Factorization – a three year journey

Preview Release v3.1

Author: Essbee Vanhoutte
Date: December 2025

To do before release:

1. Check for consistency with the signs in my quadratics.... right side of the congruence should be negative roots and negative linear coefficient. Left side should be negative roots and positive coefficient. And even though on the right side this always produces a positive linear term and visa versa on the left side.. I should still be consistent with it for the sake of clarity.
2. Include numerical example in final chapter once code is completed..
3. Make sure to add more references

Preface

This paper is dedicated to my family and friends, among whom are my former managers who gave me the time and support to grow my interest in cryptography and eventually the math behind it. When I started this journey three years ago, as a high-school drop-out with limited math education, this seemed like an impossible mountain to scale. This paper documents my findings, and proposes an improvement to known factorization algorithms.

Chapters

I. Integer to Quadratic (pg. 1)
II. Fermat's factorization method (pg. 2)
III. Quadratic to Quadratic congruence (pg. 3)
IV. Combining modular linear coefficient solutions (pg. 4)
V. Bridge to Quadratic Sieve (pg. 7)
VI. Quadratic coefficients (pg. 8)
VII. Lifting coefficients (pg. 9)
VIII. Improved Quadratic Sieve (pg. 9)
IX. The quest for a polynomial time algorithm (pg. 9)
X. Conclusion (pg. 11)
XI. References (pg. 11)

I. Integer to Quadratic

In attempting to break factorization, it is important to find the correct representation of the problem.
To describe a problem, one of your first steps will be to define some type of algebraic expression with variables representing the solutions you are trying to find.
The Number Field Sieve algorithm[1] is a great example of this, where the very beginning of the algorithm is the polynomial selection step.
I would like to preface this chapter, by giving a short review of the steps that led me to my representation of quadratics of the form:

$x^2 + y_0x + N = 0$ and $x^2 + y_1x - N = 0$. (And eventually $zx^2 + y_0x + N = zx^2 + y_1x - N$, but we ignore the quadratic coefficients until later)

This entire paper is about semi−primes. A semi−prime is a composite number, which has exactly 2 unique prime factors (excluding itself and 1). For the security of the RSA algorithm, this is the only format that matters. Numbers with more factors then this will actually weaken the RSA algorithm[2] (since it reduces the possible search space for factors).
You will need to make your own generalizations if you want to apply this to composites with more factors.

One example of a semi−prime I've used often in the last three years is:

$p = 41$ and $q = 107$ => $N = 41 \times 107 = 4387$ (Where semi-prime N is the product of factors p and q)

One reason I would often use the same semi−prime while doing research, is that you eventually become so familiar with its structure, that any patterns immediately stand out. However, I always make sure my findings generalize to any other semi−prime as well. In this paper I will exclusively use $p = 41$ and $q = 107$. You can use my proof of concepts to verify these findings against arbitrary semi−primes. I am not an educated Mathematician, hence I will not even attempt to write a paper with algebraic letter−soup.

Around three years ago, one of the first things I very quickly started to zero in on, was using modular reduction on N. My initial thought process was that perhaps I could find some type of pattern in the remainders that would enable me to construct an algorithm.

For example if $N = 4387$:

4387	=	47	mod 70
4387	=	40	mod 69
4387	=	35	mod 68

4387	=	32	mod 67
4387	=	31	mod 66 ($\sqrt{N}$)
4387	=	32	mod 65
4387	=	35	mod 64
4387	=	40	mod 63
4387	=	47	mod 62

The rounded $\sqrt{N}$ (square root of *N*) is a point of symmetry for the remainders.
In the above example, starting from mod 66, we get the following sequence as remainders in both directions if we increase or decrease the modulus:

31, 32, 35, 40, 47, .. $= 31 + \sum_{i=1}^{N} 2i - 1$

Any trained mathematician will immediately recognize this sequence as simply 31 plus the squares.
This is as expected, since we are modulo reducing a number close to its square root.

Now we can rephrase factorization as finding some modulus *m*, such that *N* = 0 mod *m* (aka m is a divisor of *N*).
And we also know that the remainders can be calculate by simply adding squares to the initial remainder at the (rounded) square root of *N*.
Around two years ago, I was maybe a couple of weeks in to my research, when I made these realizations and thought :

“Okay, I can do this, this can't be very hard if I know by how much the remainder increases or decreases with each increment or decrement of the modulus, hence it should be easy to figure out when it hits 0 mod *m*!”.

Math is full of problems that are very deceptive. Intuitively it would seem like a straight forward mathematical tool to solve this should exist, but the reality of it is much more complicated. At the surface it looks like a shallow pond, but below the surface, are depths of complexity deeper then the deepest oceans.

Thus if we take *N* mod *m*, we need to figure out a way to adjust modulus *m* such that the remainder reaches 0 mod *m*. At first I would mess around with some algebraic expressions which had a variable in the modulus. Since to make the remainder change in value, we have to increment or decrement the modulus. However, soon after I was able to generalize it to this, removing any unknowns in the modulus:

$x^2 + yx + N = 0 \bmod m$

The root, *x*, would represent the smallest factor. In this case 41. And we negate the *x* so it becomes negative: −41.
The coefficient *y*, would represent the sum of both factors, in this case 41 + 107 or 148.

Thus we get: $(-41)^2 + 148 \times -41 + 4387 = 0$

or simplified:

$41^2 - 148 \times 41 + 4387 = 0$

And of−course this will also equal 0 for any mod *m*.

Note: My motivation for making the root, *x,* negative sign, is that it represents the distance to the factor, rather then simply the factor itself. This is related to how I came up originally to represent factorization this way. I went a little bit into the background on this in my original paper, but truth is, its not very relevant, it's simply one of many ways to represent this problem.

We can also flip the sign for the constant in the quadratic, but then the linear coefficient instead of being $p + q$ becomes $p - q$.

If $p + q = 148$ then $p - q = -66$

Setting *y* to −66 and flipping the sign of the constant the quadratic becomes: $41^2 + (-66) \times -41 - 4387 = 0$
Hence: $41^2 - 148 \times 41 + 4387 = 41^2 + 66 \times 41 - 4387$

We have two unknowns, the root and the linear coefficient (ignoring the quadratic coefficient for now). However, isolating the linear coefficient on both sides - which we can do if they share the same root – of both the quadratics and squaring them mod *N* we get:

$148^2 = 66^2 \bmod 4387$ (we can ignore the signs when squaring)

And because these two squares are congruent mod *N*, we can take the GCD (greatest common divisor) of their difference:

GCD(148 + 66, 4387) = 107
GCD(148 − 66, 4387) = 41

Note: Calculating the greatest common divisor can be done very quickly using the Euclidean algorithm[3].

Because of this property, just finding the correct pair of linear coefficients of a quadratic will result in the factorization of *N*.
We can formalize this as:

$x^2 + y_0x + N = 0$ and $x^2 + y_1x - N = 0$

We can factor *N* by finding y_0 and y_1 and taking the GCD of their difference.
This is very similar to what Fermat's factorization method[4] does.

Further down in the paper we will further simplify this to using the roots rather the linear coefficients. This is not simply a math paper. It is a description of my mathematical journey. Starting with my discoveries made in the first year when I was still a novice with no math education, and slowly adding complexity to it in chronological order as I uncovered more and more in the years that followed.

II. Fermat's factorization method

It is important to now explain Fermat's factorization method, so we can draw some parallels with my own findings, which accidentally ended up converging with Fermat's factorization method. But this just shows how fundamental this is to the factorization problem.

In its most basic form the procedure is as this:

1. Calculate the rounded $\sqrt{N}$.
2. Starting from $x = \sqrt{N}$, calculate $y = x^2 - N$ (do so in a loop while incrementing x).
3. If y is also a square, calculate the greatest common divisor (GCD) on the difference.

Taking $N = 4387$ as an example:

Step 1 (Calculate the +/− square root of N):

$\sqrt{4387}$ = 66 (rounded)

Step 2 (Starting from the square root calculate $y = x^2 - N$):

$66^2 - 4387 = -31$ (not a square)
$67^2 - 4387 = 102$ (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 y is also a square, calculate the GCD on the difference.):

$74^2 = 33^2 \bmod 4387$

GCD(74 + 33, 4387) = 107
GCD(74 – 33, 4387) = 41

You may notice that 74 and 33 are simply 148 and 66 divided by 2.
This brings us to another important point. Many such square relations can be found mod N. What will be different is the amount of times N is in−between both squares.

$148^2 = 66^2 + 4 \times 4387$ (four times N in between)
$74^2 = 33^2 + 4387$ (one times N in between)

This difference of N, is related to the quadratic coefficient. More on this in chapter VI.

Do be aware, there are two types of square relations mod N. One which will yield a trivial factorization (1 or N) and one which will yield a non−trivial factorization (a prime factor of N). The square relations that will always yield a trivial factorization are of the form:

$a^2 = -a^2 \bmod N$

We are not interested in theses square relations.

Both Quadratic Sieve[5] and Number Field Sieve - the current fastest factorization algorithms - are more elaborate ways of finding these square relations mod N. Both algorithms were invented by Carl Pomerance[6]. These algorithms are now almost 40 years old, and not much progress aside from a handful of tweaks to these algorithms has been made since. In my opinion, this is not acceptable. A problem as important as factorization should not go without major progress for 40 years. And simply hoping Quantum computing will solve everything is foolish. Thinking as such is the same as thinking AI will replace everything. It is but an excuse to stop trying. We should never stop trying. The day we stop trying, we surrender ourselves to ignorance.

III. Quadratic to Quadratic congruence

Going back to representing factorization as the following Quadratic:

$x^2 + yx + N = 0$

Finding a root and linear coefficient solution in the integers to this is very hard and not easily solve-able.
One approach is to create “fragments” of a possible solution by reducing everything to mod $p_0, p_1, p_2, .., p_{n-1}$ (where p is prime), finding integer solutions mod $p_0, p_1, p_2, .., p_{n-1}$ and then combining them. This in essence turns the problem into a subset sum[7] type of problem, because then it becomes a matter of which “fragments”, aka solutions mod p_i , to combine.

Our representation now becomes the same quadratic but with modular reduction:

$x^2 + yx + N = 0 \bmod m$

Instead of finding solutions in the integers, we reduce the scope to mod m.

Example:

$N = 4387 = 6 \bmod 13$
$p = 41 = 2 \bmod 13$
$q = 107 = 3 \bmod 13$

The residue of N mod 13 is the residue of pq mod 13 ($2 \times 3 = 6 \bmod 13$).
The residue of linear coefficient y mod 13 is $p + q$ mod 13 ($2 + 3 = 5 \bmod 13$).
If the coefficient y, is 5 mod 13 and we know our residue of N mod 13 is 6 then only 2 + 3 and 3 + 2 can be our two residues for p and q mod 13.

Thus the root, x, is either −3 or −2. (remember we negate the root)

Plugging in for $y = 5$ and $x = -3$ or $-2 \bmod 13$:

$-2^2 + 5 \times -2 + 4387 = 0 \bmod 13$
$\Rightarrow 4 - 10 + 6 = 0 \bmod 13$

$-3^2 + 5 \times -3 + 4387 = 0 \bmod 13$
$\Rightarrow 9 - 15 + 6 = 0 \bmod 13$

In real life we don't know y is 5 (since we do not know $p + q$).
We do know N mod 13 is 6.
Thus all possible coefficient solutions y mod 13 can be enumerated by summing up each of the two residues mod 13 that multiply to 6.

A coefficient solution y can also be told to exist, if for a given coefficient value y a root solution x exists. This can be trivially determined using the Legendre symbol[8] without actually having to find roots.

All root x and coefficient y solutions mod 13 that solve the quadratic congruence:

$y = 1$ $x = -5 \Rightarrow -5^2 + 1 \times -5 + 4387 = 0 \bmod 13$
$y = 1$ $x = -9 \Rightarrow -9^2 + 1 \times -9 + 4387 = 0 \bmod 13$
$y = 5$ $x = -2 \Rightarrow -2^2 + 5 \times -2 + 4387 = 0 \bmod 13$
$y = 5$ $x = -3 \Rightarrow -3^2 + 5 \times -3 + 4387 = 0 \bmod 13$
$y = 6$ $x = -7 \Rightarrow -7^2 + 6 \times -7 + 4387 = 0 \bmod 13$
$y = 6$ $x = -12 \Rightarrow -12^2 + 6 \times -12 + 4387 = 0 \bmod 13$
$y = 7$ $x = -1 \Rightarrow -1^2 + 7 \times -1 + 4387 = 0 \bmod 13$
$y = 7$ $x = -6 \Rightarrow -6^2 + 7 \times -6 + 4387 = 0 \bmod 13$
$y = 8$ $x = -10 \Rightarrow -10^2 + 8 \times -10 + 4387 = 0 \bmod 13$
$y = 8$ $x = -11 \Rightarrow -11^2 + 8 \times -11 + 4387 = 0 \bmod 13$
$y = 12$ $x = -4 \Rightarrow -4^2 + 12 \times -4 + 4387 = 0 \bmod 13$
$y = 12$ $x = -8 \Rightarrow -8^2 + 12 \times -8 + 4387 = 0 \bmod 13$

IV. Combining modular linear coefficient solutions

Now that we can find (linear) coefficient solutions mod p_i. Let us calculate coefficient solutions mod $p_0, p_1, p_2, .., p_{n-1}$ now and combine the results into mod m (where $m = p_0 \times p_1 \times p_2 \times \ldots \times p_{n-1}$)

y mod 3 = { 1, 2 }
y mod 5 = { 2, 3 }
y mod 7 = { 0, 1, 6 }
y mod 11 = { 1, 2, 5, 6, 9, 10 }
y mod 13 = { 1, 5, 6, 7, 8, 12 }

These are the solutions mod 3,5,7,11,13 that solve the coefficient of the linear term in: $x^2 + yx + N = 0 \bmod m$

If we calculate the solution set mod 15015 we get:

y mod 15015 = { 83, 97, 98, 112, 148, 188, 203, 287, 307, 343, 358, 398, 428, 463, 482, 512, 538, 617, 643, 658, 727, 742, 812, 827, 853, 937, 967, 1007, 1022, 1028, 1058, 1072, 1112, 1253, 1267, 1282, 1288, 1358, 1373, 1442, 1457, 1462, 1483, 1513, 1553, 1567, 1568, 1618, 1637, 1652, 1813, 1828, 1847, 1897, 1912, 1982, 2003, 2008, 2023, 2092, 2107, 2113, 2177, 2183, 2198, 2267, 2282, 2393, 2437, 2443, 2458, 2477, 2528, 2542, 2612, 2638, 2653, 2722, 2723, 2737, 2738, 2807, 2822, 2828, 2848, 2932, 2983, 3002, 3023, 3037, 3067, 3158, 3178, 3262, 3268, 3277, 3283, 3353, 3368, 3437, 3452, 3478, 3548, 3563, 3613, 3632, 3647, 3697, 3752, 3808, 3823, 3892, 3893, 3907, 3947, 3977, 3983, 3998, 4003, 4087, 4102, 4178, 4192, 4193, 4207, 4388, 4402, 4432, 4438, 4453, 4493, 4523, 4577, 4607, 4633, 4648, 4718, 4753, 4802, 4817, 4907, 4922, 4948, 4978, 5032, 5062, 5102, 5117, 5153, 5257, 5312, 5348, 5362, 5363, 5377, 5468, 5543, 5557, 5572, 5578, 5608, 5648, 5663, 5732, 5747, 5803, 5818, 5858, 5908, 5923, 5942, 5972, 6007, 6077, 6103, 6118, 6187, 6202, 6272, 6287, 6293, 6313, 6397, 6467, 6488, 6518, 6532, 6572, 6623, 6727, 6733, 6742, 6748, 6818, 6832, 6833, 6902, 6917, 6943, 6973, 7013, 7027, 7028, 7078, 7097, 7112, 7118, 7162, 7273, 7288, 7357, 7372, 7442, 7448, 7463, 7468, 7547, 7552, 7567, 7573, 7643, 7658, 7727, 7742, 7853, 7897, 7903, 7918, 7937, 7987, 7988, 8002, 8042, 8072, 8098, 8113, 8182, 8183, 8197, 8267, 8273, 8282, 8288, 8392, 8443, 8483, 8497, 8527, 8548, 8618, 8702, 8722, 8728, 8743, 8813, 8828, 8897, 8912, 8938, 9008, 9043, 9073, 9092, 9107, 9157, 9197, 9212, 9268, 9283, 9352, 9367, 9407, 9437, 9443, 9458, 9472, 9547, 9638, 9652, 9653, 9667, 9703, 9758, 9862, 9898, 9913, 9953, 9983, 10037, 10067, 10093, 10108, 10198, 10213, 10262, 10297, 10367, 10382, 10408, 10438, 10492, 10522, 10562, 10577, 10583, 10613, 10627, 10808, 10822, 10823, 10837, 10913, 10928, 11012, 11017, 11032, 11038, 11068, 11108, 11122, 11123, 11192, 11207, 11263, 11318, 11368, 11383, 11402, 11452, 11467, 11537, 11563, 11578, 11647, 11662, 11732, 11738, 11747, 11753, 11837, 11857, 11948, 11978, 11992, 12013, 12032, 12083, 12167, 12187, 12193, 12208, 12277, 12278, 12292, 12293, 12362, 12377, 12403, 12473, 12487, 12538, 12557, 12572, 12578, 12622, 12733, 12748, 12817, 12832, 12838, 12902, 12908, 12923, 12992, 13007, 13012, 13033, 13103, 13118, 13168, 13187, 13202, 13363, 13378, 13397, 13447, 13448, 13462, 13502, 13532, 13553, 13558, 13573, 13642, 13657, 13727, 13733, 13748, 13762, 13903, 13943, 13957, 13987, 13993, 14008, 14048, 14078, 14162, 14188, 14203, 14273, 14288, 14357, 14372, 14398, 14477, 14503, 14533, 14552, 14587, 14617, 14657, 14672, 14708, 14728, 14812, 14827, 14867, 14903, 14917, 14918, 14932 }

Each of these solutions mod 15015 represents a unique combination of solutions mod 3, 5, 7, 11 and 13 (Cartesian product).
There is a number theoretical trick we can use to make findings these solutions mod 15015 a lot easier.

Lets say that we want to calculate the following combination for mod 15015:

y = 1 mod 3
y = 3 mod 5
y = 1 mod 7
y = 5 mod 11
y = 5 mod 13

For each solutions mod p, we divide and then multiple by every other prime. Using the following procedure:

m = total modulus
p = prime

$a = m / p$
$b = y \times a^{-1} \bmod p$
$y = a \times b \bmod m$

For y = 1 mod 3 we get:

a = 15015 / 3 = 5005
b = 1 × 1 = 1 mod 3
y = 5005 × 1 = 5005 mod 15015

This gives us 5005 mod 15015

5005 = 5 × 7 × 11 × 13 and 5005 mod 3 = 1

Hence this simply changes 1 mod 3 to mod 15015 by adding multiples of mod 5,7,11,13 while keeping it congruent to 1 mod 3.

Now to do the rest:

For y = 3 mod 5 we get:

a = 15015 / 5 = 3003
b = 3 × 2 = 1 mod 5
y = 3003 × 1 = 3003 mod 15015

This gives us 3003 mod 15015

For y = 1 mod 7 we get:

a = 15015 / 7 = 2145
b = 1 × 5 = 5 mod 7
y = 2145 × 5 = 10725 mod 15015

This give us 10725 mod 15015

For y = 5 mod 11 we get:

a = 15015 / 11 = 1365
b = 5 × 1 = 5 mod 11
y = 1365 × 5 = 6825 mod 15015

This gives us 6825 mod 15015

For y = 5 mod 13 we get:

a = 15015 / 13 = 1155
b = 5 × 6 = 4 mod 13
y = 1155 × 4 = 4620 mod 15015

This gives us 4620 mod 15015

Adding the results for mod 3,5,7,11,13 together: 5005 + 3003 + 10725 + 6825 + 4620 = 148 mod 15015

In my original paper I would call these intermediate results, partial results. I think that is a fitting name. When we sum up these partial results, we get 148 mod 15015. This way we can reduce finding combinations modulo a composite number to summing together partial results constructed from the prime factors of the composite modulus.

Now what we could do is, calculate all the possible coefficient solutions mod 3,5,7,11,13 and then use the above calculations to create partial results from them mod 15015:

Before:

y mod 3 = { 1, 2 }
y mod 5 = { 2, 3 }
y mod 7 = { 0, 1, 6 }
y mod 11 = { 1, 2, 5, 6, 9, 10 }
y mod 13 = { 1, 5, 6, 7, 8, 12 }

After:

y mod 3 = { 5005, 10010 }
y mod 5 = { 12012, 3003 }
y mod 7 = { 0, 10725, 4290 }
y mod 11 = { 1365, 2730, 6825, 8190, 12285, 13650 }
y mod 13 = { 6930, 4620, 11550, 3465, 10395, 8085 }

One of my original research approaches was the insight that if we could growing the modulus by adding more and more primes, eventually $p + q$ will end up being the smallest solution mod m. The other solutions will keep growing to a number bigger then N.
So my initial idea is, if we generate these partial results, and we select one partial result from each prime modulus and sum them together mod m, how do we find the smallest sum mod m? This however is a modular multiple-choice subset-sum problem. Not easily solve-able.
In my original paper I also discussed constructing partial results in mod m_0 and m_1 each from unique primes and as long as m_0 and m_1 are large enough, then we can use the intersection between both sets of solutions to further narrow down the possible set of solutions.

For example:

Mod $m_0 = 3 \times 7 \times 13 \times 19$ (5187)

y mod 3 = { 1729, 3458 }
y mod 7 = { 0, 4446, 741 }
y mod 13 = { 1197, 798, 1995, 3192, 4389, 3990 }
y mod 19 = { 3003, 3822, 1638, 2457, 273, 4914, 2730, 3549, 1365, 2184 }

Mod $m_1 = 5 \times 11 \times 17 \times 23$ (21505)

y mod 5 = { 8602, 12903 }
y mod 11 = { 13685, 5865, 3910, 17595, 15640, 7820 }
y mod 17 = { 0, 12650, 10120, 16445, 7590, 13915, 5060, 11385, 8855 }
y mod 23 = { 0, 18700, 7480, 1870, 17765, 14960, 6545, 3740, 19635, 14025, 2805 }

From m_0 we select: 1729 + 4446 + 798 + 3549 = 148 mod 5187 and from m_1 we select: 12903 + 3910 + 11385 + 14960 = 148 mod 21505
As you can see, 148 ($p + q$) can be found as a sum mod m_i , this will hold true for any modulus bigger then $p + q$. Hence by inspecting intersections between solution sets, we can quickly narrow it down to one single solution. But in practice, since we need a modulus bigger then $p + q$, the amount of possible sums, aka the order of the Cartesian product, quickly grows. Hence this is not feasible, but nonetheless, it is an interesting direction to approach this problem.
In my first paper I attempted to find these intersections using the LLL algorithm[9]. However many improvement can be made there, and would I write it today, there would be many things I would change and simplify further.

V. Bridge to Quadratic Sieve

After my attempt at finding intersections between solutions sets in mod m_i using LLL, I instead used these findings to generate smooth candidates for the Quadratic Sieve algorithm. I will now quickly describe the transformations I used to achieve this.

Let us factor 4387.

We set the factor base b to:

b = { 3, 5, 7, 11, 13 }

Next using the quadratic formula: $x^2 + yx + N$ mod b_i we calculate all the possible coefficient solutions for each prime in the factor base:

y mod 3 = { 1, 2 }
y mod 5 = { 2, 3 }
y mod 7 = { 0, 1, 6 }
y mod 11 = { 1, 2, 5, 6, 9, 10 }
y mod 13 = { 1, 5, 6, 7, 8, 12 }

Next we create a hashmap and go over each coefficient and calculate the following linear congruence:

$x \times N = y^2$ mod b_i

For coefficient solution $y = 5$ mod 11 we would get:

$x \times 4387 = 5^2$ mod 11

=> $x = 4$

We save the x solution as key in the hashmap and the coefficient solution as value:

mod 3 = 1 : { 1, 2 }
mod 5 = 2 : { 2, 3 }
mod 7 = 0 : { 0 }, 3 : { 1, 6 }
mod 11 = 5 : { 1, 10 }, 9 : { 2, 9 }, 4 : { 5, 6 }
mod 13 = 11 : { 1, 12 }, 2 : { 5, 8 }, 6 : { 6, 7 }

Next we iterate sieve interval i from 0 to i_{n-1}

For example when $i = 97$ we check if 97 mod 3, 97 mod 5, .., is a key in the hashmap and we collect the results.

97 mod 3 = 1 : { 1, 2 }
97 mod 5 = 2 : { 2, 3 }
97 mod 7 = /
97 mod 11 = 9 : { 2, 9 }
97 mod 13 = 6 : { 6, 7 }

At $i = 97$ we found results in 4 out 5 elements in the coefficient solution set for factor base b.
We multiply the moduli together for which we found a result:

$3 \times 5 \times 11 \times 13 = 2145$.

And if this is bigger then $\sqrt{i \times N}$ we continue.

Next we calculate the partial results for the coefficient solutions we just collected (Chapter IV):

y mod 3 = { 715, 1430 }
y mod 5 = { 1287, 858 }
y mod 11 = { 585, 1560 }
y mod 13 = { 825, 1320 }

Next we generate combinations mod 2145 choosing at most one partial result per modulus.

For example: $715 + 1287 + 585 + 825 = 1267 \bmod 2145$

Lets call this the coefficient candidate y.

The useful thing about this setup is that if we now calculate $y^2 - i \times N$ we know the result will be divisible by the moduli from which we collected the partial results.

Hence $1267^2 - 97 \times 4387 = 1179750$ $(2 \times 3 \times 5^3 \times 11^2 \times 13)$

All but one of the factors are in the factor base (2) in this example.

The closer y^2 is to $i \times N$, the smaller the smooth candidate will be.
Once enough such smooth candidates are found, you finish the rest of the algorithm using the default Quadratic Sieve proceedings.
Which I won't reiterate as this is widely documented. But in short you would use Gaussian Elimination[10] or Block Lanczos[11] to find a combination of smooths that can be multiplied together to form a square relation on both sides of the congruence mod N.

VI. Quadratic coefficients

In the above, we generate possible linear coefficients y and then square them. By subtracting $i \times N$, which I shall henceforth refer to as simply iN, we can then predict at-least some of the factors of the smooth candidates.

Let us explore how this iN value relates to the coefficient of the quadratic term, an important subject we have not touched on yet.

For example if we have $y_0 = 148$ we see that $148^2 - 4 \times 4387 = 66^2$ ($i = 4$ thus $iN = 4 \times 4387$) satisfies our square relation mod N.
Also note that $148^2 - 4 \times 4387 = 66^2$ is the formula for the quadratic discriminant, which makes sense.
We can see where $i = 4$ comes from when we subtract and add both linear coefficients from each-other and look at the factorization of the result:

$148 - 66 = 82$ (41×2)
$148 + 66 = 214$ (107×2)

When subtracting we get 2 times the lower factor and when adding we get 2 times the upper factor.
We multiply the factors we found, excluding the factors of N and we get: $2 \times 2 = 4$.

Another example when $y_1 = 3$ and $y_0 = 1602$:

$1602^2 = 3^2 \bmod N$
$1602 - 3 = 1599$ (41×39)
$1602 + 3 = 1605$ (107×15)

Hence we get $i = 39 \times 15 = 585$ and verifying this:

$1602^2 – 4387 \times 585 = 3^2$

And when $y_1 = 1$ and $y_0 = 534$:

$1^2 = 534^2 \bmod N$
$534 – 1 = 41 \times 13$
$534 + 1 = 107 \times 5$

Hence we get $i = 13 \times 5 = 65$ and verifying this:

$534^2 – 4387 \times 65 = 1^2$

This i value is actually the coefficient for the quadratic term (you may recognize this in the discriminant formula).
In the example that $y_1 = 1$ and $y_0 = 534$ we have a i value of 13×5.
We know from the above explanation in the first chapters that the roots represent the factors of N.

We can see that the following holds:

$13 \times (–41)^2 – 1 \times (–41) + 41 = 0 \bmod 4387$
$5 \times (–107)^2 + 1 \times (–107) – 107 = 0 \bmod 4387$

I will define the quadratic coefficient as z.

However, in the above example we note that if we have an odd linear coefficient we get quadratics of the shape: $zx^2 + yx + x$.
Because the quadratic for even coefficients is simpler ($zx^2 + yx$) we shall restrict ourselves to these alone.

Thus working with even coefficient if we have $y_1 = 2$ and $y_0 = 1068$, we see that:

$1068 – 2 = 26 \times 41$
$1068 + 2 = 10 \times 107$

Since we are now working with two even coefficients we get a quadratic of the shape: $zx^2 + yx$

This means the quadratic coefficients become 26/2 and 10/2 (since when we take the derivative, it gets multiplied by 2 from the quadratic exponent).

$13 \times (–41)^2 – 2 \times (–41) = 5 \times 4387$ or $0 \bmod 4387$
$5 \times (–107)^2 + 2 \times (–107) = 13 \times 4387$ or $0 \bmod 4387$

And the derivative reveals the other linear coefficient:

$26 \times (–41) – 2= –1068$

And the quadratic for $y_0 = 1068$:

$13 \times (–41)^2 + 1068 \times (–41) = –5 \times 4387$ or $0 \bmod 4387$
$5 \times (–107)^2 + 1068 \times (–107) = –13 \times 4387$ or $0 \bmod 4387$

Our quadratic with the addition of quadratic coefficients is now: $zx^2 + y_0x + N = zx^2 + y_1x – N$

VII. Lifting coefficients

As we build up to the final algorithm, another important step must first be explained. This is the lifting of our coefficients mod p^e, where e is the exponent.
This is very straight forward, so I will just show you a single example.
Let us say we want to lift $y_0 = 5$ and $z = 1 \bmod 11$ to mod 11^2 with $N = 4387$.

We will only focus on lifting solutions where the resulting y_1 is equal to 0 mod p^e.

The very first calculation we must perform is figuring out the root for our polynomial:

$zx^2 + y_0x + N = 0 \bmod 11$

Plugging in $z = 1$ and $y_0 = 5$:

$1 \times x^2 + 5x + N = 0 \bmod 11$

The procedure is as following:

We multiply y_0 by the inverse of z mod p and then multiply that result by the inverse of 2 mod p.
Inverse of z = 1 is 1. So nothing changes. Inverse of 2 mod 11 is 6 so we get:

$5 \times 6 = 8 \bmod 11$.

And plugging in this root we see that the result is correct:

$1 \times 8^2 – 5 \times 8 + 4387 = 0 \bmod 11$ (note: remember that we need to use negative roots, hence flipping the sign for the linear term)

If we have our statement: $zx^2 + y_0x + N = zx^2 + y_1x - N$
Then any y_0 and y_1 pairings must also share a common root.

Calculating the derivative to get y_1 will yield a 0 solution mod 11 as expected:

$y_1 = 2 \times 1 \times 8 - 5 = 0 \bmod 11$

And the polynomial for y_1:

$1 \times 8^2 - 0 \times 8 - 4387 = 0 \bmod 11$

If we now set $y_1 = 0 \bmod 11^2$, then finding the new root is simply a matter of finding $1 \times 8 + 11 \times v \bmod 11^2$

We can apply Hensel's lifting lemma for this.

$x = 8$
$s = (2x)^{-1} = 9 \bmod 11$
$t = (zN - x^2) / 11 = 8 \bmod 11$
$v = t \times s = 6 \bmod 11$
lifted $x = z^{-1} \times (x + v \times 11) = 74 \bmod 11^2$

And our root for modulo 11^2 thus becomes $x = 74$.
We can now take the derivative to generate a new y_0:

$y_0 = 2 \times 1 \times 74 + 0 = 148 \bmod 11^2$

And verifying the polynomial for $y_0 = 148$:

$1 \times 74^2 - 148 \times 74 + 4387 = 0 \bmod 11^2$

Hence lifting linear coefficients is very straight forward. The same principles would apply to lifting quadratic coefficients.

VIII. Improved Quadratic Sieve

We ended the previous chapter having found the following root and linear coefficient mod 11^2

$1 \times 74^2 + 148 \times (-74) + 4387 = 0 \bmod 11^2$

Setting y_1 to 0 mod 11^2 and calculating the quadratic polynomial yields the following result in the integers:

$1 \times 74^2 + 0 \times (-74) - 4387 = 33^2$

Since the root, 74^2 and polynomial output, 33^2 is square we can take the GCD and we find the factorization of 4387.
Let us for now assume we always set y_1 to 0. In this case we can drop the linear term and shorten the polynomial for y_1 to $zx^2 - N$.
The condition that must be true to have a valid square relation in the integers is that the output of the quadratic $zx^2 - N$ must be square, and z must also be square, because then we can take the square root of z and multiply x with this square root to produce $x^2 - N$. We can get this to work as an algorithm by producing smooth candidates with $zx^2 - N$ and performing trail factorization on both z and the polynomial output and feeding this into Gaussian elimination over gf(2). While using this quadratic coefficient z as a linear multiplier to a quadratic root gives greater control over the size of produced smooth candidates, we could additionally add the modulus to the linear coefficient to achieve even greater control. Getting a quadratic sieve type of algorithm to work using quadratics of the shape $zx^2 - N$ is trivial as described above. The goal is now to get it to work with $zx^2 + y_1x - N$ as well as this would grant near absolute control over the size of produced smooth candidates.

Previously we found the following two quadratics, the first one with $y_0 = 148$ and the second one with $y_1 = 0$.

$74^2 + 148 \times (-74) + 4387 = -(33)^2$
$74^2 + 0 \times (-74) - 4387 = 33^2$

Since both produce the same output we can rearrange them like this: $74^2 + 148 \times (-74) + 4387 = 74^2 + 0 \times (-74) - 4387$

Eureka! If our quadratic polynomial for y_0 produces a square, and our quadratic polynomial for y_1 produces that same square we have a valid square relation in the integers, and can proceed to taking the GCD. We also see that the quadratic for y_0 has a non-zero linear coefficient. Hence, this gives us a clue as to how we can achieve a sieving setup with non-zero linear coefficients.

To do: Finish this

IX. The quest for a polynomial time algorithm

A Quadratic Sieve style algorithm is unlikely to ever achieve polynomial time speeds. However, we should never stop trying to find a polynomial time algorithm specifically for the factorization problem as this would allow us to also make progress toward $P = NP$. One of the most famous open math problems in computer science.

I have quickly created a short python script which will iterate odd quadratic coefficient for a given N, p and q and prints the linear coefficients that solve the quadratic for 0 in the integers:

Code: https://github.com/BigPolarBear1/factorization/blob/main/debug.py

If N = 4387 and p = 41 and q = 107 the code would print the following for any odd quadratic coefficient z up to 50:

```
1*41^2-148*41+4387*1 = 0 ( x=41 z=1 y0=148 y1=66 )
3*41^2-337*41+4387*2 = 0 ( x=41 z=3 y0=337 y1=91 )
5*41^2-419*41+4387*2 = 0 ( x=41 z=5 y0=419 y1=9 )
7*41^2-608*41+4387*3 = 0 ( x=41 z=7 y0=608 y1=34 )
9*41^2-797*41+4387*4 = 0 ( x=41 z=9 y0=797 y1=59 )
11*41^2-986*41+4387*5 = 0 ( x=41 z=11 y0=986 y1=84 )
13*41^2-1068*41+4387*5 = 0 ( x=41 z=13 y0=1068 y1=2 )
15*41^2-1257*41+4387*6 = 0 ( x=41 z=15 y0=1257 y1=27 )
17*41^2-1446*41+4387*7 = 0 ( x=41 z=17 y0=1446 y1=52 )
19*41^2-1635*41+4387*8 = 0 ( x=41 z=19 y0=1635 y1=77 )
21*41^2-1824*41+4387*9 = 0 ( x=41 z=21 y0=1824 y1=102 )
23*41^2-1906*41+4387*9 = 0 ( x=41 z=23 y0=1906 y1=20 )
25*41^2-2095*41+4387*10 = 0 ( x=41 z=25 y0=2095 y1=45 )
27*41^2-2284*41+4387*11 = 0 ( x=41 z=27 y0=2284 y1=70 )
29*41^2-2473*41+4387*12 = 0 ( x=41 z=29 y0=2473 y1=95 )
31*41^2-2555*41+4387*12 = 0 ( x=41 z=31 y0=2555 y1=13 )
33*41^2-2744*41+4387*13 = 0 ( x=41 z=33 y0=2744 y1=38 )
35*41^2-2933*41+4387*14 = 0 ( x=41 z=35 y0=2933 y1=63 )
37*41^2-3122*41+4387*15 = 0 ( x=41 z=37 y0=3122 y1=88 )
39*41^2-3204*41+4387*15 = 0 ( x=41 z=39 y0=3204 y1=6 )
41*41^2-3393*41+4387*16 = 0 ( x=41 z=41 y0=3393 y1=31 )
43*41^2-3582*41+4387*17 = 0 ( x=41 z=43 y0=3582 y1=56 )
45*41^2-3771*41+4387*18 = 0 ( x=41 z=45 y0=3771 y1=81 )
47*41^2-3960*41+4387*19 = 0 ( x=41 z=47 y0=3960 y1=106 )
49*41^2-4042*41+4387*19 = 0 ( x=41 z=49 y0=4042 y1=24 )
```

Of-course we don't know p nor q. So both root and linear coefficient would be unknown to us. But we can solve mod m. The trick then becomes solving this system of quadratics for 0 for any mod m (aka, solve it in the integers).

Figuring out a way to solve this system of quadratics in polynomial time, this will be my life's quest, and I won't ever quit until I succeed. And perhaps this is a fool's quest, but I would rather throw away my life chasing the impossible then never to have tried at all. Factorization must fall, no matter what.

X. Conclusion

Note to self: rewrite this chapter once paper is finished.

We have managed to make many reductions in complexity in the factorization problem. Where traditionally modern variants of Fermat's factorization method, which includes Number Field sieve and Quadratic sieve, had to find smooth numbers within a factor base, to then hopefully complete a square relation mod N. We have now found a way to instead enumerate the roots of the squares in this square relation by calculating coefficients of the linear term of a quadratic mod p_i. This is done in an attempt to break the almost 40-year stalemate in factorization algorithms. I would now encourage the reader, to continue this work, as this is merely the beginning, and many unknown lands of modular magic and polynomials lay ahead of us to explore.

XI. References

1. Number Field Sieve: https://en.wikipedia.org/wiki/General_number_field_sieve
2. RSA algorithm: https://en.wikipedia.org/wiki/RSA_cryptosystem
3. Euclidean algorithm: https://en.wikipedia.org/wiki/Euclidean_algorithm
4. Fermat's factorization method: https://en.wikipedia.org/wiki/Fermat's_factorization_method
5. Quadratic sieve: https://en.wikipedia.org/wiki/Quadratic_sieve
6. Carl Pomerance: https://en.wikipedia.org/wiki/Carl_Pomerance
7. Subset-sum problem: https://en.wikipedia.org/wiki/Subset_sum_problem
8. Legendre Symbol: https://en.wikipedia.org/wiki/Legendre_symbol
9. LLL algorithm: https://en.wikipedia.org/wiki/Lenstra–Lenstra–Lovász_lattice_basis_reduction_algorithm
10. Gaussian Elimination: https://en.wikipedia.org/wiki/Gaussian_elimination
11. Block Lanczos: https://en.wikipedia.org/wiki/Block_Lanczos_algorithm