ARITH 22, Lyon, France — June 25, 2015

Factoring integers with CADO-NFS

Jérémie Detrey

CARAMEL team, LORIA

INRIA Nancy – Grand Est, France

Jeremie.Detrey@loria.fr

Why factor?

- Cryptography:
  - Integer factorization is a (supposedly) difficult problem, but integer multiplication is not
  - E.g., basis for the security of the RSA public-key cryptosystem:
    - $\rightarrow$ private key: large primes p and q
    - $\rightarrow$ public key: $N = p \cdot q$
  - Key length recommendations
  - Break weak instances of RSA (short keys)

Why factor?

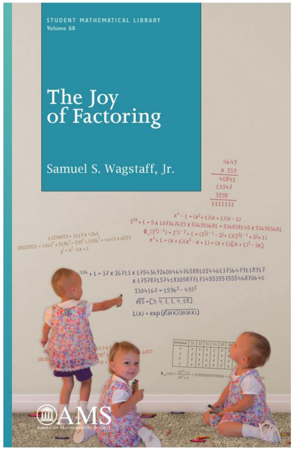
- Cryptography:
  - Integer factorization is a (supposedly) difficult problem, but integer multiplication is not
  - E.g., basis for the security of the RSA public-key cryptosystem:
    - $\rightarrow$ private key: large primes p and q
    - $\rightarrow$ public key: $N = p \cdot q$
  - Key length recommendations
  - Break weak instances of RSA (short keys)

- Number theory:
  - Cunningham tables: factorizations of $b^n \pm 1$
  - Aliquot sequences: $s_{n+1} = \sum_{d|s_n} d - s_n$
  - etc.

Why factor?

- Cryptography:
  - Integer factorization is a (supposedly) difficult problem, but integer multiplication is not
  - E.g., basis for the security of the RSA public-key cryptosystem:
    - $\rightarrow$ private key: large primes p and q
    - $\rightarrow$ public key: $N = p \cdot q$
  - Key length recommendations
  - Break weak instances of RSA (short keys)

- Number theory:
  - Cunningham tables: factorizations of $b^n \pm 1$
  - Aliquot sequences: $s_{n+1} = \sum_{d \mid s_n} d - s_n$
  - etc.



- For fun ☺

Factorization algorithms (I)

- Find small- to medium-size prime factors p of an integer N:

Factorization algorithms (I)

- Find small- to medium-size prime factors p of an integer N:
  - Trial division: $O(p)$

Factorization algorithms (I)

- Find small- to medium-size prime factors p of an integer N:
  - Trial division: $O(p) = O\left(\exp\left(\log p\right)\right)$
    $\rightarrow$ complexity exponential in $\log p$

Factorization algorithms (I)

- Find small- to medium-size prime factors p of an integer N:
  - Trial division: $O(p) = O\left(\exp\left(\log p\right)\right)$
    $\rightarrow$ complexity exponential in $\log p$
  - ρ method [Pollard, 1975]:

$$O(\sqrt{p})$$

Factorization algorithms (I)

- Find small- to medium-size prime factors p of an integer N:
  - Trial division: $O(p) = O\left(\exp\left(\log p\right)\right)$
    $\rightarrow$ complexity exponential in $\log p$
  - ρ method [Pollard, 1975]:

$$O(\sqrt{p}) = O\left(\exp\left(\frac{1}{2}\log p\right)\right)$$

Factorization algorithms (I)

- Find small- to medium-size prime factors p of an integer N:
  - Trial division: $O(p) = O\left(\exp\left(\log p\right)\right)$
    $\rightarrow$ complexity exponential in $\log p$
  - ρ method [Pollard, 1975]:
    $$O(\sqrt{p}) = O\left(\exp\left(\frac{1}{2}\log p\right)\right)$$
  - $p-1$ [Pollard, 1974] and $p+1$ [Williams, 1982]

Factorization algorithms (I)

- Find small- to medium-size prime factors p of an integer N:
  - Trial division: $O(p) = O\left(\exp\left(\log p\right)\right)$
    $\rightarrow$ complexity exponential in $\log p$
  - ρ method [Pollard, 1975]:

$$O(\sqrt{p}) = O\left(\exp\left(\frac{1}{2}\log p\right)\right)$$

  - $p-1$ [Pollard, 1974] and $p+1$ [Williams, 1982]
  - ECM (Elliptic Curve Method) [Lenstra, 1987]:

$$O\left(\exp\left(\sqrt{2\log p \log\log p}\right)\right)$$

    $\rightarrow$ subexponential complexity!

Factorization algorithms (II)

- Find all prime factors of an integer N:

Factorization algorithms (II)

- Find all prime factors of an integer N:
  - SQUFOF (SQUare FOrms Factorization) [Shanks, ca. 1975]:

$$O(\sqrt[4]{N}) = O\left(\exp\left(\frac{1}{4}\log N\right)\right)$$

$\rightarrow$ complexity exponential in $\log N$

Factorization algorithms (II)

- Find all prime factors of an integer N:
  - SQUFOF (SQUare FOrms Factorization) [Shanks, ca. 1975]:

$$O(\sqrt[4]{N}) = O\left(\exp\left(\frac{1}{4}\log N\right)\right)$$

    $\rightarrow$ complexity exponential in $\log N$
  - CFRAC (Continued FRACtions) [Morrison & Brillhart, 1975]:

$$O\left(\exp\left(\sqrt{2\log N \log\log N}\right)\right)$$

    $\rightarrow$ subexponential complexity!

Factorization algorithms (III)

- Find all prime factors of an integer N:
  - QS (Quadratic Sieve) [Pomerance, 1981] and MPQS (Multiple Polynomial QS) [Silverman, 1987] in

$$O\left(\exp\left(\sqrt{\log N \log\log N}\right)\right)$$

Factorization algorithms (III)

- Find all prime factors of an integer N:
  - QS (Quadratic Sieve) [Pomerance, 1981] and MPQS (Multiple Polynomial QS) [Silverman, 1987] in

$$O\left(\exp\left(\sqrt{\log N \log\log N}\right)\right)$$

  - SNFS (Special Number Field Sieve) [Lenstra, Lenstra, Manasse, & Pollard, 1990]:

$$O\left(\exp\left(\sqrt[3]{\frac{32}{9}}\,(\log N)^{1/3}\,(\log\log N)^{2/3}\right)\right)$$

Factorization algorithms (III)

- Find all prime factors of an integer N:
  - QS (Quadratic Sieve) [Pomerance, 1981] and MPQS (Multiple Polynomial QS) [Silverman, 1987] in

$$O\left(\exp\left(\sqrt{\log N \log\log N}\right)\right)$$

  - SNFS (Special Number Field Sieve) [Lenstra, Lenstra, Manasse, & Pollard, 1990]:

$$O\left(\exp\left(\sqrt[3]{\frac{32}{9}}\,(\log N)^{1/3}\,(\log\log N)^{2/3}\right)\right)$$

  - (G)NFS (General Number Field Sieve) [Buhler, Lenstra, & Pomerance, 1993]:

$$O\left(\exp\left(\sqrt[3]{\frac{64}{9}}\,(\log N)^{1/3}\,(\log\log N)^{2/3}\right)\right)$$

Current factorization records

- ECM (small- to medium-size factors):
  - 2013: found 83-digit-factor of $7^{337} + 1$ (285 digits)

Current factorization records

- ECM (small- to medium-size factors):
  - 2013: found 83-digit-factor of $7^{337} + 1$ (285 digits)
- SNFS (numbers of a special form):
  - 1990: factorization of $F_9 = 2^{2^9} + 1$ (155 digits) in $\sim$ 340 CPU-years
  - ...
  - 2011–12: fact. of $2^{1061} - 1$ (320 digits) in $\sim$ 335 CPU-years
  - 2010–14: fact. of 17 numbers of the form $2^n - 1$ for $1007 \leq n \leq 1199$ (304–361 digits) in $\sim$ 7500 core-years

Current factorization records

- ECM (small- to medium-size factors):
  - 2013: found 83-digit-factor of $7^{337}+1$ (285 digits)
- SNFS (numbers of a special form):
  - 1990: factorization of $F_9 = 2^{2^9}+1$ (155 digits) in $\sim$ 340 CPU-years
  - ...
  - 2011–12: fact. of $2^{1061}-1$ (320 digits) in $\sim$ 335 CPU-years
  - 2010–14: fact. of 17 numbers of the form 2^n-1 for $1007 \leq n \leq 1199$ (304–361 digits) in $\sim$ 7500 core-years
- GNFS (general numbers, esp. RSA moduli):
  - 1996: fact. of RSA-130 (130 digits) in $\sim$ 17 CPU-years
  - ...
  - 2007–09: fact. of RSA-768 (232 digits) in $\sim$ 2000 core-years

Current factorization records

- ECM (small- to medium-size factors):
  - 2013: found 83-digit-factor of $7^{337}+1$ (285 digits)
- SNFS (numbers of a special form):
  - 1990: factorization of $F_9 = 2^{2^9}+1$ (155 digits) in $\sim$ 340 CPU-years
  - ...
  - 2011–12: fact. of $2^{1061}-1$ (320 digits) in $\sim$ 335 CPU-years
  - 2010–14: fact. of 17 numbers of the form 2^n-1 for $1007 \le n \le 1199$ (304–361 digits) in $\sim$ 7500 core-years
- GNFS (general numbers, esp. RSA moduli):
  - 1996: fact. of RSA-130 (130 digits) in $\sim$ 17 CPU-years
  - ...
  - 2007–09: fact. of RSA-768 (232 digits) in $\sim$ 2000 core-years
- Quantum computer:
  - 2012: fact. of 56153 (a whopping 5 digits!)

Free (as in free speech) factorization software

- $p - 1$, $p + 1$, and ECM:
  - GMP-ECM [Zimmermann *et al.*]:
    `http://ecm.gforge.inria.fr/`

Free (as in free speech) factorization software

- $p - 1$, $p + 1$, and ECM:
  - GMP-ECM [Zimmermann *et al.*]:
    `http://ecm.gforge.inria.fr/`
- QS and MPQS:
  - YAFU [Buhrow]:
    `http://yafu.sourceforge.net/`

Free (as in free speech) factorization software

- $p-1$, $p+1$, and ECM:
  - GMP-ECM [Zimmermann *et al.*]:
    `http://ecm.gforge.inria.fr/`
- QS and MPQS:
  - YAFU [Buhrow]:
    `http://yafu.sourceforge.net/`
- SNFS and GNFS:
  - NFS@home [Childers]:
    `http://escatter11.fullerton.edu/nfs/`
  - Msieve [Papadopoulos]:
    `http://www.boo.net/~jasonp/qs.html`
  - CADO-NFS:
    `http://cado-nfs.gforge.inria.fr/`

CADO-NFS

- Mostly developed in the CARAMEL team in Nancy, France, with several regular external contributors:
  - Shi Bai (AriC team, LIP, Lyon, France)
  - Cyril Bouvier (CARAMEL)
  - Alain Filbois (Inria Nancy – Grand Est, France)
  - Pierrick Gaudry (CARAMEL)
  - Laurent Imbert (ECO team, LIRMM, Montpellier, France)
  - Alexander Kruppa (CARAMEL)
  - François Morain (GRACE team, LIX, Saclay, France)
  - Emmanuel Thomé (CARAMEL)
  - Paul Zimmermann (CARAMEL)

CADO-NFS

- Mostly developed in the CARAMEL team in Nancy, France, with several regular external contributors:
  - Shi Bai (AriC team, LIP, Lyon, France)
  - Cyril Bouvier (CARAMEL)
  - Alain Filbois (Inria Nancy – Grand Est, France)
  - Pierrick Gaudry (CARAMEL)
  - Laurent Imbert (ECO team, LIRMM, Montpellier, France)
  - Alexander Kruppa (CARAMEL)
  - François Morain (GRACE team, LIX, Saclay, France)
  - Emmanuel Thomé (CARAMEL)
  - Paul Zimmermann (CARAMEL)
- Started in 2007, last release (2.1.1) in 2014, still under heavy development (10k commits, almost 300k lines of code)

CADO-NFS

- Mostly developed in the CARAMEL team in Nancy, France, with several regular external contributors:
  - Shi Bai (AriC team, LIP, Lyon, France)
  - Cyril Bouvier (CARAMEL)
  - Alain Filbois (Inria Nancy – Grand Est, France)
  - Pierrick Gaudry (CARAMEL)
  - Laurent Imbert (ECO team, LIRMM, Montpellier, France)
  - Alexander Kruppa (CARAMEL)
  - François Morain (GRACE team, LIX, Saclay, France)
  - Emmanuel Thomé (CARAMEL)
  - Paul Zimmermann (CARAMEL)
- Started in 2007, last release (2.1.1) in 2014, still under heavy development (10k commits, almost 300k lines of code)
- Support for integer factorization (GNFS and SNFS), but also discrete logarithm in finite fields (FFS, NFS-DL, NFS-HD)

CADO-NFS

- Mostly developed in the CARAMEL team in Nancy, France, with several regular external contributors:
  - Shi Bai (AriC team, LIP, Lyon, France)
  - Cyril Bouvier (CARAMEL)
  - Alain Filbois (Inria Nancy – Grand Est, France)
  - Pierrick Gaudry (CARAMEL)
  - Laurent Imbert (ECO team, LIRMM, Montpellier, France)
  - Alexander Kruppa (CARAMEL)
  - François Morain (GRACE team, LIX, Saclay, France)
  - Emmanuel Thomé (CARAMEL)
  - Paul Zimmermann (CARAMEL)
- Started in 2007, last release (2.1.1) in 2014, still under heavy development (10k commits, almost 300k lines of code)
- Support for integer factorization (GNFS and SNFS), but also discrete logarithm in finite fields (FFS, NFS-DL, NFS-HD)
- Website: `http://cado-nfs.gforge.inria.fr/`

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields

$$\mathbb{Z}[X]$$

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields
  - f_1 and $f_2 \in \mathbb{Z}[X]$ two polynomials, irreducible and coprime over $\mathbb{Q}$

$$\mathbb{Z}[X]$$

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields
  - f_1 and $f_2 \in \mathbb{Z}[X]$ two polynomials, irreducible and coprime over $\mathbb{Q}$

$$\begin{tikzcd}
 & \mathbb{Z}[X] \arrow[dl, "X \mapsto X \bmod f_1"'] \\
\mathbb{Z}[X]/(f_1(X)) &
\end{tikzcd}$$

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields
  - f_1 and $f_2 \in \mathbb{Z}[X]$ two polynomials, irreducible and coprime over $\mathbb{Q}$
  - α_i root of f_i: $\mathbb{Q}(\alpha_i)$ is an algebraic number field

$$\begin{tikzcd} & \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \\ \mathbb{Z}[\alpha_1] \subset \mathcal{O}_{\mathbb{Q}(\alpha_1)} & \end{tikzcd}$$

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields
  - f_1 and $f_2 \in \mathbb{Z}[X]$ two polynomials, irreducible and coprime over $\mathbb{Q}$
  - α_i root of f_i: $\mathbb{Q}(\alpha_i)$ is an algebraic number field

$$\begin{tikzcd}
 & \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\mathbb{Z}[\alpha_1] & & \mathcal{O}_{\mathbb{Q}(\alpha_2)} \supset \mathbb{Z}[\alpha_2]
\end{tikzcd}$$

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields
  - f_1 and $f_2 \in \mathbb{Z}[X]$ two polynomials, irreducible and coprime over $\mathbb{Q}$
  - α_i root of f_i: $\mathbb{Q}(\alpha_i)$ is an algebraic number field
  - f_1 and f_2 chosen such that they have a common root m in $\mathbb{Z}/N\mathbb{Z}$

$$\begin{tikzcd} & \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\ \mathbb{Z}[\alpha_1] & & \mathbb{Z}[\alpha_2] \end{tikzcd}$$

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields
  - f_1 and $f_2 \in \mathbb{Z}[X]$ two polynomials, irreducible and coprime over $\mathbb{Q}$
  - α_i root of f_i: $\mathbb{Q}(\alpha_i)$ is an algebraic number field
  - f_1 and f_2 chosen such that they have a common root m in $\mathbb{Z}/N\mathbb{Z}$

$$
\begin{tikzcd}
 & \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}
$$

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields
  - f_1 and $f_2 \in \mathbb{Z}[X]$ two polynomials, irreducible and coprime over $\mathbb{Q}$
  - α_i root of f_i: $\mathbb{Q}(\alpha_i)$ is an algebraic number field
  - f_1 and f_2 chosen such that they have a common root m in $\mathbb{Z}/N\mathbb{Z}$

$$\begin{tikzcd}
& \Gamma(X) \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
& \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields
  - f_1 and $f_2 \in \mathbb{Z}[X]$ two polynomials, irreducible and coprime over $\mathbb{Q}$
  - α_i root of f_i: $\mathbb{Q}(\alpha_i)$ is an algebraic number field
  - f_1 and f_2 chosen such that they have a common root m in $\mathbb{Z}/N\mathbb{Z}$

$$\begin{tikzcd}
& \Gamma(X) \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\Gamma(\alpha_1) \in \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \ni \Gamma(\alpha_2) \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
& \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields
  - f_1 and $f_2 \in \mathbb{Z}[X]$ two polynomials, irreducible and coprime over $\mathbb{Q}$
  - α_i root of f_i: $\mathbb{Q}(\alpha_i)$ is an algebraic number field
  - f_1 and f_2 chosen such that they have a common root m in $\mathbb{Z}/N\mathbb{Z}$

$$\begin{tikzcd}
 & \Gamma(X) \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\gamma_1(\alpha_1)^2 \stackrel{?}{=} \Gamma(\alpha_1) \in \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \ni \Gamma(\alpha_2) \stackrel{?}{=} \gamma_2(\alpha_2)^2 \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- Based on Fermat's factoring method (congruence of squares):
  - Find two integers x and y such that $x^2 \equiv y^2 \pmod{N}$
  - With good probability, $\gcd(x \pm y, N)$ gives a non-trivial factor of N
- Obtain such equalities through two number fields
  - f_1 and $f_2 \in \mathbb{Z}[X]$ two polynomials, irreducible and coprime over $\mathbb{Q}$
  - α_i root of f_i: $\mathbb{Q}(\alpha_i)$ is an algebraic number field
  - f_1 and f_2 chosen such that they have a common root m in $\mathbb{Z}/N\mathbb{Z}$

$$
\begin{tikzcd}
 & \Gamma(X) \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\gamma_1(\alpha_1)^2 \stackrel{?}{=} \Gamma(\alpha_1) \in \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \ni \Gamma(\alpha_2) \stackrel{?}{=} \gamma_2(\alpha_2)^2 \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}
$$

$$\Rightarrow \gamma_1(m)^2 \equiv \gamma_2(m)^2 \bmod N$$

The Number Field Sieve

- How can one find such a polynomial $\Gamma(X)$?

The Number Field Sieve

- How can one find such a polynomial $\Gamma(X)$?

$$\begin{tikzcd}
 & \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- How can one find such a polynomial $\Gamma(X)$?
- For all pairs of coprime integers $(a, b) \in [-A, A] \times]0, A]$:

$$\begin{tikzcd}
 & \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- How can one find such a polynomial $\Gamma(X)$?
- For all pairs of coprime integers $(a, b) \in [-A, A] \times]0, A]$:
  - Consider the polynomial $a - bX$ in the diagram

$$
\begin{tikzcd}
& \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
& \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}
$$

The Number Field Sieve

- How can one find such a polynomial $\Gamma(X)$?
- For all pairs of coprime integers $(a, b) \in [-A, A] \times]0, A]$:
  - Consider the polynomial $a - bX$ in the diagram

$$\begin{tikzcd}
 & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- How can one find such a polynomial $\Gamma(X)$?
- For all pairs of coprime integers $(a, b) \in [-A, A] \times]0, A]$:
  - Consider the polynomial $a - bX$ in the diagram

$$\begin{tikzcd}
 & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
a - b\alpha_1 \in \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \ni a - b\alpha_2 \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- How can one find such a polynomial $\Gamma(X)$?
- For all pairs of coprime integers $(a, b) \in [-A, A] \times]0, A]$:
  - Consider the polynomial $a - bX$ in the diagram
  - Try to factor each $a - b\alpha_i$ into a product of primes $\leq$ bound B_i

$$\begin{tikzcd}
 & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
a - b\alpha_1 \in \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \ni a - b\alpha_2 \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- How can one find such a polynomial $\Gamma(X)$?
- For all pairs of coprime integers $(a, b) \in [-A, A] \times]0, A]$:
  - Consider the polynomial $a - bX$ in the diagram
  - Try to factor each $a - b\alpha_i$ into a product of primes $\leq$ bound B_i

$$\begin{tikzcd}
 & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\prod_j p_{1,j}^{e_{1,j}} = a - b\alpha_1 \in \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \ni a - b\alpha_2 = \prod_j p_{2,j}^{e_{2,j}} \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- How can one find such a polynomial $\Gamma(X)$?
- For all pairs of coprime integers $(a, b) \in [-A, A] \times]0, A]$:
  - Consider the polynomial $a - bX$ in the diagram
  - Try to factor each $a - b\alpha_i$ into a product of primes $\leq$ bound B_i
  - Such a pair is called a relation: add (a, b) to $\mathcal{R}$ (set of relations)

$$\begin{tikzcd}
 & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\prod_j p_{1,j}^{e_{1,j}} = a - b\alpha_1 \in \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \ni a - b\alpha_2 = \prod_j p_{2,j}^{e_{2,j}} \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$(a_1, b_1) : \qquad a_1 - b_1\alpha_1 = p_{1,1}^2\, p_{1,2}\, p_{1,3}^2 \qquad\qquad a_1 - b_1\alpha_2 = p_{2,1}\, p_{2,2}^4$$

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i]\text{, for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{lll} (a_1, b_1): & a_1 - b_1\alpha_1 = p_{1,1}^2\, p_{1,2}\, p_{1,3}^2 & a_1 - b_1\alpha_2 = p_{2,1}\, p_{2,2}^4 \\ (a_2, b_2): & a_2 - b_2\alpha_1 = p_{1,2}^3\, p_{1,3} & a_2 - b_2\alpha_2 = p_{2,1}\, p_{2,2}\, p_{2,3} \end{array}$$

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{llll}
(a_1, b_1): & a_1 - b_1\alpha_1 = p_{1,1}^2\, p_{1,2}\, p_{1,3}^2 & & a_1 - b_1\alpha_2 = p_{2,1}\, p_{2,2}^4 \\
(a_2, b_2): & a_2 - b_2\alpha_1 = \phantom{p_{1,1}^2}\, p_{1,2}^3\, p_{1,3} & & a_2 - b_2\alpha_2 = p_{2,1}\, p_{2,2}\, p_{2,3} \\
(a_3, b_3): & a_3 - b_3\alpha_1 = p_{1,1}\, p_{1,2}^2\, p_{1,3} & & a_3 - b_3\alpha_2 = \phantom{p_{2,1}}\, p_{2,2}\, p_{2,3}^3
\end{array}$$

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{llll}
(a_1, b_1): & a_1 - b_1\alpha_1 = p_{1,1}^2\, p_{1,2}\, p_{1,3}^2 & & a_1 - b_1\alpha_2 = p_{2,1}\, p_{2,2}^4 \\
(a_2, b_2): & a_2 - b_2\alpha_1 = \phantom{p_{1,1}}\; p_{1,2}^3\, p_{1,3} & & a_2 - b_2\alpha_2 = p_{2,1}\, p_{2,2}\, p_{2,3} \\
(a_3, b_3): & a_3 - b_3\alpha_1 = p_{1,1}\, p_{1,2}^2\, p_{1,3} & & a_3 - b_3\alpha_2 = \phantom{p_{2,1}}\; p_{2,2}\, p_{2,3}^3 \\
(a_4, b_4): & a_4 - b_4\alpha_1 = p_{1,1}\, \phantom{p_{1,2}}\; p_{1,3} & & a_4 - b_4\alpha_2 = p_{2,1}^2\, p_{2,2}\, p_{2,3}
\end{array}$$

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{llll}
(a_1, b_1): & a_1 - b_1\alpha_1 = p_{1,1}^2\, p_{1,2}\, p_{1,3}^2 & & a_1 - b_1\alpha_2 = p_{2,1}\, p_{2,2}^4 \\
(a_2, b_2): & a_2 - b_2\alpha_1 = p_{1,2}^3\, p_{1,3} & & a_2 - b_2\alpha_2 = p_{2,1}\, p_{2,2}\, p_{2,3} \\
(a_3, b_3): & a_3 - b_3\alpha_1 = p_{1,1}\, p_{1,2}^2\, p_{1,3} & & a_3 - b_3\alpha_2 = p_{2,2}\, p_{2,3}^3 \\
(a_4, b_4): & a_4 - b_4\alpha_1 = p_{1,1}\, p_{1,3} & & a_4 - b_4\alpha_2 = p_{2,1}^2\, p_{2,2}\, p_{2,3}
\end{array}$$

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{lllll}
(a_1, b_1): & a_1 - b_1\alpha_1 = p_{1,1}^2\, p_{1,2}\, p_{1,3}^2 & & a_1 - b_1\alpha_2 = p_{2,1}\, p_{2,2}^4 & \\
(a_2, b_2): & a_2 - b_2\alpha_1 = p_{1,2}^3\, p_{1,3} & & a_2 - b_2\alpha_2 = p_{2,1}\, p_{2,2}\, p_{2,3} & \\
(a_3, b_3): & a_3 - b_3\alpha_1 = p_{1,1}\, p_{1,2}^2\, p_{1,3} & & a_3 - b_3\alpha_2 = p_{2,2}\, p_{2,3}^3 & \\
(a_4, b_4): & a_4 - b_4\alpha_1 = p_{1,1}\, p_{1,3} & & a_4 - b_4\alpha_2 = p_{2,1}^2\, p_{2,2}\, p_{2,3} &
\end{array}$$

$$\prod_{i\in\{1,2,4\}} (a_i - b_i\alpha_1) = p_{1,1}^4\, p_{1,2}^4\, p_{1,3}^4 \qquad \prod_{i\in\{1,2,4\}} (a_i - b_i\alpha_2) = p_{2,1}^4\, p_{2,2}^6\, p_{2,3}^2$$

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{llll}
(a_1, b_1): & a_1 - b_1\alpha_1 = p_{1,1}^2\, p_{1,2}\, p_{1,3}^2 & & a_1 - b_1\alpha_2 = p_{2,1}\, p_{2,2}^4 \\
(a_2, b_2): & a_2 - b_2\alpha_1 = p_{1,2}^3\, p_{1,3} & & a_2 - b_2\alpha_2 = p_{2,1}\, p_{2,2}\, p_{2,3} \\
(a_3, b_3): & a_3 - b_3\alpha_1 = p_{1,1}\, p_{1,2}^2\, p_{1,3} & & a_3 - b_3\alpha_2 = p_{2,2}\, p_{2,3}^3 \\
(a_4, b_4): & a_4 - b_4\alpha_1 = p_{1,1}\, p_{1,3} & & a_4 - b_4\alpha_2 = p_{2,1}^2\, p_{2,2}\, p_{2,3}
\end{array}$$

$$\prod_{i\in\{1,2,4\}} (a_i - b_i\alpha_1) = p_{1,1}^4\, p_{1,2}^4\, p_{1,3}^4 \qquad \prod_{i\in\{1,2,4\}} (a_i - b_i\alpha_2) = p_{2,1}^4\, p_{2,2}^6\, p_{2,3}^2$$

- Tantamount to finding a vector of the left-kernel of the matrix over $\mathbb{F}_2$ formed by the exponents of the primes in the relations

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{llll}
(a_1, b_1): & a_1 - b_1\alpha_1 = p_{1,1}^2\, p_{1,2}\, p_{1,3}^2 & & a_1 - b_1\alpha_2 = p_{2,1}\, p_{2,2}^4 \\
(a_2, b_2): & a_2 - b_2\alpha_1 = p_{1,2}^3\, p_{1,3} & & a_2 - b_2\alpha_2 = p_{2,1}\, p_{2,2}\, p_{2,3} \\
(a_3, b_3): & a_3 - b_3\alpha_1 = p_{1,1}\, p_{1,2}^2\, p_{1,3} & & a_3 - b_3\alpha_2 = p_{2,2}\, p_{2,3}^3 \\
(a_4, b_4): & a_4 - b_4\alpha_1 = p_{1,1}\, p_{1,3} & & a_4 - b_4\alpha_2 = p_{2,1}^2\, p_{2,2}\, p_{2,3}
\end{array}$$

- Tantamount to finding a vector of the left-kernel of the matrix over $\mathbb{F}_2$ formed by the exponents of the primes in the relations

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{lllll}
(a_1, b_1): & a_1 - b_1\alpha_1 \equiv & (\,2 \quad 1 \quad 2\,) & a_1 - b_1\alpha_2 \equiv & (\,1 \quad 4 \quad 0\,) \\
(a_2, b_2): & a_2 - b_2\alpha_1 = & \phantom{p_{1,1}}\, p_{1,2}^3\, p_{1,3} & a_2 - b_2\alpha_2 = & p_{2,1}\, p_{2,2}\, p_{2,3} \\
(a_3, b_3): & a_3 - b_3\alpha_1 = & p_{1,1}\, p_{1,2}^2\, p_{1,3} & a_3 - b_3\alpha_2 = & \phantom{p_{2,1}}\, p_{2,2}\, p_{2,3}^3 \\
(a_4, b_4): & a_4 - b_4\alpha_1 = & p_{1,1}\, \phantom{p_{1,2}}\, p_{1,3} & a_4 - b_4\alpha_2 = & p_{2,1}^2\, p_{2,2}\, p_{2,3}
\end{array}$$

- Tantamount to finding a vector of the left-kernel of the matrix over $\mathbb{F}_2$ formed by the exponents of the primes in the relations

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{lll}
(a_1, b_1) : & a_1 - b_1\alpha_1 \equiv \begin{pmatrix} 2 & 1 & 2 \end{pmatrix} & a_1 - b_1\alpha_2 \equiv \begin{pmatrix} 1 & 4 & 0 \end{pmatrix} \\
(a_2, b_2) : & a_2 - b_2\alpha_1 \equiv \begin{pmatrix} 0 & 3 & 1 \end{pmatrix} & a_2 - b_2\alpha_2 \equiv \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\
(a_3, b_3) : & a_3 - b_3\alpha_1 \equiv \begin{pmatrix} 1 & 2 & 1 \end{pmatrix} & a_3 - b_3\alpha_2 \equiv \begin{pmatrix} 0 & 1 & 3 \end{pmatrix} \\
(a_4, b_4) : & a_4 - b_4\alpha_1 \equiv \begin{pmatrix} 2 & 0 & 1 \end{pmatrix} & a_4 - b_4\alpha_2 \equiv \begin{pmatrix} 2 & 1 & 1 \end{pmatrix}
\end{array}$$

- Tantamount to finding a vector of the left-kernel of the matrix over $\mathbb{F}_2$ formed by the exponents of the primes in the relations

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{lll}
(a_1, b_1): & a_1 - b_1\alpha_1 \equiv \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} & a_1 - b_1\alpha_2 \equiv \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \\
(a_2, b_2): & a_2 - b_2\alpha_1 \equiv \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} & a_2 - b_2\alpha_2 \equiv \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\
(a_3, b_3): & a_3 - b_3\alpha_1 \equiv \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} & a_3 - b_3\alpha_2 \equiv \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \\
(a_4, b_4): & a_4 - b_4\alpha_1 \equiv \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} & a_4 - b_4\alpha_2 \equiv \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}
\end{array}$$

- Tantamount to finding a vector of the left-kernel of the matrix over $\mathbb{F}_2$ formed by the exponents of the primes in the relations

The Number Field Sieve

- Once enough relations were collected, find subset $\mathcal{S} \subset \mathcal{R}$ such that

$$\prod_{(a,b)\in\mathcal{S}} (a - b\alpha_i) \text{ is a square in } \mathbb{Z}[\alpha_i], \text{ for both } i \in \{1, 2\}$$

- Example:

$$\begin{array}{llll}
(a_1, b_1): & a_1 - b_1\alpha_1 \equiv \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} & & a_1 - b_1\alpha_2 \equiv \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \\
(a_2, b_2): & a_2 - b_2\alpha_1 \equiv \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} & & a_2 - b_2\alpha_2 \equiv \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \\
(a_3, b_3): & a_3 - b_3\alpha_1 \equiv \begin{pmatrix} 1 & 0 & 1 \end{pmatrix} & & a_3 - b_3\alpha_2 \equiv \begin{pmatrix} 0 & 1 & 1 \end{pmatrix} \\
(a_4, b_4): & a_4 - b_4\alpha_1 \equiv \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} & & a_4 - b_4\alpha_2 \equiv \begin{pmatrix} 0 & 1 & 1 \end{pmatrix}
\end{array}$$

$$\prod_{i\in\{1,2,4\}} (a_i - b_i\alpha_1) \equiv \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \qquad \prod_{i\in\{1,2,4\}} (a_i - b_i\alpha_2) \equiv \begin{pmatrix} 0 & 0 & 0 \end{pmatrix}$$

- Tantamount to finding a vector of the left-kernel of the matrix over $\mathbb{F}_2$ formed by the exponents of the primes in the relations

The Number Field Sieve

$$\begin{tikzcd}[column sep=small]
 & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\prod_j p_{1,j}^{e_{1,j}} = a - b\alpha_1 \in \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \ni a - b\alpha_2 = \prod_j p_{2,j}^{e_{2,j}} \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- Slight problem: no unique factorization of numbers in $\mathbb{Z}[\alpha_i]$ or $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$

$$
\begin{tikzcd}
& a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\cancel{\prod_j \mathfrak{p}_{1,j}^{e_{1,j}}} = a - b\alpha_1 \in \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \ni a - b\alpha_2 = \cancel{\prod_j \mathfrak{p}_{2,j}^{e_{2,j}}} \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
& \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}
$$

The Number Field Sieve

- Slight problem: no unique factorization of numbers in $\mathbb{Z}[\alpha_i]$ or $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$
- However, $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$ is a Dedekind domain: unique factorization of ideals into products of prime ideals

$$\begin{tikzcd} & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\ \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\ & \mathbb{Z}/N\mathbb{Z} & \end{tikzcd}$$

The Number Field Sieve

- Slight problem: no unique factorization of numbers in $\mathbb{Z}[\alpha_i]$ or $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$
- However, $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$ is a Dedekind domain: unique factorization of ideals into products of prime ideals

$$\begin{tikzcd}
 & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\langle a - b\alpha_1 \rangle \subset \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \supset \langle a - b\alpha_2 \rangle \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- Slight problem: no unique factorization of numbers in $\mathbb{Z}[\alpha_i]$ or $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$
- However, $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$ is a Dedekind domain: unique factorization of ideals into products of prime ideals
  - Prime ideals $\mathfrak{p}$ of $\mathbb{Z}[\alpha_i]$ given by integers (p, r) such that p is prime and $f_i(r) \equiv 0 \pmod{p}$

$$
\begin{tikzcd}
 & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\langle a - b\alpha_1 \rangle \subset \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \supset \langle a - b\alpha_2 \rangle \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}
$$

The Number Field Sieve

- Slight problem: no unique factorization of numbers in $\mathbb{Z}[\alpha_i]$ or $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$
- However, $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$ is a Dedekind domain: unique factorization of ideals into products of prime ideals
  - Prime ideals $\mathfrak{p}$ of $\mathbb{Z}[\alpha_i]$ given by integers (p, r) such that p is prime and $f_i(r) \equiv 0 \pmod{p}$
  - $\mathfrak{p}^e$ "divides" $\langle a - b\alpha_i \rangle$ iff. $a - br \equiv 0 \pmod{p}$ and $p^e | N_i(a - b\alpha_i)$, where $N_i(a - b\alpha_i) = f_i(a/b)b^{\deg f_i}$ is called the norm of $a - b\alpha_i$

$$
\begin{tikzcd}
 & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\langle a - b\alpha_1 \rangle \subset \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \supset \langle a - b\alpha_2 \rangle \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}
$$

The Number Field Sieve

- Slight problem: no unique factorization of numbers in $\mathbb{Z}[\alpha_i]$ or $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$
- However, $\mathcal{O}_{\mathbb{Q}(\alpha_i)}$ is a Dedekind domain: unique factorization of ideals into products of prime ideals
  - Prime ideals $\mathfrak{p}$ of $\mathbb{Z}[\alpha_i]$ given by integers (p, r) such that p is prime and $f_i(r) \equiv 0 \pmod{p}$
  - $\mathfrak{p}^e$ "divides" $\langle a - b\alpha_i \rangle$ iff. $a - br \equiv 0 \pmod{p}$ and $p^e | N_i(a - b\alpha_i)$, where $N_i(a - b\alpha_i) = f_i(a/b)b^{\deg f_i}$ is called the norm of $a - b\alpha_i$

$$\begin{tikzcd}
 & a - bX \in \mathbb{Z}[X] \arrow[dl, "X \mapsto \alpha_1"'] \arrow[dr, "X \mapsto \alpha_2"] & \\
\mathfrak{u}_1 \prod_j \mathfrak{p}_{1,j}^{e_{1,j}} = \langle a - b\alpha_1 \rangle \subset \mathbb{Z}[\alpha_1] \arrow[dr, "\alpha_1 \mapsto m \bmod N"'] & & \mathbb{Z}[\alpha_2] \supset \langle a - b\alpha_2 \rangle = \mathfrak{u}_2 \prod_j \mathfrak{p}_{2,j}^{e_{2,j}} \arrow[dl, "\alpha_2 \mapsto m \bmod N"] \\
 & \mathbb{Z}/N\mathbb{Z} &
\end{tikzcd}$$

The Number Field Sieve

- Let's recap!

The Number Field Sieve

- Let's recap!
  - Sieving domain: coprime pairs (a, b) in $[-A, A] \times]0, A]$

The Number Field Sieve

- Let's recap!
  - Sieving domain: coprime pairs (a, b) in $[-A, A]\times]0, A]$
  - Factor base $\mathcal{B}_i$: prime ideals $\mathfrak{p} = (p, r)$ of $\mathbb{Z}[\alpha_i]$ with $p \leq B_i$

The Number Field Sieve

- Let's recap!
  - Sieving domain: coprime pairs (a, b) in $[-A, A] \times]0, A]$
  - Factor base $\mathcal{B}_i$: prime ideals $\mathfrak{p} = (p, r)$ of $\mathbb{Z}[\alpha_i]$ with $p \leq B_i$
- For each (a, b) pair in the sieving domain:

The Number Field Sieve

- Let's recap!
  - Sieving domain: coprime pairs (a, b) in $[-A, A]\times]0, A]$
  - Factor base $\mathcal{B}_i$: prime ideals $\mathfrak{p} = (p, r)$ of $\mathbb{Z}[\alpha_i]$ with $p \leq B_i$
- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$

The Number Field Sieve

- Let's recap!
  - Sieving domain: coprime pairs (a, b) in $[-A, A] \times]0, A]$
  - Factor base $\mathcal{B}_i$: prime ideals $\mathfrak{p} = (p, r)$ of $\mathbb{Z}[\alpha_i]$ with $p \leq B_i$
- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$
  - Check if $N_i(a - b\alpha_i)$ is B_i-smooth (all its prime factors are $\leq B_i$)

The Number Field Sieve

- Let's recap!
  - Sieving domain: coprime pairs (a, b) in $[-A, A] \times]0, A]$
  - Factor base $\mathcal{B}_i$: prime ideals $\mathfrak{p} = (p, r)$ of $\mathbb{Z}[\alpha_i]$ with $p \leq B_i$
- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$
  - Check if $N_i(a - b\alpha_i)$ is B_i-smooth (all its prime factors are $\leq B_i$)
  - If both norms are smooth, then (a, b) is a relation

The Number Field Sieve

- Let's recap!
  - Sieving domain: coprime pairs (a, b) in $[-A, A] \times]0, A]$
  - Factor base $\mathcal{B}_i$: prime ideals $\mathfrak{p} = (p, r)$ of $\mathbb{Z}[\alpha_i]$ with $p \leq B_i$
- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$
  - Check if $N_i(a - b\alpha_i)$ is B_i-smooth (all its prime factors are $\leq B_i$)
  - If both norms are smooth, then (a, b) is a relation
- We need more relations than elements of the factor bases:

$$\#\mathcal{R} > \#\mathcal{B}_1 + \#\mathcal{B}_2$$

The Number Field Sieve

❶ **Relation collection** (a.k.a. sieving): build set of relations $\mathcal{R}$

The Number Field Sieve

❶ **Relation collection** (a.k.a. sieving): build set of relations $\mathcal{R}$

❷ **Linear algebra**: find vector of left-kernel of the matrix over $\mathbb{F}_2$

The Number Field Sieve

❶ **Relation collection** (a.k.a. sieving): build set of relations $\mathcal{R}$

❷ **Linear algebra**: find vector of left-kernel of the matrix over $\mathbb{F}_2$

❸ **Square root**: compute elements $\gamma_1 \in \mathbb{Z}[\alpha_1]$ and $\gamma_2 \in \mathbb{Z}[\alpha_2]$ such that $\gamma_1(m)^2 \equiv \gamma_2(m)^2 \pmod{N}$

The Number Field Sieve

❶ **Polynomial selection**: find suitable polynomials f_1 and f_2

❷ **Relation collection** (a.k.a. sieving): build set of relations $\mathcal{R}$

❸ **Linear algebra**: find vector of left-kernel of the matrix over $\mathbb{F}_2$

❹ **Square root**: compute elements $\gamma_1 \in \mathbb{Z}[\alpha_1]$ and $\gamma_2 \in \mathbb{Z}[\alpha_2]$ such that $\gamma_1(m)^2 \equiv \gamma_2(m)^2 \pmod{N}$

The Number Field Sieve

❶ **Polynomial selection**: find suitable polynomials f_1 and f_2

❷ **Factor base generation**: build factors bases $\mathcal{B}_1$ and $\mathcal{B}_2$

❸ **Relation collection** (a.k.a. sieving): build set of relations $\mathcal{R}$

❹ **Linear algebra**: find vector of left-kernel of the matrix over $\mathbb{F}_2$

❺ **Square root**: compute elements $\gamma_1 \in \mathbb{Z}[\alpha_1]$ and $\gamma_2 \in \mathbb{Z}[\alpha_2]$ such that $\gamma_1(m)^2 \equiv \gamma_2(m)^2 \pmod{N}$

The Number Field Sieve

❶ **Polynomial selection**: find suitable polynomials f_1 and f_2

❷ **Factor base generation**: build factors bases $\mathcal{B}_1$ and $\mathcal{B}_2$

❸ **Relation collection** (a.k.a. sieving): build set of relations $\mathcal{R}$

❹ **Filtering**: build and simplify matrix from relations

❺ **Linear algebra**: find vector of left-kernel of the matrix over $\mathbb{F}_2$

❻ **Square root**: compute elements $\gamma_1 \in \mathbb{Z}[\alpha_1]$ and $\gamma_2 \in \mathbb{Z}[\alpha_2]$ such that $\gamma_1(m)^2 \equiv \gamma_2(m)^2 \pmod{N}$

The Number Field Sieve

❶ **Polynomial selection**: find suitable polynomials f_1 and f_2

❷ **Factor base generation**: build factors bases $\mathcal{B}_1$ and $\mathcal{B}_2$

❸ **Relation collection** (a.k.a. sieving): build set of relations $\mathcal{R}$

❹ **Filtering**: build and simplify matrix from relations

❺ **Linear algebra**: find vector of left-kernel of the matrix over $\mathbb{F}_2$

❻ **Characters**: deal with number-field-related technicalities (e.g., units)

❼ **Square root**: compute elements $\gamma_1 \in \mathbb{Z}[\alpha_1]$ and $\gamma_2 \in \mathbb{Z}[\alpha_2]$ such that $\gamma_1(m)^2 \equiv \gamma_2(m)^2 \pmod{N}$

The Number Field Sieve

1. **Polynomial selection**: find suitable polynomials f_1 and f_2
2. **Factor base generation**: build factors bases $\mathcal{B}_1$ and $\mathcal{B}_2$
3. **Relation collection** (a.k.a. sieving): build set of relations $\mathcal{R}$
4. **Filtering**: build and simplify matrix from relations
5. **Linear algebra**: find vector of left-kernel of the matrix over $\mathbb{F}_2$
6. **Characters**: deal with number-field-related technicalities (e.g., units)
7. **Square root**: compute elements $\gamma_1 \in \mathbb{Z}[\alpha_1]$ and $\gamma_2 \in \mathbb{Z}[\alpha_2]$ such that $\gamma_1(m)^2 \equiv \gamma_2(m)^2 \pmod{N}$
8. **Profit!**

Back to CADO-NFS

- Each step is handled by a specific binary/script

❶ **Polynomial selection**	`polyselect/polyselect2l`
❷ **Factor base generation**	`sieve/makefb`
❸ **Relation collection**	`sieve/{freerel,las}`
❹ **Filtering**	`filter/{dup1,dup2,purge,merge,replay}`
❺ **Linear algebra**	`linalg/bwc/bwc.pl`
❻ **Characters**	`linalg/characters`
❼ **Square root**	`sqrt/sqrt`

Back to CADO-NFS

- Each step is handled by a specific binary/script
- `cadofactor.py`: Python script to run whole factorization
  → All NFS parameters in a single parameter file

Python script + parameter file
`scripts/cadofactor/cadofactor.py`

❶ **Polynomial selection**	`polyselect/polyselect2l`
❷ **Factor base generation**	`sieve/makefb`
❸ **Relation collection**	`sieve/{freerel,las}`
❹ **Filtering**	`filter/{dup1,dup2,purge,merge,replay}`
❺ **Linear algebra**	`linalg/bwc/bwc.pl`
❻ **Characters**	`linalg/characters`
❼ **Square root**	`sqrt/sqrt`

Back to CADO-NFS

- Each step is handled by a specific binary/script
- `cadofactor.py`: Python script to run whole factorization
  $\rightarrow$ All NFS parameters in a single parameter file
- `factor.sh`: Bash script for simple factorizations

❶ **Polynomial selection**	`polyselect/polyselect2l`
❷ **Factor base generation**	`sieve/makefb`
❸ **Relation collection**	`sieve/{freerel,las}`
❹ **Filtering**	`filter/{dup1,dup2,purge,merge,replay}`
❺ **Linear algebra**	`linalg/bwc/bwc.pl`
❻ **Characters**	`linalg/characters`
❼ **Square root**	`sqrt/sqrt`

Bash script `factor.sh`

Python script + parameter file `scripts/cadofactor/cadofactor.py`

Let's play!

- Requirements:
  - GNU/Linux (or Mac OS X + Xcode)
  - GCC 4.4 or later
  - GMP 5 or later
  - GNU Make and CMake 2.6.3 or later
  - Python 3.2 or later
  - SQLite 3, including Python bindings
  - GNU Wget or cURL
  - GNU Gzip
  - GNU Bash

Let's play!

- Go and download CADO-NFS 2.1.1 from

  `http://cado-nfs.gforge.inria.fr/`

Let's play!

- Go and download CADO-NFS 2.1.1 from

  http://cado-nfs.gforge.inria.fr/

- Un-tar:

```
$ tar xzvf cado-nfs-2.1.1.tar.gz
$ cd cado-nfs-2.1.1
```

Let's play!

- Go and download CADO-NFS 2.1.1 from

  http://cado-nfs.gforge.inria.fr/

- Un-tar:

```
$ tar xzvf cado-nfs-2.1.1.tar.gz
$ cd cado-nfs-2.1.1
```

- Optional: tweak build configuration (esp. for Mac OS X):

```
$ cp local.sh.example local.sh
$ vi local.sh
```

Let's play!

- Go and download CADO-NFS 2.1.1 from

  `http://cado-nfs.gforge.inria.fr/`

- Un-tar:

```
$ tar xzvf cado-nfs-2.1.1.tar.gz
$ cd cado-nfs-2.1.1
```

- Optional: tweak build configuration (esp. for Mac OS X):

```
$ cp local.sh.example local.sh
$ vi local.sh
```

- Build:

```
$ make
```

A toy factorization

- Let's factor this 59-digit composite integer:

  $$c_{59} = 90377629292003121684002147101760858109247336549001090677693$$

  (you can just copy-paste it from `http://www.loria.fr/~detreyje/cado-nfs.txt`)

A toy factorization

- Let's factor this 59-digit composite integer:

  $c_{59} = 90377629292003121684002147101760858109247336549001090677693$

  (you can just copy-paste it from http://www.loria.fr/~detreyje/cado-nfs.txt)

- Run:

```
$ export CADO_DEBUG=1
$ mkdir /tmp/c59
$ t=/tmp/c59 ./factor.sh 903...693 -t 2
```

A toy factorization

- Let's factor this 59-digit composite integer:

  $$c_{59} = 90377629292003121684002147101760858109247336549001090677693$$

  (you can just copy-paste it from
  http://www.loria.fr/~detreyje/cado-nfs.txt)

- Run:

```
$ export CADO_DEBUG=1
$ mkdir /tmp/c59
$ t=/tmp/c59 ./factor.sh 903...693 -t 2
```

- Get factors!

```
...
Info:Complete Factorization: ...
588120598053661 260938498861057
760926063870977 773951836515617
OK
```

Diving into details – Polynomial selection

- Find polynomials f_1 and $f_2 \in \mathbb{Z}[X]$ such that
  - f_1 and f_2 are irreducible and coprime over $\mathbb{Q}$
  - they have a common root $m \in \mathbb{Z}/N\mathbb{Z}$:

$$f_1(m) \equiv 0 \pmod{N} \quad \text{and} \quad f_2(m) \equiv 0 \pmod{N}$$

Diving into details – Polynomial selection

- Find polynomials f_1 and $f_2 \in \mathbb{Z}[X]$ such that
  - f_1 and f_2 are irreducible and coprime over $\mathbb{Q}$
  - they have a common root $m \in \mathbb{Z}/N\mathbb{Z}$:

$$f_1(m) \equiv 0 \pmod{N} \quad \text{and} \quad f_2(m) \equiv 0 \pmod{N}$$

- In practice:
  - Take a linear polynomial for f_2: this is called the "rational side"
  - Take a degree-d polynomial for f_1, with $d \in \{4, 5, 6\}$: this is called the "algebraic side"

$$f_1(X) = f_{1,d}X^d + f_{1,d-1}X^{d-1} + \cdots + f_{1,1}X + f_{1,0}$$

Diving into details – Polynomial selection

- Find polynomials f_1 and $f_2 \in \mathbb{Z}[X]$ such that
  - f_1 and f_2 are irreducible and coprime over $\mathbb{Q}$
  - they have a common root $m \in \mathbb{Z}/N\mathbb{Z}$:

$$f_1(m) \equiv 0 \pmod{N} \quad \text{and} \quad f_2(m) \equiv 0 \pmod{N}$$

- In practice:
  - Take a linear polynomial for f_2: this is called the "rational side"
  - Take a degree-d polynomial for f_1, with $d \in \{4, 5, 6\}$: this is called the "algebraic side"

$$f_1(X) = f_{1,d}X^d + f_{1,d-1}X^{d-1} + \cdots + f_{1,1}X + f_{1,0}$$

- Look for a polynomial f_1 of degree d:
  - such that norms $N_1(a - b\alpha_1) = f_1(a/b)b^d$ are as small as possible for pairs (a, b) in the sieving domain
  - which has many roots modulo small primes

Diving into details – Polynomial selection

- Two main steps:
  - Size optimization: find polynomials with small norm
  - Root optimization: translate/rotate candidates so that they have many roots modulo small primes

Diving into details – Polynomial selection

- Two main steps:
  - Size optimization: find polynomials with small norm
  - Root optimization: translate/rotate candidates so that they have many roots modulo small primes
- CADO-NFS parameters (`tasks.polyselect.*`):
  - `degree`: degree d of polynomial f_1
  - `admin` (0): minimum value for leading coefficient $f_{1,d}$
  - `admax`: maximum value for leading coefficient $f_{1,d}$
  - `incr` (60): force $f_{1,d}$ to be a multiple of this smooth number
  - `nrkeep`: how many candidates to keep after first step

Diving into details – Polynomial selection

- Two main steps:
  - Size optimization: find polynomials with small norm
  - Root optimization: translate/rotate candidates so that they have many roots modulo small primes
- CADO-NFS parameters (`tasks.polyselect.*`):
  - `degree`: degree d of polynomial f_1
  - `admin` (0): minimum value for leading coefficient $f_{1,d}$
  - `admax`: maximum value for leading coefficient $f_{1,d}$
  - `incr` (60): force $f_{1,d}$ to be a multiple of this smooth number
  - `nrkeep`: how many candidates to keep after first step
  - `adrange`: split search interval for $f_{1,d}$ into ranges of this size $\rightarrow$ easy parallelization

Diving into details – Polynomial selection

- Two main steps:
  - Size optimization: find polynomials with small norm
  - Root optimization: translate/rotate candidates so that they have many roots modulo small primes
- CADO-NFS parameters (`tasks.polyselect.*`):
  - `degree`: degree d of polynomial f_1
  - `admin` (0): minimum value for leading coefficient $f_{1,d}$
  - `admax`: maximum value for leading coefficient $f_{1,d}$
  - `incr` (60): force $f_{1,d}$ to be a multiple of this smooth number
  - `nrkeep`: how many candidates to keep after first step
  - `adrange`: split search interval for $f_{1,d}$ into ranges of this size $\rightarrow$ easy parallelization
- Best polynomial stored in:

$\langle name \rangle$`.polyselect2.poly`

Diving into details – Relation collection

- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$
  - Check if $N_i(a - b\alpha_i)$ is B_i-smooth (all its prime factors are $\leq B_i$)
  - If both norms are smooth, then (a, b) is a relation

Diving into details – Relation collection

- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$
  - Check if $N_i(a - b\alpha_i)$ is B_i-smooth (all its prime factors are $\leq B_i$)
  - If both norms are smooth, then (a, b) is a relation

- Special-$\mathfrak{q}$ sieving:

Diving into details – Relation collection

- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$
  - Check if $N_i(a - b\alpha_i)$ is B_i-smooth (all its prime factors are $\leq B_i$)
  - If both norms are smooth, then (a, b) is a relation
- Special-$\mathfrak{q}$ sieving:
  - Fix a prime ideal $\mathfrak{q} = (q, \rho)$ of $\mathbb{Z}[\alpha_1]$

Diving into details – Relation collection

- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$
  - Check if $N_i(a - b\alpha_i)$ is B_i-smooth (all its prime factors are $\leq B_i$)
  - If both norms are smooth, then (a, b) is a relation

- Special-$\mathfrak{q}$ sieving:
  - Fix a prime ideal $\mathfrak{q} = (q, \rho)$ of $\mathbb{Z}[\alpha_1]$
  - The set of (a, b) pairs such that $\mathfrak{q}$ divides $\langle a - b\alpha_i \rangle$ is a Euclidean lattice of $\mathbb{Z}^2$

Diving into details – Relation collection

- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$
  - Check if $N_i(a - b\alpha_i)$ is B_i-smooth (all its prime factors are $\leq B_i$)
  - If both norms are smooth, then (a, b) is a relation
- Special-$\mathfrak{q}$ sieving:
  - Fix a prime ideal $\mathfrak{q} = (q, \rho)$ of $\mathbb{Z}[\alpha_1]$
  - The set of (a, b) pairs such that $\mathfrak{q}$ divides $\langle a - b\alpha_i \rangle$ is a Euclidean lattice of $\mathbb{Z}^2$
  - Compute basis $(\mathbf{u}, \mathbf{v})$ of this lattice

Diving into details – Relation collection

- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$
  - Check if $N_i(a - b\alpha_i)$ is B_i-smooth (all its prime factors are $\leq B_i$)
  - If both norms are smooth, then (a, b) is a relation
- Special-$\mathfrak{q}$ sieving:
  - Fix a prime ideal $\mathfrak{q} = (q, \rho)$ of $\mathbb{Z}[\alpha_1]$
  - The set of (a, b) pairs such that $\mathfrak{q}$ divides $\langle a - b\alpha_i \rangle$ is a Euclidean lattice of $\mathbb{Z}^2$
  - Compute basis $(\mathbf{u}, \mathbf{v})$ of this lattice
  - Enumerate lattice elements as pairs $(a, b) = i\mathbf{u} + j\mathbf{v}$ with $(i, j) \in [-I, I] \times]0, I]$

Diving into details – Relation collection

- For each (a, b) pair in the sieving domain:
  - Compute the norms $N_i(a - b\alpha_i) = f_i(a/b)b^i$
  - Check if $N_i(a - b\alpha_i)$ is B_i-smooth (all its prime factors are $\leq B_i$)
  - If both norms are smooth, then (a, b) is a relation

- Special-$\mathfrak{q}$ sieving:
  - Fix a prime ideal $\mathfrak{q} = (q, \rho)$ of $\mathbb{Z}[\alpha_1]$
  - The set of (a, b) pairs such that $\mathfrak{q}$ divides $\langle a - b\alpha_i \rangle$ is a Euclidean lattice of $\mathbb{Z}^2$
  - Compute basis $(\mathbf{u}, \mathbf{v})$ of this lattice
  - Enumerate lattice elements as pairs $(a, b) = i\mathbf{u} + j\mathbf{v}$ with $(i, j) \in [-I, I] \times]0, I]$
  - One independent subtask for each special-$\mathfrak{q}$ $\rightarrow$ easy parallelization

Diving into details – Relation collection

- Example from c_{59}:

Diving into details – Relation collection

- Example from c_{59}:
  - Polynomials:

$$\begin{aligned} f_1(X) = {} & 60 \cdot X^4 + 164823 \cdot X^3 + 2561101187 \cdot X^2 \\ & - 4872316534587 \cdot X - 9288039622841198 \\ f_2(X) = {} & 4827001309 \cdot X - 19261601140604 1 \end{aligned}$$

Diving into details – Relation collection

- Example from c_{59}:
  - Polynomials:

$$\begin{aligned} f_1(X) = &\quad 60 \cdot X^4 + 164823 \cdot X^3 + 2561101187 \cdot X^2 \\ &- 4872316534587 \cdot X - 9288039622841198 \\ f_2(X) = &\quad 4827001309 \cdot X - 19261601140604 1 \end{aligned}$$

  - Special-$\mathfrak{q}$: $(q, \rho) = (200003, 74941)$

Diving into details – Relation collection

- Example from c_{59}:
  - Polynomials:

$$\begin{aligned} f_1(X) = {} & \quad 60 \cdot X^4 + 164823 \cdot X^3 + 2561101187 \cdot X^2 \\ & - 4872316534587 \cdot X - 9288039622841198 \\ f_2(X) = {} & \quad 4827001309 \cdot X - 192616011406041 \end{aligned}$$

  - Special-$\mathfrak{q}$: $(q, \rho) = (200003, 74941)$
  - Sieving position: $(a, b) = (-876877, 31)$

Diving into details – Relation collection

- Example from c_{59}:
  - Polynomials:

$$\begin{aligned} f_1(X) = {} & \quad 60 \cdot X^4 + 164823 \cdot X^3 + 2561101187 \cdot X^2 \\ & - 4872316534587 \cdot X - 9288039622841198 \\ f_2(X) = {} & \quad 4827001309 \cdot X - 19261601140604 1 \end{aligned}$$

  - Special-$\mathfrak{q}$: $(q, \rho) = (200003, 74941)$
  - Sieving position: $(a, b) = (-876877, 31)$

- Is (a, b) a relation? Factor its norms

Diving into details – Relation collection

- Example from c_{59}:
  - Polynomials:

$$\begin{aligned} f_1(X) = {} & \quad 60 \cdot X^4 + 164823 \cdot X^3 + 2561101187 \cdot X^2 \\ & - 4872316534587 \cdot X - 9288039622841198 \\ f_2(X) = {} & \quad 4827001309 \cdot X - 19261601140604 1 \end{aligned}$$

  - Special-$\mathfrak{q}$: $(q, \rho) = (200003, 74941)$
  - Sieving position: $(a, b) = (-876877, 31)$

- Is (a, b) a relation? Factor its norms

$$\begin{aligned} N_1(a - b\alpha_1) &= 340397725772199663711302 85 \\ N_2(a - b\alpha_2) &= -1020378278041926 4 \end{aligned}$$

Diving into details – Relation collection

- Example from c_{59}:
  - Polynomials:

$$\begin{aligned} f_1(X) = {} & \quad 60 \cdot X^4 + 164823 \cdot X^3 + 2561101187 \cdot X^2 \\ & - 4872316534587 \cdot X - 9288039622841198 \\ f_2(X) = {} & \quad 4827001309 \cdot X - 19261601140604 1 \end{aligned}$$

  - Special-$\mathfrak{q}$: $(q, \rho) = (200003, 74941)$
  - Sieving position: $(a, b) = (-876877, 31)$

- Is (a, b) a relation? Factor its norms

$$\begin{aligned} N_1(a - b\alpha_1) &= \quad 170196309941450710095 \cdot q \\ N_2(a - b\alpha_2) &= -1020378278041926 4 \end{aligned}$$

Diving into details – Relation collection

- Example from c_{59}:
  - Polynomials:

$$\begin{aligned} f_1(X) = {} & 60 \cdot X^4 + 164823 \cdot X^3 + 2561101187 \cdot X^2 \\ & - 4872316534587 \cdot X - 9288039622841198 \\ f_2(X) = {} & 4827001309 \cdot X - 19261601140604 1 \end{aligned}$$

  - Special-$\mathfrak{q}$: $(q, \rho) = (200003, 74941)$
  - Sieving position: $(a, b) = (-876877, 31)$
- Is (a, b) a relation? Factor its norms
  - Remove small factors by sieving techniques (up to bound B_i')

$$\begin{aligned} N_1(a - b\alpha_1) &= 17019630994145071009 5 \cdot q \\ N_2(a - b\alpha_2) &= -1020378278041926 4 \end{aligned}$$

Diving into details – Relation collection

- Example from c_{59}:
  - Polynomials:

$$\begin{aligned} f_1(X) = &\quad 60 \cdot X^4 + 164823 \cdot X^3 + 2561101187 \cdot X^2 \\ &- 4872316534587 \cdot X - 9288039622841198 \\ f_2(X) = &\quad 4827001309 \cdot X - 19261601140604 1 \end{aligned}$$

  - Special-$\mathfrak{q}$: $(q, \rho) = (200003, 74941)$
  - Sieving position: $(a, b) = (-876877, 31)$

- Is (a, b) a relation? Factor its norms
  - Remove small factors by sieving techniques (up to bound B_i')

$$\begin{aligned} N_1(a - b\alpha_1) &= \quad 3^2 \cdot 5 \cdot 43 \cdot 53 \cdot 59 \cdot 61 \cdot 151 \cdot 3053757221 \cdot q \\ N_2(a - b\alpha_2) &= -2^6 \cdot 67 \cdot 311 \cdot 617 \cdot 709 \cdot 17491 \end{aligned}$$

Diving into details – Relation collection

- Example from c_{59}:
  - Polynomials:

$$\begin{aligned} f_1(X) = {} & 60 \cdot X^4 + 164823 \cdot X^3 + 2561101187 \cdot X^2 \\ & - 4872316534587 \cdot X - 9288039622841198 \\ f_2(X) = {} & 4827001309 \cdot X - 19261601140604 1 \end{aligned}$$

  - Special-$\mathfrak{q}$: $(q, \rho) = (200003, 74941)$
  - Sieving position: $(a, b) = (-876877, 31)$

- Is (a, b) a relation? Factor its norms
  - Remove small factors by sieving techniques (up to bound B'_i)
  - Co-factor remaining parts only if not too large

$$\begin{aligned} N_1(a - b\alpha_1) &= 3^2 \cdot 5 \cdot 43 \cdot 53 \cdot 59 \cdot 61 \cdot 151 \cdot 3053757221 \cdot q \\ N_2(a - b\alpha_2) &= -2^6 \cdot 67 \cdot 311 \cdot 617 \cdot 709 \cdot 17491 \end{aligned}$$

Diving into details – Relation collection

- Example from c_{59}:
  - Polynomials:

$$\begin{aligned} f_1(X) = &\quad 60 \cdot X^4 + 164823 \cdot X^3 + 2561101187 \cdot X^2 \\ &- 4872316534587 \cdot X - 9288039622841198 \\ f_2(X) = &\quad 4827001309 \cdot X - 19261601140604 1 \end{aligned}$$

  - Special-$\mathfrak{q}$: $(q, \rho) = (200003, 74941)$
  - Sieving position: $(a, b) = (-876877, 31)$

- Is (a, b) a relation? Factor its norms
  - Remove small factors by sieving techniques (up to bound B_i')
  - Co-factor remaining parts only if not too large

$$\begin{aligned} N_1(a - b\alpha_1) &= \quad 3^2 \cdot 5 \cdot 43 \cdot 53 \cdot 59 \cdot 61 \cdot 151 \cdot 22447 \cdot 136043 \cdot q \\ N_2(a - b\alpha_2) &= -2^6 \cdot 67 \cdot 311 \cdot 617 \cdot 709 \cdot 17491 \end{aligned}$$

Diving into details – Relation collection

- General parameters (`tasks.*`)
  - `alim` / `rlim`: the maximum norm of sieved primes (B_i')
  - `lpba` / `lpbr`: the so-called large prime bound, in bits ($\log_2 B_i$)
  - `I`: bounds on sieving domain

Diving into details – Relation collection

- General parameters (`tasks.*`)
  - `alim` / `rlim`: the maximum norm of sieved primes (B_i')
  - `lpba` / `lpbr`: the so-called large prime bound, in bits ($\log_2 B_i$)
  - `I`: bounds on sieving domain
- Sieving parameters (`tasks.sieve.*`)
  - `mfba` / `mfbr`: co-factorization threshold, in bits
  - `qmin`: first special-$\mathfrak{q}$ to sieve
  - `rels_wanted`: number of relations to collect

Diving into details – Relation collection

- General parameters (`tasks.*`)
  - `alim` / `rlim`: the maximum norm of sieved primes (B'_i)
  - `lpba` / `lpbr`: the so-called large prime bound, in bits ($\log_2 B_i$)
  - `I`: bounds on sieving domain
- Sieving parameters (`tasks.sieve.*`)
  - `mfba` / `mfbr`: co-factorization threshold, in bits
  - `qmin`: first special-$\mathfrak{q}$ to sieve
  - `rels_wanted`: number of relations to collect
  - `qrange`: number of special-$\mathfrak{q}$'s to sieve per subtask

Thank you for your attention

Happy factoring!