

# Kryptographie I

## Symmetrische Kryptographie

Alexander May

Fakultät für Mathematik  
Ruhr-Universität Bochum

Wintersemester 2009/10

# Organisatorisches

- Vorlesung: **Mo 12-14** in HZO 80 (2+2 SWS, 6 CP)
- Übung: **Mo 16-18** und **Mi 10-12** in NA 5/99
- Assistent: **Thomas Dullien**, Korrektor: **Florian Giesen**
- Übungsbetrieb: jeweils abwechselnd alle 2 Wochen
  - ▶ Präsenzübung, Start 19. Oktober
  - ▶ Zentralübung, Start 26. Oktober
- Übungsaufgaben werden korrigiert.
- Gruppenabgaben bis 3 Personen
- Bonussystem:
  - 1/3-Notenstufe für 50%, 2/3-Notenstufe für 75%
- Klausur: Ende Februar

# Literatur

Vorlesung richtet sich nach

- Jonathan Katz, Yehuda Lindell, “Introduction to Modern Cryptography”, Taylor & Francis, 2008

Weitere Literatur

- S. Goldwasser, M. Bellare, “Lecture Notes on Cryptography”, MIT, online, 1996–2008
- O. Goldreich, “Foundations of Cryptography – Volume 1 (Basic Tools)”, Cambridge University Press, 2001
- O. Goldreich, “Foundations of Cryptography – Volume 2 (Basic Applications)”, Cambridge University Press, 2004s
- A.J. Menezes, P.C. van Oorschot und S.A. Vanstone, “Handbook of Applied Cryptography”, CRC Press, 1996

# Effiziente Algorithmen

## Ziel:

- Ver-/Entschlüsseln soll effizient möglich sein.
- Unser Berechnungsmodell ist die Turingmaschine (s. DiMa I+II)
- Verwenden polynomielle Algorithmen  $A \in \mathcal{P}$ .

## Definition Polynomialzeit-Algorithmus

Sei  $A$  ein Algorithmus.  $A$  heißt *polynomial-Zeit* ( $pt$ ), falls  $A$  bei Eingaben der Länge  $n$  in Laufzeit  $\mathcal{O}(n^k)$  für ein festes  $k$  anhält.  $A$  heißt *probabilistisch polynomial-Zeit* ( $ppt$ ), falls  $A$  ein  $pt$ -Algorithmus ist, der Zufallsbits verwendet.

# Verschlüsselungsverfahren

## Definition Symmetrisches Verschlüsselungsverfahren

Sei  $n$  ein Sicherheitsparameter und  $\mathcal{K}, \mathcal{M}, \mathcal{C}$  der Schlüssel-, Nachrichten- bzw. Chiffretextraum.

Ein *symmetrisches Verschlüsselungsverfahren*  $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$  besteht drei ppt-Algorithmen:

- 1 **Gen:** *Gen* liefert bei Eingabe  $1^n$  einen Schlüssel  $k \in_R \mathcal{K}$ .
- 2 **Enc:** *Enc* liefert bei Eingabe  $k$  und Nachricht  $m \in \mathcal{M}$  einen Chiffretext  $c \in \mathcal{C}$ . D.h. *Enc* berechnet eine Funktion  $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$ . Wir schreiben  $c \leftarrow \text{Enc}_k(m)$ .
- 3 **Dec:** *Dec* liefert bei Eingabe  $k$  und  $c = \text{Enc}_k(m) \in \mathcal{C}$  eine Nachricht mit der Eigenschaft

$$\text{Dec}_k(\text{Enc}_k(m)) = m \text{ f\"ur alle } k \in \mathcal{K}, m \in \mathcal{M}.$$

Wir schreiben  $m \leftarrow \text{Dec}_k(c)$ .

- $\text{Enc}_k(m)$  ist für jedes feste  $k$  injektiv.

# Kerckhoffs' Prinzip (1883)

## Forderung Kerckhoffs' Prinzip

Die Sicherheit eines Verschlüsselungsverfahrens  $\Pi = (Gen, Enc, Dec)$  darf ausschließlich auf der Geheimhaltung des Schlüssels beruhen. D.h. *Gen*, *Enc* und *Dec* sind bekannt.

## Anmerkungen:

- Schlüssel lassen sich besser geheimhalten als Algorithmen.
- Schlüssel lassen sich besser austauschen als Algorithmen.
- Schlüssel lassen sich besser verwalten als Algorithmen.
- Öffentliche Untersuchung von  $\Pi$  durch Experten ist erforderlich.

# Typen von Angreifern

## Definition Angreiferszenarien

Wir unterscheiden folgende vier Angriffe auf Verschlüsselungsverfahren in aufsteigender Stärke.

- 1 **Ciphertext Only Angriff (COA, passiver Angriff):**  
Angreifer erhält nur Chiffretexte.
- 2 **Known Plaintext Angriff (KPA, passiv):**  
Angreifer erhält Paare Klartext/Chiffretext.
- 3 **Chosen Plaintext Angriff (CPA, aktiv):**  
Angreifer erhält Chiffretexte von Klartexten seiner Wahl.
- 4 **Chosen Ciphertext Angriff (CCA, aktiv):**  
Angreifer erhält Entschlüsselung von Chiffretexten seiner Wahl.

## Ziele eines Angreifers:

- Ziel bei 1: Bestimme zugrundeliegende Klartexte.
- Ziel bei 2-4: Bestimme Klartexte anderer Chiffretexte.
- Wir werden bald schon schwächere Ziele für Angreifer definieren.

# Monoalphabetische Substitution – Verschiebechiffre

**Idee** der Verschiebe-Chiffre: Verschiebe jeden Buchstaben um  $k$  Position zyklisch im Alphabet. Identifizieren  $A, \dots, Z$  mit  $0, \dots, 25$ .

## Definition Verschiebe-Chiffre (ca. 50 v. Chr.)

Es gilt  $\mathcal{M} = \mathcal{C} = \mathbb{Z}_{26}^n$  und  $\mathcal{K} = [25] := \{1, \dots, 25\}$ .

① **Gen:** Ausgabe  $k \in_R [25]$ .

② **Enc:** Verschlüssele  $m = m_0 \dots m_{n-1} \in \mathbb{Z}_{26}^n$  als  
 $c := Enc_k(m_0) \dots Enc_k(m_{n-1})$  mit

$$Enc_k(m_i) \leftarrow m_i + k \bmod 26 \text{ für } i = 0, \dots, n - 1.$$

③ **Dec:** Entschlüssele  $c := c_0 \dots c_{n-1}$  als  
 $m_0 \dots m_{n-1} := Dec_k(c_0) \dots Dec_k(c_{n-1})$  mit

$$Dec_k(c_i) \leftarrow c_i - k \bmod 26 \text{ für } i = 0, \dots, n - 1.$$

- Beispiel: KRYPTO wird mit  $k = 2$  als MTARVQ verschlüsselt.
- $|\mathcal{K}| = 25$ , d.h. der Schlüsselraum kann leicht durchsucht werden.
- Benötigen Schlüsselräume mit mindestens  $2^{80}$  Elementen.

# Polyalphabetische Substitution – Vigenère Chiffre

**Idee** der Vigenère Chiffre:

- Verwende  $t$  hintereinandergeschaltete Verschiebungen.

## Definition Vigenère Chiffre (1553)

Es gilt  $\mathcal{M} = \mathcal{C} = \mathbb{Z}_{26}^n$  und  $\mathcal{K} = \mathbb{Z}_{26}^t$ .

1 **Gen:** Berechne  $k = k_0 \dots k_{t-1} \in_R \mathbb{Z}_{26}^t$ .

2 **Enc:** Verschlüssele  $m = m_0 \dots m_{n-1} \in \mathbb{Z}_{26}^n$  als  
 $c := Enc_k(m_0) \dots Enc_k(m_{n-1})$  mit

$$Enc_k(m_i) \leftarrow m_i + k_{i \bmod t} \bmod 26 \text{ für } i = 0, \dots, n-1.$$

3 **Dec:** Entschlüssele  $c := c_0 \dots c_{n-1}$  als  
 $m_0 \dots m_{n-1} := Dec_k(c_0) \dots Dec_k(c_{n-1})$  mit

$$Dec_k(c_i) \leftarrow c_i - k_{i \bmod t} \bmod 26 \text{ für } i = 0, \dots, n-1.$$

- Sonderfall  $t = 1$  liefert Verschiebechiffre.
- Sonderfall  $t = n$  liefert perfekt sichere (!) Vernam-Chiffre (1918).
- Kryptanalyse mittels Häufigkeitsanalyse für  $t \ll n$  möglich.