Public-key cryptography
and the Discrete-Logarithm
Problem
Tanja Lange
Technische Universiteit Eindhoven

with some slides by
Daniel J. Bernstein
Cryptography

Let’s understand what our browsers do.
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 $q$, to speed up decryption (CRT).
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 
\[
c^d \equiv (m^e)^d \equiv m^{k\phi(N)+1} \equiv m \pmod{N}.
\]
Encryption of message $m < N$:

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

Decryption of ciphertext $c$:

Compute

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

Signature on message $m$:

Uses cryptographic hash function $h : \{0, 1\}^* \rightarrow \mathbb{Z}/N$

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, if A sends same message to B, C, D, who all use $e = 3$.

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 $c' = (m')^e \pmod{N}$. 
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 $n = \lceil \log_2 N \rceil$. 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 $$X = m00 \ldots 0 \oplus G(r).$$
4. Compute $Y = r \oplus H(X)$. 
5. Output $X, Y$.
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” ≠ “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:

\( (0, 1) = \text{“12:00”} \).

\( (0, -1) = \text{“6:00”} \).

\( (1, 0) = \text{“3:00”} \).

\( (-1, 0) = \text{“9:00”} \).
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:

$(0, 1) = \text{"12:00"}$.

$(0, -1) = \text{"6:00"}$.

$(1, 0) = \text{"3:00"}$.

$(-1, 0) = \text{"9:00"}$.

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

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

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

$(0, 1) = \text{“12:00”}$.
$(0, -1) = \text{“6:00”}$.
$(1, 0) = \text{“3:00”}$.
$(-1, 0) = \text{“9:00”}$.
$(\sqrt{3}/4, 1/2) = \text{“2:00”}$.
$(1/2, -\sqrt{3}/4) = \text{“5:00”}$.
$(-1/2, -\sqrt{3}/4) = \text{“7:00”}$.
$(\sqrt{1}/2, \sqrt{1}/2) = \text{“1:30”}$.
Many more.
Addition on the clock:

\[ x^2 + y^2 = 1, \text{ parametrized by } x = \sin \alpha, \; y = \cos \alpha. \]
Addition on the clock:

$x^2 + y^2 = 1$, parametrized by $x = \sin \alpha, \quad y = \cos \alpha$. Recall $(\sin(\alpha_1 + \alpha_2), \cos(\alpha_1 + \alpha_2)) = \ldots$
Addition on the clock:

\[ \text{neutral} = (0, 1) \]

\[ P_1 = (x_1, y_1) \]

\[ P_2 = (x_2, y_2) \]

\[ P_3 = (x_3, y_3) \]

\[ x^2 + y^2 = 1, \text{ parametrized by} \]

\[ x = \sin \alpha, \quad y = \cos \alpha. \]

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:

\[ x^2 + y^2 = 1, \text{ parametrized by } x = \sin \alpha, \quad y = \cos \alpha. \]

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 $\alpha_1$ and $\alpha_2$. Angles modulo $360^\circ$ 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 $\alpha$ is point with $-\alpha$ since $\alpha + (-\alpha) = 0$.

There are many more points where angle $\alpha$ is not “nice.”
Clock addition without \(\sin, \cos:\)

\[
\text{neutral} = (0, 1)
\]

\[
P_1 = (x_1, y_1)
\]

\[
P_2 = (x_2, y_2)
\]

\[
P_3 = (x_3, y_3)
\]

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

\[
\text{sum } (x_1, y_1) + (x_2, y_2) = (x_3, y_3)
\]

\[
= (x_1y_2 + y_1x_2, y_1y_2 - x_1x_2)
\]

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

\[
kP = P + P + \cdots + P \text{ for } k \geq 0.
\]

\(k\) copies
Examples of clock addition:

“2:00” + “5:00”

= \(\left(\sqrt{3/4}, \frac{1}{2}\right) + (1/2, -\sqrt{3/4})\)

= \((-1/2, -\sqrt{3/4}) = “7:00”\).

“5:00” + “9:00”

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

= \((\sqrt{3/4}, 1/2) = “2:00”\).

\[2 \begin{pmatrix} 3 & 4 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 24 & 7 \\ 25 & 25 \end{pmatrix}.\]
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”
= \((1/2, -\sqrt{3}/4) + (-1, 0)\)
= \((\sqrt{3}/4, 1/2) = “2:00”\).

\[
2 \begin{pmatrix}
3 & 4 \\
5 & 5 \\
\end{pmatrix} = \begin{pmatrix}
24 & 7 \\
25 & 25 \\
\end{pmatrix}.
\]

\[
3 \begin{pmatrix}
3 & 4 \\
5 & 5 \\
\end{pmatrix} = \begin{pmatrix}
117 & -44 \\
125 & 125 \\
\end{pmatrix}.
\]
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”
= \((1/2, -\sqrt{3}/4) + (-1, 0)\)
= \((\sqrt{3}/4, 1/2) = “2:00”\).

\[
2 \begin{pmatrix} 3 & 4 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 24 & 7 \\ 25 & 25 \end{pmatrix}.
\]

\[
3 \begin{pmatrix} 3 & 4 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 117 & -44 \\ 125 & 125 \end{pmatrix}.
\]

\[
4 \begin{pmatrix} 3 & 4 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 336 & -527 \\ 625 & 625 \end{pmatrix}.
\]
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”

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

= \((\sqrt{3}/4, 1/2) = “2:00”\).

\[
2 \begin{pmatrix} 3 & 4 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 24 & 7 \\ 25 & 25 \end{pmatrix}.
\]

\[
3 \begin{pmatrix} 3 & 4 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 117 & -44 \\ 125 & 125 \end{pmatrix}.
\]

\[
4 \begin{pmatrix} 3 & 4 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 336 & -527 \\ 625 & 625 \end{pmatrix}.
\]

\((x_1, y_1) + (0, 1) = \)
Examples of clock addition:

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

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

\[2 \begin{pmatrix} 3 & 4 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 24 & 7 \\ 25 & 25 \end{pmatrix}.\]

\[3 \begin{pmatrix} 3 & 4 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 117 & -44 \\ 125 & 125 \end{pmatrix}.\]

\[4 \begin{pmatrix} 3 & 4 \\ 5 & 5 \end{pmatrix} = \begin{pmatrix} 336 & -527 \\ 625 & 625 \end{pmatrix}.\]

\[(x_1, y_1) + (0, 1) = (x_1, y_1).\]
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".

\[
2 \begin{pmatrix} \frac{3}{5} & \frac{4}{5} \\ \end{pmatrix} = \begin{pmatrix} 24 & 7 \\ 25' & 25 \\ \end{pmatrix}.
\]

\[
3 \begin{pmatrix} \frac{3}{5} & \frac{4}{5} \\ \end{pmatrix} = \begin{pmatrix} 117 & -44 \\ 125' & 125 \\ \end{pmatrix}.
\]

\[
4 \begin{pmatrix} \frac{3}{5} & \frac{4}{5} \\ \end{pmatrix} = \begin{pmatrix} 336 & -527 \\ 625' & 625 \\ \end{pmatrix}.
\]

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

\((x_1, y_1) + (-x_1, y_1) =\)
Examples of clock addition:

“2:00” + “5:00”

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

= \( (-1/2, -\sqrt{3}/4) = “7:00”. \)

“5:00” + “9:00”

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

= \( (\sqrt{3}/4, 1/2) = “2:00”. \)

\[
2 \begin{pmatrix} 3 \\ 5' \\ 5 \\
\end{pmatrix} = \begin{pmatrix} 24 \\ 25' \\ 25 \end{pmatrix}.
\]

\[
3 \begin{pmatrix} 3 \\ 5' \\ 5 \\
\end{pmatrix} = \begin{pmatrix} 117 \\ 125' \\ 125 \end{pmatrix}.
\]

\[
4 \begin{pmatrix} 3 \\ 5' \\ 5 \\
\end{pmatrix} = \begin{pmatrix} 336 \\ 625' \\ 625 \end{pmatrix}.
\]

\[
(x_1, y_1) + (0, 1) = (x_1, y_1).
\]

\[
(x_1, y_1) + (-x_1, y_1) = (0, 1).
\]
Clocks over finite fields

Clock\((F_7) = \{(x, y) \in F_7 \times F_7 : x^2 + y^2 = 1\}\). Here \(F_7 = \{0, 1, 2, 3, 4, 5, 6\}\) with \(+, -, \times\) modulo 7.
Larger example: $\text{Clock}(F_{1000003})$.

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 $(X, Y) \in \text{Clock}(\mathbb{F}_p)$.

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 $a$

$\downarrow$

Alice’s public key $a(X, Y)$

\{Alice, Bob\}’s shared secret $ab(X, Y)$

$\rightarrow$

Bob’s secret key $b$

$\downarrow$

Bob’s public key $b(X, Y)$

\{Bob, Alice\}’s shared secret $ba(X, Y)$
Alice’s secret key $a$

↓

Alice’s public key $a(X, Y)$

↓

{Alice, Bob}’s shared secret $ab(X, Y)$

Bob’s secret key $b$

↓

Bob’s public key $b(X, Y)$

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
\[
\left(\frac{x_1y_2 + y_1x_2}{1 - 30x_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 + 30x_1x_2y_1y_2}\right).
\]
The clock again, for comparison:

\[ x^2 + y^2 = 1. \]

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

\[
(x_1y_2 + y_1x_2, \quad y_1y_2 - x_1x_2).
\]
“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 \( \sqrt{30} \vert xy \vert < 1 \).

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 \( \sqrt{30} \vert x_1y_1 \vert < 1 \) and \( \sqrt{30} \vert x_2y_2 \vert < 1 \) so \( 30 \vert x_1y_1x_2y_2 \vert < 1 \) so \( 1 \pm 30x_1x_2y_1y_2 > 0 \).
The Edwards addition law
\[(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + y_1x_2}{1 - 30x_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 + 30x_1x_2y_1y_2}\right)\]
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 $d \in \mathbb{F}_q$.

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

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

\[
x_3 = \frac{x_1 y_2 + y_1 x_2}{1 + dx_1 x_2 y_1 y_2},
\]
\[
y_3 = \frac{y_1 y_2 - x_1 x_2}{1 - dx_1 x_2 y_1 y_2}.
\]
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 \(x_1^2 + y_1^2 = 1 + dx_1^2y_1^2\)
and \(x_2^2 + y_2^2 = 1 + dx_2^2y_2^2\)
and \(dx_1x_2y_1y_2 = \pm 1\)
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 $x_1^2 + y_1^2 = 1 + dx_1^2y_1^2$
and $x_2^2 + y_2^2 = 1 + dx_2^2y_2^2$
and $dx_1x_2y_1y_2 = \pm 1$
then $dx_1^2y_1^2(x_2 + y_2)^2$

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

$= dx_1^2 y_1^2 (x_2^2 + y_2^2 + 2x_2 y_2)$

$= dx_1^2 y_1^2 (dx_2^2 y_2^2 + 1 + 2x_2 y_2)$

$= d^2 x_1^2 y_1^2 x_2 y_2^2 + dx_1^2 y_1^2 + 2dx_1^2 y_1^2 x_2 y_2$

$= 1 + dx_1^2 y_1^2 \pm 2x_1 y_1$

$= x_1^2 + y_1^2 \pm 2x_1 y_1$
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 $dx_1^2y_1^2(x_2 + y_2)^2$
$= dx_1^2y_1^2(x_2^2 + y_2^2 + 2x_2y_2)$
$= dx_1^2y_1^2(dx_2^2y_2^2 + 1 + 2x_2y_2)$
$= d^2x_1^2y_1^2x_2^2y_2^2 + dx_1^2y_1^2 + 2dx_1^2y_1^2x_2y_2$
$= 1 + dx_1^2y_1^2 \pm 2x_1y_1$
$= x_1^2 + y_1^2 \pm 2x_1y_1$
$= (x_1 \pm y_1)^2$. 
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!
Eliminating divisions

Typical computation:
\( P \mapsto nP \).

Decompose into additions:
\( P, Q \mapsto P + Q \).

Addition \((x_1, y_1) + (x_2, y_2) = (x_1y_2 + y_1x_2)/(1 + dx_1x_2y_1y_2), (y_1y_2 - x_1x_2)/(1 - dx_1x_2y_1y_2))\) uses expensive divisions.

Better: postpone divisions and work with fractions.

Represent \((x, y)\) as \((X : Y : Z)\) with \(x = X/Z\) and \(y = Y/Z\) for \(Z \neq 0\).
Addition now has to handle fractions as input:

\[
\left( \frac{X_1}{Z_1}, \frac{Y_1}{Z_1} \right) + \left( \frac{X_2}{Z_2}, \frac{Y_2}{Z_2} \right) =
\]

\[
\left( \frac{X_1}{Z_1} \frac{Y_2}{Z_2} + \frac{Y_1}{Z_1} \frac{X_2}{Z_2}, \frac{Y_1}{Z_1} \frac{X_2}{Z_2} + X_2 \frac{Y_2}{Z_2} \right),
\]

\[
1 + d \frac{X_1}{Z_1} \frac{X_2}{Z_2} \frac{Y_1}{Z_1} \frac{Y_2}{Z_2},
\]

\[
1 - d \frac{X_1}{Z_1} \frac{X_2}{Z_2} \frac{Y_1}{Z_1} \frac{Y_2}{Z_2}
\]

\[
= \left( \frac{Z_1 Z_2 \left( X_1 Y_2 + Y_1 X_2 \right)}{Z_1^2 Z_2^2 + dX_1 X_2 Y_1 Y_2}, \frac{Z_1 Z_2 \left( Y_1 Y_2 - X_1 X_2 \right)}{Z_1^2 Z_2^2 - dX_1 X_2 Y_1 Y_2} \right).
\]
i.e. \[
\left( \frac{X_1}{Z_1}, \frac{Y_1}{Z_1} \right) + \left( \frac{X_2}{Z_2}, \frac{Y_2}{Z_2} \right)
\]
\[
= \left( \frac{X_3}{Z_3}, \frac{Y_3}{Z_3} \right)
\]
where
\[
F = Z_1^2 Z_2^2 - dX_1 X_2 Y_1 Y_2,
\]
\[
G = Z_1^2 Z_2^2 + dX_1 X_2 Y_1 Y_2,
\]
\[
X_3 = Z_1 Z_2 (X_1 Y_2 + Y_1 X_2) F,
\]
\[
Y_3 = Z_1 Z_2 (Y_1 Y_2 - X_1 X_2) G,
\]
\[
Z_3 = FG.
\]

Input to addition algorithm: \(X_1, Y_1, Z_1, X_2, Y_2, Z_2\).

Output from addition algorithm: \(X_3, Y_3, Z_3\). No divisions needed!
Save multiplications by eliminating common subexpressions:

\[ A = Z_1 \cdot Z_2; \quad B = A^2; \]
\[ C = X_1 \cdot X_2; \]
\[ D = Y_1 \cdot Y_2; \]
\[ E = d \cdot C \cdot D; \]
\[ F = B - E; \quad G = B + E; \]
\[ X_3 = A \cdot F \cdot (X_1 \cdot Y_2 + Y_1 \cdot X_2); \]
\[ Y_3 = A \cdot G \cdot (D - C); \]
\[ Z_3 = F \cdot G. \]

Cost: \(11\text{M} + 1\text{S} + 1\text{D}\).

Can do better: \(10\text{M} + 1\text{S} + 1\text{D}\).
Faster doubling

\[(x_1, y_1) + (x_1, y_1) = ((x_1y_1 + y_1x_1)/(1 + dx_1x_1y_1), (y_1y_1 - x_1x_1)/(1 - dx_1x_1y_1)) = ((2x_1y_1)/(1 + dx_1^2y_1^2), (y_1^2 - x_1^2)/(1 - dx_1^2y_1^2)).\]

\[x_1^2 + y_1^2 = 1 + dx_1^2y_1^2 \text{ so} \]
\[(x_1, y_1) + (x_1, y_1) = ((2x_1y_1)/(x_1^2 + y_1^2), (y_1^2 - x_1^2)/(2 - x_1^2 - y_1^2)).\]

Again eliminate divisions using \(P^2\): only 3M + 4S.

Much faster than addition.
Useful: many doublings in ECC.
More addition strategies

Dual addition formula:

\[(x_1, y_1) + (x_2, y_2) = \left( \frac{x_1 y_1 + x_2 y_2}{x_1 x_2 + y_1 y_2}, \frac{x_1 y_1 - x_2 y_2}{x_1 y_2 - x_2 y_1} \right)\]  

Low degree, no need for \(d\).

Warning: fails for doubling!

Is this really “addition”?

Most EC formulas have failures.
More addition strategies

Dual addition formula:
\[(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_1 + x_2y_2}{x_1x_2 + y_1y_2}, \frac{x_1y_1 - x_2y_2}{x_1y_2 - x_2y_1}\right)\).

Low degree, no need for \(d\).

Warning: fails for doubling!

Is this really “addition”? Most EC formulas have failures.

More coordinate systems:
Inverted: \(x = \frac{Z}{X}, y = \frac{Z}{Y}\).
Extended: \(x = \frac{X}{Z}, y = \frac{Y}{T}\).
Completed: \(x = \frac{X}{Z}, y = \frac{Y}{Z}, xy = \frac{T}{Z}\).
More elliptic curves

Edwards curves are elliptic. Easiest way to understand elliptic curves is Edwards.

Geometrically, all elliptic curves are Edwards curves.

Algebraically, more elliptic curves exist. Every odd-char curve can be expressed as Weierstrass curve

\[ v^2 = u^3 + a_2 u^2 + a_4 u + a_6. \]

Warning: “Weierstrass” has different meaning in char 2.
Addition on Weierstrass curve

\[ v^2 = u^3 + u^2 + u + 1 \]

Slope \( \lambda = (v_2 - v_1)/(u_2 - u_1) \).
Note that \( u_1 \neq u_2 \).
Doubling on Weierstrass curve

\[ v^2 = u^3 - u \]

Slope \( \lambda = \frac{3u_1^2 - 1}{2v_1} \).
In most cases
\[(u_1, v_1) + (u_2, v_2) = (u_3, v_3)\]
where \((u_3, v_3) = (\lambda^2 - u_1 - u_2, \lambda(u_1 - u_3) - v_1)\).

\(u_1 \neq u_2\), “addition” (alert!):
\[
\lambda = \frac{v_2 - v_1}{u_2 - u_1}.
\]
Total cost \(1I + 2M + 1S\).

\((u_1, v_1) = (u_2, v_2)\) and \(v_1 \neq 0\),
“doubling” (alert!):
\[
\lambda = \frac{(3u_1^2 + 2a_2u_1 + a_4)}{2v_1}.
\]
Total cost \(1I + 2M + 2S\).

Also handle some exceptions:
\((u_1, v_1) = (u_2, -v_2)\);
inputs at \(\infty\).
Birational equivalence

Starting from point \((x, y)\)
on \(x^2 + y^2 = 1 + dx^2 y^2:\)

Define \(A = \frac{2(1 + d)}{(1 - d)}\), \(B = \frac{4}{(1 - d)}\);
\(u = \frac{1 + y}{(B(1 - y))}\), \(v = \frac{u}{x} = \frac{(1 + y)}{(Bx(1 - y))}\).
(Skip a few exceptional points.)

\(v^2 = u^3 + \frac{A}{B}u^2 + \frac{1}{B^2}u.\)

Maps Edwards to Weierstrass.
Compatible with point addition!

Easily invert this map:
\(x = \frac{u}{v}, \ y = \frac{(Bu - 1)}{(Bu + 1)}.\)
Some history

There are many perspectives on elliptic-curve computations.


1984 (published 1985) Miller, and independently

Bosma, Goldwasser–Kilian, Chudnovsky–Chudnovsky, Atkin: elliptic-curve primality proving.
The Edwards perspective is new!

1761 Euler, 1866 Gauss introduced an addition law
for \( x^2 + y^2 = 1 - x^2 y^2 \),
the “lemniscatic elliptic curve.”

2007 Edwards generalized to
many curves \( x^2 + y^2 = 1 + c^4 x^2 y^2 \).
Theorem: have now obtained
all elliptic curves over \( \mathbb{Q} \).

2007 Bernstein–Lange:
Edwards addition law is complete
for \( x^2 + y^2 = 1 + dx^2 y^2 \) if \( d \neq 0 \);
and gives new ECC speed records.
\[ y^2 = x^3 - 0.4x + 0.7 \]
The Weierstrass-turtle: old, trusted and slow. Warning: (picture) incomplete!
\[ x^2 + y^2 = 1 - 300x^2y^2 \]
The Edwards starfish: new, fast and complete!
Start!
Weierstrass sets off, Edwards left behind sleeping
Weierstrass has made some progress – finally Edwards wakes up.
Exciting progress: Edwards about to overtake!!
And the winner is: Edwards!