SafeCurves:
choosing safe curves for
elliptic-curve cryptography

Daniel J. Bernstein
University of Illinois at Chicago &
Technische Universiteit Eindhoven

Tanja Lange
Technische Universiteit Eindhoven

---

http://safecurves.cr.yp.to

Cryptography

Public-key signatures:
e.g., RSA, DSA, ECDSA.
Some uses: signed OS updates, SSL certificates, e-passports.

Public-key encryption:
e.g., RSA, DH, ECDH.
Some uses: SSL key exchange, locked iPhone mail download.

Secret-key encryption:
e.g., AES, Salsa20.
Some uses: disk encryption, bulk SSL encryption.

Why ECC?

"Index calculus":
fastest method we know
to break original DH and RSA.

Long history, including
many major improvements:
1975, CFRAC;
1977, linear sieve (LS);
1982, quadratic sieve (QS);
1990, number-field sieve (NFS);
1994, function-field sieve (FFS);
2006, medium-prime FFS/NFS;
2013, $x^q - x$ FFS.

(FFS is not relevant to RSA.)

Also many smaller improvements:
$\approx$ 100 scientific papers.

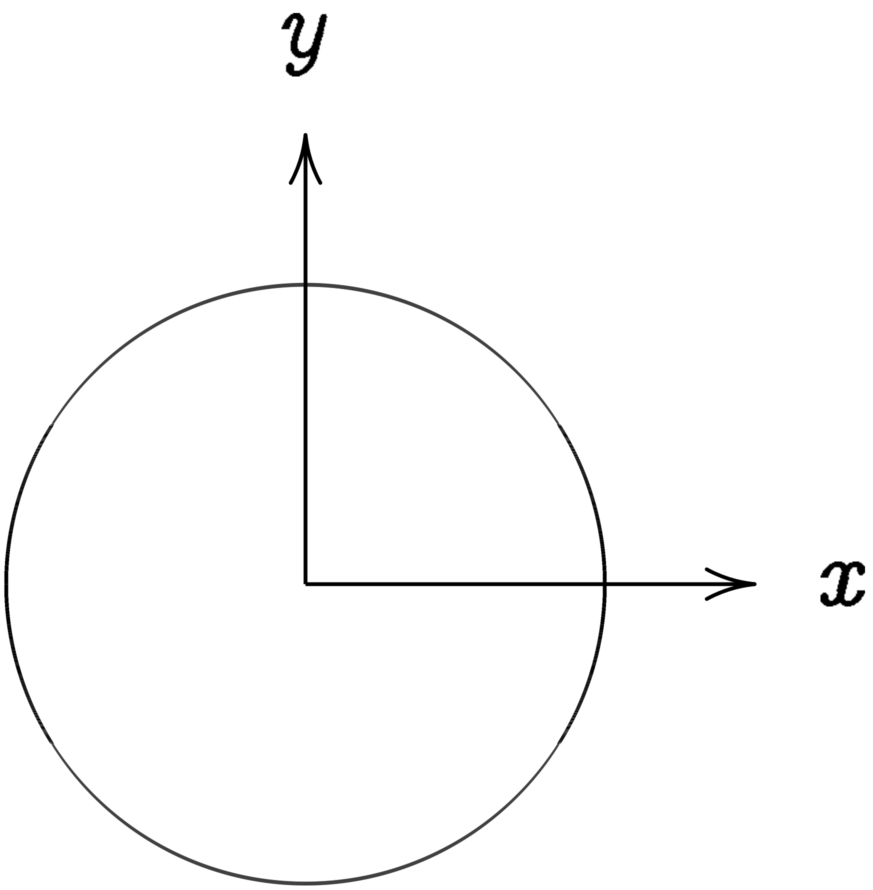
Costs of these algorithms for
breaking RSA-1024, RSA-2048:
$\approx 2^{120}$, 2^{170}, CFRAC;
$\approx 2^{110}$, 2^{160}, LS;
$\approx 2^{100}$, 2^{150}, QS;
$\approx 2^{80}$, 2^{112}, NFS.

Also many smaller improvements:
$\approx$ 100 scientific papers.

Costs of these algorithms for breaking RSA-1024, RSA-2048:
$\approx 2^{120}$, 2^{170}, CFRAC;
$\approx 2^{110}$, 2^{160}, LS;
$\approx 2^{100}$, 2^{150}, QS;
$\approx 2^{80}$, 2^{112}, NFS.

1986 Miller "Use of elliptic curves in cryptography": "It is extremely unlikely that an 'index calculus' attack on the elliptic curve method will ever be able to work."

The clock



This is the curve $x^2 + y^2 = 1$.

Warning:

This is *not* an elliptic curve.

"Elliptic curve" $\neq$ "ellipse."

Examples of points on this curve:

Examples of points on this curve:

$(0, 1) =$ “12:00”.

Examples of points on this curve:

$(0, 1) =$ "12:00".

$(0, -1) =$ "6:00".

Examples of points on this curve:

$(0, 1) =$ "12:00".

$(0, -1) =$ "6:00".

$(1, 0) =$ "3:00".

Examples of points on this curve:

$(0, 1) =$ "12:00".

$(0, -1) =$ "6:00".

$(1, 0) =$ "3:00".

$(-1, 0) =$ "9:00".

Examples of points on this curve:

$(0, 1) =$ "12:00".

$(0, -1) =$ "6:00".

$(1, 0) =$ "3:00".

$(-1, 0) =$ "9:00".

$(\sqrt{3/4}, 1/2) =$

Examples of points on this curve:

$(0, 1) =$ "12:00".

$(0, -1) =$ "6:00".

$(1, 0) =$ "3:00".

$(-1, 0) =$ "9:00".

$(\sqrt{3/4}, 1/2) =$ "2:00".

Examples of points on this curve:

$(0, 1) =$ "12:00".

$(0, -1) =$ "6:00".

$(1, 0) =$ "3:00".

$(-1, 0) =$ "9:00".

$(\sqrt{3/4}, 1/2) =$ "2:00".

$(1/2, -\sqrt{3/4}) =$

Examples of points on this curve:

$(0, 1) =$ "12:00".

$(0, -1) =$ "6:00".

$(1, 0) =$ "3:00".

$(-1, 0) =$ "9:00".

$(\sqrt{3/4}, 1/2) =$ "2:00".

$(1/2, -\sqrt{3/4}) =$ "5:00".

$(-1/2, -\sqrt{3/4}) =$

Examples of points on this curve:

$(0, 1) =$ "12:00".

$(0, -1) =$ "6:00".

$(1, 0) =$ "3:00".

$(-1, 0) =$ "9:00".

$(\sqrt{3/4}, 1/2) =$ "2:00".

$(1/2, -\sqrt{3/4}) =$ "5:00".

$(-1/2, -\sqrt{3/4}) =$ "7:00".

Examples of points on this curve:

$(0, 1) =$ “12:00”.

$(0, -1) =$ “6:00”.

$(1, 0) =$ “3:00”.

$(-1, 0) =$ “9:00”.

$(\sqrt{3/4}, 1/2) =$ “2:00”.

$(1/2, -\sqrt{3/4}) =$ “5:00”.

$(-1/2, -\sqrt{3/4}) =$ “7:00”.

$(\sqrt{1/2}, \sqrt{1/2}) =$ “1:30”.

$(3/5, 4/5)$. $(-3/5, 4/5)$.

Examples of points on this curve:

$(0, 1) =$ "12:00".

$(0, -1) =$ "6:00".

$(1, 0) =$ "3:00".

$(-1, 0) =$ "9:00".

$(\sqrt{3/4}, 1/2) =$ "2:00".

$(1/2, -\sqrt{3/4}) =$ "5:00".

$(-1/2, -\sqrt{3/4}) =$ "7:00".

$(\sqrt{1/2}, \sqrt{1/2}) =$ "1:30".

$(3/5, 4/5)$. $(-3/5, 4/5)$.

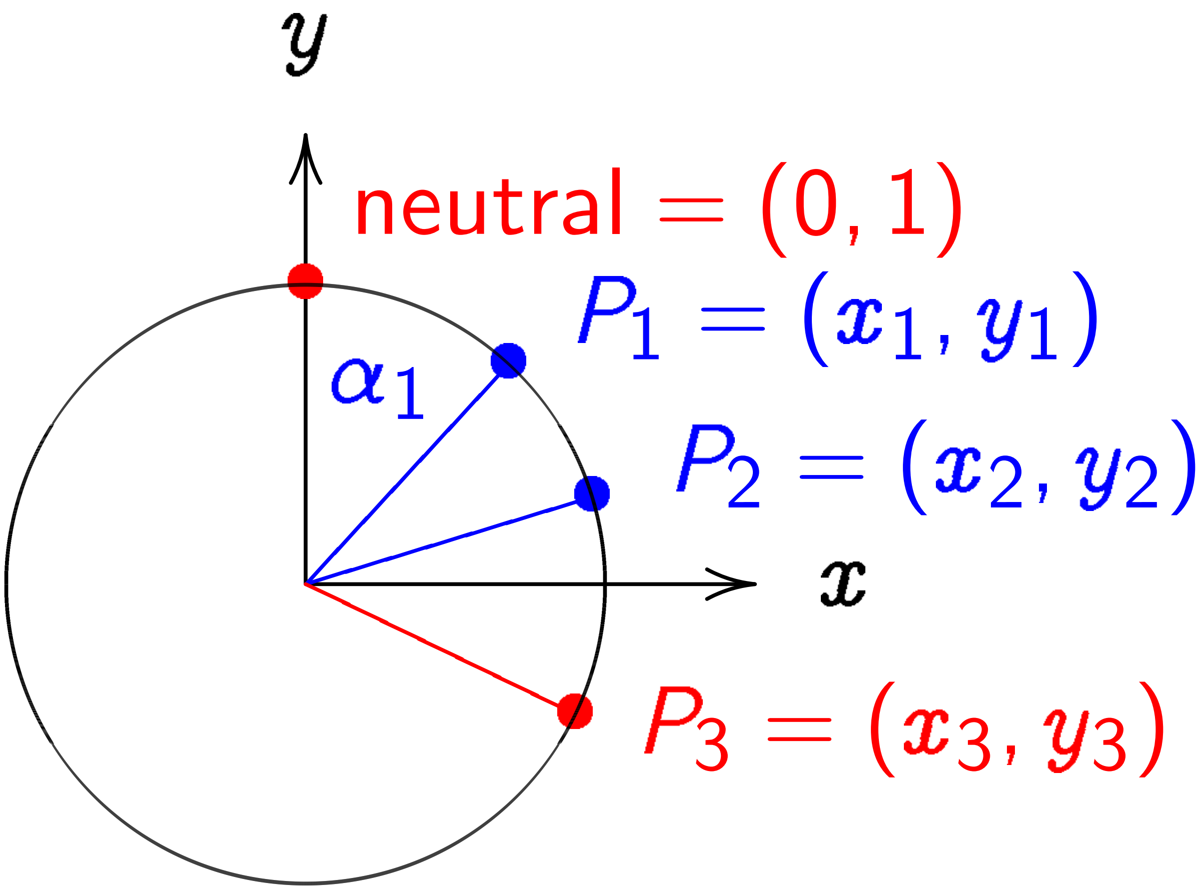
$(3/5, -4/5)$. $(-3/5, -4/5)$.

$(4/5, 3/5)$. $(-4/5, 3/5)$.

$(4/5, -3/5)$. $(-4/5, -3/5)$.

Many more.

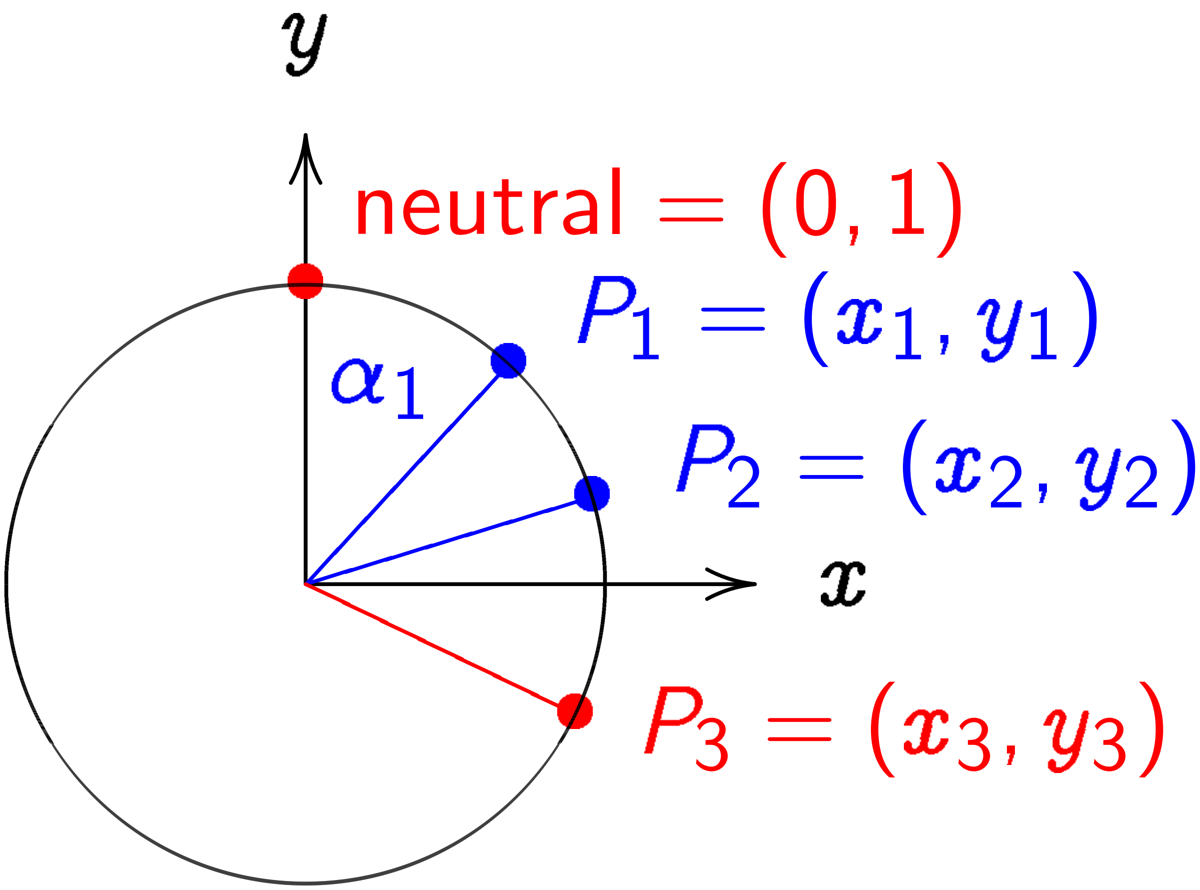
Addition on the clock:



$x^2 + y^2 = 1$, parametrized by

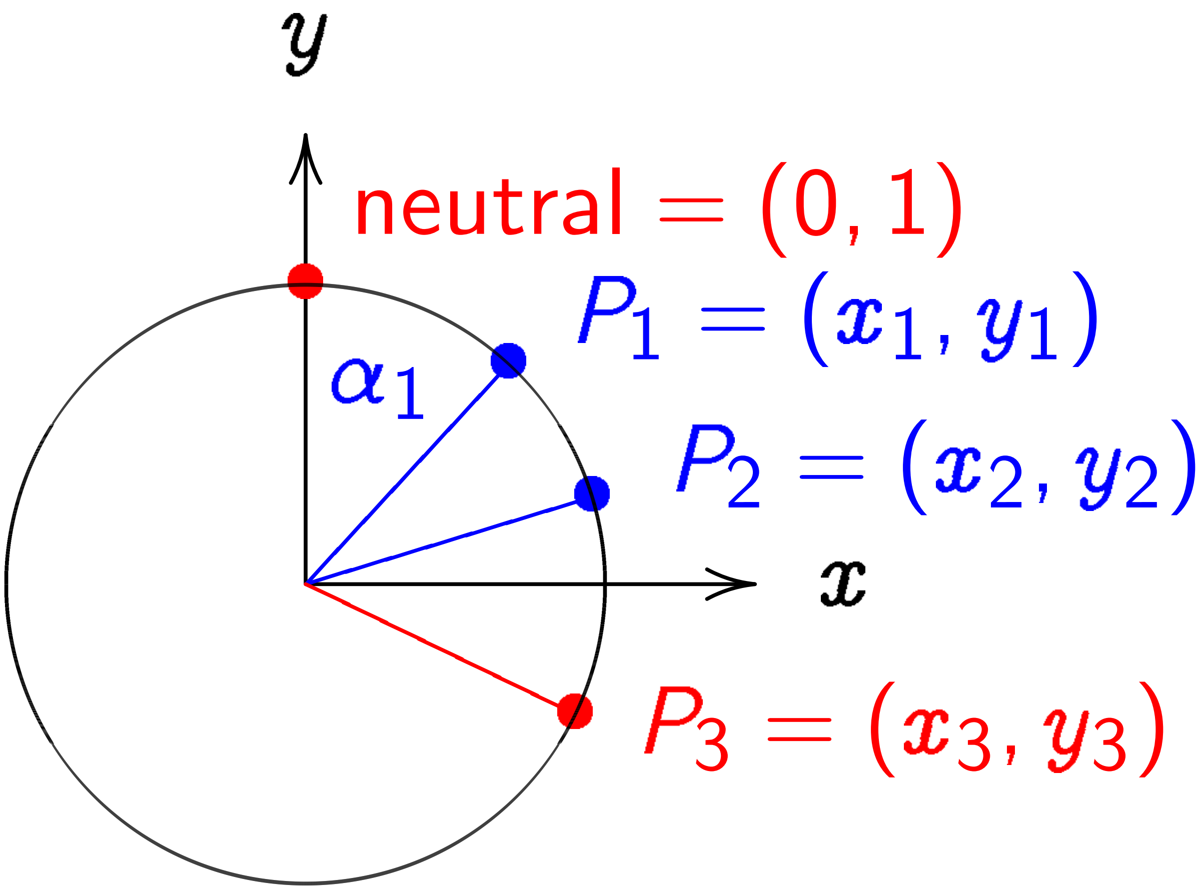
$x = \sin\alpha$, $y = \cos\alpha$.

Addition on the clock:



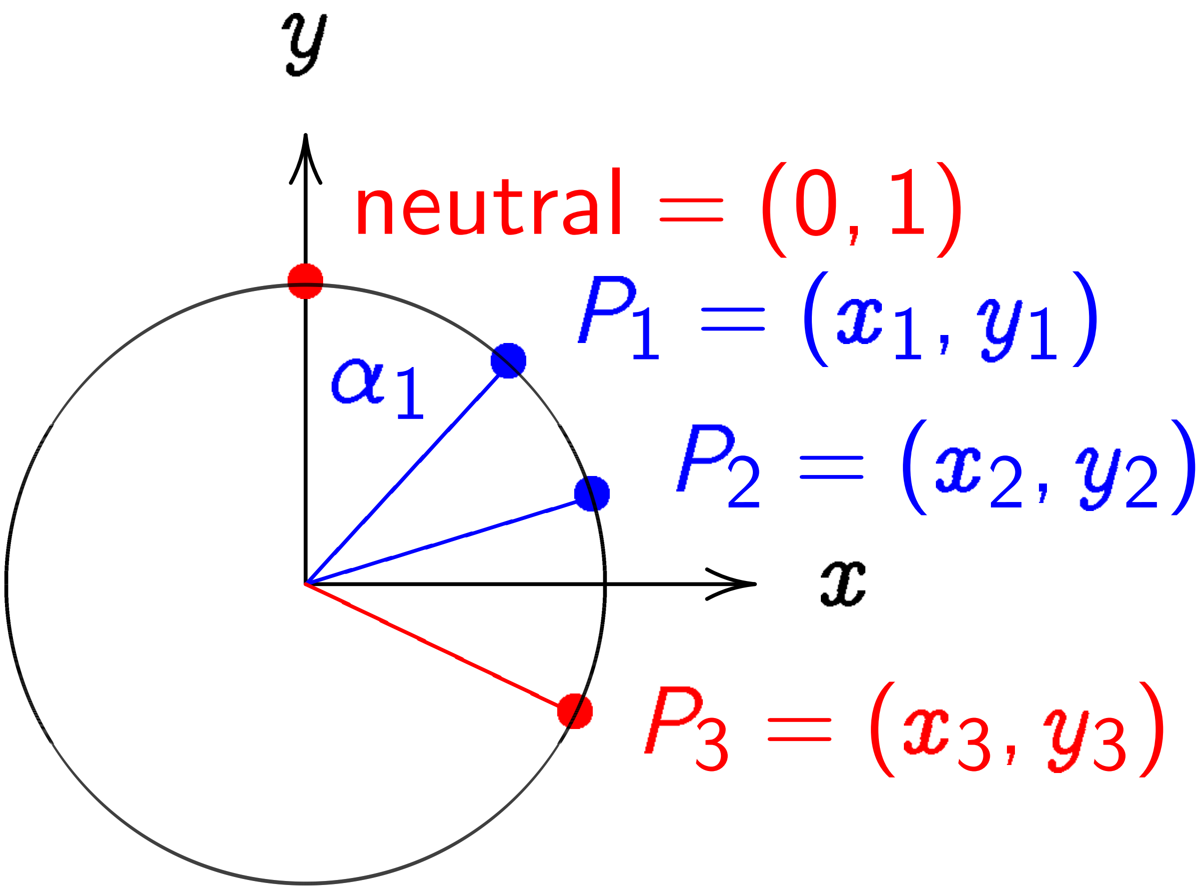
$x^2 + y^2 = 1$, parametrized by
$x = \sin\alpha$, $y = \cos\alpha$. Recall
$(\sin(\alpha_1 + \alpha_2), \cos(\alpha_1 + \alpha_2)) =$

Addition on the clock:



$x^2 + y^2 = 1$, parametrized by
$x = \sin \alpha$, $y = \cos \alpha$. Recall
$(\sin(\alpha_1 + \alpha_2), \cos(\alpha_1 + \alpha_2)) =$
$(\sin \alpha_1 \cos \alpha_2 + \cos \alpha_1 \sin \alpha_2,$

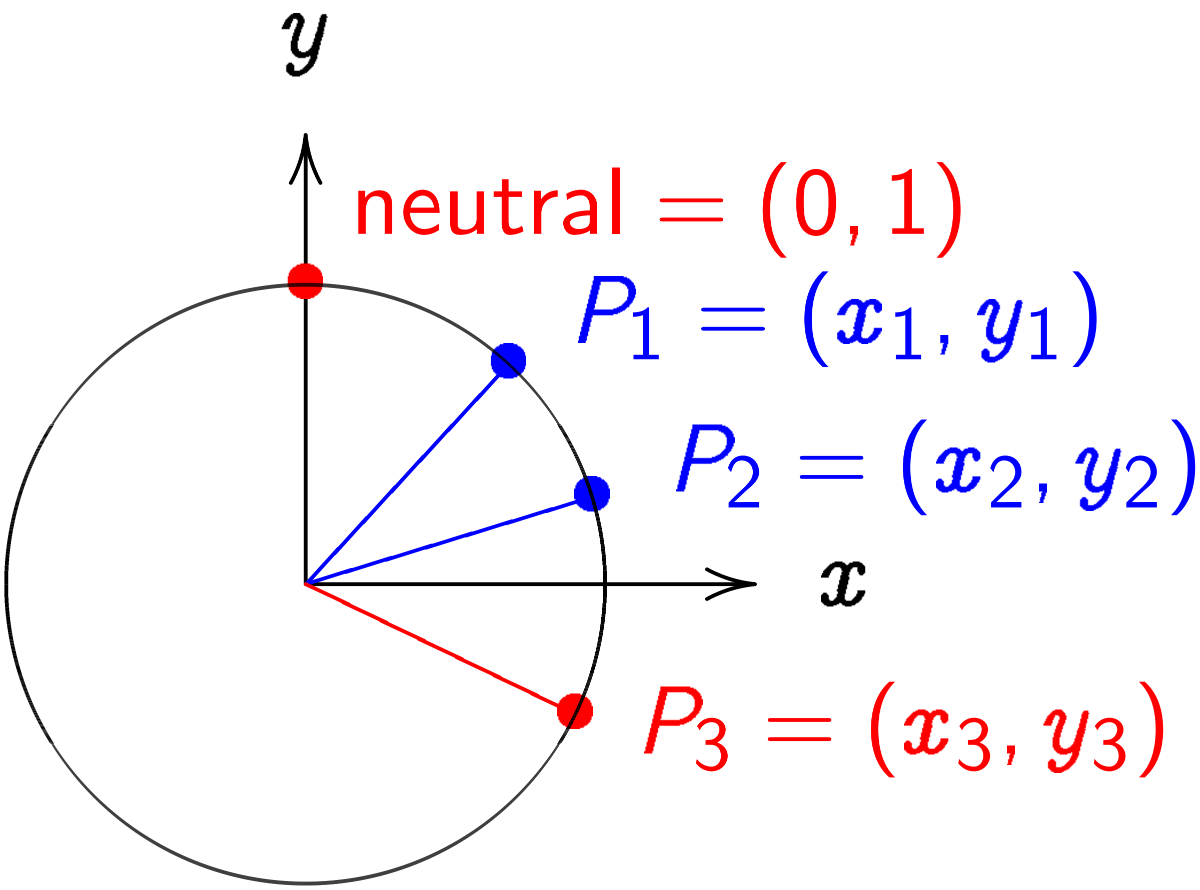
Addition on the clock:



$x^2 + y^2 = 1$, parametrized by $x = \sin\alpha$, $y = \cos\alpha$. Recall

$$(\sin(\alpha_1 + \alpha_2), \cos(\alpha_1 + \alpha_2)) =$$
$$(\sin\alpha_1 \cos\alpha_2 + \cos\alpha_1 \sin\alpha_2,$$
$$\cos\alpha_1 \cos\alpha_2 - \sin\alpha_1 \sin\alpha_2).$$

Clock addition without sin, cos:



Use Cartesian coordinates for addition. Addition formula for the clock $x^2 + y^2 = 1$: sum of (x_1, y_1) and (x_2, y_2) is $(x_1 y_2 + y_1 x_2, y_1 y_2 - x_1 x_2)$.

Examples of clock addition:

$$\text{“2:00”} + \text{“5:00”}$$
$$= (\sqrt{3/4}, 1/2) + (1/2, -\sqrt{3/4})$$
$$= (-1/2, -\sqrt{3/4}) = \text{“7:00”}.$$

$$\text{“5:00”} + \text{“9:00”}$$
$$= (1/2, -\sqrt{3/4}) + (-1, 0)$$
$$= (\sqrt{3/4}, 1/2) = \text{“2:00”}.$$

$$2\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{24}{25}, \frac{7}{25}\right).$$

Examples of clock addition:

$$\text{“2:00”} + \text{“5:00”}$$
$$= (\sqrt{3/4}, 1/2) + (1/2, -\sqrt{3/4})$$
$$= (-1/2, -\sqrt{3/4}) = \text{“7:00”}.$$

$$\text{“5:00”} + \text{“9:00”}$$
$$= (1/2, -\sqrt{3/4}) + (-1, 0)$$
$$= (\sqrt{3/4}, 1/2) = \text{“2:00”}.$$

$$2\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{24}{25}, \frac{7}{25}\right).$$

$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

Examples of clock addition:

$$\text{“2:00”} + \text{“5:00”}$$
$$= (\sqrt{3/4}, 1/2) + (1/2, -\sqrt{3/4})$$
$$= (-1/2, -\sqrt{3/4}) = \text{“7:00”}.$$

$$\text{“5:00”} + \text{“9:00”}$$
$$= (1/2, -\sqrt{3/4}) + (-1, 0)$$
$$= (\sqrt{3/4}, 1/2) = \text{“2:00”}.$$

$$2\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{24}{25}, \frac{7}{25}\right).$$

$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

$$4\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{336}{625}, \frac{-527}{625}\right).$$

Examples of clock addition:

$$\text{“2:00”} + \text{“5:00”}$$
$$= (\sqrt{3/4}, 1/2) + (1/2, -\sqrt{3/4})$$
$$= (-1/2, -\sqrt{3/4}) = \text{“7:00”}.$$

$$\text{“5:00”} + \text{“9:00”}$$
$$= (1/2, -\sqrt{3/4}) + (-1, 0)$$
$$= (\sqrt{3/4}, 1/2) = \text{“2:00”}.$$

$$2\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{24}{25}, \frac{7}{25}\right).$$

$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

$$4\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{336}{625}, \frac{-527}{625}\right).$$

$$(x_1, y_1) + (0, 1) =$$

Examples of clock addition:

"2:00" + "5:00"

$$= (\sqrt{3/4}, 1/2) + (1/2, -\sqrt{3/4})$$
$$= (-1/2, -\sqrt{3/4}) = \text{“7:00”}.$$

"5:00" + "9:00"

$$= (1/2, -\sqrt{3/4}) + (-1, 0)$$
$$= (\sqrt{3/4}, 1/2) = \text{“2:00”}.$$

$$2\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{24}{25}, \frac{7}{25}\right).$$

$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

$$4\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{336}{625}, \frac{-527}{625}\right).$$

$$(x_1, y_1) + (0, 1) = (x_1, y_1).$$

Examples of clock addition:

"2:00" + "5:00"

$= (\sqrt{3/4}, 1/2) + (1/2, -\sqrt{3/4})$

$= (-1/2, -\sqrt{3/4}) =$ "7:00".

"5:00" + "9:00"

$= (1/2, -\sqrt{3/4}) + (-1, 0)$

$= (\sqrt{3/4}, 1/2) =$ "2:00".

$$2\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{24}{25}, \frac{7}{25}\right).$$

$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

$$4\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{336}{625}, \frac{-527}{625}\right).$$

$(x_1, y_1) + (0, 1) = (x_1, y_1)$.

$(x_1, y_1) + (-x_1, y_1) =$

Examples of clock addition:

"2:00" + "5:00"

$$= (\sqrt{3/4}, 1/2) + (1/2, -\sqrt{3/4})$$
$$= (-1/2, -\sqrt{3/4}) = \text{"7:00"}.$$

"5:00" + "9:00"

$$= (1/2, -\sqrt{3/4}) + (-1, 0)$$
$$= (\sqrt{3/4}, 1/2) = \text{"2:00"}.$$

$$2\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{24}{25}, \frac{7}{25}\right).$$

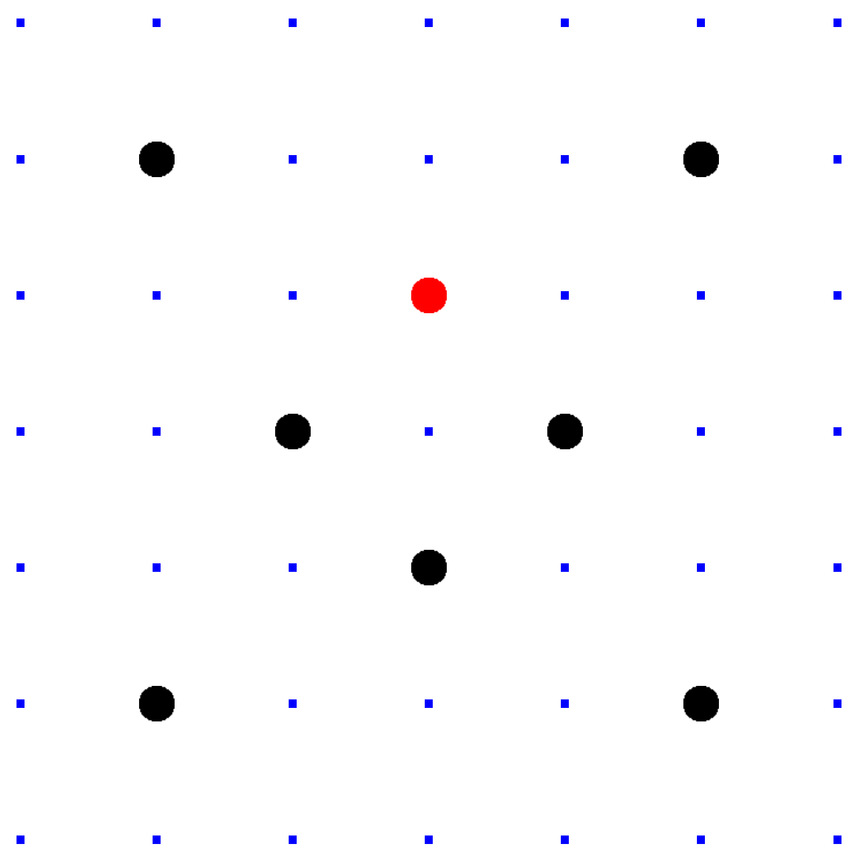
$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

$$4\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{336}{625}, \frac{-527}{625}\right).$$

$$(x_1, y_1) + (0, 1) = (x_1, y_1).$$

$$(x_1, y_1) + (-x_1, y_1) = (0, 1).$$

Clocks over finite fields



$\text{Clock}(\mathbf{F}_7) =$
$\{(x, y) \in \mathbf{F}_7 \times \mathbf{F}_7 : x^2 + y^2 = 1\}$.
Here $\mathbf{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$
$= \{0, 1, 2, 3, -3, -2, -1\}$
with arithmetic modulo 7.
e.g. $2 \cdot 5 = 3$ and $3/2 = 5$ in $\mathbf{F}_7$.

Larger example: Clock($\mathbf{F}_{1000003}$).

Examples of addition
on Clock($\mathbf{F}_{1000003}$):
$2(1000, 2) = (4000, 7)$.

Larger example: Clock($\mathbf{F}_{1000003}$).

Examples of addition
on Clock($\mathbf{F}_{1000003}$):
$2(1000, 2) = (4000, 7)$.
$4(1000, 2) = (56000, 97)$.

Larger example: Clock($\mathbf{F}_{1000003}$).

Examples of addition
on Clock($\mathbf{F}_{1000003}$):
$2(1000, 2) = (4000, 7)$.
$4(1000, 2) = (56000, 97)$.
$8(1000, 2) = (863970, 18817)$.

Larger example: Clock($\mathbf{F}_{1000003}$).

Examples of addition
on Clock($\mathbf{F}_{1000003}$):

$2(1000, 2) = (4000, 7)$.

$4(1000, 2) = (56000, 97)$.

$8(1000, 2) = (863970, 18817)$.

$16(1000, 2) = (549438, 156853)$.

Larger example: Clock($\mathbf{F}_{1000003}$).

Examples of addition
on Clock($\mathbf{F}_{1000003}$):

$2(1000, 2) = (4000, 7)$.

$4(1000, 2) = (56000, 97)$.

$8(1000, 2) = (863970, 18817)$.

$16(1000, 2) = (549438, 156853)$.

$17(1000, 2) = (951405, 877356)$.

Larger example: $\text{Clock}(\mathbf{F}_{1000003})$.

Examples of addition on $\text{Clock}(\mathbf{F}_{1000003})$:

$2(1000, 2) = (4000, 7)$.

$4(1000, 2) = (56000, 97)$.

$8(1000, 2) = (863970, 18817)$.

$16(1000, 2) = (549438, 156853)$.

$17(1000, 2) = (951405, 877356)$.

"Scalar multiplication" on a clock:
Given integer $n \geq 0$ and clock point (x, y), compute $n(x, y)$.

"Binary method":
If n is even, compute $n(x, y)$
by doubling $(n/2)(x, y)$.
Otherwise compute $n(x, y)$
by adding (x, y) to $(n - 1)(x, y)$.
This is very fast.

"Binary method":

If n is even, compute $n(x, y)$
by doubling $(n/2)(x, y)$.
Otherwise compute $n(x, y)$
by adding (x, y) to $(n - 1)(x, y)$.
This is very fast.

But figuring out n
given (x, y) and $n(x, y)$
is much more difficult.

With 30 clock additions
we computed
$n(1000, 2) = (947472, 736284)$
for some 6-digit n.
Can you figure out n?

Clock cryptography

Standardize a large prime p
and some $(x, y) \in \text{Clock}(\mathbf{F}_p)$.

Alice chooses big secret a.
Computes her public key $a(x, y)$.

Bob chooses big secret b.
Computes his public key $b(x, y)$.

Alice computes $a(b(x, y))$.
Bob computes $b(a(x, y))$.
They use this shared secret
to encrypt with AES-GCM etc.

Warning #1:
Many choices of p are bad!

$$
\begin{tikzcd}
\begin{array}{c}\text{Alice's}\\ \text{secret key } a\end{array} \arrow[d] \arrow[dd, bend right] & & \begin{array}{c}\text{Bob's}\\ \text{secret key } b\end{array} \arrow[d] \arrow[dd, bend left] \\
\begin{array}{c}\text{Alice's}\\ \text{public key}\\ a(x,y)\end{array} \arrow[drr] & & \begin{array}{c}\text{Bob's}\\ \text{public key}\\ b(x,y)\end{array} \arrow[dll] \\
\begin{array}{c}\{\text{Alice, Bob}\}\text{'s}\\ \text{shared secret}\\ ab(x,y)\end{array} & = & \begin{array}{c}\{\text{Bob, Alice}\}\text{'s}\\ \text{shared secret}\\ ba(x,y)\end{array}
\end{tikzcd}
$$

$$
\begin{tikzcd}
\begin{array}{c}\text{Alice's}\\ \text{secret key } a\end{array} \arrow[d] \arrow[dd, bend right] & & \begin{array}{c}\text{Bob's}\\ \text{secret key } b\end{array} \arrow[d] \arrow[dd, bend left] \\
\begin{array}{c}\text{Alice's}\\ \text{public key}\\ a(x,y)\end{array} \arrow[drr] & & \begin{array}{c}\text{Bob's}\\ \text{public key}\\ b(x,y)\end{array} \arrow[dll] \\
\begin{array}{c}\{\text{Alice, Bob}\}\text{'s}\\ \text{shared secret}\\ ab(x,y)\end{array} & = & \begin{array}{c}\{\text{Bob, Alice}\}\text{'s}\\ \text{shared secret}\\ ba(x,y)\end{array}
\end{tikzcd}
$$

Warning #2:

Clocks aren't elliptic!

Can use index calculus

to attack clock cryptography.

To match RSA-3072 security

need $p \approx 2^{1536}$.

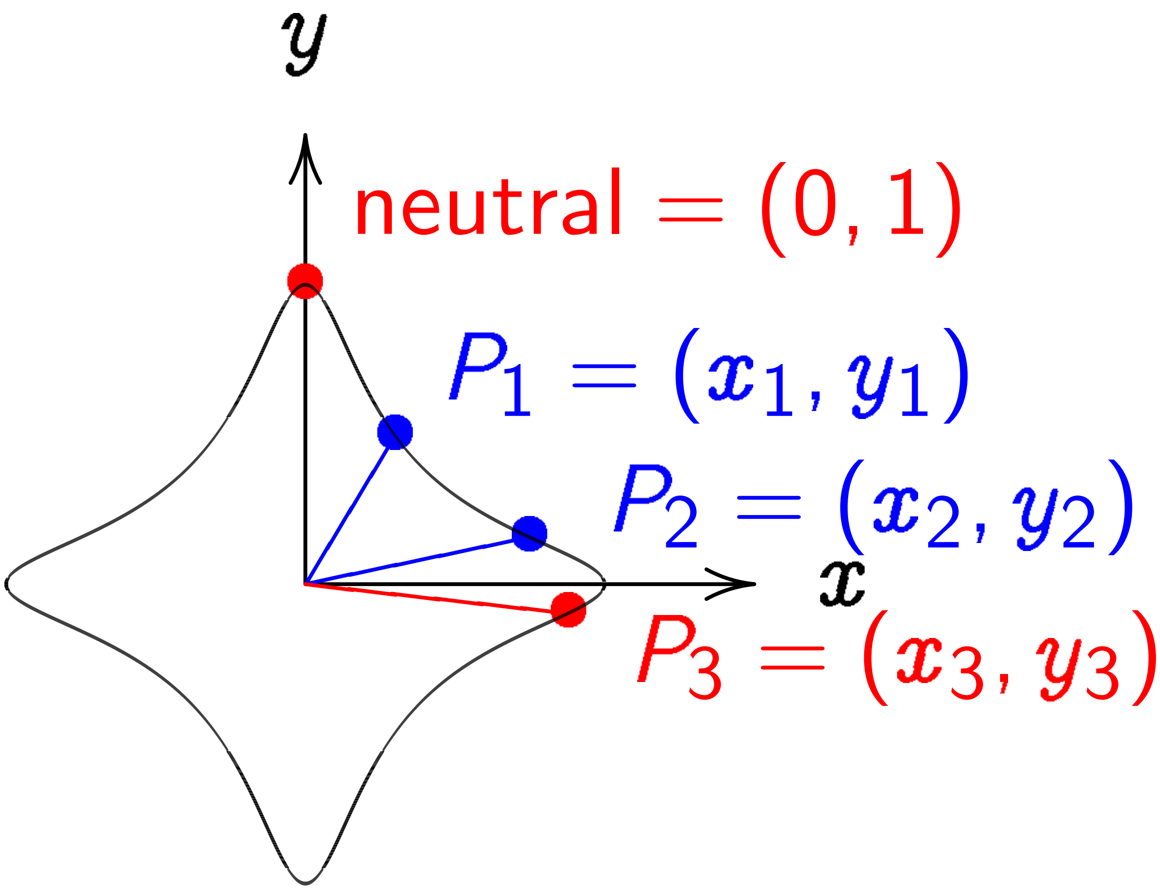
Timing attacks

Attacker sees more than $a(x, y)$ and $b(x, y)$.

Attacker sees *time* for Alice to compute $a(b(x, y))$. Often attacker can see time for *each operation* performed by Alice, not just total time. This reveals secret a.

Fix: **constant-time** code, performing same operations no matter what scalar is.

Addition on an elliptic curve

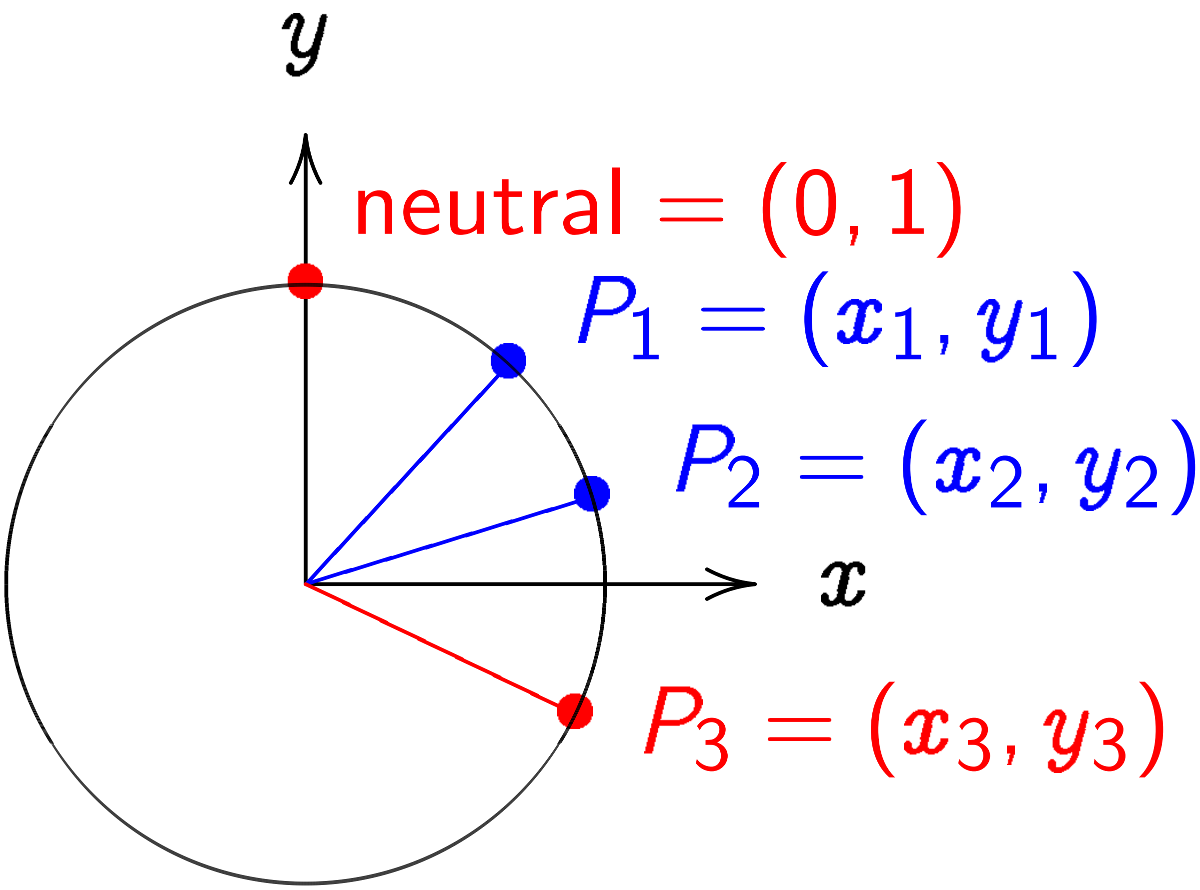


$x^2 + y^2 = 1 - 30x^2y^2$.

Sum of (x_1, y_1) and (x_2, y_2) is

$((x_1y_2+y_1x_2)/(1-30x_1x_2y_1y_2),$
$(y_1y_2-x_1x_2)/(1+30x_1x_2y_1y_2))$.

The clock again, for comparison:



$x^2 + y^2 = 1$.

Sum of (x_1, y_1) and (x_2, y_2) is

$(x_1 y_2 + y_1 x_2,$

$y_1 y_2 - x_1 x_2)$.

More elliptic curves

Choose an odd prime p.
Choose a *non-square* $d \in \mathbf{F}_p$.

$$\{(x, y) \in \mathbf{F}_p \times \mathbf{F}_p : \\ x^2 + y^2 = 1 + dx^2y^2\}$$

is a "complete Edwards curve".

"The Edwards addition law":
$(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$
where

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

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

“Hey, there are divisions
in the Edwards addition law!
What if the denominators are 0?”

“Hey, there are divisions
in the Edwards addition law!
What if the denominators are 0?”

Answer: Can prove that
the denominators are never 0.
Addition law is **complete**.

“Hey, there are divisions
in the Edwards addition law!
What if the denominators are 0?”

Answer: Can prove that
the denominators are never 0.
Addition law is **complete**.

This proof relies on
choosing *non-square* d.

"Hey, there are divisions in the Edwards addition law! What if the denominators are 0?"

Answer: Can prove that the denominators are never 0. Addition law is **complete**.

This proof relies on choosing *non-square* d.

If we instead choose square d: curve is still elliptic, and addition *seems to work,* but there are failure cases, often exploitable by attackers. Safe code is more complicated.

A safe example

Choose $p = 2^{255} - 19$.
Choose $d = 121665/121666$;
this is non-square in $\mathbf{F}_p$.

$x^2 + y^2 = 1 + dx^2y^2$
is a safe curve for ECC.

A safe example

Choose $p = 2^{255} - 19$.
Choose $d = 121665/121666$;
this is non-square in $\mathbf{F}_p$.

$x^2 + y^2 = 1 + dx^2y^2$
is a safe curve for ECC.

$-x^2 + y^2 = 1 - dx^2y^2$
is another safe curve
using the same p and d.

A safe example

Choose $p = 2^{255} - 19$.
Choose $d = 121665/121666$;
this is non-square in $\mathbf{F}_p$.

$x^2 + y^2 = 1 + dx^2y^2$
is a safe curve for ECC.

$-x^2 + y^2 = 1 - dx^2y^2$
is another safe curve
using the same p and d.

Actually, the second curve
is the first curve in disguise:
replace x in first curve
by $\sqrt{-1} \cdot x$, using $\sqrt{-1} \in \mathbf{F}_p$.

Even more elliptic curves

Edwards curves:
$x^2 + y^2 = 1 + dx^2y^2$.

Twisted Edwards curves:
$ax^2 + y^2 = 1 + dx^2y^2$.

Weierstrass curves:
$v^2 = u^3 + au + b$.

Montgomery curves:
$bv^2 = u^3 + au^2 + u$.

Many relationships:
e.g., substitute $x = u/v$,
$y = (u-1)/(u+1)$ in Edwards
to obtain Montgomery.

Addition on Weierstrass curves

$v^2 = u^3 + au + b$:

Addition on Weierstrass curves

$v^2 = u^3 + au + b$:

for $u_1 \neq u_2$, $(u_1, v_1) + (u_2, v_2) = (u_3, v_3)$ with $u_3 = \lambda^2 - u_1 - u_2$, $v_3 = \lambda(u_1 - u_3) - v_1$, $\lambda = (v_2 - v_1)/(u_2 - u_1)$; for $v_1 \neq 0$, $(u_1, v_1) + (u_1, v_1) = (u_3, v_3)$ with $u_3 = \lambda^2 - u_1 - u_2$, $v_3 = \lambda(u_1 - u_3) - v_1$, $\lambda = (3u_1^2 + a)/2v_1$;

$(u_1, v_1) + (u_1, -v_1) = \infty$;

$(u_1, v_1) + \infty = (u_1, v_1)$;

$\infty + (u_2, v_2) = (u_2, v_2)$;

$\infty + \infty = \infty$.

Addition on Weierstrass curves

$v^2 = u^3 + au + b$:

for $u_1 \neq u_2$, $(u_1, v_1) + (u_2, v_2) = (u_3, v_3)$ with $u_3 = \lambda^2 - u_1 - u_2$, $v_3 = \lambda(u_1 - u_3) - v_1$, $\lambda = (v_2 - v_1)/(u_2 - u_1)$; for $v_1 \neq 0$, $(u_1, v_1) + (u_1, v_1) = (u_3, v_3)$ with $u_3 = \lambda^2 - u_1 - u_2$, $v_3 = \lambda(u_1 - u_3) - v_1$, $\lambda = (3u_1^2 + a)/2v_1$;

$(u_1, v_1) + (u_1, -v_1) = \infty$;

$(u_1, v_1) + \infty = (u_1, v_1)$;

$\infty + (u_2, v_2) = (u_2, v_2)$;

$\infty + \infty = \infty$.

Messy to implement and test.

Much nicer than Weierstrass:

Montgomery-curve ECDH using the "Montgomery ladder"— our recommended method for Diffie–Hellman key exchange (e.g., for forward secrecy).

Montgomery ladder works only with u-coordinates of curve points P.

Montgomery ladder computes nP and $(n+1)P$ recursively from $\lfloor n/2 \rfloor P$ and $(\lfloor n/2 \rfloor + 1)P$ using one bit of n with **no branches**.

Curve selection

Many different standards:

1999 ANSI X9.62.

2000 IEEE P1363.

2000 SEC 2.

2000 NIST FIPS 186-2.

2001 ANSI X9.63.

2005 Brainpool.

2005 NSA Suite B.

2011 ANSSI FRP256V1.

Our new evaluation site:

`http://safecurves.cr.yp.to`

Avoiding known attacks

The curve must be elliptic.

The number of curve points
must be divisible by
a large prime number ℓ.
Standard attacks take time $\sqrt{\ell}$.
$\ell \approx 2^{200}$ is adequate;
$\ell \approx 2^{256}$ is conservative.

ℓ must not divide
p; $p - 1$; $p^2 - 1$;
$p^3 - 1$; $\ldots$; $p^{20} - 1$.
This guarantees that there are
no “transfers” to clocks etc.

Avoiding unnecessary structure

Simplify the security story:
avoid possible attack vectors
even if no attacks are known.

Require large "CM field discriminant". See, e.g., SafeCurves.

Brainpool, Suite B, ANSSI, SafeCurves: require prime p.

Brainpool and SafeCurves: prohibit ℓ dividing $p^k - 1$ for each $k < (\ell - 1)/100$.

Rigidity

Another conceivable source of security problems:

- there's another attack against a small fraction of curves;
- public ECC cryptanalysis has missed this attack;
- the attacker has figured out this attack;
- the attacker has **manipulated** choices of standard curves to allow the attack.

NIST curves claim to be
"verifiably random":
$y^2 = x^3 - 3x + b$ where
b is derived from
SHA-1 hash of a public seed.

NIST curves claim to be "verifiably random": $y^2 = x^3 - 3x + b$ where b is derived from SHA-1 hash of a public seed.

But is the seed actually random?

Attacker could have tried many seeds to find a curve with a one-in-a-billion weakness. Not "verifiable" at all!

NIST curves claim to be
"verifiably random":
$y^2 = x^3 - 3x + b$ where
b is derived from
SHA-1 hash of a public seed.

But is the seed actually random?

Attacker could have tried
many seeds to find a curve with
a one-in-a-billion weakness.
Not "verifiable" at all!

ANSSI response: use our
"random" curve instead.

Rigidity limits number of curves
that can be generated
by a curve-generation process.

Brainpool, somewhat rigid:
b is some sort of hash
of digits of π and e.

Rigidity limits number of curves
that can be generated
by a curve-generation process.

Brainpool, somewhat rigid:
b is some sort of hash
of digits of π and e.

Not completely explained:
why this particular hash?
why π and not $\sqrt{2}$? etc.
But not much flexibility.

Rigidity limits number of curves that can be generated by a curve-generation process.

Brainpool, somewhat rigid:
b is some sort of hash of digits of π and e.

Not completely explained:
why this particular hash?
why π and not $\sqrt{2}$? etc.
But not much flexibility.

Our recommendation, fully rigid:
b is *smallest* positive integer passing explained criteria.

ECC security

Covered so far:
hard to compute ECC user's secret key from public key.

But real-world ECC is still being broken!

ECC implementations

- produce incorrect results for some rare inputs;
- leak secret data for input points *off* curve;
- leak secret data through timing;

etc. Attackers exploit this.

Better choices of curves
allow **simple** implementations
to be **secure** implementations.

This is the primary
motivation for SafeCurves.

Better choices of curves
allow **simple** implementations
to be **secure** implementations.

This is the primary
motivation for SafeCurves.

Example of new requirement:
twist security.

If curve isn't twist-secure:
Twist attacks break
ladder implementations
that don't check whether
input point is on curve.
Security-simplicity conflict.

		Parameters:			
Curve	**Safe?**	[field]	[equation]	[base]	[rho]
Anomalous	**False**	True ✔	True ✔	True ✔	True ✔
M-221	True ✔	True ✔	True ✔	True ✔	True ✔
E-222	True ✔	True ✔	True ✔	True ✔	True ✔
NIST P-224	**False**	True ✔	True ✔	True ✔	True ✔
Curve1174	True ✔	True ✔	True ✔	True ✔	True ✔
Curve25519	True ✔	True ✔	True ✔	True ✔	True ✔
BN(2,254)	**False**	True ✔	True ✔	True ✔	True ✔
brainpoolP256t1	**False**	True ✔	True ✔	True ✔	True ✔
ANSSI FRP256v1	**False**	True ✔	True ✔	True ✔	True ✔
NIST P-256	**False**	True ✔	True ✔	True ✔	True ✔
secp256k1	**False**	True ✔	True ✔	True ✔	True ✔
E-382	True ✔	True ✔	True ✔	True ✔	True ✔
M-383	True ✔	True ✔	True ✔	True ✔	True ✔
Curve383187	True ✔	True ✔	True ✔	True ✔	True ✔
brainpoolP384t1	**False**	True ✔	True ✔	True ✔	True ✔
NIST P-384	**False**	True ✔	True ✔	True ✔	True ✔
Curve3617	True ✔	True ✔	True ✔	True ✔	True ✔

ECDLP security:			ECC security:			
transfer	disc	rigid	ladder	twist	complete	ind
False	False	True ✔	False	False	False	False
True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
True ✔	True ✔	False	False	False	False	False
True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
False	False	True ✔	False	False	False	False
True ✔	True ✔	True ✔	False	False	False	False
True ✔	True ✔	False	False	False	False	False
True ✔	True ✔	False	False	True ✔	False	False
True ✔	False	True ✔	False	True ✔	False	False
True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
True ✔	True ✔	True ✔	False	True ✔	False	False
True ✔	True ✔	False	False	True ✔	False	False
True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔