RSA Project paper

RSA Group
Li Qiu
Yiwei
Leester Mei
Akeem

November 27, 2016

1 Quadratic Sieve

Quadratic Sieve method is actually discovered step by step. It is a optimization of Dixon's Factorization Method. So, to learn what is Quadratic Sieve, we have to understand Dixon's Method. And Dixon's Method is also related to Fermat's Factorization and Kraitchik's Factorization Method. Here, I will briefly introduce these three factorization methods.

1.1 Brief Hierarchies Of Quadratic Sieve

1.1.1 Fermat's Factorization

Fermat factorization method is very straight forward, if we are going to factor n, since:

$$n = ab \Rightarrow \left[\frac{1}{2}(a+b)\right]^2 - \left[\frac{1}{2}(a-b)\right]^2$$

let

$$x = \frac{1}{2}(a+b),\ y = \frac{1}{2}(a-b) \Rightarrow n = x^2 - y^2$$

Therefore, we just need to find a x satisfy:

$$Q(x) = y^2 = x^2 - n$$

and obviously, the smallest x is $\lceil \sqrt[2]{n} \rceil$ and if we can find a $Q(x)$ is a square root, then we can find the factors of n which is $a = x + y, b = x - y$

1.1.2 Kraitchik's Factorization

Kraitchik's method is instead of checking $x^2 - n$ is a square, he suggests to check $x^2 - kn$ a square number, which is equivalent to find $y^2 \equiv x^2 \pmod{n}$. And the only interesting solution is $x \not\equiv \pm y \pmod{n}$. Besides this, instead of seeking one $x^2 - n$ is square, he was looking for a set of number $\{x_1, x_2, \ldots, x_k\}$ such that $y^2 \equiv \prod_{i=1}^{k}(x_i^2 - n) \equiv \prod_{i=1}^{k} x_i^2 \equiv \left(\prod_{i=1}^{k} x_i\right)^2 \pmod{n}$ is square. if he can find a relation like this, then, the factors of n is $gcd(|y \pm \prod_{i=1}^{k} x_i|, n)$

1.1.3 Dixon's Factorization

One of the great improvement of Dixon's method comparing to the method before is that he replace the requirement from "is a square of an integer" to "has only small prime factors". To explain this we need introduce 2 concepts which are **Factor Base** and **Smooth Number**.

Factor Base is a set of prime factors $S_{fb} = \{p | p \leq B\}$ where B is some integer.

Smooth Number is a integer that all its prime factor within the **Factor Base** which means if we choose integer B and Q, $Q = \prod_{i=1}^{k} p_i^{a_i}$ where $a_i, k \in \mathbb{Z}, p_i \in S_{fb}$. We called Q is a **B-Smooth** Number.

Recall Kraitchik's Method, he suggests to find a relation $y^2 \equiv \left(\prod_{i=1}^{k} x_i\right)^2 \pmod{n}$. But Dixon's idea is different, he is looking for the relation of $y^2 \equiv Q \equiv \prod_{i=1}^{k} p_i^{a_i} \pmod{n}$ where $a_i \in \mathbb{Z},\ k = |S_{fb}|,\ p_i \in S_{fb}$. For each relation is found, it can represent as a exponent vector $\vec{v} = \{a_1, a_2, \ldots, a_k\}$ over $\mathbb{F}_2$, and after finding k relations, we have a matrix $m = \{\vec{v_i} | i \leq k\}$. And searching the null space of this matrix, could help us to find the square integer. Since $M\vec{v} = 0$ where $\vec{v} \in Null\ Space$, we know which rows sum together are equal to 0, which also implies the products of corresponding Q to that row is a square integer. After found the relation, let $x^2 = \prod_i^k Q_i$, the factor of n is $gcd(y \pm x, n)$

1.2 Math and Algorithm in Quadratic Sieve

1.2.1 Legendre Symbol & Laws of Quadratic Reciprocity

From now, we know two ways to test whether n is a quadratic residue mod p, one is Euler Criterion, and the other is Legendre symbol. In practice, probably because we factor base is not insanely large, we did not noticed how much faster does Legendre symbol than the Euler criterion (since Euler criterion is an $\mathcal{O}(\log n)$ algorithm).

1.2.2 Shanks Tonelli's Algorithm

Shanks Tonelli's Algorithm is a faster algorithm to find the root of a quadratic residue. In class, we only learn the simple case, which is when $p \equiv 3 \pmod{4}$. Since this algorithm is very important for quadratic sieve, we are going to prove the algorithm.

Given $x^2 \equiv n \pmod{p}$, we want to find x.

let $p - 1 = 2^e S,\ x \equiv n^{\frac{S+1}{2}} \pmod{p},\ t \equiv n^S \pmod{p}$.

we have, $x^2 \equiv n^{S+1} \equiv n^S n \equiv tn \pmod{p}$, notice that,

if $t \equiv 1 \pmod{p}$, our $x = \pm n^{\frac{S+1}{2}} \pmod{p}$

if $t \not\equiv 1 \pmod{p}$, then find a quadratic non-residue a and let $b \equiv a^S \pmod{p}$, then, $b^{2^e} \equiv (a^S)^{2^e} \equiv a^{2^e S} \equiv a^{p-1} \equiv 1 \pmod{p}$, Since we know a is a quadratic non-residue, By Euler Criterion, $a^{\frac{p-1}{2}} \equiv b^{2^{e-1}} \equiv -1 \pmod{p} \rightarrow 2^e$ is the order of b.

we have $t^{2^e} \equiv 1 \pmod{p}$, and we let $2^{e'}$ be the order of $t \pmod{p}$, since n is a quadratic residue, $e' \leq e - 1$

let $c \equiv b^{2^{e-e'-1}} \pmod{p}$, $b' \equiv c^2$, $t' = tb'$, $x' = cx$, after this construction, $x'^2 \equiv t'n \pmod{p}$ still holds, since $x'^2 \equiv b^{2^{e-e'}} x^2 \equiv tnb^{2^{e-e'}} \equiv tnb' \equiv t'n \pmod{p}$. And we can repeat this process until $e' = 0$ we can find a $t' = 1$, and our final solution is $\pm x' \pmod{p}$

1.2.3 Logarithm Approximation

Logarithm approximation plays a very import role in implementing the sieving process. Since we use the polynomial $Q(x) = (x)^2 - n$, and we want to sieving in the interval $[\lfloor\sqrt{n}\rfloor - M, \lfloor\sqrt{n}\rfloor + M]$, therefore, $\log(Q(x)) \approx \log(Q(\lfloor\sqrt{n}\rfloor + M)) = \log(2M\sqrt{n} + M^2)$, because of M^2 is trivial if $\sqrt{n}$ is huge, $\log(Q(x)) \approx 0.5 \log n + logM$. We also know $Q(x) = \prod_{i=1}^{k} p_i^{a_i}$ where $p_i \in S_{fb}, k = |S_{fb}|$ if $Q(x)$ has a smooth relationship in our factor base, then we have, $\log(Q(x)) = \log \prod_{i=1}^{k} p_i^{a_i} = \sum_{i=1}^{k} a_i \log p_i$. This formula tell us if we can find a $\log Q(x) = \sum_{i=1}^{k} a_i \log p_i \approx 0.5 \log n + logM$ it probably a candidates of smooth number.

1.3 Sieving In Quadratic Sieve

Quadratic Sieve's sieving method is inspired by the ancient Eratosthenes Sieve. The Eratosthenes sieve is to crossing out the multiples of prime, and the leftovers are the candidates of prime. In quadratic sieve, we want a relation $x^2 \equiv n \equiv r \pmod{p}$ where $r < p$, in order to keep this relation, we marking the multiples of p plus r as our candidates.

1.3.1 Pre-Sieving

Before sieving we need to build our factor base, and in order to build our factor base, we need to choose a number B which all the prime in our factor base must less than or equal to B. If B is too small, it will be very hard to find such smooth relations, but if B is too large, we will be facing a huge matrix and finding the null space of it will be very time consuming. And our B choice is one million (we will explain why later). After choose B, we have another criteria for our factor base, that is the prime factor p in our factor base must make our N(the number we want to factorize) to be a quadratic residue.

1.3.2 Sieving

For each prime factor in factor base, we use shank-tonelli's algorithm to find the roots(which means the x) of $x^2 \equiv N \pmod{p}$, if $p = 2$, the root can only be 1, and for every other odd prime factor we have two roots which is x and $p - x$, we save the position of $x + i * p$ and $p - x + i * p$ where $i \in \{0, 1, \ldots\}$ until $x + i * p > M$ where M is our bound of searching space. And at the meantime, accumulate $\log p$ to the position in our sieve array.

1.3.3 Saving the result

During the sieving stage, when we found a position and it's value in our sieve array is close to $0.5 \log n + logM$ we consider it is our smooth candidate. But to define what is close to, we need a threshold. To determine value of the threshold little articles we read mentioned about it, some of articles said that the value of this threshold is a small error. We made a bunch of tests, and we found when the threshold is less than 8 it seems accurate(we will explain the detail of the tests later). And we also found that the threshold is good to be small, because we do not need find all the smooth relations in this range, we just need to find the relation as fast as possible. Since if the threshold is large, we need to double check the relation, but if the threshold is small, we do not have to.

1.4 Linear Algebra In Quadratic Sieve

If we use logarithm approximation, it is very likely to find partial relations instead of full relation. A full relation means the relation is exactly the smooth number. A partial relation means that there are factors out of our factor base. A relation like this is also useful, since we are working over $\mathbb{F}_2$, if we can find another partial relation which have the leftover in common, the leftover will be canceled. In other words, if we can find two partial relations have the same leftover factors we can treat them as two full relations.

After we collecting enough relations, we use them to create a exponent matrix. The common way to do this, is use trial division. For each relation we divide out all its prime factor in the factor base, and use the powers mod 2 to build a exponent vector. Then, if we can find k relations where $k = |S_{fb}|$, we have k exponent vectors to form a exponent matrix.

Since in our approach, $|S_{fb}| \approx 40000$, we can simply use Gaussian elimination to find null space. But for larger matrix, something like 100000x100000 it is better to use Block Lanczos algorithm. By the way, the Meataxe is using Gaussian Elimination to find null space as well.

1.5 Details of Our implementation & optimization

I *Choosing B:*

1.6 Problems we met & the solutions

1.7 Tables and Graphs of Experiments