Aufgabe 11.1 (3 Pkt.)

Beim sogenannten *universellen Hashing* betrachtet man Hashfunktionen H, die gemäß einer Wahrscheinlichkeitsverteilung $\mathcal{P}$ zufällig aus einer Menge $\mathcal{H}$ gezogen werden. Wir nehmen an, alle Funktionen in $\mathcal{H}$ seien Abbildungen von einer Menge M in eine Menge N.

Für solche zufällig gezogenen Hashfunktionen H definiert man folgende Analoga zum Begriff der Kollisionsresistenz bei gewöhnlichen Hashfunktionen:

- H heißt *ε-universell*, wenn für alle $x, y \in M$, $x \neq y$, gilt:

$$\Pr_{H \sim \mathcal{P}} \big[H(x) = H(y)\big] \leq \varepsilon \,.$$

- H heißt *stark ε-universell*, wenn für alle $x, y \in M$, $x \neq y$, und alle $a, b \in N$ gilt:

$$\Pr_{H \sim \mathcal{P}} \big[H(x) = a \wedge H(y) = b\big] \leq \frac{\varepsilon}{|N|} \,.$$

Es sei K ein endlicher Körper und A eine auf K gleichverteilte Zufallsvariable. Wir betrachten die zufällige Funktion $H \colon K \to K$, $x \mapsto Ax$.

a) Zeigen Sie: H ist $1/|K|$-universell, aber nicht stark $1/|K|$-universell.

b) Wie kann man H modifizieren, um eine stark $1/|K|$-universelle Funktion zu erhalten?

Tipp: Führen Sie eine zusätzliche Zufallsvariable ein.

Aufgabe 11.2 (3 Pkt.)

Bestimmen Sie den diskreten Logarithmus von $y = 82$ bezüglich der Primitivwurzel $a = 5$ modulo $p = 103$ mit Pollards Rho-Algorithmus. Wählen Sie als Startwert $z_0 = a^{s_0} y^{t_0} \bmod p$ mit $s_0 \coloneqq 1$, $t_0 \coloneqq 1$. Verwenden Sie

$$z_{i+1} \coloneqq \begin{cases} z_i^2 \bmod p & \text{falls } 0 < z_i \leq 34 \\ z_i a \bmod p & \text{falls } 34 < z_i \leq 68 \\ z_i y \bmod p & \text{falls } 68 < z_i \leq 102 \end{cases}$$

Protokollieren Sie die auftretenden Werte von z_i, s_i und t_i.

Aufgabe 11.3 (3 Pkt.)

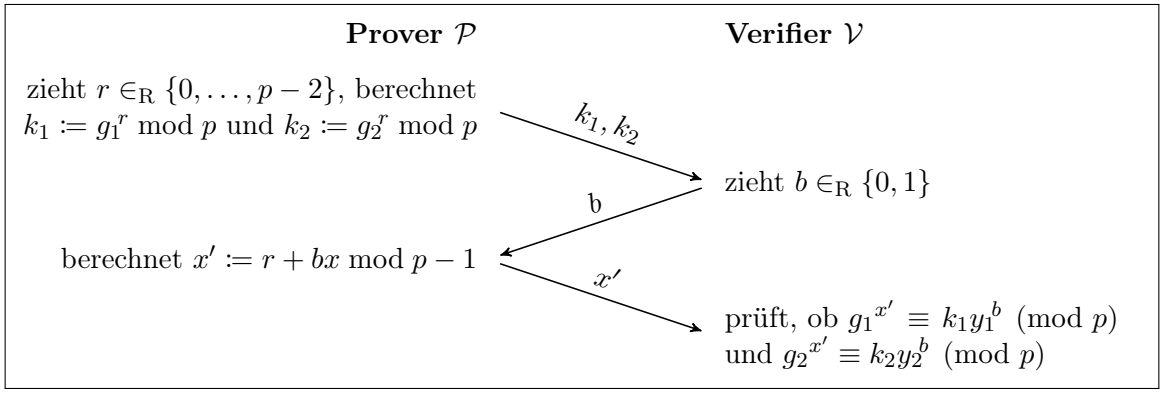
Victor und Peggy versuchen, das 9×9-Sudoku in der aktuellen Ausgabe ihrer Fernsehzeitung zu lösen. Peggy hat bereits eine Lösung gefunden, weiß aber nicht, wie sie das Victor beweisen kann, noch bevor dieser eine Lösung hat und ohne ihm die Lösung zu verraten.

Konstruieren Sie ein Zero-Knowledge-Protokoll, mit dem ein solcher Beweis möglich wäre. Sie dürfen annehmen, dass den beiden kryptographische Verfahren zur Verfügung stehen, z. B. könnte Peggy einzelne Zahlen im Sudoku verschlüsseln, MACs berechnen usw.

Hinweis: Ihr Protokoll braucht nicht die „Perfect-Zero-Knowledge“-Eigenschaft zu erfüllen – in dieser Aufgabe begnügen wir uns damit, wenn die Wahrscheinlichkeitsverteilungen der Transkripte und der Ausgaben des Simulators algorithmisch „sehr schwierig“ zu unterscheiden sind.

Aufgabe 11.4 (4 Pkt.)

Das folgende Protokoll ist ein Beweis für die Gleichheit zweier Logarithmen. Öffentlich sind eine Primzahl p, zwei Primitivwurzeln g_1, g_2 modulo p sowie $y_1 \coloneqq {g_1}^x \bmod p$ und $y_2 \coloneqq {g_2}^x \bmod p$, wobei nur der Prover x kennt. Die Behauptung des Provers lautet: $\log_{g_1} y_1 = \log_{g_2} y_2$.



a) Zeigen Sie: Es handelt sich um einen Zero-Knowledge-Beweis für „$\log_{g_1} y_1 = \log_{g_2} y_2$“, wobei beide Logarithmen geheim bleiben.

b) Gehen Sie nun von einem betrügenden Verifier aus, der für die Challenge b mit fester Wahrscheinlichkeit $P \neq 1/2$ den Wert 1 zieht und den Wert 0 sonst. Die Wahrscheinlichkeitsverteilung auf den Transkripten wird sich also vom regelkonformen Fall unterscheiden. Konstruieren Sie für diesen Fall einen Simulator, der P nicht kennt, aber den Algorithmus von $\mathcal{V}$ Schritt für Schritt ausführen kann. (Mit Begründung!)

Aufgabe 11.5 (3 Pkt.)

Lösen Sie die SPOX-Aufgabe „Solovay-Strassen-Test“.