FTK3, WS 2023/24
1. Übungsblatt für den 13.10.2023

1. (a) Bestimme alle Elemente von $\mathbb{Z}_{33}^*$.
   (b) Wie viele Elemente hat $\mathbb{Z}_{33}^*$? Kann man herausfinden, wie viele Elemente $\mathbb{Z}_{33}^*$ hat, ohne alle Elemente zu bestimmen?
   (c) Berechne $\varphi(29)$
   (d) Berechne $\varphi(77)$
   (e) Berechne $\varphi(125)$
   (f) Berechne $\varphi(6469693230)$
   (g) Berechne $\varphi(7922124000)$

2. Von der Zahl $n = 9788741$ ist bekannt, dass sie das Produkt zweier unterschiedlicher Primzahlen ist, und dass $\varphi(n) = 9782412$ ist. Berechne die Primfaktoren von n.

3. Berechne **unter Verwendung von Korollar 1.12**
   (a) $12345^{1234} \bmod 131$
   (b) $987654321^{87654321} \bmod 841$
   (c) $1354^{1231} \bmod 667$

   Verwende zum modularen Potenzieren den Square-and-Multiply-Algorithmus (Algorithmus 1.16), den wir schon in GDK2 kennengelernt haben.

4. Bestimme die Lösungsmenge des modularen Gleichungssystems

$$\begin{aligned} x &= 45 \pmod{79} \\ x &= 49 \pmod{89} \end{aligned}$$

5. Nach dem Eiskellerfest wollen die Helfer*innen das übrig gebliebene Bier gerecht untereinander aufteilen. Jede*r der 23 Helfer*innen bekommt gleich viele Flaschen, dabei bleiben 4 Flaschen über. Zwei Helfer*innen trinken nicht gerne Bier und lehnen dankend ab. Es wird neu aufgeteilt, nun bleiben 6 Flaschen über. Beim Anblick des vielen Biers wird einer Helferin schlecht, sie geht lieber ohne Bier nach Hause. Es wird neu aufgeteilt, nun bleiben 10 Flaschen über. Wie viele Flaschen Bier sind also beim Eiskellerfest übrig geblieben?

6. Entschlüssle die Nachricht 1969 mit dem RSA Private Key $(p, q, d) = (47, 53, 2153)$ unter Verwendung des chinesischen Restsatzes.

7. Implementiere folgende Funktionen in der Programmiersprache deiner Wahl:

- `DecryptSlow` entschlüsselt einen übergebenen Ciphertext mit einem **fixen** RSA-Schlüssel und dem herkömmlichen RSA-Algorithmus OHNE Verwendung von Algorithmus 1.16, d.h. sie berechnet zuerst die Potenz und rechnet anschließend modulo n.
- `DecryptClassic` entschlüsselt einen übergebenen Ciphertext mit einem **fixen** RSA-Schlüssel und dem herkömmlichen RSA-Algorithmus (unter Verwendung von Algorithmus 1.16).

Erzeuge 10 zufällige Chiffrate, teste deine beiden Funktionen mit diesen Chiffraten, miss dabei jeweils die benötigte Zeit und ermittle den Mittelwert deiner Messungen. Der fixe Private Key lautet $(p, q, d) = (1913, 1297, 1723265)$.

8. Implementiere folgende Funktionen in der Programmiersprache deiner Wahl:

   - `DecryptClassic` entschlüsselt einen übergebenen Ciphertext mit einem **fixen** RSA-Schlüssel und dem herkömmlichen RSA-Algorithmus.
   - `DecryptCRS` entschlüsselt einen übergebenen Ciphertext mit einem **fixen** RSA-Schlüssel und Algorithmus 1.19 (RSA-CRS).

   Erzeuge 10 zufällige Chiffrate, teste deine beiden Funktionen mit diesen Chiffraten, miss dabei jeweils die benötigte Zeit und ermittle den Mittelwert deiner Messungen. Der fixe Private Key befindet sich im beiliegenden Textfile. Denk bitte daran, **alle** Komponenten des Private Keys, die du beim Entschlüsseln verwendest, schon außerhalb der Funktionen vorzuberechnen!