Towards Optimally Small Smoothness Bounds for Cryptographic-Sized Twin Smooth Integers and its Isogeny-based Applications

Bruno Sterner

▶ **To cite this version:**

Bruno Sterner. Towards Optimally Small Smoothness Bounds for Cryptographic-Sized Twin Smooth Integers and its Isogeny-based Applications. 2023. hal-04254512

HAL Id: hal-04254512
https://inria.hal.science/hal-04254512

Preprint submitted on 23 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Towards Optimally Small Smoothness Bounds for Cryptographic-Sized Twin Smooth Integers and its Isogeny-based Applications

Bruno Sterner

Inria and Laboratoire d'Informatique de l'École polytechnique (LIX), Institut Polytechnique de Paris, Palaiseau, France
bruno-sydney.sterner@inria.fr

Abstract. We give a new approach for finding large twin smooth integers. Those twins whose sum is a prime are of interest in the parameter setup of certain isogeny-based cryptosystems such as SQISign. The approach to find such twins is to find two polynomials in $\mathbb{Q}[x]$ that split into a product of small degree factors and differ by 1; then evaluate them on a particular smooth integer. This was first explored by Costello, Meyer and Naehrig at EUROCRYPT'21 using polynomials that split completely into linear factors which were found from some Diophantine number theory. The polynomials used in this work split into mostly linear factors with the exception of a few quadratic factors. Some of these linear factors are repeated and so the overall smoothness probability is either better or comparable to that of the prior polynomials. We utilise these polynomials to search for large twin smooth integers whose sum is prime. In particular, the smoothness bound of the 384 and 512-bit instances that we find are significantly smaller than those found in EUROCRYPT'21.

Keywords: Post-quantum cryptography, isogeny-based cryptography, twin smooth integers, extended Euclidean algorithm, SQISign.

1 Introduction

Efficient instances of many new isogeny-based cryptosystems require a large prime p such that p^2-1 is either B-smooth or has a large B-smooth divisor for some small B. Most notably, this includes the digital signature scheme SQISign [19] which was submitted to NIST's recent call for alternative signature schemes [32,11] as part of their on-going effort to standardise post-quantum cryptography [31]. By B-smooth, we mean that each prime divisor is at most B.

This condition on p ensures that supersingular curves over $\mathbb{F}_{p^2}$ and their quadratic twists both have many rational points of small prime orders, which permits efficient isogeny computations[1]. The *smoothness bound*, B, of p^2-1 or its smooth cofactor is the dominant factor in the performance of these cryptosystems [4]. Hence finding parameters that minimise B is vitally important. Having said this, making B as small as one would like is not possible. This is because there exists a theoretical bound for how small B can be [30]. This paper addresses the problem of finding large primes p that reduce the smoothness bound of p^2-1 to something which is close to the theoretical optimum.

One can translate this problem of finding primes p with p^2-1 being smooth in terms of finding twin smooth integers in the sense of the following definition.

Definition 1. *We call a pair of consecutive integers* $(r, r+1)$ B-smooth twins *or* twin B-smooth integers *if their product,* $r\cdot(r+1)$*, is* B*-smooth*[2]*.*

[1] A priori, this requires using $\mathbb{F}_{p^4}$-rational points in order to make sense of the quadratic twist but, working in the Montgomery model and doing x-only arithmetic, all the computation can be done over $\mathbb{F}_{p^2}$ [13, §3].

[2] We drop B from these definitions when B is polynomial in the bitsize of r.

Method	$\log_2(B)$ of smallest smoothness bounds for b-bit primes p			Where
	$b \approx 256$	$b = 384$	$b = 512$	
XGCD over $\mathbb{Z}$	22.7	—	—	[4]
Cyclotomic factors	18.9	24.4	—	[13,18]
PTE sieve	15.0	21.9	27.9	[14]
XGCD over $\mathbb{Q}$[x]	**15.4**	**20.2**	**24.3**	**this work**

Table 1: A comparison of smoothness bounds, B, of $p^2 - 1$ for cryptographic-sized primes p found using prior and our methods. The entries marked with a dash indicate that no experiments have been done.

Numerous applications arise from finding such twin smooth integers including the computation of logarithms of integers [22] and the ABC conjecture [12]. In the context of this work, if the sum of a twin, $p = 2r + 1$, is a prime, then $p^2 - 1 = 4r(r + 1)$ is smooth and we recover a suitable instance isogeny-based applications. Much like for the primes p, there is a theoretical optimum for the smoothness bound of twin smooth integers. We refer to an *optimally small smoothness bound* for twins of a certain size as the minimal smoothness bound, B, such that B-smooth twins of that size exist.

Related work. Broadly speaking, the known techniques to find twin smooth integers can be separated into two categories: constructive and probabilistic methods. The constructive methods [30,12] fix a smoothness bound B and find all or almost all B-smooth twins including those with an optimally small B. The probabilistic methods [13,14] searches for twins of a fixed size and guarantee finding them up to some probability which depends on the choice of B. Thus expecting to find B-smooth twins for an optimally small B is not realistic. Nevertheless, as long as B is not too small, one can expect to find such a twin by looking over a large search space.

Constructive methods. There are only two known approaches that enumerate all or almost all B-smooth twins. One requires solving exponentially many Pell equations [30,22] with respect to B, and the other is a recursive algorithm referred to as CHM [12]. These are the only known approaches for tackling the problem of finding such twins that have an optimally small smoothness bound. However, these algorithms become computationally infeasible when finding instances which are at least 256-bits. The largest twin found using these methods whose sum is prime is a 127-bit prime p such that $p^2 - 1$ is 2^{10}-smooth [6] and was found using the CHM algorithm. Moreover, the analysis done by Bruno et. al. [7, §4.1] suggests that the optimally smallest smoothness bound for a 256-bit twin is around $B \approx 5000$. Even with the current computing resources it is infeasible to run these algorithms for those smoothness bounds. We will refer to a *cryptographic-sized* smooth twin or a prime if they has at least 256-bits.

Probabilistic methods. To counter this hindrance from the constructive methods, one resorts to the probabilistic methods to find such cryptographic-sized twins. This sacrificies the optimal smoothness bound for the ability to find concrete cryptographic-sized twins that could have practical isogeny-based applications. Almost all of the methods that fall into this category use some polynomial evaluation. The high level idea is to find two polynomials $f, g \in \mathbb{Z}[x]$ that differ by an integer C and factorise nicely. Subsequently, one evaluates these polynomials at an integer, such that their evaluation on f and g are divisible by C, to generate twin smooth integers.

Prior to this work, only two classes of polynomial pairs have been used to find such twins: first are the polynomials $f(x) = x^n - 1$, $g(x) = x^n$ [13]; and the second are polynomials f, g that completely split over the integers [14]. The first polynomial pair already differ by 1 so there is no divisibility checked needed in order to find smooth twins. The usage of the latter polynomials is currently the state-of-the-art in terms of minimising the smoothness bound of the twins. The authors that suggested the use of these polynomials ran

experiments to find b-bit primes p for $b \in \{256, 384, 512\}$ such that $p^2 - 1$ is B-smooth for $B \in \{2^{16}, 2^{22}, 2^{28}\}$ (resp.). Their best results are contained in Table 1.

Contributions. In this work we revisit and generalise the polynomial-based probabilistic methods for finding twin smooth integers. Recall that the polynomials used in [14] split completely over the integers. In contrast, we use polynomials f and g that (once again) differ by an integer C and split mostly in to linear factors with the exception of a few quadratic factors. At first glance, the introduction of the quadratic factors would significantly decrease our smoothness probability. However, this is largely compensated by some of the linear factors in f and g being repeated and thus fewer smoothness checks from the linear factors are needed. Additionally, one can find such polynomial pairs with a smaller C compared to those that completely split over the integers. As a result, the smoothness probability that arise from these polynomial pairs are either better or comparable to that of the prior polynomial pairs when fixing its degree.

For example, the following pair of degree 8 polynomials was found in [14] – they differ by an integer and split completely over the integers:

$$\begin{aligned} f(x) &= x(x+4)(x+9)(x+23)(x+27)(x+41)(x+46)(x+50), \text{ and} \\ g(x) &= (x+1)(x+2)(x+11)(x+20)(x+30)(x+39)(x+48)(x+49). \end{aligned}$$

The next pair of degree 8 polynomials are found in this work – again they differ by an integer but $g(x)$ is a square product of linear factors and $f(x)$ factors into linear factors except for one single quadratic factor:

$$\begin{aligned} f(x) &= (x+1)(x+4)(x+9)(x+10)(x+15)(x+18)(x^2+19x-12), \text{ and} \\ g(x) &= x^2(x+6)^2(x+13)^2(x+19)^2. \end{aligned}$$

The probability of finding 384-bit primes p such that $p^2 - 1$ is 2^{23}-smooth is approximately $2^{-48.7}$ from the first polynomial pair and $2^{-40.2}$ from the second pair. The latter probability is significantly larger than the former (see Section 3 on how these smoothness probabilities are computed).

We searched for these polynomial pairs with the aid of the extended Euclidean (XGCD) algorithm over rational polynomial rings. A naïve search computes this XGCD over $\mathbb{Q}[x]$. However, a more fruitful computes this XGCD over $\mathbb{Q}(a_1, \cdots, a_n)[x]$ where a_i's lie in some function field. This can be viewed as a precomputation step and then, after solving some equations in the variables $a_1, \cdots, a_n$, one obtains a more fine-grained searching criterion. This makes the search faster. In Section 5, we apply this strategy to find polynomial pairs of degrees 8, 10 and 12.

We use these polynomials to find b-bit twin smooth integers and primes p such that $p^2 - 1$ is smooth for $b \in \{256, 384, 512\}$. Table 1 summaries the best results. It shows that our polynomials result in a comparable smoothness bound when searching for 256-bit instances, and give significanty better smoothness bounds when searching for larger instances (with $b = 384, 512$).

Organisation We begin in Section 2 by reviewing known techniques for finding such twins – making a clear distinction between the constructive and probabilistic methods. In Section 3 we describe some existing results on smoothness probabilities. In Section 4 we describe the general framework of our method for finding twin smooth integers. In Section 5 we detail the concrete computations with the XGCD algorithm that allow us to find polynomials with a better or comparable smoothness probability. Finally, in Section 6 we search for twin smooth integers and primes p using the polynomials that give the best smoothness probabilities.

2 Existing Techniques for Finding twin smooth Integers

We start by reviewing known techniques to find twin smooth integers. As mentioned in the introduction, we shall separate the techniques into either constructive or probabilistic methods.

2.1 Constructive Methods

The methods presented here is to fix an integer B and attempt to find all or almost all B-smooth twins. It turns out that the set of B-smooth twins is finite for a fixed B. Thus it makes sense try and enumerate all or almost all B-smooth twins.

Solving Pell equations. Let $P_B \coloneqq \{2, 3, \ldots, q\}$ be the set of primes up to B with cardinality $\pi(B)$. Suppose that $(r, r+1)$ is a B-smooth twin and let $x = 2r+1$ so that, as mentioned in the introduction, $x-1$ and $x+1$ are B-smooth. Decompose their product x^2-1 into its squarefree part, D, and its square part, y. Thus the pair (x, y) is a solution to the Pell equation $X^2 - DY^2 = 1$. Additionally, Dy^2 is B-smooth, which means $D = 2^{\alpha_2} \cdot 3^{\alpha_3} \cdot \cdots \cdot q^{\alpha_q}$ with $\alpha_i \in \{0, 1\}$ for each $i \in P_B$. For each of the $2^{\pi(B)}$ squarefree choices for D, Størmer [30] (and later improved algorithmically by Lehmer [22]) reverses the above argument and proposes to solve the $2^{\pi(B)}$ Pell equations

$$X^2 - DY^2 = 1,$$

to find solutions (x, y) such that y is B-smooth. Størmer showed that this set of solutions is finite and thus finds the *complete* set of B-smooth twins $(r, r+1)$.

Solving all $2^{\pi(B)}$ Pell equations becomes computationally infeasible when B is large but is practical when B is small. For instance, with $B = 5$ ($B = 7$ resp.) solving all $2^{\pi(B)}$ Pell equations gives 10 (23 resp.) B-smooth twins – the largest of which is the pair $(80, 81)$ ($(4374, 4375)$ resp.). Lehmer ran his algorithm [22] with $B = 40$ – the complete number of 40-smooth twins is 653. To date, the largest run of this algorithm was done by Costello [13] with $B = 113$ – the complete number of 113-smooth twins is 33,233.

Remark 1. Recent work [8] modifies this approach of using solutions to Pell equations to find smooth twins. Instead of collating all B-smooth twins for a smoothness bound B, they find B-smooth twins in an extremely large interval.

The Conrey-Holmstrom-McLaughlin algorithm. Start with an initial set of integers $S^{(0)} = \{1, 2, \cdots, B-1\}$ that represent the B-smooth twins $(1, 2)$, $(2, 3), \cdots, (B-1, B)$. The algorithm by Conrey-Holmstrom-McLaughlin (CHM) proposes to iteratively add to this initial set with new integers that represent B-smooth twins. For each $r, s \in S^{(i)}$ with $r < s$ compute the following expression

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

where t/t' is written in lowest order terms. Thus one forms a new set of integers $S^{(i+1)}$ to be the set $S^{(i)}$ coupled with the set of integer solutions t where $t' = t+1$ and are not in $S^{(i)}$. Since the set of B-smooth twins is finite, we must have $S^{(d+1)} = S^{(d)}$ for some integer d. At this point the algorithm terminates.

In practice, this algorithm does a good job of finding either all or a majority of B-smooth twins[3]. For instance, with $B = 5$ the algorithm finds all 10 pairs of 5-smooth twins while with $B = 7$ the algorithm finds 22 pairs of 7-smooth twins which is all of them except for the largest twin $(4374, 4375)$. The original authors of the algorithm [12] ran it with $B = 200$ to obtain a total of 346,192 pairs of 200-smooth twins. As a smoothness bound, this is larger than the computations with the Pell equation. More recently, Bruno et al. [7] made improvements to the algorithm and ran it with $B = 547$ to obtain a total of 82,026,426 pairs of 547-smooth twins. Additionally, an extra 2,649 pairs of 200-smooth twins were found from this computation – proving the point that in general the algorithm does not find all smooth twins. The only way to know the exact number of 200-smooth twins is to solve 2^{46} Pell equations which is computationally infeasible given the current computing resources.

[3] The proportion of B-smooth twins that the CHM algorithm finds over all possible B-smooth twins is conjectured to be $1 - o(1)$.

2.2 Probabilistic Methods

Instead of fixing the smoothness bound, the methods presented here fix a target size one attempts to find twin smooth integers of that size. One can guarantee to find twin smooth integers from these methods up some probability. Thus after attempting these methods with sufficiently many inputs one can expect to get such a twin.

The extended Euclidean algorithm over the integers. The most natural approach to find twin smooth integers is to choose random B-smooth integers r until either $r-1$ or $r+1$ is B-smooth. A slightly more fruitful approach [13,19] is to choose two B-smooth integers α and β that are coprime and also $\alpha \cdot \beta$ is roughly the target size of r and $r+1$. Then one uses the extended Euclidean algorithm over the integers and feeds α and β as inputs to the algorithm. The result of this algorithm gives two integers, s and t, such that $\alpha s + \beta t = 1$ with $|s| < |\beta/2|$ and $|t| < |\alpha/2|$. Repeating this with many choices of α and β until the integers s and t are B-smooth one obtains the following B-smooth twins

$$(r, r+1) = (|\alpha s|, |\beta t|).$$

Concretely, the probability r and $r+1$ being B-smooth is now the probability that $s \cdot t$ is B-smooth which is large than the probability that a random integer of similar size being B-smooth.

Searching using cyclotomic factors. In Costello's computations with the Pell equations [13], he noticed that a lot of the largest B-smooth twins for a fixed smoothness bound B were of the form (x^2-1, x^2). In order to find larger twins, he generalised this to consider finding twins of the form

$$(r, r+1) = (x^n - 1, x^n),$$

for small $n \in \mathbb{Z}$, and exploit the fact that $x^n - 1$ factors into its cyclotomic factors. For instance, finding b-bit twins of the form (x^6-1, x^6) requires searching for three $(b/6)$-bit smooth numbers (i.e. $x-1, x$ and $x+1$) and two $(b/3)$-bit smooth numbers (i.e. $x^2 - x + 1$ and $x^2 + x + 1$). This increases the probability of finding such smooth twins.

Searching with PTE solutions. The large degree factors that arise from searching with $r = x^n - 1$ is a bottleneck when making the smoothness bounds of such twins as small as possible. In more recent work, the approach taken in [14] is to find polynomials $f, g \in \mathbb{Z}[x]$ with $g - f \equiv C$ for some integer C and split completely over the integers. Namely, there exists integers $a_1, a_2, \cdots, a_n$, $b_1, b_2, \cdots, b_n$ such that

$$\begin{aligned} f(x) &= (x+a_1) \cdot (x+a_2) \cdots (x+a_n), \text{ and} \\ g(x) &= (x+b_1) \cdot (x+b_2) \cdots (x+b_n). \end{aligned}$$

Once such polynomials are known, when searching for b-bit twins, one sieves through an interval of roughly (b/n)-bit integers and identifies the integers, ℓ, in this interval such that $(\ell + a_i)$ and $(\ell + b_j)$ are all smooth. Thus the evaluations of f and g at the integer ℓ are smooth. A final check needs to be done to determine whether one gets twin smooth integers – namely whether the evaluations $f(\ell)$ and $g(\ell)$ are divisible C. If all of this holds, then the pair

$$(f(\ell)/C, g(\ell)/C)$$

generate twin smooth integers. The completely linear factors in these polynomials increases the smoothness probability again. However, such polynomial pairs f, g is quite non-trivial to find when the degree $n \geq 4$. Fortunately, one can construct such polynomials using solutions to a Diophantine equation referred to as the Prouhet-Tarry-Escott (PTE) problem (see [9] for more background on this problem).

3 Smoothness Probabilities

We recall some known facts about the distribution of smooth numbers in both intervals and evaluated integer-valued polynomials. As a result, one can result about calculating the probability of identifying smoothness in these settings, something which will be commonly referred to as *smoothness probability*. The exposition given here follows the direction from [14, §2].

3.1 Dickman-rho Function and Distribution of Smooth Integers

One can attempt to count the number of B-smooth integers up to some bound N. This is often given the notation

$$\Psi(N, B) \coloneqq \#\{1 \leq m \leq N : m \text{ is } B\text{-smooth}\}.$$

In order to do this counting, we define the Dickman-de Bruijn (rho) function. It is a function $\rho : \mathbb{R}^+ \to \mathbb{R}^+$ that is continuous at $u = 1$, differentiable for $u > 1$ and satisfies the following difference differentiable equation

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

When $1 \leq u \leq 2$, one can explicitly write $\rho(u) = 1 - \ln(u)$. But, for $u \geq 2$, there is no known closed form for this function in terms of elementary functions. However, for such inputs u, one can still evaluate this function using numerical techniques [33,24] which are built in to many popular computer algebra packages (including Magma and SageMath).

Relating this to the context of counting smooth integers, Dickman [17] and independently by de Bruijn [16] proved that as $N \to \infty$ we have

$$\Psi(N, B) \sim \rho(u)N,$$

where $u = \log(N)/\log(B)$. Hence the proportion of B-smooth integers approaches this Dickman-de Bruijn function. While this formula is asymptotic, it yields a good approximation for concrete values of N and B. As a result, assuming that the set of B-smooth integers in the interval $[1, N]$ are uniformly distributed with $B = N^{1/u}$, the Dickman-de Bruijn function can be used to approximate the probability that an integer less than N is B-smooth.

3.2 Smoothness of Polynomial Evaluations

For a polynomial $f \in \mathbb{Z}[x]$, define

$$\Psi_f(N, B) \coloneqq \#\{1 \leq m \leq N : f(m) \text{ is } B\text{-smooth}\}$$

which counts the number of smooth evaluations of f. There has been numerous works that have looked into the quantity $\Psi_f(N, B)$ and heuristically one can argue that the smoothness probability of $f(m)$ is equal to the product of the smoothness probabilities of the evaluation of each irreducible factor of f at m. While this heuristic is proven for certain ranges of N and B [25], these ranges do not apply for the specific ranges that are of cryptographic interest. Having said this, backed up with not only our experiments but also the experimentation done in [14], these heuristics are a close approximation to their true values especially when searching for twin smooth intgers of cryptographic-size. As a result we formally restate the heuristic given in [14].

Heuristic 1 *Suppose that a polynomial $f \in \mathbb{Z}[x]$ has distinct irreducible factors over $\mathbb{Z}[x]$ of degrees $d_1, \cdots, d_k \geq 1$. Then, as $X \to \infty$, we have*

$$\Psi_f(X, B) \sim \rho(d_1 u) \cdots \rho(d_k u)X,$$

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

Smoothness of Rational Polynomial Evaluations. As mentioned already with the PTE sieve but will also crop up in later sections, when searching for twin smooth integers one typically is tasked to find smooth values of a rational-valued polynomial $f \in \mathbb{Q}[x]$ rather than an integer-valued polynomial. With some small modifications one can still use the heuristic in order to compute these smoothness probabilities. Write

$$f(x) = \frac{1}{C}\hat{f}(x),$$

where $C \in \mathbb{Z}$ is an integer and $\hat{f} \in \mathbb{Z}[x]$ is an integer-valued polynomial such that C is coprime to the greatest common divisor of the coefficients of $\hat{f}$. One applies Heuristic 1 to compute the probability that the evaluation $\hat{f}(m)$ is smooth. On top of this, one also needs to calculate the probablity that $\hat{f}(m)/C$ is an integer. This depends on a congruence condition modulo C, namely the number of integers $m \in \mathbb{Z}/C\mathbb{Z}$ such that $\hat{f}(m) = 0 \mod C$. By the Chinese remainder theorem, this depends of a system of congruences

$$\hat{f}(m) = 0 \mod p_i^{e_i}, \quad 1 \leq i \leq k,$$

where $C = \prod p_i^{e_i}$ for distinct primes p_i. As long as each prime power $p_i^{e_i}$ is not too big, then one can directly compute the number of integers modulo $p_i^{e_i}$ such that their evaluation at $\hat{f}$ is 0. Then multiplying all of these numbers together gives the number of integers modulo C such that their evaluation at $\hat{f}$ is 0 and dividing this number by C gives the associated probability.

The result of this is one can calculate the probability the that rational polynomial f is not only an integer but is also smooth. At least experimentally, the condition that $\hat{f}(m)$ is smooth and $\hat{f}(m) = 0 \mod C$ appear to be mutually independent. We assuming that these are actually mutually independent events. So the probability that $f(m)$ is smooth can be heuristically compute as the product of the probability that $\hat{f}(m)$ is smooth and the probability that $\hat{f}(m)/C$ is an integer. We note that this point was addressed in the work using the PTE sieve [14], however they did not take this is additional step into account when computing the smoothness probabilities for finding twin smooth integers.

4 Smooth Twins using XGCD over Polynomial Rings

This section aims to describe a novel probabilistic method for finding twin smooth integers. As it will be shown, one can view this method as a generatlisation of the probabilistic techniques described in §2.2. Before describing the algorithm we review the extended Euclidean algorithm over polynomials rings $\Bbbk[x]$ for a field $\Bbbk$. This will serve as a core ingredient to the method.

4.1 Extended Euclidean Algorithm over Polynomial Rings

Suppose $\Bbbk$ is a field and work with the polynomial ring $\Bbbk[x]$. The goal of the extended Euclidean algorithm (XGCD) is, upon the input of two polynomials $F, G \in \Bbbk[x]$, find two polynomials S, T such that we have the following polynomial Bezout identity

$$F \cdot S + G \cdot T \equiv \gcd(F, G),$$

where $\gcd(F, G) \in \Bbbk[x]$ is the greatest common divsor[4] of the polynomials F and G. The classical procedure is to recursively do division with remainder of two polynomials that initially start with F and G, and use this information to compute the gcd along with S and T. A more formal description of the algorithm is given in [29, §17.3]. Moreover, the polynomials S and T are determined uniquely under the condition that $\deg(S) < \deg(G)$ and $\deg(T) < \deg(F)$ [29, Theorem 17.4].

The following lemma is a statement that will be frequently used in later sections and thus we state it here in its full generality.

[4] Meaning the monic polynomial $H \in \Bbbk[x]$ such that $H \mid F$ and $H \mid G$; and for any monic polynomial H' with $H' \mid F$ and $H' \mid G$ we have $H' \mid H$.

Lemma 1. *Let $F, G \in \Bbbk[x]$ be coprime polynomials and let $S, T \in \Bbbk[x]$ be the result of applying the XGCD algorithm to F and G with $\deg(S) < \deg(G)$ and $\deg(T) < \deg(F)$. For a positive integer n, set $F_n(x) \coloneqq F(x^n)$ and $G_n(x) \coloneqq G(x^n)$. Let S_n, T_n be the result of applying the XGCD algorithm to F_n and G_n with $\deg(S_n) < \deg(G_n)$ and $\deg(T_n) < \deg(F_n)$. Then we have*

$$S_n(x) = S(x^n), \qquad \text{and} \qquad T_n(x) = T(x^n).$$

Proof. By the coprimality of F and G, we have the following relation

$$F(x)S(x) + G(x)T(x) = 1.$$

Replacing x by x^n in this equation we get

$$F_n(x)S(x^n) + G_n(x)T(x^n) = 1.$$

Writing $S'(x) = S(x^n)$ and $T'(x) = T(x^n)$, we have $\deg(S') = n\deg(S) < n\deg(G) = \deg(G_n)$ and $\deg(T') < \deg(F_n)$. Since S_n and T_n are determined uniquely, we deduce that $S_n \equiv S'$ and $T_n \equiv T'$. □

While the polynomials S and T are uniquely determined when $\deg(S) < \deg(G)$ and $\deg(T) < \deg(F)$, the opposite is true when $\deg(S) \geq \deg(G)$ and $\deg(T) \geq \deg(F)$. One can construct a family of polynomials S' and T' using S and T such that

$$F \cdot S' + G \cdot T' \equiv \gcd(F, G). \tag{1}$$

This family can be given as $S'(x) \coloneqq S(x) + P(x)G(x)$ and $T'(x) \coloneqq T(x) - P(x)F(x)$ for any $P \in \Bbbk[x]$. This family completely parameterises Equation (1) in the sense that any polynomials S', T' that satisfy this relation has to be of this form. We give the name of *perturbing the solution* for this procedure.

4.2 The General Strategy to find Smooth Twins

Choose two coprime polynomials, $F, G \in \mathbb{Z}[x]$, that split completely into linear factor and the number of distinct roots of $F \cdot G$ is not too big. We will assume for expositions sake that F and G are monic polynomials so that each irreducible divisor of F and G are of the form $(x + a)$ for an integer a. Detail we make some comments as to how to proceed with the case when F and G are not monic. Use the extended Euclidean (XGCD) algorithm over the polynomial ring $\mathbb{Q}[x]$ to find the corresponding polynomials $S, T \in \mathbb{Q}[x]$ such that $F \cdot S + G \cdot T \equiv 1$. Assume without loss of generality that S and T do not have any linear factors and the leading coefficient of $F \cdot S$ is positive. Then the polynomials $F \cdot S$ and $-G \cdot T$ differ by one and thus one can search for twin smooth integers be finding well chosen inputs, m, to these polynomials. In the vast majority of circumstances, these polynomials have rational coefficients. So one has to adopt a PTE style search where one lifts these polynomials to $f(x) \coloneqq C \cdot F(x) \cdot S$ and $g \coloneqq -C \cdot G(x) \cdot T(x)$, where C is the lowest common multiple of the denominators of the coefficients S and T. Instead of differing by 1, these polynomials now differ by the integer v. Then one searches for integers ℓ such that $f(\ell)$ and $g(\ell)$ are smooth and $f(\ell) \equiv g(\ell) \equiv 0 \mod v$. This latter condition ensures that $f(\ell)$ and $g(\ell)$ are divisible by v and result in twin smooth integers. The algorithmic procedure is given below and implicitly take F, G and a smoothness bound B as input:

1. Use the extended Euclidean algorithm to find polynomials S and T such that $F \cdot S + G \cdot T \equiv 1$. Let C is the lowest common multiple of the denominators of the coefficients S and T and set $f(x) \coloneqq CF(x)S(x)$, $g(x) \coloneqq -CG(x)T(x)$. So $f, g \in \mathbb{Z}[x]$ with $|g - f| \equiv C$.
2. Sieve a large interval of consecutive integers, I, identifying which integers $\ell \in I$ are B-smooth and produce a bit string where a '1' indicates a B-smooth integer. This sieving procedure can be done using the sieve of Eratosthenes as described in the work on the PTE sieve [14, §4.1] (see also [15, §3.2.5]).
3. Iteratively go through each entry in the bit string and identify the integers $\ell \in I$ such that $\ell + a$ are B-smooth for each integer a with $(x + a) \mid F(x) \cdot G(x)$.

4. Isolate those integers, ℓ, for which $f(\ell) \equiv g(\ell) \equiv 0 \mod C$.
5. Determine the integers ℓ such that $CS(x)$ and $CT(x)$ are B-smooth using one of the following techniques:
   (a) Evaluate each irreducible factor of $v^2S(x)T(x)$ at the integer ℓ and use either trial division or fast factoring methods such as ECM [26] to determine whether it is B-smooth.
   (b) Use an Eratosthenes style sieve as but instead of sieving an interval, one sieves a list of evaluations of an irreducible polynomial [15, §3.2.7], say $\{P(\ell) : \ell \in I\}$ for $P \in \mathbb{Z}[x]$. Once again, this outputs a bitstring. Do this for each irreducible factor of $v^2S(x)T(x)$. We note that this can be done in paralell to that of the sieve in Step 2.
   (c) Collate all integers of the form $P(\ell)$ for each irreducible factor P of $v^2S(x)T(x)$ into a list and apply Bernstein's sieving algorithm [3] to identify the B-smooth instances. For an integer m, the computation
   $$m_B \coloneqq \left(\prod_{p \leq B} p \mod m \right)^{2^e} \mod m,$$
   where $e > 0$ is the smallest integer such that $2^{2^e} \geq m$, gives the largest B-smooth divisor of m [3, Theorem 2.2]. Moreover, if $m_B = m$, then m is B-smooth. One does this computation for each integer in the list with aid of a remainder tree (see [15, §3.3]).
6. The remaining integers ℓ yield twin smooth integers $(f(\ell)/C, g(\ell)/C)$.

Deciding which technique in Step 5 to use in the search for twin smooth integers depends on the specific choice of polynomials F, G. This will be discussed in Section 6.

Remark 2. One can replace S and T with these perturbed solutions $S' = S + P \cdot G$ and $T' = T - P \cdot F$ that satisfy Equation (1) and attempt to find twin smooth integers. This results in the polynomials f and g having larger degree and hence would most likely decrease the smoothness probability. This is also apparent in the XGCD approach over the integers as mentioned in §2.2 but the difference here is that we have flexibility on the smooth input ℓ that we choose. However this discrepancy is necessary for some choices of polynomials F, G if the intention is some cryptographic purpose. We refer the reader of this text to Appendix A for some details on this discrepancy.

Precomputed Polynomials. There are two natural choices for choosing the input polynomials F and G. The first way is to simply choose them at random. In other words, construct polynomials $F(x) = \prod(x+a)^{e_a}$ and $G(x) = \prod(x+b)^{f_b}$ by choosing integers $a, b \in \mathbb{Z}$ at random as long as $a \neq b$ to ensure coprimality. A better approach is to have a precomputed list of polynomials F, G and choose one or many of them for the twin smooth searches. The advantage now is that one can choose the input polynomials F and G that result in the best smoothness probabilities. Additionally, the polynomials S and T can be precomputed which saves the computation done in Step 1. So instead of choosing those polynomials F, G at random, one enumerates over a large set of polynomials of the form given above until some are found that give good smoothness probabilities. We note that this precomputation approach was also adopted by Costello et. al. [14] in their PTE sieve – namely they precompute and collect many PTE solutions using either known parameterisations or using interpolation techniques [5].

This naïve precomputation approach requires doing XGCD computations over $\mathbb{Q}[x]$. A slightly better approach is to initially do XGCD algorithm over a polynomial ring with coefficients lie in some rational function field. Effectively the integers a, b that give the polynomials F and G become parametrised as variables over the rational function field. Write $\Bbbk = \mathbb{Q}(a_1, \cdots, a_n)$ for this rational function field and $\Bbbk[x]$ for its polynomial ring. Once the XGCD computation over $\Bbbk[x]$ is done, one gets polynomials $S, T \in \Bbbk[x]$. Then one translates each irreducible factor of $S \cdot T$ back to $\mathbb{Q}[x]$ by evaluating each variable in the function field. Finally, one computes the factorisations of these resulting polynomials and record their factorisation structure. This significantly reduces the amount of XGCD computations which is the main bottleneck in the naïve approach. This idea will be exploited in Section 5 to search for certain degree 8, 10 and 12 polynomials.

Seaching with one vs many polynomial pairs. Much like with the PTE sieve, one can search for twin smooth integers using either a single polynomial pair or using many polynomial pairs. The search with a single polynomial pair is exactly as described above but one might prefer to swap the order of Step 3 and Step 4 and only check that the integers $\ell + a$ are all B-smooth for the ℓ's which satisfy $f(\ell) \equiv g(\ell) \equiv 0 \mod C$. When this integer C is large, this proportion of integers that satisfy this condition may be quite small so it would be beneficial to only search for those ℓ's. As mentioned in §3.2, computing all of these residues modulo C can be done effectively through a Chinese remainder computation. Then computing all these suitable integers ℓ can easily be done through this list of residues.

Alternatively, one can store a list of polynomial pairs and search for twin smooth integers using all of these pairs simultaneously. To make this effective one employs a tree that requires a minimal number of checks to fully traverse all polynomial pairs. The construction of the tree can be seen as an instance of the *hitting set problem* as described in detail in [14, §4.3]. In particular, when lots of these pairs share a root, then this approach is particularly effective in the overall this. In the context of the general algorithm, this construction of the tree as well as the subsequent sieving is incorporated into Step 3.

Realising the generalisation. One can view this method as a generalisation of the probabilistic methods described in §2.2. To obtain the polynomial pair $x^n - 1, x^n$ using this approach, one simply computes the XGCD of $F(x) = x^n$ and $G(x) = x - 1$. The polynomial pair that result from a PTE solution,

$$\begin{aligned} f(x) &= (x + a_1) \cdot (x + a_2) \cdots (x + a_n), \text{ and} \\ g(x) &= (x + b_1) \cdot (x + b_2) \cdots (x + b_n), \end{aligned}$$

can be recovered as follows. Iterate over all polynomials of the form $F(x) = (x + A_1) \cdots (x + A_{n_1})$ and $G(x) = (x+B_1) \cdots (x+B_{n_2})$ with $n_1+n_2 > n$ and apply the XGCD algorithm until the resulting polynomials S and T completely split. This idea is exploited in Appendix B to find new solutions that have not been seen in the literature.

Smoothness probabilities. Recall from §3.2 that says the smoothness probability of an evaluated polynomial depends on the irreducible factors of the polynomial. Thus the probability of obtaining twin smooth integers from this method depends on the irreducible factors of $F \cdot G \cdot S \cdot T$. Given this, one might suggest that the smoothness probabilities would be optimised when the degrees of each of the factors are all 1 resulting in polynomial the split completely. This would certainly be the case when there are repeated factors in either polynomials since there are fewer smoothness checks in Step 3 of the algorithm. However, such polynomial pairs only exist when the degree of the polynomial is $n \in \{2, 3, 4, 6\}$. For the larger degree pairs this suggests the following: instead of having all of the factors being linear, one replaces some of the linear factors with quadratic (or potentially higher degree) factors and counterbalance that by having more square (or potentially larger power) factors. One example of such a polynomial pair is the following degree 8 pair that differ by an integer and factors into linear factors up to one quadratic factor:

$$\begin{aligned} f(x) &= (x + 1)(x + 4)(x + 9)(x + 10)(x + 15)(x + 18)(x^2 + 19x - 12), \text{ and} \\ g(x) &= x^2(x + 6)^2(x + 13)^2(x + 19)^2. \end{aligned} \tag{2}$$

As mentioned in Section 1, the smoothness probability from this pair is much larger than that of a known degree 8 pair that split completely into linear factors. In addition, here is another polynomial pair but of degree 12:

$$\begin{aligned} f(x) &= (x + 4)(x + 7)(x + 22)(x + 50)(x + 56)(x + 84)(x + 99)(x + 102) \\ &\qquad (x^2 + 75x - 136)(x^2 + 137x + 3150), \text{ and} \\ g(x) &= x^2(x + 14)^2(x + 39)^2(x + 67)^2(x + 92)^2(x + 106)^2. \end{aligned} \tag{3}$$

A more in depth discussion about how to search for such polynomials will be given in Section 5 detailing how to optimise this search.

Remark 3. In the setting of searching for twins whose their sum is a prime, an extra probability is included which, by the prime number theorem, is approximately $1/(\log(2)b)$. We heuristically assume that This probability can be computed independently from the other smoothness probabilities.

5 Searching for Polynomials with Better Smoothness Probabilities

In this section, we attempt to find polynomials using the extended euclidean algorithm such that the corresponding smoothness probability is larger than the corresponding ideal PTE solution of the same size. As mentioned previously, this will only be possible when the degree of the resulting polynomial that differ by a constant is not $2, 3, 4$ or 6. This is because for these degrees there are known ideal PTE solutions that feature some repetition in the sets. From the perspective of smoothness probabilities, this is the optimal and one cannot expect to find other polynomials of these degrees that yield a larger probability. The solutions found in the previous section are one such example of optimal solutions. For other degrees, the only known ideal PTE solutions have distinct elements. Additionally, other than the size $2, 3, 4$ and 6 mentioned previously, such solutions are only known for sizes $5, 7, 8, 9, 10$ and 12. The fact that such solutions only feature distinct elements is a bottleneck from the perspective of the smoothness probability.

The aim here is to find polynomials, f, g, that differ by a constant (much like those that arise from a ideal PTE solution), such that the number of linear factors of $f \cdot g$ is a lot smaller than a corresponding ideal PTE solution of the same size but is counterbalanced with the inclusion of one or more quadratic factors. As long as the number of such quadratic factors is not too large then the smoothness probability can be either similar to a corresponding ideal PTE solution or maybe larger. In order to reduce the number of linear factors, the general strategy is to find we attempt to find polynomials of the form $f(x) = h(x)^k - C$ and $g(x) = h(x)^k$ where $k > 1$ and C are integers and h is a completely split polynomial over the rationals such that the factorisation of $f(x)$ decomposes into lots of linear factors and only a few larger degree factors. Moreover, if f has a root a, then $C = h(a)^k$ is a k^{th} power and the polynomial f factorises into its k^{th} cyclotomic factors composed with h. In addition we reduce the search space of such polynomials by only searching for symmetric polynomials. In the setting of even degree polynomials, which will be the primary focus of this section, is done by ensuring that the resulting polynomials are of the form $f(x) = \hat{f}(x^2)$ for some polynomial $\hat{f}$. This trick has also been used to find symmetric PTE solutions that use interpolation techniques [5]. Towards the end of this section we highlight some differences when searching for odd degree instances.

In order to do achieve this in practice, the general idea is work over the rational function field $\mathbb{Q}(a_1, \cdots, a_m, a)$ and to apply the XGCD algorithm over $\mathbb{Q}(a_1, \cdots, a_m, a)[x]$ to the polynomials $F(x) \coloneqq x^2 - a^2$ and $G(x) \coloneqq h(x)^k$ for $k \in \mathbb{Z}$ with $k > 1$ and $h \in \mathbb{Q}(a_1, \cdots, a_m)[x]$. This polynomial h should be chosen such that it splits completely into linear factors and $h(x)^k$ is an even function – in order words either

$$h(x) = \prod_{i=1}^{m}(x^2 - a_i^2) \quad \text{or} \quad h(x) = x\prod_{i=1}^{m}(x^2 - a_i^2).$$

This is so that we can apply Lemma 1 and deduce that the result of this XGCD computation gives two polynomials S and T whose degrees are $\deg(S) = \deg(h) \cdot k - 2$ and $\deg(T) = 0$. Since $\deg(T) = 0$, we let $C \in \mathbb{Q}(a_1, \cdots, a_m, a)$ be the element of the rational function field such that $T(x) = 1/C$. Hence the polynomials f, g can be recovered as follows: $g(x) \coloneqq C \cdot (G(x) \cdot T(x)) = h(x)^k$ and $f(x) \coloneqq C \cdot (-F(x) \cdot S(x)) = C \cdot (G(x) \cdot T(x) - 1) = h(x)^k - C$.

With this precomputation complete, we now enumerate over all possible integers $a_1, \cdots, a_m, a$ and evaluate the coefficients of S at these integers to give a polynomial in $\mathbb{Q}[x]$. Factorise this polynomial and record the polynomials f, g if it has lots of linear factors and some quadratic factors. Finally, for sake of cleaning the polynomials, one shifts the polynomials f, g by a linear shift, $x \mapsto x + A$, so that the polynomials have linear factors of the form $(x + \alpha)$ for $\alpha \geq 0$ and $x \mid f(x)g(x)$. In some circumstances, we can consider half integers $a_1, \cdots, a_m, a$. This is the case when $\deg(h)$ is even and all $a_1, \cdots a_m, a$ are all odd since applying the appropriate shift gives α's which are all even. If a mixture of odd and even integers encompass $a_1, \cdots a_m, a$,

then one can not do this trick since the resulting polynomials after applying the shift will not have integer coefficients.

For the sake of illustration we go through a few examples to see this in action. These polynomial pairs in a better smoothness probability compared to those that arises from a corresponding and existing PTE solutions.

Example 1. Working with the rational function field $\mathbb{Q}(a_1, a_2, a)$, let $n = 2$ and $h(x) = (x^2 - a_1^2)(x^2 - a_2^2)$ which splits completely over $\mathbb{Q}(a_1, a_2, a)$. By choice we have

$$F(x) = x^2 - a^2, \text{ and } G(x) = \left(\left(x^2 - a_1^2\right)\left(x^2 - a_2^2\right)\right)^2.$$

Applying the XGCD algorithm to these polynomials gives

$$S(x) = -\frac{1}{C}\left(x^2 - (a_1^2 + a_2^2 - a^2)\right)\left(x^4 - (a_1^2 + a_2^2)x^2 + a_1^2a_2^2 + (a_1^2 - a^2)(a_2^2 - a^2)\right), \text{ and } T(x) = \frac{1}{C},$$

where $C = ((a_1^2 - a^2)(a_2^2 - a^2))^2$. The quadratic and quartic factors in $S(x)$ are irreducible as polynomials in $\mathbb{Q}(a_1, a_2, a)[x]$. But for certain rational choices for a_1, a_2 and a, these factors will be reducible. For instance, when $a_1 = 19/2$, $a_2 = 7/2$ and $a = 1/2$, then the polynomial S factors into four linear factors and one irreducible quadratic. One can compute $f(x)$ and $g(x)$ from the formulas mentioned above and, after doing a linear shift $x \mapsto x + 19/2$ to ensure that all linear factors of f and g are of the form $(x + a)$ with $a \geq 0$, the resulting f, g are

$$\begin{aligned} f(x) &= (x+1)(x+4)(x+9)(x+10)(x+15)(x+18)(x^2+19x-12), \text{ and} \\ g(x) &= x^2(x+6)^2(x+13)^2(x+19)^2. \end{aligned}$$

These are exactly the polynomials mentioned in Equation (2).

Example 2. Working with the rational function field $\mathbb{Q}(a_1, a_2, a_3, a)$, let $n = 2$ and $h(x) = (x^2 - a_1^2)(x^2 - a_2^2)(x^2 - a_3^2)$. So we have

$$F(x) = x^2 - a^2, \text{ and } G(x) = \left((x^2 - a_1^2)(x^2 - a_2^2)(x^2 - a_3^2)\right)^2.$$

Applying the XGCD algorithm to these polynomials gives

$$S(x) = -\frac{1}{C}p_4(x)p_6(x), \text{ and } T(x) = \frac{1}{C},$$

where $C = ((a_1^2 - a^2)(a_2^2 - a^2)(a_3^2 - a^2))^2$ and $p_4, p_6 \in \mathbb{Q}(a_1, a_2, a_3, a)[x]$ are algebraically computatable irreducible polynomials of degree 4 and 6 (resp.). Once again, for select choices of rationals a_1, a_2, a_3 and a, these factors can be reduced. For instance, when $a_1 = 53$, $a_2 = 39$, $a_3 = 14$ and $a = 3$, the polynomial S factors into six linear and two irreducible quadratic factors. After computing $f(x)$ and $g(x)$ from the formulas mentioned above and applying the linear shift $x \mapsto x + 53$, the resulting polynomials are

$$\begin{aligned} f(x) &= (x+4)(x+7)(x+22)(x+50)(x+56)(x+84)(x+99)(x+102) \\ &\qquad (x^2+75x-136)(x^2+137x+3150), \text{ and} \\ g(x) &= x^2(x+14)^2(x+39)^2(x+67)^2(x+92)^2(x+106)^2, \end{aligned}$$

which were mentioned in Equation (3).

A minor bottleneck in this approach is that as $\deg(h)$ increases finding polynomials S that ideally factors to the product of quadratic factors reduces. One can alleviate this bottleneck slightly by considering the following modified approach. Instead of doing the XGCD computation with the polynomials F, G described above, replace it by doing the XGCD computation of $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = h(x)^k$. This time the computation is done over the polynomial ring $\mathbb{Q}(a_1, \cdots, a_m, a, b)[x]$. Now we get polynomials S, T whose degrees are $\deg(S) = \deg(h) \cdot k - 2$ and $\deg(T) = 2$. For certain choices of $a_1, \cdots, a_m, a, b$ we have $\deg(S) = \deg(h) \cdot k - 4$ and $\deg(T) = 0$. This exactly happens when the leading coefficient of T is equal to 0. Note that we do not need to worry about the x coefficient since it is already 0 by Lemma 1. Algebraically computing this leading coefficient and setting it equal to 0 gives us a relation between the variables of the function field: $a_1, \cdots, a_m, a, b$. Given this relation, all that is needed is to be able to isolate one of the terms and write it in terms of the other terms. Once this is done, replace this isolated term in the expression of F and G. With this replacement, we can ensure that the degree of S and T are $\deg(h) \cdot k - 4$ and 0 (resp.). We note that it will be necessary so slightly modify the structure of how $h(x)$ looks so that isolating one of the variables is possible. This time either

$$h(x) = (x^2 - a_1)\prod_{i=2}^{m}(x^2 - a_i^2) \quad \text{or} \quad h(x) = x(x^2 - a_1)\prod_{i=2}^{m}(x^2 - a_i^2),$$

and the variable that will be isolated is a_1. The fact that the degree of S decreases slightly alleviates this factoring bottleneck. Additionally, a larger space of polynomial pairs can be searched over compared to the inital approach since h may not split completely. In some circumstances, this allows us to find polynomial pairs with a small C.

It is natural to ask whether one can further build on this and choose $F(x) = (x^2 - a^2)(x^2 - b^2)(x^2 - c^2)$ (or more products). The challenge with this is being able to solve the respective equation such that one ensures that $\deg(T) = 0$. This becomes quite non-trivial when F has three or more products of this type. When $F(x) = (x^2 - a^2)(x^2 - b^2)$ it becomes quite a bit easier so we focus on this. We describe this procedure to find degree $8, 10$ and 12 polynomial pairs and detail the underlying equation that needs to be solved.

Table 2 collates many polynomial pairs, $(f(x), g(x))$ with $g - f \equiv C$, found from these searches and computes the approximate probability of finding b-bit primes p such that $p^2 - 1$ is B-smooth. As mentioned in §3.2 and Remark 3, we can heuristically compute this probability as

$$\rho(d_1 u) \cdots \rho(d_k u) + \frac{\#\{0 \leq m < C : f(m) \equiv g(m) \equiv 0 \mod C\}}{C} + \frac{1}{\log(2) b},$$

where the d_i's are the degrees of each irreducible factor of $f \cdot g$, $u = \log(X)/\log(B)$ and X is the largest integer that can find in b-bit primes from the polynomials f and g. We note that the size of the integer C has a role in the computation of the smoothness probability and the smaller it is the larger the probability. We also include some probabilities from polynomials found from prior work.

5.1 Degree 8 Polynomials

We begin with the search for degree 8 polynomials. The precomputation step consists of working over the rational function field $\mathbb{Q}(a_1, a_2, a, b)$ and its polynomial ring $\mathbb{Q}(a_1, a_2, a, b)[x]$. This search will prove to be the easiest and, up to the XGCD precomputation over $\mathbb{Q}(a_1, a_2, a, b)[x]$, an implementation of the search strategy can be done without needing any polynomial arithmetic.

First we choose $n = 2$ and write $h(x) = (x^2 - a_1)(x^2 - a_2^2) \in \mathbb{Q}(a_1, a_2, a, b)[x]$. We intend to apply the XGCD algorithm over $\mathbb{Q}(a_1, a_2, a, b)[x]$ to the polynomials $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = h(x)^2$. Using computer algebra softwares such as Magma or Sagemath, one can explicitly compute the resulting polynomials S and T and assertain that $\deg(S) = 6$ and $\deg(T) = 2$. This computationally verifies the more general statements when specifically applied to this setting. We note that the denominator of the leading coefficient of S and T is equal to $(a_1 - a^2)^2(a_1 - b^2)^2(a_2^2 - a^2)^2(a_2^2 - b^2)^2$ and its numerator is (up to sign) equal to

$$(a_1 + a_2^2 - a^2 - b^2) \cdot (2a_1 a_2^2 - a_1 a^2 - a_1 b^2 - a_2^2 a^2 - a_2^2 b^2 + a^4 + b^4).$$

Method	n	$(f(x), g(x))$	$\lceil \log_2(C) \rceil$	(m_1, m_2)	Smoothness Probability		
					$b = 256$	$b = 384$	$b = 512$
Cyclotomic factors [13]	6	$(x^6 - 1, x^6)$	0	$(3, 2)$	$2^{-43.0}$	$2^{-49.8}$	$2^{-53.9}$
	8	$(x^8 - 1, x^8)$	0	$(3, 1)^*$	$2^{-44.9}$	$2^{-51.8}$	$2^{-55.9}$
	10	$(x^{10} - 1, x^{10})$	0	$(3, 0)^*$	$2^{-44.8}$	$2^{-51.4}$	$2^{-55.6}$
	12	$(x^{12} - 1, x^{12})$	0	$(3, 3)^*$	$2^{-31.7}$	$2^{-37.1}$	$2^{-40.3}$
PTE sieve [14]	6	PTE_1^6	14	$(9, 0)$	$2^{-42.8}$	$2^{-48.9}$	$2^{-52.6}$
		PTE_2^6	17	$(12, 0)$	$2^{-55.3}$	$2^{-63.0}$	$2^{-67.7}$
	8	PTE_1^8	31	$(16, 0)$	$2^{-47.4}$	$2^{-52.8}$	$2^{-56.0}$
		PTE_2^8	35	$(16, 0)$	$2^{-50.1}$	$2^{-55.1}$	$2^{-58.1}$
		PTE_3^8	38	$(16, 0)$	$2^{-52.5}$	$2^{-57.2}$	$2^{-60.0}$
	10	PTE^{10}	73	$(20, 0)$	$2^{-57.5}$	$2^{-59.2}$	$2^{-60.3}$
	12	PTE^{12}	76	$(24, 0)$	$2^{-44.7}$	$2^{-45.9}$	$2^{-46.7}$
XGCD over $\mathbb{Q}[x]$	6	XGCD^6	6	$(7, 1)$	$2^{-45.2}$	$2^{-52.5}$	$2^{-56.8}$
	8	XGCD_1^8	10	$(8, 2)$	$2^{-38.6}$	$2^{-44.8}$	$2^{-48.5}$
		XGCD_2^8	12	$(8, 2)$	$2^{-39.6}$	$2^{-45.6}$	$2^{-49.2}$
		XGCD_3^8	14	$(4, 3)$	$2^{-41.6}$	$2^{-47.2}$	$2^{-50.5}$
		XGCD_4^8	15	$(7, 2)$	$2^{-39.7}$	$2^{-45.2}$	$2^{-48.4}$
		XGCD_5^8	16	$(8, 2)$	$2^{-41.4}$	$2^{-47.2}$	$2^{-50.6}$
		XGCD_6^8	21	$(10, 1)$	$2^{-38.3}$	$2^{-43.4}$	$2^{-46.5}$
		XGCD_7^8	30	$(10, 1)$	$2^{-43.6}$	$2^{-48.1}$	$2^{-50.8}$
		XGCD_8^8	32	$(10, 1)$	$2^{-43.8}$	$2^{-48.2}$	$2^{-50.8}$
		XGCD_9^8	35	$(10, 1)$	$2^{-45.3}$	$2^{-49.5}$	$2^{-52.0}$
	10	XGCD_1^{10}	22	$(9, 3)$	$2^{-36.0}$	$2^{-40.8}$	$2^{-43.7}$
		XGCD_2^{10}	26	$(11, 2)$	$2^{-34.1}$	$2^{-38.3}$	$2^{-40.9}$
		XGCD_3^{10}	28	$(9, 3)$	$2^{-38.5}$	$2^{-43.0}$	$2^{-45.7}$
		XGCD_4^{10}	41	$(11, 2)$	$2^{-39.4}$	$2^{-42.9}$	$2^{-45.0}$
		XGCD_5^{10}	43	$(11, 2)$	$2^{-40.3}$	$2^{-43.7}$	$2^{-45.7}$
	12	XGCD_1^{12}	14	$(8, 5)$	$2^{-31.4}$	$2^{-36.3}$	$2^{-39.3}$
		XGCD_2^{12}	24	$(10, 4)$	$2^{-31.0}$	$2^{-35.2}$	$2^{-37.7}$
		XGCD_3^{12}	24	$(10, 4)$	$2^{-31.0}$	$2^{-35.2}$	$2^{-37.7}$
		XGCD_4^{12}	29	$(10, 4)$	$2^{-32.6}$	$2^{-36.5}$	$2^{-38.9}$
		XGCD_5^{12}	42	$(12, 3)$	$2^{-34.3}$	$2^{-37.3}$	$2^{-39.1}$
		XGCD_6^{12}	56	$(12, 3)$	$2^{-38.8}$	$2^{-41.1}$	$2^{-42.6}$
		XGCD_7^{12}	59	$(12, 3)$	$2^{-42.2}$	$2^{-44.4}$	$2^{-45.8}$
		XGCD_8^{12}	60	$(14, 2)$	$2^{-37.6}$	$2^{-39.7}$	$2^{-40.9}$
		XGCD_9^{12}	69	$(12, 3)$	$2^{-44.8}$	$2^{-46.4}$	$2^{-47.4}$

Table 2: A collection of polynomial pairs, $(f(x), g(x))$ with $g - f \equiv C \in \mathbb{Z}$, used as a tool to search for twin smooth integers. The integer $n = \deg(f) = \deg(g)$ and the integers m_1 and m_2 denote the combined number of linear and quadratic (resp.) factors counted without repetition. An astrisk, $(m_1, m_2)^*$, is marked when there are factors of degree greater than 2. The smoothness probability column consists of the approximate probability of finding b-bit primes p, for $b \in \{256, 384, 512\}$, such that $p^2 - 1$ is B-smooth, for $B \in \{2^{16}, 2^{22}, 2^{28}\}$ (resp.). The probabilities shaded in gray means that the search space is much too small to expect to find twin smooth integers. The description of the polynomial pairs can be found in the Appendix C.

Occasionally, this leading coefficient could be 0 and $\deg(S) = 4$ and $\deg(T) = 0$. As mentioned in the more general setting, this amounts to solving the equation whereby one sets the numerator of this leading coefficient equal to 0. By its factorisation given above, we either have $a_1 + a_2^2 - a^2 - b^2 = 0$ or $2a_1a_2^2 - a_1a^2 - a_1b^2 - a_2^2a^2 - a_2^2b^2 + a^4 + b^4 = 0$. Isolating the a_1 term in each case gives

$$a_1 = a^2 + b^2 - a_2^2, \quad \text{or} \quad a_1 = \frac{a_2^2(a^2+b^2) - a^4 - b^4}{2a_2^2 - a^2 - b^2}.$$

Thus after replacing a_1 in the definition of h with these new expressions found here the prior leading coefficient will be 0 and thus we get $\deg(S) = 4$ and $\deg(T) = 0$. In the first case, we have $T(x) \equiv 1/((a_2^2-a^2)(a_2^2-b^2))^2$ the polynomial S is irreducible over the function field $\mathbb{Q}(a_1, a_2, a, b)$ and can be explicitly computed as

$$S(x) = \frac{-1}{(a_2^2-a^2)^2(a_2^2-b^2)^2}\left(x^4 - (a^2+b^2)x^2 + a^2b^2 - 2(a_2^2-a^2)(a_2^2-b^2)\right).$$

Whereas in the second case we have $T(x) \equiv 1/C$, the polynomial S splits into a product of two irreducible quadratic polynomials that can be explicitly computed, with aid of the algebraic expression for a_1, as

$$S(x) = -\frac{1}{C}\left(x^2 - (a_1 + a_2^2 - a^2)\right)\left(x^2 - (a_1 + a_2^2 - b^2)\right),$$

and $C = \left((a_2^2-a^2)(a_2^2-b^2)(a^2-b^2)/(2a_2^2-a^2-b^2)\right)^2$. The reason why it splits is due to where $x^2 - a^2$ and $x^2 - b^2$ lie in the factorisation of f. We can write $f(x) = h(x)^2 - C = (h(x) - \sqrt{C})(h(x) + \sqrt{C})$ and note that $(h(x) - \sqrt{C})$ and $(h(x) + \sqrt{C})$ are both even polynomials. This means that $x^2 - a^2$ and $x^2 - b^2$ must factor one of these two polynomials. In the first case $(x^2 - a^2)(x^2 - b^2)$ will be equal to either $(h(x) - \sqrt{C})$ or $(h(x) + \sqrt{C})$ while in the second case $(x^2 - a^2)$ divides one of these polynomials and $(x^2 - b^2)$ will divide the other.

The rest of this exploration will assume that $a_1 = (a_2^2(a^2+b^2) - a^4 - b^4)/(2a_2^2 - a^2 - b^2)$. For certain rational choices of a_2, a, b, these quadratic factors might reduce to linear factors. Moreover, if $a_1, a_1 + a_2^2 - a^2, a_1 + a_2^2 - b^2$ are all squares, then the polynomial pair will split completely over the rationals – yielding a PTE solution. As mentioned earlier there are no known ideal PTE solutions of this type in the literature. Our experiments did not produce solutions a_2, a, b such that $a_1, a_1 + a_2^2 - a^2, a_1 + a_2^2 - b^2$ are all squares.

We can relax the condition that $a_1, a_1 + a_2^2 - a^2, a_1 + a_2^2 - b^2$ are all squares to only require two of them to be squares. This results in polynomials pairs with 10 linear factors and 1 quadratic factor. Plenty of polynomial pairs of this type can be found. The pair mentioned in Equation 2 would be found when $a_2 = 7/2$, $a = 1/2$ and $b = 11/2$ and is the example which features the smallest constant difference with $C = 1166400$. The next smallest can be found when $a_2 = 8$, $a = 3$ and $b = 12$ and, after applying the appropriate linear shift, results in the pair

$$\begin{aligned} f(x) &= (x+2)(x+9)(x+18)(x+24)(x+33)(x+40)(x^2+42x-55), \text{ and} \\ g(x) &= x^2(x+13)^2(x+29)^2(x+42)^2. \end{aligned}$$

which differ by $C = 564537600$. An example where a_1 is not a square but $a_1 + a_2^2 - a^2, a_1 + a_2^2 - b^2$ are both squares occur when $a_2 = 25/2$, $a = 3/2$ and $b = 45/2$. After applying the appropriate linear shift, it results in the polynomials

$$\begin{aligned} f(x) &= x(x+9)(x+10)(x+31)(x+34)(x+55)(x+56)(x+65), \text{ and} \\ g(x) &= (x+20)^2(x+45)^2(x^2+65x+154)^2. \end{aligned}$$

which differ by $C = 19209960000$. This polynomial pair would not be found using the initial approach since $h(x)$ does not split completely in this case.

We can relax this condition even further and only require one of these integers to be a square. This will results in polynomial pairs with 8 linear factors and 2 quadratic factors. While the smoothness probability

could decrease as a result of doing this, one can find instances whereby the constant difference between the pair is much smaller. Hence the smoothness probabilities are comparable to those that feature only 1 quadratic factor. In particular when $a_2 = 3/2$, $a = 1/2$, $b = 5/2$, one gets such a polynomial pair. It results in the polynomials

$$f(x) = (x+1)(x+3)(x+4)(x+6)(x^2+7x-2)(x^2+7x+4), \text{ and}$$
$$g(x) = x^2(x+2)^2(x+5)^2(x+7)^2.$$

which differ by $C = 576$. Moreover, for this specific polynomial pair, the evaluation of each residue class modulo C is 0. So there is no additional probability associated to the division by C when finding twin smooth integers from this pair.

Additionally, one example has been found whereby either a or b is 0. This introduces another square factor in the resulting polynomial pair which reduces the number of linear factors by one and hence decreases the smoothness probability compared to other pairs that feature two quadratic factors. This instance occurs when $a_2 = 2$, $a = 0$, and $b = 3$, resulting in the polynomial pair

$$f(x) = x(x+4)(x+7)^2(x+10)(x+14)(x^2+14x+9), \text{ and}$$
$$g(x) = (x+5)^2(x+9)^2(x^2+14x+4)^2.$$

which differ by $C = 32400$.

Now we look at the other setting when $n = 4$ and write $h(x) = (x^2 - a_1)$. Applying the XGCD algorithm to $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = h(x)^4$ with return polynomials S and T that have degrees 4 and 0 (resp.) only when either $2a_1^2 - 2a_1a^2 - 2a_1b^2 + a^4 + b^4 = (a_1 - a^2)^2 + (a_1 - b^2)^2 = 0$ or $a_1 = (a^2 + b^2)/2$. The first case can only occur when $a_1 = a^2 = b^2$ which we assume is not the case given that F and G to be coprime. In the second case, replacing $a_1 = (a^2 + b^2)/2$ results in the polynomial S which can be computed as

$$S(x) = -\frac{16}{(a^2-b^2)^4}\left(x^4 - (a^2+b^2)x^2 + \frac{a^4+b^4}{2}\right).$$

We already saw in the first case that S can never be factored into linear factors. So the question is can this be factored into irreducible quadratic factors or will it always be irreducible. If it can be factored, then the monic part of the polynomial will either be $(x^2-c)(x^2-d)$ or $(x^2-cx+d)(x^2+cx+d)$. The first situation can never happen for a similar reason as done above, so suppose that $S(x) = -16(x^2 - cx + d)(x^2 + cx + d)/(a^2 - b^2)^4$ for some c and d. By comparing the constant coefficients we have $a^4+b^4 = 2d^2$. As a consequence of Fermat's last theorem this equation has no rational solutions. Hence, for every $a, b \in \mathbb{Q}$, the polynomial $S \in \mathbb{Q}[x]$ will always be irreducible over the rationals.

Instead of choosing $F(x) = (x^2-a^2)(x^2-b^2)$, we revert back to $F(x) = (x^2-a^2)$ and $G(x) = (x^2-a_1^2)^4$. The idea is that one might be able to find pairs that split into a product of irreducible quadratic instead of having some linear factors. This is a special case of the computation from Example 1 with $a_2 = a_1$ and results in

$$S(x) = -\frac{1}{(a_1^2-a^2)^4}\left(x^2 - 2a_1^2 + a^2\right)\left(x^4 - 2a_1^2x^2 + 2a_1^4 - 2a_1^2a^2 + a^4\right).$$

Note that when asserting that $2a_1^2 - a^2$ is a square we do back to the previous setting. Thus it suffices to check when the quartic polynomial factors into a product of two quadratics. A search finds a relatively small instance with $a_1 = 3/2$ and $a = 7/2$. This results in the following polynomial pair

$$f(x) = x(x+7)(x^2+2x+5)(x^2+7x+20)(x^2+12x+40), \text{ and}$$
$$g(x) = (x+2)^4(x+5)^4,$$

which differ by $C = 10000$.

5.2 Degree 10 Polynomials

Now we deal with the search for degree 10 polynomials. Once again we work with the rational function field $\mathbb{Q}(a_1, a_2, a, b)$ and its polynomial ring $\mathbb{Q}(a_1, a_2, a, b)[x]$.

Choose $n = 2$ and $h(x) = x(x^2 - a_1)(x^2 - a_2^2) \in \mathbb{Q}(a_1, a_2, a, b)[x]$. We shall apply the XGCD algorithm to the polynomials $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = h(x)^2$ to give two polynomials S and T. These polynomials have degrees $\deg(S) = 8$ and $\deg(T) = 2$. Occassionally, these degrees will be $\deg(S) = 6$ and $\deg(T) = 0$. Once again this occurs when the numerator of the leading coefficient of S is 0. Isolating the a_1 term gives either

$$a_1 = \frac{a_2^2(a^2 + ab + b^2) - a^4 - a^3b - a^2b^2 - ab^3 - b^4}{a_2^2 - a^2 - ab - b^2}, \text{ or}$$
$$a_1 = \frac{a_2^2(a^2 - ab + b^2) - a^4 + a^3b - a^2b^2 + ab^3 - b^4}{a_2^2 - a^2 + ab - b^2}.$$

These two resulting expressions are the same up to changing the sign of one of either a or b. Thus the forthcoming computation is the same in either choicee – so we simply do it for the first of these expressions. Replacing the first expression for a_1 into the polynomials F, G, S and T, one gets $\deg(S) = 6$ and $\deg(T) = 0$. Concretely, we have $T(x) = 1/C$ and

$$S(x) = -\frac{1}{C}\left(x^3 + (a+b)x^2 + c_1x + c_2\right)\left(x^3 - (a+b)x^2 + c_1x - c_2\right),$$

where $c_1, c_2 \in \mathbb{Q}(a_2, a, b)$ are algebraically computable expressions and

$$C = \left(\frac{ab(a+b)(a_2^2 - a^2)(a_2^2 - b^2)}{a_2^2 - a^2 - ab - b^2}\right)^2 \in \mathbb{Q}(a_2, a, b).$$

Over $\mathbb{Q}(a_2, a, b)[x]$, the cubic polynomials in the factorisation of S are irreducible. However, after evaluating the variables a_2, a, b with either integers or rationals to make these polynomials in $\mathbb{Q}[x]$, these cubic polynomials could be reducible. Note that, given the shape of these cubic polynomials, if one of them has a root α then the other polynomial must have a root $-\alpha$. This also follows from the fact that the starting polynomial $h(x)$ is an odd function.

In the circumstances when the cubic polynomials can be factored, then the resulting polynomial pair have, at the very least, 9 combined linear factors and 3 combined quadratic factors. The instance that gives the smallest possible integer C occurs when $a_2 = 1$, $a = 4$ and $b = 6$, resulting in the polynomial pair

$$f(x) = x(x+1)(x+3)(x+11)(x+13)(x+14)(x^2 + 11x + 38)$$
$$(x^2 + 17x + 80), \text{ and}$$
$$g(x) = (x+6)^2(x+7)^2(x+8)^2(x^2 + 14x + 5)^2.$$

with $C = 2822400$.

One can ask when one or more of the quadratic factors can be factored into two linear factors. The case when all quadratic factors can be factor into linear factors results in a PTE solution. However, as already mentioned, extensive searches have been done to find such solutions but only one has been found and it does not have any repeated factors. Additionally, we note that the setting where one (and hence both) of the quadratic factors in S is reducible can not happen since it would yield a PTE solution of size 5 that has repeated factors. Again such solutions do not exist in the literature even with extensive searches. The final case is the setting when a_1 is a square. This can happen and does so on many occasions – resulting in polynomials f, g such 11 combined linear factors and 2 combined quadratic factors. The example that gives the smallest possible integer value of C in this setting occurs when $a_2 = 8$, $a = 1$ and $b = 7$, resulting in the

polynomial pair

$$\begin{aligned} f(x) &= (x+1)(x+4)(x+10)(x+12)(x+18)(x+21)(x^2+20x-9) \\ &\qquad (x^2+24x+35), \text{ and} \\ g(x) &= x^2(x+3)^2(x+11)^2(x+19)^2(x+22)^2. \end{aligned} \tag{4}$$

with $C = 57153600$.

We make a few brief comments on the other setting with $n = 5$. The XGCD computation over $\mathbb{Q}(a_1, a)[x]$ of the polynomials $F(x) = (x^2 - a^2)$ and $G(x) = (x^2 - a_1^2)^5$ gives rise to the polynomials S and T with $\deg(S) = 8$ and $\deg(T) = 0$. Over $\mathbb{Q}(a_1, a)[x]$, the polynomial S is irreducible and, after evaluating the variables a_1, a at some rationals, the best one can hope for is that the polynomial S to be factored into a product of two irreducible degree 4 factors. However, no examples have been found where the polynomial S factors in this way.

5.3 Degree 12 Polynomials

We finally detail the search for degree 12 polynomial pairs. This time we work with a slightly larger rational function field than that for the previous settings. In particular, we work with $\mathbb{Q}(a_1, a_2, a_3, a, b)$ and its polynomial ring $\mathbb{Q}(a_1, a_2, a_3, a, b)[x]$.

Choose $n = 2$ and $h(x) = (x^2 - a_1)(x^2 - a_2^2)(x^2 - a_3^2) \in \mathbb{Q}(a_1, a_2, a_3, a, b)[x]$. Apply the XGCD algorithm to the polynomials $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = h(x)^2$ which results in two polynomials S and T with $\deg(S) = 10$ and $\deg(T) = 2$. When the numerator of the leading coefficient of S and T is 0, then $\deg(S) = 8$ and $\deg(T) = 0$. Isolating the a_1 from this numerator gives either

$$\begin{aligned} a_1 &= \frac{(a_2^2 + a_3^2)(a^2 + b^2) - (a_2^2 a_3^2 + a^4 + a^2 b^2 + b^4)}{a_2^2 + a_3^2 - a^2 - b^2}, \text{ or} \\ a_1 &= \frac{a_2^2 a_3^2 (a^2 + b^2) - (a_2^2 + a_3^2)(a^4 + b^4) + a^6 + b^6}{2a_2^2 a_3^2 - (a_2^2 + a_3^2)(a^2 + b^2) + a^4 + b^4}. \end{aligned}$$

The choice one chooses for a_1 results in different factorisation structure for the polynomial S. For the first choice, the polynomial S splits as $S = S_2 \cdot S_6/C$ where S_2 is an irreducible monic quadratic, S_6 is an irreducible degree 6 monic polynomial over $\mathbb{Q}(a_2, a_3, a, b)[x]$ and

$$C = \left(\frac{(a_2^2 - a^2)(a_2^2 - b^2)(a_3^2 - a^2)(a_3^2 - b^2)}{a_2^2 + a_3^2 - a^2 - b^2} \right)^2.$$

Moreover, the polynomial S as well as its irreducible factors are even polynomials in the sense that $S(x) = S'(x^2)$ for some polynomial S' (and same for its factors). Now the question is when does S factorise into a product of at most quadratic factors after evaluating the variables of the function field. Since S_2 is automatically quadratic, this only depends on the polynomial S_6. The strategy to efficiently check whether it has the desired factorisation structure is to only factorise S_6 when the associated cubic polynomial S_3', where $S_6(x) = S_3'(x^2)$, has a root. This step can easily be checked by the rational root test. Now it is a matter of searching for $a_2, a_3, a, b \in \mathbb{Q}$ such that one gets this factorisation. Choosing $a_2 = 1/2$, $a_3 = 13/2$, $a = 5/2$ and $b = 7/2$ results in polynomial pairs that have 4 quadratic factors. This results in

$$\begin{aligned} f(x) &= (x+1)(x+3)(x+4)(x+9)(x+10)(x+12)(x^2+13x-3) \\ &\qquad (x^2+13x+6)(x^2+13x+45), \text{ and} \\ g(x) &= x^2(x+6)^2(x+7)^2(x+13)^2(x^2+13x+21)^2. \end{aligned}$$

which differ by $C = 10497600$. Additionally choosing $a_2 = 19/2$, $a_3 = 43/2$, $a = 23/2$ and $b = 29/2$ the results in polynomial pairs that have 3 quadratic factors. Concretely, this pair is

$$\begin{aligned} f(x) &= (x+1)(x+3)(x+7)(x+10)(x+33)(x+36)(x+40)(x+42) \\ &\qquad (x^2+43x-24)(x^2+43x+396) \text{ and} \\ g(x) &= x^2(x+12)^2(x+31)^2(x+43)^2(x^2+43x+186)^2. \end{aligned}$$

which differ by $C = 3983377305600$.

For the second choice, the polynomial S splits into a product of two distinct irreducible quartic polynomials over $\mathbb{Q}(a_2, a_3, a, b)[x]$. Write this as product as $S = S_4 \cdot S_4'/C$ where S_4, S_4' are irreducible degree 4 monic polynomials and

$$C = \left(\frac{(a_2^2 - a^2)(a_2^2 - b^2)(a_3^2 - a^2)(a_3^2 - b^2)(a^2 - b^2)}{2a_2^2a_3^2 - (a_2^2 + a_3^2)(a^2 + b^2) + a^4 + b^4} \right)^2.$$

Once again, each polynomial S, S_4, S_4' are even polynomials. Now one has to check when the polynomials S_4 and S_4' factorise into quadratic polynomials. Since these polynomials are even, this can be done with no polynomial arithmetic. This was discussed towards the end of §5.1 and the idea is that any polynomial of the form $x^4 + Ax^2 + B$ factorises either into $(x^2 - \alpha)(x^2 - \beta)$ or $(x^2 - \alpha x + \beta)(x^2 + \alpha x + \beta)$ for some α and β. Note that these quadratic factors might not be irreducible but the point is that if the quartic can be factored then it has must have at most quadratic factors. The first case boils down to doing some discriminant calculation, namely checking if $A^2 - 4B$ is a square. In the second case, there are a few arithmetic checks needed: firstly $B = \beta^2$ must be a square and then either $2\beta - A$ or $-2\beta - A$ must be a square. All of this arithmetic computation and lead us to determine the factorisation of the polynomials S_4 and S_4'. In particular, choosing $a_2 = 3$, $a_3 = 4$, $a = 1$ and $b = 2$ results in polynomial pairs that have 5 quadratic factors. This results in

$$\begin{aligned} f(x) &= (x+2)(x+3)(x+5)(x+6)(x^2+8x-1)(x^2+8x+2) \\ &\qquad (x^2+8x+4)(x^2+8x+10), \text{ and} \\ g(x) &= x^2(x+1)^2(x+7)^2(x+8)^2(x^2+8x+14)^2. \end{aligned}$$

which differ by $C = 14400$. Additionally, choosing $a_2 = 14$, $a_3 = 39$, $a = 3$ and $b = 31$ the resulting polynomials are those mentioned in Equation (3) which have 2 quadratic factors.

In the setting with $n = 3$ one chooses $h(x) = (x^2 - a_1^2)(x^2 - a_2^2)$ and then does the XGCD computation to the polynomials $F(x) = (x^2 - a^2)$ and $g(x) = h(x)^3$. However, this does not lead to polynomials that factor into small degree factors.

Choosing $n = 4$ is slightly more fruitful than the previous choice. Here one chooses $h(x) = x(x^2 - a_1^2)$ and then do the XGCD computation to the polynomials $F(x) = (x^2 - a^2)$ and $G(x) = h(x)^4$. This gives two polynomials $S(x)$ and $T(x)$. The best one can expect after evaluating the variables at rationals, is to be able to factor the polynomial S into at most quartic factors. However, if one loosens the polynomial h to be $h(x) = x(x^2 - a_1)$ then one can find instances where the polynomial S factor into at most quadratic factors. However the number of linear and quadratic factors that appear in the resulting polynomials are 3 and 6 (resp.). This results in a worse smoothness probability than those found using $n = 2$.

The final setting with $n = 6$ can be done by choosing $F(x) = (x^2 - a^2)$ and $G(x) = (x^2 - a_1^2)^6$. This is a special case of the $n = 3$ setting with $a_1 = a_2$. Hence it does not result in any meaningful pairs.

5.4 Other Degrees

We make a few comments on searches that could be made to find other degree polynomials. We leave the implementation of these searches as future work.

Larger even degrees. The same strategy can be applied for finding polynomials of degree 14, 16 and larger even degrees. For instance, to find degree 14 polynomials, one would do the XGCD computation of the polynomials $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = h(x)^2$ where $h(x) = x(x^2 - a_1)(x^2 - a_2^2)(x^2 - a_3^2)$. Subsequently, one determines the algebraic expression for a_1 in order to determine when the degrees of the resulting polynomials S and T are as small as possible. Once this is done one attempts to factor the polynomials until it has small degree factors. A similar design can be done for the degree 16 search.

The XGCD precomputation that would be required in these settings would be a lot more involved compared to the searches that we did for degree $8, 10$ and 12 polynomials. Additionally, expecting to find polynomial pairs that factor into at most quadratic factors would take much longer compared to the other degrees. We note that these polynomials could be quite useful if one needs to find smooth twins that are much larger than 512-bits. For instance this would be the case in a B-SIDH [13] variant of proposals [20,2] that offer potential countermeasures to the now famous polynomial time attacks [10,23,28] on SIDH [21].

Odd degrees. A common theme in all of the conducted searches is that one does a search with even polynomials. This strategy can not be applied in the same way when searching for odd degree polynomials. Additionally, the search for odd degree n pairs of the form $f(x) = h(x)^k - C$ and $g(x) = h(x)^k$ only becomes fruitful when n is composite. However, most small odd integers are prime and so this strategy will not result in anything more than the polynomial pair $f(x) = x^n - 1$ and $g(x) = x^n$.

We demostrate an alternative strategy that could be used when searching for odd degree pairs. We do this to find degree 7 polynomials but it generalises to other odd degrees. Start with the polynomials $F(x) = x^2(x+a)^2$ and $G(x) = (x+a_1)^2(x+a_2)^2$. The result of applying the XGCD algorithm over $\mathbb{Q}(a_1, a_2, a)[x]$ gives two degree 3 polynomials S and T. Over $\mathbb{Q}(a_1, a_2, a)[x]$, these cubic polynomials are irreducible but, after evaluating the variables at rationals, they might factor into at most quadratic factors. This simple strategy was experimented with but the resulting polynomials pairs did not give a good smoothness probability compared to the even degree pairs. An alternative choice for polynomial F and G might be favourable and we leave this direction open.

6 Cryptographic Instances

We implemented the search for twin smooth integers and primes p such that $p^2 - 1$ is smooth using these new polynomials. The implementation uses the PTE Python3 code [27] as a foundational starting point. In particular we use the C implementation of the sieve of Eratothenes (see Step 2 from §4.2) from that code base to identify smooth integers in an interval. Additionaly, we include code for both sieving with multiple polynomial pairs as done in [14] but also a sieving with just a single polynomial pair as done in [1]. The degree 8 polynomials benefited from searching with multiple solutions since many pairs share some roots and also have a small C. On the other hand, the degree 10 and 12 polynomials benefited from searching with just a single solution and only check those ℓ with $f(\ell) \equiv g(\ell) \equiv 0 \mod C$. This is because among the polynomial pairs found the difference C are vastly different. Hence, the search for b-bit twins would require searching on completely different intervals when utilising different polynomial pairs. Additionally, this proportion of integers ℓ is quite small, so only searching with those ℓ's saves lots of computation. We chose to implement Step 5a, to deal with checking whether the evaluations of the quadratic factors are smooth. This post-processing can be done in the Python code itself but we found it beneficial to do it in a computer algebra system, such as Magma, whereby fast factoring algorithms are implemented. This strategy is particular effective when the number of quadratic factors is at most 2 and also the search using the degree 12 polynomials that consist of three quadratic factors. For the other polynomials, it might be benefitical to do one of the other strategies (either Step 5b or Step 5c) for this post-processing. This was not implemented in this work and is left as an avenue for future research.

Experiments were conducted to find 256, 384 and 512-bit instances using Table 2 to assess which smoothness bound to choose for these searches. We ran these experiments on a server with a Xeon E7-4850v2 2.30GHz, 1007GB of RAM. The total number of parallel threads available on this machine is 96. Table 3 records the best results from these experiments as well as including the results from prior work.

Method	Where	$\lceil \log_2(p) \rceil$	$(B, \lceil \log_2(B) \rceil)$
XGCD over $\mathbb{Z}$	[4, App. A]	256	$(6548911, 23)$
Cyclotomic factors	[13, Ex.5]	247	$(652357, 20)$
	[13, Ex.8]	250	$(486839, 19)$
	[18, §6.2]	388	$(20884693, 25)$
PTE sieve	[14, p_{241}]	241	$(32039, 15)$
	[14, p_{245}]	245	$(49711, 16)$
	[14, p_{246}]	246	$(40151, 16)$
	[14, p_{247}]	247	$(40289, 16)$
	[14, p_{249}]	249	$(38119, 16)$
	[14, p_{250}]	250	$(32191, 15)$
	[14, p_{252}]	252	$(35291, 16)$
	[14, p_{255}]	255	$(52069, 16)$
	[14, p_{257}]	257	$(42979, 16)$
	[14, p_{376}]	376	$(1604719, 21)$
	[14, p_{384}]	384	$(3726773, 22)$
	[14, p_{512}]	512	$(238733063, 28)$

Method	Where		$\lceil \log_2(p) \rceil$	$(B, \lceil \log_2(B) \rceil)$
XGCD over $\mathbb{Q}[\mathbf{x}]$	10622157951	XGCD_5^8	239	$(69833, 17)$
	5128781232	XGCD_7^8	239	$(73771, 17)$
	13024987664	XGCD_6^8	250	$(77029, 17)$
	19371175757	XGCD_6^8	255	$(77687, 17)$
	38295031104	XGCD_6^8	263	$(42577, 16)$
	1447964653205910	XGCD_7^8	375	$(1915117, 21)$
	2054426379410766	XGCD_7^8	379	$(1502581, 21)$
	1213633306317077	XGCD_6^8	382	$(1445533, 21)$
	1435534016858820	XGCD_6^8	384	$(1296167, 21)$
	1471680421245912	XGCD_6^8	384	$(1140157, 21)$
	2062439636622939	XGCD_6^8	388	$(1733527, 21)$
	2249069326428162	XGCD_6^8	389	$(1932703, 21)$
	1290853259901	XGCD_2^{10}	378	$(1766099, 21)$
	12538742222491880	XGCD_2^{10}	511	$(34102657, 26)$
	14334163549504404	XGCD_2^{10}	512	$(43346161, 26)$
	64343906330928	XGCD_5^{12}	510	$(42485491, 26)$
	188327931771336	XGCD_8^{12}	511	$(24984383, 25)$
	192093987758508	XGCD_8^{12}	512	$(20003833, 25)$

Table 3: A list and comparsion of cryptographic-sized primes p such that $p^2 - 1$ is B-smooth. The primes found our method consist of an integer along with a polynomial pair, XGCD_i^n, where the integer is used as an input to the polynomial pair to derive the prime. The description of these polynomial pairs can either be found in the text or in Appendix C.

***b* = 256.** Our polynomial pairs have degree at least 8, so one suffers with a reduced search space when finding 256-bit instances. When searching with the degree 8 pairs, the interval $[2^{31}, 2^{37}]$ scans all 256-bit instances. Table 2, says the approximate probability of finding B-smooth twins using these pairs is at most 2^{-39} with $B = 2^{16}$. This suggests that a search with this B would be unsuccessful. As a result, we ran our code in this interval with $B = 2^{17}$ and found a few primes p with $p^2 - 1$ being 2^{17}-smooth. Of these primes, only one prime was found whose smoothness bound is 2^{16}. This prime was found using the polynomial pair from Equation (2). Concretely this prime is

$$p = \frac{2\ell^2(\ell+6)^2(\ell+13)^2(\ell+19)^2}{1166400} - 1,$$

with $\ell = 38295031104$. In contrast, the search with the PTE sieve [14] produced some smoother instances. The main reason for this is because they searched using the degree 6 polynomials that completely split into linear factors and feature repeated linear factors. This not only gives an optimal smoothness probability when searching using degree 6 polynomials but also increases the search space to a point whereby one can guarantee finding such instances.

***b* = 384.** When searching for larger instances, the search space limitations become less of an issue as long as we use polynomials whose degree is not too large. When searching for 384-bit instances, the degree 8 polynomials offer the best option for decreasing the smoothness bound. They allow for a large search space while also attempting to minimise the smoothness bound. The search with $B = 2^{22}$ using our degree 8 polynomials found lots of primes p with $p^2 - 1$ being 2^{22}-smooth. Of these primes lots of primes were found that are 2^{21}-smooth. This surpasses the smoothness bound of both instances that were recorded from the PTE sieve. The best prime found from this search is

$$p = \frac{2\ell^2(\ell+6)^2(\ell+13)^2(\ell+19)^2}{1166400} - 1,$$

with $\ell = 1471680421245912$. Additionally, we searched with the degree 10 polynomials with $B = 2^{22}$. This search produced fewer primes compared to the degree 8 search but one was found with $B = 2^{21}$.

$b = 512$. This final setting will result in the most gain in reducing the size of the smoothness bound. The searches done with the PTE sieve found such instances whose smoothness bound is $B = 2^{28}$. According to the probabilities mentioned in Table 2, the polynomials found in this work should offer smoother instances when searching using the degree 10 and degree 12 polynomials. These searches using $B = 2^{26}$ and found lots of instances. Among those found using the degree 10 polynomials, the pair of degree 10 polynomials from Equation (4) proved to be most effective in finding instances that minimise the smoothness bound. The smoothest instance found from this experiment results from the following prime

$$p = \frac{2\ell^2(\ell+3)^2(\ell+11)^2(\ell+19)^2(\ell+22)^2}{57153600} - 1,$$

with $\ell = 14334163549504404$. Of the primes found using the degree 12 polynomials, two were found with p^2-1 being 2^{25}-smooth. These were found using the single degree 12 polynomial, mentioned in Equation (3), that features only 2 quadratic factors. Concretely, the larger of these primes is

$$p = \frac{2\ell^2(\ell+14)^2(\ell+39)^2(\ell+67)^2(\ell+92)^2(\ell+106)^2}{626762289162240000} - 1,$$

with $\ell = 192093987758508$.

Acknowledgements. The author of this work would like to thank Maria Corte-Real Santos, Craig Costello, Jonathan Komada Eriksen and Robert Granger for the many valuable discussions that helped shaped the work to its final presentation. Additionally, the author would like to thank Michael Meyer for his insights to the PTE experiments as well as Benjamin Smith for his assistance with running the experiments and comments to drafts of the work. This work was supported by the HYPERFORM consortium funded by France through Bpifrance.

References

1. K. Ahrens. Sieving for large twin smooth integers using single solutions to prouhet-tarry-escott. Cryptology ePrint Archive, Paper 2023/219, 2023. https://eprint.iacr.org/2023/219.
2. A. Basso and T. B. Fouotsa. New sidh countermeasures for a more efficient key exchange. Cryptology ePrint Archive, Paper 2023/791, 2023. https://eprint.iacr.org/2023/791.
3. D. J. Bernstein. How to find smooth parts of integers. *URL: http://cr. yp. to/papers. html# smoothparts. ID 201a045d5bb24f43f0bd0d97fcf5355a. Citations in this document*, 20, 2004.
4. D. J. Bernstein, L. De Feo, A. Leroux, and B. Smith. Faster computation of isogenies of large prime degree. *Open Book Series*, 4(1):39–55, 2020.
5. P. Borwein, P. Lisoněk, and C. Percival. Computational investigations of the prouhet-tarry-escott problem. *Mathematics of computation*, 72(244):2063–2070, 2003.
6. G. Bruno, L. Batina, and W. Bosma. Crypto security optimizations. *Radboud University Nijmegen: Nijmegen, The Netherlands*, 2021.
7. G. Bruno, M. Corte-Real Santos, C. Costello, J. K. Eriksen, M. Naehrig, M. Meyer, and B. Sterner. Cryptographic smooth neighbors. 2022. https://eprint.iacr.org/2022/1439.
8. J. Buzek, J. Hasan, J. Liu, M. Naehrig, and A. Vigil. Finding twin smooth integers by solving pell equations. 2022.
9. T. Caley. The prouhet-tarry-escott problem. 2013.
10. W. Castryck and T. Decru. An efficient key recovery attack on SIDH. In *EUROCRYPT*, volume 14008 of *Lecture Notes in Computer Science*, pages 423–447. Springer, 2023.
11. J. Chavez-Saab, M. Corte-Real Santos, L. De Feo, J. K. Eriksen, B. Hess, D. Kohel, A. Leroux, P. Longa, M. Meyer, L. Panny, et al. SQIsign, 2023. https://sqisign.org.

12. J. B. Conrey, M. A. Holmstrom, and T. L. McLaughlin. Smooth neighbors. *Experimental Mathematics*, 22(2):195–202, 2013.
13. C. Costello. B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion. In *ASIACRYPT*, volume 12492 of *Lecture Notes in Computer Science*, pages 440–463. Springer, 2020.
14. C. Costello, M. Meyer, and M. Naehrig. Sieving for twin smooth integers with solutions to the Prouhet-Tarry-Escott problem. In *EUROCRYPT*, volume 12696 of *Lecture Notes in Computer Science*, pages 272–301. Springer, 2021.
15. R. Crandall and C. Pomerance. *Prime numbers*. Springer, 2001.
16. N. G. de Bruijn. On the number of positive integers $\leq$ x and free of prime factors $> y$, ii. *Indag. Math*, 38:239–247, 1966.
17. K. Dickman. On the frequency of numbers containing prime factors of a certain relative magnitude. *Arkiv for matematik, astronomi och fysik*, 22(10):A–10, 1930.
18. L. De Feo, C. Delpech de Saint Guilhem, T. B. Fouotsa, P. Kutas, A. Leroux, C. Petit, J. Silva, and B. Wesolowski. Séta: Supersingular encryption from torsion attacks. In *ASIACRYPT*, volume 13093 of *Lecture Notes in Computer Science*, pages 249–278. Springer, 2021.
19. L. De Feo, D. Kohel, A. Leroux, C. Petit, and B. Wesolowski. SQISign: compact post-quantum signatures from quaternions and isogenies. In *ASIACRYPT*, volume 12491 of *Lecture Notes in Computer Science*, pages 64–93. Springer, 2020.
20. T. B. Fouotsa, T. Moriya, and C. Petit. M-SIDH and MD-SIDH: Countering sidh attacks by masking information. In *EUROCRYPT*, volume 14008 of *Lecture Notes in Computer Science*, pages 282–309. Springer, 2023.
21. D. Jao and L. De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *PQCrypto*, volume 7071 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2011.
22. D. H. Lehmer. On a problem of Störmer. *Illinois Journal of Mathematics*, 8(1):57–79, 1964.
23. L. Maino, C. Martindale, L. Panny, G. Pope, and B. Wesolowski. A direct key recovery attack on SIDH. In *EUROCRYPT*, volume 14008 of *Lecture Notes in Computer Science*, pages 448–471. Springer, 2023.
24. G. Marsaglia, A. Zaman, and J.C.W. Marsaglia. Numerical solution of some classical differential-difference equations. *Mathematics of Computation*, 53(187):191–201, 1989.
25. G. Martin. An asymptotic formula for the number of smooth values of a polynomial. *Journal of Number Theory*, 93:108–182, 1999.
26. P. L. Montgomery. Speeding the pollard and elliptic curve methods of factorization. *Mathematics of computation*, 48(177):243–264, 1987.
27. Microsoft Research. Twin smooth integers. `https://github.com/microsoft/twin-smooth-integers`, 2021.
28. D. Robert. Breaking SIDH in polynomial time. In *EUROCRYPT*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503. Springer, 2023.
29. V. Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, USA, 2 edition, 2009.
30. C. Størmer. Quelques théorèmes sur l'équation de Pell $x^2 - dy^2 = \pm 1$ et leurs applications. *Christiania Videnskabens Selskabs Skrifter, Math. Nat. Kl*, (2):48, 1897.
31. The National Institute of Standards and Technology (NIST). Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, December, 2016.
32. The National Institute of Standards and Technology (NIST). Call for additional digital signature schemes for the post-quantum cryptography standardization process, October, 2022.
33. 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(106):417–421, 1969.

A Choosing incorrect polynomials for cryptographic applications

We have to be careful in how we choose F, G in the general description given in §4.2 if we wish to find twins whose sum is a prime. Here we describe a certain class of polynomials for which one will categorically never find primes when using these polynomials. Thus cannot be used for cryptographic purposes. One example of such an instance is when $F(x) = (x+1)^n$ and $G(x) = x^n$. When summing the polynomial pair that arise from this choice it will always contain $2x + 1$ as a factor.

Proposition 1. *Let n be a positive integer and $F, G \in \mathbb{Q}[x]$ be polynomials with $m \coloneqq F - G \in \mathbb{Q}$. Let $S_n, T_n \in \mathbb{Q}[x]$ be the result of applying the XGCD algorithm to the polynomials $F^n \coloneqq F \cdot F \cdots F$ and $G^n \coloneqq G \cdot G \cdots G$. Then there is a constant $C_{n,F,G} \in \mathbb{Q}$ such that*

$$\begin{aligned} F(x)S_{n+1}(x) - S_n(x) &= \quad C_{n,F,G}G(x)^n(F(x)+G(x)), \quad \text{and} \\ G(x)T_{n+1}(x) - T_n(x) &= -C_{n,F,G}F(x)^n(F(x)+G(x)). \end{aligned}$$

Moreover, in the setting when $F(x) = x+1$ and $G(x) = x$, the polynomial $H_n(x) \coloneqq F^n(x)S_n(x) - G^n(x)T_n(x)$ will always have $2x+1$ as a factor.

Proof. By construction, F and G are coprime, so we have

$$F(x)^n S_n(x) + G(x)^n T_n(x) = 1, \quad \deg(S_n) = \deg(T_n) < n. \tag{$*$}$$

In order to prove the proposition, we need two key ingredients. These are formulated in the following claims.

Claim. For every n, we have the following $\deg(S_n) = \deg(T_n) = \deg(G) \cdot (n-1)$.

Proof. Let's start by applying the Euclidean algorithm step by step to $F(x)^n$ and $G(x)^n$. Write $F(x)^n = (G(x)+m)^n = G(x)^n + R_1(x)$. It is straight forward to see that $\deg(r_1) = \deg(G)(n-1)$. Now write $G(x)^n = Q_2(x)R_1(x) + R_2(x)$ with $\deg(R_2) < \deg(R_1)$. Again, it is straight forward to see that $\deg(Q_2) = \deg(G)$ and $\deg(R_2) = \deg(G)(n-2)$ since the $x^{\deg(G)(n-2)}$ coefficient of $Q_2(x)R_1(x)$ is non-zero. One can repeat this iteratively and deduce that $\deg(Q_k) = \deg(G)$ and $\deg(R_k) = \deg(G)(n-k)$ for all $2 \le k \le n$.

Using these polynomials, we can apply the extended Euclidean algorithm to recover the polynomials S_n and T_n. In fact, using Theorem 17.4(iv) from [29], we obtain the desired result.

Claim. For every n, we have the following closed forms for S_n and T_n:

$$\begin{aligned} S_n(x) &= m^{-n} \sum_{k=0}^{n-1} \binom{n+k-1}{k} (-G(x)/m)^k, \text{ and} \\ T_n(x) &= (-m)^{-n} \sum_{k=0}^{n-1} \binom{n+k-1}{k} (F(x)/m)^k. \end{aligned}$$

Proof. Let $p_n(x) = m^{-n} \sum_{k=0}^{n-1} \binom{n+k-1}{k} (-G(x)/m)^k$. Since $\deg(S_n) = \deg(p_n) = \deg(g)(n-1)$ then, by the uniqueness of S_n, T_n, it suffices to show that $F(x)^n p_n(x) - 1$ is divisible by $G(x)^n$.

Firstly, write $F(x) = m(G(x)/m+1)$. When multiplying $F(x)^n$ and $p_n(x)$, write the result as a polynomial in $G(x)$ – so we have $F(x)^n p_n(x) = \sum_{k=0}^{2n-1} a_k G(x)^k$. For $k < n$ the coefficient a_k in this product is equal to

$$m^{-k} \sum_{i=0}^{k} \binom{n}{i} \binom{n+k-i-1}{k-i} (-1)^{k-i}.$$

When $k = 0$ this is equal to 1. As a simple exercise in using the "Upper Negation" and Vandermonde's identities for binomial coefficients, when $0 < k < n$ this is equal to 0. This proves the closed form of S_n and proving the closed form of T_n can be done with a similar strategy.

With the closed form for S_n it is an algebraic exercise to obtain the first of these formulas. In the process one shows that $C_{n,F,G} = m^{-2n-1}(-1)^n \binom{2n-1}{n-1}$. One could do the same algebraic exercise for T_n but by considering the equation $(*)$ for n and $n+1$, we have

$$F(x)^n \left(F(x)S_{n+1}(x) - S_n(x)\right) = G(x)^n \left(T_n - G(x)T_{n+1}(x)\right).$$

This shows that the second of these formulas can easily be found from the first.

Now suppose that $F(x) = x+1$ and $G(x) = x$. Using the derived recursive formulas for S_n, T_n and by induction, we have $S_n(-1/2) = 2^{n-1}$ and $T_n(-1/2) = -(-2)^{n-1}$. Plugging $-1/2$ into the defining formula for H_n gives the result. ∎

Remark 4. The concluding statement in the proposition is not specific to $F(x) = x + 1$ and $G(x) = x$ and it applies more generally. In the general setting the factor that appears is the polynomial $F + G$. As another remark, it is straight forward to adopt the arguments addressed above to the setting when F is a linear transform of G (i.e. $F(x) = aG(x) + b$)

B New ideal PTE solutions

We demonstrate an application of this XGCD approach to find new PTE solutions that have not been recorded in the literature. In particular, we present a new parametrised family of size 4 ideal PTE solutions that feature one repetition on both sides. Such PTE solutions was considered to find smooth B-SIDH [13] parameters before the polynomial time attacks on SIDH surfaced [10,23,28]. There one requires to find primes p such that there are large smooth cofactors of $p+1$ and $p-1$ which are roughly the same size. The repetition in both sides of these PTE solution would have made our chances of finding such parameters more profitable.

Proposition 1 *Let α, β be rational parameters and let*

$$\begin{aligned} a &= \alpha\beta(\beta+1)(\beta^2-2\beta+3),\\ b &= \alpha\beta(\beta+1)(\beta^2+1),\\ c &= \alpha(\beta^2+1)(\beta^2+2\beta-1),\\ d &= \alpha(\beta+1)(\beta-1)^3,\\ e &= 4\alpha\beta, \text{ and}\\ C &= \alpha^4\beta^3(\beta-1)^3(\beta+1)^3(\beta^2+1)^2. \end{aligned}$$

Then the following two polynomials in $\mathbb{Q}[\alpha, \beta]$

$$\begin{aligned} f(x) &= x(x+a)^2(x+c), \text{ and}\\ g(x) &= (x+b)^2(x+d)(x+e) \end{aligned}$$

differ by C.

For concreteness of an example, when choosing $\alpha = -16$ and $\beta = -1/2$, one gets $a = 17$, $b = 5$, $c = 35$, $d = 27$ and $e = 32$. One can easily show that, among all such PTE solutions for which a, b, c, d, e are all positive integers and all coprime, this is the smallest solution – as emphasised by the solutions listed in Table 4. Additionally, it is a straight forward algebraic computation to check the proposition however the proof we give derives these algebraic formulae as a result of some XGCD computation.

Proof. We will work with polynomials with coefficients over a function field $\mathbb{Q}(\alpha, \beta)$ and start out by applying the XGCD algorithm over this field to the polynomials $F(x) \coloneqq x(x+a)^2$ and $G(x) \coloneqq (x+b)^2$ where $a, b \in \mathbb{Q}(\alpha, \beta)$ are elements of the function field to be chosen later. This results in the polynomials S, T such that $\deg(S) = 1$ and $\deg(T) = 2$ which are

$$\begin{aligned} S(x) &= \frac{-a+3b}{b^2(a-b)^3}x + \frac{-2a+4b}{b(a-b)^3}\\ T(x) &= \frac{a-3b}{b^2(a-b)^3}x^2 + \frac{2a^2-6ab+2b^2}{b^2(a-b)^3}x + \frac{1}{b^2}. \end{aligned}$$

Since the polynomial s is linear it only suffices to check when t factors. This happens only when its discriminant is a square: $\mathrm{disc}(T) = D^2$. If we let $\mathbf{a} = b(a-b)^3D$, $\mathbf{b} = 2a - 4b$ and $\mathbf{c} = 2b$, then this is equivalent to solving the equation

$$\mathbf{a}^2 + \mathbf{b}^2 = 2\mathbf{c}^2.$$

a	b	c	d	e	a	b	c	d	e	a	b	c	d	e
17	5	35	27	32	26010	8070	88501	36501	87880	93456	29616	439921	128625	438976
86	26	221	125	216	26672	8720	314465	35937	314432	97247	31775	1114175	131072	1114047
171	51	391	256	375	28170	8790	103429	39304	102885	98021	28721	196571	163296	171875
243	75	775	343	768	29358	8610	59245	48013	52728	98825	32525	1757651	132651	1757600
524	164	2009	729	2000	31160	9320	72929	46305	70304	102476	31076	271001	148176	265625
594	174	1189	1000	1029	31437	10185	255595	42592	255507	104585	32045	307139	148955	303264
605	185	1739	864	1715	31841	10421	396611	42875	396576	105066	30966	219541	164616	203125
965	305	4331	1331	4320	33561	10461	121411	46875	120736	110619	35139	544999	151959	544000
1463	455	5135	2048	5103	33885	9945	68731	54880	61731	110942	36530	2047085	148877	2047032
1602	510	8245	2197	8232	34047	10335	90895	49152	89167	114653	34265	266555	170723	256608
1790	530	3869	2744	3645	35684	10604	79289	54000	75449	114950	36650	610589	157464	609725
2471	791	14351	3375	14336	37638	12330	493885	50653	493848	116721	38181	1415011	157216	1414875
2628	780	5785	3993	5488	39542	12410	158045	54872	157437	124011	40851	2370871	166375	2370816
2889	909	12019	4000	11979	40871	13271	359471	55296	359375	126770	38990	388229	179685	384104
3608	1160	23345	4913	23328	41445	12465	101659	60835	98784	127688	37400	255425	216513	219488
3735	1095	7519	6144	6655	44099	14459	608039	59319	608000	135812	41420	379865	194672	373977
3962	1190	9605	5832	9317	51260	16820	740921	68921	740880	138068	45500	2731625	185193	2731568
4455	1335	10591	6591	10240	52025	16925	492179	70304	492075	138635	45395	1772999	186624	1772855
5027	1595	24215	6912	24167	52415	15455	109871	81920	101871	139139	40859	283319	224000	255879
5049	1629	36019	6859	36000	52767	16575	213775	73167	212992	140670	41490	295501	219501	274360
6620	1940	13289	10985	11664	52988	15860	124745	78608	120393	141372	42420	339865	208537	329232
6830	2210	53261	9261	53240	53618	16910	231845	74088	231173	142722	45630	805285	195112	804357
7398	2250	20125	10648	19773	54824	16184	115889	85169	108000	144245	43265	345611	212960	334611
7749	2289	16459	12000	15379	59157	19425	894475	79507	894432	150993	48165	810355	206763	809248
8021	2561	43931	10976	43875	64638	20370	275965	89373	275128	151317	47265	563755	210912	560947
8987	2915	76055	12167	76032	65043	21195	658615	87808	658503	153149	50489	3132059	205379	3132000
10269	3129	28459	14739	28000	65583	19215	131455	109503	114688	162459	47619	326599	268279	288000
11556	3756	105481	15625	105456	67779	20859	208999	96000	206839	163133	53465	2194955	219488	2194803
12015	3855	73759	16384	73695	67826	22286	1070741	91125	1070696	169290	55830	3574981	226981	3574920
12386	3806	37541	17576	37125	68255	20735	183599	98415	180224	171899	51779	427319	251559	416000
13076	3836	26441	21296	23625	70686	22386	328861	97336	328125	174339	51579	374599	268119	352000
14472	4440	43105	20577	42592	71631	21231	156031	109375	147456	174420	55740	974521	238521	973360
14573	4745	142715	19683	142688	73062	21450	148525	117912	133837	174638	55970	1043165	238328	1042173
15930	4710	34069	24565	31944	75060	22620	187369	109744	182505	175644	53844	518569	250000	512169
17153	5525	116675	23328	116603	76505	22685	167171	116640	158171	179192	52520	360065	296352	317057
18074	5894	189029	24389	189000	77303	25415	1271855	103823	1271808	180080	55760	597329	253265	592704
19214	5954	64349	27000	63869	80069	26129	864059	108000	863939	186527	61535	4063295	250047	4063232
20195	5915	40391	34391	34560	85140	25980	239449	121945	235824	189335	57095	475391	276480	463391
22095	7215	245791	29791	245760	86616	25416	175441	140625	157216	189675	59475	747799	263424	744775
22473	6765	55555	32928	54043	87624	28824	1500049	117649	1500000	189945	55965	396019	298144	365835
22572	6660	47545	35152	44217	90801	28101	299251	127776	296875	190359	62439	2688079	256000	2687919
22715	6755	50759	34295	48384	91034	28934	453509	125000	452709	193698	57630	434485	292008	414613
23579	7619	176039	32000	175959	91490	27230	205781	137781	196520					

Table 4: A list of all degree 4 polynomial pairs of the form given in Proposition 1 and Corollary 2 with $a, b, c, d, e > 0$ and $b < a < 200000$ and $d < e$, and $\gcd(a, b, c, d, e) = 1$.

This is a genus 0 curve and solutions can be parameterised as $\mathbf{a} = 2\alpha(\beta^2 - 2\beta - 1)$, $\mathbf{b} = -2\alpha(\beta^2 + 2\beta - 1)$ and $\mathbf{c} = 2\alpha(\beta^2 + 1)$ for $\alpha, \beta \in \mathbb{Q}$. From this recovering what a, b are in this context is straightforward which are $a = \alpha(\beta^2 - 2\beta + 3)$ and $b = \alpha(\beta^2 + 1)$. These expressions for a, b are not quite what is stated in the proposition since S and T (and hence the resulting polynomials f and g) are not monic. Currently the leading coefficient of these polynomials is $(a - 3b)/(b^2(a - b)^3) = \beta(\beta + 1)/(\alpha^4(\beta - 1)^3(\beta^2 + 1)^2)$. To make this monic we first apply the linear transform $x \mapsto x/(\beta(\beta + 1))$ and then multiply these polynomials through by $C = \alpha^4\beta^3(\beta - 1)^3(\beta + 1)^3(\beta^2 + 1)^2$. This makes these polynomials monic and, after doing all the algebra, the expressions for a, b, c, d, e materialise as stated in the proposition. □

Corollary 1. *Using the language of PTE solutions from Costello et. al. [14], we get an ideal PTE solution of size* 4 *of the form*

$$[0, a, a, c] =_3 [b, b, d, e].$$

Remark 5. The strategy laid out in the proof of the proposition can be generalised in order to obtain a complete parametrisation of all ideal PTE solutions of size 4 not just those with this specific shape.

Corollary 2. *Suppose we have an ideal PTE solution of the form* $[0, a, a, c] =_3 [b, b, d, e]$ *with* $a > b$ *and* $a, b, c, d, e > 0$. *Then we have*

$$3b < a < (2 + \sqrt{2})b.$$

Proof. By the parametrisation of such solutions given in Proposition 1, we have

$$\frac{a}{b} = \frac{\beta^2 - 2\beta + 3}{\beta^2 + 1},$$

for some $\beta \in \mathbb{Q}$. As a rational function, the right hand expression attains a global maximum at $\beta = 1 - \sqrt{2}$. Thus, after evaluation, we get $a/b < 2 + \sqrt{2}$ which proves the upper bound.

For the lower bound, suppose that $b < a \leq 3b$. Once again, substitute the parametric expressions for a and b. After solving the inequality, one deduces that $\alpha \geq 0$ and either $\beta < -1$ or $0 < \beta < 1$.

Recall that d and e can be written in terms of this parametrisation as $d = \alpha(\beta + 1)(\beta - 1)^3$ and $e = 4\alpha\beta$. If $\beta < -1$ then, since $\alpha \geq 0$, we must have $e = 4\alpha\beta < 0$. Similarly, if $0 < \beta < 1$ then $d = \alpha(\beta+1)(\beta-1)^3 < 0$. In either case, this contradicts to the positivity of d and e and thus proves the intended lower bound. □

As a consequence of the above proof, we must have $\alpha < 0$ and $-1 < \beta < 0$. Hence we can write $\beta = -p_0/q_0$ for some positive coprime integers p_0, q_0 with $p_0 < q_0$. Moreover, if the PTE solution is normalised in the sense that not only does it satisfy the condition given in the above Corollary but also a, b, c, d, e are integers such that $\gcd(a, b, c, d, e) = 1$, then we must have $\alpha = -q_0^4$. We note that this is a necessary condition to find such normalised solutions but is not sufficient. Substituting these in, one gets an integral parametrisation of such PTE solutions rather than a rational one.

Using this parameterisation with the help of the bounds given in Corollary 2, one can find concrete PTE solutions of this type. Table 4 lists all possible solutions of this type such that $0 < b < a < 200000$.

Alternative Strategy to Section 5.1. The search for degree 8 polynomials that was mentioned in §5.1 have almost all of the repeated factors are on one side of the polynomial pair. Inspired by these new PTE solutions mentioned above, we could attempt to alternative degree eight polynomials that factor into lots of linear factors but the repeated factors are balanced between each side of the pair. To do this, one works over the polynomial ring $Q(a_1, a_2, a)[x]$ and let $F(x) = (x^2 - a^2)^2$ and $G(x) = (x^2 - a_1^2)^2(x^2 - a_2^2)$. Applying the XGCD algorithm to F and G yields polynomials S and T of degrees 4 and 2 (resp.). Again, after evaluating the variables a_1, a_2 and a, S may be factored into at most quadratic factors. In particular, up to permutations, only one example has been found whereby the number of linear factors of $S \cdot T$ is 4. This occurs when $a_1 = 41/2$, $a_2 = 85/2$ and $a = 71/2$ and results in the the polynomials

$$f(x) = (x + 20)^2(x + 48)(x + 63)(x + 93)^2(x^2 + 111x - 70), \text{ and}$$
$$g(x) = x(x + 13)(x + 35)^2(x + 76)^2(x + 98)(x + 111),$$

which differ by $C = 701168832000$.

C List of Polynomial Pairs

Here we list all polynomal pairs PTE_i^n and XGCD_j^n that were used in Table 2 and Table 3 for computing the smoothness probabilities and presenting the resulting smooth twins (resp.).

$$\mathsf{PTE}_1^6 = \begin{cases} f(x) = x(x+3)(x+5)(x+11)(x+13)(x+16), \text{ and} \\ g(x) = (x+1)^2(x+8)^2(x+15)^2. \end{cases}$$

$$\mathsf{PTE}_2^6 = \begin{cases} f(x) = x(x+5)(x+6)(x+16)(x+17)(x+22), \text{ and} \\ g(x) = (x+1)(x+2)(x+10)(x+12)(x+20)(x+21). \end{cases}$$

$$\mathsf{PTE}_1^8 = \begin{cases} f(x) = x(x+4)(x+9)(x+23)(x+27)(x+41)(x+46)(x+50), \text{ and} \\ g(x) = (x+1)(x+2)(x+11)(x+20)(x+30)(x+39)(x+48)(x+49). \end{cases}$$

$$\mathsf{PTE}_2^8 = \begin{cases} f(x) = x(x+9)(x+10)(x+29)(x+38)(x+57)(x+58)(x+67), \text{ and} \\ g(x) = (x+2)(x+3)(x+18)(x+22)(x+45)(x+49)(x+64)(x+65). \end{cases}$$

$$\mathsf{PTE}_3^8 = \begin{cases} f(x) = x(x+14)(x+19)(x+43)(x+57)(x+81)(x+86)(x+100), \text{ and} \\ g(x) = (x+1)(x+9)(x+30)(x+32)(x+68)(x+70)(x+91)(x+99). \end{cases}$$

$$\mathsf{PTE}^{10} = \begin{cases} f(x) = x(x+12)(x+125)(x+213)(x+214)(x+412)(x+413)(x+501)(x+614)(x+626), \text{ and} \\ g(x) = (x+5)(x+6)(x+133)(x+182)(x+242)(x+384)(x+444)(x+493)(x+620)(x+621). \end{cases}$$

$$\mathsf{PTE}^{12} = \begin{cases} f(x) = x(x+11)(x+24)(x+65)(x+90)(x+129)(x+173)(x+212)(x+237)(x+278) \\ \qquad (x+291)(x+302), \text{ and} \\ g(x) = (x+3)(x+5)(x+30)(x+57)(x+104)(x+116)(x+186)(x+198)(x+245)(x+272) \\ \qquad (x+297)(x+299). \end{cases}$$

Now we list the pairs XGCD_j^n found in this work and give only a small sample of such polynomial pairs compared to the total number. This additionally includes an example that can be found from a degree 6 search which can be used to compare with the degree 6 PTE polynomials.

$$\mathsf{XGCD}^6 = \begin{cases} f(x) = x(x+1)(x+2)(x+4)(x+5)(x+6), \text{ and} \\ g(x) = (x+3)^2(x^2+6x+2)^2. \end{cases}$$

$$\mathsf{XGCD}_1^8 = \begin{cases} f(x) = (x+1)(x+3)(x+4)(x+6)(x^2+7x-2)(x^2+7x+4), \text{ and} \\ g(x) = x^2(x+2)^2(x+5)^2(x+7)^2. \end{cases}$$

$$\mathsf{XGCD}_2^8 = \begin{cases} f(x) = x(x+1)(x+3)(x+5)(x+7)(x+8)(x^2+8x-8), \text{ and} \\ g(x) = (x+2)^2(x+6)^2(x^2+8x-5)^2. \end{cases}$$

$$\mathsf{XGCD}_3^8 = \begin{cases} f(x) = x(x+7)(x^2+2x+5)(x^2+7x+20)(x^2+12x+40), \text{ and} \\ g(x) = (x+2)^4(x+5)^4. \end{cases}$$

$$\mathsf{XGCD}_4^8 = \begin{cases} f(x) = x(x+4)(x+7)^2(x+10)(x+14)(x^2+14x+9), \text{ and} \\ g(x) = (x+5)^2(x+9)^2(x^2+14x+4)^2. \end{cases}$$

$$\mathsf{XGCD}_5^8 = \begin{cases} f(x) = (x+1)(x+4)(x+9)(x+12)(x^2+13x-6)(x^2+13x+18), \text{ and} \\ g(x) = x^2(x+3)^2(x+10)^2(x+13)^2. \end{cases}$$

$$\mathsf{XGCD}_6^8 = \begin{cases} f(x) = (x+1)(x+4)(x+9)(x+10)(x+15)(x+18)(x^2+19x-12), \text{ and} \\ g(x) = x^2(x+6)^2(x+13)^2(x+19)^2. \end{cases}$$

$$\mathsf{XGCD}_7^8 = \begin{cases} f(x) = (x+2)(x+9)(x+18)(x+24)(x+33)(x+40)(x^2+42x-55), \text{ and} \\ g(x) = x^2(x+13)^2(x+29)^2(x+42)^2. \end{cases}$$

$$\mathsf{XGCD}_8^8 = \begin{cases} f(x) = x(x+7)(x+9)(x+38)(x+40)(x+47)(x^2+47x+622), \text{ and} \\ g(x) = (x+2)^2(x+19)^2(x+28)^2(x+45)^2. \end{cases}$$

$$\mathsf{XGCD}_9^8 = \begin{cases} f(x) = x(x+9)(x+10)(x+31)(x+34)(x+55)(x+56)(x+65), \text{ and} \\ g(x) = (x+20)^2(x+45)^2(x^2+65x+154)^2. \end{cases}$$

$$\mathsf{XGCD}_1^{10} = \begin{cases} f(x) = x(x+1)(x+3)(x+11)(x+13)(x+14)(x^2+11x+8)(x^2+17x+80), \text{ and} \\ g(x) = (x+6)^2(x+7)^2(x+8)^2(x^2+14x+5)^2. \end{cases}$$

$$\mathsf{XGCD}_2^{10} = \begin{cases} f(x) = (x+1)(x+4)(x+10)(x+12)(x+18)(x+21)(x^2+20x-9)(x^2+24x+35), \text{ and} \\ g(x) = x^2(x+3)^2(x+11)^2(x+19)^2(x+22)^2. \end{cases}$$

$$\mathsf{XGCD}_3^{10} = \begin{cases} f(x) = (x+1)(x+7)(x+8)(x+14)(x+15)(x+21)(x^2+19x-10)(x^2+25x+56), \text{ and} \\ g(x) = x^2(x+11)^2(x+22)^2(x^2+22x+77)^2. \end{cases}$$

$$\mathsf{XGCD}_4^{10} = \begin{cases} f(x) = (x+2)(x+18)(x+22)(x+36)(x+40)(x+56)(x^2+49x-60)(x^2+67x+462), \text{ and} \\ g(x) = x^2(x+12)^2(x+29)^2(x+46)^2(x+58)^2. \end{cases}$$

$$\mathsf{XGCD}_5^{10} = \begin{cases} f(x) = (x+6)(x+20)(x+22)(x+40)(x+42)(x+56)(x^2+57x-90)(x^2+67x+220), \text{ and} \\ g(x) = x^2(x+12)^2(x+31)^2(x+50)^2(x+62)^2. \end{cases}$$

$$\mathsf{XGCD}_1^{12} = \begin{cases} f(x) = (x+2)(x+3)(x+5)(x+6)(x^2+8x-1)(x^2+8x+2)(x^2+8x+4) \\ \qquad (x^2+8x+10), \text{ and} \\ g(x) = x^2(x+1)^2(x+7)^2(x+8)^2(x^2+8x+14)^2. \end{cases}$$

$$\mathsf{XGCD}_2^{12} = \begin{cases} f(x) = (x+1)(x+3)(x+4)(x+9)(x+10)(x+12)(x^2+13x-3)(x^2+13x+6) \\ \qquad (x^2+13x+45), \text{ and} \\ g(x) = x^2(x+6)^2(x+7)^2(x+13)^2(x^2+13x+21)^2. \end{cases}$$

$$\mathsf{XGCD}_3^{12} = \begin{cases} f(x) = x(x+3)(x+6)(x+8)(x+11)(x+14)(x^2+14x+9)(x^2+14x+15) \\ \qquad (x^2+14x+39), \text{ and} \\ g(x) = (x+2)^2(x+5)^2(x+9)^2(x+12)^2(x^2+14x+3)^2. \end{cases}$$

$$\mathsf{XGCD}_4^{12} = \begin{cases} f(x) = (x+1)(x+6)(x+7)(x+9)(x+10)(x+15)(x^2+16x-6)(x^2+16x+18) \\ \qquad (x^2+16x+84), \text{ and} \\ g(x) = x^2(x+3)^2(x+13)^2(x+16)^2(x^2+16x+78)^2. \end{cases}$$

$$\mathsf{XGCD}_5^{12} = \begin{cases} f(x) = (x+1)(x+3)(x+7)(x+10)(x+33)(x+36)(x+40)(x+42)(x^2+43x-24) \\ \qquad (x^2+43x+396), \text{ and} \\ g(x) = x^2(x+12)^2(x+31)^2(x+43)^2(x^2+43x+186)^2. \end{cases}$$

$$\mathsf{XGCD}_6^{12} = \begin{cases} f(x) = x(x+9)(x+20)(x+30)(x+59)(x+69)(x+80)(x+89)(x^2+89x+330) \\ \qquad (x^2+89x+2100), \text{ and} \\ g(x) = (x+14)^2(x+44)^2(x+45)^2(x+75)^2(x^2+89x+120)^2. \end{cases}$$

$$\mathsf{XGCD}_7^{12} = \begin{cases} f(x) = x(x+21)(x+60)(x+69)(x+71)(x+80)(x+119)(x+140)(x^2+140x-99) \\ \qquad (x^2+140x+2301), \text{ and} \\ g(x) = (x+20)^2(x+63)^2(x+77)^2(x+120)^2(x^2+140x-51)^2. \end{cases}$$

$$\mathsf{XGCD}_8^{12} = \begin{cases} f(x) = (x+4)(x+7)(x+22)(x+50)(x+56)(x+84)(x+99)(x+102)(x^2+75x-136) \\ \qquad (x^2+137x+3150), \text{ and} \\ g(x) = x^2(x+14)^2(x+39)^2(x+67)^2(x+92)^2(x+106)^2. \end{cases}$$

$$\mathsf{XGCD}_9^{12} = \begin{cases} f(x) = x(x+43)(x+52)(x+138)(x+147)(x+190)(x^2+97x+810)(x^2+190x+2856) \\ \qquad (x^2+283x+18480), \text{ and} \\ g(x) = (x+3)^2(x+28)^2(x+70)^2(x+120)^2(x+162)^2(x+187)^2. \end{cases}$$