Semi-prime factorization – Analysis of Fermat's Factorization method

By Essbee Vanhoutte

Version 7.2

Foreword

I am self-taught without formal math education as I dropped out of high-school. This represents 3 years of research and teaching myself mathematics.

Before this I was a self-taught software vulnerability researcher who briefly worked at companies such as Microsoft and Trend Micro and has well over 50+ CVEs in software such as Windows OS, OpenSsl and Adobe Reader attributed to their name (often under the handle SandboxEscaper or Polar Bear).

No AI was used for this research, I have only used it in the last couple of weeks to review my paper and get feedback and occasionally to produce code for well documented mathematical functions.

This paper mainly illustrates findings by example and companion code rather than rigorous mathematical notation and proofs. So perhaps rather than a formal academic paper, view this as a mathematical exploration by an enthusiast with prior experience in IT security.

To do: Add references and closing thoughts and clean everything up.
To do: Bunch of inconsistencies with signs on how I represent these quadratics on both sides

Chapter I – Fermat's Factorization method

In its most basic form the procedure for Fermat's Factorization method is as follows (where N is the semi-prime being factored).

1. Calculate $\lceil \sqrt{N} \rceil$
2. Starting from $x = \lceil \sqrt{N} \rceil$, calculate $f(x) = x^2 - N$ in a loop, incrementing x.
3. If $f(x)$ evaluates to a square, y^2, compute $GCD(x + y, N)$ and $GCD(x - y, N)$

It is plain to see why this works, because if $x^2 - N = y^2$ then it also follows that $N = x^2 - y^2 = (x - y)(x + y)$ hence anytime we have $x^2 \equiv y^2 \pmod{N}$ where $x \not\equiv \pm y \pmod{N}$ the factors can be computed via GCD

Taking $N = 4387$ as an example:

Step 1 (Calculate $\lceil \sqrt{N} \rceil$)

$\lceil \sqrt{4387} \rceil = 67$

Step 2 (Starting from $x = \lceil \sqrt{N} \rceil$, calculate $f(x) = x^2 - N$):

67^2–4387 = 102 (not a square)
68^2–4387 = 237(not a square)
69^2–4387 = 374 (not a square)
70^2–4387 = 513 (not a square)
71^2–4387 = 654 (not a square)
72^2–4387 = 797 (not a square)
73^2–4387 = 942 (not a square)
74^2–4387 = $1089 = 33^2$ (square)

Step 3 (If the result is also a square, calculate the GCD on the difference.):

$74^2 \equiv 33^2 \pmod{4387}$

$GCD(74 + 33, 4387) = 107$
$GCD(74 - 33, 4387) = 41$

Both the Quadratic Sieve and Number Field Sieve algorithms (to do: insert references) are expansions on this method.

Chapter II – Analysis

a. Binomial expansion

As seen in chapter 1 the goal is to find the following numbers that we can use to take the GCD with:

$GCD(x + y, N) = q$
$GCD(x–y, N) = p$

x and y can thus be rewritten as the differences of the factors of N divided by 2, $(p + q)/2$ or $(p–q)/2$.
In addition, many such possible x and y solutions exist created by $(ap + bq)/2$ and $(ap–bq)/2$ where a and b are multipliers.
In practice we will not know the factors p and q, but treating them as unknowns we can calculate residues in Z/ℓ (for a prime ℓ) for them using the framework we're establishing here.

If $a = 13$, $b = 5$ and $N = 4387$ (and thus factors as $p = 41$ and $q = 107$)

We get:

$(13 \times 41 + 5 \times 107)/2 = 534$
$(13 \times 41–5 \times 107)/2 = –1$

And we can see this creates the following expression when we square 534 $\pmod N$:

$534^2–4387 \times 13 \times 5 = 1^2$

One way to represent x and y is through binomial expansion.
I will shorten binomial expansion as $BE(x–y)^e$.
For example $BE(x–74)^2 = x^2–148x + 5476$.

Substituting the binomial term x in $BE(x–74)^e$, with for example 33, we get:

$BE(33–74)^1 = 33–74 = –41$
$BE(33–74)^2 = 33^2–148 \times 33 + 5476 = –41^2$
$BE(33–74)^3 = 33^3–222 \times 33^2 + 16428 \times 33–405224 = –41^3$
$BE(33–74)^4 = 33^4–296 \times 33^3 + 32856 \times 33^2–1620896 \times 33 + 29986576 = –41^4$

Conversely we also see the following occurring if we substitute in the factor as root instead (or swapping a binomial term with the binomial value):

$BE(41–74)^1 = 41–74 = –33$
$BE(41–74)^2 = 41^2–148 \times 41 + 5476 = –33^2$
$BE(41–74)^3 = 41^3–222 \times 41^2 + 16428 \times 41–405224 = –33^3$
$BE(41–74)^4 = 41^4–296 \times 41^3 + 32856 \times 41^2–1620896 \times 41 + 29986576 = –33^4$

In addition, if we have even exponents we can substitute the constant by a multiple of N and generate a polynomial $g(x)$ to solve for 0.

$BE(41–74)^2 = 41^2–148 \times 41 + 5476 \rightarrow g(x) = 41^2–148 \times 41 + 4387 = 0$
$BE(41–74)^4 = 41^4–296 \times 41^3 + 32856 \times 41^2–1620896 \times 41 + 29986576 \rightarrow g(x) = 41^4–296 \times 41^3 + 32856 \times 41^2–1620896 \times 41 + 4387 \times 6565 = 0$

Note: If the binomial terms generated with $(ap + bq)/2$ and $(ap–bq)/2$ have multipliers a and b other than 1, then you may need to translate your polynomial into a non-monic to get p or q to solve for 0, otherwise if it is monic, it will be a multiple of p or q, decided by multipliers a and b.

Similarly, we can expand the other binomial term and generate a quadratic that solve for 0:

$BE(41 + 33)^2 = 41^2 + 66 \times 41 + 1089 \rightarrow g(x) = 41^2 + 66 \times 41 – 4387 = 0$
$BE(41 + 33)^4 = 41^4 + 132 \times 41^3 + 6534 \times 41^2 + 143748 \times 41 + 1185921 \rightarrow g(x) = 41^4 + 132 \times 41^3 + 6534 \times 41^2 + 143748 \times 41 – 4387 \times 6565 = 0$

From this we can constructed a two-sided quadratic which we will use a lot in this paper:

x^2–$148x + 4387 = x^2 + 66x - 4387$

This two sided setup has been the premise for my research.

b. The discriminant

Of-course we cannot blindly perform binomial expansion on random numbers hoping to find a difference of squares. We will need to construct a way to sieve this.

Take for example the following binomial expansion $BE(x–74)^2 = x^2–148x + 5476$ and on this quadratic polynomial , we substitute the constant with N (for example $N = 4387$):

$g(x) = x^2–148x + 4387$

If we now calculate the discriminant of $g(x)$, $DISC(g(x)) = 148^2–4 \times 4387 = 66^2$, we see that the result is also square.
And we now have the square relation $148^2 \equiv 66^2 \pmod{4387}$, where $GCD(148–66, 4387) = 41$.

An observation:

From this we can make the following generalization that we can derive the factorization of a semi-prime N, if we find a quadratic polynomial with a constant set to N, or a multiple there-of, and whose discriminant yields a square. And from this it is also true that the discriminant would yield a quadratic residue modulo any prime if the discriminant is square in Z. Hence we can use this property to calculate residues in Z/ℓ where valid solutions in Z/ℓ are a quadratic residue and discard residues for quadratics where this is not the case. Now it is also true, that if we find a polynomial whose discriminant is square in Z then it must have a multiple of the factor of N as root for polynomials g(x) at degree 2. That relation between polynomial and discriminant is still something I am researching. One way this is definitely useful is the fact that if the discriminant is square in Z, then the quadratic producing the discriminant must be reducible in Z. Furthermore, if a quadratic is reducible in Z/ℓ for prime ℓ, then this also tells us the discriminant is a quadratic residue in Z/ℓ. This lets us inspect wether or not a residue is square in Z/ℓ even when ℓ does not divide the discriminant.

c. An extended example

Let $N = 4387$
Let binomial $b = x–534$.
Hence $BE(x–534)2 = x^2–1068x + 534^2$

We now need to substitute the constant in the resulting polynomial with a multiple of N.
For example let N multiplier $k = 1$:

$k = 1 \rightarrow g(x) = x^2$–$1068x + 4387$

$DISC(g(x))$ is not a square in Z hence we know $k = 1$ cannot be a valid solution.
However if $k = 65$:

$k = 65 \rightarrow g(x) = x^2$–$1068x + 4387 \times 65$ and $DISC(g(x)) = 2^2$

The discriminant is defined as $DISC(g(x)) = (2t)^2$, where t is the unknown binomial term, we can now substitute this into binomial b.
$b(1) = 1$–$534 = -533$ and $g(533) = 0$. Since g(x) evaluates to 0 at 533 we know we have a correct solution and we can take the gcd using our binomial terms:

$GCD(1 + 534, 4387) = 107$
$GCD(534$–$1, 4387) = 41$

In the next chapter we will see how to find these solutions in Z/ℓ

Chapter III – Improvements

a. Calculating polynomial residues in Z/ℓ

Let us restrict ourselves to binomial expansions to the second degree.
The way we calculate polynomial residues is that we must consider all possible coefficients whose discriminant evaluates to a quadratic residue.
Here we calculate all possible residues for the quadratic polynomial $ax^2 + bx + Nk$ where $N = 4387$ (you can also calculate them for $ax^2 + bx$–Nk instead, it just changes the side we are calculating in the expression $x^2 = y^2 \pmod N$). We will ignore the cases where the leading coefficient is 0, as this would downgrade the degree of our polynomial. And in addition we will also ignore the case where k = 0 for the sake of brevity.

Also note that we can figure out the coefficient for the other side, say we have $ax^2 + b_0x + Nk = ax^2 + b_1x$–$Nk$, by taking the derivative of either b_0 or b_1. You could use this to construct a residue map for both sides rather then one. You cannot simply choose a random b_0, b_1 pairing, they must be each-other's derivative.

Mod 3

$k = 1, a = 1$ and $b = -1 \rightarrow x = 2, k = 1, a = 1$ and $b = -2 \rightarrow x = 1, k = 1, a = 2$ and $b = 0 \rightarrow x = 1, 2$
$k = 2, a = 1$ and$b = 0 \rightarrow x = 2, 1, k = 2, a = 2$ and $b = -1 \rightarrow x = 1, k = 2, a = 2$ and $b = -2 \rightarrow x = 2$

Mod 5

$k = 1, a = 1$ and $b = -2 \rightarrow x = 4, 3, k = 1, a = 1$ and $b = -3 \rightarrow x = 2, 1, k = 1, a = 2$ and $b = 0 \rightarrow x = 3, 2,$
$k = 1, a = 2$ and $b = -1 \rightarrow x = 4, k = 1, a = 2$ and $b = -4 \rightarrow x = 1, k = 1, a = 3$ and $b = 0 \rightarrow x = 1, 4,$

$k = 1, a = 3$ and $b = -2 \rightarrow x = 2, k = 1, a = 3$ and $b = -3 \rightarrow x = 3, k = 1, a = 4$ and $b = -1 \rightarrow x = 3, 1,$

$k = 1, a = 4$ and $b = -4 \rightarrow x = 4, 2$

$k = 2, a = 1$ and $b = 0 \rightarrow x = 4, 1, k = 2, a = 1$ and $b = -1 \rightarrow x = 3, k = 2, a = 1$ and $b = -4 \rightarrow x = 2,$

$k = 2, a = 2$ and $b = -1 \rightarrow x = 2, 1, k = 2, a = 2$ and $b = -4 \rightarrow x = 3, 4, k = 2, a = 3$ and $b = -2 \rightarrow x = 3, 1,$

$k = 2, a = 3$ and $b = -3 \rightarrow x = 4, 2, k = 2, a = 4$ and $b = 0 \rightarrow x = 2, 3, k = 2, a = 4$ and $b = -2 \rightarrow x = 4,$

$k = 2, a = 4$ and $b = -3 \rightarrow x = 1$

$k = 3, a = 1$ and $b = 0 \rightarrow x = 3, 2, k = 3, a = 1$ and $b = -2 \rightarrow x = 1, k = 3, a = 1$ and $b = -3 \rightarrow x = 4,$

$k = 3, a = 2$ and $b = -2 \rightarrow x = 2, 4, k = 3, a = 2$ and $b = -3 \rightarrow x = 1, 3, k = 3, a = 3$ and $b = -1 \rightarrow x = 3, 4,$

$k = 3, a = 3$ and $b = -4 \rightarrow x = 2, 1, k = 3, a = 4$ and $b = 0 \rightarrow x = 4, 1, k = 3, a = 4$ and $b = -1 \rightarrow x = 2,$

$k = 3, a = 4$ and $b = -4 \rightarrow x = 3$

$k = 4, a = 1$ and $b = -1 \rightarrow x = 2, 4, k = 4, a = 1$ and $b = -4 \rightarrow x = 1, 3, k = 4, a = 2$ and $b = 0 \rightarrow x = 4, 1,$

$k = 4, a = 2$ and $b = -2 \rightarrow x = 3, k = 4, a = 2$ and $b = -3 \rightarrow x = 2, k = 4, a = 3$ and $b = 0 \rightarrow x = 2, 3,$

$k = 4, a = 3$ and $b = -1 \rightarrow x = 1, k = 4, a = 3$ and $b = -4 \rightarrow x = 4, k = 4, a = 4$ and $b = -2 \rightarrow x = 1, 2,$

$k = 4, a = 4$ and $b = -3 \rightarrow x = 3, 4$

The problem is, that we can Chinese remainder these together to modulo 15, but then the solution set of possible residues grows quickly (the product of the size of both sets of solutions). So we need to come up with a smart way to generate solutions with this setup.

b. Singular quadratic roots vs non-singular quadratic roots

In the above residue mapping, we will observe that sometimes a set of coefficients will have a single root solution, and sometimes it will have two root solutions.
For example, looking at Mod 5 we see the following two coefficient, root pairings:

$k = 1, a = 4$ and $b = -1 \rightarrow x = 3, 1$ Non-singular roots
$k = 1, a = 3$ and $b = -2 \rightarrow x = 2$ Singular root

Singular roots indicate that the corresponding binomial term has prime ℓ (5 here) as divisor and that the discriminant is also divided by prime ℓ.
We have calculated the above residues using polynomials of the shape $ax^2 + bx + Nk$.
If we calculate the derivative in Z/ℓ we get the residue for linear coefficient b in corresponding to a quadratic with the signs flipped $ax^2 + bx–Nk$ (the other side of the expression $x^2 = y^2 \pmod N$ that we are trying to solve for).
Calculating the derivative for coefficients with a singular root in Z/ℓ gives us a derivative that is 0 in Z/ℓ. Since a singular root means $DISC(g(x)) = 0 \pmod \ell$ and since $DISC(g(x)) = (2t)^2$,we have that $t \equiv 0 \pmod \ell$. (strictly speaking about degree 2 polynomials here).

For example:

Using $k = 1, a = 3$ and $b = -2 \rightarrow x = 2$ we construct the polynomial $g(x) = 3 \times 2^2–2 \times 2 + 2$ in $Z/5$.
And we see that the derivative $g(x)' \equiv 2 \times 3 \times 2–2 \equiv 0 \pmod 5$ (or in other words 0 in $Z/5$).
This implies that the binomial term y, in binomial b = t – y equals to 0 in Z/5.

Non-singular roots always yield non-zero residues for the binomial terms in Z/ℓ .

C. p-adic lifting of polynomials in $\mathbf{Z}/\ell$

Let us say we have the following coefficient and root pairing from our residue mapping:

$k = 4, a = 3$ and $b = -1 \rightarrow x = 1$ in $Z/5$ gives us polynomial $g(x) = 3 \times 1^2 - 1 \times 1 + 4387 \times 4 = 0 \pmod 5$

This gives us all of the following possible solutions in $Z/5^2$ where also the derivative yields 0 in $Z/5^2$, for the sake of brevity we will focus on these cases only:

$k = 4, a = 3$ and $b = -1 \rightarrow x = 21, k = 4, a = 8$ and $b = -6 \rightarrow x = 16, k = 4, a = 13$ and $b = -11 \rightarrow x = 11$
$k = 4, a = 18$ and $b = -16 \rightarrow x = 6, k = 4, a = 23$ and $b = -21 \rightarrow x = 1$
$k = 9, a = 3$ and $b = -11 \rightarrow x = 6, k = 9, a = 8$ and $b = -16 \rightarrow x = 1, k = 9, a = 13$ and $b = -21 \rightarrow x = 21$
$k = 9, a = 18$ and $b = -1 \rightarrow x = 16, k = 9, a = 23$ and $b = -6 \rightarrow x = 11$
$k = 14, a = 3$ and $b = -21 \rightarrow x = 16, k = 14, a = 8$ and $b = -1 \rightarrow x = 11, k = 14, a = 13$ and $b = -6 \rightarrow x = 6$
$k = 14, a = 18$ and $b = -11 \rightarrow x = 1, k = 14, a = 23$ and $b = -16 \rightarrow x = 21$
$k = 19, a = 3$ and $b = -6 \rightarrow x = 1, k = 19, a = 8$ and $b = -11 \rightarrow x = 21, k = 19, a = 13$ and $b = -16 \rightarrow x = 16$
$k = 19, a = 18$ and $b = -21 \rightarrow x = 11, k = 19, a = 23$ and $b = -1 \rightarrow x = 6$
$k = 24, a = 3$ and $b = -16 \rightarrow x = 11, k = 24, a = 8$ and $b = -21 \rightarrow x = 6, k = 24, a = 13 and b = -1 \rightarrow x = 1$
$k = 24, a = 18$ and $b = -6 \rightarrow x = 21, k = 24, a = 23$ and $b = -11 \rightarrow x = 16$

Each of these solutions map to the same solution in $Z/5$. We see a total of 25 (or 125 if we also count those with non-zero derivatives $Z/5^2$) possible solutions in $Z/5^2$ generated from a single solution in $Z/5$. I refer to p-adic lifting, but usually it is used in the context of lifting roots of polynomials, but in the context that I use this term, it is also used to lift coefficient residues in Z/ℓ p-adically. So please keep that in mind. I tend to create these abstractions in my head and the names I give these abstract concepts may not be strictly correct in formal mathematical terms. But to me it would seem, that whether we are lifting root residues or coefficient residues, the underlying mechanism are very similar and connected.

D. Converting singular roots to non-singular roots and visa versa

Because we are searching for two squares such that $x^2 = y^2 \pmod N$, and since both x and y can also be represented as binomials which can be expanded to the second degree, we can substitute the expression $x^2 = y^2 \pmod N$ with the more elaborate: $ax^2 - b_0x + Nk = ax^2 + b_1x - Nk$.

This also means that we can convert singular roots to non-singular roots temporarily, simply by transferring a root to the other side of the equation in Z/ℓ. As long as the side we transfer to has a discriminant that is not divisible by ℓ. (there is a few other edge cases that will complicate things, for example when k divides by ℓ).

For example $1x^2 + 66x - 4387 = 0 \pmod{37}$ has a singular root, while the other side, $1x^2 - 0x + 4387 = 0 \pmod{37}$ has two roots in Z/ℓ.

$b_1 = 66 \rightarrow 1 \times 4^2 + 66 \times 4 – 4387 = 0 \pmod{37}$

$b_0 = 0 \rightarrow 1 \times 4^2 – 0 \times 33 + 4387 = 0 \pmod{37}$
$b_0 = 0 \rightarrow 1 \times 33^2 – 0 \times 33 + 4387 = 0 \pmod{37}$

A polynomial that produces the other singular root can also be found, in this case all we have to do is flip the sign on linear coefficient b_1, then the singular root becomes the other non-singular root for b_0 (we must also swap a and k, but both are 1 here):
$b_1 = –66 \rightarrow 1 \times 33^2 – 66 \times 33 – 4387 = 0 \pmod{37}$

One way this ends up being useful is that with this construction we never have to p-adically lift singular roots (except for a couple of edge cases like $Z/2^e$ or when the leading coefficient or k is divisble by ℓ) , as we can just change them to non-singular, lift them and use them as lifted root for the singular side.

Chapter IV - Residue and two-sided sieving

a. Residue sieving

As mentioned earlier, we cannot simply precompute residues and combine them with Chinese Remainder Theorem hoping to find a valid solution, because the possible set of solutions grows too quickly. The easiest way to take advantage of these residues is with a Quadratic Sieve style sieving setup. In addition, due to the setup we have build in prior chapter, there is a unique two-sided variation that we can construct that are not seen in standard SIQS/MPQS variants.

Let us say that $N = 4387$

We can setup our sieve easily such that we always have a guaranteed factor up to a maximum of $\sqrt{N}$. This can be achieved in two different ways, either we can do it the way SIQS/MPQS does by encoding the modulus into the leading coefficient or through using a modulus instead.

Lets say at each interval step, we always want 11 as a divisor for every B-smooth candidate generated. SIQS/MPQS uses the following procedure:

$N = 4387$ (semi-prime to be factored)
$m = 11$(modulus)

Next we generate a quadratic polynomial this way:

First find x such that $x^2 - 4387 \equiv 0 \pmod{m}$
This gives us that $x = \pm 3 \pmod{11}$
Calculate the coefficient a,b and constant c as following:
$a = m$
$b = 2x$
$c = (x^2 - N)/m$

This gives us:

$g(x) = 11x^2 + 6x - 398$

And we sieve the roots of this polynomial. This is a non-monic quadratic, meaning it can be translated into a monic by multiplying the root and constant with the leading coefficient. This in essence multiplies the polynomial value with the leading coefficient.

For example if evaluating $g(x)$ with 5:

$g(5)$ = -93

Translating non-monic $g(x)$ into monic $h(x)$, $h(x)$ becomes:
$h(x) = z^2 + 6z - (398 \times 11)$ and evaluating with $z = x \times a$ we get $h(55) = 11 \times -93$, and we see this effectively transfers the leading coefficient to the polynomial value.

Furthermore we can add linear coefficient b to the constant c, using the formula $c- = (b/2)^2$, making c equal to N again, and now evaluating $u^2 - 4387$ with $u = z + (b/2)$ gives use $58^2 - 4387 = 11 \times -93$ which is exactly the format that we require.

The problem with this is, that the smallest polynomial values that we can generate with $x^2 - N$, live near x = $\lceil\sqrt{N}\rceil$, and with each increment of x, the polynomial value changes by roughly $2\sqrt{N}$, meaning if we are to sieve that range, our polynomial values will be roughly of size $2\sqrt{N}$. So no matter what we do, we cannot strip away more known factors then roughly $\sqrt{N}$, since we're in essence just dividing the polynomial values in advance.

But I independently also (re)discovered that we can simply use a modulus instead without manipulating the quadratic (although this is still a fairly basic number theoretical setup and I assume would be known already, hence I do not want to claim novelty here):

If we know that for $x = 3 + 11i$ we generate polynomial values that are divisible by 11 with $x^2 - 4387$, then we can just create a sieve interval where each step iterates i. And using basic number theory, such as CRT, all the other optimizations can also be applied that SIQS achieves, such as graycodes and self-initialization. I have demonstrated this in my code. So functionally, there really isn't much of an a difference. But what is different is that with my way of doing things, we still have the option of experimenting with coefficients, which are kind of locked in with the SIQS/MPQS method.

This was something that I achieved at around 1.5 years into my research. The next question I was then left with was: How to improve from here, can we use these SIQS/MPQS type transformations, which we have not utilized yet, to add more performance?

b. Two-sided approach and algorithmic improvements

EDIT: I think the better approach would be to try and setup a novel

two-sided construction using number fields... staying in Z really doesn't work. There may be a way to use binomial expansions and have number fields on both sides of the square relation $z^2 + Nk = y^2$ rather then have one sides pinned to the rationals like NFS does... I will do some experimentation shortly.

Earlier I have shown how a square relation like $z^2 + Nk = y^2$ can also be represented as two quadratics: $ax^2 + b_0x + Nk = ax^2 + b_1x - Nk$. In addition we also know, from earlier discussions, that if the discriminant for both the left-sided quadratic and right sided quadratic is a square, then we can continue to take the $GCD(b_0 + b_1, N)$ and if $b0 \not\equiv \pm b1 \pmod N$ we find a non-trivial factor of N, since b_0 and b_1 are simply the binomial terms.

This also means that if we find two reducible quadratics, the same condition is met. Ideally we find an algorithm that completely eliminates sieving, and this is still an ongoing area of research for me. However, something we can do is, if we find a B-smooth, calculate residues within our factor base and do a quick sweep to see if we can construct a reducible quadratic.

The biggest bottleneck of Quadratic Sieve style algorithms is the fast growth of the required factor base size beyond circa 110 digits.
So even if we were to find a faster sieve, it is just not going to punch past that and hence would not be the correct research direction. The only real chance we have is to either find a way to quickly find near identical B-smooths or find a way to turn a B-smooth square under certain conditions.

I now want to propose an approach to quickly sweep a large residue region to see if we can turn a B-smooth square in Z. Lets say we attack $N = 4387$, and during the first phase of the algorithm, we simply sieve $z^2 + N = r$, if r is B-smooth we continue to the steps that will be described below. Further more, we make sure that the modulus we sieve with is square, so that ~half of the B-smooth is a square. As for practical purposes the factorization of squares can be ignored, since they get canceled out with Gaussian elimination over $Z/2$, hence giving us a cheap reduction in the bit-length of B-smooths.

Lets say $r = 37$ after dividing out all the square factors.

First we calculate roots for$z^2 + N = 0 \pmod{37}$. This gives use $z = 4, 33$
Next we construct our polynomial $f(x)$.

In SIQS/MPQS we would calculate the constant for $f(x)$ as $c = (z^2 - N)/m$, (note, $m = r$ here as we're setting r to be our modulus), since our roots are calculated using $z^2 + N \equiv 0 \pmod{37}$, and this is linked to the quadratic $ax^2 + b_1x - Nk$, we can calculate the constant for the quadratic polynomial using the root and formula: $c = -(N - (r^2 - (r - z)^2))$.
This gives us a constant we can use for $f(x)$. So our polynomial $f(x)$ now becomes $f(x) = 1x^2 + r2x - c$. and now we also are able to generate two linear polynomial that will represent the factorization of the other quadratic: $g(x) = 1x + (y2 - z)$ and $h(x) = 1x + z$ and $i(x) = g(x) \times h(x)$. next if we iterate this process with $z + ri$, where i is an iterator, then $i(x)$ will always be reducible.

Taking the polynomial product $i(x) = g(x) \times h(x)$ we can construct a quadratic where the discriminant $DISC(i(x))$ always yields a square. So in essence we can now sieve the other way around and all that ends up mattering is for $DISC(f(x))$ to be square in Z, or in other words, f(x) must be reducible. And we will also see that $DISC(i(x) \equiv DISC(f(x)) \pmod{N}$

Example:

If $z = 33$, then we calculate the constant c for $f(x)$ as $c = -(4387-(37^2-(37-33)^2)) = -3034$ hence $f(x) = 1x^2+(37\times 2)x-3034$, $g(x) = 1x+((37\times 2)-33)$and $h(x)1x+33$ Calculating the polyomial product $i(x)$ of $g(hx)\times h(x)$ we get $i(x) = 1x^2+74x+1353$.

Calculating the discriminants of $i(x)$ and $f(x)$ we now get $DISC(i(x)) = 74^2 - 4 \times (1353 \times 1) = 8^2$ and $DISC(f(x)) = 74^2 + 4 \times (3034 \times 1) = 37 \times 476$, and we see that $DISC(i(x)) \equiv DISC(f(x)) \pmod{N}$, however in this case $DISC(f(x))$ is not square and $f(x)$ also not reducible in Z.

However we can calculate polynomial residues generated by iterating $z + ri$, and considering only those that are reducible in Z/ℓ, and we can also use p-adic lifting, since reducibility for the correct solution in Z will also remain true there. If a correct solution exists within a residue region, determined by iterator i (discard any residues that grow beyond this), then it will show up for any ℓ^e. In practice you would probably take a handful of primes and perform p-adic lifting to see if any solution in z settles within that region as we lift.

To do: I still need to implement the code for this. A sieving version that implements much of the math behind it can already be found on my github under the "polysieve" folder: https://github.com/BigPolarBear1/factorization/tree/main , specifically in the function find_same().. now that code needs to be replaced with residues and lifting rather then a sieving approach, but the underlying math is the same.

c. Benchmarks

to do

Closing thoughts and future work

to do