# Factoring methods

We would like to factor $N = pq$, which breaks the RSA cryptosystem. We present two methods that both rely on the creation of $x \neq y$ such that $x^2 = y^2 \mod N$ (in fact several other methods proceed by solving this equation). If we solve that equation, we have

$$x^2 - y^2 = 0 \bmod N$$

$$\iff (x-y)(x+y) = 0 \bmod N$$

$$\iff N|(x-y)(x+y)$$

If $N = pq$, $\gcd(x-y, N) =$ p or q or 1 or N. (good cases: p or q. Bad cases: 1 or N). It is only probabilistic, but with constant probability it yields a non-trivial prime factor.

## 1 Polland's Rho method

One way to solve our equation is to look for $x, y$ is to try random values of $P(x) = ax^2 + b$ for some fixed $a, b$. Indeed, if $P(x) = P(y) \bmod N$, then $x^2 = y^2 \bmod N$. We can even do better: let $p|N$, if $P(x) = P(y) \bmod p$, then $p|(x-y)(x+y)$ and $gcd(N, x-y) = p$ (or $gcd(N, x+y) = p$). As there are less values $P(x) \bmod p$ than $P(x) \bmod N$, we expect this search to go faster. The only problem is to check if $P(x) = P(y) \bmod p$ (because $p$ is unknown). But it suffices to check if $\gcd(N, x-y)$ is non-trivial (i.e., $\neq 1, N$).

The other challenge is to find these collisions modulo N (or p) efficiently. Drawing lots of $x_i$ at random and checking if $P(x_i) = P(x_j) \mod N$ for some $x_j$ previously drawn (or checking if $gcd(x_i - x_j, N) \neq 1, N$) can be very long.

From now on, we only care about testing collisions modulo $p$ for $p|N$. I.e., we test if $x - y$ and $N$ have non-trivial gcd. We look at the series defined by $x_{i+1} = P(x_i)$ for $P$ of the form $ax^2 + b$. We know that $P(x_i) = P(x_j) \bmod p$ if $gcd(x_i - x_j, N)$ is non-trivial. The series of $P(x_i) \bmod p$ looks like Figure **??**.

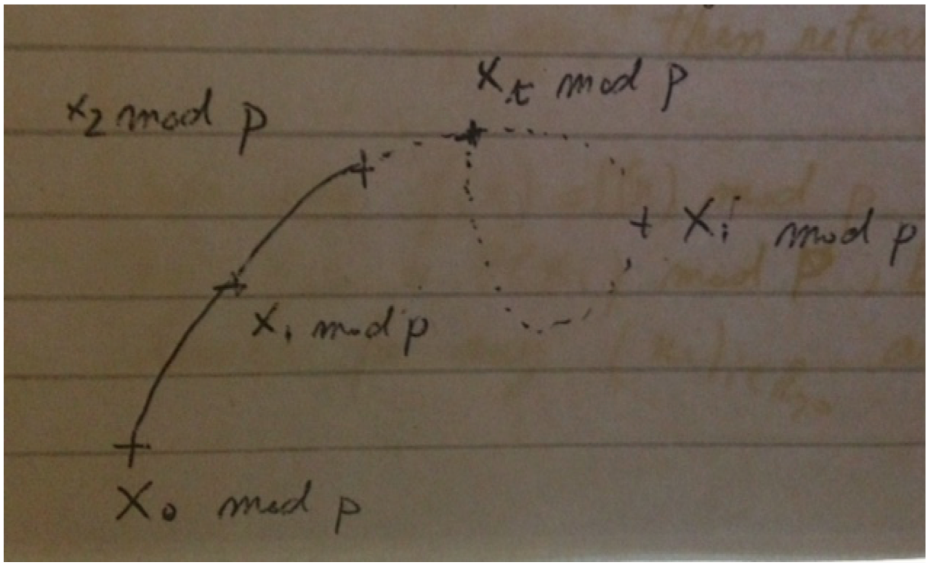
**Definition 1.** *Let $t$ be the smallest index such that there is $j$ with $x_{t+j} = x_t \mod p$. Let $l$ be the smallest index such that $x_{t+l} = x_l$.*

**Floyd's collision finding method** To find a collision $x_i = x_j \mod p$, we use Floyd's collision finding algorithm. Given $n_i, i \in \mathbb{Z}$ (defined by $n_{i+1} = f(n_i)$), and a function $f$ (here $f(n) = n \bmod p$), it returns $i, j$ such that $x_i = x_j$.

We use $f(n) = P(n) \bmod p$ and the series given by $x_{i+1} = P(x_i) \bmod (n_i), i \in \mathbb{Z}$ and function $p$, but Floyd's algorithm works for any $f$. We defined $t$ minimal such that $x_{t+j} = x_t$ for some $j$ and $l$ minimal such that $x_{t+l} = x_l$.

**Proposition 1.** *Floyd's algorithm outputs a collision after less (or exactly) $t + l$ steps.*

Figure 1: Series $P(x_i) \mod p$




**Algorithm 1** Floyd's Algorithm

**Require:** The function $f$ and initial value $n_0$.
**Ensure:** A collision for $f$.

$y_0 \leftarrow x_0$.
**for all** $i$ **do**
  $y_i \leftarrow x_{2i}$
  **if** $f(y_i) = f(x_i)$ **then**
    **return** $(i, 2i)$.
  **end if**
**end for**

*Proof.* Let $j = t - (t \mod l) + l$. The index $j$ has two important properties:

- $j \geq t$ (so $j$ is in the loop).
- $l|j$ and because $2j - j = j$, $l|(2j - j)$, so $x_{2j}$ and $x_j$ are on the same spot in the loop (modulo the length of the loop).

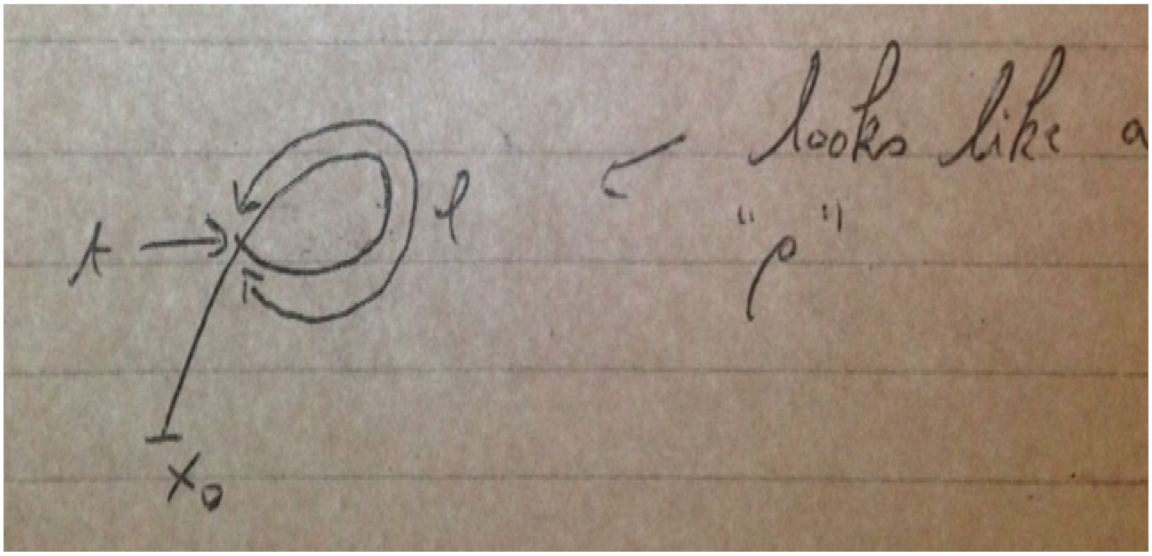
So clearly $y_j = n_j$ and we have a collision. □

**Polland's Rho algorithm** In this case we pick $f(n) = P(n) \bmod p$ for $P(x) = ax^2 + b$. But we don't know $p$ so we don't actually compute $x_i, y_i$, but we can still test if $f(x_i) = f(y_i)$ at each step.

**Example 1.** $N = 8051,\ P(x) = x^2 + 1,\ x_0 = y_0 = 2$.

| $i$ | $x_i$ | $y_i$ | $gcd(\|x_i - y_i\|, 8051)$ |
|---|---|---|---|
| 1 | 5 | 26 | 1 |

Figure 2: Index $t$




**Algorithm 2** Pollard's $\rho$

**Require:** $N$ and a polynomial $P$

**Ensure:** A non-trivial factor of $N$.

$y_0 \leftarrow x_0$.

**for all** $i$ **do**

  $x_{i+1} \leftarrow P(x_i) \bmod N$.

  $y_{i+1} \leftarrow P(P(y_i)) \bmod N$

  **if** $\gcd(x_i - y_i, N)$ is non trivial **then**

    **return** $p = \gcd(x_i - y_i, N)$.

  **end if**

**end for**

| | | | |
|---|---|---|---|
| 2 | 26 | 7474 | 1 |
| 3 | 677 | 871 | 97 |

*So 97 is a non-trivial factor of 8051 (the other being 83).*

## 2 The quadratic sieve

This is another method to find non-trivial solutions of the equation $x^2 = y^2 \bmod N$. Let $P(x) = (a+x)^2 - N$ be a "sieving polynomial." If $y = P(x)$ is a square $B^2$ then $A^2 = B^2 \bmod N$ is a non-trivial solution for

$A = a + x$. But this happens very rarely.

So the way around this is to collect many values $y_i = P(x_i) = (x_i) = (x_i + a)^2 - N$ and to recombine them. Let $(e_i), i \in \mathbb{Z}$, be exponents such that $\prod y_i^{e_i} = B^2$ for some $B$, then:

$$\begin{aligned} B^2 = \prod y_i^{e_i} &= \prod ((x_i + a)^2 - N)^{e_i} \\ &= \prod ((x_i + a)^2)^{e_i} \bmod N \\ &= (\prod (x_i + a)^{e_i})^2 \bmod N \\ &= A^2 \bmod N \end{aligned}$$

for $A = \prod (x_i + a)^{e_i}$.

So our problem really boils down to finding such $e_i$. It is hard to guess them at random, but there is a way to make it work: We only keep the $y_i$ that can be decomposed as a product of primes in a set $\beta$ called the "factor base." Such elements $y_i$ are called "$\beta$-smooth". So each $y_i$ we keep has the form

$$y_i = p_1^{m_i,1} \cdots p_k^{m_i,k}$$

We call the matrix $M = (m_{i,j})$ the "relation matrix". We compute $x \in KerM \mod 2$. For such $x$:

$$\begin{aligned} \prod y_i^{x_i} &= p_i^{\sum x_i m_{i,1}} \cdots p_k^{\sum x_i m_{i,k}} = p_1^{0 mod 2} \cdots p_k^{0 mod 2} = p_1^{2d_1} \cdots p_k^{2d_k} \\ &= (p_1^{d_1} \cdots p_k^{d_k})^2 \text{ for some } d_i \\ &= B^2 \end{aligned}$$

This solves our equation $x^2 = y^2 \mod N$. Before moving on to an example, let us address two issues:

- how to choose $a$.
- how to choose $\beta$.

Choice of $a$: To maximize the chances of $y_i$ being $\beta$-smooth, we make them as small as possible. So $a \approx \sqrt{N}$ and $y_i = 2ax + x^2$.

Choice of $\beta$: Because we want the $p \in \beta$ to divide (at least some of) the $y_i$, we assume that for each $p \in \beta$, there is a $y_i$ with

$$p | y_i, \ so \ N = (a + x_i)^2 \mod p$$

So for $p$ to appear in at least one of the relations, $N$ has to be a quadratic residue modulo $p$ (a square). So we pick p's such that $N$ is a square module $p$. It is characterized by the following theorem.

**Theorem 1.** *$N$ is a square modulo $p > 2$ iff $N^{\frac{p-1}{2}} = 1 \mod p$. $N$ is always a square modulo 2.*

**Example 2.** *Example of the execution of the quadratic sieve: $N = 15347$. We use the sieve polynomial $y(x) = (\lceil \sqrt{N} \rceil + x)^2 - N = (124 + x)^2 - N$.*

*For the factor base, we pick the first 4 primes such that $N$ is a square: $p = 2, 17, 23, 29$.*

*The relation matrix is*

$$M = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

| $x$ | $x+124$ | $y$ | Factorization |
|---|---|---|---|
| 0 | 124 | 29 | $2^0 17^0 23^0 29^1$ |
| 3 | 127 | 782 | $2^1 17^1 23^1 29^0$ |
| 71 | 195 | 22678 | $2^1 17^1 23^1 29^1$ |

$S = [111]$ *is in* $ker(M) \mod 2$.

*Then we have* $124^2 127^2 195^2 = 2^2 17^2 23^2 29^2 \mod N \iff 3070860^2 = 22678^2 \mod N$ *and* $gcd(3070860 - 22670, N) = 103$, *a non-trivial factor of* $N$.