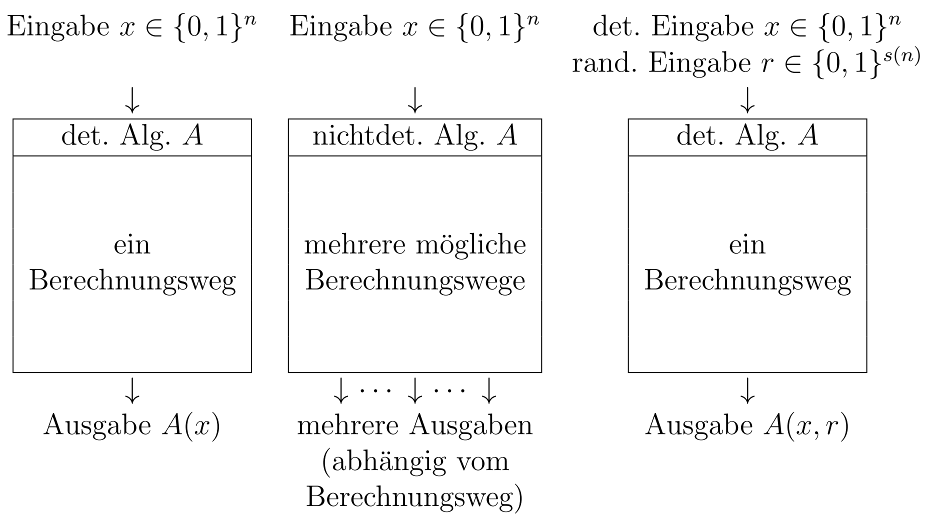
8 Komplexitätstheorie und Kryptologie

- Verschlüsselung, Authentisierung, ... müssen schnell berechenbar sein.
  Formal: polynomiell zeitbeschränkte Funktionen/Algorithmen

- Angreifer hat beschränkte Ressourcen. Formal: polynomiell zeitbeschränkt

Ziel dieses Abschnittes: Wenn $P = NP$, gibt es keine sichere Kryptographie.
Das $P \stackrel{?}{=} NP$ Problem ist eines der bekanntesten offenen Probleme der TI

Erinnerung Algorithmen (deterministisch und nichtdeterministisch)



2 und 3 sind nur verschiedene Interpretationen des selben Modells
(in r ist Berechnungsweg kodiert)

Pol. Laufzeit: A benötigt für $x \in \{0,1\}^n$ nur $p(n)$ Schritte (p Polynom)
(unabhängig vom Berechnungsweg oder Zufall)

Erinnerung Komplexitätsklassen P und NP

- Entscheidungsprobleme: Probleme, die nur ja/nein-Antworten haben
  - Beispiel 1 (Clique):

    Eingabe: Graph $G = (V, E)$ und Zahl $k \in \mathbb{N}$

    Frage: Gibt es in G eine Clique $\geq k$?
  - Beispiel 2 Wortproblem $L \subseteq \{0,1\}^*$:

    Eingabe: $x \in \{0,1\}^*$

    Frage: Gilt $x \in L$?
- P: Probleme, für die es einen det. pol. Alg. A gibt mit
  - $x \in L \Longrightarrow A(x) = 1$
  - $x \notin L \Longrightarrow A(x) = 0$
- NP: Probleme, für die es einen randomisierten pol. Alg. gibt mit
  - $x \in L \Longrightarrow$ es existiert $r \in \{0,1\}^{s(n)}$ mit $A(x, r) = 1$
  - $x \notin L \Longrightarrow$ für alle $r \in \{0,1\}^{s(n)}$ gilt $A(x, r) = 0$

Pseudozufallszahlengeneratoren (PRNG): z.B. für Stromchiffren

Definition 8.1. $R : \{0,1\}^n \longrightarrow \{0,1\}^{l(n)}$ ist sicherer PRNG, wenn gilt

- R ist in polynomieller Zeit berechenbar
- Angreifer A stellt keinen Unterschied fest zwischen
  - $y \in \{0,1\}^{l(n)}$ (gewählt bzgl. Gleichverteilung) und
  - $R(x) \in \{0,1\}^{l(n)}$ (x gewählt bzgl. Gleichverteilung in $\{0,1\}^n$)

Praktisch nicht möglich: nicht lösbar in polynomieller Zeit.

Gute Verallgemeinerung OTP:

- Schlüssel $k \in \{0,1\}^n$, Nachricht $m \in \{0,1\}^{l(n)}$
- Verschlüsselung $c = m \oplus R(k)$
- Für pol. Angreifer: $R(k)$ ist gleichverteilter Zufallswert
  Also genauso sicher wie das originale OTP

Etwas formaler:

- Angreifer ist pol. Algorithmus
- Angreifer muss für $y \in \{0,1\}^{l(n)}$ entscheiden, ob $y \in L := R(\{0,1\}^n)$
- Frage: Ist die formale Sprache L aus P?
  Dann ex. ein pol. zeitbeschränkter Alg., um dieses Problem zu lösen

Satz 8.2. *Unter der Voraussetzung $P = NP$ gibt es keine sicheren PRNG.*

Proof. Zu zeigen: L ist in P. Erster Schritt: Zeige $L \in NP$.

Betrachte folgenden randomisierten Algorithmus A
Eingabe $(\underbrace{y \in \{0,1\}^{l(n)}}_{deterministische}, \underbrace{x \in \{0,1\}^n}_{randomisierte})$:

1. Berechne $R(x)$ (polynomielle Laufzeit)
2. Prüfe $y \stackrel{?}{=} R(x)$ (polynomiell Laufzeit)
3. Gib Ergebnis aus.

A hat polynomielle Laufzeit. Weiter gilt

- Ist $y \in L$, so ex. x mit $A(y,x) = 1$
- Ist $y \notin L$, so gilt f.a. x: $A(y,x) = 0$.

Also gilt $L \in NP$. Wegen $P = NP$ gilt dann auch $L \in P$. □

Definition 8.3. $f : \{0,1\}^* \longrightarrow \{0,1\}^*$ heißt sichere Einwegfunktion, wenn es einen pol. Alg. A gibt, der f berechnet, es aber praktisch nicht möglich ist, Urbilder zu berechnen.

Beispiele für Einwegfunktionen:

- RSA: $(p,q) \mapsto p \cdot q$ — Faktorisierungsproblem
- DSA: $x \mapsto g^x$ — Diskretes Logarithmusproblem
- Hashfunktionen

Satz 8.4. *Gilt $P = NP$, so ex. keine sicheren Einwegfunktionen.*

Für $f : \{0,1\}^* \longrightarrow \{0,1\}^*$ schreiben wir auch $f : \{0,1\}^n \longrightarrow \{0,1\}^{s(n)}$ wenn $|f(x)| \leq s(|x|)$. Da f in pol. Zeit berechenbar ist, ist auch s durch ein Polynom beschränkt.

Proof. Sei $f : \{0,1\}^n \longrightarrow \{0,1\}^{s(n)}$ in Polynomialzeit berechenbar.

Gesucht: Alg., der zu $y \in \{0,1\}^{s(n)}$ ein Urbild berechnet.

Dazu: Untersuchung der folgenden Sprache:
$L = \{(y,(x_1,\dots,x_k) \in \{0,1\}^{s(n)} \times \{0,1\}^k; \exists x_{k+1},\dots,x_n : f(x_1,\dots,x_n) = y\}$
(Lässt sich $(x_1,\dots,x_k)$ zu einem Urbild $(x_1,\dots,x_k,\dots,x_n)$ von y erweitern?)

Wenn $L \in P$, können wir dies in pol. Zeit beantworten.

L ist aus NP:

Algorithmus A, Eingabe: $(\underbrace{(y,x_1,\dots,x_k)}_{deterministisch},\underbrace{(z_1,\dots,z_{n-k})}_{randomisiert})$

1. Berechne $y' = f(x_1,\dots,x_k,z_1,\dots,z_{n-k})$
2. Prüfe $y \stackrel{?}{=} y'$

A ist pol., da f pol. ist. Wegen P=NP ex. pol. Alg. B für L.

Mit B lässt sich pol. Alg. zur Berechnung des Urbildes konstruieren

Algorithmus Urbild für f, Eingabe $y \in \{0,1\}^{s(n)}$

1. Prüfe, ob y ein Urbild hat (in Polynomialzeit möglich)
2. for $i = 1$ to n do
3.   if $B(y, x_1, \dots, x_{i-1}, 0) = 1$
4. $x_i = 0$
5.   else $x_i = 1$
6.   fi
7. od
8. return $(x_1, \dots, x_n)$

Laufzeit: Polynomiell, da B polynomiell (Erhöhung um eine Potenz) $\square$

Allgemein: Folgende Fragestellungen sind äquivalent zu $P \stackrel{?}{=} NP$

1. Decision-Version:
   Existiert für jeden randomisierten polynomiellen Algorithmus
   $$A : \{0,1\}^n \times \{0,1\}^{s(n)} \longrightarrow \{0,1\}$$
   ein polynomieller Algorithmus
   $$B : \{0,1\}^n \longrightarrow \{0,1\}$$
   mit
   $$B(x) = \begin{cases} 1, & \text{falls } A(x,z) = 1 \text{ für ein } z \in \{0,1\}^{s(n)} \\ 0, & \text{falls } A(x,z) = 0 \text{ für alle } z \in \{0,1\}^{s(n)} \end{cases}$$
2. Search-Version:
   Existiert für jeden randomisierten polynomiellen Algorithmus
   $$A : \{0,1\}^n \times \{0,1\}^{s(n)} \longrightarrow \{0,1\}$$
   ein polynomieller Algorithmus
   $$B : \{0,1\}^n \longrightarrow \{0,1\}^{s(n)}$$
   mit: aus $A(x,z) = 1$ für ein $z \in \{0,1\}^{s(n)}$ folgt $A(x, B(x)) = 1$.

Ausblick

- Formale Definition von sicher: Erfolg-Zeit-Quotient klein.
- Formale Definition von Angreifer: polynomiell. zeitbeschränkter Algorithmus.
- Reduktionen:
  - Konstruktion von Verschlüsselungsfunktionen aus Pseudozufallszahlengeneratoren
  - Konstruktion von Public-Key-Verfahren aus Einwegfunktionen
  - ...
- Hier wichtig: Reduktionen müssen Sicherheitsparameter übertragen.