

Socket-C Programmierung

(Client/Server-Modell)

Socket-Programmierung kommt von UNIX und ist auch unter Windows möglich. Es ist ein Client-Server Modell wie alle im Internet angebotenen Dienste. Es setzt auf Schicht 4 auf, die Byte-Order ist deshalb durch die Applikation selbst zu lösen.

	Server	Client
	socket	socket
Verbindungsaufbau	bind	
	listen	
	accept	Connect
Datenphase	send/recv	
Verbindungsabbau	close	close

Server:

```
#define SERVER_PORT 5432
#define MAX_PENDING 1

int main()
{
    struct sockaddr_in sin;
    char buf[256];
    int s, new_s;
    int len;
    WSADATA wData;

    WSStartup(MAKEWORD(1, 0), &wData);

    // Initialisiere die Datenstruktur für die Adresse
    memset(&sin, 0, sizeof(sin));
    sin.sin_family = AF_INET;
    sin.sin_addr.s_addr = INADDR_ANY;
    sin.sin_port = htons(SERVER_PORT);

    // Richte passives Öffnen ein
    if((s = socket(PF_INET, SOCK_STREAM, 0)) < 0)
    {
        fprintf(stderr, "socket\n");
        return 1;
    }
    if(bind(s, (struct sockaddr *)&sin, sizeof(sin)) < 0)
    {
        fprintf(stderr, "bind\n");
        return 2;
    }
    listen(s, MAX_PENDING);

    // Warte auf Verbindung, dann Daten empfangen und ausgeben
    if((new_s = accept(s, (struct sockaddr *)&sin, &len)) < 0)
    {
        fprintf(stderr, "accept\n");
        return 3;
    }
    while((len = recv(new_s, buf, sizeof(buf), 0)) > 0)
    {
        puts(buf);
    }
    closesocket(new_s);
    WSACleanup();
    return 0;
}
```

Client:

```
#define SERVER_PORT 5432

int main(int argc, char *argv[])
{
    struct hostent *hp;
    struct sockaddr_in sin;
    char buf[256];
    int s;
    int len;
    WSADATA wData;

    WSASStartup(MAKEWORD(1, 0), &wData);

    if(argc != 2)
    {
        fprintf(stderr, "host / ip-addr ?\n");
        return 1;
    }

    // Initialisiere die Datenstruktur für die Adresse
    memset(&sin, 0, sizeof(sin));
    sin.sin_family = AF_INET;
    sin.sin_port = htons(SERVER_PORT);

    // Übersetze angegebene IP-Adresse
    if((sin.sin_addr.s_addr = inet_addr(argv[1])) == -1)
    {
        // Übersetze Host-Name in IP-Adresse des Partners
        if(!(hp = gethostbyname(argv[1])))
        {
            fprintf(stderr, "unknown host: %s\n", argv[1]);
            return 2;
        }
        memcpy(&sin.sin_addr, hp->h_addr, hp->h_length);
    }

    // Aktives Öffnen
    if((s = socket(PF_INET, SOCK_STREAM, 0)) < 0)
    {
        fprintf(stderr, "socket\n");
        return 3;
    }

    if(connect(s, (struct sockaddr *)&sin, sizeof(sin)) < 0)
    {
        fprintf(stderr, "connect\n");
        closesocket(s);
        return 4;
    }

    // Hauptschleife: lesen und senden
    for(;;)
    {
        gets(buf);
        len = strlen(buf) + 1;
        send(s, buf, len, 0);
    }
    closesocket(s);
    WSACleanup();
    return 0;
}
```

Datenkompression

- verlustlos
- verlustbehaftet

Verlustlose Kompression bedeutet, dass nach dem komprimieren und anschließenden dekomprimieren der original Datensatz wieder vorliegt. Bei verlustbehafteter Kompression bekommt man nur mehr so ähnliche Daten.

Verlustlos kann man nur komprimieren, wenn Redundanz vorhanden ist. Verlustbehaftet kann man stärker komprimieren als verlustlos.

Verlustlose Datenkompression:

Laufängerkodierung: AAABBBCCCCDDD 3A2B4C3D Es wird einmal der Datensatz gespeichert und dann die Anzahl daran gehängt.

Differenzial Puls Code Methode: ABCACBAC A1202102 Es wird der kleinste Wert gespeichert und dann nur mehr die Differenz zu diesem.

Wörterbuchmethode: Gleich vorkommende Datenblöcke werden gesucht und im Wörterbuch durchnummeriert gespeichert. Anschließend wird nur mehr durch die Nummer auf den Datenblock verwiesen. Verwendet unter anderem bei ZIP, GIF

Entropiekodierung:	0	30%	<u>0 0</u>	60
	1	20%	<u>0 1</u>	40
	2	20%	1 <u>0 0</u>	60
	3	10%	1 0 <u>1</u>	30
	4	5%	1 1 <u>0 0</u>	20
	5	5%	1 1 <u>0 1</u>	20
	6	3%	1 1 1 <u>0 0</u>	15
	7	3%	1 1 1 <u>0 1</u>	15
	8	2%	1 1 1 <u>1 0</u>	10
	9	2%	1 1 1 1 <u>1</u>	10
	4bit * 100 : 400bit			280

1	0	1	1	1	0	0	0	0	1	0	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---

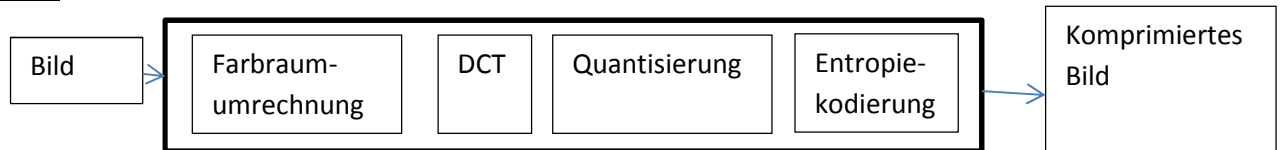
3 4 0 1 0 1

Verfahren:

1. Die Werte nach den vorkommenden Wahrscheinlichkeiten sortieren
2. Immer bei der Hälfte der Wahrscheinlichkeiten teilen und anschließend kodieren

Verlustbehaftete Datenkompression:

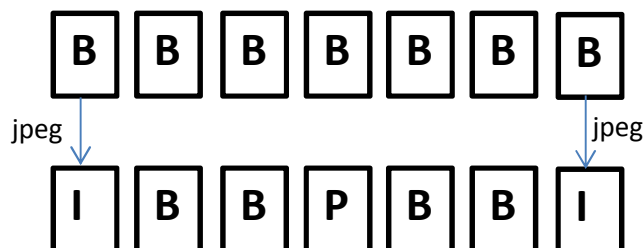
JPEG:



DCT ... diskrete Cosinus-Transformation (verwandt mit der Fast Foyer Transformation)

Die Farbraumumrechnung ist Verlustbehaftet. Bei der Quantisierung werden Daten absichtlich verworfen (verlustbehaftet).

MPEG:



I-Frames sind JPEG-komprimierte Bilder, P-Frames sind die Differenzen zum vorigen I-Frame und B-Frames sind die Differenz zu den vorigen/dahinerliegenden I oder P-Frames.

Es gibt unterschiedliche Verfahren: MPEG1, MPEG2 (verwendet bei DVD, DVB-T, DVB-S, ...), MPEG4 (wird verwendet bei Blue Ray, DVB-T2, DVB-S2, ...)

MP3:

MP3 ist ein Kompressionsverfahren für digitale Audiodaten (8kb/s – 320kb/s). MP3 bedient sich dabei der Psychoakustik mit dem Ziel, vom Menschen nicht wahrnehmbare Signalteile wegzulassen.

normale Musik-CD: $2 * 44,1 \text{ kHz} * 16b = \sim 1,4 \text{ Mb/s}$

Verschlüsseln

- guter Algorithmus
- Schlüssellänge symmetrisch: $\geq 90\text{bit}$
- Implementierung

- symmetrisch
- DES 56bit
 - 3DES 112bit
 - AES 128bit/256bit

- asymmetrisch
- RSA 1024/2048/4096bit
 - elliptische Kurven 160/224bit

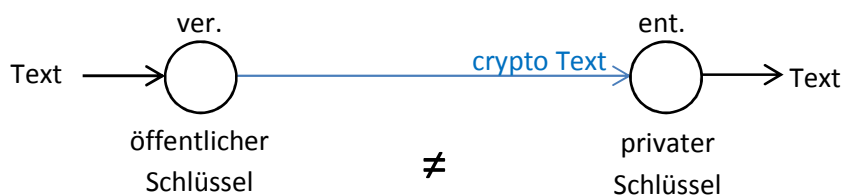
Brute Force:

$$\frac{2^{112} \text{ Schlüssel}}{10^6 \frac{\text{Schlüssel}}{\text{sec}}} = 5,2 * 10^{27} \text{ sec} = 1,6 * 10^{20} \text{ Jahre}$$

symmetrische Verschlüsselung:

Man muss zuerst komprimieren und dann verschlüsseln, weil nach der Verschlüsselung keine Redundanz mehr vorhanden ist. Man sollte so viel wie möglich verschlüsseln, damit wichtige Übertragungen nicht auffallen.

asymmetrische Verschlüsselung:

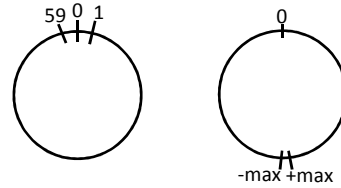
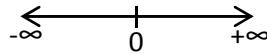


Es gibt 2 verschiedene Schlüssel, die gemeinsam erzeugt werden müssen. Die CA verteilt die öffentlichen Schlüssel weltweit und garantiert für deren Echtheit. Die asymmetrische Verschlüsselung braucht im Vergleich zur symmetrischen Verschlüsselung etwa 100...1000 Mal mehr Zeit.

RSA

math. Grundlage:

1. Zahlenraum



2. Kehrwert

$$x * \frac{1}{x} = 1 = x * y$$

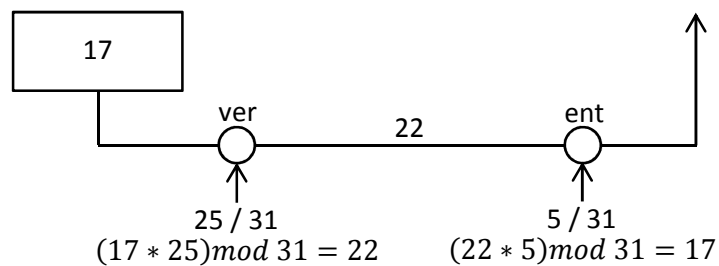
$$\frac{1}{x} = y \dots \text{Kehrwert zu } x$$

1 ... *neutrales Element*

$$(x * y) \bmod r = 1$$

$$x = 25 \quad r = 31 \quad y = ?$$

$$(25 * y) \bmod 31 = 1 \rightarrow y = 5$$



in Echt:

$$N = p * q$$

$p, q \dots$ Primzahlen, $p \neq q$

$$\varphi(N) = (p-1)(q-1)$$

$$1 < e < \varphi(N)$$

$e \dots$ Teilerfremd zu $\varphi(N)$ [= kein Vielfaches]

$$(e*d) \bmod \varphi(N) = 1$$

$$\text{verschlüsseln: } K^e \bmod N = C$$

$K \dots$ Klartext, $C \dots$ Verschlüsselter Text

$$\text{entschlüsseln: } C^d \bmod N = K$$

Beispiel:

$$p = 11 \quad q = 13$$

$$N = 11 * 13 = 143$$

$$\varphi(N) = 10 * 12 = 120$$

$$e = 23 \quad (23*d) \bmod 120 = 1$$

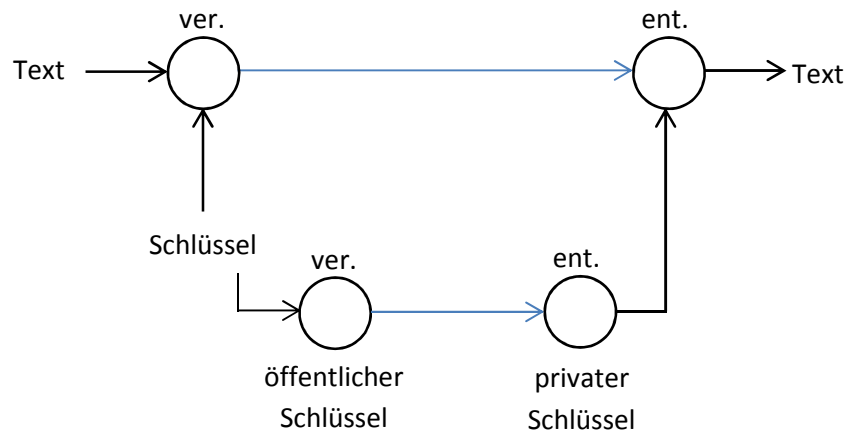
$$d = ? \quad d = 47$$

$$k = 7$$

$$7^{23} = 7 * 7^{22} = 7 * (7^{11})^2 = 7 * (7 * (7^5)^2)^2$$

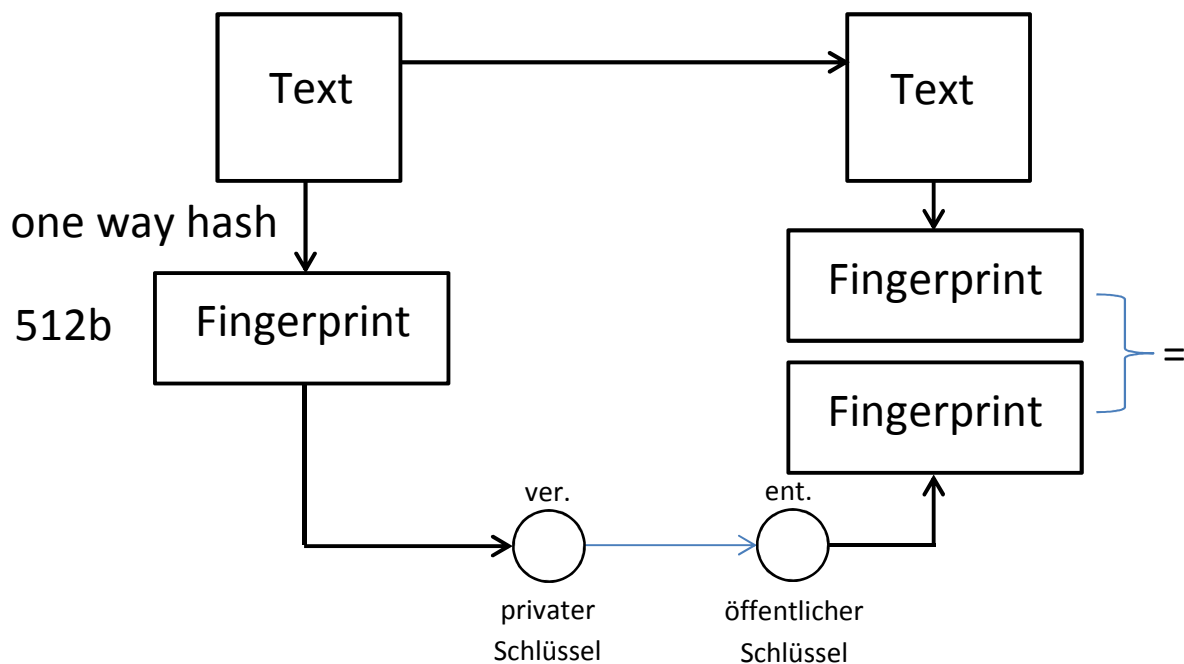
$$7^{23} \bmod 143 = 2$$

Schlüsselaustausch



Als weitere Schlüsselaustauschmethode gibt es den Schlüsselaustausch nach Diffie-Hellman.

digitale Signatur



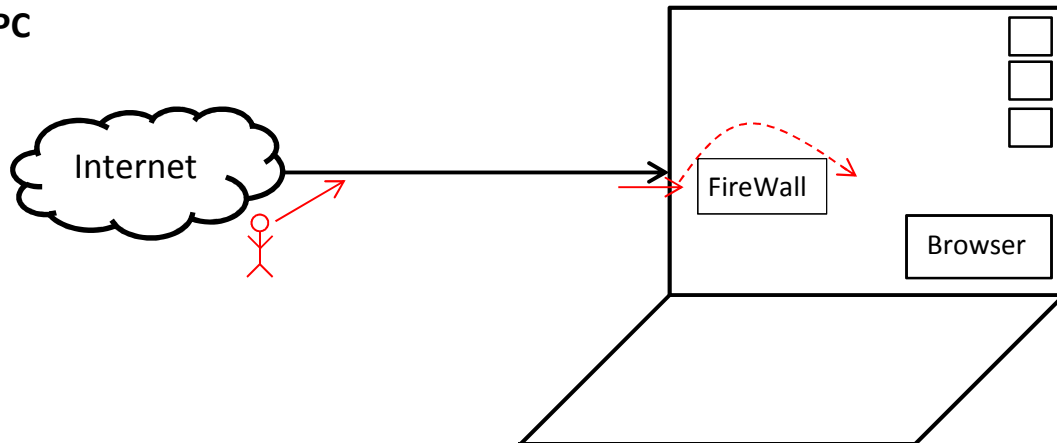
Die digitale Signatur ist gleichwertig einer händischen Unterschrift. Sind bei der Überprüfung die beiden Fingerprints gleich, so ist das Dokument unverändert und der Richtige hat unterschrieben. Die digitale Signatur ist kein Schutz gegen Einsichtnahme. In Österreich gibt es eine digitale Signatur-

Gesetz, welches die digitale Unterschrift unter Einhaltung gewisser Voraussetzungen der händischen Unterschrift gleichsetzt. Ausnahme sind nur Dokumente, die Notaren vorbehalten sind.

Firewall

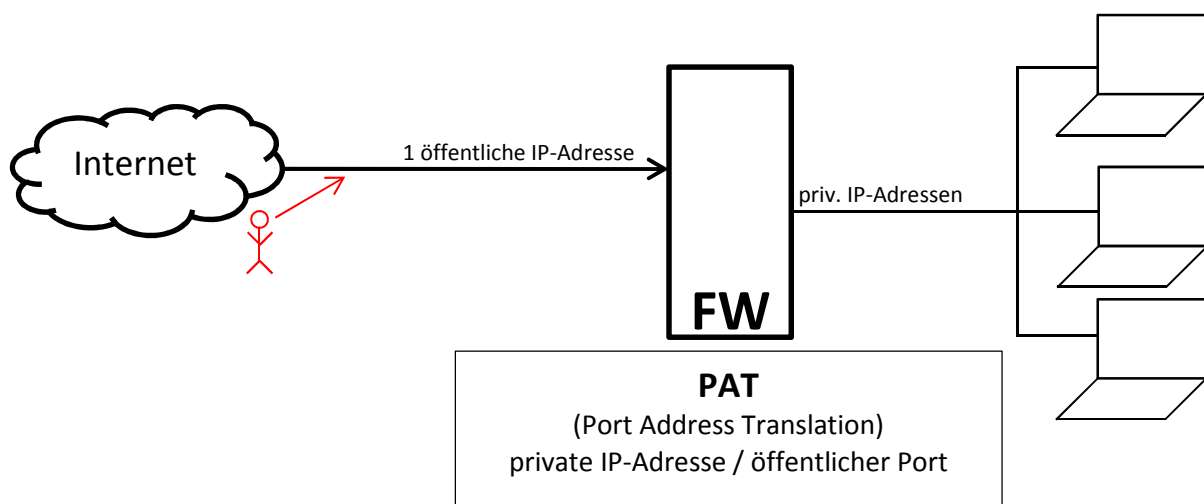
Firewall ist ein Schutz gegen Zugriffe von außen. Virenschutz ist der Schutz gegen das eigene Herunterladen von Viren.

1. PC



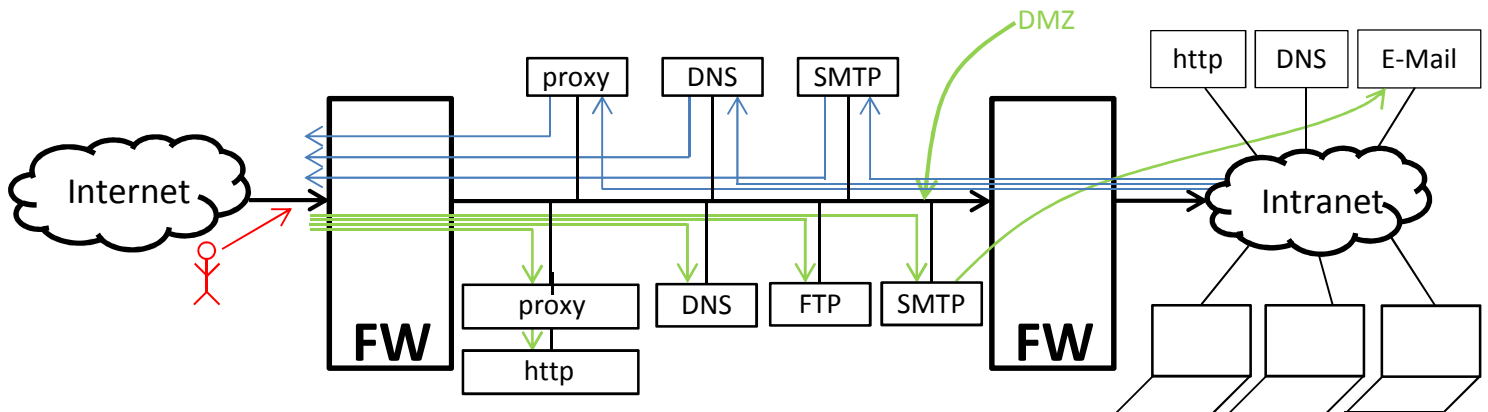
Die Firewall blockiert die Portzugriffe (Angriffe) von außen. Nachteil dieser Lösung ist, dass der Angreifer bereits das Zielsystem sieht und die Firewall erst nach dem IP-Stack des Betriebssystems ist.

2. Heimnetz



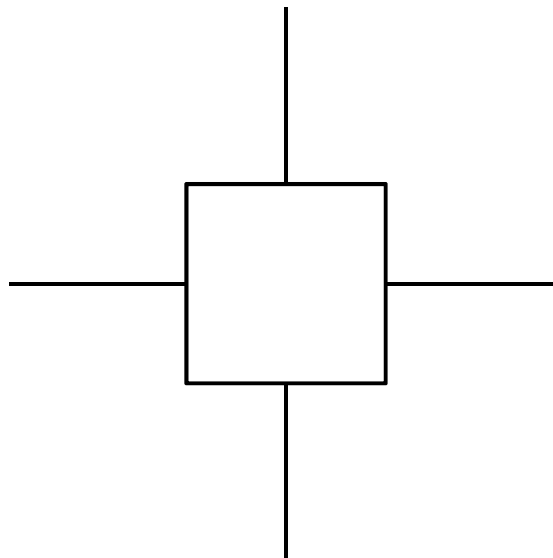
Der lokale Netztransfer ist frei, Zugriffe von innen nach außen werden per PAT (Port-Address-Translation) übersetzt. Die Firewall merkt sich diese Tabelle damit Antworten aus dem Internet wieder richtig im lokalen Netzwerk zugestellt werden. Unbekannte Pakete (nicht in der Tabelle vorhanden) müssen verworfen werden. Vorteil: der Angreifer ist noch nicht auf dem Zielsystem, der Programmcode auf der Firewall ist viel kürzer, dadurch weniger Fehler.

3. Firma

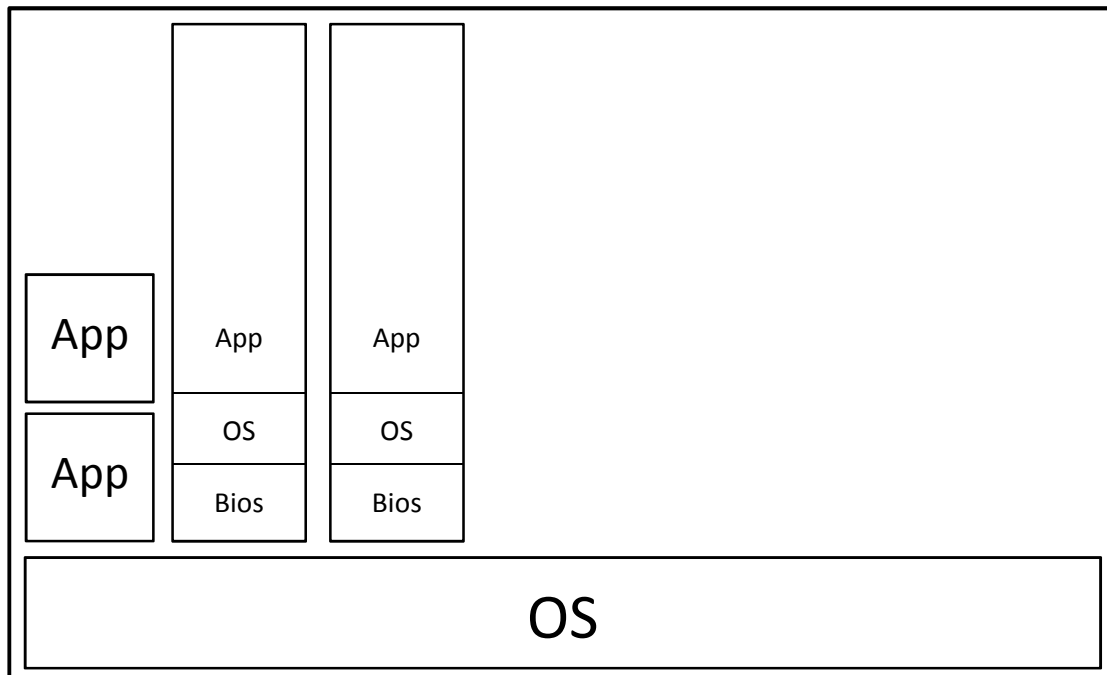


In der DMZ sind private als auch öffentliche IP-Adressen möglich. Für jeden Dienst in der DMZ ist aus Sicherheitsgründen immer nur 1 Computer zuständig. Aus Sicherheitsgründen wird auch der Ein- und Ausgehende Verkehr auf verschiedene Rechner geteilt.

IP-Fire, IP-Cop



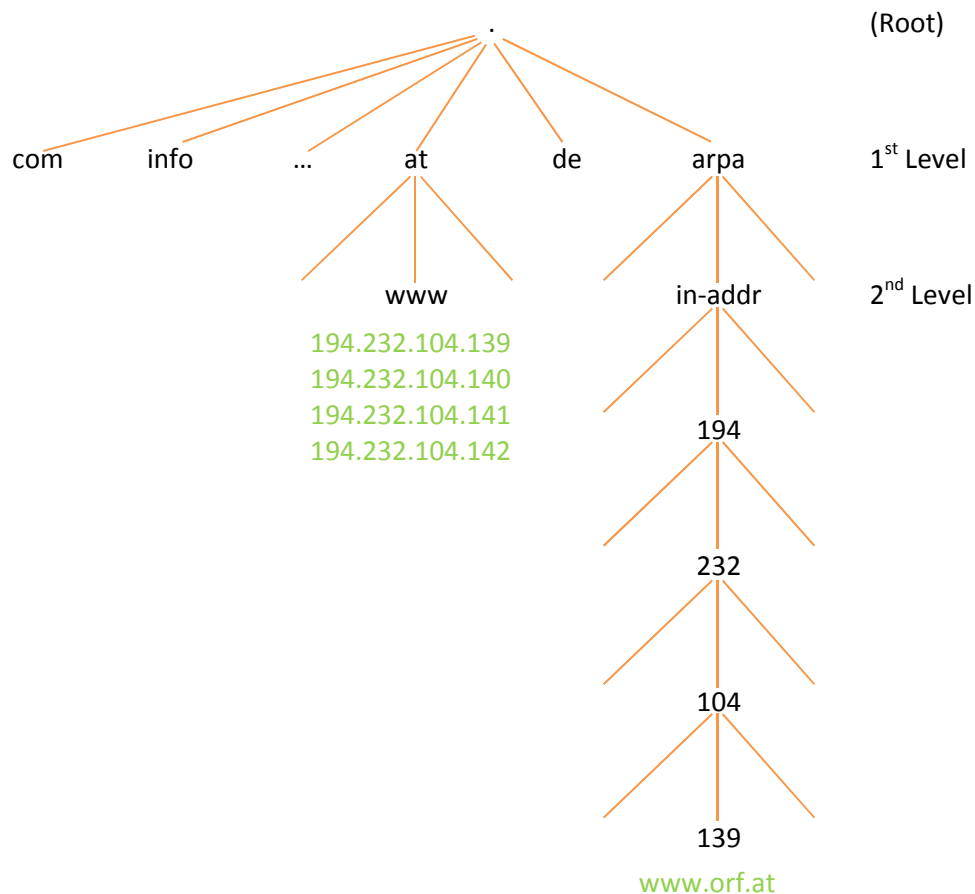
virtuelle Maschinen



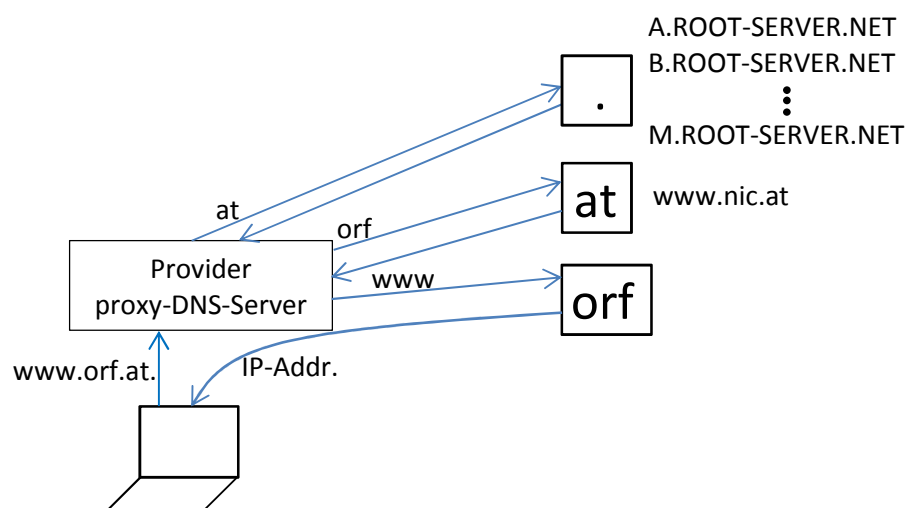
DNS

(Domain Name Service)

Mittels DNS erfolgt die Umsetzung von Namen auf IP-Adressen.



Die Umsetzung einer IP-Adresse auf einen Namen nennt man Reverse-DNS. Bei first-Level-Domains gibt es 2-buchstabige Domainnamen, das sind länderspezifische Domainnamen, alle anderen sind generische Domains. Die Root-Server stehen weltweit verteilt (Ausfall und Lastverteilungsgründe), wobei hinter jeder Adresse mehrere Server stehen.



Die Antworten auf die Anfragen werden zwischengespeichert, wie lange die Antwort gültig ist, wird vom Server mitgesandt. Die Zeit wird möglichst hoch gewählt, um am Server so wenig wie möglich Anfragen (Last), aber auch so kurz wie möglich, um Änderungen im Netzwerk in vernünftiger Zeit durchführen zu können.

Felder in der DNS-Antwort:

A	IPv4
AAAA	IPv6
MX	Mailserver

Der weg zu eigenen DNS:

1. ein eigener Name
2. mind. 2 DNS-Server
3. Geld

Zuständig für die TLD-Namensvergabe: **IANA.org** / **ICANN.org** / **RIPE.net**

Aufgaben:

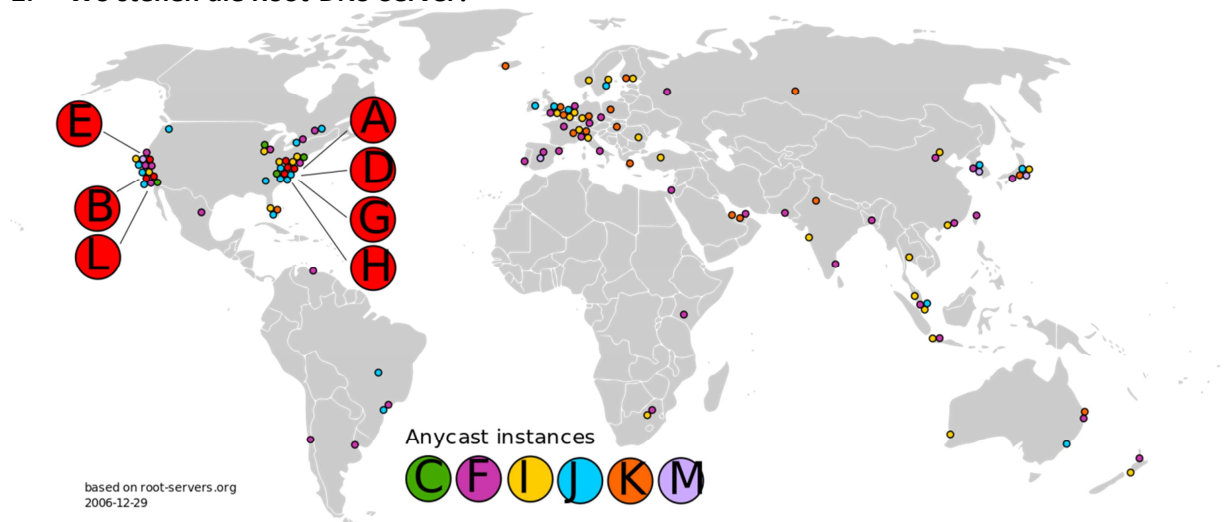
1. Welche First-Level-Domains gibt es?
Wer sind die Verwalter davon?
Wie viele DNS Server, wo stehen sie?
2. Wo stehen die Root-DNS-Server?
3. Voraussetzungen für eigene Domain, was kostet sie?
4. trace-route probieren
5. nslookup probieren

1.a Welche First-Level-Domains, wer sind die Verwalter?

Österreich	.at	NIC (Network Information Center)
Deutschland	.de	DENIC (DEutsches Network Information Center)
Vatikanstadt	.va	Heiliger Stuhl
Königreich Tonga	.to	Tonic (Tongan Network Information Center)
Tuvalu	.tv	VeriSign
Europäische Union	.eu	EURid

1.b Wie viele Root-DNS-Server, wo stehen sie?

a.root-servers.net	198.41.0.4, 2001:503:ba3e::2:30	VeriSign, Inc.
b.root-servers.net	192.228.79.201	University of Southern California
c.root-servers.net	192.33.4.12	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	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	US Department of Defence (NIC)
h.root-servers.net	128.63.2.53, 2001:500:1::803f:235	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, 2001:500:3::42	ICANN
m.root-servers.net	202.12.27.33, 2001:dc3::35	WIDE Project

2. Wo stehen die Root-DNS-Server?

3. Voraussetzungen für eigene Domain, was kostet sie?

TLD-spezifisch

.at:	Vergabe uneingeschränkt	10€ / Jahr
.eu:	Wohnsitz oder Niederlassung in der EU	6,50€ / Jahr
.de	Vergabe uneingeschränkt	4,20€ / Jahr

4. Trace-Route (CMD-Befehl "tracert")

Routenverfolgung zu neonclass.at [84.200.213.28] über maximal 30 Abschnitte:

1	1 ms	1 ms	1 ms	pptp-206-064.pptp.htbl.at [193.170.206.64]
2	1 ms	2 ms	3 ms	inside2.htl-hl.ac.at [193.170.206.17]
3	4 ms	7 ms	4 ms	193.170.206.244
4	18 ms	16 ms	31 ms	host193171231017.asn-noe.ac.at [193.171.231.17]
5	19 ms	16 ms	24 ms	gigabitethernet1-0-7.stpoelten1.aco.net [193.171.24.89]
6	13 ms	9 ms	8 ms	vlan376.wien21.aco.net [193.171.24.85]
7	34 ms	41 ms	62 ms	bundle-ether-21-73.core21.aco.net [193.171.23.42]
8	32 ms	28 ms	43 ms	vix.vie.at.tnib.net [193.203.0.136]
9	27 ms	21 ms	21 ms	ae1-309-cr1.ffm.de.tnib.net [81.92.175.14]
10	142 ms	65 ms	42 ms	decix.accelerated.de [84.200.230.1]
11	22 ms	37 ms	21 ms	fra4.xe-0-1-0.accelerated.de [84.200.230.81]
12	30 ms	36 ms	28 ms	neonMedia.eu [84.200.213.28]

Routenverfolgung zu wikipedia.org [208.80.154.224] über maximal 30 Abschnitte:

1	813 ms	784 ms	1007 ms	pptp-206-064.pptp.htbl.at [193.170.206.64]
2	1079 ms	1013 ms	1273 ms	inside2.htl-hl.ac.at [193.170.206.17]
3	511 ms	387 ms	271 ms	193.170.206.244
4	195 ms	196 ms	167 ms	host193171231017.asn-noe.ac.at [193.171.231.17]
5	21 ms	50 ms	95 ms	gigabitethernet1-0-7.stpoelten1.aco.net [193.171.24.89]
6	7 ms	6 ms	9 ms	vlan376.wien21.aco.net [193.171.24.85]
7	282 ms	1118 ms	501 ms	bundle-ether-21-73.core21.aco.net [193.171.23.42]
8	1153 ms	838 ms	685 ms	vix2.swip.net [193.203.0.105]
9	2504 ms	720 ms	194 ms	wen3-core-2.bundle-ether3.tele2.net [130.244.74.244]
10	97 ms	96 ms	114 ms	cbv-core-4.bundle-ether5.tele2.net [130.244.71.52]
11	138 ms	117 ms	201 ms	ash1-peer-1.ae0-unit0.tele2.net [130.244.38.219]
12	1619 ms	662 ms	918 ms	130.244.6.243
13	145 ms	134 ms	125 ms	text-lb.eqiad.wikimedia.org [208.80.154.224]

5. DNS-Lookup (CMD-Befehl "nslookup")

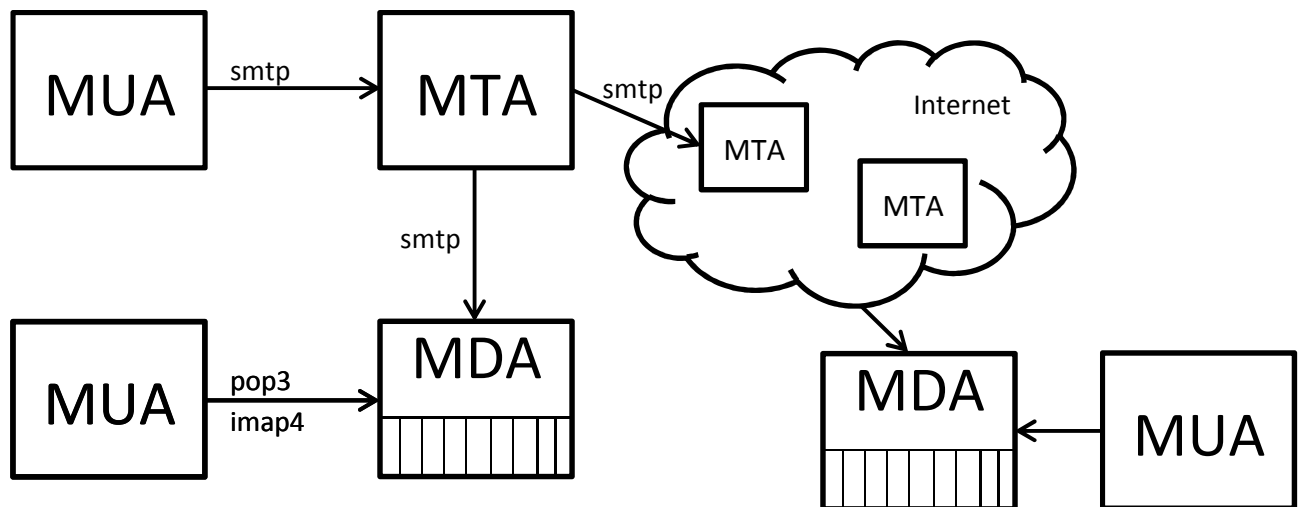
```

Server: node3.intra
Address: 172.16.1.3

Nicht autorisierende Antwort:
Name: google.com
Addresses: 2a00:1450:4001:c02::65
          173.194.70.139
          173.194.70.100
          173.194.70.101
          173.194.70.102
          173.194.70.113
          173.194.70.138

```

e-mail



Problem Spam (unerwünschte Werbeemail): Grund ist, dass jeder MTA von jedem im Netz E-Mails annimmt und weitersendet (Früher absichtlich so gewollt). Zur Abwehr nehmen Provider nur E-Mails von eigenen Kunden an. Bei E-Mails gibt es Text-Mails und Mails im S/MIME-Format. S/MIME sind HTML-Mails mit der zusätzlichen Möglichkeit der Verschlüsselung und Signatur.

smtp

```

HELO      Name
MAIL      From: ....
RCPT      TO: ....
DATA
  From: ....
  To: ....
  Subject: ....
  (Leerzeile)
  Text
  .
  
```

HELP	Gibt Hilfe zurück
RSET	Bricht die Verbindung ab
NOOP	No Operation, damit Server die Verbindung nicht abbricht
QUIT	Schließt die Verbindung

POP3

Port: 110

POP3 wird zum Empfangen von E-Mails verwendet. Im Gegensatz zu IMAP4 können E-Mails nicht als gelesen markiert oder verschoben werden.

Befehle:

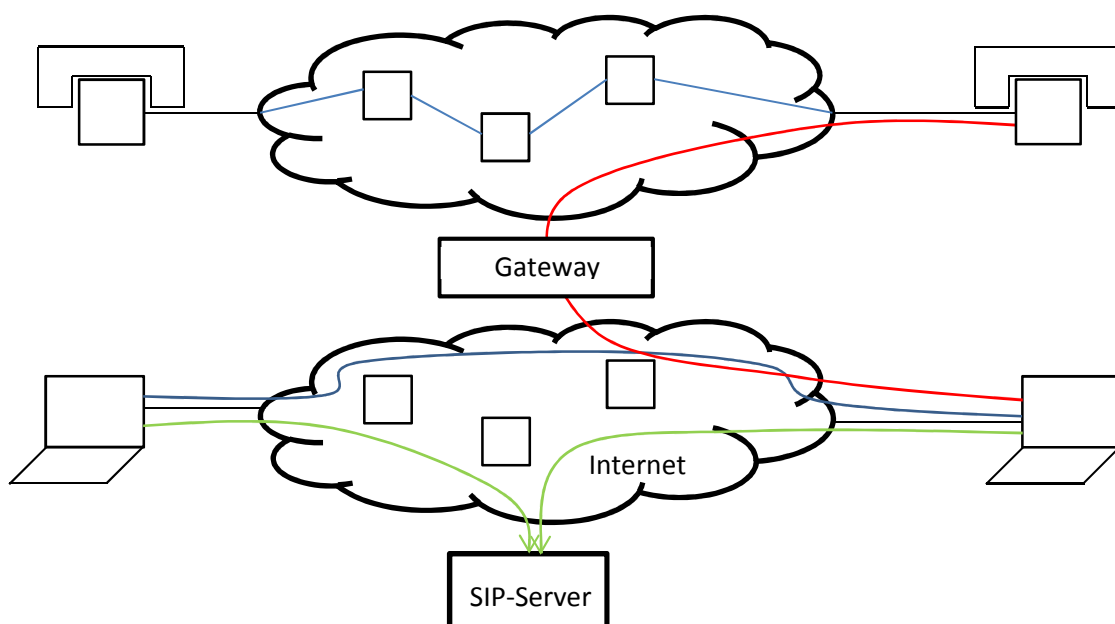
USER	Name	
PASS	Passwort	
STAT		(liefert die Anzahl der Mails und die Größe der Mailbox)
LIST		(liefert eine Liste der Mails)
RETR	E-Mail-Nummer	(zeigt eine E-Mail an)
DELE	E-Mail-Nummer	(löscht eine E-Mail)
NOOP		(erhält die Verbindung aufrecht)
RSET		(Verbindung wird zurückgesetzt)
QUIT		(schließt die Verbindung)

Aufgaben:

1. Hamster installieren, User anlegen
2. E-Mail senden / empfangen mit E-Mail Client
3. Senden mit telnet, empfangen mit E-Mail Client
4. Senden mit E-Mail Client, empfangen mit telnet

VoIP

SIP H.323



Zeit bei VoIP:

Paket	Laufzeit	Jitter-Buffer	
10 ... 30ms	?	?	= Gesamtzeit

Die Gesamtzeit für eine gute Sprachverbindung sollte kleiner 200ms sein. Beim klassischen Telefon ist sie ca. 5 ms, bei Handys bis zu 800ms.

Die Sprache wird in Pakete von 10 ... 30 ms geteilt und über das Netzwerk gesandt. UDP garantiert weder die Reihenfolge noch weiß man ob und wann die Pakete ankommen, dies wird erst durch den Jitter-Buffer am Zielsystem gewährleistet.

Der Jitter-Buffer sollte möglichst klein sein um die Sprachverzögerung möglichst gering zu halten aber groß genug um möglichst alle Laufzeitdifferenzen auszugleichen.

Es gibt verschiedene Sprachkodex mit unterschiedlicher Qualität und unterschiedlicher Bitrate

Aufgaben:

SIP:

1. Anbieter
2. Geräte
3. Asterisk

PGP

1. Zertifikat erstellen
2. File verschlüsseln
3. File digital Signieren