Die Entwicklung des RSA-Kryptosystems in den 1970er Jahren läutete ein goldenes Zeitalter der Kryptographie ein. Auf einmal war es jeder einfachen Person möglich, unknackbare Verschlüsselung zu verwenden, und das nur durch die Kraft der Mathematik. Vor diesem mächtigen Werkzeug hatte die US-Regierung damals so viel Angst, dass die Algorithmen dafür zwischendurch strengen Exportbeschränkungen unterlagen, mit denen sonst nur Kriegswaffen belegt wurden.

Inzwischen gibt es eine Vielzahl an Kryptosystemen, die auf unterschiedlichen mathematischen Strukturen basieren. Trotzdem ist RSA nach wie vor extrem weit verbreitet, und das Prinzip ist so einfach, dass ihm allein dadurch eine gewisse Schönheit innewohnt. Außerdem gehört es zu der eigenartigen Klasse der asymmetrischen Kryptosysteme (es war sogar das erste seiner Art): Anders als bei traditionellen symmetrischen Kryptosystemen ist es nicht notwendig, dass beide Parteien vorher einen geheimen Schlüssel austauschen.

Es klingt wie Magie, aber mit RSA hat jede Partei stattdessen ihren eigenen öffentlichen Schlüssel. Diese öffentlichen Schlüssel kann man sich gegenseitig senden und danach komplett geheim kommunizieren, selbst wenn man sich vorher nie getroffen hat und ein Gegenspieler die komplette Kommunikation mitliest.

Diese Magie basiert darauf, dass es keine bekannte Methode gibt, auch nur halbwegs effizient die n-te Wurzel eines Elements der Restklassengruppe modulo N zu berechnen, sofern man die Primfaktorzerlegung von N nicht kennt. Und für die Berechnung der Primfaktorzerlegung ist (bis auf einige Spezialfälle) ebenfalls kein effizienter Algorithmus bekannt.

Natürlich können moderne Computer durch ihre schiere Rechenleistung einfach so lange herumprobieren, bis sie zufällig die Lösung finden. Aber man kann die Größe von N so wählen, dass die Verschlüsselung einer Nachricht auf einem normalen Laptop sehr schnell ist, und gleichzeitig die Faktorisierung selbst mit dem größten Supercomputer länger dauern würde, als das Universum alt ist.

Konkret funktioniert das so: Wichtel Alice erstellt sich ihr persönliches RSA-Schlüsselpaar, das aus einem privaten und einem öffentlichen Schlüssel besteht. Mit dem öffentlichen Schlüssel kann dann jeder Nachrichten an Alice senden, die nur sie selbst mit ihrem privaten Schlüssel entschlüsseln kann.

Dafür wählt Alice zwei zufällige, ungefähr gleich große Primzahlen p und q, und berechnet $N := pq$. Alle folgenden Operationen zur Verschlüsselung und Entschlüsselung finden im Restklassenring $\mathbb{Z}/N\mathbb{Z}$ statt.

Wir erinnern uns: $\mathbb{Z}/k\mathbb{Z}$ ist ein Körper, genau dann, wenn k eine Primzahl ist. Das heißt, alle Elemente haben ein multiplikatives Inverses. Da N keine Primzahl ist, ist $\mathbb{Z}/N\mathbb{Z}$ nur ein Ring. Das heißt, es gibt Elemente, für die kein multiplikatives Inverses existiert. Doch welche sind das?

Aufgabe (2 ginger) Beweise: Ein Element $a \in \mathbb{Z}/N\mathbb{Z}$ besitzt ein multiplikatives Inverses, genau dann, wenn $\mathrm{ggT}(a, N) = 1$, also a und N teilerfremd sind. Lösung: Kajetan

Diese Elemente bilden die Untergruppe der Einheiten (oder auch „prime Restklassengruppe“) $(\mathbb{Z}/N\mathbb{Z})^*$ (oder auch oder $\mathbb{Z}_N^*$). Die sogenannte Eulersche Phi-Funktion $\phi(N)$ beschreibt, wie viele Zahlen $x \leq N$ teilerfremd zu N sind.

Es gilt also:

$$\phi(N) = (\mathbb{Z}/N\mathbb{Z})^*$$

Ihre Berechnung ist nur möglich, wenn man die Primfaktorzerlegung von N kennt. In unserem Fall mit $N = pq$ gilt $\phi(N) = (p-1)(q-1)$.

Nach dem Satz von Euler-Fermat gilt für alle $a \in (\mathbb{Z}/N\mathbb{Z})^*$:

$$a^{\phi(n)} \equiv 1 \pmod{N},$$

Alice hat sich 137 und 173 als Primzahlen ausgesucht und erhält damit

- $N = 23701$
- $\phi(N) = 23392$

Als Nächstes wählt sie den Verschlüsselungs-Exponenten e (von englisch *encrypt*). In der Praxis hat sich $e = 2^{16} + 1$ etabliert, aber diese Zahl ist für Wichtel viel zu groß, deswegen wählt Alice $e = 3$. Das Paar (e, N) bildet Alices öffentlichen Schlüssel, den sie frei verteilen kann. Er ermöglicht es jedem anderen Wichtel, eine verschlüsselte Nachricht zu generieren, die nur noch Alice entschlüsseln kann. Sei $m \leq N$ eine Zahl, die Nachricht repräsentiert (von englisch *(*message)). Dann erhält man den verschlüsselten Text c (von englisch *cypertext*) folgend:

$$c := m^e \pmod{N}$$

Aufgabe (1 ginger): Verschlüssele den Wichtel-Gruß "ho". Die Buchstaben h und o werden von Computern als jeweils 1 Byte (8Bit) verarbeitet. Beide zusammen ergeben das Bitmuster `01001000 01101111`b (siehe ASCII-Tabelle) verarbeitet. Dieses Bitmuster entspricht der Dezimalzahl 18543 =: m. Lösung: $c = 6472$

In der Zwischenzeit hat Alice ihren privaten Schlüssel erstellt, um die ankommenden Nachrichten entschlüsseln zu können. Er besteht aus dem Paar (d, N), wobei d (*decryption*) der Entschlüsselungsexponent ist. d wird immer so gewählt, dass $e \cdot d = 1 + k\phi(N)$ für ein beliebiges $k \in \mathbb{N}$ ist. Mit anderen Worten, d ist multiplikatives Inverses von e modulo $\phi(N)$ (nicht wie sonst modulo N!)

Erhält Alice eine verschlüsselte Nachricht c, berechnet sie:

$$\begin{aligned} c^d \equiv (m^e)^d \equiv m^{ed} \equiv m^{1+k\phi(N)} &\pmod{N} \\ \equiv m \cdot m^{k\phi(N)} \equiv m \cdot 1 &\pmod{N} \end{aligned}$$

Alice potenziert c also exakt so lange weiter, bis sie wieder beim Klartext angekommen ist.

Aufgabe (1 ginger): Der Wichtel Bob sendet eine verschlüsselte Nachricht an Alice. Die Nachricht lautet $c = 759$. Alices privater Schlüssel ist $(d = 15595, N)$.

- Berechne die Zahl m

- Stelle m als Binärzahl da und Unterteile das resultierende Bitmuster in Blöcke zu je 1 Byte (=8 Bit). Nutze eine binäre ASCII-Tabelle, um die Bytes in die ursprünglichen Zeichen zu konvertieren. Alternativ kannst du auch jedes Byte in eine Dezimalzahl konvertieren, und so in jeder beliebigen ASCII-Tabelle nachschauen.

Lösung: 42, aber als ascii characters