Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

April 2020

Vorlesung
Verteilte Systeme und Systemsicherheit (VSS)

Prof. Dr. Hannes Federrath
Sicherheit in verteilten Systemen (SVS)
http://svs.informatik.uni-hamburg.de

Wegweiser durch die Vorlesung

- Einführung in verteilte Systeme und Betriebssysteme
- Safety (Backup, RAID, Fehlertoleranz)
- Prozesse, Synchronisation und Koordination
- Datenkommunikation und Rechnernetze
- IT-Security
- Rechnersicherheit
- Firewalls und Malware
- Grundlagen der Kryptographie
- Steganographie (und Watermarking)

Die bereitgestellten Kursmaterialen dienen ausschließlich dem persönlichen Gebrauch. Die Veröffentlichung, Vervielfältigung, Verbreitung oder Weitergabe, auch auszugsweise, ist nur mit schriftlicher Genehmigung des Verfassers erlaubt.

Organisatorisches

Ziele der vorlesungsbegleitenden Übungen

- Verständnisprobleme lösen
  - Offene Fragen der Vorlesung können im kleineren Kreis geklärt werden.
- Vertiefung und Anwendung des Vorlesungsstoffs
  - In der Vorlesung wird teilweise nur eine Auswahl des Stoffs präsentiert.
  - Bearbeitung von Aufgaben mit Bezug zur Vorlesung
  - Bearbeitung von Aufgaben, die über die Vorlesung hinausgehen
- Vorstellung und Diskussion der Lösungen innerhalb der Übungsgruppe
  - Wissenslücken können nur beseitigt werden, wenn sie bekannt sind.
  - Aktive Beteiligung ist notwendig und sinnvoll.
- Übungsstoff ist auch klausurrelevant!

Übungleistung

- Übungsleistung wird Online erbracht
  - https://lernen.min.uni-hamburg.de/course/view.php?id=165
  - Anmelden mit Kennung und Passwort der Uniseiten
- Voraussetzung für erfolgreiches Bestehen der Übung (als Teil des Moduls)
  - Alle Übungstests bis auf einen müssen erfolgreich absolviert werden.
  - Ein Test ist erfolgreich absolviert, wenn mindestens 75 Prozent korrekt gelöst wurden.
- Randbedingungen
  - beliebig viele Versuche bis zum Ablauf der Frist
  - bestes Ergebnis wird gewertet
  - Ergebnisse werden direkt angezeigt
- Zeitplan für die Abgabe der Übungstests
  - Freischaltung der Übung am Montag Mittag (vorher nicht sichtbar)
  - Bearbeitungszeit je Blatt typischerweise 2 Wochen
  - Abgabefrist ist auf Blatt angegeben (Montag Mittag)

Übungen

- Durchführung
  - Hintergründe zu den Übungsaufgaben werden erklärt
  - Aufgaben werden erläutert und besprochen
  - Gelegenheit zum Stellen von Fragen

Übungen zu machen hilft dabei, den Vorlesungsstoff besser zu verstehen.

- Es gibt typischerweise 8 bis 10 Übungstermine bzw. -gruppen.
  - Termine für das Sommersemester 2020:
    - Di 10-12 Uhr
    - Mi 10-12 Uhr
    - Do 10-12 Uhr
  - https://bbb.informatik.uni-hamburg.de/b/joh-9fy-ue6

Die Übungen werden im Sommersemester 2020 per Videokonferenz angeboten.

Klausur

- Formalia
  - 60 Minuten Dauer
  - Mit Unterlagen (»open book«)
  - Nicht-programmierbarer Taschenrechner erlaubt

- Inhalte
  - Inhalte der Vorlesung
  - Inhalte der Übung

- Klausurtermine zu finden unter
  - https://www.inf.uni-hamburg.de/studies/orga/dates.html

- Bitte prüfen Sie bei Bedarf die Barrierefreiheit des Zugangs zu den Räumen und nehmen frühzeitig Kontakt zu uns auf, sollten Sie hierbei Probleme feststellen.

Kommunikation zur Vorlesung und Übung

- Infos wo?
  - Ankündigungen, Verschiebungen in Stine und/oder im Moodle
  - Vorlesungsfolien in Stine
  - Übungsaufgaben im Moodle

- Fragen?
  - zum Übungsbetrieb: Übungsleiter des Vertrauens fragen
  - zur Vorlesung: Prof. Dr. Hannes Federrath fragen
  - gerne auch per E-Mail

- Alle zur Vorlesung und Übung angemeldet?
  - Wenn nicht: umgehend nachholen!

Einführung in verteilte Systeme und Betriebssysteme

Def. Verteiltes System

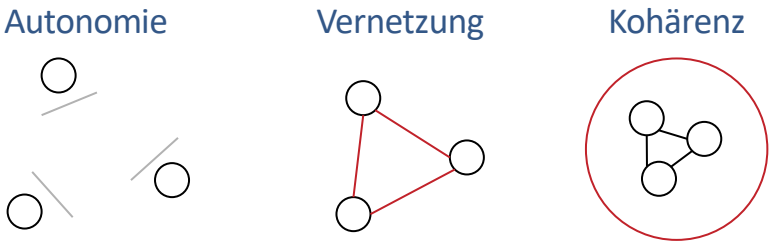
Ein verteiltes System ist eine vernetzte Anordnung von autonomen Berechnungsfunktionen, die aus Sicht ihres Nutzers als ein einzelnes kohärentes System erscheint.

- Verteilte Berechnungsfunktion (auch: Knoten, Nodes)
  - Knoten bestehen aus Hard- und Softwarekomponenten
- Autonomie
  - Knoten handeln eigenständig und unabhängig
  - Koordinations- und Synchronisationanforderungen
  - keine globale Zeit
- Vernetzung
  - Aufgabenerfüllung erfordert Kommunikation der autonomen Knoten
- Kohärenz
  - Knoten bilden aus Sicht des Nutzers ein zusammenhängendes Ganzes (lat. cohaerere zusammenhängen)

Def. Verteiltes System

Ein verteiltes System ist eine vernetzte Anordnung von autonomen Berechnungsfunktionen, die aus Sicht ihres Nutzers als ein einzelnes kohärentes System erscheint.

- Autonomie
- Vernetzung
- Kohärenz



eng verbunden:

- Transparenzeigenschaft verteilter Systeme: Endnutzer hat keine Kenntnis (mehr) darüber, wo und wie innerhalb des verteilten Systems
  - die Berechnung stattfindet und
  - die Daten gespeichert werden.

- Vorteile verteilter Systeme
  - Verteilungstransparenz (distribution transparency)
  - Resourceneffizienz
  - Offenheit
  - Skalierbarkeit

Transparency	Characteristics
Access	Hide data representation and how an object is accessed
Location	Hide where an object is located
Relocation	Hide that an object may be moved to another location while used
Migration	Hide that an object may move to another location
Replication	Hide that an object is duplicated
Concurrency	Hide that an object may be shared by several independent users
Failure	Hide the failure and recovery of an object

- Vorteile verteilter Systeme
  - Verteilungstransparenz (distribution transparency)
  - Resourceneffizienz
    - Mehrfachnutzung (sharing)
  - Offenheit
    - wohldefinierte Schnittstellen
    - Interoperabilität
    - Portabilität, bestenfalls Plattformunabhängigkeit
    - Erweiterbarkeit
  - Skalierbarkeit: Anpassbarkeit des Systems hinsichtlich
    - Anzahl der Knoten (size)
    - Entfernung der Knoten zueinander (location)
    - organisationsspezifischer Aspekte (administrative)

Bezahlung von Diensten
Datenschutzaspekte
Managementaspekte
Schutzrechte

Allgemein: Policies

Anpassbarkeit des Systems hinsichtlich Anzahl der Knoten

- Lösungsansätze für die Skalierbarkeit
  - asynchrone Kommunikationsprotokolle
  - Threading und Parallelisierung
- Kompromiss zwischen Übertragungskosten und Rechenzeit

Verteiltes Rechnen

- entweder: Verlagerung von teuren Berechnungen in die Cloud
- oder: Vorberechnungen im Peer (Edge Computing)

Verteiltes Speichern

- Caching
- Mirroring
- Replication

- Neu zu lösende Probleme
  - Konsistenz: gleiche Daten und Ergebnisse auf autonomen Knoten
  - Synchronisation: Abgleich der Daten und Ergebnisse zwischen den autonomen Knoten

Verteilte Systeme

- Architekturen verteilter Systeme
  - Multiprozessor- und Multicore-Architekturen
  - Cluster- und Grid-Computing
  - Cloud-Architekturen
  - Mobile Networks
  - Sensor Networks

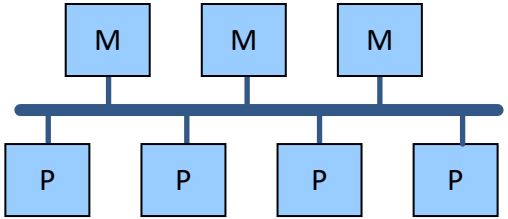
- Beispiele verteilter Systeme
  - Cloud-basierte Speicher und Cloud Computing
  - World Wide Web (http(s) und Browser)
  - E-Mail (Client-Sever)
  - Peer-to-Peer Systeme
  - Location-based Services
  - Mobilfunksysteme (2G, 3G, ...)

- Multiprozessor- und Multicore-Architekturen

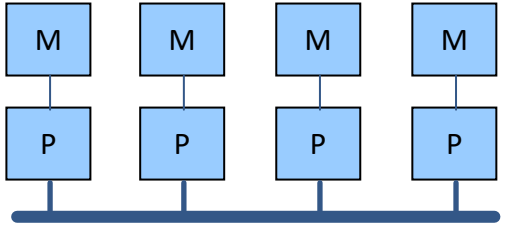
Multiprozessor-Architekturen:
P teilen sich gemeinsamem Speicher

Multicore-Architekturen:
autonome P-M-Zuordnung

Shared Memory



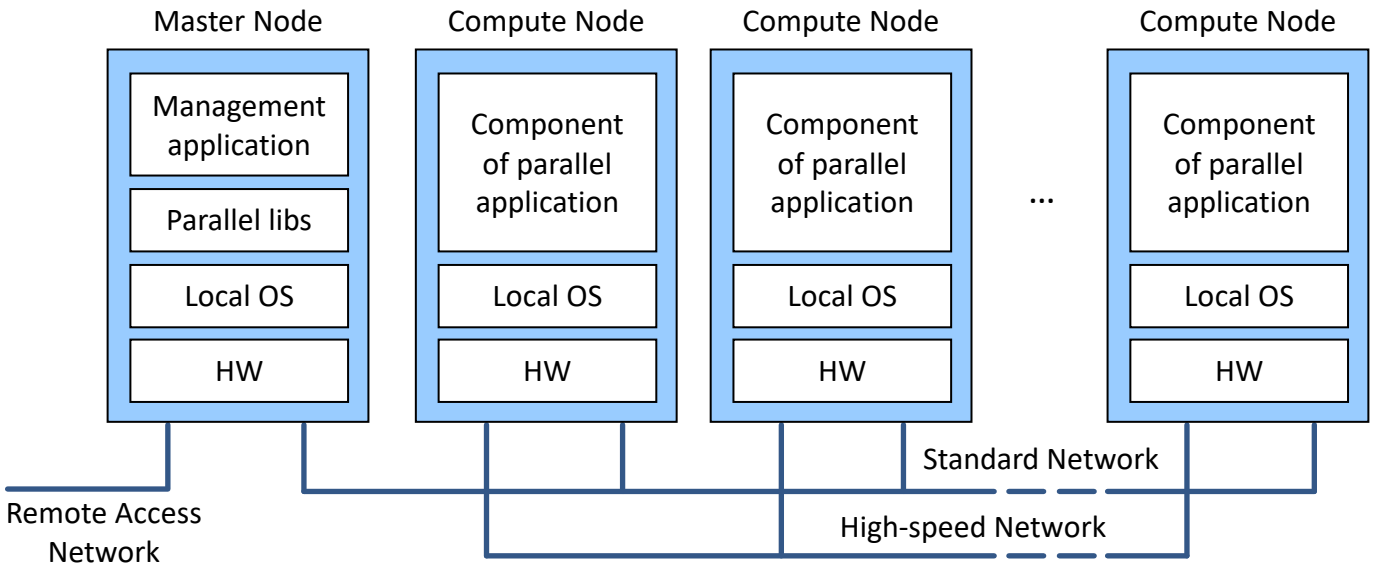
Private Memory



M Memory

P Processor

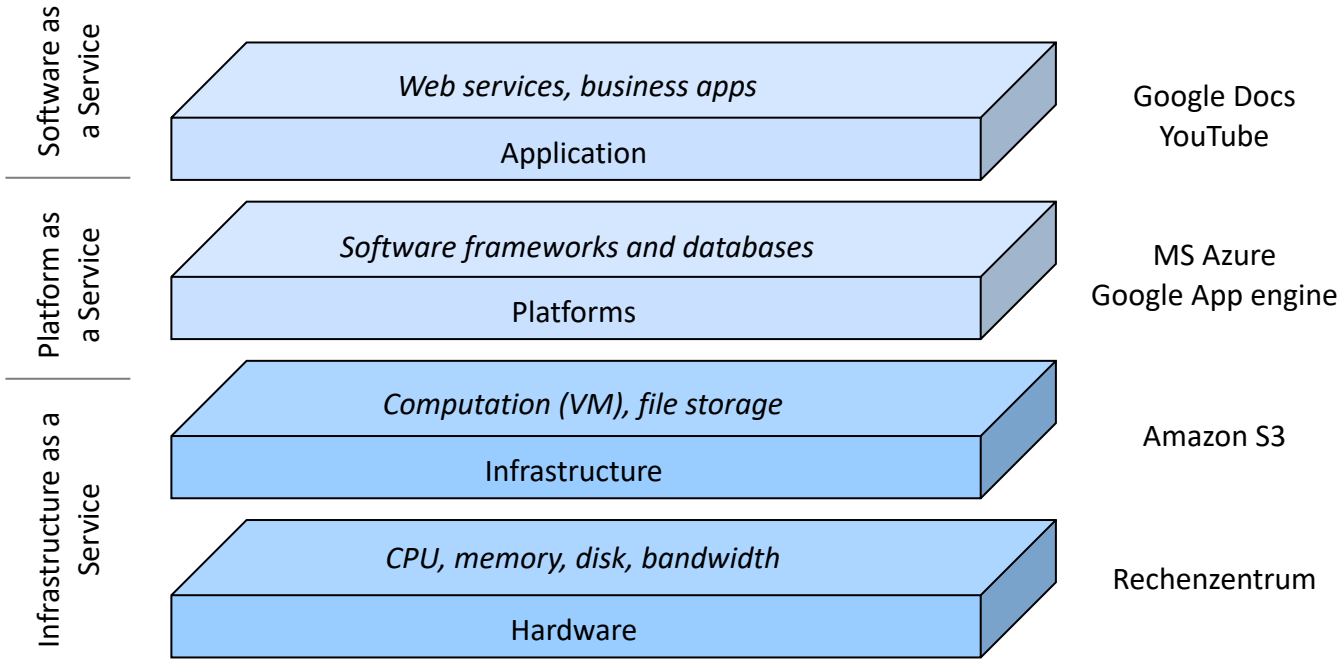
- Cluster- und Grid-Computing
  - Homogene Anordnung mehrerer gleichartiger vernetzter Knoten
  - Zentralstelle (Master Node) übernimmt Koordinationsaufgabe



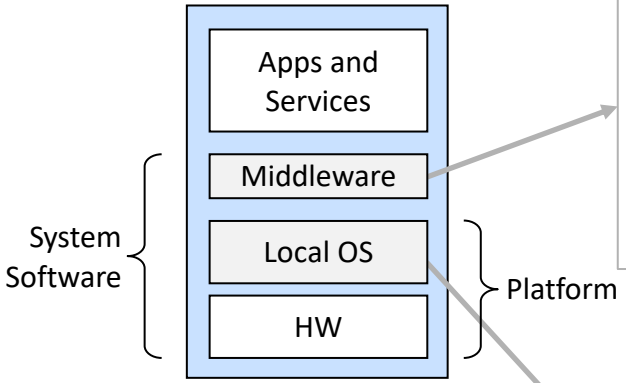
Cluster: LAN (Local Area Network)
Grid: WAN (Wide Area Network)

VSS 17.04.2020

- Cloud-Architektur



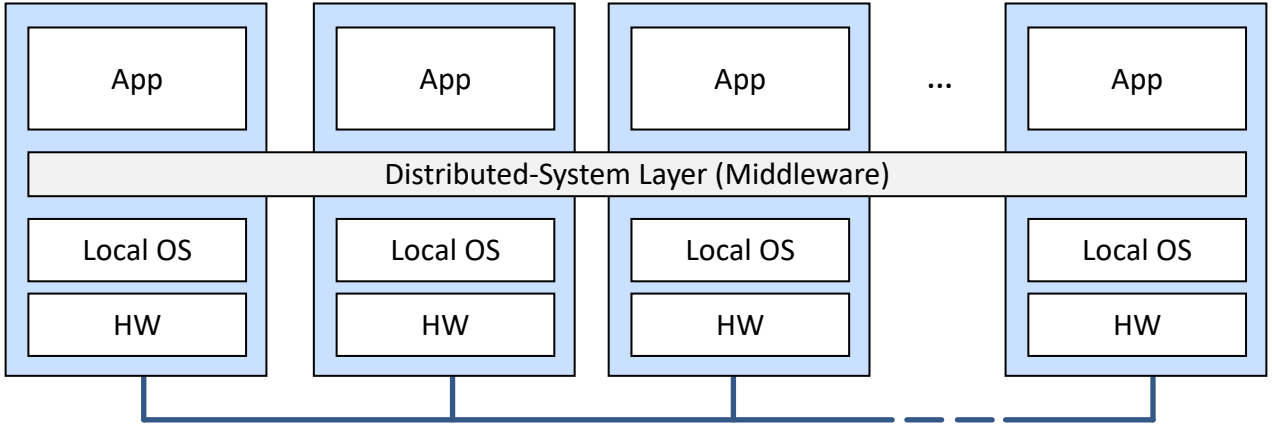
- Software and hardware service layers in distributed systems



»middleware is the same to a distributed system as what an operating system is to a computer: a manager of resources offering its applications to efficiently share and deploy those resources across a network.« M. van Steen, A. Tanenbaum, 2017

- Abstrakte / virtuelle Maschine
- Betriebsmittelverwalter
- Isolation / Rechtemanagement

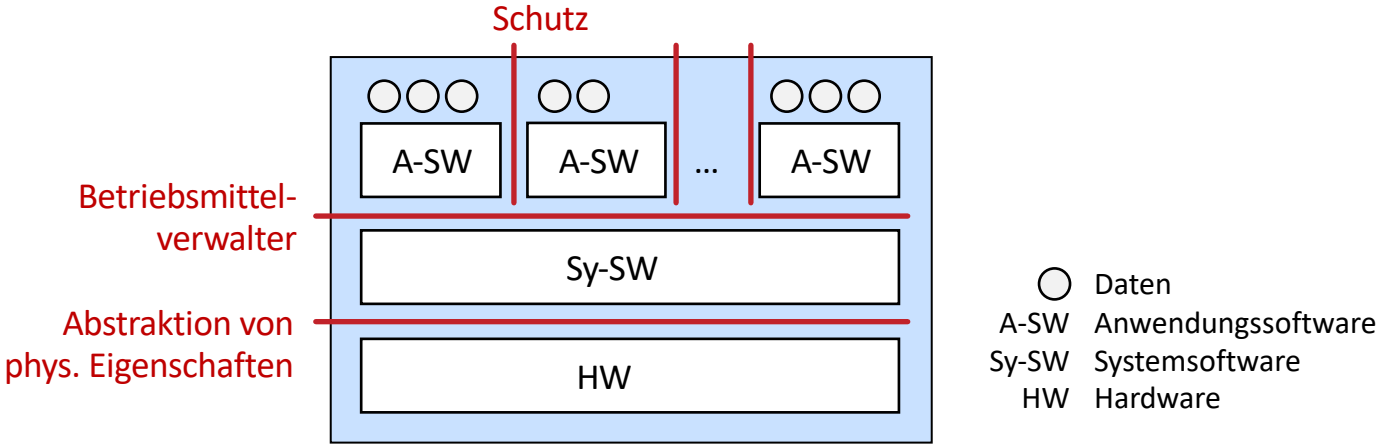
- Middleware
  - »middleware is the same to a distributed system as what an operating system is to a computer: a manager of resources offering its applications to efficiently share and deploy those resources across a network.«



Middleware bietet für alle Systeme eine einheitliche Dienstschnittstelle an.

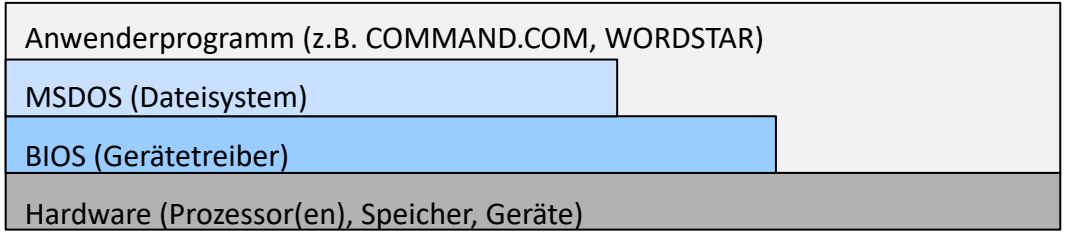
(Verteiltes) Betriebssystem

- Wesentliche Aufgaben
  - (komfortable) Abstraktion von physikalischen Systemeigenschaften des zugrunde liegenden (Hardware)-Systems (abstrakte / virtuelle Maschine)
  - Betriebsmittelverwalter für übergeordnete Anwendungsprozesse (Allokation von Speicher, Geräten, Dateien, Rechenzeit)
  - Durchsetzen von Schutz- und Sicherheitseigenschaften zwischen und gegenüber den Anwendungsprozessen und –daten (Isolation, Rechte)

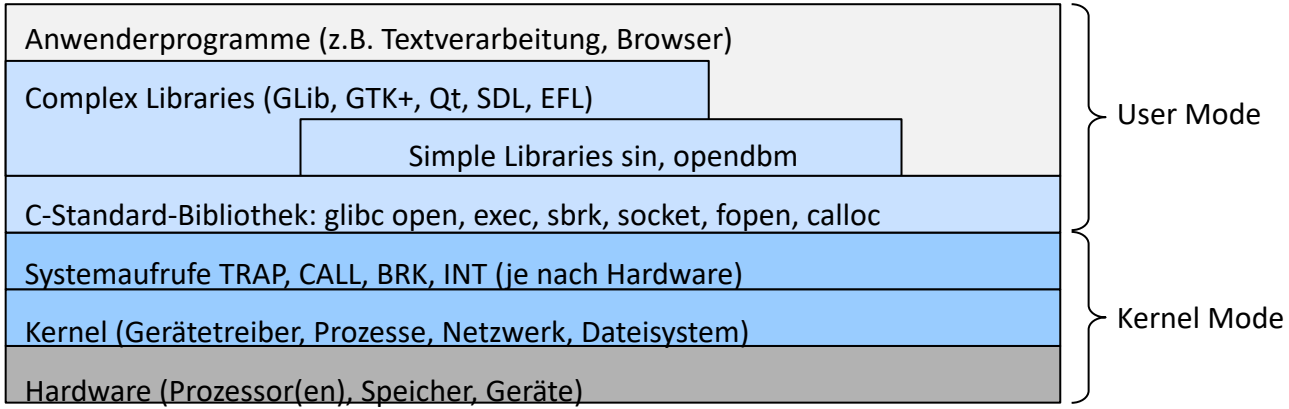


Quelle: Wikipedia

Abstraktionsschichten eines PC unter DOS

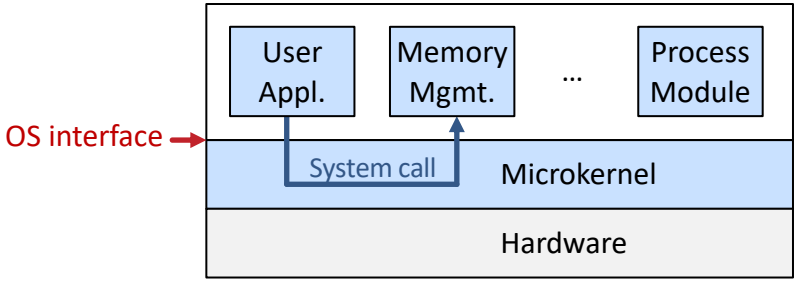


Abstraktionsschichten unter Linux



Betriebssystem-Architekturen

- Ein-Prozessor-Architektur
  - Microkernel



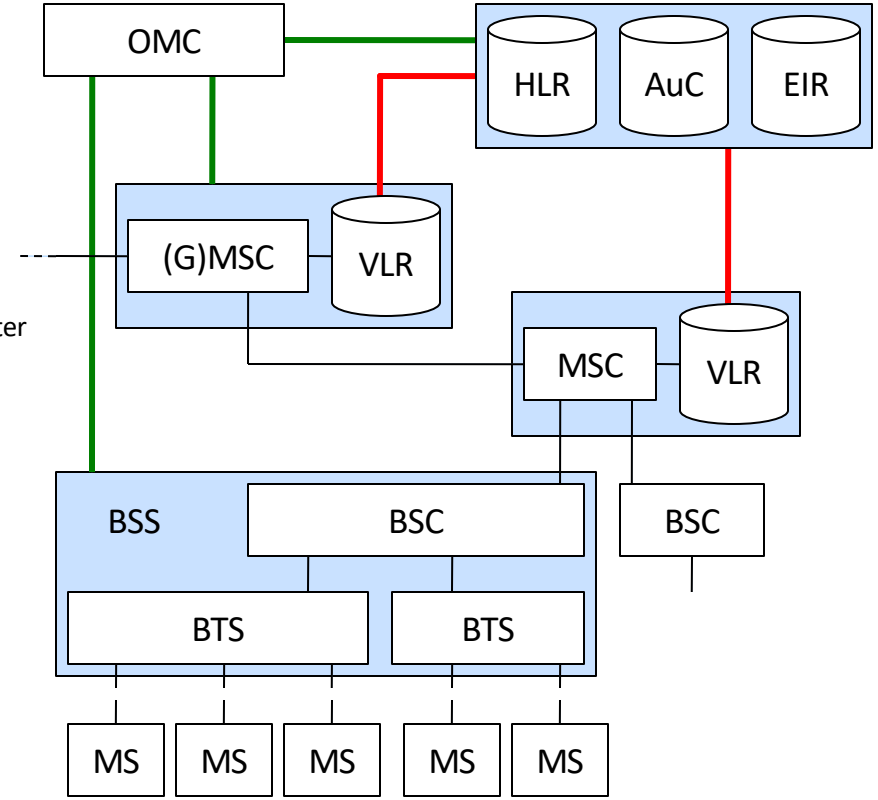
- Mehrprozessor-Architekturen
  - Multiprocessor- und Multicore-Architekturen
  - Cluster- und Grid-Architekturen
  - Cloud-Architekturen
  - Mobile Networks
  - Sensornetze

- Mobile Networks

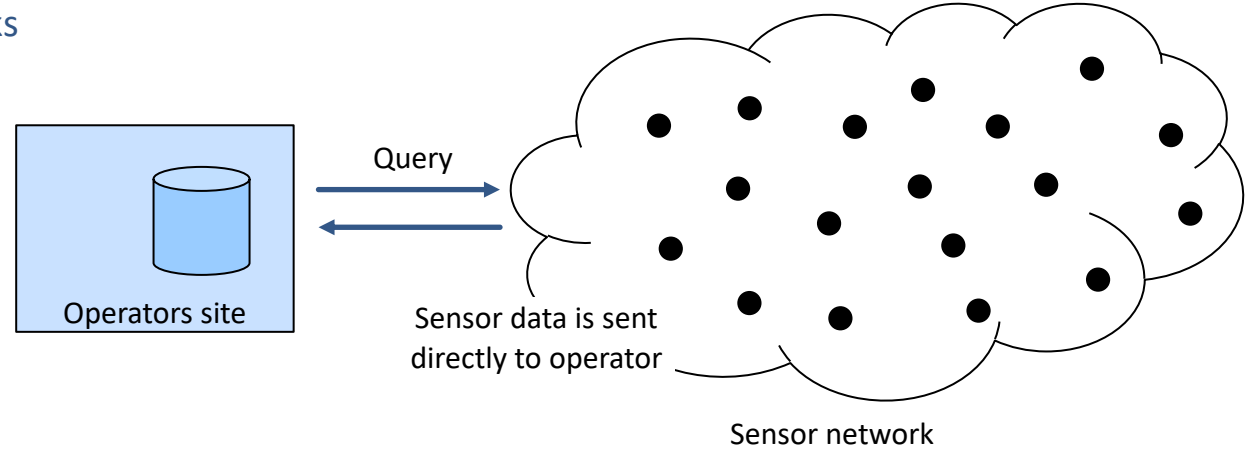
Beispiel GSM (2G)

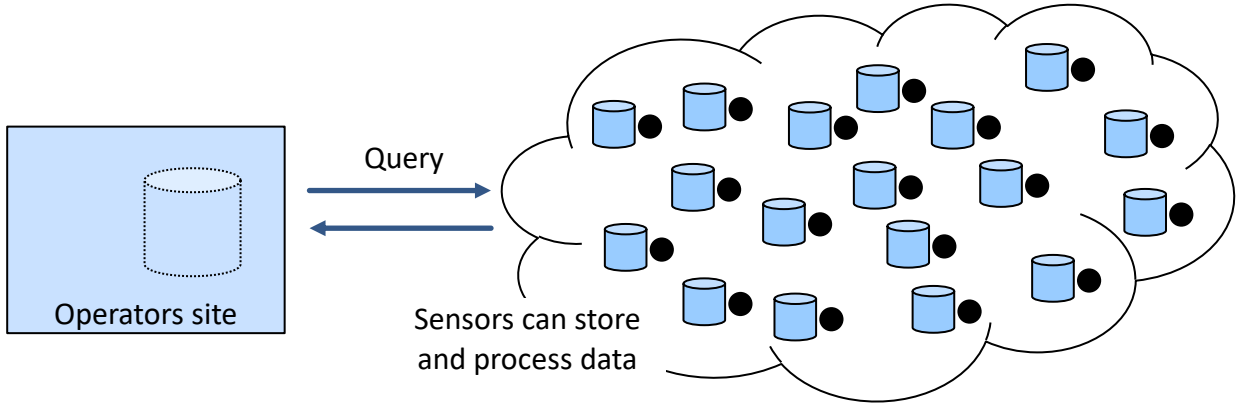
Network Management
Call Management
Database Management

OMC: Operation and Maintainance Center
HLR: Home Location Register
AuC: Authentication Center
EIR: Equipment Identity Register
MSC: Mobile Switching Center
GMSC: Gateway MSC to fixed network
VLR: Visitor Location Register
BSS: Base Station Subsystem
BSC: Base Station Controller
BTS: Base Transceiver Station
MS : Mobile Station
LA : Location Area

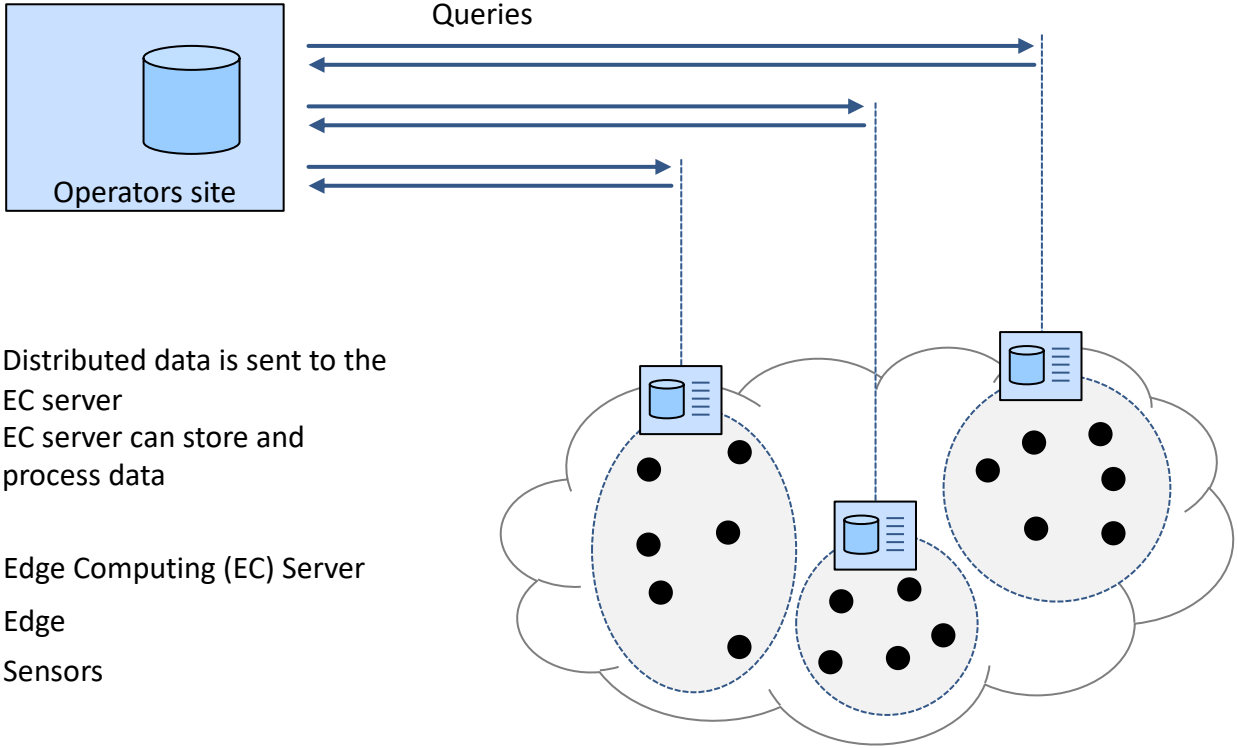


- Sensor Networks





Edge Computing



- Distributed data is sent to the EC server
- EC server can store and process data

Edge Computing (EC) Server

Edge

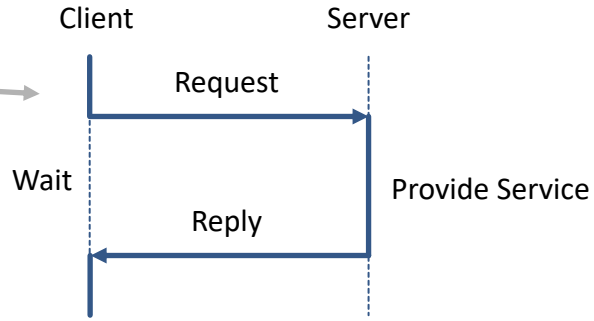
Sensors

Sichten auf die Architektur verteilter Systeme (views)

- Schichtenorientierte Sicht, z.B.
  - Cloud-Architektur
  - Open Systems Interconnection (OSI) Referenzmodell

Das OSI-Referenzmodell ist ein abstraktes, logisch-funktionelles Architekturmodell der ISO (International Standards Organization) für die Datenkommunikation in offenen Systemen (Open Systems Interconnection, OSI).

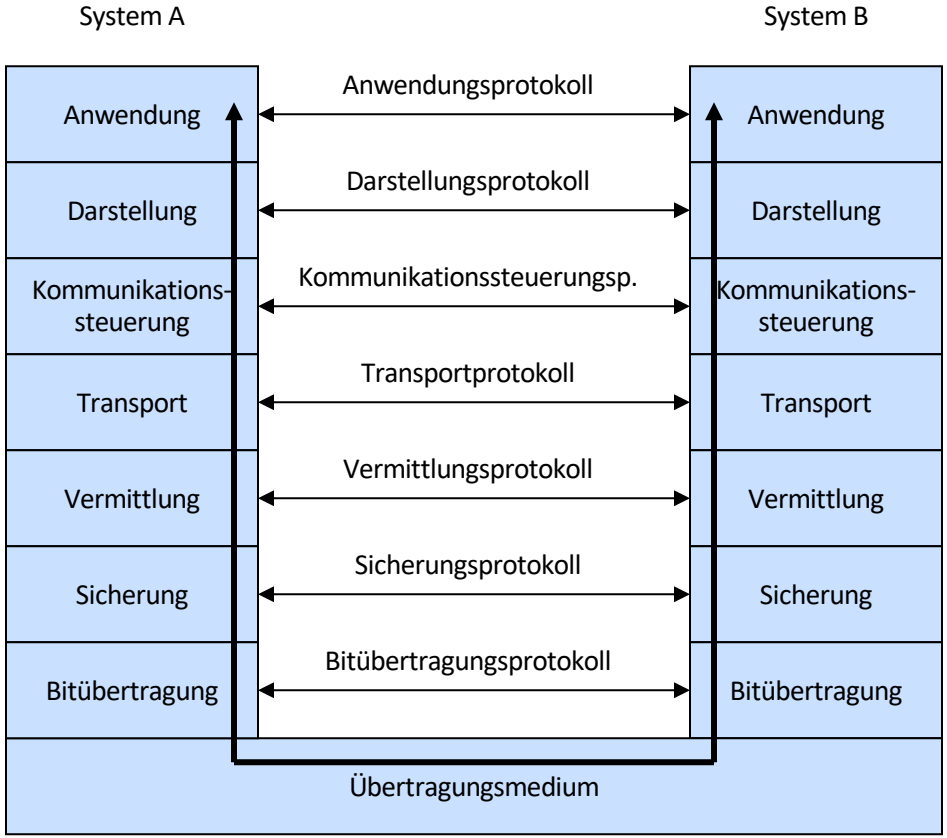
- Kommunikationsorientierte Sicht
  - Protokolle



- Topologieorientierte Sicht, z.B.
  - Client-Server
  - Peer-to-Peer (P2P)
  - Broker

Die 7 Schichten des OSI-Referenzmodells

ISO, 1977

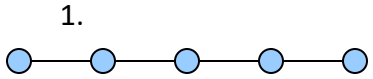


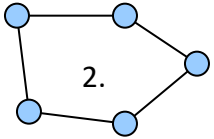
VSS 17.04.2020

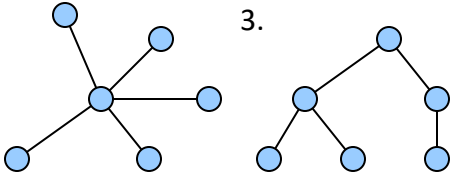
Topologien

Die topologische Struktur gibt an, wie die *Knoten* wechselseitig zugeordnet sind.

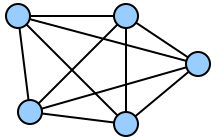
- Topologien
  1. Liniennetze
  2. Ringnetze
  3. Stern- und baumartige Netze
  4. Vermaschte Netze
     - 4.1 Vollvermaschung: jeder Teilnehmer mit jedem
     - 4.2 Teilvermaschung

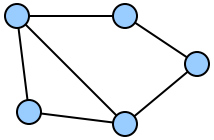






- ein- oder mehrdimensional darstellbar

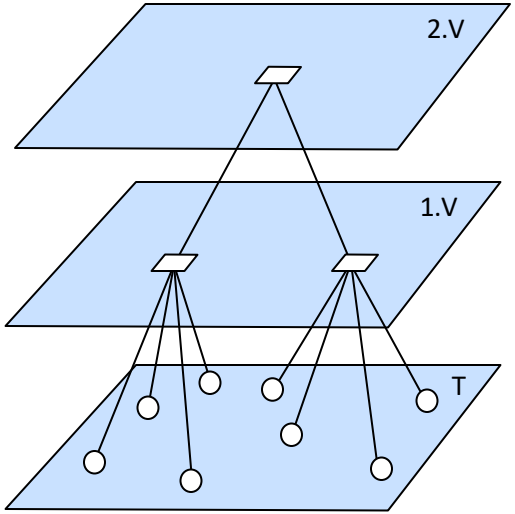




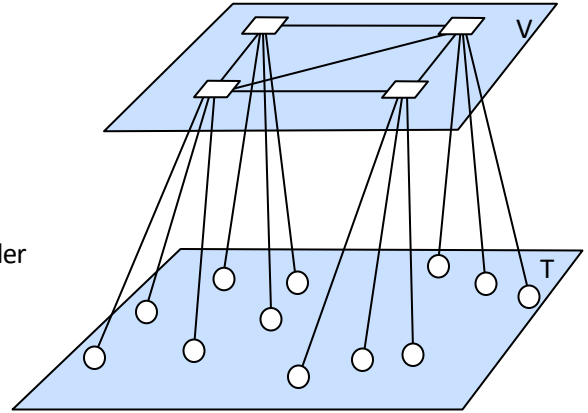
Mehrdimensionale Darstellung von Netztopologien

- Unterscheidung nach
  - Teilnehmerebene (Datenendstellen)
  - Vermittlungsebenen (ausschließlich Weiterlenkung der Datenströme)

T…Teilnehmerebene
V…Vermittlungsebenen



Kommunikationspartner der Teilnehmerebene in öffentlichen Netzen sind normalerweise nicht vermascht

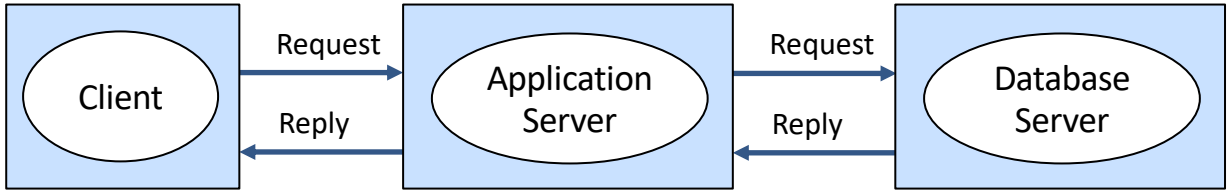


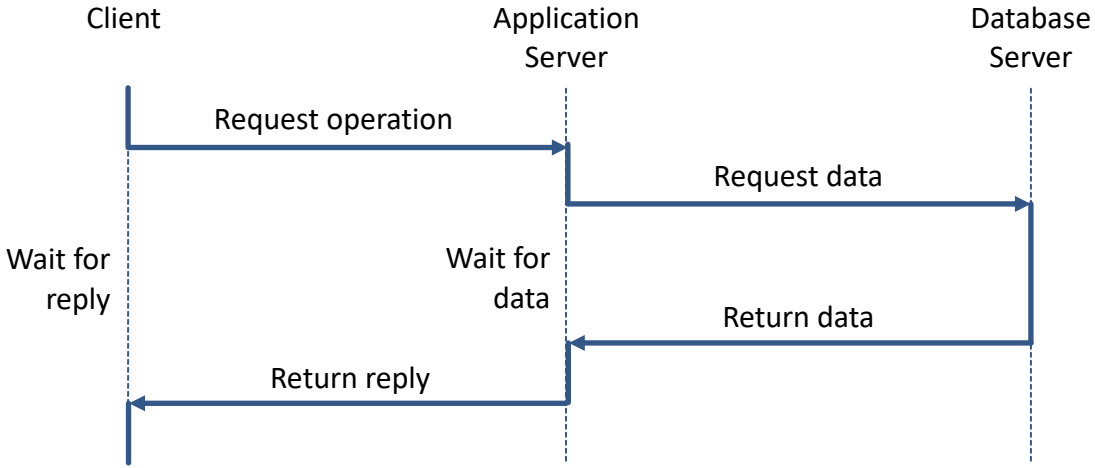
Sternnetz

vermaschtes Netz

Client-Server-Architektur

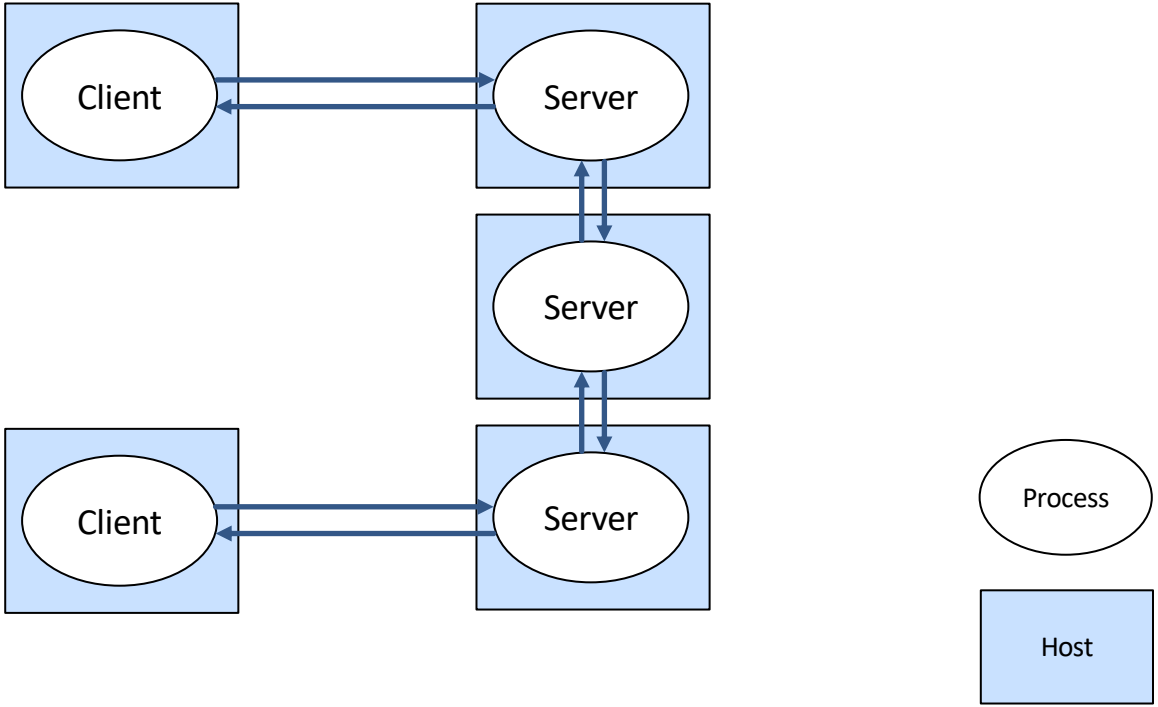
- Clients invoke individual servers





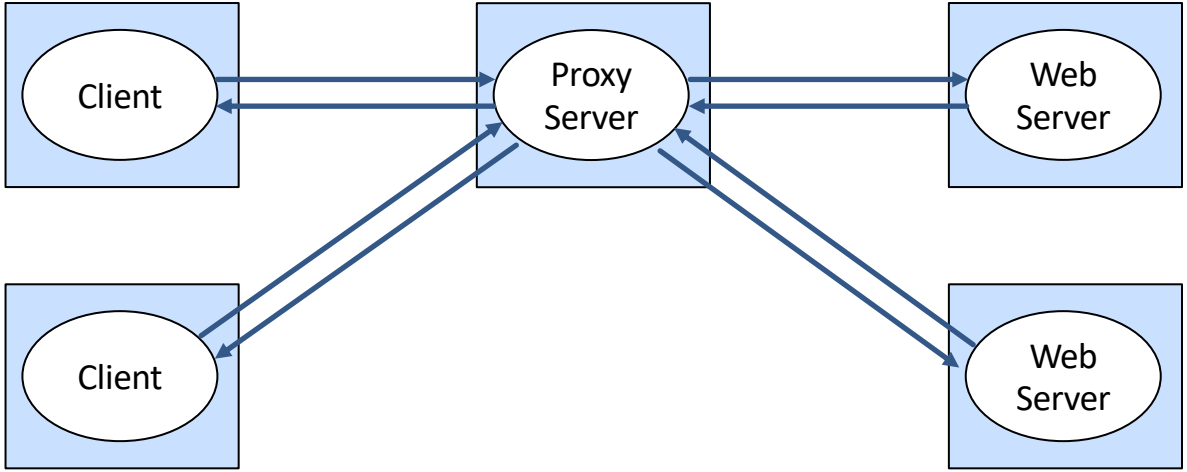
Client-Server-Architektur

- A service provided by multiple servers



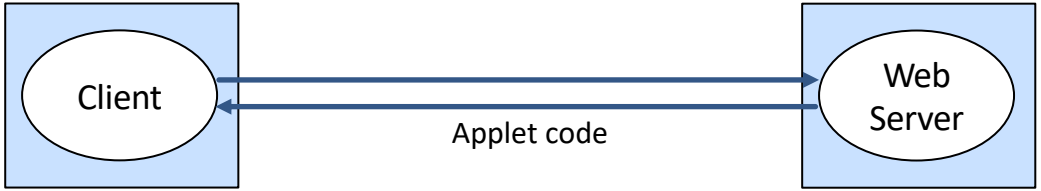
Client-Server-Architektur

- Web Proxy Server: A shared cache of Web resources

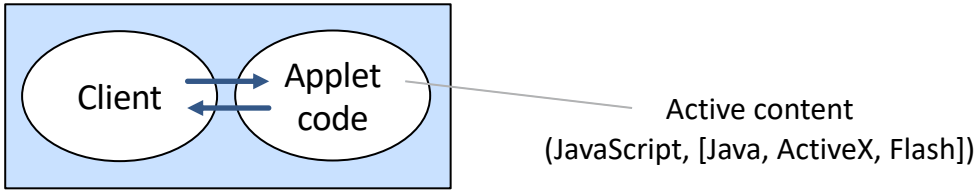


Web Applications

1. Client request results in the downloading of applet code

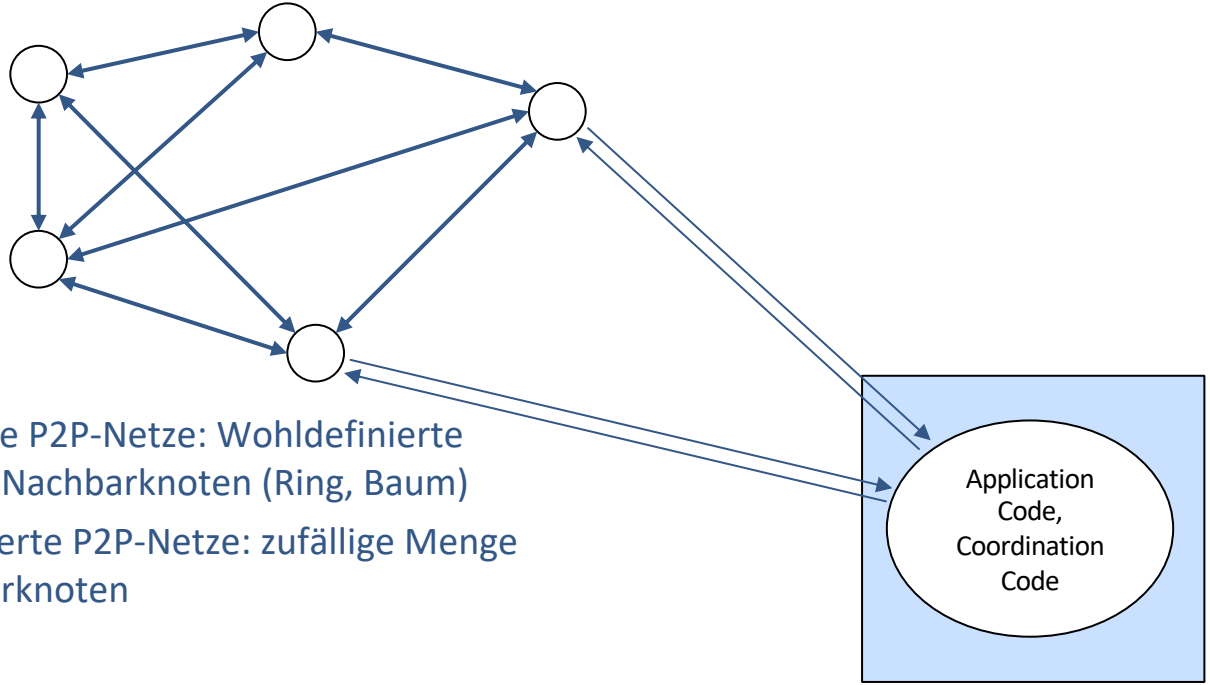


2. Client interacts with the applet



Peer-to-Peer-Architektur (P2P)

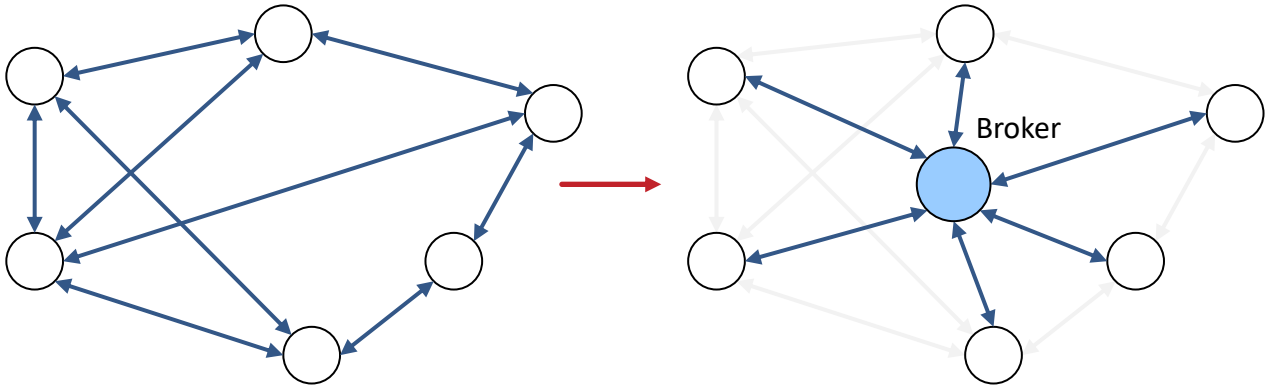
- Peer-to-Peer-Netz: Anordnung gleichartiger und gleichberechtigter Knoten, die sowohl Dienste zur Verfügung stellen als auch Dienste in Anspruch nehmen.



- Strukturierte P2P-Netze: Wohldefinierte Menge von Nachbarknoten (Ring, Baum)
- Unstrukturierte P2P-Netze: zufällige Menge von Nachbarknoten

Peer-to-Peer-Architektur (P2P)

- Peer-to-Peer-Netz: Anordnung gleichartiger und gleichberechtigter Knoten, die sowohl Dienste zur Verfügung stellen als auch Dienste in Anspruch nehmen.



- P2P-System konstituiert meist ein (logisches) Overlay von Prozessen (über ggf. beliebigen Netztopologien)

Safety (Backup, RAID, Fehlertoleranz)

Sicherheit: Abgrenzung von Security & Safety

Security Schutz gegen beabsichtigte Angriffe	Safety Schutz vor unbeabsichtigten Ereignissen
Vertraulichkeit • Abhörsicherheit • Sicherheit gegen unbefugten Gerätezugriff • Anonymität • Unbeobachtbarkeit	Fehlertoleranz **Verfügbarkeit** • Funktionssicherheit • Technische Sicherheit • Schutz vor Überspannung, Überschwemmung, Temperaturschwankungen • Schutz vor Spannungsausfall
Integrität • Übertragungsintegrität • Zurechenbarkeit • Abrechnungsintegrität	
Verfügbarkeit • Ermöglichen von Kommunikation	**Sonstige Schutzziele** • Maßnahmen gegen hohe Gesundheitsbelastung • …

Sicherheit: Informationssicherheit vs. Technische Sicherheit

Informationssicherheit

Freiheit v. Gefährd. f.d. *Daten* des IT-Systems

Vertraulichkeit

- Abhörsicherheit
- Sicherheit gegen unbefugten Gerätezugriff
- Anonymität
- Unbeobachtbarkeit

Integrität

- Übertragungsintegrität
- Zurechenbarkeit
- Abrechnungsintegrität

Verfügbarkeit

- Ermöglichen von Kommunikation

Technische Sicherheit

Freiheit v. Gefährd. f.d. *Umgebung* des IT-Systems

Verfügbarkeit

- Funktionssicherheit
- Technische Sicherheit
- Schutz vor Überspannung, Überschwemmung, Temperaturschwankungen
- Schutz vor Spannungsausfall

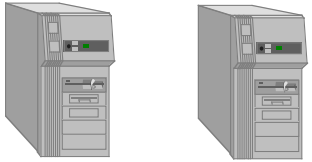
Zuverlässigkeit (reliability)

- Korrektheit

Sonstige Schutzziele

- Maßnahmen gegen hohe Gesundheitsbelastung
- …

Verfügbarkeit: Redundanz und Diversität

Redundanz	Diversität
Mehrfache Auslegung von Systemkomponenten	Verschiedenartigkeit der Herkünfte
Bei Ausfall übernimmt Ersatzkomponente	Tolerieren von systematischen Fehlern und verdeckten trojanischen Pferden
	Unabhängige Entwicklung von redundanten (Software)-Komponenten

Backup

Backup (Datensicherung)

- Bedrohungen (Beispiele)
  - Hardwaredefekte
  - Diebstahl
  - höhere Gewalt
  - versehentliches Löschen
  - Viren und andere Schadprogramme

- Auswahl der Daten

  Nicht wiederbeschaffbare Daten genießen höchste Backup-Priorität

  - selbst erstellte Dateien ++
  - Konfigurationsdateien +
  - Internet-Downloads -/+
  - Programme - (Installations-CD)
  - Betriebssystem -- (Setup-CD)

Backup und Archivierung

- Backup:

  Sicherungskopie von Daten als Schutz vor Verlust

  - einzelne Dateien und Verzeichnisse
  - ganzes Abbild (Image) eines Speichermediums (typ. Festplattenpartition), z.B. sichern inkl. Betriebssystem
  - zur Erfüllung gesetzlicher Auflagen (z.B. KonTraG?)

- Archivierung:

  Aufbewahrung von Daten in einem bestimmten Zustand

  - zur Erfüllung gesetzlicher Auflagen
  - zur eigenen Dokumentation
  - zur Versionenkontrolle

Backup-Strategien

- Voll-Backup
  - sichert alle Daten

- Inkrementelles Backup
  - sichert alle seit dem letzten Backup hinzugekommenen und veränderten Daten
  - mehrfach hintereinander durchführbar
  - alle Zwischenstände zwischen Vollbackups sind rekonstruierbar
  - Restore erfolgt in der Reihenfolge der Sicherungen
  - alle Sicherungen seit dem letzten Voll-Backup erforderlich

- Differenzielles Backup
  - sichert alle seit dem letzten Voll-Backup hinzugekommenen und veränderten Daten
  - Restore erfordert nur Voll-Backup und letztes differenzielles Backup

In welchem Format sollte gesichert werden?

- Als Rohdaten (raw)
  - überall lesbar
  - je nach Rohdaten geringe Speichereffizienz
- In verbreitetem Archivformat (z.B. zip)
  - fast überall lesbar (auch über Plattformen hinweg)
  - bei Komprimierung hohe Speichereffizienz
- Im proprietärem Format der Backup-Software
  - Abhängigkeit von Betriebssystem und Backup-Software berücksichtigen

---

- Welche Daten müssen gesichert werden?
  - Archiv-Bit vom Dateisystem (z.B. NTFS, FAT) verwaltet
    - vom Betriebssystem nach Anlegen oder Verändern einer Datei gesetzt und von Backup-Software zurückgesetzt
    - nur bei inkrementellem Backup nützlich
  - Backup-Software führt Liste der gesicherten Dateien

Auf welchen Medien sollte gesichert werden?

- Externe Standardmedien (Streamer, CD/DVD-R/W)
  - meist überall lesbar
  - empfehlenswert für Voll-Backup
  - Haltbarkeitsdauer (DVD 40...200 Jahre) beachten
  - Treiber nicht nur auf Backup sichern

- Festplatten- und Netzlaufwerke
  - sehr effizient für inkrementelle und differenzielle Sicherungen

  - Beachte: RAID (Redundant Array of Inexpensive Disks)
    - ersetzt nicht das Backup
    - schützt nicht vor versehentlichem oder bösartigem Löschen

- Spezielle Backup-Systeme

Spezielle Backup-Systeme

Bilder: http://www.ualberta.ca/CNS/vrtour/tsm/

Vorsichtsmaßnahmen beim Sichern

- Schutzziel Integrität: Sicherungskopie sollte nach dem Schreiben mit Originaldaten verglichen werden

- Schutzziel Verfügbarkeit: Mehrere Mediensätze verwenden
  - Schutz vor Fehlern während des Sicherns
    - letzten Mediensatz aufbewahren
    - vorletzten (oder besser vorvorletzten) Mediensatz überscheiben
  - Katastrophenschutz:
    - Originaldaten und Backup an verschiedenen Orten sichern
    - Schutz vor Feuer, Wasser, ...
    - Datasafe (Lampertz, Sistec)

- Schutzziel Vertraulichkeit: Wer Zugriff auf Backup hat, hat Zugriff auf Daten.
  - sichere Aufbewahrung (Datasafe)
  - Verschlüsselung (Beachte: Schlüsselbackup ggf. notwendig)

Wechselprinzip für inkrementelle Sicherungen

- 1 mal pro Zeiteinheit (z.B. Woche) Voll-Backup (z.B. Montag)
  - auf Medium 1
- n mal inkrementelles Backup (z.B. Di, Mi, Do, Fr)
  - auf Medien 2,3,...,n+1
- Anzahl der benötigten Medien:
  - 1+n+1 Medien

- nächstes Voll-Backup auf Medium n+2

Mo	Di	Mi	Do	Fr	Mo	Di	Mi	Do	Fr	Mo	Di	Mi
v1	i2	i3	i4	i5	v6	i2	i3	i4	i5	v1	i2	i3

- Medium 1 und 6: etwa gleiche (hohe) Kapazität
- Medien 2-5: geringere Kapazität als 1 und 6 nötig

Wechselprinzip f. differenzielle Sicherungen

- Anzahl der benötigten Medien:
  - 2+2 Medien
  - 1 mal pro Zeiteinheit (z.B. Woche) Voll-Backup (z.B. Montag) auf Medium 1
  - differenzielle Backups (z.B. Di, Mi, Do, Fr) abwechselnd auf Medien 2 und 3
  - nächstes Voll-Backup auf Medium 4

Mo	Di	Mi	Do	Fr	Mo	Di	Mi	Do	Fr	Mo	Di	Mi	Do	Fr
v1	d2	d3	d2	d3	v4	d2	d3	d2	d3	v1	d2	d3	...	

- Falls alle Medien ausreichend groß:
  - 3 Medien genügen
  - vorletztes differenzielles Backup für Voll-Backup

Mo	Di	Mi	Do	Fr	Mo	Di	Mi	Do	Fr	Mo	Di	Mi	Do	Fr
v1	d2	d3	d2	d3	v2	d3	d1	d3	d1	v3	d1	d2	...	

!!!....bloß nicht durcheinander kommen....!!! :-)

RAID

VSS 17.04.2020

RAID

- Akronym für
  - Redundant Array of Inexpensive Disks (1982)
  - Redundant Array of Independent Disks (Neudefinition 1992)

- Steigerung von
  - Leistung (I/O-Durchsatz, Speicherkapazität)
  - Zuverlässigkeit

- RAID-Levels

keine Redundanz	RAID-0	RAID-01, RAID-10
Datenreplikation	RAID-1	
Fehlerkorrigierende Codes	RAID-2	
Paritätsbits	RAID-3, RAID-4, RAID-5, RAID-6	

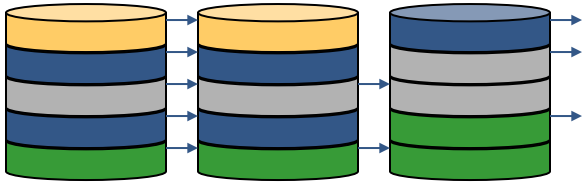
RAID: Grundprinzipien 1/3

- Problem Leistung
  - CPU-Leistung und Speicherbedarf wachsen exponentiell
  - I/O-Performance wächst dagegen deutlich langsamer

- Ausweg
  - Mehrere Disks und I/O-Kanäle gleichzeitig verwenden

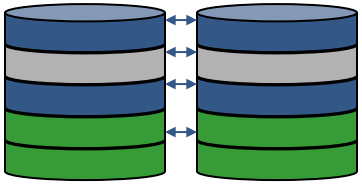
- Verteilung der Daten auf mehrere Disks (Striping, Interleaving)
  - Bit-Interleaving (bitweise Verteilung über mehrere Disks)
  - Sector-Interleaving (Sektor-weise Verteilung)

  - Paralleles Lesen und Schreiben möglich
  - Bei Verlust einer Disk --> Datenverlust



RAID: Grundprinzipien 2/3

- Problem Zuverlässigkeit
  - MTBF(disk array) = MTBF(disk) / Anzahl disks
  - Beispiel: (MTBF: Mean Time Before Failure)
    - MTBF einer 80GB-Festplatte sei >50 000 Stunden (5,7 Jahre)
    - MTBF eines Disk-Arrays mit 2 Terabyte (25Disks x 80GB) ist dann nur 83 Tage
- Ausweg
  - Einsatz geeigneter Fehlertoleranzmaßnahmen
- Spiegelung (mirroring, shadowing)
  - Doppelung der Daten auf einer zweiten Disk
  - Teuer, aber sehr zuverlässig



- Kompromiss zwischen Zuverlässigkeit und Kosten, je nach RAID-Level auch verwendet:
  - Paritätsbits
  - Fehlerkorrigierende Codes

RAID: Grundprinzipien 3/3

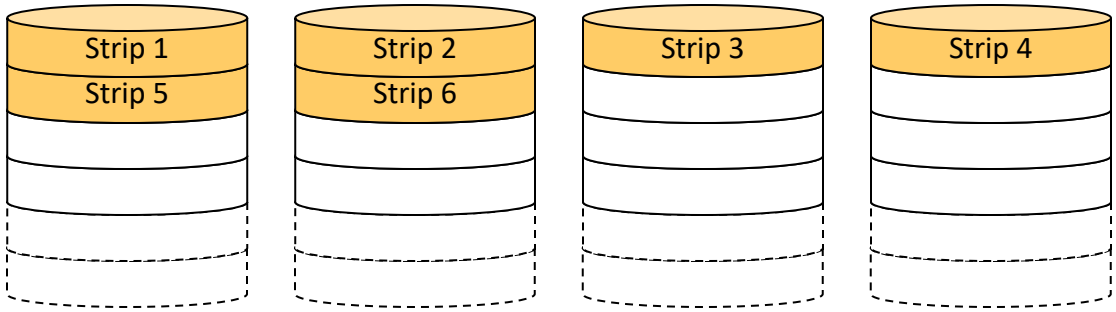
- Problem Kosten
  - Einsatz standardisierter, kostengünstiger Komponenten
    - Festplattenlaufwerke (Massenmarkt)
    - Standard-Datenbusse (z.B. SCSI, Firewire, Fibre Channel)

- Teure Alternative:
  - Spezial-Hardware, z.B. Festplattenfarm im Großrechnerbereich

- Einsatzziel von RAID-Technik
  - kostengünstige Erhöhung der Datentransferrate und Ausfallsicherheit von Festplatten

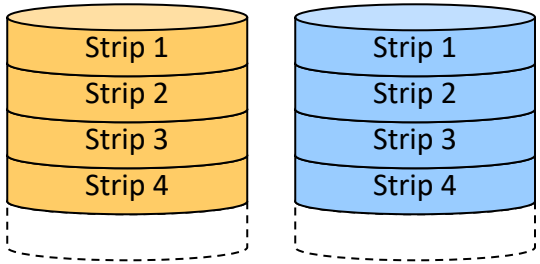
RAID-0

- Merkmale
  - Aufteilung der Daten in Blöcke, paralleles Schreiben auf vorhandene Festplatten (disk striping)
  - keine Redundanz, geringe MTBF
  - Sehr hohe Datentransferrate (für kleine Strips)
  - Sehr geringe I/O-Request-Verarbeitungszeit (für große Strips)
- Verwendungszweck: Anwendungen mit
  - hohen I/O-Performance-Anforderungen
  - geringen Anforderungen an die Verfügbarkeit



RAID-1

- Merkmale
  - gespiegelte Platten (mirroring, shadowing), gleiche Daten werden gleichzeitig auf unterschiedliche Platten geschrieben
  - bei Datenverlust steht Duplikat der Daten zur Verfügung
  - hohe Kosten
  - Datentransferrate, I/O-Request-Verarbeitungszeit:
    - Lesen: gut, Platte mit geringerer Zugriffszeit (seek distance)
    - Schreiben: mittel, beide Platten müssen schreiben
- Verwendungszweck: Anwendungen mit
  - sehr hohen Anforderungen an die Verfügbarkeit



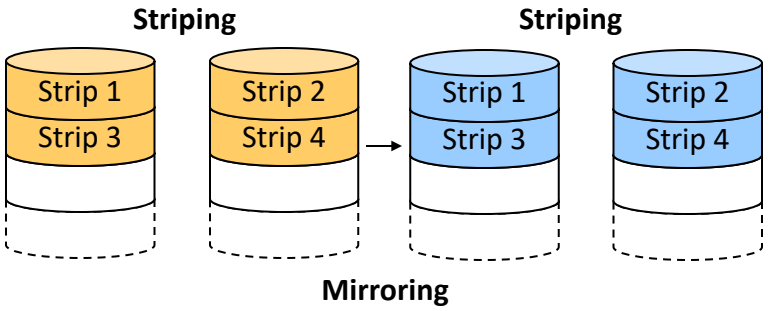
1 Logischer Schreibzugriff = 2 physische Schreibzugriffe

Kombination von RAID-0 und RAID-1

gespiegelte Verteilung

Mirrored Strips

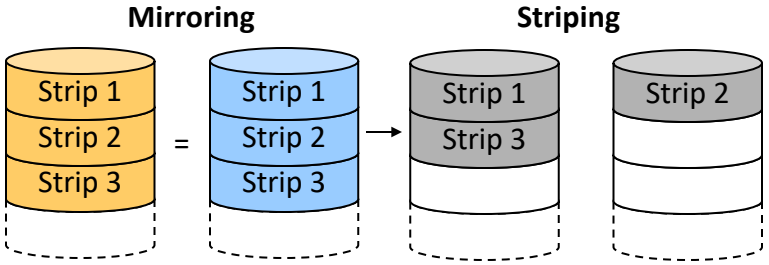
RAID 0+1 oder RAID 01



verteilte Spiegelung

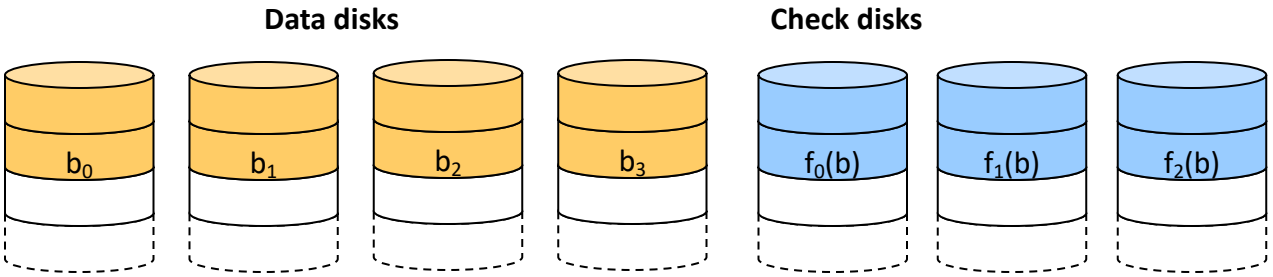
Striped Mirrors

RAID 1+0 oder RAID 10



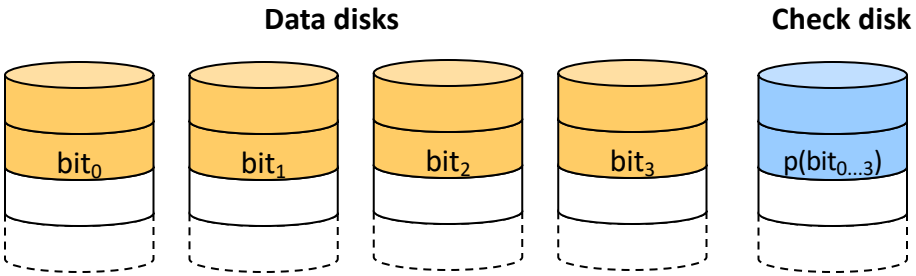
RAID-2

- Merkmale
  - Verteilung (striping)
  - fehlerkorrigierender Hamming-Code (20-40 Prozent Overhead)
  - Datentransferrate
    - hervorragend für große Datensätze, aber gering für kleine Datensätze, da immer gesamter Strip gelesen werden muss
  - I/O-Request-Verarbeitungszeit
    - gering: keine parallelen I/O-Requests verarbeitbar, da alle Laufwerke mit einem Request befasst sind
  - geringe praktische Bedeutung



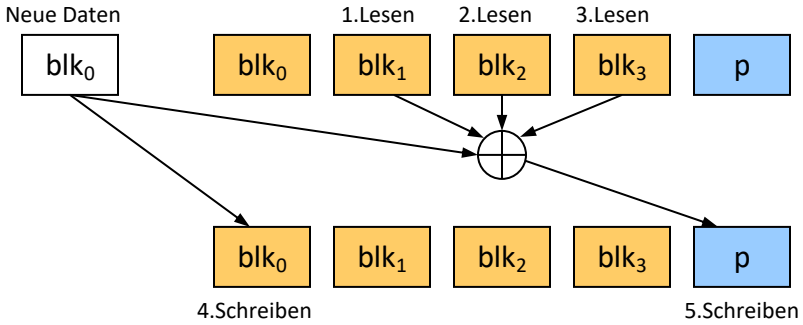
RAID-3

- Merkmale
  - Verteilung (striping): bit interleaving
  - Paritätsinformation auf zusätzlicher Platte
  - bei Ausfall genau einer Platte Rekonstruktion der Daten möglich
  - geringfügig zuverlässiger als RAID-2, da weniger Platten
  - Performance: geringfügig besser als RAID-2, wg. bitweiser Verarbeitung, aber immer noch schlecht
- Verwendungszweck:
  - Anwendungen mit wenigen großen Dateien (z.B. Bildverarb.)
  - nicht geeignet für Transaktionssysteme und Direktzugriff

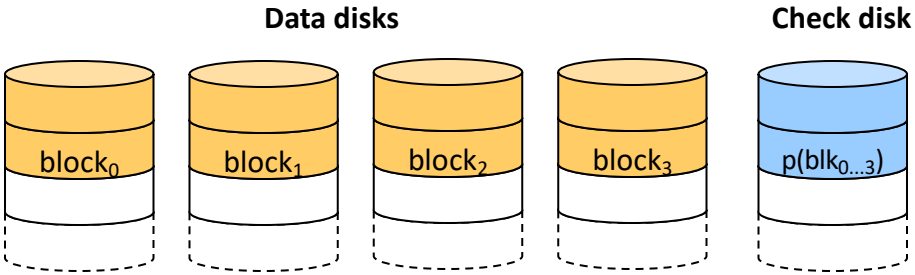


RAID-4

- Merkmale
  - Verteilung (striping): sector interleaving
  - Paritätsinformation auf zusätzlicher Platte

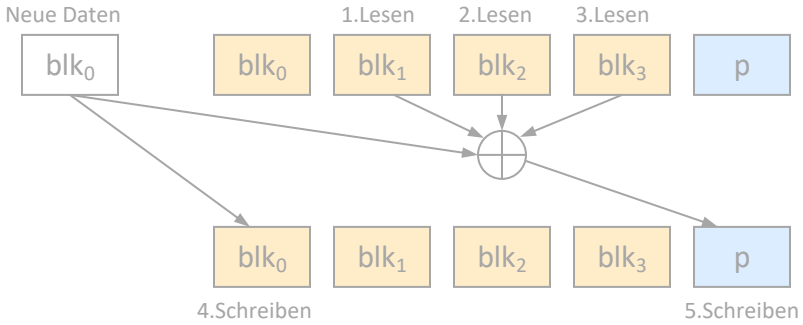


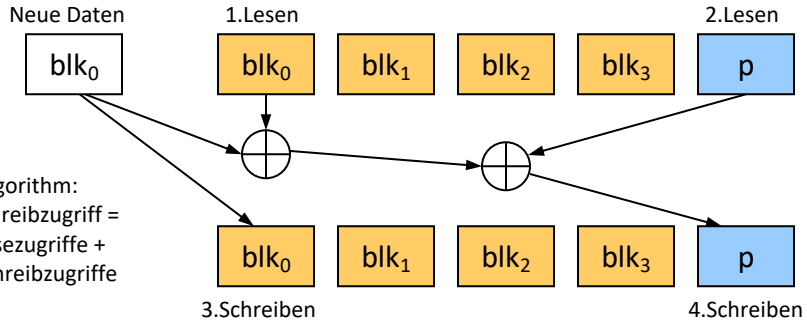
Ineffiziente Berechnung der Parität



RAID-4

- Merkmale
  - Verteilung (striping): sector interleaving
  - Paritätsinformation auf zusätzlicher Platte



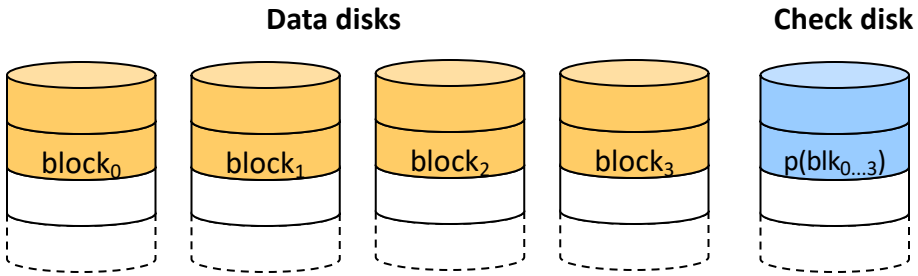


Small-Write Algorithm:
1 Logischer Schreibzugriff =
2 physische Lesezugriffe +
2 physische Schreibzugriffe

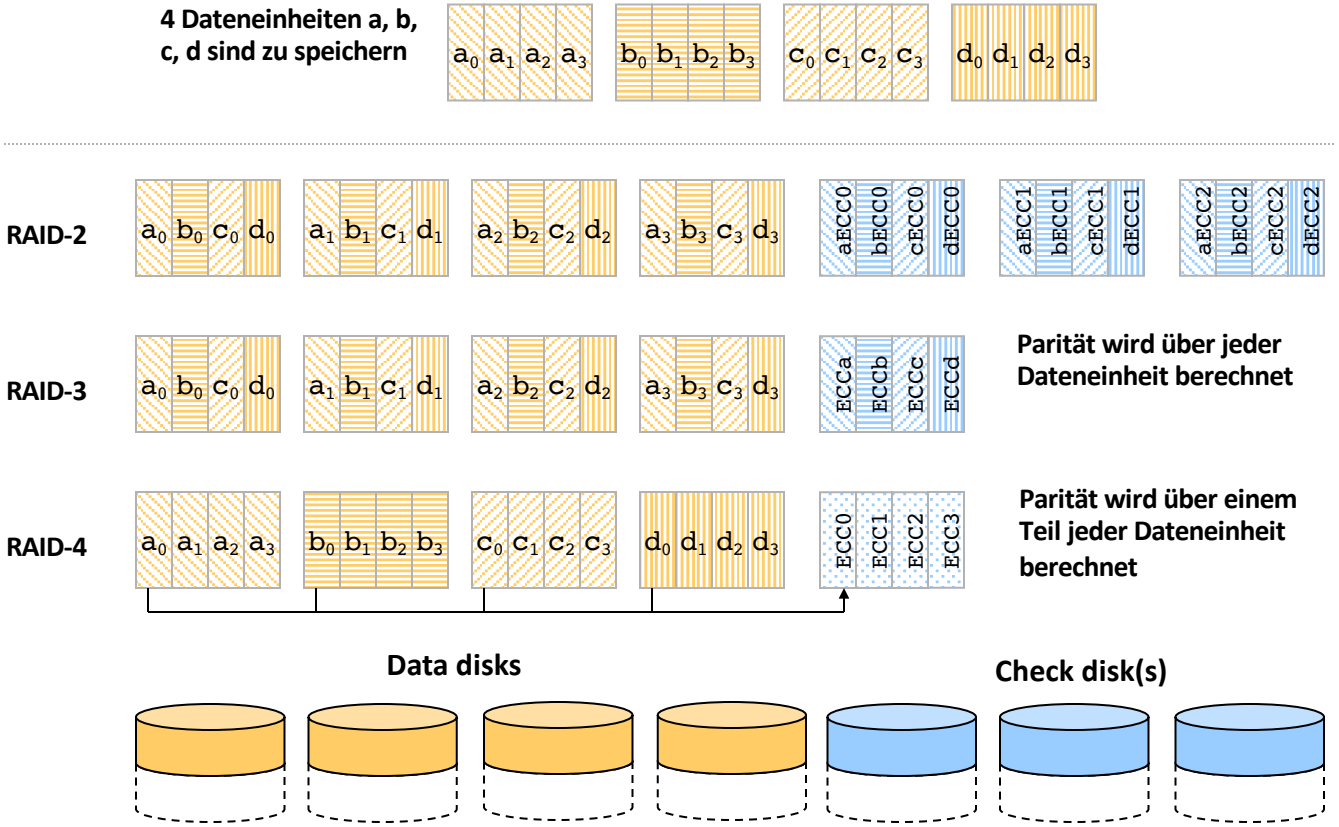
Besser: Neue Parität wird aus der Differenz des alten und neuen Datenblocks berechnet

RAID-4

- Merkmale
  - Verteilung (striping): sector interleaving
  - Paritätsinformation auf zusätzlicher Platte
  - große Lese- und Schreibzugriffe parallel möglich
  - Datendurchsatz: etwa wie RAID-3
  - I/O-Request-Verarbeitungszeit: sehr gut für Lesen, etwas besser als RAID-3 für Schreiben
  - Paritätsplatte bleibt Flaschenhals
- Anwendungen wie RAID-3

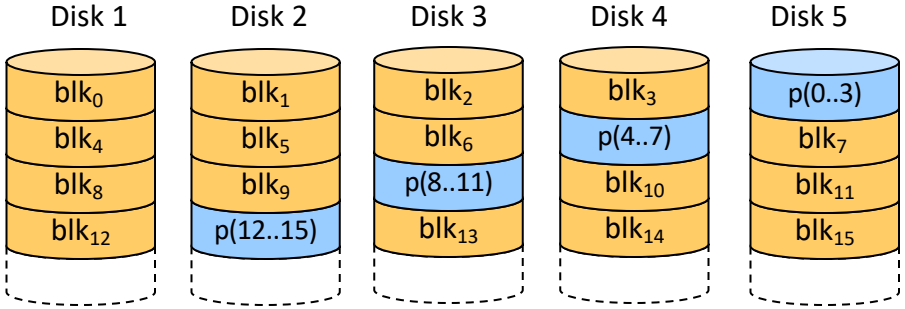


Vergleich der RAID-Leves 2, 3 und 4



RAID-5

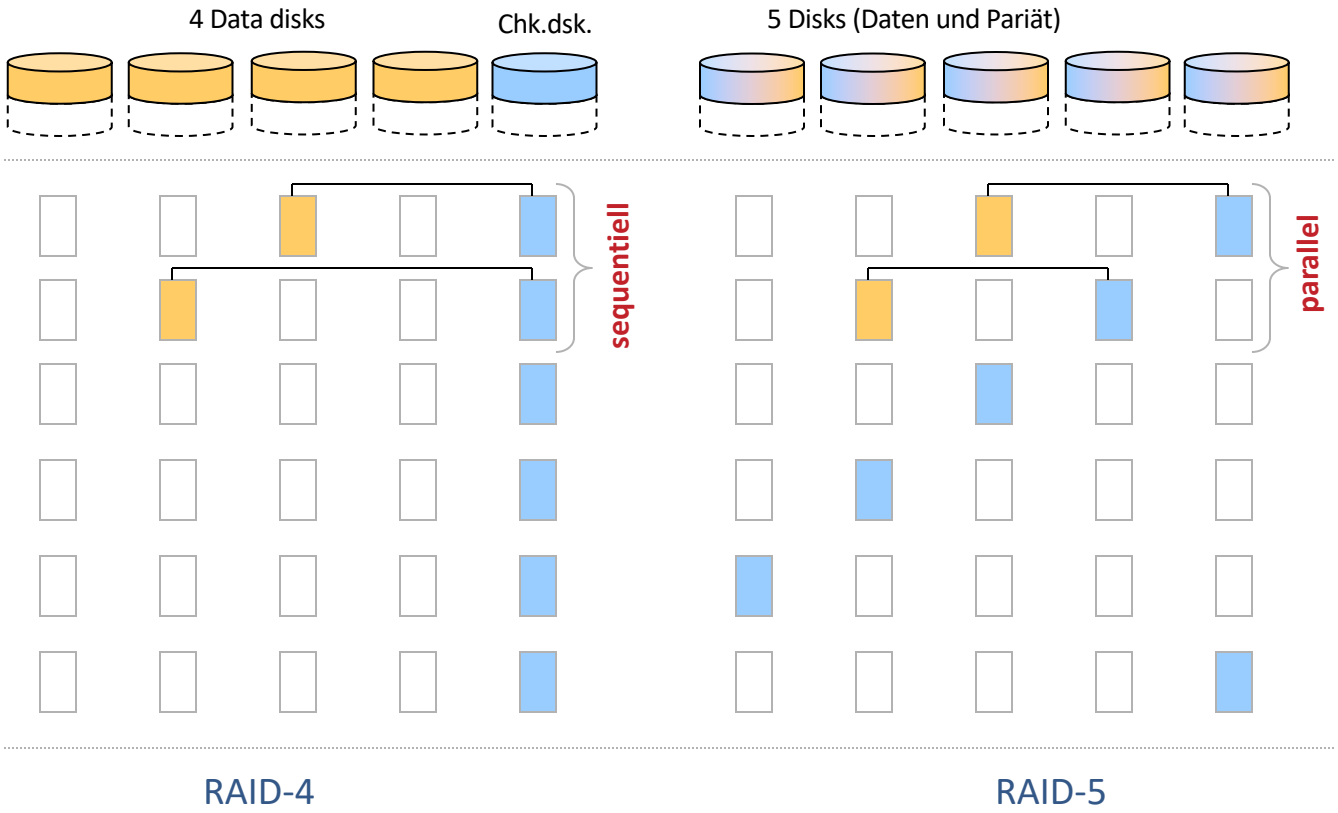
- Merkmale
  - Verteilung (striping) der Daten (sector interleaving) und Paritätsinformation zur Verbesserung der Performance
  - Ziel: Verteilen der Last auf Paritätsplatte, Schreibzugriffe können simultan erfolgen
  - Datendurchsatz: gut für Lesen und Schreiben
  - I/O-Request-Verarbeitungszeit: sehr gut für Lesen und Schreiben
- Verwendungszweck:
  - Transaktionsorientierte Anwendungen, Datenbanken



Write(blk0) benötigt Disk 1 und 5

Write(blk6) benötigt Disk 3 und 4

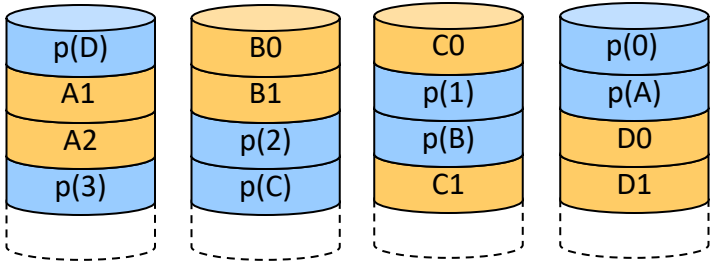
Vergleich von RAID-4 und RAID-5 (Schreiben von Daten)

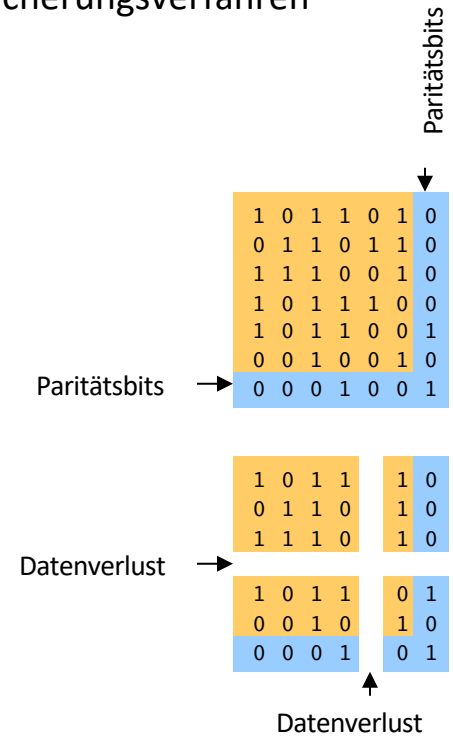


RAID-6

- Merkmale
  - Verwendung zweidimensionaler Parität ähnlich Kreuzsicherungsverfahren
  - kann zwei gleichzeitige Plattencrashes tolerieren
  - recht aufwändige Elektronik
  - Schreiben ist langsam

- Schematisch





Fehlertoleranz

Arten von Fehlern, (Nicht)-Reparierbare Systeme

Redundanztechniken und Aspekte der Bewertung

Anwendungsbeispiele für fehlertolerante Systeme

Fehlertoleranz durch redundante Systeme

- Ziel
  - Verbesserung der Zuverlässigkeit (Ausfallsicherheit) von Systemkomponenten
- Typische Kenngrößen
  - Mean Time To Failure (MTTF)
    - Erwartungswert für die Zeit (von der Inbetriebnahme) bis zum (ersten) Ausfall eines Systems
  - Mean Time To Repair (MTTR)
    - Erwartungswert für die Zeit zur Ersetzung des Systems und ggf. der Rekonstruktion der Daten
- ferner
  - Mean Time To Data Loss (MTTDL)
    - Erwartungswert für die Zeit bis zu einem nicht-maskierbaren Fehler (hauptsächlich verwendet in Speicherkomponenten)

Arten von Fehlern

Hardware-Fehler
- betreffen *physische* Teile des Systems

Software-Fehler
- betreffen *logische* Teile des Systems

Transiente Fehler
- vorübergehende Fehler, evtl. nicht reproduzierbar, z.B. Übertragungsfehler
- können behoben werden durch wiederholte Ausführung der fehlerhaften Operation

Permanente Fehler
- dauerhaft
- leicht erkennbar

Einzel-Fehler
- betreffen einzelne Komponenten
- unabhängig von der Funktion anderer Systemteile

Mehrfach-Fehler
- betreffen mehrere Komponenten des Systems gleichzeitig
- durch *Fehlerfortpflanzung* oder gemeinsame Fehlerquellen hervorgerufen

Nicht-reparierbare Systeme

- Zuverlässigkeit (reliability): $R(t)$
  - bedingte Wahrscheinlichkeit, dass ein System das Zeitintervall $[0,t]$ überlebt, wenn es in t=0 funktionstüchtig war
- Es gilt

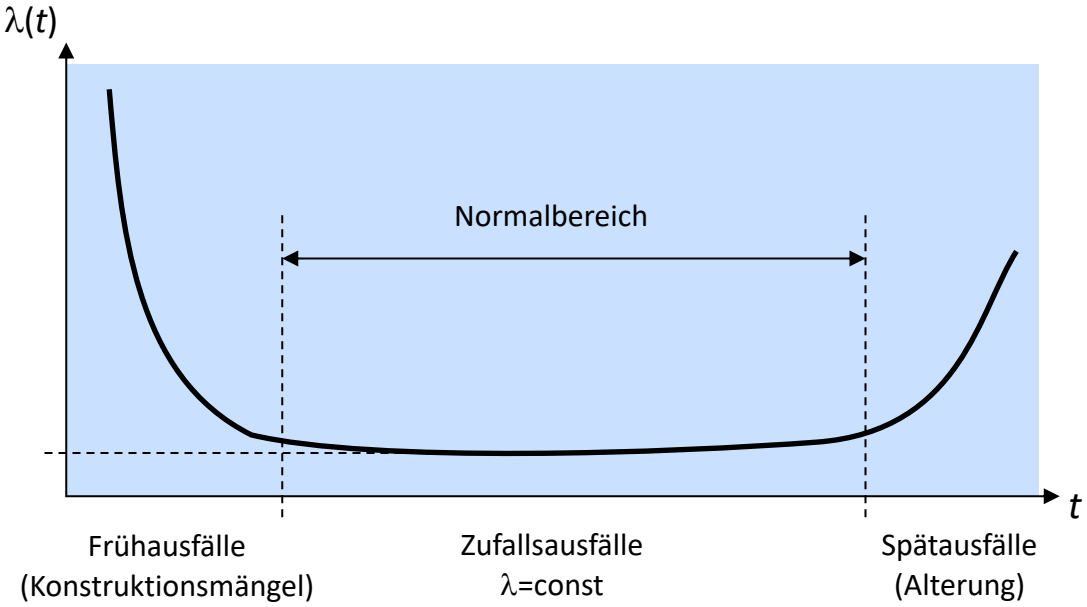
$$\lim_{t \to \infty} R(t) = 0$$

- Ausfallrate: $\lambda(t)$
  - bedingte Wahrscheinlichkeit, dass das System im differentiellen Zeitintervall $[t+dt]$ ausfällt
- Typischerweise gilt im »Normalbereich« der Nutzung eines Systems (z.B. Halbleiter-Bausteine)

$$\lambda(t)=\text{const}$$

»Badewannenkurve«

- gilt für Halbleiter,
- nicht jedoch für bewegliche Teile (z.B. Lüfter, Festplatten etc.)



Nicht-reparierbare Systeme

- Mit $\lambda = \lambda(t) = \text{const}$ gilt

$$R(t) = e^{-\lambda t} \quad \text{(Exponentialverteilung)}$$

- Mittlere Zeit bis zu einem Fehler (MTTF)

$$MTTF = \int_{t=0}^{\infty} R(t)\, dt = \frac{1}{\lambda}$$

Reparierbare Systeme

- Verfügbarkeit (availability): $A(t)$
  - Wahrscheinlichkeit, dass ein System zu einem vorgegebenen Zeitpunkt t bzw. in einem Zeitraum $[0,t]$ in einem funktionsfähigen Zustand angetroffen wird

- Für Nicht-reparierbare Systeme gilt:

  $A(t) = R(t)$

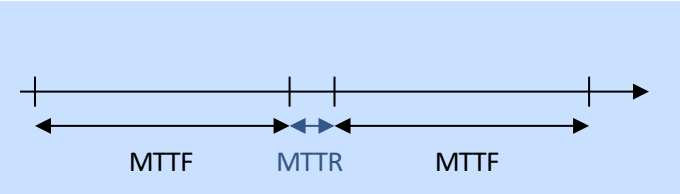
- Für reparierbare Systeme Ermittlung der stationären Verfügbarkeit

$$A = \lim_{t \to \infty} A(t)$$

Reparierbare Systeme

- Stationäre Verfügbarkeit

$$A = \frac{\mu}{\lambda + \mu} = \frac{MTTF}{MTTF + MTTR}$$



- Mit

$\lambda = \lambda(t) = \text{const}$ (Ausfallrate)

$$\lambda = \frac{1}{MTTF}$$

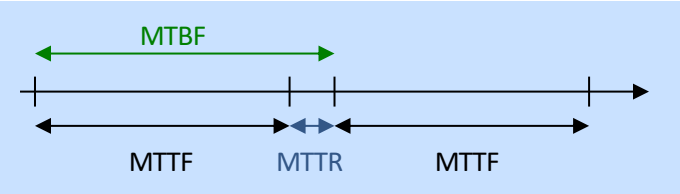
MTTF: Mittlere Zeit bis zum Ausfall

$\mu = \mu(t) = \text{const}$ (Reparaturrate)

$$\mu = \frac{1}{MTTR}$$

MTTR: Mittlere Reparaturzeit

Reparierbare Systeme



- Bei reparierbaren Systemen findet man auch häufig die Kenngröße
  - Mean Time Between Failure (MTBF)
    - Erwartungswert für die Zeit zwischen zwei Ausfällen

vgl. Lexikon Informatik und Kommunikationstechnik, VDI Verlag, 1990, S. 419

  - Achtung!
    - Meist wird definiert: MTBF = MTTF + MTTR
    - Manchmal auch: MTBF ≡ MTTF (umgangssprachlich)
      - Beispiel Festplatten: Wenn Platte defekt, wird sie üblicherweise ausgetauscht
      - MTBF als Mean Time Before Failure verstanden

- Fehlerrate

$$\frac{\text{nicht erkannte fehlerhaft durchgeführte Transaktionen}}{\text{Gesamtzahl durchgeführter Transaktionen}}$$

- irrelevant:
  - Transaktionen, die aufgrund eines Fehlers abgebrochen wurden
- Zusammenhang zur Verfügbarkeit
  - Maßnahmen zur Reduzierung der Fehlerrate können zu Verschlechterung der Verfügbarkeit führen

Redundanztechniken

- Zeitredundanz
  → *Berechnungen werden mehrfach durchgeführt*
  – transiente Hardware-Fehler erkennbar
  – keine Erkennbarkeit von Konstruktionsmängeln und permanenten Hardwarefehlern

- Informationsredundanz
  → *Daten werden mehrfach gespeichert oder übertragen oder Prüfinformation gebildet*
  – transiente Fehler werden toleriert

Redundanztechniken

- Funktionelle Redundanz

  → *Hinzufügen speziell entworfener Komponenten*

  – Tolerierung von Konstruktionsmängeln

  – Sebsttest-Komponente, Plausibilitätsprüfungen

  – N-Version-Programming: Schutzziel Diversität

    - N (mind. 3) verschiedene Versionen eines Programms, Mehrheitslogik vergleicht Ergebnisse zur Laufzeit (Triple Modular Redundancy, TMR)

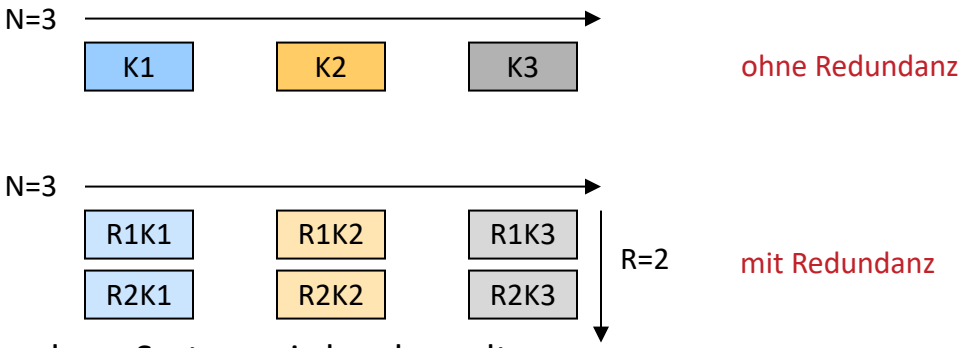
      – sehr gute Verfügbarkeit, sehr hohe Integrität

    - N (mind. 2) Versionen eines Programms werden von konkurrierenden Teams entwickelt und gegenseitig untersucht, *ein* optimiertes Programm auf Zielsystem

      – sehr gute Verfügbarkeit, sehr gute Vertraulichkeit (Schutz vor verdeckten Kanälen und Trojanischen Pferden)

Redundanztechniken

- Strukturelle Redundanz
  - → *System oder Teile davon werden mehrfach ausgelegt*
  - – Tolerierung von permanenten Hardwarefehlern
  - → *Unterscheidungen*
    - RN-Redundanz: Jede der N Komponenten wird ver-R-facht.
      - – R-1 Komponenten stehen jeder der N Komponenten *exklusiv* zur Verfügung



      - – Meist 2N-Redundanz: System wird gedoppelt

Redundanztechniken

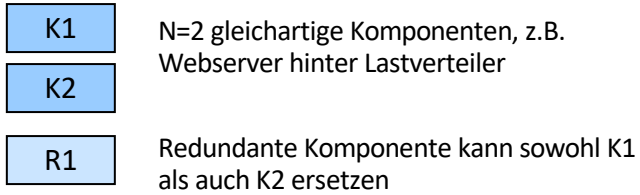
- Strukturelle Redundanz

  → *System oder Teile davon werden mehrfach ausgelegt*

  – Tolerierung von permanenten Hardwarefehlern

  → *Unterscheidungen*

  - R+N-Redundanz: Dem System
    - bestehend aus N gleichartigen Komponenten
    - werden R redundante Komponenten hinzugefügt.
    - Jede der R Komponenten kann eine *beliebige* der N Komponenten ersetzen.
    - Bsp: 1+2-Redundanz



    - Mit R=1 auch als 1:N-Redundanz bezeichnet

      1:1-Redundanz entspricht Doppelung

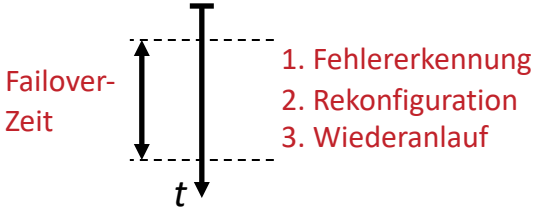
Redundanztechniken

- Statische und dynamische Redundanz

  Alle Komponenten sind ständig in Betrieb.

  In Kombination mit funktioneller Redundanz: Triple-Modular-Redundancy (TMR) oder allgemein N-Modular-Redundancy (NMR): Hardware-Komponenten werden 3- bzw. N-fach ausgeführt, Mehrheitslogik ermittelt Ergebnis

  Redundante Komponente wird erst bei Ausfall in Betrieb genommen (Standby-Verfahren).



Sicherungspunkt-Erzeugung (checkpointing): Festlegen eines Wiederaufsetzpunktes im laufenden Betrieb

- erzeugt meist niedrigere Verfügbarkeit als statische R.
- redundante Komponenten können im fehlerfreien Betrieb für andere Aufgaben verwendet werden
- höhere MTTF, da redundante Komponenten sich nicht abnutzen

Aspekte der Bewertung eines fehlertoleranten Systems

- Überdeckungsgrad (coverage factor)

$$\frac{\text{Anzahl der durch FT-Maßnahmen tolerierbaren Fehler}}{\text{Anzahl der überhaupt spezifizierten Fehler}}$$

- Schutz vor Mehrfachfehlern
  - Ausfall der redundanten Komponente (z.B. weil sie gleichen Fehler ebenfalls enthält)
  - Schutz vor gegenseitiger Fehlerfortpflanzung (Diversität)
    - z.B. geograph. getrennte Aufstellung, Hard- und Software verschiedener Hersteller

- Gesamtverfügbarkeit richtet sich nach schwächster Stelle
  - Produkt der Einzelverfügbarkeiten
  - z.B. TMR: Vergleichslogik nur einmal vorhanden
    - single-point-of-failure

      →selbst hochverfügbar auslegen

Anwendungsbeispiele für fehlertolerante Systeme

- Bereiche
  - Raumfahrt, Medizintechnik, Luftfahrt
  - Kritische Infrastrukturen
  - Signalsysteme (fail-stop)
  - Systeme mit erheblicher wirtschaftlicher Bedeutung

- Mit Schwerpunkt auf Integrität
  - Datenbanksysteme von Banken, Versicherungen etc.

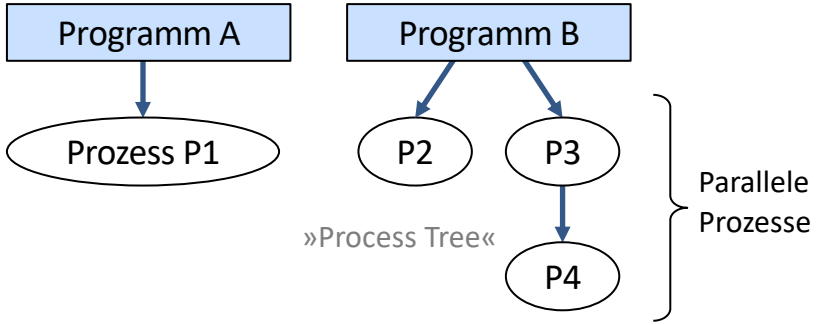
- Mit Schwerpunkt auf Verfügbarkeit
  - Telefonvermittlungsanlagen
  - Buchungssysteme
  - lebenserhaltende Systeme
  - eingebettete Systeme (z.B. im Auto)

Prozesse, Synchronisation und Koordination

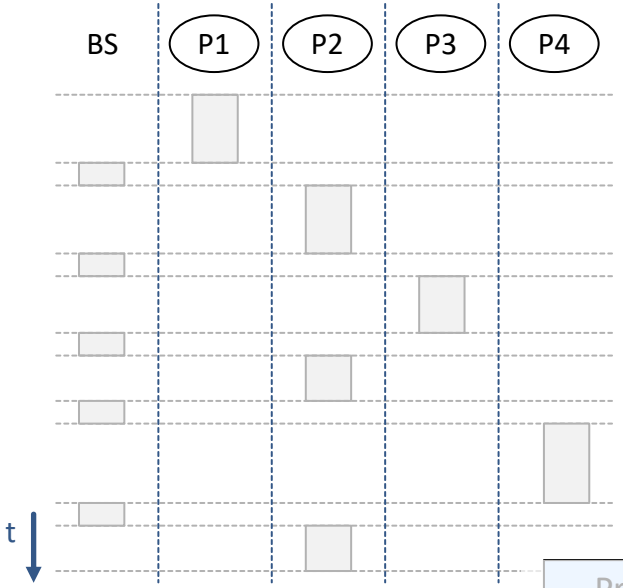
Programme vs. Prozesse

Schneider, 2012

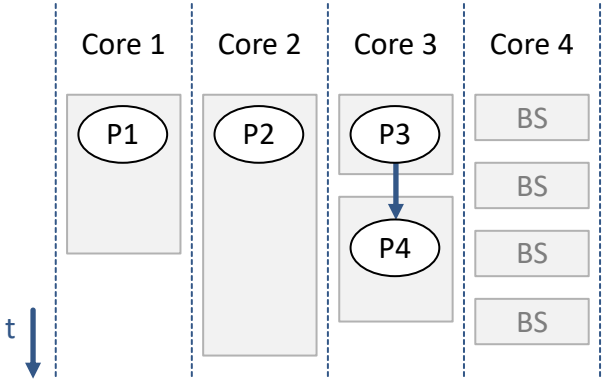
- Programm: statische Folge von Anweisungen in einer Programmiersprache unter Nutzung von Daten
- Prozess (task): dynamische Folge von Aktionen, die durch Ausführung eines Programms auf einem Prozessor zustande kommt
  - Prozess als zeitliche Folge von Zustandsänderungen
  - Adressraum: alle dem Prozess zugeordneten Speicherbereiche
  - Parallele Prozesse: alle Prozesse zwischen Erzeugung und Abschluss
    - 1-Prozessor-Systeme: Nebenläufigkeit (quasiparallel, concurrency)
    - Mehrprozessor-Systeme: echt gleichzeitige Ausführung
  - Scheduler erledigt Zuteilung von Prozessen zu Prozessor(en)

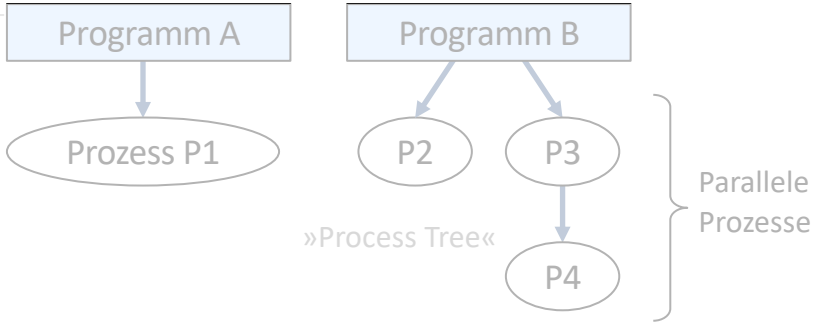


Quasiparallele Ausführung auf 1-Prozessor-System



Echt gleichzeitige Ausführung z.B. auf 4-Kern-Prozessor

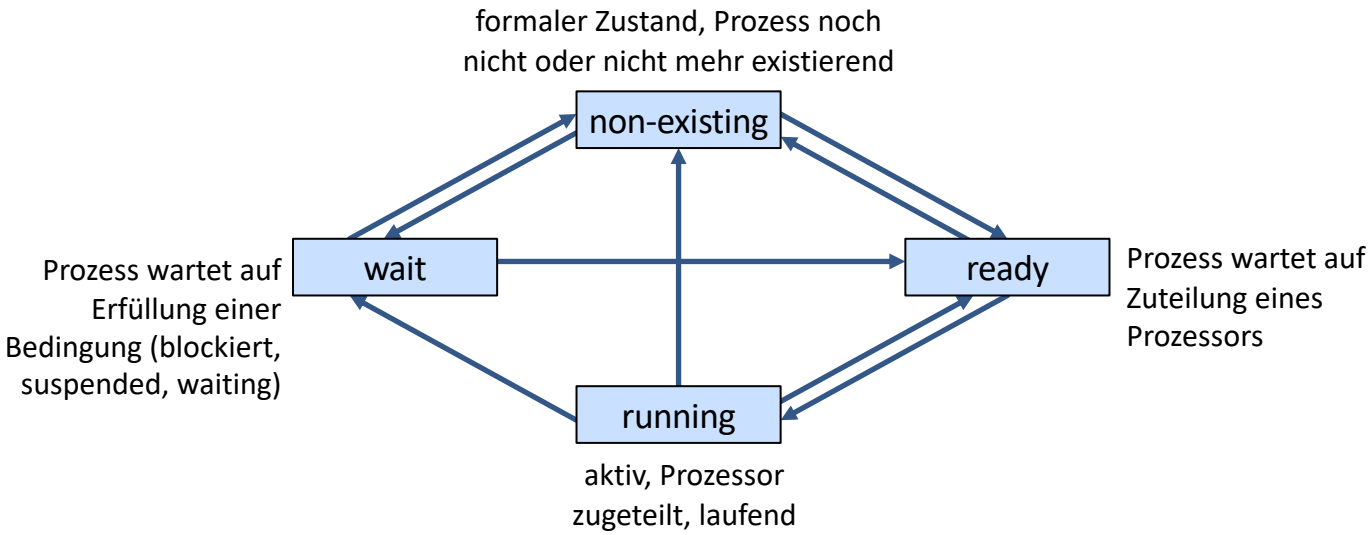




Programme vs. Prozesse

Schneider, 2012

- Programm: statische Folge von Anweisungen in einer Programmiersprache unter Nutzung von Daten
- Prozess (task): dynamische Folge von Aktionen, die durch Ausführung eines Programms auf einem Prozessor zustande kommt
  - Prozess als zeitliche Folge von Zustandsänderungen

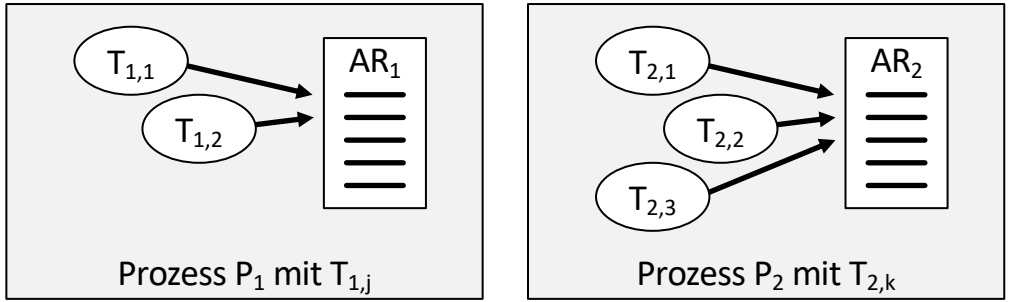


Thread: Aktivitätsträger (sequenzieller Ausführungsfaden) mit minimalem Kontext (Stack und Register) innerhalb einer Ausführungsumgebung (Prozess)

- Prozesse vs. Threads

  »schwergewichtig« »leichtgewichtig«

  Alle Threads $T_{i,j}$ desselben Prozesses P_i nutzen Adressraum AR_i und ggf. weitere Betriebsmittel gemeinsam.



- Mechanismen zur Synchronisation und Koordination von Threads entsprechen denen von Prozessen

Transaktion

Transaktion: Folge von Aktionen (Anweisungen), die (u.a.) ununterbrechbar ausgeführt werden soll

- Transaktionsverwaltung sorgt dafür, dass unvollständige Transaktionen zurückgenommen werden

- Befehle der Transaktionsverwaltung
  - BOT (begin of transaction)
  - EOT (end of transaction, commit)
  - ROB (rollback, abort) — Abbruch und Rücksetzen auf Zustand vor Transaktion

Eigenschaften von Transaktionen (ACID-Paradigma)

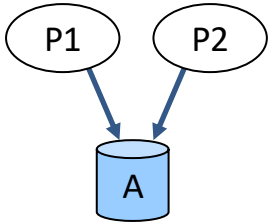
Transaktion: Folge von Aktionen (Anweisungen), die (u.a.) ununterbrechbar ausgeführt werden soll

- Atomicity (Atomarität)
  - Eine Transaktion wird als unteilbare Einheit behandelt (»alles-oder-nichts«).
- Consistency (Konsistenz)
  - Eine Transaktion hinterlässt nach (erfolgreicher oder erfolgloser) Beendigung eine konsistente Datenbasis.
- Isolation
  - Nebenläufig ausgeführte Transaktionen beeinflussen sich nicht gegenseitig.
- Durability (Dauerhaftigkeit)
  - Eine erfolgreich abgeschlossene Transaktion hat dauerhafte Wirkung auf die Datenbasis, auch bei Hardware- und Softwarefehlern (nach EOT).

Nebenläufige Transaktionsausführungen sind serialisierbar, gdw. ihr Ergebnis dem irgendeiner (!) seriellen Ausführungsreihenfolge entspricht.

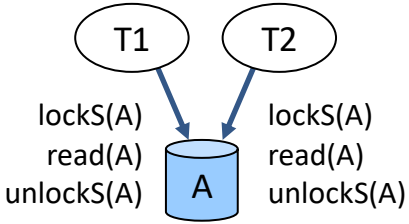
- Beispiel: read und write von Prozessen P1 und P2 auf dasselbe Datum A:

P1:read(A)	P2:read(A)	Kein Konflikt, Reihenfolge irrelevant
P1:read(A)	P2:write(A)	Konflikt! Reihenfolge bedeutungsvoll
P1:write(A)	P2:read(A)	Konflikt! Reihenfolge bedeutungsvoll
P1:write(A)	P2:write(A)	Konflikt! Reihenfolge bedeutungsvoll

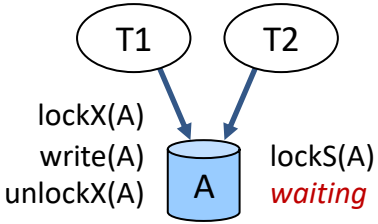


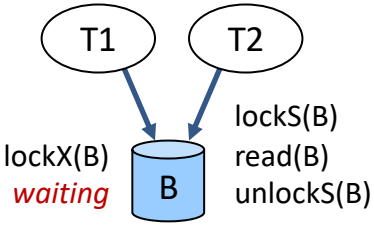
Synchronisation durch Sperren

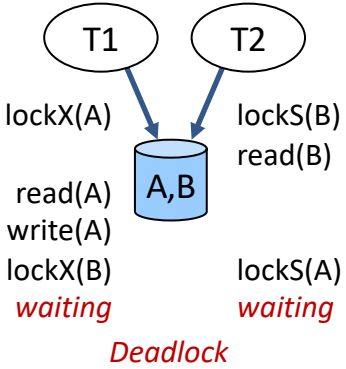
- Sperrmodus S („shared“, read lock, lockS, Lesesperre)
  - Wenn Transaktion Ti eine S-Sperre für ein Datum A besitzt, kann Ti read(A) ausführen. Mehrere Transaktionen können gleichzeitig eine S-Sperre für dasselbe Objekt A besitzen.



- Sperrmodus X („exclusive“, write lock, lockX, Schreibsperre)
  - Nur eine einzige Transaktion, die eine X-Sperre für A besitzt, darf write(A) ausführen.





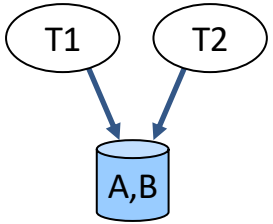


Synchronisation durch Sperren

Beispiel:

T1: Modifikation von A und B (z.B. Umbuchung auf ein Konto)

T2: Lesen von B und A (z.B. Addieren der Salden der Konten)



T1	T2	
BOT		
lockX(A)		
read(A)		
write(A)		
	BOT	
	lockS(A)	T2 muss auf T1 warten
lockX(B)		
read(B)		
unlockX(A)		T2 wecken
	read(A)	
	lockS(B)	T2 muss auf T1 warten
write(B)		
unlockX(B)		T2 wecken
	read(B)	
EOT		
	unlockS(A)	
	unlockS(B)	
	EOT	

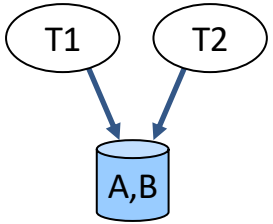
t

Synchronisation: Deadlock

Beispiel:

T1: Modifikation von A und B (z.B. Umbuchung auf ein Konto)

T2: Lesen von B und A (z.B. Addieren der Salden der Konten)



T1	T2
BOT	
lockX(A)	
	BOT
	lockS(B)
	read(B)
read(A)	
write(A)	
lockX(B)	
	lockS(A)

t

Transaktionsablauf leicht modifiziert, aber mit schwerwiegender Folge …

T1 muss auf T2 warten
T2 muss auf T1 warten
Deadlock!

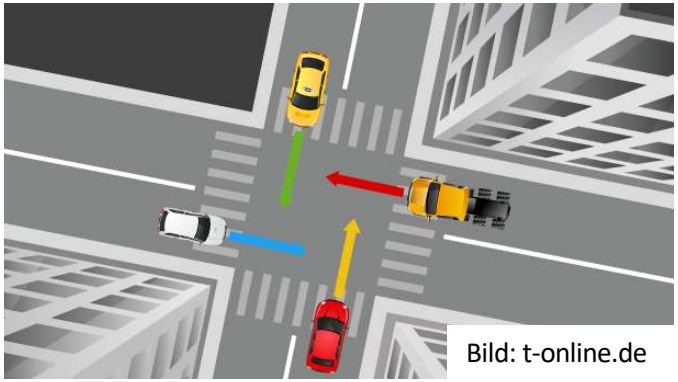


Bild: t-online.de

Synchronisation: Deadlock

Beispiel:

T1: Modifikation von A und B (z.B. Umbuchung auf ein Konto)

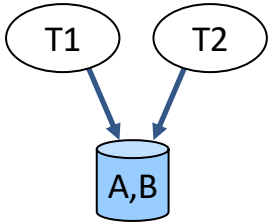
T2: Lesen von B und A (z.B. Addieren der Salden der Konten)

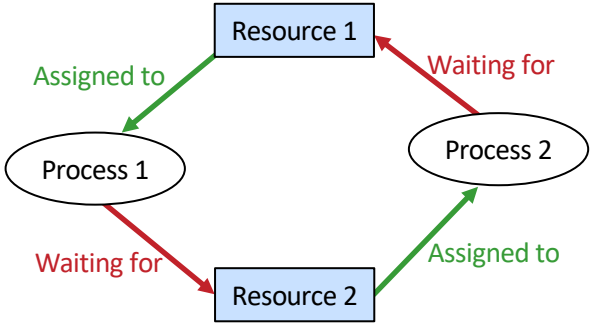
T1	T2
BOT	
lockX(A)	
	BOT
	lockS(B)
	read(B)
read(A)	
write(A)	
lockX(B)	
	lockS(A)

t

Transaktionsablauf leicht modifiziert, aber mit schwerwiegender Folge …

T1 muss auf T2 warten
T2 muss auf T1 warten
Deadlock!





Vermeiden von Deadlocks

- Verfahren 1: Preclaiming: Vorab-Anforderung aller Sperren
  - Beginn einer Transaktion erst nachdem die für diese Transaktion insgesamt erforderlichen Sperren erfolgt sind
    - »2-Phasen-Sperrprotokoll« (2-phase-locking, 2PL)
    - 2PL gewährleistet Serialisierbarkeit von Transaktionen

- 2PL: Jede Transaktion durchläuft 2 Phasen
  - Wachstumsphase: Sperren werden angefordert, aber nicht freigegeben
  - Schrumpfungsphase: Sperren freigegeben, aber nicht mehr angefordert

Vermeiden von Deadlocks

- Preclaiming mittels 2PL:

Für jede individuelle Transaktion muss gelten:

1. Jedes von einer Transaktion betroffene Objekt muss vor Beginn des Zugriffs von der Transaktion entsprechend gesperrt werden.
2. Eine Transaktion fordert eine Sperre, die sie schon einmal besessen hat, niemals wieder erneut an.
3. Eine Transaktion muss bei jedem Zugriff so lange warten, bis sie alle erforderlichen Sperren erhalten kann.
4. Spätestens wenn die erste Sperre frei gegeben wurden, darf keine neue mehr angefordert werden
5. Spätestens bei Transaktionsende (EOT) muss eine Transaktion alle ihre Sperren zurück geben.

Vermeiden von Deadlocks

- Verfahren 2: Zeitstempel-Verfahren:
  - Transaktionen werden durch Zeitstempel (z.B. Zeit des BOT) priorisiert

- Beispiel:
  - T1 hält eine exklusive (!) Sperre auf A.
  - T2 fordert ebenfalls Sperre auf A.

- Allgemeine Strategien für parallele Transaktionen, die auf dasselbe Datum A zugreifen wollen:
  - »wound-wait«
    - Abbruch (rollback) von T1, wenn T1 jünger ist als T2, sonst wartet T2
  - »wait-die«
    - Abbruch (rollback) von T2, wenn T2 jünger ist als T1, sonst wartet T2

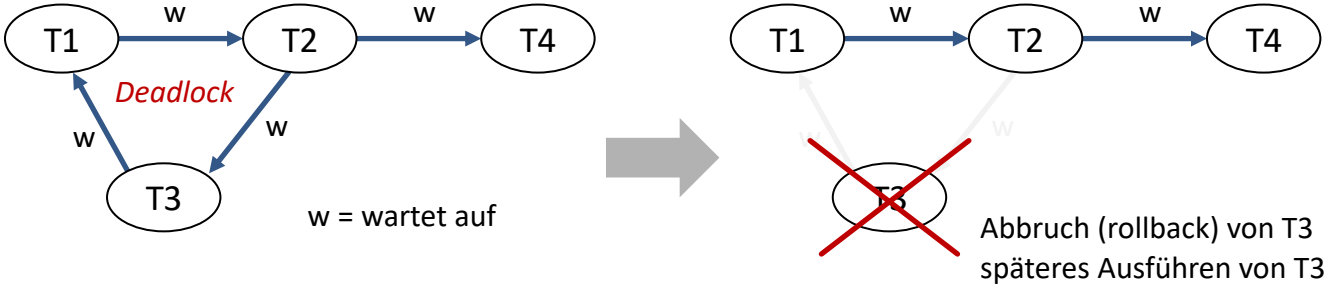
Vermeiden von Deadlocks

»wound-wait« Abbruch (rollback) von T1, wenn T1 jünger ist als T2, sonst wartet T2

»wait-die« Abbruch (rollback) von T2, wenn T2 jünger ist als T1, sonst wartet T2

Zulassen, aber Erkennen von Deadlocks

- Wartegraph mit Zyklen



- Beseitigen der Verklemmung durch Abbruch (rollback) und zeitliches Verschieben einer »geeigneten« Transaktion (z.B. der jüngsten)

Datenkommunikation und Rechnernetze

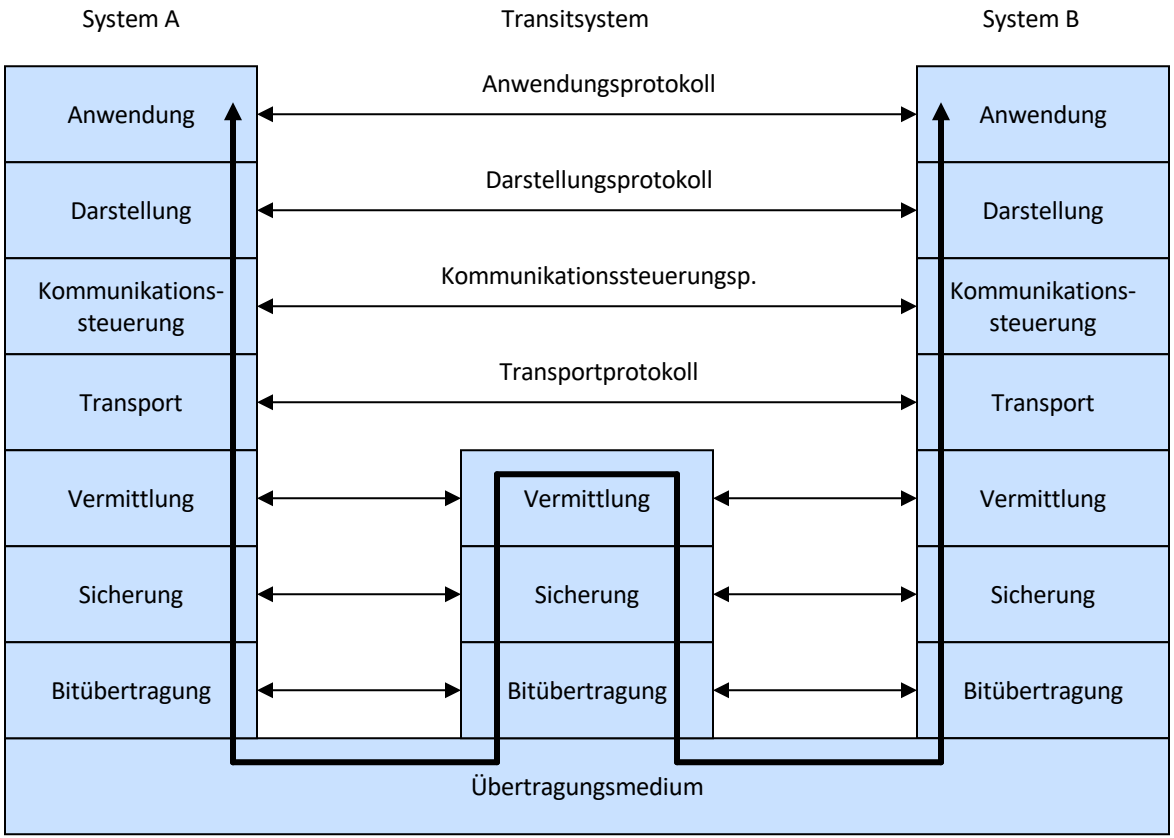
OSI-Referenzmodell: Aufgaben der 7 Schichten

ISO, 1977

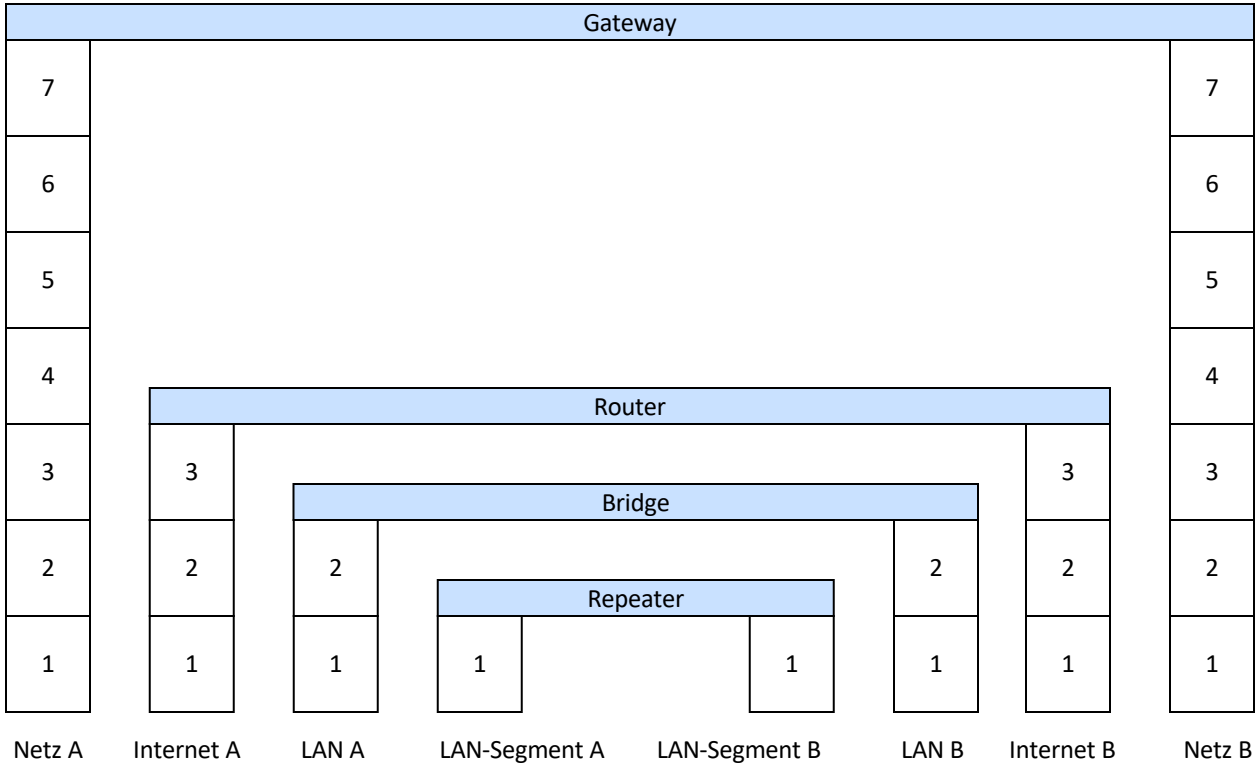
Das OSI-Referenzmodell ist ein abstraktes, logisch-funktionelles Architekturmodell für die Datenkommunikation in offenen Systemen (Open Systems Interconnection, OSI).

Application Layer	Anwendung	Anwendungsunterstützende Dienste Netzmanagement
Presentation Layer	Darstellung	Umsetzung von Daten in Standardformate Interpretation dieser gemeinsamen Formate
Session Layer	Kommunikations-steuerung	Prozess-zu-Prozess-Verbindung Prozesssynchronisation
Transport Layer	Transport	Logische Ende-zu-Ende-Verbindungen in Abstraktion der technischen Übertragungssysteme
Network Layer	Vermittlung	Wegbestimmung im Netz: Routing Datenflusskontrolle
Data Link Layer	Sicherung	Logische Verbindungen mit Datenpaketen Elementare Fehlererkennungsmechanismen
Physical Layer	Bitübertragung	Nachrichtentechnische Hilfsmittel für die Übertragung von Bits

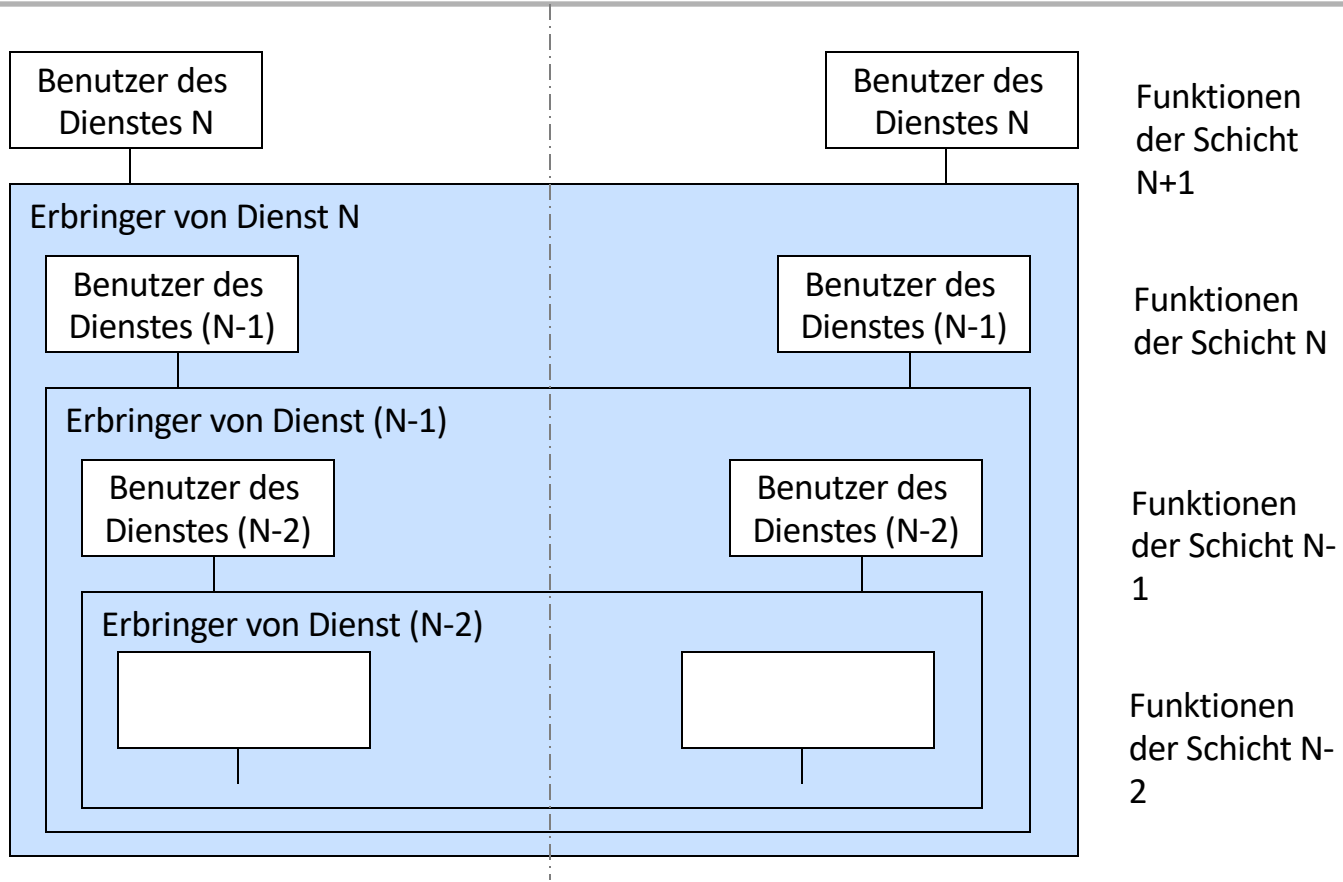
Schichten im OSI-Modell



Bezeichnungen der Funktionseinheiten beim »Internetworking«



Dienststruktur im OSI-Modell



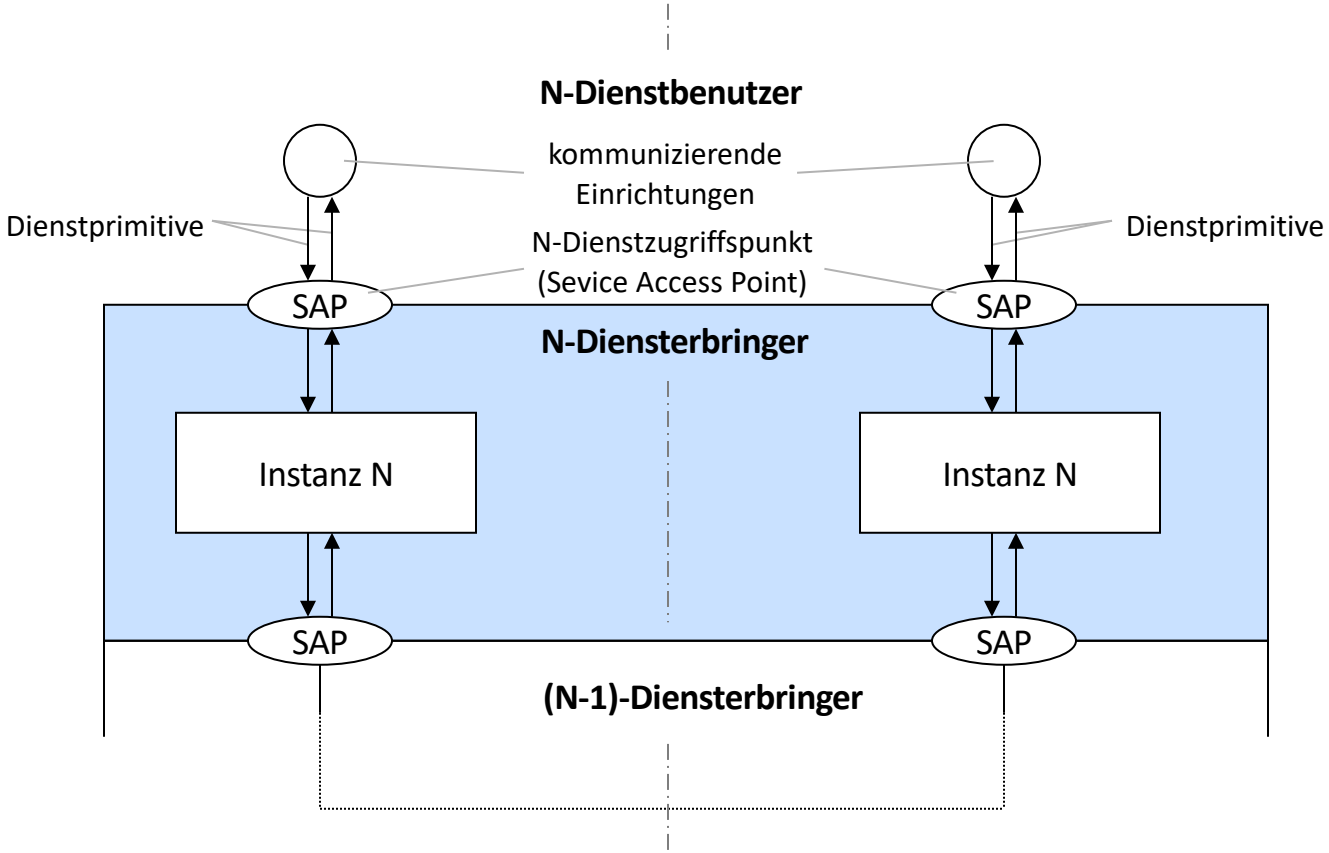
Funktionen der Schicht N+1

Funktionen der Schicht N

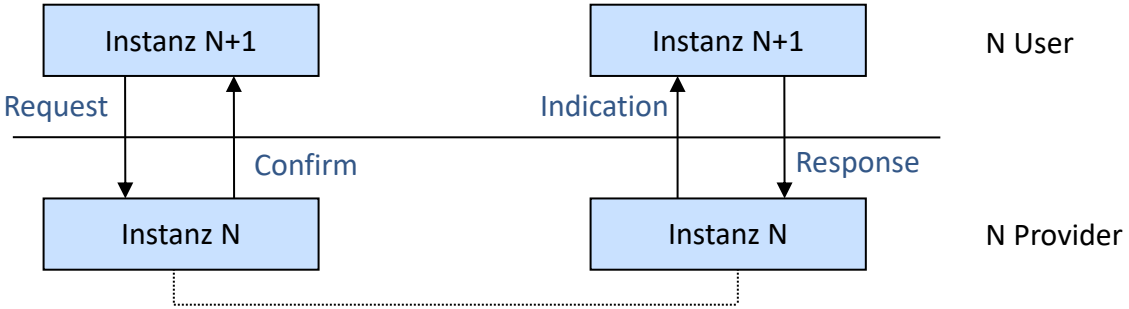
Funktionen der Schicht N-1

Funktionen der Schicht N-2

Prinzipieller Aufbau einer Kommunikationsschicht

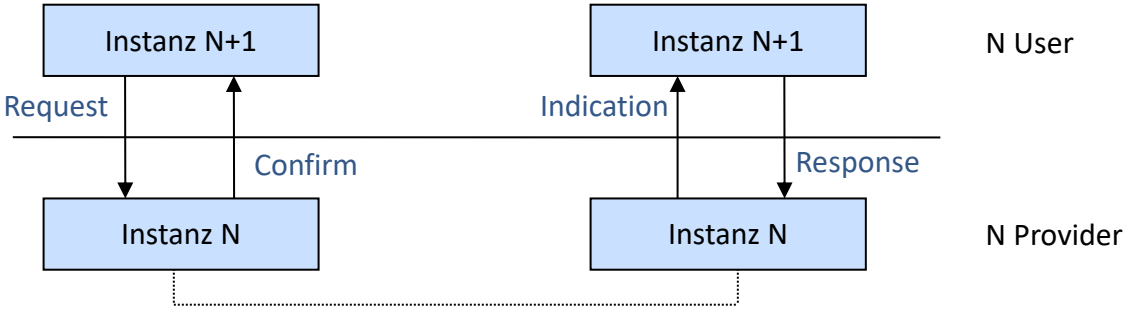


Aufrufschema (Dienstprimitive)



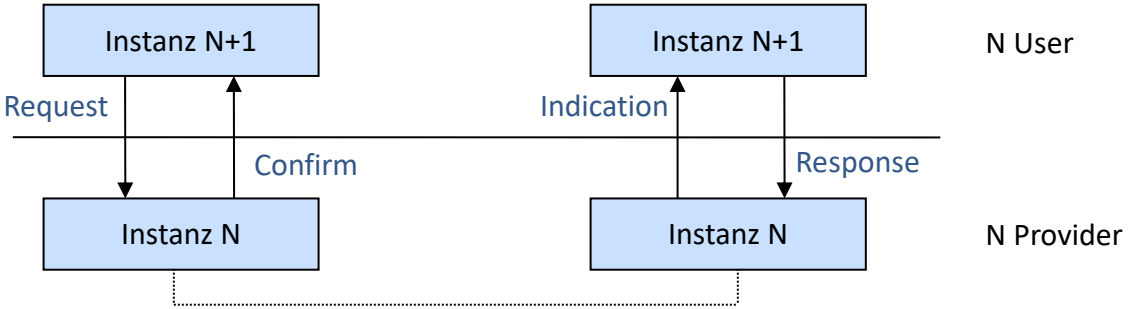
- Dienstmodell
  - Ein Dienstmodell ist ein abstraktes Darstellungsmittel zur Veranschaulichung der Kommunikationsbeziehungen in hierarchischen Mehrschichtenstrukturen oder an Netzgrenzen, an denen Kommunikationsdienste bereitgestellt werden.
- Dienstprimitive
  - abstrakte sprachliche Hilfsmittel zur Beschreibung von Kommunikationsvorgängen

Dienstprimitive



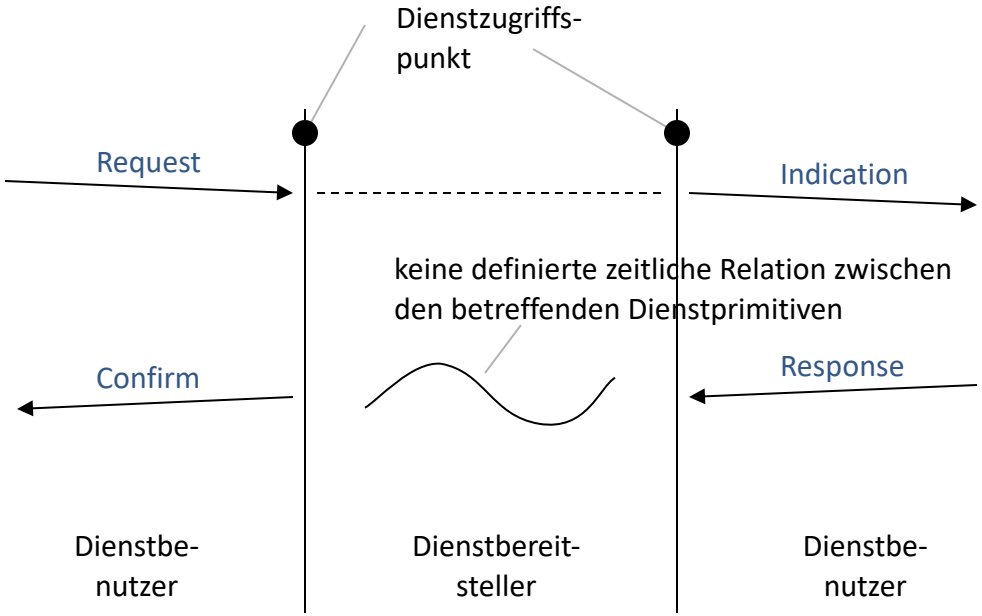
Primitivtyp	Bedeutung
Request	Eine Instanz fordert einen Dienst an.
Indication	Eine Instanz wird über ein Ereignis informiert.
Response (+ / –)	Eine Instanz möchte auf ein Ereignis antworten.
Confirm (+ / –)	Die Reaktion auf einen Request trifft ein.

Dienstprimitive



Primitivtyp	Richtung	Bedeutung
Request	User → Provider	Dienstanforderung
Indication	Provider → User	Dienstanzeige
Response (+ / –)	User → Provider	Dienstbeantwortung (positiv oder negativ)
Confirm (+ / –)	Provider → User	Dienstbestätigung (positiv oder negativ)

Ablaufdiagramm für Dienstmodelle



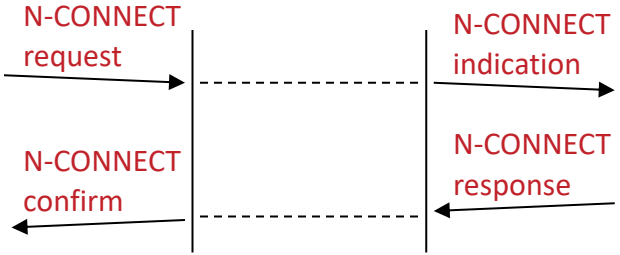
Dienstprimitive

- Syntax
  - <Primitivname> ::= <Initial> <Dienstart> <Primitivtyp>
- Initiale
  - symbolisieren eine bestimmte Schicht des OSI-Modells
  - angegeben durch eine Kurzbezeichnung der Schicht
- Dienstart
  - gibt die Art des Kommunikationsvorgangs an
  - Beispiele für Dienstarten:
    - CONNECT: Herstellen einer Verbindung
    - DATA: normaler Datentransfer
    - DISCONNECT: Auflösen einer Verbindung
- Primitivtyp
  - beschreibt ganz bestimmten Kommunikationseffekt

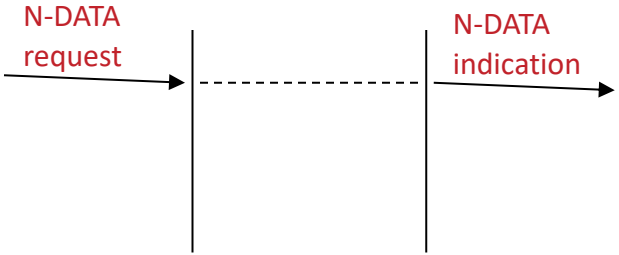
Beispiele für Kommunikationssituationen

- Syntax
  - <Primitivname> ::= <Initial> <Dienstart> <Primitivtyp>
- Beispiele
  - N-DISCONNECT indication
    - Netzwerkschicht zeigt Verbindungsauflösung an
  - T-CONNECT confirm
    - Transportschicht bestätigt Verbindungsaufbau

Erfolgreiches Herstellen einer N-Verbindung



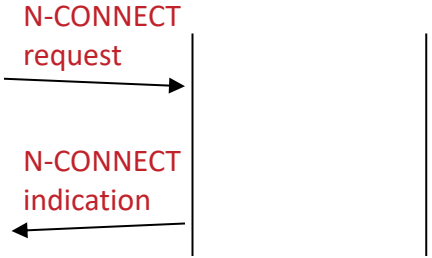
Üblicher Datentransfer



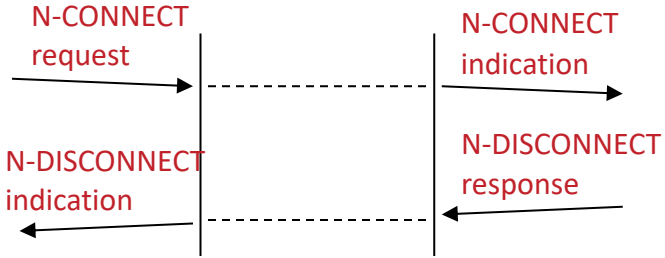
VSS 17.04.2020

Beispiele für Kommunikationssituationen

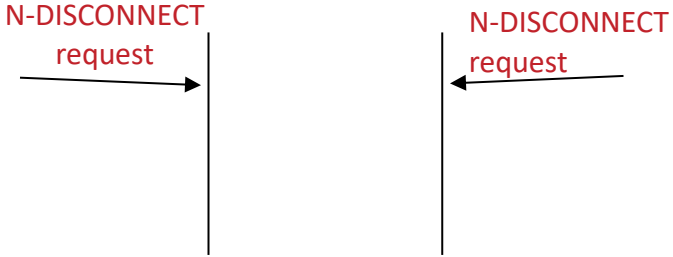
Zurückweisen einer Aufforderung zur Verbindungsherstellung vom Dienstbereitsteller



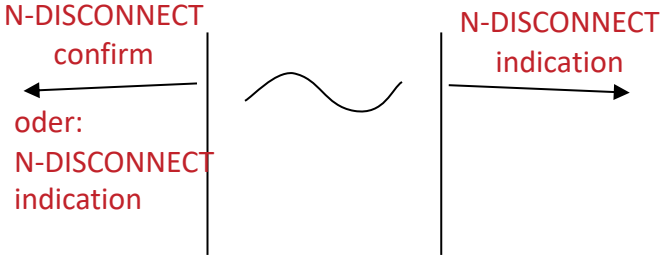
Zurückweisen einer Aufforderung zur Verbindungsherstellung durch den Kommunikationspartner



Auflösen einer Verbindung durch beide Dienstbenutzer



Auflösen einer Verbindung durch den Dienstbereitsteller

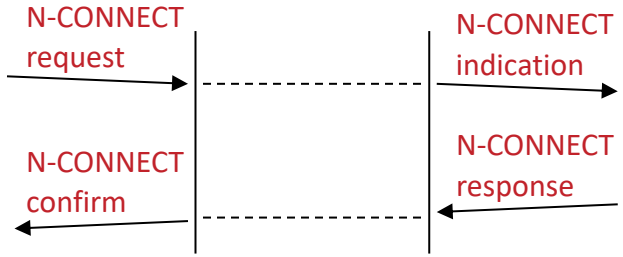


VSS 17.04.2020

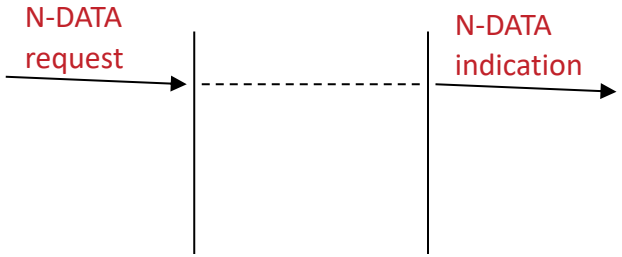
Bestätigte und unbestätigte Dienste

Man unterscheidet

- bestätigte Dienste (confirmed services)
  - request, indication, response, confirm



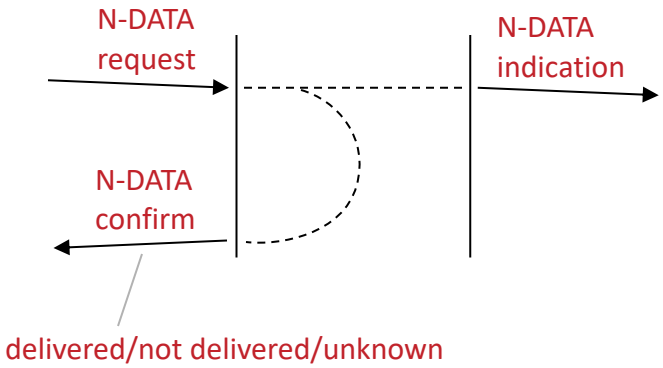
- unbestätigte Dienste (unconfirmed services)
  - request, indication



Verbindungslose und verbindungsorientierte Dienste

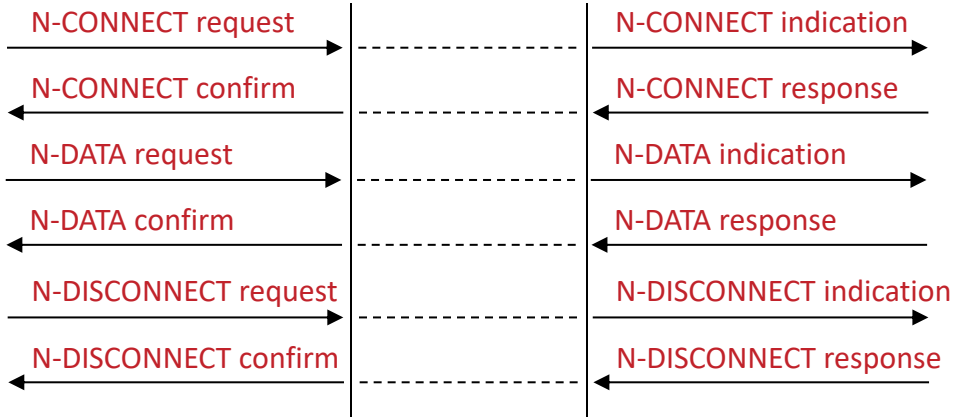
- Verbindungslos (CL, connectionless)
  - es existiert keine ständige logische Verbindung zwischen den Kommunikationspartnern
  - Daten werden mit Adresse des Empfängers versehen
  - tiefere Kommunikationsschicht sorgt für Übermittlung
  - Reihenfolgetreue der Nachrichten ist nicht gewährleistet

Bsp.: Übergabebestätigungsdienst

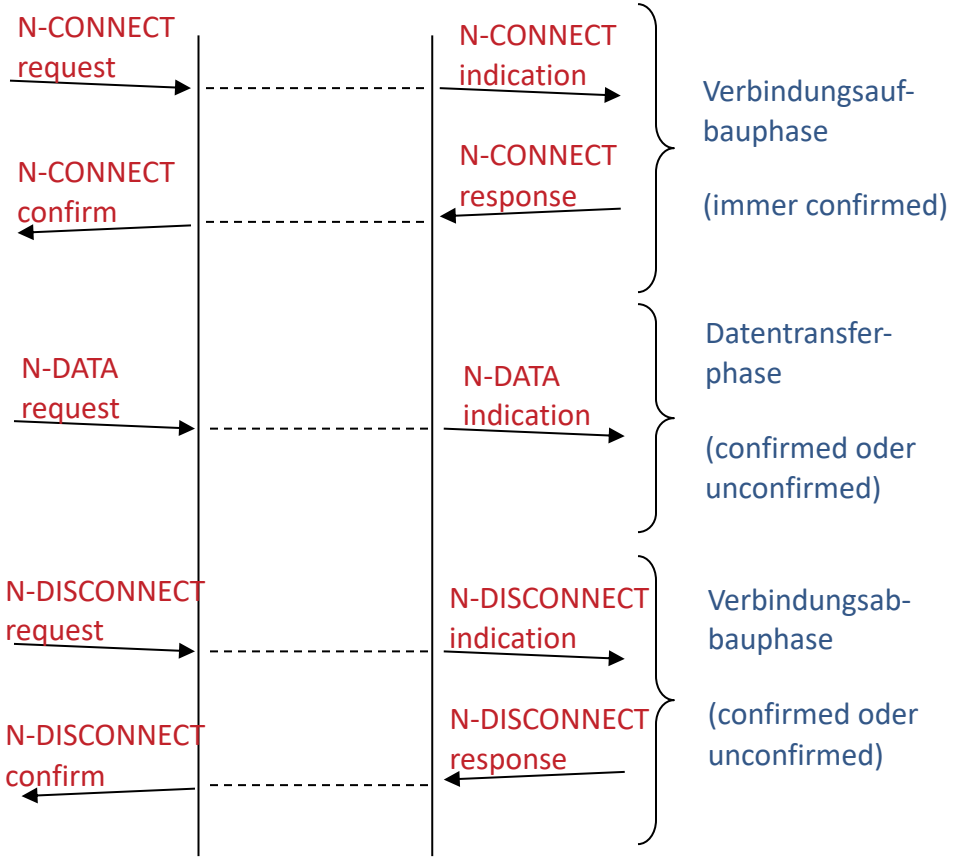


Verbindungslose und verbindungsorientierte Dienste

- Verbindungsorientiert (CO, connection-oriented)
  - Drei Kommunikationsphasen
    - Herstellen einer logischen Sender-Empfänger-Verbindung
    - Transferphase Übermittlung der Nutzdaten
    - Auflösen der logischen Verbindung
  - Reihenfolge der Nachrichten ist gewährleistet
  - Beispiel: zuverlässiger bestätigter verbindungsorientierter Datendienst



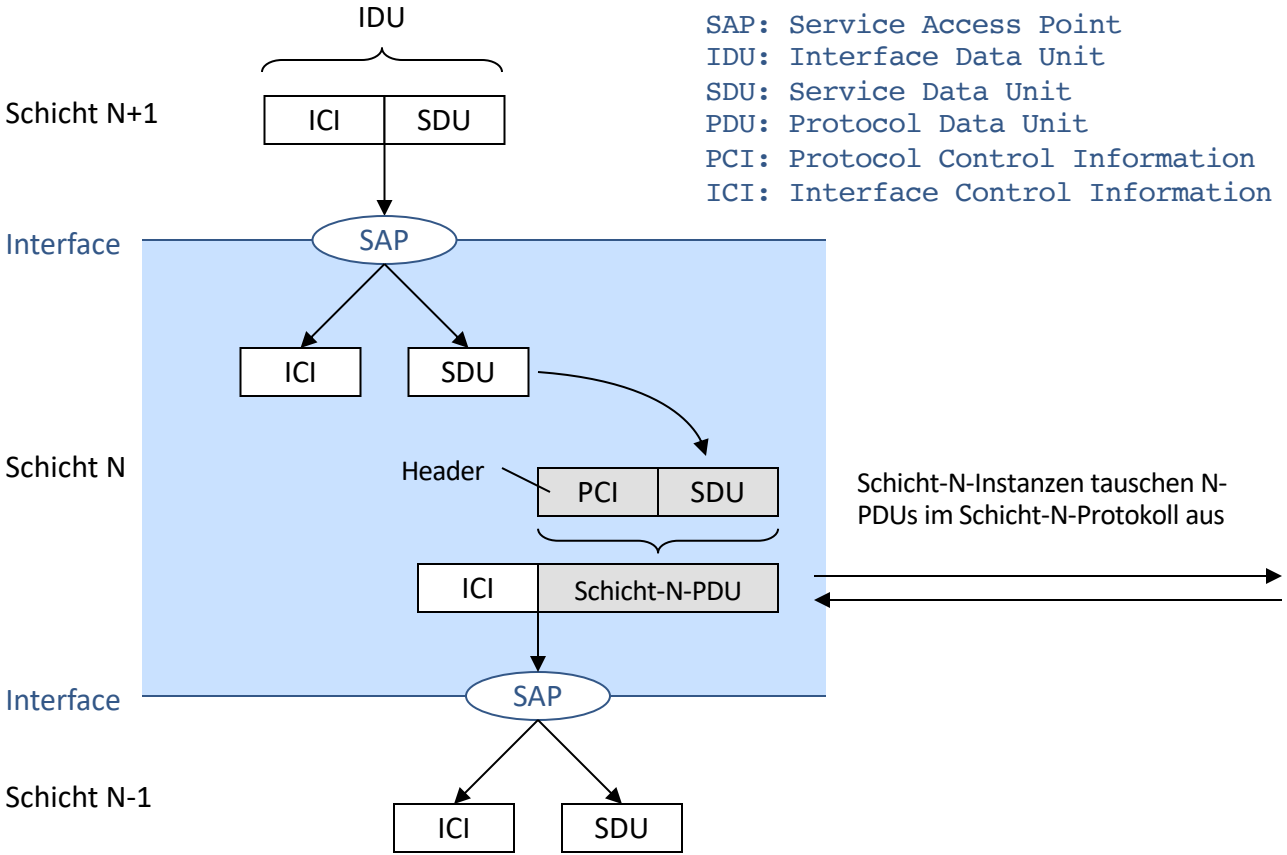
Allgemein: Die drei Phasen eines CO Service



Verschiedene Typen von CO und CL Services

	verbindungslos CL	verbindungsorientiert CO
unbestätigt (unconfirmed)	Beacons, Broadcasts (z.B. ARP)	Streaming
bestätigt (confimed)	Übergabebestätigungs- dienst	HTTP, SMTP, Remote Login

Arten von Dateneinheiten



SAP: Service Access Point
IDU: Interface Data Unit
SDU: Service Data Unit
PDU: Protocol Data Unit
PCI: Protocol Control Information
ICI: Interface Control Information

Zusammenfassung: Begriffe

- N-Dienstzugriffspunkt (Service Acces Point SAP)
  - Stelle an der Grenze zwischen Schichten N und N+1, an dem ein N-Dienst bereitgestellt wird
- N-Instanz (Entity, Einrichtung)
  - Objekt der Schicht N, das in der Lage ist, Daten zu senden oder zu empfangen
- N-Verbindung
  - logisches Sender-/Empfänger-verhältnis zwischen N-SAPs
- N-Protokoll
  - Menge aller semantischen und syntaktischen Festlegungen, die das Kommunikationsverhalten von N-Entities definieren
- N-ICI (Interface-Steuerinformation, Interface Control Information)
  - werden zwischen N+1-Entities und N-Entities ausgetauscht, um deren Zusammenwirken zu koordinieren
- N-IDU (Interface-Dateneinheit, Interface Data Unit)
  - IDU = ICI + SDU
- N-SDU (Dienst-Dateneinheit, Service Data Unit)
  - Teil von Interface-Daten, der unverändert übertragen wird
- N-PCI (Protokoll-Steuerinformation, Protocol Control Information)
  - koordinieren das Zusammenwirken von N-Entities
- N-PDU (Protokoll-Dateneinheit)
  - PDU = PCI + SDU

Bezeichnungen der Schicht-N-PDUs

A	Anwendungsschicht	(Application Layer)	APDU
P	Darstellungsschicht	(Presentation Layer)	PPDU
S	Kommunikationssteuerungsschicht	(Session Layer)	SPDU
T	Transportschicht	(Transport Layer)	TPDU
N	Vermittlungsschicht	(Network Layer)	Packet
DL	Sicherungsschicht	(Data Link Layer)	Frame
PH, PHY	Bitübertragungsschicht	(Physical Layer)	Bit

Beispiel: Schicht-2-PDU (Frame, Rahmen)

- Schicht-2-Rahmen von Frame Relay (ITU-T I.233)

Flag 01111110	Address 8 Bit	Control 8 Bit	Datenfeld variable Länge	FCS 16 Bit	Flag 01111110

Eindeutiges Bitmuster zur Erkennung der Länge PDU

Adresse der Empfänger-instanz

Steuerinformation für die Protokollabwicklung:
– Informationsrahmen (Daten)
– Steuer-/Meldungsrahmen (zur Komm. zwischen Entities)
– Kombinierter Rahmen

Inhalt, übertragen mit Bitstuffing: Nach 5x1 folgt ein zusätzliches 0 Bit, das der Empfänger entfernt

Fehler-siche-rungs-feld (Prüf-summe)

Eindeutiges Bitmuster zur Erkennung der Länge PDU

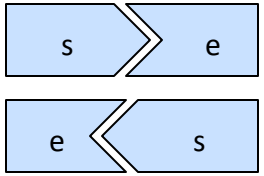
Arbeitsweise einer Instanz

- Beschreibung der Funktion einer Instanz als Zustandsautomat
  - Communication Extended Finite State Machine (EFSM)

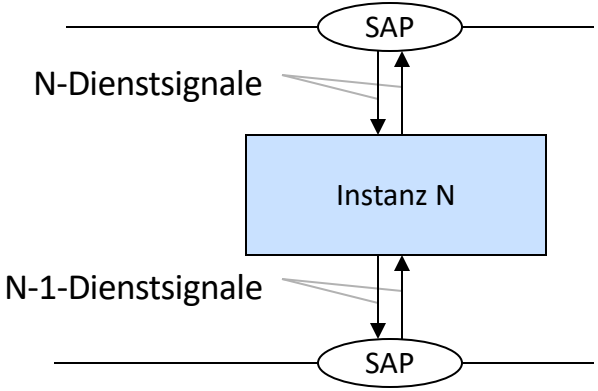
- Elemente

Zustand

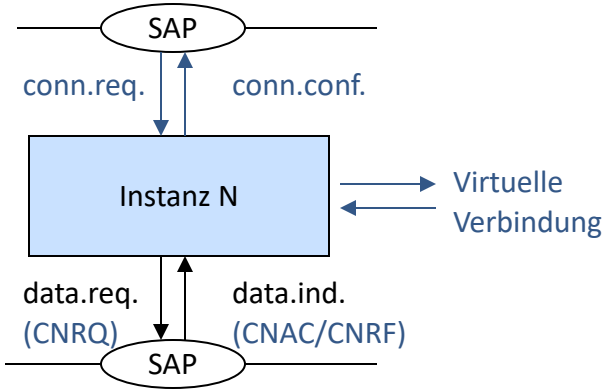
Signal / Ereignis



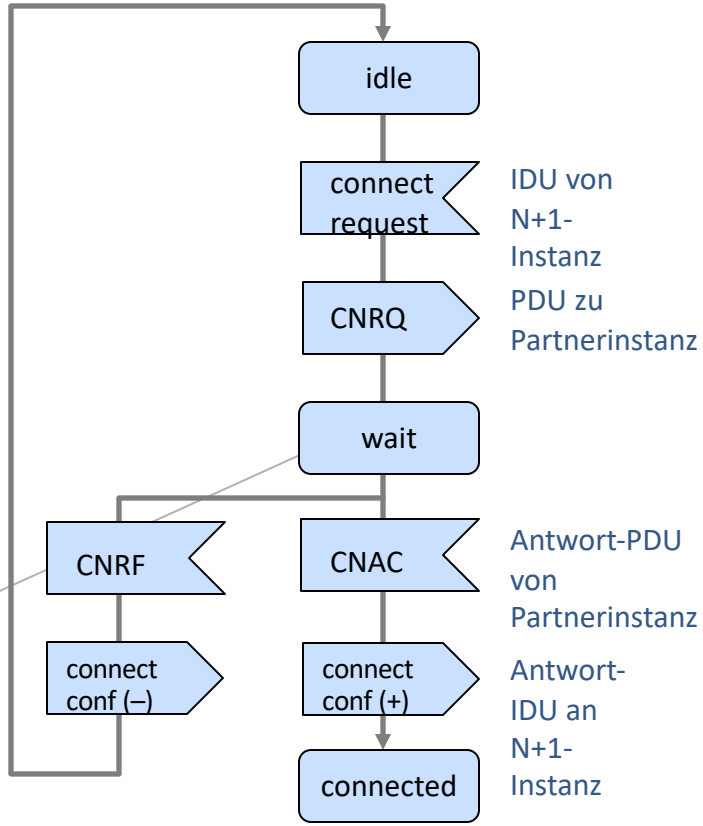
Transition



Beispiel: Verbindungsaufbau

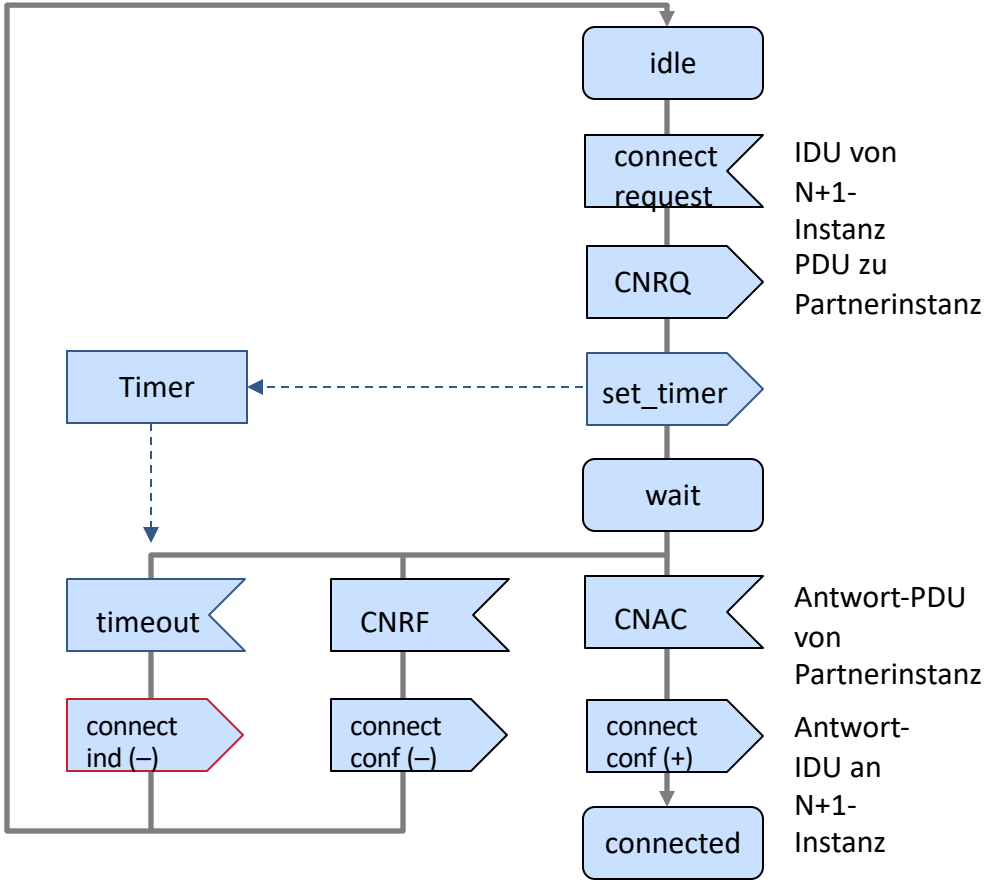


CNRQ: Connection Request
CNRF: Connection Refused
CNAC: Connection Accepted



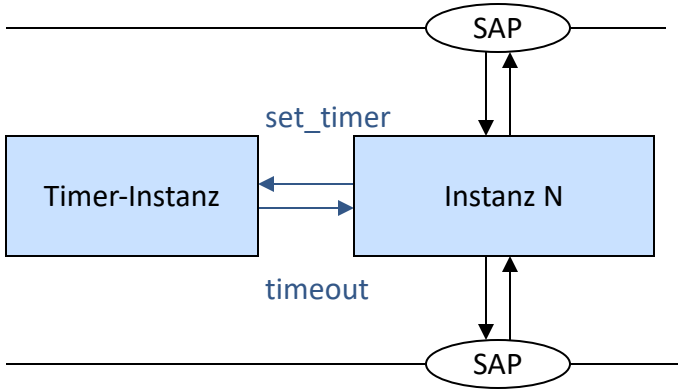
terminiert nicht, falls Partnerinstanz nicht antwortet

Beispiel: Verbindungsaufbau mit Zeitsteuerung

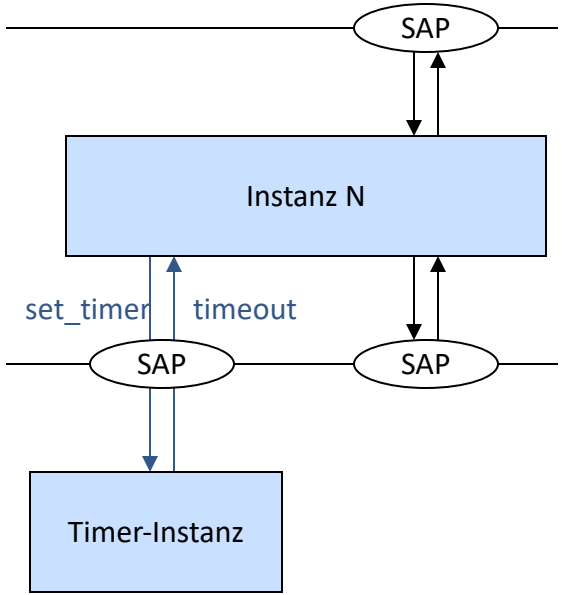


Beispiel: Verbindungsaufbau mit Zeitsteuerung

Timer als schicht-interne Funktion



Timer als N-1-Dienst

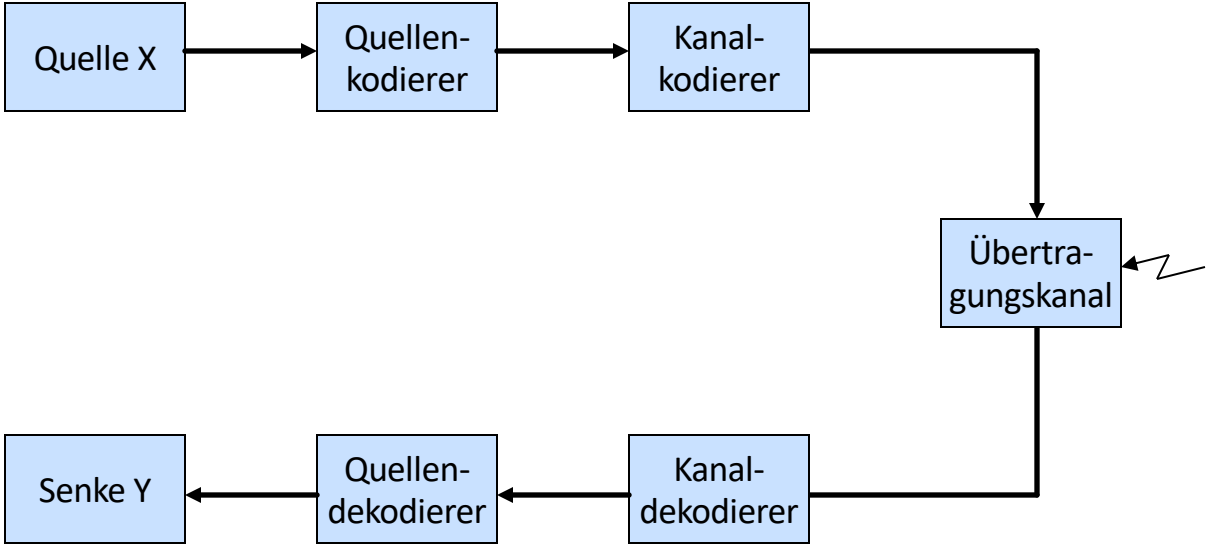


Physikalische Schicht

- Medium
  - freier Raum
  - verdrillte Telefonleitung (twisted pair)
  - Koaxialkabel
  - Lichtwellenleiter
- Allgemeines Nachrichtenübertragungsmodell
- Belegungsstrategien
- gestörte Kanäle

A	(Application Layer)
P	(Presentation Layer)
S	(Session Layer)
T	(Transport Layer)
N	(Network Layer)
DL	(Data Link Layer)
PH, PHY	(Physical Layer)

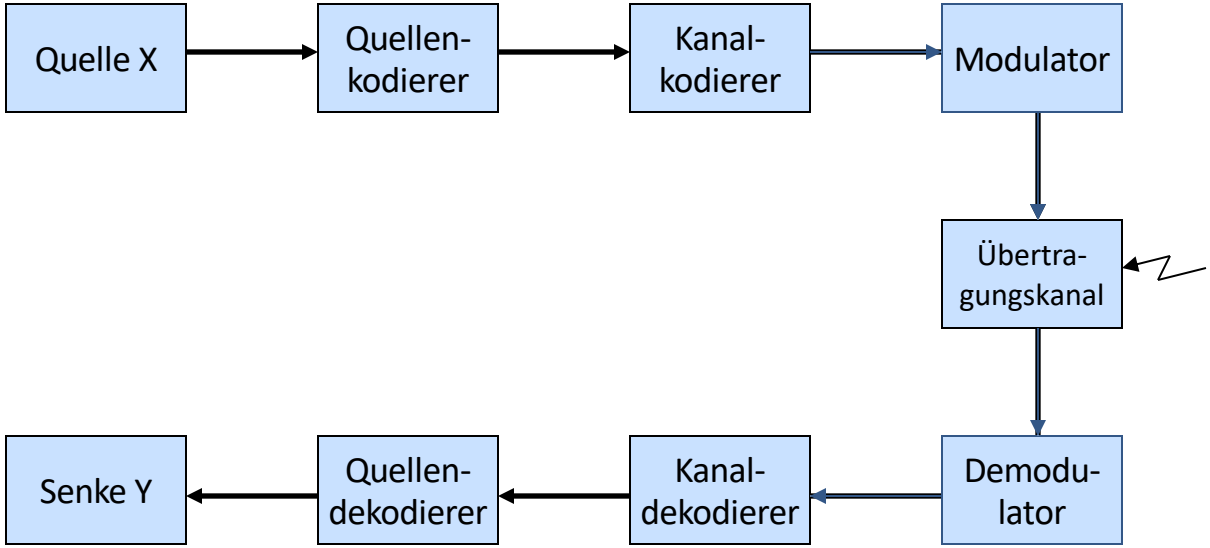
Allgemeines Nachrichtenübertragungsmodell



Belegungsstrategien

- Basisbandübertragung
  - Signale werden in der ursprünglichen Frequenzlage im Übertragungskanal übermittelt
  - keine Modulation
- Geträgerte Übertragung
  - Modulation: deterministisches Trägersignal wird durch informationstragende Signale beeinflusst
  - Senderseitig:
    - Modulation
  - Empfängerseitig:
    - Demodulation
  - Mit oder ohne Mehrfachnuntzung der Bandbreite
    - Zeitmultiplex (Time Division Multiple Access TDMA)
    - Frequenzmultiplex (Frequency Division Multiple Access FDMA)
    - neuerdings auch: Codemultiplex (CDMA)

Geträgerte Übertragung



Modulationsformen

- Harmonische Schwingung:

$$x_T(t) = A_T \cos(\omega_T + \varphi_T)$$

  - Amplitudenmodulation (AM)

$$A_T = A_T(t) = f(x_T(t))$$

  - Frequenzmodulation (FM)

$$\omega_T = \omega_T(t) = f(x_T(t))$$

  - Phasenmodulation (PM)

$$\varphi_T = \varphi_T(t) = f(x_T(t))$$

Modulationsverfahren

modulierendes Signal

Amplitudentastung

Frequenzumtastung

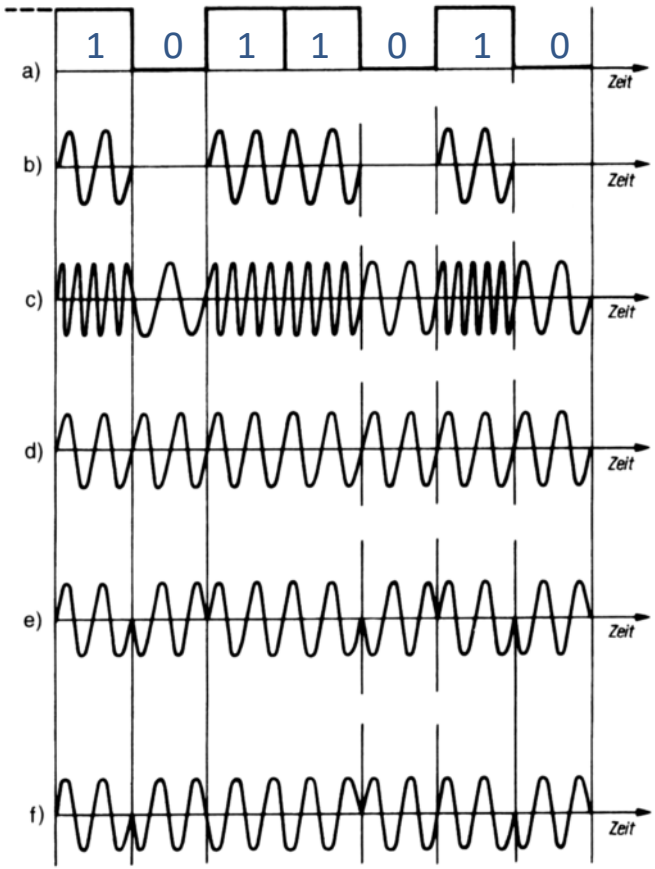
Bezugsschwingung für Phasenmodulation

Phasenumtastung

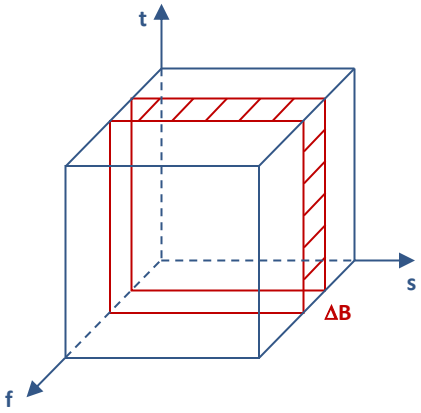
1:$\varphi(t)=0$ 0:$\varphi(t)=\pi$

Differenzphasenumtastung

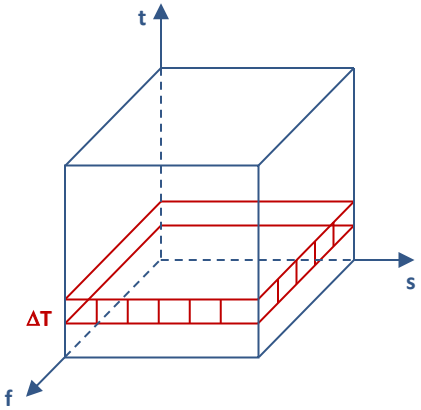
1:$\varphi(t)=\varphi(t-1)$ 0:$\varphi(t)=-\varphi(t-1)$



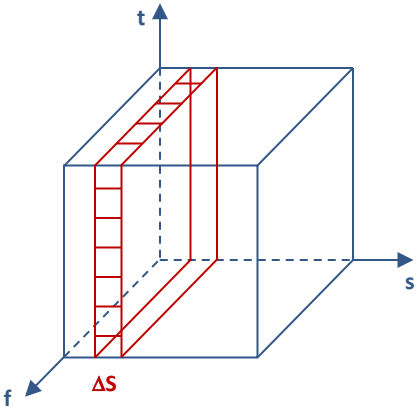
Multiplexing



Frequenzmultiplex



Zeitmultiplex



Codemultiplex

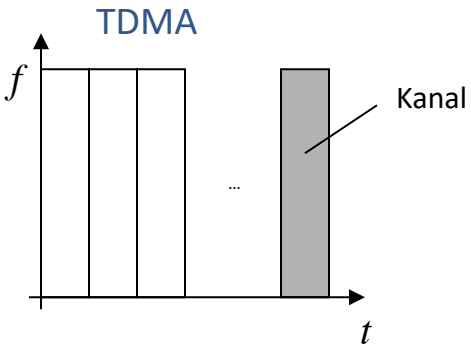
f Frequenzachse
s Leistungsdichteachse
t Zeitachse

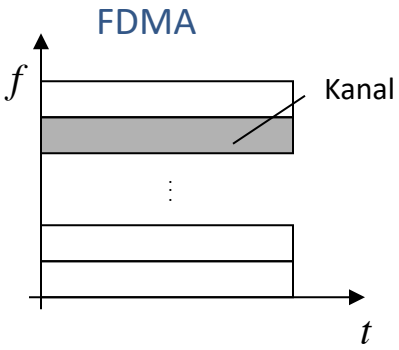
B Systembandbreite
S Leistungsdichte
T Rahmentaktzeit

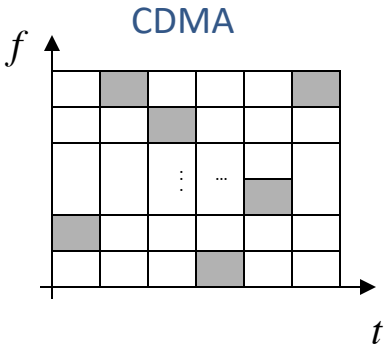
ΔB Kanalbandbreite
ΔS Kanalleistungsdichte
ΔT Kanalbelegungszeit

VSS 17.04.2020

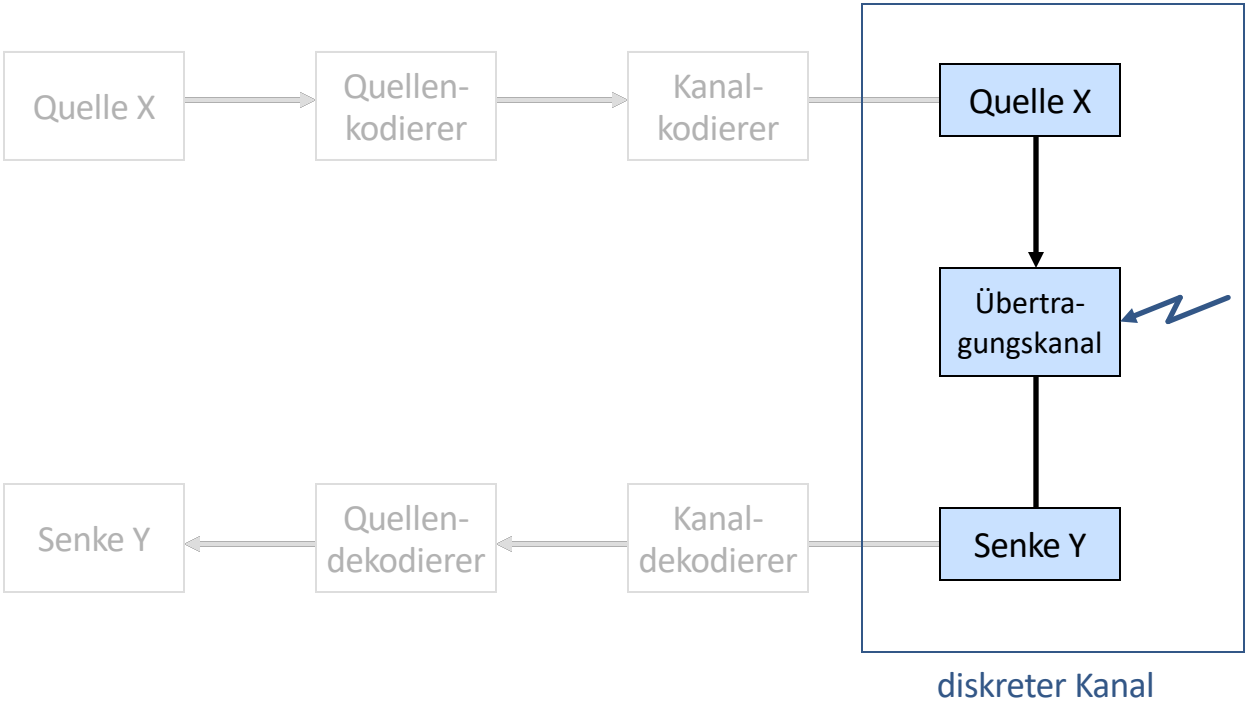
Multiplexing (vereinfacht)







Allgemeines Nachrichtenübertragungsmodell

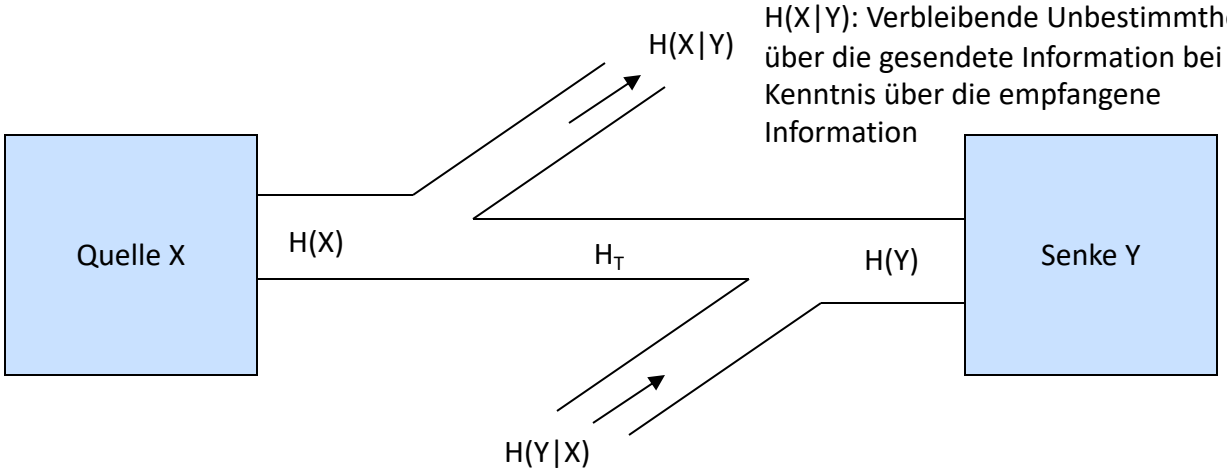


Kanalmodell: Bergersches Entropiemodell

- H(X) Entropie am Kanaleingang
- H(Y) Entropie am Kanalausgang
- H(X|Y) Äquivokation (Rückschlussentropie)
- H(Y|X) Irrelevanz (Störentropie)
- $H_T = H(X) - H(X|Y)$ Transinformation

Transinformation: Informationsmenge, die im Mittel durch ein Kanalzeichen (KZ) vom Sender zum Empfänger übertragen werden kann.

H(X|Y): Verbleibende Unbestimmtheit über die gesendete Information bei Kenntnis über die empfangene Information



Idealer Kanal (ungestörter Kanal): $H(X) = H(Y) = H_T$

Exkurs in die Informationstheorie

- Quelle:
  - beschrieben durch ein statisches Modell (diskrete Quelle)
  - verfügbare Zeichen x_i mit $i = 1...N$
  - deren Auftrittswahrscheinlichkeiten $p(x_i)$
  - N Elementarzeichen

- Quelle wählt Zeichen aus endlichem Zeichenvorrat von N Elementarzeichen aus und produziert Zeichenfolgen (und damit Nachrichten)

- Maß für den Codieraufwand, der bei Bildung einer Nachricht entsteht:
  - Anzahl der Binärentscheidungen bis zur Auswahl eines bestimmten Zeichens.
  - Beispiel:
    - Zeichenvorrat 2 Elemente –> binäre Entscheidung –> Maß von 1 Bit

Exkurs in die Informationstheorie

- Entscheidungsgehalt einer Quelle mit N Zeichen:

$$H_0 = \log_2 N \quad \text{bit/Zeichen}$$

- Informationsgehalt eines einzelnen Zeichens xi hängt von seiner Auftrittswahrscheinlichkeit ab:

$$I(x_i) = \log_2\left(\frac{1}{p(x_i)}\right) = -\log_2 p(x_i) \quad \text{bit}$$

(d.h. der Informationsgehalt eines Zeichens ist umso größer, je geringer seine Auftrittswahrscheinlichkeit ist)

- Entropie beschreibt den mittleren Informationsgehalt einer Quelle:

$$H(X) = \sum_{i=1}^{N} p(x_i) I(x_i) = \sum_{i=1}^{N} p(x_i) \log_2 \frac{1}{p(x_i)} = -\sum_{i=1}^{N} p(x_i) \log_2 p(x_i)$$

- Redundanz ist die Differenz zwischen Entscheidungsgehalt H_O und Entropie H(X) einer Quelle:

$$R_Q = H_0 - H(X)$$

- Beispiel: Typ. Auftrittswkten. der Zeichen d. dt. und engl. Sprache:

Zeichen:	a	b	c	d	e	f	g	h	i	j	k	l	m
deutsch	6.47	1.93	2.68	4.83	17.48	1.65	3.06	4.23	7.73	0.27	1.46	3.49	2.58
englisch	8.04	1.54	3.06	3.99	12.51	2.30	1.96	5.49	7.26	0.16	0.67	4.14	2.53

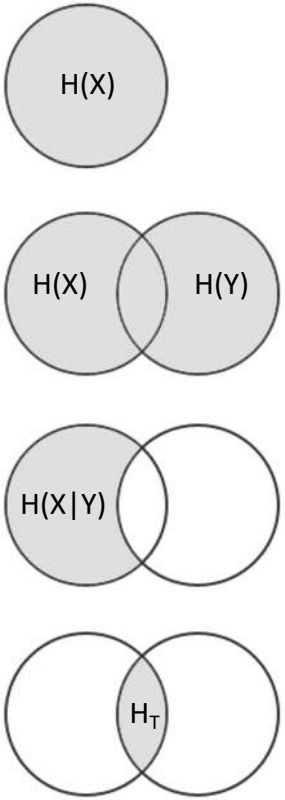
Zeichen:	n	o	p	q	r	s	t	u	v	w	x	y	z
deutsch	9.84	2.98	0.96	0.02	7.54	6.83	6.13	4.17	0.94	1.48	0.04	0.08	1.14
englisch	7.09	7.60	2.00	0.11	6.12	6.54	9.25	2.71	0.99	1.92	0.19	1.73	0.09

(Berechnung von R ggf. als Übung)

- Beachte: Redundanz der Sprache ist tatsächlich wesentlich höher:
  - Abhängigkeiten zwischen Zeichen, Wörtern und Sätzen bleiben hier völlig unberücksichtigt.

Exkurs in die Informationstheorie

- Für ein Alphabet X ist die
  - Entropie H(X) ein Maß für die Unbestimmtheit über das Auftreten eines konkreten Elements.
  - Verbundentropie H(X,Y) ist die Entropie der Vereinigung der Alphabete X und Y.
  - Bedingte Entropie H(X|Y) ist die verbleibende Unbestimmtheit über X bei Kenntnis von Y.
  - Transinformation $H_T(X;Y)$ ist die Information, die über X aus der Beobachtung von Y gewonnen werden kann.



- Berechnung der Entropie H(X)
  - Es seien x_i die Elemente der Menge X und $p(x_i)$ deren Auftrittswahrscheinlichkeiten.

$$H(X) = \sum_{i=1}^{n} p(x_i) \log \frac{1}{p(x_i)} = -\sum_{i=1}^{n} p(x_i) \log p(x_i)$$

Darstellung gestörter diskreter Kanäle

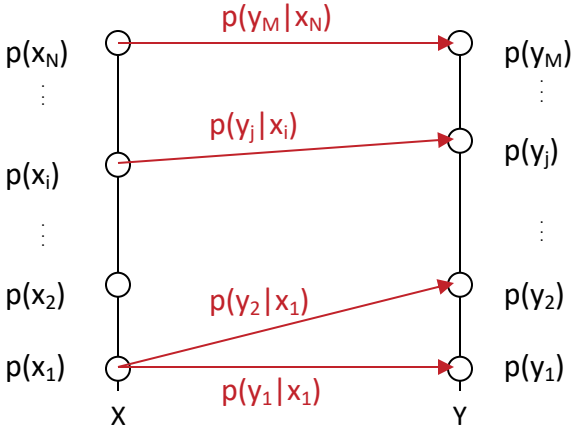
Quelle $X = \{ x_1, x_2, x_3, \dots , x_N \}$ mit N Zeichen x_i

Senke $Y = \{ y_1, y_2, y_3, \dots , y_M \}$ mit M Zeichen y_j

$p(x_i)$ Wkt f. d. Auftretens des Zeichens am x_i Kanaleingang

$p(y_j)$ Wkt f. d. Auftretens des Zeichens am y_j Kanalausgang

$$(p(x_i)) = (p(x_1) \quad p(x_2) \quad \cdots \quad p(x_N))$$
$$(p(y_j)) = (p(y_1) \quad p(y_2) \quad \cdots \quad p(y_M))$$



Darstellung gestörter diskreter Kanäle

- $p(y_j|x_i)$ bedingte Wkt., dass bei gesendetem Zeichen x_i das Zeichen am y_j empfangen wird.

$$\left(p(y_j \mid x_i)\right) = \begin{pmatrix} p(y_1 \mid x_1) & p(y_2 \mid x_1) & \cdots & p(y_M \mid x_1) \\ p(y_1 \mid x_2) & p(y_2 \mid x_2) & \cdots & p(y_M \mid x_2) \\ \vdots & \vdots & \ddots & \vdots \\ p(y_1 \mid x_N) & p(y_2 \mid x_N) & \cdots & p(y_M \mid x_N) \end{pmatrix}$$

Bei fehlerfreier Übertragung:

$p(y_j|x_i) = 1$ für $i=j$ und

$p(y_j|x_i) = 0$ für $i \neq j$

$$\left(p(x_i \mid y_j)\right) = \begin{pmatrix} p(x_1 \mid y_1) & p(x_2 \mid y_1) & \cdots & p(x_N \mid y_1) \\ p(x_1 \mid y_2) & p(x_2 \mid y_2) & \cdots & p(x_N \mid y_2) \\ \vdots & \vdots & \ddots & \vdots \\ p(x_1 \mid y_M) & p(x_2 \mid y_M) & \cdots & p(x_N \mid y_M) \end{pmatrix}$$

Darstellung gestörter diskreter Kanäle

- Zusammenhang von $p(x_i)$ und $p(y_j)$

$$\begin{pmatrix} p(y_1) \\ p(y_2) \\ \vdots \\ p(y_M) \end{pmatrix} = \left(p(y_j \mid x_i)\right)^T \cdot \begin{pmatrix} p(x_1) \\ p(x_2) \\ \vdots \\ p(x_N) \end{pmatrix}$$

$$\begin{pmatrix} p(y_1) \\ p(y_2) \\ \vdots \\ p(y_M) \end{pmatrix} = \begin{pmatrix} p(y_1 \mid x_1) & p(y_1 \mid x_2) & \cdots & p(y_1 \mid x_N) \\ p(y_2 \mid x_1) & p(y_2 \mid x_2) & \cdots & p(y_2 \mid x_N) \\ \vdots & \vdots & \ddots & \vdots \\ p(y_M \mid x_1) & p(y_M \mid x_2) & \cdots & p(y_M \mid x_N) \end{pmatrix} \cdot \begin{pmatrix} p(x_1) \\ p(x_2) \\ \vdots \\ p(x_N) \end{pmatrix}$$

Berechnung von H(X), H(Y) und H_T

$$H(Y) = -\sum_j p(y_j)\,\mathrm{ld}\,p(y_j)$$

$$H(X) = -\sum_i p(x_i)\,\mathrm{ld}\,p(x_i)$$

$$H(Y \mid X) = -\sum_i p(x_i) \sum_j p(y_j \mid x_i)\,\mathrm{ld}\,p(y_j \mid x_i) \quad \text{Störentropie}$$

$$H(X \mid Y) = -\sum_j p(y_j) \sum_i p(x_i \mid y_j)\,\mathrm{ld}\,p(x_i \mid y_j) \quad \text{Äquivokation}$$

$$\begin{aligned} H_T &= H(Y) - H(Y \mid X) \\ &= H(X) - H(X \mid Y) \end{aligned}$$

Maßeinheit: bit/KZ (bit/Kanalzeichen)

Beispiel: Aufg.

Quelle $X = \{x_1, x_2, x_3\}$ mit $(p(xi)) = (0{,}1 \quad 0{,}3 \quad 0{,}6)$

Senke $Y = \{y_1, y_2, y_3\}$

Übergangswahrscheinlichkeiten $p(y_j \mid x_i)$:

$$\left(p(y_j \mid x_i)\right) = \begin{pmatrix} 0{,}70 & 0{,}20 & 0{,}10 \\ 0{,}05 & 0{,}80 & 0{,}15 \\ 0{,}10 & 0{,}10 & 0{,}80 \end{pmatrix}$$

Die Transinformation ist zu bestimmen.

Lös.:

$$p(y_1) = p(y_1 \mid x_1)p(x_1) + p(y_1 \mid x_2)p(x_2) + p(y_1 \mid x_3)p(x_3)$$
$$p(y_2) = p(y_2 \mid x_1)p(x_1) + p(y_2 \mid x_2)p(x_2) + p(y_2 \mid x_3)p(x_3)$$
$$p(y_3) = p(y_3 \mid x_1)p(x_1) + p(y_3 \mid x_2)p(x_2) + p(y_3 \mid x_3)p(x_3)$$

oder $$\begin{pmatrix} p(y_1) \\ p(y_2) \\ p(y_3) \end{pmatrix} = \left(p(y_j \mid x_i)\right)^T \begin{pmatrix} p(x_1) \\ p(x_2) \\ p(x_3) \end{pmatrix} = \begin{pmatrix} 0{,}15 \\ 0{,}32 \\ 0{,}53 \end{pmatrix}$$

$$H(Y) = -\sum_{j=1}^{3} p(y_j)\,\mathrm{ld}\,p(y_j) = 1{,}42 \text{ bit/KZ}$$

$$H(Y \mid X) = -\sum_{i=1}^{3} p(x_i) \sum_{j=1}^{3} p(y_j \mid x_i)\,\mathrm{ld}\,p(y_j \mid x_i) = 0{,}93 \text{ bit/KZ}$$

$$H_T = H(Y) - H(Y \mid X) = 0{,}49 \text{ bit/KZ}$$

VSS 17.04.2020

Sicherungsschicht

- Aufgaben
  - Framing
  - Fehlersicherung
  - Übertragungssteuerung

- Beispiele für Schicht-2-Dienste
  - Local Area Networks: Ethernet
  - Wireless LANs

A	(Application Layer)
P	(Presentation Layer)
S	(Session Layer)
T	(Transport Layer)
N	(Network Layer)
DL	(Data Link Layer)
PH, PHY	(Physical Layer)

Fehlerarten

```
                    00010010 11010101 10111101 01111001
Einzelfehler:       00010011 11010001 10111101 01111101
Unabh. Mehrfachf.:  00010010 11010101 11111001 01111001
Bursts:             000101** ******11 10111101 01111001
```

Aufgaben der Sicherungsschicht

- Framing
  - Verpacken der Daten der höheren Schichten in einen Senderahmen mit Header und ggf. Trailer

Flag 01111110	Header-Info	Datenfeld	Trailer-Info	Flag 01111110

- Übertragungssteuerung
  - Steuerung des Zugriffs der Teilnehmerstationen auf das Medium
  - Verhindern oder Erkennen von Übertragungskolissionen
  - Medienzugriffsverfahren

- Fehlersicherung
  - Schutz gegen Störeinflüsse auf dem Medium
  - Fehlererkennung, -korrektur, wiederholte Übertragung

Fehlererkennung

- Parität
  - Quellcodewort wird ergänzt um zusätzliches Prüfbit P
  - Einzelfehler im Kanalcodewort sind erkennbar
  - Beispiel: (x_1 x_2 x_3 P) mit P = $x_1+x_2+x_3$ (mod 2)
    - Anzahl der Einsen in x_i ist gerade: P=0, sonst: P=1
    - Es gilt: 0 = $x_1+x_2+x_3+P$ (mod 2)

0000 ist ein Kanalcodewort	1000 ist kein Kanalcodewort
0001 ist kein Kanalcodewort	1001 ist ein Kanalcodewort
0010 ist kein Kanalcodewort	1010 ist ein Kanalcodewort
0011 ist ein Kanalcodewort	1011 ist kein Kanalcodewort
0100 ist kein Kanalcodewort	1100 ist ein Kanalcodewort
0101 ist ein Kanalcodewort	1101 ist kein Kanalcodewort
0110 ist ein Kanalcodewort	1110 ist kein Kanalcodewort
0111 ist kein Kanalcodewort	1111 ist ein Kanalcodewort

Beobachtung: Gültige Kodewörter unterscheiden sich stets in (wenigstens) 2 Stellen.

Fehlererkennung: Cyclic Redundancy Check (CRC)

- Polynom G(x) vom Grad r
- Rahmen M(x) mit m Bits

```
M(x)=1101011011
G(x)=x^4+x+1=10011
```

- r Nullbits an niederwertigstes Bit des Rahmens anhängen
  - entspricht Polynom $x^r \cdot M(x)$, bestehend aus m+r Bit
- teile $x^r \cdot M(x)$ durch G(x) (modulo 2)

```
11010110110000
10011
 10011
 10011
  000010110
      10011
       010100
        10011
         01110=Divisionsrest
```

- subtrahiere *Divisionsrest* von $x^r \cdot M(x)$ (modulo 2)
- Ergebnis T(x) ist Frame mit CRC

```
11010110110000
         01110
11010110111110=T(x)
```

Fehlererkennung: CRC-Prüfung

- Polynom G(x) vom Grad r
- Empfänger dividiert empfangenes Frame durch G(x)
  - Divisionsrest == 0: OK
  - sonst: FEHLER

    ↓

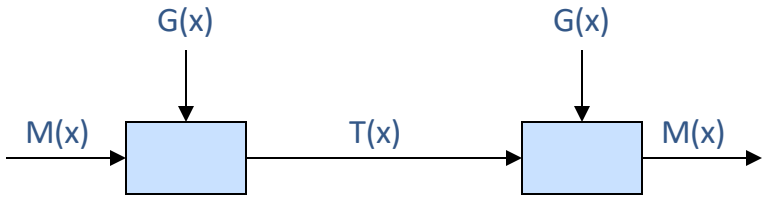
    Nachrichtenwiederholung

M(x)=1101011011

G(x)=x^4+x+1=10011

T(x)=11010110111110

```
      10011
       10011
       10011
       0000010111
             10011
             0010011
               10011
               000000 –> OK
```



Fehlererkennung: CRC: G(x)

- Typische Generatorpolynome:
  - CRC-12: $x^{12} + x^{11} + x^3 + x^2 + x + 1$
    - geeignet für Zeichenlängen von 6 Bit
  - CRC-16: $x^{16} + x^{15} + x^2 + x + 1$
    - geeignet für Zeichenlängen von 8 Bit
  - CRC-CCITT: $x^{16} + x^{12} + x^5 + 1$
    - geeignet für Zeichenlängen von 8 Bit

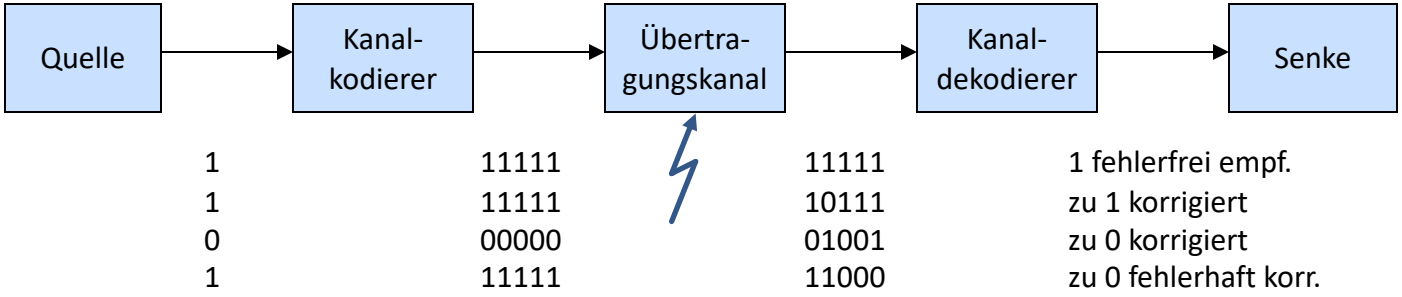
CCITT: Comite Consulatif International Telegraphique et Telephonique, später ITU-T (International Telecommunications Union)

- Leistungsfähigkeit von CRC mit G(x) vom Grad r
  - Einzelfehlererkennung
  - Erkennung aller Doppelfehler mit einem Abstand von k Bit, falls G(x) nicht den Faktor (x^k+1) enthält (k=2...Framelänge)
  - Erkennung aller Fehler mit ungerader Anzahl Fehlerbits, da G(x) den Faktor (x+1) enthält
  - Erkennung von Burst-Fehlern <= r Bit

Fehlerkorrektur

- Motivation:
  - Nachrichtenwiederholung (Retransmission) ist teuer.
  - Nicht immer ist Retransmission möglich.

- Die Anfänge:
  - Repetition code (Wiederholungscode)
    - zusätzliche Bits werden zu Quellbits hinzugefügt



Fehlerkorrektur: Beispiel Kreuzsicherungsverfahren

- zweidimensionale Parität
- Korrektur von Einzelfehlern

Paritätsbits

1	0	1	1	0	1	0
0	1	1	0	1	1	0
1	1	1	0	0	1	0
1	0	1	1	1	0	0
1	0	1	1	0	0	1
0	0	1	0	0	1	0
0	0	0	1	0	0	1

Paritätsbits

Datenverlust

1	0	1	1	1	0
0	1	1	0	1	0
1	1	1	0	1	0
1	0	1	1	0	1
0	0	1	0	1	0
0	0	0	1	0	1

Datenverlust

Fehlerkorrektur: Beispiel Lineare Blockcodes

- Gleichungssystem von Paritätsprüfungen

  x_i : Quellbit

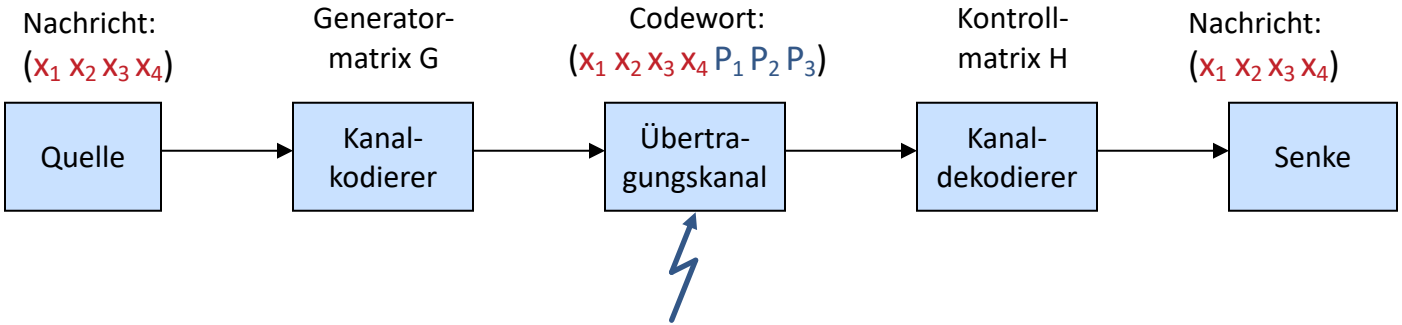
  P_j : Paritätsbit

- Beispiel: (4,7)-Code

$$x_1+x_2+x_3 \quad +P_1 = 0$$

$$x_2+x_3+x_4 \quad +P_2 = 0$$

$$x_1 \quad +x_3+x_4 \quad +P_3 = 0$$



Fehlerkorrektur: Beispiel Lineare Blockcodes

- Beispiel: (4,7)-Code

$$x_1+x_2+x_3 \quad +P_1 \quad = 0$$
$$x_2+x_3+x_4 \quad +P_2 \quad = 0$$
$$x_1 \quad +x_3+x_4 \quad +P_3 = 0$$

- Position eines Einfachfehlers wird erkennbar

Die mit • markierten Positionen einer Zeile hängen voneinander ab.

Empfang von (1 1 1 0 0 0 0):

```
   1 2 3 4 5 6 7                          1 2 3 4 5 6 7
 1 • • •   •       failed             1 • • •   x     Bit 5 gestört
 2   • • •   •     OK        -->      2   • • •   •
 3 •   • •     •   OK                 3 •   • •     •
```

Empfang von (1 1 1 0 0 1 0):

```
   1 2 3 4 5 6 7                          1 2 3 4 5 6 7
 1 • • •   •       failed             1 • x •   •     Bit 2 gestört
 2   • • •   •     failed    -->      2   x • •   •
 3 •   • •     •   OK                 3 •   • •     •
```

Lineare Blockcodes

- Beispiel: (4,7)-Code

$$
\begin{aligned}
x_1+x_2+x_3 \quad +P_1 \quad &= 0\\
x_2+x_3+x_4 \quad +P_2 \quad &= 0\\
x_1 \quad +x_3+x_4 \quad +P_3 &= 0
\end{aligned}
$$

- Prüfen eines Codeworts (Fehlersyndrom s berechnen)

$$s = H \cdot (x_1\ x_2\ x_3\ x_4\ P_1\ P_2\ P_3)^T \qquad s = \begin{bmatrix} \circ \\ \circ \\ \circ \end{bmatrix} = 0$$

bedeutet OK, sonst Fehler

- Kontrollmatrix H

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Beispiel:

Empfang von (1 1 1 0 0 0 0):

$$s = \begin{bmatrix} 1110100 \\ 0111010 \\ 1011001 \end{bmatrix} \cdot (1110000)^T = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Lineare Blockcodes

- Beispiel: (4,7)-Code

$$
\begin{aligned}
x_1+x_2+x_3 \quad +P_1 \quad &= 0\\
x_2+x_3+x_4 \quad +P_2 \quad &= 0\\
x_1 \quad +x_3+x_4 \quad +P_3 &= 0
\end{aligned}
$$

- Erzeugen eines Codeworts

$$(x_1\ x_2\ x_3\ x_4) \cdot G = (x_1\ x_2\ x_3\ x_4\ P_1\ P_2\ P_3)$$

- Generatormatrix G (Es gilt $G \cdot H^T = 0$)

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = IC$$

(Einheitsmatrix konkateniert mit C)

Fehlerkorrektur: Beispiel Lineare Blockcodes

- Beispiel: (4,7)-Code

$$x_1+x_2+x_3 \quad +P_1 \quad = 0$$
$$x_2+x_3+x_4 \quad +P_2 \quad = 0$$
$$x_1 \quad +x_3+x_4 \quad +P_3 = 0$$

- Quellcodeworte Kanalcodeworte

0000	0000000	0110	0110001
1000	1000101	0101	0101101
0100	0100110	0011	0011100
0010	0010111	1110	1110100
0001	0001011	1101	1101000
1100	1100011	1011	1011001
1010	1010010	0111	0111010
1001	1001110	1111	1111111

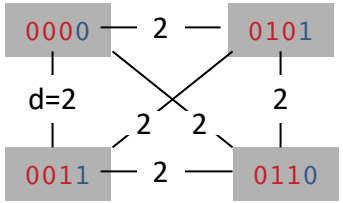
Beobachtung: Gültige Kodewörter unterscheiden sich stets in (wenigstens) 3 Stellen.

Kenngrößen von Kanalcodes

- (l,n)-Code
  - l Informationsstellen
  - k Kontrollstellen
  - n = l+k
- Hamming-Abstand (Hamming-Distanz) d zweier Codewörter
  - Anzahl der Stellen, in denen sich zwei Codewörter unterscheiden
  - Beispiel: (010) und (100) haben d=2
- Minimaler Hamming-Abstand eines Codes d_{min}
  - Minimalwert aller Hamming-Abstände aller Codewörter
  - Es gilt:
    - Erkennbarkeit: $d_{min} = f_e+1$
    - Korrigierbarkeit: $d_{min} = 2f_k+1$

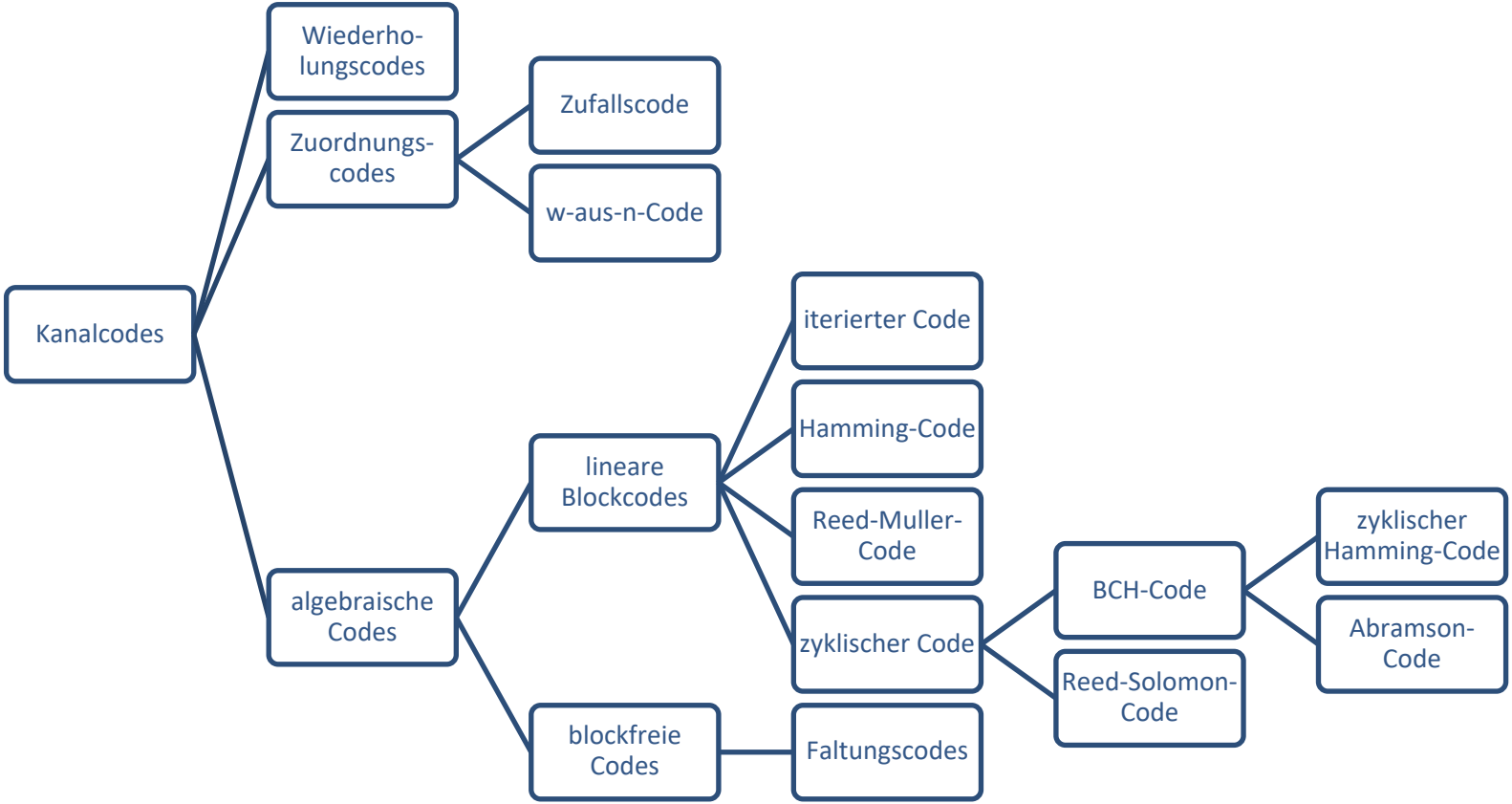
    mit
    - f_e : Anzahl erkennbarer Fehler
    - f_k : Anzahl korrigierbarer Fehler

Beispiel Parität:
(x_1 x_2 x_3 P)



d_{min}=2

Klassifizierung bekannter Kanalcodes



Codes

- Wiederholungscode (repetition code)
  - Bit n mal wiederholen $d_{min}=n$

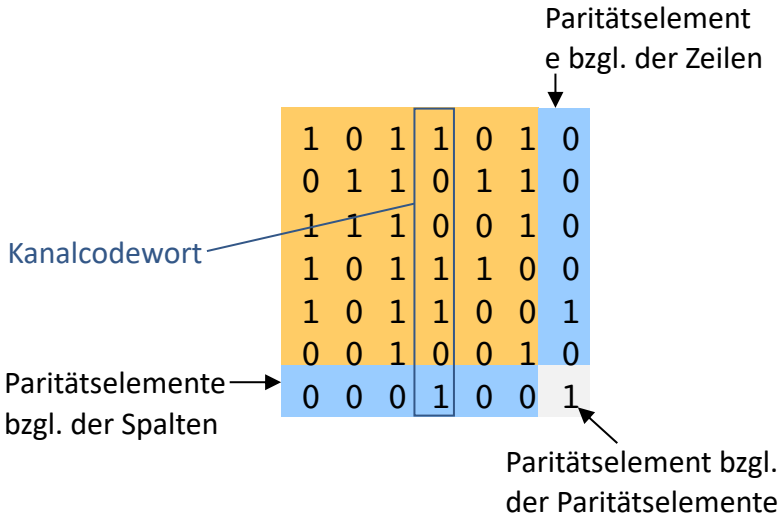
- Zufallscodes
  - Zufällige Zuordnung von Quell- und Kanalcodeworten
  - keine praktische Bedeutung

- w-aus-n-Code
  - alle Codewörter haben das gleiche Gewicht (Anzahl der Einsen) im Kanalcodewort
  - Bsp: 2-aus-5-Code mit $d_{min}=2$: (11000), (10100), (10010), (10001), (01100), (01010), (01001), (00110), (00101), (00011)

$$\binom{n}{w} \text{Codewörter}$$

Codes

- Algebraische Codes
  - Bildung der Kanalcodewörter erfolgt nach einer algebraischen Vorschrift
- Lineare Blockcodes
  - Blockcode: Kanalcodewörter haben eine feste Länge
  - linear: Kodewörter und Verknüpfungsregeln erfüllen die Eigenschaften einer Gruppe
- Iterierte Codes
  - binäre Paritätscodes
  - Kreuzsicherungsverfahren (d_{min}=4)



VSS 17.04.2020

Codes

- Hamming-Code
  - linearer Blockcode, besonders geeignet zur Einfehlerkorrektur
  - angewendet z.B. bei Hauptspeicher
- Reed-Muller-Code
  - linearer Blockcode, besonders geeignet bei Mehrfachfehlern
  - bei Mariner Expeditionen zum Mars 1969-1975 mit d_{min}=16 genutzt
- Zyklische Codes (cyclic codes)
  - besonders geeignet zur Erkennung und Korrektur von Fehlerbündeln
  - bei Satellitenübertragung und bei CD angewendet
- Blockfreie Codes (convolutional codes), auch Faltungscodes
  - Redundanz wird kontinuierlich durch Faltung der Information, die der Quellenkodierer abgibt, eingefügt
  - keine Kodewortgrenzen
  - angewendet z.B. im Mobilfunk

Sicherungsschicht

- Aufgaben
  - Framing
  - Fehlersicherung
  - Übertragungssteuerung

- Beispiele für Schicht-2-Dienste
  - Local Area Networks: Ethernet
  - Wireless LANs

A	(Application Layer)
P	(Presentation Layer)
S	(Session Layer)
T	(Transport Layer)
N	(Network Layer)
DL	(Data Link Layer)
PH, PHY	(Physical Layer)

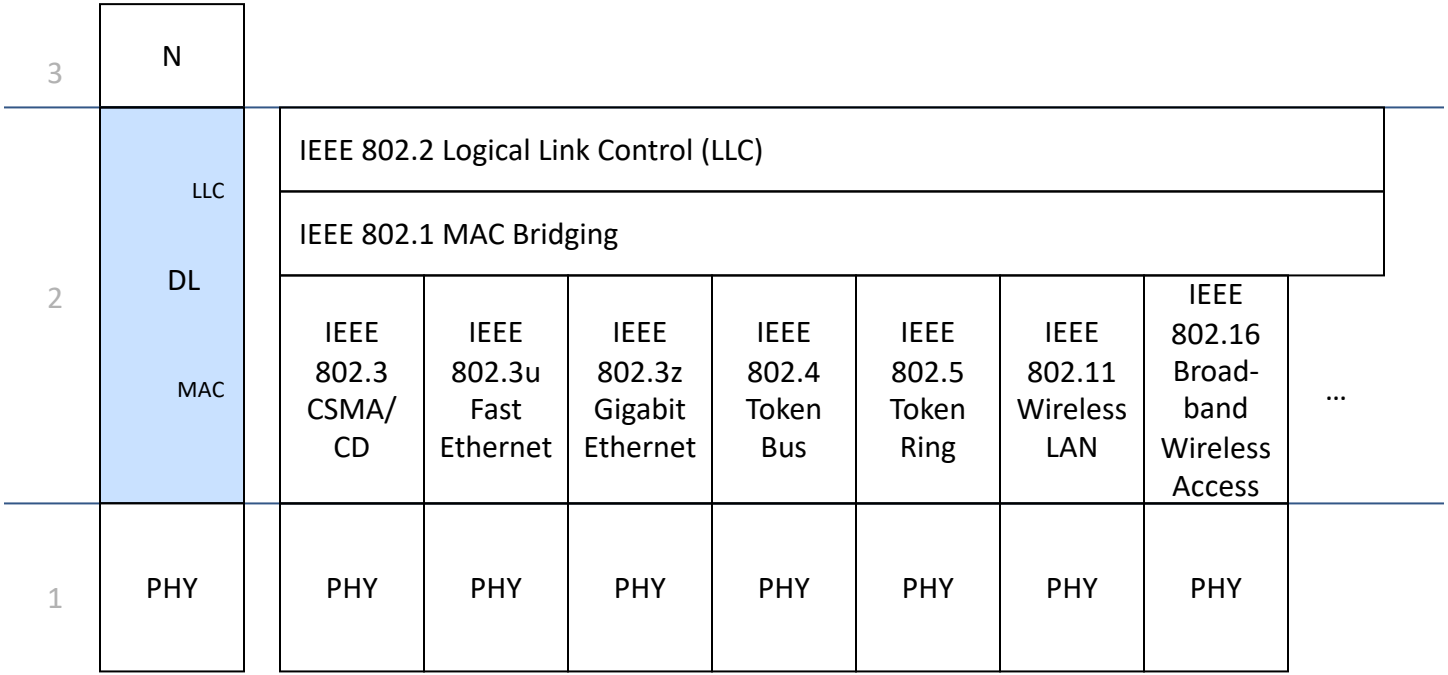
Local Area Networks (LAN)

- Merkmale
  - territorial beschränkte Ausdehnung
  - besitzt mindestens die Funktionalität der Schichten 1 und 2

- Logisch funktionale Unterteilung der beiden unteren Schichten
  - LLC-Subschicht (Logical Link Control)
    - Fehlersicherung, -feststellung und -korrektur
  - MAC-Subschicht (Medium Access Control)
    - Steuerung des Zugriffs der Teilnehmerstationen auf das Medium

- Übertragungsmodus
  - meist Basisbandübertragung

Standards der IEEE-802-Familie (Auswahl)



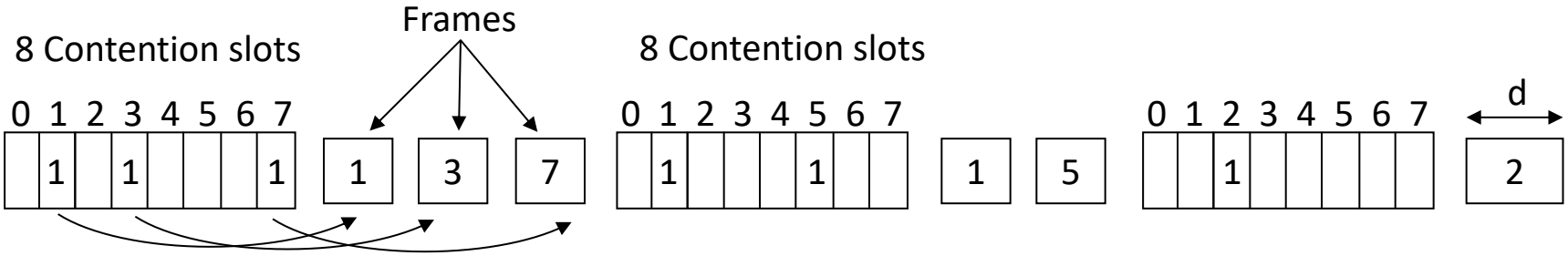
MAC-Sublayer: Medienzugriffsverfahren

- Problem
  - mehrere Stationen senden auf gleichem Medium

- Kollisionsvermeidung
  - Bit-Map-Protocol (Bitmusterprotokoll)
  - Binary Countdown
- Kollisionserkennung
  - ALOHA
    - Pure ALOHA
    - Slotted ALOHA
  - Carrier Sense Multiple Access
    - p-persistent
    - non-persistent
    - with Collision Detection (CSMA/CD)

Kollisionsfrei: Bitmusterprotokoll

- Das Bitmusterprotokoll ist ein Reservierungsprotokoll.
- Verfahren
  - Jeder Station ist ein *bestimmtes* Bit eines Contention slots zugeordnet. Wenn Station ein Frame zu senden hat, meldet sie es im Contention slot mit einer 1 an. Alle Stationen wissen jetzt, welche Stationen senden wollen. Frames werden in der Reihenfolge der Anmeldung gesendet.
- Beispiel für N=8 Stationen

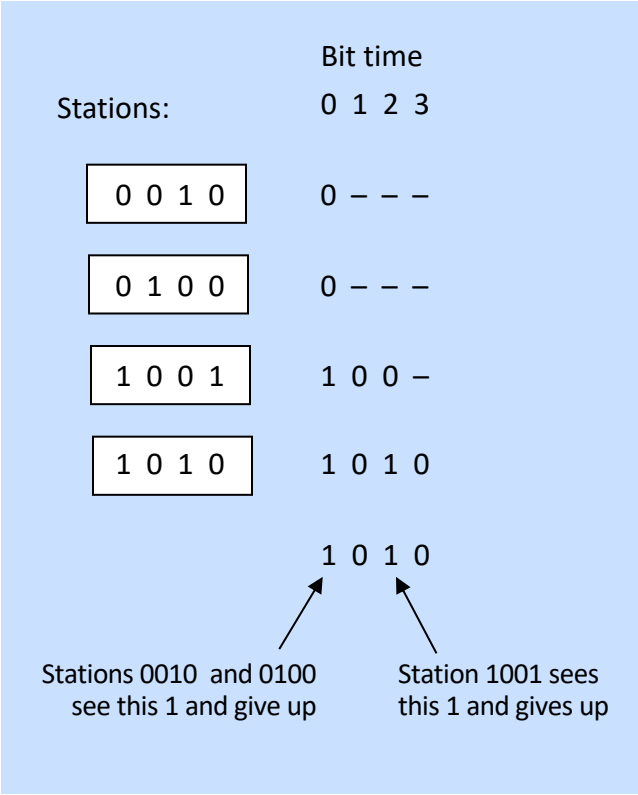


- Effizienz: d/(N+d)
  - d: Frame-Länge in Bit
  - N: Anzahl der Stationen

Kollisionsfrei: Binärer Countdown

- Bitweises Überlagerungsprotokoll
  - Überlagerungsfunktion: bitweise ODER-Verknüpfung

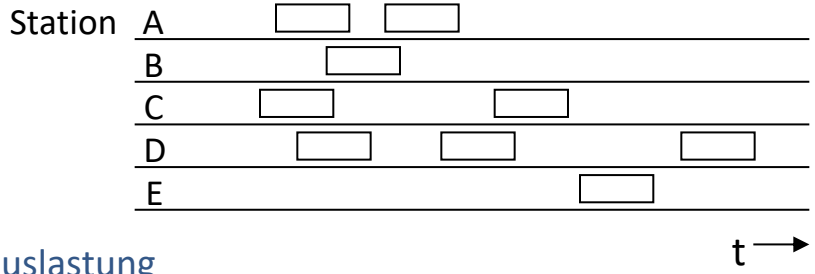
- Verfahren:
  - Jede Station, die ein Frame senden möchte, sendet bitweise ihre Adresse, beginnend mit dem MSB und hört das Ergebnis der Überlagerung ab
  - Wenn sie eine 0 gesendet hat, aber eine 1 empfängt, gibt sie auf und wartet auf die nächste Runde



Pure ALOHA (Abramson, ca. 1970)

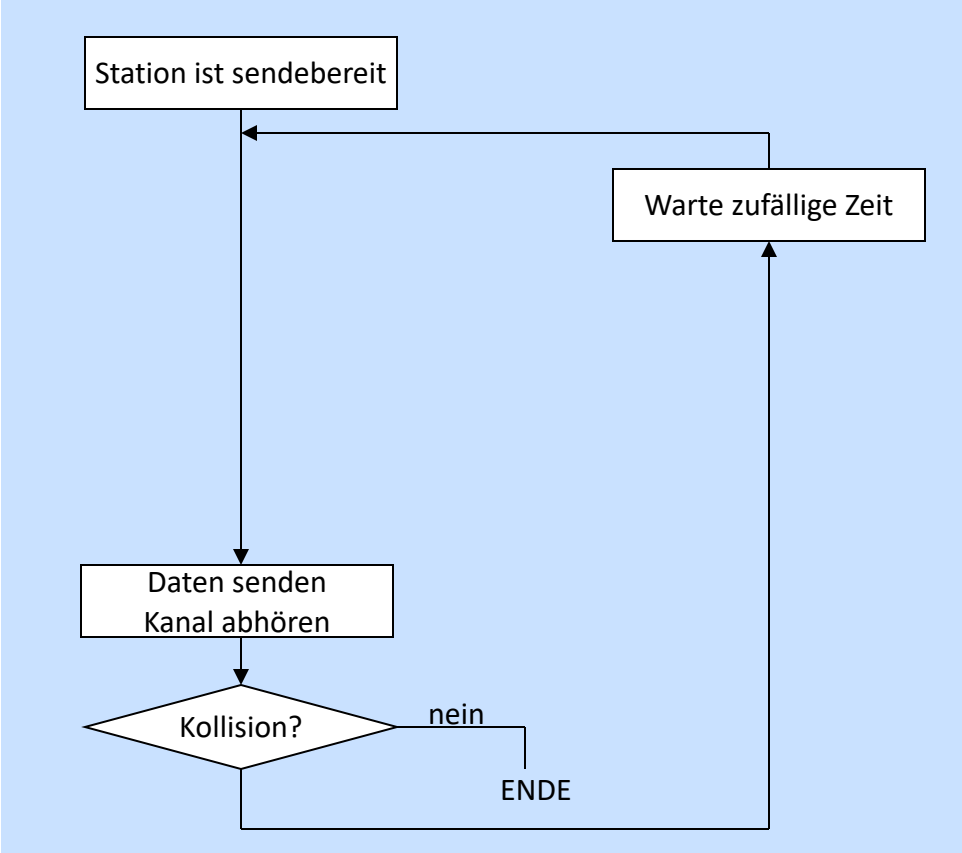
- Warum überhaupt Verzicht auf Reservierung?
  - Koordination zwischen den Stationen zu aufwendig oder nicht möglich
  - Koordination erfordert u.U. zu viel Zeit: z.B. bei Satellitenübertragung Verzögerung von 270 ms, bis Sender weiß, ob er aufgeben muss

- Verfahren Pure ALOHA
  - Station beginnt zu senden und hört anschließend Kanal ab, ob eigenes Frame korrekt empfangen wird
  - wenn Kollision: Warte eine zufällige Zeit und versuche es erneut.



- Kanalauslastung
  - durchschnittlich etwa 18 Prozent des maximal möglichen Durchsatzes

Pure ALOHA


Station ist sendebereit
Warte zufällige Zeit
Daten senden
Kanal abhören
Kollision?
nein
ENDE

Effizienz von Pure ALOHA

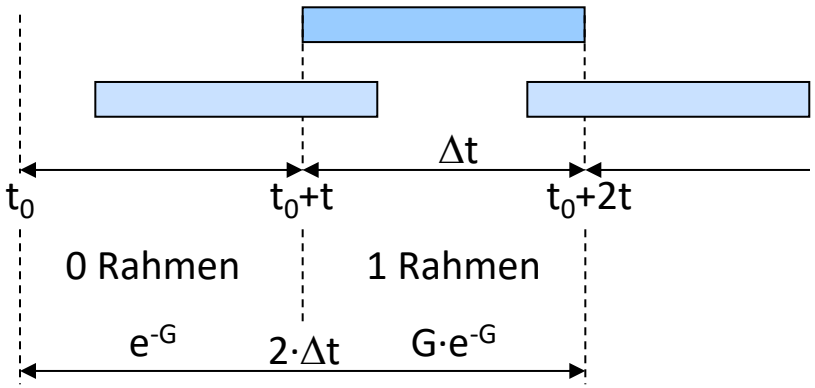
- Annahmen
  - Alle Frames haben gleiche Länge l in bit
  - Kanal mit Bitübertragungsgeschwindigkeit $v_Ü$ in bit/s
    - Rahmenübertragungszeit ist damit $t=l/v_Ü$ in s
  - Benutzer erzeugen Rahmen nach Poisson-Verteilung mit mittlerer Anzahl G von Rahmen pro Δt (Anzahl der Teilnehmer sei unendlich; nötig, damit G nicht abnimmt, wenn Teilnehmer blockiert sind)
- Poisson-Verteilung:
  - Wkt., dass k Rahmen innerhalb Δt produziert werden: $$\Pr[k] = \frac{G^k e^{-G}}{k!}$$
- Effizienzmaß: relativer Datendurchsatz S
  - Vorüberlegung: Effizienz ist maximal (S=100 %), wenn jeweils pro Δt ein Rahmen fehlerfrei übertragen wurde
  - Tatsächlich: S ergibt sich aus der Wkt., dass pro Δt ein Rahmen fehlerfrei übertragen wird

Effizienz von Pure ALOHA

- Poisson-Verteilung

$$\Pr[k] = \frac{G^k e^{-G}}{k!}$$

- Bedingung für fehlerfreie Übertragung innerhalb Δt
  - innerhalb einer Zeitspanne von $2 \cdot \Delta t$ darf nur ein einziger Rahmen gesendet werden



$2 \cdot \Delta t$ ist deshalb anzusetzen, weil alle Teilnehmer unabhängig voneinander Rahmen erzeugen.

  - Wahrscheinlichkeit, dass genau k=1 Rahmen und genau k=0 Rahmen in $2 \cdot \Delta t$ erzeugt werden:

$$S = G \cdot e^{-2G}$$

VSS 17.04.2020

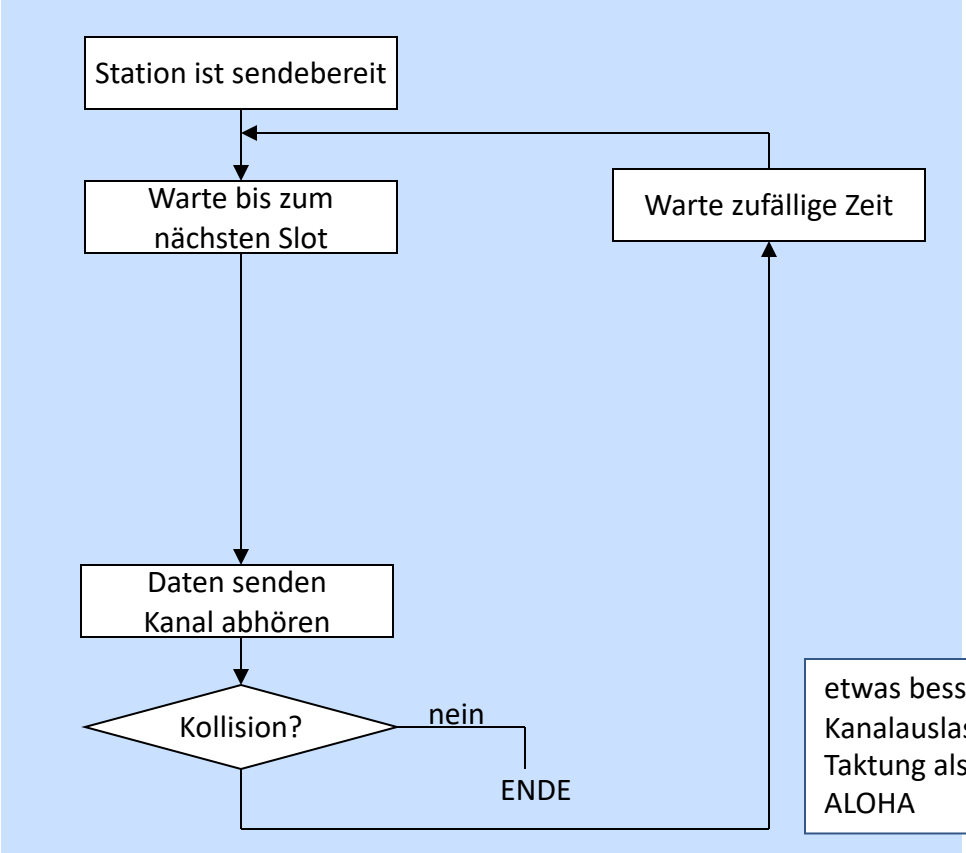
Slotted ALOHA (Robertson, 1972)

- System erfordert globale Synchronisation
  - eine spezielle Station sendet ein Taktsignal, das den Kanal in Slots einteilt
- Verfahren
  - warte bis zum nächsten Slotbeginn, dann sende
  - wenn Kollision, warte eine zufällige Zeit
- Kanalauslastung
  - erreicht durchschnittlich etwa 37 Prozent Durchsatz, da sich Zeitspanne, innerhalb derer keine zweite Station senden darf, auf Δt verkürzt

$$S = G \cdot e^{-G}$$

  - S-ALOHA erreicht Spitzenwert der Kanalauslastung bei G=1:
    - 37 % leere Zeitschlitze (k=0), 37 % erfolgreiche Übertragungen (k=1), 26 % Kollisionen
  - Mit wachsendem Verkehr steigt die Kollisionswahrscheinlichkeit exponentiell an.

Slotted ALOHA



etwas bessere Kanalauslastung durch Taktung als Pure ALOHA

Carrier Sense Multiple Access-Protokolle

- Problem von ALOHA:
  - Stationen hören nicht den Kanal ab, bevor sie mit Sendung beginnen
- Anders bei Carrier Sense Protokollen:
  - Stationen hören Träger (bzw. andere Übertragungen) ab, bevor sie selbst senden

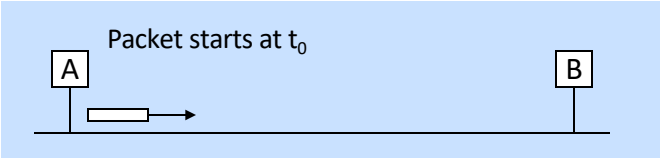
- 1-persistent CSMA
  1. Station hört Kanal ab
  2. wenn frei, sendet sie
  3. wenn Kollision, wartet sie eine zufällige Zeit und geht zu 1.

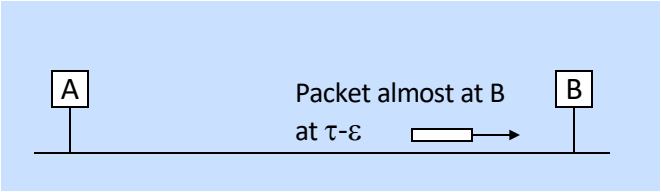
- Warum kann überhaupt noch eine Kollision auftreten?
  - Zwei Stationen könnten *gleichzeitig* »Kanal frei« erkennen.
  - Ausbreitungsverzögerung des Signals –>

Problem der Ausbreitungsverzögerung

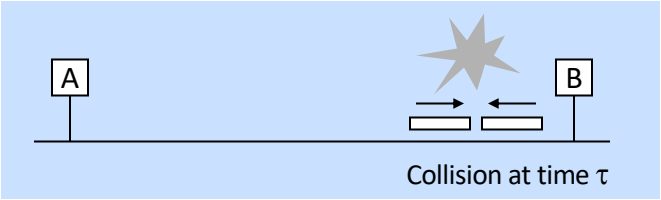
Problembeschreibung:

- Station A sendet zu Zeitpunkt t_0
- Verzögerungszeit bis zum Eintreffen des Signals bei der entferntesten Station: τ
- Station B sendet zu Zeitpunkt $\tau-\varepsilon$ (mit ε nahe 0)
- Station A erkennt Kollision erst nach nach 2τ



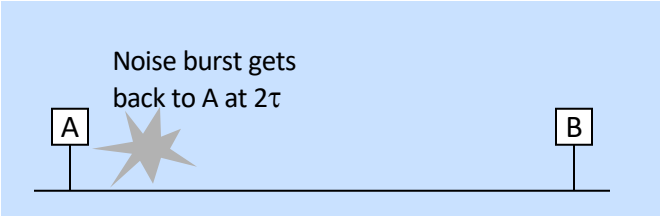


Es existiert eine geringe Wahrscheinlichkeit, dass eine Station innerhalb der Verzögerungszeit zu senden beginnt.



Typische Ausbreitungsverzögerungen (Ethernet):

- 1 km langes Kabel: $\tau \approx 5\ \mu s$
- 2,5 km Kabel, 4 Repeater: $\tau \approx 50\ \mu s$



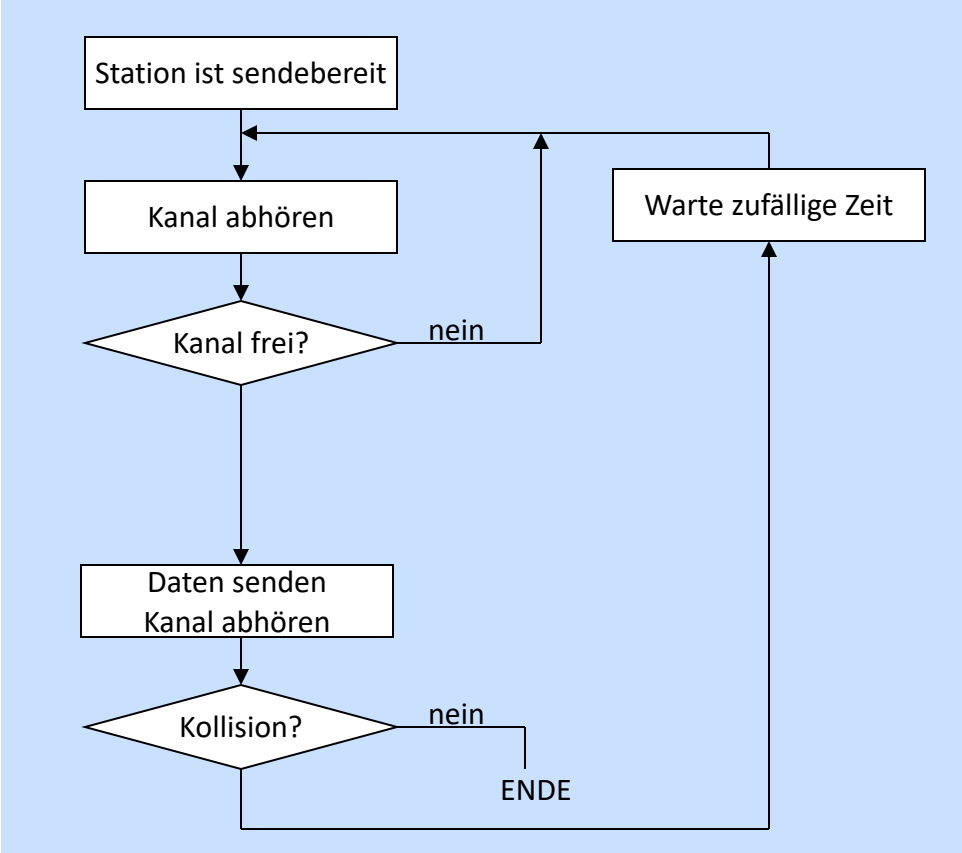
Kollisionsproblematik

- Problem der Ausbreitungsverzögerung
  - Verzögerungszeit zwischen Senden eines Signals und Eintreffen bei der letzten Station
  - geringe Wahrscheinlichkeit, dass eine Station innerhalb der Verzögerungszeit zu senden beginnt.
  - Folge: Kollision
- Selbst wenn Ausbreitungsverzögerung == 0
  - Zwei Stationen fangen nach erkanntem Sendeende gleichzeitig zu senden an (bei 1-persistent CSMA)
  - Folge: Kollision
- Möglichkeiten zur Verringerung von Kollisionen
  1. Stationen warten auch bei »Kanal belegt« eine zufällige Zeit
  2. Stationen senden nur mit einer Wahrscheinlichkeit p, nachdem Kanal frei geworden ist

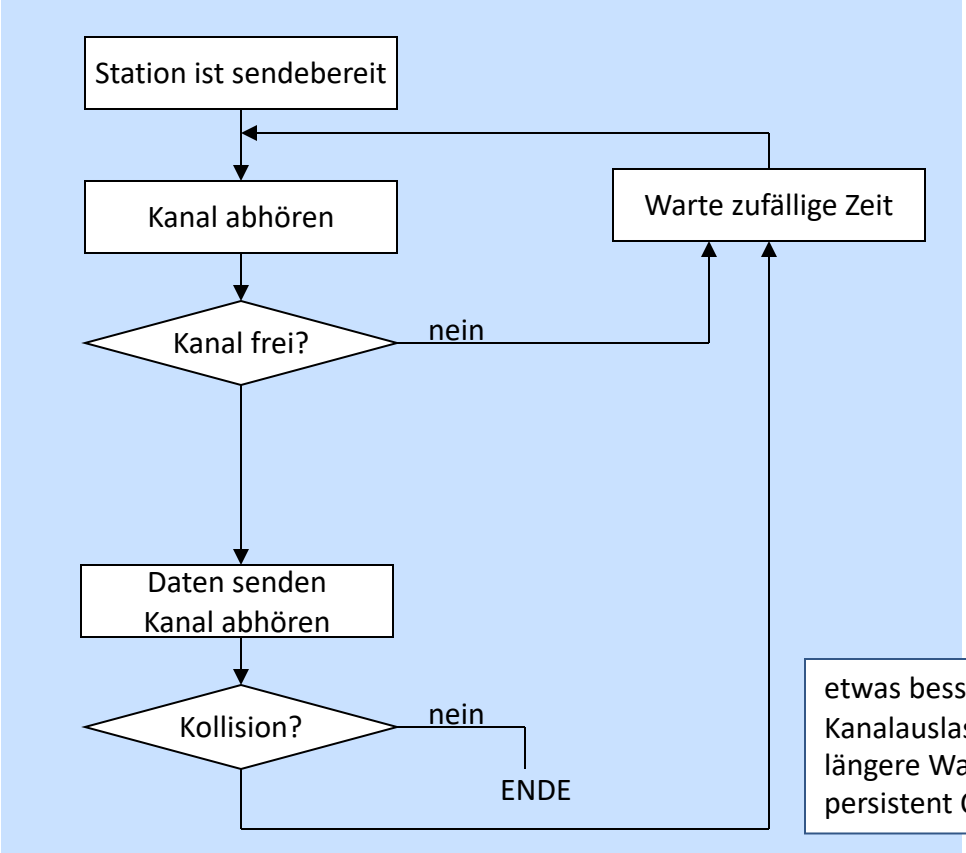
Möglichkeiten zur Verringerung von Kollisionen

- Non-persistent CSMA:
  - Stationen warten auch bei »Kanal belegt« eine zufällige Zeit
    1. Station S hört Kanal ab
    2. Wenn frei, sendet sie. Sonst: S wartet zufällige Zeit und geht zu 1.
    3. Wenn Kollision, wartet sie eine zufällige Zeit und geht zu 1.
- p-persistent CSMA (für Kanäle mit Zeitschlitzen):
  - Stationen senden nur mit einer Wahrscheinlichkeit p, nachdem Kanal frei geworden ist
    1. Station S hört Kanal ab
    2. Wenn frei, sendet sie mit einer Wahrscheinlichkeit p.
       Mit 1-p wartet sie auf den nächsten Slot und geht zu 1.
    3. Wenn nicht frei o. Kollision: S wartet zufällige Zeit und geht zu 1.
- Collision Detection (CD):
  - Stationen hören sofort auf zu senden, wenn sie eine Kollision entdecken

1-persistent CSMA

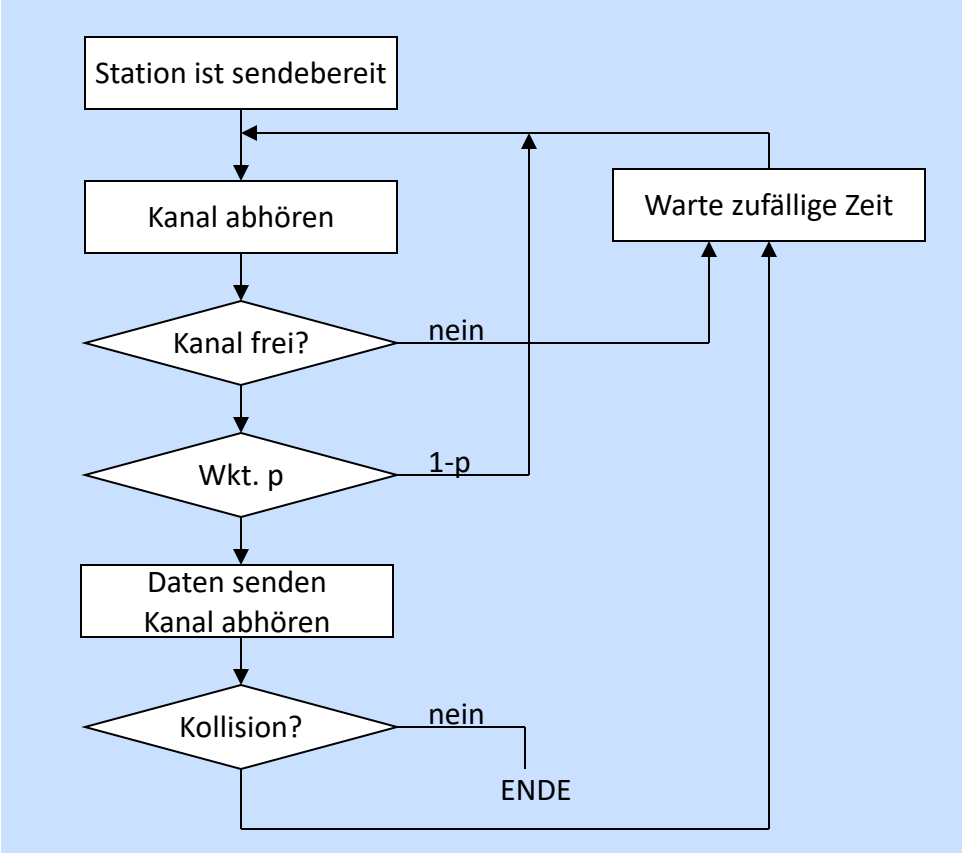


non-persistent CSMA

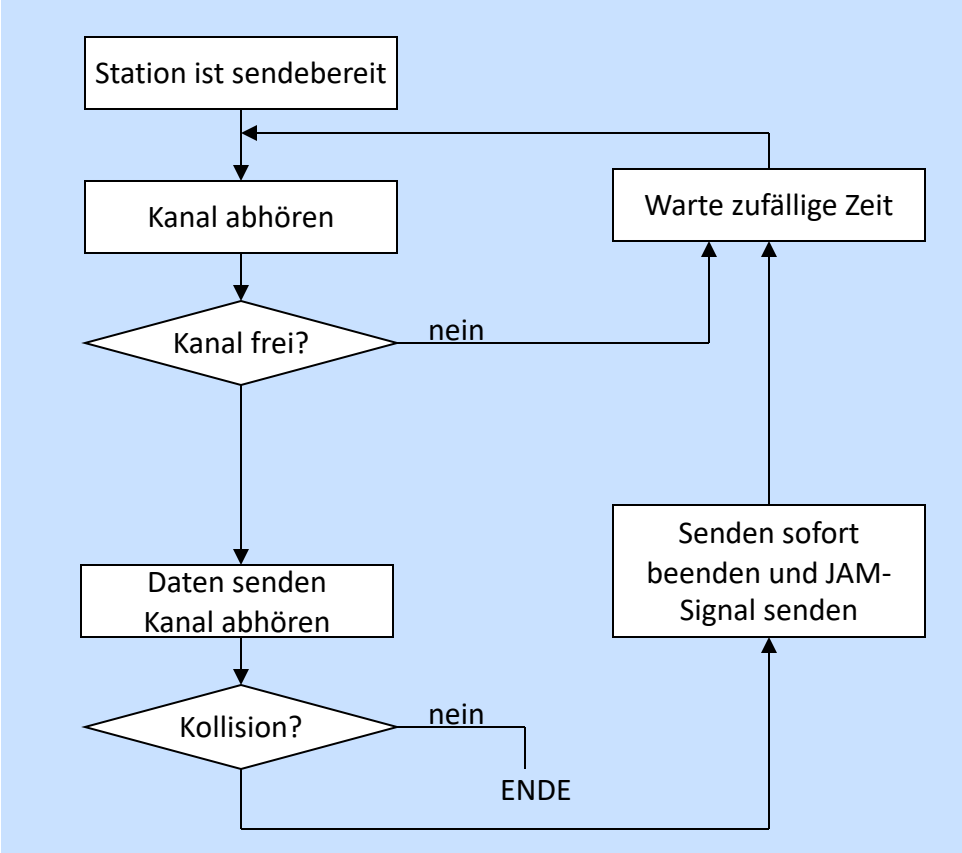


etwas bessere Kanalauslastung, aber längere Wartezeit als 1-persistent CSMA

p-persistent CSMA

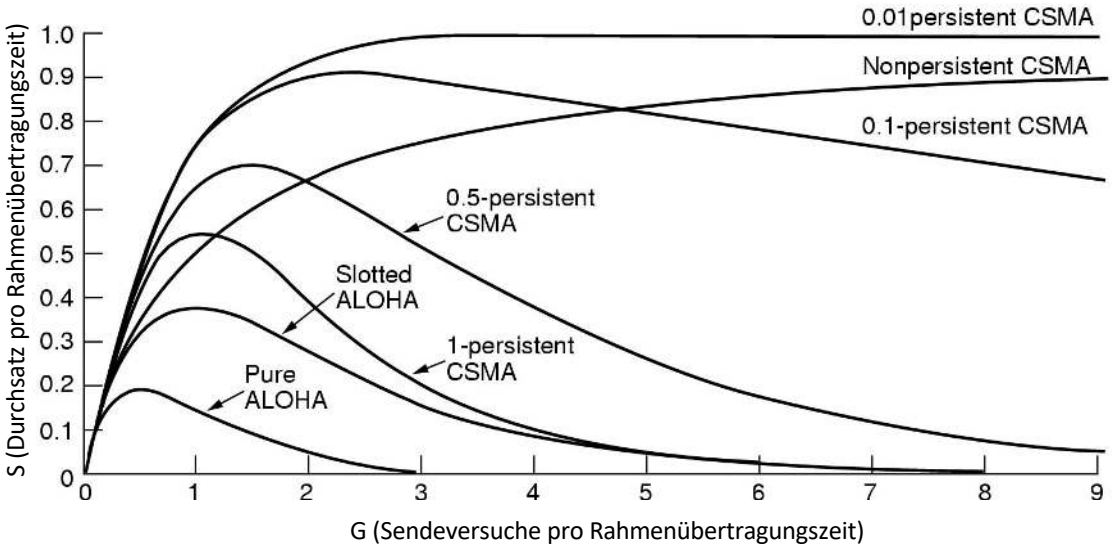


CSMA/CD


Station ist sendebereit
Kanal abhören
Warte zufällige Zeit
Kanal frei?
nein
Daten senden
Kanal abhören
Senden sofort
beenden und JAM-
Signal senden
Kollision?
nein
ENDE

Praktische Aspekte

- Vergleich Kanalauslastung



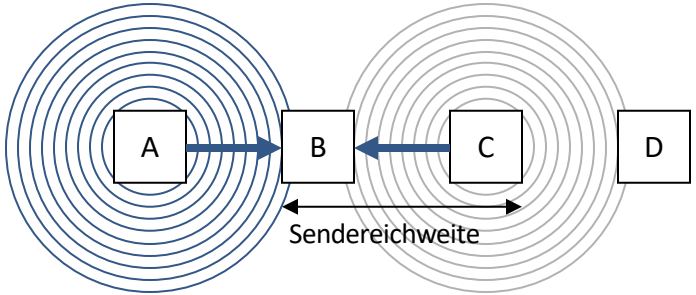
- Im Ethernet wird CSMA/CD verwendet.
- Frage: Wie lange sollte nach einer Kollision gewartet werden?
  - Binary exponential backoff algorithm –>

Binary exponential backoff algorithm

- Wie lange sollte nach einer Kollision gewartet werden?
  - Wartezeit in Slots:
    - Wähle eine Zufallszahl z im Bereich $0 \leq z \leq 2^i-1$
      i: Anzahl der Kollisionen
  - Nach 10 Kollisionen wird die maximale Wartezeit auf $2^{10}-1$ =1023 Slots eingefroren.
  - Nach 16 Kollisionen wird aufgegeben (–> Fehlermeldung)

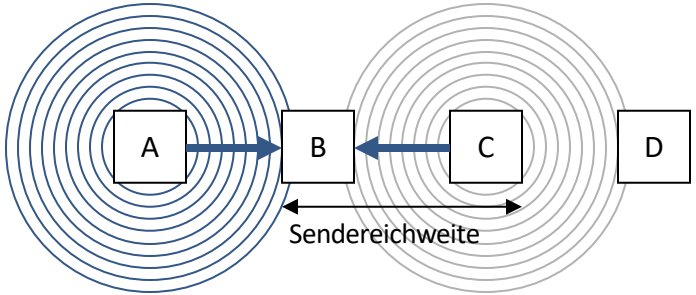
Wirless LANs benötigen spezielle MAC-Subschicht

- Funktioniert Carrier Sense Multiple Access (CSMA)?
  - Ausgangssituation:
    - Nur benachbarte Stationen (egal, ob mobile Computer oder Basisstationen) sind jeweils in Reichweite
  - Fall 1: Hidden station problem
    - A möchte an B senden und hört das Medium ab
    - A erkennt: Medium frei und beginnt zu senden
    - C möchte an B senden und hört das Medium ab
    - C ist außerhalb As Reichweite und erkennt Medium ebenfalls als frei

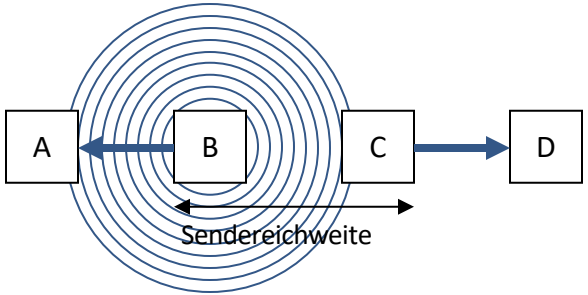


Wirless LANs benötigen spezielle MAC-Subschicht

- Funktioniert Carrier Sense Multiple Access (CSMA)?
  - Ausgangssituation:
    - Nur benachbarte Stationen (egal, ob mobile Computer oder Basisstationen) sind jeweils in Reichweite
  - Fall 1: Hidden station problem
    - Störung tritt beim Empfänger auf, wird aber möglicherweise vom Sender nicht erkannt

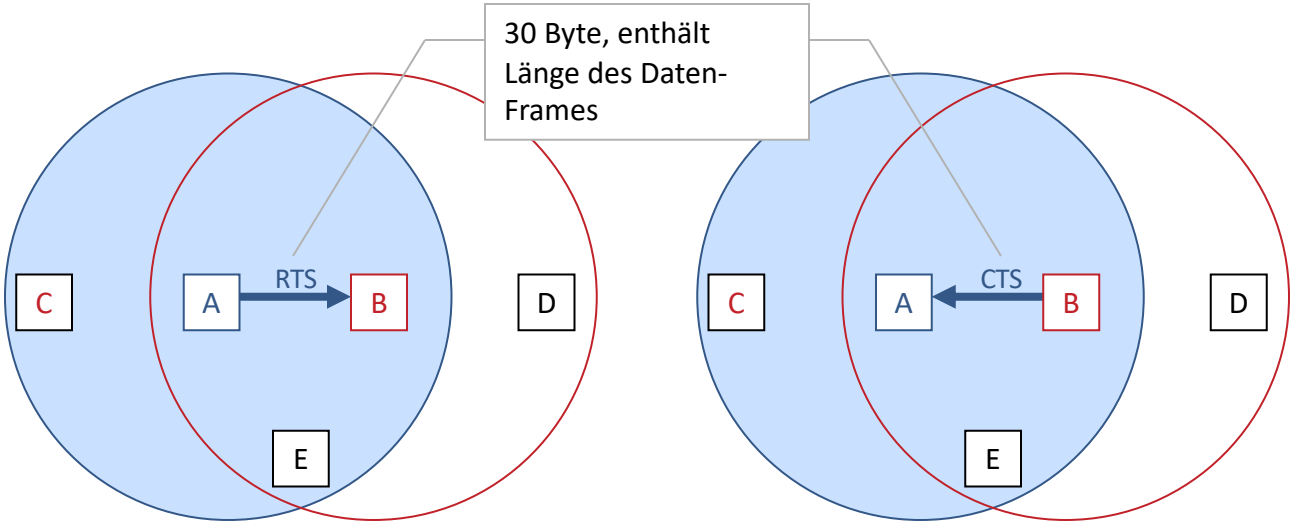


- Funktioniert Carrier Sense Multiple Access (CSMA)?
  - Fall 1: Hidden station problem
    - Störung tritt beim Empfänger auf, wird aber möglicherweise vom Sender nicht erkannt
  - Fall 2: Exposed station problem
    - B sendet an A
    - C möchte an D senden und hört das Medium ab
    - C erkennt das Medium fälschlich als belegt und wartet

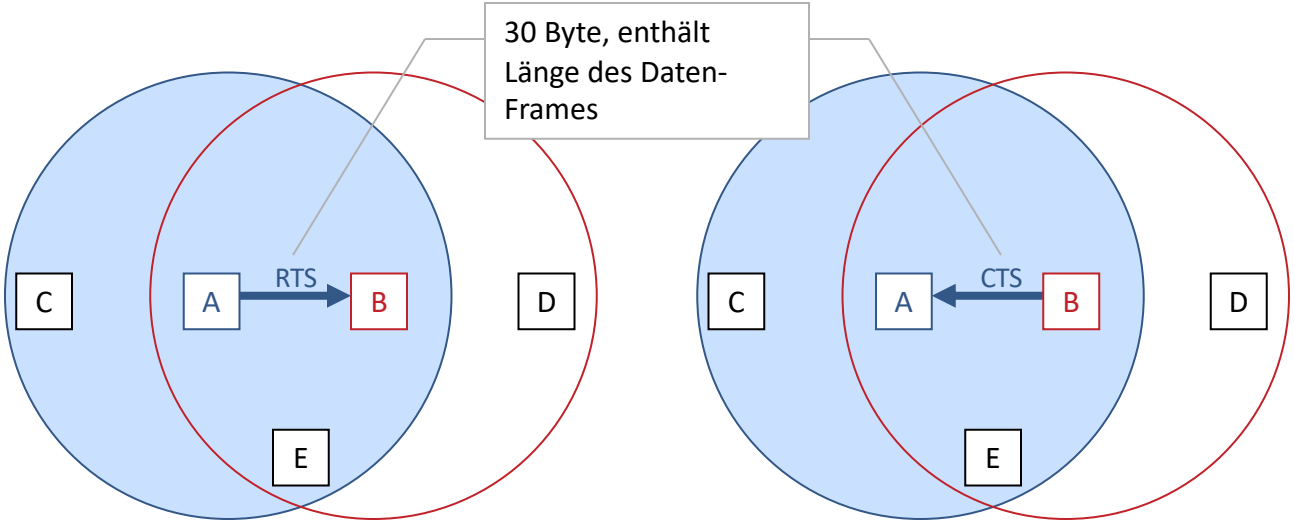


- Generell:
  - Sender muss sich vergewissern, ob der Empfänger frei ist oder nicht

- Multiple Access with Collision Avoidance (MACA)
  - angewendet im IEEE-802.11-Standard (WLAN)
- Verfahren
  - Sender sendet kurzes (30-Byte) RTS-Frame (Request To Send)
  - Empfänger bestätigt durch CTS-Frame (Clear To Send)



- Multiple Access with Collision Avoidance (MACA)
  - angewendet im IEEE-802.11-Standard (WLAN)
- RTS/CTS-Verfahren verhindert jedoch Kollisionen nicht vollständig
  - Zwei Stationen könnten gleichzeitig RTS-Frame senden
  - Beim Ausbleiben von CTS: Warte zufällige Zeit und probiere es wieder
    - Binary exponential backoff algorithm



Verbesserungen von MACA

- Power Aware Multi-Access Protocol with Signaling Ad Hoc Networks (PAMAS)
  - nicht sendewillige Stationen schalten nach Empfang eines RTS-Frames für die Dauer der (anderen) Übertragung ab –> soll Stromverbrauch um bis zu 70 Prozent senken

- Multiple Access with Collision Avoidance for Wireless (MACAW)
  - anwendbar in Wireless Sensor Networks
  - Zusätzliches ACK-Frame für Empfangsbestätigung von Daten-Frames, um Frameverlust bereits auf MAC-Subschicht erkennen zu können
  - Vor Senden eines RTS-Frames wird geprüft, ob Kanal frei (carrier sense)
  - Mechanismus zur Erkennung von Verkehrsstau (congestion)
  - Backoff-Algorithmus arbeitet nicht stationsbezogen, sondern bezogen auf ein Sender-Empfänger-Paar (verbessert Fairness) –> Verbesserung der Reaktionszeit

Network Layer (Schicht 3)

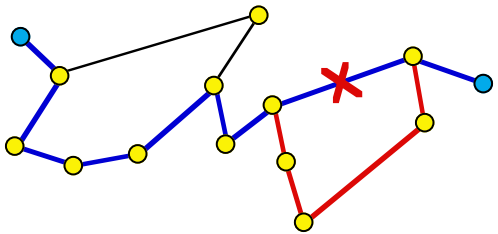
- Aufgaben
  - Bereitstellung von Diensten, die unabhängig vom darunter liegenden lokalen Netz sind
  - Bereitstellung eines einheitlichen Adressierungsschemas über lokalen Netzgrenzen hinweg
  - Routing: Wegewahl zwischen Sender und Empfänger

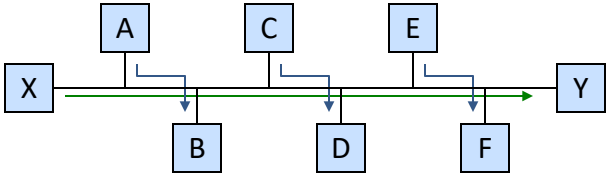
- Beispiele für Routing-Algorithmen
  - Shortest Path Routing
  - Distance Vector Routing
  - Link State Routing

A	(Application Layer)
P	(Presentation Layer)
S	(Session Layer)
T	(Transport Layer)
N	(Network Layer)
DL	(Data Link Layer)
PH, PHY	(Physical Layer)

Routing-Algorithmen

- Anforderungen
  - Einfachheit: Router besitzen wenig Speicher und Rechenleistung, Wegewahl soll wenig Zeit kosten
  - Stabilität: Router sollen lange ohne Neustart laufen
  - Robustheit: Router sollen alternative Wege bei Ausfall einer Route finden
  - Fairness
  - Optimalität





Optimalität: Wenn A, B, C, D, E, F gemeinsam voll auslasten können, sollten X und Y warten

Fairness: X und Y sollten jedoch nicht benachteiligt werden

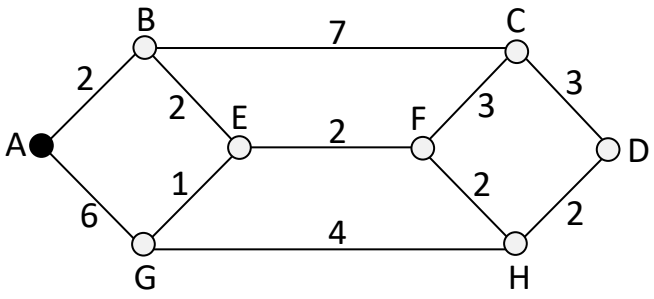
Routing-Algorithmen

- Nichtadaptive Algorithmen (statisches Routing)
  - Route I–J (Weg, der genommen werden muss, um von I nach J zu gelangen) wird im Voraus offline berechnet und beim Hochfahren des Netzes in alle Router geladen
  - Routing-Entscheidung basiert *nicht* auf Messungen oder Schätzungen des momentanen Verkehrs und der aktuellen Topologie

- Adaptive Algorithmen (dynamisches Routing)
  - modifiziere Routing-Entscheidung bei Veränderung der Netz-Topologie
  - Routing-Entscheidung basiert oft auf aktueller Verkehrssituation

Shortest Path Routing

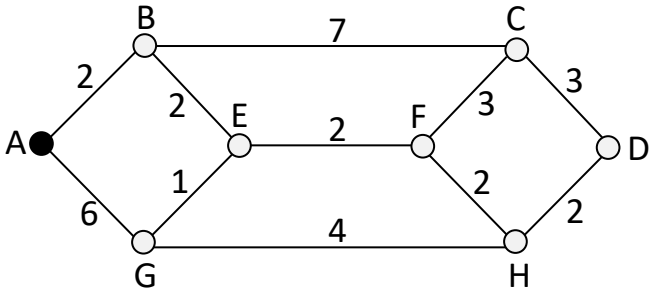
- Modellierung
  - Jeder Knoten ist ein Router
  - Jede bewertete Kante ist eine Kommunikationsleitung
- Kantenbewertung
  - gibt Kosten der Übertragung auf der Kommunikationsleitung an
  - Wahl hängt ab von Optimalitätskriterium, z.B.
    - Geographische Entfernung
    - Verzögerungszeit
  - typisch: Funktion von Entfernung, Bandbreite, durchschnittlicher Verkehrslast, Kommunikationskosten, mittlerer Warteschlangenlänge, Verzögerungszeit, …



(a)

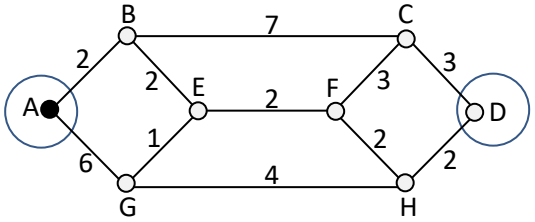
Shortest Path Routing

- Kantenbewertung (Gewicht)
  - ist Entfernung zwischen beiden Knoten
- Algorithmus (Dijkstra, 1959)
  - Alle Knoten erhalten eine Markierung. Zu Beginn sind alle Knoten mit ∞ markiert und im Zustand tentative.
  - Markiere jeden Knoten mit der kürzesten Entfernung vom Quellknoten über den betreffenden Knoten
  - Sobald der kürzeste Pfad für einen Knoten gefunden ist, wird die Markierung in den Zustand permanent versetzt
- Beispiel
  - Gesucht: Kürzester Pfad zwischen A und D

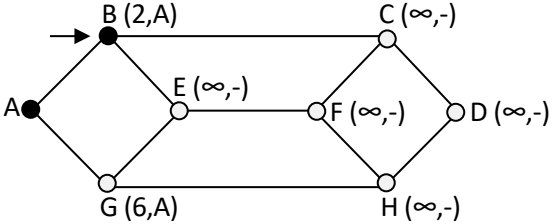


(a)

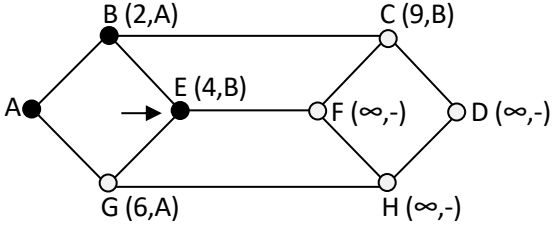
Shortest Path Routing



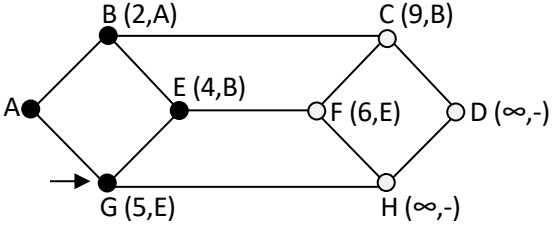
(a)



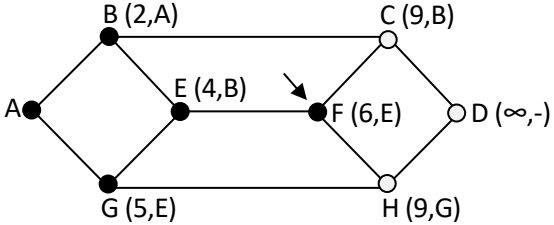
(b)



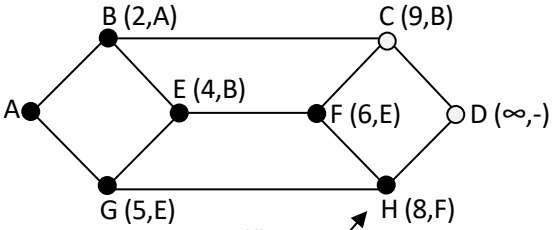
(c)



(d)



(e)



(f)

Distance-Vector-Routing

- dynamischer Routing-Algorithmus
- Vorläufer des heute im Internet eingesetzten Routing-Algorithmus

- Jeder Router verwaltet einen sog. Routing Table
  - Ein Eintrag je Router im Subnetz:
    - erwartete Verzögerungszeit oder Entfernung oder Anzahl der Zwischenstationen (Hops) zum Ziel oder …
    - Bevorzugter Router (Line) für jedes Ziel
  - Router kennt Verzögerungszeit zu jedem Nachbar-Router
  - Nachbar-Router senden alle t msek eine Liste der Verzögerungszeiten zu jedem Ziel

Routing Table for J:

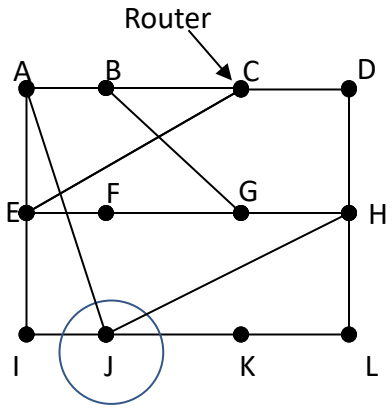
To	Delay	Line(=via)
A	8	A
B	20	A
C	28	I
D	20	H
…	…	…

From A:

To	Delay
A	0
B	12
C	25
D	40
…	…

Distance-Vector-Routing

- Router berechnet aus Verzögerungszeit zu seinen Nachbarn und den erhaltenen Listen der Verzögerungszeiten seinen neuen Routing-Table



To	From A	From I	From H	From K	New estimated delay from J	Line
A	0	24	20	21		
B	12	36	31	28		
C	25	18	19	36		
D	40	27	8	24		
E	14	7	30	22		
F	23	20	19	40		
G	18	31	6	31		
H	17	20	0	19		
I	21	0	14	22		
J	9	11	7	10		
K	24	22	22	0		
L	29	33	9	9		

JA delay is 8 | JI delay is 10 | JH delay is 12 | JK delay is 6

Vectors received from J´s four neighbors

New routing table for J

Distance-Vector-Routing

- Count-to-Infinity-Problem: Distance-Vector-Routing
  - reagiert sehr schnell auf »good news«,
  - aber sehr langsam auf »bad news«
- Beispiel:
  - Metrik: Anzahl der Hops

Topologie

A	B	C	D	E	
∞	∞	∞	∞		Initially
1	∞	∞	∞		After 1 exchange
1	2	∞	∞		After 2 exchanges
1	2	3	∞		After 3 exchanges
1	2	3	4		After 4 exchanges

A geht online: »good news«

A ist für B nicht mehr erreichbar. B lernt, dass C einen Pfad zu A mit 2 Hops hat.

C lernt, dass B oder D einen Pfad zu A mit 3 Hops haben.

B und D lernen, dass C einen Pfad zu A mit 4 Hops hat.

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	∞	∞	∞	∞	

A geht offline: »bad news«

»Infinity« sollte auf Pfadlänge+1 gesetzt werden

Link-State-Routing

- Distance-Vector-Routing bis 1979 verwendet im ARPANET …
  - Metrik: Warteschlangenlänge für die Pakete
  - keine Berücksichtigung der auf einem Link zur Verfügung stehenden Bandbreite (am Anfang wurden alle Links mit 56 kbps betrieben)
- … dann abgelöst von Link-State-Routing
  - berücksichtigt unterschiedliche Bandbreite der Links
  - verbessert die Situation bzgl. des Count-to-Infinity-Problems

- Jeder Router muss
  1. seine Nachbarn entdecken und ihre Netzadressen feststellen
  2. Verzögerung oder Leitungskosten seiner Nachbarn messen
  3. Paket zusammenstellen, in dem alles steht, was er gelernt hat
  4. Link-State-Pakete an alle anderen Router senden
  5. kürzesten Pfad zu allen anderen Routern berechnen (Dijkstra)

Link-State-Routing

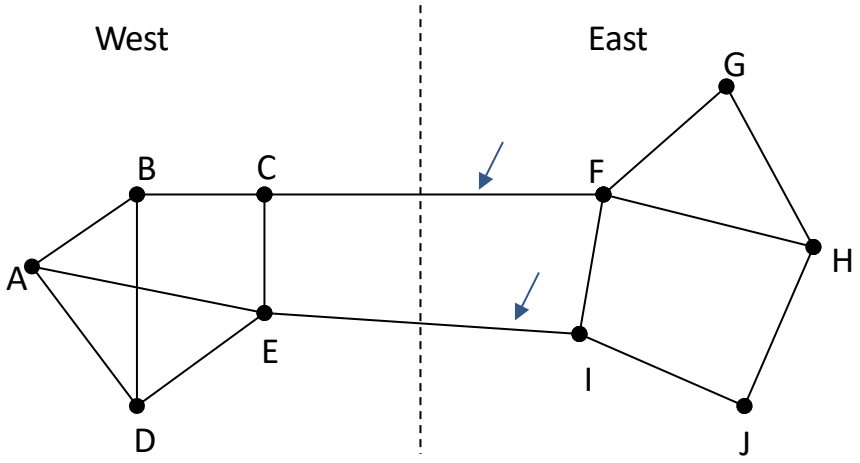
- Nachbarn entdecken und ihre Netzadressen feststellen
  - Router sendet nach Hochfahren HELLO-Pakete auf jeder Kommunikationsverbindung aus
  - Router antwortet auf HELLO mit einem global eindeutigen Namen

- Verzögerung oder Leitungskosten seiner Nachbarn messen
  - Router sendet ECHO-Pakete auf Kommunikationsverbindung
  - Router antwortet auf ECHO-Pakete, indem er das Paket zurücksendet
  - Round-Trip-Time: Zeit bis zum Empfang des eigenen Pakets wird gemessen

  - Frage: Sollte Auslastung der Kommunikationsverbindung berücksichtigt werden oder nicht?

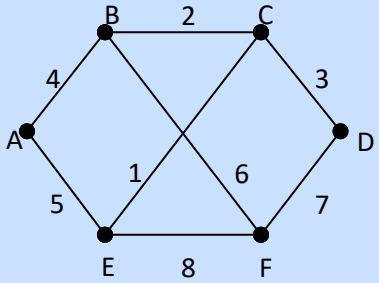
Problem: Ermittlung Leitungskosten

- Frage: Sollte Auslastung der Kommunikationsverbindung berücksichtigt werden oder nicht?
  - Pro: Wenn Router die Wahl zwischen zwei Links gleicher Bandbreite hat, verbessert die Berücksichtigung der Auslastung die Performance.
  - Contra: Angenommen, C-F ist momentan stark ausgelastet, wird der nächste Routing Table E-I bevorzugen, was zu starker Auslastung von E-I führt. Im nächsten Schritt wird wieder C-F bevorzugt, u.s.w.



Link-State-Routing

- Link-State-Pakete
  - Identität
  - Sequenznummer: 32-Bit-Feld (1 Paket/sek = Zykluszeit 137 Jahre)
  - Alter = (Rest)-Lebenszeit eines Pakets
  - Liste der Nachbarn mit Verzögerungszeit



Link State Packets

A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
E	1

D	
Seq.	
Age	
C	3
F	7

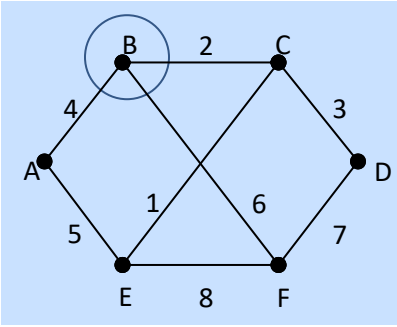
E	
Seq.	
Age	
A	5
C	1
F	8

F	
Seq.	
Age	
B	6
D	7
E	8

- Verteilen von Link-State-Paketen: Flooding
  - Empfangene Pakete mit Seq > Seq_last werden an alle Links außer dem eingehenden gesendet
  - Empfang jedes Flooding-Pakets muss bestätigt werden

Link-State-Routing

- Flooding
  - Empfang jedes Flooding-Pakets muss bestätigt werden
- Paketpuffer
  - Restlebenszeit wird (jede Sekunde) erniedrigt, wenn 0, wird der Eintrag weggeworfen
- Beispiel: Paketpuffer für Router B



Source	Seq.	Age	Send flags A	Send flags C	Send flags F	ACK flags A	ACK flags C	ACK flags F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Link-State-Routing: Bedeutung der Flags

Paket von A erhalten, muss an C und F weiter gesendet werden

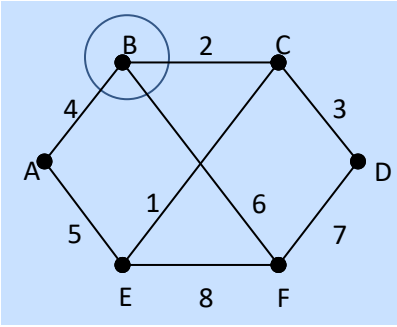
Paket von F erhalten, muss an A und C weiter gesendet werden

Paket von E 2 x erhalten (1x über EAB, 1x über EFB), muss nur an C weiter gesendet werden

Bestätigung muss an A gesendet werden

Bestätigung muss an F gesendet werden

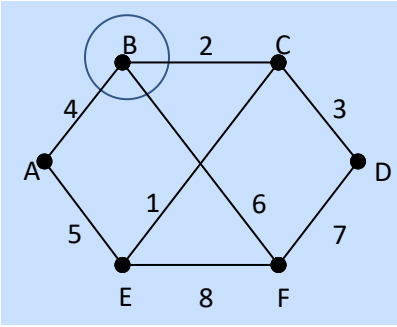
Bestätigung muss an A und F gesendet werden



Source	Seq.	Age	Send flags A	Send flags C	Send flags F	ACK flags A	ACK flags C	ACK flags F	Data
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	0	0	0	1	1	

Link-State-Routing

- Ziel: kürzesten Pfad zu allen anderen Routern berechnen
  - Jeder Router kennt vollständigen Graphen
  - eigentlich gerichteter Graph: entweder getrennt benutzen oder mitteln
  - Dijkstra-Algorithmus (Shortest Path) für alle Ziele berechnen



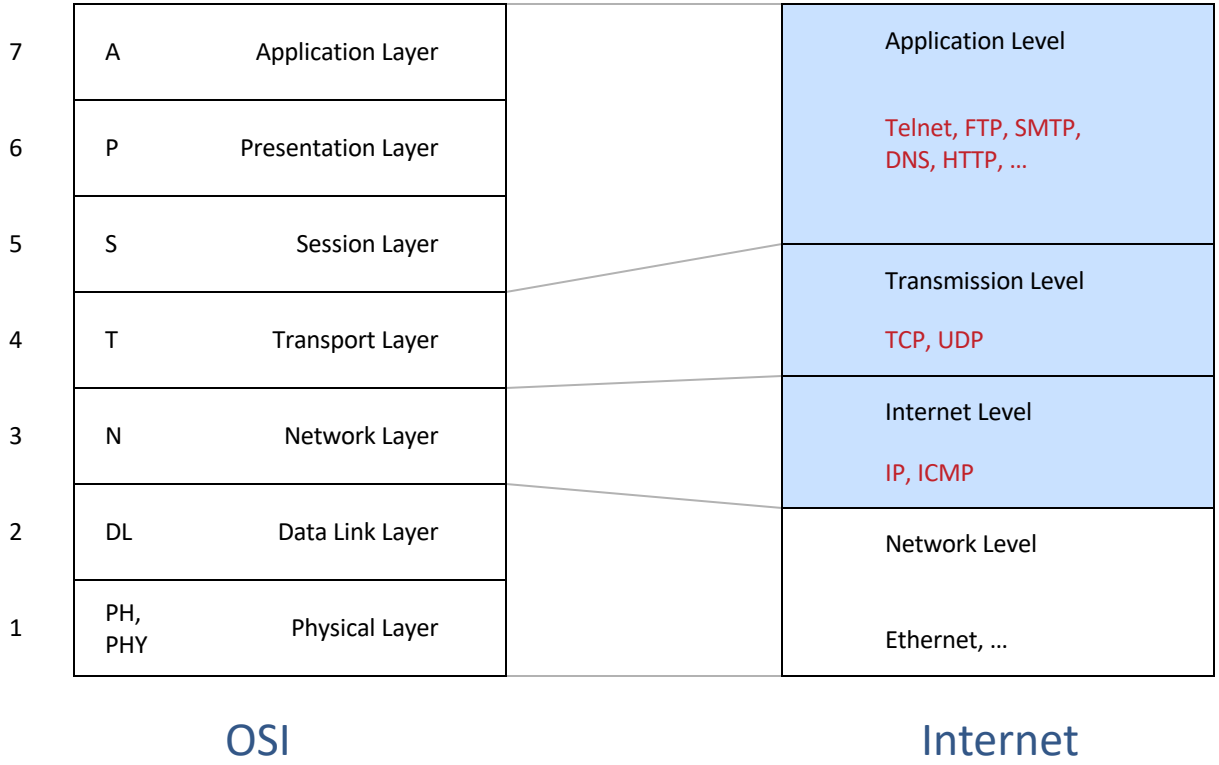
- Zusammenfassung: Jeder Router muss
  1. seine Nachbarn entdecken und ihre Netzadressen feststellen
  2. Verzögerung oder Leitungskosten seiner Nachbarn messen
  3. Paket zusammenstellen, in dem alles steht, was er gelernt hat
  4. Link-State-Pakete an alle anderen Router senden
  5. kürzesten Pfad zu allen anderen Routern berechnen (Dijkstra)

Internet (ARP, IP, TCP, …)

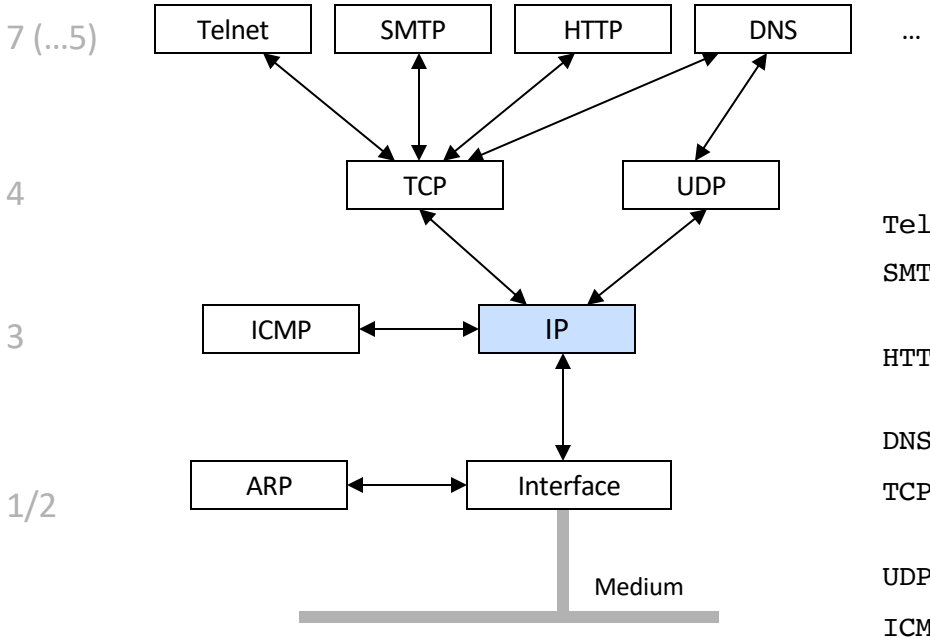
- Schicht 3 und 4: ARP, IP, TCP, UDP, ICMP
- Subnetting, Fenstermanagement, Überlaststeuerung
- Schicht 7: HTTP, SMTP, weitere Anwendungsdienste

A	Telnet, FTP, SMTP, DNS, HTTP, …
T	TCP, UDP
N	IP
DL	(Data Link Layer)
PH, PHY	(Physical Layer)

Schichtenmodelle OSI und Internet



Protokollebenen



Telnet Teletype Network

SMTP Simple Mail Transfer Protocol

HTTP Hypertext Transmission Protocol

DNS Domain Name System

TCP Transmission Control Protocol

UDP User Datagram Protocol

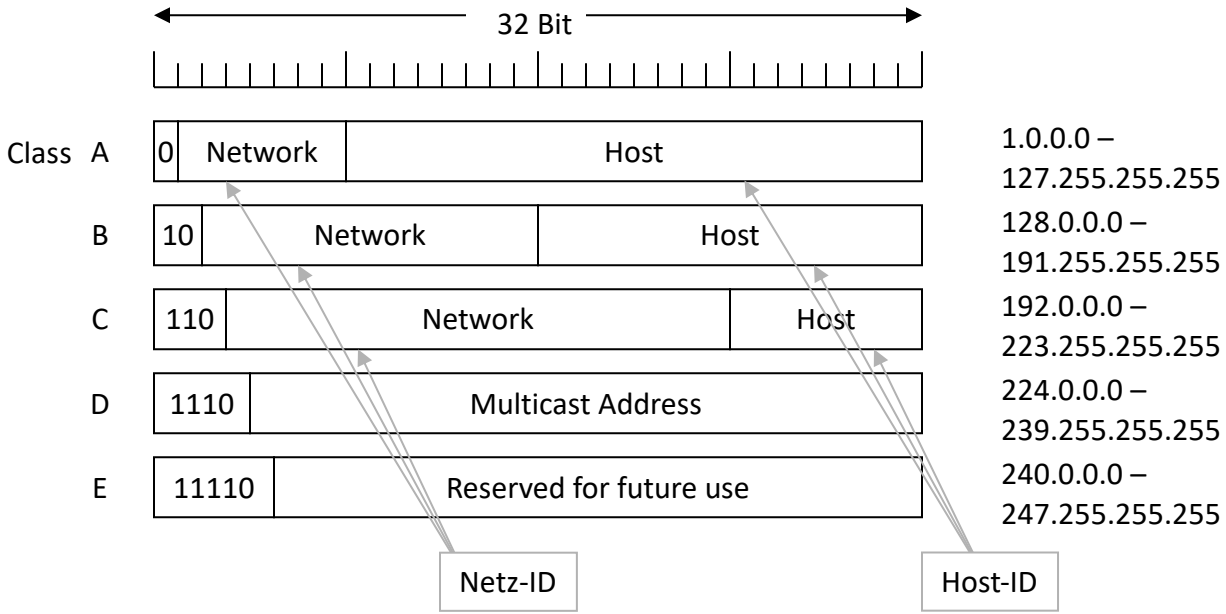
ICMP Internet Control Message Protocol

IP Internet Protocol

ARP Adress Resolution Protocol

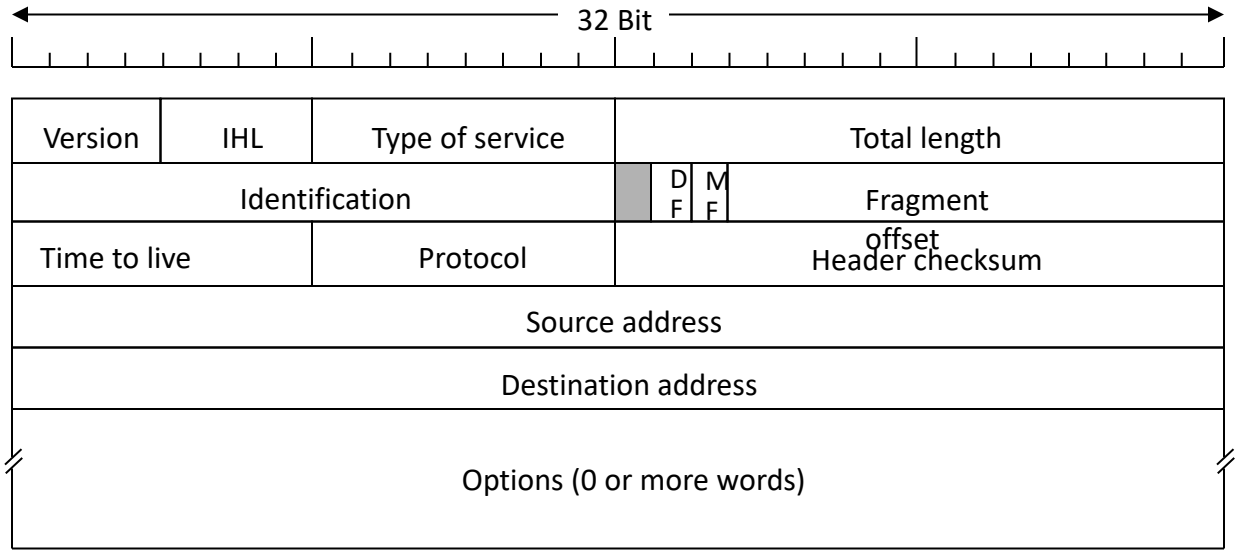
Internet Protocol

- IP
  - stellt Paketvermittungsdienst zur Verfügung
  - Adressierung über IP-Adresse: (IPv4)
    4 durch Punkte getrennte Zahlen zwischen 0 und 255



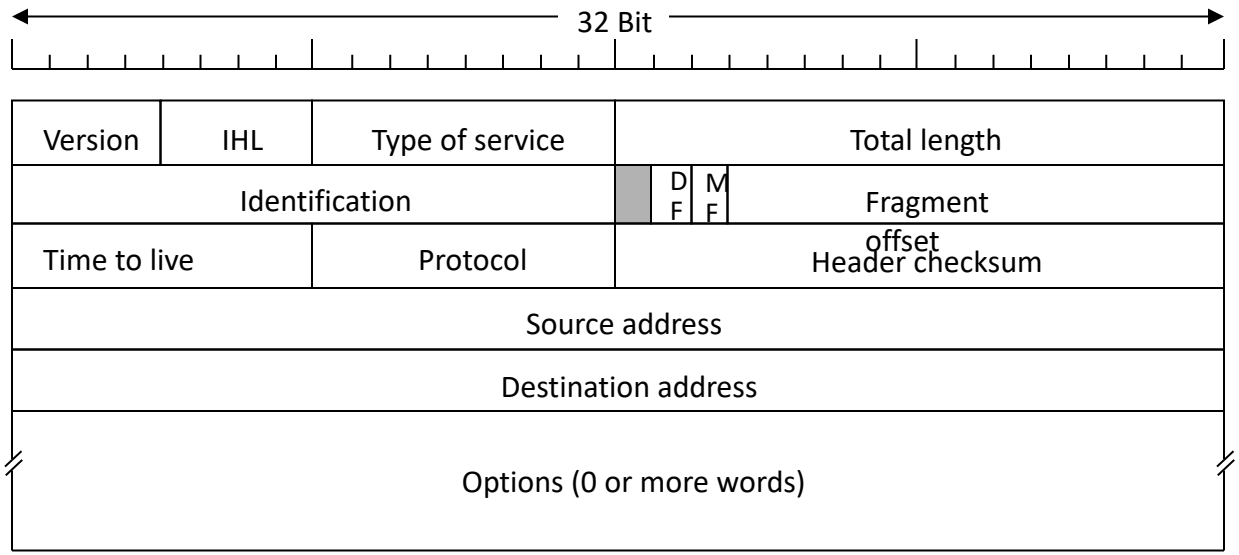
Header des IP-Protokolls

- Version (4 Bit): v4 oder v6
- IHL=Headerlänge (4 Bit) in 32-Bit-Worten: maximale Länge des Headers 15*32Bit = 60 Byte
- Type of Service (8 Bit): Prioritätslevel (erste 3 Bit), Quality of Service-Merkmale, meist ignoriert
- Total length in Byte: maximale Länge eines Pakets: 64 Kbyte



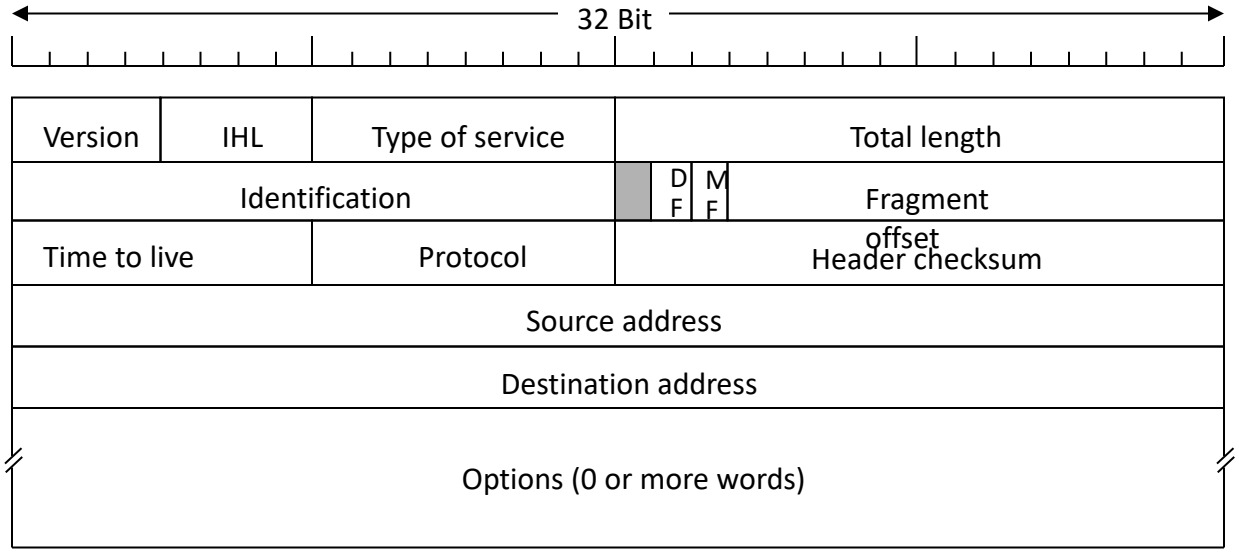
Header des IP-Protokolls

- Fragmentierung: große IP-Pakete passen möglicherweise nicht in eine einzige Schicht 2-PDU
- Identification (16 Bit): dient der Zusammensetzung fragmentierter Pakete
- DF (1 Bit): Do-Not-Fragment-Flag (1=Fragmentierung nicht erlaubt)
- MF (1 Bit): More-Flag: (1=weitere Fragmente folgen)
- Fragment offset (13 Bit): Position des Fragments im unfragmentierten Datagramm als Vielfaches von 8 Bytes

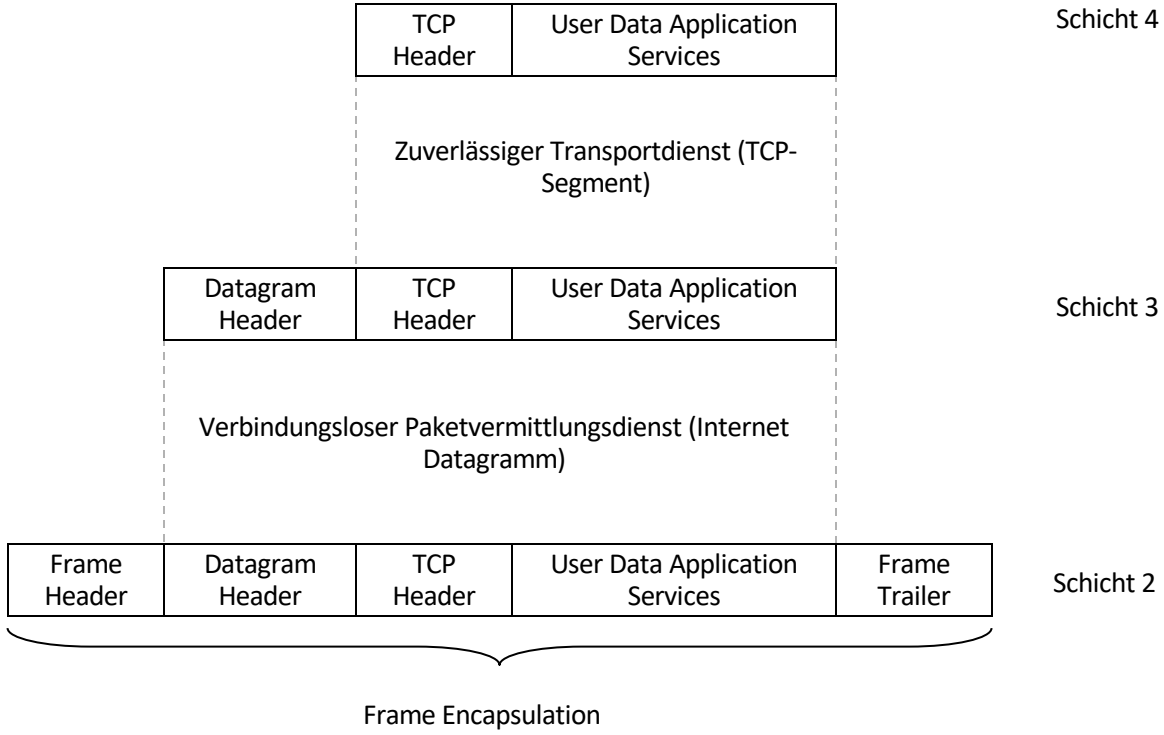


Header des IP-Protokolls

- Time to live (TTL, 8 Bit): verhindert, dass Paket in einer Schleife innerhalb des Netzes feststeckt, als Hop-Count verwendet, d.h. Router verringert Wert um 1
- Protocol (8 Bit): identifiziert das Transportprotokoll (17_{dez}=UDP, 6_{dez}=TCP)
- Header checksum (16 Bit): Prüfsumme für den Header, muss neuberechnet werden, wenn TTL-Feld sich ändert

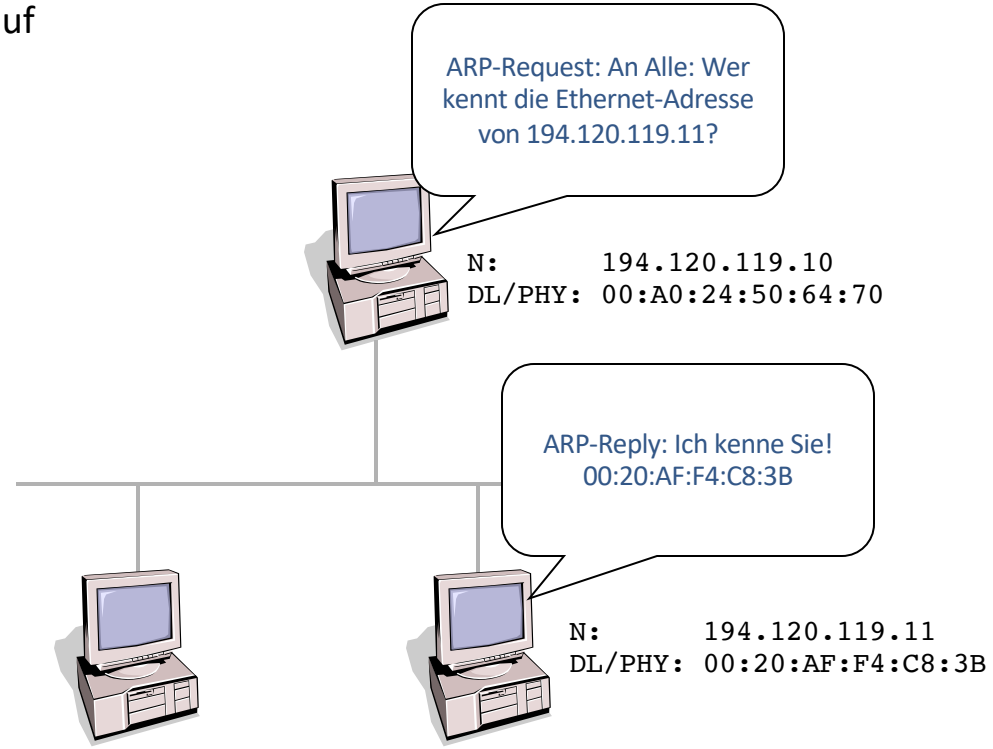


Transportpakete im Internet



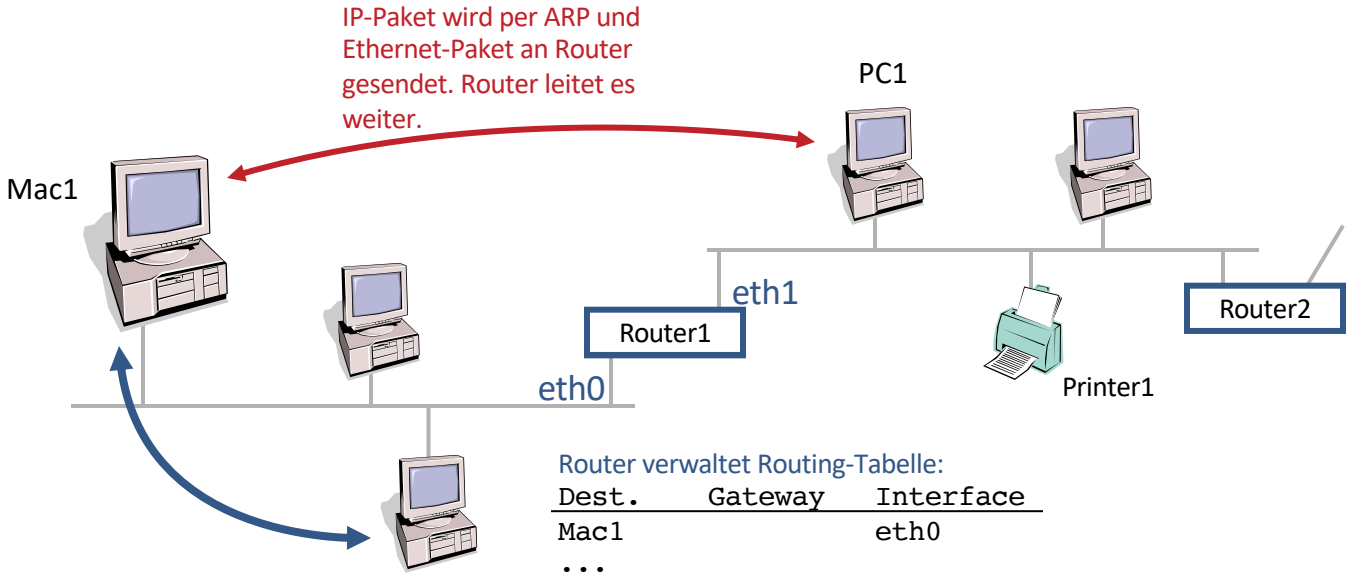
Paket-Transport auf unterster Ebene

- Zuordnung IP-Adressen — Ethernet-Adressen
  - ARP: Address Resolution Protocol
  - Abbildung von IP-Adresse auf Hardwareadresse
  - Anfrage an das lokale Netz (Broadcast)
  - nur in lokalen Netzen
  - Zuordnung wird im ARP-Cache zwischen-gespeichert



Paket-Transport auf unterster Ebene

- Transport über LAN-Grenzen
  - Router

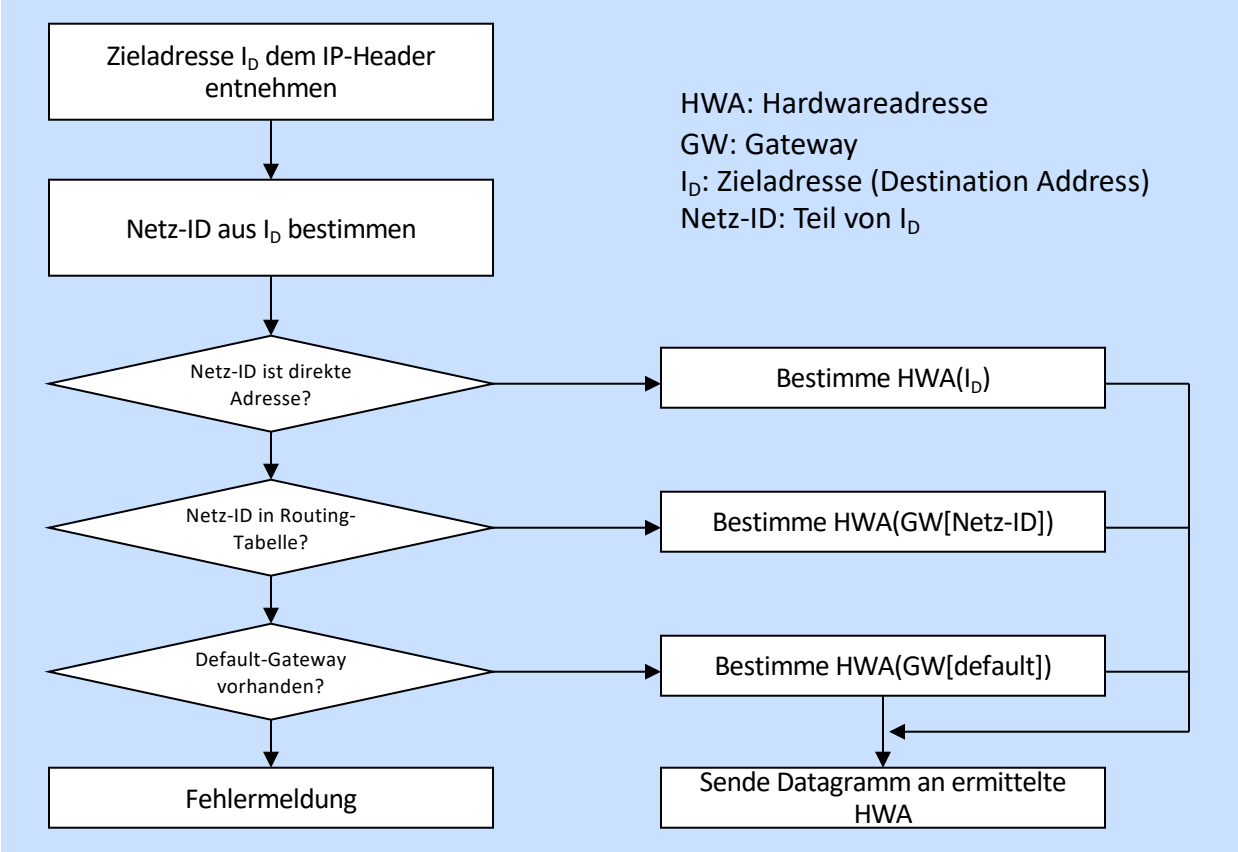


Router verwaltet Routing-Tabelle:

Dest.	Gateway	Interface
Mac1		eth0
...		
PC1		eth1
Printer1		eth1
...		
PC18	Router2	eth1
default	Router2	eth1

VSS 17.04.2020

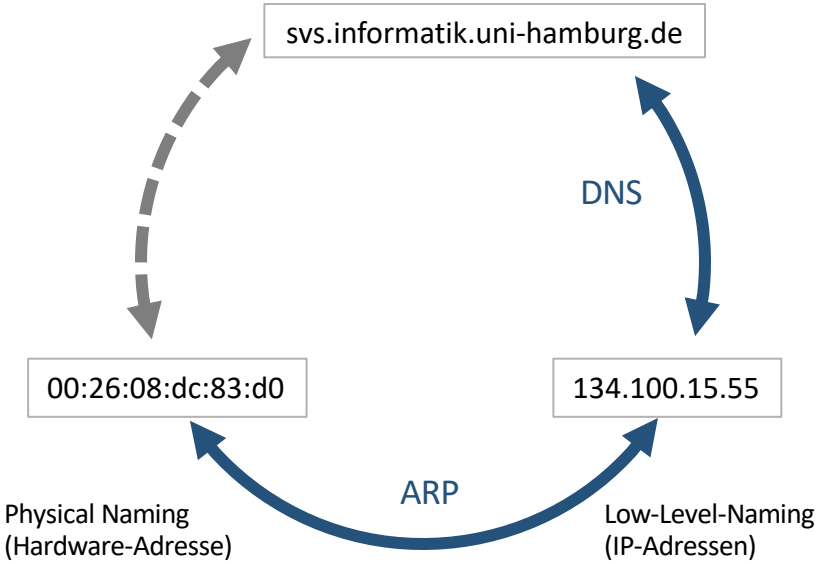
Routing Algorithmus von IP



Einordnung ARP, IP, DNS

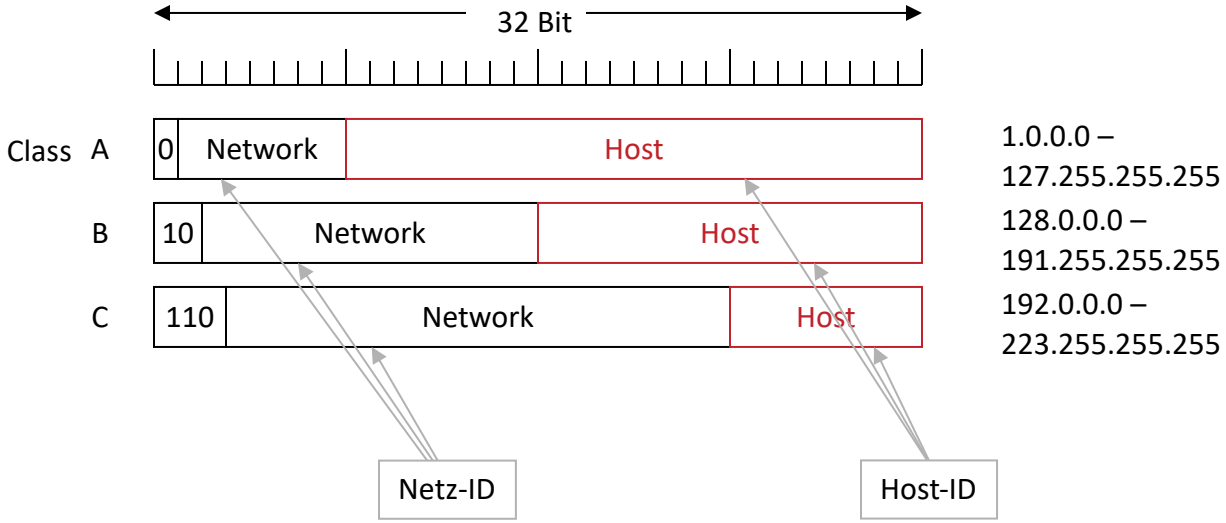
- DNS: Domain Name System
  - Abbildung des Rechnernamens auf IP-Adresse
  - Anfrage an Nameserver
  - typischerweise in WANs

- ARP: Address Resolution Protocol
  - Abbildung von IP-Adresse auf Hardwareadresse
  - Anfrage an das lokale Netz (Broadcast)
  - nur in lokalen Netzen



Subnetting

- Anwendungsbereich
  - Erlaubt, einen Teil des Adressraums der Host-ID zur Verwaltung von Subnetzen zu verwenden
  - Der Teil der Host-ID, der für das Subnetting verwendet wird, wird Subnet-ID oder Subnet-Number genannt



Subnetting

- Beispiel

IP Address:	141.76.75.109
Subnet Mask:	255.255.255.0
Router:	141.76.75.1

Class B	10	Network	Host	128.0.0.0 – 191.255.255.255

Network-ID ⟷ Host-ID

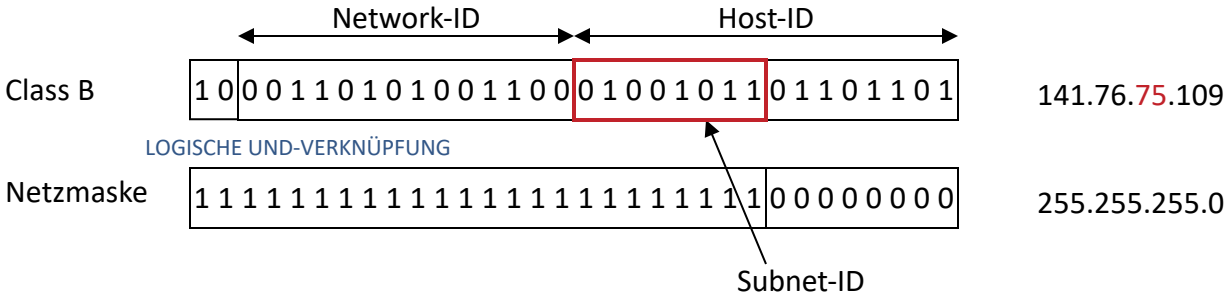
Class B	1 0	0 0 1 1 0 1 0 1 0 0 1 1 0 0	0 1 0 0 1 0 1 1	0 1 1 0 1 1 0 1	141.76.75.109
Netzmaske	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		1 1	0 0 0 0 0 0 0 0	255.255.255.0

LOGISCHE UND-VERKNÜPFUNG

Subnet-ID

Subnetting

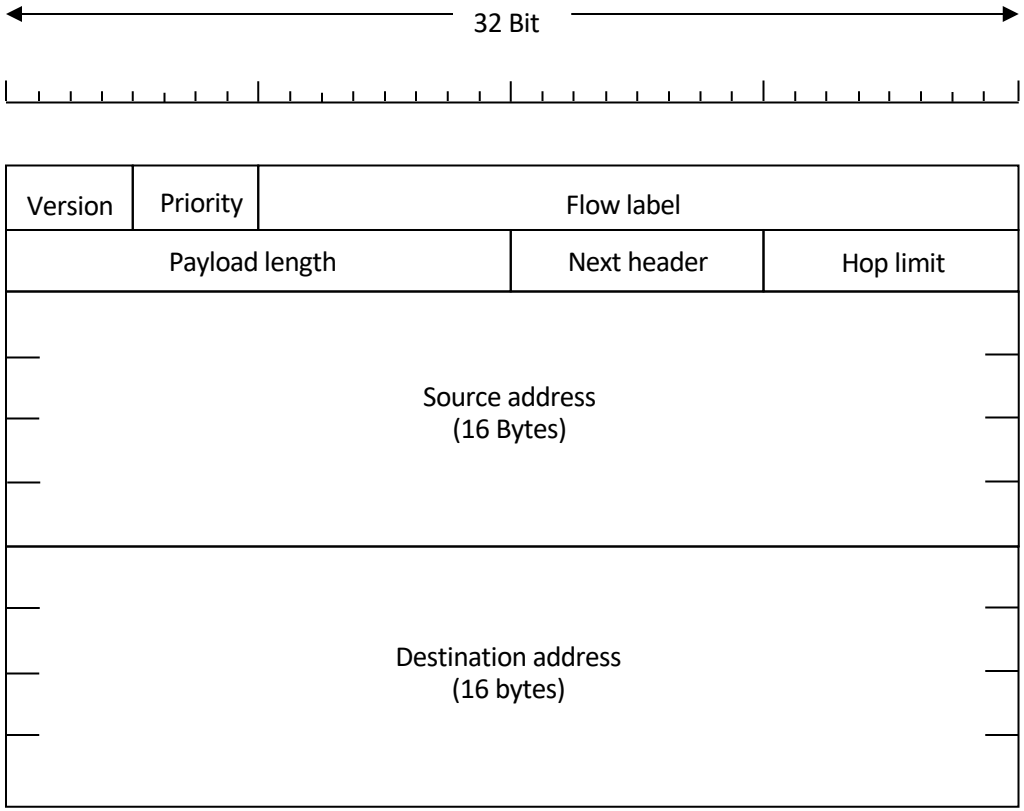
- Ergebnis
  - Routerentscheidung nicht mehr aufgrund ursprünglicher Network-ID (141.76), sondern nach Network-ID AND Netzmaske (141.76.75)
  - Innerhalb einer Organisation mehrere Subnetze möglich
- Vorteile
  - Dezentralisierung: Verwaltung des Adressraums unterliegt der jeweiligen Struktureinheit
  - Lastverteilung: Reduktion der Verkehrslast am zentralen Router (bei interner Kommunikation)



IP Version 6

- IPv6
  - Erweiterung des Adressraums
    - 4 Byte-Adressen (IPv4) --> 16-Byte-Adressen (IPv6)
  - bessere Unterstützung der QoS-Anforderungen von Echtzeitanwendungen: Priorisierung von IP-Paketen
    - Streaming
  - Verbesserung der Sicherheit
    - Authentication Header
    - Encapsulated Security Payload

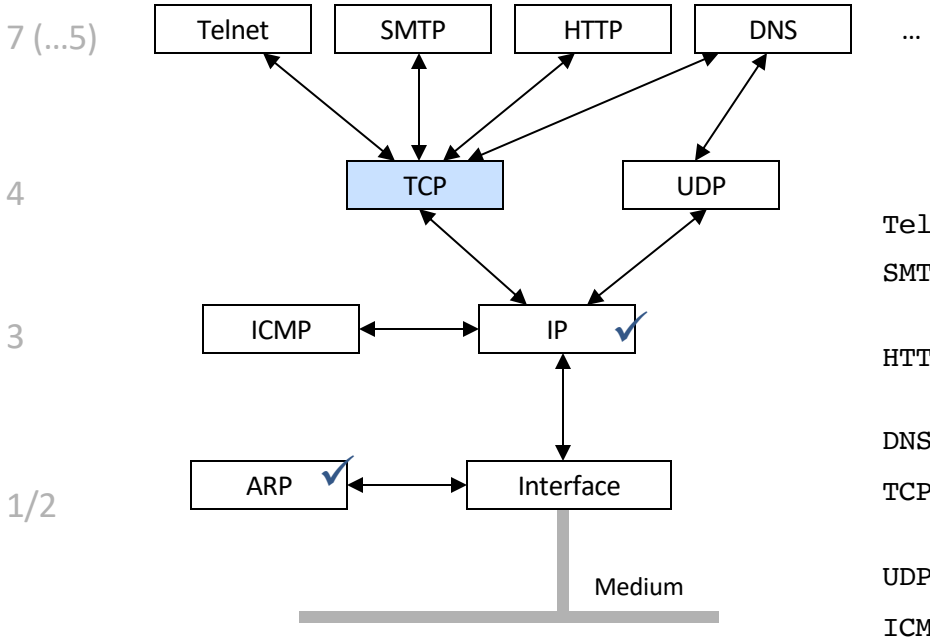
Der feste IPv6-Header



Die IPv6-Erweiterungsheader

Erweiterungsheader	Beschreibung
Optionen für Teilstrecken (Hop-by-Hop)	Verschiedene Informationen für Router
Routing	Definition einer vollen oder teilweisen Route
Fragmentierung	Verwaltung von Datengrammfragmenten
Authentifikation	Echtheitsüberprüfung des Senders
Verschlüsselung	Vertraulichkeit des Inhalts (Payload)
Optionen für Ziele	Zusätzliche Informationen für das betreffende Ziel

Protokollebenen



Telnet Teletype Network

SMTP Simple Mail Transfer Protocol

HTTP Hypertext Transmission Protocol

DNS Domain Name System

TCP Transmission Control Protocol

UDP User Datagram Protocol

ICMP Internet Control Message Protocol

IP Internet Protocol

ARP Adress Resolution Protocol

Transmission Control Protocol (TCP)

- Aufgabe: Bereitstellung von verbindungsorientierten Diensten für die Anwendungsschicht
  - Zugangspunkt eines bestimmten mittels TCP realisierten Dienstes: Portnummer (16 Bit)
  - Dienstprimitive werden Sockets genannt

- TCP-Schicht
  - zerteilt Datenstrom von Anwendungsschicht in TCP-Segmente,
  - versieht sie mit TCP-Header und
  - übergibt Segmente der IP-Schicht

- Verbindungssteuerung
  - Jedes TCP-Segment erhält Sequenznummer
  - Empfang muss bestätigt werden

Socket-Primitive für TCP

Primitiv	Bedeutung	Server	Client
SOCKET	Neuen Kommunikationsendpunkt erzeugen	x	x
BIND	Lokale Adresse an eine Socket binden	x	
LISTEN	Socket zum Empfang von Verbindungen vorbereiten	x	
ACCEPT	Blockiert Kontrollfluss, bis ein Verbindungsversuch eintrifft	x	
CONNECT	Verbindungsversuch zu einer (Server)-Socket		x
SEND	Daten über Verbindung senden	x	x
RECEIVE	Daten über Verbindung empfangen	x	x
CLOSE	Verbindung schließen	x	x

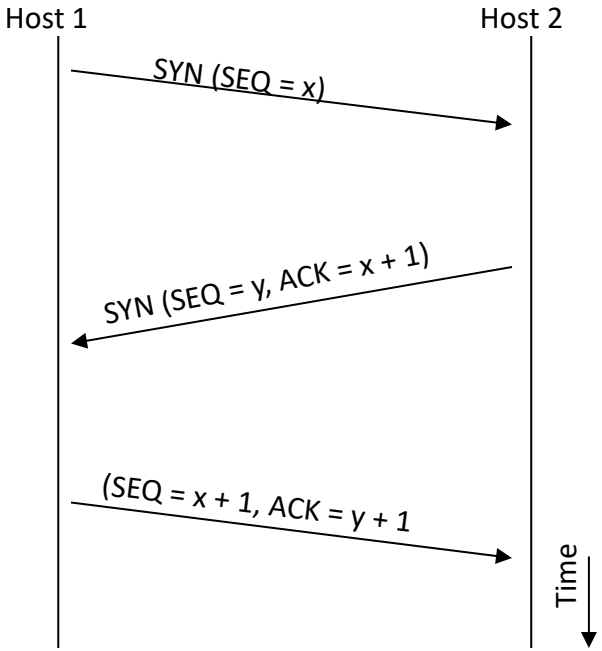
Portnummern

- Source Port, Destination Port (16 Bit): identifiziert SAP
  - 0-1023: well known ports
  - 1024-49151: registrierte Dienste
  - Beispiele für well known ports:

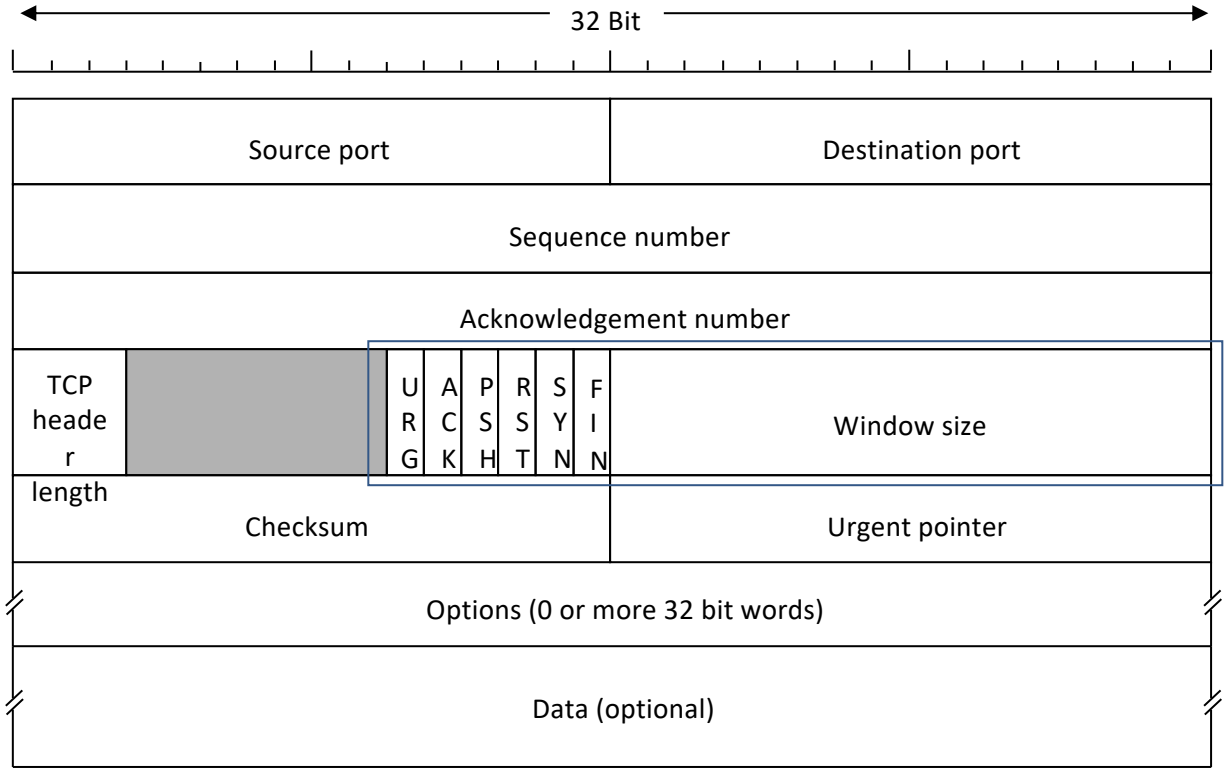
```
 20, 21   FTP File Transfer (Data, Control)
 22       SSH Remote Login
 23       Telnet
 25       SMTP Simple Mail Transfer
 53       DNS Domain Name Server
 79       Finger
 80       HTTP World Wide Web HTTP
 88       Kerberos
 110      POP3 Post Office Protocol Version 3
 119      NNTP Network News Transfer Protocol
 143      IMAP Internet Message Access Protocol
 194      IRC Internet Relay Chat Protocol
 257      SET Secure Electronic Transaction
 443      HTTPS - HTTP over TLS/SSL
siehe auch: http://www.iana.org/assignments/port-numbers
```

TCP-Verbindungsaufbau

- 3-Wege-Handshake mittels sog. SYN-Flags



TCP-Header

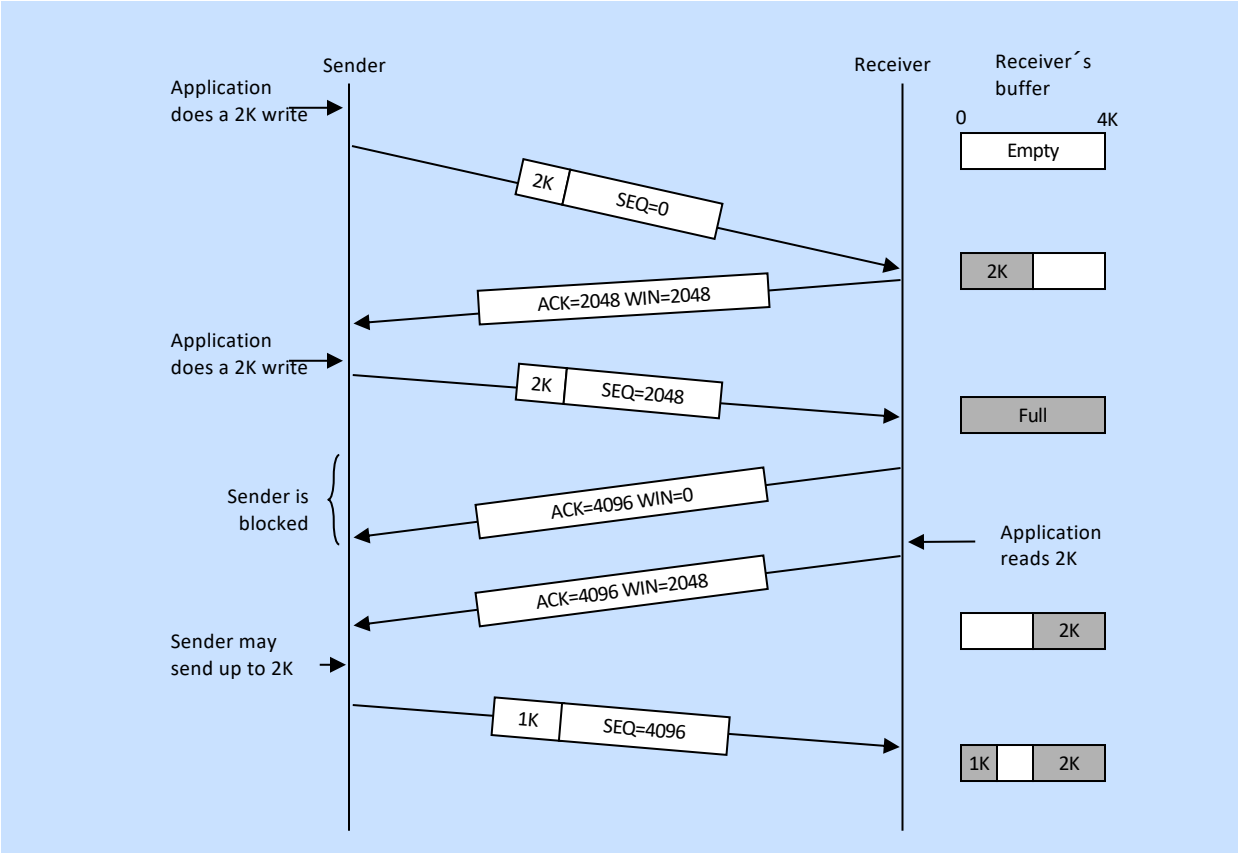

32 Bit
Source port
Destination port
Sequence number
Acknowledgement number
TCP header length
URG
ACK
PSH
RST
SYN
FIN
Window size
Checksum
Urgent pointer
Options (0 or more 32 bit words)
Data (optional)

TCP-Header

- Flags:
  - URG: Urgent-Flag, markiert »dringende« Teile des Datenstroms, Zeiger auf letztes dringendes Datenbyte: Urgent Pointer
  - ACK: Zeigt an, dass in Acknowledgment number gültiger Wert steht
  - PSH: Daten werden auf Senderseite sofort gesendet und auf Empfängerseite sofort an Anwendung übergeben (noch bevor der Sende-/Empfangspuffer voll ist)
  - RST: Reset-Flag, zeigt an, dass Empfänger Verbindung aufgrund einer Fehlersituation beenden soll
  - SYN: zeigt an, dass Sequence number eine *Initial sequence number* ist
  - FIN: zeigt Abbau der Verbindung an

URG	ACK	PSH	RST	SYN	FIN

Fenstermanagement im TCP



Datenfluss-Steuerung in TCP

- TCP stellt zuverlässigen verbindungsorientierten Dienst bereit
  - TCP-Segmente müssen bestätigt werden (ACK)
  - Jedes einzelne TCP-Segment zu bestätigen, ist nicht performant.

- Sliding window
  - Empfänger quittiert eine Gruppe von TCP-Segmenten
  - Anzahl der unbestätigten Segmente (Fenstergröße) wird zwischen Sender und Empfänger dynamisch angepasst
    - Sender sendet maximal Fenstergröße-Anzahl Segmente, ohne eine Bestätigung vom Empfänger erhalten zu haben
  - Wenn Segmente innerhalb des Fensters verloren gehen, bestätigt der Empfänger durch Senden der nächsten erwarteten Byte-Nummer, bis zu welchem Segment er fehlerfrei empfangen hat.

Sliding Window

Sendersicht:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

ACK(5+1) trifft bei Sender ein, verschiebt Window, sendet Segmente 6-10

Empfängersicht:

1	2	3	4	5	6 ✓	7 ✓	8	9 ✓	10 ✓	11	12	13	14	15	16	17	18	19

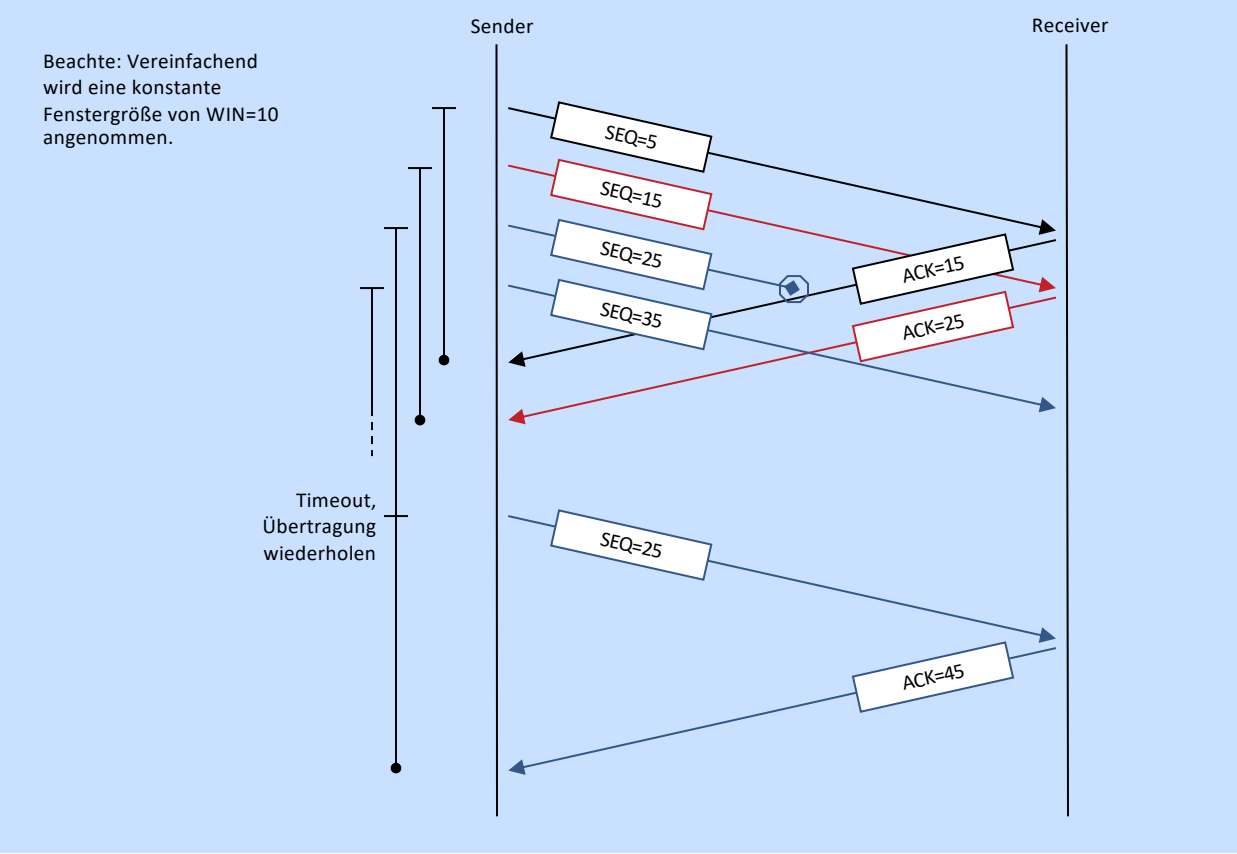
Segment 8 geht verloren, Timeout f. Seg. 8 beim Empfänger, Empfänger sendet ACK(7+1)

Sendersicht:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Timeout für ACK(8+1), Sender wiederholt Sendung von Seg. 8

Fenstermanagement in TCP



TCP Überlastungssteuerung

- Anfangs nur TCP-Header-Feld »Window Size« (win)
  - Dieser Ansatz ist nicht optimal, da er Verzögerungen innerhalb des Netzes nicht berücksichtigt!

- Erweiterungen
  - Slow Start
  - Congestion Avoidance
  - Fast Retransmit
  - Fast Recovery

TCP Slow Start

- Vorgehen:
  - Initialisierung:
    - Festlegen der Segmentgröße s
    - Congestion Window (cwnd) = s
  - Ablauf:
    - Senden von Daten der Größe Min(cwnd, win)
    - Für jede eintreffende Bestätigung (ACK) cwnd um s erhöhen

- Bemerkungen:
  - Normalerweise wird für jedes Segment ein ACK geschickt
  - Daraus ergibt sich ein exponentielles (!) Wachstum von cwnd

Bsp. Slow Start

- Annahme:
  - win = 4 KB und konstant
  - ACKs treffen gleichzeitig ein
- Initialisierung: cwnd = s = 512 Byte
- Ablauf:
  - Senden von 512 Byte
    - ACK trifft ein: cwnd = cwnd + s = 1 KB
  - Senden von 1 KB
    - Zwei ACK treffen ein: cwnd = cwnd + s + s = 2 KB
  - Senden von 2 KB
    - Vier ACK treffen ein: cwnd = cwnd + 4*s = 4 KB
  - Senden von 4 KB
    - Acht ACK treffen ein: cwnd = cwnd + 8*s = 8 KB
  - Senden von 4 KB (Min aus cwnd und win!)
  - …

TCP Congestion Avoidance

- Problem:
  - Die Überlastungsgrenze des Netzwerks wird von Slow Start relativ schnell überschritten.

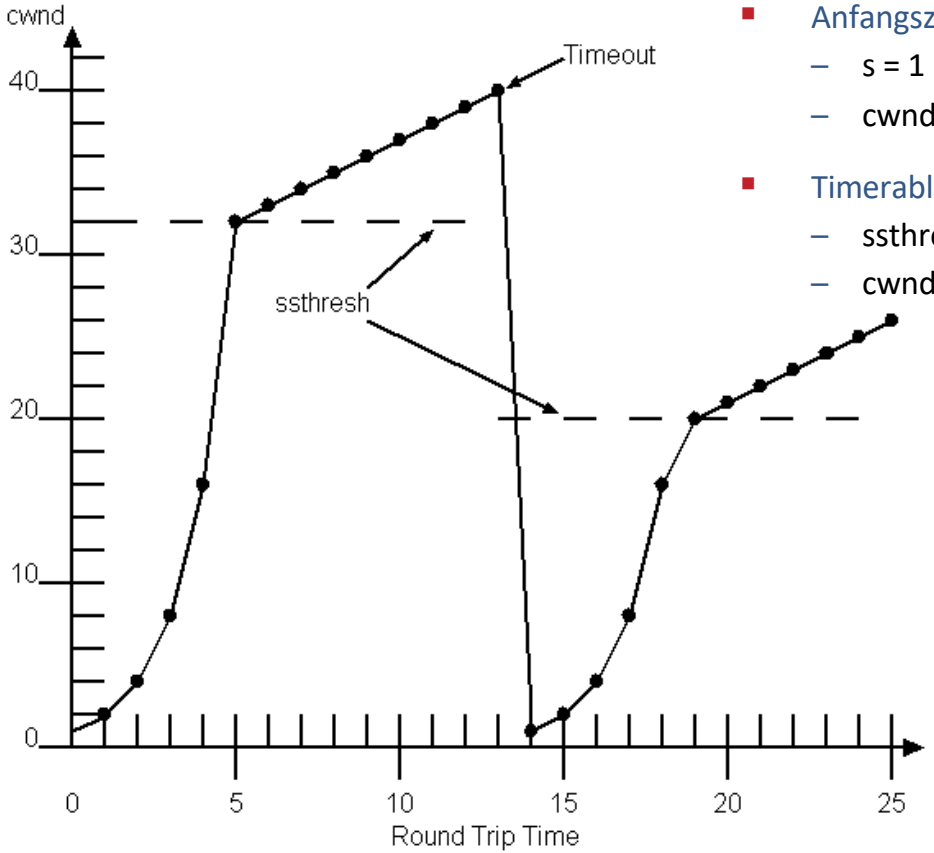
- Lösung:
  - Congestion Avoidance

- Funktionsweise:
  - Ab einer bestimmten Größe wird cwnd nicht mehr exponentiell vergrößert, sondern nur noch linear.
  - Neue Variable ssthresh (»Slow Start Threshold Size«) nötig, die angibt, ab wann die Vergrößerung nur noch linear nach dem Congestion Avoidance Algorithmus erfolgen soll.

Slow Start mit Congestion Avoidance

- Vorgehen:
  - Initialisierung:
    - Festlegen der Segmentgröße s
    - Congestion Window (cwnd) = s
    - Slow Start Threshold ssthresh = 65.535 Byte
  - Ablauf:
    - Beginne nach der Initialisierung mit Slow Start
    - A) Eine ACK trifft ein:
      - Wenn cwnd <= ssthresh: Slow Start
      - Wenn cwnd > ssthresh: Congestion Avoidance
    - B) Ein Timeout tritt ein:
      - ssthresh = Max(2*s, Min(cwnd,win)/2)
      - cwnd = s

Bsp. Slow Start mit Congestion Avoidance



- Anfangszustand:
  - s = 1 KB
  - $cwnd_{-1}$ = 64 KB
- Timerablauf vor 0 =>
  - ssthresh = 32 KB
  - $cwnd_0$ = 1 KB

TCP Fast Retransmit

- verwendet ab 1988

- Idee:
  - TCP schickt sofort eine doppelte ACK, wenn ein Segment ankommt, das noch nicht an der Reihe wäre.
  - Wenn nur eine oder zwei solcher doppelter ACKs ankommen, kann es sich um normale Netzverzögerung handeln.
  - Kommen mehr als zwei doppelte ACKs an, geht man von einem Paketverlust aus und sendet das fehlende Segment nochmals, ohne auf das Ablaufen des Retransmit Timers zu warten.

TCP Fast Recovery

- verwendet ab 1990

- Idee:
  - Nach dem Erhalt der dritten doppelten ACK wird ssthresh auf die Hälfte des aktuellen cwnd gesetzt und das fehlende Segment erneut gesendet (Fast Retransmit).
  - cwnd wird auf ssthresh + 3*s gesetzt und für jede weitere doppelte ACK um s erhöht.
  - Beim Eintreffen der ersten einfachen ACK (sollte die ACK für das wiederholte Segment sein) wird cwnd = ssthresh gesetzt und mit Congestion Avoidance weitergemacht.

- nur verwendet im Zusammenspiel mit Fast Retransmit

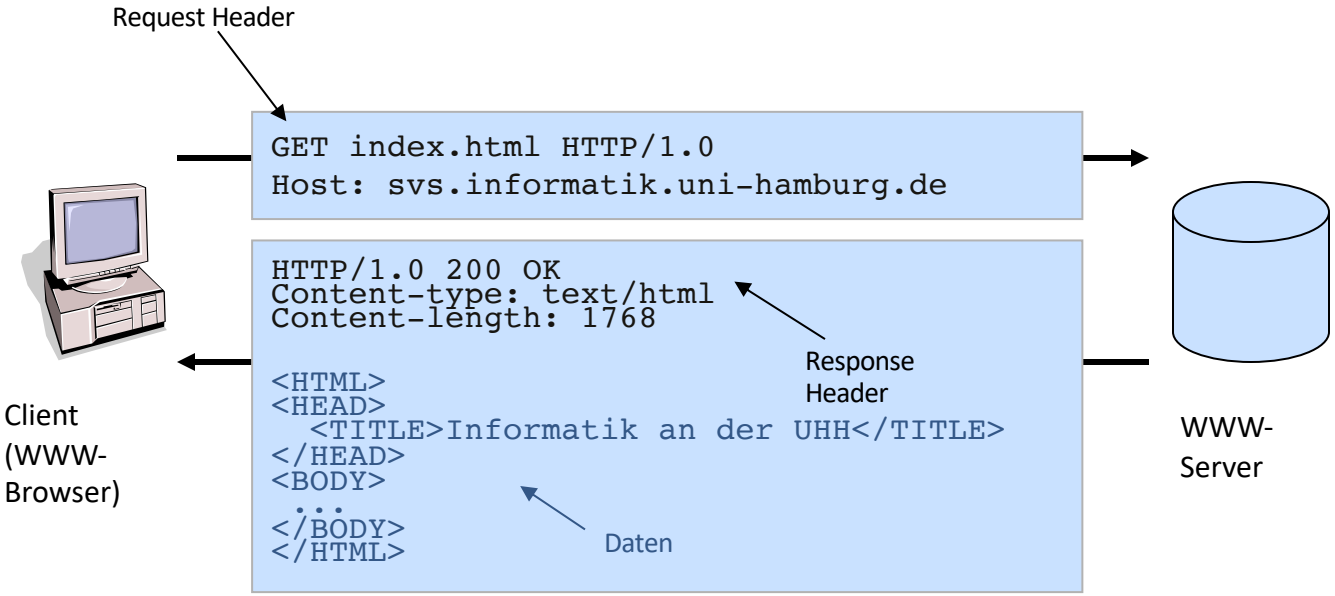
Schicht 7 Beispiel: HTTP (Hypertext Transfer Protocol)

- Server-Socket
  - erzeugt Socket, bindet sie an lokale IP-Adressen
  - akzeptiert Verbindungswünsche auf TCP-Port 80
  - wartet auf eingehende Verbindungen

- Client:
  - Erzeugt Socket(Server-IP-Adresse, Port 80)
  - öffnet Verbindung zu Server
  - Sendet Request (schreibt ASCII-Zeichen auf Output-Stream)
  - Liest Response (liest byteweise aus Input-Stream)
  - schließt Verbindung

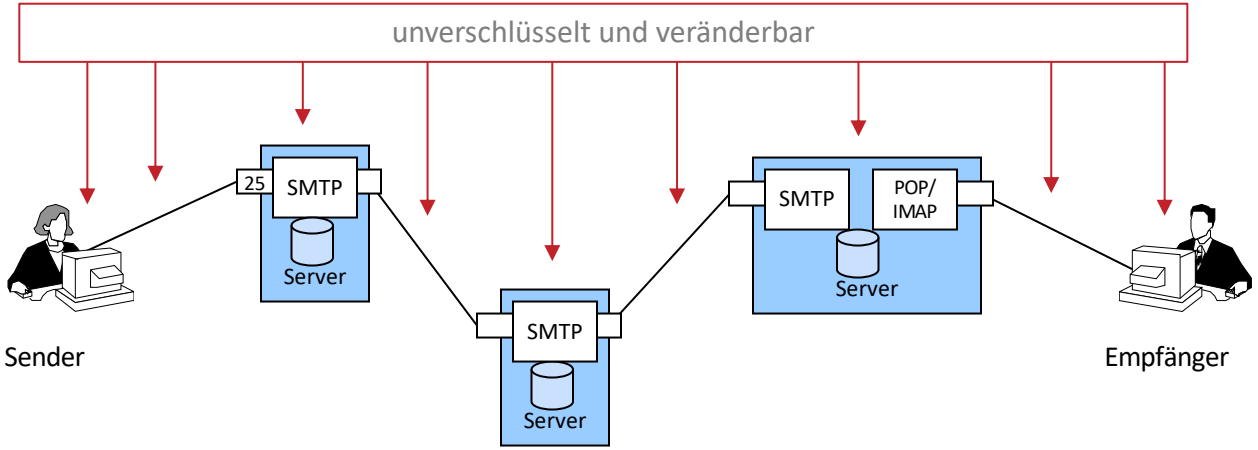
- Protokollspezifikation, Nachrichtenaufbau
  - RFC (Request for Comments) 2068

HTTP (Hypertext Transfer Protocol)



Schicht 7 Beispiel: SMTP (Simple Mail Transfer Protocol)

- Store & Forward-Prinzip:
  - E-Mail wird nicht direkt an Empfänger, sondern über MTA (Mail Transfer Agents) geschickt
- Senden via SMTP: Textbasiertes Protokoll auf TCP-Port 25
- Empfangen: POP, IMAP

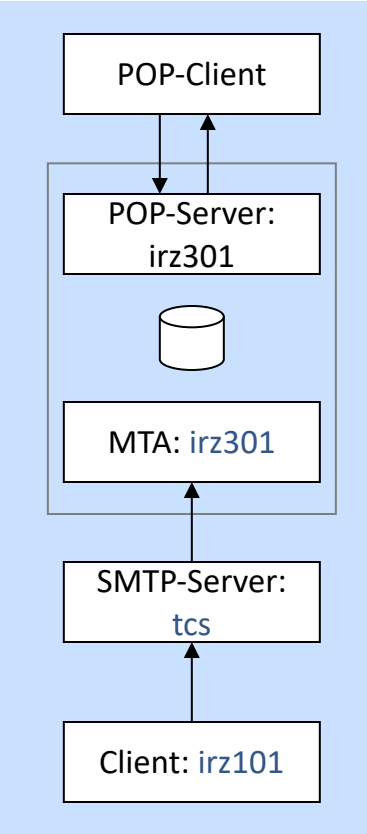


SMTP (Simple Mail Transfer Protocol)

```
> telnet tcs.inf.tu-dresden.de 25
Trying 141.76.75.101...
Connected to tcs.inf.tu-dresden.de.
Escape character is '^]'.
220 tcs.inf.tu-dresden.de Sendmail 4.1/SMI-4.1
ready at Tue, 30 Jan 96 11:13:45  +0100
> MAIL FROM: attacker@nirgends.de
250 attacker@nirgends.de... Sender ok
> RCPT TO: federrath@inf.tu-dresden.de
250 federrath@inf.tu-dresden.de... Recipient ok
> DATA
354 Enter mail, end with "." on a line by itself
> Das ist eine gefaelschte email.
> Mr Nobody
> .
250 Mail accepted
> QUIT
221 tcs.inf.tu-dresden.de delivering mail
Connection closed by foreign host.
```

SMTP (Simple Mail Transfer Protocol)

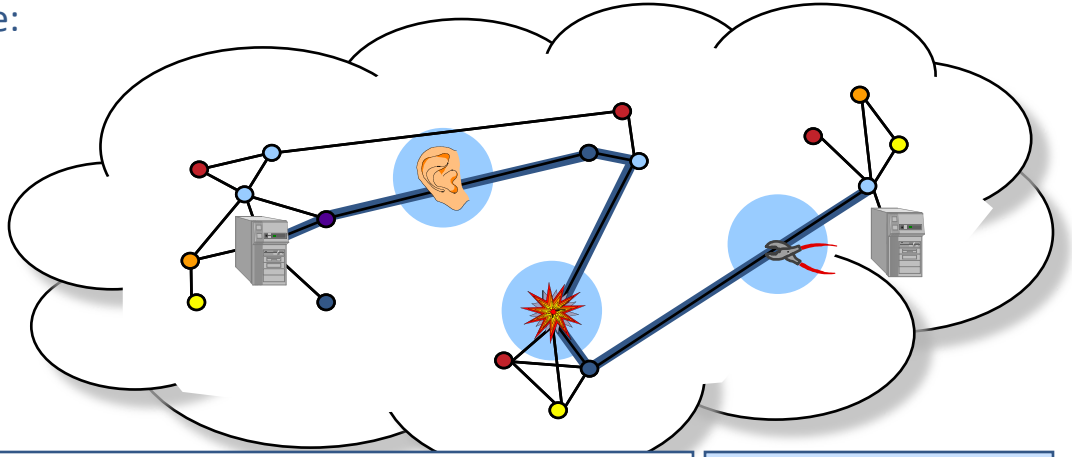
```
X-POP3-Rcpt: hf2@irz301.inf.tu-dresden.de
Received: by irz301.inf.tu-dresden.de (8.6.12/8.6.12-s1) id LAA17026
        for hf2@irz.inf.tu-dresden.de; Tue, 30 Jan 1996 11:14:59 +0100
From: attacker@nirgends.de
Received: from tcs.inf.tu-dresden.de by irz301.inf.tu-dresden.de
        (8.6.12/8.6.12-s1) with SMTP id LAA16965
        for <federrath@inf.tu-dresden.de>; Tue, 30 Jan 1996 11:13:58 +0100
X-PH: V4.2@irz301.inf.tu-dresden.de
Received: from  (irz101.inf.tu-dresden.de)
        by tcs.inf.tu-dresden.de (4.1/SMI-4.1)
        id AA20259; Tue, 30 Jan 96 11:15:50 +0100
Date: Tue, 30 Jan 96 11:15:04 +0100
Message-Id: <9601301015.AA20259@tcs.inf.tu-dresden.de>
Apparently-To: federrath@inf.tu-dresden.de

Das ist eine gefaelschte email.
Mr Nobody
```



IT-Security

VSS 17.04.2020

- Telekommunikationsnetze:
  - sehr viele Betreiber
  - sehr viele Anwender



Bedrohungen:

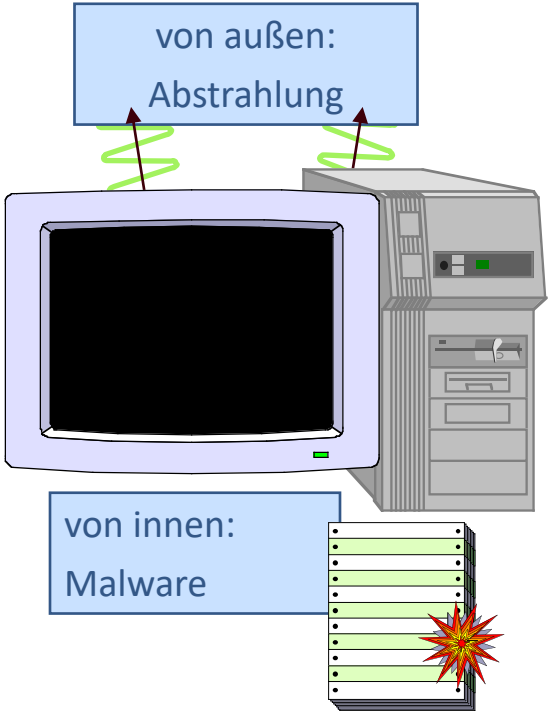
- unbefugter Informationsgewinn
- unbefugte Modifikation
- unbefugte Beeinträchtigung der Funktionalität

Schutz der

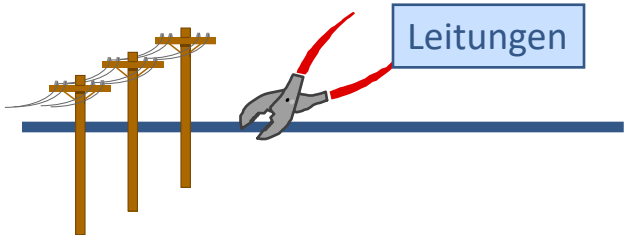
- Vertraulichkeit
- Integrität
- Verfügbarkeit

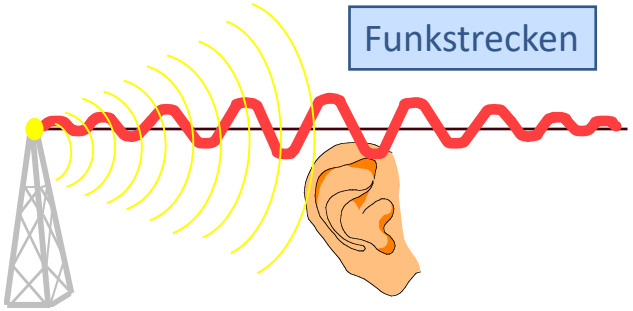
Angriffspunkte

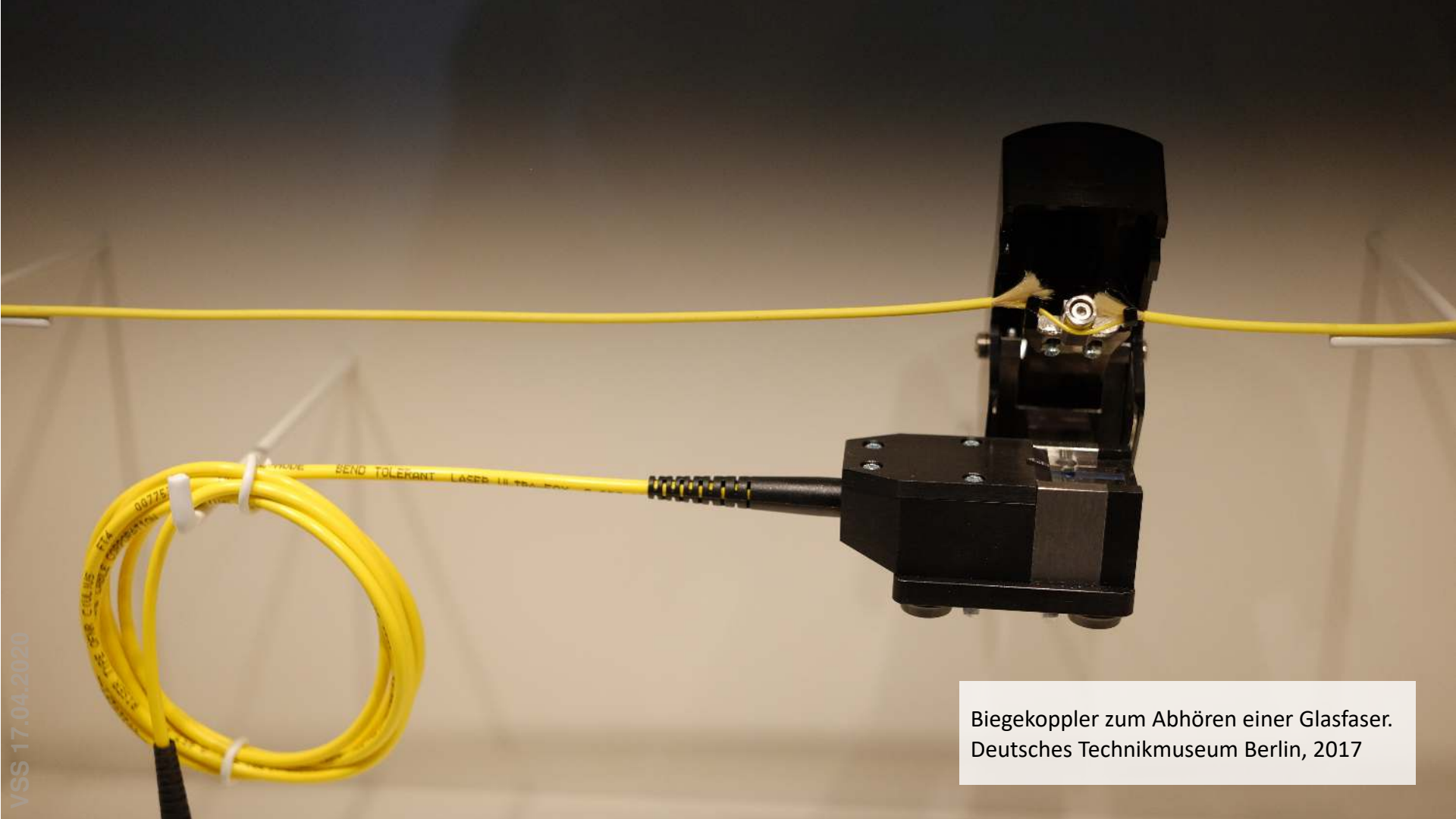
Rechner



Übertragungswege







Biegekoppler zum Abhören einer Glasfaser.
Deutsches Technikmuseum Berlin, 2017

Angriffspunkte

Rechner

Übertragungswege

von außen:
Abstrahlung

Angreifer kann nur
Vertraulichkeit
verletzen

Leitungen

Funkstrecken

Angreifer kann alle drei
Schutzziele verletzen

- Vertraulichkeit
- Integrität
- Verfügbarkeit

von innen:
Malware

Sicherheit: Abgrenzung von Security & Safety

Security Schutz gegen beabsichtigte Angriffe	Safety Schutz vor unbeabsichtigten Ereignissen
Vertraulichkeit • Abhörsicherheit • Sicherheit gegen unbefugten Gerätezugriff • Anonymität • Unbeobachtbarkeit	Fehlertoleranz **Verfügbarkeit** • Funktionssicherheit • Technische Sicherheit • Schutz vor Überspannung, Überschwemmung, Temperaturschwankungen • Schutz vor Spannungsausfall
Integrität • Übertragungsintegrität • Zurechenbarkeit • Abrechnungsintegrität	
Verfügbarkeit • Ermöglichen von Kommunikation	**Sonstige Schutzziele** • Maßnahmen gegen hohe Gesundheitsbelastung • …

Sicherheit: Informationssicherheit vs. Technische Sicherheit

Informationssicherheit

Freiheit v. Gefährd. f.d. *Daten* des IT-Systems

Vertraulichkeit

- Abhörsicherheit
- Sicherheit gegen unbefugten Gerätezugriff
- Anonymität
- Unbeobachtbarkeit

Integrität

- Übertragungsintegrität
- Zurechenbarkeit
- Abrechnungsintegrität

Verfügbarkeit

- Ermöglichen von Kommunikation

Technische Sicherheit

Freiheit v. Gefährd. f.d. *Umgebung* des IT-Systems

Verfügbarkeit

- Funktionssicherheit
- Technische Sicherheit
- Schutz vor Überspannung, Überschwemmung, Temperaturschwankungen
- Schutz vor Spannungsausfall

Zuverlässigkeit (reliability)

- Korrektheit

Sonstige Schutzziele

- Maßnahmen gegen hohe Gesundheitsbelastung
- …

Schutzziele der Daten- und IT-Sicherheit

Voydock, Kent, 1983

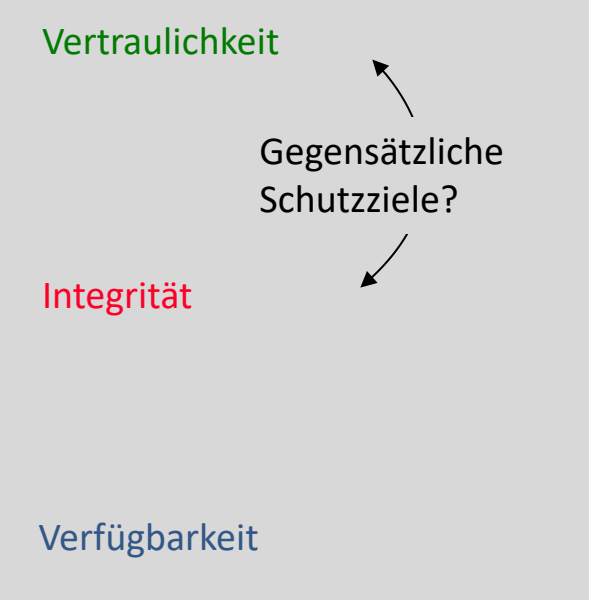
- Klassische IT-Sicherheit berücksichtigt im Wesentlichen Risiken, die durch *regelwidriges Verhalten* in IT-Systemen entstehen.

Vertraulichkeit	unbefugter Informationsgewinn
Gegensätzliche Schutzziele?	
Integrität	unbefugte Modifikation
Verfügbarkeit	unbefugte Beeinträchtigung der Funktionalität

Mehrseitige Sicherheit

Müller et al., 1997

- Mehrseitige Sicherheit bedeutet die Einbeziehung der Schutzinteressen aller Beteiligten sowie das Austragen daraus resultierender Schutzkonflikte.



- Voraussetzung
  - regelwidriges Verhalten hält Systeme und Nutzer schadlos
- Ziel
  - gegensätzliche Sicherheitsinteressen werden erkannt, Lösungen ausgehandelt und durchgesetzt

Schutzziele der mehrseitigen Sicherheit

Kommunikationsgegenstand Was?, Worüber? Inhaltsdaten	Kommunikationsumstände Wann?, Wo?, Wer? Verkehrsdaten
Vertraulichkeit Verdecktheit Inhalte	Anonymität Unbeobachtbarkeit Sender, Ort, Empfänger
Integrität Inhalte	Zurechenbarkeit Rechtsverbindlichkeit Absender, Bezahlung, Empfänger
Verfügbarkeit Inhalte	Erreichbarkeit Nutzer, Rechner

Vertraulichkeit: Verfahren und Algorithmen

Was wird geschützt?

Beispiele für Verfahren

Beispiele für Algorithmen

Verfahren des technischen Datenschutzes

Inhaltsdaten

Vertraulichkeit

Verschlüsselung

Inhalte

DES, 3-DES, OTP, IDEA, AES, RSA, ElGamal, …

Verdecktheit

Steganographie

Inhalte + Existenz

F5, …

Verkehrsdaten

Anonymität
Unbeobachtbarkeit

Sender

Ort

Empfänger

Web-Anonymisierer, Remailer, anonyme Zahlungssysteme

Pseudonyme, Proxies, umkodierende Mixe, DC Netz, Private Information Retrieval, …

Integrität und Zurechenbarkeit, Rechtsverbindlichkeit

Verfahren des technischen Datenschutzes

Verfahren

Algorithmen

Kommunikationsumstände
Wann?, Wo?, Wer?
Verkehrsdaten

Inhaltsdaten

Integrität

Zurechenbarkeit
Rechtsverbindlichkeit

auch:
Authentizität
Unabstreitbarkeit

Inhalte

Message Authentication Codes

Challenge-Response-Authentikation

Absender

Bezahlung

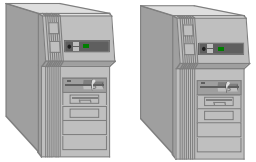
Empfänger

Digitale Signaturen

RSA, ElGamal, …

Verfügbarkeit: Redundanz und Diversität

- Redundanz
  - Mehrfache Auslegung von Systemkomponenten
  - Bei Ausfall übernimmt Ersatzkomponente



Beispiel: Doppelung

- Diversität
  - Verschiedenartigkeit der Herkünfte
  - Tolerieren von systematischen Fehlern und verdeckten trojanischen Pferden
  - Unabhängige Entwicklung von redundanten (Software)-Komponenten

Verfügbarkeit

Inhalte

Erreichbarkeit

Nutzer

Rechner

Schutzziele: Definitionen

Vertraulichkeit: Geheimhaltung von Daten während der Übertragung. Niemand außer den Kommunikationspartnern kann den Inhalt der Kommunikation erkennen.

Verdecktheit: Versteckte Übertragung von vertraulichen Daten. Niemand außer den Kommunikationspartnern kann die Existenz eines vertraulichen Inhalts erkennen.

Anonymität: Nutzer können Ressourcen und Dienste benutzen, ohne ihre Identität zu offenbaren. Selbst der Kommunikationspartner erfährt nicht die Identität.

Unbeobachtbarkeit: Nutzer können Ressourcen und Dienste benutzen, ohne dass andere dies beobachten können. Dritte können weder das Senden noch den Erhalt von Nachrichten beobachten.

Integrität: Modifikationen der kommunizierten Inhalte (Absender eingeschlossen) werden durch den Empfänger erkannt.

Zurechenbarkeit: Sendern bzw. Empfängern von Informationen kann das Senden bzw. der Empfang der Informationen bewiesen werden. Wechselwirkungen zwischen Schutzzielen

Rechtsverbindlichkeit: Ein Nutzer kann rechtlich belangt werden, um seine Verantwortlichkeiten innerhalb einer angemessenen Zeit zu erfüllen.

Verfügbarkeit: Nutzbarkeit von Diensten und Ressourcen, wenn ein Teilnehmer sie benutzen will.

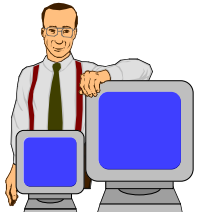
Erreichbarkeit: Zu einer Ressource (Nutzer oder Maschine) kann Kontakt aufgenommen werden, wenn gewünscht.

Einseitige oder mehrseitige Sicherheit?

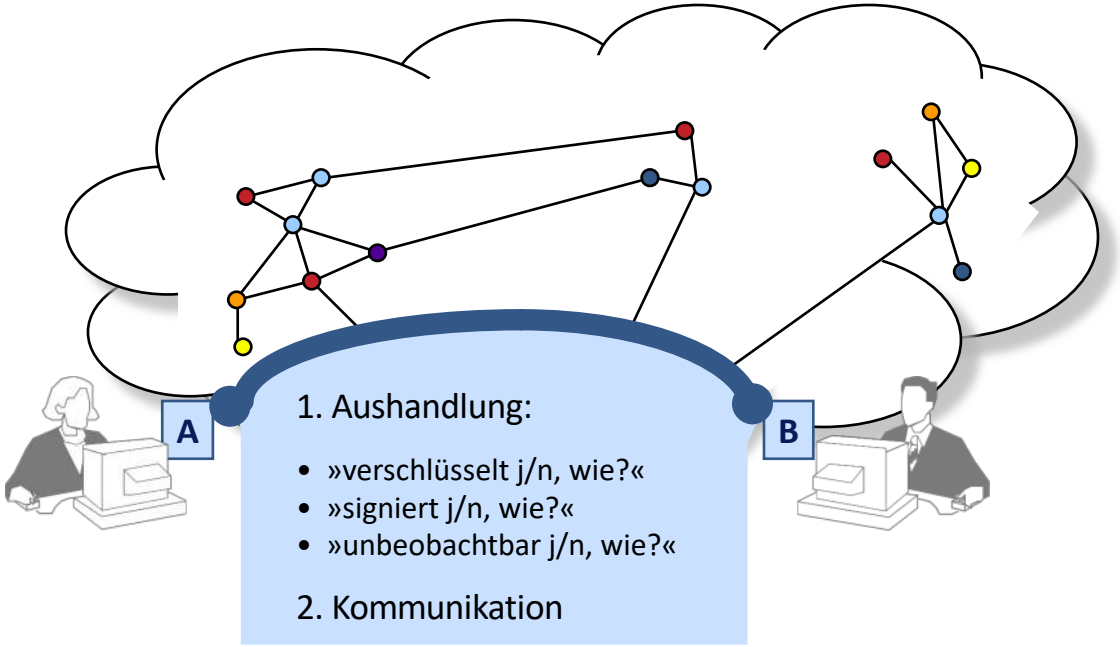
Kommunikationspartner haben nicht immer gleiche Sicherheitsinteresssen

Kunde | **Händler**

- Digitale Signatur
  - Der Kunde soll an seine Bestellung gebunden sein.
  - Der Händler soll an meine Bestellung gebunden sein.
- Digitale Signatur
- Vertrauen nötig
  - Pseudonymität: Treuhänder kennt Identität des Kunden, prüft Ware und Geld vor Lieferung
  - Der Kunde soll sich identifizieren.
  - Der Bezahlvorgang soll sicher sein (Kein Betrug durch Kunden).
  - Ich möchte anonym bleiben, solange ich nichts kaufe.
  - Der Zustand der Ware soll einwandfrei sein, sonst: Geld zurück!
- Anonyme Zahlungssysteme
  - Ich möchte anonym bleiben beim Einkauf.
- Vertrauen nötig
  - Selbstverpflichtung, P3P
  - Der Händler soll keine Kundenprofile anlegen dürfen.



Formulierung und Aushandlung



Mehrseitige Sicherheit

- Mehrseitige Sicherheit bedeutet die Einbeziehung der Schutzinteressen aller Beteiligten sowie das Austragen daraus resultierender Schutzkonflikte beim Entstehen einer Kommunikationsverbindung.
- Vorgehen
  - Sicherheitsinteressen formulieren
    - Setzt Verständnis des Benutzers voraus
    - Gute Bedienoberflächen sind nötig
  - Konflikte erkennen und Lösungen aushandeln
    - Setzt entsprechende Tools und
    - Technische Protokolle voraus
  - Sicherheitsinteressen durchsetzen
    - Anwender brauchen Werkzeuge zum Selbstschutz

Ziel: Möglichst wenig Vertrauen in andere setzen müssen, d.h.
»Sicherheit mit minimalen Annahmen über andere«

Vor wem ist zu schützen? — Rollen des Angreifers

Rollen des Angreifers:

1. Outsider (Außenstehende)

2. Insider

Innentäter:

- Benutzer des Systems
- Kommunikationspartner
- Betreiber des Systems
- Wartungsdienst
- Produzenten des Systems
- Entwerfer des Systems

Transitive Trojanische Pferde:

- Produzenten oder Entwerfer der Entwurfs- und Produktionshilfsmittel
- Produzenten oder Entwerfer der Entwurfs- und Produktionshilfsmittel der Entwurfs- und Prod…
- Entwerfer der …

Transitiv: Schaden pflanzt sich über einen oder mehrere Entwicklungsschritte fort

Angreifermodell

Das Angreifermodell definiert die maximal berücksichtigte Stärke eines Angreifers, gegen den ein Schutzmechanismus gerade noch wirkt. Schutz vor einem allmächtigen Angreifer ist unmöglich.

Aspekte des Angreifermodells:

Rollen	*roles*		Außenstehender, Innentäter, auch kombiniert
Verhalten	*behaviour*		passiv/beobachtend, aktiv/verändernd
Fähigkeiten	*abilities*		Wissen über das System und Verbreitung im System
Absicht	*intention*		versehentlich, fahrlässig, absichtlich
Ressourcen	*resources*		Zeit und Geld, Rechenkapazität

unbeschränkt: informationstheoretisch
beschränkt: komplexitätstheoretisch

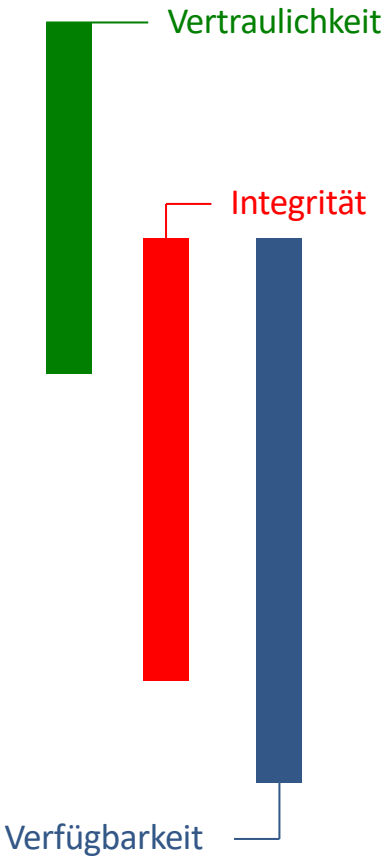
Verhalten des Angreifers

	Aktiv	Passiv	Verletzbare Schutzziele
Beobachtend	(√) etwas Berechtigtes tun und dabei Information gewinnen	√	Vertraulichkeit
Verändernd	√	Ø	Vertraulichkeit Integrität Verfügbarkeit

Verhalten des Angreifers und Angriffsformen

- Passive/beobachtende Angriffe
  - Lauschangriff (eavesdropping, sniffing)
  - Verkehrsflussanalyse (traffic analysis)

- Aktive/verändernde Angriffe
  - Maskerade (masquerading)
    - Man-in-the-middle attack
  - Verändern von Daten (modification)
  - Einfügen von Daten (injection, spoofing)
    - Wiederholen (replay)
    - Fluten (flooding, spamming)
  - Dienstverweigerung (denial of service)



Typische Angriffsfolge

1. Informationsgewinnung
   - IP-Adressen, Passwörter, Eindringpunkte

Beispiele: Social Engineering, Security Scanner, Port Scanner

2. Angriff (über das Netz)
   - Ausnutzen von Schwächen, Missbrauch von Daten

3. Erweiterung der Rechte
   - insb. Schaffen einer unauffälligen Hintertür

4. Spuren verwischen
   - Löschen oder Manipulieren von Log-Dateien

Schutz z.B. TrueWORM – Write Once Read Multiple mittels CD-R, DVD±R, »Datendiode«

Angriffsfolge nach der Lockheed Martin Cyber Kill Chain

Lockheed Martin, 2015

1. Informationsgewinnung
   - IP-Adressen, Passwörter, Eindringpunkte

2. Angriff (über das Netz)
   - Ausnutzen von Schwächen, Missbrauch von Daten

3. Erweiterung der Rechte
   - insb. Schaffen einer unauffälligen Hintertür

4. Spuren verwischen
   - Löschen oder Manipulieren von Log-Dateien

1. Reconnaissance: Auskundschaften des Ziels
2. Weaponization: Vorbereitung zur Übertragung der Schadsoftware
3. Delivery: Übertragung der Schadsoftware (E-Mail, USB-Speicher)
4. Exploitation: Initialen Zugang zum Opfer herstellen, Ausnutzen der Schwachstelle beim Opfer
5. Installation: permanenten Zugang zum Opfersystem schaffen (z.B. Backdoors)
6. Command and Control: Fernsteuern des Opfersystems
7. Actions on Objective: (Massenhafte) Angriffe im Netz ausführen (z.B. mittels Botnetz)

Cyber Kill Chain

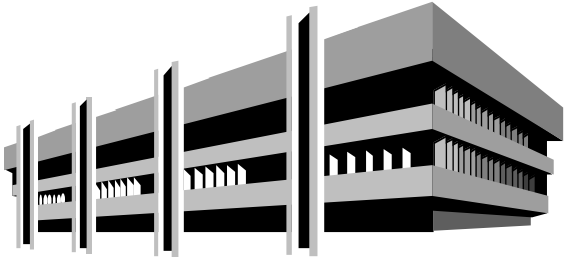
Rechnersicherheit

Physische Sicherheit

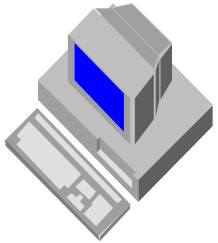
- Physische Sicherheit
  - Alle technischen Schutzmaßnahmen benötigen eine physische »Verankerung« in Form eines Systemteils, auf den der Angreifer keinen physischen Zugriff hat.

- Physische Sicherheit zu erhalten, gelingt bestenfalls auf Zeit.

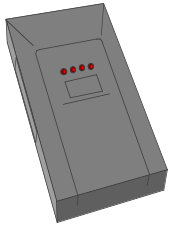
Rechenzentrum



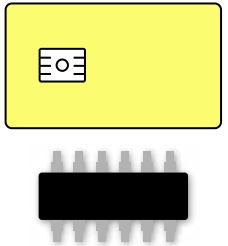
einzelner Rechner



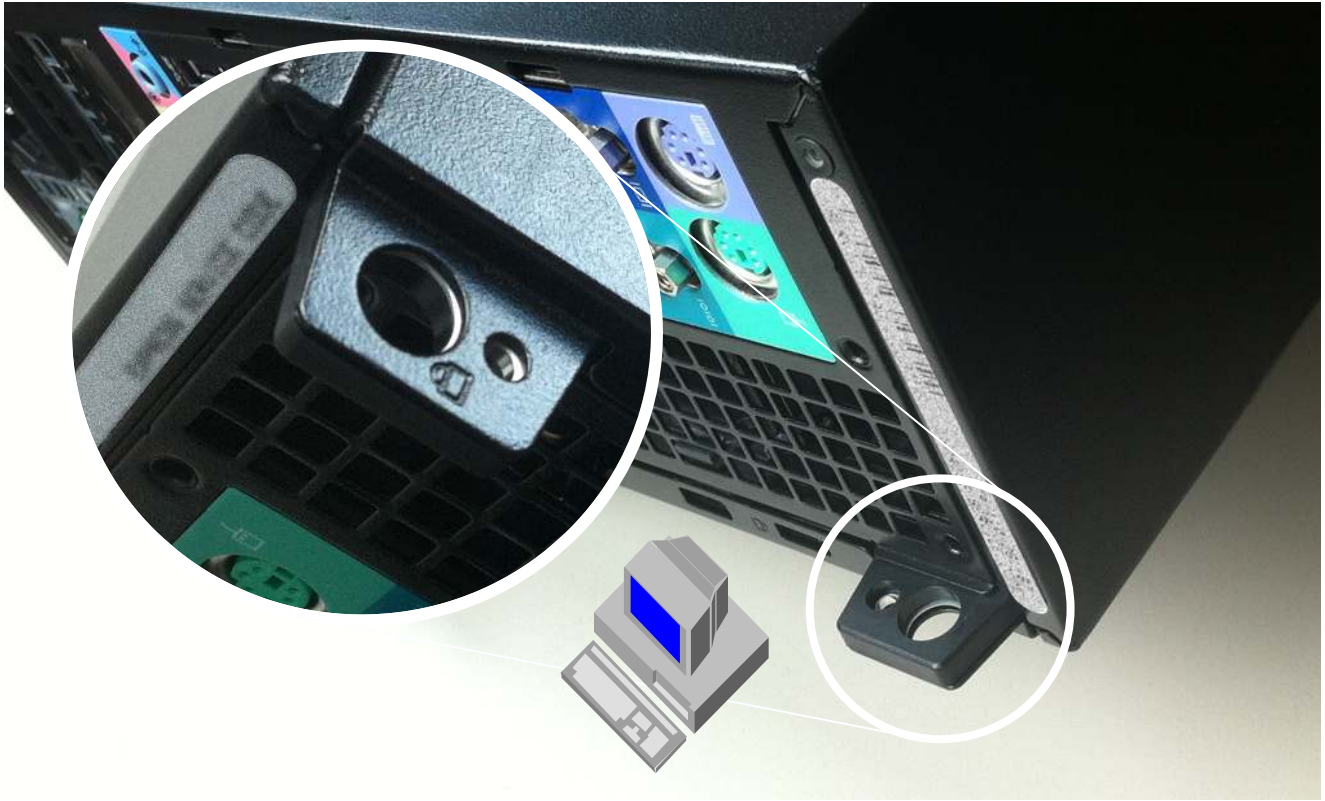
Sicherheitsmodul



Chipkarte, TPM

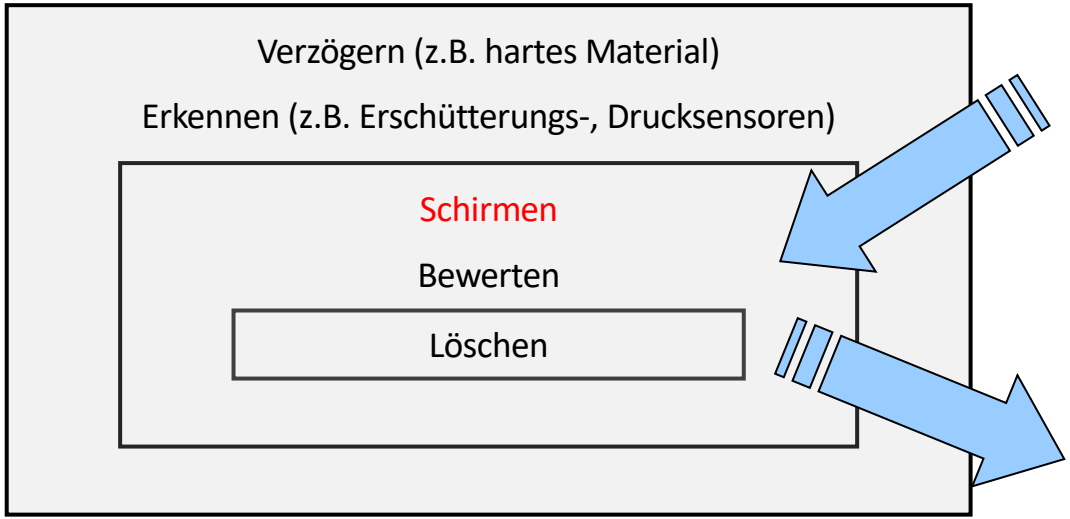


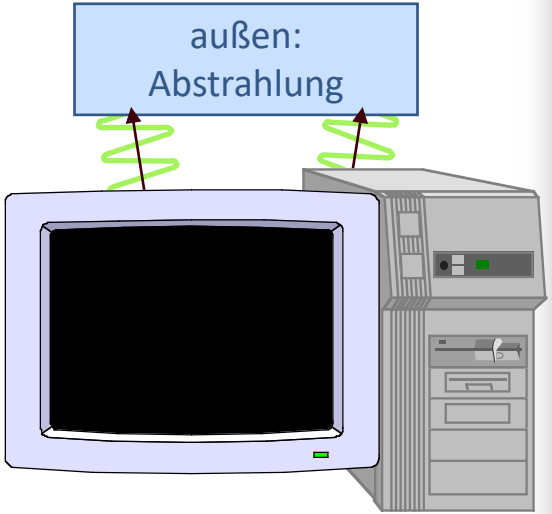
Physische Sicherheit

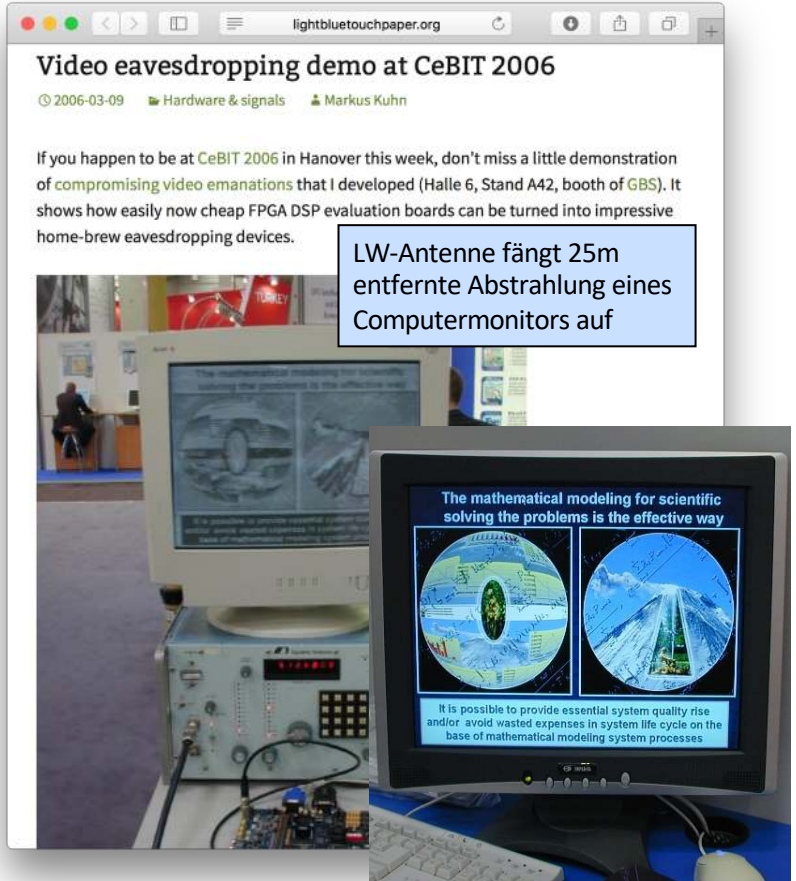


Physische Sicherheit: Grundfunktionen

- Schutz gegen beobachtende Angriffe:
  - Schirmung (Schutz vor elektromagnetischer Abstrahlung, Schutz vor Seitenkanalangriffen: Energie- und Rechenzeitverbrauch unabhängig von den zu schützenden Geheimnissen)
- Schutz gegen verändernde Angriffe:
  - Erkennen, Bewerten, Verzögern und ggf. Löschen der geheimen Informationen.



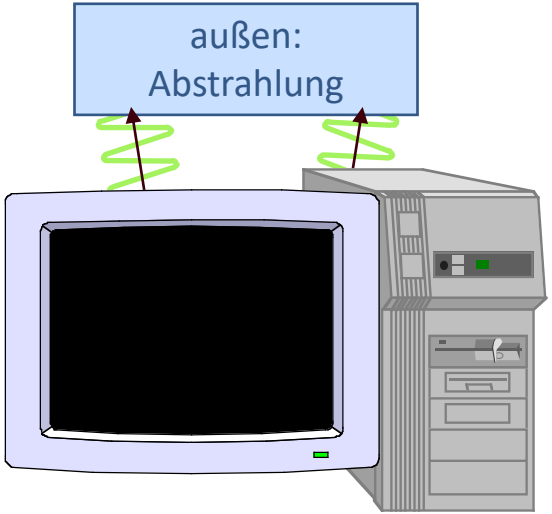




LW-Antenne fängt 25m entfernte Abstrahlung eines Computermonitors auf

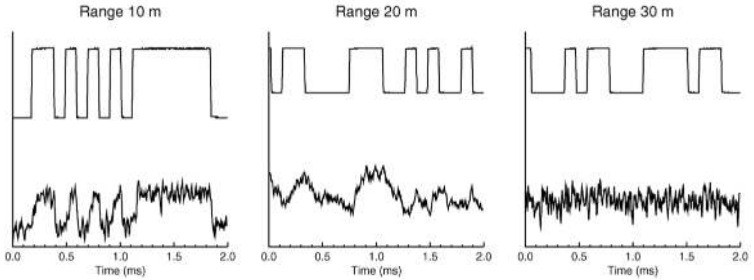
Für VGA (2006):
https://www.lightbluetouchpaper.org/2006/03/09/video-eavesdropping-demo-at-cebit-2006/

Auch HDMI (2018)
https://www.heise.de/security/meldung/l-f-Hackerin-demonstiert-Van-Eck-Phreaking-trotz-HDMI-4123699.html



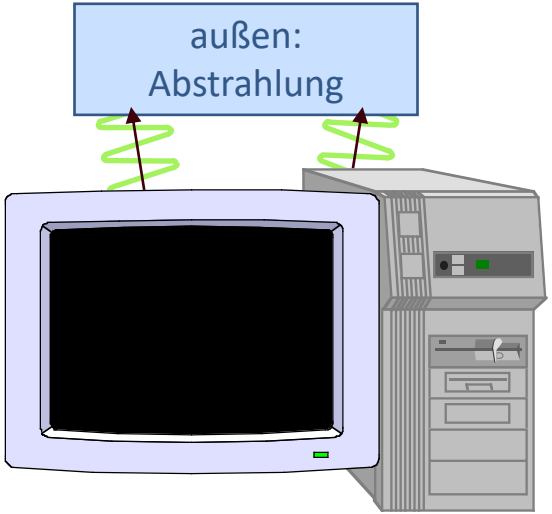
auch optische Abstrahlung in Betracht ziehen!

Flackernde LEDs an (alten) Modems häufig direkt an den Datenbus angeschlossen

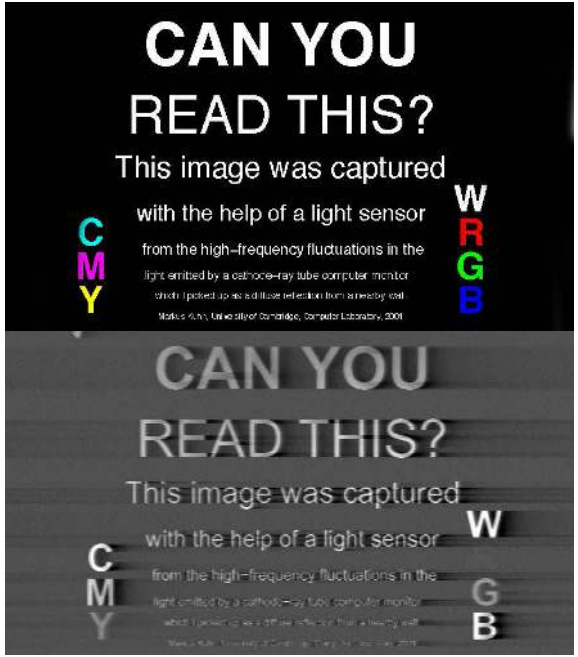


Degradation of the optical signal with increasing distance from the target. The data rate was 9600 bits/s. Quelle: J. Loughry, D. A. Umphress: Information Leakage from Optical Emanations. ACM Transactions on Information and System Security 5/3 (2002) 262–289.

Gilt nicht für schnelle Router, Switches und Netzwerkkarten

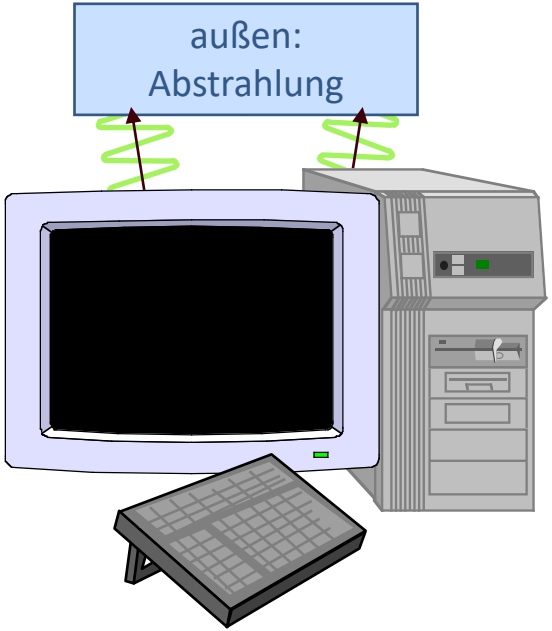


auch optische Abstrahlung in Betracht ziehen!



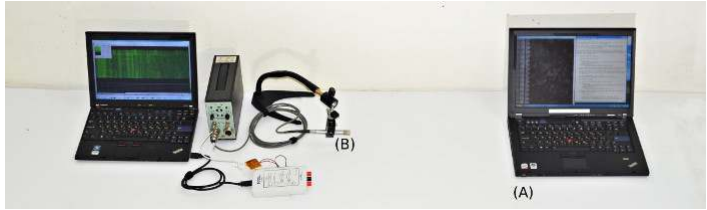
Quelle: Markus G. Kuhn: Optical Time-Domain Eavesdropping Risks of CRT Displays. Proc. IEEE Symposium on Security and Privacy, 2002.

Rechner

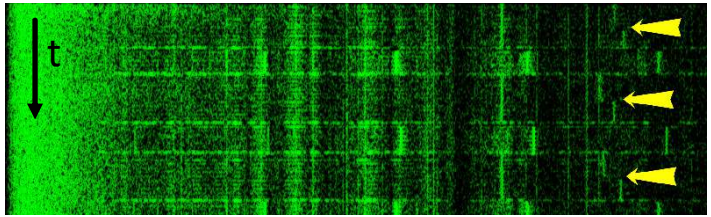


… acoustic cryptanalysis?

RSA key extraction attack by use of acoustic emanations by recording the vibration of electronic components of a computer

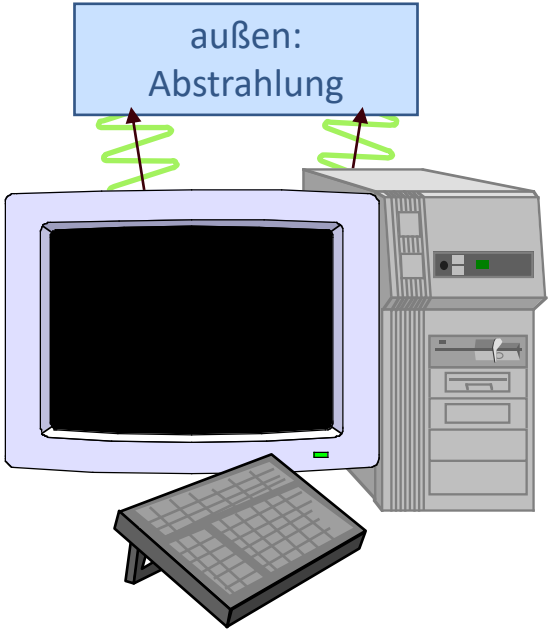


Basic idea: determine specific function calls and estimate states (Bit=0 or 1) during encryption



Source: https://www.tau.ac.il/~tromer/acoustic/ (2013)

Rechner



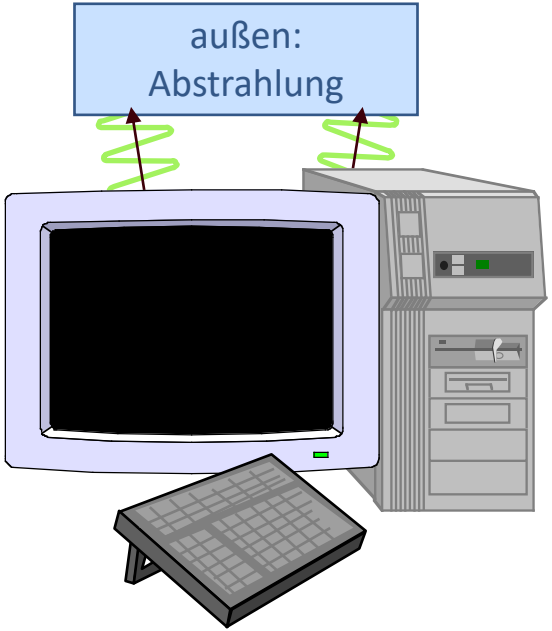
… auch Tastaturklappern?

Attacks by keyboard acoustic emanations

- 10-minute sound recording of a user typing English text using a keyboard
  - recovers <= 96% of typed chars
  - makes use of machine learning
  - no need for a labeled training
- can even recognize random text
  - < 20 attempts for 90% recognition of 5-character random passwords
  - < 75 attempts for 80% recognition of 10-character random passwords

Source: Li Zhuang, Feng Zhou, J. D. Tygar: Keyboard Acoustic Emanations Revisited. Proc. 12th ACM Conference on Computer and Communications Security, November 2005, pp. 373-382.

Abschirmen



… und Vibrationen!

(sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers

Philip Marquardt*
MIT Lincoln Laboratory
244 Wood Street, Lexington, MA USA
philip.marquardt@ll.mit.edu

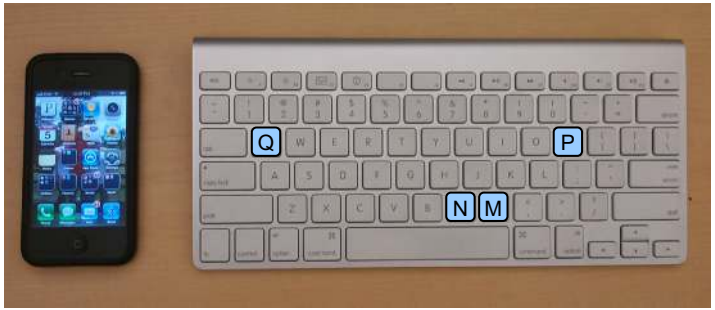
Arunabh Verma, Henry Carter and Patrick Traynor
Georgia Institute of Technology
{arunabh.verma@, carterh@, traynor@cc.}gatech.edu

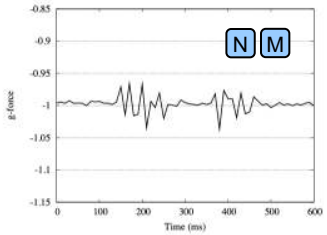
ABSTRACT

Mobile phones are increasingly equipped with a range of highly responsive sensors. From cameras and GPS receivers to three-axis accelerometers, applications running on these devices are able to experience rich interactions with their environment. Unfortunately, some applications may be able to use such sensors to monitor their surroundings in unintended ways. In this paper, we demonstrate that an application with access to accelerometer readings on a modern mobile phone can use such information to recover text entered on a nearby keyboard. Note that unlike previous emanation recovery papers, the accelerometers on such devices sample at near the Nyquist rate, making previous techniques unworkable

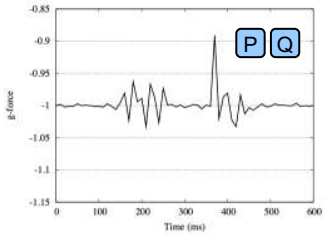
to web browsers, a progressively sophisticated set of sensors are enabling these devices to more actively interface with the world around them. From gestures captured by accelerometers for games to augmented reality applications displaying metadata tags on video from the camera in real-time, mobile phones are becoming adept at capturing and harnessing rich features and data from their surroundings.

Unfortunately, the array of sensors in these devices can also be used in unintended ways. As many have suggested, malware could potentially gain access to a mobile phone's camera to take photos or video of the owner and their surroundings [16, 37]. In cases where the camera may not be pointed at an interesting target, a malicious



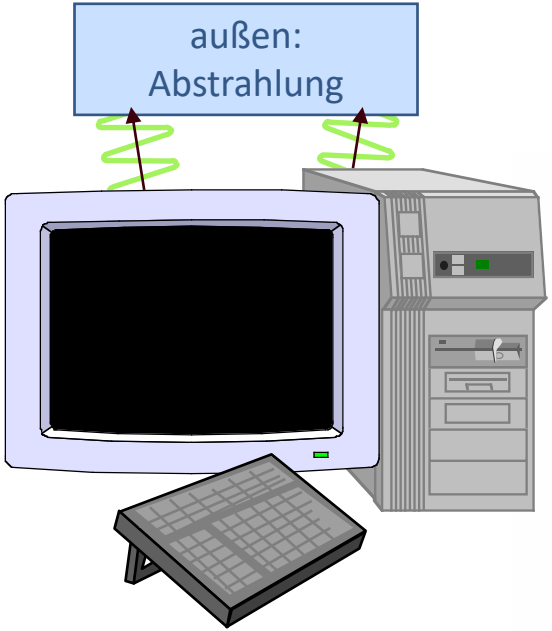


Character pair "nm"



Character pair "pq"

Abschirmen



… auch Schmierspuren auf Smartphones!

Smudge Attacks on Smartphone Touch Screens

Adam J. Aviv, Katherine Gibson, Evan Mossop, Matt Blaze, and Jonathan M. Smith
Department of Computer and Information Science – University of Pennsylvania
{aviv,gibsonk,emossop,blaze,jms}@cis.upenn.edu

Abstract

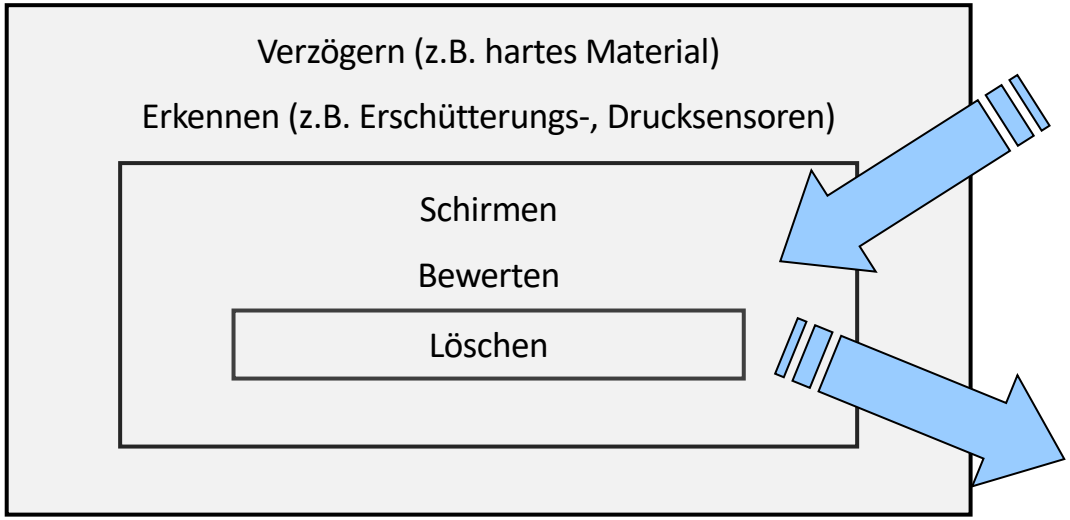
Touch screens are an increasingly common feature on personal computing devices, especially smartphones, where size and user interface advantages accrue from consolidating multiple hardware components (keyboard, number pad, *etc.*) into a single software definable user interface. Oily residues, or *smudges*, on the touch screen surface, are one side effect of touches from which frequently used patterns such as a graphical password might be inferred.

We believe smudge attacks are a threat for three reasons. First, smudges are surprisingly[1] persistent in time. Second, it is surprisingly difficult to incidentally obscure or delete smudges through wiping or pocketing the device. Third and finally, collecting and analyzing oily residue smudges can be done with readily-available equipment such as a camera and a computer[2].

To explore the feasibility of smudge attacks against the Android password pattern, our analysis begins by evaluating the conditions by which smudges can be photo-

Physische Sicherheit: Grundfunktionen

- Schutz gegen beobachtende Angriffe:
  - Schirmung (Schutz vor elektromagnetischer Abstrahlung, Schutz vor Seitenkanalangriffen: Energie- und Rechenzeitverbrauch unabhängig von den zu schützenden Geheimnissen)
- Schutz gegen verändernde Angriffe:
  - Erkennen, Bewerten, Verzögern und ggf. Löschen der geheimen Informationen.

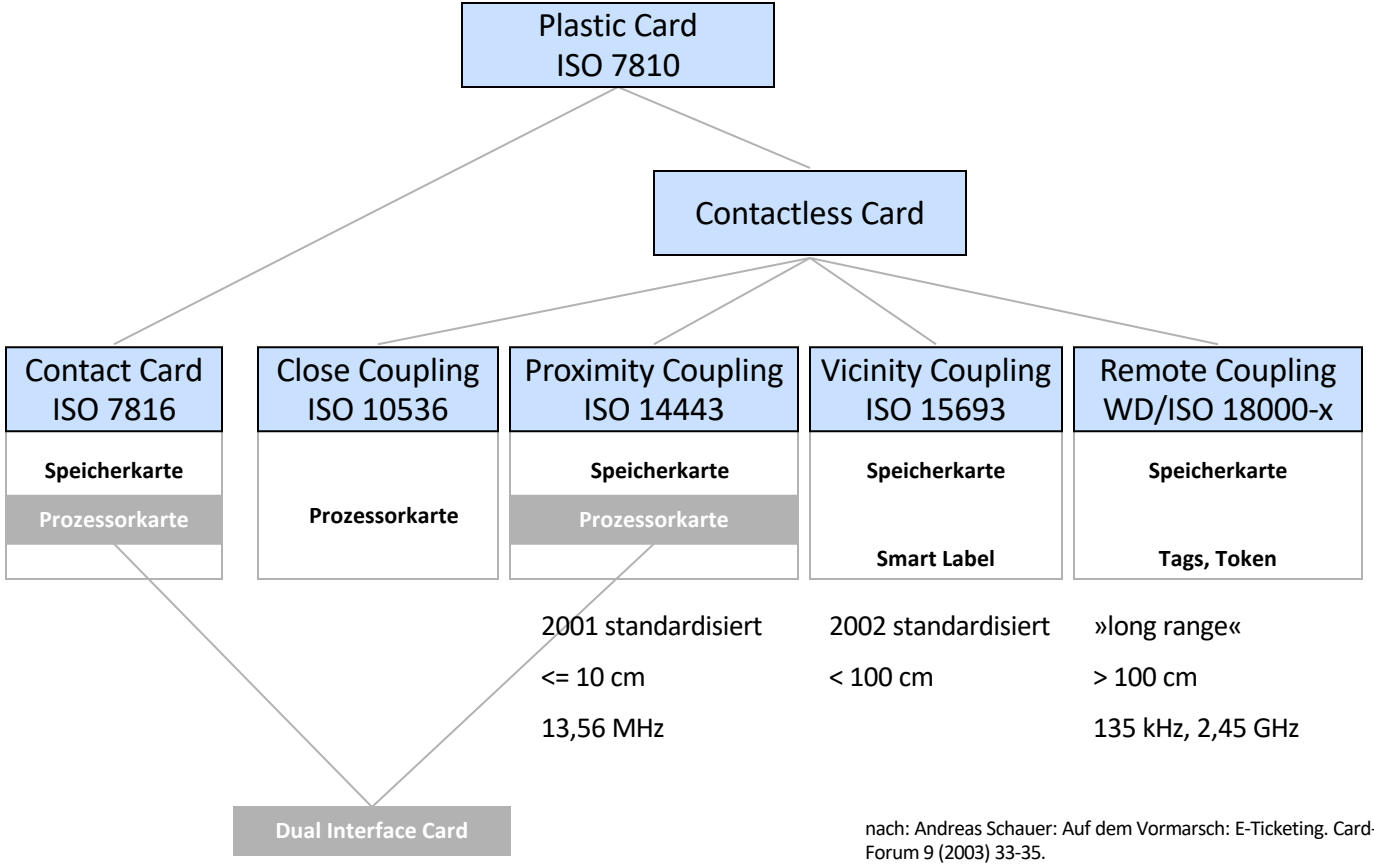


- Sicherheitsmodul

Bild: www.lampertz.de

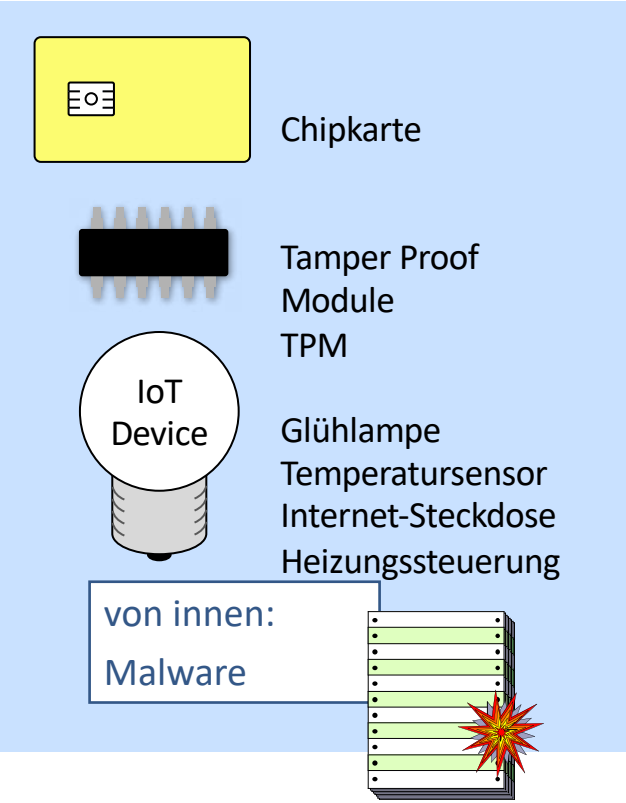
  - Brandschutz
  - Zutrittsschutz
  - Klimatisierung
  - Unabhängige Stromversorgung

Chipkarten, RFID-Tags



nach: Andreas Schauer: Auf dem Vormarsch: E-Ticketing. Card-Forum 9 (2003) 33-35.

Physische Sicherheit von IoT Devices, Chipkarten und TPMs



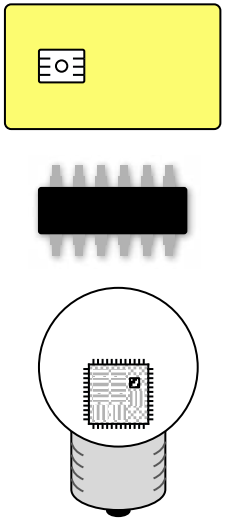
Physische Sicherheit

- Alle technischen Schutzmaßnahmen benötigen eine physische »Verankerung« in Form eines Systemteils, auf den der Angreifer keinen physischen Zugriff hat.
- Physische Sicherheit zu erhalten, gelingt bestenfalls auf Zeit.

Probleme

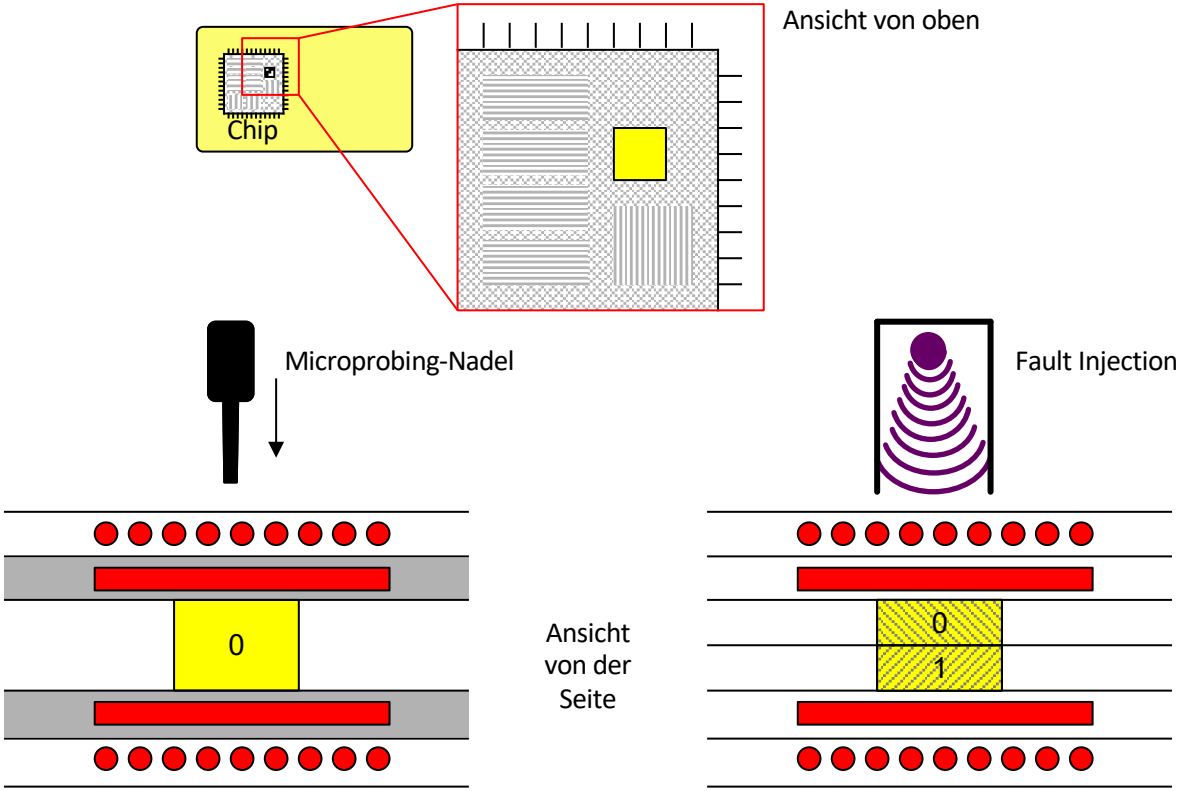
- Schirmung schwierig
- Angriffserkennung schwierig
- kein Löschen vorgesehen, selbst bei Stromversorgung

Beispiele für Angriffe



- Zerstörend
  - Abschleifen und Anätzen der Schutzschichten
  - Reverse Engineering: Untersuchung der Strukturen unter Elektronenmikroskop, wenn Funktion unbekannt
  - Microprobing-Nadel
  - Fault Injection: gezielte Manipulation von Bits durch Beschuss mit elektromagnetischer Strahlung
- Zerstörungsfrei meist sog. sidechannel attacks
  - Messung des Energieverbrauchs: power analysis
  - Messung der benötigten Rechenzeit: timing attacks

Zerstörende Angriffe und Schutz davor



„Aber wer hat denn schon ein Elektronenmikroskop zu Hause?“

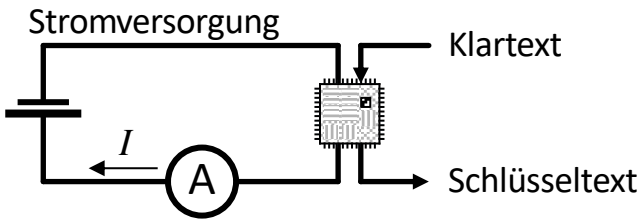
4. Bereite alles für den ersten Test vor

Pictures: Private Archive M.Janke, P.Laackmann

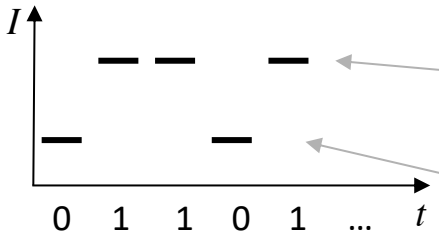
Page 32

Quelle: http://events.ccc.de/congress/2013/Fahrplan/system/attachments/2227/original/25_Jahre_Chipkartenangriffe-Marcus_Janke_Peter_Laackmann.pdf

Timing Attack / Power Analysis (Skizze)



Angriffsziel: Key ermitteln



1 = hoher Stromverbrauch
0 = niedriger Stromverbrauch

Schlüsselbits sind direkt auslesbar.

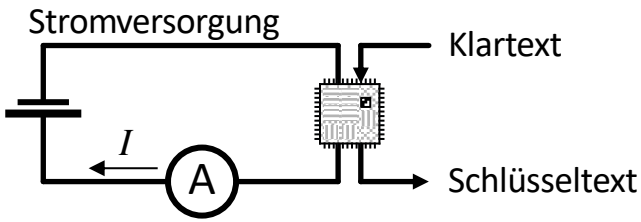
Schlüssel ist dem Angreifer unbekannt:

```
key = array of bit
    = ( 0, 1, 1, 0, 1, ...)
```

Quellcode ist dem Angreifer bekannt:

```
encrypt (Klartext, key) {
    ...
    for i = 1 to length(key){
        ...
        if (key[i] == 1){
            <code1>
            // ΔI ≈ I1
            // Δt ≈ t1
        } else {
            <code2>
            // ΔI ≈ I2
            // Δt ≈ t2
        }
    }
}
```

Timing Attack / Power Analysis (Skizze)



Schlüssel ist dem Angreifer unbekannt:

```
key = array of bit
    = ( 0, 1, 1, 0, 1, ...)
```

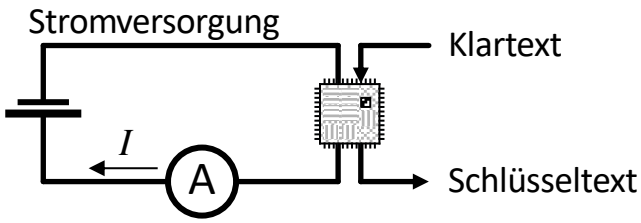
Quellcode ist dem Angreifer bekannt:

```
encrypt (Klartext, key) {
    ...
    for i = 1 to length(key){
        ...
        if (key[i] == 1){
            <code1>
            // ΔI ≈ I1
            // Δt ≈ t1
        } else {
            <code2>
            // ΔI ≈ I2
            // Δt ≈ t2
        }
    }
}
```

Auswege:

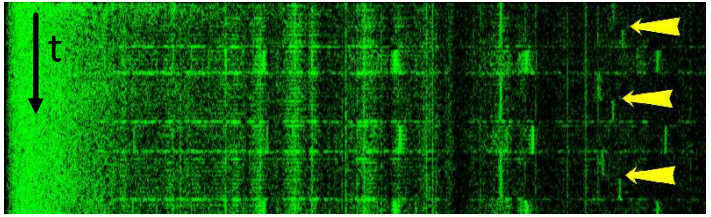
- interner Energiepuffer (wenn möglich)
- Verzweigungen vermeiden
- <code2> durch Dummy-Befehle der exakt gleichen Zykluszeit wie <code1> ersetzen
- Funktionen mit konstanter Ausführungszeit unabhängig von den Eingaben verwenden: Beispiel: django.utils.crypto.constant_time_compare(a,b)

Timing Attack / Power Analysis (Skizze)



Angriffsziel: Key ermitteln

Symbolische Darstellung für Abstrahlung:



Source: https://www.tau.ac.il/~tromer/acoustic/ (2013)

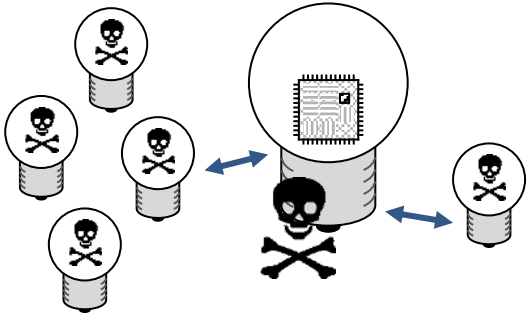
Schlüssel ist dem Angreifer unbekannt:

```
key = array of bit
    = ( 0, 1, 1, 0, 1, ...)
```

Quellcode ist dem Angreifer bekannt:

```
encrypt (Klartext, key) {
    ...
    for i = 1 to length(key){
        ...
        if (key[i] == 1){
            <code1>
            // ΔI ≈ I1
            // Δt ≈ t1
        } else {
            <code2>
            // ΔI ≈ I2
            // Δt ≈ t2
        }
    }
}
```

Physische Sicherheit und Internet of Things



Firmware der Hue-Lampen wird durch bösartige Malware ersetzt

Malware verbreitet sich über fehlerhafte ZigBee-Implementierung

Quelle: Ronen, O'Flynn, Shamir, Weingarten: IoT Goes Nuclear: Creating a ZigBee Chain Reaction, 2016

IT Mobiles Entertainment Wissen Netzpolitik Wirtschaft

TOPTHEMEN: ARTIKEL 13 ELEKTROAUTO DSGVO WINDOWS 10 ANDROID KI

Security 〉 7-Tage-News 〉 11/2016 〉 **Licht an, Licht aus: ZigBee-Wurm befällt smarte Glühbirnen**

Fehlerhafte ZigBee-Implementierung

Um den Angriff auszuführen, nutzen die Forscher zwei getrennte Sicherheitslücken in der Firmware der Hue-Birnen von Philips aus. Bei einer handelt es sich um eine Schwachstelle in der Umsetzung des Funkprotokolls ZigBee. Zusätzlich gelang es ihnen, bösartige Firmware mit den Schlüsseln von Philips zu unterschreiben und auf die Birnen zu laden. Die entsprechenden Schlüssel extrahierten sie mit Hilfe einer Seitenkanalattacke aus der Firmware einer Birne. Als die Forscher die Lücken im Juli an den Hersteller meldeten waren die meisten der bis dahin verkauften Birnen angreifbar, mit einem Update im Oktober hat Philips die Lücken mittlerweile geschlossen.

https://www.heise.de/security/meldung/Licht-an-Licht-aus-ZigBee-Wurm-befaellt-smarte-Gluehbirnen-3459004.html

Physische Sicherheit zu erhalten, gelingt bestenfalls auf Zeit.

Beispiel EC-Terminals:

- unzureichende Überprüfung des Schutzes des Hardware-Security-Moduls (HSM)
- Angreifer konnten erfolgreich Mesh-Gewebe entfernen

golem.de

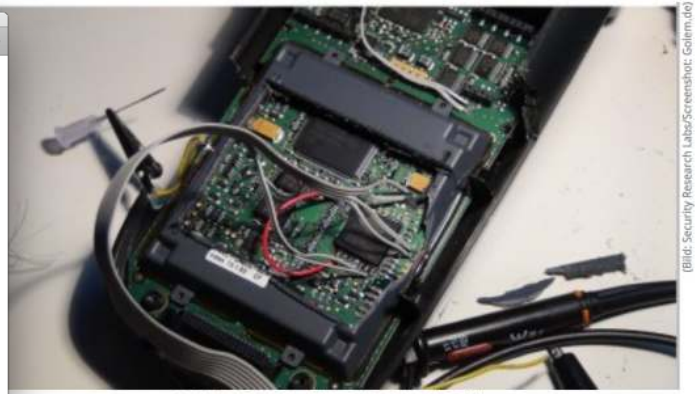
EC-TERMINALS

Hacker knacken Hardware-Security-Modul

32C3

Nicht nur die Protokolle zur Abwicklung des EC-Zahlungsverkehrs sind schwach, auch die verwendete Hardware ist offenbar nicht besonders gut gegen Angriffe abgesichert. Auf dem 32C3 stellten Berliner Hacker jetzt vor, wie sich das Hardware-Security-Modul eines EC-Terminals hacken lässt.

28. Dezember 2015, 10:40 Uhr, Hauke Gierow



(Bild: Security Research Labs/Screenshot: Golem.de)

Das HSM von EC-Terminals ist angreifbar - mit einigen Tricks.

golem.de

Das HSM beherbergt unter anderem den kryptographischen Message-Authentication-Code (MAC), mit dem sicherheitsrelevante Informationen wie die PIN abgesichert werden. Das HSM befindet sich unter einer Abdeckung - wird diese entfernt, "vergisst" der RAM alle Informationen, weil automatisch die Stromversorgung gekappt wird. Um zu prüfen, ob die Abdeckung intakt ist, ist an der Unterseite der Abdeckung ein dünnes, leitendes Mesh-Gewebe angebracht. Bei Beschädigungen wäre der Stromfluss unterbrochen - und das HSM stromlos.

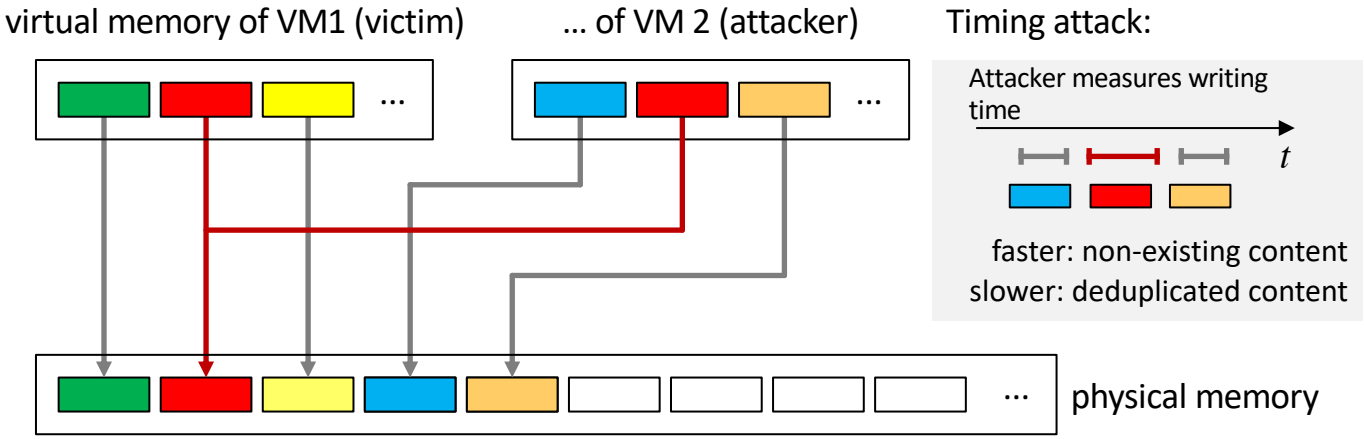
Mesh-Status wird nur an den Ecken überprüft

Doch wie die Hacker herausfanden, wird der Status des Mesh nur an den vier Ecken überprüft, nicht aber an den langen Kanten. Daher war es ihnen möglich, eine dünne Nadel unter die Abdeckung zu führen und dort den Prüfmechanismus für das Mesh außer Kraft zu setzen. Danach konnten sie die Abdeckung problemlos aufschneiden und den RAM des HSM auslesen. Dazu nutzten sie einen speziell angepassten Arduino.

https://www.golem.de/news/ec-terminals-hacker-knacken-hardware-security-modul-1512-118210.html

Memory deduplication attacks on virtual machines

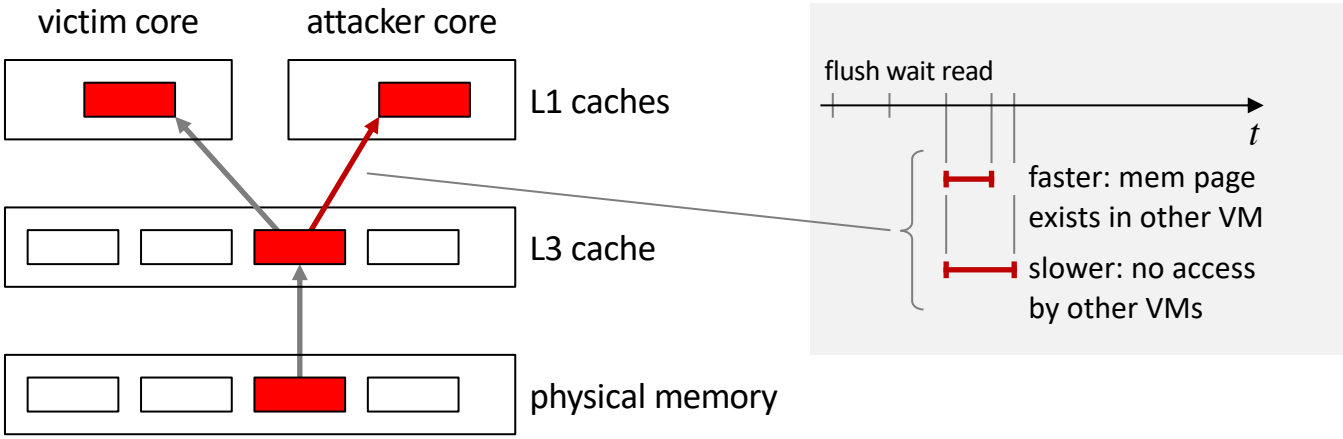
- Cloud Computing and memory deduplication
  - VMs on a host are often in control of different users
  - Memory deduplication can save physical memory
- Attacker measures writing time for memory page
  - knows whether or not a content exists in foreign VMs



Flush-and-reload attack on virtual machines

Yarom, Falkner, 2013

- Timing attack on L3 cache on Intel X86 processors
  - Flush memory page (delete it from all caches) and wait
  - Measure time to reload memory line
  - Repeat for other memory lines



Cold Boot und Hot Plug: Brechen der Festplattenverschlüsselung

- Cold Boot (2008): https://citp.princeton.edu/research/memory/
  - Voraussetzung: Hauptspeicher des Rechner enthält den (flüchtigen) Key der Festplattenverschlüsselung
  - Nach Ausschalten verbleibt (nutzbare) Restladung
  - Abkühlen verlangsamt Speicherremanenz
  - Angreifer hat genügend Zeit, den Speicher auszubauen und auszulesen

- Hot Plug (2012): http://www1.informatik.uni-erlangen.de/sed
  - Annahme: Festplattenverschlüsselung direkt auf der Platte; Hauptspeicher enthält nicht mehr den Key
  - Hausdurchsuchungsszenario: Rechner wird im laufenden Betrieb gefunden, nicht ausgeschaltet, die Festplatte (unter Strom) an einen externen Controller angeschlossen
  - einige Festplatten bemerken Controllerwechsel nicht

Identifikation von Menschen durch IT-Systeme

- Was der Mensch *ist*
  - Fingerabdruck
  - Aussehen*
  - eigenhändige Unterschrift*
  - Retina-Muster
  - …
- Was der Mensch *hat*
  - Papierdokument*
  - Metallschlüssel
  - Chipkarte
  - …
- Was der Mensch *weiß*
  - Passwort
  - Antworten auf Fragen
  - …

*=Ausweis (Multifaktorauthentifizierung)

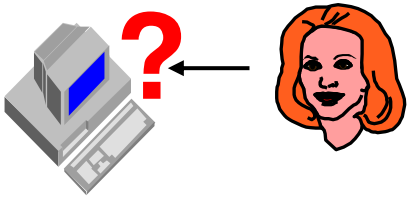


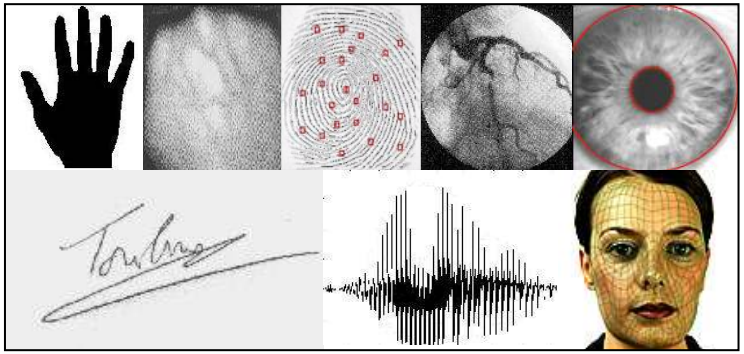
Bild: ntz, Heft 3-4/2006, S. 35

SecurID

Bild: http://www.rsasecurity.com

Biometrische Merkmale

- Physiologische
  - Handgeometrie
  - Handvenenmuster
  - Fingerabdruck
  - Retina
  - Iris
  - Gesicht
  - DNA
  - Ohrmuscheln
- Verhaltensbasierte
  - Handschrift
  - Stimme
  - Lippenbewegung
  - Tipp-Charakteristik
  - Gang

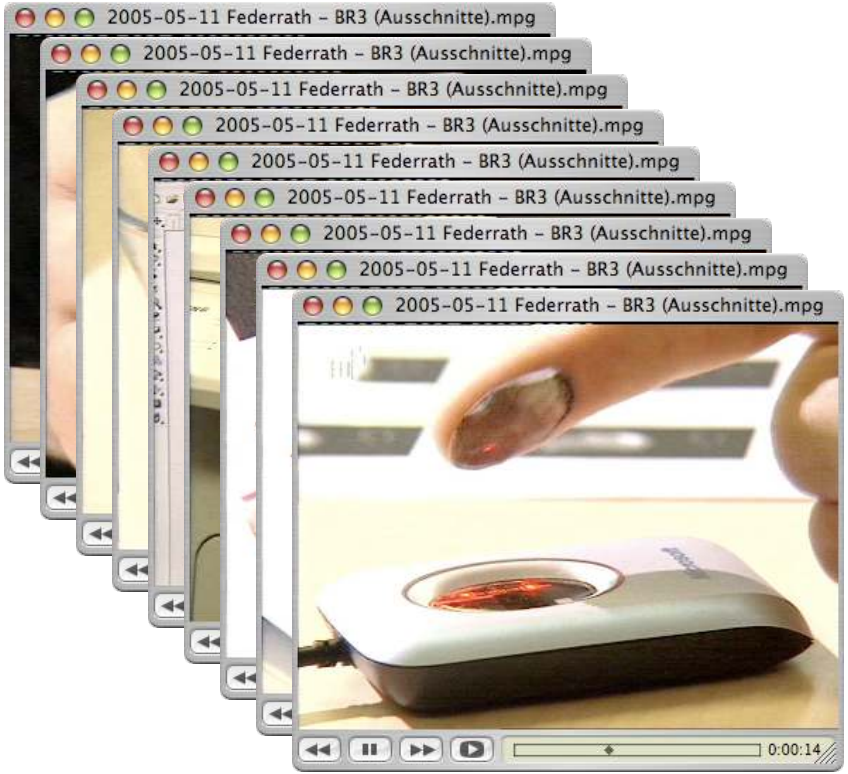


Bilder:
http://biometrics.cse.msu.edu/
http://www.atica.pm.gouv.fr/dossiers/documents/biometrie.shtml
http://www.br-online.de/wissen-bildung/thema/biometrie/koerper.shtml

Fälschen eines Fingerabdrucks

- Vom Chaos Computer Club im Jahre 2005 praktisch demonstriert
  - Fingerabdruck sichtbar machen
  - fotografieren
  - nachbearbeiten
  - ausdrucken
  - Leim drauf
  - warten
  - abziehen

  - Von uns im Rahmen einer Fernsehsendung praktisch nachvollzogen
  - Ergebnis: Es funktioniert wirklich (nicht).



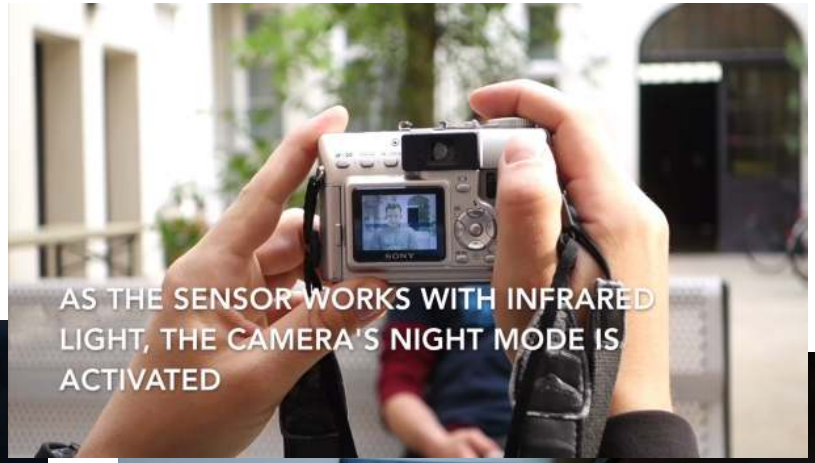
Fälschen eines Fingerabdrucks aus einem Foto

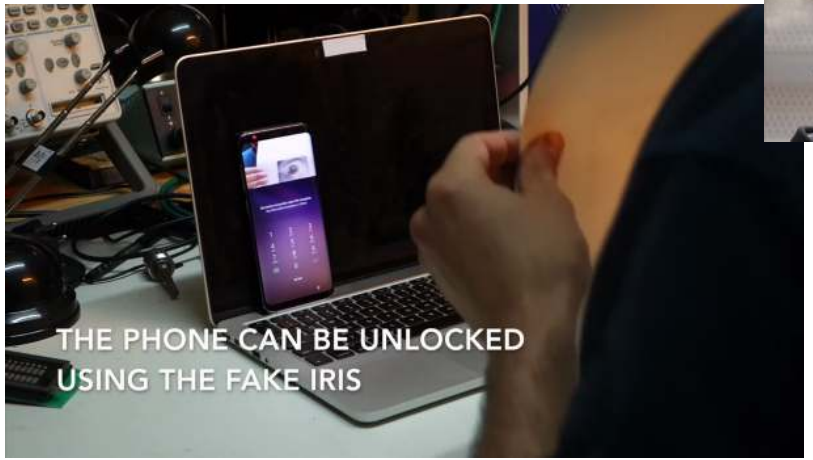
- Auf dem Chaos Communication Congress 2014 demonstriert.

Quelle: https://events.ccc.de/congress/2014/Fahrplan/events/6450.html

Iris-Erkennung eines Samsung Galaxy S8 mit einem Foto überlisten

- Hochauflösendes Infrarot-Foto des Auges
- Krümmung des Auges wird mit auf Foto aufgelegter Kontaktlinse simuliert





https://media.ccc.de/v/biometrie-s8-iris-en
http://www.zeit.de/digital/datenschutz/2017-05/samsung-galaxy-s8-biometrie-auge-gehackt/komplettansicht

Passwortregeln

- Falsch
  - Namen, Telefonnummern, Geburtsdaten, Autonummern
  - Wörter aus Wörterbüchern, Eigennamen
  - Tastaturmuster (vgl. »wertzuio«)
- Richtig
  - viele verschiedene Zeichen
  - kombinierte Groß- und Kleinschreibweise
  - Ziffern und Sonderzeichen
- Trick
  - Die Anfangsbuchstaben eines »verrückten« Satzes verwenden, der auch Zahlen und Sonderzeichen enthält
- Wichtig
  - Passwort nicht für mehrere Dienste verwenden
  - Passwort niemals unverschlüsselt in Dateien ablegen
  - [Passwort in regelmäßigen Abständen ändern]

Single-Sign-On-Verfahren reduzieren die Anzahl zu merkender Passwörter.

Welche Passwörter werden tatsächlich genutzt?

- Im Dezember 2009 wurden 32 Millionen Passwörter von einem Hacker veröffentlicht:
  - Ein blinder Log-in-Versuch mit 123456 führt in 0,9 % der Fälle zum Erfolg.
  - Rund 50 % der Passwörter können als schwach bezeichnet werden:
    - Wörter aus Wörterbüchern
    - Tastaturmuster

Rang	Passwort	Häufigkeit
1	123456	290731
2	12345	79078
3	123456789	76790
4	Password	61958
5	iloveyou	51622
6	princess	35231
7	rockyou	22588
8	1234567	21726
9	12345678	20553
10	abc123	17542
11	Nicole	17168
12	Daniel	16409

Prüfung der Passwortqualität durch Systembetreiber zwingend erforderlich

  - Quelle: The Imperva Application Defense Center (ADC): Consumer Password Worst Practices, 2010

Welche Passwörter werden tatsächlich genutzt?

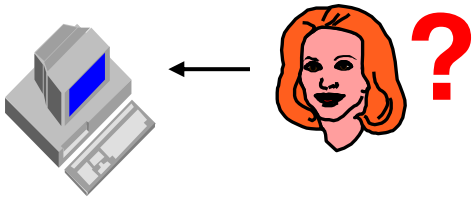
- Kompromittierte Passwörter eines Internet-Dating-Portals wurden 2006 auf einer Mailingliste veröffentlicht:
  - 1,4 % der Passwörter sind 123456
  - 2,5 % der Passwörter beginnen mit 1234
  - ca. 40 % der Passwörter nur 1x vorhanden
  - Quelle: Security: Flirt-Passwörter. ct 13 (2006) S. 64

Rang	Passwort	Häufigkeit
1	123456	1375
2	ficken	404
3	12345	367
4	hallo	362
5	123456789	260
6	schatz	253
7	12345678	215
8	daniel	215
9	askim	184
10	nadine	177
11	1234	176
12	passwort	173
…		

- Studie des Hasso-Plattner-Instituts 2016:
  - »hallo« ist meistgenutztes deutsches Passwort, auf Platz zehn steht »ficken«
  - Quelle: https://hpi.de/news/jahrgaenge/2016/hpi-wissenschaftler-ermitteln-die-zehn-meistgenutzten-deutschsprachigen-passwoerter.html

Identifikation von IT-Systemen durch Menschen

- Was es ist
  - Gehäuse
  - Siegel
  - Hologramm
  - Verschmutzung

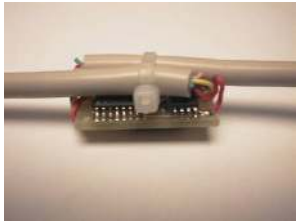


- Was es weiß
  - Passwort
  - Antworten auf Fragen
  - Rechenergebnisse für Zahlen

Warum ist das wichtig?

- Faked Login-Screen
- Phishing
- Keylogger
- Manipulation Geldautomat

- Wo es steht



Bilder: http://keyspy.de.vu/

Manipulationen an Geldautomaten

Bilder: Bundesgrenzschutzamt Flensburg, 2006

1. Aufsatz auf Kartenschlitz liest Magnetstreifen …

Manipulationen an Geldautomaten

Bilder: Bundesgrenzschutzamt Flensburg, 2006

2. Kamera erfasst PIN-Eingabe und überträgt die Daten per Funk

Antenne

Phishing

- E-Mail des Angreifers (Spam-E-Mail)

Fur alle Volksbanken-Raiffeisenbanken Kunden — Eingang

E-Mail(s) löschen | Ist Werbung | Antworten | An alle | Weiterleiten | Drucken

Von: VOLKSBANK <root@volksbank.de>
Betreff: **Fur alle Volksbanken-Raiffeisenbanken Kunden**
Datum: 8. März 2007 11:53:17 MEZ
An: federrath@sicherheit2005.de
Antwort an: VOLKSBANK <root@volksbank.de>

Sehr geehrter Nutzer der Volksbanken Raiffeisenbanken Online-Bankings,

wir freuen uns Ihnen neue Informationen uber die Sicherheit im Internet erteilen zu durfen.
Bitte lesen sie es aufmerksam!

Weltweit gilt das Online-Banking durch TAN Verfahren als eines der sichersten Legitimations-Verfahren fur Online-Bankgeschafte. Dennoch gab es in letzter Zeit immer wieder Versuche, auf betrugerische Art und Weise das Geld von Volksbanken Raiffeisenbanken Kunden ins Ausland zu uberweisen.

Leider ist uns momentan das Verfahren, dass die Betruger benutzen, nicht bekannt.

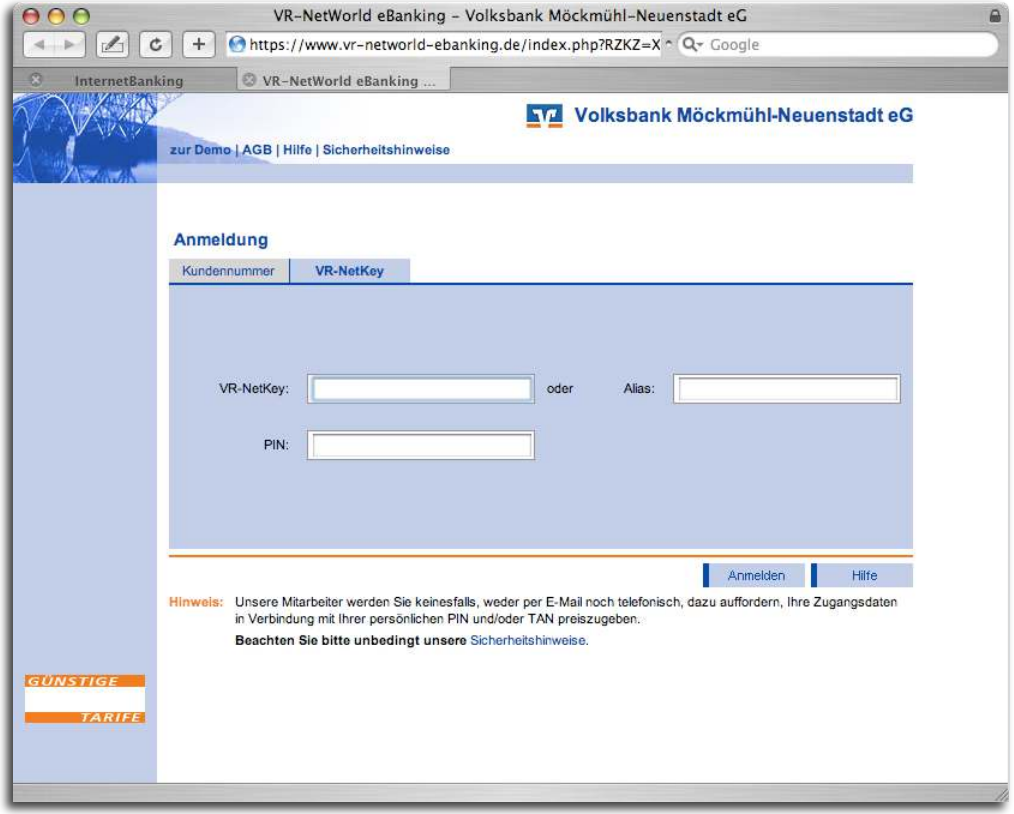
Um unsere Kunden von Betruger zu schutzen, hat unser Sicherheitsteam fur neue Schutzmassnahmen entschieden. Beachten sie bitte, dass die Einsetzung dieser Schutzmassnahmen erforderlich fur alle Volksbanken Raiffeisenbanken Kunden ist!

Um diese Massnahmen einfuhren zu konnen, mussen sie 2 TANs aus ihrer aktuellen Tan-Liste eingeben.

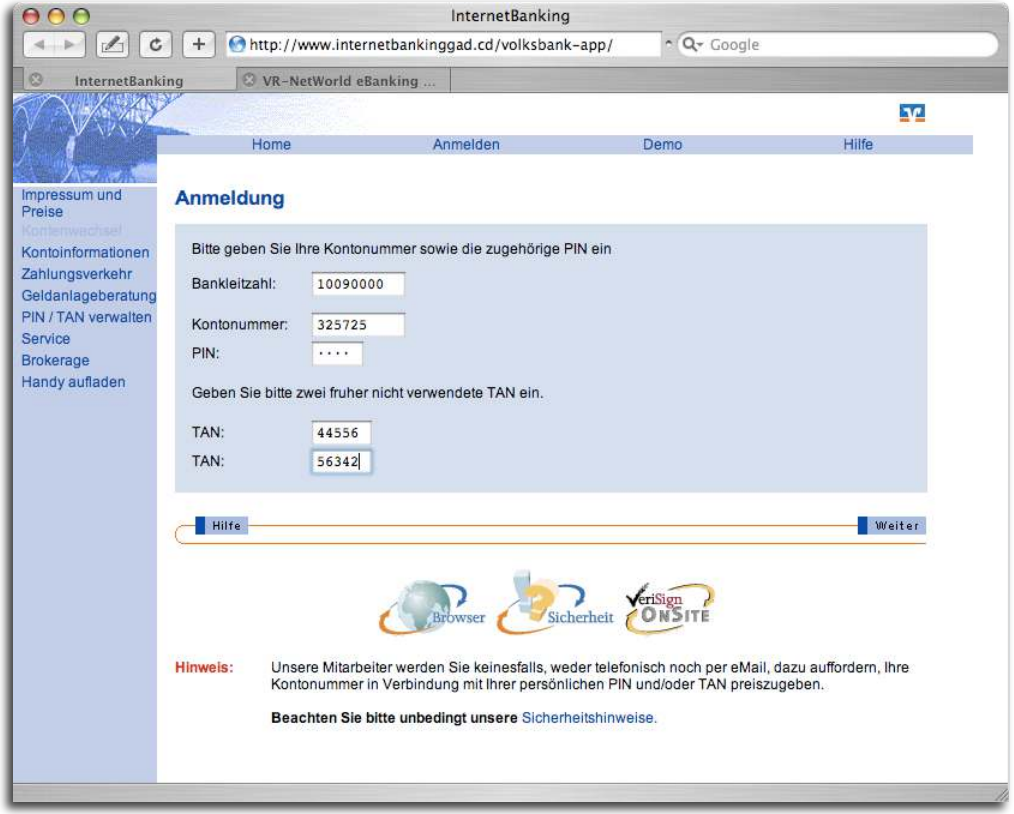
Folgen sie bitte diesen Link, um Ihr Konto bei der Volksbanken Raiffeisenbanken zu authentifizieren -
https://www.volksbank.vr-networld.de/frames/verify.php

Achtung! Wir bitten unsere Kunden um Verstandnis fur diese Uberprufung. Alle Volksbanken-Raiffeisenbankenkonten die nicht innerhalb eines Tages authentifiziert werden, werden gesperrt!

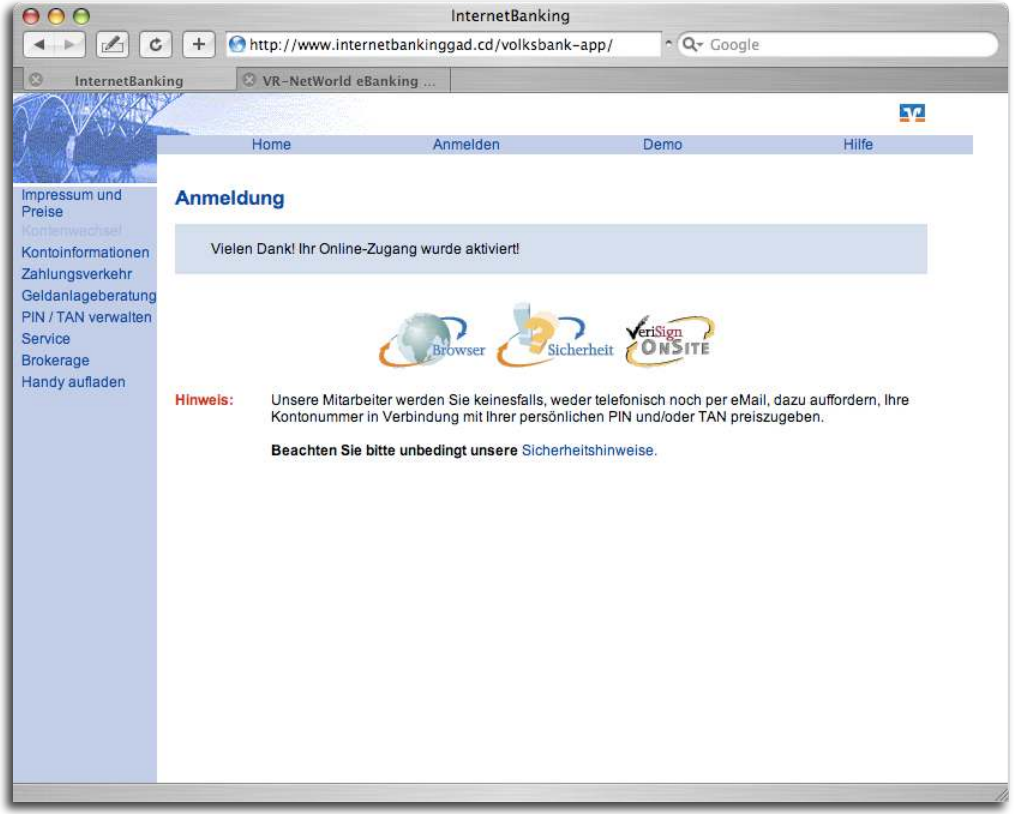
Original


VR-NetWorld eBanking - Volksbank Möckmühl-Neuenstadt eG
https://www.vr-networld-ebanking.de/index.php?RZKZ=X
Google
InternetBanking
VR-NetWorld eBanking ...
Volksbank Möckmühl-Neuenstadt eG
zur Demo | AGB | Hilfe | Sicherheitshinweise
Anmeldung
Kundennummer
VR-NetKey
VR-NetKey:
oder
Alias:
PIN:
Anmelden
Hilfe
Hinweis: Unsere Mitarbeiter werden Sie keinesfalls, weder per E-Mail noch telefonisch, dazu auffordern, Ihre Zugangsdaten in Verbindung mit Ihrer persönlichen PIN und/oder TAN preiszugeben.
Beachten Sie bitte unbedingt unsere Sicherheitshinweise.
GÜNSTIGE
TARIFE

Fälschung

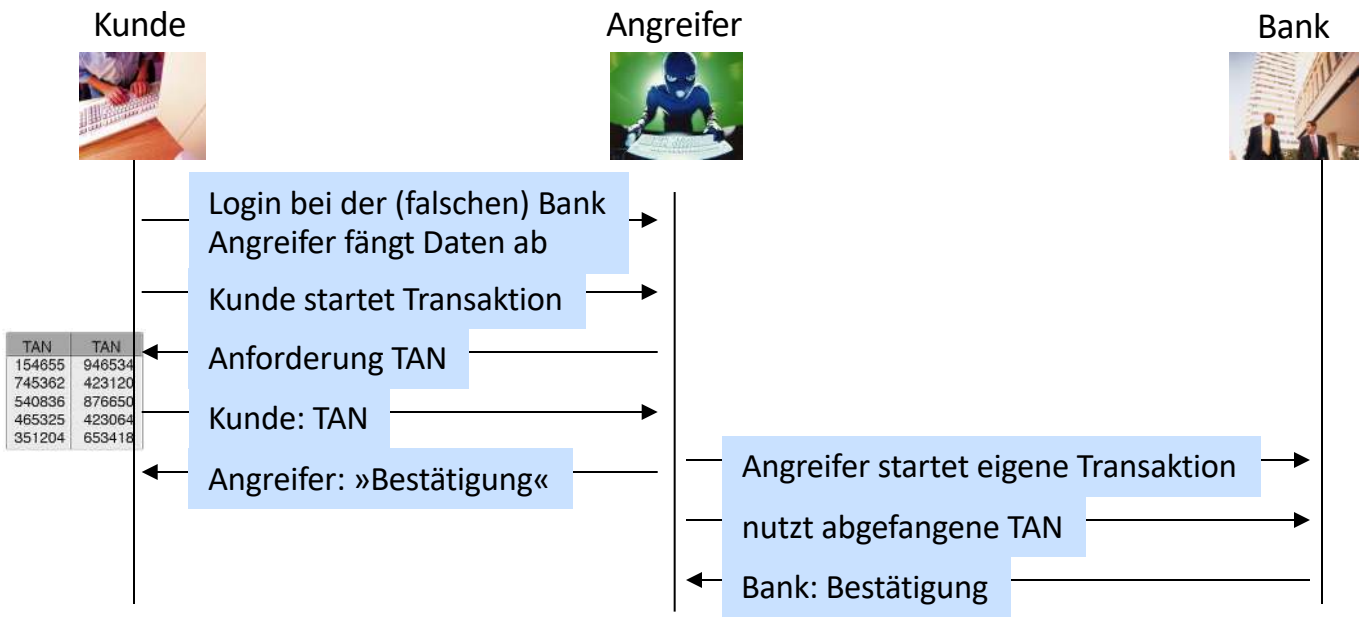

InternetBanking
http://www.internetbankinggad.cd/volksbank-app/
Google
InternetBanking
VR-NetWorld eBanking ...
Home
Anmelden
Demo
Hilfe
Impressum und Preise
Kontenwechsel
Kontoinformationen
Zahlungsverkehr
Geldanlageberatung
PIN / TAN verwalten
Service
Brokerage
Handy aufladen
Anmeldung
Bitte geben Sie Ihre Kontonummer sowie die zugehörige PIN ein
Bankleitzahl: 10090000
Kontonummer: 325725
PIN: ....
Geben Sie bitte zwei fruher nicht verwendete TAN ein.
TAN: 44556
TAN: 56342
Hilfe
Weiter
Browser
Sicherheit
VeriSign OnSite
Hinweis: Unsere Mitarbeiter werden Sie keinesfalls, weder telefonisch noch per eMail, dazu auffordern, Ihre Kontonummer in Verbindung mit Ihrer persönlichen PIN und/oder TAN preiszugeben.
Beachten Sie bitte unbedingt unsere Sicherheitshinweise.

Fälschung


InternetBanking
http://www.internetbankinggad.cd/volksbank-app/
Google
InternetBanking
VR-NetWorld eBanking ...
Home
Anmelden
Demo
Hilfe
Impressum und Preise
Kontenwechsel
Kontoinformationen
Zahlungsverkehr
Geldanlageberatung
PIN / TAN verwalten
Service
Brokerage
Handy aufladen
Anmeldung
Vielen Dank! Ihr Online-Zugang wurde aktiviert!
Browser
Sicherheit
VeriSign ONSITE
Hinweis:
Unsere Mitarbeiter werden Sie keinesfalls, weder telefonisch noch per eMail, dazu auffordern, Ihre Kontonummer in Verbindung mit Ihrer persönlichen PIN und/oder TAN preiszugeben.
Beachten Sie bitte unbedingt unsere Sicherheitshinweise.

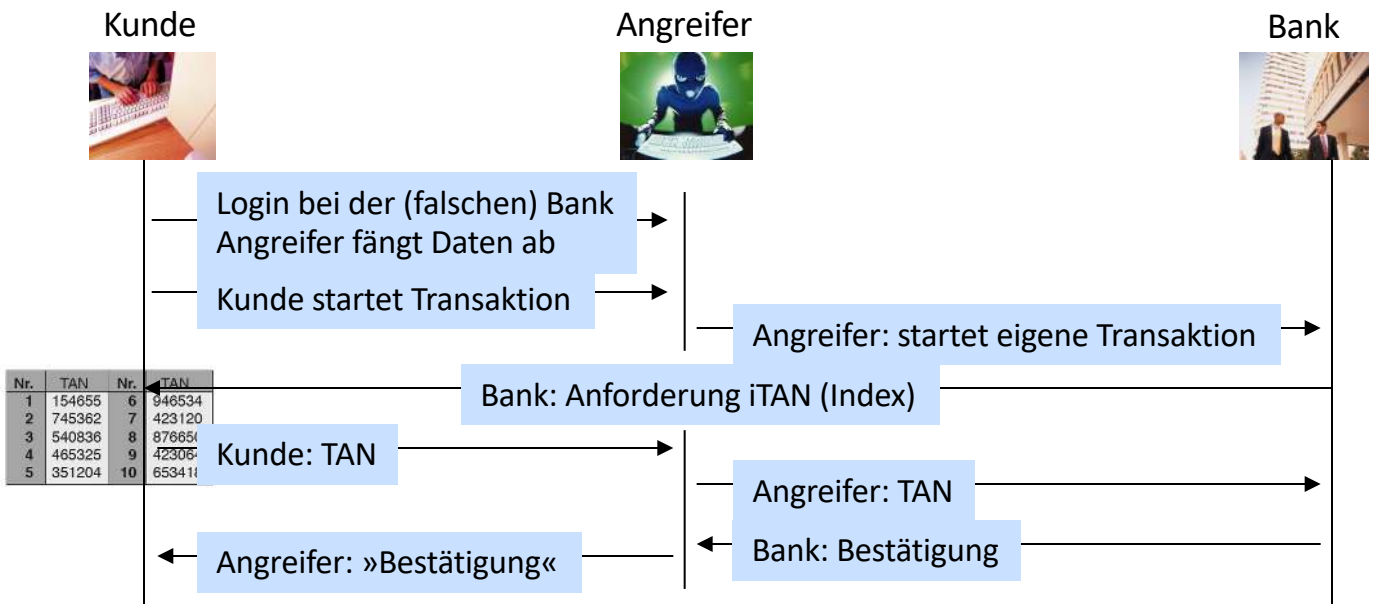
Man-in-the-middle-attack auf TAN-Verfahren (Skizze)

- Voraussetzung: Angreifer
  - betreibt täuschend echte Webseite der Bank
  - bewegt den Kunden zum Besuch dieser Seite



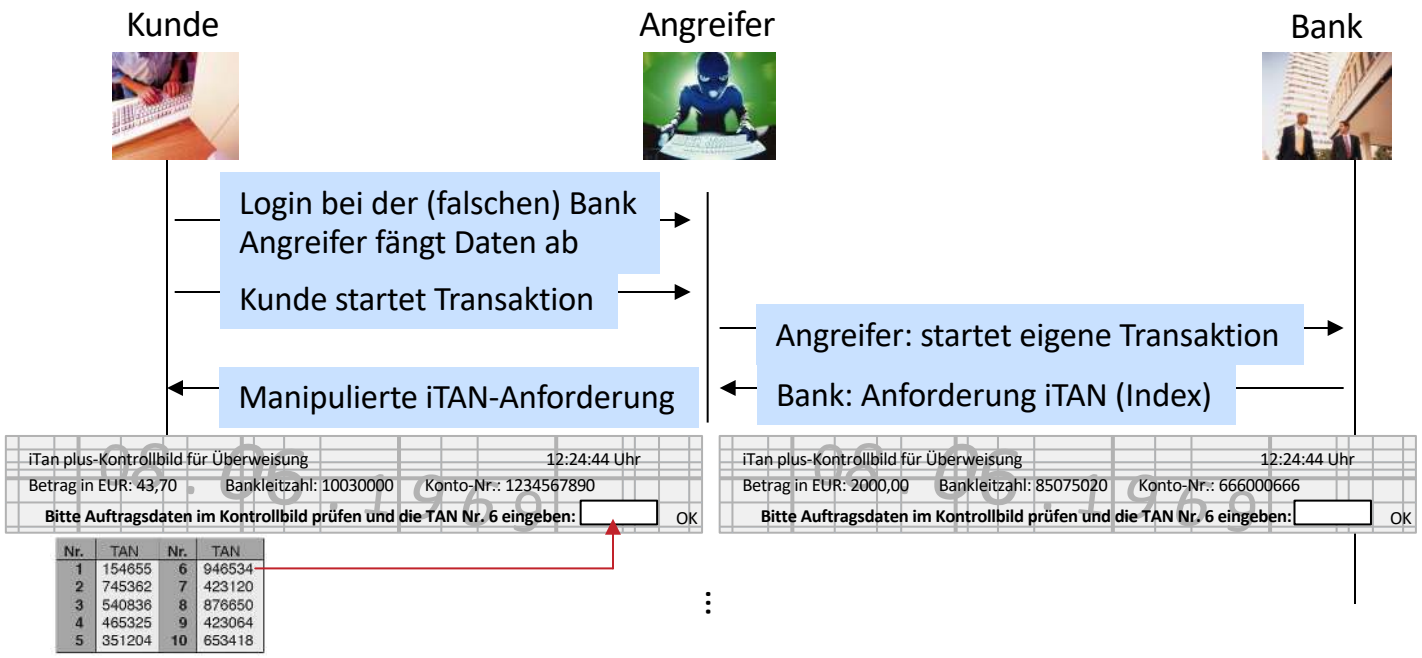
Man-in-the-middle-attack auf iTAN-Verfahren (Skizze)

- Verbesserungen gegenüber normalem TAN-Verfahren:
  - Angreifer benötigt »Online-Hilfe« durch Kunden, d.h. er kann nur Transaktionen erfolgreich durchführen, wenn Kunde dies selbst gerade tun will



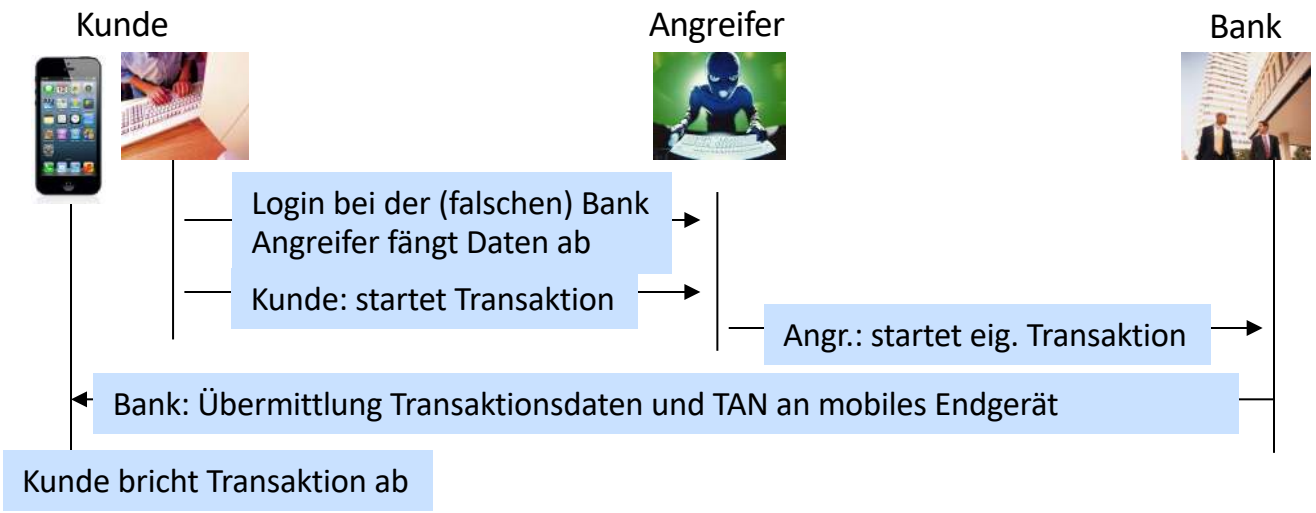
Man-in-the-middle-attack auf iTAN plus

- Verbesserungen durch iTAN plus gegenüber iTAN-Verfahren:
  - Angreifer muss korrektes Geburtsdatum des Kunden automatisiert aus dem Hintergrund-CAPTCHA extrahieren, um passende Grafik zur TAN-Anforderung zu konstruieren



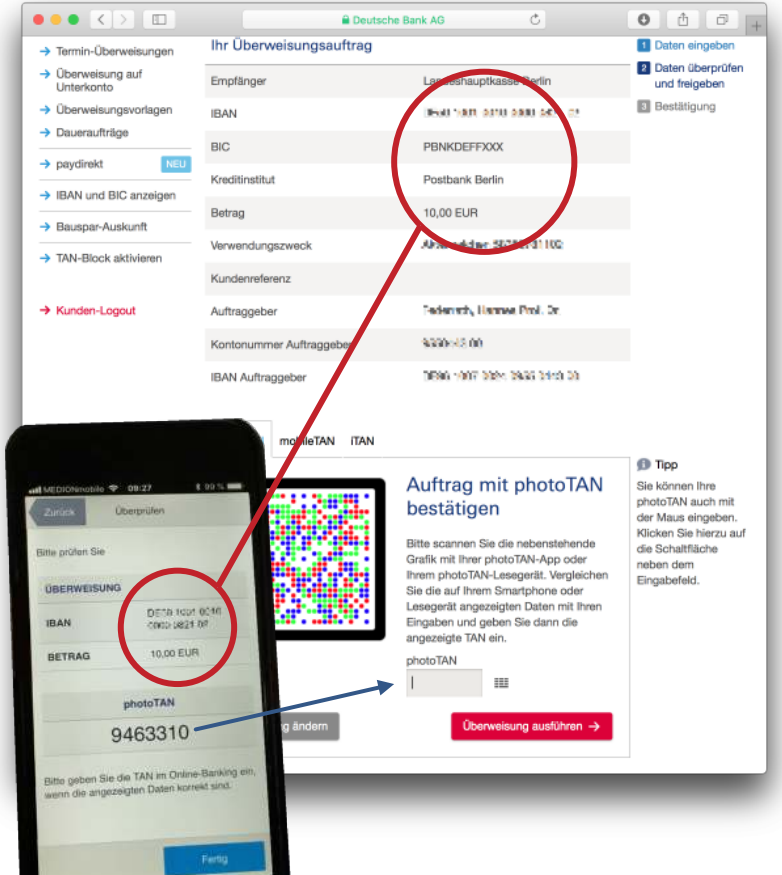
mTAN-Verfahren (Skizze)

- Voraussetzung für Sicherheit:
  - Mobiles Gerät wird nicht gleichzeitig für die Transaktion verwendet (Medienbruch beim Kunden)



mTAN-Verfahren

- Voraussetzung für Sicherheit:
  - Mobiles Gerät wird nicht gleichzeitig für die Transaktion verwendet (Medienbruch beim Kunden)
- Transaktionsdaten werden auf zweitem Gerät angezeigt und vom Kunden überprüft
  - anschließend Eingabe der TAN



mTAN-Verfahren (verbleibende Angriffe)

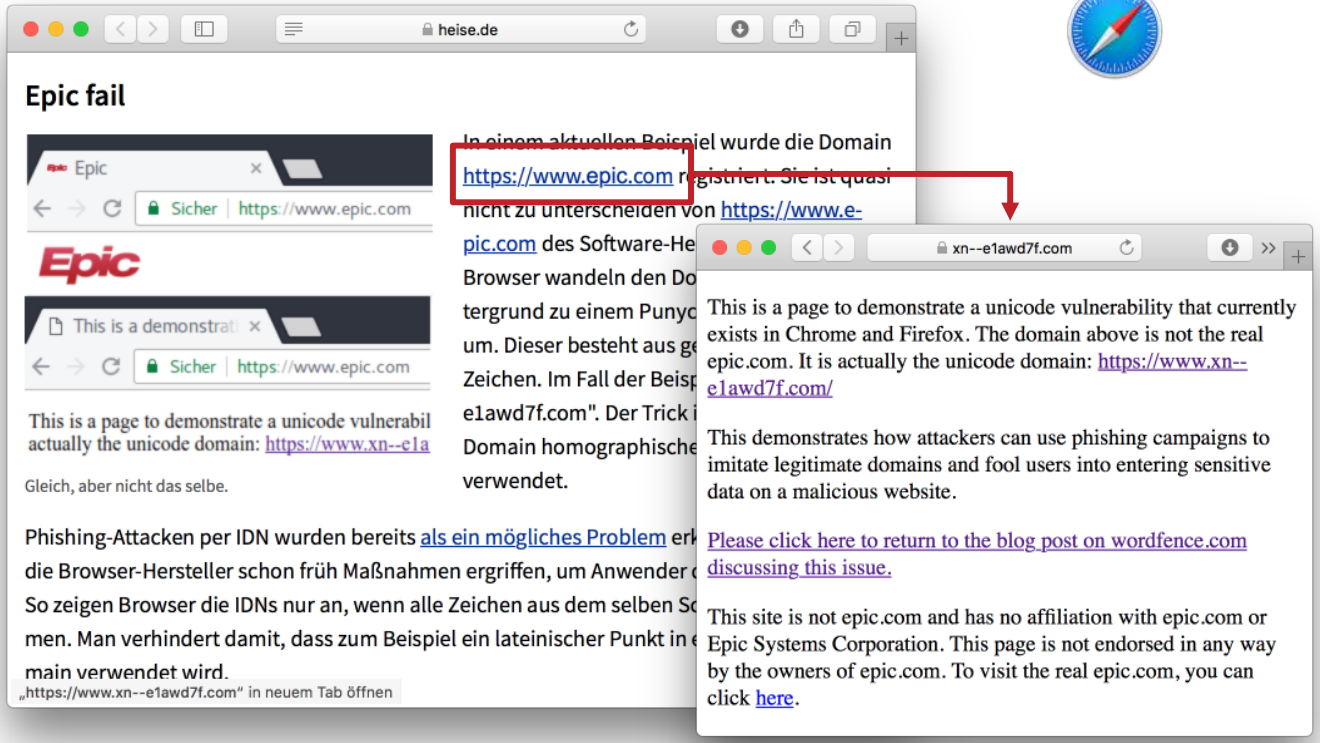
- Medienbruch wird vom externen Angreifer ausgehebelt
  - Angreifer weist auf Phishing-Webseite auf eine vermeintliche App der Bank hin, die man auf seinem Smartphone installieren soll (Auf diese Weise sollen Angreifer ca. 40 Mio. Euro erbeutet haben.)
  - Angreifer fordert zweite SIM-Karte vom Proivder an, ändert bei der Gelegenheit die Adresse des Kunden, damit SIM-Karte bei ihm ankommt

- Mobilfunknetzinternes Signalling System #7 (SS7) wird von Hacker, Nachrichtendienst oder bösartigem Netzbetreiber selbst missbraucht
  - Angreifer kann sicherheitsrelevante Daten (z.B. Sitzungsschlüssel) von anderen Netzbetreibern (Heimatnetzbetreiber oder Netzbetreiber, bei dem das Opfer eingebucht ist) abfragen, um SMS mitzulesen
  - Angreifer nutzt SS7 zum kurzzeitigen Einrichten einer Rufumleitung auf andere Nummer, um SMS abzufangen (etwa in der Nacht)

Unicode-Domains

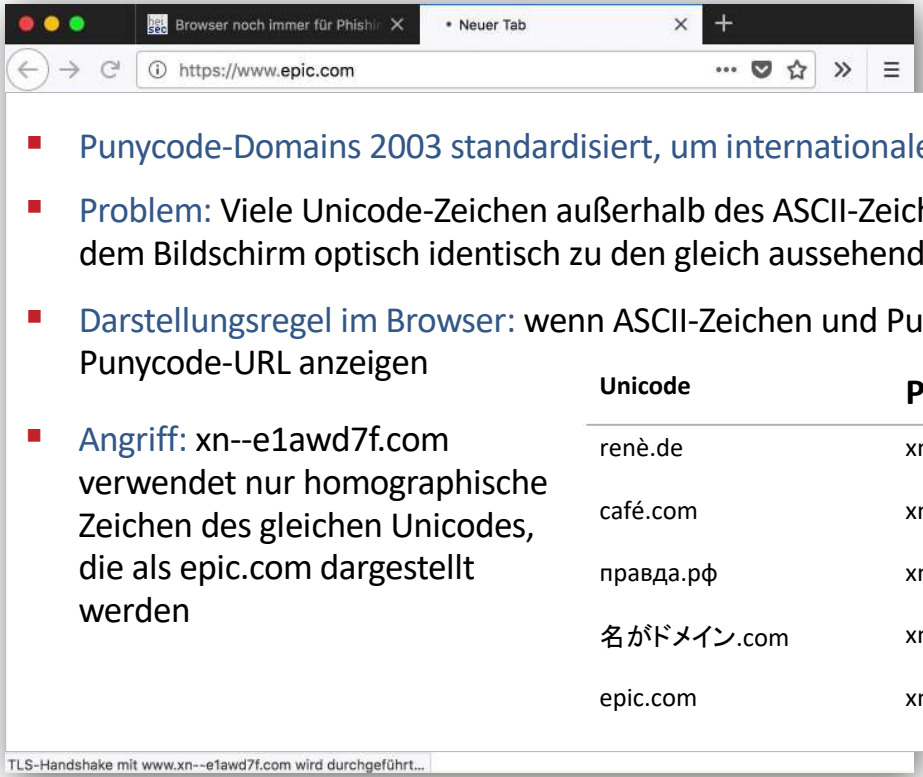
https://heise.de/-3686474

- Angreifer registriert Domain mit identisch aussehenden Unicode-Zeichen



Unicode-Domains erlauben u.U. homographische Angriffe

- Angreifer registriert Domain mit identisch aussehenden Unicode-Zeichen



- Punycode-Domains 2003 standardisiert, um internationale Domains zu ermöglichen
- Problem: Viele Unicode-Zeichen außerhalb des ASCII-Zeichenraumes erscheinen auf dem Bildschirm optisch identisch zu den gleich aussehenden ASCII-Zeichen.
- Darstellungsregel im Browser: wenn ASCII-Zeichen und Punycodes gemischt, dann Punycode-URL anzeigen
- Angriff: xn--e1awd7f.com verwendet nur homographische Zeichen des gleichen Unicodes, die als epic.com dargestellt werden

Unicode	Punycode
renè.de	xn--ren-8la.de
café.com	xn--caf-dma.com
правда.рф	xn--80aafi6cg.xn--p1ai
名がドメイン.com	xn--v8jxj3d1dzdz08w.com
epic.com	xn--e1awd7f.com

Screenshots vom Juni 2018

Tippfehler und Phishing

- Ähnlich lautende Domains
  - www.whitehouse.gov
  - meine.deutsche-bank.de
  - paypal.com
- Typosquatting-Domains
  - wikipedia.org
  - google.de

www.whitehouse.com
www.whitehouse.org

meine-deutsche-bank.de

crt.sh Identity Search

Criteria	Identity LIKE 'paypal%'; Issuer CA ID = 16418

C=US, O=Let's Encrypt, CN=Let's Encrypt Authority X3

1 to 100 Next

crt.sh ID	Not Before	Not After	Subject Name
108371769	2017-03-24	2017-06-22	CN=paypal.com.login-myaccount.policy.country
108370216	2017-03-24	2017-06-22	CN=paypal.com.updates-information-accounts.ga
108368412	2017-03-24	2017-06-22	CN=paypal.com.updates-information-accounts.gq
108362962	2017-03-24	2017-06-22	CN=paypal.com.account.update.amquipac.org
108361574	2017-03-24	2017-06-22	CN=paypal.info.com.secure-info-update.com
108361200	2017-03-24	2017-06-22	CN=paypal.com.login.monthly-limited-account.cf
108361075	2017-03-24	2017-06-22	CN=paypal.com.verifedlockedaccount.tk
108355147	2017-03-24	2017-06-22	CN=aljeyad.com
108354806	2017-03-24	2017-06-22	CN=paypal.com.login.summary-limited-account.gq
108354169	2017-03-24	2017-06-22	CN=paypal.com-check-activity.info
108338084	2017-03-24	2017-06-22	CN=paypal.com.resolution-ticket.tk

www.googel.de

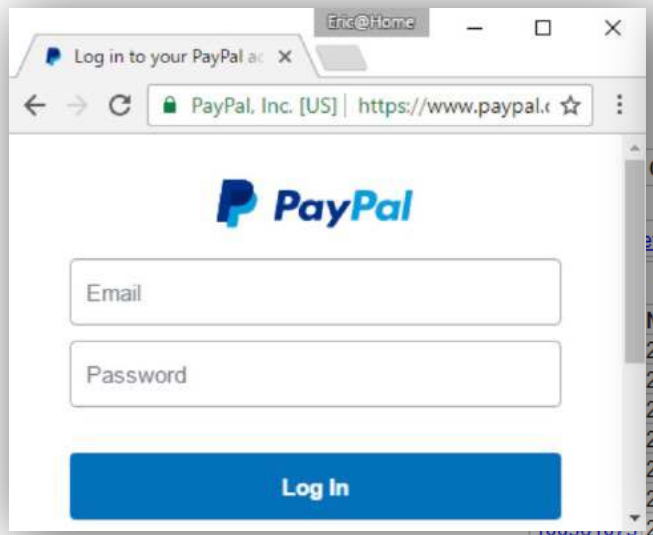
wikiwpdia.org

https://www.bleepingcomputer.com/news/security/14-766-lets-encrypt-ssl-certificates-issued-to-paypal-phishing-sites/

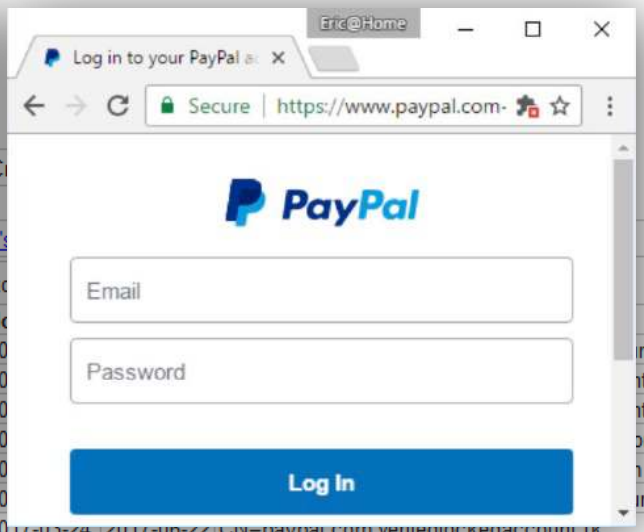
Zertifizierung von Fake-Domains

- letsencrypt.org ermöglicht unkomplizierte Zertifizierung von Domains

Original



Fälschung

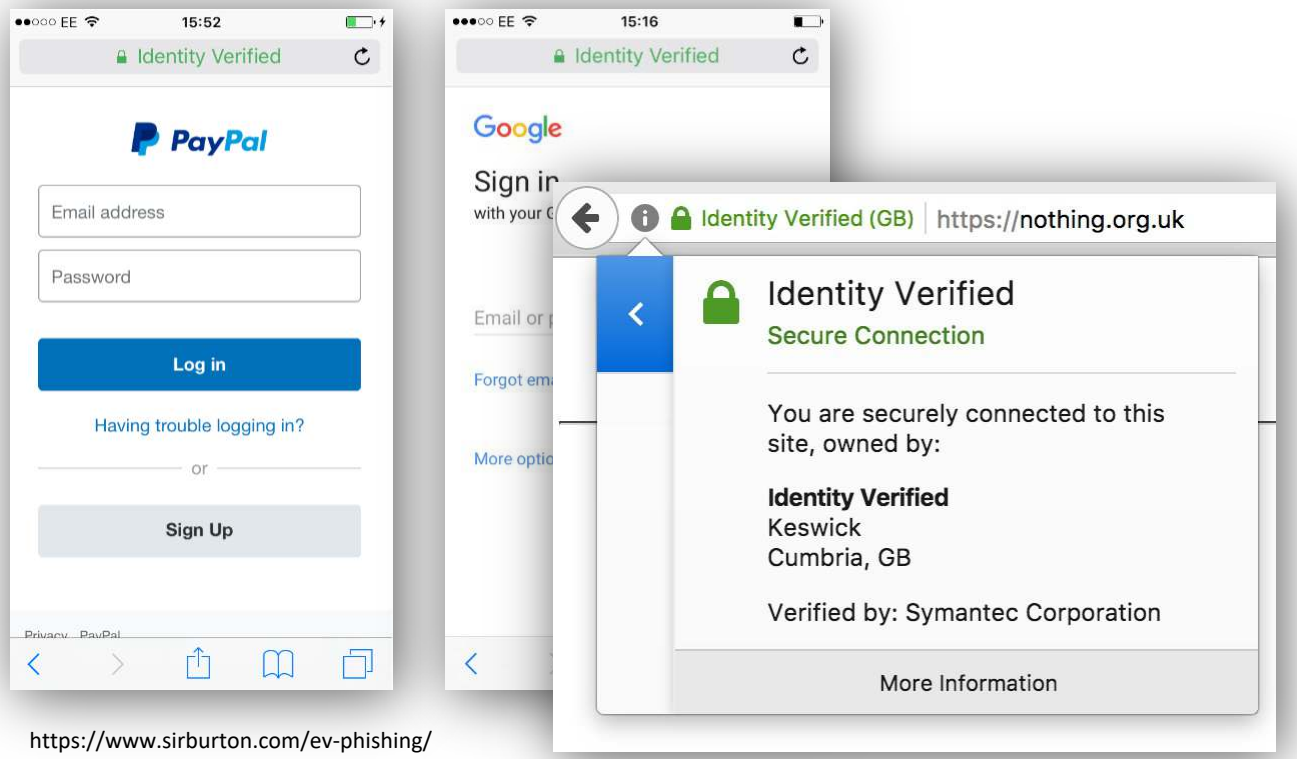


108355147	2017-03-24	2017-06-22	CN=aljeyad.com
108354806	2017-03-24	2017-06-22	CN=paypal.com.login.summary-limited-account.gq
108354169	2017-03-24	2017-06-22	CN=paypal.com-check-activity.info
108338084	2017-03-24	2017-06-22	CN=paypal.com.resolution-ticket.tk

https://www.bleepingcomputer.com/news/security/14-766-lets-encrypt-ssl-certificates-issued-to-paypal-phishing-sites/

Extended Validation Certificates

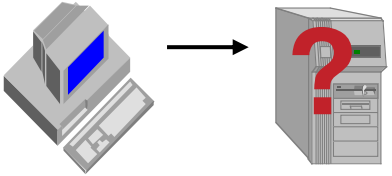
- kein Schutz vor Fake-Domains, aber Domain-Besitzer ist ermittelbar



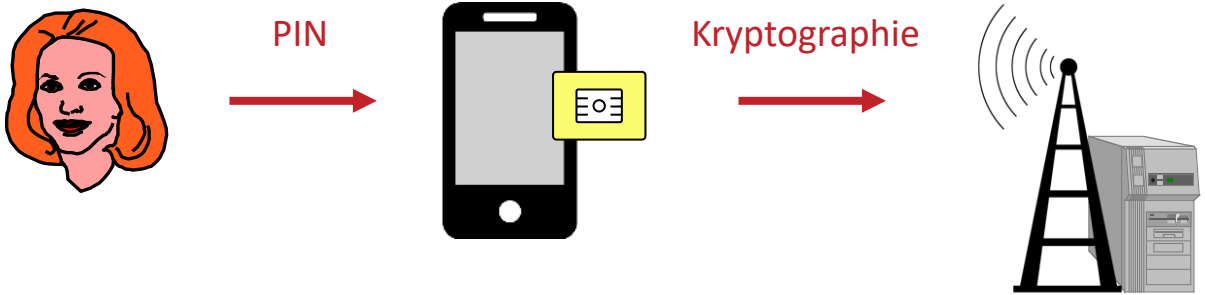
https://www.sirburton.com/ev-phishing/

Identifikation von IT-Systemen durch IT-Systeme

- Was es weiß
  - Passwort
  - Antworten auf Fragen
  - Rechenergebnisse für Zahlen
  - Kryptographie
- Leitung woher

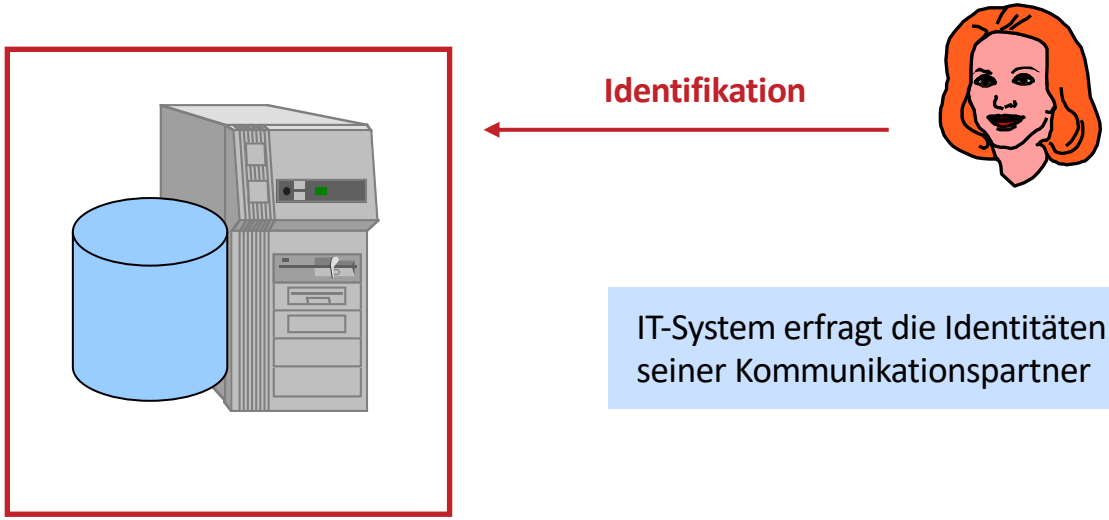


Zusammenspiel (Beispiel)



Zugangskontrolle (admission control)

- Zweck
  - Nur mit berechtigten Partnern weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

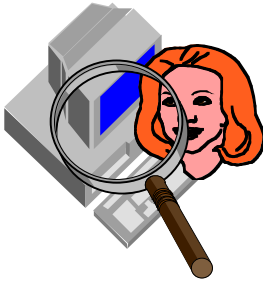


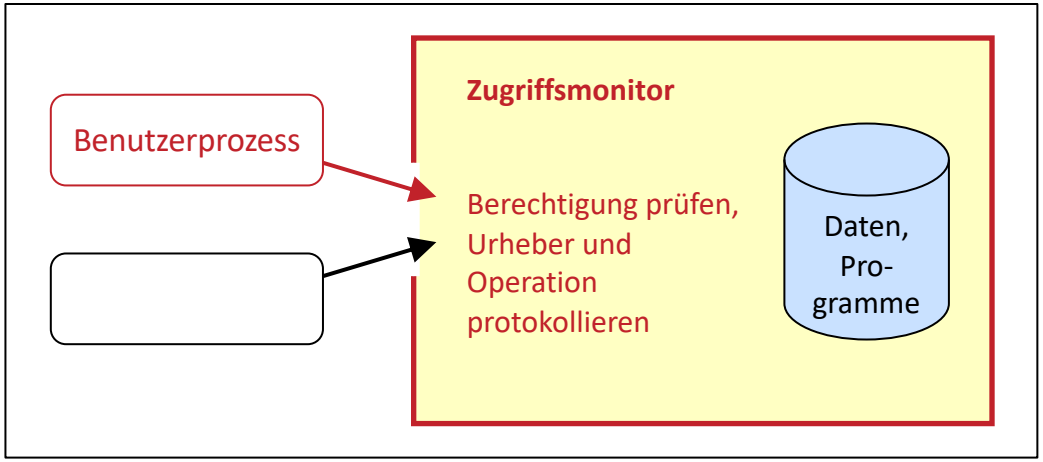
IT-System erfragt die Identitäten seiner Kommunikationspartner

Zugriffskontrolle (access control)

Subjekt kann Operation auf Objekt nur ausführen, wenn es das *Recht* dazu hat.

Zugriffskontrolle setzt eine vorangegangene Zugangskontrolle voraus.



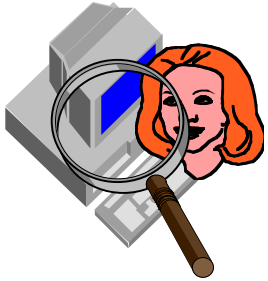


Zugriffskontrolle (access control)

- Zugriffskontrollpolitiken
  - Es ist alles erlaubt, was nicht explizit verboten ist.
  - Es ist alles verboten, was nicht explizit erlaubt ist.

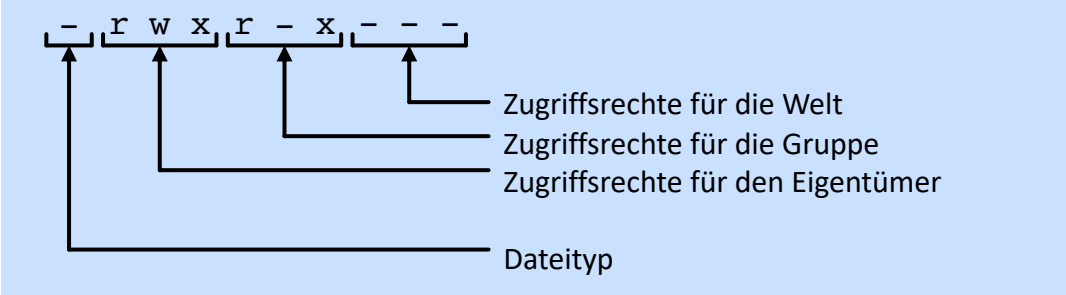
- Autorisierung
  - Konkrete Festlegung von Zugriffsrechten

- Übliche Prinzipien bei der Autorisierung
  - Principle of least privilege
  - Need to know principle
  - Separation of duty



Zugriffskontrollmodelle

- Discretionary Access Control (DAC)
  - Nutzerbezogene Zugriffskontrolle
  - Eigentümer-Prinzip:
    - Jedes Objekt hat einen Eigentümer, der Berechtigungen an Subjekte vergeben kann
  - Beispiel: Linux-Dateirechte
    - Mögliche Rechte sind Lese-, Schreib- und Ausführungsrecht (rwx)
    - Unterscheidung nach Eigentümer (user), Gruppe (group) und alle anderen (world)



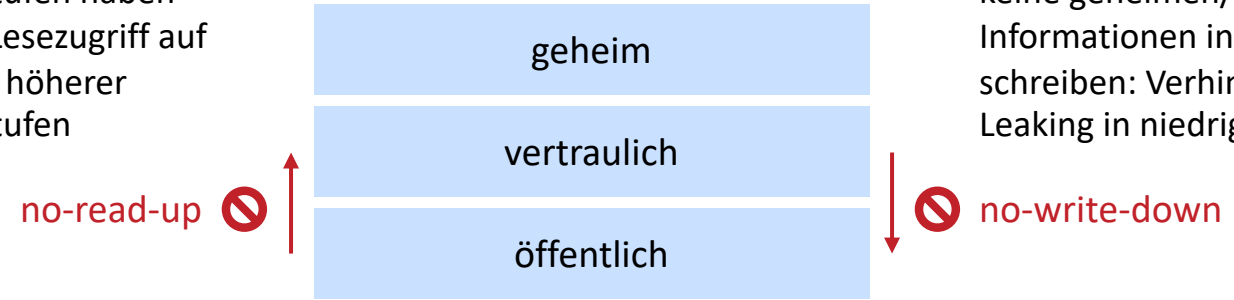
Zugriffskontrollmodelle

- Mandatory Access Control (MAC)
  - Systembestimmte Zugriffskontrolle
  - Hinzunahme weiterer Eigenschaften (z.B. Schutzstufen)

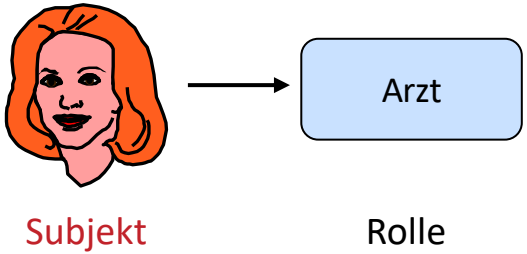
  - Beispiel: Bell LaPadula Modell
    - Informationsflusskontrolle (Schutzziel Vertraulichkeit)
    - Subjekte und Objekte werden in Schutzstufen eingeteilt, die eine Ordnung bilden

Subjekte niedrigerer Schutzstufen haben keinen Lesezugriff auf Objekte höherer Schutzstufen

Subjekte höherer Schutzstufen dürfen keine geheimen/vertraulichen Informationen in niedrigere Schutzstufen schreiben: Verhindern von Information Leaking in niedrigere Schutzstufen



- Role Based Access Control (RBAC)
  - Rolle beschreibt Berechtigungen
  - Nutzer werden einer oder mehreren Rollen zugeordnet
  - Änderungen an Berechtigungen der Rolle wirken sich auf alle Nutzer aus
  - Vertreter: SELinux
  - Verschiedene Ausprägungen von RBAC, um Widersprüche zu erkennen
    - z.B. (r und –) an ein Subjekt wäre widersprüchlich

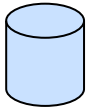


Zugriffskontrollstrukturen

- Zugriffskontrollmatrix
  - in realen Systemen nicht vorhanden (Modellstruktur)
  - Großteil der Tabellenzellen ist leer (kein Recht)
  - kombinierbar mit RBAC {user4 -> rolle1, …}

Subjekte

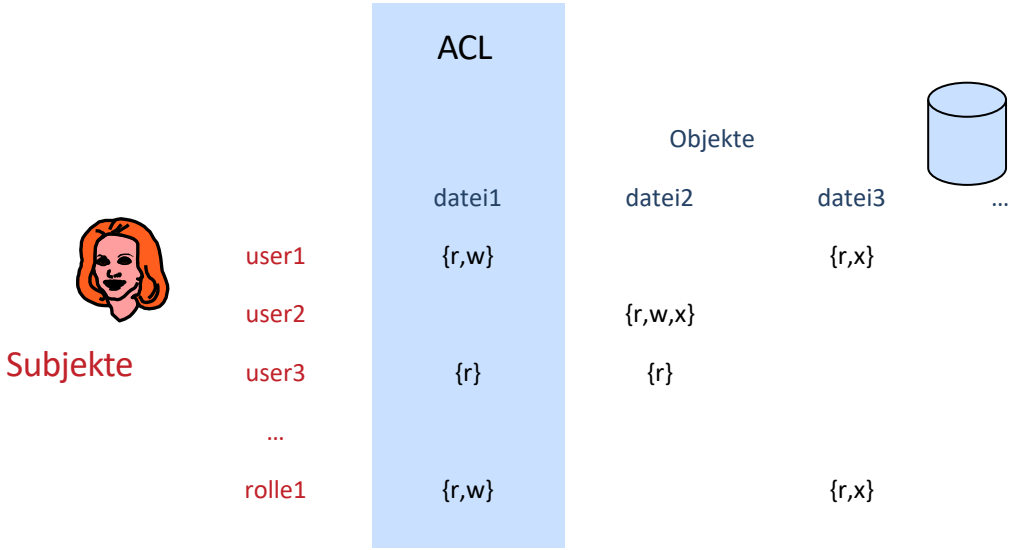
Objekte



	datei1	datei2	datei3	…
user1	{r,w}		{r,x}	
user2		{r,w,x}		
user3	{r}	{r}		
…				
rolle1	{r,w}		{r,x}	

Zugriffskontrollstrukturen

- Access Control List (ACL) – Zugriffskontrollliste
  - Spaltenweise (kompakte) Sicht auf Zugriffskontrollmatrix
    - datei1: user1:rw user3:r … rolle1:rw
  - in realen Systemen häufig verwendet, da Zugriffsentscheidung sehr schnell getroffen werden kann



Zugriffskontrollstrukturen

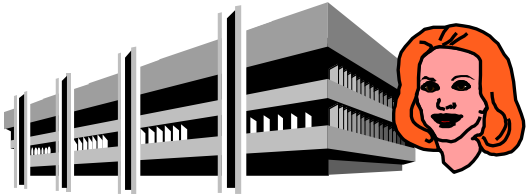
- Capability List (CL)
  - Zeilenweise Sicht auf Zugriffskontrollmatrix
  - Welche Berechtigungen hat Nutzer x?
    - user1: datei1:rw datei3:rx …
  - selten implementiert, meist als Snapshot zur Dokumentation von Berechtigungen (Traversieren Dateisystem)

Subjekte

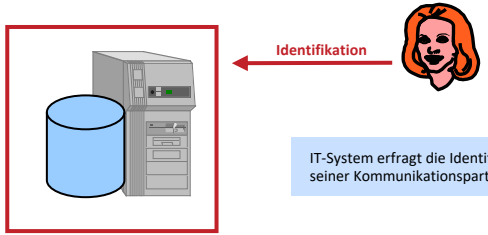
	Objekte			
	datei1	datei2	datei3	…
user1	{r,w}		{r,x}	
user2		{r,w,x}		
user3	{r}	{r}		
…				
rolle1	{r,w}		{r,x}	

Abgrenzung: Zutritts-, Zugangs-, Zugriffskontrolle

Zutrittskontrolle

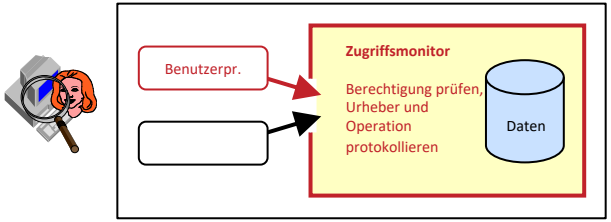


Zugangskontrolle



IT-System erfragt die Identitäten seiner Kommunikationspartner

Zugriffskontrolle



Firewalls

Paketfilter, Circuit Level Gateways, Proxies

Firewall-Architekturen

Firewalls

- Zweck
  - Abschottung eines Intranets bzw. eines sicherheitsempfindlichen Teilnetzes vor unberechtigten Zugriffen von außen
  - alle Datenpakete zum und vom Intranet müssen Firewall passieren
  - regelbasierte Filtersysteme

Typen von Firewalls

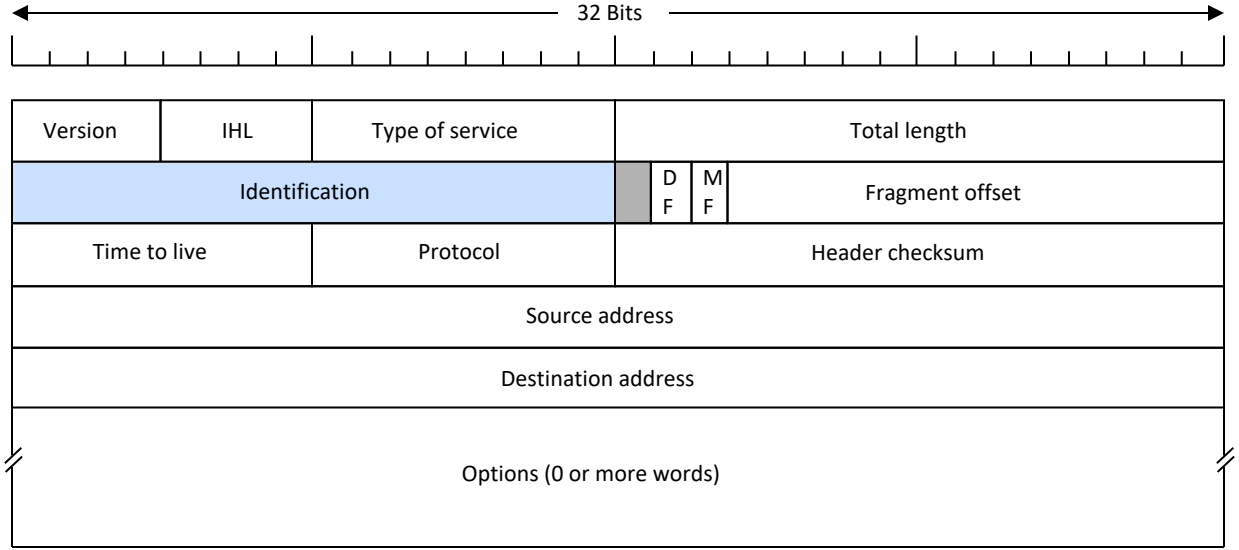
- Paketfilter
  - überprüfen anhand der IP-Adresse des Absenders und Empfängers und Portnummer (TCP oder UDP), ob Datenpakete die Firewall passieren dürfen
  - teilweise auch Analyse des Inhalts der Datenpakete
- Circuit Level Gateways
  - überprüfen TCP- oder UDP-Verbindungen
  - Verbindung: viele Datenpakete
  - Firewall ersetzt bei gehenden Verbindungen die ursprüngliche Absenderadresse durch die eigene IP-Adresse
  - verbirgt somit interne Netzstruktur
- Application Level Gateways (auch: Proxies)
  - implementieren die Schnittstelle des Clients als auch des Servers eines Dienstes

Merkmale von Firewalls

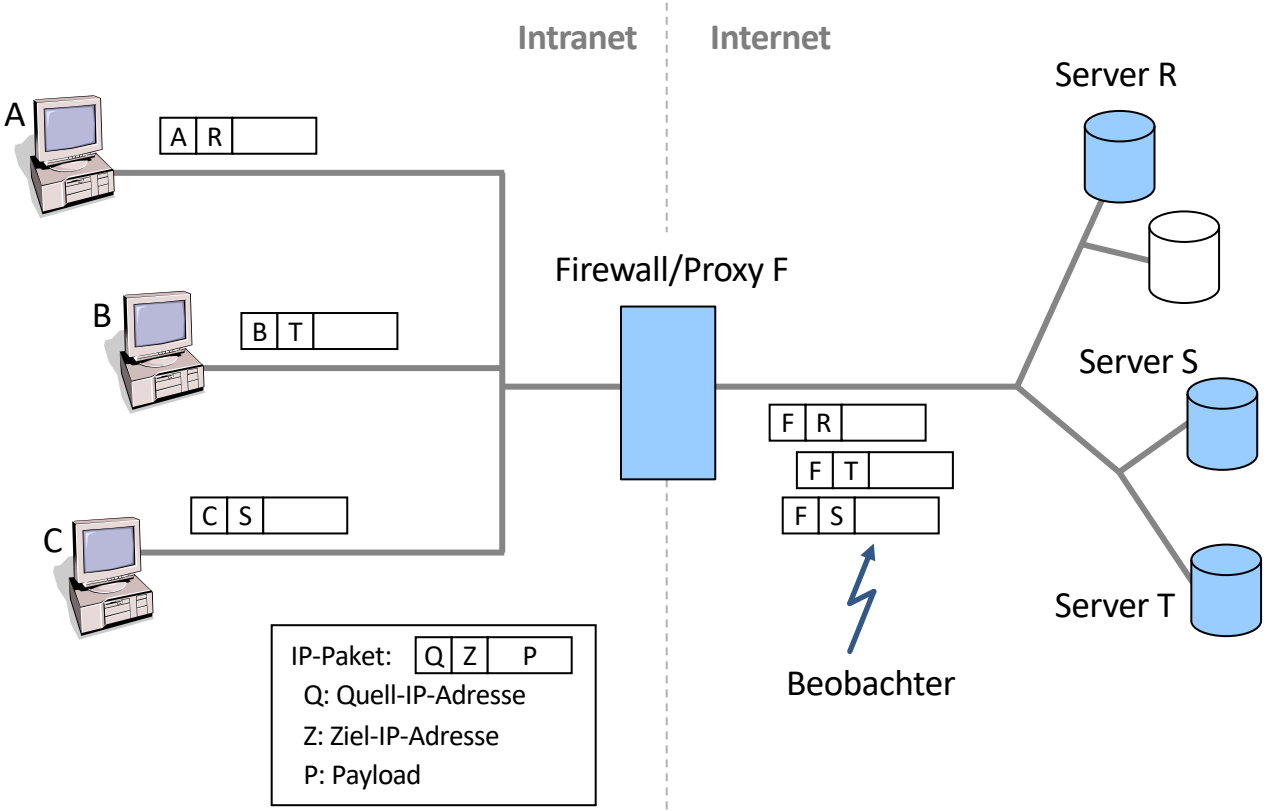
- Paketfilter und Circuit Level Gateways:
  - können transparent für Endbenutzer eingesetzt werden
  - Anwender im Intranet muss von Existenz nichts mitbekommen
- Proxies:
  - Anwendungsunterstützung notwendig: Anwendungen müssen proxy-tauglich sein
  - existieren für fast alle relevanten Anwendungen (Web, Filetransfer, News)
  - Problem:
    - für jede Anwendung (bzw. deren Protokoll) muss eigener Proxy vorhanden sein
  - Lösung:
    - SOCKS-Proxy
    - universelle Proxy-Schnittstelle (anwendungsunabhängig)

Circuit Level Gateways und NAT

- Kann man feststellen, wieviele Hosts hinter einem NAT-GW sind?
  - S. Bellovin: A Technique for counting NATed hosts. Proc. 2nd ACM SIGCOMM Workshop on Internetmeasurment 2002. http://www.cs.columbia.edu/~smb/papers/fnat.pdf
  - Viele Betriebssysteme benutzen das Header-Feld ID als Zähler: »The technique is based on the observation that on many operating systems, the IP header's ID field is a simple counter.«

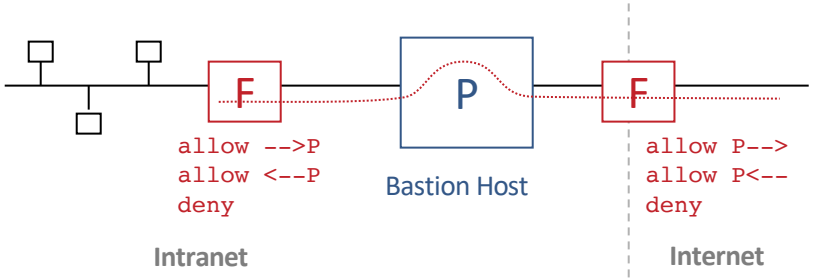


Proxies – Schutz vor Beobachtung im Internet



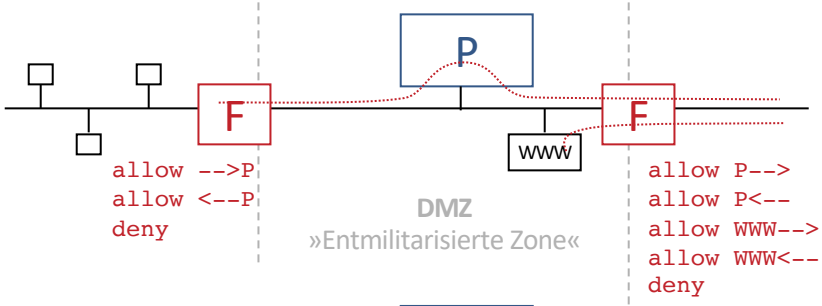
Firewall-Architekturen

Dual Homed Gateway



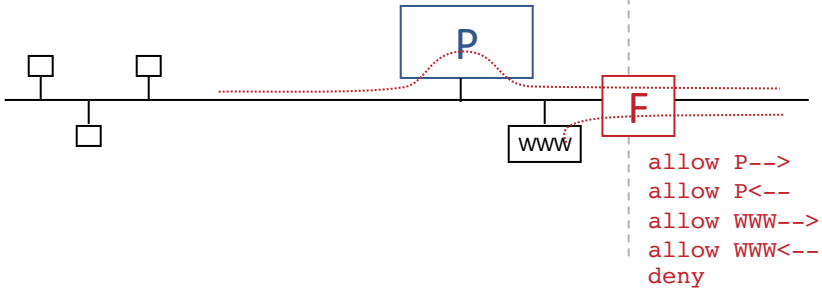
Proxy benötigt 2 Network-Interfaces; unflexibel, dafür sehr sicher, da direkte Kommunikation zwischen Inter- und Intranet unmöglich, Paketfilter sind theoretisch entbehrlich

Screened Subnet



weit verbreitet, erreicht annähernd Sicherheit des Dual Homed Gateways, Aufstellen von Internet-Servern innerhalb der DMZ; flexibel, da direkte Kommunikation zwischen Inter- und Intranet bei Bedarf möglich

Screened Host



wie Screened Subnet, aber kein Schutz vor ausgehenden Paketen mit gespoofter Quell-IP-Adresse

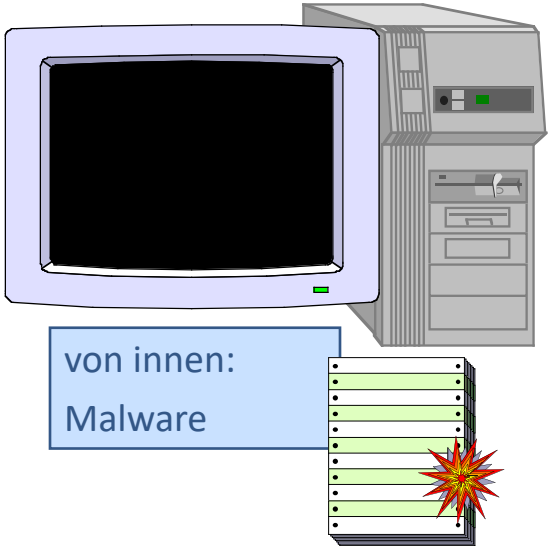
Grenzen von Firewalls

- Firewalls sind eine typische Best-Practice-Technik
  - Kompromiss zwischen Schutz und Kosten
  - besser aber teurer: jeden einzelnen Rechner im Intranet mit entsprechenden Filterfunktionen ausstatten
  - kein Schutz vor Angriffen von innen
  - defensive Konfiguration hemmt Produktivität → Aufweichen der Restriktionen
  - selbst bei defensivster Konfiguration ist Überbrücken möglich, Beispiel HTTP-Tunneling
  - kein perfekter Schutz vor Viren (verschlüsselte Kommunikation) und überhaupt kein Schutz vor trojanischen Pferden

Schadsoftware

Viren, Würmer, Trojanische Pferde

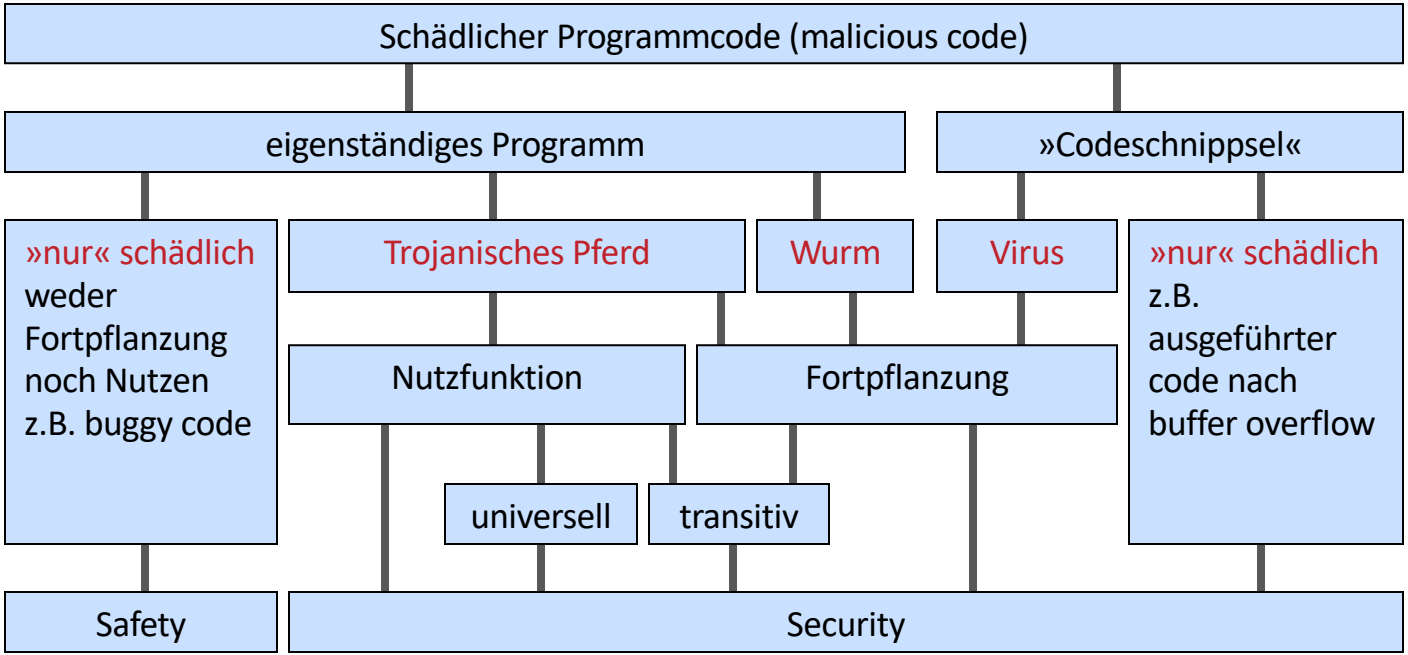
Angriffspunkte



Angreifer kann alle drei Schutzziele verletzen:

- Vertraulichkeit
- Integrität
- Verfügbarkeit

Schädlicher Programmcode (malicious code)



»Trojanisches Pferd« wird auch als Oberbegriff für Schadsoftware verwendet, die ohne Wissen des Opfers auf dessen System läuft.

Viren – Würmer – Trojanische Pferde

- Virus
  - Programmcode, der eine ungewollte (Schadens)-Funktion ausführt (Sicht des Opfers)
  - kein eigenständiges Programm, benötigt einen »Wirt«
  - enthält Mechanismus zur Fortpflanzung auf andere (lokale) Programme
  - Schutz: keine unbekannten Programme ausführen; keine unnötigen Schreibrechte vergeben (Prinzip der geringstmöglichen Privilegierung)
- Wurm
  - eigenständiges Programm, z.B. ein Visual Basic Script, das eine ungewollte (Schadens)-Funktion ausführt
  - enthält Funktionen zur Fortpflanzung auf andere Rechner eines Netzes
  - Schutz: keine unbekannten Programme ausführen; keine unnötigen Rechte (insb. Ausführungsrechte) vergeben

Würmer: Wie gefährlich können pdf-Dateien sein?

- Beispiel eines Paches für iOS (iPhone, iPod touch) vom August 2010:

- Trojanisches Pferd
  - Programm, das (eine offen sichtbare Nutzfunktion und zusätzlich) eine verdeckte Schadensfunktion ausführt
  - nicht notwendigerweise Fortpflanzung
  - spezielle Formen:
    - Universelles Trojanisches Pferd
    - Transitives Trojanisches Pferd
  - Schutz: keine unbekannten Programme ausführen

  - Auch: Oberbegriff für Schadsoftware, die ohne Wissen des Opfers auf dessen System läuft

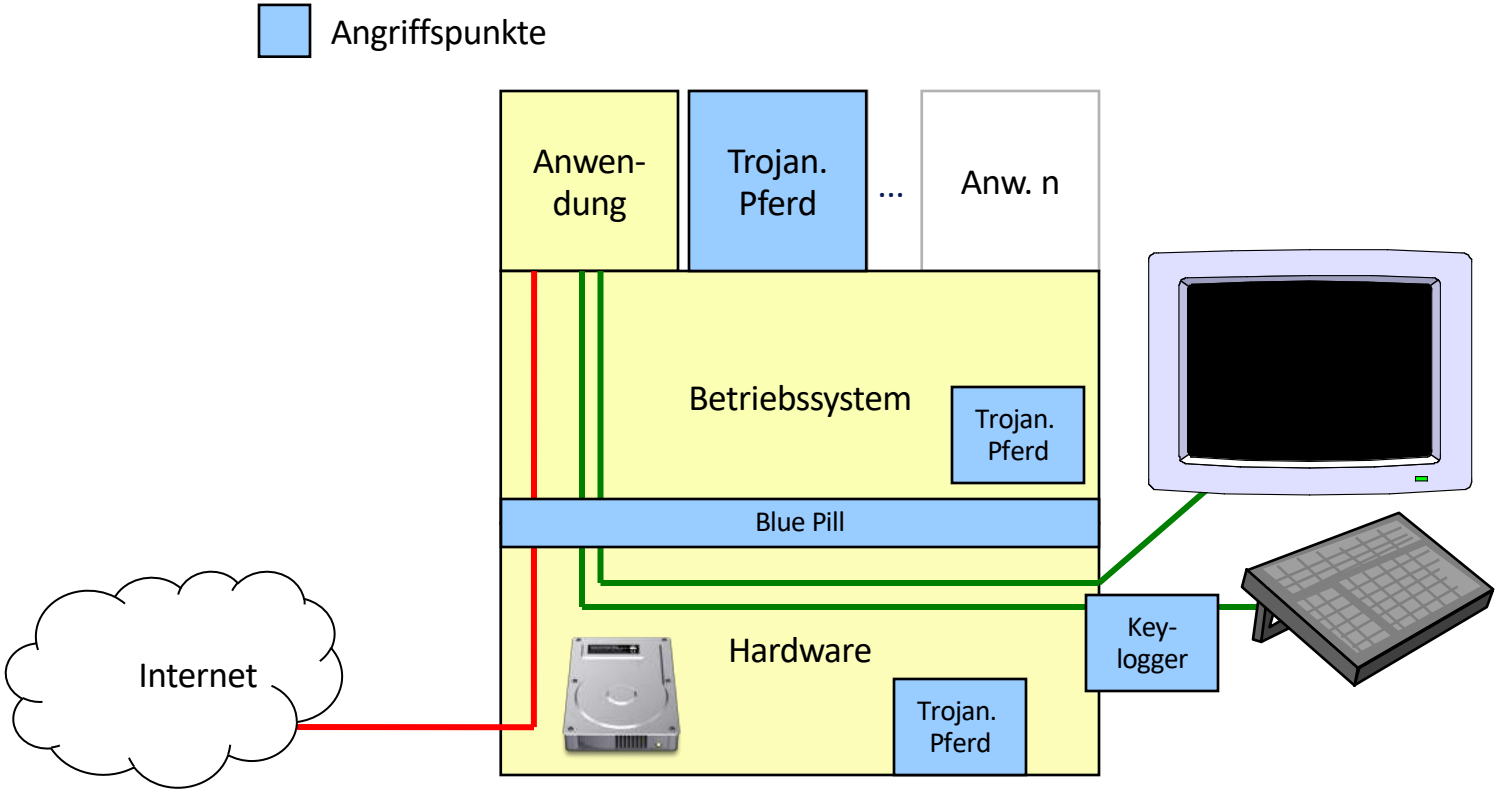
Falschmeldungen

- Hoaxes
  - Fortpflanzung: durch menschliche Fehleinschätzung
  - Hoax-Liste: http://www.hoax-info.de/

```
Von: Doris Külper
Datum: 04.03.2004 11:25:12
Betreff: WICHTIGE INFO

Handy Fangnummern im Umlauf

Für alle zur Info

Wenn auf dem Handydisplay die Mitteilung "Anruf in Abwesenheit" und
dann die Nummer: +49137799090269 oder +49172332233333 erscheint,
nicht zurückrufen. Es handelt sich hierbei um eine Fangnummer, die den
Anruf bis zu einer Stunde und länger hält.

Der Anrufer selbst hat keine Möglichkeit, den Anruf zu beenden.

Bitte geben Sie diese Nummer jedem weiter, den Sie kennen, damit böse
Überraschungen im Vorfeld schon vermieden werden.

Mit freundlichem Gruss

Doris Külper
Staatsanwaltschaft Hamburg
```

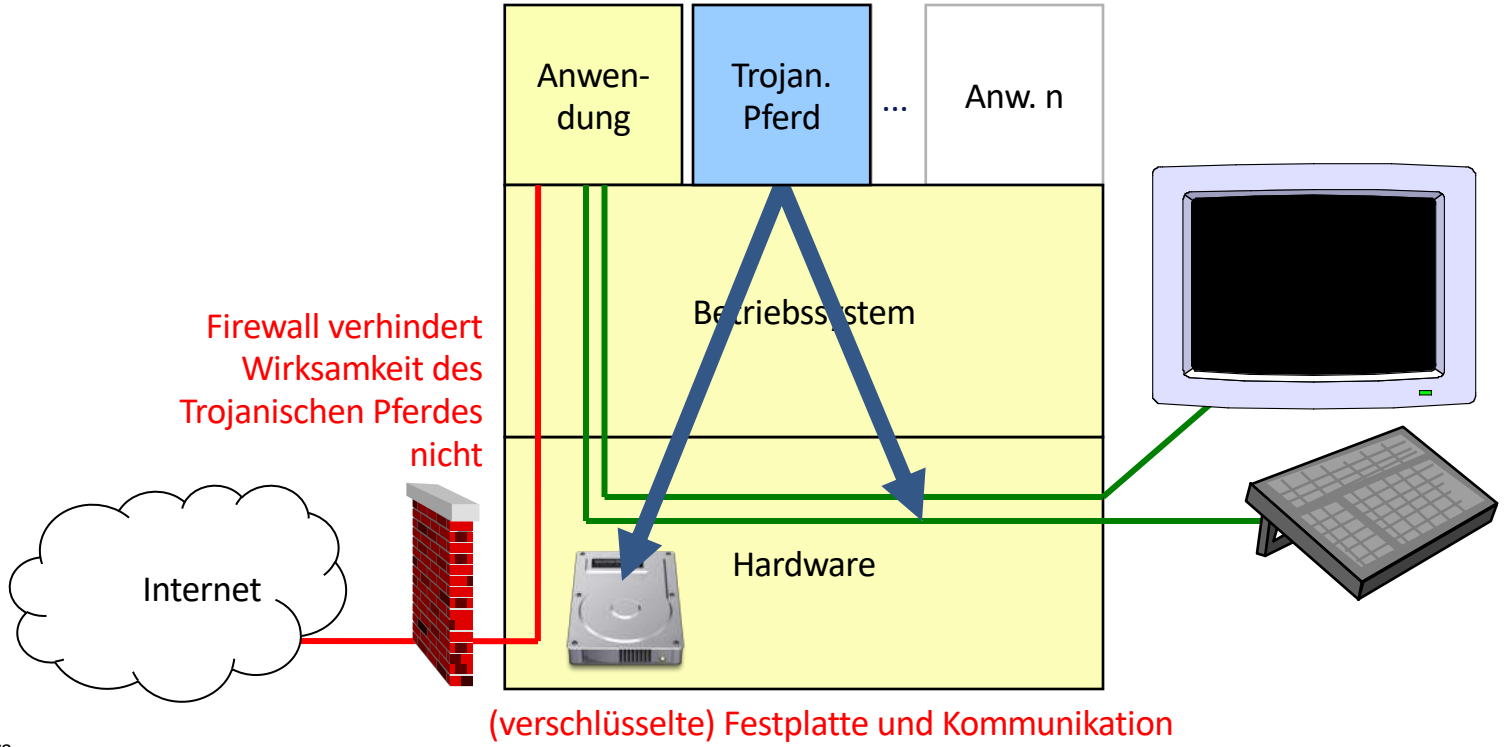
siehe auch: Wahr & falsch: Handy-Betrug mit 0137-Nummern.
https://hoax-info.tubit.tu-berlin.de/hoax/telefon0137.shtml

Angriffspunkte für Malware

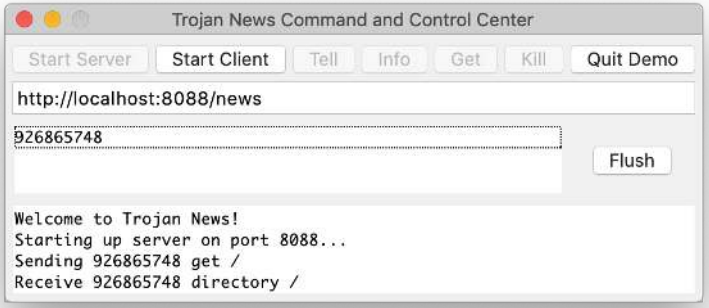


Trojanisches Pferd greift von innen an

Bösartige *Anwendung* hat Zugriffsrechte des Nutzers



Tunneln einer Firewall durch universelles Trojanisches Pferd



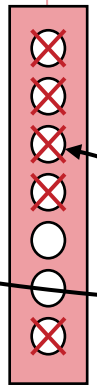
TrojanNews hat ca. 340 Zeilen Python-Code, davon 15 Zeilen Schadcode. Zum Vergleich: Loveletter (I-Love-You-Virus) 330 Zeilen.

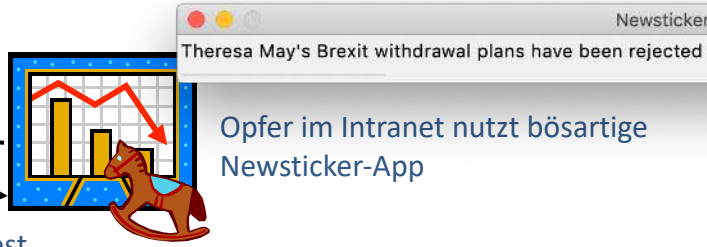
Es ist weniger eine Kunst, ein Trojanisches Pferd zu programmieren. Das Problem für den Angreifer besteht darin, es unbemerkt beim Opfer zu platzieren bzw. diesen zu überlisten, es selbst zu installieren.

Angreifer im Internet

Anstelle eines auffälligen User-Ports wird ein unauffälliger Port, z.B. http (80) verwendet.

2000





Opfer im Intranet nutzt bösartige Newsticker-App

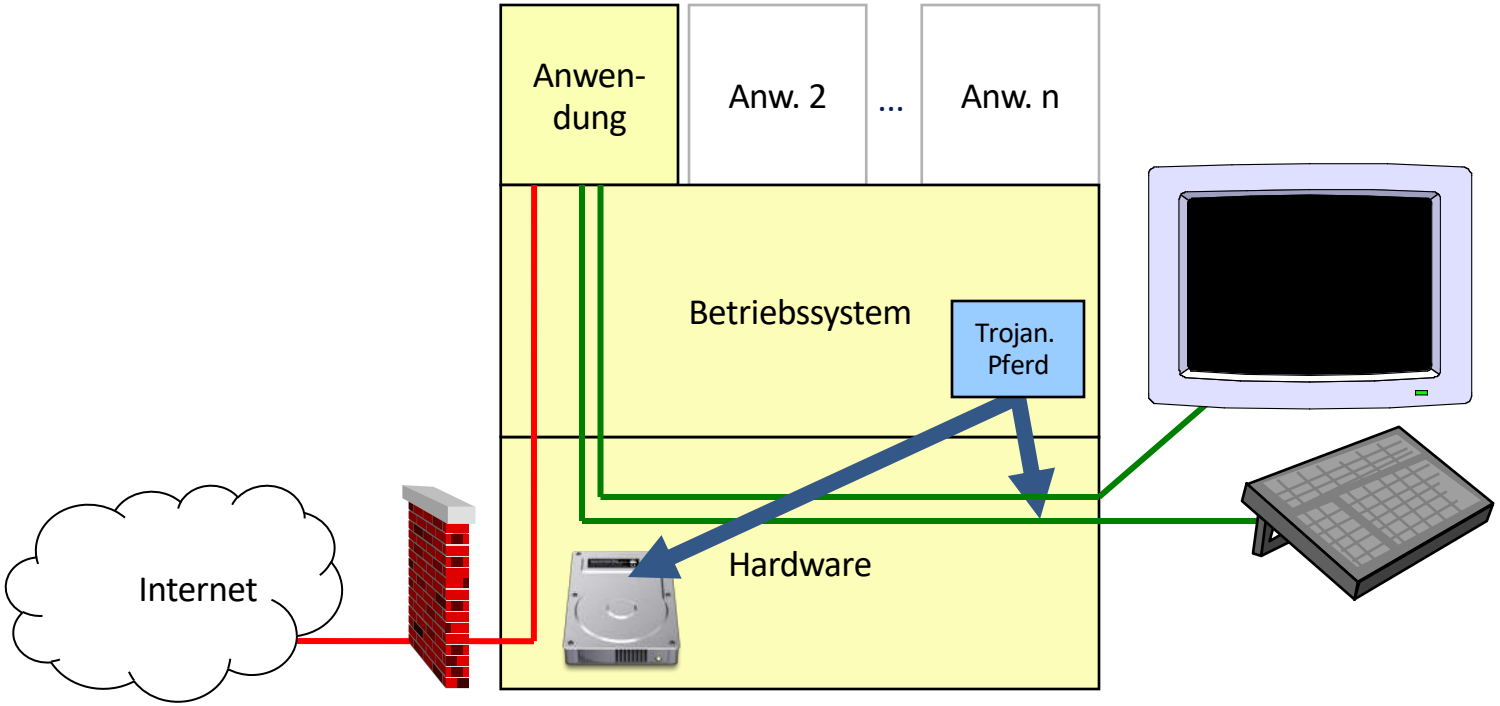
80

Verdeckter Eingabekanal:
Web-Response, die keine ist; Firewall verhindert nicht die Wirkung des trojanischen Pferdes

http-Request,
der in Wirklichkeit den Steuerbefehl abholt oder vertrauliche Daten versendet

Trojanisches Pferd greift von innen an

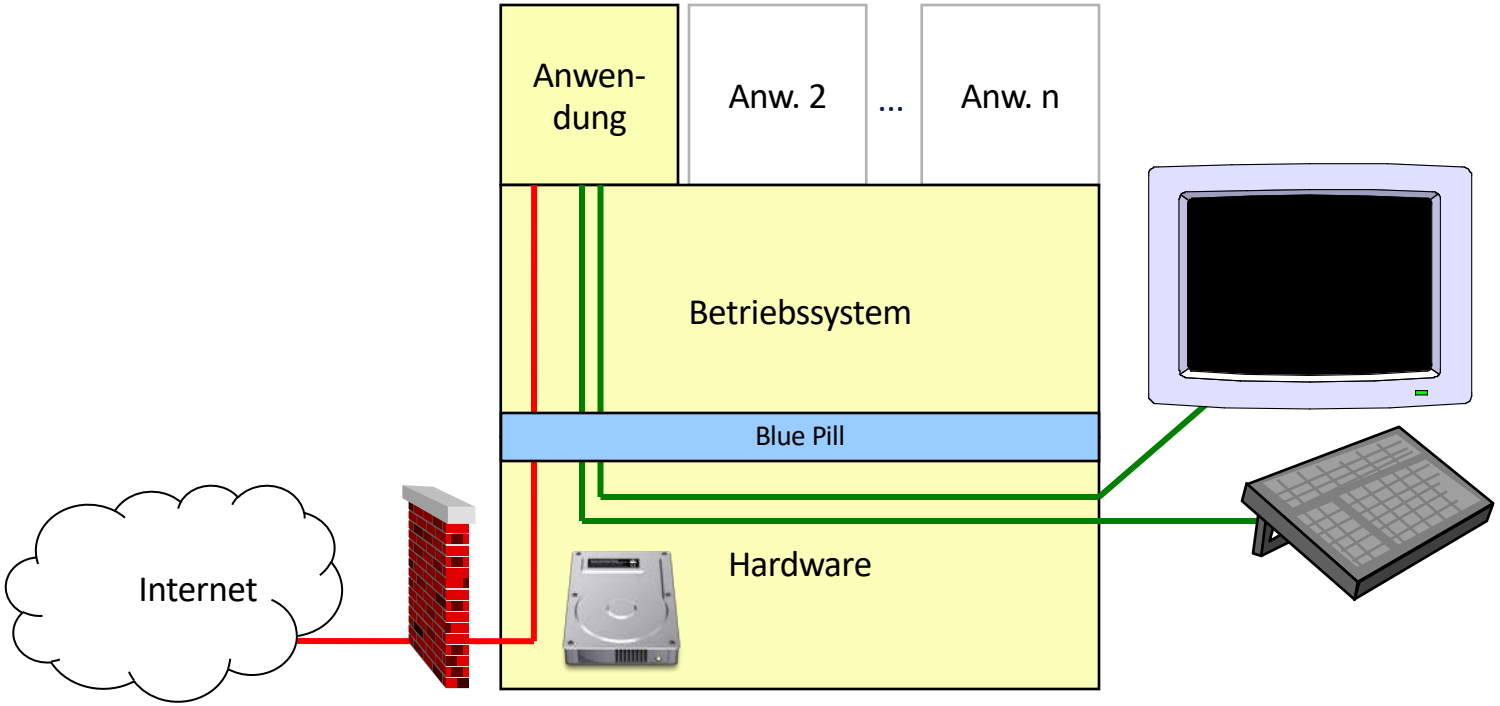
Bösartige *Betriebssystemkomponente (z.B. Treiber)* könnte Texteingaben abfangen, verschlüsselte Festplatten lesen, …



verschlüsselte Festplatte und Kommunikation

Trojanisches Pferd greift von innen an

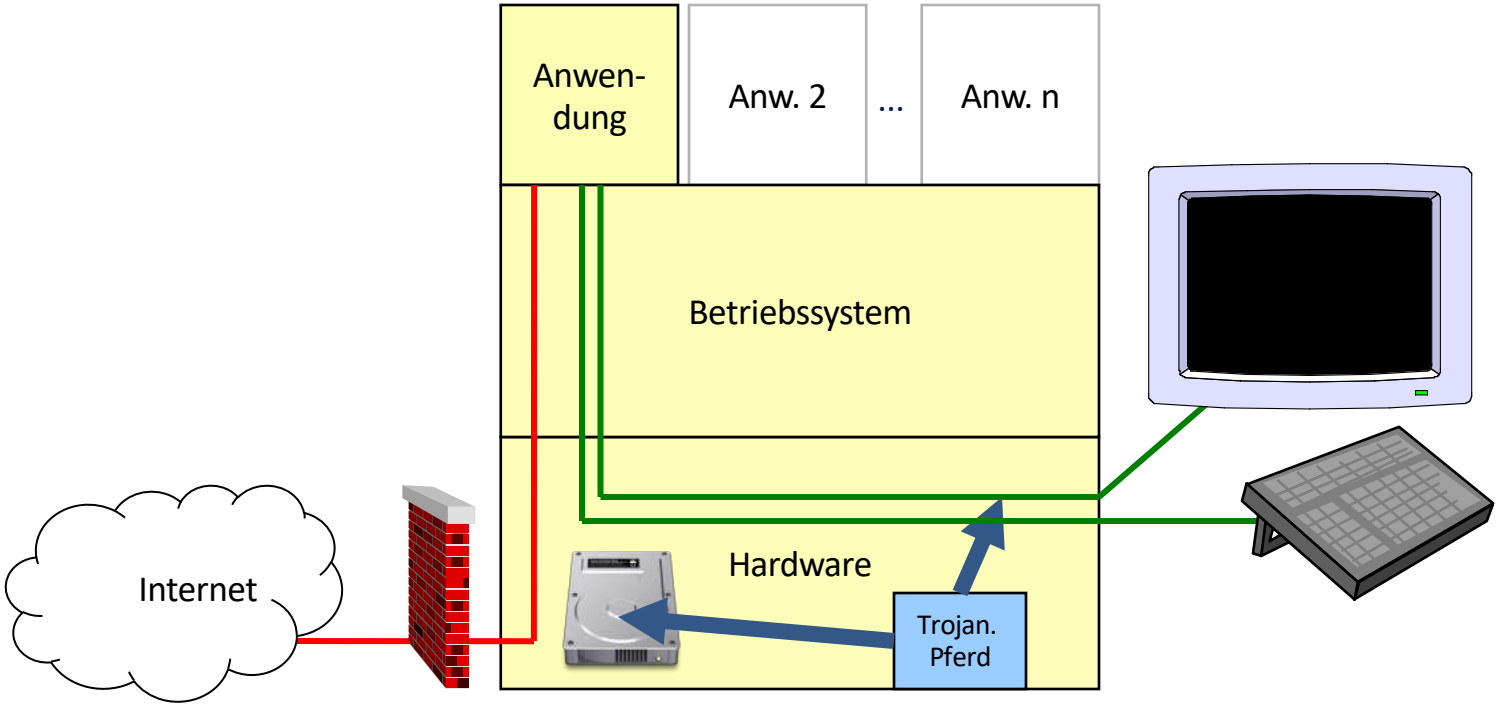
Bösartige *Virtualisierungsschicht (z.B. Blue Pill)* könnte dem Betriebssystem einen »sauberen« Rechner vorgaukeln



verschlüsselte Festplatte und Kommunikation

Trojanisches Pferd greift von innen an

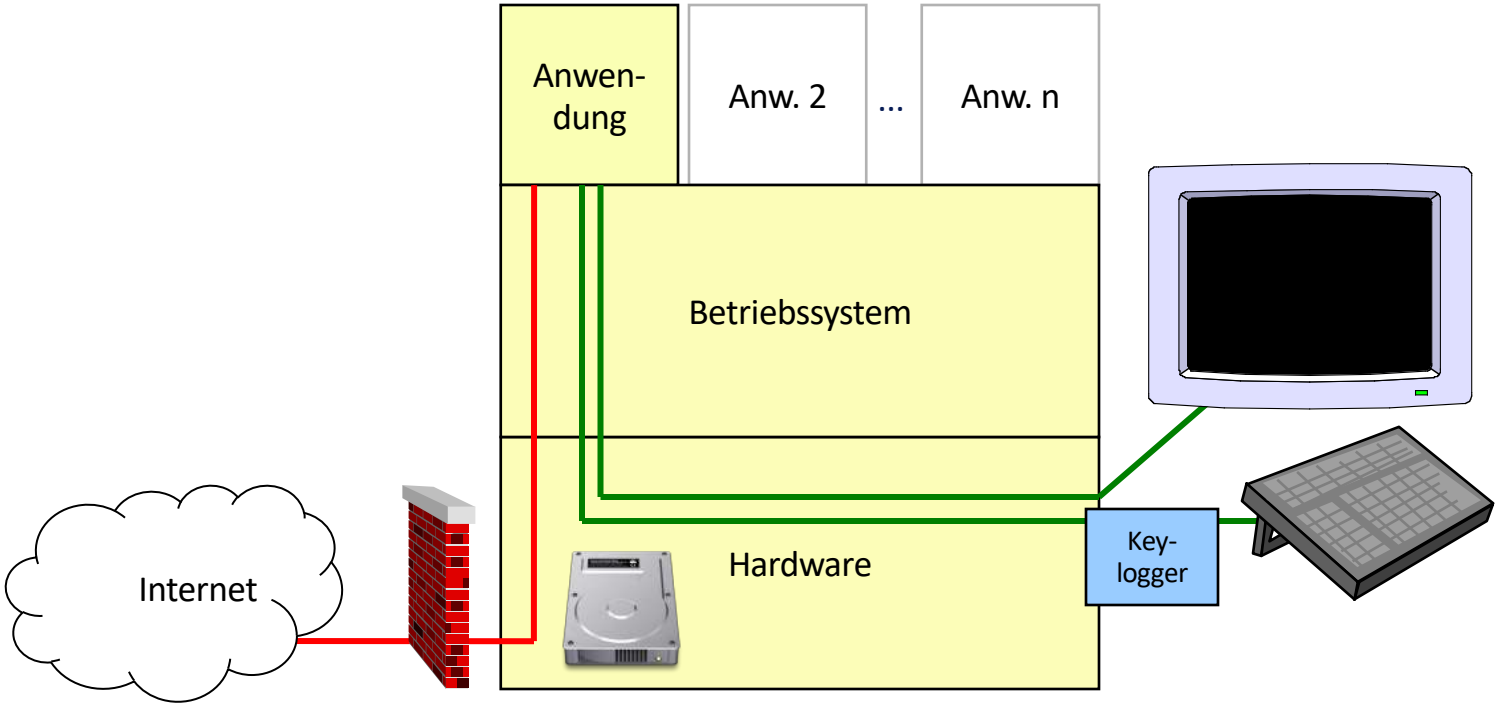
Bösartige *Hard-/Firmware* könnte Texteingaben abfangen, verschlüsselte Festplatten lesen, …



verschlüsselte Festplatte und Kommunikation

Trojanisches Pferd greift von innen an

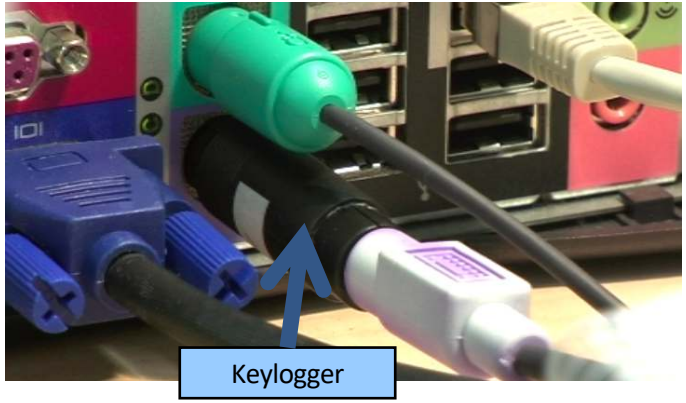
Bösartige *Hardware (z.B. Keylogger)* könnte Texteingaben (z.B. Passwort der Festplattenverschlüsselung) abfangen

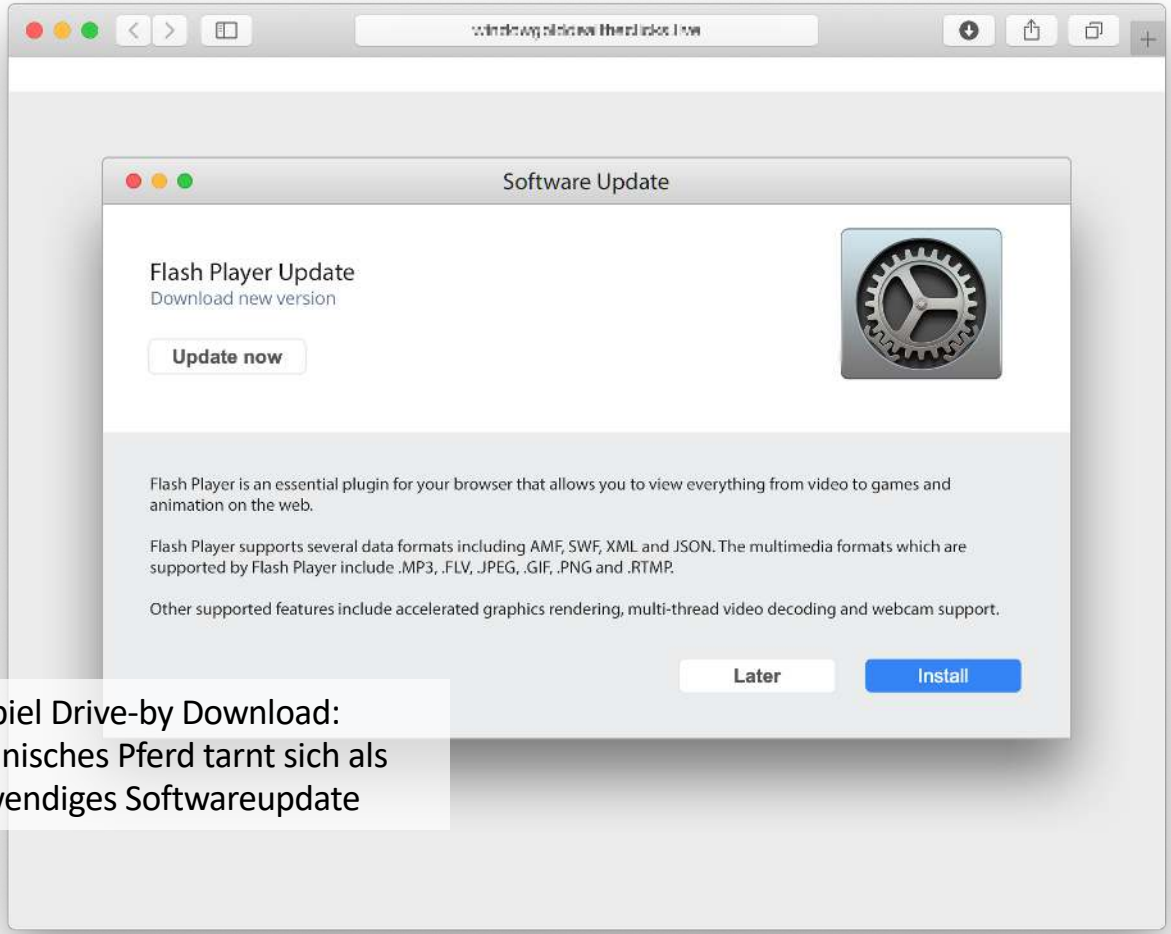


Beispiele für Malware-Angriffe

- Präparierte USB-Sticks
  - auf Parkplatz »verlieren«

- Bewerbung auf offene Stelle
  - Begleitbrief und Unterlagen auf bewusst infizierter, beigelegter CD-ROM
    Quelle: Ivan Bütler: Inside-Out Angriffe. DuD 30/10 (2006) 607
- Reinigungspersonal
  - installiert Hardware-Keylogger zum Abfangen von Passwörtern
- Drive-by Downloads
  - hinterlistiges Installieren von Schadsoftware
- Manipulierte USB-Ladesteckdosen
  - Schutz: USB-Kondom





Beispiel Drive-by Download:
Trojanisches Pferd tarnt sich als notwendiges Softwareupdate

Software Update

Flash Player Update

Download new version

Elemente | Netzwerk | Ressourcen | Timelines | Debugger | Speicher | Canvas | Konsole | Suchen

Alle Ressourcen | Dokumente

- InnYqmGlFccZSSAELzLK893sAhOnuVE0yAnCi...
- morphonebrities.info
- arrow__blue.png — d2qxrzejb7t810.cloudfro...
- clean_k@2x.png — d2qxrzejb7t810.cloudfro...
- pattern__safari-arrow.png — d2qxrzejb7t810...
- pattern__safari1.jpg — d2qxrzejb7t810.cloud...

InnYqmGlFccZSSAELzLK893sAhOnuVE0yAnCiwMQSvg

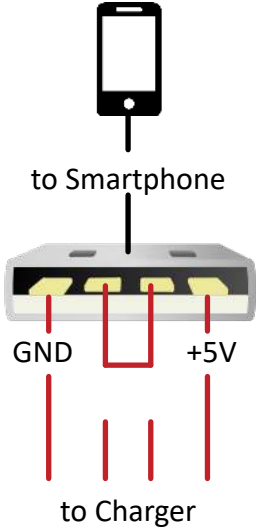
```
<!DOCTYPE html><html> <head><meta charset=UTF-8><meta name=viewport
content="width=device-width"><title>Please update</title><link rel=icon
href="
LGPC/xhBQAAAAFzUkdCAK7OHOkAAABIUExURZoAAPrw8KogIPTk5Pz19e01tezPz5wAAKcXF////6UQEJ4DA+
a+vrtQUM6CgrAxMcRpabVBQdmenujExKQODtaXl+jDw9OOjmMvznAAAABQSURBVBgZlcG3EcAwDASwVySp7Lz
/pm58FlsBWJETQaEaCkMZRiJj6s43S5geCRbKGcXcjGkz3h2MH+1yMRM+ZJt3HRNHMQMKl1AJCqWMBS8h5AIo
iQfkRgAAAABJRU5ErkJggg==" type=image/png><meta name=robots content="noindex,
nofollow"><style>html,body,div{margin:0;padding:0;border:0;font-size:100%;font:inheri
t;vertical-align:baseline}body{line-height:1}body:after{display:none;content:url(http
://d2qxrzejb7t810.cloudfront.net/lps/flash_mac/images/clean_k.png)
url(http://d2qxrzejb7t810.cloudfront.net/lps/flash_mac/images/downloadgif.gif)
url(http://d2qxrzejb7t810.cloudfront.net/lps/flash_mac/images/downloadactive.png)
url(http://d2qxrzejb7t810.cloudfront.net/lps/flash_mac/images/ok.png)
url(http://d2qxrzejb7t810.cloudfront.net/lps/flash_mac/images/okactive.png)
url(http://d2qxrzejb7t810.cloudfront.net/lps/flash_mac/images/okactive@2x.png)
url(http://d2qxrzejb7t810.cloudfront.net/lps/flash_mac/images/downloadactive@2x.png)}
body{background:#ececec;color:#000;font:14px
Arial,Helvetica,sans-serif}a:hover,a:active{outline:0}.clearfix:before,.clearfix:afte
r{content:"";display:table}.clearfix:after{clear:both}.clearfix{*zoom:1}.video{width:
775px;margin:20px auto;height:435px;background:#fffffd;border:1px solid
#ccc;border-radius:10px}.content{width:609px;margin:0
auto}#prozor{position:relative;z-index:2;width:786px;height:578px;margin:30px
auto;background:url(http://d2qxrzejb7t810.cloudfront.net/lps/flash_mac/images/clean_k
```

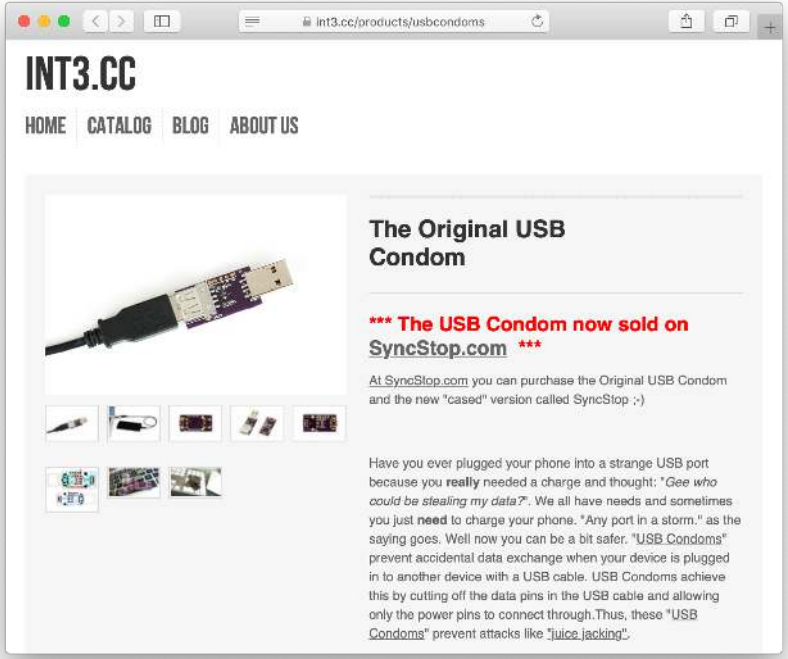
Filter

Beispiel Drive-by Download:
Trojanisches Pferd tarnt sich als notwendiges Softwareupdate

USB charging condom

- USB-A has 4 wires
- cut 2 inner data wires and short-circuit
- connect power wires only





Fernwartung, mobile Datenträger und Schadsoftware

- Fernwartung
  - Fernwartung ist zu vergleichen mit der Existenz eines universellen trojanischen Pferdes
  - Schutz: wenn einmal vorhanden: keiner
  - Vertrauen ist nötig

- Mobile Datenträger
  - Dateien auf mobilen Datenträgern (USB-Sticks, MP3-Player, Handies, CDs, etc.) sind analog dem Download von Dateien aus dem Internet zu behandeln:
    - nur Berechtigte nach Virenscan
    - Anwendung einer Dateischleuse, gelegentlich auch auch Wechseldatenträgerschleuse oder Sheep-Dip Machine genannt

Bundestrojaner, Stuxnet, Flame

- Darf sich der Staat auch solcher Angriffsmethoden bedienen,
  - die zwar dem Schutz des Staates und seiner Bürger dienen, jedoch die Integrität und Vertrauenswürdigkeit von IT-Systemen untergraben und schlimmstenfalls auch gegen ihn selbst verwendet werden können?

- Antwort: »Neues Computergrundrecht«
  - Bundesverfassungsgericht im Februar 2008:
    - Grundrecht auf »Gewährleistung von Vertraulichkeit und Integrität informationstechnischer Systeme«
    - Erlaubte Einschränkungen:
      - Gefährdung von Leib, Leben und Freiheit einer Person
      - Gefährdung der Grundlagen des Staates
      - Gefährdung der Grundlagen der Existenz der Menschen

Darf sich der Staat auch solcher Angriffsmethoden bedienen?

- Implementierungen
  - politisch motivierte staatliche Angriffe mit oder auf IT-Systeme anderer Staaten (Cyberwarfare)
    - Stuxnet, (Duqu,) Flame
  - Gesetzlich erlaubte Telekommunikationsüberwachung und Beweissicherung (Online Durchsuchung) direkt auf dem PC eines Verdächtigen
    - Staatstrojaner / Bundestrojaner

Stuxnet: Angriffsszenario Industrieanlage

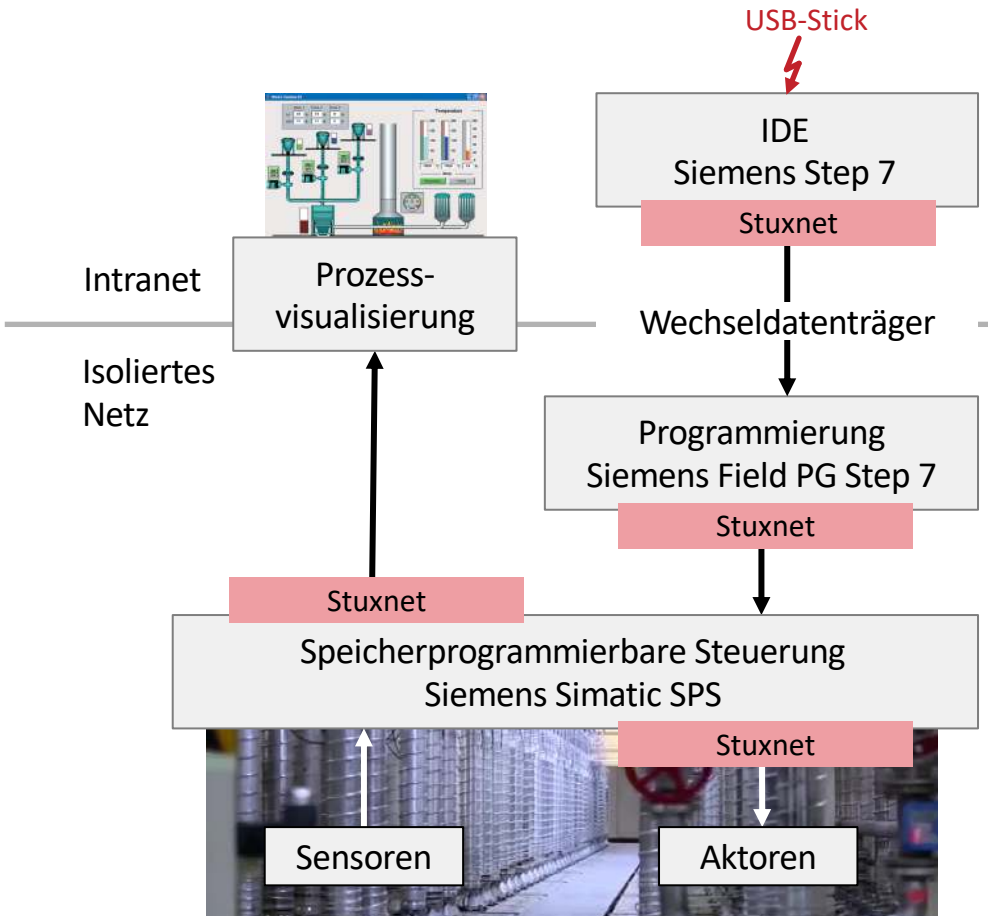
- Internetwurm, der mit dem Ziel entwickelt wurde, die innerbetrieblichen Abläufe eines speziellen Typs von Industrieanlagen empfindlich zu stören
  - Entdeckung im Juli 2010
  - Ziel: Unbemerkte Änderung von Programmteilen in speicherprogrammierbaren Steuerungen (SPS)
  - Verwendet vier Zeroday-Exploits zur Verbreitung und Rechteausweitung
  - Selbstzerstörung (nur Windows-Komponente) nach 35 Tagen

- Urheber: USA und Israel
  - Anordnung von 2006 von US-Präsident Gerorge W. Bush
  - in 2010 bestätigt durch Präsident Obama

  Quelle: New York Times: Obama Order Sped Up Wave Of Cyberattacks Against Iran. June 1, 2012, page A1

Bild von Natanz (Majid Saeedi/Getty Images)

Stuxnet:
Angriffszenario Industrieanlage



Treiber mit gestohlenen Zertifikaten (Realtek, JMicron) signiert

Manipulation Step-7-IO-Treiber und -Projektdateien

Vermutlich genaue Kenntnis der Anlagen vorhanden gewesen

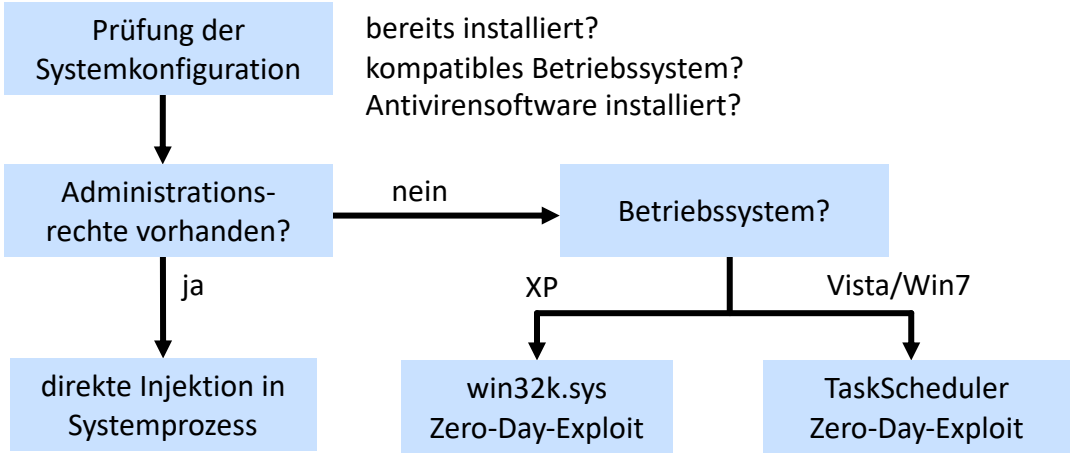
Stuxnet überprüft Konfiguration der SPS und befällt nur bestimmte Systeme

Transitives trojanisches Pferd, da zunächst die IDE befallen wird, um anschließend die SPS zu manipulieren.

Mechanische Zerstörung (Drehzahlmanipulation) der Zentrifugen

Installation (Dropper)

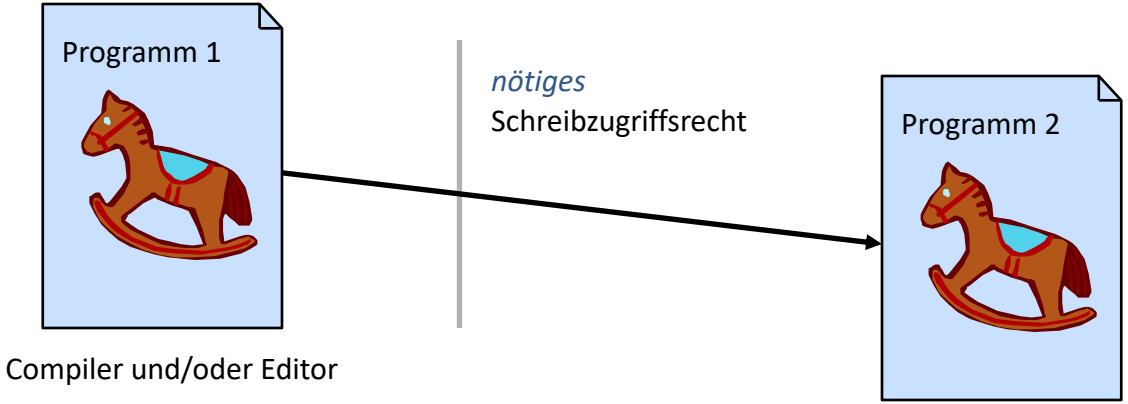
- Code-Injektion in Systemprozess



- Besonderheiten
  - befällt nur bestimmte Systeme (laut Symantec ca. 70% im Iran)
  - transitiver Infektionsweg über mehrere Systemgrenzen hinweg
  - P2P-Kommunikation infizierter Systeme
  - Update-Mechanismus

Transitives trojanisches Pferd

- pflanzt sich über mehrere Entwicklungsschritte fort
- Beispiele:
  - Compiler enthält Trojanisches Pferd; jedes neu compilierte Programm enthält ebenfalls wieder das trojanische Pferd
  - Texteditor schleust heimlich Schadcode (Quellcode, Makrocode, etc.) in ein Dokument ein
  - Kein Schutz durch geringstmögliche Privilegierung

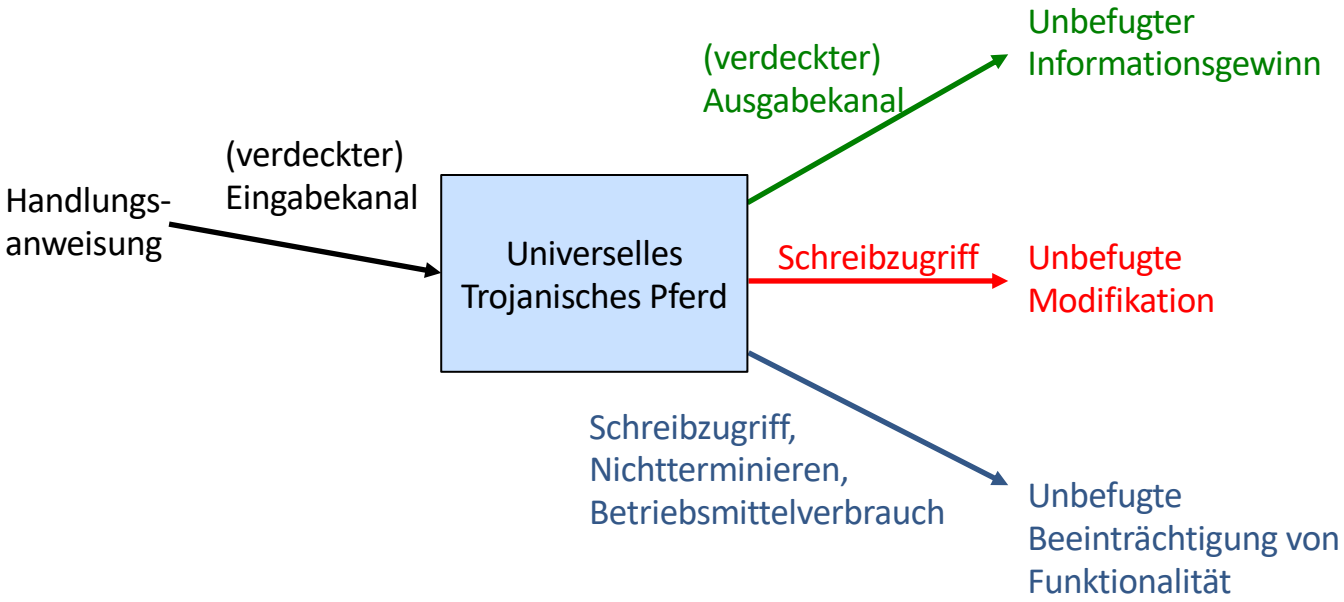


Flame

- Universelle Schadsoftware zur Spionage auf Windows-Rechnern
  - Screenshots, Einschalten des Mikrofons am Rechner
  - Sniffing des lokalen Netzverkehrs
  - Zugriff auf Bluetooth-Geräte
  - Nachladen weiterer Schadfunktionen
  - Keine automatische Selbstzerstörung, Deinstallation per Befehl
- Stuxnet-Nachfolger?
  - Ziel des Angriffs: Rechner im Iran (wie Stuxnet)
  - Angreifer: mutmaßlich programmiert durch USA und Israel
- Merkmale
  - im Mai 2012 entdeckt, Teile des Codes deutlich älter
  - modularer Aufbau
  - zusammen ca. 20 Mbyte Code = Drohkulisse?
  - Dezentrale Steuerzentrale (Command and Control Server)

Universelles Trojanisches Pferd

- Schadensfunktion kann von außen beeinflusst werden
- empfängt Handlungsanweisung von außen



Flame

- Infektions- und Replikationsmechanismus
  - vollständig aktuelles Windows 7 mittels Windows Update Funktion infizierbar:
    - Umleitung der Update Requests noch nicht infizierter Rechner auf bereits infizierten Rechner im (lokalen) Netz
    - Installation eines falschen, korrekt signierten »Systemupdates«

Flame nutzte unbekannte Schwachstelle in den Zertifizierungsfunktionen von Windows -> Code von Flame wurde unberechtigt signiert

> Zero-Day Exploits ?

  - Zertifikat wurde inzwischen zurückgerufen
  - Aktuelle Windows-Systeme nicht mehr infizierbar

Staatstrojaner

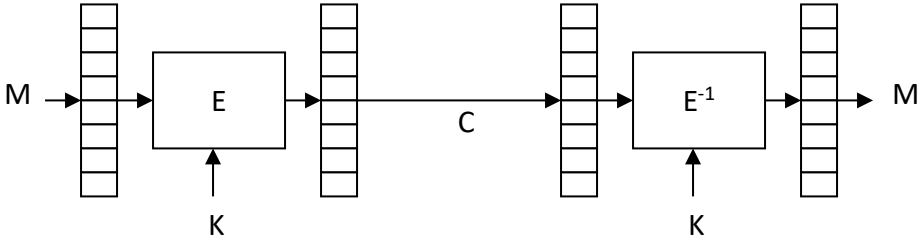
- Haupteinsatzgebiet ist die sog. Quellen-Telekommunikationsüberwachung (Quellen-TKÜ)
  - Oktober 2011 entdeckt
  - Mehrfach unabhängig durch Reverse Engineering analysiert und publiziert
    - Chaos Computer Club
    - Universität Mannheim

- Auftragsarbeit:
  - Auftraggeber: deutsche Sicherheitsbehörden
  - entwickelt von Digitask GmbH
  - geht vermutlich zurück auf eine Skype-Capture-Unit vom September 2007

Staatstrojaner

- Funktionen
  - Ausleiten des Skype-Datenverkehrs
  - Screenshots, Mikrofon einschalten, Keylogger
  - Nachladefunktion

  - Schwach verschlüsselte Kommunikation mit einem Command-and-Control-Server im Ausland (USA)
    - AES im ECB-Mode mit fest kodiertem Schlüssel

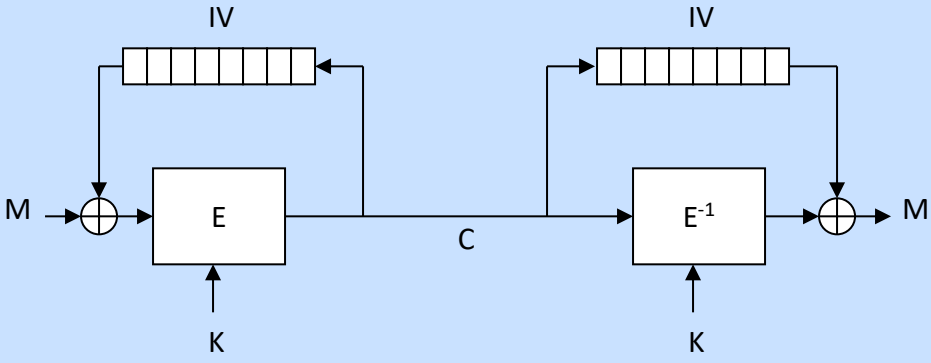


$C=E(K,M)$ $M=E^{-1}(K,C)$

Staatstrojaner

- Schwach verschlüsselte Kommunikation mit einem Command-and-Control-Server im Ausland (USA)
  - AES im ECB-Mode mit fest kodiertem Schlüssel

- Richtig wäre: mindestens Cipher Block Chaining (CBC) verwenden



$C_0=IV$
$C_i=E(K, M_i \oplus C_{i-1})$

$C_0=IV$
$M_i=E^{-1}(K, C_i) \oplus C_{i-1}$

Spannungsfeld von Freiheit und Sicherheit

- Ziel der Informationssicherheit: möglichst wenig Vertrauen in andere setzen müssen
  - Wo keine Sicherheit erreichbar ist, bleibt nur Vertrauen [müssen]

- Freiheit: insbesondere Grundrechte
  - Recht auf informationelle Selbstbestimmung
  - Gewährleistung von Vertraulichkeit und Integrität informationstechnischer Systeme (Computer-Grundrecht)

- Sicherheit
  - Vorratsdatenspeicherung
  - Bundestrojaner

»Nur in schweren Fällen …«

… aber die Bürger erleiden Vertrauensverlust gegenüber dem Staat

Eingriffstiefe in die Freiheit

- Darf sich der Staat solcher Angriffsmethoden bedienen?

	Spurensuche	Protokollierung	Überwachung	Angriff
Reaktiv	✓	Anfrage an Dienstanbieter	TKÜ	Tallin Manual
Präventiv	Raster-fahndung	Vorratsdaten-speicherung	XKeyScore	⊘

VSS 17.04.2020

Grundlagen der Kryptographie

Kriterien zur Einteilung von Kryptosystemen

- Kryptographische Basisbausteine
  - Konzelationssysteme
  - Authentikationssysteme
  - Hashfunktionen
  - Pseudozufallszahlengeneratoren
- Schlüsselbeziehung Sender–Empfänger
  - Symmetrische Systeme
  - Asymmetrische Systeme
- Alphabet, auf dem die Chiffre operiert
  - Blockchiffre: Operiert auf Blöcken von Zeichen
  - Stromchiffren: Operiert auf einzelnen Zeichen
- Längentreue
- Erreichbare Sicherheit

Anwendungsfall x Schlüsselbeziehung

	Konzelation (Verschlüsselung)	Authentikation
symmetrische	*One-time-pad, DES, Triple-DES, AES, IDEA, A5/1 (GSM), A5/2 (GSM) …* GnuPG/PGP, WPA2, IPSec, TLS	*Symmetrische Authentikationscodes, CCM, A3 (GSM), …* SecurID, WPA2, IPSec, TLS
asymmetrische	*RSA, ElGamal, McEliece, …* GnuPG/PGP, HBCI, TLS	*RSA, ElGamal, DSA, GMR, …* GnuPG/PGP, HBCI, TLS

Die übliche Bezeichnung für ein asymmetrisches Authentikationssystem ist »Digitales Signatursystem«.

Algorithmus Anwendung

Erreichbare Sicherheit

- Sicherheit
  - (informations) theoretisch sicher
  - kryptographisch stark (beweisbar)
    - gegen aktive Angriffe
    - gegen passive Angriffe
  - wohluntersucht (praktisch sicher)
    - Chaos
    - Zahlentheorie

    komplexitäts-theoretisch sicher
  - geheim gehaltene

- Kerckhoffs-Prinzip
  - Die Sicherheit eines kryptographischen Verfahrens soll nur von der Geheimhaltung des kryptographischen Schlüssels abhängen.
    - Geht zurück auf
      Auguste Kerckhoffs: La cryptographie militaire, 1883

Angriffsarten

- Ciphertext Only Attack
  - Angreifer kennt nur Schlüsseltext
- Known Plaintext Attack
  - Angreifer kennt Klartext-Schlüsseltext-Paare
- [Adaptively] Chosen Plaintext (Ciphertext) Attack
  - Adaptively:
    - Angreifer kann in Abhängigkeit vorheriger gewählter Nachrichten neue Nachrichten wählen
  - Non-adaptively:
    - Angreifer muss alle Nachrichten zu Beginn wählen, kann also nicht abhängig vom Verschlüsselungsergebnis, weitere Nachrichten wählen

Angriffsarten

		Authentikations-systeme:	Konzelations-systeme:
Brechen	=	Fälschen	Entschlüsseln

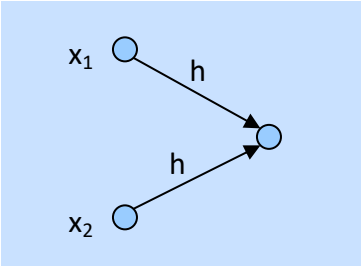
- Vollständiges Brechen: Finden des Schlüssels
- Universelles Brechen: Finden eines zum Schlüssel äquivalenten Verfahrens
- Nachrichtenbezogenes Brechen: Brechen für einzelne Nachrichten, ohne den Schlüssel selbst in Erfahrung zu bringen
  - selektives Brechen: für eine bestimmte Nachricht
  - existenzielles Brechen: für irgendeine Nachricht

- Aufwand/Kosten:
  - Einmalige Kosten, jeder Schlüssel effizient knackbar
  - Jeder Angriff verursacht Kosten beim Angreifer

Hashfunktionen

- Abbildung h: X $\rightarrow$ Y
  - Einwegfunktion (auch: Falltürfunktion)
  - Umkehrfunktion nicht effizient berechenbar
- Hashfunktionen sind verkürzend
  - Beliebig lange Inputs werden auf Output bestimmter Länge abgebildet, z.B. SHA3 mit 256 Bit (16 Blöcke zu je 16 Bit in Hex)

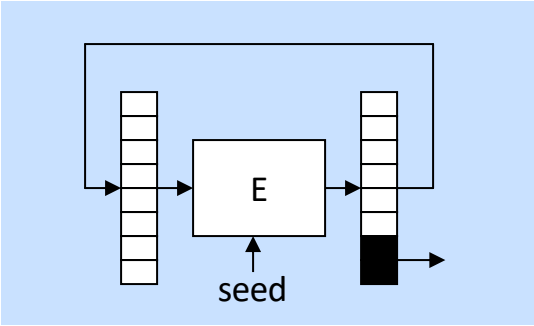
```
17EC 1A95 14E4 F581 7C68 2AC1 0939 D2CA
9879 FFBF AA3B 43BD 9A95 55DB 891C F3BF
```



  - Kollision:
    - $h(x_1) = h(x_2)$ mit x_1 != x_2
- Kryptographische Hashfunktionen sind kollisionsresistent
  - nicht mit vertretbarem Aufwand möglich, eine Kollision gezielt herbeizuführen, z.B. Finden eines x_2 zu einem gegebenen $h(x_1)$

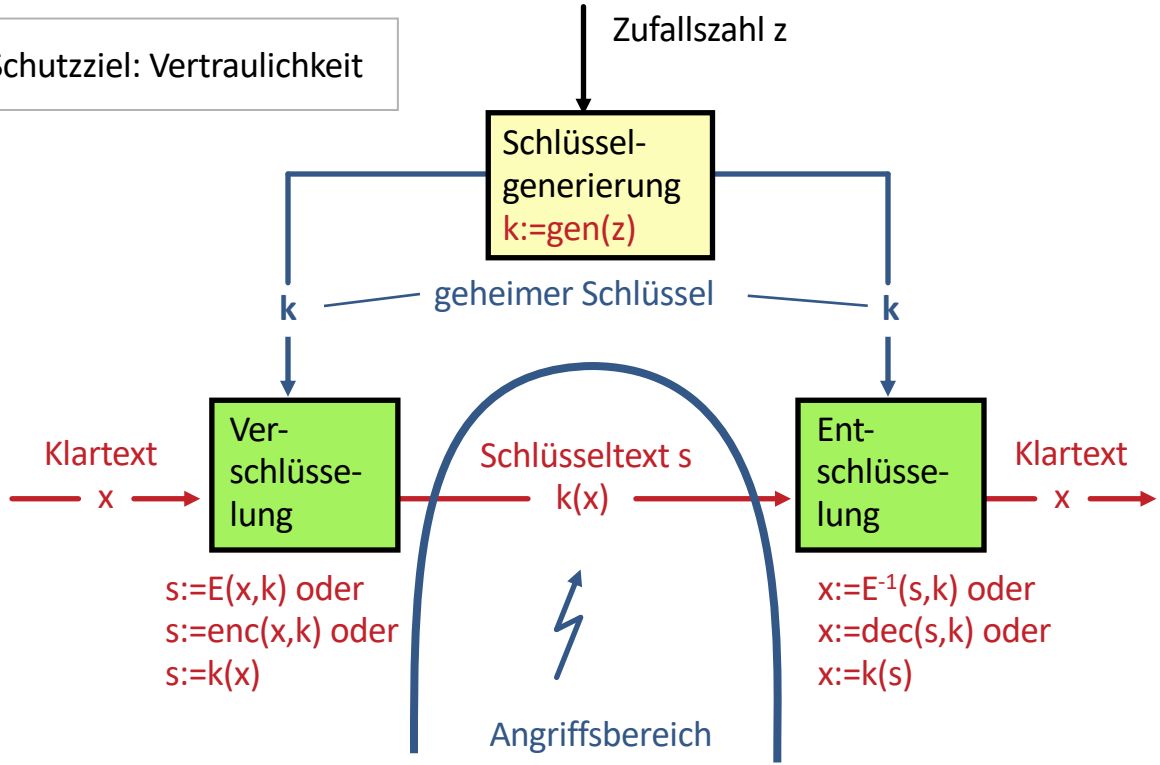
Pseudozufallszahlengeneratoren (PZZG)

- Anwendungsbereich: Erzeugung von
  - Sitzungsschlüsseln
  - einfachen Pseudonymen
  - (kurzen) Zufallswerten: salts, seeds, nonces, challenges, padding
- Eigenschaften
  - PZZG erzeugen aus gleichem Startwert (seed) stets die gleiche Zufallsbitfolge
  - erfüllen ansonsten die statistischen Eigenschaften von echten Zufallszahlen
  - Kenntnis einer Teilmenge vom Zufallsbitfolgen ermöglicht keine Rückschlüsse auf innere Zustände
- Umsetzungsmöglichkeiten, z.B.
  - Zahlentheoretische Generatoren: z.B.
    s^2 mod n-Generator
  - Rückgekoppelte (Nicht)-lineare Schieberegister
  - Rückgekoppelte Blockchiffren:



Symmetrische Verschlüsselung

Schutzziel: Vertraulichkeit

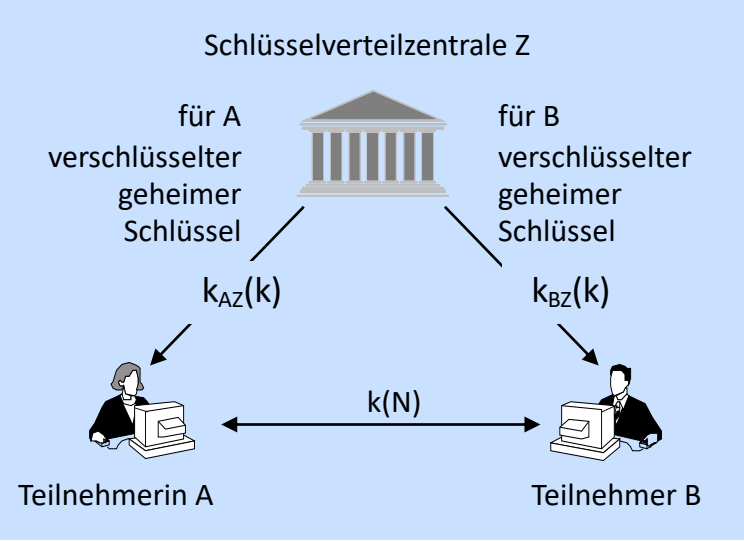


»Undurchsichtiger Kasten mit Schloss. Es gibt zwei gleiche Schlüssel.«

Schlüsselverteilung für symmetrische Systeme

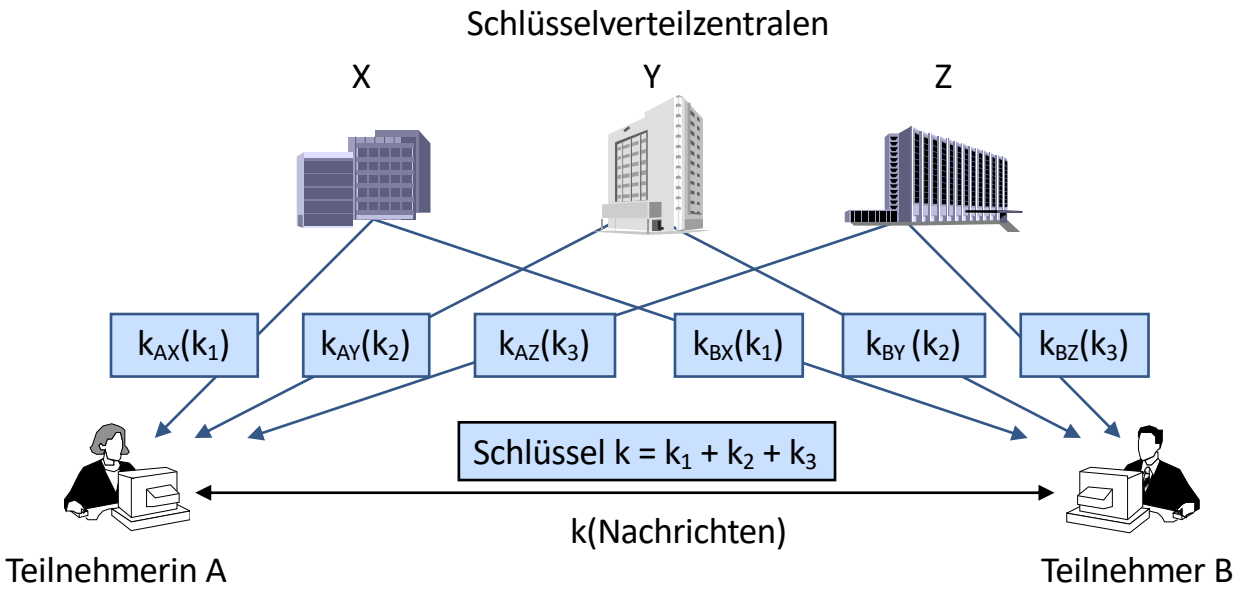
- Schlüsselaustausch:
  1. A und B tauschen Offline jeweils einen symmetrischen Schlüssel mit Z aus: k_{AZ} und k_{BZ}
  2. Z generiert auf Anforderung einen symmetrischen Kommunikationsschlüssel k und verschlüsselt diesen für A und B:
     $k_{AZ}(k) \rightarrow A$
     $k_{BZ}(k) \rightarrow B$
  3. A und B entschlüsseln k

- Kommunikation:
  – Sender verschlüsselt Nachricht N mit k:
    $k(N)$

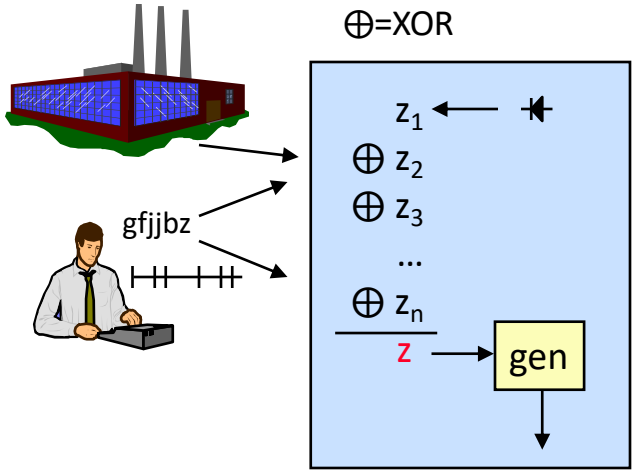


Dezentralisierte Variante

- Dezentralisierte Schlüsselverteilung ist möglich
- Ziel: Alle beteiligten Schlüsselverteilzentralen müssen zusammen arbeiten, damit sie den Kommunikationsschlüssel k erfahren
- Überlagerung der Teilschlüssel z.B. mit XOR-Verknüpfung



Schlüsselgenerierung

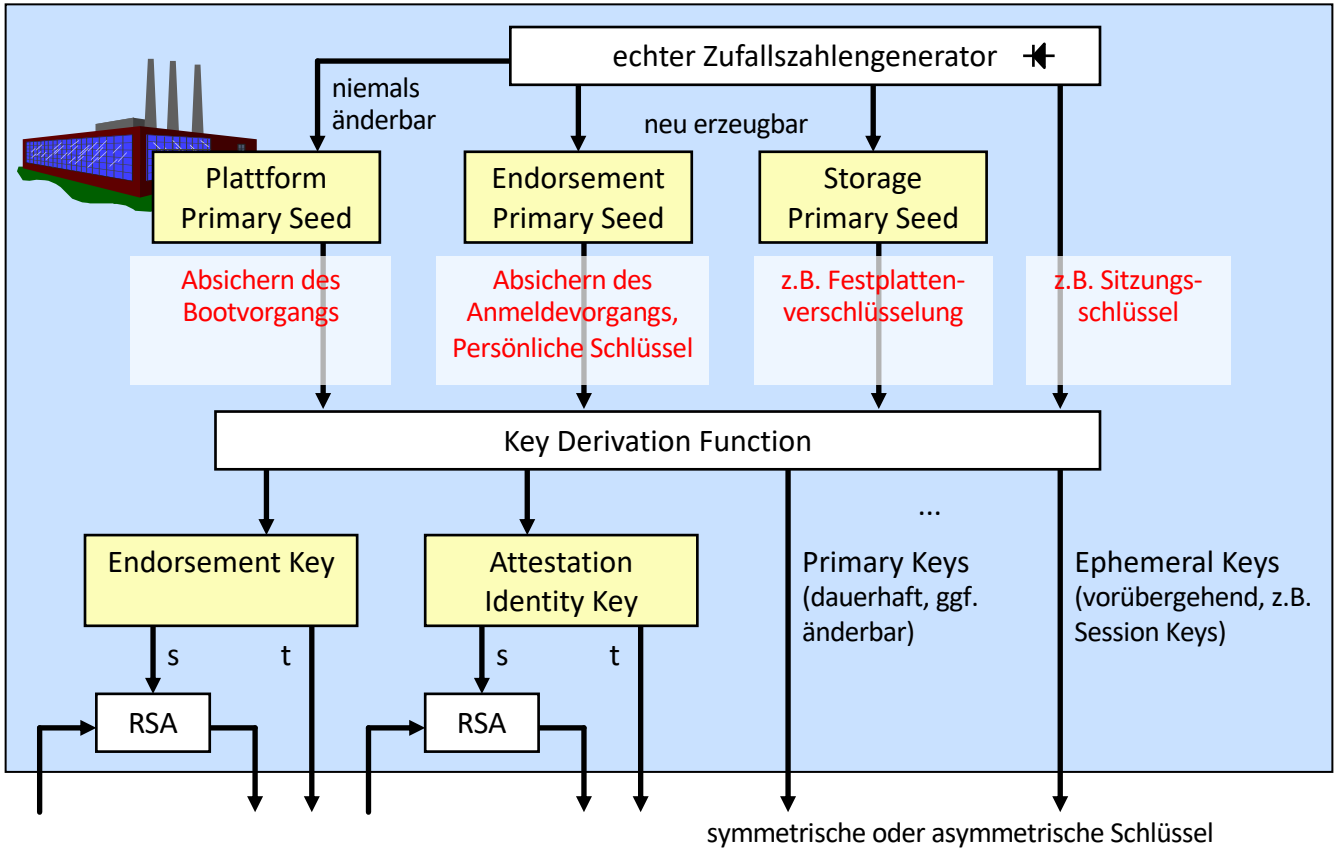


Erzeugung einer Zufallszahl z für die Schlüsselgenerierung:

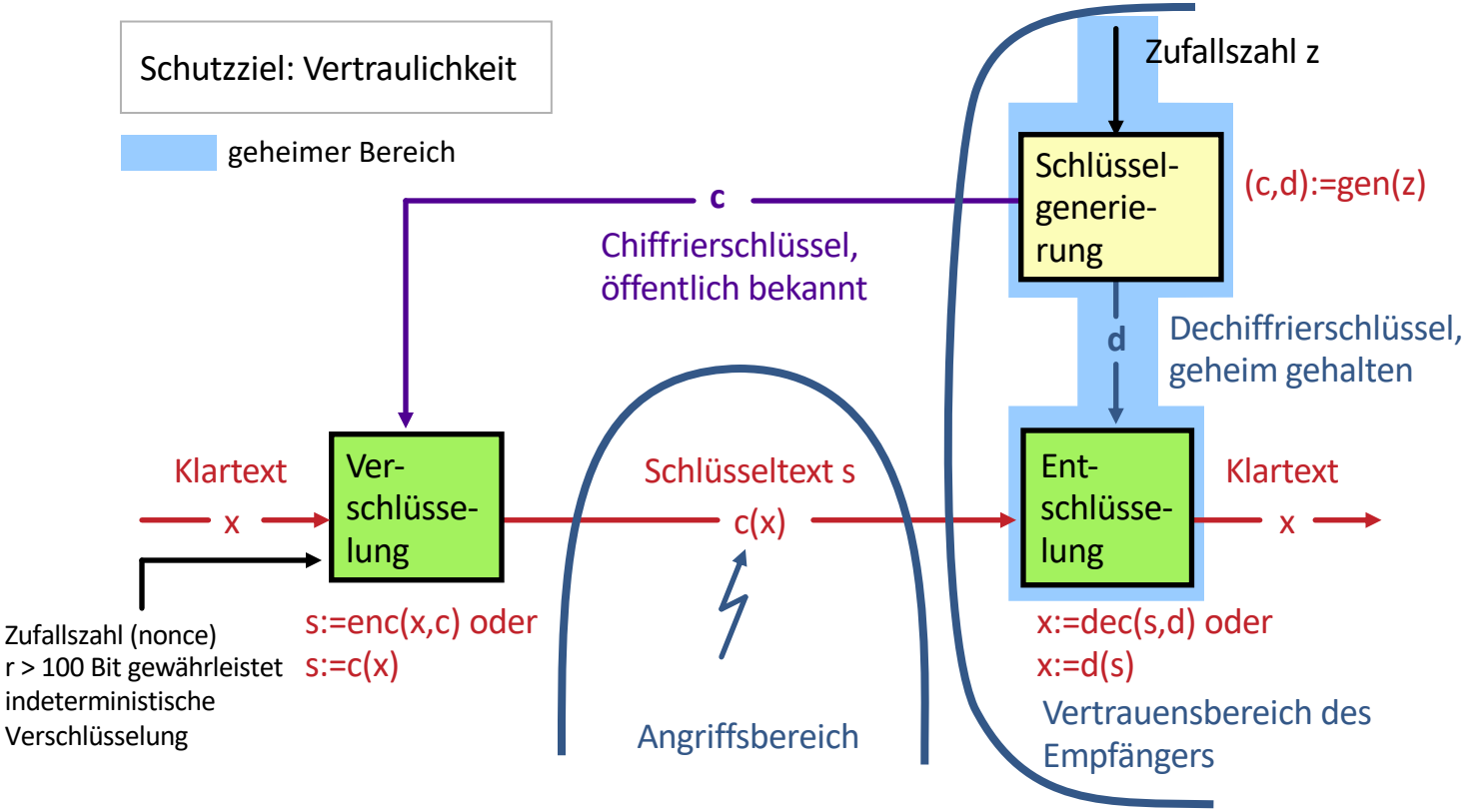
XOR aus

- z_1, einer im Gerät erzeugten,
- z_2, einer vom Hersteller gelieferten,
- z_3, einer vom Benutzer gelieferten,
- z_n, einer aus Zeitabständen errechneten.

Beispiel: Trusted Platform Module (TPM) 2.0



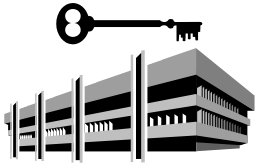
Asymmetrische Verschlüsselung



»Kasten mit Schnappschloss. Es gibt nur einen Schlüssel.«

Zertifizierung des öffentlichen Schlüssels

Zertifizierungsstelle (Certification Authority) CA



1. A lässt ihren öffentlichen Chiffrierschlüssel c_A nach Identitätsprüfung eintragen und erhält ein Schlüsselzertifikat $sig_{CA}(A, c_A)$ zurück.

3. B prüft das Schlüsselzertifikat (Signatur von CA) und fragt ggf. bei der CA, ob der Schlüssel noch gültig ist

2. B besorgt sich den öffentlichen Chiffrierschlüssel von A, konkret: $sig_{CA}(A, c_A)$.

Teilnehmerin A

c_A(Nachricht an A)

4. B verschlüsselt seine Nachricht mit dem öffentlichen Chiffrierschlüssel von A.

Teilnehmer B

Maskerade-Angriff 1/2

Alice hat Schlüsselpaar generiert und will ihn veröffentlichen.

```
Alice <alice@abc.de>
-----BEGIN PGP PUBLIC KEY BLOCK-----
mQGiBDOyJk0RBADVPjcdvwmyOtqsZBt6z4/5M9MYDB
i+dYN[illegible]DEBXQcH/RGe2i30LRvRk4asX++JSTylku
8LMO[illegible]+lbmsVNXeQSdmbSAUfd3d9bI/+fGwQcz
6W8[illegible]xPI7oVZUY1I7cqEfTvic003bgL
sUZ[illegible]ukKj01O66wVmqlnXcbi2XUebka
LQ[illegible]ZW59gf5I0eUBevSmydIaliH9Pm
-----END[illegible]C KEY BLOCK-----
```

c_{Alice}

Angreifer

- hält c_{Alice} zurück (blockiert Verteilung)
- generiert selbst ein Schlüsselpaar c_{Mask}, d_{Mask} unter falschem Namen
- schickt c_{Mask} an Bert

c_{Mask}

Bert besitzt jetzt nicht authentischen Schlüssel von Alice.

```
Alice <alice@abc.de>
-----BEGIN PGP PUBLIC KEY BLOCK-----
OTUAoLncfli6Yit0Kqgp/N9h37uopJHbiQCVAw
xBBPLRdmalP22ij0dARxbJLO7u7XOrnyV3b[illegible]
l4ydps/ruj9yaY62BwQNMEoGjAnZGA5t3MM[illegible]
7ZLp1dmFYYVYPL4xRfOJ+MF5ifb8RXaDAl[illegible]
CwMBAgAKCRDhQCBhSe8dhOYYAJsE[illegible]
u64hbO2wuFQlwwq1yb+JAD8DBRA[illegible]
-----END PGP PUBLIC KEY BLO[illegible]
```

Maskerade-Angriff 2/2

Ohne die Gewissheit über die Echtheit eines öffentlichen Schlüssels funktioniert keine sichere asymmetrische Kryptographie. Deshalb: Schlüsselzertifizierung

Bert will Alice eine Nachricht N schicken.

$c_{Mask}(N)$

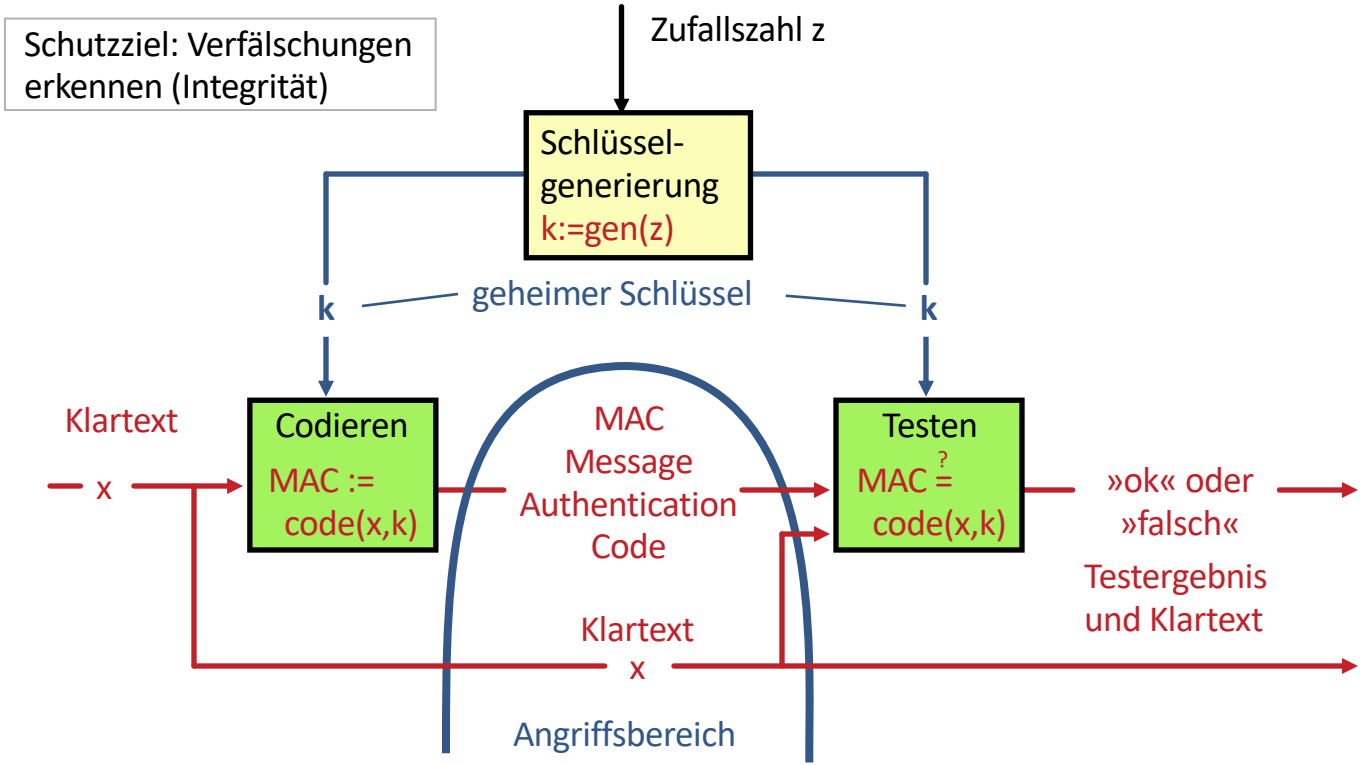
Angreifer:

- Weiterleitung verhindern
- entschlüsseln von $c_{Mask}(N)$ mit d_{Mask}
- verschlüsseln von N mit c_{Alice}

$c_{Alice}(N)$

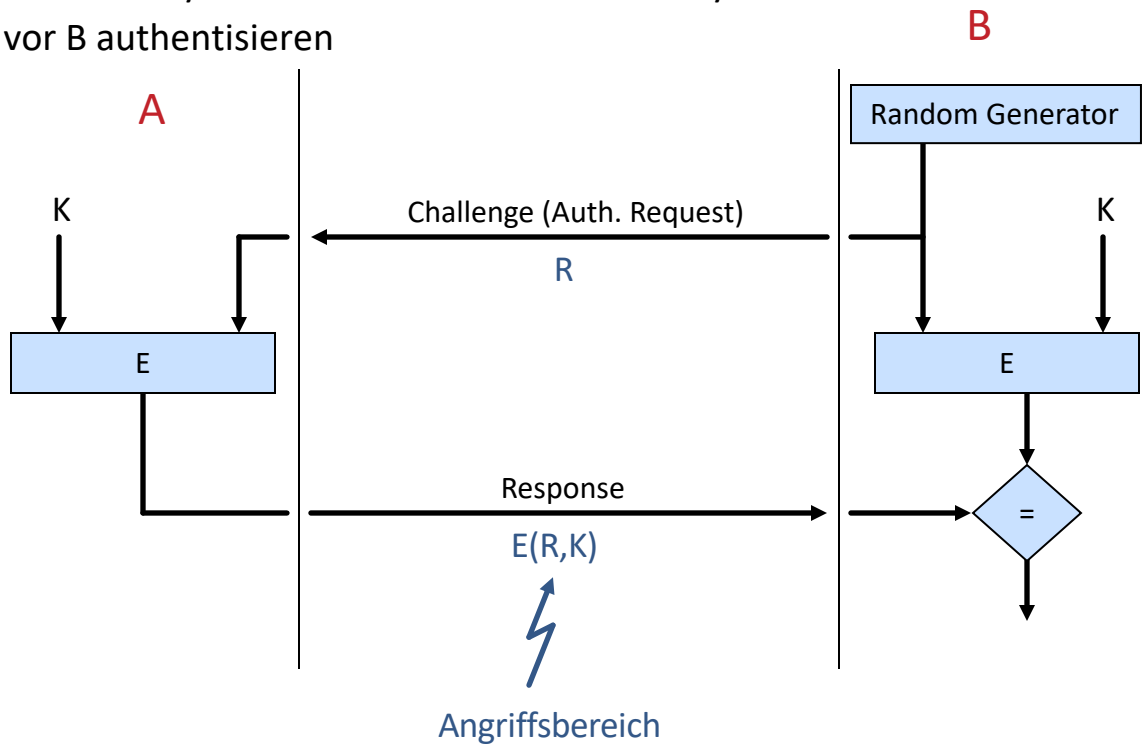
Alice erhält die Nachricht N.
N ist verschlüsselt mit ihrem öffentlichen Schlüssel.

Symmetrische Authentikation

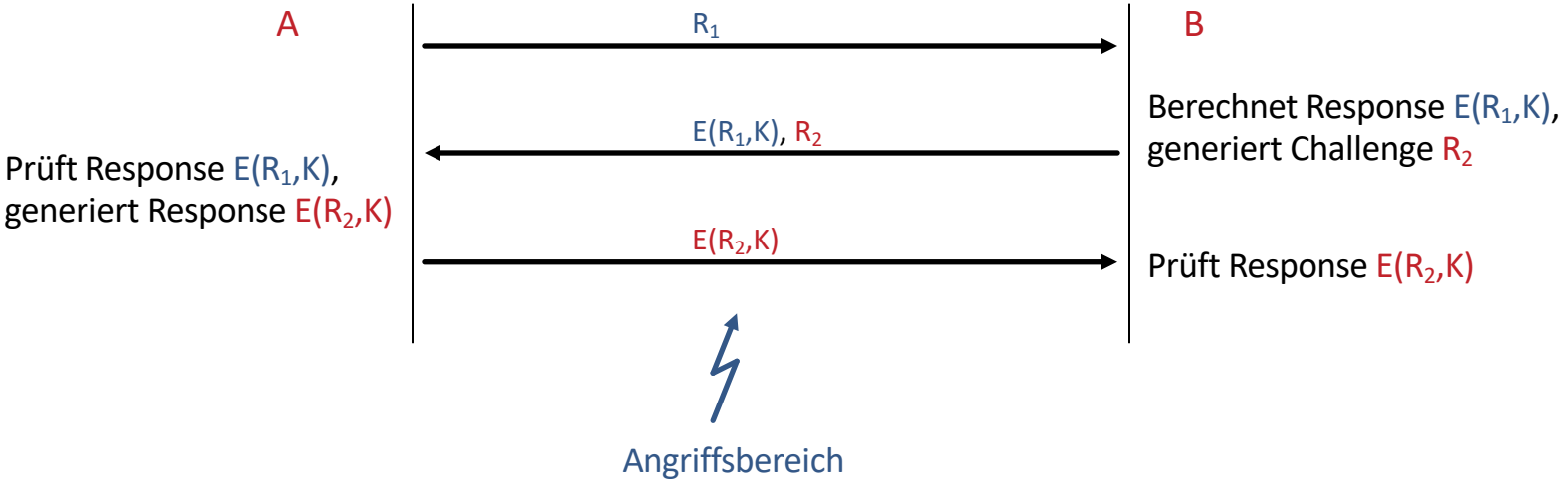


Challenge-Response-Authentikation

- Frage-Antwort-Verfahren
  - meist basierend auf symmetrischem Authentikationssystem
  - A soll sich vor B authentisieren

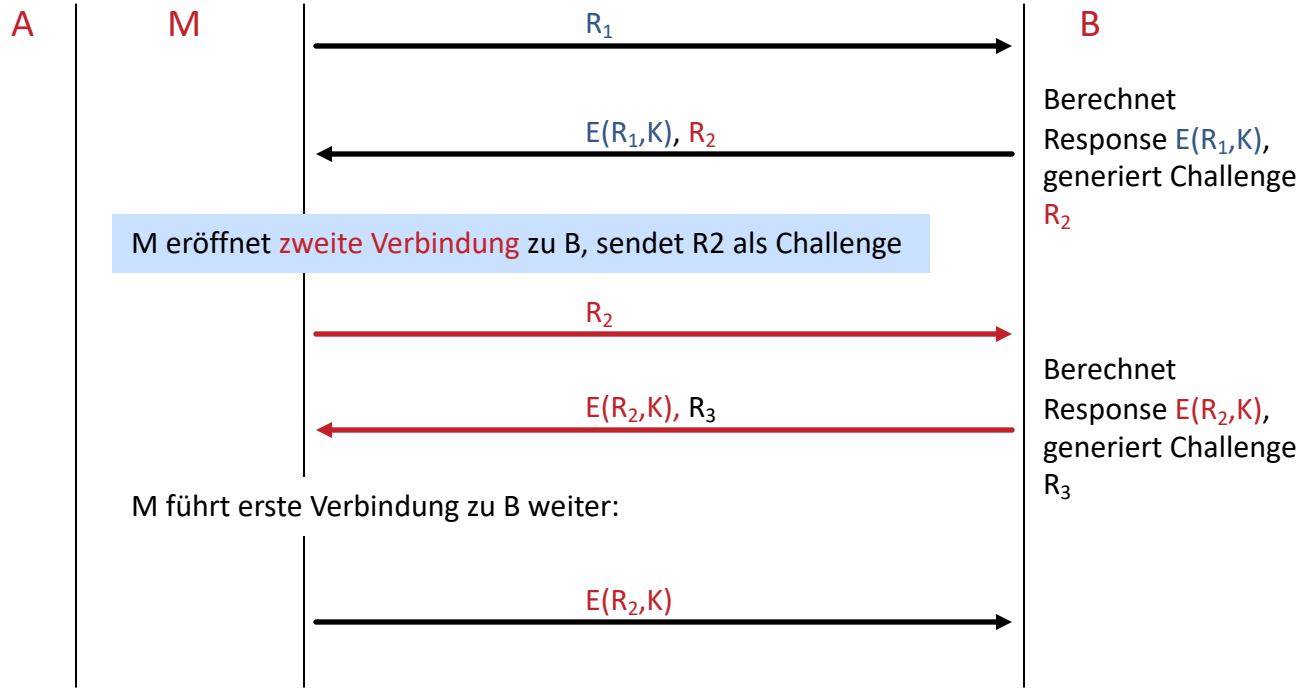


Gegenseitige Authentikation

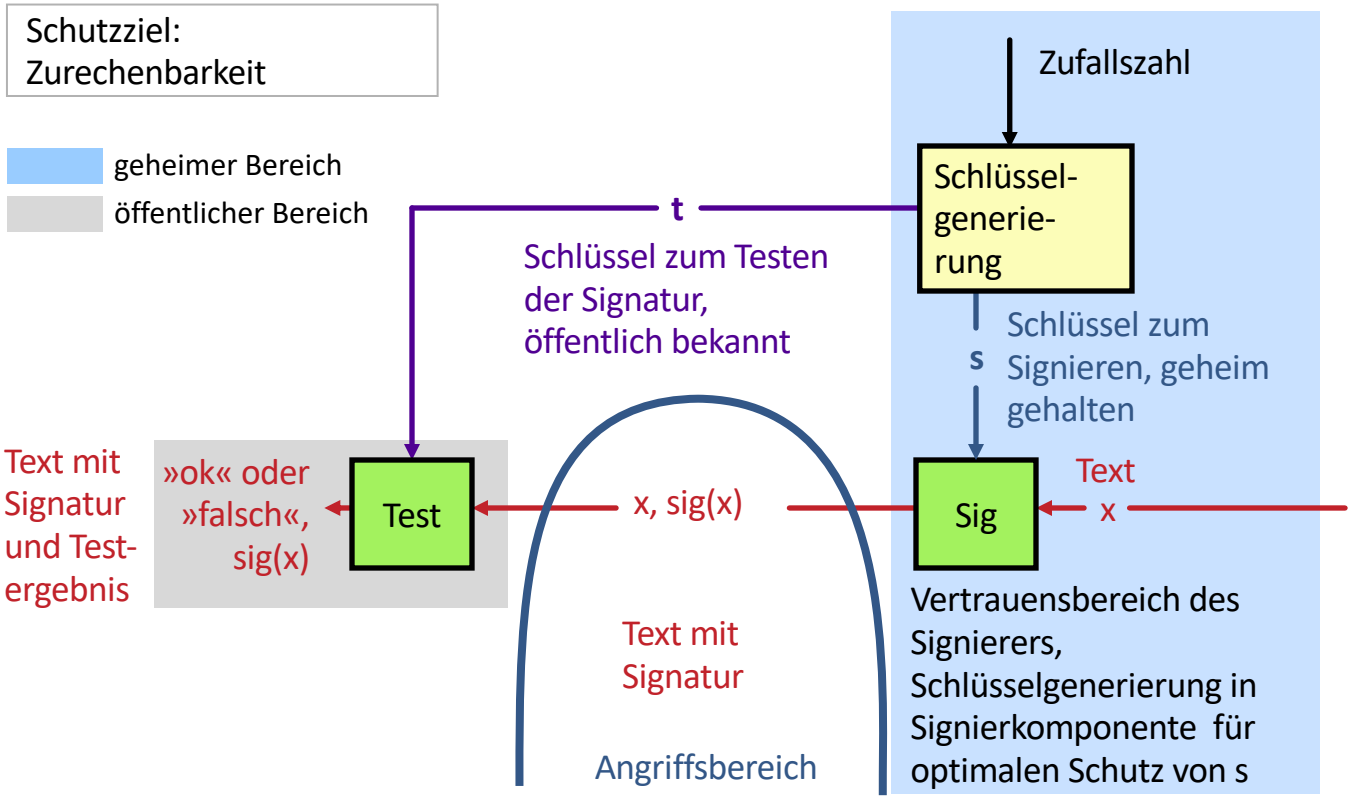


Gegenseitige Authentikation

- Aktiver Angriff auf gegenseitige Authentikation auf der Basis symmetrischer Kryptosysteme
  - Angreifer M maskiert sich als A, kennt K *nicht*

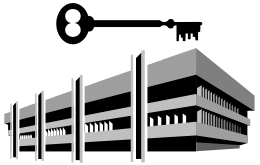


Digitales Signatursystem (asymmetrisches Authentikationssytem)



Zertifizierung des öffentlichen Testschlüssels

Zertifizierungsstelle (Certification Authority) CA



1. A lässt t_A, den Schlüssel zum Testen seiner Signatur, nach Identitätsprüfung eintragen und erhält ein Schlüsselzertifikat $sig_{CA}(A, t_A)$ zurück.

3. B besorgt sich ggf. $sig_{CA}(A, t_A)$, prüft (Signatur von CA) und fragt, ob t_A noch gültig ist.

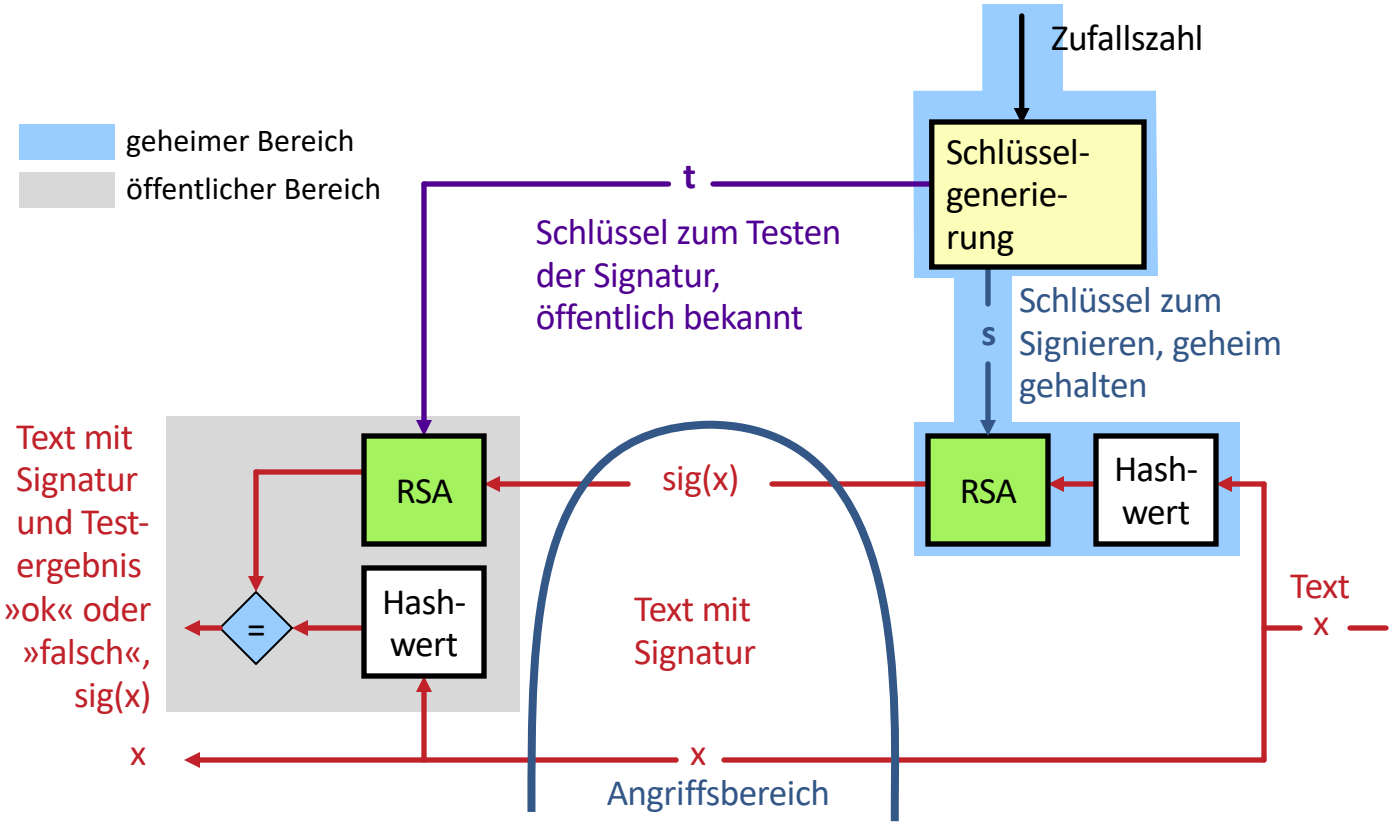
2. A signiert ihre Nachricht N mit t_A und schickt in manchen Systemen das Zertifikat gleich mit.

Teilnehmer B

$N, sig_A(N), sig_{CA}(A, t_A)$

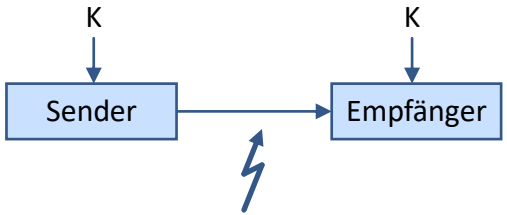
Teilnehmerin A

Digitales Signatursystem mit RSA



Schlüssellängen

- Symmetrische Systeme
  - Vergrößerung des Schlüssels um 1 Bit
    - Verdoppelung des Suchraums
  - Jeder Schlüssel ist aus Sicht des Angreifers gleichwahrscheinlich.



  - Schlüssellängen: 128-256 Bit gelten als auf »absehbare Zeit« sicher.

Beispiel: 56 Bit (DES) sind heute unsicher.

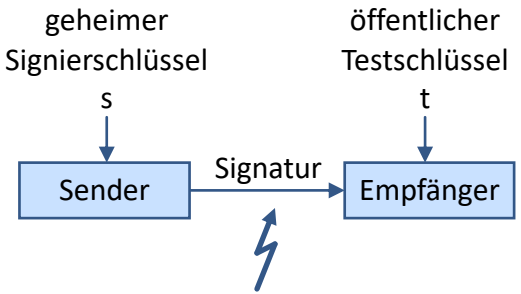
56 Bit Schlüssellänge $\rightarrow$ 2^{56} mögliche Schlüssel (ca. $7 \cdot 10^{16}$)

Ausprobieren eines Schlüssels in angenommen 1 Nanosekunde (10^{-9} s)

Ausprobieren aller Schlüssel: $2^{56} \cdot 10^{-9}$ s = 72057594 s = 2,28 Jahre

Schlüssellängen

- Asymmetrische Systeme
  - meist Vergrößerung des Zahlenbereichs nötig, da nur bestimmte Zahlen (z.B. Primzahlen) Schlüssel sein können
  - Schlüssellängen bei Verfahren auf Basis der Faktorisierungsannahme und des diskreten Logarithmus: 2048-4096 Bit
  - Schlüssellängen bei Verfahren mit elliptischen Kurven: ab 256 Bit



Wieviele Zahlen sind Primzahlen? l: Schlüssellänge

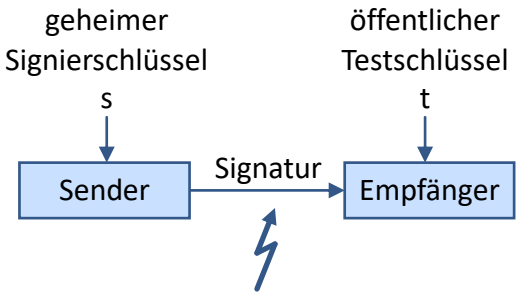
Primzahlsatz: $\pi(2^l) = 2^l / \ln(2^l)$. Mit l=256 ist $\pi(2^{256}) \approx 6{,}5 * 10^{74} \approx 248$ Bit

Tatsächliches Problem bei asymmetrischen Systemen: Existenz subexponentieller Algorithmen für die zahlentheoretisch zugrunde liegenden Annahmen.

  - Beachte: Die nahende Existenz des Quantencomputers ändert die Lage dramatisch!

Schlüssellängen asymmetrische Kryptosysteme

- Asymmetrische Systeme
  - Sender und Empfänger verwenden unterschiedliche Schlüssel.



- Beispiel RSA (vereinfacht):
  - Signierschlüssel s wird aus geheimen 500-stelligen Primzahlen p und q gebildet.
  - Testschlüssel t wird als Produkt von p mal q berechnet und veröffentlicht.

- Faktorisierungsproblem (Beispiel):
  - Zerlege 35 in p und q.
  - Zerlege 391.
  - Zerlege 523 619.
  - Zerlege 8 093 145 481.

Faktorisierungsproblem

Die Länge der nachfolgenden Dezimalzahl beträgt 2048 Bit
bzw. 616 Dezimalstellen.

Sicher wären
heute 4096 Bit.

1331402728893351929221084092606621744763038316523836716885470094842532359405869171404826691822563682852609928294472079801831701748676203589522309699864475593305834924296366272986403385965318945565460131131543468232122717489278596479945345861335532180229838481084214654420899190906105423447682944817251037572224219171159710630268065871412875870372651506536690943231166865745365588665916473610533110465160130696690368667341265580177443937511616112191957695784885598829023972483090339116614750058546968200210690725022485333287548326986162384052213812521451374399190908000859552743893827218449566611138745095472005761807

Zerlege sie in ihre Primfaktoren p und q.

Quelle: http://library.open.oregonstate.edu/cryptography/chapter/chapter-13-the-rsa-function/

Prognostizierte Sicherheit

- These: In >= 10 Jahren sind Public Key Kryptoverfahren mit 1024-Bit-Schlüsseln unsicher.

Sicher wären heute 4096 Bit.

- Warum?
  - Der Quantencomputer ist in der Lage, bestimmte Klassen von Algorithmen weitaus effizienter zu berechnen, als dies auf dem herkömmlichen Supercomputer möglich ist.

Quantencomputer rechnen mit Qubits oder Quantenbits:

Quantenphysikalisches Systen, das neben den zwei diskreten Zuständen 0 und 1 noch unendliche viele weitere Zustände der Superposition kennt

- Effizient lösbar z.B.
  - Suche in unsortierten Daten
  - Faktorisieren großer Zahlen
- Auswirkungen
  - Heutige asymmetrische Kryptosysteme werden mit der Existenz eines entsprechend leistungsfähigen Quantencomputers unsicher sein.

Quantencomputer

- Bestandteile
  - mehrere verschränkte Qubits bilden ein Quantengatter (Register)

    Zwei Teilchen sind verschränkt, wenn sie sich einen gemeinsamen Zustand teilen — Quantenmechanische Messung legt konkreten Teilchenzustand fest.

  - Quantenoperatoren arbeiten auf diesen Registern

- Quantenoperatoren
  - erzeugen beim Messen auf allen Qubits zugleich das Berechnungsergebnis
  - hochparallele gleichzeitige Ausführung der Operation

Bild: IBM Research

Quantencomputer

- Anwendungsbeispiele
  - Simulation chemischer Prozesse
  - Optimierungsprobleme
  - Medizinische Forschung
  - Künstliche Intelligenz (Machine Learning)
- Aufbau aus supraleitenden Schaltkreisen und Mikrowellentechnik

arbeitet bei Temperaturen nahe des absoluten Nullpunkts

16 Qubit auf IBM-Chip, eingefasst in supraleitende Mikrowellenresonatoren

Bild: IBM Research

Quantencomputer

- Problem: Quantencomputer ist erst dann für die effizient lösbaren Probleme auch praktisch effizient, wenn
  - Registerbreite ausreichende Größenordnung (hier: Schlüssellänge) erreicht
  - Fehlerrate drastisch gesenkt werden kann
    - Pro logischem Qubit werden Hunderte bis Tausende physische Qubits benötigt.

- Annahme des BSI:
  - Bei Registerbreite von 1 Mio. physischer Qubits und einer Fehlerrate von 1/10.000 kann der Quantencomputer einen RSA-2048 in 100 Tagen knacken.
  - Heutige Systeme:
    100 Qubits bei Fehlerrate von >1/100

Sicher wären heute 4096 Bit.

Bild: IBM Research

Quantencomputer

13314027288933519292210840926066217447630383165238367168854700948425323594058691714048266918225636828526099282944720798018317017486762035895223096998644755933058349242963662729864033859653189455654601311315434682321227174892785964799453458613355321802298384810842146544208991909061054234476829448172510375722242191711597106302680658714128758703726515065366909432311668657453655886659164736105331104651601306966903686673412655801774439375116161121919576957848855988829023972483090339116614750058546968200210690725022485333287548326986162384052213812521451374399190908000859552743893827218449566611138745095472005761807

2048 Bit

Bild: IBM Research

- Annahme des BSI:
  - Bei Registerbreite von 1 Mio. physischer Qubits und einer Fehlerrate von 1/10.000 kann der Quantencomputer einen RSA-2048 in 100 Tagen knacken.
  - Heutige Systeme:
    100 Qubits bei Fehlerrate von >1/100

Sicher wären heute 4096 Bit.

2018

spectrum.ieee.org

IEEE SPECTRUM

Follow on:

Topics | Reports | Blogs | Multimedia | Magazine | Resources | Search

Tech Talk | Computing | Hardware

9 Jan 2018 | 0:00 GMT

CES 2018: Intel's 49-Qubit Chip Shoots for Quantum Supremacy

Intel's new superconducting quantum chip, called Tangle Lake, has enough qubits to make things very interesting from a scientific standpoint

By **Jeremy Hsu**

https://spectrum.ieee.org/tech-talk/computing/hardware/intels-49qubit-chip-aims-for-quantum-supremacy

2018

computerbase.de

Computer Base | Ticker | Tests | Themen | Downloads | Preisvergleich | Forum | Anmelden

COMPUTERBASE › CPU

Quantencomputer: Der erste 72-Qubit-Chip kommt von Google

7.3.2018 8:16 Uhr | Volker Rißka

58 Kommentare

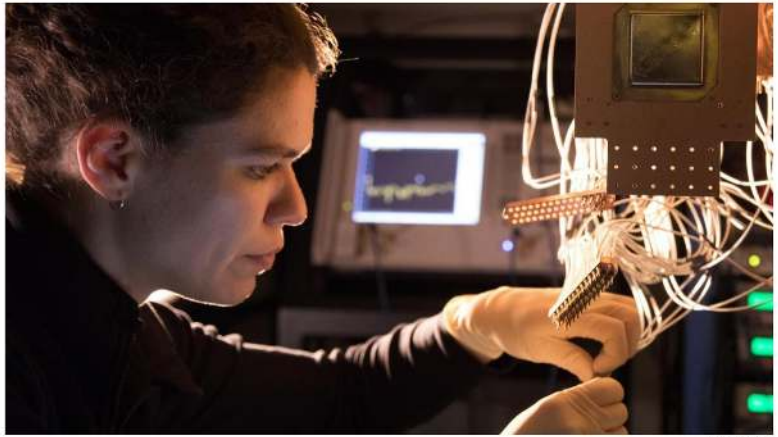


Bild: Google

Mit einem 72-Qubit-Chip hat Google die Messlatte in der Forschung im Bereich Quantencomputer auf ein neues Level gehoben. Der komplexe Chip soll sich laut Forschern trotzdem so verhalten wie bisherige 9-Qubit-Chips, die relativ leicht zu beherrschen sind und deshalb von vielen Forschungsgruppen genutzt werden.

https://www.computerbase.de/2018-03/quantencomputer-72-qubit-chip-google/

Ziel: Universeller Quantencomputer mit >1000 Qubits

Jahr	vor 2016	2017	2018	2021	2026	2031	20xx
Qubits	<10	17	49, 72	100	(1:7)	(1:2)	>1000

Intel, Google

Bild: Intel 49-Qubit-Chip

Wahrscheinlichkeit eines Quantencomputers, der Kryptoverfahren brechen kann; Quelle: Michele Mosca, U Waterloo, aus: ct 1 (2019) 100

Projekt OpenSuperQ will OpenSource-Quantencomputer-Betriebssystem und 100-Qubit-Rechner bis 2021 entwickeln

Prognose: erster »Post-Quantum-Krypto-relevanter« Quantencomputer in 2030

Welche Schlüssellängen und Kryptoalgorithmen sind sicher?

Jährlicher Algorithmenkatalog nach § 17 (1) SigG des Bundesamts für die Sicherheit in der Informationstechnik (BSI)

Tabelle 2: Geeignete Schlüssellängen für DSA

Parameter \ Zeitraum	bis Ende **2022**	bis Ende **2023**
p	2048	3000
q	250	250

Tabelle 1: Geeignete Schlüssellängen für RSA-Verfahren

Parameter \ Zeitraum	bis Ende **2022**	bis Ende 2023
n	1976 (Mindestwert) 2048 (Empfehlung)	3000

Entwurf vom 15.11.2016

Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen

Bekanntmachung zur elektronischen Signatur nach dem Signaturgesetz und der Signaturverordnung

(Übersicht über geeignete Algorithmen)

Vom x.y.2016

Die Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen als zuständige Behörde gemäß § 3 Signaturgesetz (SigG) vom 16. Mai 2001 (BGBl. I S. 876), zuletzt geändert durch Artikel 4 des Gesetzes vom 17. Juli 2009 (BGBl. I S. 2091), veröffentlicht gemäß Anlage 1 Abschnitt 1 Nr. 2 Signaturverordnung (SigV) vom 16. November 2001 (BGBl. I S. 3074), zuletzt geändert durch die Verordnung vom 15. November 2010 (BGBl. I S. 1542), im Bundesanzeiger eine Übersicht über die Algorithmen und zugehörigen Parameter, die zur Erzeugung von Signaturschlüsseln, zum Hashen zu signierender Daten oder zur Erzeugung und Prüfung qualifizierter elektronischer Signaturen als geeignet anzusehen sind, sowie den Zeitpunkt, bis zu dem die Eignung jeweils gilt.

Geeignete Algorithmen zur Erfüllung der Anforderungen nach § 17 Abs. 1 bis 3 SigG vom 16. Mai 2001 in Verbindung mit Anlage 1 Abschnitt I Nr. 2 SigV vom 16. November 2001

Vorbemerkung: Wie in den Vorjahren werden im Folgenden geeignete Algorithmen und Schlüssellängen für den Zeitraum der kommenden sieben Jahre anstatt des in der SigV vorgesehenen Mindestzeitraums von sechs Jahren aufgeführt. Das heißt konkret, dass geeignete Algorithmen und Schlüssellängen bis Ende 2023 statt bis Ende 2022 aufgeführt sind. Im Allgemeinen sind solche längerfristigen Prognosen schwer möglich. Die vorliegende Übersicht über geeignete Algorithmen unterscheidet sich von der zuletzt veröffentlichten Übersicht vom15. Dezember 2014 (BAnz AT 30.01.2015 B3) im Wesentlichen in folgenden Punkten:

1. Die Eignungsdauern der RSA- und DSA-Verfahren werden für eine Schlüssellänge kleiner als 3000 Bit nicht über das Jahr 2022 hinaus verlängert. Ab dem Jahr 2023 wird daher für diese Verfahren eine Schlüssellänge von mindestens 3000 Bit verpflichtend. Hiermit wird in diesen Verfahren die Erreichung eines Sicherheitsniveaus von etwa 120 Bit ermöglicht. Die Durchsetzung dieses Sicherheitsniveaus wird künftig grundsätzlich im Anwendungsbereich der vorliegenden Bekanntmachung angestrebt. Hiermit wird einerseits ein angemessenes Sicherheitspolster gegenüber Fortschritten in der Kryptoanalyse und in der Rechentechnik geschaffen, wie es für eine Fortschreibung der in den letzten Jahren üblichen 7-Jahres-Prognosen benötigt wird. Andererseits wird hiermit ein ähnliches Sicherheitsniveau in allen als geeignet angesehenen Signaturverfahren erreicht; in den Vorjahren war hier der Sicherheitsspielraum der auf elliptischen Kurven basierenden Verfahren etwas größer als bei den Verfahren, die auf dem RSA-Problem oder dem Problem der Berechnung diskreter Logarithmen in endlichen Körpern beruhen. Zudem erfolgt auf diesen

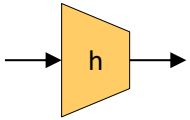
https://www.bsi.bund.de/Algorithmenkatalog

Vollständiges Durchsuchen (brute-force, exhaustive search)

- Angriff über Supercomputer und künftig Quantencomputer
  - betrifft nur komplexitätstheoretisch sichere Systeme
- Schutz gegen Supercomputer
  - Schlüssel ausreichend lang wählen
- Schutz gegen Quantencomputer
  - symmetrisch: Schlüssellänge verdoppeln auf mind. 256 Bit
  - asymmetrisch: Hoffen auf Post-Quantum Kryptographie

	Key lengths	Complexity		
		Super Computer	Quantum Computer	nach: Bernstein, Buchmann, Dahmen: Post Quantum Cryptography. Springer, 2009
Symm.	128 Bit	2^{127}	2^{64}	Grover, 1996
	256 Bit	2^{255}	2^{128}	
Asymm.	1024 Bit	$\approx 2^{90}$	$\approx 2^{25}$	Shor, 1994
	2048 Bit	$\approx 2^{117}$	$\approx 2^{28}$	

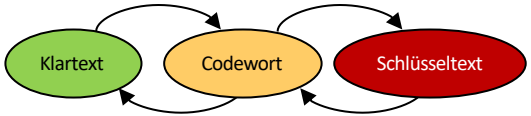
- Hash-basierte Verfahren
  - XMSS (Extended Merkle Signature Scheme)
  - standardisiert im RFC 8391
  - Anwendung für digitale Signatur



  - Problem
    - jeder Schlüssel nur einmal verwendbar oder Nutzer muss sich Statusinfo innerhalb einer Baumstruktur speichern (stateful)
  - weiteres Verfahren:
    - SHPINCS (stateless)
  - $|sig| >> |sig|_{RSA}$

Post-Quantum Kryptographie: asymmetrisch und resistent gegen Angriffe mittels Quantencomputer

Post-Quantum Kryptographie



- Code-basierte Algorithmen
  - Basis: McElice (1978)
  - Anwendung für asymmetrische Verschlüsselung
  - ursprünglich entwickelt für Fehlerkorrektur

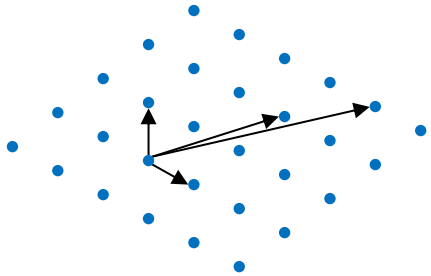
  - Idee
    - Nachricht wird in Codewort umgewandelt, vom Sender gezielt verändert
    - Empfänger kann bei Kenntnis des geheimen Schlüssels Fehler korrigieren

  - $|key| > 1000\ |key|_{RSA}$

Post-Quantum Kryptographie: asymmetrisch und resistent gegen Angriffe mittels Quantencomputer

Post-Quantum Kryptographie

- Gitter-basierte Algorithmen (Lattice-Verfahren)
  - inzwischen recht ausgereift
  - kleine Schlüssellängen
  - universell einsetzbar
    - digitale Signaturen
    - asymm. Verschlüsselung
    - asymm. Schlüsselvereinbarung
  - Algorithmus New Hope
    - testweise eingesetzt in Google Crome



Post-Quantum Kryptographie: asymmetrisch und resistent gegen Angriffe mittels Quantencomputer

Post-Quantum Kryptographie

- Algorithmen auf Basis von multivariaten Polynomen
  - einsetzbar für digitale Signaturen und asymm. Verschlüsselung
  - zahlreiche Vorschläge bereits gebrochen
  - $|key| >> |key|_{RSA}$

- Isogenie-basierte Algorithmen
  - nutzt nicht Werte einer elliptischen Kurve, sondern Zuordnungen von Kurven (maps)
  - sehr neu, bisher wenige Untersuchungen
  - gut geeignet für asymm. Schlüsselvereinbarung

- Algorithmen auf Basis nichtlinearer Gleichungssysteme

Post-Quantum Kryptographie: asymmetrisch und resistent gegen Angriffe mittels Quantencomputer

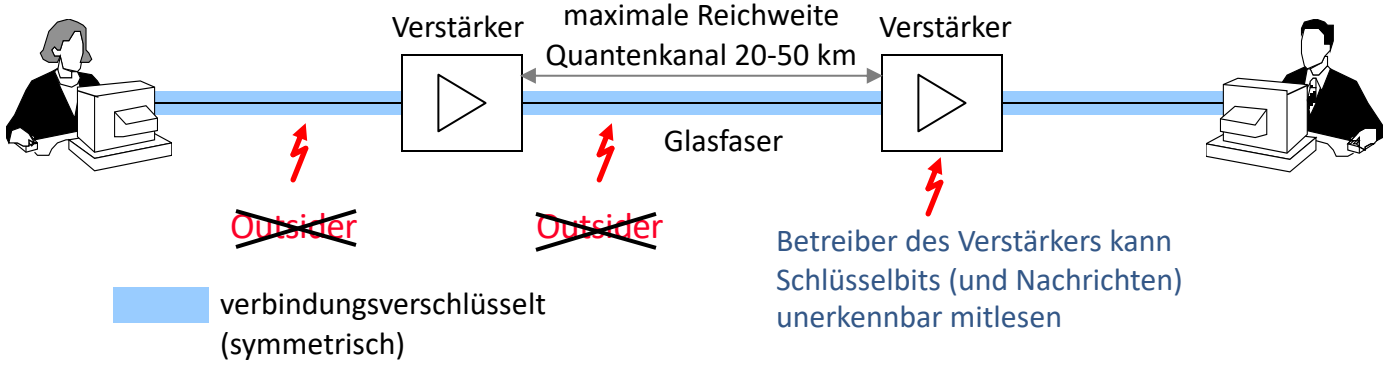
Post-Quantum Kryptographie

- NIST Challenge zur Post-Quantum Kryptographie 2016
  - 1. Runde
    - Dezember 2017: 69 Vorschläge
  - 2. Runde
    - 2018 bis 2020
  - 3. Runde
    - 2020 bis 2022 (Finalisten)
  - Standardisierung
    - im Jahr 2024

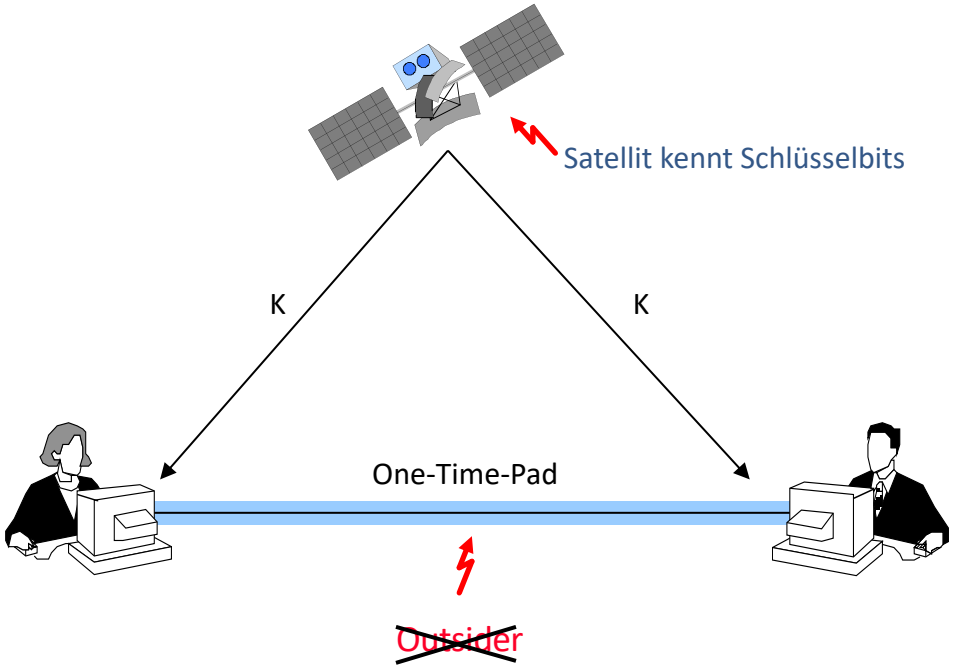
Prognose:
erster »Post-Quantum-Krypto-relevanter« Quantencomputer in 2030

Quantenkryptographie

- Austausch von symmetrischen Schlüsselbits
  - Übertragung der Schlüsselbits meist über eine Glasfaser
  - Verschlüsselung mit informationstheoretisch sicherem One-Time-Pad
- Idee der Quantenkryptographie
  - Wenn ein Schlüsselbit während der Übertragung vom Angreifer mitgelesen wird, verändert sich sein (Energie)-Zustand erkennbar.
  - Es werden zur Verschlüsselung nur Schlüsselbits verwendet, deren (Energie)-Zustand bei Ankunft beim Empfänger erkennbar unverändert ist.



Satellitengestützte Quantenkommunikation

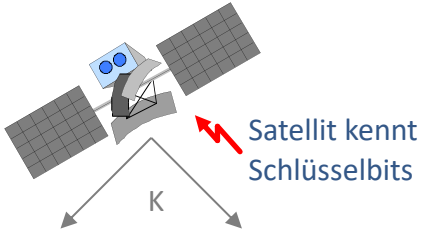


Ende-zu-Ende verschlüsselt
(symmetrisch)

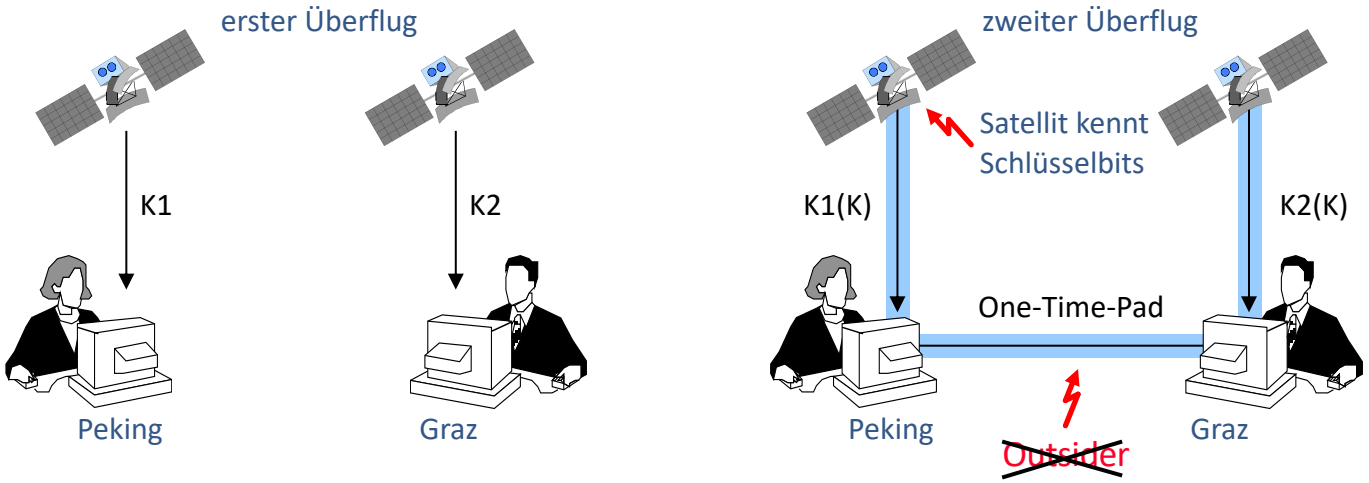
Satellitengestützte Quantenkommunikation

- Prinzip
  - Satellit sendet zwei verschränkte Lichtteilchen an zwei Empfänger
  - Verschränkungseigenschaft: Polarisation des Lichts
  - Wenn einer der Kanäle abgehört wird (entspricht einer Messung), wird die Polarisation des Teilchens bestimmt
    - Messung erkennbar (No-Cloning-Theorem)
    - Schlüsselbit ignorieren
  - Wenn nicht, besitzen beide Empfänger nun ein nachweisbar nicht abgehörtes Schlüsselbit

- Satellit Micius (China)
  - ca. 5 Jahre Entwicklungszeit
  - Umlaufbahn in ca. 500 km Höhe
  - sendet ca. 6 Mio. verschränkte Photonen pro Sekunde



Überwindung größerer Entfernungen mit Satelliten

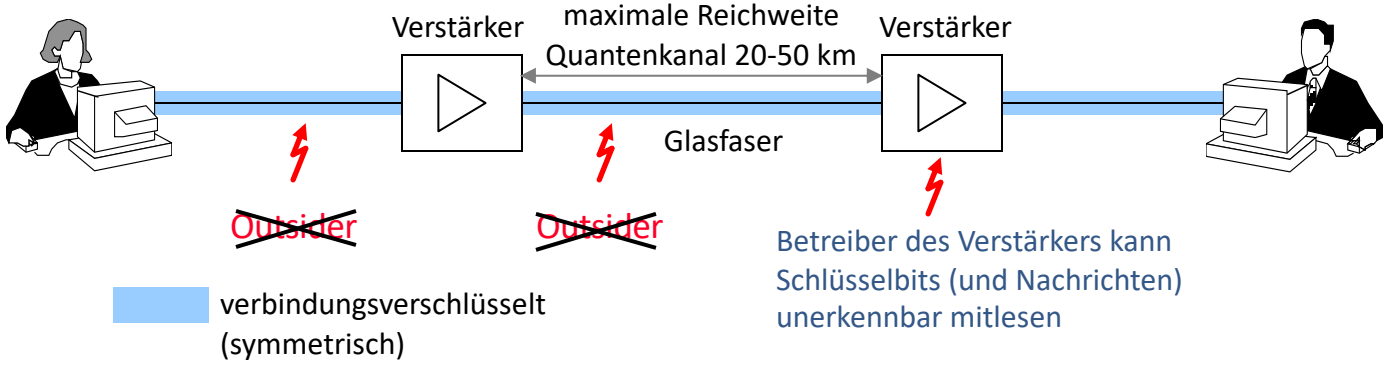


- Satellit dient als Relaisstation für den Schlüsselaustausch
  - praktisch erprobt zwischen Peking und Graz (7600 km)
  - erster Überflug: paarweiser Schlüsselaustausch zwischen Satellit und Peking (K1) bzw. Satellit und Graz (K2)
  - zweiter Überflug: K wird verschlüsselt an Peking und Graz übermittelt
  - 5 Satelliten sollen Flächendeckung für China erreichen

Quantenkryptographie

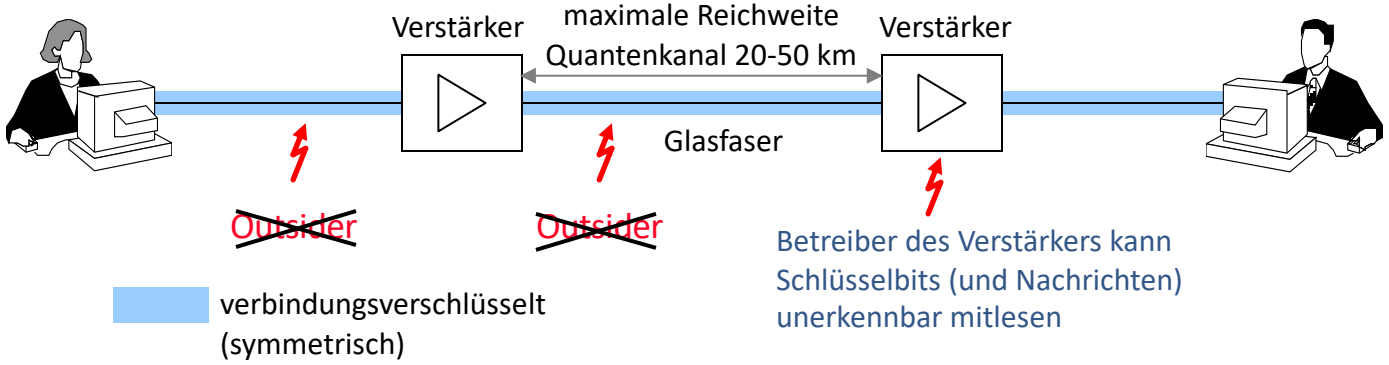
- Beschränkte Entfernungen überwindbar
  - Einsatz von Relaisstationen (Verstärker)
  - entspricht Man-in-the-Middle-Angriff

- Verfügbarkeits- und Erreichbarkeitsproblem der Quantenkommunikation
  - Aktiver Angreifer kann jegliche vertrauliche Kommunikation unterdrücken
  - Fehleranfälligkeit der Quantenübertragung



Glasfaserbasierte Quantenkryptographie

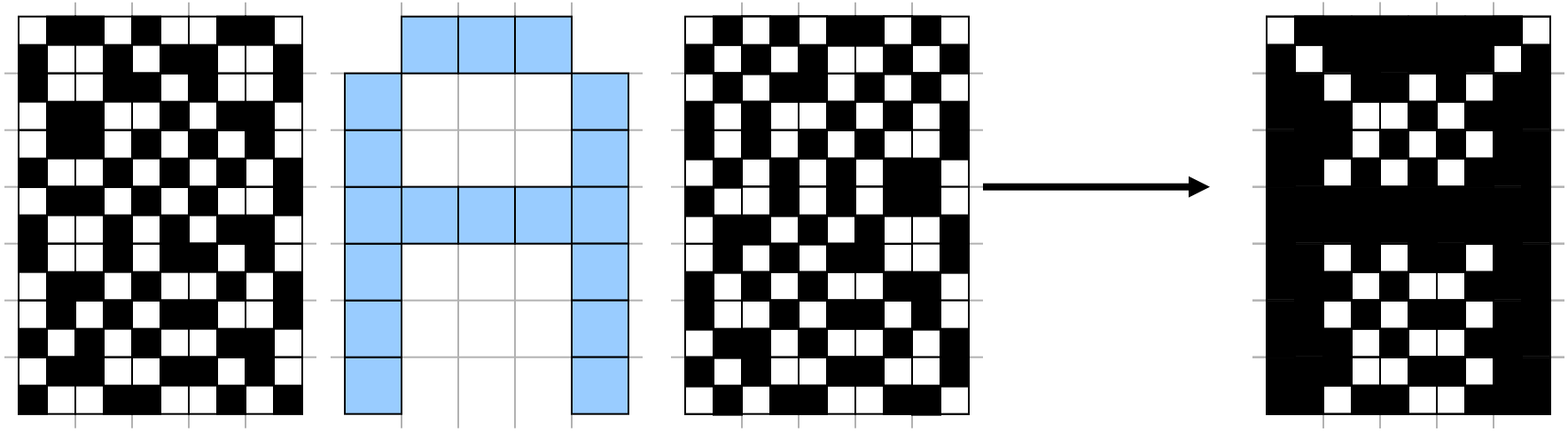
- Signaldämpfung
  - pro Kommunikationsabschnitt nur 20-50 km (100-200 km) überbrückbar
- China: 1900 km mit 32 Relaisstationen
  - Verbindungsverschlüsselung zwischen den 32 Relaisstationen
- Aktuelle Projekte:
  - Quantum Internet Alliance will Quantenrepeater entwickeln
  - Projekt UNIQORN will Miniaturisierung der Quantenkommunikationsbausteine erreichen



Kryptographie: Anwendungsbeispiele

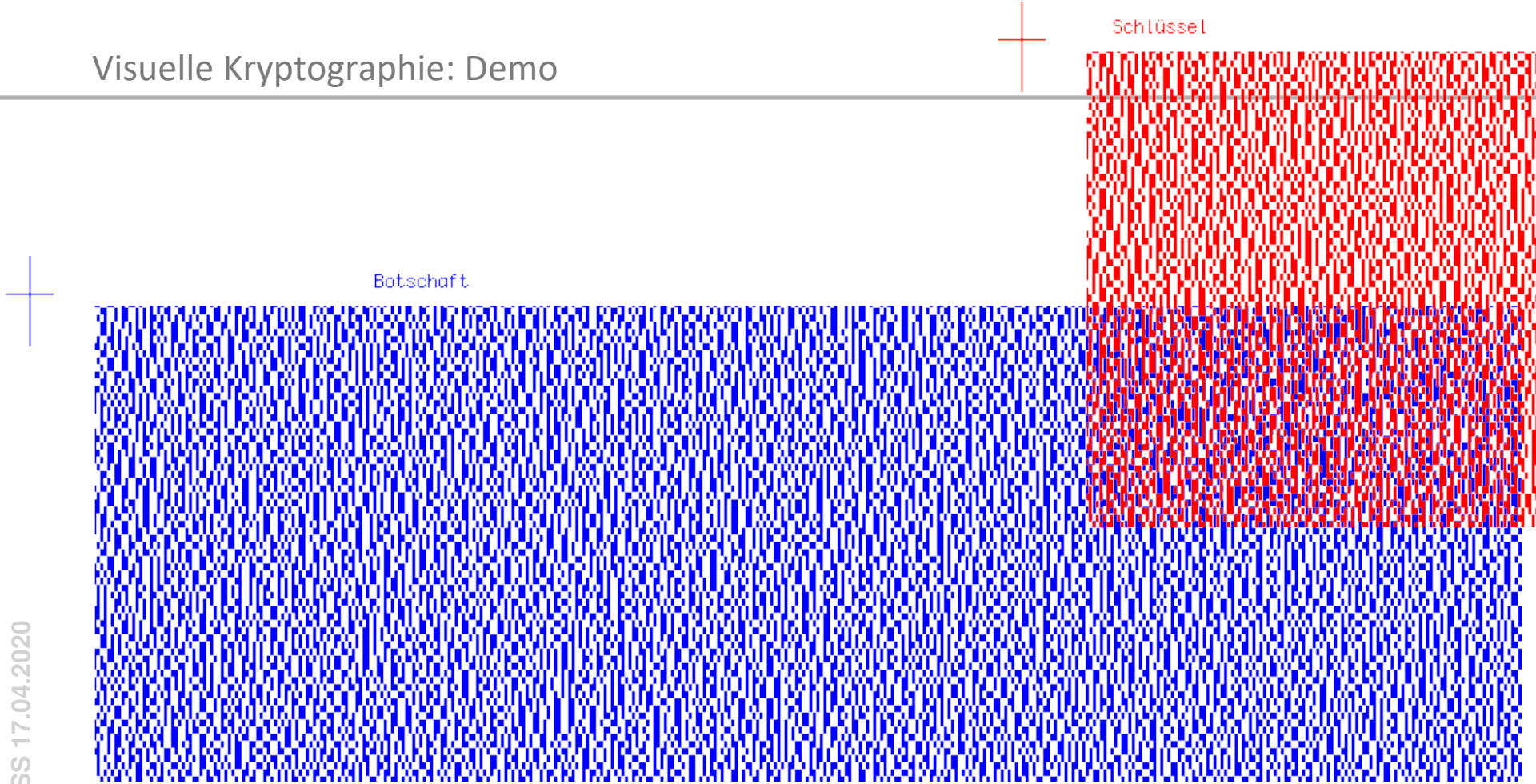
Visuelle Kryptographie

- Symmetrisches Verfahren: Sender und Empfänger erzeugen sich Zufallsmuster aus zwei komplementären Basismustern: Zufallsmuster nur für genau eine Botschaft verwenden!
- Visuelle Botschaft:
  - Sender verwendet negiertes Muster für schwarze Bildpunkte
  - Für »weiße« Bildpunkte: keine Veränderung



Schlüssel Klartext Schlüsseltext Empfänger

Visuelle Kryptographie: Demo



Vergleich: symmetrische-asymmetrische Systeme

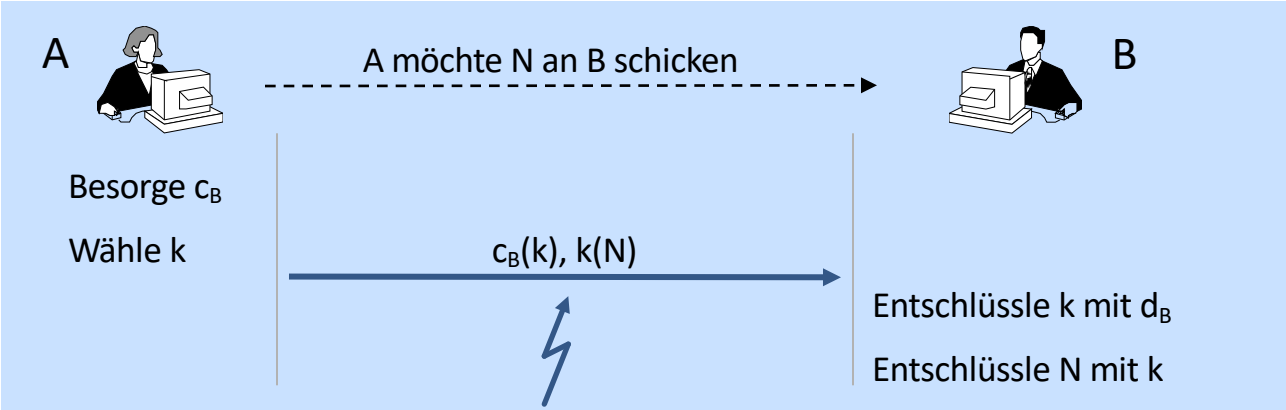
- Wieviele Schlüssel müssen bei *n* Teilnehmern ausgetauscht werden?
  - symmetrische Systeme:
  - asymmetrische Systeme:

- Typische Schlüssellängen: (bei vergleichbarem Sicherheitsniveau)
  - symmetrische Systeme: 128–256 Bit
  - asymmetrische Systeme: 2048–4096 Bit
    Elliptische Kurven: ca. 250 Bit
- Performance:
  - symmetrische Systeme ver- bzw. entschlüsseln etwa um den Faktor 100–10.000 schneller

- Asymmetrische Systeme: Geringere Effizienz und größere Schlüssellängen werden aufgewogen durch den stark vereinfachten Schlüsselaustausch

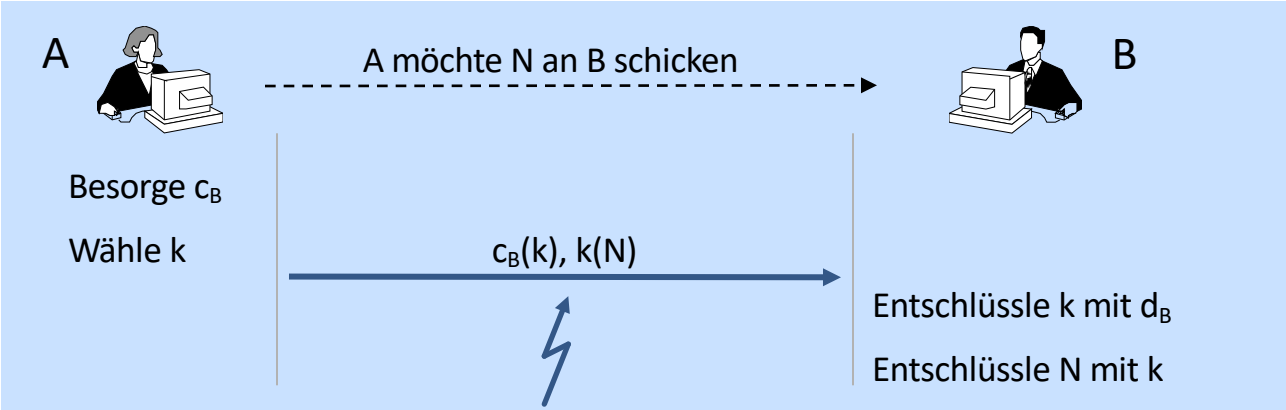
Hybride Verschlüsselung

- Kombination von asymmetrischer und symmetrischer Verschlüsselung
  - Erzeugen eines symmetrischen Sitzungsschlüssels
  - Nachrichten selbst werden symmetrisch verschlüsselt
  - Nachrichtenschlüssel werden asymmetrisch verschlüsselt



Hybride Verschlüsselung

- Problem der unten stehenden Lösung hybrider Verschlüsselung: Fehlende Forward Secrecy
  - Forward Secrecy bedeutet: verwendeter Sitzungsschlüssel k lässt sich nach Beendigung der Sitzung nicht mehr aus dem verwendeten Langzeitschlüssel c_B rekonstruieren.
  - Beispiel für Verfahren mit Forward Secrecy: interaktiver Diffie-Hellman-Schlüsselaustausch

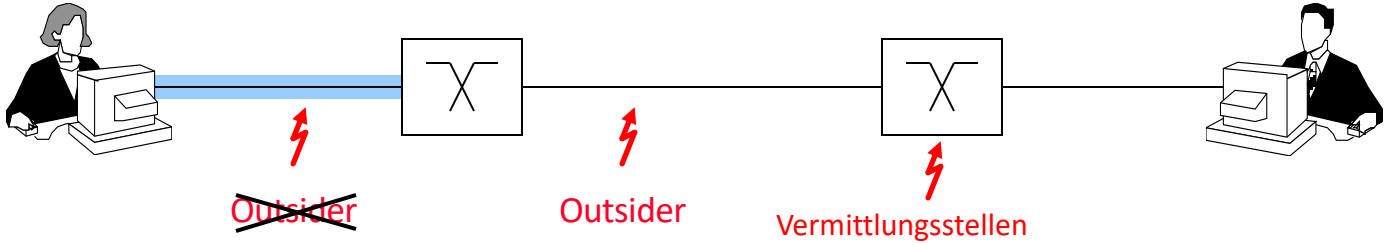


Beobachter zeichnet gesamte verschlüsselte Kommunikation in der Hoffnung auf, später einmal d_B in Erfahrung zu bringen, um *rückwirkend* alle Session Keys und damit alle von B empfangenen Nachrichten entschlüsseln zu können

Verbindungsverschlüsselung

- Nicht alle Teilstrecken müssen verschlüsselt sein
- Wenig Kontrolle durch den Endnutzer
- In Vermittlungsstelle liegt Klartext vor

- Anwendungsgebiete:
  - Virtuelle Private Netze (VPN)
  - Leitungsverschlüsselung in Telekommunikationsnetzen
  - Richtfunkstrecken
  - WLAN



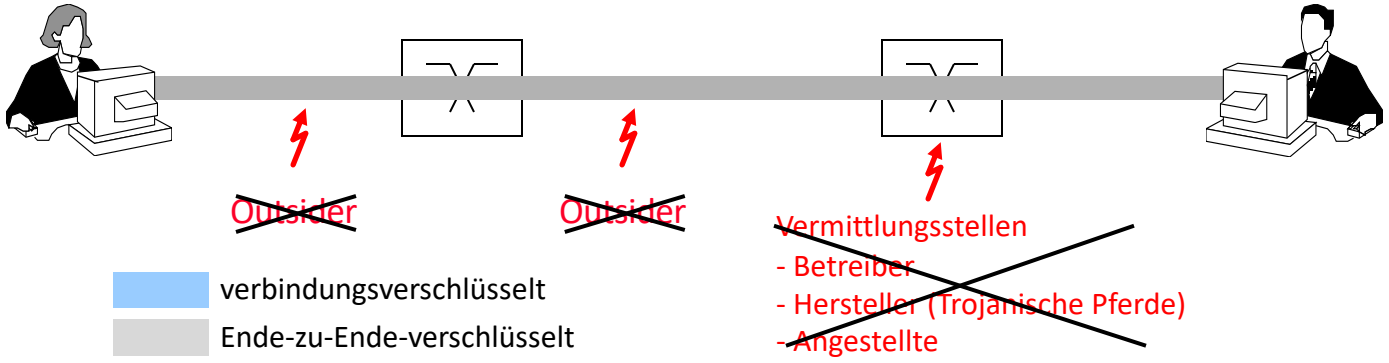
verbindungsverschlüsselt

Ende-zu-Ende-verschlüsselt

Ende-zu-Ende-Verschlüsselung

- Verschlüsselung der Inhalte von Endgerät zu Endgerät
- Symmetrisch und asymmetrische Verschlüsselung möglich
- Adressierungsinformation kann nicht mit verschlüsselt werden

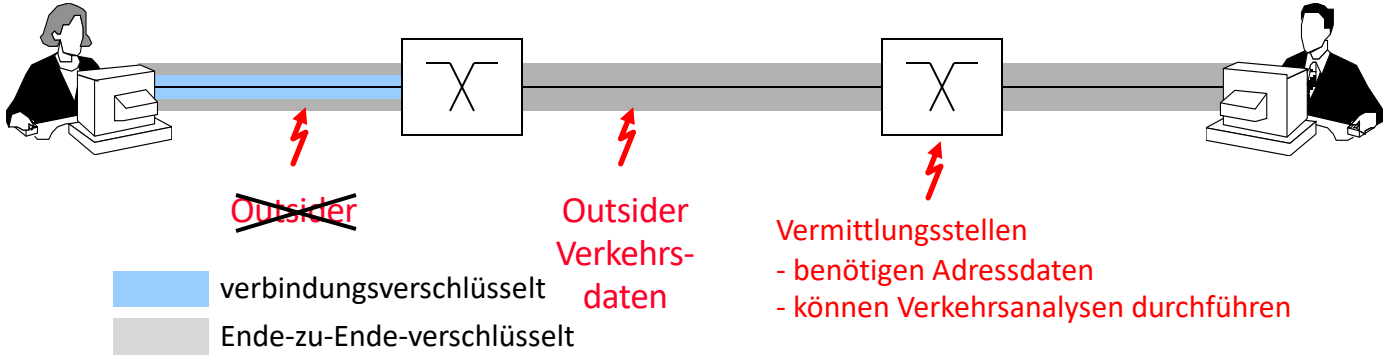
- Anwendungsgebiete:
  - Transport Layer Security (TLS)
  - Secure Shell (SSH)
  - E-Mail-Verschlüsselung mit PGP oder S/MIME



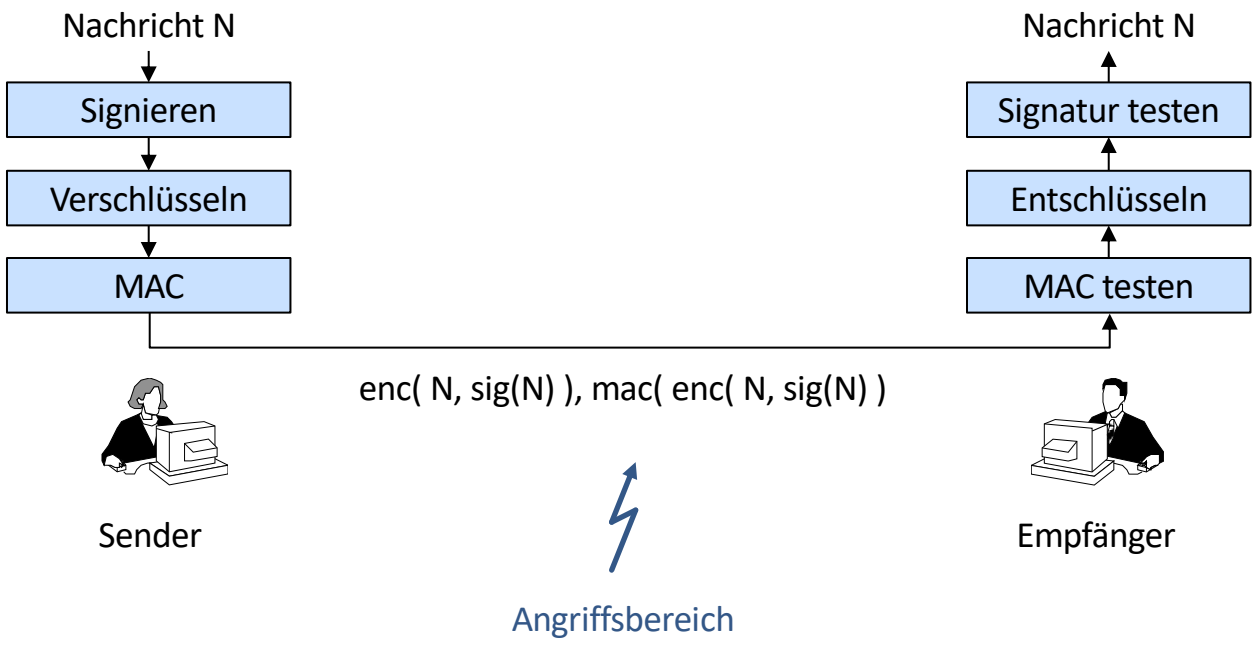
Kombination von Verbindungs- und Ende-zu-Ende-Verschlüsselung

- Ende-zu-Ende-Verschlüsselung allein schützt nicht die Adressierungsdaten vor Außenstehenden
- Zusätzliche Verbindungsverschlüsselung schützt auch Adressierungsdaten vor Außenstehenden

- Restproblem Verkehrsdaten
  - Netzbetreiber kann weiterhin feststellen, wer mit wem, wann, wie lange, wo, wieviel Information ausgetauscht hat



Kombination von Signatur, Verschlüsselung und MAC

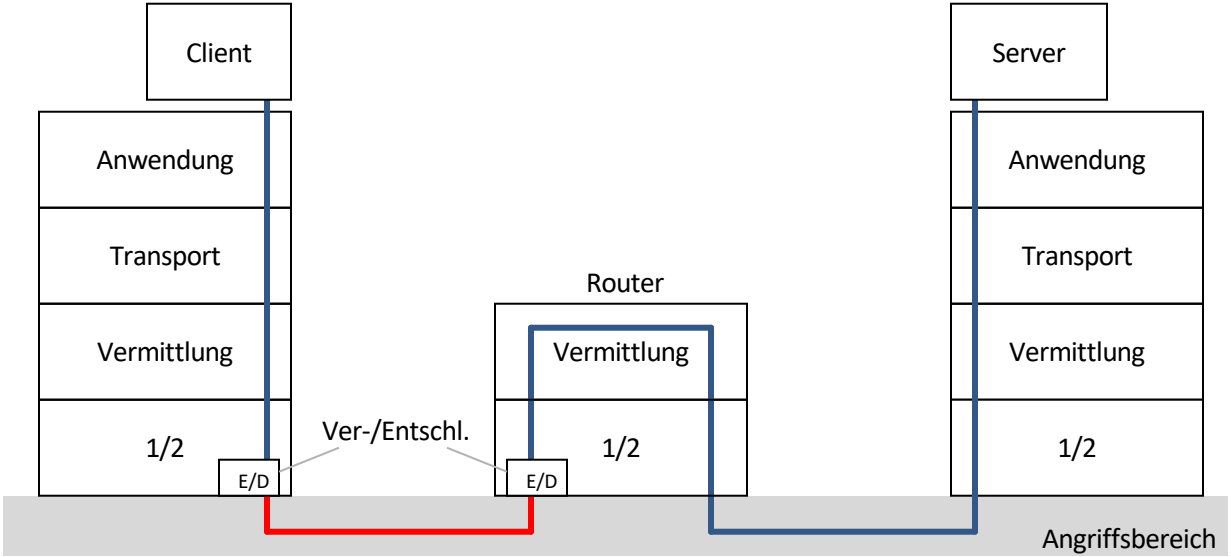


Sicherheitsfunktionen nach Schichten geordnet

Kommunikations-schicht im OSI-Referenzmodell	Sicherheitsfunktion
Anwendungsschicht	Pretty Good Privacy (PGP), S/MIME (Secure Multipurpose Internet Mail Extensions), Secure Shell (SSH)
Transportschicht	Transport Layer Security (TLS)
Vermittlungsschicht	Authentication Header (AH) zur Integritätssicherung von Datagrammen, Encapsulated Security Payload (ESP) zur Verschlüsselung von Datagrammen
Schichten 1 und 2	Challenge Handshake Protocol (CHAP, Passwort), Encrypt Control Protocol (ECP), WiFi Protected Access (WPA) 2

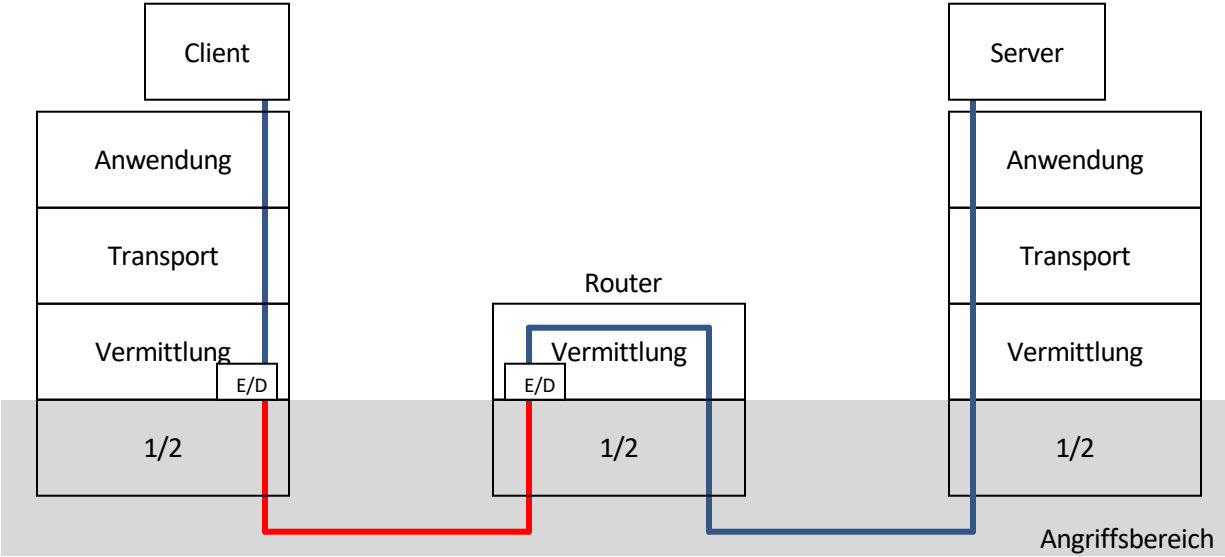
Verschlüsselung in Schichten 1 und 2

- Verbindungsverschlüsselung
- Beispiel: WLAN-Verschlüsselung
  - nur der Kommunikationsabschnitt zwischen Endgerät und WLAN-Router ist verschlüsselt



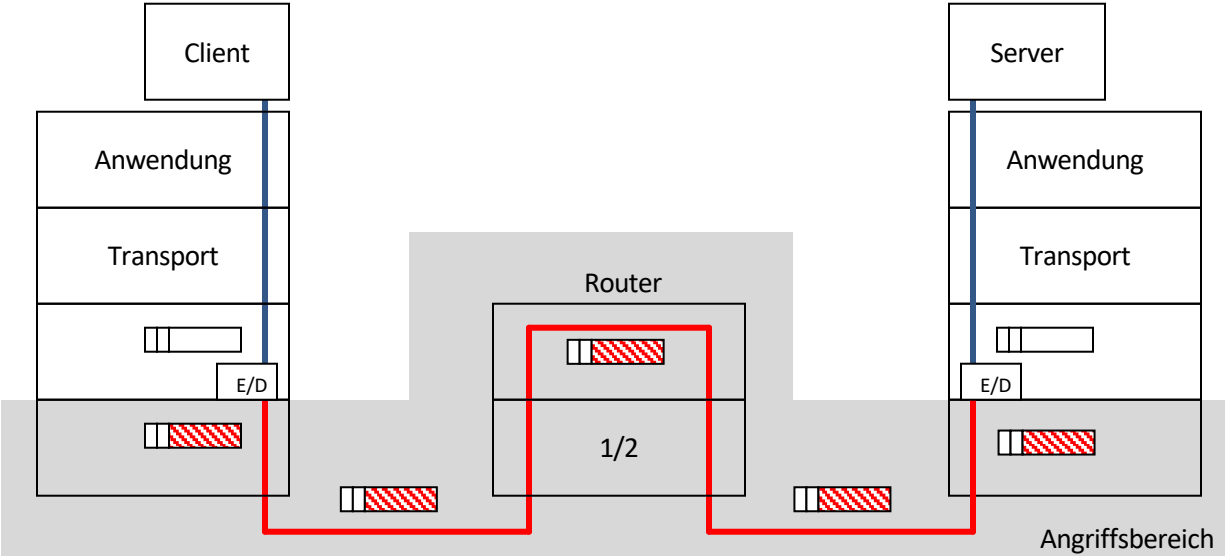
Verschlüsselung in Vermittlungsschicht: IPSec

- Transportmodus
  - Verbindungs- und Ende-zu-Ende-Verschlüsselung möglich



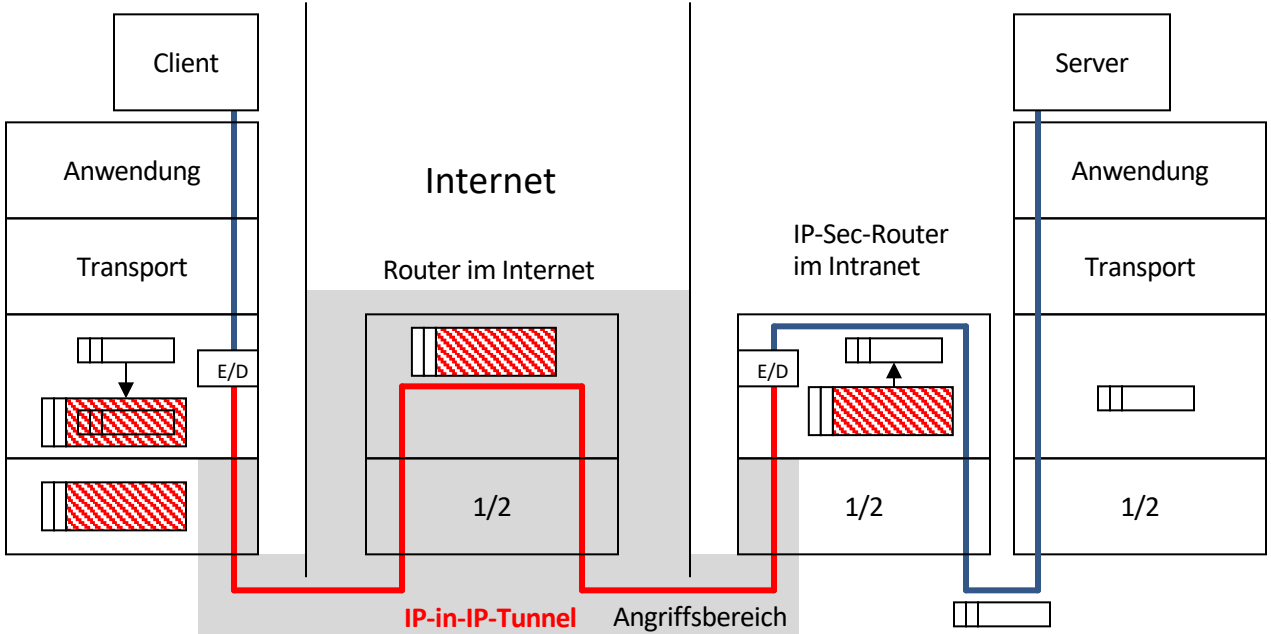
Verschlüsselung in Vermittlungsschicht: IPSec

- Transportmodus
  - Verbindungs- und Ende-zu-Ende-Verschlüsselung möglich



Verschlüsselung in Vermittlungsschicht: IPSec

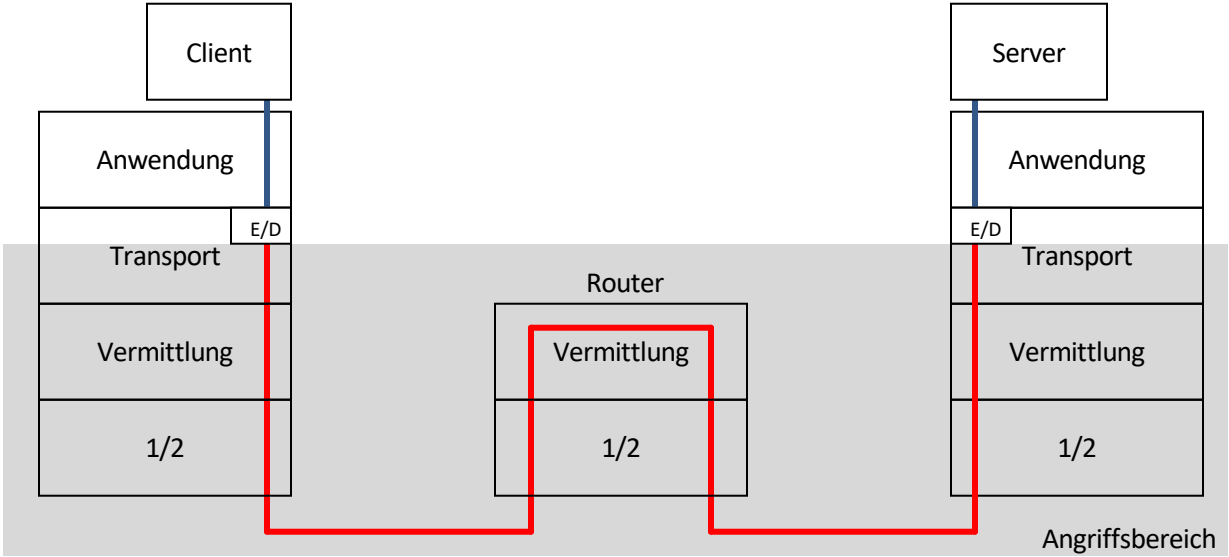
- Tunnelmodus
  - Momentane Hauptanwendung: Virtuelles Privates Netz
  - Verbindungsverschlüsselung zwischen Client und VPN-Gateway
  - Beispiele: IPSec, OpenVPN



VSS 17.04.2020

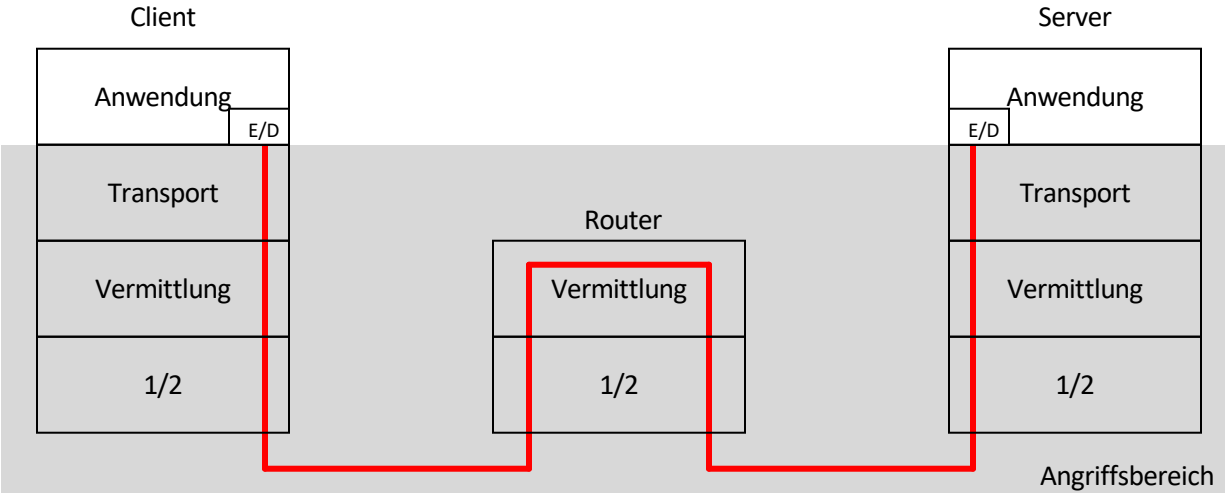
Verschlüsselung in Transportschicht: Transport Layer Security (TLS)

- Verschlüsselung von TCP-Verbindungen
- Beispiele: HTTPS, SMTP over SSL (Secure Sockets Layer)



Verschlüsselung in Anwendungsschicht

- Ende-zu-Ende-Verschlüsselung zwischen Client und Server
- Beispiele:
  - Secure Shell (SSH)
  - OpenPGP
  - S/MIME



Vergleich TLS – IPSec

	Schicht 7	Schicht 4	Schicht 3	Schicht 2
Beispiel	SSH	TLS	IPSec	WLAN
Komplexität	meist hoch	hoch	gering	gering
Anwendungsnähe	sehr hoch	hoch	gering	gering
Integrationsaufwand für Anwendungsentwickler	sehr hoch	gering	hoch	–
Für VPNs geeignet?	nein	bedingt	ja	–
Für paketorientierte Dienste geeignet?	ja+	ja*	ja	–
Für verbindungsorientierte Dienste geeignet?	ja	ja	ja	–

+ Quick UDP Internet Connections (QUIC)
* Datagram Transport Layer Security (DTLS)

E-Mail-Verschlüsselung

- Unverschlüsselte E-Mail-Kommunikation
  - SMTP (Simple Mail Transfer Protocol)
  - POP (Post Office Protocol)
  - IMAP (Internet Message Access Protocol)
- Verbindungsverschlüsselung
  - SMTP/POP/IMAP over SSL (Secure Sockets Layer)
  - Verschlüsselung zwischen MTAs (Mail Transfer Agents)
  - Sicherheit von DE-Mail etc.
- Ende-zu-Ende-E-Mail-Verschlüsselung
  - S/MIME
  - PGP

Herkömmliche E-Mails sind unsicher

- Unverschlüsselte E-Mail (schematisch):

Von: alice@example.de
An: bob@mail.com
Betreff: Abendessen

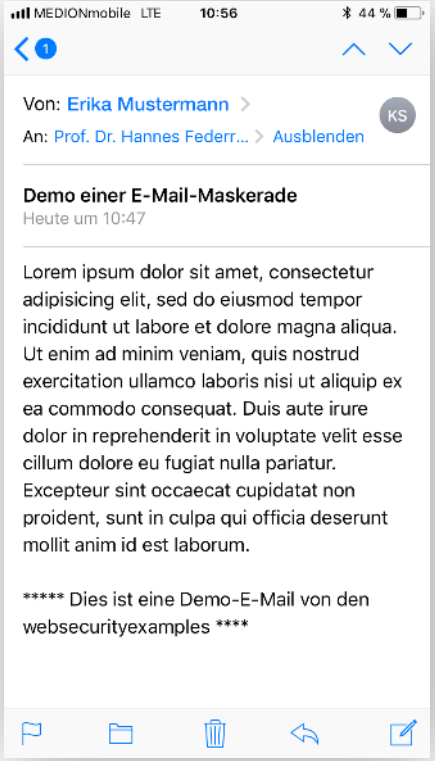
Hallo Bob,

wollen wir uns morgen zum Abendessen treffen?

Schöne Grüße, Alice

- Übertragung von E-Mail ist unsicher
  - Mitlesen möglich
  - Fälschen des Absenders möglich
  - Fälschen des Nachrichteninhalts möglich

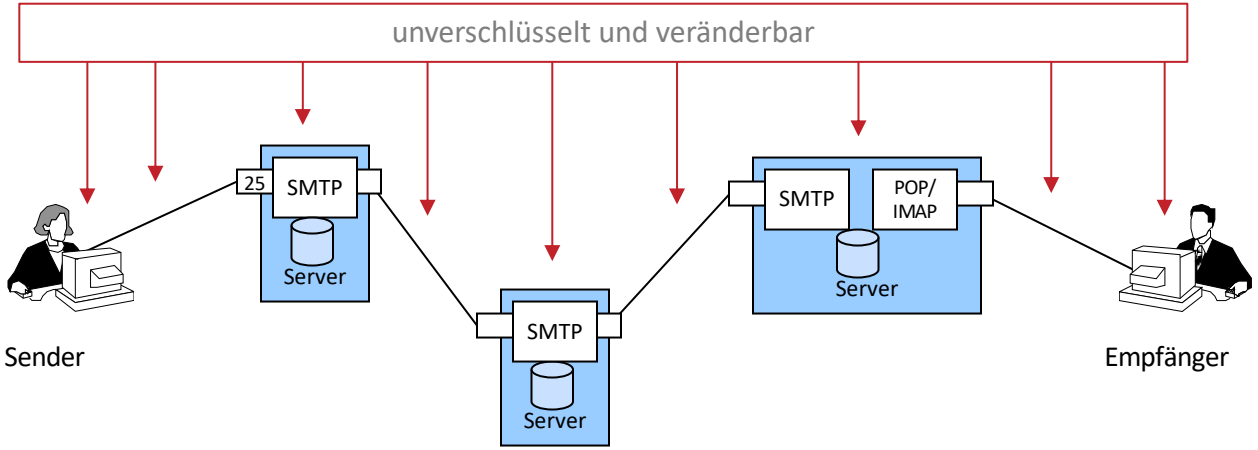
Herkömmliche E-Mails unsicher – Fälschen des Absenders möglich




```
Delivered-To: federrath@informatik.uni-hamburg.de
Received: from localhost (localhost [127.0.0.1])
  by mailhost.informatik.uni-hamburg.de (Postfix) with ESMTP id 3C70411A
  for <federrath@informatik.uni-hamburg.de>; Wed, 14 Mar 2018 10:47:55 +0100 (CET)
Received: from mailhost.informatik.uni-hamburg.de ([127.0.0.1])
  by localhost (mailhost.informatik.uni-hamburg.de [127.0.0.1]) (amavisd-new, port 10024)
  with LMTP id i4cyteXpTQ2I for <federrath@informatik.uni-hamburg.de>;
  Wed, 14 Mar 2018 10:47:54 +0100 (CET)
Received: from svs.informatik.uni-hamburg.de (svs.informatik.uni-hamburg.de)
  by mailhost.informatik.uni-hamburg.de (Postfix) with ESMTP id A60DB119
  for <federrath@informatik.uni-hamburg.de>; Wed, 14 Mar 2018 10:47:54 +0100 (CET)
Received: by svs.informatik.uni-hamburg.de (Postfix, from userid 33)
  id AA666610E8; Wed, 14 Mar 2018 10:47:54 +0100 (CET)
To: federrath@informatik.uni-hamburg.de
Subject: Demo einer E-Mail-Maskerade
From: “Erika Mustermann" <Erika.Mustermann@example.com>
Message-Id: <20180314094754.AA666610E8@svs.informatik.uni-hamburg.de>
Date: Wed, 14 Mar 2018 10:47:54 +0100 (CET)


Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud
exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure
dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit
anim id est laborum.


 ***** Dies ist eine Demo-E-Mail von den websecurityexamples ****
```

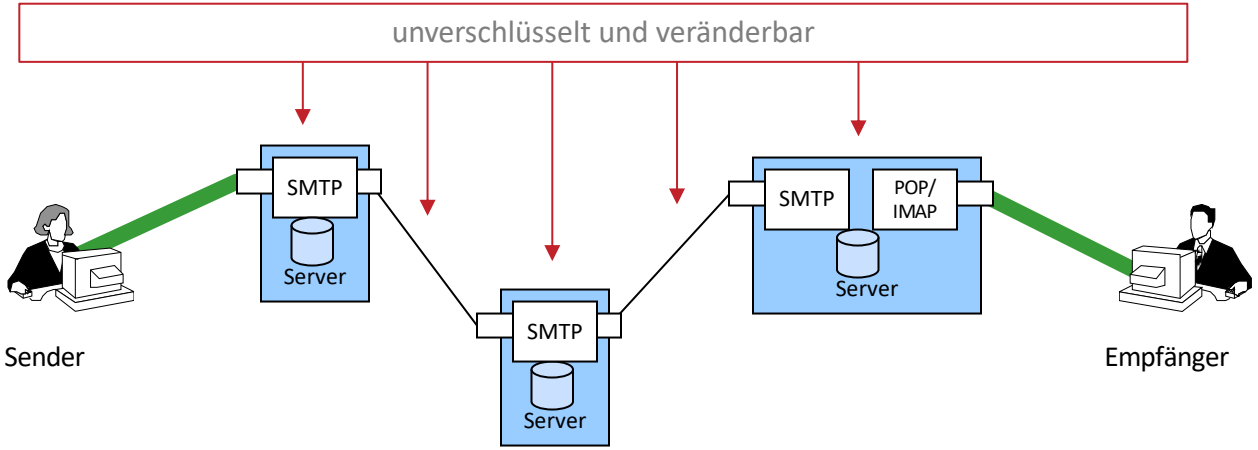
Grundlagen der E-Mail-Kommunikation

- Store & Forward-Prinzip:
  - E-Mail wird nicht direkt an Empfänger, sondern über MTA (Mail Transfer Agents) geschickt
- Senden via SMTP: Textbasiertes Protokoll auf TCP-Port 25
- Empfangen: POP, IMAP



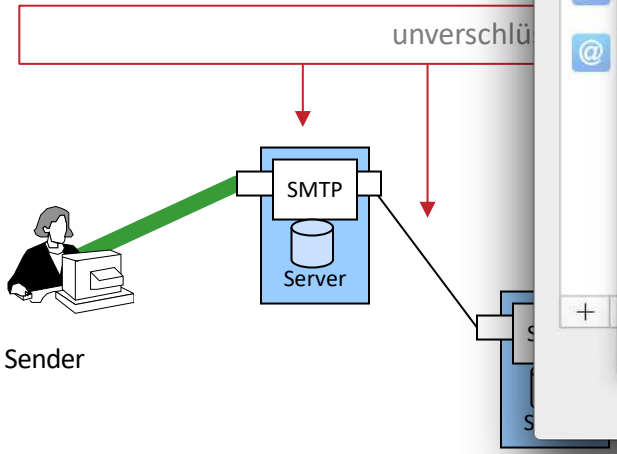
Verbindungsverschlüsselung

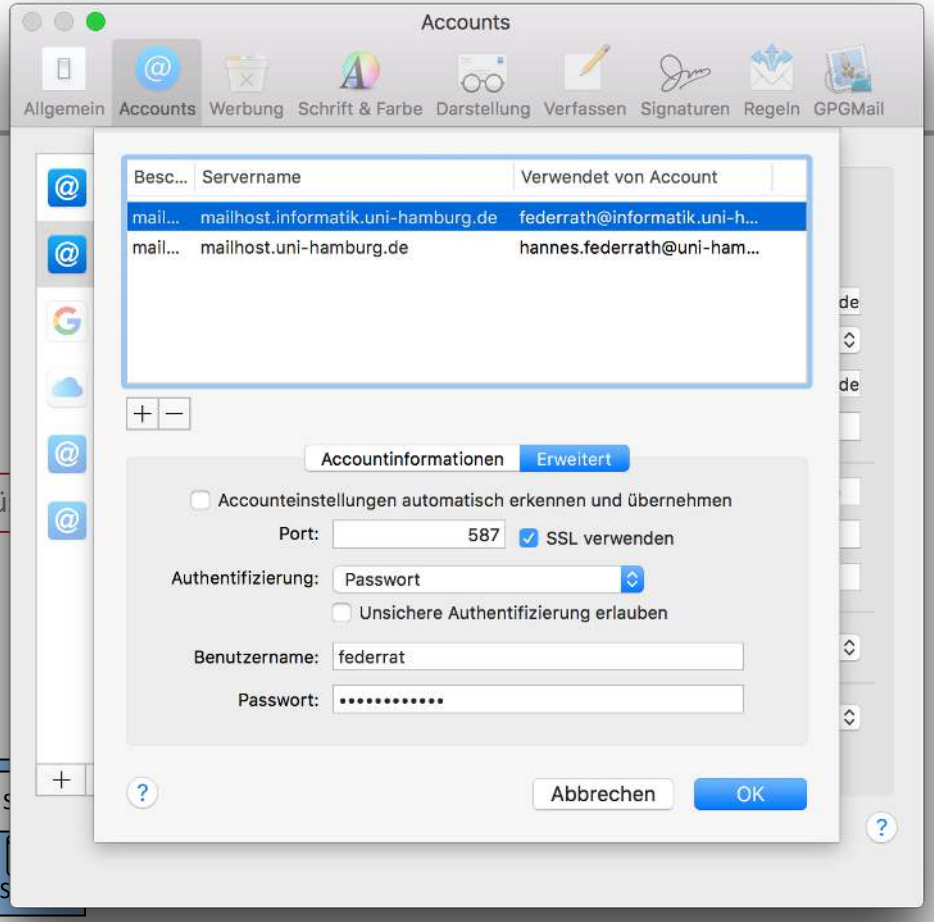
- SMTP/POP/IMAP over SSL (Secure Sockets Layer)
  - Verschlüsselte Teilstrecke zwischen
    - Sender und MTA
    - MTA und Empfänger
  - alle MTAs können mitlesen



Verbindungsverschlüsselung

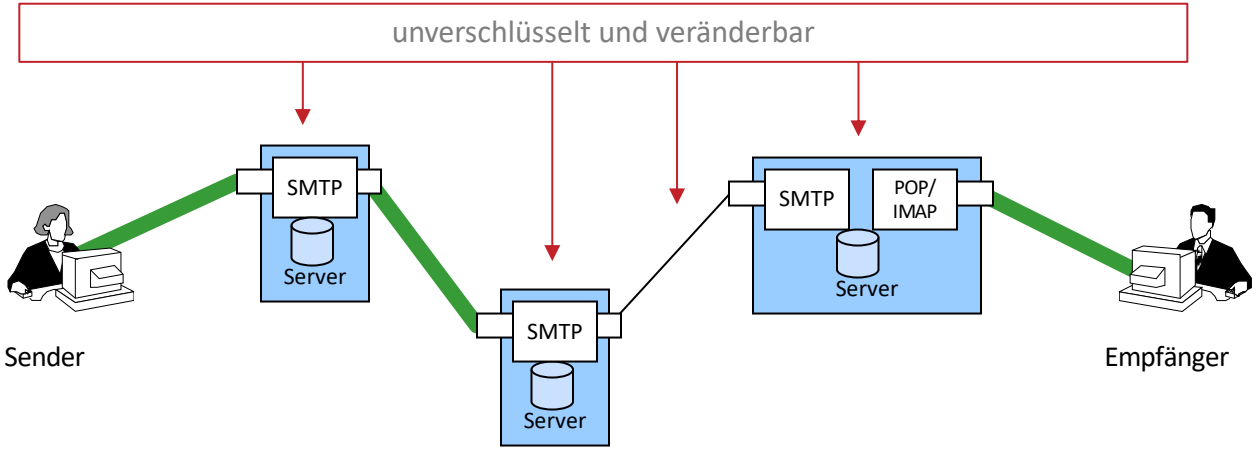
- SMTP/POP/IMAP over SSL
  - Verschlüsselte Teilstrecke zwischen
    - Sender und MTA
    - MTA und Empfänger
  - alle MTAs können mitlesen





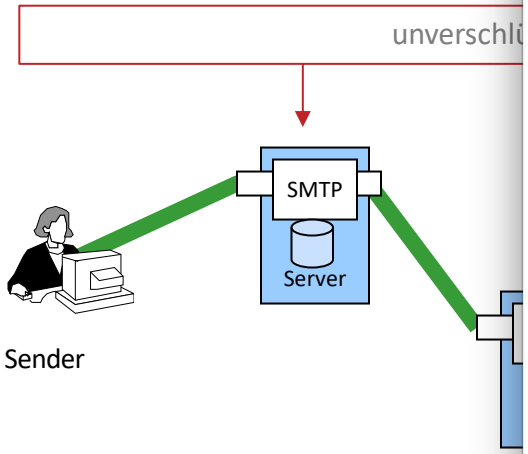
Verbindungsverschlüsselung

- SMTP/POP/IMAP over SSL (Secure Sockets Layer)
- Verschlüsselung zwischen MTAs
  - Verschlüsselte Teilstrecke zwischen MTAs
  - nicht alle Teilstrecken müssen verschlüsselt sein
  - alle MTAs können mitlesen



Verbindungsverschlüsselung

- SMTP/POP/IMAP over SSL
- Verschlüsselung zwischen MTAs
  - Verschlüsselte Teilstrecke zwischen MT
  - nicht alle Teilstrecken müssen verschlüs
  - alle MTAs können mitlesen



www.heise.de/newsticker/meldung/E

heise online

"E-Mail Made in Germany": SSL-Verschlüsselung für (fast) alle

UPDATE

09.08.2013 12:40 Uhr – Detlef Borchers

Die Deutsche Telekom und United Internet haben das Projekt **"E-Mail made in Germany"** [1] in Berlin vorgestellt. Dabei handelt es sich um eine SSL/TSL-Verschlüsselung zwischen den Mail-Servern und Rechenzentren der beteiligten Firmen, die ab sofort genutzt werden kann. Ab 2014 sollen nur noch SSL-verschlüsselte Mails transportiert werden. Rund 20 Millionen Kunden der Telekom und 30 Millionen von GMX und Web.de sollen die "deutsche E-Mail" nutzen können. Sie ist bereits in den Webmailern der beteiligten Firmen freigeschaltet und kommt automatisch zum Einsatz: Bei Eingabe von Empfängeradressen aus dem Mail-Verbund wird eine Information angezeigt, dass die Mails verschlüsselt übertragen werden. Bislang gibt es allerdings noch keine entsprechende Anzeige bei empfangenen Mails.

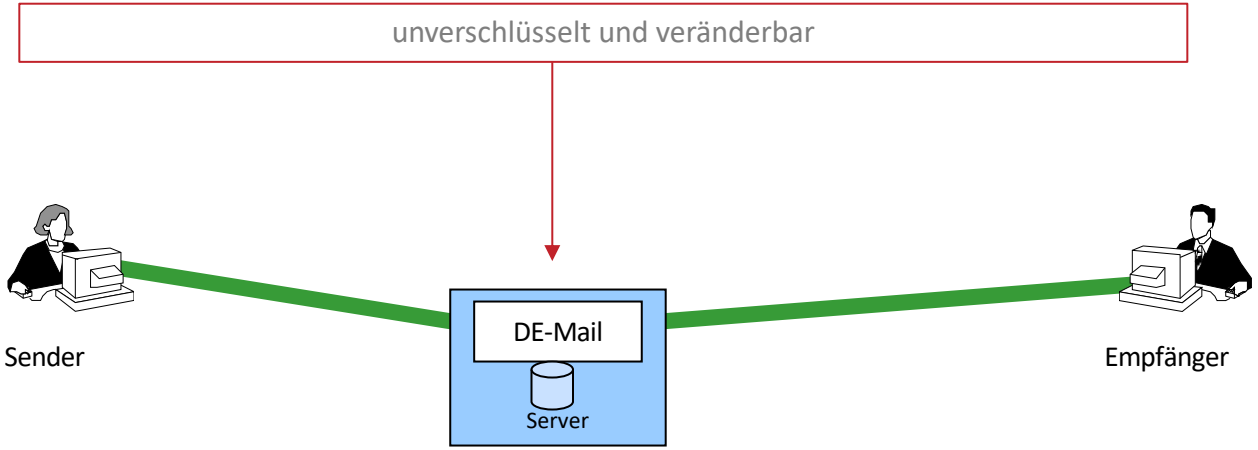
Die Web-Mailer der Telekom, von GMX und web.de zeigen ab sofort an, wenn ein E-Mail-Empfänger aus dem Verschlüsselungs-Verbund "E-Mail made in Germany" kommt.

Wie Telekom-Chef René Obermann erklärte, hofft man, dass sich andere Wettbewerber wie Freenet oder Arcor der deutschen Initiative anschließen werden. Geprüft werde derzeit, wie die Tochterfirmen Strato (Telekom) und 1&1 (United Internet) in das System eingebunden werden können, damit auch Kleingewerbetreibende mit Domains wie z.B. Obermann.de die sichere deutsche "Mail made in Germany" nutzen können. Zudem hofft Obermann darauf, dass aus der deutschen Initiative eine europäische wird. Ralph Dommermuth von United Internet erklärte, dass 90 Prozent der Mail innerhalb von Deutschland ausgetauscht werde und somit die "E-Mail made in Germany" zum Standard werde.

Meldung: https://heise.de/-1932962

Verbindungsverschlüsselung

- Sicherheit von DE-Mail etc.
  - gleiche Sicherheit wie »E-Mail Made in Germany«
  - keine Ende-zu-Ende-Verschlüsselung
  - Server kann mitlesen



Verbindungsverschlüsselung

- Sicherheit von DE-Mail etc.
  - gleiche Sicherheit wie »E-Mail Made in Germany«
  - keine Ende-zu-Ende-Verschlüsselung
  - Server kann mitlesen

Systeme von Absender und Empfänger die Identität der jeweils anderen Seite sicherstellen können. Eine Möglichkeit hierfür ist der Einsatz digitaler Zertifikate.

- Eine abschnittsweise Verschlüsselung der Übertragungswege soll Sicherheit gegen den Zugriff durch Unbefugte bieten.

Quelle: Wikipedia

Bei besonders hohen Anforderungen an die Vertraulichkeit der Nachrichten haben De-Mail-Nutzer wie bei herkömmlichen E-Mails die Möglichkeit, die mit De-Mail übermittelten Inhalte noch zusätzlich selbst zu verschlüsseln („Ende-zu-Ende-Verschlüsselung"). In diesem Fall erfolgt die Verschlüsselung auf dem Rechner des Absenders und die Entschlüsselung der Inhalte erst auf dem Rechner des Empfängers. Hierfür ist jedoch die Installation zusätzlicher Software erforderlich, die die Ver- und Entschlüsselung durchführt. Der Verzeichnisdienst, der bei De-Mail obligatorisch durch den Diensteanbieter angeboten werden muss, unterstützt den Anwender, indem dieser seinen öffentlichen Schlüssel anderen Anwendern zur Verfügung stellen kann. Es soll also ermöglicht werden, an einer zentralen Stelle nach öffentlichen Schlüsseln von Personen suchen zu können, um mit diesen vertraulich zu kommunizieren. Dies ist bisher nur schwer möglich und stellt den wesentlichen „Hemmschuh" für die Verbreitung von Verfahren zur Ende-zu-Ende-Verschlüsselung („Wo finde ich den gültigen Verschlüsselungsschlüssel meines Kommunikationspartners?"). Mit De-Mail soll auf diese Weise der Einsatz der Ende-zu-Ende-Verschlüsselung unterstützt und gefördert werden.

S/MIME oder OpenPGP können zusätzlich durch den Nutzer für die Abbildung der Ende-zu-Ende-Sicherheit eingesetzt werden. Die Liste der hauptsächlichen Sicherheitsfunktionen wurden in einem Dokument des BMI zusammengefasst.

Technische Konzeption [Bearbeiten | Quelltext bearbeiten]

Das technische Konzept ist innerhalb von technischen Richtlinien beschrieben, die auf der Web-Seite des BSI[19] veröffentlicht sind.

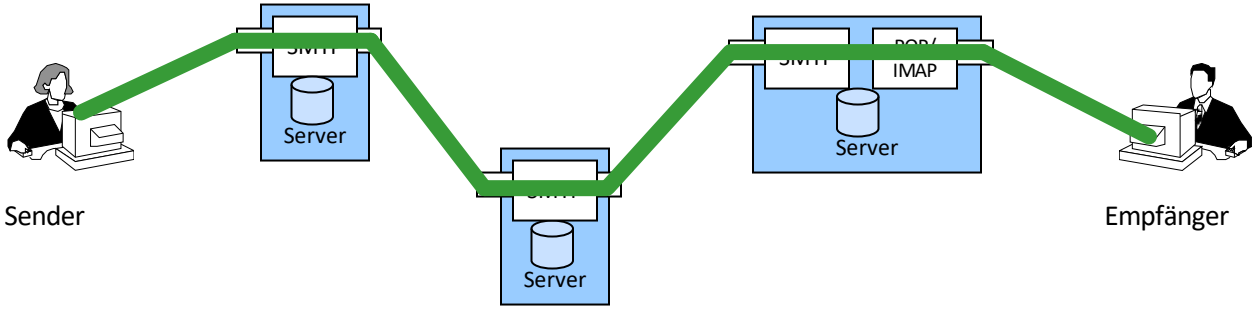
Standardmäßige Transportsicherheit [Bearbeiten | Quelltext bearbeiten]

Sowohl die Kommunikation der De-Mail-Nutzer mit ihren De-Mail-Provider als auch die Kommunikation von De-Mail-Anbietern untereinander verläuft grundsätzlich über TLS-gesicherte Kommunikationskanäle.[5] Reicht einem Nutzer das dadurch realisierte Sicherheitsniveau nicht

VSS 17.04.2020

Ende-zu-Ende-Verschlüsselung schützt die Vertraulichkeit

- Ende-zu-Ende-Verschlüsselung
  - Inhalte sind durchgehend verschlüsselt
  - MTAs (Server) können nicht mitlesen
  - Header (Adressen, Datum, Betreff etc.) bleiben unverschlüsselt
- Konkrete Ende-zu-Ende-Verschlüsselungslösungen
  - S/MIME
  - PGP



Ende-zu-Ende-Verschlüsselung schützt die Vertraulichkeit

- Verschlüsselte E-Mail (schematisch):

Von: alice@example.de

An: bob@mail.com

Betreff: Abendessen

-----BEGIN ENCRYPTED MESSAGE-----
AkRFMRcwFQYDVQQDEw5Sb2xmIFdlbmRvbHNreTEUMBI
GA1UECBMLRGV1dHNjaGxhbmxEzARBgNVBAcTClJlZ2V
uc2J1cmcxDzANBgNVBAoTBnBaXZhdDEkMCIGCSqGSIb
3DQEJARYVcm9sZi53ZW5kxza3lAZ214LmRlMIGfMA0G
CSqGSIb3DQEBAQUAA4GNABDCBiQKBgQDUVvgaQK9OQP
XvgZm2bU/QqDnsbemv83gDiSuCq07S/cSMiGFjEZas6
5MZ47W951LlNLvFTjwS2fUfsZ5oAxfU+RDWb3GgijZp
5cAxTfFKQ/amaWAtmCkt1FMntRXZ393gOkSSUlWQ7Cr
6GWAYF+deC5CuWptRPpSLYRqSwIDAQABMA0GCSqGSIb
3DQEBBAUAA4GBAIGVTNbu0eOTfGuuL0MWHLfVD
-----END ENCRYPTED MESSAGE-----

Signatur schützt Integrität und Authentizität

- Signierte E-Mail (schematisch):

Von: alice@example.de
An: bob@mail.com
Betreff: Abendessen

-----BEGIN SIGNED MESSAGE-----
Hallo Bob,

wollen wir uns morgen zum Abendessen treffen?

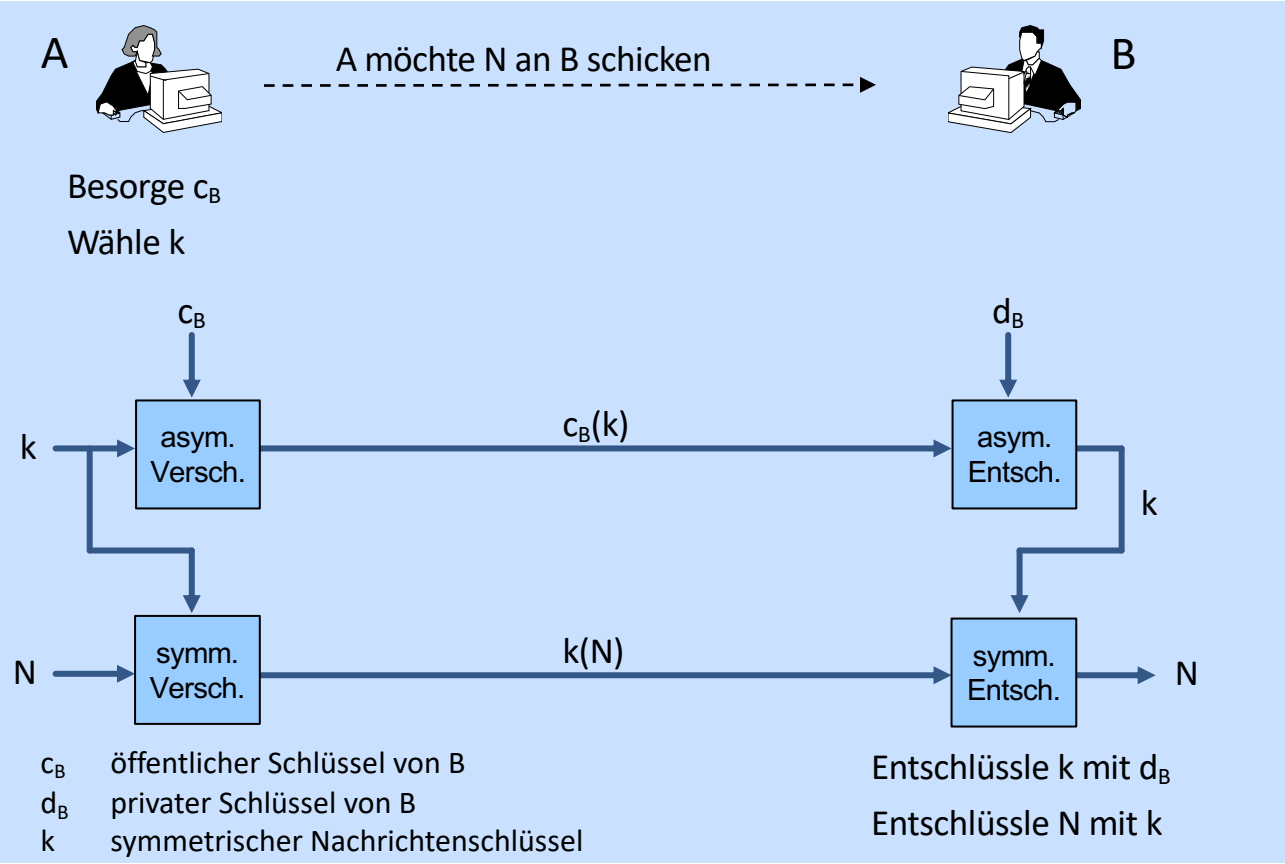
Schöne Grüße, Alice

-----BEGIN SIGNATURE-----
U/QqDnsbemv8p83gDiSuCq07S/cSMiGFjEZas65MZ47W
951LlNLvFTSdkjwS2fUfsZ5oAxfU+RDWb3GgijZAxTfF
F+deC5CuWpRPpSLYRqSwIDAQABMA0GCSqGSIb
-----END SIGNATURE-----

Konkrete Ende-zu-Ende-Verschlüsselungslösungen

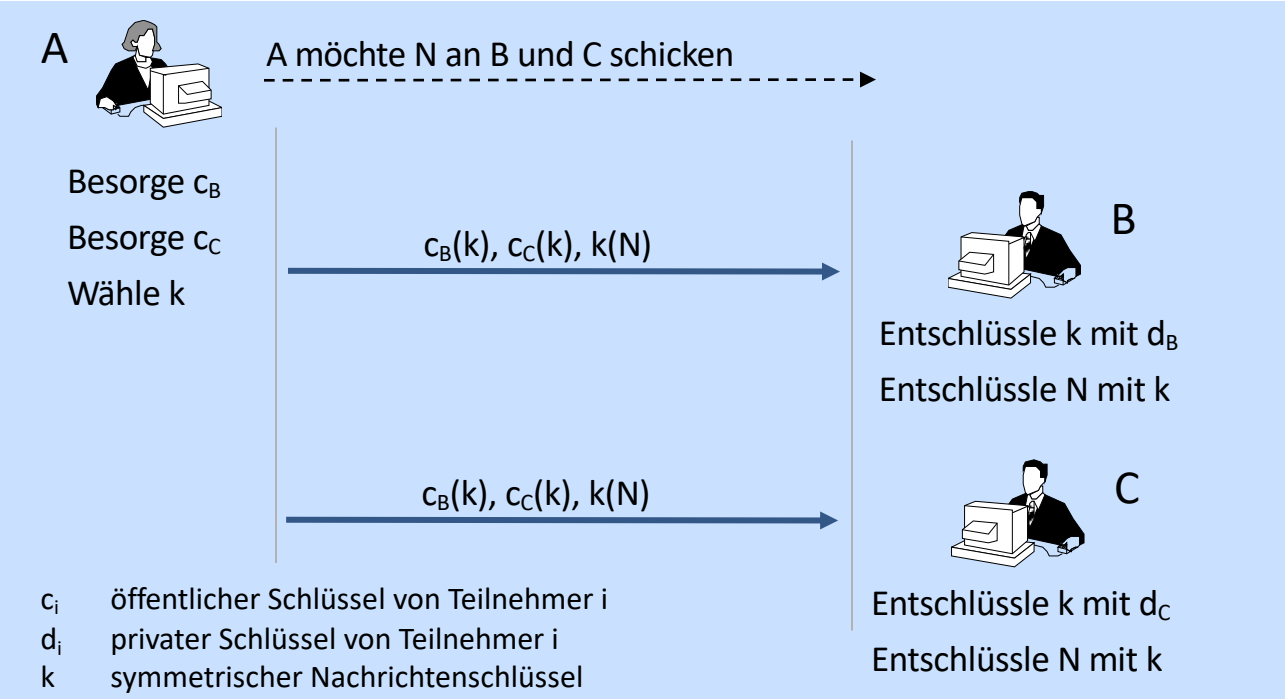
- S/MIME (Secure Multipurpose Internet Mail Extensions)
  - ursprünglich von RSA Data Security Inc.
  - S/MIME v3 im Juli 1999 als IETF-Standard verabschiedet
  - Internet Standards RFCs 2632-2634 (und weitere)
  - in die meisten E-Mail-Clients integriert
  - E-Mailverschlüsselung und -signatur
  - One-pass processing: Die Mail ist in einem Schritt verarbeitbar, da alle benötigten Daten in die Mail selbst integriert sind
- PGP (Pretty Good Privacy)
  - 1991 von Philip Zimmermann entwickelt
  - E-Mail- und Dateiverschlüsselung und -signatur
  - heute: Open PGP, GnuPG (Gnu Privacy Guard)
  - Zahlreiche grafische Frontends erhältlich, z.B. GPA, WinPT
  - Plugins für verschiedene Mailclients, z.B. Outlook, Thunderbird, Pegasus, KMail

S/MIME und PGP nutzen hybride Verschlüsselung



S/MIME und PGP nutzen hybride Verschlüsselung

- Hybrides Verschlüsselungsverfahren
  - Nachrichten selbst werden symmetrisch verschlüsselt
  - Nachrichtenschlüssel werden asymmetrisch verschlüsselt



- Ende-zu-Ende-Verschlüsselung schützt die Vertraulichkeit


```
-----BEGIN PGP MESSAGE-----
Version: GnuPG v1.4.8 (Darwin)

hQIOA2ThYngSETJKEAgA4I9+HpuDVc95Sip7xgHXQXooxpEQZb7xaAV84XhS1y48
wDDhe4Dk9kwqKlYZJgw5Df/pW9MzhJHi1Q9jgU90AE5t5IkmcB+EJSor1YqxUjMZ
Q7baKGRNBQhVNP/+5i3K2GuuXVeYaccYfGvF4mSAnremmbHeH0L9j6cSrGCsqQqa
Vaml6/s2j1ufdPLuTF9g3i0xeQuJnv5pKv0DcTxwPR0mLBOKJUS6DDUq1lY6rviO
l7kM72jIz83wl6PZAfWmzj4IyZq8ktPcZ/fdnrZ50FE34Vfwzvh0lbRqdeFY6GPW
f6Y3FnF9DJUkM1kYuAp65X6E19fapJdAnTvjB2WV9XWrmPypJIcF5kTXL8vOLCtu
yZ6R+PS0q6c=
=x491
-----END PGP MESSAGE-----
```


- Digitale Signatur schützt Integrität und Authentizität


```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

Das ist die Nachricht.

-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.8 (Darwin)

iEYEARECAAYFAkjh9yQACgkQ4UAgYUnvHYSahQCfaWrrHll9s4tXeFToa6aQPryw
TX4AoL7l7WQHHXPzxVG6SX9fSOAskCzn
=Ebit
-----END PGP SIGNATURE-----
```

Gnu Privacy Guard (GnuPG)

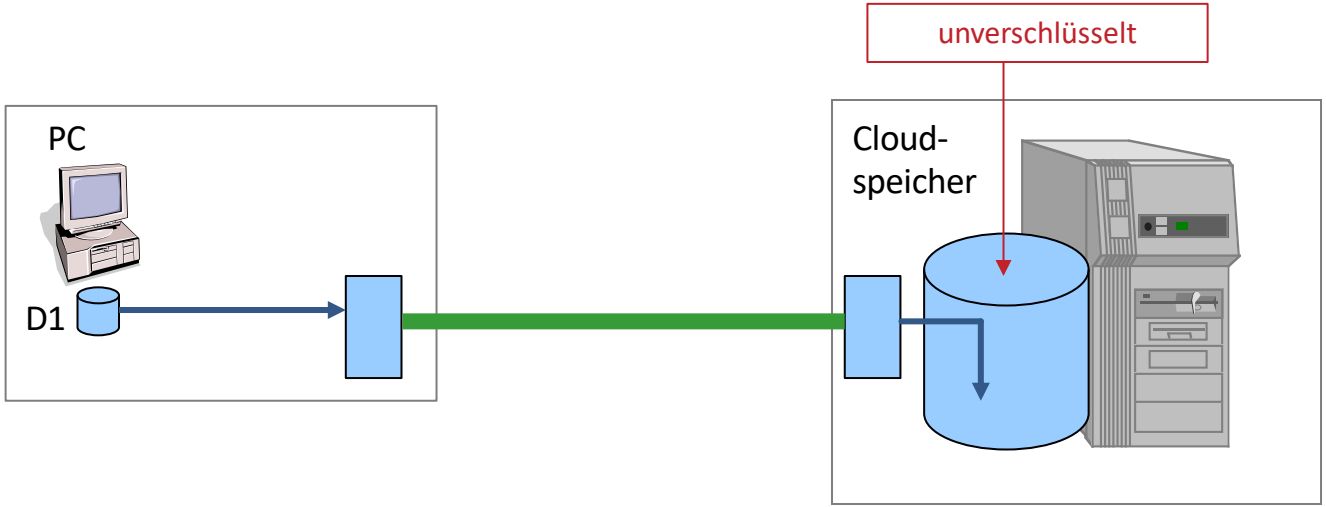
- GnuPG unter Android
  - K-9 Mail und OpenKeychain: Easy PGP
    https://play.google.com/store/apps/details?id=com.fsck.k9
    https://play.google.com/store/apps/details?id=org.sufficientlysecure.keychain

- GnuPG unter iOS
  - iPGMail
    https://itunes.apple.com/de/app/ipgmail/id430780873

  - oPenGP
    https://itunes.apple.com/de/app/opengp/id414003727

Verschlüsselte Cloud-Speicher

- Verbindungsverschlüsselung ist heute Standard

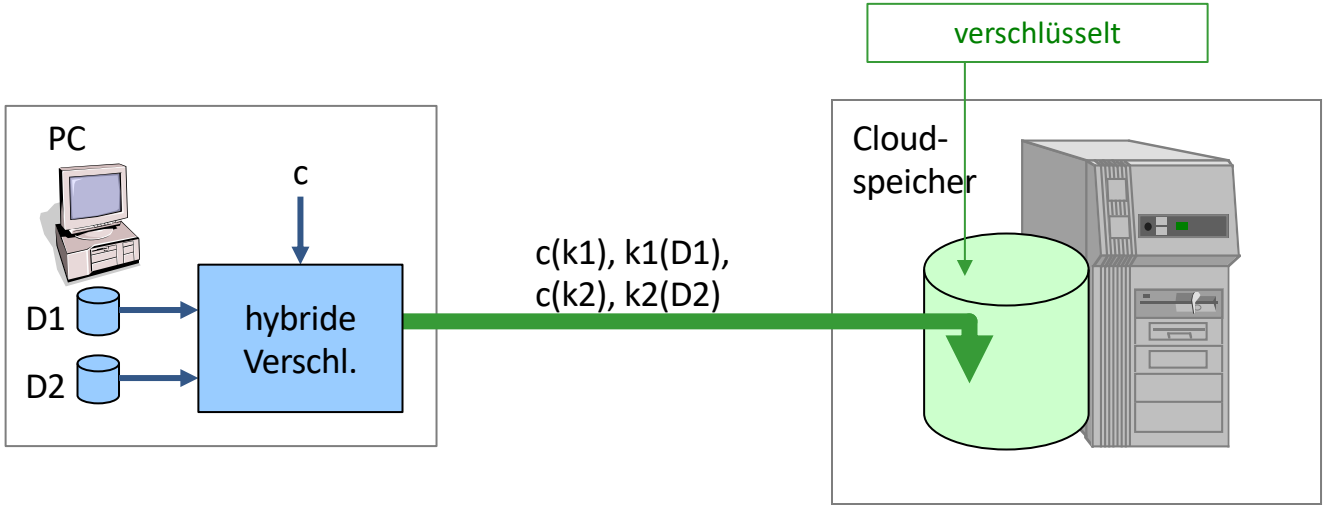


Verschlüsselte Cloud-Speicher

- Ende-zu-Ende-Verschlüsselung wäre kein Problem

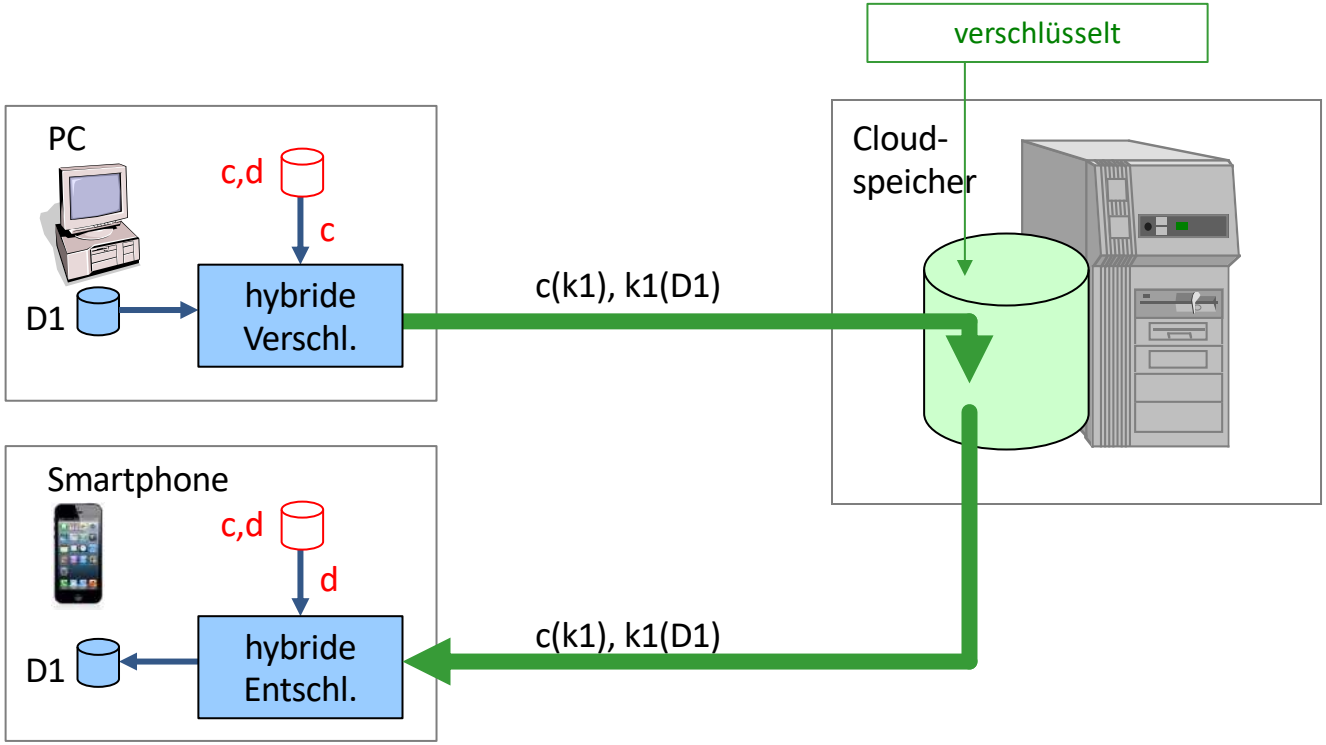
Verschlüsselte Cloud-Speicher

- Ende-zu-Ende-Verschlüsselung besser hybrid …



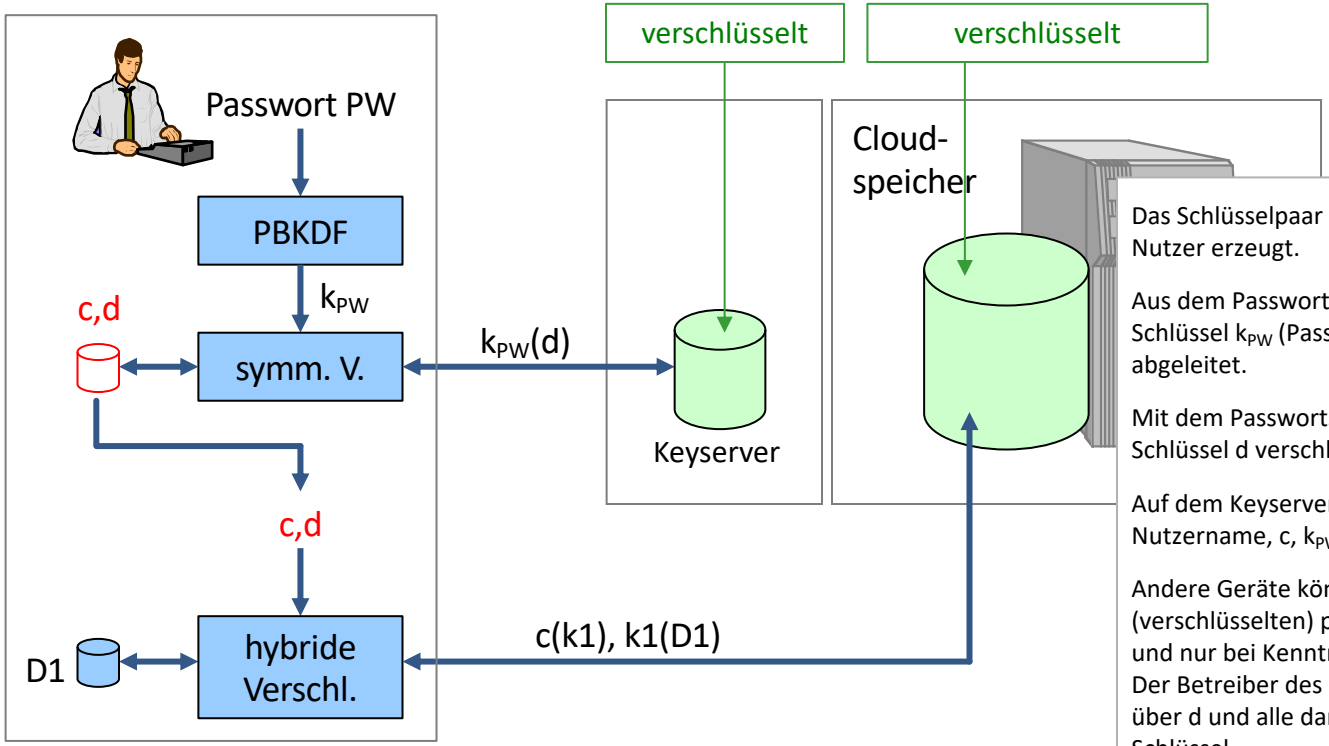
Verschlüsselte Cloud-Speicher – Usability-Aspekte

- Wie bekommt der Nutzer auf alle seine Geräte seine(n) Schlüssel?



Verschlüsselte Cloud-Speicher – Usability-Aspekte

- Wie bekommt der Nutzer auf alle seine Geräte seine(n) Schlüssel?



Das Schlüsselpaar c,d wird lokal beim Nutzer erzeugt.

Aus dem Passwort PW wird ein symm. Schlüssel k_{PW} (Passwortschlüssel) abgeleitet.

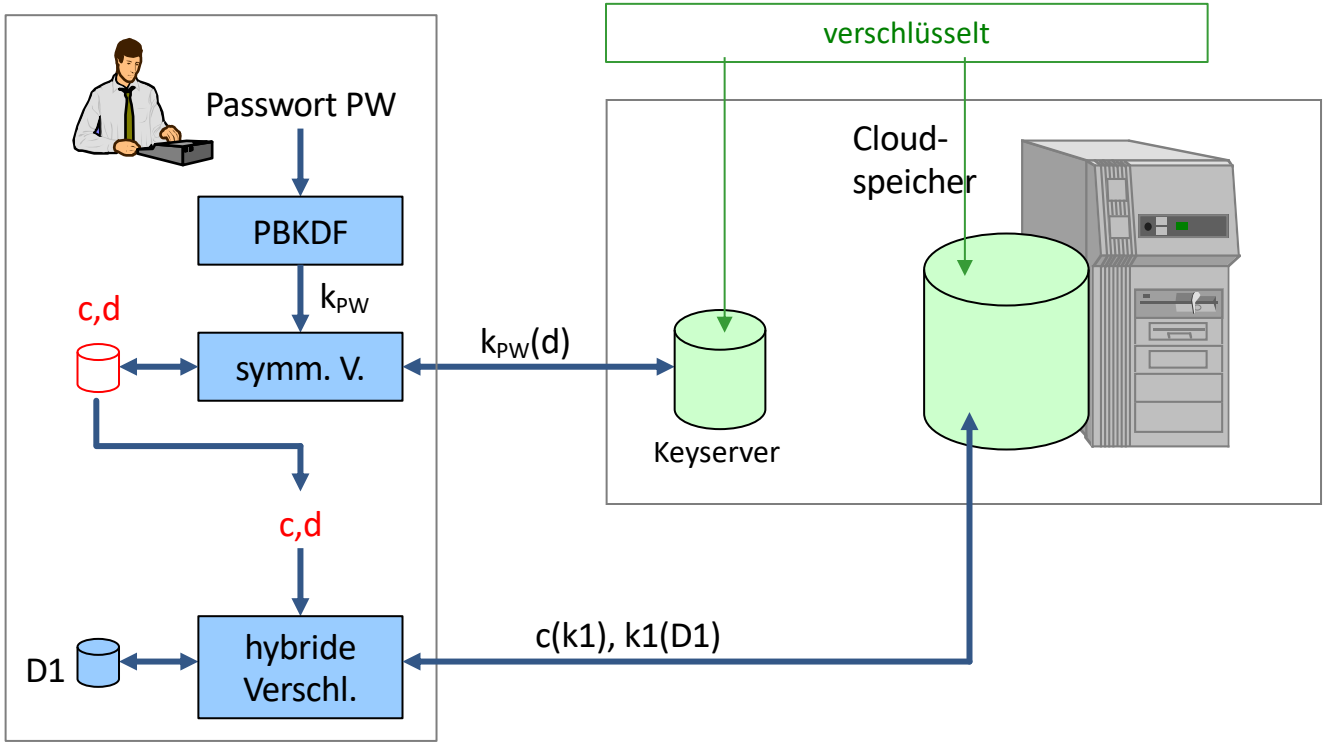
Mit dem Passwortschlüssel wird der private Schlüssel d verschlüsselt.

Auf dem Keyserver werden abgelegt: Nutzername, c, $k_{PW}(d)$.

Andere Geräte können sich den (verschlüsselten) privaten Schlüssel holen und nur bei Kenntnis von PW entschlüsseln. Der Betreiber des Keyservers erfährt nichts über d und alle damit verschlüsselten Schlüssel.

Verschlüsselte Cloud-Speicher – Usability-Aspekte

- Wie bekommt der Nutzer auf alle seine Geräte seine(n) Schlüssel?



Key Recovery und Key Escrow

- Key Recovery
  - Hinterlegung des Entschlüsselungsschlüssels zum Zweck der Entschlüsselbarkeit bei Schlüsselverlust.
  - Schwellwertschema: Schlüssel wird in n+k Teile zerlegt. Zur Rekonstruktion werden wenigstens n Teile benötigt.
- Key Escrow
  - Hinterlegung des Entschlüsselungsschlüssels zum Zweck der Strafverfolgung.
  - so dass alle Nachrichten ab einem bestimmten Zeitpunkt entschlüsselt werden können
  - so dass Nachrichten auch rückwirkend entschlüsselt werden können
- Beachte
  - Signaturschlüssel müssen nie hinterlegt werden, da eine Signatur stets testbar bleibt.
  - Bei Verlust des Signierschlüssels: neuen erzeugen.

Key Recovery

		Schutz der Kommunikation	Langfristige Speicherung
Verschlüsselung		Key Recovery für Funktion unnötig, aber zusätzliches Sicherheitsrisiko	Key Recovery sinnvoll
Authen-tikation	symmetrisch (MACs)		
	asymmetrisch (dig. Signatur)		

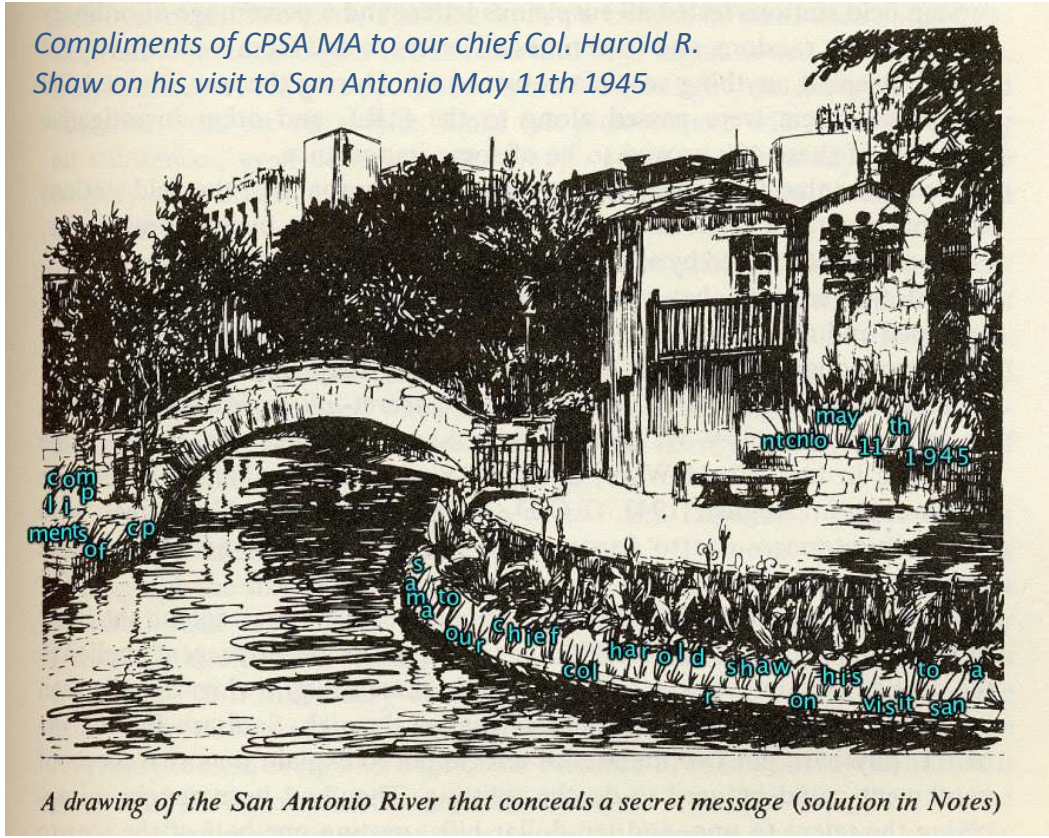
Steganographie (und Watermarking)

Steganographie und Watermarking

- Steganographie historisch: Verdecktes Schreiben
  - Geheimtinte
  - Worte in bestimmten Abständen
- Moderne Steganographie
  - geheimzuhaltende Nachricht wird in eine Hülle eingebettet
  - minimale Veränderungen nicht erkenn- und nachweisbar bzw. entfernbar
- Abgrenzung zur Kryptographie
  - Verschlüsselung
    - Schlüsseltext ist als solcher erkennbar
  - Steganographie
    - Einer Hülle sieht man die eingebettete Botschaft nicht an
  - Watermarking
    - Hülle mit schwer erkenn- und entfernbarer Markierung

Semagramm

David Kahn: The Codebreakers. Scribner, 1996, S. 532

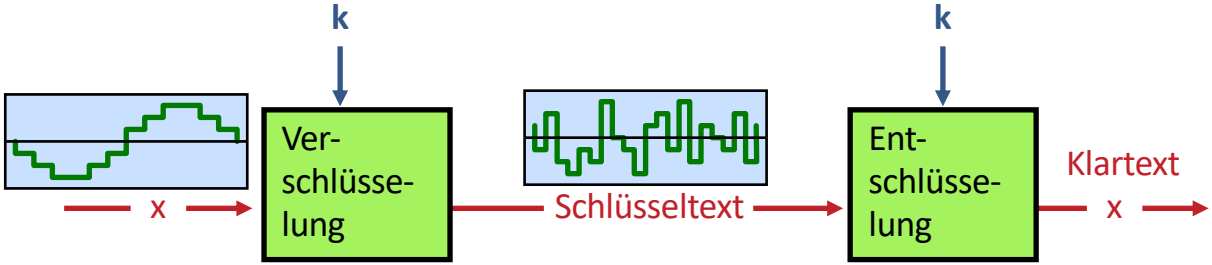


A drawing of the San Antonio River that conceals a secret message (solution in Notes)

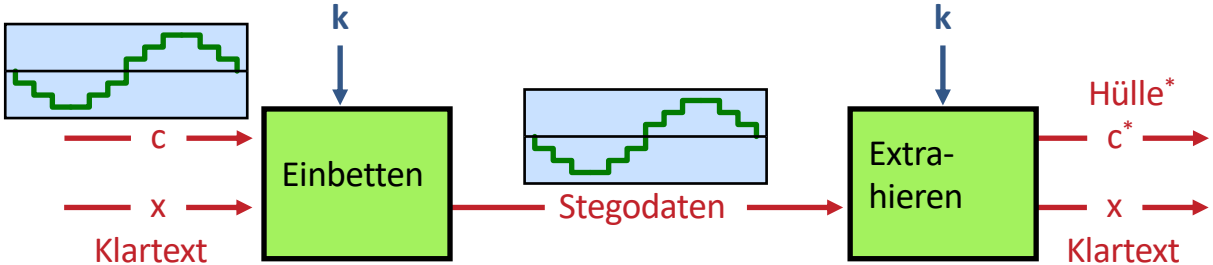
VSS 17.04.2020

Abgrenzung zur Verschlüsselung

- Verschlüsselung ist erkennbar

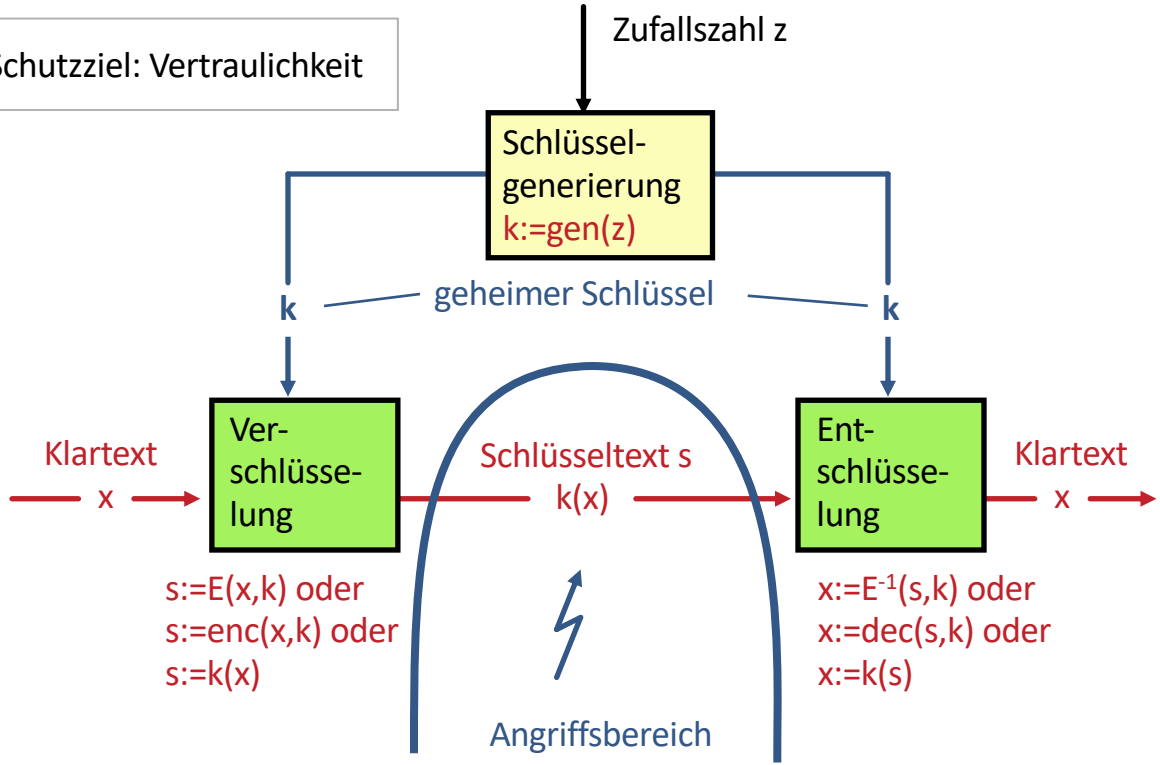


- Steganographie und Watermarking sind nicht erkennbar



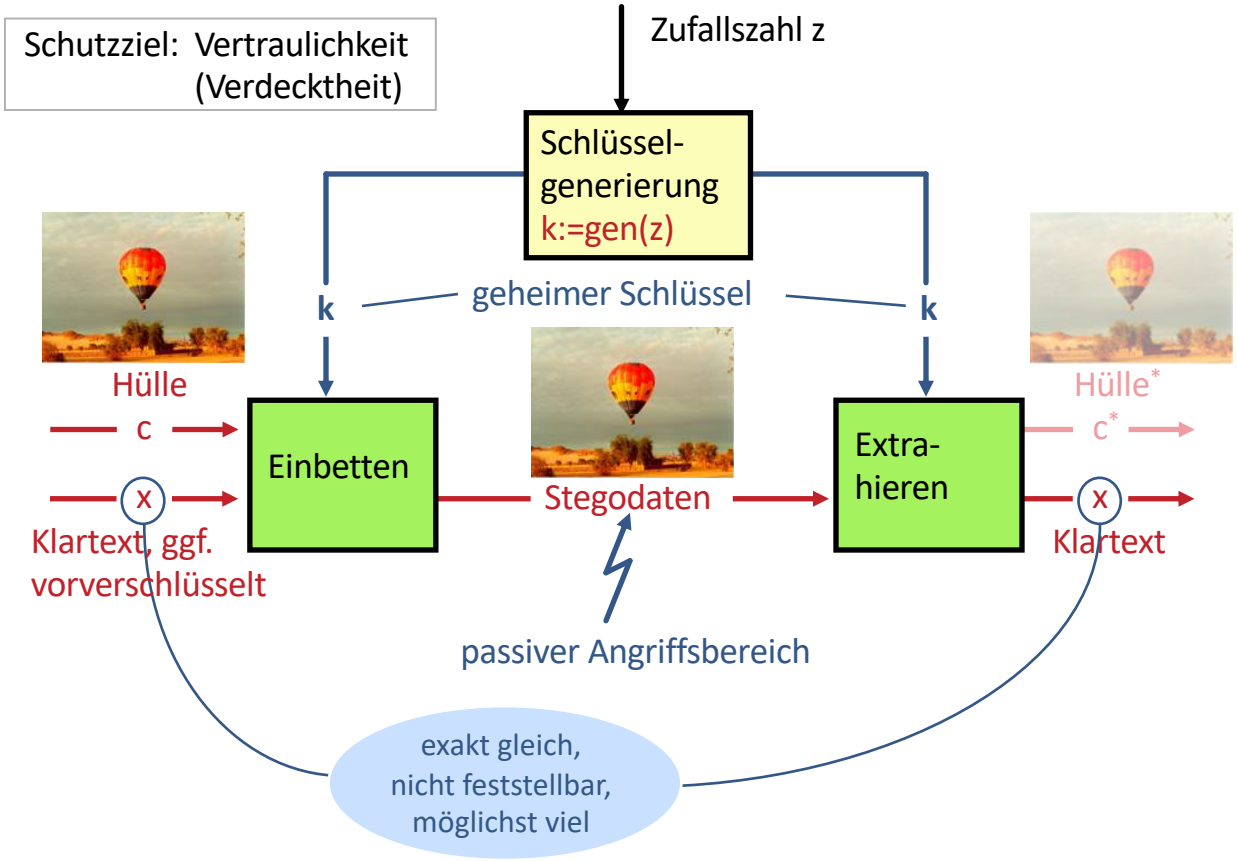
Symmetrische Verschlüsselung

Schutzziel: Vertraulichkeit



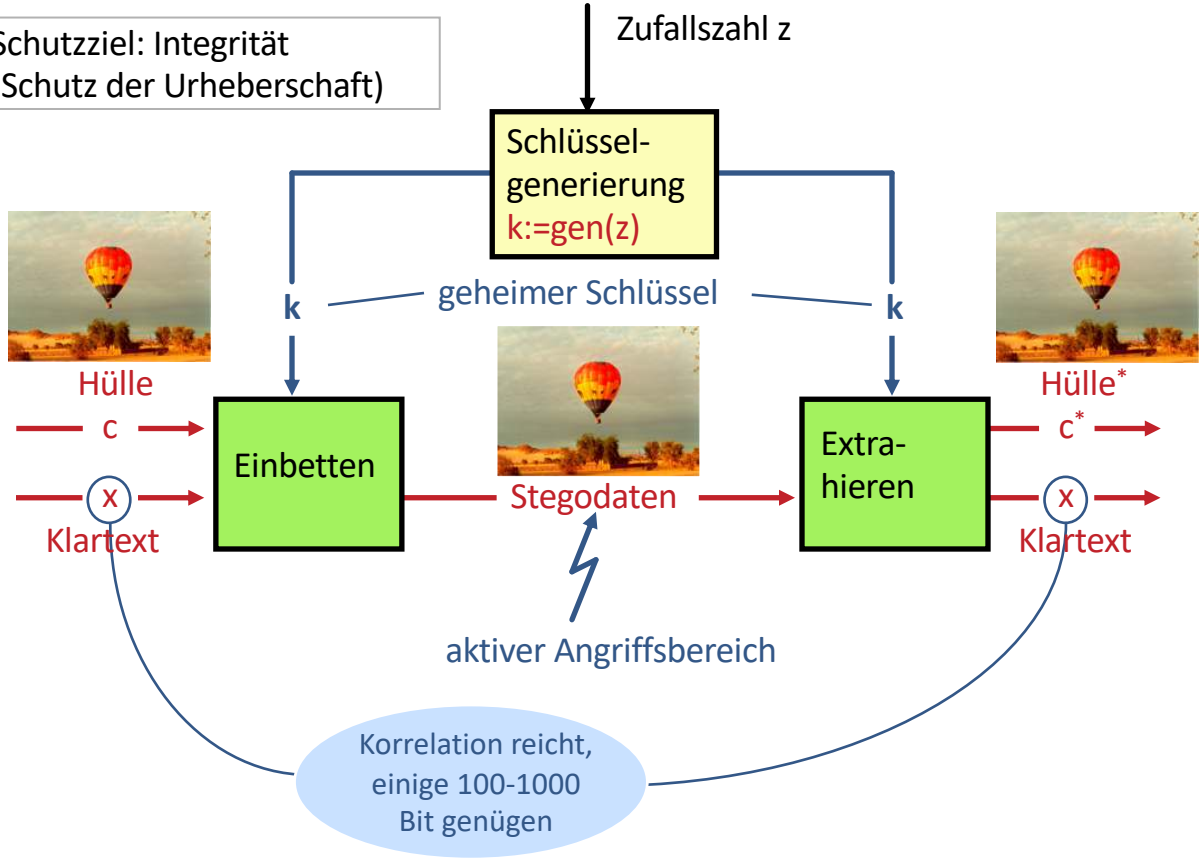
»Undurchsichtiger Kasten mit Schloss. Es gibt zwei gleiche Schlüssel.«

Steganographie



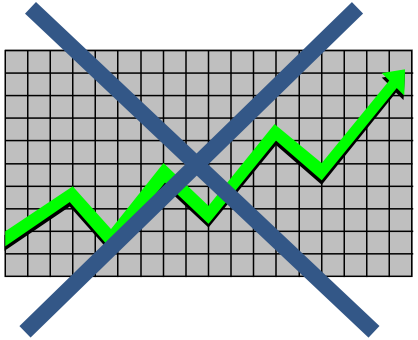
Watermarking

Schutzziel: Integrität
(Schutz der Urheberschaft)



Güte steganographischer Systeme?

- Was zeichnet gute steganographische Systeme aus?
  - Algorithmus ist vollständig offengelegt
  - Parametrisierung durch steganographischen Schlüssel
  - Original der Hülle unwiederbringlich vernichten
  - Hülle niemals ein zweites Mal verwenden
- Ausnutzung indeterministischer Prozesse
  - z.B. natürliches Rauschen bei der Analog-Digital-Umsetzung
  - Einbetten in künstliche Computergrafiken vermeiden



Güte steganographischer Systeme?

- Brechen steganographischer Systeme ist zweistufig
  - Erkennen, dass etwas eingebettet wurde
  - Ermitteln, was eingebettet wurde

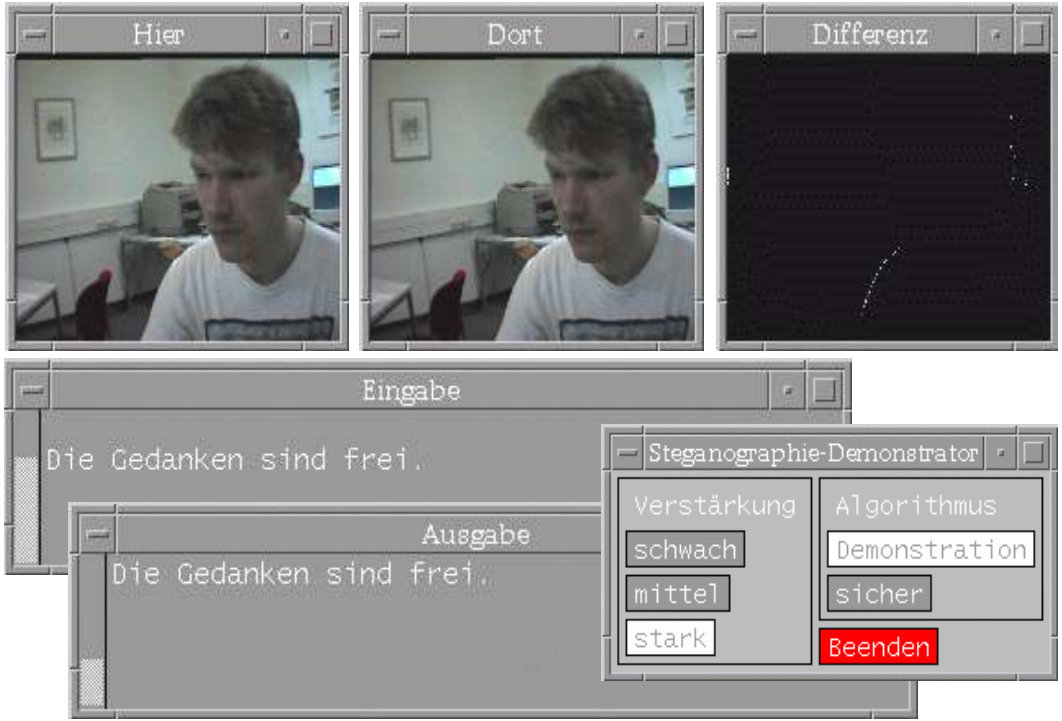
- Ein beweisbar sicheres steganographisches Verfahren existiert bisher nicht.

- Einbettungsverhältnis
  - in Videos: einige 10 kbit/s (z.B. komprimiertes Telefongespräch)
  - in Audio: einige hundert Bit/s
  - in Bildern: ca. 1% der Größe der Hülle

- Kryptoreglementierung wird auf Dauer zu besseren Stegosystemen führen.

Steganographie in Videokonferenzen

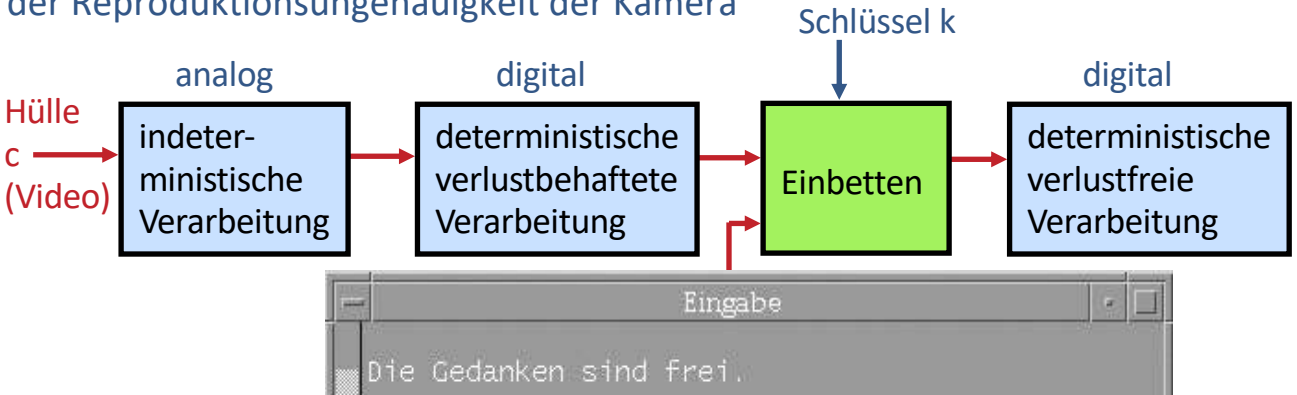
- Flackern an Konturen des Videobildes ist zufällig



Steganographie in Videokonferenzen

- Flackern an Konturen des Videobildes ist zufällig

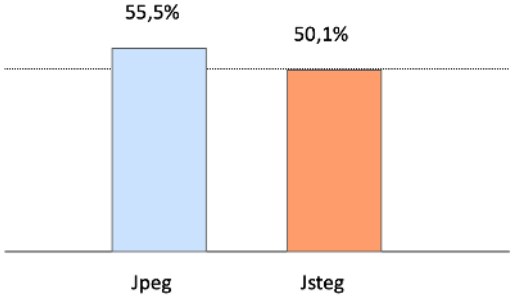
- Ausnutzen der Reproduktionsungenauigkeit der Kamera



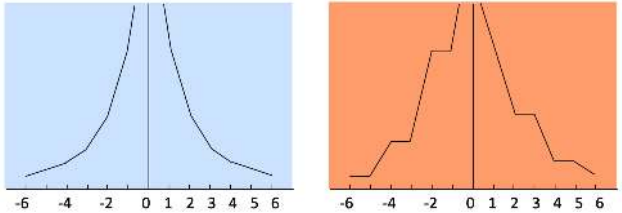
Angriffe auf Jsteg (basiert auf Jpeg-Kompression)

- Visuelle Analyse
  - liefert keine Anhaltspunkte

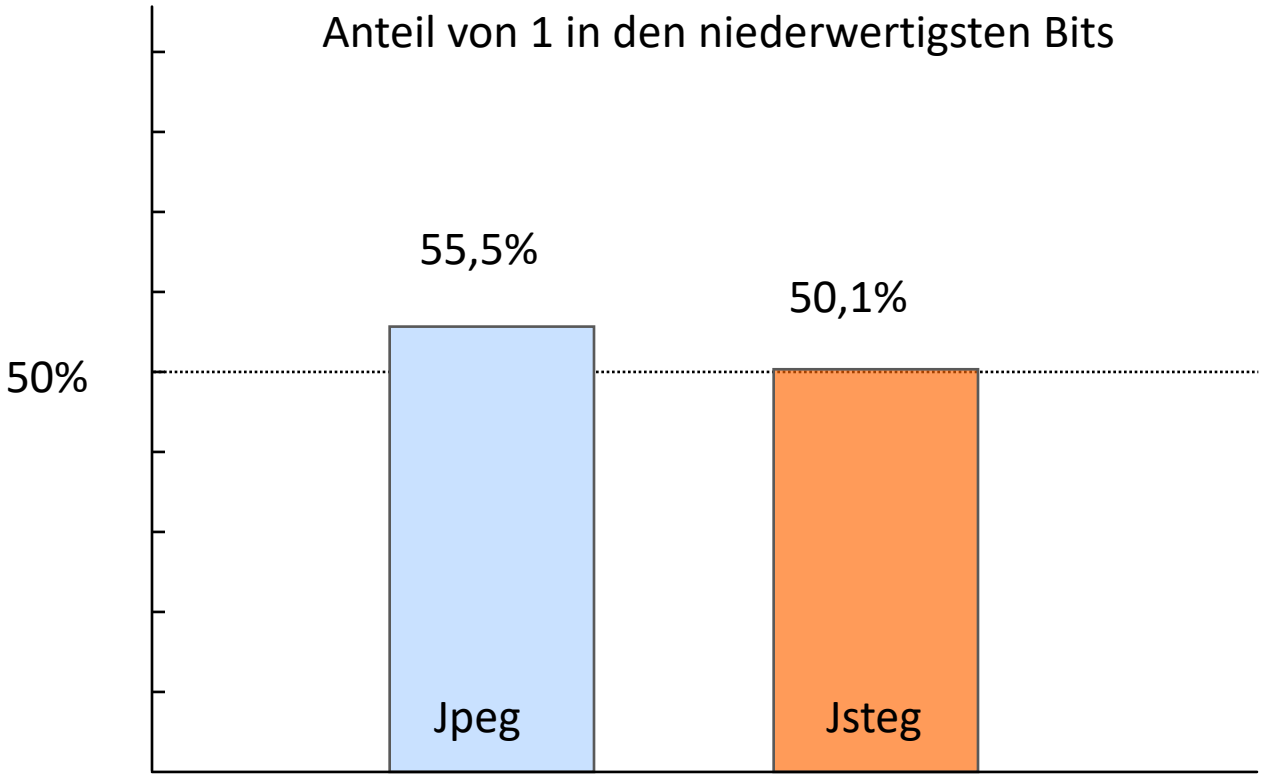
- Verteilung der niederwertigsten Bits
  - liefert in ungünstigen Fällen Verdacht



- »Treppenangriff«
  - deckt Verwendung von Steganographie auf
  - führt jedoch nicht unmittelbar zum Aufdecken der geheimen Nachricht



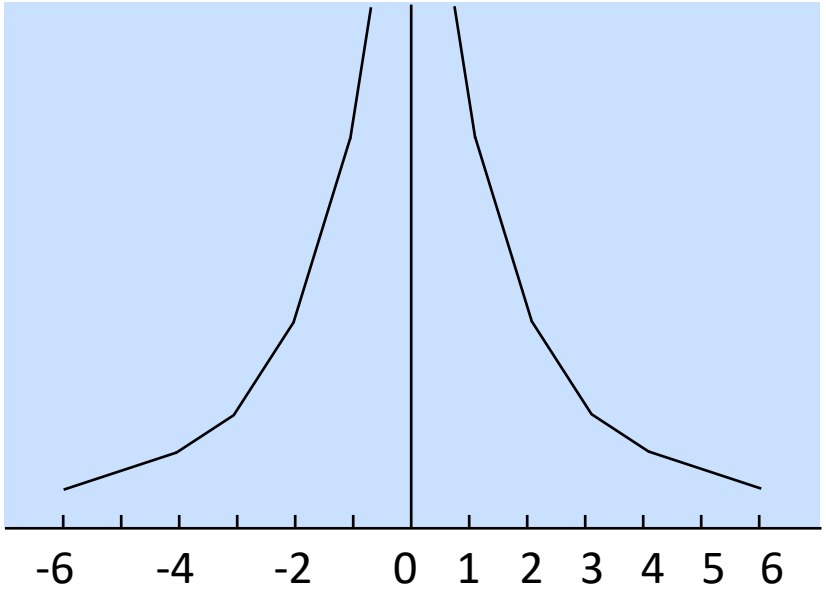
Verteilung der niederwertigsten Bits



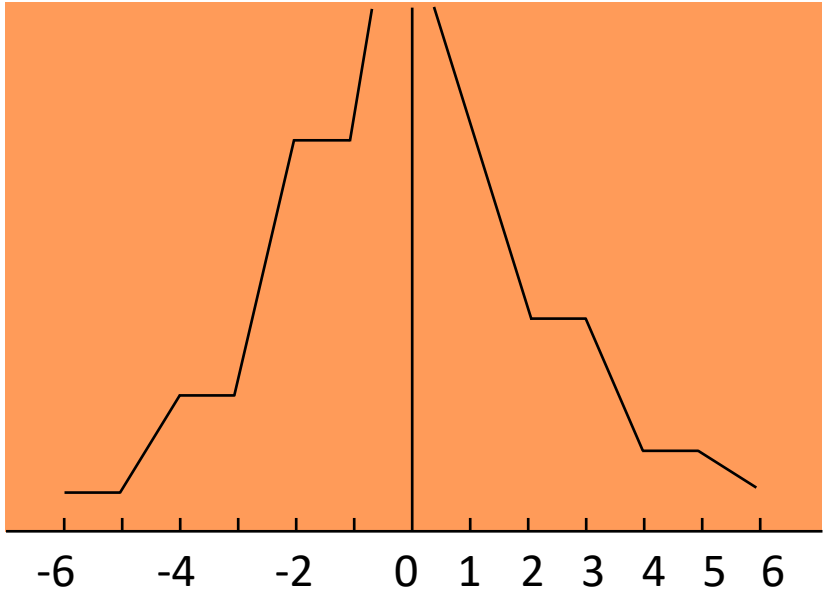
Treppenangriff

Häufigkeit der Koeffizienten der Diskreten Kosinustransformation

Natürliche Verteilung bei Jpeg



Entartete Verteilung bei Jsteg



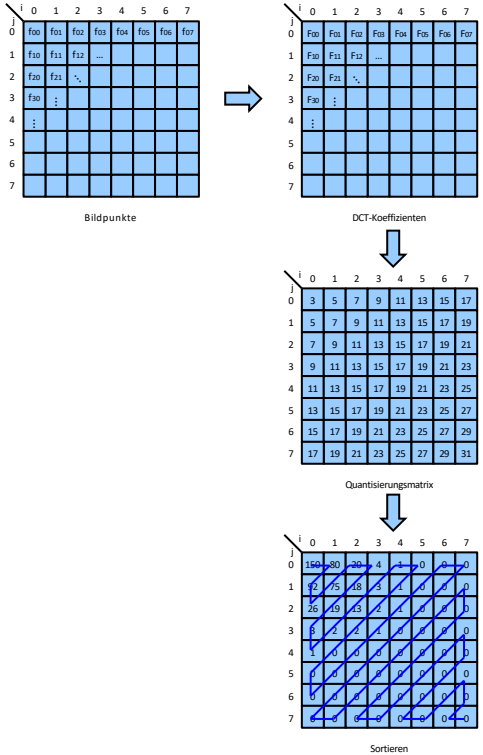
Es entstehen »Treppenstufen«

VSS 17.04.2020

Schritte bei der JPEG-Kompression

Exkurs

1. Farbraumanpassung und Indexverschiebung (-128…128)



2. Diskrete Kosinustransformation (DCT)

- Umwandlung der Ort-Wert-Zuordnung in eine Frequenz-Amplitudenzuordnung

3. Quantisierung

- Koeffizienten mit geringer Bedeutung werden stärker quantisiert

4. Sortierung

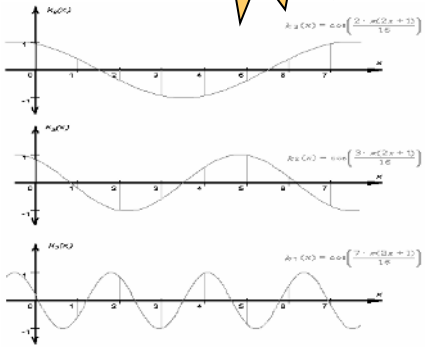
- Koeffizienten mit geringer Bedeutung landen hinten und »beieinander«

5. Kodierung

Diskrete Kosinustransformation (DCT)

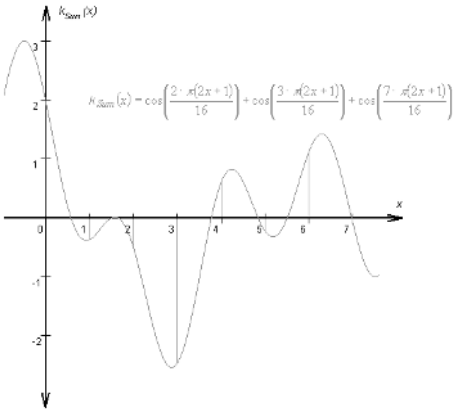
Exkurs

- Pixelwerte werden als Überlagerung von Kosinusfunktionen verschiedener Frequenzen gesehen
- Umwandlung der Ort-Wert-Zuordnung in eine Frequenz-Amplitudenzuordnung



Beispiel:

Überlagerung von
3 Kosinusfunktionen mit
verschiedenen Frequenzen,
hier zweidimensional



Diskrete Kosinustransformation (DCT)

Exkurs

- Einzelne Bearbeitung von 8x8-Matrizen
- Da scharfe Linien und Kanten im Bild selten sind, haben Koeffizienten höherer Frequenzen geringere Werte
- theoretisch noch nicht verlustbehaftet

$$F_{x,y} = \frac{C(x) \cdot C(y)}{4} \cdot \sum_{i=0}^{7} \sum_{j=0}^{7} f_{i,j} \cdot \cos\left(\frac{(2i+1) \cdot x \cdot \pi}{16}\right) \cdot \cos\left(\frac{(2j+1) \cdot y \cdot \pi}{16}\right)$$

j \ i	0	1	2	3	4	5	6	7
0	f_{00}	f_{01}	f_{02}	f_{03}	f_{04}	f_{05}	f_{06}	f_{07}
1	f_{10}	f_{11}	f_{12}	…				
2	f_{20}	f_{21}	⋱					
3	f_{30}	⋮						
4	⋮							
5								
6								
7								

Bildpunkte

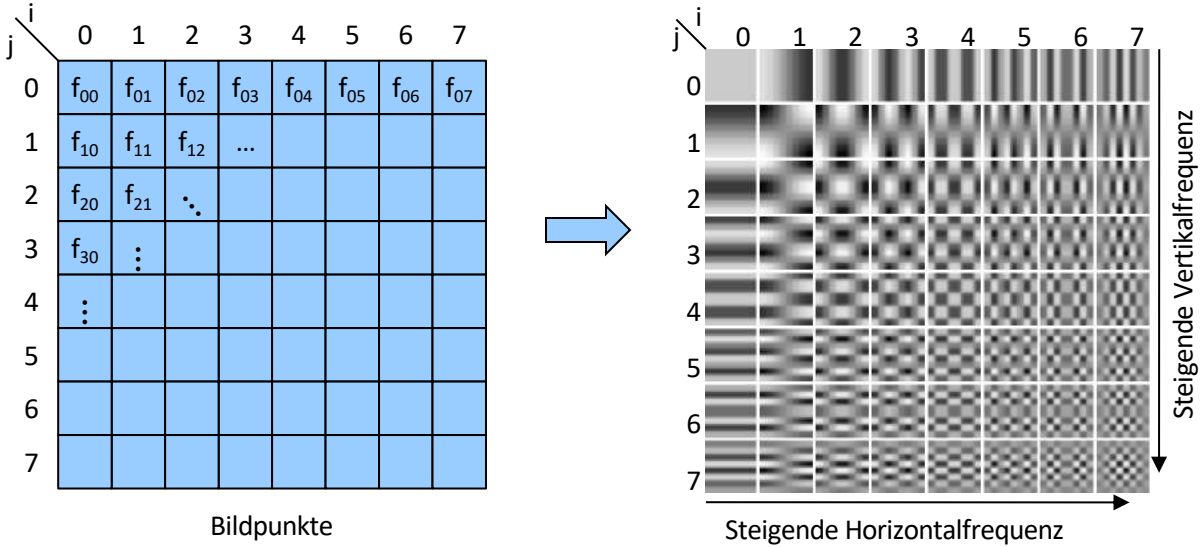
j \ i	0	1	2	3	4	5	6	7
0	F_{00}	F_{01}	F_{02}	F_{03}	F_{04}	F_{05}	F_{06}	F_{07}
1	F_{10}	F_{11}	F_{12}	…				
2	F_{20}	F_{21}	⋱					
3	F_{30}	⋮						
4	⋮							
5								
6								
7								

DCT-Koeffizienten

Diskrete Kosinustransformation (DCT)

Exkurs

- DCT-Basisbilder $B_{j,i}$
  - Koeffizient $F_{j,i}$ gibt Faktor an, mit dem das jeweilige Basisbild zum Gesamtspektrum hinzuaddiert wird.



Quantisierung und Sortierung

Exkurs

- Quantisierung: Werte $F_{j,i}$ der DCT-Matrix werden durch Werte einer je nach Kompression zu wählenden Quantisierungsmatrix geteilt
- Dadurch Berücksichtigung der unterschiedlichen Wahrnehmung verschiedener Frequenzen möglich
- Sortierung mit »zig-zag-scan«
- Ketten von Nullen ermöglichen Kompression

j \ i	0	1	2	3	4	5	6	7
0	3	5	7	9	11	13	15	17
1	5	7	9	11	13	15	17	19
2	7	9	11	13	15	17	19	21
3	9	11	13	15	17	19	21	23
4	11	13	15	17	19	21	23	25
5	13	15	17	19	21	23	25	27
6	15	17	19	21	23	25	27	29
7	17	19	21	23	25	27	29	31

Quantisierungsmatrix

j \ i	0	1	2	3	4	5	6	7
0	150	80	20	4	1	0	0	0
1	92	75	18	3	1	0	0	0
2	26	19	13	2	1	0	0	0
3	3	2	2	1	0	0	0	0
4	1	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Sortieren