

# Einführung in Netzwerksicherheit

1. Zielstellungen
2. Grundlagen der Verschlüsselung
3. Sichere Kommunikationsdienste
  - 3.1 PGP
  - 3.2 SSL/TLS
  - 3.3 Untere Schichten
4. Sicherheit auf dem Internet Layer
  - 4.1 IPSec
5. Firewalls - schichtenübergreifendes Paketfiltern



# Zum Inhalt

In diesem Kapitel geht es um Grundelemente und Verfahren der Netzwerksicherheit. Hierfür betrachten wir zunächst die kryptographischen Grundlagen, um danach ihre Einsatzmöglichkeit und typischen Anwendungen auf den verschiedenen Netzwerkschichten kennen zu lernen.

Das zugehörige Kapitel im Tannenbaum ist 8, im Meinel/Sack wird Sicherheit bei den jeweiligen Schichten diskutiert.



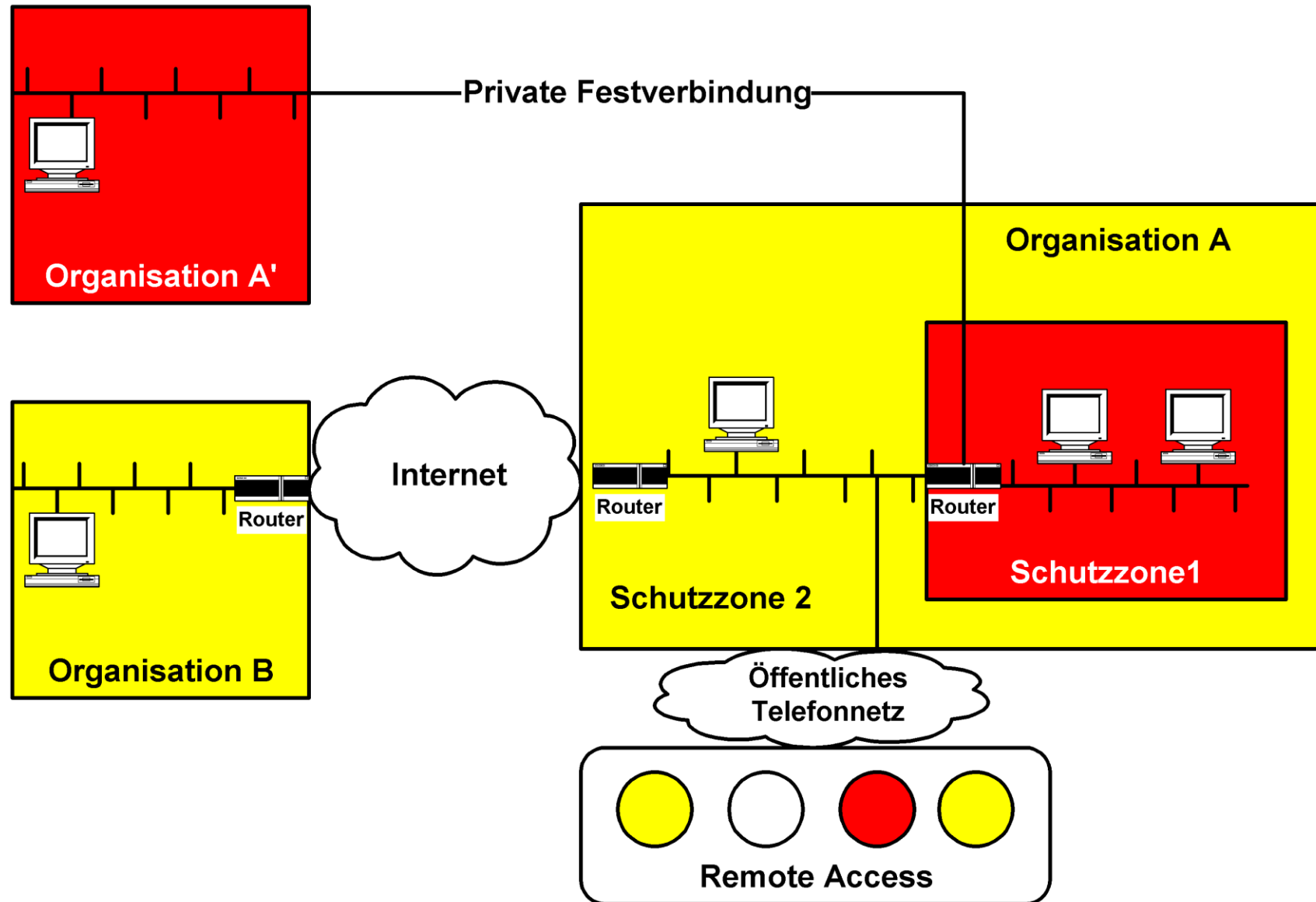
# 1. Sicherheitsbedrohungen im Netzwerk

- Ausspähen von Daten
- Manipulation von Daten
- Computer und Systemsabotage
- Beeinträchtigung der Verfügbarkeit
- Analyse von Kommunikationsprofilen
- ...

**Problem:** Die physikalische Kontrolle über den Netzwerktransport erhält man nur teuer oder häufig gar nicht.



# 1. Verteilte Szenarien



# 1. Schutzziele

- **Vertraulichkeit** (secrecy) – Geheimhaltung von Daten
- **Unversehrtheit** von Daten (integrity)
- **Authentizität** (authenticity) – Nachweis der ursprünglichen Unversehrtheit
- **Verbindlichkeit** (accountability) – Sicherstellung des (unbestreitbaren) Empfangs
- **Verfügbarkeit** von Ressourcen (availability)
- **Anonymität** von Benutzern und Kommunikation (anonymity)
- **Einbruchssicherheit** der Endsysteme (intrusion protection)



# 1. Netzwerksicherheit

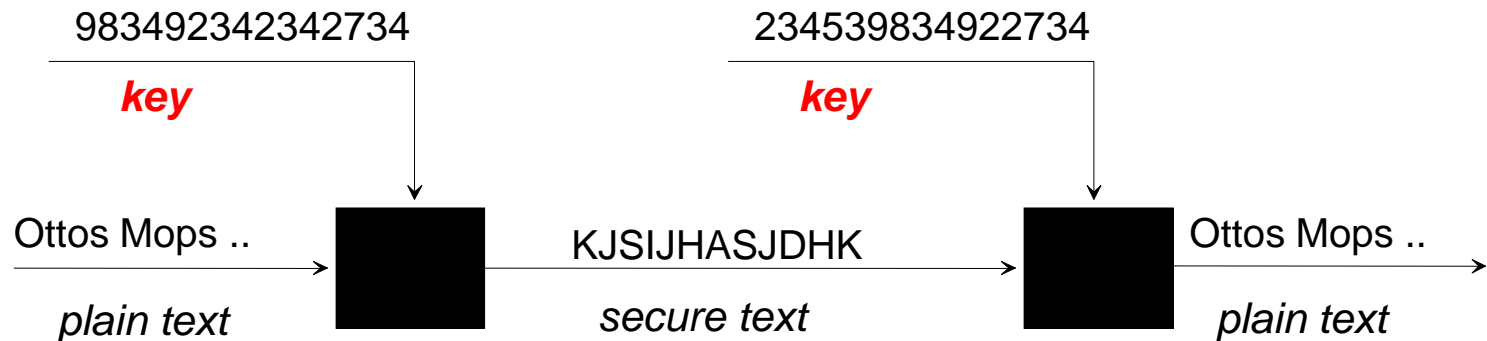
- ▶ Sicherheit im Rechnernetz
- ▶ Sicherheit vor unerwünschten Manipulationen aus dem Netz
  
- ▶ Typische Schutzmechanismen:
  - ▶ Kryptographie
  - ▶ Firewalls
  - ▶ Intrusion Detection

**Caveat: Das Internet operiert im Ende-zu-Ende Paradigma!**



# 2. Grundlagen: Verschlüsselung

Aufgabe: Erreichen von Schutzziele durch kryptographische Verfahren



## Symmetrische Verschlüsselung:

- Ver- und Entschlüsseln mit demselben „Geheimnis“
- Schlüssel bleibt geheim

## Asymmetrische Verschlüsselung:

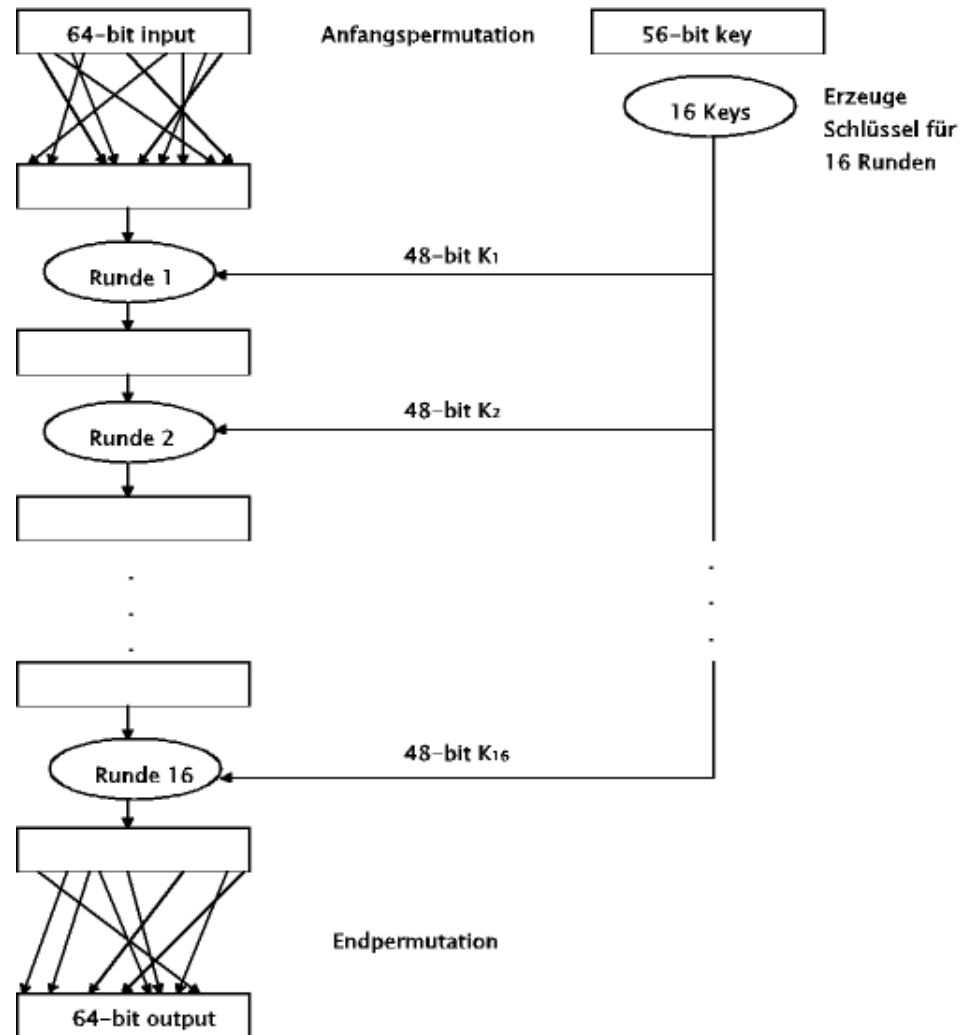
- Unterschiedliche Schlüssel zum Ver- und Entschlüsseln
- Ein Schlüssel kann veröffentlicht werden



# 2. Symmetrische Verschlüsselung

## Beispiel: DES

- Private key Methode
- Klassisch, performant
- Key Austausch zur Laufzeit
- Initialer Seed wird (out of band) benötigt
- Problem:  
Keine Methode der Signatur
- Authentication:  
Challenge-Response-Scheme





# 2. Asymmetrische Verschlüsselung

- Public key Methode  
(Rivest, Shamir, Adleman 1978)
- Berechnungen numerisch  
komplex (lange Schlüssel!)
- Individuelle Schlüsselerzeugung
- Öffentlicher Schlüsselaustausch
- Erlaubt die Absender-  
authentifizierung
- Erlaubt Signierung zur  
Integritätsattestierung

## RSA-Algorithm

**p, q large prime number,  $n = p * q$**

**let e, d and k with**

$$e * d = k * (p - 1) * (q - 1) + 1$$

**Number Theory: for every m**

$$(m^{**e})^{**d} \text{ mod } n = m$$

**m: message to send**

**e: Encryptor (public key)**

**d: Decryptor (private key)**



## 2. Key Agreement: Diffie-Hellmann

**Problem:** Zwei einander unbekannte Teilnehmer (A & B) wollen einen gemeinsamen geheimen Schlüssel über einen öffentlichen Kanal verhandeln

**Ansatz:** Spontane Schlüsselerzeugung mithilfe der ‚Public Key Kryptographie‘

**Methode:** Diffie-Hellmann “New Directions in Cryptography” (1976)

**Einschränkung:** Gegenseitige Authentifizierung bleibt unberücksichtigt – möglich z.B. durch Public Key Infrastructure



## 2. Diffie-Hellmann Algorithm

Let  $p$  be a sufficiently large prime,

$$g : g^n \bmod p = p \text{ for some } n,$$

$p$  and  $g$  publicly available.

Then:

1. A chooses  $0 \leq a \leq p - 2$  at random and sends  $c := g^a$  to B
2. B chooses  $0 \leq b \leq p - 2$  at random and sends  $d := g^b$  to A
3. A computes the shared key  $k = d^a = (g^b)^a$
4. B computes the shared key  $k = c^b = (g^a)^b$

The strength of the algorithm relies on the secrets  $a$  and  $b$ , which are discrete logarithms  $\bmod p$



## 2. Public Key Infrastructures (PKI)

**Problem:** Wie können Teilnehmer überprüfen, dass die Schlüssel tatsächlich zur richtigen Gegenstelle gehören?

**Ansatz:** Kryptographische Bindung wird durch eine Instanz des Vertrauens bestätigt: Certificate Authority (CA)

**Methode:** Die CA stellt ein Zertifikat aus, welches

- ▶ die ID des Schlüsselinhabers (den Namen des Subjekts)
- ▶ den öffentlichen Schlüssel des Inhabers
- ▶ die eigene ID (den Namen der CA)
- ▶ ggfs. weitere Metainformationen

enthält und **von der CA signiert** wird.



# 2. Zertifikat im Browser

Certificate Viewer: "www.google.de"

General Details

**This certificate has been verified for the following uses:**

- SSL Client Certificate
- SSL Server Certificate

---

**Issued To**

Common Name (CN)	www.google.de
Organization (O)	Google Inc
Organizational Unit (OU)	<Not Part Of Certificate>
Serial Number	7C:7F:32:37:41:07:B6:31

**Issued By**

Common Name (CN)	Google Internet Authority G2
Organization (O)	Google Inc
Organizational Unit (OU)	<Not Part Of Certificate>

**Period of Validity**

Begins On	Mittwoch, 18. Mai 2016
Expires On	Mittwoch, 10. August 2016

**Fingerprints**

SHA-256 Fingerprint	08:AA:B4:0C:4D:52:F3:2E:9B:9F:EE:BE:6A:7E:5B:A8:F0:9F:FF:D9:79:63:DE:BC:0F:31:8B:31:96:B9:86:78
SHA1 Fingerprint	A9:A8:8A:BE:10:69:04:1F:36:32:C2:DF:7C:32:48:E8:75:53:C7:F6

Close

Certificate Viewer: "www.google.de"

General Details

**Certificate Hierarchy**

- GeoTrust Global CA
  - Google Internet Authority G2
    - www.google.de

---

**Certificate Fields**

- Issuer
  - Validity
    - Not Before
    - Not After
  - Subject
    - Subject Public Key Info
      - Subject Public Key Algorithm
      - Subject's Public Key

---

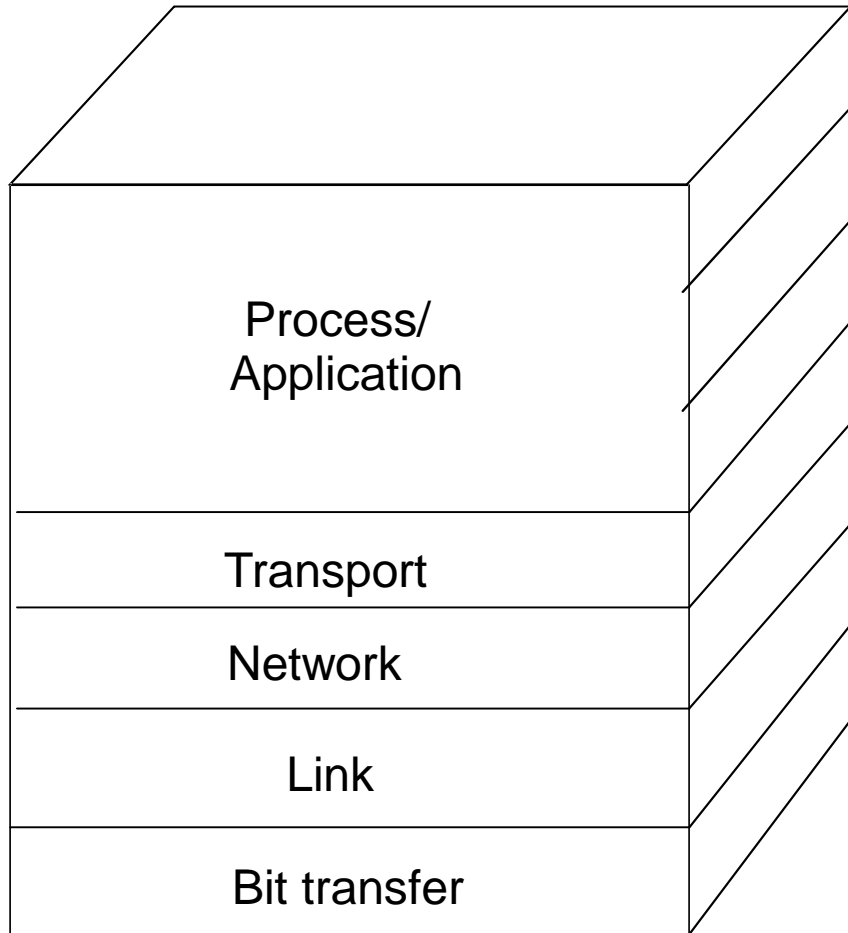
**Field Value**

Modulus (2048 bits):

```
a8 88 3d 1f 30 f1 f9 9a 67 11 a3 97 3f 8c af 39
d6 f2 bd 15 49 d9 17 d2 a2 fc f3 39 ce ed 67 6f
88 50 07 9d 88 3b ca c1 42 e1 d5 72 37 39 e7 09
7b 28 72 30 cc 92 ab b2 97 f9 ac b3 05 03 be 06
bc 47 12 b0 6d 21 8c b0 b6 b6 06 f2 6e 32 11 0f
42 6d ab 67 f7 4a 39 df 55 24 d8 5f 63 53 1c cf
a6 50 e7 dc 78 f9 1b 6a 0e da 0d 8d 80 21 56 6d
49 2a eb 55 24 f7 a7 f4 0b d6 fe 13 a0 eb 3c 61
70 fb ff 0b 0f 6f fa 65 15 28 c0 db db fd 72 20
```

Export...

# 3. Ebenen der Verschlüsselung



Layer 7: Application encryption

Layer 4+: Socket layer security

Layer 3: Network encryption

Layer 2: Logic tunnelling

Layer 1: Line encryption



# 3.1 Anwendungsschicht

**Beispiel:** Pretty Good Privacy (Mail)

**Vorteil:**

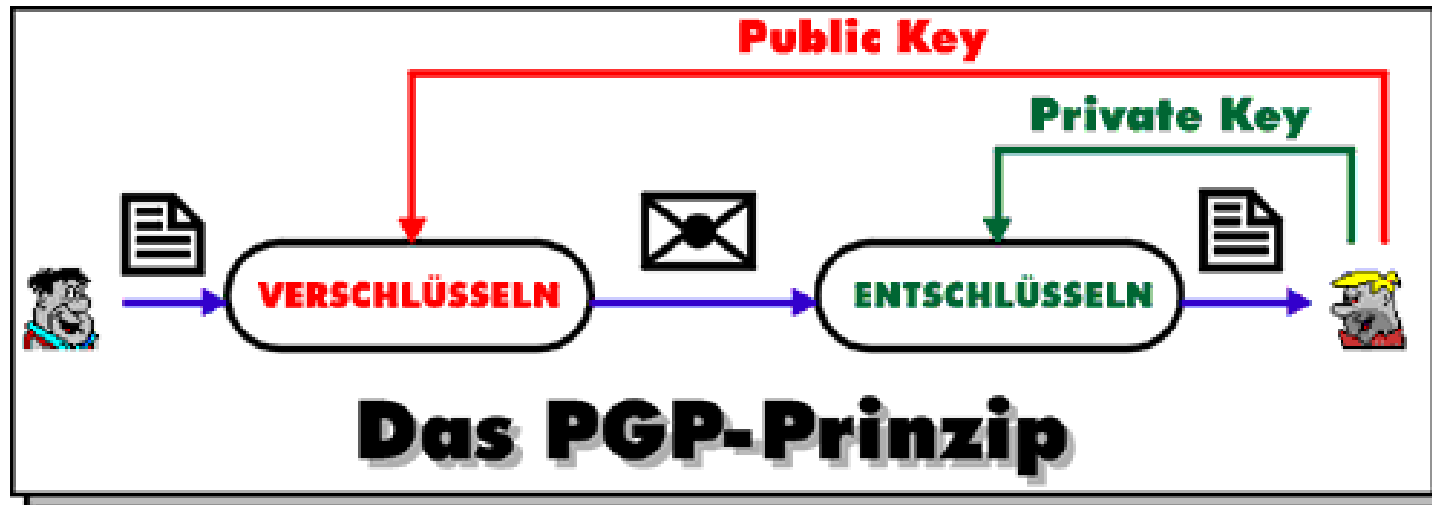
- Geeignet für alle Anwendungsfälle
- Infrastrukturungebunden
- Anwendungsspezifisch optimierbar

**Nachteile:**

- Kommunikationsprofile bleiben sichtbar
- Anwendungsprogramme müssen Methoden implementieren



# 3.1 Beispiel: Pretty Good Privacy



- **Public key basierend:**  
Fred verschlüsselt seine Nachricht mit dem public key von Barney.
- Zur **authentication** hängt Fred eine ‚signature‘ an seine mail.
- Nur Barney kann den Inhalt der mail mit seinem private key entschlüsseln.
- Barney entschlüsselt die Signatur mit dem public key von Fred.



## 3.2 Socket Layer (4+)

**Beispiel:** Secure Socket Layer (SSL/TLS)

**Vorteil:**

- end-to-end Sicherheitsmodell
- Transparent im Hinblick auf Anwendungsdaten
- Einfach integrierbar (secure socket library)

**Nachteile:**

- Kommunikationprofile bleiben sichtbar auf der Transportschicht (einschl. Anwendungsprotokoll)
- Anwendungsprogramme müssen Bibliothek benutzen



## 3.2 Beispiel: SSL/TLS

- ▶ Transport Layer Security: RFC 2246, 5246 (TLSv1.2)
- ▶ Protokoll für verschlüsselten Datenaustausch zwischen unbekanntem Klienten und einem bekannten Server (akzeptiert durch Zertifikat).
- ▶ Public key basierte Session-Initiierung:  
Auf Anfrage sendet der Server seinen public key zum Client.
- ▶ Client erzeugt dann ein pre-shared secret (symmetrischer Schlüssel) und sendet diesen – verschlüsselt mit dem erhaltenen public key – zum Server zurück.
- ▶ Die folgende Kommunikation wird symmetrisch verschlüsselt.



# 3.3 Leitungsverschlüsselung (L 1)

**Beispiel:** Bit-Scrambling, WEP

**Vorteile:**

- Vollständige Informationsverschlüsselung
- Völlig transparent gegenüber höheren Schichten

**Nachteile:**

- Leitungsgebunden, nicht end-to-end
- Erfordert Hardwareunterschützung



# 3.3 Beispiel: WEP/WPA

- Protokoll für die Verschlüsselung von Funkdaten zwischen Access Point und Stationen.
- **Private key based**: AP & STA halten pre-shared secret.
  - Feste Länge: 40 or 104 bits
  - Statisch: kein Schlüsselwechsel ohne Rekonfiguration
- **Authentication**: Challenge (AP) – Response (STA) Schema.
- **Encryption**: RC4 Verschlüsselung (XOR mit pseudo-Zufallsstrom) basierend auf (ungenügend vielfältigen) Initialisierungsvektoren (IV).
- **Verbesserung**: WPA – upgrade auf Temporal Key Integrity Protocol (TKIP) – mindert Defizite durch **verbesserte IV Auswahl** und **re-keying**.



# 3.3 Layer 2: MAC Sicherung + Tunnel

## Beispiele:

- MAC Sicherung: ACLs, 802.1x port authentication
- Tunnel: PPP/PPTP, L2TP (+encryption), ...

## Vorteile:

- Verhindert ARP spoofing + network intrusion
- Transparent zur Netzwerkschicht (nur Tunnel sichtbar)

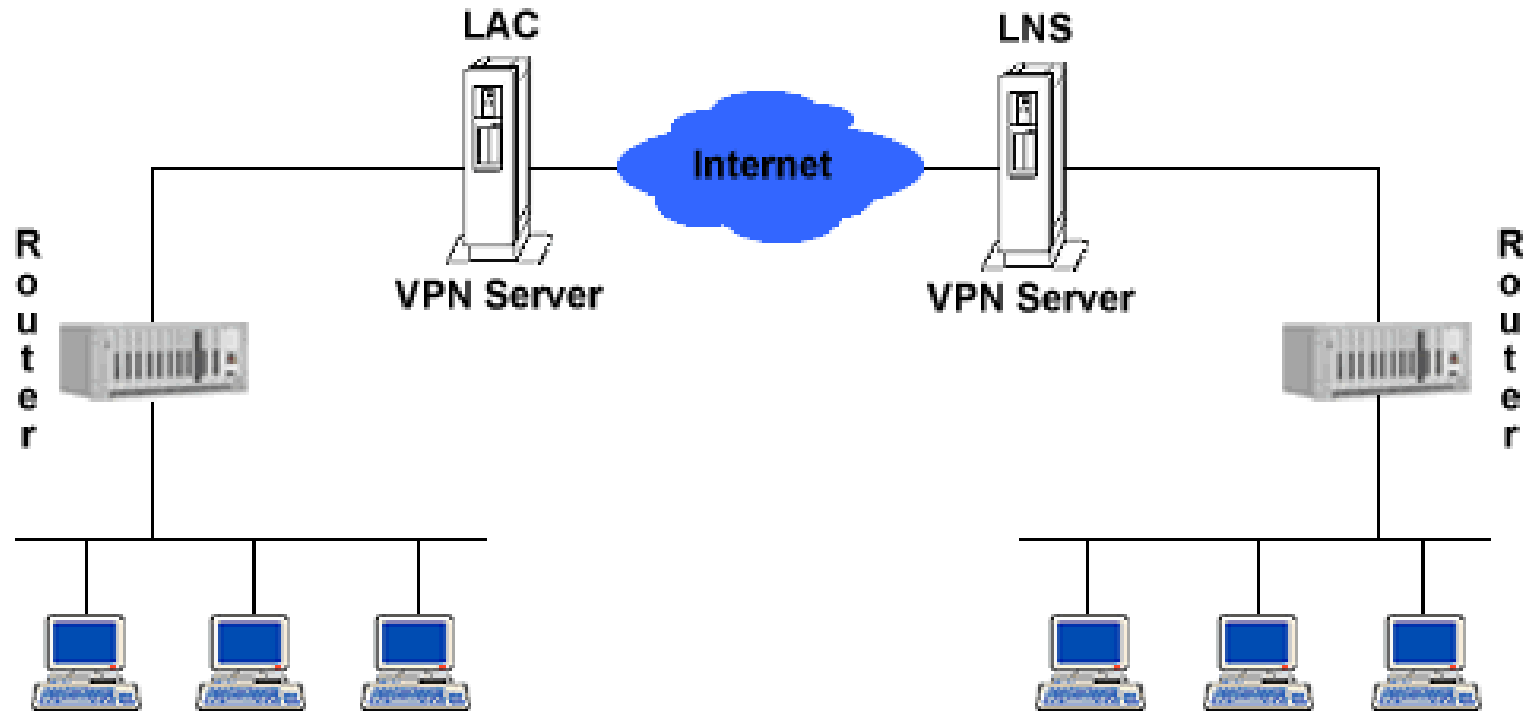
## Nachteile:

- Benötigt Server / Provider Unterstützung
- Begrenzte Skalierbarkeit / Performanz



# Compulsory Tunnel (Carrier / ISP Model)

IP(Message)  
PPP(IP(Message))  
PPP(IP'(L2TP(××××××××)))



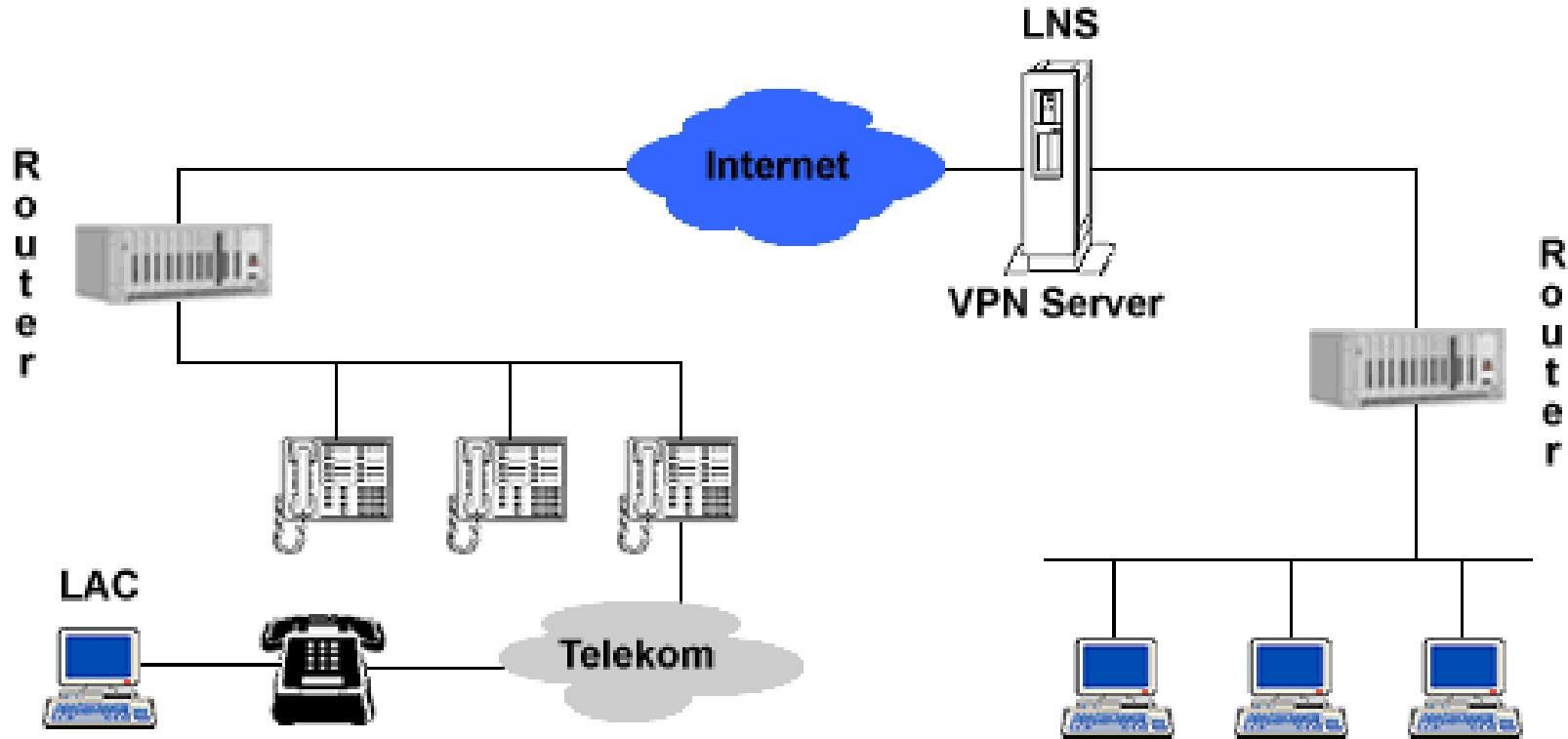
© 2000

Anton Meller , Alexander Zaika



# Voluntary Tunnel (Client Model)

IP(Message)  
 PPP(IP(Message))  
 PPP(IP'(L2TP(××××××××)))



© 2000

Anton Meller , Alexander Zaika



# 4. Internet Layer (IP)

**Beispiel:** Paketverschlüsselung, Adressauthentifikation

**Vorteile:**

- ▶ Transporttransparent
- ▶ Effizient & weitverkehrs-routebar

**Nachteile:**

- ▶ Kommunikationsprofile auf dem IP layer sichtbar

**Ansatz:** IP-in-IP secure tunnelling: IPSec



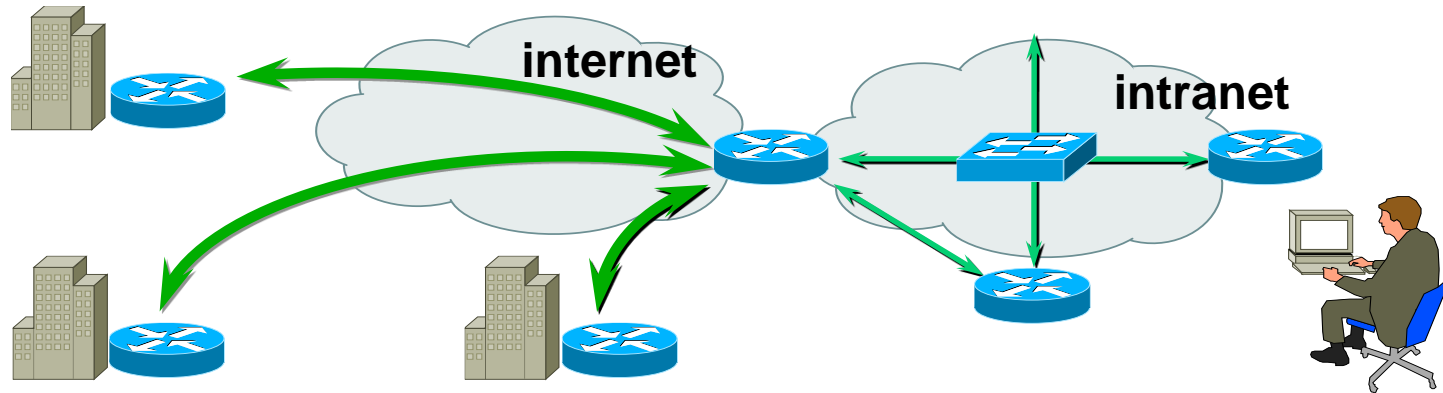


# 4.1 Was ist IPSec?

- Eine Sicherheitsarchitektur
  - Zwei IP Sicherheitsprotokolle
    - Authentication Header (AH)
    - Encapsulation Security Payload (ESP)
  - Internet Key Exchange (IKE)
    - Verhandlung von IPSec security seeds
  - Ein offener Standard (RFC 2401, 4301)
- ⇒ Eine end-to-end Sicherheitslösung auf dem IP layer



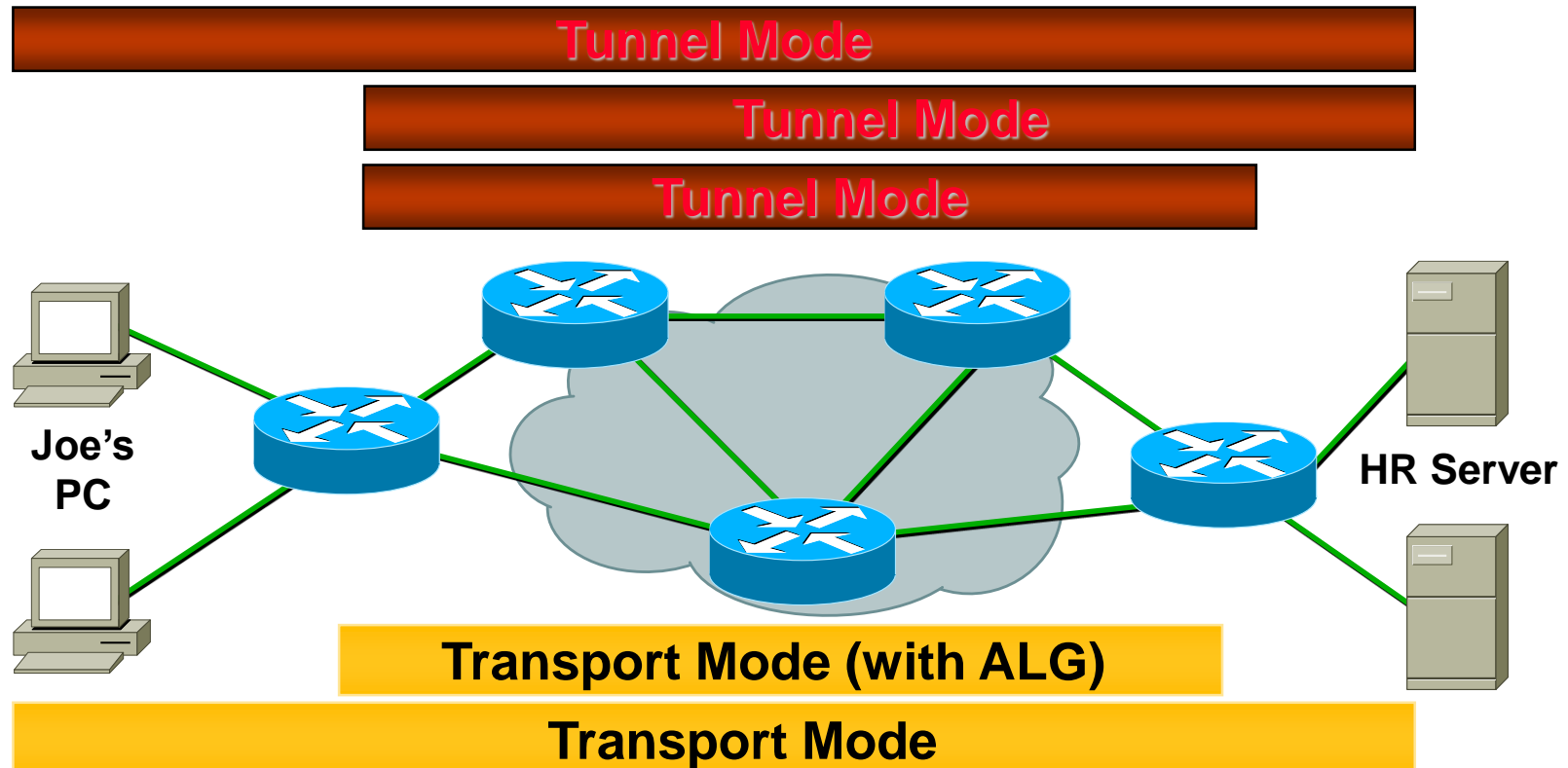
# 4.1 Konzepte von IPSec



- Schützt Datentransfers durch das Internet mittels  
**Authentication, Integrity, Encryption**
- Transparent zu und angepasst an die Netzwerkinfrastruktur
- End-to-end Konzept



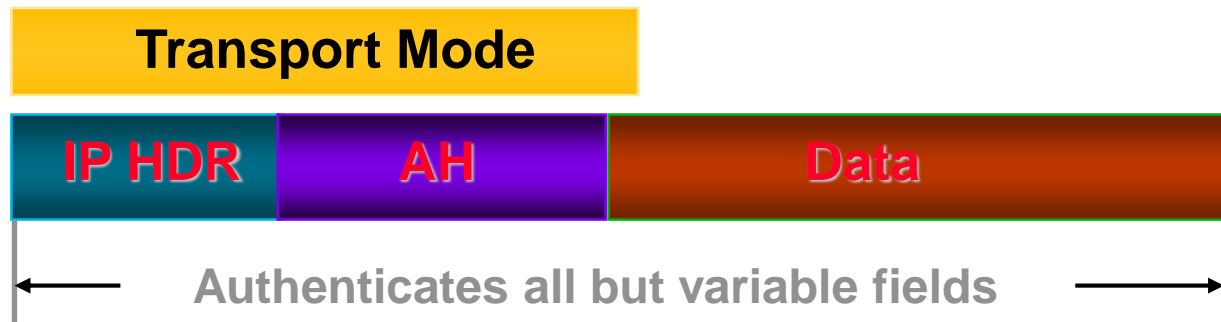
# 4.1 Tunnel und Transport Mode



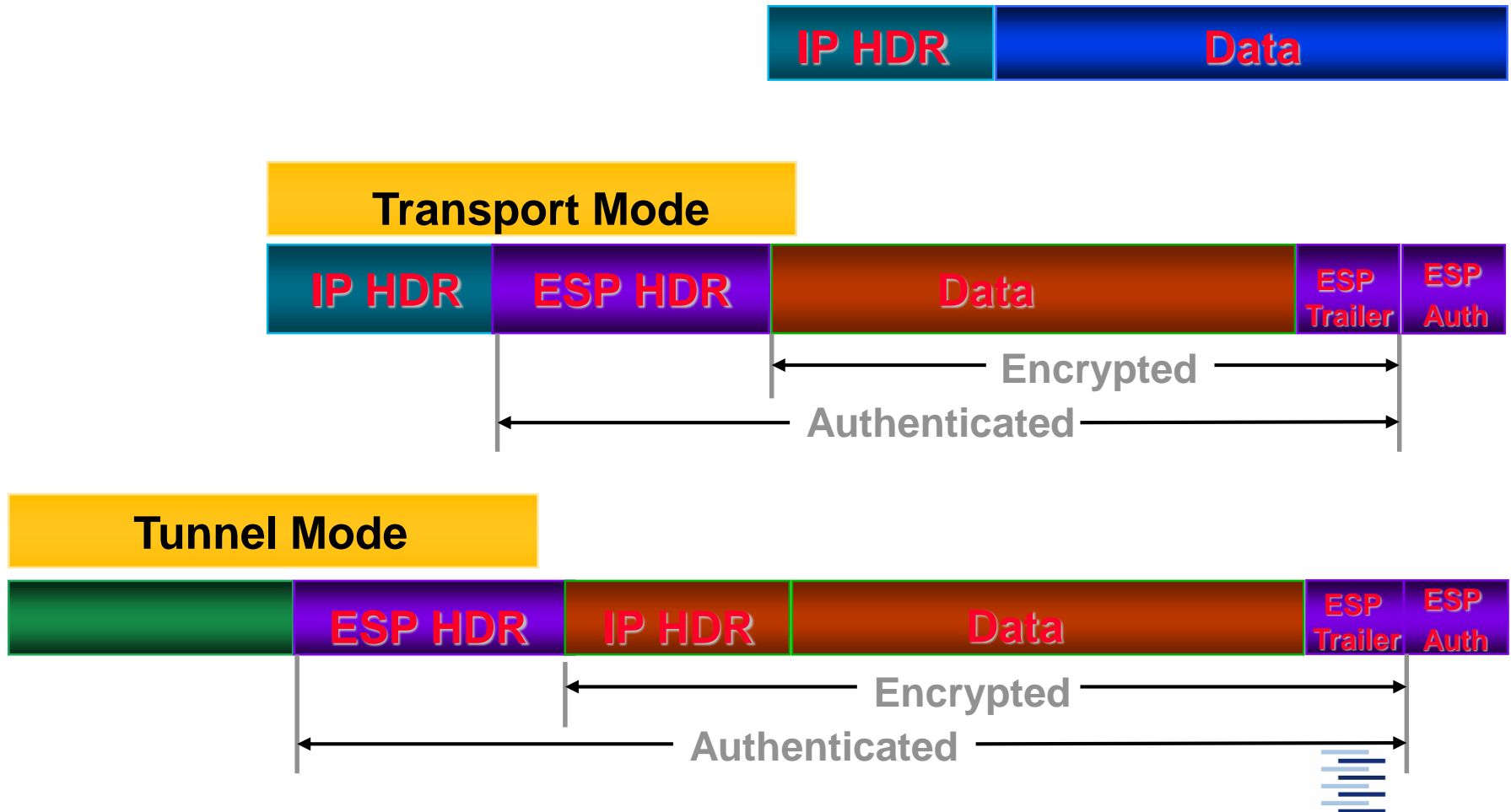
- ▶ Transport Mode End-to-End or via ALG
- ▶ Tunnel Mode for all connection types



# 4.1 IPSec Authentication Header (AH)



# 4.1 Encapsulating Security Payload (ESP)

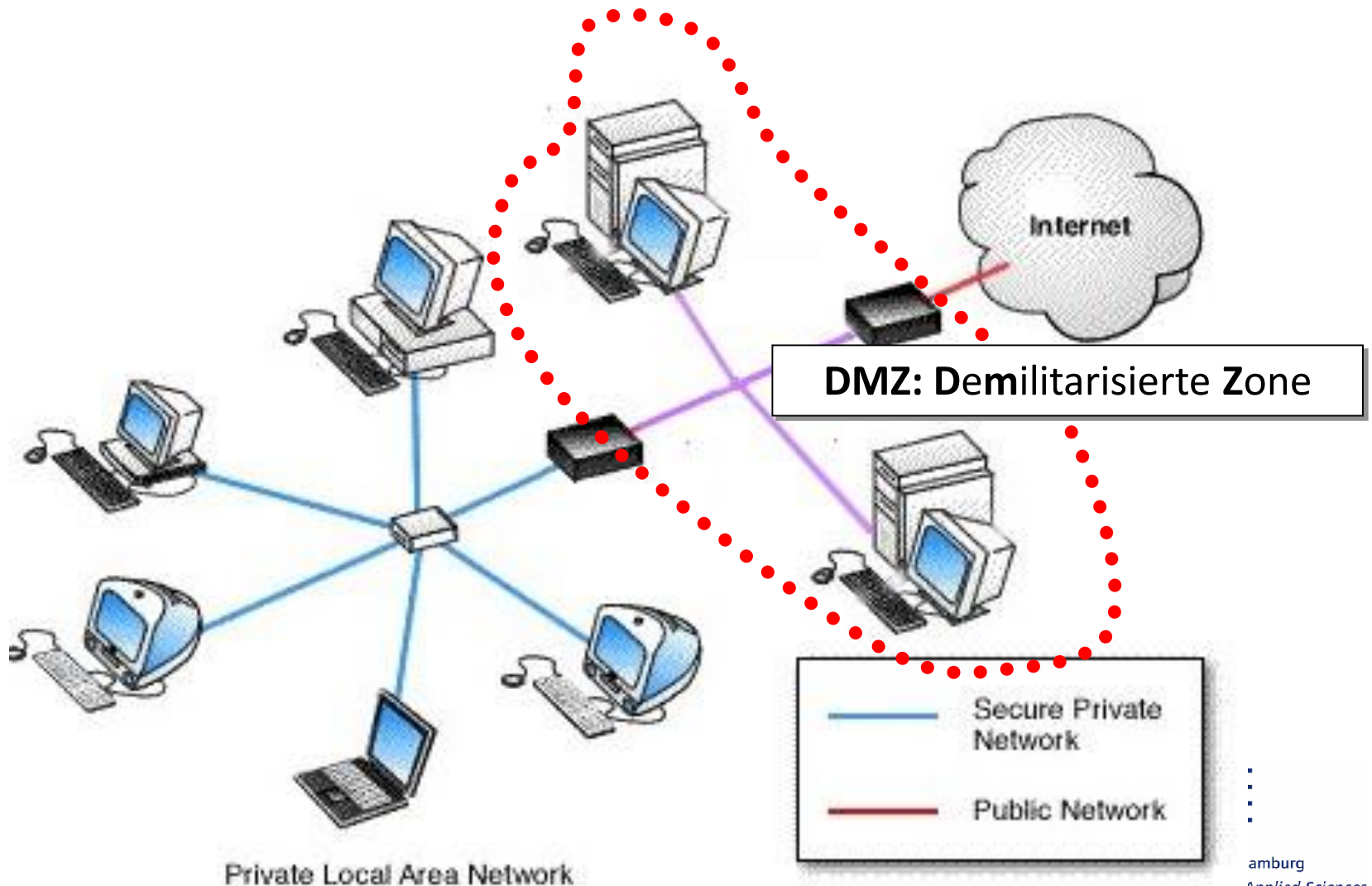


# 5. Firewalls - schichtenübergreifendes Paketfiltern

- ▶ Konzept: Ungewollter Datenverkehr wird an einem Kontrollpunkt blockiert
- ▶ Anwendung von Filterregeln auf Ströme
  - ▶ Quell-/Ziel-IP, -Port + Transport Protokoll – Bsp: „Mail nur via Mailserver“
  - ▶ Kommunikationsrichtung – Bsp: „von außen initiiert“, dies geht für TCP zustandslos: eingehend + ACK=0
  - ▶ Benutzerdefinierte Bitmasken – Bsp: „ICMP Redirect“
- ▶ Kontrollpunkt lokal oder an Netzwerkübergängen
  - ▶ Idee des vertrauenswürdigen Innenbereichs



# 5.1 Firewall Standardarchitektur: abgeschottetes privates LAN

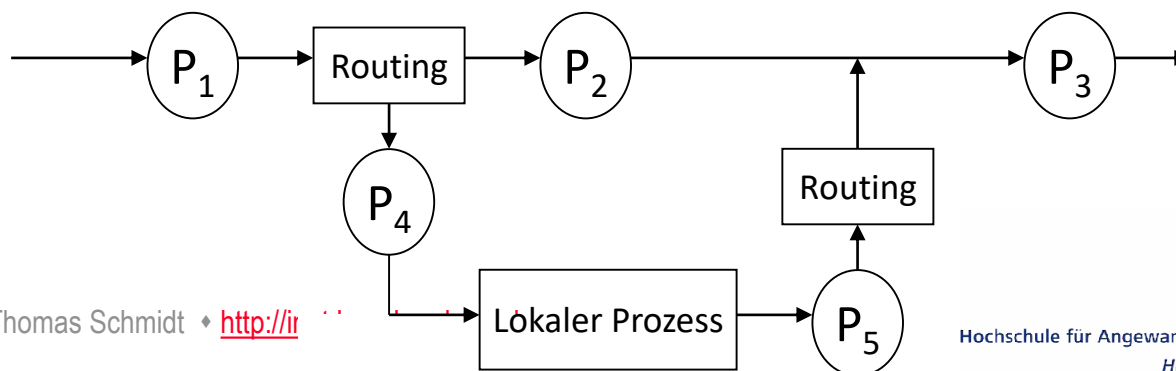


# 5.2 Paketfilter unter Linux: iptables

## - Grundsätzliches

- Pflege von Tabellen für IP-Paket Regeln im Linux Kern
  - mehrere Tabellen: **filter**, **nat**, **mangle** (s.f.F.)
- Regeln besagen wie ein passendes IP-Paket zu behandeln ist:
  - **ACCEPT**: lasse das Paket passieren
  - **DROP**: verwerfe das Paket
  - **QUEUE**: Paket wird zu Nutzer-definierten Chain weitergereicht
  - **RETURN**: beende den Durchlauf in dieser Tabelle

## - Verschiedene Interaktionspunkte von iptables mit einem Paket





# 5.2 Paketfilter unter Linux: iptables

## Drei unabhängige Tabellen

**filter:** (Default Tabelle) mit

- bei  $P_4$ : **INPUT** für Pakete an einen lokalen Prozess
- bei  $P_2$ : **FORWARD** für Pakete durch ein Network Interface
- bei  $P_5$ : **OUTPUT** für lokal generierte Pakete

**nat:** für Network-Address-Translation

- bei  $P_1$ : **PREROUTING** für ankommende Pakete
- bei  $P_5$ : **OUTPUT** für lokal generierte Pakete vor dem Routing
- bei  $P_3$ : **POSTROUTING** für Pakete nach dem Routing

**mangle:** für spezielle Paketveränderungen

- bei  $P_1$ : **PREROUTING** für eingehende Pakete
- bei  $P_5$ : **OUTPUT** für lokal generierte Pakete vor dem Routing

# Zusammenfassung

- Sicherheit im Netz kann auf vielen Schichten erhöht werden
- Die Entscheidung für eine Technologie benötigt eine sorgfältige Problemanalyse
- Der Grad der erreichten Sicherheit wird bestimmt durch die Konzepte und Algorithmen, die Schlüsselstärke und die Managementqualität
  - Achtung: die Komplexität von schichtenübergreifenden Verfahren wächst sehr schnell
- So etwas wie "sicher" gibt es nicht, nur "sicherer"



# Literatur

- William Stallings: *Cryptography and Network Security*, 6<sup>th</sup> Ed., Pearson, 2013.
- Dieter Gollmann: *Computer Security*, 3<sup>rd</sup> Ed., Wiley, 2011.
- Hans Delfs, Hartmut Knebl: *Introduction to Cryptography*, Springer, 2002.
- Christof Paar, Jan Pelzl: *Understanding Cryptography*, Springer 2010.
- Claudia Eckert: *IT Sicherheit*, 9<sup>th</sup> Ed., Oldenbourg Verlag, 2014.
- Internet Standards at: [www.rfc-editor.org](http://www.rfc-editor.org).



# Selbsteinschätzungsfragen

1. Worin besteht der Unterschied zwischen „Vertraulichkeit“ und „Authentizität“?
2. Wie kann eine gegenseitig Authentifizierung mithilfe symmetrischer bzw. asymmetrischer Kryptographie durchgeführt werden?
3. Wie erlangt man über SSL/TLS einen symmetrisch verschlüsselten Kanal ohne ‚pre-shared secret‘?
4. Wie kann man mit einer Firewall alle SMTP-Pakete (Port 25) auf einen bestimmten Mailserver umlenken? Wie ginge das einfacher, besser und richtiger?

