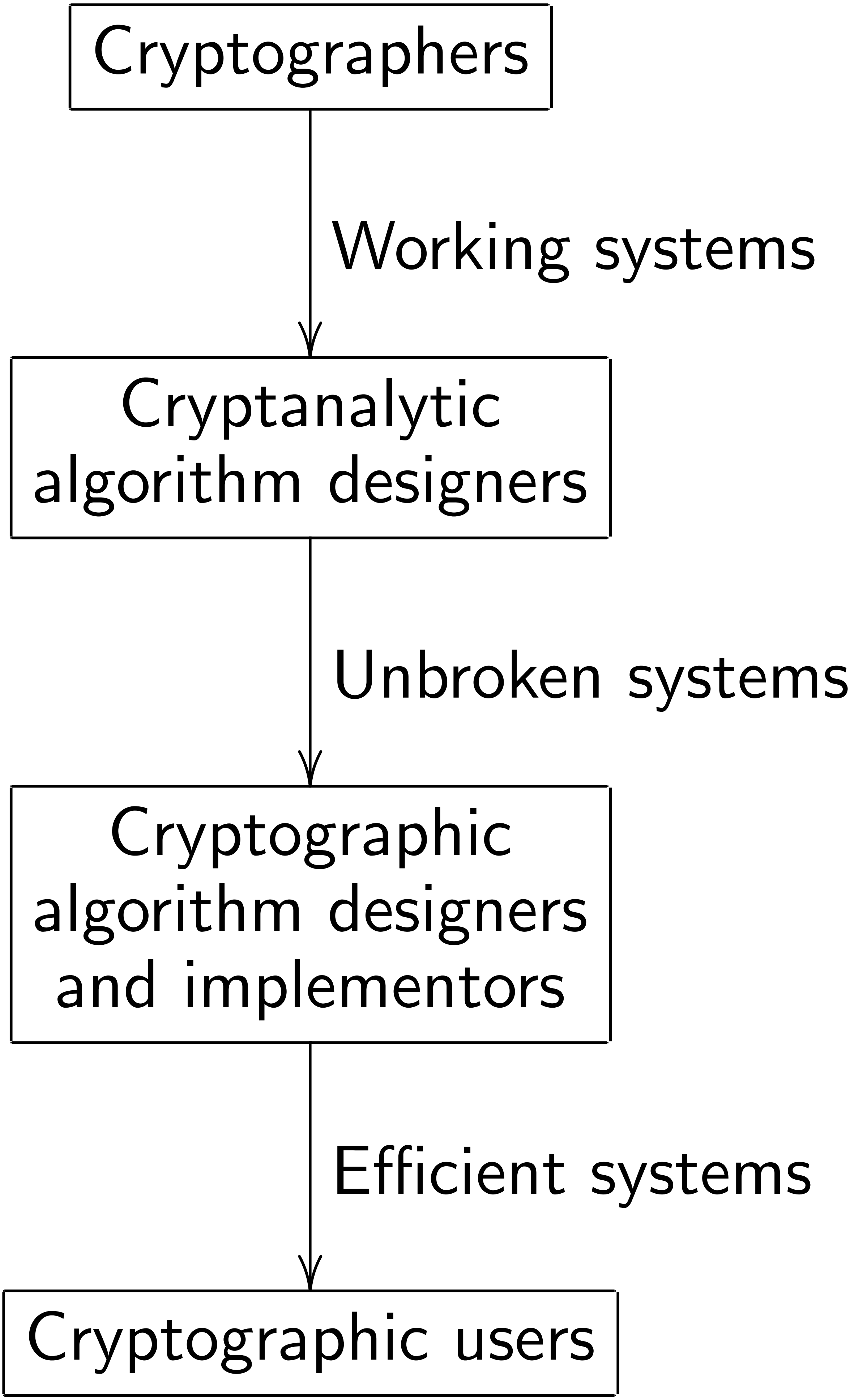
High-speed cryptography, part 3: more cryptosystems

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


Cryptographers
Working systems
Cryptanalytic algorithm designers
Unbroken systems
Cryptographic algorithm designers and implementors
Efficient systems
Cryptographic users

1. Working systems

Fundamental question for cryptographers:
How can we encrypt, decrypt, sign, verify, etc.?

Many answers:
DES, Triple DES, FEAL-4, AES, RSA, McEliece encryption, Merkle hash-tree signatures, Merkle–Hellman knapsack encryption, Buchmann–Williams class-group encryption, ECDSA, HFE^{v-}, NTRU, et al.

2. Unbroken systems

Fundamental question for
pre-quantum cryptanalysts:
What can an attacker do
using $<2^b$ operations
on a *classical* computer?

Fundamental question for
post-quantum cryptanalysts:
What can an attacker do
using $<2^b$ operations
on a *quantum* computer?

Goal: identify systems that are
not breakable in $<2^b$ operations.

Examples of RSA cryptanalysis:

Schroeppel's "linear sieve", mentioned in 1978 RSA paper, factors pq into p, q using $(2+o(1))^{(\lg pq)^{1/2}(\lg\lg pq)^{1/2}}$ simple operations (conjecturally).

To push this beyond 2^b, must choose pq to have at least $(0.5+o(1))b^2/\lg b$ bits.

Note 1: $\lg = \log_2$.

Note 2: $o(1)$ says *nothing* about, e.g., $b = 128$.
Today: focus on asymptotics.

1993 Buhler–Lenstra–Pomerance, generalizing 1988 Pollard "number-field sieve", factors pq into p, q using $(3.79\ldots + o(1))^{(\lg pq)^{1/3}(\lg \lg pq)^{2/3}}$ simple operations (conjecturally).

To push this beyond 2^b, must choose pq to have at least $(0.015\ldots + o(1))b^3/(\lg b)^2$ bits.

Subsequent improvements: $3.73\ldots$; details of $o(1)$.
But can reasonably conjecture that $2^{(\lg pq)^{1/3+o(1)}}$ is optimal —for classical computers.

Cryptographic systems surviving *pre-quantum* cryptanalysis:

Triple DES (for $b \le 112$), AES-256 (for $b \le 256$), RSA with $b^{3+o(1)}$-bit modulus, McEliece with code length $b^{1+o(1)}$, Merkle signatures with "strong" $b^{1+o(1)}$-bit hash, BW with "strong" $b^{2+o(1)}$-bit discriminant, ECDSA with "strong" $b^{1+o(1)}$-bit curve, HFE$^{\mathsf{v}-}$ with $b^{1+o(1)}$ polynomials, NTRU with $b^{1+o(1)}$ bits, et al.

Typical algorithmic tools for *pre-quantum* cryptanalysts: NFS, ρ, ISD, LLL, F4, XL, et al.

Post-quantum cryptanalysts have all the same tools *plus* quantum algorithms.

Spectacular example:
1994 Shor factors pq into p, q using $(\lg pq)^{2+o(1)}$ simple quantum operations.
To push this beyond 2^b, must choose pq to have at least $2^{(0.5+o(1))b}$ bits. Yikes.

Cryptographic systems surviving *post-quantum* cryptanalysis:

AES-256 (for $b \le 128$),
McEliece code-based encryption with code length $b^{1+o(1)}$,
Merkle hash-based signatures with "strong" $b^{1+o(1)}$-bit hash,
$\mathrm{HFE}^{\mathrm{v}-}$ MQ signatures with $b^{1+o(1)}$ polynomials,
NTRU lattice-based encryption with $b^{1+o(1)}$ bits,
et al.

3. Efficient systems

Fundamental question for designers and implementors of cryptographic algorithms: Exactly how efficient are the unbroken cryptosystems?

Many goals: minimize encryption time, size, decryption time, etc.

Pre-quantum example: RSA encrypts and verifies in $b^{3+o(1)}$ simple operations. Signature occupies $b^{3+o(1)}$ bits.

ECC (with strong curve/$\mathbf{F}_q$, reasonable padding, etc.):

ECDL costs $2^{(1/2+o(1))\lg q}$ by Pollard's rho method. Conjecture: this is the optimal attack against ECC.

Can take $\lg q \in (2+o(1))b$.

Encryption: Fast scalar mult costs $(\lg q)^{2+o(1)} = b^{2+o(1)}$.

Summary: ECC costs $b^{2+o(1)}$. Asymptotically faster than RSA. Bonus: also $b^{2+o(1)}$ *decryption*.

Efficiency is important:
users have cost constraints.

Cryptographers, cryptanalysts,
implementors, etc. tend to
focus on RSA and ECC,
citing these cost constraints.

But Shor breaks RSA and ECC!

Efficiency is important:
users have cost constraints.

Cryptographers, cryptanalysts, implementors, etc. tend to focus on RSA and ECC, citing these cost constraints.

But Shor breaks RSA and ECC!

We think that the most efficient unbroken *post-quantum* systems will be hash-based signatures, code-based encryption, lattice-based encryption, multivariate-quadratic sigs.

1978 McEliece system (with length-n classical Goppa codes, reasonable padding, etc.):

Conjecture: Fastest attacks cost $2^{(\beta+o(1))n/\lg n}$.
Quantum attacks: smaller β.

Can take $n \in (1/\beta + o(1))b \lg b$.

Encryption: Matrix mult costs $n^{2+o(1)} = b^{2+o(1)}$.

Summary: McEliece costs $b^{2+o(1)}$.

Hmmm: is this *faster* than ECC? Need more detailed analysis.

ECC encryption:
$\Theta(\lg q)$ operations in $\mathbf{F}_q$.
Each operation in $\mathbf{F}_q$ costs
$\Theta(\lg q \lg \lg q \lg \lg \lg q)$.
Total $\Theta(b^2 \lg b \lg \lg b)$.

ECC encryption:
$\Theta(\lg q)$ operations in $\mathbf{F}_q$.
Each operation in $\mathbf{F}_q$ costs
$\Theta(\lg q \lg\lg q \lg\lg\lg q)$.
Total $\Theta(b^2 \lg b \lg\lg b)$.

McEliece encryption,
with 1986 Niederreiter speedup:
$\Theta(n/\lg n)$ additions in $\mathbf{F}_2^n$,
each costing $\Theta(n)$.
Total $\Theta(b^2 \lg b)$.

ECC encryption:
$\Theta(\lg q)$ operations in $\mathbf{F}_q$.
Each operation in $\mathbf{F}_q$ costs
$\Theta(\lg q \lg \lg q \lg \lg \lg q)$.
Total $\Theta(b^2 \lg b \lg \lg b)$.

McEliece encryption,
with 1986 Niederreiter speedup:
$\Theta(n/\lg n)$ additions in $\mathbf{F}_2^n$,
each costing $\Theta(n)$.
Total $\Theta(b^2 \lg b)$.

McEliece is asymptotically faster.
Bonus: Even faster decryption.
Another bonus: Post-quantum.

Algorithmic advances can change the competition. Examples:

1. Speed up ECC: can reduce $\lg \lg b$ using 2007 Fürer; maybe someday eliminate $\lg \lg b$?

Algorithmic advances can change the competition. Examples:

1. Speed up ECC: can reduce $\lg\lg b$ using 2007 Fürer; maybe someday eliminate $\lg\lg b$?

2. Faster attacks on McEliece: 2010 Bernstein–Lange–Peters, 2011 May–Meurer–Thomae, 2012 Becker–Joux–May–Meurer. … but still $\Theta(b^2 \lg b)$.

Algorithmic advances can change the competition. Examples:

1. Speed up ECC: can reduce $\lg\lg b$ using 2007 Fürer; maybe someday eliminate $\lg\lg b$?

2. Faster attacks on McEliece: 2010 Bernstein–Lange–Peters, 2011 May–Meurer–Thomae, 2012 Becker–Joux–May–Meurer. . . . but still $\Theta(b^2 \lg b)$.

3. We're optimizing "subfield AG" variant of McEliece.
Conjecture: Fastest attacks cost $2^{(\alpha+o(1))n}$; encryption $\Theta(b^2)$.

Code-based encryption

Modern version of McEliece:

Receiver's public key is "random" $t \lg n \times n$ matrix K over $\mathbf{F}_2$. Specifies linear $\mathbf{F}_2^n \to \mathbf{F}_2^{t \lg n}$.

Typically $t \lg n \approx 0.2n$; e.g., $n = 2048$, $t = 40$.

Messages suitable for encryption: $\left\{m \in \mathbf{F}_2^n : \#\{i : m_i = 1\} = t\right\}$.

Encryption of m is $Km \in \mathbf{F}_2^{t \lg n}$.

Use hash of m as secret AES-GCM key to encrypt more data.

Attacker, by linear algebra, easily works backwards from Km to *some* $v \in \mathbf{F}_2^n$ such that $Kv = Km$.

i.e. Attacker finds *some* element $v \in m + \text{Ker}K$. Note that $\#\text{Ker}K \geq 2^{n-t\lg n}$.

Attacker wants to decode v: to find element of $\text{Ker}K$ at distance only t from v. Presumably unique, revealing m. But decoding isn't easy!

Receiver builds K with *secret* Goppa structure for fast decoding.

Goppa codes

Fix $q \in \{8, 16, 32, \ldots\}$;
$t \in \{2, 3, \ldots, \lfloor (q-1)/\lg q \rfloor\}$;
$n \in \{t \lg q + 1, t \lg q + 2, \ldots, q\}$.
e.g. $q = 1024$, $t = 50$, $n = 1024$.
or $q = 4096$, $t = 150$, $n = 3600$.

Receiver builds a matrix H as the parity-check matrix for the classical (genus-0) irreducible length-n degree-t binary Goppa code defined by a monic degree-t irreducible polynomial $g \in \mathbf{F}_q[x]$ and distinct $a_1, a_2, \ldots, a_n \in \mathbf{F}_q$.

. . . which means: $H =$

$$\begin{pmatrix} \dfrac{1}{g(a_1)} & \cdots & \dfrac{1}{g(a_n)} \\ \dfrac{a_1}{g(a_1)} & \cdots & \dfrac{a_n}{g(a_n)} \\ \vdots & \ddots & \vdots \\ \dfrac{a_1^{t-1}}{g(a_1)} & \cdots & \dfrac{a_n^{t-1}}{g(a_n)} \end{pmatrix}.$$

View each element of $\mathbf{F}_q$ here as a column in $\mathbf{F}_2^{\lg q}$.

Then $H : \mathbf{F}_2^n \to \mathbf{F}_2^{t \lg q}$.

More useful view: Consider the map $m \mapsto \sum_i m_i/(x - a_i)$ from $\mathbf{F}_2^n$ to $\mathbf{F}_q[x]/g$.

H is the matrix for this map where $\mathbf{F}_2^n$ has standard basis and $\mathbf{F}_q[x]/g$ has basis $\lfloor g/x \rfloor$, $\lfloor g/x^2 \rfloor$, $\ldots$, $\lfloor g/x^t \rfloor$.

One-line proof: In $\mathbf{F}_q[x]$ have

$$\frac{g - g(a_i)}{x - a_i} = \sum_{j \ge 0} a_i^j \left\lfloor g/x^{j+1} \right\rfloor.$$

Receiver generates key K as row reduction of H, revealing only $\mathrm{Ker} H$.

Lattice-based encryption

1998 Hoffstein–Pipher–Silverman NTRU (textbook version, without required padding):

Receiver's public key is "random" $h \in ((\mathbf{Z}/q)[x]/(x^p - 1))^*$.

Ciphertext: $m + rh$ given $m, r \in (\mathbf{Z}/q)[x]/(x^p - 1)$; all coefficients in $\{-1, 0, 1\}$; $\#\{i : r_i{=}{-1}\} = \#\{i : r_i{=}1\} = t$.

p: prime; e.g., $p = 613$.
q: power of 2 around $8p$, with order $\geq (p-1)/2$ in $(\mathbf{Z}/p)^*$.
t: roughly $0.1p$.

Receiver built $h = 3g/(1+3f)$
where $f, g \in (\mathbf{Z}/q)[x]/(x^p - 1)$,
all coeffs in $\{-1, 0, 1\}$,
$\#\{i : f_i{=}{-1}\} = \#\{i : f_i{=}1\} = t$,
$\#\{i : g_i{=}{-1}\} \approx \#\{i : g_i{=}1\} \approx \frac{p}{3}$,
both $1 + 3f$ and g invertible.

Given ciphertext $c = m + rh$,
receiver computes
$(1+3f)c = (1+3f)m + 3rg$
in $(\mathbf{Z}/q)[x]/(x^p - 1)$,
lifts to $\mathbf{Z}[x]/(x^p - 1)$ with
coeffs in $\{-q/2, \ldots, q/2 - 1\}$,
reduces modulo 3
to obtain m.

Basic attack tool:
Lift pairs (u, uh) to $\mathbf{Z}^{2p}$
to obtain a lattice.

Attacking key h:
$(1 + 3f, 3g)$ is a short vector
in this lattice.

Attacking ciphertext c:
$(0, c)$ is close to
lattice vector (r, rh).

Standard lattice algorithms
(SVP, CVP) cost $2^{\Theta(p)}$.
Nothing subexponential known,
even post-quantum.

Take $p \in \Theta(b)$ for security 2^b against all known attacks.

$\Theta(b \lg b)$ bits in key.

Time $b(\lg b)^{2+o(1)}$ to multiply in $(\mathbf{Z}/q)[x]/(x^p - 1)$.

Time $b(\lg b)^{2+o(1)}$ for encryption, decryption.

Excellent overall performance.

Take $p \in \Theta(b)$ for security 2^b against all known attacks.

$\Theta(b \lg b)$ bits in key.

Time $b(\lg b)^{2+o(1)}$ to multiply in $(\mathbf{Z}/q)[x]/(x^p - 1)$.

Time $b(\lg b)^{2+o(1)}$ for encryption, decryption.

Excellent overall performance.

The McEliece cryptosystem inspires more confidence but has much larger keys.

Something completely different

1985 H. Lange–Ruppert:
$A(\overline{k})$ has a complete system of addition laws, degree $\leq (3, 3)$.
Symmetry $\Rightarrow$ degree $\leq (2, 2)$.

"The proof is nonconstructive. . .
To determine explicitly a complete system of addition laws requires tedious computations already in the easiest case of an elliptic curve in Weierstrass normal form."

1985 Lange–Ruppert:
Explicit complete system
of 3 addition laws
for short Weierstrass curves.

Reduce formulas to 53 monomials
by introducing extra variables
$x_i y_j + x_j y_i$, $x_i y_j - x_j y_i$.

1987 Lange–Ruppert:
Explicit complete system
of 3 addition laws
for long Weierstrass curves.

$$\begin{aligned}
Y_3^{(2)} = {} & Y_1^2 Y_2^2 + a_1 X_2 Y_1^2 Y_2 + (a_1 a_2 - 3a_3) X_1 X_2^2 Y_1 \\
& + a_3 Y_1^2 Y_2 Z_2 - (a_2^2 - 3a_4) X_1^2 X_2^2 \\
& + (a_1 a_4 - a_2 a_3)(2X_1 Z_2 + X_2 Z_1) X_2 Y_1 \\
& + (a_1^2 a_4 - 2a_1 a_2 a_3 + 3a_3^2) X_1^2 X_2 Z_2 \\
& - (a_2 a_4 - 9a_6) X_1 X_2 (X_1 Z_2 + X_2 Z_1) \\
& + (3a_1 a_6 - a_3 a_4)(X_1 Z_2 + 2X_2 Z_1) Y_1 Z_2 \\
& + (3a_1^2 a_6 - 2a_1 a_3 a_4 + a_2 a_3^2 + 3a_2 a_6 - a_4^2) X_1 Z_2 (X_1 Z_2 + 2X_2 Z_1) \\
& - (3a_2 a_6 - a_4^2)(X_1 Z_2 + X_2 Z_1)(X_1 Z_2 - X_2 Z_1) \\
& + (a_1^3 a_6 - a_1^2 a_3 a_4 + a_1 a_2 a_3^2 - a_1 a_4^2 + 4a_1 a_2 a_6 - a_3^3 - 3a_3 a_6) Y_1 Z_1 Z_2^2 \\
& + (a_1^4 a_6 - a_1^3 a_3 a_4 + 5a_1^2 a_2 a_6 + a_1^2 a_2 a_3^2 - a_1 a_2 a_3 a_4 - a_1 a_3^3 - 3a_1 a_3 a_6 \\
& - a_1^2 a_4^2 + a_2^2 a_3^2 - a_2 a_4^2 + 4a_2^2 a_6 - a_3^2 a_4 - 3a_4 a_6) X_1 Z_1 Z_2^2 \\
& + (a_1^2 a_2 a_6 - a_1 a_2 a_3 a_4 + 3a_1 a_3 a_6 + a_2^2 a_3^2 - a_2 a_4^2 \\
& + 4a_2^2 a_6 - 2a_3^2 a_4 - 3a_4 a_6) X_2 Z_1^2 Z_2 \\
& + (a_1^3 a_3 a_6 - a_1^2 a_3^2 a_4 + a_1^2 a_4 a_6 + a_1 a_2 a_3^3 \\
& + 4a_1 a_2 a_3 a_6 - 2a_1 a_3 a_4^2 + a_2 a_3^2 a_4 \\
& + 4a_2 a_4 a_6 - a_3^4 - 6a_3^2 a_6 - a_4^3 - 9a_6^2) Z_1^2 Z_2^2, \\
Z_3^{(2)} = {} & 3X_1 X_2 (X_1 Y_2 + X_2 Y_1) + Y_1 Y_2 (Y_1 Z_2 + Y_2 Z_1) + 3a_1 X_1^2 X_2^2 \\
& + a_1 (2X_1 Y_2 + Y_1 X_2) Y_1 Z_2 + a_1^2 X_1 Z_2 (2X_2 Y_1 + X_1 Y_2) \\
& + a_2 X_1 X_2 (Y_1 Z_2 + Y_2 Z_1) \\
& + a_2 (X_1 Y_2 + X_2 Y_1)(X_1 Z_2 + X_2 Z_1) \\
& + a_1^3 X_1^2 X_2 Z_2 + a_1 a_2 X_1 X_2 (2X_1 Z_2 + X_2 Z_1) \\
& + 3a_3 X_1 X_2^2 Z_1 + a_3 Y_1 Z_2 (Y_1 Z_2 + 2Y_2 Z_1) \\
& + 2a_1 a_3 X_1 Z_2 (Y_1 Z_2 + Y_2 Z_1) \\
& + 2a_1 a_3 X_2 Y_1 Z_1 Z_2 + a_4 (X_1 Y_2 + X_2 Y_1) Z_1 Z_2 \\
& + a_4 (X_1 Z_2 + X_2 Z_1)(Y_1 Z_2 + Y_2 Z_1) \\
& + (a_1^2 a_3 + a_1 a_4) X_1 Z_2 (X_1 Z_2 + 2X_2 Z_1) + a_2 a_3 X_2 Z_1 (2X_1 Z_2 + X_2 Z_1) \\
& + a_3^2 Y_1 Z_1 Z_2^2 + (a_3^2 + 3a_6)(Y_1 Z_2 + Y_2 Z_1) Z_1 Z_2 \\
& + a_1 a_3^2 (2X_1 Z_2 + X_2 Z_1) Z_1 Z_2 + 3a_1 a_6 X_1 Z_1 Z_2^2 \\
& + a_3 a_4 (X_1 Z_2 + 2X_2 Z_1) Z_1 Z_2 + (a_3^3 + 3a_3 a_6) Z_1^2 Z_2^2.
\end{aligned}$$

1995 Bosma–Lenstra:
Explicit complete system
of 2 addition laws
for long Weierstrass curves:

$$X_3, Y_3, Z_3, X'_3, Y'_3, Z'_3$$
$$\in \mathbf{Z}[a_1, a_2, a_3, a_4, a_6,$$
$$X_1, Y_1, Z_1, X_2, Y_2, Z_2].$$

1995 Bosma–Lenstra:
Explicit complete system
of 2 addition laws
for long Weierstrass curves:

$$X_3, Y_3, Z_3, X_3', Y_3', Z_3'$$
$$\in \mathbf{Z}[a_1, a_2, a_3, a_4, a_6,$$
$$X_1, Y_1, Z_1, X_2, Y_2, Z_2].$$

My previous slide in this talk:
Bosma–Lenstra Y_3', Z_3'.

1995 Bosma–Lenstra:
Explicit complete system
of 2 addition laws
for long Weierstrass curves:
$X_3, Y_3, Z_3, X_3', Y_3', Z_3'$
$\in \mathbf{Z}[a_1, a_2, a_3, a_4, a_6,$
$\quad X_1, Y_1, Z_1, X_2, Y_2, Z_2]$.

My previous slide in this talk:
Bosma–Lenstra Y_3', Z_3'.
Actually, slide shows
$\text{Publish}(Y_3'), \text{Publish}(Z_3')$,
where Publish introduces typos.

What this means:

For all fields k,
all $\mathbf{P}^2$ Weierstrass curves
$E/k : Y^2Z + a_1XYZ + a_3YZ^2 = X^3 + a_2X^2Z + a_4XZ^2 + a_6Z^3$,
all $P_1 = (X_1 : Y_1 : Z_1) \in E(k)$,
all $P_2 = (X_2 : Y_2 : Z_2) \in E(k)$:

$(X_3 : Y_3 : Z_3)$
is $P_1 + P_2$ or $(0 : 0 : 0)$;
$(X_3' : Y_3' : Z_3')$
is $P_1 + P_2$ or $(0 : 0 : 0)$;
at most one of these is $(0 : 0 : 0)$.

2009 Bernstein–T. Lange:

For all fields k with $2 \neq 0$,
all $\mathbf{P}^1 \times \mathbf{P}^1$ Edwards curves E/k :
$X^2T^2 + Y^2Z^2 = Z^2T^2 + dX^2Y^2$,
all $P_1, P_2 \in E(k)$,
$P_1 = ((X_1 : Z_1), (Y_1 : T_1))$,
$P_2 = ((X_2 : Z_2), (Y_2 : T_2))$:

$(X_3 : Z_3)$ is $x(P_1 + P_2)$ or $(0 : 0)$;
$(X_3' : Z_3')$ is $x(P_1 + P_2)$ or $(0 : 0)$;
$(Y_3 : T_3)$ is $y(P_1 + P_2)$ or $(0 : 0)$;
$(Y_3' : T_3')$ is $y(P_1 + P_2)$ or $(0 : 0)$;
at most one of these is $(0 : 0)$.

$$X_3 = X_1Y_2Z_2T_1 + X_2Y_1Z_1T_2,$$
$$Z_3 = Z_1Z_2T_1T_2 + dX_1X_2Y_1Y_2,$$
$$Y_3 = Y_1Y_2Z_1Z_2 - X_1X_2T_1T_2,$$
$$T_3 = Z_1Z_2T_1T_2 - dX_1X_2Y_1Y_2,$$

$$X_3' = X_1Y_1Z_2T_2 + X_2Y_2Z_1T_1,$$
$$Z_3' = X_1X_2T_1T_2 + Y_1Y_2Z_1Z_2,$$
$$Y_3' = X_1Y_1Z_2T_2 - X_2Y_2Z_1T_1,$$
$$T_3' = X_1Y_2Z_2T_1 - X_2Y_1Z_1T_2.$$

Much, much, much simpler than Lange–Ruppert, Bosma–Lenstra.

Also much easier to prove.

5. Explicit Formulae

From [5, Chapter III, 2.3] it follows that $f = m^*(X/Z)$ and $g = m^*(Y/Z)$ are given by

$$f = \lambda^2 + a_1\lambda - \frac{X_1Z_2 + X_2Z_1}{Z_1Z_2} - a_2, \qquad g = -(\lambda + a_1) f - \nu - a_3,$$

where

$$\lambda = \frac{Y_1Z_2 - Y_2Z_1}{X_1Z_2 - X_2Z_1} \qquad \text{and} \qquad \nu = -\frac{Y_1X_2 - Y_2X_1}{X_1Z_2 - X_2Z_1}.$$

Applying the automorphism of $E \times E$ mapping (P_1, P_2) to $(P_1, -P_2)$ we find that

$$s^*(X/Z) = \kappa^2 + a_1\kappa - \frac{X_1Z_2 + X_2Z_1}{Z_1Z_2} - a_2$$

and

$$s^*(Y/Z) = -(\kappa + a_1)\, s^*(X/Z) - \mu - a_3,$$

where

$$\kappa = \frac{Y_1Z_2 + Y_2Z_1 + a_1X_2Z_1 + a_3Z_1Z_2}{X_1Z_2 - X_2Z_1}$$

and

$$\mu = -\frac{Y_1X_2 + Y_2X_1 + a_1X_1X_2 + a_3X_1Z_2}{X_1Z_2 - X_2Z_1}.$$

The bijection of Theorem 2 maps $(0:0:1)$ to the addition law given by $X_3^{(1)} = fZ_0$, $Y_3^{(1)} = gZ_0$, $Z_3^{(1)} = Z_0$, which in explicit terms is found to be given by

$$\begin{aligned}
X_3^{(1)} = {} & (X_1Y_2 - X_2Y_1)(Y_1Z_2 + Y_2Z_1) + (X_1Z_2 - X_2Z_1)\, Y_1Y_2 \\
& + a_1X_1X_2(Y_1Z_2 - Y_2Z_1) + a_1(X_1Y_2 - X_2Y_1)(X_1Z_2 + X_2Z_1) \\
& - a_2X_1X_2(X_1Z_2 - X_2Z_1) + a_3(X_1Y_2 - X_2Y_1)\, Z_1Z_2 \\
& + a_3(X_1Z_2 - X_2Z_1)(Y_1Z_2 + Y_2Z_1) \\
& - a_4(X_1Z_2 + X_2Z_1)(X_1Z_2 - X_2Z_1) \\
& - 3a_6(X_1Z_2 - X_2Z_1)\, Z_1Z_2,
\end{aligned}$$

$$\begin{aligned}
Y_3^{(1)} = {} & -3X_1X_2(X_1Y_2 - X_2Y_1) \\
& - Y_1Y_2(Y_1Z_2 - Y_2Z_1) - 2a_1(X_1Z_2 - X_2Z_1)\,Y_1Y_2 \\
& + (a_1^2 + 3a_2)\,X_1X_2(Y_1Z_2 - Y_2Z_1) \\
& - (a_1^2 + a_2)(X_1Y_2 + X_2Y_1)(X_1Z_2 - X_2Z_1) \\
& + (a_1a_2 - 3a_3)\,X_1X_2(X_1Z_2 - X_2Z_1) \\
& - (2a_1a_3 + a_4)(X_1Y_2 - X_2Y_1)\,Z_1Z_2 \\
& + a_4(X_1Z_2 + X_2Z_1)(Y_1Z_2 - Y_2Z_1) \\
& + (a_1a_4 - a_2a_3)(X_1Z_2 + X_2Z_1)(X_1Z_2 - X_2Z_1) \\
& + (a_3^2 + 3a_6)(Y_1Z_2 - Y_2Z_1)\,Z_1Z_2 \\
& + (3a_1a_6 - a_3a_4)(X_1Z_2 - X_2Z_1)\,Z_1Z_2, \\
Z_3^{(1)} = {} & 3X_1X_2(X_1Z_2 - X_2Z_1) - (Y_1Z_2 + Y_2Z_1)(Y_1Z_2 - Y_2Z_1) \\
& + a_1(X_1Y_2 - X_2Y_1)\,Z_1Z_2 - a_1(X_1Z_2 - X_2Z_1)(Y_1Z_2 + Y_2Z_1) \\
& + a_2(X_1Z_2 + X_2Z_1)(X_1Z_2 - X_2Z_1) - a_3(Y_1Z_2 - Y_2Z_1)\,Z_1Z_2 \\
& + a_4(X_1Z_2 - X_2Z_1)\,Z_1Z_2.
\end{aligned}$$

The corresponding exceptional divisor is $3 \cdot \Delta$, so a pair of points P_1, P_2 on E is exceptional for this addition law if and only if $P_1 = P_2$.

Multiplying the addition law just given by $s^*(Y/Z)$ we obtain the addition law corresponding to $(0:1:0)$. It reads as follows:

$$\begin{aligned}
X_3^{(2)} = {} & Y_1Y_2(X_1Y_2 + X_2Y_1) + a_1(2X_1Y_2 + X_2Y_1)\,X_2Y_1 + a_1^2X_1X_2^2Y_1 \\
& - a_2X_1X_2(X_1Y_2 + X_2Y_1) - a_1a_2X_1^2X_2^2 + a_3X_2Y_1(Y_1Z_2 + 2Y_2Z_1) \\
& + a_1a_3X_1X_2(Y_1Z_2 - Y_2Z_1) - a_1a_3(X_1Y_2 + X_2Y_1)(X_1Z_2 - X_2Z_1) \\
& - a_4X_1X_2(Y_1Z_2 + Y_2Z_1) - a_4(X_1Y_2 + X_2Y_1)(X_1Z_2 + X_2Z_1) \\
& - a_1^2a_3X_1^2X_2Z_2 - a_1a_4X_1X_2(2X_1Z_2 + X_2Z_1) \\
& - a_2a_3X_1X_2^2Z_1 - a_3^2X_1Z_2(2Y_2Z_1 + Y_1Z_2) \\
& - 3a_6(X_1Y_2 + X_2Y_1)\,Z_1Z_2 \\
& - 3a_6(X_1Z_2 + X_2Z_1)(Y_1Z_2 + Y_2Z_1) - a_1a_3^2X_1Z_2(X_1Z_2 + 2X_2Z_1) \\
& - 3a_1a_6X_1Z_2(X_1Z_2 + 2X_2Z_1) + a_3a_4(X_1Z_2 - 2X_2Z_1)\,X_2Z_1 \\
& - (a_1^2a_6 - a_1a_3a_4 + a_2a_3^2 + 4a_2a_6 - a_4^2)(Y_1Z_2 + Y_2Z_1)\,Z_1Z_2 \\
& - (a_1^3a_6 - a_1^2a_3a_4 + a_1a_2a_3^2 + 4a_1a_2a_6 - a_1a_4^2)\,X_1Z_1Z_2^2 \\
& - a_3^3(X_1Z_2 + X_2Z_1)\,Z_1Z_2 - 3a_3a_6(X_1Z_2 + 2X_2Z_1)\,Z_1Z_2 \\
& - (a_1^2a_3a_6 - a_1a_3^2a_4 + a_2a_3^3 + 4a_2a_3a_6 - a_3a_4^2)\,Z_1^2Z_2^2,
\end{aligned}$$

$$
\begin{aligned}
Y_3^{(2)} = {} & Y_1^2 Y_2^2 + a_1 X_2 Y_1^2 Y_2 + (a_1 a_2 - 3a_3) X_1 X_2^2 Y_1 \\
& + a_3 Y_1^2 Y_2 Z_2 - (a_2^2 - 3a_4) X_1^2 X_2^2 \\
& + (a_1 a_4 - a_2 a_3)(2X_1 Z_2 + X_2 Z_1) X_2 Y_1 \\
& + (a_1^2 a_4 - 2a_1 a_2 a_3 + 3a_3^2) X_1^2 X_2 Z_2 \\
& - (a_2 a_4 - 9a_6) X_1 X_2 (X_1 Z_2 + X_2 Z_1) \\
& + (3a_1 a_6 - a_3 a_4)(X_1 Z_2 + 2X_2 Z_1) Y_1 Z_2 \\
& + (3a_1^2 a_6 - 2a_1 a_3 a_4 + a_2 a_3^2 + 3a_2 a_6 - a_4^2) X_1 Z_2 (X_1 Z_2 + 2X_2 Z_1) \\
& - (3a_2 a_6 - a_4^2)(X_1 Z_2 + X_2 Z_1)(X_1 Z_2 - X_2 Z_1) \\
& + (a_1^3 a_6 - a_1^2 a_3 a_4 + a_1 a_2 a_3^2 - a_1 a_4^2 + 4a_1 a_2 a_6 - a_3^3 - 3a_3 a_6) Y_1 Z_1 Z_2^2 \\
& + (a_1^4 a_6 - a_1^3 a_3 a_4 + 5a_1^2 a_2 a_6 + a_1^2 a_2 a_3^2 - a_1 a_2 a_3 a_4 - a_1 a_3^3 - 3a_1 a_3 a_6 \\
& - a_1^2 a_4^2 + a_2^2 a_3^2 - a_2 a_4^2 + 4a_2^2 a_6 - a_3^2 a_4 - 3a_4 a_6) X_1 Z_1 Z_2^2 \\
& + (a_1^2 a_2 a_6 - a_1 a_2 a_3 a_4 + 3a_1 a_3 a_6 + a_2^2 a_3^2 - a_2 a_4^2 \\
& + 4a_2^2 a_6 - 2a_3^2 a_4 - 3a_4 a_6) X_2 Z_1^2 Z_2 \\
& + (a_1^3 a_3 a_6 - a_1^2 a_3^2 a_4 + a_1^2 a_4 a_6 + a_1 a_2 a_3^3 \\
& + 4a_1 a_2 a_3 a_6 - 2a_1 a_3 a_4^2 + a_2 a_3^2 a_4 \\
& + 4a_2 a_4 a_6 - a_3^4 - 6a_3^2 a_6 - a_4^3 - 9a_6^2) Z_1^2 Z_2^2, \\
Z_3^{(2)} = {} & 3X_1 X_2 (X_1 Y_2 + X_2 Y_1) + Y_1 Y_2 (Y_1 Z_2 + Y_2 Z_1) + 3a_1 X_1^2 X_2^2 \\
& + a_1 (2X_1 Y_2 + Y_1 X_2) Y_1 Z_2 + a_1^2 X_1 Z_2 (2X_2 Y_1 + X_1 Y_2) \\
& + a_2 X_1 X_2 (Y_1 Z_2 + Y_2 Z_1) \\
& + a_2 (X_1 Y_2 + X_2 Y_1)(X_1 Z_2 + X_2 Z_1) \\
& + a_1^3 X_1^2 X_2 Z_2 + a_1 a_2 X_1 X_2 (2X_1 Z_2 + X_2 Z_1) \\
& + 3a_3 X_1 X_2^2 Z_1 + a_3 Y_1 Z_2 (Y_1 Z_2 + 2Y_2 Z_1) \\
& + 2a_1 a_3 X_1 Z_2 (Y_1 Z_2 + Y_2 Z_1) \\
& + 2a_1 a_3 X_2 Y_1 Z_1 Z_2 + a_4 (X_1 Y_2 + X_2 Y_1) Z_1 Z_2 \\
& + a_4 (X_1 Z_2 + X_2 Z_1)(Y_1 Z_2 + Y_2 Z_1) \\
& + (a_1^2 a_3 + a_1 a_4) X_1 Z_2 (X_1 Z_2 + 2X_2 Z_1) + a_2 a_3 X_2 Z_1 (2X_1 Z_2 + X_2 Z_1) \\
& + a_3^2 Y_1 Z_1 Z_2^2 + (a_3^2 + 3a_6)(Y_1 Z_2 + Y_2 Z_1) Z_1 Z_2 \\
& + a_1 a_3^2 (2X_1 Z_2 + X_2 Z_1) Z_1 Z_2 + 3a_1 a_6 X_1 Z_1 Z_2^2 \\
& + a_3 a_4 (X_1 Z_2 + 2X_2 Z_1) Z_1 Z_2 + (a_3^3 + 3a_3 a_6) Z_1^2 Z_2^2.
\end{aligned}
$$

1987 Lenstra: Use Lange–Ruppert complete system of addition laws to computationally define group $E(R)$ for more general rings R—rings with trivial class group.

Define $\mathbf{P}^2(R) = \{(X : Y : Z) : X, Y, Z \in R;\ XR + YR + ZR = R\}$ where $(X : Y : Z)$ is the module $\{(\lambda X, \lambda Y, \lambda Z) : \lambda \in R\}$.

Define $E(R) = \{(X : Y : Z) \in \mathbf{P}^2(R) : Y^2 Z = X^3 + a_4 X Z^2 + a_6 Z^3\}$.

To define (and compute) sum
$(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : Z_2)$:

Consider (and compute)
Lange–Ruppert $(X_3 : Y_3 : Z_3)$,
$(X_3' : Y_3' : Z_3')$, $(X_3'' : Y_3'' : Z_3'')$.

Add these R-modules:

$$\{ \quad (\lambda X_3, \lambda Y_3, \lambda Z_3) + (\lambda' X_3', \lambda' Y_3', \lambda' Z_3') + (\lambda'' X_3'', \lambda'' Y_3'', \lambda'' Z_3'') : \lambda, \lambda', \lambda'' \in R\}.$$

Express as $(X : Y : Z)$,
using trivial class group of R.