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.

ves:
g safe curves for
curve cryptography

. Bernstein
ty of Illinois at Chicago &
che Universiteit Eindhoven

ange
che Universiteit Eindhoven

/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 EC

"Index c
fastest n
to break

Long his
many m
1975, Cl
1977, lin
1982, qu
1990, nu
1994, fu
2006, m
2013, x^q

(FFS is

ves for
tography

n

is at Chicago &
siteit Eindhoven

siteit Eindhoven

ves.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
to break original D

Long history, inclu
many major impro
1975, CFRAC;
1977, linear sieve (
1982, quadratic sie
1990, number-field
1994, function-fiel
2006, medium-prin
2013, $x^q - x$ FFS.

(FFS is not relevar

ago &
hoven

hoven

p.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 RS

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

(FFS is not relevant to RSA

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.)

raphy

ey signatures:
A, DSA, ECDSA.
ses: signed OS updates,
tificates, e-passports.

ey encryption:
A, DH, ECDH.
ses: SSL key exchange,
Phone mail download.

ey encryption:
S, Salsa20.
ses: disk encryption,
_ 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 ma
$\approx$ 100 s

Costs of
breaking
$\approx 2^{120}$,
$\approx 2^{110}$,
$\approx 2^{100}$,
$\approx 2^{80}$, 2

res:
ECDSA.
l OS updates,
-passports.

tion:
CDH.
ey exchange,
il download.

tion:
).
ncryption,
on.

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
$\approx$ 100 scientific pa

Costs of these algo
breaking RSA-102
$\approx 2^{120}$, 2^{170}, CFR
$\approx 2^{110}$, 2^{160}, LS;
$\approx 2^{100}$, 2^{150}, QS;
$\approx 2^{80}$, 2^{112}, NFS.

tes,

.

ge,

d.

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 improvem
$\approx$ 100 scientific papers.

Costs of these algorithms fo
breaking RSA-1024, RSA-20
$\approx 2^{120}$, 2^{170}, CFRAC;
$\approx 2^{110}$, 2^{160}, LS;
$\approx 2^{100}$, 2^{150}, QS;
$\approx 2^{80}$, 2^{112}, NFS.

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.

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:
≈ 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."

C?

alculus":
method we know
original DH and RSA.

story, including
ajor improvements:
FRAC;
near sieve (LS);
uadratic sieve (QS);
umber-field sieve (NFS);
nction-field sieve (FFS);
edium-prime FFS/NFS;
$- x$ FFS.

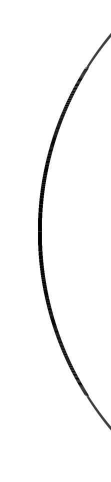
not relevant to RSA.)

Also many smaller improvements:
≈ 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 cloc



This is t

Warning
This is *n*
"Elliptic

e know
DH and RSA.

ding
ovements:

(LS);
eve (QS);
d sieve (NFS);
d sieve (FFS);
me FFS/NFS;
.

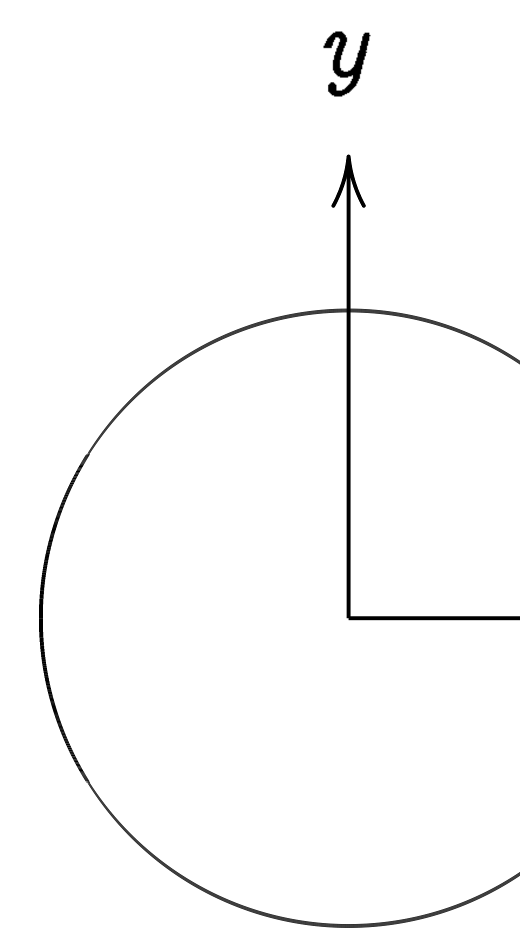
nt to RSA.)

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.

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

Warning:
This is *not* an ellip
"Elliptic curve" $\neq$

SA.

FS);

FS);

FS;

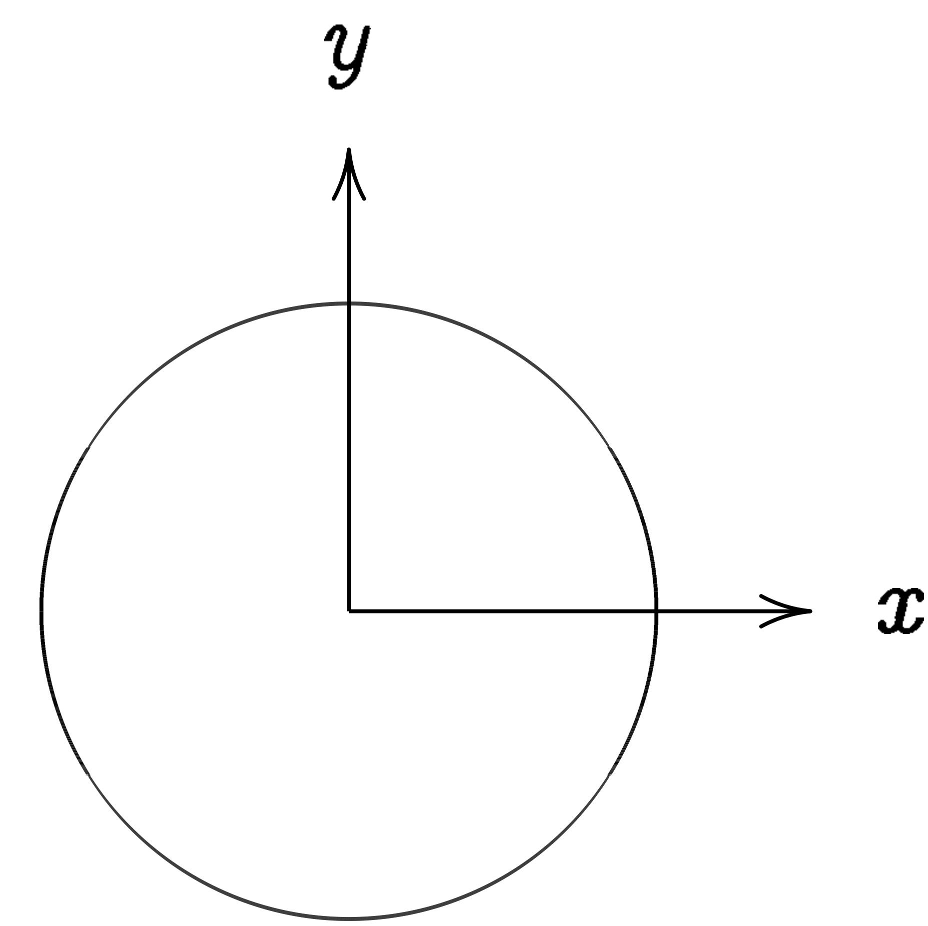
.)

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."

<u>The clock</u>



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

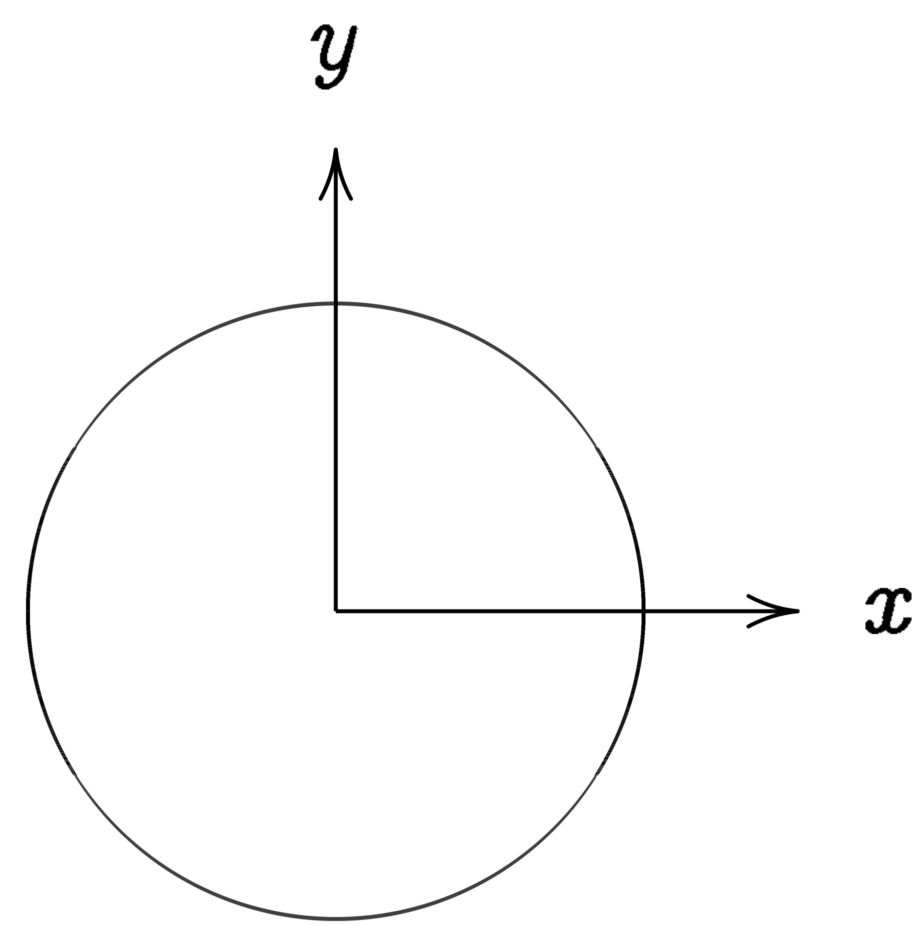
Warning:
This is *not* an elliptic curve.
"Elliptic curve" $\neq$ "ellipse."

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

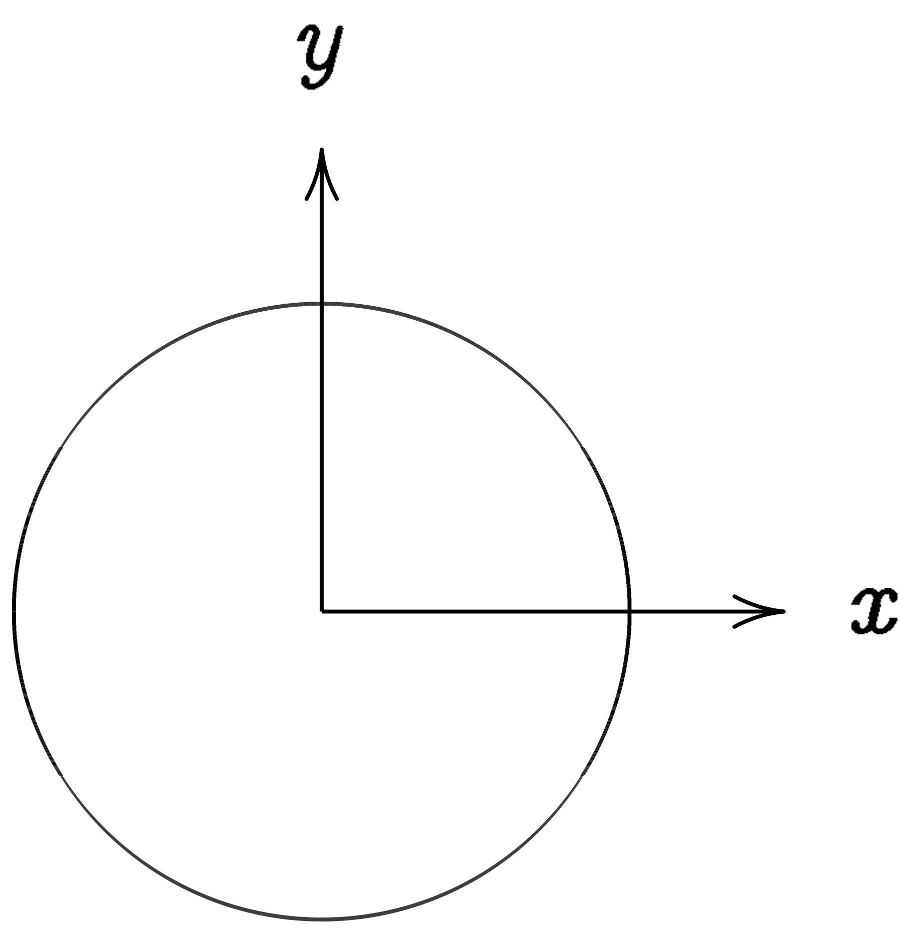
Warning:
This is *not* an elliptic curve.
"Elliptic curve" $\neq$ "ellipse."

ny smaller improvements:
cientific papers.

these algorithms for
RSA-1024, RSA-2048:
2^{170}, CFRAC;
2^{160}, LS;
2^{150}, QS;
2^{112}, NFS.

ller "Use of
curves in cryptography":
tremely unlikely
'index calculus' attack
lliptic curve method
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."

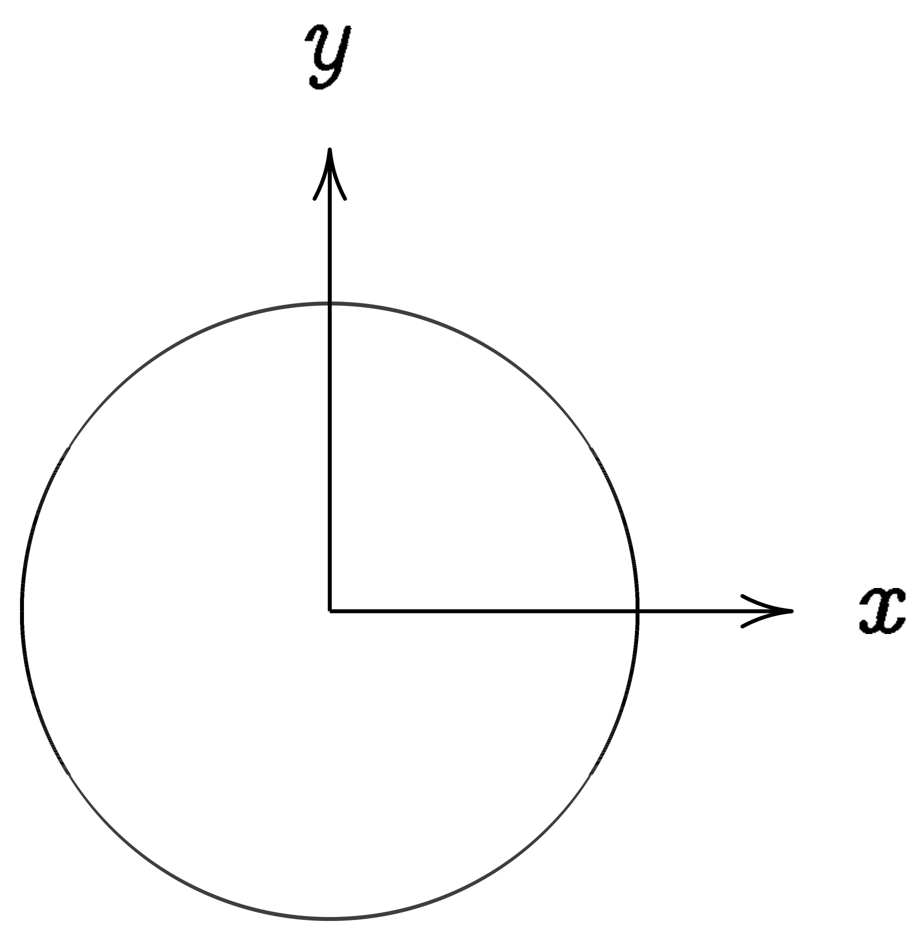
Example

improvements:
apers.

orithms for
4, RSA-2048:
AC;

of
ryptography":
nlikely
culus' attack
ve method
o 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 point

nents:

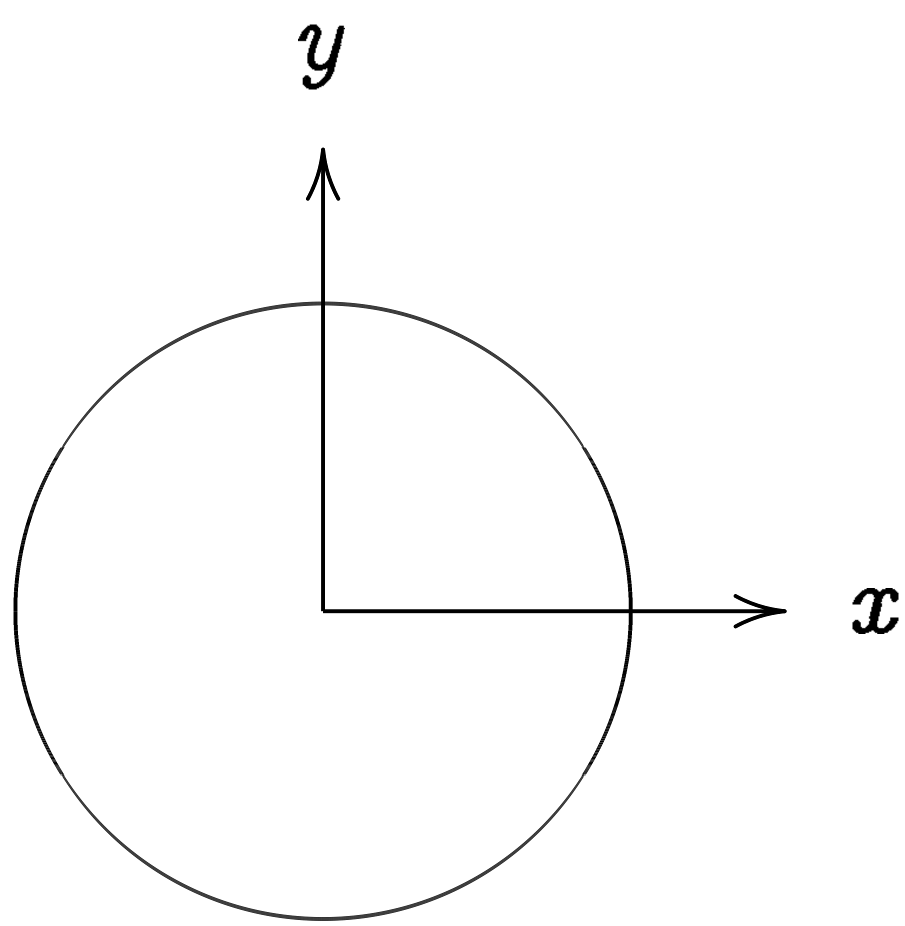
r

048:

ny":

ck

The clock



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

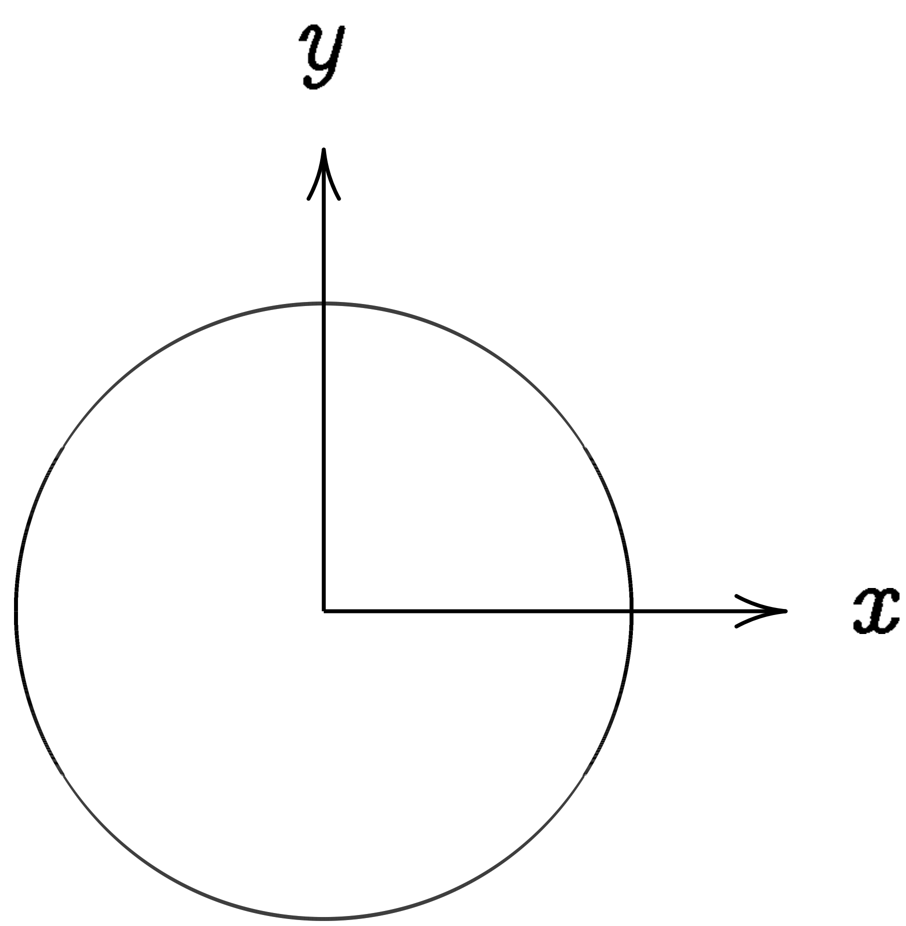
Warning:

This is *not* an elliptic curve.

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

Examples of points on this c

The clock



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

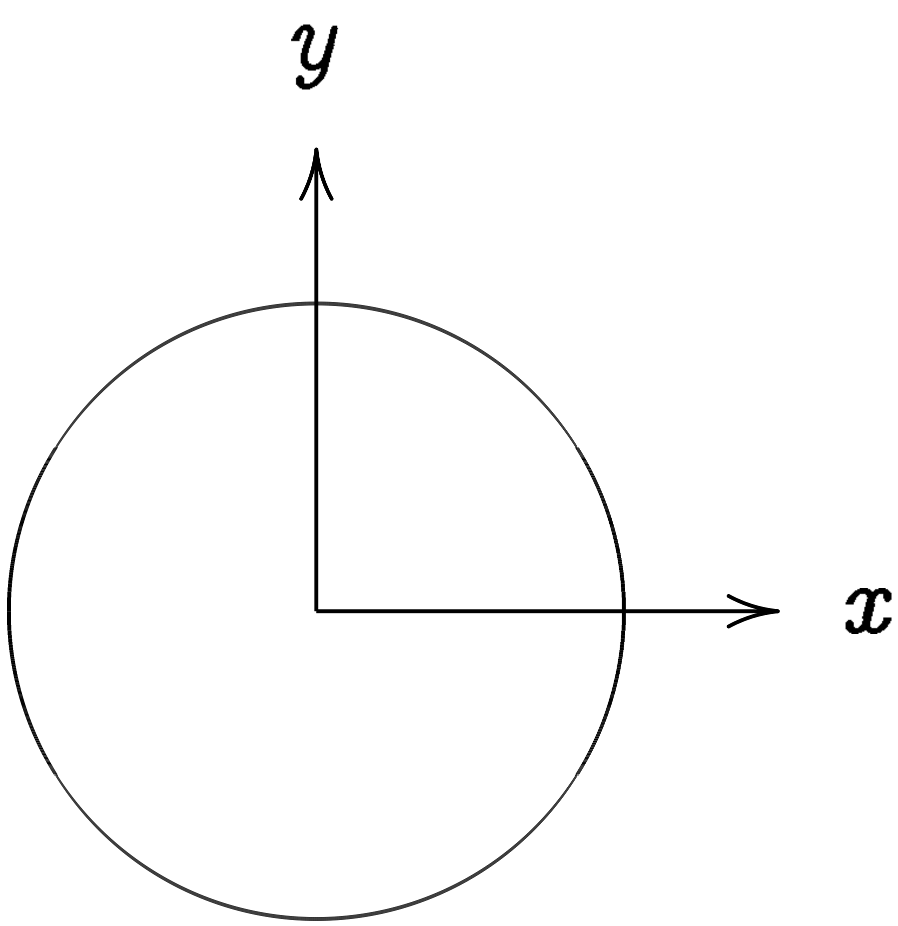
Warning:

This is *not* an elliptic curve.

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

Examples of points on this curve:

The clock



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

Warning:

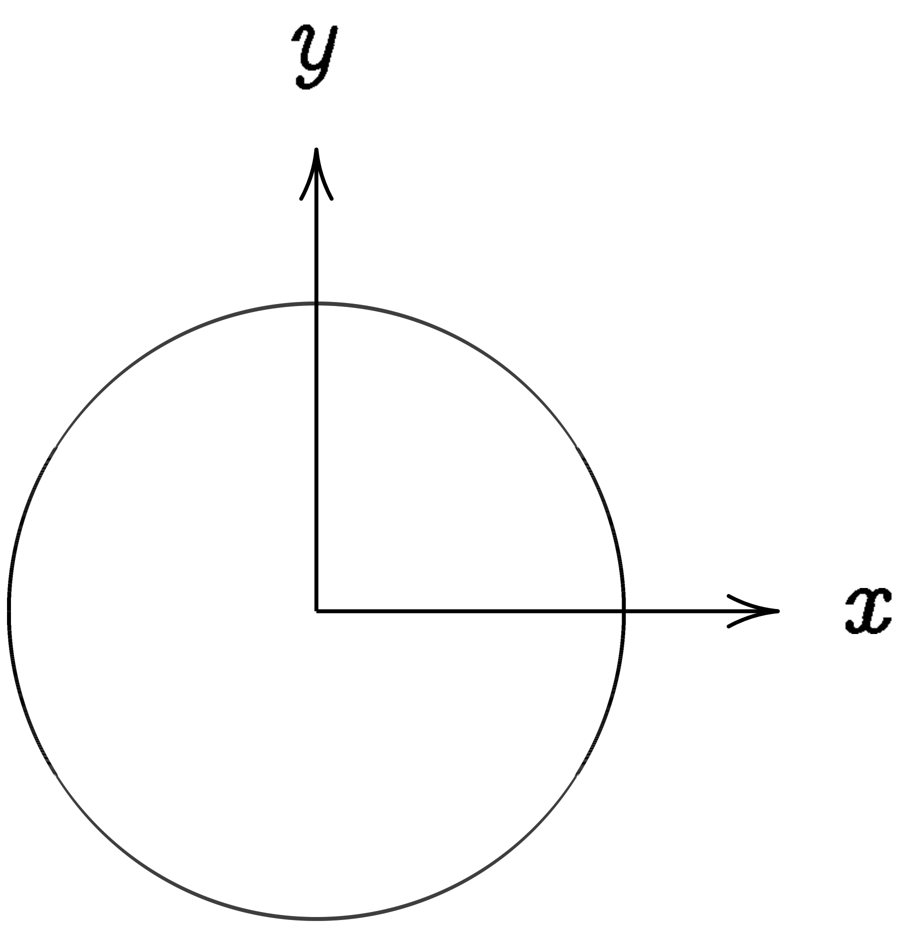
This is *not* an elliptic curve.

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

Examples of points on this curve:

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

The clock



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

Warning:

This is *not* an elliptic curve.

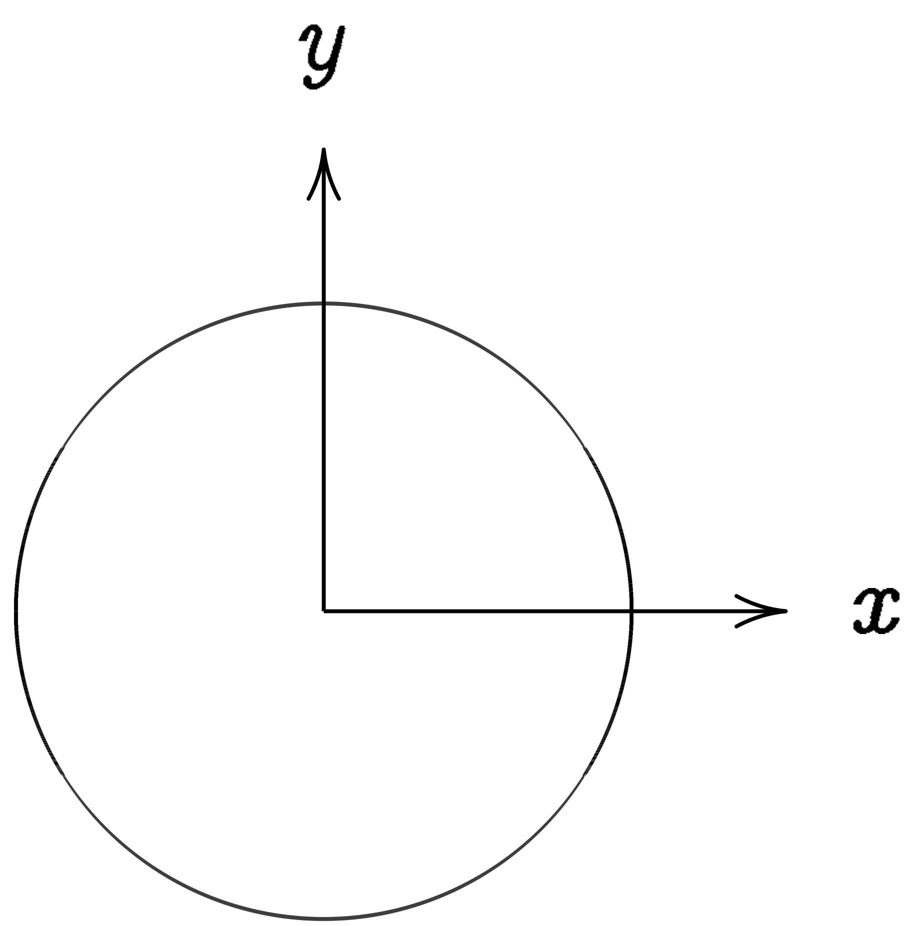
"Elliptic curve" $\neq$ "ellipse."

Examples of points on this curve:

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

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

The clock



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

Warning:

This is *not* an elliptic curve.

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

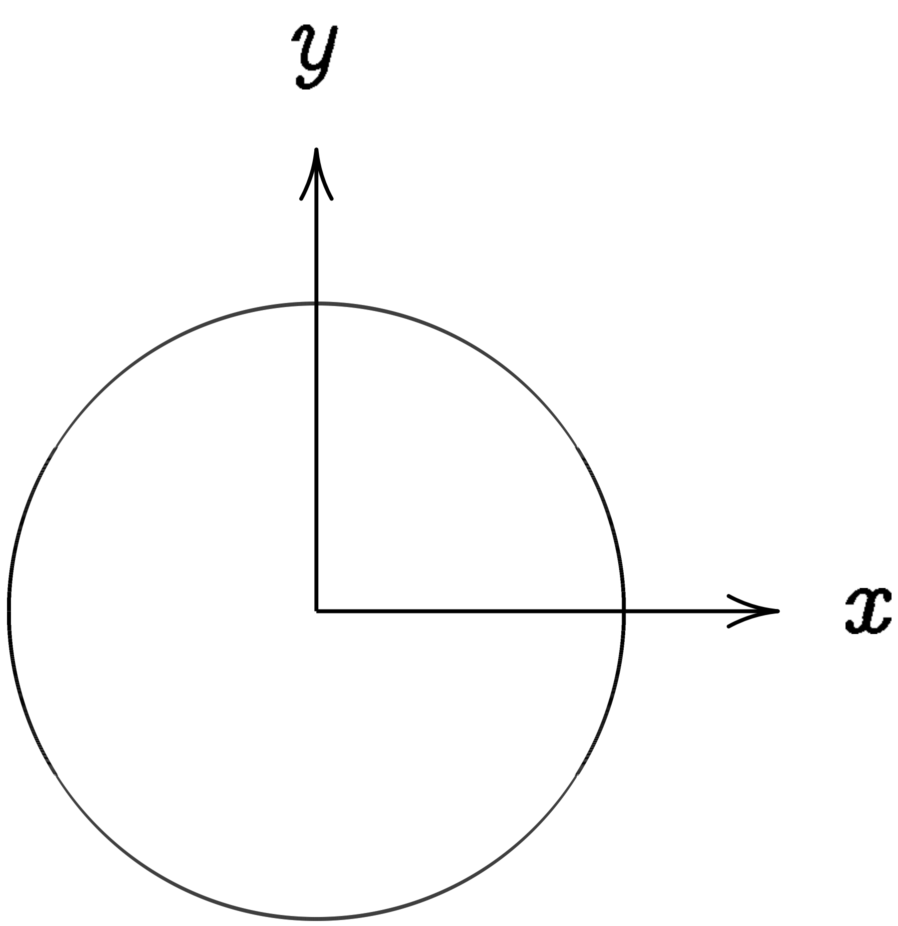
Examples of points on this curve:

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

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

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

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:

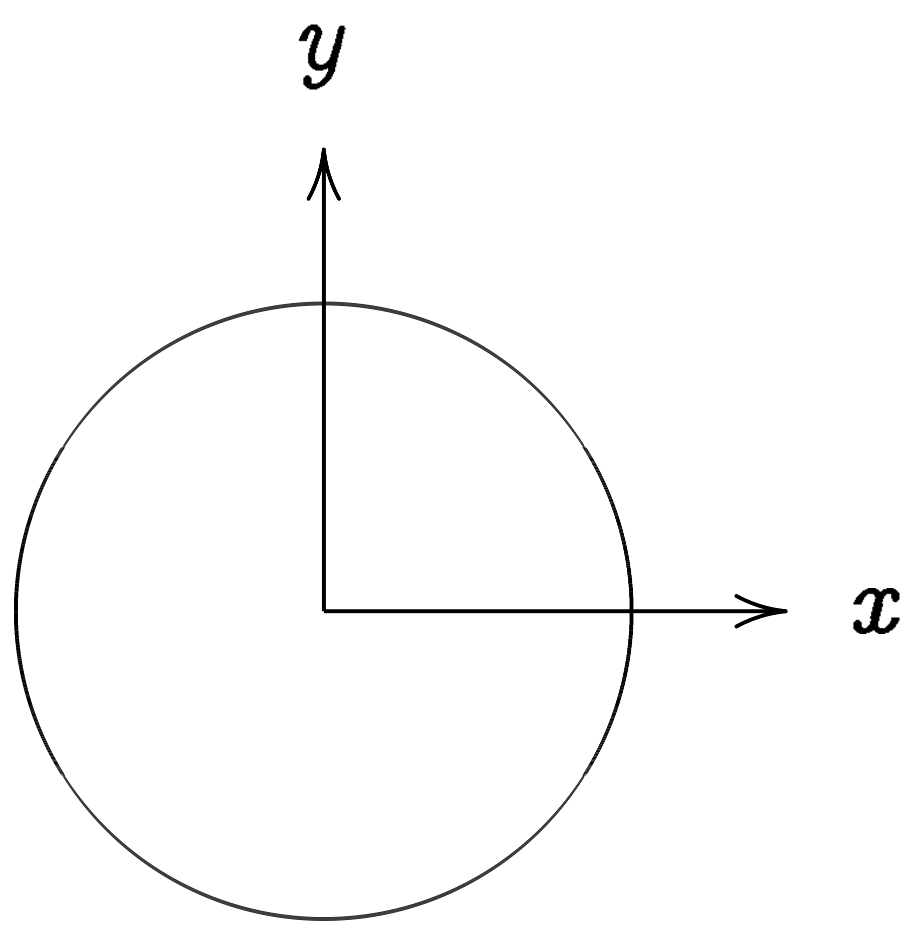
$(0, 1) =$ "12:00".

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

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

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

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:

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

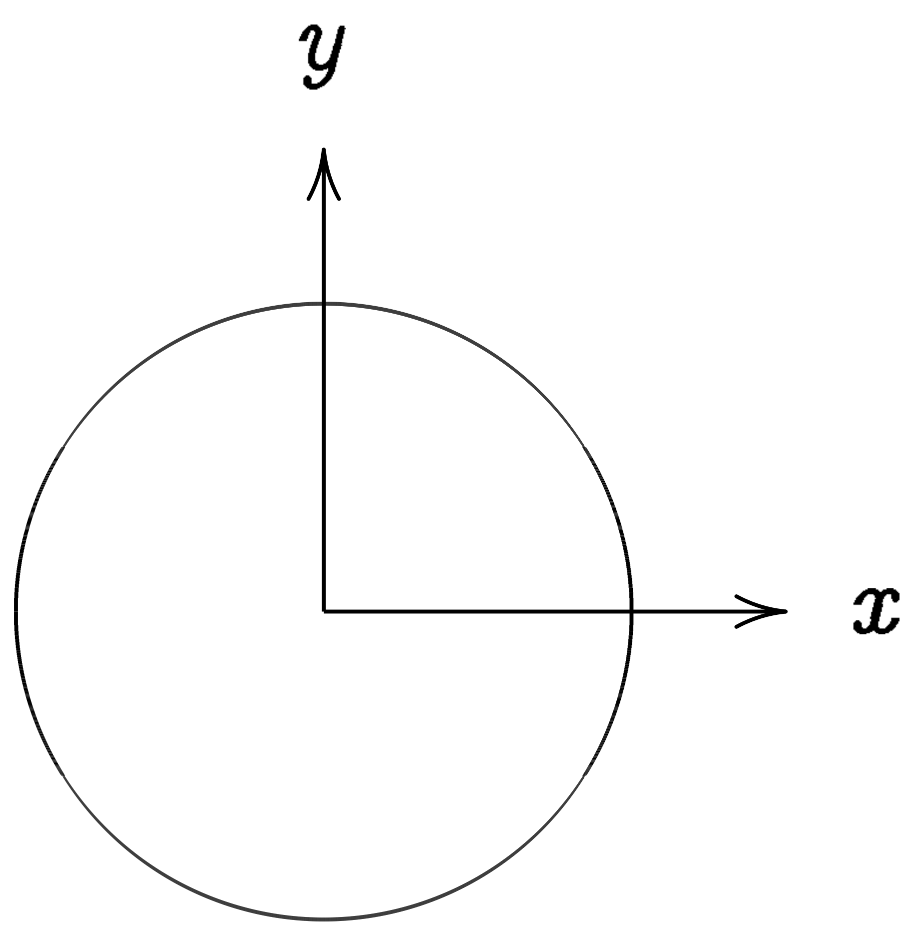
$(0, -1) =$ "6:00".

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

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

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

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:

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

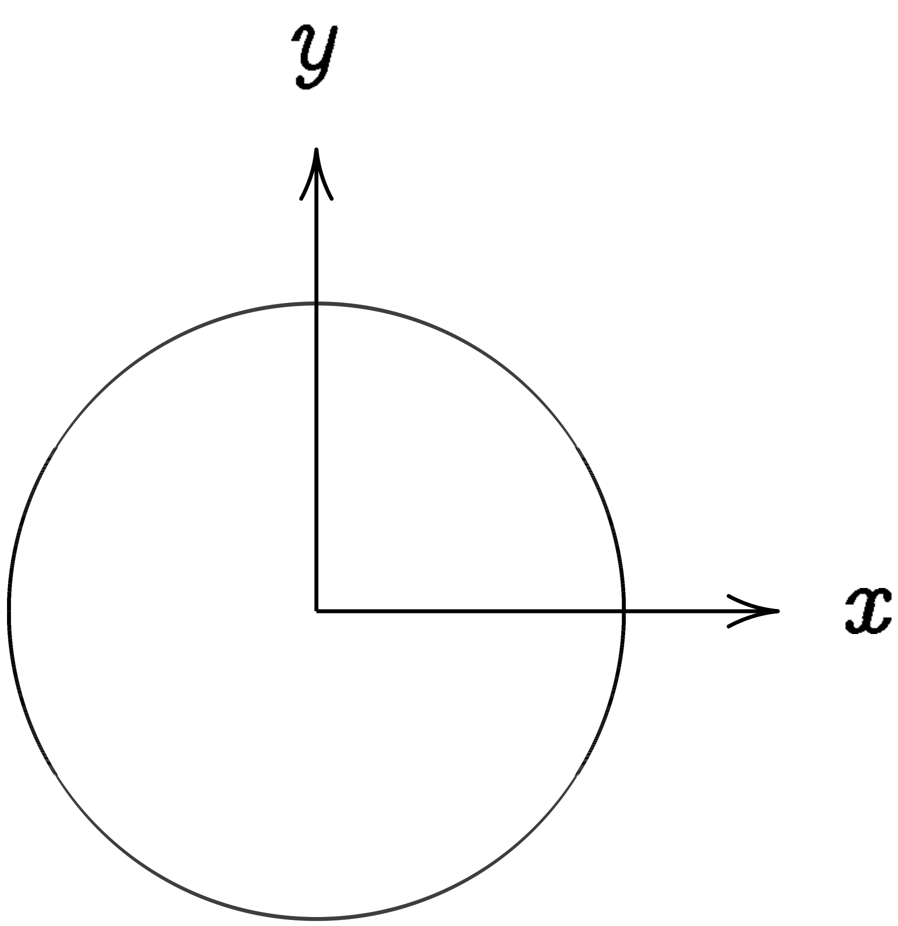
$(0, -1) =$ "6:00".

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

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

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

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:

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

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

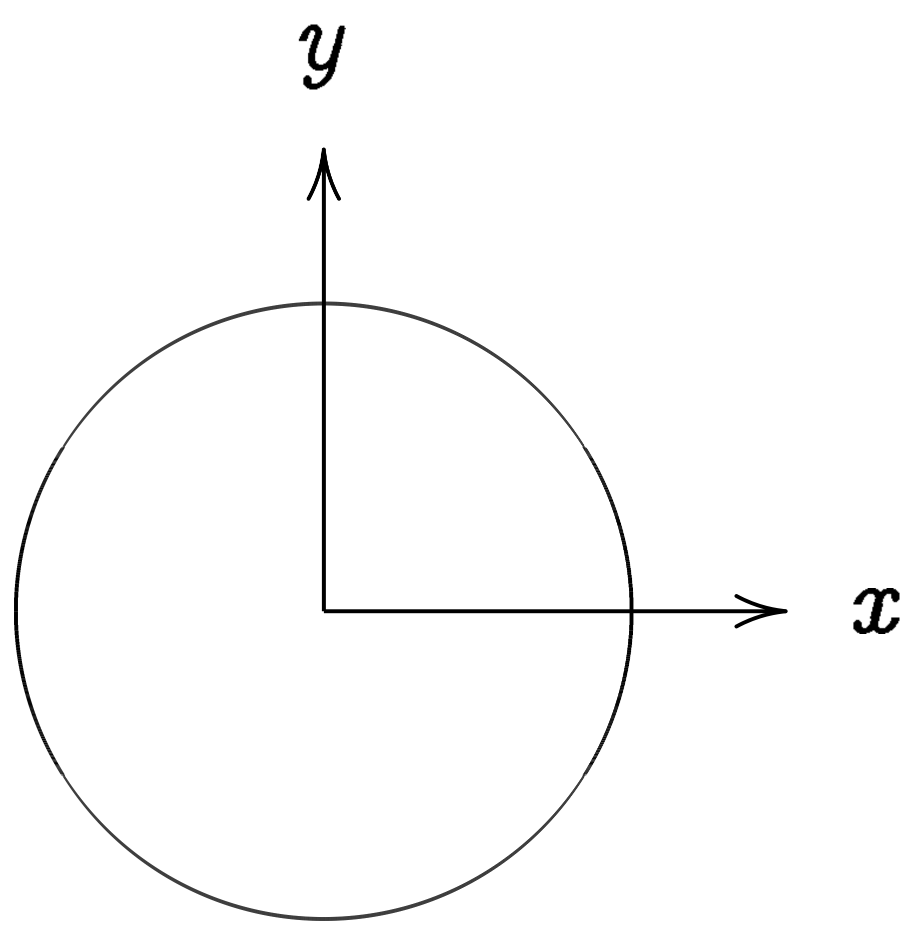
$(1, 0) =$ "3:00".

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

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

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

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:

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

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

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

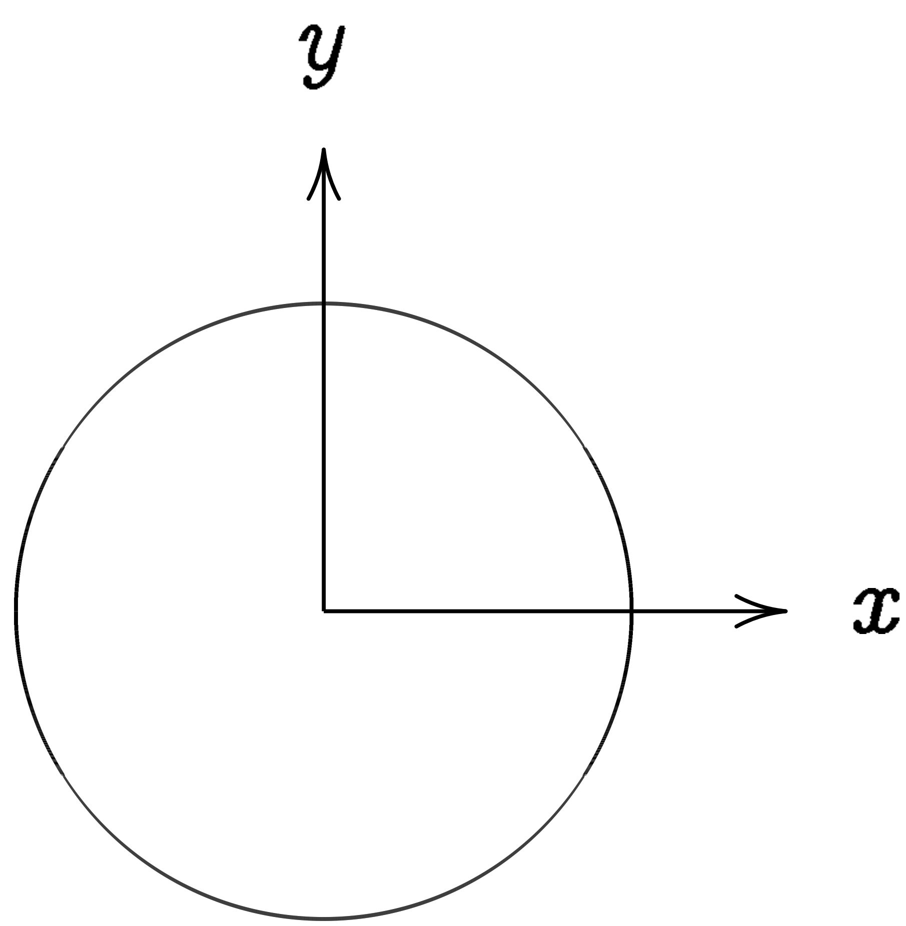
$(-1, 0) =$ "9:00".

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

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

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

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:

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

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

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

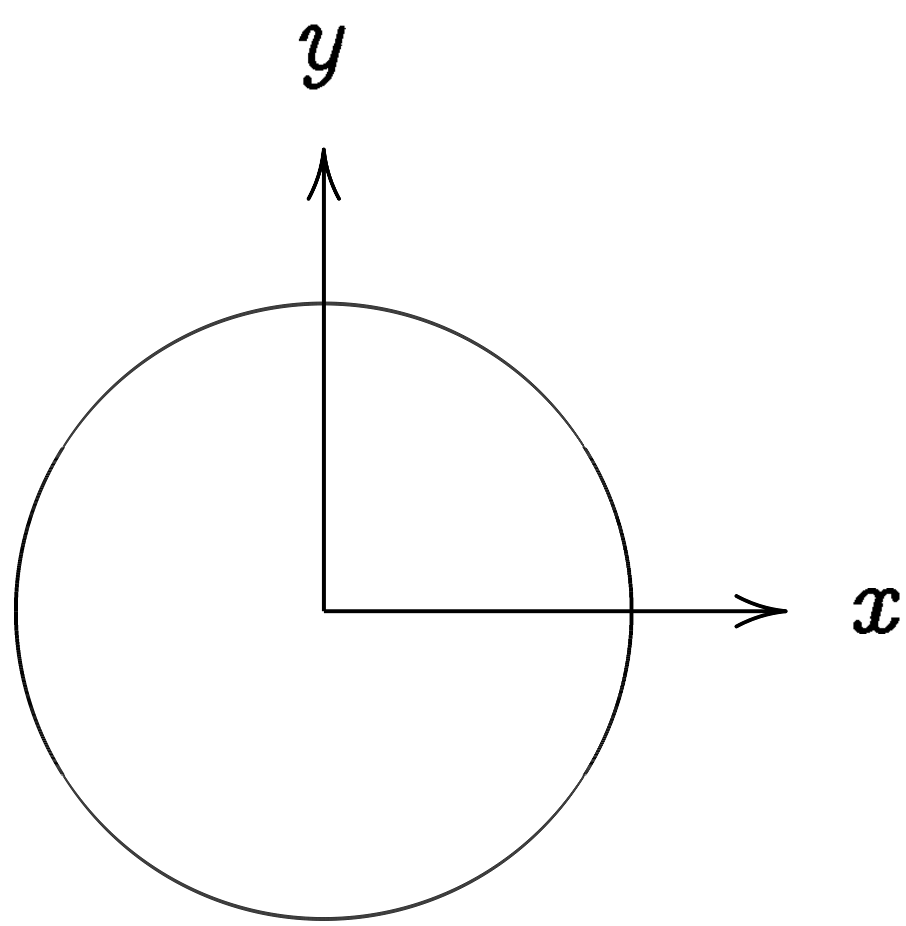
$(-1, 0) =$ "9:00".

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

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

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

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:

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

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

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

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

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

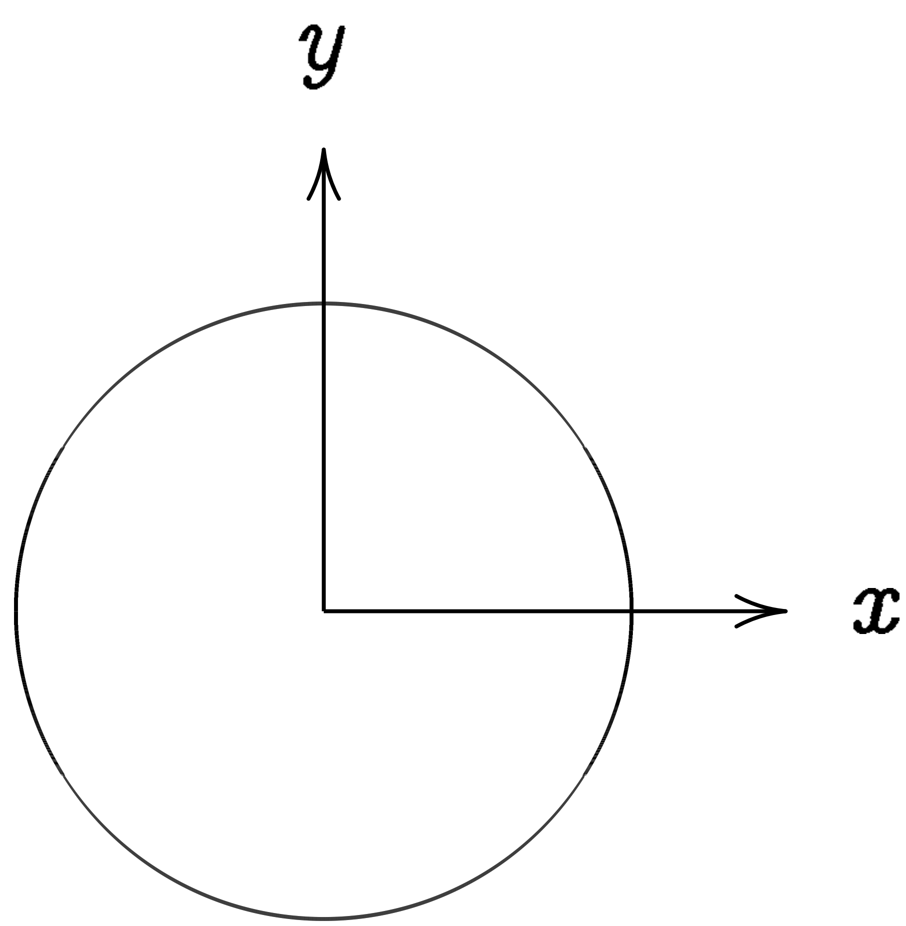
$(1/2, -\sqrt{3/4}) =$ "5:00".

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

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

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

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:

$(0, 1) =$ "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".

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

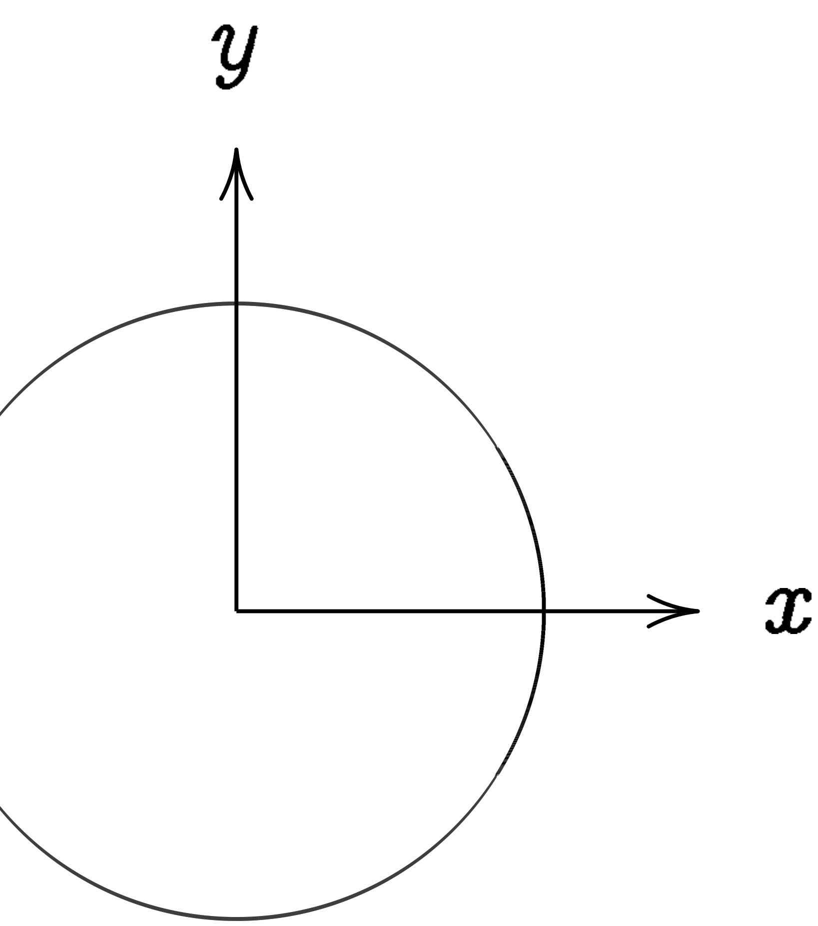
$(3/5, -4/5)$. $(-3/5, -4/5)$.

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

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

Many more.

ck



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

:

not an elliptic curve.

curve" $\neq$ "ellipse."

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".

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

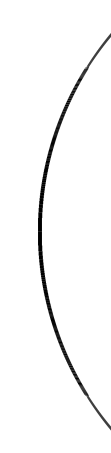
$(3/5, -4/5)$. $(-3/5, -4/5)$.

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

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

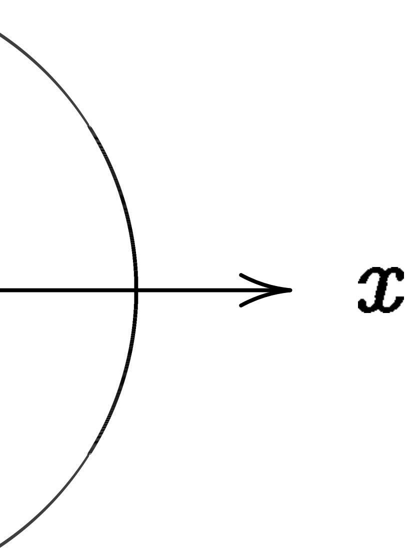
Many more.

Addition



$x^2 + y^2$

$x = \sin$



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

ptic curve.

"ellipse."

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".

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

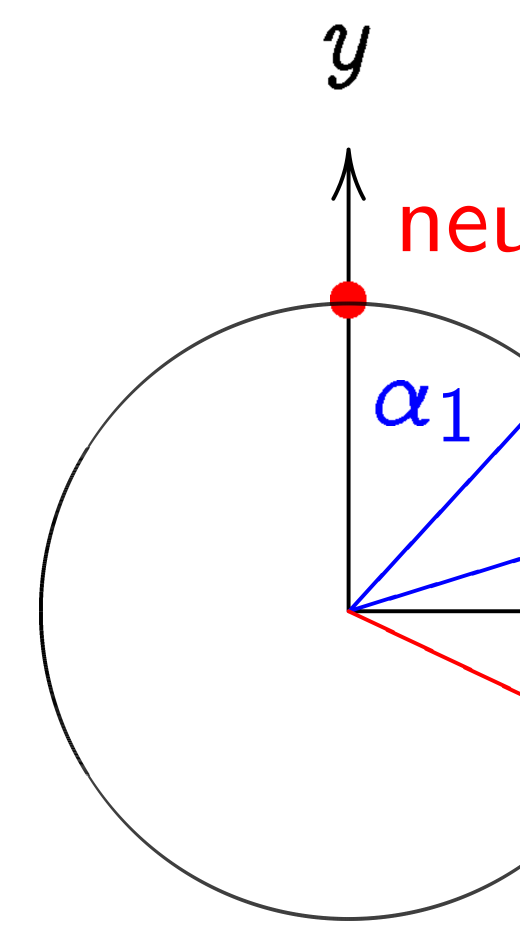
$(3/5, -4/5)$. $(-3/5, -4/5)$.

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

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

Many more.

Addition on the cl



$x^2 + y^2 = 1$, para

$x = \sin\alpha$, $y =$ co

1.

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".

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

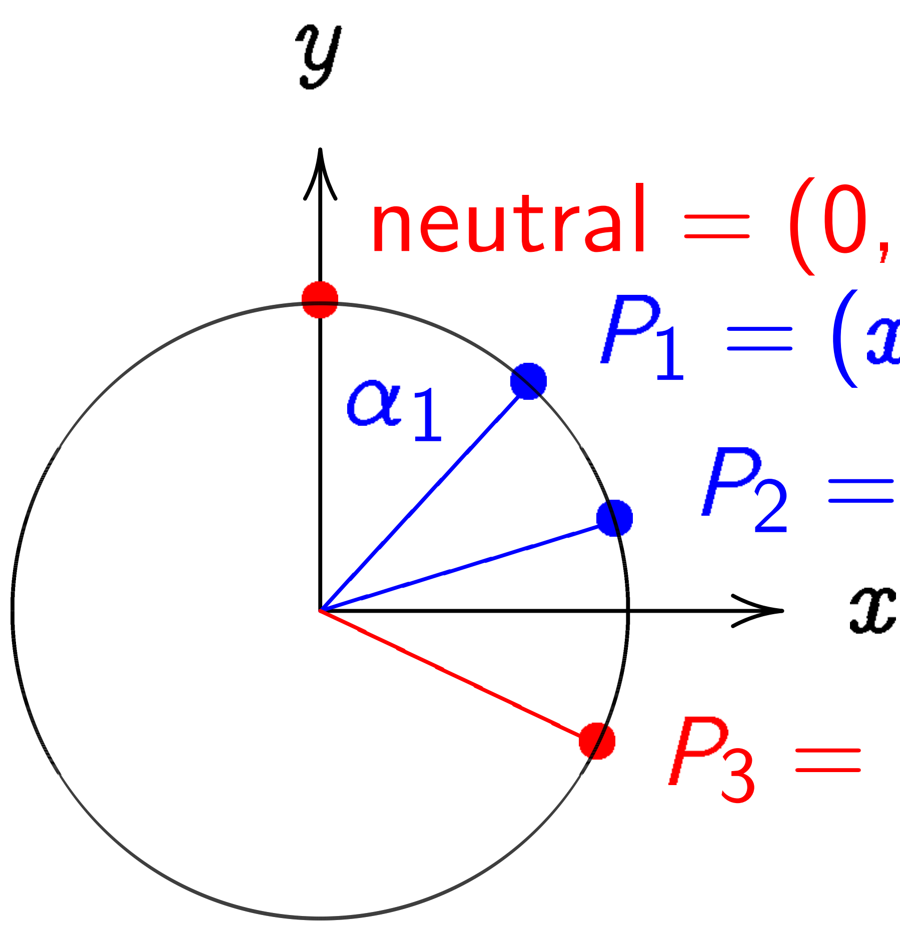
$(3/5, -4/5)$. $(-3/5, -4/5)$.

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

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

Many more.

Addition on the clock:



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

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

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".

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

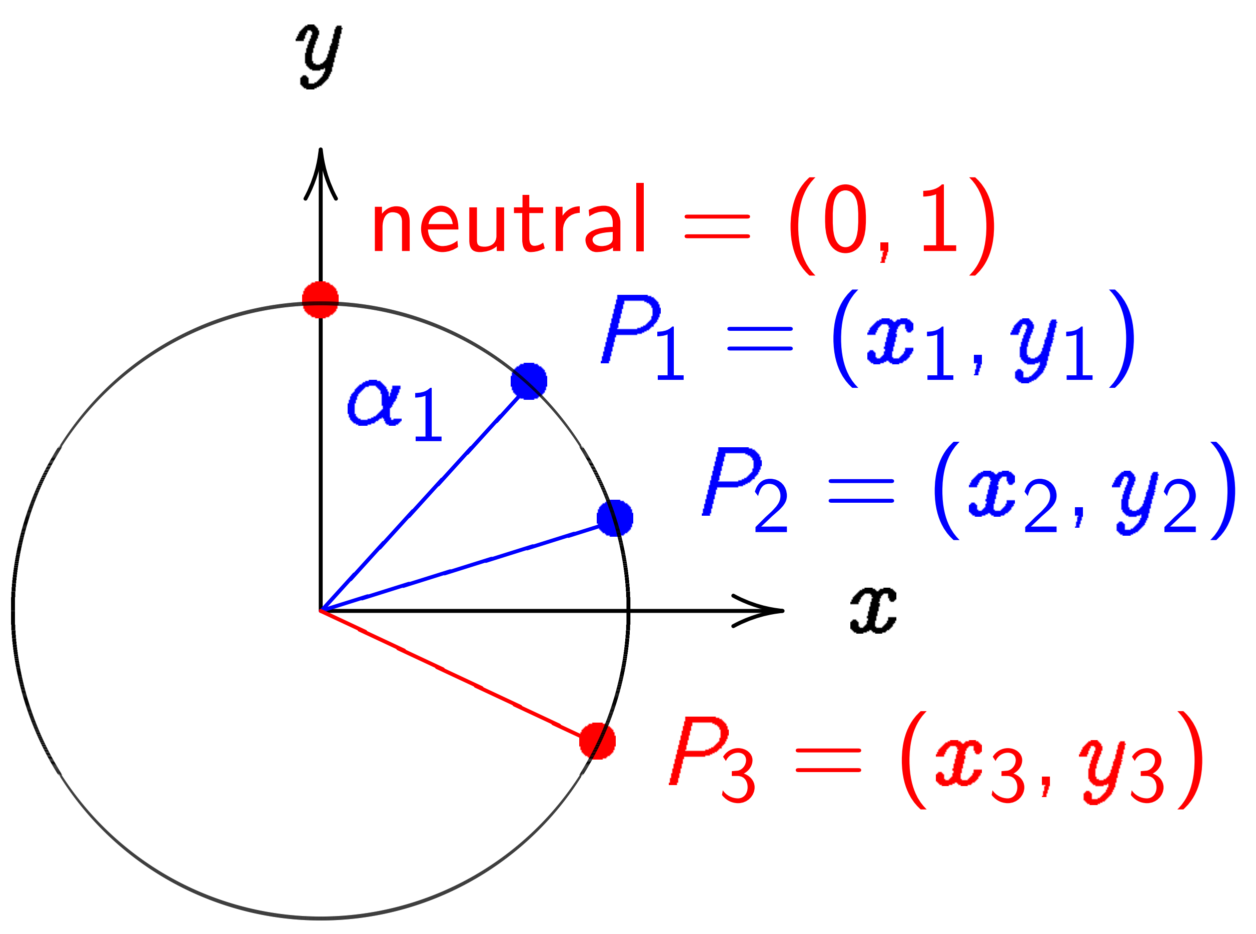
$(3/5, -4/5)$. $(-3/5, -4/5)$.

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

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

Many more.

Addition on the clock:



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

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

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".

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

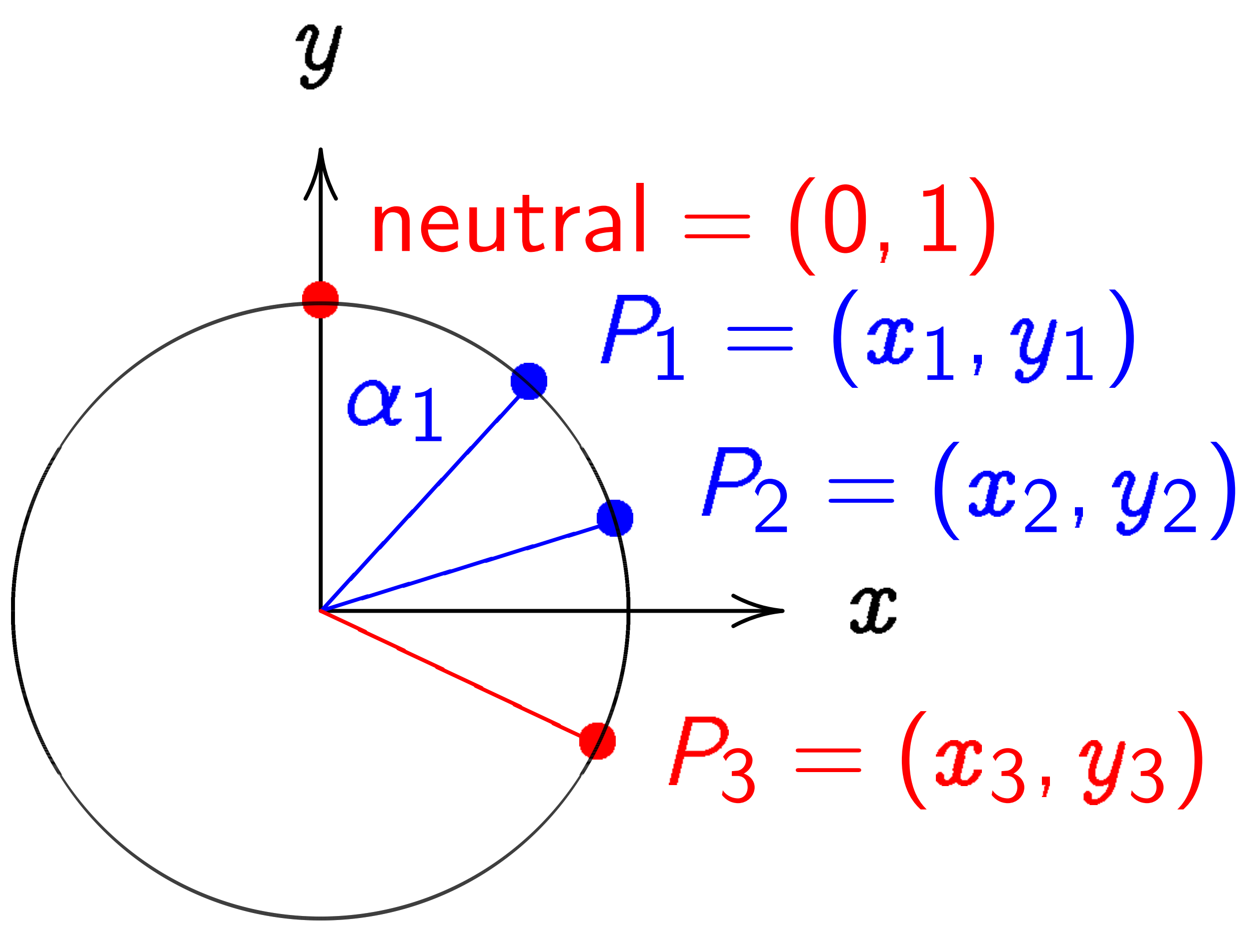
$(3/5, -4/5)$. $(-3/5, -4/5)$.

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

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

Many more.

Addition on the clock:



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

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".

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

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

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

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

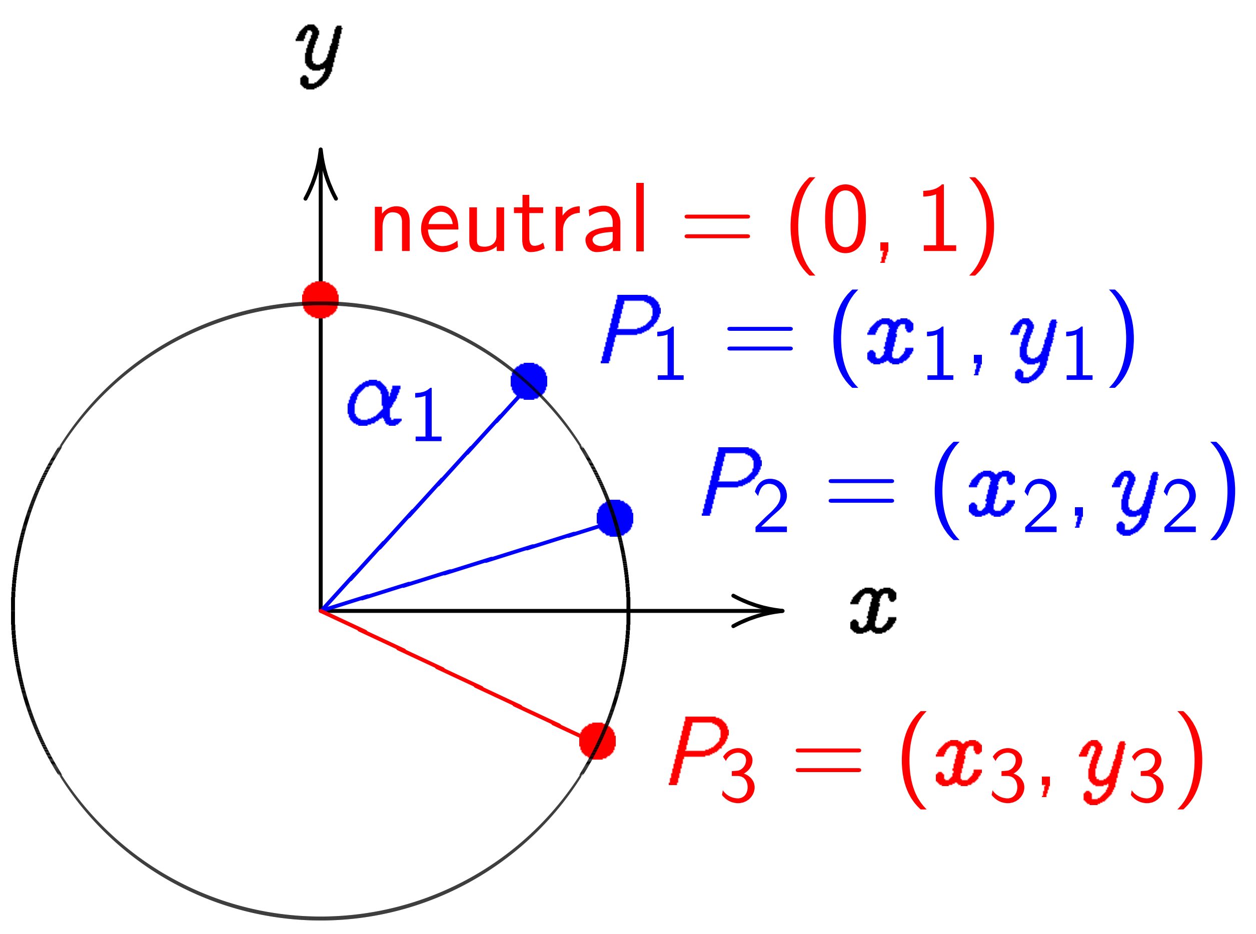
$(3/5, -4/5)$. $(-3/5, -4/5)$.

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

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

Many more.

Addition on the clock:



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

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

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

$(\sin\alpha_1\cos\alpha_2 + \cos\alpha_1\sin\alpha_2,$

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".

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

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

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

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

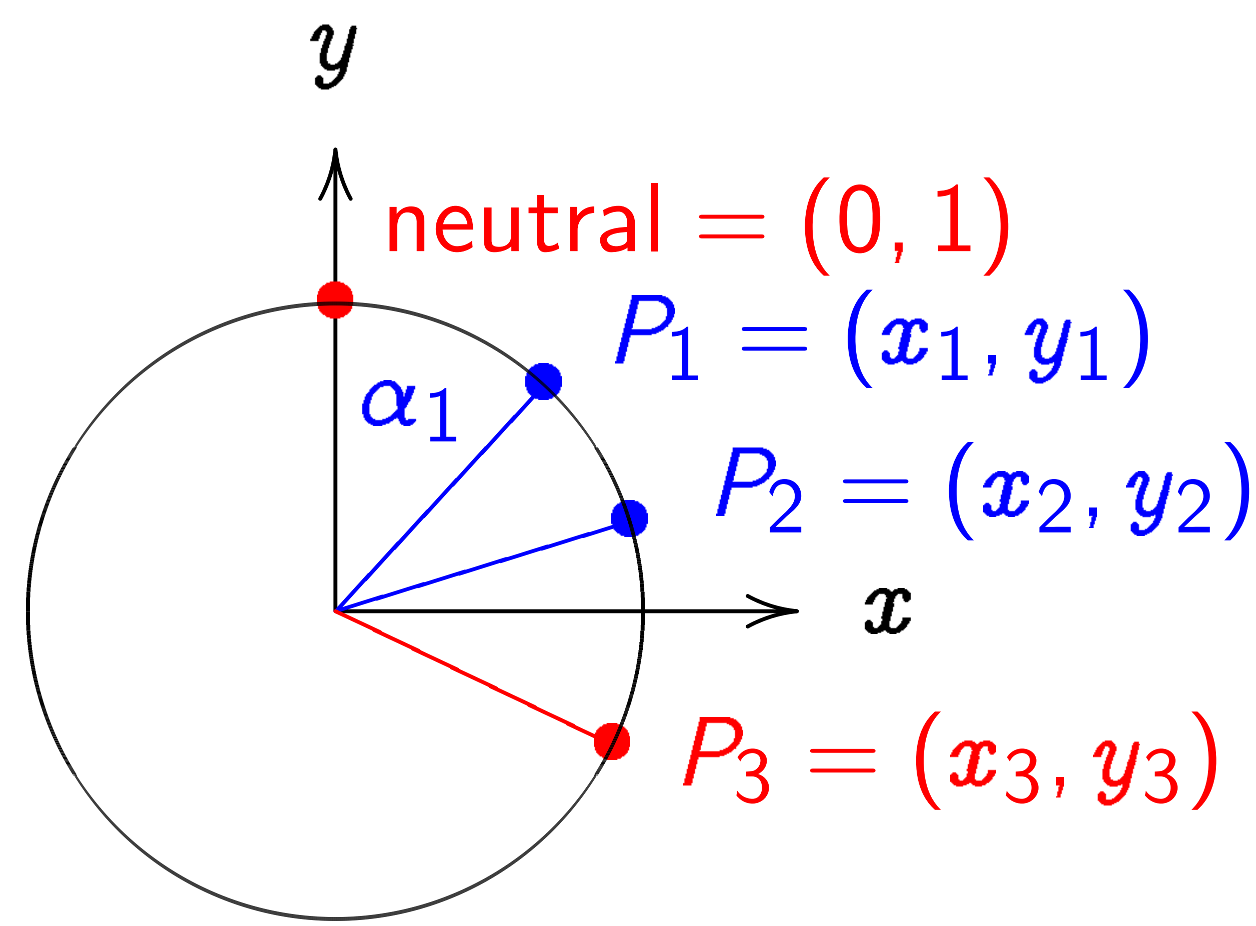
$(3/5, -4/5)$. $(-3/5, -4/5)$.

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

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

Many more.

Addition on the clock:



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

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

es of points on this curve:

“12:00”.

= “6:00”.

“3:00”.

= “9:00”.

1/2) = “2:00”.

$\sqrt{3/4}$) = “5:00”.

$-\sqrt{3/4}$) = “7:00”.

$\sqrt{1/2}$) = “1:30”.

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

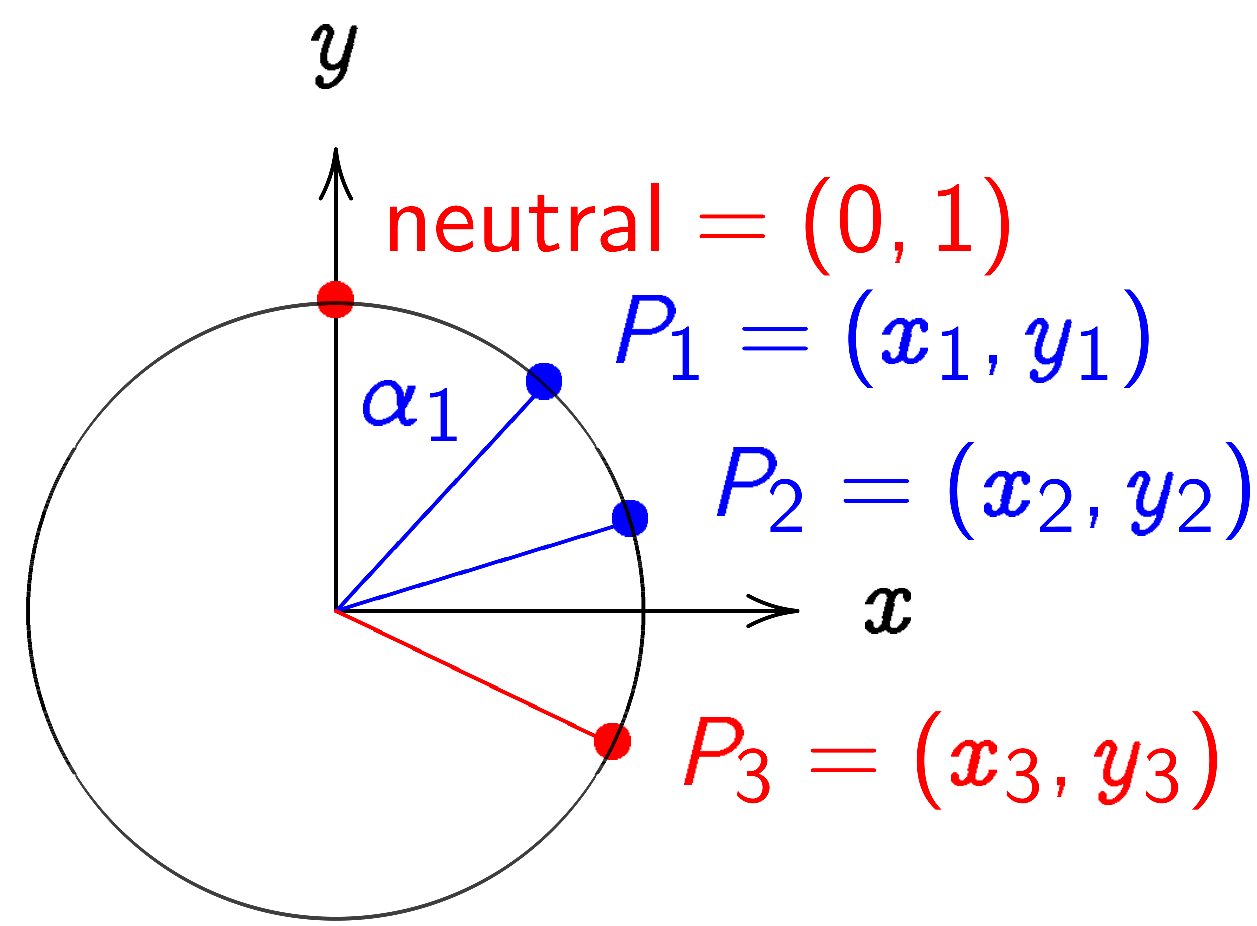
4/5). $(-3/5, -4/5)$.

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

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

ore.

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 ac

Use Cart

addition

for the c

sum of (

$(x_1 y_2 +$

s on this curve:

:00”.

“5:00”.

= “7:00”.

“1:30”.

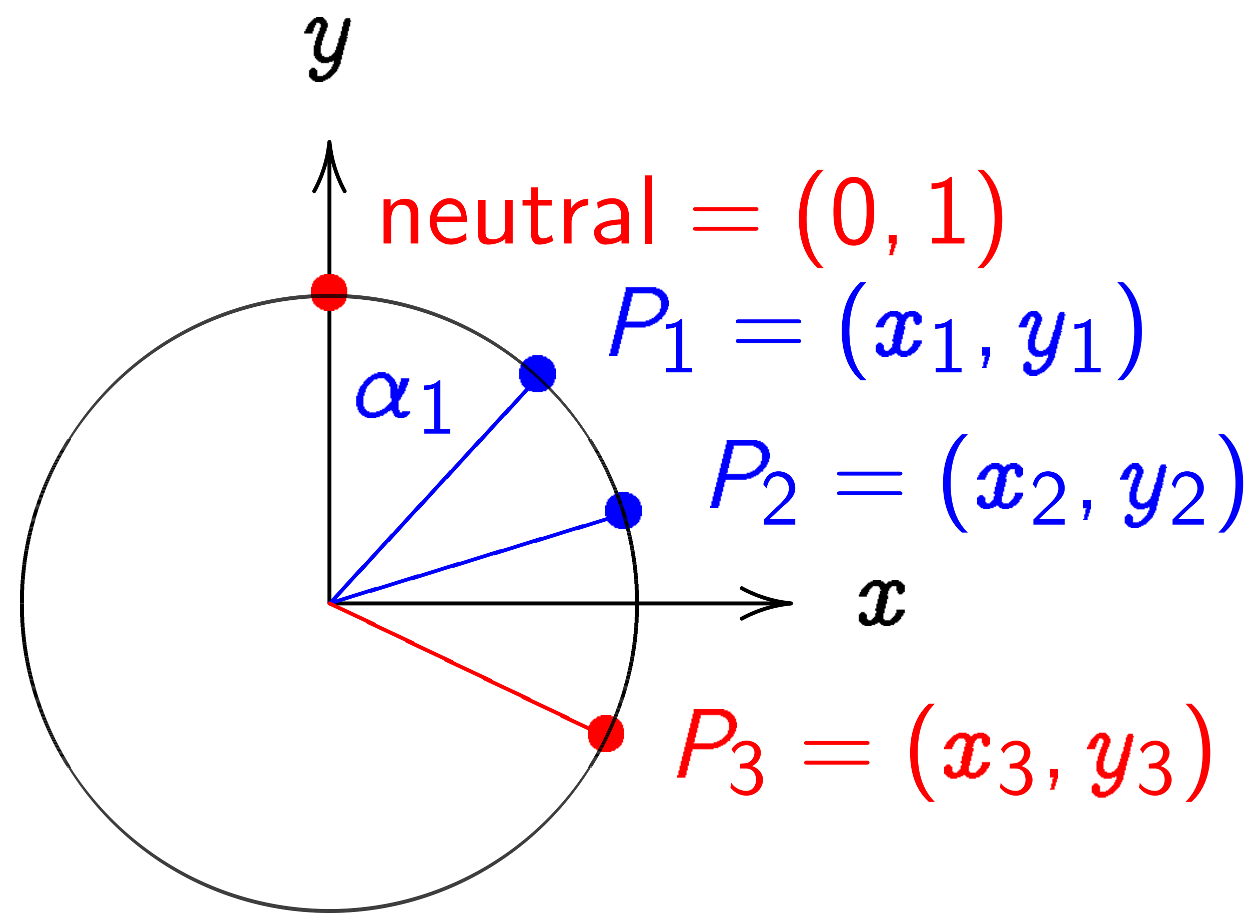
, 4/5).

/5, −4/5).

, 3/5).

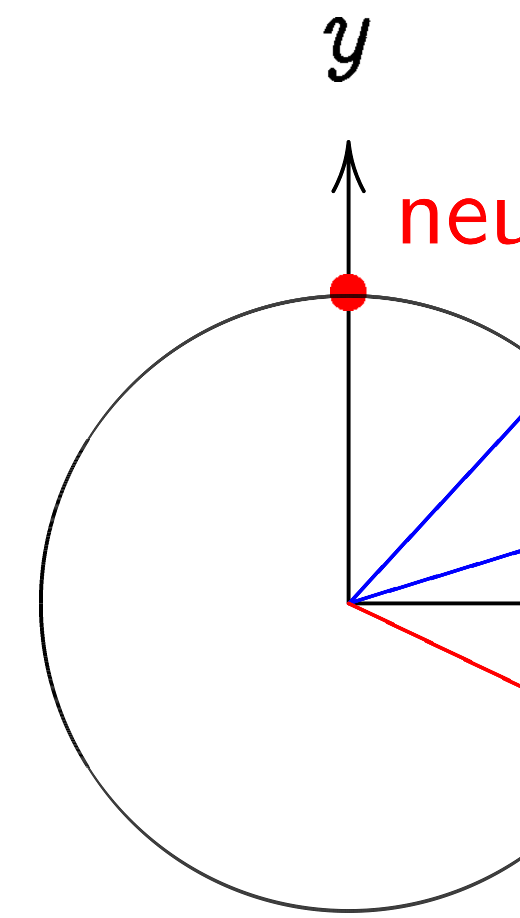
/5, −3/5).

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 wit



Use Cartesian coor

addition. Addition

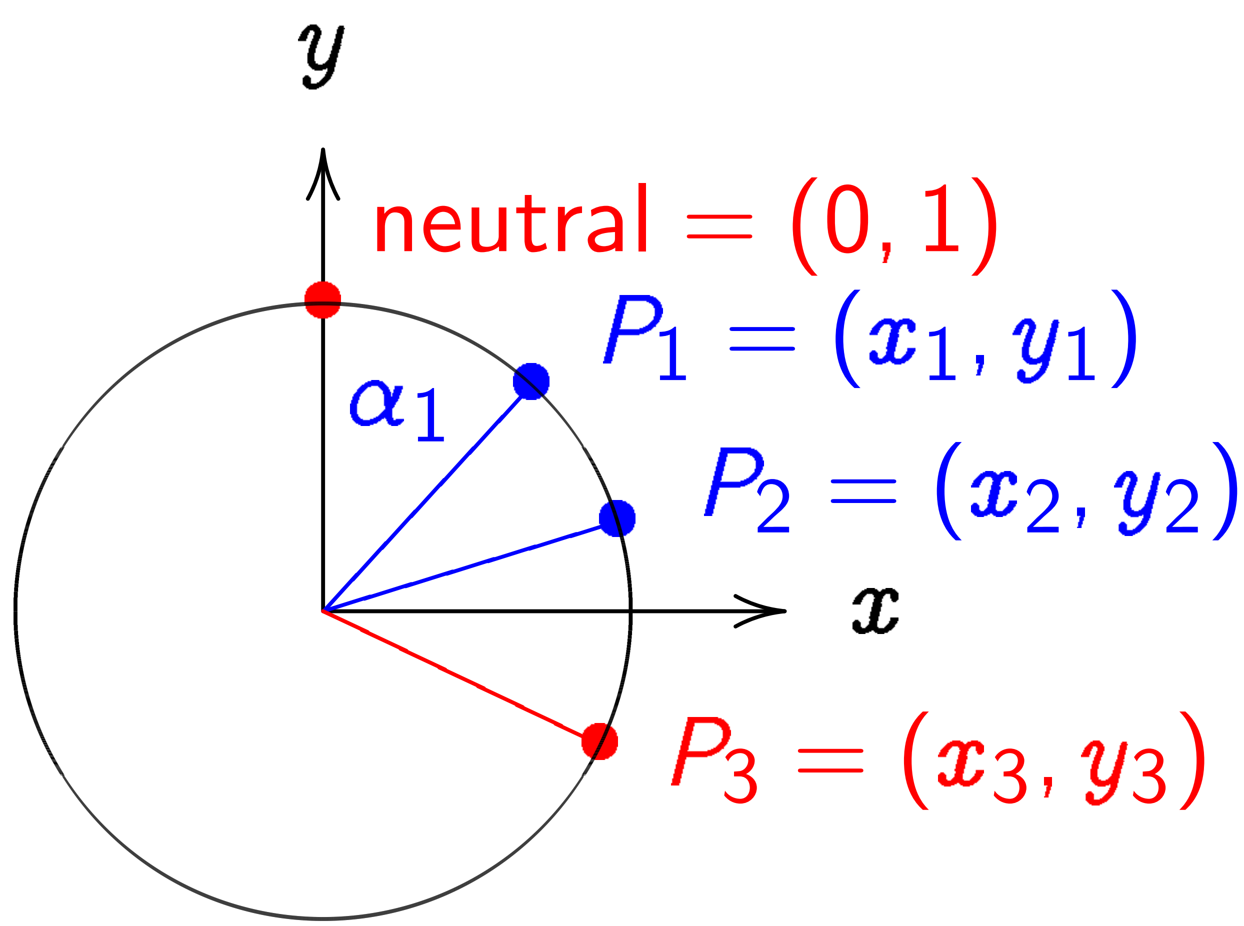
for the clock $x^2 +$

sum of (x_1, y_1) an

$(x_1 y_2 + y_1 x_2, y_1 y$

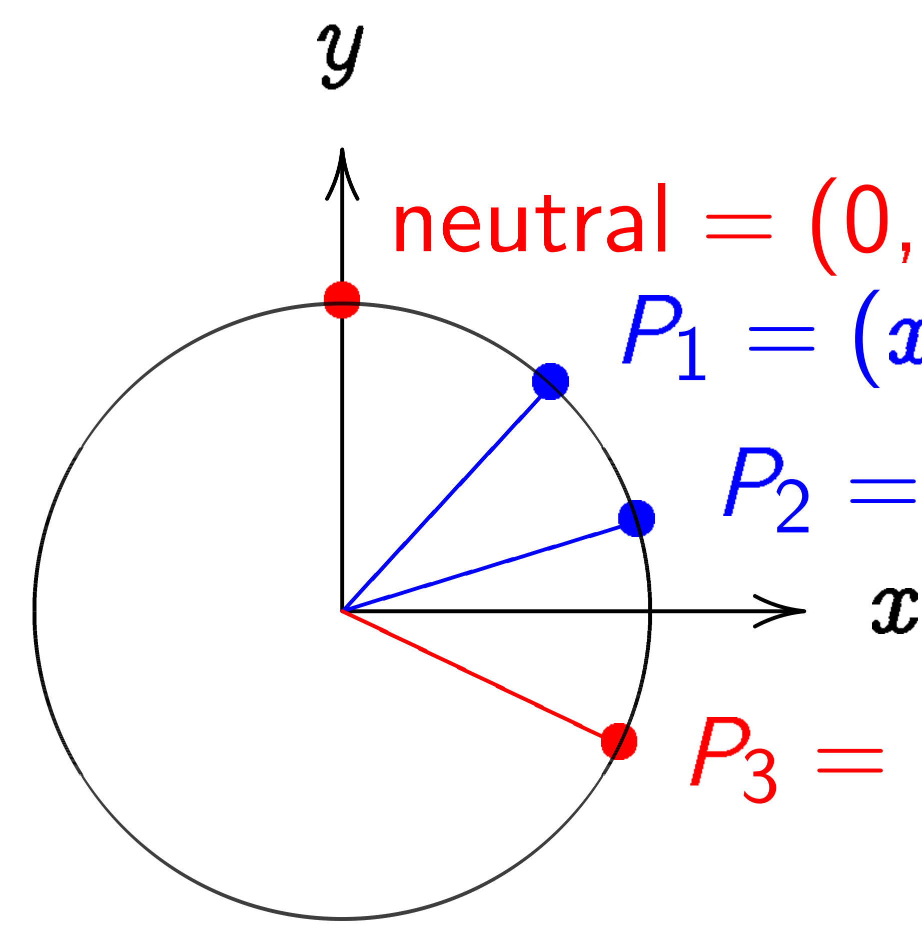
curve:

Addition on the clock:



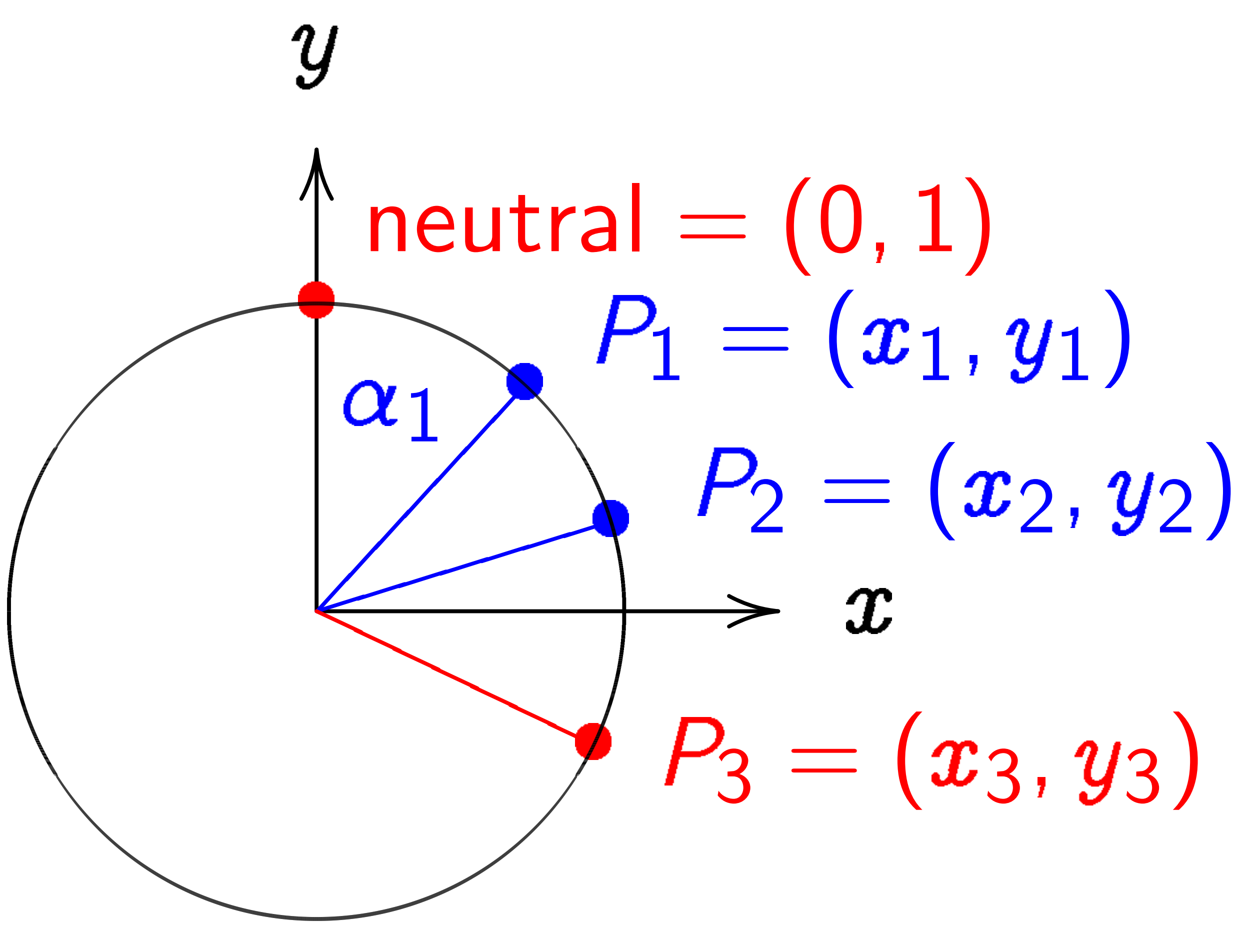
$x^2 + y^2 = 1$, parametrized by
$x = \sin\alpha$, $y = \cos\alpha$. Recall
$(\sin(\alpha_1 + \alpha_2), \cos(\alpha_1 + \alpha_2)) =$
$(\sin\alpha_1 \cos\alpha_2 + \cos\alpha_1 \sin\alpha_2,$
$\cos\alpha_1 \cos\alpha_2 - \sin\alpha_1 \sin\alpha_2)$.

Clock addition without sin, c



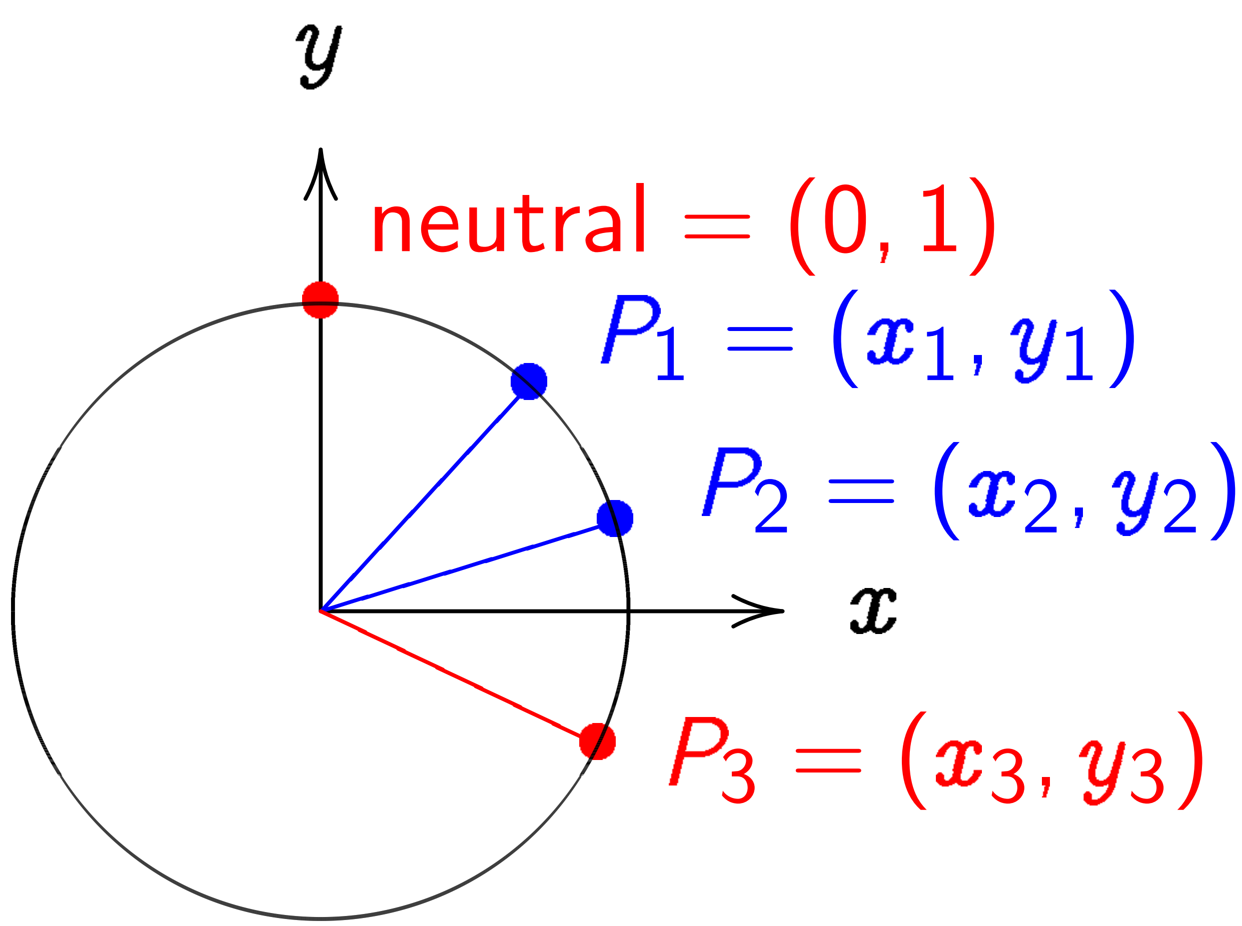
Use Cartesian coordinates fo
addition. Addition formula
for the clock $x^2 + y^2 = 1$:
sum of (x_1, y_1) and (x_2, y_2)
$(x_1 y_2 + y_1 x_2, y_1 y_2 - x_1 x_2)$.

Addition on the clock:



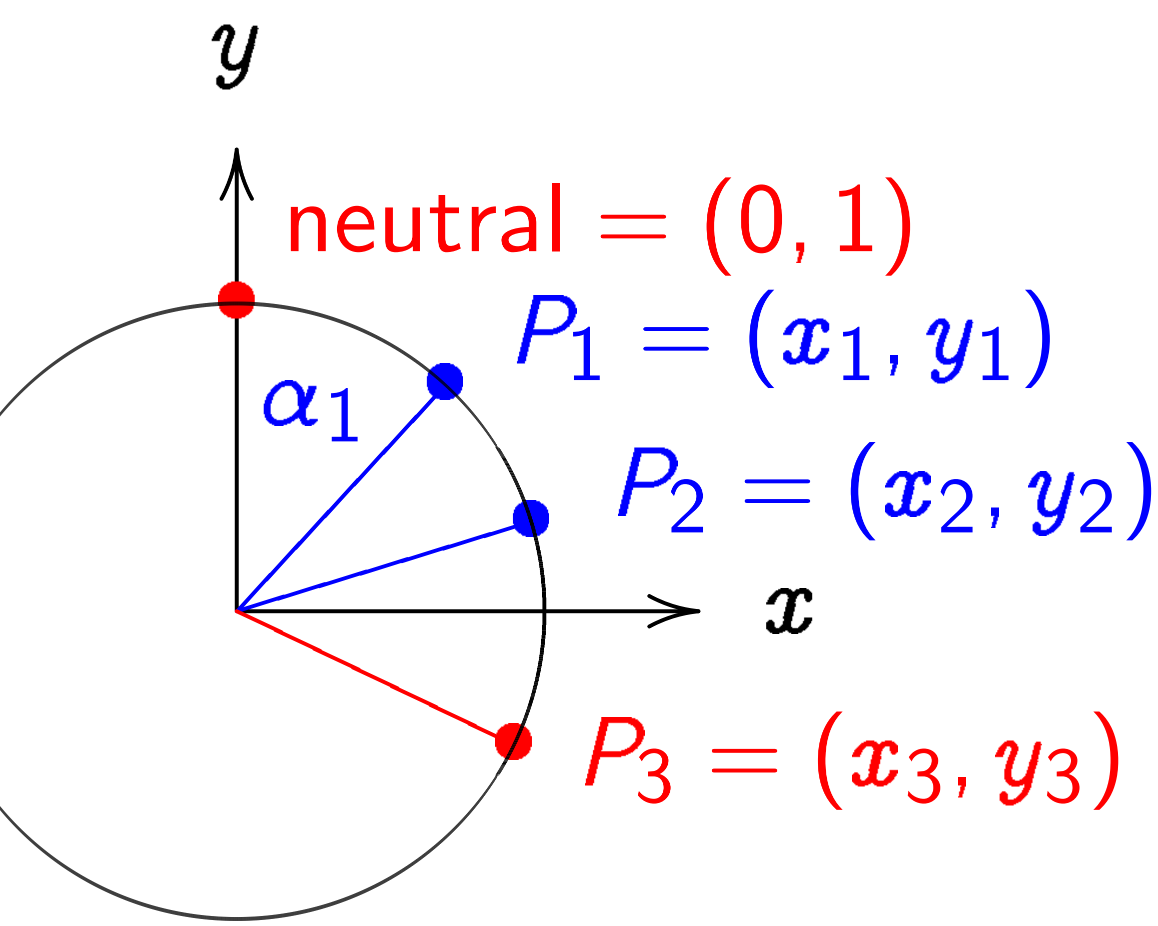
$x^2 + y^2 = 1$, parametrized by $x = \sin\alpha$, $y = \cos\alpha$. Recall $(\sin(\alpha_1 + \alpha_2), \cos(\alpha_1 + \alpha_2)) = (\sin\alpha_1 \cos\alpha_2 + \cos\alpha_1 \sin\alpha_2, \cos\alpha_1 \cos\alpha_2 - \sin\alpha_1 \sin\alpha_2)$.

Clock addition without sin, cos:



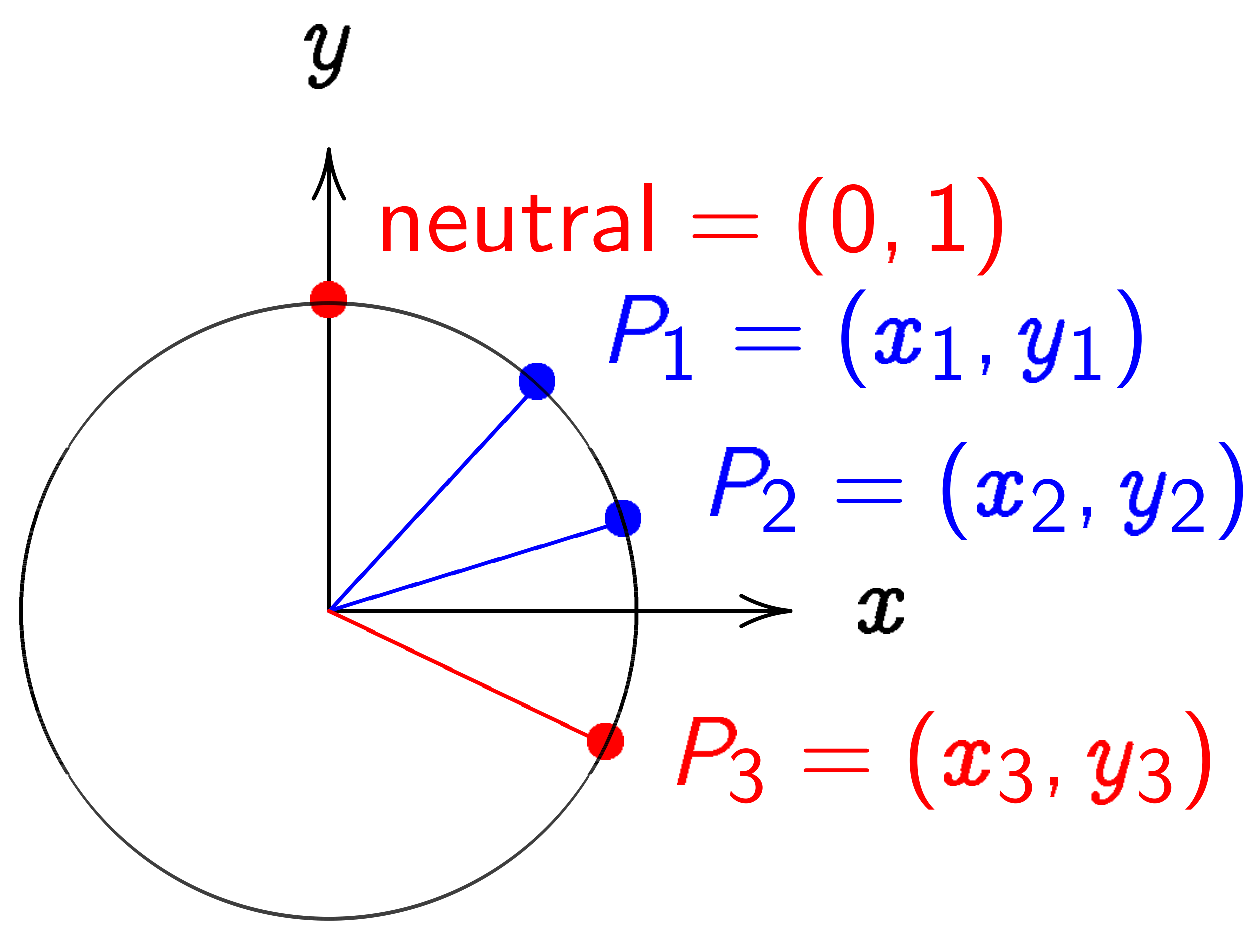
Use Cartesian coordinates for addition. Addition formula for the clock $x^2 + y^2 = 1$: sum of (x_1, y_1) and (x_2, y_2) is $(x_1 y_2 + y_1 x_2, y_1 y_2 - x_1 x_2)$.

n on the clock:



$= 1$, parametrized by
α, $y = \cos\alpha$. Recall
$+ \alpha_2), \cos(\alpha_1 + \alpha_2)) =$
$os\, \alpha_2 + \cos\alpha_1 \sin\alpha_2,$
$\cos\alpha_2 - \sin\alpha_1 \sin\alpha_2).$

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 $(x_1 y_2 + y_1 x_2, y_1 y_2 - x_1 x_2)$.

Example

“2:00” +

$= (\sqrt{3/$

$= (-1/2$

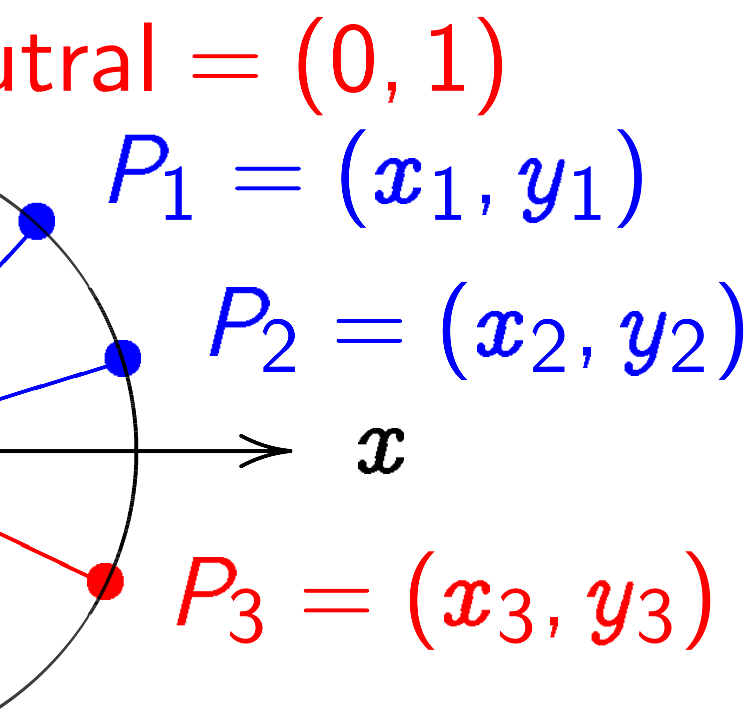
“5:00” +

$= (1/2,$

$= (\sqrt{3/$

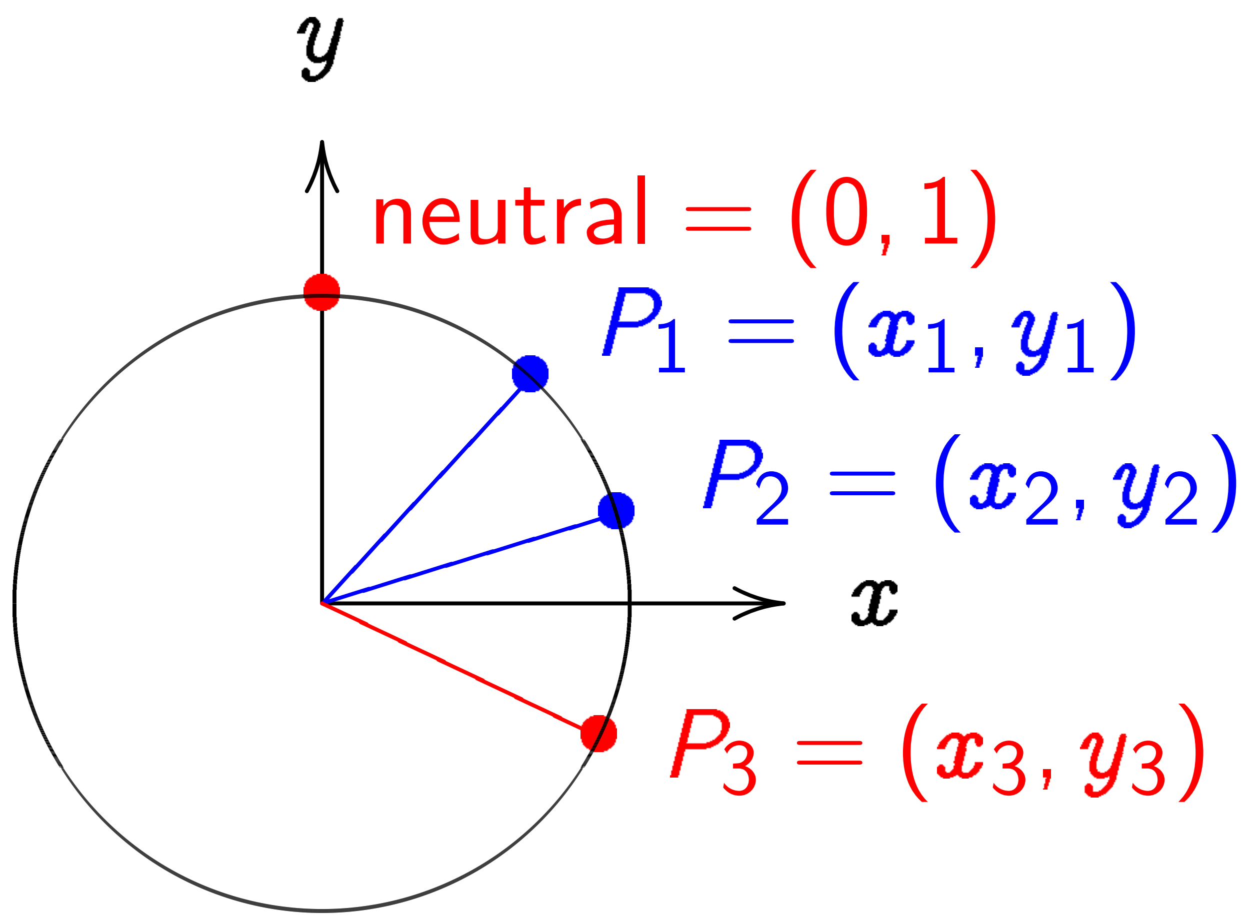
$2\left(\frac{3}{5}, \frac{4}{5}\right.$

ock:



metrized by
s α. Recall
$(\alpha_1 + \alpha_2)) =$
os $\alpha_1 \sin \alpha_2$,
n $\alpha_1 \sin \alpha_2)$.

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

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

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

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

1)

$x_1, y_1)$

(x_2, y_2)

(x_3, y_3)

by

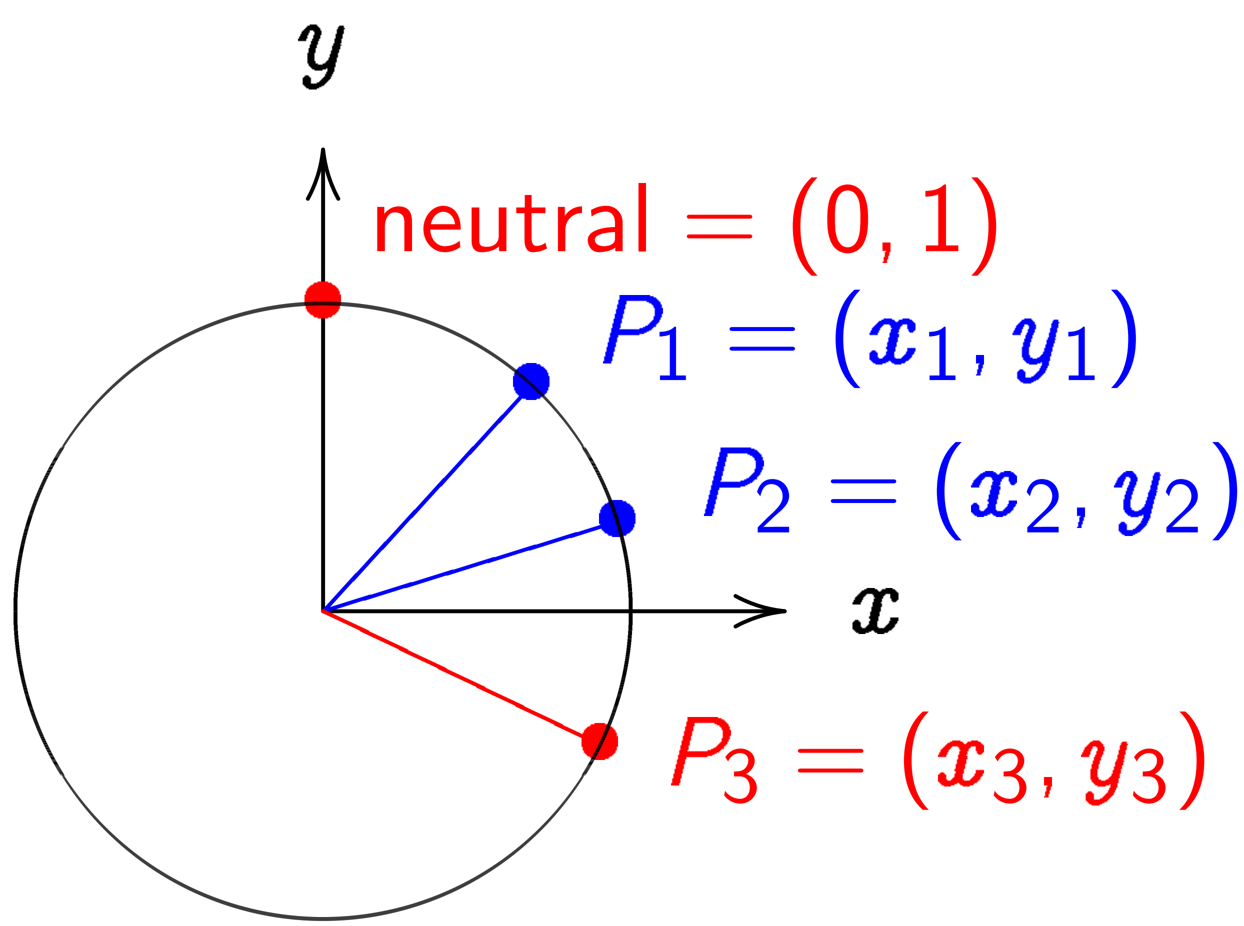
ll

$) =$

$2,$

$2).$

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:

“2:00” + “5:00”

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

$= (-1/2, -\sqrt{3/4}) =$ “7:00”

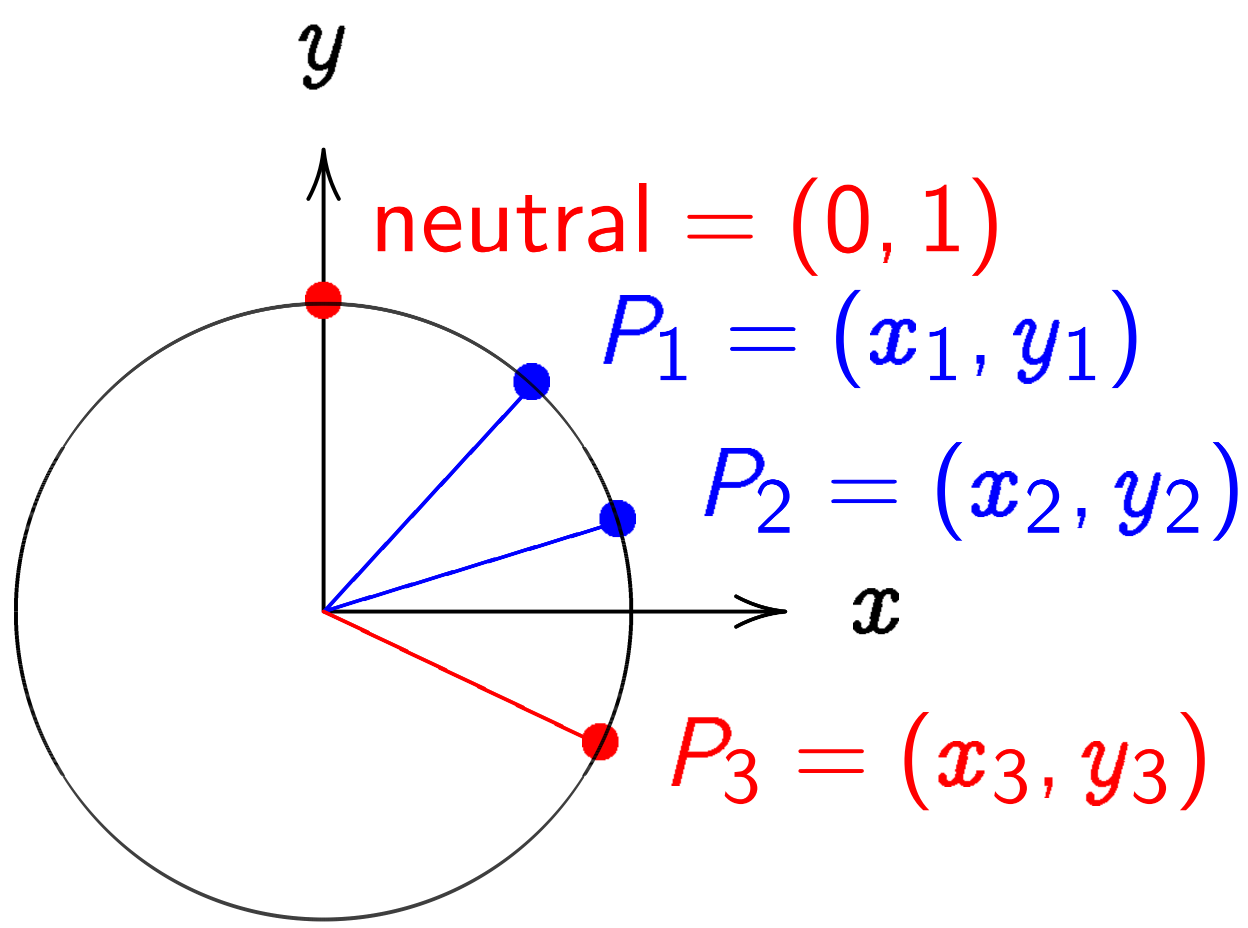
“5:00” + “9:00”

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

$= (\sqrt{3/4}, 1/2) =$ “2:00”.

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

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 $(x_1y_2 + y_1x_2, y_1y_2 - x_1x_2)$.

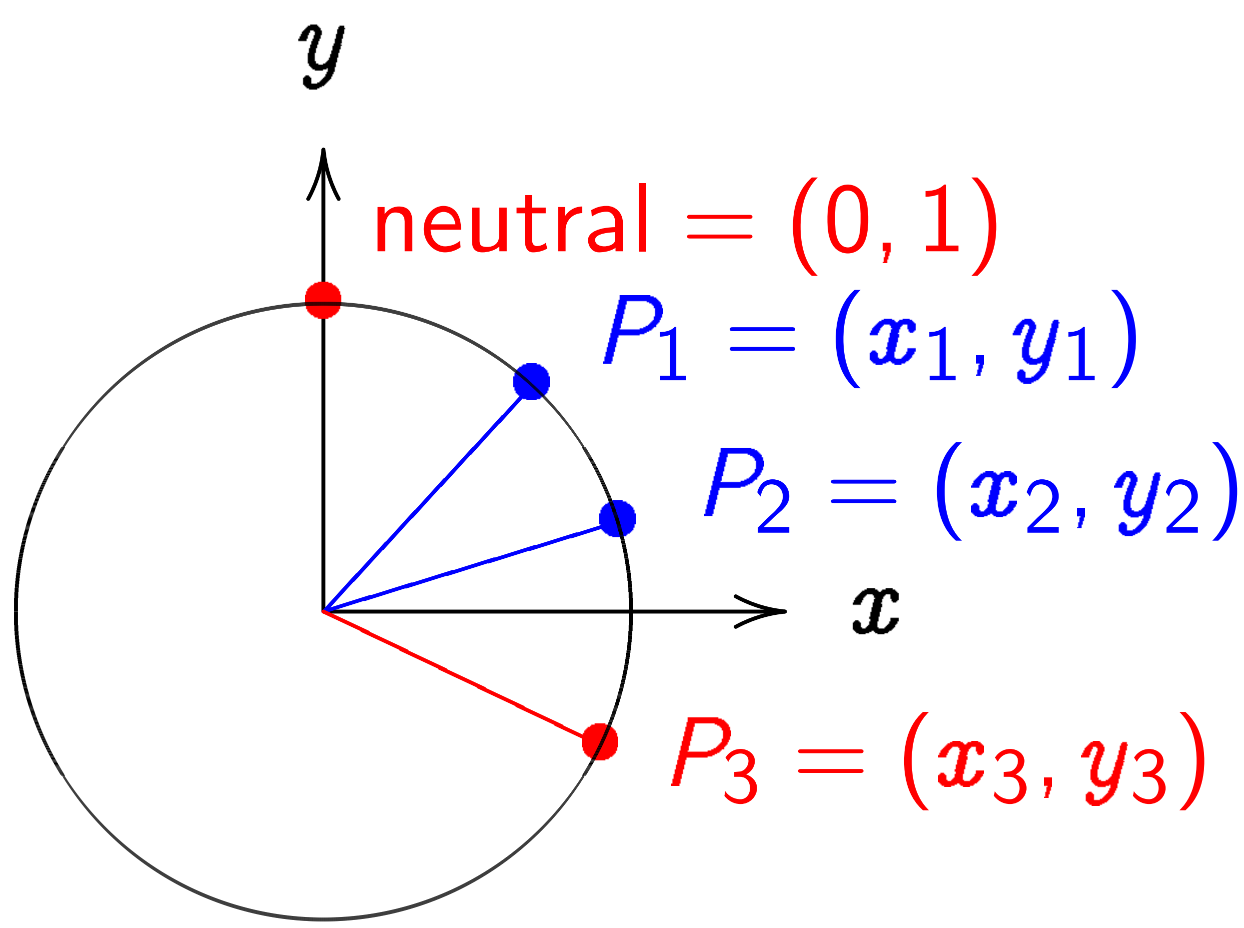
Examples of clock addition:

"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".

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

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:

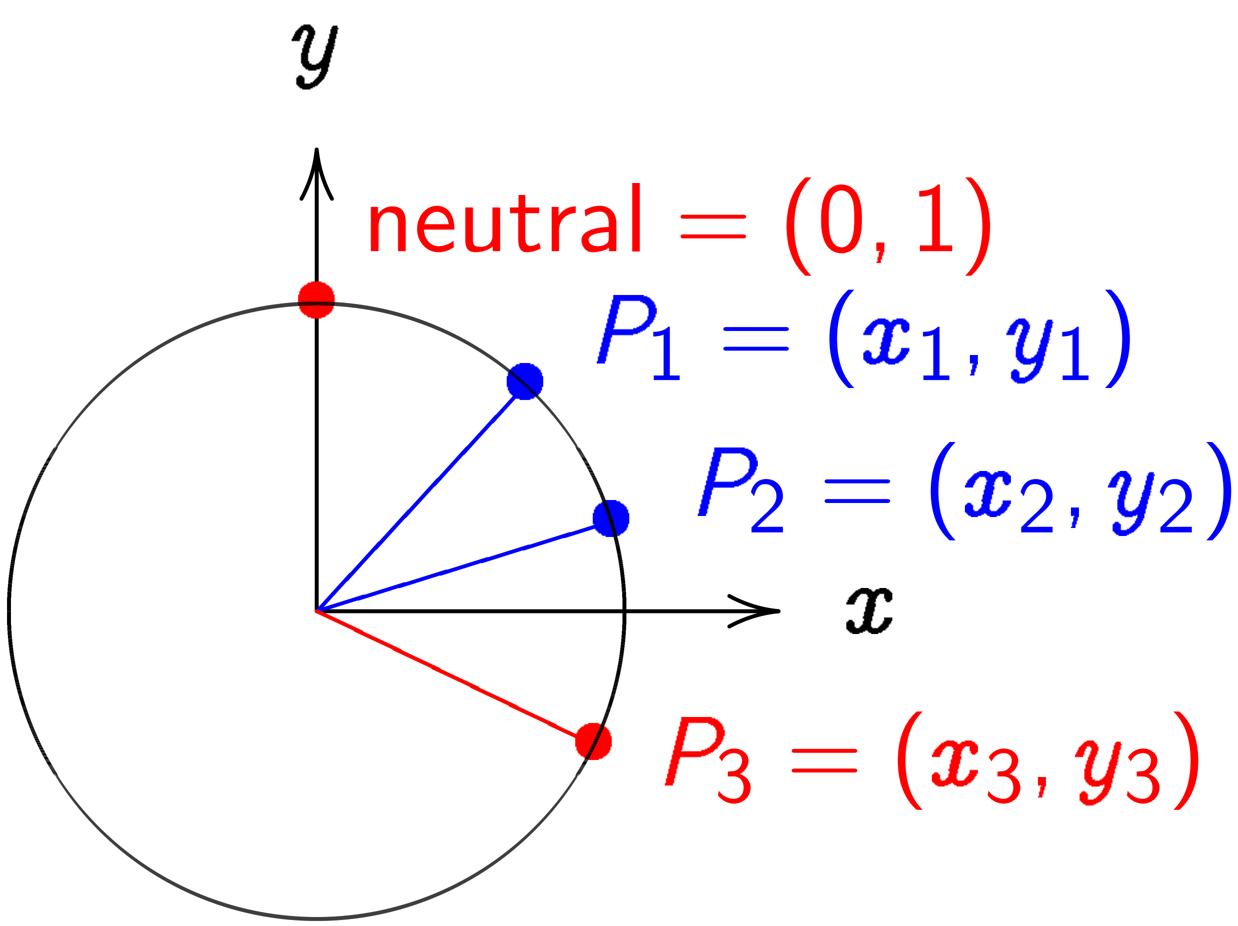
"2:00" + "5:00"
$= (\sqrt{3/4}, 1/2) + (1/2, -\sqrt{3/4})$
$= (-1/2, -\sqrt{3/4}) =$ "7:00".

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

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

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

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 $(x_1y_2 + y_1x_2, y_1y_2 - x_1x_2)$.

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".

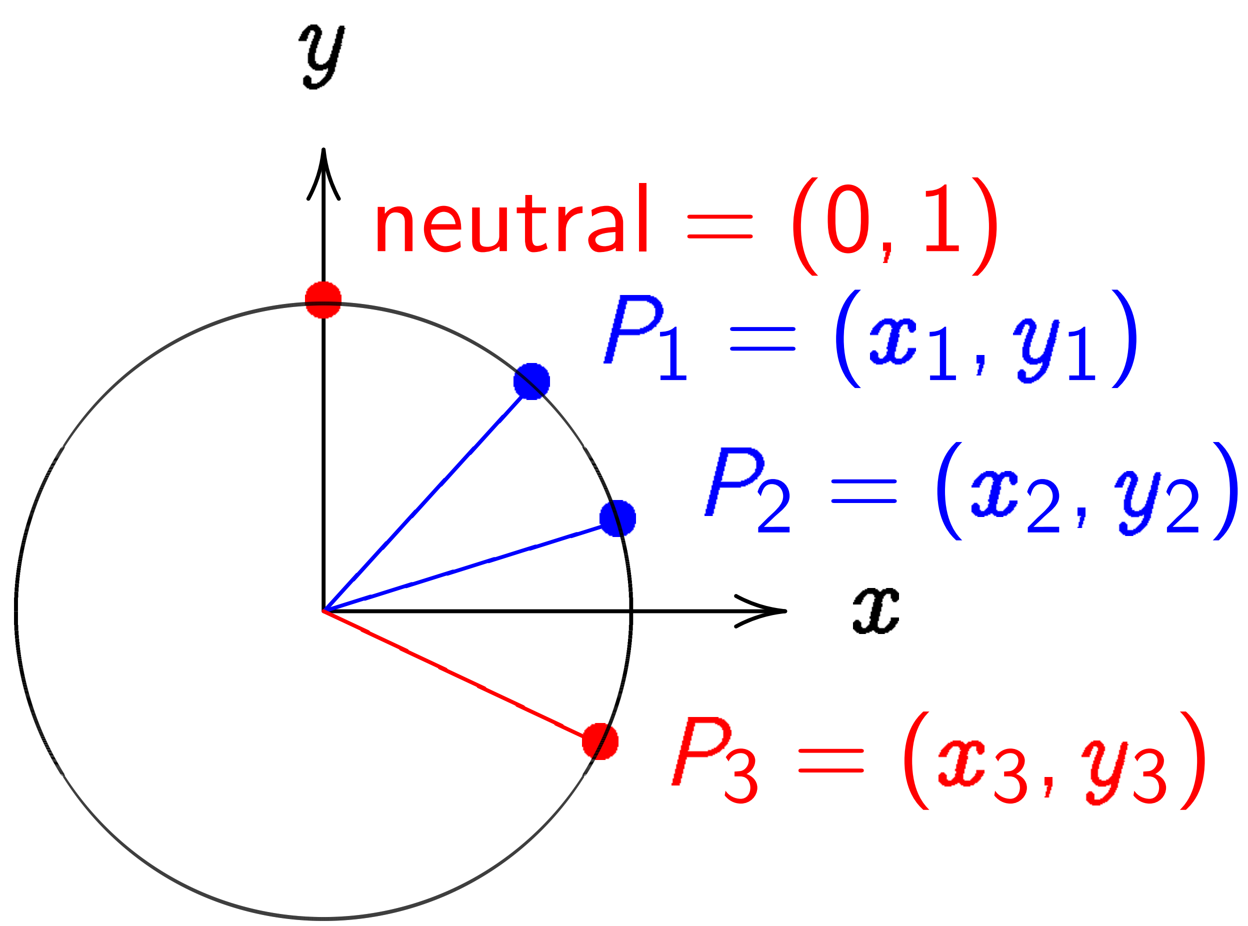
"5:00" + "9:00"
$= (1/2, -\sqrt{3/4}) + (-1, 0)$
$= (\sqrt{3/4}, 1/2) =$ "2:00".

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

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

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

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:

"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".

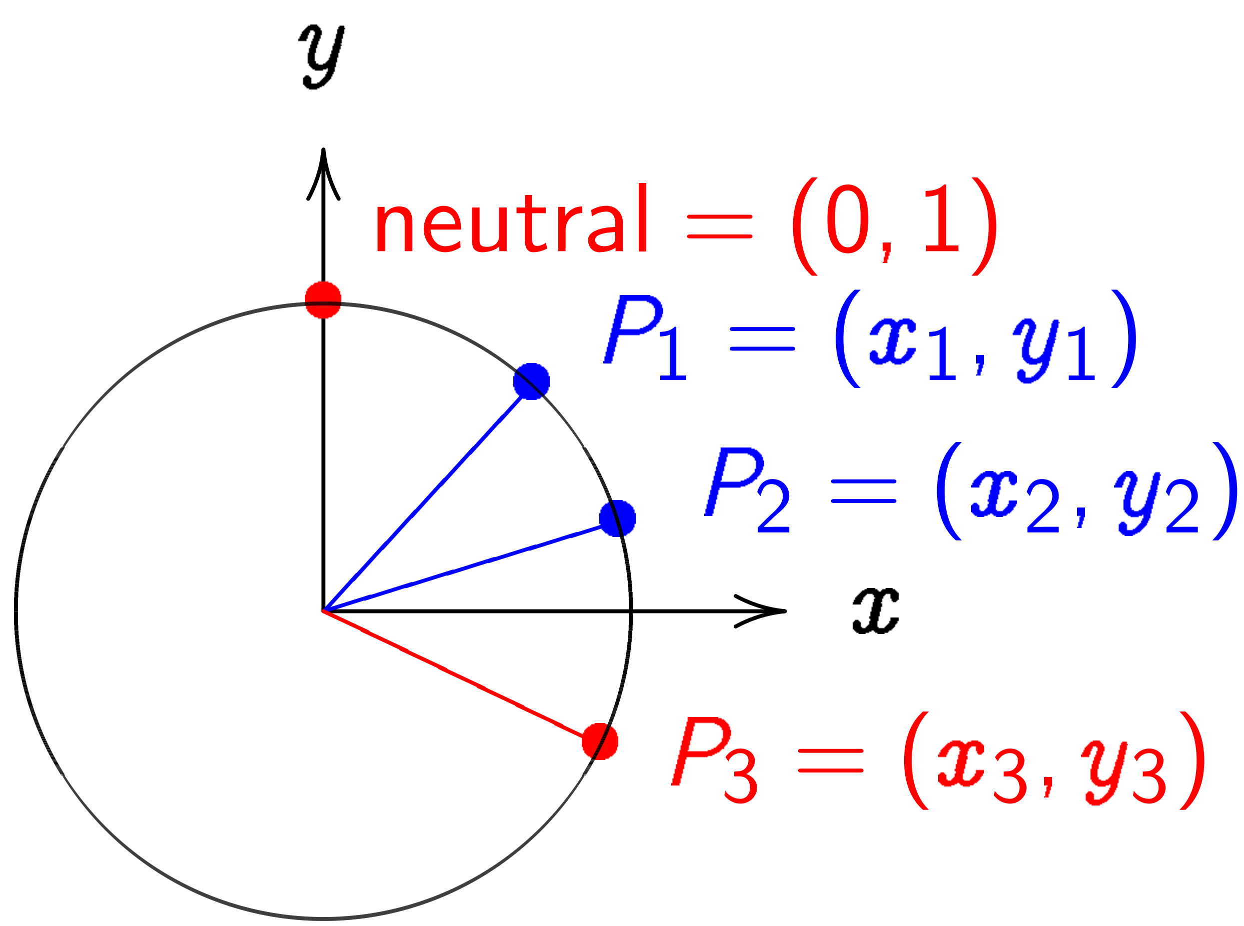
$$2\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{24}{25}, \frac{7}{25}\right).$$

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

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

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

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 $(x_1y_2 + y_1x_2, y_1y_2 - x_1x_2)$.

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”.

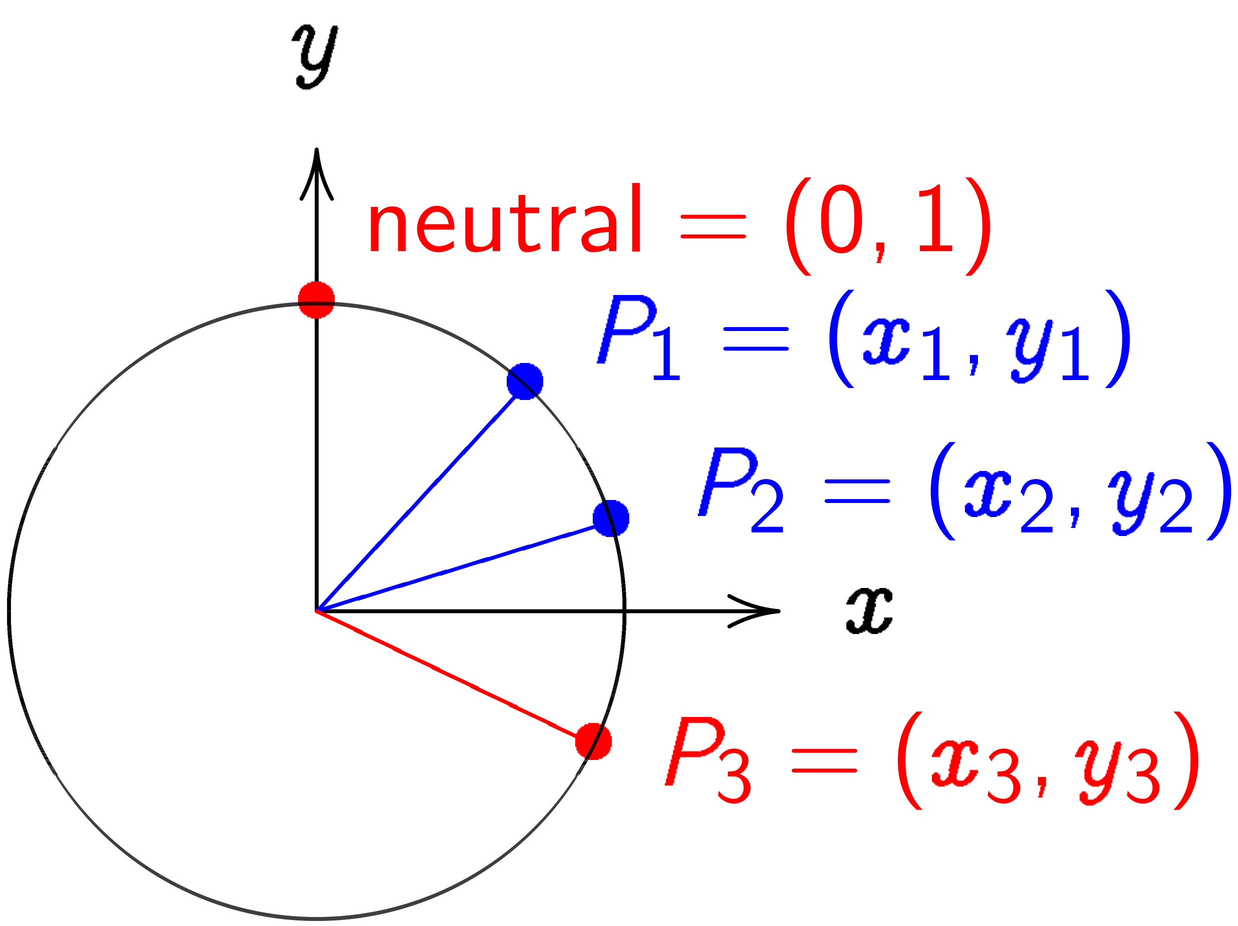
$$2\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{24}{25}, \frac{7}{25}\right).$$

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

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

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

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:

"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".

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

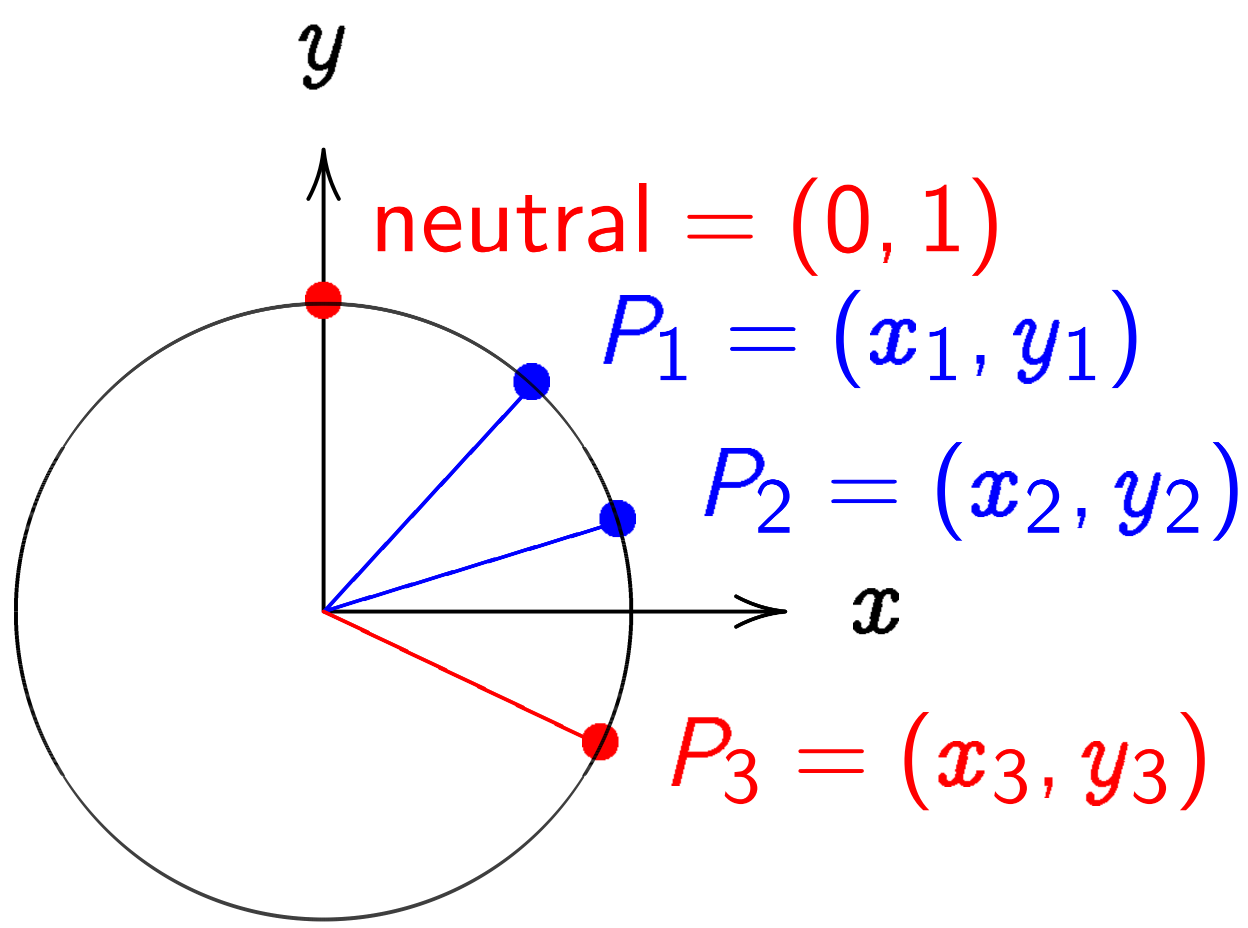
$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

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

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

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

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 $(x_1y_2 + y_1x_2, y_1y_2 - x_1x_2)$.

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".

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

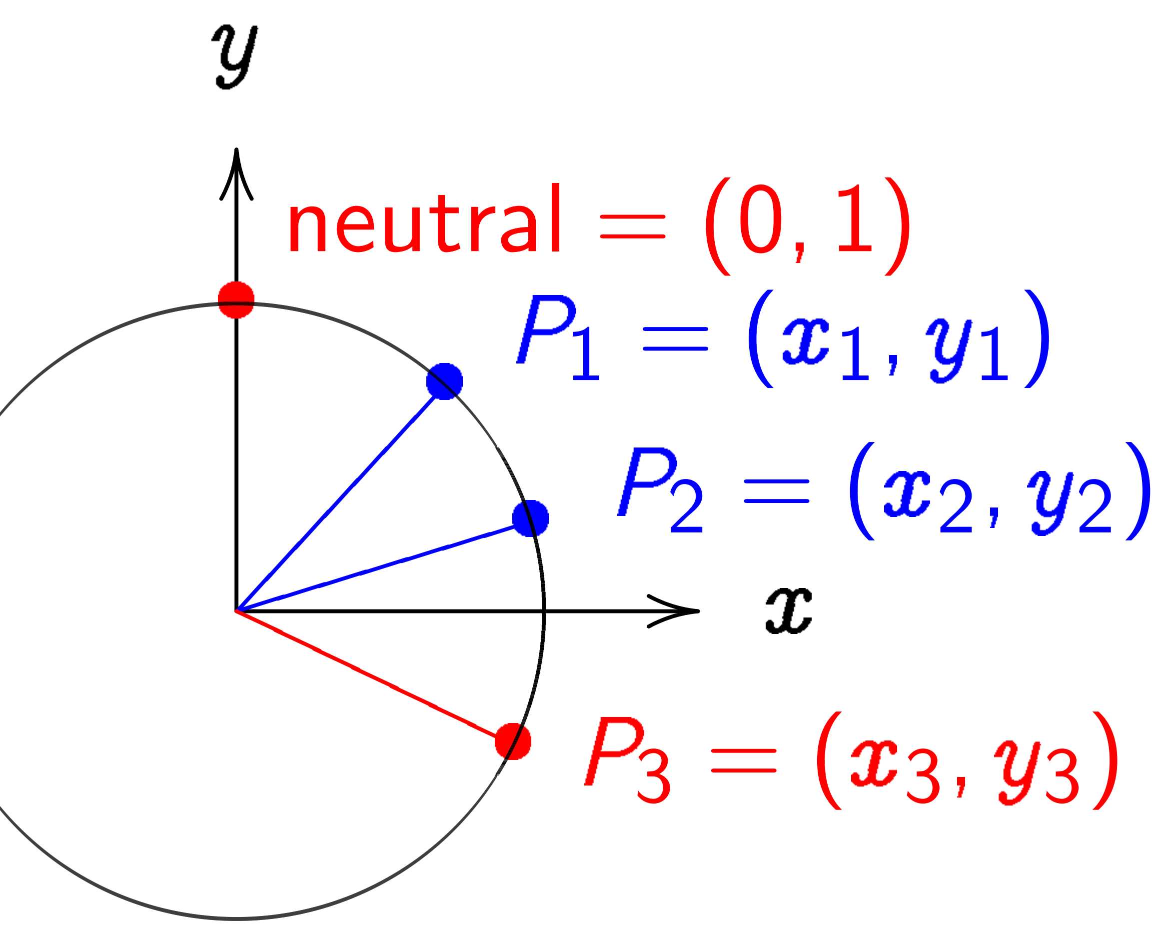
$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

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

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

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

ddition without sin, cos:



tesian coordinates for
. Addition formula
clock $x^2 + y^2 = 1$:
(x_1, y_1) and (x_2, y_2) is
$y_1 x_2, y_1 y_2 - x_1 x_2)$.

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".

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

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

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

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

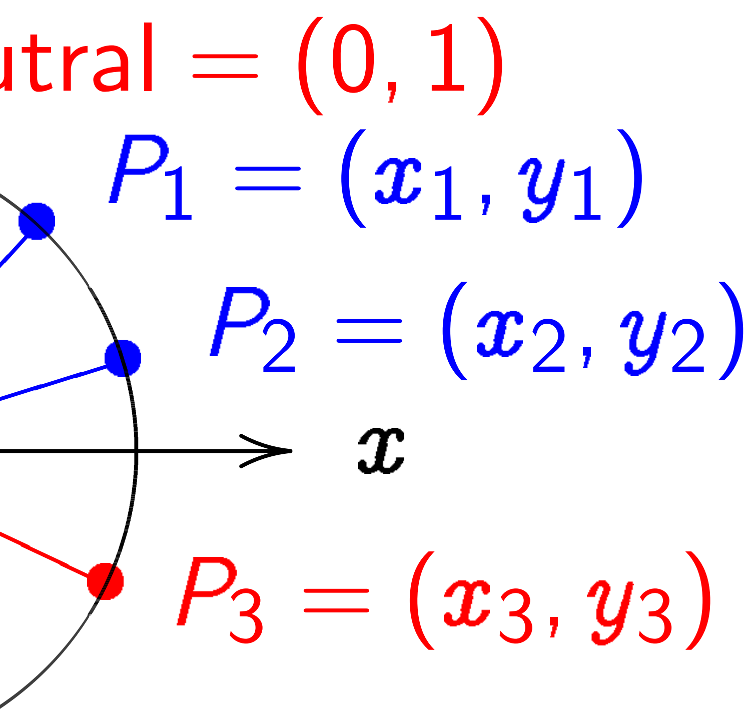
$(x_1, y_1) + (-x_1, y_1) = (0, 1)$.

Clocks o

.
.
.
.
.
.
.

Clock($\mathbf{F}$
$\{(x, y) \in$
Here $\mathbf{F}_7$
$= \{0, 1,$
with arit
e.g. $2 \cdot 5$

thout sin, cos:



rdinates for
formula
$y^2 = 1$:
nd (x_2, y_2) is
$2 - x_1 x_2)$.

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".

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

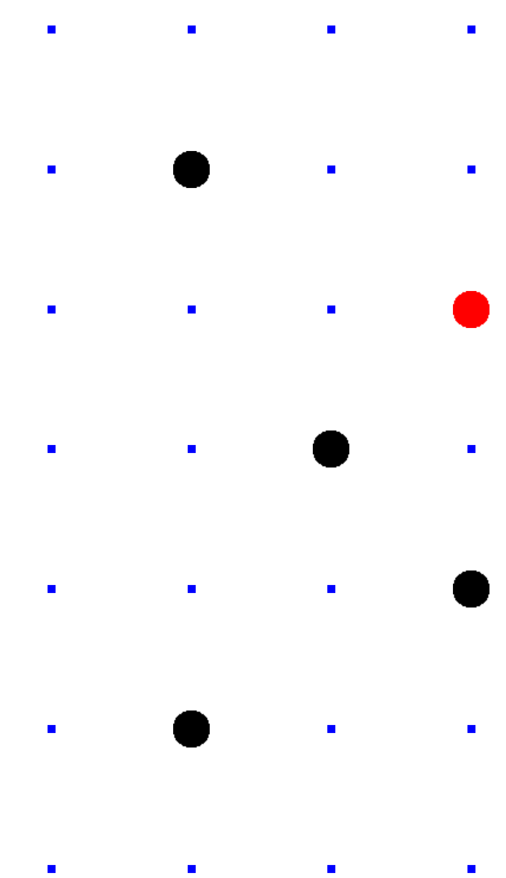
$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

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

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

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

Clocks over finite



$\text{Clock}(\mathbf{F}_7) =$
$\{(x, y) \in \mathbf{F}_7 \times \mathbf{F}_7$
Here $\mathbf{F}_7 = \{0, 1, 2$
$= \{0, 1, 2, 3, -3, -$
with arithmetic m
e.g. $2 \cdot 5 = 3$ and

cos:

$1)$

$x_1, y_1)$

(x_2, y_2)

(x_3, y_3)

or

is

.

Examples of clock addition:

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

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

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

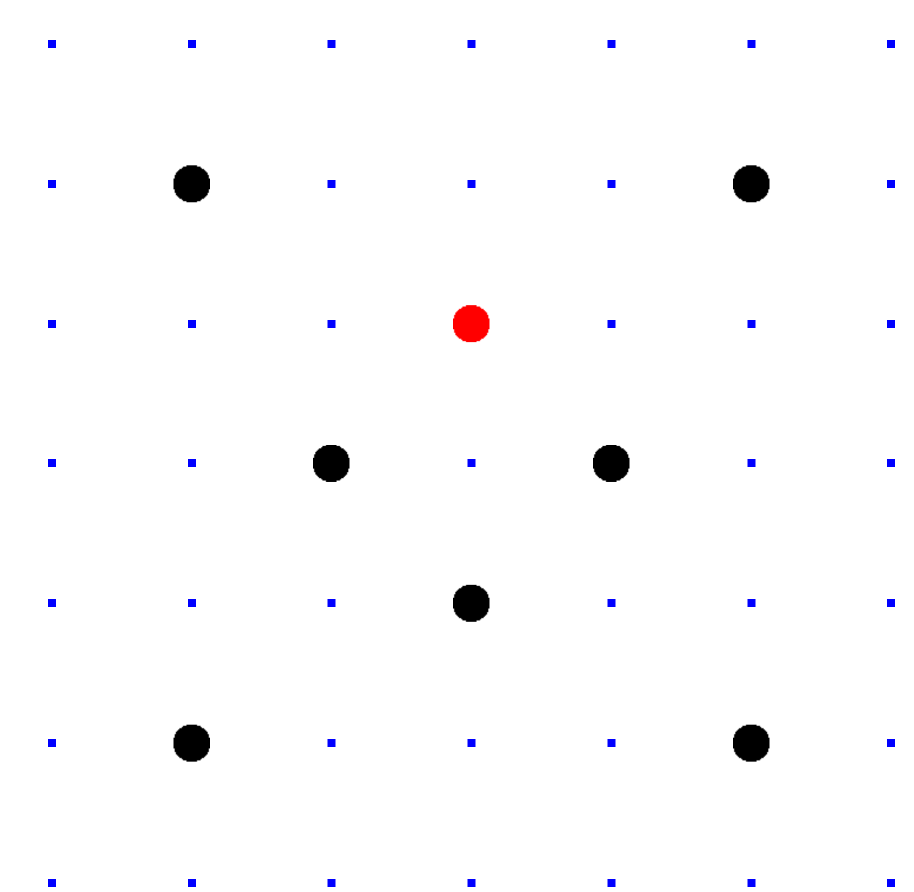
$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

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

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

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

Clocks over finite fields



$\text{Clock}(\mathbf{F}_7) =$
$\{(x, y) \in \mathbf{F}_7 \times \mathbf{F}_7 : x^2 + y^2$

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 $3/2 = 5$ i

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"

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

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

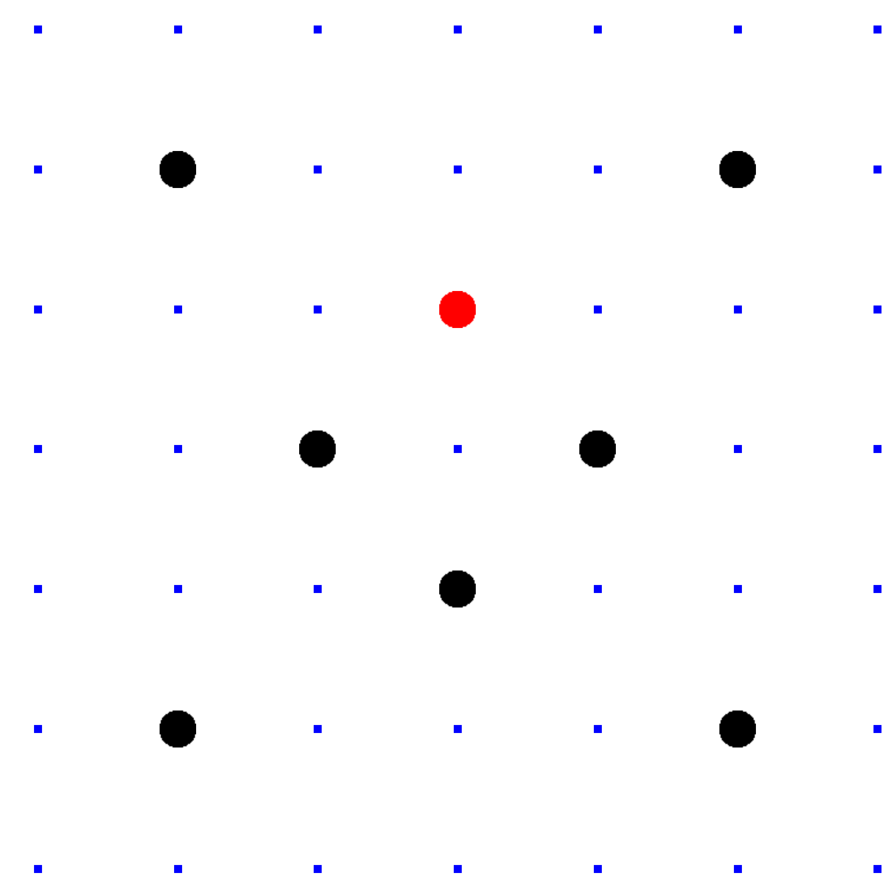
$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

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

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

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

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 $\mathbf{F}_7$.

es of clock addition:

– “5:00”

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

$2, -\sqrt{3/4}) =$ “7:00”.

– “9:00”

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

$4, 1/2) =$ “2:00”.

$) = \left(\frac{24}{25}, \frac{7}{25}\right).$

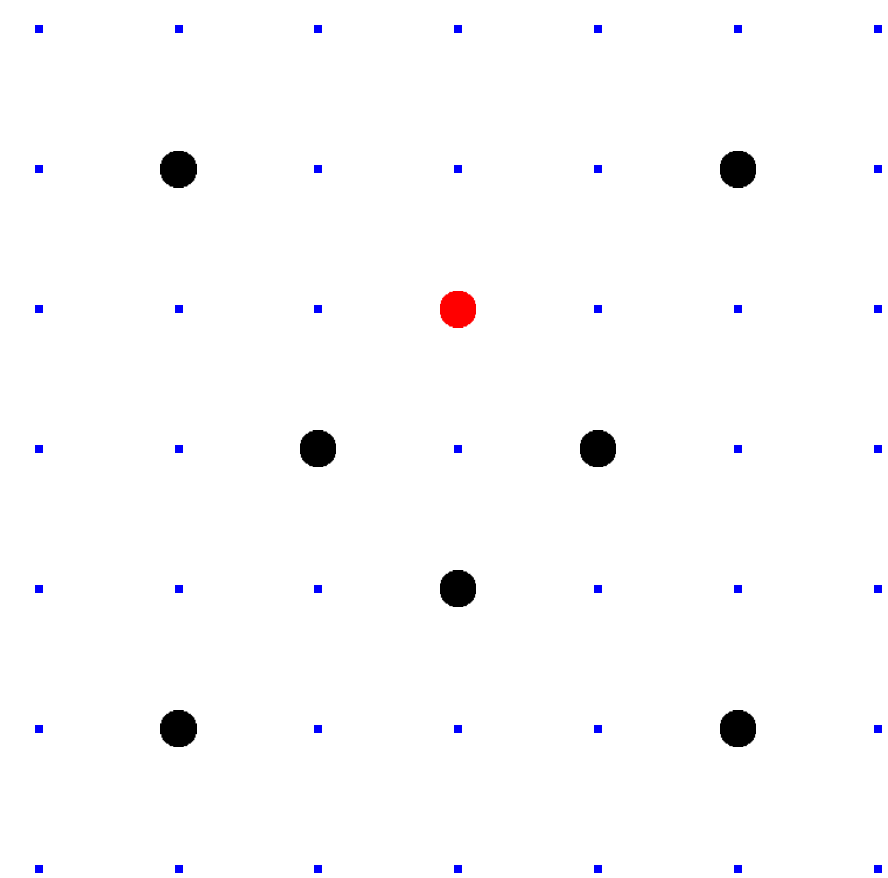
$) = \left(\frac{117}{125}, \frac{-44}{125}\right).$

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

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

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

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 e

Example

on Clock

2(1000, 2

addition:

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

$) =$ "7:00".

$+ (-1, 0)$

"2:00".

$\frac{7}{25}\Big)$.

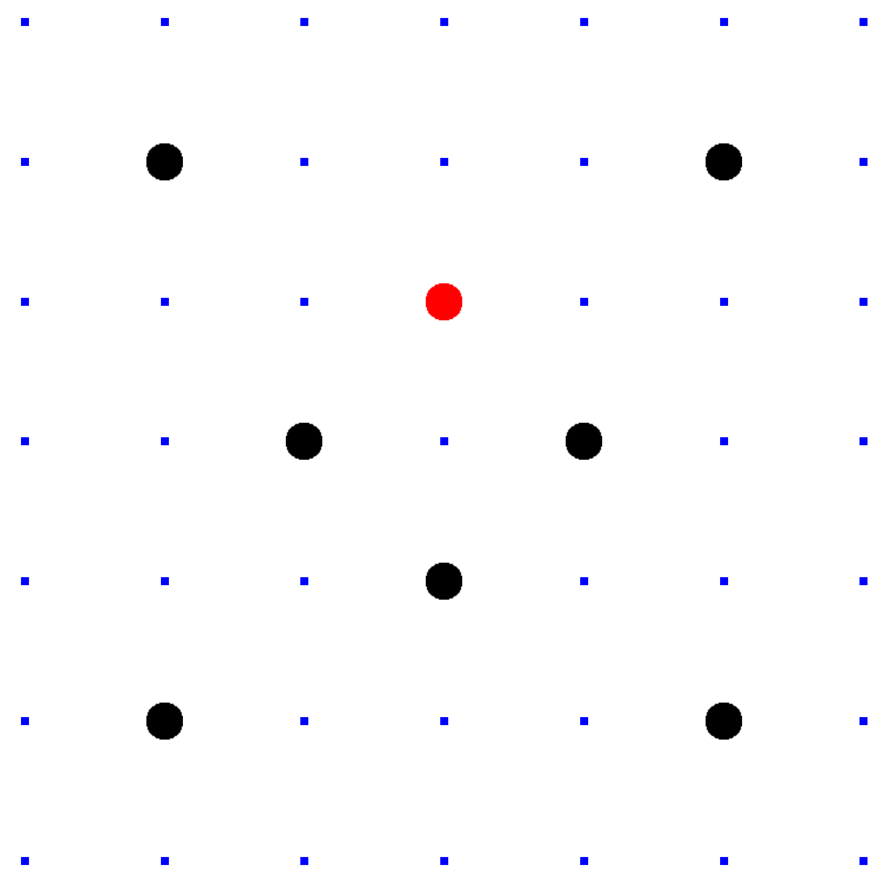
$, \frac{-44}{125}\Big)$.

$, \frac{-527}{625}\Big)$.

$= (x_1, y_1)$.

$_1) = (0, 1)$.

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 $\mathbf{F}_7$.

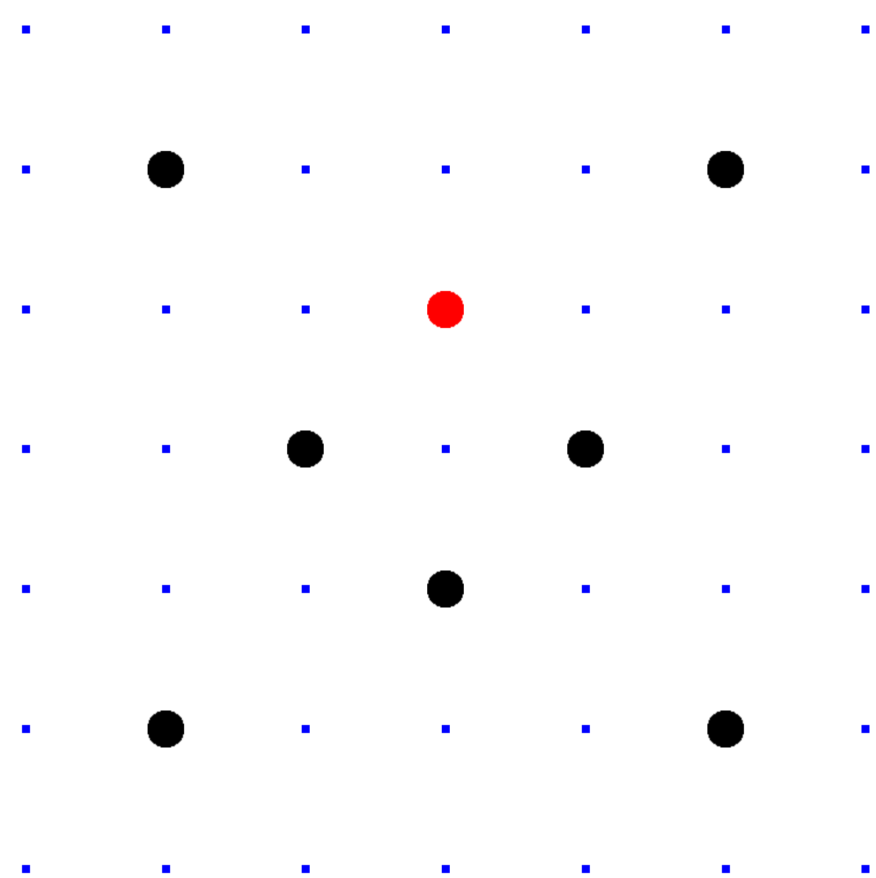
Larger example: C

Examples of addit
on $\text{Clock}(\mathbf{F}_{1000003}$
$2(1000, 2) = (4000$

$\overline{}$
3/4)
.

).

Clocks over finite fields

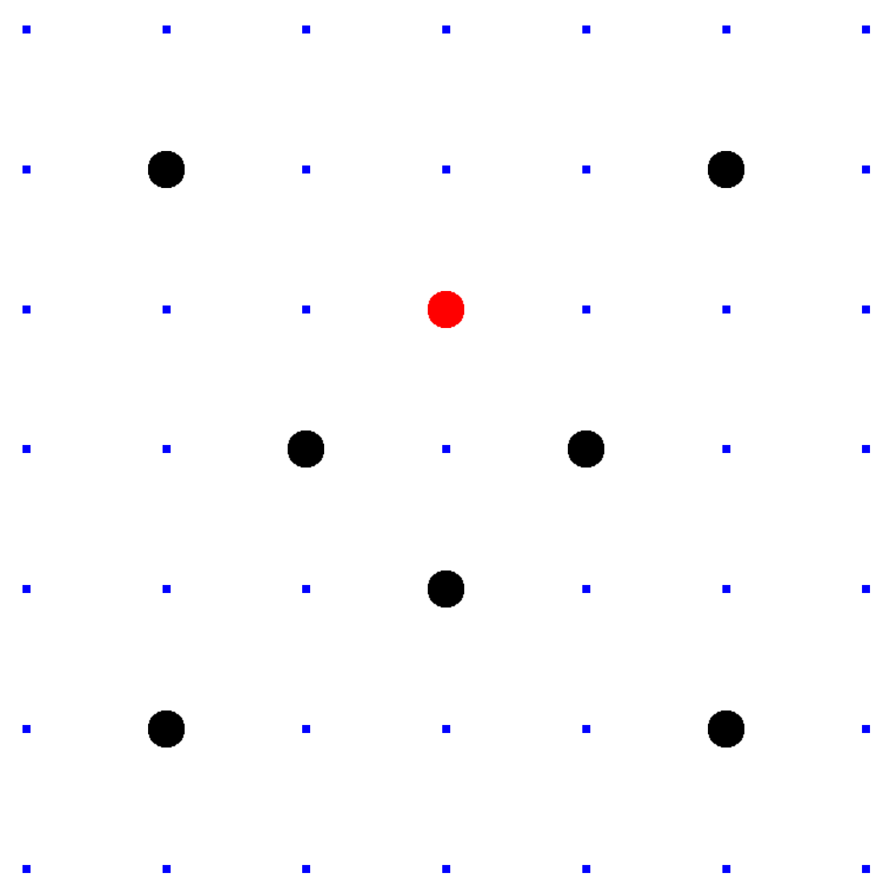


$\text{Clock}(\mathbf{F}_7) =$
$\{(x, y) \in \mathbf{F}_7 \times \mathbf{F}_7 : x^2 + y^2 = 1\}$.
Here $\mathbf{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$
$= \{0, 1, 2, 3, -3, -2, -1\}$
with arithmetic modulo 7.
e.g. $2 \cdot 5 = 3$ and $3/2 = 5$ in $\mathbf{F}_7$.

Larger example: Clock($\mathbf{F}_{100}$

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

Clocks over finite fields

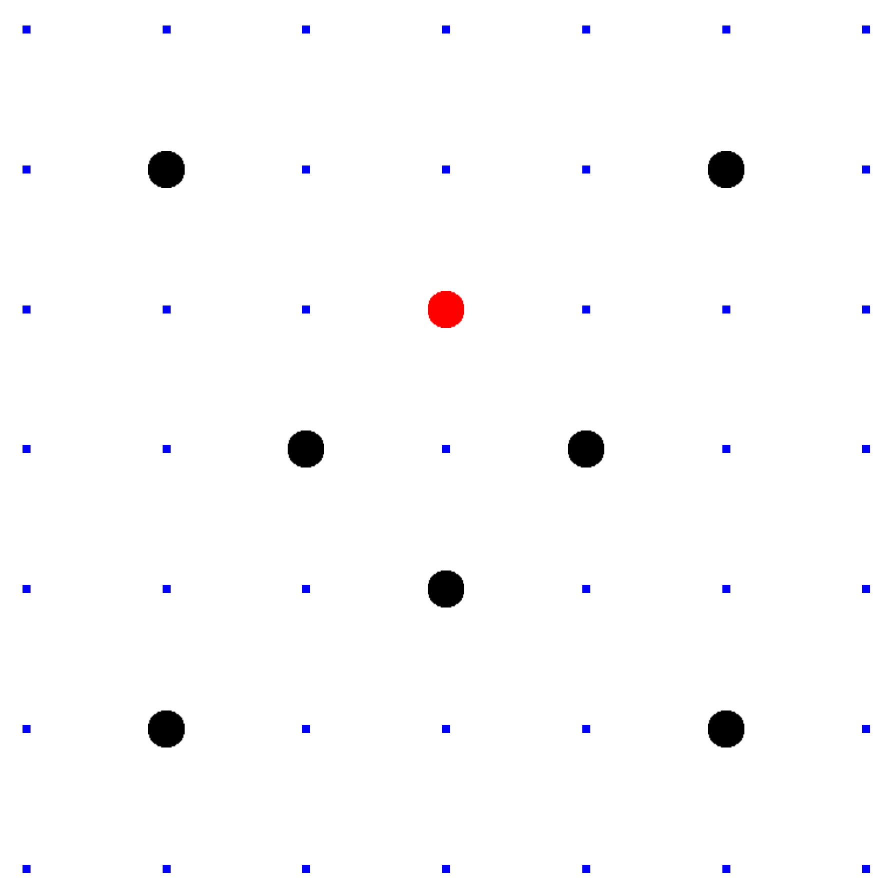


Clock($\mathbf{F}_7$) =
$\{(x, y) \in \mathbf{F}_7 \times \mathbf{F}_7 : x^2 + y^2 = 1\}$.
Here $\mathbf{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$
$= \{0, 1, 2, 3, -3, -2, -1\}$
with arithmetic modulo 7.
e.g. $2 \cdot 5 = 3$ and $3/2 = 5$ in $\mathbf{F}_7$.

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

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

Clocks over finite fields

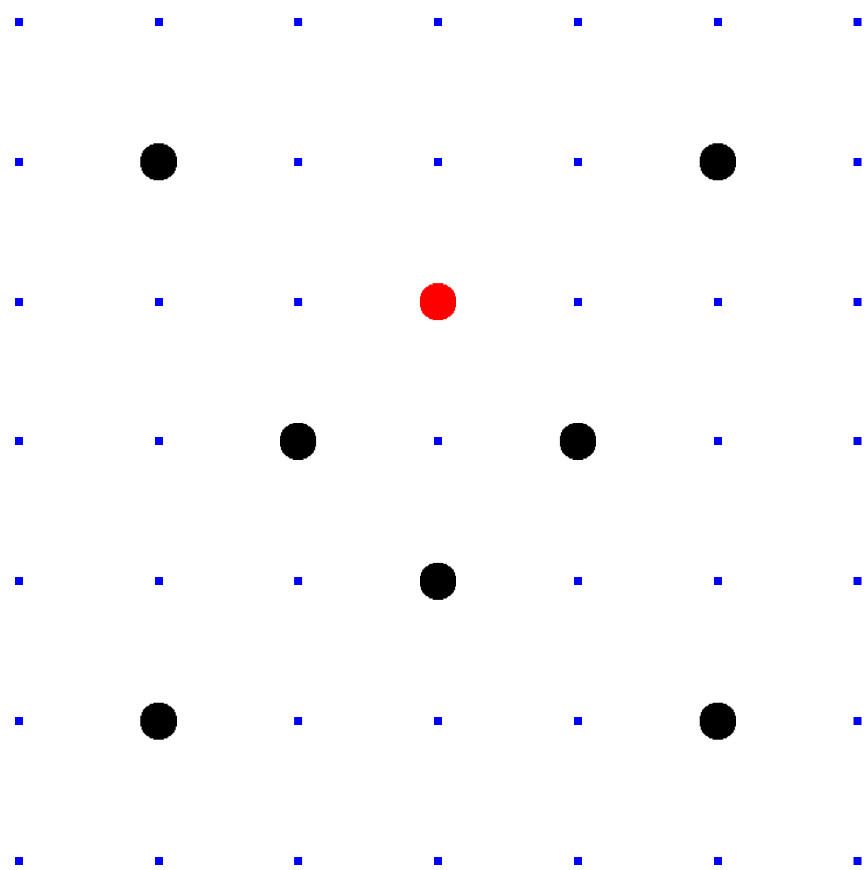


$\text{Clock}(\mathbf{F}_7) =$
$\{(x, y) \in \mathbf{F}_7 \times \mathbf{F}_7 : x^2 + y^2 = 1\}$.
Here $\mathbf{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$
$= \{0, 1, 2, 3, -3, -2, -1\}$
with arithmetic modulo 7.
e.g. $2 \cdot 5 = 3$ and $3/2 = 5$ in $\mathbf{F}_7$.

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

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

Clocks over finite fields

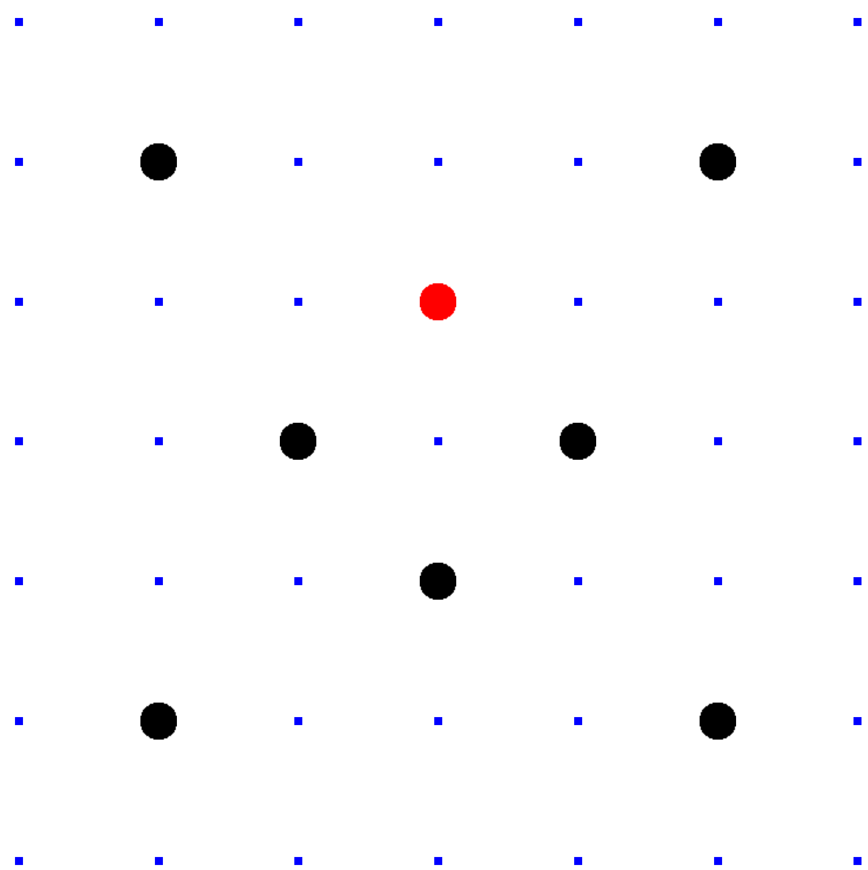


Clock($\mathbf{F}_7$) =
$\{(x, y) \in \mathbf{F}_7 \times \mathbf{F}_7 : x^2 + y^2 = 1\}$.
Here $\mathbf{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$
$= \{0, 1, 2, 3, -3, -2, -1\}$
with arithmetic modulo 7.
e.g. $2 \cdot 5 = 3$ and $3/2 = 5$ in $\mathbf{F}_7$.

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

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

Clocks over finite fields



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($\mathbf{F}_{1000003}$):

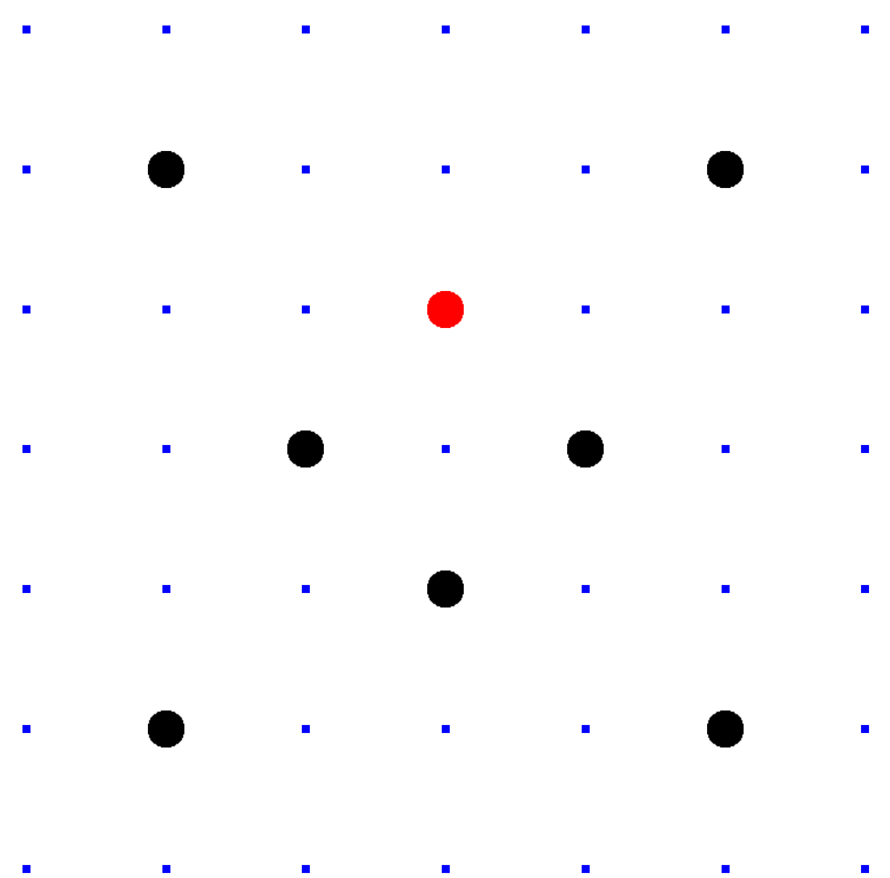
$2(1000, 2) = (4000, 7)$.

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

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

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

Clocks over finite fields

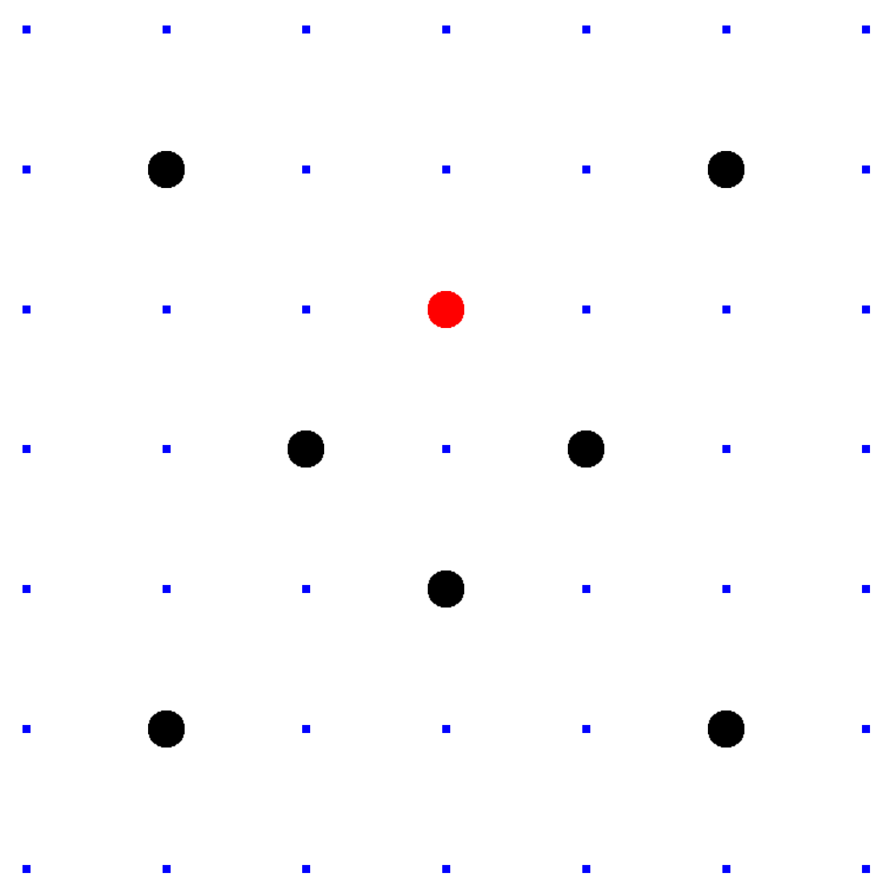


$\text{Clock}(\mathbf{F}_7) =$
$\{(x, y) \in \mathbf{F}_7 \times \mathbf{F}_7 : x^2 + y^2 = 1\}$.
Here $\mathbf{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$
$= \{0, 1, 2, 3, -3, -2, -1\}$
with arithmetic modulo 7.
e.g. $2 \cdot 5 = 3$ and $3/2 = 5$ in $\mathbf{F}_7$.

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

Examples of addition
on $\text{Clock}(\mathbf{F}_{1000003})$:
$2(1000, 2) = (4000, 7)$.
$4(1000, 2) = (56000, 97)$.
$8(1000, 2) = (863970, 18817)$.
$16(1000, 2) = (549438, 156853)$.
$17(1000, 2) = (951405, 877356)$.

Clocks over finite fields



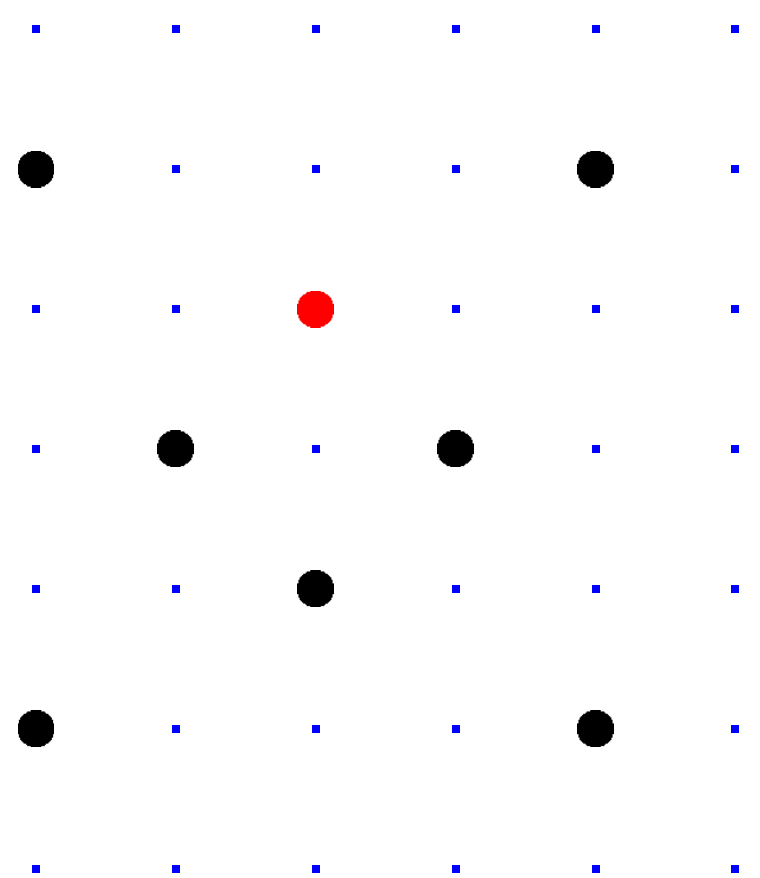
Clock($\mathbf{F}_7$) $=$
$\{(x, y) \in \mathbf{F}_7 \times \mathbf{F}_7 : x^2 + y^2 = 1\}$.
Here $\mathbf{F}_7 = \{0, 1, 2, 3, 4, 5, 6\}$
$= \{0, 1, 2, 3, -3, -2, -1\}$
with arithmetic modulo 7.
e.g. $2 \cdot 5 = 3$ and $3/2 = 5$ in $\mathbf{F}_7$.

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

Examples of addition
on Clock($\mathbf{F}_{1000003}$):
$2(1000, 2) = (4000, 7)$.
$4(1000, 2) = (56000, 97)$.
$8(1000, 2) = (863970, 18817)$.
$16(1000, 2) = (549438, 156853)$.
$17(1000, 2) = (951405, 877356)$.

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

over finite fields



$_7) =$

$\in \mathbf{F}_7 \times \mathbf{F}_7 : x^2 + y^2 = 1\}$.

$= \{0, 1, 2, 3, 4, 5, 6\}$

$2, 3, -3, -2, -1\}$

thmetic modulo 7.

$5 = 3$ and $3/2 = 5$ in $\mathbf{F}_7$.

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

Examples of addition
on Clock($\mathbf{F}_{1000003}$):
$2(1000, 2) = (4000, 7)$.
$4(1000, 2) = (56000, 97)$.
$8(1000, 2) = (863970, 18817)$.
$16(1000, 2) = (549438, 156853)$.
$17(1000, 2) = (951405, 877356)$.

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

“Binary

If n is e

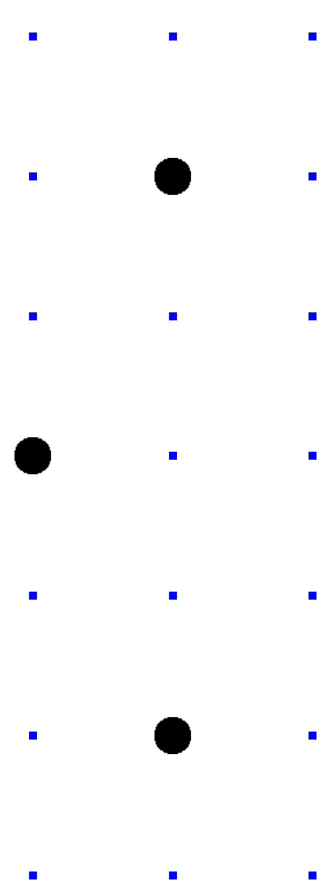
by doub

Otherwis

by addin

This is v

fields



$: x^2 + y^2 = 1\}$.

$, 3, 4, 5, 6\}$

$-2, -1\}$

odulo 7.

$3/2 = 5$ in $\mathbf{F}_7$.

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

Examples of addition
on Clock($\mathbf{F}_{1000003}$):
$2(1000, 2) = (4000, 7)$.
$4(1000, 2) = (56000, 97)$.
$8(1000, 2) = (863970, 18817)$.
$16(1000, 2) = (549438, 156853)$.
$17(1000, 2) = (951405, 877356)$.

"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, comp
by doubling $(n/2)$
Otherwise comput
by adding (x, y) to
This is very fast.

$= 1\}$.

.

n $\mathbf{F}_7$.

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

Examples of addition
on Clock($\mathbf{F}_{1000003}$):
$2(1000, 2) = (4000, 7)$.
$4(1000, 2) = (56000, 97)$.
$8(1000, 2) = (863970, 18817)$.
$16(1000, 2) = (549438, 156853)$.
$17(1000, 2) = (951405, 877356)$.

"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 $(n - 1)($
This is very fast.

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

Examples of addition
on Clock($\mathbf{F}_{1000003}$):
$2(1000, 2) = (4000, 7)$.
$4(1000, 2) = (56000, 97)$.
$8(1000, 2) = (863970, 18817)$.
$16(1000, 2) = (549438, 156853)$.
$17(1000, 2) = (951405, 877356)$.

"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.

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

Examples of addition
on Clock($\mathbf{F}_{1000003}$):
$2(1000, 2) = (4000, 7)$.
$4(1000, 2) = (56000, 97)$.
$8(1000, 2) = (863970, 18817)$.
$16(1000, 2) = (549438, 156853)$.
$17(1000, 2) = (951405, 877356)$.

"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.

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?

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

es of addition

k($\mathbf{F}_{1000003}$):

2) $= (4000, 7)$.

2) $= (56000, 97)$.

2) $= (863970, 18817)$.

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

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

multiplication"

ck:

teger $n \geq 0$

k point (x, y),

e $n(x, y)$.

"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.

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 cr

Standard

and som

Alice ch

Comput

Bob cho

Comput

Alice co

Bob com

They us

to encry

Warning

Many ch

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

ion

):

0, 7).

00, 97).

970, 18817).

9438, 156853).

1405, 877356).

tion"

0

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.

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 cryptograph

Standardize a larg
and some $(x, y) \in$

Alice chooses big s
Computes her pub

Bob chooses big s
Computes his pub

Alice computes $a($
Bob computes $b(a$
They use this shar
to encrypt with Al

Warning #1:
Many choices of p

0003).

7).

353).

356).

"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 $a($

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

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

Warning #1:
Many choices of p are bad!

"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!

method":
ven, compute $n(x, y)$
ling $(n/2)(x, y)$.
se compute $n(x, y)$
g (x, y) to $(n-1)(x, y)$.
very fast.

ring out n
$, y)$ and $n(x, y)$
more difficult.

clock additions
puted
$2) = (947472, 736284)$
e 6-digit n.
figure out n?

Clock cryptography

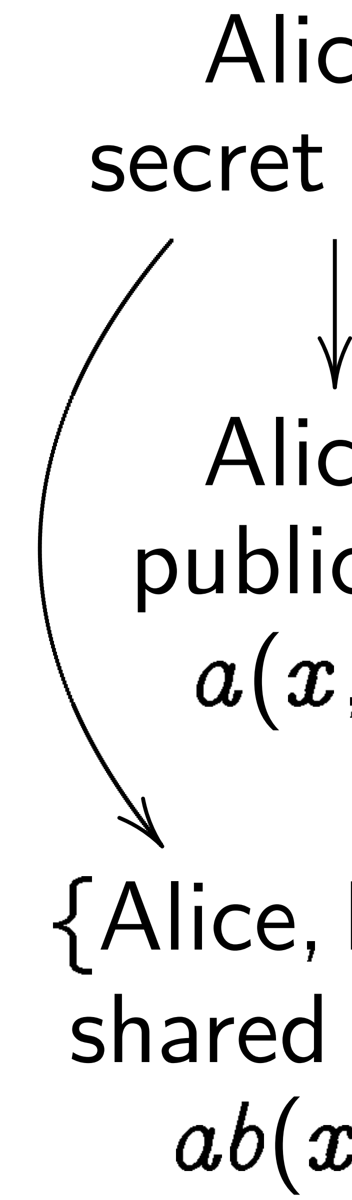
Standardize a large prime p
and some $(x, y) \in \text{Clock}(\mathbf{F}_p)$.

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!



ute $n(x, y)$
(x, y).
e $n(x, y)$
o $(n-1)(x, y)$.

,

(x, y)
cult.

litions

$472, 736284)$

.
t n?

Clock cryptography

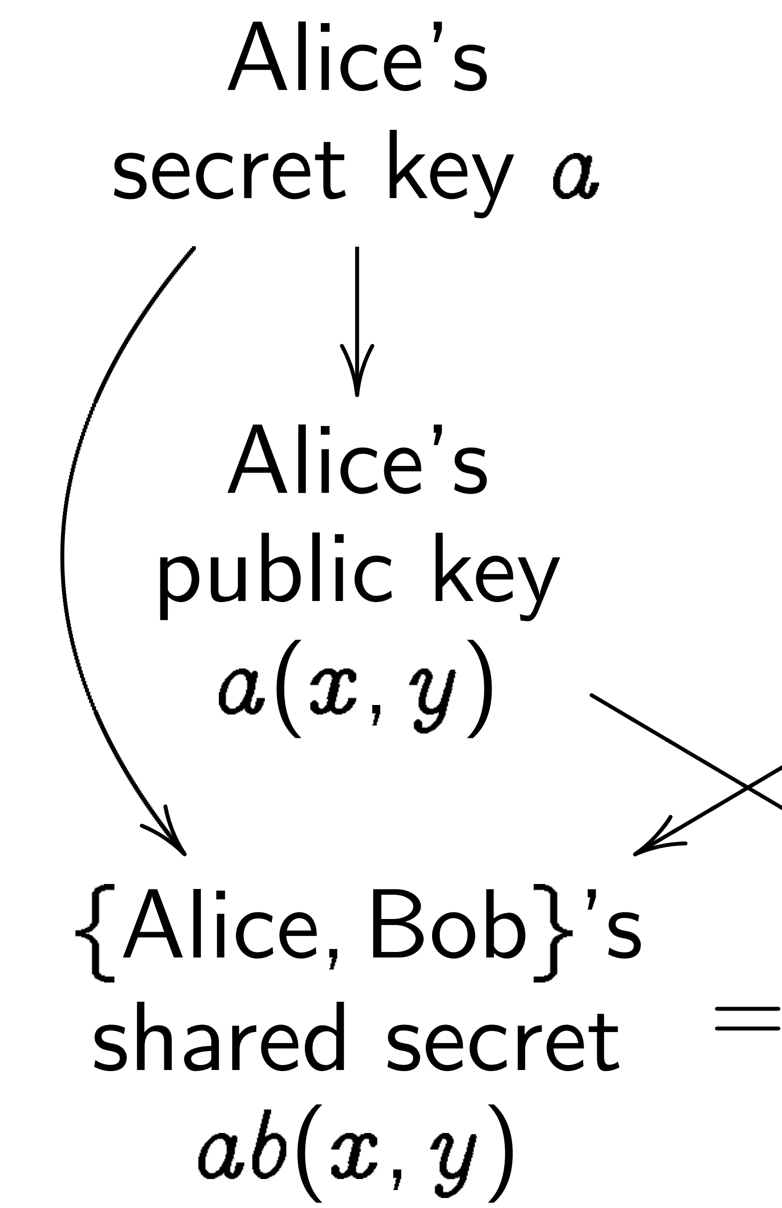
Standardize a large prime p
and some $(x, y) \in \text{Clock}(\mathbf{F}_p)$.

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!



$y)$

(x, y).

34)

Clock cryptography

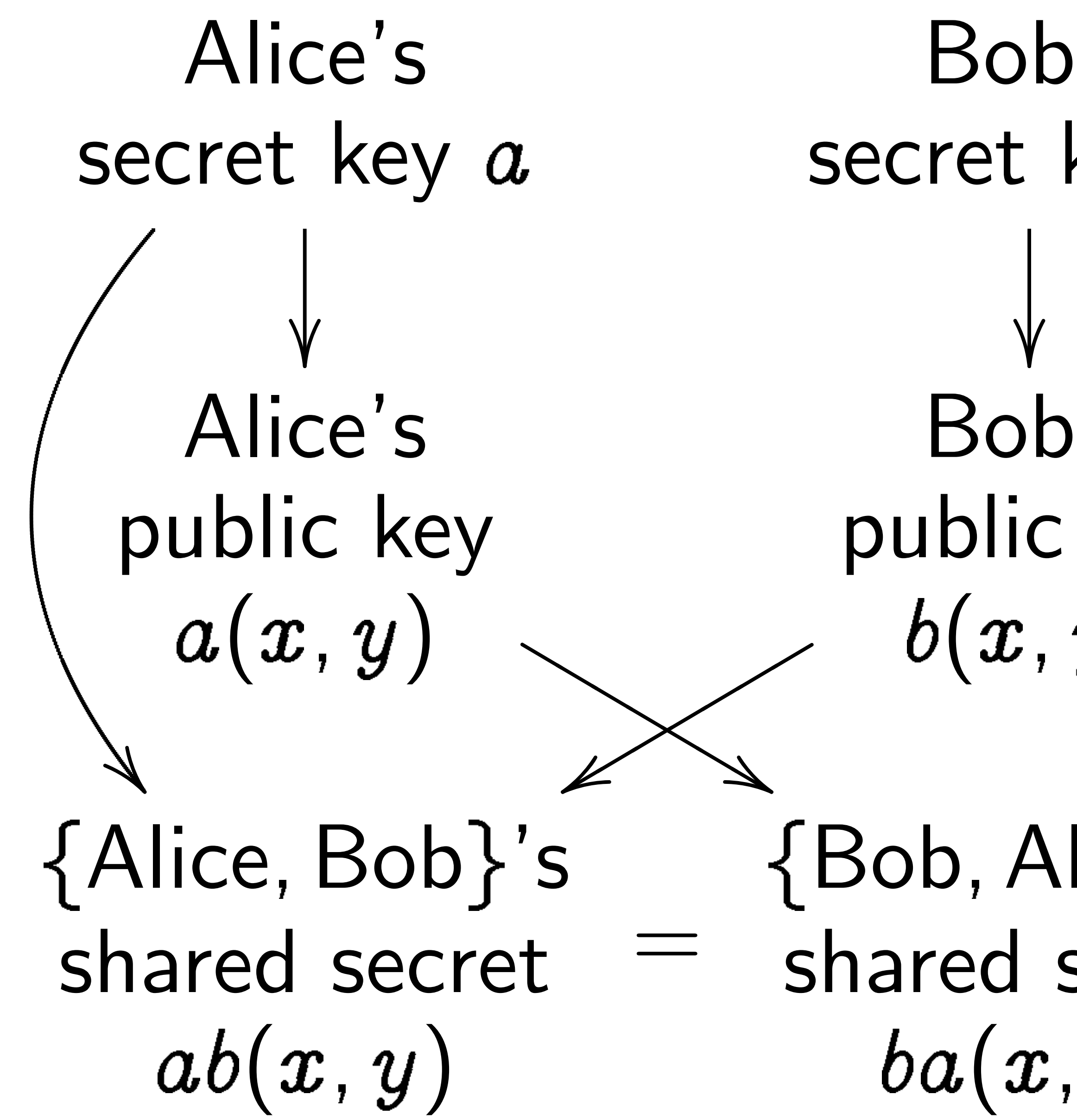
Standardize a large prime p and some $(x, y) \in \text{Clock}(\mathbf{F}_p)$.

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!



Clock cryptography

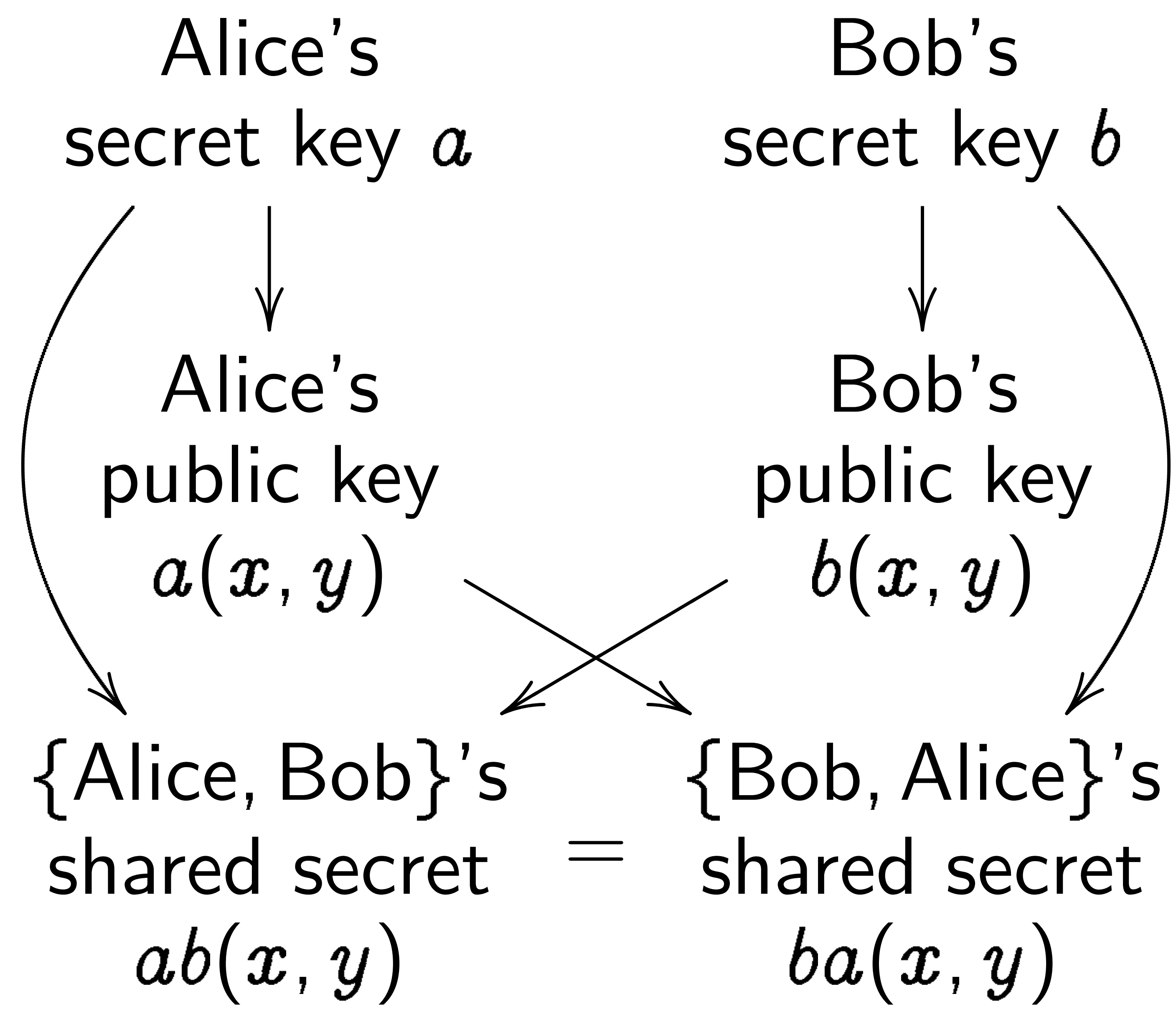
Standardize a large prime p and some $(x, y) \in \text{Clock}(\mathbf{F}_p)$.

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!



Clock cryptography

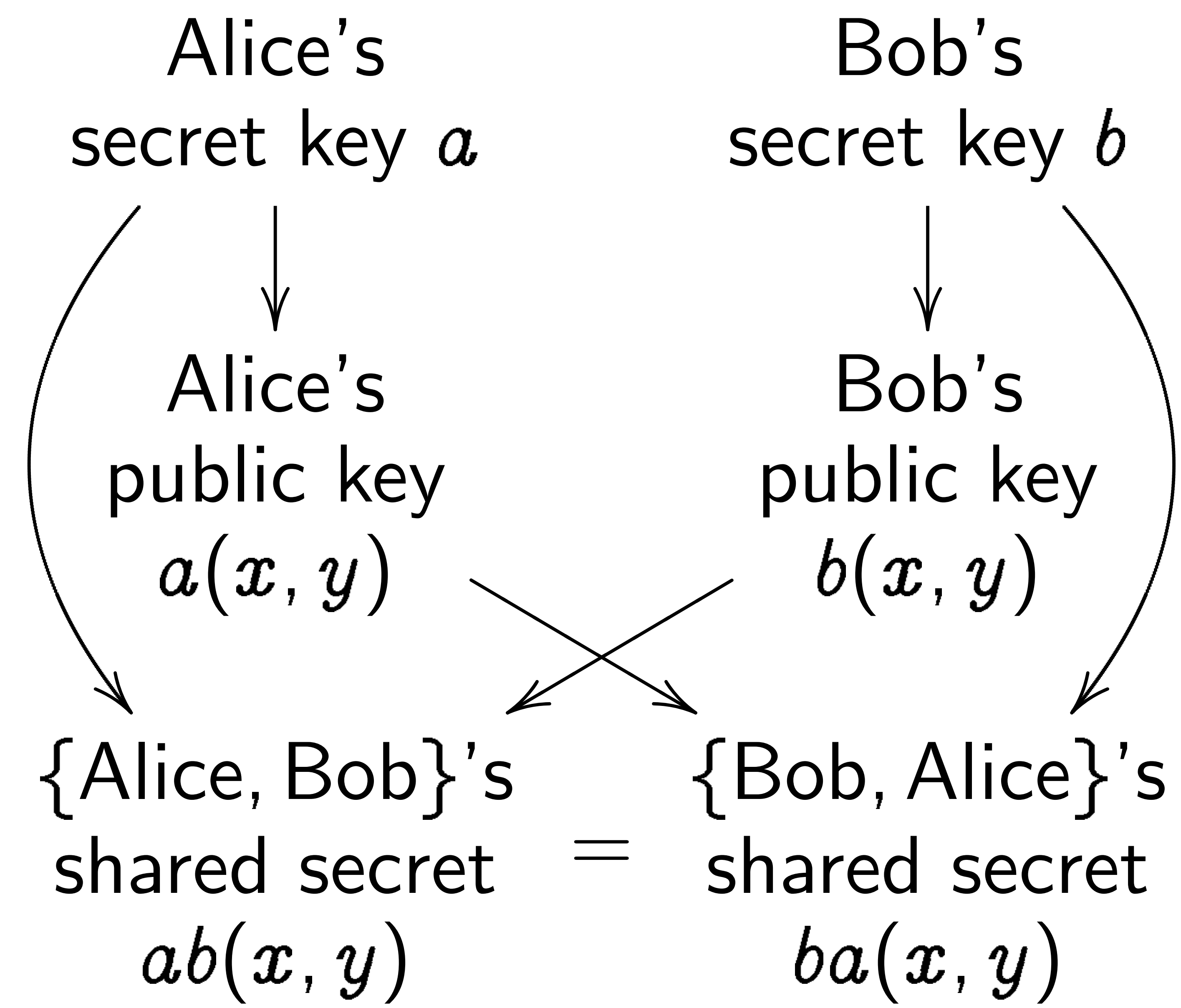
Standardize a large prime p
and some $(x, y) \in \text{Clock}(\mathbf{F}_p)$.

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!



Warning #2:
Clocks aren't elliptic!
Can use index calculus
to attack clock cryptography.
To match RSA-3072 security
need $p \approx 2^{1536}$.

yptography

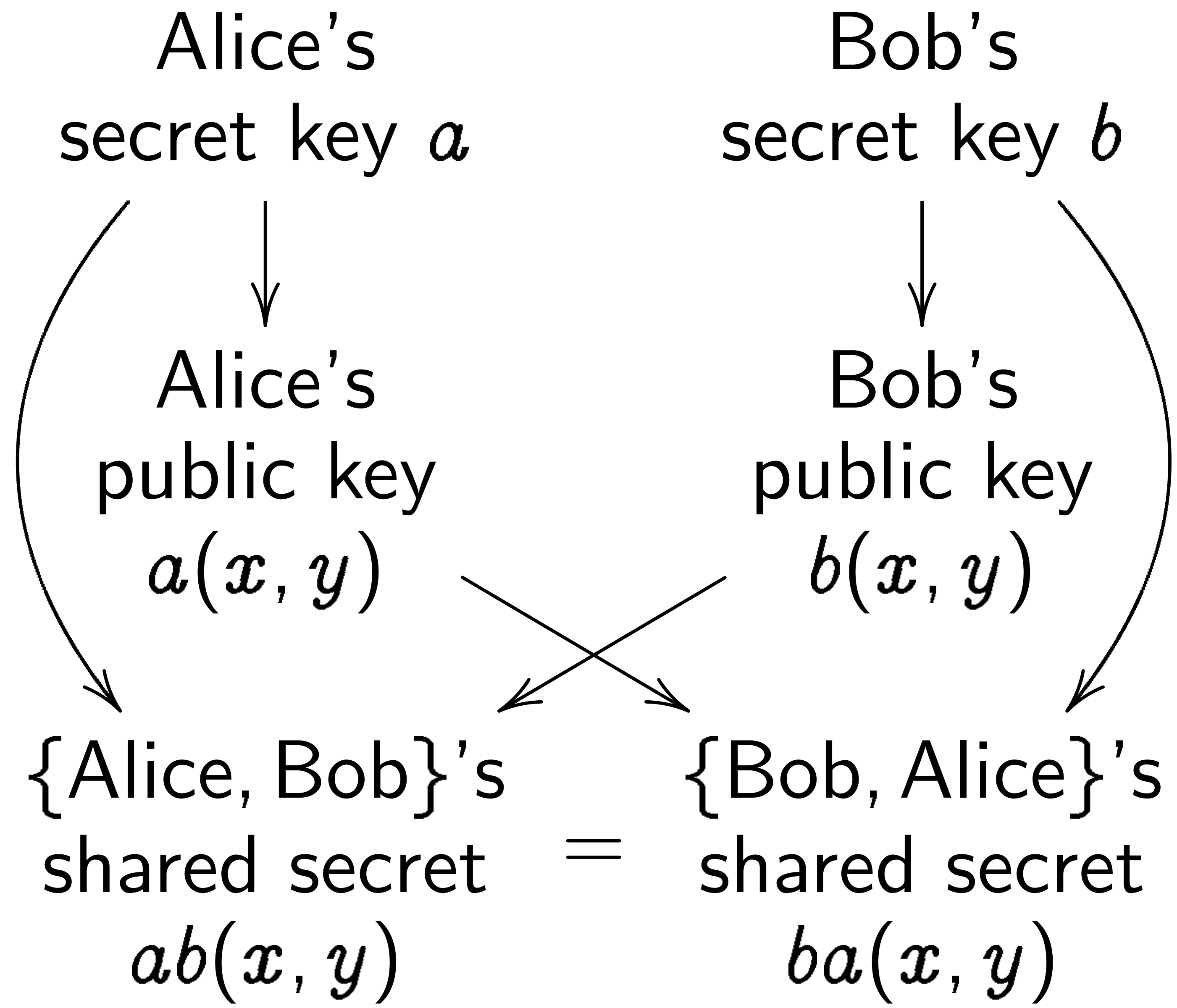
dize a large prime p
e $(x, y) \in \text{Clock}(\mathbf{F}_p)$.

ooses big secret a.
es her public key $a(x, y)$.

oses big secret b.
es his public key $b(x, y)$.

mputes $a(b(x, y))$.
nputes $b(a(x, y))$.
e this shared secret
pt with AES-GCM etc.

#1:
hoices of p are bad!



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 a

Attacker
$a(x, y)$ a

Attacker
Alice to
Often at
time for
performe
not just
This rev

Fix: **con**
performi
no matt

y

e prime p

Clock($\mathbf{F}_p$).

secret a.

lic key $a(x, y)$.

ecret b.

lic key $b(x, y)$.

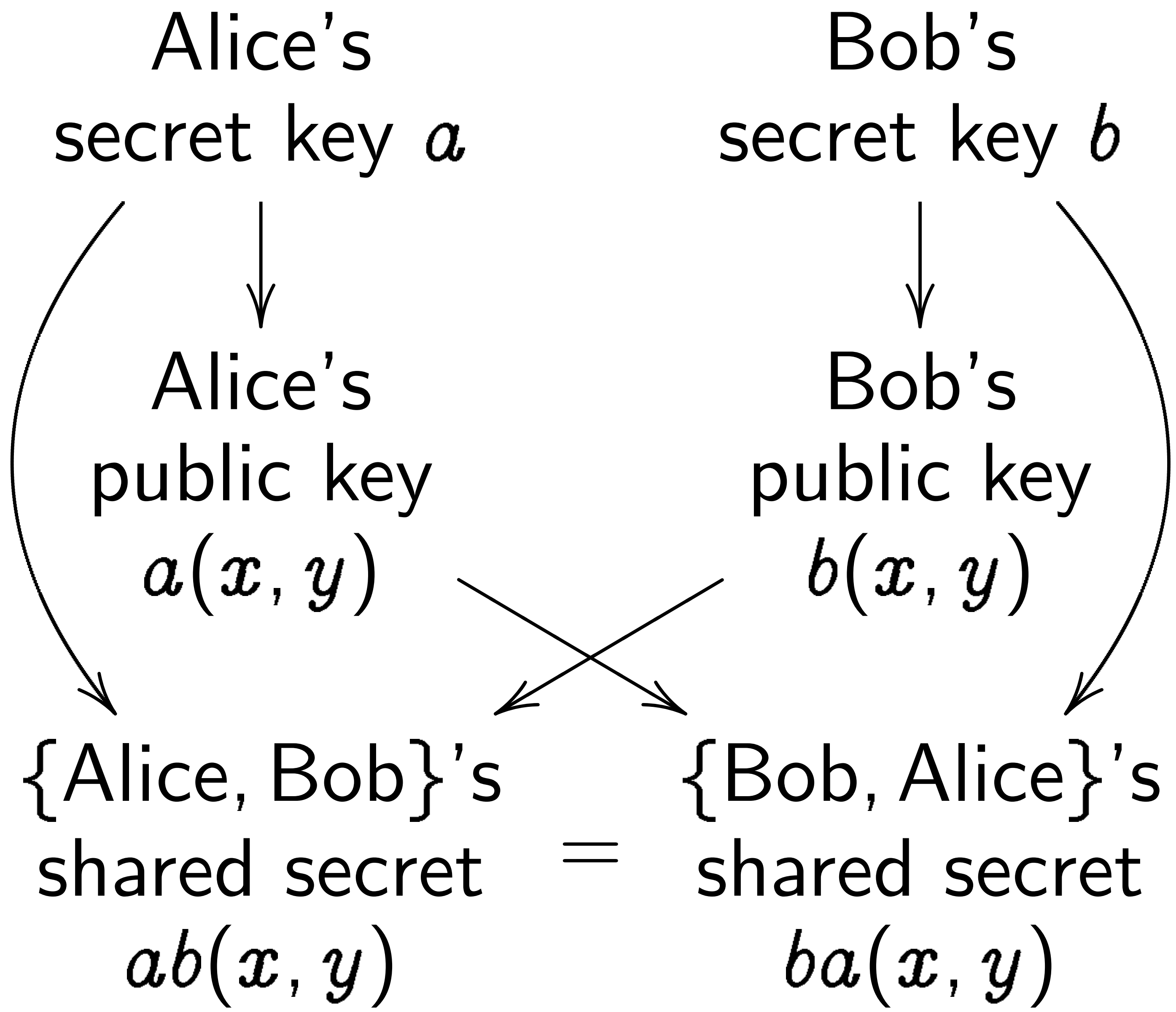
$b(x, y)$).

(x, y)).

red secret

ES-GCM etc.

are bad!



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
$a(x, y)$ and $b(x, y$

Attacker sees *time*
Alice to compute
Often attacker can
time for *each oper*
performed by Alice
not just total time
This reveals secret

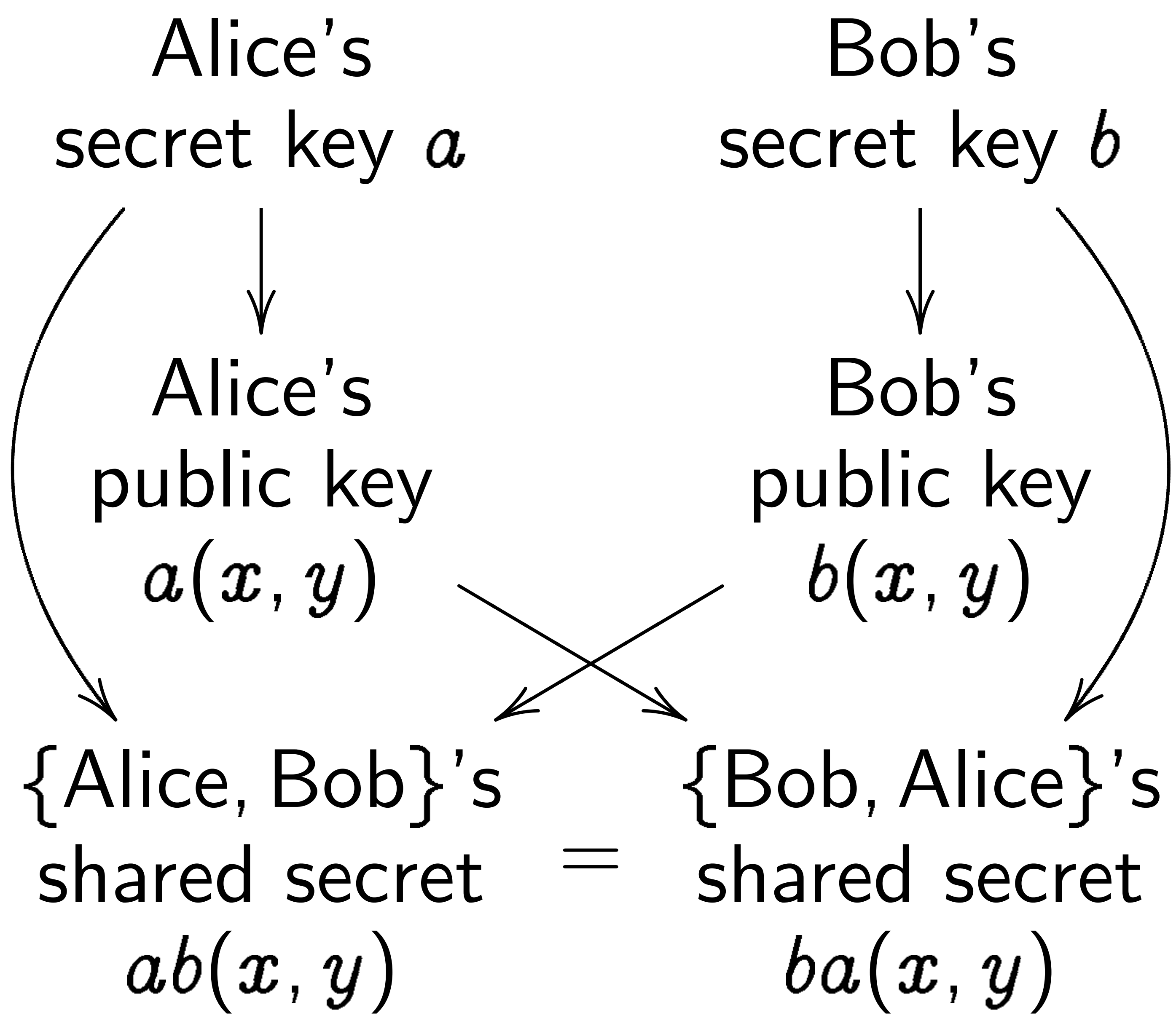
Fix: **constant-tim**
performing same o
no matter what sc

).

$x, y)$.

$, y)$.

etc.



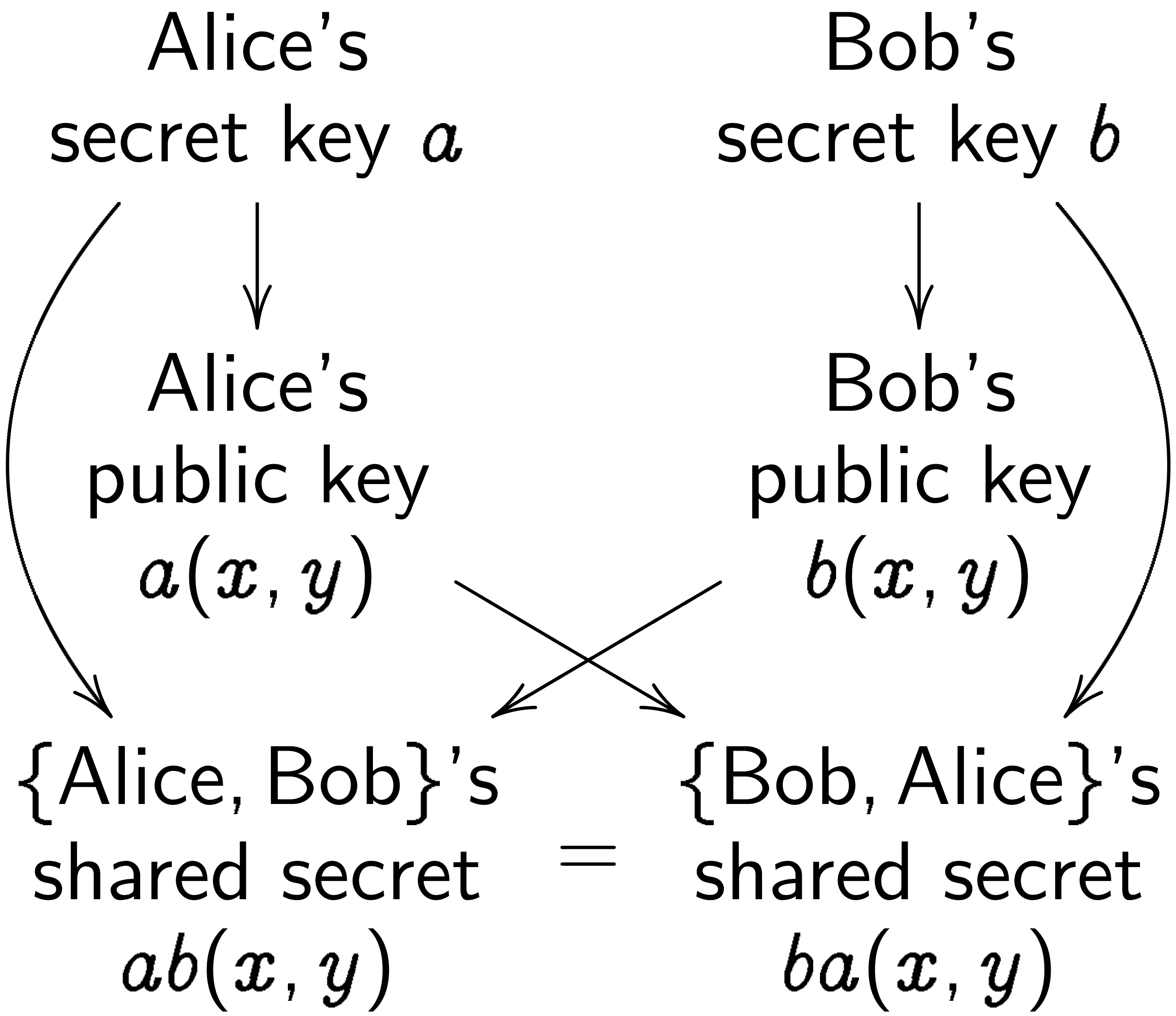
Warning #2:
Clocks aren't elliptic!
Can use index calculus
to attack clock cryptography.
To match RSA-3072 security
need $p \approx 2^{1536}$.

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.



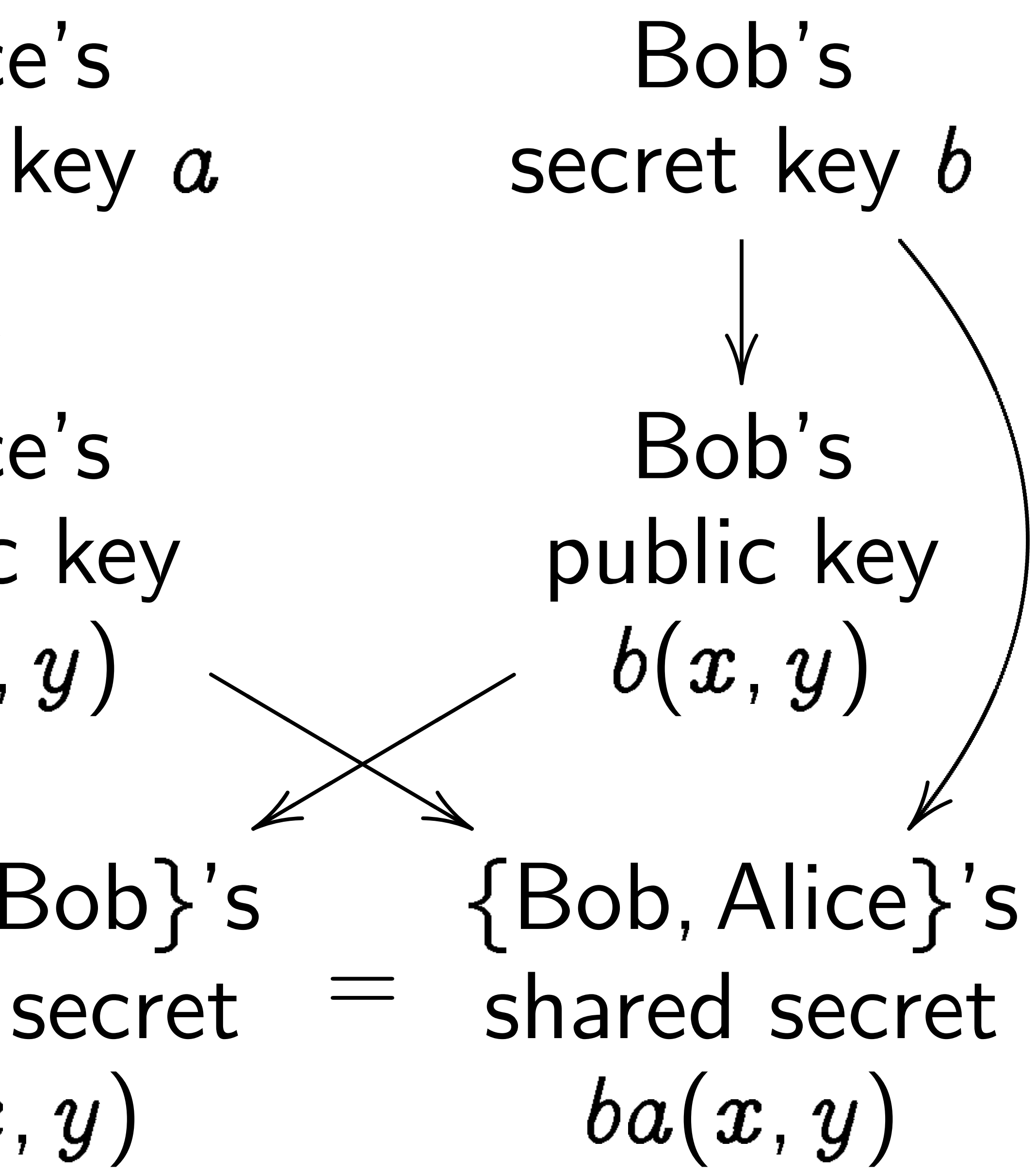
Warning #2:
Clocks aren't elliptic!
Can use index calculus
to attack clock cryptography.
To match RSA-3072 security
need $p \approx 2^{1536}$.

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.



#2:
aren't elliptic!
index calculus
k clock cryptography.
ch RSA-3072 security
$\approx 2^{1536}$.

Timing attacks

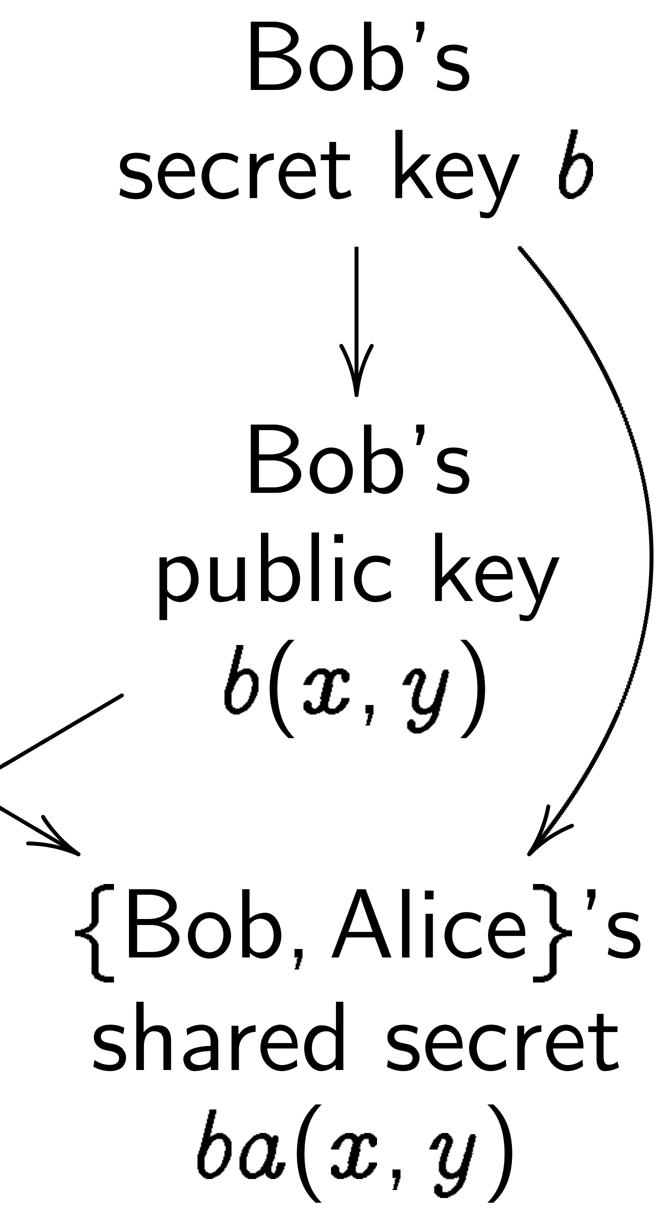
Attacker sees more than $a(x, y)$ and $b(x, y)$.

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

$x^2 + y^2$
Sum of
$((x_1 y_2 +$
$(y_1 y_2 -$



tic!

culus

yptography.

72 security

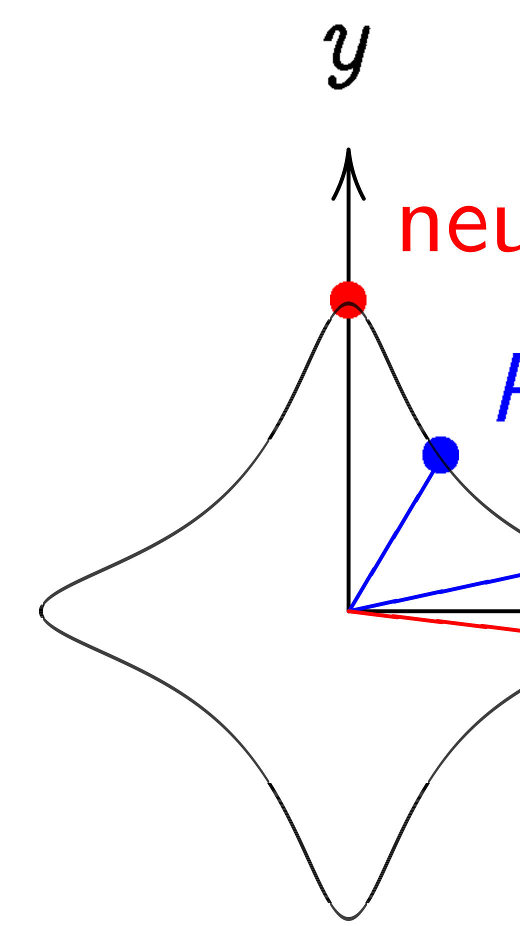
Timing attacks

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 elli



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

Sum of (x_1, y_1) ar

$((x_1y_2+y_1x_2)/(1-$

$(y_1y_2-x_1x_2)/(1-$

’s
key b

’s
key
$y)$

lice}’s
secret
$y)$

y.

y

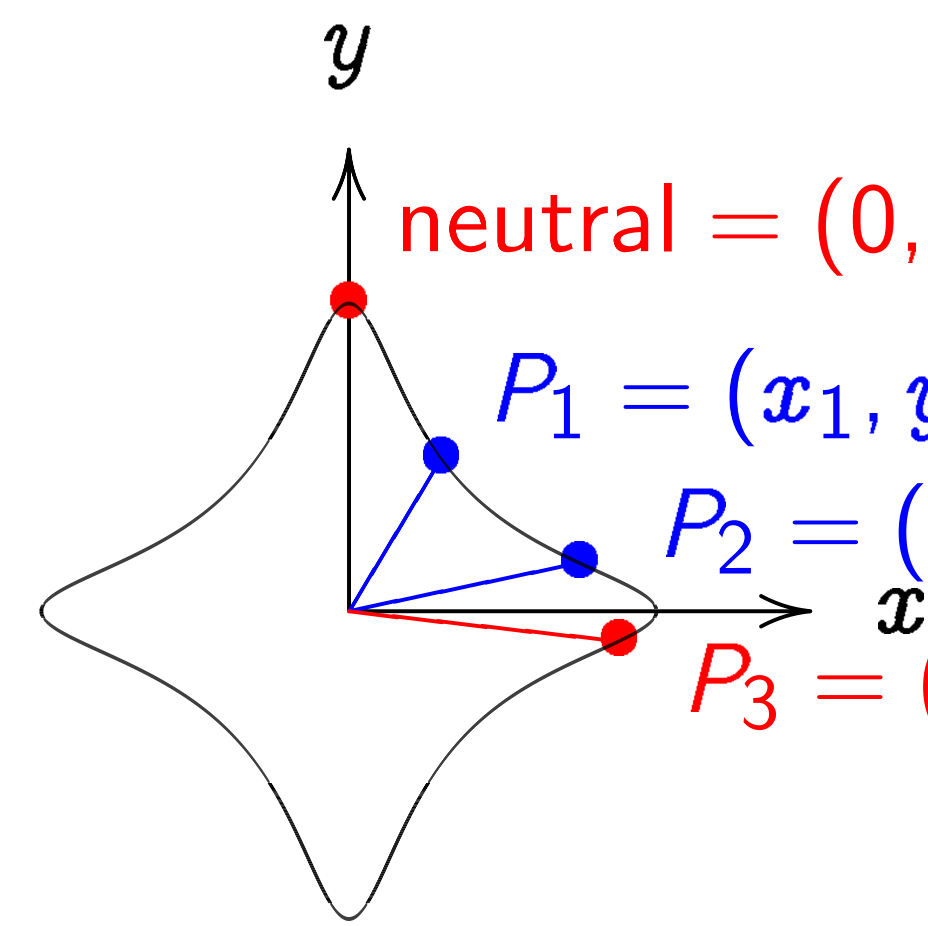
Timing attacks

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)

$((x_1y_2{+}y_1x_2)/(1{-}30x_1x_2y$

$(y_1y_2{-}x_1x_2)/(1{+}30x_1x_2y$

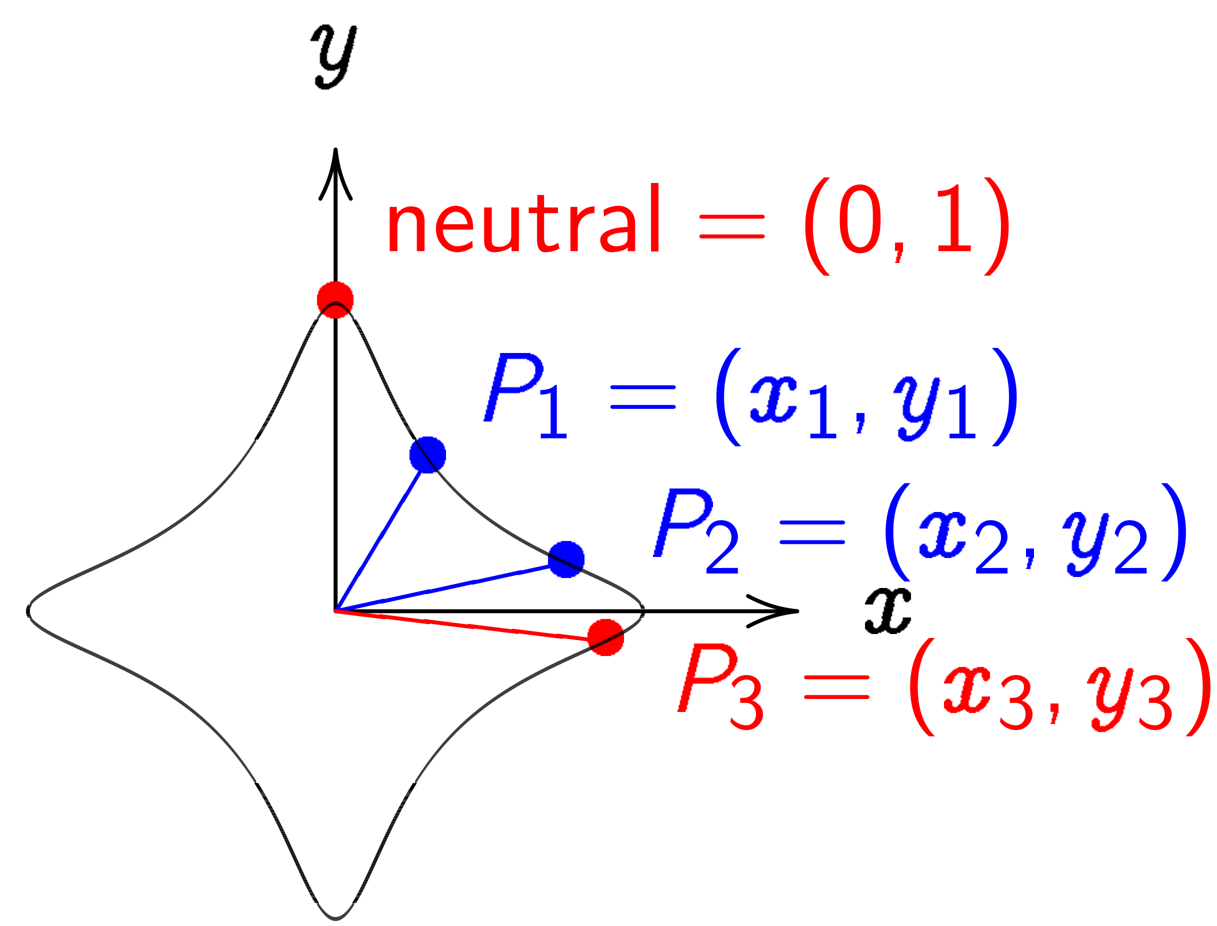
Timing attacks

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

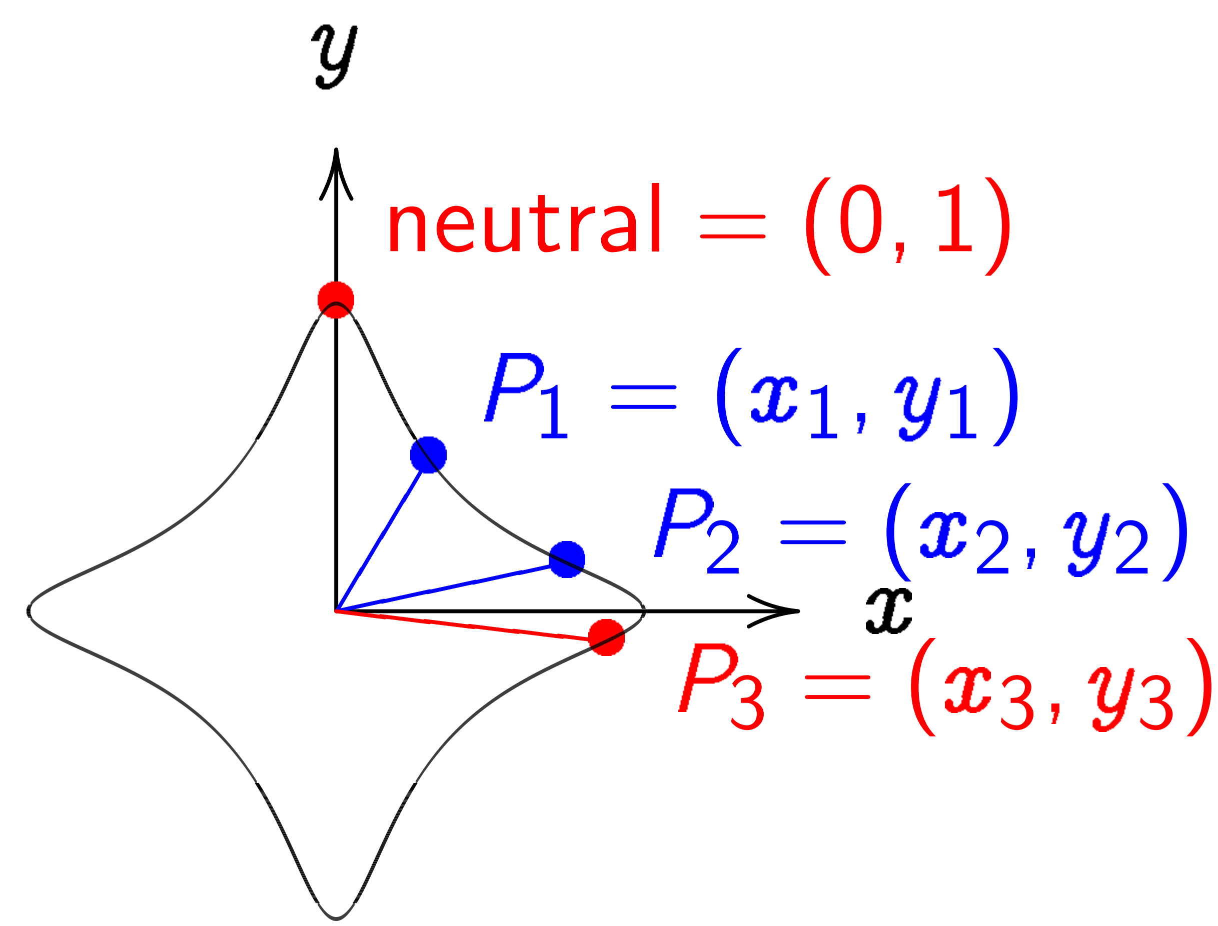
attacks

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

r sees *time* for
compute $a(b(x, y))$.
ttacker can see
each operation
ed by Alice,
total time.
eals secret a.

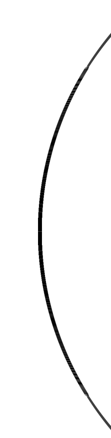
nstant-time code,
ng same operations
er what scalar is.

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 cloc



$x^2 + y^2$
Sum of (
$(x_1y_2 +$
$y_1y_2 -$

e than

).

e for

$a(b(x, y))$.

n see

ration

e,

e.

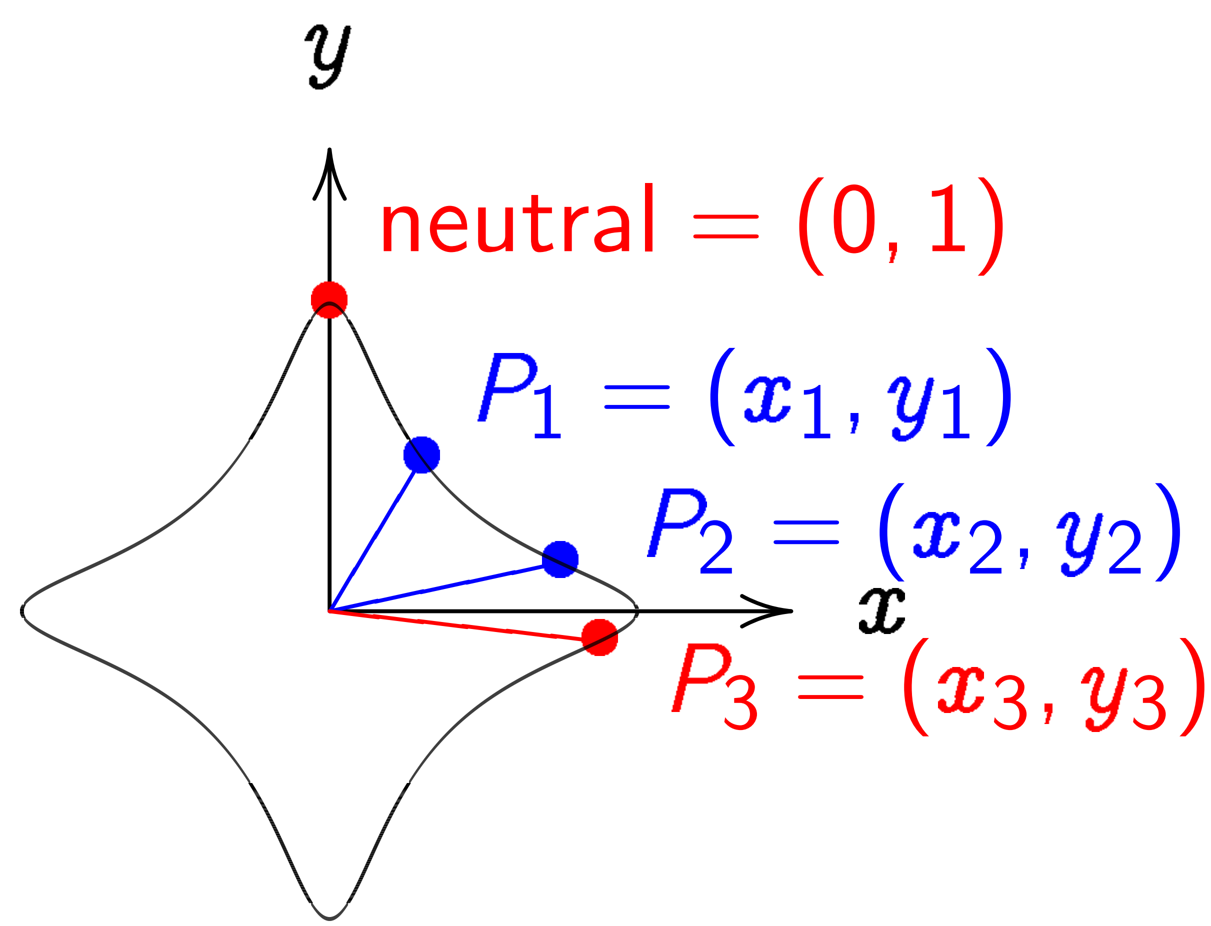
t a.

ne code,

operations

calar is.

Addition on an elliptic curve



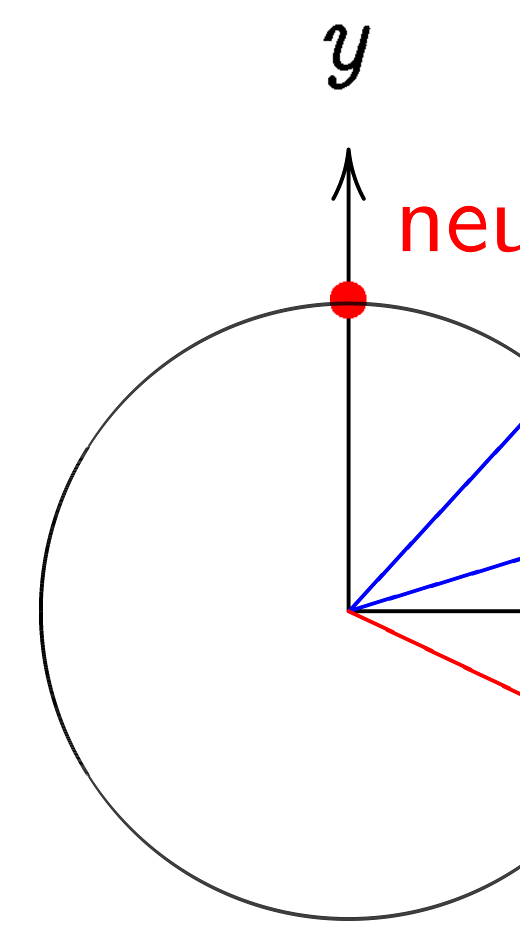
$x^2 + y^2 = 1 - 30x^2y^2$.

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, f



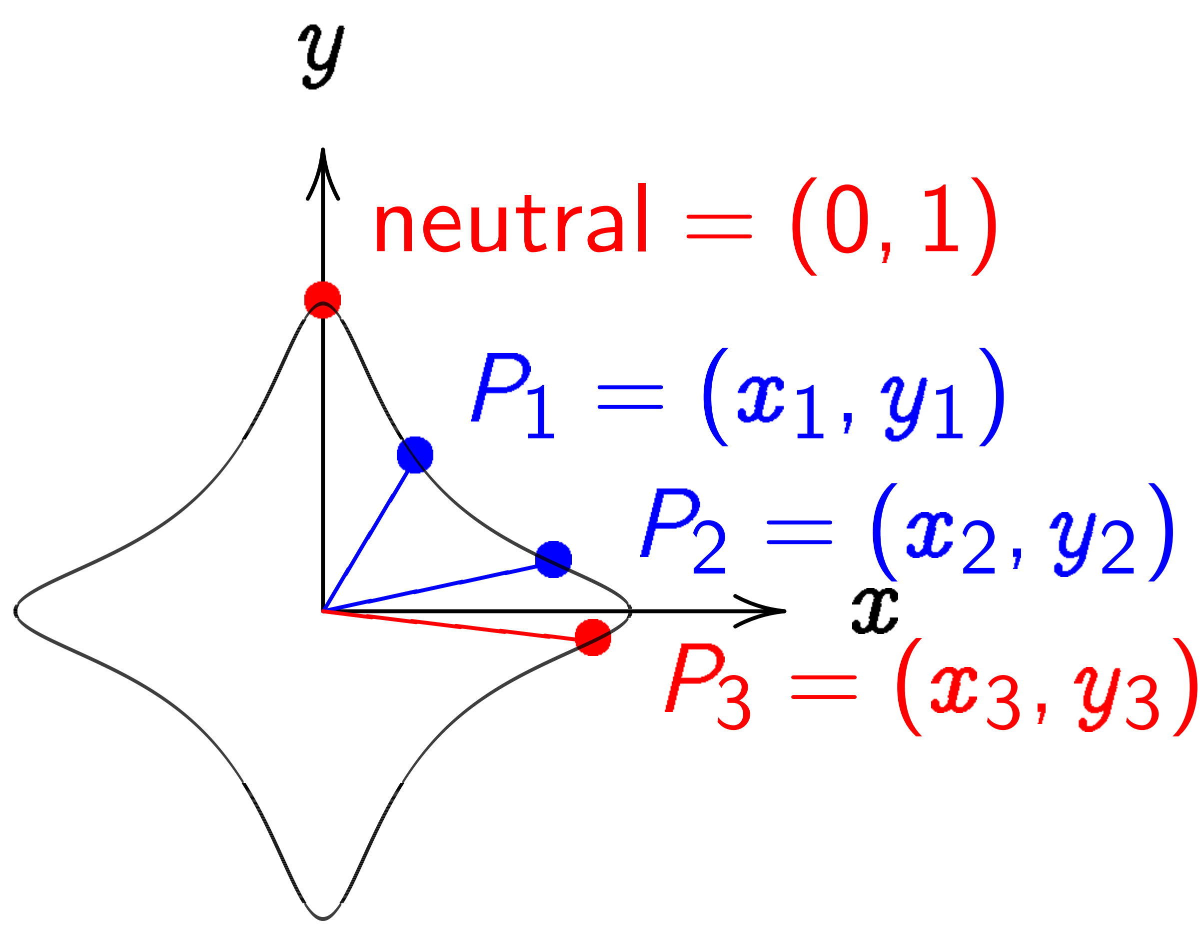
$x^2 + y^2 = 1$.

Sum of (x_1, y_1) ar

$(x_1y_2 + y_1x_2,$

$y_1y_2 - x_1x_2)$.

Addition on an elliptic curve



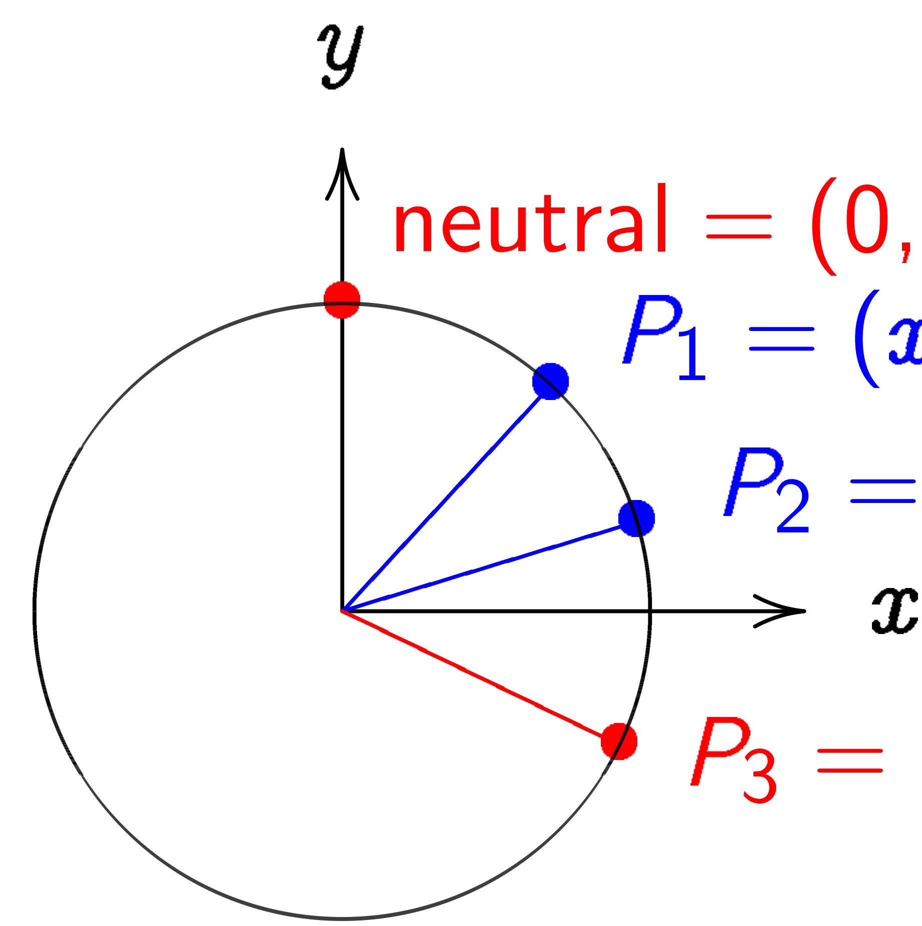
$x^2 + y^2 = 1 - 30x^2y^2$.

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 compar



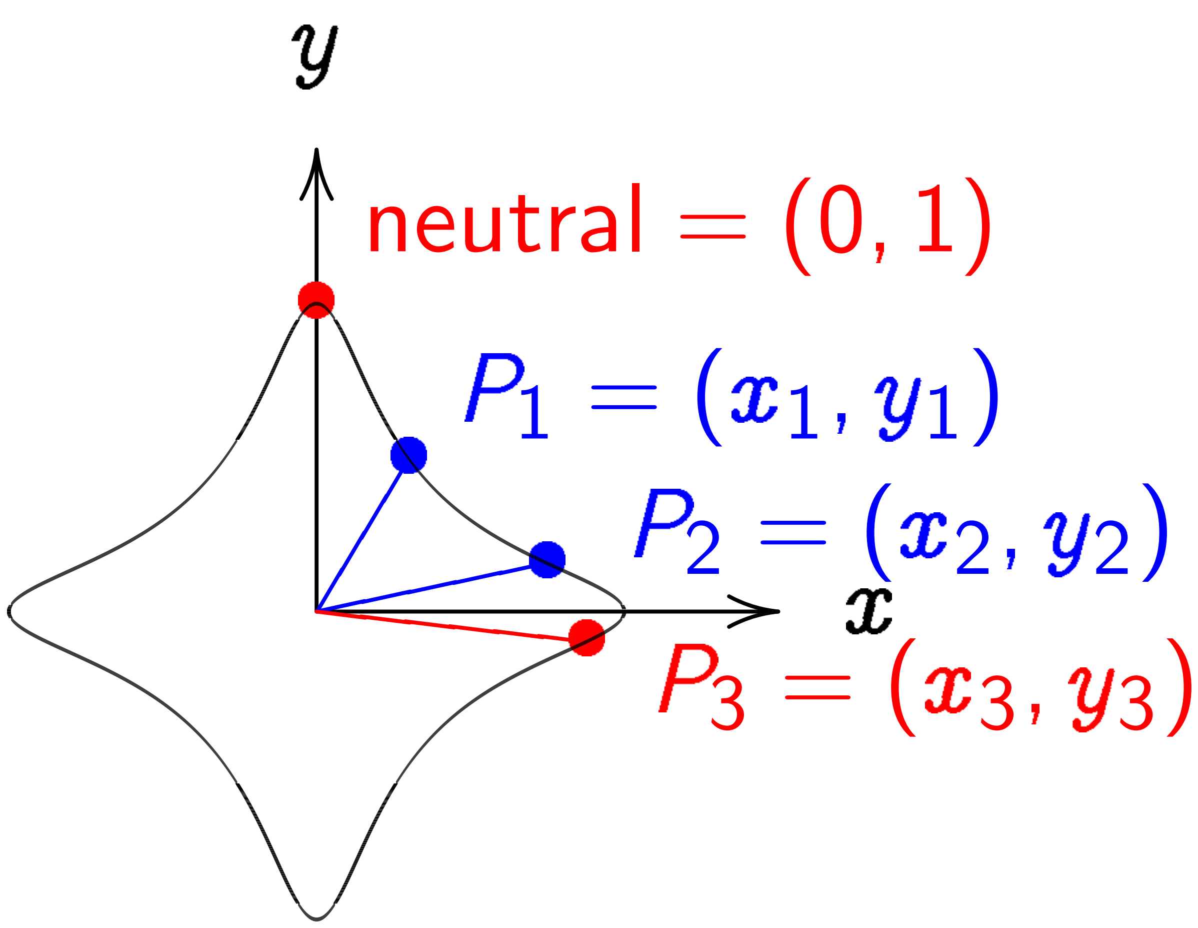
$x^2 + y^2 = 1$.

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

$(x_1y_2 + y_1x_2,$

$y_1y_2 - x_1x_2)$.

Addition on an elliptic curve



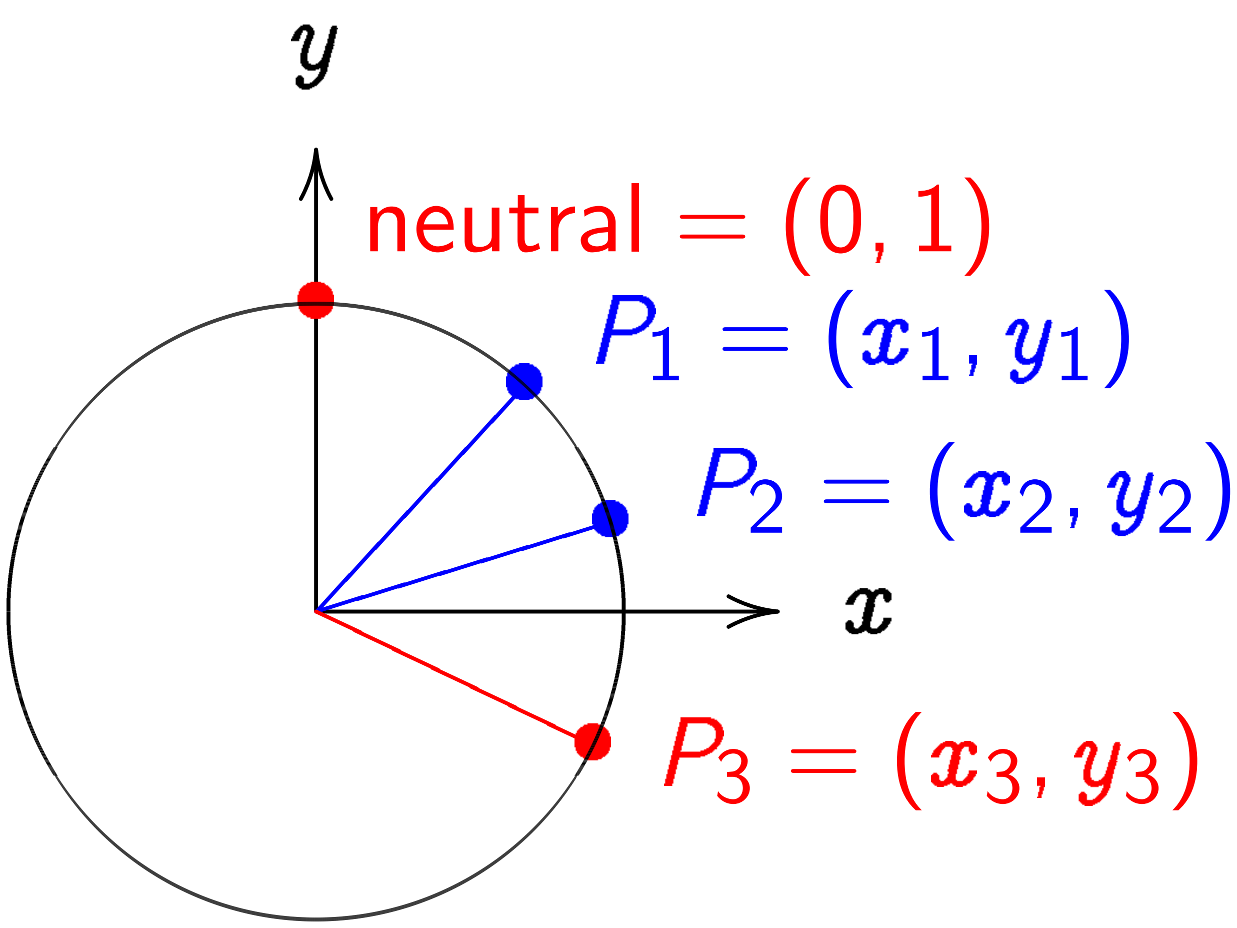
$x^2 + y^2 = 1 - 30x^2y^2$.

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:



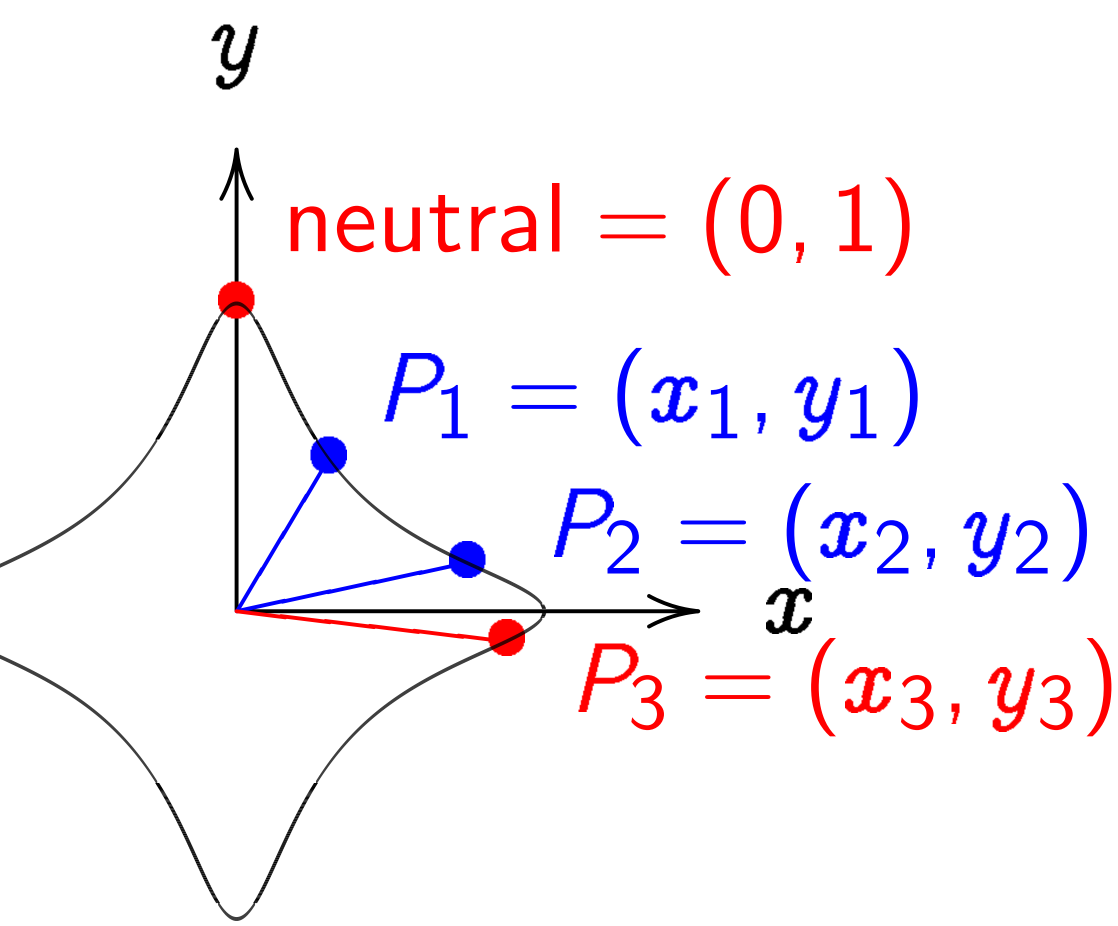
$x^2 + y^2 = 1$.

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

$(x_1y_2 + y_1x_2,$

$y_1y_2 - x_1x_2)$.

n on an elliptic curve



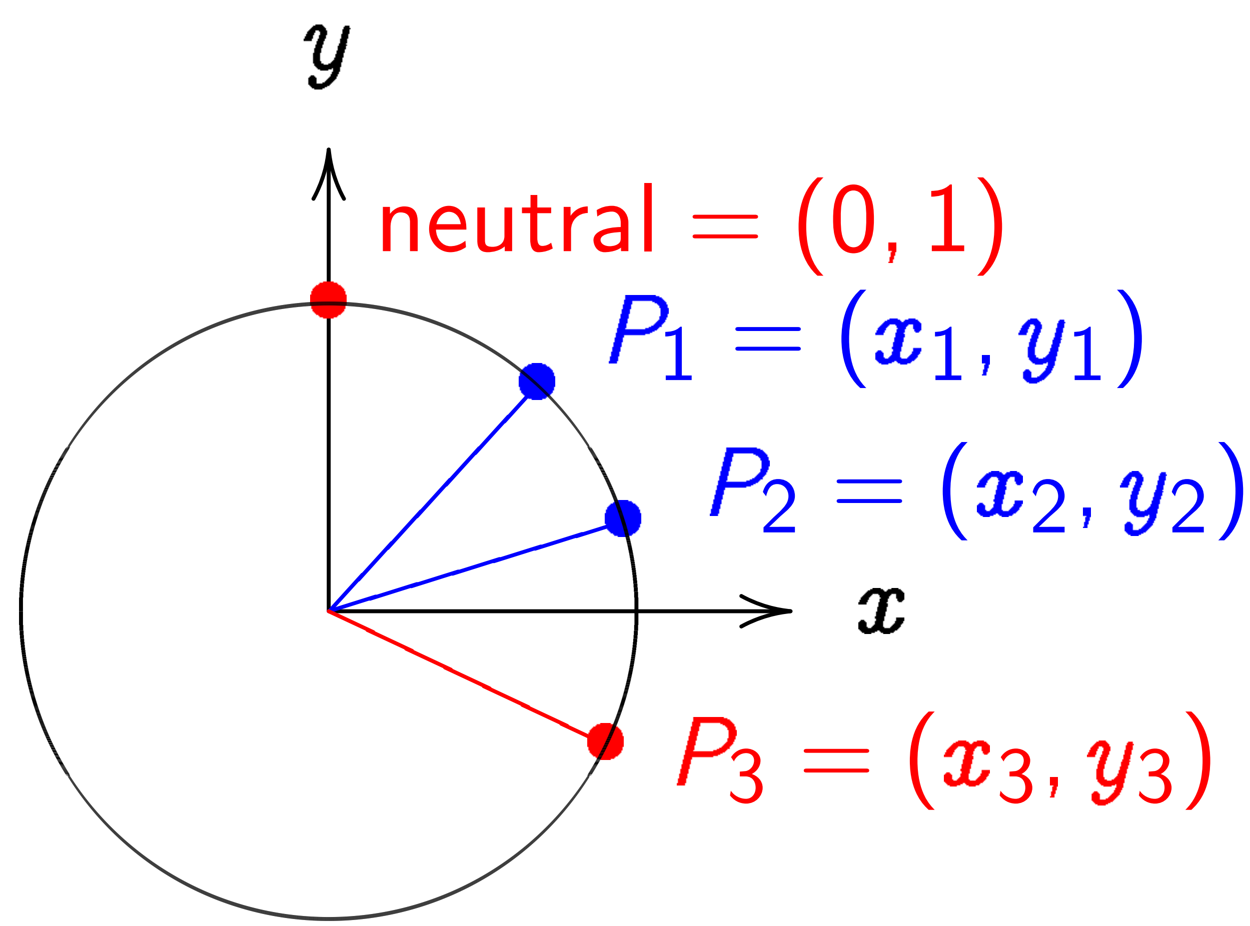
$= 1 - 30x^2y^2$.

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

$y_1x_2)/(1-30x_1x_2y_1y_2)$,

$x_1x_2)/(1+30x_1x_2y_1y_2))$.

The clock again, for comparison:



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

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

$(x_1y_2 + y_1x_2,$

$y_1y_2 - x_1x_2)$.

More ell

Choose

Choose

$\{(x, y) \in$

$x^2 +$

is a “co

“The Ed

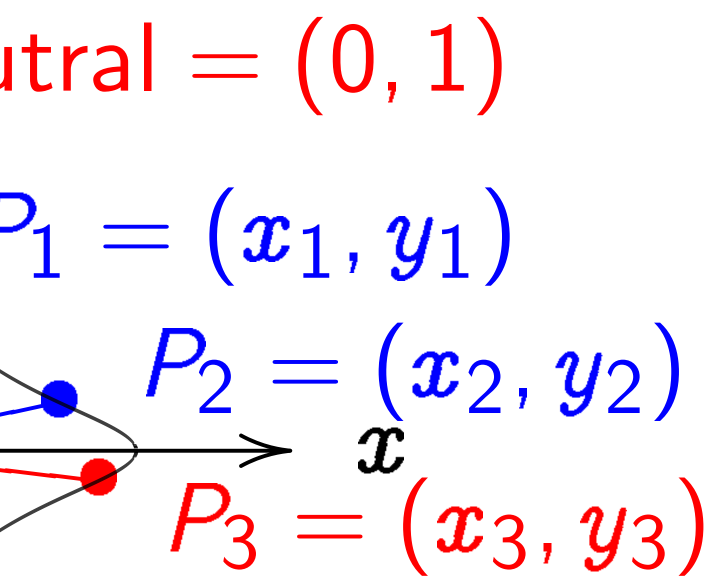
(x_1, y_1)

where

$x_3 = \frac{x}{1 +}$

$y_3 = \frac{y}{1 -}$

iptic curve



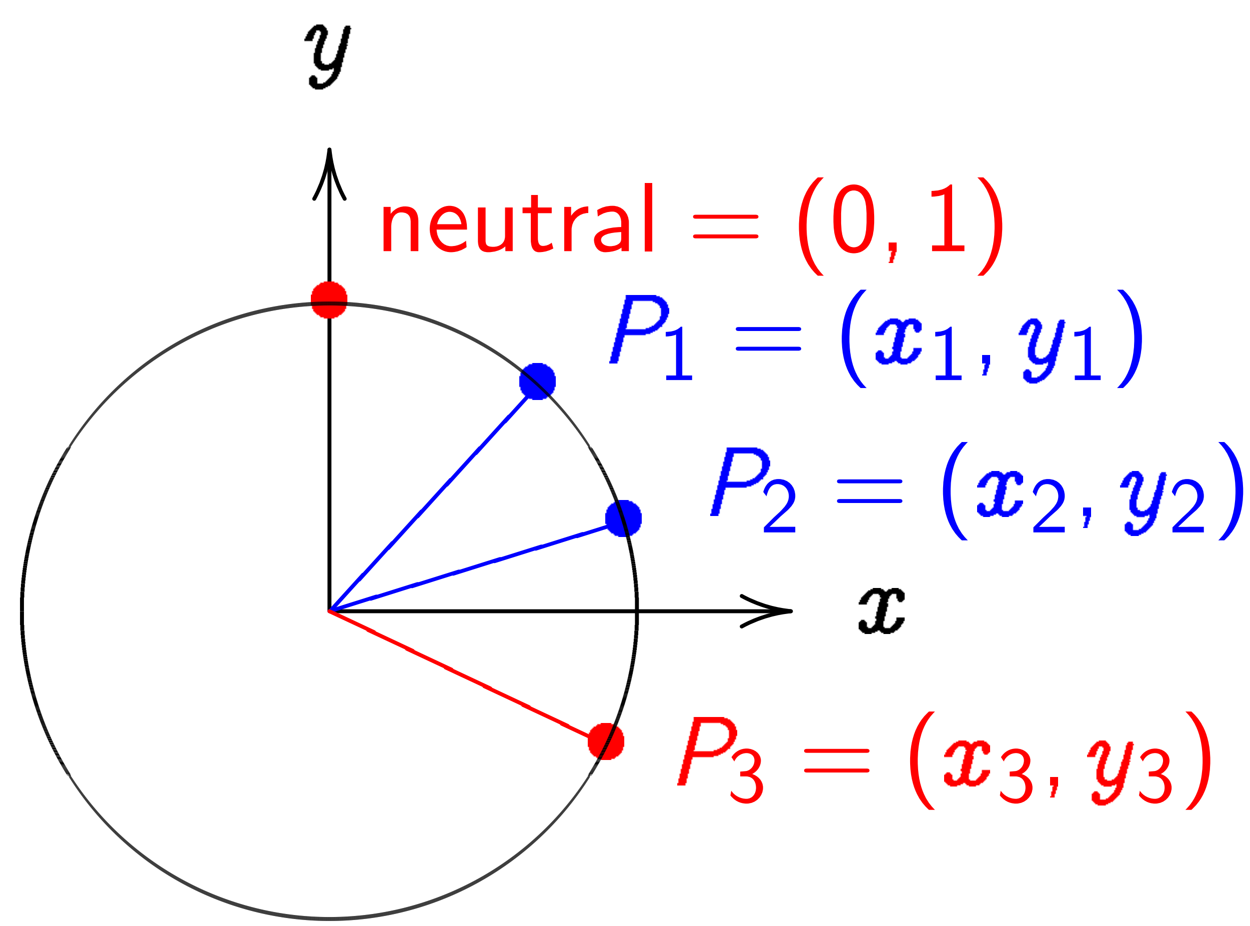
c^2y^2.

nd (x_2, y_2) is

$-30x_1x_2y_1y_2)$,

$+30x_1x_2y_1y_2))$.

The clock again, for comparison:



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

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

$(x_1y_2 + y_1x_2,$

$y_1y_2 - x_1x_2)$.

More elliptic curve

Choose an odd pri

Choose a *non-squa*

$\{(x, y) \in \mathbf{F}_p \times \mathbf{F}_p$

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

is a "complete Edw

"The Edwards add

$(x_1, y_1) + (x_2, y_2)$

where

$x_3 = \frac{x_1y_2 + y_1x}{1 + dx_1x_2y_1}$

$y_3 = \frac{y_1y_2 - x_1x}{1 - dx_1x_2y_1}$

e

1)

$y_1)$

$x_2, y_2)$

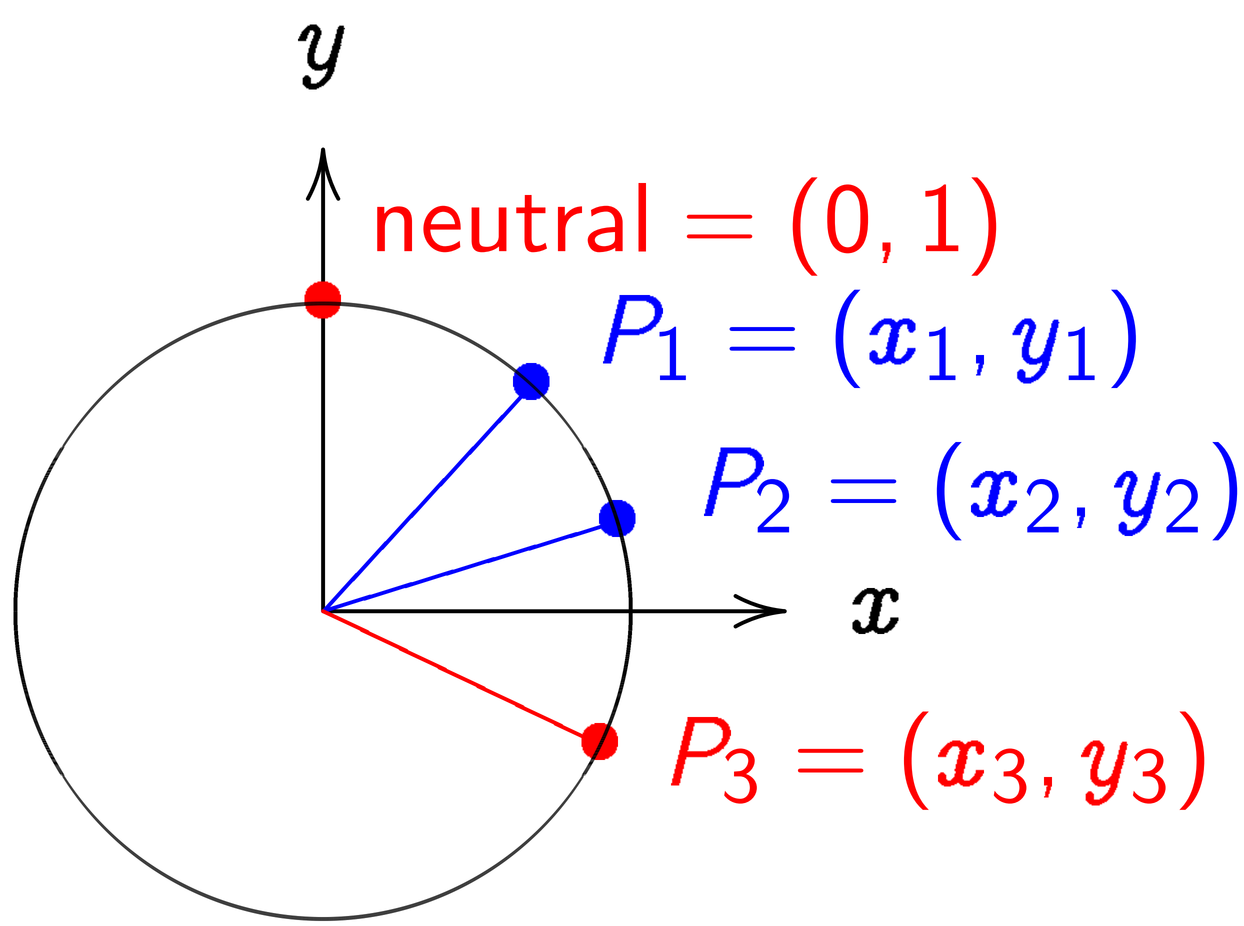
(x_3, y_3)

) is

$_1y_2)$,

$_1y_2))$.

The clock again, for comparison:



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

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

$(x_1y_2 + y_1x_2,$

$\ y_1y_2 - x_1x_2)$.

More elliptic curves

Choose an odd prime p.

Choose a *non-square* $d \in \mathbf{F}_p$

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

is a "complete Edwards curv

"The Edwards addition law"

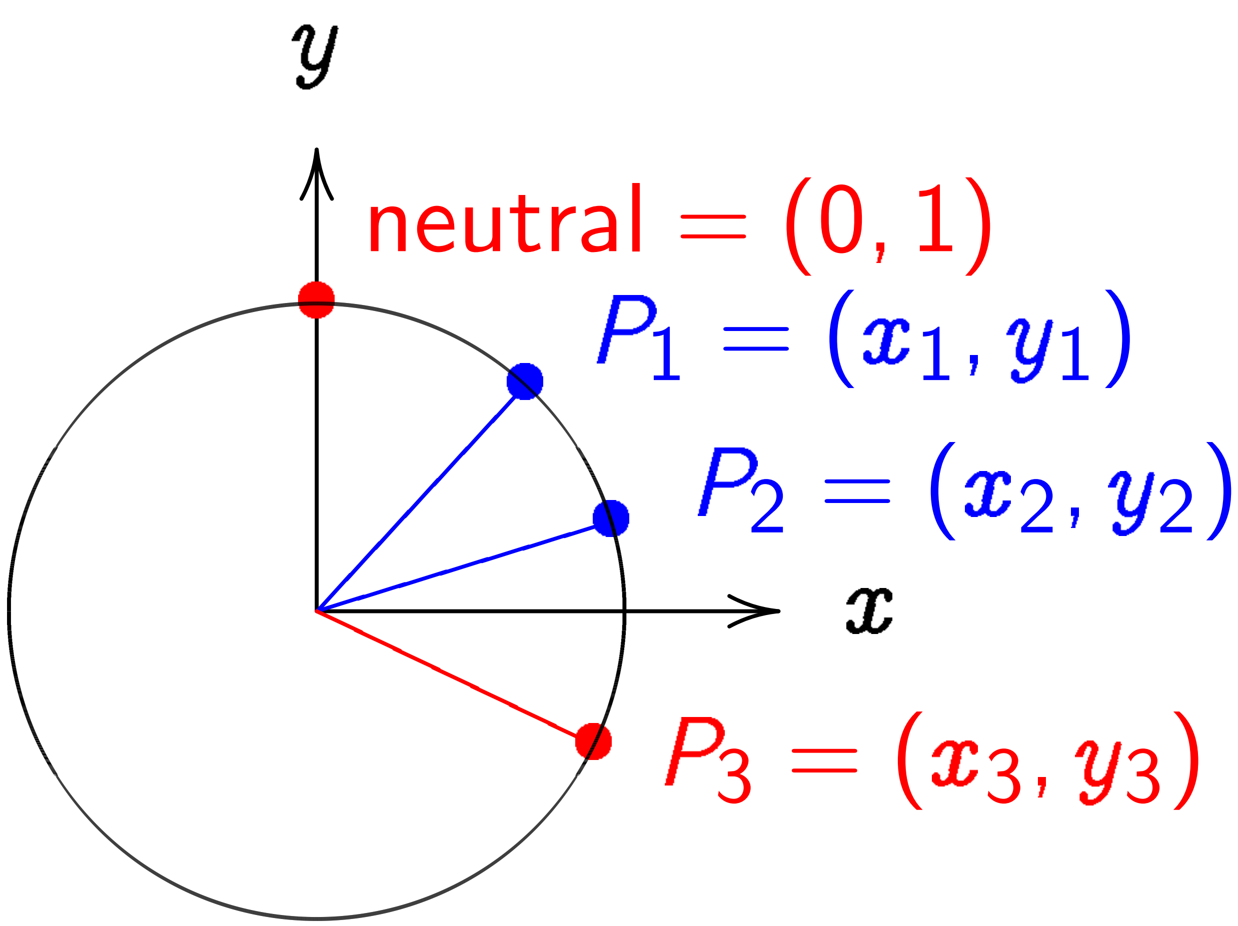
$(x_1, y_1) + (x_2, y_2) = (x_3, y_3$

where

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

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

The clock again, for comparison:



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

Sum of (x_1, y_1) and (x_2, y_2) is
$(x_1y_2 + y_1x_2,$
$y_1y_2 - x_1x_2).$

More elliptic curves

Choose an odd prime p.

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

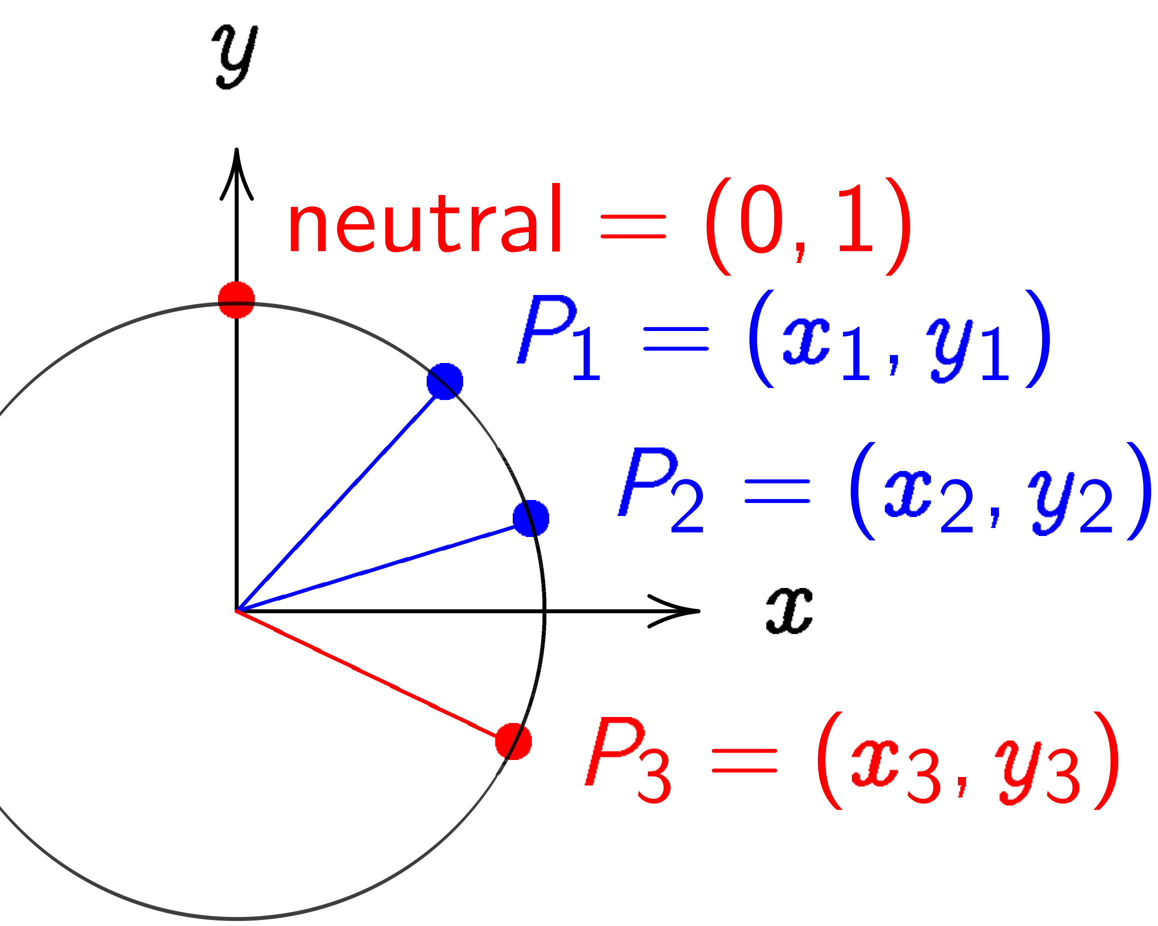
$\{(x, y) \in \mathbf{F}_p \times \mathbf{F}_p :$
$x^2 + y^2 = 1 + dx^2y^2\}$
is a "complete Edwards curve".

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

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

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

ck again, for comparison:



$= 1.$

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

$y_1 x_2,$

$x_1 x_2).$

<u>More elliptic curves</u>

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

$$x_3 = \frac{x_1 y_2 + y_1 x_2}{1 + d x_1 x_2 y_1 y_2},$$

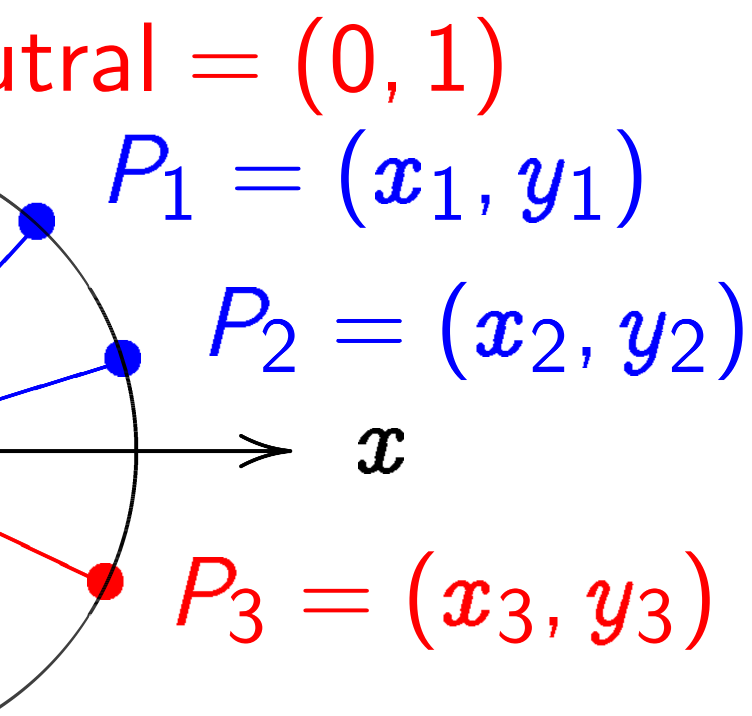
$$y_3 = \frac{y_1 y_2 - x_1 x_2}{1 - d x_1 x_2 y_1 y_2}.$$

"Hey, th

in the E

What if

or comparison:



nd (x_2, y_2) is

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

$$x_3 = \frac{x_1 y_2 + y_1 x_2}{1 + d x_1 x_2 y_1 y_2},$$

$$y_3 = \frac{y_1 y_2 - x_1 x_2}{1 - d x_1 x_2 y_1 y_2}.$$

"Hey, there are div

in the Edwards ad

What if the denon

rison:

$1)$

$x_1, y_1)$

(x_2, y_2)

(x_3, y_3)

$)$ is

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 ar

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?"

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?"

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

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?"

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

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

<u>More elliptic curves</u>

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?"

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.

iptic curves

an odd prime p.

a *non-square* $d \in \mathbf{F}_p$.

$\in \mathbf{F}_p \times \mathbf{F}_p$:

$+ y^2 = 1 + dx^2y^2\}$

mplete Edwards curve”.

lwards addition law”:

$+ (x_2, y_2) = (x_3, y_3)$

$\frac{{}_1y_2 + y_1x_2}{+ dx_1x_2y_1y_2}$,

$\frac{{}_1y_2 - x_1x_2}{- dx_1x_2y_1y_2}$.

“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 e

Choose

Choose

this is n

$x^2 + y^2$

is a safe

es

me p.

are $d \in \mathbf{F}_p$.

:

$+ dx^2y^2\}$

wards curve".

dition law":

$= (x_3, y_3)$

$\frac{\ldots 2}{\ldots y_2}$,

$\frac{\ldots 2}{\ldots y_2}$.

"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 $p = 2^{255}$ –

Choose $d = 12166$

this is non-square

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

is a safe curve for

p.

ve".

:

)

"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.

"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.

“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.

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

$-x^2 + y^2 = 1 - dx^2y^2$ 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 $\sqrt{-1} \in \mathbf{F}_p$.

here are divisions
dwards addition law!
the denominators are 0?"

Can prove that
ominators are never 0.
law is **complete**.

of relies on
g *non-square* d.

stead choose square d:
still elliptic, and
seems to work,
e are failure cases,
ploitable by attackers.
de 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.

$-x^2 + y^2 = 1 - dx^2y^2$
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 mo

Edwards
$x^2 + y^2$

Twisted
$ax^2 + y$

Weierstr
$v^2 = u^3$

Montgor
$bv^2 = u$

Many re
e.g., sub
$y = (u$
to obtai

visions
dition law!
ninators are 0?"

e that
are never 0.
mplete.

n
are d.

se square d:
c, and
work,
re cases,
by attackers.
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.

$-x^2 + y^2 = 1 - dx^2y^2$
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

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

Twisted Edwards c
$ax^2 + y^2 = 1 + dx$

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

Montgomery curve
$bv^2 = u^3 + au^2 +$

Many relationships
e.g., substitute $x =$
$y = (u - 1)/(u +$
to obtain Montgor

!

re 0?"

0.

d:

rs.

ed.

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 Edw
to obtain Montgomery.

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.

xample

$p = 2^{255} - 19$.

$d = 121665/121666$;

on-square in $\mathbf{F}_p$.

$= 1 + dx^2y^2$

curve for ECC.

$^2 = 1 - dx^2y^2$

er safe curve

e same p and d.

, the second curve

rst curve in disguise:

x in first curve

$\cdot x$, 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

$v^2 = u^3$

– 19.

55/121666;

in $\mathbf{F}_p$.

$^2y^2$

ECC.

x^2y^2

rve

and d.

nd curve

n disguise:

curve

$\sqrt{-1} \in \mathbf{F}_p$.

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 Weier

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

;

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 curv

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

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

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

$v^2 = u^3 + au + b$:
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 $u_3 = \lambda^2 - u_1 - u_2$, $v_3 = \lambda(u_1 - u_3) - v_1$, $\lambda = (3u_1^2 + a)/2v_1$;
$(u_1, v_1) + (u_1, -v_1) = \infty$;
$(u_1, v_1) + \infty = (u_1, v_1)$;
$\infty + (u_2, v_2) = (u_2, v_2)$;
$\infty + \infty = \infty$.

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

ore elliptic curves

curves:

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

Edwards curves:

${}^2 = 1 + dx^2y^2$.

rass curves:

$+ au + b$.

mery curves:

${}^3 + au^2 + u$.

lationships:

bstitute $x = u/v$,

$- 1)/(u + 1)$ in Edwards

n Montgomery.

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 ni

Montgor

the "Mo

our reco

Diffie–H

(e.g., fo

Montgor

only wit

of curve

Montgor

nP and

$\lfloor n/2 \rfloor P$

using on

with **no**

curves

$^2y^2$.

curves:

x^2y^2.

s:

.

es:

u.

s:

$= u/v$,

1) in Edwards

mery.

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 W

Montgomery-curve
the “Montgomery
our recommended
Diffie–Hellman key
(e.g., for forward s

Montgomery ladde
only with u-coordi
of curve points P.

Montgomery ladde
nP and $(n+1)P$
$\lfloor n/2 \rfloor P$ and $(\lfloor n/$
using one bit of n
with **no branches**

ards

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 u the "Montgomery ladder"— our recommended method fo Diffie–Hellman key exchange (e.g., for forward secrecy).

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

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

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**.

n on Weierstrass curves

$+ au + b$:

$\neq u_2$, $(u_1, v_1) + (u_2, v_2) =$

with $u_3 = \lambda^2 - u_1 - u_2$,

$u_1 - u_3) - v_1$,

$_2 - v_1)/(u_2 - u_1)$; for

$(u_1, v_1) + (u_1, v_1) =$

with $u_3 = \lambda^2 - u_1 - u_2$,

$u_1 - u_3) - v_1$,

$_1^2 + a)/2v_1$;

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

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

$_2, v_2) = (u_2, v_2)$;

$= \infty$.

o 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 se

Many di

1999 AN

2000 IE

2000 SE

2000 NI

2001 AN

2005 Br

2005 NS

2011 AN

Our new

http://

rstrass curves

:

$v_1) + (u_2, v_2) =$

$= \lambda^2 - u_1 - u_2,$

$- v_1,$

$u_2 - u_1)$; for

$(u_1, v_1) =$

$= \lambda^2 - u_1 - u_2,$

$- v_1,$

$1;$

$v_1) = \infty;$

$u_1, v_1);$

$u_2, v_2);$

nt 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 sta

1999 ANSI X9.62.

2000 IEEE P1363.

2000 SEC 2.

2000 NIST FIPS 1

2001 ANSI X9.63.

2005 Brainpool.

2005 NSA Suite B

2011 ANSSI FRP2

Our new evaluatio

http://safecurv

ves

$v_2) =$

$- u_2,$

for

$=$

$- u_2,$

t.

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.y

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`

cer than Weierstrass:

mery-curve ECDH using
ontgomery ladder"—
mmended method for
ellman key exchange
r forward secrecy).

mery ladder works
h u-coordinates
points P.

mery ladder computes
$(n+1)P$ recursively from
and $(\lfloor n/2 \rfloor + 1)P$
e bit of n
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

The curv

The nun
must be
a large p
Standard
$\ell \approx 2^{200}$
$\ell \approx 2^{256}$

ℓ must n
p; $p-1$
p^3-1;
This gua
no "tran

Weierstrass:

e ECDH using
ladder"—
method for
y exchange
secrecy).

er works
inates

er computes
recursively from
$2\rfloor + 1)P$

.

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 at

The curve must be

The number of cu
must be divisible b
a large prime num
Standard attacks t
$\ell \approx 2^{200}$ is adequa
$\ell \approx 2^{256}$ is conserv

ℓ must not divide
p; $p-1$; p^2-1;
p^3-1; $\ldots$; $p^{20}-$
This guarantees th
no "transfers" to

s:

sing

-

or

e

es

y from

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
$\ell \approx 2^{200}$ is adequate;
$\ell \approx 2^{256}$ is conservative.

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

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; ...; $p^{20}-1$.
This guarantees that there are no "transfers" to clocks etc.

election

fferent standards:

NSI X9.62.

EE P1363.

C 2.

ST FIPS 186-2.

NSI X9.63.

ainpool.

SA Suite B.

NSSI FRP256V1.

evaluation site:

/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; ...; $p^{20}-1$.
This guarantees that there are no "transfers" to clocks etc.

Avoiding

Simplify

avoid po

even if n

Require

discrimin

SafeCurv

Brainpoo

SafeCurv

Brainpoo

prohibit

$p^k - 1$ f

ndards:

186-2.

3.
256V1.

on site:
ves.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; ...; $p^{20}-1$. This guarantees that there are no "transfers" to clocks etc.

Avoiding unnecess

Simplify the securi avoid possible atta even if no attacks

Require large "CM discriminant". See SafeCurves.

Brainpool, Suite B SafeCurves: requir

Brainpool and Saf prohibit ℓ dividing $p^k - 1$ for each k

p.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 structu

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 $k < (\ell - 1)$/

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 $k < (\ell - 1)/100$.

g known attacks

ve must be elliptic.

nber of curve points
divisible by
prime number ℓ.
d attacks take time $\sqrt{\ell}$.
is adequate;
is conservative.

not divide
; $p^2 - 1$;
. . . ; $p^{20} - 1$.
arantees that there are
nsfers" 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
of securi
- there's
  a smal
- public
  has mi
- the at
  figured
- the at
  choice
  to allo

ttacks

e elliptic.

rve points
oy
ber ℓ.
take time $\sqrt{\ell}$.
te;
vative.

1.
hat there are
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 conceivab
of security problen

- there's another a
  a small fraction
- public ECC cryp
  has missed this a
- the attacker has
  figured out this
- the attacker has
  choices of stand
  to allow the atta

$\sqrt{\ell}$.

are

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 aga
  a small fraction of curves;
- public ECC cryptanalysis has missed this attack;
- the attacker has figured out this attack;
- the attacker has **manipula**
  choices of standard curves
  to allow the attack.

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.

g unnecessary structure

the security story:
ossible attack vectors
no attacks are known.

large “CM field
nant”. See, e.g.,
ves.

ol, Suite B, ANSSI,
ves: require prime p.

ol and SafeCurves:
ℓ dividing
or 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 cu
“verifiab
$y^2 = x^3$
b is deriv
SHA-1 h

sary structure

ity story:
ack vectors
are known.

1 field
e, e.g.,

3, ANSSI,
re prime p.

feCurves:

$< (\ell - 1)/100$.

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
"verifiably random
$y^2 = x^3 - 3x + b$
b is derived from
SHA-1 hash of a p

ure

s

n.

.

/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

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.

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.

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!

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.

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.

conceivable source
ty problems:
s another attack against
ll fraction of curves;
ECC cryptanalysis
issed this attack;
tacker has
d out this attack;
tacker has **manipulated**
s of standard curves
w 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.

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
that can
by a cur

Brainpoo
b is som
of digits

ble source
ns:
attack against
of curves;
tanalysis
attack;

attack;
manipulated
ard curves
ack.

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 nu
that can be genera
by a curve-generat

Brainpool, somew
b is some sort of h
of digits of π and

inst

ated

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 cu that can be generated by a curve-generation proces

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

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.

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.

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

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.

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.

rves claim to be
ly random":
$- 3x + b$ where
ved from
ash of a public seed.

he seed actually random?

could have tried
eds to find a curve with
-a-billion weakness.
rifiable" at all!

esponse: use our
n" 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.

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 sec

Covered
hard to
secret ke

But real-
is still be

ECC imp
- produc
  for sor
- leak se
  for inp
- leak se
  throug

etc. Att

n to be
n":
where

public seed.

tually random?

ve tried
d a curve with
weakness.
t all!

use our
nstead.

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 E
secret key from pu

But real-world EC
is still being broke

ECC implementati
- produce incorrec
  for some rare inp
- leak secret data
  for input points
- leak secret data
  through timing;

etc. Attackers exp

l.

dom?

with

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.

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.

limits number of curves
be generated
ve-generation process.

ol, somewhat rigid:
e sort of hash
of π and e.

npletely explained:
particular hash?
nd not $\sqrt{2}$? etc.
much flexibility.

ommendation, fully rigid:
llest positive integer
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 c
allow **si**
to be **se**

This is t
motivati

mber of curves
ated
tion process.

hat rigid:
hash
e.

plained:
r hash?
2? etc.
kibility.

ion, fully rigid:
ive integer
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 c
allow **simple** imple
to be **secure** imple

This is the primary
motivation for Saf

urves

ss.

rigid:

r

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** implementatio to be **secure** implementatio

This is the primary motivation for SafeCurves.

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.

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.

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.

urity

so far:
compute ECC user's
ey from public key.

-world ECC
eing broken!

plementations
ce incorrect results
me rare inputs;
ecret data
put points *off* curve;
ecret data
gh timing;
ackers exploit this.

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.

Curve
Anomalous
M-221
E-222
NIST P-224
Curve1174
Curve25519
BN(2,254)
brainpoolP2
ANSSI FRP25
NIST P-256
secp256k1
E-382
M-383
Curve38318
brainpoolP3
NIST P-384
Curve3617

ECC user's
ublic key.

C
n!

ons
t results
puts;

off curve;

loit this.

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.

Curve	Safe?	
Anomalous	False	Tr
M-221	True ✔	Tr
E-222	True ✔	Tr
NIST P-224	False	Tr
Curve1174	True ✔	Tr
Curve25519	True ✔	Tr
BN(2,254)	False	Tr
brainpoolP256t1	False	Tr
ANSSI FRP256v1	False	Tr
NIST P-256	False	Tr
secp256k1	False	Tr
E-382	True ✔	Tr
M-383	True ✔	Tr
Curve383187	True ✔	Tr
brainpoolP384t1	False	Tr
NIST P-384	False	Tr
Curve3617	True ✔	Tr

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.

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

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 ✔

hoices of curves
mple implementations
cure implementations.

he primary
on for SafeCurves.

e of new requirement:
curity.

isn't twist-secure:
ttacks break
mplementations
n't check whether
oint is on curve.
-simplicity conflict.

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

curves
ementations
ementations.

y
eCurves.

equirement:

-secure:
ak
tions
vhether
curve.
 conflict.

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

ns

ns.

t:

		Parameters:			ECDLP security:				ECC secu		
Curve	**Safe?**	**field**	**equation**	**base**	**rho**	**transfer**	**disc**	**rigid**	**ladder**	**twist**	**com**
Anomalous	False	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	False	False	Fal
M-221	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	Tru
E-222	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	Tru
NIST P-224	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	False	Fal
Curve1174	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	Tru
Curve25519	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	Tru
BN(2,254)	False	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	False	False	Fal
brainpoolP256t1	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	Fal
ANSSI FRP256v1	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	False	Fal
NIST P-256	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	Fal
secp256k1	False	True ✔	True ✔	True ✔	True ✔	True ✔	False	True ✔	False	True ✔	Fal
E-382	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	Tru
M-383	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	Tru
Curve383187	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	Tru
brainpoolP384t1	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	True ✔	Fal
NIST P-384	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	Fal
Curve3617	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	Tru

		Parameters:			ECDLP security:				ECC security:			
Curve	**Safe?**	field	equation	base	rho	transfer	disc	rigid	ladder	twist	complete	ind
Anomalous	False	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	False	False	False	False
M-221	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
E-222	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
NIST P-224	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	False	False	False
Curve1174	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
Curve25519	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
BN(2,254)	False	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	False	False	False	False
brainpoolP256t1	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	False	False
ANSSI FRP256v1	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	False	False	False
NIST P-256	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	False	False
secp256k1	False	True ✔	True ✔	True ✔	True ✔	True ✔	False	True ✔	False	True ✔	False	False
E-382	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
M-383	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
Curve383187	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
brainpoolP384t1	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	True ✔	False	False
NIST P-384	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	False	False
Curve3617	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔