Informatik 6ai Schuljahr 2024/2025 als PDF exportieren

Informatik 6. Klasse - Schuljahr 2024/25

Lehrplan

- Lehrplaninhalte

Themengebiete

- 7) Web Design Grundlagen (HTML, CSS)
- 8) Netzwerksicherheit
- 9) Kommunikation in Rechnernetzwerke
- 10) Algorithmen und Datenstrukturen
- 11) Betriebssysteme

Leistungsbeurteilung

1/3 - Test (SA)

- 2x Tests pro Semester
  - 1. Test – Do, 17.10.2024 - Themengebiet 7 - 8 (bis Symmetrische Verschlüsselung inkl. Skytale, Cäsar, Vigenere)
  - 2. Test – Do, 12.12.2024 - Themengebiet 8 (ab 8.1.2.2 Asymmetrische Verschlüsselung)
  - 3. Test – Mi, 25.03.2025 - Themengebiet 9 (bis inklusive Routing)
  - 4. Test – Do, 15.05.2025 - Themengebiet 9 (Netzwerksimulation - WWW, DNS, E-Mail + Protokolle, Ports) - 10 (??)

1/3 - Mitarbeit (MA)

- Aktive Mitarbeit im Unterricht (aMA)
- Mündliche Stundenwiederholungen (mMA)
- Schriftliche Stundenwiederholungen (sMA)

1/3 - Praktische Arbeiten (PA)

- 1x praktischer Arbeitsauftrag pro Woche via Google Classroom

Leistungsstand

Den aktuellen Leistungsstand könnt ihr jederzeit einsehen!

Lehrplaninhalte

Die nachstehenden Lehrplaninhalte werden in der angegebenen Reihenfolge (=Priorität) durchgenommen werden. Im Unterrichtsfach IT-Labor werden die Themen projektbasiert behandelt.

5 + 6. Klasse IT-Labor/WPF

1. Videobearbeitung mit DaVinci Resolve
2. Hardware – PC Systeme (Laptop, Spielekonsole, Handy,..) kennenlernen, Komponenten identifizieren
3. 3D – Modellierung & Animation & Druck
4. Mediendesign (Canva – Flyer, Plakate) & Inkscape Vektorgrafiken (Logo) & Bild (Pixlr) & Podcast bzw. Audiobearbeitung / Videoblog
5. Kamerasysteme (Handy + Gimbal, Drohne, GoPro,..)
6. Netzwerktechnik – Praxis
7. Robotik (Wetterstation,..)
8. Handy-App Programmierung
9. VR – Programmierung
10. Unity – 3D Spielprogrammierung

5. Klasse

1. Zahlensysteme
2. Informationseinheiten
3. Zeichencodierung
4. Schaltalgebra
5. Algorithmik und Programmierung Basics (C++ Grundstrukturen bis Funktionen)
6. Tabellenkalkulation & Datenanalyse (Datenimport, filtern, exportieren, SVERWEIS, WENN-DANN, Pivot-Tabellen)

6. Klasse

1. Webentwicklung Basics (HTML & CSS)
2. Datenschutz- & Sicherheit & Lizensierung & Kryptographie & Verschlüsselung
3. Netzwerktechnik Theorie & Simulation
4. Algorithmik und Programmierung Datenstrukturen (C++ Rekursionen, Arrays + Sortieralgorithmen)

7. Klasse

1. Algorithmik und Programmierung Objektorientierung (Klassen, Vererbung)
2. Betriebssysteme (Windows / Linux – Scripts, Serverdienste – Webserver, DB-Server) & Virtualisierung

3. Datenbanksysteme (ER-Modelle, Relationenmodell, SQL)
4. Textverarbeitung (mehrseitige Dokumente, VWA Vorlagen,..)

8. Klasse

1. Webentwicklung Advanced (PHP Formulare & DB-Connection, Javascript, Frameworks Bootstrap, CMS - Systeme)
2. Grundlagen der KI

8) Netzwerksicherheit

Zweifel zu haben ist ein unangenehmer Zustand, sich in Sicherheit zu wiegen ist ein absurder Zustand (Voltaire)

Not, Person und Zeit, machen die Gesetze eng und weit.

Es gibt keine Sicherheit, nur mehr oder weniger Unsicherheit (Josef Maler)

Sei vorsichtig, öffne keinem Fremden die Haustür.

kryptographie.mp4

Definition

Netzwerksicherheit steht als Begriff stellvertretend für sämtliche Schutzmaßnahmen, um IT-Infrastrukturen gegen unbefugte Zugriffe, Schäden und Verluste abzusichern. Diese Maßnahmen können technischer oder organisatorischer Natur sein und sorgen dafür, dass ein Netzwerk vertraulich, integer und verfügbar bleibt. So zählen Verschlüsselungstechnologien und Firewalls ebenso zur professionellen Netzwerksicherheit wie Sicherheits- und Passwortrichtlinien und Security-Schulungen.

Um ein IT-Netzwerk umfassend zu sichern, braucht es mehrere Schutzschichten, die jeden Bereich im Netzwerk bedenken. Außerdem ist es sinnvoll, Netzwerk und Sicherheit individuell aufeinander abzustimmen. Wir zeigen Ihnen, welche Security-Schichten notwendig sind und wie umfangreich Ihre IT-Security je nach Unternehmensart und Angriffsrisiken ausfallen sollte.

Schutz des eigenen IT-Netzwerkes

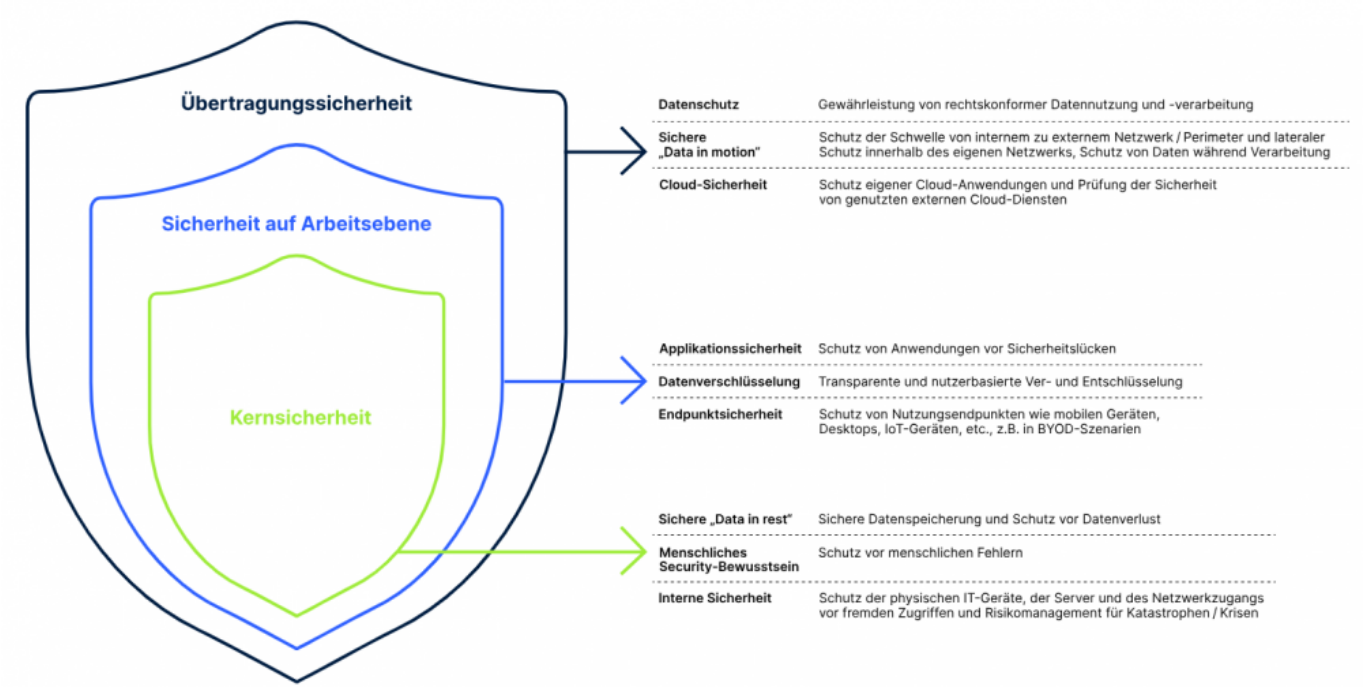
Ein IT-Netzwerk besteht aus einer Vielzahl an Einzelkomponenten: Wichtige Daten lagern auf internen oder externen Servern, Mitarbeitende sind mit Desktop-PCs, Laptops und Smartphones ausgestattet, ein Gateway oder ein Router sorgt für den Internetzugang und je nach Unternehmensgröße vernetzen ein oder mehrere Switches intern weitere Geräte (LAN) wie beispielsweise Access Points für eine professionelle WLAN-Abdeckung oder auch Arbeitsgeräte oder Drucker.

Bei IT-Netzwerken, die lokal weit verteilt und stark verzweigt sind, bedarf es zum Schutz einer mehrschichtigen und skalierbaren IT-Security, die im besten Fall möglichst automatisiert für mehr Sicherheit im Netzwerk sorgt.

Schichten und Bestandteile professioneller IT-Security

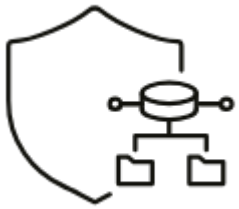
Zum Schutz des eigenen IT-Netzwerkes zählt zum einen die Absicherung sämtlicher sich im Netzwerk befindlichen Daten, Geräte, Server und Applikationen (Datensicherheit, interne Sicherheit und Applikationssicherheit), zum anderen aber auch Perimeter Security, also die Absicherung der Schwelle zwischen internem und externem Netz – demnach auch sichere Übertragungswege und Netzwerkzugänge (Endpunktsicherheit und Datenverschlüsselung).

Zu guter Letzt darf aber auch die Sicherheit der bewegten Daten und externen Cloud-Anwendungen nicht außer Acht gelassen werden (Sicherheit von „data in motion“, Datenschutz und Cloud-Sicherheit), ebenso wenig wie die menschliche Komponente (menschliches Security-Bewusstsein).



Kernsicherheit

Sichere „Data in rest“



Sichere Datenspeicherung und Schutz vor Datenverlust durch

- Strenge Passwort-Richtlinien (Komplexität, Ablaufdatum, Zwei-Faktor-Authentifizierung)
- Sicherheitseinstellungen für Dateien
- Getrennte Datenlagerung
- Klar definierte Zugriffs- und Bearbeitungsrechte
- Archive
- Datenbackups

Interne Sicherheit

Schutz der physischen IT-Geräte, Server und des Netzwerkzugangs vor fremden Zugriffen sowie Risikomanagement für Katastrophen / Krisen durch

- Zugangskontrollen auf Firmengelände
- ggf. Live-Monitoring mit Kameraüberwachung und Aktivitäten-Logs
- Professionelle Next-Generation Web Application UTM-Firewall
- Regelmäßige Security-Patches, Funktionsfähigkeitschecks und Software Updates
- Netzwerksegmentierung

Menschliches Security-Bewusstsein

Schutz vor menschlichen Fehlern durch

- Umfangreiche IT-Security- und Compliance-Schulungen, insb. zu E-Mail-, Passwort- und Social Media-Sicherheit, Vertraulichkeitsregelungen und Verhalten im Ernstfall
- Tests wie z.B. Phishing-Simulationen
- Vertraulichkeitsklassifikationen für Dateien und Informationen
- Gut zugängliche Sicherheitsrichtlinien
- Regelmäßige Erinnerungen und Auffrischungen

IT-Sicherheit auf Arbeitsebene

Endpunktsicherheit

Schutz von Nutzungsendpunkten wie mobilen Geräten, Desktops, IoT-Geräten und -Sensoren, etc., insb. in dezentralen und hybriden Arbeitsumgebungen (BYOD, Remote Work, externe Dienstleister, etc.) durch

- Nutzung von VPN-Clients und ZTNA (Zero-Trust-Network-Access)
- Multifaktor-Authentifizierung von Nutzer:innen
- Übersichtliches, zentral gemanagtes Asset-Inventar aller Geräte und virtuellen Ressourcen
- Individuelle, feingranulare Zugriffsrechte pro Nutzer:in
- Regelmäßige Systemupdates und Securitypatches
- Klare Nutzungs- und Sicherheitsrichtlinien
- Abschalten nicht zwingend notwendiger Ports

Datenverschlüsselung

Transparente und nutzerbasierte Ver- und Entschlüsselung im Hintergrund durch

- VPN- oder ZTNA-Netzwerke
- Verschlüsselungsparameter und - algorithmen nach aktuellem BSI-Standard
- Regelmäßige Überprüfung der aktuellen Standards
- Konzept zur Schlüsselverwaltung (PKI - Public Key Infrastructure) inklusive regelmäßiger Audits

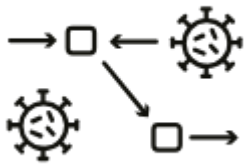
Applikationssicherheit

Schutz von Anwendungen vor Sicherheitslücken durch

- Zugriff auf ausschließlich vertrauenswürdige und geprüfte, freigegebene Apps
- Gezielter Einsatz von VPNs
- Zugangskontrolle und Application Monitoring / Steering
- Automatische Sessionterminierung bei Nicht-Nutzung
- Regelmäßige (Security-)Updates
- Aufbewahrungsrichtlinien

Übertragungssicherheit

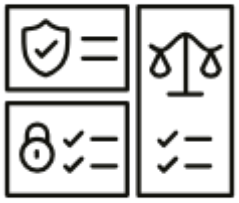
Sichere „Data in motion“



Schutz der Schwelle von internem zu externem Netzwerk (Perimeter) und lateraler Schutz innerhalb des eigenen Netzwerks durch

- Mehrschichtige Verschlüsselung von Daten während Übertragung
- Ausschließliche Nutzung von VPN- oder ZTNA-Verbindungen
- Netzwerksegmentierung durch VLANs
- Caching Routines zur Vermeidung von öffentlichem Zugang
- Regelmäßige Penetrationstests
- Nutzer- und systemspezifische Zugriffsschlüssel nur auf benötigte Daten

Datenschutz



Gewährleistung von rechtkonformer Datennutzung und -verarbeitung durch

- DSGVO-konforme Anwendungen und Systeme
- Backdoor-freie Netzwerkkomponenten
- In der EU gehostete Clouddienste

Cloud-Sicherheit

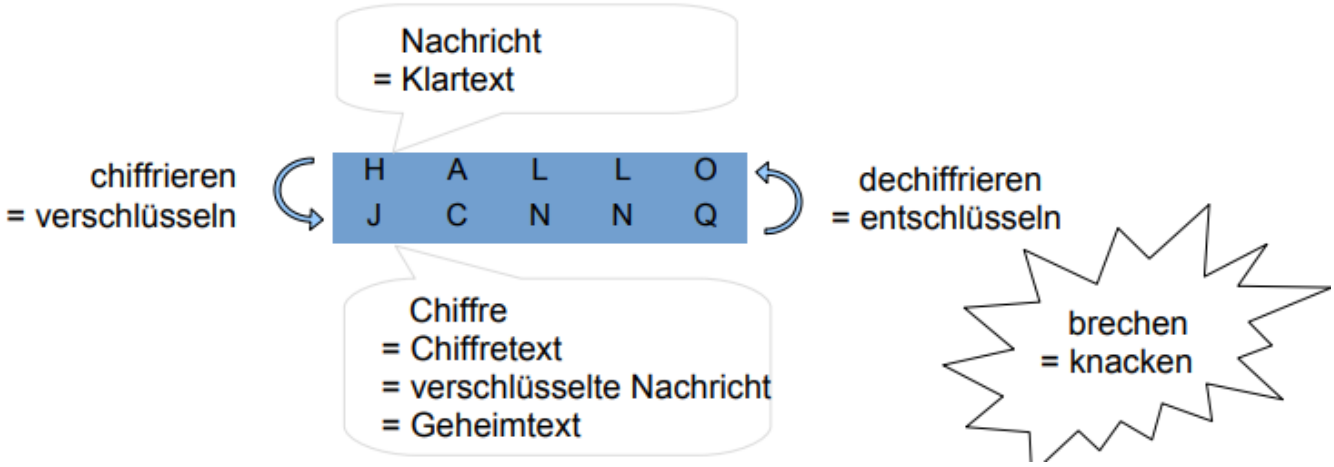
Schutz eigener Cloud-Anwendungen und sorgfältige Prüfung der Sicherheit von genutzten externen Cloud-Diensten durch

- Klare Klärung von Security-Angelegenheiten zwischen Cloud-Anbieter und -Nutzer
- Physische Host-Zugangskontrollen
- Sichere, DSGVO-konforme Infrastruktur
- Georedundanz des Hosts
- Security Patches
- Zugangskontrollen zu Unternehmensdaten in der Cloud
- Backdoor-Freiheit

Kryptologie

Umgangssprachlich werden kryptologische Begriffe oft nicht eindeutig verwendet. Daher ist inbesonderem Maße auf eine korrekte Verwendung der Begriffe zu achten. Hier eine kurze Zusammenfassung:

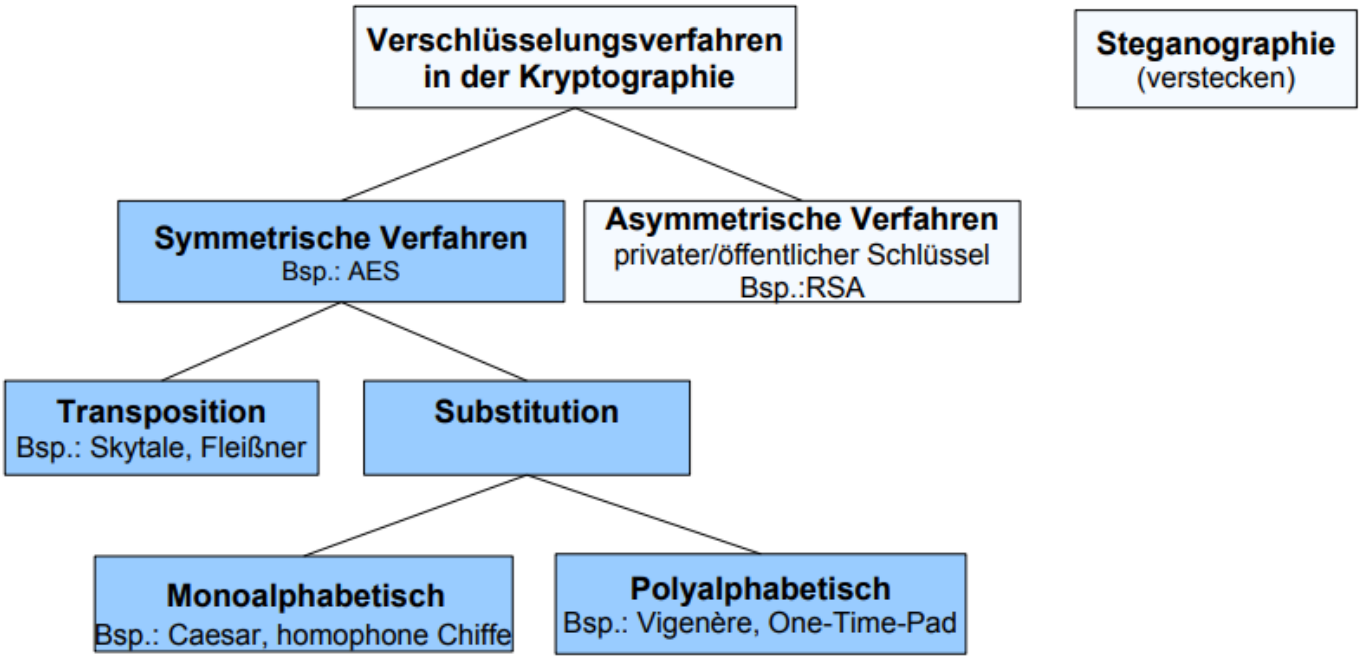
- Beim **Datenschutz** wird die Person mit ihren Rechten geschützt (Persönlichkeitsrecht,Urheberrecht,...).
- Bei der **Datensicherheit** werden die Daten vor unberechtigten Zugriffen geschützt.



Die Kryptologie ist die Wissenschaft der Verschlüsselung und Entschlüsselung von Informationen sowie Analyse kryptografischer Verfahren. Sie umfasst die

- **Kryptografie**: Lehre von den Methoden zur Ver- und Entschlüsselung von Nachrichten zum Zweck der Geheimhaltung von Informationen gegenüber Dritten.
- **Kryptoanalyse**: Analyse und Bewertung der Sicherheit von kryptografischen Verfahren gegen unbefugte Angriffe.

Bei einer **Verschlüsselung** ist das Verfahren (meist) bekannt, der Schlüssel ist geheim. Es geht um den Austausch von Informationen, die nicht für alle bestimmt sind.. Bei einer **Codierung** ist das Verfahren bekannt, und die Anleitung zum Codieren und Decodieren öffentlich. Einen Schlüssel gibt es nicht, und die ausgetauschten Informationen sind nicht geheim. (Blindenschrift, Morsecode, ...).



Kryptografie

Kurz: Kryptografie ist die Lehre der Verschlüsselung von Daten.

Lang: Kryptografie ist eine Wissenschaft, die sich mit Methoden beschäftigt, die durch Verschlüsselung und verwandte Verfahren Daten von unbefugten Manipulation schützen sollen.

Ursprung: Das Wort Kryptografie kommt aus dem Griechischen, wo kryptein „verstecken“ und gráphein „schreiben“ bedeutet.

Die beiden **wichtigsten Hilfsmittel der Kryptografie** sind:

- Die **Mathematik**, denn nur mit Hilfe von mathematischen Kenntnissen ist es möglich, Verfahren zur sicheren Verschlüsselung von Daten zu entwickeln
- Und der **Computer**, weil er die Verschlüsselungsverfahren ausführt und wichtige Dienste bei der Untersuchung von kryptografischen Methoden auf Schwachstellen leistet.

Motive der Kryptografie

- **Vertraulichkeit**

Geheimhaltung ist die offensichtlichste und bekannteste Anwendung kryptografischer Verfahren

- **Intigrität**

Für den Empfänger nachprüfbar sein, das er die Nachricht unversehrt erhalten hat

- **Authentizität**

Identität des Absenders einer Nachricht soll für den Empfänger nachprüfbar sein

- **Gültigkeit**

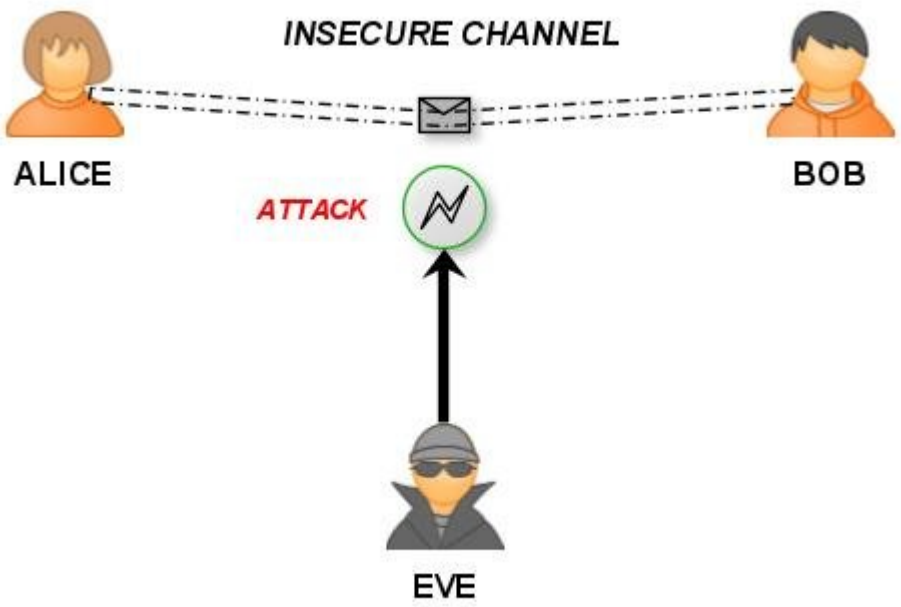
Nachricht kann durch zwischenzeitliche Ereignisse ihre Bedeutung verlieren

- **Nichtabstreitbarkeit**

Dass ist Absender einer Nachricht seine Urheberschaft später nicht verleugnen kann.

Allgemeines Modell

2 Personen (Alice und Bob) tauschen Daten über einen abhörbaren Kanal aus, heute meist das Internet, es kann dies aber auch eine Telefonleitung, eine Funkverbindung oder der Transport einer Diskette sein. Eine „böse“ gesinnte Person (Mallory/Eve) kann den Übertragungskanal beliebig beeinflussen. Er kann die Daten abfragen, mitlesen, analysieren, manipulieren und weiterleiten.



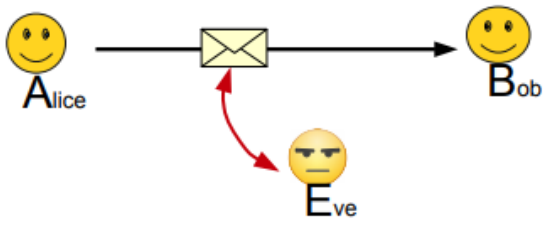
Auf Basis dieses einfachen Modells, kann die Kryptografie durch Verschlüsselung und ähnliche Maßnahmen verhindern dass

- Mallory mit den abgefangenen Daten etwas anfangen kann,
- Mallory übertragene Daten unbemerkt verändert,
- Mallory sich unbemerkt Alice gegenüber als Bob ausgibt (und umgekehrt)
- Alice unerkannt behaupten kann, dass eine von ihr gesendete Nachricht in Wirklichkeit eine Fälschung von Mallory sei.

Die Kryptografie kann aber nicht verhindern, dass

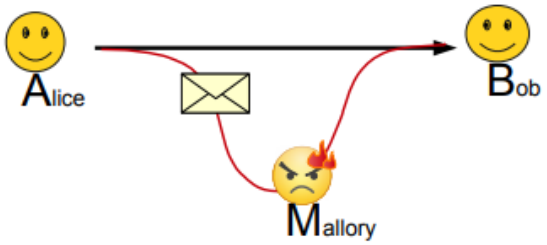
- Mallory Nachrichten verändert (er kann es nur nicht unbemerkt),
- Mallory Daten abfängt (er nur nichts von verschlüsselten Daten),
- Mallory die Leitung zerstört (physikalisch, durch Softwarefehler u.ä.).

Szenarien:	**Gefahren:**	**Ziele** der Kryptologie



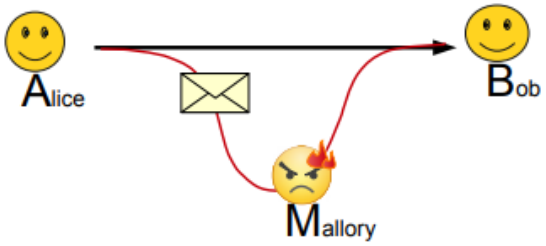
mitlesen => **Vertraulichkeit**

Können wirklich nur Alice und Bob die Nachricht lesen?



ändern => **Integrität**

Ist die Nachricht unverändert?
Sind die Daten original?



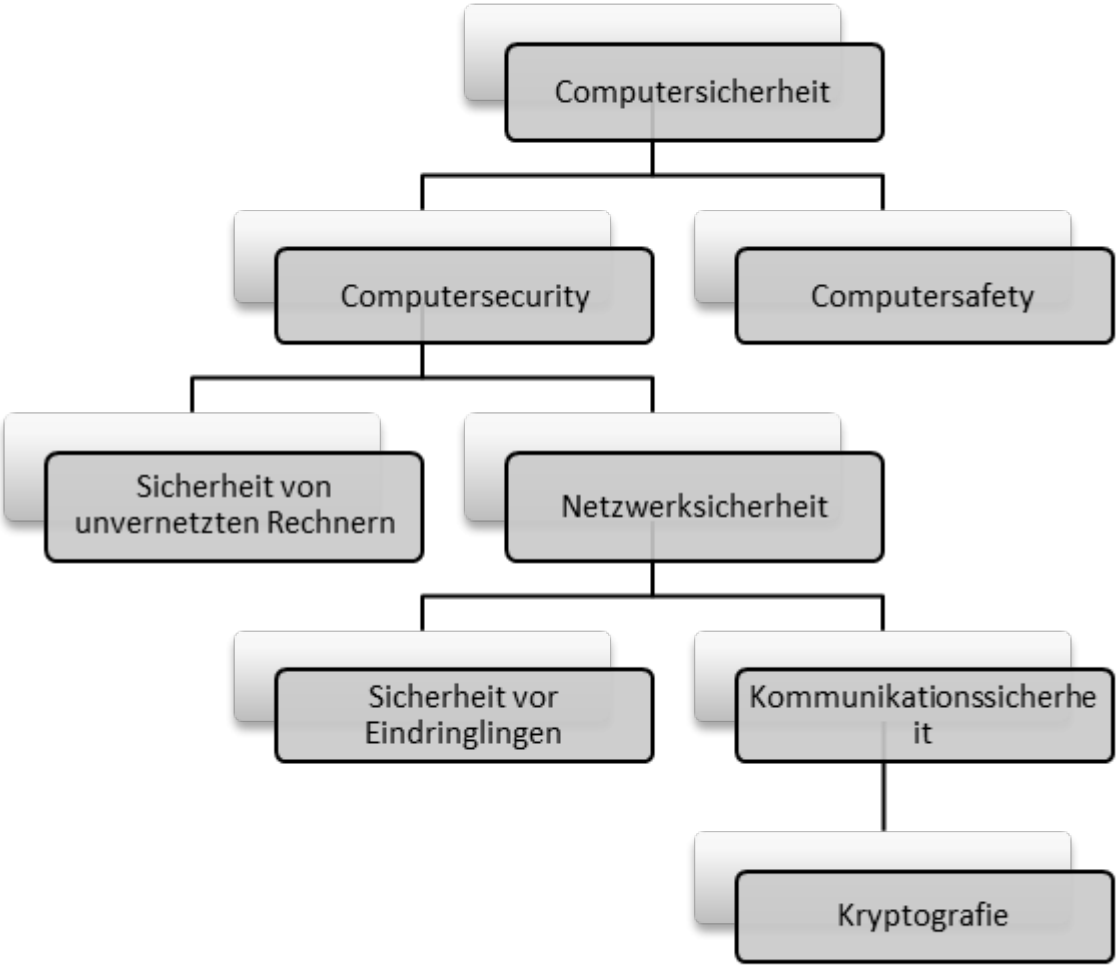
als A ausgeben
=> **Authentizität**
Kommt die Nachricht wirklich von Alice?
Landet die Nachricht wirklich bei Bob?

=> **Verbindlichkeit**
Kann Bob beweisen, dass die Nachricht von Alice kommt, selbst wenn sie es abstreitet? ('*Habe ich nie gesagt.*')
Kann Alice beweisen, dass Bob die Nachricht erhalten hat? ('*Habe ich nicht bekommen.*')

=> Unterschiedliche **Ziele** erfordern unterschiedliche **Verfahren**.

Bsp.: Eine Verschlüsselung liefert Vertraulichkeit, aber keine Authentizität.

Teilgebiet der Computersicherheit



- In der Computer-Safety geht es um den Schutz vor unbeabsichtigten Schäden. Dazu gehören defekte Geräte, unbeabsichtigtes Löschen, Übertragungsfehler, Festplatten-Crashes, Blitzeinschläge, Überschwemmungen, falsche Bedienung, defekte Speichermedien und Ähnliches.
- Die Computer-Security dagegen bezeichnet die Sicherheit vor absichtlichen Störungen. Dazu gehören die Sabotage von Hardware, Hackereinbrüche, das Schnüffeln in geheimen Dateien und dergleichen.

Der Bereich der Netzwerksicherheit beschäftigt sich vor allem mit zwei Sicherheitsfragen:

- Wie kann ein vernetzter Computer davor beschützt werden, dass ein Unbefugter über das Netzwerk darauf zugreift (man spricht dabei von hacken oder cracken).
- Wie können Nachrichten, die den Computer verlassen vor einem Abhörer oder Manipulierer geschützt werden (Kommunikationssicherheit).

Gründe für Kryptografie

- **Wirtschaftsspionage**

Staatliche Geheimdienste sind nach dem Ende des kalten Krieges vermehrt auf neue

Beschäftigungsbereiche verlegt worden. Wirtschaftliche Interessen gelten heute als häufiger Grund Spionage zu betreiben. Weltweit führend – nicht nur im Bereich der Wirtschaftsspionage – ist die amerikanische Geheimorganisation NSA (National Security Agancy). Die NSA ist der weltweit größte Arbeitgeber von Mathematikern und größter Hardwareabnehmer. Firmen nutzen heute auch die Kenntnisse von Hackern um die Konkurrenz auszuspionieren. Der geschätzte Schaden durch Wirtschaftsspionage übersteigt in Ländern wie Deutschland die Milliardengrenze. Auch wenn nur ein Teil davon über das Internet erfolgt, so ist die Gefahr vielfach unterschätzt oder nicht bewusst.

- **Kommerzielle Nutzung des Internets**

Ein guter Grund für den Einsatz von Kryptografie ist die Tatsache, dass sich in einem abhör- und manipulationssicheren Internet mittels Online-Shops, Auktionsbörsen, OnlineBanking u. ä. eine große Menge an Geld verdienen/bewegen lässt.

- **Privatsphäre**

Es gibt auch Gründe für den Einsatz von Kryptografie, die keine kommerziellen Gedanken verfolgen. So ist es das Recht jedes Bürgers, eine Privatsphäre zu behalten durch den Einsatz von Kryptografie möglich. Ein privater Brief wird ja auch in einem Umschlag versandt! Es ist im Internet auf jeden Fall einfacher, abgefangene Daten maschinell auszuwerten, als dies mit herkömmlicher Briefpost der Fall war.

Bei allen Vorteilen der Kryptografie darf aber nicht vergessen werden, dass sie auch Gefahren bringt: Kriminelle können durch den Einsatz geeigneter Verschlüsselungsverfahren nach Belieben Nachrichten austauschen.

Anwendungen

- **Passwörter**

Kryptografie wird häufig eingesetzt, um die Authentizität von Passwörtern zu überprüfen und gleichzeitig gespeicherte Passwörter zu verschleiern. Auf diese Weise können Dienstanbieter Passwörter authentifizieren, ohne eine Klartextdatenbank mit allen Passwörtern führen zu müssen, die für Hacker anfällig sein könnte.

- **Sicheres Surfen im Internet**

Beim Surfen auf sicheren Websites schützt die Kryptografie die Benutzer vor Lauschangriffen und „Man-in-the-Middle“-Angriffen (MitM). Die Protokolle Secure Sockets Layer (SSL) und Transport Layer Security (TLS) basieren auf der Verschlüsselung mit öffentlichen Schlüsseln, um die zwischen Webserver und Client gesendeten Daten zu schützen und sichere Kommunikationskanäle

herzustellen.

- **Elektronische Signaturen**

Elektronische Signaturen, oder E-Signaturen, werden zum Unterzeichnen wichtiger Dokumente im Internet verwendet und gelten oftmals als rechtsverbindlich. Mit Kryptografie erstellte elektronische Signaturen können validiert werden, um Betrug und Fälschungen zu verhindern.

- **Kryptowährungen**

Kryptowährungen wie Bitcoin und Ethereum beruhen auf einer komplexen Verschlüsselung von Daten, deren Entschlüsselung erhebliche Mengen an Rechenleistung erfordert. Durch diese Entschlüsselungsprozesse erfolgt das sogenannte „Minting“ neuer Coins, die dann in Umlauf gebracht werden. Kryptowährungen stützen sich zudem auf fortschrittliche Kryptografie, um Krypto-Wallets zu sichern, Transaktionen zu verifizieren und Betrug zu verhindern.

- **Authentifizierung**

In Situationen, in denen eine Identitätsauthentifizierung erforderlich ist, wie z. B. bei der Anmeldung bei einem Online-Bankkonto oder beim Zugriff auf ein sicheres Netzwerk, kann die Kryptografie bei der Verifizierung der Identität von Benutzern und der Authentifizierung ihrer Zugriffsberechtigungen helfen.

- **Sichere Kommunikation**

Ganz gleich, ob es um den Austausch von Staatsgeheimnissen oder um eine private Unterhaltung geht – die End-to-End-Verschlüsselung wird zur Authentifizierung von Nachrichten und zum Schutz von Zwei-Wege-Kommunikation wie Videokonferenzen, Sofortnachrichten und E-Mails verwendet. Die End-to-End-Verschlüsselung bietet ein hohes Maß an Sicherheit und Privatsphäre für die Nutzer und wird häufig in Kommunikations-Apps wie WhatsApp und Signal verwendet.

Steganografie

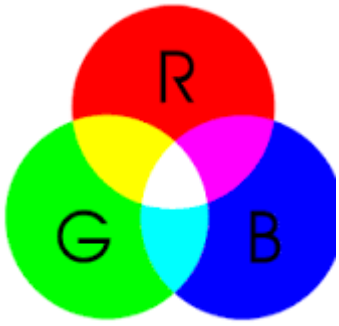
Ein steganografisches Verfahren verheimlicht, dass überhaupt geheime Daten existieren. Der Gedanke dahinter: Wo niemand geheimen Daten vermutet, wird sie auch niemand suchen. Steganografie-Software versteckt die geheimen Daten in einer anderen Datei. Also so genannte Trägerdateien dienen in der Regel meist Bilder, Sound-, Text- und Video-Dateien. Dieses Verfahren kann man nicht nur zum Schutz von Daten benutzen, sondern es wird auch zur Kenntlichmachung von Urheberrechten verwendet. Wer von der Verschlüsselung nichts weiß, nutzt die betreffende Trägerdatei ohne Einschränkungen mit der passenden Anwendung. Nur wer über die Verschlüsselung informiert ist und zudem Zugriff auf den verwendeten Kodierungs-Schlüssel hat, kann die in der Trägerdatei enthaltenen Informationen entschlüsseln und für sich nutzbar machen.

„Vertraue keinem Verschlüsselungsverfahren, das du nicht selbst geknackt hast.“

Beispiel:

Nehmen wir folgendes Bild als Grundlage:

Nun nehmen wir pro Pixel den Hexadezimalen Farbcode und rechnen diesen in das Binäre Zahlensystem um. Dann ergeben sich folgende Werte für die ersten acht Pixel im Bild.



Ein hexadezimaler Farbcode hat 6 Ziffern (z.B.: #ed1c24). Immer zwei Ziffern bilden eine Farbe ab (Rot, Grün, Blau). Eine Hex-Ziffer kann 16 mögliche Zeichen annehmen. Sprich Ein Farbcode (z.B.: Rot) hat nun 16*16 = 256 Möglichkeiten (0-255). Für 256 Möglichkeiten benötigen wir insgesamt 8 Bits. Nachdem wir bei RGB drei Farben darstellen und beliebig kombinieren können, benötigen wir 3×8 = 24Bits. Somit kann man 3^24 = 16777216 = ca. 16,8 Mio. verschiedene Farben mit dem RGB-Modell darstellen.

```
1.Pixel: 111011010001110000100100
2.Pixel: 111011010001110000100100
3.Pixel: 111011010001110000100100
4.Pixel: 111011010001110000100100
5.Pixel: 111011010001110000100100
6.Pixel: 111011010001110000100100
7.Pixel: 111011010001110000100100
8.Pixel: 111011010001110000100100
```

Jetzt nehmen wir beispielsweise einen Text, welcher lautet „Das hier ist ein geheiner Text.“ und rechnen auch hier erstmal das erste Zeichen des Textes in das Binär-System um. Es ergeben sich hierbei folgende Werte für, in dem Fall, den Buchstaben „D“.

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

Laut der ASCII-Tabelle hat der Buchstabe D den dezimalen Wert 68. Binär ergibt sich somit folgende Ziffernfolge:

```
01000100
```

Jetzt wirds interessant. Wir nehmen jetzt quasi pro Pixel im Bild die schwächste Bit (die letzte) und ändern es so um, wie wir es brauchen. Das heißt, bei dem ersten Pixel, nehmen wir, in dem Fall die Null und schauen, ob das mit dem ersten Bit des Buchstaben „D“ übereinstimmt.

Danach kommt der nächste Pixel. Auch hier nehmen wir das die schwächste Bit (auch wieder eine Null) und schauen ob dieses mit dem zweiten Bit des Buchstaben „D“ übereinstimmt. Da die zweite Bit des Buchstaben „D“ allerdings eine 1 ist ändern wir die letzte Bit des zweiten Pixels auf eine 1 ab. Somit verändert sich die Farbe fast garnicht. Hier ein vergleich der Farbe (vorher/nachher):

Vorher:

```
111011010001110000100100
```

Nachher:

```
111011010001110000100101
```

Dieses vorgehen wenden wir nun an jedem Pixel des Bildes vor, bis der ganze Satz im Bild versteckt ist. In folgendem Beispiel ist im Bild der folgende Text versteckt:

```
Steganographie ist im Grunde genommen eine Technik um Daten jeglicher Art zu
verstecken. Das Wort Steganographie stammt vom griechichen Wort „steganos“
ab, was so viel wie Verbergen heißt. Es gibt verschiedene Arten der
Steganographie. Die am verbreitetste Art der Steganographie ist mittlerweile
die technische Steganographie. Hier werden meistens bestimmte Daten
innerhalb eines Bildes versteckt.
Eine spezifische Art dieser Kunst/Technik wurde auch beim weltbekannten
Internet-Rätsel „Cicada 3301“ verwendet.
Im Folgenden werde ich versuchen eine bestimmte Art der Steganographie zu
erklären.
```

Originalbild:

Bild mit Steganografie:

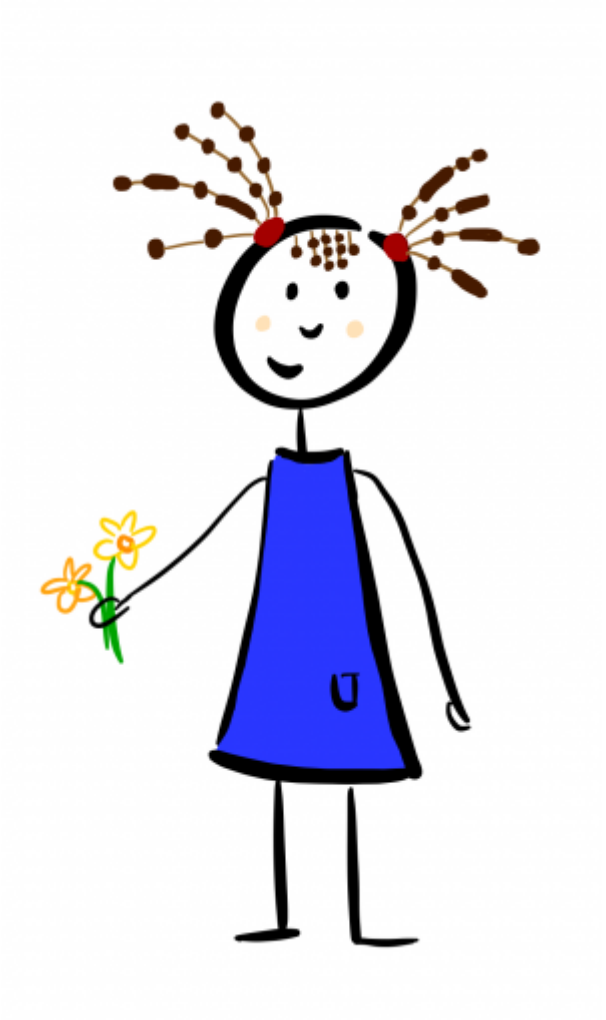
Wie man Sieht kann so eine Nachricht oder eine Datei vollkommen unerkannt übertragen werden. Diese Technik kann so weit geführt werden, dass ganze Bilder in anderen Bildern versteckt werden.

Je mehr Informationen in dem Grundbild sind desto Mehr Daten können darin versteckt werden.

Weitere Beispiele

In dem Bild des Mädchens wurde ihr Name verborgen. Die Nachricht ist mit einem bekannten Verfahren codiert.

Tipp: Um die Nachricht zu lesen, benötigst du den „Morse-Code“, bei dem jeder Buchstabe durch eine Kombination aus Punkten und Strichen ersetzt wird!



Aufgabe 1) Finde die verbotene Nachricht!

Aufgabe 2) Erstelle selbst ein Bild mit einer geheimen Nachricht!

Mehr zu Steganographie

Kryptografie

Die Kryptografie basiert auf mathematischen Verfahren. Die Sicherheit eines Kryptosystems lässt sich also mathematisch beweisen und berechnen. Die mathematische Beweisführung einer gewissen Sicherheit beruht jedoch oft nur auf Annahmen. Zum Beispiel: „Solange diese Bedingung erfüllt ist, ist dieses Verschlüsselungsverfahren sicher.“ Das hat Konsequenzen. Denn ein ungeschickt implementiertes Kryptosystem kann ein eigentlich sicheres Verschlüsselungsverfahren unsicher machen.

Wie sicher ein kryptografisches Verfahren ist, ist zu allen Zeiten immer zu optimistisch gewesen. Prinzipiell neigen wir zur Selbstüberschätzung, was die Sicherheit einer Technik angeht. Dabei zeigt die Erfahrung, dass kein Aufwand zu groß ist, um ein Verfahren zu brechen. Die Fragestellung ist nur, ob sich der Aufwand, in Erwartung des Inhalts verschlüsselter Daten, lohnt.

Kerckhoffs Prinzip

Die Sicherheit des Verschlüsselungsverfahrens beruht nur auf der Geheimhaltung des Schlüssels und nicht auf der Geheimhaltung des Verschlüsselungsverfahrens! Die Sicherheit eines Systems sollte nie allein von der Geheimhaltung der Funktionsweise abhängig sein (sonst: Security by Obscurity).

Das Gegenprinzip „security by obscurity“ besagt, dass man Sicherheit dadurch gewinnen will, indem man den Verschlüsselungsvorgang verschleiert. Dieses Gegenprinzip hat sich vielfach als wenig tauglich erwiesen. Verfahren kann man meist nicht geheimhalten (irgendjemand hält sich nicht an die Geheimhaltung). Zudem ist es oft möglich, durch eine Art Reverse-Engineering das benutzte Verfahren zu rekonstruieren.

Bei der Entwicklung neuer Verfahren versucht man daher gar nicht erst, die Verfahren selbst geheim zu halten. Im Gegenteil, die Verfahren werden zur öffentlichen Diskussion allen Expertinnen zur Verfügung gestellt. Nur die Verfahren, die eine solche Prüfung bestehen, haben eine Chance, in moderen Chiffriersystemen verwendet zu werden.

Gute kryptografische Verfahren erfüllen heute in der Regel also die folgenden Kriterien:

Sie beruhen auf dem Kerckhoffs-Prinzip. Sie werden von Kryptologinnen (bzw. -analytikerinnen) weltweit untersucht. Sie durchlaufen erfolgreich alle möglichen Angriffszenarien.

- 8.1.2.1) Symmetrische Kryptografie
  - 8.1.2.1.1) Cäsar-Chiffre
  - 8.1.2.1.2) Vigenere-Chiffre
  - 8.1.2.1.3) One Time Pad (Vernam-Chiffre)
  - 8.1.2.1.4) Skytale
  - 8.1.2.1.5) DES
  - 8.1.2.1.6) AES
- 8.1.2.2) Asymmetrische Kryptografie
  - 8.1.2.2.1) RSA
- 8.1.2.3) Hybride Chiffriersysteme

Symmetrische Kryptografie/Verschlüsselung

Die Verschlüsselungsverfahren, die mit einem geheimen Schlüssel arbeiten, der zum Ver- und Entschlüsseln dient, nennt man symmetrische Verfahren oder Secret-Key-Verfahren. Üblich sind auch die Bezeichnungen Secret-Key-Kryptografie und Secret-Key-Verschlüsselung. Fast alle symmetrischen Verfahren sind auf ressourcenschonende Umgebungen optimiert. Sie zeichnen sich durch geringe Hardwareanforderungen, geringen Energieverbrauch und einfache Implementierung in Hardware aus.

Prinzip

Die Verschlüsselungsverfahren der symmetrischen Kryptografie arbeiten **mit einem einzigen Schlüssel**, der bei der Ver- und Entschlüsselung vorhanden sein muss. Diese **Verfahren sind schnell** und bei entsprechend **langen Schlüsseln bieten sie auch eine hohe Sicherheit**.

Der **Knackpunkt liegt in der Schlüsselübergabe** zwischen den Kommunikationspartnern. Vor der sicheren Datenübertragung mit Verschlüsselung müssen sich die Kommunikationspartner auf den Schlüssel einigen und austauschen. Wenn der Schlüssel den selben Kommunikationspfad nimmt, wie die anschließend verschlüsselten Daten, dann besteht die Gefahr, dass ein Angreifer in Besitz des Schlüssels gelangt, wenn er die Kommunikation abhört. Wenn der Angreifer den Schlüssel hat, dann kann er nicht nur die Daten entschlüsseln, sondern auch selber Daten verschlüsseln, ohne das es die Kommunikationspartner bemerken. Knackpunkt ist der unsichere Schlüsselaustausch und die Authentifizierung der Kommunikationspartner.

Sicher ist die Schlüsselübergabe nur dann, wenn sich zwei Personen persönlich treffen und den Schlüssel austauschen oder der Schlüssel einen anderen Weg nimmt (Seitenkanal), wie es die Daten tun. Ein Möglichkeit wäre der postalische Weg (Brief, Einschreiben mit Rückschein). Allerdings nicht per E-Mail (Postkarten-Effekt). Zur Unsicherheit trägt außerdem bei, wenn einer der Kommunikationspartner den Schlüssel nur ungenügend sicher aufbewahrt.

Der sichere Schlüsselaustausch ist eines der vielen Probleme der Kryptografie. Mit der asymmetrischen Kryptografie versucht man dieses Problem zu lösen. Weil die asymmetrische Kryptografie weit komplexere Verfahren umfasst, kombinieren die übliche kryptografischen Protokolle sowohl symmetrische als auch asymmetrische Verfahren.

Vorteile

- Gleicher Schlüssel zum Verschlüsseln und Entschlüsseln
- Je zwei Teilnehmer benötigen einen Schlüssel
- Beide müssen den Schlüssel stets geheim halten
- Anzahl der Schlüssel wächst quadratisch mit der Teilnehmerzahl

Nachteile

- Sichere Verteilung des Schlüssels (Telefon, schriftlich,...)
- Nicht geeignet für Digitale Signatur

Symmetrische Verschlüsselungsverfahren

Jede symmetrische Verschlüsselung basiert auf einem bestimmten Algorithmus. Bei einem Verschlüsselungsalgorithmus bzw. Chiffre wird in den Klartext eine Geheiminformation, den Schlüssel, eingebracht und so der Geheimtext gebildet. Der Schlüssel kann ein Passwort, eine geheime Nummer oder auch nur eine zufällige Bitfolge sein.

Monoalphabetische Substitutionschiffren

Die einfachste Art der Verschlüsselung erreicht man, in dem man jeden Buchstaben ein festes Symbol zuordnen. Diese Verfahren sind monoalphabetisch. Sie sind bei genügend Verschlüsselungsmaterial leicht durch eine Häufigkeitsanalyse zu brechen. In jeder Schriftsprache kommen bestimmte Buchstaben häufiger vor. Man kann also mit einfachen statistischen Mitteln eine Kryptoanalyse machen. Mit Computer-Unterstützung geht es automatisch und noch schneller.

- 8.1.2.1.1) Cäsar-Chiffre

Polyalphabetische Substitutionschiffren

Wesentlich schwieriger sind polyalphabetische Geheimtexte. Hier kann ein Buchstabe mehreren Symbole entsprechen. Statistische Verfahren funktionieren hier nicht mehr so einfach.

- 8.1.2.1.2) Vigenere-Chiffre
- 8.1.2.1.3) One Time Pad (Vernam-Chiffre)

Permutationschiffren

Eine Umordnung, eine Permutation einer gegebenen Zeichenfolge, nennt man Permutations- oder Transpositionschiffre. Dies trifft in diesem Fall auf die Skytale zu. Permutationschiffren werden auch als Transposition bezeichnet. Die Skytale ist ein Spezialfall der Transposition. Denkbar wäre nämlich eine Permutationschiffre, die zur Erstellung des Geheimtextes erst den ersten, dann den 47-ten, danach den 32-ten Buchstaben nimmt, usw. Bei der Skytale wird jedoch, wie oben als Matrix betrachtet, die Nachricht zeilenweise aufgetragen und chiffriert liegt diese spaltenweise vor. Die Skytale ist also letztendlich eine einfache Matrixtransposition.

Bei einer Permutations-Chiffre werden somit die Buchstaben den Klartext nicht ersetzt sondern durcheinander gewürfelt. Fast man zB immer 5 Buchstaben des Klartextes zusammen und lässt sich durch die Permutations-Chiffre mit dem Schlüssel (4,1,2,5,3), dann erhält man zB folgenden Chiffretext:

```
IE SBGE TZIW EARNT LVU ENNUTS: EHOLEC, ZDI UE EENB DGRIENS; NWS ANI
EFAEANNG.

ES GIBT ZWEI ARTEN VON LEUTEN: SOLCEH DIE ZU ENDE BRINGEN, WAS SIE ANFANGEN.
```

Die Art der Verschlüsselung lässt sich durch Probieren – abhängig von der Schlüssellänge – mehr oder weniger schnell knacken. Auch die Häufigkeitsanalyse liefert wieder Rückschlüsse über die verwendete Sprache etc.

- 8.1.2.1.4) Skytale

Operationen

Alle gängigen symmetrische Verfahren arbeiten ausschließlich mit Bit-weisen Operationen. Hier werden Schlüssel, Klartext und Geheimtext in Form von Bitfolgen verarbeitet. In dem die Funktionen nahezu beliebig miteinander kombiniert werden, lassen sich neu symmetrische Verfahren in nahezu beliebiger Zahl entwickeln und mit bekannten Angriffen auf Schwächen testen. In der Regel kombinieren symmetrische Verschlüsselungsalgorithmen Substitutionschiffren und Permutationschiffren miteinander und wiederholen den Vorgang mehrmals (Runden), wobei eine härtere Verschlüsselung entsteht. Typische Bestandteile von symmetrischen Verschlüsselungsalgorithmen sind:

- Exklusiv-oder-Verknüpfung
- Permutation: Reihenfolge einer Bit-Folge wird verändert.
- Substitution: Eine Bit-Folge wird durch eine andere ersetzt.

Erfahrungsgemäß sind für eine wirkungsvolle Verschlüsselung keine aufwendigen Funktionen notwendig. Insbesondere beim Hardware-nahen Programmieren oder der Implementierung in Hardware ist das von Vorteil, weil sich so eine hohe Geschwindigkeit erreichen lässt. Beim praktischen Einsatz von Verschlüsselungsalgorithmen stellt sich auch immer die Frage, wie groß die Rechenleistung für die Verschlüsselung ist. Generell gilt, je schneller ein Verschlüsselungsverfahren arbeitet, desto niedriger sind die Hardwarekosten.

Moderne(re) Verschlüsselungsverfahren

Bei den symmetrischen Verschlüsselungsverfahren gilt der AES als Maß der Dinge. Es gibt aber auch weitere...

- 8.1.2.1.5) DES
- 8.1.2.1.6) AES
- 3DES - Triple DES
- IDEA - International Data Encryption Algorithm
- RC4 (Rivest-Cipher 4)

- Blowfish (von Bruce Schneier)
- RC5, RC5a, RC6 (Rivest-Cipher 5 bzw. 5a bzw. 6)
- A5 (GSM)
- Serpent
- Twofish (von Bruce Schneier)
- MARS
- SAFER/SAFER+
- CAST (Carlisle Adams und Stafford Tavares)
- MAGENTA
- MISTY1
- Camellia
- Ascon

Quellen

- Elektronik Kompendium
- Kryptografie.de
- Cryptool
- Wikipedia

Cäsar Chiffre

Die Cäsar-Chiffre ist eines der einfachsten, aber auch unsichersten Verfahren, um Texte zu verschlüsseln. Das Verfahren wurde nach dem römischen Kaiser Julius Cäsar benannt, der auf diese Weise bereits vor über 2000 Jahren Nachrichten verschlüsselt haben soll.

Die Cäsar-Chiffre ist eine monoalphabetische Substitution, das heißt, jeder Buchstabe des Textes wird durch genau einen anderen Buchstaben des Alphabets ersetzt. Dieser Austausch geschieht jedoch nicht zufällig, sondern basiert auf zyklischer Rotation des Alphabets um *k* Zeichen, wobei *k* der verwendete Schlüssel ist.

Als eines der einfachsten und unsichersten Verfahren dient es heute hauptsächlich dazu, Grundprinzipien der Kryptologie anschaulich darzustellen. Der Einfachheit halber werden oftmals nur die 26 Buchstaben des lateinischen Alphabets ohne Unterscheidung von Groß- und Kleinbuchstaben als Alphabet für Klartext und Geheimtext verwendet und Sonderzeichen, Satzzeichen usw. nicht beachtet.

Verschlüsselung

Die Verschlüsselung einer Nachricht erfolgt buchstabenweise mit einem Schlüssel *k* aus der Menge Z_{26} = *{0,1,2,3...25}*, wobei der Wert *k = 0* nicht sinnvoll ist, da der Originaltext in diesem Fall keine Änderung erfährt.

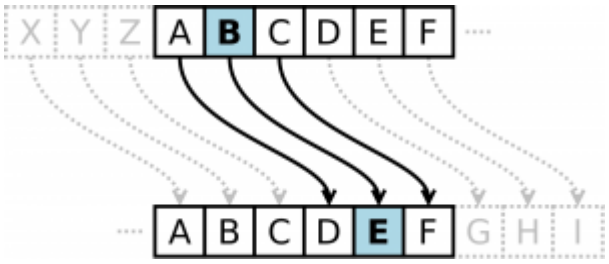
Für einen gegebenen Buchstaben wird zunächst anhand der folgenden Tabelle seine Position *m* im Alphabet bestimmt.

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

Anschließend erhält man den Wert *c* des verschlüsselten Buchstaben durch folgende kurze Berechnungsformel:

```
c = (m+k) mod 26
```

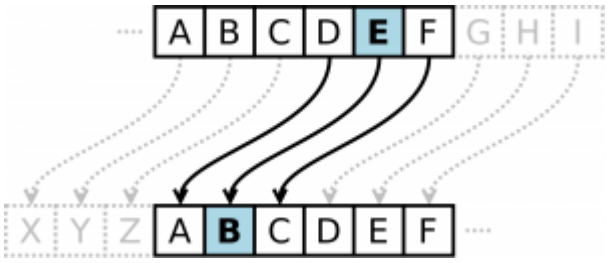
Mit Hilfe obiger Tabelle kann dieser Wert *c* wieder in einen Buchstaben transformiert werden.



Entschlüsselung

Die Entschlüsselung einer Nachricht erfolgt ähnlich wie die Verschlüsselung mit Schlüssel , wir verwenden jedoch die Formel:

```
m = (26+c-k) mod 26
```



Beispiel

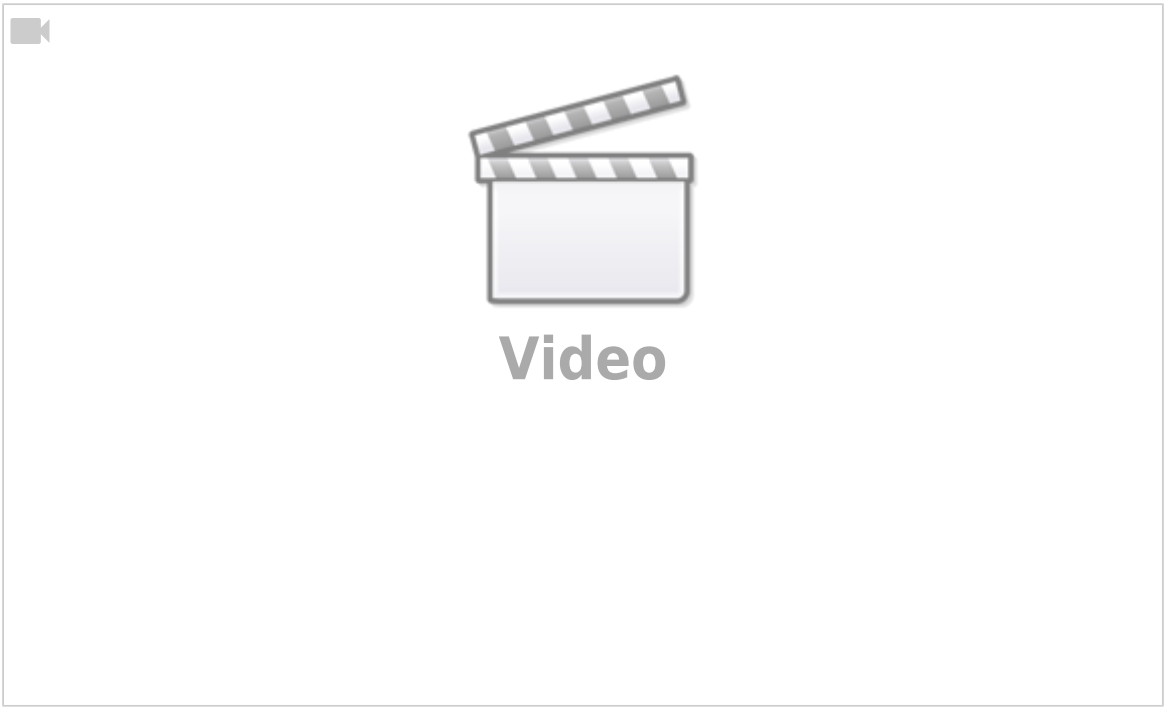
Der Satz „OTTO KOMMT“ wird mit dem Schlüssel *k=3* verschlüsselt.

unverschlüsselt	O	T	T	O		K	O	M	M	T
m	14	19	19	14		10	14	12	12	19
$c \equiv (m + k) \bmod 26$	17	22	22	17		13	17	15	15	22
verschlüsselt	R	W	W	R		N	R	P	P	W

Statt nur über dem Alphabet Z_{26} kann man analog allgemeine Cäsar-Chiffre über beliebigen endlichen Alphabeten Z_{26} = {0,1,..a-1} definieren.

Klartext: H A L L O

Geheimtext: K D O O R



Vignere Chiffre

Die Vigenère-Chiffre (auch: Vigenère-Verschlüsselung) ist eine aus dem 16. Jahrhundert stammende Handschlüsselmethode zur Verschlüsselung von geheim zu haltenden Textnachrichten.

Es handelt sich um ein monographisches polyalphabetisches Substitutionsverfahren. Der Klartext wird in Monogramme (Einzelzeichen) zerlegt und diese durch Geheimtextzeichen substituiert (ersetzt), die mithilfe eines Kennworts aus mehreren (poly) unterschiedlichen Alphabeten des „Vigenère-Quadrats“ ausgewählt werden. Dabei handelt es sich um eine quadratische Anordnung von untereinander stehenden verschobenen Alphabeten (siehe Bild).

Die Vigenère-Chiffre steht im Gegensatz zu den einfacheren monoalphabetischen Substitutionsmethoden, bei denen nur ein einziges (mono) Alphabet verwendet wird. Aufgrund ihrer für die damalige Zeit als besonders hoch eingeschätzten kryptographischen Sicherheit wurde sie auch als le chiffre indéchiffrable (frz. für „die unentzifferbare Chiffre“) bezeichnet, eine aus damaliger Sicht vielleicht zutreffende, aber aus heutiger Sicht falsche Beurteilung.

Methode

Ausgehend vom Standardalphabet mit seinen 26 Großbuchstaben werden alle möglichen Caesar-verschobenen Alphabete daruntergeschrieben. Man erhält eine quadratische Anordnung von 26 × 26 Buchstaben, ursprünglich als Tabula recta, später auch als carré de Vigenère (frz. für „Vigenère-Quadrat“) bezeichnet. In der folgenden Darstellung sind der Deutlichkeit halber oberhalb des eigentlichen Quadrats eine Zeile mit den Klartextbuchstaben und links eine Spalte mit den Schlüsselbuchstaben ergänzt worden, die prinzipiell nicht benötigt werden.

Vigenère-Quadrat

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

Zur Verschlüsselung eines Klartextes wie beispielsweise des Satzes „Werde Mitglied bei Wikipedia“ benötigt der Verschlüssler zunächst einen Schlüssel. Idealerweise sollte dieser möglichst lang sein und aus einer möglichst „zufälligen“ Buchstabenfolge bestehen. Erreicht die Länge des Schlüssels die des Klartextes und wird der Schlüssel nicht mehrfach verwendet, dann erhält man ein tatsächlich „unknackbares“ Verfahren, wie es aber erst Jahrhunderte später, im Jahr 1882, vom amerikanischen Kryptologen Frank Miller (1842–1925) vorgeschlagen wurde, und das heute als One-Time-Pad (Abkürzung: OTP, deutsch: „Einmalschlüssel-Verfahren“) bezeichnet wird. Zur Zeit von Vigenère und noch bis ins 20. Jahrhundert hinein wurden allerdings regelmäßig relativ kurze und häufig auch leicht

zu erratende Schlüssel benutzt, die zudem mehrfach verwendet wurden. Ein Beispiel wäre die Verwendung von WILLKOMMEN als Schlüsselwort.

Als praktisches Hilfsmittel kann der Verschlüssler den zu verschlüsselnden Text in eine Zeile schreiben und darüber das Kennwort so oft wiederholen, wie es nötig ist:

```
WILLKOMMEN WILLKOMMEN WILLK
WerdeMitgl iedbeiWiki pedia
```

Die entsprechenden Geheimtextbuchstaben kann er nun leicht mithilfe des Vigenère-Quadrats ermitteln. Dazu sucht er den Kreuzungspunkt der durch den jeweiligen Schlüsselbuchstaben gekennzeichneten Zeile und der Spalte des Quadrats, die oben durch den Klartextbuchstaben gekennzeichnet ist. Beispielsweise zur Vigenère-Verschlüsselung des ersten Buchstabens W des Textes sucht er den Kreuzungspunkt der Zeile W mit der Spalte W und findet als Geheimtextbuchstaben das S. Der auf diese Weise vollständig verschlüsselte Geheimtext lautet:

```
SMCOOAUFKY EMOMOWIUOV LMOTK
```

Üblicherweise wird er in Gruppen fester Länge, beispielsweise in Fünfergruppen übertragen. Diese Maßnahme dient auch dazu, die Länge des Kennworts (hier zehn) nicht zu verraten. Der zu übermittelnde Geheimtext lautet hier:

```
SMCOO AUFKY EMOMO WIUOV LMOTK
```

Der befugte Empfänger ist, wie der Absender, im Besitz des geheimen Kennworts (hier: WILLKOMMEN) und kann durch Umkehrung der oben beschriebenen Verschlüsselungsschritte aus dem Geheimtext durch Entschlüsselung mithilfe des Kennworts den ursprünglichen Klartext wieder zurückgewinnen:

```
SMCOOAUFKYEMOMOWIUOVLMOTK
WILLKOMMENWILLKOMMENWILLK
WERDEMITGLIEDBEIWIKIPEDIA
```

Vigenere Chiffre Erklärung

Kryptoanalyse

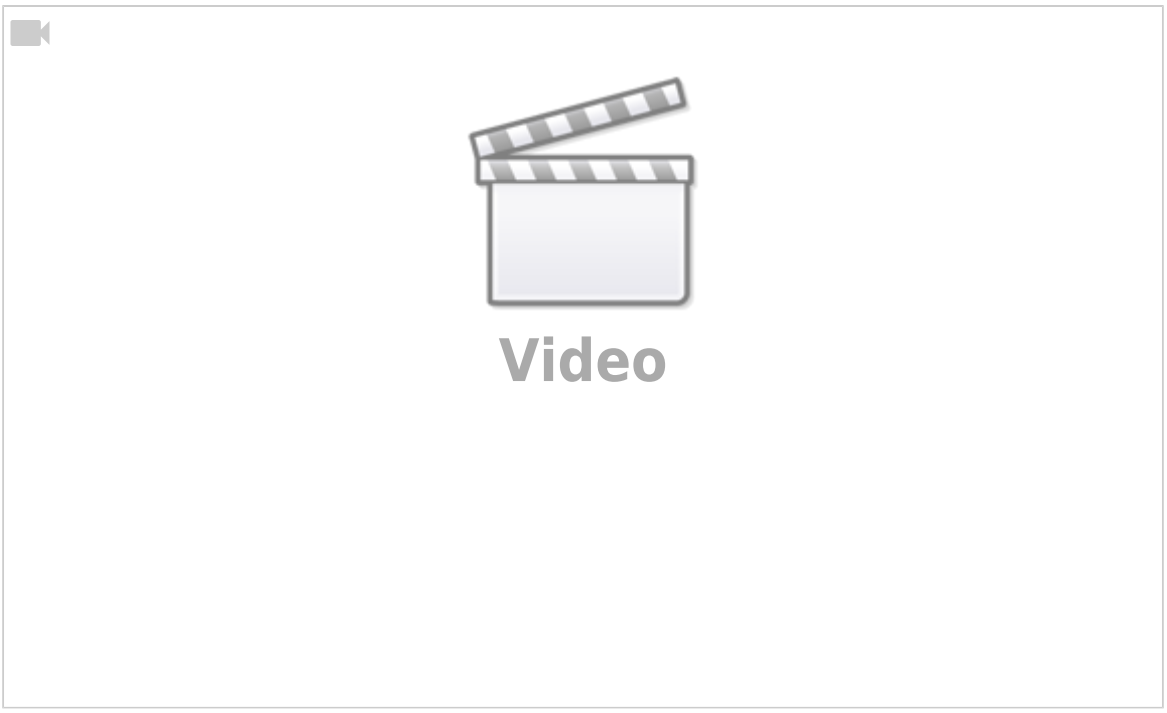
Vorteile einer polyalphabetischen Methode wie der Vigenère-Chiffre gegenüber den in den damaligen Jahrhunderten üblichen einfachen monoalphabetischen Methoden – dazu gehören auch die damals sehr beliebten Nomenklatoren – ist das durch die Verwendung von vielen unterschiedlichen Alphabeten bewirkte Abschleifen des bei den monoalphabetischen Verfahren so verräterischen Häufigkeitsgebirges. Der systematische Wechsel der Alphabete stärkt das Verfahren gegenüber statistischen Angriffsmethoden. Auch der erst im 20. Jahrhundert entwickelte Koinzidenzindex, ein universell einsetzbares kryptanalytisches Hilfsmittel, wird bei polyalphabetischen Verfahren wesentlich abgeschwächt. Lange wurde – abgesehen von Ausnahmen, in denen der Codeknacker das Schlüsselwort oder Teile des Klartextes erraten konnte – keine systematische Angriffsmethode gegen die Vigenère-Verschlüsselung gefunden, die sich über die Jahrhunderte den Ruf einer „unknackbaren Chiffre“ erwarb. Dennoch wurde sie nur selten verwendet und stattdessen lieber auf die althergebrachten Verfahren, wie Nomenklatoren, zurückgegriffen, wohl auch, weil viele Anwender die

Chiffre als zu kompliziert in der Anwendung empfanden.

Im Jahr 1854 fand der englische Wissenschaftler Charles Babbage (1791–1871) eine Lösung der Chiffre, die er jedoch nie publizierte. Der Erste, der eine allgemeingültige Angriffsmethode auf die Vigenère-Chiffre beschrieb, war der preußische Infanteriemajor und Kryptologe Friedrich Wilhelm Kasiski (1805–1881). Er veröffentlichte 1863 in Berlin sein Buch „Die Geheimschriften und die Dechiffrir-Kunst“ und erläuterte darin seine Idee zur Entzifferung von Vigenère-verschlüsselten Texten. Seine Entzifferungsmethode ist noch heute unter seinem Namen als Kasiski-Test bekannt. Als Erstes ist die Länge des verwendeten Schlüsselworts zu ermitteln. Dazu durchsuchte Kasiski den Geheimtext nach Buchstabenfolgen der Länge zwei (Bigramme) oder länger (Trigramme, Tetragramme etc.), die mehrmals vorkommen, genannt: „Doppler“. Anschließend bestimmte er den Abstand zwischen den Dopplern. Er erzeugte so eine möglichst vollständige Liste mit im Geheimtext auftretenden Dopplern und deren Abständen. In dieser suchte er mithilfe der Faktorisierung (Primfaktorzerlegung) nach gemeinsamen Längen, um so auf die vermutliche Schlüsselwortlänge zu schließen. Im Cryptologia-Artikel Breaking Short Vigenère Ciphers (siehe Literatur) ist die wichtige Seite 41 aus Kasiskis Buch abgebildet.[9] Nach der Untersuchung seines Vigenère-verschlüsselten Beispieltextes mit 180 Buchstaben zieht er das Fazit: „Hier kommt der Faktor 5 am häufigsten vor, der Schlüssel muß demnach 5 Buchstaben enthalten.“

Hat man die Schlüssellänge gefunden, so kann man im zweiten Schritt der Entzifferung den Geheimtext in seine Bestandteile zerlegen, die mit jeweils demselben Alphabet verschlüsselt wurden. In Kasiskis Beispielfall würde man den ersten, sechsten, elften Buchstaben und so fort als erste Gruppe betrachten. Die zweite Gruppe besteht aus dem zweiten, siebten, zwölften Buchstaben und so fort. Die dritte aus dem dritten, achten, dreizehnten und so weiter. Innerhalb jeder Gruppe liegt eine einfache Caesar-Verschlüsselung vor, die mithilfe der Häufigkeitsanalyse leicht zu knacken ist. In vielen Fällen entspricht schlicht der am häufigsten auftretende Geheimtextbuchstabe jeder Gruppe dem Klartext-„e“, also dem in den meisten europäischen Sprachen häufigsten Buchstaben. Hat man das „e“ identifiziert, dann ergeben sich unmittelbar alle anderen Buchstaben, denn die Vigenère-Chiffre benutzt ja nur verschobene Alphabete und keine verwürfelten, wie es der Namensgeber eigentlich vorgeschlagen hatte.

Nach „Rohrbachs Forderung“ sollte der Codeknacker zum Schluss seiner Arbeit noch versuchen, das Schlüsselwort zu erschließen. Erst dann gilt seine Arbeit als erfolgreich beendet. Im Idealfall gelingt ihm dies einfach mit Kenntnis des Klartextes durch anschließendes direktes Ablesen im Quadrat. In der Praxis wurden jedoch nicht immer plump einfache Wörter als Schlüssel benutzt. Dann gilt es, auch noch den Algorithmus zu erschließen, nach dem der Verschlüssler das Schlüsselwort (beispielsweise aus einem Merksatz) bildet und möglichst auch, wie und in welchem Rhythmus er es wechselt.



One Time Pad (Vernam-Chiffre)

Das One Time Pad (Abk. OTP, dt. Einmalverschlüsselung) Verschlüsselungsverfahren, auch Vernam Verschlüsselung nach seinem Erfinder.

Vernam arbeitete bei der US-amerikanischen Telefongesellschaft AT&T und war dort mit der damals noch neuen Fernschreiber-Technik betraut. Ein Problem war, dass man Fernschreiben leicht abhören konnte - besonders, wenn diese per Funk übertragen wurden. 1917 hatte er eine Idee für die Lösung: er wollte die Bits des Baudot-Codes, den man damals für Fernschreiben nutzte und die aus einer Null oder einer Eins bestanden, verschlüsseln, indem er jedes Bit mit einem zufälligen, anderen Bit kombinierte. Dazu benutzte er einen zweiten Lochstreifen mit Zufallsmuster zum ersten mit der Botschaft. Zuerst nahm er nur einen kurzen Lochstreifen, dessen Ende er an den Anfang klebte und so einen sich wiederholenden Endlosstreifen erhielt. Doch dann merkte er, dass absolute Sicherheit nur ein Schlüsselstreifen bieten konnte, der genau so lang war wie der Lochstreifen mit dem Klartext.

Der amerikanische Major (und später General) Joseph O. Mauborgne setzte die Idee 1918 als Erster für militärische Zwecke um und erweiterte sie um die Prämisse, dass ein Schlüsselcode zufällig und nur einmal benutzt werden darf. Das Verfahren wurde als One-time-system bekannt. Aus Gründen der Praktikabilität verwendete er allerdings einen sich wiederholenden Schlüssel. Alsbald beschäftigten sich auch die Deutschen mit dem Verfahren und setzten es im diplomatischen Dienst der Weimarer Republik ein.

One Time Pad ist also ein Verfahren, bei dem jedes Zeichen des Klartextes mit einem Zeichen eines Schlüssels kombiniert wird, um zu einem Chiffrat zu gelangen. Dies bedeutet aber auch, dass der Schlüssel genau so lang sein muss wie der zu verschlüsselnde Text.

Damit das Verfahren sicher ist, ist es außerdem wichtig, dass der Schlüssel rein zufällig ist und dass der Schlüssel nur ein einziges mal verwendet wird. Denn würde der Schlüssel zweimal verwendet und wäre dem Gegner bekannt, dass zweimal derselbe Schlüssel für zwei unterschiedliche Klartexte verwendet wurden, so ließe sich ein Datenstrom aus den Differenzen erstellen, der wiederum durch Häufigkeitsanalyse der verwendeten Zeichen angreifbar wäre.

Von Prinzip her könnte man die One Time Pad Verschlüsselung auch als polyalphabetische Substitution bezeichnen, bei dem für jedes Zeichens des Klartextes ein anderer Schlüssel verwendet wird.

Auf der anderen Seite stellt die Länge des Schlüssels doch einige Anforderungen bei längeren Texten, so dass der Schlüssel wohl zumeist der Output eines Pseudo-Zufallsgenerators sein wird, wobei dann der Terminus Stromchiffre wieder passen würde.

Ein Vorteil des One Time Pad Verfahrens ist außer der Sicherheit bei richtiger Anwendung auch, dass es leicht mit Papier und Bleistift bewerkstelligt werden kann. So war es im kalten Krieg unter Geheimdiensten oft eingesetzt. Dabei wurden die Zeichen einer Geheimbotschaft mittels Dekodierschablonen zu Ziffern umgewandelt, die dann mittels langen Ziffernkolonnen in einem Heft oder Block, sogenannte Wurmtabellen kombiniert wurden.

Z. B.: Klartext 6, Schlüsselziffer 7 ergibt 13 → 3 (Addition Modulo 10). Mit der Umkehrrechnung (Subtraktion Absolut), hier also 7 - 13 = -6 → 6 konnte dann wieder auf den Klartext entschlüsselt werden. Ein einmal verwendete Wurmtabelle wurde nach dem Verschlüsseln dann nach einem festen Muster (z. B. alle angefangenen Blätter) vernichtet.

Auch der Heiße Draht (das sogenannte Rote Telefon) zwischen dem amerikanischen Präsidenten und dem sowjetischen Generalsekretär wurde durch ein One Time Pad Verfahren gesichert.

Das OTP lässt sich auch einfach per Computer realisieren. Die Bits der dort vorliegende Binärdaten werden dann aber meistens mittels XOR verknüpft, weil dies weniger Rechenoperationen erfordert. Außerdem ist XOR eine reversible Operation und kann so für Ver- und Entschlüsselung zugleich eingesetzt werden.

Der Hauptnachteil des OTP in der modernen Umgebung liegt in der erforderlichen Schlüssellänge. Wollte man zum Beispiel eine gesamte Festplatte verschlüsseln, so bräuchte man eine zweite, mindestens genau so große, die den Schlüssel enthält. Noch dazu müsste der Schlüssel aus echten Zufallszahlen und nicht aus berechneten Pseudozufallszahlen bestehen, um wirklich sicher zu sein. Dies würde viel Aufwand bedeuten. Außerdem kann der Schlüssel bei dieser Größe nicht mehr gemerkt werden, so dass er an Medien gebunden ist, die dem Feind in die Hände fallen könnten.

Darum hat das OTP Verfahren in der Moderne zunehmend an Bedeutung verloren, insbesondere, wenn größere Datenmengen verschlüsselt werden müssen.

Beispiel

Eine einfache Handmethode zur Verschlüsselung ist beispielsweise die buchstabenweise Addition von Klartext und Schlüssel. Hierzu ersetzt man zunächst mithilfe einer beliebigen Substitutionstabelle die Buchstaben des Klartextalphabets durch Zahlen. Im einfachsten Fall ordnet man den 26 Großbuchstaben des lateinischen Alphabets Zahlen zu, die ihrer Position im Alphabet entsprechen. Mit anderen Worten, man nummeriert das Alphabet wie folgt durch:

```
A B C D E F G H I J  K  L  M  N  O  P  Q  R  S  T  U  V  W  X  Y  Z
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
```

Jetzt ist eine buchstabenweise Addition leicht möglich. Beispielsweise ergibt die Addition von A und F den Buchstaben G, entsprechend ihren Platznummern 1 + 6 = 7. Falls die Summe den Wert 26 überschreiten sollte, so zieht man einfach 26 ab (Modulo-Operation) und erhält so wieder einen der 26 Alphabetbuchstaben. Beispielsweise X plus U ist numerisch 24 + 21 = 45, nach Abziehen von 26 ergibt sich 19 und damit der Buchstabe S, also X + U = S.

Die Zusammenhänge bei der Addition von Buchstaben lassen sich an der folgenden Tabelle, die Ähnlichkeit mit einer klassischen Tabula recta (Vigenere Quadrat) hat, übersichtlich darstellen.

Zur Verschlüsselung wird man einen zufälligen Schlüssel benutzen, der in diesem Beispielfall passenderweise ebenfalls aus den 26 Großbuchstaben zusammengesetzt ist und dessen Länge (mindestens) der Länge des zu verschlüsselnden Klartextes entspricht. Entscheidend für die Sicherheit der Verschlüsselung ist, dass die einzelnen Buchstaben des Schlüssels wirklich zufällig verteilt sind, unvorhersagbar sind und in keinerlei Zusammenhang untereinander stehen. Als Beispiel für einen zufälligen Schlüssel dient die folgende Buchstabenfolge:

```
S = WZSLXWMFQUDMPJLYQOXXB
```

Der Schlüssel S ist in diesem Beispiel recht kurz, er umfasst nur 21 Buchstaben und ist bei bestimmungsgemäßer Verwendung sehr schnell verbraucht, nämlich bereits nach Verschlüsselung eines Textes aus 21 Buchstaben.

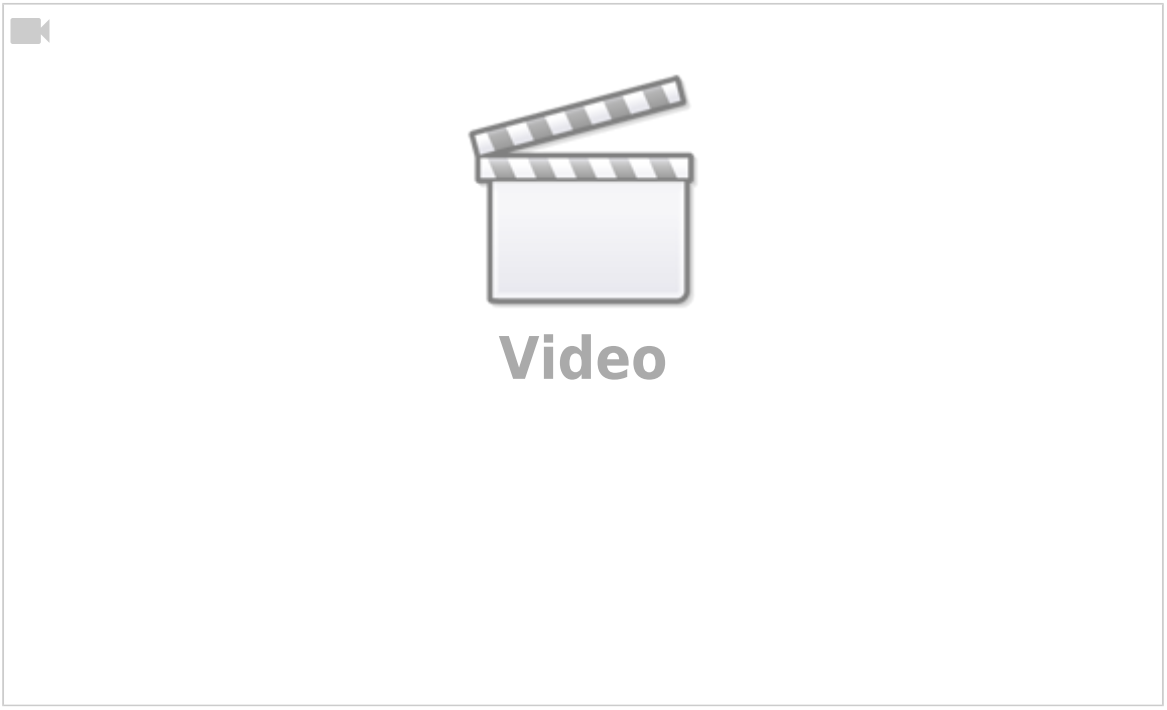
Beispielsweise soll der folgende Klartext K verschlüsselt werden:

```
K = ANGRIFFIMMORGENGRAUEN
```

Zur Verschlüsselung werden Klartext K und Schlüssel S, wie oben erläutert, buchstabenweise addiert. Als Summe (K + S = G) erhält man nach der so durchgeführten Einmalverschlüsselung den Geheimtext G:

```
G = XNZDGCSODHSEWOZFIPSCP
```

Der im Ergebnis erhaltene Geheimtext G ist von einem Zufallstext nicht zu unterscheiden und kann prinzipiell mit keiner noch so gearteten kryptanalytischen Angriffsmethode (weder jetzt noch in Zukunft) entziffert werden. Allein die Kenntnis des Schlüssels S erlaubt es, aus dem Geheimtext G durch Subtraktion des Schlüssels wieder den Klartext K zu gewinnen. Ohne den Schlüssel kann man prinzipiell alle denkbaren und mehr oder weniger sinnvollen Buchstabenkombinationen aus 21 Buchstaben konstruieren. Theoretisch könnte ein Angreifer dies probieren. Das wären aber 26^21 = 518 131 871 275 444 637 960 845 131 776 Möglichkeiten.



Skytale

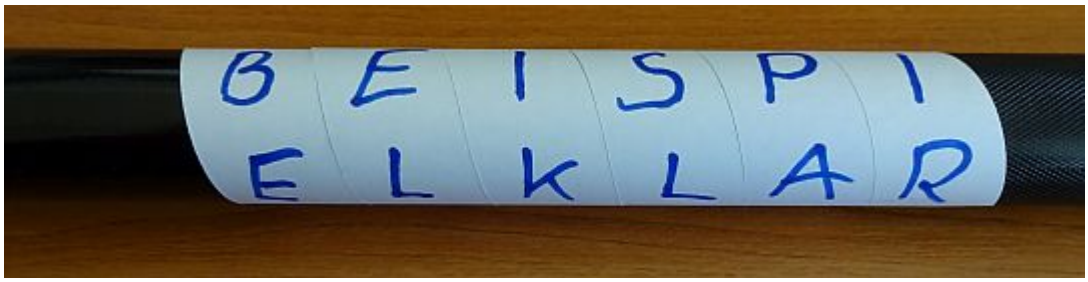
Die Skytale von Sparta (griech.: „scytale“; Stock, Stab) ist das älteste (5. Jh. v. Chr.) bekannte militärische Verschlüsselungsverfahren und basiert auf einem Stock mit einem bestimmten Durchmesser, auf den ein Lederstreifen wendelförmig gewickelt wurde. Dann wurde die Nachricht quer über den Stab auf das Leder geschrieben. Nach dem Abwickeln waren dann alle Buchstaben durcheinander und konnten erst wieder gelesen werden, wenn sie um einen Stab mit dem richtigen Durchmesser gewickelt wurden.

Der Durchmesser entspricht dem Versatz, also dem Schlüssel dieser Transpositions-Chiffre. Bitte beachten Sie, dass auch Leerzeichen mitkodiert werden, da diese ja eine Leerraum darstellen und somit einen Versatz bedeuten. Sollen keine Leerzeichen mitkodiert werden, löschen Sie diese vorher.

Die Chiffre ist nicht sonderlich sicher, denn man einfach Stöcke verschiedener Durchmesser ausprobieren. Oder mathematisch die Versätze durchrechnen. Dann muss man nur noch den Klartext erkennen.

Den Lederstreifen, auf den die Nachricht geschrieben war, konnte man auch umgedreht als Gürtel tragen, so dass er nicht weiter auffiel. Damit war die Skytale auch eine frühe Form der Steganografie.

Beispiel



Klartext:

```
BEISPIELKLARTEXT
```

Schlüssel:

```
6
```

Kodiert:

```
BETELEIKXSLTPAIR
```

Chiffrierung Versatz 6:

```
1 2 3 4 5 6
B E I S P I
E L K L A R
```

```
T E X T
^ ^ ^ ^ ^ ^ --- spaltenweise auslesen

BET ELE IKX SLT PA IR
```

Dechiffrierung per Bruteforce:

```
01: BETELEIKXSLTPAIR
02: BXESTLETLPEAIIKR
03: BITEKPTXAESILLRE
04: BLXPEESATILIEKTR
05: BLKLAEEXTITISPRE
06: BEISPIELKLARTEXT
07: BEIXLPIELKSTARTE
08: BTLIXLPIEEEKSTAR
09: BTLIXLPIREEEKSTA
10: BTLIXLPAIREEEKST
11: BTLIXLTPAIREEEKS
12: BTLIXSLTPAIREEEK
13: BTLIKXSLTPAIREEE
14: BTLEIKXSLTPAIREE
15: BTELEIKXSLTPAIRE
16: BETELEIKXSLTPAIR
```

DES (Data Encryption Standard)

- Entwickelt Anfang der 70er Jahre bei IBM („Lucifer“, „Feistelchiffre“)
- 1977 wurde er von der US-Standardisierungsbehörde NIST (National Institute of Standards and Technology) als Standard anerkannt
- NSA an der Entwicklung beteiligt und die Designkriterien unter Verschluss gehalten sowie Schlüssellänge verkürzt
- DES gilt als Muster aller modernen Chiffren
- DES ist eine Blockchiffre (64 Bit Blöcke)
- Schlüssellänge 56 Bit (+8 Bit Prüfsumme)
- Entwickelt für Hardwareverschlüsselung

Der Data Encryption Standard (DES) ist eine Blockchiffre mit 8 Byte Blocklänge und weit verbreiteter symmetrischer Verschlüsselungsalgorithmus und wurde als offizieller Standard für die US-Regierung im Jahr 1977 bestätigt und wird seither international vielfach eingesetzt. Seine Entstehungsgeschichte hat wegen der Beteiligung der NSA am Design des Algorithmus immer wieder Anlass zu Spekulationen über seine Sicherheit gegeben. Heute wird DES aufgrund der verwendeten Schlüssellänge von nur 56 Bits für viele Anwendungen als nicht ausreichend sicher erachtet.

Ursprünglich hieß der bei IBM unter der Leitung von Horst Feistel entwickelte Algorithmus Lucifer und bot eine Schlüssellänge von 16 Byte / 128 Bit. Die NSA soll dafür verantwortlich gewesen sein, dass die Schlüssellänge für DES auf 56 Bit gekürzt wurde, weil dies die Schlüssellänge ist, die man mit den Supercomputern bei der NSA in den 1970er-Jahren gerade noch so per Brute Force hätte knacken können.

Die Schlüssellänge kann durch Mehrfachanwendung des DES jedoch auf einfache Weise vergrößert werden. Als Triple-DES, auch als TDES, 3DES oder DESede bezeichnet, wird der DES weiterhin am häufigsten, zum Beispiel von Banken in Chipkartenanwendungen, eingesetzt, obwohl der TDES als offizieller Standard für die USA durch den Advanced Encryption Standard (AES) abgelöst wurde.

Weil die Schlüssellänge nur 56 Bit beträgt, konnte DES bereits durch Brute-Force-Angriffe gebrochen werden, indem systematisch alle möglichen Schlüssel (256 = ca. 72 Billiarden) getestet wurden. Die EFF baute 1998 eine etwa 250.000 Dollar teure Maschine mit dem Namen Deep Crack. Dieser Superrechner enthielt 1536 spezielle Krypto-Chips und konnte pro Sekunde etwa 88 Milliarden Schlüssel testen. Im Juli 1998 gelang es mit dieser Maschine, einen DES-Code in 56 Stunden zu knacken.

Die einzige andere öffentlich bekannte Maschine zum Brechen von DES ist COPACOBANA. Sie wurde 2006 an den Universitäten Bochum und Kiel gebaut. Im Gegensatz zu Deep Crack besteht eine COPACOBANA aus rekonfigurierbaren Hardware-Bausteinen, sog. FPGAs. COPACOBANA kann 65 Milliarden DES-Schlüssel pro Sekunde testen, woraus sich eine durchschnittliche Suchzeit von 6,4 Tagen für eine DES-Attacke ergibt. Durch den Einsatz rekonfigurierbarer Hardware kann COPACOBANA auch zum Brechen anderer Chiffren wie A5 eingesetzt werden. Die Material- und Herstellungskosten von COPACOBANA belaufen sich auf „nur“ etwa 10.000 Dollar.

Verfahren

Bei DES handelt es sich um einen symmetrischen Algorithmus, das heißt zur Ver- und Entschlüsselung

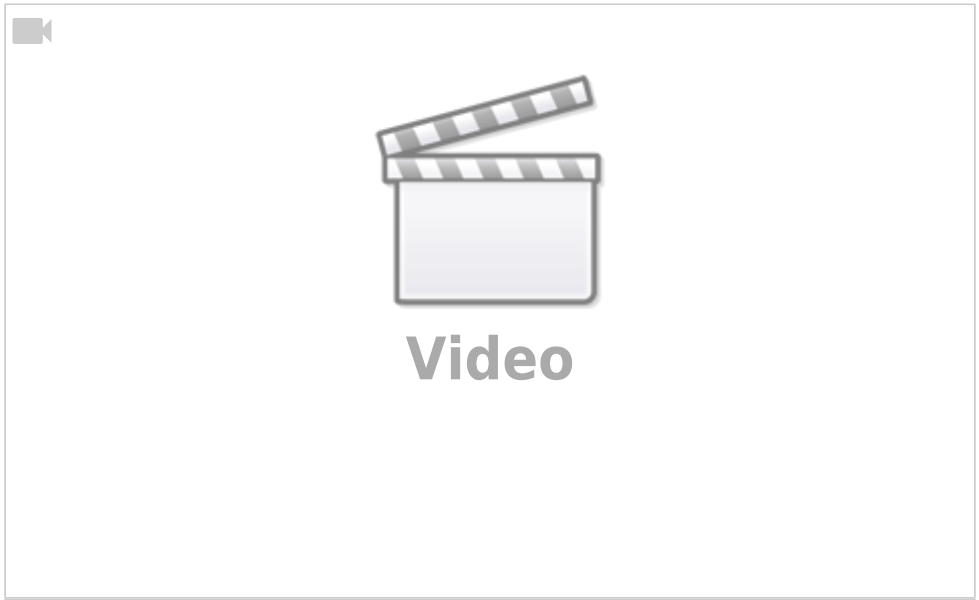
wird derselbe Schlüssel verwendet. DES funktioniert als Blockchiffre, jeder Block wird also unter Verwendung des Schlüssels einzeln chiffriert, wobei die Daten in 16 Runden von Substitutionen und Transpositionen (Permutation) nach dem Schema von Feistel verwürfelt werden.

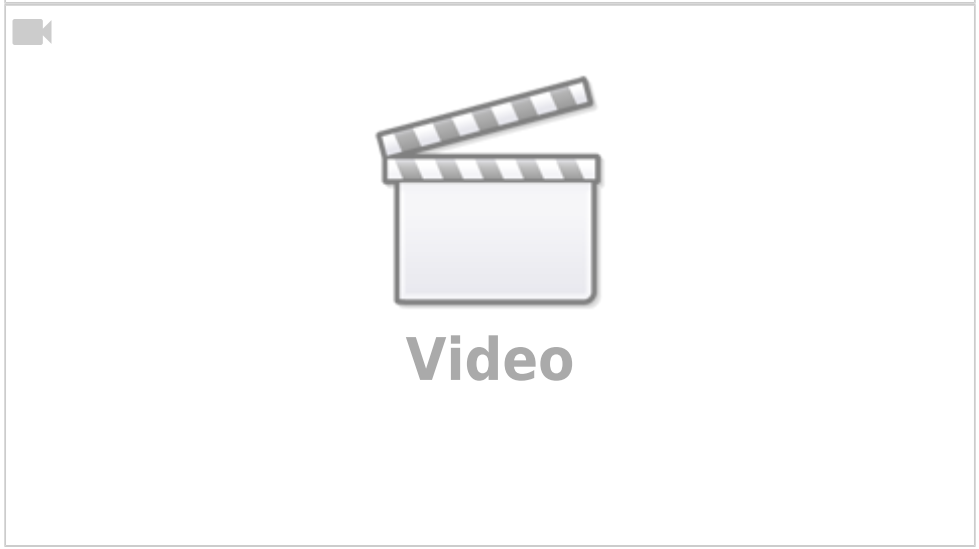
Die Blockgröße beträgt 64 Bits, das heißt ein 64-Bit-Block Klartext wird in einen 64-Bit-Block Chiffretext transformiert. Auch der Schlüssel, der diese Transformation kontrolliert, besitzt 64 Bits. Jedoch stehen dem Benutzer von diesen 64 Bits nur 56 Bits zur Verfügung; die übrigen 8 Bits (jeweils ein Bit aus jedem Byte) werden zum Paritäts-Check benötigt. Die effektive Schlüssellänge beträgt daher nur 56 Bits. Die Entschlüsselung wird mit dem gleichen Algorithmus durchgeführt, wobei die einzelnen Rundenschlüssel in umgekehrter Reihenfolge verwendet werden.

Auf den 64 Bit Block wird eine initiale Permutation angewandt. Danach wird der Block in zwei Teile aufgeteilt und jeder Teil in ein 32 Bit Register gespeichert, auf die das Prinzip eines Feistel-Netzwerkes angewandt wird.

Der DES-Algorithmus beschreibt zunächst nur, wie ein Datenblock mit 64 Bits verarbeitet wird. Zur Verarbeitung einer Nachricht beliebiger Länge lässt sich der DES wie auch jede andere Blockchiffre in verschiedenen Betriebsmodi verwenden. Für bestimmte Betriebsmodi, wie zum Beispiel ECB oder CBC, ist ein Auffüllen des Klartextes auf ein Vielfaches der vollen Blocklänge notwendig (Padding). Dies geschieht indem die Bitfolge 1000… angehängt wird.

Die genaue Spezifikation findet sich als FIPS 46-3 beim NIST.





AES (Advanced Encryption Standard) / Rijndael-Chiffre

AES steht für 'Advanced Encryption Standard', ist eine Blockchiffre und der Sieger-Algorithmus einer Ausschreibung in 2000 des NIST und gilt als Nachfolger von DES (Data Encryption Standard) von 1977. Der Algorithmus wurde von Joan Daemen und Vincent Rijmen entwickelt und die Chiffre wird deshalb auch Rijndael-Chiffre genannt. AES lässt einem die Wahl bei der Schlüssellänge von 128, 192 und 256 Bit. AES-192 und AES-256 sind in den USA für staatliche Dokumente mit höchster Geheimhaltungsstufe zugelassen.

AES benutzt eine Blocklänge von 16 Bytes, das heißt, dass ein Chiffrat 15 Zeichen länger werden kann als der ursprüngliche Klartext. Es empfiehlt sich, als Schlüssel den Hash eines Klartextpasswortes zzgl. eines (wenn gewünscht gehashten) Salts zu benutzen, z. B. SHA-256 für die 256-bit-Variante von AES. Dies ergibt eine gute Sicherheit..

AES hat sich mittlerweile als Standard durchgesetzt und neuere CPUs enthalten inzwischen spezielle Instruktionen, um die Verschlüsselung damit zu beschleunigen. Er wird u. a. bei der WLAN-Verschlüsselung WPA2, bei SSH, IPSec und in der IP-Telefonie benutzt.

Während Rijndael der Sieger-Algorithmus und zukünftiger Namensträger von AES wurde, waren die folgende vier weiteren Kandidaten in der engere Auswahl für AES gezogen worden, haben es letztendlich aber nicht geschafft: MARS, RC6, Serpent und Twofish. Weitere Kandidaten, die es nicht in die Endrunde schafften sind: CAST-256, CRYPTON, DEAL, DFC, E2, FROG, HPC, LOKI97, MAGENTA und SAFER+.

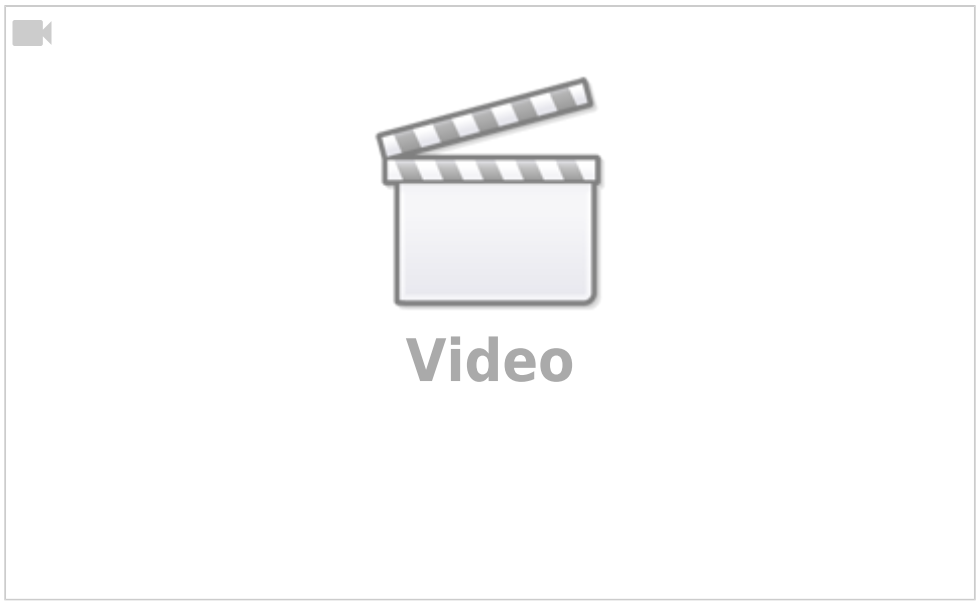
Beschreibung

Der in AES implementierte Algorithmus heißt Rijndael und ist ein als Substitutions-Permutations-Netzwerk entworfene Blockchiffre. Jeder Block wird zunächst in eine zweidimensionale Tabelle mit vier Zeilen geschrieben, deren Zellen ein Byte groß sind. Die Anzahl der Spalten variiert je nach Blockgröße von 4 (128 Bits) bis 8 (256 Bits). Jeder Block wird nun nacheinander bestimmten Transformationen unterzogen. Aber anstatt jeden Block einmal mit dem Schlüssel zu verschlüsseln, wendet Rijndael verschiedene Teile des erweiterten Originalschlüssels nacheinander auf den Klartext-Block an. Die Anzahl der Runden variiert und ist von Schlüssellänge und Blockgröße abhängig (bei AES also nur von der Schlüssellänge).

Eine S-Box (Substitutionsbox) mit 256 Bytes dient als Basis für eine monoalphabetische Verschlüsselung. Sie gibt an, wie in jeder Runde jedes Byte eines Blocks durch einen anderen Wert zu ersetzen ist. Typischerweise wird die S-Box in Blockchiffren eingesetzt, um die Beziehung zwischen Klar- und Geheimtext zu verwischen (in der kryptologischen Fachsprache Konfusion genannt). Die S-Box des AES setzt auch teilweise das Shannon’sche Prinzip der Diffusion um. Die Konstruktion der S-Box unterliegt Designkriterien, die die Anfälligkeit für die Methoden der linearen und der differentiellen Kryptoanalyse sowie für algebraische Attacken minimieren sollen.

Die genaue Spezifikation findet sich beim NIST.

Hier der Link zum Ausprobieren: Rijndael Animation



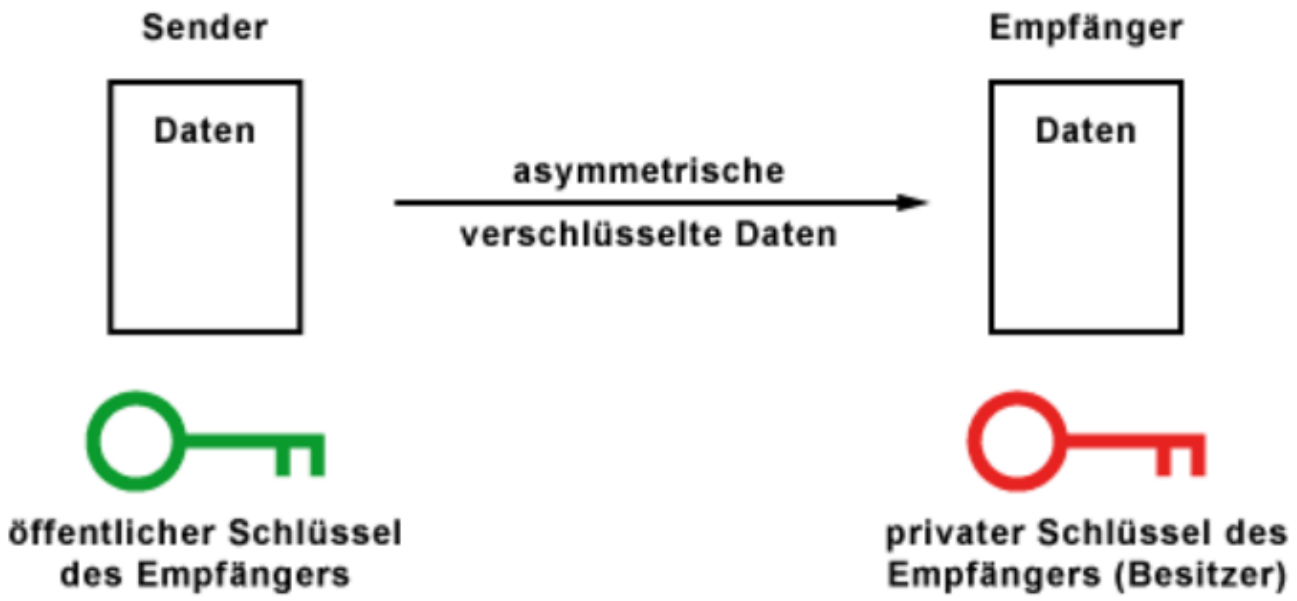
Asymmetrische Kryptografie (Verschlüsselung)

In der asymmetrischen Kryptografie arbeitet man nicht mit einem einzigen Schlüssel, sondern mit einem **Schlüsselpaar**. Bestehend aus einem **öffentlichen** und einem privaten Schlüssel. Man bezeichnet diese Verfahren als asymmetrische Verfahren oder **Public-Key-Verfahren**. Üblich sind auch die Bezeichnungen Public-Key-Kryptografie und Public-Key-Verschlüsselung.

Ein fundamentales Problem der Kryptografie ist, dass sich die Kommunikationspartner auf einen gemeinsamen Schlüssel verständigen müssen. Man bezeichnet das als **Schlüsselaustauschproblem**.

Während ein manueller Schlüsselaustausch durch ein persönliches Treffen oder per Telefon bei einer handvoll Kommunikationspartner sicherlich kein Problem wäre. Wird es bei vielen Schlüsseln oder vielen Kommunikationspartnern schnell unübersichtlich und aufwendig. Hier kommt das Thema **Schlüsselverwaltung und -verteilung** zum Tragen. Alternativ bestünde die Möglichkeit einen Authentifizierungsserver einzusetzen. Beispielsweise Kerberos. Alternativ bietet sich die asymmetrische Kryptografie an.

Prinzip der asymmetrischen Kryptografie (Public-Key-Verfahren)

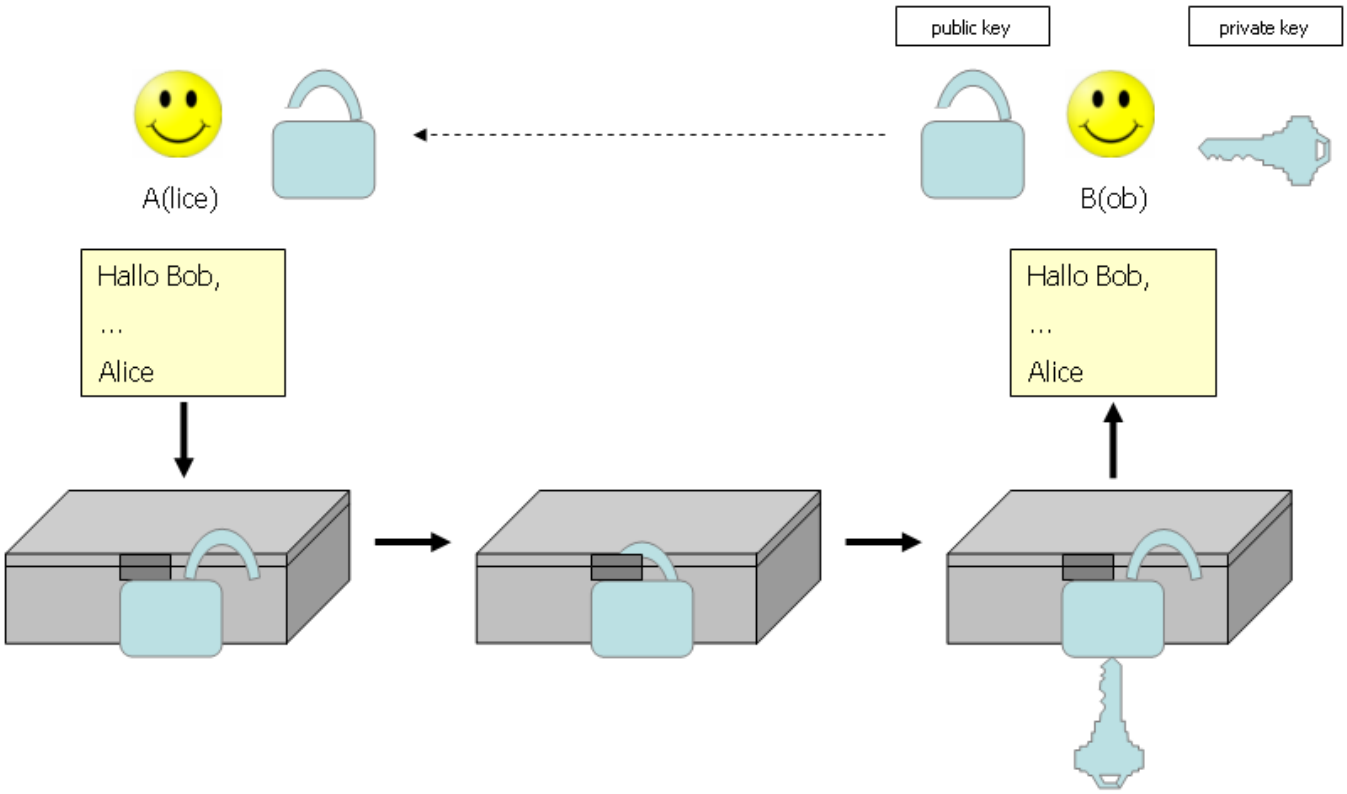


Asymmetrische Verschlüsselungsverfahren arbeiten mit Schlüsselpaaren. Ein Schlüssel ist der **öffentliche Schlüssel (Public Key)**, der andere ist der **private Schlüssel (Private Key)**. Dieses Schlüsselpaar hängt über einen **mathematischen Algorithmus** eng zusammen. **Daten**, die mit dem **öffentlichen Schlüssel verschlüsselt** werden, können nur mit dem **privaten Schlüssel entschlüsselt** werden. Deshalb muss der private Schlüssel vom Besitzer des Schlüsselpaares geheim gehalten werden.

Der konkrete Anwendungsfall sieht so aus: Will der Sender Daten verschlüsselt an den Empfänger senden, benötigt er den öffentlichen Schlüssel des Empfängers. Mit dem öffentlichen Schlüssel können die Daten verschlüsselt, aber nicht mehr entschlüsselt werden (Einwegfunktion). Nur noch der Besitzer des privaten Schlüssels, also der richtige Empfänger kann die Daten entschlüsseln. Wichtig bei diesem Verfahren ist, dass der private Schlüssel vom Schlüsselbesitzer absolut geheim gehalten wird. Kommt eine fremde Person an den privaten Schlüssel muss sich der Schlüsselbesitzer ein neues Schlüsselpaar besorgen.

Das Problem bei der asymmetrischen Kryptografie ist die Verteilung der öffentlichen Schlüssel. Typischerweise erfolgt die Übergabe des öffentlichen Schlüssels beim Erstkontakt. Doch hierbei stellt sich die Frage, ob dieser Schlüssel tatsächlich der echte Schlüssel des Kommunikationspartner ist.

Hinweis: Asymmetrische Verfahren benötigen viel mehr Rechenleistung als symmetrische Verfahren. Wenn man RSA und AES miteinander vergleicht, dann ist RSA ungefähr um den Faktor 1.000 langsamer als AES.



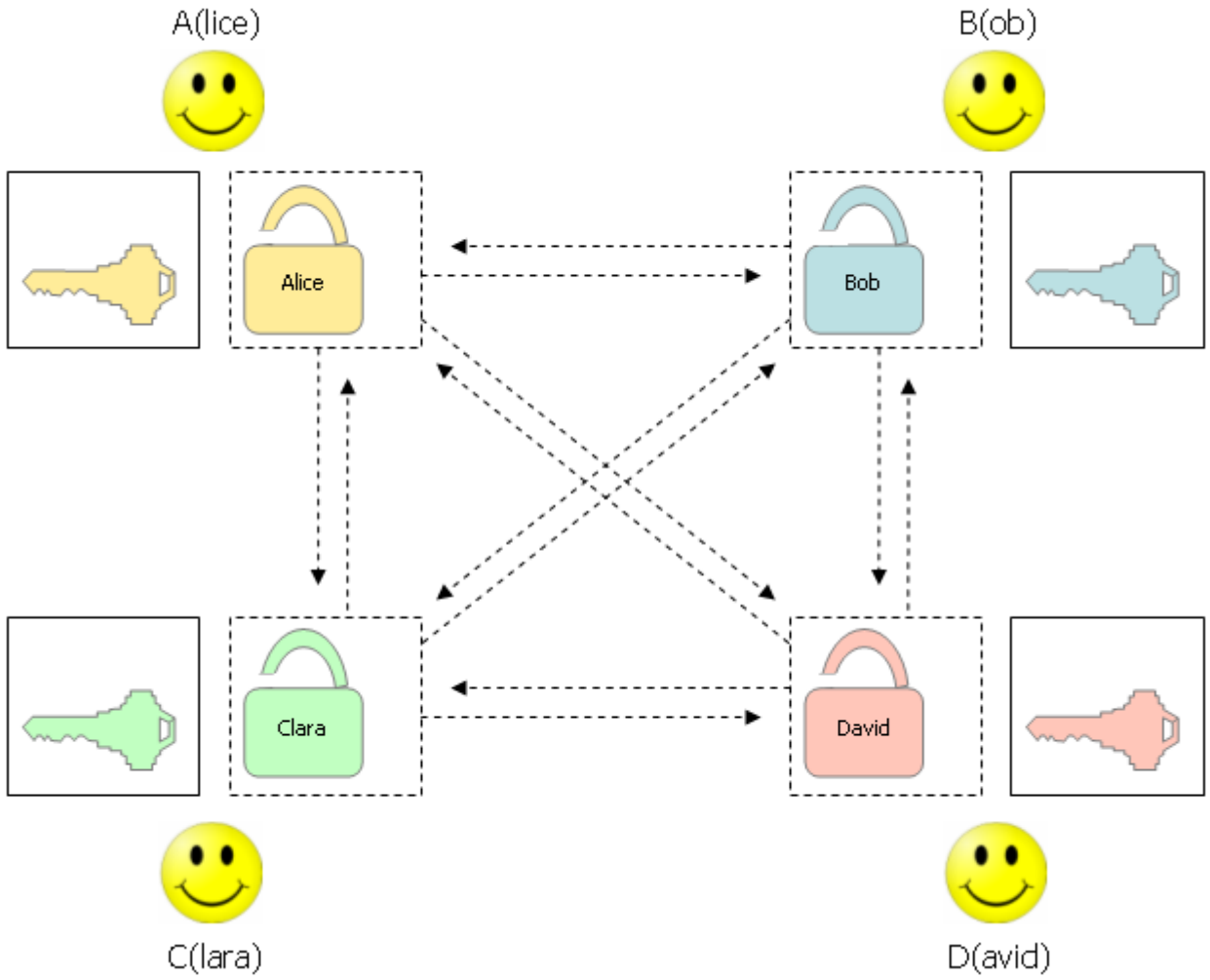
Vereinfachtes Beispiel

Ich verteile Sparschweine, zu deren Schloss nur ich den Schlüssel habe. Wenn mir jemand die Nachricht „135“ senden möchte, nimmt er eines dieser Sparschweine, steckt einen Zettel mit „135“ hinein und schickt es mir per Post. Ich öffne mit meinem Schlüssel das Sparschwein und kann den Zettel lesen.

Ein Schlüsselpaar für jede Kommunikationsteilnehmer/in

Wenn mehrere Personen verschlüsselte Nachrichten austauschen wollen, dann reicht es, wenn jede

Person ein Schlüsselpaar aus öffentlichem und privatem Schlüssel besitzt.



Bei 4 Personen A(lice), B(ob), C(lara) und D(avid), die alle miteinander kommunizieren wollen, werden somit 4 Schlüsselpaare benötigt. Bei n Personen benötigt man entsprechend genau n Schlüsselpaare. Es werden also viel weniger Schlüssel benötigt als bei symmetrischen Chiffriersystemen.

Einwegfunktion und Falltürfunktion

Damit so ein Verfahren funktioniert, muss Folgendes gelten:

- Mithilfe der öffentlich verfügbaren Informationen (insbesondere dem öffentlichen Schlüssel) muss es in sinnvoller Zeit möglich sein, den Klartext zum Geheimtext zu verschlüsseln. Außerdem muss es mit dem privaten Schlüssel in sinnvoller Zeit möglich sein, den Geheimtext zu entschlüsseln. Ansonsten könnte man das Verfahren nie anwenden.
- Mithilfe der öffentlich verfügbaren Informationen darf es – ohne den privaten Schlüssel – nicht in sinnvoller Zeit möglich sein, den Geheimtext zu knacken, also den Klartext zu rekonstruieren. Daraus folgt, dass auch der private Schlüssel nicht mithilfe der öffentlichen Informationen in sinnvoller Zeit berechnet werden kann.

Asymmetrische Chiffriersysteme nutzen Funktionen mit den oben dargestellten Eigenschaften – sogenannte Einwegfunktionen. In eine Richtung lassen sie sich in sinnvoller Zeit berechnen, aber die Umkehrung ist – ohne den privaten Schlüssel als Hilfsmittel – kaum möglich.

Auf den ersten Blick klingt es verwunderlich, dass hier immer von „in sinnvoller Zeit“ die Rede ist. Das liegt daran, dass man ja z.B. alle möglichen privaten Schlüssel durchprobieren könnte. Es ist also nicht möglich, das Knacken ganz auszuschließen. Aber, wenn es mit aller Rechenleistung der Welt viele Hundert Millionen Jahre dauern würde, gilt das als ausreichend sicher.

Bei der asymmetrischen Verschlüsselung geht es darum, eine **Funktion** zu wählen, die **sehr einfach zu rechnen ist, aber deren Umkehrung dagegen sehr aufwendig**. Realisiert wird das mit **Modulo-Rechenarten**.

Einige davon sind tatsächlich sehr einfach zu rechnen, während die Umkehrung sehr aufwendig ist. Sie entsprechen also einer **Einwegfunktion**.

Es gibt allerdings auch Funktionen, bei denen sich mit einer **zusätzlichen Information die Umkehrung abkürzen** lässt. In so einem Fall spricht man von einer **Falltürfunktion**.

Mathematische Grundlagen

Modulo

Bei der Modulo Operation wird der Rest einer Division berechnet. Der Rückschluss vom Ergebnis (Rest) auf die Ausgangszahl ist nicht möglich.

Beispiel:

```
17 % 5 = 2
6 % 2 = 0
18 % 6 = 0
```

Potenzieren

Potenzieren im Bereich natürlicher, ganzer, reeller oder sogar komplexer Zahlen ist eine grundlegende Operation der Mathematik:

$g^e = y$

Beispiel:

```
10^2 = 100
```

Logarithmieren

Die Umkehrfunktion zum Potenzieren in Bezug auf das Auffinden der Exponenten ist das Logarithmieren.

$\log_g y = e$

Beispiel: $\log_{10} 100 = 2$

Hierbei ist es nicht unüblich, dass dieser Exponent außerhalb des Definitionsbereichs liegt (z.B. $\log_{10} 8 \approx 0, 9$).

Nichtsdestotrotz lässt sich dieser Exponent mit Hilfe von Computern in sehr schneller Zeit (näherungsweise) errechnen. Beispielhaft wäre da die Berechnung über die Potenzreihenentwicklung.

Modulo Addition/Subtraktion

Ist bei der Addition das Ergebnis größer oder gleich n, dann zieht man n davon ab. Ist analog dazu, das Ergebnis bei der Subtraktion kleiner null, dann zählt man n dazu.

Beispiel:

```
(3+5) % 7 = 1
(2+2) % 13 = 4
(3-6) % 9 = 6
(3+6) % 9 = 0
```

Modulo Multiplikation

Beispiel:

```
(4*5) % 7 = 20 -7 -7 = 6
```

Modulo Division

Fragestellung: gibt es zu jeder Zahl a eine Zahl b mit der Eigenschaft a*b=a (mod n)

Gibt es eine solche Zahl, dann nennt man sie das zu a inverse Element und schreibt dafür a-1.

Es gibt auch eine Modulo-Division, weil $b*a^{-1}$ (mod n) gleich ist wie b/a (mod n).

Potenzieren mit Modulo

Beispiel:

```
3^4 % 7 = 3*3*3*3 % 7 = 4
```

In der Kryptografie spielt vor allem die Umkehrung dieser Rechenart eine Rolle!

Diskreter Logarithmus (Logarithmieren mit Modulo)

Das Problem des diskreten Logarithmus beschreibt ein mathematisches Problem. Eine der beiden

Umkehrungen der Potenzierung mit Modulo ist der Modulo-Logarithmus, der auch als diskreter Logarithmus bezeichnet wird. Sind g, x und p gegeben, dann ist der diskrete Logarithmus die Zahl x, für die gilt:

$g^x \pmod p = S$

g ... Generator p ... Primzahl

$\{g,p,S\} \in \mathbb{N}$

$\{x\} \in \mathbb{N}_0$

x ⇒ geheimer Schlüssel
S ⇒ öffentlicher Schlüssel

Ein Angreifer, der diese Gleichung nach dem geheimen Schlüssel nach x auflösen könnte, wäre in der Lage die geheime Nachricht zu entschlüsseln.

Bei kleinen Zahlen kann man durch systematisches Ausprobieren dieses Problem lösen.

Beispiel

g=8, p=29 und S=21 ⇒ $8^x \bmod 29 = 21$

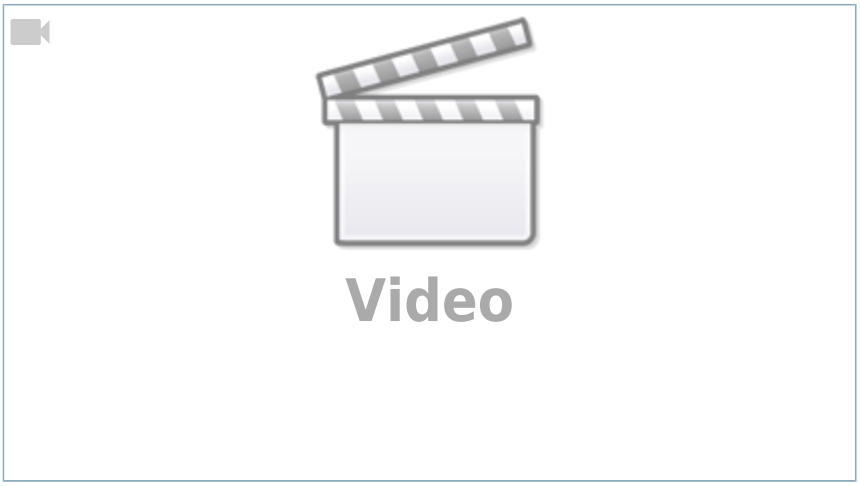
n	$8^n \bmod 29$
0	1
1	8
2	6
3	19
4	7
5	27
6	13
...	...
15	21
...	...
28	1
29	8

Der diskrete Logarithmus von b zur Basis x, kurz $\log_x b$, ist die kleinste natürliche Zahl g:

$g=\log_x b \pmod n$

Die Frage, wann zu gegebenen Zahlen g, S und n der diskreter Logarithmus existiert, ist kein triviales Problem, in der Kryptografie hat man jedoch nur mit solchen Problemstellungen zu tun, in denen der diskrete Logarithmus existiert.

Sind die Zahlen groß genug, so schaffen es auch nicht die leistungsfähigsten Computer durch systematisches Ausprobieren zu einem Ergebnis zu kommen. Bis jetzt ist kein Algorithmus bekannt, der dieses Problem für große Zahlen lösen kann.



Der **diskrete Logarithmus** fällt hier als **Einwegfunktion** besonders auf, weil man diesen sehr leicht berechnen kann. Umgekehrt ist es schlichtweg nicht möglich eine große Zahl in praktikabler Zeit zurückzurechnen. Man bezeichnet das als Diskreter-Logarithmus-Problem. Viele asymmetrische Verfahren basieren darauf. Allerdings bedeutet das nicht, dass nicht doch irgendwann ein Weg gefunden wird, den diskreten Logarithmus zu lösen.

Faktorisierungsproblem

Eine weitere Einwegfunktion ist das Multiplizieren von Primzahlen. Während die Multiplikation für einen Computer kein Problem darstellt, ist der umgekehrte Weg, beim dem das Primzahlprodukt in seine Faktoren zerlegt werden soll, nicht in akzeptabler Zeit machbar. Man spricht von Faktorisierung und in dem Zusammenhang vom Faktorisierungsproblem.

Beispiel:

n = p * q

p=17
q=19

n = p*q = 17*19 = 323

Wenn man 17 x 19 berechnet (beides Primzahlen), dann kommt 323 heraus. Und jetzt soll man die beiden unbekannten Faktoren (17 und 19) daraus zurückberechnen. Es gibt im Prinzip nur einen Weg. Man muss alle Möglichkeiten durchprobieren. Bei hinreichend großen Primzahlen dauert das ewig. Damit ist das Faktorisierungsproblem gemeint.

n	q	n/q=p
323	2	161,5
323	3	107,6666..
323	5	64,6
323	7	46,142...

323	9	35,88..
323	11	29,363...
323	13	24,846...
323	**17**	**19**

Alle gängigen asymmetrische Verfahren basieren auf komplexen mathematischen Berechnungen, die gemeinsam haben, dass es für sie noch keine Vereinfachung gibt. Schlüssel, Klartext und Geheimtext stellen große Zahlen bzw. Zahlenpaare dar. Die Verfahren sind aber nur so lange sicher sind, bis jemand eine Vereinfachung gefunden hat.

Weil es nur begrenzt geeignete mathematische Berechnungen mit Einwegfunktion gibt, lassen sich nicht beliebig viele asymmetrische Verfahren entwickeln.

Diffie-Hellman-Schlüsseltausch

Zurück zum Schlüsseltauschproblem: Alice und Bob können Einweg- und Falltürfunktionen zur Lösung des Schlüsseltauschproblems verwenden.

Ein Verfahren, das den diskreten Logarithmus zur Lösung des Schlüsseltauschproblems verwendet, wurde von den Kryptografen Whitfield Diffie und Martin Hellman erfunden.

Der Algorithmus mit Farben

Die Mathematik hinter dem Algorithmus ist leider nicht ganz einfach zu verstehen. Dafür gibt es aber eine schöne Analogie mit Farben, die man zum Verständnis nutzen kann.

Der Ablauf:

1. Alice und Bob einigen sich auf eine gemeinsame (öffentliche) Farbe.
2. Jeder wählt sich zudem eine geheime weitere Farbe.
3. Alice und Bob mischen sich aus ihrer geheimen und der öffentlichen Farbe eine weitere Farbe.
4. Sie schicken jeweils die neu gemischte Farbe zu ihrem Kommunikationspartner.
5. Am Ende mischt jeder seine geheime Farbe in die zuvor ausgetauschten Mischfarbe.

Die gemeinsame Farbe (der gemeinsame Cocktail) am Ende des Prozesses besteht also aus drei gemischten Farben. Und Eve in der Mitte? Sie hat nur die öffentliche Farbe und zwei gemischte Farben. Und wie du vielleicht aus dem Kunstunterricht weißt, ist es ziemlich schwierig, aus einer Mischfarbe zu erkennen, welche Ausgangsfarben genau darin stecken.

Exkurs Diffie-Hellmann von INF-Schule

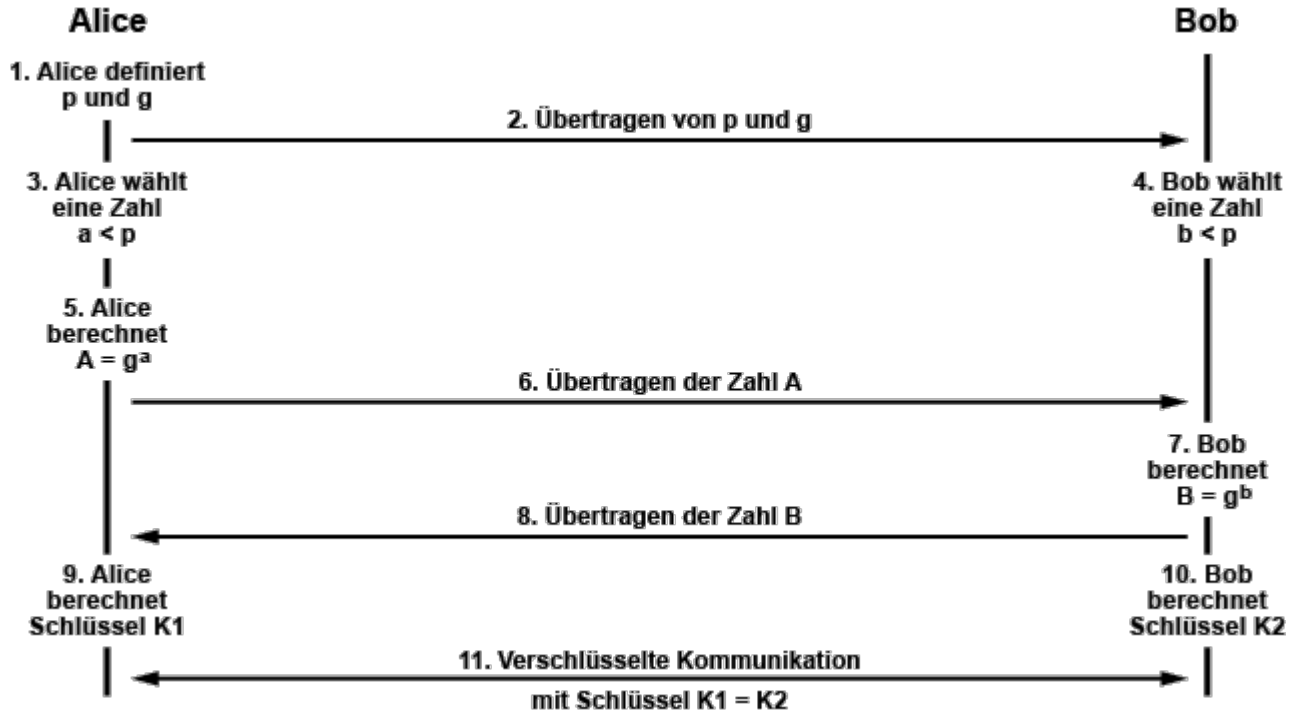
Diffie-Hellmann-Schlüsseltausch Algorithmus

Der Algorithmus mit Zahlen

In der folgenden Beschreibung ist von Alice und Bob die Rede. Beide stehen beispielhaft für zwei Kommunikationspartner, die ihre Kommunikation verschlüsseln wollen und den dazu notwendigen

geheimen Sitzungsschlüssel zum Ver- und Entschlüsseln vorab austauschen müssen. Um den Sitzungsschlüssel vor einem Angreifer zu schützen, der eventuell die Kommunikation abhört oder aufzeichnet, in der Hoffnung den Sitzungsschlüssel abgreifen zu können, vereinbaren sie den Schlüsselaustausch nach Diffie-Hellman-Merkle.

Das mathematische Verfahren beruht auf dem sog. „modularen Potenzieren“. Auch wenn der mathematische Hintergrund nicht so einfach zu verstehen ist, kannst du das Verfahren aber sicherlich einmal hier nachvollziehen. Die wesentlichen Schritte sind (ganz analog zu den Farben oben) die Folgenden.



1. Alice definiert p und g

Zuerst müssen sich Alice und Bob auf eine große Primzahl p und eine natürliche Zahl g, die ein Generator aus Gruppe Z(p) sein sollte, einigen. Die Zahl g kann aber auch einen Wert kleiner p annehmen. Weil Alice die Kommunikation zu Bob aufbaut, legt typischerweise Alice die Zahlen p und g fest. Diese Vorgehensweise kann in der Praxis auch anders erfolgen. Für dieses Beispiel wählt Alice **p = 11** und **g = 7**.

2. Alice schickt p = 11 und g = 7 zu Bob

Beide Werte dürfen bekannt sein und können deshalb über einen unsicheren Kanal übertragen werden.

3. Alice wählt eine Zahl a < p

Alice erzeugt nun zusätzlich eine Zufallszahl a, die kleiner als die gewählte Primzahl p sein muss (1 … p - 1). Für dieses Beispiel wählen wir a = 3.

4. Bob wählt eine Zahl b < p

Bob erzeugt nun zusätzlich eine Zufallszahl b, die kleiner als die gewählte Primzahl p sein muss (1 ... p - 1). Für dieses Beispiel wählen wir b = 6.

5. Alice berechnet A

$$A = g^a \bmod p$$
$$A = 7^3 \bmod 11 = 2$$

6. Alice schickt die Zahl A = 2 zu Bob

Übertragung der Zahl A = 2 über den unsicheren Kanal zu Bob.

7. Bob berechnet B

$$B = g^a \bmod p$$
$$B = 7^6 \bmod 11 = 4$$

8. Bob schickt die Zahl B = 4 zu Bob

Übertragung der Zahl B = 4 über den unsicheren Kanal zu Alice.

9. Alice berechnet Schlüssel K1

$$K1 = B^a \bmod p$$
$$K1 = 4^3 \bmod 11 = 9$$

10. Bob berechnet Schlüssel K2

$$K2 = A^b \bmod p$$
$$K2 = 2^6 \bmod 11 = 9$$

11. Verschlüsselte Kommunikation mit Schlüssel K1 = K2

Beide kommen auf das gleiche Ergebnis und haben so einen gemeinsamen geheimen Schlüssel. Dieser Schlüssel kann zum Beispiel in einem symmetrischen Verfahren als temporärer Sitzungsschlüssel genutzt werden.

$$K1 = K2$$

Der Angreifer kennt nur p und g. Außerdem A und B, da beide Werte übertragen werden. Allerdings kann er den Schlüssel K nur berechnen, wenn er a und b hat. Da diese Werte nicht übertragen werden, muss der Angreifer sich den Schlüssel anders berechnen, was bei einer ausreichend großen Primzahl fast unmöglich ist. Dieses Verfahren beruht darauf, dass es mit wenig Rechenleistung möglich ist, die Potenz g^x mod p zu errechnen. Aber der umgekehrte Weg von g^x auf x zu schließen ist sehr schwierig (diskreter Logarithmus).

RSA - Verfahren

RSA ist das ein asymmetrisches kryptografisches Verfahren bzw. ein Public-Key-Verfahren von den Kryptografen Ron Rivest, Adi Shamir und Leonard Adleman aus dem Jahr 1977. Kein anderes asymmetrisches Verfahren ist so vielseitig einsetzbar, so gut erforscht und so einfach zu implementieren, wie RSA. An RSA kommt man im Zusammenhang mit asymmetrischen Verfahren einfach nicht vorbei. Im Vergleich zum Diffie-Hellman-Schlüsselaustausch eignet sich RSA auch für die Verschlüsselung und als Signaturverfahren. Der RSA-Algorithmus basiert auf dem Faktorisierungsproblem.

Das RSA-Verfahren ist ein modernes asymmetrisches kryptografisches Verfahren, das zum Verschlüsseln und zum digitalen Signieren verwendet wird. Um es zu durchschauen, musst du dich mit zahlentheoretischen Überlegungen und Zusammenhängen auseinandersetzen.

RSA-Verfahren Step-by-Step

Hybride Chiffriersysteme

Fazit von symmetrische Verschlüsselungsverfahren

Die Algorithmen, die bei modernen symmetrischen Chiffriersystemen benutzt werden, sind sehr effizient. Man kann mit symmetrischen Chiffriersystemen daher recht schnell einen gegebenen Text verschlüsseln.

Schwierig ist bei symmetrischen Chiffriersystemen in der Regel der Schlüsselaustausch zwischen Sender/in und Empfänger/in. Da beide denselben Schlüssel benutzen, muss er sicher zwischen den Kommunikationspartner/innen ausgetauscht werden.

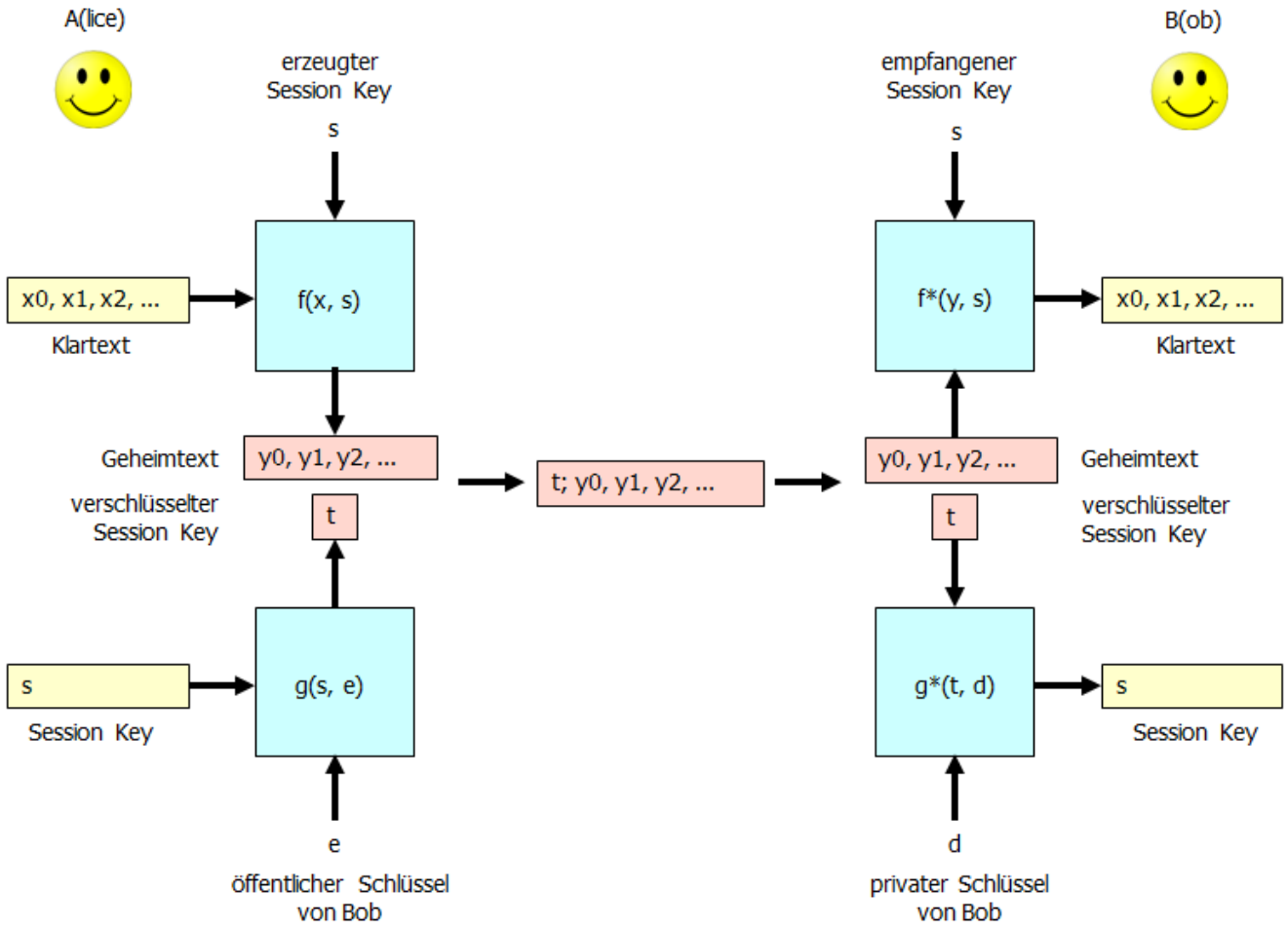
Fazit von asymmetrische Verschlüsselungsverfahren

Die Algorithmen, die bei modernen asymmetrischen Chiffriersystemen benutzt werden, sind recht aufwendig. Bei langen Texten dauert es daher eine Weile, bis der gesamte Text verschlüsselt ist.

Da bei asymmetrischen Chiffriersystemen ein Schlüsselpaar erzeugt wird und einer der beiden Schlüssel veröffentlicht wird, ist allerdings kein sicherer Schlüsselaustausch zwischen Sender:in und Empfänger:in erforderlich.

Kombination aus symmetrischen und asymmetrischen Verschlüsselungsverfahren

Die Stärken von symmetrischen und asymmetrischen Chiffriersystemen lassen sich nutzen, wenn man beide Systeme geeignet kombiniert. Die folgende Abbildung zeigt die Struktur eines kombinierten (man sagt auch hybriden) Chiffriersystems:



Wenn Alice Bob eine Nachricht senden möchte, erzeugt Alice in einem ersten Schritt einen sogenannten Session Key.

Dieser Session Key wird als Schlüssel zum Verschlüsseln der Nachricht benutzt. Dabei kommt ein schnelles symmetrisches Verfahren zum Einsatz.

Als nächstes wird der Session Key selbst verschlüsselt. Alice benutzt hierzu den öffentlichen Schlüssel von Bob bei einem asymmetrischen Chiffrierverfahren.

Die verschlüsselte Nachricht und der verschlüsselte Session Key werden jetzt an Bob geschickt.

Bob benutzt seinen privaten Schlüssel, um den Session Key zu rekonstruieren.

Anschließend kann Bob den Session Key - zusammen mit dem von Alice benutzten symmetrische Chiffrierverfahren - benutzen, um den Geheimtext zu entschlüsseln.

Digitale Signatur

Die digitale bzw. elektronische Signatur ist eine schlüsselabhängige Prüfsumme, die von einer Nachricht oder einem Dokument in Kombination mit einem Schlüssel erzeugt wird. Wird die Signatur an eine Nachricht oder ein Dokument angehängt, dann gilt das als unterschrieben. Für digitale Nachrichten und Dokumente werden digitale Signaturen verwendet, um ihre Echtheit glaubhaft und prüfbar zu machen. Die Echtheit der Signatur kann elektronisch geprüft werden.

Digitale Signaturen sind in der Datenübertragung deshalb notwendig, weil sich der Absender von Nachrichten und Dokumenten fälschen lässt. Beispielsweise ist es ganz einfach den Absender einer E-Mail zu fälschen. Das heißt, es ist möglich, dass sich jemand als eine andere Person ausgibt. Auch im wirklichen Leben kann man eine beliebige Absender-Adresse auf einen Brief schreiben. Um die Glaubwürdigkeit des Briefs zu unterstreichen setzen wir an das Briefende unsere Unterschrift. Genauso wird es mit der digitalen Signatur gemacht.

Digitale Signaturen gewährleisten:

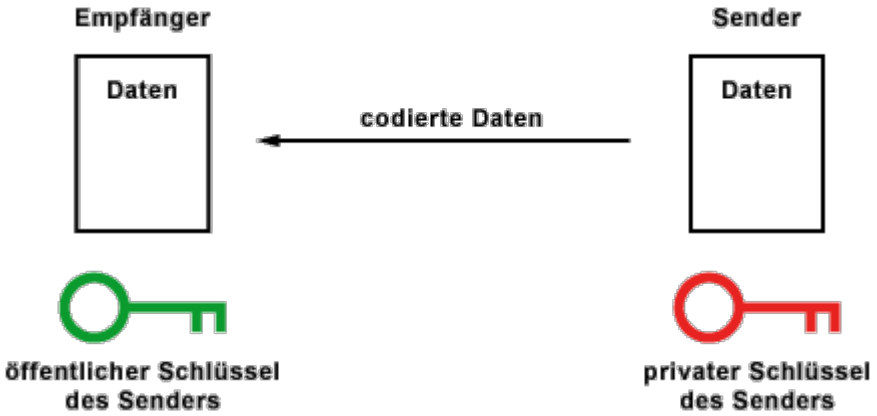
- **Integrität**: Die Nachricht, die man erhält, ist von keiner dritten Person manipuliert worden.
- **Authentizität**: Die Nachricht, die man erhält, stammt wirklich von der Person, die als Absender angegeben ist.
- **Verbindlichkeit**: Der Urheber kann nachträglich nicht bestreiten, die Nachricht verfasst zu haben.

dafür müssen folgende Anforderungen gelten:

- Die digitale Signatur darf nicht (unbemerkt) fälschbar sein.
- Die Echtheit der digitalen Signatur muss überprüfbar sein.
- Die digitale Signatur darf nicht von einem Dokument auf ein anderes (unbemerkt) übertragbar sein.
- Das signierte Dokument darf nicht (unbemerkt) veränderbar sein.

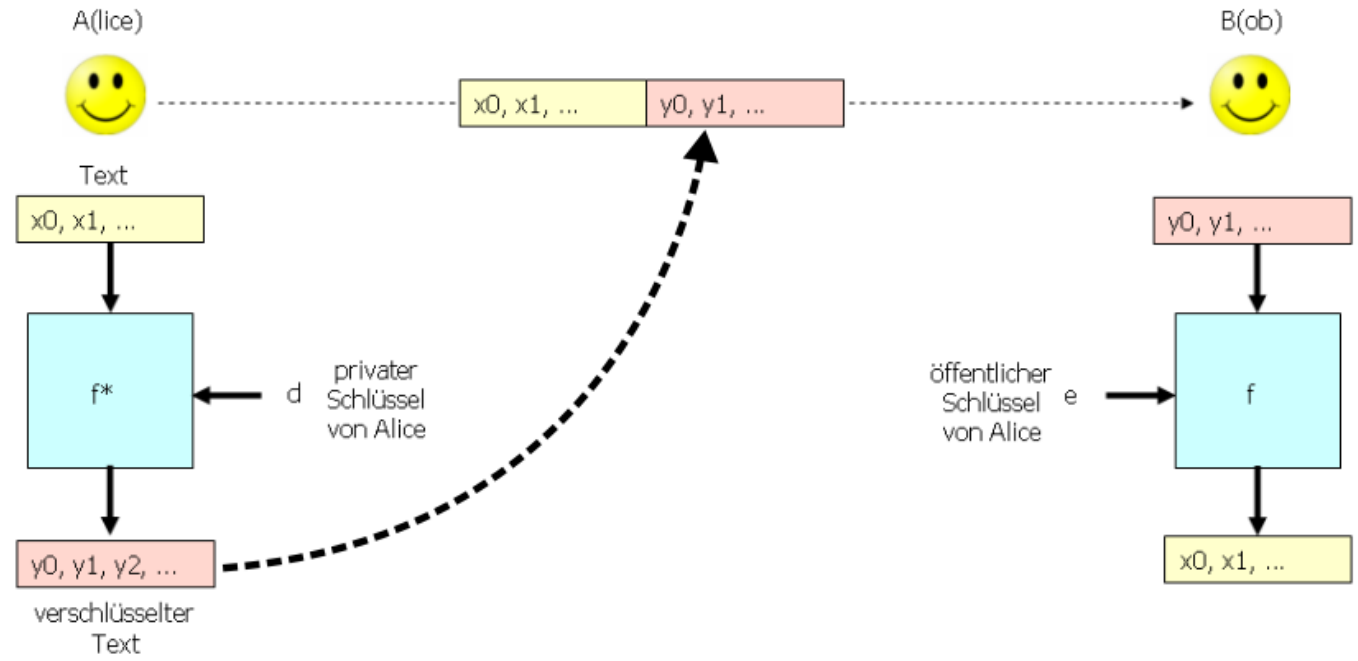
Eine Infrastruktur mit digitalen Signaturen macht nur dann Sinn, wenn die Signaturen in übertragenen Nachrichten und Dokumenten ständig geprüft werden. Nur dann kann erkannt werden, wenn eine Signatur, eine Nachricht oder ein Dokument gefälscht wurde. Wenn die Prüfung nicht erfolgt, dann bleiben Manipulationen „unbemerkt“, was nicht den Anforderungen der digitalen Signatur entspricht. Wird muss ein signiertes Dokument geändert werden, dann muss es erneut signiert werden, weil die alte Signatur nicht mehr zum Dokument passt.

Funktionsweise der digitalen Signatur



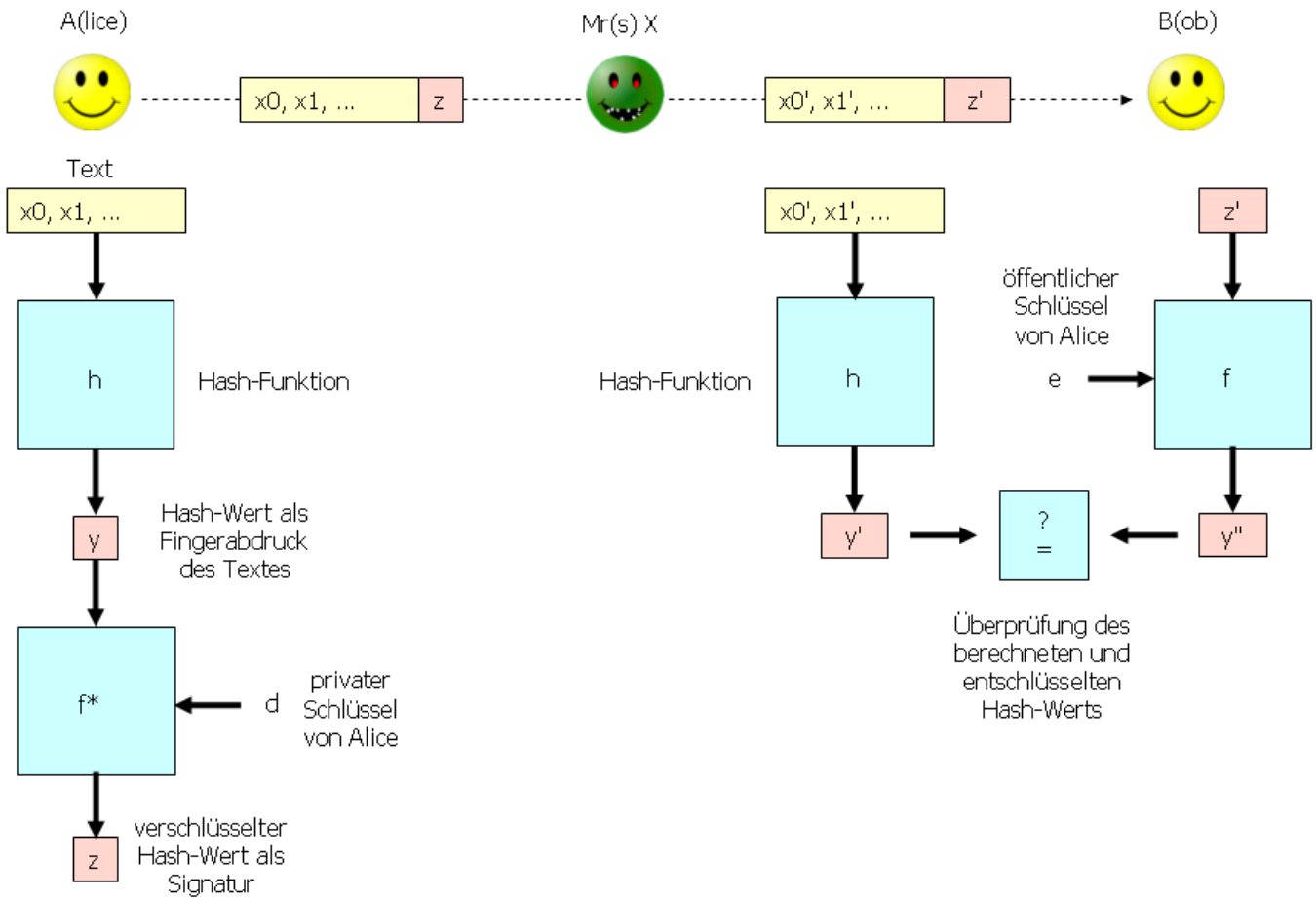
Die digitale Signatur basiert auf der asymmetrischen Kryptografie, wobei das verwendete **asymmetrische Verfahren umgekehrt** wird. Bei der asymmetrischen Verschlüsselung dient der öffentliche Schlüssel zum Verschlüsseln und der private Schlüssel zum Entschlüsseln. Bei der digitalen Signatur werden die Daten mit Kennzeichen versehen, die durch den privaten Schlüssel hinzugefügt werden. Mit dem öffentlichen Schlüssel kann man feststellen, ob die Daten von demjenigen stammen, der mit seinem privaten Schlüssel signiert hat und ob die Daten unverändert sind. Die Tatsache, dass der private Schlüssel durch seinen Besitzer geheim gehalten wird, erlaubt die Annahme, dass Daten, die mit dem privaten Schlüssel codiert sind, tatsächlich vom Schlüsselbesitzer stammen.

Um Nachrichten und Dokumente zu signieren muss dessen Ersteller die Nachricht mit seinem **privaten Schlüssel „entschlüsseln“**. Der dabei **entstandene „Klartext“ ist die digitale Signatur**. Sie wird an das Dokument angehängt. Die Signatur kann von jedem anderen mit dem **öffentlichen Schlüssel des Erstellers „verschlüsselt“ werden**. Dabei entsteht die ursprüngliche Nachricht, die mit der unverschlüsselten Nachricht verglichen werden kann. Sind beide gleich, ist das Dokument unverändert und korrekt signiert.



Blockweise erstellte Signaturen haben zusammen mindestens die gleiche Länge, wie die Nachricht oder das Dokument selbst. Das ist recht unpraktisch. Das Signieren kann ein erheblicher Aufwand bezüglich Aufwand und Rechenleistung sein. Ebenso die Prüfung der Signatur. Deshalb wird **nie die ganze Nachricht signiert**. Statt dessen wird **aus der Nachricht zuerst eine Prüfsumme gebildet**, die viel kürzer sein kann als die Nachricht selbst. Und erst dann wird das Signaturverfahren auf diese Prüfsumme angewendet. Das Verfahren, mit dem die Prüfsumme gebildet wird, ist eine

kryptografische Hash-Funktion. Das Ergebnis der kryptografischen Hash-Funktion ist der Hash-Wert oder auch nur Hash genannt.



Ein Signiersystem benutzt in der Regel eine Hash-Funktion, um eine Art Fingerabdruck des zu signierenden Textes zu erstellen. Mit Hilfe eines Schlüsselpaares bestehend aus einem öffentlichen und einem privaten Schlüssel wird aus dem Fingerabdruck dann die digitale Signatur erstellt. Die Vorgehensweise soll anhand des in der Abbildung gezeigten Szenarios beschrieben werden.

Alice will eine signierte Nachricht an Bob senden.

In einem ersten Schritt erzeugt sie mit Hilfe einer Hash-Funktion einen Fingerabdruck des zu versendenden Textes. Bei dem Fingerabdruck handelt es sich um ein Bitmuster, das dem Text zugeordnet wird.

Diesen Fingerabdruck verschlüsselt Alice mit ihrem privaten Schlüssel. Das Ergebnis ist ein Bitmuster, das die digitale Signatur zum vorgegebenen Text bildet.

Alice sendet jetzt den Text mit der digitalen Signatur an Bob.

Wie überprüft Bob die Integrität der erhaltenen Nachricht?

Bob benutzt dieselbe Hash-Funktion wie Alice, um einen Fingerabdruck zum übermittelten Text zu erzeugen.

Bob ist im Besitz des öffentlichen Schlüssels von Alice und benutzt ihn, um die übermittelte Signatur zu entschlüsseln.

Wenn die Nachricht nicht verändert wurde, dann erhält Bob durch die Entschlüsselung der Signatur den von Alice erzeugten Fingerabdruck zum versendeten Text. Dieser ist dann identisch mit dem von Bob bestimmten Fingerabdruck zum empfangenen Text.

Wenn die Nachricht in Teilen verändert wurde, dann müsste das Bob beim Vergleich der Fingerabdrücke auffallen. Wenn Mr(s) X. z.B. den Text abändert, dann ändert sich auch der Fingerabdruck zum Text. Mr(s) X. kann zwar einen Fingerabdruck zum veränderten Text erzeugen, Mr(s) X. kann ihn aber nicht passend verschlüsseln, da Mr(s) X. keinen Zugang zum privaten Schlüssel von Alice hat (davon gehen wir hier natürlich aus). Mr(s) X. ist demnach nicht in der Lage, ein stimmiges Paar bestehend aus einem veränderten Text und einer hierzu passenden mit dem privaten Schlüssel von Alice erzeugten Signatur zu erzeugen.

Bob kann zudem die Authentizität der Nachricht überprüfen, d.h., ob die Nachricht tatsächlich von Alice stammt. Nur Alice hat Zugriff auf ihren privaten Schlüssel (davon gehen wir hier aus). Wenn Bob den öffentlichen Schlüssel von Alice benutzt (und dieser tatsächlich auch von Alice stammt), dann passt dieser öffentliche Schlüssel nur zum privaten Schlüssel von Alice. Bob erhält nur dann identische Fingerabdrücke, wenn der gesendete Text mit dem privaten Schlüssel von Alice - also von Alice - signiert wurde.

Da nur Alice Zugriff auf ihren privaten Schlüssel hat, kann Alice nachträglich nicht bestreiten, die signierte Nachricht an Bob verschickt zu haben. Mit der digitalen Signatur wird also auch die Verbindlichkeit des Nachrichtenaustauschs garantiert.

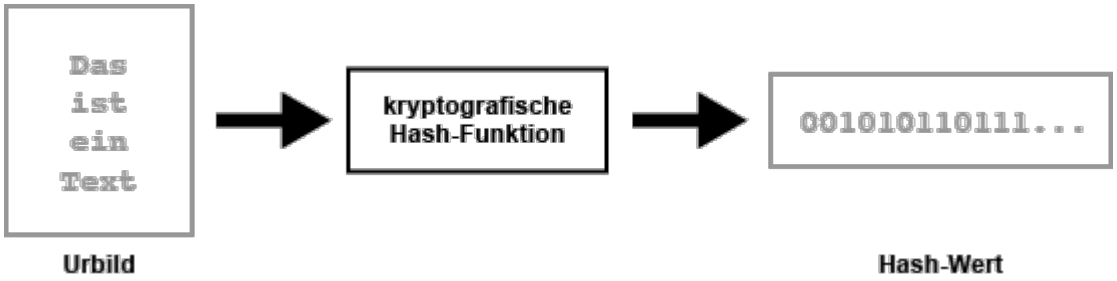
Bekannte Verfahren

Bekanntlich basieren Signaturverfahren auf asymmetrischen Verfahren, die sehr langsam arbeiten und von denen es nicht viele gibt. Das bekannteste und wohl am meisten eingesetzte Signaturverfahren ist RSA. Es gibt aber auch noch die Discrete Logarithm Signature Systems (DLSS). Dabei handelt es sich um eine Gruppe von Signaturverfahren auf Basis des diskreten Logarithmus. Dazu gehören ElGamal und DSA.

- RSA
- ElGamal
- DSA

Es gibt weitere DLSS-Verfahren, die in der Praxis nicht so häufig anzutreffen sind und deshalb hier nicht genannt werden.

Kryptografische Hash-Funktionen



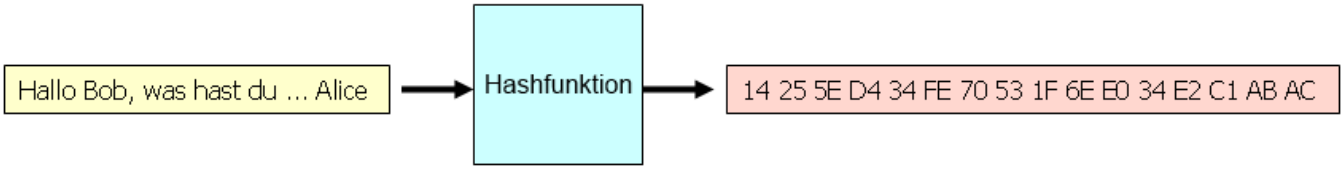
Kryptografische Hash-Funktionen sind ein wichtiges kryptografisches Instrument und bilden einen eigenen Bereich in der Kryptografie. Kryptografische Hash-Funktionen generieren aus beliebig langen Datensätzen eine Zeichenkette mit einer festen Länge (Angabe in Bit). Ein Datensatz kann ein Wort, ein Satz, ein längerer Text oder auch eine ganze Datei sein. Die erzeugte Zeichenkette wird als digitaler Fingerabdruck (Fingerprint), kryptografische Prüfsumme, Message Digest (MD) oder Message Authentication Code (MAC) bezeichnet. Gemeint ist damit in der Regel immer der sogenannte Hash-Wert oder auch nur Hash. Das ist ein digitaler Code, der nach Anwendung der kryptografischen Hash-Funktion als Ergebnis herauskommt.

Das Bilden eines Hash-Werts hat erst einmal nichts mit Kryptografie zu tun. Denn nicht alle Hash-Funktionen sind nach den Gesichtspunkten der Kryptografie eine kryptografische Hash-Funktion. Für „echte“ kryptografische Hash-Funktionen gibt es die unterschiedlichsten Begriffe und zusätzlich auch noch Produktbezeichnungen oder Leistungsmerkmale, die allerdings nichts darüber aussagen, ob sie kryptografischen Anforderungen entsprechen.

- Footprint-Funktion
- sichere Hash-Funktion
- Manipulation Detection Code (MDC)
- Message Integrity Code (MIC)
- Prüfsummenverfahren

Hash-Funktion

Eine Hash-Funktion ist eine Funktion, die Zeichenketten neue Zeichenketten einer fest vorgegebenen Länge zuordnet. Man nennt Funktionswerte von Hash-Funktionen auch Hash-Werte.



Im Prinzip erzeugt eine Hash-Funktion aus einem Datensatz, das als Urbild oder im Englischen Preimage bezeichnet wird, eine duale Zahl, die meist in hexadezimaler Schreibweise dargestellt und als Hash-Wert bezeichnet wird. Die Funktionsweise einer kryptografischen Hash-Funktion basiert auf einer Einwegfunktion, die sich sehr einfach rechnen lässt, aber deren Umkehrung dagegen sehr aufwendig bis unmöglich ist. Die Umkehrung vom Hash-Wert auf das Urbild zu schließen ist das was

man verhindern möchte.

Hash-Funktion als Einwegfunktion

Die in der Kryptologie benutzten Hash-Funktionen sind in der Regel Einwegfunktionen.

Bei einer Einwegfunktion ist es praktisch unmöglich, aus einem möglichen Zielwert einen Ausgangswert so zu bestimmen, dass der Zielwert Funktionswert zum Ausgangswert ist.

In mathematischer Kurzform kann man das so beschreiben: Eine Funktion f ist eine Einwegfunktion, wenn es praktisch unmöglich ist, zu gegebenem y aus der Zielmenge ein x aus der Definitionsmenge von f zu finden, so dass f(x) = y gilt.

Anforderungen an kryptografische Hash-Funktionen

- **Eindeutigkeit**: Eine identische Zeichenfolge muss zum selben Hash-Wert führen.
- **Reversibilität**: Der Hash-Wert darf nicht in die ursprüngliche Zeichenfolge zurückberechnet werden können.
- **Kollisionsresistenz**: Zwei unterschiedliche Zeichenfolgen dürfen nicht den gleichen Hash-Wert ergeben.

Nicht alle Hash-Funktionen erfüllen alle diese Anforderungen. Deshalb eignen sich nicht alle Hash-Funktionen für kryptografische Anwendungen, wie Authentisierung und Verschlüsselung.

Reversibilität

Grundsätzlich sollte es nicht möglich sein aus einem Hash-Wert die ursprünglichen Daten zurückzuberechnen. Weil mit der Zeit doch Möglichkeiten gefunden werden und die Rechenleistung steigt, gibt es immer bessere Verfahren aus einem Hash-Wert die ursprünglichen Daten zurück zu berechnen. Deshalb stellt sich mit der Zeit immer wieder heraus, dass Hash-Funktionen reversibel sind.

Kollisionsresistenz

Prinzipiell ist es so, dass ein Urbild beliebig viele Stellen und beliebig viele Werte einnehmen kann. Ein Hash-Wert ist allerdings auf eine bestimmte Länge begrenzt. So kann es vorkommen, dass ein beliebiger Hash-Wert unterschiedlichen Urbildern entspricht. Man spricht dann von einer Kollision. Bei einer guten Hash-Funktion sollte eine Kollision so wenige wie möglich vorkommen. Nehmen wir als Beispiel die Quersummenbildung. Hier kann es vorkommen, dass die Quersumme mehreren Zahlenwerten entsprechen kann. Aus Sicht der Kryptografie ist die Quersummenbildung also keine kryptografische Hash-Funktion. Die Kryptografie stellt an Hash-Funktionen und ihre Anwendungen höhere Anforderungen. Es sollte für einen Angreifer unmöglich sein Kollisionen zu erzeugen.

- Statistisch gesehen sollte jeder Hash-Wert etwa gleich oft vorkommen.
- Der Hash-Wert sollte auch bei kleinen Änderungen des Urbilds anders sein.

Um die Wahrscheinlichkeit von Kollisionen zu vermeiden, verwendet man immer bessere Verfahren, die meist längere Hash-Werte erzeugen. Beispielsweise sind die bekannten und beliebten Hash-Funktionen MD5 und SHA1 für Kollisions-Attacken verwundbar. Damit ist gemeint, dass ein anderer Datensatz den gleichen Hash-Wert erzeugen kann. Das heißt, dass ein MD5- oder SHA1-Hash nicht einzigartig ist. Besser ist es, SHA256 oder gleich SHA512 zu verwenden.

Angriffsszenarien

Bei einem **Urbildangriff** (engl. preimage attack) verfolgt der Angreifer das Ziel, zu einem gegebenen Hashwert einer unbekannten Nachricht (Erster Urbildangriff) oder zu einer gegebenen Nachricht selbst (Zweiter Urbildangriff) eine weitere Nachricht zu konstruieren, die denselben Hashwert besitzt.

Beispiel: Angenommen, ein Angreifer fängt ein signiertes Dokument ab. Er ist dann im Besitz des Dokumenttextes (z.B. „Hiermit bestelle ich 2 Konzertkarten zu je 40 €.“) sowie des zugehörigen Hashwerts. Der Angreifer versucht jetzt, aus der Nachricht oder dem Hashwert eine veränderte Nachricht mit demselben Hashwert zu erzeugen. Eine zusätzliche Schwierigkeit besteht darin, dass die neue Nachricht auch noch Sinn machen soll.

Eine andere Form von Angriff betrifft die Erzeugung von Kollisionen:

Bei einem **Kollisionsangriff** (engl. collision attack) verfolgt der Angreifer das Ziel, zwei verschiedene Dokumente zu konstruieren, die beide denselben Hashwert besitzen.

Beachte, dass es sich hier um unterschiedliche Angriffsszenarien handelt. Das zeigt sich auch in der Praxis. Während Kollisionsangriffe bei SHA-1 möglich sind, sind Urbildangriffe bei SHA-2 derzeit noch nicht möglich.

Kryptografische Hash-Funktionen

Kryptografische Hash-Funktionen bilden einen eigenen Bereich in der Kryptografie. An deren Entwicklung waren oft bekannte Kryptografen beteiligt, die man von anderen kryptografischen Verfahren her kennt.

- MD2 - Message Digest 2 mit 128 Bit
- MD4 - Message Digest 4 mit 128 Bit
- MD5 - Message Digest 5 mit 128 Bit
- RIPEMD
- RIPEMD-160
- Tiger
- WHIRLPOOL
- SHA-1 mit 160 Bit
- SHA-2 mit 224, 256, 384 und 512 Bit
- SHA-3 mit 224, 256, 384 und 512 Bit

Mehr Infos zu MD5 und SHA

Sicherheitsinfrastruktur

Hier geht es um die Frage, wie man gewährleisten kann, dass ein öffentlich bereit gestellter Schlüssel tatsächlich zu der Person gehört, die sich als Eigentümer ausgibt.

Mit Hilfe eines asymmetrischen Kryptosystems können Nachrichten in einem Netzwerk digital signiert und verschlüsselt werden. Sichere Kryptosysteme können bei geeigneter Wahl der Parameter (z. B. der Schlüssellänge) auch bei Kenntnis des Verfahrens (vgl. Kerckhoffs' Prinzip) zumindest nach heutigem Kenntnisstand nicht in überschaubarer Zeit gebrochen werden.

In asymmetrischen Kryptosystemen benötigt der Sender für eine verschlüsselte Übermittlung den öffentlichen Schlüssel (public key) des Empfängers. Dieser könnte z. B. per E-Mail versandt oder von einer Web-Seite heruntergeladen werden. Dabei muss sichergestellt sein, dass es sich tatsächlich um den Schlüssel des Empfängers handelt und nicht um eine Fälschung eines Betrügers.

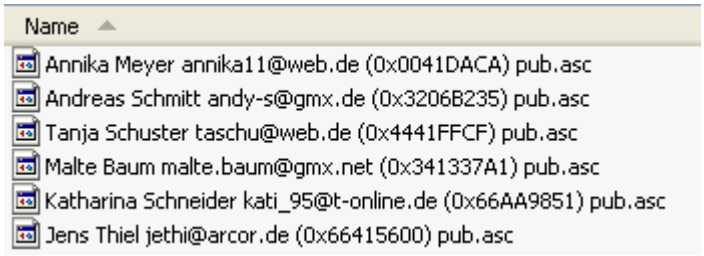
Hierzu dienen nun digitale Zertifikate, die die Authentizität eines öffentlichen Schlüssels und seinen zulässigen Anwendungs- und Geltungsbereich bestätigen. Das digitale Zertifikat ist selbst durch eine digitale Signatur geschützt, deren Echtheit mit dem öffentlichen Schlüssel des Ausstellers des Zertifikates geprüft werden kann.

Um die Authentizität des Ausstellerschlüssels zu prüfen, wird wiederum ein digitales Zertifikat benötigt. Auf diese Weise lässt sich eine Kette von digitalen Zertifikaten aufbauen, die jeweils die Authentizität des öffentlichen Schlüssels bestätigen, mit dem das vorhergehende Zertifikat geprüft werden kann. Eine solche Kette von Zertifikaten wird Validierungspfad oder Zertifizierungspfad genannt. Auf die Echtheit des letzten Zertifikates (und des durch dieses zertifizierten Schlüssels) müssen sich die Kommunikationspartner ohne ein weiteres Zertifikat verlassen können.

Vertrauen in Schlüssel

Blindes Vertrauen in Schlüssel als Problem

Andreas, Annika, Jens, Katharina, Malte und Tanja haben ihre öffentlichen Schlüssel in ein gemeinsam zugängliches Verzeichnis kopiert, so dass jeder sich die benötigten öffentlichen Schlüssel kopieren kann.



Bei einem erneuten Blick in das Verzeichnis taucht plötzlich ein weiterer Schlüssel auf.

Mit welchen öffentlichen Schlüssel soll Annika nun Jens die Nachricht verschlüsseln und schicken?

Schlüsselserver?

Öffentliche Schlüssel von Kommunikationspartnern kann man sich auch auf Schlüsselservern wie z.B. keys.openpgp.org besorgen. Aus Datenschutzgründen sind Recherchen auf diesen Servern heutzutage nur noch gezielt möglich. D.h. man muss nach einem Fingerabdruck (Fingerprint) einer Person suchen, den man erhalten hat, oder nach der E-Mail Adresse, die einem bekannt ist. Beispielsweise findet man den Fingerabdruck von Linus Neumann auf seiner Website. Zu diesem Fingerabdruck kann man dann auf https://keys.openpgp.org oder https://keyserver.ubuntu.com den passenden Key herunterladen.

Schlüssel überprüfen

Bevor du den öffentlichen Schlüssel einer anderen Person benutzt, solltest du dich von der Echtheit des Schlüssels überzeugen.

Wenn der öffentliche Schlüssel dir direkt von einem (vertrauenswürdigen) Bekannten übergeben wird, dann kannst du (in der Regel) von einem echten Schlüssel ausgehen.

Wenn du einen öffentliche Schlüssel über ein unsicheres Kommunikationsmedium (z.B. per E-Mail) erhalten hast oder von einem Schlüsselserver heruntergeladen hast, dann solltest du dich vergewissern, ob die Angaben zum Eigentümer stimmen. Du kannst dich z.B. mit der Person treffen (oder - wenn du die Stimme eindeutig erkennst - mit der Person telefonieren) und einen Datenabgleich machen.

Beim Datenabgleich solltest du den Fingerabdruck des Schlüssels genauestens überprüfen.

Schlüssel zertifizieren

Erst wenn du die Echtheit eines Schlüssels genauestens geprüft hast bzw. wenn du ganz sicher bist, dass ein Schlüssel zu einer bestimmten Person gehört, dann solltest du den betreffenden Schlüssel zertifizieren.

Einen Schlüssel zertifiziert man, indem man ihn mit einer digitalen Signatur versieht.

Web of Trust

Echtheit öffentlicher Schüssel als Problem

Zum Verschlüsseln von Dokumenten bzw. zum Überprüfen digitaler Signaturen benötigt man den öffentlichen Schlüssel des Kommunikationspartners. Aber, wie kommt man an den gewünschten öffentlichen Schlüssel? Oft ist eine persönliche Übergabe nicht möglich und man muss sich den öffentliche Schlüssel (über ein unsicheres Kommunikationsmedium) schicken lassen oder auf einen Schlüsselserver besorgen. In beiden Fällen ergibt sich dann das Problem, dass man nicht sicher sein kann, dass der erhaltene Schlüssel tatsächlich zu der angegebenen Person gehört.

Schlüssel zertifizieren

Wenn man (nach einer sorgfältigen Überprüfung) sicher ist, dass ein Schlüssel zu der angegebenen Person gehört, dann kann man den Schlüssel mit seiner digitalen Unterschrift beglaubigen / zertifizieren.

In der folgenden Abbildung kann man solche Zertifizierungen erkennen (z.B.: snerz@bvpk.org)

Search results for 'snerz@bvpk.org'

```
Type bits/keyID            cr. time   exp time   key expir
```

```
pub (4)dsa1024/39c40c2fca5b31804c6f82e184d2b243449d222e 2008-11-28T21:30:27Z

uid Sebastian Nerz <basti@tirsales.de>
sig cert  84d2b243449d222e 2008-11-28T21:33:32Z ____________________ ____________________ [selfsig]
sig cert  84d2b243449d222e 2008-11-28T21:30:27Z ____________________ ____________________ [selfsig]
sig cert  926ae196022ce281 2009-08-30T00:30:30Z ____________________ ____________________ 926ae196022ce281
sig cert  9f587ee862a25e4e 2009-08-30T11:44:39Z ____________________ ____________________ 9f587ee862a25e4e
sig cert  021b54a7fe1dd1df 2009-08-30T11:47:14Z ____________________ ____________________ 021b54a7fe1dd1df
sig cert  a48fb68bd58cb000 2009-08-30T14:49:28Z ____________________ ____________________ a48fb68bd58cb000
sig cert  2189b0bd2c8c1429 2009-09-02T10:00:21Z ____________________ ____________________ 2189b0bd2c8c1429
sig cert  208f84c45ff25b4d 2010-04-19T19:37:21Z ____________________ ____________________ 208f84c45ff25b4d
sig cert  c0ac6eb914fe16c1 2010-06-01T14:30:48Z ____________________ ____________________ c0ac6eb914fe16c1
sig cert  be7d227833742d65 2010-06-01T14:31:26Z ____________________ ____________________ be7d227833742d65
sig cert  4473759d0191d5ed 2010-12-27T03:24:49Z ____________________ ____________________ 4473759d0191d5ed
```

Klickt man auf die erste Zertifizierung, sieht man dass hier z.B. ein Benutzer mit dem Namen Matthias Binninger den öffentlichen Schlüssel von Sebastian Nerz mit seinem privaten Schlüssel signiert hat. Zusätzlich ist der öffentlichen Schlüssel von Sebastian Nerz mit seinem eigenen privaten Schlüssel signiert (selfsig). Eine solche Selbstzertifizierung wird vom benutzen System standardmäßig vorgenommen.

keyserver.ubuntu.com/pks/lookup

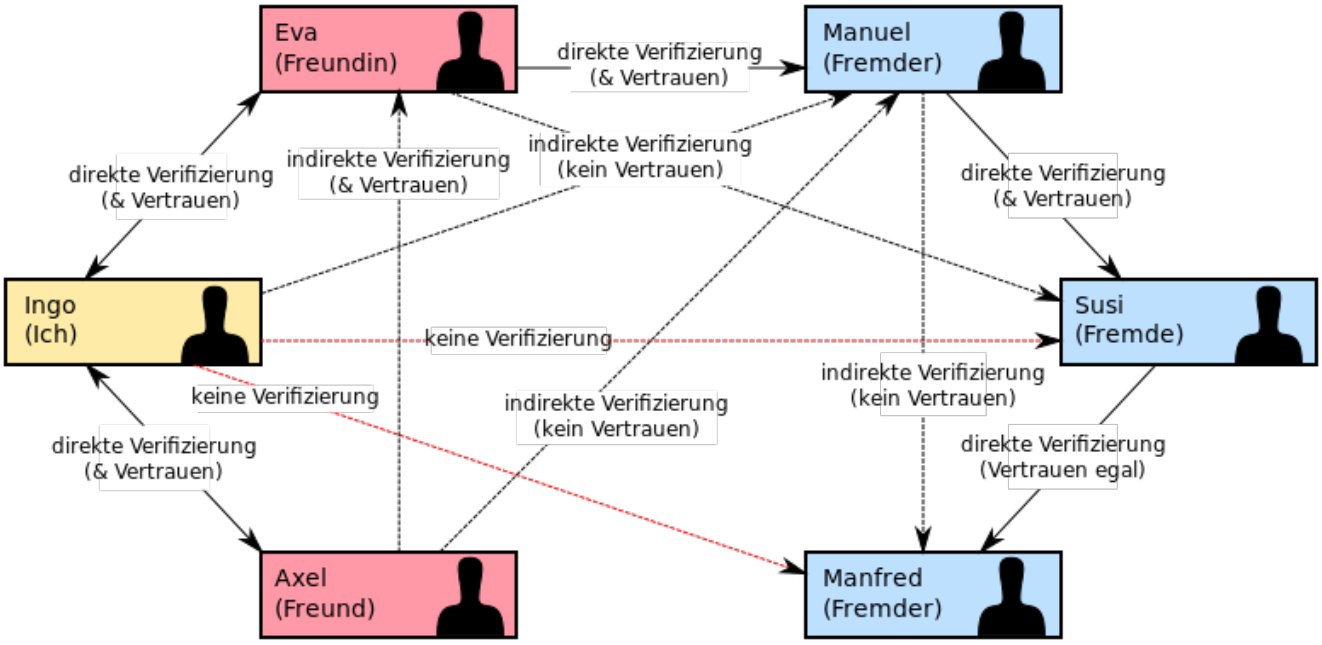
Search results for '0x926ae196022ce281'

```
Type bits/keyID           cr. time   exp time   key expir
```

```
pub (4)rsa2048/926ae196022ce281 2009-08-28T01:35:51Z

uid Matthias Binninger <mail@matthias-binninger.de>
sig cert  926ae196022ce281 2009-08-28T01:39:35Z ____________________ ____________________ [selfsig]
sig cert  926ae196022ce281 2009-08-28T01:35:51Z ____________________ ____________________ [selfsig]
sig cert  f2bb81d8cc8d3de2 2009-08-30T08:40:39Z ____________________ ____________________ f2bb81d8cc8d3de2
sig cert  9f587ee862a25e4e 2009-08-30T10:26:06Z ____________________ ____________________ 9f587ee862a25e4e
sig cert  1133ff1848a587a7 2009-08-30T10:50:03Z ____________________ ____________________ 1133ff1848a587a7
sig cert  021b54a7fe1dd1df 2009-08-30T11:49:20Z ____________________ ____________________ 021b54a7fe1dd1df
sig cert  58c691ffedcb6ea7 2009-08-30T12:00:50Z ____________________ ____________________ 58c691ffedcb6ea7
sig cert  a48fb68bd58cb000 2009-08-30T14:49:16Z ____________________ ____________________ a48fb68bd58cb000
sig cert  7d5ba9edd43794cc 2009-08-30T20:51:57Z ____________________ ____________________ 7d5ba9edd43794cc
sig cert  f8c376a1a2c51749 2009-08-30T21:32:22Z ____________________ ____________________ f8c376a1a2c51749
sig cert  2189b0bd2c8c1429 2009-09-02T09:48:46Z ____________________ ____________________ 2189b0bd2c8c1429
sig cert  02907183f0818b12 2009-09-06T18:59:09Z ____________________ ____________________ 02907183f0818b12
sig cert  646cbf3ad40db3d7 2010-03-31T23:19:15Z ____________________ ____________________ 646cbf3ad40db3d7
```

Aufbau eines Vertrauensnetzes (Web of Trust)

Durch das Zertifizieren von Schlüsseln lässt sich ein Vertrauensnetz aufbauen.



Die vorliegende Abbildung geht von folgenden Zertifizierungen aus:

```
Ingo zertifiziert den öffentlichen Schlüssel von Eva und Axel.
```

```
Eva zertifiziert den öffentlichen Schlüssel von Ingo und Manuel.
...
```

Hierdurch werden Vertrauensbeziehungen erstellt:

```
Ingo vertraut den Angaben des öffentlichen Schlüssels von Eva und Axel.
Eva vertraut den Angeben des öffentlichen Schlüssels von Ingo und Manuel.
...
```

Nach dem Motto der Freund meines Freundes ist auch mein Freund pflanzen sich Vertrauensbeziehungen fort:

```
Ingo vertraut den Angaben des öffentlichen Schlüssels von Eva.
Eva vertraut den Angeben des öffentlichen Schlüssels von Manuel.
```

also:

```
Ingo vertraut den Angaben des öffentlichen Schlüssels von Manuel.
```

Insgesamt ergibt sich auf diese Weise ein Netzwerk von Vertrauensbeziehungen.

In einem Web of Trust funktioniert das so:

1. Alice erzeugt für sich ein Schlüsselpaar und signiert es. Außerdem schickt sie den öffentlichen Teil an einen Schlüsselserver (key server), damit andere Teilnehmer leichten Zugriff darauf haben.
2. Bob möchte mit Alice verschlüsselt kommunizieren. Dazu besorgt er sich Alices Schlüssel von einem Schlüsselserver, muss aber noch sicherstellen, dass er wirklich den richtigen Schlüssel bekommen hat: Ein Angreifer könnte sich für Alice ausgeben und einen von ihm erzeugten Schlüssel an den Schlüsselserver schicken. Jeder, der meint, eine Nachricht nur für Alice zu verschlüsseln, würde sie in Wirklichkeit für den Angreifer verschlüsseln.
3. Bob bittet Alice (z. B. bei einem Telefonanruf oder einem persönlichen Treffen) um den Fingerprint ihres öffentlichen Schlüssels. Diesen vergleicht er mit dem des Schlüssels, den er vom Schlüsselserver erhalten hat.
4. Stimmen beide Fingerprints überein, kann Bob davon ausgehen, den richtigen Schlüssel erhalten zu haben. Darum signiert er den öffentlichen Schlüssel von Alice (genauer: eine oder mehrere ihrer User-IDs) mit seinem privaten und schickt diese Signatur an den Schlüsselserver.
5. Möchte jetzt Carl mit Alice verschlüsselt kommunizieren, besorgt er sich genau wie Bob Alices öffentlichen Schlüssel vom Schlüsselserver. Dann stellt er fest, dass Bob Alices Schlüssel bereits überprüft hat. Wenn Carl Bobs Schlüssel schon kennt und er Bob vertraut, dass Bob vor der Signatur fremder Schlüssel eine gründliche Überprüfung durchführt, dann muss er nicht erst Alice treffen und diese Prüfung wiederholen. Er vertraut dem Schlüssel von Alice allein aufgrund Bobs vertrauenswürdiger Signatur. Wenn Carl sein Sicherheitsniveau erhöhen möchte oder er speziell Bobs Signaturen nur eingeschränkt vertraut, kann er sein Kryptosystem so konfigurieren, dass mehrere von ihm akzeptierte Signaturen vorhanden sein müssen, damit ein Schlüssel automatisch als gültig angesehen wird.

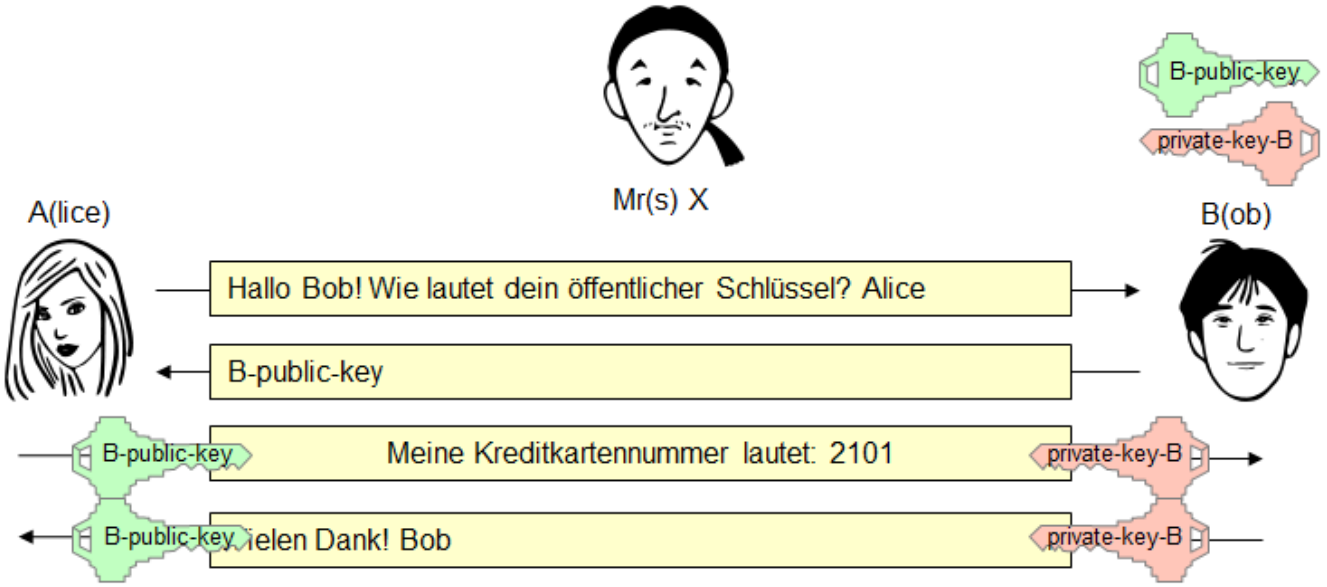
Datenschutzprobleme

Beachte, dass beim Veröffentlichen von Schlüsseln auch persönliche Daten veröffentlicht werden. Beachte auch, dass einmal veröffentliche Schlüssel mitsamt der persönlichen Daten nicht wieder gelöscht werden können.

(Wo)Man in the middle Angriff

Mr(s) X hört mit!

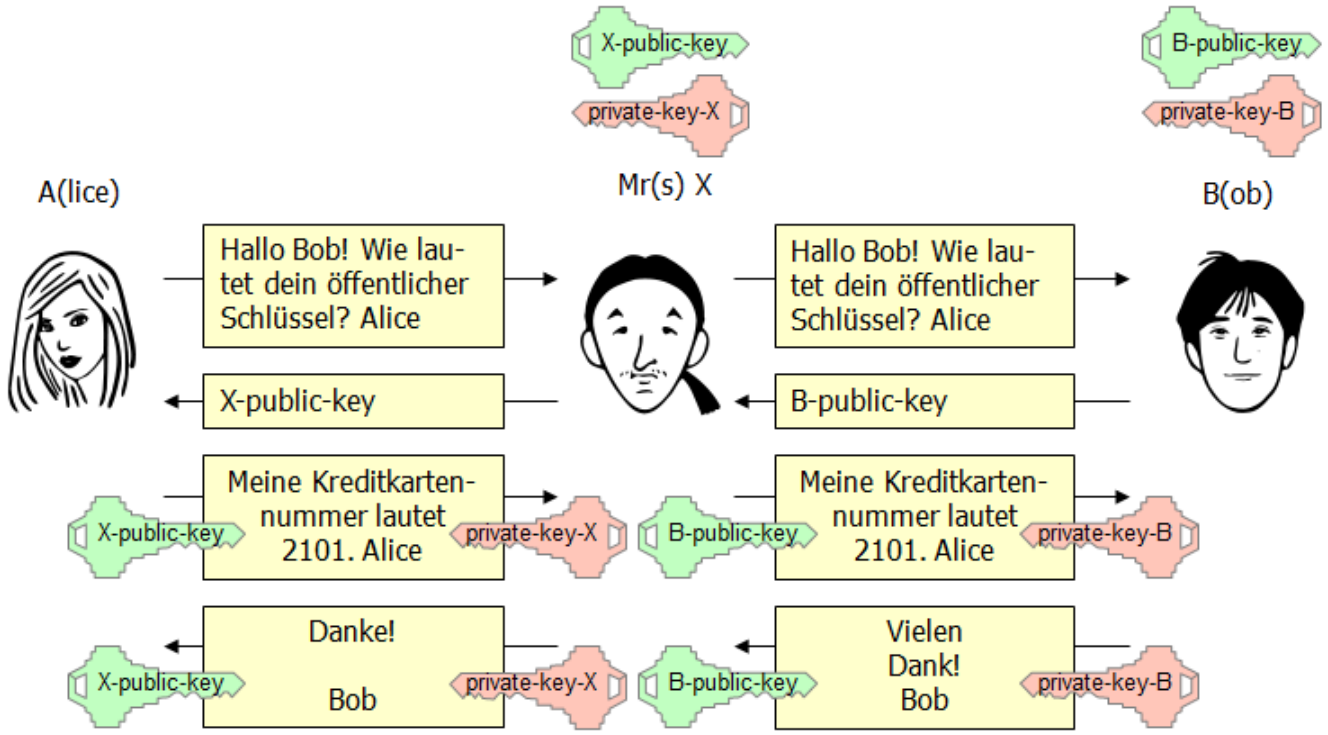
Alice möchte bei Bobs Internetversand eine Bestellung aufgeben und muss ihm deshalb die Nummer ihrer Kreditkarte übermitteln. Diese vertrauliche Information soll keinesfalls in die Hände von Mr(s) X fallen.



Bob hat sich ein Schlüsselpaar bestehend aus einem öffentlichen und einem privaten Schlüssel erzeugt. Bob schickt seinen öffentlichen Schlüssel auf Anfrage an Alice. Alice benutzt nun diesen Schlüssel, um ihre Kreditkartennummer an Bob zu verschicken. Bob bestätigt den Erhalt der Kreditkartennummer mit einer signierten Nachricht. Beurteile die Sicherheit des beschriebenen Szenarios: Kann sich Alice sicher sein, dass sie wirklich Bobs öffentlichen Schlüssel erhält?

Mr X schiebt sich dazwischen

Mr(s) X hat eine Möglichkeit gefunden, den Datenverkehr zwischen Alice und Bob abzufangen.

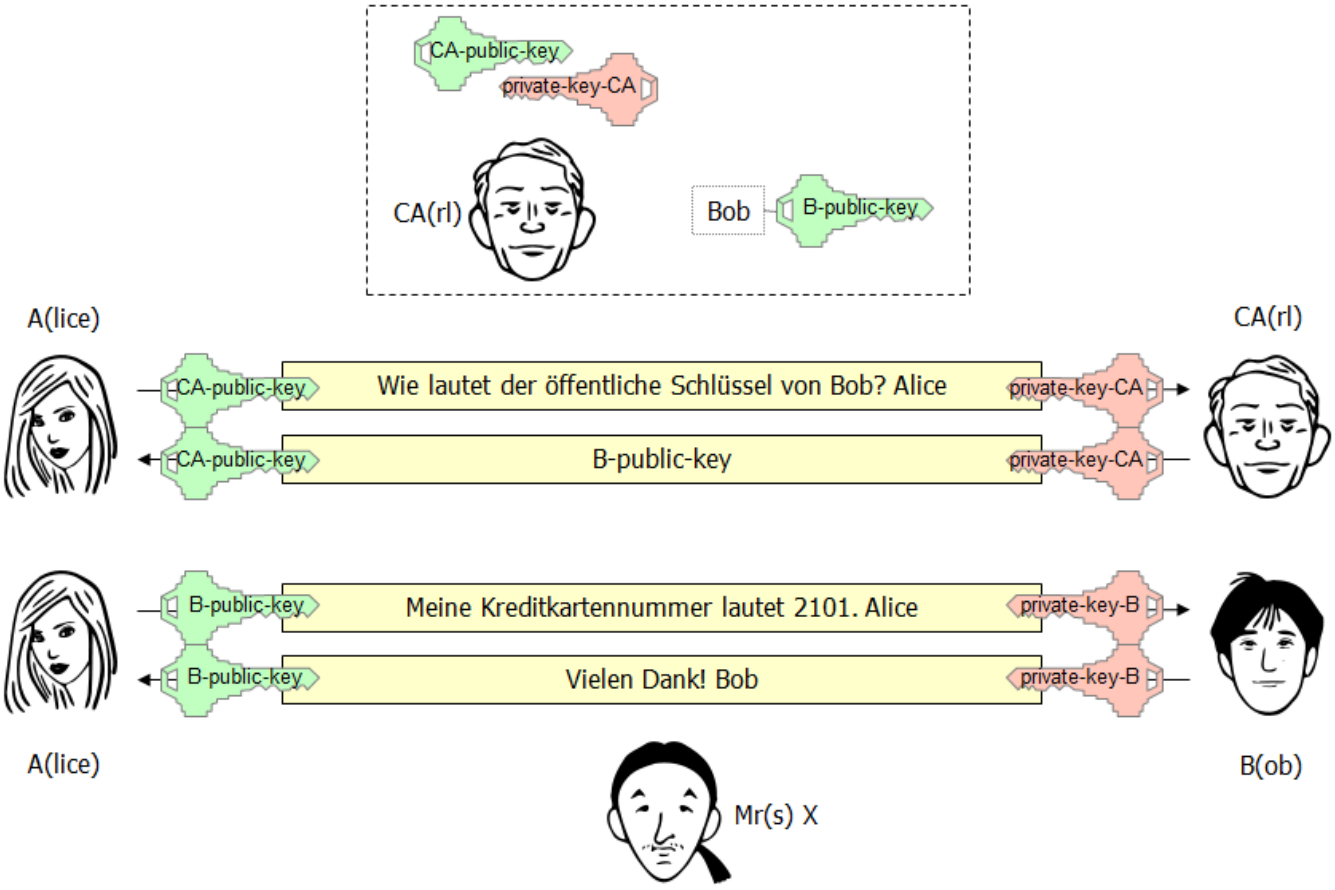


Warum merkt Alice oder Bob hier nichts vom (Wo)Man in the middle Angriff? Worin besteht die Schwierigkeit? Siehst du Lösungsansätze?

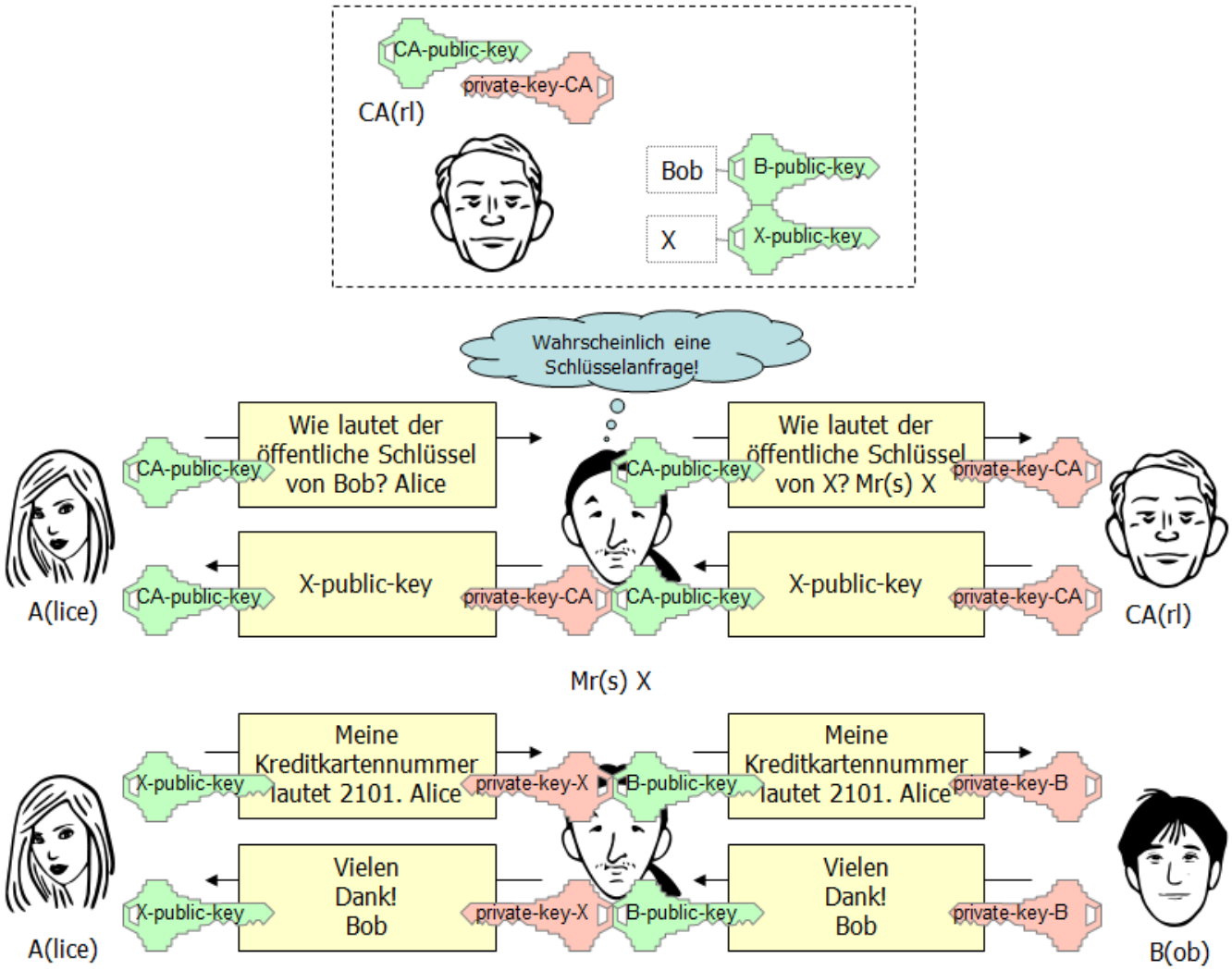
CA(rl) kommt ins Spiel!

CA(rl) bietet an, gegen ein Entgeld öffentliche Schlüssel zu verwalten. Er macht das aber nur, wenn man sie ihm vorab persönlich übergibt und sich ausweist. Als Gegenleistung überbringt CA(rl) seinen eigenen öffentlichen Schlüssel persönlich an alle gewünschten Personen.

Bob ist Internethändler und nimmt das Angebot von CA(rl) in Anspruch. Er hinterlässt seinen öffentlichen Schlüssel bei CA(rl) und beauftragt ihn, seinen - also CA(rl)s - öffentlichen Schlüssel an Alice zu übergeben.



- Wie erhält Alice hier den öffentlichen Schlüssel von Bob? Kann sie sicher sein, dass sie wirklich Bobs öffentlichen Schlüssel erhält?
- Mr(s) X kann sich auch hier zwischen Alice und Bob schieben. Analysiere hierzu die folgende Abbildung. Von welchen Annahmen geht man hier aus?

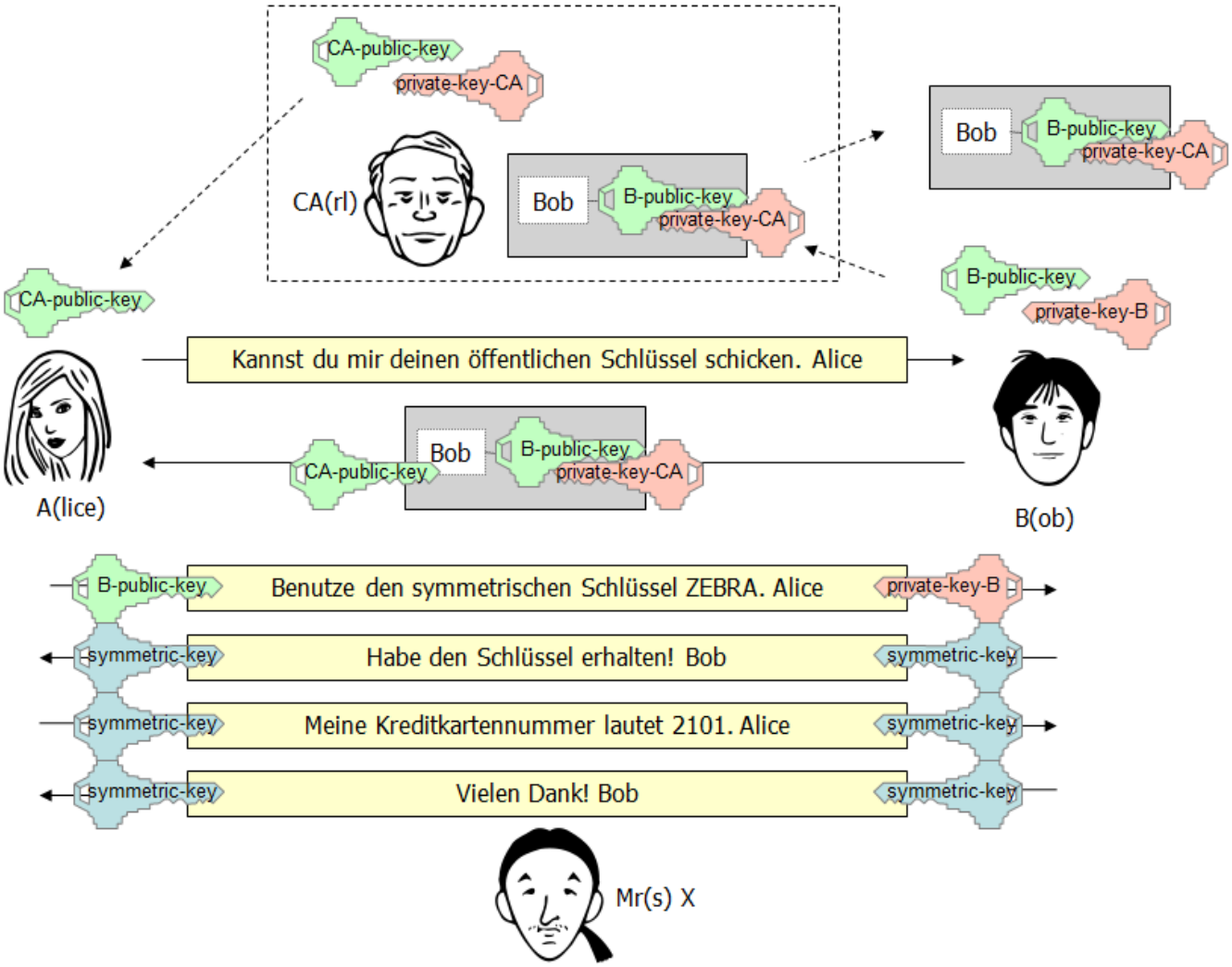


CA(rl) signiert die hinterlegten Schlüssel!

CA(rl) übernimmt jetzt die Rolle der sogenannten Certification Authority (CA). Seine Aufgabe ist es, die Echtheit von Schlüsseln zu gewährleisten.

CA(rl) bietet hier an, gegen ein Entgeld öffentliche Schlüssel zu signieren. Er macht das aber nur, wenn man sie ihm vorab persönlich übergibt und sich ausweist. Zusätzlich überbringt CA(rl) seinen eigenen öffentlichen Schlüssel persönlich an alle gewünschten Personen.

Bob ist Internethändler und nimmt das Angebot von CA(rl) in Anspruch. Er lässt seinen öffentlichen Schlüssel von CA(rl) signieren und beauftragt CA(rl), seinen - also CA(rl)s - öffentlichen Schlüssel an Alice zu übergeben.



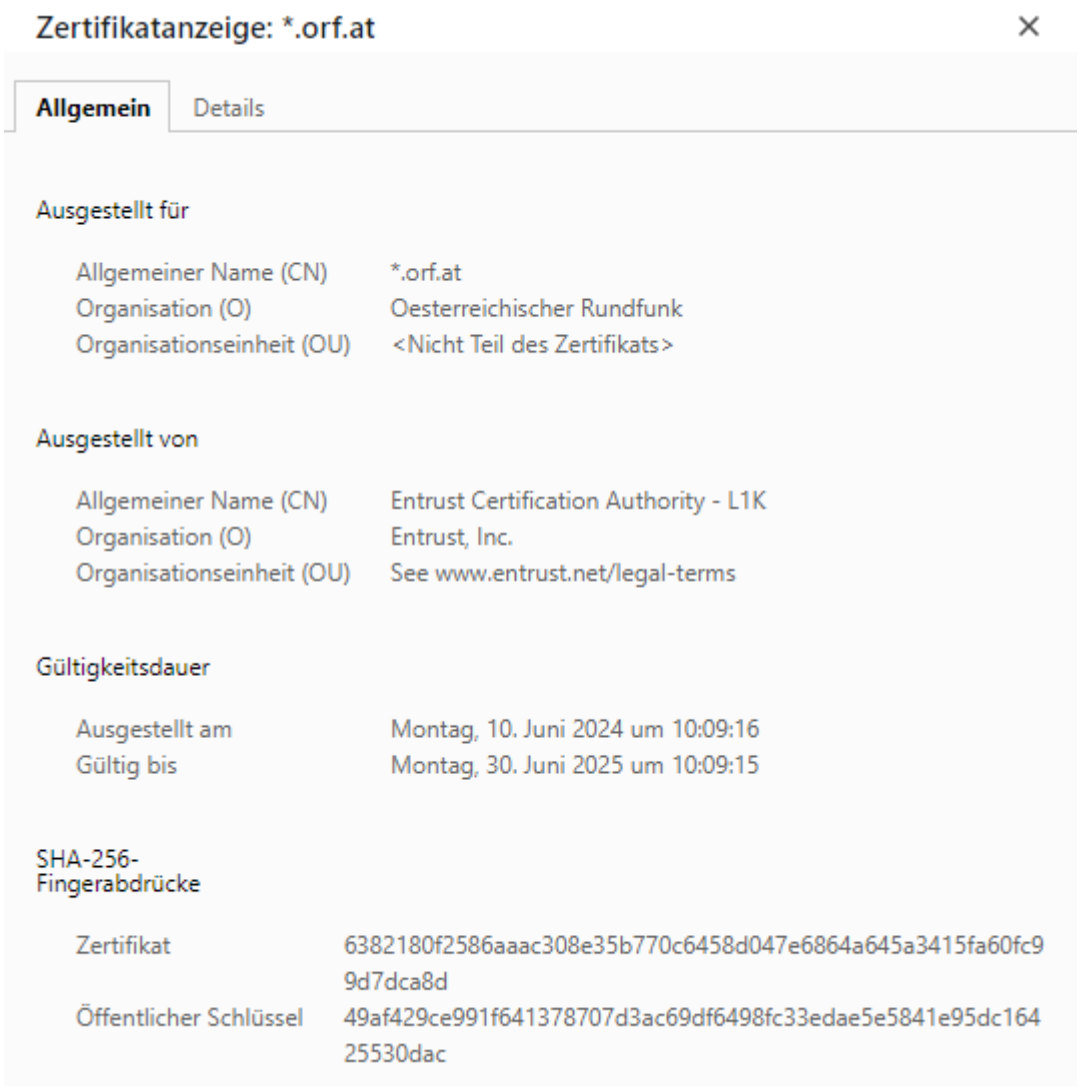
Was ist ein digitales Zertifikat

Wenn man den öffentlichen Schlüssel einer Person zum Verschlüsseln benutzt, dann sollte man auch sicher sein, dass dieser Schlüssel tatsächlich zur angegebenen Person gehört.

Ein **Public-Key-Zertifikat** dient dazu, die Zugehörigkeit eines öffentlichen Schlüssels zu einem bestimmten Eigentümer zu bestätigen.

Ein Public-Key-Zertifikat enthält in der Regel eine ganze Reihe von Informationen, u a.:

- den zu bestätigenden öffentlichen Schlüssel
- den Eigentümer des Schlüssels
- den Aussteller des Zertifikats
- die benutzten kryptografischen Verfahren
- die Gültigkeitsdauer des Zertifikats
- ...
- eine digitale Signatur des Ausstellers zur Bestätigung aller Informationen

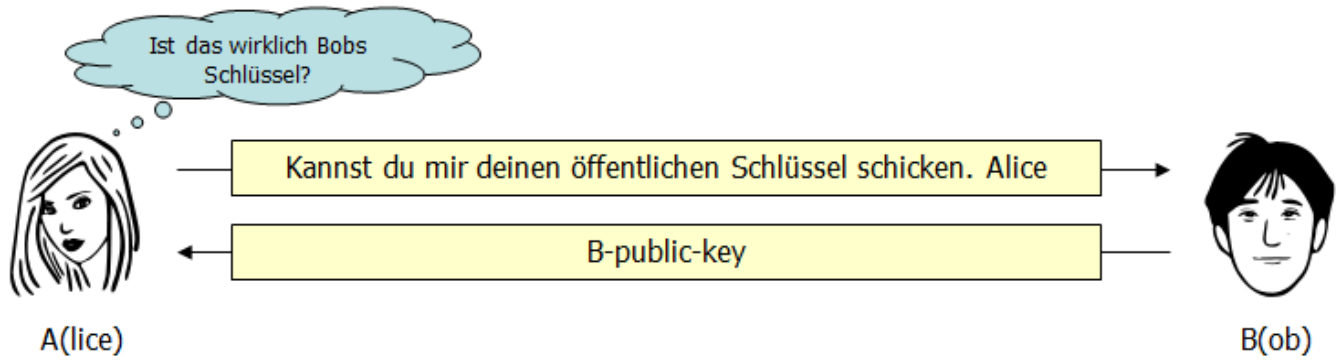


Public Key Infrastruktur (PKI)

Eine Public Key Infrastruktur (kurz: PKI) basiert auf einem asymmetrischen Kryptosystem, bei dem öffentliche und private Schlüssel zum Verschlüsseln und zum Signieren von Dokumenten benutzt werden.

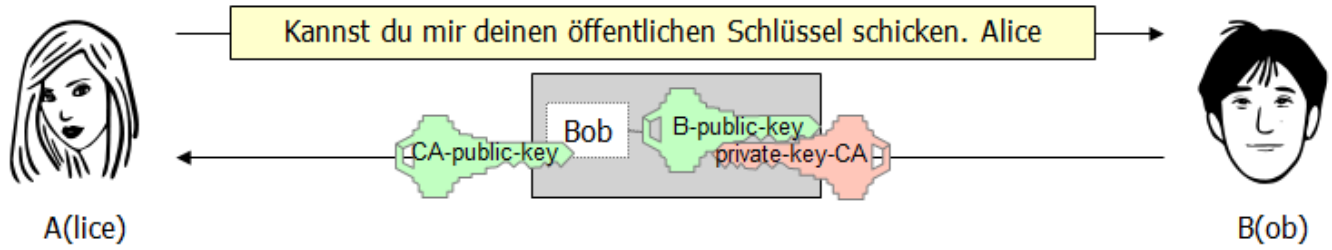
Zertifikate

Zum Verschlüsseln von Dokumenten bzw. zum Überprüfen digitaler Signaturen benötigt man den öffentlichen Schlüssel des Kommunikationspartners. Ein Grundproblem einer PKI besteht darin, an den öffentlichen Schlüssel des Kommunikationspartners zu gelangen. Oft ist eine persönliche Übergabe nicht möglich und der öffentliche Schlüssel muss über ein unsicheres Kommunikationsmedium übertragen oder öffentlich bereitgestellt werden.



Bei dieser (unpersönlichen) Weitergabe des öffentlichen Schlüssels muss dann sichergestellt werden, dass der übergebene öffentliche Schlüssel tatsächlich zu der Person gehört, die sich als Eigentümer ausgibt.

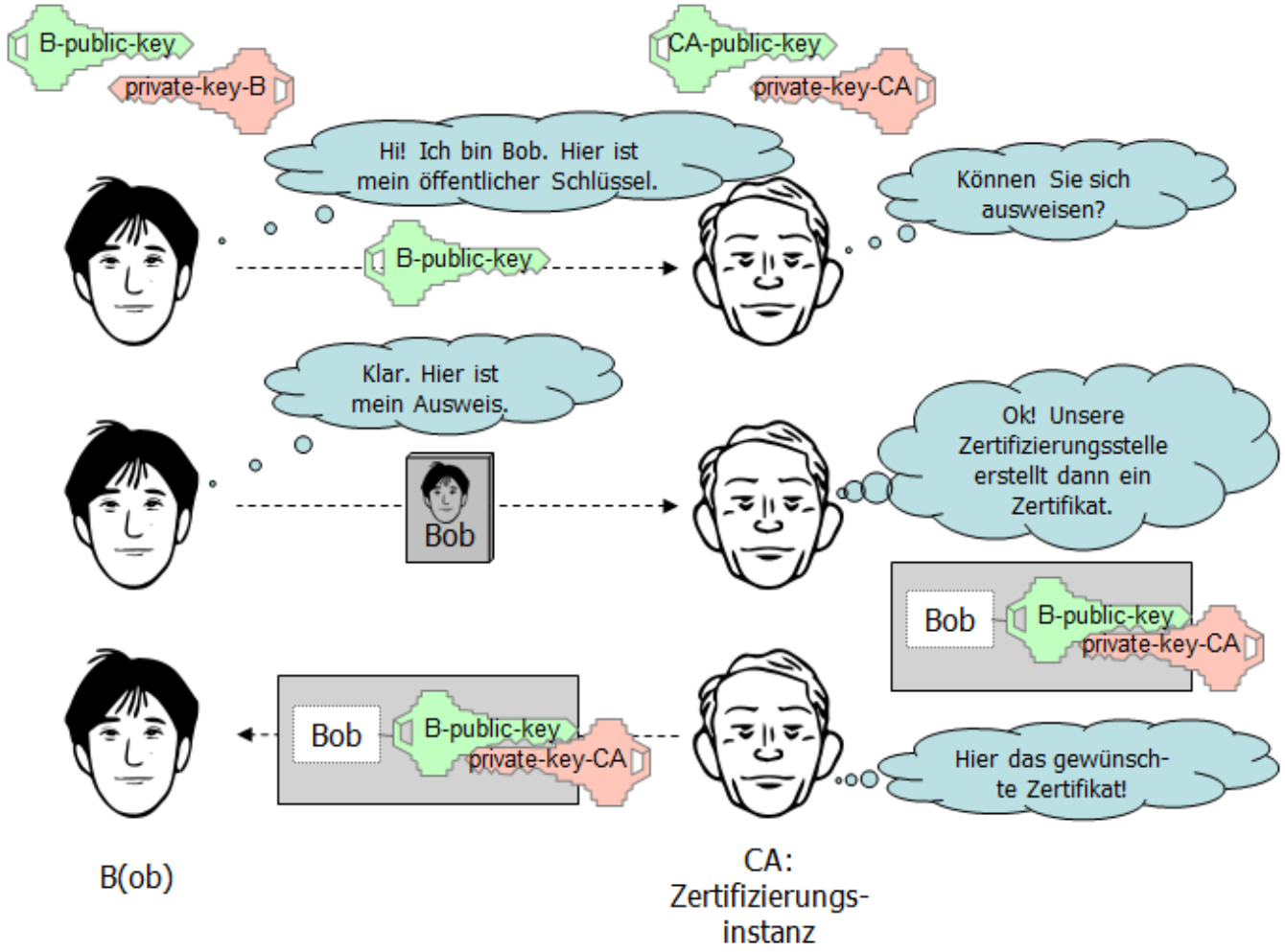
Hierzu benutzt man Zertifikate. Mit einem (Public Key) Zertifikat wird die Authentizität eines öffentlichen Schlüssels gewährleistet. Eine vertauenswürdige Instanz die **Zertifizierungsinstanz (CA - Certificate authority)** - beglaubigt mit ihrer digitalen Signatur, dass der öffentliche Schlüssel tatsächlich zur Person gehört, die sich als Eigentümer ausgibt.



Der Empfänger eines Zertifikats hat jetzt die Möglichkeit, den signierten Schlüssel (mit Angabe des Eigentümers) zu überprüfen.

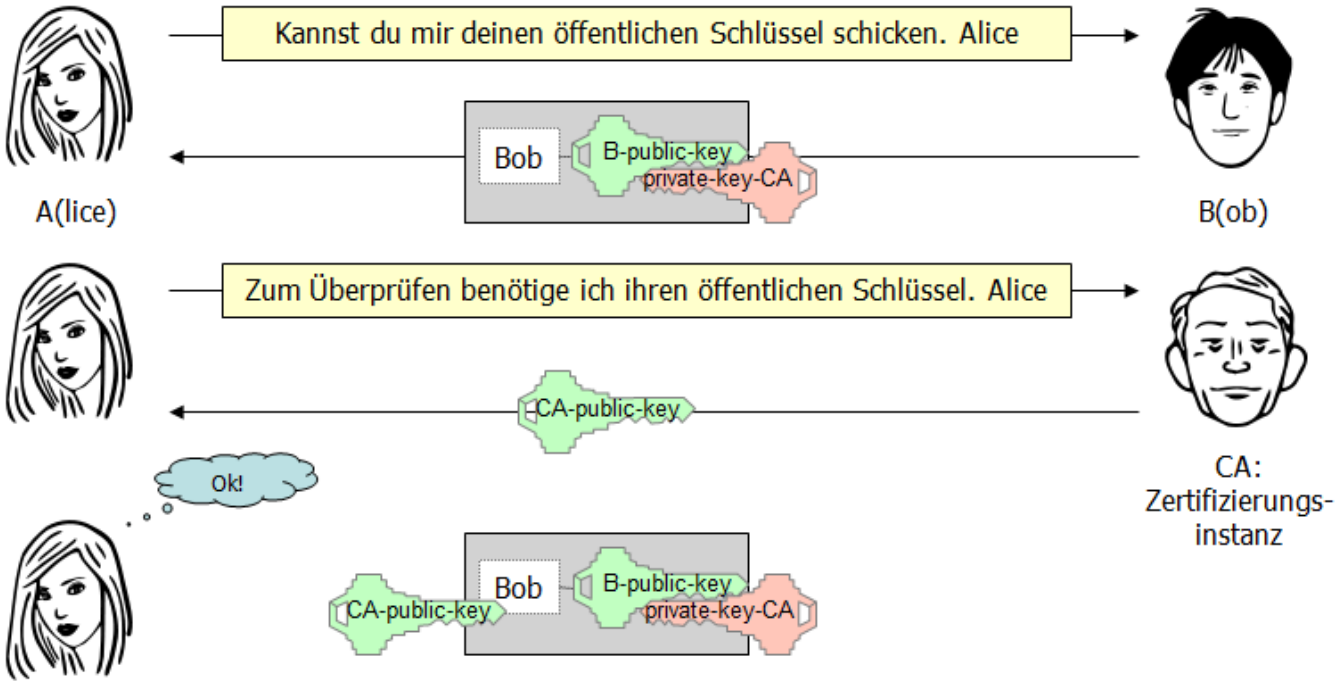
Ausstellung von Zertifikaten

Zertifikate werden von sogenannten Zertifizierungsstellen (engl. certification authority, kurz CA) ausgestellt. Das sind vertrauenswürdige Organisationen, die gegen eine Gebühr vorgelegte öffentliche Schlüssel überprüfen und signieren. Diese Zertifizierungsstellen müssen strenge Auflagen erfüllen und werden in Österreich von der Rundfunk & Telekom Regulierungs GmbH, kurz RTR überwacht.

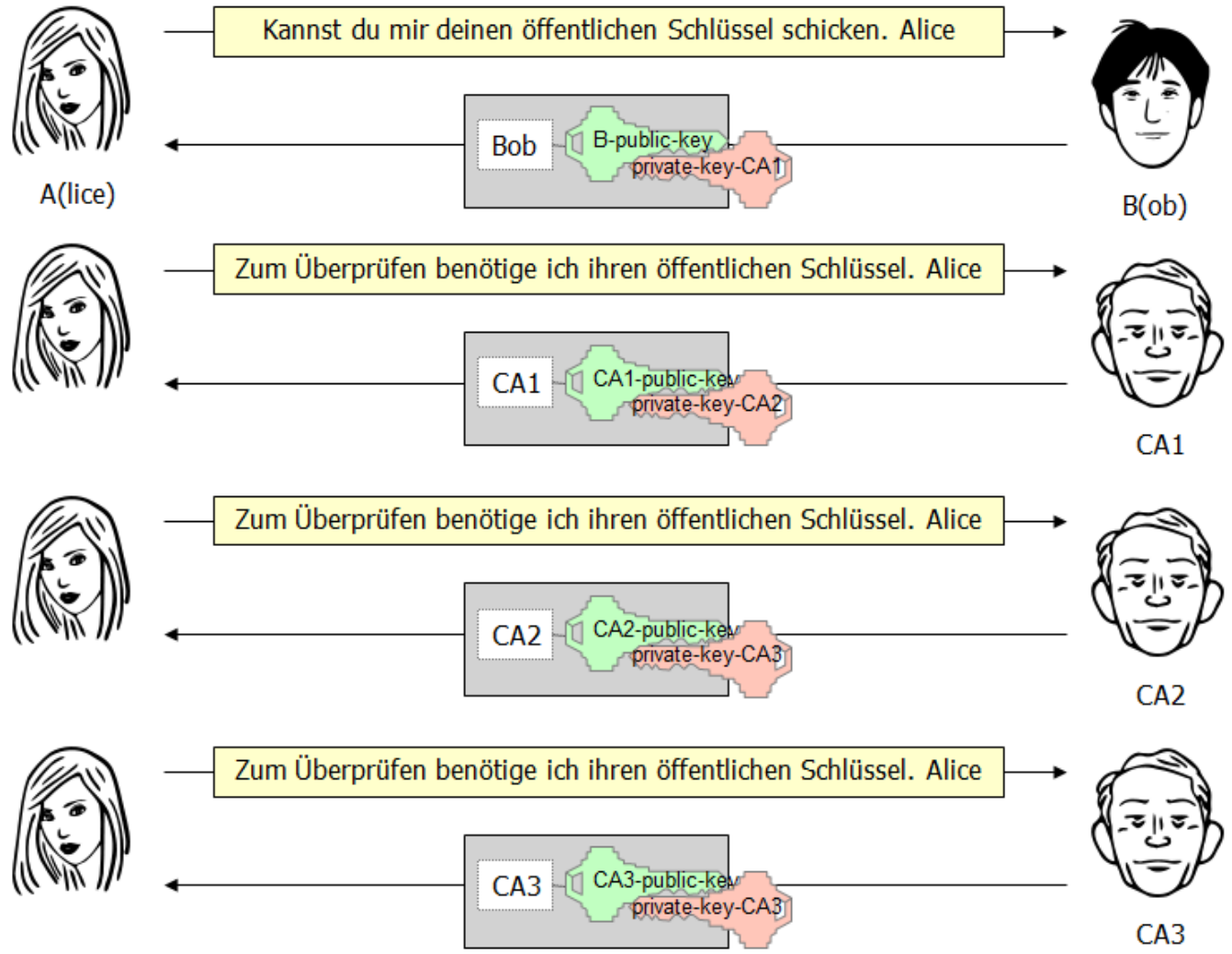


Überprüfung von Zertifikaten

Zertifikate werden überprüft, indem man die digitale Signatur der Zertifizierungsstelle überprüft. Hierzu benötigt man den öffentlichen Schlüssel der Zertifizierungsstelle.

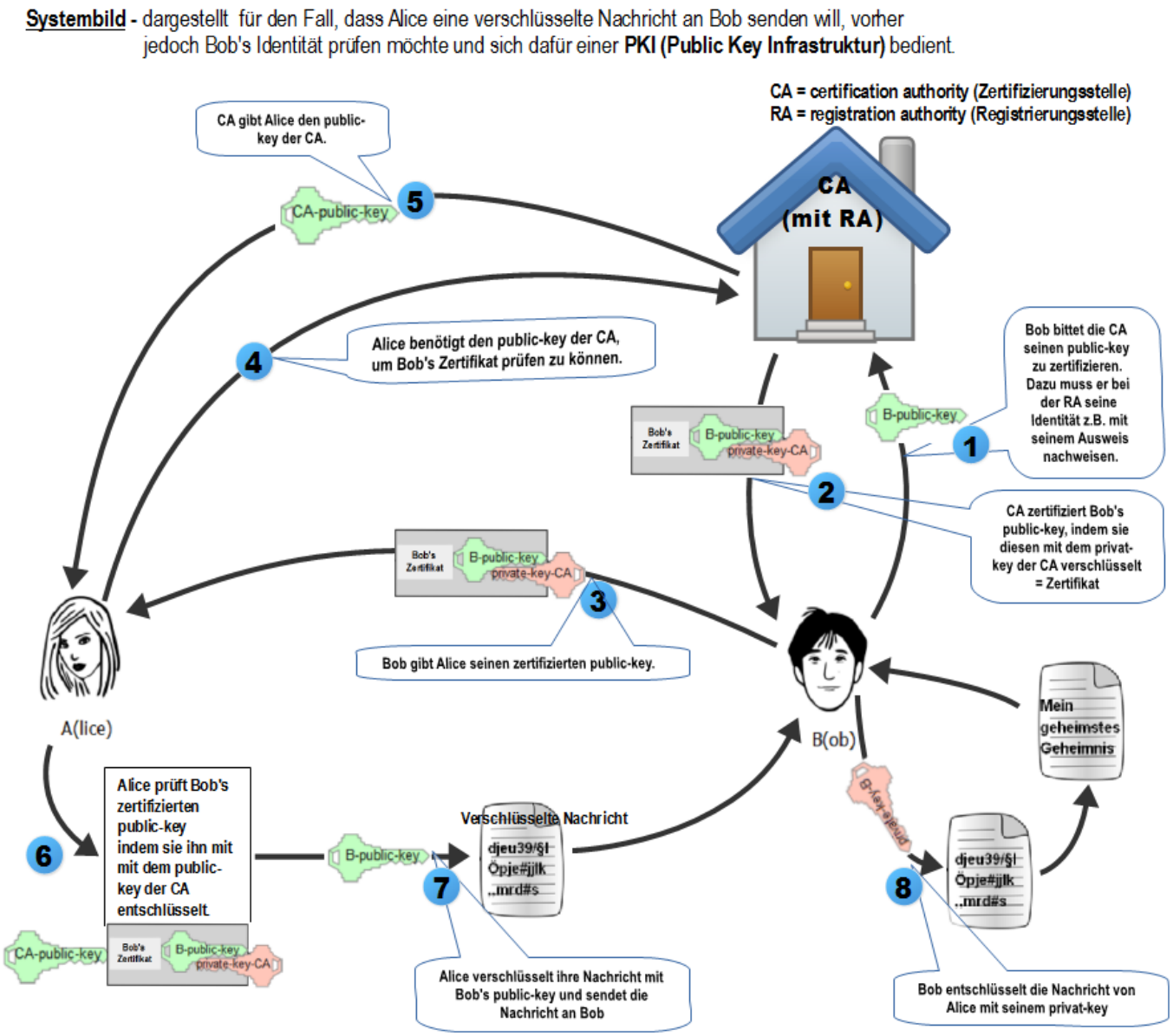


Nur, wer garantiert, dass der öffentliche Schlüssel der Zertifizierungsstelle tatsächlich zur Zertifizierungsstelle gehört? Das macht man natürlich wieder mit Zertifikaten.



Es ergibt sich eine ganze Hierarchie von Zertifizierungsstellen, die sich schrittweise Zertifikate ausstellen. Damit die Kette irgendwann zu einem Ende kommt, gibt es oberste Zertifizierungsstellen, die sogenannte Wurzelzertifikate ausstellen. Das sind Zertifikate, bei denen die Zertifizierungsstelle sich selbst das Vertrauen ausstellt.

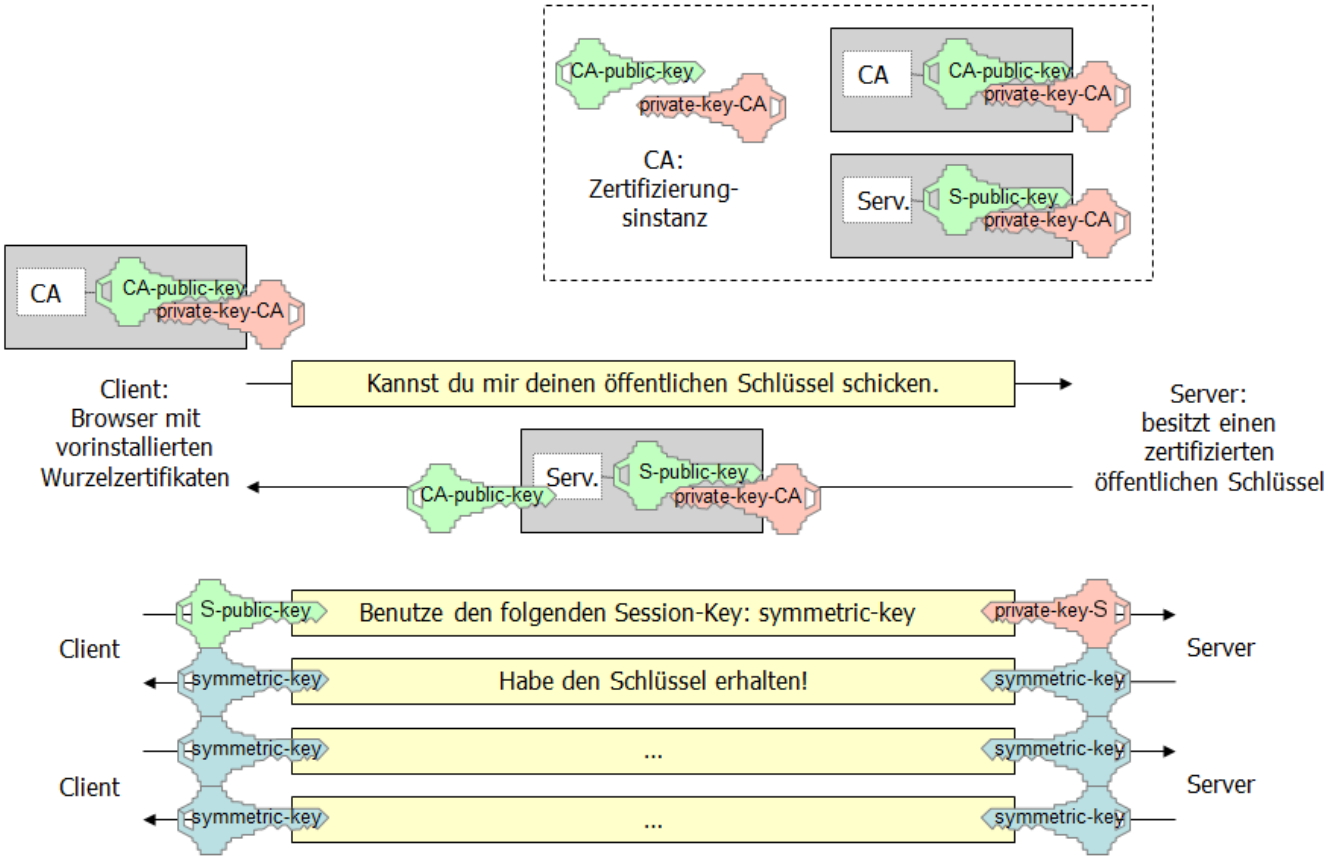
In der folgenden Abbildung wird das Zusammenspiel aller Beteiligten nochmal verdeutlicht.



Übertragung von Webseiten

HTTPS (Abkürzung für HyperText Transfer Protocol Secure) ist eine Protokoll zur sicheren Übertragung von Webseiten im Internet.

Die folgende Abbildung zeigt (in vereinfachter Form), wie Daten bei diesem Protokoll übertragen werden.



Beim Verbindungsaufbau sendet der Server ein Server-Zertifikat (mit seinem öffentlichen Schlüssel) an den Client (Browser).

Der Browser überprüft das übermittelte Server-Zertifikat - in der Regel mit einem vorinstallierten Wurzelzertifikat. Der Browser hat also vorinstalliertes Vertrauen in bestimmte Zertifizierungsstellen.

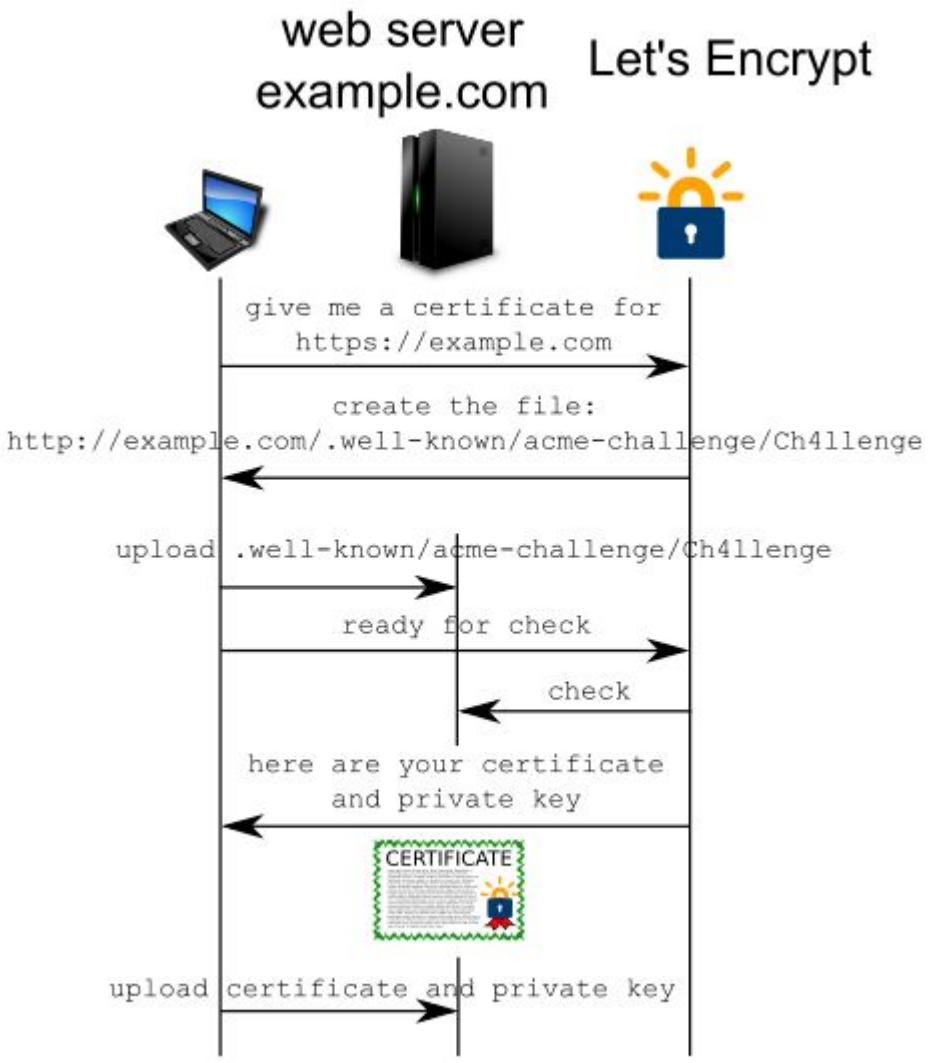
Wenn die Überprüfung zu einem positiven Ergebnis gekommen ist, dann erzeugt der Browser einen Session-Key und übermittelt ihn - natürlich verschlüsselt - an den Server. Nachdem der Server den Erhalt bestätigt hat, kann der sichere Nachrichtenaustausch beginnen.

Let's encrypt

Für die Verschlüsselung des Datenverkehrs zwischen Webbrowser und Webserver werden normalerweise teure Zertifikate benötigt. Seit Ende 2015 gibt es mit Let's Encrypt ein Projekt, das kostenlose Zertifikate vergibt. Während des Zertifizierungsprozesses muss lediglich nachgewiesen werden, dass man Zugriff auf den Webserver hat und dort zum Beispiel Dateien speichern darf. Ein

kleiner Nachteil dieses vereinfachten Zertifizierungsprozesses ist, dass unklar bleibt, welche Person oder welches Unternehmen das Zertifikat beantragt hat und wer hinter einer Webseite steckt. Trotzdem ist es aus meiner Sicht sinnvoll den Datenverkehr überhaupt erst einmal zu verschlüsseln, so dass der Datenverkehr nicht so leicht abgehört oder manipuliert werden kann. Außerdem bewertet die Suchmaschine Google eine Webseite mit HTTPS in seinen Trefferlisten etwas besser - als Anreiz zum Umstieg. Eine verschlüsselte Verbindung wird im Webbrowser Firefox links neben der Adressleiste mit einem grünen Schloss symbolisiert.

Während des Zertifizierungsprozesses müssen auf dem Webserver Dateien als „Challenge“ hochgeladen werden, um nachzuweisen, dass das Zertifikat zum Webserver passt.



Let's Encrypt ist eine Zertifizierungsstelle welche durch Spenden finanziert wird. Ziel des Projekts ist es, das World Wide Web sicher zu machen. Um die Privatsphäre der User zu wahren, sollen verschlüsselte Verbindungen zum Normalfall werden - und das so einfach wie nur möglich.

Passwörter

220117_bsi_onlineschutz_password_final_v4.mp4

Was Sie sich immer schon gefragt haben:

- Was sind gute Passwörter?
- Wie merke ich mir meine Passwörter?
- Wie werden Passwörter sicher auf Computern gespeichert?
- Kann die Systemadministratorin eigentlich mein Passwort auslesen?
- Sind Fingerabdruckscanner besser als Passwörter?
- Warum sind die Passwörter beim Online-Banking eigentlich so lächerlich kurz und einfach?
- Was versteht man unter 2-Faktor-Authentifizierung?

Auf diese Fragen soll in dem folgenden Kapitel eingegangen werden.

Empfehlungen

Als Empfehlung für gute Passwörter liest man häufig, dass ein gutes Passwort

- mindestens 8 Zeichen lang sein sollte.
- aus Groß- und Kleinbuchstaben, Sonderzeichen und Ziffern bestehen sollte.
- nach einem möglichst komplexen Muster gebildet werden sollte.

Ein gutes Passwort wäre somit zum Beispiel:

f8$)?,G3

Liest man dann weiter, so folgen meist die Empfehlungen:

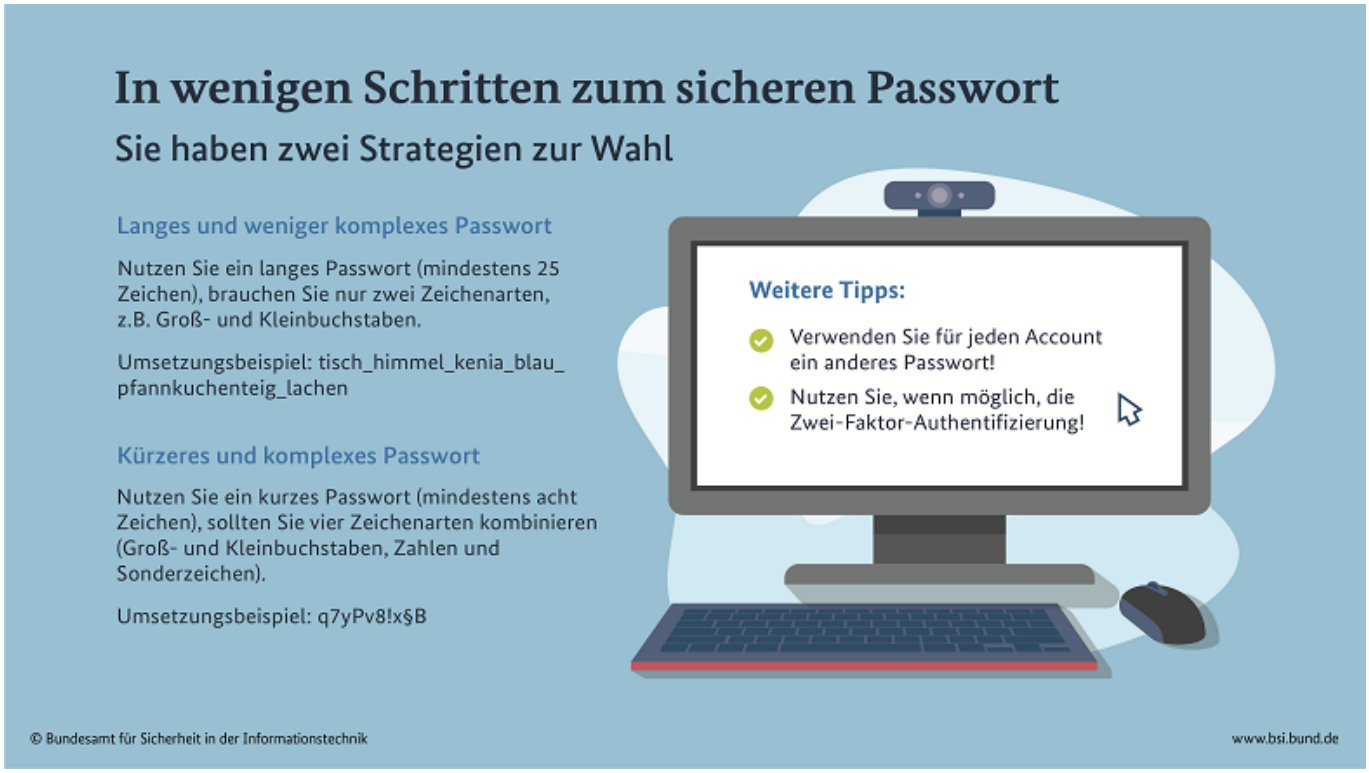
- Niemals für verschiedene Dienste das gleiche Passwort verwenden.
- Passwörter nicht auf Zetteln oder in Dateien auf einem Computer notieren.
- Passwörter unbedingt in regelmäßigen Abständen ändern.

Ein typischer Computernutzer kommt heute wahrscheinlich leicht auf 10-20 oder sogar noch deutlich mehr Passwörter, die er oder sie sich auf diese Weise merken müsste. Damit ist die Umsetzung der obigen Forderungen für die meisten Nutzer völlig unrealistisch.

Als Konsequenz werden dann meist:

- Viel zu einfache Passwörter verwendet, in denen dann oft auch noch leicht zu erratende persönliche Daten wie Namen oder Geburtstage vorkommen.
- Das gleiche Passwort mehrmals bei verschiedenen Diensten verwendet.
- Passwörter nie oder viel zu selten geändert.
- Passwörter in einfachen Textdateien auf unsicheren Computern gespeichert.

Ein einfacher Ausweg aus dem beschriebenen Dilemma ist es, sogenannte Passphrasen statt Passwörter zu verwenden. Im Folgenden sollen nun solche Passphrasen und deren Eigenschaften genauer betrachtet werden.



Microsoft Kennwortrichtlinien

Kennwortrichtlinien für Administratoren

Das primäre Ziel eines sichereren Kennwortsystems ist die Vielfalt der Kennwörter. Sie möchten, dass Ihre Kennwortrichtlinie viele verschiedene und schwer zu erratende Kennwörter umfasst. Hier sind ein paar Empfehlungen, wie Sie Ihre Organisation am besten schützen können.

- Beibehalten einer Mindestlänge von vierzehn Zeichen
- Legen Sie keine Anforderungen an die Zeichenzusammensetzung fest. Beispiel: *&(^%$
- Fordern Sie kein obligatorisches periodisches Zurücksetzen der Kennwörter für Benutzerkonten.
- Verbieten Sie gängige Kennwörter, um die anfälligsten Kennwörter von Ihrem System fernzuhalten.
- Schulen Sie Ihre Benutzer, ihre organization Kennwörter nicht für nicht arbeitsbezogene Zwecke wiederzuverwenden.
- Erzwingen Sie die Registrierung für die mehrstufige Authentifizierung
- Aktivieren von Risikobasierten Multi-Factor Authentication-Herausforderungen

Kennwortratgeber für Benutzer

Hier finden Sie einige Kennwortanleitungen für Benutzer in Ihrer Organisation. Informieren Sie Ihre Benutzer über diese Empfehlungen und setzen Sie die empfohlenen Kennwortrichtlinien auf Organisationsebene durch.

- Verwenden Sie kein Kennwort, das einem auf anderen Websites verwendeten Kennwort entspricht oder ähnlich ist.
- Verwenden Sie kein einzelnes Wort wie beispielsweise Kennwort, und keinen häufig

verwendeten Ausdruck, z. B. Ichliebedich
- Machen Sie Kennwörter schwer zu erraten, auch von Personen, die viel über Sie wissen, z. B. die Namen und Geburtstage Ihrer Freunde und Familie, Ihre Lieblingsbands und Ausdrücke, die Sie verwenden möchten

Aufgabe 1)

Wie viele mögliche Passwörter (PIN 4-stellig) gibt es bei der EC-Karte?

a) wenn man sich an gar nichts mehr erinnern kann?

Man muss immer Anzahl der Möglichkeiten pro Stelle betrachten (Variation mit Zurücklegen).

$10*10*10*10 = 10^4 = 10000$

b) wenn man noch weiß, dass unter den richtigen Ziffern genau eine 3 war?

Da die Ziffer 3 nur genau einmal vorkommt, gibt es für die verbleibenden Stellen noch 9 (statt vorher 10) mögliche Ziffern (in jeweils 4 Anordnungen), somit gilt:

4 * 9^3 = 2.916.

c) wenn man noch weiß, dass diese eine 3 definitiv an der dritten Stelle stand?

Da sicher ist, dass die 3 an der dritten Stelle stand, sucht man hier nun die Anzahl der möglichen Ziffernkombinationen für 3 aus 10 (eine Stelle ist ja bekannt und fällt somit komplett weg) im Modell der Variation ohne Zurücklegen.

Es ergibt sich: 9^3 = 729.

Wie versuchen Banken, dennoch eine gewisse Sicherheit zu garantieren?

Nach 4 falschen Versuchen wird zumeist die Karte eine Zeit lang (z.B. 2 Tage) gesperrt.

Aufgabe 2)

Wie viele achtstellige Passwörter gibt es über dem Alphabet der 95 sinnvoll über die Tastatur einzugebenden ASCII-Zeichen? Wie lange würde ein Brute-Force-Angriff im sogenannten Worst-Case dauern, wenn ein Angreifer Hundert Milliarden (100 000 000 000) Passwörter pro Sekunde testen kann?

95^8 = 6 634 204 312 890 625 / 100 000 000 000 = 66 342,04312890625s = 1105,7 min = 18,43 h

==⇒ https://www.passwortcheck.ch

Passphrasen

Unter der Annahme, dass ein Angreifer 100 Milliarden Passwörter pro Sekunde testen kann, dauert ein Brute-Force-Angriff auf ein 8-stelliges Passwort über dem Alphabet aller 95 ASCII-Zeichen im Worst-Case gerade einmal

$95^8 / 10^{11}$ s = 18,4 h.

Ein entsprechender Angriff auf ein 15-stelliges Passwort über dem Alphabet {a, ..., z} würde

$26^{15} / 10^{11}$ s = 194 127 Tage ~ 531 Jahre

dauern.

Damit sind Passwörter, die aus

- mindestens 8 Zeichen bestehen,
- Groß- und Kleinbuchstaben, alle möglichen Sonderzeichen und Ziffern beinhalten
- und nach einem möglichst zufälligen Muster gebildet werden,

für einen Menschen schwer zu memorieren und für einen Angreifer leicht zu dechiffrieren.

Im Vergleich dazu sind etwas längere Passwörter, selbst wenn sie lediglich aus den Kleinbuchstaben {a, ..., z} gebildet werden, für einen Menschen deutlich leichter zu memorieren, und für einen Angreifer sehr viel schwer zu dechiffrieren.

Relativ lange Passwörter, die aber dennoch gut für memorieren sind, werden oft als sogenannte Passphrasen bezeichnet.

Beim Erstellen solcher Passphrasen sind der Phantasie keine Grenzen gesetzt, so dass eine mögliche Passphrase zum Beispiel sein könnte:

```
nadaborder4fanta=hinreichend4heitERsicht
```

Wird gefordert, dass eine Passphrase unbedingt Sonderzeiten und Großbuchstaben enthalten muss, so lassen sich diese meist leicht in eine beabsichtigte Passphrase einbauen.

Noch besser ist natürlich, solche Passphrasen zufällig zu generieren, etwa mittels des sogenannten Diceware-Verfahrens.

Aufgabe 1)

Recherchiere im Internet, was man unter dem sogenannten Diceware-Verfahren versteht. Erzeuge mittels des Diceware-Verfahrens eine Passphrase, die aus mindestens 6 Diceware-Wörtern besteht. Verwende einen herkömmlichen Spielwürfel, um echte Zufallszahlen zu generieren.

Aufgabe 2)

Die folgende Passphrase wurde zufällig mit dem Diceware-Verfahren erzeugt:

tintenfleck-biergarten-hinauf-selber-backt-verarbeiten-parkbank-fragst-bringen-gerahmt

Beurteile die Sicherheit dieser Passphrase.

Wortliste-Deutsch

Entropie von Passwörtern

Mit der sogenannten Passwort-Entropie wird versucht, eine Aussage darüber zu treffen, wie gut oder schlecht es durch einen Angreifer vorhersagbar ist.

Die Entropie eines Passworts wird dabei von den verwendeten Zeichen bestimmt. Sie kann zum Beispiel durch die Verwendung von Groß- und Kleinbuchstaben, Ziffern sowie Sonderzeichen erweitert werden. Dazu kommt die Länge des Passworts, die eine sehr große Rolle spielt. Die Entropie eines Passworts besagt, wie schwierig es ist, es durch Erraten, Brute Force, einen Wörterbuchangriff oder andere Methoden zu knacken.

Die Entropie von Passwörtern wird meist in Form von Bits angegeben. Ein bereits bekanntes Passwort hat beispielsweise eine Entropie von 0 Bit. Ein Passwort, das in der Hälfte der Fälle direkt vorausgesagt werden kann, hat eine Entropie von 1 Bit. Die Entropie eines Passwortes kann berechnet werden, indem die Entropie für jedes verwendete Zeichen bestimmt wird. Für ein Kennwort mit einer Entropie von zum Beispiel 30 Bit werden 2 hoch 30 Versuche benötigt, um alle Möglichkeiten durchzurechnen.

Beispiel:

Die Anzahl an 20-stelligen Passphrasen über dem Alphabet A={a,b,c,...,z} der Kleinbuchstaben beträgt:

$(\#A)^{20} = 26^{20}$

Sei nun e die Länge der Bitfolge, mit der die gleiche Anzahl an Passphrasen dargestellt werden kann, also:

$2^{e} = 26^{20}$

$e = \log_2(26^{20}) = 20\ \log_2(26) = 20\ \log(26)/\log(2) = 94$

Damit können durch Bitfolgen der Länge 94 Bit genauso viele Passphrasen dargestellt werden wie durch 20 Stellen über dem Alphabet A der Kleinbuchstaben. Man sagt, eine 20-stellige Passphrase über dem Alphabet A der Kleinbuchstaben enthält 94 Bit Entropie.

Aufgabe 1)

Wie viel Bit Entropie enthält ein 8-stelliges Zufallswort über

a) dem Alphabet A={a,b,c,...,z} der Kleinbuchstaben?

b) dem Alphabet der ASCII-Zeichen?

Aufgabe 2)

Wie lang muss ein Passwort über dem Alphabet A={a,b,c,...,z} der Kleinbuchstaben gewählt werden, damit es genau so viel Entropie enthält wie

a) ein 8-stelliges Passwort über den Alphabet der ASCII-Zeichen?

b) ein 10-stelliges Passwort über den Alphabet der ASCII-Zeichen?

c) ein n-stelliges Passwort über den Alphabet der ASCII-Zeichen?

Aufgabe 3)

Unter der Voraussetzung, dass eine Angreiferin Kenntnis darüber hat, dass das Diceware-Verfahren verwendet wurde (und sogar auch noch die verwendete Wortliste kennt): Wie viel Bit Entropie enthält ein Zufallswort, das nach dem Diceware-Verfahren generiert worden ist? Welche Entropie ergibt sich daraus für eine Passphrase, die aus 8 Zufallswörtern gebildet wurde?

Aufgabe 4)

Wie viele Passwörter müssen mittels des Diceware-Verfahrens zu einer Passphrase zusammengesetzt werden, die mindestens 128 Bit Entropie enthalten soll?

Zufallspasswörter und Passwortmanager

Viele Menschen benötigen heute leicht 20-30 oder sogar noch mehr Passwörter, so dass es auch bei der Verwendung des Diceware-Verfahrens oft nicht mehr gelingt, sich eine so große Anzahl an Passwörtern auswendig zu merken.

Daher ist es oft nützlich, einen sogenannte Passwortmanager zu verwenden, mit dessen Hilfe man auch eine große Anzahl an Passwörtern noch effizient verwalten kann, ohne sie alle auswendig lernen zu müssen. Die einzelnen Passwörter werden dabei durch ein sogenanntes Hauptpasswort geschützt verschlüsselt auf der Festplatte gespeichert. Typischerweise kann man sich Passwörter auch gleich automatisiert generieren lassen und diese bei Bedarf mittels eines ausgeklügelten Cut-and-Paste-Systems in Formularfelder übertragen, ohne sie mühsam in Handarbeit abtippen zu müssen.

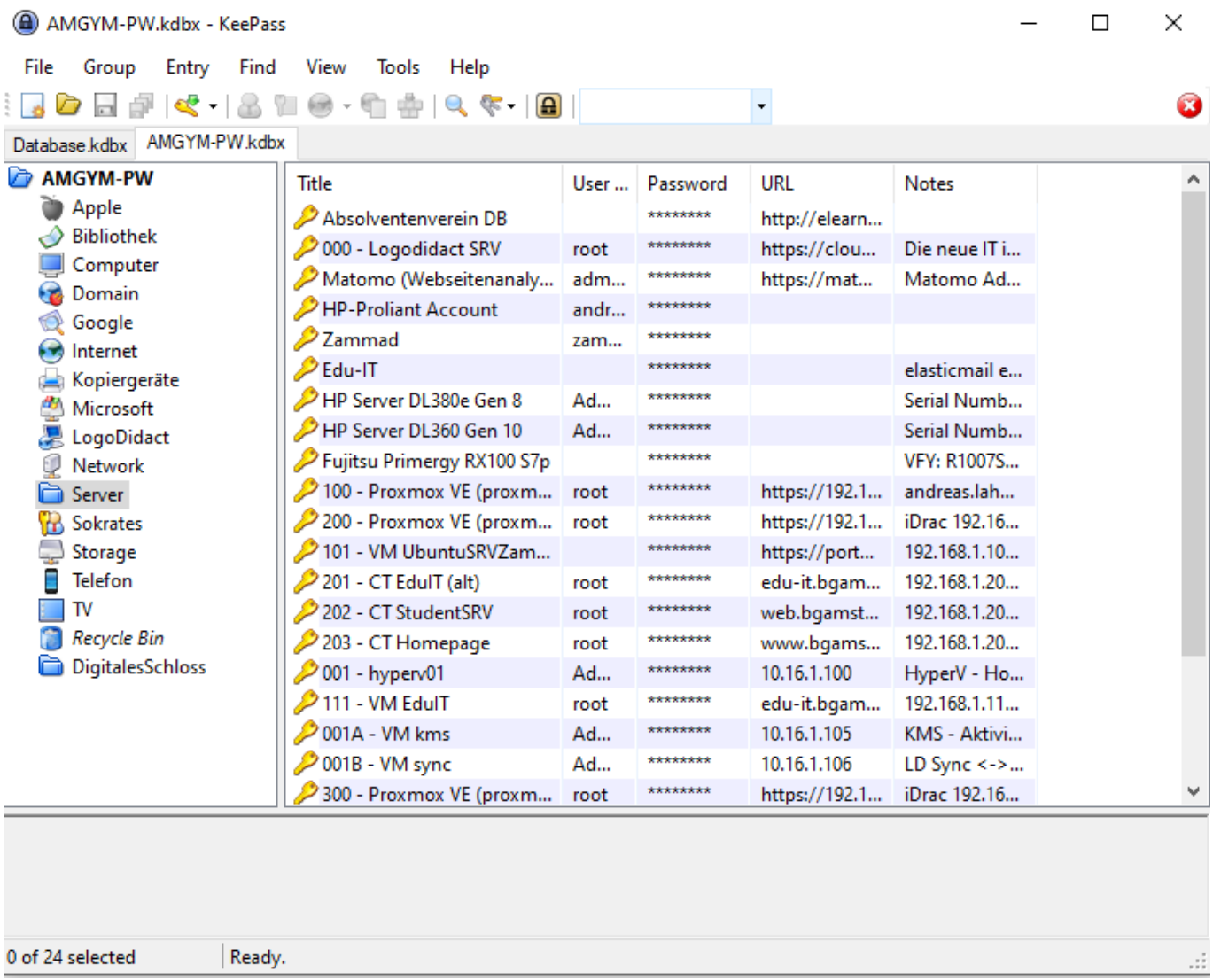
Ein weit verbreiteter Passwort-Safe ist die freie Software KeePass, die unter einer freien GPL-Lizenz für alle gängigen Betriebssysteme verfügbar ist.

Typischer Funktionsumfang eines guten Passwort-Safes in Stichworten:

- Hauptpasswort
- generieren von Pseudozufallspasswörtern
- Speichermöglichkeit von benötigten URLs und Kommentaren
- automatisches Übertragen von Passwörtern in Formularfelder per Tastenkombination
- Speicherung der Versionsgeschichte bei Passwortänderungen
- Beachtung des Kerckhoffs'schen Prinzips durch Veröffentlichung als freie Software

Bei der Auswahl eines konkreten Passwort-Safes sollte man unbedingt darauf achten, dass dieser unter einer freien Softwarelizenz veröffentlicht ist. Denn nur in diesem Fall ist sichergestellt, dass unabhängige Sicherheitsfachleute langfristig die Möglichkeit haben, Funktion und Vertrauenswürdigkeit der Software überprüfen und kontrollieren zu können. Aber Achtung: Die Möglichkeit zu einer solchen Kontrolle bedeutet noch nicht, dass diese auch in allen Fällen immer durchgeführt wurde. Man kann sich aber an den Empfehlungen von renommierten Expert:innen orientieren.

Die Sicherheit der gespeicherten Passwörtern hängt natürlich entscheidend von der Sicherheit des Hauptpasswortes ab. Deshalb sollte man als Hauptpasswort eine hinreichend lange Passphrase wählen, welche man zum Beispiel mittels des Diceware-Verfahrens generieren kann.



Speicherung von Passwort-Dateien

In Betriebssystemen oder bei Diensten im Internet müssen in irgendeiner Form Informationen über die Passwörter der angemeldeten Nutzer:innen gespeichert werden. Ein erster Ansatz dazu könnte zunächst sein, Passwörter einfach im Klartext in einer einfachen Textdatei zu speichern:

Beispiel für eine Klartext-Passwortdatei passwd.txt:

```
mustermann:qwertz
falken:joshua
administrator:passw123
root:fmvku76%d
edward:jdjhfZhd8/6ei:dk*askd_ddk?(djsh%hdZH5:dk
```

Die Benutzer:in mit dem Login „mustermann“ hat also das Passwort „qwertz“ usw.

Der Nachteil einer solchen Speicherung im Klartext ist natürlich, dass jede, die Lesezugriff auf die Passwortdatei bekommt, sofort die Passwörter von allen Nutzer:innen im Klartext auslesen kann.

Deshalb sollten von verantwortungsbewussten Systemadministrator:innen Passwörter von Nutzer:innen niemals im Klartext in Passwortdateien abgespeichert werden.

Speicherung als Hashwerte

Passwörter lassen sich auch so abspeichern, dass sie nicht im Klartext ausgelesen werden können. Dazu wird eine kryptographische Einwegfunktion, eine sogenannte Hashfunktion, verwendet. Mittels einer solchen Hashfunktion lassen sich Klartextpasswörter leicht und effizient zu Hashwerten verrechnen. Umgekehrt ist es aber praktisch unmöglich, aus einem vorgegebenen Hashwert das Klartextpasswort wieder effizient zurückzuberechnen. Da der Hashwert eines Passwortes praktisch einmalig sind, wird er anschaulich oft auch als dessen Fingerabdruck bezeichnet.

Das sogenannte SHA1-Hashverfahren ordnet einem beliebig langen Text einen 20 Byte langen hexadezimalen Hashwert zu. Für das Passwort des Benutzers „mustermann“ aus dem obigen Beispiel ergibt sich damit der folgende Hashwert:

```
sha1('qwertz') = 8c829ee6a1ac6ffdbcf8bc0ad72b73795fff34e8
```

Damit bekommt die gesamte Passwortdatei aus dem obigen Beispiel die folgende Form:

```
mustermann:8c829ee6a1ac6ffdbcf8bc0ad72b73795fff34e8
falken:d6955d9721560531274cb8f50ff595a9bd39d66f
administrator:4de730479c592c0619802013bc9883dfbde67fea
root:277c21563ade912ffa1f183f04ed482648790fed
edward:712778715c24c68886ecf69d7191cc2acec05919
```

Enthält eine Passwortdatei oder eine Datenbank statt der Klartext-Passwörter nur ihre Hashwerte, so kann eine potentielle Angreiferin mit diesen Hashwerten erst einmal nichts anfangen, weil sich

Hashwerte nicht zurückrechnen lassen. Das gleiche gilt auch für die Systemadministrator:in des Computersystems, die Passwörter nicht im Klartext auslesen kann, obwohl sie natürlich Zugriff auf die Passwortdatei hat. Beachte an dieser Stelle jedoch, dass die Nutzer:in beim späteren Einloggen nicht etwa den Hashwert, sondern ihr Klartext-Passwort in das Passwort-Eingabefeld eingibt. Anschließend wird der Hashwert des eingegebenen Passworts berechnet. Nur wenn dieser berechnete Hashwert mit dem gespeicherten übereinstimmt, wird der Zugang gewährt.

Anmerkung 1

Berechnung von SHA1-Hashwerten in einer GNU/Linux-Shell:

```
echo -n 'qwertz' | sha1sum
```

Berechnung von SHA1-Hashwerten in Python:

```
import hashlib
hashlib.sha1("qwertz".encode('utf-8')).hexdigest()
```

Anmerkung 2

Aufgrund der Entwicklung immer schnellerer Rechner entspricht das SHA1-Verfahren nicht mehr den heutigen Anforderungen an ein sicheres Hashverfahren. Es wurde im obigen Beispiel lediglich zur Illustration der prinzipiellen Funktionsweise von Passwortdateien verwendet. Aktuelle Verfahren wie SHA512 erzeugen sehr viel längeren Hashwerte, so dass die Übersichtlichkeit der obigen Beispiel-Passwortdatei (für eine menschlichen Leserin) damit erschwert wäre.

Pepper

Werden gehashte Passwörter nach dem oben beschriebenen einfachen Hashverfahren gespeichert, so ergibt sich allerdings das Problem, dass sich die Hashwerte typischer Passwörter vorherberechnen und in sogenannten Rainbowtables speichern lassen. Wird dann zu einem späteren Zeitpunkt eine gehashte Passwortdatei erbeutet, so können die dort gespeicherten Hashwerte einfach mit den vorherberechneten verglichen werden, ohne dass dazu eine erneute zeitaufwendige Berechnung nötig wäre. Findet der Angreifer eine Übereinstimmung von Hashwerten, so hat er damit auch das zugehörige Klartextpasswort gefunden.

Um solche Angriffe auf Passwortdateien mittels Rainbowtables zu erschweren, wird ein sogenanntes Pepper eingesetzt. Dazu werden die Klartextpasswörter pw um ein hinreichend lange Zeichenfolge, das sogenannte Pepper p, ergänzt. Anschließend wird auf die konkatenierte Zeichenfolge p+pw ein Hashverfahren wie z.B. SHA1 angewandt.

```
    pw = 'qwertz'
    p    = '9cm9hsgkdcucz7ckdje-z7652cv_mnbsdsj_'
    sha1(p+pw) = sha1('9cm9hsgkdcucz7ckdje-z7652cv_mnbsdsj_qwertz')
               = 5f684e914ba9840cd0a0b2af79251c476ec5ffc2
```

Mit dem Pepper p = '9cm9hsgkdcucz7ckdje-z7652cv_mnbsdsj_' bekommt die gesamte Passwortdatei aus dem obigen Beispiel damit die folgende Form. Beachte, dass für alle Passwörter das gleiche Pepper p verwendet wird.

```
    mustermann:5f684e914ba9840cd0a0b2af79251c476ec5ffc2
    falken:08eb33988697de0ad395ba94290e2a324887d0ab
    administrator:a1becea837841197fb847940653c66b84859a576
    root:162b0d49d63b2b280e313ec7ae012ce7871b578a
    edward:6d47b7c8a88bc2e633d05d8defb0b76405859044
```

Wird nun die Passwortdatei durch einen Angreifer erbeutet, so ist für diesen nicht erkennbar, dass ein Pepper verwendet wurde. Aufgrund der hohen Entropie der um das Pepper verlängerten Klartextpasswörter funktioniert auch ein Angriff über vorherberechnete Rainbowtables nicht mehr.

Eine Schwachstelle des Hashen von Passwörtern mit Pepper entsteht aber, sobald ein Angreifer zusätzlich zur Passwortdatei auch noch das Pepper p erbeutet. Denn dann kann er z.B. Wörterbuch- oder Bruteforceangriffe starten, indem er die zu testenden Passwörter einfach vor dem Hashen um das Pepper ergänzt und damit einen Rainbowtable für das erbeutete Pepper berechnet.

Ein Ausweg daraus bietet wiederum das Hashen der Passwörter mit einem sogenannten Salt:

Salt

Während beim Hashen von Passwörtern mit einem Pepper für die gesamte Passwortdatei ein festes Pepper verwendet wird, wird für Passwortdateien mit Salt für jedes Passwort ein unterschiedlicher sogenanntes Salt s verwendet.

```
    pw = 'qwertz'
    s    = 'dkkfjerfjc983883(7akfrjklfds8/akdjf=adfsh*'
    sha1(s+pw) = sha1('dkkfjerfjc983883(7akfrjklfds8/akdjf=adfsh*qwertz')
               = b1abab1a8efe6bd81df13f6f2fa3ae2aadce960d
```

Die Saltwerte werden dann jeweils zusammen mit dem Hashwert in der Passwortdatei gespeichert, so dass sich für das obige Beispiel dann eine Datei der folgenden Form ergibt:

```
mustermann:dkkfjerfjc983883(7akfrjklfds8/akdjf=adfsh*:b1abab1a8efe6bd81df13f
6f2fa3ae2aadce960d
falken:dk38d7/%&jdjdsjd9djsu;(djja8nvmnbx:a53fba661a4e031a7de12bfc55a422d8fb
fad6e2
administrator:aslkfa8e4jfsaksakfdjlalkdsfjdsa:286752d79187249ffa28d2068e460c
2c12a8f093
root:ivcjcxy8737dki9cyyxvjlkj28347askf:2e4fc670daa272fac4bf7d9c67fbdadddf558
78a
edward:aslkfj8vcx7j38h2983hfnc832fjc:55f2ee3ce9296cb9fbaa49581c998c26cbb0c6c
e
```

Da für alle Passwörter unterschiedliche Saltwerte verwendet werden, müsste bei einem Rainbowtable-Angriff für jedes Passwort ein eigener kompletter Rainbowtable berechnet werden. Dies ist so aufwendig, dass ein solcher Angriff nicht effizient möglich ist.

Zwei-Faktor-Authentifizierung (2FA)

Die Zwei-Faktor-Authentifizierung (auch: Zwei-Schritte- oder Zwei-Wege-Authentifizierung bzw. 2FA) ist eine zusätzliche Sicherheitsmaßnahme zum Schutz von Benutzerkonten: Selbst wenn Ihre Passwörter in die falschen Hände gelangen, haben Unbefugte auf diese Weise keinen Zugriff auf Ihre Accounts.

Zusätzlich zum Passwort müssen Sie beim Login eine weitere Sicherheitskomponente eingeben, etwa einen PIN-Code. Dieser Code wird z. B. an die in Ihrem Konto hinterlegte Handynummer gesendet.

Eine weitere Möglichkeit sind für kurze Zeit gültige Einmalkennwörter, die von sogenannten Authentifizierungs-Apps erstellt werden. Solche Apps bieten den Vorteil, dass Sie damit Ihre mit 2FA geschützten Konten übersichtlich verwalten können.

Bekannte Apps für die Zwei-Schritte-Authentifizierung sind z. B. Google Authenticator (für iOS/Android), Microsoft Authenticator (für iOS/Android) oder Twilio Authy (für iOS/Android/Desktop).

Die Zwei-Faktor-Authentifizierung wird inzwischen von den meisten großen Online-Diensten unterstützt – von Apple über Google oder Microsoft bis hin zu Twitter, Facebook und Instagram.

bsifb_animation_zwei-faktor-authentisierung.mp4

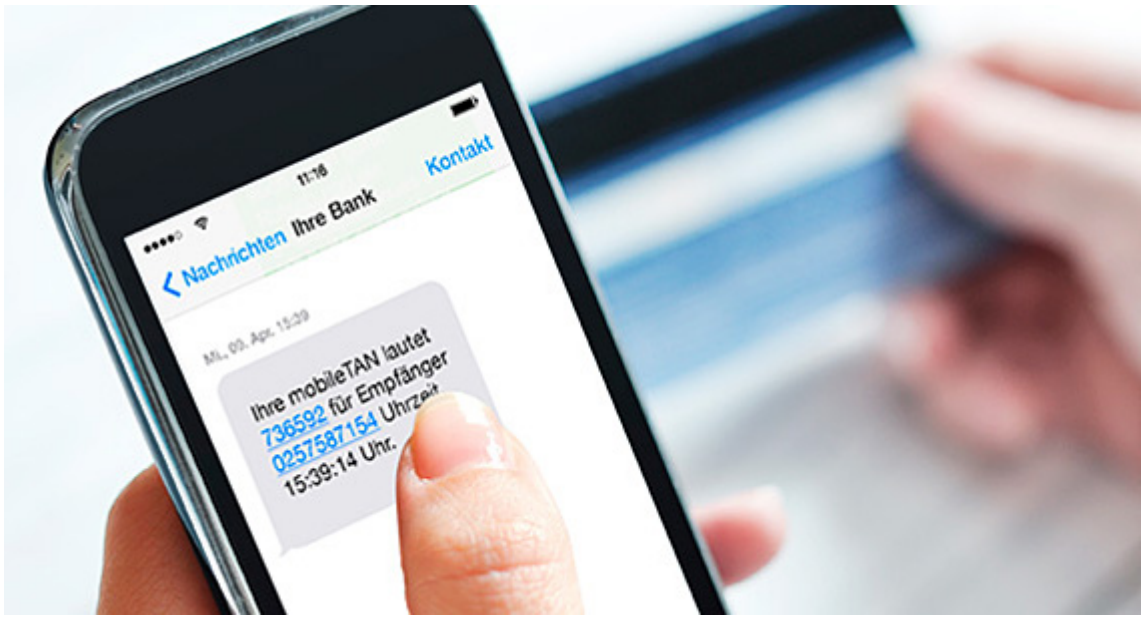
Grundsätzlich gibt es drei unterschiedliche Möglichkeiten, um sich auszuweisen:

- durch Wissen (z. B. Passwort)
- durch Besitz (z. B. Bankomatkarte, Token)
- durch biometrische Merkmale (z. B. Fingerabdruck)

„Token“ ist ein Überbegriff für jegliche Technologien, mit deren Hilfe sich eine Person authentisieren kann. Hierzu zählen sowohl Hardware-Tokens als auch Software-Tokens. Hardware-Tokens sind beispielsweise Smartcards, USB-Tokens und RFID-Chips, mit denen Einmalpasswörter generiert werden, oder auch Smartphones, an die ein Authentifikationsfaktor übertragen wird. Software-Tokens benötigen keine zusätzliche Hardware – hier werden via Software One-Time-Passwords (OTPs) generiert. Solche Einmalpasswörter sind Passwörter, die ein einziges Mal verwendet werden können und danach ihre Gültigkeit verlieren.

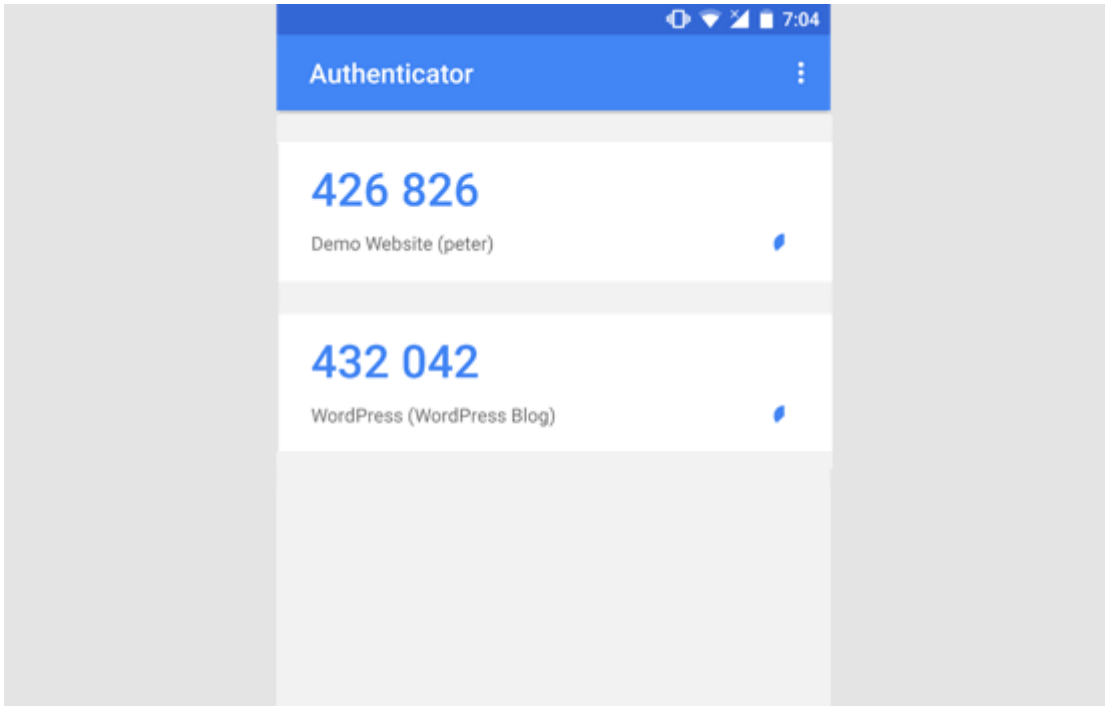
Eine bekannte Zwei-Faktor-Authentifizierung ist das Zusammenspiel von **Bankomatkarte** (der Besitz) und die dazugehörige **PIN** (das Wissen). Um Bargeld beheben zu können, werden beide Bestandteile benötigt. Das Gleiche gilt für Online-Überweisungen, bei denen die Anmeldedaten (das Wissen) und – zur Freigabe einer Überweisung – die TAN (der Besitz) vorliegen müssen. Früher funktionierte dies mit einer im Voraus generierten **TAN-Liste**, welche von der Bank postalisch der Kundin oder dem Kunden zugestellt wurde. Dabei war jede TAN ein einziges Mal gültig. Heutzutage sind die meisten Banken von Papier-TANs auf ihre digitalen Pendants umgestiegen: per SMS (je nach Bank wird dieses Verfahren unterschiedlich benannt: **smsTAN**, **mobileTAN**, **TAC-SMS**), TAN-Generator oder digitaler Signatur. Ein TAN-Generator ist ein selbstständiges Gerät, in das eine EC-Karte gesteckt wird und nach der Eingabe eines PIN-Codes eine TAN generiert wird. So ein TAN-Generator hat durch die zusätzliche Einbeziehung der EC-Karte und der fehlenden Vernetzung eine hohe Sicherheit. Nichtsdestotrotz sind smsTAN wegen ihrer einfachen Bedienung und schnellen Verfügbarkeit bei Kundinnen und Kunden am beliebtesten, da kein weiteres Gerät – außer dem in der Regel bereits vorhandenen Mobiltelefon – benötigt wird. So eine smsTAN hat einen

Gültigkeitszeitraum von typischerweise nur einigen wenigen Minuten, was einen Missbrauch stark einschränkt.



Die Verwendung eines Smartphones als Empfänger für OTPs wird auch als tokenlose Authentifizierung bezeichnet, da kein extra Hardware-Token von Anbietern zur Verfügung gestellt werden muss, sondern das Smartphone der Kundin oder des Kunden als Empfangs- oder Anzeigegerät für Tokens dient.

Die tokenlose Authentifizierung etabliert sich mittlerweile auch außerhalb der Bankenbranche und unternehmensinternen Systemen. Google unterstützt beispielsweise eine Zwei-Faktor-Authentifizierung als Anmeldeverfahren für seine Dienste, bei dem das Smartphone verwendet wird, um OTPs zu empfangen. Dies bedeutet, dass entweder ein SMS-Dienst verwendet wird, der bei einem Anmeldungsversuch in das Google-Konto ein OTP per SMS sendet, oder alternativ dazu eine von Google bereitgestellte App (Google Authenticator) verwendet wird, um gültige OTPs anzuzeigen. Die App-Lösung ist nicht auf eine Mobilfunkverbindung angewiesen und bietet daher mehr Flexibilität.

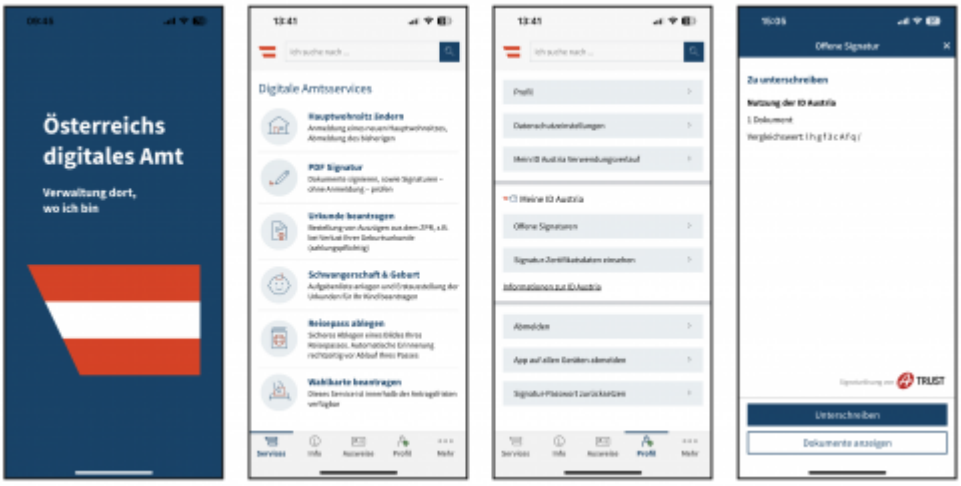


Das System der tokenlosen Authentifizierung mit dem Mobiltelefon erhöht jedoch nur dann die Sicherheit, wenn das Mobiltelefon nicht auch zur Anmeldung auf dem Konto verwendet wird, für welches das OTP angefordert wird. Der Sinn des Tokens ist es einen zweiten, unabhängigen Authentifizierungsfaktor zu liefern, der über einen gesonderten Kanal übertragen oder generiert wird. Falls aber zum Beispiel das Google-Konto über das Smartphone aufgerufen wird und das OTP für die Anmeldung ebenfalls auf das Smartphone gesendet wird, ist die Risikominimierung, die durch Verwendung der Zwei-Faktor-Authentifizierung entstehen sollte, nicht mehr gegeben, da beide Authentifizierungsschritte auf demselben Gerät stattfinden und somit eine mögliche Angriffsstelle für Kriminelle gegeben ist.

Jede Art des Tokens hat seine Vor- und Nachteile, daher ist es wichtig, einen gewissen Grad an Flexibilität zwischen den verschieden Methoden zu erlauben. Der Nachteil bei der Verwendung eines Smartphones ist, dass es zum Zeitpunkt der Authentifikation aufgeladen und verfügbar sein muss, da ansonsten kein OTP zugesendet oder angezeigt werden kann. Zu bedenken ist auch, dass es bei Verlust, Diebstahl oder Defekt des Smartphones nicht möglich sein wird, sich ohne größeren Aufwand in ein durch Zwei-Faktor-Authentifizierung gesichertes Konto einzuloggen.

In Österreich besteht eine weitere Möglichkeit der Zwei-Faktor-Authentifizierung. Im Zuge der Forcierung des e-Governments wurde mit der Bürgerkarte als Token eine elektronische Möglichkeit geschaffen, sich amtlich auszuweisen. Dies ist auch im österreichischen E-Government-Gesetz verankert. Bei der Bürgerkarte fallen keine zusätzlichen Kosten an, da die Funktion der Bürgerkarte auf der bereits verbreiteten „e-card“ lediglich aktiviert werden muss.

Auch an eine tokenlose Authentifizierung wurde gedacht und mit der „Handy-Signatur“ geschaffen. Das Mobiltelefon stellt dabei den Faktor des Besitzes dar und empfängt den Einmalcode (OTP), der fünf Minuten gültig ist. Ein weiterer Vorteil in Österreich: die Handy-Signatur und die Bürgerkarte können für eine rechtsgültige elektronische Unterschrift genutzt werden, diese ist durch das Signaturgesetz der eigenhändigen Unterschrift gleichgestellt.



Fazit

Die Zwei-Faktor-Authentifizierung bietet durch die Kombination mehrerer Authentifizierungsschritte eine höhere Sicherheit für Konten. Jede Nutzerin und jeder Nutzer muss für sich entscheiden, ob der Mehraufwand, der mit der Zwei-Faktor-Authentifizierung verbunden ist, für das jeweilige Konto sinnvoll ist. Falls eine tokenlose Authentifizierung verwendet wird, sollten Nutzerinnen und Nutzer unbedingt darauf achten, dass die OTPs nicht auf das Gerät geschickt werden, welches für die Anmeldung oder Überweisung verwendet wird.

Mögliche Fallstricke

Im Folgenden werden einige Beispiele für mögliche Fallstricke aufgezählt, durch die man schnell wieder die Sicherheit eines eigentlich starken Passwortes gefährden kann. Die Aufzählung ist beispielhaft und erhebt keinen Anspruch auf Vollständigkeit.

Überall gleiches Passwort

Verwendet man dasselbe Passwort mehrfach für unterschiedliche Dienste im Internet, so kann eine Angreifer, der dieses Passwort bei einem dieser Dienste erbeutet, sofort auch auf alle anderen zugreifen. Insbesondere hat man als Benutzer keinerlei Kontrolle darüber, ob ein Diensteanbieter mit den Passwörtern seiner Kunden auch tatsächlich verantwortlich umgeht und die Passwörter beispielsweise ausschließlich verschlüsselt speichert. In der Vergangenheit ist es leider recht häufig vorgekommen, dass Einbrecher in Computersysteme völlig unverschlüsselte Dateien oder Datenbanken mit den Passwörtern einer erschreckend großen Anzahl an Benutzern erbeutet haben. Aus Angst vor Reputationsverlust verschweigen Diensteanbieter solche Einbrüche oft über lange Zeit, so dass die Nutzer des Dienstes nicht einmal bemerken, dass ihr Passwort schon längst kompromittiert worden ist. Hinzu kommt, dass ein verantwortungsvoller und sicherheitsbewusster Umgang mit den Kundenpasswörtern oft sehr viel aufwendiger und kostenintensiver für den Diensteanbieter ist als eine einfache unverschlüsselte Speicherung. Daher sollte man unbedingt für alle Dienste im Internet unterschiedliche Passwörter verwenden. Zur effizienten Verwaltung der Passwörter kann man, wie oben beschrieben, ein Passwortspeicher-Programm einsetzten.

Die Sicherheitsfrage kann schnell wieder alles kaputt machen

Bei vielen Diensten im Internet kann man beim Einrichten eine Antwort auf eine sogenannte Sicherheitsfrage angeben, mit deren Hilfe man später ein vergessenes Passwort wieder zurücksetzen kann. Als Sicherheitsfrage wird dann oft zum Beispiel nach dem Mädchenname der eigenen Mutter oder nach dem Namen des ersten Haustieres gefragt. Da solche Antworten aber relativ leicht zu erraten sind, wird die Sicherheit des eigentlichen Passwortes durch solche Sicherheitsfragen schnell ad absurdum geführt. Ein möglicher Ausweg ist es, als Antwort auf die Sicherheitsfrage einfach auch wieder eine starke Zufallspassphrase hinreichender Länge zu verwenden. Diese Sicherheits-Passphrase kann man dann getrennt von dem eigentlichen Passort wiederum in einem Passwortspeicher oder auch ausgedruckt an einem sicheren Ort verwahren.

Testen von Passwörtern im Internet

Im Internet findet man oft kostenlose Seiten, auf denen man angeblich die Sicherheit von Passwörtern testen kann. Dazu soll man das Passwort auf der Seite eintippen, worauf dann meist eine Zeit berechnet wird, die angeben soll, wie lange Brute-Force-Angriff auf das Passwort dauern würde. Diese Zeit hängt dann oft lediglich von der Stellenzahl des eingegebenen Passwortes ab, so dass es völlig ausreichen würde, statt nach dem Passwort lediglich nach dessen Stellenzahl zu fragen. Da man als

Nutzer keinerlei Kontrolle darüber hat, ob der Anbieter einer solchen Seiten nicht vielleicht doch die eingegebenen Passwörter speichert, um sie in Rainbowtables zu verwenden oder um sie weiterzuverkaufen, sollte man kritisch hinterfragen, ob es tatsächlich sinnvoll ist, seine Passwörter solchen Seiten anzuvertrauen. Insbesondere sollte man auch kritisch hinterfragen, wie der Seitenbetreiber sein Angebot wohl finanziert und warum er es kostenlos im Internet anbietet.

Auswahl eines Passwortspeicher-Programms

Da in Passwortspeicher-Programmen besonders sensible Daten abgelegt werden, sollte man sorgfältig auswählen, welches Programm man dazu verwendet. Da bei proprietären Programmen der Quelltext in der Regel nicht veröffentlicht wird, ist hier das Kerckhoffs'sche Prinzip meist verletzt, sodass es keinerlei Möglichkeit gibt, die Vertrauenswürdigkeit solcher proprietärer Programme zu prüfen. Bei freier Software gibt es zwar eine solche Überprüfungsmöglichkeit. Allerdings ist dies alleine noch keine Garantie dafür, dass auch tatsächlich jemand diese Möglichkeit wahrgenommen und die Software ausgiebig getestet oder sogar vollständig auditiert hat. Eine gute Möglichkeit sich zu informieren, welche Programme in der IT-Security-Community als vertrauenswürdig gelten, bietet das https://www.kuketz-blog.de.

Internetbrowser bieten in der Regel auch die Möglichkeit, Passwörter verschlüsselt zu speichern. Allerdings sind Browser relativ komplexe und sehr vielseitige Programme, so dass sie im Vergleich zu einem auf Sicherheit optimierten Passwortspeicher-Programm auch fehleranfälliger sind. Hinzu kommt, dass Internet-Browser bei Angriffen aus dem Internet anders als externe Passwortspeicher auch oft das erste und damit einfachste Angriffsziel darstellen. Daher sollte man zumindest für wertvolle Passwörter tendenziell eher einen externen Passwortspeicher bevorzugen.

IOT (Internet of Things)

Ebenso wie im Internet sollte man auch zu Hause für verschiedene Dienste und Geräte stets unterschiedliche Passwörter verwenden. Besonders gefährdet für Angriffe sind billig hergestellte IOT-Geräte, da die Hersteller aus Kostengründen nach dem Verkauf teilweise nur mit großer zeitlicher Verzögerung oder sogar überhaupt keine Updatemöglichkeiten bei auftretenden Sicherheitslücken anbieten.

Fingerabdruck-Scanner

Auch durch die Verwendung eines Fingerabdruckscanners als alternative Authentifizierungsmöglichkeit kann man leicht die Sicherheit eines starken Passwortes wieder gefährden. Denn im Gegensatz zu Passwörtern lassen sich Fingerabdrücke kaum geheim halten und das Wechseln eines Fingerabdrucks bei Verdacht auf Missbrauch ist sogar überhaupt nicht möglich. Hinzu kommt, dass gerade die zu schützenden Geräte meist übersät sind von Fingerabdrücken, die eine potentieller Angreifer nutzen kann, um sich unbefugten Zugang zu verschaffen. Es sind Fälle bekannt, bei denen es Angreifern gelungen ist, Fingerabdruckscanner zu kompromittieren, indem sie einfach einen auf Folie ausgedruckten Fingerabdruck oder einen 3D-Druck eines Fingerabdrucks verwendet haben.

Online-Banking

Um zusätzliche Sicherheit zu gewinnen, werden beim Online-Banking sogenannte TANs (Transaction Numbers) verwendet. Dabei werden verschiedene Verfahren unterschieden.

PIN-TAN-Verfahren

Der Benutzer erhält eine Papierliste mit TANs, die er bei Bedarf eingeben kann. Da bei diesem Verfahren sowohl PIN (Personal Identification Number) als auch TAN in den Browser eingegeben werden, ist bei einem Man-in-the-Browser-Angriff gleich auch das PIN-TAN-Verfahren mit betroffen. Das PIN-TAN-Verfahren gilt daher allgemein als relativ unsicher, so dass es viele Banken heute nicht mehr verwenden.

Mobile-TAN-Verfahren

Die Benutzerin erhält bei Bedarf eine TAN auf ihr Mobiltelefon gesendet. Diese TAN tippt sie dann in den Browser mit ihrer Online-Banking-Sitzung ein. Hierbei sollte man darauf achten, unterschiedliche Geräte für das Online-Banking und den TAN-Empfang zu verwenden, da man ansonsten schnell die zunächst gewonnene zusätzliche Sicherheit wieder gefährdet, da ein Angreifer dann doch wieder nur ein Gerät kompromittieren müsste. Auch ist zu bedenken, dass sich ein Mobiltelefon nur schwer so einrichten lässt, dass es hohen Sicherheitsanforderungen genügt.

Chip-TAN-Verfahren

Die TAN wird mit einem externen Gerät, dem sogenannten TAN-Generator, erzeugt, in das man die EC-Karte einsteckt. Die Überweisungsdaten werden mittels eines sogenannten Flickercodes über den Bildschirm auf den TAN-Generator übertragen. Die generierte TAN ist dann ausschließlich für die eingegebenen IBAN und den eingegebenen Betrag gültig. Zur Sicherheit werden IBAN und Betrag noch einmal auf dem externen Gerät angezeigt und sollten von dort (und nicht von der Anzeige des Browsers!) noch einmal kontrolliert werden. Dieses Verfahren gilt allgemein als relativ sicher, da der externe TAN-Generator keine Verbindung zum Internet aufbaut, so dass sich ein potentieller Angreifer physikalischen Zugang zu dem Gerät und zu der EC-Karte verschaffen müsste.

2-Faktor-Authentifizierung

Wird hierbei nur ein einziges Gerät für beide Authentifizierungen verwendet, so gefährdet man wiederum die zunächst eigentlich durch den zweiten Faktor gewonnene Sicherheit, da ein Angreifer dann nur ein einziges Gerät kompromittieren müsste. Daher sollte man darauf achten, dass bei dem zweiten Faktor ein unterschiedliches Gerät involviert ist, so dass ein potentieller Angreifer dann auch tatsächlich die Kontrolle über beide beteiligten Geräte erlangen müsste.

Vertrauenswürdiges Computersystem

Passwörter sollten nur auf vertrauenswürdigen Computersystemen eingegeben werden. Insbesondere sollte man vorsichtig sein bei der Verwendung von fremden Rechnern, in Internetcafes sowie in öffentlichen Netzwerken. Im Zweifelsfall sollte man sein Passwort ändern, wenn man den Verdacht hat, dass es auf wenig vertrauensvollen Computersystemen mitgeschnitten worden sein könnte. Bei der Auswahl von Betriebssystemen und Software sollte man bedenken, dass proprietäre Software aufgrund der Geheimhaltung des Quelltextes und der damit einhergehenden Verletzung des Kerckhoffs'schen Prinzip meist nicht auf Hintertüren oder Schadsoftware untersucht werden kann.

Praxistest

Erstelle beliebige Hashwerte von Passwörter

Hashgenerator

und versuche diese mithilfe von bereitgestellten Tools wie z.B.:

Crackstation

zu knacken.

Have I beeen pwned?

Have I Been Pwned? (';–have i been pwned?, „pwned“ steht für „owned“, wird jedoch wie „poned“ ausgesprochen, übersetzt in etwa „Hat's mich erwischt?“) ist eine Website, auf der Nutzer überprüfen können, ob ihre persönlichen Daten durch Datenlecks kompromittiert wurden. Der Dienst greift auf eine Vielzahl von Datenbankdumps und Pastebins zu und ermöglicht es dem Nutzer so Milliarden von geleakten Konten auf die eigenen Informationen zu durchsuchen. Über eine Anmeldung können Nutzer sich auch benachrichtigen lassen, wenn ihre Daten in zukünftigen Dumps auftauchen. Die Website gilt weithin als wertvolle Ressource für Internetnutzer, um ihre eigene Sicherheit und Privatsphäre zu überprüfen.

Have I been password pwned

Python - Passwort Brute Force Attack

5-stelliges Passwort

```
import hashlib
import datetime
import time


print("Given Hash: ", hashlib.sha1("hallo".encode('utf-8')).hexdigest())
passhash = hashlib.sha1("hallo".encode('utf-8')).hexdigest()


#Passwort 5 Zeichen lang
alphabet_lowercase=["a","b","c","d","e","f","g","h","i","j","k","l","m","n",
"o","p","q","r","s","t","u","v","w","x","y","z"]
alphabet_uppercase=["A","B","C","D","E","F","G","H","I","J","K","L","M","N",
"O","P","Q","R","S","T","U","V","W","X","Y","Z"]
digits=["0","1","2","3","4","5","6","7","8","9"]

alphabet=alphabet_lowercase+alphabet_uppercase+digits
```

```
finished = False

print("\n#### Start Bruteforce Attack #### \n")

ts_start=time.time()
print(datetime.datetime.now())

while finished == False:
    for char1 in alphabet:
        for char2 in alphabet:
            for char3 in alphabet:
                for char4 in alphabet:
                    for char5 in alphabet:
                        # Kombination erstellen und ausgeben
                        password = char1+char2+char3+char4+char5
                        #print(password)
hash=hashlib.sha1(password.encode('utf-8')).hexdigest()
                        if hash == passhash:
                            print("\nHash found:", hash)
                            print("Password:", password,"\n")

                            ts_end=time.time()
                            print(datetime.datetime.now())

                            finished = True
                            break
                    if finished == True:
                        break
                if finished == True:
                    break
            if finished == True:
                break
        if finished == True:
            break
    if finished == True:
        break

print("\nTime for Hacking:" , ts_end - ts_start, "s")
```

Python - Attacke mithilfe von Listen von häufig vorkommenden / gestohlenen Passwörtern

Die häufigsten 10 Millionen Passwörter Passwortliste

Passwortliste von Crackstation Cracking Dictionary

```
import hashlib
import datetime
import time

wahl = int(input("Wollen Sie einen Hash-wert eines Passworts (1) oder das
Passwort (2) selbst eingeben: "))
wort=""
passhash=""

if wahl == 2:
    wort = input("Geben Sie das gewünschte Passwort ein, nachdem in der
Passwortliste gesucht werden soll: ")
    print("Given Hash: ", hashlib.sha1(wort.encode('utf-8')).hexdigest())
    passhash = hashlib.sha1(wort.encode('utf-8')).hexdigest()
else:
    passhash = input("Geben Sie einen Hashwert (SHA-1, z.B.: von
http://www.sha1-online.com generiert) ein: ")


foundpassword = False

ts_start = time.time()

print("\n...Suche in Passwortliste 1....\n")

file1 = open("passwords.txt", "r")
for line in file1:
    line=line[:-1]
    #print(line)
    hash = hashlib.sha1(line.encode('utf-8')).hexdigest()
    if passhash == hash:
        foundpassword = True
        break
file1.close()

if foundpassword == False:
    print("\n...Suche in Passwortliste 2....\n")
    file2 = open("passwords2.txt", "r", encoding="utf-8", errors="ignore")
    for line in file2:
        line=line[:-1]
        #print(line)
        hash = hashlib.sha1(line.encode('utf-8')).hexdigest()
        if passhash == hash:
            foundpassword = True
            break
    file2.close()


if foundpassword == False:
    print("\nKeine Übereinstimmung!")
else:
```

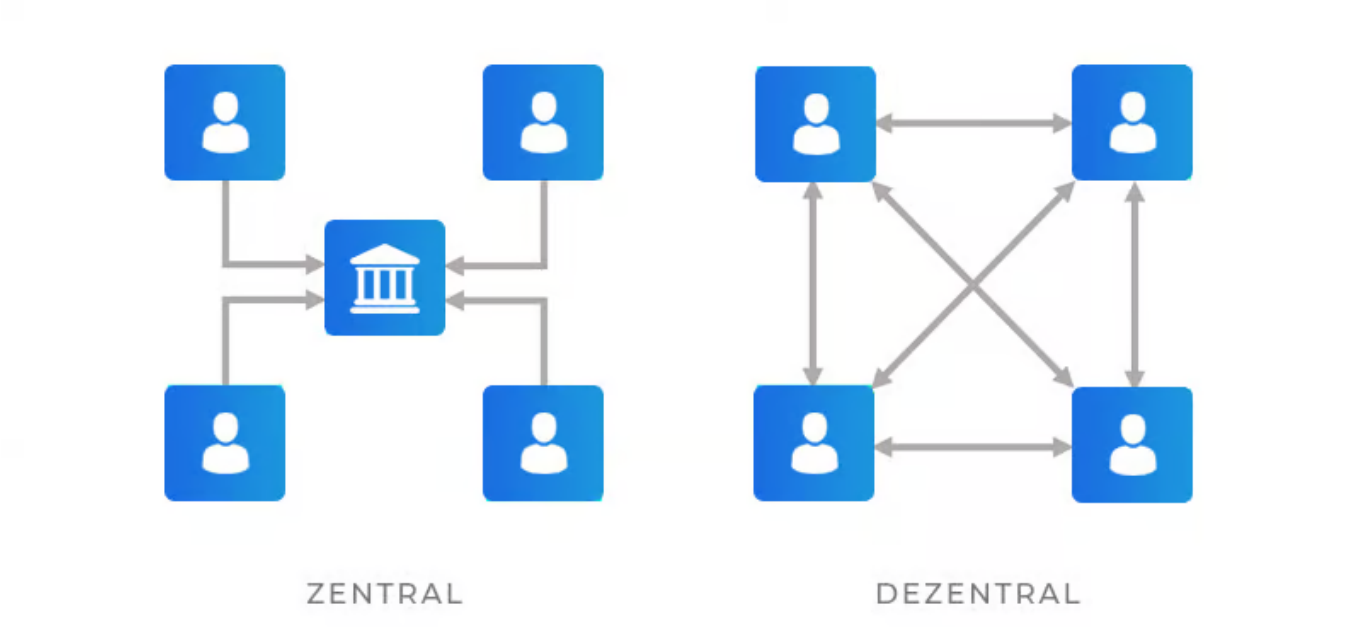
```
    print("Passwort gefunden:", line)
    print("Hashwert:", hash)

ts_end = time.time()
print("Zeitdifferenz: ", ts_end - ts_start)
```

Bitcoin

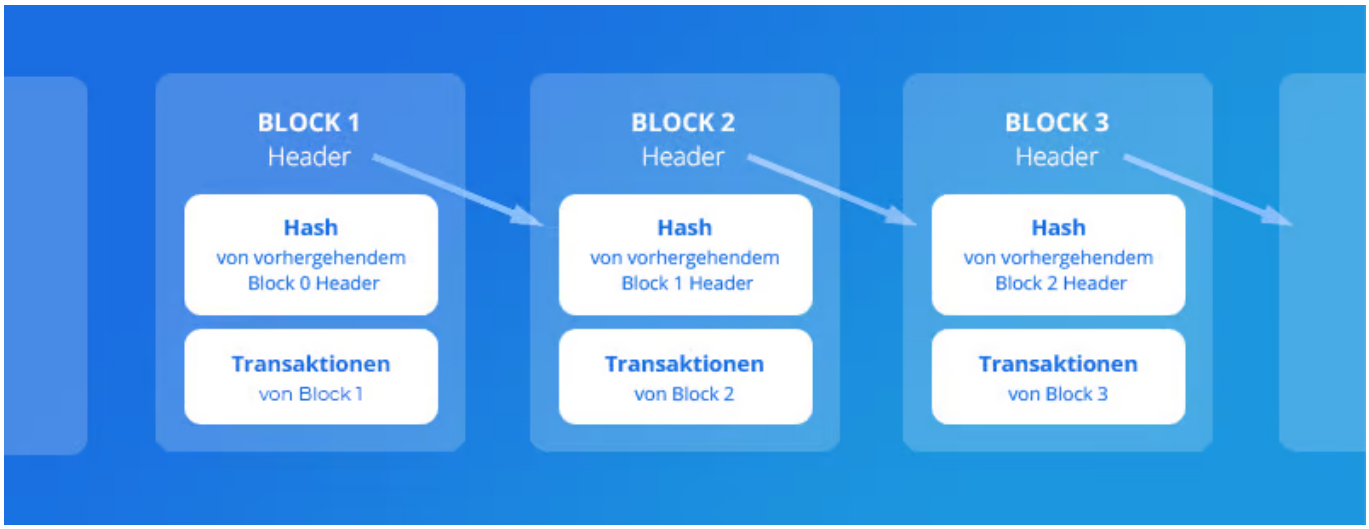
Seit Bitcoin 2009 veröffentlicht wurde, erfreuen sich zahlreiche weitere Kryptowährungen wie Ethereum, Tether, Ripple, Monero oder zuletzt auch Dogecoin großer Beliebtheit.

Doch wie funktioniert eigentlich eine Kryptowährung wie Bitcoin und wodurch wird beispielsweise sichergestellt, dass man sein Geld nicht mehrfach ausgibt oder ausgeraubt wird?

Typischerweise basieren Kryptowährungen wie Bitcoin auf einem **Peer-to-Peer-Netzwerk** mit einem **Buchungssystem**, das keine **zentrale Regulierungsstelle** benötigt, sondern vielmehr durch alle Benutzer*innen des Netzwerks reguliert wird. Die Benutzer*innen tauschen über das Netzwerk **Informationen über neue Transaktionen aus und überprüfen diese**. Alle gültigen Transaktionen werden in einen **Distributed Ledger** eingetragen, quasi ein **digitales verteiltes Kassenbuch**, das zwischen allen Benutzer*innen verteilt wird. Durch den Einsatz kryptographischer Methoden wird sichergestellt, dass Transaktionen in eine korrekte zeitliche Abfolge gebracht und auf Gültigkeit überprüft (verifiziert) werden können.

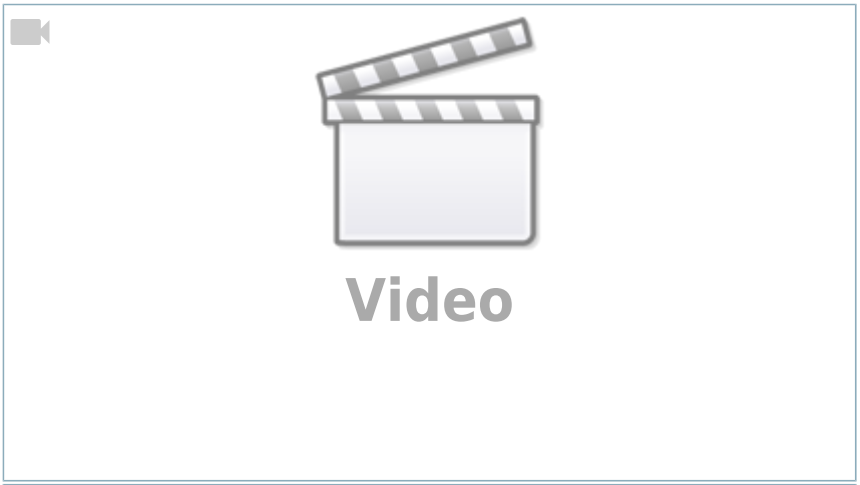


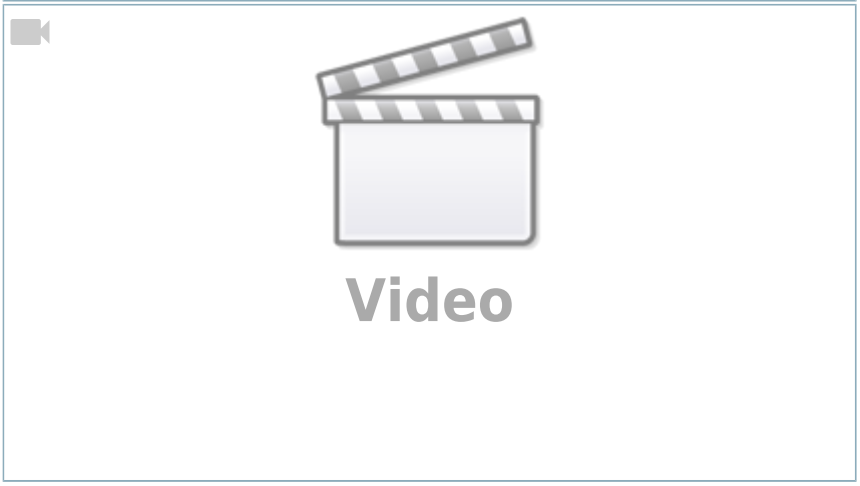
Für Bitcoin wurde als Distributed Ledger die sogenannte **Blockchain** entworfen. Um zu verstehen, wie die Blockchain sicherstellt, dass Bitcoins überhaupt entstehen können, dass diese nicht mehrfach ausgegeben, kopiert, gestohlen oder ungültig gemacht werden können, werden **zwei kryptographische Mechanismen** eingesetzt: **Kryptographische Hashes** und eine **Public-Key-Infrastruktur (PKI)**.



Funktionsweise

Blockchain Funktionsweise





Bitcoin Mining

Kommunikation in Rechnernetzwerken

Grundlagen

Was ist ein Netzwerk?

Ein Netzwerk ist die **physikalische und logische Verbindung von Computersystemen**. Ein einfaches Netzwerk besteht aus zwei Computersystemen. Sie sind über ein Kabel miteinander verbunden und somit in der Lage ihre Ressourcen gemeinsam zu nutzen. Wie zum Beispiel Rechenleistung, Speicher, Programme, Daten, Drucker und andere Peripherie-Geräte. Ein netzwerkfähiges Betriebssystem stellt den Benutzern auf der Anwendungsebene diese Ressourcen zur Verfügung.

Notwendigkeit für ein Netzwerk

Als es die ersten Computer gab, waren diese sehr teuer. Peripherie-Geräte und Speicher waren fast unbezahlbar. Zudem war es erforderlich zwischen mehreren Computern Daten auszutauschen. Aus diesen Gründen wurden Computer miteinander verbunden bzw. vernetzt.

Daraus ergaben sich einige Vorteile gegenüber unvernetzten Computern:

- zentrale Steuerung von Programmen und Daten
- Nutzung gemeinsamer Datenbeständen
- erhöhter Datenschutz und Datensicherheit
- größere Leistungsfähigkeit
- gemeinsame Nutzung der Ressourcen

Die erste Möglichkeit, Peripherie-Geräte gemeinsam zu nutzen, waren manuelle Umschaltboxen. So konnte man von mehreren Computern aus einen Drucker nutzen. An welchem Computer der Drucker angeschlossen war, wurde über die Umschaltbox bestimmt. Leider haben Umschaltboxen den Nachteil, dass Computer und Peripherie beieinander stehen müssen, weil die Kabellänge begrenzt ist.

Größenordnung von Netzwerken

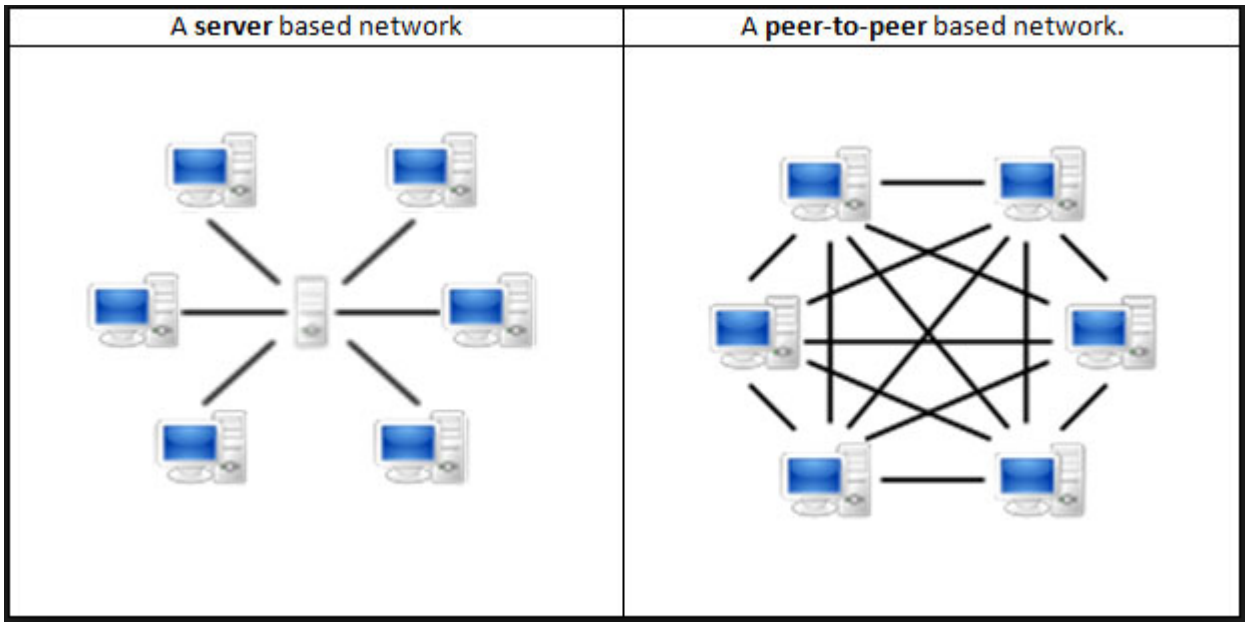
Jedes Netzwerk basiert auf Übertragungstechniken, Protokollen und Systemen, die eine Kommunikation zwischen den Netzwerk-Teilnehmern ermöglichen. Bestimmte Netzwerktechniken unterliegen dabei Beschränkungen, die insbesondere deren Reichweite und Ausdehnung begrenzt. Hierbei haben sich verschiedene Netzwerk-Dimensionen durchgesetzt für die es unterschiedliche Netzwerktechniken gibt.

- PAN - Personal Area Network: personenbezogenes Netz, z. B. Bluetooth
- LAN - Local Area Network: lokales Netz, z. B. Ethernet
- MAN - Metropolitan Area Network: regionales Netz
- WAN - Wide Area Network: öffentliches Netz, z. B. ISDN
- GAN - Global Area Network: globales Netz, z. B. das Internet

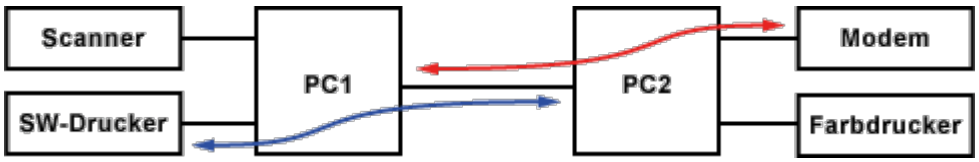
In der Regel findet ein Austausch zwischen den Netzen statt. Das heißt, dass Netzwerk-Teilnehmer eines LANs auch ein Teilnehmer eines WANs oder eines GANs ist. Eine 100%ig klare Abgrenzung zwischen diesen Dimensionen ist nicht immer möglich, weshalb man meist nur eine grobe Einteilung vornimmt. So unterscheidet man in der Regel zwischen LAN und WAN, wobei es auch Techniken und Protokolle gibt, die sowohl im LAN, als auch im WAN zum Einsatz kommen.

Peer-to-Peer-Netze und Client-Server-Architekturen

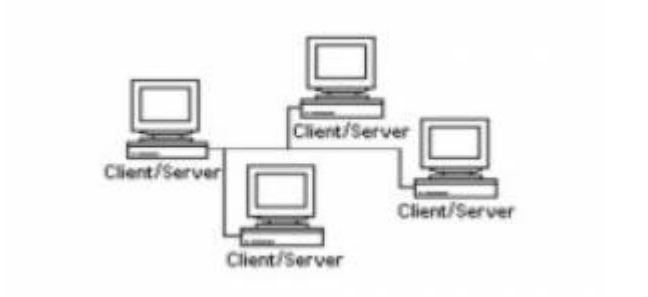
Man unterscheidet zwei „Philosophien“:



Peer-to-Peer-Netzwerke



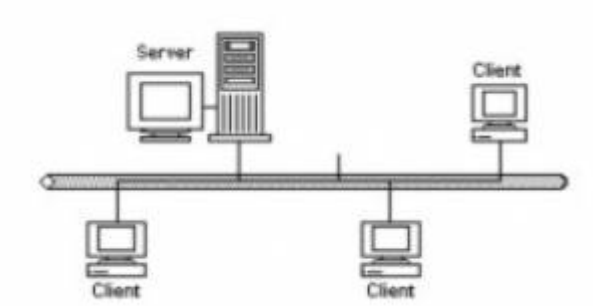
In einem Peer-to-Peer-Netzwerk ist jeder angeschlossene Computer zu den anderen gleichberechtigt. Jeder Computer stellt den anderen Computern seine Ressourcen zur Verfügung. Ein Peer-to-Peer-Netzwerk eignet sich für bis zu 10 Stationen. Bei mehr Stationen wird es schnell unübersichtlich. Diese Art von Netzwerk ist relativ schnell und kostengünstig aufgebaut. Die Teilnehmer sollten möglichst dicht beieinander stehen. Einen Netzwerk-Verwalter gibt es nicht. Jeder Netzwerk-Teilnehmer ist für seinen Computer selber verantwortlich. Deshalb muss jeder Netzwerk-Teilnehmer selber bestimmen, welche Ressourcen er freigeben will. Auch die Datensicherung muss von jedem Netzwerk-Teilnehmer selber vorgenommen werden.



Peer-to-Peer-Netze brauchen keinen eigenen Server-Rechner, da jeder PC Server-Funktionen übernehmen kann.

Client/Server-Architekturen

In einem serverbasierten Netzwerk werden die Daten auf einem zentralen Computer gespeichert und verwaltet. Man spricht von einem dedizierten Server, auf dem keine Anwendungsprogramme ausgeführt werden, sondern nur eine Server-Software und Dienste ausgeführt werden. Diese Architektur unterscheidet zwischen der Anwender- bzw. Benutzerseite und der Anbieter- bzw. Dienstleisterseite. Der Anwender betreibt auf seinem Computer Anwendungsprogramme (Client), die die Ressourcen des Servers auf der Anbieterseite zugreifen. Hier werden die Ressourcen zentral verwaltet, aufgeteilt und zur Verfügung gestellt. Für den Zugriff auf den Server (Anfrage/Antwort) ist ein Protokoll verantwortlich, dass sich eine geregelte Abfolge der Kommunikation zwischen Client und Server kümmert. Die Client-Server-Architektur ist die Basis für viele Internet-Protokolle, wie HTTP für das World Wide Web oder SMTP/POP3 für E-Mail. Der Client stellt eine Anfrage. Der Server wertet die Anfrage aus und liefert eine Antwort bzw. die Daten zurück.



Netzwerk-Topologien

Die Struktur eines Netzwerks bezeichnet man als Topologie. Wie wichtig die Struktur eines Netzwerks ist, merkt man bei einem Leitungsausfall: ein gutes Netzwerk findet bei einem Leitungsausfall selbstständig einen neuen Pfad zum Empfänger.

Physikalische und logische Topologie

Interessant ist, dass sich die **sichtbare Topologie** (also die physische Verkabelungsstruktur) vom tatsächlichen Datenfluss unterscheiden kann. Deshalb verwendet man für die hardwaremäßige Realisierung den Begriff **physikalische Topologie** während man für den tatsächlichen Datenfluss den Begriff „logische Topologie“ verwendet.

Die wichtigsten Netzwerktopologien sind:

Bus-Topologie

ALLE GERÄTE nutzen DASSELBE KABEL

Bei einem Bussystem sind alle Rechner hintereinander geschaltet und über Abzweige (T-Stücke) an das Netzwerkkabel angeschlossen. Problem: Eine Verbindungsunterbrechung betrifft den ganzen Bus!

Vorteile

- Relativ niedrige Kosten, da geringe Kabelkosten
- Ausfall einer Station führt zu keinem Netzausfall

Nachteile

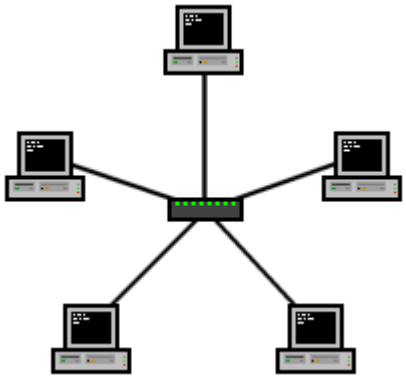
- Alle Daten über ein Kabel
- Nur eine Station kann senden. Alle anderen sind blockiert.
- Eine Störung an einer Stelle (z.B.: Defektes Kabel) führt zu einem Netzausfall (⇒ aufwendige Fehlersuche)
- Unverschlüsselter Netzwerkverkehr kann direkt am Bus (=Kabel) mitgelesen werden

Einsatzgebiet

Früher aufgrund der niedrigen Kosten häufig verwendet, heute spielt die Bus-Topologie keine Rolle mehr und wurde von der Stern-Topologie verdrängt.

Bus-Topologie - Details

Stern-Topologie



JEDES GERÄT nutzt EIN KABEL.
Damit ist es zu einem Verteiler verbunden. Es existiert eine Punkt-zu-Punkt Verbindung zwischen Verteiler und Gerät. Als Verteiler kann ein HUB oder ein SWITCH dienen.

Vorteile

- Ausfall einer Station oder eines Defekts an einem Kabel führt zu keinem Netzausfall
- Aktive Verteiler (Switch, Hub) dienen gleichzeitig als Signalverstärker
- Bei richtiger Konfiguration können zwei Stationen die volle Bandbreite des Übertragungsmediums für ihre Kommunikation nutzen, ohne andere Stationen dabei zu behindern. Diese physikalische Topologie erlaubt somit sehr hohe Datendurchsatzraten.
- Weitere Stationen oder Verteiler können einfach hinzugefügt werden. Sehr leicht skalierbar.

Nachteile

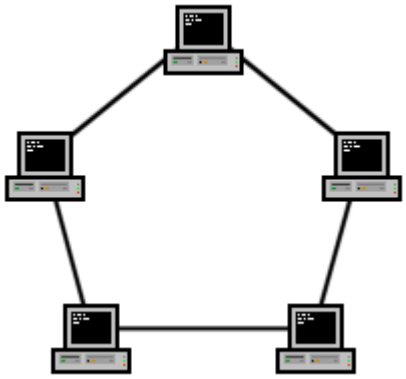
- Große Kabelmengen
- Beim Ausfall des Verteilers ist kein Netzverkehr mehr möglich

Einsatzgebiet

Im praktischen Einsatz bei lokalen Netzwerken findet die Stern-Topologie Verwendung. Häufigste Form der Verkabelung.

Stern-Topologie - Details

Ring-Topologie



JEDES GERÄT ist mit ZWEI NACHBARN verbunden.

Die Ring-Topologie ist eine geschlossene Form, es gibt keinen Kabelanfang und kein Kabelende.
Es handelt sich jeweils um eine Punkt-zu-Punkt Verbindung zwischen den Rechnern.
Jede Station hat genau einen Vorgänger und einen Nachfolger.
Datenverkehr findet immer nur in eine Richtung statt.

Vorteile

- Vorgänger und Nachfolger sind festgelegt
- Alle Stationen verstärken das Signal
- Alle Stationen haben gleiche Zugriffsmöglichkeit
- Leicht umsetzbar

Nachteile

- Ausfall einer Station oder eines Kabelteils führt zu einen Netzausfall
- Hoher Aufwand bei der Verkabelung (Jede Station braucht 2 Netzwerkkarten)
- Leicht abhörbar
- langsame Datenübertragung bei vielen Stationen

Einsatzgebiet

Physikalische Anwendung gibt es heute keine mehr.
Logische Anwendung findet sie im Token Ring.

Ring-Topologie - Details

Mischformen

Sind zumeist **Kombinationen aus Bus, Stern und Ring**.

Backbone

Unter einem Backbone („Rückgrat“) wird die physikalische Verbindung zwischen einzelner Teilnetze verstanden. Es wird auch oft als Hintergrundnetz betitelt und verbindet z.B. mehrere Gebäude.

Stern-Bus-Netz

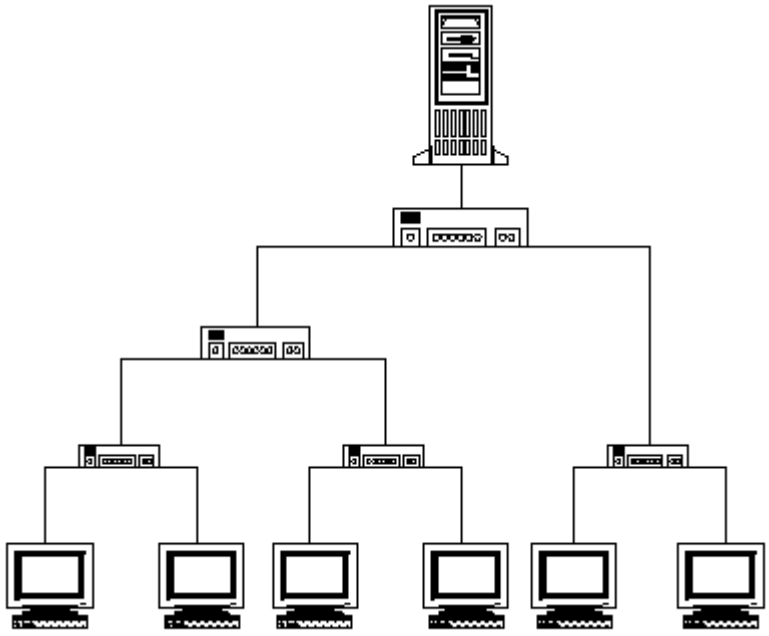
Ein Stern-Bus-Netz entsteht, wenn mehrere Verteiler über einen Bus miteinander verbunden sind. Häufig sind so mehrere Stockwerke miteinander verbunden.

Stern-Stern-Netz

Ein Stern-Stern-Netz entsteht, wenn mehrere Verteiler wiederum über einen Verteiler verbunden sind. Häufigste Anwendung ist wiederum das Verbinden von mehreren Subnetzen (z.B. Netze in verschiedenen Stockwerken).
Fällt der Hauptverteiler aus, so kann zwischen den Stockwerken nicht mehr kommuniziert werden. Jedoch kann man auch die Hauptverteiler redundant auslegen.

Baum

Eine Baum-Topologie wird so aufgebaut, dass, ausgehend von der Wurzel, eine Menge von Verzweigungen zu weiteren Verteilungsstellen existiert.
Es handelt sich somit um eine Erweiterung der Stern-Stern-Topologie auf mehrere Ebenen.



Maschen-Topologie

Vorherrschende Netzstruktur in großflächigen Netzen (z.B. öffentliche Telekommunikationsnetze).
Maschen-Topologie - Details

Zell-Topologie

Die Zell-Topologie kommt hauptsächlich bei drahtlosen Netzen zum Einsatz. Eine Zelle ist der Bereich um eine Basisstation (z.B. Wireless Access Point), in dem eine Kommunikation zwischen den Endgeräten und der Basisstation möglich ist.
Zell-Topologie - Details

Zusammenfassung

Topologie	Vorteile	Nachteile
Bus-Topologie	• einfach installierbar • kurze Leitungen	• Netzausdehnung begrenzt • bei Kabelbruch fällt Netz aus • aufwändige Zugriffsmethoden
Ring-Topologie	• verteilte Steuerung • große Netzausdehnung	• aufwendige Fehlersuche • bei Störungen Netzausfall • hoher Verkabelungsaufwand
Stern-Topologie	• einfache Vernetzung • einfache Erweiterung • hohe Ausfallsicherheit	• hoher Verkabelungsaufwand • Netzausfall bei Ausfall oder Überlastung des Hubs
Maschen-Topologie	• dezentrale Steuerung • unendliche Netzausdehnung • hohe Ausfallsicherheit	• aufwendige Administration • teure und hochwertige Vernetzung

Übertragungsmedien

Die Übertragungsmedien sind die Straßen der Daten. Der Aufbau dieser Straßen muss sehr gut geplant werden, um alle aktuellen Anforderungen bzw. eventuelle zukünftige Anforderungen ohne großartige Veränderungen zu erfüllen.

Als **Maßeinheit für die Übertragungsgeschwindigkeit** werden die Werte in **bit/s, b/s bps, -> also Bit pro Sekunde** angegeben. Achtung: Nicht zu verwechseln mit **Byte/s -> Byte pro Sekunde**!!

`C= D/t` `(bits)/(s)`

1) Rechenbeispiel:

Es werden 100MB in 10s übertragen. Wie hoch ist die Übertragungsgeschwindigkeit?

- D=100MByte
- t=10s

Rechenschritt	**Berechnung**
D umwandeln in bits	`100*1024*1024*8 = 838860800` `bits`
C berechnen	`C=D/t=(838 860 800)/10=83886080` `(bit)/(s)`
C umwandeln in Mbit/s	`(83886080)/(1024)/(1024)=80` `(Mbit)/(s)`

2) Rechenbeispiel:

Max hat eine Datentransferrate von 10Mbit/s Download und 2Mbit/s Upload.

a) Wie lange braucht er, um 10MB runterzuladen?

- D=10MB
- C=10Mbit/s

Rechenschritt	**Berechnung**
D umwandeln in bits	`10*1024*1024*8 = 83886080` `bits`
C umwandeln in bit/s	`10*1024*1024=10485760` `(bit)/(s)`
Formel umformeln	`t=(D)/(C)` `(bits)/((bit)/s)`
In Formel einsetzen	`t=83886080/10485760=8` `s`

b) Wie lange braucht er, um 10MB hochzuladen?

Rechenschritt	**Berechnung**
D umwandeln in bits	`10*1024*1024*8 = 83886080` `bits`
C umwandeln in bit/s	`2*1024*1024=2097152` `(bit)/(s)`
Formel umformeln	`t=(D)/(C)` `(bits)/((bit)/s)`
In Formel einsetzen	`t=83886080/2097152=40` `s`

Leitergebundene Übertragung

Bei einer leitergebundenen Übertragung werden Medien in Form von Kabeln benötigt (Metallische Leiter, Glasfaser).
Ein Kabel besteht zumindest aus einer Ader (=Faser).
Mehrere Adern sind durch entsprechende Isolationssschichten getrennt.
Alle Adern wiederum werden von einem Mantel als Schutz umgeben.
Die Übertragung selbst erfolgt durch elektromagnetische Schwingungen.

Koaxialkabel

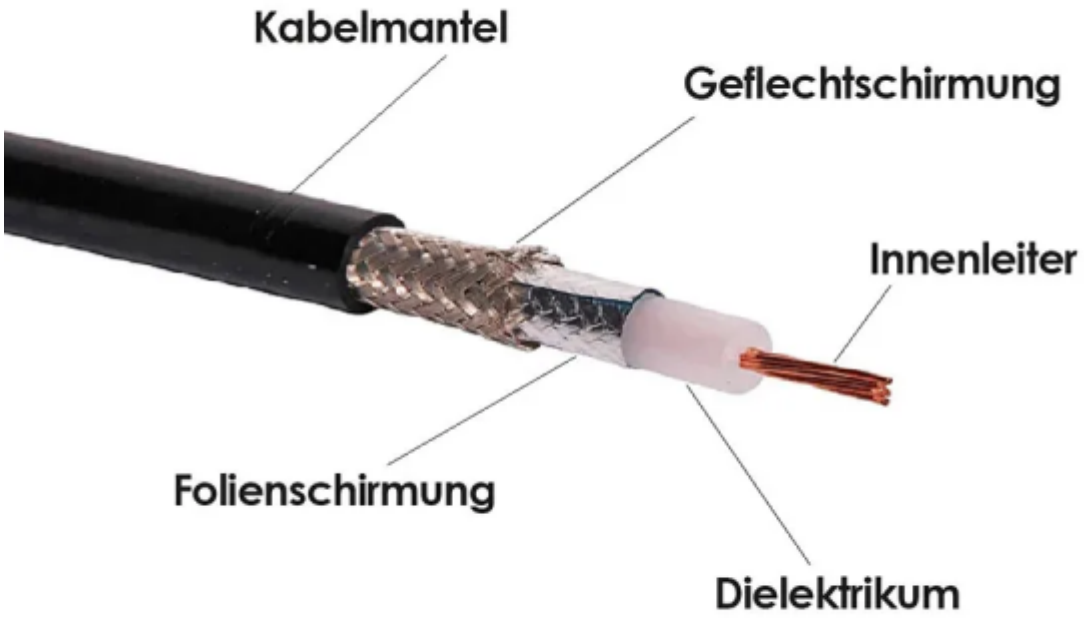
Das früher verwendete Koaxialkabel ist in modernen Netzen praktisch vollständig von Twisted Pair-Kabeln (TP) und Lichtwellenleiter (LWL) abgelöst worden.

Es besteht aus

- einem Innenleiter (Kupfer, Stahlkupfer)
- einer Isolation (Dielektrikum)
- einer Abschirmung (Metallgeflecht schützt vor magnetischen Störungen –> Rauschen & Übersprechen)
- einem Mantel

Es waren bis zu 10Mbps möglich:

- Thicknet (10Base5)
- Thinnet (10Base2) - Heute noch bei Satellitenempfang im Einsatz

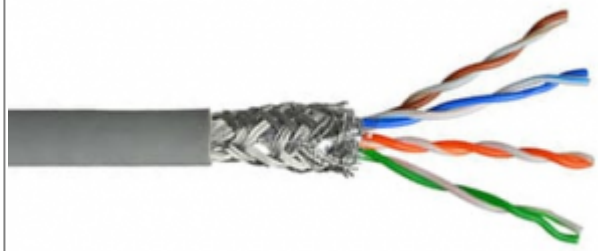
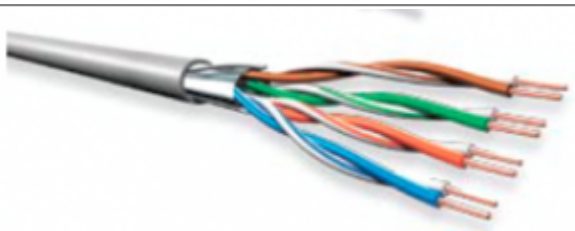
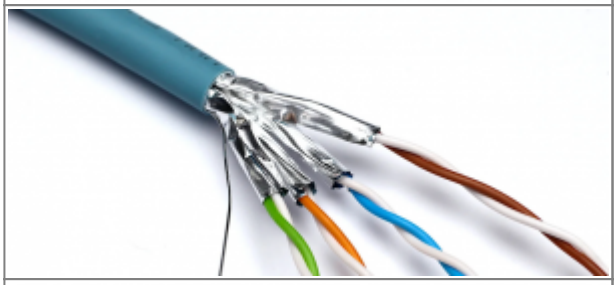
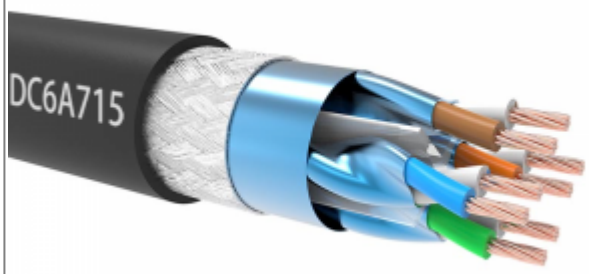


Twisted-Pair Kabel

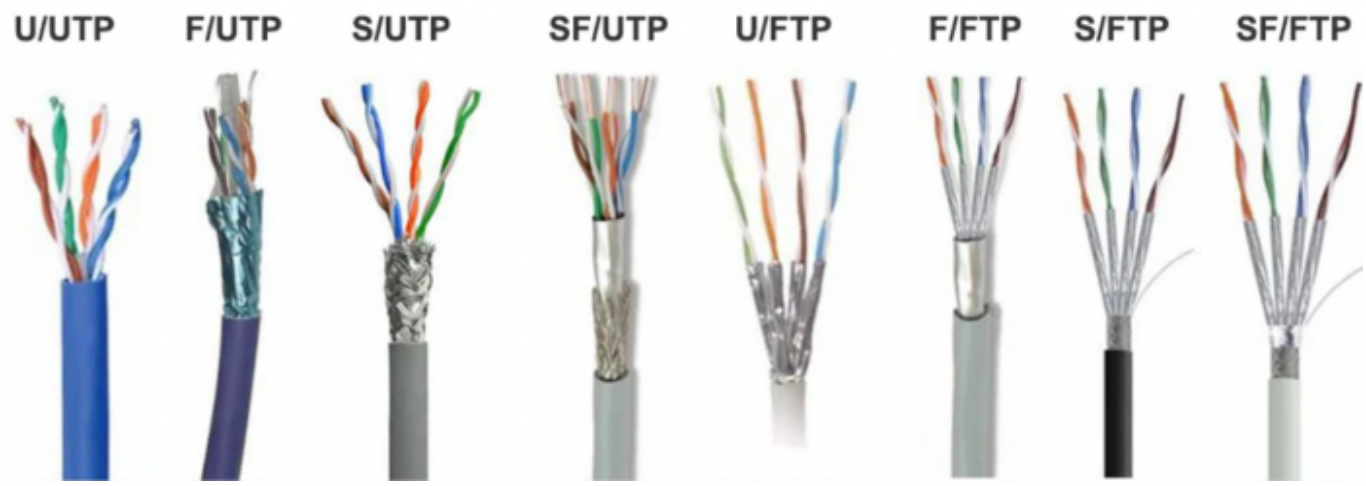
Twisted Pair ist ein vier-, acht- oder mehr-adriges Kupferkabel, bei dem jeweils zwei Adern miteinander verdrillt sind. Durch die Verdrillung kompensieren sich Leitungskapazität und -induktivität. Dadurch steigt die Übertragungsbandbreite und die mögliche Übertragungsreichweite wird praktisch nur durch die Dämpfung des Wirkwiderstandes begrenzt. Die Verwendung von symmetrischen Signalen (Differentialspannungen) erhöht die Festigkeit gegen elektromagnetische Störstrahlung.

Twisted Pair-Kabel gibt es in zahlreichen Varianten. Twisted Pair-Verbindungen werden außer in der Kommunikationstechnik (Netzwerkkabel, Telefonkabel) auch bei HDMI-, DVI- und LVDS-(in LCD/Plasma-TV zwischen Signalprozessor und Display) Verbindungen eingesetzt. Die Anzahl der Leiterpaare im Kabel hängt dabei von der benötigten Datenübertragungsrate ab. In Netzwerken wird für jede Übertragungsrichtung (senden, empfangen) wird jeweils ein Adernpaar (bei 100BaseT4 und

1000BaseT jeweils zwei) genutzt. Die Übertragungsreichweite ist abhängig vom Aufbau des Kabels, von der Dämpfung (=Länge) des Kabels und von den externen Störeinflüssen. Twisted Pair-Kabel für Netzwerke gibt es in zahlreichen Varianten:

Bild	Benennung	Beschreibung
	**U/UTP-Kabel**	**Unshielded/Unshielded Twisted Pair** sind nicht abgeschirmte verdrillte Leitungen und gehörten früher typischerweise der CAT3 an. UTP-Kabel sollten im industriellen Bereich oder in der Datentechnik mit hohen Datenraten nicht verwendet werden.
	**S/UTP-Kabel**	**Screened/Unshielded Twisted Pair** haben einen Gesamtschirm aus einem Kupfergeflecht zur Reduktion der äußeren Störeinflüsse
	**F/UTP-Kabel**	**Foilshielded/Unshielded Twisted Pair** besitzen zur Abschirmung einen Gesamtschirm, zumeist aus einer alukaschierten Kunststofffolie
	**U/FTP-Kabel**	**Unshielded/Foilshielded Twisted Pair** auch genannt **CAT5** oder **CAT5e**. Die Leitungsadern sind paarweise mit Folie abgeschirmt
	**S/FTP-Kabel**	**Screened/ Foilshielded Twisted Pair** auch genannt **CAT6** sollten in Bereichen mit hoher Störstrahlung (z.B. Büros mit mehreren PCs) eingesetzt werden.
	**SF/FTP-Kabel**	**Screened Foilshielded/Foilshielded Twisted Pair** auch genannt **CAT6e** oder **CAT7** besitzen eine Abschirmung für jedes Kabelpaar sowie eine doppelte Gesamtschirmung. Hierdurch kann eine optimale Störleistungsunterdrückung erreicht werden. Auch das Übersprechen zwischen den einzelnen Adernpaaren wird so wirksam unterdrückt

Die Preisunterschiede zwischen CAT-5e- Kabeln und CAT-7-Kabeln ist so gering, dass es sich bei Neuinstallation auf jeden Fall empfiehlt, CAT-7-Kabel einzusetzen. Dieses ist als einziges Kupfermedium in der Lage mit dem kommenden 10GBit-LAN verwendet zu werden.



Categories and Types of CAT Cables

Category	Frequency	Data Transmission	Distance
CAT-5	up to 100 MHz	up to 100 Mbps	100 m
CAT-5e	up to 100 MHz	up to 1 Gbps	100 m
CAT-6	up to 250 MHz	up to 10 Gbps	100 m
CAT-6a	up to 500 MHz	up to 10 Gbps	100 m
CAT-7	up to 600 MHz	up to 10 Gbps	100 m
CAT-8	up to 2 GHz	up to 40 Gbps	30 m

Verbinder - RJ45

Der typische Standardverbinder für die Twisted-Pair-Verkabelung eines kupfergebundenen Ethernet-Netzwerkes ist der **8polige Western-Modularstecker RJ-45** (8P8C), auch RJ-48 oder RJ-49 genannt. RJ-45 Steckverbindungen können auf zwei Arten belegt sein, wobei die Belegung nach T568B am weitesten verbreitet zu sein scheint:

Belegung nach EIA/T-T568A		**Belegung nach EIA/T-T568B**	
Pin	**Farbe**	**Pin**	**Farbe**
1	weiß-grün	1	weiß-orange
2	grün	2	orange
3	weiß-orange	3	weiß-grün
4	blau	4	blau
5	weiß-blau	5	weiß-blau
6	orange	6	grün
7	weiß-braun	7	weiß-braun
8	braun	8	braun

Bei 1:1-Verbindungen sind beide Beschaltungen sind elektrisch zueinander kompatibel. Nur bei Erweiterungen von fest verdrahteten Netzen ist festzustellen, welche Belegung bereits vorgegeben ist. Normale Verbindungskabel („ Patchkabel“) mit RJ-45-Steckern sind 1:1 verschaltet, d.h. Pin 1 des einen Steckers geht auf den Pin 1 des anderen Steckers usw. Nur in besonderen Fällen, wenn z.B. zwei Netzwerkkarten direkt miteinander verbunden werden sollen oder wenn Netzwerkkomponenten (z.B. Hubs älterer Bauart) über keinen dedizierten Uplink-Port verfügen, kann der Einsatz von Crossover-Kabeln notwendig werden.

LichtWellenLeiter (LWL)

Sind mit der Netzwerkverkabelung weite Strecken zu überwinden, z.B. zwischen einzelnen Gebäuden auf einem Fabrikgelände („Campusbereich“), sind sehr hohe Datenübertragungsraten (z.Zt. bis zu 170Gb/s) gefordert oder wenn sich die Datenübertragung per Kupferkabel aus technischen Gründen

(z.B. bei extremer Störstrahlung) oder aus Gründen der Sicherheit verbietet, werden Lichtwellenleiter (LWL, Glasfasern) als Übertragungsmedium eingesetzt. Die Lieferprogramme der Hersteller erlauben mittlerweile die Übertragungsstrecken bis zum Einzelplatz komplett auf der Basis von LWL auszuführen.

Aufbau und Prinzip

In einem LWL werden die Informationen nicht, wie in einem Kupferkabel, elektrisch übertragen, sondern mit **Licht**.
Der eigentliche LWL ist eine Faser aus Glas oder Kunststoff. Jede Faser besteht aus zwei Schichten. Der konzentrische Kern besteht aus einem optischen Material mit einem hohen Brechungsindex, das Mantelglas („ Cladding“) aus einem Material mit niedrigem. Licht, das in einem bestimmten Winkelbereich auf den Übergang von Kern zum Mantel trifft wird dort vollständig reflektiert. Über solche fortlaufenden Totalreflexionen pflanzt sich das Licht durch den LWL bis zum Ende der Faser fort.
Je steiler der Einfallswinkel des Lichts bei der Einspeisung in den LWL ist, desto häufiger wird die Lichtwelle reflektiert. Mit jeder Reflektion der Lichtwelle wird der Weg, des sogenannten Modes, länger.

Licht, das wenig häufig reflektiert wird, hat einen kürzeren Weg und durchläuft die Faser schneller. Es ist Licht niedrigen Modes.

Licht, das sehr häufig reflektiert wird, hat eine niedrige Ausbreitungsgeschwindigkeit in der Faser. Er ist Licht hohen Modes.

Erzeugt die Lichtquelle des Senders ein nicht-kohärentes Licht, tritt das Licht mit einer Vielzahl unterschiedlicher Winkel in die Faser ein. Dadurch entstehen natürlich durch die unterschiedlichen Moden Laufzeitunterschiede zwischen den Signalanteilen. Ein Eingangsimpuls mit steilen Flanken wird dadurch verschliffen und in seiner Breite gedehnt. Je länger ein Kabel ist, desto höher wird auch diese sog. Dispersion (Einheit: ns/km). Die Dispersion beeinflusst direkt die Übertragungsbandbreite der Glasfaserverbindung.

Da die Fasern sehr dünn und empfindlich sind, werden sie zum mechanischen Schutz mit einer Kunststoffbeschichtung („ Coating“) und einem Schutzüberzug versehen. In einem LWL-Kabel können mehrere Fasern, sogar in mehreren Bündeln, zusammen gefasst sein.

Verbinder

Die Hersteller von Netzwerkzubehör bieten konfektionierte Verbindungs- und Patchkabel mit einer Vielzahl von verschiedenen Steckerformen an. Meist sind die Kabel paarweise angelegt um beide Datenflussrichtungen (TX und RX) gleichzeitig herstellen zu können.

Vorteile

- hohe Reichweite
- hohe Übertragungsbandbreite
- Potentialfrei, daher auch für explosionsgefährdete Bereiche geeignet
- hohe Störfestigkeit, LWL können sogar zu Energieversorgungskabeln parallel verlegt werden
- hohe Abhörsicherheit

Nachteile

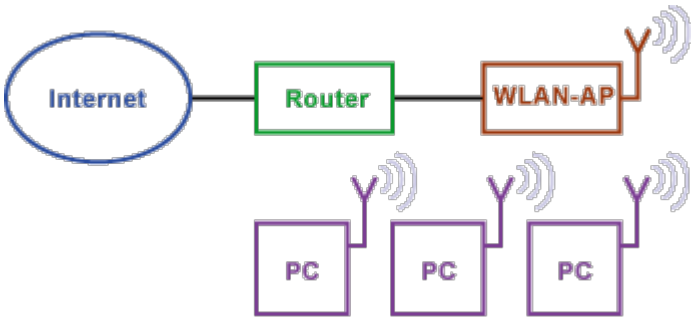
- Material für die Verkabelung ist teuer
- teure Verbindungstechnik
- Die Montagekosten sind wegen des höheren technischen Aufwandes höher
- komplexe und teure Messtechnik
- zusätzliche Kosten für Medienkonverter auf Kupfer-Ethernet

Leiterungebundene Übertragung

Als leiterungebundene Übertragung bezeichnet man eine Übertragung per

- Funk
- Ultraschall
- Infrarot
- Laser
- Licht

Drahtlose Übertragung (WLAN)



Die Übertragung von Informationen ohne Kabel ist mittlerweile in vielen Lebensbereichen als

praktische Alternative eingezogen. Von der Fernbedienung eines Fernsehers über drahtlose Lautsprecher bis zum Smartphone gibt es viele Beispiele für die Umsetzung dieser Technik. Dabei werden Funksignale in frei verfügbaren Frequenzbändern anstelle von Kabeln für die Datenübertragung verwendet.

Vorteile

- Es sind keine baulichen Maßnahmen innerhalb eines Gebäudes nötig.
- Die baulichen Maßnahmen zwischen verschiedenen Gebäuden sind geringer als bei einer Verkabelung.
- Höhere Mobilität, da theoretisch jeder Punkt eines Firmengeländes drahtlos erreichbar ist.

Nachteile

- Oft geringere Datenübertragungsraten als bei Kabeln, die abhängig von Hindernissen sind.
- Anfällig für Störeinflüsse und Abhören durch Unbefugte.
- Probleme mit Ausleuchtung und Reflexionen.
- Bei vielen gleichzeitigen Nutzern an einem WLAN-Zugang bricht die Übertragungsrate ein (Shared Media).

Sicherheit als kritischer Bereich

Gerade im Bereich Sicherheit gibt es bei WLANs einige Punkte zu beachten, die sich auch in einem etwas größeren Konfigurationsaufwand äußern. Die sogenannte **SSID (Service Set Identifier)** kann eine eindeutige Identifikation (Firmenname etc.) enthalten, damit bei Problemen eine Kontaktaufnahme mit dem Betreiber möglich ist. Ein Verbergen bringt nicht viel, da die SSID in jedem Paket mitgeschickt wird und es Programme gibt, die auch verborgene SSIDs auslesen können. Eine versehentliche Verbindung durch Unbefugte ist bei verschlüsselten Zugängen nicht zu erwarten. Es gibt auch Empfehlungen, als SSID eine zufällige Zeichenfolge einzugeben, damit die SSID keine Rückschlüsse auf den Betreiber zulässt.

Letztendlich sollte die Übertragung im WLAN nur **verschlüsselt** erfolgen, wobei die Verschlüsselung mit **WEP (Wired Equivalent Privacy)** unsicher ist. Besser ist der Einsatz von **WPA (Wi-Fi Protected Access)** bzw. der Nachfolgetechnologie **WPA2**, da hier deutlich stärkere Verschlüsselungsmechanismen mit **AES (Advanced Encryption Standard)** verwendet werden. Zusammen mit **TKIP (Temporal Key Integrity Protocol)** sind allerdings nur max. 54 Mbit/s möglich! Höhere Raten erreicht z. B. WPA2 zusammen mit CCMP (Counter-Mode/CBC-Mac Protocol)

Grundlegende Beschreibung

Kommunikation über WLAN erfolgt entweder als Punkt-zu-Punkt- oder als Mehrpunkt-Kommunikation. Die erste Variante dient z. B. der Überwindung größerer Distanzen durch den Einsatz zweier Richtantennen.

Bei der Mehrpunkt-Kommunikation werden ein oder mehrere sogenannte Access Points eingesetzt, die im Prinzip jeweils wie Zentralen (Verteiler) fungieren und die Datenströme mehrerer Clients koordinieren.

Diese Access Points können bei größeren Installationen über Kabel und geeignete Managed Switches eine Verbindung zu einem sogenannten RADIUS (Remote Authentication Dial-In User Service)-Server erhalten, um zwischen berechtigten und nicht berechtigten Sendern zu unterscheiden (Authentifizierung).

Eine Unterscheidung über die MAC-Adresse sollte nur den zum Zugang berechtigten Geräten vorbehalten bleiben, die sich nicht per RADIUS authentifizieren können. Dabei wird in einer Zugangsliste (Access Control Table) vom Managed Switch die MAC-Adresse eingetragen. Auf einem Access-Point bringt dies keine höhere Sicherheit, da per Funk eine MAC-Adresse unverschlüsselt verschickt wird und somit gefälscht werden kann. Die kleinste Einheit ist eine sogenannte Funkzelle, womit der Bereich gemeint ist, der von einem Sender (=Access Point) abgedeckt werden kann. Er umfasst ca. 30m im Gebäude und bis zu 300m im Freien. Mit speziellen Antennen können auch mehrere Kilometer überbrückt werden.

ISM-Frequenzbänder

In den meisten Fällen wird von den Herstellern ein sogenanntes **ISM-Band (Industrial, Scientific and Medical)** verwendet. Manchmal finden Sie auch die Abkürzung ISMO, wobei der letzte Buchstabe für den Begriff „Office“ (Büro) steht. Der Einsatz dieser Frequenzbänder bietet zwei Vorteile. Sie sind

- **gebührenfrei**
- **genehmigungsfrei**

Hierin liegt aber auch gleichzeitig der Nachteil. Sie werden von sehr vielen Herstellern für die unterschiedlichsten Zwecke genutzt, wie z. B. drahtlose Lautsprecher, elektronische Türöffnung bei Autos oder Garagen, und so ist die Gefahr, dass sich Geräte gegenseitig stören, relativ hoch.

Die für WLAN wichtigsten ISM-Bänder sind

- das **2,4-GHz-Band** (2,3995 bis 2,4845 GHz) mit max. **13 überlappenden Kanälen** von 20 MHz Bandbreite; auch 40 MHz Bandbreite ist möglich, dann aber mit weit weniger nutzbaren Kanälen. Es sind Geschwindigkeiten bis zu 300MBit/s möglich.
- das **5-GHz-Band** (5,150 bis 5,350 GHz für Kanalnummer 36–64 und 5,470 bis 5,725 GHz für Kanalnummer 100–140) mit Kanälen von 20, 40, 80 oder 160 MHz Bandbreite, wobei **max. 19 Kanäle** bei 20 MHz Bandbreite nicht überlappend nutzbar sind. Beim Funken mit 40 MHz Bandbreite sind 2 dieser Kanäle gebündelt erforderlich, mit 80 MHz 4 Kanäle usw. Die **Reichweite ist geringer** als im 2,4 GHz-Band. Es sind Geschwindigkeiten bis zu 600MBit/s möglich.
- das **6-GHz-Band** mit 80- und 160-MHz-Funkkanäle (WiFi 6) bzw. 320 MHz breite Funkkanäle (WiFi 7)
- zukünftig das 60-GHz-Band (57 bis 66 GHz) mit vier 2000 MHz breiten Funkkanälen für kurze Distanzen

Frequenzbereiche für IEEE 802.11

Frequenzbereich	2,4 GHz	5 GHz	6 GHz	60 GHz
Frequenzen	2,3995 bis 2,4845 GHz	5,150 bis 5,350 GHz 5,470 bis 5,725 GHz	5,925 bis 6,425 GHz	57,0 bis 66,0 GHz
Reichweite	innerhalb eines Wohnhauses	begrenzt auf eine Wohnung oder Stockwerk	begrenzt auf eine Wohnung oder Stockwerk	begrenzt auf einen Raum
Kanalbreite	20 und 40 MHz	20, 40, 80, 160 MHz	20, 40, 80, 160 MHz	2 GHz
Nutzung	stark überfüllt	gering	zukünftig	selten

2.4 GHz Band

Im 2,4-GHz-Frequenzband existieren insgesamt 79 schmalbandige Kanäle, die in Kanäle mit je 5 MHz zusammengefasst sind. In Europa gibt es 13, in den USA 11 und in Japan 14 solcher Kanäle. Diese Kanäle sind allerdings eng aneinandergereiht und überlappen sich. Deshalb kann man nicht alle der 11, 13 oder 14 Kanäle verwenden, sondern je nach Kanal-Verteilung nur 3 oder 4. Und das bei einer Kanalbreite von 20 MHz. Bei einer Kanalbreite von 40 MHz würde sich die Anzahl parallel nutzbarer Kanäle halbieren.

Kanal	Trägerfrequenz	Frequenzbereich	Europa	USA	Japan
1	2412 MHz	2399,5 - 2424,5 MHz	×	×	×
2	2417 MHz	2404,5 - 2429,5 MHz	×	×	×
3	2422 MHz	2409,5 - 2434,5 MHz	×	×	×
4	2427 MHz	2414,5 - 2439,5 MHz	×	×	×
5	2432 MHz	2419,5 - 2444,5 MHz	×	×	×
6	2437 MHz	2424,5 - 2449,5 MHz	×	×	×
7	2442 MHz	2429,5 - 2454,5 MHz	×	×	×
8	2447 MHz	2434,5 - 2459,5 MHz	×	×	×
9	2452 MHz	2439,5 - 2464,5 MHz	×	×	×
10	2457 MHz	2444,5 - 2469,5 MHz	×	×	×
11	2462 MHz	2449,5 - 2474,5 MHz	×	×	×
12	2467 MHz	2454,5 - 2479,5 MHz	×		×
13	2472 MHz	2459,5 - 2484,5 MHz	×		×
14	2484 MHz				**× (11b)**

5 GHz Band

Das 5-GHz-Frequenzband dient als Erweiterung, um ein WLAN zu beschleunigen. Es wird allerdings nicht so oft genutzt. Die WLAN-Clients müssen dafür die entsprechende Hardware-Ausstattung mitbringen. Weltweit existiert hierfür eine Bandbreite zwischen 200 und fast 500 MHz.

In Europa werden die Frequenzen von 5,15 bis 5,35 GHz mit den Kanälen von 36 bis 64 und von Frequenzen 5,5 bis 5,7 GHz mit den Kanälen von 100 bis 140 verwendet. In den USA werden die Frequenzen von 5,15 bis 5,35 GHz mit den Kanälen von 36 bis 64 und Frequenzen von 5,5 bis 5,7 GHz mit den Kanälen von 100 bis 140, mit Ausnahme der Kanäle 120, 124 sowie 128 verwendet. Nachteilig ist, dass dieses Frequenzband weltweit nicht einheitlich geregelt ist. Nicht nur der verfügbaren Bandbreite wegen, sondern auch in der Nutzung. So gibt es mit Flug- und Wetterradar einen Primärnutzer, für den dieser Frequenzbereich reserviert ist, aber regional unterschiedlich in Verwendung ist. In der Regel in der Nähe von Flughäfen. Deshalb ist die Erweiterung Dynamic Frequency Selection (DFS) zum Schutz der Primärnutzer in der EU Pflicht. Viele Hersteller preisgünstiger WLAN-Router sparen sich die DFS-Technik und dürfen im 5-GHz-Band nur auf den Kanälen von 36 bis 48 arbeiten. Desweiteren gibt es Bestrebungen den 5-GHz-Frequenzbereich für Mobilfunk und andere Funksysteme nutzen zu dürfen.

Im Vergleich zum 2,4-GHz-Frequenzband ist die Reichweite, auch wegen geringerer Sendeleistung, geringer. Was zur Folge hat, dass die Geschwindigkeit eines WLANs bei 5 GHz nicht so schnell ist, wie erhofft.

6 Ghz Band

6 GHz Das 6-GHz-Frequenzband ist für die klassische Mobilfunknutzung ungeeignet. Und wegen dem hohen Bedarf für zusätzliche Frequenzen für die WLAN-Technik, wurde mit Wi-Fi 6E und dem Standard IEEE 802.11ax das 6-GHz-Frequenzband zur Nutzung für WLAN freigegeben. Das 6-GHz-Frequenzband ist für WLANs deshalb interessant, weil es in dicht besiedelten Gebieten mehrere 80- und 160-MHz-Funkkanäle ermöglicht, und somit das 5-GHz-Frequenzband entlastet. Desweiteren können mit der WLAN-Generation Wi-Fi 7 und dem Standard IEEE 802.11be 320 MHz breite Funkkanäle im 6-GHz-Frequenzband tatsächlich realisierbar sein.

Die Nutzung des 6-GHz-Frequenzbandes unterliegt der Allgemeinzuteilung, ist aber regional unterschiedlich geregelt.

- In den USA existiert insgesamt 1.200 MHz Bandbreite (5.925 - 7.125 MHz) für bis zu vierzehn 80 MHz Kanäle oder bis zu sieben 160 MHz Kanäle.
- In Europa existiert insgesamt 500 MHz Bandbreite (5.925 - 6.425 MHz).
- In Europa liegt aber keine alleinige Nutzung vor. Teile des Frequenzbereichs sind Primärnutzen zugeteilt. Allerdings regional begrenzt. Zum Beispiel für lizenzierte Punkt-zu-Punkt-Richtfunk-Verbindungen. Außerdem können in einigen Ländern nur zwei Kanäle mit je 160 MHz genutzt werden, da hier ein Funksystem für autonome Metro-/S-Bahn-Systeme am unteren Ende des Frequenzbandes einen dritten Kanal blockiert.

Lizenzfreie Funkdienste wie WLAN müssen als Sekundärnutzer Rücksicht auf etablierte Funksysteme nehmen. Dazu müssen WLAN-Geräte permanent nach charakteristischen Funksignalen Ausschau halten und Funkkanäle räumen, wenn sie der Primärnutzer gerade belegt. Ein Problem ist dabei, dass proprietäre Funktechniken nicht immer erkannt werden können. Einerseits weil das charakteristische Funksignal unbekannt ist oder der Empfangspegel so gering, dass ein WLAN-Gerät das fremde

Funksignal nicht erkennt. Das kann zur Folge haben, dass ein WLAN-Gerät den Frequenzbereich als frei erkennt und dann die bestehende und bevorrechtigte Funkverbindung mit dem eigenen WLAN-Signal stört.

Zum Vermeiden von Störungen gibt es Maßnamen, deren Umsetzung in der EU diskutiert wird (Stand Anfang 2021).

Für WLAN-Geräte im 6-GHz-Bereich ist eine Sendeleistung bis maximal 200 Milliwatt EIRP (inklusive Antennengewinn) in Innenräumen erlaubt (Low Power Indoor, LPI). Im Freien (Outdoor) dürfen Geräte mit maximal 25 mW EIRP senden (Very-Low-Power, VLP). WLAN-Geräte müssen täglich in einer Datenbank im Internet nachsehen, ob und mit welcher maximalen Sendeleistung der gewünschte Funkkanal am Betriebsort verfügbar ist (Automatic Frequency Coordination, AFC). Ein Primärnutzer kann bei Beeinträchtigungen ihrer Funksysteme über die AFC-Datenbank veranlassen, dass lizenzfreie Geräte ihre Sendeleistung senken müssen. Kann ein Gerät die AFC-Datenbank nicht erreichen, muss es den Betrieb im 6-GHz-Frequenzbereich sofort einstellen oder die Sendeleistung in Innenräumen auf LPI-Niveau senken. Die Nutzung von 6 GHz hat folgende Vorteile:

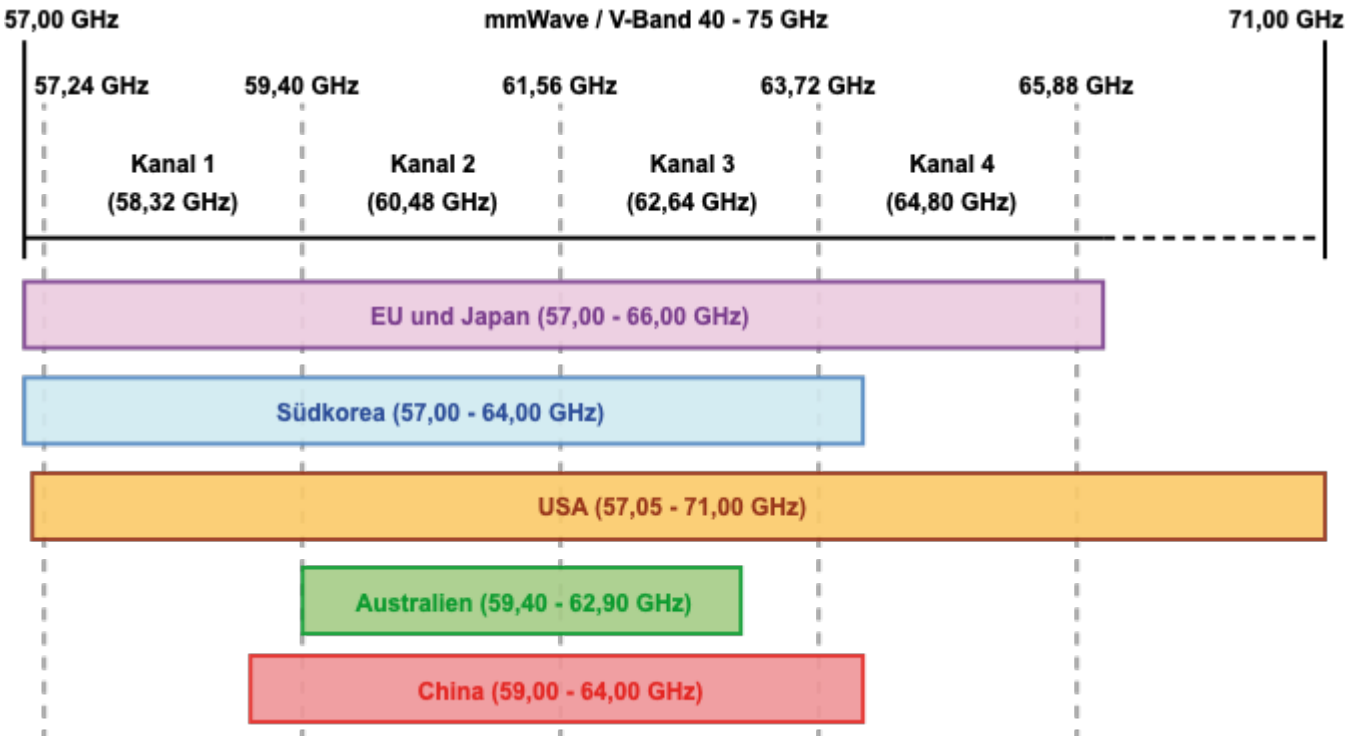
- ein großes und zusammenhängendes Frequenzspektrum
- 160 MHz breite Kanäle
- geringere Interferenzen
- hohe Geschwindigkeit
- Latenzzeit beträgt weniger als 2 Millisekunden
- hohe Kapazität für viele Teilnehmer

Der Frequenzbereich ist nützlich, wenn viele User parallel das gleiche Netz nutzen oder viele WLAN-Netze parallel auf engem Raum betrieben werden. Allerdings ist nicht jede WLAN-Hardware für 6 GHz geeignet. Dafür sind WLAN-Basisstationen und WLAN-Clients mit „Wi-Fi 6E“ notwendig (IEEE 802.11ax).

Zur Reichweite eines 6-GHz-WLANs kann man sagen, dass diese im Vergleich zu einem 5-GHz-WLAN etwas geringer ausfällt. Das liegt aber nicht an der regulatorisch begrenzten Sendeleistung, sondern daran, dass die entsprechende Verstärker-Elektronik teuer ist, Platz braucht und viel Strom verbraucht. Das lässt sich in hochpreisigen Basisstationen realisieren, aber nicht im günstigen Endkundenbereich und schon gar nicht in mobilen Geräten.

60 GHz Band

Das 60-GHz-Band erstreckt sich von 57 bis 66 GHz (EU) und hat einen rund 7 GHz breiten Funkkanal. Dieser wird in vier einzelne Kanäle mit einer Bandbreite von 1.760 MHz unterteilt, die in Europa lizenzfrei nutzbar sind.



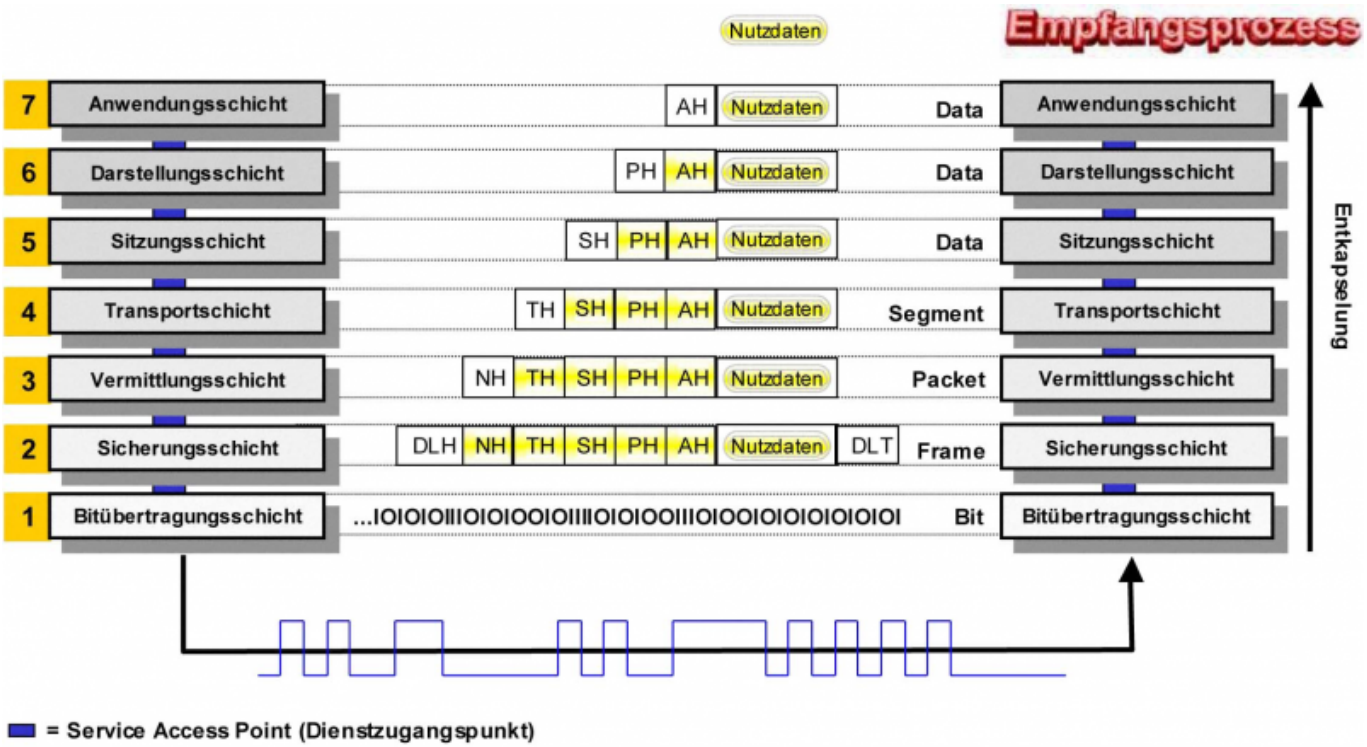
Der Nachteil von 60 GHz ist die Streckendämpfung für das Funksignal. Bei dieser Frequenz erreicht die Absorption durch den atmosphärischen Sauerstoff rund 20 dB pro Kilometer (dB/km). Um genauer zu sein, der Sauerstoff erreicht hier sein Absorbationsmaximum. Die hochfrequenten Signale haben eine sehr begrenzte Reichweite, die so gut wie nicht durch Zimmerwände dringen. Hohe Geschwindigkeiten erreicht man damit in der Regel nur auf ein paar Meter. Am besten nur wenige Zentimeter und mit Sichtkontakt. Und damit ist ein WLAN bei 60 GHz ein reiner Zimmer-Funker.

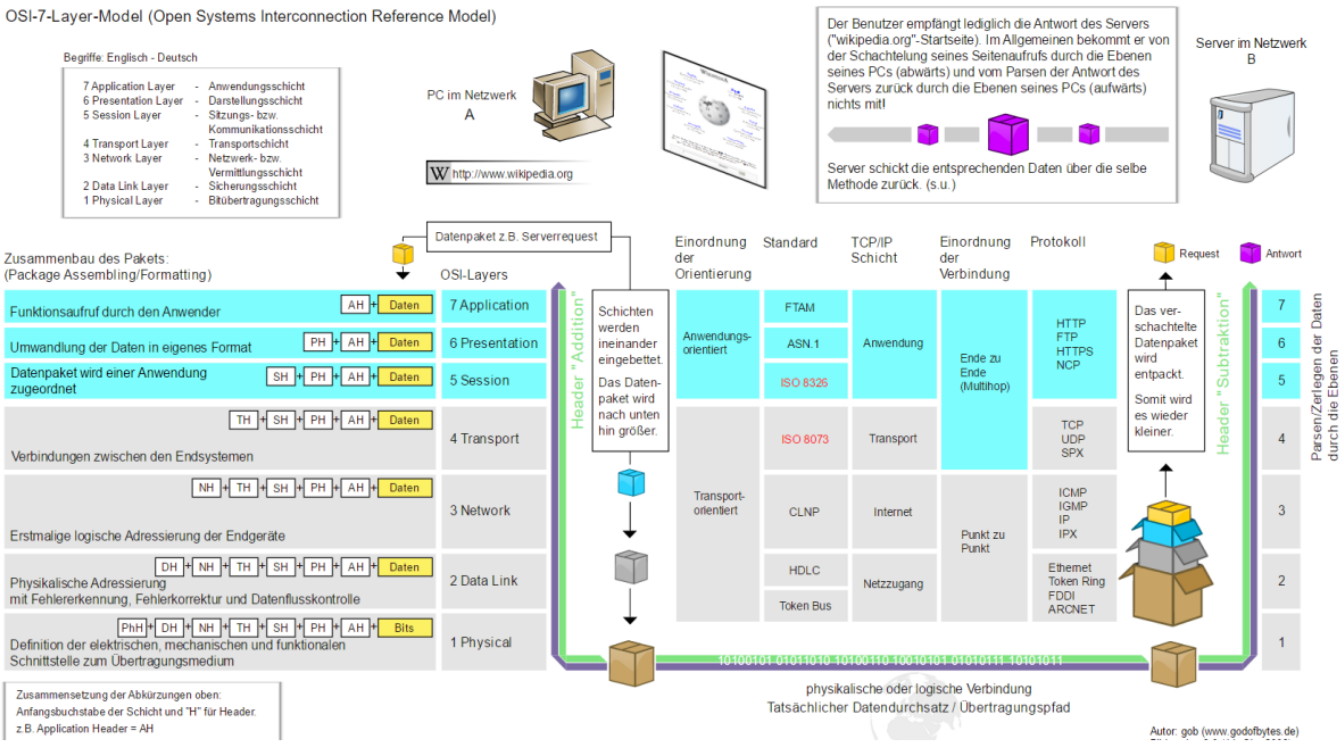
OSI - Schichtenmodell

Das OSI-7-Schichtenmodell ist ein Referenzmodell für herstellerunabhängige Kommunikationssysteme bzw. eine Design-Grundlage für Kommunikationsprotokolle und Computernetze. OSI steht für Open System Interconnection (Offenes System für Kommunikationsverbindungen) und wurde von der ISO (International Organization for Standardization), das ist die Internationale Organisation für Normung, als Grundlage für die Bildung von offenen Kommunikationsstandards entworfen. Bei allen ISO-Standards handelt es sich um Handlungsempfehlungen. Die Einhaltung einer ISO-Norm ist freiwillig. In der Regel wird die Einhaltung der ISO-Standards von verschiedenen Seiten, zum Beispiel Kooperationspartnern, Herstellern und Kunden, gefordert.

Das Modell

(Offenes System für Kommunikationsverbindungen) und wurde von der ISO (International Organization for Standardization), das ist die Internationale Organisation für Normung, als Grundlage für die Bildung von offenen Kommunikationsstandards entworfen. Bei allen ISO-Standards handelt es sich um Handlungsempfehlungen. Die Einhaltung einer ISO-Norm ist freiwillig. In der Regel wird die Einhaltung der ISO-Standards von verschiedenen Seiten, zum Beispiel Kooperationspartnern, Herstellern und Kunden, gefordert.





Die Schichten im Detail

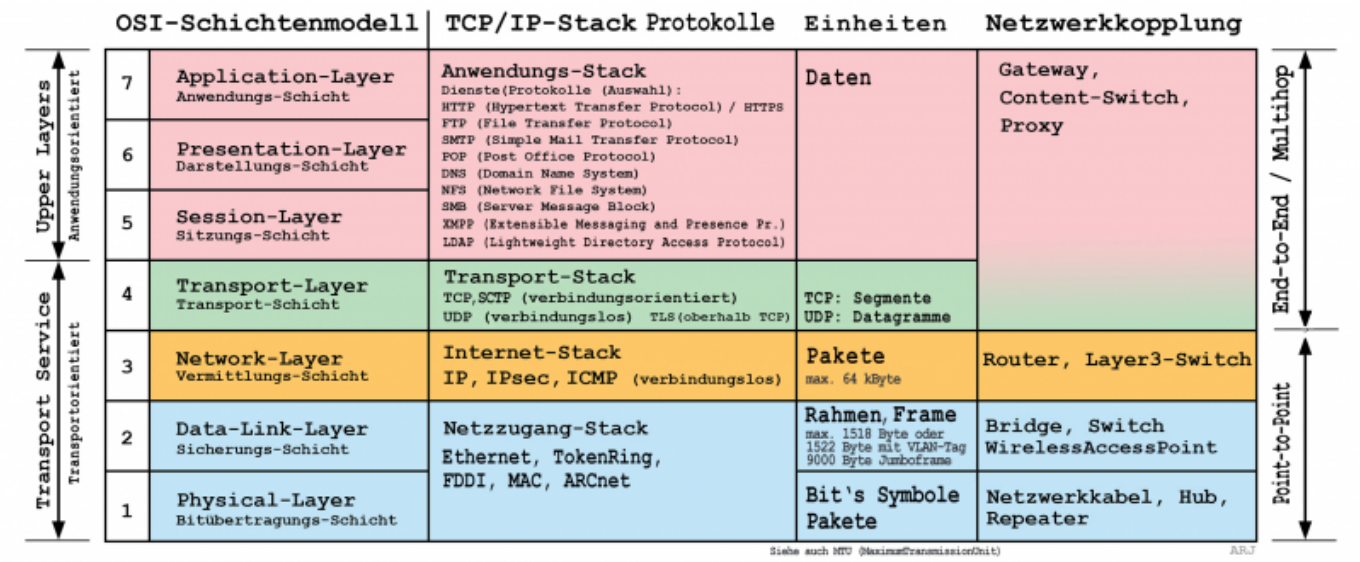
Application Layer (Anwendungsschicht)	**Benutzerschnittstelle, Dienste, Anwendungen und Netzmanagement**
Schicht 7	Die Anwendungsschicht stellt Funktionen für die Anwendungen zur Verfügung. Diese Schicht stellt die Verbindung zu den unteren Schichten her. Auf dieser Ebene findet auch die Dateneingabe und -ausgabe statt.

Application Layer (Anwendungsschicht)	**Benutzerschnittstelle, Dienste, Anwendungen und Netzmanagement**
Presentation Layer (Darstellungsschicht)	**Übersetzung, Verschlüsselung , Kompression in Standardformate**
Schicht 6	Die Darstellungsschicht setzt die Daten der Anwendungsebene in ein Zwischenformat um. Diese Schicht ist auch für Sicherheitsfragen zuständig. Durch sie werden Dienste zur Verschlüsselung von Daten bereitgestellt und gegebenenfalls Daten komprimiert.
Session Layer (Sitzungsschicht)	**Erstellung einer Verbindung, Freigabe von Verbindungen, Dialogsteuerung**
Schicht 5	Diese Schicht ermöglicht zwei Anwendungen auf verschiedenen Computern, eine gemeinsame Sitzung aufzubauen, damit zu arbeiten und sie zu beenden. Sie übernimmt ebenfalls die Dialogsteuerung zwischen den beiden Computern einer Sitzung und regelt, welcher der beiden wann und wie lange Daten überträgt.
Transport Layer (Transportschicht)	**Logische Ende-zu-Ende-Verbindungen (Transportkontrolle, Paketbildung)**
Schicht 4	Die Transportschicht stellt die zuverlässige Auslieferung der Nachrichten sicher und erkennt sowie behebt allfällige Fehler. Sie ordnet bei Bedarf auch die Nachrichten in Paketen neu, indem sie lange Nachrichten zur Datenübertragung in kleinere Pakete aufteilt. Am Ende des Weges stellt sie die kleinen Pakete wieder zur ursprünglichen Nachricht zusammen. Die empfangene Transportebene sendet auch eine Empfangs bestätigung.
Network Layer (Vermittlungsschicht)	**Routing (Internet), Datenflusskontrolle, Adressierung**
Schicht 3	Die Vermittlungsschicht steuert die zeitliche und logische getrennte Kommunikation zwischen den Endgeräten, unabhängig vom Übertragungsmedium und der Topologie. Auf dieser Schicht erfolgt erstmals die logische Adressierung der Endgeräte. Die Adressierung ist eng mit dem Routing (Wegfindung vom Sender zum Empfänger) verbunden.
Data Link Layer (Sicherungsschicht)	**Logische Verbindungen mit Datenpaketen und elementare Fehlererkennungsmechanismen**
Schicht 2	Die Sicherungsschicht sorgt für eine zuverlässige und funktionierende Verbindung zwischen Endgerät und Übertragungsmedium. Zur Vermeidung von Übertragungsfehlern und Datenverlust enthält diese Schicht Funktionen zur Fehlererkennung, Fehlerbehebung und Datenflusskontrolle. Auf dieser Schicht findet auch die physikalische Adressierung von Datenpaketen statt.
Physical Layer (Physikalische Schicht)	**Maßnahmen und Verfahren zur Übertragung von Bitfolgen**
Schicht 1	Die Bitübertragungsschicht definiert die elektrische, mechanische und funktionale Schnittstelle zum Übertragungsmedium. Die Protokolle dieser Schicht unterscheiden sich nur nach dem eingesetzten Übertragungsmedium und -verfahren. Das Übertragungsmedium ist jedoch kein Bestandteil der Schicht 1.

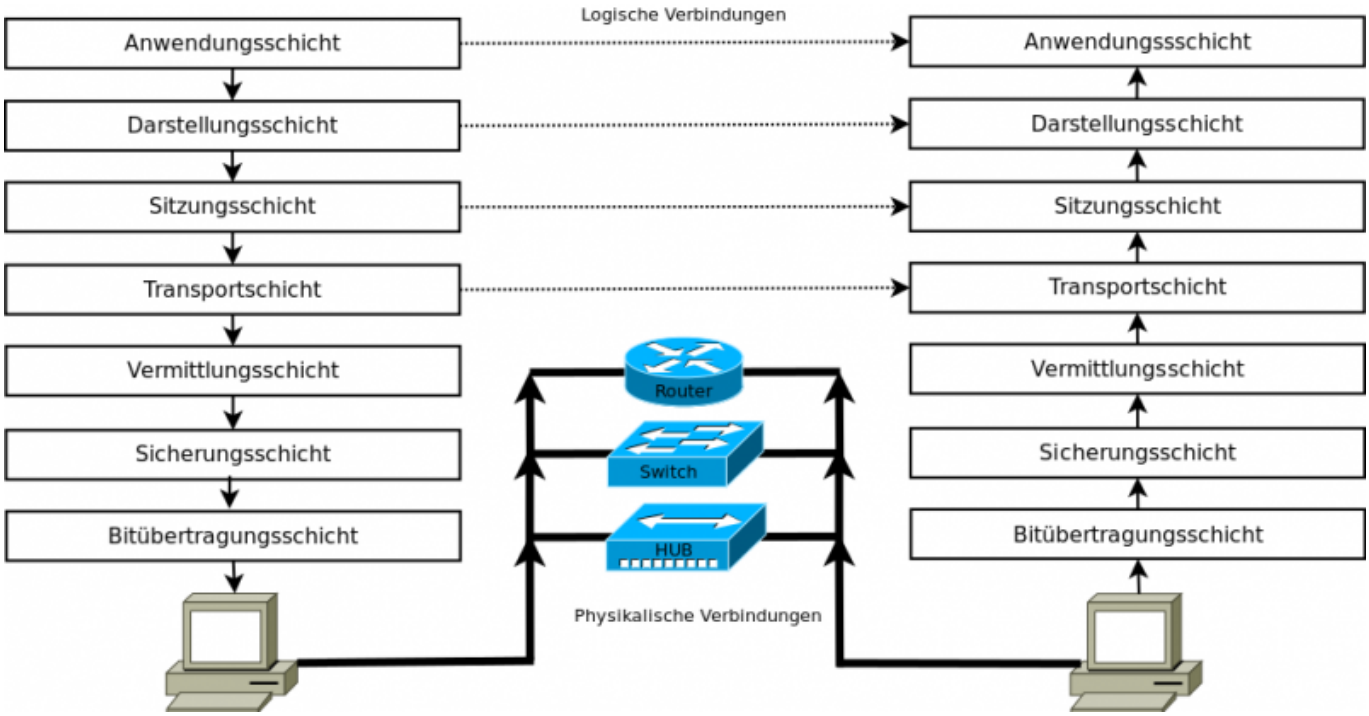
Protokolle

Protokolle sind eine **Sammlung von Regeln zur Kommunikation** auf einer bestimmten Schicht des OSI-Schichtenmodells. Die Protokolle einer Schicht sind zu den Protokollen der über- und untergeordneten Schichten weitestgehend transparent, so dass die Verhaltensweise eines Protokolls sich wie bei einer direkten Kommunikation mit dem Gegenstück auf der Gegenseite darstellt.

Die **Übergänge zwischen den Schichten sind Schnittstellen**, die von den Protokollen verstanden werden müssen. Weil manche Protokolle für ganz bestimmte Anwendungen entwickelt wurden, kommt es auch vor, dass sich **Protokolle über mehrere Schichten** erstrecken und mehrere Aufgaben abdecken. Dabei kommt es vor, dass in manchen Verbindungen einzelne Aufgaben in mehreren Schichten und somit mehrfach ausgeführt werden.



Anmerkung zur Skizze: End-zu-End-Verbindungen finden laut Definition erst ab Schicht 4 statt.



Das wichtigste Protokoll im Netzwerkverkehr, ist das **TCP & IP Protokoll** (Transmission Control Protocol & Internet Protocol), welche sich heute als Standard durchgesetzt haben.

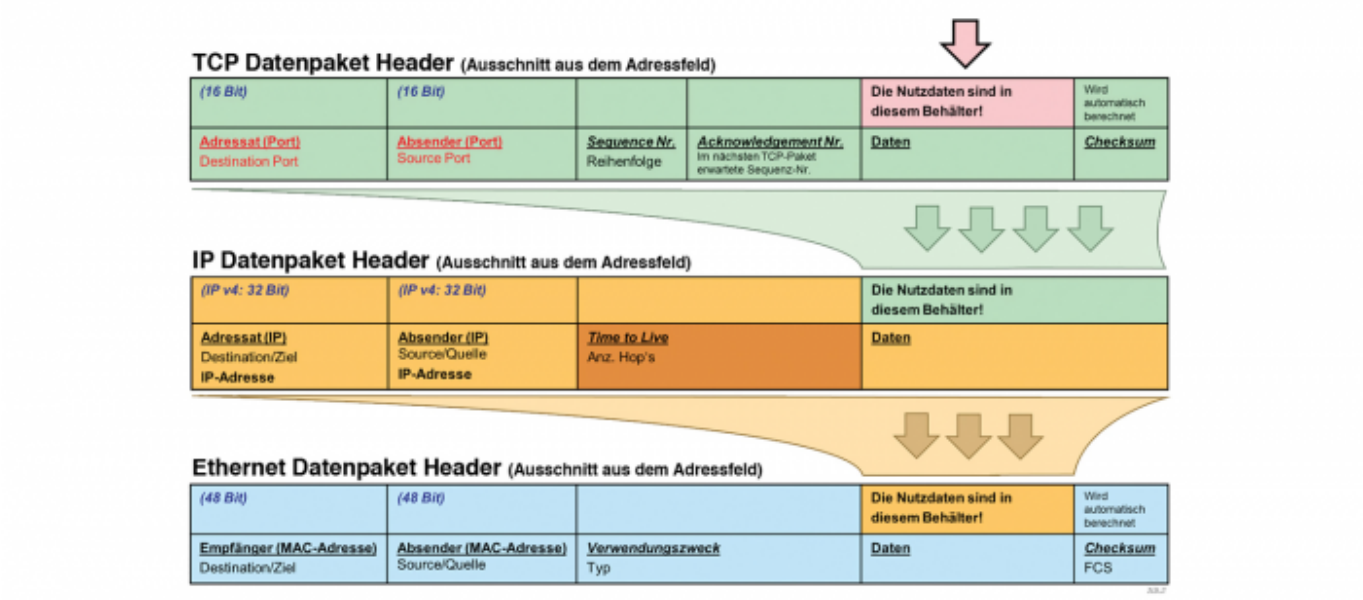
Neben TCP und IP gibt es natürlich noch viele weitere Protokolle, wie in der obigen Abbildung zu sehen ist.

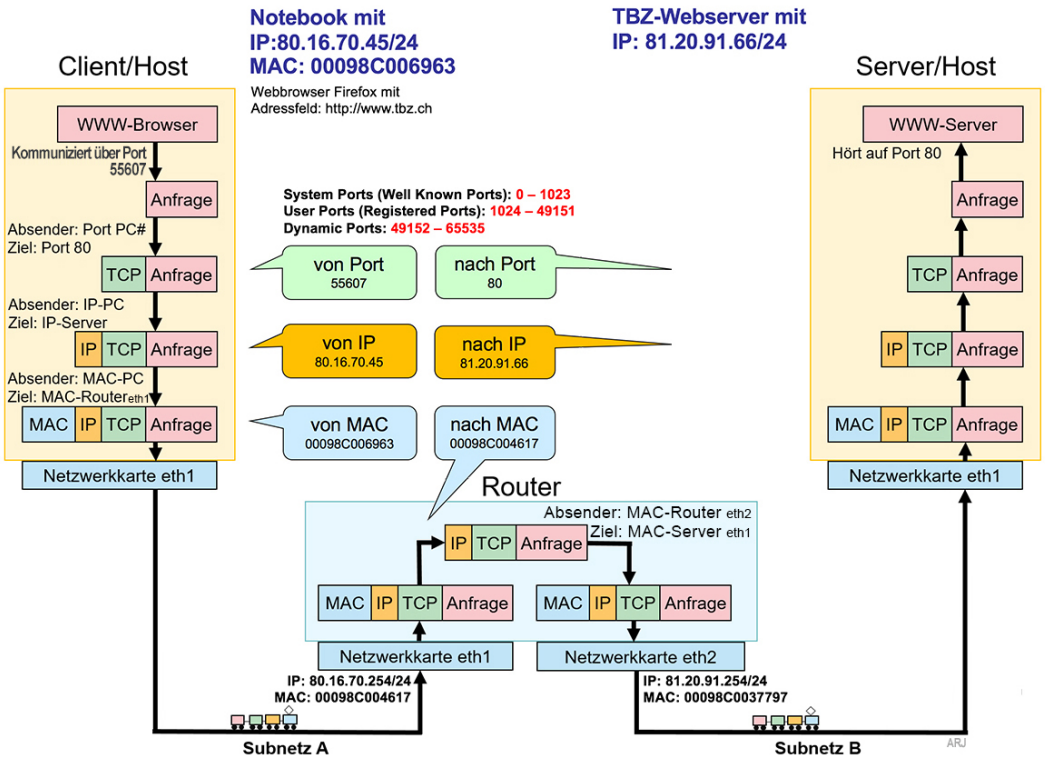
Die folgende Tabelle listet einige dieser Protokolle auf und ordnet sie ins obige Modell ein:

Protokoll	**Schicht**	**Name**	**Beschreibung**
FTP	5	File Transfer Protocol	Datenaustausch zwischen Rechnern
Telnet	5	Telecommunication Network Protocol	Terminalemulation zur Host-Kommunikation
SMTP	5	Simple Mail Transfer Protocol	Versenden von EMails
HTTP	5	Hypertext Transfer Protocol	Übertragen von HTML-Seiten
POP	5	Post Office Protocol	Abrufen von E-Mails
TCP	4	Transmission Control Protocol	Aufbau logischer Verbindungen zwischen Applikationen
UDP	4	User Datagram Protocol	Verbindungsloses Übertragungsprotokoll. Es ist nicht so gesichert wie TCP dafür aber schneller
IP	3	Internet Protocol	Verbindungsloses Protokoll zur Paketlenkung und Paketvermittlung über IP-Adressen
IPSec	3	IP Secure	Erweitert das reguläre IP-Protokoll um ein Bündel von Sicherheitsmechanismen
ARP	3	Address Resolution Protocol	Dien dazu logische IP-Adressen physikalischen MAC-Adressen zuzuordnen

Datenkapselung

Unter der Datenkapselung versteht man den Prozess im OSI-Modell und im TCP/IP-Referenzmodell, der die zu versendenden Daten im Header (und ggf. Trailer) der jeweiligen Schichten ergänzt. Im OSI-Modell betrifft dies die Datagramme der Schichten 2 bis 4, die gekapselt (verpackt) werden.





OSI-Schichtenmodell

TCP/IP-Referenzmodell

https://edu.juergarnold.ch/fach_it/netzwerktheorie/article1.html?q=osi#nw_3_19

Ethernet

Ethernet ist eine Technik, die **Software (Protokolle usw.) und Hardware (Kabel, Verteiler, Netzwerkkarten usw.) für kabelgebundene Datennetze spezifiziert**, welche ursprünglich für lokale Datennetze (LANs) gedacht war und daher auch als LAN-Technik bezeichnet wird. Sie ermöglicht den Datenaustausch in Form von Datenframes zwischen den in einem lokalen Netz (LAN) angeschlossenen Geräten (Computer, Drucker und dergleichen). Derzeit sind Übertragungsraten von 1, 10, 100 Megabit/s (Fast Ethernet), 1000 Megabit/s (Gigabit-Ethernet), 2,5, 5, 10, 40, 50, 100, 200 und 400 Gigabit/s spezifiziert. In seiner ursprünglichen Form erstreckt sich das LAN dabei nur über ein Gebäude; Ethernet-Varianten über Glasfaser haben eine Reichweite von bis zu 70 km.

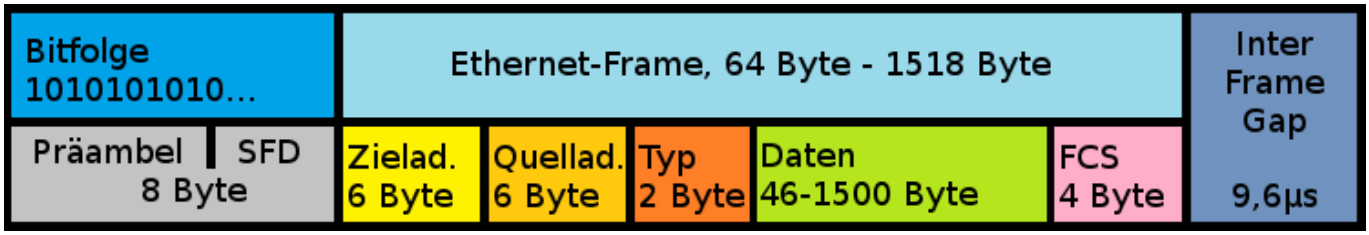
Die Ethernet-Protokolle umfassen Festlegungen für Kabeltypen und Stecker sowie für Übertragungsformen (Signale auf der Bitübertragungsschicht, Paketformate). Im OSI-Modell ist mit Ethernet sowohl die **physische Schicht (OSI Layer 1)** als auch die **Data-Link-Schicht (OSI Layer 2)** festgelegt.

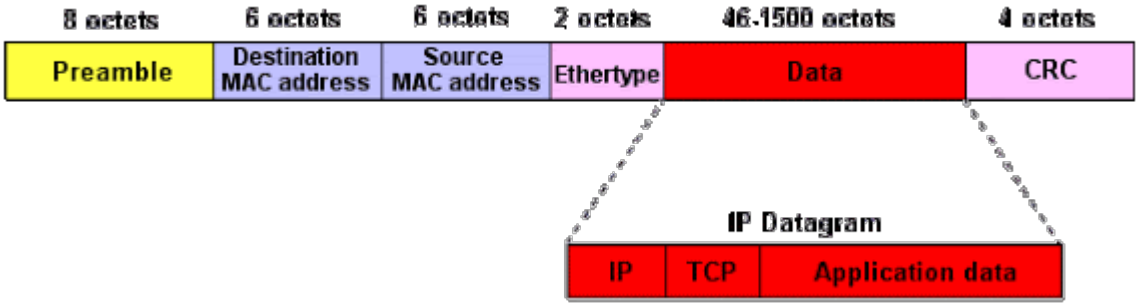
Ethernet basiert auf der Idee, dass die Teilnehmer eines LANs Nachrichten durch Hochfrequenz übertragen, allerdings **nur innerhalb eines gemeinsamen Leitungsnetzes**. Jede Netzwerkschnittstelle hat **einen global eindeutigen 48-Bit-Schlüssel**, der als **MAC-Adresse** (=Media-Access-Control-Adress) bezeichnet wird. Das stellt sicher, dass alle Systeme in einem Ethernet unterschiedliche Adressen haben.

Ethernet Frame

Bei der Übertragung von Daten über Ethernet ist das Ethernet-Frame hauptverantwortlich für die korrekte Regelsetzung und erfolgreiche Übermittlung von Datenpaketen. Versendete Daten über Ethernet werden vom Frame sozusagen getragen. Ein Ethernet-Frame ist zwischen 64 Byte und 1518 Byte groß, abhängig von der Größe der zu transportierenden Daten.

Im OSI-Modell befindet sich der Frame auf der Sicherungsschicht, die für die fehlerfreie Übertragung verantwortlich ist und trennt den Bitdatenstrom in Blöcke bzw. Frames auf.





Ein Ethernet-Frame muss standardmäßig mindestens 64 Byte groß sein, damit die Kollisionserkennung funktioniert, und kann maximal 1.518 Byte groß sein. Das Paket beginnt immer mit einer Präambel, die die Synchronisation zwischen Sender und Empfänger regelt und einem „Start Frame Delimiter“ (SFD), der das Frame definiert. Beide Informationen sind eine Bitfolge im Format 10101010… Im eigentlichen Frame finden sich Informationen zu Ziel- und Quelladressen (MAC-Format) und Steuerinformationen (im Fall von Ethernet II das Type-Field, später eine Längenangabe), dann folgt der zu übermittelnde Datensatz. Eine „Frame Check Sequence“ (FCS) schließt als Prüfsumme das gesamte Frame (ausgenommen Präambel und SFD). Das Paket wird von einem „Inter Frame Gap“ abgeschlossen, der eine 9,6 µs lange Sendepause festlegt.

Ethernet II benutzt die klassische Framestruktur, die das sogenannte Type-Field („Typ“) beinhaltet, womit verschiedene Protokolle der Vermittlungsschicht definiert werden. Im OSI-Modell ist die Vermittlungsschicht (auch „Network Layer“) wichtig für die Schaltung von Verbindungen und die Bereitstellung von Netzwerkadressen. Das Type-Field wurde in späteren Frame-Formaten durch eine Längenangabe ersetzt.

CSMA/CD

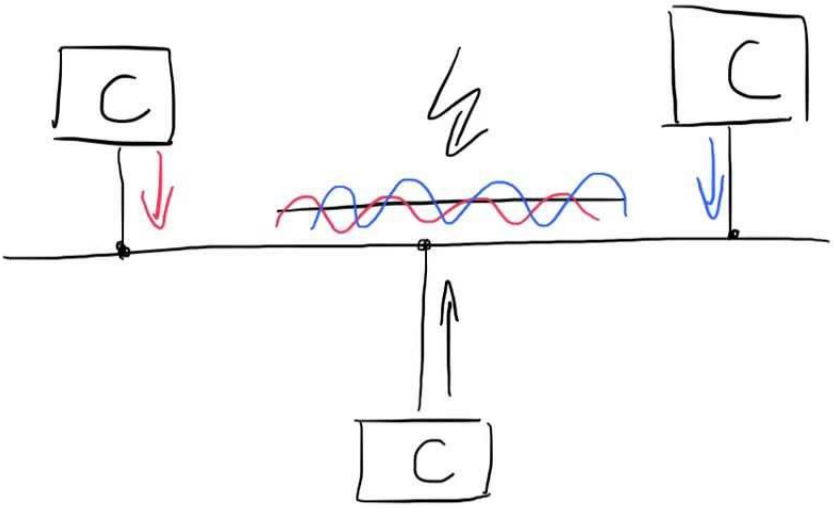
CSMA/CD (Carrier Sense Multiple Access with Collision Detection) ist das Zugriffsverfahren des Ethernets. Die Grundidee dabei ist, dass jede Station zu senden beginnen kann, wann sie will. Die einzelnen Stationen haben jederzeit und konkurrierend Zugang (Multiple Access) zum gemeinsamen Übertragungsmedium. Das grundlegende Motto könnte damit lauten:

Jeder darf, wann er will

Eingesetzt wird dieses Verfahren bei logischen Bus-Topologien. Dabei ist egal, ob physikalisch eine Bus- oder eine Stern-Topologie vorliegt.

Vorgehen

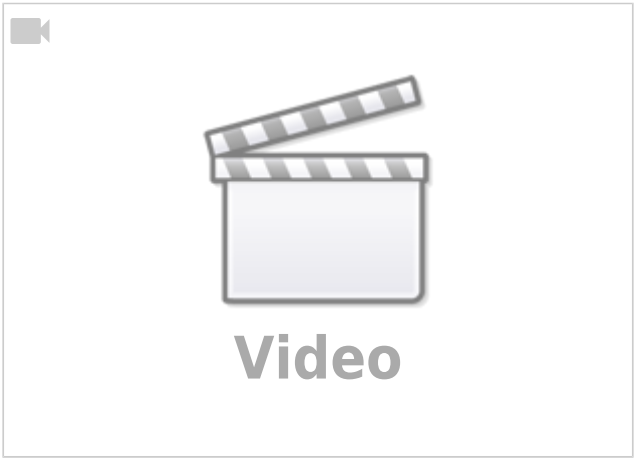
Durch Regelungen wird versucht, das Risiko zu minimieren, dass zwei Stationen ungünstigerweise gleichzeitig zu senden beginnen und somit die Signale auf dem Übertragungsweg zerstört/gestört werden **(=Kollision)**.



1) Leitung prüfen (Carrier Sense)

Der erste Teil der Abkürzung steht für Kollisionsverhinderung. Dabei wird vor einer geplanten Sendung das Übertragungsmedium abgehört, ob dieses frei ist. Wenn das Medium frei ist, wird gesendet. **–> LISTEN BEFORE TALKING**

2) Erkennen von Kollisionen (Collision Detection) Kommt es trotzdem zu einer Kollision, weil zwei Stationen gleichzeitig zu senden beginnen (**Multiple Access**). dann muss diese Kollision erkannt (**Collision Detection**) und reagiert werden. Dabei müssen alle Stationen immer am Medium horchen, ob eine Kollision auftritt. Ist dies der Fall, so sendet die erste Station, die eine Kollision erkennt, ein sogenanntes JAM-Signal aus. Jede Station, die das JAM-Signal registriert, stoppt unmittelbar das Senden von Daten. Nach einer zufälligen Zeitspanne, wird das Medium wieder überprüft und anschließend wieder begonnen zu senden.



Frage: Wie kann bei einer physikalischen Stern-Topologie eine Kollision auftreten?

Vorteile

- Jeder kann zu jeder Zeit senden

Nachteile

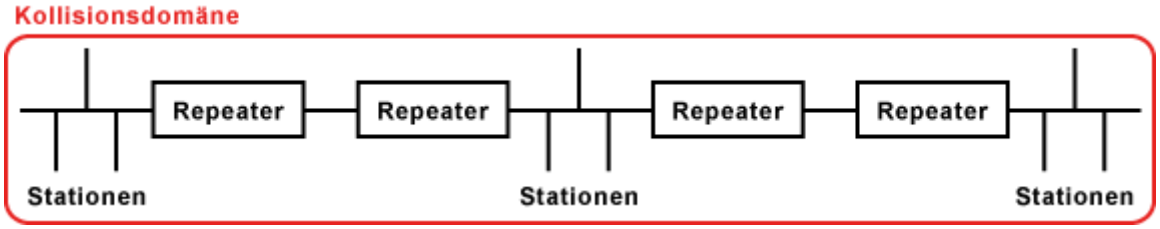
- Je mehr Stationen in einer Kollisionsdomäne, desto häufiger treten Kollisionen auf
- Der Zeitpunkt einer Sendung ist zufällig
- Das Verfahren ist ungeeignet für zeitkritischen Anwendungen

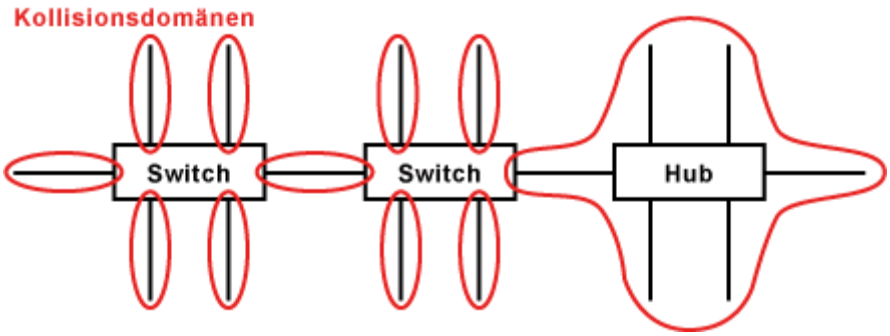
Kollisionsdomäne

Mit dem Begriff Kollisionsdomäne wird in einem Computernetz ein Teilbereich aus Teilnehmerstationen in derselben OSI-Modell-Schicht 1 bezeichnet. Eine Kollisionsdomäne umfasst alle Netzwerkgeräte, die um den Zugriff auf ein gemeinsames Übertragungsmedium konkurrieren. Das Übertragungsmedium ist daher eine zwischen allen Netzstationen geteilte Ressource. Grundlegende Vorstellung dabei ist, dass alle Netzwerkteilnehmer die Chance zur gleichberechtigten Nutzung des Netzwerkes besitzen.

Bei einem gemeinsamen Medium kann zu einer bestimmten Zeit nur jeweils eine Station Informationen übertragen, die an alle anderen Stationen übertragen bzw. von diesen empfangen wird. Fangen in einem derartigen gemeinsamen Schicht-1-Segment zwei Stationen gleichzeitig an zu senden, kommt es zu Kollisionen.

Beispiele für Kollisionsdomänen:





Netzwerkkomponenten

In der Netzwerktechnik unterscheidet man zwischen aktiven und passiven Netzwerk-Komponenten. Während **aktive Netzwerk-Komponenten eine eigene Logik haben**, zählen die passiven Netzwerk-Komponenten zur fest installierten Netzwerk-Infrastruktur. In der Regel dienen Netzwerk-Komponenten zur Kopplung der Netzwerk-Stationen. Man spricht deshalb auch von Kopplungselementen.

Passive Netzwerk-Komponenten

- Patchkabel und Installationskabel
- Anschlussdose
- Steckverbinder
- Patchfeld / Patchpanel
- Netzwerk-Schrank / Patch-Schrank

Hinweis: Zu den passiven Netzwerk-Komponenten zählen die Bestandteile der Verkabelung. Diese ist im OSI-Schichtenmodell nicht definiert.

Aktive Netzwerk-Komponenten

In kleinen privaten Netzwerken, haben Netzwerk-Komponenten noch klare Bezeichnung, wie Switch oder Router. In großen Unternehmensnetzwerken ist die Benennung der Kopplungselemente nicht immer eindeutig.

- Netzwerkkarte
- Repeater
- Hub
- Bridge
- Switch
- Router
- Gateway
- Server

Netzwerkkarte

Eine Netzwerkkarte wird auch als Netzwerkadapter bezeichnet. Die englische Bezeichnung ist Network Interface Card (NIC). Eine Netzwerkkarte ermöglicht es, auf ein Netzwerk zuzugreifen und arbeitet auf der Bitübertragungsschicht (Schicht 1) und der Datensicherungsschicht (Schicht 2) des OSI-Schichtenmodells. Jede Netzwerkkarte hat eine Hardware-Adresse (Format: XX-XX-XX-XX-XX-XX), die es auf der Welt nur einmal gibt. Anhand dieser Adresse lässt sich eine Station auf der Bitübertragungsschicht adressieren.

Im Falle von Ethernet-Netzen besteht die **MAC-Adresse aus 48 Bit (sechs Bytes)**. Die Adressen werden in der Regel **hexadezimal** geschrieben. Üblich ist dabei eine **byteweise Schreibweise**,

wobei die einzelnen Bytes durch Bindestriche oder Doppelpunkte voneinander getrennt werden, z. B. 00-80-41-ae-fd-7e oder 00:80:41:ae:fd:7e. Seltener zu finden sind Angaben wie 008041aefd7e oder 0080.41ae.fd7

In den **ersten 24 Bits** (Bit 3 bis 24) wird eine von der IEEE vergebene **Herstellerkennung** (auch OUI – Organizationally Unique Identifier genannt) beschrieben, die weitgehend in einer Datenbank einsehbar sind[6]. Die **verbleibenden 24 Bits** (Bit 25 bis 48) werden **vom jeweiligen Hersteller** für jede Schnittstelle individuell **festgelegt**.

Repeater

Ein Repeater ist ein Kopplungselement, um die Übertragungsstrecke innerhalb von Netzwerken, zum Beispiel Ethernet, zu verlängern. Ein Repeater empfängt ein Signal und bereitet es neu auf. Danach sendet er es weiter. Auf diese Weise verlängert der Repeater die Übertragungsstrecke und räumliche Ausdehnung des Netzwerks. Im einfachsten Fall hat ein Repeater zwei Ports, die wechselweise als Ein- und Ausgang funktionieren (bidirektional).

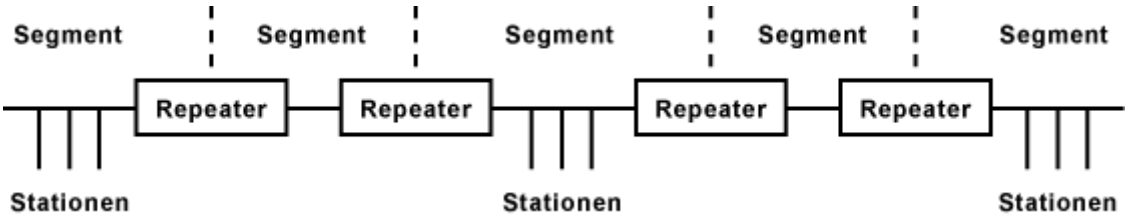
Repeater versteht man in der Regel als Verstärker von Übertragungsstrecken. Die weitere Beschreibung bezieht sich auf Repeater in kabelgebundenen Netzwerken, speziell in Ethernet-Netzwerken.

Ein Repeater arbeitet auf der Schicht 1, der Bitübertragungsschicht des OSI-Schichtenmodells. Der Repeater übernimmt keinerlei regulierende Funktion in einem Netzwerk. Er kann nur Signale empfangen und weiterleiten. Für angeschlossene Geräte ist nicht erkennbar, ob sie an einem Repeater angeschlossen sind. Er verhält sich völlig transparent.

Ein Repeater erweitert somit eine Kollisionsdomäne!!

Ein Repeater mit mehreren Ports wird auch als Hub (Multiport-Repeater) bezeichnet. Er kann mehrere Netzwerk-Segmente miteinander verbinden.

Die Repeater-Regel (5-4-3)



Um ein großes Netzwerk mit einer möglichst großen Reichweite aufzubauen, können mehrere Repeater hintereinandergeschaltet werden. Allerdings, nicht in beliebiger Anzahl. Der Grund liegt im Laufzeitverhalten und der Phasenverschiebung zwischen den Signalen an den Enden des Netzwerks. Deshalb gilt folgende Repeater-Regel:

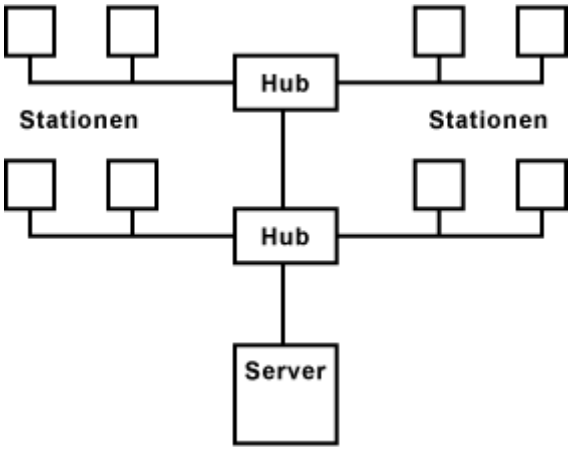
Es dürfen nicht mehr als fünf (5) Kabelsegmente verbunden werden. Dafür werden vier (4) Repeater eingesetzt. An nur drei (3) Segmenten dürfen Endstationen angeschlossen werden.

Diese **Repeater-Regel hat nur in den Ethernet-Netzwerken 10Base2 und 10BASE5** eine

Bedeutung. In Netzwerken, die mit Switches und Router aufgebaut sind, hat diese Repeater-Regel keine Bedeutung. Um die Nachteile von Repeatern in Ethernet-Netzwerken zu umgehen, werden generell Switches zur Kopplung der Hosts eingesetzt. In großen Netzwerken, insbesondere über unterschiedliche Übertragungssysteme hinweg, werden zusätzlich Router eingesetzt.

Hub

Ein Hub ist ein Kopplungselement, das mehrere Hosts in einem Netzwerk miteinander verbindet. In einem Ethernet-Netzwerk, das auf der Stern-Topologie basiert dient ein Hub als Verteiler für die Datenpakete. Hubs arbeiten auf der Bitübertragungsschicht (Schicht 1) des OSI-Schichtenmodells und sind damit auf die **reine Verteilfunktion** beschränkt. **Hubs erweitern somit die Kollisionsdomäne**. Ein Hub nimmt **ein Datenpaket entgegen und sendet es an alle anderen Ports** weiter. Das bedeutet, er **broadcastet**. Dadurch sind nicht nur **alle Ports**, sondern **auch alle Hosts belegt**. Sie bekommen alle Datenpakete zugeschickt, auch wenn sie nicht die Empfänger sind. Für die Hosts bedeutet das auch, dass sie nur dann senden können, wenn der Hub gerade keine Datenpakete sendet. Sonst kommt es zu Kollisionen.



Wenn die Anzahl der Anschlüsse an einem Hub für die Anzahl der Hosts nicht ausreicht, dann benötigt man noch einen zweiten Hub. Zwei Hubs werden über einen Uplink-Port eines der beiden Hubs oder mit einem Crossover-Kabel (Sende- und Empfangsleitungen sind gekreuzt) verbunden. Es gibt auch spezielle „stackable“ Hubs, die sich herstellerspezifisch mit Buskabeln kaskadieren lassen. Durch die Verbindung mehrerer Hubs lässt sich die Anzahl der möglichen Hosts im Netzwerk erhöhen. Allerdings ist die Anzahl der anschließbaren Hosts begrenzt. Hier gilt die Repeater-Regel.

Nachteile

- ineffizient
- unsicher (Jeder bekommt jede Nachricht)

Vorteile

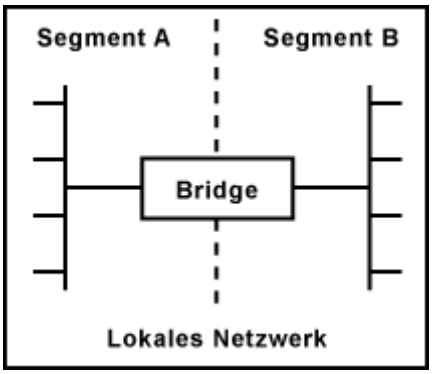
- zentraler Verteiler

Bridge

Eine Bridge ist ein Kopplungselement, das ein lokales Netzwerk in zwei Segmente aufteilt. Dabei

werden die Nachteile von Ethernet, die besonders bei großen Netzwerken auftreten ausgeglichen. Als Kopplungselement ist die Bridge eher untypisch. Man vermeidet die Einschränkungen durch Ethernet heute eher durch Switches.

Eine Bridge teilt eine Kollisionsdomäne!!



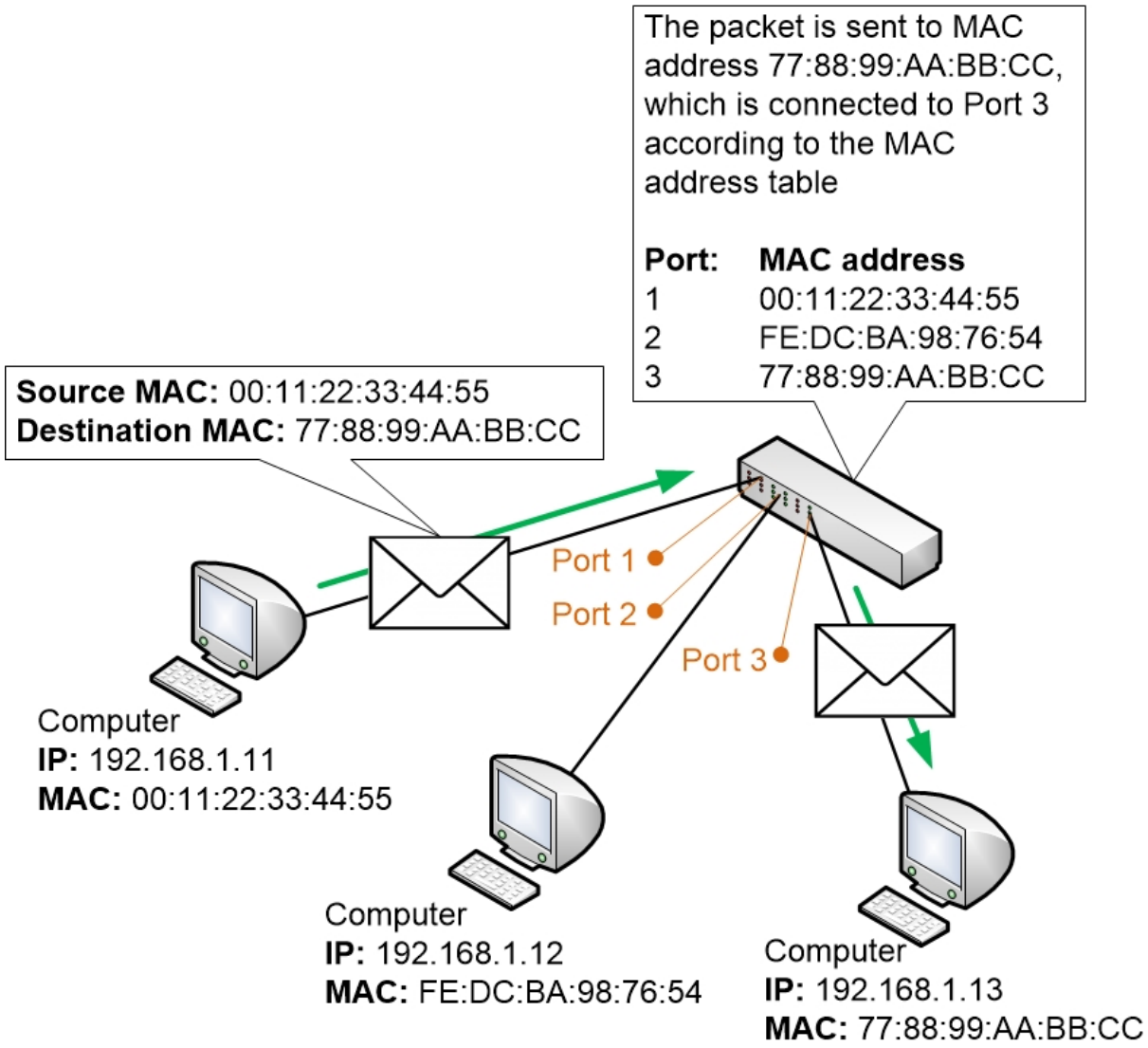
Switch

Ein Switch ist ein Kopplungselement, das mehrere Stationen in einem Netzwerk miteinander verbindet. In einem Ethernet-Netzwerk, das auf der Stern-Topologie basiert dient ein Switch als Verteiler für die Datenübertragung.

Die Funktion ist ähnlich einem Hub, mit dem Unterschied, das ein Switch direkte Verbindungen zwischen den angeschlossenen Geräten schalten kann, sofern ihm die Ports der Datenpaket-Empfänger bekannt sind. Somit kommunizieren wirklich nur jene miteinander, die auch miteinander reduzieren wollen. Wenn nicht, dann broadcastet der Switch die Datenpakete an alle Ports. Wenn die Antwortpakete von den Empfängern zurück kommen, dann merkt sich der Switch die MAC-Adressen

der Datenpakete und den dazugehörigen Port und sendet die Datenpakete dann nur noch dorthin. Er baut also eine sogenannte MAC-Adressen-Tabelle auf:

Beispiel:



Während ein Hub die Bandbreite des Netzwerks limitiert, steht der Verbindung zwischen zwei Hosts, die volle Bandbreite der Ende-zu-Ende-Netzwerk-Verbindung zur Verfügung.

Ein Switch arbeitet auf der Sicherungsschicht (Schicht 2) de s OSI-Modells und arbeitet ähnlich wie eine Bridge. Daher haben sich bei den Herstellern auch solche Begriffe durchgesetzt, wie z. B. Bridging Switch oder Switching Bridge. Die verwendet man heute allerdings nicht mehr.

Switches unterscheidet man hinsichtlich ihrer Leistungsfähigkeit mit folgenden Eigenschaften:

- Anzahl der speicherbaren MAC-Adressen für die Quell- und Zielports
- Verfahren, wann ein empfangenes Datenpaket weitervermittelt wird (Switching-Verfahren)
- Latenz (Verzögerungszeit) der vermittelten Datenpakete

Ein Switch ist im Prinzip nichts anderes als ein intelligenter Hub, der sich merkt, über welchen Port welcher Host erreichbar ist. Teure Switches können zusätzlich auf der Schicht 3, der Vermittlungsschicht, des OSI-Schichtenmodells arbeiten (Layer-3-Switch oder Schicht-3-Switch). Sie

sind in der Lage, die Datenpakete anhand der IP-Adresse an die Ziel-Ports weiterzuleiten. Im Gegensatz zu normalen Switche lassen sich auch ohne Router logische Abgrenzungen erreichen.

Switching-Verfahren

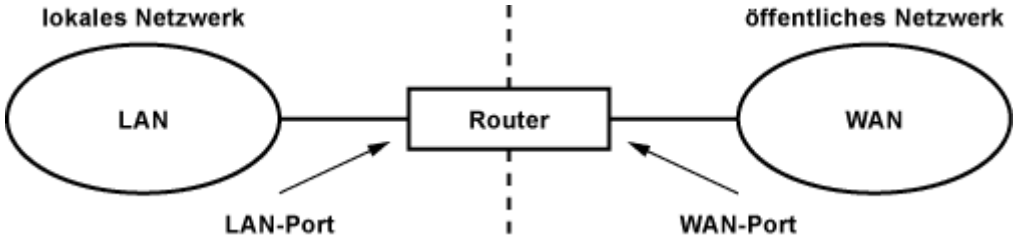
Switching-Verfahren	Beschreibung	Vorteile	Nachteile
Cut-Through	Beim Cut-Through-Verfahren wird unterschieden zwischen dem Fast-Forward und dem Fragment-Free Verfahren. **Fast-Forward** Der Switch leitet das Datenpaket sofort weiter, wenn er die Adresse des Ziels erhalten hat. **Fragment-Free** Der Switch empfängt die ersten 64 Byte des Daten-Paketes. Ist dieser Teil fehlerlos werden die Daten weitergeleitet. Die meisten Fehler und Kollisionen treten während den ersten 64 Byte auf. Dieses Verfahren wird trotz seiner effektiven Arbeitsweise selten genutzt.	Die Latenz, die Verzögerungszeit, zwischen Empfangen und Weiterleiten ist äußerst gering.	Fehlerhafte Datenpakete werden nicht erkannt und trotzdem an den Empfänger weitergeleitet.
Store-and-Forward	Der Switch nimmt das gesamte Datenpaket in Empfang und speichert es in einem Puffer. Dort wird dann das Paket mit verschiedenen Filtern geprüft und bearbeitet. Erst danach wird das Paket an den Ziel-Port weitergeleitet.	Fehlerhafte Datenpakete können so im voraus aussortiert werden.	Die Speicherung und Prüfung der Datenpakete verursacht eine Verzögerung, abhängig von der Größe des Datenpaketes.
Error Free Cut-Through	Eine Mischung aus mehreren der obigen Methoden. Wird ebenfalls meist nur von teueren Switches implementiert. Der Switch arbeitet zunächst im „Cut through“-Modus und schickt das Paket auf dem korrekten Port weiter ins LAN. Es wird jedoch eine Kopie des Frames im Speicher behalten, über das dann eine Prüfsumme berechnet wird. Wenn zu viele Fehler in kurzer Zeit auftreten, fällt der Switch in den Store and Forward-Modus zurück. Wenn die Fehlerrate wieder niedrig genug ist, schaltet er in den Cut through-Modus um. Ebenso kann der Switch temporär in den Fragment-Free-Modus schalten, wenn zuviele Fragmente mit weniger als 64 Byte Länge ankommen.	Sehr effizient	keine

Router

Ein Router verbindet mehrere Netzwerke mit unterschiedlichen Protokollen und Architekturen. Ein Router befindet sich häufig an den Außengrenzen eines Netzwerks, um es mit dem Internet oder einem anderen, größeren Netzwerk zu verbinden. Über die Routing-Tabelle entscheidet ein Router, welchen Weg ein Datenpaket nimmt. Es handelt sich dabei um ein dynamisches Verfahren, das Ausfälle und Engpässe ohne den Eingriff eines Administrators berücksichtigen kann. Ein Router hat mindestens zwei Netzwerkanschlüssen. Er arbeitet auf der Vermittlungsschicht (Schicht 3) des OSI-Schichtenmodells.

Die Aufgabe eines Routers ist ein komplexer Vorgang, der sich in 4 Schritte einteilen lässt:

- Ermittlung der verfügbaren Routen
- Auswahl der geeignetsten Route unter Berücksichtigung verschiedener Kriterien
- Herstellen einer physikalischen Verbindung zu anderen Netzwerken
- Anpassen der Datenpakete an die Übertragungstechnik (Fragmentierung)



Ein Router hat in der Regel zwei Anschlüsse. Einen für die LAN-Seite und einen für die WAN-Seite. Häufig sind die Ports mit der Bezeichnung LAN und WAN gekennzeichnet. Manchmal gibt es Port-Beschriftungen, bei denen nicht immer eindeutig ist, um was es sich handelt. Mit LAN ist immer das lokale Netzwerk mit privaten IP-Adressen gemeint, während die WAN-Seite das öffentliche Netzwerk kennzeichnet.

Zusammenfassung

Hub, Switch & Router

IP-Adressen

Damit die Wegewahl in Netzwerken und im Internet zur Übermittlung von Datenpaketen vom Senden zum Empfänger funktioniert, wird jedem Knotenpunkt im **Internet eine weltweit eindeutige (und einmalige) Adresse (IP-Adresse)** zugeordnet.

Für Rechner die ohne Router direkt mit dem Internet verbunden sind, heißt das, das deren IP-Adressen weltweit eindeutig sind.

Standardmäßig haben diese IP-Adressen eine Länge von **32 Bit (4 x 8Bit)**.

Diese werden aus Gründen der **Übersichtlichkeit in 4 Zahlen zu je 1 Byte** aufgeteilt. Meist werden die einzelnen Bytes durch einen Punkt getrennt und dezimal dargestellt.

Öffentliche IP-Adressen

Wenn man von öffentlichen Adressen spricht, meint man die IP-Adressen, die im Internet erreichbar sind. Die Zuweisung einer öffentlichen IP-Adresse erfolgt in der Regel durch einen Provider (z.B. A1, Telekom, ...) welche diese wiederum in Europa durch die Organisation RIPE-NCC , die wiederum Adressen von der IANA zugewiesen bekommt.

Private IP-Adressen

Private IP-Adressen (abgekürzt Private IP) sind IP-Adressen, die von der **IANA** nicht im Internet vergeben sind. Sie wurden für die **private Nutzung aus dem öffentlichen Adressraum ausgespart**, damit sie ohne administrativen Mehraufwand (Registrierung der IP-Adressen) in lokalen Netzwerken genutzt werden können. Als die **IP-Adressen des Internet Protokolls v4 knapp** wurden und dadurch eine **bewusste Einsparung öffentlicher IP-Adressen** notwendig wurde, war es umso wichtiger, private IP-Adressen in lokalen Netzwerken zur Verfügung zu haben, die **beliebig oft bzw. in beliebigen Netzwerken genutzt** werden können.

Netzklasse: Anzahl Netze (ohne Subnetting)	**Netzadressbereich**	**Subnetzmaske**	**CIDR-Notation**	**Anzahl Adressen**
Klasse A: 1 privates Netz mit 16.777.216 Adressen	10.0.0.0 bis 10.255.255.255	255.0.0.0	10.0.0.0/8	2^{24} = 16.777.216
Klasse B: 16 private Netze mit jeweils 65.536 Adressen	172.16.0.0 bis 172.31.255.255	255.240.0.0	172.16.0.0/12	2^{20} = 1.048.576
Klasse C: 256 private Netze mit jeweils 256 Adressen	192.168.0.0 bis 192.168.255.255	255.255.0.0	192.168.0.0/16	2^{16} = 65.536

Loopback Adresse

Die Class-A-Netzwerkadresse 127 ist weltweit reserviert für das sogenannte local loopback; sie dient zu Testzwecken der Netzwerkschnittstelle des eigenen Rechners. Die IP-Adresse **127.0.0.1** ist standardmäßig dem Loopback-Interface jedes Rechners zugeordnet.

Alle an diese Adresse geschickten Datenpakete werden nicht nach außen ins Netzwerk gesendet, sondern an der Netzwerkschnittstelle reflektiert. Die Datenpakete erscheinen, als kämen sie aus einem angeschlossenem Netzwerk.

Vergleich IP-Adresse vs. Telefonnummer

Ähnlich wie bei einer Telefonnummer setzt sich eine IP-Adresse aus mehreren Segmenten zusammen. Eine Telefonnummer besteht aus einer Vorwahl und einer Teilnehmernummer. Führen Sie ein Ortsgespräch, so muss die Vorwahl nicht angegeben werden. Ähnlich ist es bei IP-Adressen.

Diese bestehen auch aus zwei Teilen, wobei die ID für Identifikation steht:

- **Netzwerk-ID** im vorderen linken Teil entspricht der Vorwahl und gibt das entsprechende IP-Subnetz an.
- **Host-ID** im hinteren rechten Teil kennzeichnet eine einzelne Netzwerkkarte und entspricht dem Teilnehmernummer im Ortsnetz.

Entsprechend können Rechner im selben Subnetz direkt miteinander kommunizieren. Dagegen erfordert Kommunikation zischen Subnetzen eine Vermittlungsstelle, einen Router (Standardgateway), wo alle nicht im selben Netz adressierten Pakete hingeschickt werden.

Um zu erkennen, wo die Netzwerk-ID endet und die Host-ID beginnt, muss zusätzlich zur IP-Adresse zwingend eine sogenannte Subnetzmaske mit angegeben werden.

Subnetzmaske

Eine Subnetzmaske ist ein Bitmuster, das (von links nach rechts) Teile der IP-Adresse „maskiert“, um den Übergang zwischen Netz-ID und Host-ID zu kennzeichnen. Binär betrachtet besteht eine Subnetzmaske aus einer Folge von Einsen, die ab einer bestimmten Stelle umschlägt in eine Folge von Nullen. Dieser Umschlagpunkt gibt an, wie viele Bits zur Netzwerk-ID (Einsen) und zur Host-ID (Nullen) gehören.

Ein Auszug aus den möglichen Subnetzmasken für ein Klasse C-Netz

Maske	Maske (kurze Schreibweise)	Anzahl Hosts pro Netz	Netze	Beispiel Klasse C-Netz (192.168.1.0)
255.255.255.0	/24	256	1	192.168.1.0 - 192.168.1.255
255.255.255.128	/25	128	2	192.168.1.0 - 192.168.1.127 192.168.1.128 - 192.168.1.255
255.255.255.192	/26	64	4	192.168.1.0 - 192.168.1.63 192.168.1.64 - 192.168.1.127 192.168.1.128 - 192.168.1.191 192.168.1.192 - 192.168.1.255
255.255.255.224	/27	32	8	192.168.1.0 - 192.168.1.31 192.168.1.32 - 192. 168.1.63 … 192.168.192 - 192.168.1.223 192.168.1.224 - 192.168.1.255
255.255.255.240	/28	16	16	192.168.1.0 - 192.168.1.15 192.168.1.16 - 192.168.1.31 … 192.168.1.224 - 192.168.1.239 192.168.1.240 - 192.168.1.255
255.255.255.248	/29	8	32	192.168.1.0 - 192.168.1.7 192.168.1.8 - 192.168.1.15 … 192.168.1.240 - 192.168.1.247 192.168.1.248 - 192.168.1.255
255.255.255.252	/30	4	64	192.168.1.0 - 192.168.1.3 192.168.1.4 - 192.168.1.7 … 192.168.1.248 - 192.168.1.251 192.168.1.252 - 192.168.1.255
255.255.255.254	/31	2	128	192.168.1.0 - 192.168.1.1 192.168.1.2 - 192.168.1.3 … 192.168.1.252 - 192.168.1.253 192.168.1.254 - 192.168.1.255
255.255.255.255	/32	1	256	192.168.1.0 192.168.1.1 … 192.168.1.254 192.168.1.255

Subnetting

Die Aufteilung eines zusammenhängenden Adressraums von IP-Adressen in mehrere kleinere Adressräume nennt man Subnetting. Ein Subnet, Subnetz bzw. Teilnetz ist ein physikalisches Segment eines Netzwerks, in dem IP-Adressen mit der gleichen Netzwerkadresse benutzt werden. Diese Teilnetze können über Routern miteinander verbunden werden und bilden dann ein großes zusammenhängendes Netzwerk.

Beispiel 1:

IP-Adresse: 192.168.128.17

```
Dezimale Darstellung:           192.        168.        128.          17
Bit-Darstellung:          1100 0000   1010 1000    1000 0000   0001 0001
Hex-Darstellung:                C0          A8          80          11
```

Subnetzmaske: 255.255.255.0

```
Dezimale Darstellung:           255.        255.        255.          0
Bit-Darstellung:           1111 1111   1111 1111   1111 1111  0000 0000
Hex-Darstellung:                FF          FF          FF          00
```

Verknüpft man nun die binäre IP-Adresse mit der Subnetzmaske mit einem Logischen AND, so bekommt man die Netz-ID (=Netzadresse).

```
    <----------- NETZ-ID ------------><- HOST-ID ->
    1100 0000   1010 1000   1000 0000   0001 0001    (IP-Adresse)
AND 1111 1111   1111 1111   1111 1111   0000 0000    (Subnetzmaske)
    ---------------------------------------------------------------
    1100 0000   1010 1000   1000 0000   0000 0000    (Netz-ID)
Netz-ID in Dezimaldarstellung:
 192.168.128
```

Verknüpft man nun die binäre IP-Adresse mit der negierten Subnetzmaske mit einem Logischen AND, so bekommt man die Host-ID.

```
    <----------- NETZ-ID ------------><- HOST-ID ->
```

```
    1100 0000   1010 1000   1000 0000   0001 0001    (IP-Adresse)
AND 0000 0000   0000 0000   0000 0000   1111 1111    (negierte Subnetzmaske)
    ------------------------------------------------------------------------
-
    0000 0000   0000 0000   0000 0000   0001 0001    (Host-ID)
Host-ID in Dezimaldarstellung:
 17
```

Das heißt im Netzwerk 192.168.128.0 stehen theoretisch 256 (0-255) Adressen zur Adressierung von Netzwerkgeräten zur Verfügung. Praktisch sind es aber nur 254 Adressen, da zwei Adressen

- die **Netzadresse** (= 1. Adresse im Netz –> 192.168.128.0)
- die **Broadcastadresse** (= letzte Adresse im Netz –> 192.168.128.255)

reserviert sind.

Beispiel 2:

```
IP-Adresse 130.94.122.195/27

              Dezimal          Binär
Berechnung

IP Adresse    130.094.122.195  10000010 01011110 01111010 11000011
ip-adresse
Netzmaske     255.255.255.224  11111111 11111111 11111111 11100000      AND
netzmaske
------------------------------------------------------------------------
----------
Netzwerkteil  130.094.122.192  10000010 01011110 01111010 11000000   =
netzwerkanteil

IP Adresse    130.094.122.195  10000010 01011110 01111010 11000011
ip-adresse
Netzmaske     255.255.255.224  00000000 00000000 00000000 00011111  AND(NOT
netzmaske)
------------------------------------------------------------------------
----------
Geräteteil                  3  00000000 00000000 00000000 00000011       =
geräteteil
```

Bei einer Netzmaske mit 27 gesetzten Bits ergibt sich ein Netzwerkteil von 130.94.122.192. Es verbleiben 5 Bits und damit `2^5=32` Adressen für den Geräteteil. Hiervon werden noch je 1 Adresse für das Netzwerk selbst und für den Broadcast benötigt, sodass 30 Adressen für Geräte zur Verfügung stehen.

Beispiel 3:

```
IP-Adresse 130.94.122.117/28
```

```
              Dezimal          Binär                                    
Berechnung

IP Adresse    130.094.122.117   10000010 01011110 01111010 01110101
ip-adresse
Netzmaske     255.255.255.240   11111111 11111111 11111111 11110000      AND
netzmaske
-------------------------------------------------------------------------
----------
Netzwerkteil 130.094.122.112   10000010 01011110 01111010 01110000   =
netzwerkanteil


IP Adresse    130.094.122.117   10000010 01011110 01111010 01110101
ip-adresse
Netzmaske     255.255.255.240   00000000 00000000 00000000 00001111  AND(NOT
netzmaske)
-------------------------------------------------------------------------
----------
Geräteteil      0.  0.  0.  5   00000000 00000000 00000000 00000101       =
geräteteil
```

Bei einer Netzmaske mit 28 gesetzten Bits ergibt sich ein Netzwerkteil von 130.94.122.112. Es verbleiben 4 Bits und damit `2^4=16` Adressen für den Geräteteil. Hiervon werden noch je 1 Adresse für das Netzwerk selbst und für den Broadcast benötigt, sodass 14 Adressen für Geräte zur Verfügung stehen.

- Zusammenfassung zu Subnetzmasken

Netzwerkklassen

Es wurden fünf verschiedene Netzwerkklassen festgelegt: Class A, B, C, D, E.
In der Praxis sind nur A, B, C von Bedeutung. (D und E werden nur für Testzwecke genutzt.)

Class-A-Netze:

Adresse beginnt mit einer binären 0, 7 Bit für Netzwerk-Adresse, 24 Bit für Host-Adresse. Damit gibt es weltweit 127 derartige Netzwerke, ein Class-A-Netz kann bis zu 16 Mio. Teilnehmer haben. Alle derartigen Netzadressen sind bereits belegt.

```
IP-Adressen von Class-A-Netzen
0.0.0.0 bis 127.255.255.255
```

Class-B-Netze:

Adresse beginnt mit der binären Ziffernkombination 10, 14 bit für Netzwerk-Adresse, 16 Bit für Host-Adresse. Damit gibt es weltweit 16384 derartige Netzwerke, ein Class-B-Netz kann bis zu 65536 Teilnehmer haben. Alle derartigen Netzadressen sind bereits belegt.

```
IP-Adressen von Class-B-Netzen
128.0.0.0 bis 191.255.255.255
```

Class-C-Netze:

Adresse beginnt mit der binären Ziffernkombination 110, 21 Bit für Netzwerk-Adresse, 8 Bit für Host-Adresse. Damit gibt es weltweit 2 Mio. derartige Netzwerke, ein Class-C-Netz kann bis zu 256 Teilnehmer haben. Neu zugeteilte Netzadressen sind heute immer vom Typ C. Es ist abzusehen, dass bereits in Kürze alle derartigen Adressen vergeben sein werden.

```
IP-Adressen von Class-C-Netzen
192.0.0.0 bis 223.255.255.255
```

Netzwerkmasken der verschiedenen Netze

Netzwerkmasken unterscheiden sich in der Länge des Netzwerk- (alle Bitstellen auf 1) und Hostanteils (alle Bitstellen auf 0) abhängig von der Netzwerkklasse

```
          1.Byte 2.Byte 3.Byte 4.Byte
Class A     255.     0.     0.     0
Class B     255.   255.     0.     0
Class C     255.   255.   255.     0
```

Netzwerkmasken stellen einen Filter dar, an dem Rechner entscheiden können, ob sie sich im selben (logischen) Netz befinden

Broadcast-Adresse

Die Broadcast-Adresse ergibt sich aus der IP-Adresse, bei der alle Bitstellen des Hostanteils auf 1 gesetzt sind. Sie bietet die Möglichkeit, Datenpakete an alle Rechner eines logischen Netzwerkes zu senden. Sie wird ermittelt, indem die Netzwerkadresse mit der invertierten Netzwerkmaske bitweise ODER-verknüpft wird.

```
           192.168.100.000 11000000 10101000 01100100 00000000
           000.000.000.255 00000000 00000000 00000000 11111111
Broadcast  192.168.100.255 11000000 10101000 01100100 11111111
```

Loopback-Adresse

Die Class-A-Netzwerkadresse 127 ist weltweit reserviert für das sogenannte local loopback; sie dient zu Testzwecken der Netzwerkschnittstelle des eigenen Rechners. Die IP-Adresse 127.0.0.1 ist standardmäßig dem Loopback-Interface jedes Rechners zugeordnet.

Alle an diese Adresse geschickten Datenpakete werden nicht nach außen ins Netzwerk gesendet, sondern an der Netzwerkschnittstelle reflektiert. Die Datenpakete erscheinen, als kämen sie aus einem angeschlossenem Netzwerk.

Vergabe der IP-Adressen

Private und öffentliche IP-Adressen

Da der IP-Adressbereich begrenzt ist, wurden sog. private IP-Adressen festgelegt, die im globalen Internet nicht bekannt werden:

```
  10.   0. 0. 0 –  10.255.255.255
  172. 16. 0. 0 – 172. 31.255.255
  192.168. 0. 0 – 192.168.255.255
  127.  0. 0. 1  (loopback-Adresse)
```

Es muss ein Network Address Translator (NAT) verwendet werden, um auf das globale Internet zugreifen zu können.

Vorteil: einfach, einen ISP (Internet Service Provider) zu wechseln, da nur die IP-Adresse nach außen verändert werden muss.

- am Client händisch eintragen (incl. Subnetmask und Gateway)
- mittels DHCP (dynamic host configuration protocol)
- ein DHCP-Server vergibt einem Client, der sich im Netzwerk befindet, automatisch eine IP-

Adresse für eine bestimmte Zeit (lease-time)

Domain Name Service (DNS)

DNS ist ein Protokoll, das die Zuordnung von Computernamen zu IP-Adressen regelt.
Systematischer Aufbau: **hostname.[subdomain].domain.topleveldomain** z.B. hyper.mat.univie.ac.at

Zuordnung erfolgt über eigene Rechner im Internet (DNS-Server)

Verwaltung der Domains: **InterNIC** (.com, .net, .org, .int), **NIC.AT** (.at)

- Viele Tools rund um IP-Adressen

Übungen IP-Adressierung

Bsp 1

Befinden sich 192.168.0.93/27 und 192.168.0.97/27 im gleichen Netzwerk-Segment?

Bsp 2

Wie lauten die Neztwerkadressen des ersten/letzten Hosts des Netzwerkes 192.168.0.96/27 und die Broadcastadresse?

Bsp 3

Das Netz 195.1.31.0 soll in 30 Subnetze aufgeteilt werden. Bestimme die Subnetzmaske, und die Anzahl Adressen pro Subnet!

Bsp 4

Das Netz 10.0.0.0 soll in 200 Subnetze aufgeteilt werden. Bestimme die Subnetzmaske, und die Anzahl Adressen pro Subnet!

Bsp 5

Das Netz 192.168.1.0 soll in Subnetze mit je 18 Host-Adressen aufgeteilt werden. Bestimme die Subnetzmaske, und die Anzahl Subnetze!

Bsp 6

Das Netz 192.168.100.0 soll in Subnetze mit je 5 Host-Adressen aufgeteilt werden. Bestimmen Sie die Subnetzmaske, und die Anzahl Subnetze!

Bsp 7

Bestimme Netz- und Broadcastadresse des Subnets, in dem die Adresse 195.1.31.135 mit Netzmaske 255.255.255.128 liegt!

Bsp 8

Bestimme Netz- und Broadcastadresse des Subnets in dem die Adresse 195.1.31.135 mit Netzmaske

255.255.255.192 liegt!

Bsp 9

Bestimme Netz- und Broadcastadresse des Subnets in dem die Adresse 15.3.128.222 mit Netzmaske 255.255.255.240 liegt!

Bsp 10

Gib den Bereich der Rechneradressen an, den das Teilnetz Nummer drei (132.45.96.0/19) hat.

Bsp 11

Berechne Netz-, Broadcast und Hostadressen des 0.,1.,15. und 31 Subnetz des Netzes 192.168.1.0/29

Bsp 12

Vervollständige folgende Tabelle:

IP-Adresse	10.1.1.1	Netzwerkadresse des Subnet	. . .
Netmask	255.255.0.0	Broadcastadresse des Subnet	. . .
Anzahl Host pro Subnet		Anzahl Subnets in diesem Netz	

Bsp 13

Vervollständige folgende Tabelle:

IP-Adresse	10.1.1.1/24	Netzwerkadresse des Subnet	. . .
Netmask	255.255.255.0	Broadcastadresse des Subnet	. . .
Anzahl Host pro Subnet		Anzahl Subnets in diesem Netz	

Übungen IP-Adressierung

Bsp 1

Befinden sich 192.168.0.93/27 und 192.168.0.97/27 im gleichen Netzwerk-Segment?

Lösung

```
1) NW-ID: 192.168.0.64
2) NW-ID: 192.168.0.96
=> Nein
```

Bsp 2

Wie lauten die Neztwerkadressen des ersten/letzten Hosts des Netzwerkes 192.168.0.96/27 und die Broadcastadresse?

Lösung

```
1. Host: 192.168.0.97
letzter Host: 192.168.0.126
BC: 192.168.0.127
```

Bsp 3

Das Netz 195.1.31.0 /24 soll in weitere 30 Subnetze aufgeteilt werden. Bestimme die Subnetzmaske, und die Anzahl Adressen pro Subnet!

Lösung

```
Subnetzmaske: 255.255.255.248
Adressen: 6
```

Bsp 4

Das Netz 10.0.0.0 /16 soll in mindestens 200 Subnetze weiter unterteilt werden. Bestimme die Subnetzmaske, und die Anzahl Adressen pro Subnet!

Lösung

```
Subnetzmaske: 255.255.255.0
```

```
Adressen: 256
```

Bsp 5

Das Netz 192.168.1.0 /24 soll in Subnetze mit je 18 Host-Adressen aufgeteilt werden. Bestimme die Subnetzmaske, und die Anzahl Subnetze!

Lösung

```
Subnetzmaske: 255.255.255.224
Subnetze: 8
```

Bsp 6

Das Netz 192.168.100.0 /24 soll in Subnetze mit je 5 Host-Adressen aufgeteilt werden. Bestimmen Sie die Subnetzmaske, und die Anzahl Subnetze!

Lösung

```
Subnetzmaske: 255.255.255.248
Subnetze: 32
```

Bsp 7

Bestimme Netz- und Broadcastadresse des Subnets, in dem die Adresse 195.1.31.135 mit Netzmaske 255.255.255.128 liegt!

Lösung

```
NW-ID: 195.1.31.128
BC: 195.1.31.255
```

Bsp 8

Bestimme Netz- und Broadcastadresse des Subnets in dem die Adresse 195.1.31.135 mit Netzmaske 255.255.255.192 liegt!

Lösung

```
NW-ID: 195.1.31.128/26
BC: 195.1.31.191
```

Bsp 9

Bestimme Netz- und Broadcastadresse des Subnets in dem die Adresse 15.3.128.222 mit Netzmaske 255.255.255.240 liegt!

Lösung

```
NW-ID: 15.3.128.208
Broadcast: 15.3.128.223
```

Bsp 10

Gib den Bereich der Rechneradressen an, den das Teilnetz Nummer drei (132.45.96.0/19) hat.

Lösung

```
132.45.96.1
bis
132.45.127.254
```

Bsp 11

Berechne Netz-, Broadcast und Hostadressen des 0.,1.,15. und 31 Subnetz des Netzes 192.168.1.0/29

Lösung

```
0. Netz:
* NW-ID: 192.168.1.0
* BC: 192.168.1.7
* Hosts: 192.168.1.1-6
```

```
1. Netz:
* NW-ID: 192.168.1.8
* BC: 192.168.1.15
* Hosts: 192.168.1.9-14
```

```
15. Netz:
* NW-ID: 192.168.1.120
* BC: 192.168.1.127
* Hosts: 192.168.1.121-126
```

```
31. Netz:
* NW-ID: 192.168.1.248
* BC: 192.168.1.255
* Hosts: 192.168.1.249-254
```

Bsp 12

Vervollständige folgende Tabelle:

IP-Adresse	10.1.1.1	Netzwerkadresse des Subnet	. . .
Netmask	255.255.0.0	Broadcastadresse des Subnet	. . .
Anzahl Hosts pro Subnet		Anzahl Subnets mit je 256 Hosts in diesem Netz	

Lösung

IP-Adresse	10.1.1.1	Netzwerkadresse des Subnet	10.1.0.0
Netmask	255.255.0.0	Broadcastadresse des Subnet	10.1.255.255
Anzahl Hosts pro Subnet	65536	Anzahl Subnets mit je 256 Adressen in diesem Netz	256

Bsp 13

Vervollständige folgende Tabelle:

IP-Adresse	10.1.1.1/24	Netzwerkadresse des Subnet	. . .
Netmask	255.255.255.0	Broadcastadresse des Subnet	. . .
Anzahl Hosts pro Subnet		Anzahl Subnets mit je 4 Adressen in diesem Netz	

Lösung

IP-Adresse	10.1.1.1/24	Netzwerkadresse des Subnet	10.1.1.0
Netmask	255.255.255.0	Broadcastadresse des Subnet	10.1.1.255
Anzahl Hosts pro Subnet	254	Anzahl Subnets mit je 4 Adressen in diesem Netz	64

Netzwerksimulation mit FILIUS

Software

- Download Filius

Anleitungen

- http://lernsoftware-filius.de
- einfuehrung_filius.pdf
- beispielaufgaben_filius.pdf
- aufgabenblatt_filius.pdf

Vernetzung von zwei Rechnern

Die einfache Aufgabe, mehrere Rechner per Netzwerkkabel zu einem lokalen Netzwerk zu verbinden, führt zu verschiedenen Lösungen. In diesem Abschnitt kannst du verschiedene Lösungsideen vergleichen, das Vernetzen von Rechnern in Filius selbst ausprobieren und erste Verbindungstests durchführen.

Adressierung von Rechnern

Um einzelne Rechner in einem Rechnernetz direkt ansprechen zu können, ordnet man ihnen eindeutige Adressen zu. Es haben sich zwei Arten der Adressierung etabliert:

- Die **MAC-Adresse** ist eine **Hardware-Adresse**, die einer LAN-Schnittstelle eines Rechners fest zugeordnet ist. In der Regel wird diese Adresse bei der Herstellung der Schnittstelle festgelegt und kann im Nachhinein nicht mehr verändert werden.
- Die **IP-Adresse** ist eine **veränderbare Adresse**, die einer LAN-Schnittstelle entweder am Rechner lokal oder über das Rechnernetz zentral zugeordnet wird.

Filius

Name	Rechner 1
MAC-Adresse	70:39:B4:F1:4E:A3
IP-Adresse	192.168.0.1

Windows 10 - Einstellungen

Windows - Kommandozeile

Eingabeaufforderung

```
C:\Users\Andi>ipconfig /all

Windows-IP-Konfiguration

   Hostname  . . . . . . . . . . . . : DESKTOP-HG7AS0I
   Primäres DNS-Suffix . . . . . . . :
   Knotentyp . . . . . . . . . . . . : Hybrid
   IP-Routing aktiviert  . . . . . . : Nein
   WINS-Proxy aktiviert  . . . . . . : Nein

Ethernet-Adapter Ethernet:

   Verbindungsspezifisches DNS-Suffix:
   Beschreibung. . . . . . . . . . . : Realtek PCIe GbE Family Controller
   Physische Adresse . . . . . . . . : 2C-F0-5D-0D-F5-66
   DHCP aktiviert. . . . . . . . . . : Ja
   Autokonfiguration aktiviert . . . : Ja
   Verbindungslokale IPv6-Adresse  . : fe80::c82f:6df:9609:9210%13(Bevorzugt)
   IPv4-Adresse  . . . . . . . . . . : 192.168.1.173(Bevorzugt)
   Subnetzmaske  . . . . . . . . . . : 255.255.255.0
   Lease erhalten. . . . . . . . . . : Mittwoch, 19. März 2025 19:24:04
   Lease läuft ab. . . . . . . . . . : Donnerstag, 20. März 2025 19:23:32
   Standardgateway . . . . . . . . . : 192.168.1.1
   DHCP-Server . . . . . . . . . . . : 192.168.1.1
   DHCPv6-IAID . . . . . . . . . . . : 53276765
   DHCPv6-Client-DUID. . . . . . . . : 00-01-00-01-26-50-7C-EC-2C-F0-5D-0D-F5-66
   DNS-Server  . . . . . . . . . . . : 192.168.1.1
   NetBIOS über TCP/IP . . . . . . . : Aktiviert

C:\Users\Andi>
```

Ping-Befehl

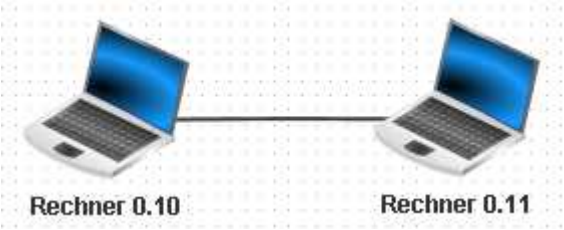
Der Befehl „ping“ wird auf einem Rechner verwendet, um zu testen, ob ein bestimmter anderer

Rechner erreichbar ist.

Anschaulich entspricht der Ablauf eines solchen ping-Tests einem Ballwechsel beim Ping-Pong-Spiel (ein Synonym für Tischtennis): Der Rechner, auf dem der ping-Befehl ausgeführt wird, schickt nacheinander viermal ein Datenpaket zu einem anderen Rechner (ping) und erhält von diesem – wenn alles gut geht – viermal ein Antwortpaket (pong).

Aufgaben

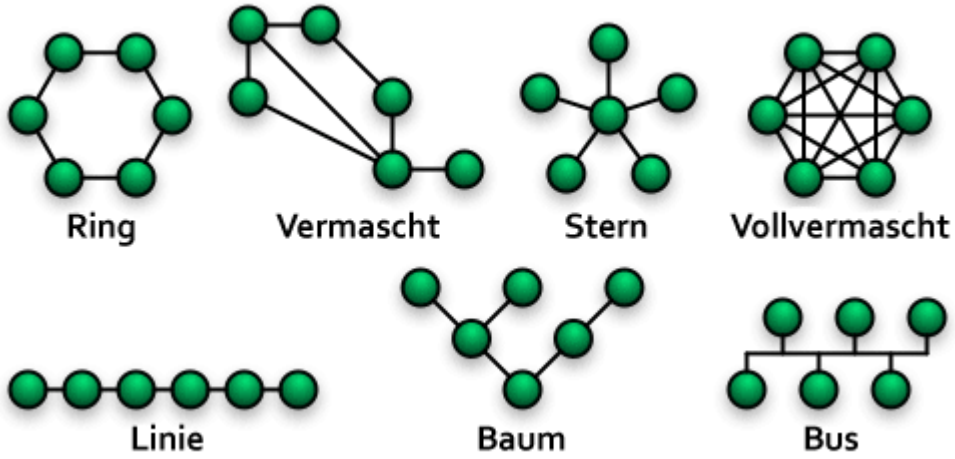
Aufgabe 1



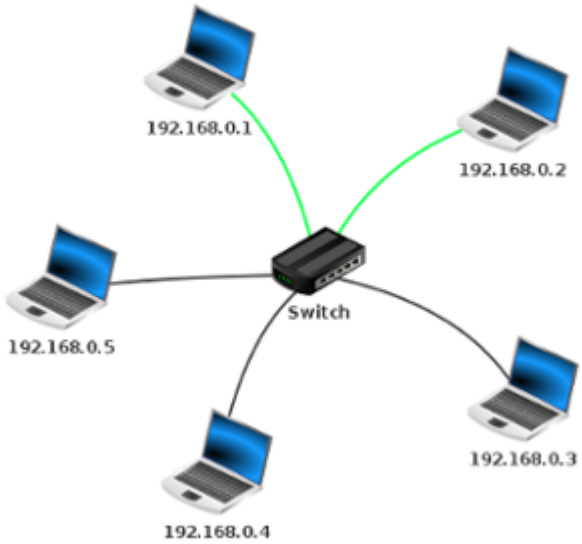
1. Erstellen Sie ein Netzwerk mit zwei vernetzten Computern, welche beide eine Client-Funktion haben (Symbol: Notebook). Die Computer sollen die abgebildeten Namen sowie die IPs 192.168.0.10/24 und 192.168.0.11/24 besitzen. (Durch die richtige Subnetzmaske 255.255.255.0 stellen Sie sicher, dass beide Computer im selben Netzwerk liegen.)
2. Installieren Sie auf dem Rechner 0.10 eine „Befehlszeile“ (Terminal). Starten Sie das Terminal und testen Sie die Verbindung zum Rechner 0.11 mit dem Befehl `ping 192.168.0.11`. Beobachten Sie die Netzwerkaktivität, indem Sie sich den Datenaustausch von Rechner 0.10 anzeigen lassen.
3. Testen Sie auch andere Befehle auf dem Terminal, wie z. B. die Befehle `ipconfig`, oder `host localhost` oder `dir`. Der Sinn des host-Befehls wird zu einem späteren Zeitpunkt im Zusammenhang mit einem DNS-Server evtl. deutlicher.

Vernetzung von mehreren Rechnern

Mehrere Rechner lassen sich über geeignete Schnittstellen mit Netzwerkkabeln oder anderen Medien zu einem lokalen Netzwerk (englisch: Local Area Network, kurz LAN) verbinden. Die Struktur einer solchen Verkabelung nennt man Topologie des lokalen Netzwerks.



Wenn mehrere Rechner intuitiv miteinander vernetzt werden sollen, entsteht oftmals das Problem, dass die einzelnen Rechner viele Schnittstellen benötigen. In Wirklichkeit jedoch besitzen die meisten Rechner - wie auch alle „Notebook“-Rechner in Filius - nur eine geeignete Netzwerk-Schnittstelle. Umgekehrt enthalten lokale Netzwerke von Rechnern unabhängige Verteiler, sogenannte Switches, die mehrere Netzwerkkabel an einem Punkt miteinander verbinden.



Aufgaben

Aufgabe 1

a) Konstruiere in Filius ein lokales Netzwerk mit fünf („Notebook“-)Rechnern. Verwende dabei ein

Switch. Konfiguriere die Rechner, indem du ihnen geeignete Namen und IP-Adressen (z.B. 192.168.0.1, 192.168.0.2, ...) gibst. Hinweis: Bei der Konfiguration eines Rechners in Filius kannst du zur Vereinfachung „IP-Adresse als Name verwenden“ anklicken.

b) Installiere auf einem der Rechner ein Befehlszeilenterminal, reduziere die Simulations-Geschwindigkeit im Aktionsmodus auf Filius Geschwindigkeitsregler und teste die Erreichbarkeit eines anderen Rechners mit dem ping-Befehl.

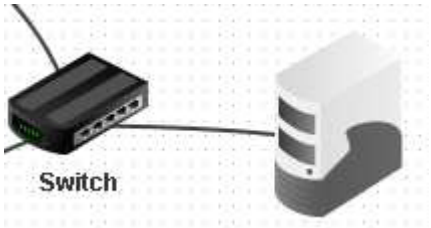
Beobachte genau die Wege, die die Daten des Ping-Befehls durch dein lokales Netzwerk nehmen (grün leuchtende Leitungen). Welche Rolle übernimmt hier das Switch?

c) Führe selbst weitere Experimente mit dem ping-Befehl auf verschiedenen Rechnern und auch mit veränderter Topologie (z.B. mehrere Switches) durch.

d) Löse das Rätsel

Aufgabe 2

Erweitere nun das Netzwerk um einen dritten Computer, einen Server, mit dem abgebildeten Namen und einer passenden IP-Adresse im gleichen Netzwerk. Achten

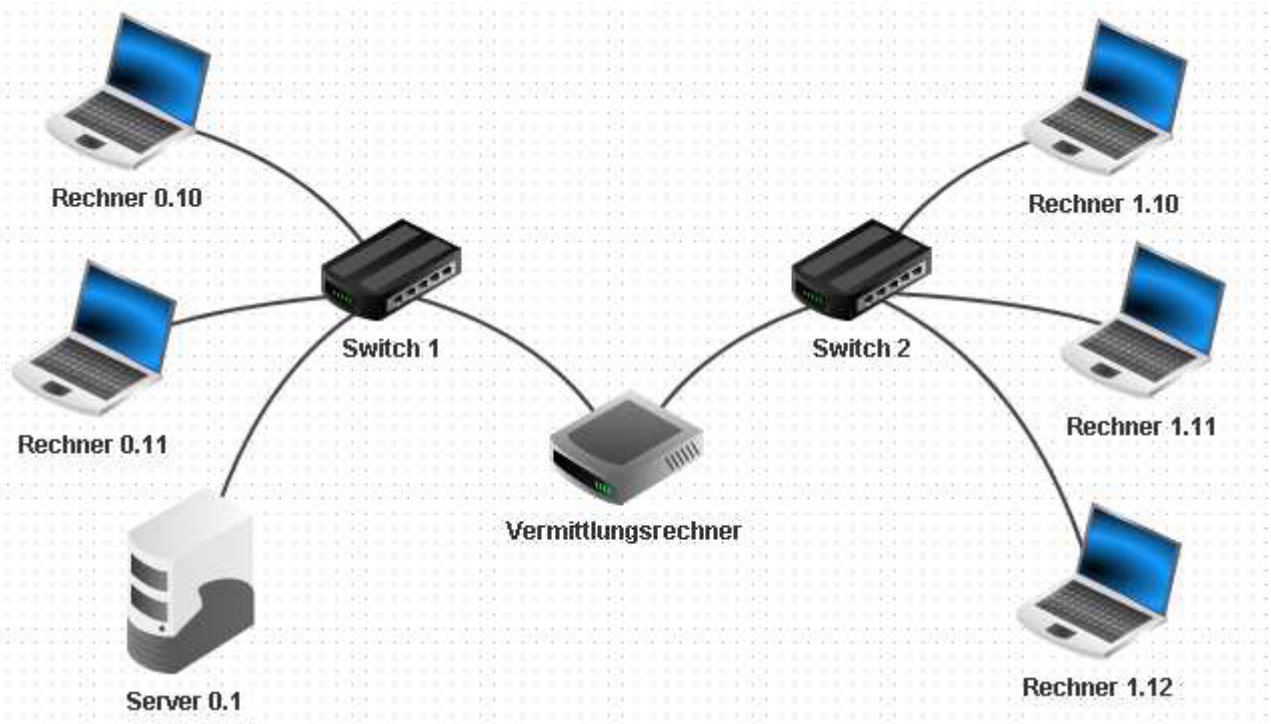


Installiere auf dem Server einen Echo-Server und starten diesen auf dem voreingestellten Port 55555. Installiere auf einem Client einen Echo-Client und verbinde diesen mit dem Server. Sende vom Client einige Textnachrichten und beobachte den Effekt. Schaue dir auch die Netzwerkaktivität im Datenaustausch-Fenster des Clients an.

Aufgabe 3

- Erstellen Sie zwei Netzwerke mit je drei Rechnern wie abgebildet.
- Die Computer des ersten Netzwerks sollen die abgebildeten Namen besitezen sowie die IPs 192.168.0.10/24 und 192.168.0.11/24 zugewiesen bekommen.
- Der Server 192.168.0.1/24 soll die IP-Adressen in diesem Netzwerk vergeben. Rechner 0.10 soll die IP-Adresse statisch vom Server bekommen, Rechner 0.11 die IP-Adresse aus einem Pool.
- Die Rechner des zweiten Netzwerks sollen sich einem logisch anderen Netzwerk befinden. Wählen Sie dafür die IPs 10.1.1.10/16 bis 10.1.1.12/16. (fix vergeben). Verbinden Sie anschließend die beiden Netzwerke mit einem Vermittlungsrechner (Router), welcher die

Netzwerkkarten mit den IPs 192.168.0.254/24 und 10.1.1.254/16 besitzt.

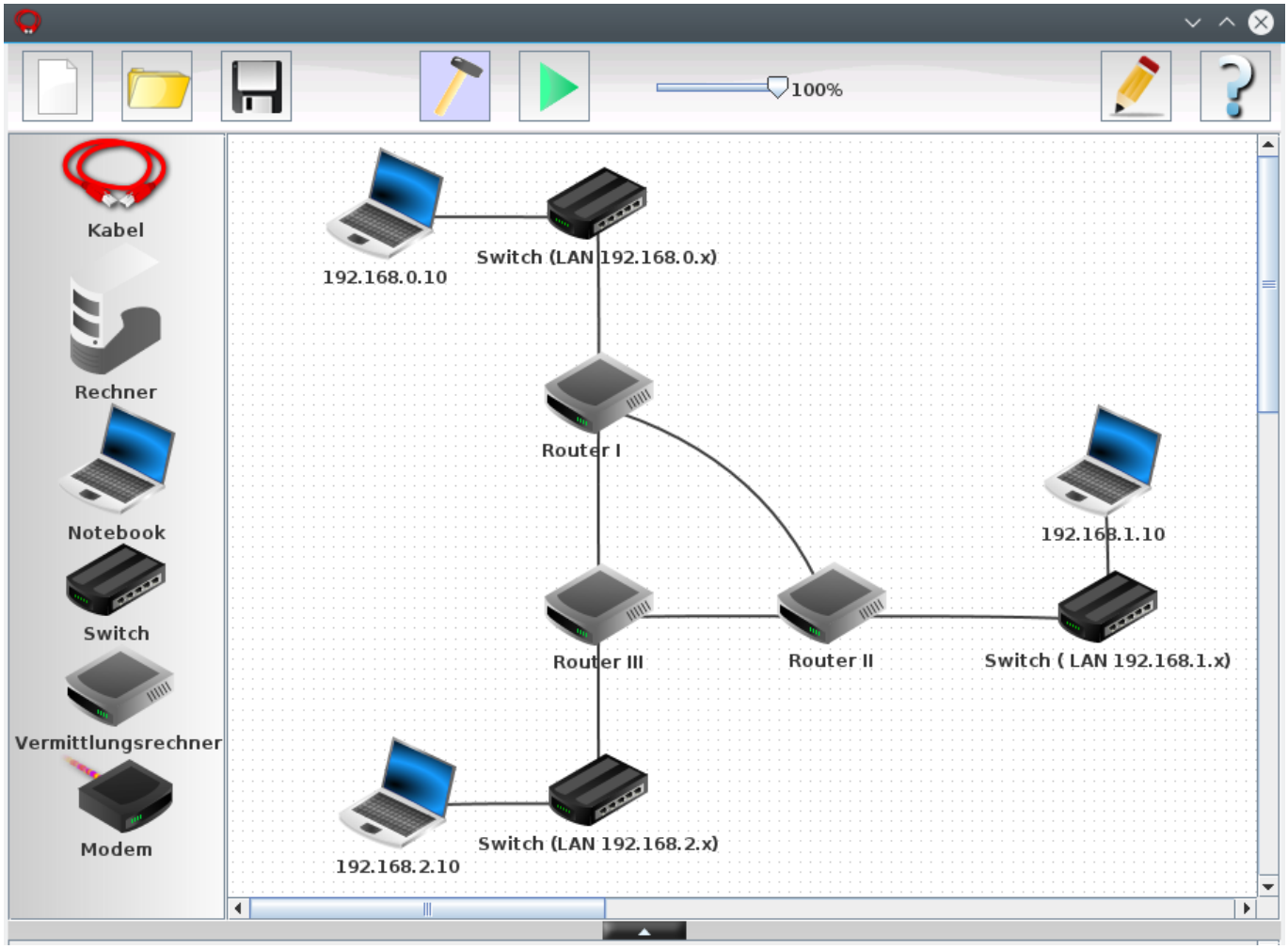


- Prüfen Sie anschließend im Terminal mit einem ping-Befehl die Verbindung der Rechner vom Netzwerk 192.168.0.0/24 zu den Rechnern des Netzes 10.1.0.0/16.
    - Beschreiben Sie, welches Problem auftreten kann und wie man sicherstellen kann, dass die Rechner aus einem fremden Netzwerk erreicht werden können.
- Testen Sie die Netzwerkverbindung auch mit dem Echo-Client und Echo-Server. Installieren Sie dazu auf einem Rechner aus dem Netz 192.168.0.0/24 einen Echo-Server und auf einem Rechner im Netz 10.1.0.0/16 einen Echo-Client.

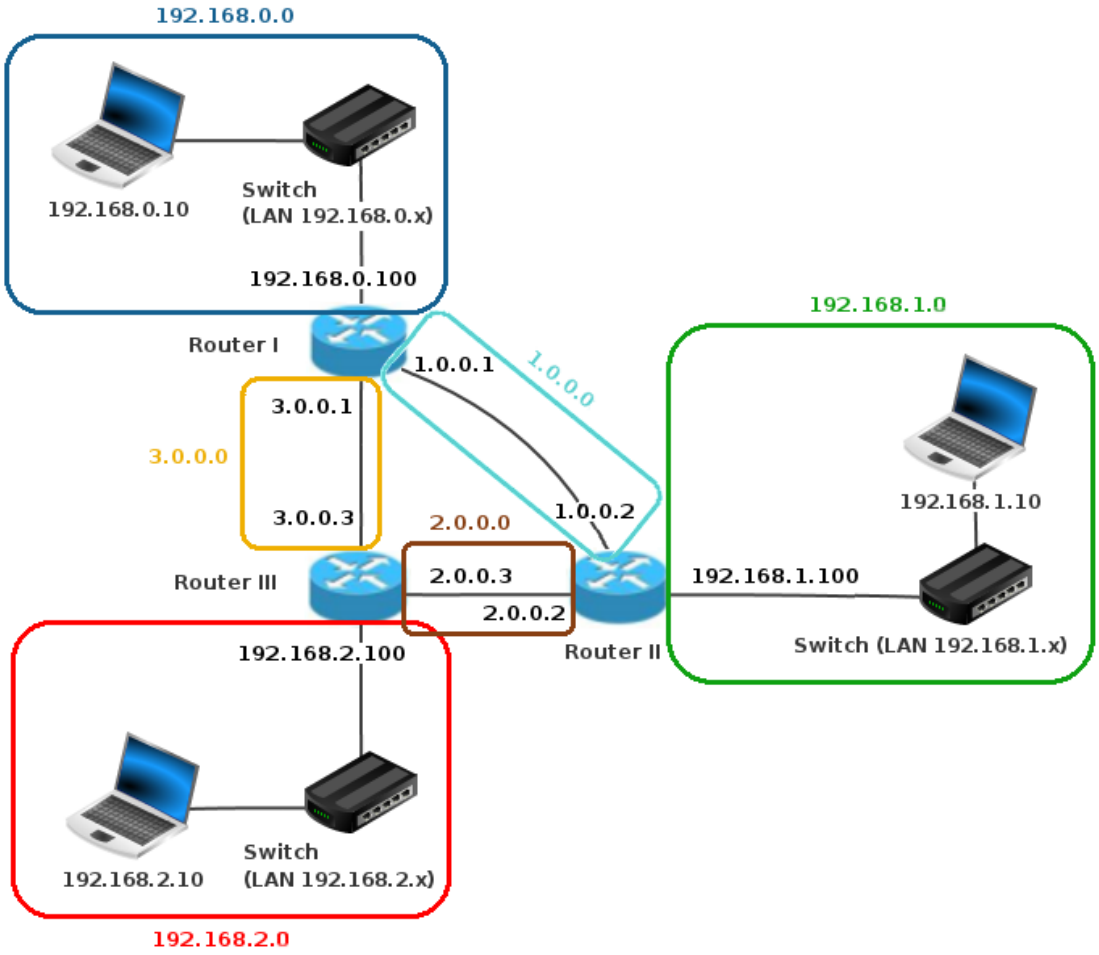
Vernetzung mehrerer Netze mit Router

Bei der Einrichtung eines lokalen Netzwerks in unserem Start-up-Unternehmen haben wir uns darauf beschränkt, Teilnetze über einen einzigen Router miteinander zu vernetzen. In großen Rechnernetzen sieht die Realität dagegen anders aus: Üblicherweise sind Teilnetze über viele Router in Form einer vermaschten Topologie miteinander verbunden und der Weg eines Datenpakets durch das Rechnernetz ist nicht unbedingt eindeutig bestimmt.

Wir betrachten also ein komplexeres Rechnernetz, das aus mehreren Teilnetzen besteht, die mit Routern wie in der Abbildung gezeigt verbunden sind.



Hier sind jetzt insgesamt 6 Teilnetze beteiligt. Die folgende Abbildung veranschaulicht diese Teilnetze mit den zugehörigen IP-Adressen.



Statische Routingtabellen

Die Kommunikation in einem solchen Rechnernetz funktioniert nicht ohne Weiteres, da jeder Router zunächst nur die an ihn mündenden Teilnetze kennt. Um Nachrichten zuverlässig auch in entfernte Teilnetze zu transportieren, arbeiten Router mit sogenannten Routingtabellen.

Eine vereinfachte Routingtabelle von Router I in unserem Beispielnetz könnte wie folgt aussehen:

Ziel	Nächstes Gateway	Über Schnittstelle
192.168.1.10	1.0.0.2	1.0.0.1
192.168.2.10	3.0.0.3	3.0.0.1

Aufgabe 1

a) Kannst du die Einträge der Routingtabelle deuten?

b) Welche Tabellenzeile könnte man zu dem Zielrechner 192.168.1.10 zusätzlich ergänzen? Welche Vorteile hätte eine solche Ergänzung und welche Probleme würde sie möglicherweise mit sich bringen?

Konfiguration von statischen Routen in Filius

Routingtabellen in Filius (sie heißen dort „Weiterleitungstabellen“) sehen ähnlich aus wie die obige Beispieltabelle.

Aufgabe 2

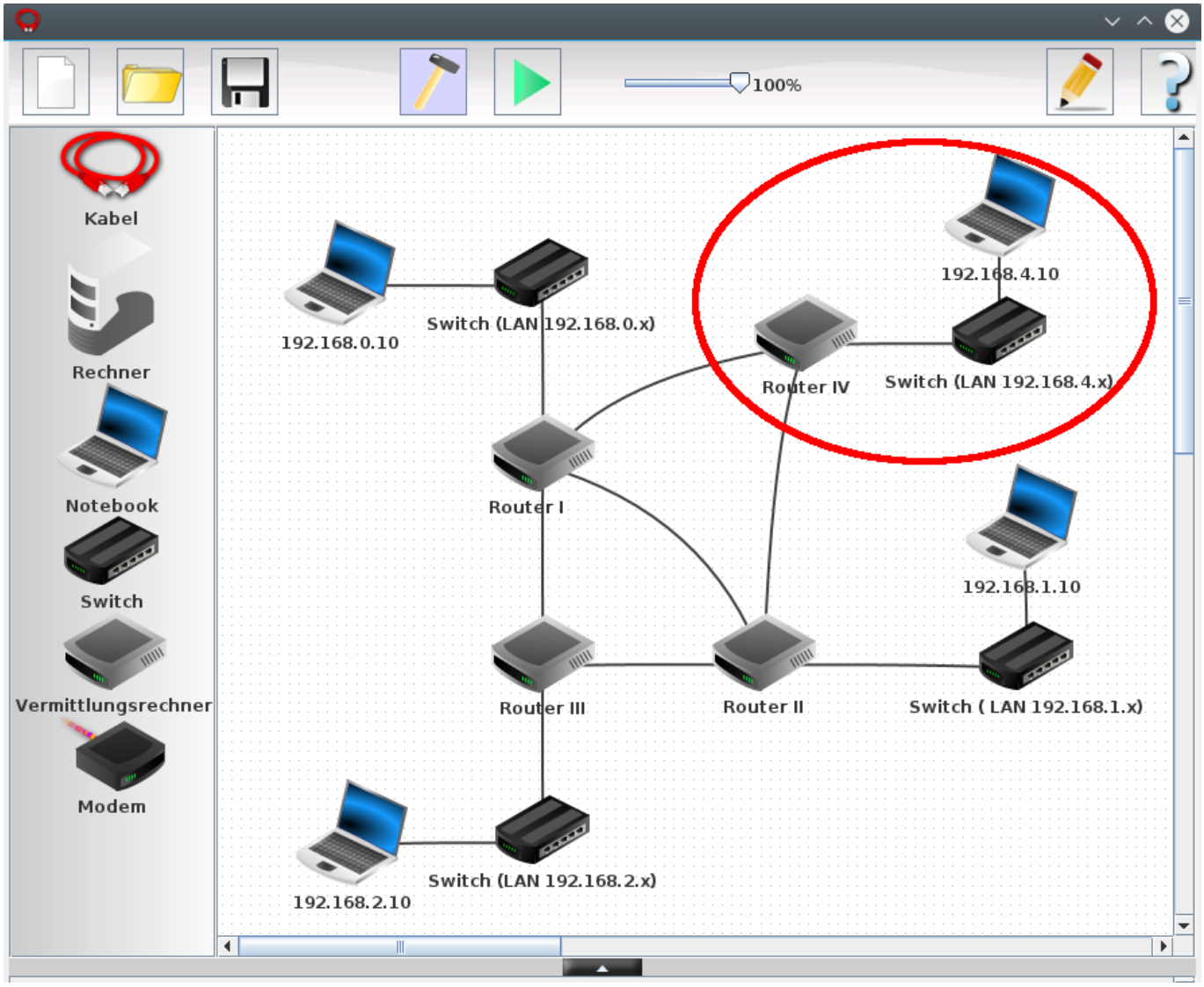
a) Öffne in Filius die Datei filius_mehrere_netze.fls. Die Routingtabelle eines Routers findest du in dessen Einstellungen unter dem Punkt „Weiterleitungstabelle“. Entferne hier zum Betrachten und Bearbeiten der Tabelle stets das Häkchen bei der Einstellung „Alle Einträge anzeigen“.

Wie unterscheiden sich die Routingtabellen von Router I und II in Filius von der Beispieltabelle zu Aufgabe 1? Kannst du die Unterschiede deuten? Mache auch Verbindungstests mit dem ping-Befehl zwischen den Rechnern 192.168.0.10, 192.168.1.10 und 192.168.2.10.

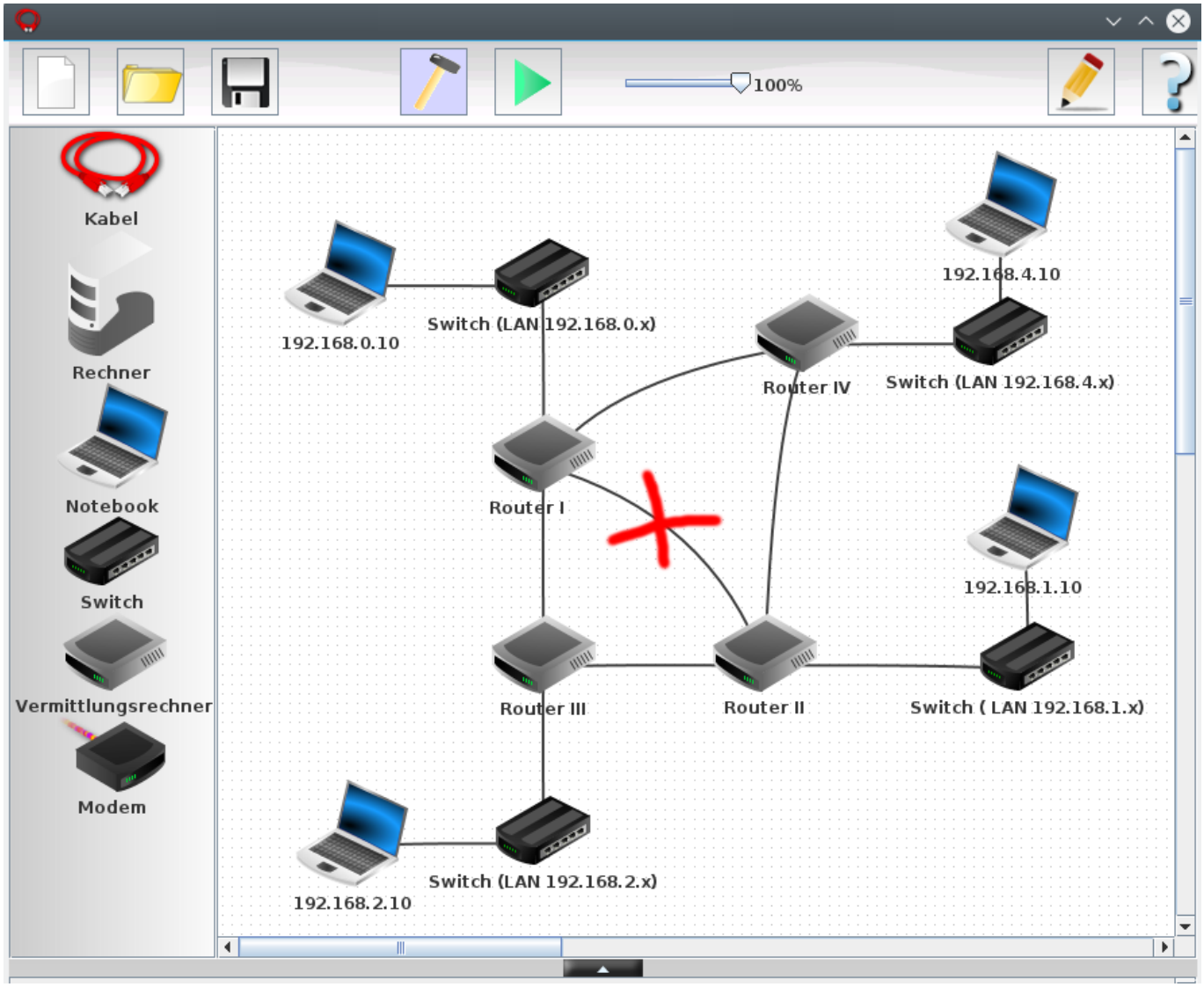
(b) Der Rechner 192.168.2.10 antwortet noch nicht auf ping-Anfragen der anderen Rechner, da die Routingtabelle von Router III noch nicht konfigueriert ist. Füge die fehlenden Einträge hinzu.

Aufgabe 3)

Erweitere das Rechnernetz aus Aufgabe 2 wie abgebildet. Ergänze auch die Routingtabellen der Router so, dass Rechner 192.168.4.10 von den anderen Rechnern aus erreichbar ist.


100%
Kabel
Rechner
Notebook
Switch
Vermittlungsrechner
Modem
192.168.0.10
Switch (LAN 192.168.0.x)
192.168.4.10
Router IV
Switch (LAN 192.168.4.x)
Router I
192.168.1.10
Router III
Router II
Switch (LAN 192.168.1.x)
192.168.2.10
Switch (LAN 192.168.2.x)

Aufgabe 4)



a) Entferne das in der Abbildung markierte Kabel aus deinem Rechnernetz. Warum schlägt nun der Verbindungstest zwischen den Rechnern 192.168.0.10 und 192.168.1.10 trotz zweier alternativer Routen fehl?

b) Passe die Routingtabellen der Router so an, dass Nachrichten zwischen den Rechnern 192.168.0.10 und 192.168.1.10 über eine der Alternativrouten gesendet werden.

Aufgabe 5)

Löse das Rätsel

Weitere Beispiele

- Routing-Beispiele

Dynamisches Routing

Da es in Rechnernetzen regelmäßig Änderungen gibt - sei es das Einfügen eines neuen Rechners oder

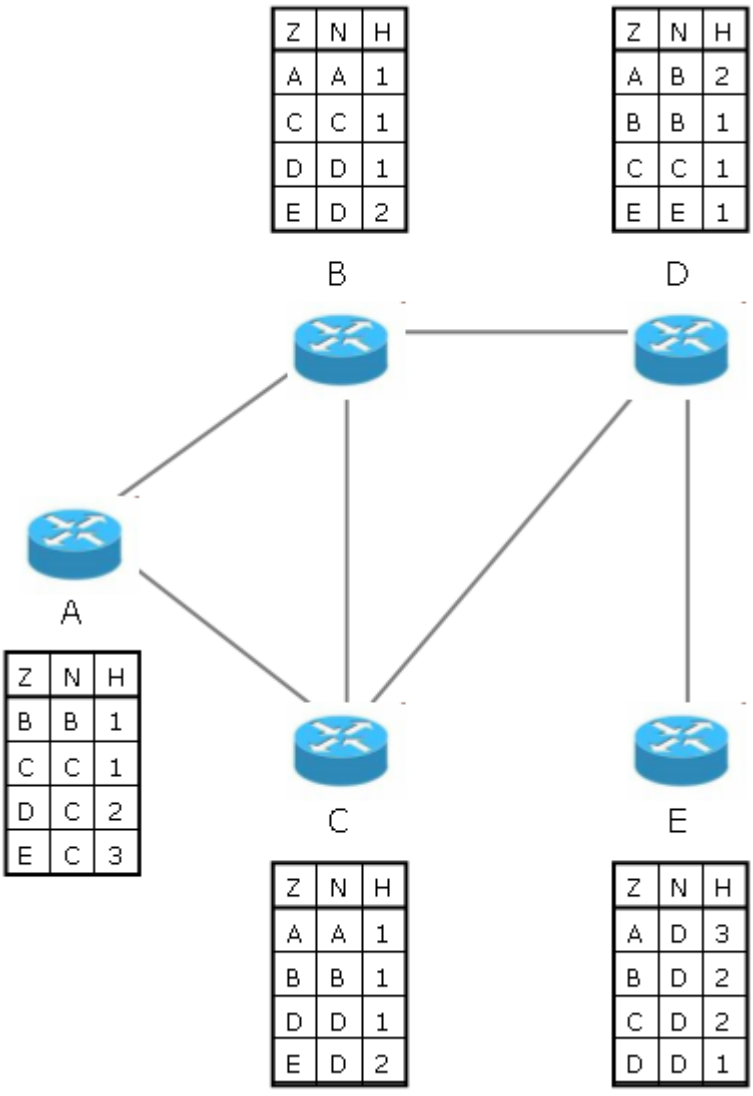
der Ausfall einer Verbindung - müssen auch die Routingtabellen der Router immer wieder angepasst beziehungsweise neu erstellt werden. Es liegt auf der Hand, dass ein Administrator eines großen Rechnernetzes schnell überfordert ist, wenn er regelmäßig alle Routingtabellen von Hand auf den neusten Stand bringen muss. Abhilfe schafft hier ein Routingprotokoll, nach dem die Router ihre Routingtabellen selbst dynamisch ermitteln.

In Filius kannst du das dynamische Ermitteln einer Routingtabelle in der Konfiguration des Routers durch das setzen des Häkchens „Automatisches Routing“ aktivieren. Die Routingtabelle ist in Filius dann jedoch nicht mehr einsehbar.

Wie ein Routingprotokoll zum dynamischen Erstellen von Routingtabellen funktioniert, kannst du im Kapitel „Vermittlung von Datenpaketen“ entlang eines Rollenspiels erkunden:

Wie findet ein Datenpaket seine Route im Netz der Vermittlungseinheiten?

Die folgende Abbildung zeigt ein einfaches Netz aus Routern, die der Einfachheit halber statt über IP-Adressen mit den Buchstaben A, B, … identifiziert werden.



Jeder Router verfügt über eine Routingtabelle, in der genau festgelegt ist, wie ein Datenpaket weitergeleitet wird.

Die Routingtabelle ist folgendermaßen zu lesen: Zum Zielknoten Z gelangt ein Datenpaket über den

Nachbarknoten N mit einer bestimmten Anzahl H von Schritten / Hops. Mit Hilfe solcher Routingtabellen wird ein Datenpaket, das sich z.B. aktuell im Knoten A befindet und zum Zielknoten E weitertransportiert werden soll, zum Nachbarknoten C weitervermittelt. Beachte, dass das Datenpaket mit der gleichen Anzahl von Hops auch an den Nachbarknoten B vermittelt werden könnte. In der Routingtabelle ist eine von mehreren günstigsten Möglichkeiten vermerkt.

Erstellung von Routingtabellen

Routingtabellen werden dynamisch von den Routern nach dem folgenden Routingprotokoll ermittelt.

Wir gehen davon aus, dass jeder Router seine Nachbarknoten kennt. In einem ersten Schritt trägt jeder Router die direkt zu erreichenden Nachbarn in seiner Routingtabelle ein.

Beispiel:

Routingtabelle von Router A:

```
Z N H
=====
B B 1
C C 1
```

Jetzt beginnt der Informationsaustausch zwischen den Routern. Jeder Router sendet eine Kopie seiner aktuellen Tabelle an alle seine Nachbarknoten.

Router B sendet folgende Tabelle an Router A:

```
Z N H
=====
A A 1
C C 1
D D 1
```

Router C sendet folgende Tabelle an Router A:

```
Z N H
=====
A A 1
B B 1
D D 1
```

Router A verarbeitet die Informationen, die in den ihm übermittelten Tabellen stecken. Z.B. „weiß“ Router A jetzt, dass Router D über Router C mit insgesamt 2 Hops erreichbar ist. Router A „weiß“ auch, dass Router D über Router B mit insgesamt 2 Hops erreichbar ist. Nach internen Strategien (wie z. B der Reihenfolge der verarbeiteten Tabellen) erstellt Router A jetzt eine neue Tabelle mit seinem aktuellen Wissen.

Routingtabelle von Router A:

```
Z N H
=====
B B 1
C C 1
D C 2
```

Jetzt beginnt der nächste Schritt der Informationsweitergabe. Router A sendet seine neue Tabelle wieder an alle seine Nachbarn und empfängt im Gegenzug von seinen Nachbarn deren aktuelle Tabellen. Wenn sich zusätzliche Informationen in diesen übermittelten Tabellen befinden, dann werden diese Informationen in die eigenen Tabellen übernommen.

Dieser Prozess der Informationsweitergabe führt in der Regel nach wenigen Schritten zu stabilen Tabellen.

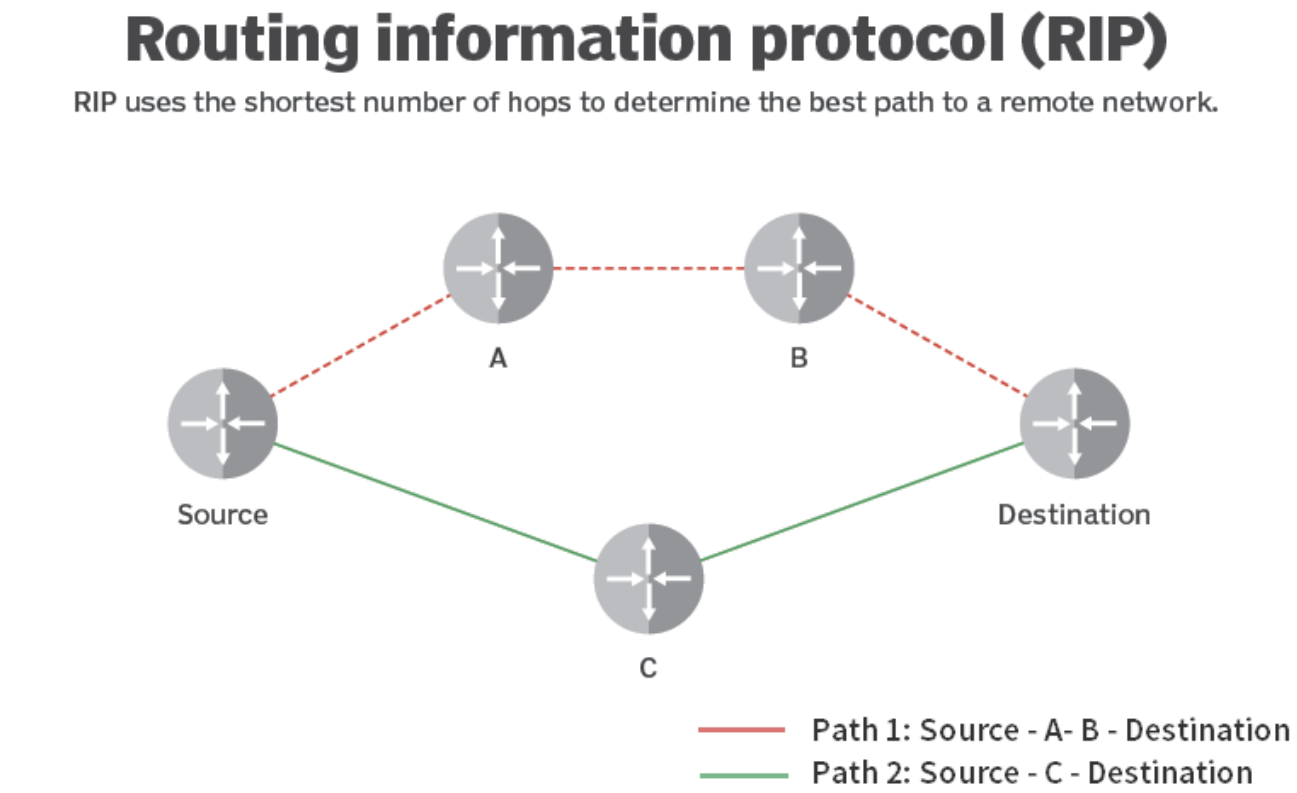
Diese Tabellen werden trotzdem in einem festgelegten Rhythmus aktualisiert. Wenn eine Verbindungsleitung z.B. ausfällt, dann müssen die Routen ggf. dynamisch angepasst werden. Das geschieht, indem die Tabellen durch neu übermittelte Informationen aktualisiert werden.

Kategorien von dynamischen Routingprotokollen

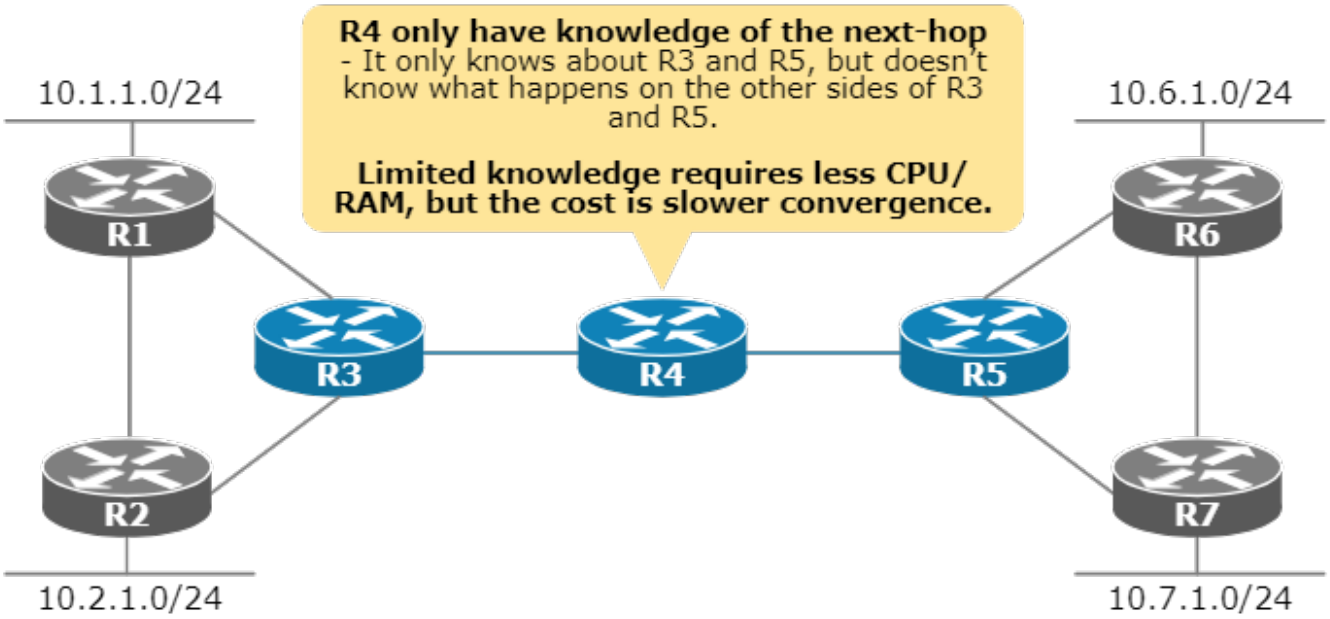
Distance Vectoring

Distance-Vector-Protokolle basieren auf der Idee, den besten Pfad zu einem Zielnetzwerk basierend auf der Entfernung (Anzahl der Hops) und der Richtung (Vektor) zum Ziel zu wählen. Router, die Distance-Vector-Protokolle verwenden, teilen ihre Routingtabellen nur mit ihren direkten Nachbarn. Bei jedem Update wird die Entfernung zu jedem Zielnetzwerk und die Richtung (der nächste Router auf dem Pfad) übertragen.

Ein bekanntes Beispiel für ein Distance-Vector-Protokoll ist das Routing Information Protocol (RIP). Distance-Vector-Protokolle sind jedoch anfällig für Routing-Loops und können in großen Netzwerken langsam konvergieren.



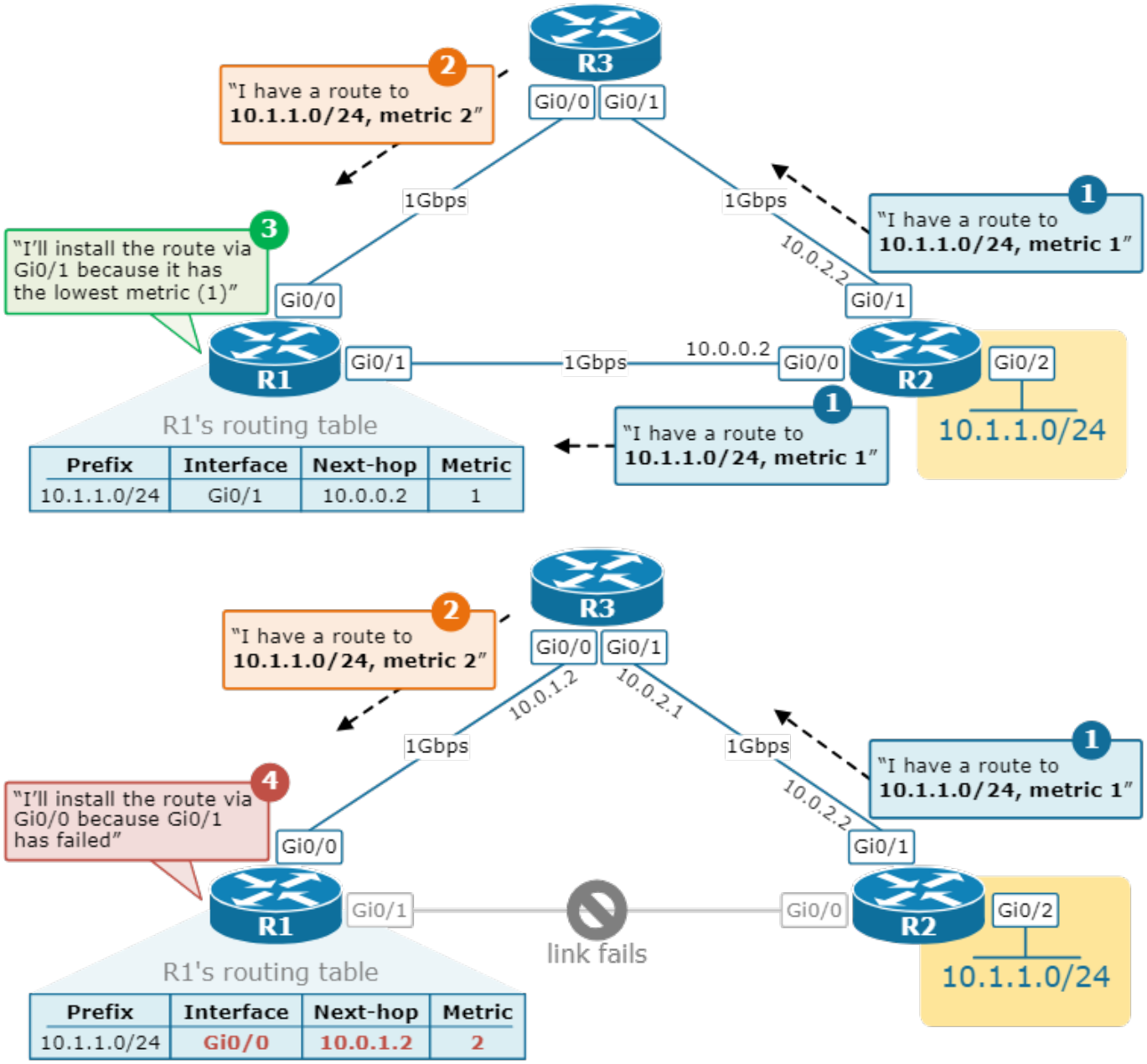
Router R4 kennt nur seine Nachbarn und die Distanzvektoren (Hops) zu den jeweiligen Netzwerken



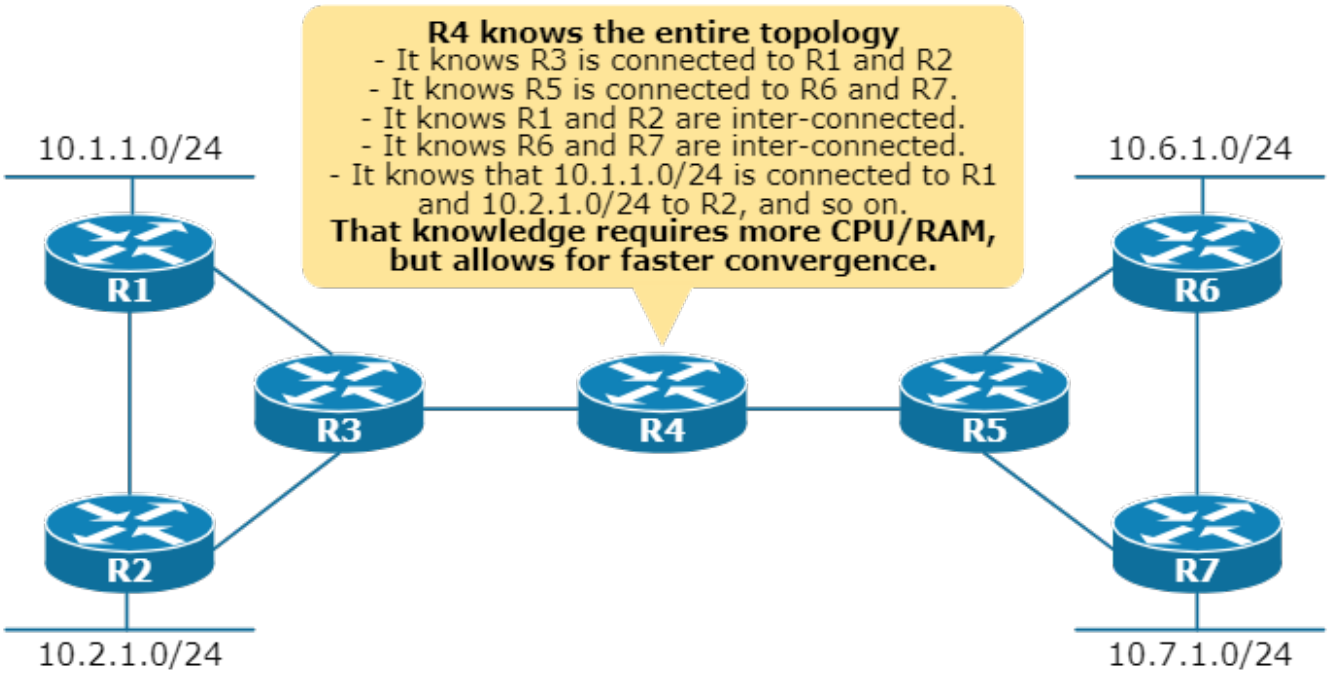
Link State

Link-State-Protokolle funktionieren anders als Distance-Vector-Protokolle. Anstatt nur Entfernungen und Richtungen mit direkten Nachbarn auszutauschen, teilen Router, die Link-State-Protokolle verwenden, Informationen über ihre direkten Verbindungen (Links) und deren Zustand mit allen anderen Routern im Netzwerk. Jeder Router erstellt daraus eine vollständige Topologiekarte des Netzwerks und berechnet den kürzesten Pfad zu jedem Zielnetzwerk mithilfe von Algorithmen wie dem Dijkstra-Algorithmus.

Ein bekanntes Beispiel für ein Link-State-Protokoll ist das Open Shortest Path First (OSPF) Protokoll. Link-State-Protokolle sind weniger anfällig für Routing-Loops und konvergieren in der Regel schneller als Distance-Vector-Protokolle.



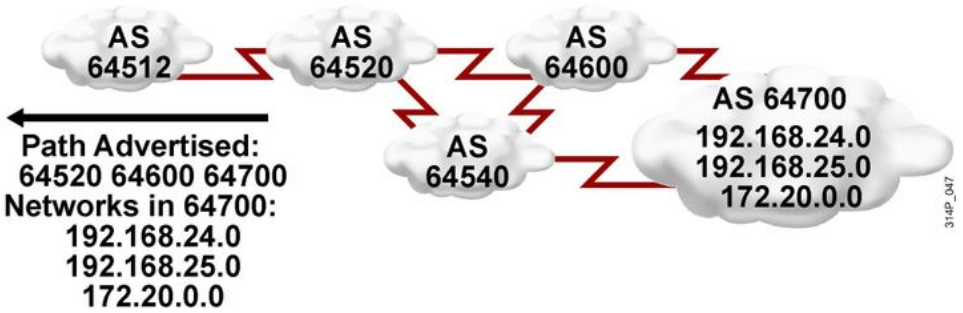
Router R4 kennt die ganze Topologie



Path Vector

Path-Vector-Protokolle sind eine Erweiterung der Distance-Vector-Protokolle und teilen Informationen über den gesamten Pfad (eine Sequenz von Netzwerkadressen) zum Zielnetzwerk. Jeder Router aktualisiert und speichert Informationen über die Pfade zu verschiedenen Zielen und trifft Routingentscheidungen basierend auf diesen Pfaden.

BGP Path-Vector Routing



- **IGPs announce networks and describe the metric to reach those networks.**
- **BGP announces paths and the networks that are reachable at the end of the path. BGP describes the path by using attributes, which are similar to metrics.**
- **BGP allows administrators to define policies or rules for how data will flow through the autonomous systems.**

Ein Beispiel für ein Path-Vector-Protokoll ist das Border Gateway Protocol (BGP). Path-Vector-Protokolle sind in der Regel besser skalierbar und vermeiden Routing-Loops, die in Distance-Vector-Protokollen auftreten können.

Übersicht der Routingprotokolle

Routingprotokoll	Eigenschaften
RIP (Routing Information Protocol)	**Distance-Vector-Protokoll** • Anzahl der Hops als Metrik • Einfach, aber anfällig für Routing-Loops • Begrenzt auf 15 Hops
OSPF (Open Shortest Path First)	**Link-State-Protokoll** • Dijkstra-Algorithmus zur Berechnung des kürzesten Pfades • Schnellere Konvergenz als RIP • Skalierbar und weniger anfällig für Routing-Loops
EIGRP (Enhanced Interior Gateway Routing Protocol)	**Hybrid aus Distance-Vector und Link-State** • Metrik basierend auf Bandbreite und Verzögerung • Schnelle Konvergenz • Proprietäres Cisco-Protokoll, aber teilweise offengelegt
BGP (Border Gateway Protocol)	**Path-Vector-Protokoll** • Verwendet vollständige Pfade (Sequenzen von Netzwerkadressen) • Skalierbar und eignet sich für große Netzwerke wie das Internet • Kann Routing-Policies und -Regeln anwenden

Dienste im Internet

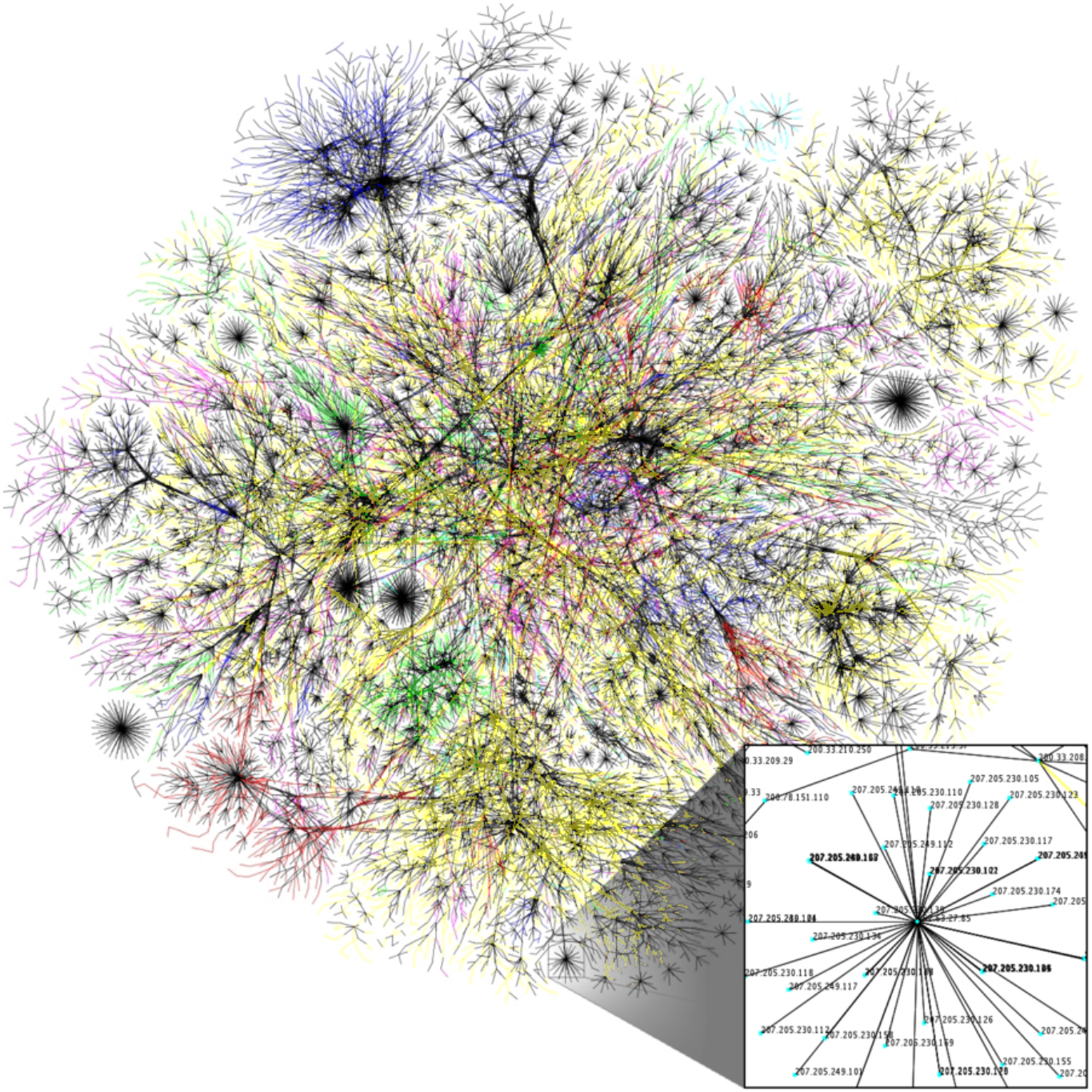
Das Internet als weltweiter Verbund von Rechnernetzen ist für uns zum alltäglichen Kommunikationsmedium geworden. Zum Leben erweckt wird das Internet durch seine Dienste wie das World Wide Web zum Übertragen von Webseiten oder der E-Mail-Dienst. In diesem Abschnitt kannst du die Kommunikation zwischen Anwendungsprozessen, die diese Internet-Dienste nutzen, simulieren und analysieren.

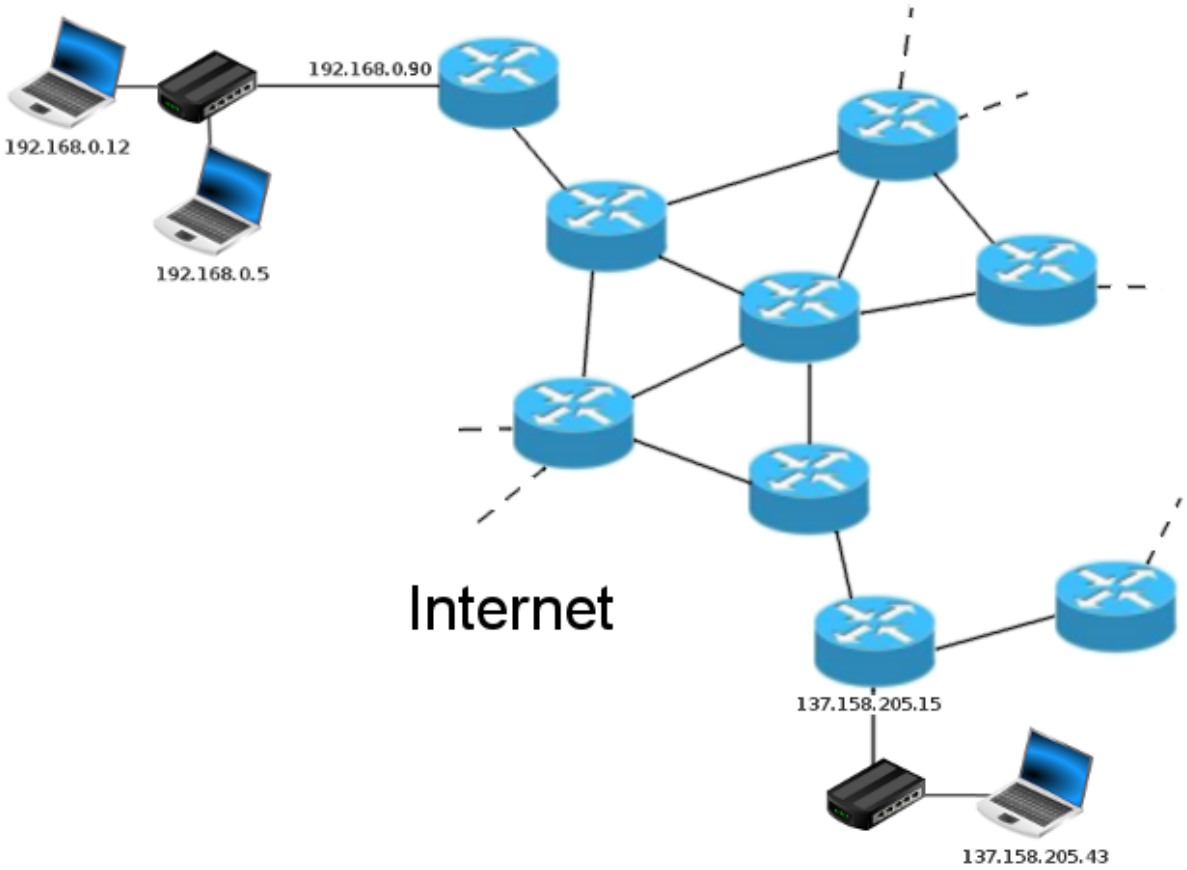
Möchtest Du exemplarisch die E-Mail-Protokolle SMTP und POP3 mit Hilfe von Filius und RFC-Auszügen untersuchen, so kannst Du dies in dem von hier abgetrennten Kapitel E-Mail-Protokolle tun.

Grundstruktur des Internets

Von Rechnernetzen mit verschiedenen Teilnetzen, die über Router miteinander verbunden sind, ist es nur noch ein kleiner Schritt bis zum Grobkonzept des Internets. Hier sind eine große Menge von Routern weltweit miteinander verbunden und diese wiederum verbinden eine große Menge lokaler Netzwerke miteinander.

Hier eine Karte von einem Teil des Internets, ermittelt von einem Datensammelprogramm:


207.205.230.105
207.205.230.110
207.205.230.128
207.205.230.117
207.205.230.102
207.205.230.174
207.205.230.118
207.205.249.117
207.205.230.126
207.205.230.155
207.205.249.101



Dienste im Internet

Das Internet selbst stellt lediglich die Infrastruktur zur Übertragung der Daten zur Verfügung. Ein Nutzen entsteht erst dadurch, dass einem in dieser Infrastruktur verschiedene Dienste zur Verfügung stehen. Die bekanntesten Internetdienste sind das World Wide Web zum Übertragen von Webseiten und der E-Mail-Dienst.

Webserver

Das **World Wide Web** (kurz www) ist ein Dienst des Internets, der für die Übertragung von Webseiten zuständig ist. Gleichzeitig steht das World Wide Web als **Hypertext-System (Hypermedia-System)** für die Struktur von weltweit miteinander verknüpften Webseiten und Multimedia-Inhalten.

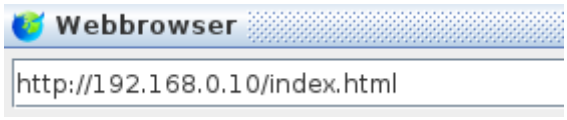
Web-Client und Web-Server

Zum Anzeigen von Webseiten auf einem Client und zur Navigation im Hypertext-System der Webseiten benötigt man ein spezielles Programm, den sogenannten **Browser**.

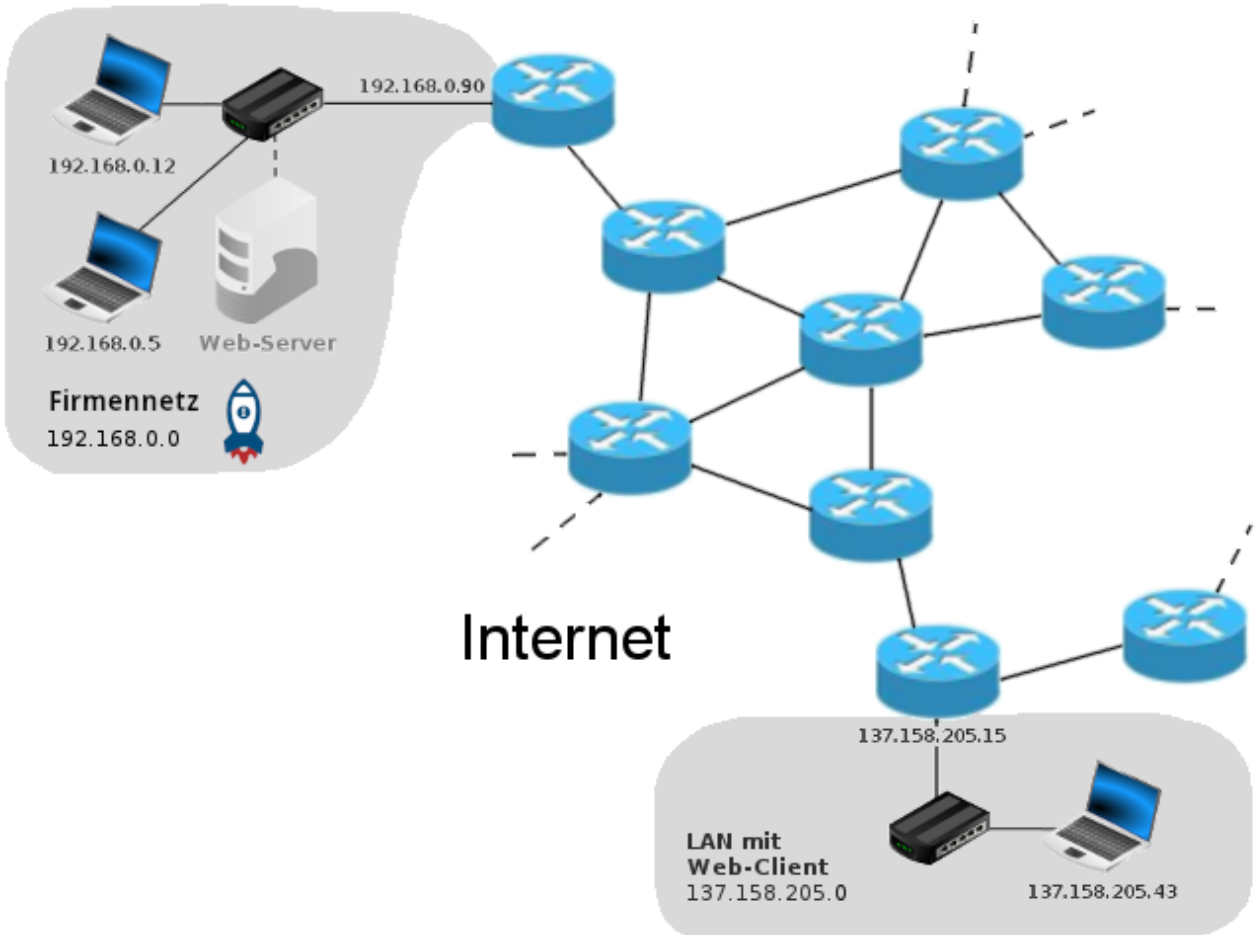
Der Browser fordert die Webseite vom adressierten Web-Server an. Auf diesem - in der Regel im Internet entfernt liegenden - Rechner läuft eine Web-Server-Software, die für die Auslieferung der Webseite zuständig ist.

Adressierung von Webseiten

Will man eine bestimmte Webseite im Browser anzeigen, so kann man in der Adresszeile direkt die Web-Adresse dieser Webseite eingeben. Web-Adressen sind nach einem bestimmten Muster aufgebaut:



Protokoll:*Rechneradresse/Dokumentenadresse* *In unserem Simulationsaufbau ist das lokale Netzwerk des Unternehmens aus Platzgründen nur noch klein in der linken oberen Ecke abgebildet. Zudem ist beispielhaft am unteren Rand ein weiteres lokales Netzwerk mit einem Rechner an das Internet angebunden. Von hier aus soll die Webseite aufgerufen werden.*



*Eine Adressbeschreibung der Gestalt Protokoll:*Rechneradresse/Dokumentenadresse nennt man **URL**

(Kurzform von Uniform Resource Locator). Mit solchen Adressbeschreibungen können Web-Dokumente eindeutig im Internet lokalisiert und damit auch identifiziert werden.

Jede Web-Adresse (URL) beginnt mit dem Protokoll, das für das Anfordern und Ausliefern der Webseite zuständig ist. Hier wird standardmäßig das Protokoll HTTP oder dessen Variante HTTPS zur verschlüsselten Übertragung eingesetzt.

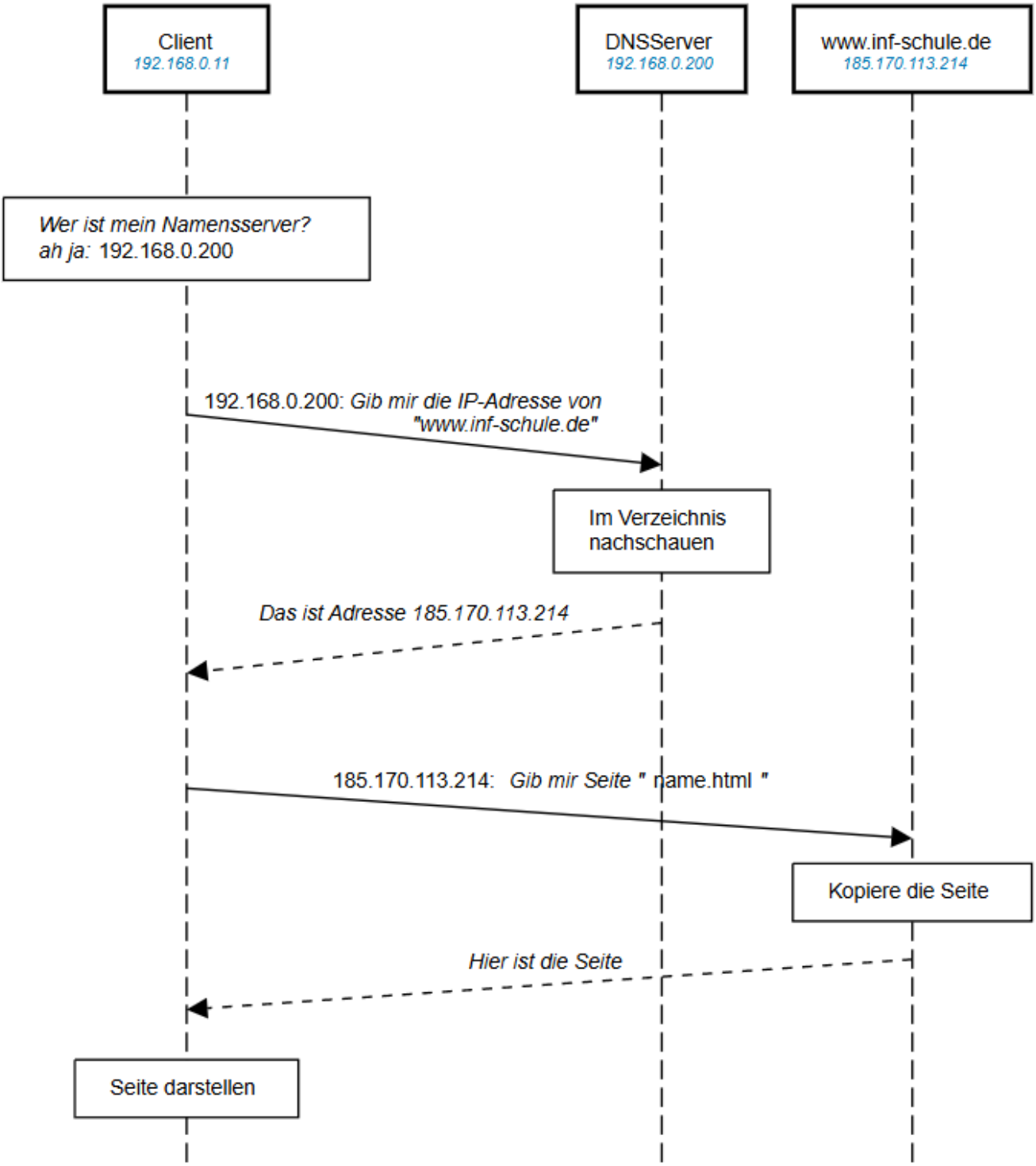
Namensauflösung mit DNS-Server

Da man sich eine Rechneradresse in Form einer IP-Adresse nur schwer merken kann, ist ihr häufig (mindestens) ein Domainname zugeordnet, durch den sie in der Web-Adresse (URL) ersetzt werden kann.

Die Zuordnung wird auf einem sogenannten **Domain Name System-Server** (kurz DNS-Server) festgehalten. Beim Aufrufen der Webseite wird dieser kontaktiert, um den eingegebenen Domainnamen in die zugeordnete IP-Adresse zu übersetzen. Diese Übersetzung wird Namensauflösung genannt.

Ablauf der Client-Server-Kommunikation als Sequenzdiagramm

Wenn du eine Web-Adresse (hier www.inf-schule.de) im Webbrowser eingibst, läuft im Hintergrund der folgende Prozess ab:



1) Der Client schaut in seinen Einstellungen nach, welcher DNS-Server für ihn zuständig ist

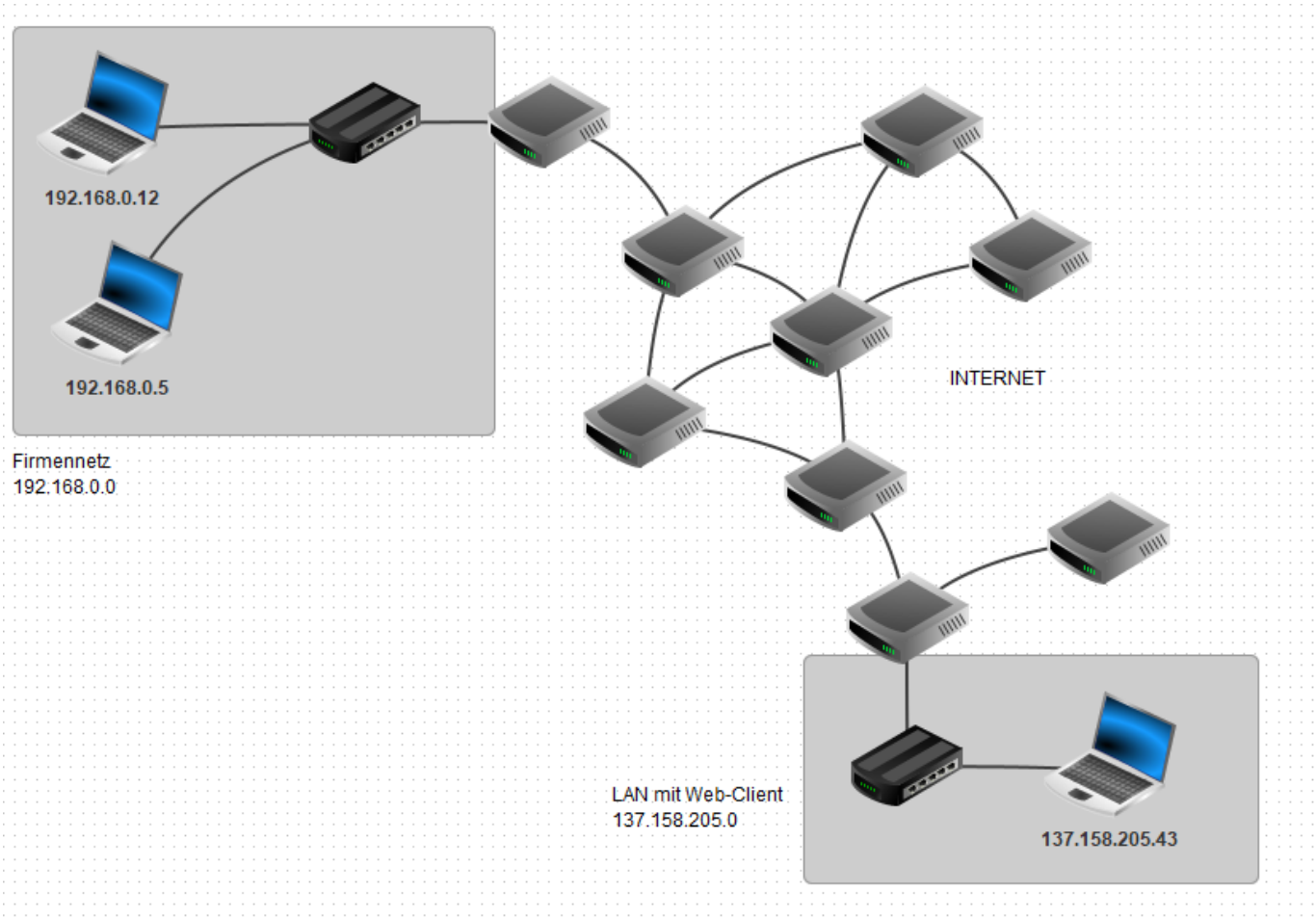
2) Der Client muss zunächst beim DNS-Server nachfragen, welche IP-Adresse zum jeweiligen Namen gehört.

3) Der Client kann nun die Webseite unter ihrer IP-Adresse abrufen (ohne dass der Benutzer das überhaupt sieht).

Aufgabe 1 - Web Server

a) Aufbau Infrastruktur

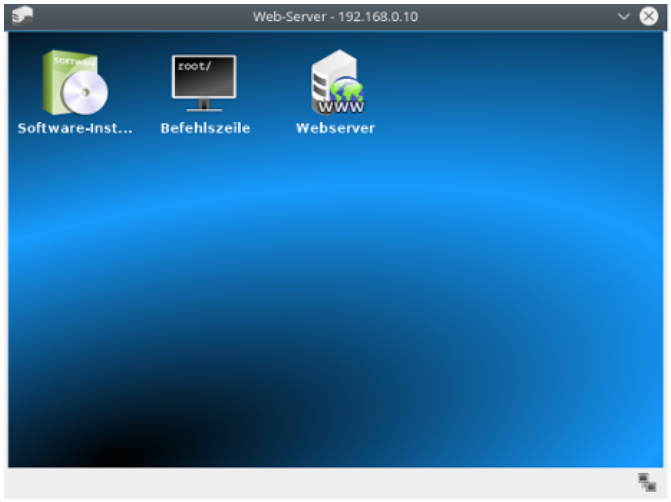
Öffne in Filius die Datei filius_webserver.fls und erstelle im lokalen Netzwerk deines Start-Up-Unternehmens einen Server. Teste mit dem Ping-Befehl, ob du den Server über das Internet von dem Rechner mit der IP-Adresse 137.158.205.43 aus erreichen kannst.

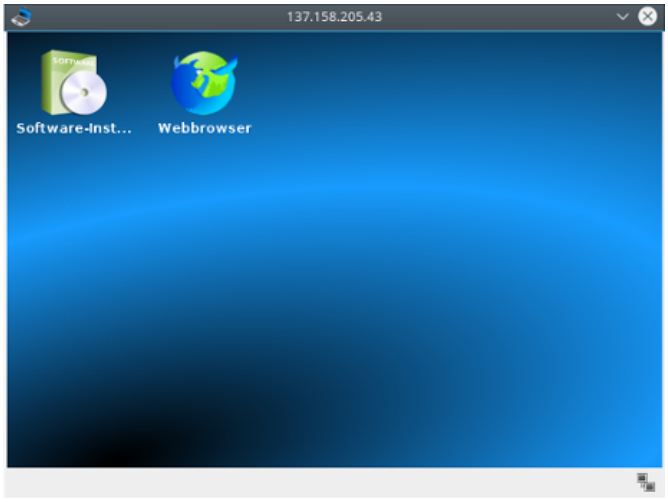


b) Installation von Software

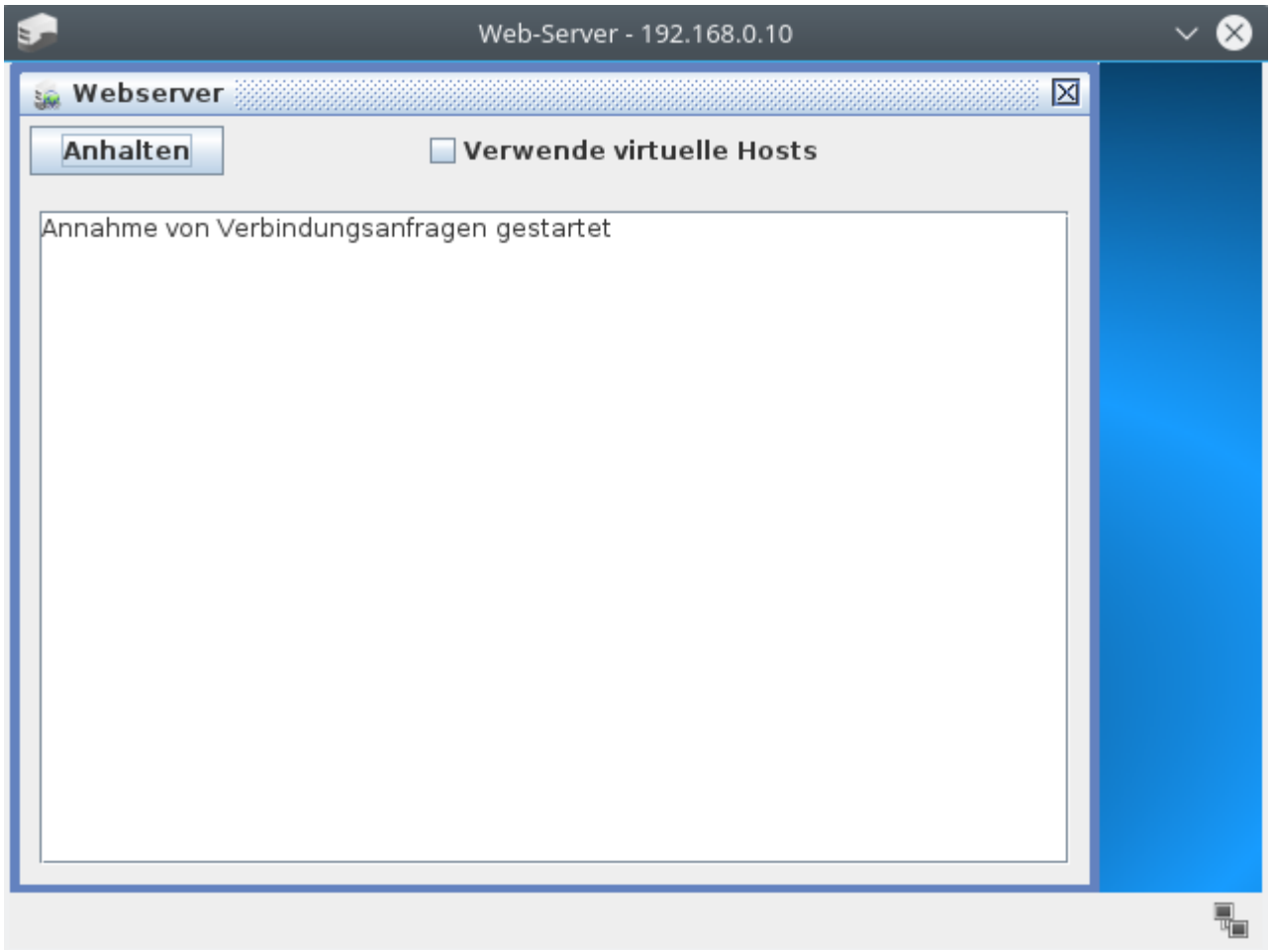
Der Server soll nun als Web-Server eine Webseite anbieten und der Rechner mit der IP-Adresse 137.158.205.43 soll diese als Web-Client mit einem Browser abrufen.

Dazu wird auf dem Server eine Web-Server-Software und auf dem Client ein Webbrowser installiert.





Zunächst muss der Web-Server gestartet werden:



Jetzt kann man auf dem Client den installierten Webbrowser starten und eine http-Anfrage mit der korrekten IP-Adresse des Web-Servers formulieren. Wenn alles korrekt eingegeben ist, erhält man folgende Webseite.

c) Analysiere den Datenaustausch beim Aufruf der Webseite

d) Erstelle eine eigene Webseite auf dem Web-Server

- Installiere auf dem Web-Server einen Datei-Explorer. Untersuche mit diesem Explorer, wo sich die Dateien zur angezeigten Webseite befinden.
- Installiere auf dem Web-Server zusätzlich einen Text-Editor. Schaue dir mit diesem Editor den Quelltext zur Webseite an.
- Erstelle selbst mit dem Text-Editor eine einfache Webseite für dein persönliches Start-up-Unternehmen. Mit dem Datei-Explorer kannst du gegebenenfalls Bilder auf den Server „hochladen“. Teste deine Webseite, indem du sie vom Webbrowser auf dem Client anzeigen lässt.

- Hinweis: Du kannst auch mehrere Webseiten unter verschiedenen Namen abspeichern und miteinander verlinken. Lässt man beim Aufruf im Webbrowser den Dateinamen weg, so wird automatisch die Datei mit dem Namen „index.html“ aufgerufen.

Aufgabe 2 - DNS-Server

Beim Aufruf Deiner Webseite in Filius musstest Du bisher die IP-Adresse des Web-Servers, auf dem die Dateien zur Webseite gespeichert sind, angeben. Wenn wir im Internet Webseiten aufrufen, benutzen wir in der Regel jedoch Domainnamen wie www.inf-schule.de.

Damit das funktioniert, benötigt man einen DNS-Server:

a) Konfiguriere den DNS Server laut Video

DNS-Server Konfiguration

b) Der DNS-Server mit der im Video verwendeten IP-Adresse 5.9.164.112 existiert auch in der realen Welt. Informationen dazu kannst Du durch Aufruf der IP-Adresse über die Adresszeile Deines Webbrowsers einholen. Informiere Dich über das Thema DNS-Sperren und sammele Gründe für und gegen deren Einsatz.

Aufgabe 3 - Echo-Server

Installiere auf dem Web-Server zusätzlich die Echo-Server-Software. Starte beide Server-Prozesse - den Echo-Server und den Web-Server und teste beide, indem du ihre Dienste von Clients in Anspruch nimmst.

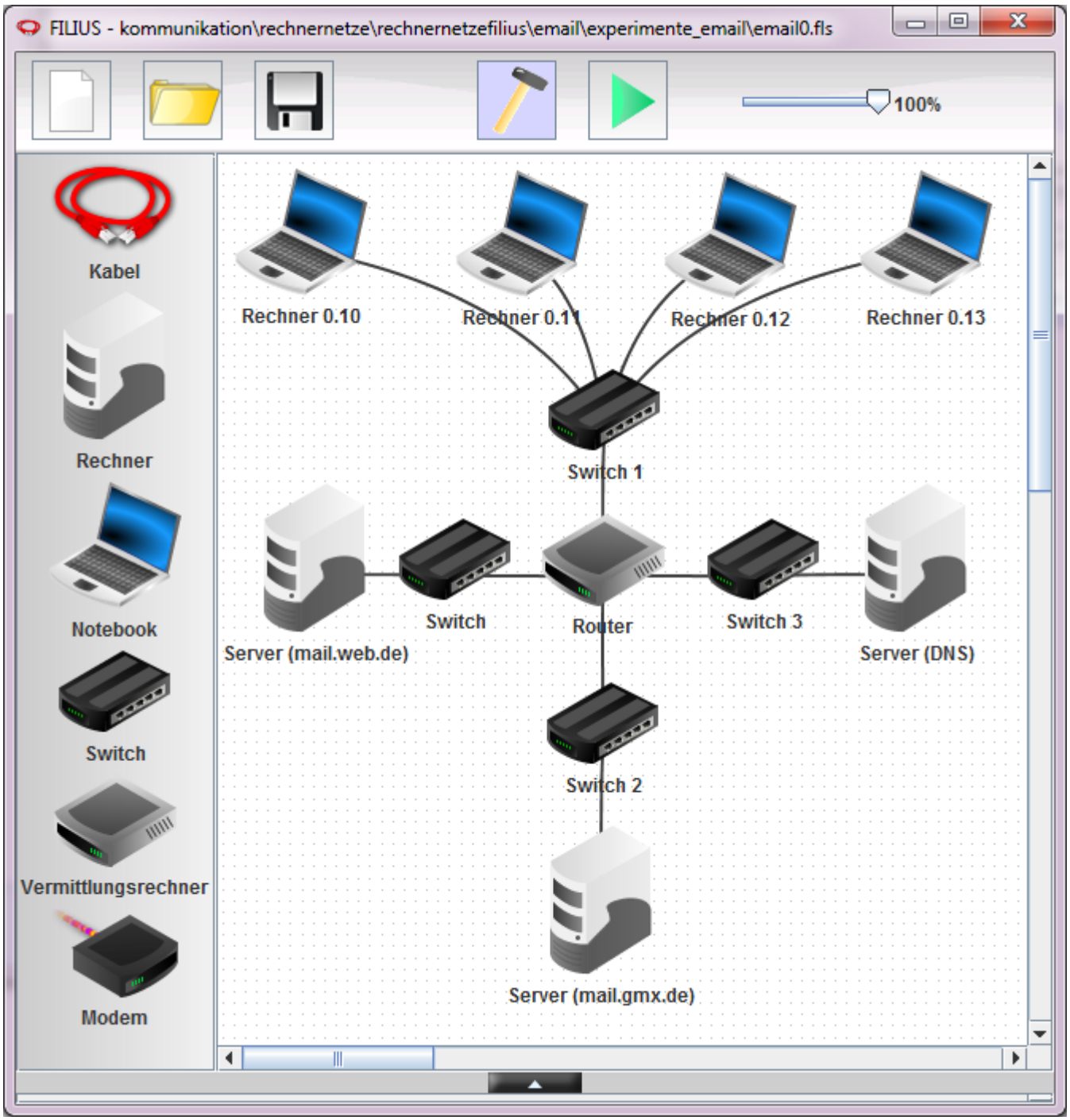
Beobachte auch den Datenaustausch. Zusätzlich zur IP-Adresse wird auch jeweils auch eine Portnummer angegeben. Wozu könnten die Portnummern gut sein?

E-Mail Dienst

Filius ermöglicht die Simulation des E-Mail-Dienstes, welche auf dieser Seite näher erklärt wird.

1) Aufbau des Rechnernetzes

Das Rechnernetz, das im Folgenden benutzt werden soll, hat folgenden Aufbau:



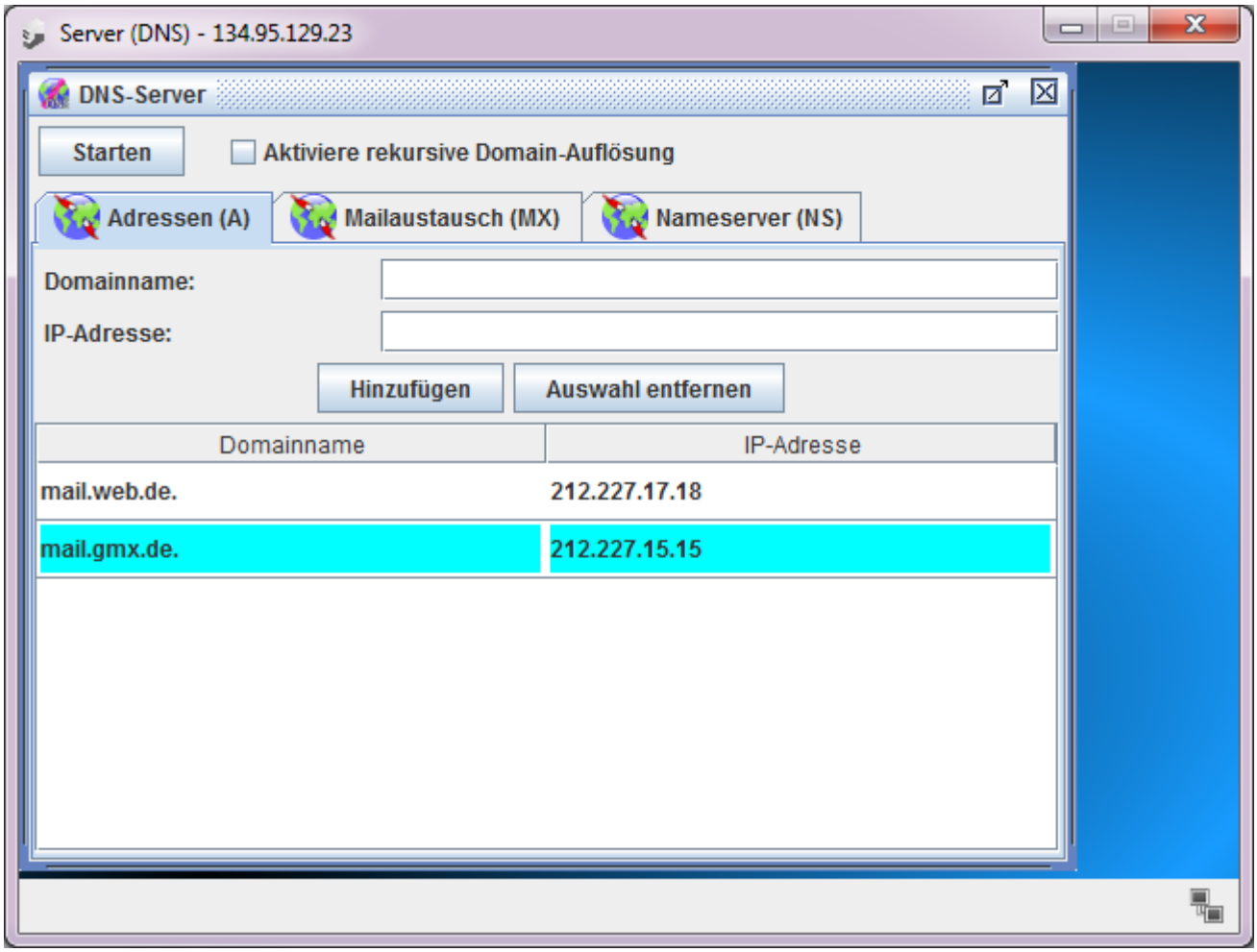
Verwende dafür diese E-Mail-Vorlage

Neben einem Rechner, auf dem ein DNS-Server installiert ist, gibt es zwei weitere Rechner, auf denen

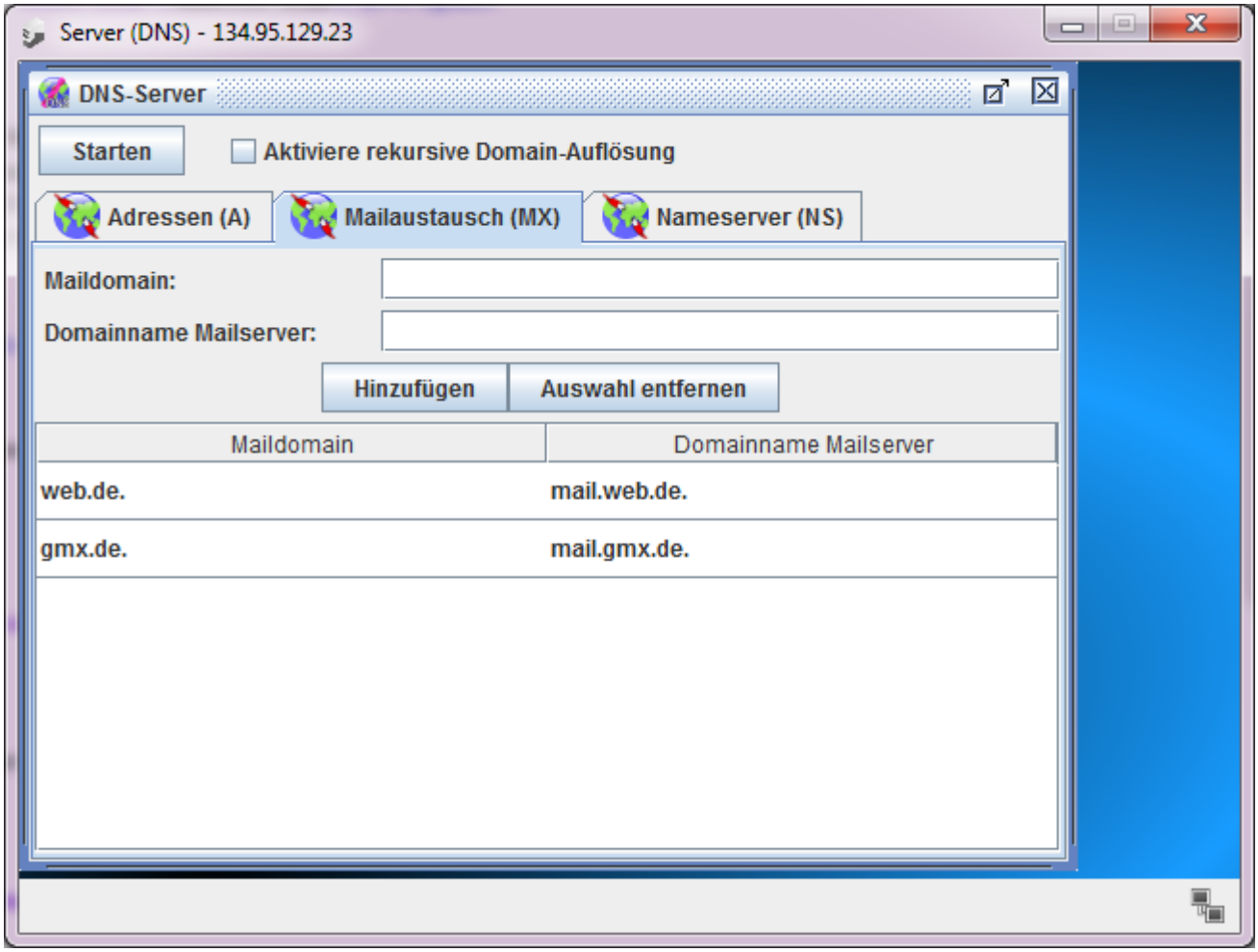
E-Mail-Server installiert werden: ein Rechner mit dem Domainnamen mail.web.de und der IP-Adresse 212.227.17.18 sowie ein Rechner mit dem Domainnamen mail.gmx.de und der IP-Adresse 212.227.15.15.

2) Konfiguration des DNS-Servers

Auf dem DNS-Server werden zunächst die beiden Rechner mit ihren Domainnamen eingetragen.



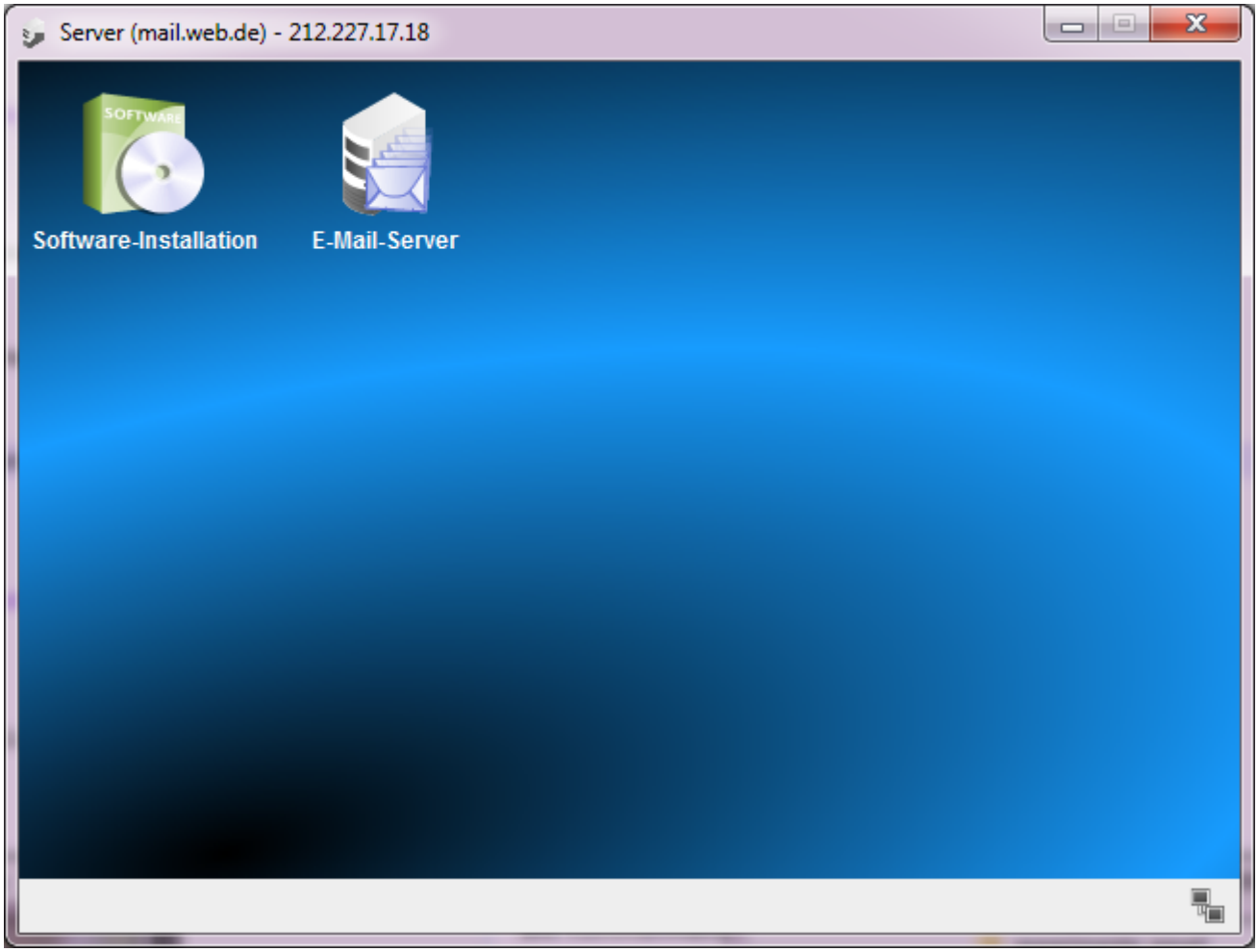
Zusätzlich werden die Maildomains hier im Fenster Mailaustausch eingetragen.



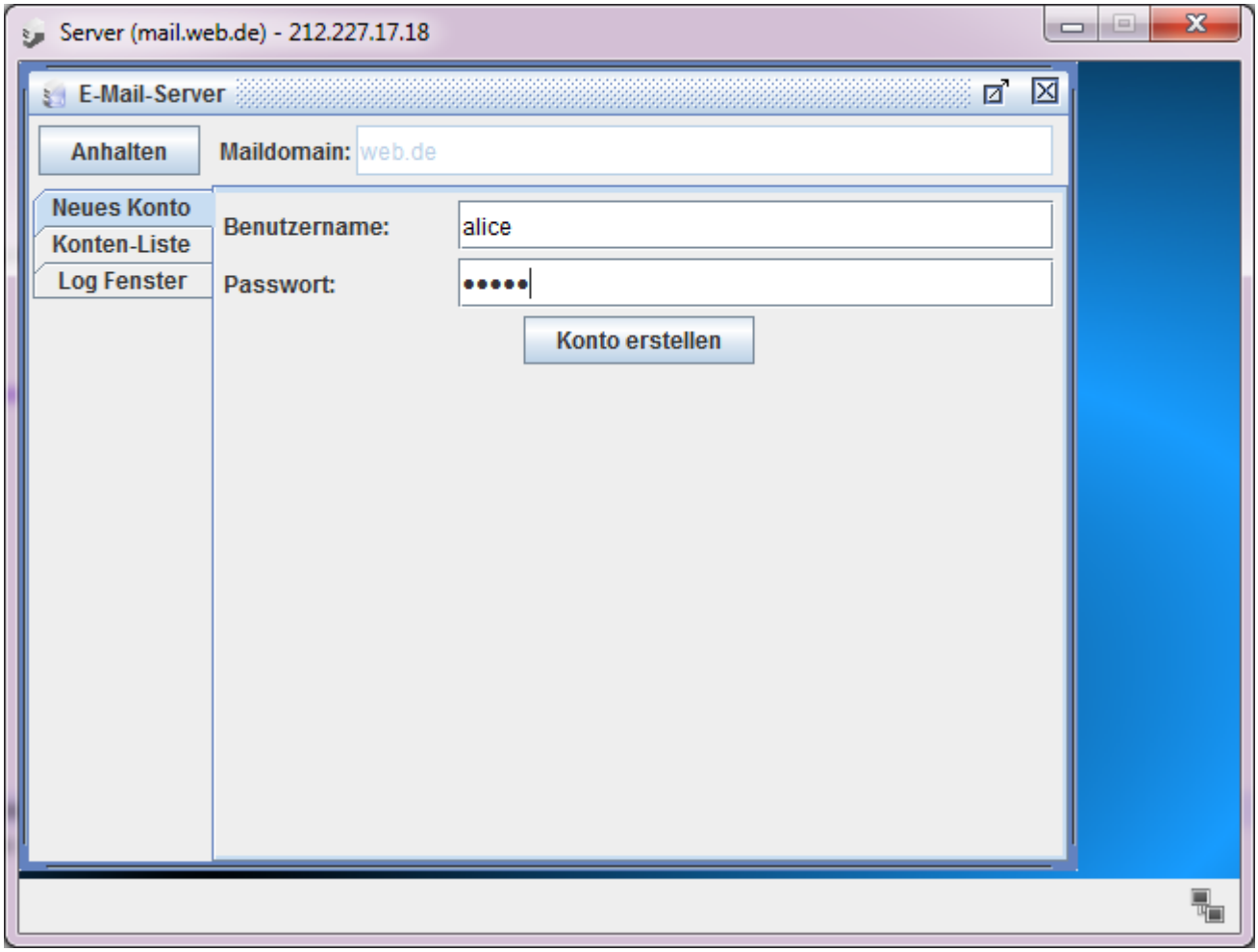
Jetzt kann der DNS-Server gestartet werden.

3) Installation und Konfiguration des E-Mail Servers

Auf den Rechnern mail.gmx.de und mail.web.de wird jetzt die Mail-Server-Software installiert.



Auf beiden Mail-Servern lassen sich neue Benutzerkonten erstellen.



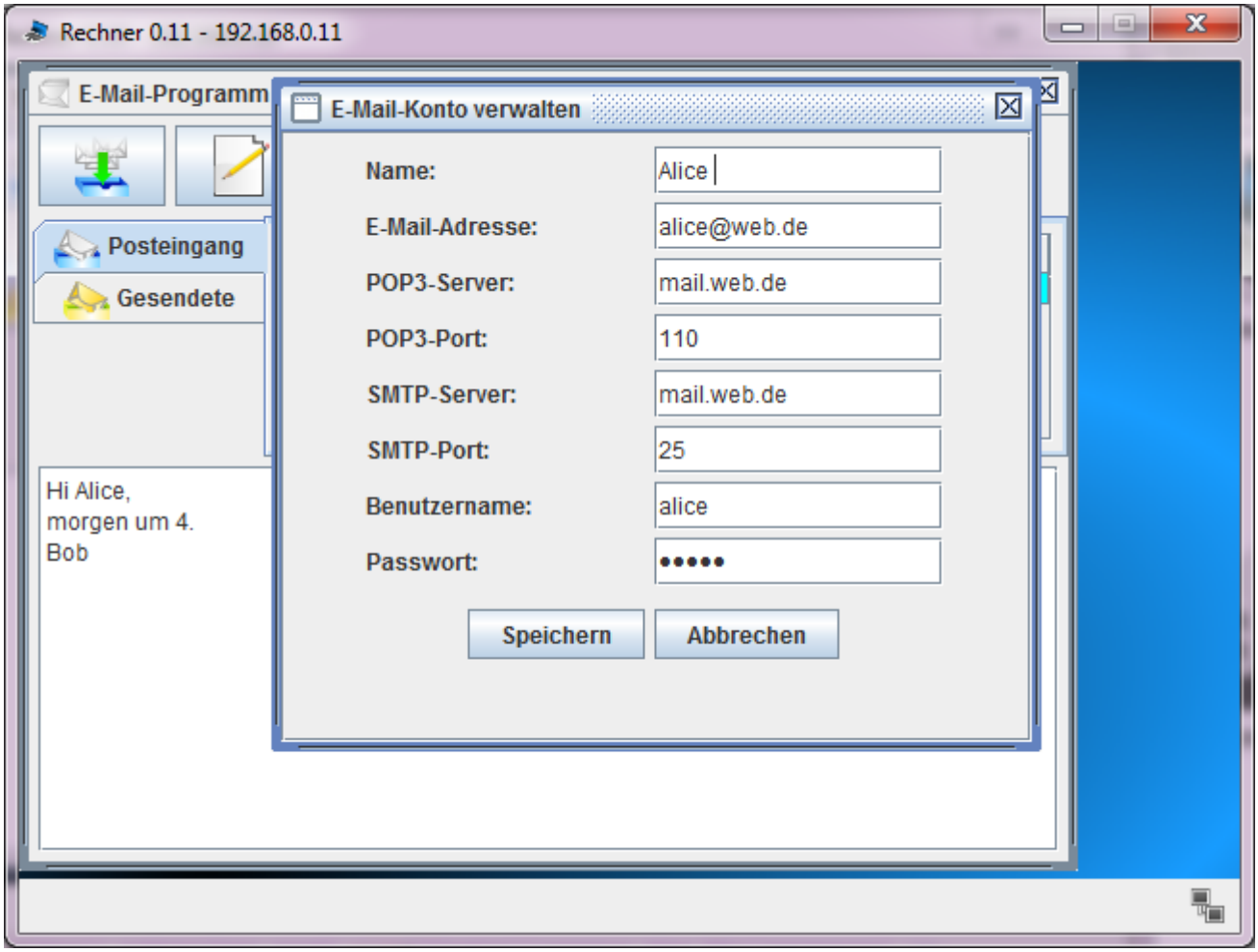
Wir richten hier auf dem Mail-Server mail.web.de ein Konto für Alice (Benutzername: alice; Passwort: Gu7+e) und auf dem Mail-Server mail.gmx.de ein Konto für Bob (Benutzername: bob; Passwort: Tb1-a) ein.

Anschließend werden die Mail-Server gestartet.

4) Installation und Konfiguration des E-Mail Clients

Auf den (Laptop-)Rechnern wird jetzt die E-Mail-Client-Software installiert.

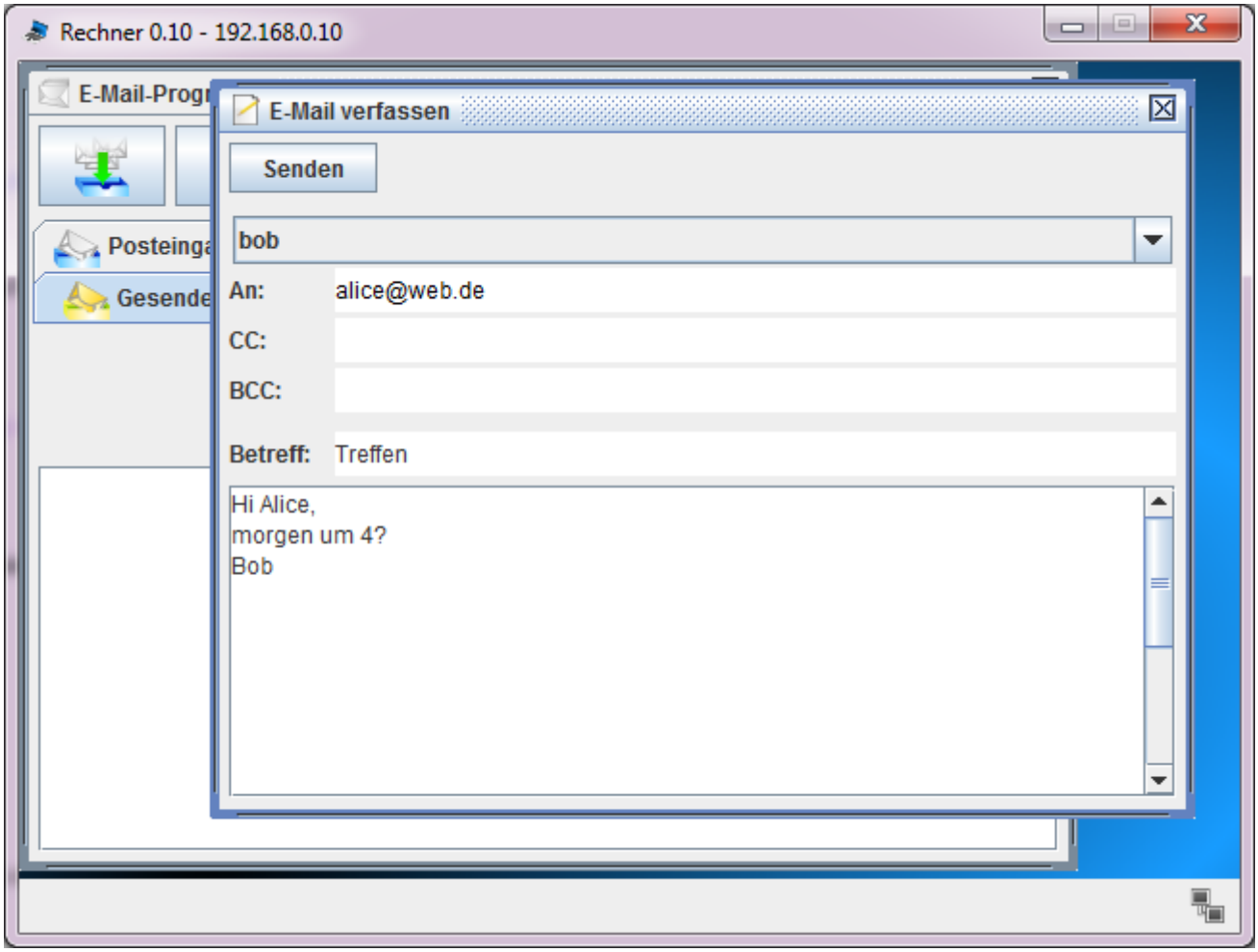
Wenn man das E-Mail-Programm startet, dann sollte man zunächst ein E-Mail-Konto einrichten.



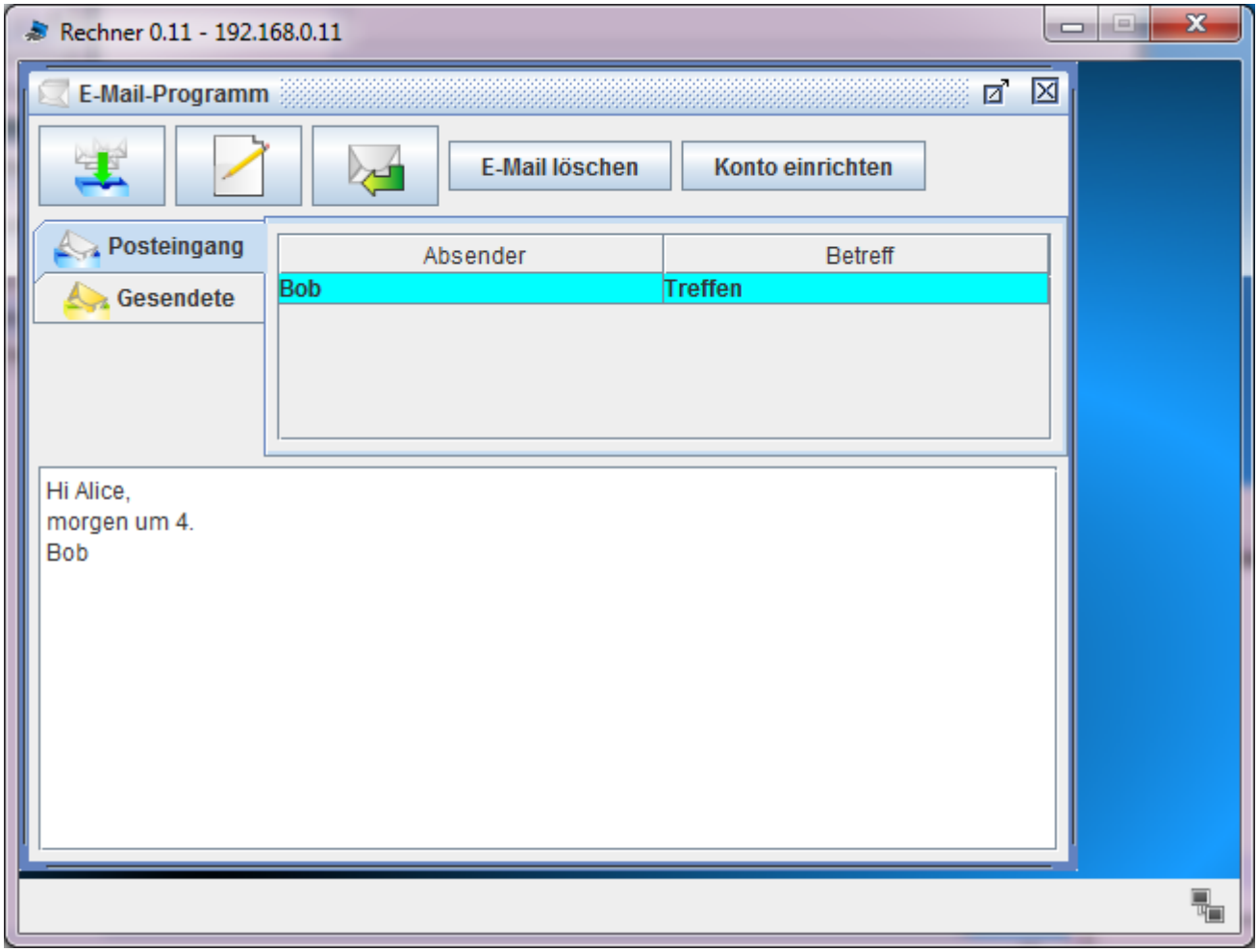
Entsprechend lässt sich ein Konto für Bob (z.B. mit dem E-Mail-Programm auf dem Rechner 192.168.0.10) einrichten.

5) E-Mail senden und abrufen

Mit den getroffenen Vorbereitungen kann jetzt beispielsweise Bob eine E-Mail an Alice schreiben und versenden.



Alice kann jetzt ihre E-Mails vom E-Mail-Server abrufen.



6) Analyse

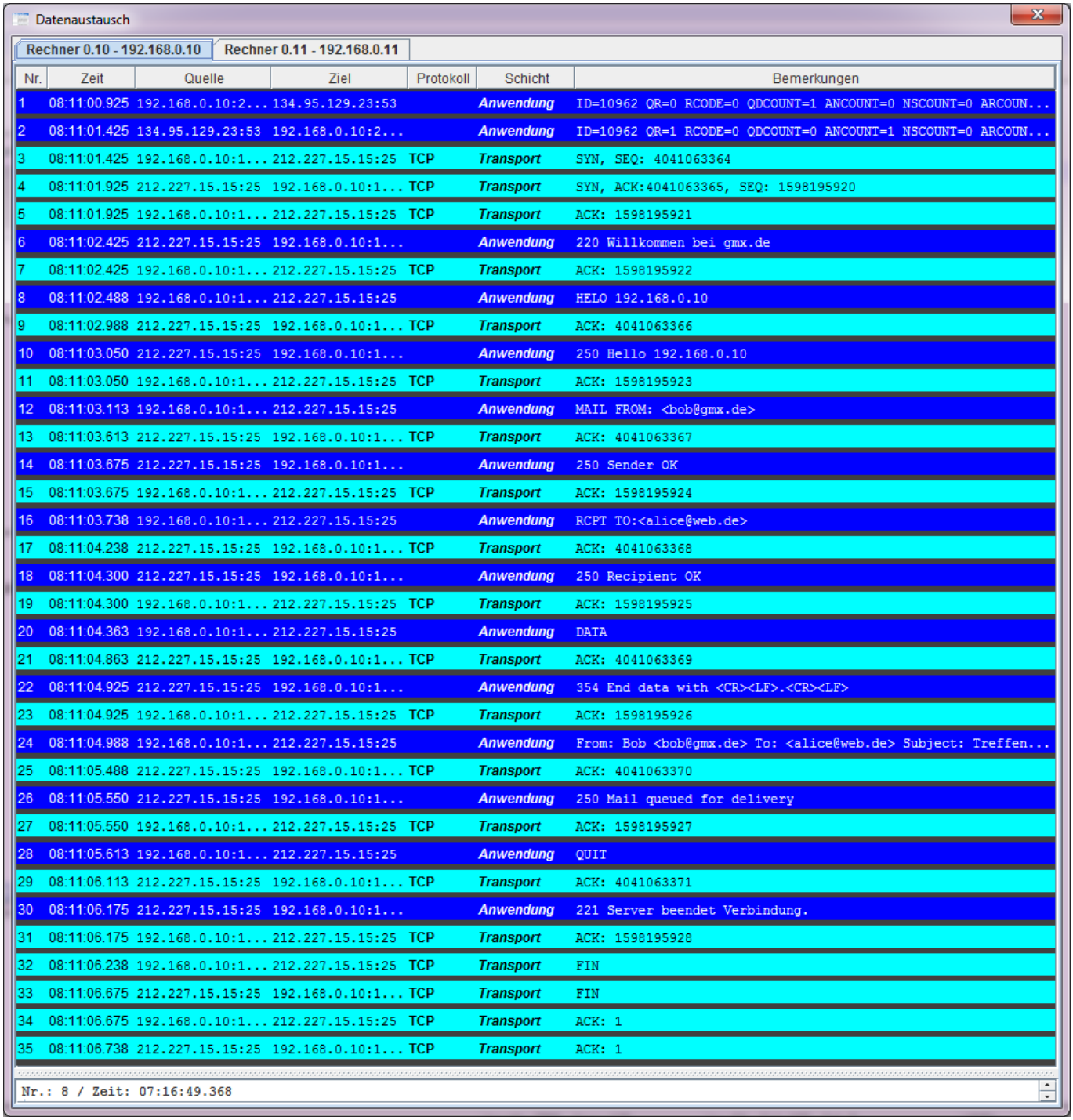
Starte noch einmal die Simulation und reduziere evtl. die Geschwindigkeit, damit du den Datenfluss genauer beobachten kannst. Wir betrachten den gesamten Kommunikationsvorgang von „Bob schreibt eine E-Mail an Alice.“ bis zu „Alice liest die E-Mail von Bob.“

Beobachtet und notiert in Partnerarbeit: Wann leuchtet welche Verbindung aus welchem Grund auf?

Natürlich könnt ihr hier mehrere „aufgeleuchtete“ Verbindungen zusammenfassen. Aber Vorsicht! Bei dem gesamten Prozess handelt es sich um etliche Schritte.

Der Datenaustausch beim Senden und Abrufen von E-Mails wird durch Protokolle geregelt. Einen ersten Einblick in diese Protokolle erhält man, indem man sich den Datenaustausch anhand einer Beispielkommunikationssituation genauer anschaut.

Bob sendet eine E-Mail an Alice. Hier der Datenaustausch zu dieser Kommunikationssituation:



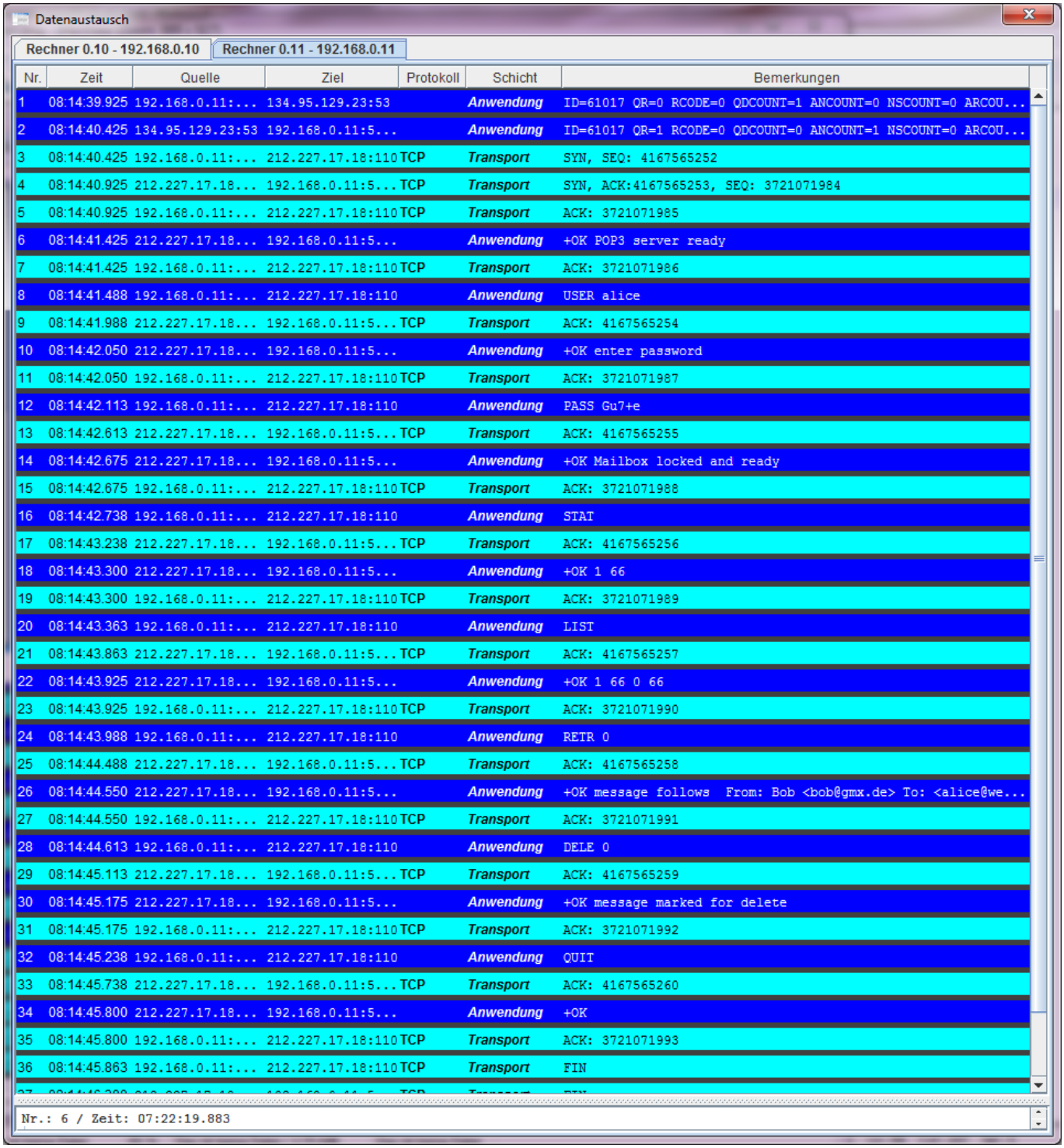
Datenaustausch

Rechner 0.10 - 192.168.0.10 | Rechner 0.11 - 192.168.0.11

Nr.	Zeit	Quelle	Ziel	Protokoll	Schicht	Bemerkungen
1	08:11:00.925	192.168.0.10:2...	134.95.129.23:53		Anwendung	ID=10962 QR=0 RCODE=0 QDCOUNT=1 ANCOUNT=0 NSCOUNT=0 ARCOUN...
2	08:11:01.425	134.95.129.23:53	192.168.0.10:2...		Anwendung	ID=10962 QR=1 RCODE=0 QDCOUNT=0 ANCOUNT=1 NSCOUNT=0 ARCOUN...
3	08:11:01.425	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	SYN, SEQ: 4041063364
4	08:11:01.925	212.227.15.15:25	192.168.0.10:1...	TCP	Transport	SYN, ACK:4041063365, SEQ: 1598195920
5	08:11:01.925	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	ACK: 1598195921
6	08:11:02.425	212.227.15.15:25	192.168.0.10:1...		Anwendung	220 Willkommen bei gmx.de
7	08:11:02.425	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	ACK: 1598195922
8	08:11:02.488	192.168.0.10:1...	212.227.15.15:25		Anwendung	HELO 192.168.0.10
9	08:11:02.988	212.227.15.15:25	192.168.0.10:1...	TCP	Transport	ACK: 4041063366
10	08:11:03.050	212.227.15.15:25	192.168.0.10:1...		Anwendung	250 Hello 192.168.0.10
11	08:11:03.050	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	ACK: 1598195923
12	08:11:03.113	192.168.0.10:1...	212.227.15.15:25		Anwendung	MAIL FROM: <bob@gmx.de>
13	08:11:03.613	212.227.15.15:25	192.168.0.10:1...	TCP	Transport	ACK: 4041063367
14	08:11:03.675	212.227.15.15:25	192.168.0.10:1...		Anwendung	250 Sender OK
15	08:11:03.675	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	ACK: 1598195924
16	08:11:03.738	192.168.0.10:1...	212.227.15.15:25		Anwendung	RCPT TO:<alice@web.de>
17	08:11:04.238	212.227.15.15:25	192.168.0.10:1...	TCP	Transport	ACK: 4041063368
18	08:11:04.300	212.227.15.15:25	192.168.0.10:1...		Anwendung	250 Recipient OK
19	08:11:04.300	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	ACK: 1598195925
20	08:11:04.363	192.168.0.10:1...	212.227.15.15:25		Anwendung	DATA
21	08:11:04.863	212.227.15.15:25	192.168.0.10:1...	TCP	Transport	ACK: 4041063369
22	08:11:04.925	212.227.15.15:25	192.168.0.10:1...		Anwendung	354 End data with <CR><LF>.<CR><LF>
23	08:11:04.925	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	ACK: 1598195926
24	08:11:04.988	192.168.0.10:1...	212.227.15.15:25		Anwendung	From: Bob <bob@gmx.de> To: <alice@web.de> Subject: Treffen...
25	08:11:05.488	212.227.15.15:25	192.168.0.10:1...	TCP	Transport	ACK: 4041063370
26	08:11:05.550	212.227.15.15:25	192.168.0.10:1...		Anwendung	250 Mail queued for delivery
27	08:11:05.550	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	ACK: 1598195927
28	08:11:05.613	192.168.0.10:1...	212.227.15.15:25		Anwendung	QUIT
29	08:11:06.113	212.227.15.15:25	192.168.0.10:1...	TCP	Transport	ACK: 4041063371
30	08:11:06.175	212.227.15.15:25	192.168.0.10:1...		Anwendung	221 Server beendet Verbindung.
31	08:11:06.175	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	ACK: 1598195928
32	08:11:06.238	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	FIN
33	08:11:06.675	212.227.15.15:25	192.168.0.10:1...	TCP	Transport	FIN
34	08:11:06.675	192.168.0.10:1...	212.227.15.15:25	TCP	Transport	ACK: 1
35	08:11:06.738	212.227.15.15:25	192.168.0.10:1...	TCP	Transport	ACK: 1

Nr.: 8 / Zeit: 07:16:49.368

Alice holt ihre E-Mails vom E-Mail-Server ab. Hier der Datenaustausch zu dieser Kommunikationssituation:



Datenaustausch

Rechner 0.10 - 192.168.0.10 | Rechner 0.11 - 192.168.0.11

Nr.	Zeit	Quelle	Ziel	Protokoll	Schicht	Bemerkungen
1	08:14:39.925	192.168.0.11:...	134.95.129.23:53		Anwendung	ID=61017 QR=0 RCODE=0 QDCOUNT=1 ANCOUNT=0 NSCOUNT=0 ARCOU...
2	08:14:40.425	134.95.129.23:53	192.168.0.11:5...		Anwendung	ID=61017 QR=1 RCODE=0 QDCOUNT=0 ANCOUNT=1 NSCOUNT=0 ARCOU...
3	08:14:40.425	192.168.0.11:...	212.227.17.18:110	TCP	Transport	SYN, SEQ: 4167565252
4	08:14:40.925	212.227.17.18...	192.168.0.11:5...	TCP	Transport	SYN, ACK:4167565253, SEQ: 3721071984
5	08:14:40.925	192.168.0.11:...	212.227.17.18:110	TCP	Transport	ACK: 3721071985
6	08:14:41.425	212.227.17.18...	192.168.0.11:5...		Anwendung	+OK POP3 server ready
7	08:14:41.425	192.168.0.11:...	212.227.17.18:110	TCP	Transport	ACK: 3721071986
8	08:14:41.488	192.168.0.11:...	212.227.17.18:110		Anwendung	USER alice
9	08:14:41.988	212.227.17.18...	192.168.0.11:5...	TCP	Transport	ACK: 4167565254
10	08:14:42.050	212.227.17.18...	192.168.0.11:5...		Anwendung	+OK enter password
11	08:14:42.050	192.168.0.11:...	212.227.17.18:110	TCP	Transport	ACK: 3721071987
12	08:14:42.113	192.168.0.11:...	212.227.17.18:110		Anwendung	PASS Gu7+e
13	08:14:42.613	212.227.17.18...	192.168.0.11:5...	TCP	Transport	ACK: 4167565255
14	08:14:42.675	212.227.17.18...	192.168.0.11:5...		Anwendung	+OK Mailbox locked and ready
15	08:14:42.675	192.168.0.11:...	212.227.17.18:110	TCP	Transport	ACK: 3721071988
16	08:14:42.738	192.168.0.11:...	212.227.17.18:110		Anwendung	STAT
17	08:14:43.238	212.227.17.18...	192.168.0.11:5...	TCP	Transport	ACK: 4167565256
18	08:14:43.300	212.227.17.18...	192.168.0.11:5...		Anwendung	+OK 1 66
19	08:14:43.300	192.168.0.11:...	212.227.17.18:110	TCP	Transport	ACK: 3721071989
20	08:14:43.363	192.168.0.11:...	212.227.17.18:110		Anwendung	LIST
21	08:14:43.863	212.227.17.18...	192.168.0.11:5...	TCP	Transport	ACK: 4167565257
22	08:14:43.925	212.227.17.18...	192.168.0.11:5...		Anwendung	+OK 1 66 0 66
23	08:14:43.925	192.168.0.11:...	212.227.17.18:110	TCP	Transport	ACK: 3721071990
24	08:14:43.988	192.168.0.11:...	212.227.17.18:110		Anwendung	RETR 0
25	08:14:44.488	212.227.17.18...	192.168.0.11:5...	TCP	Transport	ACK: 4167565258
26	08:14:44.550	212.227.17.18...	192.168.0.11:5...		Anwendung	+OK message follows From: Bob <bob@gmx.de> To: <alice@we...
27	08:14:44.550	192.168.0.11:...	212.227.17.18:110	TCP	Transport	ACK: 3721071991
28	08:14:44.613	192.168.0.11:...	212.227.17.18:110		Anwendung	DELE 0
29	08:14:45.113	212.227.17.18...	192.168.0.11:5...	TCP	Transport	ACK: 4167565259
30	08:14:45.175	212.227.17.18...	192.168.0.11:5...		Anwendung	+OK message marked for delete
31	08:14:45.175	192.168.0.11:...	212.227.17.18:110	TCP	Transport	ACK: 3721071992
32	08:14:45.238	192.168.0.11:...	212.227.17.18:110		Anwendung	QUIT
33	08:14:45.738	212.227.17.18...	192.168.0.11:5...	TCP	Transport	ACK: 4167565260
34	08:14:45.800	212.227.17.18...	192.168.0.11:5...		Anwendung	+OK
35	08:14:45.800	192.168.0.11:...	212.227.17.18:110	TCP	Transport	ACK: 3721071993
36	08:14:45.863	192.168.0.11:...	212.227.17.18:110	TCP	Transport	FIN

Nr.: 6 / Zeit: 07:22:19.883

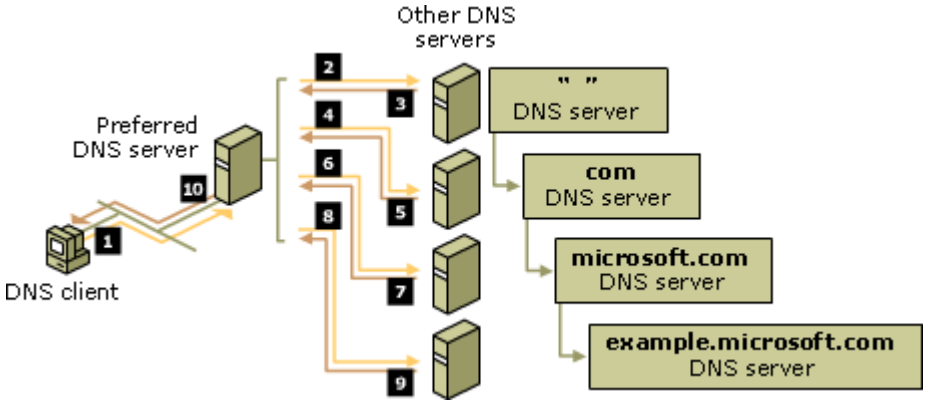
Beantworte kurz die folgenden Fragen mit Blick auf die letzten beiden Screenshots:

1. Am Anfang des ersten Screenshots sind drei Zeilen nacheinander hellblau - um was geht es in diesen drei Zeilen?
2. Welches ist der erste und welches der letzte Befehl auf der Ebene des SMTP-Protokolls (Anwendungsebene, dunkelblaue Zeilen), den Bob an den Server sendet?
3. Wie authentifiziert sich Bob bei seinem Mailserver?
4. Wo wird sichergestellt, dass der Absender „bob@gmx.de“ korrekt ist?
5. Den E-Mail-Versand nach SMTP-Protokoll kann man mit einem Brief in einem Briefumschlag vergleichen.
    1. a) In welcher Zeile wird die Adresse auf den „Briefumschlag“ „geschrieben“?
    2. b) In welcher Zeile wird die Adresse auf den „Brief“ „geschrieben“?
6. (Betrachte nun den zweiten Screenshot)

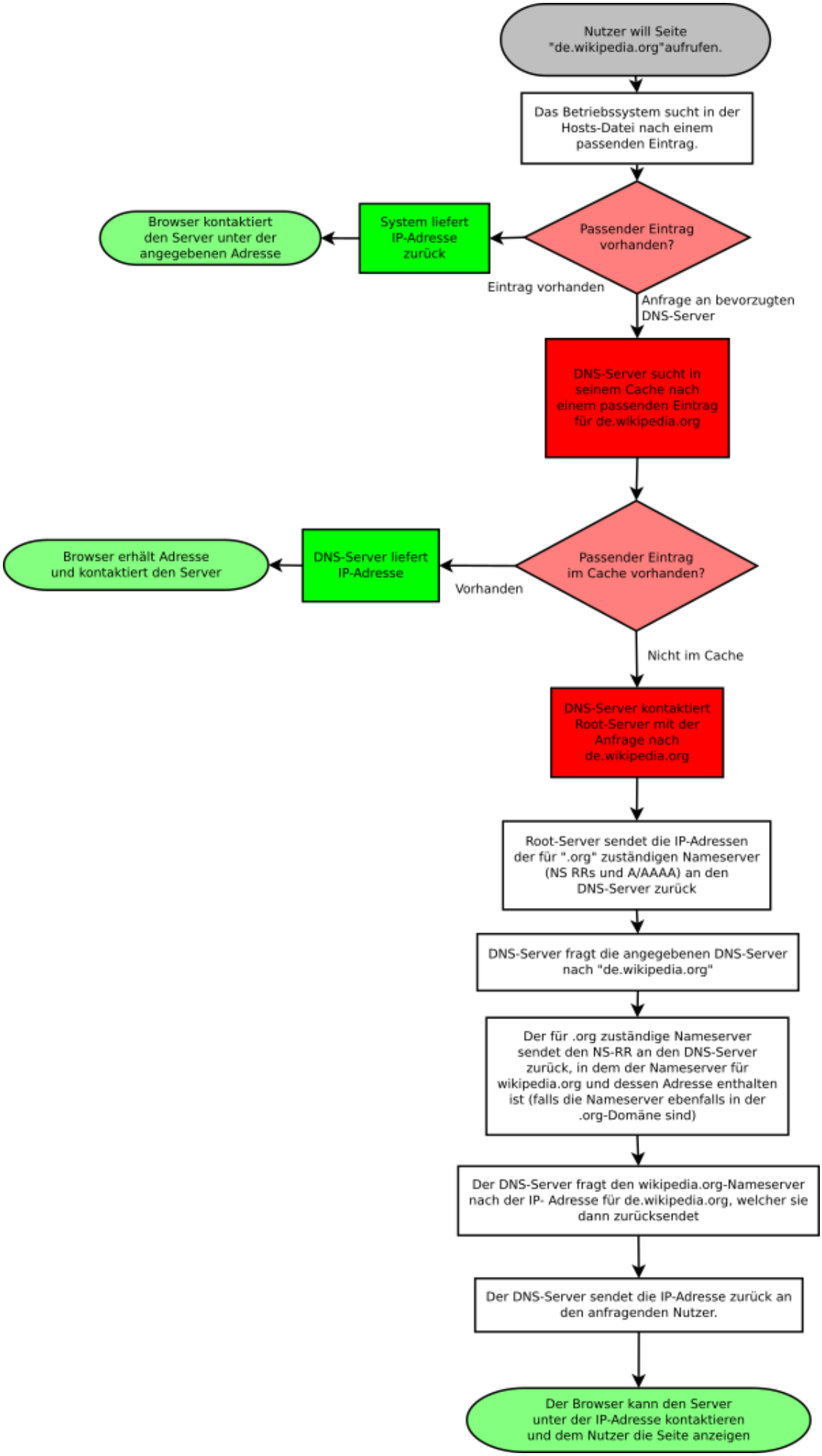
1. Welches ist der erste und welches der letzte Befehl auf der Ebene des POP3-Protokolls (Anwendungsebene, dunkelblaue Zeilen), den Alice an den Server sendet?
7. In welcher Zeile wird das Passwort zum Konto von Alice übermittelt?
8. Wie wird dieses Passwort gegenüber Eve verschlüsselt?
9. Was bewirkt der Befehl „RETR 0“ in Zeile 24?

Rekursive Namensauflösung - Rekursive DNS-Serverstruktur

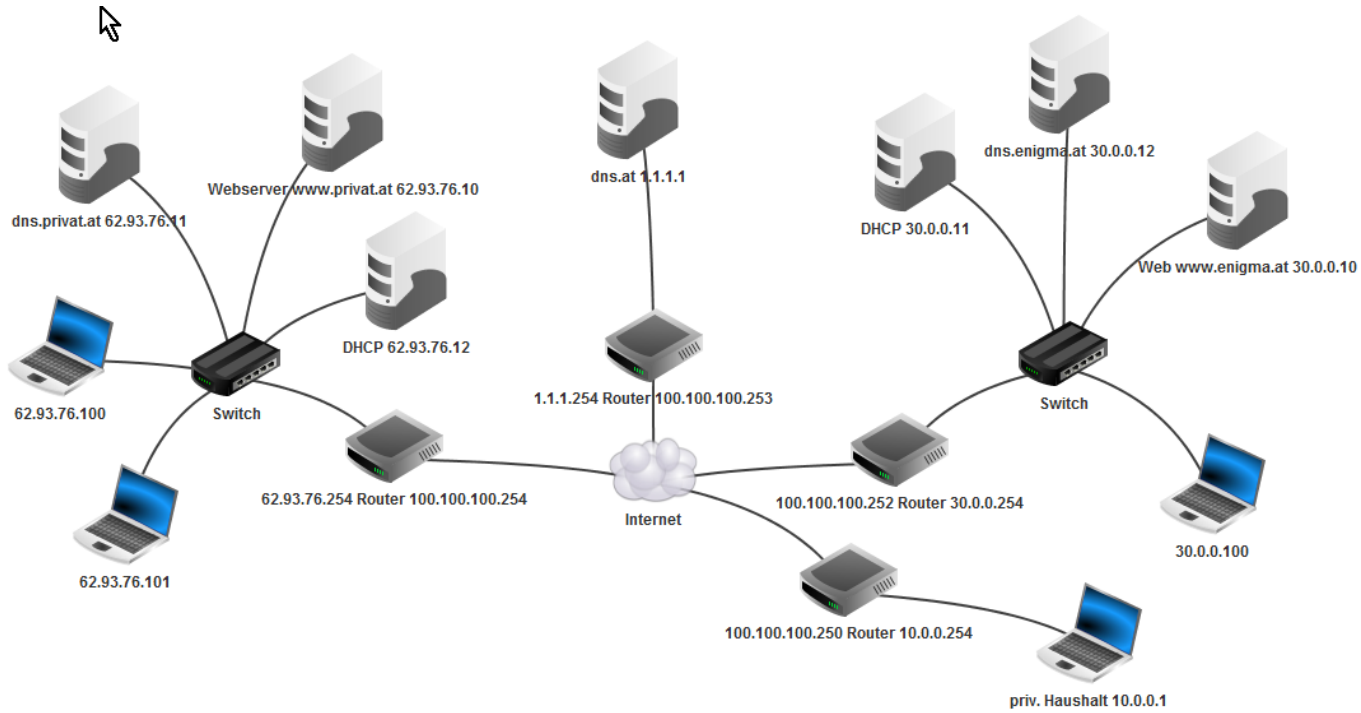
- Einzelne DNS-Server haben nicht jeden Domainnamen gespeichert. Informationen werden hierarchisch von übergeordneten DNS-Servern abgefragt. Die folgenden Grafiken sollen den Prozess veranschaulichen:



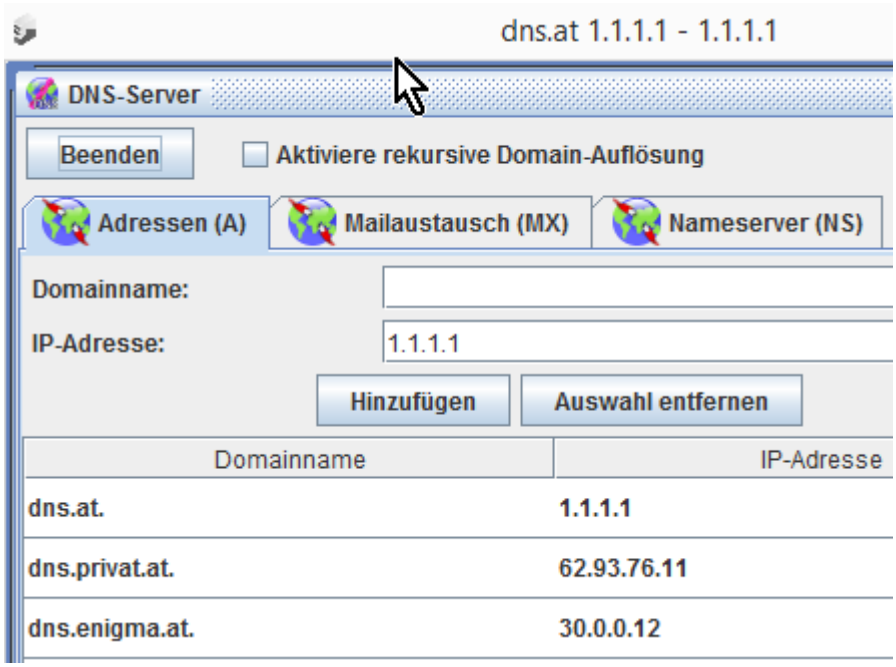
- Hier eine genauere Abfolge der Anfragen:

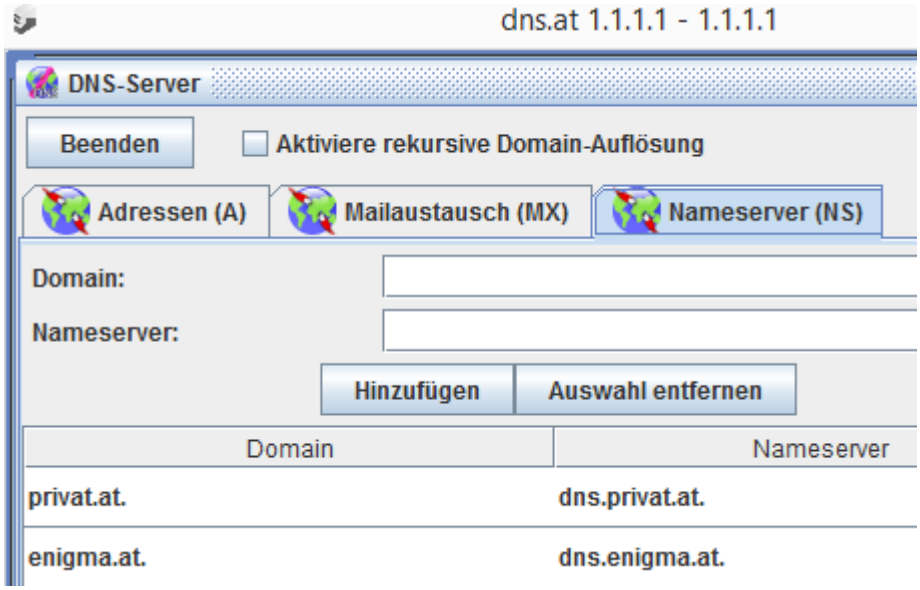

Nutzer will Seite "de.wikipedia.org"aufrufen.
Das Betriebssystem sucht in der Hosts-Datei nach einem passenden Eintrag.
Passender Eintrag vorhanden?
System liefert IP-Adresse zurück
Browser kontaktiert den Server unter der angegebenen Adresse
Eintrag vorhanden
Anfrage an bevorzugten DNS-Server
DNS-Server sucht in seinem Cache nach einem passenden Eintrag für de.wikipedia.org
Passender Eintrag im Cache vorhanden?
DNS-Server liefert IP-Adresse
Browser erhält Adresse und kontaktiert den Server
Vorhanden
Nicht im Cache
DNS-Server kontaktiert Root-Server mit der Anfrage nach de.wikipedia.org
Root-Server sendet die IP-Adressen der für ".org" zuständigen Nameserver (NS RRs und A/AAAA) an den DNS-Server zurück
DNS-Server fragt die angegebenen DNS-Server nach "de.wikipedia.org"
Der für .org zuständige Nameserver sendet den NS-RR an den DNS-Server zurück, in dem der Nameserver für wikipedia.org und dessen Adresse enthalten ist (falls die Nameserver ebenfalls in der .org-Domäne sind)
Der DNS-Server fragt den wikipedia.org-Nameserver nach der IP- Adresse für de.wikipedia.org, welcher sie dann zurücksendet
Der DNS-Server sendet die IP-Adresse zurück an den anfragenden Nutzer.
Der Browser kann den Server unter der IP-Adresse kontaktieren und dem Nutzer die Seite anzeigen

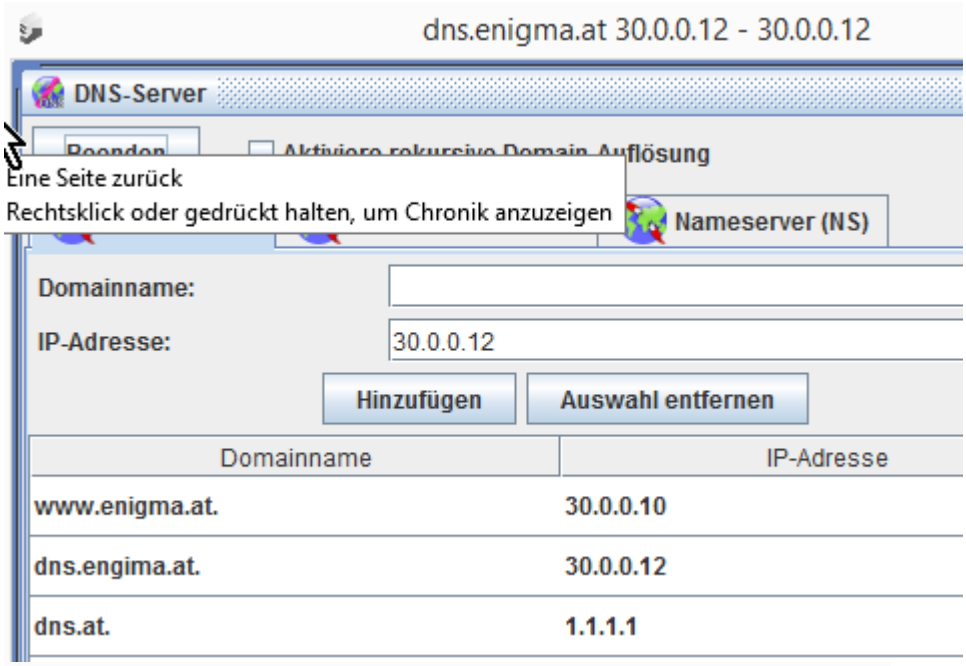
Mittels Filius kann eine hierarchische DNS-Serverstruktur aufgebaut werden:

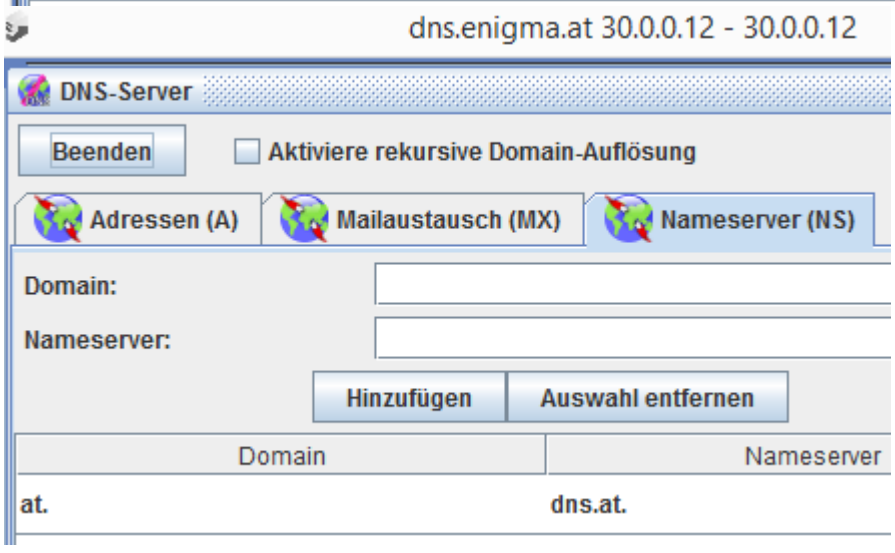


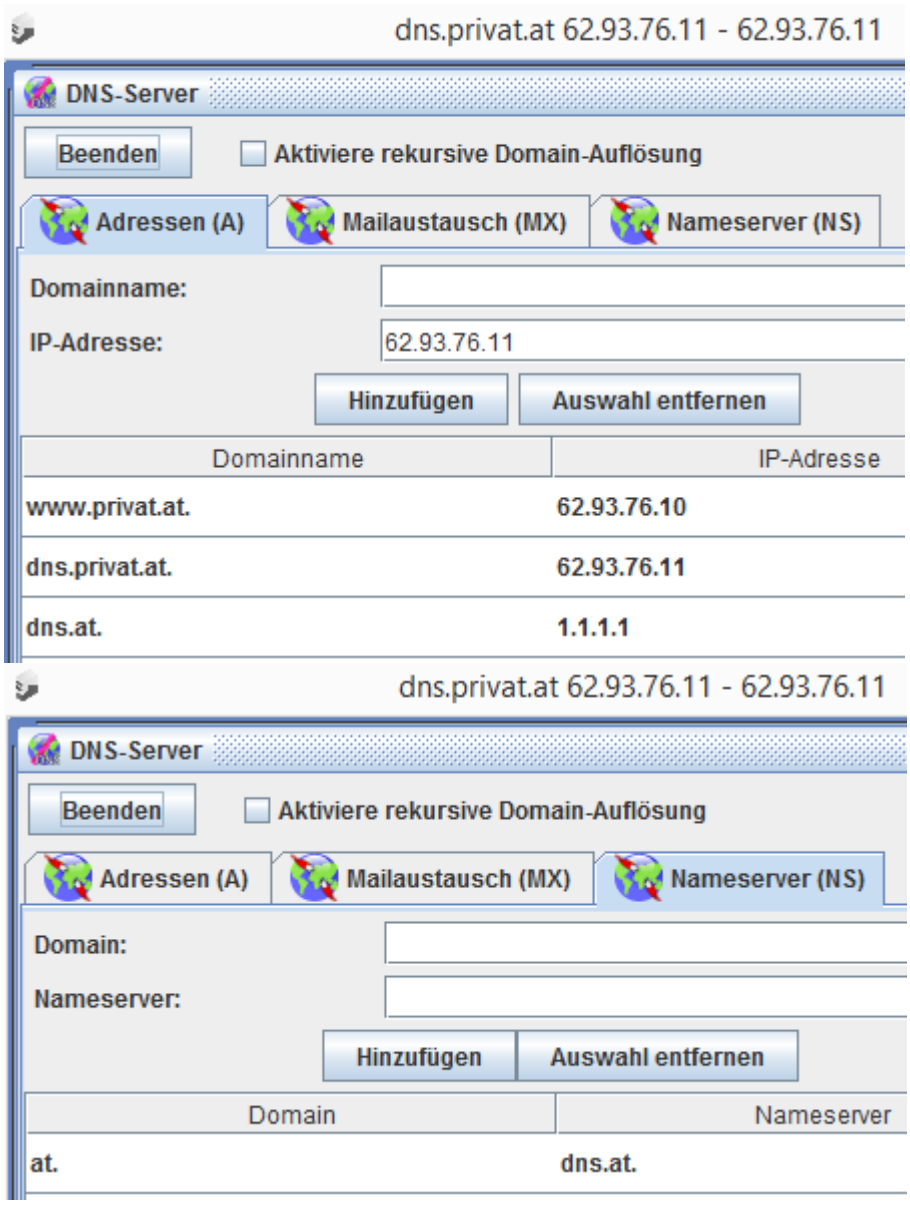
Hier die Screenshots über die entsprechenden Eintragungen in den Nameservern.




dns.at 1.1.1.1 - 1.1.1.1
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domain:
Nameserver:
Hinzufügen
Auswahl entfernen
Domain
Nameserver
privat.at.
dns.privat.at.
enigma.at.
dns.enigma.at.


dns.enigma.at 30.0.0.12 - 30.0.0.12
DNS-Server
Eine Seite zurück
Rechtsklick oder gedrückt halten, um Chronik anzuzeigen
Nameserver (NS)
Domainname:
IP-Adresse:
30.0.0.12
Hinzufügen
Auswahl entfernen
Domainname
IP-Adresse
www.enigma.at.
30.0.0.10
dns.engima.at.
30.0.0.12
dns.at.
1.1.1.1


dns.enigma.at 30.0.0.12 - 30.0.0.12
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domain:
Nameserver:
Hinzufügen
Auswahl entfernen
Domain
Nameserver
at.
dns.at.



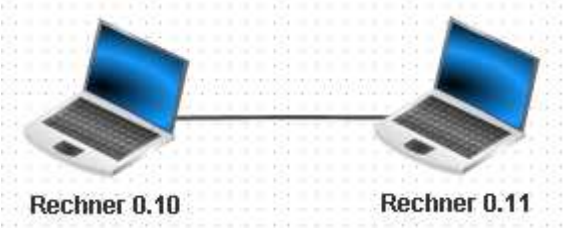
Router mit Firewall

- arbeitsblatt_firewall.pdf

Filius Übungen

Basisaufgaben

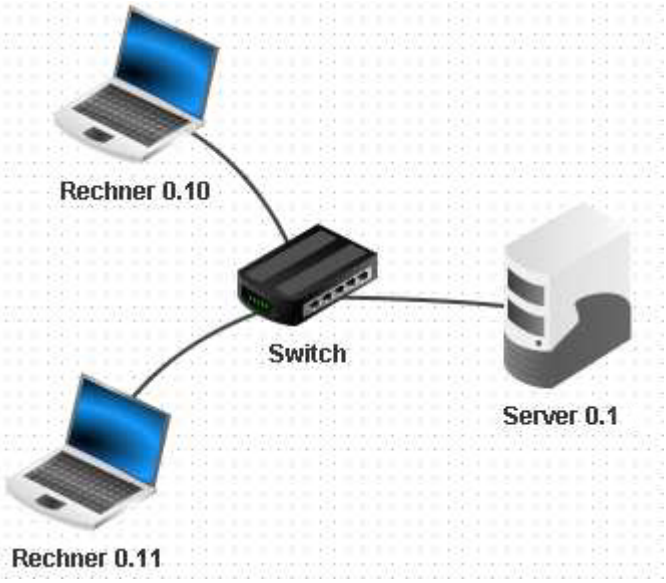
Aufgabe 1



1. Erstellen Sie ein Netzwerk mit zwei vernetzten Computern, welche beide eine Client-Funktion haben (Symbol: Notebook). Die Computer sollen die abgebildeten Namen sowie die IPs 192.168.0.10/24 und 192.168.0.11/24 besitzen. (Durch die richtige Subnetzmaske 255.255.255.0 stellen Sie sicher, dass beide Computer im selben Netzwerk liegen.)
2. Installieren Sie auf dem Rechner 0.10 eine „Befehlszeile“ (Terminal). Starten Sie das Terminal und testen Sie die Verbindung zum Rechner 0.11 mit dem Befehl `ping 192.168.0.11`. Beobachten Sie die Netzwerkaktivität, indem Sie sich den Datenaustausch von Rechner 0.10 anzeigen lassen.
3. Testen Sie auch andere Befehle auf dem Terminal, wie z. B. die Befehle `ipconfig`, oder `host localhost` oder `dir`. Der Sinn des host-Befehls wird zu einem späteren Zeitpunkt im Zusammenhang mit einem DNS-Server evtl. deutlicher.

Aufgabe 2

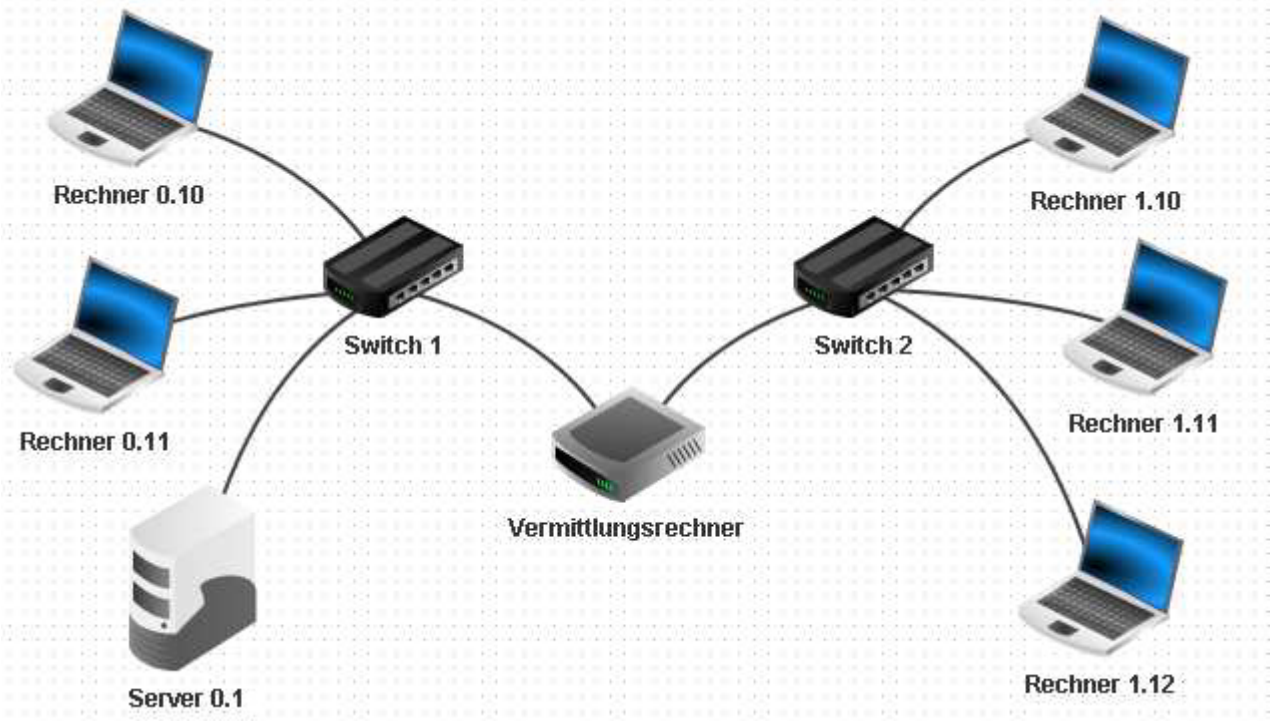
Erweitern Sie nun das Netzwerk um einen dritten Computer, einen Server, mit dem abgebildeten Namen und der IP 192.168.0.1/24, welcher im gleichen Netzwerk liegen soll. Achten Sie darauf, dass Sie von nun an für Computer mit der Funktion eines Servers, das Rechner-Symbol (Standrechner) wählen. Verbinden Sie alle Computer mit einem Switch wie abgebildet.



Installieren Sie auf dem Server 0.1 einen Echo-Server und starten diesen auf dem voreingestellten Port 55555. Installieren Sie auf einem Client einen Echo-Client und verbinden Sie diesen mit dem Server. Senden Sie vom Client einige Textnachrichten und beobachten Sie den Effekt. Schauen Sie sich auch die Netzwerkaktivität im Datenaustausch-Fenster des Clients an.

Aufgabe 3

- Erstellen Sie zwei Netzwerke mit je drei Rechnern wie abgebildet.
- Die Computer des ersten Netzwerks sollen die abgebildeten Namen besitezen sowie die IPs 192.168.0.10/24 und 192.168.0.11/24 zugewiesen bekommen.
- Der Server 192.168.0.1/24 soll die IP-Adressen in diesem Netzwerk vergeben. Rechner 0.10 soll die IP-Adresse statisch vom Server bekommen, Rechner 0.11 die IP-Adresse aus einem Pool.
- Die Rechner des zweiten Netzwerks sollen sich einem logisch anderen Netzwerk befinden. Wählen Sie dafür die IPs 10.1.1.10/16 bis 10.1.1.12/16. (fix vergeben). Verbinden Sie anschließend die beiden Netzwerke mit einem Vermittlungsrechner (Router), welcher die Netzwerkkarten mit den IPs 192.168.0.254/24 und 10.1.1.254/16 besitzt.



- Prüfen Sie anschließend im Terminal mit einem ping-Befehl die Verbindung der Rechner vom Netzwerk 192.168.0.0/24 zu den Rechnern des Netzes 10.1.0.0/16.
    - Beschreiben Sie, welches Problem auftreten kann und wie man sicherstellen kann, dass die Rechner aus einem fremden Netzwerk erreicht werden können.
- Testen Sie die Netzwerkverbindung auch mit dem Echo-Client und Echo-Server. Installieren Sie dazu auf einem Rechner aus dem Netz 192.168.0.0/24 einen Echo-Server und auf einem Rechner im Netz 10.1.0.0/16 einen Echo-Client.

Aufgabe 4

- Installieren Sie auf dem Server 192.168.0.1 einen Webserver und einen Texteditor. Starten Sie den Texteditor und öffnen Sie hiermit die Datei index.html aus dem virtuellen Verzeichnis root/webserver. Passen Sie den html-Code so an, dass eine Seite mit Ihren Informationen angezeigt wird. Erstellen Sie auch eine neue Seite kontakt.html, welche von der Startseite verlinkt werden soll.
- Starten Sie die Anwendung „Webserver“ mit einem Doppelklick. Starten Sie dann den virtuellen Webserver über den Button Starten.
- Installieren Sie dann auf dem Rechner 10.1.1.10 einen Webbrowser. Starten Sie den Browser und bauen Sie eine Verbindung zum Webserver auf, indem Sie die URL http://192.168.0.1 in die Adressleiste des Webbrowsers eingeben.

DNS-Server

Aufgabe 5

- Erstellen Sie einen neuen Server mit der IP 1.1.1.1/24 und dem Gateway 1.1.1.254. Ändern Sie die Anzahl der Schnittstellen am Vermittlungsrechner auf drei ab. Ergänzen Sie die auf der

neuen Registerkarte zur dritten Netzwerkkarte die Einstellungen: IP-Adresse 1.1.1.254/24 (Netzmaske 255.255.255.0). Verbinden Sie anschließend den neuen Server mit dem Vermittlungsrechner.
- Tragen Sie bei jedem Rechner in den Einstellungen die DNS-Server-Adresse 1.1.1.1 ein. Dies entspricht dem gerade erstellten DNS-Server.
- Installieren Sie auf dem Server 1.1.1.1 die Anwendung „DNS-Server“. Tragen Sie in die Eingabefelder den Domainnamen www.myweb.at und die zugehörige IP-Adresse 1.1.1.1 ein. Starten Sie abschließend den DNS-Server mit dem Button „Starten“. Testen Sie die Verbindung von Ihrem Webbrowser nun mit der URL http://www.myweb.at

Weitere Übungsaufgaben

Übung 01 - Schulnetzwerk

- aufgabe_schulnetzwerk.pdf

Übung 02 - Firma

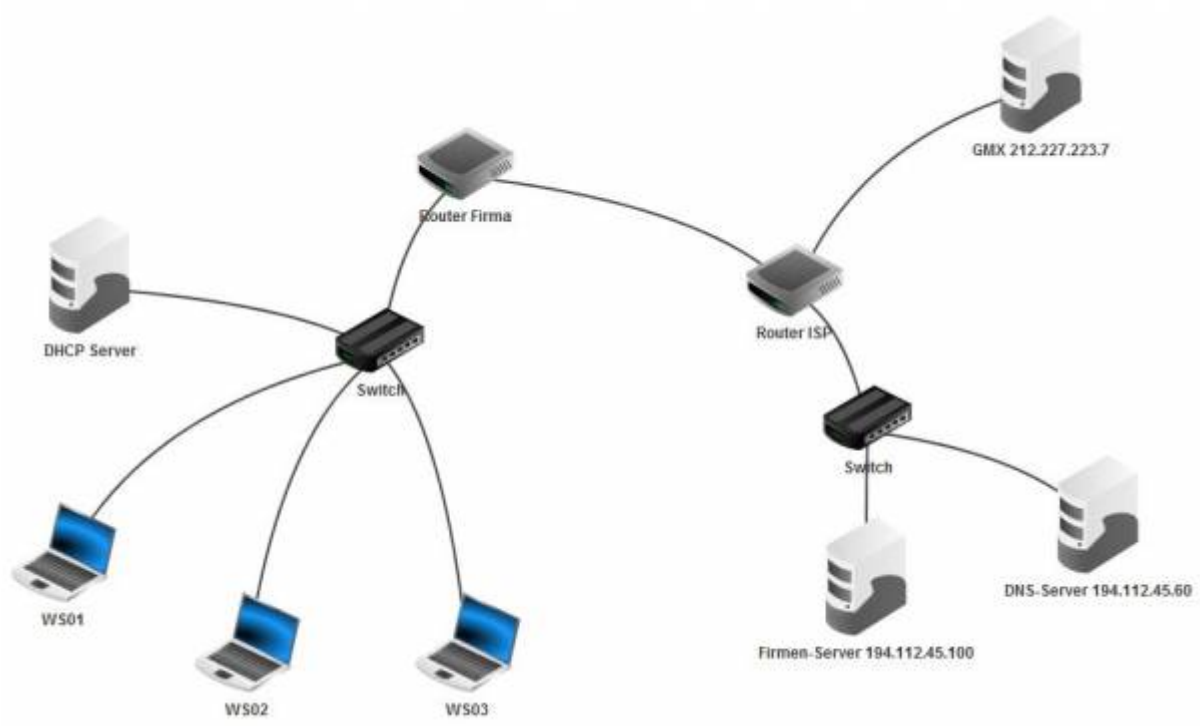
Du bekommst den Auftrag, in einer kleinen Firma mit drei PCs und einem Server das Netzwerk neu einzurichten. Ebenso soll für Web- und Maildienste ein eigener Server beim Provider angemietet und konfiguriert werden.

Starte das Programm Filius und öffne die Datei netzwerk-vorlage.fls.

Die Geräte des Firmennetzwerks und der Firmen-Server müssen vollständig konfiguriert werden, ebenso müssen am DNS-Server zusätzliche Einstellungen getätigt werden.

Alle bereits vorkonfigurierten Einstellungen sind funktionstüchtig und sollten nicht mehr verändert werden!

„Das Internet“ wird repräsentiert durch den Server von GMX, der per „Waterwolf“-Browser über http://www.gmx.at erreichbar ist. Auch sind zwei Emailadressen eingerichtet: alice@gmx.at und bob@gmx.at. POP3-Server: pop3.gmx.at, SMTP-Server: smtp.gmx.at, Passwörter: Benutzername in Großbuchstaben (ALICE bzw. BOB). Der DNS-Server und der beim Provider zu konfigurierende Firmen-Server stellen das Netzwerk des Internet Service Providers (ISP) dar.



Folgende Unterlagen zur Konfiguration bekommst du ausgehändigt:

- Firmennetzwerk:
  - Externe IP: 194.112.174.1
  - IP Router: 10.0.0.254
  - IP DHCP-Server: 10.0.0.100
  - Konfiguration der PCs via DHCP: 10.0.0.1 - 10.0.0.99
- Internetzugang für Firmen-Router
  - Externe IP: 194.112.174.1
  - Gateway: 194.112.174.254
  - DNS-Server: 194.112.45.60
- Firmenserver:
  - IP: 194.112.45.100
  - Gateway: 194.112.45.254
  - Domain (A): www.firma.at
  - Domain (MX): firma.at
  - Emailadresse: office@firma.at, Passwort: OFFICE

---

1. Richte den DHCP-Server und die Workstations ein.
2. Konfiguriere den Firmen-Router und stelle eine Verbindung zum ISP her. Prüfe die Verbindung mit dem Befehl ping zum ISP-Router und zum GMX-Server.
3. Bearbeite die Netzwerkeinstellungen des Firmen-Servers und prüfe die Erreichbarkeit vom DNS-Server und vom Firmennetzwerk aus.
4. Installiere und starte Web- und Mailserver auf dem Firmen-Server. Die Homepage soll umgestelltet werden mit „Herzlich willkommen auf unserer Homepage!“ (o.ä.)
5. Erweitere den DNS-Server um die Einträge für www.firma.at und firma.at
6. Installiere auf der Workstation WS01 Browser und Email-Client (mit der Firmenadresse office@firma.at)
7. Auf der Workstation WS02 und WS03 installiere ebenfalls Browser und Email-Client (WS02: alice@gmx.at, WS03: bob@gmx.at)

8. Sende eine Email von office@firma.at an alice@gmx.at und bob@gmx.at
9. Beantworte beide Mails.

Übung 03 - DNS-Serverstruktur

- dns-serverstruktur2.pdf

Übung 04 - Routing

- aufgabe-routingtabelle.pdf

IP-Routing

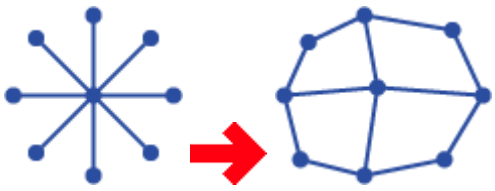
Das **Internet Protocol (IP)** ist ein **routingfähiges Protokoll** und sorgt dafür, dass **Datenpakete über Netzgrenzen** hinweg einen Weg zu anderen Hosts finden. Es kann die Daten über jede Art von physikalischer Verbindung oder Übertragungssystem vermitteln. Der hohen Flexibilität steht ein hohes Maß an Komplexität bei der Wegfindung vom Sender zum Empfänger gegenüber. Der **Vorgang der Wegfindung wird Routing** genannt

Wozu Routing?

Das grundlegende Verbindungselement in einem Ethernet-Netzwerk ist der Hub oder Switch. Daran sind alle Netzwerk-Teilnehmer angeschlossen. Wenn ein Host Datenpakete verschickt, dann werden die Pakete im Hub an alle Stationen verschickt und von diesen angenommen. Jedoch verarbeitet nur der adressierte Host die Pakete weiter. Das bedeutet aber auch, dass sich alle Hosts die Gesamtbandbreite dieses Hubs teilen müssen.

Um die Nachteile von Ethernet in Verbindung mit CSMA/CD auszuschließen, wählt man als Kopplungselement einen Switch und nutzt Fast Ethernet (kein CSMA/CD mehr). Der Switch merkt sich die Hardware-Adressen (MAC-Adressen) der Stationen und leitet die Ethernet-Pakete nur an den Port, hinter dem sich die Station befindet. Ist einem Switch die Hardware-Adresse nicht bekannt, leitet er das Datenpaket an alle seine Ports weiter (Broadcast) und funktioniert in diesem Augenblick wie ein Hub. Neben der begrenzten Speichergröße des Switches machen sich viele unbekannte Hardware-Adressen negativ auf die Performance eines Netzwerks bemerkbar.

Daher eignet sich zum **Verbinden großer Netzwerke** weder ein Hub noch ein Switch. Aus diesem Grund wird ein Netzwerk **durch Router** und IP-Adressen in **logische Segmente bzw. Subnetze** unterteilt. Die Adressierung durch das Internet Protocol ist so konzipiert, dass der Netzwerkverkehr innerhalb der Subnetze bleibt und erst dann das Netzwerk verlässt, wenn das Ziel in einem anderen Netzwerk liegt.



Insbesondere folgende **Probleme** in einem Ethernet-Netzwerk machen **IP-Routing notwendig**:

- **Vermeidung von Kollisionen und Broadcasts durch Begrenzung der Kollisions- und Broadcastdomäne**
- **Routing über unterschiedliche Netzarchitekturen und Übertragungssysteme**
- **Paket-Filter und Firewall**
- **Routing über Backup-Verbindungen bei Netzausfall**

Router

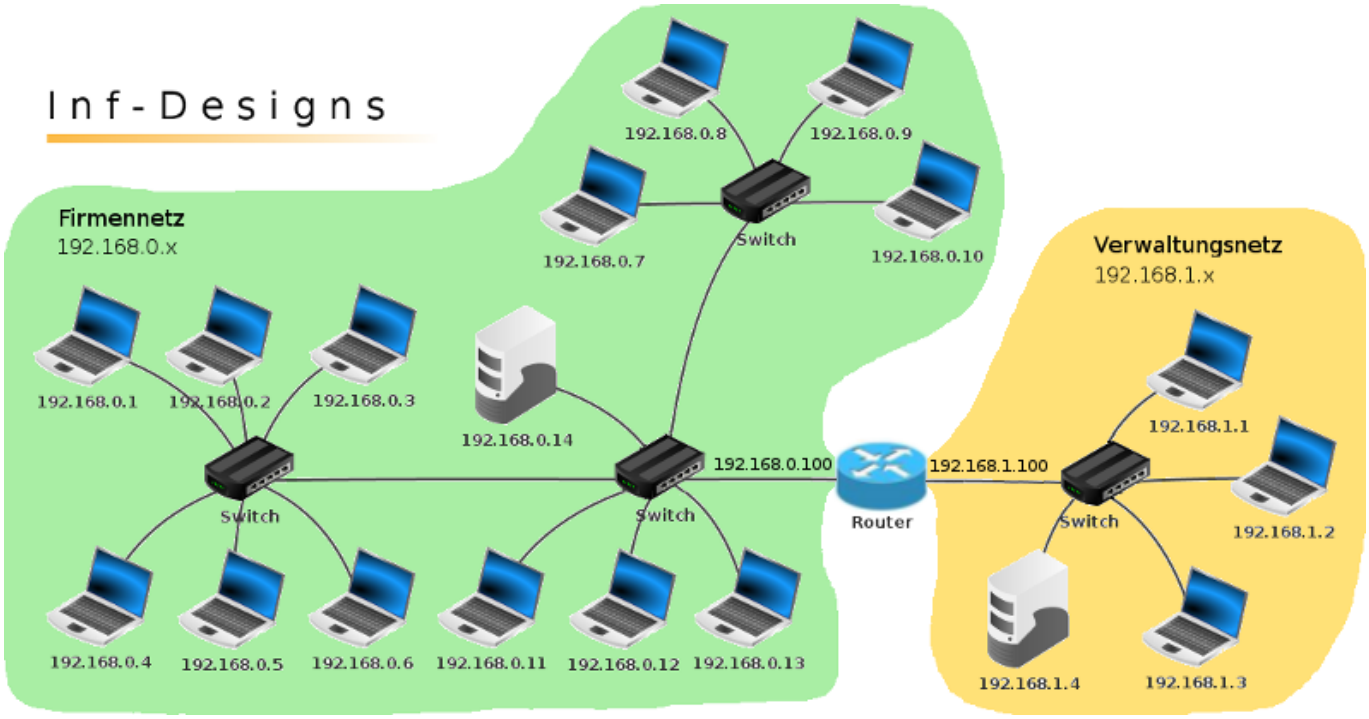
Große Rechnernetze bestehen häufig aus mehreren Teilnetzen. Liegen zwei Rechner, zwischen denen

Nachrichten ausgetauscht werden sollen, in benachbarten Teilnetzen, so benötigt man ein spezielles Gerät - einen Router - der die beiden Teilnetze miteinander verbindet.

Ein Router ist ein Rechner mit mindestens zwei Netzwerk-Schnittstellen.

Wichtige Aufgaben eines Routers sind ..

- die Weitervermittlung von Datenpaketen zwischen (lokalen) Netzwerken.
- ggf. die Kontrolle und Beschränkung des Datenverkehrs zwischen Netzwerken mit Hilfe einer Firewall.



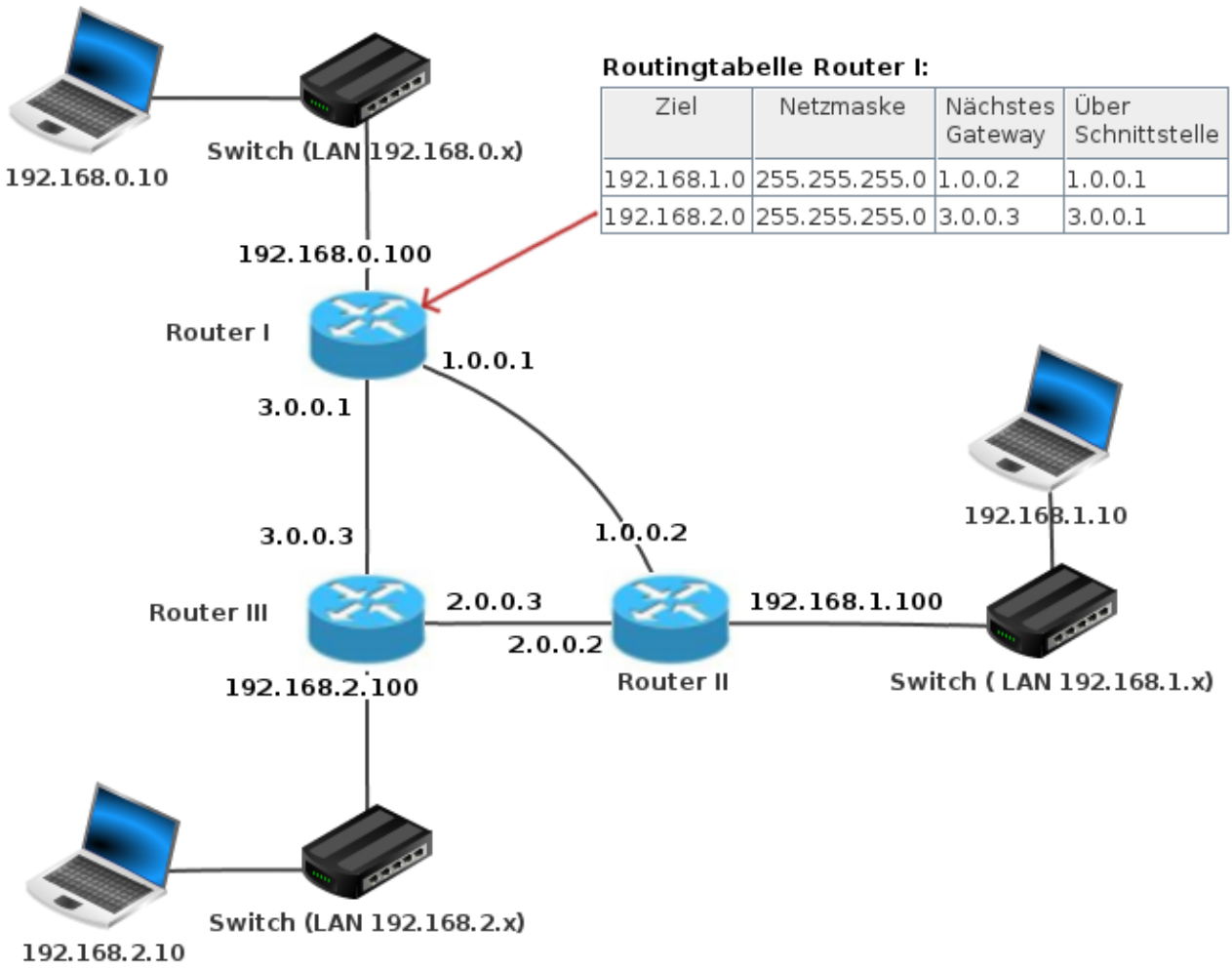
Wie man am Beispiel des Rechnernetzes unseres Startup-Unternehmens erkennen kann, besitzt ein Router Schnittstellen mit IP-Adressen hin zu allen angeschlossenen Teilnetzen (hier: Firmennetz und Verwaltungsnetz). Möchte ein Rechner eine Nachricht an einen Rechner in einem benachbarten Teilnetz schicken, so leitet er sie zunächst an den Router weiter, dessen IP-Adresse bei ihm als Gateway eingetragen ist. Für die Weitervermittlung der Nachricht im benachbarten Teilnetz ist dann der Router zuständig.

Viele Router können den Datenverkehr zwischen den Teilnetzen mit Hilfe einer Firewall beschränken, d.h. nur bestimmte Datenpakete weiterleiten und andere blockieren.

Routingtabellen

Häufig verläuft die Kommunikation zwischen zwei Rechnern über mehrere Router und Teilnetze hinweg. Hier stehen die einzelnen Router oft vor dem Problem, dass die Ziel-IP-Adresse zu gar keinem der angeschlossenen Teilnetze gehört. Woher soll ein Router aber wissen, in welches Teilnetz er eine Nachricht weiterleiten soll?

Die Lösung des Problems: Jeder Router verfügt über eine Routingtabelle, in der genau festgelegt ist, wie eine Nachricht weitergeleitet wird.



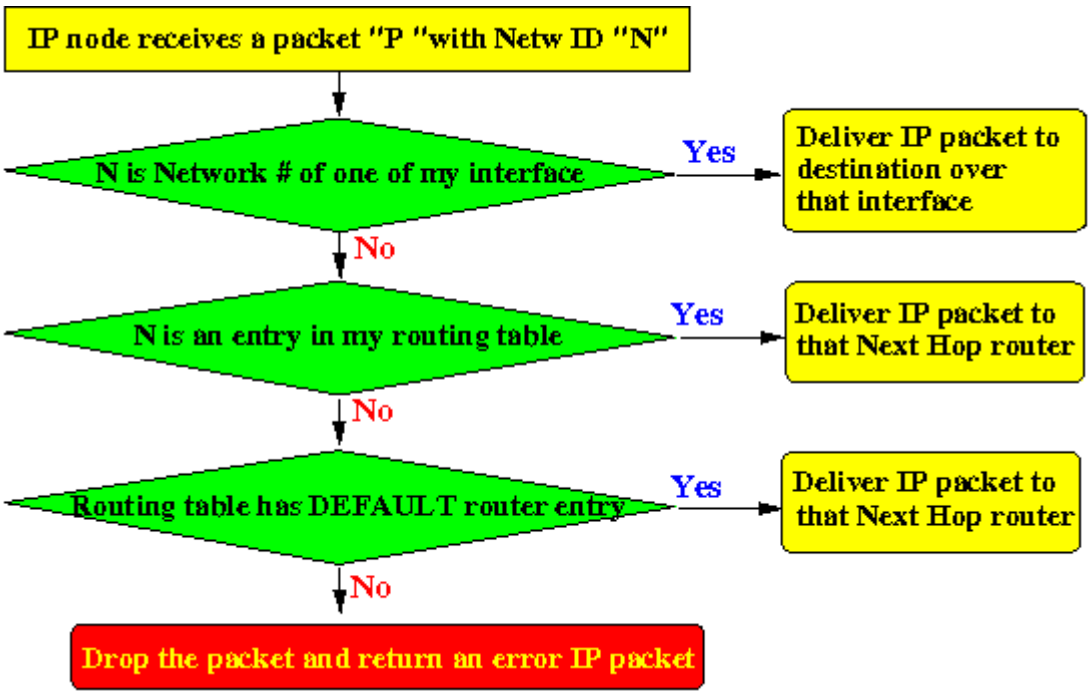
Zu jedem eingetragenen Ziel werden die Schnittstelle, über die eine Nachricht den Router verlassen soll und die im nachfolgenden Teilnetz sichtbare IP-Adresse des nächsten Routers („nächstes Gateway“) angegeben. Als Ziele werden in einer Routingtabelle in der Regel nicht einzelne Rechner, sondern ganze Teilnetze in Form einer IP-Adresse des Netzes und der Netzmaske angegeben. Das spart, wie man sich denken kann, sehr viele Einträge.

Oft gibt es mehrere mögliche Routen, die eine Nachricht von einem zum anderen Rechner zurücklegen kann. Im besten Fall ist in den Routingtabellen die jeweils kürzeste Route zum Zielrechner, also die Route mit den wenigsten verbleibenden Zwischenstationen („Hops“) eingetragen.

Um sicherzustellen, dass in den Routingtabellen zu jedem Zeitpunkt aktuelle und optimale Routen eingetragen sind, werden sie von den Routern dynamisch über Routingprotokolle aktualisiert.

IP-Routing-Algorithmus

Der IP-Routing-Algorithmus gilt nicht nur für IP-Router, sondern für alle Host, die IP-Datenpakete empfangen können. Die empfangenen Datenpakete durchlaufen diesen Algorithmus bis das Datenpaket zugeordnet oder weitergeleitet werden kann.



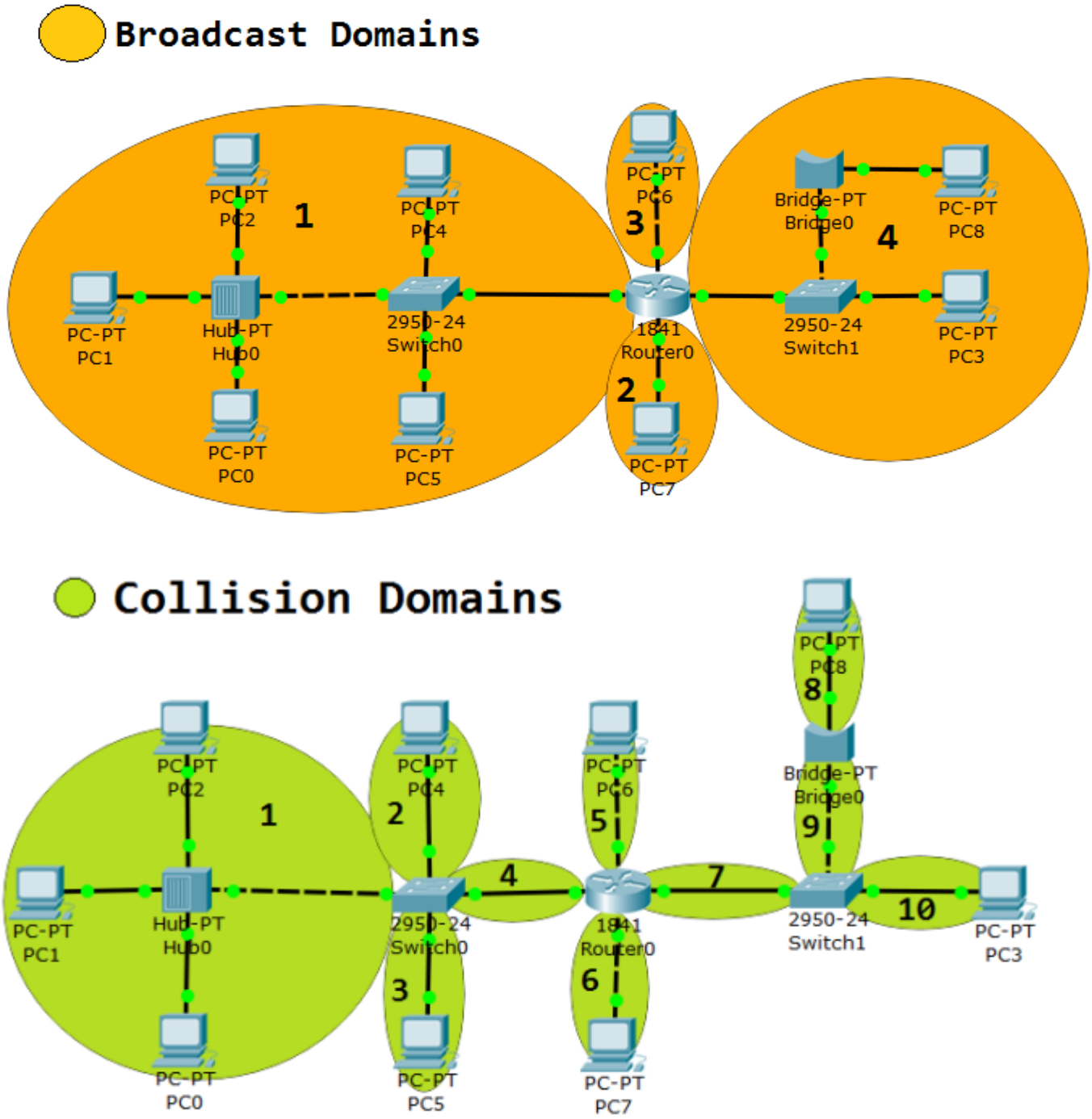
- Prüfe, ob das Datenpaket mir gehört?
  - Wenn ja, dann hat das Datenpaket sein Ziel erreicht und kann verarbeitet werden.
  - Wenn nein, prüfe ob das Datenpaket in mein Subnetz gehört?
    - Wenn ja, schick es in das eigene Subnetz weiter!
    - Wenn nein, prüfe ob die Route zum Empfänger bekannt ist?
      - Wenn ja, schicke es über die bekannte Route zum nächsten Router!
      - Wenn nein, prüfe ob es ein Standard-Gateway gibt?
        - Wenn ja, schick das Paket zum Standard-Gateway!
        - Wenn nein, schreibe eine Fehlermeldung und verwirf das Datenpaket!

Broadcastdomäne

Eine Broadcast-Domäne ist ein logischer Verbund von Netzwerkgeräten in einem lokalen Netzwerk, der sich dadurch auszeichnet, dass ein **Broadcast alle Domänenteilnehmer erreicht**.

Ein lokales Netzwerk auf der 2. Schicht des OSI-Modells (Sicherungsschicht) besteht durch seine Hubs, Switches und/oder Bridges aus einer Broadcast-Domäne. Erst durch die Unterteilung in VLANs oder durch den Einsatz von Routern, die auf Schicht 3 arbeiten, wird die Broadcast-Domäne aufgeteilt.

Eine Broadcast-Domäne besteht aus einer oder mehreren Kollisionsdomänen.



From:
http://elearn.bgamstetten.ac.at/wiki/ - **Wiki**

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf6ai_202425:09_netzwerke:08

Last update: **2025/03/19 20:50**

Netzwerkbefehle

Windows-Kommandos

Wichtige Netzwerkbefehle unter Windows XP

- ipconfig
  - /all ... detaillierte Informationen über Netzwerk-Konfiguration
  - /renew ... erneuert IP-Adressen
  - /release ... gibt IP-Adressen frei
- ping (sendet Datenpakete zu Rechner)
  - ping IP-Adresse
  - ping hostname
  - -n Anzahl ... Anzahl der Pakete
- tracert (Route zu Rechner)
- nslookup
  - nslookup (DNS-Abfragen)
  - nslookup IP-Adresse
  - nslookup Domain-Name

Linux-Kommandos

- ifconfig

Zeigt Informationen zu Netzwerk-Interfaces

- ping

```
ping -c 4 www.example.com
```

Pingt 4 mal www.example.com

- tracepath

```
tracepath www.example.com
```

Zeigt Hops zum Host an (benötigt eventuell SU-Rechte)

```
sudo tracepath www.example.com
```

- nslookup

```
nslookup www.example.com
```

Überprüft DNS-Eintrag

weitere hilfreiche Befehle:

- nmap (scannt („mappt“) das Netzwerk, führt Portscans aus und findet die Software eines fremden PCs heraus)
- mtr (kombiniert tracert (ohne su-Rechte) und ping, anschauliche Darstellung)

Fragen - Aufgaben - Arbeitsaufträge

Arbeitsaufträge:

1. Gib deiner/m Nachbarn/in die IP-Adresse deines Computers und notiere dir die IP-Adresse seines/ihres Computers.
2. Versuche deinen Nachbarcomputer anzupingen. Wie lange brauchte das Datenpaket zu diesem Rechner?
3. Gib deine IP-Adresse frei und versuche den Nachbarn anzupingen bzw. dich anpingen zu lassen. Erneuere anschließend deine IP-Adresse und überprüfe, wie sie nun lautet.
4. Versuche deinen Rechner (mit deiner IP-Adresse und der Loopback-Adresse) anzupingen.
5. Finde heraus, welche IP-Adresse der Domain-Name mail.bgamstetten.ac.at hat.
6. Finde einen Rechner im Internet, bis zu welchem ein Datenpaket sehr lange dauert und schicke zu diesem Rechner 20 Datenpakete hintereinander.
7. Finde heraus, welcher Domain-Name zu folgender IP-Adresse gehört 131.130.250.250
8. Lasse dir die Route zu www.yahoo.com anzeigen. Wie viele Rechner liegen zwischen dir und dem Webserver von www.yahoo.com ?
9. Versuche mit dem Internetexplorer die Website von www.google.at aufzurufen, indem du die IP-Adresse von Google in der Browserzeile eingibst.
10. Suche im Internet die Homepage von der australischen Regierung. Welche IP-Adresse hat der Rechner, auf dem die Homepage liegt? Wie viele Rechner liegen zwischen dir und diesem Rechner?
11. Finde einen Rechner im Internet, bis zu dem möglichst viele Hops dazwischen sind (Wer findet die meisten?).

Portnummern

Ein Port oder eine Portnummer ist in Rechnernetzen eine Netzwerkadresse, mit der das Betriebssystem die Datenpakete eines Transportprotokolls zu einem Prozess zuordnet. Zusammen mit der IP-Adresse ermöglicht der Port die Adressierung eines Servers oder Clients. Durch Angabe von Quell- und Zieladresse, jeweils bestehend aus IP-Adresse und Port, ist es möglich, eine bestehende Verbindung eindeutig zu identifizieren.

Die weit verbreiteten Transportprotokolle Transmission Control Protocol (TCP) und User Datagram Protocol (UDP) verwenden 16 Bit als Port. Daraus ergibt sich ein Portnummernbereich von 0 bis 65.535

Zweck

Ports dienen zwei Zwecken:

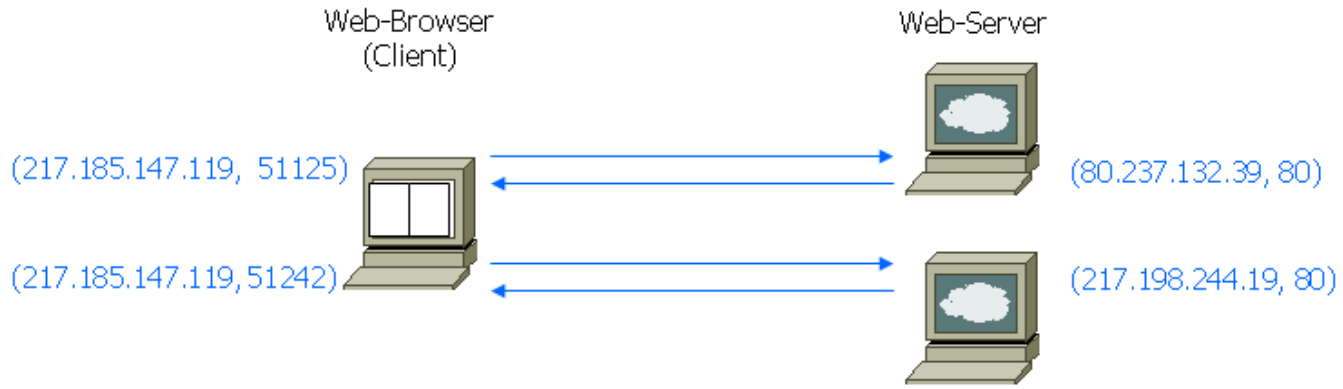
- Primär sind Ports ein **Merkmal zur Unterscheidung mehrerer Verbindungen** zwischen demselben **Paar von Kommunikationsteilnehmern (Hosts)** in einem Rechnernetz. Die Angabe von IP-Adressen ist alleine nicht eindeutig, da es mehrere Verbindungen gleichzeitig zwischen den beiden IP-Adressen geben kann.
- Ports können auch Netzwerkprotokolle und entsprechende Netzwerkdienste identifizieren. Dies ergibt sich daraus, dass Netzwerkdienste üblicherweise standardisierte Portnummern verwenden.

Beispiele

- Ein Webbrowser kann während eines laufenden HTTPS-Downloads einen weiteren Download von ein und demselben Webserver starten, weil der Browser dann (client-seitig) einen weiteren Port öffnet und so eine zusätzliche Verbindung zum selben Port 443 des Servers aufbaut. Der Server antwortet den unterschiedlichen Ports des Browsers mit unterschiedlichen jeweils zusammengehörigen Inhalten. Für eine Unterscheidung der Verbindungen genügen also verschiedene Portnummern an nur einem der beiden Endpunkte.
- Ein HTTPS-Webserver läuft standardmäßig über den TCP-Port 443. Ohne explizite Angabe einer Portnummer versucht ein Webbrowser einen Verbindungsaufbau standardmäßig zum Zielport 443. Beobachtet man im Rechnernetz eine TCP-Verbindung zu einem Port 443, so ist dies ein starkes Indiz, dass es sich dabei um eine HTTPS-Verbindung handelt.

Funktionsweise

Auf einem Rechner, der an ein Netzwerk angeschlossen ist, können gleichzeitig mehrere Prozesse laufen, die über das Internet mit anderen Prozessen Datenpakete austauschen. Zum Beispiel können gleichzeitig zwei Prozesse in einem Webbrowser aktiv sein. Wie soll der Rechner, auf dem die Prozesse laufen, entscheiden, zu welchem der Prozesse ein ankommendes Datenpaket weitergeleitet werden soll?



Die Identifikation von Prozessen erfolgt mit Hilfe von Portnummern. Wenn beispielsweise ein Browser mit einem Webserver im Internet kommuniziert und ihm eine Anfrage zu einer Webseite schickt, so beschreibt die Zielportnummer die Art der Anwendung (HTTP-Prozess, festgelegt durch Portnummer 80). Als Quellportnummer wird irgendeine zur Zeit noch nicht verwendete Zahl > 49.151, z.B. 51125. Wird ein zweiter HTTP-Prozess gestartet, so wird als Zielportnummer ebenfalls die Zahl 80 festgelegt, als Quellportnummer wird eine andere Zahl, z.B. 51242, verwendet. Antwortet der Webserver dem ersten HTTP-Prozess, so gibt er umgekehrt als Quellportnummer die 80 und als Zielportnummer die 51125an. Der Rechner weiß jetzt, dass die Antwort für genau diesen ersten HTTP-Prozess bestimmt war.

Standardisierung

System Ports

Diese Ports werden auch „well-known Ports“ genannt und sind für Netzwerkdienste vorgesehen. Einige der Portnummern, wie beispielsweise 0 und 1023, sind durch die IANA reserviert und werden nicht vergeben. Neue Zuordnungen erfolgen nur unter Beteiligung der Internet Engineering Task Force (IETF) und setzen die Zustimmung der Internet Engineering Steering Group voraus. Der Antragsteller muss begründen, warum eine Portnummer aus den anderen Bereichen für den Anwendungszweck ungeeignet ist.

Damit man beim Aufbau einer Verbindung nicht immer IP-Adresse und Port angeben muss, sondern den Port weglassen kann, gibt es sogenannte „well known ports“. Diese definieren Standardports für bestimmte Anwendungen bzw. Protokolle. So ist z.B. der Port 80 für Webserver bzw. das http-Protokoll vorgesehen. Dies hat zur Folge, dass ein Benutzer nur die IP-Adresse bzw. den Rechnernamen angeben muss. Der Browser nutzt automatisch Port 80, da er ja weiß, dass er eine Verbindung zu einem Webserver mit Hilfe des http-Protokolls aufbaut.

User Ports

Ports 1024 bis 49.151 (400hex bis BFFFhex)

Diese Ports sind für registrierte Netzwerkdienste vorgesehen. In diesem Bereich können auf Antrag neue Zuordnungen je nach Verfahren mit oder ohne Beteiligung der IETF erfolgen.

Bei gängigen Betriebssystemen steht dieser Portbereich zur Verwendung durch Client und Server zur

Verfügung, sofern die Anwendung den Port explizit anfordert.

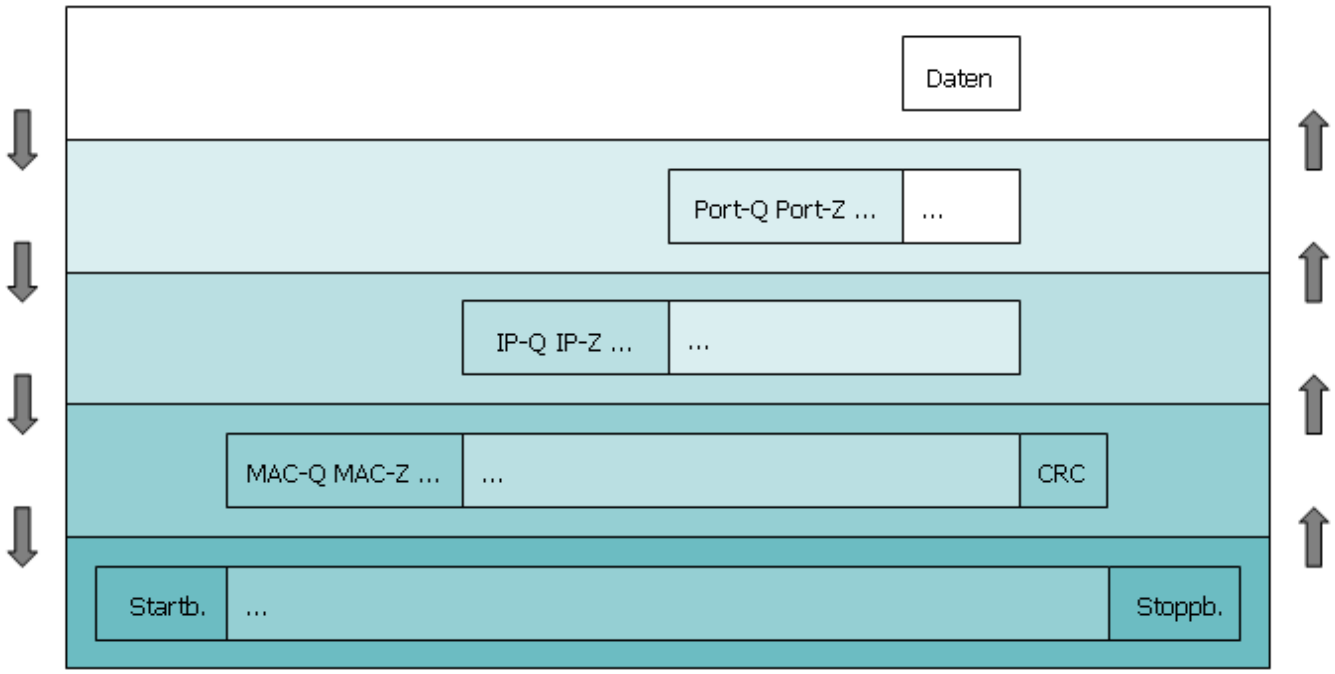
Dynamic Ports

Ports 49.152 bis 65.535 (C000hex bis FFFFhex)

Diese Ports sind für eine dynamische Zuweisung durch das Betriebssystem vorgesehen. Die IANA führt in diesem Bereich keine Registrierungen durch.

Microsoft verwendet seit Windows Vista und Windows Server 2008 standardmäßig diesen Bereich für die dynamische Zuordnung an Clientprogramme.[7] Linux verwendet standardmäßig einen davon abweichenden Bereich und vergibt dynamische Ports im Bereich von 32.768 bis 60.999.

Datenkapselung



In der Transportschicht werden den eigentlichen Daten u.a. die Quell- und Zielportnummern der kommunizierenden Prozesse hinzugefügt. In den darunterliegenden Schichten werden diese Datenpakete dann weiter mit Zusatzdaten versehen, bevor sie verschickt werden.

Wireshark Übungen

1. Übung

1. Starte Wireshark
2. Starte einen Webbrowser mit einer beliebigen Seite
3. Filtere nach dem Protokoll http.
4. Öffnen die Seite http://elearn.bgamstetten.ac.at
5. Starte die Aufzeichnung nochmals neu und aktualisiere die Seite mit STRG+F5
6. Dokumentiere mittels Screenshot, wie viele gesendete und empfangen Pakete über die Schnittstelle mitgesnifft wurden.
7. Dokumentiere, welche IP-Adresse sich hinter dem FQDN elearn.bgamstetten.ac.at steckt.
8. Dokumentiere, welche Protokolle bei der ersten Antwort (text/html) zum Einsatz kamen.
9. Dokumentiere, wie groß die Antwort (text/html) in Bytes war.
10. Dokumentiere, wie groß
    - der Header des Ethernet 2 – Frames
    - der Header des IP-Paket-Pakets
    - der Header des TCP-Segments
    - und die Daten des http-Protokolls (gzip) der Antwort (text/html) waren.
11. Überprüfe, ob die die Größen zusammen die Gesamtgröße von zuvor ergeben.
12. Finde im Bereich Paketdetails den HTML-Quellcode und eine im Quellcode versteckte Botschaft. Der Name des input-Tags lautet STRENGGEHEIM und die Botschaft verbirgt sich value-Attribut.
13. Kontrolliere deine eigene MAC-Adresse (ipconfig /all) und finde die Ziel-MAC-Adresse heraus.
14. Finde heraus, ob IPv4 oder IPv6 als Protokoll auf der Netzwerkschicht verwendet wurde.
15. Finde heraus, welcher Port auf deinem PC bzw. am ELEARN-Server für die Netzwerkverbindung verwendet wurde.
16. Finde heraus, warum beim ersten Paket im TCP-Header die Sequence Number=1 ist und die Sequence Number beim übernächsten Paket nicht 2 sondern z.B.: 603 ist.
17. Finde heraus, welche HTTP-Version 1.0 oder 1.1 verwendet wurde.
18. Finde heraus, welcher User-Agent verwendet wurde.
19. Finde heraus, welche Sprachen vom Browser akzeptiert werden.
20. Bei der Antwort (text/html) wird ein HTTP-Status-Code mitgeschickt. Finde ihn heraus und recherchiere, was dieser Status Code bedeutet.
21. Finde heraus, welches Betriebssystem bzw. welcher Webserver verwendet wird.
22. Melde dich von der elearn-Plattform ab und anschließend wieder an. Finde heraus, ob und wie deine Login-Informationen mitgeschickt werden (Tipp Info=POST &do=login).

Protokolle

Damit sich zwei Personen in einem Gespräch verstehen, müssen diese die gleiche Sprache sprechen. Auch bei der Kommunikation zwischen zwei Computern ist das ähnlich.

Damit zwei Computer miteinander kommunizieren können, müssen diese eine Menge von gemeinsamen Regeln befolgen, welche die Kommunikation anleiten. Diese definieren eine Syntax, eine Semantik und schließlich die Synchronisation der Kommunikation.

Protokolle können dabei auf verschiedenen Ebenen definiert sein und reichen von grundlegenden Kommunikationsprotokollen auf Bitebene bis zu komplexeren Protokollen, welche die Kommunikation zwischen zwei Programmen definiert.

Protokolle sind eine **Sammlung von Regeln zur Kommunikation** auf einer bestimmten Schicht des OSI-Schichtenmodells. Die Protokolle einer Schicht sind zu den Protokollen der über- und untergeordneten Schichten weitestgehend transparent, so dass die Verhaltensweise eines Protokolls sich wie bei einer direkten Kommunikation mit dem Gegenstück auf der Gegenseite darstellt.

Die **Übergänge zwischen den Schichten sind Schnittstellen**, die von den Protokollen verstanden werden müssen. Weil manche Protokolle für ganz bestimmte Anwendungen entwickelt wurden, kommt es auch vor, dass sich **Protokolle über mehrere Schichten** erstrecken und mehrere Aufgaben abdecken. Dabei kommt es vor, dass in manchen Verbindungen einzelne Aufgaben in mehreren Schichten und somit mehrfach ausgeführt werden.

Protokoll-Stack

Da sich ein einzelnes Protokoll immer nur um eine Teilaufgabe im Rahmen der Kommunikation kümmert, werden mehrere Protokolle zu Protokollsammlungen oder Protokollfamilien, den sogenannten Protokoll-Stacks, zusammengefasst. Die wichigsten Einzel-Protokolle werden dann oft stellvertertretend als Bezeichnung des gesamten Protokoll-Stapels genutzt.

Kommunikation zwischen Netzwerkkomponenten funktioniert nur dann, wenn sie denselben Protokoll-Stack benutzen oder wenn Geräte eingesetzt werden, die zwischen verschiedenen Stacks vermitteln können.

Portnummern (ab Layer 5)

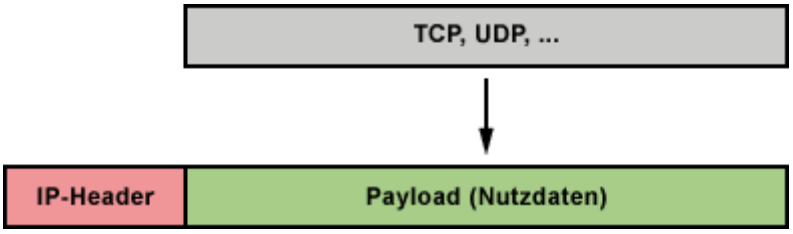
Um die einzelnen Dienste (Protokolle), die bei einem Rechner über dieselbe IP-Adresse ausgeführt werden, voneinander zu differenzieren, wurden Portnummern eingeführt, um bei einer Anfrage deutlich zu machen, welcher Dienst gemeint ist.
Diese Portnummern, die im TCP- oder UDP Header angegeben werden, sind weltweit eindeutig festgelegt und können z.B. auf der Homepage der IANA eingesehen werden.
Allgemein lässt sich also sagen, dass die **IP-Adresse** den Rechner, **und** Die **Port-Nummer** den Dienst auf dem jeweiligen Rechner angibt. Diese beiden Informationen zusammen werden als **Socket** bezeichnet.

Wichtige Protokolle

OSI-LAYER	PROTOKOLL (Port)
5-7	DHCP (67+68/UDP)
	DNS (53/UDP)
	HTTP (80/TCP)
	HTTPS (443/TCP)
	FTP (20+21/TCP)
	SSH (22/TCP)
	SMTP (465 und 587 oder 25/TCP)
	POP3 (995 oder 110/TCP)
	IMAP (993 oder 143/TCP)
4	TCP
	UDP
3	IPv4
	IPv6
	NAT
	ICMP
2	Ethernet 802.3 , Wireless 802.x, MAC, ISDN,...
1	

IPv4 - Internet Protocol Version 4 (Layer 3)

Das Internet Protocol, kurz IP, wird im Rahmen der Protokollfamilie TCP/IP zur Vermittlung von Datenpaketen verwendet. Es arbeitet auf der **Schicht 3 des OSI-Schichtenmodells** und hat maßgeblich die Aufgabe, **Datenpakete zu adressieren** und in einem dezentralen, **verbindungslosen und paketorientierten Netzwerk zu übertragen**. Dazu haben alle Netzwerk-Teilnehmer eine eigene IP-Adresse im Netzwerk. Sie dient nicht nur zur Identifikation eines Hosts, sondern auch des Netzes, in dem sich der jeweilige Host befindet.



Aufgaben und Funktionen von IPv4

- Logische Adressierung (IPv4-Adresse)
- IPv4-Konfiguration
- IPv4-Header
- IP-Routing
- Namensauflösung (DNS-Dienst)

IPv4 Header

Jedes IPv4-Datenpaket besteht aus einem Header (Kopf) und dem Payload, in dem sich die Nutzdaten befinden. Der Header ist den Nutzdaten vorangestellt. Im IP-Header sind Informationen enthalten, die für die Verarbeitung durch das Internet Protocol notwendig sind.

Feldinhalt	Bit	Beschreibung
Version	4	Hier ist die Version des IP-Protokolls abgelegt, nach der das IP-Paket erstellt wurde.
IHL	4	IHL = Internet Header Length gibt die Länge des IP-Headers als Vielfaches von 32 Bit an. Der Maximalwert von Binär 1111 (15) entspricht einer Header-Länge von 15 x 32 Bit = 480 Bit = 60 Byte.
ToS	8	ToS = Type of Service legt die Qualität des angeforderten Dienstes fest. Das Feld unterteilt sich in Priorität (Priority) von 3 Bit und Eigenschaften für die Übertragung von 5 Bit.
Paketlänge (Total Length)	16	Enthält die Gesamtlänge des IP-Paketes. Abzüglich des IHL ergibt sich die Länge der reinen Nutzdaten.
Kennung(Identifiction)	16	Der Wert wird zur Nummerierung der Datenpakete verwendet. Die Kennung ist eindeutig und fortlaufend.

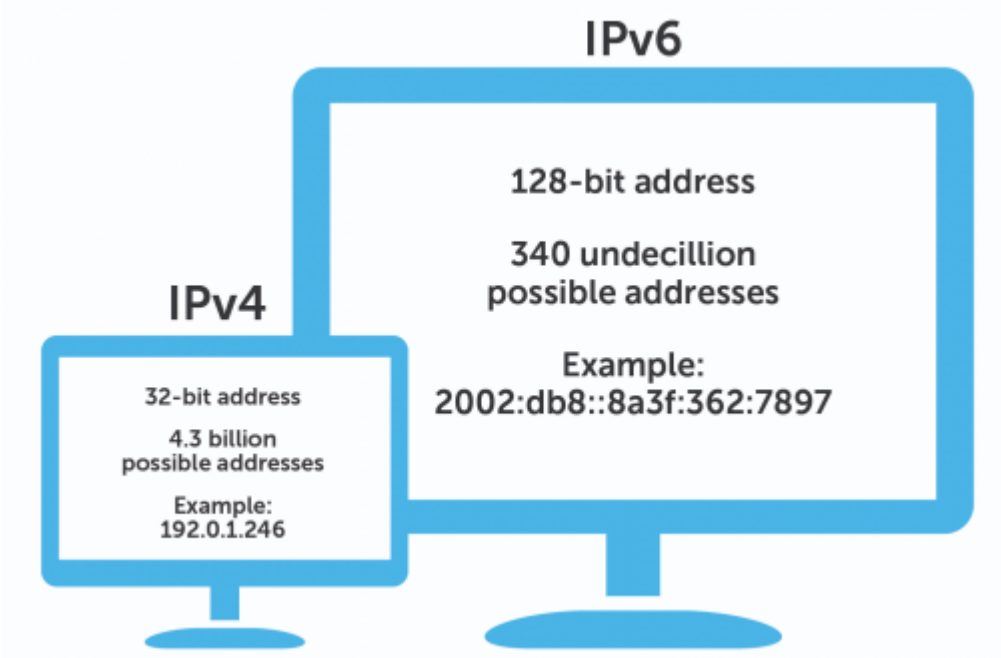
Feldinhalt	Bit	Beschreibung
Flags	3	Da die Nutzdaten in der Regel nicht in ein IP-Paket hineinpassen, werden die Daten zerlegt und in mehrere IP-Pakete verpackt und verschickt. Man spricht dann von Fragmentierung. Die Flags gehen näher darauf ein. Das erste Flag ist immer 0. Das zweite Flag (DF) verbietet die Fragmentierung des Datenpakets, wenn es gesetzt ist. Das dritte Flag (MF) gibt weitere Datenpaket-Fragmente an, wenn es gesetzt ist.
Fragment-Offset	13	Enthält ein IP-Paket fragmentierte Nutzdaten, steht in diesem Feld die Position der Daten im ursprünglichen IP-Paket.
TTL (Time-to-Live)	8	Mit TTL gibt der Sender die Lebensdauer des Pakets in Sekunden an. Jede Station, die ein IP-Paket weiterleiten muss, zieht von diesem Wert mindestens 1 ab. Hat der TTL-Wert 0 erreicht, wird das IP-Paket verworfen. Dieser Mechanismus verhindert, dass Pakete ewig Leben, wenn sie nicht zustellbar sind. TTL-Werte zwischen 30 und 64 sind typisch.
Protokoll	8	Dieses Feld enthält den Port des übergeordneten Transport-Protokolls (z. B. TCP oder UDP).
Header Checksumme	16	Diese Checksumme sichert die Korrektheit des IP-Headers. Für die Nutzdaten muss ein übergeordnetes Protokoll die Fehlerkorrektur übernehmen. Da sich die einnzelnen Felder des IP-Headers ständig ändern, muss jede Station auf dem Weg zum Ziel die Checksumme prüfen und auch wieder neu berechnen. Um die Verzögerung gering zu halten wird deshalb nur der IP-Header des Paketes geprüft.
Quell-IP-Adresse (Source IP-Address)	32	An dieser Stelle steht die IP-Adresse der Station, die das IP-Paket abgeschickt hat (Sender).
Ziel-IP-Adresse (Destination IP-Address)	32	An dieser Stelle steht die IP-Adresse der Station, für die das IP-Paket bestimmt ist. Soll das IP-Paket an mehrere Stationen zugestellt werden, muss hier eine Multicast-Adresse stehen.
Optionen/Füllbits (Options/Padding)	32	Das Optionsfeld des IP-Headers enthält hauptsächlich Informationen zu Routing-, Debugging-, Statistik- und Sicherheitsfunktionen. Dieses Feld ist optional und kann bis zu 40 Byte lang sein. Es ist immer in 32 Bit aufgeteilt und wird bei Bedarf mit Nullen aufgefüllt. Auf die genauen Funktionen dieses Feldes wird hier nicht weiter eingegangen. Nur soviel sei noch gesagt: das Optionsfeld wird meist zu Diagnosezwecken verwendet.

Der Header ist in jeweils 32-Bit-Blöcke unterteilt. Dort sind Angaben zu Servicetypen, Paketlänge, Sender- und Empfängeradresse abgelegt. Ein IP-Paket muss mindestens 20 Byte Header und 8 Byte Nutzdaten bzw. Nutz- und Fülldaten enthalten. Die Gesamtlänge eines IP-Pakets darf 65.535 Byte nicht überschreiten.

IPv6 - Internet Protocol Version 6 (Layer 3)

IPv6 ist als Internet Protocol (Version 6) für die Vermittlung von Datenpaketen durch ein paketvermittelndes Netz, die Adressierung von Netzknoten und -stationen, sowie die Weiterleitung von Datenpaketen zwischen Teilnetzen zuständig. Mit diesen Aufgaben ist IPv6 der Schicht 3 des OSI-Schichtenmodells zugeordnet. Die Aufgabe des Internet-Protokolls besteht im Wesentlichen darin, Datenpakete von einem System über verschiedene Netzwerke hinweg zu einem anderen System zu vermitteln (Routing).

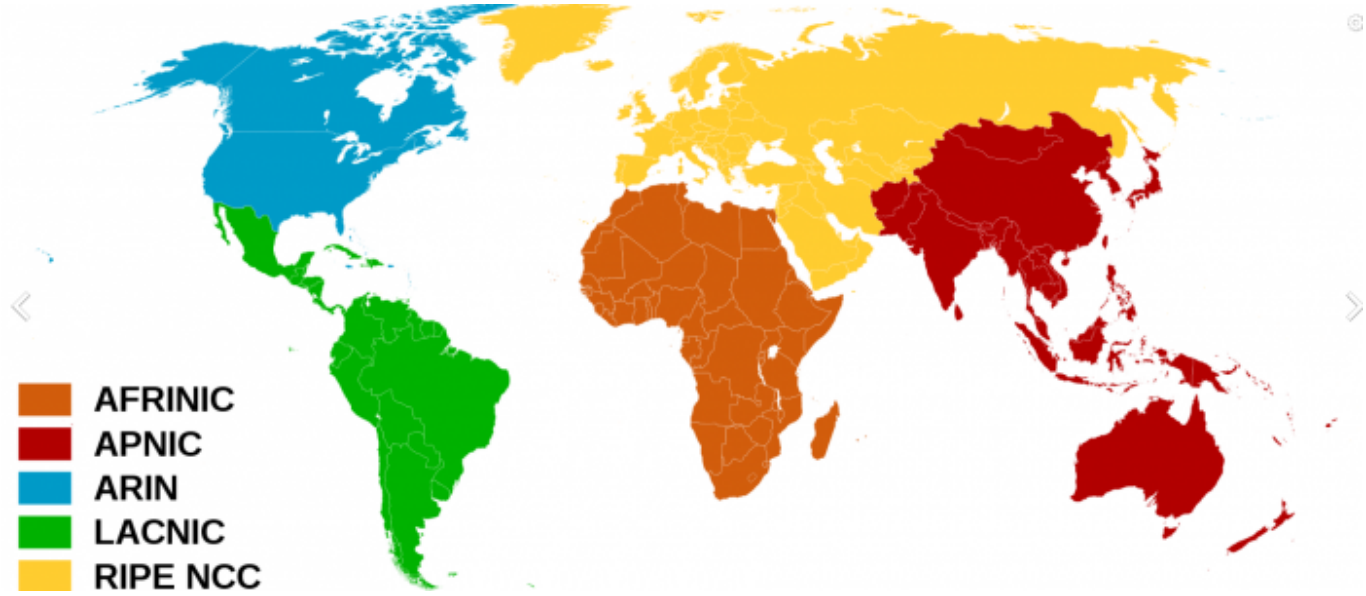
IPv6 ist der direkte Nachfolger von IPv4 und Teil der Protokollfamilie TCP/IP. Seit Dezember 1998 steht IPv6 bereit und wurde hauptsächlich wegen der Adressknappheit und verschiedener Unzulänglichkeiten von IPv4 entwickelt spezifiziert. Da weltweit immer mehr Menschen, Maschinen und Geräte an das Internet mit einer eindeutigen Adresse angeschlossen werden sollen, reichen die 4 Milliarden IPv4-Adressen nicht mehr aus.

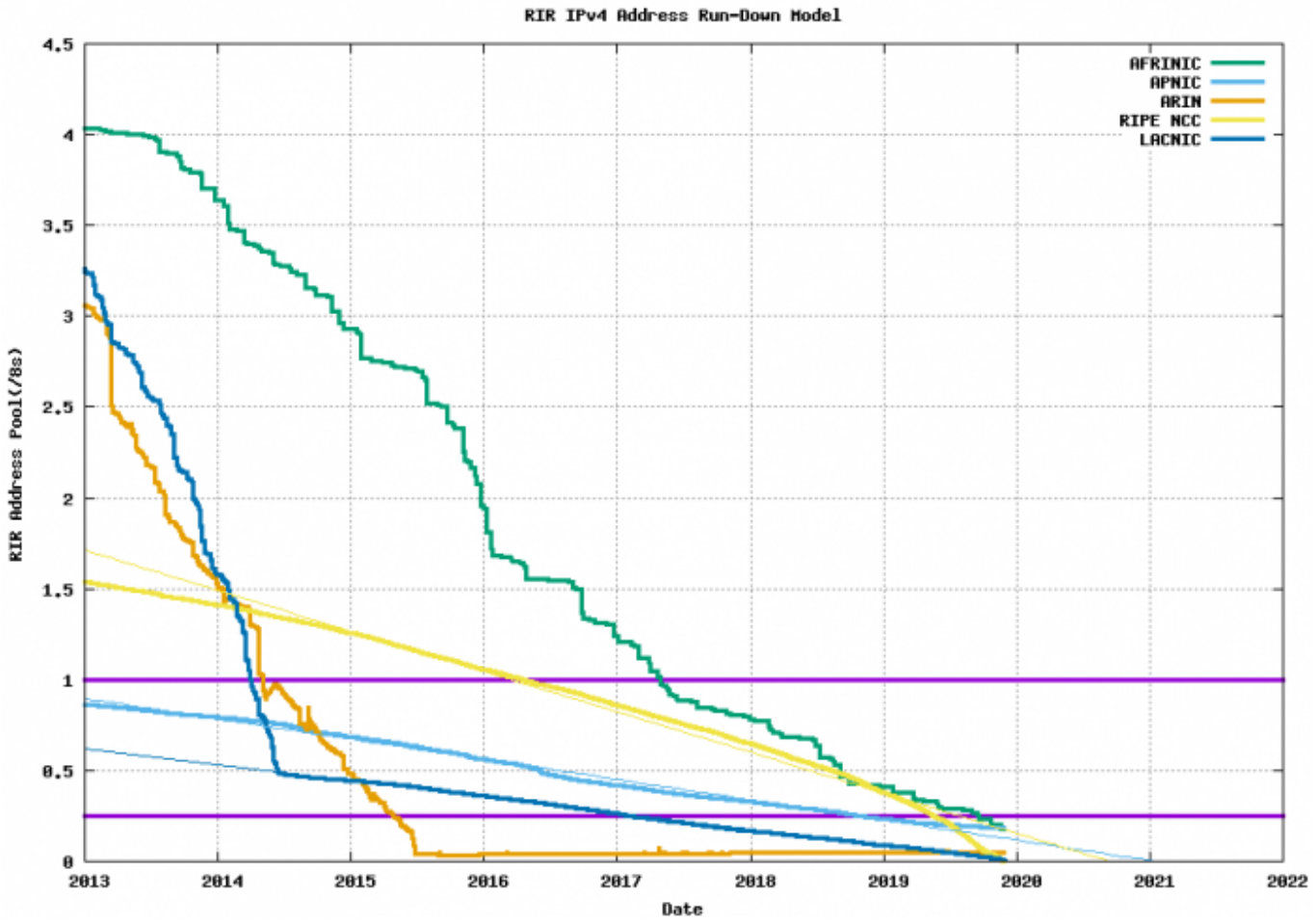


Warum IPv6?

IPv6 gilt als Wunderwaffe gegen so manche Probleme mit Netzwerkprotokollen und gleichzeitig wird es als Teufelszeug verdammt, das wieder neue unbekannte Probleme hervorruft. Eine Tatsache ist, dass Administratoren, Programmierer und Hersteller IPv6 neu lernen müssen. Viele Rezepte aus der IPv4-Welt taugen unter IPv6 nicht mehr. Erschwerend kommt hinzu, dass es bei IPv6 allen Beteiligten an Erfahrung fehlt. IPv6-Gurus, die man bei einem großen Problem befragen kann, gibt es nicht so viele.

Bei IPv6 ist das Ende-zu-Ende-Prinzip konsequent weiter gedacht. Ein Interface kann mehrere IPv6-Adressen haben und es gibt spezielle IPv6-Adressen, denen mehrere Interfaces zugeordnet sind. IPv6 löst also nicht nur die Adressknappheit, sondern bietet auch Erleichterungen bei der Konfiguration und im Betrieb. Die zustandslose IPv6-Konfiguration und verbindungslokalen Adressen, die bereits nach dem Computerstart verfügbar sind, vereinfachen die Einrichtung und den Betrieb eines lokalen Netzwerks. Damit das gelingt sind Planer und Errichter von IP-Netzen gefordert sich eine neue Denkweise anzueignen.





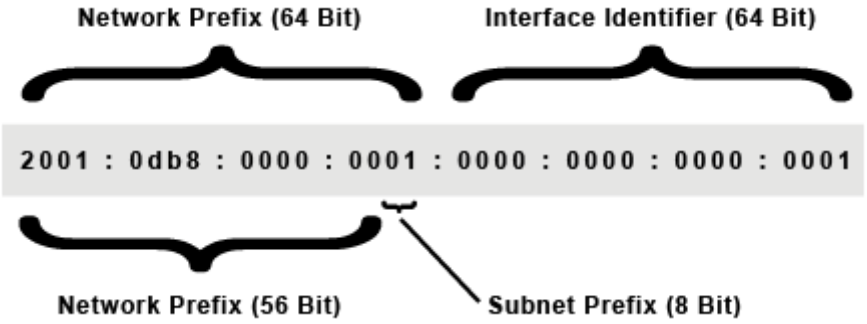
das_war_s_mit_ipv4-adressen_in_europa_heise_online.pdf

IPv6 Adressen

Eine IPv6-Adresse ist eine Netzwerk-Adresse, die einen Host eindeutig innerhalb eines IPv6-Netzwerks logisch adressiert. Die Adresse wird auf IP- bzw. Vermittlungsebene (des OSI-Schichtenmodells) benötigt, um Datenpakete verschicken und zustellen zu können. Im Gegensatz zu anderen Adressen hat ein IPv6-Host mehrere IPv6-Adressen, die unterschiedliche Gültigkeitsbereiche haben. Konkret

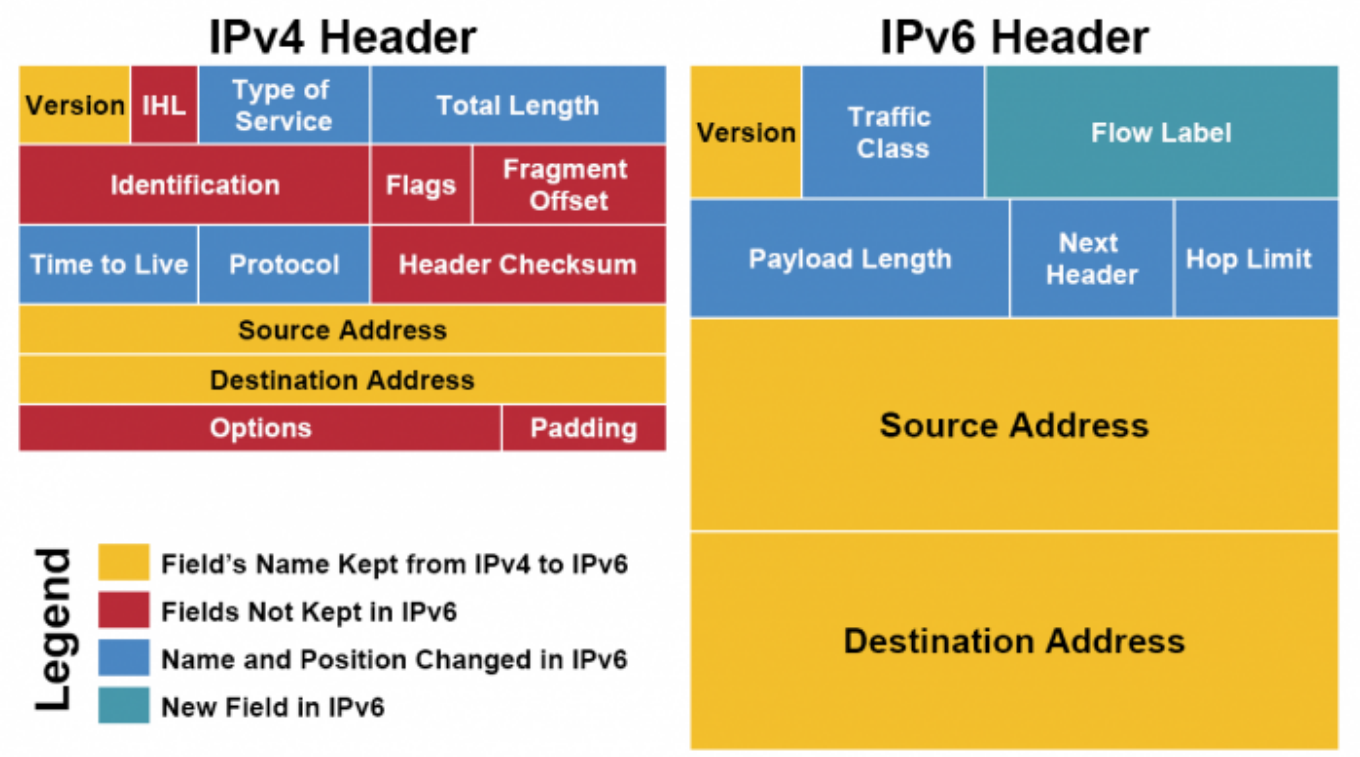
bedeutet das, dass wenn von IPv6-Adressen die Rede ist, dass nicht immer klar ist, welchen Gültigkeitsbereich diese IPv6-Adressen aufweisen. Grob unterscheidet man zwischen verbindungslokalen und globalen IPv6-Adressen. Die verbindungslokale IPv6-Adresse ist nur im lokalen Netzwerk gültig und wird nicht geroutet. Die globale IPv6-Adresse ist über das lokale Netzwerk hinaus im Internet gültig.

Eine IPv6-Adresse hat eine Länge von 128 Bit. Diese Adresslänge erlaubt eine unvorstellbare Menge von 2128 oder 3,4 x 1038 IPv6-Adressen. Das sind 340.282.366.900.000.000.000.000.000.000.000.000 IPv6-Adressen, also rund 340 Sextillionen Adressen. Bei IPv4 spricht man von rund 4,3 Milliarden Adressen. Der Adressraum von IPv6 reicht aus, um umgerechnet jeden Quadratmillimeter der Erdoberfläche inklusive der Ozeane mit rund 600 Billiarden Adressen zu pflastern.



Eine IPv6-Adresse besteht aus 128 Bit. Wegen der unhandlichen Länge werden die 128 Bit in 8 mal 16 Bit unterteilt. Je 4 Bit werden als eine hexadezimale Zahl dargestellt. Je 4 Hexzahlen werden gruppiert und durch einen Doppelpunkt („:“) getrennt. Um die Schreibweise zu vereinfachen lässt man führende Nullen in den Blöcken weg. Eine Folge von 8 Nullen kann man durch zwei Doppelpunkte („::“) ersetzen.

Eine IPv6-Adresse besteht aus zwei Teilen. Dem Network Prefix (Präfix oder Netz-ID) und dem Interface Identifier (Suffix, IID oder EUI). Der Network Prefix kennzeichnet das Netz, Subnetz bzw. Adressbereich. Der Interface Identifier kennzeichnet einen Host in diesem Netz. Er wird aus der 48-Bit-MAC-Adresse des Interfaces gebildet und dabei in eine 64-Bit-Adresse umgewandelt. Es handelt sich dabei um das Modified-EUI-64-Format. Auf diese Weise ist das Interface unabhängig vom Network Prefix eindeutig identifizierbar.



ipv6_-_neues_internet-protokoll.pdf

Einsatzstatistik von IPv6

IPv6 - die unterschätzte Revolution

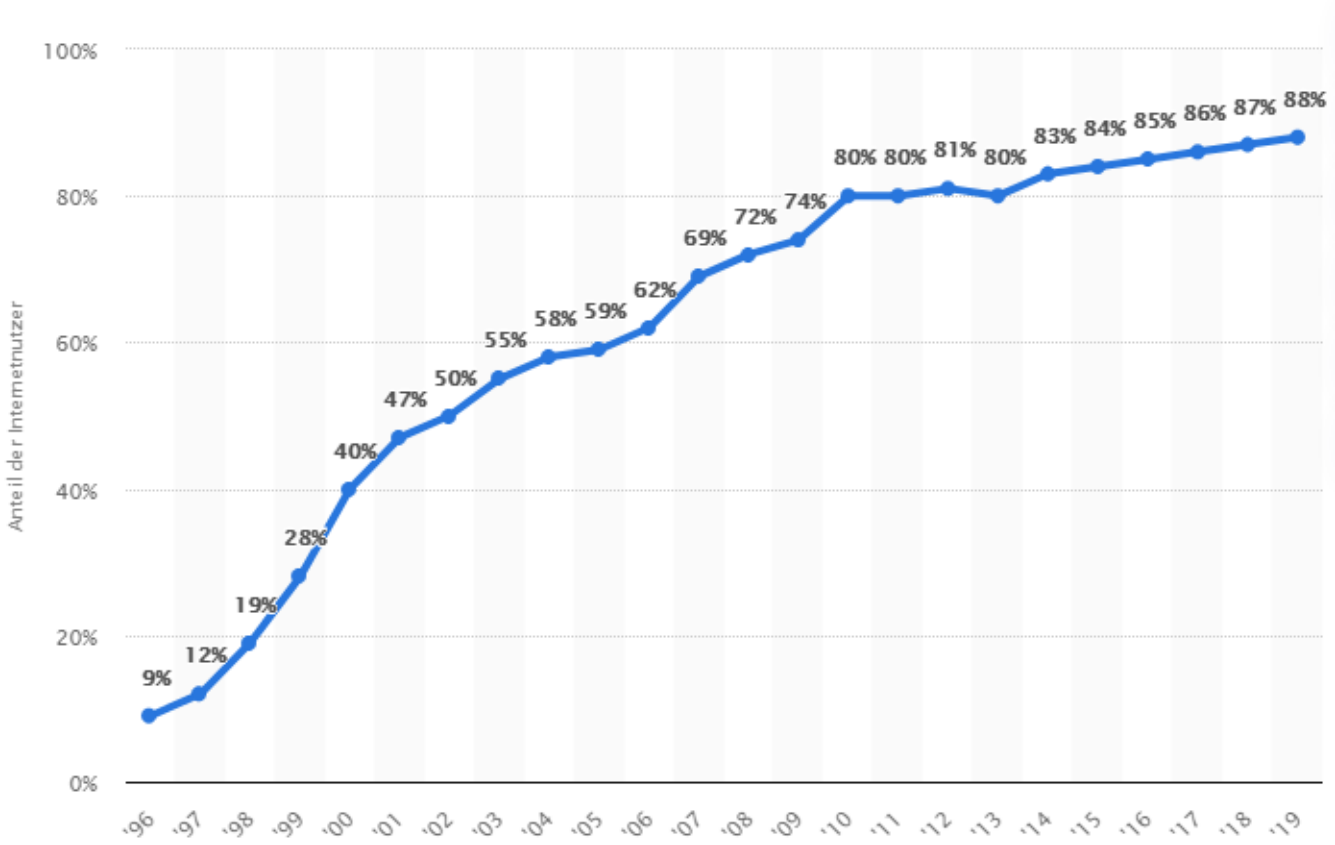
NAT - Network Address Translation (Layer 3)

NAT (Network Address Translation) ist ein Verfahren, dass in **IP-Routern eingesetzt** wird, die lokale Netzwerke mit dem Internet verbinden. Weil Internet-Zugänge in der Regel nur über **eine einzige öffentliche** und damit routbare IPv4-Adresse verfügen, müssen sich alle anderen Hosts im lokalen Netzwerk mit privaten IPv4-Adressen begnügen. **Private IP-Adressen** dürfen zwar **mehrfach verwendet** werden, aber besitzen **in öffentlichen Netzen keine Gültigkeit**. Hosts mit einer privaten IPv4-Adresse können somit nicht mit Hosts außerhalb des lokalen Netzwerks kommunizieren.

NAT ist allerdings nur eine mittlerweile sehr lang andauernde Notlösung, um die Adressknappheit von IPv4 zu umgehen. Um die damit einhergehenden Probleme zu lösen muss langfristig auf ein Internet-Protokoll mit einem größeren Adressraum umgestellt werden. IPv6 ist ein solches Protokoll.

Warum NAT?

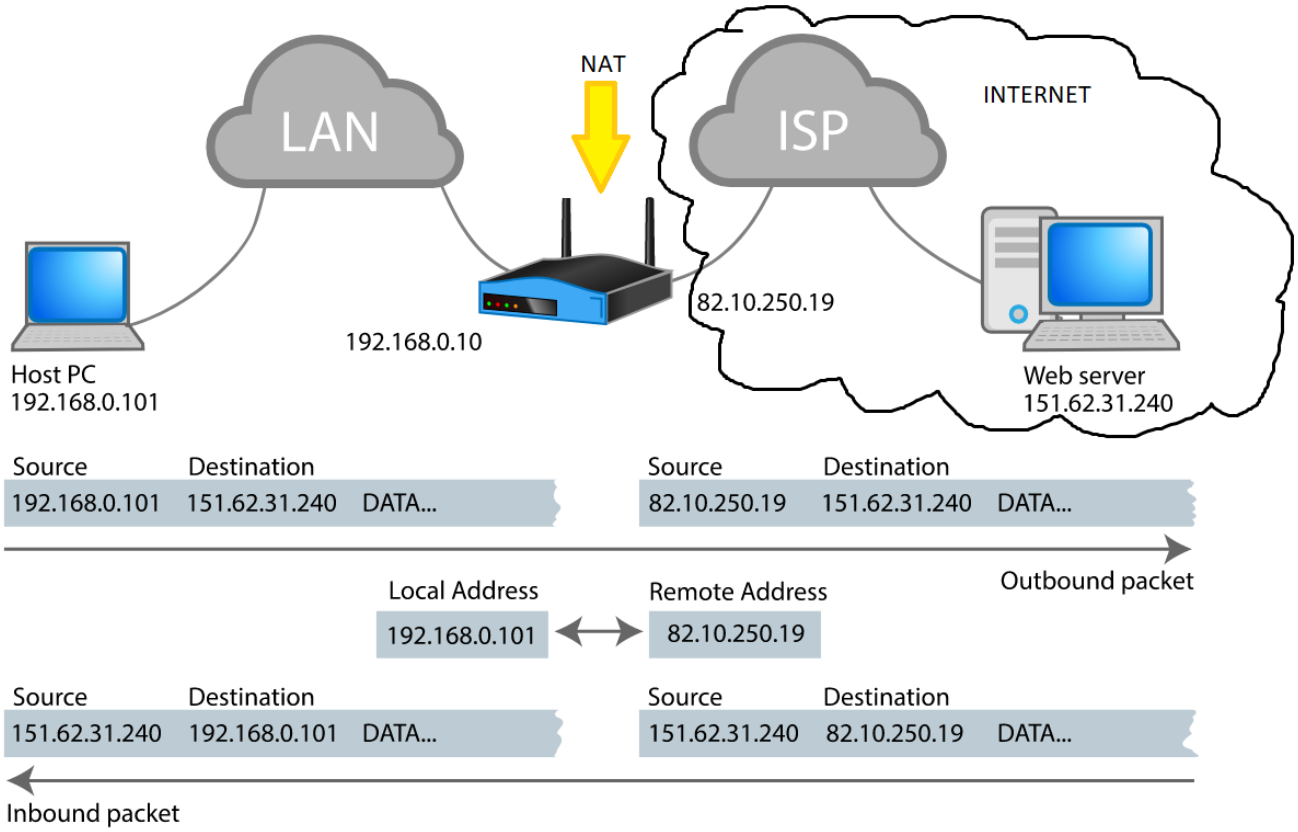
Die **ersten IPv4-Netze** waren **anfangs** eigenständige Netz **ohne Verbindung nach außen**. Hier begnügte man sich mit IPv4-Adressen aus den privaten Adressbereichen. Parallel dazu kam es bereits **Ende der 1990er** Jahre zu **Engpässen bei öffentlichen IPv4-Adressen**. Die steigende Anzahl der Einwahlzugänge über das Telefonnetz mussten mit IPv4-Adressen versorgt werden. Bis heute bekommt **ein Internet-Anschluss** nur eine **IPv4-Adresse für ein Gerät**. Damals war es undenkbar, dass an einem Internet-Anschluss ein ganzes Heimnetzwerk betrieben wird. Wenn ein Haushalt einen PC per Modem an das Telefonnetz angeschlossen und sich ins Internet eingewählt hat, dann war das schon etwas besonderes.



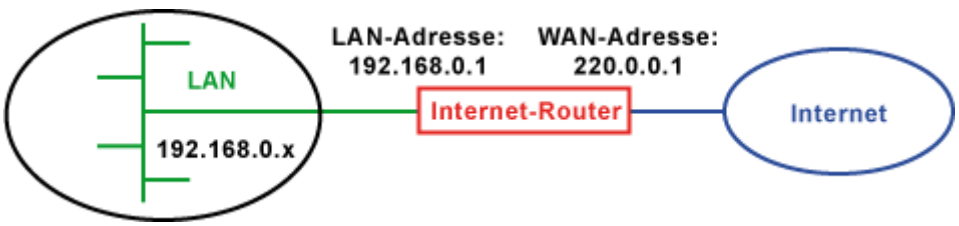
das_war_s_mit_ipv4-adressen_in_europa_heise_online.pdf

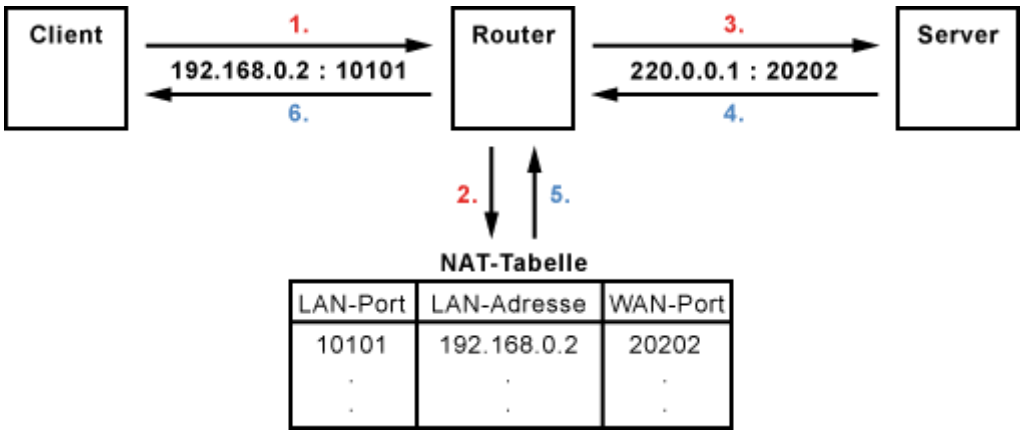
NAT

Der Betrieb eines NAT-Routers ist üblicherweise an einem gewöhnlichen Internet-Anschluss. Zum Beispiel über DSL oder Kabelmodem. Der eingesetzte Router dient als Zugang zum Internet und als Standard-Gateway für das lokale Netzwerk. In der Regel wollen über den Router mehr Geräte ins Internet, als öffentliche IP-Adressen zur Verfügung stehen. In der Regel nur eine einzige. Beispielsweise bekommt der Router des lokalen Netzwerks die öffentliche IP-Adresse 222.0.0.1 für seinen WAN-Port vom Internet Service Provider (ISP) zugewiesen. Weil nur eine öffentliche IP-Adresse vom Internet-Provider zugeteilt wurde, bekommen die Stationen im LAN private IP-Adressen aus speziell dafür reservierten Adressbereichen zugewiesen. Diese Adressen sind nur innerhalb des privaten Netzwerks gültig. Private IP-Adressen werden in öffentlichen Netzen nicht geroutet. Das bedeutet, dass Stationen mit privaten IP-Adressen keine Verbindung ins Internet bekommen können. Damit das trotzdem funktioniert, wurde NAT entwickelt.



Ablauf von NAT





1. Der Client schickt seine Datenpakete mit der IP-Adresse 192.168.0.2 und dem TCP-Port 10101 an sein Standard-Gateway, bei dem es sich um einen NAT-Router handelt.
2. Der NAT-Router tauscht IP-Adresse (LAN-Adresse) und TCP-Port (LAN-Port) aus und speichert beides mit der getauschten Port-Nummer (WAN-Port) in der NAT-Tabelle.
3. Der Router leitet das Datenpaket mit der WAN-Adresse 220.0.0.1 und der neuen TCP-Port 20202 ins Internet weiter.
4. Der Empfänger (Server) verarbeitet das Datenpaket und schickt seine Antwort zurück.
5. Der NAT-Router stellt nun anhand der Port-Nummer 20202 (WAN-Port) fest, für welche IP-Adresse (LAN-Adresse) das Paket im lokalen Netz gedacht ist.
6. Er tauscht die IP-Adresse und die Port-Nummer wieder aus und leitet das Datenpaket ins lokale Netz weiter, wo es der Client entgegen nimmt.

Probleme

- Durch NAT können nur noch die, die über öffentliche IPv4-Adressen und in der Regel auch über das notwendige Kleingeld verfügen, Dienste im Internet anbieten.
- Die Einträge in der NAT-Tabelle des Routers sind nur für eine kurze Zeit gültig. Für eine Anwendung, die nur sehr unregelmäßig Daten austauscht, bedeutet das, dass ständig die Verbindung abgebrochen wird und dadurch die Erreichbarkeit eingeschränkt ist.

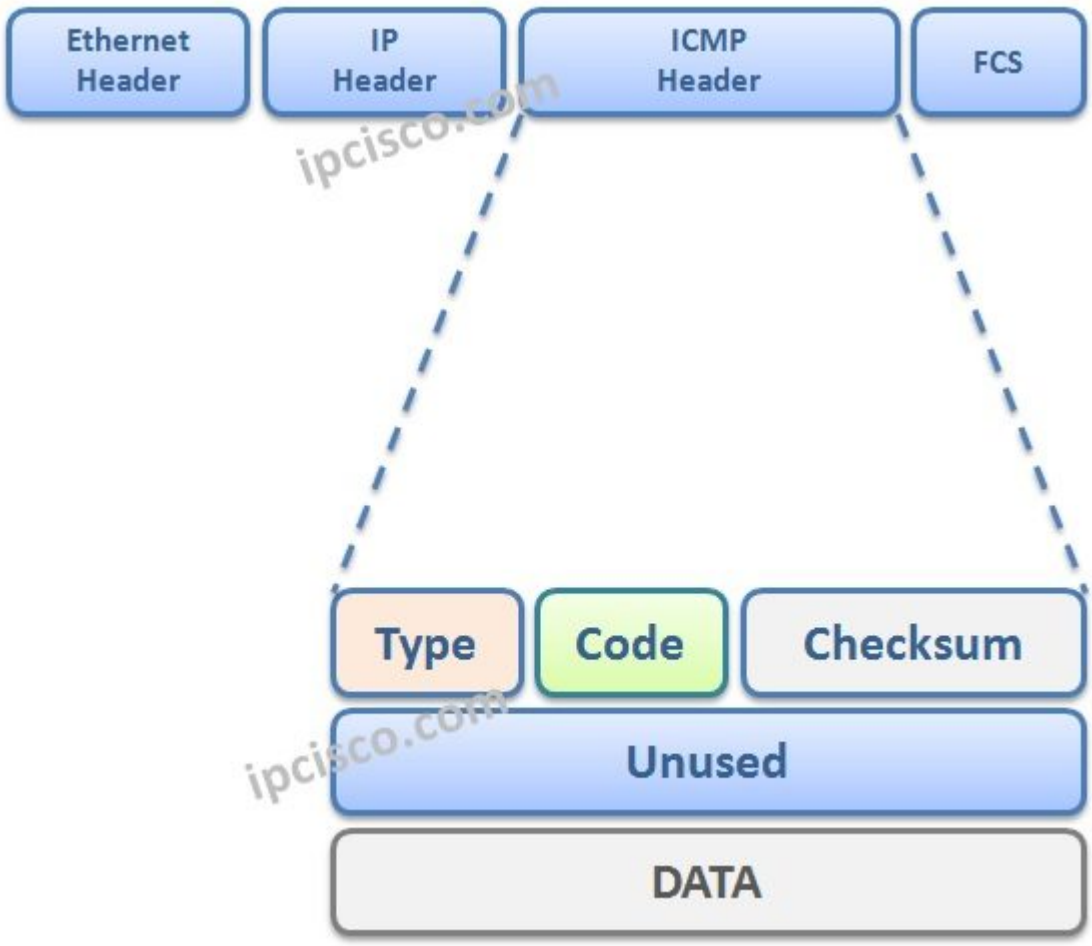
Vorteile

- Mehr Sicherheit - PCs im Netzwerk nicht direkt ansprechbar = mehr Privatsphäre
- Firewall - keine von außen eingehenden Verbindungen erlaubt, so fern keine Verbindung von intern aufgebaut wurde

ICMP - Internet Control Message Protocol (Layer 3)

Das Internet Control Message Protocol (ICMP) ist Bestandteil des Internet Protocols (IP). Es wird aber als eigenständiges Protokoll behandelt, das zur Übermittlung von Meldungen über IP dient. Hauptaufgabe von ICMP ist die **Übertragung von Statusinformationen und Fehlermeldungen der Protokolle IP, TCP und UDP**. Die ICMP-Meldungen werden zwischen Rechnern und aktiven Netzknoten, z. B. Routern, benutzt, um sich gegenseitig Probleme mit Datenpaketen mitzuteilen. Ziel ist, die Übertragungsqualität zu verbessert.

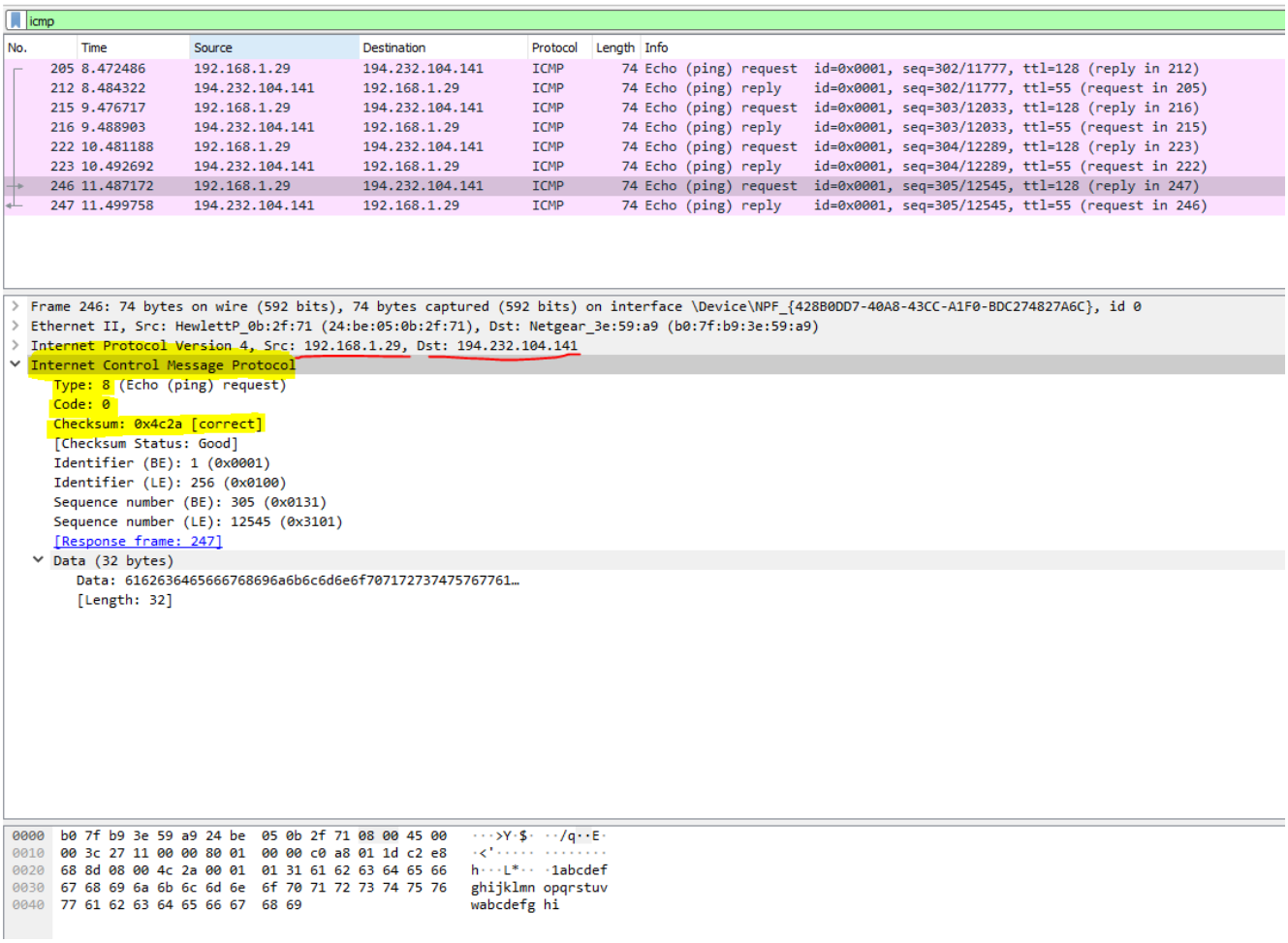
Jedes Betriebssystem mit TCP/IP hat Tools, die ICMP nutzen. Zwei bekannte Tools sind Ping und Trace Route. Beides sind sehr einfache Programme, die zur Analyse von Netzwerk-Problemen gedacht sind und damit wesentlich zur Problemlösung beitragen können.



Typ	Code	Bedeutung
0	0	Ping, Echo Antwort
3	0	Netzwerk nicht erreichbar
3	1	Host nicht erreichbar
3	2	Ziel-Protokoll nicht verfügbar
3	3	Ziel-Port nicht erreichbar
3	6	Netzwerk unbekannt
3	7	Ziel-Host unbekannt
4	0	Überlastkontrolle, ungenutzt
8	0	Ping, Echo-Anfrage
9	0	Routen-Bekanntmachung
10	0	Router-Discovery
11	0	Trace-Route
12	0	Schlechter IP-Header

Aufzeichnung mit Wiresharek

```
C:\Users\Andy>ping www.orf.at

Ping wird ausgeführt für www.orf.at [194.232.104.141] mit 32 Bytes Daten:
Antwort von 194.232.104.141: Bytes=32 Zeit=11ms TTL=55
Antwort von 194.232.104.141: Bytes=32 Zeit=12ms TTL=55
Antwort von 194.232.104.141: Bytes=32 Zeit=11ms TTL=55
Antwort von 194.232.104.141: Bytes=32 Zeit=12ms TTL=55

Ping-Statistik für 194.232.104.141:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
Ca. Zeitangaben in Millisek.:
    Minimum = 11ms, Maximum = 12ms, Mittelwert = 11ms
```



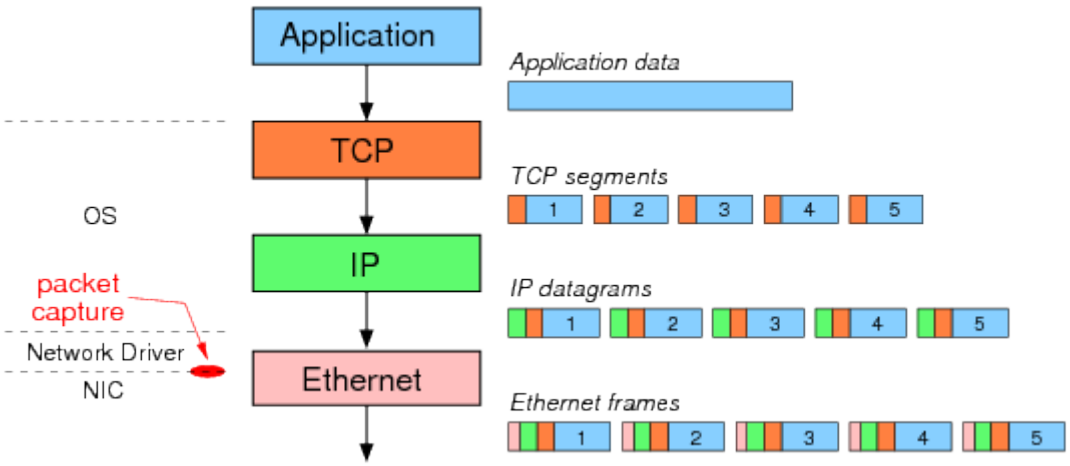
TCP - Transmission Control Protocol (Layer 4)

Das **Transmission Control Protocol** (=Übertragungssteuerungsprotokoll) ist ein Netzwerkprotokoll, das definiert, auf welche Art und Weise Daten zwischen Netzwerkkomponenten ausgetauscht werden sollen. Nahezu sämtliche aktuellen Betriebssysteme moderner Computer beherrschen TCP und nutzen es für den Datenaustausch mit anderen Rechnern. Das Protokoll ist ein zuverlässiges, verbindungsorientiertes, paketvermitteltes Transportprotokoll in Computernetzwerken. Es ist Teil der Internetprotokollfamilie, der Grundlage des Internets.

Aufgaben des TCP-Protokolls

Segmentierung (Data Segmenting)

Eine Funktion von TCP besteht darin, den von den Anwendungen kommenden **Datenstrom in Datenpakete bzw. Segmente aufzuteilen (Segmentierung)** und beim Empfang wieder zusammenzusetzen. Die Segmente werden mit einem **Header** versehen, in dem Steuer- und Kontroll-Informationen enthalten sind. Danach werden die **Segmente an das Internet Protocol (IP) übergeben**. Da beim IP-Routing die **Datenpakete unterschiedliche Wege** gehen können, entstehen unter Umständen **zeitliche Verzögerungen**, die dazu führen, dass die Datenpakete beim Empfänger in einer anderen Reihenfolge eingehen, als sie ursprünglich hatten. Deshalb werden die Segmente beim Empfänger auch wieder in die **richtige Reihenfolge** gebracht und erst dann an die adressierte Anwendung übergeben. Dazu werden die Segmente mit einer **fortlaufenden Sequenznummer** versehen (Sequenzierung).



Verbindungsmanagement (Connection Establishment and Termination)

Als **verbindungsorientiertes Protokoll** ist TCP für den **Verbindungsaufbau und Verbindungsabbau** zwischen zwei Stationen einer **Ende-zu-Ende-Kommunikation** zuständig. Durch TCP stehen Sender und Empfänger ständig in Kontakt zueinander.

Fehlerbehandlung (Error Detection)

Obwohl es sich eher um eine virtuelle Verbindung handelt, werden während der Datenübertragung

ständig **Kontrollmeldungen** ausgetauscht. Der Empfänger bestätigt dem Sender jedes empfangene Datenpaket. Trifft keine Bestätigung beim Absender ein, wird das Paket noch mal verschickt. Da es bei Übertragungsproblemen zu doppelten Datenpaketen und Quittierungen kommen kann, werden alle TCP-Pakete und TCP-Meldungen mit einer **fortlaufenden Sequenznummer** gekennzeichnet. So sind Sender und Empfänger in der Lage, die Reihenfolge und Zuordnung der Datenpakete und Meldungen zu erkennen.

Flusssteuerung (Flow Control)

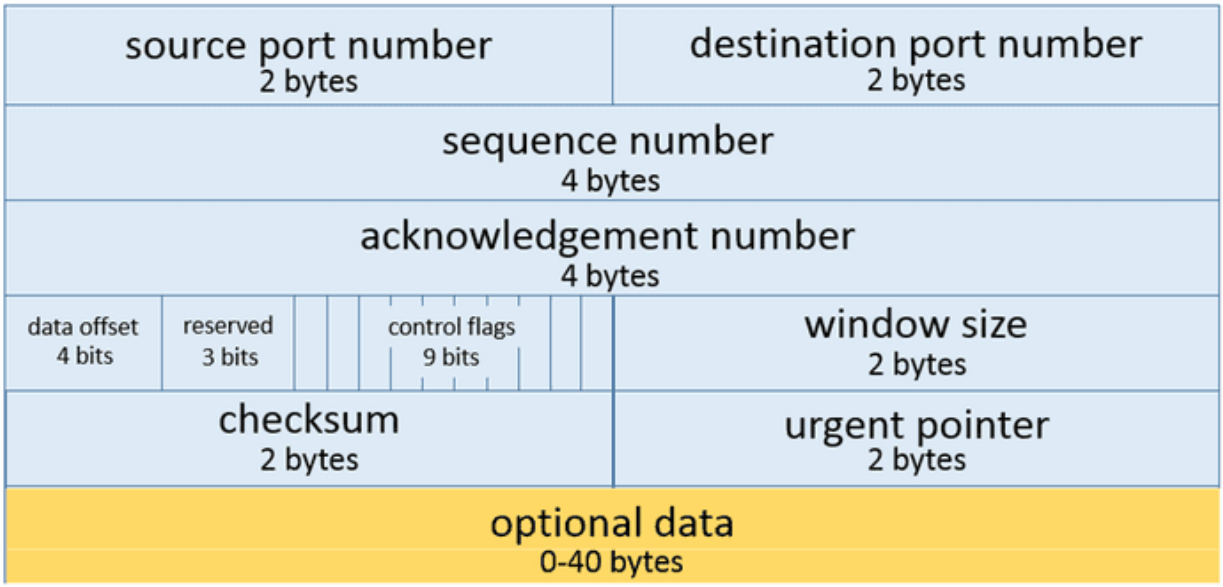
Bei einer paketorientierten Übertragung ohne feste zeitliche Zuordnung und ohne Kenntnis des Übertragungswegs erhält das Transport-Protokoll vom Übertragungssystem **keine Information über die verfügbare Bandbreite**. Mit der **Flusssteuerung** werden beliebig langsame oder schnelle **Übertragungsstrecken dynamisch auszulasten** und auch auf unerwartete Engpässe und Verzögerungen reagiert.

Anwendungsunterstützung (Application Support)

TCP- und UDP-Ports sind eine Software-Abstraktion, um Kommunikationsverbindungen voneinander unterscheiden zu können. Ähnlich wie IP-Adressen Rechner in Netzwerken adressieren, **adressieren Ports spezifische Anwendungen** oder Verbindungen, die auf einem Rechner laufen.

Aufbau eines TCP Headers

Aufbau des TCP-Headers TCP-Pakete setzen sich aus dem Header-Bereich und dem Daten-Bereich zusammen. Im Header sind alle Informationen enthalten, die für eine gesicherte TCP-Verbindung wichtig sind. Der TCP-Header ist in mehrere 32-Bit-Blöcke aufgeteilt. Mindestens enthält der Header 5 solcher Blöcke. Somit hat ein TCP-Header eine Länge von **mindestens 20 Byte**.



TCP Kommunikation

Verbindungsaufbau - 3-Way-Handshake

Der Client, der eine Verbindung aufbauen will, sendet dem Server ein **SYN-Paket** (von englisch synchronize) mit einer Sequenznummer x. Die Sequenznummern sind dabei für die Sicherstellung einer vollständigen Übertragung in der richtigen Reihenfolge und ohne Duplikate wichtig. Es handelt sich also um ein Paket, dessen SYN-Bit im Paketkopf gesetzt ist (siehe TCP-Header). Die Start-Sequenznummer ist eine beliebige Zahl, deren Generierung von der jeweiligen TCP-Implementierung abhängig ist. Sie sollte jedoch möglichst zufällig sein, um Sicherheitsrisiken zu vermeiden.

Der Server (siehe Abbildung) empfängt das Paket. Ist der Port geschlossen, antwortet er mit einem TCP-RST, um zu signalisieren, dass keine Verbindung aufgebaut werden kann. Ist der Port geöffnet, bestätigt er den Erhalt des ersten SYN-Pakets und stimmt dem Verbindungsaufbau zu, indem er ein **SYN/ACK-Paket** zurückschickt (ACK von engl. acknowledgement ‚Bestätigung'). Das gesetzte ACK-Flag im TCP-Header kennzeichnet diese Pakete, welche die Sequenznummer x+1 des SYN-Pakets im Header enthalten. Zusätzlich sendet er im Gegenzug seine Start-Sequenznummer y, die ebenfalls beliebig und unabhängig von der Start-Sequenznummer des Clients ist.

Der Client bestätigt zuletzt den Erhalt des SYN/ACK-Pakets durch das Senden eines eigenen **ACK-Paket**s mit der Sequenznummer x+1. Dieser Vorgang wird auch als **Forward Acknowledgement** bezeichnet. Aus Sicherheitsgründen sendet der Client den Wert y+1 (die Sequenznummer des Servers + 1) im ACK-Segment zurück. Die Verbindung ist damit aufgebaut. Im folgenden Beispiel wird der Vorgang abstrakt dargestellt:

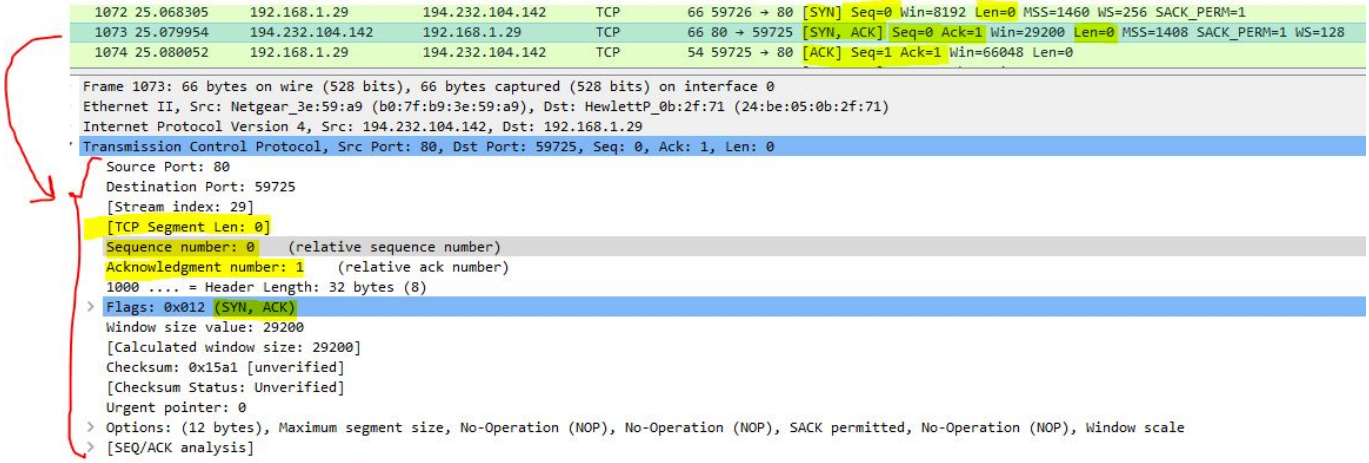
```
1.  SYN-SENT              →    <SEQ=100><CTL=SYN>              →    SYN-RECEIVED
2.  SYN/ACK-RECEIVED      ←    <SEQ=300><ACK=101><CTL=SYN,ACK>    ←
```

```
SYN/ACK-SENT
3.  ACK-SENT            →    <SEQ=101><ACK=301><CTL=ACK>    →    ESTABLISHED
```

Einmal aufgebaut, ist die Verbindung für beide Kommunikationspartner gleichberechtigt, man kann einer bestehenden Verbindung auf TCP-Ebene nicht ansehen, wer der Server und wer der Client ist. Daher hat eine Unterscheidung dieser beiden Rollen in der weiteren Betrachtung keine Bedeutung mehr.

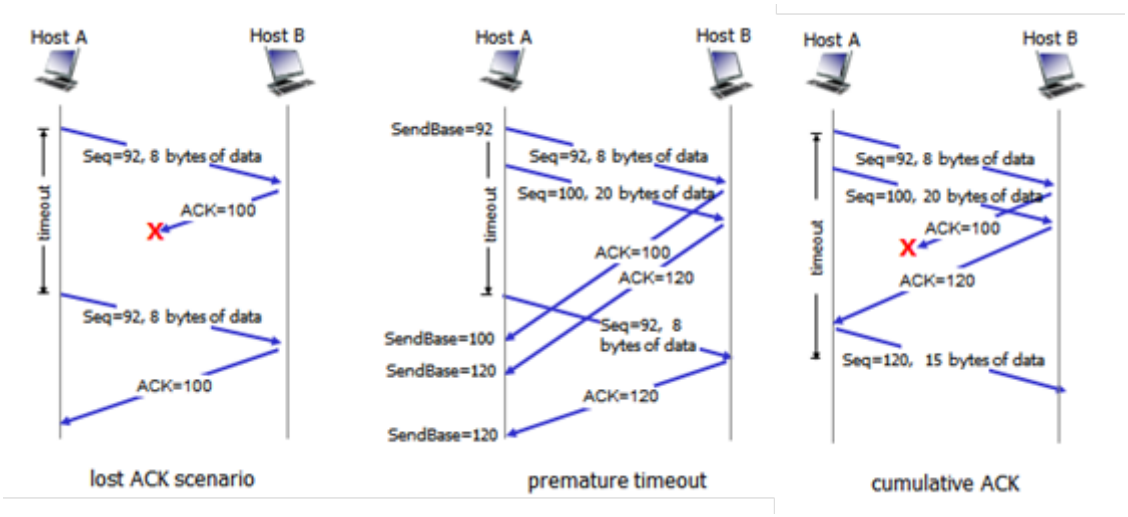
Client-Server-Konzept - inf-schule.de

Aufzeichnung mit Wireshark

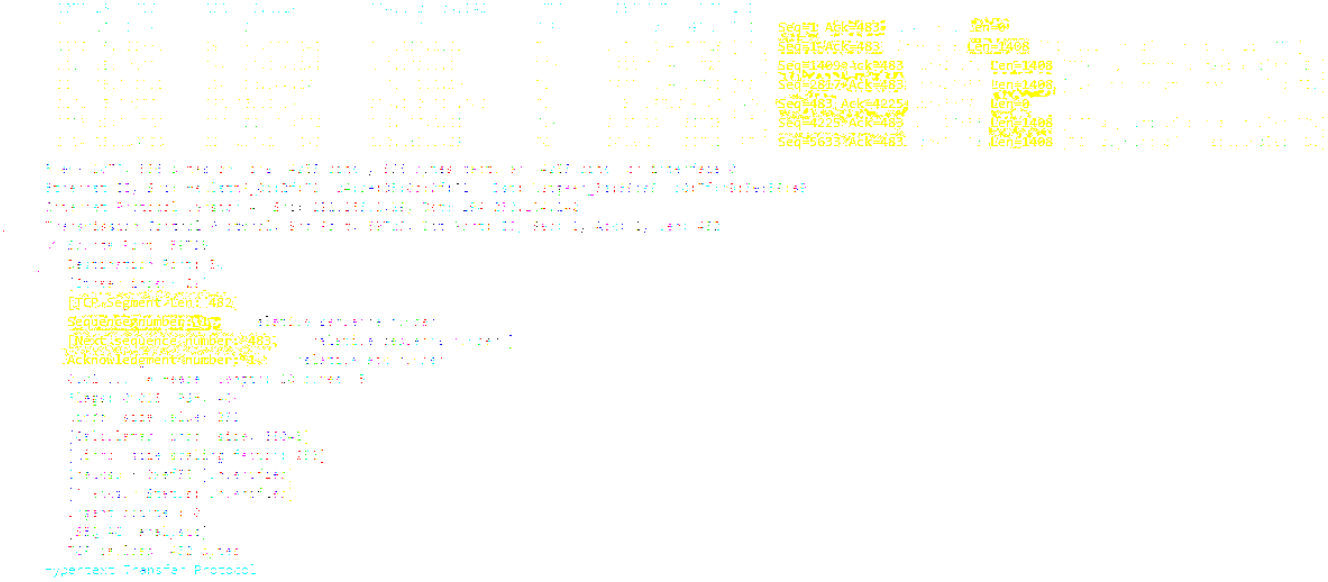


Datenaustausch

Der Sender beginnt mit dem Senden des ersten Datenpakets (Send Paket 1). Der Empfänger nimmt das Paket entgegen (Receive Paket 1) und bestätigt den Empfang (Send ACK Paket 1). Der Sender nimmt die Bestätigung entgegen (Receive ACK Paket 1) und sendet das zweite Datenpaket (Send Paket 2). Der Empfänger nimmt das zweite Paket entgegen (Receive Paket 2) und bestätigt den Empfang (Send ACK Paket 2). Der Sender nimmt die zweite Bestätigung entgegen (Receive ACK Paket 2). Und so läuft der Datenaustausch weiter, bis alle Pakete übertragen wurden.

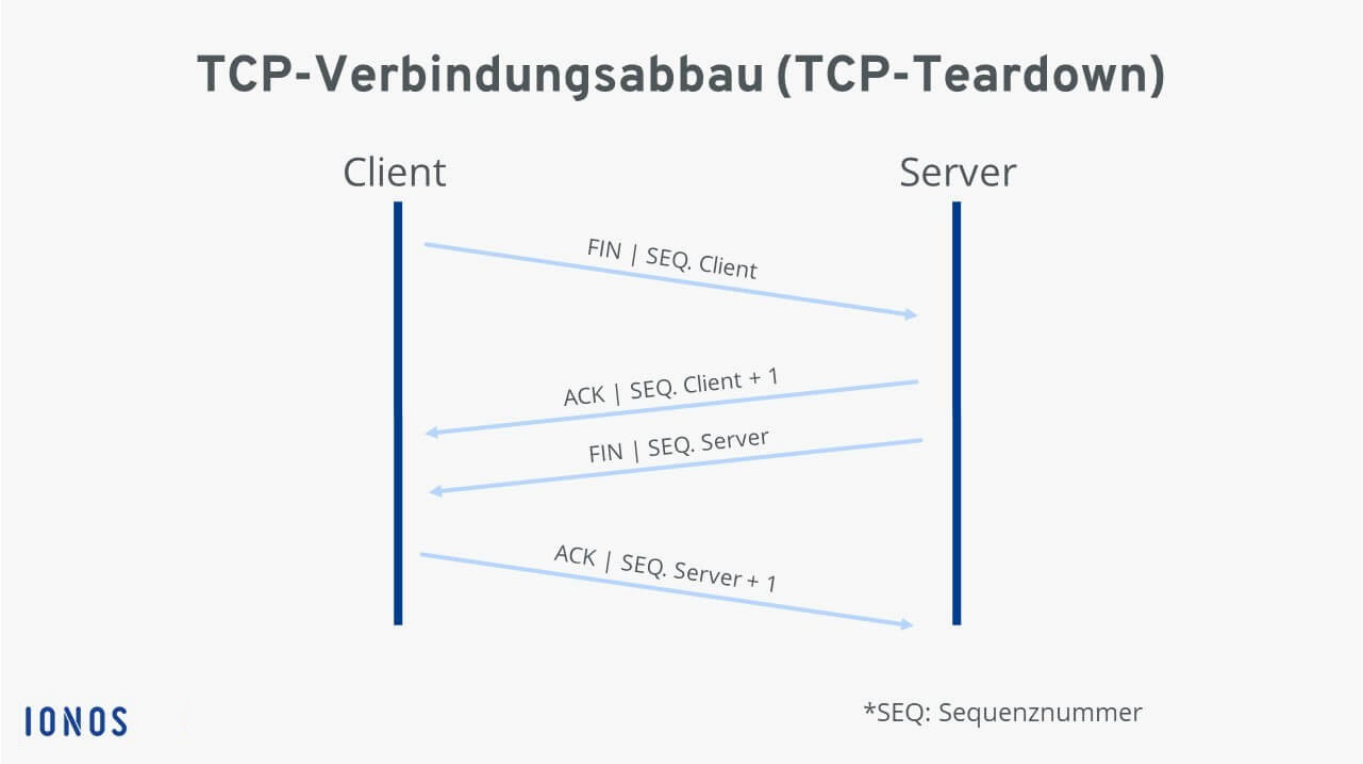


Aufzeichnung mit Wireshark



Verbindungsabbau

Der Verbindungsabbau kann sowohl vom Client als auch vom Server vorgenommen werden. Zuerst schickt einer der beiden der Gegenstelle einen Verbindungsabbauwunsch (**FIN**). Die Gegenstelle bestätigt den Erhalt der Nachricht (ACK) und schickt gleich darauf ebenfalls einen Verbindungsabbauwunsch (**FIN**). Danach bekommt die Gegenstelle noch mitgeteilt, dass die Verbindung abgebaut ist (**ACK**).



Aufzeichnung mit Wireshark

```
353 4.740497    194.232.104.142   192.168.1.29      TCP   60 80 → 61868 [FIN, ACK] Seq=130 Ack=629 Win=30464 Len=0
354 4.740526    192.168.1.29      194.232.104.142   TCP   54 61868 → 80 [ACK] Seq=629 Ack=131 Win=65792 Len=0
355 4.740643    192.168.1.29      194.232.104.142   TCP   54 61868 → 80 [FIN, ACK] Seq=629 Ack=131 Win=65792 Len=0
364 4.751437    194.232.104.142   192.168.1.29      TCP   60 80 → 61868 [ACK] Seq=131 Ack=630 Win=30464 Len=0
```

```
> Frame 355: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
> Ethernet II, Src: HewlettP_0b:2f:71 (24:be:05:0b:2f:71), Dst: Netgear_3e:59:a9 (b0:7f:b9:3e:59:a9)
> Internet Protocol Version 4, Src: 192.168.1.29, Dst: 194.232.104.142
v Transmission Control Protocol, Src Port: 61868, Dst Port: 80, Seq: 629, Ack: 131, Len: 0
    Source Port: 61868
    Destination Port: 80
    [Stream index: 26]
    [TCP Segment Len: 0]
    Sequence number: 629    (relative sequence number)
    Acknowledgment number: 131    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x011 (FIN, ACK)
    Window size value: 257
    [Calculated window size: 65792]
    [Window size scaling factor: 256]
```

UDP - User Datagram Protocol (Layer 4)

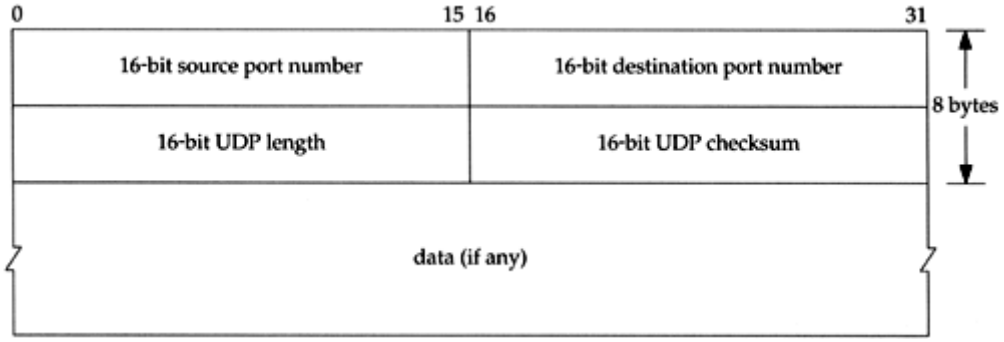
UDP ist ein **verbindungsloses Transport-Protokoll** und arbeitet auf der Schicht 4, der **Transportschicht**, des OSI-Schichtenmodells. Es hat damit eine vergleichbare Aufgabe, wie das verbindungsorientierte TCP. Allerdings arbeitet es **verbindungslos und damit unsicher**. Das bedeutet, der **Absender weiß nicht**, ob seine verschickten **Datenpakete angekommen sind**. Während TCP Bestätigungen beim Datenempfang sendet, verzichtet UDP darauf. Das hat den Vorteil, dass der Paket-Header viel kleiner ist und die Übertragungsstrecke keine Bestätigungen übertragen muss. Typischerweise wird UDP bei DNS-Anfragen, VPN-Verbindungen, Audio- und Video-Streaming verwendet.

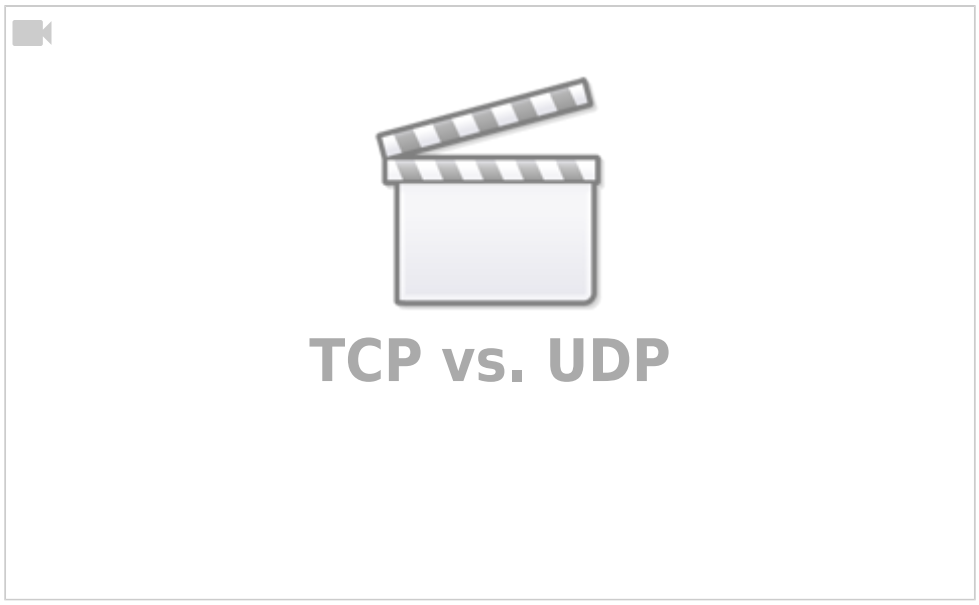
Funktionsweise

UDP hat die selbe Aufgabe wie TCP, nur das nahezu alle Kontrollfunktionen fehlen, dadurch schlanker und einfacher zu verarbeiten ist. So besitzt UDP keinerlei Methoden die sicherstellen, dass ein Datenpaket beim Empfänger ankommt. Ebenso entfällt die Nummerierung der Datenpakete. UDP ist nicht in der Lage die Datenpakete in der richtigen Reihenfolge zusammenzusetzen. Statt dessen werden die UDP-Pakete direkt an die Anwendung weitergeleitet. Für eine sichere Datenübertragung ist deshalb die Anwendung zuständig. In der Regel wird UDP für Anwendungen und Dienste verwendet, die mit Paketverlusten umgehen können oder sich selber um das Verbindungsmanagement kümmern. UDP eignet sich auch für Anwendungen, die nur einzelne, nicht zusammenhängende Datenpakete transportieren müssen.

UDP Header

UDP-Pakete setzen sich aus dem Header-Bereich und dem Daten-Bereich zusammen. Im Header sind alle Informationen enthalten, die eine einigermaßen geordnete Datenübertragung zulässt und die ein UDP-Paket als ein solches erkennen lassen. Der UDP-Header ist in 32-Bit-Blöcke unterteilt. Er besteht aus zwei solcher Blöcke, die den Quell- und Ziel-Port, die Länge des gesamten UDP-Pakets und die Check-Summe enthalten. Der UDP-Header ist mit insgesamt 8 Byte sehr schlank und lässt sich mit wenig Rechenleistung verarbeiten.





DHCP - Dynamic Host Configuration Protocol (UDP, Ports 67 + 68, Layer 7)

DHCP ist ein Protokoll, um **IP-Adressen** in einem TCP/IP-Netzwerk **zu verwalten** und an die anfragenden Hosts zu verteilen. Mit DHCP ist jeder **Netzwerk-Teilnehmer** in der Lage sich selber **automatisch zu konfigurieren**.

Warum DHCP?

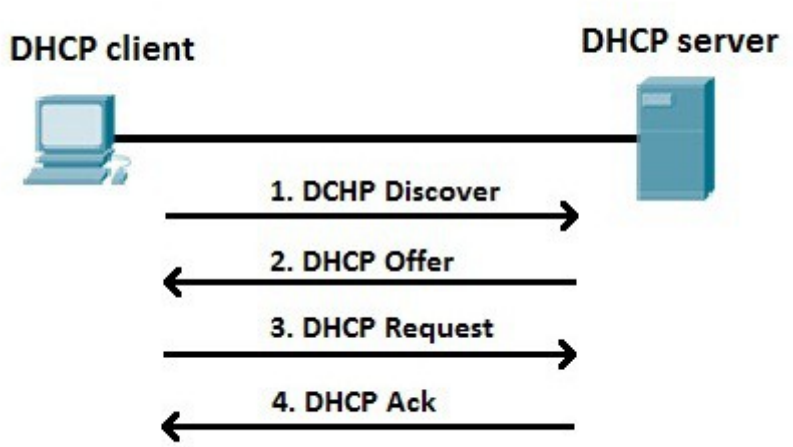
Um ein Netzwerk per TCP/IP aufzubauen ist es notwendig an jedem Host eine IP-Konfiguration vorzunehmen. Für ein TCP/IP-Netzwerk müssen folgende Einstellungen an jedem Host vorgenommen werden:

- Vergabe einer eindeutigen IP-Adresse
- Zuweisen einer Subnetzmaske (Subnetmask)
- Zuweisen des zuständigen Default- bzw. Standard-Gateways
- Zuweisen des zuständigen DNS-Servers

In den ersten IP-Netzen wurden IP-Adressen noch von Hand **aufwendig vergeben** und **fest** in die Systeme **eingetragen**. Die dazu **erforderliche Dokumentation** war jedoch nicht immer fehlerfrei und schon gar nicht aktuell und vollständig. Der Ruf nach einer einfachen und automatischen Adressverwaltung wurde deshalb besonders bei Betreibern großer Netze laut. Hier war durch die manuelle Verwaltung und Konfiguration sehr **viel Planungs- und Arbeitszeit** notwendig. Um für die Betreiber der immer größer werdenden Netze eine Erleichterung zu verschaffen wurde DHCP entwickelt. Mit DHCP kann jede IP-Host die IP-Adresskonfiguration von einem DHCP-Server anfordern und sich selber automatisch konfigurieren. So müssen IP-Adressen nicht mehr manuell verwaltet und zugewiesen werden.

Funktionsweise

1. Der DHCP-Client **schickt an alle Rechner** im Netzwerk (=Broadcast) eine **DHCP-Server-Suchanfrage (DHCP-Discover)**. Bis auf den DHCP-Server verwerfen alle Rechner das Datenpaket.
2. Nur der **DHCP-Server** empfängt den DHCP-Discover und bietet dem Client mittels Broadcast eine **freie IP-Adresse an (DHCP-Offer)**
3. Nur der **DHCP-Client** empfängt den DHCP-Offer und antwortet mit einer eine **DHCP-Anfrage (DHCP-Request)**. Alle anderen Clients verwerfen wiederum das DHCP-Angebot.
4. Der **DHCP-Server** empfängt den DHCP-Request und **bestätigt** dem DHCP-Client die angefragte IP-Adresse **mittels DHCP-ACK**.



DNS - Domain Name System (UDP, Port 53 , Layer 7)

Das Domain Name System, kurz DNS, wird auch als **Telefonbuch des Internets** bezeichnet. Ähnlich wie man in einem Telefonverzeichnis nach einem Namen sucht, um die Telefonnummer heraus zu bekommen, schaut man im DNS nach einem Computernamen, um die dazugehörige IP-Adresse zu bekommen. Die IP-Adresse wird benötigt, um eine Verbindung zu einem Server aufbauen zu können, über den nur der Computername bekannt ist.

Das Domain Name System ist ein System zur Auflösung von Computernamen in IP-Adressen und umgekehrt. DNS kennt keine zentrale Datenbank. Die Informationen sind auf vielen tausend Nameservern (DNS-Server) verteilt. Möchte man zum Beispiel die Webseite www.orf.at besuchen, dann fragt der Browser einen DNS-Server, der in der IP-Konfiguration hinterlegt ist. Das ist in der Regel der Router des Internet-Zugangs. Je nach dem, ob die DNS-Anfrage beantwortet werden kann oder nicht, wird eine Kette weiterer DNS-Server befragt, bis die Anfrage positiv beantwortet und eine IP-Adresse an den Browser zurück geliefert werden kann.

Wenn ein Computername oder Domain-Name nicht aufgelöst werden kann, dann kann auch keine Verbindung zu dem betreffenden Host aufgebaut werden. Es sei denn, der Nutzer verfügt über das Wissen der IP-Adresse. Das bedeutet, ohne DNS ist die Kommunikation im Netzwerk und im Internet praktisch nicht möglich. Deshalb existieren viele tausend DNS-Server auf der ganzen Welt, die zusätzlich hierarchisch angeordnet sind und sich gegenseitig über Änderungen informieren.

Top-Level-Domain & Second-Level-Domain

Domain-Namen sind hierarchisch von rechts nach links gegliedert. Der ganz rechte Abschnitt nach dem letzten Punkt heißt Top-Level-Domain (TLD), der davor Second-Level-Domain (SLD) oder einfach „Domain“. Alle weiteren Namensteile links davon sind jeweils Sub- bzw. Third-Level-Domains (Fourth Level, Fifth Level, Sixth Level usw.). Ein Beispiel verdeutlicht die Begrifflichkeiten: Der Name „www.example.com“ besteht aus drei Ebenen:

- „.com“: die erste Ebene, auch Top-Level-Domain oder Domain-Endung genannt
- „example“: die zweite Ebene, auch als Second-Level-Domain oder Domain bezeichnet
- „www“: die dritte Ebene, auch Sub- oder Third-Level-Domain genannt

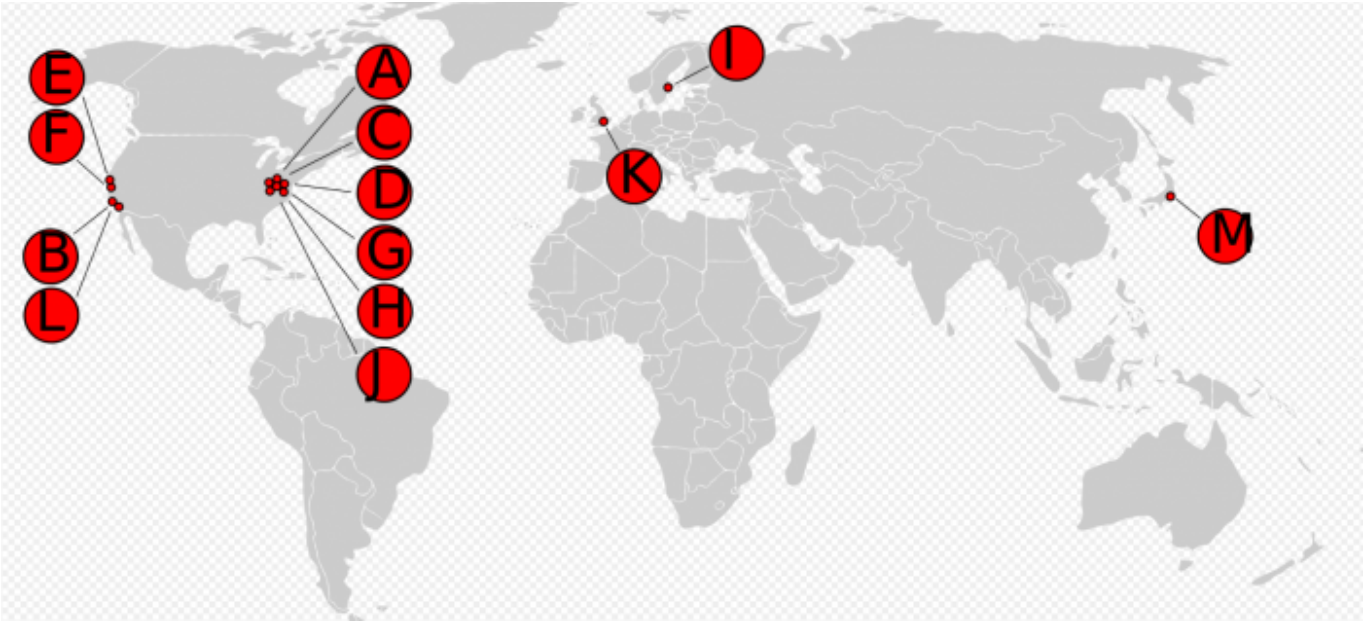
ROOT DNS Server

Root-Nameserver, kurz Root-Server, sind Server zur **Namensauflösung an der Wurzel (Root)** des Domain Name Systems im Internet. Die Zone der Root-Server umfasst Namen und **IP-Adressen aller Nameserver aller Top-Level-Domains (TLD)**.

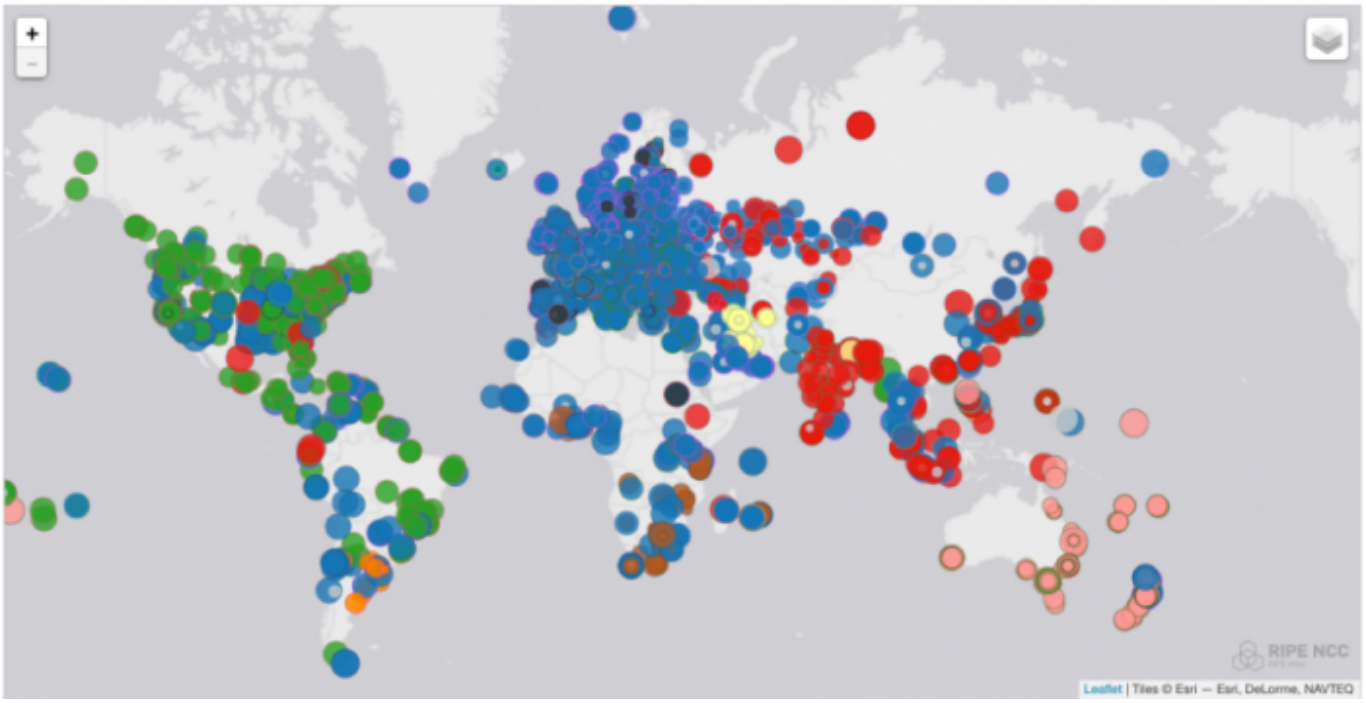
Es gibt 13 Root-Nameserver (A bis M) , die fortlaufend nach dem Schema <Buchstabe>.root-servers.net benannt sind. Jeder Root-Nameserver ist unter einer IPv4-Adresse und einer IPv6-Adresse erreichbar.

List of Root Servers

HOSTNAME	IP ADDRESSES	MANAGER
a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	199.9.14.201, 2001:500:200::b	University of Southern California (ISI)
c.root-servers.net	192.33.4.12, 2001:500:2::c	Cogent Communications
d.root-servers.net	199.7.91.13, 2001:500:2d::d	University of Maryland
e.root-servers.net	192.203.230.10, 2001:500:a8::e	NASA (Ames Research Center)
f.root-servers.net	192.5.5.241, 2001:500:2f::f	Internet Systems Consortium, Inc.
g.root-servers.net	192.112.36.4, 2001:500:12::d0d	US Department of Defense (NIC)
h.root-servers.net	198.97.190.53, 2001:500:1::53	US Army (Research Lab)
i.root-servers.net	192.36.148.17, 2001:7fe::53	Netnod
j.root-servers.net	192.58.128.30, 2001:503:c27::2:30	VeriSign, Inc.
k.root-servers.net	193.0.14.129, 2001:7fd::1	RIPE NCC
l.root-servers.net	199.7.83.42, 2001:500:9f::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project



Alle Root-Nameserver setzen Anycast zur Lastverteilung ein, sodass die 13 Adressen von tatsächlich mehreren hundert Servern an verschiedenen Orten der Welt bedient werden.



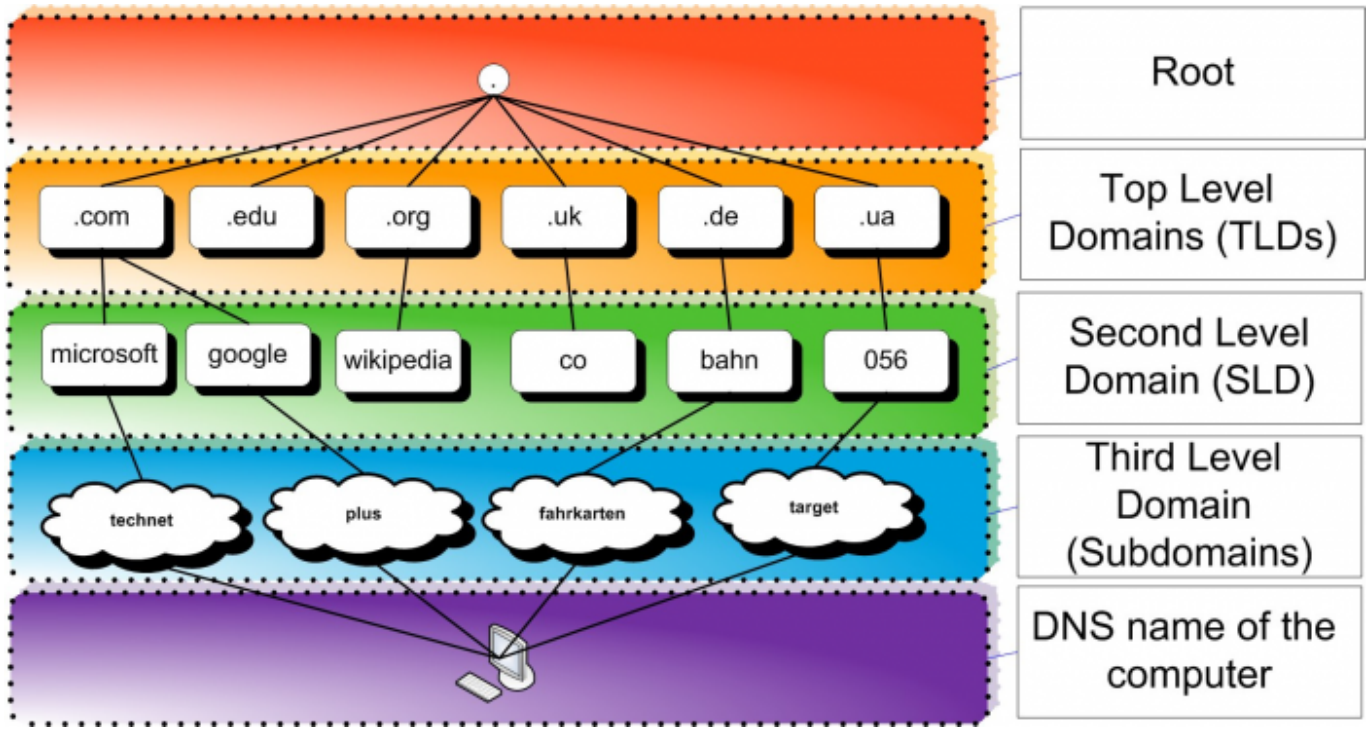
FQDN (Fully Qualified Domain Name)

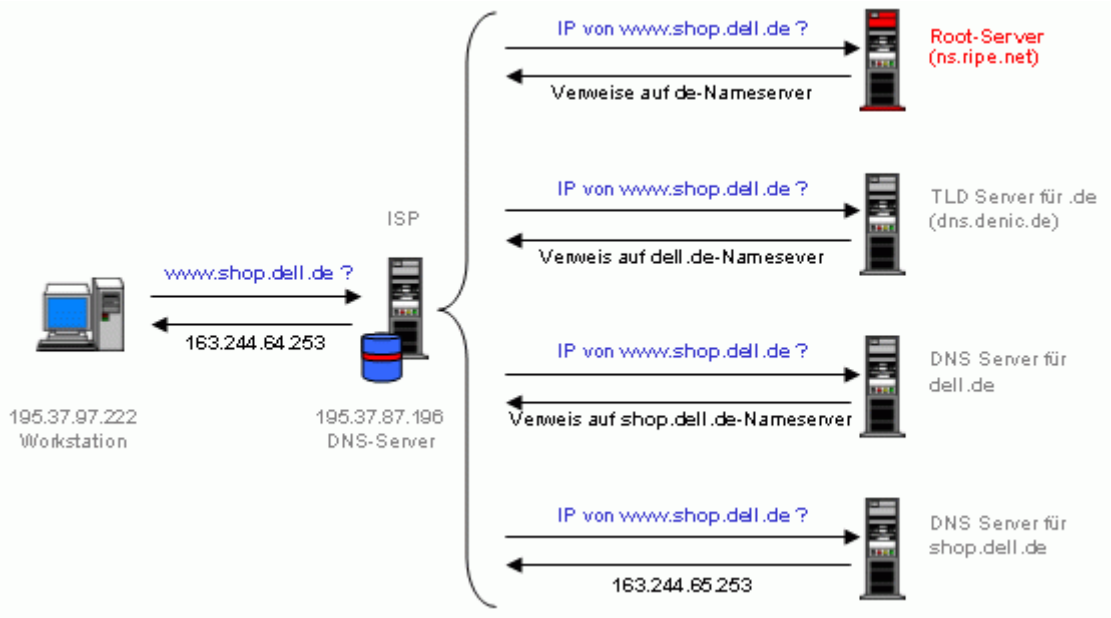
Der vollständige Name einer Domain wird als ihr Fully Qualified Domain Name (FQDN) bezeichnet. Der Domain-Name ist in diesem Fall eine absolute Adresse.

Der FQDN www.example.com. ergibt sich durch:

```
3rd-level-label. 2nd-level-label. Top-Level-Domain. root-label
```

Da das Root-Label immer leer ist (es besteht aus einer leeren Zeichenkette), wird bei den meisten Benutzer-Anwendungen (zum Beispiel Browsern) in der Regel auf die Eingabe des Punktes zwischen dem Label der Top Level Domain und dem root-label verzichtet. Streng genommen handelt es sich bei dieser Schreibweise nicht mehr um eine absolute, sondern um eine relative Adresse und damit nicht mehr um einen FQDN.


Root
Top Level Domains (TLDs)
Second Level Domain (SLD)
Third Level Domain (Subdomains)
DNS name of the computer
.com
.edu
.org
.uk
.de
.ua
microsoft
google
wikipedia
co
bahn
056
technet
plus
fahrkarten
target

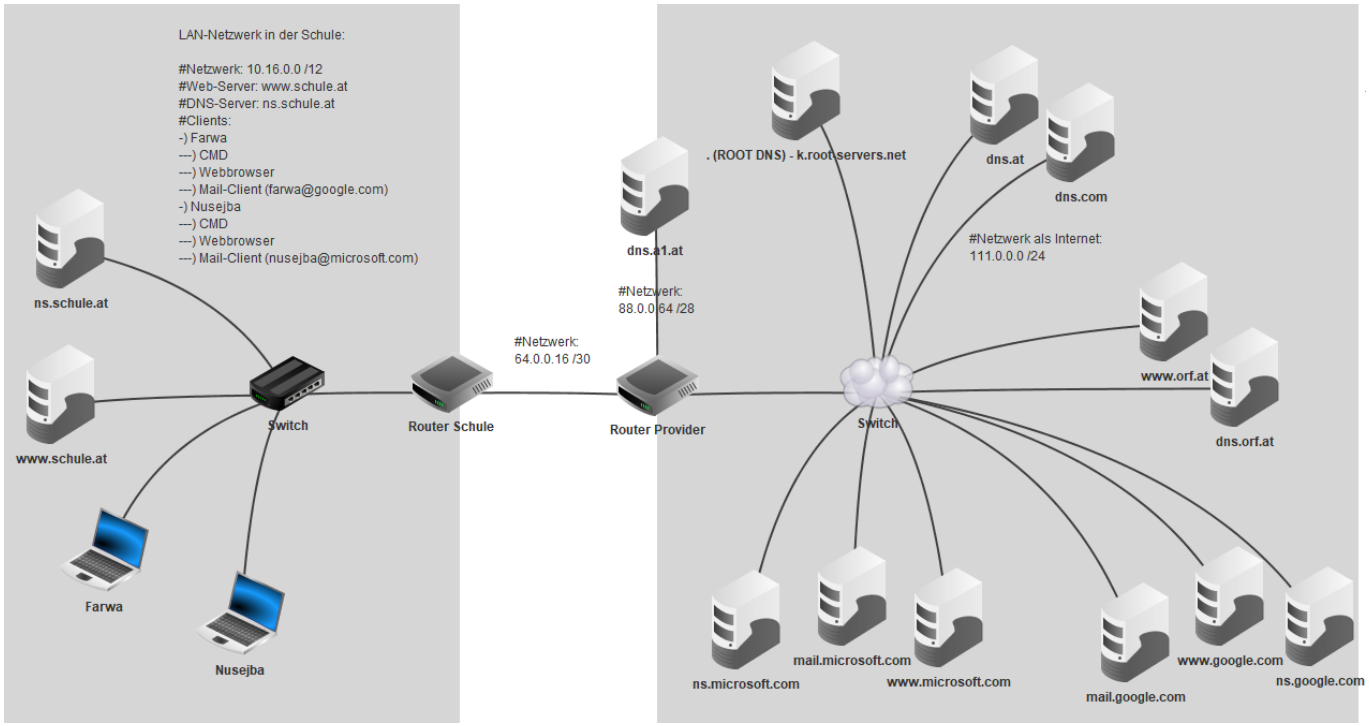

IP von www.shop.dell.de ?
Verweise auf de-Nameserver
Root-Server (ns.ripe.net)
IP von www.shop.dell.de ?
Verweis auf dell.de-Nameserver
TLD Server für .de (dns.denic.de)
ISP
www.shop.dell.de ?
163.244.64.253
IP von www.shop.dell.de ?
Verweis auf shop.dell.de-Nameserver
DNS Server für dell.de
IP von www.shop.dell.de ?
163.244.65.253
DNS Server für shop.dell.de
195.37.97.222
Workstation
195.37.87.196
DNS-Server

Video

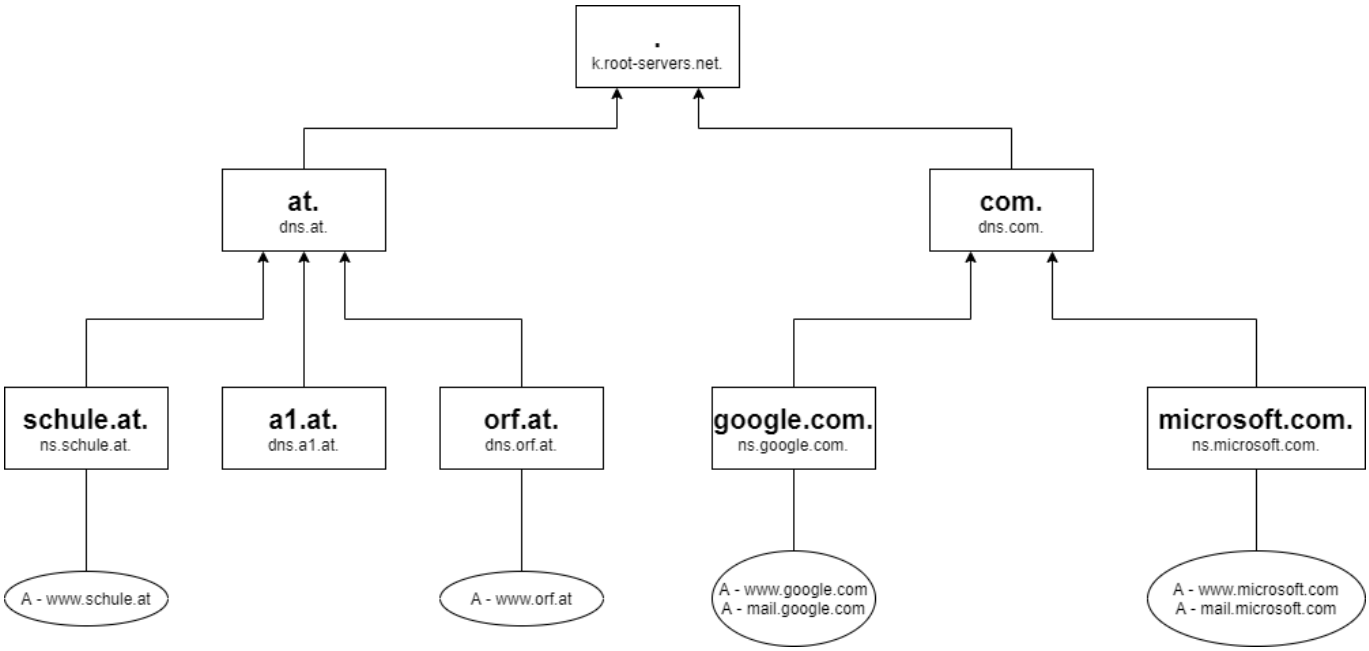
Beispiel zur rekursiven Namensauflösung in Filius

FILIUS-BEISPIEL zur rekursiven Namensauflösung

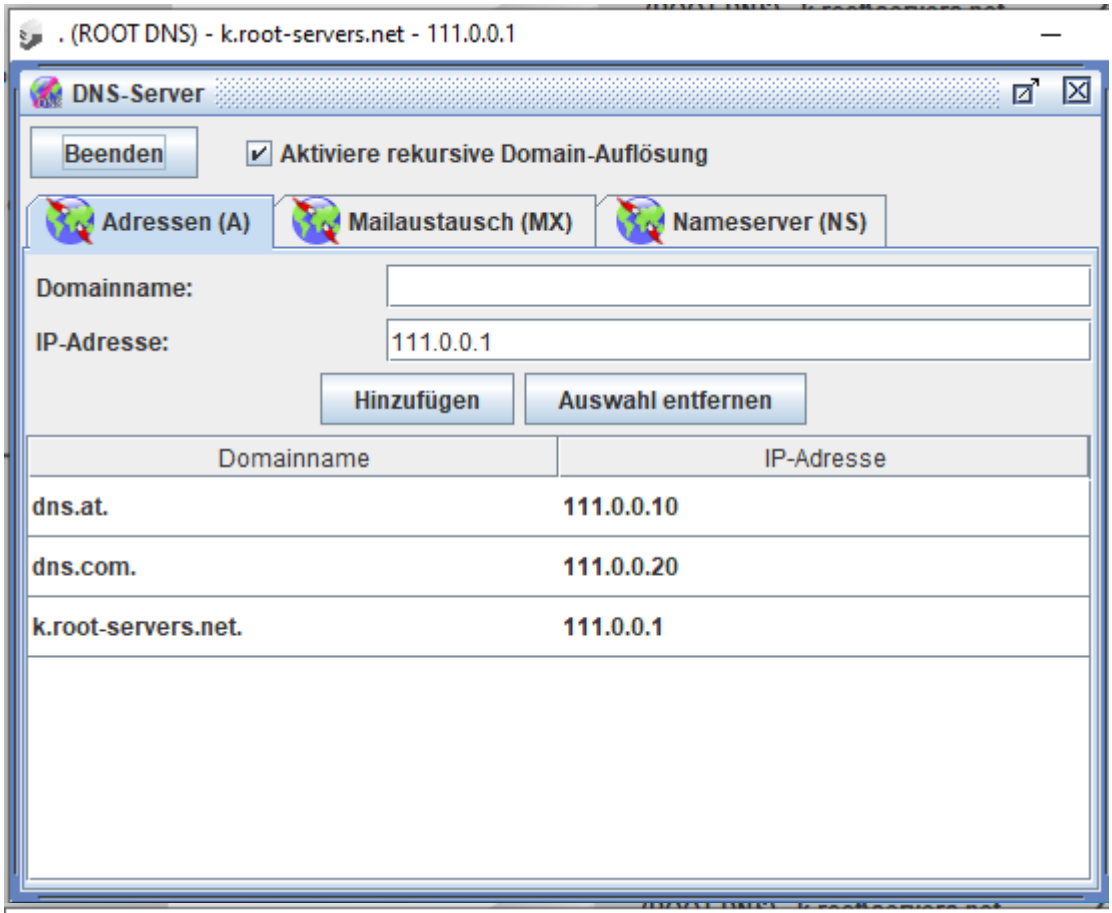
NW-Übersicht

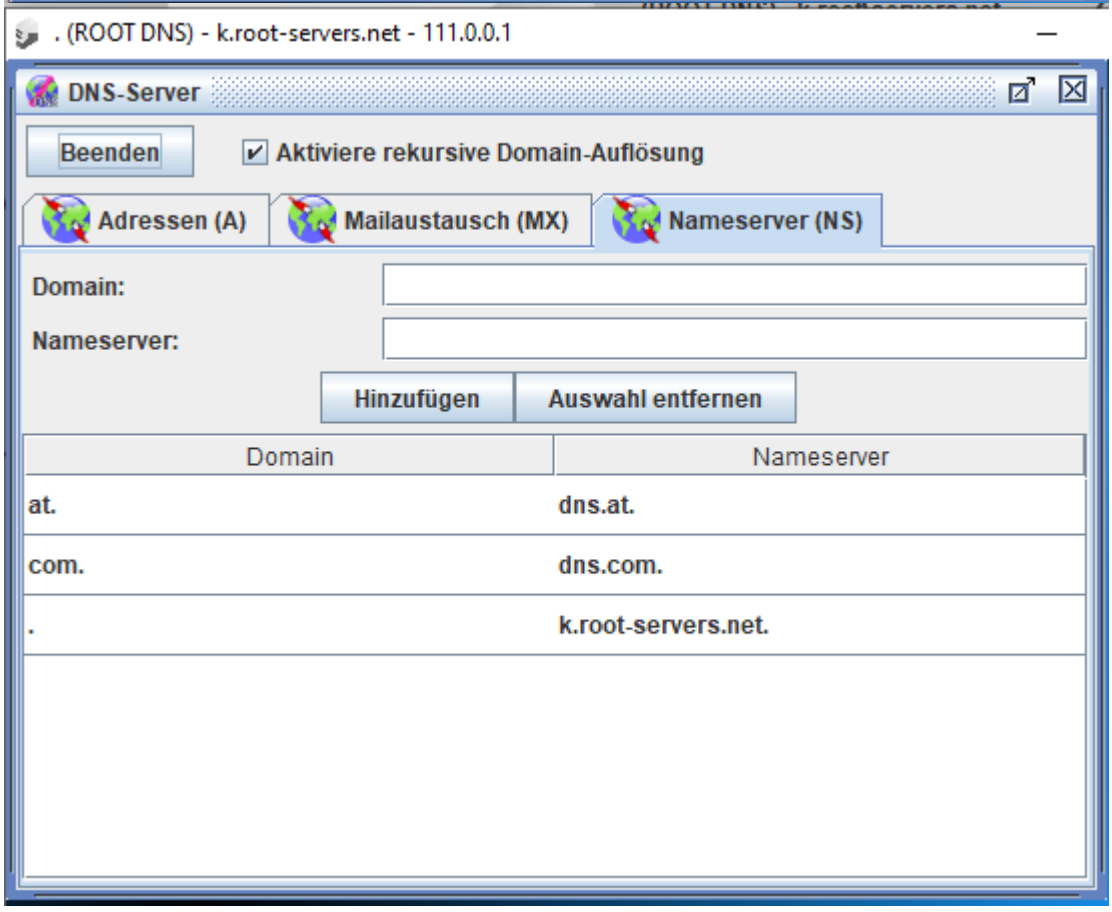


DNS-Hierarchie

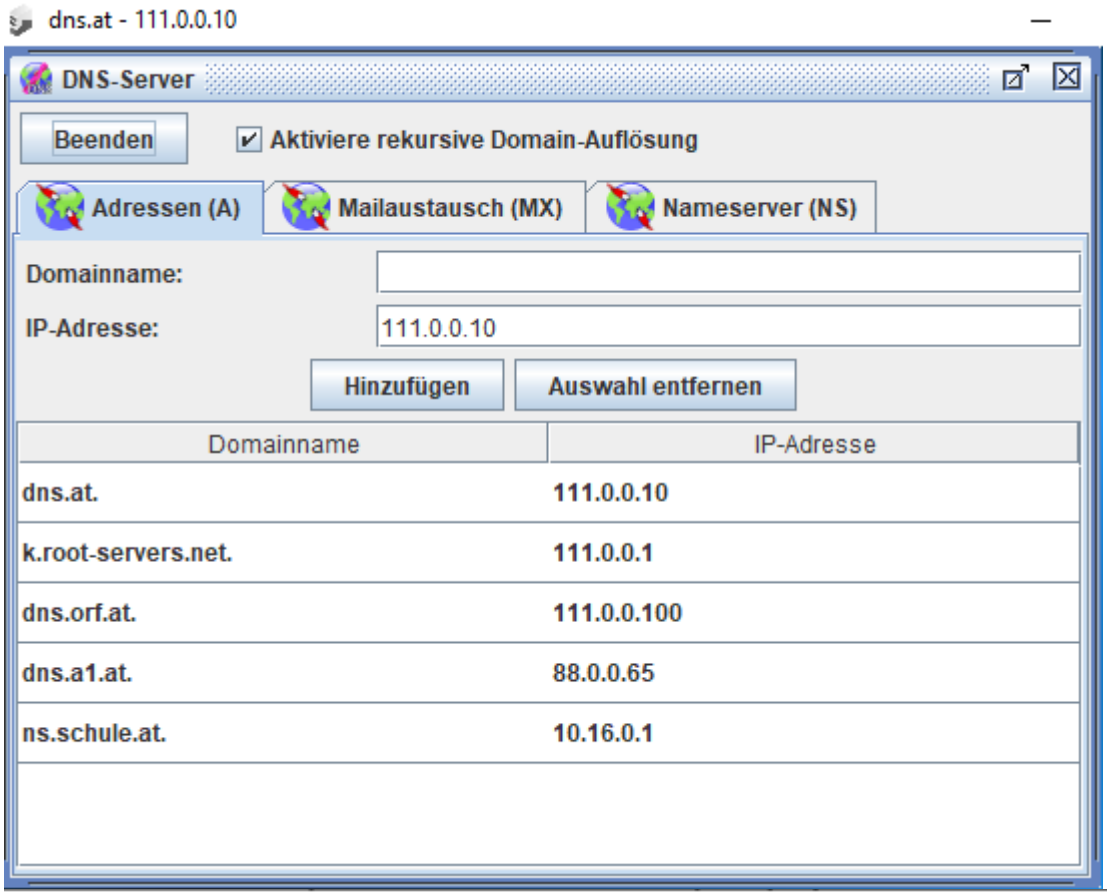


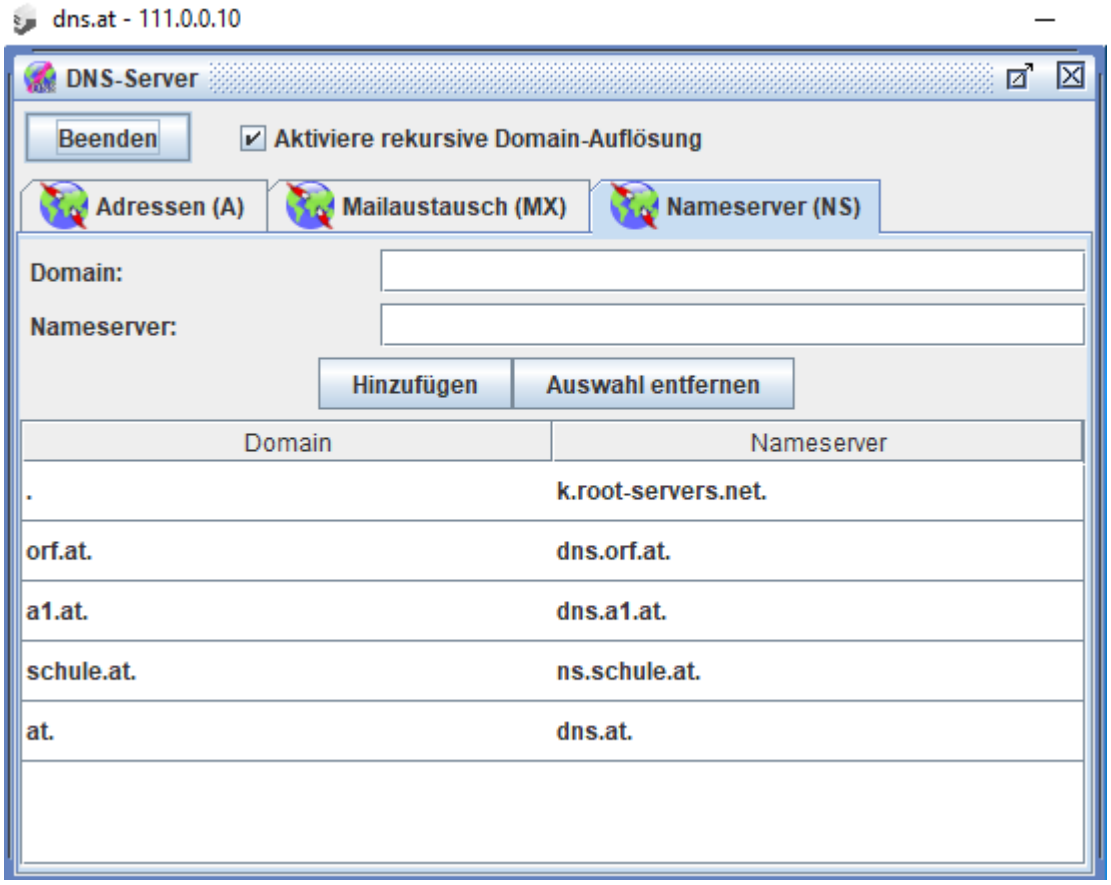
. (ROOT) - k.root-servers.net



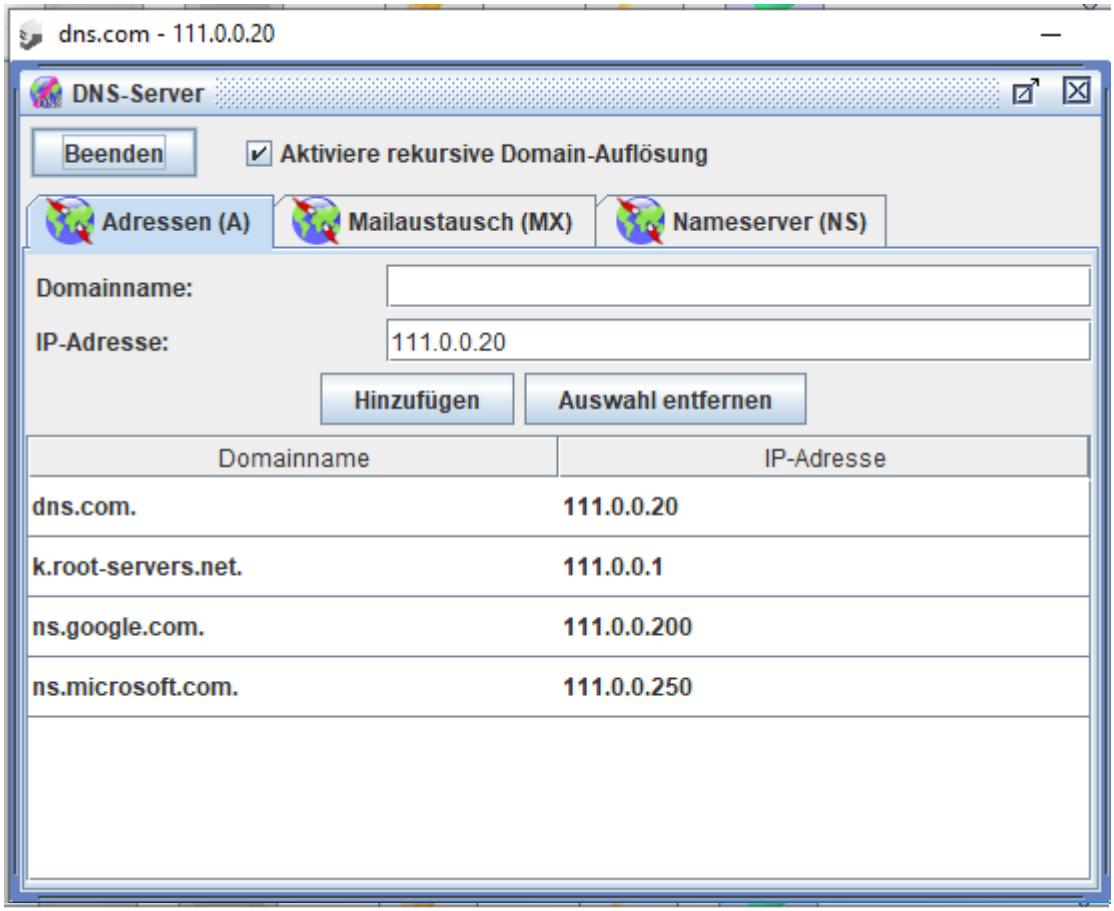


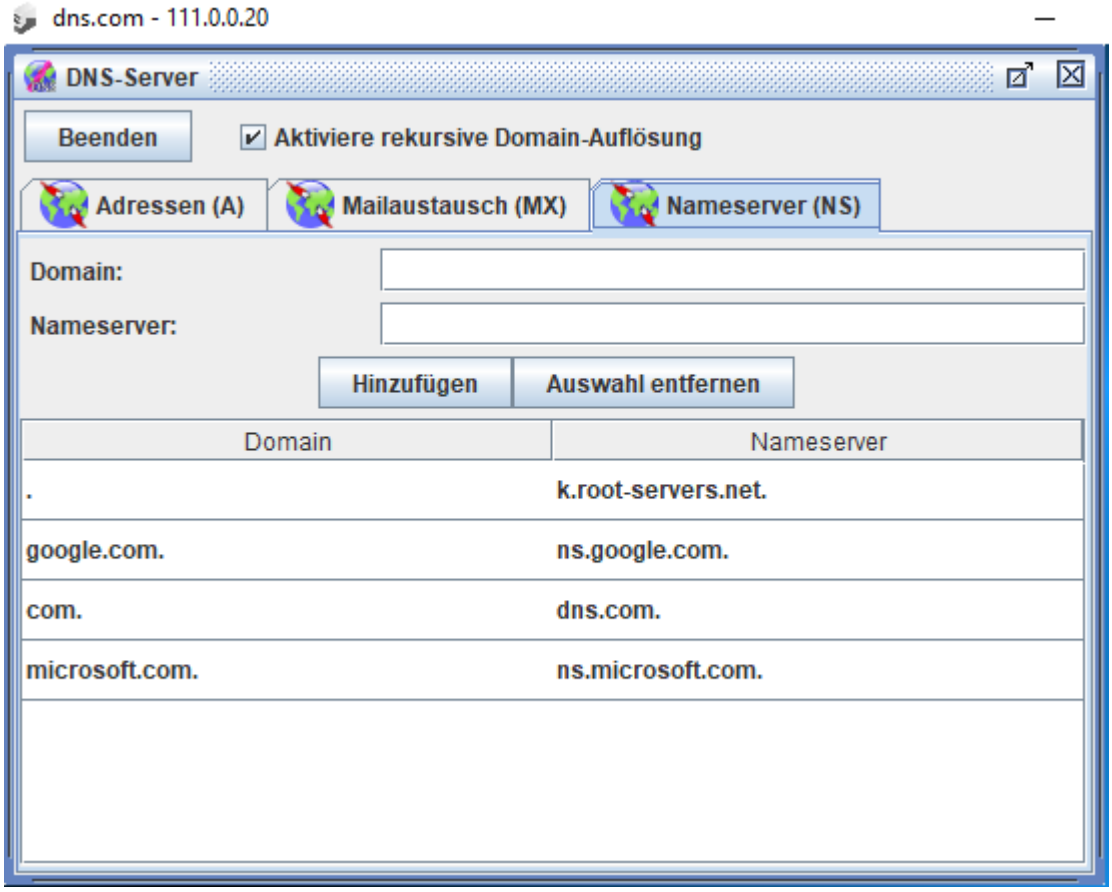
dns.at


dns.at - 111.0.0.10
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domainname:
IP-Adresse:
111.0.0.10
Hinzufügen
Auswahl entfernen
Domainname
IP-Adresse
dns.at. 111.0.0.10
k.root-servers.net. 111.0.0.1
dns.orf.at. 111.0.0.100
dns.a1.at. 88.0.0.65
ns.schule.at. 10.16.0.1

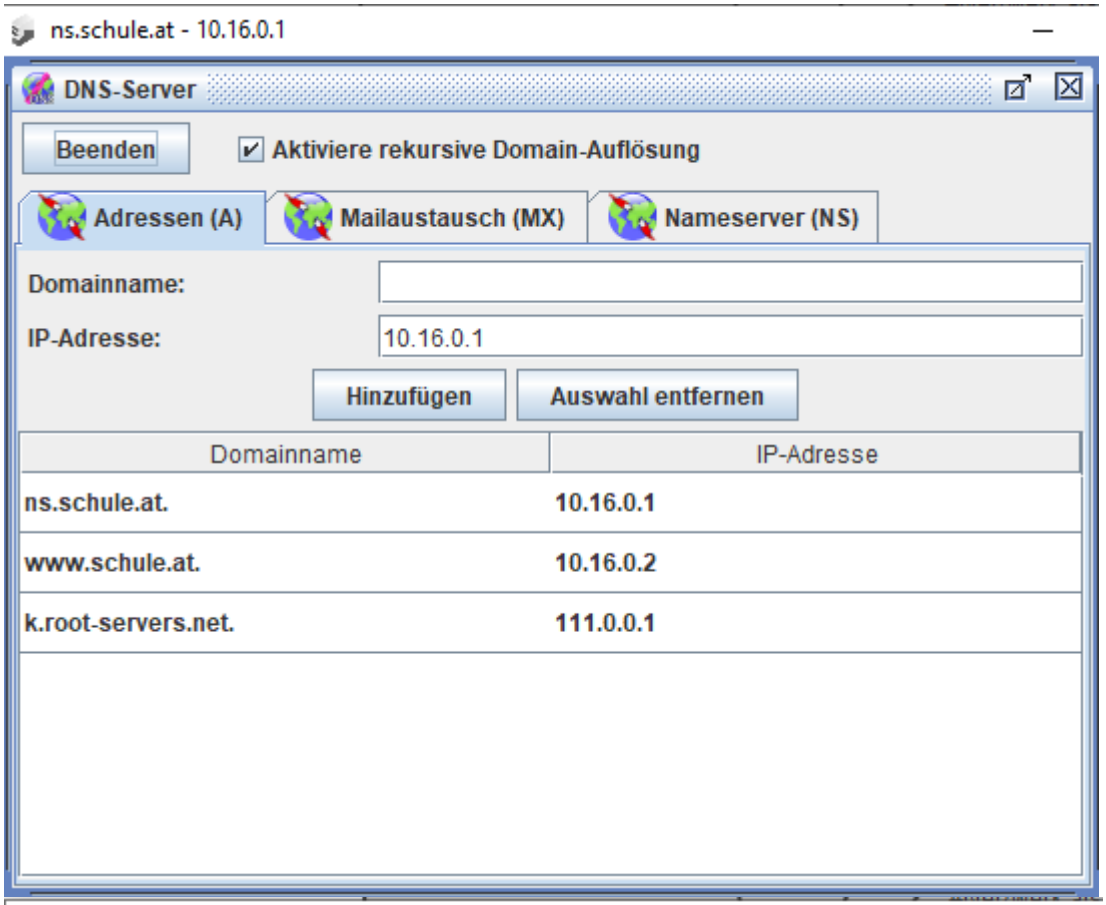

dns.at - 111.0.0.10
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domain:
Nameserver:
Hinzufügen
Auswahl entfernen
Domain
Nameserver
. k.root-servers.net.
orf.at. dns.orf.at.
a1.at. dns.a1.at.
schule.at. ns.schule.at.
at. dns.at.

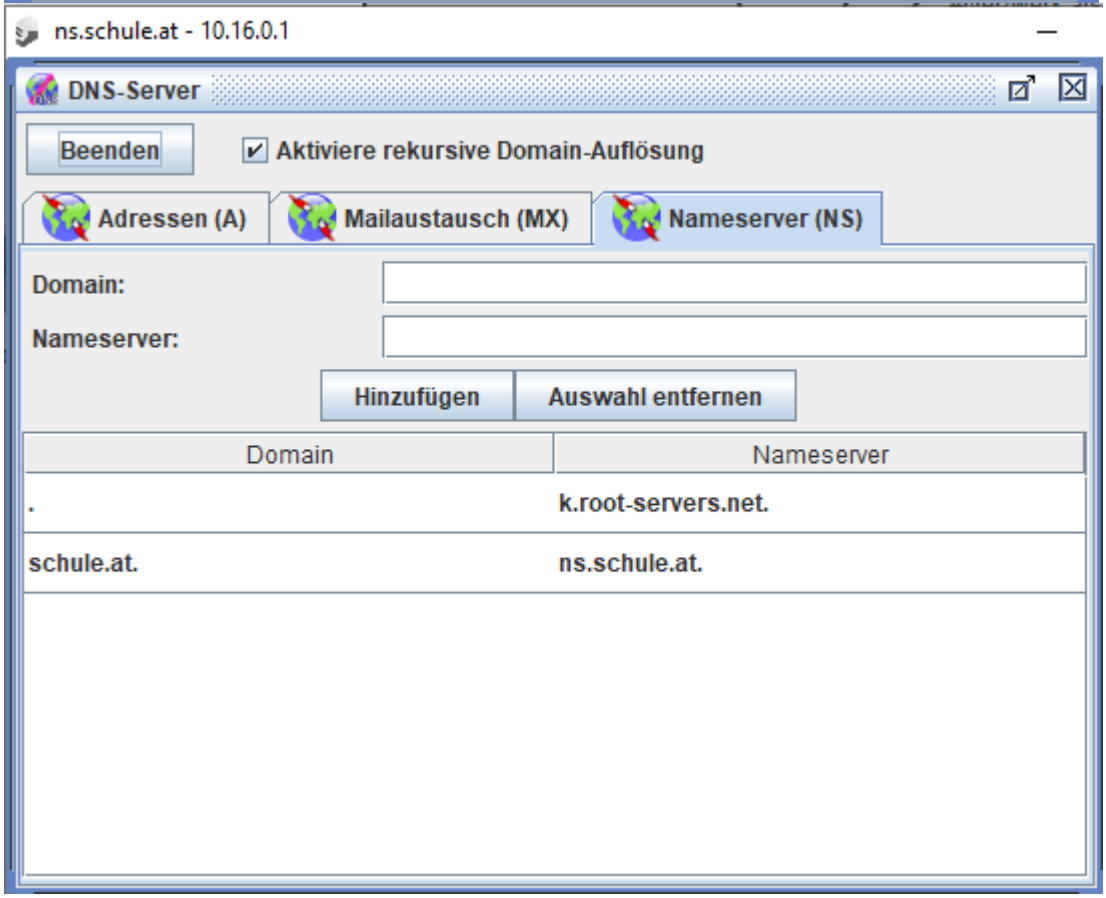
dns.com


dns.com - 111.0.0.20
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domainname:
IP-Adresse:
111.0.0.20
Hinzufügen
Auswahl entfernen
Domainname
IP-Adresse
dns.com.
111.0.0.20
k.root-servers.net.
111.0.0.1
ns.google.com.
111.0.0.200
ns.microsoft.com.
111.0.0.250

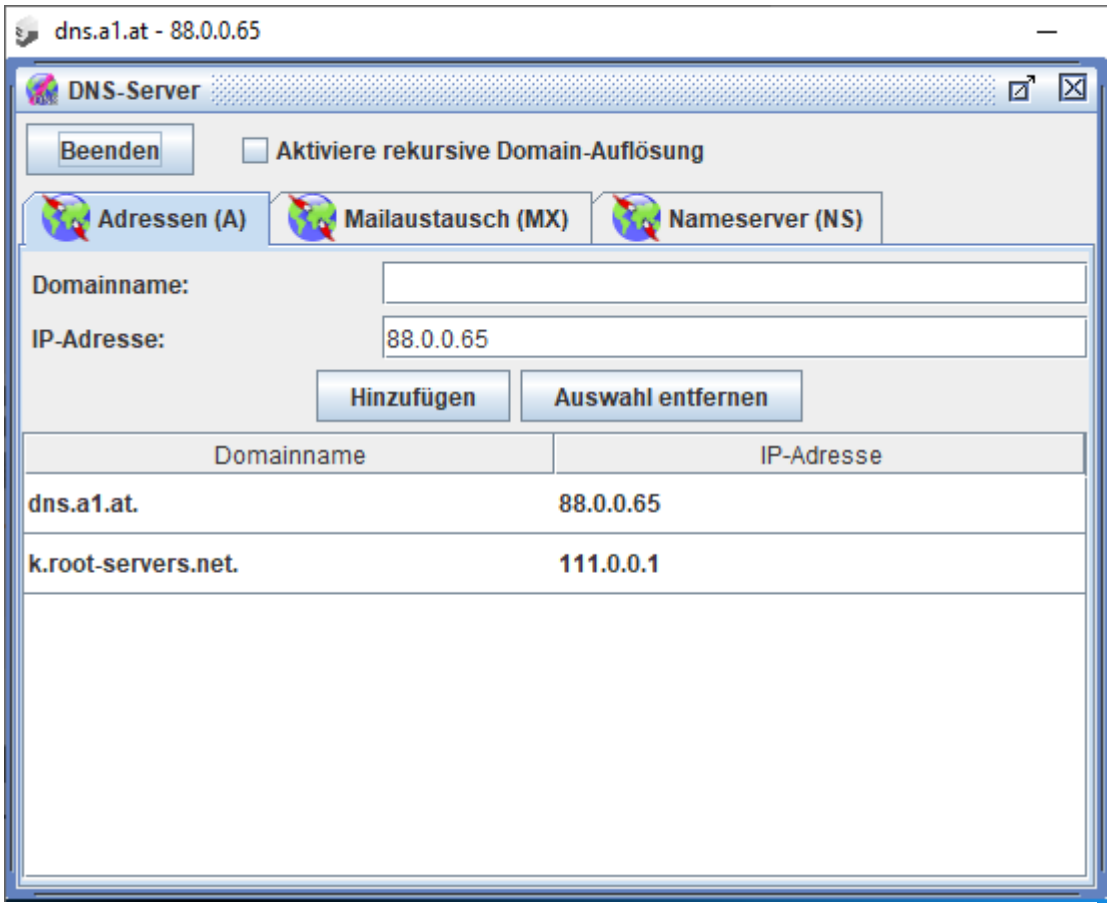

dns.com - 111.0.0.20
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domain:
Nameserver:
Hinzufügen
Auswahl entfernen
Domain
Nameserver
.
k.root-servers.net.
google.com.
ns.google.com.
com.
dns.com.
microsoft.com.
ns.microsoft.com.

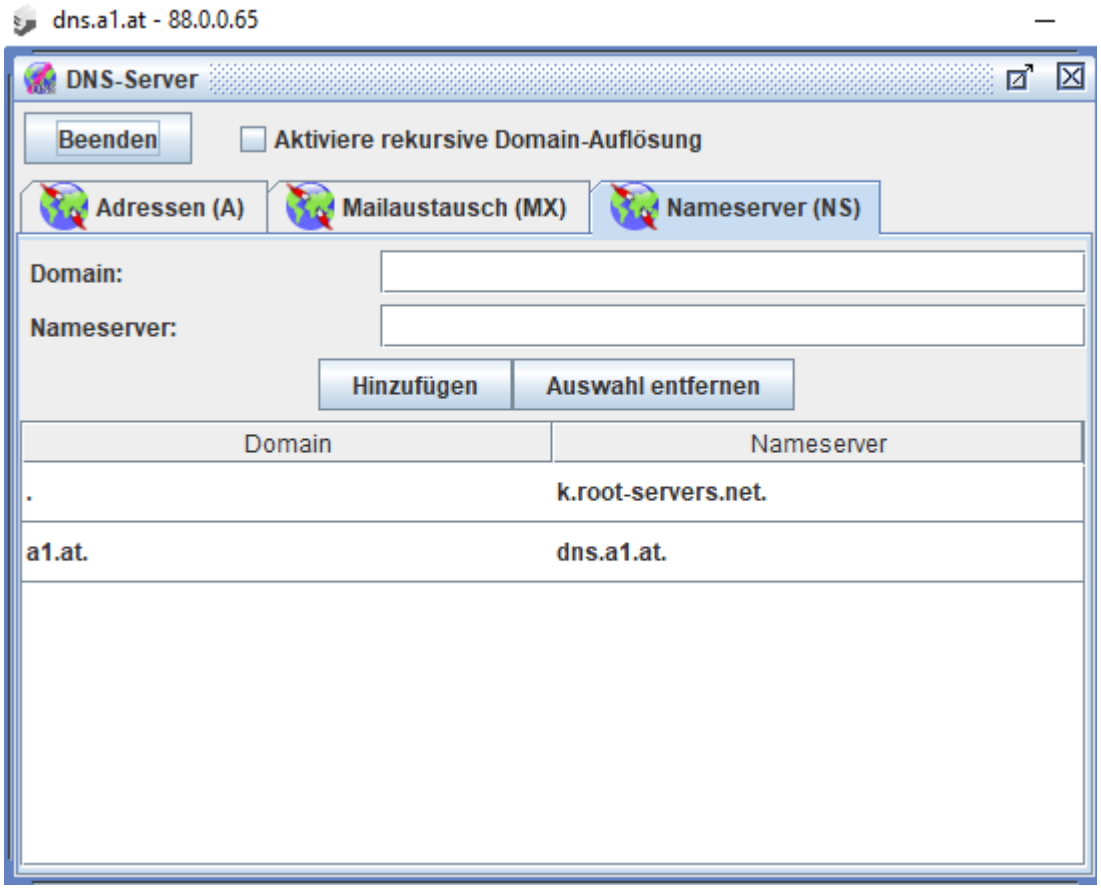
ns.schule.at


ns.schule.at - 10.16.0.1
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domainname:
IP-Adresse:
10.16.0.1
Hinzufügen
Auswahl entfernen
Domainname
IP-Adresse
ns.schule.at.
10.16.0.1
www.schule.at.
10.16.0.2
k.root-servers.net.
111.0.0.1

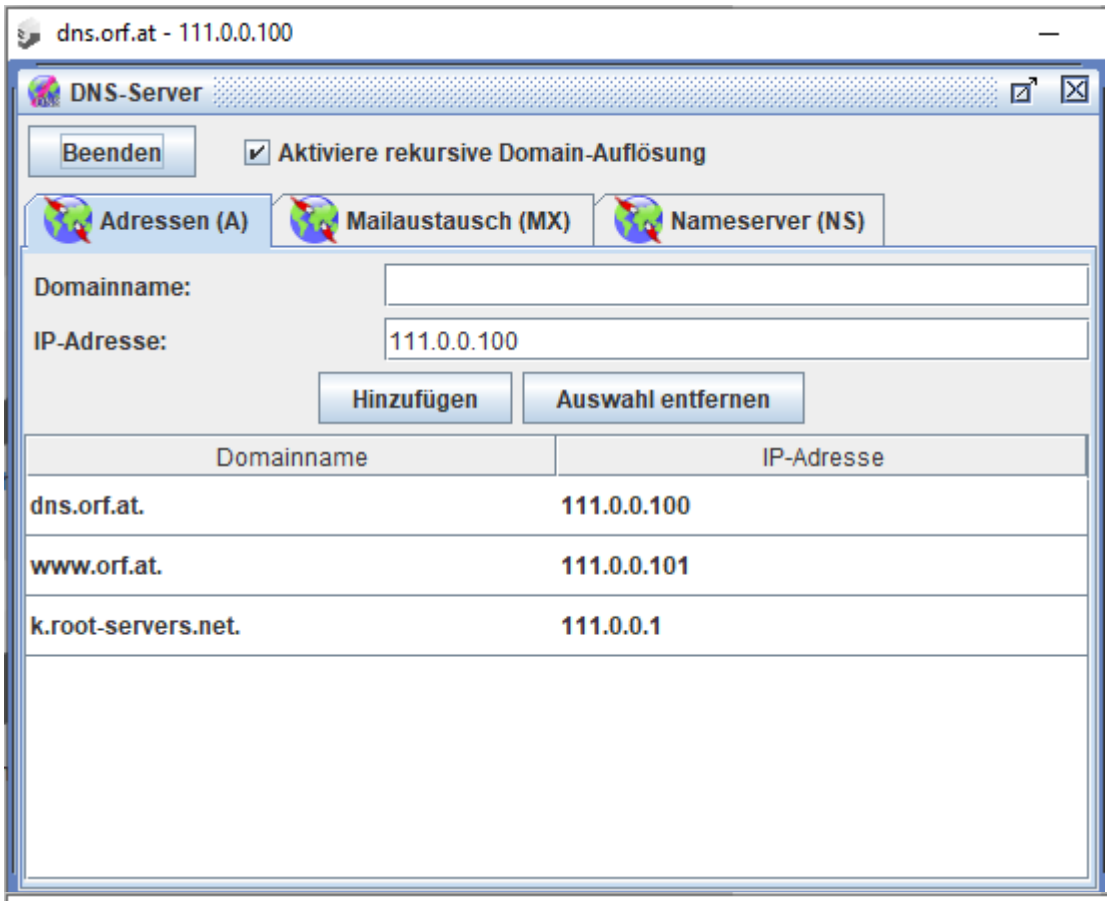

ns.schule.at - 10.16.0.1
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domain:
Nameserver:
Hinzufügen
Auswahl entfernen
Domain
Nameserver
.
k.root-servers.net.
schule.at.
ns.schule.at.

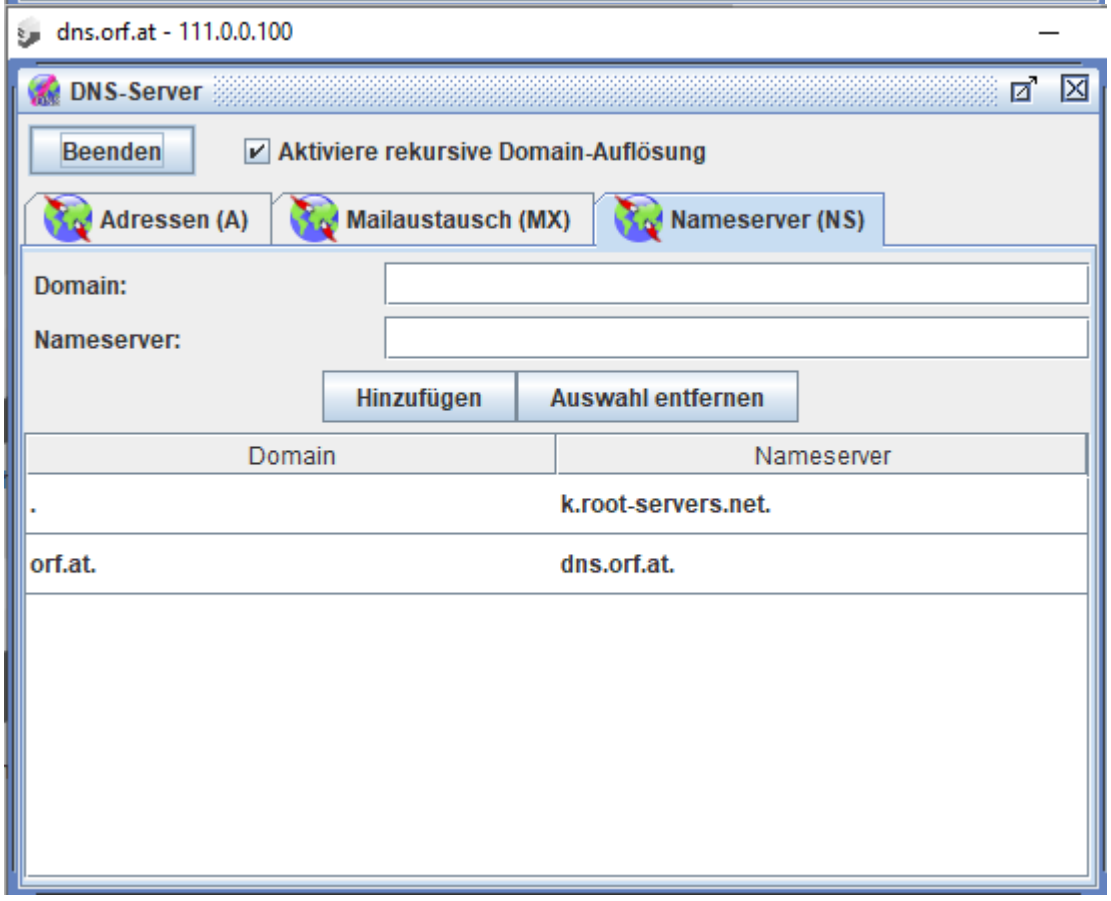
dns.a1.at


dns.a1.at - 88.0.0.65
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domainname:
IP-Adresse:
88.0.0.65
Hinzufügen
Auswahl entfernen
Domainname
IP-Adresse
dns.a1.at.
88.0.0.65
k.root-servers.net.
111.0.0.1

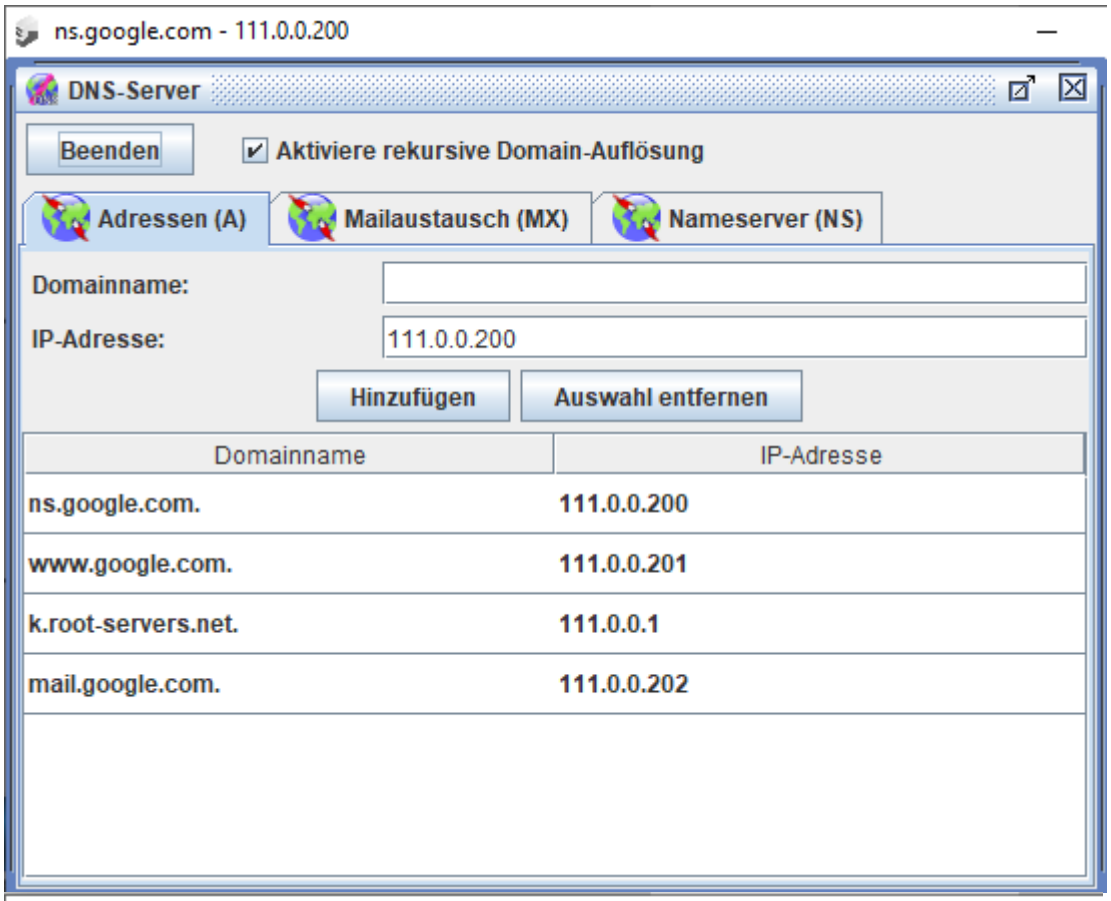

dns.a1.at - 88.0.0.65
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domain:
Nameserver:
Hinzufügen
Auswahl entfernen
Domain
Nameserver
.
k.root-servers.net.
a1.at.
dns.a1.at.

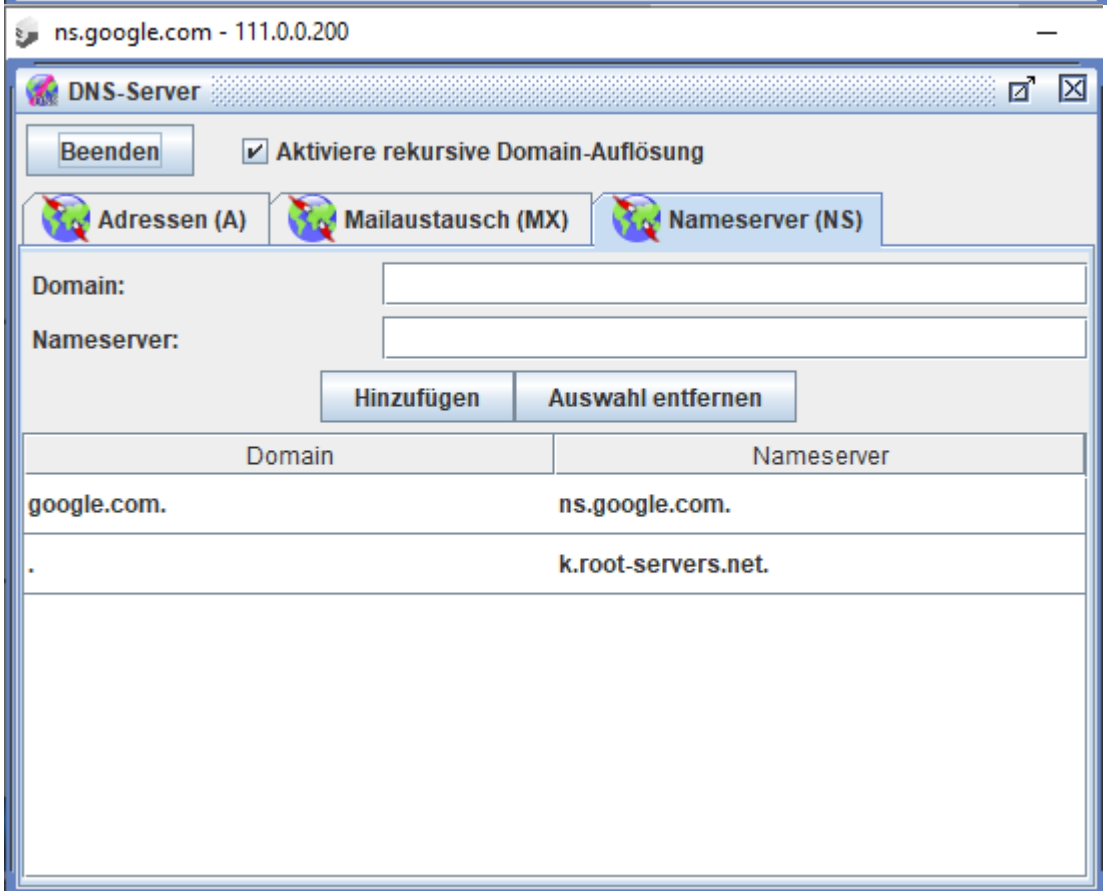
dns.orf.at


dns.orf.at - 111.0.0.100
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domainname:
IP-Adresse:
111.0.0.100
Hinzufügen
Auswahl entfernen
Domainname
IP-Adresse
dns.orf.at.
111.0.0.100
www.orf.at.
111.0.0.101
k.root-servers.net.
111.0.0.1

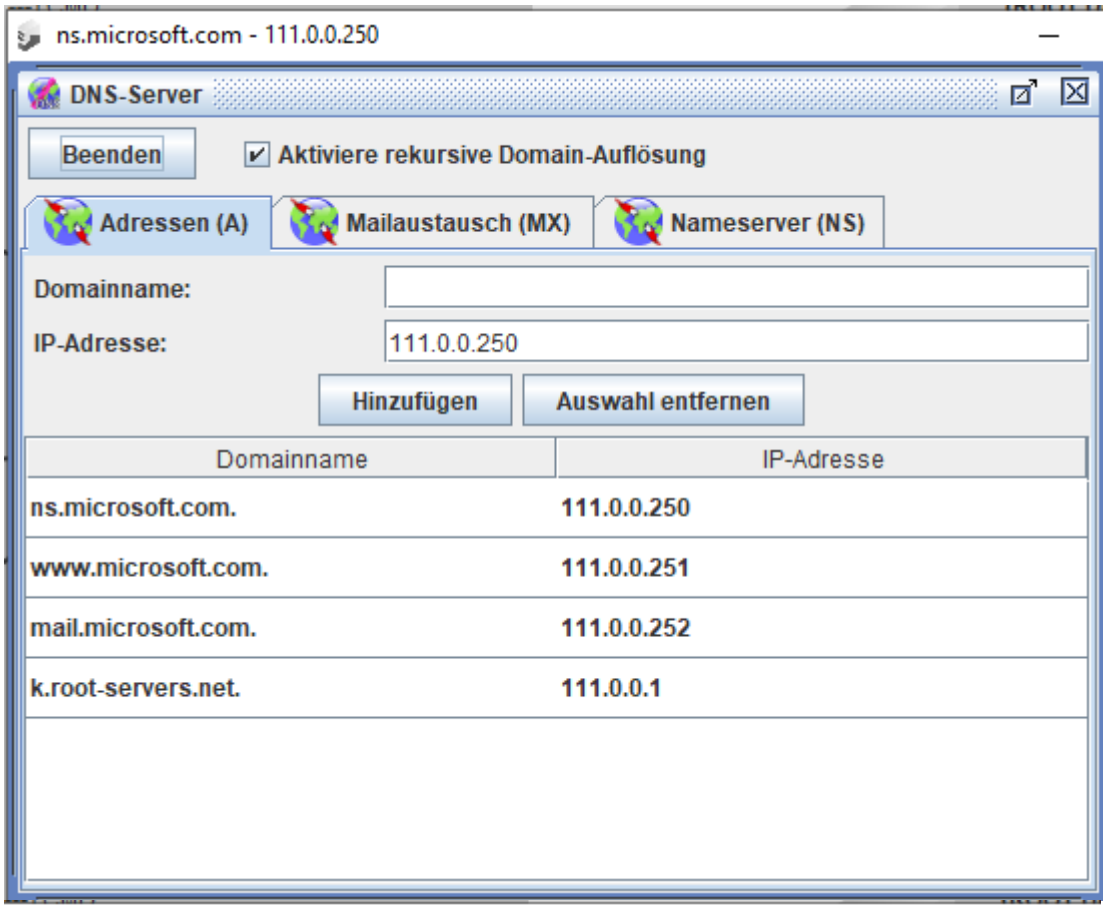

dns.orf.at - 111.0.0.100
DNS-Server
Beenden
Aktiviere rekursive Domain-Auflösung
Adressen (A)
Mailaustausch (MX)
Nameserver (NS)
Domain:
Nameserver:
Hinzufügen
Auswahl entfernen
Domain
Nameserver
.
k.root-servers.net.
orf.at.
dns.orf.at.

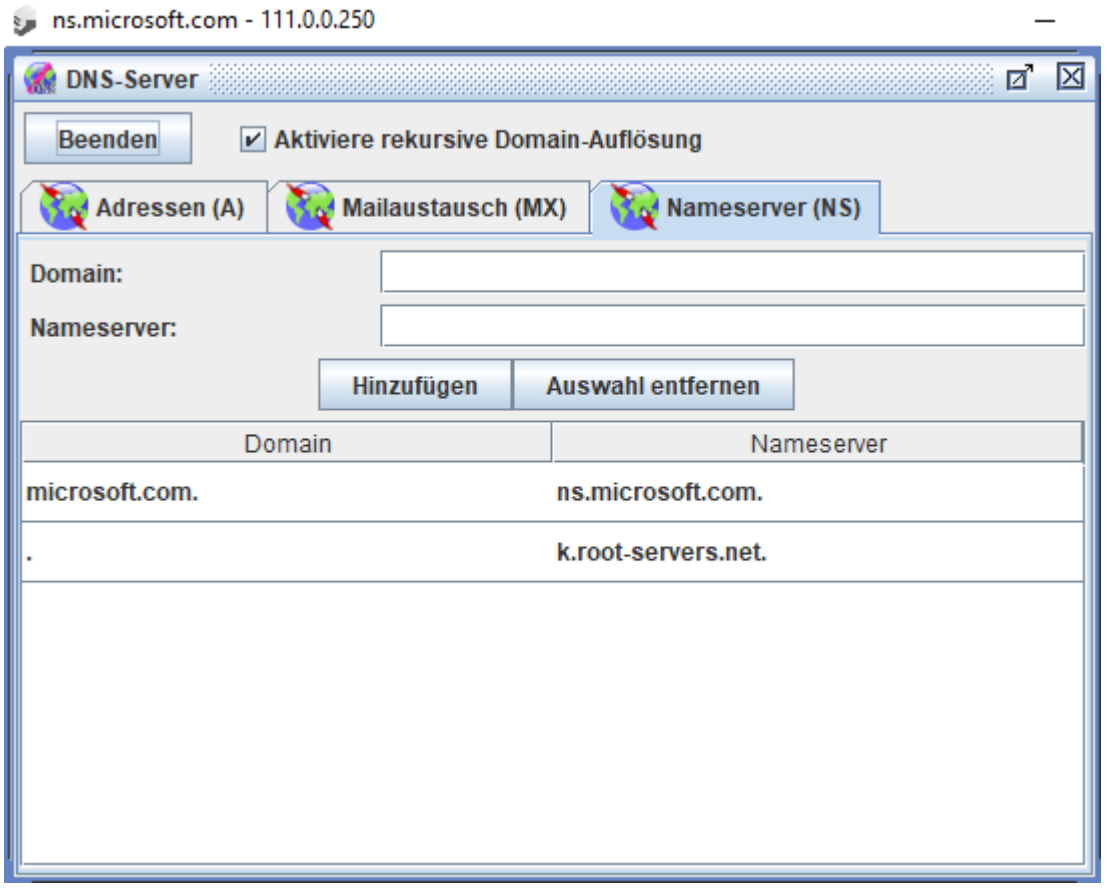
ns.google.com





ns.microsoft.com

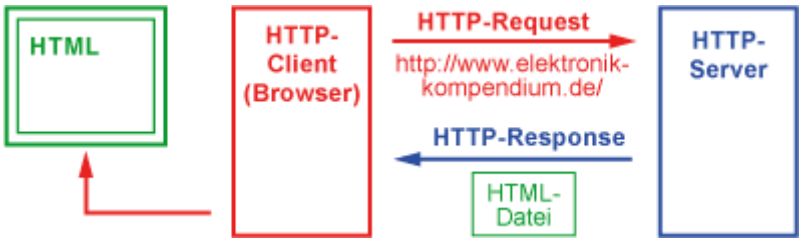




Hypertext Transfer Protocol (HTTP) (Port 80 - Layer 7)

HTTP ist das Kommunikationsprotokoll im World Wide Web (WWW). Die wichtigsten Funktionen sind Dateien vom Webserver anzufordern und in den Browser zu laden. Der Browser übernimmt dann die Darstellung von Texten und Bilder und kümmert sich um das Abspielen von Audio- und Video-Daten.

Funktionsweise



Die Kommunikation findet nach dem **Client-Server-Prinzip** statt. Der **HTTP-Client (Browser)** sendet seine **Anfrage (HTTP-Request)** an den **HTTP-Server (Webserver/Web-Server)**. Dieser bearbeitet die Anfrage und schickt seine **Antwort (HTTP-Response)** zurück. Nach der Antwort durch den Server ist diese Verbindung beendet. Typischerweise finden gleichzeitig mehrere HTTP-Verbindungen statt.

HTTP Adressierung

Damit der Server weiß, was er dem HTTP-Client schicken soll, adressiert der HTTP-Client eine Datei, die sich auf dem HTTP-Server befinden muss. Dazu wird vom HTTP-Client ein **URL (Uniform Resource Locator)** im HTTP-Header an den HTTP-Server übermittelt:

```
http://Servername.Domainname.Top-Level-Domain:TCP-Port/Pfad/Datei

z. B.
http://www.elektronik-kompendium.de:80/sites/kom/0902231.html
```

HTTP Request

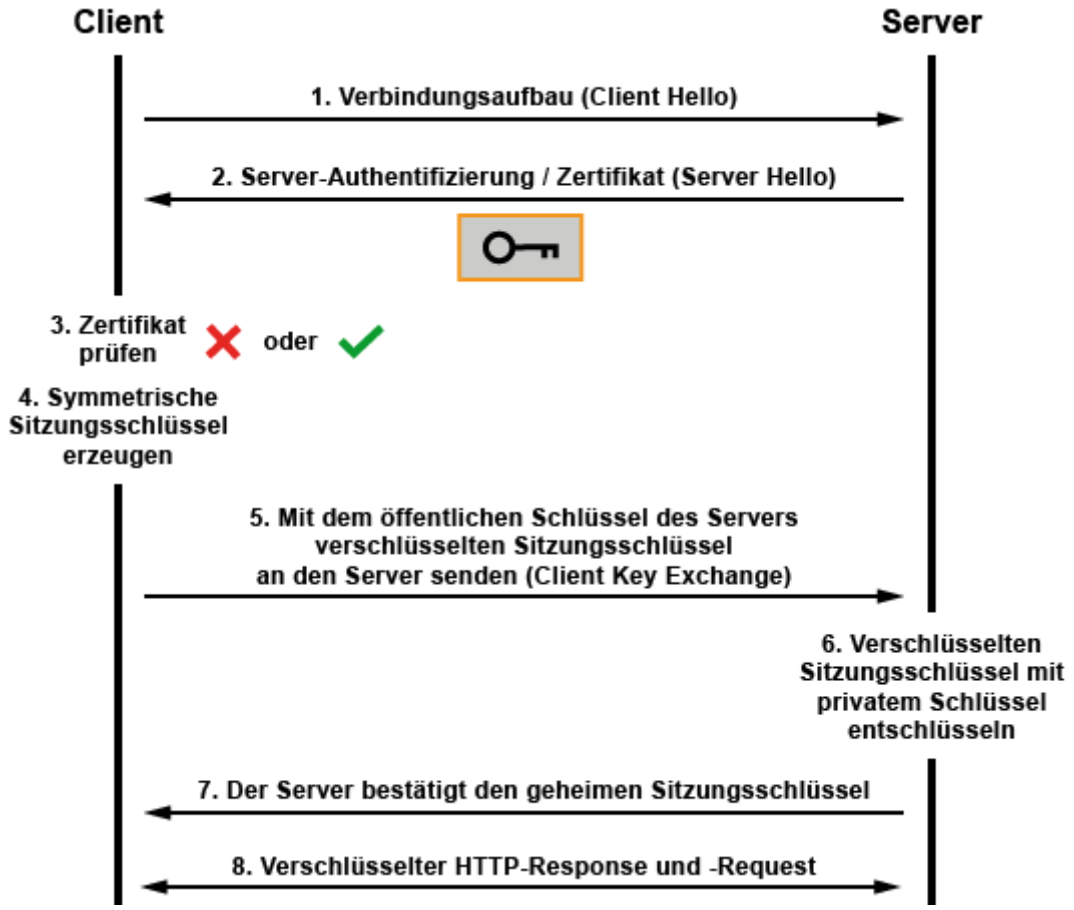
Der HTTP-Request ist die Anfrage des HTTP-Clients an den HTTP-Server. Ein HTTP-Request besteht aus den Angaben **Methode, URL und dem Request-Header**. Die häufigste Methoden sind **GET und POST**. Dahinter folgt durch ein Leerzeichen getrennt der URL und die verwendete HTTP-Version. In weiteren Zeilen folgt der Header und bei der Methode POST durch eine Leerzeile (!) getrennt die Formular-Daten.

HyperText Transfer Protocol Secure (HTTPS) (Port 443 - Layer 7)

HTTPS bzw. HTTP Secure ist die Anwendung von HTTP in **Verbindung mit Verschlüsselung und Authentifizierung**. Wobei in der Regel nur der **angefragte Webserver sich mit einem Zertifikat authentisieren** muss.

Eine **verschlüsselte Verbindung mit einem Browser** signalisiert man mit einem „https:“ *(TCP-Port 443) statt „http:*“ (TCP-Port 80). Dabei muss sich der Webserver dem Client gegenüber authentisieren, ob er tatsächlich der Webserver ist, der sich unter der eingegebenen Adresse befindet. Zusätzlich wird die **Verbindung bzw. Sitzung Ende-zu-Ende-verschlüsselt**. Das bedeutet, die **Stationen zwischen Client und Server** können die Kommunikation **nicht entschlüsseln**.

Für die Authentifizierung und Verschlüsselung ist SSL/TLS verantwortlich. Es schiebt sich zwischen HTTP und dem Transportprotokoll TCP. Damit steht SSL/TLS auch für andere Anwendungsprotokolle zur Verfügung. Beispielsweise SMTPS, IMAPS und FTPS. SSL arbeitet für den Anwender nahezu unsichtbar.



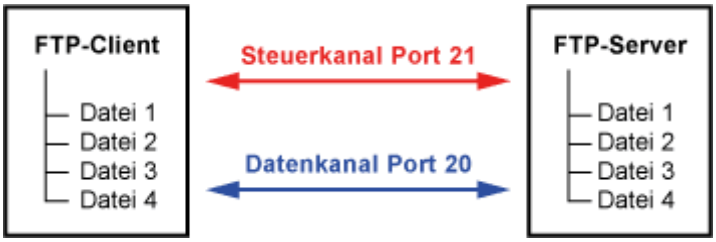
1. Client Hello: Der Client kontaktiert den Server über ein Protokoll mit Verschlüsselungsoptionen.
2. Server Hello, Certificate, Server Key Exchange, Server Hello Done: Der Server nimmt die Verbindung an und schickt sein Zertifikat mit dem öffentlichen Schlüssel seines Schlüsselpaares zur Authentifizierung an den Client.
3. Der Client überprüft das Server-Zertifikat und dessen Gültigkeit (Validierung). Erkennt der Client das Zertifikat als ungültig wird die Verbindung an dieser Stelle abgebrochen.
4. Erkennt der Client das Zertifikat als gültig erzeugt der Client den symmetrischen Sitzungsschlüssel.

5. Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message: Mit dem öffentlichen Schlüssel des Servers verschlüsselt der Client den Sitzungsschlüssel und schickt ihn an den Server.
6. Mit seinem privaten Schlüssel kann der Server den verschlüsselten Sitzungsschlüssel entschlüsseln.
7. Encrypted Handshake Message, Change Cipher Spec, Encrypted Handshake Message: Der Server bestätigt den geheimen Sitzungsschlüssel.
8. Danach werden alle HTTP-Requests und -Responses verschlüsselt, bis die Verbindung abgebaut wird.

File Transfer Protocol (FTP) (Port 20 und 21 - Layer 7)

FTP ist ein Kommunikationsprotokoll, um Dateien zwischen unterschiedlichen Computersystemen zu übertragen. Die Übertragung findet nach dem Client-Server-Prinzip statt. Ein FTP-Server stellt dem FTP-Client Dateien zur Verfügung. Der FTP-Client kann Dateien auf dem FTP-Server ablegen, löschen oder herunterladen. Mit einem komfortablen FTP-Client arbeitet man ähnlich, wie mit einem Dateimanager.

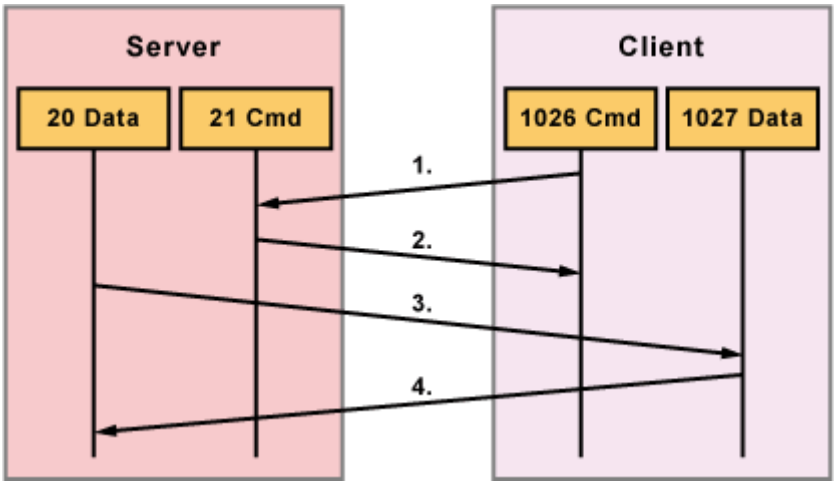
Funktionsweise

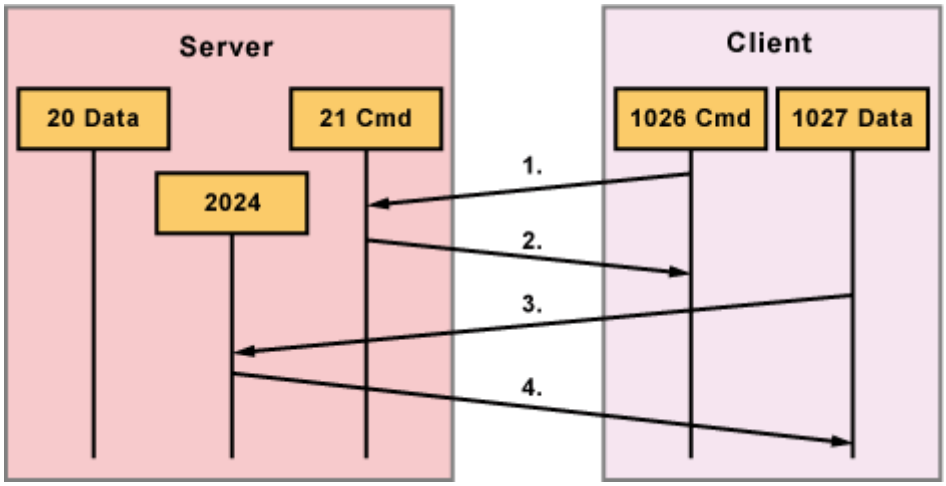


Die Kommunikation findet nach dem Client-Server-Prinzip statt. Wobei FTP zwischen Client und Server zwei logische Verbindungen herstellt. Eine Verbindung ist der Steuerkanal (command channel) über den TCP-Port 21. Dieser Kanal dient ausschließlich zur Übertragung von FTP-Kommandos und deren Antworten. Die zweite Verbindung ist der Datenkanal (data channel) über den TCP-Port 20. Dieser Kanal dient ausschließlich zur Übertragung von Daten. Über den Steuerkanal tauschen Client und Server Kommandos aus, die eine Datenübertragung über den Datenkanal einleiten und beenden.

Aktives vs. Passives FTP

Der FTP-Verbindungsaufbau sieht vor, dass der Steuerkanal vom FTP-Client zum FTP-Server aufgebaut wird. Steht der Steuerkanal wird der Datenkanal vom FTP-Server zum FTP-Client initiiert (aktives FTP). Befindet sich der FTP-Client hinter einem NAT-Router oder einer Firewall und verfügt parallel dazu nur über eine private IPv4-Adresse, dann kommt die Verbindung nicht zustande. Die Verbindungsanforderung vom Server an den Client wird von der Firewall bzw. dem Router abgeblockt, bzw. kann wegen der privaten IPv4-Adresse gar nicht geroutet werden. Für diesen Fall gibt es das passive FTP, bei dem auch der Client den Datenkanal initiiert.





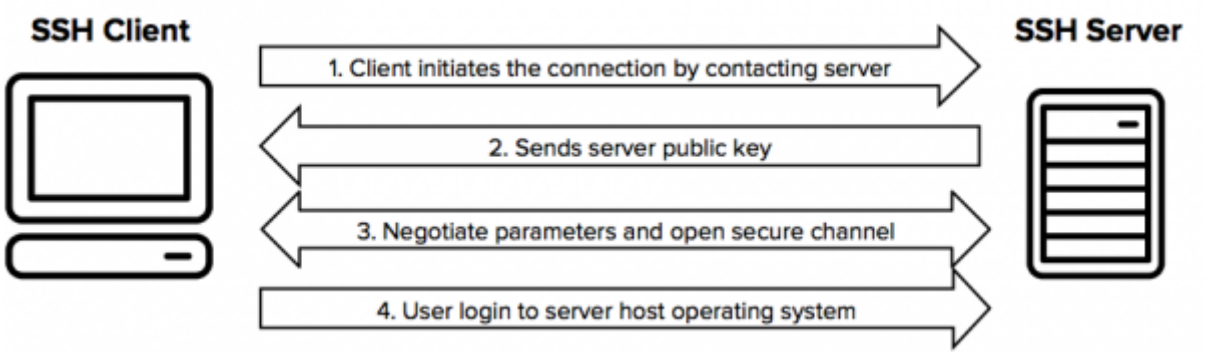
Am Anfang jeder FTP-Verbindung steht die Authentifizierung des Benutzers. Danach erfolgt der Aufbau des Steuerkanals über Port 21 und des Datenkanals über Port 20. Wenn die Dateiübertragungen abgeschlossen sind, werden die Verbindungen vom Benutzer oder vom Server (Timeout) beendet.

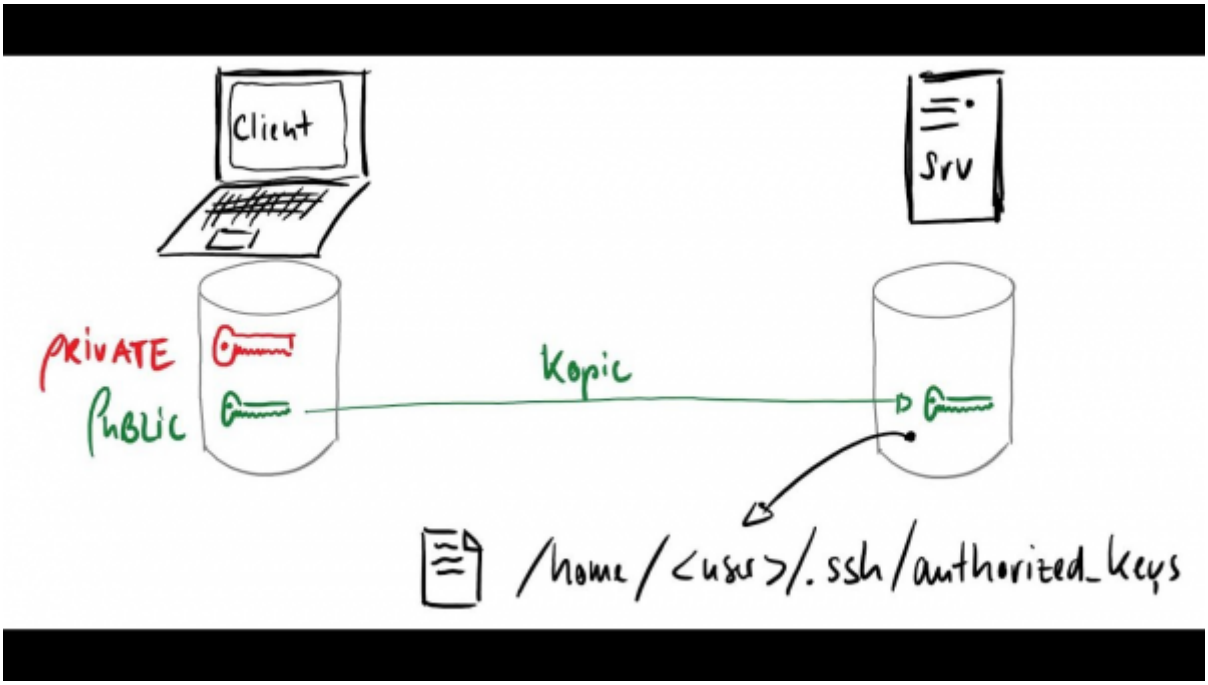
Secure Shell (SSH) (Port 22 - Layer 7)

SSH bzw. Secure Shell ist ein **kryptografisches Protokoll** mit dem man auf einen entfernten Rechner mittels einer **verschlüsselten Verbindung über ein unsicheres Netzwerk** zugreifen kann.

Die Shell (Kommandozeile) bietet **vollen Zugriff auf das Dateisystem und alle Funktionen** des Rechners.

Die Funktionen der Secure Shell beinhalten den Login auf entfernte Rechner, die interaktive und nicht interaktive Ausführung von Kommandos und das Kopieren von Dateien zwischen verschiedenen Rechnern eines Netzwerks. SSH bietet dazu eine kryptografisch gesicherte Kommunikation über das unsichere Netzwerk, eine zuverlässige gegenseitige Authentisierung, Verschlüsselung des gesamten Datenverkehrs auf Basis eines Passworts oder Public/Private-Key-Login-Methoden

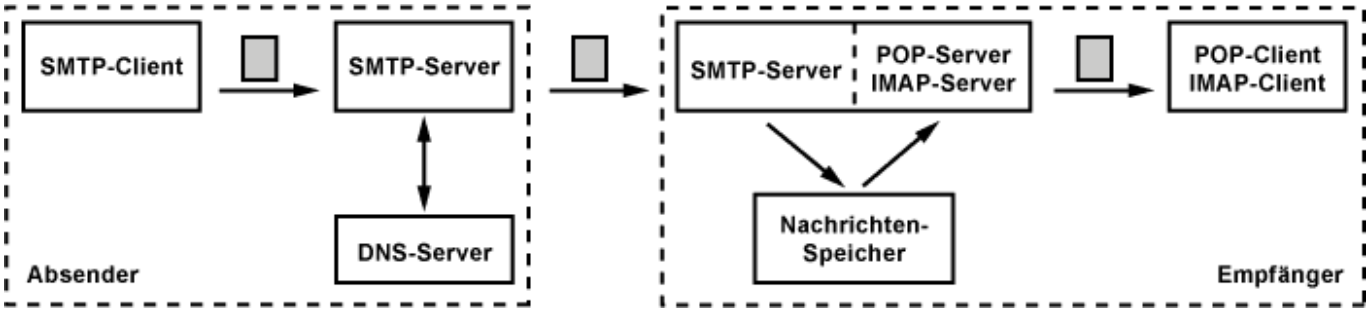




Simple Mail Transfer Protocol (SMTP) (Port 25 - Layer 7)

SMTP ist ein **Kommunikationsprotokoll für die Übertragung von E-Mails**. Die **Kommunikation** erfolgt **zwischen** einem **E-Mail-Client und einem SMTP-Server (Postausgangsserver)** oder zwischen zwei SMTP-Server. Für den **Austausch** der E-Mails sind die **Mail Transfer Agents (MTAs)** zuständig. Untereinander verständigen sich die MTAs mit dem SMTP-Protokoll.

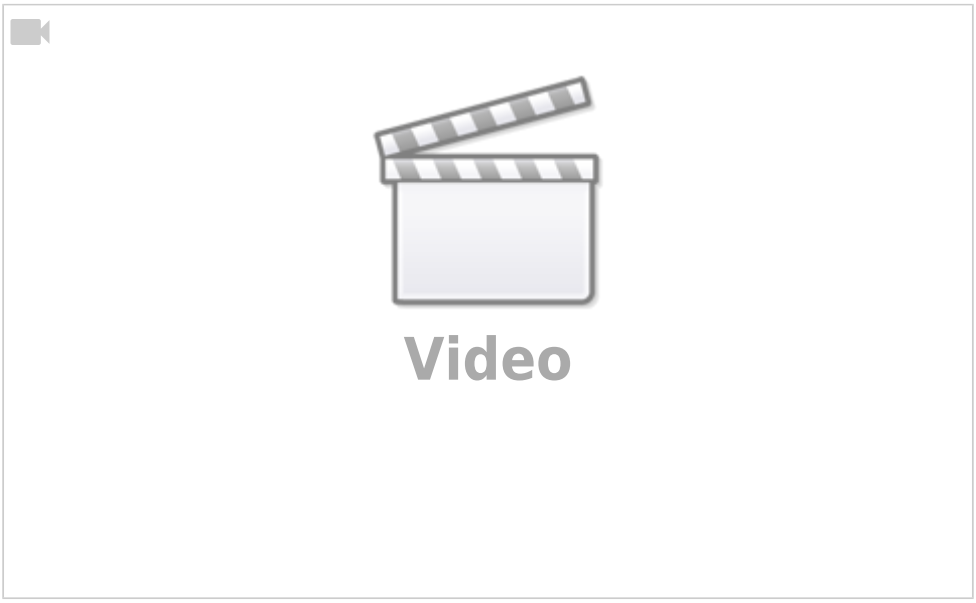
Neben SMTP gibt mit POP und IMAP noch zwei weitere Protokolle für den E-Mail-Austausch. Diese beiden Protokoll dienen jedoch nur dazu, um E-Mail abzuholen oder online zu verwalten. SMTP dagegen ist ein Kommunikationsprotokoll, das **E-Mails entgegennehmen und weiterleiten kann**.

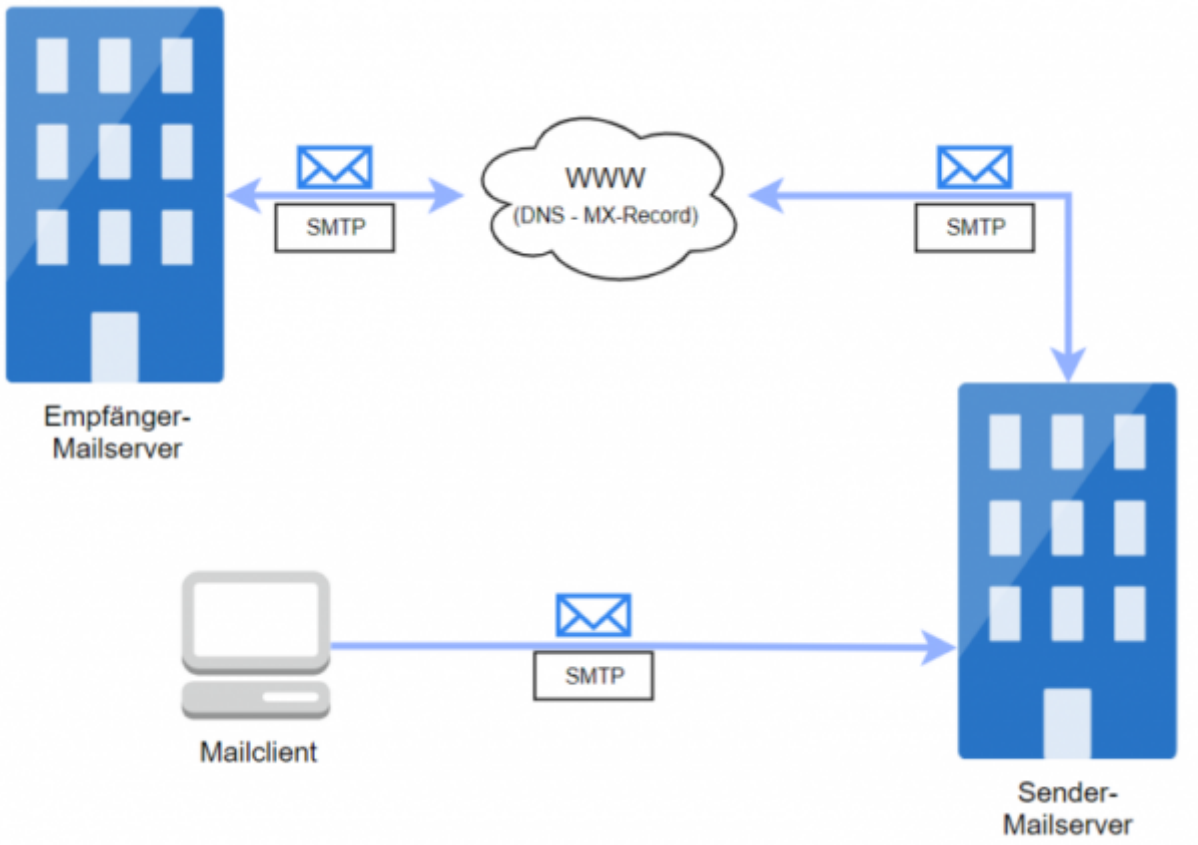


Der Ablauf des E-Mail-Routings sieht in etwa so aus: Der SMTP-Server fragt einen DNS-Server ab und erhält eine Aufstellung von Mail-Servern, die E-Mails für den Ziel-SMTP-Server entgegennehmen. Jeder dieser Mail-Server (Mail Exchange) ist mit einer Priorität versehen. Der SMTP-Server versucht die Mail-Server in der vorgegebenen Reihenfolge zu kontaktieren, um die E-Mail zu übermitteln

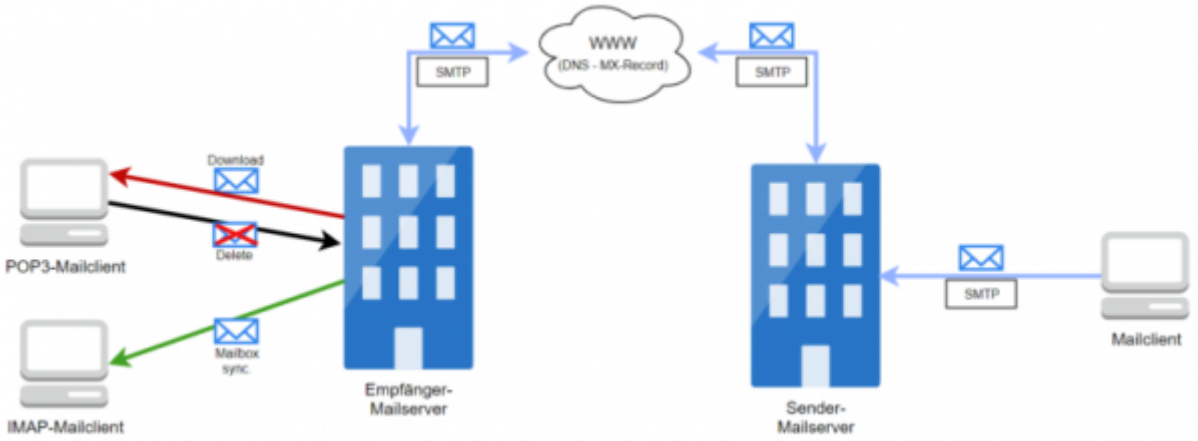
Nachteile

- Für versendete E-Mails keine Versandbestätigung
- Geht eine E-Mail verloren, werden weder Sender noch Empfänger darüber informiert
- Nicht vorhandene Authentisierung des Benutzers beim Verbindungsaufbau zwischen SMTP-Client und SMTP-Server. Das führt dazu, dass eine beliebige Absenderadresse beim Versand einer E-Mail angegeben werden kann





E-Mail Routing



Simple Mail Transfer Protocol Secure (SMTPS) (Port 465 und 587- Layer 7)

SMTPS (Simple Mail Transfer Protocol Secure) bezeichnet ein Verfahren zur **Absicherung der Kommunikation** beim E-Mail-Transport via **SMTP über SSL/TLS** und ermöglicht dadurch **Authentifizierung der Kommunikationspartner** auf Transportebene sowie **Integrität und Vertraulichkeit** der übertragenen Nachrichten.

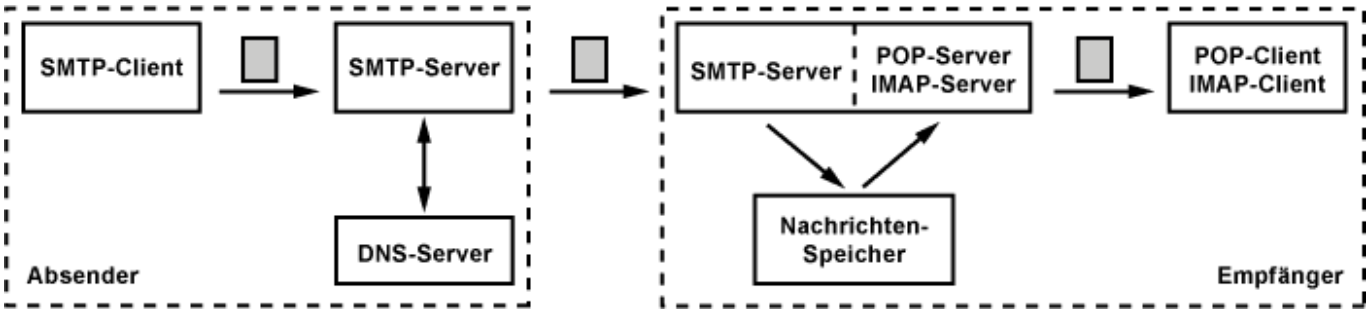
SMTPS ist **kein eigenes Protokoll** und auch keine Erweiterung von SMTP, da es **vollkommen transparent und unabhängig** von diesem auf der Transportschicht arbeitet.

Das bedeutet, dass die Verbindung, über die SMTP abgewickelt wird, softwaremäßig mit den Verfahren **SSL oder TLS abgesichert** wird. Dies geschieht direkt beim Verbindungsaufbau, noch

bevor irgendwelche Maildaten ausgetauscht werden. Da also die Verwendung der Sicherungsschicht nicht verhandelt wird, sind SMTPS-Dienste in der Regel auf einem eigenen TCP-Port erreichbar.

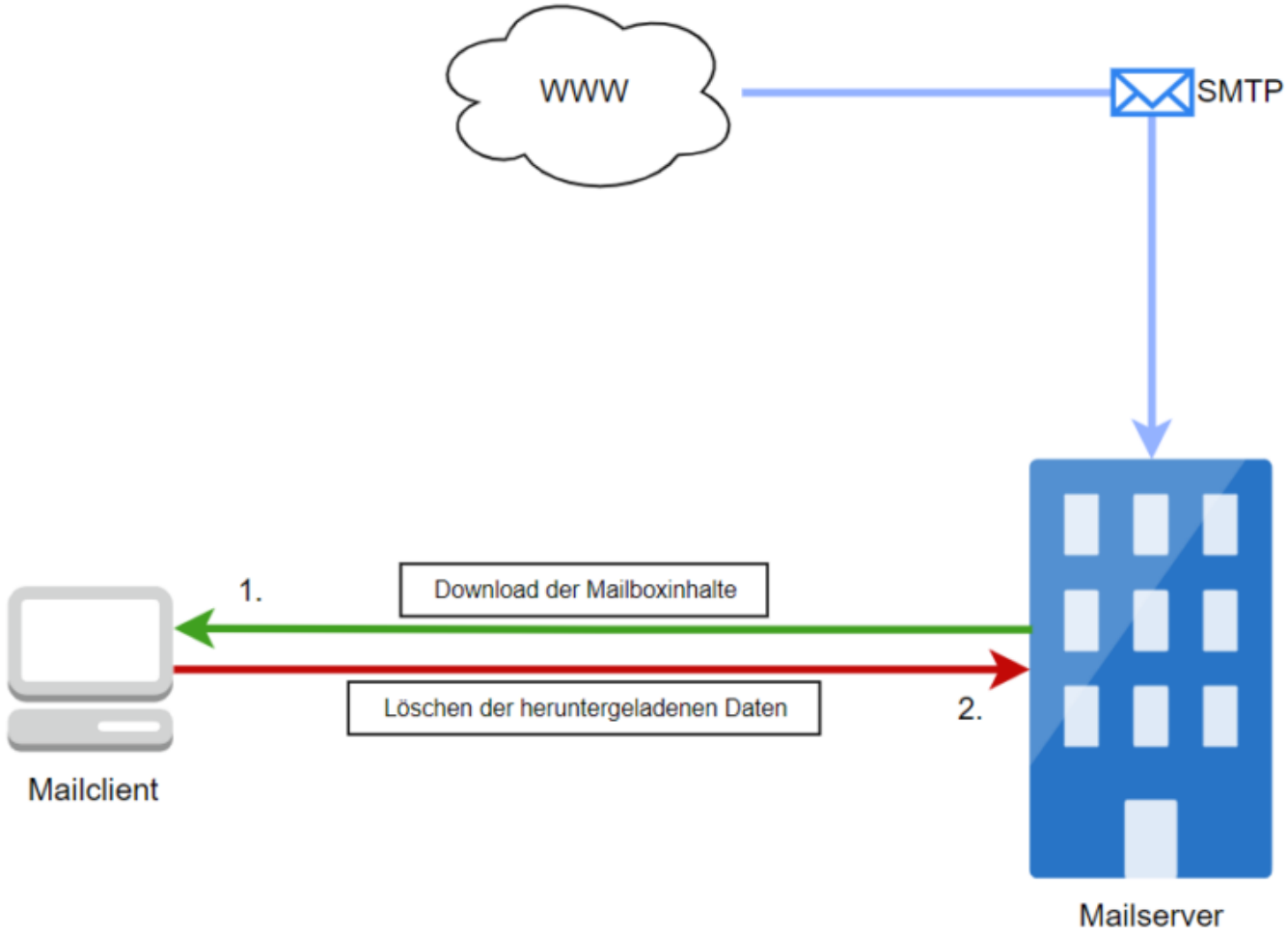
Post Office Protocol Version 3 (POP3) (Port 110 - Layer 7)

POP ist ein Kommunikationsprotokoll, um **E-Mails von einem Posteingangsserver (POP-Server) abzuholen**. Die **Kommunikation** erfolgt **zwischen** einem **E-Mail-Client** und einem **E-Mail-Server (Posteingangsserver)**. Das Protokoll, das diesen Zugriff regelt, nennt sich POP (aus dem Jahr 1984), das in der aktuellen Version 3 vorliegt, und deshalb manchmal auch als POP3 bezeichnet wird.



Per Fernzugriff werden die gespeicherten E-Mails abgerufen und auf dem lokalen Computer gespeichert. POP sieht das Prinzip der Offline-Verarbeitung von E-Mails vor. Online werden die E-Mails vom Posteingangsserver vom E-Mail-Client heruntergeladen. Wenn sich darunter E-Mails mit einem großen Dateianhang befinden, kann der Download schon mal etwas länger dauern. Erst nach erfolgreichem und vollständigem Zugriff werden die E-Mails auf dem Server gelöscht. Die Bearbeitung der eingegangenen E-Mails erfolgt anschließend auf dem lokalen Computer des Benutzers ohne Verbindung (offline) POP-Server.

Die Verbindung zwischen POP-Server und E-Mail-Client erfolgt über TCP auf Port 110.

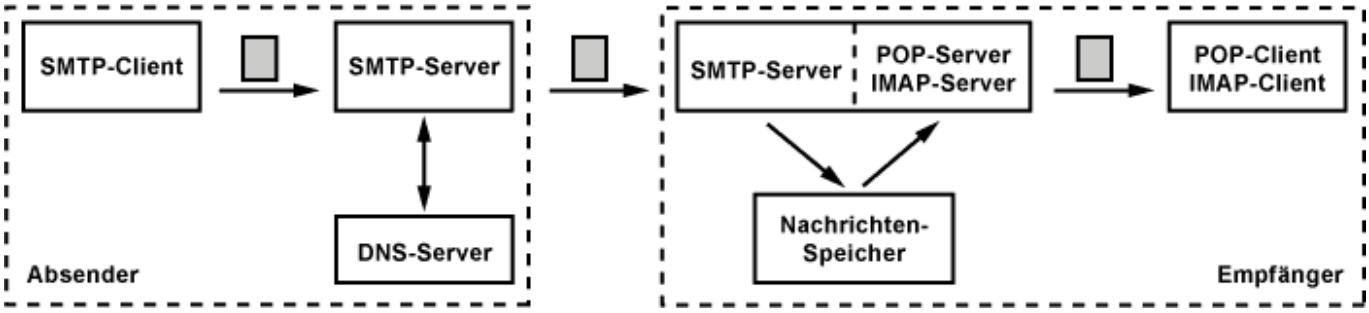


Post Office Protocol Version 3 Secure (POP3S) (Port 995 - Layer 7)

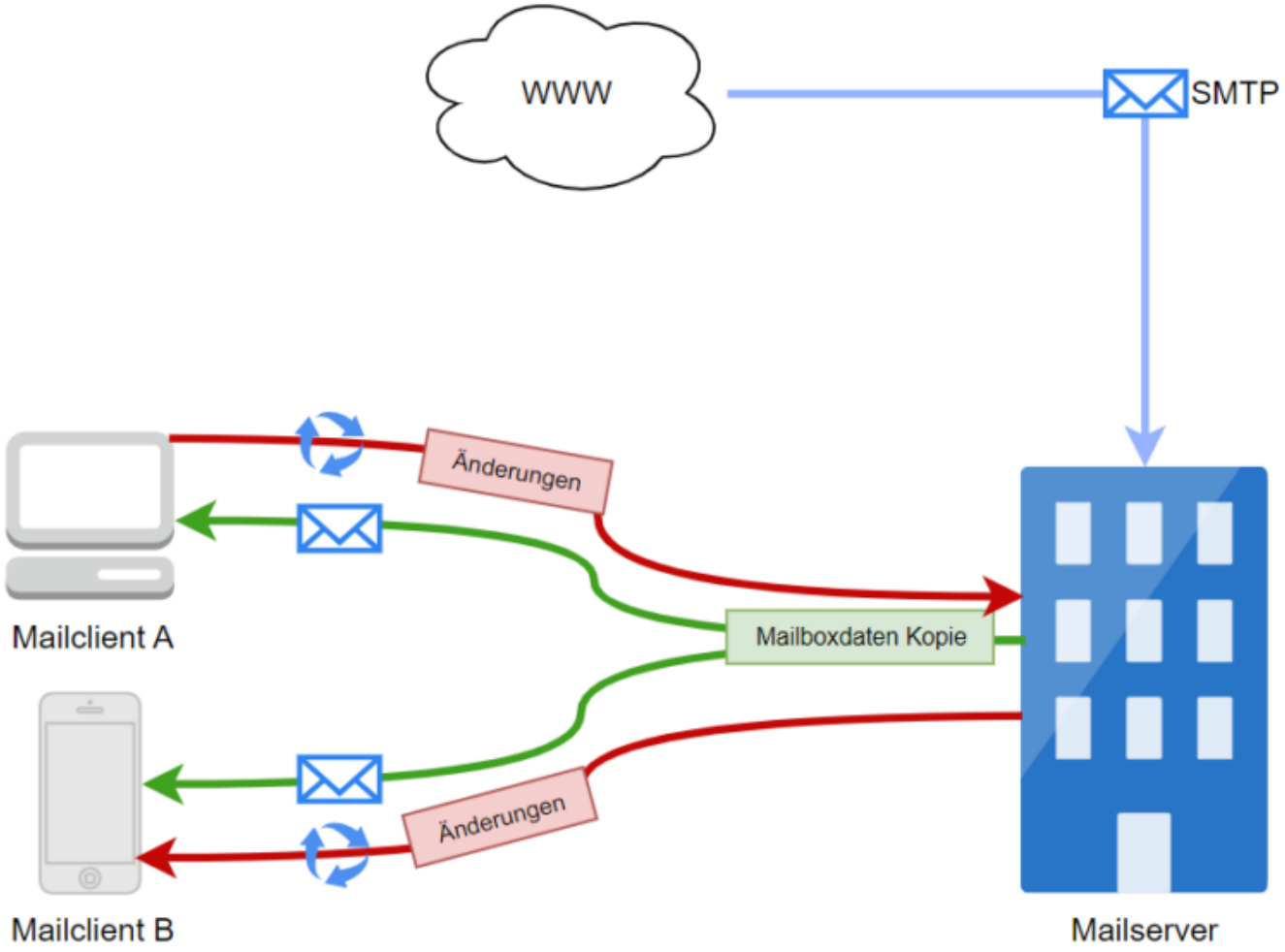
POP3S bezeichnet ein Netzwerkprotokoll zur Erweiterung des E-Mail-Übertragungsprotokolls POP3 um eine Verschlüsselung durch SSL/TLS. Üblicherweise wird für POP3S TCP auf Port 995 genutzt.

Internet Mail Access Protocol (IMAP) (TCP-Port 143 - Layer 7)

IMAP ist ein Kommunikationsprotokoll, um **E-Mails** auf einem **entfernten Server** ähnlich wie Dateien **zu verwalten**. Dabei **bleiben alle E-Mails auf dem IMAP-Server**. Erst wenn eine E-Mail gelesen werden soll, wird sie heruntergeladen.



IMAP erlaubt den Zugriff auf eine Mailbox, **ähnlich wie mit POP**. Der entscheidende **Unterschied** zwischen beiden Protokollen ist der **Online-Modus von IMAP**, über den der **E-Mail-Client ständig in Verbindung mit dem E-Mail-Server** steht. Während einer IMAP-Sitzung kann auf einzelne E-Mails zugegriffen werden, die so lange auf dem Server bleiben, bis sie gelöscht werden. Dadurch kann von überall auf dem Server zugegriffen werden. Auch mit einem Endgerät, das nur mit geringer Bandbreite am Netzwerk angeschlossen ist. Die **E-Mails werden nur dann heruntergeladen, wenn der Anwender diese Lesen will**. E-Mails mit einem großen Dateianhang verstopfen dann nicht mehr ungewollt den Zugang zum Netzwerk.



Internet Mail Access Protocol Secure (IMAPS) (TCP-Port 143 - Layer 7)

Bei der Verwendung von IMAPS wird die Verbindung zum Server bereits während des Verbindungsaufbaus durch SSL verschlüsselt. Damit der Server das erkennt, muss ein anderer Port verwendet werden. Dafür wurde der Port 993 reserviert.

Nach dem Aufbau der SSL-Verbindung wird IMAP verwendet. Die SSL-Schicht ist für das IMAP-Protokoll transparent, d. h., es werden keine Änderungen am IMAP-Protokoll vorgenommen.

IMAP vs POP3

POP3 entstammt noch einer Zeit, in der Mails mit einem Rechner – ob zu Hause oder auf der Arbeit – abgerufen wurden. Viele Menschen greifen heutzutage allerdings mit zahlreichen Geräten (Arbeitsplatz-Rechner, Tablet, Rechner zu Hause, Smartphone,…) auf ihre Mails zurück. Entsprechend offenbaren sich die Schwächen der POP3-Technologie beim Versuch Posteingang und Postausgang auf verschiedenen Geräten in Einklang zu bringen.

Hat der Rechner im Büro die Mails bereits abgerufen und vom Server gelöscht, wird das Tablet mitteilen, es gebe keine neuen Mails. Ein klassischer „Workaround“ wäre zum Beispiel dem Rechner im Büro mitzuteilen, er soll die Mails zwar abrufen, aber er sollte diese bitte auf dem Server belassen. Dann wiederum werden Mails auf allen Geräten als neu betrachtet und entsprechend abgerufen.

Ebenfalls lästig: Markiert man eine Mail als gelesen, passiert dies nur auf dem jeweils einen Gerät. Alle anderen Geräte kriegen davon schlichtweg nichts mit.

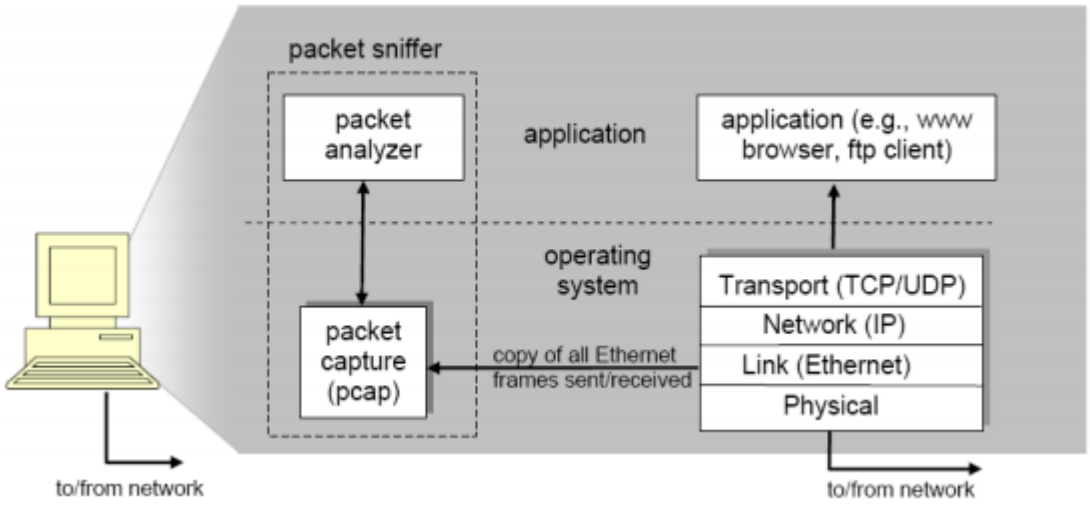
Und: Greift man gerne auf seine „gesendeten Mails“ zurück, wird man ebenfalls feststellen, dass nur die Mails gespeichert werden, die auf dem entsprechenden Gerät verschickt worden sind. Das gleiche gilt im Übrigen auch für die Entwürfe. Mal eben eine Mail im Büro anfangen, in den Entwürfen speichern und auf dem Weg nach Hause in der Bahn via Smartphone beenden und verschicken…geht halt nicht.

Netzwerkanalyse mit Wireshark

Das Monitoring inklusive der notwendigen Detailanalyse des Datenverkehrs im Netzwerk ist ohne ein leistungsfähiges Analysesystem unmöglich. Eines der **wichtigsten Netzwerktools** für jeden **Administrator** ist **Wireshark** – ein Open Source-Netzwerkanalysator, mit dem Sie alle **Pakete im Netzwerk aufzeichnen** und die Paketinhalte detailliert analysieren.

Packet Sniffer

Das grundlegende Werkzeug für die Beobachtung von Daten zwischen Rechnern wird als **packet sniffer** bezeichnet. Wie der Name schon sagt, **fängt** dieses Werkzeug **empfangene/gesendete Daten** Ihres Rechners **ab**. Ein Sniffer ist immer **passiv**, das heißt er v**erschickt keine Daten**, sondern s**peichert Kopien der Daten** der Kommunikation auf Ihrem Rechner.



Das Bild 1 zeigt die Struktur eines „Sniffers“. Rechts unten im Bild sind die beteiligten Protokolle abgebildet, es werden also Protokolle der Layer 1 bis 4:

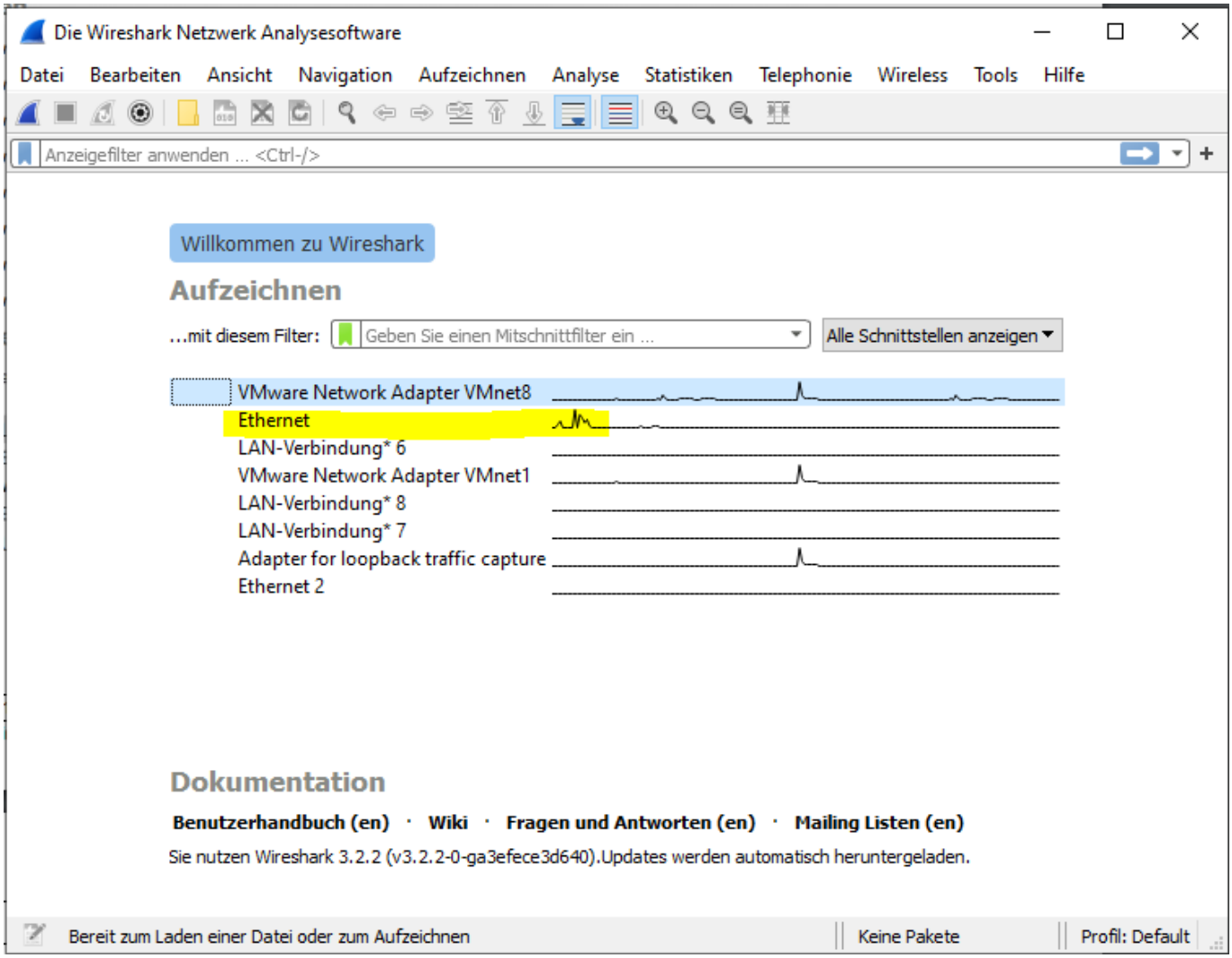
- oben ist die Applikation zu sehen, in unserem Fall ist dies der Webbrowser (Firefox, IE).
- die Blöcke die in den gestrichelten Linien eingerahmt sind gehören zu unserem „Sniffer“.

Am einfachsten lässt sich ein Netzwerk sniffen wenn Hubs als Verbindung der Netzwerkssegmente zwischengeschaltet sind. Bei anderen Verbindungen wie z.B. bei einem Switch bekommt man im Normalfall Probleme den Netzwerkverkehr abzuhören, da der Switch die Daten vom Sender nur an den tatsächlichen Empfänger weiterleitet. Somit würde man nur die eigenen Daten, sowie unwichtigeren Netzwerkverkehr wie z.B. Broadcasts aufzeichnen. Für diesen Fall haben viele Hersteller dieser Komponenten **Switches** bzw. **Router** mit einer **Monitorfunktion** in Ihrem Portfolio. Damit ist es möglich Netzwerkdaten eines gewünschten Ports auf einen anderen zu spiegeln. Auf diesen gespiegelten Port kann man nun direkt zugreifen.

Einführung in Wireshark

Startet man das Programm, muss man zuerst eine Schnittstelle (z.B.: Ethernet) wählen, die man

abhören möchte:

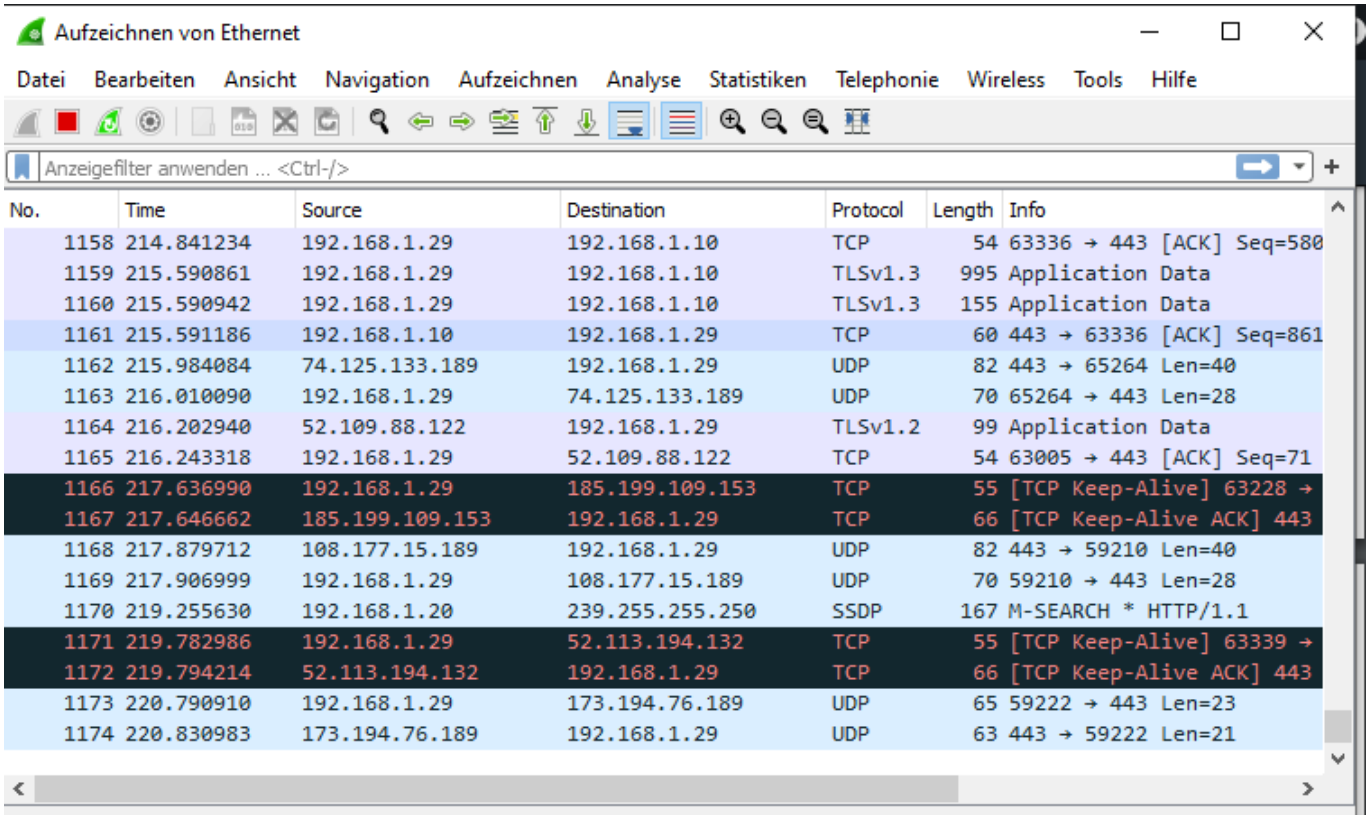


Der Bildschirm, in dem die aufgezeichneten Daten bearbeitet und analysiert werden ist in 3 Bereiche aufgeteilt:

1) Paketliste

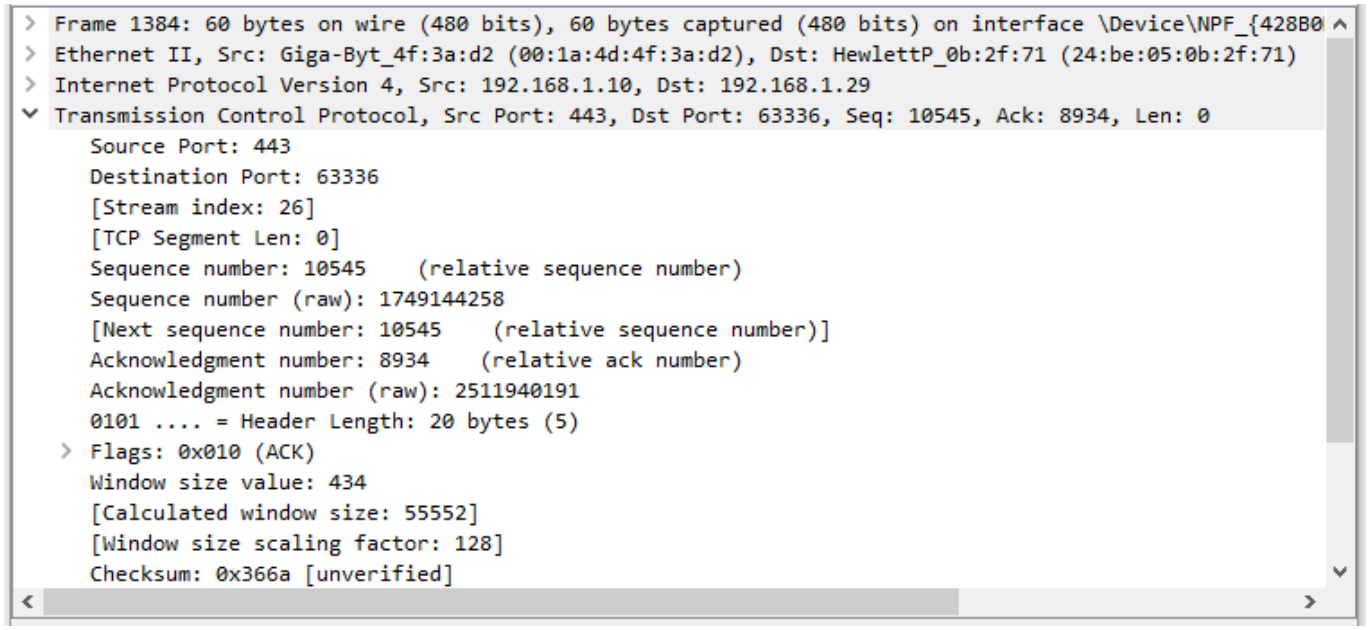
In der Paketliste, sieht man alle aufgezeichneten Frames. Folgende Spalten werden standardmäßig angezeigt:

1. No. = ist eine fortlaufende Nummerierung der Frames 2. Time = zeigt den Zeitabschnitt der Aufzeichnung an 3. Source = zeigt den Absender eines Frames an (meist die IP) 4. Destination = zeigt den Empfänger des Frames an (meist die IP) 5. Protocol = zeigt das verwendete Protokoll des Frame an 6. Info = gibt zusätzliche Informationen zum Frame bekannt



2) Paketdetails

In den Paketdetails werden die OSI-Layer (Schichten) des Datenframes angezeigt. Durch anklicken des Pfeil-Symbols kann der gewählte Layer erweitert werden.



Natürlich variieren die Protokolldetails von Protokoll zu Protokoll. Nur die ersten 2 Schichten sind immer vorhanden.

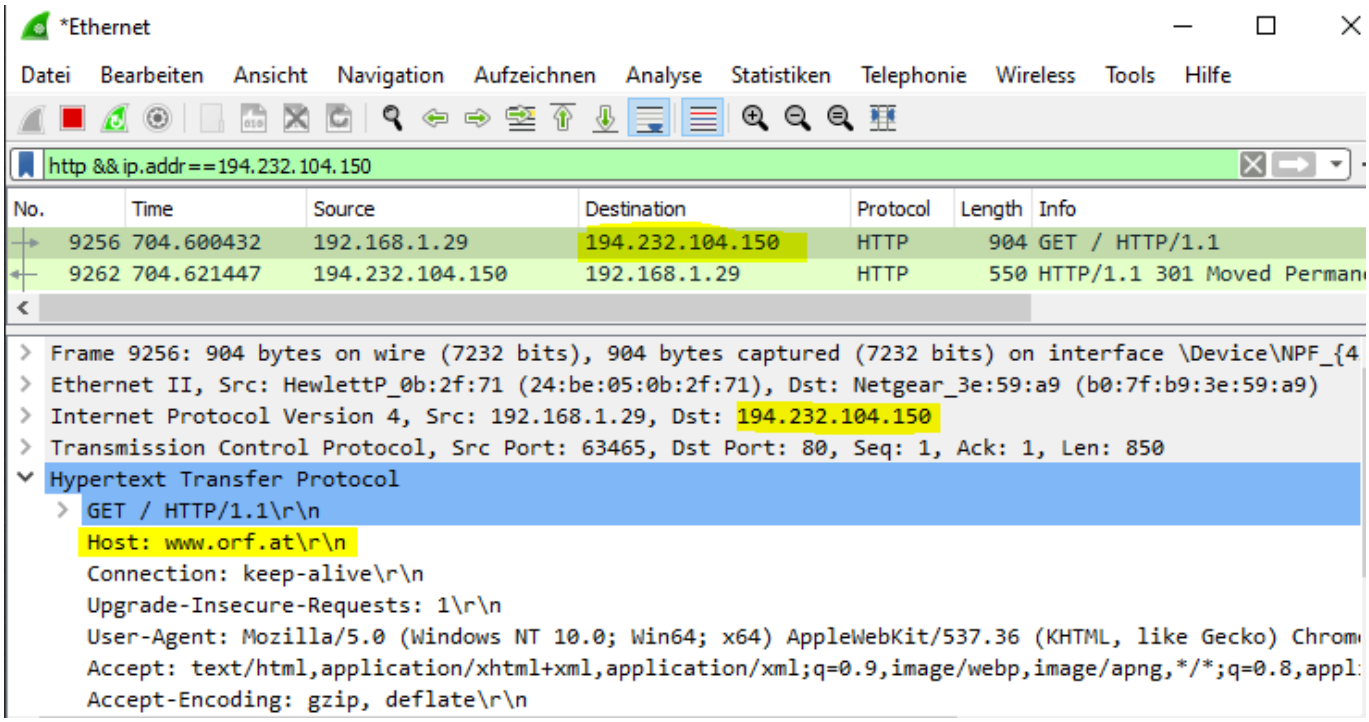
3) Paketdaten (Hexadezimal)

Hier sind die Daten einmal im Hexadezimalsystem (links) und nebendran im Klartext (rechts) bzw. in entschlüsselter Form angezeigt.

```
0000  24 be 05 0b 2f 71 00 1a  4d 4f 3a d2 08 00 45 00   $···/q·· MO:···E·
0010  00 28 8b 6c 40 00 40 06  2b ec c0 a8 01 0a c0 a8   ·(·l@·@· +·····
0020  01 1d 01 bb f7 68 68 41  d2 c2 95 b9 2a 5f 50 10   ·····hhA ····*_P·
0030  01 b2 36 6a 00 00 00 00  00 00 00 00              ··6j···· ····
```

Filter

Nachdem praktisch ständig Netzwerkpakete gesendet und empfangen werden, ist es wichtig, dass es Möglichkeit gibt, um den Netzwerktraffic zu filtern. In dem in der Abbildung gezeigten Textfeld können beliebige Filter eingestellt werden.



Im obigen Beispiel wurden nur http-Pakete mit der IP-Adresse 194.232.104.150 angezeigt. Wie man in der Abbildung erkennen kann, ist/war dies die IP-Adresse der Internetseite www.orf.at. Da http natürlich das TCP-Protokoll, IP-Protokoll & Ethernet-Protokoll nutzt, werden auch diese Daten angezeigt.

Wireshark Übungen

1. Übung

1. Starte Wireshark
2. Starte einen Webbrowser mit einer beliebigen Seite
3. Filtere nach dem Protokoll http.
4. Öffnen die Seite http://elearn.bgamstetten.ac.at
5. Starte die Aufzeichnung nochmals neu und aktualisiere die Seite mit STRG+F5
6. Dokumentiere mittels Screenshot, wie viele gesendete und empfangen Pakete über die Schnittstelle mitgesnifft wurden.
7. Dokumentiere, welche IP-Adresse sich hinter dem FQDN elearn.bgamstetten.ac.at steckt.
8. Dokumentiere, welche Protokolle bei der ersten Antwort (text/html) zum Einsatz kamen.
9. Dokumentiere, wie groß die Antwort (text/html) in Bytes war.
10. Dokumentiere, wie groß
    - der Header des Ethernet 2 – Frames
    - der Header des IP-Paket-Pakets
    - der Header des TCP-Segments
    - und die Daten des http-Protokolls (gzip) der Antwort (text/html) waren.
11. Überprüfe, ob die die Größen zusammen die Gesamtgröße von zuvor ergeben.
12. Finde im Bereich Paketdetails den HTML-Quellcode und eine im Quellcode versteckte Botschaft. Der Name des input-Tags lautet STRENGGEHEIM und die Botschaft verbirgt sich value-Attribut.
13. Kontrolliere deine eigene MAC-Adresse (ipconfig /all) und finde die Ziel-MAC-Adresse heraus.
14. Finde heraus, ob IPv4 oder IPv6 als Protokoll auf der Netzwerkschicht verwendet wurde.
15. Finde heraus, welcher Port auf deinem PC bzw. am ELEARN-Server für die Netzwerkverbindung verwendet wurde.
16. Finde heraus, warum beim ersten Paket im TCP-Header die Sequence Number=1 ist und die Sequence Number beim übernächsten Paket nicht 2 sondern z.B.: 603 ist.
17. Finde heraus, welche HTTP-Version 1.0 oder 1.1 verwendet wurde.
18. Finde heraus, welcher User-Agent verwendet wurde.
19. Finde heraus, welche Sprachen vom Browser akzeptiert werden.
20. Bei der Antwort (text/html) wird ein HTTP-Status-Code mitgeschickt. Finde ihn heraus und recherchiere, was dieser Status Code bedeutet.
21. Finde heraus, welches Betriebssystem bzw. welcher Webserver verwendet wird.
22. Melde dich von der elearn-Plattform ab und anschließend wieder an. Finde heraus, ob und wie deine Login-Informationen mitgeschickt werden (Tipp Info=POST &do=login).