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

Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

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.

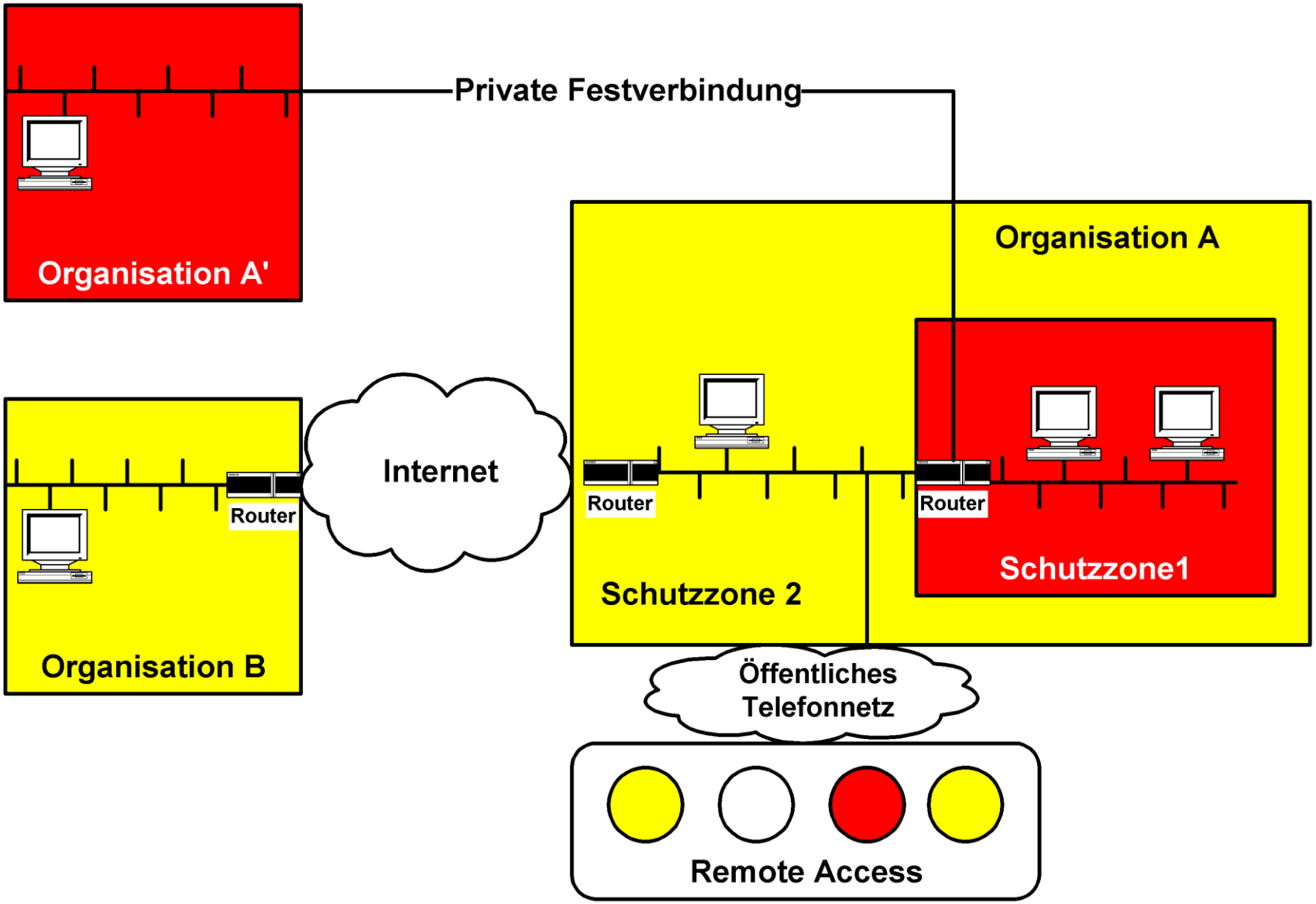
Das zugehörige Kapitel im Tannenbaum ist 8, bei Bonaventure finden Sie die Grundlagen in Teil 1 und TLS in Teil 2.

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

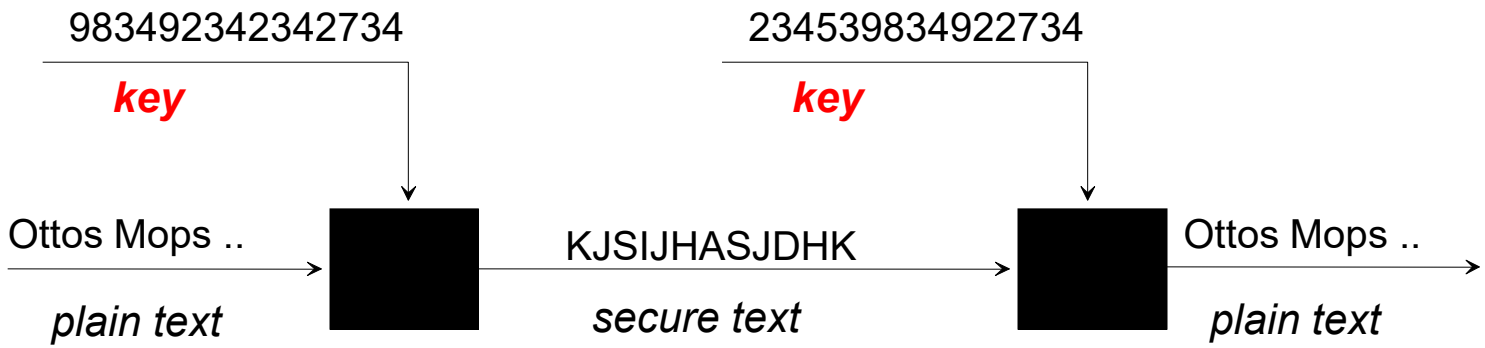
- Sicherheit im Rechnernetz
- Sicherheit vor unerwünschten Manipulationen aus dem Netz

- Typische Schutzmechanismen:
  - Kryptographie
  - Firewalls
  - Intrusion Detection

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

2. Grundlagen: Verschlüsselung

Aufgabe: Erreichen von Schutzzielen durch kryptographische Verfahren



Symmetrische Verschlüsselung:

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

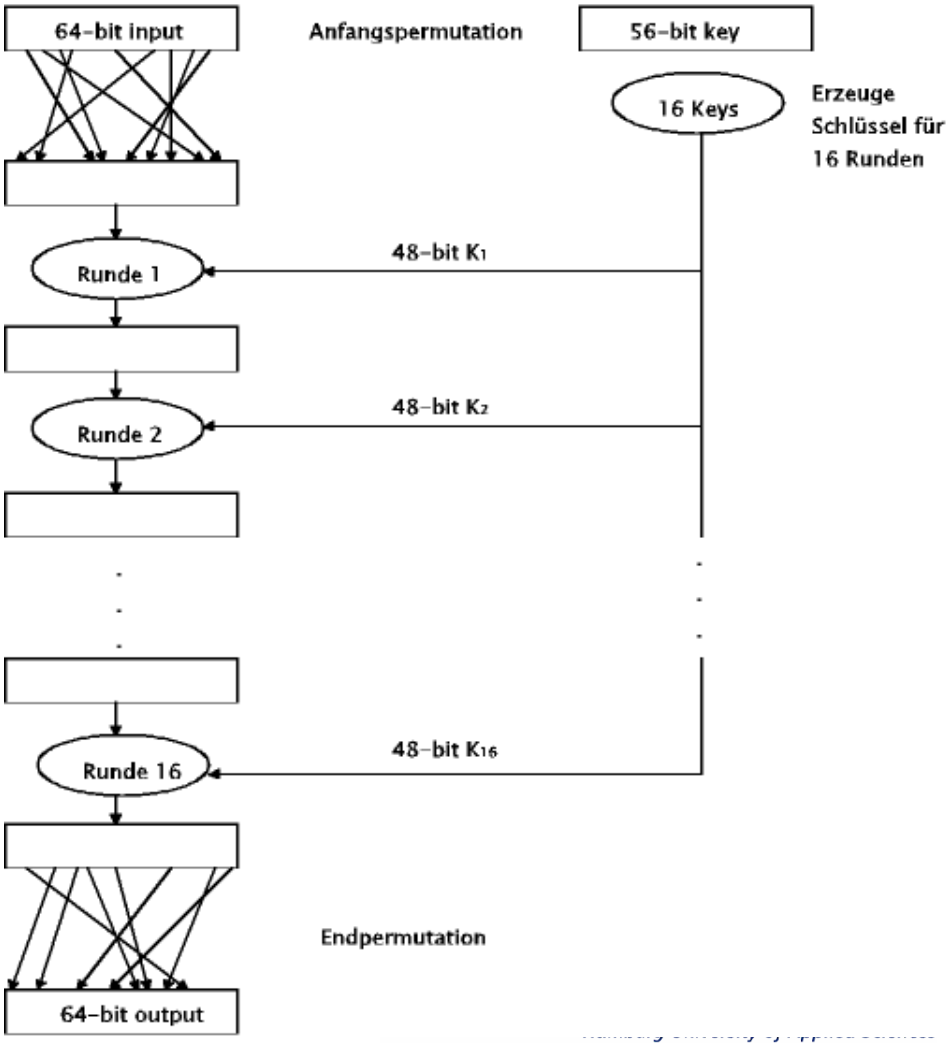
Asymmetrische Verschlüsselung:

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

2. Symmetrische Verschlüsselung

Historisches 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



Symmetrisch: Advanced Encryption Standard (AES)

- Symmetrische Blockchiffre mit variabler Block- und Schlüssellänge (typisch: 128 – 256 Bit)
- NIST Standard (2000): Rijndael Algorithmus
- Hocheffizient, auf SmartCards lauffähig
- Läuft wie DES in Runden (10-14) ab, allerdings sind diese Runden mehrschichtig:
  1. Substitution: Byte-Ersetzung per Substitutionsbox
  2. Permutation: Bytes werden gegeneinander verschoben
  3. Diffusion: Byte-Mixing durch Polynommultiplikation
  4. Schlüsselverknüpfung: XOR mit Rundenschlüssel

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

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

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?

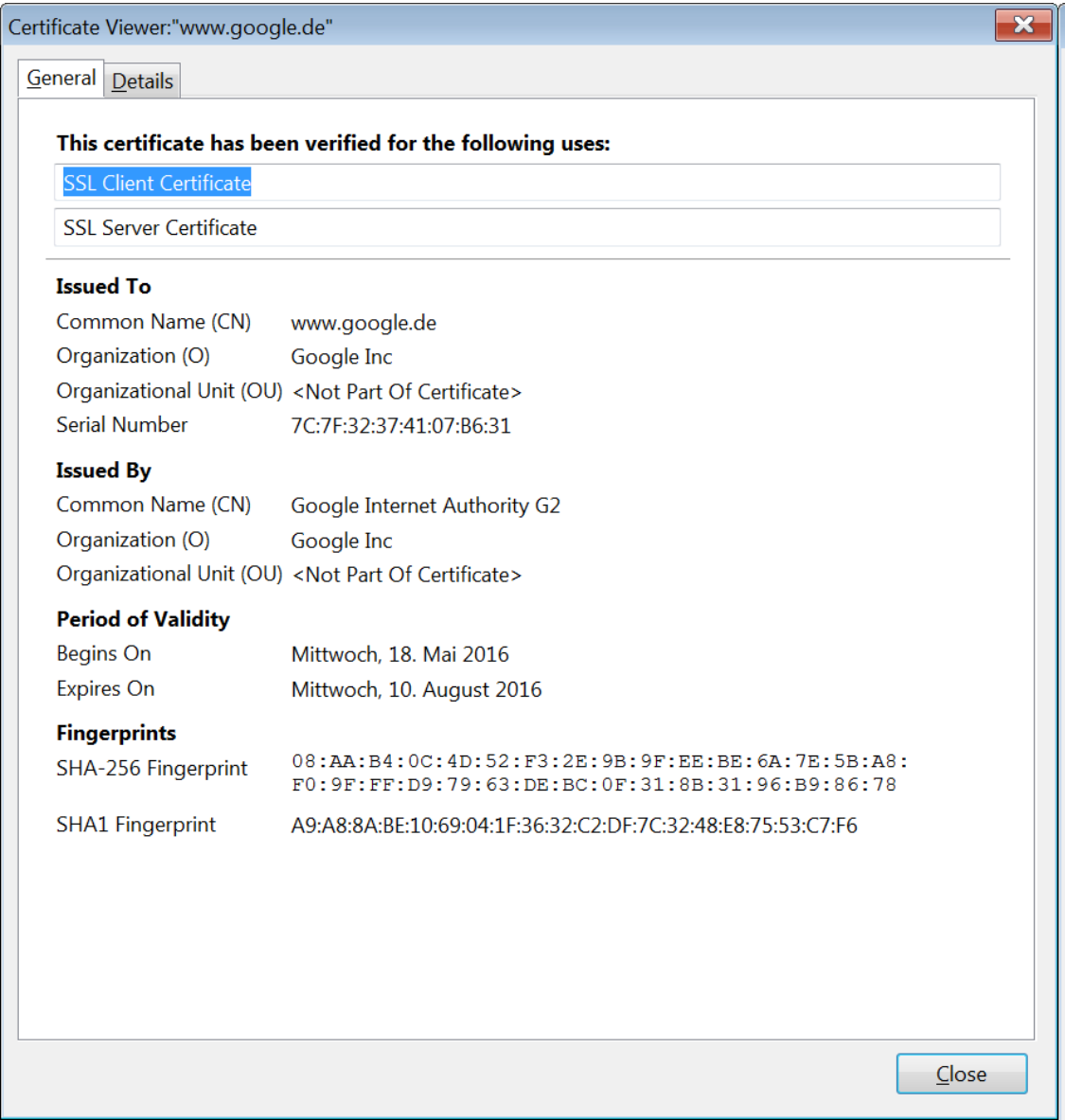
Ansatz: Kryptographische Bindung wird durch eine Instanz des Vertrauens bestätigt: Certificate Authority (CA)

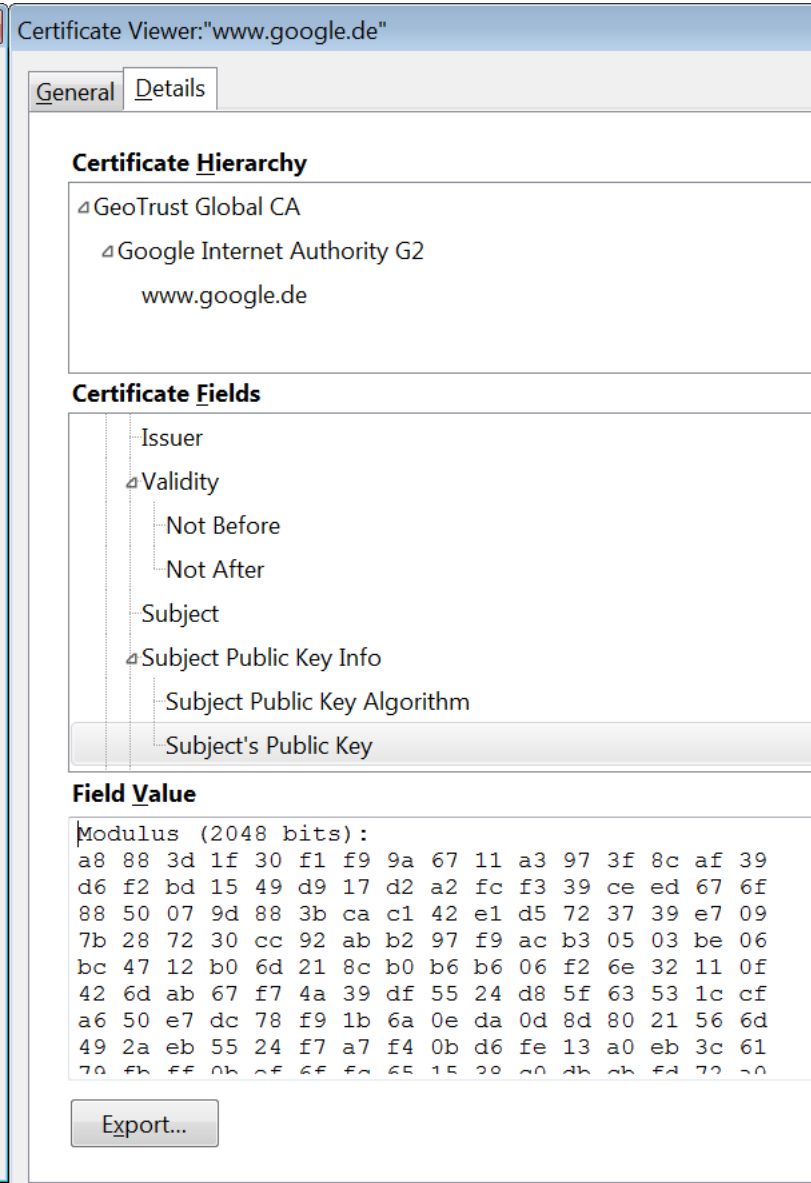
Methode: Die CA stellt ein Zertifikat aus, welches

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

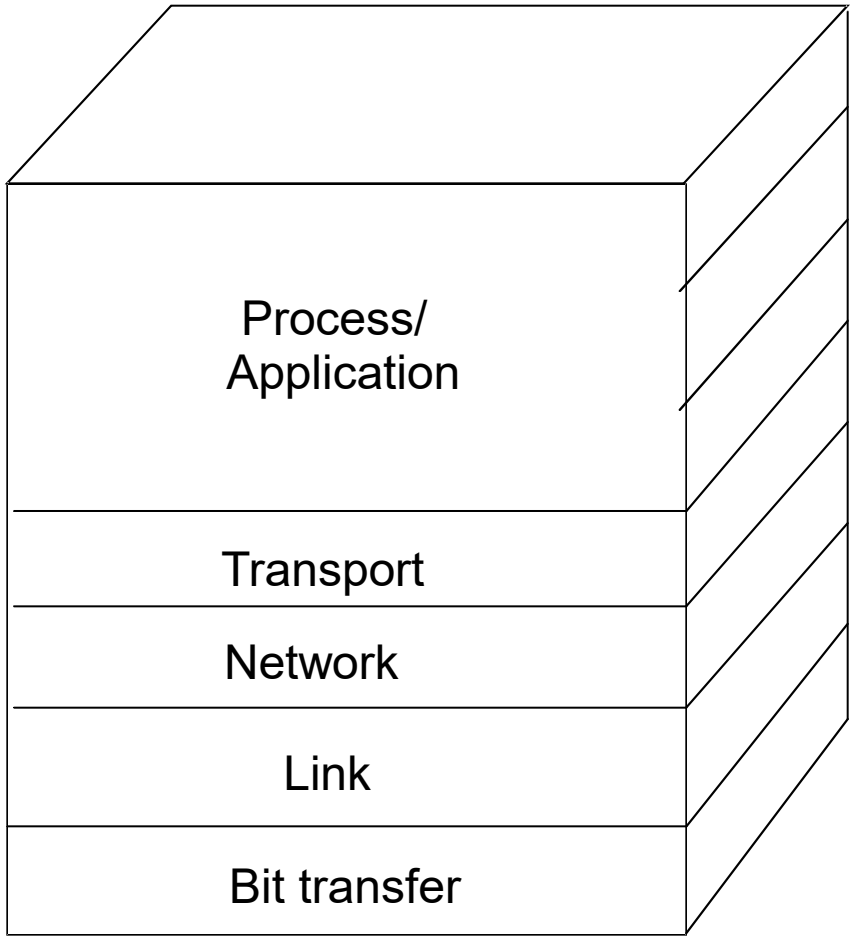
enthält und von der CA signiert wird.

2. Zertifikat im Browser





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)

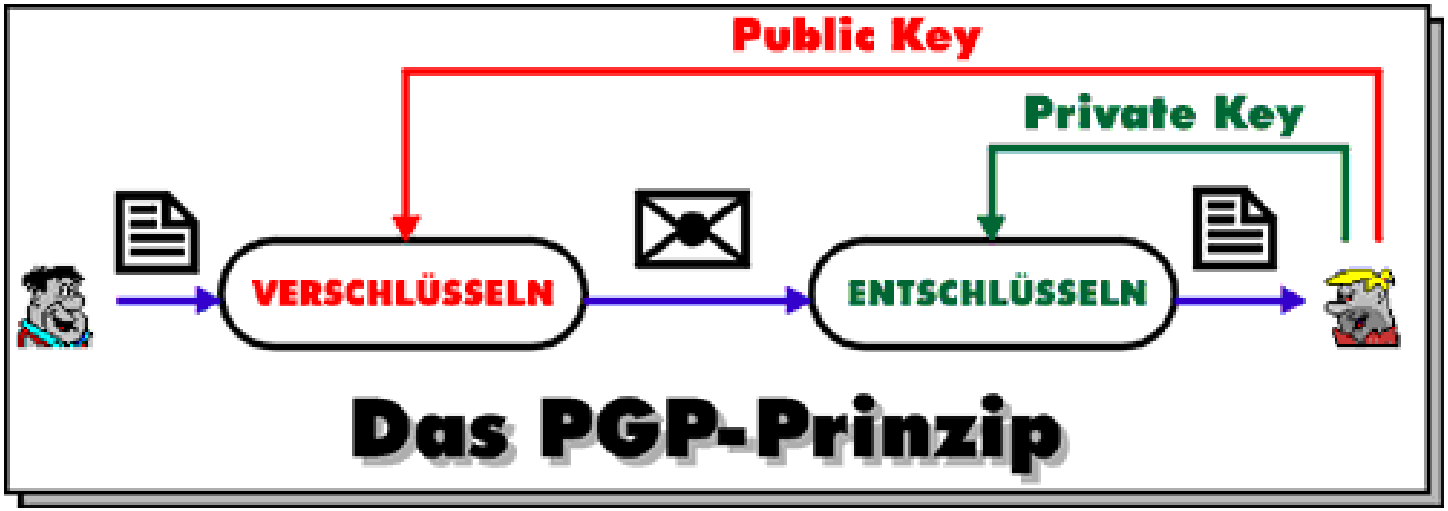
Vorteil:

- 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:

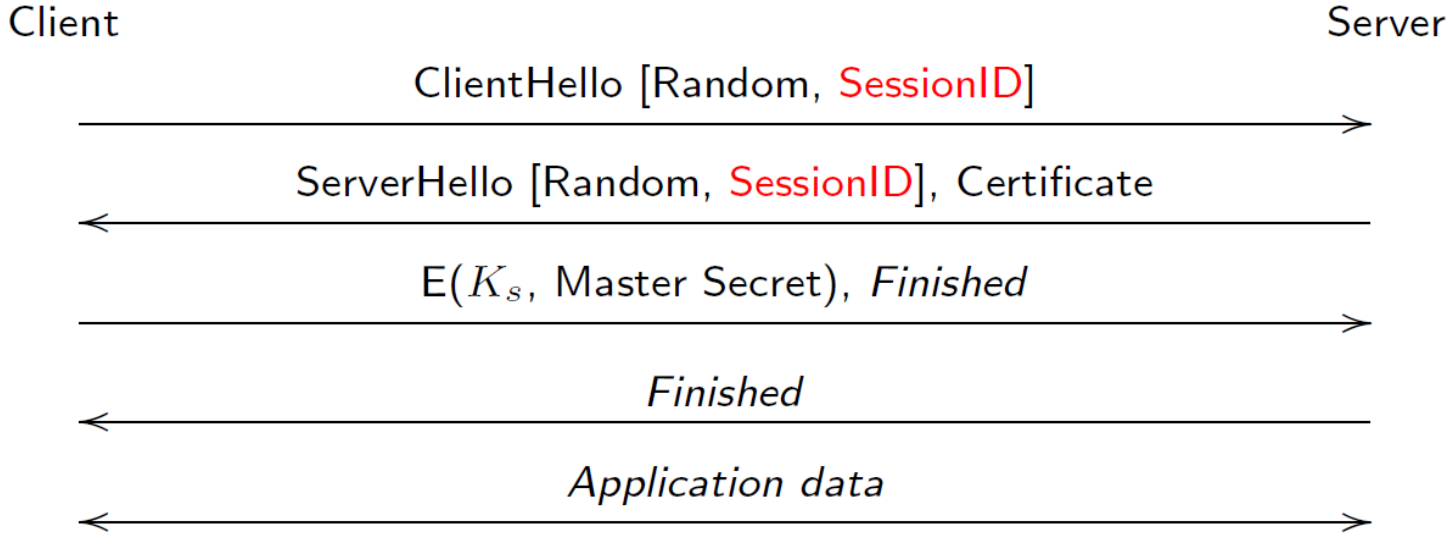
- Kommunikationprofile bleiben sichtbar auf der Transportschicht (einschl. Anwendungsprotokoll)
- Anwendungsprogramme müssen Bibliothek benutzen

3.2 Beispiel: SSL/TLS

- Transport Layer Security: RFC 2246, 5246 (v1.2), 8446 (v1.3)
- Protokoll für verschlüsselten Datenaustausch zwischen unbekannten 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.

3.2 Basisdialog zur Sitzungserzeugung

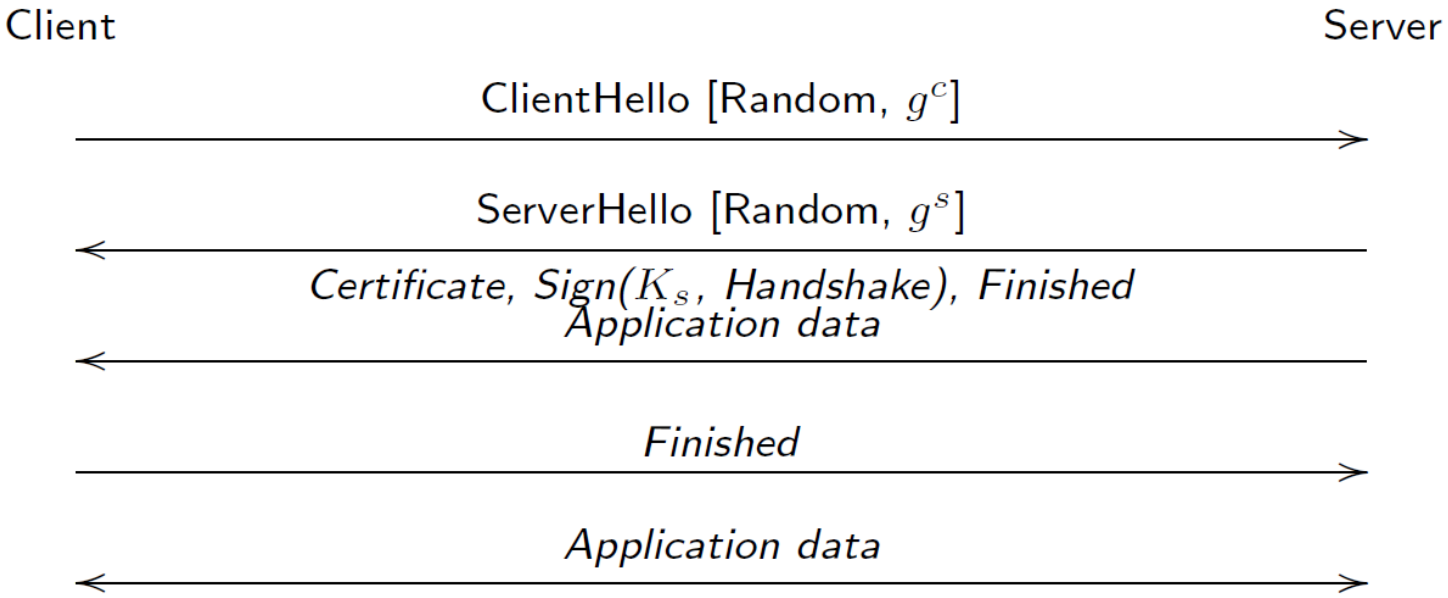
- TLS 1.2:



- Probleme:
  - TLS-Proxy erleichtert M-i-t-M
  - RSA Schlüsseltausch ohne ‚Perfect Forward Secrecy'

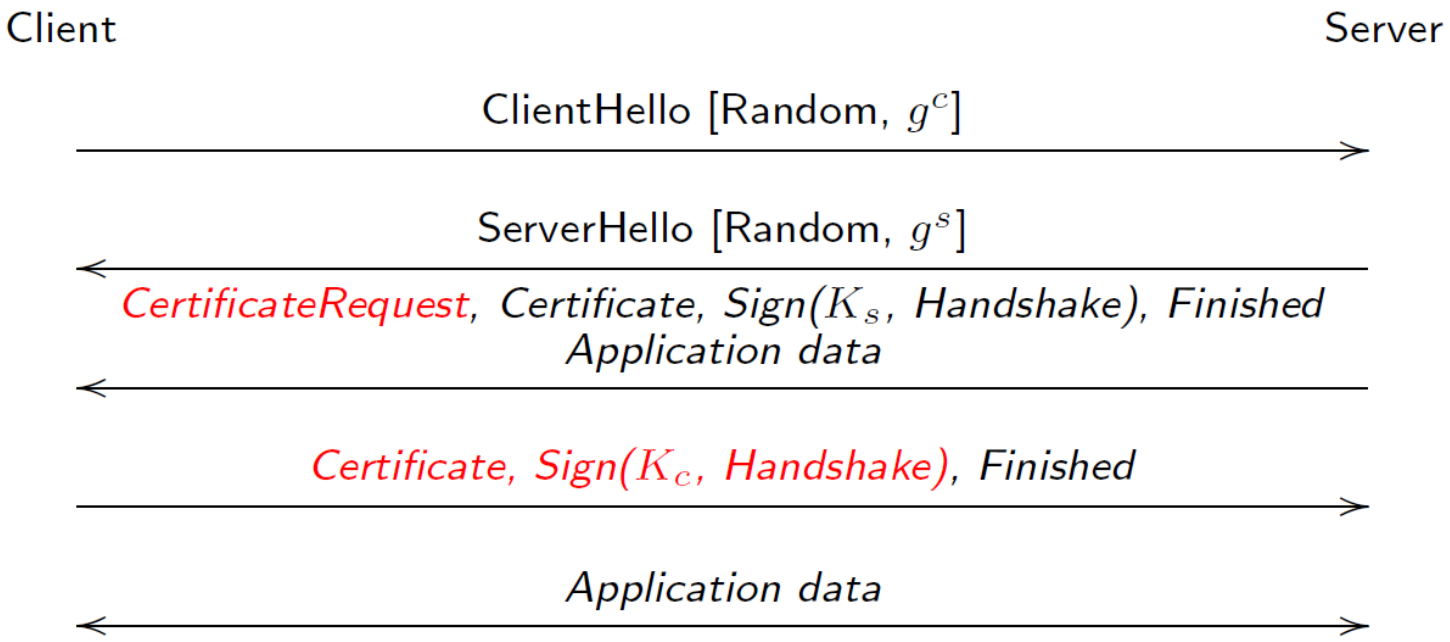
3.2 TLS 1.3 – RFC 8446 (2018)

- Nutzt Diffie-Hellmann mit Named Groups



- Gesichert gegen passive Angreifer

3.2 TLS 1.3 mit Client Authentication



- Gesichert gegen aktive Angreifer

3.3 Leitungsverschlüsselung (L 1)

Beispiel: Bit-Scrambling, WEP

Vorteile:

- Vollständige Informationsverschlüsselung
- Völlig transparent gegenüber höheren Schichten

Nachteile:

- Leitungsgebunden, nicht end-to-end
- Erfordert Hardware-Unterschützung

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:

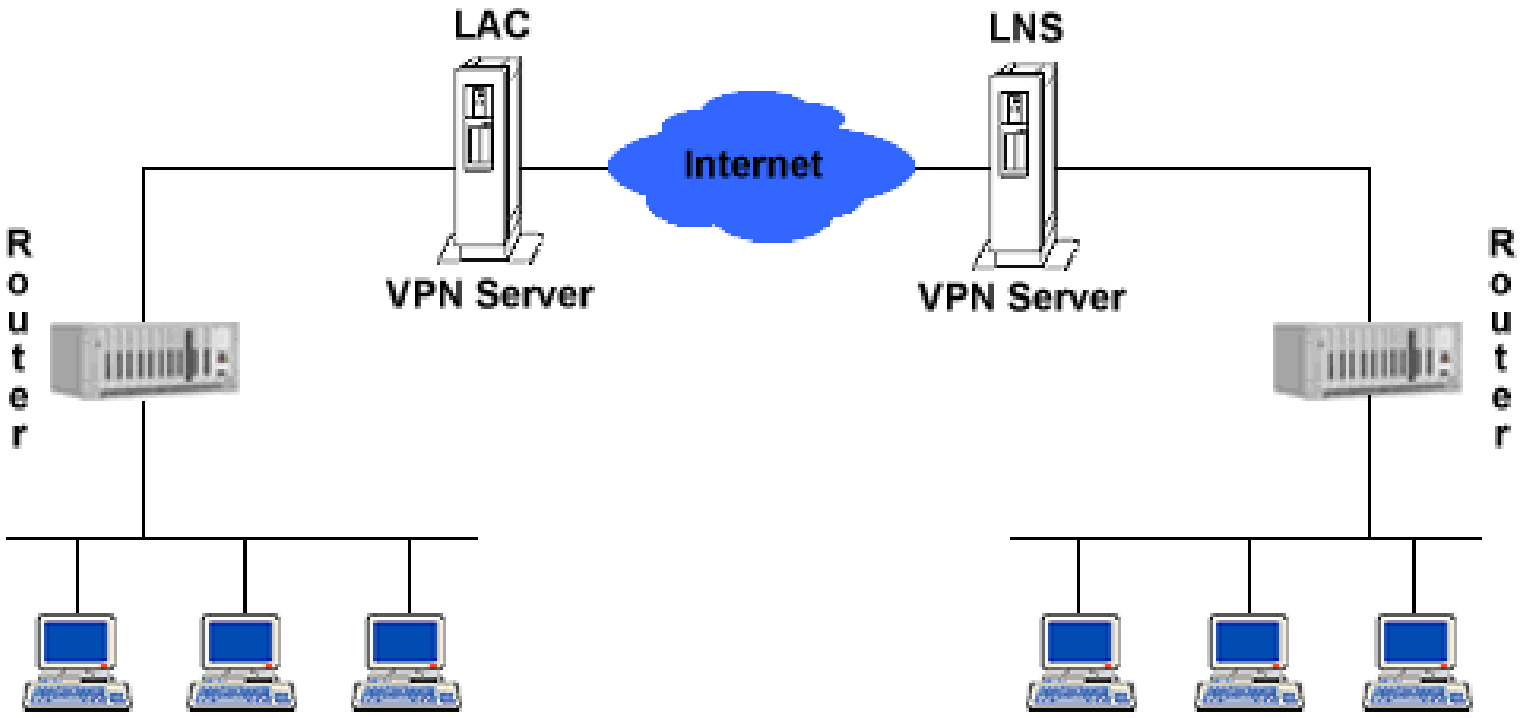
- Verhindert ARP spoofing + network intrusion
- Transparent zur Netzwerkschicht (nur Tunnel sichtbar)

Nachteile:

- Benötigt Server / Provider Unterstützung
- Begrenzte Skalierbarkeit / Performanz

Compulsory Tunnel (Carrier / ISP Model)

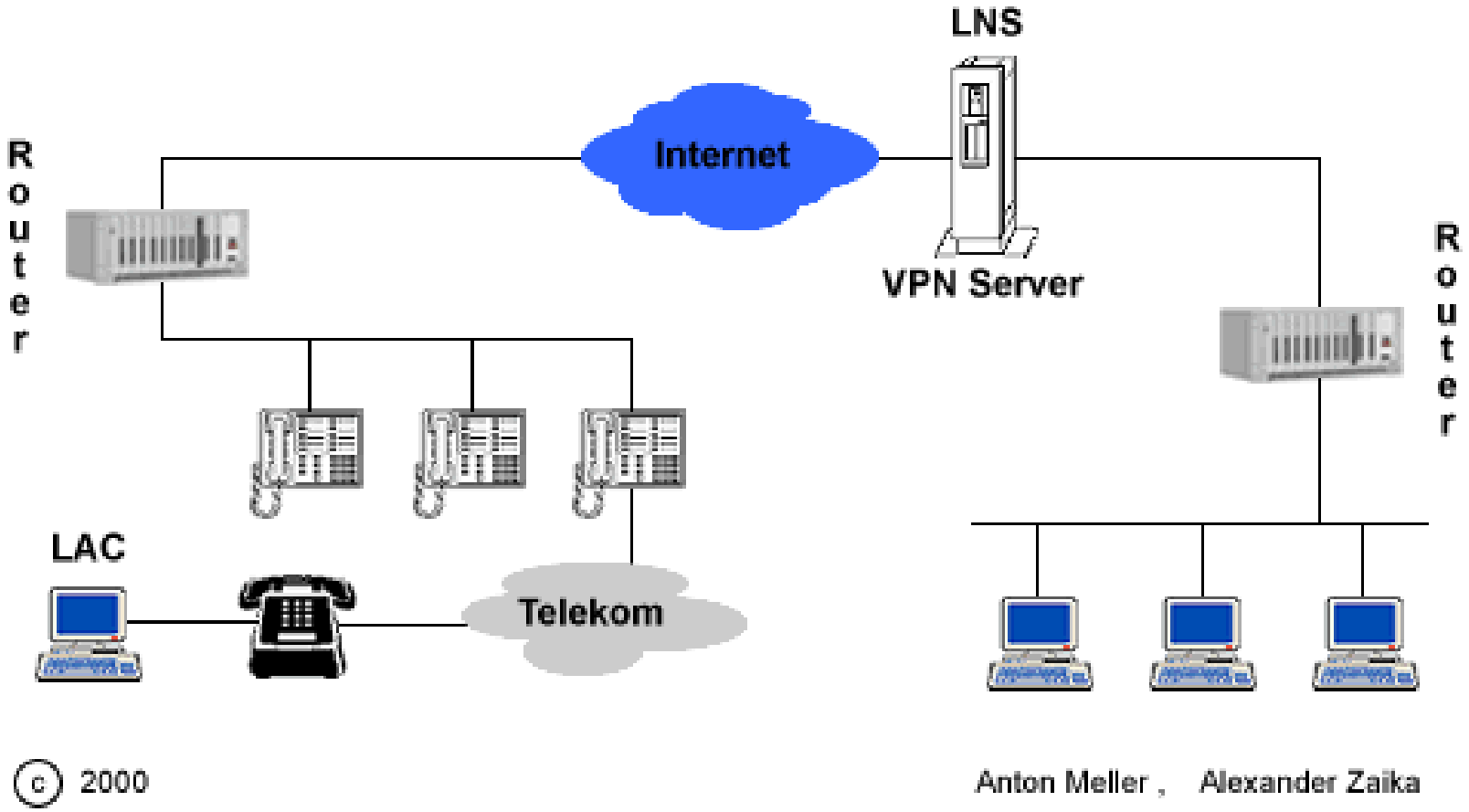
IP(Message)
PPP(IP(Message))
PPP(IP'(L2TP(××××××××××)))



© 2000 Anton Meller , Alexander Zaika

Voluntary Tunnel (Client Model)

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

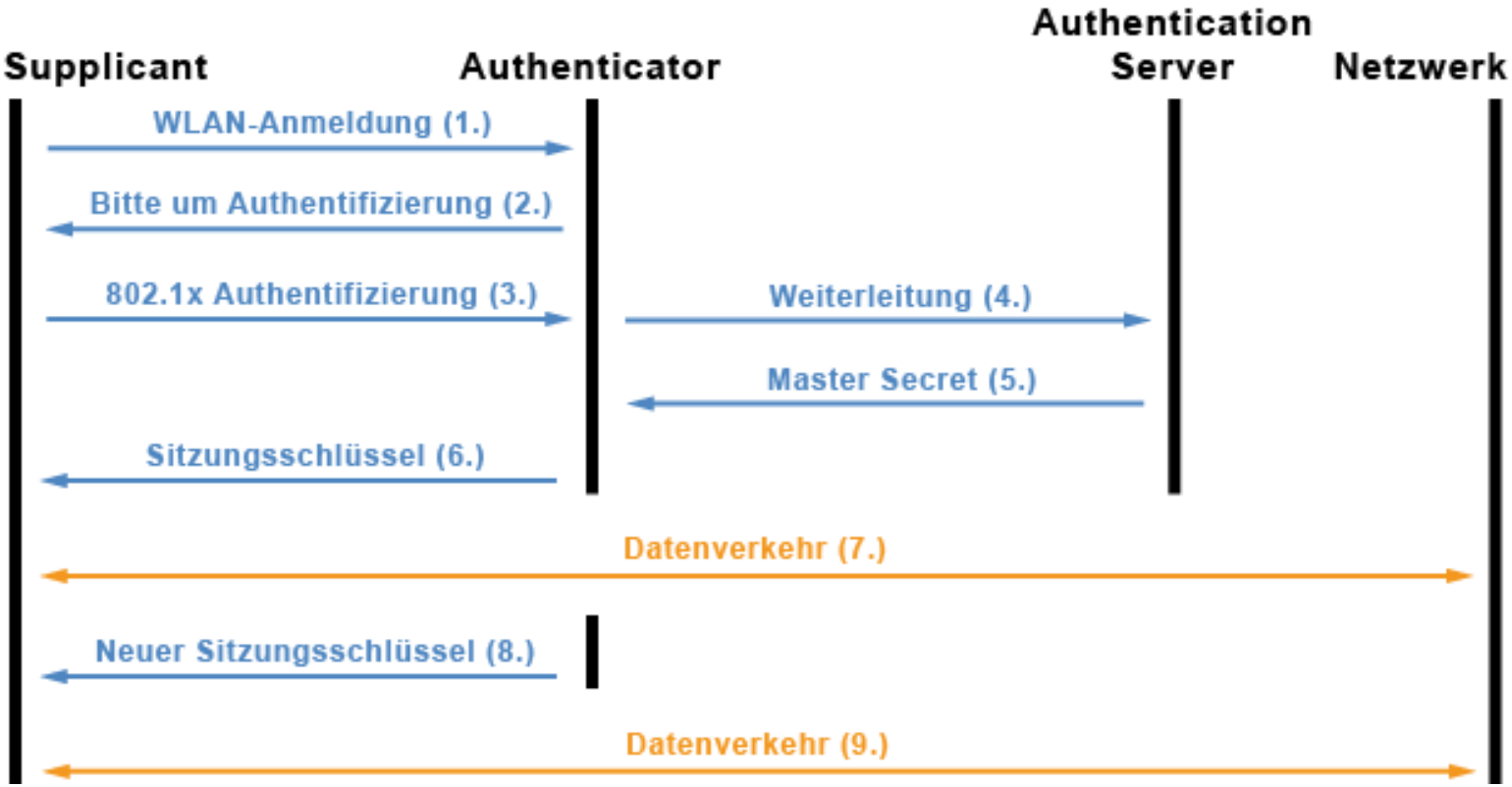


© 2000 Anton Meller , Alexander Zaika

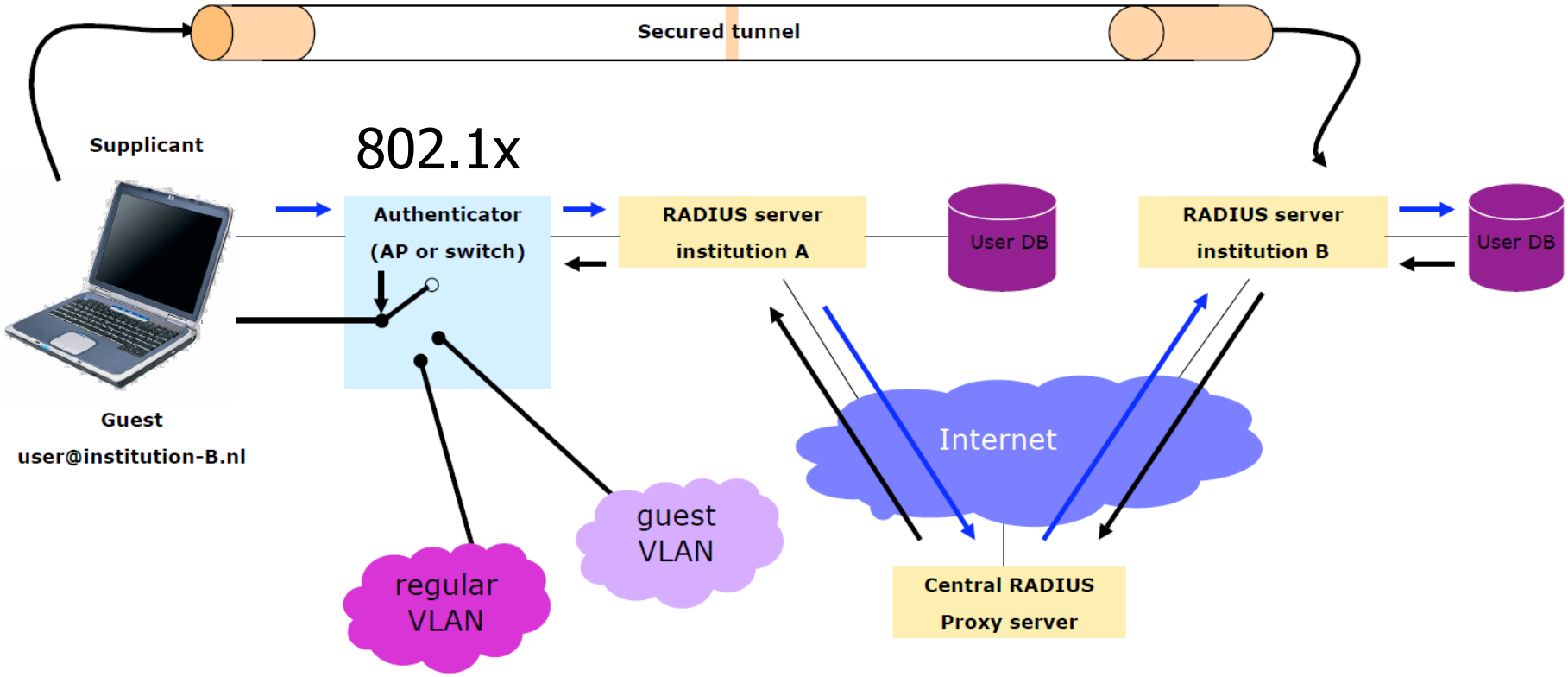
3.3 Eduroam

- Initiative der National Research Networks (NRENs), insbesondere SURFNet, und TERENA/Geant
- Start in 2004
- Heute: mehr als 26.000 Einrichtungen in 101 Ländern
- Basiert auf einer Kombination von Authentifizierungs-protokollen
  - IEEE 802.1x am Zugang mit dynamischer VLAN-Zuordnung
  - Extensible Authentication Protocol (EAP)
  - Hierarchische RADIUS Föderation

3.3 802.1x Netzwerk-Authentifizierung



3.3 Eduroam: Föderierte Authentifizierung mit RADIUS/EAP

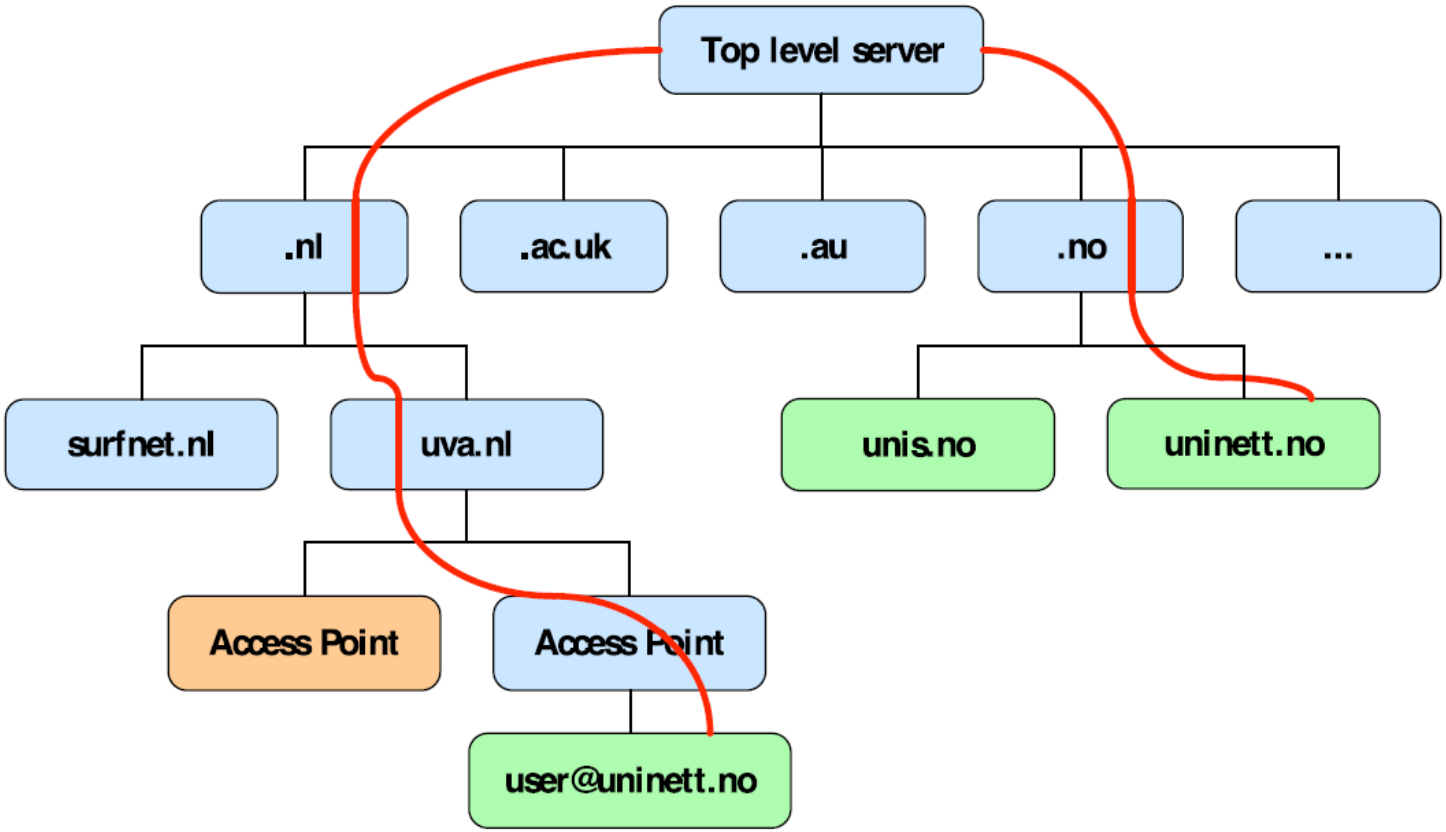


Trust based on RADIUS plus policy documents

3.3 RADIUS: Remote Authentication Dial-In User Service – RFC 2865

- Client-Server-Protokoll zur Authentifizierung, Autorisierung und Accounting am Netzwerk
- Drei Protokolloperationen
  - Access-Request (Bitte um Freigabe des Zugriffs)
  - Access-Accept (Annahme für die Freigabe des Zugriffs)
  - Access-Reject (Ablehnung der Freigabe)
- Nach erfolgreicher Authentifizierung übermittelt RADIUS eine konfigurierbare Anzahl von Zugangsattributen
  - Eduroam hat RADIUS um Attribute erweitert

3.3 Eduroam: RADIUS Proxy Hierarchie



4. Internet Layer (IP)

Beispiel: Paketverschlüsselung, Adressauthentifikation

Vorteile:

- Transporttransparent
- Effizient & weitverkehrs-routebar

Nachteile:

- Kommunikationsprofile auf dem IP layer sichtbar

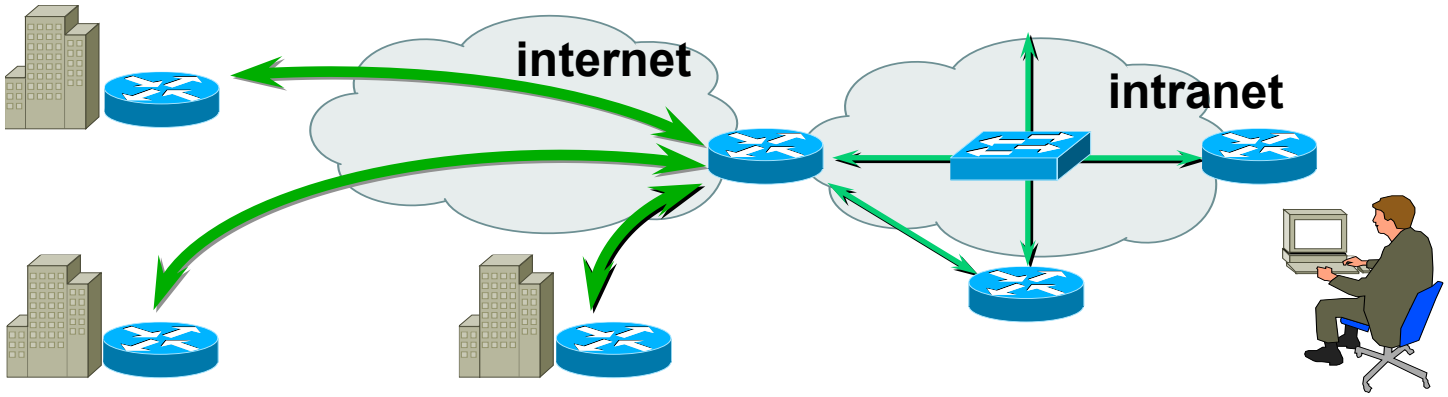
Ansatz: IP-in-IP secure tunnelling: IPSec

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)

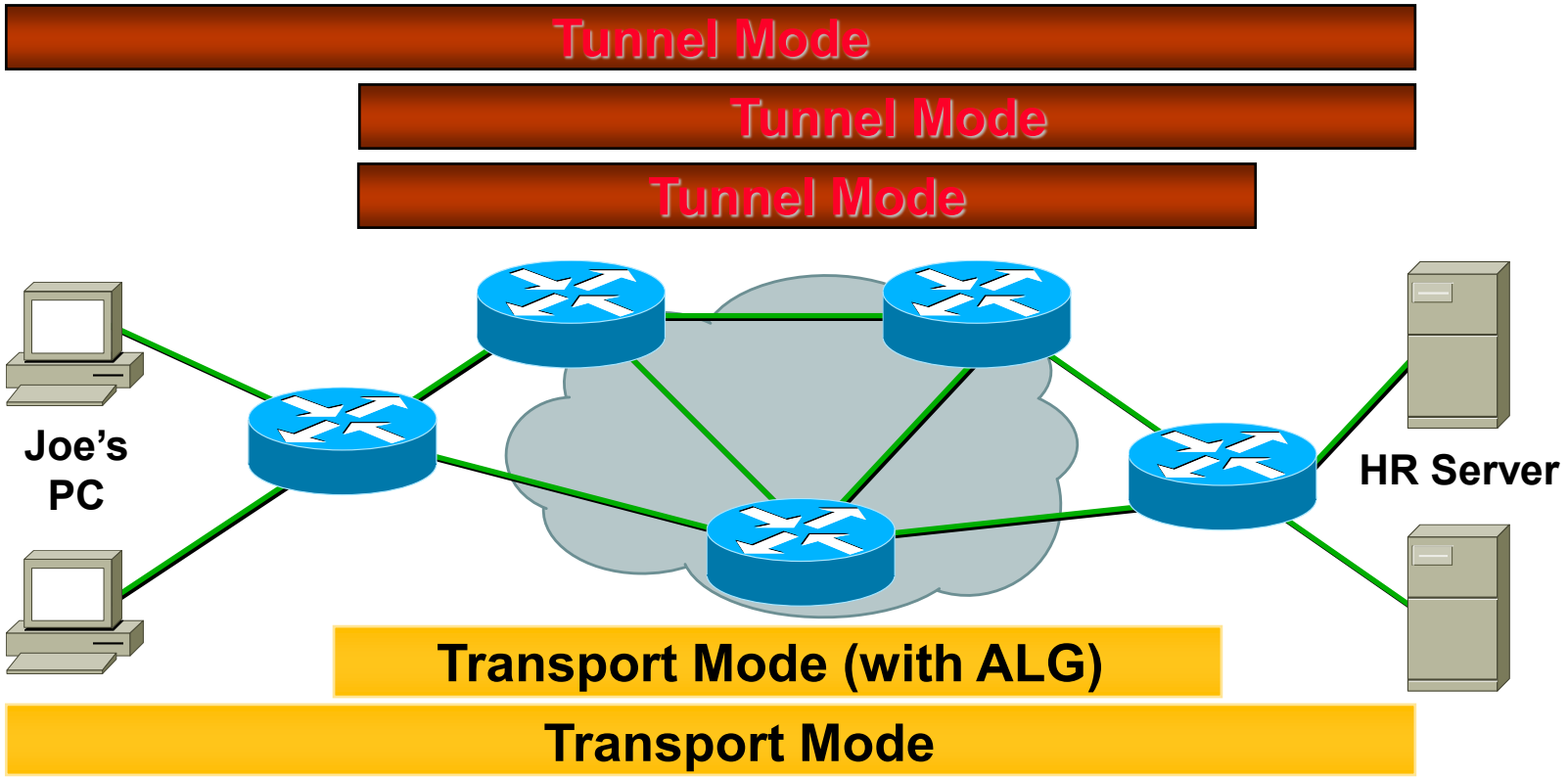
⇒ Eine end-to-end Sicherheitslösung auf dem IP layer

4.1 Konzepte von IPSec



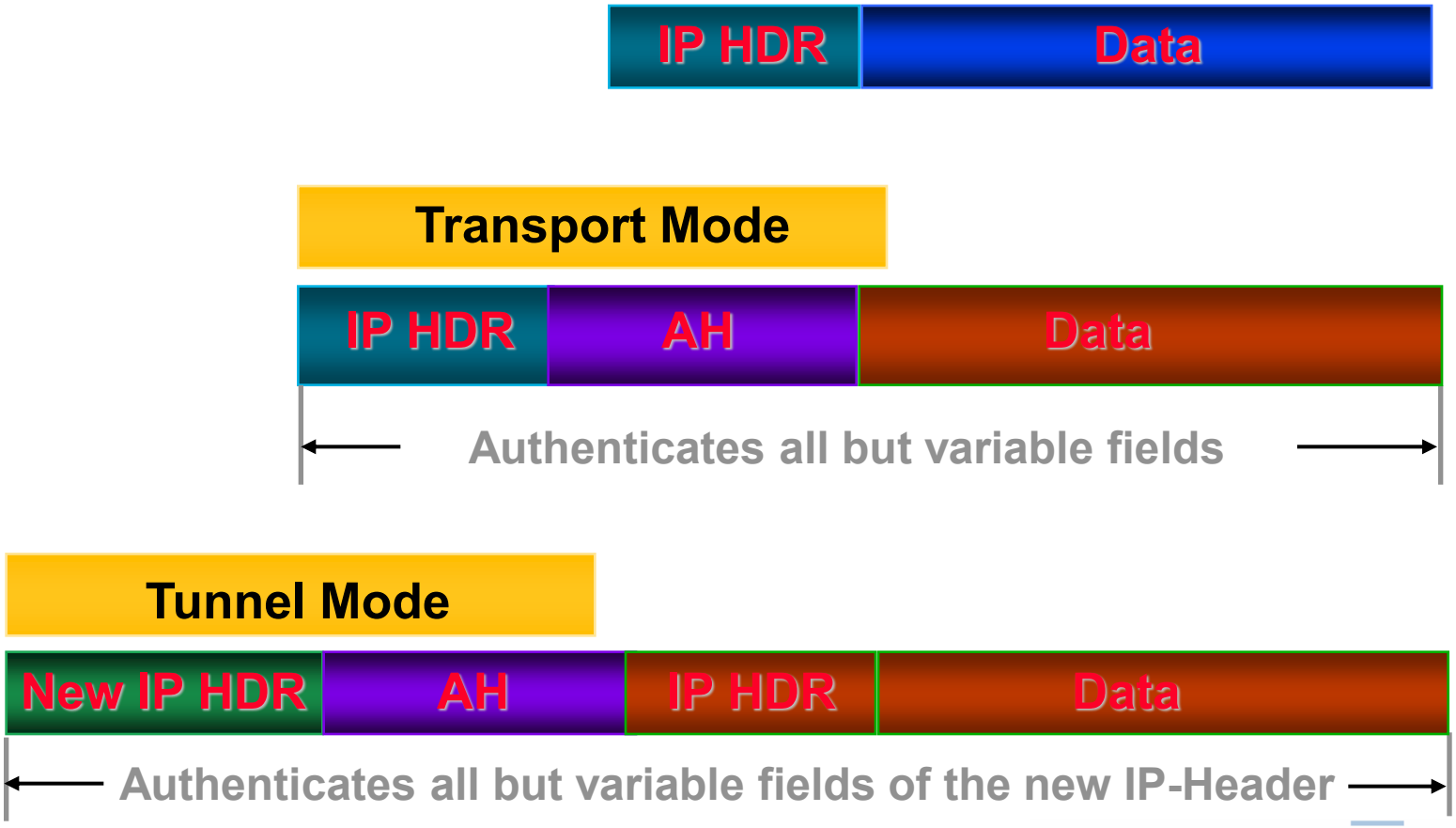
- Schützt Datentransfers durch das Internet mittels
  Authentication, Integrity, Encryption
- Transparent zu und angepasst an die Netzwerkinfrastruktur
- End-to-end Konzept

4.1 Tunnel und Transport Mode

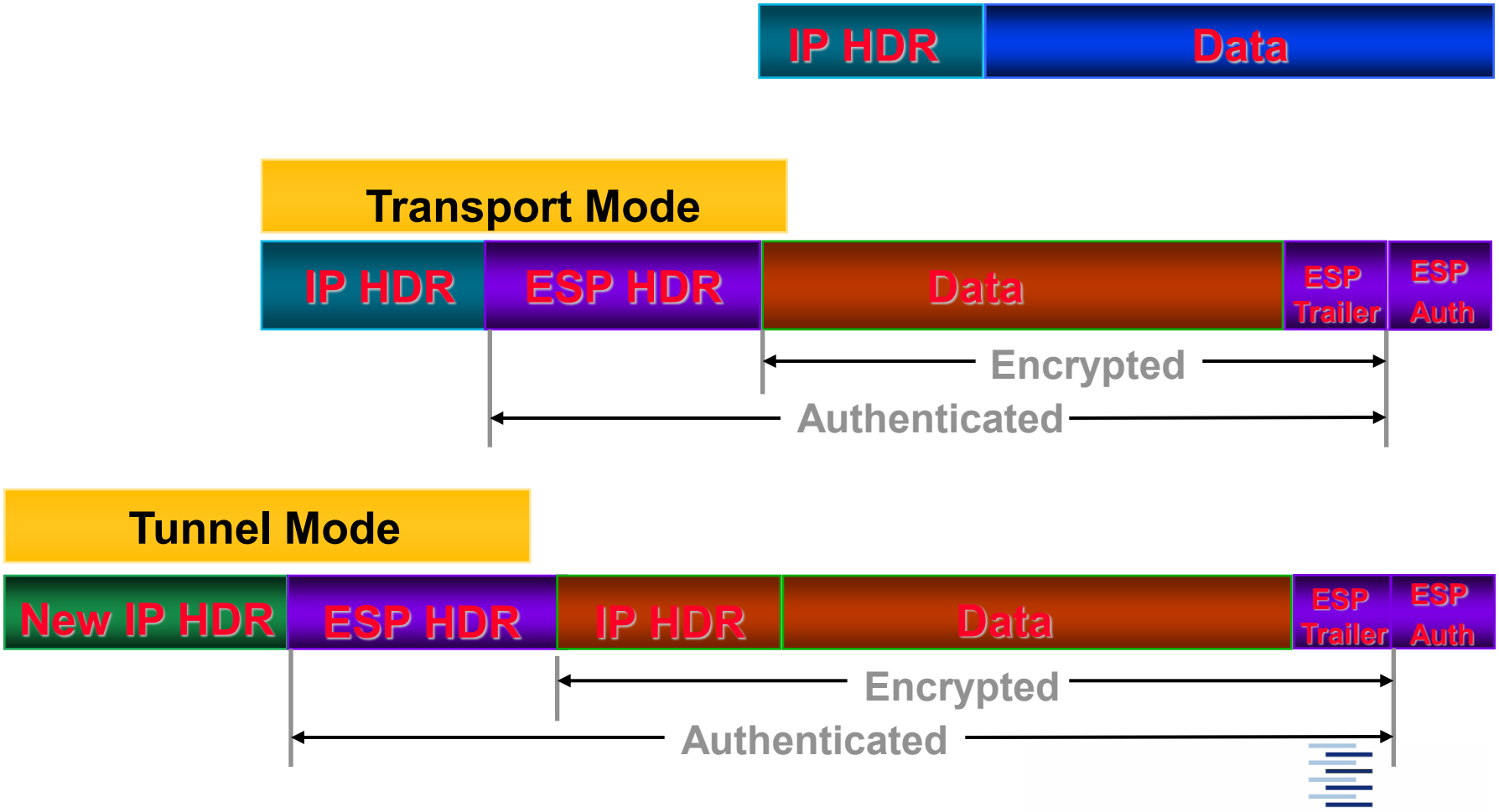


- Transport Mode End-to-End or via ALG
- Tunnel Mode for all connection types

4.1 IPSec Authentication Header (AH)



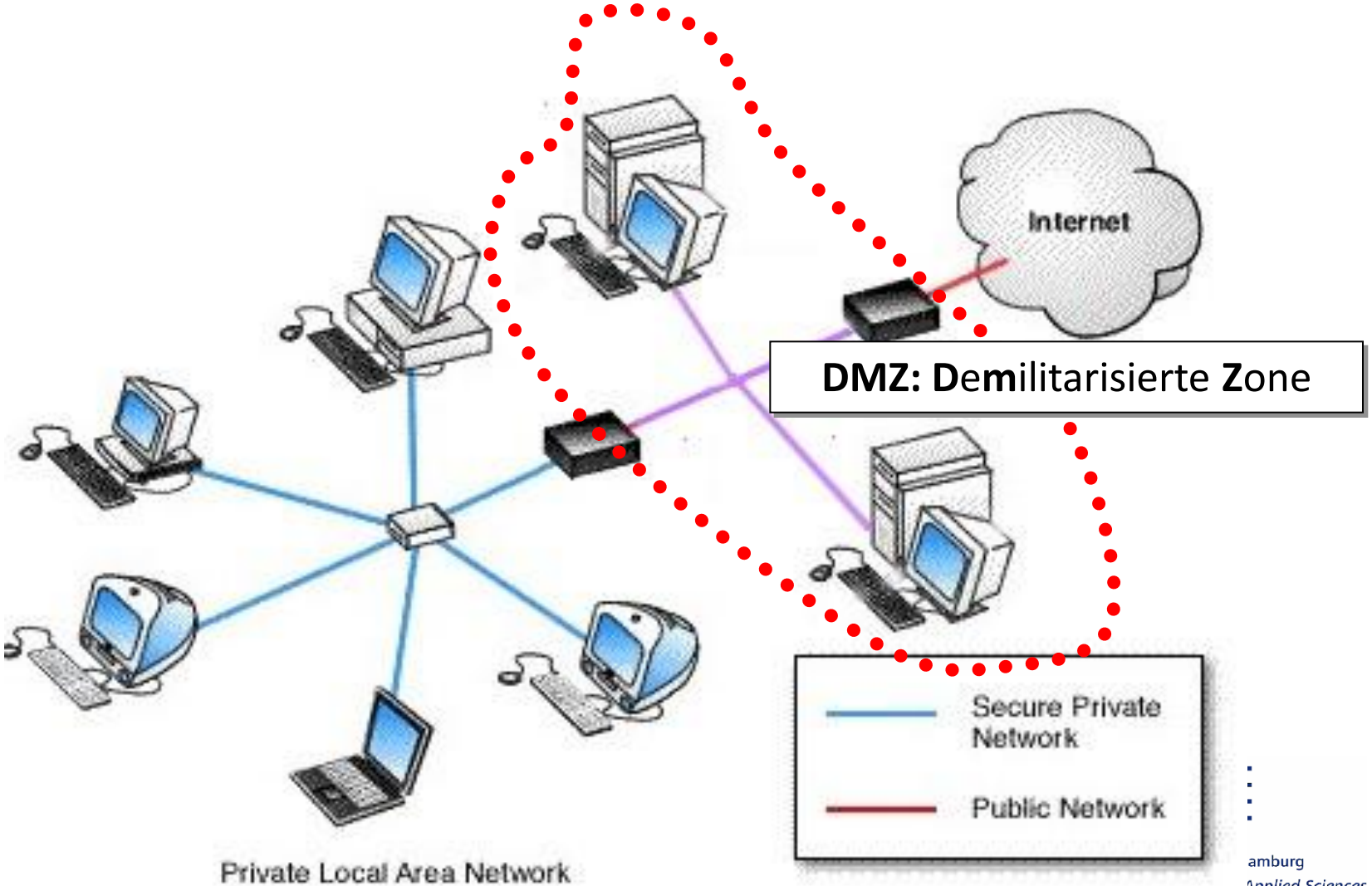
4.1 Encapsulating Security Payload (ESP)



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:

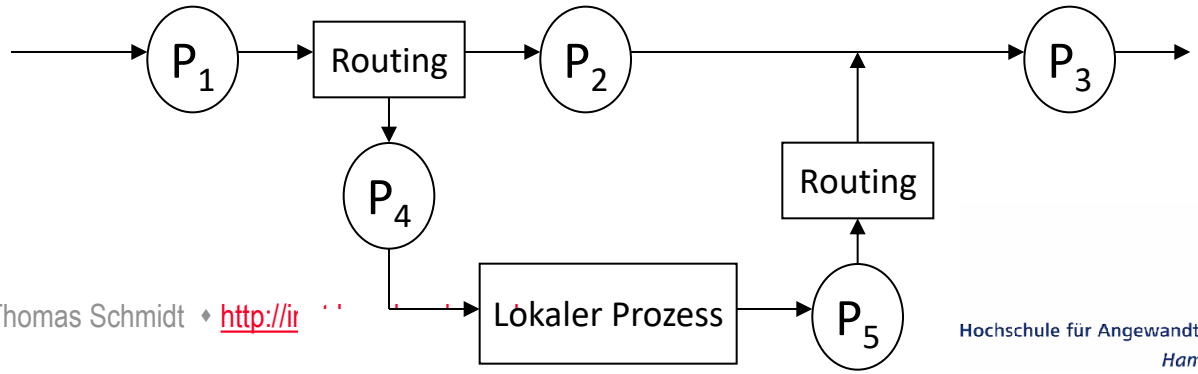
  → **`ACCEPT`**: lasse das Paket passieren

  → **`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*, 6th Ed., Pearson, 2013.
- Dieter Gollmann: *Computer Security, 3rd Ed.,* Wiley, 2011.
- Hans Delfs, Hartmut Knebl: *Introduction to Cryptography*, Springer, 2002.
- Christof Paar, Jan Pelzl: *Understanding Cryptography*, Springer 2010.
- Claudia Eckert: *IT Sicherheit*, 11th Ed., Oldenbourg Verlag, 2023.
- Internet Standards at: 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. Wer authentifiziert einen EDUROAM Nutzer beim Netzzugang?
5. Wie kann man mit einer Firewall alle SMTP-Pakete (Port 25) auf einen bestimmten Mailserver umlenken? Wie ginge das einfacher, besser und richtiger?