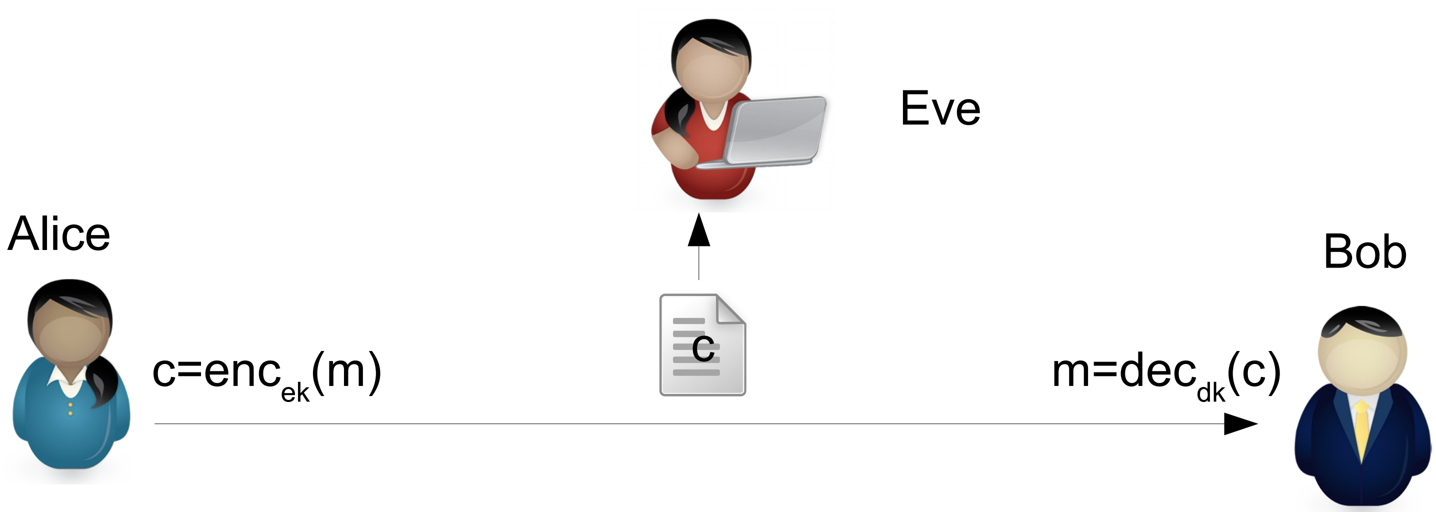
Themen zur Computersicherheit

Verschlüsselung

PD Dr. Reinhard Bündgen
buendgen@de.ibm.com

Kryptografisches System

- Alphabete $\Sigma_{1,}$ Σ_2
- Verschlüsselungsverfahren / Chiffre / englisch: cipher
  - enc: eK x $\Sigma_1^* \rightarrow \Sigma_2^*$, dec: dK x $\Sigma_2^* \rightarrow \Sigma_1^*$
  - Schlüssel (ek,dk) $\in$ eK x dK mit $dec_{dk}(enc_{ek}(m)) = m$ für alle $m \in \Sigma_1^*$
  - verschlüsselt Klartext (plain text) zu Geheimtext (cipher text)



Angriffe auf ein kryptografisches System

- Ziele eines Angriffs
  - finde Klartext zu gegebenem Geheimtext
  - finde Schlüssel
- oft erreicht ein Angriff nur ein Teilziel
  - Eigenschaften des Geheimtextes
  - Eigenschaften des Schlüssels
- das allerdings von einem weiteren Angriff genutzt werden kann
- I.A braucht man zusätzliche Informationen über den Klartext oder Geheimtext um entscheiden zu können, ob ein Angriff erfolgreich war
- Seitenkanalangriffe
  - nutzen ein Informationsleck des Verschlüsselungsverfahrens
  - z.B Dauer der Berechnung, Strahlung, ...

Angriffsformen

- Art des Zusatzinformationsgewinns
  - Angriff mit bekanntem Geheimtext (cipher text only attack, COA)
    - Angreifer kennt einen oder mehrere Geheimtexte
  - Angriff mit bekanntem Klartext (known plaintext attack, KPA)
    - Angreifer kennt die zu einem oder mehreren Geheimtexten gehörigen Klartexte
  - Angriff mit gewähltem Klartext (chosen plain text attack, CPA)
    - Angreifer kann sich zu gewählten Klartexten die dazugehörigen Geheimtexte berechnen lassen
  - Angriff mit gewählten Geheimtext (chosen cipher text attack, CCA)
    - Angreifer kann sich zu gewählten Geheimtexten die dazugehörigen Klartexte berechnen lassen
- Zeitpunkt des Zusatzinformationsgewinns
  - direkter Angriff (offline, batch)
    - vor bekannt werden der Herausforderung
  - adaptiv (online, adaptive)
    - nach bekannt werden der Herausforderung

Orakel

- Angriffs-Orakel: Funktion die Verschlüsselungsaufgaben als „black box“ löst
  - COA-Orakel erzeugt zufälligen Geheimtext
  - KCA-Orakel erzeugt zufälligen Klartext mit zugehörigem Geheimtext
  - CPA-Orakel berechnet zu gegebenem Klartext den Geheimtext
  - CCA-Orakel berechnet zu gegebenem Geheimtext den dazugehörigen Klartext
- IND-Orakel verschlüsselt zufällig einen von zwei gegebenen Klartexten und gibt den Geheimtext aus
- Orakel berechnet *nicht* die Lösung der Herausforderung

Ununterscheidbarkeit (indistinguishability)

Ununterscheidbarkeitstest:

- Angreifer befragt das Angriffs-Orakel
- Angreifer wählt 2 Nachrichten m_1, m_2
- IND-Orakel wählt ein $m \in \{ m_1, m_2 \}$
- IND-Orakel gibt $c = enc_k(m)$ aus
- bei adaptivem Angriff befragt Angreifer das Angriffs-Orakel
- Angreifer gibt als Resultat seiner Angriffs: $b \in \{ 1, 2 \}$ aus

Der Test ist (im Sinne des Angreifers) bestanden, wenn $c = enc_k(m_b)$

Definition: Ein kryptografisches System ist gegen einen Angriff *sicher im Sinne der polynomiellen Ununterscheidbarkeit*, wenn der Angriff den Ununterscheidbarkeitstest nur mit einer Wahrscheinlichkeit von 0,5 besteht.

Definition: Ein kryptografisches System ist *perfekt sicher*, wenn es gegen beliebige Angriffe sicher im Sinne der polynomiellen Ununterscheidbarkeit ist.

Alternative Definition: Ein kryptografisches System ist *perfekt sicher*, wenn nach jedem Angriff, die a priori Wahrscheinlichkeit eines beliebigen Klartextes gleich der a posteriori Wahrscheinlichkeit des Klartextes ist.

One Time Pad

- **Definition** Das folgende Kryptosystem heißt *one time pad*:
  - Sei $\sum = \sum_1 = \sum_2 = \{0,1\}$, eK = dK = $\sum^*$,
  - $m \in \sum^*$
  - $k \in \sum^*$ ist ein Zufallsbitstring mit $|k| = |m|$
  - und $enc_k(m) = dec_k(m) = k \oplus m$ ($\oplus$ bit-weises xor)
- **Satz** Das one time pad ist perfekt sicher, wenn jeder Schlüssel genau einmal zur Verschlüsselung einer Nachricht verwendet wird.
- **Achtung**: Ein Zufallsbitstring darf nur einmal als one time pad Schlüssel verwendet werden!
  - seien $m_1, m_2, k \in \sum^*$, $|m_1| = |m_2| = |k|$, $c_1 = m_1 \oplus k$, $c_2 = m_2 \oplus k$
  - dann ist $c_1 \oplus c_2 = m_1 \oplus m_2$
  - wenn m_2 bekannt: $m_1 = c_1 \oplus c_2 \oplus m_2$

Historische Chiffren I

- Caesar Chiffre

a	b	c	d	...	v	w	x	y	z
↓	↓	↓	↓		↓	↓	↓	↓	↓
d	e	f	g	...	y	z	a	b	c

- Verschiebechiffren („Caesar Chiffren“)
  - $\sum = \sum_1 = \sum_2 = \{ s_0, s_1, \ldots, s_{n-1} \}$, $ek = dk \in \{0, \ldots, n-1\}$
  - enc_k: $s_i \,||\, m \mapsto s_{(i+k) \bmod n} \,||\, enc_k(m)$ für $s_i \in \sum$, $m \in \sum^*$
  - dec_k: $s_i \,||\, m \mapsto s_{(i-k) \bmod n} \,||\, dec_k(m)$ für $s_i \in \sum$, $m \in \sum^*$
  - Schlüsselraum: n verschiedene Schlüssel
  - Angriff: vollständige Suche
- Monoalphabetische Chiffren
  - $\sum = \sum_1 = \sum_2 = \{ s_0, s_1, \ldots, s_{n-1} \}$
  - $ek=dk \in \{ \pi \in \sum \rightarrow \sum \mid \pi$ ist Permutation $\}$
  - enc_π: $s||m \mapsto \pi(s)||enc_\pi(m)$ für $s \in \sum$, $m \in \sum^*$
  - dec_π: $s||m \mapsto \pi^{-1}(s)||dec_\pi(m)$ für $s \in \sum$, $m \in \sum^*$
  - Schlüsselraum: $|\sum|!$ verschiedene Schlüssel
  - Angriff: statistische Analyse

Historische Chiffren II

Polyalphabetische Chiffren

- Vigenère Chiffre (um 1500)
  - Sei $|\sum| = n$, ord: $\sum \rightarrow \{0,1,...,n-1\}$ bijektiv
  - $eK = dK = \sum^*$ mit $ek = dk = k \in \sum^*$ für alle Schlüssel
  - seien $a, s \in \sum$ und $k, m \in \sum^*$ dann ist
  - $enc_{a||k}(s||m) = enc_{ord(a)}(s) \,||\, enc_{k||a}(m) =$ $ord^{-1}((ord(s) + ord(a)) \bmod n) \,||\, enc_{k||a}(m)$

	A	B	C	D	...	X	Y	Z
A	A	B	C	D	...	X	Y	Z
B	B	C	D	E	...	Y	Z	A
C	C	D	E	F	...	Z	A	B
D	D	E	F	G	...	A	B	C
...								
X	X	Y	Z	A	...	U	V	W
Y	Y	Z	A	B	...	V	W	X
Z	Z	A	B	C	...	W	X	Y

- Angriffe
  - Vigenère Chiffre galt lange Zeit als unentzifferbar
  - Bestimmung der Schlüssellänge (Kasiski)
    - gleiche Geheimtextstücke (die zu gleichen Klartextstücken gehören) haben einen Abstand, der ein vielfaches der Schlüssellänge beträgt
  - Bestimmung von sprachlicher Redundanz im Geheimtext (Friedman)
    - Friedmanscher Koinzidenzindex: Wahrscheinlichkeit dass zwei beliebig aus einem Text gewählte Buchstaben gleich sind (Ziehen ohne zurücklegen).
    - Für jede Sprache gibt es charakteristische Wahrscheinlichkeit, dass zwei aus einem Text gewählte Buchstaben gleich sind (folgt aus statistischer Buchstabenhäufigkeit)
    - Sei $|k|$ die gesuchte Schlüssellänge, f der sprachtypische Koinzidenzindex, g der Koinzidenzindex eines Zufallstextes (Gleichverteilung der Buchstaben), n die Länge des Textes, p die Anzahl der gezogenen Paare aus gleichen Buchstaben
    - theoretisch: $p = f \cdot n \cdot (n/|k| - 1) / 2 + g \cdot n \cdot (n - n/|k|) / 2$
    - $F = p \cdot 2 / (n \cdot (n-1))$ ist Approximation des Friedmanschen Koinzidenzindexes
    - $\Rightarrow |k| = (f - g) \cdot n / ((n-1) \cdot F - g \cdot n + f)$

Rechnung

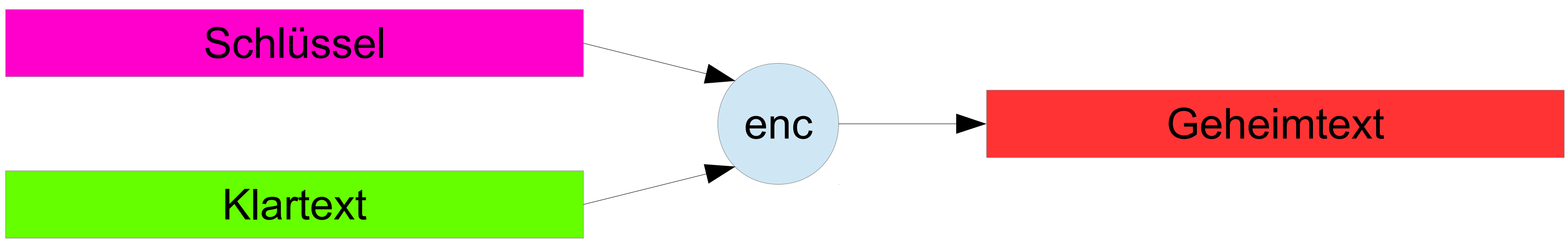
- gesuchte Schlüssellänge: $|k|$
- sprachtypischer Koinzidenzindex: f
- Koinzidenzindex eines Zufalltextes: g
- Paare in Geheimtext P
- Buchstaben pro Spalte: $n / |k|$
- Buchstabenpaare pro Spalte: $n \cdot (n/|k| - 1) / 2$
- Buchstabenpaare aus verschiedenen Spalten: $n \cdot (n - n/|k|) / 2$
- Erwartete Anzahl von Paaren aus gleichen Buchstaben:
  - $p = f \cdot n \cdot (n/|k| - 1) / 2 + g \cdot n \cdot (n - n/|k|) / 2$
- Approximation des Friedmanschen Koinzidenzindexes:
  - $F = P \cdot 2 / (n \cdot (n-1))$
- P in F durch p ersetzen und umformen:
  - $\Rightarrow |k| = (f - g) \cdot n / ((n-1) \cdot F - g \cdot n + f)$

Historische Chiffren III

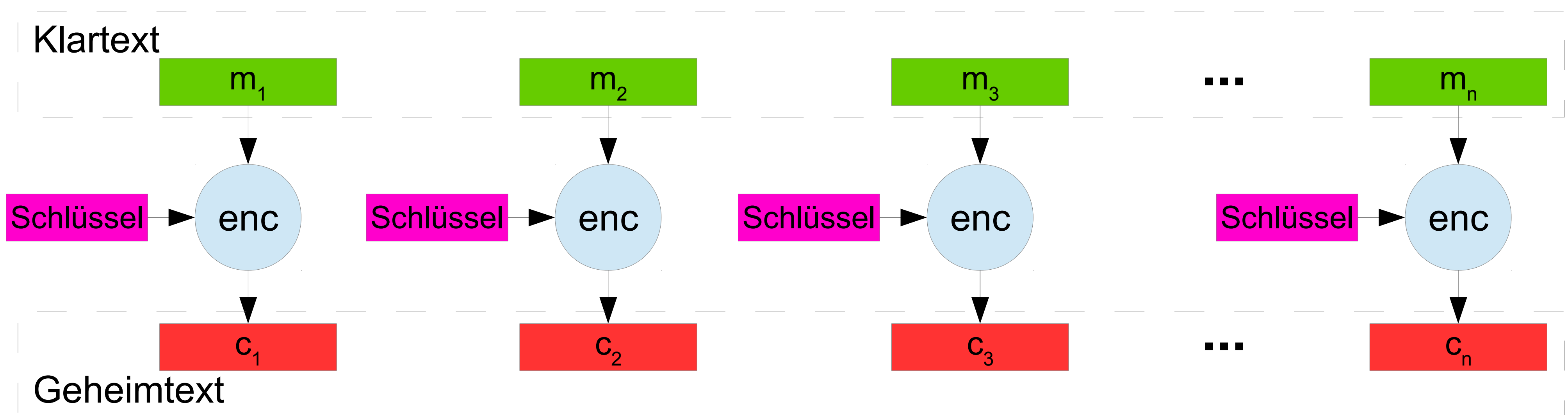
- die Enigma → getrennte Präsentation

Moderne Chiffren

- Stromchiffren
  - Verknüpfung eines Klartextes mit Schlüsselstrom
  - Verknüpfung: oft xor
  - Schlüssel: generiert mit Pseudozufallsgenerator (mit vereinbartem initialen Zustand)
  - Klartext kann beliebige Länge haben

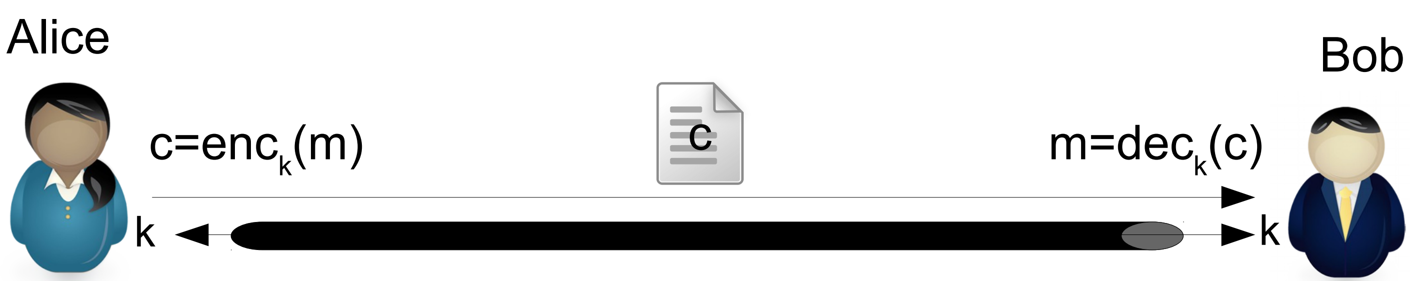


- Blockchiffren
  - Verschlüsselung von Nachrichtenblöcken mit Hilfe eines konstanten Schlüssels
  - monoalphabetische Verschlüsselung auf großem Alphabet
  - Klartextlänge muss Vielfaches der Blocklänge sein



Symmetrische und asymmetrische Chiffren

- **Symmetrische Chiffren**
  - eK=dK (= K) und für alle Schlüssel (ek,dk) $\in$ eK x dK gilt ek = dk (=k)



- **Asymmetrische Chiffren**
  - für alle Schlüssel (ek,dk) $\in$ eK x dK gilt
    - ek $\neq$ dk und
    - dk kann nicht effizient aus ek berechnet werden
  - ek heißt auch öffentlicher Schlüssel (public key pk)
  - dk heißt auch privater Schlüssel (private/secret key sk)