Sieving for large twin smooth integers using single solutions to Prouhet-Tarry-Escott

Knud Ahrens

Faculty of Computer Science and Mathematics
University of Passau, Germany
knud.ahrens@uni-passau.de

1 Introduction

Cryptography based on isogenies of elliptic curves produced some candidates for post-quantum cryptography. One example from this field is the signature scheme SQISign [6]. It uses isogenies of degree $p+1$ and $p-1$, and to be able to evaluate them efficiently one needs both $p+1$ and $p-1$ to be smooth in which case p is called a twin smooth prime. A smooth integer only has small prime factors below a fixed smoothness bound. There are approaches to find such primes using the extended Euclidean algorithm or polynomials of the form $x^n - 1$, but in 2021 Costello, Meyer and Naehrig [5] found a new way using solutions to the Prouhet-Tarry-Escott (PTE) problem and produced integers with lower smoothness bounds. They used several PTE solutions at once and optimised for a setting with many solution. We will only use a single solution at a time and customise the sieve accordingly. This is advantageous when only a few suitable solutions are available.

2 Sieving with a PTE solution

In this section we shortly present the Prouhet-Tarry-Escott (PTE) problem and how [5] used it to find twin smooth integers.

2.1 The Prouhet-Tarry-Escott problem

The Prouhet-Tarry-Escott problem of degree[1] n and integer k is a set of equations $a_1^j + \cdots + a_n^j = b_1^j + \cdots + b_n^j$ with $\{a_1, \ldots, a_n\} \cap \{b_1, \ldots, b_n\} = \emptyset$ for $1 \leq j \leq k$ and $a_i, b_i \in \mathbb{Z}$. A solution $\mathcal{A} = \{a_1, \ldots, a_n\}$, $\mathcal{B} = \{b_1, \ldots, b_n\}$ with $k = n-1$ is called an ideal solution. There are several known solutions and even some parametric solutions, but for $n = 11$ and $n \geq 13$ no ideal solutions are known [4].

[1] Since we will mostly deal with ideal solutions and the polynomials in equation (1), we deviate from the usual notation of size n and degree k.

A statement from Borwein and Ingalls [2] using Newton's identities tells us that for each ideal solution the difference $C = |a(x) - b(x)|$ of the corresponding polynomials

$$a(x) = \prod_{i=1}^{n}(x - a_i), \quad b(x) = \prod_{i=1}^{n}(x - b_i) \tag{1}$$

is an integer constant. Since the polynomials already have n factors, chances are high that the evaluations are smooth. Ideal solutions with small differences C seem to produce a higher rate of twin smooth integers, but their number is limited (see [5, table 2]) since the C in parametric solutions often increases fast.

The paper [5, table 3] shows heuristically that this approach has a higher chance of success for a given smoothness bound or rather allows for a lower smoothness bound for a given probability to find twin smooth integers compared to previous attempts, e.g. using the extended Euclidean algorithm.[2]

2.2 The sieve

For ease of notation an ideal PTE solution will only be called a solution in the following. To check smoothness we basically use a sieve of Eratosthenes on the interval $I \subset \mathbb{Z}$ with different powers of p instead of different primes.

The smoothness of $a(\ell)$ and $b(\ell)$ is then tested using their factors. Let $\{a_1, \ldots, a_n, b_1, \ldots, b_n\}$ be the set of roots of $a(x)$ and $b(x)$. Then for each integer ℓ in the search interval I we look up the values $\{\ell - a_1, \ldots, \ell - a_n, \ell - b_1, \ldots, \ell - b_n\}$. If all of them are smooth, then surely their product is smooth and we have two smooth integers $a(\ell)$ and $b(\ell)$ that differ by C.

For those smooth integers we finally compute $a(\ell) \bmod C$ (or $b(\ell) \bmod C$) to see if $a(\ell)/C$ and $b(\ell)/C$ are integers. If they are, we have found twin smooth integers and can check if $a(\ell)/C + b(\ell)/C$ is a prime.

If there are multiple PTE solutions of the same degree n one may speed up the search as follows. All PTE solutions can be normalised in a way such that 0 is always in the zero set of the corresponding polynomials and from now on all solutions are taken to be normalised. Costello, Meyer and Naehrig [5] presented a tree structure based on a hitting set problem to test many solutions at the same time with the minimal number of look-ups in the list of smooth integers. For $n = 6$ there are 2438 solutions with $C \leq 2^{50}$ [5, table 2] and this tree approach is very efficient.

3 Optimising for single PTE solutions

Since different solutions have different C, the tree approach forbids to do the modulo calculation fist, but for a single solution this is possible. Costello, Meyer and Naehrig [5] left it as an open question to explore this path. In this section we will see why this is relevant and how it can be done efficiently. We will also discuss implementations and some experimental results.

[2] There has been a recent publication [3] by the same authors with a different approach.

3.1 Motivation

According to [5, table 3] a search for twin smooth primes in the range of 512 bits and beyond favours solutions of higher degree and there are only a few solutions with high degree n and small C. Therefore checking them individually is feasible. In the full paper we discuss how n and C influence the chance to find twin smooth integers.

The idea is to calculate the set R of all elements r in $[0, C)$ that solve $a(r) \equiv b(r) \equiv 0 \bmod C$. Obviously $a(r + C) \equiv a(r) \bmod C$ and thus all relevant $\ell \in I$ are of the form $\ell = r + kC$. Hence, instead of looking at all ℓ in the search interval I it is sufficient to look at $\{r + kC \mid r \in R,\, k \in \mathbb{Z}\} \cap I$. This subset has size $(|R|/C) \cdot |I|$ and can be significantly smaller than I. Another improvement is that the modulo computation only needs to be done once per PTE solution and not for every smooth integer $a(\ell)$ (or $b(\ell)$) anew.

3.2 Sieving is still optimal

When using only one solution, we have just seen that it suffices to check smoothness for $a(\ell), b(\ell)$ with $\ell \in \{r + kC \mid r \in R,\, k \in \mathbb{Z}\} \cap I$.

There is an algorithm by Daniel J. Bernstein [1] to test smoothness for an arbitrary set or we could use a polynomial sieve to check the set $f(I)$ instead of an interval I for any polynomial $f \in \mathbb{Z}[X]$.

In the full paper we discuss why the standard sieve or a logarithmic variant of it are still the best choices to check smoothness for this new approach. Since the SQISign protocol tolerates a large non-smooth factor, the modification to use rounded logarithms and to ignore the smallest primes or at least their small powers is a good option to speed up this part of the calculations.

3.3 Computing Congruences

When we have computed the smoothness for all elements in the interval, we need to find the set of relevant residues $R = \{r \in \mathbb{Z}/C\mathbb{Z} \mid a(r) \equiv b(r) \equiv 0\}$. Calculating $a(r) \bmod C$ (or $(b(r))$ for all $0 \leq r < C$ becomes too slow quite soon. Here the Chinese Remainder Theorem comes to the rescue, but the list can still get very long.

As soon as C and especially when $|R|$ becomes greater than $|I|$, the advantage over the naive approach disappears when we still consider all $r \in R$. This would be the case if we calculate the CRT on the fly. If we have an ordered list of residues, we can find the smallest representative in the interval (for example using nested intervals) and then take the next one until we leave the interval. This way we still only need to check about $|R|/C$ of the whole interval.

3.4 Experimental Results

In the full paper there are tables with the running times for the Sage and C implementations and we discuss them in more detail.

For an interval of fixed size the naive algorithm has similar running times for different solutions and seems to depend more on the number of smooth integers in the interval. The optimised algorithm depends heavily on the ratio of relevant residues $|R|/C$. It is always significantly faster than the naive approach and probably even faster than the tree approach for higher degrees n. Due to memory restrictions for larger C not all PTE solutions for $n \geq 7$ have been tested. For $n = 7$ we need less than 1/20 of the time and there are only 8 solutions with $C \leq 60$ [5, table 2]. So this approach is definitively faster. For $n \geq 9$ there are only one or two solutions with small C, so the tree is at most twice as fast as the naive algorithm and by the heuristic in section 4 the optimised approach is faster.

4 Roots mod p^k

In this section we will give a formula to compute the (maximal) number of roots of a given polynomial modulo prime powers. Let p be a prime and $f(x) = \prod_{i=1}^{n}(x - z_i)$ a polynomial with integer roots z_i and degree n.

The Taylor expansion of the polynomial f at x is given by

$$f(x + \delta) = \sum_{i=0}^{\infty} \frac{f^{(i)}(x)}{i!}\delta^i$$

where $f^{(i)}$ is the i^{th} derivative of f. Since $f^{(i)} = 0$ for $i > n$ we can truncate the infinite sum to a finite one.

4.1 For one root

First we will only look at one fixed root z of f and the integers that are congruent to z modulo p. Let $S = \{\sum_{i=1}^{\infty} d_i p^i \mid 0 \leq d_i < p,\, d_i \in \mathbb{Z}\}$, then $z + S$ is the set of all integers that are congruent to z modulo p. Since $f^{(0)}(z) = 0$ we get

$$f(z + s) \equiv \sum_{i=1}^{\min\{k-1,n\}} \frac{f^{(i)}(z)}{i!}s^i = \sum_{i=1}^{\min\{k-1,n\}} \frac{f^{(i)}(z)}{i!}\Big(\sum_{j=1}^{k-1} d_j p^j\Big)^i \mod p^k$$

with $s \in S$ since $\nu_p(s^i) \geq i$.

In this extended abstract we will only look at the worst case where all roots are equal, i.e. $f(x) = (x - z)^n$. We get $\nu_p(\frac{f^{(i)}(z)}{i!}) = \infty$ for $1 \leq i < n$ and the least valuation is $\nu_p(\frac{f^{(n)}(z)}{n!}(d_1 p^1)^n) = \nu_p((d_1 p^1)^n) = n$. In that case $f(z + s) \equiv (d_1 p^1)^n \bmod p^{n+1}$ for $s \in S$ and this is zero if and only if $d_1 = 0$. Therefore we get that all $z + s$ that solve $f(z + s) = 0 \bmod p^k$ have arbitrary d_1 for $k \leq n$ and $d_1 = 0$ for all $k > n$. Then the next smallest valuation for $d_1 = 0$ is $\nu_p((d_2 p^2)^n) = 2n$. Continuing in the same fashion we get that $d_i = 0$ for all solutions modulo p^k with $k > in$. This shows that there are at most

$$p^{k - \lceil \frac{k}{n} \rceil} \leq p^{\frac{n-1}{n}k} \tag{2}$$

solutions $z + s$ above z modulo p^k for $k \in \mathbb{N}$.

If not all roots are equal, then the restrictions on d_i appear already modulo smaller powers of p so the number of solutions will be smaller. Modulo small powers there can be more total solutions if not all z_i are equal, but for large powers equation (2) gives an upper bound. If the roots are pairwise distinct there is even a finite upper bound to the number of solutions.

4.2 Bounding $|R|$

Since $|R|$ is the number of solutions to $a(x) \equiv 0 \bmod C$ (or $b(x) \equiv 0$ equivalently) it is sufficient to find a bound on the number of solutions per prime power factor of C. Therefore the power of p is bounded by $p^k < C$ and we can bound the number of solutions per prime by

$$p^{\frac{n-1}{n} \log_p C} = C^{\frac{n-1}{n}}.$$

By [7] we know that C has $\log \log C$ different prime factors, at least asymptotically. So $|R|/C$ is bounded via Chinese Remainder Theorem by

$$\frac{|R|}{C} \leq \frac{C^{\frac{n-1}{n} \log \log(C)}}{C} = C^{\frac{n-1}{n} \log \log(C) - 1}.$$

In practice these values are significantly smaller and the ratio tends to decrease for larger C. For solutions with $n = 7$ we find that about half the prime power factors of C are square free and therefore admit at most $n = 7$ solutions in contrast to $C^{\frac{n-1}{n}} \geq 2^{30}$.

References

1. Bernstein, D.J.: How to find Smooth Parts of Integers (2004), https://cr.yp.to/factorization/smoothparts-20040510.pdf
2. Borwein, P., Ingalls, C.: The Prouhet-Tarry-Escott problem revisited. Enseign. Math. (2) **40**(1-2), 3–27 (1994)
3. Bruno, G., Santos, M.C.R., Costello, C., Eriksen, J.K., Naehrig, M., Meyer, M., Sterner, B.: Cryptographic smooth neighbors. Cryptology ePrint Archive, Paper 2022/1439 (2022), https://eprint.iacr.org/2022/1439, https://eprint.iacr.org/2022/1439
4. Caley, T.: The Prouhet-Tarry-Escott problem. PhD thesis, University of Waterloo (2012), https://uwspace.uwaterloo.ca/bitstream/handle/10012/7205/Caley_Timothy.pdf
5. Costello, C., Meyer, M., Naehrig, M.: Sieving for Twin Smooth Integers with Solutions to the Prouhet-Tarry-Escott Problem. In: Canteaut, A., Standaert, F.X. (eds.) Advances in Cryptology – EUROCRYPT 2021. pp. 272–301. Springer International Publishing, Cham (2021)
6. De Feo, L., Kohel, D., Leroux, A., Petit, C., Wesolowski, B.: SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies. In: Moriai, S., Wang, H. (eds.) Advances in Cryptology – ASIACRYPT 2020. pp. 64–93. Springer International Publishing, Cham (2020)
7. Hardy, G.H., Ramanujan, S.: The normal number of prime factors of a number n. Quart. J. **48**, 76–92 (1917)