1. a) Wie erzeugt sich Alice zu p = 11 und q = 19 und zu e = 7 ihren zugehörigen privaten RSA-Schlüssel? (Warum ist z.B. e = 3 kein zulässiger öffentlicher Schlüssel?)

$n = p \cdot q = 209$ $m = (p - 1)(q - 1) = 180$

e muss teilerfremd zu m sein. 3 ist nicht teilerfremd zu 180 (Faktor 60).

$e \cdot d \bmod m = 1 = 7d \bmod 180$ Nebenrechnung: $7 \cdot 26 = 182 = 2 \pmod m$
$7 \cdot (26 \cdot 4) \bmod 180 = 7 \cdot 104 \bmod 180 = 728 \bmod 180 = 8$ → $7 \cdot 103 \bmod 180 = 1$

Privater Schlüssel: (209, 103) Öffentlicher Schlüssel: (209, 7)

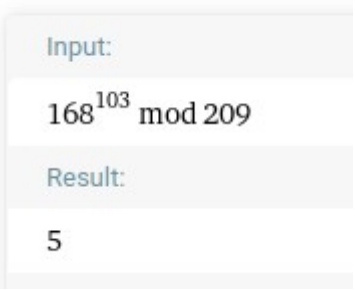
b) Bob möchte an Alice die Nachricht x = 5 verschlüsselt schicken. Wie geht er vor?

$y = x^e \pmod n = 5^7 \pmod{209} = 168$

Geheimtext: 168

c) Wie entschlüsselt Alice diese Nachricht? (Verwenden Sie z.B. Wolfram Alpha.)

$x = y^d \pmod n = 168^{103} \pmod{209} = 5$



Passt!

2. Alice erhält von Bob die verschlüsselte Nachricht y = 43.

a) Wie entschlüsselt sie die Nachricht, wenn ihr geheimer Schlüssel (d, n) = (27, 55) ist? (Verwenden Sie WolframAlpha, da die Zahlen zu groß sind.)

$x = y^d \pmod n = 43^{27} \bmod 55 = 32$

b) Beschleunigung der Entschlüsselung mit dem CRT: Alice kann die Entschlüsselung beschleunigen, indem sie ihre geheimen Primzahlen (p = 5, q = 11), den kleinen Satz von Fermat und den CRT verwendet. Dazu geht sie so vor:

- *Zerlegung von y = 43 in die Reste modulo der Primzahlen:*
  43 = (43 mod 5, 43 mod 11) = (3 mod 5, 10 mod 11).
- *Daraus folgt 43 27 = (3 27 mod 5, 10 27 mod 11) = (3 3 mod 5, 10 7 mod 11) =*
  = (2 mod 5, 10 mod 11), wobei im zweiten Schritt der kleine Satz von Fermat verwendet (3 4 = 1 mod 5 bzw. 10 10 = 1 mod 11) wurde und im dritten Schritt in der ersten/zweiten Komponente mod 5 bzw. mod 11 gerechnet wurde.
- *Nun muss nur noch x mithilfe des CRT aus x = 2 mod 5 x = 10 mod 11 gefunden werden.*

Berechnen Sie x!

$M_1 = 11$ $M_2 = 5$ $N_1 = 1$ $N_2 = 9$ $x = 2 \cdot 11 \cdot 1 + 10 \cdot 5 \cdot 9 = 22 + 450 = 472 = 32 \pmod{55}$

c) Kann man diese Beschleunigung auch für das Verschlüsseln anwenden? Begründen Sie!

Nein, da Bob die geheimen Primzahlen nicht kennt (bzw falls doch haben wir größere Sorgen). Beschleunigung ist beim Verschlüsseln auch nicht notwendig, da e sehr klein gewählt werden kann.

3. Faktorisieren Sie n = 38911 mit der Methode von Fermat.

$n = a^2 - b^2 = (a - b)(a + b)$ $a \geq \sqrt{n}$ $\sqrt{n} \approx 197{,}3$ $a_0 = 198$ Durchprobieren bis $\sqrt{a_k^2 - n} \in \mathbb{N}$

$a_0 = 198$ $\sqrt{198^2 - 38911} = \sqrt{293} \approx 17{,}1$ $a_1 = 199$ $\sqrt{199^2 - 38911} = \sqrt{690} \approx 26{,}3$

$a_2 = 200$ $\sqrt{200^2 - 38911} = \sqrt{1089} = 33$ Gefunden! a=200 b=33 $38911 = 200^2 - 33^2$

$m = 38911 = 167 \cdot 233$

4. Was passiert, wenn Alice bei der RSA-Schlüsselerzeugung irrtümlich nicht zwei Primzahlen verwendet? Betrachten wir ein einfaches Beispiel. Alice wählt p = 12, q = 11; dann ist n = 132 und m = (p − 1)(q − 1) = 110. Wenn sie als öffentlichen Schlüssel e = 3 wählt, ergibt sich für den geheimen Schlüssel d = 3 1 mod 110 = 37.

a) Wie lautet der Geheimtext, wenn Bob den Klartext INFORMATIK, also (in ASCII) 73, 78, 70, 79, 82, 77, 65, 84, 73, 75 mit Alices öffentlichem Schlüssel verschlüsselt?

$y = x^e \pmod n = x^3 \pmod{132}$

73, 78, 70, 79, 82, 77, 65, 84, 73, 75 → 13, 12, 64, 19, 4, 77, 65, 24, 13, 3

b) Wie lautet der „Klartext", wenn Alice diesen Geheimtext wieder entschlüsselt?

$x = y^d \pmod n = y^{37} \pmod{132}$

13, 12, 64, 19, 4, 77, 65, 24, 13, 3 → 73, 12, 4, 79, 16, 77, 65, 84, 73, 75 I ^**J**^**D** O ^**N** MATIK

5. Anzahl der Primzahlen mit Länge l Bit: Zeigen Sie durch Verwendung der Näherung

$\pi(n) \approx \frac{n}{\ln(n)}$ *(Satz 5.29), dass gilt:* $\pi(2^l) - \pi(2^{l-1}) \approx \frac{2^l(l-2)}{l(l-1)\ln(4)}$

$$\pi(2^l) - \pi(2^{l-1}) \approx \frac{2^l}{\ln(2^l)} - \frac{2^{l-1}}{\ln(2^{l-1})} = \frac{2^l}{l \cdot \ln(2)} - \frac{2^{l-1}}{(l-1)\ln(2)}$$
$$= \frac{2^l(l-1)\ln(2)}{l \cdot \ln(2)(l-1)\ln(2)} - \frac{2^{l-1} l \cdot \ln(2)}{l \cdot \ln(2)(l-1)\ln(2)} = \frac{(2^l(l-1)\ln(2)) - (2^{l-1} l \cdot \ln(2))}{l \cdot \ln(2)(l-1)\ln(2)}$$
$$= \frac{\ln(2)((2^l(l-1)) - (2^{l-1} l))}{l(l-1)\ln(2)\ln(2)} = \frac{(2^l(l-1)) - (2^{l-1} l)}{l(l-1)\ln(2)} = \frac{2^{l-1}(2(l-1) - l)}{l(l-1)\ln(2)} = \frac{2^{l-1}(2l-2-l)}{l(l-1)\ln(2)}$$
$$= \frac{2^{l-1}(l-2)}{l(l-1)\ln(2)} = \frac{2^l(l-2)}{l(l-1)\ln(2) \cdot 2} = \frac{2^l(l-2)}{l(l-1)\ln(4)}$$

6. a) Wie viele Primzahlen der Länge l = 128 Bit gibt es näherungsweise?

$$\pi(2^{128}) - \pi(2^{127}) \approx \frac{2^{128} \cdot 126}{128 \cdot 127 \cdot \ln(4)} \approx \frac{4{,}29 \times 10^{40}}{22535{,}6} \approx 1{,}9 \times 10^{36}$$

b) Wie groß ist die Wahrscheinlichkeit, bei zufälliger Wahl einer 128-Bit Zahl eine Primzahl zu treffen?

Anzahl 128-Bit Zahlen dividiert durch Anzahl 128-Bit Primzahlen:
$1{,}7\times10^{38} \div 1{,}9\times10^{36} \approx 0{,}011182307 \approx 1{,}1\%$

7. Einfacher Primzahltest: Der kleine Satz von Fermat lautet:

$a^{p-1} = 1 \pmod p$ *für jede Primzahl p und jede Zahl* $a \in Z^*_p$

Um daher zu testen, ob eine Zahl t eine Primzahl ist, wählt man eine zufällige Zahl $a \in \{2, \dots, t-1\}$. *Falls* $a^{t-1} \not\equiv 1 \pmod t$ *gilt, so ist t sicher keine Primzahl. Falls* $a^{t-1} = 1 \pmod t$ *ist, so kann t eine Primzahl sein oder auch nicht. Die Fehlerwahrscheinlichkeit reduziert sich, wenn man weitere Zahlen* $a \in \{2, \dots, t-1\}$ *wählt und den Test damit nochmals durchführt.*

Testen Sie mithilfe dieses einfachen Primzahltests, ob t eine Primzahl ist.
Wählen Sie jeweils: a = 2, 3, 4, 5, 6, 7. Wie entscheidet der Test?

a) t = 2821

$2^{2820} \bmod 2821 = 1$
$3^{2820} \bmod 2821 = 1$
$4^{2820} \bmod 2821 = 1$
$5^{2820} \bmod 2821 = 1$
$6^{2820} \bmod 2821 = 1$
$7^{2820} \bmod 2821 = 2016 \rightarrow$ eindeutig keine Primzahl

b) t = 809

$2^{808} \bmod 809 = 1$
$3^{808} \bmod 809 = 1$
$4^{808} \bmod 809 = 1$
$5^{808} \bmod 809 = 1$
$6^{808} \bmod 809 = 1$
$7^{808} \bmod 809 = 1 \rightarrow$ gut möglich, dass es eine Primzahl ist