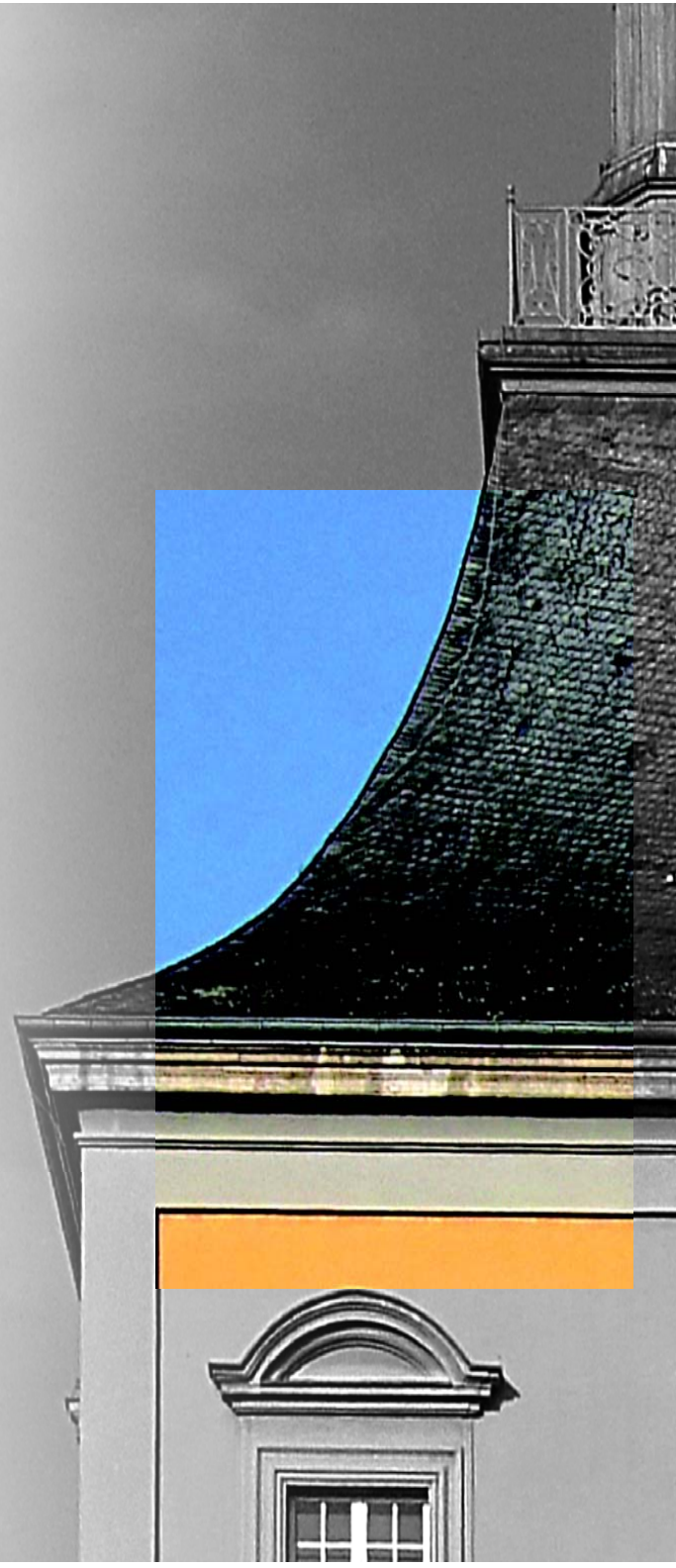
UNIVERSITÄT BONN

VORLESUNG

NETZWERKSICHERHEIT

SOMMERSEMESTER 2021

MO. 14-16 UHR



UNIVERSITÄT BONN

KAPITEL 6

VPN

EIGENSCHAFTEN PRIVATER NETZWERKE

Zugriff eingeschränkt (physikalische Anwesenheit notwendig)

Vertrauen

- Netzwerkinfrastruktur
  - Router / Switches
  - Access Points
- Server (Fileserver, Webserver, Mailserver, …)
- Arbeitsplatzrechner
- Scanner / Drucker
- Teilnehmer
  - Mitarbeiter
  - Administratoren
  - Gäste (evtl. in eigenem VLAN)

PRIVATE NETZWERKE

Private Netzwerke sind

- lokale Netzwerke (LAN) innerhalb eines Gebäudes
- WLAN mit Zugangsbeschränkung
- “Dark-Fiber”-Verbindungen zwischen Geschäfts-Standorten
- ~~Das Internet…~~

PRIVATE NETZWERKE

- “Nicht-Private Netzwerke”?
  - Offensichtlich: Netzwerke, von jemand anderem
    - Öffentliche WLANs
    - Das Internet
  - Vertrauen? Eher nicht!
  - Aber: Insb. das Internet ist schnell und günstig (im Vergleich zu “Dark-Fiber”)
  - Vertrauliche Unternehmenskommunikation über das Internet
    - Erlaubt Homeoffice (insb. zu Corona-Zeiten)
    - Erlaubt “Roadwarrior” (Reisende Verkäufer, Kundendienst, …)
    - Aber:
      - Kommunikation (Metadaten und Inhalte) einsehbar
      - Kommunikation manipulierbar
      - Was ist mit Zugriff auf interne Ressourcen (Intranet, Drucker, …)

UNIVERSITÄT BONN

PRIVATE NETZWERKE

Vertrauenswürdige Erweiterung eines privaten Netzwerks

- Unterhalt eigener Leitungen zwischen Standorten
  - Keine Anbindung aller Mitarbeiter im Homeoffice
  - Außerhalb des eigenen Geschäftsbereichs und teuer
- Miete einer MPLS-Verbindung eines Telekommunikationsanbieters
  - Geswitchte Verbindung meiner Standorte durch das Netz des Telkos “IPVPN”
  - Ist der Anbieter vertrauenswürdig?

- Einrichtung eines Virtuellen Privaten Netzwerks (VPN)
  - Kommunikation (abgesichert: verschlüsselt, authentifiziert) über das Internet
  - Günstig
  - Flexibel an unterschiedlichen Standorten einsetzbar
  - Vertrauen in den Hersteller und die Implementierung

UNIVERSITÄT BONN

VIRTUELLE PRIVATE NETZWERKE

Anforderungen an VPNs

- Selbst, wenn jemand übertragene Pakete mitlesen kann, soll der Inhalt der Verbindung vertraulich bleiben
- Pakete können nicht ohne Kenntnisnahme durch Dritte verändert werden
- Es können keine Pakete oder Informationen in eine Verbindung eingeschleust werden
- Es können einzelne Pakete anhand von Metadaten unterdrückt werden, es können aber nicht spezifische Inhalte gezielt unterdrückt werden

Wahrung der o.g. Schutzziele

- Verschlüsselung von Paketen zum Schutz der Vertraulichkeit
- MAC-Funktion zum Schutz der Integrität und Authentizität von Inhalten
- Sichere Schlüsselverteilung

VIRTUELLE PRIVATE NETZWERKE

Tunnel

- Grundlegendes theoretisches Konzept für VPNs
- Idee eines virtuellen “Overlay”-Netzwerks über das Internet
  - D.h. bei zwei unterschiedlichen Standorten (etwa Köln und Bonn), ist die Netzwerkdistanz zwischen zwei VPN-Routern lediglich ein Hop
    - Unabhängig von der tatsächlichen Anzahl an Hops durch das Internet
  - Pakete werden zwar von Routern im Internet weitergeleitet, dabei aber nicht verändert (etwa die TTL bleibt erhalten)

UNIVERSITÄT BONN

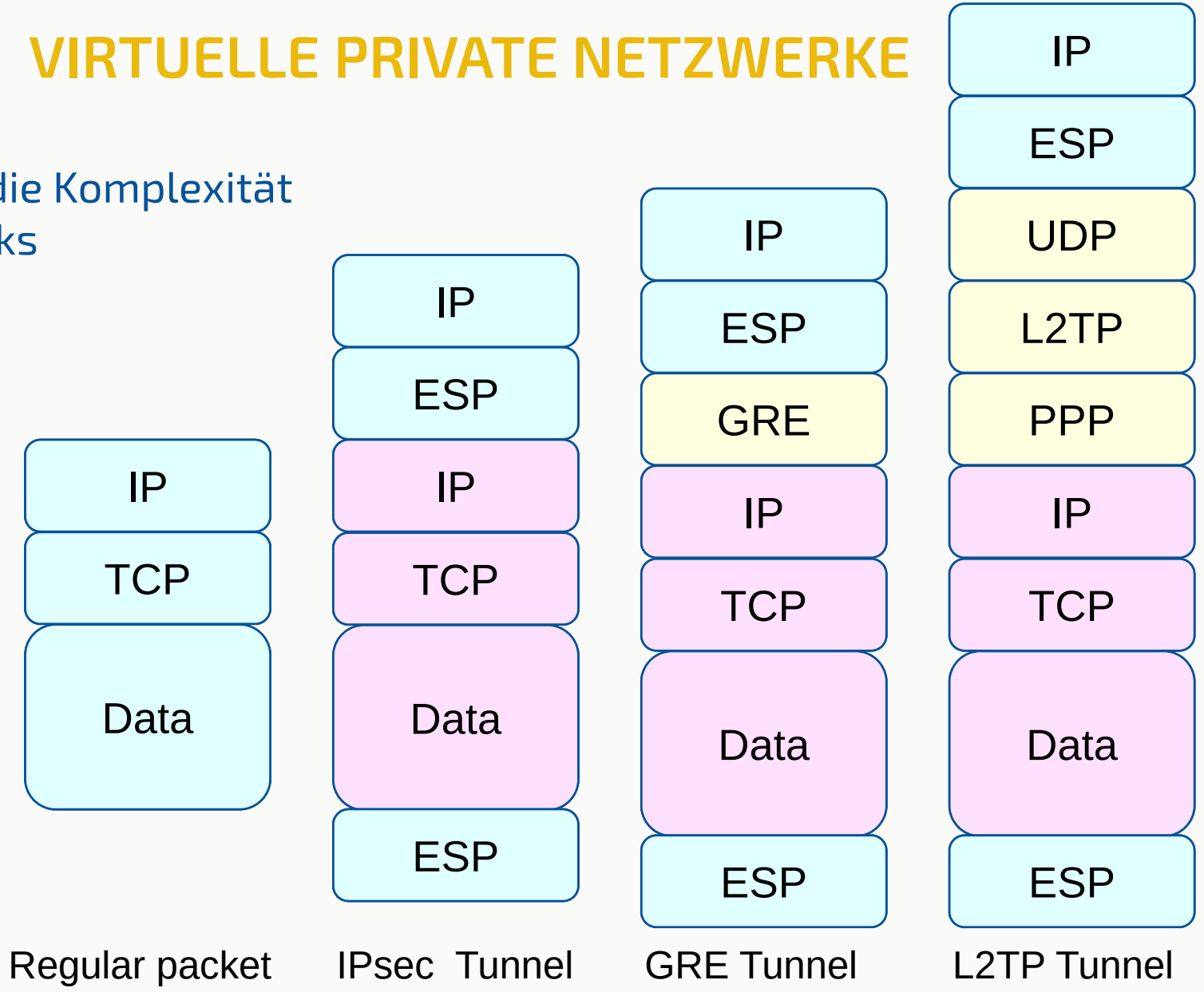
VIRTUELLE PRIVATE NETZWERKE

Tunnel

- Symmetrische Verschlüsselung & Message Authentication Codes (MAC)
- Sicherer Schlüsseltausch beim Verbindungsaufbau
- Schlüsselverteilung analog zum authentifizierten Netzzugang
  - Zugangskontrolle: Wer Zugang benötigt, bekommt einen Schlüssel
  - Authentifikation des Tunnels
  - Vertraulichkeit des Schlüsseltauschs
    - Keine Schlüssel-Metadaten (z.B. Benutzerkennung, E-Mail, …) im Klartext
- Implementiert mittels Key-Derivation-Funktionen und Diffie-Hellman- oder RSA-Verfahren zum Aushandeln von Verbindungsschlüsseln

VIRTUELLE PRIVATE NETZWERKE

Tunnel vergrößern die Komplexität des Protokollstacks



UNIVERSITÄT BONN

VIRTUELLE PRIVATE NETZWERKE

Tunnel (Beispiel: IPSec)

- IPSec ist nur eine Möglichkeit, virtuelle Overlays zu erstellen
- Es gibt weitere Tools, die einen einfachen Tunnelaufbau ermöglichen
  - GRE – Generic Routing and Encapsulation
    - Fügt einen eigenen Header (Nummer 47) hinzu
    - Hauptsächlich von Cisco für Site-2-Site-VPNs verwendet
  - L2TP – Layer 2 Transport Protocol
    - Baut eine “Tunnel”-Verbindung mittels PPP über UDP (Port 1701) auf
    - Hauptsächlich verfügbar für Windows, Mac, iOS, Android
- Aber: keine Verschlüsselung
- IPSec-Verschlüsselung sollte aufbauend genutzt werden
  - GRE und L2TP erzeugen beim Verbindungsaufbau Netzwerk-Interfaces
- GRE- und L2TP-Tunnel haben mehr “per packet overhead” als IPSec

VIRTUELLE PRIVATE NETZWERKE

- Tunnel-Verschüsselung
  - Symmetrische Verschlüsselung mit zwei Funktionen
    - m' = F(k,m) - Symmetrische Versschlüsslungsfunktion
    - m = F^{-1}(k,m') - Symmetrische Entschlüsselungsfunktion
    - Dabei gilt: Wenn m' = F(k,m), dann m = F^{-1}(k,m')
  - Anforderungen an die Verschlüsselung
    - Große Schlüssel, so dass der Schlüssel nicht erraten werden kann
    - Keine schwachen Schlüssel
    - Große Blöcke, so dass der Initialisierungsvektor nicht mehrfach genutzt wird
    - Ausgabe soll nicht von zufälligen Daten unterscheidbar sein
      - Formell: Wenn die Cipher-Stärke n Bits ist, dann soll es 2^n - Schritte benötigen, um zu zeigen, dass die Daten nicht zufällig sind

UNIVERSITÄT BONN

VIRTUELLE PRIVATE NETZWERKE

- Tunnel-Integrität
  - Hashfunktionen mappen beliebig lange Nachricht auf Digest fester Länge
    - d = H(m)
    - Üblicherweise: 128 – 512 Bits
  - Für jede Hashfunktion soll das berechnete Digest nicht von zufälligen Daten unterschieden werden
  - Wenn $H(m_1) = H(m_2)$, dann $m_1 = m_2$
    - Gilt natürlich nicht!
      - Kollisionen müssen selten sein (versehentliche oder bewusst herbeigeführte)

VIRTUELLE PRIVATE NETZWERKE

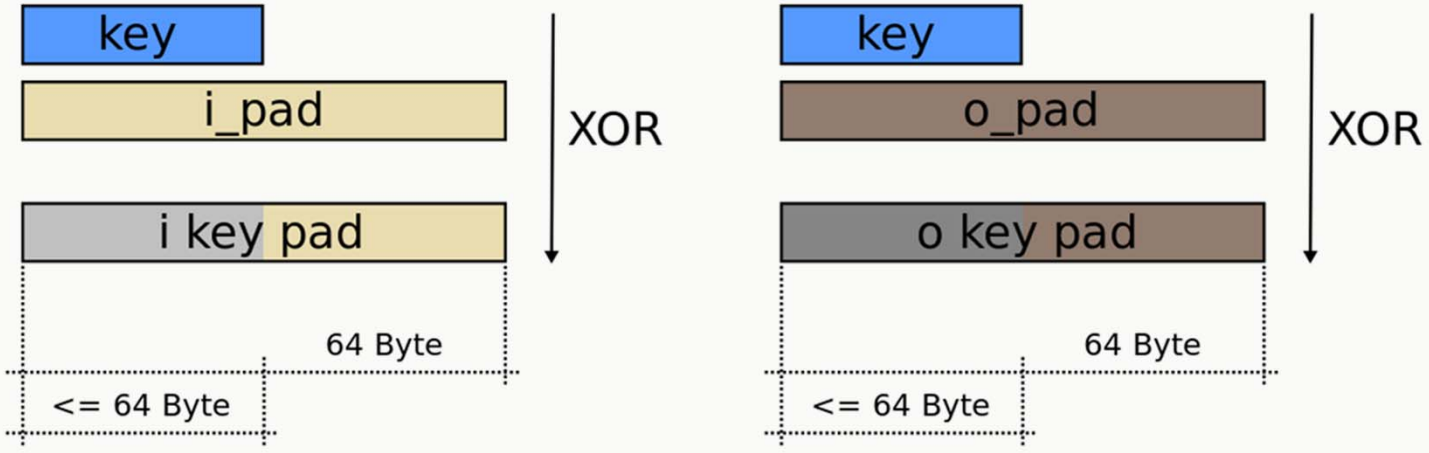
- Tunnel-Integrität
  - Angriffe gegen Hashfunktionen
    - Urbild-Angriff (1)
      - Für ein gegebenes Digest d soll eine Nachricht m gefunden werden, so dass d = H(m)
    - Urbild-Angriff (2)
      - Für eine gegebene Nachricht m soll eine weitere Nachricht m' gefunden werden, für die gilt
        $m \neq m'$
        $H(m) = H(m')$

VIRTUELLE PRIVATE NETZWERKE

- Tunnel-Integrität
  - Kollisionen erzwingen
    - Es sollen zwei Nachrichten m, m' gefunden werden, für die gilt
      H(m) = H(m')
  - Die Anzahl der Bits ist entscheidend (Geburtstagsproblem)
  - Mit n Bits wird eine Kollision erwartet nach $\sqrt{2^n} = 2^{n/2}$ Nachrichten
    - MD5
      - 128 Bits
    - SHA-1
      - 160 Bits
    - SHA-256
      - 256 Bits

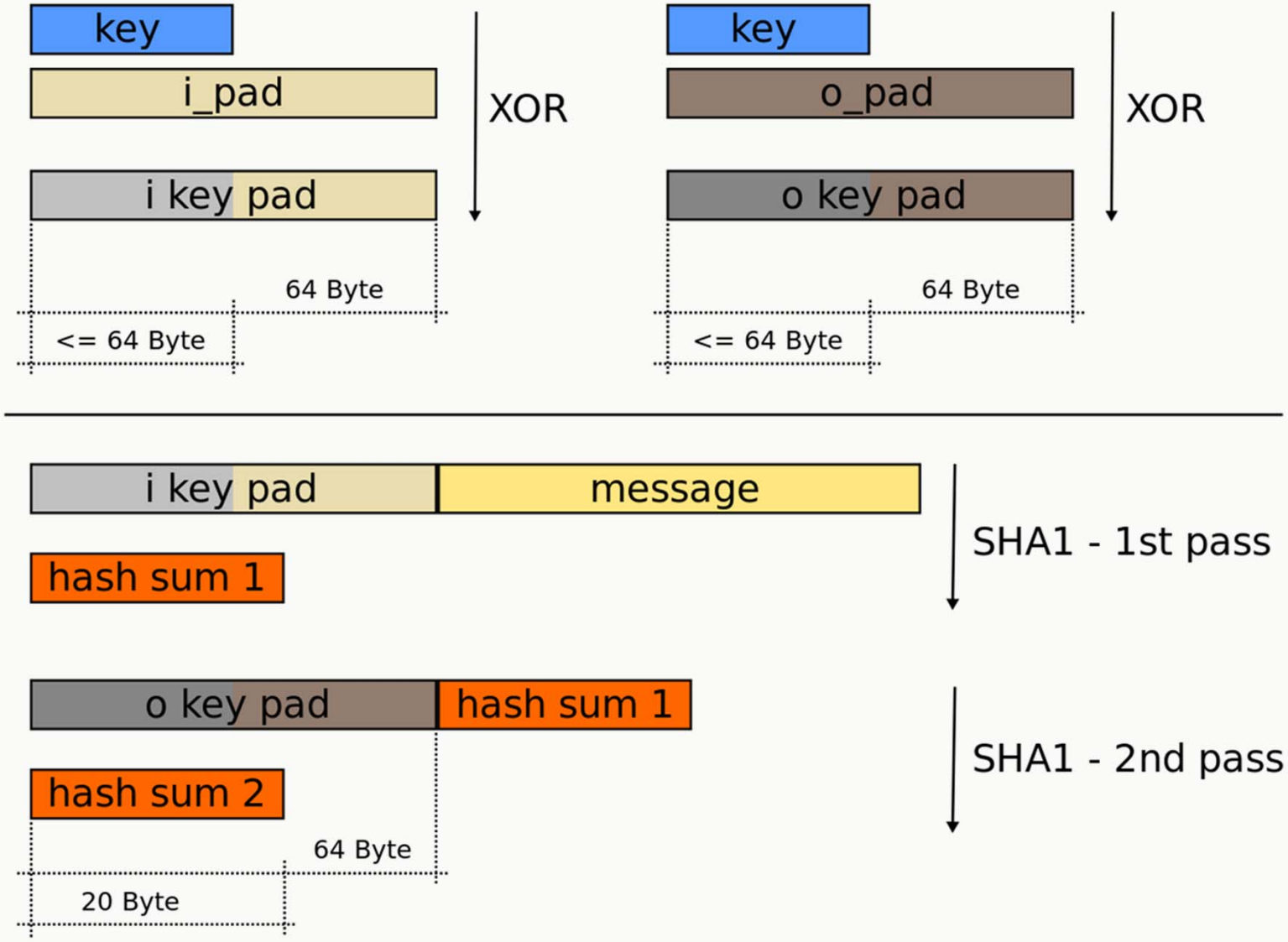
VIRTUELLE PRIVATE NETZWERKE

- Tunnel-Integrität (HMAC)
  - Keyed Hash-Funktionen (als Message Authentication Code)
    - Wie Hash-Funktionen, nur mit einem weiteren Parameter, dem Schlüssel
    - Für die Erstellung und Verifikation eines HMAC wird der Key benötigt
      - Dieser ist i.d.R. ebenfalls geheim



VIRTUELLE PRIVATE NETZWERKE

- Tunnel-Integrität (HMAC)



UNIVERSITÄT BONN

IPSEC

IPSec ist ein Sicherheitsprotokoll für das Internet

- Arbeitet auf Layer 3 und ist daher transparent für Anwendungen
- In allen gängigen Betriebssystemen implementiert
- Ermöglicht Verschlüsselung und Authentifikation von IP-Paketen
  - Bestandteil der IPv6 Extension Header
  - Authentication Header (AH)
    - Paket-Authentifikation mittels HMAC (Paket stammt vom Absender)
    - Erlaubt Integritätsprüfung (Paket wurde nicht verändert)
  - Encapsulated Security Protocol (ESP)
    - Paket Authentifikation mittels HMAC
    - (Symmetrische) Verschlüsselung sichert Vertraulichkeit

IPSEC

Begriffserklärung

- Peer
  - Computer, der IPSec unterstützt
- Traffic-Selector
  - Beschreibt eine Seite der IPSec-Verbindungen
    - IP-Adressen
    - Layer-4-Protokoll (TCP, UDP, ICMP, …)
    - Ports (falls für L4 relevant)
- Action
  - Beschreibt, was mit den Paketen beim Versand passieren soll
    - Bypass: Pakete ohne Änderung weiterleiten
    - Drop: Pakete verwerfen
    - Protect: IPSec auf das Paket anwenden

IPSEC

Begriffserklärung

- Methode / Modus
  - Definiert, wie Pakete geschützt werden (ESP? / AH?)
  - Tunnel- oder Transportmodus (IPv6-Kapitel)
- Security Association (SA)
  - Beschreibt, was mit Traffic passiert
  - Enthaltene Informationen
    - Source traffic selector
    - Destination traffic selector
    - Methods & Keys
    - Peer
    - Richtung (inbound / outbound)
    - Time and/or volume limited
    - OS-eindeutiger 32-Bit-Wert (Security Parameter Index)

IPSEC

Begriffserklärung

- Security Policy (SP)
  - Source traffic selector
  - Destination traffic selector
  - Request Flags
  - Action
  - Ähnlich einer Security Association aber unidirektional
    - Implementierungen verbinden in- und outbound SPs und erzwingen ähnliche Aktionen und Methoden
- Security Policy Database (SPD)
  - Liste von SP-Elementen
  - Ändern sich nicht dynamisch (es handelt sich um definierte Richtlinien)
  - Existierende SP bedeuten nicht, dass auch passende SA existieren

IPSEC

Begriffserklärung

- SPD Cache
  - Hält SP-Einträge aus der Datenbank im Speicher
  - Einträge müssen nicht (immer) mit der SPD übereinstimmen
- Security Association Database (SAD)
  - Beinhaltet alle Security Associations
  - Inbound-Tabelle sortiert nach SPI
  - Outbound-Tabelle sortiert nach Traffic Selectoren
  - Jeder Eintrag hat eine Entsprechung im SPD-Cache
- PAD - Peer Authentication Database
  - Beschreibt Peers und wie sie sich ausweisen (IP-Addressen, DNS-Namen)
  - Hält Information darüber, wie Peers authentifiziert werden (IKE, Zertifikate, PSK, ...)

IPSEC-ABLAUF

Aufbau eines IPSec-Tunnels mittels Internet Key Exchange-Protokoll (IKE)

- 2 Phasen: **IKE Phase I** und IKE Phase II

- ISAKMP-Tunnel (Internet Security Association and Key Management Protocol)
- Austausch von Management-Daten (Grundlage für den Aufbau des zweiten Tunnels, Versand von Keepalives, …)

IPSEC-ABLAUF

Aufbau eines IPSec-Tunnels mittels Internet Key Exchange-Protokoll (IKE)

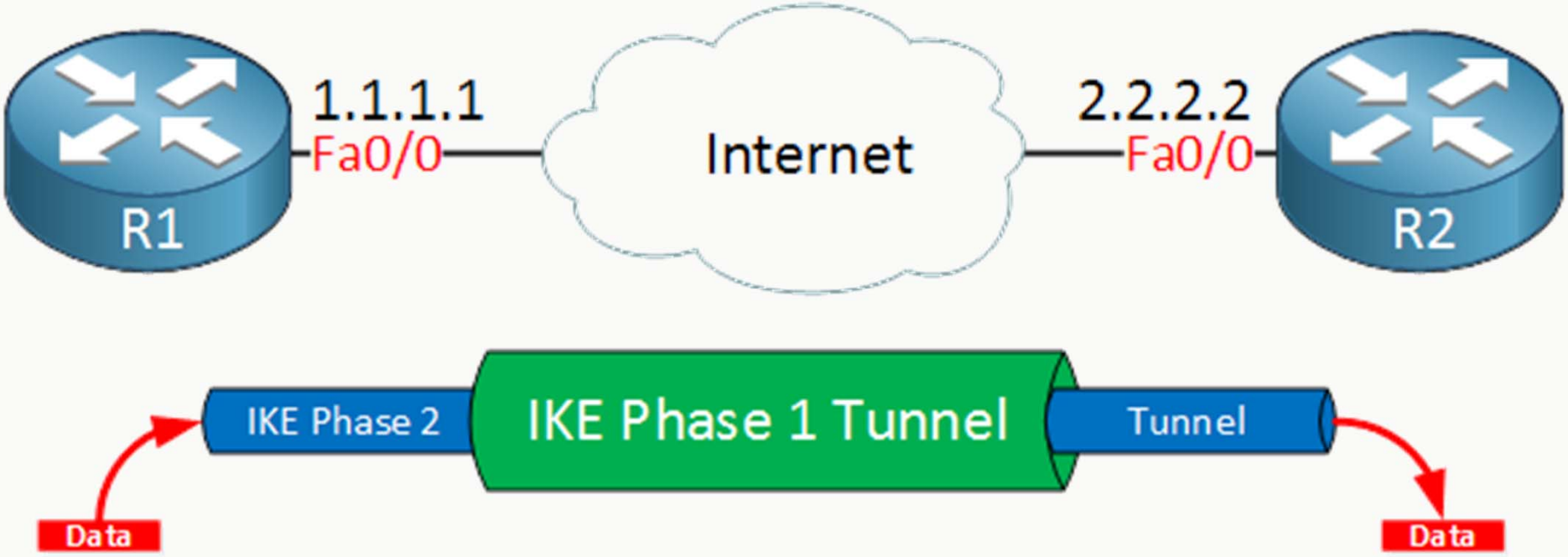
- 2 Phasen: IKE Phase I und **IKE Phase II**

- Phase-II-Tunnel oder IPSec-Tunnel auf Basis der Security Associations

IPSEC-ABLAUF

Aufbau eines IPSec-Tunnels mittels Internet Key Exchange-Protokoll (IKE)

- 2 Phasen: IKE Phase I und IKE Phase II



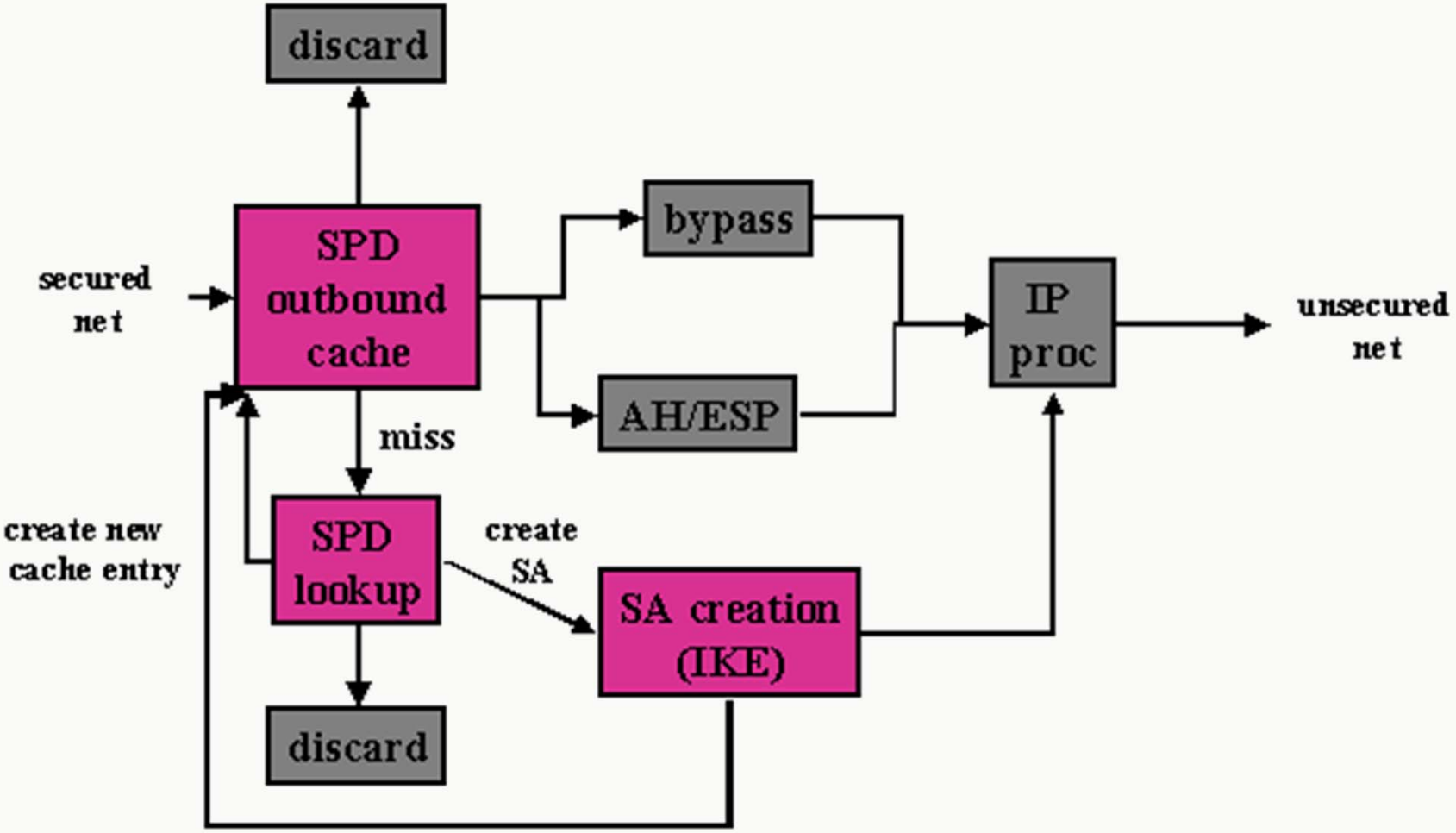
- Austausch von Payload über IPSec-Tunnel
- Verschlüsselung und Authentification im Anschluss mittels AH/ESP-Protokoll

UNIVERSITÄT BONN

IPSEC-ABLAUF

- Annahme: SPD wird für die Verarbeitung aller Pakete befragt
  - Default „Drop“ oder „Bypass“
- Ein Daemon erstellt Einträge im SPD-Cache und in der SAD (entspr. lokaler Konfiguration)
- IPSec-Stack des Betriebssystems fragt bei der Paketbearbeitung SAs von diesem Daemon ab
- Ausgehende Paketverarbeitung
  - Paket wird gegen SPD-Cache getestet
  - Match: Folge Anweisung der SA
  - Kein Match: teste Paket gegen SPD
    - Falls „Drop“ oder „Bypass“ führe Aktion aus, füge Eintrag in SPD-Cache hinzu
    - Falls „Protect“, frage SA vom Daemon ab und handle entsprechend
      - Füge Einträge der SAD und dem SPD-Cache hinzu

IPSEC-ABLAUF

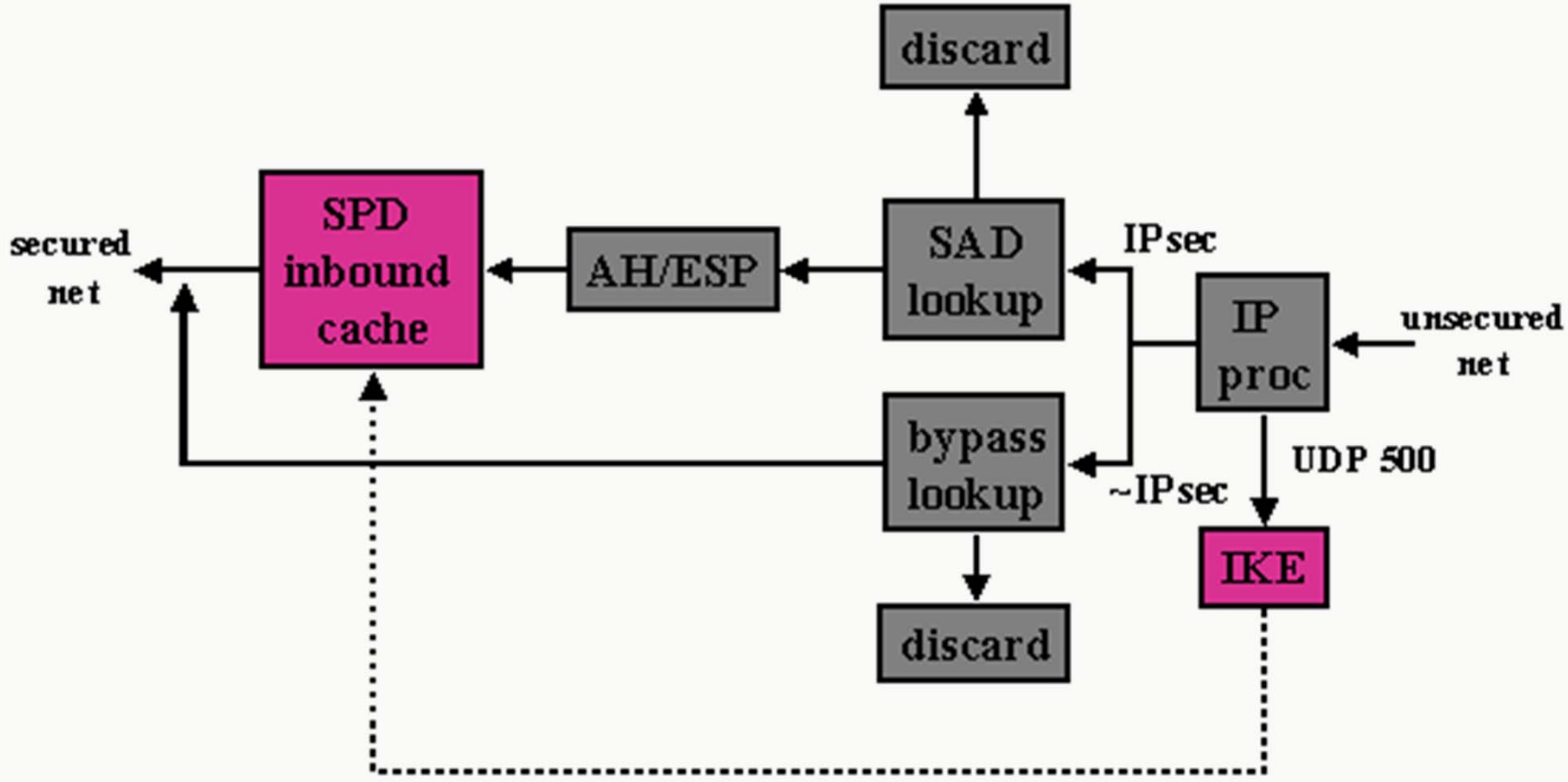


UNIVERSITÄT BONN

IPSEC-ABLAUF

- Eingehende Paketverarbeitung
  - SPI aus dem Paket-Header wird gesucht
  - Matching auf passenden Eintrag in der SAD
  - Kein Match: Drop
  - Match: Entschlüsseln des Payloads und Prüfen der Authentifikation
  - Prüfe im SPD-Cache, ob das entschlüsselte Paket einer Security Association (SA) aus der Security Association Database (SAD) entspricht
    - Falls nicht: Drop

IPSEC-ABLAUF



WEITERE VPN-IMPLEMENTIERUNGEN

Wireguard

- (schauen wir uns gleich näher an)

OpenVPN

- (hat vermutlich jeder schon einmal genutzt?)

Microsoft Always-On-VPN / DirectAccess

- erlaubt unterschiedliche Technologien für das VPN
- „Always-On“ heißt, wird schon vor der Anmeldung gestartet
  - Wichtig für den Betrieb in Windows-Domänen

UNIVERSITÄT BONN

WIREGUARD

Motivation

- Andere Lösungen sind zu groß und zu kompliziert
  - Wireguard soll einfach zu auditieren sein!
  - Wireguard soll ein einfaches Interface haben
    - Linux-Kernel basierter Tunnelaufbau
      - Erstellen eines Netzwerk-Interfaces
  - Authentifikation und Schlüsseltausch mittels Public-Key-Crypto
  - Einfache Konfiguration
- Wireguard ist Bestandteil des Linux-Kernels
  - Es gibt Userspace-Implementierungen (z.B. für Windows)
    - deutlich weniger performant

UNIVERSITÄT BONN

WIREGUARD

Einfach zu auditieren?

OpenVPN	Linux XFRM	StrongSwan	SoftEther	WireGuard
116,730 LoC Plus OpenSSL!	119,363 LoC Plus StrongSwan!	405,894 LoC Plus XFRM!	329,853 LoC	**3,771 LoC**

Einfaches Benutzerinterface

- Erstellen eines Netzwerkinterfaces

```
tschita ~ # ip link add wg0 type wireguard
tschita ~ # ifconfig wg0
wg0: flags=144<POINTOPOINT,NOARP>  mtu 1420
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

UNIVERSITÄT BONN

WIREGUARD

Authentifikation und Schlüsseltausch mittels Public-Key-Crypto („Cryptokey-Routing“)

- Userspace-Tools („wireguard-tools“-Paket)
  - wg – Konfigurationswerkzeug

```
tschita ~ # wg genkey > wg.priv
Warning: writing to world accessible file.
Consider setting the umask to 077 and trying again.
tschita ~ # wg pubkey < wg.priv > wg.pub
```

  - wg-quick – Tunnel-Steuerungswerkzeug

```
tschita /etc/wireguard # wg-quick down wg0
[#] ip link delete dev wg0
tschita /etc/wireguard # wg-quick up /etc/wireguard/wg0.conf
[#] ip link add wg0 type wireguard
[#] wg setconf wg0 /dev/fd/63
[#] ip -4 address add 10.10.0.1/24 dev wg0
[#] ip link set mtu 1420 up dev wg0
```

WIREGUARD

Erfolgreiche Verbindung

```
tschita /etc/wireguard # wg show
interface: wg0
  public key: rupfGRNxQPJBWLPP4reb9w5k1iVjiwdI3hutRySeKQg=
  private key: (hidden)
  listening port: 48443

peer: xp8BuuERNAy1epqYJG9dmd9rltk01EzST4kPWvrfrgM=
  endpoint: 131.220.240.173:48443
  allowed ips: 10.10.0.2/32
  transfer: 0 B received, 148 B sent
  persistent keepalive: every 25 seconds
```

UNIVERSITÄT BONN

WIREGUARD

Konfiguration CryptokeyRouting

- Grundlegendes Konzept ist die Assoziation zwischen einem öffentlichen Schlüssel eines Peers (Clients) und der IP, die der Peer verwenden darf
- Ein Wireguard-Interface (lokal) hat
  - Einen privaten Schlüssel
  - Einen UDP-Listen-Port
  - Eine Liste von Peers
- Ein Peer
  - Wird über den öffentlichen Schlüssel identifiziert
  - Liste assoziierter (erlaubter) IP-Adressen
  - Optional eine Endpunkt-IP inkl. Port

UNIVERSITÄT BONN

WIREGUARD

Server Config

```
[Interface]
PrivateKey =
yAnz5TF+lXXJte14tji3zlMNq+hd2rYUIgJBgB3fBmk=
ListenPort = 41414

[Peer]
PublicKey =
xTIBA5rboUvnH4htodjb6e697QjLERt1NAB4mZqp8Dg=
AllowedIPs = 10.192.122.3/32,10.192.124.1/24

[Peer]
PublicKey =
TrMvSoP4jYQlY6RIzBgbssQqY3vxI2Pi+y71lOWWXX0=
AllowedIPs = 10.192.122.4/32,192.168.0.0/16
```

Client Config

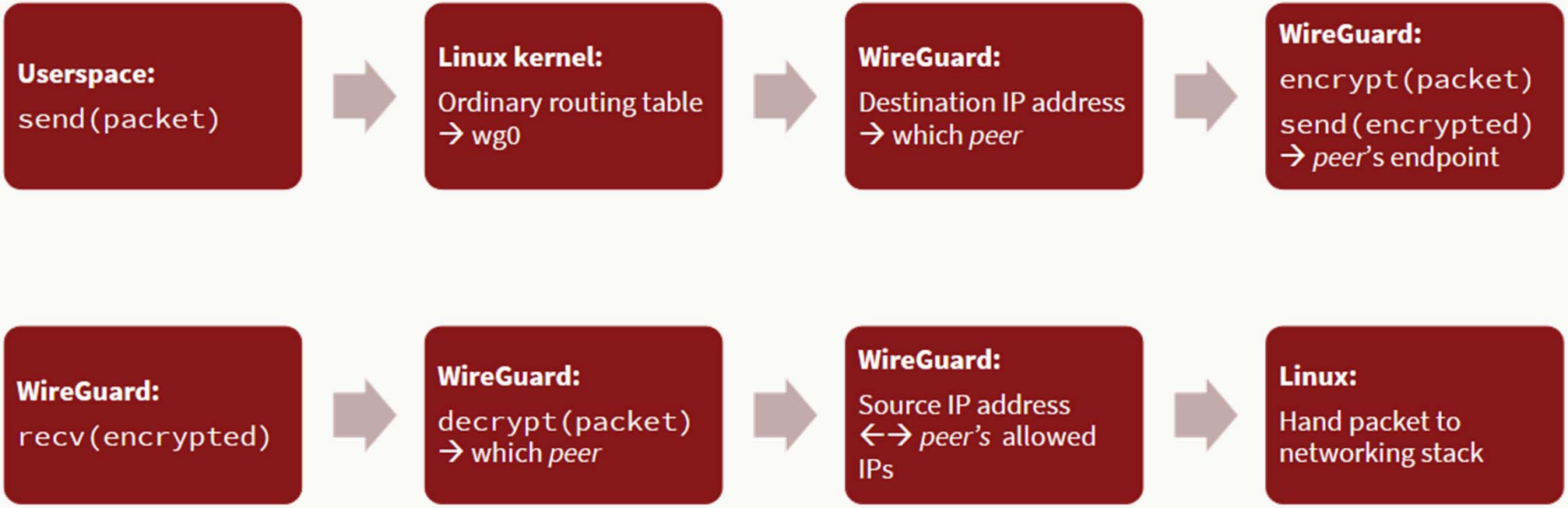
```
[Interface]
PrivateKey =
gI6EdUSYvn8ugXOt8QQD6Yc+JyiZxIhp3GInSWRfWGE=
ListenPort = 21841

[Peer]
PublicKey =
HIgo9xNzJMWLKASShiTqIybxZ0U3wGLiUeJ1PKf8ykw=
Endpoint = 192.95.5.69:41414
AllowedIPs = 0.0.0.0/0
```

UNIVERSITÄT BONN

WIREGUARD-ABLAUF

CryptokeyRouting



UNIVERSITÄT BONN

WIREGUARD-ABLAUF

CryptokeyRouting

- Vereinfacht die Systemadministration, z.B. von iptables
  - Eingehende Pakete von 10.10.0.2 auf wg0
    - sind definitv von dem entsprechenden Peer
    - wurden verschlüsselt übertragen
  - Klar strukturierte (Interface-orientierte) Regeln in iptables

UNIVERSITÄT BONN

WIREGUARD-TECHNIK

- Zugrundeliegende Crypto:
  - Noise Protocol Framework (noiseprotocol.org)
    - Extra für den Kernelspace implementiertes Protokoll Noise_IKpsk2
  - Moderne Crypto
    - Curve25519 (Elliptische Kurve)
    - BLAKE2s (Hash-Funktion)
    - ChaCha20 (Verschlüsselung)
    - Poly1305 (MAC)
  - Negativ: Keine großen Spielräume bei der Auswahl der Crypto-Verfahren
    - Derzeit vielleicht gar nicht nötig?

UNIVERSITÄT BONN

ENDE

Vielen Dank für die Aufmerksamkeit!

Fragen?

Nächste Vorlesung:

- Montag, 5. Juli 2021 (letzte Vorlesung)

Nächste Übung:

- Dienstag, 29. Juni 2021 – 16 Uhr
- Abgabe des Übungszettels 9 bis morgen – 16 Uhr

- Klausurtermine (vorläufig):
  - 1. Klausurtermin: 5. August 2021 im Zeitraum 10 - 13 Uhr
  - 2. Klausurtermin: 30. September 2021 im Zeitraum 10 - 13 Uhr