Twisted Towers: Fully Costed Special Tower NFS Attacks on BLS12-381 and BN254

Gautam Botrel
gautam.botrel@zclair.io

November 2026

Abstract

BLS12-381 is commonly said to offer "126 bits" of security and BN254 "about 100 bits". Both figures are estimates of the special tower number field sieve (STNFS) in the target group $\mathbb{F}_{p^{12}}^*$, and both count abstract operations in models that leave out whole phases of the attack. We replace them by complete ledgers in which every step is priced in machine instructions: relation collection, cofactorisation, filtering, linear algebra, both individual logarithms and all failures. Under explicitly listed heuristics, a complete attack executes $2^{127.5}$ instructions for BLS12-381 and $2^{103.2}$ for the BN254 curve used in Ethereum, with success probability above 0.998. For BLS12-381 this is 9.6 bits below Pollard's rho on the curve counted the same way, where operation counts suggested one bit: the target group is clearly the weaker side. It is also 9.8 bits below what the published design costs when it is completed with standard components and charged in full ($2^{137.3}$ instructions, level with rho). For BN254 our figure confirms the accepted level and gives it a concrete meaning.

The method rests on three ideas and on a systematic reduction of the work per sieved cell. First, a *twisted tower*: a unit c of the sextic base field with $c\sigma(c) = -1$ turns the usual sign symmetry into a symmetry of order four that acts as a Frobenius power, so that relations and ideals come in orbits of four and the linear system has Gaussian-integer entries. Second, *counting each relation once*: coefficient-cube estimates count unit multiples of a relation as distinct; for the published BLS12-381 design only $2^{-2.8}$ of the counted relations are distinct, and we sieve a fundamental domain instead. Third, a sieve organised around *reuse*: only the harder side is sieved, tiny primes are handled by batch arithmetic, and the hits of a prime ideal are replayed from lists shared by all slices. A product-form linear algebra and a planned descent complete the method. Each technique is stated for a general special pair of tower shape $(6, 2)$ and then instantiated on the two curves. Nothing was run at full size; the final designs assume up to 2^{112} bytes of storage, which the cost model does not charge; and the practical security of both curves is unchanged.

1 Introduction

1.1 The question

BLS12-381 [10] and BN254 [7, 11] are the two most widely deployed pairing-friendly curves; they secure signatures and zero-knowledge proofs in most blockchain systems. Their security level is usually quoted as "about 126 bits" and "about 100 bits" [4, 17, 33]. These numbers come from estimates of the number field sieve in the target group, and they count *operations* in a simplified model: one operation per sieved element, a fixed constant per matrix entry, no cofactorisation, no individual logarithm, no failures.

This paper asks a more concrete question:

> *How many machine instructions does a complete attack execute, from the public parameters to a verified discrete logarithm, when every step is charged?*

We answer it with a *ledger*: an itemised bill in which each row is a count multiplied by a measured price, together with the list of hypotheses that connect the measurements to a full-size run. The method is general: it applies to any pairing-friendly curve of embedding degree 12 whose prime is given by a polynomial, and most of it to any tower NFS computation. We develop it in general terms and instantiate it on BLS12-381 and BN254.

1.2 The answer

BLS12-381. A complete attack on the target group executes $2^{127.5}$ instructions. The folklore "126 bits" survives as a number, but not as it was derived: the published design does not collect enough distinct relations, and completed with standard components it costs $2^{137.3}$ instructions, no less than Pollard's rho on the curve ($2^{137.2}$). The techniques of this paper remove 9.8 bits. Counted in the same currency, the target group is 9.6 bits weaker than the curve, not one.

BN254. A complete attack executes $2^{103.2}$ instructions. This confirms the accepted "about 100 bits" for the Ethereum curve, for which no worked estimate existed, and says what the bits are: instructions, with every phase included.

Table 1: Headline results. Cost: expected work of a complete run, which succeeds with probability above 0.998. A "unit" is 20,000 instructions, about one multiplication in $\mathbb{F}_{p^{12}}$. Baseline: the published design with standard components in the same model (Section 3.4). Storage is reported but not charged.

	Cost		Storage	Baseline
Curve	$\log_2$ units	$\log_2$ instr.	$\log_2$ bytes	$\log_2$ instr.
BLS12-381	113.23	127.5	< 112	137.3
BN254	88.95	103.2	< 100	–

Three remarks put Table 1 in context; Section 3 develops them.

- *These are not the literature's numbers in different units.* The published 2^{126} for BLS12-381 is an operation count for a design that, as published, does not find enough distinct relations (Section 5). No instruction-level figure existed. The baseline column is what the published approach costs when every phase is implemented in the obvious way and charged.
- *Instructions are not "bits of security".* In the same metric, Pollard's rho on the curve costs $2^{137.2}$ instructions. The meaningful statements are relative: the sieve is about 9.6 bits cheaper than rho for BLS12-381. For BN254 the gap is about 31 bits; that the target group of BN254 is far weaker than its curve has been known since [4, 21], and our figure agrees with the published level of about 100 bits (Section 3.4).
- *Nothing here is practical.* 2^{103} instructions with 2^{100} bytes of storage is far beyond any existing computer. The contribution is a method, better sieve and solver engineering, and a careful accounting; it is not a break.

1.3 Contributions

The paper is organised around a small number of ideas, each simple to state. Table 9 gives the measured effect of each.

1. **The twisted tower** (Section 4). The special tower NFS has an obvious symmetry, $x \mapsto -x$. We show that with the right sextic base field one can take its *square root*, so that relations and prime ideals come in groups of four: half as many relations are needed and the linear system has half the dimension. This is a new construction of an automorphism of order

four for special pairs of tower shape $(6, 2)$, where the known constructions do not apply: the Frobenius twist $\tau : (y, x) \mapsto (\sigma y, cx)$ (Proposition 1). We give explicit towers for BLS12-381 and BN254 and derive the quotient linear system: the virtual-logarithm identity, the exceptional orbits, Gaussian-integer entries, and a minimal set of 10 (respectively 7) Schirokauer columns (Proposition 2).

2. **The unit-duplicate correction** (Section 5). If (a, b) gives a relation and ε is a unit of the base field, $(\varepsilon a, \varepsilon b)$ gives the *same* relation. A lemma and a measurement showing that coefficient-cube yield estimates overcount relations, by a factor of seven for the published BLS12-381 design, and a fundamental-domain sieve that avoids it.

3. **New sieve algorithms** (Section 6): the one-sided sieve with exact batch tests; the two-stage detector; the coset sieve at a ramified prime with exact local pricing; the record walk along runs of the fundamental domain; and coset replay lists, which replace lattice enumeration by table lookups for most primes. Together they bring the work per sieved cell to 13 to 17 instructions.

4. **A product-form block Wiedemann solver** for Gaussian-integer systems, with subset-sum tables for dense columns and an explicit statement of the finite-length conditions under which it succeeds (Section 7).

5. **The first complete instruction-level ledgers** for the target groups of BLS12-381 and BN254 (Section 9), with their hypotheses (Table 10), the measured contribution of each technique, and an account of why neighbouring designs (larger symmetry groups, other tower shapes, special-q collection) do not do better.

What this paper does not claim. We do not claim a practical attack, a proof of the costs, or a revised security level in the sense of standards documents. No step was run at full size, and no end-to-end computation with these towers has been carried out, even at toy size (Section 10). Every number is either an instruction count measured on a kernel at reduced scale, an exact count, or the output of a model whose hypotheses are listed in Table 10.

Origin. The ledgers were developed and stress-tested in the CurveSmash challenge [23], whose cost model we adopt; Appendix B relates the figures of this paper to the public packages.

1.4 Related work

Estimates for pairing-friendly curves. After the extended and special tower sieves [6, 21, 22], key sizes were re-estimated by Menezes, Sarkar and Singh [28], Barbulescu and Duquesne [4], and Guillevic and Singh [16, 17, 19], whose simulation code [18] is the reference for current recommendations [33]. These works count operations (Section 3.3). Ours differs in the unit, in the phases charged, and in removing unit-equivalent relations.

Automorphisms in the tower sieve. Barbulescu, Gaudry, Guillevic and Morain [5] construct NFS-compatible automorphisms of orders 2, 3, 4 and 6 with the conjugation method; Barbulescu [3] shows how fixed subfields force units to have zero virtual logarithm; De Micheli, Gaudry and Pierrot [14] use an order-2 automorphism in the linear algebra of a 521-bit record, and Zhu and Liu [39] build CM fields for the same purpose. The closest work is that of Al Aswad, Pierrot and Thomé [1], who solve the linear-algebra problem for tower automorphisms of order up to 12, with Galois Schirokauer maps and a virtual-logarithm identity that we reuse. Their automorphisms come from the conjugation method with coprime tower degrees, $(3, 4)$ or $(4, 3)$, and forgo the special form of p. Ours is the special shape $(6, 2)$, where the degrees are not coprime and $x \mapsto -x$ is the only automorphism of f over $\mathbb{Q}$; the order four comes from the twist $x \mapsto cx$. They give small examples and matrix-size reductions; we give instruction-level costs at

Table 2: Notation.

Symbol	Meaning
$p,\ r,\ u$	field characteristic, prime group order, seed of the curve family; $\mathrm{ord}_r(p) = 12$
$\mathbb{G}_T$	the subgroup of order r of $\mathbb{F}_{p^{12}}^*$
$K_h = \mathbb{Q}[y]/h$, $\mathcal{O}_h$	sextic base field and its ring of integers; p is inert in K_h
$f,\ g$; $K_f,\ K_g$	the two polynomials in x over $\mathcal{O}_h$, and the fields $K_s = K_h[x]/(s)$
(a, b); cell; slice; run	a pair in $\mathcal{O}_h^2$ (12 integers); one pair; all b for one a; consecutive a on a line
N_f, N_g; B; u_s	the two norms of $a - bx$; smoothness bound; $u_s = \log_2 N_s / \log_2 B$ (with corrections)
ρ; α	Dickman's function; Murphy's correction for small primes [30]
$\sigma,\ c,\ w$	involution of K_h; unit with $c\,\sigma(c) = -1$; element with $\sigma(w) = c^2 w$
$\tau,\ \chi$	$\tau : (y, x) \mapsto (\sigma y, cx)$ of order 4; $\chi = p^3 \bmod r$, $\chi^2 = -1$
U'	number of τ-orbits of factor-base ideals: the columns of the system before filtering
N, ν; N_l, N_{sw}	rows after filtering and $\nu = N/U'$; live and evaluated rows of the product form
(m, n)	block sizes of block Wiedemann
unit; C	$C = 20{,}000$ scalar AArch64 instructions $= 2^{14.29}$ instructions

cryptographic sizes. A conjugation-type design with an order-12 automorphism for BLS12-381 is a different point of the design space, which we priced only roughly (Section 9.4).

Filtering and linear algebra. Our filtering follows CADO-NFS [12, 36]; the product form is related to the partial triangulation of Boudot [9]; block Wiedemann and its generator follow [13, 37]; the batch smoothness tests follow Bernstein [8].

1.5 How to read this paper

The paper is written for three kinds of readers.

- *If you want the picture*: read Sections 2 and 3, the first subsection of each of Sections 4 to 8 (each opens with its idea in plain words), and Section 9. No algebraic number theory is needed.
- *If you estimate the security of pairings*: Sections 3, 5, 9 and 10 contain what affects other estimates.
- *If you implement or study NFS*: Sections 4, 6, 7 and the appendices contain the constructions, proofs and kernel-level details.

Table 2 collects the notation.

2 Background: the number field sieve in four ideas

This section is a primer. A reader who knows the tower number field sieve can skip to Section 3. Everything later in the paper is a way to make one of the four ideas below cheaper.

2.1 Why the target group matters

A pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ that is linear in both arguments: $e([a]P, [b]Q) = e(P, Q)^{ab}$. $\mathbb{G}_1$ and $\mathbb{G}_2$ are groups of points of an elliptic curve; $\mathbb{G}_T$ is the subgroup of order r of the multiplicative group of a finite field $\mathbb{F}_{p^k}$. For the two curves studied here, $k = 12$.

Linearity moves discrete logarithms across the pairing: if $P = [a]P_0$ then $e(P, Q) = e(P_0, Q)^a$, so whoever can compute logarithms in $\mathbb{G}_T$ can compute them in $\mathbb{G}_1$ [27]. A pairing-friendly curve is thus as strong as the *weaker* of two problems.

- On the curve, the best known attack is generic (Pollard's rho [32]) and costs about $\sqrt{r}$ group operations: about 2^{127} for a 255-bit r.

- In $\mathbb{F}^*_{p^{12}}$, the number field sieve (NFS) is *sub-exponential*: its cost grows much more slowly with the size of the field. For $\mathbb{F}_{p^{12}}$ with a p of special form, the relevant variants are the tower NFS [6] and its extended and special forms [21, 22], abbreviated here STNFS.

For BN254 and BLS12-381 the second problem is the easier one, and this paper is about its exact cost.

2.2 Idea 1: logarithms of small primes first (index calculus)

NFS belongs to the family of *index calculus* methods. The principle fits in a toy example. Work modulo $p = 1019$ with generator $g = 2$, and suppose we want $\log 787$, the exponent k with $2^k \equiv 787$.

1. *Collect relations.* Choose a *factor base* of small primes, here $\{2, 3, 5, 7\}$. Compute powers of g and keep those that happen to factor completely over the factor base (they are *smooth*):

$$2^{15} \equiv 160 = 2^5 \cdot 5, \qquad 2^{72} \equiv 686 = 2 \cdot 7^3, \qquad 2^{183} \equiv 189 = 3^3 \cdot 7.$$

   Taking logarithms turns each one into a *linear* equation modulo $p - 1 = 1018$ between the unknowns $\log 3, \log 5, \log 7$ (and $\log 2 = 1$):

$$15 = 5 + \log 5, \qquad 72 = 1 + 3\log 7, \qquad 183 = 3\log 3 + \log 7.$$

2. *Linear algebra.* Solve the system: $\log 5 = 10$, $\log 7 = 363$, $\log 3 = 958$.

3. *Individual logarithm (descent).* Multiply the target by powers of g until the result is smooth: $787 \cdot 2^{16} \equiv 147 = 3 \cdot 7^2$. Hence $\log 787 = 958 + 2 \cdot 363 - 16 \equiv 650 \pmod{1018}$, and indeed $2^{650} \equiv 787$.

Steps 1 and 2 are done once per field; step 3 is done once per target and is much cheaper. The whole cost is governed by one question: *how likely is a random-looking number to be smooth?*

2.3 Idea 2: smoothness is rare, and its probability is known

An integer is B-smooth if all its prime factors are at most B. For an integer near N, write $u = \log N / \log B$: the number is "u times longer" than the primes it should split into. The probability that it is B-smooth is close to Dickman's function $\rho(u)$ [15], which falls roughly like u^{-u} (Figure 1). Two consequences shape every NFS design.

- *Size is everything.* Shaving a few percent off the bit-length of the numbers tested multiplies the yield. At $u \approx 10$, lowering u by one tenth (one percent fewer bits) multiplies the probability by 1.45.
- *B is a dial.* A larger B makes smoothness likelier, so fewer candidates are tested; but there are more primes below B, so more relations are needed and the linear system grows. The optimum balances the two (Figure 3).

2.4 Idea 3: two number fields make the numbers small (NFS)

In the toy example the numbers tested for smoothness are as large as p. NFS makes them much smaller with an algebraic detour. Take two polynomials f and g that share a root modulo p (more generally, a common factor defining the target field). Each defines a number field, K_f and K_g, and both map to the same finite field (Figure 2). A small element such as $a - bx$ can be viewed in K_f and in K_g; its two images in the finite field are equal. If the element factors into small prime *ideals* on both sides, equating the two factorisations gives a relation between the (virtual) logarithms of those ideals.

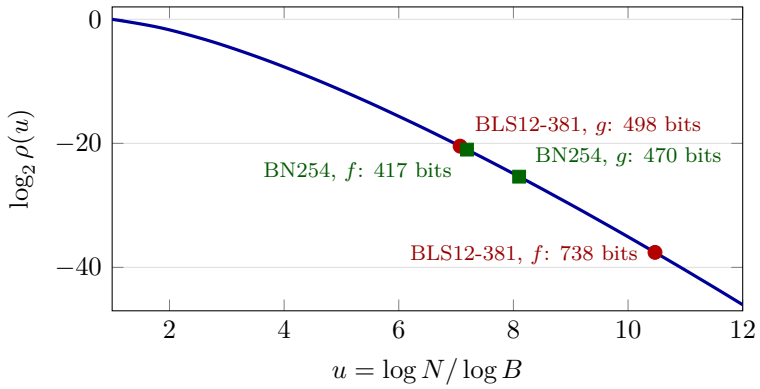


Figure 1: Dickman's function: the probability that a number u times longer than the smoothness bound is smooth. The marks show the operating points of the two designs of this paper (mean norm size over $\log_2 B$, with $B = 2^{70.5}$ for BLS12-381 and 2^{58} for BN254). A relation needs both sides smooth at once, so the two probabilities multiply.

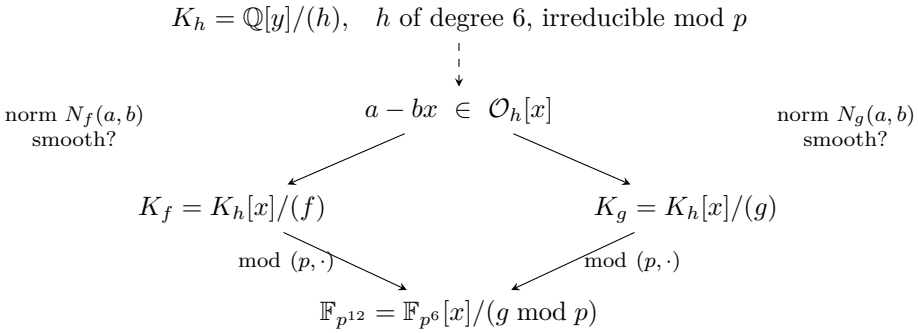


Figure 2: The tower number field sieve. Coefficients a, b live in the ring of integers of a base field K_h of degree 6 (six integers each: a 12-dimensional sieve). Both paths to $\mathbb{F}_{p^{12}}$ give the same element; when both norms are smooth, the two factorisations give one linear relation.

Whether $a - bx$ factors into small ideals is decided by an ordinary integer, its *norm*: the element is B-smooth when both integers $N_f(a,b)$ and $N_g(a,b)$ are. So the algorithm *sieves* over many small pairs (a,b), looking for pairs whose two norms are both smooth. The art of polynomial selection is to make both norms small.

Two technicalities will return later.

- *Units.* Number fields have units, elements such as $1+\sqrt{2}$ whose inverse is also an algebraic integer. An ideal factorisation does not see units, so their contribution is tracked by a few extra dense columns in the linear system, the *Schirokauer maps* [34]. Units are also why the same relation can be found many times (Section 5).
- *Symmetries.* If the fields have an automorphism that is compatible with the map to the finite field, then relations and ideals come in orbits, and only one representative per orbit needs to be found and stored. This is the subject of Section 4.

2.5 Idea 4: a tower, and the special form of p (STNFS)

For a field $\mathbb{F}_{p^{12}}$ one can spread the degree 12 between a base field and the two polynomials. The *tower* NFS first builds $\mathbb{F}_{p^6}$ from a sextic field K_h in which p stays prime, and then uses polynomials f, g of small degree in x with coefficients in K_h. The pairs (a,b) now have six integer coordinates each.

Pairing-friendly primes are not random: they are values $p = F(u)$ of a polynomial with tiny

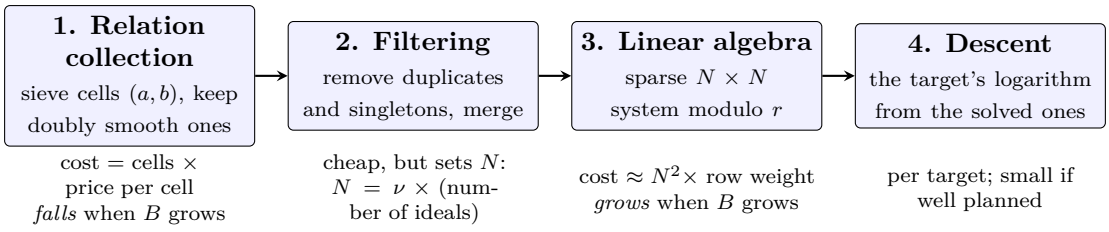


Figure 3: The four phases. The smoothness bound B trades the first phase against the third.

coefficients at a 64-bit *seed* u. For BN curves F has degree 4, for BLS12 curves degree 6. The *special* variant uses exactly this: $g = x^2 - uw$ has the root that F vanishes on modulo p, and f inherits the tiny coefficients of F. Both norms are then far smaller than for a general prime of the same size. This is why the 4569-bit field of BLS12-381 is much weaker than a 4569-bit field with a random p, and why the security of these curves was re-estimated after 2016 [4, 17, 21, 28].

2.6 The four phases and their costs

1. *Relation collection.* Visit a region of pairs (a, b), called *cells*, and detect those whose two norms are B-smooth. Its cost is (number of cells) × (work per cell). The number of cells is fixed by the smoothness probabilities: one needs about as many relations as there are prime ideals below B. A *sieve* does not factor each norm; for each prime ideal it steps through the cells it divides and adds $\log q$ there, like the sieve of Eratosthenes, so that the work per cell is a few dozen instructions.
2. *Filtering.* Discard useless relations and combine others to shrink the system, at the price of denser rows. Filtering is cheap; what matters is the ratio ν between the final dimension N and the number of ideals.
3. *Linear algebra.* Solve a sparse $N \times N$ system modulo the 255-bit prime r with the block Wiedemann algorithm [13]: about $(1 + o(1))N$ multiplications of the matrix by a vector, split into independent sequences, and then a "generator" computation that combines them [37]. The cost is about N^2 times the row weight.
4. *Descent.* Write the target as a product of ideals with known logarithms, through a tree of smaller and smaller sieving problems.

Four ways to be cheaper. Everything in this paper falls under one of four headings, and we will say which each time:

(A) *fewer cells*: more relations per cell (smaller norms, better regions), or fewer relations needed (symmetries);

(B) *cheaper cells*: sieve organisation and machine-level kernels;

(C) *a smaller matrix*: symmetries and better filtering;

(D) *cheaper matrix products*: arithmetic, layout and the structure of the solver.

A fifth heading is not an improvement but is just as important: *counting correctly*. Two of our findings are corrections that make estimates more expensive (Sections 5 and 10.3).

3 One cost model, and what the state of the art was

Cost estimates for NFS are hard to compare because they count different things. This section fixes one metric, says what published estimates count, and places every number of the paper on one scale.

3.1 The metric: every instruction, once

We use the cost model of the CurveSmash challenge [23], which is simple to state.

- *What is counted.* Every scalar AArch64 instruction executed by the attack, summed over all processors: arithmetic, loads, stores, addressing and control, in preprocessing, sieving, cofactorisation, sorting, transfers, filtering, linear algebra, descent, failed trials, verification and output. Vector (SIMD) instructions are not allowed; the atomic add-to-memory instructions of ARMv8.1 are, and our counter-update loops use them.
- *What is not counted.* Memory, latency, communication beyond the instructions that move data, energy and hardware. Section 10 returns to this; it matters, because our final designs use a great deal of memory.
- *Unit.* Costs are reported in units of $C = 20{,}000$ instructions, the rounded cost of one dense multiplication in $\mathbb{F}_{p^{12}}$ of BLS12-381 in the `blst` library [35] (22,996 instructions measured). One unit is $2^{14.29}$ instructions. *Unless stated otherwise, a number such as* 113.2 *means* $2^{113.2}$ *units*; we also give instructions for the headline figures.
- *Evidence.* Nothing is run at full size. A claim is a *ledger*: a list of rows, each the product of a count (cells, relations, matrix rows) and a price (instructions per item). Prices are measured by instruction counters on real kernels at reduced scale. Counts come from exact lattice-point counts, Monte Carlo pools and simulations. The hypotheses that connect them to a full-size run are listed in Table 10.

3.2 What "cost" means

An attack is randomised. Throughout this paper its *cost* is the expected work of one complete run, and every ledger comes with a lower bound on the probability that the run returns the correct logarithm. For both curves that probability is above 0.998 (Section 8.3), so the cost per solved instance is the same number to within 0.003 bit. The descent is charged at its expected work; a work meter set far above the expectation aborts runaway runs, and the probability of reaching it is part of the failure bound.

3.3 What published estimates count

Asymptotics. The special tower NFS reaches $L_Q[1/3, (32/9)^{1/3}]$ in the medium-prime range [21], against $(64/9)^{1/3}$ for the classical NFS. With the $o(1)$ dropped, these formulas say little about concrete cost [28].

Barbulescu–Duquesne (BD) [4]. Norm sizes are measured over the sieving domain, and

$$\text{cost} = \frac{2B}{A \ln B}\, \rho\Big(\tfrac{\log_2 N_f}{\log_2 B}\Big)^{-1} \rho\Big(\tfrac{\log_2 N_g}{\log_2 B}\Big)^{-1} + 2^7 \frac{B^2}{A^2 (\ln B)^2 (\log_2 B)^2}$$

is minimised over B, with A the automorphism factor. The model takes one operation per sieved element and one operation per modular addition in the linear algebra.

Guillevic–Singh (GS) [17, 18]. The sieved set is a coefficient cube $|a_i|, |b_i| \le A$ of volume V. The cost is

$$\text{RC} = \frac{V}{\text{aut}} \ln \ln B, \qquad \text{LA} = \left\lceil \frac{\log_2 r}{64} \right\rceil \cdot 200 \cdot \left(\frac{2\,\text{Li}(B)}{\text{flt}} \right)^2,$$

with aut $= 2$; the yield is a Monte Carlo mean of $\rho(u_f)\rho(u_g)$ with Murphy's α correction [30]. We ran the public code (commit `57702fa3`) with 10^5 samples. For BLS12-381 it returns $A = 686$, $B = 2^{65.3}$, RC $=$ LA $= 2^{125.0}$: the published 2^{126} of their Table 10.

Table 3: What each model counts. "Operation" is the model's own abstract unit.

	BD [4]	GS [17]	this paper
Unit	operation	operation	instruction
Work per sieved cell	1	$\ln \ln B$	measured (13 to 17)
Cofactorisation	no	no	yes
Filtering ratio	$\log_2 B$	fixed (9 or 20)	simulated
Matrix product	1 per entry	200 per row, per word	measured
Individual logarithm	no	no	yes
Failure probability	no	no	yes
Unit-equivalent relations	counted	counted	removed
Memory	no	no	reported, not charged

Two different "BN254". Published BN figures (BD: $2^{99.7}$; GS: 2^{102}) are for the curve with seed $u = -(2^{62} + 2^{55} + 1)$. The curve deployed in Ethereum's precompiles [11], studied here, has $u = 4965661367192848881$. No refereed worked estimate exists for it; the GS code lists 103.

3.4 What was the state of the art?

Published estimates give a polynomial pair, a sieving region and two operation counts. Comparing with them in instructions requires a price per operation, and it is easy to get this wrong in either direction: by reading an operation count as an instruction count, or by pricing a published design with machinery it does not contain and calling the result prior art. We therefore use two reference points (Table 4, Figure 4).

1. *Published estimates, in their own units.* They omit whole phases (Table 3) and cannot be converted without a price per operation. To connect with them, we also express our designs in *their* conventions.

2. *A baseline in our model*: the published $(6, 2)$ design for BLS12-381 implemented with standard components and charged completely. Concretely: the polynomial shape of [17]; unit duplicates removed, since otherwise the attack does not terminate (Section 5); a conventional sieve with cofactorisation by ECM; classical filtering and block Wiedemann with the sign symmetry; a special-q descent. Its rows are $2^{122.2}$ (collection), $2^{121.0}$ (linear algebra) and $2^{120.5}$ (descent). It answers the question "what does the known attack cost in this model?".

BLS12-381: what changes.

- *In instructions, the baseline is level with generic attacks.* $2^{123.0}$ units is $2^{137.3}$ instructions, against $2^{137.2}$ for Pollard's rho on the curve. Whether the literature's "126 against 127" is a comfortable gap depends entirely on what an operation costs.

- *The techniques of this paper lower the cost by* 9.8 *bits*, to $2^{127.5}$ instructions. About one bit comes from the twisted tower. The rest is sieve organisation, kernels, filtering, the structure of the solver and descent planning (Section 9): exactly what operation-count models treat as constants.

- *The closeness of two numbers is a coincidence.* That the published 2^{126} operations and our $2^{127.5}$ instructions are close does not mean that the published figure was right in instructions. The published design does not collect enough distinct relations (Section 5); its operation counts exclude phases that we charge; and our figure rests on techniques that it does not use.

Table 4: Reference points on one scale. Costs are expected work of a complete run. Entries marked * are operation counts in the cited model and are not comparable with the other rows. Generic attacks are shown for scale; they are the best instruction counts we know for Pollard's rho [32, 38] on each group, taken from ledgers of the same kind by other authors in the CurveSmash challenge. The baseline is in expected work, with the descent charged at its expectation.

	$\log_2$ units	$\log_2$ instructions
BLS12-381		
Published estimate [17]	126*	–
Pollard rho in $\mathbb{G}_1$ (negation and endomorphism classes; best known instruction count)	122.9	137.2
Baseline: published design, standard components	123.0	137.3
This paper	**113.23**	**127.5**
BN254 (Ethereum parameters)		
Published estimates for another BN prime [4, 17]	99.7*, 102*	–
Our design counted in those conventions (BD; GS)	99.1*, 102.2*	–
Pollard rho in $\mathbb{G}_T$ (best known instruction count)	120.1	134.4
This paper	**88.95**	**103.2**

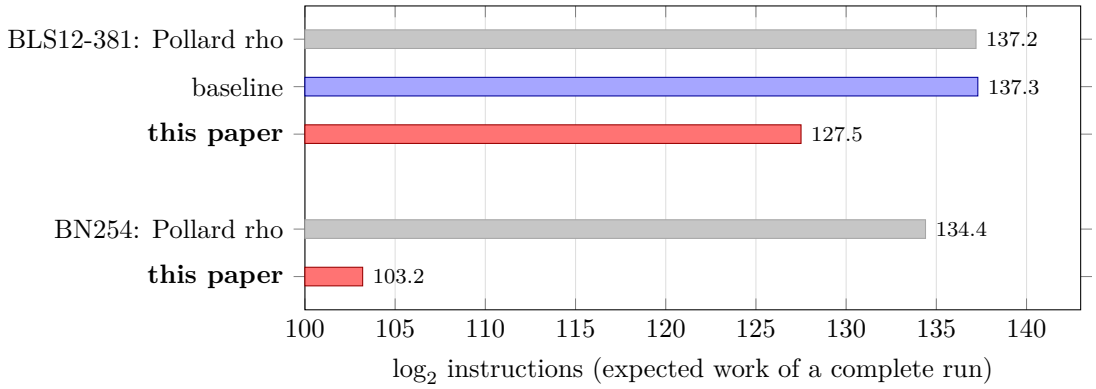


Figure 4: The reference points of Table 4 in instructions.

BN254: what changes, and what does not. For BN254 the picture is different, and we do not claim a new security level. That its target group offers about 100 bits, far below the 127 bits of the curve, has been the consensus since [4, 21, 28]. Counted in the conventions of the published models, our design gives $2^{99.1}$ (BD) and $2^{102.2}$ (GS) operations[1]: it sits where the published figures for the other BN prime are. What is new is (i) a valid, duplicate-free design for the Ethereum prime, for which no worked estimate existed; (ii) the price of the operations, 17 instructions per sieved cell and about 1,350 per row of a matrix product, which turns "about 2^{100} operations" into $2^{103.2}$ instructions including every phase; and (iii) the fact that this required the descent to be designed, not ignored: with the parameters that collection and linear algebra alone would choose, the descent dominates the attack (Section 8). The gap of 31 bits to Pollard's rho in Table 4 restates the known weakness in instructions; it is not a 31-bit improvement.

4 The twisted tower: a symmetry of order four

4.1 The idea

Symmetries are the cheapest savings in NFS. If an automorphism of order k acts on both number fields compatibly with the finite field, then relations and prime ideals come in orbits of k. One

[1] Computed for the design at $B = 2^{56}$, before the sieve techniques moved the optimum to 2^{58}; cells and matrix size changed little.

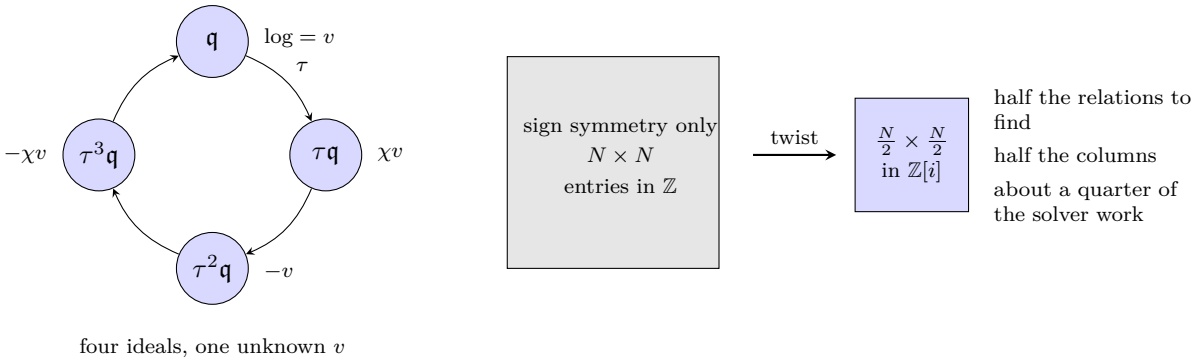


Figure 5: What a symmetry of order 4 does. Left: the four ideals of an orbit share one unknown. Right: compared with the sign symmetry alone, the system has half as many rows and columns, with entries $a + b\,i$ in the Gaussian integers read modulo r through $i \mapsto \chi$.

representative per orbit is enough: the sieve needs k times fewer relations, the matrix has k times fewer columns, and, since linear algebra costs about N^2, it becomes about k^2 times cheaper (Figure 5).

In the special tower NFS with $g = x^2 - uw$, both polynomials are polynomials in x^2, so $x \mapsto -x$ is a symmetry of order 2. Every published estimate uses it. Can one do better *without giving up the special form of* p? The known constructions of larger automorphisms [1, 5] use the conjugation method and tower degrees that are coprime; they do not apply to the $(6, 2)$ special shape.

Our observation is that a symmetry of the *base field* can be combined with a *rescaling* of x. Suppose the sextic field K_h has an involution σ. Applying σ to the coefficients of $g = x^2 - uw$ gives $x^2 - u\sigma(w)$, a different polynomial. But if $\sigma(w) = c^2 w$ for some unit c, then substituting $x \mapsto cx$ repairs it: $(cx)^2 - u\,c^2 w = c^2 g$. The combined map

$$\tau : \ (y, x) \longmapsto (\sigma(y),\, c\,x)$$

is then a symmetry of g, and also of f. Applying it twice sends x to $c\,\sigma(c)\,x$. If $c\,\sigma(c) = -1$, then τ^2 is the old symmetry $x \mapsto -x$, and τ itself is a *square root* of it: a symmetry of order 4. We call the pair (σ, c) a *twist*, and the resulting tower a twisted tower.

On the finite field, τ acts as the Frobenius power $z \mapsto z^{p^3}$. On the subgroup of order r this is multiplication of logarithms by the constant $\chi = p^3 \bmod r$, and $\chi^2 = -1$: logarithms of the four ideals of an orbit are $v, \chi v, -v, -\chi v$. The system can therefore be written over the Gaussian integers $\mathbb{Z}[i]$, with i read as χ.

4.2 The construction

Proposition 1 (Frobenius twists in a TNFS tower)**.** *Let $K_h = \mathbb{Q}[y]/h$ with $\deg h = \eta$ and h irreducible modulo p. Let $f = w^e F(x^2/w)$, $e = \deg F$, and $g = x^2 - uw$ be a special pair, with $w \in \mathcal{O}_h$ and uw a non-square in $\mathbb{F}_{p^\eta}$. Suppose that $\sigma \in \mathrm{Aut}(K_h)$ has order d and that $c \in \mathcal{O}_h^*$ satisfies*

$$\sigma(w) = c^2 w \qquad \text{and} \qquad c\,\sigma(c) \cdots \sigma^{d-1}(c) = -1.$$

Then:

(a) $\tau : (y, x) \mapsto (\sigma y, cx)$ satisfies $\tau(f) = c^{2e} f$ and $\tau(g) = c^2 g$, so it defines automorphisms of K_f and K_g. It has order $2d$, with $\tau^d = (x \mapsto -x)$.

(b) τ preserves the kernels of the maps $\mathcal{O}_h[x]/(f) \to \mathbb{F}_{p^{2\eta}}$ and $\mathcal{O}_h[x]/(g) \to \mathbb{F}_{p^{2\eta}}$ and induces a field automorphism Frob^j of $\mathbb{F}_{p^{2\eta}}$. If this induced map has order $2d$, then $j = m\,\eta/d$ with $\gcd(m, 2d) = 1$.

Table 5: The two twisted towers. F is the polynomial with $p = F(u)$; D is the discriminant of $\mathcal{O}_h$.

	BLS12-381	BN254
seed u	$-$`0xd201000000010000` (64 bits)	4965661367192848881 (63 bits)
F	$z^6 - 2z^5 + 2z^3 + z + 1$, with $F(u) = 3p$	$36z^4 + 36z^3 + 24z^2 + 6z + 1$, with $F(u) = p$
h	$y^6 + 2y^4 - 2y^3 - 2y^2 - 1$	$y^6 - y^5 - 2y^4 + 5y^3 - 2y^2 - y + 1$
D; signature; $[\mathcal{O}_h : \mathbb{Z}[y]]$	$2^9 \cdot 19^2 = 184832$; $(2, 2)$; 4	$98000 = 2^4 \cdot 5^3 \cdot 7^2$; $(2, 2)$; 1
w; $N(w)$	$y + 1$; 2	$y^5 - 3y^3 + 2y^2 + 2y - 2$; 5
f	$w^6 F(x^2/w)$, degree 12	$w^4 F(x^2/w)$, degree 8
g	$x^2 + \lvert u \rvert w$	$x^2 - uw$
order of τ; action	4; Frob^3	4; Frob^3
Schirokauer columns	10	7
coset sieve (Section 6)	yes, at the prime of norm 2	no

(c) If moreover r is a primitive divisor of $\Phi_{2\eta}(p)$ (so $\mathrm{ord}_r(p) = 2\eta$, as when 2η is the embedding degree), then $\chi = p^j \bmod r$ has multiplicative order exactly $2d$, and $\log \tau(z) = \chi \log z$ on the order-r subgroup.

(d) For a relation (a, b), the pair $(\sigma a, c\,\sigma b)$ is a relation whose ideals are the τ-images.

Proof. (a) $\tau(g) = c^2x^2 - u\sigma(w) = c^2 g$ and $\tau(f) = \sigma(w)^e F(c^2x^2/\sigma(w)) = c^{2e} w^e F(x^2/w)$. Further, $\tau^d(x) = c\,\sigma(c) \cdots \sigma^{d-1}(c)\, x = -x$ and $\tau^d(y) = y$, so τ has order $2d$. (b) Both maps $\mathcal{O}_h[x] \to \mathbb{F}_{p^{2\eta}}$ have kernel (p, g), since $f \equiv w^e F(u) \equiv 0$ ($F(u)$ is a multiple of p) modulo $(p, x^2 - uw)$. σ fixes $p\mathcal{O}_h$, which is prime since h is inert, and $\tau(g) = c^2 g$ with c a unit, so τ preserves (p, g) and induces a ring automorphism of $\mathbb{F}_{p^{2\eta}}$, that is, a power Frob^j. Frob^j has order $2\eta/\gcd(j, 2\eta)$; if this is $2d$, then $\gcd(j, 2\eta) = \eta/d$, whence the form of j. (c) On a cyclic group of order r, Frob^j is $z \mapsto z^{p^j}$. The order of p^j modulo r is $2\eta/\gcd(j, 2\eta) = 2d$ because $\mathrm{ord}_r(p) = 2\eta$. The order of the map on $\mathbb{F}_{p^{2\eta}}$ alone does not suffice: a subgroup on which p has smaller order would give a smaller order for χ. (d) $\tau(a - bx) = \sigma a - c\,\sigma b\, x$, and τ maps prime ideals to prime ideals of the same norm. □

The two instances. Table 5 gives the towers used for the two curves. In both, $\eta = 6$, $d = 2$, and the induced map is Frob^3, so $\chi = p^3 \bmod r$ with $\chi^2 = -1$. The artifact certifies $\sigma(y) \equiv y^{p^3}$ and $(uw)^{(p^3-1)/2} = c$ in $\mathbb{F}_{p^6}$. Since $p \equiv u$ (BLS12) or $p \equiv 6u^2$ (BN) modulo r, χ has a short representative of about 192 bits, which Section 7 exploits.

For BLS12-381 the involution is $\sigma(y) = -y^5 - 2y^3 + 2y^2 + 2y$ and the unit is $c = (y^5 + 3y^3 - 3y^2 - 3)/2$, with $c\,\sigma(c) = -1$ and $\sigma(w) = c^2 w$; the check $(uw)^{(p^6-1)/2} = -1$ gives $\mathbb{F}_p[y, x]/(h, g) = \mathbb{F}_{p^{12}}$. For BN254, $\sigma(y) = -y^5 + y^4 + 2y^3 - 5y^2 + 2y + 1$ and $c = y^4 - 2y^2 + 2y + 1$, with the same identities.

4.3 The quotient system

The next proposition makes precise the system that is actually solved. It reuses the framework of Al Aswad, Pierrot and Thomé [1]: τ is a TNFS automorphism in their sense, and the virtual-logarithm identity is their Theorem 25 applied to it.

Proposition 2 (quotient system)**.** *In the setting of Proposition 1 with $d = 2$ and $\mathrm{ord}_r(p) = 2\eta$, let $s \in \{f, g\}$, Γ_s the group of factor-base units of K_s, and K_s^+ the fixed field of τ^2. Assume:*

(A1) r is unramified in K_s and prime to the class number h_s and to the number of roots of unity of K_s;

(A2) (unit condition) the Schirokauer map $\lambda_s(\varphi) = (\varphi^\epsilon - 1)/r \bmod r\mathcal{O}_s$, with $\epsilon = \mathrm{lcm}(r^{d_i} - 1)$ over the residue degrees d_i of r, is injective on $\mathcal{O}_s^\times/(\mathcal{O}_s^\times)^r$, and r is prime to $[\mathcal{O}_s^\times : \mathcal{O}_{K_s^+}^\times \mu_s]$.

Let $\Lambda_s = \pi_\chi \circ \lambda_s$, where π_χ is the projection of $\mathcal{O}_s/r\mathcal{O}_s$ onto the τ-eigenspace of eigenvalue χ. Then:

(i) The unit contribution to any virtual logarithm map factors through Λ_s. The image of the units has dimension $(\operatorname{rk}\mathcal{O}_s^\times - \operatorname{rk}\mathcal{O}_{K_s^+}^\times)/2$. For BLS12-381 this is $(35-17)/2 = 9$ on f and $(5-3)/2 = 1$ on g, so 10 Schirokauer columns.

(ii) With virtual logarithms defined through (Λ_f, Λ_g), $\operatorname{vlog}(\tau\gamma) = \chi\operatorname{vlog}\gamma$ for $\gamma \in \Gamma_s$, and $\operatorname{vlog}(\tau\mathfrak{q}) = \chi\operatorname{vlog}\mathfrak{q}$ for every factor-base ideal.

(iii) If $\tau^2\mathfrak{q} = \mathfrak{q}$ (an orbit of length 1 or 2), then $\operatorname{vlog}\mathfrak{q} = 0$. For a relative-degree-1 ideal $\mathfrak{q} = (\mathfrak{p}, x - t)$ this happens only if $2 \in \mathfrak{p}$ or $t \equiv 0 \bmod \mathfrak{p}$, that is, only above 2 and the primes dividing uw. These $O(1)$ columns are removed.

(iv) Choose one ideal $\mathfrak{q}$ per orbit of length 4. A relation with valuation e_k at $\tau^k\mathfrak{q}$ contributes $(e_0 - e_2) + (e_1 - e_3)\chi$ to the column of $\mathfrak{q}$, and its Schirokauer part is $\Lambda_s(a - bx)$. The relation of $\tau(a, b)$ is χ times that of (a, b) and is redundant. The system therefore has one column per orbit and a few dense columns. Its entries are Gaussian integers $a + b\,i$ read in $\mathbb{F}_r$ through $i \mapsto \chi$, and it is solved over $\mathbb{F}_r$.

Proof. Every virtual logarithm map factors through $\mathbb{F}_{p^{12}}^*/(\mathbb{F}_{p^{12}}^*)^r$: $\operatorname{vlog}\gamma = \log\Psi_s(\gamma)$ up to a fixed scalar, with Ψ_s the reduction map [1, Lemma 22]. By Proposition 1, $\Psi_s(\tau\gamma) = \Psi_s(\gamma)^{p^3}$, hence $\operatorname{vlog}(\tau\gamma) = \chi\operatorname{vlog}\gamma$.

(i) τ fixes $r\mathcal{O}_s$ and ϵ, so $\lambda_s \circ \tau = \tau \circ \lambda_s$. Because $r \nmid 4$, the $\mathbb{F}_r[\langle\tau\rangle]$-module $E = \mathcal{O}_s^\times \otimes \mathbb{F}_r$ is semisimple and splits into eigenspaces for $1, -1, \chi, -\chi$. The restriction ℓ of vlog to E satisfies $\ell \circ \tau = \chi\ell$, so ℓ vanishes on every eigenvector whose eigenvalue is not χ. Hence ℓ factors through the χ-component. By (A2), λ_s maps that component injectively into the χ-eigenspace of $\mathcal{O}_s/r$, where π_χ is the identity, so $\ell = \ell' \circ \Lambda_s$. For the dimension: the τ^2-invariant part of E is $\mathcal{O}_{K_s^+}^\times \otimes \mathbb{F}_r$ by (A2). On the complement $\tau^2 = -1$, so τ is a rational operator whose characteristic polynomial there is $(X^2+1)^{(\mathrm{rk}-\mathrm{rk}^+)/2}$, and its $\pm\chi$ eigenspaces have equal dimension. For BLS12-381 the ranks follow from the certified signatures $K_f(0, 36)$, $K_f^+(0, 18)$ (the subfield generated by x^2), $K_g(0, 6)$ and $K_g^+ = K_h\,(2, 2)$.

(ii) By (A1) choose γ with $(\gamma) = \mathfrak{q}^{h_s}$. Then $h_s\operatorname{vlog}\mathfrak{q} = \operatorname{vlog}\gamma - \Lambda_s(\gamma)\cdot v$, with v the Schirokauer unknowns. Applying τ and using $\Lambda_s(\tau\gamma) = \pi_\chi(\tau\lambda_s(\gamma)) = \chi\Lambda_s(\gamma)$ gives $h_s\operatorname{vlog}(\tau\mathfrak{q}) = \chi h_s\operatorname{vlog}\mathfrak{q}$; divide by h_s. This is the argument of [1, Theorem 25].

(iii) If $\tau^2\mathfrak{q} = \mathfrak{q}$, then $\operatorname{vlog}\mathfrak{q} = \chi^2\operatorname{vlog}\mathfrak{q} = -\operatorname{vlog}\mathfrak{q}$, and r is odd. Since $\tau^2(\mathfrak{p}, x - t) = (\mathfrak{p}, x + t)$, the ideal is fixed only if $2t \in \mathfrak{p}$, and $t \equiv 0$ requires $\mathfrak{p} \mid s(0)$.

(iv) By (ii), $\sum_k e_k\operatorname{vlog}(\tau^k\mathfrak{q}) = (\sum_k e_k\chi^k)\operatorname{vlog}\mathfrak{q}$ and $\chi^2 = -1$. The relation of $\tau(a, b)$ has valuations e_{k-1} and Schirokauer part $\chi\Lambda_s(a - bx)$. □

Because χ lies in $\mathbb{F}_r$, we can project the usual Schirokauer map onto one eigenspace and keep the minimal number of dense columns, the dimension of the χ-part of the units. The general Galois Schirokauer maps of [1] would need 42 columns here (36 on K_f and 6 on K_g), and a sign-only design needs 21.

4.4 What the twist buys, and what it costs

Gain. With orbits of size 4 instead of 2:

- the number of relations to find and the matrix width both halve (heading A and heading C of Section 2.6);
- multiplying by χ costs one modular multiplication by a fixed short constant (heading D);
- the dense Schirokauer columns fall from 21 to 10.

In an ablation on BLS12-381 at an intermediate stage of the design ($B = 2^{69}$), two ledgers that differ only in the base field and what follows from it compare as follows: collection -0.60 bit, linear algebra -2.00 (115.81 → 113.82), generator -1.52, descent -0.78, total -0.94 (117.47 → 116.54). It is the largest single algebraic step in this work. A second ablation, which re-optimises the untwisted design separately (published polynomials, fundamental domain, same sieve and solver techniques), gives a difference of 0.7 to 1.1 bit. Both were measured before the last linear-algebra techniques, which lower the weight of the matrix in either design; we have not repeated them on the final ledgers (Section 10).

Price. The condition $c\,\sigma(c) = -1$ is restrictive. It says that -1 is a relative norm from K_h to the cubic subfield K_3 fixed by σ. Writing $K_h = K_3(\sqrt{\delta})$, this forces the Hilbert symbol $(-1, \delta)_v = 1$ at every place v of K_3. The real places must split (signature $(2,2)$ or $(6,0)$), and w cannot generate a prime of norm $3, 11, 23, 27, 31, \ldots$. Among the admissible fields, the discriminant is larger than that of the best untwisted sextic, and a larger discriminant means larger norms. For BLS12-381 the smallest admissible norm of w is 2, which costs primitive pairs (the coprime share falls from 0.918 to 0.696); of the factor 2 in relations, only 0.37 bit of cells survives, the rest of the gain being the matrix. A search of 754 (field, w) candidates up to $D = 3 \cdot 10^6$ found the BLS12-381 field of Table 5 best. For BN254, $p \equiv 7 \bmod 8$ makes 2 a square, so a w of norm 2 is excluded; the field of Table 5, with $N(w) = 5$, was the best of 20 twisted candidates by at least 0.5 bit.

Larger symmetries do not pay. A symmetry group of a special $(6,2)$ pair is cyclic of order dividing 12, and order 8 is impossible, because the group embeds in $\mathrm{Gal}(\mathbb{F}_{p^{12}}/\mathbb{F}_p)$. Orders 6 and 12 exist, but only over base fields with much larger discriminants. Section 9.4 gives the numbers: they lose between 0.4 and 6 bits against the order-4 twist.

Certificates. The artifact certifies the field facts on which the propositions rest: maximality of the order, the factorisation of small primes, r prime to discriminants and class numbers, the signatures, the action of τ as Frob^3, and exact local densities at the ramified primes.

5 Counting each relation once

This section is about counting, not about speed. It explains the largest single correction to earlier estimates, including our own first ones.

5.1 The idea in two dimensions

Take the field $\mathbb{Q}(\sqrt{2})$. The number $\varepsilon = 1 + \sqrt{2}$ is a unit: $\varepsilon \cdot (\sqrt{2} - 1) = 1$. If v is any element, then v, εv, $\varepsilon^2 v, \ldots$ all generate the *same ideal*. They have the same norm up to sign, so they are smooth together or not at all, and they give the same relation.

Plot each element by its two real embeddings (Figure 6). Multiplying by ε stretches one coordinate and shrinks the other, so the multiples of v slide along a hyperbola. A sieve region that is "round" in these coordinates contains several points of each orbit. A sieve that visits all of them does the work several times and finds the same relation several times; an estimate that counts all of them believes it has several times more relations than it has.

5.2 Statement

Lemma 3 (duplicates)**.** *Let $\varepsilon \in \mathcal{O}_h^*$. The pairs (a, b) and $(\varepsilon a, \varepsilon b)$ generate the same principal ideals $(a - b\theta)$ in K_f and K_g, so they have the same factor-base factorisation, and their Schirokauer rows differ by the fixed vector $\lambda(\varepsilon)$. Over a sieve region R, the number of independent relations is therefore at most the number of $\mathcal{O}_h^*$-orbits meeting R that yield a relation, plus the unit rank.*

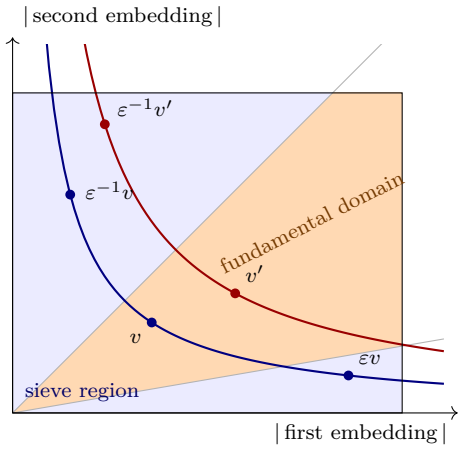


Figure 6: Unit multiples in a field with one fundamental unit ε (two real embeddings, absolute values). The points $\varepsilon^k v$ lie on a hyperbola of constant norm and represent one relation. The box contains three copies of v and two of v'. The shaded wedge is a fundamental domain: it contains exactly one point per orbit. The fields used in this paper have unit rank 3, and the same picture holds with a three-dimensional family of hyperbolic directions.

Table 6: Unit duplicates in the cube $A = 686$ ($b_5 \geq 0$, coprime pairs), verification census (10^9 stratified points, uncapped, bootstrap standard errors). $\log_2 \mathbb{E}[1/m]$: cell fraction a fundamental-domain sieve visits. $\log_2 \delta$: smoothness-weighted fraction of distinct relations at $B = 2^{65.316}$.

Pair (h; signature of K_h)	$\log_2 \mathbb{E}[1/m]$	$\log_2 \delta$
printed in [17]: $y^6 - y^2 + 1$; $(0, 3)$	-1.31	-3.003 ± 0.007
valid variant: $y^6 + y^3 + y^2 - y - 1$; $(2, 2)$	-0.67	-2.808 ± 0.006
our BLS12-381 field, in a $\mathbb{Z}[y]$ cube; $(2, 2)$	-0.10	-0.877 ± 0.005

Proof. $\varepsilon(a - b\theta)$ and $a - b\theta$ differ by a unit, so they generate the same ideals on both sides. Schirokauer maps are homomorphisms, so the rows differ by $\lambda(\varepsilon)$, and these differences span at most the unit rank. □

Let $m(v)$ be the number of unit multiples of v inside the region, up to torsion. A yield estimate over a region that is not reduced modulo units must be weighted by $\delta = \mathbb{E}_{\rho\rho}[1/m]$, the *smoothness-weighted* mean of $1/m$. A sieve over a fundamental domain visits only the fraction $\mathbb{E}[1/m]$ of the cells and finds the same distinct relations. The weighting matters: the points most likely to be smooth are the balanced ones, with all embeddings of similar size, and those are exactly the points with the most multiples inside a cube. So δ is far smaller than the unweighted $\mathbb{E}[1/m]$.

Dividing by the number of roots of unity, as published estimators do, removes only the torsion units (± 1). When $h(0) = \pm 1$, the variable y itself is a unit of infinite order, and coefficient cubes are full of multiples.

5.3 Measured size of the effect

For the coefficient cube of the published BLS12-381 estimate [17] ($|a_i|, |b_i| \leq 686$) we counted $m(v)$ with two separately written programs: a first census of $2 \cdot 10^5$ points with capped unit exponents, and a verification census of 10^9 stratified points with an uncapped search and bootstrap errors (Table 6).

For the valid pair closest to the published design, only $2^{-2.81} \approx 1/7$ of the relations that the estimate counts are distinct. At the published parameters, the estimator reports $2^{61.39}$ relations, of which $2^{58.58}$ are distinct, against $2^{60.85}$ needed: the design as published is 2.3 bits short of

relations. Our own first ledgers made the same mistake, by about 3.5 bits.

Remark 4 (a second defect of the printed pair). For a monic sextic h irreducible modulo p, $N_{\mathbb{F}_{p^6}/\mathbb{F}_p}(y) = h(0)$, so uy is a square in $\mathbb{F}_{p^6}$ if and only if $u^6 h(0)$ is a square modulo p. The pair printed for BLS12-381 in [17] has $h = y^6 - y^2 + 1$ and $g = x^2 - uy$: then $h(0) = 1$, g splits over $\mathbb{F}_{p^6}$, and the pair does not define $\mathbb{F}_{p^{12}}$ (we confirmed $(uy)^{(p^6-1)/2} = 1$). The next sextic in their list, $y^6 + y^3 + y^2 - y - 1$, is valid and gives the same published cost; we call it the *valid pair*. A check of g over $\mathbb{F}_{p^\eta}$ catches such cases.

5.4 Sieving a fundamental domain

The cure is to sieve one point per orbit. In our designs, a is required to be the least element of its class under $\langle \pm 1, \mathcal{O}_h^*, \sigma \rangle$ for a fixed quadratic form and a lexicographic tie-break, and b has a sign condition; this gives one point per orbit of the units *and* of the automorphism τ of Section 4. Canonicity is decided by an exact finite enumeration. Three practical consequences follow.

- The outer loop runs over canonical a only; b is free. Sieving is organised by *slices* (fixed a) and *runs* (consecutive canonical values of a along a line), which the walks of Section 6 exploit.
- Pairs (a, b) sharing a prime of $\mathcal{O}_h$ repeat the relation of a smaller pair and are excluded (*base-primitive* pairs). Under the smoothness heuristic they would otherwise look 3 to 60 times likelier to be smooth than they are worth.
- Every design choice downstream (polynomials, regions, thresholds) must be evaluated on canonical points only. We retracted two apparent gains (of 0.3 and 0.8 bit) that came from regions which silently contained more unit multiples.

Scope. Lemma 3 concerns every estimator that samples coefficient cubes, including published tables for BN, BLS, KSS and Cocks–Pinch curves [2, 16, 17, 19]. We measured the correction only for the two BLS12-381 pairs above; its size for other curves depends on their unit groups and is not known to us.

6 Relation collection: a sieve built on reuse

Relation collection is about 85% of the cost for both curves. Its cost is the number of cells times the work per cell. Sections 4 and 5 dealt with the number of cells; this section is about the work per cell, which the techniques below bring to 13.3 instructions for BLS12-381 and 17.3 for BN254.

6.1 The idea

A sieve does not look at each cell. For each prime ideal $\mathfrak{q}$ of the factor base, it visits the cells whose norm $\mathfrak{q}$ divides (its *hits*) and adds $\log q$ to a counter there. Cells whose counter comes close to the size of the norm are *survivors* and are examined exactly. The work is therefore one small step per hit, and a prime of norm q hits one cell in q. Three observations organise everything that follows.

1. *Do not sieve what you can test.* Only one of the two norms needs a sieve. The other can be tested exactly on the survivors, in bulk.
2. *Small primes are the expensive ones.* A prime of norm q makes $1/q$ hits per cell, so most hits come from the smallest primes, and each brings little information. They are better handled by arithmetic on survivors.

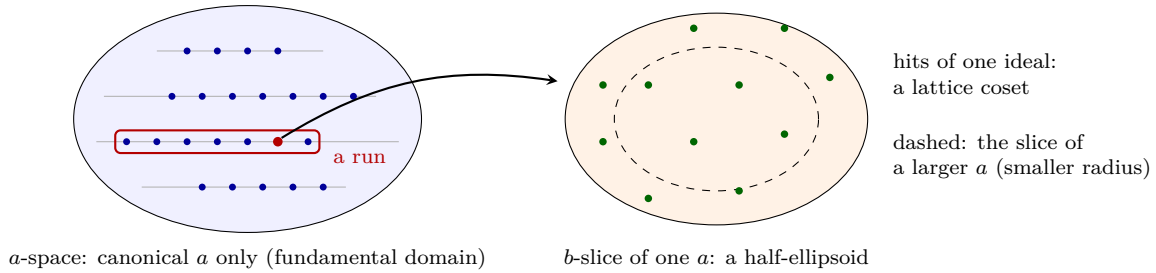


Figure 7: Anatomy of the sieve region (schematic, two dimensions out of six on each side). Left: the canonical values of a; consecutive canonical values along the first basis direction form a *run*. Right: for each a the cells are the b in a half-ellipsoid whose radius shrinks as a grows. The hits of a prime ideal in a slice form a coset of a lattice of index q; which coset depends on a, the lattice does not.

3. *The same hits occur again and again.* The region is a stack of similar slices. Where a prime ideal hits a slice depends on the slice through one residue and one radius only. Hits can be stored once and replayed.

6.2 Anatomy of the region

A cell is a pair (a, b) of elements of $\mathcal{O}_h$, twelve integers. The outer variable a ranges over the canonical representatives of Section 5.4; for each a, the *slice* is the set of admissible b (Figure 7).

Shape. The norm of an element is the product of its sizes at the embeddings of the field. Level sets of the norm are not cubes in coefficient coordinates, and a cube wastes cells in its corners, where norms are large. We use an ellipsoid $Q_a(a) \le \theta A^2$, $Q_b(b) \le \lambda A^2 - \kappa\, Q_a(a)$ for quadratic forms weighted by place. Against a cube or a box in reduced-lattice coordinates, the ellipsoid saves 1.0 to 1.6 bits of cells on BLS12-381. The scale θ is set by a pre-registered rule: the smallest region whose estimated number of relations exceeds the target by a fixed margin plus sampling error.

How the number of cells is known. The number of lattice points in the ellipsoids is measured by an acceptance experiment with exact tests (10^9 to 10^{11} trials) and bracketed by a certified count in exact rational arithmetic. The yield is a Monte Carlo sum over pools of several million uniform cells, each with its exact norms, exact powers of 2, and exact canonical and primitivity tests. Pools used to choose parameters are kept separate from pools used to state the result.

6.3 Sieve one side, test the other

The two norms are very different in size. For BLS12-381, N_f has about 738 bits and N_g about 498; for BN254, N_g has 470 and N_f 417 (Figure 1). The larger norm is far less likely to be smooth. We sieve *only that side* (f for BLS12-381, g for BN254). Its survivors are rare. On those, the other norm is tested *exactly*.

The test is a batch smoothness test [8]. Let P be the product of all primes up to B. An integer z whose prime factors have exponents below 2^e is B-smooth if and only if

$$(P \bmod z)^{2^e} \equiv 0 \pmod{z}.$$

The residues $P \bmod z$ are computed for a whole wave of survivors at once with a *remainder tree*: multiply the z's together pairwise up a tree, reduce P modulo the root, and push remainders back down. With FFT multiplication, the cost per survivor is about 2^{25} instructions, against 2^{38}

for factoring the same number with the elliptic curve method. We use a scaled variant of the tree with a proved error bound, so that every leaf is exact.

This one change was worth -0.54 bit when introduced on BLS12-381. On BN254, where the two norms are close, the rule is the same: sieving g instead of f is worth -0.36 bit.[2]

6.4 Two stages: tiny primes by arithmetic

Primes below 2^{24} make most of a sieve's hits. In the two-stage detector they are not sieved at all.

- *Stage A* sieves the primes in $[2^{24}, 2^{64})$ only. A cell passes if what is *not* explained by those primes is below a slack (356 bits on BLS12-381).
- *Stage B* takes the survivors of stage A and computes the exact part of their norm made of primes below 2^{24}, with the same remainder tree against the product P_{24} of those primes and one gcd. A survivor continues only if the remaining unexplained part fits a small number of large primes.

Stage A lets through many more cells than a full sieve would, but stage B is cheap enough per survivor (820,000 instructions) that the trade is favourable: -0.16 bit on BLS12-381. The retention of true relations through both thresholds is computed from an Euler-product model of the sieved and unsieved parts, and checked against four million integers with known factorisations.

6.5 Walking the hits

The condition "$\mathfrak{q}$ divides $a - bx$" is linear in b modulo q: for $\mathfrak{q} = (q, y - s, x - t)$ it reads $b(s) \equiv a(s)\, t^{-1} \pmod{q}$. In a slice, the hits are the points of a coset of a 6-dimensional lattice of index q inside the ellipsoid. Enumerating them is the inner loop of the whole attack. Two kernels do it.

- *Line walk* (small and medium primes). The slice is cut into lines; on each line the hits are an arithmetic progression. The walk is a fixed five-level enumeration with a tight inner loop. In the final designs it serves as the reference: Section 6.6 replaces it.
- *Record walk along a run* (large primes, above about 2^{48}). For a large prime most slices contain no hit at all, and visiting each slice costs more than the hits. But along a run of consecutive values of a, the hits of an ideal in *all* the slices of the run form one coset of a 7-dimensional lattice: the seventh coordinate is the position in the run. The walk enumerates that coset once per run and writes one small record (cell, $\log q$) per hit into a page that is applied to the counters later.

Both walks are hand-written assembly, checked hit by hit against a reference implementation on billions of hits, and charged at the envelope of their measured costs per hit as a function of hit density. Rewriting the record walk in register-allocated assembly lowered its price by 10 to 33%, depending on the hit density.

6.6 Coset replay lists

The last step removes most of the walking for small and medium primes. Look again at the hit condition:

$$b(s) \equiv \kappa \pmod{q}, \qquad \kappa = a(s)\, t^{-1} \bmod q.$$

The left side does not involve a. And every slice is the same ellipsoid, centred at the origin, cut at a radius that depends on a. Hence:

> *The hits of an ideal in the slice of a are the cells of the* largest *slice that lie in residue class κ and within the slice's radius.*

[2] Also found independently by kaankolcu.

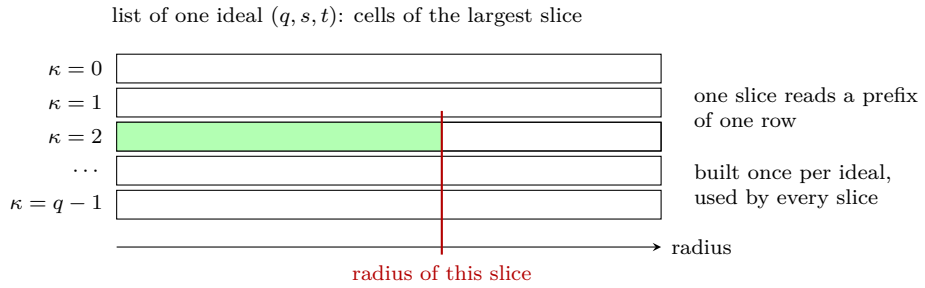


Figure 8: A replay list. Rows are residue classes of $b(s)$ modulo q; inside a row, cells are sorted by their radius in the ellipsoid. The hits of the ideal in a given slice are a prefix of the row κ selected by that slice.

So for each ideal we build, once, a list of the cells of the largest slice, grouped by residue class and sorted by radius. To sieve the ideal in a slice, we compute κ, find its group, and copy the entries up to the slice's radius into the counters (Figure 8). No lattice is enumerated; a hit costs 2.75 instructions, against 5.6 for the line walk on BN254 and more for sparse primes. The cell sums are identical to those of the walks, so yield, thresholds and everything downstream are unchanged.

A list is worth building only if it is used often. An ideal of norm q has q classes and there are about 2^{45} to 2^{63} slices, so lists pay up to a cut-off (primes below 2^{42} for BN254, 2^{48} for BLS12-381); above it the record walk remains. Building the lists costs $2^{83.4}$ and $2^{98.6}$ units, small rows of the ledgers. The gain is -0.25 and -0.26 bit.

The price is memory. The lists occupy 2^{97} bytes for BN254 and 2^{112} bytes for BLS12-381. The cost model does not charge memory, so within the model this is a genuine saving, but it is the clearest case in this paper where the model and physical reality part ways. Without the replay lists the costs of Table 1 are about 0.25 bit higher; the BLS12-381 storage then falls to about 2^{105} bytes, set by the descent, while the BN254 storage statement (2^{100} bytes) is unchanged (Section 10).

6.7 A coset sieve at the ramified prime

In the BLS12-381 tower, $2\mathcal{O}_h = P_2^2 P_4^2$ with $P_2 = w\mathcal{O}_h$ of norm 2. Pairs with both a and b in P_2 are never primitive; they are a quarter of all cells, and a sieve that visits them wastes a quarter of its work. Membership of b in P_2 is a parity condition on its coordinates, so after a unimodular change of basis a slice with $a \in P_2$ sieves only b with an odd first coordinate, on a sublattice of index 2, with all kernels unchanged. Along a run, $a \in P_2$ alternates, and the run walk splits into two cosets. Of the 0.415 bit of cells removed, 62% survived the overhead: -0.21 bit on the total.

The same local analysis prices the prime 2 exactly. Dickman's function treats a norm as a random integer, whose power of 2 is 2^1 on average. But here the power of 2 in N_f is known cell by cell: exactly 2^6 when $a \in P_2$ and $b \notin P_2$, and 2^0 otherwise. Using the exact value in the smoothness estimate, instead of the average, raises the yield by convexity (-0.05 bit), and fits a complete census of 92,108 small pairs far better than the average model. BN254 has the same structure at its prime above 2 (norm 4): N_f carries exactly 2^{12} in one fifth of the primitive cells, and the yield is priced accordingly.

6.8 From survivors to relations

A cell that passes both stages and the batch test on the other norm is almost a relation: its sieved norm still has to be split into prime ideals, and may contain a few primes just above the sieve limit.

- *Strips.* Primes in the bands $(2^{20}, 2^{48}]$ and $(2^{48}, 2^{64}]$ are removed by two more batch gcds.

- *Windows.* After the strips, a cofactor is 1, one prime below B, or a product of k primes in $(2^{64}, B]$, which confines its size to windows. The elliptic curve method [25] is run only on cofactors inside a window, with a schedule tuned for 70-bit factors ($2^{38.4}$ instructions).
- *Thresholds.* Sieve thresholds need the size of the norm in each block of cells. They come from *certified* bounds: the norm is evaluated at the embeddings with outward rounding on a hierarchy of boxes. A smooth cell is therefore never lost to a floating-point error; a loose bound only sends a few more cells to the exact test.[3]

6.9 The price of a cell

The section can be summarised in one quantity, the mean work per cell. For BN254 it is 17.3 instructions: 11.3 for the record walk of large primes, 3.1 for replayed hits, 0.8 for applying records, and 2.1 for thresholds, survivors and cofactorisation. With walks in place of replay it was 21.4; before the choice of the sieved side, the two-stage detector and the assembly walks, 70. For BLS12-381 the mean is 13.3 instructions; with walks only and a single-stage detector it was 42. The per-cell price is not a pure engineering number: the two-stage detector and the replay lists change what the sieve does, and each lower price was re-optimised against the region and the smoothness bound.

7 Filtering and linear algebra

After collection we hold slightly more relations than unknowns. This section explains how the system is shrunk (filtering), how it is solved (block Wiedemann), and three ideas that make the solve cheaper: Gaussian-integer arithmetic, subset-sum tables, and a *product form* that separates the size of the Krylov space from the cost of a product. It also states precisely which parts are proved and which are hypotheses.

7.1 Filtering, and how its outcome is predicted

Filtering works on the quotient system of Proposition 2: one column per orbit of four ideals, entries $a + b\chi$. It has three steps, as in CADO-NFS [12, 36].

- *Purge*: delete columns that occur in a single relation, with that relation, and repeat.
- *Clique removal*: delete the heaviest connected groups of relations until few more rows than columns remain.
- *Merge*: eliminate a column by adding multiples of one of its rows (the pivot) to the others. The matrix loses one row and one column and becomes denser. We use the Markowitz rule [26]: take the column that minimises $(k-2)\times$ the weight of its lightest pivot row, k being the number of rows in the column, and stop at a target mean row weight.

What matters for the cost is the ratio $\nu = N/U'$ of the final dimension to the number of orbit columns. We cannot filter 2^{64} relations, so ν is measured on synthetic systems that reproduce what purge and merge act on: each column is an ideal of relative size $t = \log q/\log B$ that occurs in a relation with the production probability $\min(1, 4\kappa_s(t)/q)$, $\kappa_s(t) = \rho(u_s - t)/\rho(u_s)$; the smallest primes, whose columns are the heavy ones and limit merging, are represented explicitly by the actual orbits of prime ideals of the tower, with their true norms; the excess and the coefficient law are those of production.

Figure 9 shows the measurements for BLS12-381. The ratio *falls* with scale at every step from 2^{16} to 2^{23}. The ledger does not extrapolate the trend: hypothesis H3 is only that production is not above the largest simulated scale, and the charge is the mean plus two standard errors

[3] The two-level organisation of these bounds is due to kaankolcu.

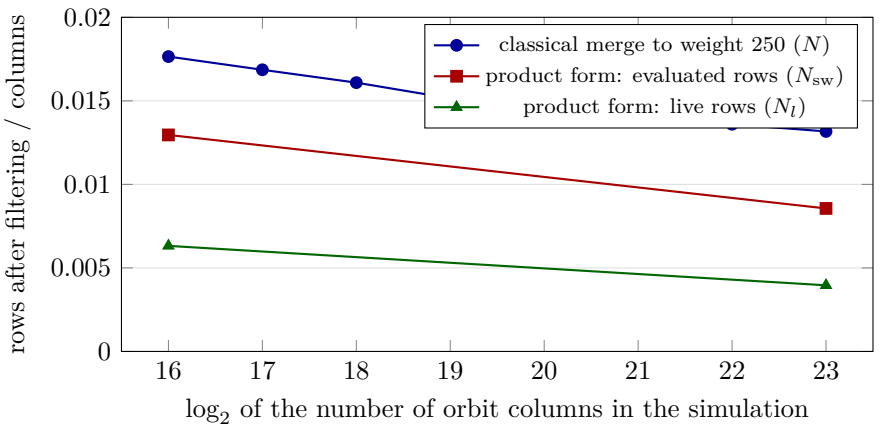


Figure 9: Filtering ratio against simulation scale, BLS12-381 profile (8 to 16 seeds per scale; for the product form only the end points are plotted, both series fall at every step in between). Production is at $2^{63.9}$ columns. Hypothesis H3 is that the ratios do not rise beyond the largest simulated scale.

at 2^{23}, times a factor 1.0159 for the spread of the explicit band. A run whose matrix exceeds a slightly higher cap aborts and counts as a failure, with a probability bounded by Chebyshev's inequality from the measured spread. This is still a 40-bit extrapolation and remains the main modelling risk of the linear algebra (Section 10).

7.2 The matrix–vector product

Block Wiedemann multiplies the matrix by vectors of elements of $\mathbb{F}_r$, about N times. The price of one row of one product is what the ledger needs.

Gaussian entries. An entry is $a + b\chi$ with small integers a, b; after merging, at least 86% of the entries are units $\pm 1, \pm\chi$. Each vector element x_j is stored together with χx_j in one 320-byte slot, so a unit entry is one multi-word addition (charged 8.29 instructions per word-group under a measured mix), and computing χx_j once per element costs one multiplication by a fixed constant. Because χ has a 192-bit representative, this multiplication is cheap: about 200 instructions in a Montgomery [29] kernel, and 133 with a truncated precomputed-quotient variant.[4] A row ends with one exact reduction modulo r (the "row tail", 572 instructions with four memory owners).

Subset-sum tables for the dense columns. Columns of very small ideals are dense: they occur in most rows. For each product, the solver first tabulates all sums of two, three or four elements among the columns of norm below 2^{28}, $2^{18.25}$ and 2^{14} (with their χ-multiples and negatives). A row's entries in that band are then grouped greedily, and each group costs one addition instead of several. The grouped rows cost 0.80 of the plain ones at the simulated scales; the ratio rises slowly with scale, and the ledger charges 0.84, the fitted trend plus two standard errors carried to production.

Memory owners. One 64-bit address space holds $2^{64}/320$ vector elements, fewer than N. The vector is split among several *owners* (three or four), and the cost of reaching another owner's memory is part of the row tail. The model charges instructions, not bytes, but it does not allow an address to exceed 64 bits for free.

[4] The latter is due to soispoke.

7.3 Product form: a small Krylov space with cheap products

Merging trades size for density: every eliminated column removes a row but makes the pivot's neighbours heavier. Block Wiedemann costs about (number of products) $\times$ (cost of a product) $\approx N \times (N \cdot \text{weight})$, and the classical optimum merges until the two effects balance, here at weight 250.

The *product form* decouples the two factors. We stop the classical merge early, at weight 200, with N_{sw} rows. We then continue to eliminate columns, but instead of substituting the pivot row into the other rows, we *record* it: the eliminated unknown is defined by its pivot row in terms of other unknowns. If the recorded rows are ordered so that each one only reads unknowns defined before it, they form a unit-triangular system, which can be applied to a vector in one pass. What remains is a *live* system of N_l rows. A product now

1. runs the recorded rows in order, filling their vector slots;
2. then applies the N_l live rows.

Rank and kernel are unchanged, so the algorithm iterates on a space of dimension N_l while each product costs N_{sw} rows *of the lighter matrix*. In the BLS12-381 ledger, $N_l \le 2^{56.00}$ and $N_{\text{sw}} \le 2^{57.10}$, against $N \le 2^{57.7}$ at weight 250 for the classical form: the number of products falls by a factor 3.3 and each product costs about half. Recording pivots is the classical idea of keeping an elimination in factored form; in the NFS context it appears in the filtering of Boudot [9]. The ordering rule, the label scheme that makes the recorded system triangular, and its pricing are ours. The gain on the total was -0.08 bit on BLS12-381 and -0.12 on BN254.

7.4 Block Wiedemann with explicit premises

The live system is $A = [M \mid S]$ with $N = N_l$ orbit columns and the dense Schirokauer columns S. Block Wiedemann [13] computes the sequence $a_i = X^T B^i Y$ for a random block X of m vectors and a block Y of n vectors, with $B = [M \mid 0]$; the Schirokauer columns enter as the first columns of Y, so no product ever multiplies by a dense column. A linear generator of the sequence [37] gives a kernel vector. For BLS12-381 the block sizes are $(m, n) = (8192, 130)$. A large m makes the sequences short; the X blocks are block-diagonal so that each vector element meets a single entry of X, which keeps the projections cheap.

What is proved and what is assumed. It is tempting to identify the growth of the Krylov space with the invariant factors of the matrix. That is wrong: invariant factors describe the operator, while the finite Krylov ranks depend on the starting block. A 12×12 diagonal matrix with distinct eigenvalues has one invariant factor of degree 12, yet a three-column start reaches ranks $3, 6, 9, 12$; the identity matrix never expands a starting space at all. Generic results for dense random projections [20] do not cover projections restricted to a block-diagonal support, nor fixed starting columns. We therefore state the requirement as two explicit premises and one conditional theorem.

H-R (saturation). The relation matrix has full column rank N, and A has a one-dimensional kernel, the line of virtual logarithms, whose Schirokauer part is nonzero.

H-X (finite lengths). At the *charged* sequence lengths d and e, the control matrix $C_d = [BZ, BY, \ldots, B^d Y]$ has rank N and the observation matrix $(X; XB; \ldots; XB^{e-1})$ is injective on $\operatorname{span} S + \operatorname{Im} B$, for the actual sparse support of X and the fixed columns S.

Conditional theorem. If the two minors of H-X are nonzero *polynomials* in the random entries, then they are nonzero at a random point except with probability at most $(2N + s - 1)/Q$, with entries drawn from $Q \ge 2^{128}$ values (Schwartz–Zippel): below 2^{-66}. Under H-R and H-X, a generator of the sequence then lifts to the kernel vector by one Horner pass.

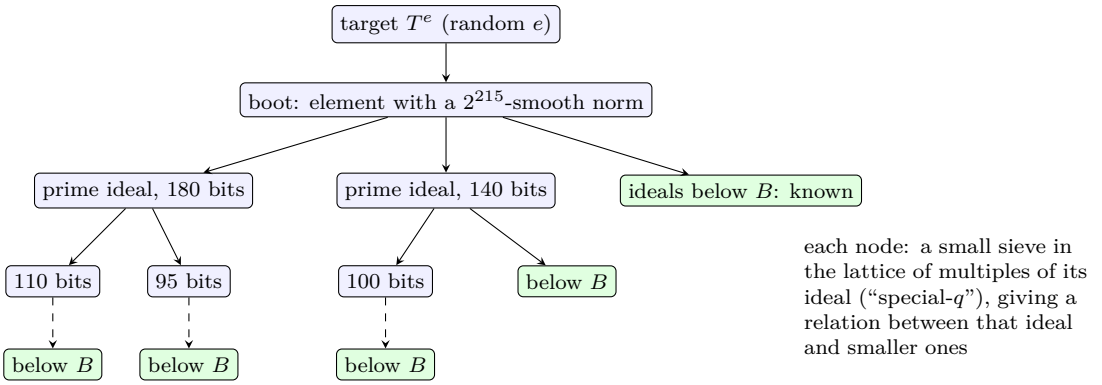


Figure 10: The descent. The boot writes the (randomised) target as a product of prime ideals of moderate size. Each ideal that is still too large is a node: a small sieve finds a relation that expresses it through smaller ideals, its children. Leaves are factor-base ideals, whose logarithms are known. Sizes are illustrative.

The lengths are $d = \lceil D/n \rceil + 1 + \lceil D/128n \rceil$ and likewise e in m, with $D = N + s$: a margin of 2^{-7} over the minimal lengths is charged. H-X is a hypothesis about the production matrix and is not proved. The evidence for it is experimental: real NFS matrices of dimension 2,400 to 4,000 from quartic and quintic fields with seven real Schirokauer maps, with the same projection support and the charged lengths, reach full ranks and recover the kernel in every case, with m up to 2048 and n/m down to 1/64; a control in which the pad coordinates share one projection block fails, as it should. A failed solve is detected by checking the kernel vector on relations that filtering dropped, and counts as a failure of the run.

Generator. The linear generator is computed by a matrix Berlekamp–Massey algorithm with half-gcd recursion and FFT-based polynomial products. With large m its cost is not negligible in general, but here it is $2^{100.2}$ units for BLS12-381, a thousand times below the products.

7.5 Schirokauer maps

The dense columns are the χ-eigenpart of the classical maps: with $R = (\mathbb{Z}/r^2)[y,x]/(h,f)$ and $\epsilon = \operatorname{lcm}(r^{d_i} - 1)$, the columns are fixed random $\mathbb{F}_r$-linear forms of $\pi_\chi\big((\varphi^\epsilon - 1)/r\big)$, where $\pi_\chi = \frac{1}{4}\sum_{k<4} \chi^{-k}\tau^k$ is the projection on the eigenspace. For BLS12-381, ϵ has 18,360 bits on the f side, and one map evaluation costs below 2^{36} instructions per relation. That the forms are injective on the χ-part of the units modulo r-th powers is a hypothesis (H-SM, part of H3); it is the standard assumption behind every use of Schirokauer maps.

8 Individual logarithms and the probability of success

After the linear algebra, the logarithms of all factor-base ideals are known. The target is an arbitrary element of $\mathbb{G}_T$, and it must be connected to them. In the toy example of Section 2.2 this was one line; here it is a tree.

8.1 The descent tree

Boot. The target T is randomised to T^e with e uniform in $[1, r)$. The boot enumerates short vectors of a lattice of determinant p^6 (pairs (a, b) whose image in $\mathbb{F}_{p^{12}}$ is proportional to the target) and looks for one whose norm is smooth up to a few larger primes: all primes below 2^{n_1}, at most k_A in a window $(B, 2^{n_A}]$ and at most one above. Its cofactorisation uses the elliptic curve method [25].

Nodes. Each prime ideal $\mathfrak{q}$ above the factor base becomes a *special-q node*: the pairs (a, b) divisible by $\mathfrak{q}$ form a lattice; the node sieves a small ball in a reduced basis of that lattice and cofactors the surviving cells. A node relation has all remaining primes below a bound that depends on the size of $\mathfrak{q}$; the primes above the factor base are the node's children. Nodes are grouped in bins by size (62 types for BLS12-381).

The plan. For every bin, the plan fixes the smoothness bound, the two windows, the number of large primes allowed and the ball radius. Loose settings find relations easily but create many expensive children; strict settings do the opposite. The plan is found by a coordinate-descent search with random restarts on the expected-cost recurrence below. It is the main lever on the descent: re-optimising it moved the BLS12-381 descent by 1.9 bits early in this work; on BN254, planning, a larger factor base and cheaper node tests together took the descent from $2^{95.8}$ to $2^{84.3}$ units (Section 9).

8.2 Cost and failure of a node

For a node, μ is the expected number of distinct relations: cells $\times$ a two-window semismooth probability $\times$ the unit-duplicate factor $\times$ ECM detection $\times$ sieve retention. The semismooth probability is a *lower* Riemann sum of the smoothness integral over measured norm bins. Relation counts are modelled as negative binomial with mean μ, which is more dispersed than Poisson. If a child subtree fails with probability φ and ψ is the sum of the children's φ in a relation, then a node fails when none of its relations has an all-successful subtree:

$$\varphi = \max\left((1 + \mu(1-\psi)/r_{\mathrm{NB}})^{-r_{\mathrm{NB}}} + \mathrm{DISC},\ 0.02\right),$$

and it tries $1/(1-\psi)$ relations on average. The recurrence gives the expected work and the expected number of nodes of the whole tree.

A per-node certificate. Each cell of a node gets a *certified upper bound* on both norms, by outward-rounded evaluation at the complex embeddings. From these bounds the node computes its own heuristic yield and discards itself, as a failure, if that yield is below what its bin was charged. The bound is rigorous; the yield it feeds is still the smoothness heuristic. Over 3,100 production-size ideals the discard rate was at most 0.001, which is charged as DISC.

Missing logarithms. Filtering deletes ideals; their logarithms are recovered afterwards by back-substitution, except those that occur in at most two relations, which the descent treats as unknown (it must avoid them or descend them). This loss is priced with a Poisson model of occurrences for BLS12-381; for BN254, where it weighs more, the logarithms are reconstructed by a peeling process seeded with the most frequent ideals, whose yield is simulated and carried to production size as part of H5.

8.3 Success probability

The run succeeds unless one of the following happens. The figures are those of the BLS12-381 ledger; BN254 is similar.

- *Main phase.* The solver fails with probability below 2^{-66} under H-R and H-X (Section 7.4); the filtering cap is exceeded with probability at most $2.2 \cdot 10^{-4}$.
- *Boot.* All boot relations of a logarithm lead to failed subtrees: at most $6.2 \cdot 10^{-9}$ each.
- *Meter.* The descent runs under a work meter set far above its expectation ($2^{121.9}$ against $2^{108.9}$ units); by Markov's inequality it is reached with probability at most $1.2 \cdot 10^{-4}$.
- *Node counter.* More than $2^{27.3}$ nodes: at most $2^{-12.8}$.

- *Repeated ideals.* The same special-q ideal met twice, which would break independence: at most $2^{-10.7}$.
- *Degenerate randomiser, field map*: $2/r$ and 2^{-48}.

Together, $P_{\text{run}} \geq 0.9989$ for BLS12-381 and ≥ 0.9985 for BN254. The independence of distinct subtrees and of the order in which relations are tried is hypothesis H5. We tested it at reduced scale: children met in production code have the same norm distribution as fresh samples (differences -0.12 ± 0.18 and $+0.04 \pm 0.06$ bit), and in toy one-level descents the failure rate after a sibling's failure (0.0179) is not above the overall rate (0.0324).

Why the descent weighs differently on the two curves. For BLS12-381 the descent is 5% of the run and for BN254 now 4%. But for BN254 the first ledgers were *dominated* by it ($2^{95.8}$ against $2^{91.4}$ for collection), because the factor base is much smaller relative to the norms a boot must split. The remedy was a larger factor base (2^{58} instead of the 2^{56} that collection alone would choose), cheap exact tests inside the nodes, and a re-optimised plan: the smoothness bound must be chosen for the whole attack, not for collection and linear algebra only.

9 Two instantiations: BLS12-381 and BN254

Sections 4 to 8 described a method. This section applies it to two curves, gives the resulting ledgers, measures what each technique contributes, and explains why neighbouring designs do not do better.

9.1 Parameters

To instantiate the method on a curve with $p = F(u)$ and embedding degree 12, one chooses, in order:

1. a twisted base field (h, σ, c, w) satisfying Proposition 1 with h inert and $N(w)$ a non-residue modulo p, by a search over sextic fields ranked by a quick yield estimate;
2. the fundamental domain and the region forms Q_a, Q_b from the embeddings of w;
3. the smoothness bound B and the region scale, jointly with the filtering and descent models;
4. the sieve organisation: which side is sieved, the stage-A range, the replay cut-off;
5. the solver shape (m, n) and the product-form merge weight;
6. the descent plan.

Table 7 gives the outcome for the two curves.

The two instances differ in instructive ways. For BLS12-381, F has degree 6, so f has degree 12 in x and its norm is much larger than that of g: the attack is dominated by the smoothness of N_f. For BN254, F has degree 4 with coefficients up to 36; the two norms are close, and it is g, with its 63-bit coefficient u, that is larger. The BLS12-381 tower has a ramified prime of norm 2 that allows a coset sieve. For BN254 the prime above 2 has norm 4: the never-primitive cells are one in sixteen instead of one in four, and skipping them does not pay.

9.2 Ledgers

Table 8 and Figure 11 give the ledgers. In both, collection is about 85% of the work. This balance is itself a result. In an operation-count model with a fixed filtering ratio, the optimum equalises collection and linear algebra. With the measured ratios, the twisted quotient and the product form, linear algebra is cheap enough that the optimum moves to a larger B, until collection dominates.

Table 7: Parameters of the two instantiations. Fields and polynomials are in Table 5.

	BLS12-381	BN254
Field size $\log_2 p^{12}$ (bits)	4569	3043
Mean norm sizes N_f, N_g (bits)	738, 498	417, 470
Smoothness bound B	$2^{70.5}$	2^{58}
Orbit columns U'	$2^{63.92}$	$2^{51.71}$
Sieved side; other side	f; batch test	g; batch test
Replay lists; record walk	$[2^{24}, 2^{48})$; $[2^{48}, 2^{64})$	$[2^{14}, 2^{42})$; $[2^{42}, 2^{57})$
Cells	$2^{123.53}$	$2^{98.92}$
Instructions per cell	13.3	17.3
Relations (estimate; target)	$2^{64.11}$; $1.05\,U'$	$2^{51.77}$; U'
Live rows N_l; evaluated rows N_{sw}	$2^{56.00}$; $2^{57.10}$	$2^{43.93}$; $2^{45.10}$
Row weight; block sizes (m, n)	200; $(8192, 130)$	200; $(1024, 40)$
Schirokauer columns	10	7
Success probability of a run	≥ 0.9989	≥ 0.9985
Storage (bytes)	$< 2^{112}$	$< 2^{100}$

Table 8: Ledgers, $\log_2$ units (one unit = 20,000 instructions). Every row is an upper bound under the hypotheses of Table 10; the descent is its expected work.

Row	BLS12-381	BN254
Relation collection	112.99	88.73
Replay lists, built once	98.63	83.40
Linear algebra (matrix products)	109.95	85.28
Block Wiedemann generator	100.17	80.29
Two individual logarithms	108.86	84.27
Factoring the kept relations (two rows for BLS12-381)	101.35; 96.5	78.45
Canonical list of a	92	76.50
Schirokauer maps	88	72.40
Everything else (factor base, filtering, bases, wrapper)	< 81	< 71
Total	**113.23**	**88.95**
in instructions	$2^{127.5}$	$2^{103.2}$

Reading the totals. $2^{127.5}$ instructions for BLS12-381 is 9.6 bits below Pollard's rho in the same metric and 9.8 bits below the baseline of Section 3.4. $2^{103.2}$ instructions for BN254 agrees with the published level of about 100 bits once operations are priced (Section 3.4). The correction of Section 10.3, which these ledgers do not include, would raise both totals by an estimated 0.1 to 0.2 bit.

9.3 What each technique contributes

Table 9 lists the techniques with their measured effect. Each figure is a difference between complete ledgers with and without the technique, with dependent parameters re-optimised; some rows add up several successive refinements of one technique, and the walk-kernel row includes kernels that later techniques replaced. Effects measured at different stages of the design are not exactly additive, and the first row is the ablation of Section 4.4.

Three patterns stand out.

- *No single technique dominates.* The largest algebraic step is under one bit. The 9.8 bits between the baseline and the BLS12-381 ledger are the sum of some twenty steps of 0.05 to 1 bit, of the measurement at scale of quantities that the baseline bounded crudely (filtering ratios, kernel prices, lattice-point counts), and of the retuning of parameters that each step allows.

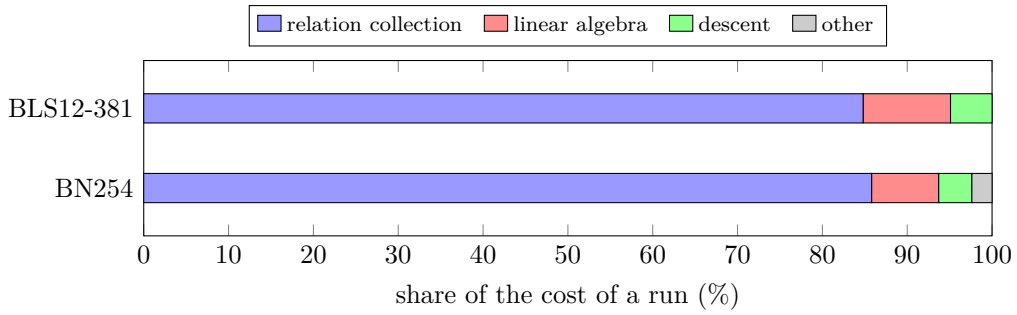


Figure 11: Where the work goes. For BN254, "other" is mostly the construction of the replay lists.

Table 9: Effect of each technique on the total, in bits (negative is cheaper). Columns: the phase heading of Section 2.6; the curve on which the difference was measured.

Technique (section)	heading	BLS12-381	BN254
Twisted tower: orbits of four, Gaussian system (4)	A, C, D	−0.94	
Fundamental domain instead of a cube, at equal relations (5)	A	−0.5	
Ellipsoid region instead of a box, in cells (6.2)	A	−1.0 to −1.6	
Coset sieve at the ramified prime; exact power of 2 (6.7)	A	−0.21; −0.05	n/a
One-sided sieve with batch test (6.3)	B	−0.54	−0.36
Two-stage detector (6.4)	B	−0.16	−0.07
Record walk along runs, extended to all large primes (6.5)	B	−0.17	
Hand-optimised walk kernels, both generations (6.5)	B	−0.67	−0.31
Record pages: emission in the walk, in-place apply, layout per tail	B	−0.16	−0.07
Coset replay lists (6.6)	B	−0.26	−0.25
Markowitz merge on the quotient system; Gaussian kernels (7.2)	C, D	−0.18	
Schirokauer columns as right-hand sides; large m (7.4)	D	−0.08	
Subset-sum tables (7.2)	D	−0.06	−0.08
Product form (7.3)	C, D	−0.08	−0.12
Larger B once linear algebra is cheap ($2^{69} \to 2^{69.5} \to 2^{70.5}$)	A, C	−0.07; −0.17	
Descent planning, node tests, larger B (8): change of the descent row	–	−11.6	−11.5

- *Gains compound through B.* A cheaper matrix product does not only lower the linear algebra row; it moves the optimal smoothness bound up, which lowers collection. The row "larger B" is this indirect effect of the linear algebra techniques, and it is as large as their direct effect.
- *The techniques transfer.* Every row measured on both curves has the same sign and a similar size, despite very different norm profiles. This is the main evidence that the method is not tuned to one instance.

9.4 Why not a neighbouring design?

We examined the natural alternatives. Three of them are worth recording, because they are what a reader would try first.

A larger symmetry group. The symmetry group of a special $(6, 2)$ pair embeds in $\mathrm{Gal}(\mathbb{F}_{p^{12}}/\mathbb{F}_p)$: it is cyclic of order dividing 12, and order 8 is impossible. With no field penalty, order 12 would be worth 1.5 bits on BLS12-381. But the order must come from automorphisms of the sextic field, and the sextic fields with automorphisms of order 3 or 6 that satisfy the twist condition have larger discriminants (4 to 90 times that of our field), hence larger norms. For BLS12-381 the best order-6 design ($D = 722{,}000$) is 1.0 bit worse than the order-4 design and the best order-12 design ($D \approx 1.6 \cdot 10^7$) is 6.3 bits worse. For BN254, an order-12 design on the

cyclic sextic of conductor 35 came out 0.4 bit worse in a full pricing; order-6 designs look 0.3 bit better on a three-row model, but they need no *fewer* cells, and their filtering, unit domain (rank 5) and kernels were not validated, so we do not count this as a gain. The order-4 twist sits where the field penalty is still smaller than the gain.

Another tower shape or polynomial pair. We priced the other tower shapes and conjugation-type pairs. For BLS12-381, $(12, 1)$ is 4.6 bits worse, $(4, 3)$ far worse, and $(3, 4)$ and $(2, 6)$ admit no valid pair of this kind. For BN254 every alternative shape is at least 9 bits worse, and rescaled or translated seed polynomials $F((z - s)/t)$ are 0.7 to 2 bits worse. The special $(6, 2)$ shape with the family's own polynomial is the only one that balances the two norms.

Special-q sieving for collection. Factoring records organise collection by special-q. Here the sieve is 12-dimensional and the fundamental domain is not a lattice-friendly set. We measured 12-dimensional walk kernels for BN254: a hit costs 4.3 to 6.3 times what it costs in the 7-dimensional run walk, more than the smaller norms give back. Large-prime variations behave similarly: with a one-sided sieve and batch tests, B already plays the role of the large-prime bound.

A recurring trap. Two apparent improvements, a polynomial variant and a region shape, turned out to be artefacts of the same kind: their cells contained more unit multiples (Section 5.4). Any comparison of designs must be made on duplicate-free cells; this is the practical content of Lemma 3.

10 Hypotheses, validation and limitations

10.1 The hypotheses

Table 10 lists the hypotheses on which the ledgers rest. They are of three kinds: number-theoretic heuristics that every NFS estimate makes (H1, H-SM), extrapolations from reduced scale (H2, H3, H4), and independence assumptions (H5, H-X).

10.2 What has been checked

- *Algebra.* Field discriminants, signatures, the identities $c\,\sigma(c) = -1$ and $\sigma(w) = c^2 w$, the validity of g and the action of τ as Frob^3 are checked by scripts in the artifact and were re-checked separately in PARI/GP and SageMath.
- *Kernels.* Every charged price is a count of executed instructions (Callgrind [31], scalar AArch64, no SIMD) on code checked against a reference by in-order traces, brute force and negative controls (Appendix A).
- *Duplicate correction.* Recomputed with separately written scripts: uncapped unit searches, a 10^9-point stratified census and brute force on small cubes.
- *Filtering.* The simulator's explicit band was checked against the actual ideals of the towers.
- *Sampling discipline.* Seeds, region-sizing rules, excess rules and merge weights were fixed before the samples used to state results were drawn.
- *Independence.* Most of the engineering and of the verification code was written by AI coding agents (mostly Claude models in Claude Code) in separate sessions, under the author's direction. That is independence of code, not of people. The ledgers also went through an automated review, which found real errors and missed others; passing it is not evidence of correctness. No external expert has reviewed the full ledgers.

Table 10: Hypotheses of the ledgers. The last column says what supports each one.

	Statement	Support
H1	Norms are B-smooth with probability $\rho(u)$, u from the exact norm of each sampled cell, Murphy's α and the exact power of 2.	Standard; toy census of 92,108 pairs. Optimistic by 4 to 5% per side (Section 10.3).
H1′	Sieve thresholds retain smooth cells as an Euler-product model of the sieved and unsieved parts predicts.	Four million integers with known factorisations; 8,000 exact norms.
H2	Uniform pools give unbiased yield and work; lattice-point counts as measured.	Pre-registered pools; counts bracketed in exact arithmetic; allowances of 2 standard errors.
H3	Filtering ratios at production are not above those at the largest simulated scale (2^{23} to 2^{26} columns against 2^{52} to 2^{64}); a one-dimensional kernel; class numbers prime to r.	Ratios fall at every doubling (Figure 9); abort above a cap.
H-SM	Schirokauer maps are injective on the χ-part of the units modulo r-th powers.	Standard.
H4	Kernel prices measured at reduced scale hold at production densities; share of unit entries in merged rows; atomic add-to-memory counted as one instruction.	Dynamic instruction counts per density stratum, +2 standard errors.
H-EA, H-FFT	The survivor law prices cofactorisation; large integer products cost at most a stated $N \log^2 N$ bound.	Measured to 2^{26} limbs.
H-R, H-X	Relation matrix of full rank; finite-length Krylov conditions on the actual projection support (Section 7.4).	Real matrices of dimension up to 4,000; not proved.
H5	Descent: negative-binomial relation counts, independent subtrees, laws for missing logarithms.	Reduced-scale conditioning tests (Section 8.3).

10.3 A correction not yet applied: smooth numbers near x

Dickman's $\rho(u)$ is the proportion of B-smooth integers *up to* x. A norm is an integer *near* x, and the density of smooth integers near x is smaller: differentiating de Bruijn's expansion of the counting function gives

$$\text{density near } x \;\approx\; \rho(u) - \gamma\,\frac{\rho(u-1)}{\ln x}, \qquad \gamma = 0.5772\ldots$$

Against exact counts of smooth integers at small sizes, plain ρ is 9 to 15% too high; the corrected form is within 2% for $u \le 4$ and somewhat too low at $u = 5$ and 6, so it is not a bound either, and we have no exact data at our operating points (u between 7 and 10.5). There the formula lowers the smoothness probability by 4 to 5% on each side. Applied row by row to earlier ledgers with the same yield model, it lowers the relation count by 0.12 bit (BLS12-381) and 0.14 bit (BN254), which is the whole of the safety margin with which the regions are sized; restoring the margins by enlarging the regions raised those ledgers by 0.1 to 0.2 bit.

The ledgers of this paper do not include this correction, and we have not recomputed it on them. We state it because it goes against us: the totals of Table 1 are probably low by 0.1 to 0.2 bit. The published estimators [4, 17] use plain ρ as well.

10.4 Limitations

- *No end-to-end computation.* Nothing has been run end to end with these towers, even at toy size. A reduced computation of the same shape, with unit canonicalisation, orbit quotient, Schirokauer maps, product-form filtering, block Wiedemann at the charged lengths and a

descent, ending in checked logarithms, would test H1, H3, H5, H-R, H-X and Proposition 2 together. It is the most valuable next step.

- *Memory.* The model does not charge memory. The final ledgers assume below 2^{100} bytes (BN254) and 2^{112} bytes (BLS12-381), dominated by the replay lists of Section 6.6; without them, about 2^{105} bytes for BLS12-381, set by descent nodes. Any model that charges storage or access time would rank the techniques differently, and the replay lists first of all. The same caveat applies, less dramatically, to the large block size m of the solver and to the subset-sum tables.
- *Extrapolation.* Filtering is simulated at up to 2^{26} columns and applied at up to 2^{64}. Kernel prices are measured at up to about 2^{30} hits per walk. Yields come from pools of a few million cells.
- *Smoothness.* H1 at $u \approx 10.5$ for 740-bit norms is untested at scale, and the open correction of Section 10.3 moves the totals by 0.1 to 0.2 bit.
- *Instructions are not time.* The metric ignores latency, parallel communication and energy. It forbids SIMD, which real implementations would use; the per-cell kernels would benefit most.
- *The ablation of the twist is not final.* It was measured at an intermediate stage (Section 4.4) and not repeated on the final ledgers.
- *Scale.* 2^{103} instructions is more than 10^{13} core-years. **Neither curve is under practical threat from these results.** They refine an accounting; they do not lower an established security level by more than the accounting already implied.

10.5 Other curves

The techniques divide into those that carry over unconditionally and those that need a search.

- *Unconditional.* Counting distinct relations and sieving a fundamental domain; the one-sided sieve with batch tests; the two-stage detector; replay lists; product-form linear algebra; the descent planning; the itemised ledger.
- *When* $4 \mid k$. $\chi = p^{k/4} \bmod r$ satisfies $\chi^2 = -1$ whenever $\mathrm{ord}_r(p) = k$. If the base field admits a twist (Proposition 1), the system is Gaussian.
- *Per curve.* The twist needs a base field with an involution σ, a unit c with $c\,\sigma(c) = -1$ and w with $\sigma(w) = c^2 w$, such that h is irreducible modulo p and $N(w)$ is a non-residue. The field found for one curve does not serve another: the BLS12-381 sextic is not inert for BN254, BLS12-377 or BLS12-446. For BLS12-377, $p \equiv 1 \bmod 4$ excludes the small-norm choices of w; we have only a model-level port for that curve and give no figure.

Curves chosen to keep $\mathbb{G}_T$ just above a target level, such as the larger BLS12 and BLS24 curves of [16], have little margin against a one- or two-bit refinement; they should be re-examined with duplicate-free counting first, since that correction goes in their favour.

11 Conclusion

We set out to count the instructions of a complete special tower NFS attack on the target groups of BLS12-381 and BN254. The answer, $2^{127.5}$ and $2^{103.2}$ instructions per run under the hypotheses of Table 10, is less interesting than what it takes to obtain it.

Operation counts and instruction counts are different objects. The published design for BLS12-381, completed with standard components and charged in full, is level with Pollard's rho. Almost ten bits then separate it from the ledger of this paper, and most of them lie in places

that operation-count models treat as constants: the work per sieved cell, the filtering ratio, the structure of a matrix product, the descent. A security estimate that fixes those constants is a statement about a model, and should be read as such.

The algebra matters in two directions. The twisted tower shows that the special form of p and a symmetry of order four are compatible, and that the linear algebra can then be done over the Gaussian integers with a minimal set of Schirokauer maps. The unit-duplicate lemma shows that a standard way of estimating yields overcounts, by a factor of seven in the one published case we measured. The first makes attacks cheaper and the second makes estimates more expensive; estimates for other curves should be re-examined for both.

The method is not specific to one curve. Every technique was stated for a general special $(6, 2)$ pair and carried from BLS12-381 to BN254 with gains of the same sign and similar size. What must be redone per curve is a search for the twisted base field and the choice of parameters.

Two steps would add more than further tuning. The first is an end-to-end computation on a small field of the same shape, with the same fundamental domain, orbit quotient, product-form solver and descent, to test the hypotheses together. The second is a cost model that charges memory: under it the designs of this paper would have to be re-optimised, and some of their techniques would not survive.

Acknowledgements. This work was developed inside the CurveSmash challenge [23, 24], whose cost model and automated review shaped it, alongside other participants whose public packages we built on and who built on ours. In particular: the starting point of our BLS12-381 ledgers after the baseline was a package by kaankolcu with contributions by mpjunior92 and bernardo-lima12; kaankolcu contributed the two-level certified thresholds, a rescaling of the b-ellipsoid, and (independently of us) the choice of the sieved side for BN254; dadon a record-walk kernel and descent plans for BN254; kshitij-hash and dadon filtering measurements at larger scales; soispoke the truncated-quotient multiplication by χ and a deterministic factor listing for kept relations; yukonautominer the merged carry chain; mpjunior92 the large block size of the solver; kaankolcu, dadon and mpjunior92 the convention of charging the descent at its expectation under a work meter; kaankolcu and kshitij-hash the lighter merge weights. The polynomial selection follows Guillevic and Singh, whose public code made the comparison possible, and Barbulescu and Duquesne; the filtering follows CADO-NFS [36]. The engineering was carried out with AI coding agents (Claude models in Claude Code, and other agents in the packages we built on) under the author's direction.

References

[1] Haetham Al Aswad, Cécile Pierrot, and Emmanuel Thomé. High-order Galois automorphisms for TNFS linear algebra. In *Advances in Cryptology – CRYPTO 2026*, 2026. Cryptology ePrint Archive, Paper 2026/560, version of 2026-06-19.

[2] Diego F. Aranha, Georgios Fotiadis, and Aurore Guillevic. A short-list of pairing-friendly curves resistant to the special TNFS algorithm at the 192-bit security level. *IACR Communications in Cryptology*, 1(3), 2024. Cryptology ePrint Archive, Report 2024/1223.

[3] Razvan Barbulescu. How to get rid of units. Slides, CATREL workshop, 2015.

[4] Razvan Barbulescu and Sylvain Duquesne. Updating key size estimations for pairings. *Journal of Cryptology*, 32(4):1298–1336, 2019. doi: 10.1007/s00145-018-9280-5. Cryptology ePrint Archive, Report 2017/334.

[5] Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Improving NFS for the discrete logarithm problem in non-prime finite fields. In *Advances in Cryptology*

– *EUROCRYPT 2015, Part I*, volume 9056 of *LNCS*, pages 129–155. Springer, 2015. doi: 10.1007/978-3-662-46800-5_6.

[6] Razvan Barbulescu, Pierrick Gaudry, and Thorsten Kleinjung. The tower number field sieve. In *Advances in Cryptology – ASIACRYPT 2015, Part II*, volume 9453 of *LNCS*, pages 31–55. Springer, 2015. doi: 10.1007/978-3-662-48800-3_2.

[7] Paulo S. L. M. Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography – SAC 2005*, volume 3897 of *LNCS*, pages 319–331. Springer, 2006. doi: 10.1007/11693383_22.

[8] Daniel J. Bernstein. How to find smooth parts of integers. Preprint, 2004.

[9] Fabrice Boudot. On improving integer factorization and discrete logarithm computation using partial triangulation. Cryptology ePrint Archive, Paper 2017/758, 2017. URL https://eprint.iacr.org/2017/758.

[10] Sean Bowe. BLS12-381: New zk-SNARK elliptic curve construction. Electric Coin Company blog, 2017.

[11] Vitalik Buterin and Christian Reitwiessner. EIP-197: Precompiled contracts for optimal ate pairing check on the elliptic curve alt_bn128. Ethereum Improvement Proposals, https://eips.ethereum.org/EIPS/eip-197, 2017.

[12] Stefania Cavallar. Strategies in filtering in the number field sieve. In *Algorithmic Number Theory – ANTS-IV*, volume 1838 of *LNCS*, pages 209–231. Springer, 2000.

[13] Don Coppersmith. Solving homogeneous linear equations over GF(2) via block Wiedemann algorithm. *Mathematics of Computation*, 62(205):333–350, 1994.

[14] Gabrielle De Micheli, Pierrick Gaudry, and Cécile Pierrot. Lattice enumeration and automorphisms for tower NFS: A 521-bit discrete logarithm computation. *Journal of Cryptology*, 37(1):6, 2024. doi: 10.1007/s00145-023-09487-x.

[15] Karl Dickman. On the frequency of numbers containing prime factors of a certain relative magnitude. *Arkiv för Matematik, Astronomi och Fysik*, 22A(10):1–14, 1930.

[16] Aurore Guillevic. A short-list of pairing-friendly curves resistant to special TNFS at the 128-bit security level. In *Public-Key Cryptography – PKC 2020, Part II*, volume 12111 of *LNCS*, pages 535–564. Springer, 2020. doi: 10.1007/978-3-030-45388-6_19. Cryptology ePrint Archive, Report 2019/1371.

[17] Aurore Guillevic and Shashank Singh. On the alpha value of polynomials in the tower number field sieve algorithm. *Mathematical Cryptology*, 1(1):1–39, 2021. Cryptology ePrint Archive, Report 2019/885.

[18] Aurore Guillevic and Shashank Singh. alpha: Experimental Magma and SageMath implementation of the alpha function and TNFS simulation. https://gitlab.inria.fr/tnfs-alpha/alpha, commit 57702fa39de6bf3413460f150f184ee5d009a2f3, 2026.

[19] Aurore Guillevic, Simon Masson, and Emmanuel Thomé. Cocks–pinch curves of embedding degrees five to eight and optimal ate pairing computation. *Designs, Codes and Cryptography*, 88:1047–1081, 2020. doi: 10.1007/s10623-020-00727-w.

[20] Seung Gyu Hyun, Vincent Neiger, Hamid Rahkooy, and Eric Schost. Block-Krylov techniques in the context of sparse-FGLM algorithms. *arXiv preprint*, 2019. URL https://cs.uwaterloo.ca/~eschost/publications/HNRS.pdf.

[21] Taechan Kim and Razvan Barbulescu. Extended tower number field sieve: A new complexity for the medium prime case. In *Advances in Cryptology – CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 543–571. Springer, 2016. doi: 10.1007/978-3-662-53018-4_20.

[22] Taechan Kim and Jinhyuck Jeong. Extended tower number field sieve with application to finite fields of arbitrary composite extension degree. In *Public-Key Cryptography – PKC 2017, Part I*, volume 10174 of *LNCS*, pages 388–408. Springer, 2017. doi: 10.1007/978-3-662-54365-8_16.

[23] Layr Labs. CurveSmash: Exploratory cryptanalysis challenges for elliptic-curve and pairing groups. https://github.com/Layr-Labs/curvesma.sh, commit 6423c2728dc03d5f2d647c7f22adf1dac0643a66, 2026.

[24] Layr Labs. llm-verifier: Shared execution, review and scoring for exploratory claims. https://github.com/Layr-Labs/llm-verifier, commit 8c1f098875fae0750c29ecfd641f7c9fc53a0f41, 2026.

[25] Hendrik W. Lenstra Jr. Factoring integers with elliptic curves. *Annals of Mathematics*, 126 (3):649–673, 1987.

[26] Harry M. Markowitz. The elimination form of the inverse and its application to linear programming. *Management Science*, 3(3):255–269, 1957.

[27] Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5): 1639–1646, 1993.

[28] Alfred Menezes, Palash Sarkar, and Shashank Singh. Challenges with assessing the impact of NFS advances on the security of pairing-based cryptography. In *Paradigms in Cryptology – Mycrypt 2016*, volume 10311 of *LNCS*, pages 83–108. Springer, 2017. doi: 10.1007/978-3-319-61273-7_5.

[29] Peter L. Montgomery. Modular multiplication without trial division. *Mathematics of Computation*, 44(170):519–521, 1985.

[30] Brian A. Murphy. *Polynomial Selection for the Number Field Sieve Integer Factorisation Algorithm*. PhD thesis, Australian National University, 1999.

[31] Nicholas Nethercote and Julian Seward. Valgrind: A framework for heavyweight dynamic binary instrumentation. In *PLDI 2007*, pages 89–100, 2007.

[32] John M. Pollard. Monte Carlo methods for index computation (mod p). *Mathematics of Computation*, 32(143):918–924, 1978.

[33] Yumi Sakemi, Tetsutaro Kobayashi, Tsunekazu Saito, and Riad S. Wahby. Pairing-friendly curves. Internet-Draft draft-irtf-cfrg-pairing-friendly-curves, IRTF CFRG, 2026. Security-level discussion: GitHub issue cfrg/draft-irtf-cfrg-pairing-friendly-curves#109.

[34] Oliver Schirokauer. Discrete logarithms and local units. *Philosophical Transactions of the Royal Society of London A*, 345:409–423, 1993.

[35] Supranational. blst: Multilingual BLS12-381 signature library. https://github.com/supranational/blst, commit f7a88067ed98385ebc6cf41ccecf21ead706a5b7, 2026.

[36] The CADO-NFS Development Team. CADO-NFS, an implementation of the number field sieve algorithm. https://cado-nfs.gitlabpages.inria.fr/, 2017.

[37] Emmanuel Thomé. Subquadratic computation of vector generating polynomials and improvement of the block Wiedemann algorithm. *Journal of Symbolic Computation*, 33(5): 757–775, 2002.

[38] Paul C. van Oorschot and Michael J. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12(1):1–28, 1999.

[39] Yuqing Zhu and Jincheng Liu. Constructing CM fields for NFS to accelerate DL computation in non-prime finite fields. *IEEE Transactions on Information Theory*, 2023.

A Kernel details

All counts are differential Callgrind or Cachegrind instruction counts [31]: Debian 12 arm64, GCC 12.2 with `-O3` and no vectorisation, or hand-written scalar AArch64 assembly; the count of two repetitions minus the count of one. Every charged table is the monotone envelope of its measured points, rounded up. The assembly sources of the walks are not part of the public packages; their measurement records and listings of the charged loops are.

- *Line walk.* Five nested fixed-level loops with per-pair constants in locals. Every fused operation is written explicitly, so the walk visits the same nodes in the same order as the reference. Each line is laid out as a box of fixed radices, so the inner run is a load, an add and a store of a 16-bit counter with no table access, unrolled eight times. It passes in-order traces on 48 runs and a brute-force comparison. The final designs do not charge it: replay lists cover its range.
- *Record walk.* Hand-allocated AArch64: the upper enumeration levels keep their state in a frame; the two innermost levels, the leaf and record emission keep theirs in registers. Exact clipping of each level to the ellipsoid. A record is 8 bytes (16 bits of $\log_2 q$, 32 bits of cell address within a page). In-order traces are identical to the reference on tens of thousands of cosets, including high-density ones ($5 \cdot 10^{10}$ hits), and one-instruction mutants fail. The second-generation walk is measured per density stratum on 512 walks per stratum at the final region; its price is the stratum mean plus two standard errors.
- *Page apply.* Records are written into 1 MiB pages, one open page per (line, chunk). A full page is handed to the owner of its cells by pointer and applied in place: 2.63 instructions per record. Sums are checked against a plain C sum on $1.7 \cdot 10^8$ production records.
- *Replay.* Copy and apply loops process 16 entries per iteration: 18 and 26 instructions. Lookup of a bucket: 24 instructions plus a branchless search for the radius cut. Exactness: on the real forms and ideals at four sizes up to $2^{24.9}$ cells, the cells selected by residue, class and radius equal the cells hit by direct evaluation on $1.5 \cdot 10^6$ (ideal, slice) pairs and $3.3 \cdot 10^9$ hits, with failing negative controls.
- *Remainder trees.* GMP 6.2.1, statically linked, every leaf checked against a direct remainder. The scaled tree keeps $L + 2$ guard words at depth L; the error per level is below 2^{-64G}, so no leaf wraps. Products above 2^{26} limbs are charged by the bound $F(N) = 8.5N \log_2(N)^2 + 1200$ instructions (H-FFT); at the largest sizes this is 2.8 times the cost extrapolated from measurements up to 2^{20} limbs.
- *Matrix kernels.* Straight-line Montgomery and truncated-quotient multiplications by χ, and the exact row reduction, checked on $5 \cdot 10^4$ random and extreme inputs each. The multi-word addition of unit entries uses a carry chain merged across owners (due to yukonautominer).

B Artifact and relation to the public packages

The two ledgers of Table 1 are public packages of the CurveSmash challenge [23] (BLS12-381: package c36 of 17 November 2026; BN254: package bn15r of the same day). A package contains its model, its ledger evaluator, its evidence scripts and the stored pool statistics, and recomputes its ledger on a pinned Python image with one CPU and no network. The challenge ranks submissions by expected work at success probability 0.39, which a randomised early exit lowers by $\log_2(0.998/0.39) = 1.36$ bits without changing the attack; the packages therefore declare $2^{111.87}$ and $2^{87.59}$ units, and state the cost of a complete run, used throughout this paper, as $2^{113.23}$ and $2^{88.95}$.

The paper repository adds the duplicate census and its separately written verification (Table 6); the matched-ablation study; the filtering campaigns; the Krylov diagnostics of Section 7.4; and the scripts that generate the figures. All computations ran in containers without network access (SageMath 10.4, PARI/GP 2.15, Python 3.12). Several components of the packages are due to other participants of the challenge and are credited in the acknowledgements and in the packages themselves.