Towards Optimally Small Smoothness Bounds for Cryptographic-Sized Smooth Twins and their 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 smooth twins. 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 using 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 use these polynomials to search for large smooth twins whose sum is prime. In particular, the smoothness bounds of the 384 and 512-bit twins 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 [31,11] as part of their on-going effort to standardise post-quantum cryptography [30]. 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 order, which permits efficient isogeny computations[1]. The *smoothness bound*, B, of $p^2 - 1$ or its smooth divisor is the dominant factor in the performance of these cryptosystems [4]. Hence finding parameters that minimise B is vitally important. Having

[1] A priori, this needs $\mathbb{F}_{p^4}$-rational points in order to make sense of the quadratic twist but all computations can be done over $\mathbb{F}_{p^2}$ using standard techniques [13, §3].

said this, making B as small as possible is not feasible. This is due to the existence of a theoretical bound for how small B can be [29]. 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 into the problem of finding smooth twins in the sense of the following definition.

Definition 1. *We call a pair of consecutive integers* $(r, r+1)$ B-smooth twins *if their product,* $r \cdot (r+1)$*, is* B*-smooth and drop* B *from this definition when it is polynomial in* $\log_2(r)$*. We refer to a* cryptographic-sized *smooth twin if it has at least* 240*-bits.*

Numerous applications arise from finding smooth twins including the computation of logarithms of integers [22] and the ABC conjecture [12]. In the context of this work, if their sum $p = 2r+1$ of a twin is a prime, then $p^2-1 = 4r(r+1)$ is smooth and suitable for isogeny-based applications. Much like for the primes p, there is a theoretical optimum for the smoothness bound of smooth twins. 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 smooth twins can be separated into two categories: constructive and probabilistic methods. The constructive methods [29,12] 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] search for twins of a fixed size and guarantee finding them up to some probability depending on B. Thus expecting to find B-smooth twins for an optimally small B is not realistic. Nevertheless, if B is not too small, one can find such twins when looking over a large search space.

Constructive methods. There are two known approaches that enumerate all or almost all B-smooth twins. One requires solving exponentially many Pell equations [29,22,8] with respect to B, and the other is a recursive algorithm referred to as CHM [12]. While these algorithms tackle the task of finding twins with an optimally small smoothness bound, they become computationally infeasible when finding twins which are at least 240-bits. The analysis in [7, §4.1] suggests that the optimally smallest smoothness bound for a 256-bit twin is around $B \approx 5000$ which is much too large even with current computing resources. 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.

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

Method	$\log_2(B)$ of smallest B for b-bit primes p			Where
	$b \approx 256$	$b \approx 384$	$b = 512$	
XGCD over $\mathbb{Z}$	22.7	—	—	[4]
Cyclotomic factors	18.9	24.4	—	[13,18]
PTE sieve	15.0	20.6	27.9	[14]
XGCD over $\mathbb{Q}$[x]	**15.4**	**19.7**	**24.3**	**this work**

Table 1: Best known smoothness bounds of $p^2 - 1$ for cryptographic-sized primes p.

an integer C and factorise nicely. Then one evaluates these polynomials at an integer, ℓ, with $f(\ell)$ and $g(\ell)$ divisible by C to generate smooth twins.

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 latter polynomials is the current state-of-the-art in terms of minimising the smoothness bound. See Table 1 for a summary of the best results using these pairs.

Contributions. In this work we revisit and generalise the probabilistic methods for finding smooth twins. In particular, we use polynomial pairs f, g that (once again) differ by an integer C and split mostly into linear factors with the exception of a few quadratic factors. At first glance, the introduction of the quadratic factors would decrease the smoothness probabilities in comparison to the completely split pairs from [14]. However, this is largely compensated by having more repeated factors in f and g. As a result, the smoothness probability that arise from these polynomial pairs are either better or comparable to that of the prior polynomial pairs of the same degree.

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

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

The next pair of degree 8 polynomials is 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 quadratic factor:

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

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 $\mathbb{Q}[x]$ but a more fruitful approach computes this over $\mathbb{Q}(a_1, \cdots, a_n)[x]$. This can be viewed as a precomputation and, after potentially solving some equations in the variables $a_1, \cdots, a_n$, results in a more fine-grained searching criterion.

We use these polynomials pairs to find b-bit smooth twins and primes p such that p^2-1 is smooth for $b \in \{256, 384, 512\}$. Table 1 summarises the best results. It shows that our polynomials result in a comparable smoothness bound when searching for 256-bit primes, and give significantly better smoothness bounds when searching for larger primes (with $b = 384, 512$). We emphasise that the primes found here only reduce the smoothness bound of $p^2 - 1$ and do not take into account any additional constraints on p that may be imposed by specific isogeny-based cryptosystems (and which may, in some cases, be incompatible with lower smoothness bounds). We discuss this further in Section 6.

Organisation. We begin in Section 2 by reviewing known techniques for finding such smooth twins. In Section 3 we describe existing results on smoothness probabilities. In Section 4 we describe the general framework of our method for finding smooth twins. In Section 5 we detail the concrete computations of these new polynomials. As part of these computations we find polynomial pairs of degrees 8, 10 and 12. Finally, in Section 6 we detail experimental results for finding smooth twins and primes p using these new polynomial pairs.

2 Probabilistic Methods for Finding Smooth Twins

We start by reviewing the probabilistic techniques to find smooth twins.

XGCD over the integers. The most natural approach to find smooth twins is to choose random B-smooth integers r until either $r-1$ or $r+1$ is B-smooth. A slightly better 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 use the extended Euclidean algorithm over the integers with inputs α and β. This gives two integers, s and t, such that $\alpha s + \beta t = 1$ with $|s| < |\beta/2|$ and $|t| < |\alpha/2|$. If s and t are B-smooth then one obtains the following B-smooth twins

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

The probability that $s \cdot t$ is B-smooth is much larger 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 some of the largest B-smooth twins are of the form $(x^2 - 1, x^2)$. He generalised this to find large smooth twins of the form

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

for small $n \in \mathbb{Z}$, and exploited the factorisation of $x^n - 1$ into cyclotomic polynomials. For instance, finding b-bit twins of the form $(x^6 - 1, x^6)$ requires searching for three $(b/6)$-bit smooth numbers and two $(b/3)$-bit smooth numbers. 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 reducing the smoothness bounds. 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

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

Once such polynomials are known, then searching for b-bit twins consists of sieving an interval of roughly (b/n)-bit integers and identifying the integers ℓ in this interval such that $(\ell + a_i)$ and $(\ell + b_j)$ are all smooth. Thus the evaluations $f(\ell)$ and $g(\ell)$ are smooth. A final check needs to be done to determine whether one gets smooth twins – that is whether $f(\ell)$ and $g(\ell)$ are divisible by C. If all of this holds, then we get a smooth twin of the form

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

These completely split polynomials increases the smoothness probability again. However, such polynomial pairs f, g are non-trivial to find when the degree $n \geq 4$. Fortunately, one can construct such polynomials using solutions to the Prouhet–Tarry–Escott (PTE) problem (see [9] for more background).

3 Smoothness Probabilities

We recall standard results on the distribution of smooth integers, and polynomial evaluations, in intervals. This allows us to compute *smoothness probabilities* in these settings. The exposition given here follows [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 expressed with 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 $1 \leq u \leq 2$, we have $\rho(u) = 1 - \ln(u)$. But, for $u \geq 2$, there is no known closed form for this function in terms of elementary functions. Despite this we can still evaluate this function using numerical techniques [32,24] 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 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 is a good approximation for concrete values of N and B. As a result, assuming that the 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 $f \in \mathbb{Z}[x]$, we count the number of *smooth evaluations* of f and define

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

Numerous works have studied this quantity $\Psi_f(N, B)$ and it can be argued that the smoothness probability of $f(m)$ is the product of the smoothness probabilities of $f_i(m)$ for each irreducible factor $f_i \mid f$. While this heuristic is proven for certain ranges of N and B [25], it does not apply for the ranges of cryptographic interest. However, our experiments (and those of [14]) suggest that these heuristics closely approximate their true values. So we formally restate the heuristic.

Heuristic 1 ([14]) *Suppose that a polynomial $f \in \mathbb{Z}[x]$ has distinct irreducible factors 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. In this work we are mostly interested in finding smooth evaluations of a polynomial $f \in \mathbb{Q}[x]$ rather than an integer-valued polynomial. One can still use the heuristic in order to compute these smoothness probabilities by writing

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

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 content (that is, the gcd of the coefficients) of f. We must

modify Heuristic 1 to include the probability that an evaluation $f(m) = \hat{f}(m)/C$ is an integerr, which 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. Computing this number of integers modulo $p_i^{e_i}$ can be done directly as long as each prime power $p_i^{e_i}$ is not too big. Then multiplying all of these together gives the desired number of residue classes. Furthermore dividing this number by C gives the associated probability.

We assume that the events "$\hat{f}(m)$ is smooth" and "$\hat{f}(m) \equiv 0 \mod C$" are mutually independent, so the probability that $f(m)$ is smooth is (heuristically) the product of the probability of these events. We note that this point was addressed in [14], but this second event was not accounted in their smoothness probabilities.

4 Smooth Twins using XGCD over Polynomial Rings

This section describes a generalisation of the techniques from Section 2. A core ingredient is the extended Euclidean (XGCD) algorithm over polynomial rings [28, Theorem 17.4]. In order to fix consistent notation, we briefly recall this.

Let $\Bbbk$ be a field. The XGCD algorithm takes as input $F, G \in \Bbbk[x]$ and finds the unique polynomials $S, T \in \Bbbk[x]$, with $\deg(S) < \deg(G)$ and $\deg(T) < \deg(F)$, satisfies the following polynomial Bézout identity

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

4.1 The General Strategy to find Smooth Twins

Choose two coprime polynomials, $F, G \in \mathbb{Z}[x]$, that split completely into mostly repeated linear factors. We assume for expositions sake that F and G are monic polynomials. Use the XGCD algorithm over $\mathbb{Q}[x]$ to find 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[2] and the leading coefficient of $G \cdot T$ is positive. Then the polynomials $-F \cdot S$ and $G \cdot T$ differ by one and thus can give smooth twins with polynomial evaluation. We adopt a PTE style search, since these are polynomials in $\mathbb{Q}[x]$, and lift these polynomials to $\mathbb{Z}[x]$. Namely define the polynomials $f(x) \coloneqq -C \cdot F(x) \cdot S(x)$ and $g(x) \coloneqq C \cdot G(x) \cdot T(x)$, where C is the smallest integer such that $f, g \in \mathbb{Z}[x]$. These polynomials now differ by C, so one searches for integers ℓ such that $f(\ell)$ and $g(\ell)$ are smooth and $f(\ell) \equiv g(\ell) \equiv 0 \mod C$. This ensures that $f(\ell)$ and $g(\ell)$ are divisible by C and thus gives smooth twins. The formal procedure is given below:

[2] If $S(x) = (x - \alpha)S'(x)$ then $F(x)(x - \alpha)S'(x) + G(x)T(x) = 1$. So XGCD of $F(x)$ and $G(x)$ results in the same polynomial pair as the XGCD of $F(x)(x-\alpha)$ and $G(x)$.

1. Choose polynomials $F, G \in \mathbb{Z}[x]$ and apply the XGCD algorithm to get polynomials $S, T \in \mathbb{Q}[x]$. Let C be the smallest integer such that $CS(x)$, $CT(x) \in \mathbb{Z}[x]$ and set $(f(x), g(x)) \coloneqq (-CF(x)S(x), CG(x)T(x))$.
2. Let I be an interval of integers and use the sieve of Eratosthenes, as described in [14, §4.1] (see also [15, §3.2.5]), to identify the B-smooth integers $\ell \in I$.
3. Use the sieve established in Step 2 to identify the integers ℓ 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 found in Step 3 for which $f(\ell) \equiv g(\ell) \equiv 0 \mod C$.
5. Using one of the following techniques, determine when $CS(\ell)$ and $CT(\ell)$, or equivalently $Q(\ell)$ for all irreducible factors $Q(x) \mid C^2S(x)T(x)$, are B-smooth for the leftover integers ℓ from Step 4:
   (a) Factorise the B-smooth part of $Q(\ell)$ directly using either trial division or fast factoring methods such as ECM [26].
   (b) Use an Eratosthenes–style sieve that sieves the list of evaluations of each irreducible factor [15, §3.2.7]. This can be done in parallel with the sieving in Step 2.
   (c) Collate all evaluations $m = Q(\ell)$ for each irreducible factor Q into a list and apply Bernstein's sieving algorithm [3].
6. The remaining integers ℓ give smooth twins of the form $(f(\ell)/C, g(\ell)/C)$.

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

Realising the generalisation. One can view this method as a generalisation of the probabilistic methods described in Secton 2. To obtain the polynomial pair $x^n - 1, x^n$ using this approach, one simply computes the XGCD of $F(x) = x - 1$ and $G(x) = x^n$. The polynomial pair that result from a PTE solution 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.

Precomputed Polynomials. The simplest way to choose the input polynomials F and G is to take them at random by choosing integers $a, b \in \mathbb{Z}$, with $a \neq b$ to ensure coprimality and exponents $e_a, e_b \in \mathbb{Z}$ at random. Then construct the polynomials $F(x) = \prod(x+a)^{e_a}$ and $G(x) = \prod(x+b)^{f_b}$. A better approach is to have a precomputed list of polynomials F, G and choose one or many of them for the search. This precomputation has the advantage that one can maximise the smoothness probability with a good choice of polynomials F and G that result in the largest probability of finding smooth twins.

This precomputation can be done naïvely over $\mathbb{Q}[x]$. However, a slightly better approach is to initially work over a polynomial ring with coefficients in some rational function field before specialising to $\mathbb{Q}[x]$. Effectively the integers a, b that give the polynomials F and G become parametrised as variables over a 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 specialises 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 records their factorisation structure. This provides a more tailored search and significantly reduces the number of XGCD computations (since these are enumerable).

Seaching with one vs many pairs. Much like with the PTE sieve, one can search for smooth twins using either a single or many polynomial pairs. The search with a single pair is exactly as described above but one swaps the order of Step 3 and Step 4. This is particularly beneficial when the integer C is large since the proportion of integers with $f(\ell) \equiv g(\ell) \equiv 0 \mod C$ may be small. At a practical level, this was explored by Ahrens [1] in the context of the PTE sieve.

Alternatively, one can store a list of polynomial pairs and search for twins using all pairs simultaneously. To make this effective one employs a tree that fully traverses all polynomial pairs in a minimal number of checks. 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 the context of the general algorithm, this construction of the tree as well as the subsequent sieving is incorporated into Step 3.

Smoothness probabilities. Recall from §3.2 that the smoothness probability of an evaluated polynomial depends on the irreducible factors of the polynomial. So the probability of finding smooth twins depends on the irreducible factors of $F \cdot G \cdot S \cdot T$. This might suggest that a maximised smoothness probability would be obtained when these factors are all linear. This is certainly the case if the polynomial pair has repeated linear factors since fewer smoothness checks in Step 3 of the algorithm are needed. 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 repeated linear 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:

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

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

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

A more in-depth discussion on searching for such polynomials will be given in the next section detailing searches for degree 8,10 and 12 pairs.

Remark 1. In the setting of searching for twins whose 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

We detail strategies to find polynomial pairs $f, g \in \mathbb{Z}[x]$ that feature repeated factors and maximise the smoothness probability. As mentioned previously, this is only possible when their degree is not $2, 3, 4$ or 6. To incoporate repeated factors we search for pairs of the form $f(x) = h(x)^k - C$ and $g(x) = h(x)^k$ where $k, C \in \mathbb{Z}$, with $k > 1$, and $h \in \mathbb{Z}[x]$. Moreover, if f has a root a, then $C = h(a)^k$ is a k^{th} power and f factorises into cyclotomic polynomials composed with h. In addition we search for even polynomials in the sense that $f(x) = \tilde{f}(x^2)$ and $g(x) = \tilde{g}(x^2)$ for $\tilde{f}, \tilde{g} \in \mathbb{Z}[x]$. This will be combined with the following Lemma to give an effective search for these polynomial pairs. We note that this last idea has been used to find symmetric PTE solutions using interpolation techniques [5].

Lemma 1. *Let $F, G \in \Bbbk[x]$ be coprime polynomials and $S, T \in \Bbbk[x]$ be the result of applying XGCD to F,G. For $n \in \mathbb{Z}_{>0}$, set $F_n(x) \coloneqq F(x^n)$ and $G_n(x) \coloneqq G(x^n)$. If XGCD of F_n,G_n gives S_n, T_n, then $S_n(x) = S(x^n)$ and $T_n(x) = T(x^n)$.*

Proof. By the coprimality of F and G and replacing x by x^n we get

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

Writing $S'(x) \coloneqq S(x^n)$ and $T'(x) \coloneqq T(x^n)$, 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 must have $S_n \equiv S'$ and $T_n \equiv T'$. □

5.1 General Search Strategies

We apply the XGCD algorithm over $\Bbbk[x]$, with $\Bbbk \coloneqq \mathbb{Q}(a_1, \cdots, a_m, a)$, to $F(x) \coloneqq x^2 - a^2$ and $G(x) \coloneqq h(x)^k$ where $h \in \Bbbk[x]$ is of the form

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

With the addition of Lemma 1, this gives two polynomials S and T with $\deg(S) = \deg(h) \cdot k - 2$ and $\deg(T) = 0$. Write $T(x) = 1/C$ for some $C \in \Bbbk$. With this the desired polynomial pair can be recovered as follows: $g(x) \coloneqq C \cdot (G(x) \cdot T(x)) = h(x)^k$ and $f(x) \coloneqq C \cdot (-F(x) \cdot S(x)) = C \cdot (G(x) \cdot T(x) - 1) = h(x)^k - C$.

We now enumerate all positive coprime integers $a_1, \cdots, a_m, a \leq \kappa$ and evaluate the irreducible factors of S at these integers to give polynomials in $\mathbb{Q}[x]$. Factorise these polynomials and record the pair f, g if it has lots of linear factors and some quadratic factors. Finally, for sake of cleaning the polynomials, we apply a linear shift $x \mapsto x + A$ to the polynomials f, g so that the polynomials each linear factor of $f \cdot g$ is of the form $(x + \alpha)$ for $\alpha \geq 0$ and $x \mid f \cdot g$.

Remark 2. When $\deg(h)$ is even and the integer product $(\prod a_i) \cdot a$ is odd, applying this linear shift gives even α's. Thus we can consider half-integers $a_1, \cdots, a_m, a$ in this circumstance. This cannot be done when $(\prod a_i) \cdot a$ is even.

Example 1. For the sake of illustration we see this in action through a search of degree 8 polynomial pairs. Let $\Bbbk = \mathbb{Q}(a_1, a_2, a)$, $k = 2$ and $h(x) = (x^2 - a_1^2)(x^2 - a_2^2) \in \Bbbk[x]$. Applying XGCD to $F(x) = x^2 - a^2$ and $G(x) = h(x)^2$ gives

$$S(x) = -\frac{1}{C}\Big(x^2 - (a_1^2 + a_2^2 - a^2)\Big)\Big(x^4 - (a_1^2 + a_2^2)x^2 + a_1^2 a_2^2 + \sqrt{C}\Big)$$

and $T(x) = 1/C$, where $C = ((a_1^2 - a^2)(a_2^2 - a^2))^2$. The quadratic and quartic factors in $S(x)$ are irreducible in $\Bbbk[x]$. For some rational choices for a_1, a_2 and a, these factors may be reducible over $\mathbb{Q}[x]$. After enumerating, one encounters $a_1 = 19/2$, $a_2 = 7/2$ and $a = 1/2$ where S factors into four linear factors and one irreducible quadratic. Computing f and g as described above with a linear shift $x \mapsto x + 19/2$ gives

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

These are exactly the polynomials mentioned in Equation (1).

As $\deg(h)$ increases, so does $\deg(S)$ and its irreducible factors. So the likeliness that S has at most quadratic factors after variable evaluation decreases. One can alleviate this slightly using the following modification. One replaces F, G described above, with $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = h(x)^k$, giving polynomials with $\deg(S) = \deg(h) \cdot k - 2$ and $\deg(T) = 2$. For certain $a_1, \ldots, a_m, a, b$ we have $\deg(S) = \deg(h) \cdot k - 4$ and $\deg(T) = 0$. This happens when the leading coefficient of T is 0. We do not need to worry about the x coefficient since it is already 0 by Lemma 1. This gives us a relation between the variables of the function field: $a_1, \ldots, a_m, a, b$. Isolating one of the variables in this relation and replacing it in F and G ensures that $\deg(S) = \deg(h) \cdot k - 4$ and $\deg(T) = 0$. To make sure isolating a variable is possible, we take h to be 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. This reduction in $\deg(S)$ reduces the cost of factoring. Also after evaluation h may not split completely, so some new polynomial pairs can be found compared to the inital approach.

Method	n	$(f(x), g(x))$	$\lceil \log_2(C) \rceil$	(m_1, m_2)	$\log_2\left(\text{Smoothness Probability}\right)$		
					$b = 256$	$b = 384$	$b = 512$
Cyclotomic factors [13]	6	$(x^6 - 1, x^6)$	0	$(3, 2)$	-43.0	-49.8	-53.9
	8	$(x^8 - 1, x^8)$	0	$(3, 1)^*$	-44.9	-51.8	-55.9
	10	$(x^{10} - 1, x^{10})$	0	$(3, 0)^*$	-44.8	-51.4	-55.6
	12	$(x^{12} - 1, x^{12})$	0	$(3, 3)^*$	-31.7	-37.1	-40.3
PTE sieve [14]	6	PTE_1^6	14	$(9, 0)$	-42.8	-48.9	-52.6
		PTE_2^6	17	$(12, 0)$	-55.3	-63.0	-67.7
	8	PTE_1^8	31	$(16, 0)$	-47.4	-52.8	-56.0
		PTE_2^8	35	$(16, 0)$	-50.1	-55.1	-58.1
		PTE_3^8	38	$(16, 0)$	-52.5	-57.2	-60.0
	10	PTE^{10}	73	$(20, 0)$	-57.5	-59.2	-60.3
	12	PTE^{12}	76	$(24, 0)$	-44.7	-45.9	-46.7

Method	n	$(f(x), g(x))$	$\lceil \log_2(C) \rceil$	(m_1, m_2)	$\log_2\left(\text{Smoothness Probability}\right)$		
					$b = 256$	$b = 384$	$b = 512$
XGCD over $\mathbb{Q}[x]$	6	XGCD_1^6	6	$(7, 1)$	-45.2	-52.5	-56.8
	8	XGCD_1^8	10	$(8, 2)$	-38.6	-44.8	-48.5
		XGCD_2^8	12	$(8, 2)$	-39.6	-45.6	-49.2
		XGCD_3^8	14	$(4, 3)$	-41.6	-47.2	-50.5
		XGCD_4^8	15	$(7, 2)$	-39.7	-45.2	-48.4
		XGCD_5^8	16	$(8, 2)$	-41.4	-47.2	-50.6
		XGCD_6^8	21	$(10, 1)$	-38.3	-43.4	-46.5
		XGCD_7^8	30	$(10, 1)$	-43.6	-48.1	-50.8
	10	XGCD_1^{10}	22	$(9, 3)$	-36.0	-40.8	-43.7
		XGCD_2^{10}	26	$(11, 2)$	-34.1	-38.3	-40.9
		XGCD_3^{10}	28	$(9, 3)$	-38.5	-43.0	-45.7
		XGCD_4^{10}	41	$(11, 2)$	-39.4	-42.9	-45.0
	12	XGCD_1^{12}	14	$(8, 5)$	-31.4	-36.3	-39.3
		$\mathsf{XGCD}_{2,3}^{12}$	24	$(10, 4)$	-31.0	-35.2	-37.7
		XGCD_4^{12}	29	$(10, 4)$	-32.6	-36.5	-38.9
		XGCD_5^{12}	42	$(12, 3)$	-34.3	-37.3	-39.1
		XGCD_6^{12}	56	$(12, 3)$	-38.8	-41.1	-42.6
		XGCD_7^{12}	59	$(12, 3)$	-42.2	-44.4	-45.8
		XGCD_8^{12}	60	$(14, 2)$	-37.6	-39.7	-40.9

Table 2: Smoothness probabilities for finding primes p with $p^2 - 1$ being $2^{16}, 2^{22}$ and 2^{28}-smooth (resp.). An asterisk, $(m_1, m_2)^*$, is marked when $m_i > 0$ for some $i \geq 3$. The probabilities shaded in grey means that the search space is too small to expect to find such primes. See Appendix C of the auxilary material for all polynomial pairs.

Can one go further and choose $F(x) = (x^2 - a^2)(x^2 - b^2)(x^2 - c^2)$? The challenge is being able to solve the respective equation to ensure that $\deg(T) = 0$ which is non-trivial when F has this form. Solving the equation when $F(x) = (x^2 - a^2)(x^2 - b^2)$ is quite a bit easier, so we focus on this.

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

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

where d_i are the degrees of each irreducible factor of $f \cdot g$, $u = \log(X)/\log(B)$ and X is an $(b + \log_2(C))/\deg(f)$-bit integer. The table also include some probabilities from polynomials found in prior work.

Definition 2. *For a pair $f, g \in \mathbb{Z}[x]$, we denote m_k by the number of irreducible degree k factors in $f \cdot g$. For instance, the pair in Equation 1 has $m_1 = 10$ and $m_2 = 1$; and in Equation 2 it has $m_1 = 14$ and $m_2 = 2$.*

5.2 Degree 8 Polynomials

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

***k* = 2.** Let $h(x) = (x^2 - a_1)(x^2 - a_2^2)$ and apply the XGCD algorithm to the polynomials $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = h(x)^2$ to get polynomials, S and T, with $\deg(S) = 6$ and $\deg(T) = 2$. The leading coefficient of S and T is

$$\frac{(a_1 + a_2^2 - a^2 - b^2) \cdot (2a_1a_2^2 - a_1a^2 - a_1b^2 - a_2^2a^2 - a_2^2b^2 + a^4 + b^4)}{(a_1 - a^2)^2(a_1 - b^2)^2(a_2^2 - a^2)^2(a_2^2 - b^2)^2}.$$

When this leading coefficient is 0, we get either

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

After replacing a_1 with these expressions we get $\deg(S) = 4$ and $\deg(T) = 0$. In the first case, we have $T(x) = 1/((a_2^2 - a^2)(a_2^2 - b^2))^2$ and the polynomial S is irreducible over $\mathbb{Q}(a_1, a_2, a, b)[x]$ that can be explicitly computed as

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

Whereas in the second case we have $T(x) \equiv 1/C$, the polynomial S splits into quadratic factors which, with aid of the expression for a_1, is

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

and $C = \big((a_2^2 - a^2)(a_2^2 - b^2)(a^2 - b^2)/(2a_2^2 - a^2 - b^2)\big)^2$. The reason why it splits is due to where $x^2 - a^2$ and $x^2 - b^2$ is in the factorisation of f. Write $f(x) = h(x)^2 - C = (h(x) - \sqrt{C})(h(x) + \sqrt{C})$ and note that $(h(x) \pm \sqrt{C})$ are both even polynomials. In the first case $(x^2 - a^2)(x^2 - b^2)$ is equal to either $(h(x) \pm \sqrt{C})$ (up to the constant factor) and in the second case $(x^2 - a^2)$ divides one of $(h(x) \pm \sqrt{C})$ and $(x^2 - b^2)$ divides the other – thus explaining why S factors.

We assume that $a_1 = (a_2^2(a^2 + b^2) - a^4 - b^4)/(2a_2^2 - a^2 - b^2)$ for the rest of this exploration. For certain $a_2, a, b \in \mathbb{Q}$, these quadratic factors might be reducible. If $a_1, a_1 + a_2^2 - a^2, a_1 + a_2^2 - b^2$ are all squares, then the polynomial pair splits completely with $m_1 = 12$ – giving a PTE solution. As mentioned earlier there are no known ideal PTE solutions of this type in the literature. Our experiments did not find 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 $m_1 = 10$ and $m_2 = 1$. Plenty of polynomial pairs of this type can be found. The pair mentioned in Equation (1) would be found when $a_2 = 7/2$, $a = 1/2$ and $b = 11/2$ and is the example which features the smallest integer difference with $C = 1166400$. The next smallest can be found when $a_2 = 8$, $a = 3$ and $b = 12$ and, after applying the linear shift $x \mapsto x + 24$, results in the pair

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

which differ by $C = 564537600$.

We can relax this condition even further and only require one of these integers to be a square. This gives polynomial pairs with $m_1 = 8$ and $m_2 = 2$. While the smoothness probability could decrease by increasing m_2, one can find instances with a smaller C. Hence the smoothness probabilities are comparable to the prior pairs. In particular when $a_2 = 3/2$, $a = 1/2$, $b = 5/2$, one gets the pair

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

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 smooth twins from this pair.

Additionally, one example was found with either $a, b = 0$. This reduces m_1 by 1 and increases the smoothness probability compared to other pairs with $m_2 = 2$. This occurs when $a_2 = 2$, $a = 0$, and $b = 3$, resulting in the pair

$$f(x) = x(x+4)(x+7)^2(x+10)(x+14)(x^2+14x+9), \text{ and}$$
$$g(x) = (x+5)^2(x+9)^2(x^2+14x+4)^2, \text{ with } C = 32400.$$

$k = 4$. Now let $h(x) = (x^2 - a_1)$ and apply XGCD to $F(x) = (x^2-a^2)(x^2-b^2)$ and $G(x) = h(x)^4$. This gives polynomials, S and T, with $\deg(S) = 4$ and $\deg(T) = 0$ 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 cannot happen by the coprimality of F and G. So $a_1 = (a^2+b^2)/2$ and gives

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

If it can be factored, then it will be either $(x^2-c)(x^2-d)$ or $(x^2-cx+d)(x^2+cx+d)$ (up to the constant factor). By looking at the discriminant of the associated quadratic the first case cannot happen, so suppose the latter. 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 evaluated polynomial $S \in \mathbb{Q}[x]$ will always be irreducible.

Instead we revert back to $F(x) = (x^2 - a^2)$ and $G(x) = (x^2 - a_1^2)^4$. This is a special case of the computation from Example 1 with $a_2 = a_1$ and gives

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

When the quartic factors into a product of quadratics it gives pairs with $m_1 = 4$ and $m_2 = 3$. This occurs when $a_1 = 3/2$ and $a = 7/2$.

5.3 Degree 10 Polynomials

Now we detail the search for degree 10 polynomial pairs.

$k = 2$. Let $h(x) = x(x^2 - a_1)(x^2 - a_2^2)$ and apply the XGCD algorithm to the polynomials $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = h(x)^2$ to get polynomials, S and T, with $\deg(S) = 8$ and $\deg(T) = 2$. The leading coefficient of S is 0 when

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

With this choice of a_1 we reduce $\deg(S) = 6$ and $\deg(T) = 0$. For simplicity, we choose a sign for this expression. With this we get $T(x) = 1/C$ and

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

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

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

The cubic polynomials in the factorisation of S are irreducible over $\mathbb{Q}(a_2, a, b)[x]$. However they might be reducible after evaluating the variables a_2, a, b. Note that if one of them has a root α then the other polynomial must have a root $-\alpha$.

When the cubics are reducible, the resulting polynomial pair has, at least, $m_1 = 9$ and $m_2 = 3$. This occurs when $a_2 = 1$, $a = 4$ and $b = 6$ giving

$$\begin{aligned} f(x) &= x(x+1)(x+3)(x+11)(x+13)(x+14)(x^2+11x+38) \\ &\qquad (x^2+17x+80), \text{ and} \\ g(x) &= (x+6)^2(x+7)^2(x+8)^2(x^2+14x+5)^2, \text{ with } C = 2822400. \end{aligned}$$

One can ask when these quadratic factors reduce to linear factors. The case when all quadratic factors can be factored gives a PTE solution of size 10. However, only one such solution exists in the literature which does not have repeated factors. Additionally, when one (and hence both) of the quadratic factors in S is reducible, it gives a PTE solution of size 5 that has repeated factors. Again such solutions do not exist. The final case is the setting when a_1 is a square. This can happen and does so on many occasions, giving pairs with $m_1 = 11$ and $m_2 = 2$. When $a_2 = 8$, $a = 1$ and $b = 7$, we get such a pair and is

$$\begin{aligned} f(x) &= (x+1)(x+4)(x+10)(x+12)(x+18)(x+21)(x^2+20x-9) \\ &\qquad (x^2+24x+35), \text{ and} \\ g(x) &= x^2(x+3)^2(x+11)^2(x+19)^2(x+22)^2, \text{ with } C = 57153600. \end{aligned} \tag{3}$$

$k = 5$. We make a few brief comments on the other setting with $k = 5$ and choose $F(x) = (x^2 - a^2)$ and $G(x) = (x^2 - a_1^2)^5$. XGCD gives polynomials with $\deg(S) = 8$ and $\deg(T) = 0$. S is irreducible over $\mathbb{Q}(a_1, a)[x]$ and, after evaluating the variables a_1, a, the best one can hope for is that S factors into two irreducible quartic factors. However, we did not find any examples of this type.

5.4 Degree 12 Polynomials

We finally detail the search for degree 12 polynomial pairs. The exposition given here is not thorough given the computations are more involved. We refer to Appendix B of the auxilary material for more details.

$k = 2$. Apply XGCD to $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = h(x)^2$ with $h(x) = (x^2 - a_1)(x^2 - a_2^2)(x^2 - a_3^2)$, giving polynomials with $\deg(S) = 10$ and $\deg(T) = 2$. Through similar computations as before, there are two expressions for a_1 in terms of a_2, a_3, a and b that reduce $\deg(S) = 8$ and $\deg(T) = 0$.

If S factors into a product of quadratic factors after evaluation then the resulting pair has, at the very least, $m_1 = 8$ and $m_2 = 5$. Only four pairs were found with $m_2 = 3$. The one with the smallest integer C results in the pair

$$\begin{aligned} f(x) &= (x+1)(x+3)(x+7)(x+10)(x+33)(x+36)(x+40)(x+42) \\ &\quad (x^2 + 43x - 24)(x^2 + 43x + 396), \text{ and} \\ g(x) &= x^2(x+12)^2(x+31)^2(x+43)^2(x^2 + 43x + 186)^2. \end{aligned} \tag{4}$$

Additionally, one pair, mentioned in Equation (2), was found with $m_2 = 2$.

$k = 3, 6$. Applying XGCD to $F(x) = (x^2 - a^2)$ and $G(x) = h(x)^3$ with $h(x) = (x^2 - a_1^2)(x^2 - a_2^2)$ (and the special case with $a_1 = a_2$) cannot give pairs with $m_i = 0$ for all $i \geq 3$. Lots of pairs can be found with $m_4 > 0$ but this gives an extremely small smoothness probability.

$k = 4$. For this final choice, one applies XGCD to $F(x) = (x^2 - a^2)$ and $G(x) = h(x)^4$ with $h(x) = x(x^2 - a_1^2)$. Similar to the previous case, the best one can expect after evaluating the variables is getting pairs with $m_4 > 0$. However, loosening h to $h(x) = x(x^2 - a_1)$ can lead to polynomial pairs with $m_i = 0$ for $i \geq 3$. With $m_1 = 3$ and $m_2 = 6$ for these pairs, they do not result in better smoothness probabilities compared to the $k = 2$ pairs.

Remark 3. This strategy can be applied to find larger even degree pairs. For instance, to find degree 14 polynomials, one computes the XGCD of $F(x) = (x^2 - a^2)(x^2 - b^2)$ and $G(x) = x^2(x^2 - a_1)^2(x^2 - a_2^2)^2(x^2 - a_3^2)^2$. Then find algebraic expressions for a_1 to reduce $\deg(S) = 10$ and $\deg(T) = 0$ and factor S. Such polynomials could be useful when finding much larger twins.

6 Experimental Results: Concrete Smooth Sandwiches

We conducted experiments to find 256, 384 and 512-bit primes p (giving 128, 192 and 256-bits of security) such that $p^2 - 1$ is smooth using these new polynomials and the strategy from §4.1. We use Table 2 to choose appropriate smoothness bounds for these searches. Table **??** records the results from these experiments.

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_{240}]	240	(54503, 16)
	[14, p_{241}]	241	(32039, 15)
	[14, p_{243}]	243	(56711, 16)
	[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 $\mathbb{Q}[x]$	10622157951	XGCD_7^8	239	(69833, 17)
	2944237003	XGCD_2^8	241	(103001, 17)
	4187092101	XGCD_5^8	242	(95441, 17)
	13024987664	XGCD_6^8	250	(77029, 17)
	19371175757	XGCD_0^8	255	(77687, 17)
	49076211087	XGCD_5^8	257	(88897, 17)
	38295031104	XGCD_6^8	263	(42577, 16)
	1473704676325530	XGCD_0^8	370	(1723177, 21)
	670305535922892	XGCD_6^8	375	(826069, 20)
	1543959040318783	XGCD_5^8	376	(1742497, 21)
	479638273270508	XGCD_4^8	377	(1137329, 21)
	567277683164610	XGCD_3^8	378	(1812263, 21)
	2054426379410766	XGCD_2^8	379	(1502581, 21)
	647738699898325	XGCD_4^8	380	(1640941, 21)
	1213633306317077	XGCD_6^8	382	(1445533, 21)
	491954219730809	XGCD_1^8	383	(1749091, 21)
	1471680421245912	XGCD_6^8	384	(1140157, 21)
	2062439636622939	XGCD_6^8	388	(1733527, 21)
	1290853259901	XGCD_1^{10}	378	(1766099, 21)
	12538742222491880	XGCD_2^{10}	511	(34102657, 26)
	64343906330928	XGCD_5^{12}	510	(42485491, 26)
	188327931771336	XGCD_8^{12}	511	(24984383, 25)
	192093987758508	XGCD_8^{12}	512	(20003833, 25)

Table 3: Cryptographic-sized primes p such that $p^2 - 1$ is B-smooth. The full list of polynomial pairs can be found in Appendix C of the auxilary material.

We use the C implementation of the sieve of Eratothenes from the PTE Python3 code[3] as a starting point for our implementation. We include sieves with multiple polynomial pairs (for the degree 8 pairs) as well as sieves with a single pair (for the degree 10 and 12 pairs). We chose to implement Step 5a, to deal with the smooth evaluations of the quadratic factors and implemented this in Magma to benefit from fast factoring algorithms. This strategy is particular effective when $m_2 \leq 2$ and also the search using the degree 12 pair with $m_2 = 3$. The other pairs might benefit from using either Step 5b or Step 5c for this post-processing – this is left as an avenue for future work.

We ran these experiments on a server (featuring a total of 96 parallel threads) with a Xeon E7-4850v2 2.30GHz, 1007GB of RAM. The timings for the experiments depends on the interval size and m_2. It took 5-6 days to do the sieve of Eratothenes and took 30-60 minutes (around a day) to do the post processing with $m_2 = 1$ ($m_2 = 2$ resp.) when searching an interval of size 2^{48}.

***b* = 256.** Finding 256-bit primes with our polynomials imposes a reduced search space. For instance, using the degree 8 pairs, the interval $[2^{31}, 2^{37}]$ scans all 256-bit primes and the probability of finding 2^{16}-smooth twins is at most $2^{-38.3}$. This suggests that a search with this B would be unsuccessful. So we ran our code in this interval with $B = 2^{17}$. Despite the low probability, one prime was found with $p^2 - 1$ being 2^{16}-smooth which is

$$p = 2\left(\frac{\ell(\ell+6)(\ell+13)(\ell+19)}{1080}\right)^2 - 1, \text{ with } \ell = 38295031104.$$

[3] https://github.com/microsoft/twin-smooth-integers

In contrast, the PTE sieve [14] produced smaller smoothness bounds. There they used degree 6 pairs that split completely into linear factors some of which are repeated – giving a larger smoothness probablity to search space size ratio.

$b = 384$. When searching for larger primes, the search space limitations become less of an issue with our polynomials. The search with $B = 2^{22}$ using our degree 8 pairs found plenty of primes p with $p^2 - 1$ being 2^{22}-smooth. Many of these 2^{21}-smooth – surpassing the smoothness bounds found with the PTE sieve. One prime was found with $p^2 - 1$ being 2^{20}-smooth which is

$$p = 2\left(\frac{\ell(\ell+6)(\ell+13)(\ell+19)}{1080}\right)^2 - 1, \text{ with } \ell = 670305535922892.$$

We also searched with the degree 10 pairs with $B = 2^{22}$. This produced fewer primes compared to the degree 8 search but one was found that is 2^{21}-smooth.

$b = 512$. This final setting will give the most gain in reducing the smoothness bound. The searches with the PTE sieve found a few primes with $p^2 - 1$ being 2^{28}-smooth. The probabilities from Table 2 suggests the degree 10 and 12 pairs should plenty of primes with $B = 2^{28}$. So we used $B = 2^{26}$ for our searches. Among our degree 10 pairs, the best prime found from these searches is

$$p = 2\left(\frac{\ell(\ell+3)(\ell+11)(\ell+19)(\ell+22)}{7560}\right)^2 - 1, \text{ with } \ell = 14334163549504404.$$

We also searched with the degree 12 pairs with $m_2 \leq 3$. The pair from Equation (2) with $m_2 = 2$ gave two primes with 2^{25}-smooth $p^2 - 1$. The larger is

$$p = 2\left(\frac{\ell(\ell+14)(\ell+39)(\ell+67)(\ell+92)(\ell+106)}{791683200}\right)^2 - 1,$$

with $\ell = 192093987758508$.

Protocol specific polynomials. None of our primes have direct impact on an isogeny-based cryptosystem. They were initially proposed for B-SIDH [13] which was broken by the recent polynomial time attacks [10,23,27] on SIDH [21]. Our primes can be used in countermeasures to these attacks have been proposed [20,2] but much larger primes are required. So further experiments are needed.

The signature scheme SQIsign [19] requires primes p with a smooth divisor $2^f T \mid p^2 - 1$ with $T \approx p^{5/4}$ and f as large as possible. The smoothness bound of T and the size of f controls the performance of signing. Our primes do not have a large power of two so would not improve the state-of-the-art. However, given the general idea from §4.1 one does not have to use the polynomial pairs found in Section 5 and one could construct pairs that can guarantee a large power of two. This is the case for the pair $(x^n - 1, x^n)$ [7,11] since one can get a large power

of two from a small power of two in x which is boosted with the n^{th} power. The trouble is the degree of the largest factor of $x^n - 1$ grows with n which does not aid the smoothness probability. Alternatively, one could construct pairs with a smaller power of x that gives better smoothness probabilities.

As mentioned in the introduction, this work focuses on simply reducing this smoothness bound and seeing how small it could be. As a result we did not conduct these application ideas and are left as future work.

Concluding remarks. These experimental results shows that our polynomials scale better compared to prior polynomials – in the sense that, when searching for larger primes and twins, our polynomials produce smaller smoothness bounds. This is relevant in the context of isogeny-based cryptosystems using finite fields of much larger characteristic and wish to benefit from these sorts of primes.

Despite all of this work, we have not answered the underlying question of finding cryptographic sized smooth twins with an optimally small smoothness bound. Given the probabilistic nature of the method, one has to sacrifice the smoothness bound and not make it as small as possible. Improving these approaches is left open to the reader.

Acknowledgements. This work was supported by the HYPERFORM consortium funded by France through Bpifrance and l'Agence nationale de la recherche through a Plan France 2030 grant (ANR-22-PETQ-0008 PQ-TLS). The author thanks Maria Corte-Real Santos, Craig Costello, Jonathan Komada Eriksen and Robert Granger for valuable discussions that helped shape the work; Michael Meyer for his insights to the PTE experiments; Benjamin Smith for his assistance running the experiments and comments to drafts of the work; as well as the anonymous reviews for their constructive feedback.

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. In *ASIACRYPT*, volume 14445 of *Lecture Notes in Computer Science*, pages 208–233. Springer, 2023.
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. Meyer, M. Naehrig, and B. Sterner. Cryptographic smooth neighbors. In *ASIACRYPT*, volume 14444 of *Lecture Notes in Computer Science*, pages 190–221. Springer, 2023.
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. D. Robert. Breaking SIDH in polynomial time. In *EUROCRYPT*, volume 14008 of *Lecture Notes in Computer Science*, pages 472–503. Springer, 2023.
28. V. Shoup. *A Computational Introduction to Number Theory and Algebra*. Cambridge University Press, USA, 2 edition, 2009.
29. 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.
30. The National Institute of Standards and Technology (NIST). Submission requirements and evaluation criteria for the post-quantum cryptography standardization process, December, 2016.
31. The National Institute of Standards and Technology (NIST). Call for additional digital signature schemes for the post-quantum cryptography standardization process, October, 2022.
32. 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.

Auxilary Material

A Constructive Methods Summary

The methods presented here find all or almost all B-smooth twins for a fixed an integer B. It turns out that the set of B-smooth twins is finite for a fixed B. Thus it makes sense to 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 [29] (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 is computationally infeasible for large[4] B but is practical for small B. For instance, with $B = 5$ ($B = 7$ resp.) solving all $2^{\pi(B)}$ Pell equations gives 10 (23 resp.) B-smooth twins. 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.

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

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

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 finds either all or a majority of B-smooth twins. For instance, with $B = 5$ the algorithm finds all 5-smooth twins while with $B = 7$

[4] Recent work [8] modifies this approach for large B. Instead of collating all B-smooth twins for a smoothness bound B, they find B-smooth twins in a large interval.

the algorithm finds all 7-smooth twins 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. An additional 2,649 pairs of 200-smooth twins were found from this computation – proving the point that one does not find all smooth twins. The only way to know this exact number is to solve $2^{\pi(200)} = 2^{46}$ Pell equations which is beyond our current computing resources.

B Detailed computations for the degree 12 polynomials

Here we describe the concrete computations necessary for the degree 12 search as mentioned in §5.4. Recall that we start with $h(x) = (x^2-a_1)(x^2-a_2^2)(x^2-a_3^2)$ and apply XGCD to $F(x) = (x^2-a^2)(x^2-b^2)$ and $G(x) = h(x)^2$ giving polynomials with $\deg(S) = 10$ and $\deg(T) = 2$. When the leading coefficient of S and T is 0, then $\deg(S) = 8$ and $\deg(T) = 0$. This occurs when either

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

This choice of expression for a_1 results in different a factorisation structure for the polynomial S. For the first choice, the polynomial S splits as $S = S_2 \cdot S_6$ where S_2 is an irreducible quadratic and S_6 is an irreducible degree 6 polynomial over $\mathbb{Q}(a_2, a_3, a, b)[x]$. Moreover, the polynomial S as well as its irreducible factors are even polynomials and $T(x) = 1/C$ where

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

Now the question is when does S factorise into a product of at most quadratic factors after variable evaluation. Since S_2 is automatically quadratic, this only depends on the S_6. One can factor S_6 for each variable evaluation but it is more economical to only factor S_6 when the associated cubic S_3' where $S_6(x) = S_3'(x^2)$ has roots. This cubic can either be factored directly or use known root tests to determine if it has roots.

When $a_2 = 1/2$, $a_3 = 13/2$, $a = 5/2$ and $b = 7/2$ it gives polynomial pairs with $m_2 = 4$ which is

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

with $C = 10497600$. Additionally choosing $a_2 = 19/2$, $a_3 = 43/2$, $a = 23/2$ and $b = 29/2$ gives the polynomial pair mentioned in Equation (4) with $m_2 = 3$.

For the second choice, $S = S_4 \cdot S_4'$ splits into a product of two distinct irreducible quartics over $\mathbb{Q}(a_2, a_3, a, b)[x]$. Once again, each polynomial S, S_4, S_4' are even polynomials and $T(x) = 1/C$ where

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

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 §5.2 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 can be checked by doing some discriminant calculation, namely whether $A^2 - 4B$ is a square. In the second case, there are a few arithmetic checks needed: firstly B must be a square and then either $2\beta - A$ or $-2\beta - A$ must be a square.

When $a_2 = 3$, $a_3 = 4$, $a = 1$ and $b = 2$ it gives polynomial pairs with $m_2 = 5$ which is

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

with $C = 14400$. Additionally, choosing $a_2 = 14$, $a_3 = 39$, $a = 3$ and $b = 31$ gives the pair mentioned in Equation (2) with $m_2 = 2$.

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 **??** for computing the smoothness probabilities and presenting the resulting smooth twins (resp.). We first list some polynomial pairs that were found in [14].

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

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

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

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

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

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

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

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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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