



LARGE SMOOTH TWINS FROM SHORT LATTICE VECTORS

ERIK MULDER, BRUNO STERNER, AND WESSEL VAN WOERDEN

ABSTRACT. Finding the largest pair of consecutive B -smooth integers for a fixed B , also called a B -smooth twin, is computationally challenging. It has only been provably done for $B \leq 100$ and heuristically for $100 < B \leq 113$. We improve this by detailing a new algorithm to find such smooth twins based on solving the shortest vector problem (SVP) in a well-constructed lattice. Using an estimate of the largest B -smooth twin that we develop in this work, we are able to significantly increase B and notably report the heuristically largest twin with $B = 751$, which has 196 bits. By slightly modifying the lattice, we are able to find even larger twins, but the resulting smoothness bound will not always be optimal. This notably includes a 213-bit twin with $B = 997$, which is the largest twin found in this work.

“Such pairs lead to sets of “nearly” dependent logarithms of primes. For instance the number $K = \log(63927525376) - \log(63927525375) = 13 \log(2) - 3 \log(3) - 3 \log(5) - 7 \log(7) + 4 \log(11) + \log(13) - \log(23) + \log(41)$, which cannot be zero because of the unique factorisation theorem, is, however, less than $1.56427 \cdot 10^{-11}$.”

- D.H. Lehmer [28]

1. INTRODUCTION

In 1991, during what can be considered the golden age of integer factorisation, Schnorr [36] proposed a factoring method¹ based on solving the closest vector problem in the so-called *prime number lattice*. While his strategy looked promising, the growing and necessary lattice dimension when you increase the input integer provides an obstruction to the feasibility of the algorithm. As a result, it did not compete with other factoring algorithms such as the number field sieve [29].

Many modern integer factoring techniques use smooth integers which are integers with only small prime factors. We quantify this properly to say that an integer n is B -smooth if $p \mid n \Rightarrow p \leq B$ for all primes p , where we refer to B as the *smoothness bound*. We refer to [26] for the factoring relevance of these integers and beyond.

Recently the problem of finding pairs of consecutive smooth integers, which we call smooth twins, has attracted attention. The primary relevance of these smooth twins comes from isogeny-based cryptography. Certain cryptosystems, including the original versions of SQIsign [24, 25] and a variant of POKÉ [8, 15], either work with large smooth twins whose sum is prime or a small variant of this. To formalise this, we call a pair of consecutive integers $(r, r+1)$ a B -smooth twin if their product $r \cdot (r+1)$ is B -smooth and drop B from this notation when it is implicit from the context. We call a B -smooth twin *strictly* B -smooth if it is not $(B-1)$ -smooth.

It is known that the set of B -smooth twins is finite [39] and one can find all such twins by solving exponentially many Pell equations in B . This has been done for $B = 100$ [31] and heuristically for $B = 113$ [16] – the largest of these twins have

2020 *Mathematics Subject Classification*. Primary .

¹Which was independently analysed by Adleman [2]

64 and 75 bits (respectively). We call the largest B -smooth twin an *optimal twin*. In the other direction, for an input bit size b , the *optimal smoothness bound* is the smallest smoothness bound B such that there is a b -bit B -smooth twin.

For cryptographic relevance one requires 256-bit smooth twins. We call these *cryptographic smooth twins*. The optimal smoothness bound for such a twin is $B \approx 1250$. So finding them from solving (at least a large proportion of) Pell equations is far beyond computational reach. Other methods [16, 17, 38] exist that find such cryptographic smooth twins but one obtains larger smoothness bounds.

Contributions. In this work we revisit Schnorr’s factoring idea and repurpose it to find large smooth twins. We observe that the shortest vectors in the prime number lattice often correspond to smooth twins. So using state-of-the-art SVP algorithms [5, 22, 41] we find short vectors in the lattice and see whether they correspond to smooth twins. By parametrising the lattice correctly, one can search for smooth twins of a particular size including the optimal twins. With current SVP algorithms, we can find heuristically optimal twins with much larger smoothness bounds. For instance, we found the following 196-bit 751-smooth twin

$$\begin{aligned}
 r &= 7^7 \cdot 11 \cdot 17 \cdot 29 \cdot 47 \cdot 59 \cdot 67 \cdot 83^2 \cdot 89 \cdot 151^3 \cdot 163 \cdot 173 \cdot 271 \cdot 347 \cdot 461 \\
 &\quad \cdot 491 \cdot 547 \cdot 587 \cdot 619 \cdot 661 \cdot 683 \cdot 701, \text{ and} \\
 (1) \quad r+1 &= 2 \cdot 3^9 \cdot 13^2 \cdot 19 \cdot 31 \cdot 41 \cdot 71 \cdot 73 \cdot 97 \cdot 157^2 \cdot 181^3 \cdot 191 \cdot 227 \cdot 241 \cdot 293 \\
 &\quad \cdot 307^3 \cdot 337 \cdot 557 \cdot 617 \cdot 727 \cdot 751
 \end{aligned}$$

which we believe is optimal. As mentioned earlier, the only known provable optimal twins have $B \leq 100$. For $100 < B \leq 113$ optimal twins are only known heuristically.

In order to convince the reader that this 751-smooth twin is likely the optimal twin, we develop theoretical bounds and estimates for optimal twins based on some heuristics. In particular, the standout result is the following.

Theorem 1.1. *Assuming Heuristic 3.1, as $B \rightarrow \infty$ we have the following asymptotic for optimal B -smooth twins $(r, r+1)$:*

$$r \sim e^{2e\sqrt{B}}.$$

This result explains the asymptotic behaviour of optimal twins, but for concrete and small B this estimate deviates from the precise size. For this regime one can use numerical techniques to get a better approximation for optimal twins.

Already in 1975, Erdős essentially showed, under analogous heuristic assumptions, that the largest B -smooth twin grows as $r \sim e^{O(\sqrt{B})}$ [23]. We improve on this by making the constant in the exponent explicit.

By introducing some trade-offs in the lattice, one can find even larger twins. One cannot expect the resulting twins to be optimal using these trade-offs. For instance we found the following 213-bit 997-smooth twin

$$\begin{aligned}
 r &= 19^2 \cdot 41 \cdot 43^2 \cdot 53 \cdot 59^2 \cdot 73^2 \cdot 83 \cdot 173 \cdot 227 \cdot 241 \cdot 281 \cdot 337 \cdot 397^2 \cdot 433 \\
 &\quad \cdot 541 \cdot 577 \cdot 593 \cdot 787 \cdot 821 \cdot 839 \cdot 857^2 \cdot 967, \text{ and} \\
 (2) \quad r+1 &= 2^2 \cdot 3 \cdot 5^2 \cdot 13^3 \cdot 23 \cdot 37 \cdot 47 \cdot 79 \cdot 107 \cdot 127 \cdot 131 \cdot 151 \cdot 157 \cdot 167 \cdot 179 \\
 &\quad \cdot 181^2 \cdot 193 \cdot 223 \cdot 283 \cdot 317 \cdot 367 \cdot 379 \cdot 601 \cdot 709^2 \cdot 743 \cdot 941 \cdot 997
 \end{aligned}$$

and, with the estimates that will be developed in Section 3, we suspect that the optimal 997-smooth twin should have approximately 227 bits. We discuss the challenges to find larger smooth twins relevant for cryptography.

For some fixed values of B , we attempt to find all B -smooth twins from our approach by expanding a known set of smooth twins [10]. In particular we conjecture to have the complete set of 200-smooth twins. The only known way to verify this is to comprehensively solve 2^{46} Pell equations which is computationally challenging.

In addition, we put our algorithm in a cryptographic context and get new parameters for the original version of SQIsign [24] with smaller smoothness bounds. With a theoretical decrease of 19.85% on the signing time, these new parameters are a competitive alternative that could be considered for future use.

A Python3 implementation of our algorithm is publicly available at

<http://github.com/WvanWoerden/SmoothTwins>,

and a dataset containing the conjectured complete set of 200-smooth twins, and some of the largest smooth twins for $B < 1100$, is publicly available at [34]

<https://zenodo.org/records/18234089>.

Notation. For a smoothness bound B , $P_B := \{2, 3, \dots, q\} = \{p \leq B\}$ will be the set of primes up to B with cardinality $\pi(B) = \#P_B$. We will denote $\log(\cdot)$ and $\log_2(\cdot)$ for the natural and base-2 logarithm respectively. We denote $\text{val}_p(y)$ for the valuation of $y \in \mathbb{Q}$ at p . For two functions $f, g : \mathbb{N} \rightarrow \mathbb{R}$ we write $f(n) = O(g(n))$ if there is some $C > 0$ and $N \in \mathbb{N}$ such that $|f(n)| \leq C|g(n)|$ for all $n \geq N$. We write $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$; $f(n) = o(g(n))$ if $f(n)/g(n) \rightarrow 0$ as $n \rightarrow \infty$; and also $f(n) \sim g(n)$ if $f(n)/g(n) \rightarrow 1$ as $n \rightarrow \infty$.

Acknowledgements. The second author was supported by two projects: first the HYPERFORM² consortium funded through Bpifrance and l’Agence nationale de la recherche through a Plan France 2030 grant (ANR-22-PETQ-0008 PQ-TLS); and second the NSERC Alliance Consortia Quantum Grant “Accelerating the transition to quantum-resistant cryptography” (ALLRP 578463–2022), and the NSERC Discovery Grant program³. Part of this work was executed while the third author was employed by IMB at Université de Bordeaux under the CHARM ANR-NSF grant (ANR-21-CE94-0003). Experiments presented in this paper were carried out using the PlaFRIM experimental testbed, supported by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d’Aquitaine⁴.

2. EXISTING METHODS FOR FINDING SMOOTH TWINS

We give a short summary of two methods that find all or almost all B -smooth twins for a fixed and small smoothness bound B .

Solving Pell equations. Let $x = 2r + 1$ so that $x^2 - 1 = 4r(r + 1)$ is B -smooth. Decompose this into its squarefree part, D , and its square part, y^2 . Then the pair (x, y) is a solution to the Pell equation $X^2 - DY^2 = 1$. One reverses this to solve all $2^{\pi(B)}$ Pell equations, one for each $D = \prod_{\ell \in P_B} \ell^{e_\ell}$ with $e_\ell \in \{0, 1\}$. Lehmer [28] gives provable conditions on these equations to ensure all B -smooth twins are found.

²See <https://project.inria.fr/hyperform>.

³See <https://nserc-crsng.canada.ca/en/funding-opportunity/archived-alliance-consortia-quantum-grants-call-proposals-supporting>

⁴See <https://www.plafrim.fr>.

This is computationally infeasible for large B – the complexity to solve these equations is at least $2^{\pi(B)+o(\pi(B))}$. This ignores the fact that the fundamental solution is (in the worst case) doubly exponential in B . Finding such solutions would add another exponential to this complexity⁵. In practice one can adopt an early abort strategy that only considers solutions up to some reasonable bound. If this bound is exponential in B , then processing each discriminant becomes polynomial in B and the total complexity becomes $2^{\pi(B)+o(\pi(B))}$. With this, one could heuristically get all B -smooth twins (as done for $B = 113$ [16]).

Conrey-Holmstrom-McLaughlin (CHM). This algorithm [12] recursively builds an increasing chain of B -smooth twins $S^{(0)} = \{1, \dots, B-1\} \subseteq S^{(1)} \subseteq \dots \subseteq S^{(i)} \subseteq S^{(i+1)} \subseteq \dots \subseteq S^{(n-1)} = S^{(n)}$ until no more twins are found (which is ensured by Størmer’s [39] result). For each $r, s \in S^{(i)}$ with $r < s$ compute

$$\frac{t}{t'} = \frac{r}{r+1} \cdot \frac{s+1}{s},$$

where $\gcd(t, t') = 1$, and include t in the next set $S^{(i+1)}$ when $t' = t+1$. This produces a B -smooth twin $(t, t+1)$ and shows the correctness of the recursion.

This procedure produces a large but incomplete set of B -smooth twins. It is conjectured that the proportion of B -smooth twins not found with CHM is $o(1)$ as $B \rightarrow \infty$. The experiments in [10] report the results with $B = 547$. The largest twin found in that run has 122 bits which we will see later is far from optimal.

3. SMOOTHNESS HEURISTICS

The Dickman-De Bruijn [19, 18] function is the continuously differentiable function $\rho : [0, \infty) \rightarrow [0, 1]$ that satisfies the following difference differential equation

$$(3) \quad \begin{aligned} \rho(u) &= 1, & (0 \leq u \leq 1); \\ u\rho'(u) &= -\rho(u-1), & (u > 1). \end{aligned}$$

Other than $\rho(u) = 1 - \log(u)$ for $1 < u \leq 2$, there is no closed form for $\rho(u)$ in terms of elementary functions for arbitrary $u > 1$. A simple approximation of this function is $\rho(u) \approx u^{-u}$ and a better one is given by $\rho(u) \approx e^u (u \log(u))^{-u}$. These approximations are derived from the asymptotic expression [26, §3.9]

$$(4) \quad \rho(u) = \left(\frac{e + o(1)}{u \log(u)} \right)^u,$$

as well as a more precise statement due to De Bruijn [18, Eqn (1.8)]. From a computational perspective, numerical methods [40] exist for evaluating this function which are built into many popular computer algebra packages.

This function is related to the distribution of smooth integers at an asymptotic level. More precisely, if $\Psi(N, B)$ denotes the number of B -smooth integers $m \leq N$, then for a fixed $u > 0$ and as $N \rightarrow \infty$ we have [19, 18]

$$\Psi(N, N^{1/u}) \sim \rho(u)N.$$

While this expression is asymptotic, experimentally $\rho(u)$ is a good estimate for the probability that an integer of size N is $N^{1/u}$ -smooth.

⁵We refer to [28, §6] and [31] for some discussion on getting around this.

3.1. Heuristics on the optimal twin. It is believed that the smoothness property in very short intervals is mutually independent. A more formal and general statement is conjectured which has only been proven in a small range [32]. This fact has been utilised in practice [17, 38] owing to its accuracy for concrete instances. This conjecture implies that the probability that two close integers of size N are $N^{1/u}$ -smooth is $\rho(u)^2$. Therefore, for large enough N , the number of $N^{1/u}$ -smooth twins in the interval $[\frac{N}{2}, \frac{3N}{2}]$ is heuristically roughly $\rho(u)^2 N$. This means that if $\rho(u)^2 N = 1$, then we expect roughly one $N^{1/u}$ -smooth twin of size N , and not many greater than N . This gives us the following heuristic on the optimal B -smooth twin. We will use this heuristic in combination with (4) to prove Theorem 1.1.

Heuristic 3.1. *As $B \rightarrow \infty$, the largest B -smooth twin $(r, r+1)$ satisfies*

$$\rho(u) \sim \frac{1}{\sqrt{r}}$$

where $u = \log(r)/\log(B)$.

As a precursor, we need the principal branch of the Lambert W function – the function $W_0 : \mathbb{R}_{>0} \rightarrow \mathbb{R}_{>0}$ such that for any $x > 0$ and $y = W_0(x) > 0$ we have $ye^y = x$. It is known that $W_0(x) = \log(x) - \log(\log(x)) + o(1)$ as $x \rightarrow \infty$ [13]. We need a slight modification of this which is summarised in this short lemma.

Lemma 3.2. *Consider the function $x \mapsto x \log(x)$ for $x > 1$ (so that $x \log(x) > 0$). Then the principal branch of its inverse is the function $x \mapsto x/W_0(x)$ for $x > 0$.*

Proof. Starting from $y \log(y) = x$, with $x > 0$ and $y > 1$, we express y solely in terms of x by first rewriting this as $(x/y)e^{x/y} = x$ and then obtain $x/y = W_0(x)$. $\square$

Proof of Theorem 1.1. We start with $\rho(u) = 1/\sqrt{r}$ from Heuristic 3.1. Using the asymptotic expression from (4) and $u = \log(r)/\log(B)$, this can be rearranged to

$$(5) \quad u \log(u) = (e + o(1))\sqrt{B}.$$

We need to address the $o(1)$ term here since this is a priori with respect to u and not B . Write $g(u)$ for this $o(1)$ term. We will show that $\lim_{B \rightarrow \infty} u = \infty$, which implies that $\lim_{B \rightarrow \infty} g(u) = 0$ as well. Let $T := \liminf_{B \rightarrow \infty} |e + g(u)|$.

Suppose that $T = 0$. Note that the optimal B -smooth twin satisfies $r \geq B^2 - 1$ and hence $u = \log(r)/\log(B) \geq 1.5$ for large enough B . So the denominator and exponent of (4) are positive for large enough B and we get $\liminf_{B \rightarrow \infty} \rho(u) = 0$. Since $\rho(u)$ is a strictly descending function for $u > 1$ with limit 0 we must have $\lim_{B \rightarrow \infty} u = \infty$. This contradicts the assumption that $T = 0$.

Hence we know that $T > 0$. By (5) we have $\liminf_{B \rightarrow \infty} |u \log(u)| = \infty$ and, as the map $x \mapsto |x \log(x)|$ has no poles other than ∞ , we must have $\lim_{B \rightarrow \infty} u = \infty$. This implies that the $o(1)$ in (5) is also with respect to B . Using Lemma 3.2 we can express u as

$$(6) \quad u = \frac{(e + o(1))\sqrt{B}}{W_0((e + o(1))\sqrt{B})} = \frac{(e + o(1))\sqrt{B}}{1 + \frac{1}{2} \log(B) - \log(1 + \frac{1}{2} \log(B)) + o(1)}.$$

We can now conclude that

$$\lim_{B \rightarrow \infty} \log(r)/\sqrt{B} = \lim_{B \rightarrow \infty} u \log(B)/\sqrt{B} = 2e. \quad \square$$

Smoothness bound B	Approximate $\log_2(r)$	Smoothness bound B	Approximate $\log_2(r)$
200	91.138890	1500	282.426575
500	154.417132	2000	329.581941
750	193.534196	2500	371.202628
1000	226.657489	3000	408.877910
1100	238.750073	4500	506.210979
1250	255.920233	10000	767.101783

Table 1. Heuristic size of optimal twins for various smoothness bounds B , i.e. approximate solutions to $\rho(u) = 1/\sqrt{r}$.

Numerical computations for small B . Theorem 1.1 gives a strong description of the asymptotic behaviour of optimal B -smooth twins. However for concrete and small B this is not effective and one needs a more explicit solution to the equation $r\rho(u)^2 = 1$ underlying Heuristic 3.1. One cannot solve this equation exactly and so we resort to numerical techniques to approximate the solution.

One starts with an initial guess for r , such as the one given in Theorem 1.1, and uses a root-finding algorithm, such as the *bisection* or *Newton's method*, to iteratively get a better approximation for r . One can exploit the differentiable condition (see (3)) for Newton's method and get a faster convergence rate. But we found that the bisection method was sufficiently fast for the values of B we were interested in.

For small $B \leq 113$ these numerical estimates for optimal twins correlate well to actual optimal twins found by solving Pell equations, see Figure 2. As a result we estimate the size of optimal twins for larger B , see Table 1.

4. LATTICE THEORY

A lattice $\mathcal{L}$ is a discrete subgroup of the Euclidean space $\mathbb{R}^m$. Equivalently, any lattice $\mathcal{L}$ can be described by a basis $\mathbf{B} \in \mathbb{R}^{m \times n}$ such that

$$\mathcal{L} = \mathcal{L}(\mathbf{B}) := \left\{ \sum_{i=1}^n x_i \mathbf{b}_i : x_i \in \mathbb{Z} \right\},$$

with $\mathbb{R}$ -linearly independent columns $\mathbf{b}_1, \dots, \mathbf{b}_n$. We denote $\text{rk}(\mathcal{L}) = n$ for the *rank* of the lattice, which coincides with the dimension of its $\mathbb{R}$ -linear span.

The discrete nature of a lattice allows us to define some geometric properties. The minimal distance between any two distinct lattice points is called its *first minimum*, denoted by $\lambda_1(\mathcal{L})$. Equivalently, it equals the length of the shortest non-zero vector:

$$\lambda_1(\mathcal{L}) := \min_{\mathbf{v} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{v}\|.$$

Next, we have the (co)volume $\text{vol}(\mathcal{L})$ of a lattice which equals the volume of the quotient space $\text{span}_{\mathbb{R}}(\mathcal{L})/\mathcal{L}$. For any basis $\mathbf{B}$ of $\mathcal{L}$ we have:

$$\text{vol}(\mathcal{L}) := \text{vol}(\text{span}_{\mathbb{R}}(\mathcal{L})/\mathcal{L}) = \det(\mathbf{B}^\top \mathbf{B})^{1/2}.$$

The volume $\text{vol}(\mathcal{L})$ of a lattice can be seen as the inverse density of the lattice, i.e., $\text{vol}(\mathcal{L})^{-1}$ is roughly the number of lattice vectors per unit volume. In a dense enough lattice of rank n there must always be some short non-zero lattice vector,

which is quantified by Minkowski's theorem saying that

$$\lambda_1(\mathcal{L}) \leq \frac{2 \operatorname{vol}(\mathcal{L})^{1/n}}{\operatorname{vol}(\mathcal{B}^n)^{1/n}} \approx \sqrt{2n/\pi e} \cdot \operatorname{vol}(\mathcal{L})^{1/n},$$

where $\operatorname{vol}(\mathcal{B}^n)$ is the volume of a unit ball of dimension n . In a typical *random lattice* one can give a more precise estimate of the first minimum. In practice, we usually never have precisely such a random lattice, however, unless they have some special structure, most lattices still satisfy the following simplified heuristic.

Heuristic 4.1 (Gaussian Heuristic). *Let $\mathcal{L}$ be a lattice and let $S \subset \operatorname{span}_{\mathbb{R}}(\mathcal{L})$ be a sufficiently nice body with positive Euclidean volume. Then the expected number of non-zero lattice points in S roughly equals $|\mathcal{L} \cap S \setminus \{0\}| \approx \operatorname{vol}_{\operatorname{span}(\mathcal{L})}(S) / \operatorname{vol}(\mathcal{L})$. In particular, for a lattice of rank n this implies that*

$$\lambda_1(\mathcal{L}) \approx \operatorname{gh}(\mathcal{L}) := \frac{\operatorname{vol}(\mathcal{L})^{1/n}}{\operatorname{vol}(\mathcal{B}^n)^{1/n}} \approx \sqrt{n/2\pi e} \cdot \operatorname{vol}(\mathcal{L})^{1/n}.$$

4.1. The shortest vector problem (SVP). One of the main computational lattice problems is that of finding short non-zero lattice vectors. Finding a lattice vector $\mathbf{v} \in \mathcal{L}$ of minimal length $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$ is known as the shortest vector problem (SVP). While the best provable algorithm for SVP runs in time and space $2^{n+o(n)}$ in the rank n of the lattice [3, 4], heuristic versions can be much better. The fastest heuristic algorithm for SVP runs in time $2^{0.292n+o(n)}$ and space $2^{0.2075n+o(n)}$ [9]. We commonly refer to these algorithms as *lattice sieving* algorithms. In practice, these algorithms can solve SVP up to dimensions 150 in reasonable time [5], with record computations stretching up to dimension 200 [22, 41].

The above complexities are for finding an (almost) shortest vector in a typical lattice for which $\lambda_1(\mathcal{L}) \approx \operatorname{gh}(\mathcal{L})$. These seem to be the hardest cases to find a shortest vector both in theory as in practice. The lattice we will consider has some special structure and therefore does not always fall into this regime. For example, when the short vector $\mathbf{v} \in \mathcal{L}$ we aim to find is shorter than what the Gaussian Heuristic predicts. Alternatively, even if the lattice satisfies $\lambda_1(\mathcal{L}) \approx \operatorname{gh}(\mathcal{L})$, the short vector $\mathbf{v} \in \mathcal{L}$ we aim to find does not always have to be the shortest vector.

The case $\|\mathbf{v}\| < \operatorname{gh}(\mathcal{L})$. We call a shortest vector $\mathbf{v} \in \mathcal{L}$ *unusually short* if $\|\mathbf{v}\| = \lambda_1(\mathcal{L}) < \operatorname{gh}(\mathcal{L})$. Finding such an unusually short vector is typically easier by running a lattice reduction algorithm such as BKZ [37] which makes polynomially many calls to an exact SVP oracle in dimension $\beta \leq n$. The dimension β depends on n and the ratio $\operatorname{gh}(\mathcal{L})/\lambda_1(\mathcal{L})$ which is called the *gap*. The larger the gap, the lower β will be relative to n in order to successfully recover the unusually short vector.

The case $\|\mathbf{v}\| \geq \operatorname{gh}(\mathcal{L})$. We consider the case where one wants to find a sufficiently short vector $\mathbf{v} \in \mathcal{L}$ for which $\operatorname{gh}(\mathcal{L}) \leq \|\mathbf{v}\| \leq C \cdot \operatorname{gh}(\mathcal{L})$ for some $C \geq 1$. Finding any such short vector is typically easier than finding a shortest vector, again by the use of a lattice reduction algorithm. In fact, in a typical lattice there are according to the Gaussian Heuristic roughly C^n of such vectors. However, in our case we will be interested in finding a specific (or a few specific) vectors among those C^n vectors. Finding a needle in such a haystack will quickly become infeasible, and thus we are typically constrained to quite low values of C . The current best heuristic algorithms for SVP naturally compute almost all vectors up to length $\sqrt{4/3} \cdot \operatorname{gh}(\mathcal{L})$ and thus in practice we typically constrain ourselves to $C \leq \sqrt{4/3}$ in large dimensions.

Dimensions for free. One can consider the projected lattice $\pi(\mathcal{L})$ of rank $n-k$ where π projects away from k (primitive) lattice vectors. For a shortest vector $\mathbf{v} \in \mathcal{L}$ (with $\|\mathbf{v}\| \approx \text{gh}(\mathcal{L})$) its projection $\pi(\mathbf{v})$ might have $\|\pi(\mathbf{v})\| \leq \sqrt{4/3} \cdot \text{gh}(\pi(\mathcal{L}))$ and thus $\pi(\mathbf{v})$ can be found by running the SVP algorithm on $\pi(\mathcal{L})$. Then $\mathbf{v}$ can be recovered by an appropriate lifting. One thus decreases the rank of the SVP by k dimensions, which for typical lattices can heuristically be picked as $k = \theta(n/\log(n))$ [20].

5. GENERALISED PRIME NUMBER LATTICE

Here we reintroduce and generalise the prime number lattice that will be the main subject throughout the rest of this work. We remark that other names have been used for the lattice including the “Schnorr-Adleman lattice”.

Let $\alpha, \alpha_i \in \mathbb{R}$, for $i \in \{1, \dots, n\}$, and $P = \{p_i\}_{i=1}^n$ be a finite subset of distinct primes of size n . We call the α_i ’s *weights* and call P a *factor base*. We say that a positive integer is *P-smooth* if all its prime divisors are in P . Note that for $P = P_B = \{p \leq B\}$ this coincides with being *B-smooth*.

Definition 5.1. We define the *prime number lattice* adjoined to α, α_i and a factor base P , denoted $\mathcal{L}_{\alpha, \alpha_i, P}$, to be the lattice with the following basis matrix:

$$\mathbf{B}_{\alpha, \alpha_i, P} := \begin{pmatrix} \log(p_1)\alpha & \log(p_2)\alpha & \cdots & \log(p_n)\alpha \\ \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_n \end{pmatrix}.$$

When $P = P_B$ we call this lattice the *full prime number lattice* (or simply the *full lattice*) with respect to the smoothness bound B .

Besides the relevance of this lattice to integer factoring which was alluded to in the introduction, the lattice also plays a role in lattice sphere packings, where it was used to prove the NP-hardness of SVP under randomised reductions [33]. Historically one restricts the weights α_i to be either $\log(p_i)$ or $\sqrt{\log(p_i)}$. We write α and α_i abstractly for the time being and make concrete choices later on.

A vector $\mathbf{B}_{\alpha, \alpha_i, P} \mathbf{x}$ in this lattice where $\mathbf{x} \in \mathbb{Z}^n$ can be associated to a quotient of coprime P -smooth integers a, b which we call a *P-smooth rational* (or just a *smooth rational*). The concrete smooth rational is apparent in the first entry of the lattice vector: $\alpha \sum_{i=1}^n \log(p_i) x_i = \alpha \log(a/b)$. Moreover, the association goes the other way since $x_i = \text{val}_{p_i}(a/b)$, so we get the following correspondence:

Lemma 5.2. *For each α and α_i , there is a one-to-one correspondence⁶ between P -smooth rationals and lattice vectors in $\mathcal{L}_{\alpha, \alpha_i, P}$ given by:*

$$\frac{a}{b} = \prod_{i=1}^n p_i^{x_i} \in \mathbb{Q} \longleftrightarrow \mathbf{B}_{\alpha, \alpha_i, P} \mathbf{x} \in \mathcal{L}_{\alpha, \alpha_i, P}.$$

The other entries in the lattice vector are $\alpha_i x_i$ for each i . These help to balance out the contribution of the first entry $\alpha \log(a/b)$ with the size of the exponents $|x_i|$.

⁶A more appropriate name for this lattice might be the “smooth rational lattice” given this correspondence. But we chose to call it the “prime number lattice” for historical context.

5.1. Properties of the prime number lattice. Due to Lemma 5.2 the length of each vector can directly be computed from the smooth rational a/b and its rational prime factorisation. Such lengths should be considered relative to the Gaussian Heuristic and thus the volume of the lattice. This gives the following lemmas.

Lemma 5.3. *Let $\mathbf{v} = \mathbf{B}_{\alpha, \alpha_i, P} \mathbf{x} \in \mathcal{L}_{\alpha, \alpha_i, P}$ be a lattice vector corresponding to a P -smooth rational $a/b = \prod_{i=1}^n p_i^{x_i}$ with $\gcd(a, b) = 1$. Then we have*

$$\|\mathbf{v}\|^2 = \alpha^2 \cdot \log\left(\frac{a}{b}\right)^2 + \sum_{i=1}^n (x_i \alpha_i)^2.$$

Moreover we have the following bounds

$$\frac{(\sum_{i=1}^n |x_i \alpha_i|)^2}{n} \leq \|\mathbf{v}\|^2 - \alpha^2 \cdot \log\left(\frac{a}{b}\right)^2 \leq \left(\sum_{i=1}^n |x_i \alpha_i|\right)^2.$$

Proof. The first statement follows from Lemma 5.2 and the second follows from the general norm-inequality $\frac{1}{n} \|\mathbf{y}\|_1^2 \leq \|\mathbf{y}\|_2^2 \leq \|\mathbf{y}\|_1^2$ where $\mathbf{y} := (x_i \alpha_i)_{i=1}^n$. $\square$

Lemma 5.4 (Volume). *We have*

$$\text{vol}(\mathcal{L}_{\alpha, \alpha_i, P}) = \sqrt{\alpha^2 \cdot \sum_{i=1}^n (\log(p_i)/\alpha_i)^2 + 1} \cdot \prod_{i=1}^n \alpha_i.$$

If $\alpha_i := \log(p_i)$, the above simplifies to $\text{vol}(\mathcal{L}_{\alpha, \log(p_i), P}) = \sqrt{\alpha^2 n + 1} \cdot \prod_{i=1}^n \log(p_i)$.

Proof. Let $\mathbf{C}$ be the basis where each i -th column of $\mathbf{B}_{\alpha, \alpha_i, P}$ is divided by α_i . Then $\text{vol}(\mathcal{L}_{\alpha, \alpha_i, P}) = \text{vol}(\mathcal{L}(\mathbf{C})) \cdot \prod_{i=1}^n \alpha_i$. The first row of $\mathbf{C}$ is a vector $\mathbf{c}^\top = (\log(p_i) \cdot \alpha / \alpha_i)_{i=1}^n$, and the remaining n rows form an identity matrix I_n . So we have $\mathbf{C}^\top \mathbf{C} = I_n^\top I_n + \mathbf{c} \mathbf{c}^\top$ which has eigenvalues $1 + \|\mathbf{c}\|^2, 1, \dots, 1$ and thus determinant $1 + \|\mathbf{c}\|^2$. So $\text{vol}(\mathcal{L}_{\alpha, \alpha_i, P}) = \sqrt{1 + \|\mathbf{c}\|^2} \cdot \prod_{i=1}^n \alpha_i$ from which we obtain the result. $\square$

5.2. Finding smooth twins from the prime number lattice. We give some intuition for finding smooth twins from this lattice. Recall from Section 4 that one can find a short vector $\mathbf{v}$ in a lattice $\mathcal{L}$ if the ratio $\|\mathbf{v}\| / \text{gh}(\mathcal{L})$ is not too large.

First we consider the norm of the vector $\mathbf{v} = \mathbf{B} \mathbf{x} \in \mathcal{L}_{\alpha, \alpha_i, P}$ corresponding to a smooth rational a/b . Lemma 5.3 says that

$$\|\mathbf{v}\|^2 = \alpha^2 \cdot \log\left(\frac{a}{b}\right)^2 + \sum_{i=1}^n (x_i \alpha_i)^2.$$

Observe that for a smooth twin $(a, b) = (r, r + 1)$ we have that a/b is very close to 1 and thus $\log(a/b) \approx 0$, while for other smooth rationals the contribution of $|\log(a/b)|$ can be much larger. More precisely, for large enough r we have the approximation $\alpha^2 \log(a/b)^2 \approx (\alpha/(r+1))^2$. From this we see that we can actually pick α up to size $O(r)$ to get $\alpha^2 \cdot \log(a/b)^2 = O(1)$. In other words, we can increase α significantly without increasing the length of $\mathbf{v}$ much.

For the ratio of the vector $\mathbf{v}$ relative to the Gaussian Heuristic, we also have to consider the sparsity or equivalently the volume of the lattice. From Lemma 5.4 we see that this is mainly determined by α , in particular increasing α makes the lattice sparser. As we can freely increase α up to $O(r)$ essentially without increasing the norm of $\mathbf{v}$ one could hope that the lattice becomes sparse enough for $\mathbf{v}$ to be a short

vector in the lattice. To make this reasoning more precise we first give a concrete and *optimal* choice of α in the following proposition.

Proposition 5.5. *Let a/b be a B -smooth rational and $P_{a,b} := \{p \in P_B : p \mid ab\}$. Choose a factor base $P = \{p_i\} \supseteq P_{a,b}$ with $n = \#P$; and weights α_i for $i = 1, \dots, n$. Set $\beta_1 := \log^2(a/b)$ and $\beta_2 := \sum_{i=1}^n (x_i \alpha_i)^2$ where $x_i = \text{val}_{p_i}(a/b)$. Then the choice*

$$\alpha = \alpha_{\text{opt}} \approx \sqrt{\frac{\beta_2}{(n-1)\beta_1}}$$

minimises the GH ratio $\|\mathbf{v}\| / \text{gh}(\mathcal{L}_{\alpha, \alpha_i, P})$, among all $\alpha \in \mathbb{R}$, where $\mathbf{v} \in \mathcal{L}_{\alpha, \alpha_i, P}$ is the corresponding vector to the smooth rational a/b .

Proof. Consider the following function $f(\alpha) = \|\mathbf{v}\|^2 / \text{gh}(\mathcal{L}_{\alpha, \alpha_i, P_B})^2$ as a one variable function in terms of α . With Lemma 5.3 and Lemma 5.4, we approximate⁷ $f(\alpha)$ as

$$f(\alpha) \approx \frac{\alpha^{2-2/n}\beta_1}{\gamma^2} + \frac{\beta_2}{\gamma^2\alpha^{2/n}}, \quad \text{where } \gamma = \sqrt{\frac{n}{2\pi e}} \left[\sqrt{\sum_{i=1}^n \left(\frac{\log(p_i)}{\alpha_i} \right)^2} \prod_{i=1}^n \alpha_i \right]^{\frac{1}{n}}.$$

Minimising this expression in α is a straightforward calculus exercise. The result of this exercise gives the desired expression for α_{opt} . $\square$

Corollary 5.6. *With P, n, β_1, β_2 and γ defined in the Proposition 5.5 and the optimal choice $\alpha = \alpha_{\text{opt}}$, we have*

$$\frac{\|\mathbf{v}\|}{\text{gh}(\mathcal{L}_{\alpha, \alpha_i, P})} \approx \left(\sqrt{\frac{\beta_2}{n-1}} \right)^{1-1/n} \cdot \frac{\sqrt{n\beta_1^{1/n}}}{\gamma}.$$

In Section 6 we analyse this expression for $f(\alpha_{\text{opt}})$ and show that, for at least the optimal smooth twins, the corresponding lattice vector is either the shortest vector or one of the shortest. A priori computing this α_{opt} requires knowledge of the smooth rational itself since the terms use it and their factorisations. However by estimating β_1 and β_2 one can choose α approximately. It might not give the absolute shortest vector among all α 's but it will still correspond to a short vector. One can also pick several α over some range in the hope that one is close to α_{opt} .

The straightforward strategy to find smooth twins is now as follows: first one picks a smoothness bound B and uses the factor base P_B . Then one makes a choice of α , which is hopefully close to α_{opt} (from Proposition 5.5), before constructing the full lattice $\mathcal{L} = \mathcal{L}_{\alpha, \alpha_i, P_B}$. Then one computes short vectors in $\mathcal{L}$ using the techniques from §4.1 (either lattice sieving, BKZ reduction and/or dimensions for free depending on the GH ratio). Finally, for each short vector $\mathbf{v} = \mathbf{B}_{\alpha, \alpha_i, P} \mathbf{x}$, one computes $a = \prod_{i: x_i > 0} p_i^{x_i}$ and $b = \prod_{i: x_i < 0} p_i^{-x_i}$ and checks if $|a - b| = 1$.

Remark 1. We note a similarity between the CHM algorithm (see Section 2) and our sieving-based SVP solver. Modulo the multiplicative versus additive subtlety (since we work with logarithms), the procedure in the CHM algorithm and lattice sieving correspond to the same process – yet the two algorithms take different paths to find large smooth twins. The CHM method starts with small pairs that

⁷Two approximations are made here: (1) is to remove the 1 from Lemma 5.4 which contributes negligibly to the volume; and (2) is the Gaussian heuristic approximation from Heuristic 4.1.

are already very close and constructs pairs of increasing size. Whereas the lattice sieve starts with large pairs that are far apart and reduces their difference.

While the optimal twins will typically correspond to the shortest vectors the same cannot be said about the smaller twins simply because there are more of them. So one will not be able to find these twins by finding the shortest vector in the full prime number lattice. Furthermore, in high dimension when $\|\mathbf{v}\|/\text{gh}(\mathcal{L}) > 1$, and especially above $\sqrt{4/3}$, recovering $\mathbf{v}$ can quickly become infeasible. Fortunately there are a few some solutions to this problem which we highlight here.

Guessing. Instead of working with the full lattice to find B -smooth twins, one can guess a set Q of prime factors that do not appear in the smooth twin and work with the factor base $P = P_B \setminus Q$. The idea is that a B -smooth twin typically only uses a small fraction of primes from P_B in its factorisation. More precisely, with this factor base P we work in a sublattice of the full lattice by removing all vectors that correspond to smooth rationals containing a factor in Q . If the guess is correct this decreases both the lattice dimension as well as the ratio $\|\mathbf{v}\|/\text{gh}(\mathcal{L})$.

Remark 2. By splitting the factor base into a fixed and a guessing part one could interpret each guess as a Bounded Distance Decoding problem (BDD) in some fixed lattice. Preprocessing [21] could allow a low amortised cost for many BDD instances [27], which we leave as future work.

Dimensions for free. We can alternatively reduce the lattice sieving dimension with the dimensions for free technique as explained in Section 4. Even if the vector is not unusually short, one could still hope to find it with this technique at a lower probability. We will refer to the number of dimensions we decrease the lattice by as the amount of *lifting* we perform. Just as for the guessing strategy varying the amount of lifting gives a trade-off between success probability and cost.

6. ANALYSIS

We analyse the resulting expression from Corollary 5.6 to determine the shortness of vectors in the full prime number lattice corresponding to optimal smooth twins. In other words we make a blanket estimate of $r = e^{2e\sqrt{B}}$ for optimal twins (coming from Theorem 1.1) and analyse $\|\mathbf{v}\|/\text{gh}(\mathcal{L}_{\alpha_{\text{opt}}, \alpha_i, P_B})$. We give both theoretical and experimental analyses for various continuous choices of α_i (in terms of $\log(p_i)$).

Recall the expressions for β_1 , β_2 and γ which are used in the GH ratio as

$$(7) \quad \beta_1 = \log^2(a/b), \quad \beta_2 = \sum_{i=1}^n (x_i \alpha_i)^2 \quad \text{and} \quad \gamma = \sqrt{\frac{n}{2\pi e}} \left[\sqrt{\sum_{i=1}^n \left(\frac{\log(p_i)}{\alpha_i} \right)^2} \prod_{i=1}^n \alpha_i \right]^{\frac{1}{n}},$$

where with our choice of factor base $P = P_B$ we have $n = \pi(B)$. The expression β_1 (and more precisely $\beta_1^{1/n}$) is straightforward to analyse.

Lemma 6.1. *Assuming the optimal twin estimate for smooth twins, $r = e^{2e\sqrt{B}}$, we have $\beta_1^{1/n} \rightarrow 1$ as $n \rightarrow \infty$.*

Proof. Start with this asymptotically more precise statement $\beta_1 = 1/r^2[1+O(1/r)]$. Using the Taylor series for $\log(1+x)$ for small x and the prime number theorem, $B = n \log(n) + O(n \log(\log(n)))$, we get $\log(\beta_1^{1/n}) \rightarrow 0$ as $n \rightarrow \infty$. $\square$

6.1. Theoretical analysis with $\alpha_i = \log(p_i)$. Now we deal with analysing β_2 and γ . This is non-trivial to do generically, i.e. without an explicit choice of α_i . The most challenging aspect in doing this generically is analysing β_2 in its entirety as well as the sum $\sum_{i=1}^n ((\log(p_i)/\alpha_i)^2)$ in γ . For certain choices of α_i these quantities can be analysed. In particular we make a choice of $\alpha_i = \log(p_i)$ for this analysis.

Lemma 6.2. *Taking $\alpha_i = \log(p_i)$, as $n \rightarrow \infty$ we have*

$$\gamma \sim \sqrt{\frac{n}{2\pi e}} \left(\log(n) + \log(\log(n)) - 1 \right).$$

Proof. With our choice of α_i we can rewrite γ from (7) as

$$\gamma = \sqrt{\frac{n}{2\pi e}} \left(\sqrt{n} \prod_{j=1}^n \log(p_j) \right)^{1/n}.$$

By Lemma A.1 we have $(\prod_{i=1}^n \log(p_i))^{1/n} \sim \log(B) - 1$. Since $n^{1/2n} \rightarrow 1$ and (once again) $B \sim n \log(n)$ we obtain the desired result. $\square$

Lemma 6.3. *Taking $\alpha_i = \log(p_i)$ and the optimal twin estimate, we have*

$$\frac{16e^2 B}{n} \leq \beta_2 \leq 4 \left(2e\sqrt{B} + \frac{1}{r} \right)^2.$$

Proof. Recall that $\beta_2 = \sum_{i=1}^n x_i^2 \log^2(p_i)$ and, by Lemma 5.3, we have the inequality

$$(8) \quad \frac{(\sum_{i=1}^n |x_i \log(p_i)|)^2}{n} \leq \beta_2 \leq \left(\sum_{i=1}^n |x_i \log(p_i)| \right)^2.$$

One can calculate that $\sum_{i=1}^n |x_i \log(p_i)| = \log(r) + \log(r+1)$. Using the optimal twin estimate, this is bounded below by $4e\sqrt{B}$ and bounded above by $4e\sqrt{B} + 2/r$. Substituting these bounds in (8) gives the desired lower and upper bounds. $\square$

Proposition 6.4. *Let $\mathbf{v} \in \mathcal{L} = \mathcal{L}_{\alpha, \alpha_i, P_B}$ be the lattice vector corresponding to an optimal B -smooth twin $(r, r+1)$ with the optimal choice $\alpha = \alpha_{\text{opt}}$ and weights $\alpha_i = \log(p_i)$. Then we have*

$$\frac{\|\mathbf{v}\|}{\text{gh}(\mathcal{L}_{\alpha, \alpha_i, P_B})} = O\left(\frac{1}{\sqrt{\log(n)}}\right).$$

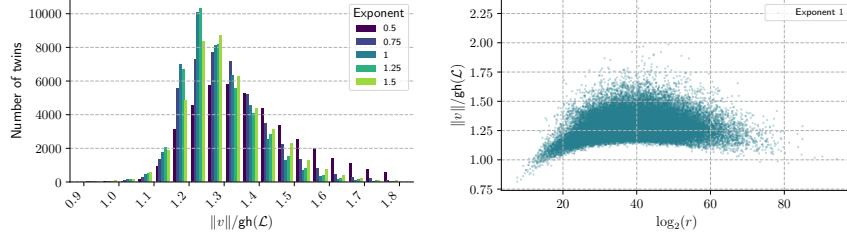
Proof. Combining Lemma 6.1 and Lemma 6.2 we get

$$\frac{\sqrt{n\beta_1^{1/n}}}{\gamma} \sim \frac{\sqrt{2\pi e}}{\log(n) + \log(\log(n)) - 1},$$

and Lemma 6.3 gives an upper bound for $(\sqrt{\beta_2/(n-1)})^{1-1/n}$ which again by the prime number theorem can be solely expressed in terms of n . In particular we have

$$\left(\sqrt{\frac{\beta_2}{n-1}} \right)^{1-1/n} = O\left(\sqrt{\log(n)}\right).$$

Combining these together with Corollary 5.6 gives the intended result. Moreover with the lower bound in Lemma 6.3 we also have $\|\mathbf{v}\|/\text{gh}(\mathcal{L}) = \Omega(1/(n \log(n))^{1/2})$. $\square$



(a) Comparison of the GH ratios achieved for varying exponents e in $\alpha_i = \log^e(p_i)$. (b) Scatter plot of the GH ratio versus the bit size of the 199-smooth twins.

Figure 1. Plots on the set of strictly 199-smooth twins with $\alpha = \alpha_{\text{opt}}$.

6.2. Experimental analysis with different α_i . The theoretical analysis suggests that, for large enough B , the lattice vector corresponding to an optimal B -smooth twin is unusually short and one should be able to find it with BKZ. However this abstracts the constant in the big O term from the generic bounds (notably from β_2) and it is unclear from this analysis whether the ratio is less than or greater than 1 for small B . So we conduct experimental analysis to assess their GH ratios $\|\mathbf{v}\|/\text{gh}(\mathcal{L})$ for *all* B -smooth twins. We will observe that for the largest twins we are in the scenario where either $\|\mathbf{v}\|/\text{gh}(\mathcal{L}) \approx 1$ or $1 < \|\mathbf{v}\|/\text{gh}(\mathcal{L}) < \sqrt{4/3}$ – so one needs to do lattice sieving to find them.

We also experiment with more generic weights $\alpha_i = \log^e(p_i)$ which is non-trivial to analyse theoretically primarily due to the more complex β_2 analysis. The influence of α_i is considered further in §7.2. For now we consider the GH ratio $\|\mathbf{v}\|/\text{gh}(\mathcal{L})$ for several exponents e in $\alpha_i = \log^e(p_i)$ on the set of strictly 199-smooth twins. We think our set of such twins is complete, see Conjecture 7.1, and it should therefore give a good representation.

In Figure 1a we see that the exponents $e = 1$ or $e = 1.25$ lead to the most twins with a small GH ratio. Among all strictly 199-smooth twins the ratio $\|\mathbf{v}\|/\text{gh}(\mathcal{L}_{\alpha_{\text{opt}}, \log(p_i), P_B})$ is on average 1.28 and no larger than 2.3. When minimising the ratio with respect to the exponent e (through numerical computations) the average and largest ratio among these twins is 1.25 and 2.2 respectively. The twins with a larger ratio typically contain an unusually large prime power due to the larger contribution from $(x_i \alpha_i)^2$. For instance this is the case for the 199-smooth twin $(r, r+1)$ where $r = 107^6 - 1$ which clearly has a large power of 107. In Figure 1b we see a dependency between the GH ratio and the size of the twins. Notably the ratio for larger twins becomes better, which is in line with our theoretical analysis.

Non-continuous weights. Instead of continuous choices of α_i (in terms of $\log(p_i)$) one might want non-continuous weights: e.g. $\alpha_1 = \log(p_1)/\eta$, for some $\eta > 1$, and $\alpha_i = \log(p_i)$ for $i \geq 2$. This might be necessary when one needs to (say) find smooth twins with a large power of a certain prime. We discuss this further in §7.2.

Remark 3. The plots in Figure 1 use the optimal $\alpha = \alpha_{\text{opt}}$ for each twin which does not correspond to the same α across the board. For a single α the largest ratio will far exceed 2. Moreover α influences the size of the resulting twin that one gets as a shortest vector.

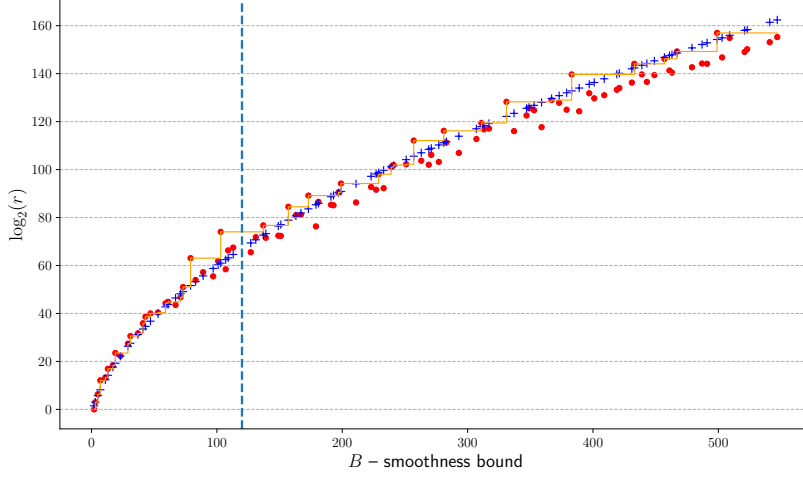


Figure 2. Comparing the numerical estimates for optimal B -smooth twins (marked with a blue plus) against known largest twins with $B \leq 547$. The red dots mark the sizes of the largest *strictly* B -smooth twins, while the orange line indicates the cumulative largest twin.

7. SMOOTH TWIN RESULTS

We report on results using the prime number lattice. Experiments ran on the PlaFRIM cluster using CPU nodes (64 ZEN2 cores, 2.35GHz, 256GB RAM) and GPU nodes (64 ZEN3 cores, 2.60GHz, 2 NVIDIA A100 GPUs, 512GB RAM). We used G6K [5] and its GPU extension [22] for lattice sieving⁸. These tools, developed for lattice-based cryptanalysis, have achieved SVP records up to dimension 180 [22].

7.1. Optimal twins from CPU sieving. Using heuristic estimates from §3.1, we search the full lattice for large, heuristically optimal twins. From Proposition 5.5, the optimal α for a B -smooth twin $(r, r+1)$ is $\alpha_{\text{opt}} \approx \sqrt{\beta_2/(n-1)} \cdot r$. With weights $\alpha_i = \log(p_i)$, $\sqrt{\beta_2/(n-1)} = O(\sqrt{\log(n)})$ (Proposition 6.4). For an unknown b -bit twin, we approximate $\alpha \approx \mu \sqrt{\log(n)} 2^b$ for small μ . Experiments suggest $\sqrt{\beta_2/(n-1)}$ remains small for other values $\alpha_i = \log^e(p_i)$.

For example take $B = 200$, the optimal twin is estimated at ≈ 91.14 bits (Table 1). Using $\alpha = 2^{94}$ and weights $\alpha_i = \log(p_i)$ in $\mathcal{L}_{2^{94}, \log(p_i), P_{200}}$, the 6th shortest vector yields a previously unknown 95-bit 199-smooth twin.

$$(9) \quad \begin{aligned} r &= 3^2 \cdot 5^2 \cdot 7^5 \cdot 11^3 \cdot 59 \cdot 71^2 \cdot 101 \cdot 127 \cdot 173^2 \cdot 197 \cdot 199 \text{ and} \\ r+1 &= 2^{10} \cdot 13 \cdot 17^2 \cdot 23 \cdot 37^2 \cdot 41 \cdot 47 \cdot 61 \cdot 79 \cdot 107^2 \cdot 113^2 \cdot 137. \end{aligned}$$

Exceeding the estimate, this twin is likely optimal. With $\alpha_{\text{opt}} \approx 2^{96.324}$, this twin corresponds to the 3rd shortest vector in $\mathcal{L}_{\alpha_{\text{opt}}, \log(p_i), P_{200}}$.

The lattice dimension in this example is $\pi(200) = 46$, well below SVP records. An SVP call costs $2^{0.292\pi(B)+o(\pi(B))}$ (Section 4.1), which is significantly less than the $2^{\pi(B)+o(\pi(B))}$ lower bound Pell equations cost (Section 2). This saving is the

⁸Available at <https://github.com/fplll/g6k> and <https://github.com/WvanWoerden/G6K-GPU-Tensor> respectively

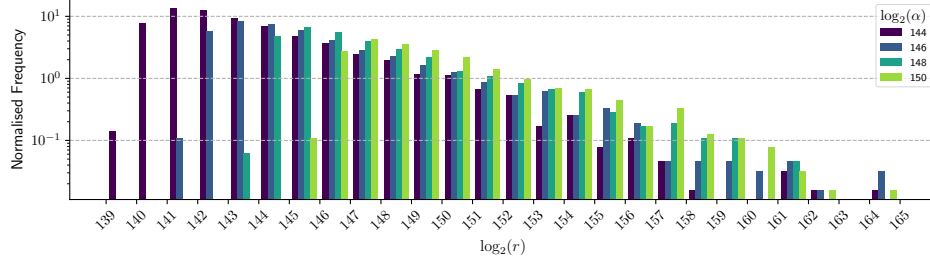


Figure 3. Histogram of the normalised number of smooth twins $(r, r + 1)$ found of each bit size versus a varying scalar α in the normal regime.

main contributing factor that allows us to find larger optimal twins. Figure 2 compares the strictly largest B -smooth twins against our estimates from §3.1.

Using CPU sieving in the lattice $\mathcal{L}_{2^{182}, \log(p_i), P_{647}}$, we found a conjectured optimal 180-bit 647-smooth twin:

$$\begin{aligned} r &= 2^4 \cdot 7^3 \cdot 17^2 \cdot 23 \cdot 47 \cdot 61 \cdot 131^4 \cdot 139^2 \cdot 163 \cdot 179 \cdot 197 \cdot 223 \cdot 257 \cdot 293 \\ &\quad \cdot 379 \cdot 443 \cdot 563 \cdot 569 \cdot 617 \cdot 647, \text{ and} \\ r + 1 &= 3 \cdot 5 \cdot 13 \cdot 19 \cdot 31^2 \cdot 41 \cdot 59 \cdot 71 \cdot 127 \cdot 137 \cdot 151^2 \cdot 233 \cdot 263 \cdot 307^2 \cdot 349^3 \\ &\quad \cdot 431 \cdot 467 \cdot 509 \cdot 523 \cdot 601 \cdot 643. \end{aligned}$$

7.2. Influence of parameters. We examine the influence of the parameters using CPU-based G6K with $B = 691$ ($\pi(B) = 125$) and a factor base $P \subseteq P_{691}$. We performed 64 seeded trials for each parameter set. In the histograms *normalised frequency* represents the total number of unique (unless otherwise stated) smooth twins found divided by 64. By default we fix the weights to be $\alpha_i = \log(p_i)^e$ with $e = 1$, leave out $k = 15$ primes in the guessing phase and lift $l = 5$ dimensions with the lifting strategy (sieving dimension $125 - k - l = 105$). Each trial changes some of these parameters and takes a few hours on 16 cores. We study two regimes: *normal* (maximising total number of twins found, $\alpha = 2^{150}$) and *power-of-two* (seeking large powers of two, $\alpha = 2^{128}$). In the latter, we scale $\alpha_1 = \log(2)/\eta$ with $\eta = 20$.

Size of found twins (α). In Figure 3 we consider the influence of α on the size of smooth twins $(r, r + 1)$ that are found in the normal regime. We varied $\log_2(\alpha)$ from 144 to 150 with increments of 2. Consistent with Proposition 5.5, twin size scales with α , peaking in the interval $\log_2(r) \in [\log_2(\alpha) - 3, \log_2(\alpha) - 2]$. Larger α yield fewer twins in total, reflecting the scarcity of larger B -smooth twins.

Diagonal exponent (e). We varied the exponent e in $\alpha_i = \log(p_i)^e$ from 0.5 to 1.5 (step 0.125) in both regimes. Results are in Figure 4. In the normal regime (plotting $\log_2(r)$), $1.125 \leq e \leq 1.375$ is optimal, peaking at $e = 1.25$. Larger e slightly reduces the average twin size relative to α . In the power-of-two regime (plotting $\text{val}_2(r(r + 1))$), $0.875 \leq e \leq 1.25$ is optimal, peaking at $e = 1.125$.

Guessing versus lifting (k, l). We compare guessing (k) and lifting (l) by varying $k \in \{5, 10, 15, 20\}$ with $k + l = 20$, balancing lattice dimension against success probability⁹. In the normal regime (Figure 5a), counting duplicates favors (10, 10)

⁹The case $k = 0$ is not considered as otherwise each trial would use exactly the same lattice.

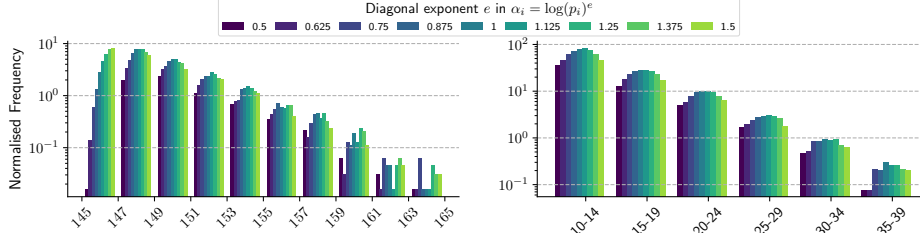
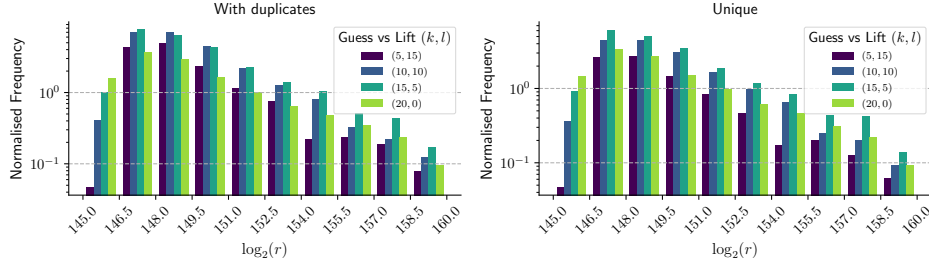
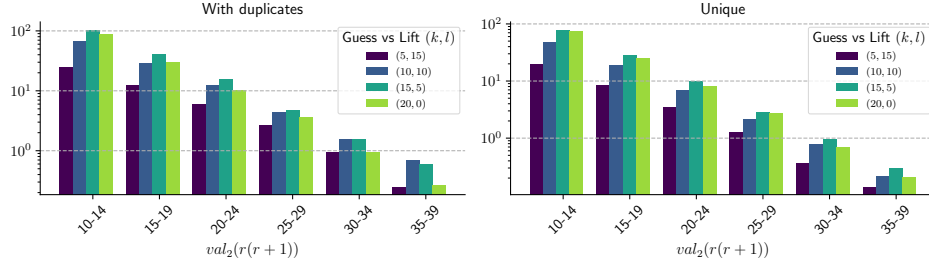


Figure 4. Normalised number of smooth twins found when changing the weight $\alpha_i = \log_2(p_i)^e$. The left (right) plot is in the normal (power-of-two) regime and plotted against the size $\log_2(r)$ (power-of-two exponent $\text{val}_2(r(r+1))$) respectively of each smooth twin $(r, r+1)$.



(a) Normal regime.



(b) Power-of-two regime.

Figure 5. Comparing guessing versus lifting (d4f); varying the guessing and lifting dimensions (k, l) while keeping their sum $k + l = 20$ fixed.

or $(15, 5)$ over $(5, 15)$ or $(20, 0)$. For unique twins, guessing dominates, with $(15, 5)$ being optimal. Lifting appears to disproportionately find the same unusually short twins repeatedly. Similarly, in the power-of-two regime (Figure 5b), $(15, 5)$ is optimal for duplicates, while $(20, 0)$ performs nearly as well for unique twins.

Scaling and exponent trick for finding large powers of two. To find twins with large powers of two (see Section 8 for motivation), we test two methods. The first method replaces $p_1 = 2$ with $p_1 = 2^f$ ($f > 1$), ensuring found twins satisfy $f \mid \text{val}_2(r(r+1))$. Increasing f makes the lattice sparser, potentially aiding the search, but misses twins where $f \nmid \text{val}_2(r(r+1))$. The second method decreases α_1 by a factor $\eta > 1$, reducing the norm penalty for large powers of two. This makes the lattice

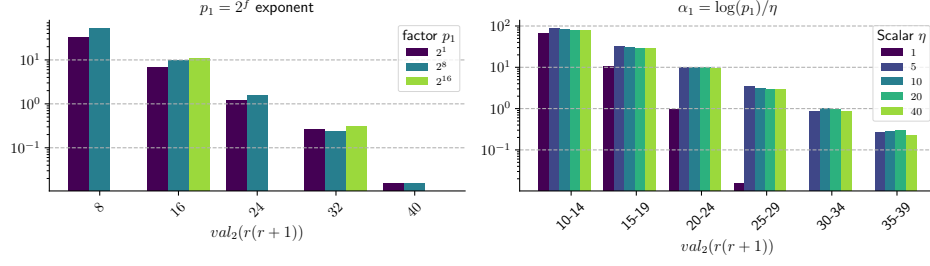


Figure 6. Influence of the power-of-two exponent f and the scalar η on finding smooth twins with a large power of two.

denser and requires balancing the choice of η . We test $\alpha_1 = \log(p_1)/\eta$ for $\eta \in \{1, 5, 10, 20, 40\}$.

Results in Figure 6 show that using $p_1 = 2^f$ yields more twins with $f | \text{val}_2(r(r+1))$. Downscaling α_1 with $\eta = 5$ finds twins with $\text{val}_2(r(r+1)) \geq 35$, whereas $\eta = 1$ finds essentially none ≥ 25 . Increasing η beyond 5 has little effect.

7.3. Results from GPU lattice sieving. We now detail our GPU experiments. The found twins are available at [34]. Memory constraints limited the effective sieving dimension to 137. Sieving at this dimension takes half a day on one PlaFRIM node. With this, using $B = 773$ ($\pi(B) = 137$), we found a 195-bit 743-smooth twin (10) and a 196-bit 751-smooth twin (1). Both align with heuristic optimality estimates.

$$\begin{aligned}
 (10) \quad r &= 7 \cdot 19^3 \cdot 23 \cdot 37 \cdot 47^2 \cdot 79 \cdot 89 \cdot 97^3 \cdot 103 \cdot 211 \cdot 229^2 \cdot 271^2 \cdot 283 \cdot 307 \\
 &\quad \cdot 401^2 \cdot 503 \cdot 541 \cdot 557 \cdot 631 \cdot 643, \text{ and} \\
 r+1 &= 2^{11} \cdot 5^2 \cdot 11 \cdot 13 \cdot 43^2 \cdot 53 \cdot 59 \cdot 131 \cdot 137 \cdot 149 \cdot 163 \cdot 223^3 \cdot 281 \cdot 313 \\
 &\quad \cdot 337 \cdot 353 \cdot 409 \cdot 449 \cdot 547 \cdot 563 \cdot 653 \cdot 709 \cdot 743.
 \end{aligned}$$

For larger $B > 773$ ($\pi(B) > 137$), we use a combination of lifting and guessing. For instance with $B = 823$ and lifting $l = 5$ dimensions, we found a 200-bit 809-smooth twin (estimated optimal size: 202 bits).

$$\begin{aligned}
 r &= 2^6 \cdot 17^3 \cdot 23 \cdot 29^2 \cdot 47 \cdot 53^2 \cdot 59 \cdot 67^2 \cdot 107 \cdot 131 \cdot 137 \cdot 163 \cdot 193 \cdot 271 \\
 &\quad \cdot 277 \cdot 283 \cdot 313 \cdot 359 \cdot 397 \cdot 433 \cdot 523 \cdot 751 \cdot 787 \cdot 809, \text{ and} \\
 r+1 &= 3^{13} \cdot 5^2 \cdot 7^2 \cdot 11 \cdot 19 \cdot 41^2 \cdot 71 \cdot 73 \cdot 89 \cdot 139 \cdot 179 \cdot 263^2 \cdot 269 \cdot 557^2 \cdot 577 \\
 &\quad \cdot 587^2 \cdot 607 \cdot 677 \cdot 683 \cdot 709 \cdot 733.
 \end{aligned}$$

This twin can also be found with guessing $k = 5$. One should remove a subset $Q \subseteq P_B$ devoid of factors of $r(r+1)$ and small primes. The success probability for choosing $Q \subseteq P_B \setminus P_{300}$ with $\#Q = 5$ is ≈ 0.3224 .

Increased lifting and guessing finds larger twins but potentially sacrifices optimality. For instance for $B = 997$ ($\pi(B) = 168$), we ran experiments for varying $k+l = 168 - 137 = 31$ and $k > l$ following §7.2. With $k = 18, l = 11$, we found a 213-bit 997-smooth twin (2), which is far from optimal. Experiments with larger B and increased lifting/guessing yielded no larger twins.

	CHM _B	CHM ₅₄₇	SVP
$B = 150 :$	73,694	74,006 (+312)	74,007 (+1)
$B = 200 :$	346,192	348,840 (+2,648)	348,865 (+25)
$B = 250 :$	826,750	835,613 (+8,863)	835,880 (+267)
$B = 300 :$	2,316,631	2,350,100 (+33,469)	2,353,597 (+3,497)
$B = 350 :$	—	5,431,970	5,456,952 (+24,982)
$B = 400 :$	—	11,840,978	11,980,453 (+139,475)

Table 2. Cumulative number of B -smooth twins found. The subscript in CHM_B denotes applying the CHM algorithm with B as input.

Cryptographic smooth twins. Isogeny-based cryptography typically requires 256-bit twins. While Section 8 discusses relaxing this requirement, we address here the challenge of finding such twins. Table 1 suggests needing $B \approx 1250$ ($\pi(B) = 204$) for 256-bit twins. Current SVP records reach dimension 200 (with lifting) [5, 22, 41], putting this just beyond reach. Even if one incorporates lifting and guessing, one would need a large amount of trials in order to find the desired smooth twin – adding to the computational cost.

7.4. Towards enumerating all B -smooth twins. The main obstruction to finding all B -smooth twins with our lattice approach is that not all smooth twins correspond to short vectors in the full lattice (see Figure 1). To find them through SVP instances a sufficient (and possibly large) amount of guessing is required (we discuss this further in Remark 4).

Alternatively one can use the lattice approach to add previously undiscovered twins to a known set of B -smooth twins. One still needs to incorporate guessing to find the obscure twins. If the known set is large then one can minimise the amount of necessary searching and plausibly conjecture the complete set of B -smooth twins.

A large proportion of B -smooth twins for $B \leq 547$ was found using the CHM algorithm [10] (Section 2). Notably the larger B -smooth twins are not found with the CHM algorithm, most of which our SVP solver should find with minimal guessing. We extensively searched for B -smooth twins with $B < 400$ and found many new twins. In particular we found 25 new 200-smooth twins including the twin in (9). We did further experiments to find more 200-smooth twins with large guessing amounts. After 1-2 weeks of effort no further twins were found. So we conjecture to have the complete set of 200-smooth twins as summarised in Conjecture 7.1, and available at [34].

No formal claims are made for larger B but we note down in Table 2 the cumulative number of B -smooth twins found in this and prior work. We did not extend our experiments to $B = 547$ due to the increasing computational effort.

Conjecture 7.1. *There are exactly 348,865 many 200-smooth twins.*

Remark 4. By Theorem 1.1, the number of distinct prime factors in B -smooth twins is heuristically at most $O(\pi(4e\sqrt{B})) = O(\sqrt{B}/\log(B))$ which for large B is only a small fraction of all $O(B/\log(B))$ primes up to B . Therefore, by doing a very large amount of guessing, with say $k = \pi(B) - O(\sqrt{B}/\log(B))$, we could use the SVP approach alone to find all B -smooth twins. Trying all $\binom{\pi(B)}{k}$ subsets $P \subseteq P_B$

of size $n - k$ and polynomially many distinct α 's, the resulting complexity turns out to be subexponential in B . Despite this potential gain, it is only practical for very large B and not the small B that we have worked with. A similar heuristic analysis can be made with the Pell equation approach. For all square-free discriminants $D \leq e^{4e\sqrt{B}}$, one would early abort solving the equation $x^2 - Dy^2 = 1$ using the continued fraction of $\sqrt{D}$, namely when x becomes larger than $2e^{2e\sqrt{B}}$. This should also give a subexponential complexity, but which only pays off for huge B .

8. APPLICATIONS

8.1. SQIsign parameters. We revisit the original version of SQIsign¹⁰, which we call SQIsign1D, and find new parameters. In particular one works with a prime p such that $2^f \cdot T \mid p^2 - 1$ where f is as large as possible and $T \approx p^{5/4}$ is odd and B -smooth. This condition implies that not all of $p^2 - 1$ needs to be smooth, which differs from the fully smooth twin scenario.

The main idea in this prime search is to adopt a *boosting* strategy and search for a prime of the form $p = p_2(r) = 2r^2 - 1$ for a smooth r . As $p^2 - 1 = 4r^2(r - 1)(r + 1)$ we need smooth divisors in $r - 1, r + 1$. We also ensure that r has a large power of two (which is amplified by the squaring in $p_2 + 1$).

Sieve-and-boost approach. Instead of doing an exhaustive search on smooth r which is expensive, one can do a tailored exhaustive search over integers of the form¹¹ $r = 2^{f'} \cdot 3^{g'} \cdot m$ where m is smooth. Then for each such r factorise the smooth parts of $r - 1$ and $r + 1$ and see if this gives the necessary amount of smoothness. This strategy was adopted for the first round of the NIST submission [11].

Twin-and-boost approach. If $(r, r \pm 1)$ is a smooth twin with $2^{f'} \mid r$ and $p = p_2(r)$, then $p^2 - 1$ has a smooth factor $4r^2(r \pm 1) \approx p^{3/2}$. If $f = 2f' + 1 \leq \log_2(p)/4$ then $2^f \cdot T \leq p^{3/2}$; so one simply needs to check primality of p . For larger f one requires more small prime factors in $r \mp 1$. Depending on how much larger f is relative to $\log_2(p)/4$, the probability that one has these factors is not too small [6].

So a sufficiently good *smooth twin oracle* can give SQIsign1D parameters. This observation was first noted in [10] but they were unable to instantiate the oracle with the CHM algorithm. On the other hand we can instantiate this properly with our lattice approach aimed at finding smooth twins with a large power of two (see §7.2). In particular we found the exceptional prime $p = p_2(r)$ with factorisations

$$\begin{aligned} p + 1 &= 2^{63} \cdot 5^4 \cdot 23^4 \cdot 67^2 \cdot 71^2 \cdot 73^2 \cdot 89^2 \cdot 113^2 \cdot 137^4 \cdot 163^2 \cdot 229^2 \cdot 263^2 \cdot 293^2, \text{ and} \\ p - 1 &= 2 \cdot 3 \cdot 7^2 \cdot 11 \cdot 13^2 \cdot 31 \cdot 47^2 \cdot 79^2 \cdot 103 \cdot 151 \cdot 241^2 \cdot 353 \cdot 367 \cdot 389 \cdot 449 \\ &\quad \cdot 463 \cdot 499 \cdot 50355971 \cdot 1032403334060991048097384477. \end{aligned}$$

With a significantly smaller smoothness bound (on the smooth part of $p^2 - 1$) compared to the parameters in [11], this offers a better signing performance in accordance to the SQIsign1D estimator sage script provided in [15]. In particular one could theoretically get a modest 19.85% improvement for signing. However we note that this will not compete with the second round submission [1] to NIST's signature call. So we present this prime to expand diversity of SQIsign1D parameters.

¹⁰For accessible resources on SQIsign and its variants we refer to [24, 25, 30, 11, 14, 7, 35, 1].

¹¹The inclusion of the power of three is not mandatory but it simplifies certain steps in the signing procedure and was a design choice made in the original NIST submission.

APPENDIX A. ASYMPTOTICS FOR THE GEOMETRIC MEAN OF THE LOGARITHMIC PRIMES

We provide an elementary proof of the following statement. The proof is adapted discussion from a stack exchange post¹² in order to get an asymptotically precise statement. To our knowledge this result is not publicly recorded.

Lemma A.1. *As $x \rightarrow \infty$ we have*

$$\left(\prod_{p \leq x} \log(p) \right)^{\frac{1}{\pi(x)}} = \log(x) - 1 - o(1),$$

where the product is over the primes less than x .

Proof. We consider $\sum_{p \leq x} \log(\log(p))$ which by Abel summation is

$$\sum_{p \leq x} \log(\log(p)) = \pi(x) \log(\log(x)) - \int_2^x \frac{\pi(t)}{t \log(t)} dt.$$

Using the prime number theorem in the integral term we have

$$\begin{aligned} \int_2^x \frac{\pi(t)}{t \log(t)} dt &= \int_2^x \frac{t / \log(t) + O(t / \log^2(t))}{t \log(t)} dt \\ &= \frac{x}{\log^2(x)} + O\left(\frac{x}{\log^3(x)}\right) = \frac{\pi(x)}{\log(x)} + O\left(\frac{\pi(x)}{\log^2(x)}\right). \end{aligned}$$

So we have

$$\sum_{p \leq x} \log(\log(p)) = \pi(x) \left(\log(\log(x)) - \frac{1}{\log(x)} - O\left(\frac{1}{\log^2(x)}\right) \right)$$

and exponentiating gives

$$\prod_{p \leq x} \log(p) = \left(\log(x) \exp \left\{ -\frac{1}{\log(x)} - O\left(\frac{1}{\log^2(x)}\right) \right\} \right)^{\pi(x)}.$$

Finally, using the elementary fact that $e^{f(x)} = 1 + f(x) + o(1)$ when $f(x) = o(1)$, we obtain the intended result. $\square$

REFERENCES

- [1] M. A. Aardal, G. Adj, D. F. Aranha, A. Basso, I. A. Canales Martínez, J. Chávez-Saab, M. Corte-Real Santos, P. Dartois, L. De Feo, M. Duparc, J. K. Eriksen, T. B. Fouotsa, D. L. Gazzoni Filho, B. Hess, D. Kohel, A. Leroux, P. Longa, L. Maino, M. Meyer, K. Nakagawa, H. Onuki, L. Panny, S. Patranabis, C. Petit, G. Pope, K. Reijnders, D. Robert, F. Rodríguez-Henríquez, S. Schaeffler, and B. Wesolowski, *SQIsign*, Tech. report, National Institute of Standards and Technology, 2025, Version 2.0, available at <https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/spec-files/sqisign-spec-round2-web.pdf>.
- [2] L. A. Adleman, *Factoring and Lattice Reduction*, Manuscript, 1995.
- [3] D. Aggarwal, D. Dadush, O. Regev, and N. Stephens-Davidowitz, *Solving the shortest vector problem in 2^n time using discrete Gaussian sampling*, STOC, ACM Press, 2015, pp. 733–742.
- [4] D. Aggarwal, D. Dadush, and N. Stephens-Davidowitz, *Solving the closest vector problem in 2^n time—the discrete Gaussian strikes again!*, FOCS, IEEE, 2015, pp. 563–582.

¹²<https://math.stackexchange.com/q/679344>

- [5] M. R. Albrecht, L. Ducas, G. Herold, E. Kirshanova, E. W. Postlethwaite, and M. Stevens, *The general sieve kernel and new records in lattice reduction*, EUROCRYPT, Lecture Notes in Computer Science, vol. 11477, Springer, 2019, pp. 717–746.
- [6] W. D. Banks and I. E. Shparlinski, *Integers with a large smooth divisor*, arXiv preprint math/0601460 (2006).
- [7] A. Basso, P. Dartois, L. De Feo, A. Leroux, L. Maino, G. Pope, D. Robert, and B. Wesolowski, *SQISign2D–West: The fast, the small, and the safer*, ASIACRYPT, Lecture Notes in Computer Science, vol. 15486, Springer, 2024, pp. 339–370.
- [8] A. Basso and L. Maino, *POKÉ: A compact and efficient PKE from higher-dimensional isogenies*, EUROCRYPT, Lecture Notes in Computer Science, vol. 15602, Springer, 2025, pp. 94–123.
- [9] A. Becker, L. Ducas, N. Gama, and T. Laarhoven, *New directions in nearest neighbor searching with applications to lattice sieving*, SODA, ACM-SIAM, 2016, pp. 10–24.
- [10] G. Bruno, M. Corte-Real Santos, C. Costello, J. K. Eriksen, M. Meyer, M. Naehrig, and B. Sterner, *Cryptographic smooth neighbors*, ASIACRYPT, Lecture Notes in Computer Science, vol. 14444, Springer, 2023, pp. 190–221.
- [11] J. Chávez-Saab, M. Corte-Real Santos, L. De Feo, J. K. Eriksen, B. Hess, D. Kohel, A. Leroux, P. Longa, M. Meyer, L. Panny, S. Patranabis, C. Petit, F. Rodríguez-Henríquez, S. Schaeffler, and B. Wesolowski, *SQISign*, Tech. report, National Institute of Standards and Technology, 2023, Version 1.0, available at <https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/sqisign-spec-web.pdf>.
- [12] J. B. Conrey, M. A. Holmstrom, and T. L. McLaughlin, *Smooth neighbors*, Experimental Mathematics **22** (2013), no. 2, 195–202.
- [13] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, *On the Lambert W function*, Advances in Computational mathematics **5** (1996), 329–359.
- [14] M. Corte-Real Santos, J. K. Eriksen, M. Meyer, and K. Reijnders, *AprèsSQI: Extra fast verification for SQISign using extension-field signing*, EUROCRYPT, Lecture Notes in Computer Science, vol. 14651, Springer, 2024, pp. 63–93.
- [15] M. Corte-Real Santos, J. K. Eriksen, M. Meyer, and F. Rodríguez-Henríquez, *Finding practical parameters for isogeny-based cryptography*, CiC **1** (2024), no. 3.
- [16] C. Costello, *B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion*, ASIACRYPT, Lecture Notes in Computer Science, vol. 12492, Springer, 2020, pp. 440–463.
- [17] C. Costello, M. Meyer, and M. Naehrig, *Sieving for twin smooth integers with solutions to the Prouhet-Tarry-Escott problem*, EUROCRYPT, Lecture Notes in Computer Science, vol. 12696, Springer, 2021, pp. 272–301.
- [18] N. G. de Bruijn, *The asymptotic behaviour of a function occurring in the theory of primes*, Journal of the Indian Mathematical Society. New Series **15** (1951), 25–32.
- [19] K. Dickman, *On the frequency of numbers containing prime factors of a certain relative magnitude*, Arkiv for matematik, astronomi och fysik **22** (1930), no. 10, A–10.
- [20] L. Ducas, *Shortest vector from lattice sieving: a few dimensions for free*, EUROCRYPT, Lecture Notes in Computer Science, vol. 10820, Springer, 2018, pp. 125–145.
- [21] L. Ducas, T. Laarhoven, and W. van Woerden, *The randomized slicer for CVPP: sharper, faster, smaller, batchier*, PKC, Lecture Notes in Computer Science, vol. 12111, Springer, 2020, pp. 3–36.
- [22] L. Ducas, M. Stevens, and W. van Woerden, *Advanced lattice sieving on GPUs, with tensor cores*, EUROCRYPT, Lecture Notes in Computer Science, vol. 12697, Springer, 2021, pp. 249–279.
- [23] P. Erdős, *Problems and results on number theoretic properties of consecutive integers and related questions*, Proc. Fifth Manitoba Conf. on Numerical Mathematics, 1975, pp. 25–44.
- [24] L. De Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski, *SQISign: compact post-quantum signatures from quaternions and isogenies*, ASIACRYPT, Lecture Notes in Computer Science, vol. 12491, Springer, 2020, pp. 64–93.
- [25] L. De Feo, A. Leroux, P. Longa, and B. Wesolowski, *New algorithms for the Deuring correspondence - towards practical and secure SQISign signatures*, EUROCRYPT, Lecture Notes in Computer Science, vol. 14008, Springer, 2023, pp. 659–690.
- [26] A. Granville, *Smooth numbers: computational number theory and beyond*, Algorithmic number theory: lattices, number fields, curves and cryptography **44** (2008), 267–323.

- [27] A. Karenin, E. Kirshanova, J. Nowakowski, and A. May, *Fast slicer for batch-CVP: Making lattice hybrid attacks practical*, ASIACRYPT, Lecture Notes in Computer Science, vol. 16247, Springer, 2025, pp. 100–132.
- [28] D. H. Lehmer, *On a problem of Størmer*, Illinois Journal of Mathematics **8** (1964), no. 1, 57–79.
- [29] A. K. Lenstra and H. W. Lenstra, *The development of the number field sieve*, Lecture Notes in Mathematics, vol. 1554, Springer Science & Business Media, 1993.
- [30] A. Leroux, *Quaternion algebra and isogeny-based cryptography*, Theses, Ecole doctorale de l’Institut Polytechnique de Paris (2022).
- [31] F. Luca and F. Najman, *On the largest prime factor of $x^2 - 1$* , Mathematics of computation **80** (2011), no. 273, 429–435.
- [32] G. Martin, *An asymptotic formula for the number of smooth values of a polynomial*, Journal of Number Theory **93** (1999), 108–182.
- [33] D. Micciancio and S. Goldwasser, *Complexity of lattice problems: a cryptographic perspective*, The Springer International Series in Engineering and Computer Science, vol. 671, Springer Science & Business Media, 2002.
- [34] E. Mulder, B. Sterner, and W. van Woerden, *Zenodo dataset: smooth twins*, January 2026, available at <https://doi.org/10.5281/zenodo.18234089>.
- [35] K. Nakagawa, H. Onuki, W. Castryck, M. Chen, R. Invernizzi, G. Lorenzon, and F. Vercauteren, *SQIsign2D-East: A new signature scheme using 2-dimensional isogenies*, ASIACRYPT, Lecture Notes in Computer Science, vol. 15486, Springer, 2024, pp. 272–303.
- [36] C. P. Schnorr, *Factoring integers and computing discrete logarithms via diophantine approximation*, EUROCRYPT, Lecture Notes in Computer Science, vol. 547, Springer, 1991, pp. 281–293.
- [37] C. P. Schnorr and M. Euchner, *Lattice basis reduction: Improved practical algorithms and solving subset sum problems*, Mathematical programming **66** (1994), 181–199.
- [38] B. Sterner, *Towards optimally small smoothness bounds for cryptographic-sized smooth twins and their isogeny-based applications*, SAC, Lecture Notes in Computer Science, vol. 15516, Springer, 2025, pp. 178–202.
- [39] C. Størmer, *Quelques théorèmes sur l’équation de Pell $x^2 - dy^2 = \pm 1$ et leurs applications*, Christiania Videnskabs Selskabs Skrifter, Math. Nat. K1 (1897), no. 2, 48.
- [40] J. van de Lune and E. Wattel, *On the numerical solution of a differential-difference equation arising in analytic number theory*, Mathematics of Computation **23** (1969), no. 106, 417–421.
- [41] Z. Zhao, J. Ding, and B. Y. Yang, *Sieving with streaming memory access*, IACR Transactions on Cryptographic Hardware and Embedded Systems **2025** (2025), no. 2, 362–384.

UNIVERSITY OF BERGEN, BERGEN, NORWAY
 Email address: `erik.mulder@uib.no`

INRIA AND ÉCOLE POLYTECHNIQUE, INSTITUT POLYTECHNIQUE DE PARIS, PALAISEAU, FRANCE
 & UNIVERSITY OF WATERLOO, WATERLOO, CANADA
 Email address: `bsterner@uwaterloo.ca`

PQSHIELD, AMSTERDAM, THE NETHERLANDS
 Email address: `wessel.vanwoerden@pqshield.com`