

# Informatik II

Dr. Henrik Brosenne  
Georg-August-Universität Göttingen  
Institut für Informatik

Sommersemester 2012

## Telematik

### Einführung

Leitungs- und Paketvermittlung

Schichtenarchitektur

Dienste und Protokolle

## Kryptographie

J. F. Kurose, K. W. Ross.

*Computernetze: Ein Top-Down-Ansatz mit Schwerpunkt Internet*,  
Pearson Studium (Addison-Wesley), August 2008.

A. S. Tanenbaum.

*Computernetzwerke*,  
Pearson Studium (Prentice-Hall), 2003.

William Stallings.

*Data & Computer Communications*,  
Prentice Hall, 2007.

<http://williamstallings.com/DCC6e.html>

Warriors of the net movie.

<http://www.warriorsofthe.net/movie.html>

# Begriffe

**Telematik** = **TELE**kommunikation und Infor**MATIK**, geprägt von Simon Nora und Alain Minc in Ihrem Bericht an den französischen Präsidenten, 1978.

Drückt die Verflechtung von Rechnern und Telekommunikationsmitteln aus.

Andere Bezeichnungen für Telematik.

- Computernetze
- Computernetzwerke
- Rechnernetze
- Kommunikationsnetze

Aufgabengebiete

- Praktische, Angewandte und Technische Informatik.
- Technische Infrastruktur verteilter Systeme (Netze).
- Netz-Dienste und darauf aufbauende Anwendungen.
- Regeln für Nachrichtenaustausch (Protokolle).
- Werkzeuge zum Entwickeln verteilter Anwendungen.

# Was ist das *Internet* (1/3)

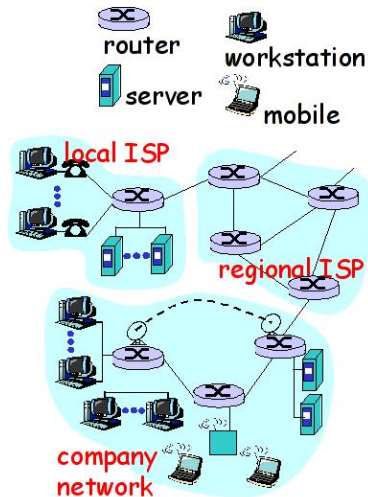
Sehr viele (Millionen) von verbundenen Rechnern **End-Geräte** (*hosts*) sind Geräte auf denen **Netzwerkanwendung** laufen.

- PCs, Workstations, Server.
- PDAs, Telephone, Toaster

## Kommunikationsverbindungen

- Kabel (Kupfer, Fiberglas, etc.), Funk (Hochfrequenz, Satellit, etc.).
- Übertragungsrate = Bandbreite (*bandwidth*).

**Vermittlungseinheiten** (*switching nodes*) schalten die Kommunikationsverbindungen zusammen.



Quelle: Kurose & Ross *Computernetze: Ein Top-Down-Ansatz mit Schwerpunkt Internet*

# Was ist das *Internet* (2/3)

**Protokolle** regeln Senden und Empfangen von Nachrichten.

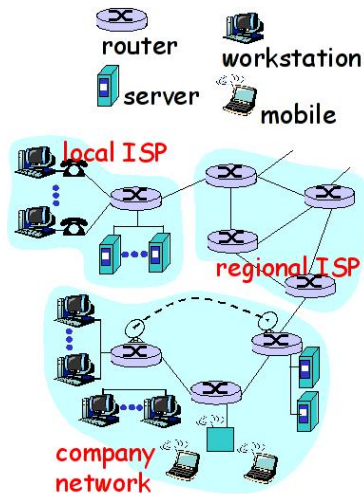
- PPP (*Point-to-Point Protocol*)
- PPPoE (*PPP over Ethernet*)
- IP (*Internet Protocol*)
- TCP (*Transmission Control Protocol*)
- HTTP (*Hypertext Transport Protocol*)
- FTP (*File Transfer Protocol*)

Internet, *Netzwerk von Netzwerken*

- Schwach hierarchisch Strukturiert.
- Öffentliches Internet steht dem privaten Intranet gegenüber.

Internet **Standards**

- IETF (*Internet Engineering Task Force*) veröffentlicht RFCs (*Request For Comments*).
- IEEE (*Institute of Electrical and Electronics Engineers*).



# Was ist das *Internet* (3/3)

## Kommunikationsinfrastruktur

ermöglicht verteilte Anwendungen

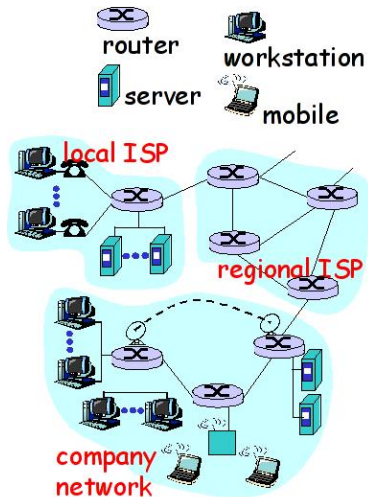
- Web, EMail, Spiele, E-Commerce, File-Sharing (z.B. MP3).

**Kommunikationsdienste** stehen den Netzwerkanwendungen zur Verfügung.

- Verbindungslos (*connectionless*).
- Verbindungsorientiert (*connection-oriented*).

**Cyberspace** von William Gibson (Neuromancer, 1984).

- *a consensual hallucination experienced daily by billions of operators, in every nation, ...*



Quelle: Kurose & Ross *Computernetze: Ein Top-Down-Ansatz mit Schwerpunkt Internet*

## Telematik

Einführung

**Leistungs- und Paketvermittlung**

Schichtenarchitektur

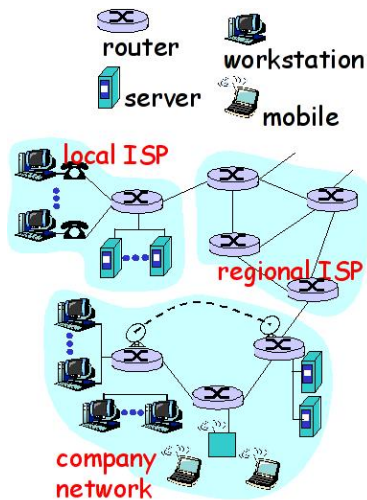
Dienste und Protokolle

## Kryptographie



**Vermittlung** ist der Mechanismus zum Zusammenschalten der Kommunikationsverbindungen (Übertragungskanäle).

- Leitungsvermittlung
- Paketvermittlung



Quelle: Kurose & Ross *Computernetze: Ein Top-Down-Ansatz mit Schwerpunkt Internet*

# Leitungsvermittlung

**Leitungsvermittlung** (*circuit switching*) ist die Einrichtung einer festen und exklusiven Verbindung der Kommunikationspartner durch das Kommunikationsnetz für die Dauer der Verbindung.

Phasen einer vermittelten Verbindung.

- **Verbindungsaufbau.** Auswahl der Kommunikationspartner, Reservierung der benötigten Betriebsmittel (Kanäle, Bandbreiten, usw.), Feststellen der Kommunikationsbereitschaft der Endeinrichtung, Herstellen der Verbindung.
- **Informationsübertragung.** Transparent basierend auf den Anforderungen beim Verbindungsaufbau und den Möglichkeiten des Netzes.
- **Verbindungsabbau.** Freigabe der verwendeten Betriebsmittel.

**Signalisierung.** Austausch von Informationen zum Auf- und Abbau der Verbindungen.

**Dienstgüte** ist sichergestellt. Qualität der Sprachübermittlung, Verfügbarkeit des Netzes, Zuverlässigkeit der Übertragung, Bitrate, Laufzeit, etc.

# Paketvermittlung (1/2)

Bei Datenkommunikation sind Pausen typisch, Leitungsvermittlung ist daher ineffizient.

**Paketvermittlung** (*packet switching*) bedeutet, dass die Daten in Datenpakete zerlegt und einzeln übertragen werden.

Dateneinheiten beinhalten Kontrollinformationen (*header*).

**Statistisches Multiplexen.** Pakete werden in den Vermittlungseinheiten in Puffern zwischengespeichert.

- Vor dem Weiterleiten können sich Wartezeiten ergeben.
- Puffer können überlaufen.

**Best Effort.** Dienstgüte ist zunächst nicht sichergestellt, hierfür sind weitere Maßnahmen notwendig.

# Paketvermittlung (2/2)

## **Datagramm-orientierte** Paketvermittlung

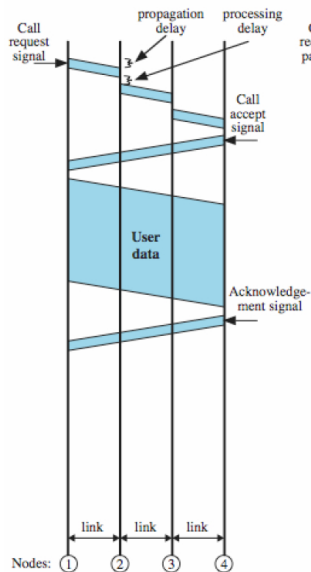
- Jedes Paket enthält eindeutige Zieladresse, aufgrund der die Vermittlungseinheit es weiterleitet.
- Pakete können unterschiedliche Pfade durchlaufen (ist aber untypisch).

## **Virtuelle Leitungsvermittlung** (*virtual circuit switching*)

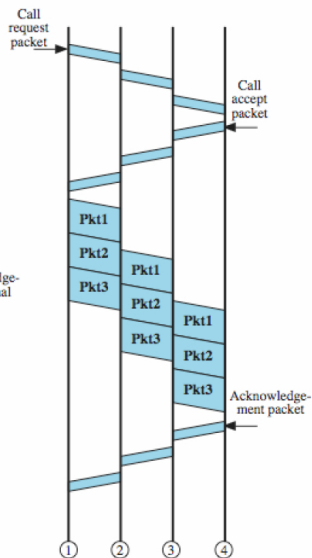
- Vor dem Übertragen von Paketen wird ein Pfad bestehend aus Vermittlungseinheiten als **virtuelle Verbindung** festgelegt.
- Jedes Paket wird über diesen Pfad geleitet, aber immer noch in den Vermittlungseinheiten zwischengespeichert.

# Beispiel

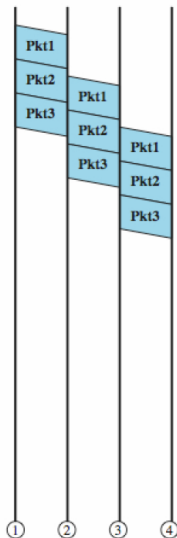
(a) Circuit switching



(b) Virtual circuit packet switching



(c) Datagram packet switching



## Telematik

Einführung

Leistungs- und Paketvermittlung

**Schichtenarchitektur**

Dienste und Protokolle

## Kryptographie

## ISO-OSI Referenzmodell

- *International Organization for Standardization (ISO)*.
- *Open Systems Interconnection (OSI)*.
- Liefert gute begriffliche Abstraktionen

**Dienste** (*services*) werden von Kommunikationssystem bereitgestellt, um Daten, Programme, und Dokumente auszutauschen bzw. abzurufen und Programme entfernt oder in Kooperation auszuführen.

Der Dienst beschreibt **was** zu Verfügung gestellt wird.

Unterscheidung nach verschiedenen Dienstarten.

- Symmetrisch. Dienste werden an zwei oder mehreren Zugangspunkten gleichzeitig angeboten, z.B. Dienste zum Austausch von Daten.
- Asymmetrisch. Z.B. beim Client/Server-Modell.
- Bestätigt/unbestätigt.
- Verbindungsorientiert/verbindungslos.



# Dienstmodell

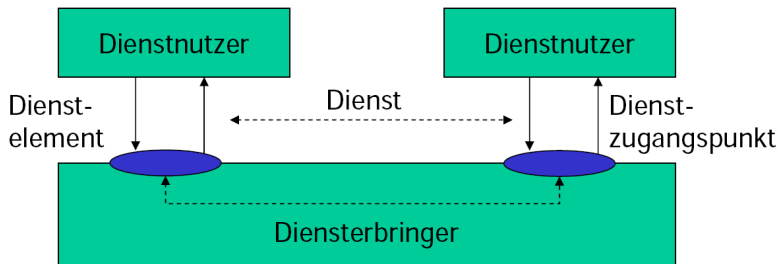
Dienstanutzer (*service user*)

Diensterbringer (*service provider*)

Dienstschnittstelle (*service interface*)

Dienstzugangspunkt (*service access point, SAP*)

Dienstdateneinheiten (*service data units, SDU*)



# Dienstelemente

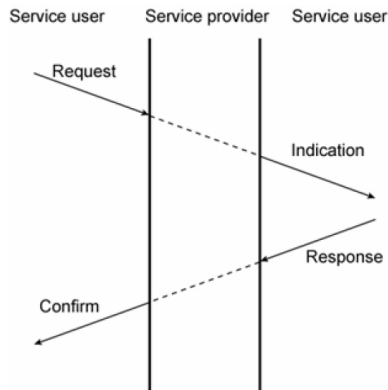
**Dienstelemente** (*service primitives*) sind die Abstraktionen für die Beschreibung der Wechselwirkung zwischen Dienstanutzer und Dienstbringer.

Der Dienstanutzer kennt die Dienstschnittstelle und tauscht, geregelt durch die Dienstelemente, Dienstdateneinheiten mit dem Dienstzugangspunkt des Dienstbringers aus.

## Beispiel

OSI-Notation *Name, Typ, Parameter*

- CONNECT request(CalleeAddr, CallerAddr, QoSParam, UserData)
- DATA indication(UserData)
- Namen. CONNECT, DISCONNECT, DATA, ABORT.
- Typen. request, indication, respond, confirm.



# Protokoll

**Instanzen** (*entities*). Aktive Objekte des Dienstbringers, die mit ihrer Umgebung durch den Austausch von Nachrichten interagieren.

**Protokoll.** Verhaltenskonvention, die die zeitliche Abfolge der Interaktionen zwischen den dienstbringenden Instanzen vorschreibt und die Formate (Syntax und Semantik) der austauschenden Nachrichten definiert.

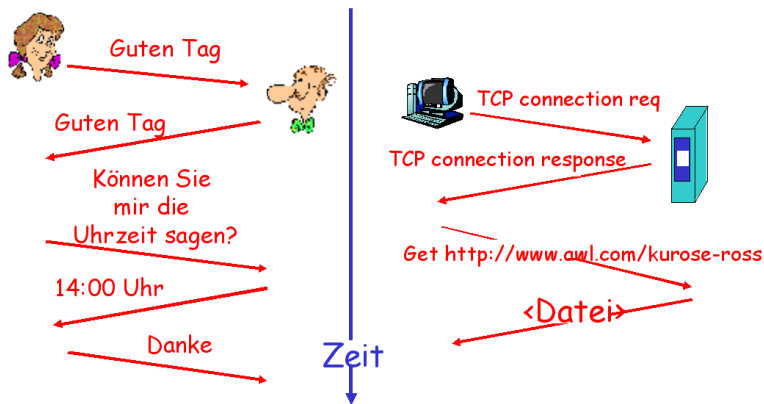
Ein Protokoll beschreibt **wie** ein Dienst zur Verfügung gestellt wird.

Ein Protokoll legt Folgendes fest.

- Beschreibung des bereitgestellten **Dienstes**.
- **Vokabular** und **Kodierungsformat** der Nachrichten.
- **Bedeutung** der Nachrichtfelder.
- **Verhalten** der Instanzen beim Eintreffen von Nachrichten (funktional und zeitlich).
- Annahmen über die **Umgebung**, in der das Protokoll abläuft.

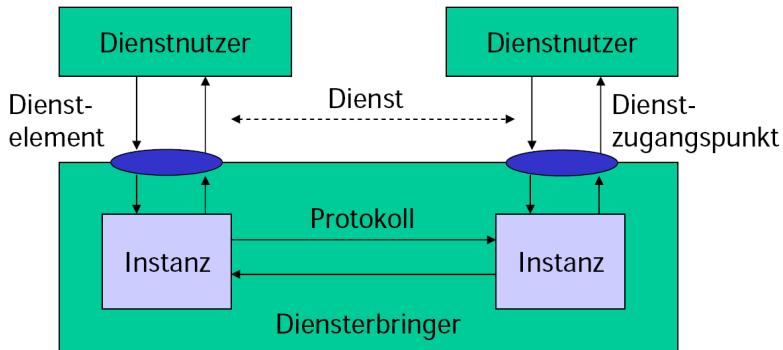
# Beispiel

Menschliches und Computernetzwerk Protokoll.



Quelle: Kurose & Ross *Computernetze: Ein Top-Down-Ansatz mit Schwerpunkt Internet*

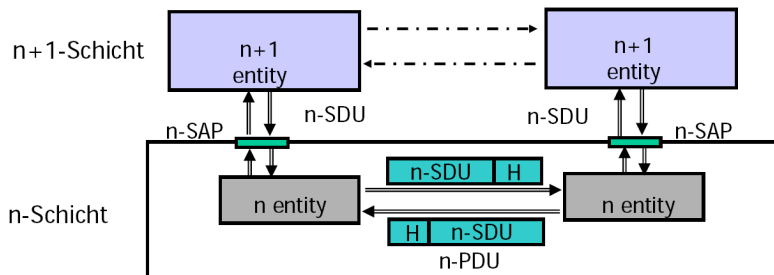
# Protokoll im Dienstmodell



# Schicht

## Schicht (*layer*)

- Umfasst alle Instanzen, die die gleichen Dienste bereitstellen.
- Für die Kommunikation der Instanzen untereinander (d.h. zur Protokollausführung) nutzen die Instanzen die Dienste der darunter liegenden Schicht.
- Eine Instanz ist zugleich Diensterbringer und Dienstanwender.
- Das Schichtenprinzip ist ein wichtiges Gestaltungsprinzip.
- **Partner-Instanzen** (*peer entities*). Instanzen der gleichen Schicht mit Kommunikationsbeziehung.



# Schichtenarchitektur

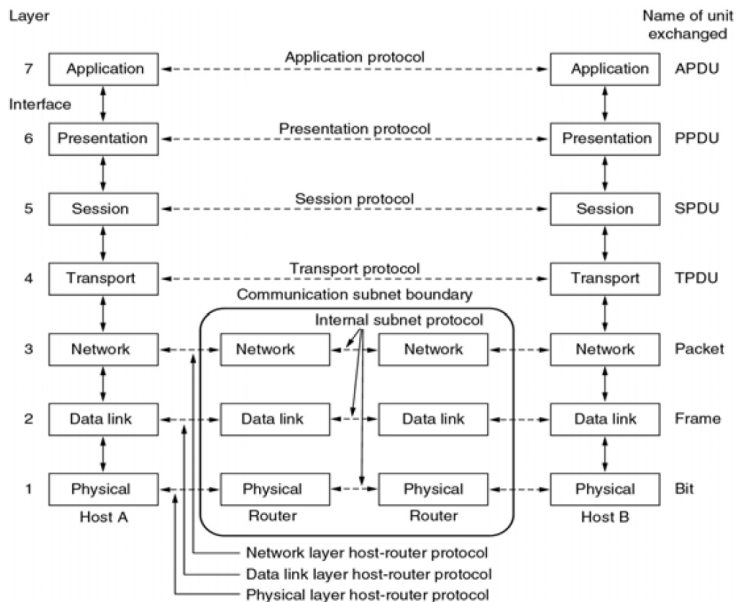
## Schichtenarchitektur

- Abgestimmte Architektur von Schichten, in die die einzelnen Kommunikationsprotokolle eingebettet werden.
- Definiert Funktionalität der einzelnen Schichten und legt die Prinzipien der Interaktion zwischen ihnen fest.
- In der Regel durch Standardisierungsgremien oder Firmenkonsortien festgelegt.

## ISO-OSI Referenzmodell

- Aufteilung in 7 Schichten.
- Gedankenmodell, Implementierungen haben sich nicht durchgesetzt.

# ISO-OSI Referenzmodell





## **Bitübertragungsschicht** (*Physical Layer, Layer 1*).

Mechanische, elektrische und prozedurale Eigenschaften zur Übertragung von **Bits**.

- Zeitsynchronisation
- Kodierung
- Modulation
- etc.

## **Verbindungs-** oder **Sicherungssicht** (*Data Link Layer, Layer 2*)

Gesicherte (weitgehend fehlerfreie) Übertragung von **Rahmen** (*frames*) durch Folgenummern, Prüfsummen, Quittungs- und Wiederholungsmechanismen.

- Rahmensynchronisation
- Fehlerkontrolle
- Flußkontrolle
- etc.

# Vermittlungs- und Transportschicht

## **Vermittlungsschicht** (*Network Layer, Layer 3*)

Übertragung von Paketen bzw. Datagrammen.

- Verbindungsaufbau
- Wegewahl
- Vermittlung
- Betriebsmittelverwaltung
- etc.

## **Transportschicht** (*Transport Layer, Layer 4*)

Ende-zu-Ende Transport von Segmenten.

Festlegen der virtuelle Verbindung, wenn virtuelle Leitungsverbindung angeboten wird.

## **Sitzungsschicht** (*Session Layer, Layer 5*)

Kommunikation zwischen Anwendungen.

## **Darstellungsschicht** (*Presentation Layer, Layer 6*)

Syntax und Semantik der ausgetauschten Informationen, z.B. mit Abstract Syntax Number One (ASN.1) oder XML.

## **Anwendungsschicht** (*Application Layer, Layer 7*)

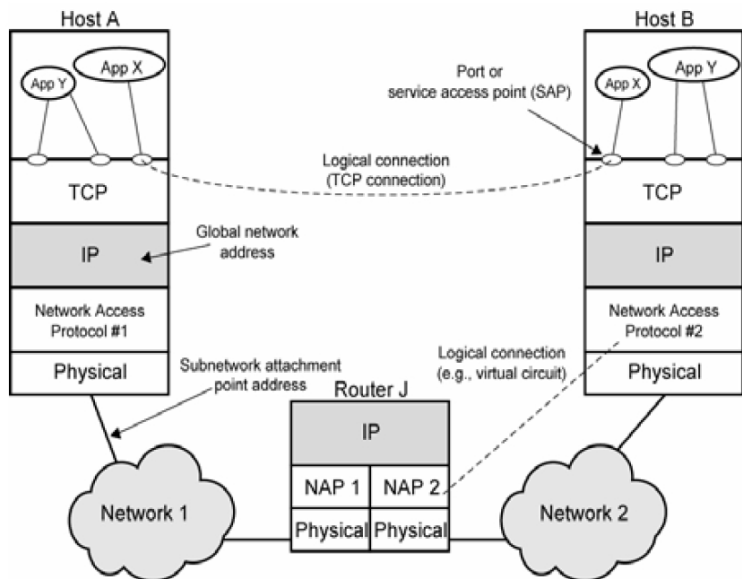
Kommunikation der Anwendungsprozesse mit anwendungsspezifischen Informationen.

# ISO-OSI Modell

7	Anwendungsschicht	<i>Application Layer</i>
6	Darstellungsschicht	<i>Presentation Layer</i>
5	Sitzungsschicht	<i>Session Layer</i>
4	Transportschicht	<i>Transport Layer</i>
3	Vermittlungsschicht	<i>Network Layer</i>
2	Verbindungsschicht	<i>Data Link Layer</i>
1	Bitübertragungsschicht	<i>Physical Layer</i>

# TCP-IP Modell

Im Internet hat sich eine vereinfachte Schichtenarchitektur durchgesetzt.



# Transportschicht

## TCP (*Transmission Control Protocol*)

- zuverlässig
- verbindungsorientiert
- paketvermittelt

## UDP (*User Datagram Protocol*)

- nicht zuverlässig
- verbindungslos
- paketvermittelt

## IP (*Internet Protocol*)

- netzübergreifende Adressierung
- nicht zuverlässig
- verbindungslos
- paketvermittelt

## ARP (*Address Resolution Protocol*)

- Zuordnung von Netzwerkadressen zu Hardwareadressen



# Verbindungsschicht und Bitübertragungsschicht

## Ethernet (IEEE 802.3)

- paketvermittelt
- Hardwareadressen
- Protokoll CSMA/CD *Carrier Sense Multiple Access with Collision Detect* (Verbindungsschicht)
- Pakete = Rahmen (Frames) (Verbindungsschicht)
- beschreibt die Signalisierung (physikalische Sicht = Bitübertragungsschicht)

## Telematik

Einführung

Leistungs- und Paketvermittlung

Schichtenarchitektur

Dienste und Protokolle

## Kryptographie

# Ethernet

Idee. *Funksystem* für Kabel

- Kommunikationsteilnehmer teilen ein Kabel.
- Jeder Teilnehmer kann alles mithören.

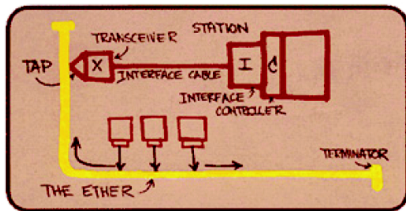
Namensgebung an Äther (*ether*) angelehnt.

Eindeutige Kennung der Teilnehmer.

Basisband-Übertragung. Gesamtes Frequenzspektrum des Mediums wird für eine Übertragung genutzt.

Zeit-Multiplex-Verfahren. In bestimmten Zeitabschnitten (Zeitschlitzen) werden die Frames verschiedener Sender auf einem Kanal übertragen.

Kollisionen können auftreten.



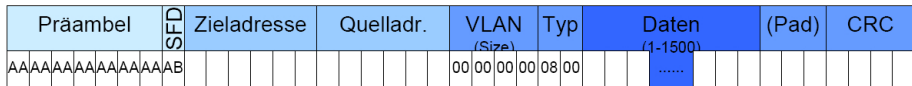
# Ethernet-Frame

## Pakete/Frames allgemein

- kleine Datenmenge (*payload*)
- plus Zusatzinformation (*Header*)

## Ethernet-Frame

- Mindestlänge von 64 Byte
- Präambel 56 Bit. Abwechselnd 0 und 1 zur Synchronisation
- SFD *Start of Frame Delimiter*
- Ziel- und Quell-Adresse.
  - ▶ 48-Bit MAC-Adresse (*Media Access Control*)
  - ▶ jede Netzwerkkarte hat eindeutige MAC-Adresse (Hersteller-Prefix)
- Längen oder VLAN-Feld
- Paket-Typ (z.B. 0x0800 = IP)
- PAD (padding). Wenn nötig Auffüllen auf Mindestlänge.
- Prüfsumme CRC (*Cyclic Redundancy Check*)



IP

# Netzwerkkarte

## Network Interface Card (NIC)

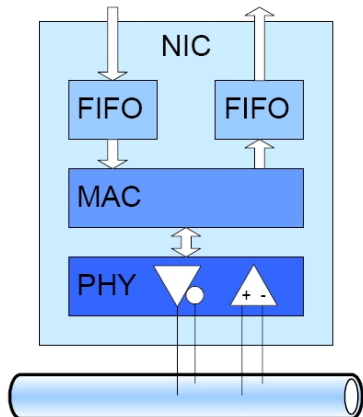
- Schnelle serielle Schnittstelle (10-1000 Mbit/s).
- FIFO-Buffer für Ein- und Ausgabe.
- Paket-orientierter Datentransfer, DMA für Datentransfer von und ins Memory.

## Media-Access Controller

- Steuert Zugriff auf Physical Layer.
- Framing des Pakets.
- Prüfsumme für Daten.

## Physical Layer

- Elektrischer Standard für Übertragung.



## Next-Hop Routing

- Ethernet ist ein Protokoll der Verbindungsschicht (*data link layer*) und der Bitübertragungsschicht (*physical layer*).
- Frames werden nur von einem Interface zum nächsten geleitet.

Ethernet kann keine Pakete routen, das passiert auf anderer Netzwerke-Ebene.

Ziel-(Ethernet)-Adresse wird z.B. durch das ARP (*Address Resolution Protocol*) auffindig gemacht.

# ARP

Zum Ausfindigmachen der MAC-Adresse

- Netzwerk-Daten-Verkehr auf IP-Ebene benötigt Routing-Information auf Ethernet-Ebene.
- An welche MAC-Adresse muss das Paket geschickt werden?

ARP-Broadcast

- Wer kennt die IP-Adresse `www.xxx.yyy.zzz`?
- ARP-Protokoll. Ethernet-Frame-Typ 0x0806
- ARP-Broadcast an alle Subnetz-Rechner  
Zieladresse `FF:FF:FF:FF:FF:FF` (=Broadcast).

ARP-Antwort

- Ein (oder mehrere) Host antworten.
- Adresse des Antwort-Hosts ist gesuchte Next-Hop-Adresse.
- Host muss nicht Zielrechner sein, sondern kann auch Gateway (Router) sein.

# Beispiel

## Windows und Linux

- Speichern MAC-Adressen in ARP-Cache.
- Einträge sind eine Zeit lang (z.B. 20 Minuten) gültig.

## Windows

---

```
C: \> arp -a
Interface: 192.168.1.101 --- 0x80004
   Internet Address      Physical Address      Typ
   192.168.1.1          00-0d-6d-bc-a8-6b    dynamic
   192.168.1.2          00-0e-1c-2b-e5-3c    dynamic
```

---

## Linux

---

```
$ sudo arp
Address                HWtype  HWaddress          Iface
192.168.1.1            ether   00:0D:6D:BC:A8:6B  eth0
192.168.1.2            ether   00:0E:1C:2B:E5:3C  eth1
```

---



## IP (*Internet Protokoll*)

- Routet Daten-Pakete durch das Internet.
  - ▶ Globale Zustellung von Daten-Paketen.
- Benötigt eindeutige Netzwerkadresse
  - ▶ 32-Bit IP-Adresse (IPv4)
  - ▶ 128-Bit IP-Adresse (IPv6)
- Pakete werden einzeln geroutet (verbindungslos).
- Unabhängig von Übertragungstechnologie.
  - ▶ Protokoll der Verbindungsschicht (z.B. Ethernet) hat andere Adressierung.
- IP ist kein verlässliches Service (*best effort delivery*).
  - ▶ Paketverlust kann auftreten.
  - ▶ Reihenfolge der Pakete kann geändert sein.
  - ▶ Paketdaten können verändert sein.

# IPv4-Adresse

Jedes End-Gerät (*host*) und jede Vermittlungseinheit (*switching node*) im Internet hat eine **IP-Adresse** der Länge 32-Bit, die die **Netz- und Hostnummer** kodiert.

Die Kodierung ist eindeutig, zwei unterschiedlichen Geräte in einem Netz haben unterschiedliche IP-Adressen.

Die IP-Adresse bezieht sich nicht auf das Gerät, sondern auf die Netzwerkschnittstelle.

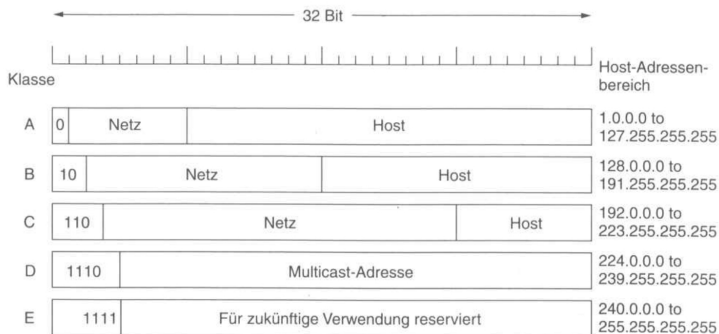
Ist ein Gerät in zwei Netzen verfügbar, muss es zwei IP-Adressen haben.

IP-Adressen werden in fünf Klassen eingeteilt, diese Zuordnung wird als klassenbasierte IP-Adressierung *Classful Addressing* bezeichnet.

## Ausblick

Klassenbasierte IP-Adressierung wird abgelöst durch *Classless InterDomain Routing (CIDR)*.

# Klassenbasierte IP-Adressierung



IP-Adressen sind 32-Bit Zahlen, die in Dezimalzahldarstellung mit Punkten (*Dotted Decimal Notation*) geschrieben werden. In diesem Format werden jeweils 4-Byte dezimal geschrieben (0-255) und durch einen Punkt getrennt.

Die niedrigste Adresse ist 0.0.0.0, die höchste 255.255.255.255. Diese beiden Adressen haben jeweils eine besondere Bedeutung.

Um Konflikte bei der Wahl/Vergabe von Netznummern zu vermeiden, werden diese von der gemeinnützigen Organisation *Internet Corporation for Assigned Names and Numbers (ICANN)* verwaltet.



# IPv4-Paket (IPv4-Datagram)

**Header** (20-60 Byte) mit Big-Endian Darstellung von 32-Bit Wörtern.

- *Version*, 4=IPv4
- *Header Length* in 32-Bit Wörtern.
- *Type of Service* für Quality-of-Service.
- *Total Length*, Header + Daten in Bytes  
20 Byte bis 65535 Bytes
- *Identification, Flags, Fragment Offset*.  
Zur Erkennung von Fragmenten.
- *Time to Live (TTL)*.
- *Protokoll*, 6=TCP, 17=UDP.
- *Quell- und Zieladresse* (32 Bit).
- *Header-Prüfsumme*, 16-Bit Summe.



## Daten

- Maximale Größe  $2^{16} - 1 = 65535$  Byte - Header.

# Time to Live (TTL)

**Time to Live (TTL)** ist die Lebensdauer des Pakets, bei Wert 0 wird das Paket verworfen.

Verhindert, dass Pakete im Fall von Daten- oder Routing-Fehlern fortlaufend (z.B. auf einer Kreisroute) weitergeleitet werden.

Der Absender initialisiert dieses Feld (anhängig von Implementierung/Konfiguration) üblicherweise mit 64, 128 oder 255.

Jeder Router, den das Paket passiert, verringert den TTL-Wert mindestens um Eins.

## Beispiel

Der TTL-Wert kann dazu genutzt werden, grob abzuschätzen, über wie viele Router die Ping-Pakete gelaufen sind.

---

```
ping -c 1 -n localhost
PING 127.0.0.1 56(84) bytes of data.
64 bytes from 127.0.0.1: ttl=64 time=0.025 ms
...
```

---

# Beispiel

---

```
> ping -c 3 www.gwdg.de
PING 134.76.10.47 56(84) bytes of data.
64 bytes from 134.76.10.47: ttl=61 time=0.794 ms
64 bytes from 134.76.10.47: ttl=61 time=0.833 ms
64 bytes from 134.76.10.47: ttl=61 time=0.873 ms

--- ping statistics ---
3 packets, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.794/0.833/0.873 ms
```

---

```
> ping -c 3 www.google.de
PING 72.14.221.99 56(84) bytes of data.
64 bytes from 72.14.221.99: ttl=53 time=17.7 ms
64 bytes from 72.14.221.99: ttl=53 time=17.0 ms
64 bytes from 72.14.221.99: ttl=53 time=15.3 ms

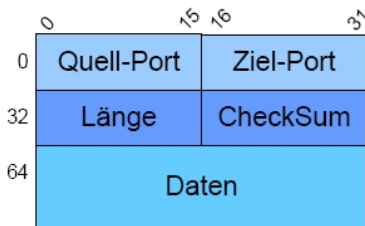
--- ping statistics ---
3 packets, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max = 15.375/16.716/17.736 ms
```

---

# UDP

## UDP (*User Datagram Protocol*)

- Nicht verbindungsorientiert, d.h. kein Verbindungsaufbau nötig.
- *Unreliable datagram protocol*
- Keine Fehlerkontrolle, deshalb nicht für Anwendungen geeignet, die fehlerfreie Übertragung benötigen.
- Wenig Overhead, deshalb gut für Anwendungen geeignet, die schnelle Übertragung benötigen.
- ca. 20% des Internet-Verkehrs





# TCP (1/2)

## TCP (*Transmission Control Protocol*)

- Verbindungsorientiert
  - 1 Aufbau einer Ende-zu-Ende Verbindung.
  - 2 Nutzen der Verbindung, d.h. Austausch von Paketen.
  - 3 Abbau der Verbindung.
- Garantierte Datenverbindung
  - ▶ Fehlerkontrolle.
  - ▶ Fehlerkorrektur durch erneutes Senden von Daten.
- Paketgröße
  - ▶ Größer als bei IP.
  - ▶ TCP segmentiert große Daten.
- Rund 75% des Internet-Verkehrs
  - ▶ WWW und Email baut auf TCP auf.

# TCP (2/2)

## Protokoll

- Empfänger bestätigt
  - ▶ Vollständigen Paketempfang
  - ▶ Acknowledge (ACK)
- Sender sendet erneut
  - ▶ Falls ACK ausbleibt.
  - ▶ Nach Ablauf einer Timeout-Periode, die von der der Round-Trip-Time (RTT) abhängt.

## TCP-Zustände

- ▶ Verbindungsorientierung, d.h. Zustände (*states*) werden benötigt.
- ▶ Zustände LISTEN, SYN-SENT, SYN-RECEIVED, ESTABLISHED, FIN-WAIT-1, FIN-WAIT-2, CLOSE-WAIT, CLOSING, LAST-ACK, TIME-WAIT, CLOSED.

# Round-Trip-Time (RTT)

Die **Round-Trip-Time (RTT)** ist die Zeit, die vom Senden einer Anfrage bis zum Empfangen der Bestätigung vergeht.

## Beispiel

---

```
> ping -c 3 www.gwdg.de
PING 134.76.10.47 56(84) bytes of data.
64 bytes from 134.76.10.47: ttl=61 time=0.794 ms
64 bytes from 134.76.10.47: ttl=61 time=0.833 ms
64 bytes from 134.76.10.47: ttl=61 time=0.873 ms

--- ping statistics ---
3 packets, 3 received, 0% packet loss, time 1998ms
rtt min/avg/max/mdev = 0.794/0.833/0.873 ms
```

---

```
> ping -c 3 www.google.de
PING 72.14.221.99 56(84) bytes of data.
64 bytes from 72.14.221.99: ttl=53 time=17.7 ms
64 bytes from 72.14.221.99: ttl=53 time=17.0 ms
64 bytes from 72.14.221.99: ttl=53 time=15.3 ms

--- ping statistics ---
3 packets, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max = 15.375/16.716/17.736 ms
```

---

# TCP-Verbindung

## Verbindungsaufbau

- 3-fach Handshake
  - ① Client sendet SYN zu Server.
  - ② Server antwortet mit SYN-ACK.
  - ③ Client schickt ACK.
- Client und Server tauschen Sequenznummern aus

## Datenverbindung

- Jedes Byte Daten erhöht die Sequenznummer, so wird die Reihenfolge der Segmente festgelegt.
- Empfänger schickt ACK mit Sequenznummer.

## Verbindungsabbau

- 4-fach Handshake
  - ▶ FIN und ACK in beide Richtungen.

# TCP-Segment

## Header

- Quell- und Ziel-Port
  - ▶ IP-Adressen stehen im IP-Paket-Header.
- Sequenz-Nummer
  - ▶ Fortlaufende Nummer für Segmente.
  - ▶ Zum Ordnen empfangener Segmente.
- Data Offset
  - ▶ Länge des TCP-Headers.
- Flags
  - ▶ SYN, ACK, RESET, FIN, ...
- Window
  - ▶ Gewünschte Antwortlänge.
- Prüfsumme
  - ▶ Einfach: 16-Bit Summe (kein CRC).
  - ▶ Header und Payload.

## Daten

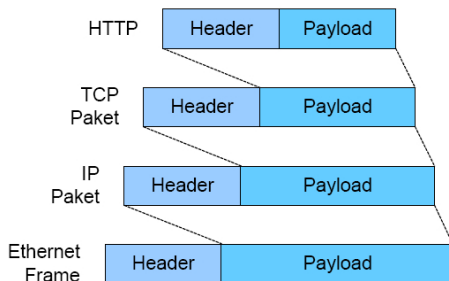
- Die eigentlichen Daten.



# TCP in IP in Ethernet

Kapselung der *protocol data units* (Segmente, Pakete, etc.) als *payload* (Daten).

- Header und Daten des höherem Protokolls werden Daten des darunter liegenden Protokolls.
- Tiefere Protokollschichten kapseln höhere Protokollschichten.



**Umgedrehte Reihenfolge**, höchstes Protokoll ist am tiefsten vergraben.

Telematik

**Kryptographie**

**Einführung**

Symmetrische Verschlüsselung

Blockchiffren und Stromchiffren

Asymmetrische Verschlüsselung

A. Beutelspacher, J. Schwenk, K.-D. Wolfenstetter  
*Moderne Verfahren der Kryptographie*,  
Vieweg+Teubner, Mai 2010.

B. Schneier, N. Ferguson  
*Practical Cryptography*,  
John Wiley & Sons, 2003.

A. J. Menezes, P. C. Van Oorschot, S. A. Vanstone  
*Handbook of Applied Cryptography*,  
Crc Press, Oktober 1996.



## Grundproblem

Wie kann man mit jemanden vertraulich kommunizieren, d.h. kein Unbeteiligter soll Kenntnis von der übermittelten Nachricht erhalten?

Das Problem der Übermittlung (und Speicherung) geheimer Nachrichten kann man durch verschiedene Maßnahmen lösen.

- Organisatorische Maßnahmen
- Physikalische Maßnahmen
- Kryptographische Maßnahmen

# Organisatorische Maßnahmen

## Beispiel

- Ein Gespräch während eines einsamen Waldspaziergangs.
- Übermittlung einer Nachricht durch einen vertrauenswürdigen Boten.
- Einstufung vertraulicher Dokumente als Verschlusssache.

## Beispiel

- Verstecken der Informationen in einem Tresor.
- Übermitteln der Nachricht in einem versiegelten Brief.
- Verheimlichen der Existenz der Nachricht, z.B. durch Geheimtinte.

# Kryptographische Maßnahmen

**Kryptographische Maßnahmen** verändern bzw. entstellen (**verschlüsseln, chiffrieren**) die Nachricht (den **Klartext**).

Dadurch ist die Nachricht für einen Außenstehenden nicht mehr erkennbar und die übertragene Information (der **Geheimtext**) erscheint diesem (meist) völlig unsinnig.

Ein berechtigter Empfänger kann die Nachricht aber (leicht) wieder herstellen (**entschlüsseln, dechiffrieren**).

Zu diesem Zweck besitzen die berechtigten Empfänger eine geheime Zusatzinformation (den **Schlüssel**), darin unterscheiden sie sich von den Außenstehenden.

# Symmetrische Verschlüsselung

Bei der **symmetrischen Verschlüsselung** gibt es nur einen Schlüssel.

Derselbe Schlüssel wird sowohl vom Sender zum Verschlüsseln des Klartext verwendet, als auch vom Empfänger für das Entschlüsseln des Geheimtext benötigt.

Beim Sender ist das kein Problem, da er den Schlüssel gewählt/erzeugt hat.

Dem Empfänger fehlt der Schlüssel, deswegen ist es bei der symmetrischen Verschlüsselung sehr wichtig, dass der Schlüssel auf einem sicheren Übertragungsweg an den Empfänger weitervermittelt wird.

Telematik

**Kryptographie**

Einführung

Symmetrische Verschlüsselung

Blockchiffren und Stromchiffren

Asymmetrische Verschlüsselung

# Verschlüsselung

Der älteste Zweig der klassischen Kryptographie beschäftigt sich mit der **Geheimhaltung** von Nachrichten **durch Verschlüsselung**.

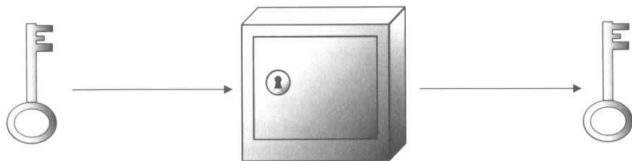
Das Ziel ist, die Nachricht so zu verändern (verschlüsseln), das nur der berechtigte Empfänger diese lesen kann.

Damit dies möglich ist, muss der Empfänger potenziellen Angreifern eine Information vorraus haben, diese (geheime) Information wird **Schlüssel** genannt.

Mit Hilfe des Schlüssels kann der Empfänger die original Nachricht (den Klartext) aus der veränderten Nachricht (dem Geheimtext) wiederherstellen (entschlüsseln).

Bei der **symmetrischen Verschlüsselung** besitzt auch der Sender diesen Schlüssel, d.h. Sender und Empfänger haben den gleichen geheimen Schlüssel, mit dem sie sich gegen die Außenwelt schützen.

# Anschauung



Verschlüsseln schützt die Nachricht davor gelesen zu werden.

Man kann sich vorstellen, dass der Sender die Nachricht in einen Tresor legt und mit Hilfe seines Schlüssels abschließt.

Der Tresor wird samt Inhalt an den Empfänger geschickt. Dieser hat einen identischen Schlüssel, um den Tresor zu öffnen und die Nachricht zu lesen.

In der Kryptographie werden die Nachrichten nicht durch physikalische Maßnahmen geschützt, sondern durch mathematische Methoden.



# Verschlüsselungsalgorithmus (1/2)

## Definition

Ein **symmetrischer Verschlüsselungsalgorithmus** besteht aus einer Funktion  $f$  mit zwei Eingabewerten, dem **Schlüssel**  $k$  und dem **Klartext**  $m$ , die Ausgabe ist der **Geheimtext**  $c$ , der sich aus  $k$  und  $m$  ergibt.

# Verschlüsselungsalgorithmus (2/2)

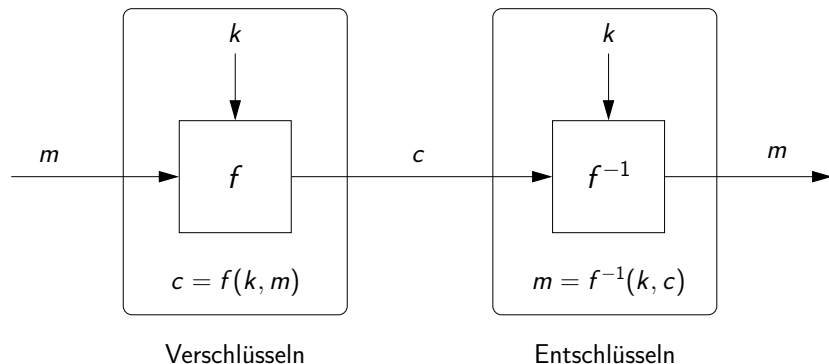
Die Verschlüsselungsfunktion  $f$  muss **umkehrbar** sein, d.h. es muss eine Funktion  $f^{-1}$  geben, die die Wirkung von  $f$  rückgängig macht.

Mit dem Schlüssel  $k$  und dem Gemeintext  $c$  kann man mit  $f^{-1}$  den Klartext  $m$  rekonstruieren.

Angenommen Sender und Empfänger benutzen den gemeinsamen (geheimen) Schlüssel  $k$ .

- Der Sender verschlüsselt einen Klartext  $m$ , indem er den Gemeintext  $c$  berechnet,  $c = f(k, m)$  (oft auch  $f_k(m)$ ).
- Der Empfänger rekonstruiert den Klartext  $m$ , indem er den Geheimtext  $c$  entschlüsselt,  $m = f^{-1}(k, c) = f_k^{-1}(c)$

# Funktionschema



Verschlüsseln und Entschlüsseln müssen in einer sicheren Umgebung stattfinden, das wird im Bild durch die Kästen symbolisiert.

Telematik

**Kryptographie**

Einführung

Symmetrische Verschlüsselung

**Blockchiffren und Stromchiffren**

Asymmetrische Verschlüsselung

# Verschlüsselungsverfahren

In der Regel sollen große Nachrichten oder Nachrichtenströme, d.h. eine kontinuierliche Abfolge von Zeichen, verschlüsselt werden, was mit einer einmaligen Anwendung der Verschlüsselungsfunktion nicht umsetzbar ist.

**Blockchiffren** und **Stromchiffren** sind Verfahren für die Anwendung von Verschlüsselungsalgorithmen auf große Nachrichten oder Nachrichtenströme.

Beide Verschlüsselungsverfahren teilen den Klartext in Blöcke gleicher Größe  $m_1, m_2, \dots$  auf.

Durch die Verschlüsselung entstehen Geheimtextblöcke  $c_1, c_2, \dots$  gleicher Größe.

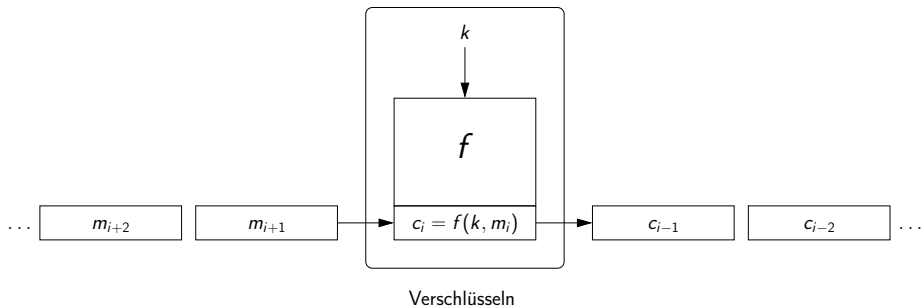
# Blockchiffren (1/2)

## Blockchiffre

- Die Klartextblöcke werden unabhängig voneinander mit der Verschlüsselungsfunktion und **demselben Schlüssel** einzeln zu Geheimtextblöcken chiffriert.
- Die Geheimtextblöcke werden entsprechend einzeln dechiffriert.

## Blockchiffren (2/2)

Blockchiffre mit Verschlüsselungsfunktion  $f$  und Schlüssel  $k$ .



# Caesar-Verschlüsselung

Die **Caesar-Verschlüsselung** ist eine Blockchiffre mit symmetrischem Verschlüsselungsalgorithmus.

Ein Klartext  $m$  kann eine beliebige Zeichenfolge über dem Alphabet  $\Sigma = \{A, B, \dots, Z\}$  der 26 Großbuchstaben sein,  $m \in \Sigma^*$ .

Ein Klartext wird in Blöcke der Größe ein Zeichen aufgeteilt.

Der Schlüssel  $k$  ist ein Zeichen aus dem Alphabet  $\Sigma$ .

Die symmetrische Verschlüsselungsfunktion  $f_k : \Sigma \rightarrow \Sigma$  verschiebt das übergebene Zeichen im Alphabet zyklisch nach **rechts**, dabei wird die Anzahl der zu verschiebenden Stellen von der Position des Schlüssels  $k$  im Alphabet bestimmt.

Die symmetrische Entschlüsselungsfunktion  $f_k^{-1} : \Sigma \rightarrow \Sigma$  verschiebt entsprechend, abhängig von  $k$ , das übergebene Zeichen im Alphabet zyklisch nach **links**.



# Beispiel

Caesar-Verschlüsselung mit Schlüssel  $C$ .

Die Position von Schlüssel  $C$  im Alphabet ist 3, d.h.  $f_C$  ( $f_C^{-1}$ ) verschiebt ein übergebenes Zeichen um 3 Stellen im Alphabet zyklisch nach rechts (links).

$$f_C(A) = D$$

$$f_C(B) = E$$

...

$$f_C(W) = Z$$

$$f_C(X) = A$$

$$f_C(Y) = B$$

$$f_C(Z) = C$$

$$f_C^{-1}(D) = A$$

$$f_C^{-1}(E) = B$$

...

$$f_C^{-1}(Z) = W$$

$$f_C^{-1}(A) = X$$

$$f_C^{-1}(B) = Y$$

$$f_C^{-1}(C) = Z$$

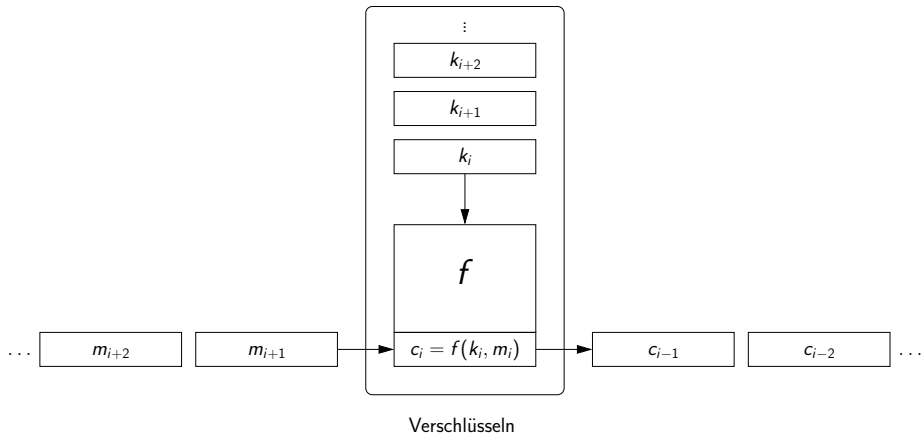
*CAESAR* wird zeichenweise mit  $C$  verschlüsselt zu *FDHVDU*, das wird zeichenweise mit  $C$  entschlüsselt wieder zu *CAESAR*.

## Stromchiffre

- Eine **Schlüsselstrom**  $k_1, k_2, \dots$  wird erzeugt, sodass für jeden Klartextblock ein eigener Schlüssel vorliegt.
- Ein Klartextblock wird jeweils mit der Verschlüsselungsfunktion und dem zugehörigen Schlüssel des Schlüsselstroms zu einem Geheimtextblock chiffriert.
- Ein Geheimtextblock kann nur mit den zugehörigen Schlüssel des Schlüsselstroms dechiffriert werden.

# Stromchiffre

Stromchiffre mit Verschlüsselungsfunktion  $f$  und Schlüsselstrom  $k_1, k_2, \dots$



# One-Time-Pad

**One-Time-Pad** (wörtlich *Einmal-Block*) ist eine Stromchiffre für (große) Nachrichten bekannter Länge mit symmetrischem Verschlüsselungsalgorithmus.

Es werden **zufällig** genauso viele Schlüssel erzeugt, wie Klartextblöcke zu verschlüsseln sind, dabei haben die Schlüssel mindestens die gleiche Länge wie die Klartextblöcke.

Jeder Schlüssel wird nur **einmal** verwendet, um einen Klartextblock mit der Verschlüsselungsfunktion in einen Geheimtextblock zu chiffrieren.

## Bemerkung

Da es für jeden Klartextblock einen Schlüssel gibt, die Schlüssel zufällig gewählt und nur einmal verwendet werden, kann man zeigen, dass One-Time-Pad ein absolut sicheres Verfahren ist.

## Bemerkung

One-Time-Password (*Einmalkennwort*) ist ein anderes Verfahren, das z.B. durch Transaktionsnummern-Listen (TAN-Listen) beim Online-Banking Anwendung findet.

# Beispiel

Für ein Umsetzung von One-Time-Pad wird die Verschlüsselungsfunktion  $f_k : \Sigma \rightarrow \Sigma$  des Caesar-Verschlüsselung benutzt, mit dem Zusatz, dass das Alphabet  $\Sigma = \{A, B, \dots, Z, +, 0, 1, 2, \dots 9\}$  an Position 27 das Pluszeichen und an den Positionen 28 bis 37 die Ziffern 0 bis 9 enthält.

Es wird eine zufällige Schlüsselfolge erzeugt, z.B. durch Beobachtung der Bewegung von Blättern auf dem *Forum Romanum*.

CT0+SHBF++00EXKKS90S+MPJM+Y+EXOPFRB+YNZSDX56TSJMI7YNQF+EHLBEE60...

Die Zeichen der zu verschlüsselnden Nachricht werden einzeln mit der Verschlüsselungsfunktion  $f_k$  und jeweils einem Zeichen der Schlüsselfolge chiffriert.

Klartext: 15+MAERZ+CAESAR+TREFFEN+DOLCHE+NICHT+VERGESSEN

Schlüsselfolge: CT0+SHBF++00EXKKS90S+MPJM+Y+EXOPFRB+YNZSDX56TS

Geheimtext: 4PECTMT4Q215XY1ABRTY5R29QE92M1E20UJJ0839K10PY5

Telematik

**Kryptographie**

Einführung

Symmetrische Verschlüsselung

Blockchiffren und Stromchiffren

Asymmetrische Verschlüsselung

# Asymmetrische Verschlüsselung

Die **asymmetrische Verschlüsselung** verwendet zwei unterschiedliche Schlüssel.

Es wird ein öffentlicher und ein privater Schlüssel, die zueinander komplementär sind, verwendet.

Daten, die mit dem privaten Schlüssel verschlüsselt werden, können mit dem öffentlichen Schlüssel entschlüsselt werden (verschlüsseln von Nachrichten).

Daten, die mit dem öffentlichen Schlüssel verschlüsselt werden, können mit dem privaten Schlüssel entschlüsselt werden (signieren von Nachrichten).

Entscheidend ist aber, dass der private Schlüssel nicht aus dem öffentlichen Schlüssel abgeleitet werden kann.

Bei der asymmetrischen Verschlüsselung erzeugt jeder Teilnehmer ein Schlüsselpaar, behält den privaten Schlüssel und macht den öffentlichen Schlüssel den anderen Teilnehmern zugänglich.

Für den Austausch von verschlüsselten Nachrichten chiffriert der Sender die Daten mit dem öffentlichen Schlüssel des Empfängers und schickt sie diesem. Der Empfänger kann die Daten anschließend mit Hilfe seines privaten Schlüssels dechiffrieren.

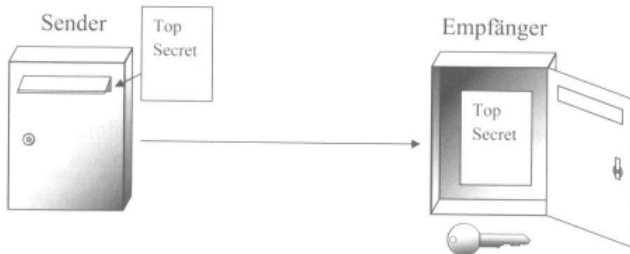
# Anschauung

Man kann einem Empfänger eine verschlüsselte Nachricht schicken, ohne eine Geheiminformation zu besitzen.

Das kann man sich wie das Einwerfen einer Nachricht in einen Briefkasten vorstellen.

Jeder, der Zugang zum Briefkasten (dem öffentlichen Schlüssel) hat, kann eine Nachricht einwerfen (Verschlüsseln).

Nur der Empfänger kann mit seinem privaten Schlüssel den Briefkasten öffnen und die Nachricht entnehmen (Entschlüsseln).





# Verschlüsselungsalgorithmus

## Definition

Ein **asymmetrischer Verschlüsselungsalgorithmus** besteht aus

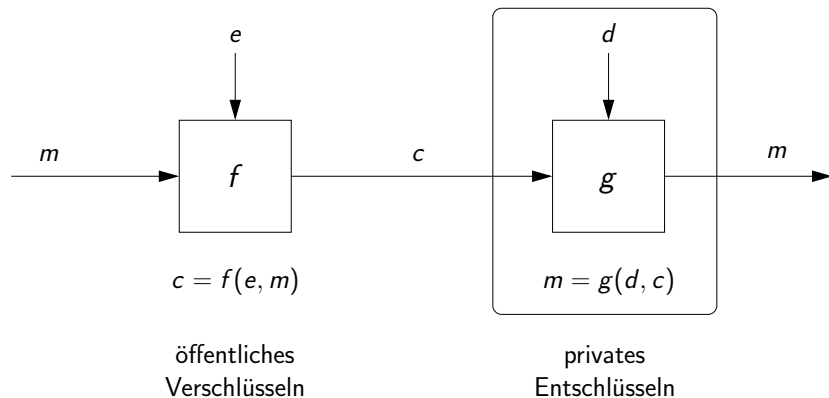
- einer Funktion  $f$  mit zwei Eingabewerten, dem **öffentlichen Schlüssel**  $e$  und einem Text  $m$ , die Ausgabe ist der Text  $f(e, m)$ ,
- einer Funktion  $g$  mit zwei Eingabewerten, dem **privaten Schlüssel**  $d$  und einem Text  $m$ , die Ausgabe ist der Text  $g(d, m)$ .

Für alle Texte  $m$  gelten die folgenden Beziehungen zwischen  $f$  und  $g$ .

$$g(d, f(e, m)) = m$$

$$f(e, g(d, m)) = m$$

# Funktionschema



Nur das Entschlüsseln muss in einer sicheren Umgebung stattfinden, das wird im Bild durch den Kasten symbolisiert.

## Satz

Sei  $n = pq$  das Produkt zweier Primzahlen  $p, q \in \mathbb{N}$ .

Dann gilt für alle natürlichen Zahl  $m, k$  mit  $m < n$  folgende Gleichung.

$$m^{k(p-1)(q-1)+1} \bmod n = m$$

## Rechenregeln

Für natürliche Zahlen  $x, i, j, \ell \in \mathbb{N}$  gilt Folgendes.

$$(x^i \bmod \ell)^j \bmod \ell = (x^i)^j \bmod \ell = x^{i \cdot j} \bmod \ell$$

## Einwegfunktion

Die Funktion  $h(r, m) = m^r \bmod n$  ist auch wenn  $r$  und  $n$  bekannt sind nicht umkehrbar. D.h. aus  $m^r \bmod n$  kann man  $m$ , in der Regel nicht rekonstruieren.

### Beispiel

Sei  $r = 3$  und  $n = 7$ , dann gilt  $3^3 \bmod 7 = 5^3 \bmod 7 = 6$ .

# RSA-Algorithmus

Der RSA-Algorithmus wurde 1978 von Rivest, Shamir and Adleman erfunden, der Algorithmus basiert auf folgendem Prinzip.

Sei  $n = pq$  das Produkt zweier Primzahlen  $p, q \in \mathbb{N}$ .

Wähle bzw. berechne die Schlüssel  $e, d \in \mathbb{N}$ , sodass für ein  $k \in \mathbb{N}$  gilt

$$e \cdot d = k(p - 1)(q - 1) + 1 .$$

Dann gilt das Folgende für jedes  $m \in \mathbb{N}$  mit  $m < n$ .

$$(m^e \bmod n)^d \bmod n = m^{e \cdot d} \bmod n = m$$

$$(m^d \bmod n)^e \bmod n = m^{e \cdot d} \bmod n = m$$

Somit sind die Funktionen  $f$  und  $g$  identisch,

$$f(r, m) = g(r, m) = m^r \bmod n.$$