5. Übungsblatt
Kryptographie und Mediensicherheit (SoSe 2018)

Bauhaus-Universität Weimar, Professur für Mediensicherheit (Prof. Lucks)
Betreuer: Eik List
URL: `http://www.uni-weimar.de/de/medien/professuren/mediensicherheit/teaching`
Abgabe: 12.06.2018, 11:00 Uhr vor Beginn der Übung oder an `eik.list(at)uni-weimar.de`.
LaTeX-Template: template.tex

Aufgabe 1 – Kleiner RSA-Modulus (4 Punkte)
Alice, Bob und Charlie haben als öffentlichen RSA-Exponenten $e = 3$ gewählt. Ihre öffentlichen RSA-Moduli n_A, n_B und n_C seien paarweise teilerfremd. Diana möchte eine Nachricht m jeweils an Alice, Bob und Charlie schicken und verschlüsselt m wie folgt:

$$\begin{aligned} c_A &\equiv m^3 \bmod n_A \quad \text{mit} \quad c_A = 43 \text{ und } n_A = 319, \\ c_B &\equiv m^3 \bmod n_B \quad \text{mit} \quad c_B = 31 \text{ und } n_B = 323, \\ c_C &\equiv m^3 \bmod n_C \quad \text{mit} \quad c_C = 103 \text{ und } n_C = 299. \end{aligned}$$

Zeigen Sie, dass jeder mit Kenntnis von c_A, c_B, und c_C die Nachricht m entschlüsseln kann. (*Hinweis:* Wiederholen Sie den chinesischen Restsatz).

Aufgabe 2 – RSA (2 Punkte)
Alice und Bob wollen RSA nutzen, um einen geheimen Schlüssel $K \in \{0,1\}^{128}$ zu übertragen. Dazu wählt Alice einen zufälligen Wert K und verschlüsselt ihn mit Bobs öffentlichem RSA-Schlüssel (n, e), wobei $n \in \{0,1\}^{2048}$ und $e = 7$ gilt, zu $Y \leftarrow (0 \ldots 0 \,\|\, K)^e \bmod n$. Anschließend sendet sie Y an Bob. Bob berechnet $K' = Y^d \bmod n$. Zeigen oder widerlegen Sie dass ein passiver Angreifer, der Y mithören kann und (n, e) kennt, K effizient berechnen kann.

Aufgabe 3 – ElGamal (4 Punkte)
Wiederholen Sie die ElGamal-Verschlüsselungsfunktion aus Kapitel 6.4 der Vorlesung. Wir definieren Semantische Chosen-Plaintext-Sicherheit (Sem-CPA) als folgendes Experiment zwischen einem Herausforderer $\mathcal{C}$ und einem Angreifer $\mathbf{A}$:

function EXPERIMENT(k)
  $\mathcal{C}$ wählt eine große Primzahl $p \in \{0,1\}^k$
  $\mathcal{C}$ wählt einen Generator $g \in \mathbb{Z}_p^*$ sowie einen geheimen Schlüssel $a \twoheadleftarrow \mathbb{Z}_p^*$.
  $\mathcal{C}$ veröffentlicht A mit $A \equiv g^a \bmod p$
  $\mathcal{C}$ wählt ein Bit $b \twoheadleftarrow \{0,1\}$ zufällig
  for $i = 1 \,..\, q$ **do**
    $\mathbf{A}$ wählt $(M_0^i, M_1^i) \in \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ sodass weder M_0^i noch M_1^i vorher angefragt wurden.
    $\mathbf{A}$ schickt (M_0^i, M_1^i) an $\mathcal{C}$.
    $\mathcal{C}$ wählt $r_i \twoheadleftarrow \mathbb{Z}_p^*$ zufällig
    $\mathcal{C}$ berechnet $C_1^i \equiv g^{r_i} \bmod p$ und $C_2^i \equiv A^{r_i} \cdot M_b^i \bmod p$.

11: $\mathcal{C}$ antwortet **A** mit (C_1^i, C_2^i).
12: **A** gibt $b' \in \{0, 1\}$ aus.
13: **A** gewinnt genau dann wenn $b = b'$.

Informieren Sie sich zum Legendre-Symbol. Zeigen Sie **entweder** dass ElGamal Sem-CPA-sicher ist **oder** geben Sie einen effizienten Angreifer **A** mit möglichst wenigen Anfragen an und seinen Vorteil an.

Aufgabe 4 – Diffie-Hellman-Protokoll (4 Punkte)
Alice und Bob wollen mittels Diffie-Hellman einen Schlüssel aushandeln. Sie wählen eine große Primzahl p und einen Generator $g \in \mathbb{Z}_p^*$. Alice wählt zufällig ein geheimes $a \in \mathbb{Z}_p^*$ und Bob ein geheimes $b \in \mathbb{Z}_p^*$. Alice schickt Bob ihren öffentlichen Schlüssel (A, g, p) mit $A = g^a \bmod p$ und Bob antwortet mit $B = g^b \bmod p$:

a) Berechnen Sie beispielhaft einmal für $g = 13$ und $p = 883$ sowie einmal für $g = 13$ und $p = 863$ die jeweils durchschnittliche Ordnung $ord(\langle g^{ab} \rangle)$ über alle $a, b \in \{1, \ldots, p-1\}$, z. B. mit einem kleinen Python-Skript (*Hinweis:* Es gilt $ord(\langle g^i \rangle) = \frac{ord(\langle g \rangle)}{ggT(ord(\langle g \rangle), i)}$).

b) Sei $\mathbb{P}$ die Menge der Primzahlen. Wir betrachten zwei Methoden um die Primzahl p zu wählen:

- $p - 1$ besitzt viele kleine Primteiler: $p - 1 = q_1^{k_1} \cdot q_2^{k_2} \cdot q_3^{k_3} \cdot \ldots$, für $q_i \in \mathbb{P}$ und Exponenten $k_i \geq 1$.
- $p - 1 = 2 \cdot q$ für $q \in \mathbb{P}$.

Erläutern Sie: Welche Wahlmethode für p bietet höhere Sicherheit für den auszuhandelnden Schlüssel und was bedeutet höhere Sicherheit in diesem Kontext?

Aufgabe 5 – Pollard's Rho (4 Punkte)
Informieren Sie sich selbständig zu Pollard's Rho-Algorithmus für den Diskreten Logarithmus. Implementieren Sie ihn in Python 3. Ihr Skript soll eine Zahl p, einen Wert $g \in \mathbb{Z}_p^*$ und einen öffentlichen Schlüssel y entgegennehmen. Sofern ein $x \in \mathbb{Z}_p^*$ existiert sodass $y \equiv g^x \bmod p$ gilt, soll Ihr Skript ein solches x mit Pollard's Rho-Algorithmus ermitteln und zurückgeben.
Schicken Sie Ihre Lösung als Anhang einer E-Mail `pollard_rho_<MatrNr>.py` mit dem Betreff `[Krypto SS18] Beleg 5` an `eik.list(at)uni-weimar.de`. Es reicht dabei eine Matrikelnummer aus der Gruppe anzugeben. Die vollständigen Namen und Matrikelnummern sollen als Kommentar im Python-Skript stehen.
Beispielaufrufe:

```
$ python3 pollard_rho_12345.py -p 1019 -g 2 -y 5
10

$ python3 pollard_rho_12345.py -p 1018 -g 2 -y 5
Error: g divides p.
```