TD3 : AN INTEGER FACTORING ALGORITHM

There is no known polynomial time algorithm to factor integers, not even a probabilistic one. We present in this text an algorithm which, although exponential time, is more efficient than trial division.

1. Birthday paradox

Assume there are 40 students in a classroom. The probability that two among them have the same birthday (assuming none of them was born on February 29th) is

$$1-(1-\frac{1}{365})(1-\frac{2}{365})\cdots(1-\frac{39}{365})\geqslant 0.89$$

This is rather close to 1 although 40 is much less than $365\ldots$

We look for a conceptual explanation of this phenomenon.

Assume we have p balls in a bag, each tagged wich a figure from 1 to p. We draw a ball at random and put it back in the bag. We iterate $n\geqslant 2$ times. We now estimate the probability $P(p,n)$ for the n drawn balls to be pairwise distinct :

$$P(p,n)=\prod_{1\leqslant i\leqslant n-1}\left(1-\frac{i}{p}\right)\leqslant\prod_{1\leqslant i\leqslant n-1}\exp(-\frac{i}{p})=\exp(-\frac{n(n-1)}{2p})\leqslant\exp(-\frac{(n-1)^2}{2p})$$

So if n is greater than $1+\sqrt{p}$ the probability that the same ball has been drawn more than once is at least $1-\exp(-1/2)>0.39$.

If the number of draw is proportional to the square root of the number of balls there is a significant probability to have drawn twice the same ball in the end.

2. Random maps from a finite set to itself

Let F be a finite set with p elements. Let $\mathcal{A}(F)$ be the set of maps from F to F. Consider the uniform probability measure on $\mathcal{A}(F)$. Fix an element O in F.

To every map $f:F\rightarrow F$ we associate the sequence $x_0=O$, $x_{i+1}=f(x_i)$ obtained by iterating f. This is an ultimately periodic sequence : there exist two integers $\pi_f\geqslant 1$ and $\mu_f\geqslant 0$ such that if $k\geqslant\mu_f$ then $x_{k+\pi_f}=x_k$. The smallest such π_f is called the period and the smallest such μ_f is called the preperiod.

The sum $\rho_f=\mu_f+\pi_f$ is a random variable on $\mathcal{A}(F)$. The probability of the event $\rho_f\geqslant n$ is $P(p,n)$. Could you explain why ?

The expectation of $E(\rho_f)$ satisfies

$$
\begin{aligned}
E(\rho_f) &= 1 + \sum_{n \geqslant 2} P(p,n) \leqslant \sum_{n \geqslant 0} \exp\left(-\frac{n^2}{2p}\right) \leqslant 1 + \int_0^{\infty} e^{-\frac{x^2}{2p}} dx \\
&= 1 + \sqrt{2p} \int_0^{\infty} e^{-x^2} dx = 1 + \sqrt{\frac{p\pi}{2}}
\end{aligned}
$$

car $\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi}$.

3. Two simple factoring algorithms

Remind there exists a polynomial time algorithm for primality testing. So factoring integers reduces to the following problem : on input a composite integer N find a non-trivial divisor M of N.

Indeed, if the factors M and $R = N/M$ are not prime, we rerun the algorithm with M and R.

An algorithm that finds a non-trivial factor to a given composite integer is called a **breaking** algorithm. So factoring reduces to iteratively breaking integers.

The simplest factoring algorithm is trial division. To factor N, compute the euclidean division of N by $r = 2, 3, 5, 7, 9, 11, 13, 15$ etc.

If N is composite you will find a factor $r \leqslant \sqrt{N}$.

The complexity of trial division is $O(N^{\frac{1}{2}+o(1)})$. This is poorly efficient but useful for small integers.

Pollard has invented an elegant but heuristic method with a better complexity.

We assume to simplify that $N = pq$ is the product of two distinct primes. We choose a polynomial f with integer coefficients (one often takes $f(X) = X^2 + 1$) and we consider the sequence with values in $\mathbb{Z}/N\mathbb{Z}$ defined by $x = x_0$ any element in $\mathbb{Z}/N\mathbb{Z}$ and $x_{k+1} = f(x_k) \bmod N$.

Since $f(X)$ is a polynomial, the map

$$
f_N : \quad \mathbb{Z}/N\mathbb{Z} \longrightarrow \mathbb{Z}/N\mathbb{Z}
$$

$$
x \bmod N \longmapsto f(x) \bmod N
$$

is the set-theoretical *product* of the two maps

$$
f_p : \quad \mathbb{Z}/p\mathbb{Z} \longrightarrow \mathbb{Z}/p\mathbb{Z}
$$

$$
x \bmod p \longmapsto f(x) \bmod p
$$

and

$$f_q : \quad \mathbb{Z}/q\mathbb{Z} \longrightarrow \mathbb{Z}/q\mathbb{Z}$$

$$x \bmod q \longmapsto f(x) \bmod q$$

More precisely we call γ the Chinese isomorphism

$$\gamma : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z}$$

and we check that $\gamma \circ f_N = (f_p \times f_q) \circ \gamma$.

Assume now that the maps f_p and f_q behave like two independent random maps. In other words assume that f_p follows a uniform probability law in the set $\mathcal{A}(\mathbb{Z}/p\mathbb{Z})$ of maps from $\mathbb{Z}/p\mathbb{Z}$ to itself. Assume also that f_q follows a uniform probability law and that the random variables f_p and f_q are independant.

This is a somewhat crazy assumption since $f = x^2 + 1$ is anything but random ...

Let $y_k = x_k \bmod p$ be the class of x_k modulo p. Let $z_k = x_k \bmod q$ be the class of x_k modulo q. One checks that $y_{k+1} = f_p(y_k)$ and $z_{k+1} = f_q(z_k)$. The Chinese isomorphism γ sends x_k onto (y_k, z_k).

Let π_p and μ_p be the period and preperiod of f_p. Let π_q and μ_q be the period and preperiod of f_q. We have good reasons to expect that π_p and μ_p (that are functions of f and p) are $O(\sqrt{p})$. We also expect that π_q and μ_q are $O(\sqrt{q})$. So we have an iterated sequence in $\mathbb{Z}/N\mathbb{Z}$ whose component modulo p (resp. q) has preperiod and period $O(\sqrt{p})$ (resp. $O(\sqrt{q})$).

If k is large enough we thus expect that

$$gcd(x_k - x_{k+\pi_p}, N) = p$$

which exhibits a non-trivial factor of N. This is of little help in this form because we do not know π_p. However for k large enough and a multiple of π_p (but not a multiple of π_q) we have

$$gcd(x_k - x_{2k}, N) = p.$$

Pollard's algorithm computes iteratively $x_k = f(x_{k-1})$ and $X_k = x_{2k} = f(f(X_{k-1}))$ and the above gcd for $k = 0, 1, 2, ...$, until a non-trivial factor of N shows up.

Heuristically this method finds a non-trivial factor p in time $O(p^{\frac{1}{2}+o(1)})$ that is $O(N^{\frac{1}{4}+o(1)})$.

4. Questions

(1) Give a cryptographic scheme whose security relies on the difficulty of factoring integers and try to quantify the effect of Pollard's algorithm on the security of this protocol. What would be a reasonable key length to resist such an attack ?

(2) Try to justify the computation of the expectation of ρ_f.

(3) Study the integral $\int_{-\infty}^{\infty} e^{-x^2} dx$, either by computing a numerical approximation or by proving that is is equal to $\sqrt{\pi}$.

You may want to prove that $\left(\int_{-\infty}^{\infty} e^{-x^2} dx\right)^2 = \pi$. To this end write

$$\left(\int_{-\infty}^{\infty} e^{-x^2} dx\right)^2 = \int_{-\infty}^{\infty} e^{-x^2} dx \int_{-\infty}^{\infty} e^{-y^2} dy = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-x^2-y^2} dxdy$$

and introduce polar coordinates $r = \sqrt{x^2+y^2}$ and the angle θ.

(4) Discuss the relevance of various choices for the polynomial $f(X)$. You may wonder if $f(X) = X^2$ is better or worse than $f(X) = X^2+1$. Considering that $(x+1/x)^2 = x^2+1/x^2+2$ what can you say about choosing $f(X) = X^2-2$?

(5) You may try to express the period π_N and preperiod μ_N of $f_N : \mathbb{Z}/N\mathbb{Z} \to \mathbb{Z}/N\mathbb{Z}$ as functions of π_p, μ_p, π_q, μ_q.

(6) Explain why Pollard's algorithm is very good at finding small prime divisors of large integers and illustrate this method with a simple implementation.