# Kryptographie

## Mit Bezug zur Komplexitätstheorie

Gerrit Gruben

Institut für Informatik
Fachbereich Mathematik und Informatik
Freie Universität Berlin

Algorithmik Seminar

15. Juni 2010

1. Einführung
2. Perfekte Geheimhaltung
   - Definition
   - Satz von Shannon und Vernams One-Time-Pad
3. Einwegfunktionen
   - Berechnungssicherheit
   - Einwegfunktionen
4. Pseudozufall
   - Pseudozufallsgeneratoren
   - Konstruktion aus Einwegpermutation
5. Zero-Knowledge Beweise
   - Einführung

# Grundprobleme

Probleme der Kryptographie:

1. **Datensicherheit** Schutz vor unbefugtem Lesen von Daten.
2. **Datenintegrität** Schutz vor ungewollter Modifikation der Daten.
3. **Authentifikation** Nachweis einer Identität in dem man
   1. Etwas weiß
   2. oder etwas besitzt
   3. oder etwas ist.

## Definition (Kryptosystem)

Ein Paar (**Enc**, **Dec**) von polynomiellen Funktionen

$$\mathbf{Enc} : \mathcal{K} \times \mathcal{M} \to \mathcal{C},\ (k, x) \mapsto \mathbf{Enc}_k(x)$$

und

$$\mathbf{Dec} : \mathcal{K} \times \mathcal{C} \to \mathcal{M},\ (k, y) \mapsto \mathbf{Dec}_k(y)$$

mit den Mengen $\mathcal{M}, \mathcal{C}, \mathcal{K}$ (**Klartext-**,**Chiffretext-** und **Schlüsselmenge**) heißt **Kryptosystem**, wenn zusätzlich die Bedingung gilt

$$\mathbf{Dec}_k \circ \mathbf{Enc}_k = \mathrm{Id}_{\mathcal{M}}\ (\forall k \in \mathcal{K})$$

## Definition

Ein Kryptosystem $(\mathbf{Enc}, \mathbf{Dec})$ über $\mathbb{B}^n$ bietet **perfekte Geheimhaltung**, falls für jede Verteilung auf $\mathcal{M}$ von Nachrichten, jeder Nachricht $x$ und jeden auftretenden Chiffretext $y$ gilt

$$\mathbf{Pr}[M = x \mid C = y] = \mathbf{Pr}[M = x]$$

Anders formuliert:

$$\mathbf{Enc}_{U_n}(x) \equiv \mathbf{Enc}_{U_n}(x') \; (x, x' \in \mathcal{M})$$

## Lemma

*Für ein Kryptosystem* $\Pi = (\mathbf{Enc}, \mathbf{Dec})$ *sind die folgenden Aussagen äquivalent*

1. $\Pi$ *bietet perfekte Geheimhaltung.*
2. *Für jede Verteilung auf* $\mathcal{M}$ *und allen* $x \in \mathcal{M}$, $y \in \mathcal{C}$ *gilt:*

$$\mathbf{Pr}[C = y \mid M = x] = \mathbf{Pr}[C = y].$$

3. *Für jede Verteilung auf* $\mathcal{M}$ *und allen* $x_0, x_1 \in \mathcal{M}$ *und* $c \in \mathcal{C}$ *gilt*

$$\mathbf{Pr}[C = y \mid M = x_0] = \mathbf{Pr}[C = y \mid M = x_1]$$

## Proof.

1. $\Leftrightarrow$ 2.: Einfache Umformung und Bayes.

1., 2. $\Leftrightarrow$ 3.:

'$\Rightarrow$': Folgt direkt aus 2..

'$\Leftarrow$': Sei eine beliebige Verteilung auf $\mathcal{M}$ und $x_0 \in \mathcal{M}$ und $y \in \mathcal{C}$ gewählt. Definiere $p := \mathbf{Pr}[C = y \mid M = m_0]$. Dann gilt

$$\begin{aligned}\mathbf{Pr}[C = y] &= \sum_{x \in \mathcal{M}} \mathbf{Pr}[C = y \mid M = x] \cdot \mathbf{Pr}[M = x] \\ &= \sum_{x \in \mathcal{M}} p \, \mathbf{Pr}[M = x] \\ &= p \\ &= \mathbf{Pr}[C = y \mid M = x_0]\end{aligned}$$

## Lemma

*Ein Kryptosystem* (**Enc**, **Dec**) *mit Klartextmenge* $\mathcal{M}$ *und Schlüsselmenge* $\mathcal{K}$ *bietet perfekte Geheimhaltung, dann gilt*

$$|\mathcal{K}| \geq |\mathcal{M}|$$

## Lemma

*Ein Kryptosystem* $(\mathbf{Enc}, \mathbf{Dec})$ *mit Klartextmenge* $\mathcal{M}$ *und Schlüsselmenge* $\mathcal{K}$ *bietet perfekte Geheimhaltung, dann gilt*

$$|\mathcal{K}| \geq |\mathcal{M}|$$

## Proof.

Annahme: $|\mathcal{K}| < |\mathcal{M}|$. Sei $M$ gleichverteilt und $y \in \mathcal{C}$ ein Chiffretext, der auftritt. Definiere

$$M(c) := \{x \mid x = \mathbf{Dec}_k(y), k \in \mathcal{K}\}$$

Es gilt $|M(y)| \leq |\mathcal{K}|$ da wir jedem Element in $M(c)$ ein Schlüssel zuordnen können. Da $|\mathcal{K}| < |\mathcal{M}|$ existiert ein $x \in M \backslash M(y)$ für das gilt

$$\mathbf{Pr}[M = x \mid C = y] = 0 < \mathbf{Pr}[M = x]$$

im Widerspruch zur perfekten Geheimhaltung. □

## Theorem (von Shannon)

*Sei* $(\mathbf{Enc}, \mathbf{Dec})$ *ein Kryptosystem mit* $|\mathcal{K}| = |\mathcal{M}| = |\mathcal{C}|$. *Dann bietet das Kryptosysteme perfekte Geheimhaltung genau dann wenn folgende Bedingungen gelten:*

1. *Die Schlüssel werden gleichverteilt gewählt, d. h.* $K \sim U_{\mathcal{K}}$.
2. *Für alle Nachrichten* $x \in M$ *und Chiffretexten* $y \in C$ *existiert genau ein* $k \in K$ *mit* $y = \mathbf{Enc}_k(x)$.

## Definition (One-Time-Pad)

Seien $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^n$ für ein $n \in \mathbb{N}$. Dann heisst $(\mathbf{Enc}, \mathbf{Dec})$

$$\mathbf{Enc}_k(x) := x \oplus k,\ \mathbf{Dec}_k(y) := y \oplus k$$

**One-Time-Pad** (Vernam Chiffre).

# Zusammenfassung

1. Satz von Shannon $\Rightarrow$ One-Time-Pad perfekt geheim.
2. One-Time-Pad ist minimal (bzgl. $\mathcal{K}$ und $\subseteq$) mit der Eigenschaft der perfekten Geheimhaltung.
3. Ungünstige Lösung: Schlüssellänge $=$ Nachrichtenlänge, wozu dann den Schlüsselaustausch?

# Zusammenfassung

1. Satz von Shannon $\Rightarrow$ One-Time-Pad perfekt geheim.
2. One-Time-Pad ist minimal (bzgl. $\mathcal{K}$ und $\subseteq$) mit der Eigenschaft der perfekten Geheimhaltung.
3. Ungünstige Lösung: Schlüssellänge = Nachrichtenlänge, wozu dann den Schlüsselaustausch?

**Ergebnis**: Es werden lockere Definition von Sicherheit benötigt.

# Die moderne Kryptographie kommt

1. Die Forderungen an die Sicherheit eines Kryptosystems werden gelockert um praktikablere Chiffren zu konstruieren.

# Die moderne Kryptographie kommt

1. Die Forderungen an die Sicherheit eines Kryptosystems werden gelockert um praktikablere Chiffren zu konstruieren.
2. Mit Hilfe der Komplexitätstheorie werden die Angreifer modelliert: Polynomialzeitalgorithmen mit Zufallszahlen $\mathcal{PPT}$.

# Die moderne Kryptographie kommt

1. Die Forderungen an die Sicherheit eines Kryptosystems werden gelockert um praktikablere Chiffren zu konstruieren.
2. Mit Hilfe der Komplexitätstheorie werden die Angreifer modelliert: Polynomialzeitalgorithmen mit Zufallszahlen $\mathcal{PPT}$.
3. Die Sicherheit der moderne Kryptographie basiert auf die Schwierigkeit von gewissen Problemen (**Reduktionsprinzip**) $\rightarrow$ Einwegfunktionen.

# Die moderne Kryptographie kommt

1. Die Forderungen an die Sicherheit eines Kryptosystems werden gelockert um praktikablere Chiffren zu konstruieren.
2. Mit Hilfe der Komplexitätstheorie werden die Angreifer modelliert: Polynomialzeitalgorithmen mit Zufallszahlen $\mathcal{PPT}$.
3. Die Sicherheit der moderne Kryptographie basiert auf die Schwierigkeit von gewissen Problemen (**Reduktionsprinzip**) $\rightarrow$ Einwegfunktionen.
4. Zuerst aber ein Schockergebnis:

## Theorem

*Sei $\mathcal{P} = \mathcal{NP}$ und $(\mathbf{Enc}, \mathbf{Dec})$ in Polynomialzeit berechenbar mit einer Schlüssellänge von $n = n(m) < m$ bei Nachrichtenlänge $m$. Dann existiert ein Polynomialzeitalgorithmus $A$, so dass für alle Eingabelängen $m$ gilt: es gibt es ein Paar $x_0, x_1 \in \{0,1\}^m$ mit*

$$\Pr_{\substack{b \in_R \{0,1\} \\ k \in_R \{0,1\}}} [A(\mathbf{Enc}_k(x_b)) = b] \geq \frac{3}{4}$$

## Definition (Vernachlässigbare Funktionen)

Sei $\epsilon : \mathbb{N} \to [0,1] \subseteq \mathbb{R}$, dann heisst $\epsilon$ **vernachlässigbar**, wenn $\epsilon(n) = n^{-\omega(1)}$. D. h. für alle $c > 0$ existiert ein $N \in \mathbb{N}$, sodass $\epsilon(n) < n^{-c}$ für alle $n \geq N$ gilt.

## Definition

**Berechnungssicherheit** bedeutet für ein Kryptosystem (**Enc**, **Dec**) mit Schlüssellänge $n$ und Eingabelänge $m$, dass wir für alle $A \in \mathcal{PPT}$ eine vernachlässigbare Funktion $\epsilon$ haben, so dass ($x_i$ bezeichne das $i$-te Bit von der Nachricht $x$):

$$\Pr_{\substack{k \in_R \{0,1\}^n \\ x \in_R \{0,1\}^m}} [A(E_k(x)) = (i, b) \text{ s. d. } x_i = b] \leq \frac{1}{2} + \epsilon(n)$$

gilt. Das heisst über kein Bit der Nachricht $x$ kann ein Angreifer mit nicht vernachlässigbarer Wahrscheinlichkeit Informationen in Polynomialzeit errechnen.

## Definition (Einwegfunktionen)

Eine in Polynomialzeit berechenbare Funktion $f : \mathbb{B}^* \to \mathbb{B}^*$ heißt **Einwegfunktion**, falls für alle $A \in \mathcal{PPT}$ gilt:

$$\Pr_{\substack{x \in_R \mathbb{B}^n \\ y = f(x)}} \left[ A(y) = x' \text{ mit } f(x') = y \right]$$

ist vernachlässigbar in $n \in \mathbb{N}$.

## Definition (Einwegfunktionen)

Eine in Polynomialzeit berechenbare Funktion $f : \mathbb{B}^* \to \mathbb{B}^*$ heißt **Einwegfunktion**, falls für alle $A \in \mathcal{PPT}$ gilt:

$$\Pr_{\substack{x \in_R \mathbb{B}^n \\ y = f(x)}} \left[ A(y) = x' \text{ mit } f(x') = y \right]$$

ist vernachlässigbar in $n \in \mathbb{N}$.
Gilt $|f(x)| = |x|$ für $x \in \mathbb{B}^*$, so heisst $f$ **längenerhaltend**. Ist $f$ eine injektive , längenerhaltende Einwegfunktion, dann heisst $f$ **Einwegpermutation**.

## Definition (Einwegfunktionen)

Eine in Polynomialzeit berechenbare Funktion $f : \mathbb{B}^* \to \mathbb{B}^*$ heißt **Einwegfunktion**, falls für alle $A \in \mathcal{PPT}$ gilt:

$$\Pr_{\substack{x \in_R \mathbb{B}^n \\ y = f(x)}} \left[ A(y) = x' \text{ mit } f(x') = y \right]$$

ist vernachlässigbar in $n \in \mathbb{N}$.
Gilt $|f(x)| = |x|$ für $x \in \mathbb{B}^*$, so heisst $f$ **längenerhaltend**. Ist $f$ eine injektive , längenerhaltende Einwegfunktion, dann heisst $f$ **Einwegpermutation**.

**Vermutung**: Es existiert eine Einwegfunktion.

**Theorem**

*Wenn $\mathcal{P} = \mathcal{NP}$ dann existieren keine Einwegfunktionen.*

# Einwegfunktion?

**Eulersche $\varphi$-Funktion**:

$$\varphi(n) := |\{1 \leq j < n \mid \mathrm{ggT}(j,n) = 1\}| = \left|\left(\mathbb{Z}/n\mathbb{Z}\right)^*\right| =: |\,\mathbb{Z}_n^*\,|$$

Für $n = \prod_{i=1}^{r} p_i^{\nu_i}$ mit verschiedenen Primzahlen $p_i$ gilt:

$$\varphi(n) = \prod_{i=1}^{r} p_i^{\nu_i - 1}(p_i - 1)$$

# Einwegfunktion?

**Eulersche $\varphi$-Funktion**:

$$\varphi(n) := |\{1 \leq j < n \mid \text{ggT}(j,n) = 1\}| = \left|\left(\mathbb{Z}/n\mathbb{Z}\right)^*\right| =: |\,\mathbb{Z}_n^*\,|$$

Für $n = \prod_{i=1}^{r} p_i^{\nu_i}$ mit verschiedenen Primzahlen $p_i$ gilt:

$$\varphi(n) = \prod_{i=1}^{r} p_i^{\nu_i - 1}(p_i - 1)$$

## RSA-Kryptosystem

Für ein $n \in \mathbb{N}$ sei $N = N(n) \in \mathbb{N}$ eine zusammengesetzte Zahl mit $n$-Bits und $e \in \mathbb{N}$ mit $\text{ggT}(\varphi(n), e) = 1$. I. d. R. $N = p \cdot q$ und $p \neq q$ prim.

$$\textbf{Enc}_{(N,e)}(x) := RSA_{(N,e)}(x) := [x^e]_N \; (x \in {\mathbb{Z}/N\mathbb{Z}}^*)$$

# Einwegfunktion?

**Eulersche $\varphi$-Funktion**:

$$\varphi(n) := |\{1 \leq j < n \mid \mathrm{ggT}(j,n) = 1\}| = \left|\left(\mathbb{Z}/n\mathbb{Z}\right)^*\right| =: |\,\mathbb{Z}_n^*\,|$$

Für $n = \prod_{i=1}^{r} p_i^{\nu_i}$ mit verschiedenen Primzahlen $p_i$ gilt:

$$\varphi(n) = \prod_{i=1}^{r} p_i^{\nu_i - 1}(p_i - 1)$$

## RSA-Kryptosystem

Für ein $n \in \mathbb{N}$ sei $N = N(n) \in \mathbb{N}$ eine zusammengesetzte Zahl mit $n$-Bits und $e \in \mathbb{N}$ mit $\mathrm{ggT}(\varphi(n), e) = 1$. I. d. R. $N = p \cdot q$ und $p \neq q$ prim.

$$\mathbf{Enc}_{(N,e)}(x) := RSA_{(N,e)}(x) := [x^e]_N \ (x \in \mathbb{Z}/N\mathbb{Z}^*)$$

Der Empfänger berechnet ein geheim gehaltenes $d$ (**Privater Schlüssel**) mit $ed \equiv 1 \mod \varphi(N)$.

$$\mathbf{Dec}_{(N,e)}(y) := [y^d]_N$$

Kennt man den Wert von $\varphi(N)$, so lässt sich $d$ effizient mit dem euklidischen Algorithmus bestimmen.

# Und diese Funktion?

**Quadratische Reste**:

$$QR_n := \{z \in \mathbb{Z}_n^* \mid \exists y \in \mathbb{Z}_n^* : y^2 = z\}$$

# Und diese Funktion?

**Quadratische Reste**:

$$QR_n := \{z \in \mathbb{Z}_n^* \mid \exists y \in \mathbb{Z}_n^* : y^2 = z\}$$

## Rabin-Kryptosystem

Sei $N = PQ$, $P, Q > 2$ prim und $P, Q \equiv 3 \mod 4$. Verschlüsseln ist das Quadrieren $\mod N$ und dechiffrieren ist das Quadratwurzel ziehen in $\mathbb{Z}_N$.

# Und diese Funktion?

**Quadratische Reste**:

$$QR_n := \{z \in \mathbb{Z}_n^* \mid \exists y \in \mathbb{Z}_n^* : y^2 = z\}$$

## Rabin-Kryptosystem

Sei $N = PQ$, $P, Q > 2$ prim und $P, Q \equiv 3 \mod 4$. Verschlüsseln ist das Quadrieren $\mod N$ und dechiffrieren ist das Quadratwurzel ziehen in $\mathbb{Z}_N$.
Da der Empfänger die Faktorisierung von $N$ kennt, kann er mit Hilfe des chinesischen Restsatzes das Problem auf die simultane Quadratwurzelbestimmung in $\mathbb{Z}_P$ und $\mathbb{Z}_Q$ reduzieren, welches sich effizient lösen lässt ($\rightarrow$ Tafel).

## Definition (Levins universelle Einwegfunktion)

Es ist $f_{\mathcal{U}}$ **Levins universelle Einwegfunktion** wie folgt definiert: Die Eingabe wird geeignet zerteilt: $x = x_1 \cdots x_{\log n}$ ($n := |x|$) mit $|x_i| = \frac{n}{\log n}$ für $i = 1, \ldots, \log n$. Sei $(M_i)_{i \in \mathbb{N}}$ eine Abzählung aller Turingmaschinen. Dann gilt

$$f_{\mathcal{U}}(x) = M_1^{n^2}(x_1) \cdots M_{\log n}^{n^2}(x_{\log n})$$

wobei

$$M_i^t(x) = \begin{cases} M_i(x), & M_i \text{ terminiert nach } \leq t \text{ Schritten.} \\ 0^{|x|}, & \text{sonst} \end{cases}$$

## Theorem

*Falls es eine Einwegfunktion gibt, dann ist $f_{\mathcal{U}}$ eine Einwegfunktion.*

**Theorem**

*Wenn Einwegfunktionen existieren, dann gibt es ein $c \in \mathbb{N}$, sodass es ein berechnungssicheres Kryptosystem* $(\mathbf{Enc}, \mathbf{Dec})$ *gibt, welches eine Schlüssellänge von $n$ und Nachrichtenlänge von $n^c$ hat.*

# Zusammenfassung

1. Einwegfunktionen sind leicht zu berechnen, schwer invertierbare Funktionen.

# Zusammenfassung

1. Einwegfunktionen sind leicht zu berechnen, schwer invertierbare Funktionen.
2. Existieren diese, so haben wir effizientere und sichere Kryptosysteme.

# Zusammenfassung

1. Einwegfunktionen sind leicht zu berechnen, schwer invertierbare Funktionen.
2. Existieren diese, so haben wir effizientere und sichere Kryptosysteme.
3. Sicherheit nun Komplexitätstheoretisch definiert und Resterfolgswahrscheinlichkeit eingeräumt (aber im vernachlässigbaren $\epsilon$ Bereich)

# Zusammenfassung

1. Einwegfunktionen sind leicht zu berechnen, schwer invertierbare Funktionen.
2. Existieren diese, so haben wir effizientere und sichere Kryptosysteme.
3. Sicherheit nun Komplexitätstheoretisch definiert und Resterfolgswahrscheinlichkeit eingeräumt (aber im vernachlässigbaren $\epsilon$ Bereich)
4. Klassische Kryptographie und die Mathematik liefern Kandidaten für Einwegfunktionen.

# Was ist Zufall?

1. **Kolmogorow-Komplexität**: $x \in \mathbb{B}^n$ ist zufällig, wenn es keine Turingmaschine mit Beschreibungslänge $< 0.99n$ gibt, welche bei leerer Eingabe $x$ ausgibt.

# Was ist Zufall?

1. **Kolmogorow-Komplexität**: $x \in \mathbb{B}^n$ ist zufällig, wenn es keine Turingmaschine mit Beschreibungslänge $< 0.99n$ gibt, welche bei leerer Eingabe $x$ ausgibt.
   Ungeeignet da Unentscheidbar!

# Was ist Zufall?

1. **Kolmogorow-Komplexität**: $x \in \mathbb{B}^n$ ist zufällig, wenn es keine Turingmaschine mit Beschreibungslänge $< 0.99n$ gibt, welche bei leerer Eingabe $x$ ausgibt.
   Ungeeignet da Unentscheidbar!
2. Ansatz der Statistiker: Erfüllen Zeichenkette die Gesetze der Statistik? Gesetz der großen Zahl z. B.

# Was ist Zufall?

1. **Kolmogorow-Komplexität**: $x \in \mathbb{B}^n$ ist zufällig, wenn es keine Turingmaschine mit Beschreibungslänge $< 0.99n$ gibt, welche bei leerer Eingabe $x$ ausgibt.
   Ungeeignet da Unentscheidbar!
2. Ansatz der Statistiker: Erfüllen Zeichenkette die Gesetze der Statistik? Gesetz der großen Zahl z. B.
   Es gibt kryptographisch unsichere Verteilungen, welche dieses erfüllen.

# Die Antwort der Kryptographen

## Definition (Pseudozufallsgenerator (PZG))

Seien $G : \mathbb{B}^* \to \mathbb{B}^*, \ell : \mathbb{N} \to \mathbb{N}$ in Polynomialzeit berechenbar und gelte $\ell(n) > n$ für alle $n \in \mathbb{N}$. $(G, \ell)$ heisst **Pseudozufallsgenerator** (kurz: **PZG**) mit Dehnung $\ell$, wenn gilt $|G(x)| = \ell(|x|)$ für alle $x \in \mathbb{B}^*$ und für alle $A \in \mathcal{PPT}$ existiert ein vernachlässigbares $\epsilon$, sodass

$$\left| \mathbf{Pr}\left[A(G(U_n)) = 1\right] - \mathbf{Pr}\left[A(U_{\ell(n)}) = 1\right] \right| < \epsilon(n) \; (n \in \mathbb{N})$$

gilt.

**Theorem (Einwegfunktionen $\Rightarrow$ PZG)**

*Existiert eine Einwegfunktion, dann existieren für alle Polynome $\ell$ mit $\ell(n) > n$ für $n \in \mathbb{N}$ ein PZG $(G, \ell)$.*

**Theorem (Einwegfunktionen $\Rightarrow$ PZG)**

*Existiert eine Einwegfunktion, dann existieren für alle Polynome $\ell$ mit $\ell(n) > n$ für $n \in \mathbb{N}$ ein PZG $(G, \ell)$.*

Im folgendem wird dieser Satz für Einwegpermutationen gezeigt.

## Definition (Unvorhersehbarkeit)

Sei $G : \mathbb{B}^* \to \mathbb{B}^*$ mit Dehnung $\ell$. $G$ und $\ell$ sind in Polynomialzeit berechenbar. $G$ heißt **unvorhersehbar**, wenn für alle $B \in \mathcal{PPT}$ gilt:

$$\Pr_{\substack{x \in_r \mathbb{B}^n \\ y = G(x) \\ i \in_R [\ell(n)]}} [B(1^n, y_1, \ldots, y_{i-1}) = y_i] \leq 1/2 + \epsilon(n)$$

mit vernachlässigbaren $\epsilon$ und $n \in \mathbb{N}$.

## Lemma (Yao)

*Sei $(G, \ell)$ ein PZG, dann existieren für alle $A \in \mathcal{PPT}$ ein $B \in \mathcal{PPT}$, so dass für alle $n \in \mathbb{N}$ und $\epsilon > 0$ aus* $\mathbf{Pr}\left[A(G(U_n))\right] - \mathbf{Pr}\left[A(U_{\ell(n)})\right] \geq \epsilon$, *folgt*

$$\mathop{\mathbf{Pr}}_{\substack{x \in_R \{0,1\}^n \\ y = G(x) \\ i \in_R [\ell(n)]}} \left[B(1^n, y_1, \ldots, y_{i-1}) = y_i\right] \geq 1/2 + \epsilon/\ell(n).$$

## Theorem ($G$ PZG $\Leftrightarrow$ unvorhersehbar)

*Seien $G : \mathbb{B}^* \to \mathbb{B}^*$ und $\ell : \mathbb{N} \to \mathbb{N}$ in Polynomialzeit berechenbar und $G$ habe die Dehnung $\ell$. Dann ist $(G, \ell)$ ein PZG genau dann wenn $G$ unvorhersehbar ist.*

## Proof.

## Theorem ($G$ PZG $\Leftrightarrow$ unvorhersehbar)

*Seien $G : \mathbb{B}^* \to \mathbb{B}^*$ und $\ell : \mathbb{N} \to \mathbb{N}$ in Polynomialzeit berechenbar und $G$ habe die Dehnung $\ell$. Dann ist $(G, \ell)$ ein PZG genau dann wenn $G$ unvorhersehbar ist.*

## Proof.

"$\Rightarrow$": Angenommen $(G, \ell)$ ist PZG und $n \in \mathbb{N}$. Wenn $y = (y_1, \ldots, y_{\ell(n)}) \in_R \mathbb{B}^{\ell(n)}$ zufällig gleichverteilt gewählt wurde, kann kein Bit vorhergesagt werden. Ist $G$ vorhersehbar, dann kann $y = G(x)$ von $y \in_R \mathbb{B}^{\ell(n)}$ unterschieden werden. Damit ist $G$ kein PZG.

## Theorem ($G$ PZG $\Leftrightarrow$ unvorhersehbar)

*Seien $G : \mathbb{B}^* \to \mathbb{B}^*$ und $\ell : \mathbb{N} \to \mathbb{N}$ in Polynomialzeit berechenbar und $G$ habe die Dehnung $\ell$. Dann ist $(G, \ell)$ ein PZG genau dann wenn $G$ unvorhersehbar ist.*

## Proof.

"$\Rightarrow$": Angenommen $(G, \ell)$ ist PZG und $n \in \mathbb{N}$. Wenn $y = (y_1, \ldots, y_{\ell(n)}) \in_R \mathbb{B}^{\ell(n)}$ zufällig gleichverteilt gewählt wurde, kann kein Bit vorhergesagt werden. Ist $G$ vorhersehbar, dann kann $y = G(x)$ von $y \in_R \mathbb{B}^{\ell(n)}$ unterschieden werden. Damit ist $G$ kein PZG.
"$\Leftarrow$": Angenommen $(G, \ell)$ ist kein PZG. Dann existiert ein $A \in \mathcal{PPT}$ mit

$$\mathbf{Pr}\left[A(G(U_n))\right] - \mathbf{Pr}\left[A(U_{\ell(n)})\right] \geq n^{-c}$$

für eine Konstante $c$ und $\infty$-vielen $n$. Die Betragsstriche in der Definition vom Pseudozufallsgenerator lassen sich ggf. durch Übergang von $A$ zu $1 - A$ entfernen. Mit dem Lemma von Yao gibt es für solche $n$ ein $B \in \mathcal{PPT}$ welches mit Wahrscheinlichkeit $\geq 1/2 + n^{-c}/\ell(n)$ ein Bit vorhersagen kann. Da $n^{-c}/\ell(n)$ nicht vernachlässigbar ist für große $n$ folgt, dass $G$ vorhersehbar ist. □

## Theorem (Goldreich-Levin Theorem)

*Sei* $f : \mathbb{B}^* \to \mathbb{B}^*$ *eine Einwegpermutation. Dann gibt es für alle* $A \in \mathcal{PPT}$ *ein vernachlässigbares* $\epsilon$ *mit*

$$\Pr_{\substack{x \in_R \mathbb{B}^n \\ r \in_R \mathbb{B}^n}} \left[ A(f(x), r) = x^t \cdot r = \sum_{i=1}^{n} x_i r_i = x \odot r \right] \leq 1/2 + \epsilon(n)$$

*für alle* $n \in \mathbb{N}$.

**Theorem**

*Existiert eine Einwegpermutation, dann existiert ein PZG $G$ mit Dehnung $n + 1$.*

**Theorem**

*Existiert eine Einwegpermutation, dann existiert ein PZG $G$ mit Dehnung $n+1$.*

**Proof.**

$G(x,r) := f(x), r, x^t r$ ist ein Pseudozufallsgenerator mit Dehnung $2n+1$. Denn $G$ ist unvorhersehbar: die ersten $2n$ Bits von $G(U_{2n})$ sind zufällig unabhängig voneinander und das $2n+1$ Bit kann wegen des Goldreich-Levin-Theorems nicht zuverlässig vorhergesagt werden. □

**Theorem (PZGs mit polynomieller Dehnung)**

*Sei $f$ eine Einwegpermutation, $c \in \mathbb{N}$ und $x, r \in \mathbb{B}^n$, setze:*

$$G(x, r) := r, f(x)^t \cdot r, f^2(x)^t \cdot r, \ldots, f^l(x)^t \cdot r$$

*mit $l = n^c$. Dann ist $G$ ein PZG mit Dehnung $l(2n) = n + n^c$.*

## Definition (Zero-Knowledge Beweise)

Sei $L \in \mathcal{NP}$ und $M$ eine Turingmaschine, die in Polynomialzeit läuft, mit

$$x \in L \Leftrightarrow \exists u \in \{0,1\}^{p(|x|)} : M(x,h) = 1. \ (p \text{ Polynom})$$

$M$ entscheidet also $L$ mit Hilfe eines Zeugen $u$. Ein Paar $(P, V)$ von interaktiven Polynomialzeitalgorithmen heißt Zero-Knowledge Beweis für $L$, falls die folgenden Eigenschaften erfüllt sind:

# ZK-Beweise Eigenschaften

1. **Vollständigkeit**: Für jedes $x \in L$ und Zertifikat $u = u(x)$ gilt

$$\mathbf{Pr}\left[\mathrm{out}_V \langle P(x,u), V(x)\rangle\right] \geq \frac{2}{3}$$

Wobei $\langle P(x,u), V(x)\rangle$ die Interaktion zwischen $P$ und $V$ mit den gegebenen Eingaben bezeichnet und $\mathrm{out}_V\, I$ beschreibt die Ausgabe von $V$ am Ende der Interaktion $I$.

# ZK-Beweise Eigenschaften

1. **Vollständigkeit**: Für jedes $x \in L$ und Zertifikat $u = u(x)$ gilt

$$\mathbf{Pr}\left[\mathrm{out}_V \langle P(x,u), V(x)\rangle\right] \geq \frac{2}{3}$$

Wobei $\langle P(x,u), V(x)\rangle$ die Interaktion zwischen $P$ und $V$ mit den gegebenen Eingaben bezeichnet und $\mathrm{out}_V\, I$ beschreibt die Ausgabe von $V$ am Ende der Interaktion $I$.

2. **Zuverlässigkeit**: Wenn $x \notin L$, dann gilt für jede Strategie $P^*$ und Eingabe $u$, dass

$$\mathbf{Pr}\left[\mathrm{out}_V \langle P^*(x,u), V(x)\rangle\right] \leq \frac{1}{3}$$

dabei ist $P^*$ in keiner Weise beschränkt.

# ZK-Beweise Eigenschaften

1. **Vollständigkeit**: Für jedes $x \in L$ und Zertifikat $u = u(x)$ gilt

$$\mathbf{Pr}\left[\text{out}_V \langle P(x,u), V(x)\rangle\right] \geq \frac{2}{3}$$

Wobei $\langle P(x,u), V(x)\rangle$ die Interaktion zwischen $P$ und $V$ mit den gegebenen Eingaben bezeichnet und $\text{out}_V\, I$ beschreibt die Ausgabe von $V$ am Ende der Interaktion $I$.

2. **Zuverlässigkeit**: Wenn $x \notin L$, dann gilt für jede Strategie $P^*$ und Eingabe $u$, dass

$$\mathbf{Pr}\left[\text{out}_V \langle P^*(x,u), V(x)\rangle\right] \leq \frac{1}{3}$$

dabei ist $P^*$ in keiner Weise beschränkt.

3. **Perfect-Zero-Knowledge-Eigenschaft**: Für alle Verifizierstrategien $V^* \in \mathcal{PPT}$ existiert ein $S^*$ mit erwarteter probabilistischer Polynomiallaufzeit, so dass für alle $x \in L$ und $u$ Zeuge dafür gilt:

$$\text{out}_{V^*} \langle P(x,u), V^*(x)\rangle \equiv S^*(x)$$

Die Gleichheit bezieht sich auf die Gleichheit der Verteilungen. $S^*$ **simuliert** $V^*$.

# ZK-Beweis für Graphenisomorphie

*Eingabe*: Graphen $G_0, G_1$ mit $V(G_i) = [n]$ in Adjazenzmatrixform gegeben.
*Eingabe von P*: $\pi : [n] \to [n]$ mit $G_1 = \pi(G_0)$.

# ZK-Beweis für Graphenisomorphie

*Eingabe*: Graphen $G_0, G_1$ mit $V(G_i) = [n]$ in Adjazenzmatrixform gegeben.
*Eingabe von P*: $\pi : [n] \to [n]$ mit $G_1 = \pi(G_0)$.

1. $P$ wählt Permutation $\pi_1 \in_R S_n$ und sendet $V$ die Adjazenzmatrix von $\pi_1(G_1) =: H$.

# ZK-Beweis für Graphenisomorphie

*Eingabe*: Graphen $G_0, G_1$ mit $V(G_i) = [n]$ in Adjazenzmatrixform gegeben.
*Eingabe von P*: $\pi : [n] \to [n]$ mit $G_1 = \pi(G_0)$.

1. $P$ wählt Permutation $\pi_1 \in_R S_n$ und sendet $V$ die Adjazenzmatrix von $\pi_1(G_1) =: H$.
2. $V$ wählt ein $b \in_R \{0, 1\}$ zufällig und schickt es zu $P$.

# ZK-Beweis für Graphenisomorphie

*Eingabe*: Graphen $G_0, G_1$ mit $V(G_i) = [n]$ in Adjazenzmatrixform gegeben.
*Eingabe von P*: $\pi : [n] \to [n]$ mit $G_1 = \pi(G_0)$.

1. $P$ wählt Permutation $\pi_1 \in_R S_n$ und sendet $V$ die Adjazenzmatrix von $\pi_1(G_1) =: H$.
2. $V$ wählt ein $b \in_R \{0, 1\}$ zufällig und schickt es zu $P$.
3. $P$ antwortet mit $\pi_1$ falls $b = 1$ und sonst mit $\pi_1 \circ \pi$. Bezeichne Antwort als $\widetilde{\pi}$.

# ZK-Beweis für Graphenisomorphie

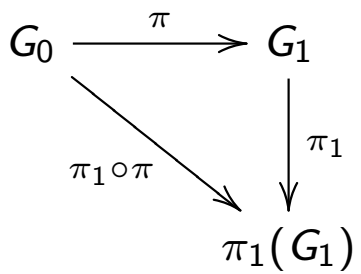
*Eingabe*: Graphen $G_0, G_1$ mit $V(G_i) = [n]$ in Adjazenzmatrixform gegeben.
*Eingabe von P*: $\pi : [n] \to [n]$ mit $G_1 = \pi(G_0)$.

1. $P$ wählt Permutation $\pi_1 \in_R S_n$ und sendet $V$ die Adjazenzmatrix von $\pi_1(G_1) =: H$.
2. $V$ wählt ein $b \in_R \{0, 1\}$ zufällig und schickt es zu $P$.
3. $P$ antwortet mit $\pi_1$ falls $b = 1$ und sonst mit $\pi_1 \circ \pi$. Bezeichne Antwort als $\widetilde{\pi}$.
4. $V$ akzeptiert gdw. $\pi_1(G_1) = \widetilde{\pi}(G_b)$.

# ZK-Beweis für Graphenisomorphie

*Eingabe*: Graphen $G_0, G_1$ mit $V(G_i) = [n]$ in Adjazenzmatrixform gegeben.
*Eingabe von P*: $\pi : [n] \to [n]$ mit $G_1 = \pi(G_0)$.

1. $P$ wählt Permutation $\pi_1 \in_R S_n$ und sendet $V$ die Adjazenzmatrix von $\pi_1(G_1) =: H$.
2. $V$ wählt ein $b \in_R \{0, 1\}$ zufällig und schickt es zu $P$.
3. $P$ antwortet mit $\pi_1$ falls $b = 1$ und sonst mit $\pi_1 \circ \pi$. Bezeichne Antwort als $\widetilde{\pi}$.
4. $V$ akzeptiert gdw. $\pi_1(G_1) = \widetilde{\pi}(G_b)$.



# Zusammenfassung

1. One-Time-Pad theoretisch die beste Lösung, praktisch aber zu ineffizient.

# Zusammenfassung

1. One-Time-Pad theoretisch die beste Lösung, praktisch aber zu ineffizient.
2. Die Komplexitätstheorie ermöglicht die Bedürfnisse der Kryptographie sauber zu formulieren.

# Zusammenfassung

1. One-Time-Pad theoretisch die beste Lösung, praktisch aber zu ineffizient.
2. Die Komplexitätstheorie ermöglicht die Bedürfnisse der Kryptographie sauber zu formulieren.
3. Einwegfunktionen sind leicht zu berechnen, schwer zu invertieren. Existieren sie, dann gilt $\mathcal{P} \neq \mathcal{NP}$. Aber aus $\mathcal{P} \neq \mathcal{NP}$ folgt nicht notwendigerweise die Existenz von Einwegfunktionen.

# Zusammenfassung

1. One-Time-Pad theoretisch die beste Lösung, praktisch aber zu ineffizient.
2. Die Komplexitätstheorie ermöglicht die Bedürfnisse der Kryptographie sauber zu formulieren.
3. Einwegfunktionen sind leicht zu berechnen, schwer zu invertieren. Existieren sie, dann gilt $\mathcal{P} \neq \mathcal{NP}$. Aber aus $\mathcal{P} \neq \mathcal{NP}$ folgt nicht notwendigerweise die Existenz von Einwegfunktionen.
4. Einwegfunktionen existieren genau dann, wenn es Pseudozufällige Generatoren gibt.

# Zusammenfassung

1. One-Time-Pad theoretisch die beste Lösung, praktisch aber zu ineffizient.
2. Die Komplexitätstheorie ermöglicht die Bedürfnisse der Kryptographie sauber zu formulieren.
3. Einwegfunktionen sind leicht zu berechnen, schwer zu invertieren. Existieren sie, dann gilt $\mathcal{P} \neq \mathcal{NP}$. Aber aus $\mathcal{P} \neq \mathcal{NP}$ folgt nicht notwendigerweise die Existenz von Einwegfunktionen.
4. Einwegfunktionen existieren genau dann, wenn es Pseudozufällige Generatoren gibt.
5. Die moderne Kryptographie hat auch ihre Problemfelder erweitert: Zero-Knowledge Proofs, digitale Signaturen, sichere Auktion-/Wahlsysteme, Datenschutz, ...

# Zusammenfassung

1. One-Time-Pad theoretisch die beste Lösung, praktisch aber zu ineffizient.
2. Die Komplexitätstheorie ermöglicht die Bedürfnisse der Kryptographie sauber zu formulieren.
3. Einwegfunktionen sind leicht zu berechnen, schwer zu invertieren. Existieren sie, dann gilt $\mathcal{P} \neq \mathcal{NP}$. Aber aus $\mathcal{P} \neq \mathcal{NP}$ folgt nicht notwendigerweise die Existenz von Einwegfunktionen.
4. Einwegfunktionen existieren genau dann, wenn es Pseudozufällige Generatoren gibt.
5. Die moderne Kryptographie hat auch ihre Problemfelder erweitert: Zero-Knowledge Proofs, digitale Signaturen, sichere Auktion-/Wahlsysteme, Datenschutz, ...
6. Es gibt noch weitere Verbindungen zur Kryptographie: Quantenrechner können effizient faktorisieren, pseudozufällige Funktionen und Verbindungen zum maschinellen Lernen, Derandomisierung von $\mathcal{BPP}$ bei Existenz von pseudozufälligen Funktionen.

# Zusammenfassung

1. One-Time-Pad theoretisch die beste Lösung, praktisch aber zu ineffizient.
2. Die Komplexitätstheorie ermöglicht die Bedürfnisse der Kryptographie sauber zu formulieren.
3. Einwegfunktionen sind leicht zu berechnen, schwer zu invertieren. Existieren sie, dann gilt $\mathcal{P} \neq \mathcal{NP}$. Aber aus $\mathcal{P} \neq \mathcal{NP}$ folgt nicht notwendigerweise die Existenz von Einwegfunktionen.
4. Einwegfunktionen existieren genau dann, wenn es Pseudozufällige Generatoren gibt.
5. Die moderne Kryptographie hat auch ihre Problemfelder erweitert: Zero-Knowledge Proofs, digitale Signaturen, sichere Auktion-/Wahlsysteme, Datenschutz, ...
6. Es gibt noch weitere Verbindungen zur Kryptographie: Quantenrechner können effizient faktorisieren, pseudozufällige Funktionen und Verbindungen zum maschinellen Lernen, Derandomisierung von $\mathcal{BPP}$ bei Existenz von pseudozufälligen Funktionen.
7. AES, gruppentheoretische Chiffren, Stromchiffren Kandidaten für PZG.

# Vortragsende

Vielen Dank für die Aufmerksamkeit!

**Fragen?**