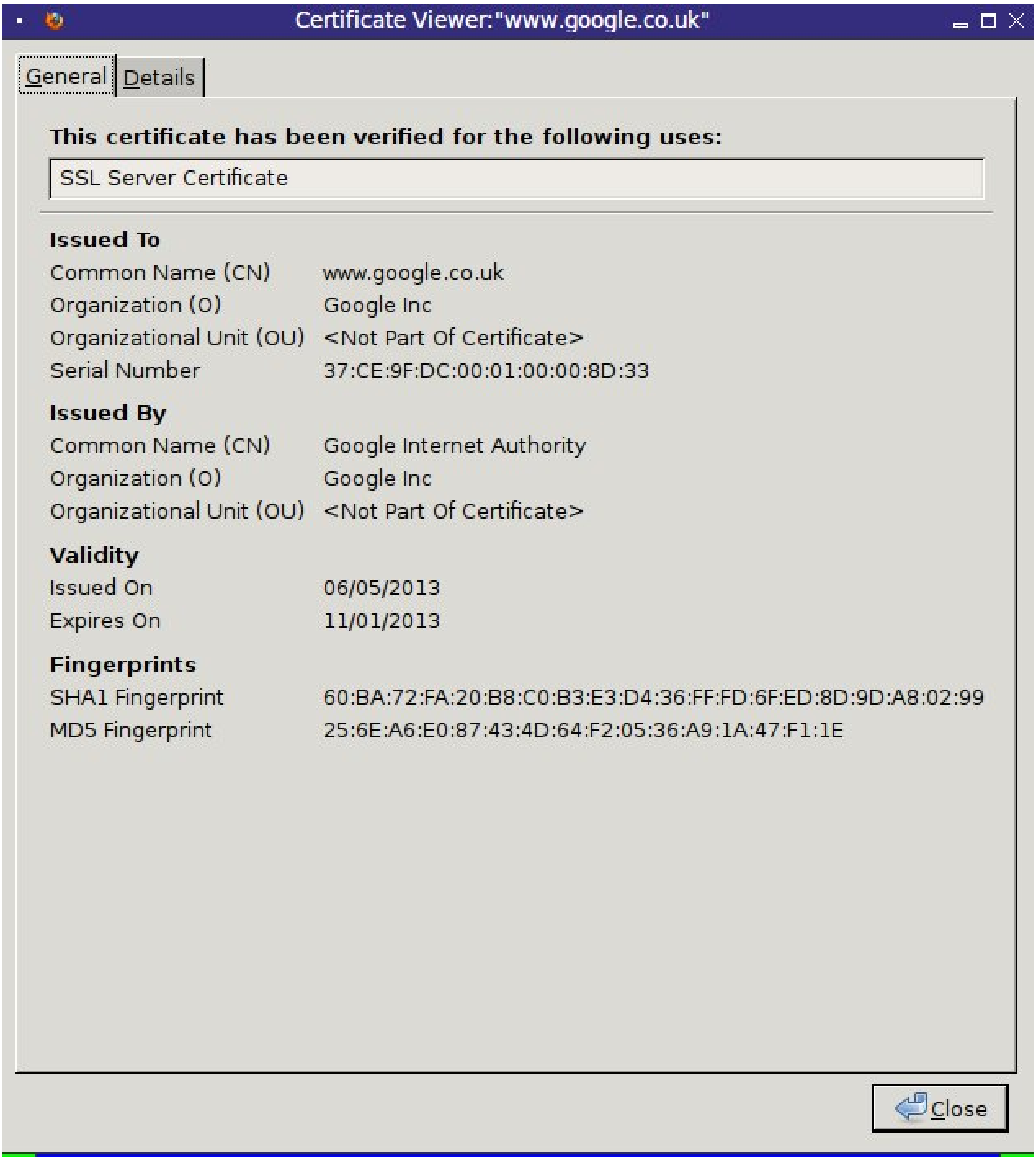
Public-key cryptography and the Discrete-Logarithm Problem

Tanja Lange
Technische Universiteit Eindhoven

Cryptography

Let’s understand what our browsers do.



This certificate has been verified for the following uses:

SSL Server Certificate

Issued To

Common Name (CN)	www.google.co.uk
Organization (O)	Google Inc
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	37:CE:9F:DC:00:01:00:00:8D:33

Issued By

Common Name (CN)	Google Internet Authority
Organization (O)	Google Inc
Organizational Unit (OU)	<Not Part Of Certificate>

Validity

Issued On	06/05/2013
Expires On	11/01/2013

Fingerprints

SHA1 Fingerprint	60:BA:72:FA:20:B8:C0:B3:E3:D4:36:FF:FD:6F:ED:8D:9D:A8:02:99
MD5 Fingerprint	25:6E:A6:E0:87:43:4D:64:F2:05:36:A9:1A:47:F1:1E

Schoolbook RSA

Pick primes p, q of same bitlength,
at least 512 bits
(2048 to feel secure).

Compute $N = p \cdot q$.
Compute $\phi(N) = (p-1)(q-1)$.
Pick e with $\gcd\{e, \phi(N)\} = 1$.
Compute e with

$$ed \equiv 1 \pmod{\phi(N)}.$$

Public key is (e, N),
secret key is d.
Some systems keep p and q,
see exercises.

Encryption of message $m < N$:

Compute $c = m^e \pmod{N}$.

Decryption of ciphertext c:

Encryption of message $m < N$:

Compute $c = m^e \pmod{N}$.

Decryption of ciphertext c:

Compute

$$c^d \equiv (m^e)^d \equiv m^{k\phi(N)+1} \equiv m$$

modulo N.

Encryption of message $m < N$:

Compute $c = m^e \pmod{N}$.

Decryption of ciphertext c:

Compute

$$c^d \equiv (m^e)^d \equiv m^{k\phi(N)+1} \equiv m$$

modulo N.

Signature on message m:

Uses cryptographic hash function

$h : \{0,1\}^* \to \mathbf{Z}/N$

Compute

$s = (h(m))^d \pmod{N}$.

Verify signature by comparing

$h(m)$ with $s^e \pmod{N}$.

Problems with schoolbook RSA

Encryption is deterministic:

- attacker can test candidate message;
- repeated messages are recognized.

Small e is dangerous for small m (no effect of modular reduction).

More number-theoretic fun, see exercises.

Encryption is homomorphic: encryption of $m_1 m_2$ is $c_1 c_2 \pmod{N}$.

Modern cryptography:
allow attacker to use oracles
for decryption or signatures.
Can query anything
except for target.

Use this to decrypt c:

Modern cryptography:
allow attacker to use oracles
for decryption or signatures.
Can query anything
except for target.

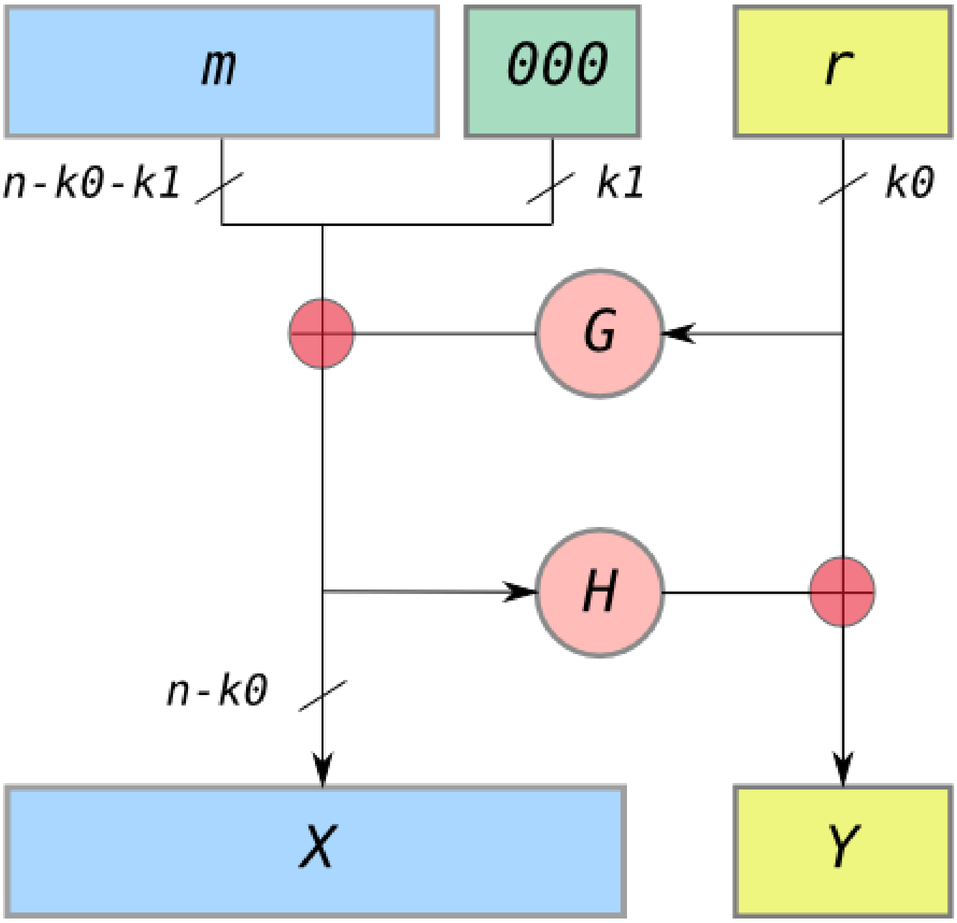
Use this to decrypt c:
Pick random m'.
Ask oracle to decrypt
$c' = (m')^e c \pmod{N}$.
Get message by dividing my m'.

RSA-OAEP

Optimal asymmetric encryption padding, included in PKCS #1v2.
Formats message (before RSA).
Formatting unlikely
to survive multiplication.
If format is incorrect
decryption will fail.

Let $n = \lfloor \log_2 N \rfloor$.
Algorithm uses parameters k_0, k_1,
messages are in $\{0,1\}^{n-k_0-k_1}$.
Uses two hash functions G, H:
$G : \{0,1\}^{k_0} \to \{0,1\}^{n-k_0}$
$H : \{0,1\}^{n-k_0} \to \{0,1\}^{k_0}$.

1. Pad m with k_1 zeros.
2. Pick random $r \in \{0,1\}^{k_0}$.
3. Compute
$$X = m00\ldots0 \oplus G(r).$$
4. Compute $Y = r \oplus H(X)$.
5. Output X, Y.



Credit: Ozga at en.wikipedia

What does your browser do?

1. Check X.509 certificate:
RSA signature verification.
2. OAEP format
random message m;
3. RSA encrypt
resulting message $M = X, Y$
(interpreted as number mod N).
4. Send ciphertext to server.
5. Derive encryption and
authentication keys from m.
6. Use these for the bulk
encryption.

Google uses RC4 for encryption;
other common choice: AES.

Authenticated encryption

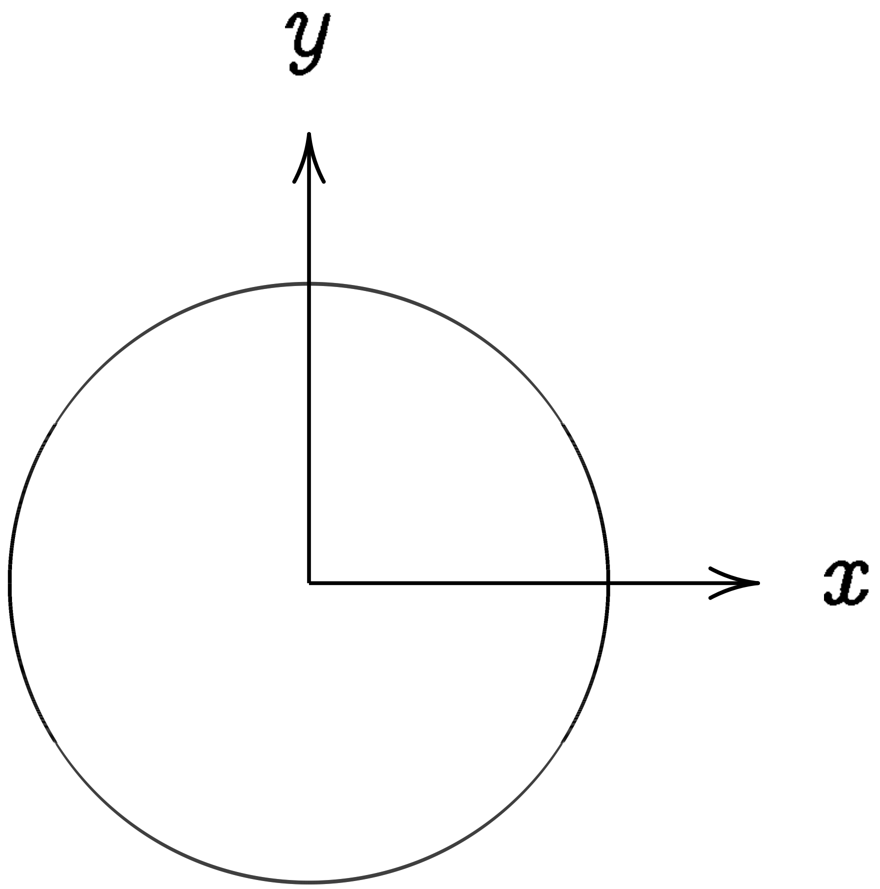
Authentication key computes tag on message so that any change makes tag invalid.

Cannot prove authenticity to third party.
Convinces owners of secret key that they are communicating with one another.

Typical examples:

- RC4-HMAC (keyed-hash message authentication code)
- AES-GCM (Galois Counter Mode)

The clock



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

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

Examples of points on this curve:

Examples of points on this curve:

$(0, 1) =$ “12:00”.

Examples of points on this curve:

$(0, 1) =$ “12:00”.

$(0, -1) =$ “6:00”.

Examples of points on this curve:

$(0, 1) =$ “12:00”.

$(0, -1) =$ “6:00”.

$(1, 0) =$ “3:00”.

Examples of points on this curve:

$(0, 1) =$ “12:00”.

$(0, -1) =$ “6:00”.

$(1, 0) =$ “3:00”.

$(-1, 0) =$ “9:00”.

Examples of points on this curve:

$(0, 1) =$ “12:00”.

$(0, -1) =$ “6:00”.

$(1, 0) =$ “3:00”.

$(-1, 0) =$ “9:00”.

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

Examples of points on this curve:

$(0, 1) =$ “12:00”.

$(0, -1) =$ “6:00”.

$(1, 0) =$ “3:00”.

$(-1, 0) =$ “9:00”.

$(\sqrt{3/4}, 1/2) =$ “2:00”.

Examples of points on this curve:

$(0, 1) =$ “12:00”.

$(0, -1) =$ “6:00”.

$(1, 0) =$ “3:00”.

$(-1, 0) =$ “9:00”.

$(\sqrt{3/4}, 1/2) =$ “2:00”.

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

Examples of points on this curve:

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

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

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

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

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

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

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

Examples of points on this curve:

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

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

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

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

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

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

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

Examples of points on this curve:

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

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

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

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

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

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

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

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

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

Examples of points on this curve:

$(0, 1) =$ “12:00”.

$(0, -1) =$ “6:00”.

$(1, 0) =$ “3:00”.

$(-1, 0) =$ “9:00”.

$(\sqrt{3/4}, 1/2) =$ “2:00”.

$(1/2, -\sqrt{3/4}) =$ “5:00”.

$(-1/2, -\sqrt{3/4}) =$ “7:00”.

$(\sqrt{1/2}, \sqrt{1/2}) =$ “1:30”.

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

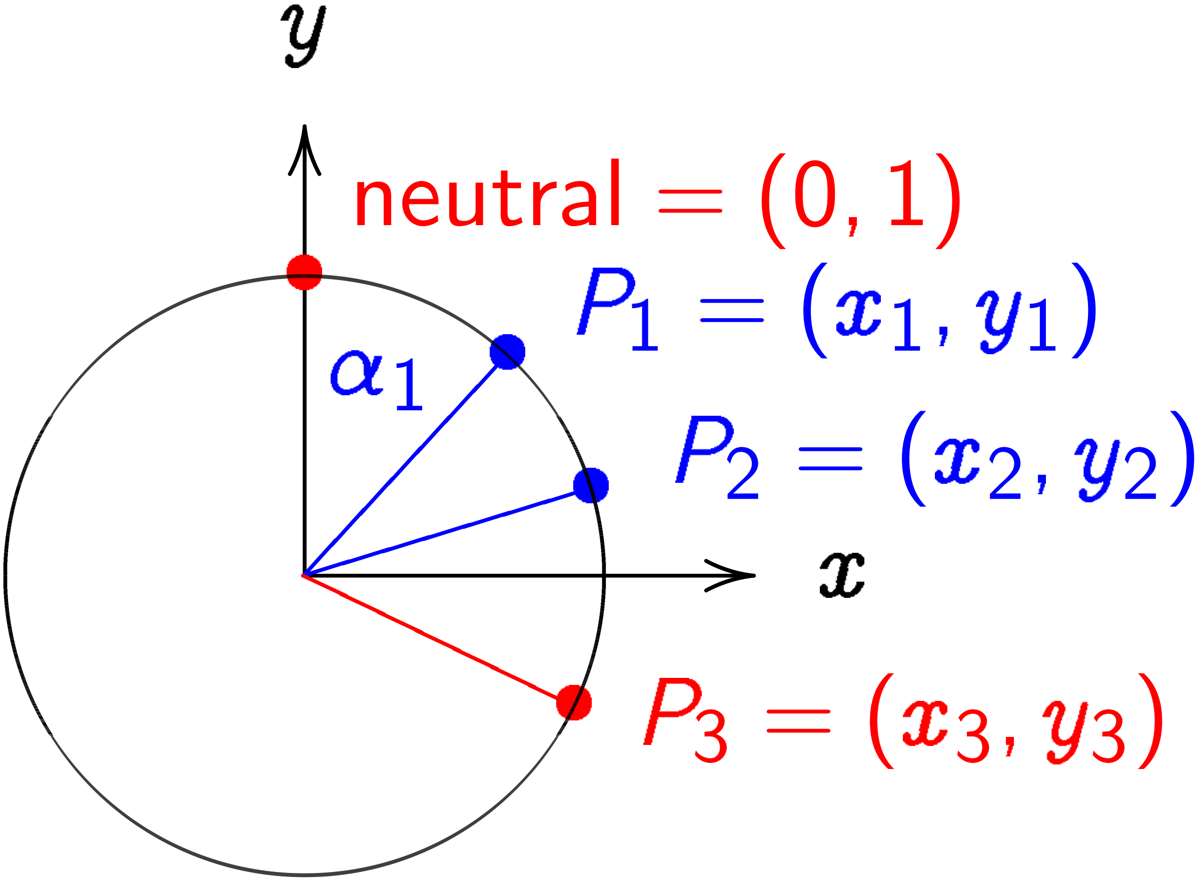
$(3/5, -4/5)$. $(-3/5, -4/5)$.

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

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

Many more.

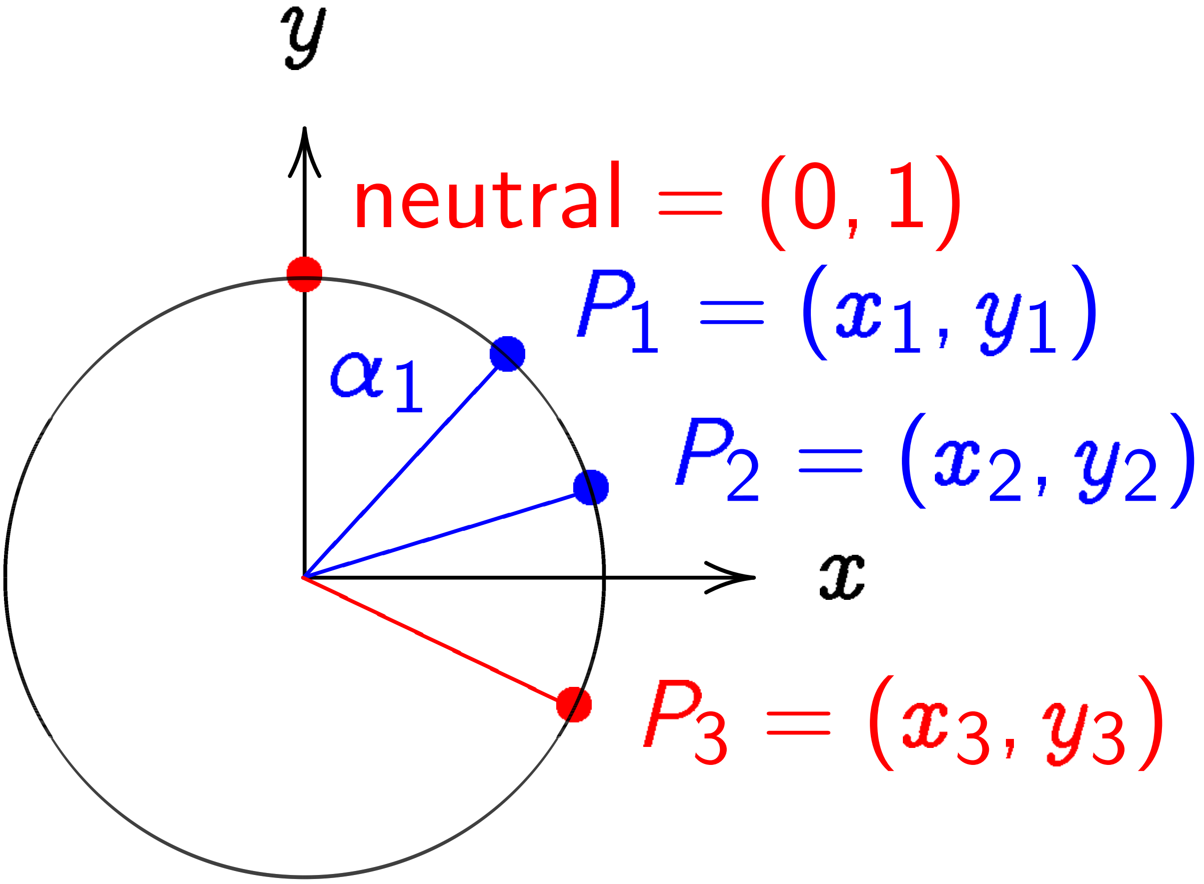
Addition on the clock:



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

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

Addition on the clock:

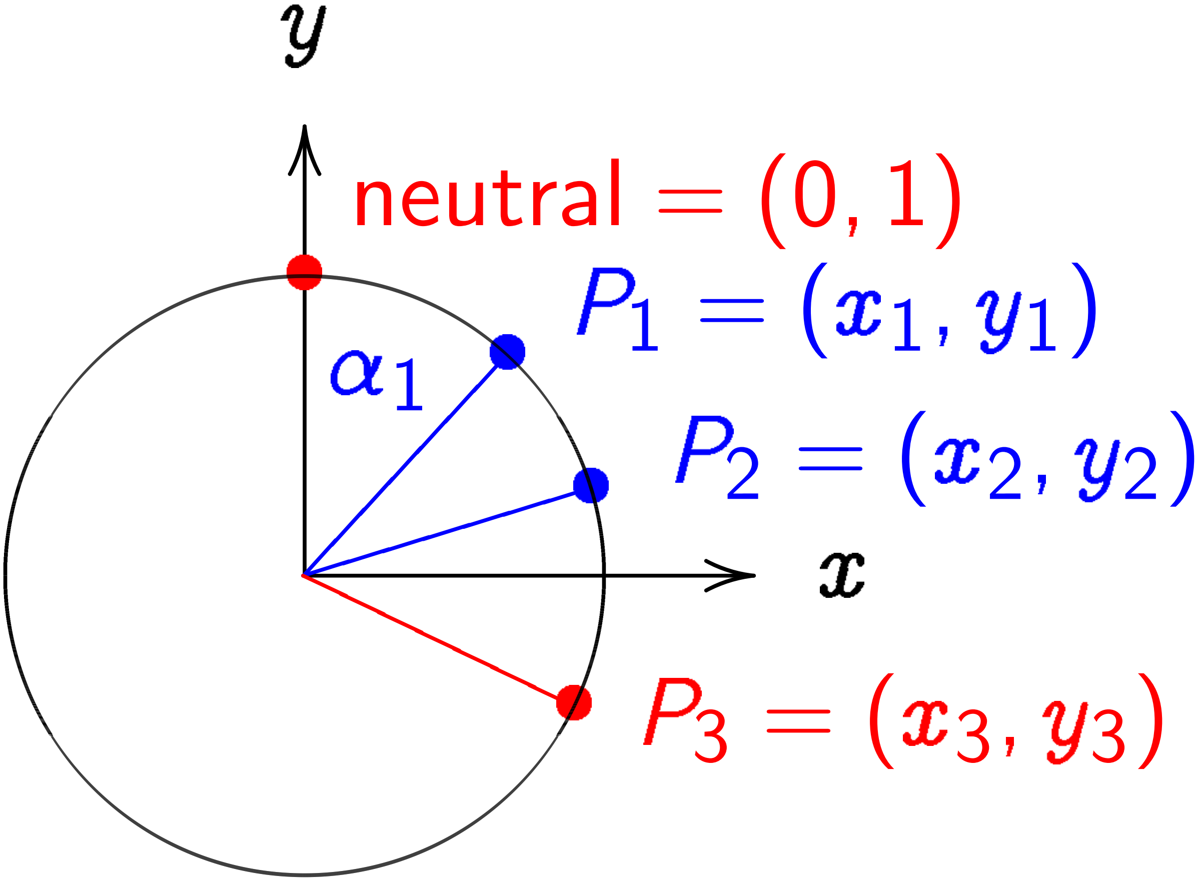


$x^2 + y^2 = 1$, parametrized by

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

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

Addition on the clock:



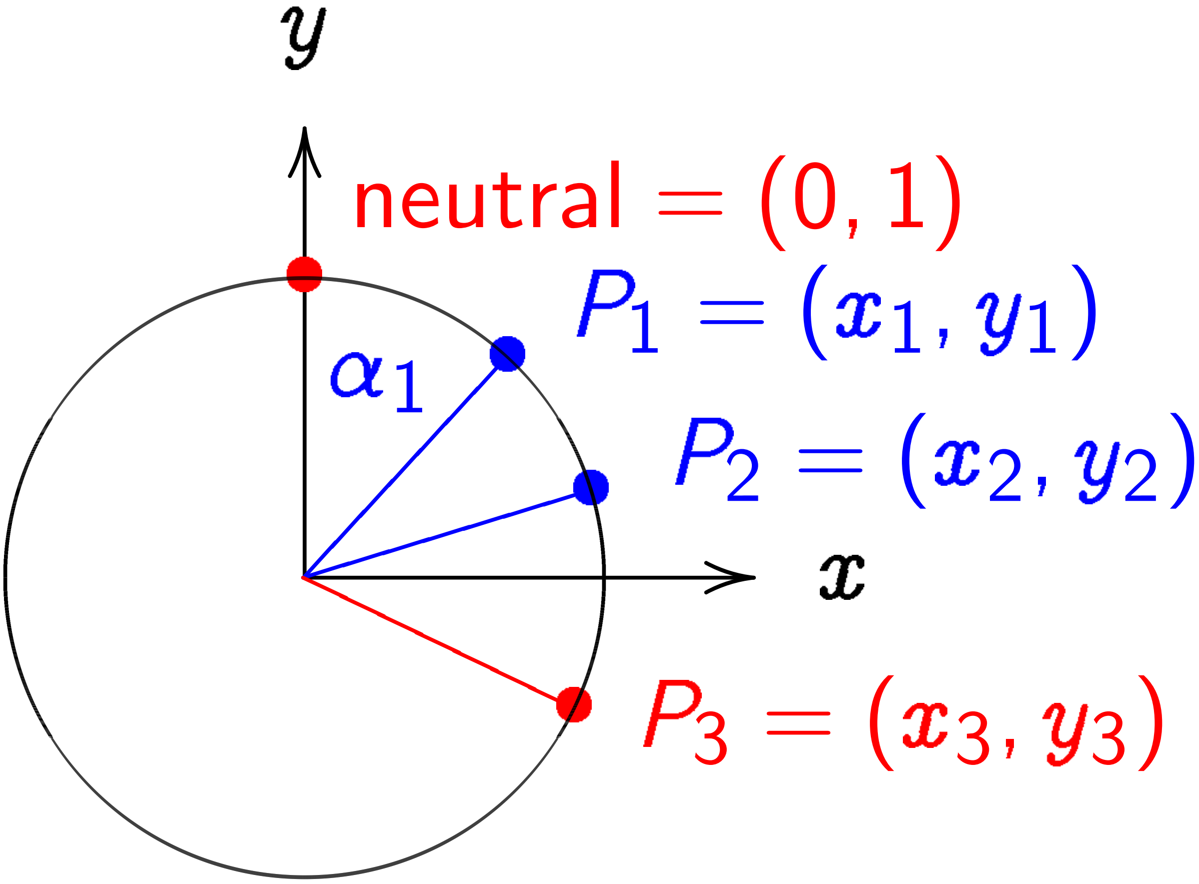
$x^2 + y^2 = 1$, parametrized by

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

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

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

Addition on the clock:

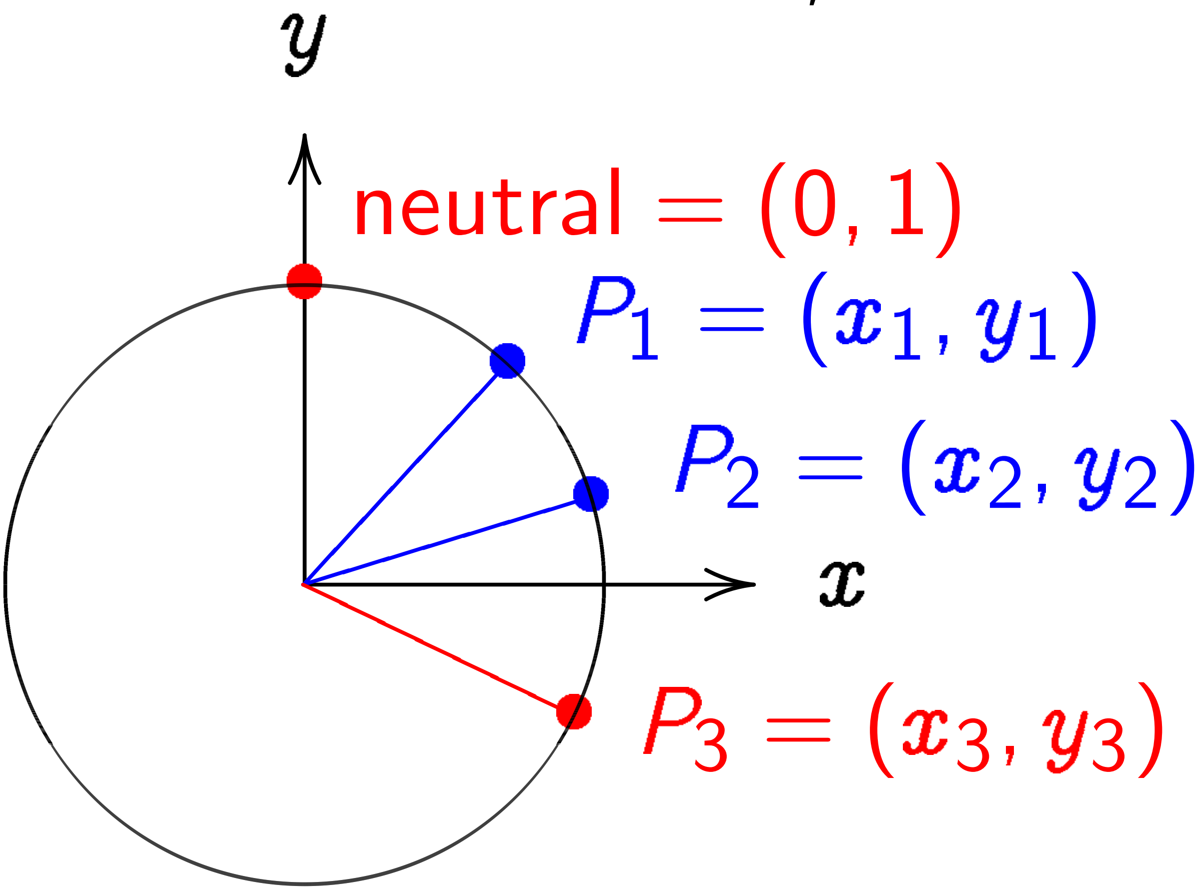


$x^2 + y^2 = 1$, parametrized by $x = \sin\alpha, \quad y = \cos\alpha$. Recall $(\sin(\alpha_1 + \alpha_2), \cos(\alpha_1 + \alpha_2)) = (\sin\alpha_1\cos\alpha_2 + \cos\alpha_1\sin\alpha_2,$
$\cos\alpha_1\cos\alpha_2 - \sin\alpha_1\sin\alpha_2)$.

Adding two points corresponds
to adding the angles α_1 and α_2.
Angles modulo 360° are a group,
so points on clock are a group.

Neutral element: angle $\alpha = 0$;
point $(0, 1)$; “12:00”.
The point with $\alpha = 180^\circ$
has order 2 and equals 6:00.
3:00 and 9:00 have order 4.
Inverse of point with α
is point with $-\alpha$
since $\alpha + (-\alpha) = 0$.
There are many more points
where angle α is not “nice.”

Clock addition without sin, cos:



Use Cartesian coordinates for addition. Addition formula for the clock $x^2 + y^2 = 1$:

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

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

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

$kP = \underbrace{P + P + \cdots + P}_{k \text{ copies}}$ for $k \geq 0$.

Examples of clock addition:

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

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

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

Examples of clock addition:

“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”.

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

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

Examples of clock addition:

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

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

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

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

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

Examples of clock addition:

“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”.

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

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

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

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

Examples of clock addition:

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

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

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

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

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

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

Examples of clock addition:

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

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

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

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

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

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

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

Examples of clock addition:

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

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

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

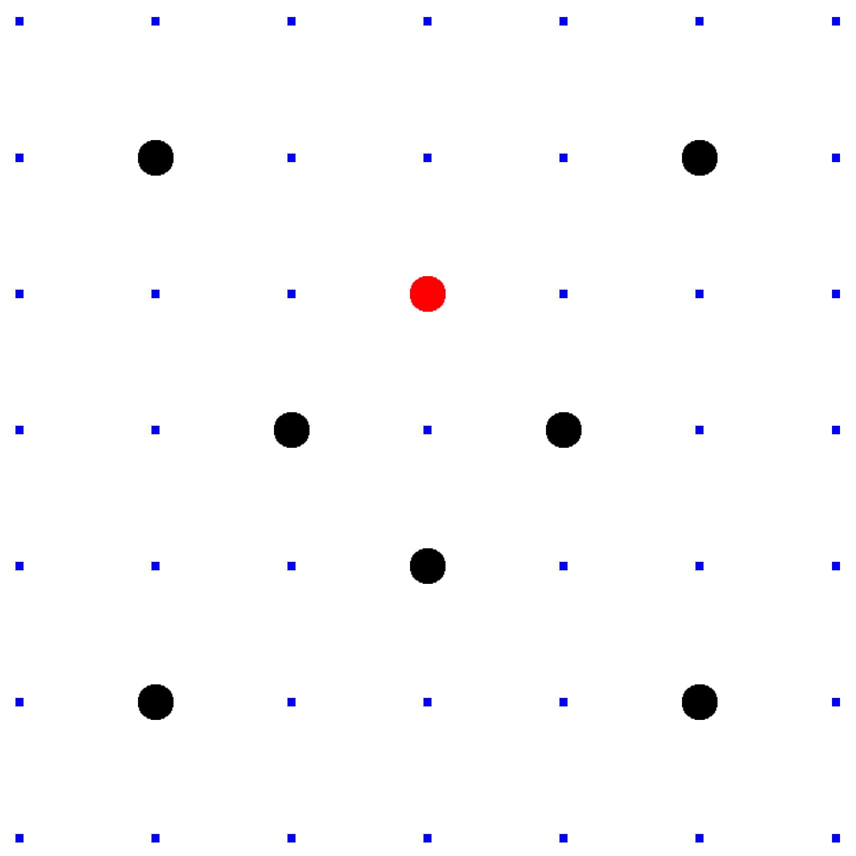
$$3\left(\frac{3}{5}, \frac{4}{5}\right) = \left(\frac{117}{125}, \frac{-44}{125}\right).$$

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

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

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

Clocks over finite fields



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

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

Examples of clock addition:

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

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

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

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

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

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

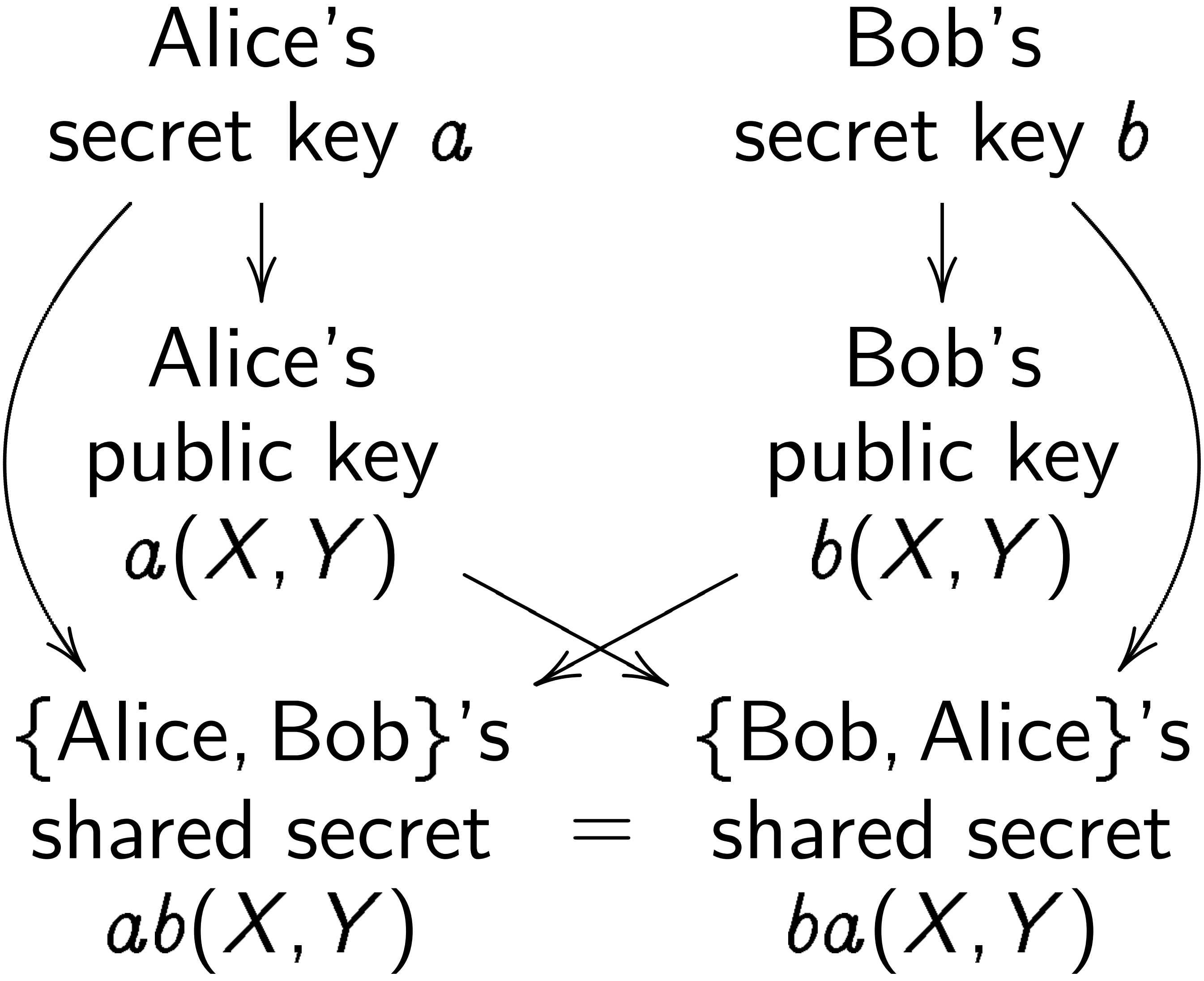
Clock cryptography

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 $b(a(X, Y))$.
I.e., both obtain $(ab)(X, Y)$.
They use this shared value
to encrypt with AES-GCM etc.



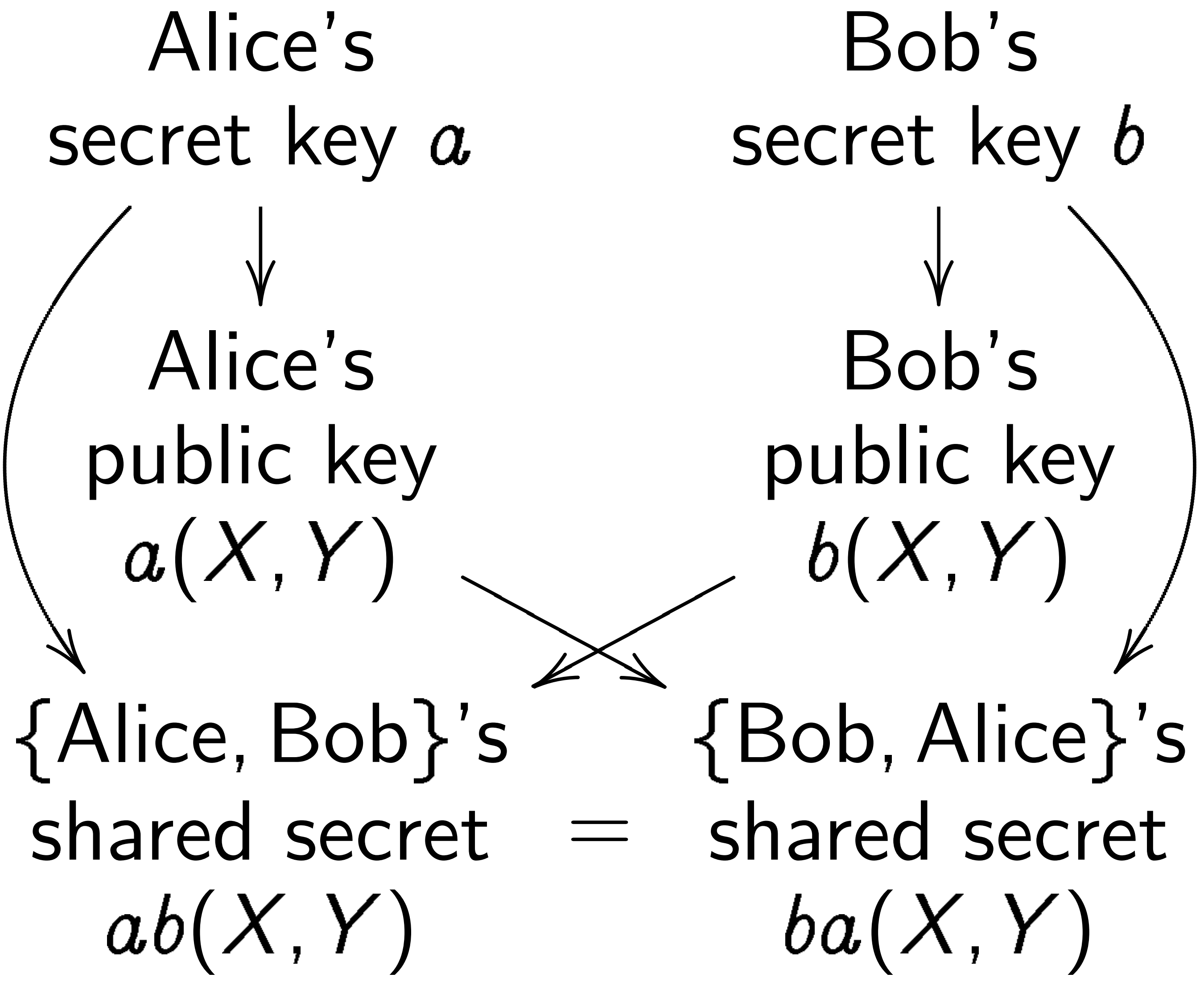
Alice's secret key a

Bob's secret key b

Alice's public key $a(X,Y)$

Bob's public key $b(X,Y)$

{Alice, Bob}'s shared secret $ab(X,Y)$ = {Bob, Alice}'s shared secret $ba(X,Y)$



Alice's secret key a

Bob's secret key b

Alice's public key $a(X, Y)$

Bob's public key $b(X, Y)$

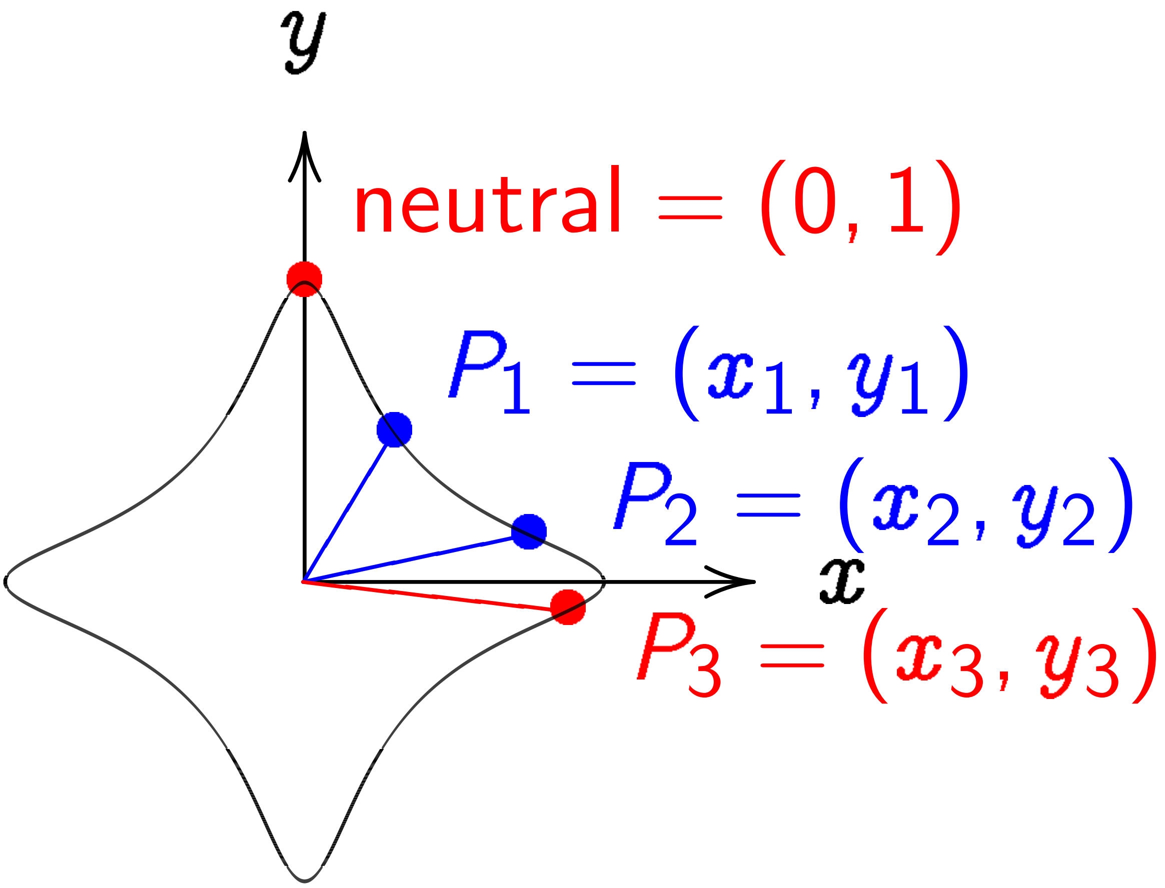
{Alice, Bob}'s shared secret $ab(X, Y)$ $=$ {Bob, Alice}'s shared secret $ba(X, Y)$

Warning: Clocks aren't elliptic! Can attack clock cryptography, e.g., compute a from public key, by combining congruences.

To match RSA-3072 security need $p \approx 2^{1536}$.

Addition on an Edwards curve

Change the curve on which Alice and Bob work.

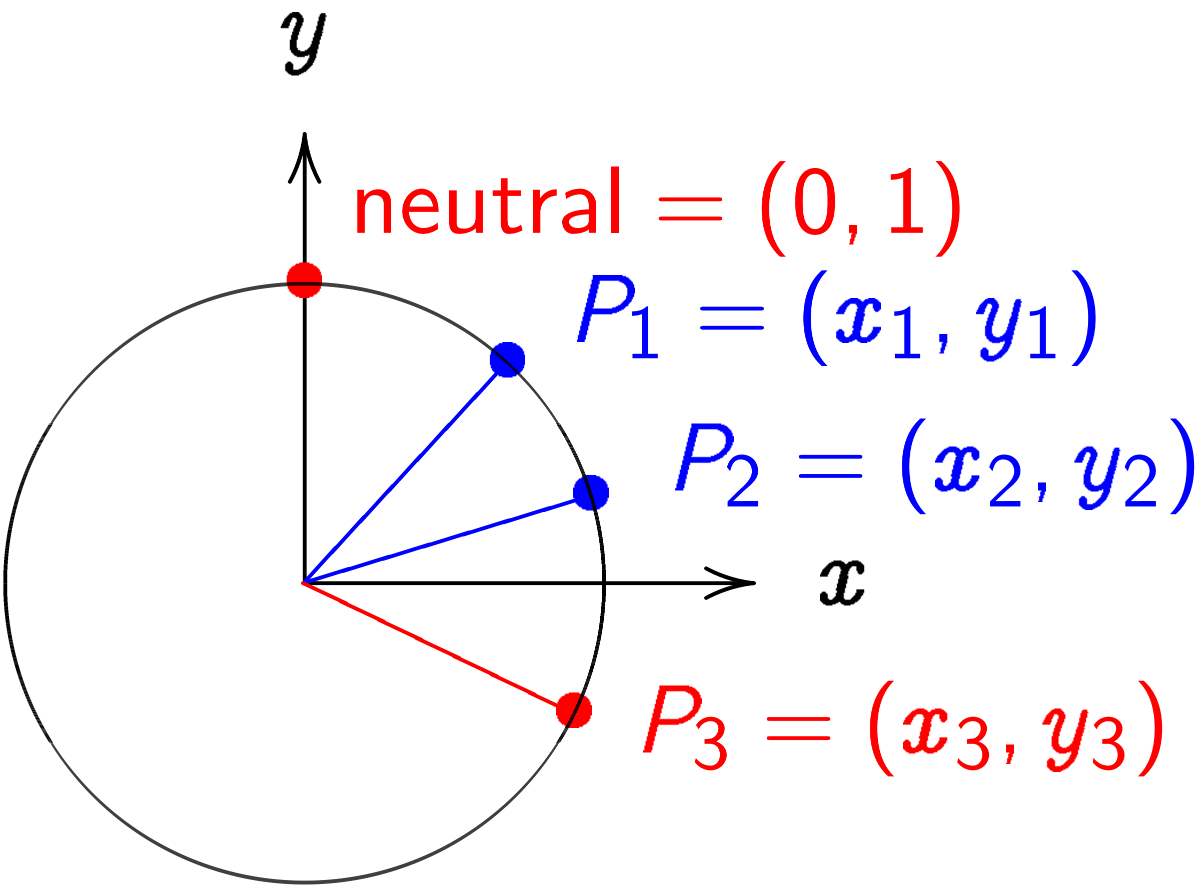


$x^2 + y^2 = 1 - 30x^2y^2$.

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

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

The clock again, for comparison:



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

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

$(x_1 y_2 + y_1 x_2,$

$y_1 y_2 - x_1 x_2)$.

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

Answer: They aren’t!

If $x_i = 0$ or $y_i = 0$ then
$1 \pm 30x_1x_2y_1y_2 = 1 \neq 0$.

If $x^2 + y^2 = 1 - 30x^2y^2$
then $30x^2y^2 < 1$
so $\sqrt{30}\ |xy| < 1$.

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

Answer: They aren’t!

If $x_i = 0$ or $y_i = 0$ then $1 \pm 30x_1x_2y_1y_2 = 1 \neq 0$.

If $x^2 + y^2 = 1 - 30x^2y^2$ then $30x^2y^2 < 1$ so $\sqrt{30}\;|xy| < 1$.

If $x_1^2 + y_1^2 = 1 - 30x_1^2y_1^2$ and $x_2^2 + y_2^2 = 1 - 30x_2^2y_2^2$ then $\sqrt{30}\;|x_1y_1| < 1$ and $\sqrt{30}\;|x_2y_2| < 1$

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

Answer: They aren’t!

If $x_i = 0$ or $y_i = 0$ then
$1 \pm 30x_1x_2y_1y_2 = 1 \neq 0$.

If $x^2 + y^2 = 1 - 30x^2y^2$
then $30x^2y^2 < 1$
so $\sqrt{30}\ |xy| < 1$.

If $x_1^2 + y_1^2 = 1 - 30x_1^2y_1^2$
and $x_2^2 + y_2^2 = 1 - 30x_2^2y_2^2$
then $\sqrt{30}\ |x_1y_1| < 1$
and $\sqrt{30}\ |x_2y_2| < 1$
so $30\,|x_1y_1x_2y_2| < 1$
so $1 \pm 30x_1x_2y_1y_2 > 0$.

The Edwards addition law

$(x_1, y_1) + (x_2, y_2) =$
$((x_1y_2{+}y_1x_2)/(1{-}30x_1x_2y_1y_2),$
$(y_1y_2{-}x_1x_2)/(1{+}30x_1x_2y_1y_2))$
is a group law for the curve
$x^2 + y^2 = 1 - 30x^2y^2$.

Some calculation required:
addition result is on curve;
addition law is associative.

Other parts of proof are easy:
addition law is commutative;
$(0, 1)$ is neutral element;
$(x_1, y_1) + (-x_1, y_1) = (0, 1)$.

More Edwards curves

Fix an odd prime power q.
Fix a *non-square* $d \in \mathbf{F}_q$.

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

is a commutative group with $(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$ defined by Edwards addition law:

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

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

Denominators are never 0.

But need different proof;
“$x^2 + y^2 > 0$” doesn’t work.

Denominators are never 0.

But need different proof; “$x^2 + y^2 > 0$” doesn’t work.

If $x_1^2 + y_1^2 = 1 + dx_1^2 y_1^2$
and $x_2^2 + y_2^2 = 1 + dx_2^2 y_2^2$
and $dx_1 x_2 y_1 y_2 = \pm 1$

Denominators are never 0.

But need different proof;

“$x^2 + y^2 > 0$” doesn’t work.

If $x_1^2 + y_1^2 = 1 + dx_1^2 y_1^2$
and $x_2^2 + y_2^2 = 1 + dx_2^2 y_2^2$
and $dx_1 x_2 y_1 y_2 = \pm 1$
then $dx_1^2 y_1^2 (x_2 + y_2)^2$
$= dx_1^2 y_1^2 (x_2^2 + y_2^2 + 2x_2 y_2)$

Denominators are never 0.

But need different proof;
"$x^2 + y^2 > 0$" doesn't work.

If $x_1^2 + y_1^2 = 1 + dx_1^2 y_1^2$
and $x_2^2 + y_2^2 = 1 + dx_2^2 y_2^2$
and $dx_1 x_2 y_1 y_2 = \pm 1$
then $dx_1^2 y_1^2 (x_2 + y_2)^2$
$= dx_1^2 y_1^2 (x_2^2 + y_2^2 + 2x_2 y_2)$
$= dx_1^2 y_1^2 (dx_2^2 y_2^2 + 1 + 2x_2 y_2)$

Denominators are never 0.

But need different proof;

"$x^2 + y^2 > 0$" doesn't work.

If $x_1^2 + y_1^2 = 1 + dx_1^2 y_1^2$
and $x_2^2 + y_2^2 = 1 + dx_2^2 y_2^2$
and $dx_1 x_2 y_1 y_2 = \pm 1$
then $dx_1^2 y_1^2 (x_2 + y_2)^2$
$= dx_1^2 y_1^2 (x_2^2 + y_2^2 + 2x_2 y_2)$
$= dx_1^2 y_1^2 (dx_2^2 y_2^2 + 1 + 2x_2 y_2)$
$= d^2 x_1^2 y_1^2 x_2^2 y_2^2 + dx_1^2 y_1^2 + 2dx_1^2 y_1^2 x_2 y_2$

Denominators are never 0.

But need different proof;
"$x^2 + y^2 > 0$" doesn't work.

If $x_1^2 + y_1^2 = 1 + dx_1^2 y_1^2$
and $x_2^2 + y_2^2 = 1 + dx_2^2 y_2^2$
and $dx_1 x_2 y_1 y_2 = \pm 1$
then $dx_1^2 y_1^2 (x_2 + y_2)^2$
$= dx_1^2 y_1^2 (x_2^2 + y_2^2 + 2x_2 y_2)$
$= dx_1^2 y_1^2 (dx_2^2 y_2^2 + 1 + 2x_2 y_2)$
$= d^2 x_1^2 y_1^2 x_2^2 y_2^2 + dx_1^2 y_1^2 + 2dx_1^2 y_1^2 x_2 y_2$
$= 1 + dx_1^2 y_1^2 \pm 2x_1 y_1$

Denominators are never 0.

But need different proof;

"$x^2 + y^2 > 0$" doesn't work.

If $x_1^2 + y_1^2 = 1 + dx_1^2y_1^2$
and $x_2^2 + y_2^2 = 1 + dx_2^2y_2^2$
and $dx_1x_2y_1y_2 = \pm 1$
then $dx_1^2y_1^2(x_2 + y_2)^2$
$= dx_1^2y_1^2(x_2^2 + y_2^2 + 2x_2y_2)$
$= dx_1^2y_1^2(dx_2^2y_2^2 + 1 + 2x_2y_2)$
$= d^2x_1^2y_1^2x_2^2y_2^2 + dx_1^2y_1^2 + 2dx_1^2y_1^2x_2y_2$
$= 1 + dx_1^2y_1^2 \pm 2x_1y_1$
$= x_1^2 + y_1^2 \pm 2x_1y_1$

Denominators are never 0.

But need different proof;

"$x^2 + y^2 > 0$" doesn't work.

If $x_1^2 + y_1^2 = 1 + dx_1^2 y_1^2$
and $x_2^2 + y_2^2 = 1 + dx_2^2 y_2^2$
and $dx_1 x_2 y_1 y_2 = \pm 1$
then $dx_1^2 y_1^2 (x_2 + y_2)^2$
$= dx_1^2 y_1^2 (x_2^2 + y_2^2 + 2x_2 y_2)$
$= dx_1^2 y_1^2 (dx_2^2 y_2^2 + 1 + 2x_2 y_2)$
$= d^2 x_1^2 y_1^2 x_2^2 y_2^2 + dx_1^2 y_1^2 + 2dx_1^2 y_1^2 x_2 y_2$
$= 1 + dx_1^2 y_1^2 \pm 2x_1 y_1$
$= x_1^2 + y_1^2 \pm 2x_1 y_1$
$= (x_1 \pm y_1)^2$.

Case 1: $x_2 + y_2 \neq 0$. Then

$$d = \left(\frac{x_1 \pm y_1}{x_1 y_1 (x_2 + y_2)} \right)^2,$$

contradiction.

Case 1: $x_2 + y_2 \neq 0$. Then

$$d = \left(\frac{x_1 \pm y_1}{x_1 y_1 (x_2 + y_2)} \right)^2,$$

contradiction.

Case 2: $x_2 - y_2 \neq 0$. Then

$$d = \left(\frac{x_1 \mp y_1}{x_1 y_1 (x_2 - y_2)} \right)^2,$$

contradiction.

Case 1: $x_2 + y_2 \neq 0$. Then

$$d = \left(\frac{x_1 \pm y_1}{x_1 y_1 (x_2 + y_2)} \right)^2,$$

contradiction.

Case 2: $x_2 - y_2 \neq 0$. Then

$$d = \left(\frac{x_1 \mp y_1}{x_1 y_1 (x_2 - y_2)} \right)^2,$$

contradiction.

Case 3: $x_2 + y_2 = x_2 - y_2 = 0$. Then $x_2 = 0$ and $y_2 = 0$, contradiction.

Using ECC sensibly

Typical starting point:
Client knows secret key a
and server's public key $b(X,Y)$.
Client computes (and caches)
shared secret $ab(X,Y)$.

Client has packet for server.
Generates unique nonce.
Uses shared secret to encrypt
and authenticate packet.

Total packet overhead:
24 bytes for nonce,
16 bytes for authenticator,
32 bytes for client's public key.

Server receives packet,
sees client's public key $a(X, Y)$.
Server computes (and caches)
shared secret $ab(X, Y)$.

Server uses shared secret
to verify authenticator
and decrypt packet.

Client and server encrypt,
authenticate, verify, and decrypt
all subsequent packets
in the same way,
using the same shared secret.

Easy-to-use packet protection:
`crypto_box` from
`nacl.cace-project.eu`.

High-security curve (Curve25519).
High-security implementation
(e.g., no secret array indices).
Extensive code validation.

Server can compute shared secrets
for 1000000 new clients
in 40 seconds of computation
on a Core 2 Quad.

Not much hope for attacker
if ECC user is running this!