Ruhr-Universität Bochum
Lehrstuhl für Kryptologie und IT-Sicherheit
Prof. Dr. Alexander May
Elena Kirshanova, Ilya Ozerov

**Hausübungen zur Vorlesung**

# Kryptanalyse

**SS 2014**

Blatt 3 / 06. Mai 2014

Abgabe: 15. Mai 2014, 14.00 Uhr, Kasten NA/02

**AUFGABE 1** (5 Punkte)**:**

In Pollards Rho-Methode habe das Anfangsstück Länge $i$ und der Kreis Länge $k$. Zeigen Sie, dass sich die beiden Kängurus im Punkt $s_m = s_{2m}$ treffen, wobei $m = k \cdot \lceil i/k \rceil$.

*Hinweis:* Es ist nützlich die Identität $x \bmod y = x - y \cdot \lfloor x/y \rfloor$ zu benutzen.

**AUFGABE 2** (5 Punkte)**:**

Schreiben Sie eine Funktion in sage, die den Pollard Rho Algorithmus aus der Vorlesung durchführt. Die Funktion soll als Eingabe die Primzahl $p$, ein Element $\alpha$, die Ordnung $n = \mathrm{ord}(\alpha)$, sowie ein Element $\beta$ erhalten. Die Ausgabe soll $z = \mathrm{dlog}_\alpha(\beta) \bmod n$ sein.

Partitionieren Sie $\mathbb{Z}_p^*$ in $S_1 = \{s \in \mathbb{Z}_p^* \mid s = 0 \bmod 3\}$, $S_2 = \{s \in \mathbb{Z}_p^* \mid s = 1 \bmod 3\}$ und $S_3 = \{s \in \mathbb{Z}_p^* \mid s = 2 \bmod 3\}$ und berechnen Sie mit Ihrer Funktion den diskreten Logarithmus von $\beta = 1208341$ zur Basis $\alpha = 2$ in $\mathbb{Z}_p^*$ mit $p = 1234547$, wobei $\alpha$ ein Generator von $\mathbb{Z}_p^*$ ist. Wie viele Schritte sind nötig? Stimmt das mit der erwarteten Anzahl an Schritten überein? Geben Sie den Quelltext mit ab.

**AUFGABE 3** (5 Punkte)**:**

Konstruieren Sie einen Algorithmus für das $k$-Listen Problem mit $k = 2^m + j$, $0 < j < 2^m$ mit Komplexität $\widetilde{\mathcal{O}}(k\, 2^{\frac{n}{m+1}})$ für den Fall, dass alle Eingabelisten jeweils $2^{\frac{n}{m+1}}$ Elemente haben. Zeigen Sie Laufzeit und Korrektheit.

*Hinweis:* Verallgemeinern Sie die Idee aus Präsenzaufgabe 3.3 für $2^m$ (statt 4) Listen.

Verwenden Sie Ihren Algorithmus um das folgende 5-Listen Problem zu lösen. Finden Sie dazu $x_1 \in L_1, \ldots, x_5 \in L_5$ mit $x_1 \oplus \ldots \oplus x_5 = 0$.

$$L_1 = \{010011, 101010, 001101, 111100\},\ L_2 = \{001011, 011101, 001000, 110111\},$$
$$L_3 = \{001110, 000001, 011000, 010010\},\ L_4 = \{100001, 010001, 001100, 000101\},$$
$$L_5 = \{100000, 001000, 000100, 000001\}$$

Bitte wenden!

**AUFGABE 4** (5 Punkte)**:**

Wir betrachten in dieser Aufgabe das in der Vorlesung beschriebene Ringsignaturverfahren (Folien 44,45). Nehmen Sie an, dass $\ell$ User jeweils einen Modul $N_i$ der Länge $\ell$ Bit haben ($e = 65537$ wird fest gewählt). In der Vorlesung haben Sie gesehen, dass man für diesen Fall mit Hilfe des $k$-Listen Algorithmus einen subexponentiellen Angriff erhält.

(a) Zeigen Sie, dass in diesem Fall sogar ein polynomieller Angriff existiert. Verwenden Sie dazu den Algorithmus von Bellare und Micciancio auf Folie 49.

(b) Implementieren Sie Ihren Angriff in sage für $\ell = 256$. In der Datei ringvrfy.sage sind die 256 Moduli $N_i$, sowie eine Implementierung der Vrfy-Funktion gegeben. Die Implementierung verwendet die Hashfunktion sha256. Ihr Ziel ist das Erzeugen einer Fälschung für die Nachricht `Kryptanalyse`, d.h. Sie müssen eine Liste $\mathsf{R} = [m_1, \ldots, m_{256}]$ liefern, so dass die Vrfy-Funktion true zurückgibt.

*Zur Abgabe:* Laden Sie die Datei ringvrfy.sage (mit eingetragener Liste R) in Moodle hoch. Den Quellcode Ihrer Implementierung geben Sie bitte wieder ausgedruckt mit ab.

*Hinweise:* Definieren Sie mit $\mathsf{M} = \mathsf{MatrixSpace}(\mathsf{GF}(2), 256, 256)$ den Raum der $256 \times 256$ Matrizen mit Elementen aus $\mathbb{F}_2$. Sie können dann z.B. die folgende Liste von Listen von $\{0,1\}$-Werten $\mathsf{L} = [[0,1,0,\ldots,1],\ldots,[1,1,0,\ldots,0]]$ mit $\mathsf{M}(\mathsf{L})$ in eine Matrix umdefinieren. Analog können Sie mit $\mathsf{V} = \mathsf{VectorSpace}(\mathsf{GF}(2), 256)$ Vektoren definieren. Schließlich kann mit $\mathsf{b} = \mathsf{A.solve\_left}(\mathsf{v})$ das Gleichungssystem $b \cdot A = v$ gelöst werden, wobei $A$ eine Matrix und $b$ und $v$ Vektoren sind.