# Grundlagen der Informatik

Prof. Dr. J. Schmidt
Fakultät für Informatik

## GDI – WS 2018/19
## Kryptographie
## Grundbegriffe, klassische Verfahren

Hochschule Rosenheim
University of Applied Sciences

# Überblick



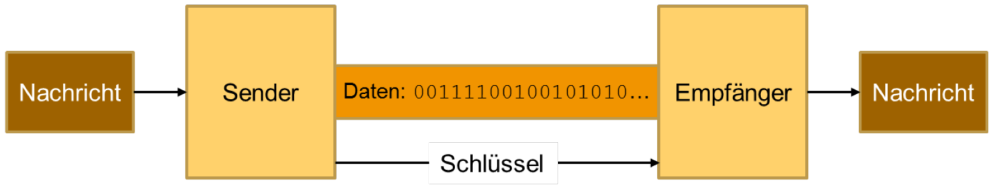
- Was ist ein Kryptosystem?
- Welche Klassen von Verschlüsselungsverfahren lassen sich prinzipiell unterscheiden?
- klassische Verfahren

# Motivation



- Verschlüsselte Übermittlung von Nachrichten ist von großem Interesse
  - nicht nur für Militärs und Geheimagenten
  - sondern auch für Unternehmen
    (z.B. Übermittlung vertraulicher Informationen zu neuen Produkten)
  - und Privatpersonen
    (z.B. Online-Banking → https)

- Ablauf




- Verschlüsselung der Botschaft (Klartext genannt) in einen Chiffretext
    - Verwendung von Verschlüsselungsalgorithmus
    - und Schlüsselparameter
- Sender sendet eine Nachricht mit dem Chiffretext an den Empfänger
- Entschlüsselung des Chiffretexts durch Empfänger
    - Verwendung eines passenden Entschlüsselungsalgorithmus
    - und der gleichen Schlüsselparameter
- Empfänger erhält den Klartext der Botschaft

# Verschlüsselung (3)

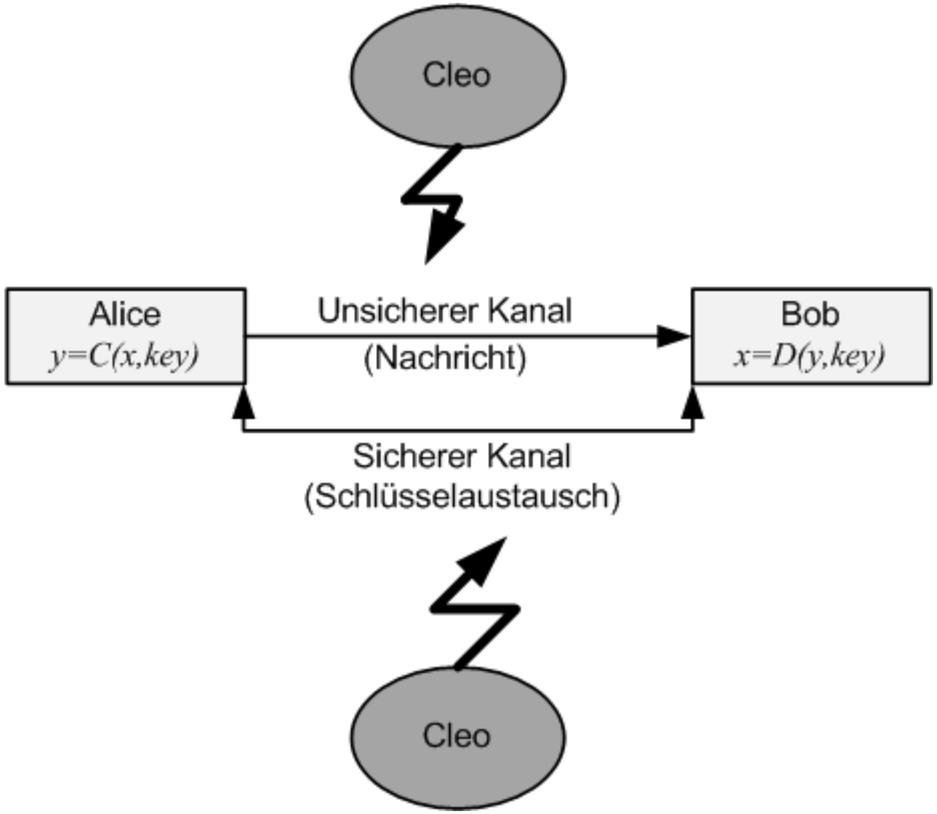


- Prinzipielle Unterscheidung
  - Symmetrische Verschlüsselungsverfahren
    - Identischer, geheimer Schlüssel
    - Austausch über sicheren Kanal
  - Asymmetrische Verschlüsselungsverfahren
    - Verschlüsselung mit öffentlichem Schlüssel
    - Entschlüsselung mit privatem Schlüssel des Empfängers

# Verschlüsselung (4)



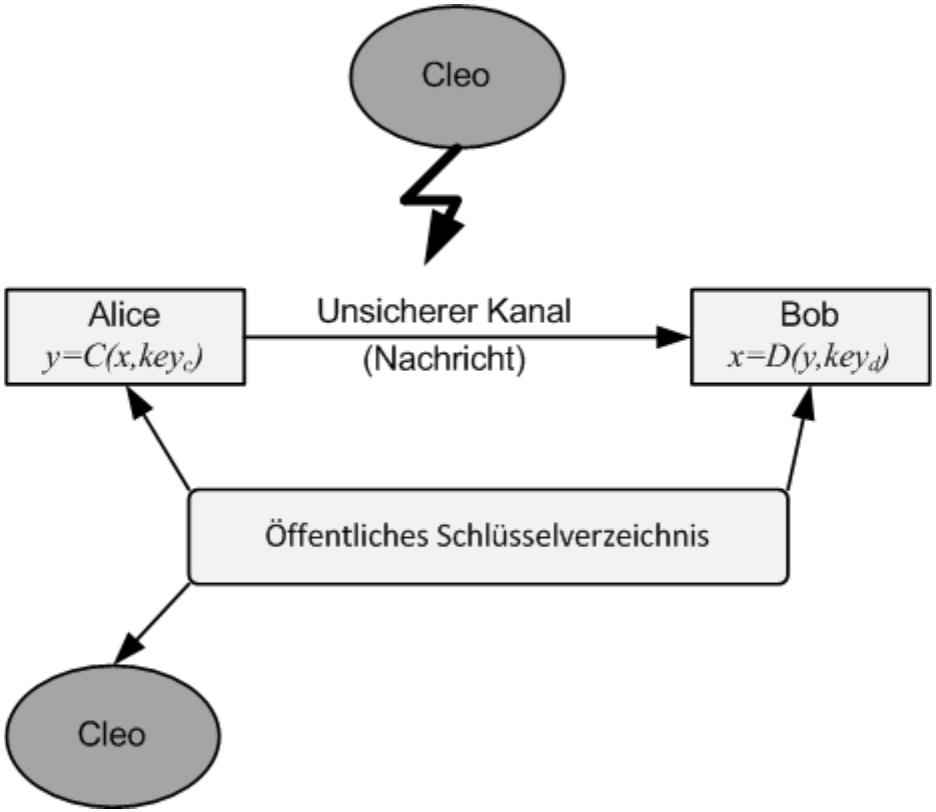
- Modell eines symmetrischen Kryptosystems



# Verschlüsselung (5)



- Modell eines asymmetrischen Kryptosystems



# Kerckhoffssches Prinzip



- Formuliert 1883
- Grundsatz aller modernen kryptographischen Verfahren

- Sicherheit eines Verfahrens
  - beruht **nicht** auf Geheimhaltung des Algorithmus
  - sondern auf Geheimhaltung des Schlüssels

- Also
  - kein „Security through Obscurity“
  - Algorithmen sind öffentlich

# Klassische Chiffren



- klassisch = vor 1950 entwickelte Verfahren
- abgesehen von One-Time-Pads heute praktisch nicht mehr im Einsatz

- hier vorgestellt zur Veranschaulichung der grundlegenden Verschlüsselungsprinzipien
  - Transpositions-Chiffren
  - One-Time-Pad

# Transpositions-Chiffren

- Permutationen des Klartexts
  - Zeichen $x_i$ eines Alphabets A mit n Zeichen werden nach der Vorschrift

    $$x_i \rightarrow x_{(k \cdot i + d) \bmod n}$$

    auf Zeichen des selben Alphabets abgebildet
  - **k** heißt multiplikativer Schlüssel
  - **d** heißt additiver Schlüssel

- Sonderfälle
  - k = 1 ➔ Cäsar-Code
  - d = 0 ➔ Produkt-Chiffren

- Durchsetzung dieser Verfahren mit der Verfügbarkeit elektromechanischer Verschlüsselungsautomaten
  - Erstes Beispiel: Enigma

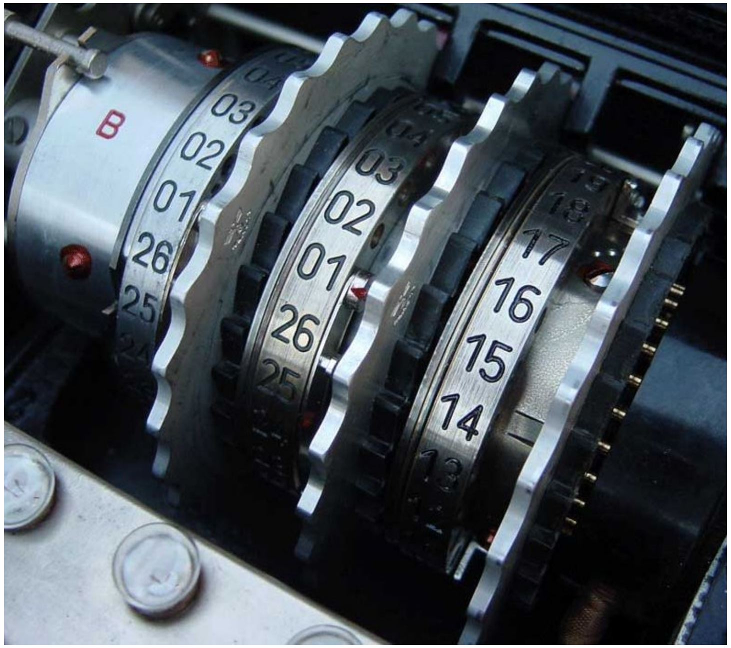
# Enigma




Zur Beachtung!
Klappe
schließen




Walzensatz
Lampenfeld
Tastatur
Steckerbrett



# Transpositions-Chiffren – Multiplikative Schlüssel (1)

- Alphabet A mit n Zeichen
- Chiffriertes Zeichen =
  - Multiplikation der Position eines Zeichens mit Schlüssel k
  - und Berechnung Modulo n
- Beliebige Kombination nicht möglich für eindeutige Abbildung
  - k = 4, n = 26, d = 0

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** | **S** | **T** | **U** | **V** | **W** | **X** | **Y** | **Z** |
| **A** | **E** | **I** | **M** | **Q** | **U** | **Y** | **C** | **G** | **K** | **O** | **S** | **W** | **A** | **E** | **I** | **M** | **Q** | **U** | **Y** | **C** | **G** | **K** | **O** | **S** | **W** |

  - Wiederholungen treten auf – untauglich!

# Transpositions-Chiffren – Multiplikative Schlüssel (2)



- Für brauchbare Kombination (k, n) muss gelten
  - k und n müssen teilerfremd sein
    - ggT(k, n) = 1
  - Nur diese Schlüssel k sind geeignet, weil sie eine modulare Inverse $k^{-1}$ haben mit
    - $k \cdot k^{-1} \bmod n = 1$
  - Zur Berechnung der modularen Inversen
    - Erweiterter euklidischer Algorithmus
    - Satz von Euler/Fermat
    - Details siehe Anhand
- Beispiel: Teilerfremd zu n = 26 sind {1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25}
  - Damit: k = 7 geeignet
  - Inverse mod 26: $7^{-1} = 15$
  - Test: $7 \cdot 15 \bmod 26 = 105 \bmod 26 = 1$

# Transpositions-Chiffren – Beispiel (1)



- Verschlüsselung des Klartexts *Liebling*
  - mit multiplikativem Schlüssel k=7

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **H** | **I** | **J** | **K** | **L** | **M** | **N** | **O** | **P** | **Q** | **R** | **S** | **T** | **U** | **V** | **W** | **X** | **Y** | **Z** |
| **A** | **H** | **O** | **V** | **C** | **J** | **Q** | **X** | **E** | **L** | **S** | **Z** | **G** | **N** | **U** | **B** | **I** | **P** | **W** | **D** | **K** | **R** | **Y** | **F** | **M** | **T** |

  - und additivem Schlüssel d=5

- Ergebnis
  - Klartext: **L I E B L I N G**
  - Multiplikation (k=7): **Z E C H Z E N Q**
  - Verschiebung (d=5): **E J H M E J S V**

# Transpositions-Chiffren – Beispiel (2)



- Entschlüsselung mit inversen Operationen

  - Verschlüsselter Text: E J H M E J S V
    Verschiebung (–d = –5): Z E C H Z E N Q
  - Multiplikation ($k^{-1}$=15): L I E B L I N G

# Verschlüsselung mit Zufallsfolgen (1)



- Ansatz
  - Schlüssel
    - Folge **zufällig** angeordneter Bits
    - Genauso lang wie der zu verschlüsselnde Text
  - XOR-Verknüpfung binärer Schlüssel mit binärem Klartext

- Vorteil
  - Verschlüsselung **und** Entschlüsselung mit XOR

- Schlüssel wird als One-Time-Pad bezeichnet

- Beispiel
  - Verschlüsselung

| | |
|---|---|
| Schlüssel: | 11010110 |
| Klartext: | 01101111 (XOR) |
| Chiffretext: | 10111001 |

  - Entschlüsselung

| | |
|---|---|
| Schlüssel: | 11010110 |
| Chiffretext: | 10111001 (XOR) |
| Klartext: | 01101111 |

# Verschlüsselung mit Zufallsfolgen (3)



- Häufig Nutzung dieses Verschlüsselungsverfahrens durch Pay-TV
  - Gesendete Signale werden verzerrt
  - Zur Entzerrung müssen zahlende Zuschauer ein Passwort haben, das regelmäßig geändert wird
  - Neues Passwort wird Kunden mit dem verzerrten Signal über Satellit mitgeteilt
    (Passwort wird auch binär kodiert = Folge von 0 und 1)
    - Erzeugung einer gleichlangen binären Zufallszahlenfolge
    - Bitweise XOR-Verknüpfung mit dem Passwort zu Chiffre
    - Übertragung Chiffre

# Verschlüsselung mit Zufallsfolgen (4)



- Häufig Nutzung dieses Verschlüsselungsverfahrens durch Pay-TV
  - Autorisierte Benutzer können mit einem zur Verfügung gestellten Dekodiergerät die empfangene Chiffre entschlüsseln
    - Mikroprozessor im Dekodiergerät enthält eine Kopie des Zufallszahlengenerators
    - D.h. wenn er die richtige Eingabe erhält, produziert er die gleiche Zufallszahlenfolge
    - Erzeugte Zufallszahlenfolge wird XOR mit Chiffre verknüpft
    - Ergebnis: dechiffriertes Passwort

# Verschlüsselung mit Zufallsfolgen (5)



- One-Time-Pads bieten prinzipiell perfekte Sicherheit
  - die erzeugten verschlüsselten Daten lassen **keine** Rückschlüsse auf den Klartext zu (außer der Länge)
  - das Verfahren kann also **nicht gebrochen** werden – egal, wie hoch die eingesetzte Rechenleistung ist
  - Beweis von Shannon 1949

- Einschränkungen in der Praxis
  - das gilt nur, wenn der Schlüssel tatsächlich aus echten Zufallszahlen erzeugt wird
  - Sender und Empfänger müssen Zugriff auf die gleiche Folge von Zufallszahlen haben → Verwendung von Pseudo-Zufallszahlengeneratoren
  - Schlüssel ist genauso lang wie die Daten
  - daher eher selten verwendet

# Anhang – modulare Inverse

# Euklidischer Algorithmus



- Es gilt: ggT(a, b) = ggT(b, a mod b)
  - Abbruch wenn b = 0
  - dann ist a der ggT

- Beispiele:
  - ggT(26, 13) = ggT(13, 0) → ggT = 13
  - ggT(26, 7) = ggT(7, 5)
    = ggT(5, 2)
    = ggT(2, 1)
    = ggT(1, 0) → ggT = 1

# Erweiterter Euklidischer Algorithmus



- Zur Bestimmung einer modularen Inversen
- Es gilt: ggT(a, b) = $s \cdot a + t \cdot b$
  - s, t ganze Zahlen
  - wenn ggT(a,b) = 1 $\Rightarrow$
    t ist das modulare Inverse von b (mod a)
- Beispiel: modulares Inverses zu 7 mod 26

$26 = 3 \cdot 7 + 5 \quad \rightarrow 5 = 26 - 3 \cdot 7$

$7 = 1 \cdot 5 + 2 \quad \rightarrow 2 = 7 - 1 \cdot 5 = 7 - (26 - 3 \cdot 7) =$
$= -26 + 4 \cdot 7$

$5 = 2 \cdot 2 + 1 \quad \rightarrow 1 = 5 - 2 \cdot 2$
$= 26 - 3 \cdot 7 - 2 \cdot (-26 + 4 \cdot 7)$
$= 3 \cdot 26 - 11 \cdot 7$

$2 = 1 \cdot 2 + 0$

Inverses existiert

und ist -11 = 15 mod 26

# Eulersche $\phi$-Funktion



- Gibt die Anzahl der natürlichen Zahlen an
  - die kleiner als n sind
  - und keinen gemeinsamen Teiler mit n haben
  - $\phi(n) = |\{1 \leq x \leq n \mid ggT(x, n) = 1\}|$
- Berechnung ($p, q$ sind Primzahlen $p \neq q$)
  - $\phi(p) = p - 1$ alle Zahlen von $1$ bis $p - 1$ sind zu $p$ teilerfremd
  - $\phi(pq) = \phi(p)\phi(q) = (p - 1)(q - 1)$
  - $\phi(p^i) = p^{i-1}(p - 1)$
  - $\phi(p^i q^j) = \phi(p^i)\phi(q^j) = p^{i-1}(p - 1)\, q^{j-1}(q - 1)$
- Beispiele
  - $\phi(5) = 4$
    - es gibt vier zu 5 teilerfremde Zahlen < 5, nämlich 1, 2, 3, 4
  - $\phi(15) = \phi(3 \cdot 5) = \phi(3)\phi(5) = 2 \cdot 4 = 8$
  - $\phi(27) = \phi(3^3) = 3^2 \cdot (3 - 1) = 9 \cdot 2 = 18$
    - die zu 27 teilerfremden Zahlen sind: 1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23, 25, 26
  - $\phi(72) = \phi(2^3 \cdot 3^2) = 2^2 \cdot (2 - 1) \cdot 3^1 \cdot (3 - 1) = 4 \cdot 3 \cdot 1 \cdot 2 = 24$

# Satz von Euler/Fermat – Modulare Inverse



- Satz von **Euler**:
  für alle $x \in \mathbb{Z}, n \in \mathbb{N}, \mathrm{ggT}(x, n) = 1$ gilt:

$$x^{\phi(n)} \bmod n = 1$$

- Spezialfall: $n$ ist eine Primzahl $p$
  $\rightarrow$ kleiner Satz von **Fermat**:

$$x^{p-1} \bmod p = 1$$

- Es gilt:

$$x \cdot x^{\phi(n)-1} \bmod n = 1$$

  und damit:

$$x^{-1} = x^{\phi(n)-1} \bmod n$$

  bzw. mit Primzahl:

$$x^{-1} = x^{p-2} \bmod p$$

# Modulare Inverse/Euler – Beispiel

- Mit Primzahl als Modul: $p = 31$
  - gesucht: modulare Inverse zu $x = 2$
  - es gilt: $2^{-1} = 2^{31-2} \bmod 31 = 2^{29} \bmod 31 = 16$
  - Test: $2 \cdot 16 = 32 \bmod 31 = 1$

- Mit $n = 26$ aus vorherigem Beispiel
  - gesucht: modulare Inverse zu $x = 7$
  - bestimme $\phi(26)$,
    d.h. die **Anzahl** der zu 26 teilerfremden Zahlen.
    - Primfaktorisierung:
      $26 = 13 \cdot 2$
    - damit: $\phi(26) = \phi(13)\phi(2) = 12 \cdot 1 = 12$
  - es gilt: $7^{-1} = 7^{12-1} \bmod 26 = 7^{11} \bmod 26 = 15$
  - Test: $7 \cdot 15 = 105 \bmod 26 = 1$