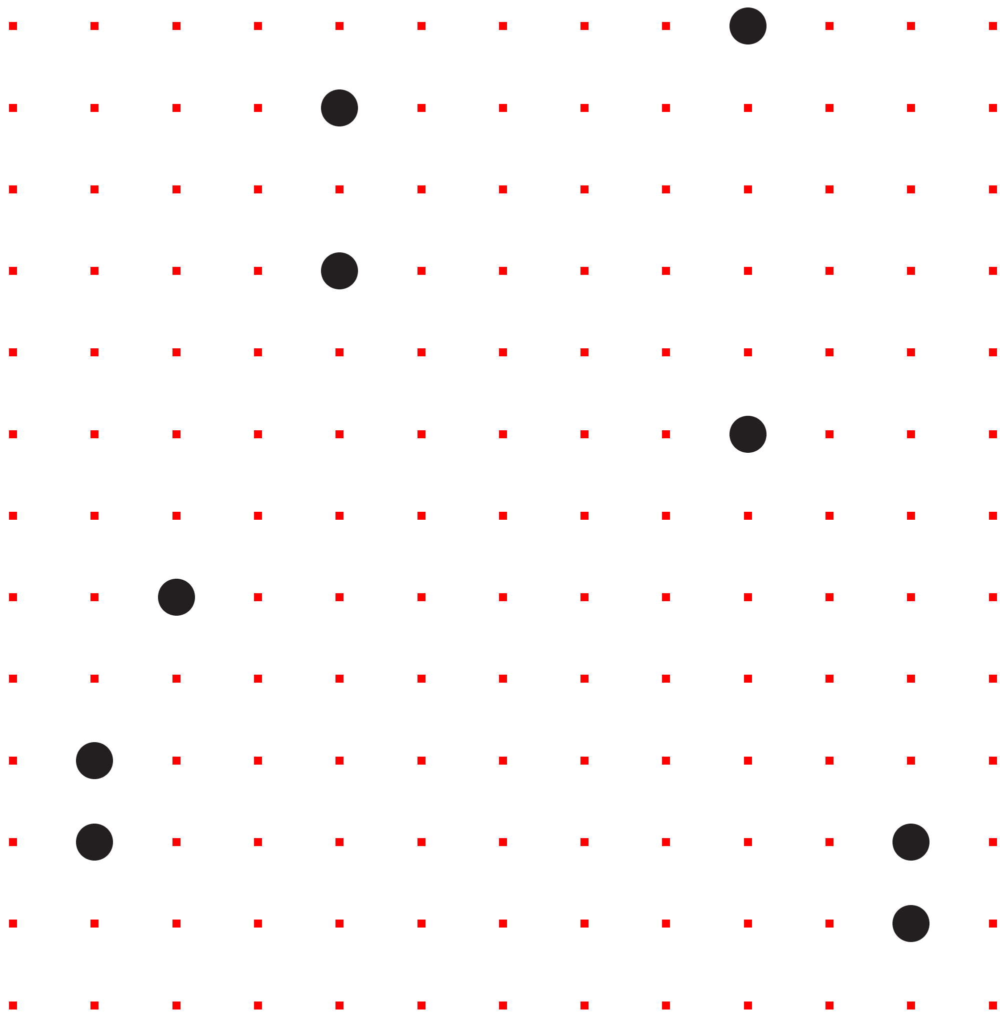
Similar example, an elliptic curve over a finite field:

Consider the prime field $\mathbf{Z}/13 = \{0, 1, 2, 3, 4, 5, \ldots, 12$ with $-, +, \cdot, /$ defined mod 13.

The "set of points on the elliptic curve $y^2 - 5xy = x^3 - 7$ over $\mathbf{Z}/13$" is

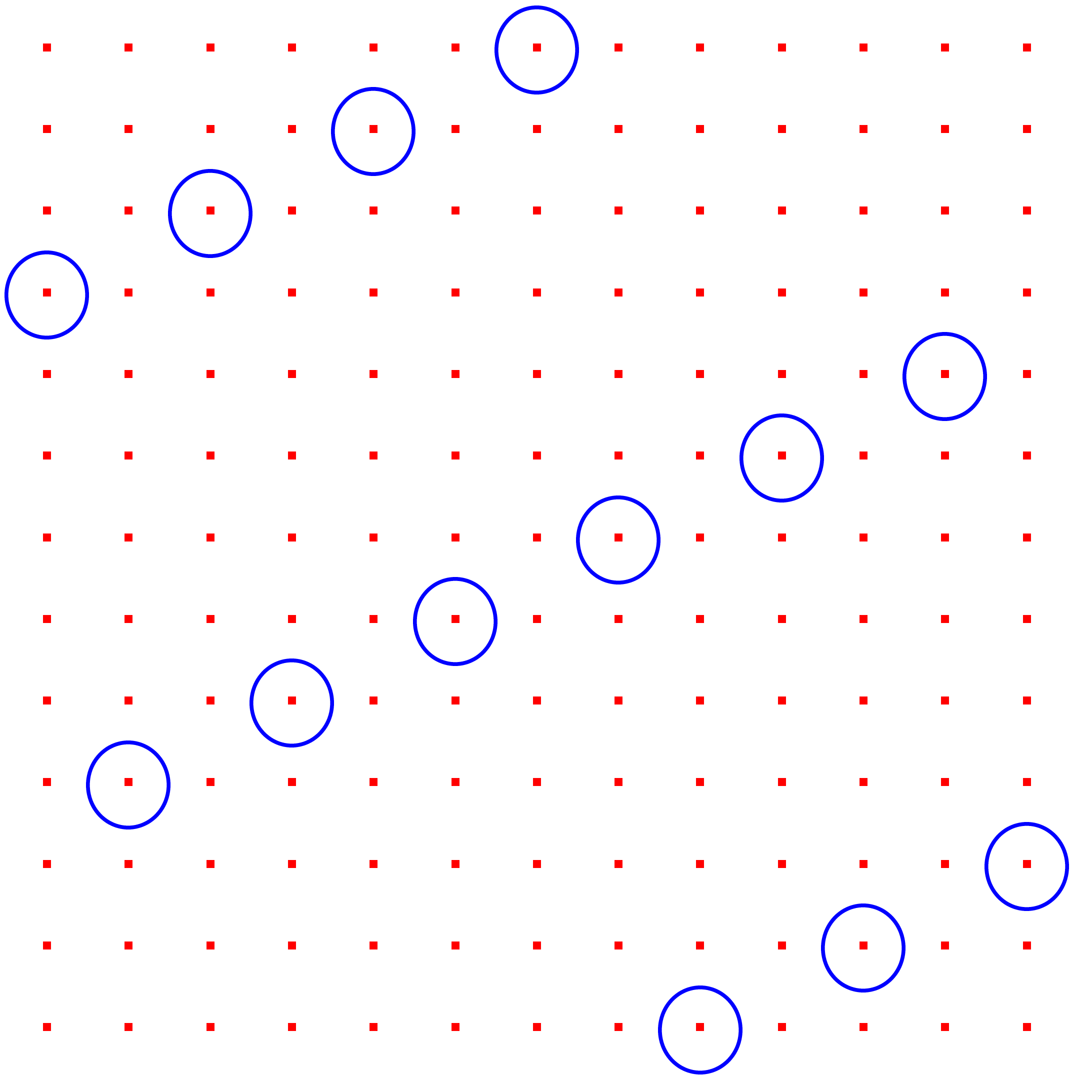
$$\{(x, y) \in (\mathbf{Z}/13)^2 :$$
$$y^2 - 5xy = x^3 - 7 \;\; \cup \{\infty \;\; .$$

Graph of this set of points:



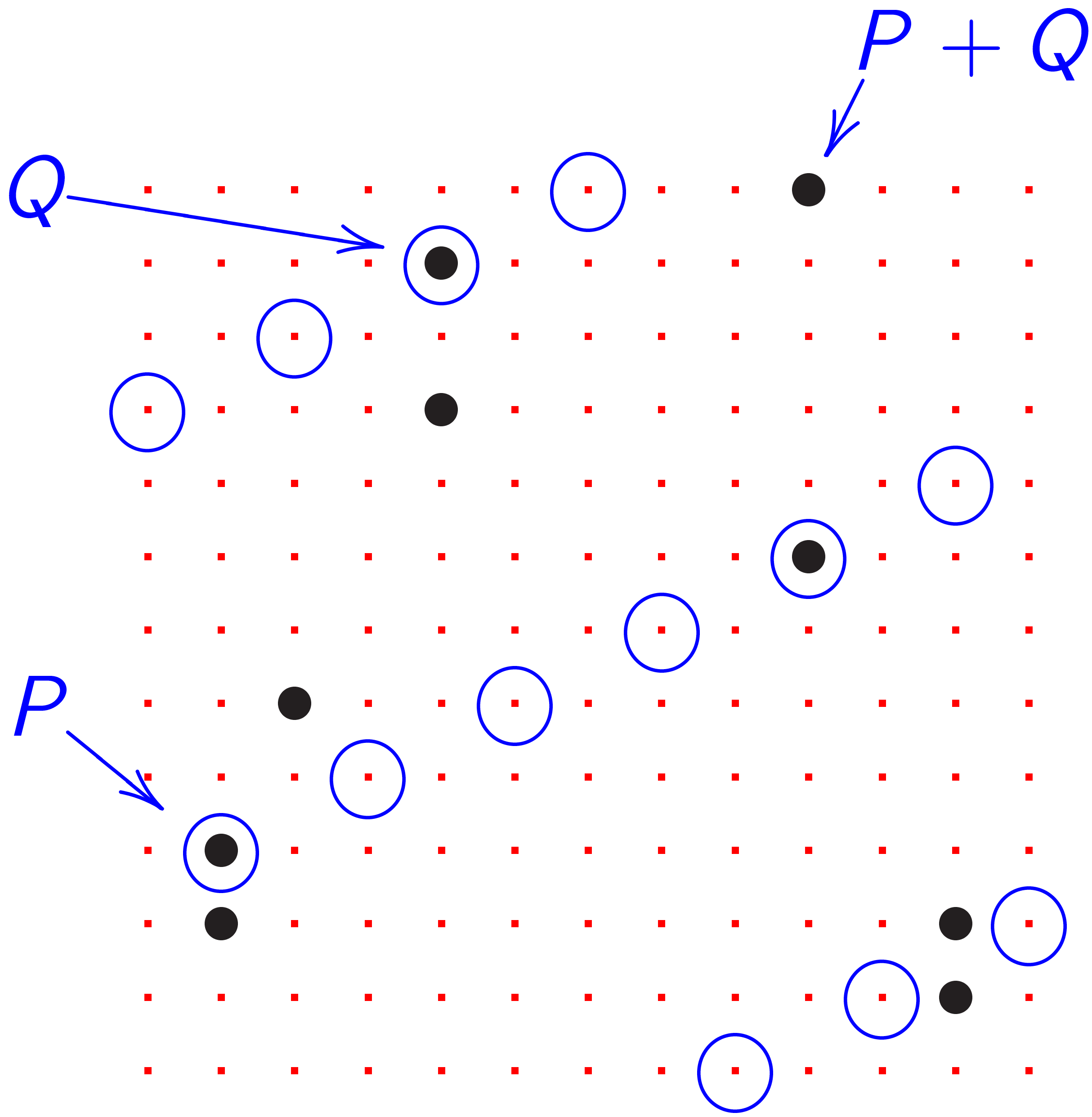
As before, don't forget ∞.

Example of line over $\mathbf{Z}/13$:



Formula for this line: $y = 7x + 9$.

Elliptic-curve addition law:



Complete definition of addition:

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

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

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

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

$\infty + \infty = \infty$.

Addition-law annoyances

1. First $(x, y) + (x', y')$ formula fails if $(x, y) = (x', y')$.
Must check, use second formula.
Can attacker see different timing?
Extra implementation work to avoid side-channel leaks.

2. More exceptional cases.
Can attacker trigger these?
Does implementation *always* follow the published protocol?

3. Tons of field arithmetic.
Is this fast enough?

Normally use fractions $X/Z, Y/Z$
(or $X/Z^2, Y/Z^3$: "Jacobian")
to avoid divisions, saving time.

But need many multiplications.
Can some be eliminated?

Some other elliptic-curve shapes
("Jacobi intersection,"
"Jacobi quartic," "Hessian")
try to unify doublings
with generic additions.

Still have exceptional cases.
Can exceptions be eliminated?

Interlude: Torus-based crypto

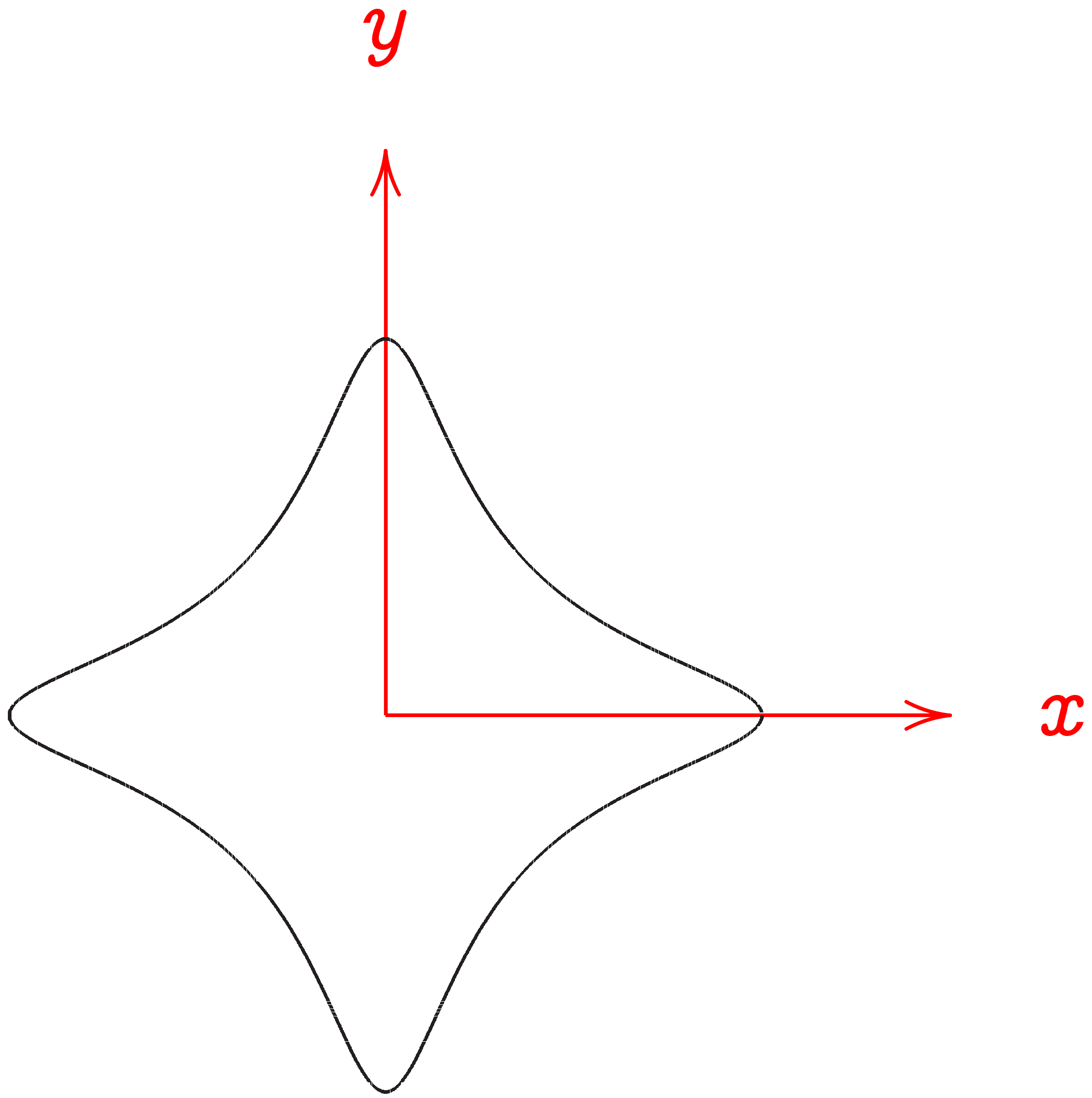
The circle
$$\{(x, y) \in (\mathbf{Z}/(2^{255} - 949))^2 :$$
$$x^2 + y^2 = 1\}$$
has a standard addition law:
$(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$
where $x_3 = x_1 y_2 + y_1 x_2$
and $y_3 = y_1 y_2 - x_1 x_2$.

Not many multiplications.
No exceptional cases.

But also not elliptic.
Broken by number-field sieve
unless field is replaced by
a much larger field.

News: Edwards curves

e.g. $x^2 + y^2 = 1 - 30x^2y^2$:



Choose a field K with $2 \neq 0$ and a parameter $d \in K - \{0, 1\}$.

Edwards addition law for $\{(x, y) \in K^2 :$
$x^2 + y^2 = 1 + dx^2y^2\}$ is

$$x_3 = \frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2},$$

$$y_3 = \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2}.$$

The Edwards addition law corresponds to the standard addition law on an elliptic curve.

If d is not a square
then the Edwards addition law
is **complete**:
no exceptional cases;
the denominators are never 0.

If $x_1^2 + y_1^2 = 1 + dx_1^2 y_1^2$
and $x_2^2 + y_2^2 = 1 + dx_2^2 y_2^2$
then $dx_1x_2y_1y_2$ can't be ± 1.

Outline of proof:
If $(dx_1x_2y_1y_2)^2 = 1$ then
$(x_1 + dx_1x_2y_1y_2y_1)^2 =$
$dx_1^2 y_1^2 (x_2 + y_2)^2$.
Conclude that d is a square.
But d is not a square! Q.E.D.

In particular,
choose $K = \mathbf{Z}/(2^{255} - 19)$
and $d = 121665/121666$.

K doesn't have $\sqrt{d}$,
so the Edwards addition law
for $x^2 + y^2 = 1 + dx^2y^2$
is complete.

This addition law corresponds to
the standard addition law
on Curve25519!
Easy map: $x = \sqrt{486664}u/v$,
$y = (u - 1)/(u + 1)$.

Can use the Edwards addition law
for Curve25519 computations.

Computations on Edwards curves

To avoid divisions, use
$(X : Y : Z)$ with $Z \neq 0$ and
$(X^2 + Y^2)Z^2 = Z^4 + dX^2Y^2$
to represent $(X/Z, Y/Z)$
on the Edwards curve
$x^2 + y^2 = 1 + dx^2y^2$.

Recall the Edwards addition law:

$$x_3 = \frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2},$$

$$y_3 = \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2}.$$

Clear denominators:

$$X_3 = Z_1 Z_2 (X_1 Y_2 + Y_1 X_2) \\ \cdot (Z_1^2 Z_2^2 - d X_1 X_2 Y_1 Y_2),$$
$$Y_3 = Z_1 Z_2 (Y_1 Y_2 - X_1 X_2) \\ \cdot (Z_1^2 Z_2^2 + d X_1 X_2 Y_1 Y_2),$$
$$Z_3 = (Z_1^2 Z_2^2 - d X_1 X_2 Y_1 Y_2) \\ \cdot (Z_1^2 Z_2^2 + d X_1 X_2 Y_1 Y_2).$$

Rewrite $x_1 y_2 + x_2 y_1$ as $(x_1 + y_1)(x_2 + y_2) - x_1 x_2 - y_1 y_2$, exploit common subexpressions.

12 multiplications (one by d, one a squaring), 7 additions. Still complete.

Comparison of addition costs

if curve parameters are small:

System	Cost
Doche/Icart/Kohel	$12\mathbf{M} + 5\mathbf{S}$
Jacobian	$11\mathbf{M} + 5\mathbf{S}$
Jacobi intersection	$13\mathbf{M} + 2\mathbf{S}$
Projective	$12\mathbf{M} + 2\mathbf{S}$
Jacobi quartic	$10\mathbf{M} + 3\mathbf{S}$
Hessian	$12\mathbf{M}$
Edwards	$10\mathbf{M} + 1\mathbf{S}$

Can save time

in "mixed additions" $(Z_2 = 1)$

and in "readditions";

slightly different order of systems.

Can save time in doubling:
rewrite $1 + dx_1^2 y_1^2$ as $x_1^2 + y_1^2$
(as suggested by Marc Joye);
rewrite $1 - dx_1^2 y_1^2$ as $2 - x_1^2 - y_1^2$;
exploit common subexpressions.

$B = (X_1 + Y_1)^2$, $C = X_1^2$, $D = Y_1^2$,
$E = C + D$, $H = Z_1^2$,
$J = E - 2H$, $X_3 = (B - E)J$,
$Y_3 = E(C - D)$, $Z_3 = EJ$.

7 multiplications
(4 of which are squarings),
6 additions.

Comparison of doubling costs if curve parameters are small:

System	Cost
Projective	$5\mathbf{M} + 6\mathbf{S}$
Projective if $a = -3$	$7\mathbf{M} + 3\mathbf{S}$
Hessian	$6\mathbf{M} + 3\mathbf{S}$
Jacobi quartic	$1\mathbf{M} + 9\mathbf{S}$
Jacobian	$1\mathbf{M} + 8\mathbf{S}$
Jacobian if $a = -3$	$3\mathbf{M} + 5\mathbf{S}$
Jacobi intersection	$3\mathbf{M} + 4\mathbf{S}$
Edwards	$3\mathbf{M} + 4\mathbf{S}$
Doche/Icart/Kohel	$2\mathbf{M} + 5\mathbf{S}$

Several new algorithms here.

Explicit-Formulas Database:

`http://www.hyperelliptic.org/EFD`

Consequences for signatures

Edwards coordinates vs. popular $a = -3$ Jacobian coordinates in standard cost model:

$\approx 5\%$ faster for $t \mapsto tB$ using typical B precomputation.

$\approx 15\%$ faster for $h, R \mapsto hR$.

$\approx 13\%$ faster for $t, h, R \mapsto tB - hR$ using "JSF."

$\approx 38\%$ faster for batch verification via Bos-Coster.

Plus: complete, low memory, . . .

Batch verification of many
$t_iB - h_iR_i - S_i = 0$:
choose random 128-bit v_i,
check $(\sum_i v_it_i)B - \sum_i(v_ih_i)R_i - \sum_i v_iS_i = 0$.
(Bellare/Garay/Rabin, LATIN '98)

Use subtractive multi-scalar multiplication algorithm (credited to Bos and Coster by de Rooij, EUROCRYPT '94).

Only $\approx$ 25.2 curve adds/bit to verify 100 signatures.

Use Edwards coordinates!

More on Edwards coordinates

Harold M. Edwards,
"A normal form
for elliptic curves,"
Bulletin of the AMS,
July 2007.

Daniel J. Bernstein
and Tanja Lange,
"Faster addition and doubling
on elliptic curves,"
Asiacrypt 2007.

`http://cr.yp.to`
`/newelliptic.html`