5.1 The Chinese remainder theorem (CRT)

With $N = pq$ and $\text{GCD}(p, q) = 1$, then

$$x = \begin{cases} x_1 \mod p \\ x_2 \mod q \end{cases}$$

has a unique solution modulo $N$ that can be found efficiently (polynomial time with regard to $\log N$). Efficiently solve it for the special cases

$$(x_1, x_2) = (1, 0) \Rightarrow \text{solution } u_1$$
$$(x_1, x_2) = (1, 0) \Rightarrow \text{solution } u_2$$

We can then calculate a general solution $x$

$$x = x_1 u_1 + x_2 u_2 \mod N$$

since

$$x = x_1 u_1 + x_2 u_2 = x_1 \cdot 1 + x_2 \cdot 0 \mod p$$
$$x = x_1 u_1 + x_2 u_2 = x_1 \cdot 0 + x_2 \cdot 1 \mod q$$

We can compute $u_1$ and $u_2$ by running the extended Euclidean algorithm on $p$ and $q$. We get $a$ and $b$ such that

$$1 = \underbrace{ap + bq}_{u_1}$$

since

$$1 = ap + bq \mod p = 0 + bq \mod p \Rightarrow bq = 1 \mod p$$
$$1 = ap + bq \mod q = ap + 0 \mod q \Rightarrow ap = 1 \mod q$$

5.2 Modular division

What is $\frac{2}{3} \mod 7$?

$$3 \cdot \frac{2}{3} = 2 \mod 7$$
$$3 \cdot x = 2 \mod 7$$

We see that $x = 3$ does it!

What is $\frac{2}{3} \mod 6$?

$$3 \cdot x = 2 \mod 6$$
This has no solution and this should not be a surprise. Already in the real numbers we know that \( \frac{2}{0} \) is not defined. It is bad with a zero in the denominator. The Chinese remainder theorem states that modulo 6 is the same as modulo 2 and modulo 3 at the same time and \( \frac{2}{3} \) modulo 6 when we look at it modulo 3, this is \( \frac{2}{0} \).

### 5.3 Efficient modular division

Think of \( \frac{2}{3} \) as \( 2 \cdot \frac{1}{3} \). How do we compute modular inverses?

Use the extended Euclidean algorithm

\[
GCD(p, b) = 1 \Rightarrow 1 = cp + db \Rightarrow d = \frac{1}{b}
\]

### 5.4 Factorization

We want to factor \( N = p \cdot q \) in less than linear time with regard to \( p \) (which is the time complexity for trial division).

#### 5.4.1 Pollard’s \( \rho \) algorithm - magic and simple algorithm

Algorithm:

\[
x_0 = 4711 \\
x_{i+1} = x_i^2 + 1 \mod N
\]

Compute \( GCD(x_{2i} - x_i, N) \) for \( i = 1, 2, \ldots \) until you find a factor \( GCD(x_{2i} - x_i, N) \neq 1 \), the value of \( GCD(...) \) is a factor of \( N \)

CRT: modulo \( N \sim \) modulo \( p \) & modulo \( q \). Squaring is pretty random; \( x_i \mod p \) looks like random numbers until we get a repeat.

\[
x_0 = 4711 \\
x_1 = \text{Some number mod } p \\
\ldots \\
x_{2711} = x_{414} \\
x_{2712} = x_{415}
\]

Iterating \( x_i \mod p \) is equivalent to running around a loop. \( x_{2i} \) is running twice as fast as \( x_i \). When they meet up \( x_{2i} - x_i \) is divisible by \( p \) and \( GCD(x_{2i} - x_i, N) \) contains the factor \( p \) (we are not sure that this is a prime but that can easily be checked).

Heuristic statement: Pollard \( \rho \) finds the factor \( p \) in \( \sim \sqrt{p} \) time. This is based on the heuristic assumption that the \( x_i \) behave like random numbers and the key is to analyze how many random numbers are needed until we get a repeated value.

#### 5.4.2 Collision probability

How many random numbers \( (x_i \mod p) \) are needed to get a repeat?
This is analogous to the birthday problem (what’s the probability of collision of birthdays in a group of a certain size?). The probability of no collision is $e^{-t^2 / 2}$, $t = \#numbers$ and $p$ is the number of possibilities.

$$t \sim \sqrt{2p} \rightarrow P(\text{no collision}) = e^{-1}$$
$$t \sim \sqrt{10p} \rightarrow P(\text{no collision}) = e^{-5}$$

### 5.4.3 Implementation

```plaintext
x = 4711
y = 4711
repeat
    x = x^2 + 1 mod N
    y = y^2 + 1 mod N
    y = y^2 + 1 mod B
    if (GCD(x−y,N) != 1) return GCD(x−y,N)
```

The squaring and modulo calculations are considerably faster than the GCD calculation, thus we want to perform few calls to GCD. We can achieve this by multiplying together a few consecutive $x_{2i} - x_i \mod N$ before calling GCD on the product.

### 5.4.4 General factorization

Find nontrivial solution ($x \neq \pm y$) to $x^2 = y^2 \mod N$.

$N$ divides $x^2 - y^2 = (x - y)(x + y)$ but not either factor. $GCD(N, x - y)$ is a factor of $N$.

First idea: Small numbers are often squares, $\lceil \sqrt{N} \rceil$ ($\lceil \rceil$ means round up to next integer).

Example:

$$N = 21$$
$$\lceil \sqrt{21} \rceil = 5$$
$$5^2 = 25 = 4 = 2^2 \mod 21$$
$$GCD(5 - 2, 21) = 3$$
$$GCD(5 + 2, 21) = 7$$
How large is $\lceil \sqrt{N} \rceil^2 - N$?

$$\lceil \sqrt{N} \rceil - \sqrt{N} \sim \frac{1}{2}$$
$$\frac{d}{d\sqrt{N}} \lceil \sqrt{N} \rceil = 2\sqrt{N}$$
$$\Rightarrow \lceil \sqrt{N} \rceil^2 - N \approx \sqrt{N}^2 + \frac{1}{2} \cdot 2\sqrt{N} = \sqrt{N}$$

In the last step we assume that the ceiling operation adds an average of $\frac{1}{2}$ to $\sqrt{N}$ and substitute $\lceil \sqrt{N} \rceil^2$ with a Taylor expansion.

What is the probability that a number of size $T$ is a perfect square?

There are $\lfloor \sqrt{T} \rfloor$ perfect squares $\leq T$, which means that the probability is $\sim \frac{1}{\sqrt{T}}$.

In our case we have $T \sim \sqrt{N}$, which gives us a probability of $\approx N^{-\frac{1}{4}}$ that $\sqrt{N}$ is a square, and a time complexity of $N^{\frac{1}{4}} \geq \sqrt{p}$ where $p$ is the smallest prime and thus Pollard’s $\rho$ algorithm is better.

Example:

$$N = 161$$
$$\lceil \sqrt{161} \rceil = 13$$
$$13^2 = 169 = 8 \mod 161 \text{ (not a square)}$$
$$\lceil \sqrt{2 \cdot 161} \rceil = 18$$
$$18^2 = 324 = 2 \mod 161 \text{ (not a square)}$$
$$13 \cdot 18 \mod 161 = 73$$
$$GCD(73 - 4, 161) = 7$$
$$GCD(73 + 4, 161) = 13$$

Example:

$$N = 123$$
$$11^2 = 121 = -2 \mod 123$$
$$12^2 = 144 = 21 = 3 \cdot 7 \mod 123$$
$$16^2 = 256 = 10 = 2 \cdot 5 \mod 123$$
$$18^2 = 324 = -45 = -5 \cdot 3^2 \mod 123$$
$$19^2 = 361 = -8 = -2^3 \mod 123$$

We can find squares by combining the above

$$(11 \cdot 19)^2 = -2 \cdot -2^3 = 2^4 = 4^2 \mod 123$$
$$(11 \cdot 16 \cdot 18)^2 = -2 \cdot 2 \cdot 5 \cdot -5 \cdot 3^2 = (2 \cdot 3 \cdot 5)^2 \mod 123$$

5.4.5 Quadric Sieve

Idea: Generate many ($\sim 10^6$) $a_i$ such that $b_i = a_i^2$ are small mod $N$. One good alternative is to use.

$$b_i = (i + \lceil \sqrt{N} \rceil)^2 - N$$

Factor all $b_i$ and combine to form perfect squares. More about this in next lecture.