



Leibniz-Rechenzentrum  
der Bayerischen Akademie der Wissenschaften

The background of the slide is a photograph of a modern, multi-story building with a glass and metal facade, partially obscured by a blue overlay. The building has several windows and a prominent vertical structure. In the foreground, there are trees and a street with a few people walking.

# Kapitel 6: Kryptographische Grundlagen

1. Kryptologie: Begriffe, Klassifikation

2. Steganographie

3. Kryptographie, Begriffe und Definitionen

- Kryptosystem
- Substitution
- Permutation
- Symmetrische versus asymmetrische Kryptosysteme
- Kryptoanalyse
- Abschätzung: Aufwand für Brute-Force Angriff

- **Kryptographie:**

  - Lehre von den Methoden zur Ver- und Entschlüsselung von Nachrichten

- **Kryptoanalyse, Kryptanalyse:**

  - Wissenschaft von den Methoden zur Entschlüsselung, ohne im Besitz des Schlüssels zu sein (Angriffe auf kryptographische Verfahren)

- **Kryptologie** = Kryptographie + Kryptoanalyse

- **Kryptographische Protokolle:**

  - Protokolle, die kryptographische Techniken verwenden, um z.B. Schlüssel auszutauschen, Kommunikationspartner zu authentisieren, ....

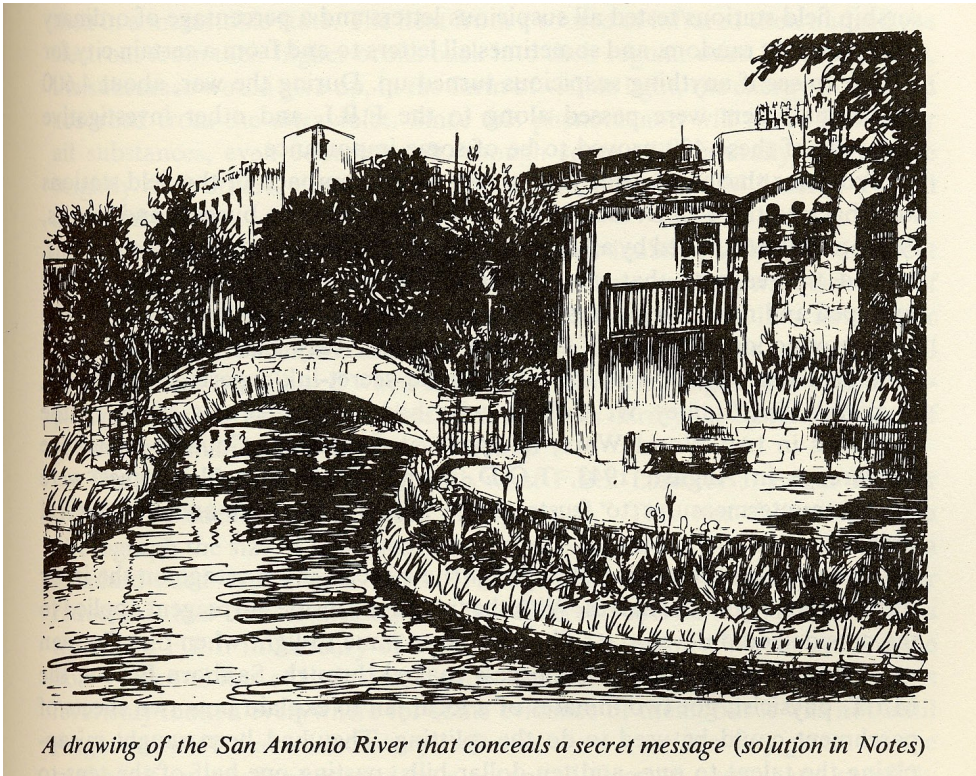
- **Steganographie** (verdecktes Schreiben):

  - Methoden, die bereits die Existenz der geheimen Nachricht verbergen (geheime Nachricht in anderer, nicht geheimen „Nachrichten“ verbergen)

  - Unterscheidung: **linguistische** und **technische** Steganographie

# Linguistische Steganographie

- **Semagramme:** Nachrichten, die in **Details** von Schriften oder Bildern verborgen sind.
- Bsp. aus David Kahn: *The Codebreakers*, Scribner, 1996



- Wo verbirgt sich die Nachricht?

## Linguistische Steganographie (Forts.)



### ■ Maskierung (Open Code):

Nachricht verborgen in offen übertragener, unverfänglicher Nachricht  
(z.B. Husten in „Wer wird Millionär“)

- **Stichworte:** Begriff, Satzteil oder Satz mit vorher vereinbarter Bedeutung;  
z.B. *HIGASHI NO KAZE AME* („Ostwind, Regen“) im japanischen Wetterbericht -  
zwei mal wiederholt - sollte „Krieg mit USA“ bedeuten.

### ■ Jargon, Millieu-Code:

Sondersprachen oder Sonderzeichen beruflicher oder gesellschaftlicher Art

- z.B. „Schnee“ für Kokain; „Kies“ für Geld; „abstauben“, ...
- Für Zensoren durch „gestelzte“ Sprache relativ leicht erkennbar.
- Umformulieren durch Synonyme kann Inhalt „zerstören“.

# Spam-Mimic



- [www.spammimic.com](http://www.spammimic.com)
- Versteckt kurze Nachricht in längerer Spam-E-Mail.

Dear E-Commerce professional ; This letter was specially selected to be sent to you . This is a one time mailing there is no need to request removal if you won't want any more . This mail is being sent in compliance with Senate bill 2316 , Title 5 , Section 306 . THIS IS NOT MULTI-LEVEL MARKETING ! Why work for somebody else when you can become rich in 55 MONTHS . Have you ever noticed nearly every commercial on television has a .com on in it and people love convenience . Well, now is your chance to capitalize on this . We will help you process your orders within seconds and increase customer response by 180% ! You can begin at absolutely no cost to you . But don't believe us ! Mrs Ames of

Tennessee tried us and says "Now I'm rich many more things are possible" . We are a BBB member in good standing . Do not delay - order today . Sign up a friend and you'll get a discount of 10% ! God Bless . Dear Friend , We know you are interested in receiving hot news ! We will comply with all removal requests ! This mail is being sent in compliance with Senate bill 2516 , Title 3 , Section 309 . Do NOT confuse us with Internet scam artists . Why work for somebody else when you can become rich within 99 WEEKS ! Have you ever noticed people are much more likely to BUY with a credit card than cash & the baby boomers are more demanding than their parents ! Well, now is your chance to capitalize on this . We will help you sell more plus decrease perceived waiting time by 170% . You can begin at absolutely no cost to you ! But don't believe us ! Mrs Ames of Kentucky tried us and says "Now I'm rich, Rich, RICH" ! This offer is 100% legal ! So make yourself rich now by ordering immediately ! Sign up a friend and you'll get a discount of 70% . Thank-you for your serious consideration of our offer .

# Technische Steganographie

- Herodot (490 v.Chr.): Nachricht auf den rasierten Schädel eines Sklaven tätowiert
- Alle Arten von „Geheimtinten“
- Steganographie in digitalen Bildern; Beispiele mit `outguess`

Original



Steganographie





# Steganographie in Bildern

- **Cover** = Bild in das die Nachricht eingebettet wird
- Finde redundante Bits im Cover
  - Least Significant Bits
  - „Rauschen“
  - Nahe zusammenliegende Farben
- Kodieren der Nachricht in diesen redundanten Bits

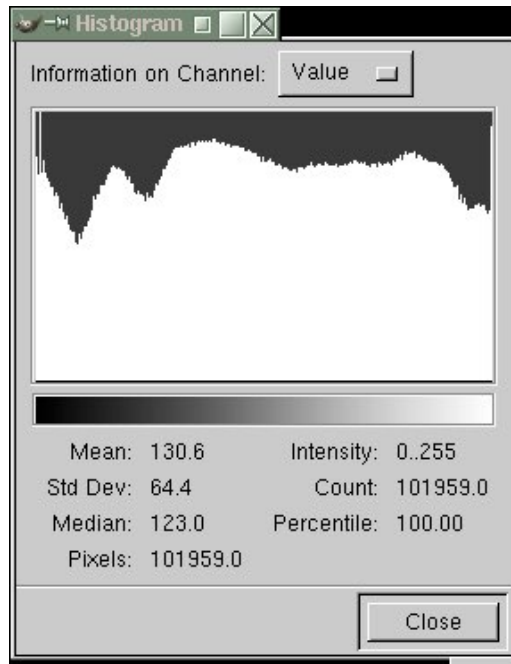
|         |      |   |   |   |   |   |   |   |
|---------|------|---|---|---|---|---|---|---|
| Pixel 1 | rot  | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
|         | grün | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
|         | blau | 1 | 1 | 0 | 1 | 1 | 0 | 0 |
| Pixel 2 | rot  | 1 | 0 | 0 | 1 | 1 | 1 | 0 |
|         | grün | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
|         | blau | 1 | 1 | 0 | 1 | 1 | 0 | 1 |

- Steganographie führt zu “sehr geringen Veränderungen” im Bild

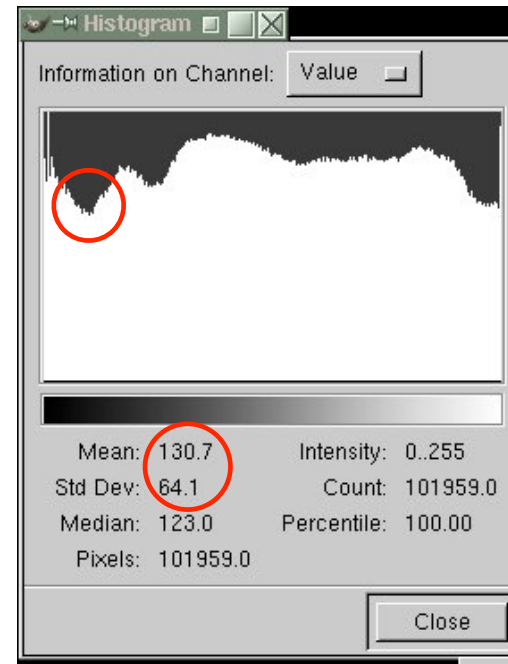
# Veränderungen im Bild

■ Histogramm:

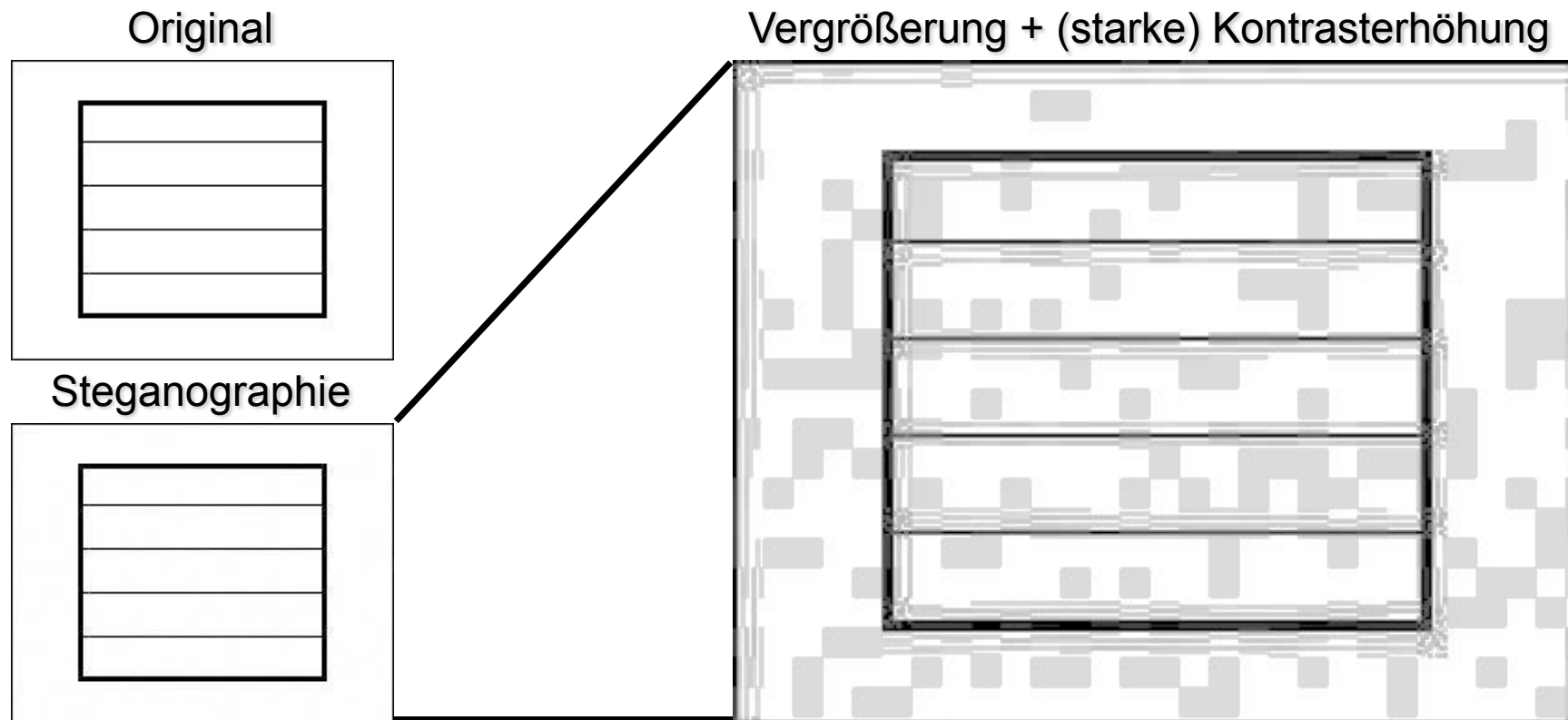
Original



Steganographie



- Unterschiede bei “sehr strukturierten Bildern” mit hohem versteckten Datenvolumen evtl. erkennbar



# Plausible Deniability (glaubhafte Abstreitbarkeit)



## ■ Praktisches Problem:

- ❑ Verschlüsselung der gesamten Festplatte schützt Vertraulichkeit der Daten
- ❑ Aber: Strafverfolgung kann evtl. Herausgabe des Passworts verlangen
  - Beispiel Großbritannien:  
2-5 Jahre Haftstrafe bei Weigerung, Passwort herauszugeben

## ■ Lösungsansatz, z.B. mit TrueCrypt/VeraCrypt:

- ❑ Verschlüsselte Festplatte enthält nur unverfängliche Dateien und ist ansonsten scheinbar leer.
- ❑ „Leerer“ Bereich enthält ein zweites, verschlüsseltes System, das von außen nicht als solches erkennbar ist.
- ❑ Zielperson gibt nur das Passwort für das äußere/erste Dateisystem preis.
- ❑ Randbedingungen in der Praxis:
  - Auf dem System sollten keine Verweise auf Dateien innerhalb des zweiten Dateisystems vorzufinden sein  
(Windows-Registry; „zuletzt benutzte Dateien“ in Anwendungen; ...).
  - Zielperson darf Existenz des zweiten Dateisystems nicht zugeben.

# Verdeckte Kanäle

- Nachrichtentransport über nicht erkennbare Kanäle/Medien
  
- Beispiele:
  - Daten im Paket-Header statt in der TCP-Payload (z.B. TCP SeqNr.)
  - Künstliches Delay in übertragenen Datenpaketen
  - Nicht Inhalt, sondern Name und Größe einer Datei sind relevant
  
- Charakterisierung durch
  - **Entdeckbarkeit (detectability):**  
Nur designierter Empfänger soll versteckte Daten erkennen können.
  - **Ununterscheidbarkeit (indistinguishability):**  
Monitor/Zensor soll bei einem ihm bekanntem verdeckten Kanal nicht erkennen können, ob aktuell versteckte Daten übertragen werden oder nicht.
  - **Bandbreite (bandwidth):**  
Länge der pro Zeiteinheit verdeckt übertragbaren Daten.

## Spreu-und-Weizen-Algorithmus



- Geheime Nachrichten sind „Nadeln im Heuhaufen“
- Alice schickt **kontinuierlich** Datenpakete an Bob
- Bob wertet aber nur einen Bruchteil aller Datenpakete aus
  - Alice und Bob müssen vorab / out-of-band ein Auswahlverfahren festlegen, um Spreu und Weizen trennen zu können.
  - Beispiel:
    - Prüfsummen-Verfahren, das nur Alice und Bob bekannt ist (oder mit einem geheimen Schlüssel parametrisiert wird)
    - Bob wertet nur Pakete mit gültiger Prüfsumme aus
- Problem ähnlich zu verdeckten Kanälen: **Geringe Bandbreite** durch viel eingestreute Spreu.

# Inhalt



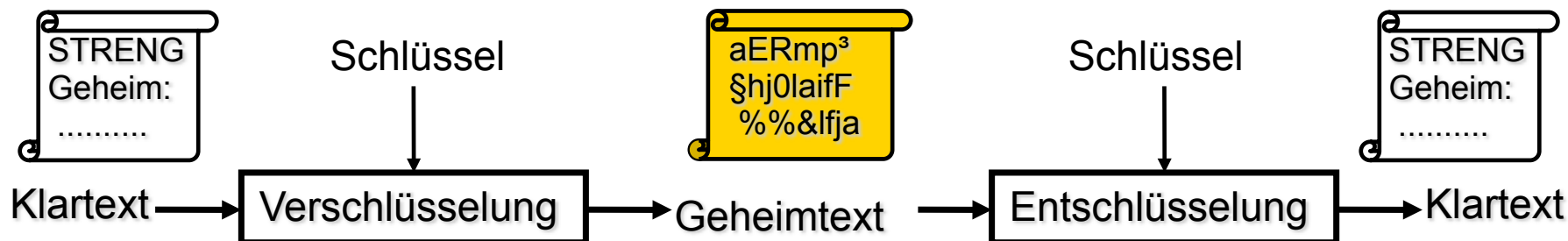
1. Kryptologie: Begriffe, Klassifikation

2. Steganographie

3. Kryptographie, Begriffe und Definitionen

- Kryptosystem
- Substitution
- Permutation
- Symmetrische versus asymmetrische Kryptosysteme
- Kryptoanalyse
- Abschätzung: Aufwand für Brute-Force Angriff

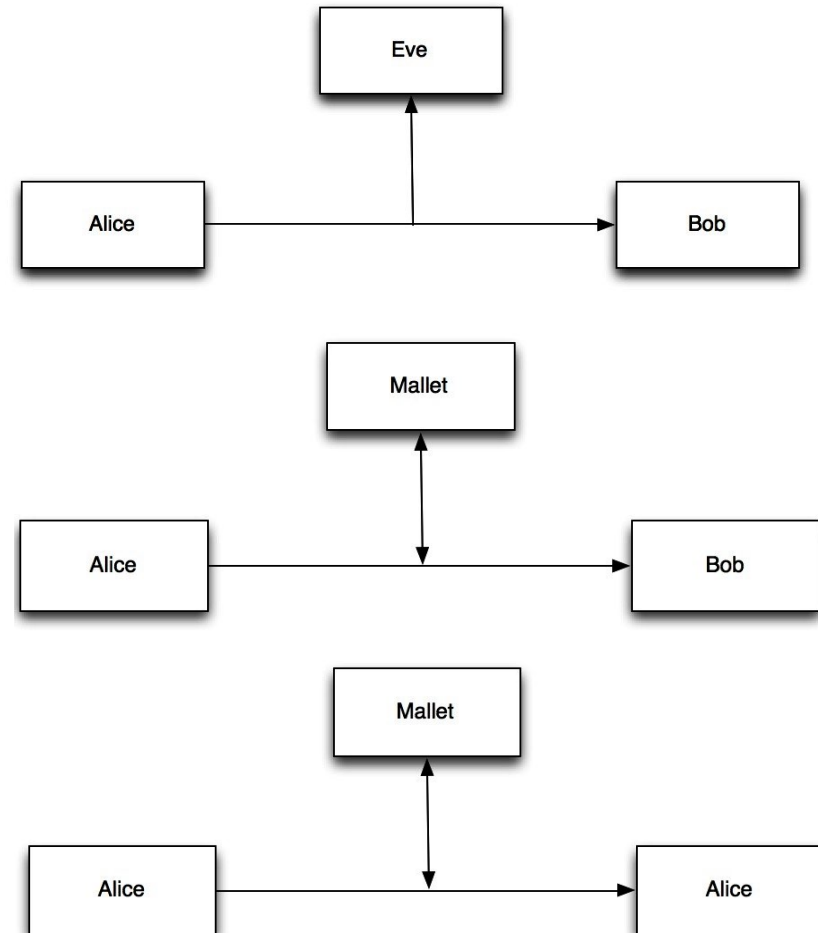
- **Klartext (Plaintext):** Zu verschlüsselnde Nachricht
- **Geheimtext (Ciphertext):** Verschlüsselte Nachricht
- **Verschlüsselung, Chiffrierung (Encryption):**  
Vorgang, der Klar- in Geheimtext (Chiffretext) überführt
- **Entschlüsselung, Dechiffrierung (Decryption):**  
Überführung von Geheim- in Klartext
- **Chiffriersystem (Cryptographic Algorithm, Cipher):**  
Algorithmisches Verfahren zur Ver- bzw. Entschlüsselung
- Algorithmen werden parametrisiert über **Schlüssel (Key)**





# Angriffsszenarien

- Eve kann die Nachrichtenübertragung (**passiv**) mithören:
- Mallet kann die Nachrichtenübertragung **aktiv** manipulieren:
- (Alice schickt sich selbst Nachrichten!?)



## Definition

# Kryptographisches System



- Ein Kryptosystem  $KS$  ist ein Fünftupel

$$KS = (M, K, C, e, d)$$

- $M$  = Nichtleere, endliche Menge aller Klartexte (Messages)
- $K$  = Nichtleere, endliche Menge aller Schlüssel (Keys)
- $C$  = Menge von Chiffretexten (Ciphertexts)
- $e = M \times K \rightarrow C$  ist Verschlüsselungsfunktion
- $d = C \times K \rightarrow M$  ist Entschlüsselungsfunktion

$$\forall k_e \in K : f(k_e) = k_d$$

$$d(e(m, k_e), k_d) = m$$

## Substitution



- **Substitution:**  $f : A_1^n \rightarrow A_2^m$
- **Alphabete:**  $A_1 = \{a, b, \dots, z\} (= Z_{25})$ ;  $A_2 = \{1, 2, 3, 4, 5\}$
- **Verschlüsselungsverfahren:**  $E : A_1^1 \rightarrow A_2^2$
- **Schlüssel**  $K_E = K_D$

|          | <b>1</b> | <b>2</b> | <b>3</b> | <b>4</b> | <b>5</b> |
|----------|----------|----------|----------|----------|----------|
| <b>1</b> | a        | b        | c        | d        | e        |
| <b>2</b> | f        | g        | h        | i        | k        |
| <b>3</b> | l        | m        | n        | o        | p        |
| <b>4</b> | q        | r        | s        | t        | u        |
| <b>5</b> | v        | w        | x        | y        | z        |

- **Beispiel**  
(pro Buchstabe Zeilen-/Spaltennummer ermitteln):  
vorlesung wird zu 513442311543453322

## Permutation

- **Permutation** als Spezialfall der Substitution:  $f : A^n \rightarrow A^n$   
gleiche Wortlänge; gleiche Alphabete  $A_1 = A_2 = \{a, b, \dots, z\}$
- $K_E = K_D$  (hier: NEWYORK)  
(Zur besseren Lesbarkeit werden Chiffrentexte trotzdem oft in Großbuchstaben dargestellt.)
- Matrixschreibweise:

|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |
| N | E | W | Y | O | R | K | A | B | C | D | F | G | H | I | J | L | M | P | Q | S | T | U | V | X | Z |

Zykelschreibweise:

(a,n,h) (b,e,o,i) (c,w,u,s,p,j) (d,y,x,v,t,q,l,f,r,m,g,k) (z)

- Beispiel:
  - TIMFOPSHKQBWPBWAOMAQBQ = vorlesung it sicherheit  
Chiffrentext wird in Blöcken übertragen  
Leer- und Satzzeichen werden nicht kodiert  
(Kryptanalyse: Leerzeichen noch häufiger als „e“)

## Symmetrische Verfahren

- Kommunikationspartner teilen **gemeinsamen, geheimen Schlüssel** (Shared Secret; deshalb: Symmetrie)
- Ver- und Entschlüsselungsschlüssel sind identisch oder jeweils trivial aus dem Shared Secret abzuleiten.
- Setzt vorherige Verständigung (**Schlüsselaustausch**) voraus.
- Protokoll:

1. Alice und Bob vereinbaren („**out of band**“) den gemeinsamen Schlüssel:

$$k_e = k_d = k_{A,B}$$

2. Alice verschlüsselt  $m$ :  $c = e(m, k_{A,B})$  und sendet  $c$  an Bob
3. Bob entschlüsselt  $c$ :

$$m = d(c, k_{A,B}) = d(e(m, k_{A,B}), k_{A,B})$$

- Beispiele: DES, AES, IDEA, RC4, Blowfish, Serpent, Twofish, ...

## Asymmetrische Verfahren



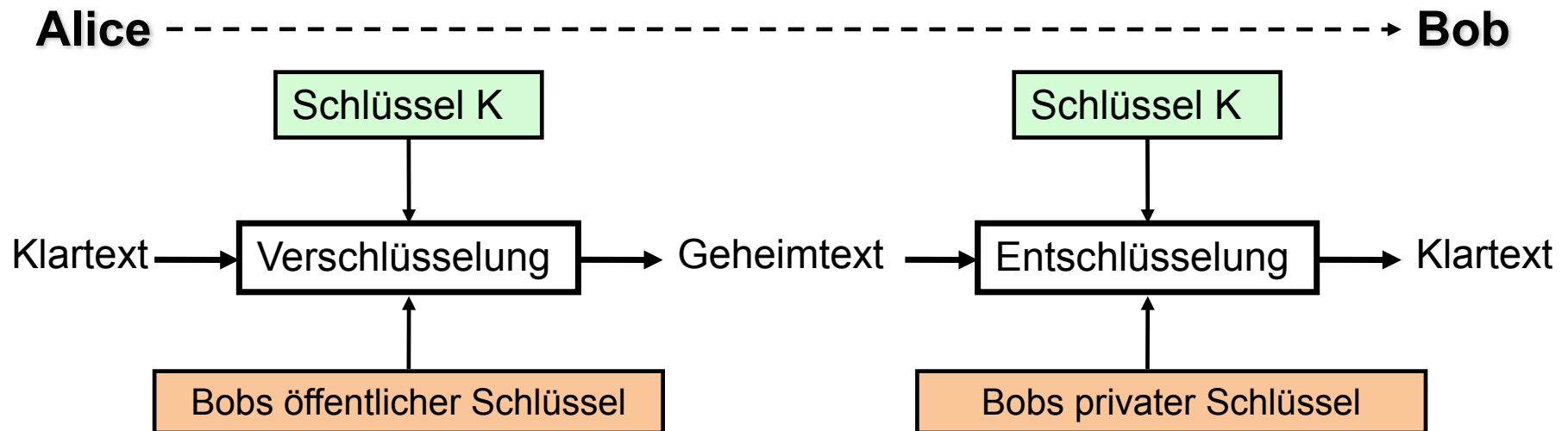
- Jeder Partner besitzt **Schlüsselpaar** aus
  - persönlichem, **geheim** zu haltenden **Schlüssel** (*private key*)  
(wird NIE übertragen)
  - und **öffentlich** bekannt zu gebenden **Schlüssel** (*public key*)  
(kann über unsichere und öffentliche Kanäle übertragen werden)
- Protokoll:
  1. Alice und Bob erzeugen sich Schlüsselpaare:  $(k_e^A, k_d^A) (k_e^B, k_d^B)$
  2. Öffentliche Schlüssel  $(k_e^A, k_e^B)$  werden geeignet öffentlich gemacht
  3. Alice will  $m$  an Bob senden; dazu benutzt sie Bobs öffentlichen Schlüssel
$$c = e(m, k_e^B)$$
  4. Bob entschlüsselt die Nachricht mit seinem privaten Schlüssel:

$$m = d(c, k_d^b) = d(e(m, k_e^b), k_d^b)$$

- Beispiele: RSA, DSA, ElGamal, ...

# Vergleich symmetrische / asymmetrische Verfahren

|                    | Symmetrisch                 | Asymmetrisch                        |
|--------------------|-----------------------------|-------------------------------------|
| Schlüsselaustausch | Sicherer Kanal erforderlich | öffentlich (aber: Authentizität!)   |
| Schlüssellänge     | meist 128 oder 256 Bit      | meist 2048 bis 8192 Bit             |
| Geschwindigkeit    |                             | meist Faktor 100 bis 1000 langsamer |



# One-Time-Pads

- Bei richtiger Verwendung „unknackbare“ Verschlüsselung
- Schlüssel
  - ist (mindestens) **genauso lang wie der Klartext**,
  - ist zufällig („*truly random*“) gewählt, und
  - wird **niemals wiederverwendet**.
- XOR-Verknüpfung von Klartext- mit Schlüssel-Zeichen.
- Praktische Einschränkungen:
  - **Schlüsselmanagement extrem aufwendig**
    - Großer Bedarf an „echten“ Zufallszahlen nicht einfach zu decken.
    - Alice und Bob müssen Schlüssel sicher untereinander austauschen.
  - Keine implizite Integritätssicherung (Angreifer modifiziert Ciphertext, so dass sich bei der Entschlüsselung ein sinnvoller anderer Plaintext ergibt)



- Wissenschaft von Methoden zur Entschlüsselung **ohne** Vorabkenntnis des Schlüssels
- Klassen kryptanalytischer Angriffe:
  - **Brute force; exhaustive search:** vollständiges Durchsuchen des Schlüsselraums
  - **Angriff auf Chiffren (ciphertext-only):** Dem Analytiker stehen mehrere Chiffren zur Verfügung. Ziel: Schlüssel und/oder Klartext berechnen
  - **Bekannter Klartext (known-plaintext):** Analytiker kennt Klartext-/Chiffren-Kombinationen, die mit selbem Schlüssel verschlüsselt wurden. Ziel: Schlüssel brechen oder Algorithmus finden, der jede mit dem Schlüssel verschlüsselte Nachricht entschlüsseln kann.
  - **Gewählter Klartext (chosen-plaintext):** Analytiker kann selber Klartexte wählen und diese verschlüsseln lassen.
  - **Gewählte Chiffre (chosen-ciphertext):** Angreifer kann sich zu ausgewählten Chiffren den Klartext berechnen lassen.
- Weitere Informationen: Vgl. F.L. Bauer: Entzifferte Geheimnisse

## Aufwand für Brute-Force-Angriff

- Annahmen, unter denen Brute-Force-Angriff sinnvoll erscheint:
  - Schlüssel ist zufällig gewählt, d.h. alle Schlüssel sind gleich wahrscheinlich
  - Es gibt kein alternatives, schneller Erfolg versprechendes Verfahren
  
- Die Schlüssellänge sei 128 Bit
- Ein Rechner schaffe 3.000.000.000 Schlüssel pro Sekunde
- Der Angreifer habe 1.000 Rechner zur Verfügung
- Schlüsselraum  $S = 2^{128} \approx 3,4 \cdot 10^{38}$
- 1 Jahr hat 31.557.600 Sekunden
- Maximaldauer D in Jahren:  
 $D = S / (3.000.000.000 \cdot 1.000 \cdot 31.557.600) = 3,6 \cdot 10^{18}$  Jahre  
(im Durchschnitt also  $1,8 \cdot 10^{18}$  Jahre)
- Bei Schlüssellänge 256 Bit:  $D = 1,2 \cdot 10^{57}$  Jahre

## Warnstufe ROT - Schwachstelle in log4j veröffentlicht



- 11.12.21: BSI, LSI und andere warnen vor kritischer Schwachstelle CVE-2021-44228 mit Score 10.0 (von 10.0 möglichen)
  - log4j beliebte java Bibliothek zur Aggregation von Logs
  - Ermöglicht Ausführung von beliebigem Code, Kompromittierung des Servers
  - PoC veröffentlicht und auf twitter geteilt Log4Shell, Bsp:
    - Angreifer schreibt `{jndi:ldap://127.0.0.1:1389/a}` ins Log
    - Verzeichnisdienst jndi kontaktiert ldap-Server und nimmt pot. böartige Java Klassen entgegen und führt diese aus
  - Breitflächige Scans werden bereits durchgeführt
  - möglicherweise Auswirkungen auf alle aus dem Internet erreichbaren Server-Anwendungen die log4j zur Protokollierung der Nutzeranfragen einsetzen
  - Breite Verbreitung von log4j - Betroffenheit nur schwer absehbar
  - Betroffene Dienste und Hersteller: <https://github.com/YfryTchsGD/Log4jAttackSurface>

## Warnstufe ROT - Schwachstelle in log4j veröffentlicht



### ■ Gegenmaßnahmen:

- ❑ Installation der aktuellen Version 2.15.0-rc2
- ❑ Patchmanagement von Java Anwendungen nicht trivial
- ❑ Kurzfristige Mitigation / Workaround empfohlen:
  - ❑ Option `log4j2.formatMsgNoLookups` oder Umgebungsvariable `LOG4J_FORMAT_MSG_NO_LOOKUPS` auf `true` setzen
  - ❑ Klasse `JndiLookup` aus dem Klassenpfad löschen:  
`zip -q -d log4j-core-*.jar org/apache/logging/log4j/core/lookup/JndiLookup.class`
- ❑ Allgemeine Maßnahmen
  - ❑ Server sollten nur zwingend benötigte Verbindungen ins Internet aufbauen dürfen
  - ❑ Nicht zwingend benötigte Systeme abschalten
  - ❑ Netze segmentieren, verwundbare Systeme von internen Systemen separieren