Agenda

- Netzwerk-Sicherheit
  - Sicherheitsaspekte bei verkabelten Geräten
  - Sicherheitsprobleme in den Protokollen TCP/IP, DNS
  - Firewalls – speziell *iptables, LIDS*
  - VPN – IPsec und OpenVPN
  - WLAN Security

Sicherheitsaspekte bei verkabelten Geräten

- Hubs
  - Nur eine Collision-Domain – jeder Client empfängt alle Frames
- Switches
  - Nur der Empfänger erhält die Frames
  - Möglicher Angriff: Bufferüberlauf – Neustart oder der Switch wird zum Hub – alle empfangen alle Packete
- Router, allgemein managed Geräte
  - Standard-Passwörter, Schwache Passwörter
  - Web-Interface/Telnet von außen erreichbar
  - Telnet: Logindaten ungeschützt
- SNMP hat bei einigen Produkten Sicherheitslücken
- Bugs in Diensten (http,telnet) > Firmwareupgrades!

Probleme mit TCP/IP

- Mögliche Angriffe
  - auf IP

    Spoofing – Packet verändern, Adressen fälschen

    DNS – Namensauflösung fälschen

  - auf TCP

    Spoofing – Packet verändern, Sequenznummern fälschen

    Session Hijacking – eine Verbindung übernehmen

    Denial of Service – Services/Server mit Anfragen überfordern

  - Lückenhafte Implementierungen

    Implementierungs/Plattform-spezifische Attacken

Probleme mit TCP/IP

- **Spoofing bei IP-Adressen Authentifizierung**

  Ein Server verlässt sich auf IP-Adressen

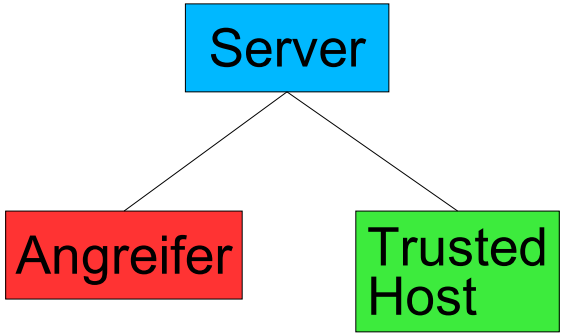
  Erraten der TCP-Sequnz-Nummer (ISN) des Servers

3-Wege TCP-Handshake

```
C > S: SYN(ISN_C)
S > C: SYN(ISN_s), ACK(ISN_C)
C > S: ACK(ISN_s)
C > S: Daten oder S > C: Daten
```

Angriff

```
X > S: SYN(ISN_x), SRC = T
S > T: SYN(ISN_s), ACK(ISN_x)
X > S: ACK(ISN_s), SRC = T (ISN_s geraten)
X > S: ACK(ISN_s), SRC = T, bel. Daten
```



C = Client, S = Server, T = Trusted Host, X = Angreifer

Probleme mit TCP/IP

- **Spoofing bei IP-Adressen Authentifizierung (Forts.)**
  - Sequenznummer erraten durch anderen Dienst z.B. Webserver

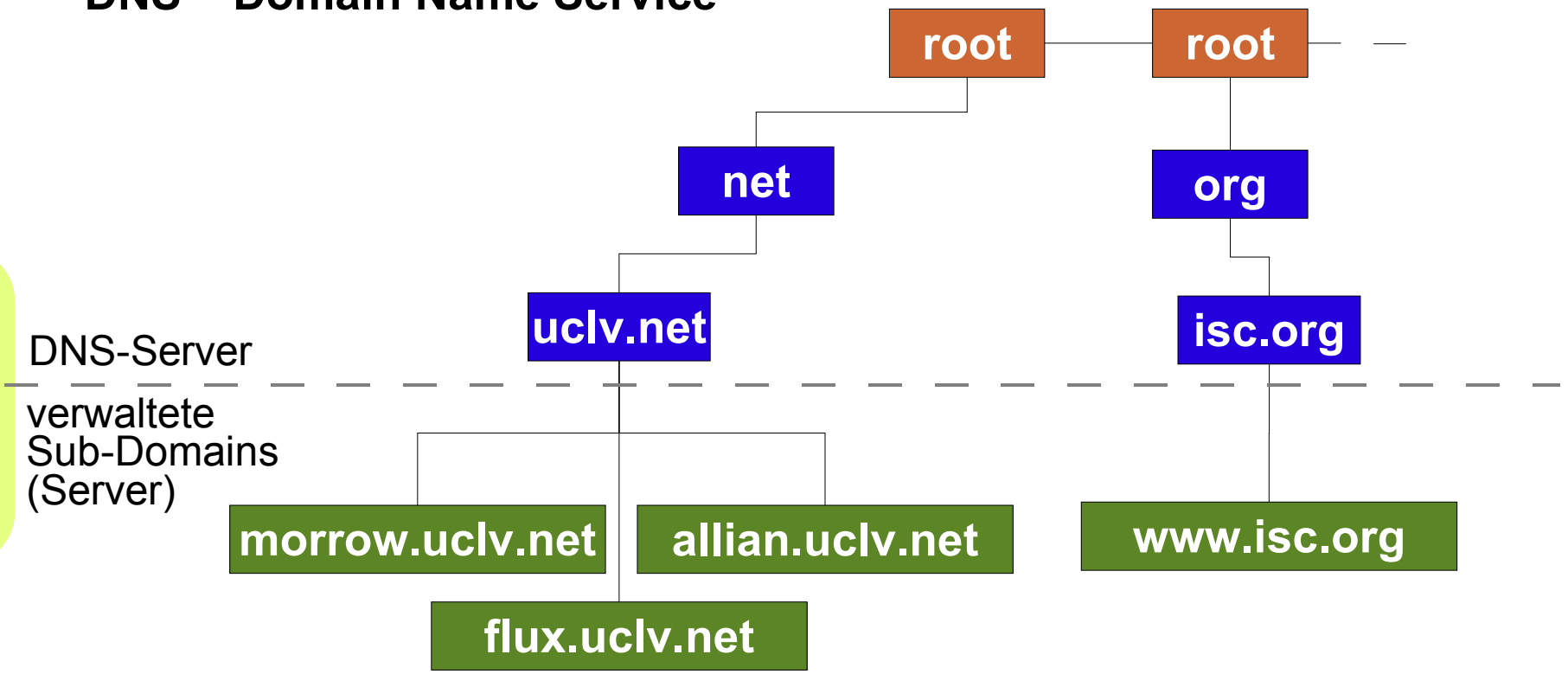
```
X > S: SYN(ISN_x)                 normaler Verbindungsaufbau
S > X: SYN(ISN_s), ACK(ISN_x)     Server sendet seine ISN
--
X > S: SYN(ISN_x), SRC = T        Gefälschtes Paket
S > T: SYN(ISN_s), ACK(ISN_x)     Server sendet seine ISN
X > S: ACK(ISN_s), SRC = T        Bestätigung an den Server, ISN_s geraten
```

C = Client, S = Server, T = Trusted host, X = Angreifer

  - Sequenznummer erraten durch Absturz und Neustart des Servers
    - z.B. mithilfe einer DOS-Attacke
    - Beim Start beginnen viele TCP-Stack-Implementierungen bei einer fixen Sequenznummer anstatt einer Zufallszahl

Probleme mit TCP/IP

- **DNS – Domain Name Service**



- **Angriffe auf DNS-Server**
  - DOS-Attacken (2002 größte DDOS-Attacke auf die 13 Root-Server)
  - DNS Spoofing

Probleme mit TCP/IP

- **DNS Spoofing**
  - Einem DNS-Server werden falsche Updates übermittelt
  - Dadurch werden URLs falsch aufgelöst
  - Der Angreifer kann so Web/eMail-Verkehr auf seinen Rechner umleiten
  - Es kann auch eine politische/wirtschaftliche Bedeutung haben sehr bekannte Webseiten zu fälschen

  - Sichere Lösung
    - Rekursive Anfragen nur an bestimmte DNS-Server weitergeben
    - Keine dynamischen Zone-Updates
    - Anfragen von außen nicht rekursiv weitergeben

Probleme mit TCP/IP

- **Zone-Transfer**
  - Beim Zone-Transfers bekommt ein Slave die Einträge des Masters
  - Der Angreifer kann Informationen über verfügbare Server aus dem Netz das dieser DNS-Server verwaltet bekommen

- **Cache Poisoning**
  - Anfällig sind schlecht konfigurierte DNS-Server die beliebige andere DNS-Server anfragen
  - Wenn ein DNS-Server selbst die URL nicht auflösen kann sollte er nur vertrauenswürdige (Whitelist) DNS-Server fragen.
  - Ansonsten holt der DNS-Server vom DNS-Server des Angreifers eine Auflösung und speichert diese in seinem Cache
  - Die Attacke kann auch auf dem Packet-Level gemacht werden, wenn der Angreifer die TransactionID des DNS-Updates erraten kann.

Probleme mit TCP/IP

- Bind (8 und 9) Konfiurationen
  - Zone-Transfer deaktivieren/einschränken – Whitelist. Überprüfung mit
    dig @<server-ip> <domain> axfr
  - Dynamisches Update einschränken (Bind spezifisches Feature)
    - Der Master benachrichtigt den Slave wenn es Updates gibt
  - Bind Beispiel für den eigenen DNS-Server (nur interne Anfragen)

```
options {
    forwarders { <forew-ip>; [..] }
    allow-query { 192.168.5.0/24; 127.0/16 }
    allow-transfer { none; };
    listen-on port 53 { 192.168.5.250; 127.0.0.1 }
};
zone "0.5.168.192.in-addr.arpa"  {
    type master;
    file "192.168.5.zone";
    allow-update { 192.168.5.0/24; localhost; };
};
```

Probleme mit TCP/IP

- Für offizielle Domains ist die Ideal-Lösung eine Aufteilung der Aufgaben auf 2 DNS-Server
  - *Advertising-Server*
    - Ist der Authority-Server für diese Domain
    - Nimmt Anfragen nur von anderen DNS-Server an
    - Ist nicht rekursiv
  - *Resolver-Server*
    - Beantwortet Anfragen aus dem internen Netz oder bekannen Forwardern
    - Rekursiv

Probleme mit TCP/IP

- **Session Hijacking**
  - Bei diesem Angriff wird eine TCP-Verbindung gekidnapped
  - Vorbedingung: der Angreifer muss alle Packete mitschneiden können
  - Ablauf
    - Der Angreifer sendet dem Server ein Reset und stellt die Verbindung zu sich her. (Ein Reset wird normalerweise nicht bestätigt)
    - Der Server bestätigt den Verbindungsaufbau – der Client gerät dadurch in den Status `DESYNCHRONIZED`. Er sendet ein ACK um die letzte gültige Sequenznummer anzuzeigen – der Server gerät auch in den Status `DESYNCHRONIZED`.
    - Es folgt ein ACK-Storm
    - Der Angreifer kann nun richtige Packete dazwischenschieben und bel. Daten hinzufügen.
    - Eigentlich kein einfacher Angriff, aber es gibt Tools dazu

      z.B. Juggernaut

Probleme mit TCP/IP

- **DoS – Denial of Service Attacke**
  - Ein Dienst wird mit Anfragen überflutet, sodass er echten Anfragen nicht mehr nachkommen kann. Führt bis zum Abstuzt des Dienstes und/oder des Systems bzw. Administrator bootet neu
  - Oft Grundlage für andere Attacken
  - Ein sog. Packet-Storm von der gleichen Adresse aus kann leicht erkannt und geblockt werden
  - DoS-Angriffe werden deshalb verteilt (von vielen Rechnern) oder mit gefälschten Adressen durchgeführt
  - DoS-Angriffe sind sehr schwer zu blocken, weil es keine Hinweise darauf gibt, ob es sich um eine Attacke handelt.

  - Syn-Flooding: Eine TCP-Verbindungsanfrage wird durch das gesetzte SYN-Flag im TCP-Header gekennzeichnet. Der Angreifer überflutet den Server mit solchen Packeten. Symptome auf dem Server: alle High-Ports werden aufbraucht. Hoher Speicherverbrauch und CPU-Belastung durch offene Netzwerksockets.

Firewalls

- Firewalls sind Paketfilter / Router
- Unter Linux (Kernel 2.4/2.6) häufig im Einsatz – *iptables*

- **iptables**

  Nachfolger von *ipchains* (2.2) und *ipfwadm* (2.0)

  Arbeitet mit dem Kernelmodul *netfilter*

  Ermöglicht

  - Packet-Filtern (IPv4 und IPv6)
  - Stateful-Firewall durch Connection-Tracking (IPv4)
  - NAT (Network Address Translation)
  - Packet-Header verändern (TOS, TTL, ...)
  - Bandbreiten-Begrenzung
  - Mitloggen
  - Einsatz von eigenen Plugins

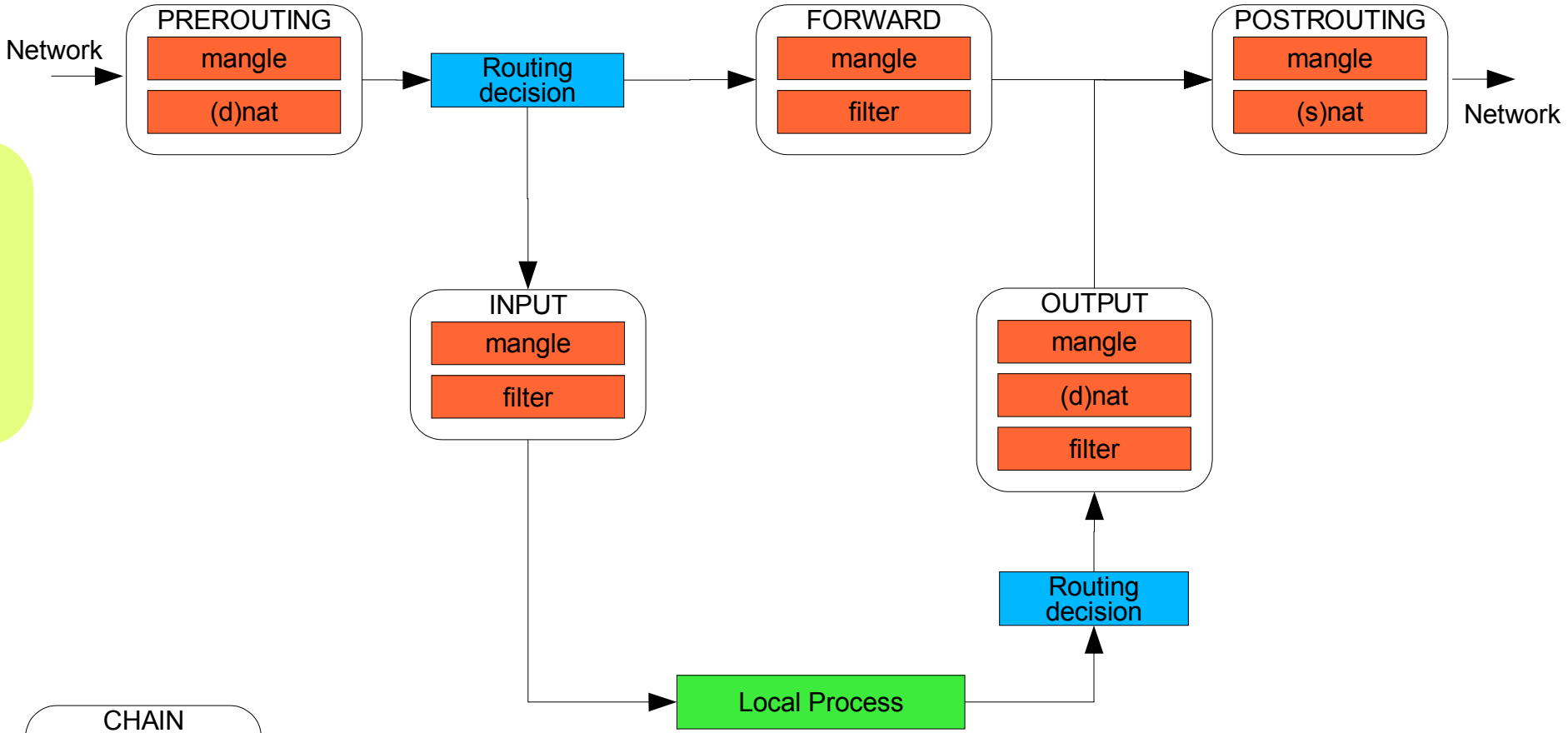
  - SuSE-Firewall (GPL) nutzt iptables
  - www.netfilter.org

Firewalls - iptables

- Aufbau von iptables
  - Chains – im Kernelspace
    - Prerouting – Vor dem Routing, Eingangspunkt aller Packete
    - Forward – Durchgeschleuste Packete
    - Input – Packete an einen lokalen Prozess
    - Output – Packete von einem lokalen Prozess
    - Postrouting – Nach dem Routing, vor dem Verlassen der FW
  - Tables – im Userspace
    - Mangle – Header-Felder verändern (TOS, TTL, ...)
    - Filter – Packete filtern
    - NAT – Network Address Translation
      SNAT, DNAT (MASQUARADING, REDIRECT)

Firewalls - iptables

Struktur von iptables



Firewalls - iptables

- **Connection-Tracking**

  Kernelmodul *conntrack*

  Es können TCP/UDP-Verbindungen erkannt und somit unterschieden werden. Es gibt folgende Connection-States

  - `NEW` – Packet das eine neue Verbindung initialisieren würde
  - `ESTABLISHED` – Packet gehört zu einer bestehenden Verbindung
  - `RELATED` – Neue Verbindungsanfrage, gehört aber zu einer anderen Verbindung z.B. FTP (Dataport/Controlport), ICMP
  - `INVALID` – ungültiges Packet das zu keiner Verbindung gehört

  Zwar spricht man bei UDP nicht von Verbindungen aber trotzdem kann man den Fluss von UDP-Packeten folgen

  Hinweis: Diese States sind nicht exakt gleich TCP-States

  z.B. `ESTABLISHED` (conntrack) = SYN, SYN+ACK

  `ESTABLISHED` (TCP) = SYN, SYN+ACK, ACK

Firewalls - iptables

- iptables Regeln

  Schema: iptables <optionen>

  Mögliche Optionen:

-A <chain>	Regel an diese Chain anhängen
-t <table>	default: filter
-s <source>	Absenderadresse
-d <destination>	Zieladresse
-i <interface>	Interface eingehend
-o <interface>	Interface ausgehend
-p <protocol>	Protokoll
-j <jump-to>	Ziel/Aktion
-sport	Port auf der Absenderseite
-dport	Port lokal
-icmp-type	ICMP-Type (echo-request, echo-response)
-syn	TCP-Verbindungsaufbau

Firewalls - iptables

- iptables Regeln – Ziele (-j)
  - ACCEPT — Packet annehmen
  - LOG — Log-Eintrag schreiben
  - DROP — Packet einfach verwerfen, keine Antwort.
  - REJECT — Packet blocken, aber dem Absender antworten.
  - SNAT — Source-NAT. Absenderadresse abändern
  - DNAT — Destination-NAT. Zieladresse abändern.
  - MASQUARADE — Absenderadresse der ausgehenden Packete auf die Adresse der Firewall abändern. Entspricht einem 1:n SNAT.

Firewalls - iptables

- iptables Regeln - Beispiele

```
iptables -A FORWARD -s 0/0 -i eth0 -d 10.0.0.8 -o eth1 -p TCP
        --sport 1024:65535 --dport 80 -j ACCEPT
```

Erlaube alle TCP-Verbindungen von eth0 über eth1 nach 10.0.0.8 Port 80(www)

```
iptables -A FORWARD -d 0/0 -o eth0 -s 10.0.0.8 -i eth1 -p TCP
        -m state --state ESTABLISHED -j ACCEPT
```

Erlaube alle Antwort-Packte in TCP-Verbindungen von 10.0.0.8 . Keine neuen Verbindungen sind dadurch erlaubt.

```
iptables -A POSTROUTING -i eth1 -p icmp --icmp-type echo-request -j
 DROP
```

Verbiete alle ICMP echo-requests (pings) auf die Firewall von eth1

Firewalls

- Firewalls unter Windows
  - Zonealarm (Freeware)
  - BlackICE
  - Sygate (Freeware)
  - McAfee
  - Symantec (Norton Internetsecurity / AV)
  - Windows XP built in Firewall

Windows-Firewalls sind definitiv kein Schutz für ein ganzes Netz. Einsatz für private Rechner.

Hochsicherheits-Systeme sind meistens BSD-Derivate, besonders OpenBSD wegen der hohen Code-Qualität

IDS – Intrusion Detection System

- LIDS – Linux Intrusion Detection System
  - Verfügbar für Kernel 2.4 und 2.6
  - Besteht aus Kernel-Patches und Konfigurations-Tools
  - Überwacht Zugriffe auf
    - Sockets / Ports
    - Dateien
    - Speicher
    - Prozesse
  - Regelt den Zugriff auf diese Ressourcen
    - Deny – Kein Zugriff (z.B. /etc/shadow)
    - Readonly – Nur Lesezugriff (z.B. auf Systemdateien)
    - Allow – auch Schreibzugriff erlauben
  - Einschränkungen gelten sogar für *root* damit die Folgen eines erfolgreichen Exploits geringer sind
  - www.lids.org

IPsec

- **IPsec stellt sicher**

  Authentifizierung, Integrität, Verschlüsselung, Anti-Replay Schutz

  auf OSI-Layer 3 – d.h. transparent für Anwendungen
- **Im Zuge mit IPv6 entwickelt**

  Deshalb wurde NAT nicht berücksichtigt > NAT Traversal mit UDP
- **Sicherheits Protokolle**
  - Authentication Header (**AH**)

    Setzt auf IP auf

    Authentifizierung, Integrität, Anti-Replay Schutz

    Inkludiert IP-Header beim Integrity-Check
  - Encapsulated Security Payload (**ESP**)

    Setzt auf IP auf

    Verschlüsselung, Authentifizierung, Integrität, Anti-Replay Schutz

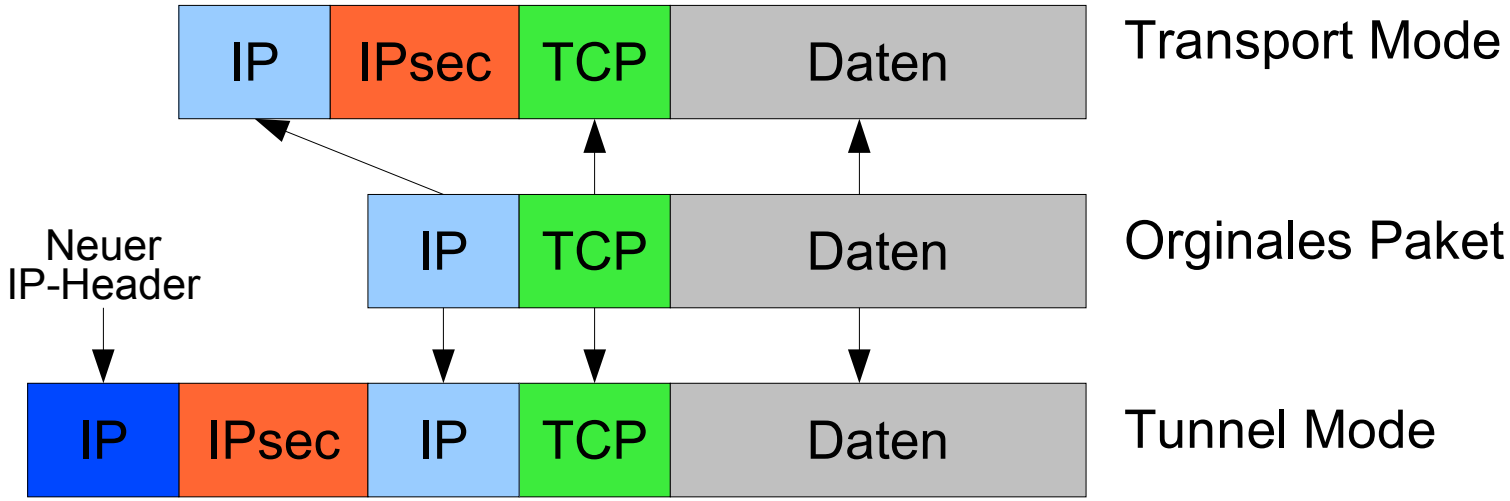
    Integrity-Check ohne IP-Header

IPsec

- 2 verschiedene Modi
  - Transport Mode

    Es werden nur die Daten verschlüsselt, nicht der IP-Header
  - Tunneling Mode (VPN)

    Das gesamte IP-Paket wird verschlüsselt



IPsec

AH (Integrity-Check inkl. IP-Header)
und/oder
ESP (Integrity-Check exkl. IP-Header)

IPsec

- **Key Management**

  IKMP - Internet Key Management Protokoll

  Beinhaltet folgende Spezifikationen

  - **ISAKMP** – Internet Security Association and Key Management Protocol (RFC 2408)

    Aushandlung von Security Associations
  - **DOI** – IP Security Domain of Interpretation for ISAKMP (RFC 2407)

    Welche Chiffren und Hashverfahren benutzt werden können
  - **OAKLEY** (RFC 2412)

    Definiert den Schlüsselaustausch, der auf dem Diffie-Hellman-Verfahren basiert
  - **IKE** Internet Key Exchange – UDP Port 500 (RFC 2409)

    Zusammenfassung von OAKLEY und ISAKMP zu einem Protokoll mit Schlüsselaustausch

IPsec

- IKE – Phase 1
  - Aushandeln einer Security Association
  - 2 Modi
    - Main Mode
      - Verschleierung der Identität – Verschlüsselung des Zertifikats
    - Aggressive Mode
      - Weniger Nachrichten als in der Main Mode
      - Identität erkennbar
- IKE – Phase 2 (QuickMode)
  - SA erneuern – Sessionschlüssel erzeugen
  - Es können mehrere SAs ausgehandelt werden

- **Verschlüsselungs-Algorithmen**
  - DES (Pflicht), meistens aber ist Triple-DES implementiert
  - je nach Implementierung alle häufigien blockbasierten Algorithmen
  - CAST-128, RC5/6, IDEA, Blowfish, AES(Rijndael), ...

IPsec

- Kritik von Sicherheitsexperten
  - Viel zu komplex
    - Führt zu Problemen in Implementierungen
    - Teilweise schlecht dokumentiert, viele RFCs
    - 2 Sicherheitsprotokolle und 2 Modi - Auf die Transport-Mode und das AH-Protokoll könnte man verzichten
  - ESP kann ohne Authentifizierung verwendet werden

  - Und trotzdem das beste IP-Security-Protokoll

IPsec

- IPsec unter Linux
  - FreeS/WAN
    - Momentan für IPv4, IPv6-Implementierung läuft
    - AH-Protokoll wird nicht mehr unterstützt
  - USAGI IPsec (UniverSAl playGround for IPv6)
    - Fix integriert ab Kernel 2.6
    - IKE-Deamon portiert von BSD (Kame-Project)
    - www.linux-ipv6.org
- IPsec unter Windows
  - Ab Win2k ist IPsec inkludiert
  - Standard-Konfigurations-Tool zu kompliziert (schlecht bei DHCP)
  - Tool von Markus Müller (vpn.ebootis.de)

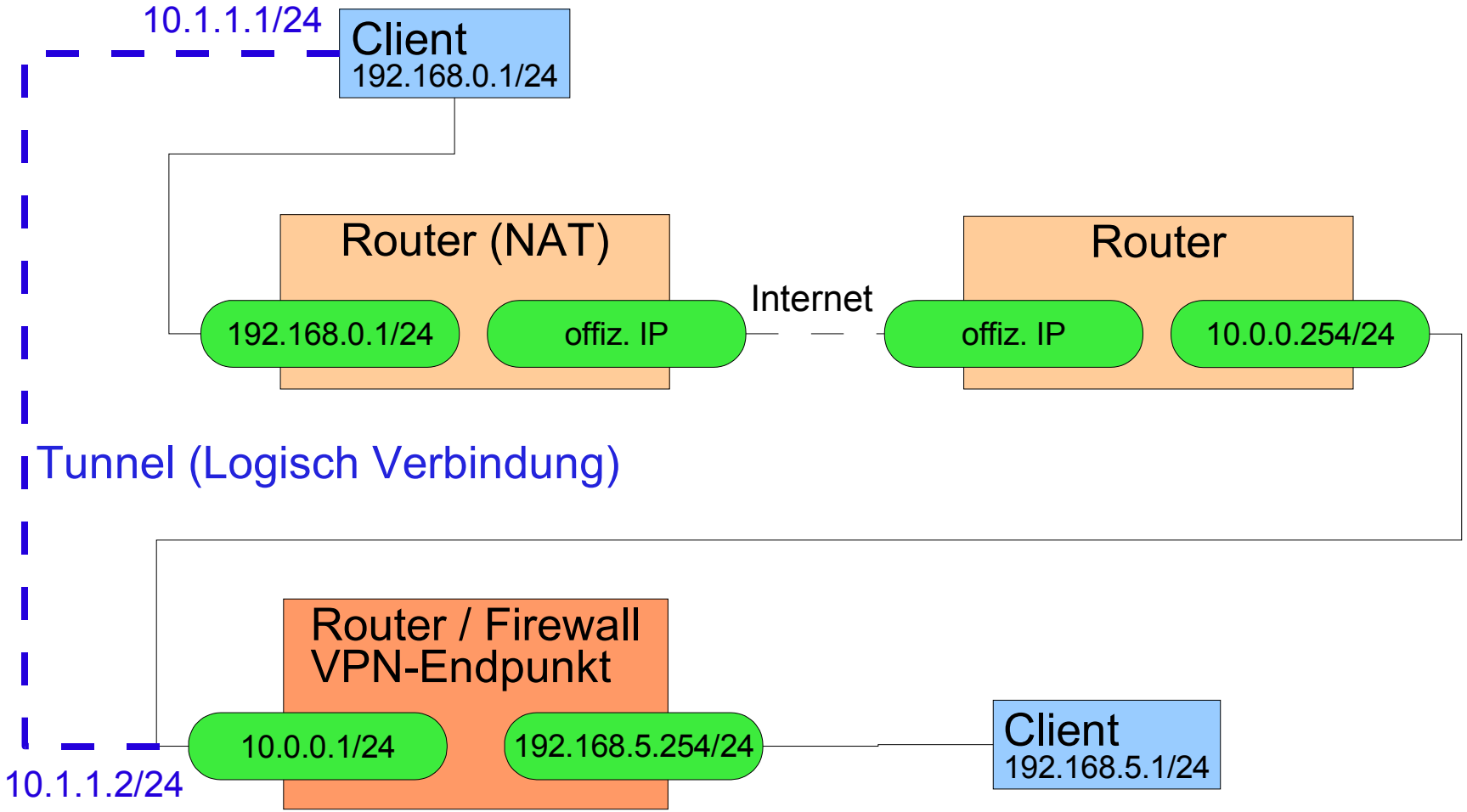
OpenVPN

- Gute und einfachere Alternative zu IPSec
- Warum eigentlich Sicherheit auf Layer 3?
- OpenVPN stellt den Tunnel mit TLS/SSL auf TCP/UDP her
- Vorteile
  - NAT ist möglich
  - Für alle wichtigen Betriebssysteme erhältlich
  - Code unter der GPL freigegeben
  - Läuft als Daemon im Userspace – kein Kernel-Modul, keine Änderungen am TCP/IP-Stack
  - Daten-Kompression
  - Verwendet für Authentifizierung, Verschlüsselung die OpenSSL-Bibliothek somit sind Verschlüsselungs- und Hash-Algorythmen aktuell durch einfaches Tauschen der Bibliothek. OpenVPN kann alles was OpenSSL implementiert.

- http://openvpn.sourceforge.net/

OpenVPN

OpenVPN Beispiel



WLAN

- WLAN – IEEE 802.11
  - 802.11a/b/g
  - 2 Modes
    - Ad-Hoc Netzwerke – die Clients kommunizieren direkt miteinander
    - Verwaltete (managed) Netze mit AccessPoint(s)
  - Mögliche States beim Verbindungsaufbau
    - Nicht authentifiziert, nicht verbunden (unassociated)
    - Authentifiziert, nicht verbunden
    - Authentifiziert, verbunden
    - Authentifizierung kann deaktiviert werden

WLAN Security

- WLAN – oder besser „Feind hört mit“

  Besonders leicht sind unverschlüsselte Logindaten aufzuspüren wie bei unsicheren Webverbindungen oder Email-Authentifizierung.
  - SSID broadcast / default / einfach zu raten
  - Nur MAC-Adressen-Filterung als Schutz
  - Fremde APs (einseitige Authentifizierung)

- 2 Lösungsansätze

  1) Das WLAN wird als unsicher betrachtet

  > Die Clients können nur per VPN das interen Netz erreichen

  2) Das WLAN wird als sicher betrachtet

  > Das Protokoll (802.11) muss selbst für Authentifizierung, Integrität und Verschlüsselung sorgen

WLAN Security > WEP, WPA, RSN

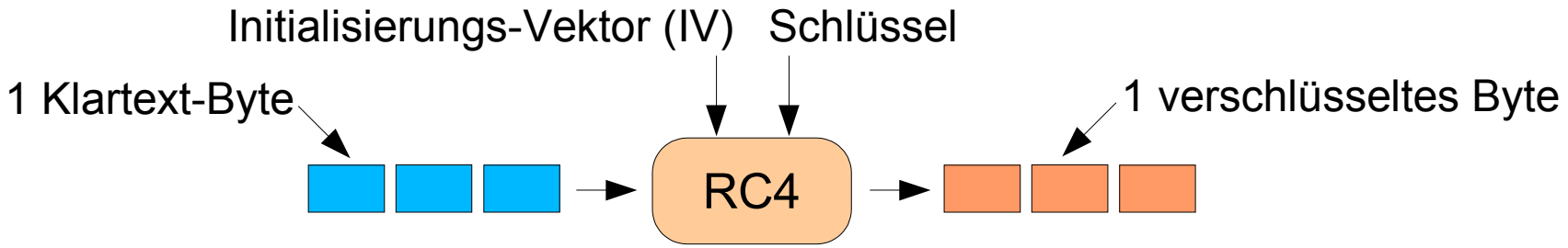
- Lösungsansatz 2 – Sicherheit in IEEE 802.11
  - WEP (Wired Equivalent Privacy)
    - Entworfen mit IEEE 802.11
    - Bekam anfänglich zuwenig Beachtung
    - Viele Sicherheitslücken und Schwächen
    - Ist dennoch sehr viel besser als unverschlüsselt!
  - RSN (Robust Security Network)
    - Standard IEEE 802.11i – kürzlich freigegeben
    - Starke Sicherheit
  - WPA (Wi-Fi Protected Access)
    - Untermenge von RSN, aber Hardware kompatibel mit vielen Geräten die WEP unterstützen
    - Wegen der großen Lücken in WEP als Übergangslösung für bestehende Hardware entworfen

WLAN Security > WEP und seine Probleme

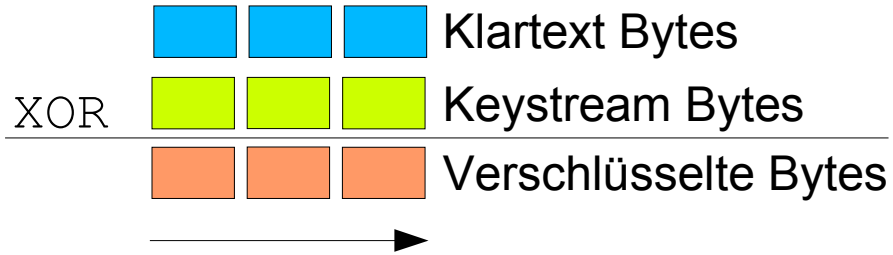
- Attacken auf WEP
  - Brute-Force-Attacke – alle Schlüsselkombinationen probieren
  - Dictonary-Attacke – häufige Passwörter probieren
  - Authentifizierung liefert Hinweise
  - Known-Plaintext-Attacke
  - Direct-Key-Attacke (LMS)
    - WEP-Schlüssel direkt raten
    - Die gefährlichste Attacke – automatisierbar mit Tools!
    - Daten mitschneiden – Sammeln von Frames mit schwachen IVs

WLAN Security > WEP und seine Probleme

- RC4
  - Entwickelt von Ronald Rivest (RC4 = Rivest Cipher Nr. 4) RSA Labs
  - Ist ein Stream-Algorithmus (Byte für Byte verschlüsseln)



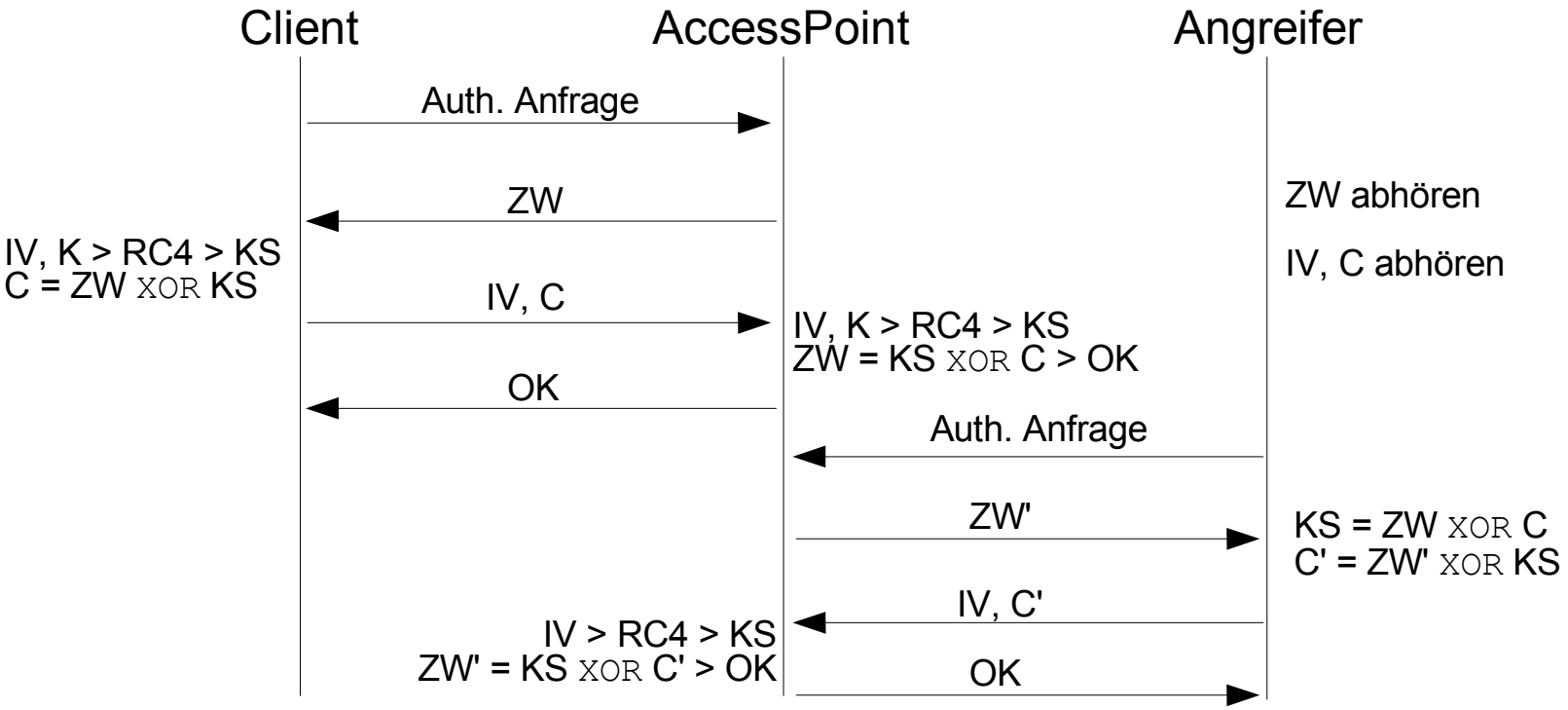
  - Das Herzstück von RC4 ist die Generierung des Keystreams. Die Verschlüsselung selbst ist ein simples `XOR` des Keaystreams mit dem Klartext. (`XOR`: zwei gleiche Bits ergeben `0`, zwei ungleiche `1`)




```
11010110
00010111   XOR
--------
11000001
```

WLAN Security > WEP und seine Probleme

- Sicherheitslücken/Schwächen in WEP
  - Authentifizierung nutzlos
    - Challenge-Response Mechanismus
    - Authentifizierung liefert IV, Klartext und verschlüsselten Klartext



KS = KeyStream, ZW = Zufallswert, C = Chiffre, K = Key
Bit1 XOR Bit2 - zwei gleiche Bits ergeben 0, zwei unterschiedliche ergeben 1

WLAN Security > WEP und seine Probleme

- Sicherheitslücken/schwächen in WEP (Forts.)
  - Authentifizierung (Forts.)
    - Einseitige Authentifizierung
    - Es wird der gleiche Schlüssel bei der Verschlüsselung verwendet
  - Zugriffskontrolle
    - Reine MAC-Adressen Überprüfung
    - Authentifizierung ist somit gleichzeitiger Zugriff
  - Schwacher Schlüssel – 40Bit

    Später 104Bit Schlüssellänge von der Industrie eingeführt

    Als Passwörter kommen meistens reine alphanumerische Zeichen zum Einsatz anstatt allen 255 Zeichen – das reduziert die möglichen Passwörter erheblich

    (Für eine BruteForce-Attacke reicht ein Packet)

WLAN Security > WEP und seine Probleme

- Sicherheitslücken/Schwächen in WEP (Forts.)
  - Kein Replay-Schutz

    Alte Packete werden nicht erkannt. Mitgeschnittene Packete können einfach wieder gesendet werden.

  - RC4 – streambasierte Verschlüsselung (Byte für Byte)
    - Basis für Known-Plaintext-Angriffe, weil die Position des Klartext der Position im verschlüsselten Text entspricht.
    - Es gibt Schlüssel die schwächer sind als andere. Die Schwäche ist schon lange bekannt, in WEP trotzdem nicht berücksichtigt. FMS-Attacke (*Weaknesses in the Key Scheduling Algorithm of RC4*)

WLAN Security > WEP und seine Probleme

- Sicherheitslücken/Schwächen in WEP (Forts.)
  - Schwacher RC4-Initialisierungs-Vektor (IV)

    hat nur 24Bit = ca. 17 Mio. Werte – wiederholt sich spätestens nach ca. 7h bei hoher Last bei 11Mbit und einem Client

    ```
    C1 = KT1 XOR RC4( IV1, Schlüssel )
    C2 = KT2 XOR RC4( IV2, Schlüssel )
    ```

    wenn jetzt IV1 = IV2

    ```
    C1 XOR C2 = KT1 XOR KT2
    ```

    Basis für eine Known-Plaintext-Attacke, weil die ersten Bits in einem Packet von Packet-Header sein müssen, dessen Aufbau man kennt

WLAN Security > WEP und seine Probleme

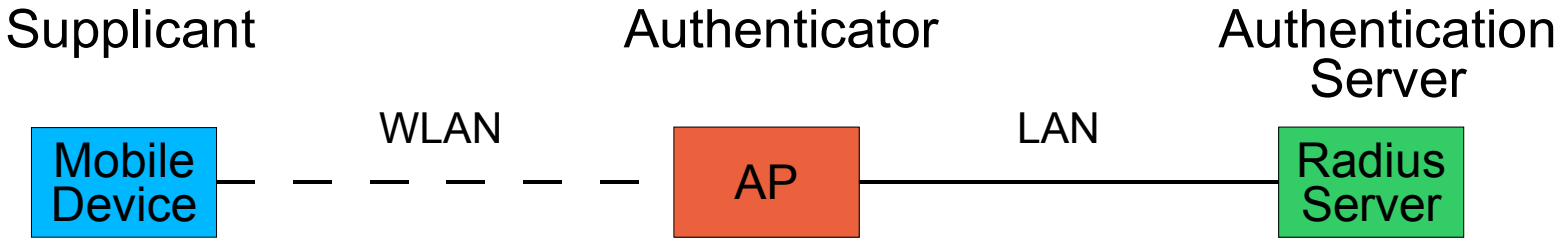
- Sicherheitslücken/Schwächen in WEP (Forts.)
  - FMS-Attacke
    - Entdeckt von Fluherer, Mantin, Shamir
    - Bestimmte Schlüssel erzeugen berechenbare Bits in dem RC4-Keystream (Schlüssel = IV + Passwort)
    - Der Angreifer sammelt Frames mit schwachen IVs, einige Tools sind ab 60-80 in der Lage den WEP-Schlüssel wiederherzustellen (das kann Stunden bis Tage dauern)
    - Die einfache Lösung wäre die ersten 256 Bytes des Keystreams zu verwerfen, nur wurde das in WEP (und deren Hardware) nicht berücksichtigt

WLAN Security > WEP verbessert

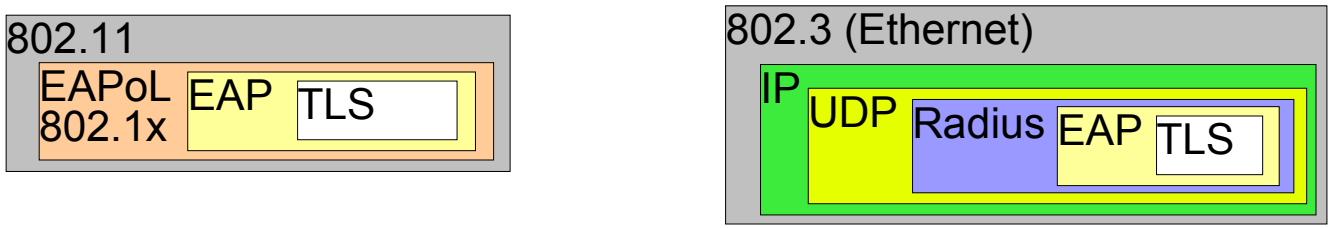
- WEP Sicherheit verbessern
  - Sicherheit steigt enorm durch dynamische Sitzungsschlüssel.
  - Jeder Client hat einen eigenen Schlüssel!
  - Der Schlüssel wird immer wieder erneuert
  - Gegenseitige Authentifizierung anhand von Zertifikaten (möglich)
  - IEEE 802.1x (Portbased Access Control)
    - EAP (Extensible Authentication Protocol) – Authentifizierung
    - TLS (Transport Layer Security)
    - Radius – Authentifizierungs-Protokoll und Server
    - EAPoL – EAP over Lan (Ethernet, Token Ring)
      - Kapselt EAP-Nachrichten um sie im LAN transportieren zu können EAP war ursprünglich für PPP konzipiert
      - Definiert noch einige Einfache Nachrichten

WLAN Security > WEP verbessert

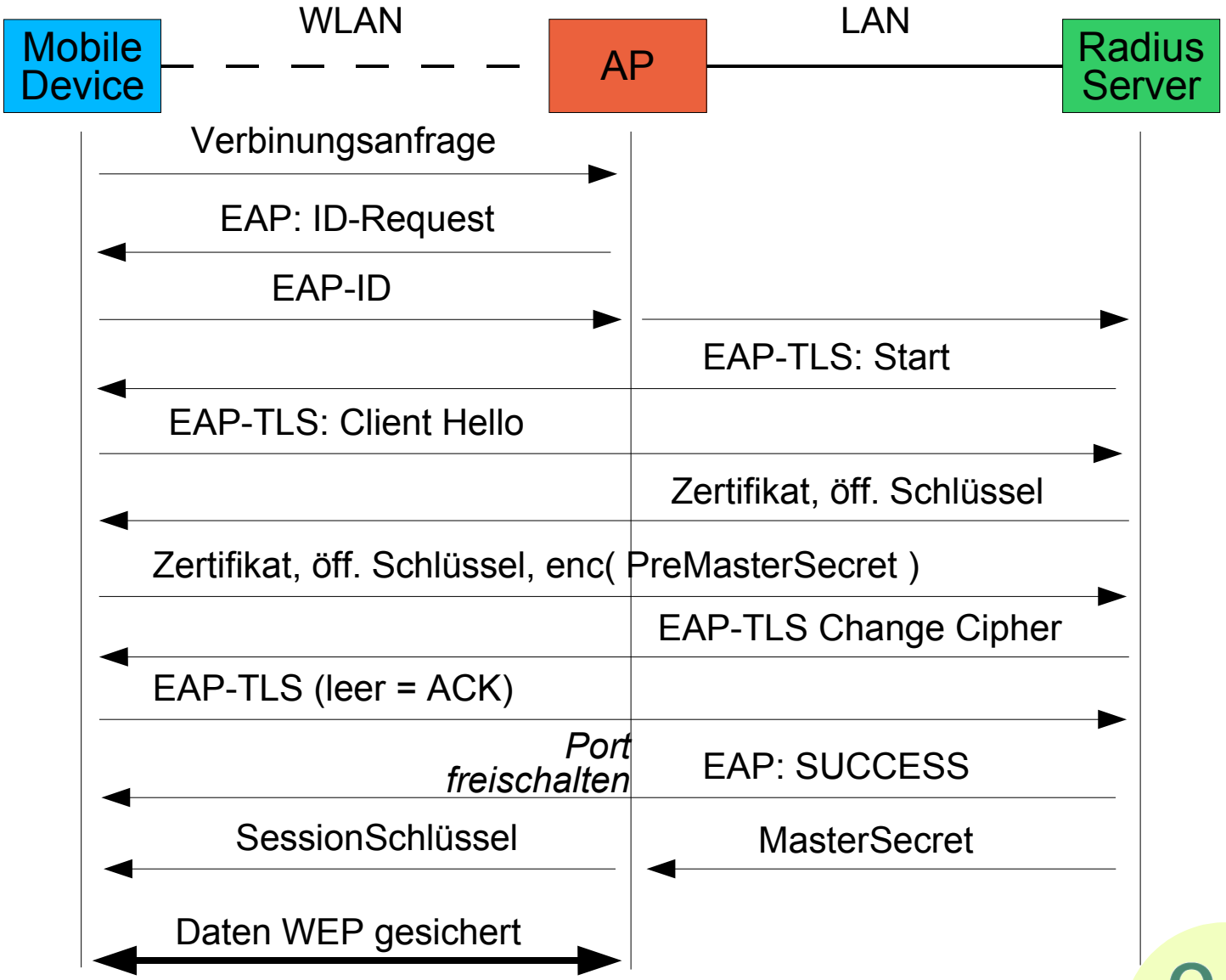
- IEEE 802.1x (Portbased Access Control)
  - Spezifiziert die Übertragung von Authentifizierungs-Anfragen
  - EAPoL – wie EAP in LANs transportiert werden soll
  - 802.1x kann mehrere Authentifizierungs-Protokolle kapseln
  - 802.1x kennt folgende 3 Parteien



Protokolle bei der Authentifizierung



WLAN Security > WEP verbessert



WLAN Security > WEP verbessert

- FreeRadius
  - Unter der GPL freigegeben
  - Läuft auf *nix-Systemen
  - Unterstützt viele EAP-Varianten (EAP-TLS, EAP-TTLS, PEAP, Cisco-LEAP, EAP-SIM)
  - Kann Benutzerdaten von einem LDAP-Server abfragen, unterstützt Zugriff auf postgreSQL, mySQL, Oracle

- WEP mit EAP-TLS
  - Eine gute Lösung für bestehende Hardware
  - Geeignet für Abgeschlossene Benutzer-Umgebungen
  - Großer Aufwand bei ständig wechselnden Clients
  - Probleme mit dem 802.1x-Stack für MacOS10.2, bestimmte Wlan-Karten unter Linux

WLAN Security > WEP

- WLan Tools (Automatische Suche und Klassifizierung von WLANs)
  - Netstumbler: Windows
  - Wellenreiter: Linux

- WEP-cracking Tools (FMS-Attacke)
  - BSD-airtools: nur BSD. Angeblich das schnellste Tool. Besserer Algorithmus. [www.dachb0den.com]
  - Airsnort: Linux

- Allgemeine Netzwerk-Monitoring-Tools
  - Ethereal: *nix/Windows – Umfangreiches Monitoring-Tool
  - Kommandozeile: netstat, iptraf

WLAN Security > WPA

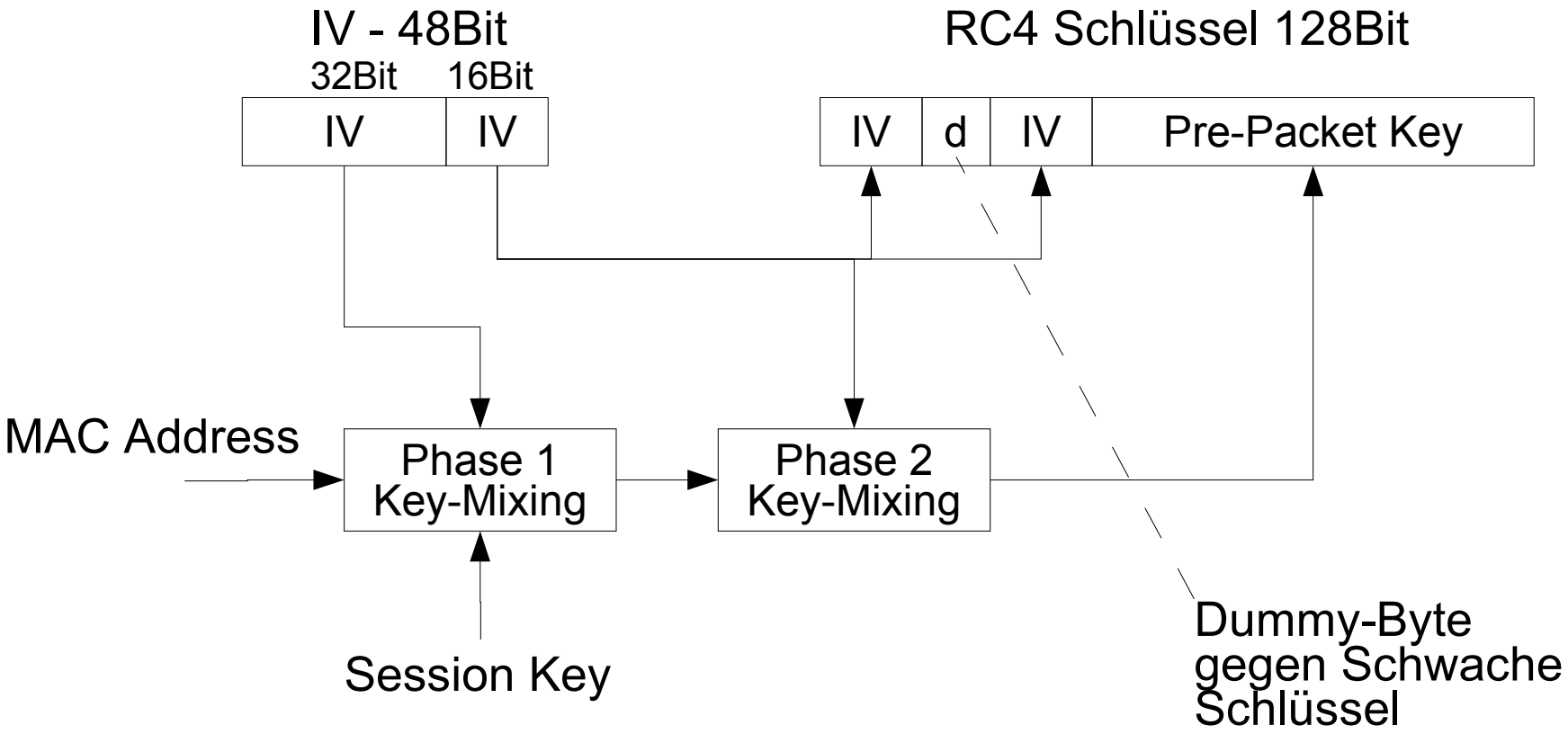
- WPA – Wi-Fi Protected Access
  - Entworfen im Zuge mit RSN (IEEE 802.11i) jedoch mit alter Hardware kompatibel
  - WPA ist eine Untermenge von RSN
  - Sicherheitsprotokoll ist TKIP
  - Verschlüsselungsalgorithmus – nur RC4 wegen der Abwärtskompatibilität verfügbar, weil dieser von der Hardware berechnet wird

WLAN Security > WPA

- TKIP – Temporarl Key Integrity Protocol
  - Löst WEP als Sicherheitsprotokoll ab, kann auch in RSN verwendet werden
  - Ist Hardware-kompatibel mit WEP-Geräten
  - Beseitigt die großen Schwächen von WEP durch
    - Es werden schwache IVs verhindert
    - 56Bit – 1Byte* = 48Bit langer IV (*schwaches RC4-Byte)
    - IV wird gleichzeitig als Sequenznummer (Replay-Schutz) verwendet
    - Key-Management für Broadcast-Keys
    - Schlüssel wird bei jedem Packet verändert
    - Message-Integrity gegen Veränderung

WLAN Security > WPA

- TKIP – Temporal Key Integrity Protocol
  - Schlüsselerzeugung für jedes Packet
    - Vermeidet einen konstanten Schlüssel, Abwehr gegen FMS



WLAN Security > RSN

- RSN (Robust Security Network)
  - Standard IEEE 802.11i – kürzlich freigegeben
  - Inkompatibel mit alter Hardware
  - Bietet starke Sicherheit
  - Verwendete Protokolle
    - Authentifizierung, Zugriff – IEEE 802.1x, EAP (EAPoL, EAPoR), Radius, Kerberos
    - Sicherheitsprotokolle – TKIP, AES-CCMP
    - Verschlüsselung – Alle modernen Blockcipher

WLAN Security > RSN

- AES-CCMP
  - AES – Advanced Encryption Standard
  - CCMP – Counter-Mode Cipher-Block-Chaining Message Authentication Code Protocol
  - Stärker als TKIP
    - AES statt RC4
    - Längerer Schlüssel 128Bit statt 40/104Bit
    - Längerer MIC (Message Integrity Check)

Crypto-Biliotheken

- Crypto-Bibliotheken mit offenem Code
  - Cryptix
    - Bibliotheken für Java (Cryptix JCE) und C/C++
  - Crypto++
    - Eine umfangreiche C++ Bibliothek mit allen gängigen Algorithmen und Hash-Funktionen
    - Enthält auch ECC-Algorithmen
  - OpenSSL
    - Assymetrische Verschlüsselung
    - Zertifikats-Erzeugung
  - Sophie
    - Public-Key Algorithmen für Python
  - Bouncy-Castle
    - Java JCE Implementierung