We Are on the Same Side Alternative Sieving Strategies for the Number Field Sieve

Charles Bouillaguet, Ambroise Fleury, Pierre-Alain Fouque, and Paul Kirchner

CB : Sorbonne Université, CNRS, LIP6, Paris
AF : Université Paris-Saclay, CEA, List, Palaiseau
PAF, PK : Univ Rennes, CNRS, IRISA

December 6, 2023 - ASIACRYPT 2023

Factorization

- RSA Cryptosystem
- Factoring a large number

Number Field Sieve (NFS)

- CADO-NFS
- Relations
- Relations in the NFS

Our contribution

- Batch factoring
- Hybrid version
- Implementation

RSA Cryptosystem

Private key

- Used for decryption
- Generated from two random prime numbers p and q

Public key

- Used for encryption
- Product $N = pq$

Factorization

- RSA security is linked to the hardness of integer factorization
- Finding p and q from N breaks RSA

Generic factorization method

Finding a square

- $x^2 = y^2 \mod N$
- $x \neq y \mod N$

Then...

- $N = x^2 - y^2 \mod N$
- $N = (x + y)(x - y) \mod N$
- $gcd(x \pm y, N)$ gives a factor of N

Finding a congruence of squares?

Dixon's factorization method

Build a square

- Generate many y_i such that
  - $y_i = x_i^2 \mod N$
  - y_i is smooth (=only small divisors)
  - it is called a *relation*
- Build $Y^2 \mod N$ as a product of y_i's

1. Relation collection

- Generate many y_i
- Find many relations

2. Linear algebra

- Combine the relations
- $Y^2 = X^2 \mod N$

From factoring a large number...
...to factoring many small numbers

Relations

What relations look like

factor base	**2**	**3**	**5**	**7**	**11**	**13**	**17**
6468	2^2	3		7^2	11		
10210200	2^3	3	5^2	7	11	13	17
1449175			5^2	7^3		13^2	
79560	2^3	3^2	5			13	17
4004	2^2			7	11	13	
175032	2^3	3^2			11	13	17

Next step is to combine them into a square
How? Combine lines to get even exponents

CADO-NFS

- Implementation of the NFS
- Open source : https://gitlab.inria.fr/cado-nfs/cado-nfs
- Can also compute discrete logarithms
- 2019 : Factorization record RSA-240 (240 digits)
- 2020 : Factorization record RSA-250 (current record)
- Computing time is dominated by relation collection

	relation collection	linear alebra
RSA-240	800 CPU years	83 CPU years
RSA-250	2450 CPU years	250 CPU years

Relations in the NFS

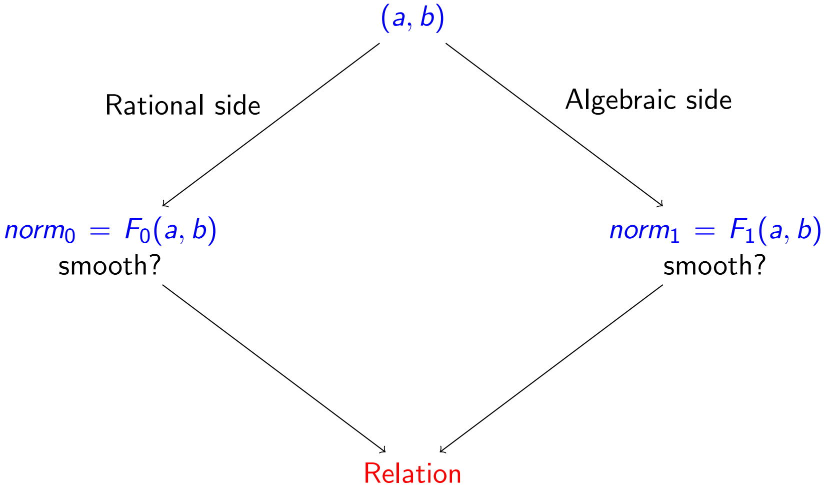
Two sides

- Pairs (a, b) of coprime and "small" integers
- Two polynomials $F_i(a, b) = f_i(a/b)b^d$
- We call norms the evaluation of a polynomial with a pair (a, b)
  - $norm_0 = F_0(a, b)$
  - $norm_1 = F_1(a, b)$

Chosen f polynomials for RSA-250 record

$$
\begin{aligned}
f_0 &= 185112968818638292881913X \\
&\quad - 3256571715934047438664355774734330386901 \\
f_1 &= 86130508464000X^6 \\
&\quad - 81583513076429048837733781438376984122961112000 \\
&\quad - 66689953322631501408X^5 \\
&\quad - 1721614429538740120011760034829385792019395X \\
&\quad - 52733221034966333966198X^4 \\
&\quad - 3113627253613202265126907420550648326X^2 \\
&\quad + 46262124564021437136744523465879X^3
\end{aligned}
$$

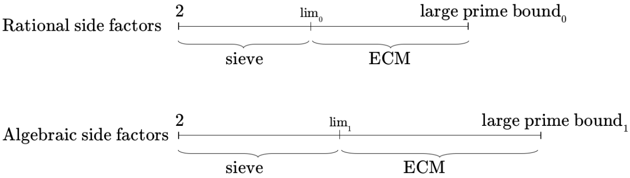
Relation collection



Factoring norms

2 methods :

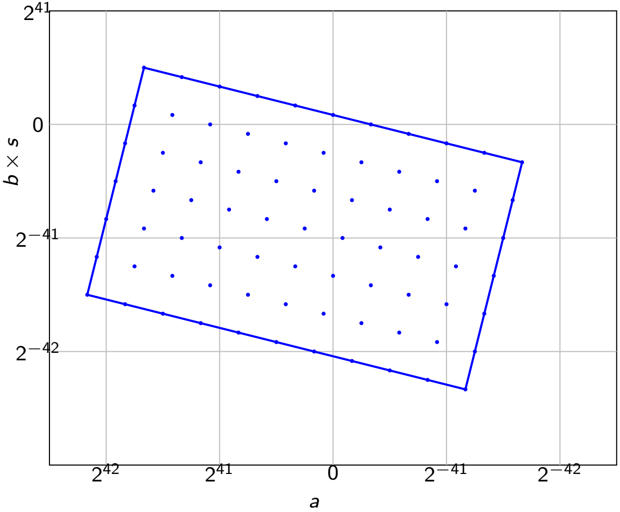
- Sieving to find small and medium factors
- Elliptic-curve factorization (ECM) to find large factors



- Step 1 : sieve all norms
- Step 2 : ECM on norms most likely to become relations

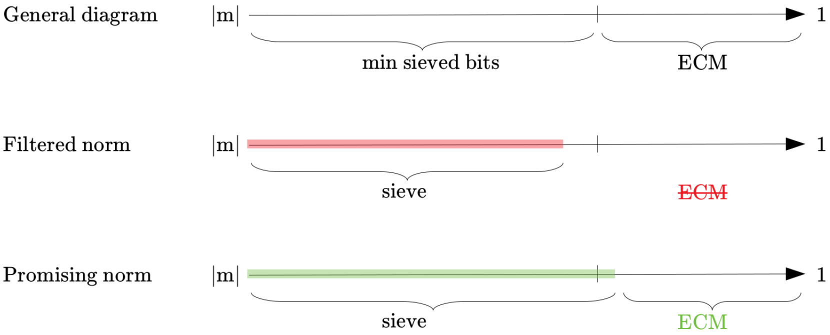
Sieving process

On each side, sieving (a, b) pairs allows to find multiples of each prime p from the factor base



Promising pairs

- Best candidates to give a relation
- Sieving factored enough for both norms
- Only promising pairs get to the ECM step



Promising bound

If the bound deciding wether or not a pair is sent to ECM is...

- Too high
  - Many pairs of low quality will take too much time in ECM
- Too low
  - Few pairs of high quality will give too few relations and additional sieving will be needed

Improving relation collection in CADO-NFS

Goal : almost as many promising pairs at a much lower cost

Small sieve

Subroutine of CADO-NFS sieving finding small primes

- Small factors are worth few bits
- Not decisive on promising pairs

Remove small sieve?

Batch factoring

How to find smooth parts of integers [Bernstein 2004]

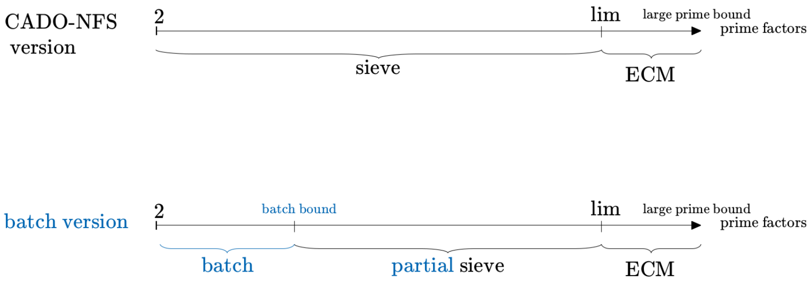
- Input : list of integers, factor base (b bits)
- Output : list of smooth parts, meaning the product of factors from the base found in each integer
- $O(b(\lg b)^{2+o(1)})$

Hybrid version

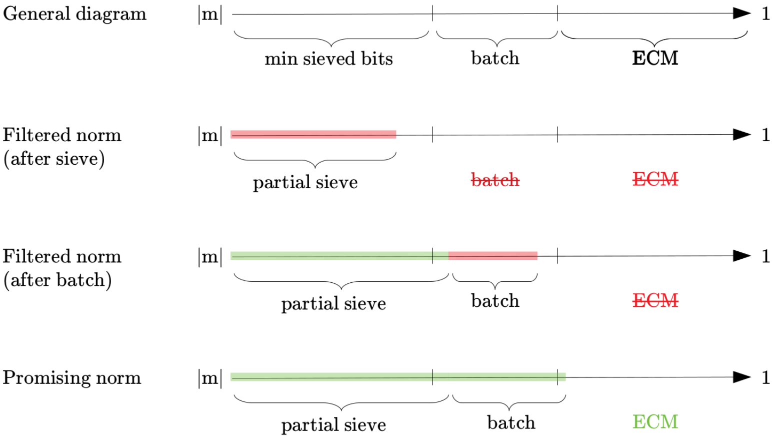
Pick an intermediate "batch promising" bound larger than the "ECM promising" bound, then :

1. Sieve only on medium primes
2. Remove non-promising pairs
3. Get small factors using batch factoring
4. Remove non-promising pairs
5. Get large factors using ECM
6. Relations!

Method for each prime factors interval



Path to ECM



General diagram
|m|
1
min sieved bits
batch
ECM
Filtered norm
(after sieve)
|m|
1
partial sieve
batch
ECM
Filtered norm
(after batch)
|m|
1
partial sieve
batch
ECM
Promising norm
|m|
1
partial sieve
batch
ECM

Implementation in CADO-NFS

RSA-250's relations

- Data to target a specific number of relations
- Allow us to pick parameters
- Benchmark baseline

Benchmarks

- Sampled sieved regions
- Easy extrapolation

Results

Results for a few example sieving areas picked randomly

Example A, with batch bounds 89 bits and 137 bits

Version	# relations	ratio	Time (s)	ratio	local speed-up
Original	390	-	8619	-	-
Hybrid	347	0.89	6940	0.81	1.10

Example B, with batch bounds 117 bits and 167 bits

Version	# relations	ratio	Time (s)	ratio	local speed-up
Original	674	-	6942	-	-
Hybrid	606	0.90	5684	0.82	1.10

Results

- Fewer relations are found
- Speedup counteracts this
- Better efficiency