

Übungen zur Kryptographie und Datensicherheit

Andre Löffler

December 16, 2013

1. Übung

1.1 Aufgabe 1

$A \subseteq \mathbb{N}$, μ probabilistische Maschine mit $P(\mu(x) = c_A(x)) = \alpha \geq \frac{3}{4}$. O.B.d.A gibt μ nur Werte aus $0, 1$ zurück. μ' arbeitet wie folgt:

1. simuliere $\mu(x)$ und weise diesen Wert y_1 zu.
2. simuliere $\mu(x)$ und weise diesen Wert y_2 zu.
3. simuliere $\mu(x)$ und weise diesen Wert y_3 zu.
4. simuliere $\mu(x)$ und weise diesen Wert y_4 zu.
5. simuliere $\mu(x)$ und weise diesen Wert y_5 zu.
6. simuliere $\mu(x)$ und weise diesen Wert y_6 zu.
7. simuliere $\mu(x)$ und weise diesen Wert y_7 zu.
8. Falls Mehrzahl der y_i gleich 1 ist, gib 1 zurück.

$$P(\mu'(x) \neq c_A(x)) = P(\text{mind. 4 der } y_i \text{ haben nicht den Wert } c_A(x))$$

$$\begin{aligned} &= \sum_{k=4}^7 P(\text{genau } k \text{ der } y_i \neq c_A(x)) \\ &= \binom{7}{4}(1-\alpha)^4\alpha^3 + \binom{7}{5}(1-\alpha)^5\alpha^2 + \binom{7}{6}(1-\alpha)^6\alpha + \binom{7}{7}(1-\alpha)^7 \end{aligned}$$

Nebenüberlegung:

$$\alpha(1-\alpha) = -(\alpha - \frac{1}{2})^2 + \frac{1}{4}. \quad \alpha \text{ ist im Intervall } [\frac{1}{2}, 1] \text{ monoton fallend:}$$

$$\alpha(1-\alpha) \leq \frac{3}{4} \cdot \frac{1}{4} = \frac{3}{16}.$$

Schätze damit $1-\alpha \leq \frac{1}{4}$ ab. Damit ist obige Summe $\leq 0,08$.

$$P(\mu'(x) = c_A(x)) = 1 - P(\mu'(x) \neq c_A(x)) \geq 1 - 0,08 \geq \frac{11}{12}$$

1.2 Aufgabe 2

1. Alphabet $\{1, 2\}$ ist endliche, nichtleere Menge. ✓
2.
 - K ist deterministisch, also auch probabilistischer Algorithmus.
 - Legendres Vermutung: zwischen n^2 und $(n+1)^2$ liegt stets eine Primzahl.
 - Angenommen, die Vermutung gilt und wir suchen ab $m = \underbrace{1 \dots 1}_{n+1}$ nach einer Primzahl, könnte es sein, dass wir erst bei $(\sqrt{m} + 1)^2 = m + 2\sqrt{m} + 1$ fündig werden.
 - Testen also, $O(\sqrt{n}) = O(n^{\frac{1}{2}}) = O(2^{\frac{1}{2}n})$ Zahlen
 \Rightarrow nicht klar, ob Polynomialzeit möglich ist.
3. $\varepsilon(e, m)$ liefert $e \cdot \text{dya}^{-1}(m)$ für $m \in \{1, 2\}^*$
 \Rightarrow Polynomialzeit-Algorithmus ✓

4. $D(d, c)$ liefert $\text{dya}(\frac{c}{q})$, wobei q der größte Primfaktor von c ist. \Rightarrow unklar ob im Polynomialzeit möglich, da Faktorisierung nötig.
5. Sei (e, d) ein von $K(1^n)$ genutztes Schlüsselpaar und $m \in \{1, 2\}$
 $\Rightarrow (e, d) = (q, 1)$, wobei q die kleine Primzahl mit $|\text{dya}(q)| > n$
 $\Rightarrow \varepsilon(e, m) = q \cdot \text{dya}^{-1}(m)$

$$D(d, \varepsilon(e, m)) = D(1, q \cdot \text{dya}^{-1}(m)) = \text{dya} \left(\frac{\overbrace{q \cdot \text{dya}^{-1}(m)}^c}{q'} \right), \text{ wobei } q' \text{ der}$$

größte Primfaktor von c ist.

$q = q'$, weil $|m| = n < |\text{dya}(q)|$, q größter Primfaktor von $q \cdot \text{dya}^{-1}(m)$

$\Rightarrow D(d, \varepsilon(e, m)) = \text{dya}(\text{dya}^{-1}(m)) = m \checkmark$

1.3 Hinweise zu Übungsblatt 2

1. Sei p eine Primzahl.
 $\mathbb{F}_p =_{\text{def}} (\mathbb{Z}_p, +_p, \cdot_p)$ mit $+_p, \cdot_p$: Addition und Multiplikation modulo p .
 \mathbb{F}_p ist ein endlicher Körper, der (bis auf Isomorphie) einzige endliche Körper mit genau p Elementen.
 Beispiel: \mathbb{F}_2 : 1 ist das Einselement, 0 ist das Nullelement. $5 \cdot 3 = 1$, also ist 3 das inverse Element zu 5.
2. Sei $q = p^n$ mit einer Primzahl p und $n \geq 2$. Ziel: der Körper \mathbb{F}_q mit q Elementen.

$$\begin{aligned} \mathbb{F}_p[x] &= \text{Menge aller Polynome mit Koeffizienten aus } \mathbb{F}_p \\ &= \{a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x^1 + a_0 \mid n \geq 0, a_0, \dots, a_n \in \mathbb{F}_p\} \\ &= \{(a_n, \dots, a_0) \mid n \geq 0, a_n, \dots, a_0 \in \mathbb{F}_p\} \end{aligned}$$

Die Multiplikation von Elementen aus $\mathbb{F}_p[x]$ entspricht der Polynommultiplikation.

$$\text{Beispiel: } \mathbb{F}_2[x]: (x^2 + 1)(x^2 + 1) = x^4 + 2x^2 + 1 = x^4 + 1$$

Definition 1. Ein Polynom $g \in \mathbb{F}_p[x]$ heißt irreduzibel über \mathbb{F}_p
 $\Leftrightarrow_{\text{def}}$ es gibt keine Polynome $p_1, p_2 \in \mathbb{F}_p[x]$ mit $\text{Grad} \geq 1$ mit $g = p_1 \cdot p_2$

Satz 1.1. $x^8 + x^4 + x^3 + x + 1$ ist irreduzibel über \mathbb{F}_p .

Definition 2. Sei $g \in \mathbb{F}_p[x]$ irreduzibel und vom Grad $k \geq 1$.

$$\begin{aligned} \mathbb{F}_p[x]/g &=_{\text{def}} \{f \in \mathbb{F}_p[x] \mid \text{Grad von } f < k\} \\ &= \text{Reste bei Polynomdivision durch } g \\ &= \{(a_{k-1}, \dots, a_0) \mid a_0, \dots, a_{k-1} \in \mathbb{F}_p\} \end{aligned}$$

Satz 1.2. [Addition in F]

Addition der Polynome, wobei die Koeffizienten entsprechend \mathbb{F}_p addiert werden.

Satz 1.3. [Multiplikation in F]

$\underbrace{p_1 \cdot p_2}_{\text{Multiplikation in } \mathbb{F}_p[x]/g} = \text{Rest von } \underbrace{p_1 \cdot p_2}_{\text{Multiplikation in } \mathbb{F}_p[x]} \text{ bei Division durch } g.$

Beispiel: $p = 2$ und $g(x) = x^8 + x^4 + x^3 + x + 1 \in \mathbb{F}_2[x]$.

Sei $p_1 = x^7 + x^2 + 1$ und $p_2 = x^2 + 1 \in \mathbb{F}_2[x]/g$.

$$\begin{aligned} p_1 \cdot p_2 &= ((x^7 + x^2 + 1) \cdot (x^2 + 1)) \mod g \\ &= (x^9 + x^4 + x^2 + x^7 + x^2 + 1) \mod g \\ &= (x^9 + x^7 + x^4 + 1) \mod g \\ &= ((x^9 + x^7 + x^4 + 1) - x \cdot g) \mod g \\ &= ((x^9 + x^7 + x^4 + 1) - (x^9 + x^5 + x^4 + x^2 + x)) \mod g \\ &= (x^7 + x^5 + x^2 + x + 1) \mod g \\ &= (x^7 + x^5 + x^2 + x + 1) \end{aligned}$$

Satz 1.4. Sei p eine Primzahl, $k \geq 2$ und $g \in \mathbb{F}_p[x]$ irreduzibel und vom Grad k . Dann ist

$$\mathbb{F}_{p^k} = \text{def}(\mathbb{F}_p[x]/g, +, \cdot)$$

der einzige endliche Körper mit p^k Elementen (bis auf Isomorphie).

Beispiel: Sei $g = x^8 + x^4 + x^3 + x + 1$. Die Elemente von $\mathbb{F}_{2^8} = \mathbb{F}_2[x]/g = \{(a_7, \dots, a_0) | a_0, \dots, a_7 \in \{0, 1\}\}$ lassen sich als Bytes interpretieren.

$$\begin{aligned} 0x03 \cdot 0xa1 &= 0b00000011 \cdot 0b10100001 \\ &= (x + 1) \cdot (x^7 + x^5 + 1) \mod g \\ &= (x^8 + x^6 + x + x^7 + x^5 + 1) \mod g \\ &= (x^7 + x^6 + x^5 + x^4 + x^3) \\ &= 0b11111000 = 0xf8 \end{aligned}$$

$$\Rightarrow 3 \cdot 161 = 248.$$

2 Übung 2

2.1 Aufgabe 1

1. $(\{0, 1, 2, 3, 4\}, f)$ mit $f(x, y) = (x + y) \bmod 5$ ist eine endliche kommutative Gruppe

- $f : \mathbb{Z}_5 \times \mathbb{Z}_5 \rightarrow \mathbb{Z}_5$ ist total ✓
- Assoziativität:

$$\begin{aligned} f(f(a, b), c) &= f((a + b) \bmod 5, c) \\ &= ((a + b) \bmod 5 + c) \bmod 5 \\ &= (a + b + c) \bmod 5 = (a + (b + c) \bmod 5) \bmod 5 \\ &= f(a, f(b, c)) \checkmark \end{aligned}$$

- Neutrales Element: $f(x, 0) = (x + 0) \bmod 5 = x$
 $f(0, x) = (0 + x) \bmod 5 = x$
- Inverses Element: Sei $a \in \mathbb{Z}_5$:

$$\begin{aligned} b &= (-a) \bmod 5 \\ &= (a + (-a) \bmod 5) \bmod 5 \\ &= 0 \bmod 5 = 0 \end{aligned}$$

Eindeutigkeit: $0 \bmod 5 = 0$ ✓

Angenommen $f(a, b') = 0 = f(a, b)$ mit $b' \in \mathbb{Z}_5$

$$\Rightarrow b' \bmod 5 = b \bmod 5$$

$$\Rightarrow b' = b$$

\Rightarrow genau ein inverses Element.

- Kommutativität:
 $f(a, b) = f(b, a) = (a + b) \bmod 5 = (b + a) \bmod 5 = f(b, a) \checkmark$

2. (\mathbb{Z}_6, f, g) mit $f(x, y) = (x + y) \bmod 6$, $g(x, y) = (x \cdot y) \bmod 6$ ist kein Körper:

Damit (\mathbb{Z}_6, f, g) ein Körper ist, muss $(\{1, 2, 3, 4, 5\}, g)$ kommutative Gruppe sein.

$$\Rightarrow \forall a \in \{1, 2, 3, 4, 5\} \exists! b \in \{1, 2, 3, 4, 5\} : [g(a, b) = 1]$$

- $g(2, 1) = 2$
- $g(2, 2) = 4$
- $g(2, 3) = 0$
- $g(2, 4) = 2$
- $g(2, 5) = 4$

$$\Rightarrow 2 \text{ besitzt kein inverses Element in } (\{1, 2, 3, 4, 5\}, g)$$

$\Rightarrow (\mathbb{Z}_6, f, g)$ ist kein Körper.

2.2 Hinweise zu Übungsblatt 3

Monoalphabetische Verschlüsselung (Substitutionschiffre):
 ABCDEFGHJKLMNOPQRSTUVWXYZ
 pfeile und so

$$S = (\Sigma, \mathcal{K}, \mathcal{E}, \mathcal{D})$$

- $\Sigma = \{A, \dots, Z\}$
- Schlüssel: $\pi : \Sigma \rightarrow \Sigma$ Bijektion

$$\begin{aligned} P &= \text{Menge aller Permutationen auf } \Sigma \\ &= \{\pi : \Sigma \rightarrow \Sigma \mid \pi \text{ bijektiv}\} \end{aligned}$$

- $\mathcal{K}(1^n)$ liefert gleichverteilt Element aus $\{(\pi, \pi) \mid \pi \in P\}$
- $\mathcal{E}(\pi, m_1, m_2, \dots, m_n) = \pi(m_1)\pi(m_2)\dots\pi(m_n)$
- $\mathcal{D} = \pi^{-1}(c_1)\pi^{-1}(c_2)\dots\pi^{-1}(c_n)$

Ist S perfekt sicher?

- Betrachte Klartext der Länge 1
 $P_{\Sigma^1}(a) = \frac{1}{26}$ für alle $a \in \Sigma$
 Sei nun $m \in \Sigma^1$ beliebig.
 $P(E_m) = \frac{1}{26}$ [Wahrscheinlichkeit, das Klartext m gewählt wird.]
 Sei $c \in C_1$ beliebig. [$C_1 = \Sigma$ alle möglichen Chiffretexte der Länge 1]

$$\begin{aligned} P(E_m|E_c) &= P(\mathcal{K}(1) \text{ liefert Permutation } \pi \text{ mit } \pi(m) = c) \\ &= \frac{25!}{26!} = \frac{1}{26} = P(E_m) \end{aligned}$$

$\Rightarrow S$ ist perfekt sicher bezüglich P_{Σ^1}

- Betrachte gleichverteilte Klartexte der Länge 2
 $P_{\Sigma^2}(m) = \frac{1}{26^2}$ für alle $m \in \Sigma^2$ [$|\Sigma^2| = 26^2$]
 Wähle $m = AF \in \Sigma^2$:
 $P(E_m) = \frac{1}{26^2}$
 $C_2 = \Sigma^2$
 Sei $c = ww \in C_2$

$$\begin{aligned} P(E_m|E_c) &= P(\mathcal{K}(11) \text{ liefert Permutation } \pi \text{ mit } \pi(A) = w \text{ und } \pi(F) = w) \\ &= 0 \neq P(E_m) \end{aligned}$$

$\Rightarrow S$ nicht perfekt sicher bezüglich P_{Σ^2}

$\Rightarrow S$ nicht perfekt sicher

3. Übung

3.1 Aufgabe 2

a) $\mathcal{M} = \{1, 2, 3, 4, 5, 6\}$

$$\mathcal{P} = \mathcal{M} \rightarrow [0, 1]$$

$$\mathcal{P}(x) = \frac{1}{6^5} \text{ für jedes } x \in \mathcal{M} \text{ wegen } |\mathcal{M}| = 6^5$$

b) $G = \{(a_1, \dots, a_5) \in \mathcal{M} \mid \{a_1, \dots, a_5\} \in \{1, 2, 3, 4, 5\}, \{2, 3, 4, 5, 6\}\}$

$$K = \{(a_1, \dots, a_5) \in \mathcal{M} \mid \begin{aligned} &\{1, 2, 3, 4\} \subseteq \{a_1, \dots, a_5\} \vee \\ &\{2, 3, 4, 5\} \subseteq \{a_1, \dots, a_5\} \vee \\ &\{3, 4, 5, 6\} \subseteq \{a_1, \dots, a_5\} \end{aligned}$$

c) $P(G) = P((a_1, \dots, a_5) \mid \{a_1, \dots, a_5\} = \{1, 2, 3, 4, 5\})$
 $+ P(G) = P((a_1, \dots, a_5) \mid \{a_1, \dots, a_5\} = \{2, 3, 4, 5, 6\})$
 $= 2 \cdot \frac{5!}{6^5} = \frac{5}{16^2} \approx 3,1\%$
 $P(K) :$

1. Fall: Alle Würfel sind verschieden, dann gibt es die Möglichkeiten (bis auf Permutation): $\underbrace{1234}5, \underbrace{1234}6, \underbrace{1345}6, \underbrace{2345}6$
 $\Rightarrow 4!$ viele Möglichkeiten

2. Fall: Genau eine Zahl kommt doppelt vor: Es gibt die Möglichkeiten: 1234, 2345, 3456

$$\begin{aligned} &\Rightarrow 3 \cdot \underbrace{4}_{\text{doppelte Ziffern}} \cdot \overbrace{\binom{5}{2}}^{\text{doppelte Ziffern}} \cdot \underbrace{3!}_{3 \text{ verschiedene Möglichkeiten}} \\ &\Rightarrow \text{Gesamt: } |K| = 4 \cdot 5! + 3 \cdot 4 \cdot \frac{5!}{2! \cdot 3!} \cdot 3! = 4 \cdot 5! + 6! \\ &\Rightarrow P(K) = \sum_{x \in K} P(x) = \sum_{x \in K} \frac{1}{6^5} = \frac{|K|}{6^5} = \frac{4 \cdot 5! + 6!}{6^5} = \frac{25}{16^2} \approx 15\% \\ &P(G|K) = \frac{P(G \cap K)}{P(K)} = \frac{P(G)}{P(K)} = \frac{1}{5} \end{aligned}$$

3.2 Aufgabe 3

$$K_n := \{e \mid (e, a) \text{ ist Ausgabe von } K(1^n)\}$$

$$C_n := \{c \mid c \text{ ist Ausgabe von } \mathcal{E}(e, m) \text{ für } e \in K_n, m \in \Sigma^n\}$$

$$E_m := \{m\} \times \Sigma^n \times C_n \text{ für } m \in \Sigma^n$$

$$E_c := \Sigma^n \times K_n \times \{e\} \text{ } c \in C_n$$

Wir wählen $n = 2$, P_{Σ^n} gleichverteilt, $m = 00$, $c = 01$.

$$E_m \cap E_c = \emptyset \Rightarrow P(E_m \cap E_c) = 0 \neq P(E_m) = \frac{1}{4}$$

$\Rightarrow S$ nicht perfekt sicher.

oder:

Proposition 3.7:

$$K_n \subseteq \Sigma^n \text{ für } n \geq 2 : K_n \subsetneq \Sigma^n$$

$$\Rightarrow |K_n| < |\Sigma^n|$$

$\Rightarrow S$ ist nicht perfekt sicher.

4 4. Übung

4.1 Aufgabe 1

a) $S = (\Sigma, \mathcal{K}, \mathcal{E}, \mathcal{D})$ mit

- (i) $\exists n \geq 1 \exists e \in K_n$ mit $P(E_e) \neq \frac{1}{|K_n|}$
- (ii) $\forall n \geq 1 \forall m \in \Sigma^n \forall c \in C_n \exists! e \in K_n$ mit $E(e, m) = c$.
 $\Sigma = \{0, 1\}$
 $\mathcal{K}(1^n)$ liefert jedes Element aus $\{(e, e) | e \in 0\Sigma^n\}$ mit Wahrscheinlichkeit $\frac{3}{4} \cdot \frac{1}{2^n}$
 $\mathcal{K}(1^n)$ liefert jedes Element aus $\{(e, e) | e \in 1\Sigma^n\}$ mit Wahrscheinlichkeit $\frac{1}{4} \cdot \frac{1}{2^n}$
 $E(e_0 \cdots e_n, m_1 \cdots m_n) := e_0 c_1 \cdots c_n$ mit $c_i = (m_i + e_i) \mod 2$
 $D(e_0 \cdots e_n, c_0 \cdots c_n) := m_1 \cdots m_n$ mit $m_i = (c_i - e_i) \mod 2$
zu (i): $n = 1, |K_n| = 4, P(e_{00}) = \frac{3}{8} \neq \frac{1}{|K_n|}$
zu (ii): Sei $m = m_1 \cdots m_n$ und $c = c_0 \cdots c_n \in C_n$
Der einzige Schlüssel $e \in K_1$ mit $E(e, m)$ ist: $e = e_0 \cdots e_m$ mit $e_0 = c_0$
und $e_i = (c_i + m_i) \mod 2$ für $i \geq 1$
Zeigen, dass S perfekt sicher ist:
Sei $n \geq 1$ und P_{Σ^n} eine Verteilung auf Σ^n
Sei $m \in \Sigma^n$ und $c = c_0 \cdots c_n \in C_n$
O.B.d.A. $c_0 = 0$, zeigen $P(E_m | E_c) = P(E_m)$
1. Fall: $P(E_m) = 0 : P(E_m | E_c) = P(E_m) = 0$
2. Fall: $P(E_m) > 0$:

- für jedes $q \in \Sigma^n$ gibt es genau ein $e \in \Sigma^{n+1}$, sodass $E(e, q) = c$
Bezeichnen dieses e mit $e_{m,c}$.
Es gilt $e_{m,c} \in 0\Sigma^n$
- Aus $P(E_m) > 0, P_{K_n} = \frac{3}{4} \frac{1}{2^n}$ und $E(e_{m,c}, m) = c$ folgt $P(E_c) > 0$
- $P(E_m | E_c) = \frac{P(E_m)P(E_c | E_m)}{P(E_c)} = \frac{P(E_m)P(e_{m,c} | E_m)}{P(E_c)} = \frac{P(E_m)P(E_c)}{P(E_c)} =$
 $\frac{P(E_m) \frac{3}{4} \frac{1}{2^n}}{P(E_c)} = \frac{P(E_m) \frac{3}{4} \frac{1}{2^n}}{\sum_{q \in \Sigma^n} P(E_q) P(E_{e_{q,c}})} = \frac{P(E_m) \frac{3}{4} \frac{1}{2^n}}{\frac{3}{4} \frac{1}{2^n} \sum_{q \in \Sigma^n} P(E_q)} = P(E_m)$

$\Rightarrow S$ ist perfekt sicher.

b) $S = (\Sigma, \mathcal{K}, \mathcal{E}, \mathcal{D})$ mit

- (i) $\exists n \geq 1 \exists e \in K_n$ mit $P(E_e) \neq \frac{1}{|K_n|}$
- (ii) $\forall n \geq 1 \forall m \in \Sigma^n \forall c \in C_n \exists e_1, e_2 \in K_n$ mit $e_1 \neq e_2$ und $E(e_1, m) = E(e_2, m) = c$.
 $\Sigma = \{0, 1\}, \mathcal{K}(1^n)$ liefert jedes Element aus Σ^{n+1} gleichverteilt.
 $\mathcal{E}(e_0 \cdots e_n, m_1 \cdots m_n) = c_1 \cdots c_n$ mit $c_i = (e_i + m_i) \mod 2$
 $\mathcal{D}(e_0 \cdots e_n, c_1 \cdots c_n) = m_1 \cdots m_n$ mit $m_i = (e_i + c_i) \mod 2$
Zu (ii): $n = 1, m = 0, c = 0 : e_1 = 10$ und $e_2 = 00$ mit $\mathcal{E}(10, 0) = 0 = \mathcal{E}(00, 0)$.
Zeige, dass S perfekt sicher ist: Sei $n \geq 1, P_{\Sigma^n}$ eine Verteilung über $\Sigma^n, m \in \Sigma^n, c = c_1 \cdots c_n \in C_n$
1. Fall: $E(E_m | E_c) = 0 = E(E_m)$
2. Fall: $E(E_m) > 0$

Zeigen, dass $P(E_m|E_c) = P(E_m)$:

Definiere $e_{0,m,c} = 0e_1 \cdots e_n$ und $e_{1,m,c} = 1e_1 \cdots e_n$

$\Rightarrow e_i = (m_i + c_i) \mod 2$

$$P(E_m|E_c) = \frac{P(E_m)P(E_c|E_m)}{P(E_c)} = \frac{P(E_m)P(E_{e_{0,m,c} \cup E_{e_{1,m,c}}} | P(E_m))}{\sum_{q \in \Sigma^n} P(E_q)P(E_{e_{0,m,c} \cup E_{e_{1,m,c}}} | P(E_q))} = \frac{P(E_m) \frac{2}{|\Sigma^n|}}{\frac{2}{|\Sigma^n|} - \sum_{q \in \Sigma^n} P(E_q)} = \frac{P(E_m)}{P(E_m)}$$

$\Rightarrow S$ ist perfekt sicher.

c) $S = (\Sigma, \mathcal{K}, \mathcal{E}, \mathcal{D})$. Annahme: $|\Sigma| \geq 2$:

zu $n \geq 1$: Wegen $|\Sigma| \geq 2$ finde $m, m' \in \Sigma^n, m \neq m'$.

Ferner gilt: Ausgaben von $\mathcal{E}(e_i, m)$ und $\mathcal{E}(e_i, m')$ stets verschieden, denn

$m = \mathcal{D}(d, \mathcal{E}(e, m)) = \mathcal{D}(d, \mathcal{E}(e, m')) = m'$

Sei P_{Σ^n} gleichverteilt auf Σ^n . Sei c Ausgabe von $\mathcal{E}(e, m')$:

$0 = P(E_m|E_c \cap E_e) < P(E_m|E_e) = P(E_m)$

4.2 Hinweise zu Blatt 5

Der erweiterte Euklidische Algorithmus für 99 und 78:

Ziel: Berechne $ggT(a, b)$

Erweiterter Euklidischer Algorithmus: Berechne $s, t \in \mathbb{Z}$ mit $ggT(a, b) = s \cdot a + t \cdot b$

$$99 = 1 \cdot 78 + 21$$

$$78 = 3 \cdot 21 + 15$$

$$21 = 1 \cdot 15 + 6$$

$$15 = 2 \cdot 6 + 3$$

$$6 = 2 \cdot 3 + 0$$

$$\Rightarrow ggT(99, 78) = 3$$

$$3 = 15 - 2 \cdot 6 = 15 - 2 \cdot (21 - 2 \cdot 15) = 3 \cdot 15 - 2 \cdot 21 = 3 \cdot (78 - 3 \cdot 21) - 2 \cdot 21 =$$

$$3 \cdot 78 - 11 \cdot 21 = 3 \cdot 78 - 11 \cdot (99 - 1 \cdot 78) = 14 \cdot 78 - 11 \cdot 99 \Rightarrow s = -11 \text{ und } t = 14$$

in RSA: $\varphi(n) = (p-1)(q-1)$

Berechne $d = e^{-1} \mod \varphi(n)$. Wir wissen: $ggT(e, \varphi(n)) = 1$.

Berechne mit euklidischem Algorithmus: $ggT(e, \varphi(n)) = s \cdot \varphi(n) + t \cdot e$

$t \cdot e = 1 - s \cdot \varphi(n) \Rightarrow t \cdot e = 1 \mod \varphi(n) \Rightarrow t$ ist Inverses von $e \mod \varphi(n)$

5 5. Übung

Satz 5.1. Für $a, b \in \mathbb{N}^+$ liefert der Algorithmus $\text{eged}(a, b)$ eine Ausgabe (d, x, y) , sodass $d = \text{ggT}(a, b)$ und $x, y \in \mathbb{Z}$ und $d = x \cdot a + y \cdot b$. Außerdem besitzt der Algorithmus polynomielle Laufzeit.

Proof. 1. Algorithmus terminiert:

$$b = d_1 > d_2 > \dots > d_n = 0$$

\Rightarrow while-Schleife wird höchstens b -mal durchlaufen.

2. Zeigen $d|a$ und $d|b$:

$$0 = d_n = d_{n-2} \% d_{n-1} \quad (*)$$

Es gilt: $d|d_i, d|d_{i+1}, \dots, d|d_{n-1} \Rightarrow d|d_{i-1}$

Beweis: $d_{i+1} = d_{i-1} \% d$, also $d_{i-1} = k \cdot d_i + d_{i+1}$ für ein $k \in \mathbb{N} \Rightarrow d|d_{i-1}$

Aus $(*)$ folgt $d|d_0 = a$ und $d|d_1 = b$.

3. Jeder Teiler d' von a und b ist auch Teiler von d_i .

$$d'|d_0 = a \text{ und } d'|d_1 = b$$

Es gilt $d'|d_i$ und $d'|d_{i+1} \Rightarrow d'|d_{i+2} \quad (**)$

Beweis: $d_{i+2} = d_i \% d_{i+1}$, also

$$d_i = k \cdot d + d_{i+2} \text{ für ein } k \in \mathbb{N}$$

$$\Rightarrow d_{i+2} = d_i - k \cdot d_{i+1} \Rightarrow d'|d_{i+2}$$

aus $(**)$ folgt $d'|d_{n-1} = d$

4. zeige $d = x \cdot a + y \cdot b$:

Wir zeigen $d_i = x_i \cdot a + y_i \cdot b$ für $i = 0, \dots, n-1$ mittels Induktion:

(IA) $i = 0, i = 1$: klar \checkmark

$$(IS) \ i \rightarrow i+1: x_{i+1} \cdot a + y_{i+1} \cdot b = (x_{i-1} - x_i \left\lfloor \frac{d_{i-1}}{d_i} \right\rfloor) \cdot a + (y_{i-1} - y_i \left\lfloor \frac{d_{i-1}}{d_i} \right\rfloor) \cdot b =$$

$$x_{i-1} \cdot a + y_{i-1} \cdot b - \left\lfloor \frac{d_{i-1}}{d_i} \right\rfloor \cdot (x_i \cdot a + y_i \cdot b) \stackrel{IV}{=} d_{i-1} - \left\lfloor \frac{d_{i-1}}{d_i} \right\rfloor \cdot d_i = d_{i-1} \% d_i = d_{i+1}$$

Die Aussage folgt wegen $d = d_{n-1} \cdot x = x_{n-1}$ und $y = y_{n-1}$

5. zeigen polynomielle Laufzeit:

Wir zeigen: $d_{i+2} \leq \frac{1}{2}d_i$ für $i \geq 1 \quad (***)$

Fallunterscheidung:

1) $d_{i+1} \leq \frac{1}{2}d_i$: Aussage folgt aus $d_{i+2} < d_{i+1}$

2) $d_{i+1} > \frac{1}{2}d_i$: Aus $d_{i+1} < d_i$ folgt $\frac{1}{2}d_i < d_{i+1} < d_i < 2 \cdot d_{i+1}$

$$\Rightarrow d_{i+2} = d_i \% d_{i+1} = d_i - d_{i+1} < d_i - \frac{1}{2}d_i = \frac{1}{2}d_i$$

Aus $(***)$ folgt, dass sich die d_i nach spätestens 2 Iterationen halbiert haben \Rightarrow polynomielle Laufzeit

Die Anzahl der Iterationen ist logarithmisch in $d_1 = b$

\Rightarrow polynomiell in $|b|$

□

6 6. Übung

6.1 Aufgabe 1

$$\begin{aligned}\varphi(75) &= \varphi(3) \cdot \varphi(5^2) = (3^1 - 3^0)(5^2 - 5^1) = 2 \cdot 20 = 40 \\ \varphi(408783) &= \varphi(11) \cdot \varphi(23) \cdot \varphi(2011) = 442200\end{aligned}$$

6.2 Aufgabe 3

- a) \mathbb{Z}_p^* hat genau $\varphi(\varphi(p)) = \varphi(p-1)$ Erzeuger.
 Habe $\varphi(p-1) = p-1$. Bemerke für $p \neq 2, 3$: $2|p-1 \Rightarrow \varphi(p-1) < p-1 \Rightarrow p \in P \setminus \{2, 3\}$ lösen diese Gleichung nicht.
 $1, \dots, \underbrace{p-1}_{\geq 2}$ Rechnung für $p \in \{2, 3\}$ zeigt, dass nur $p = 2$ eine Lösung der Gleichung ist.
 $\varphi(p-1) = |\{a \in \{1, \dots, p-1\} | ggT(a, p-1) = 1\}|$. Da $p \in \mathbb{P} \setminus \{2, 3\} \Rightarrow |\{1, \dots, p-1\}|$ und $2 \in \{1, \dots, p-1\}$
- b) $\varphi(p-1) = p-2$
 Zum einen $\mathbb{N}_+ \ni \varphi(p-1) = p-2 \Rightarrow p \geq 3$.

- $p = 3$: $\varphi(p-1) = \varphi(2) = 1 = p-2 \checkmark$
- $p > 3$: $\varphi(p-1) = |\{i \in \mathbb{N} | 1 \leq i \leq p-1 \text{ mit } \underbrace{ggT(i, p-1)}_{2, p-1 \notin} = 1\}| \leq p-3$

Tipp: p ist Primzahl ungleich 2 \Rightarrow ungerade $\Rightarrow p-1$ gerade $\Rightarrow 2|p-1$

- c) $\varphi(p-1) = \frac{1}{3}(p-1)$
 Wegen $\varphi(p-1) \in \mathbb{N}_+$ folgt $3|p-1$. Wie in a): $2|p-1$
 Also schreibe: $p-1 = 2^{n_2} \cdot 3^{n_3} \cdot q$ mit $q \in \mathbb{N}_+, 2 \nmid q, 3 \nmid q$ und $n_2, n_3 \in \mathbb{N}_+$.
 $\varphi(p-1) = \varphi(2^{n_2}) \cdot \varphi(3^{n_3}) \cdot \varphi(q) = 2^{n_2-1} \cdot (2-1) \cdot 3^{n_3-1} \cdot (3-1) \cdot \varphi(q) = 2^{n_2} \cdot 3^{n_3-1} \cdot \varphi(q) = \frac{1}{3}(p-1) = 2^{n_2} \cdot 3^{n_3-1} \cdot q \Rightarrow \varphi(q) = q \Rightarrow q = 1$
 $\Rightarrow p = 2^{n_2} \cdot 3^{n_3} + 1$, also gilt die Gleichung für $p \in \{2^k \cdot 3^l + 1 | k, l \in \mathbb{N}_+\}$

6.3 Bonusaufgabe

Für jedes $\varepsilon > 0$ soll ein Algorithmus angegeben werden, der bei Eingabe $x \geq 2$ eine Zahl y berechnet mit $\frac{\varphi(x)-y}{\varphi(x)} \leq \varepsilon$.

Eingabe: $x \in \mathbb{N}, n = \log x$

1. $Q = \{p | p \leq n \text{ und } p \text{ prim}\}$
2. Zerlege $x = \underbrace{q_1^{e_1} \cdot q_2^{e_2} \cdot \dots \cdot q_k^{e_k}}_{=x_1} \cdot x_2$ mit $q_i \in Q$ und x_2 hat keine Teiler aus Q
3. $\varphi(x_1) = (q_1^{e_1} - q_1^{e_1-1}) \cdot \dots \cdot (q_k^{e_k} - q_k^{e_k-1})$
4. return $x_2 \cdot \varphi(x_1)$

Sei y die Ausgabe des Algorithmus, d.h. $y = x_2 \cdot \varphi(x_1)$.

Falls $x_2 = 1$ wird korrekter Wert ausgegeben, nehmen im Folgenden also $x_2 > 1$ an.

Aus $ggT(x_1, x_2) = 1$ folgt: $\varphi(x) = \varphi(x_1) \cdot \varphi(x_2) \leq x_2 \cdot \varphi(x_1)$, der Algorithmus liefert also keine zu kleinen Werte aus.

\Rightarrow genügt zu zeigen, dass $\varepsilon \geq \frac{y - \varphi(x)}{\varphi(x)} = \frac{x_2 \cdot \varphi(x_1) - \varphi(x_1) \cdot \varphi(x_2)}{\varphi(x_1) \cdot \varphi(x_2)} = \frac{x_2 - \varphi(x_2)}{\varphi(x_2)}$

$\Leftrightarrow \varepsilon \cdot \varphi(x_2) \geq x_2 \cdot \varphi(x_2)$

$\Leftrightarrow (1 + \varepsilon) \cdot \varphi(x_2) \leq x_2$ (*)

Sei $x_2 = p_1^{d_1} \cdots p_m^{d_m}$ die Primfaktorzerlegung von x_2 ($m \geq 1$), $p_1 \cdots p_m > n$

$\Rightarrow m \leq \frac{\log x_2}{\log n}$ (andernfalls $x_2 \geq p_1 \cdots p_m > n^m > n^{\log x_2 / \log n} = 2^{\log n \frac{\log x_2}{\log n}} = x_2$)

Widerspruch

$$\varphi(x_2) = p_1^{d_1} \cdot \underbrace{\left(1 - \frac{1}{p_1}\right)}_{> 1 - \frac{1}{n}} \cdot p_2^{d_2} \cdot \underbrace{\left(1 - \frac{1}{p_2}\right)}_{> 1 - \frac{1}{n}} \cdots p_m^{d_m} \cdot \underbrace{\left(1 - \frac{1}{p_m}\right)}_{> 1 - \frac{1}{n}} > x_2 \cdot \underbrace{\left(1 - \frac{1}{n}\right)^m}_{< 1}$$

$$\geq x_2 \cdot \left(1 - \frac{1}{n}\right)^{\log x_2 / \log n} > x_2 \cdot \left(1 - \frac{1}{n}\right)^{\log x / \log n}$$

$$= x_2 \cdot \left(1 - \frac{1}{n}\right)^{n / \log n} = \left[\left(1 - \frac{1}{n}\right)^n\right]^{1 / \log n} \cdot x_2 \text{ geht gegen } \frac{1}{e} \text{ für } n \rightarrow \infty$$

$$> \left(\frac{1}{4}\right)^{1 / \log n} \cdot x_2 \text{ für genügend große } n > x_2 \cdot \frac{1}{1 + \varepsilon} \text{ für genügend große } n$$

\Rightarrow dies zeigt (*)

7 7. Übung

7.1 Aufgabe 1

B erzeugt Schlüssel

↓ sendet p, g, B an A

A wählt zufällig a

↓

Gemeinsamer Schlüssel $B^a \bmod p$

↓ A sendet A an B

7.2 Aufgabe 2

Für jedes $n \geq 2$ enthält $\{1, \dots, 2^n\}$ mindestens $\lfloor \frac{n}{\log_2 n} \rfloor$ Primzahlen.

Angenommen, es gäbe $n \geq 2$, sodass in $\{1, \dots, 2^n\}$ genau $k < \lfloor \frac{n}{\log_2 n} \rfloor$ viele Primzahlen $p_1 \leq p_2 \leq \dots \leq p_k \leq$ liegen.

Definiere $\varphi : \{1, \dots, 2^n - 1\} \rightarrow \{0, \dots, n-1\}^k$ mit $x \mapsto (e_1, \dots, e_k)$ mit $x = \prod_{i=1}^k p_i^{e_i}$.

φ ist injektiv, denn aus $\varphi(x) = \varphi(y) = (e_1, \dots, e_k)$ folgt, $y = \prod p_i^{e_i} = x$.

$$\Rightarrow \underbrace{|\{1, \dots, 2^n - 1\}|}_{=2^n-1} \leq |\{0, \dots, n-1\}^k| = n^k < n^{\lfloor \frac{n}{\log_2 n} \rfloor} \leq n^{\frac{n}{\log_2 n}} = (2^{\log_2 n})^{\frac{n}{\log_2 n}} =$$

2^n

$\Rightarrow n^k = 2^n - 1 \Rightarrow \{0, \dots, n-1\}^k$ und $\{1, \dots, 2^n - 1\}$ sind gleichmächtig $\Rightarrow \varphi$ surjektiv.

$\Rightarrow \exists x \in \{1, \dots, 2^n - 1\} : \varphi(x) = (n-1, n-1, 0, 0, \dots, 0)$.

Aus $n \geq 2 \Rightarrow \{2, 3\} \subseteq \{1, \dots, 2^n\} \Rightarrow k \geq 2$, da 2, 3 prim.

$$\Rightarrow x = p_1^{n-1} \cdot p_2^{n-1} \cdot p_3^0 \cdots p_k^0 = 2^{n-1} \cdot \underbrace{3^{n-1}}_{\geq 3 \geq 2} \geq 2^n,$$

ein Widerspruch zu $x \in \{1, \dots, 2^n - 1\}$

7.3 Hinweise zur 8. Übung

$$\log_b a = \frac{\log_x a}{\log_x b}$$

8 8. Übung

8.1 Aufgabe 1

$$B \equiv g^b \equiv g^{p-1-x} \equiv g^{p-1} \cdot g^{-x} \equiv g^{-x} \equiv (g^{-1})^x \pmod{p}$$

x hat höchstens 4 Einsen in Binärdarstellung.

$$x = 2^i + 2^j + 2^k + 2^l$$

$$h^x = h^{2^i+2^j+2^k+2^l} = h^{2^i} \cdot h^{2^j} \cdot h^{2^k} \cdot h^{2^l}$$

8.2 Zusatzaufgabe

$\log_{g,p}$ konnte man effizient berechnen. Man will: $\log_{h,p}$ berechnen. (p Prim, g, h Erzeuger in \mathbb{Z}_p^*)

Behauptung: $\log_{n,p} x = (\log_{g,p} x \cdot \log_{g,p} h^{-1}) \pmod{p-1}$

Beweis: Zeigen, dass $\log_{g,p} h^{-1}$ existiert, d.h.

$$ggT(\log_{g,p} h, p-1) = 1 (\Leftrightarrow \log_{g,p} h \in \mathbb{Z}_{p-1}^*)$$

$$\text{wäre } ggT(\log_{g,p} h, p-1) = d > 1. \text{ Dann gilt: } h^{\frac{p-1}{d}} \equiv (g^{\log_{g,p} h})^{\frac{p-1}{d}} \equiv \underbrace{(g^{\frac{\log_{g,p} h}{d}})^{p-1}}_{\in \mathbb{Z}_p^*} \equiv$$

$$1 \pmod{p} \Rightarrow \text{ord}_p h \leq \frac{p-1}{d} < p-1 = \text{ord}_p h, \text{ Widerspruch.}$$

$$x \equiv y \pmod{p-1} \Rightarrow z^x \equiv z^y \pmod{p}$$

$$x = y + k(p-1) \Rightarrow z^x = z^y \cdot z^{k(p-1)} = z^y \underbrace{(z^k)^{p-1}}_{\equiv 1 \pmod{p}} \equiv z^y \pmod{p}$$

$$h^{\log_{g,p} x \cdot \log_{g,p} h^{-1} \pmod{p-1}} \equiv h^{\log_{g,p} x \cdot \log_{g,p} h^{-1}} \equiv g^{\overbrace{\log_{g,p} h \cdot \log_{g,p} h^{-1}}^{\equiv 1 \pmod{p-1}} \cdot \log_{g,p} x} \equiv g^{1 \cdot \log_{g,p} x} \equiv x \pmod{p}$$

8.3 Extra

- Alice entscheidet sich für Kopf oder Zahl
- Bob wirft eine Münze

\Rightarrow Gewinner steht fest.

Alice wählt Primzahlen p, q mit $p \equiv q \equiv 3 \pmod{4}$

Rightarrow Alice sendet $n = p \cdot q$

Alice entscheidet sich für Kopf (1) oder Zahl (0) $\rightarrow b$

\Rightarrow Alice sendet $c = -1^b \cdot r^2 \pmod{n}$ für ein beliebiges $r \in \mathbb{Z}_n^*$

Bob wirft die Münze $\rightarrow x \in \{0, 1\}$

\Leftarrow Bob sendet Ergebnis x

Alice weiß jetzt, ob sie gewonnen hat.

\Rightarrow Alice schickt p, q

Bob kann c entschlüsseln und erhält b

\Rightarrow Beide kennen den Gewinner.

9. Übung

9.1 Aufgabe 1

def $A(x, \text{gamma}, p)$:
 return $x + p - 1$

Soll ein y berechnen mit $x \neq y$ und aber $f_{\gamma,p}(x) = f_{\gamma,p}(y)$.
 $f_{\gamma,p}(x) = \gamma^x \bmod p = \gamma^x \cdot 1 \bmod p = \gamma^x \cdot \gamma^{p-1} \bmod p = \gamma^{x+p-1} \bmod p =$
 $f_{\gamma,p}(x+p-1) = f_{\gamma,p}(A(x, \gamma, p))$

9.2 Aufgabe 3

Allgemeines zu den Komplexitätsklassen P und NP :

Sei $A \subseteq \mathbb{N}$. Die charakteristische Funktion c_A von A sei

$$c_A(x) = \begin{cases} 1 & \text{falls } x \in A \\ 0 & \text{sonst.} \end{cases}$$

$A \in P \Leftrightarrow c_A$ ist in polynomieller Zeit berechenbar.

$A \in NP \Leftrightarrow c_A$ ist von einer nichtdeterministischen Turing-Maschine in polynomieller Zeit akzeptiert wird.

Sei $P = NP$ und $S = (\Sigma, K, \mathcal{E}, D)$ ein Kryptosystem. Wir wollen einen Algorithmus $A(1^n, e, c)$, der uns $m \in \Sigma^n$ liefert, so dass $P(c = E(e, m)) > 0$. D.h. c ist eine mögliche Ausgabe von $\mathcal{E}(e, m)$.

Definiere die Menge $M = \{ \langle e, c, w \rangle \mid e, c, w \in \Sigma^* \text{ und es gibt ein } m \in \Sigma^* \text{ mit } |m| = |w| \text{ und } m \leq w \text{ in lexikographischer Ordnung, sodass es in der Berechnung von } \mathcal{E}(e, m) \text{ einen Rechenweg gibt, bei dem } \mathcal{E}(e, m) = c \text{ gilt} \}$

Es gilt $M \in NP$, denn folgender nichtdeterministischer Algorithmus entscheidet M :

Eingabe: (e, c, w) : Gehe nichtdeterministisch alle $m \cong w$, $|m| = |w|$ durch:

Fasse \mathcal{E} als nichtdeterministische Turingmaschine auf und durchlaufe alle Berechnungswege von $\mathcal{E}(e, m)$, ist $\mathcal{E}(e, m) = c$, gib $TRUE$ zurück.

$$M \in NP \stackrel{P=NP}{\Rightarrow} M \in P$$

Definiere $A(1^n, e, c)$:

Bestimme m zeichenweise mit $(m = m_1, \dots, m_n)$. Stelle Anfragen der Form " $\langle e, c, w \rangle \in M$?" mit $w = 1 \cdot k^{n-1}, w = 2 \cdot k^{n-1}, w = (k-1) \cdot k^{n-1}$

Nebenbemerkung: Ist $m_1 = q$, dann ist $\langle e, c, q \cdot k^{n-1} \rangle \in M$, aber $\langle e, c, (q-1) \cdot k^{n-1} \rangle \notin M$

Ist $\langle e, c, q \cdot k^{n-1} \rangle \in M$ und $\langle e, c, (q-1) \cdot k^{n-1} \rangle \notin M$, dann setze $m_1 = q$. Bestimme nun m_2 durch Anfragen " $\langle e, c, w \rangle \in M$?" mit $w = m_1 \cdot 1 \cdot k^{n-2}, \dots, w = m_1 \cdot (k-1) \cdot k^{n-2}, w = m_1 \cdot k^{n-1}$. Führt man dies so fort, hat man m komplett bestimmt und gibt dies zurück.

$\Rightarrow A$ ist deterministischer Polynomialzeit-Algorithmus.

9.3 Bonusaufgabe

Kontraposition: Sei $S = (\Sigma, K, \mathcal{E}, D)$ ein Kryptosystem und erfülle nicht die Einwegeneigenschaft. Das heißt, es existiert ein probabilistischer polynomieller

Algorithmus A , sodass für unendlich viele $n \in \mathbb{N}$ und ein Polynom q , sodass die Erfolgswahrscheinlichkeit bei folgendem Verfahren mindestens $\frac{1}{q(n)}$ beträgt.

- a) Bob erzeugt Schlüsselpaar (e, d) mit $K(1^n)$.
- b) Bob wählt zufällig $m \in \Sigma^*$, berechnet c als Ausgabe von $\mathcal{E}(e, m)$.
- c) A ist erfolgreich, wenn $A(1^n, e, c)$ m ausgibt.

Nicht Semantische Sicherheit: es existiert ein propabilistischer polynomialzeit Algorithmus B und ein Polynom p , sodass für unendlich viele $n \in \mathbb{N}$ gilt: Die Erfolgswahrscheinlichkeit von folgendem Protokoll ist mindestens $\frac{1}{2} + \frac{1}{p(n)}$:

- a) wie oben...
- b) Alice wählt $m_1, m_2 \in \Sigma^n$ und sendet diese an Bob.
- c) Bob wählt $m \in \{m_1, m_2\}$ aus und berechnet die Ausgabe c der Verschlüsselungsfunktion und sendet c an Alice.
- d) Alice muss mit B entscheiden, ob $m \stackrel{?}{=} m_1$.

Beschreibung für den Algorithmus B :

- berechne Ausgabe m^* von $A(1^n, c)$
- falls $m^* \in \{m_1, m_2\}$, gebe m^* zurück
- sonst gebe zufällig eines der beiden Elemente aus

Die Wahrscheinlichkeit, dass sich A irrt sei $E(P_1)$. Irrt sich der Algorithmus, ist die Wahrscheinlichkeit, dass Alice das richtige Element rät $\frac{1}{2}$.

$$P(\text{Alice errät das Richtige}) = P(E_1) + \frac{1}{2}P(\overline{E_1}) = P(E_1) + \frac{1}{2}[1 - P(E_1)] = \frac{1}{2} + \frac{1}{2}P(E_1) \geq \frac{1}{2} + \underbrace{\frac{1}{2 \cdot q(n)}}_{:=p(n)}$$

⌋⌋ Diese Lösung hat sich in der Übung als möglicherweise Falsch herausgestellt, da eine der beiden Nachrichten deutlich häufiger als die andere zurückgegeben werden könnte...⌋⌋