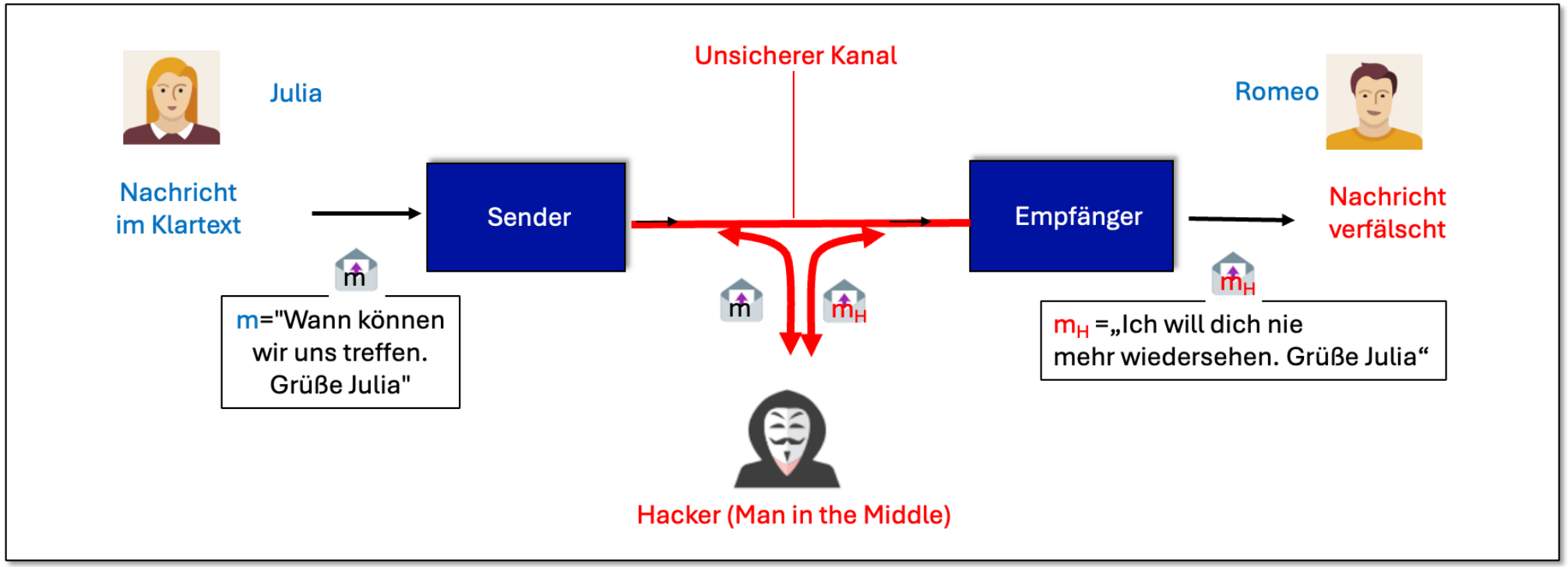
Kapitel 4: Sicherer Nachrichtentransport

Lernziele:

- Sichere versus unsichere Nachrichtenkanäle
- Zusammenspiel IKE, IPsec, RADIUS, EAP verstehen und erklären können.
- Funktionsweise und Nachrichtenformate von IKE, AH und ESP.

Überblick:

4.1 Sichere Nachrichtenkanäle

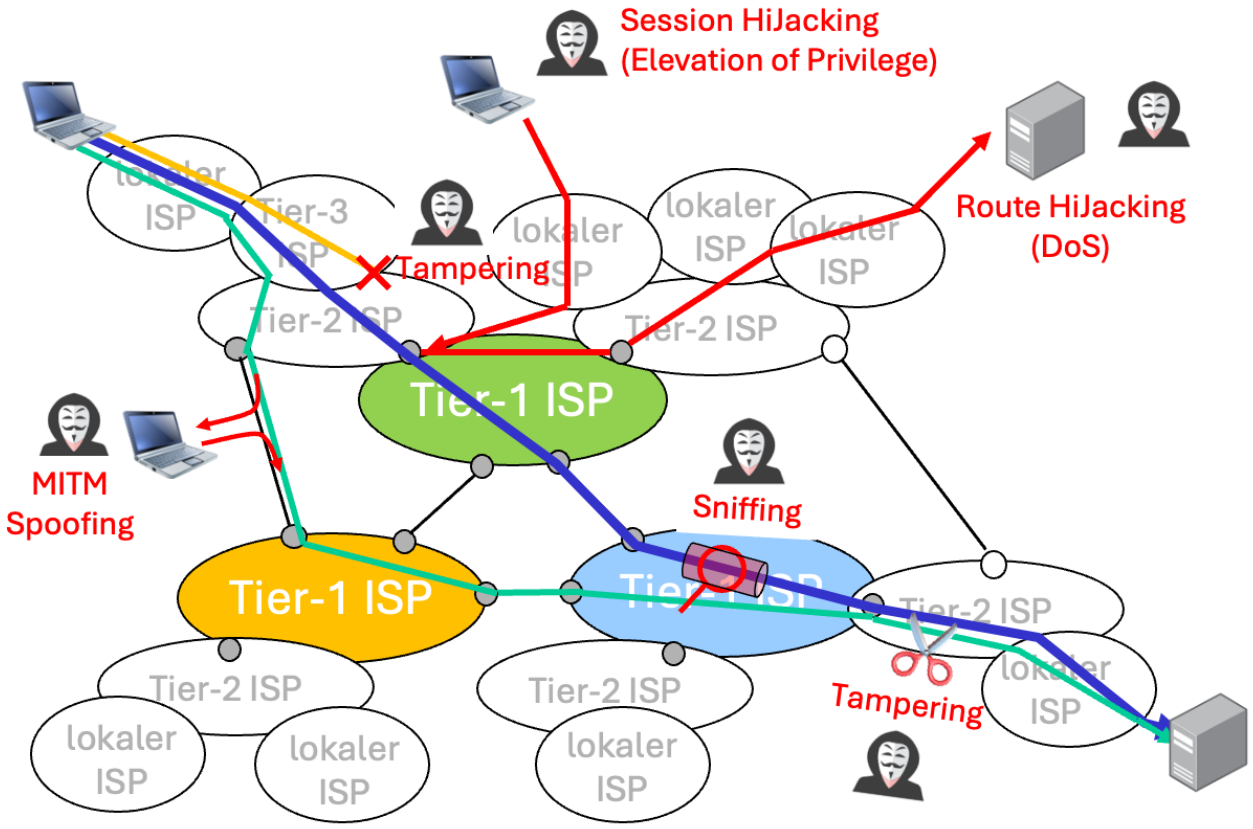


Übertragung von Nachrichten über unsicheren Kanal

Ein Kommunikationskanal wird als unsicher bezeichnet, wenn er die Kernziele (CIAA) der IT-Sicherheit nicht gewährleistet.

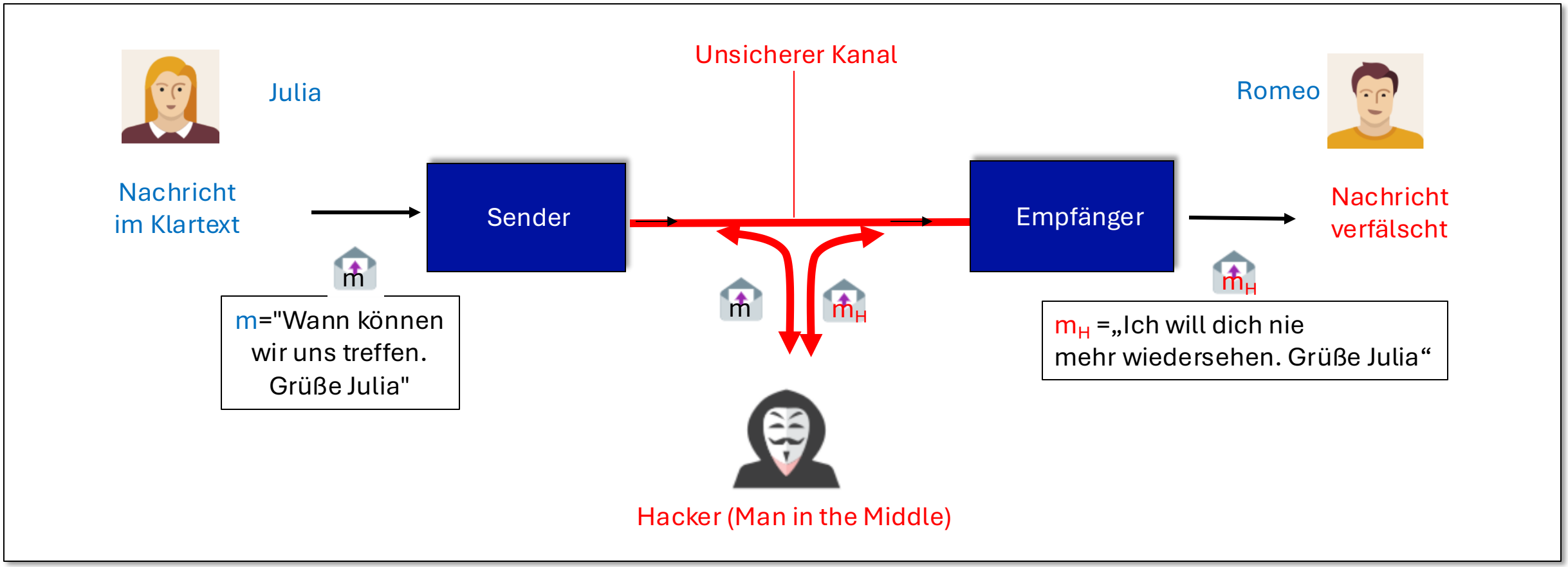
Unsichere Kommunikationskanäle

- Beispiel Internet: Vielzahl von ISPs, durch die ihre Daten weitergeleitet werden und deren Vertrauenswürdigkeit und Identität sie nicht kennen. Daten können beispielsweise blockiert, geändert oder gelesen werden.
- Beispiel Unternehmensnetzwerk: Nachrichten in einem Unternehmensnetzwerk können von nicht autorisierten Personen mitgelesen oder geändert werden.
- Beispiel Mobilfunk: Nachrichten werden auf der Funkstrecke durch den Mobilfunkbetreiber verschlüsselt und anschließend unverschlüsselt ins Internet weitergereicht. Mobilfunkbetreiber und ISPs können die Daten mitlesen oder ändern.



Übertragung von Nachrichten über unsicheren Kanal

- Die Standardprotokolle TCP/UDP/IP/Ethernet/WLAN in einem LAN übertragen die Nachrichten per Default im Klartext und ohne Schutz vor Veränderung.
- Die Daten können nicht auf Echtheit und Unversehrtheit geprüft werden und die Identität des Senders nicht verbindlich festgestellt werden.

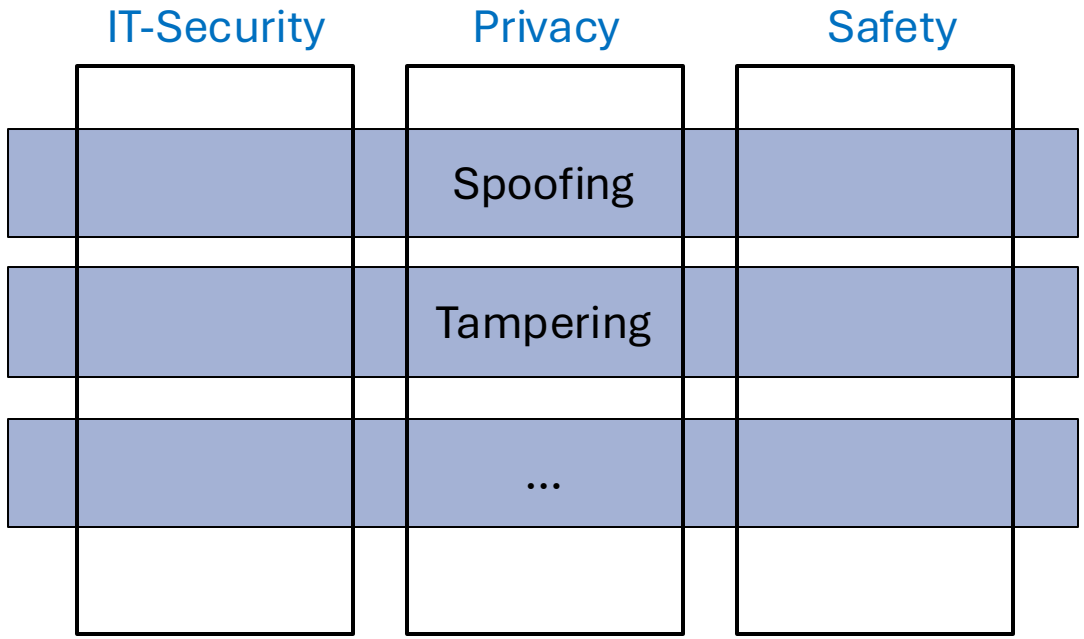


Threats

Ein Threat (Deutsch: Bedrohung) bezeichnet in der IT-Sicherheit eine potenzielle Gefahr, die die Vertraulichkeit, Integrität oder Verfügbarkeit von Daten oder Systemen beeinträchtigen kann.

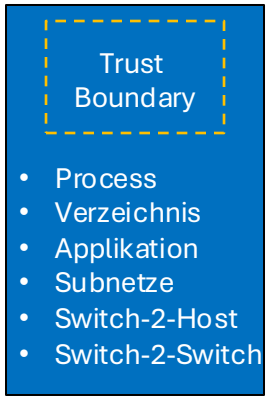
- Ein Threat kann durch beabsichtigtes oder unbeabsichti-gtes menschliches Handeln, oder durch unvorhersehbare Ereignisse wie z.B.: technische Fehler (z.B.: Feuer, defekte Festplatte, Stromausfall) oder Naturkatastrophen (z.B.: Überschwemmung, Sturm) entstehen.
- Die Planung eines sicheren Netzwerkdesigns oder die Entwicklung eines sicheren Produktes beginnt mit dem Verständnis der Bedrohungen.
- Auf Basis der Bedrohungen wählt man ein optimales Netz-werkdesign (Segmentierung, Isolation, ...) und ergänzt dieses um Sicherheitsmaßnahmen (IPS, Content-Analysis, WAF, ...).

- Threats können in 3 Hauptbereiche unterteilt werden:
  - Threats die die IT-Security bedrohen.
  - Threats die die Privacy bedrohen.
  - Threats die die Safety bedrohen.
- Threat-Typen (z.B.: Spoofing) können dann auf die Hauptbereiche angewandt werden.



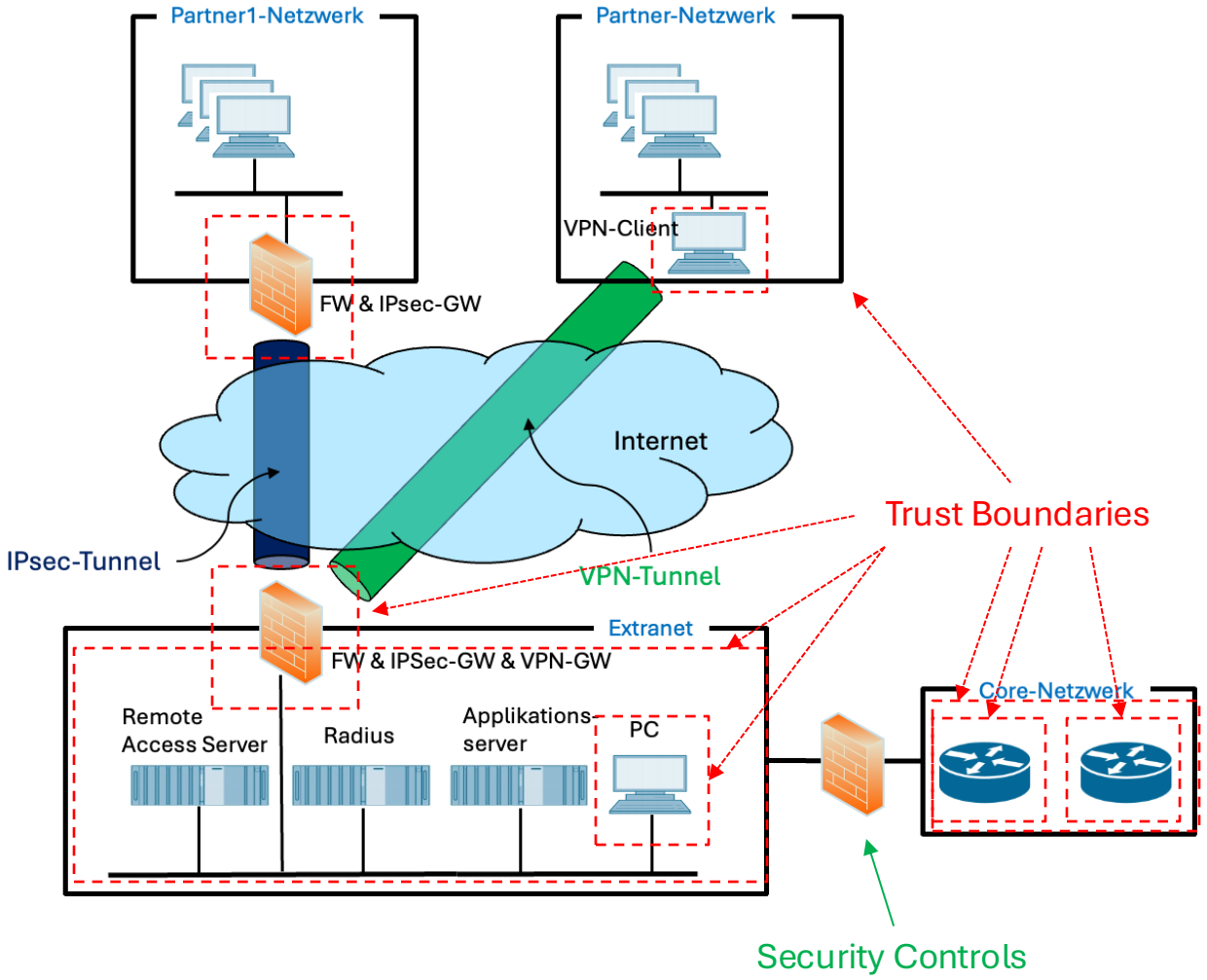
Trust Boundaries

Trust Boundary beschreibt eine Grenze, an der die Bearbeitung, Speicherung oder Übertragung von Daten ihre "Vertrauensstufe" ändert.



- Per Definition wird
  - allen Systemen
  - allen Prozessen
  - allen Daten

  innerhalb einer Trust Boundary vertraut.
- Systeme in unterschiedlichen Subnetzen, haben immer eine Vertrauensgrenze zwischen sich. Beispiele sind Internet vs. Intranet, Client-Netzwerk vs. Server-Netzwerk
- APIs HTTP-REST, SQL-API, Inter-Process-Communication, ...), sind klassische Trust-Boundaries zwischen Prozessen. Die prozessinterne Verarbeitung ist vertrauenswürdiger als die prozessexterne Verarbeitung von Daten.



Threat-Analyse mit STRIDE

- Mittels Datenflussdiagrammes können Threats an den Trust Boundaries von Netzwerken und Applikationen identifiziert werden.
- Ansatz: Verwenden von vordefinierten Threat-Kategorien (STRIDE-Modell) und Threat-Katalogen (BSI-IT-Grundschutz-Kompendium).
- STRIDE ist ein einfaches Verfahren zur Identifizierung von Threats, indem es als "Hilfsmittel" die Threats in sechs Kategorien einteilt:
  - Spoofing (AAA)
  - Tampering (CIA)
  - Repudiation (AAA)
  - Information Disclosure (CIA)
  - Denial of Service (CIA)
  - Elevation of Privilege (AAA)

CIA & AAA

CIA-Triade: Confidentiality, Integrity, Availability

AAA-Kerndienste: Authentifizierung, Autorisierung, Auditierung

STRIDE-Model

Threat-Kategorien	Definition	Verletztes Sicherheitsziel
Spoofing	Sich für etwas oder jemand anderen auszugeben.	Authentifizierung (Authentification)
Tampering	Unautorisiert Daten, Software oder Hardware verändern.	Integrität (Integrity)
Repudiation	Behaupten, eine Aktion nicht ausgeführt zu haben, durch Fälschen vom Log-Information.	Auditierung (Non-Repudiation)
Information Disclosure	Informationen für Personen zugänglich machen, die nicht berechtigt sind, sie zu sehen.	Vertraulichkeit (Confidentiality)
Denial of Service	Dienst für einen Benutzer verweigern oder unbrauchbar machen.	Verfügbarkeit (Availability)
Elevation of Privilege	Berechtigungen (Fähigkeiten) ohne entsprechende Autorisierung erlangen	Autorisierung (Authorization)

Threats in Kommmunikationskanälen mit STRIDE

- Für **unsichere Kommunikationskanäle** lassen sich beispielsweise die folgenden Bedrohungen identifizieren:
- **Spoofing (Identitätsvortäuschung)**
  - IP-Spoofing: Ein Angreifer gibt sich als legitimer Kommunikationspartner aus und erzeugt einen Man-in-the-Middle-Angriff (MITM).
  - DNS-Spoofing: Der Angreifer leitet den Datenverkehr mittels gefälschter DNS-Namensauflösung an eine Adresse des Angreifers um.
- **Tampering (Manipulation von Daten)**
  - Packet Injection: Ein Angreifer fügt manipulierte Pakete in eine unverschlüsselte Kommunikation ein.
  - Message Tampering: Ein Angreifer verändert Pakete, um falsche oder irreführende Daten zu übermitteln.
- **Repudiation (Abstreitbarkeit)**
  - Ein Angreifer oder legitimer Benutzer kann eine gesendete oder empfangene Nachricht abstreiten, wenn keine digitale Signatur und Protokollierung existiert.
  - Ein Angreifer manipuliert die Log-Einträge eines Servers, um seine Identität zu verbergen.
- **Information Disclosure (Informationsoffenlegung)**
  - Sniffing: Das Abhören von Klartext-Datenpaketen auf einem ungesicherten Netzwerk (z. B. WLAN ohne Verschlüsselung).
  - Side-Channel-Angriffe: Analyse von Metadaten (CPU-Last, FileIO, Stromverbrauch, ...) oder Messung der Zeitdauer zur Berechnung bestimmter Operationen, um Rückschlüsse auf die Kommunikation zu ziehen.

Threats in Kommmunikationskanälen mit STRIDE

- **Denial of Service (Dienstverweigerung)**
  - Flooding-Angriffe: Eine Überlastung des Kommunikationskanals durch massenhaften Datenverkehr (ICMP-Flooding)
  - TCP-SYN-Flood: Ein Angreifer hält Ressourcen auf einem Server durch unvollständige Verbindungen blockiert.
  - Black-Holing: Beim BGP Route Hijacking schleust ein Angreifer falsche Routing-Informationen in das Border Gateway Protocol (BGP) ein, wodurch der Internetverkehr fehlgeleitet wird.

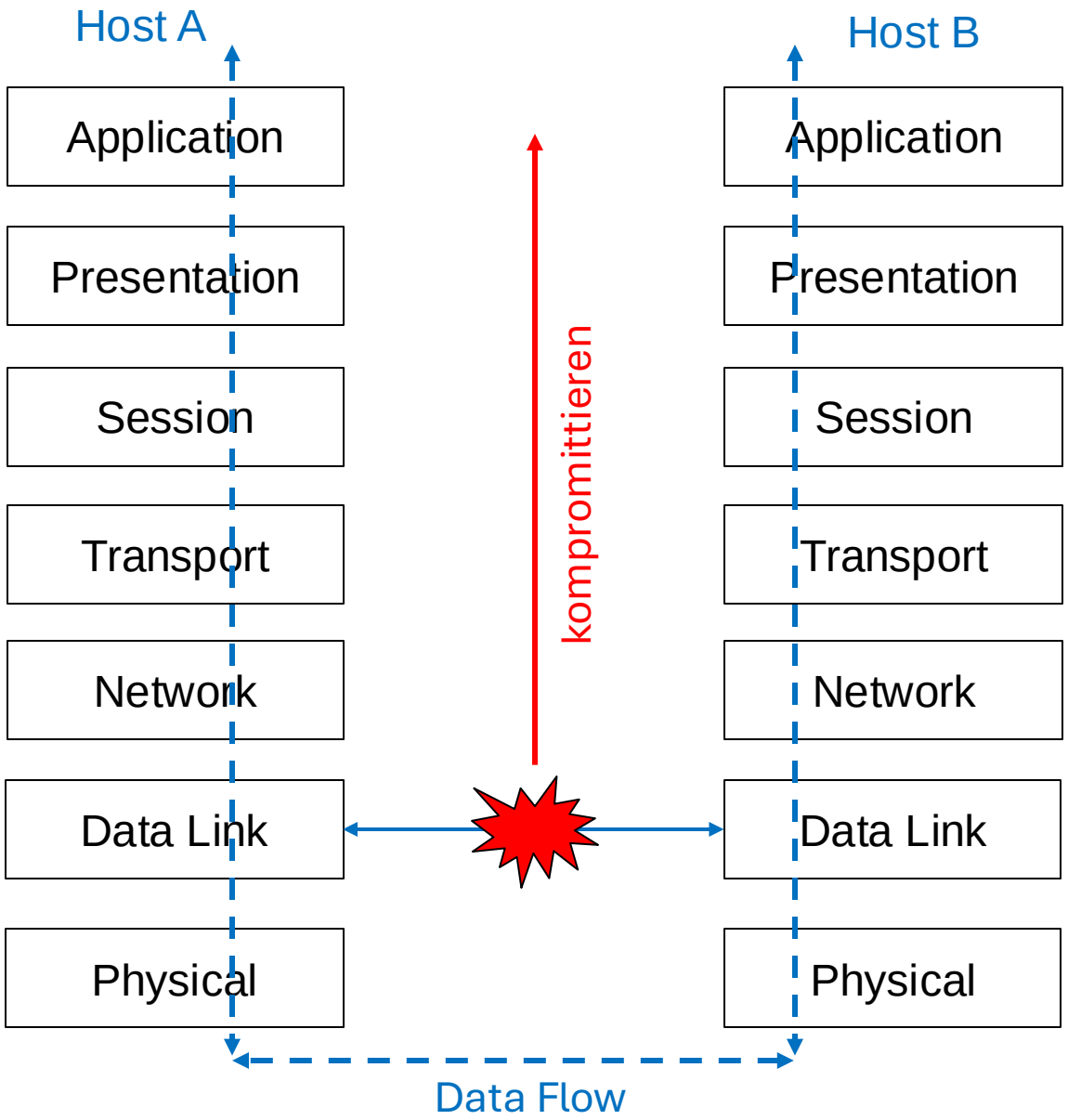
- **Elevation of Privilege (Rechteausweitung)**
  - Weak Authentication: Ein Angreifer nutzt schwache Authentifizierungsmechanismen (z.B.: einfaches Passwort) aus, um höhere Rechte zu erlangen.
  - Missing Autorization: Ein Angreifer kann auf ein kritisches Netzwerk (Produktionsnetz, Managementnetz) zugreifen, da der Zugriff ohne Autorisierung möglich ist.
  - Session Hijacking: Eine bestehende Sitzung wird übernommen (lesen und verwenden des Sitzungs-Cookies), wenn die Kommunikation nicht ausreichend gesichert ist.

Security in Layers

- Die Netzwerkprotokolle des Internets verwenden eine Schichtenarchitektur (siehe Vorlesung Netztechnik).
- Die Daten fließen durch die einzelnen Schichten bei der Übertragung von Host A zu Host B und umgekehrt.
- Die Kompromittierung einer tieferen Schicht zum Beispiel der Data Link Schicht (Layer 2) kann dazu führen, dass die Nutzdaten und Header der oberen Schichten gelesen und geändert werden können.
- Dies ermöglicht dann eine Vielzahl von Cyberangriffen.

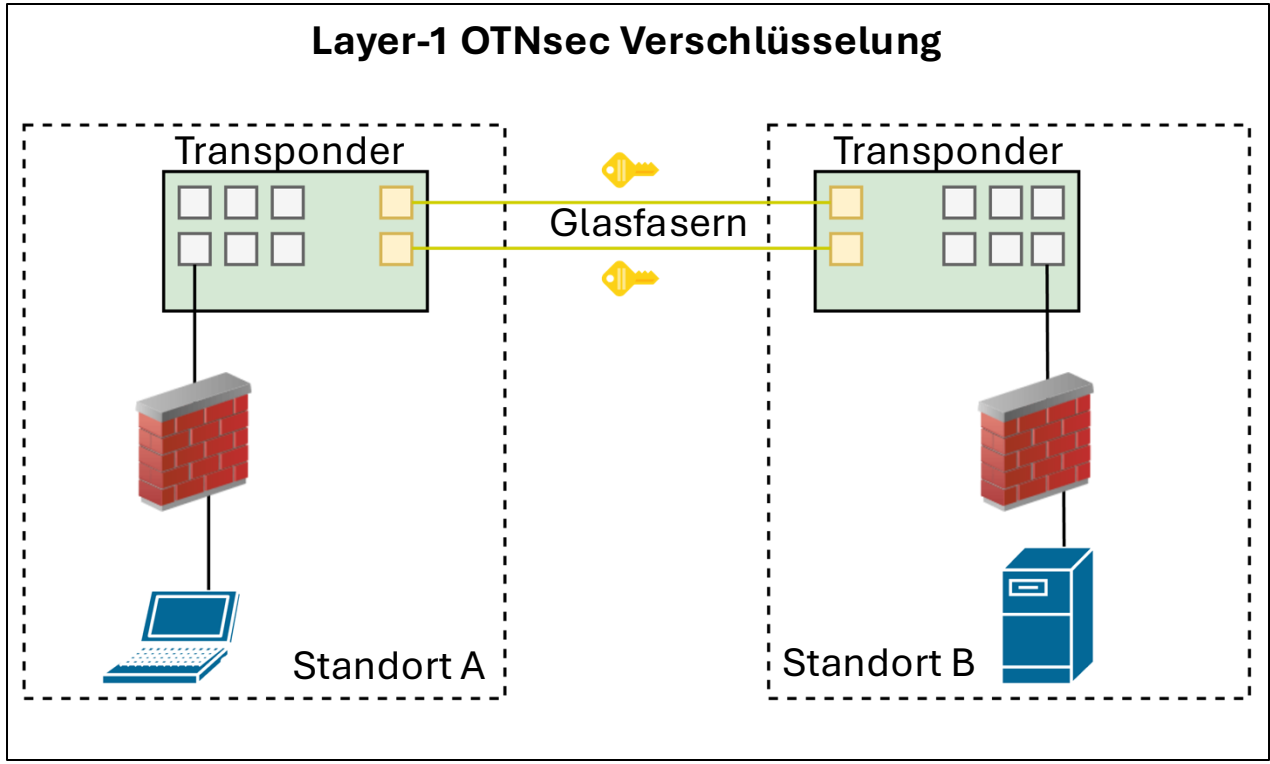
  Sicherheits-Maßnahmen:

  - Sichere Konfiguration der Netzwerkgeräte, die dann bestimmte Angriffsarten unterbindet (DHCP Snooping, Dynamic ARP Inspection, Port-Security, IP Source Guard, ...)
  - Verwendung von sicheren Kommunikationskanälen.



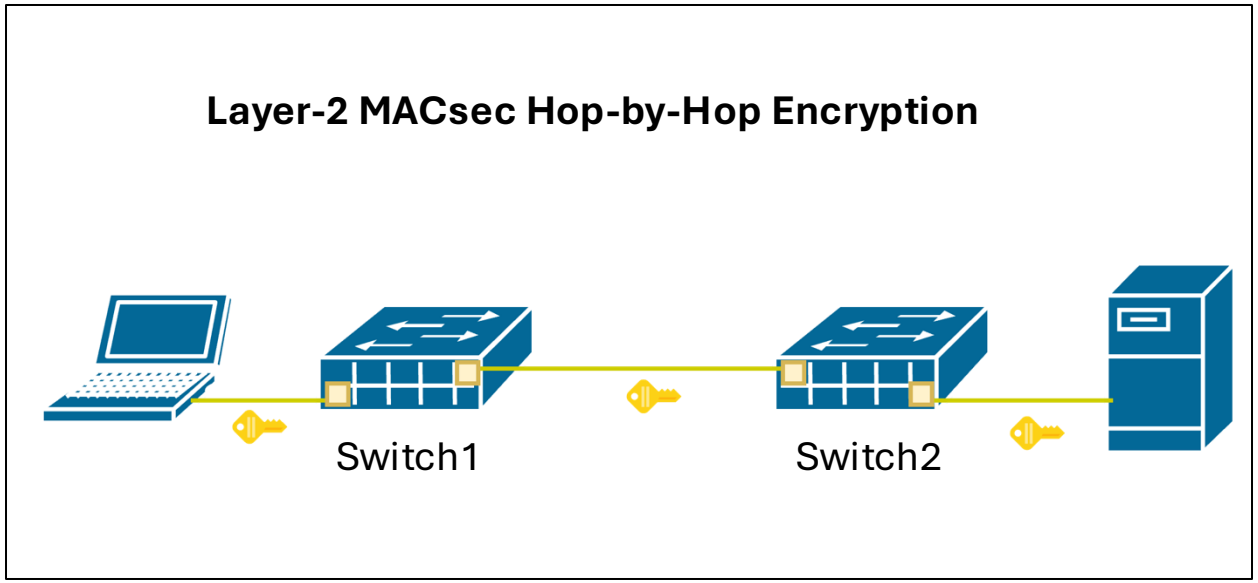
Sichere Kommunikationskanäle – Layer 1

- Um einen sicheren Kommunikationskanal zu erhalten, müssen kryptografische Protokolle zur Verschlüsselung und Integritätssicherung der übermittelten Daten eingesetzt werden.
- Je tiefer im Protokollstapel die Protokolle implementiert werden, umso mehr Schichten profitieren davon.
- Layer1-Verschlüsselung erfolgt mit Leitungsgeschwindigkeit in speziellen Hardwaregeräten. Das phys. Signal wird schon verschlüsselt über die Leitung gesendet.
  - Es entsteht kein Protokoll-Overhead.
  - Abbildung zeigt die Verbindung zweier Standorte über optische Transponder, die direkt mit einer Glasfaser verbunden sind und den Datenverkehr auf der Glasfaser im Layer-1 verschlüsseln.
  - Bezeichnung: OTNSec (Optical Transport Network Security)



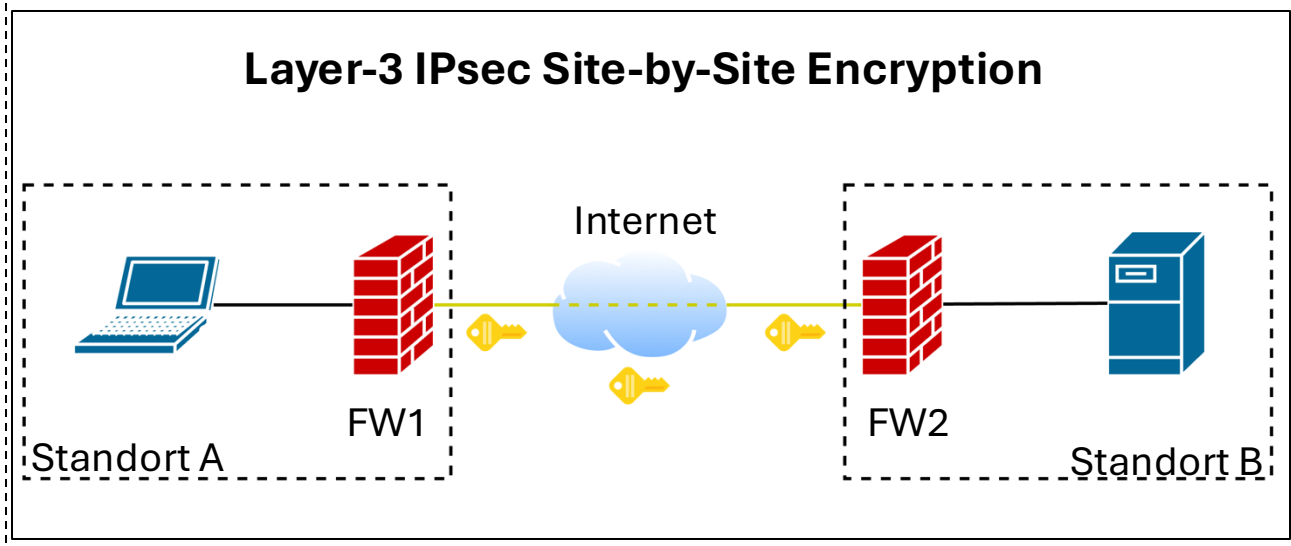
Sichere Kommunikationskanäle – Layer 2

- Layer2-Verschlüsselung ist in der Hardware von Netzwerk-adaptern (Netzwerk Ports in Switchen) implementiert.
- Das verwendete Protokoll heißt MACsec.
- MACsec verwendet einen 16B Header und einen 16B Integrity Check Value (ICV), sodass der gesamte Protokolloverhead der Verschlüsselung 32B beträgt.
- Der Datendurchsatz nimmt für kleine Pakete stark ab
  - 64B → ≅ 50%
- Beispiel: Layer-2 MACsec Hop-by-Hop
  - Der Client verschlüsselt die Daten mittels MACsec, bevor er sie an Switch 1 sendet.
  - Switch 1 entschlüsselt die Daten, forwarded das Paket an den richtigen Ausgangsport und verschlüsselt dort die Daten erneut, bevor er Sie an Switch 2 sendet.
  - Switch2 wiederholt den Vorgang und sendet die Daten verschlüsselt an den Server.



Sichere Kommunikationskanäle – Layer 3

- Layer3-Verschlüsselung ist eine Punkt-2-Punkt-Verbindung oder Site-2-Site-Verbindung, die zwei Standorte über das Internet hinweg, verschlüsselt verbindet.
  - Jeder Endpunkt (Firewall, Router) verschlüsselt den Datenverkehr mittels IPSec, bevor er diesen über das Internet sendet.
  - Der Endpunkt auf der Gegenseite entschlüsselt den Datenverkehr und leitet diesen in das interne LAB weiter.
- Der Protokolloverhead hängt in IPsec vom gewählten Betriebsmodus ab.
  - Im Falle eines Tunnels mit Verschlüsselung und Integritätsicherung beträgt er **68B**.
  - Ohne Tunnel und nur mit Integritätssicherung beträgt er **24B**.



VPN – Virtual Private Networks

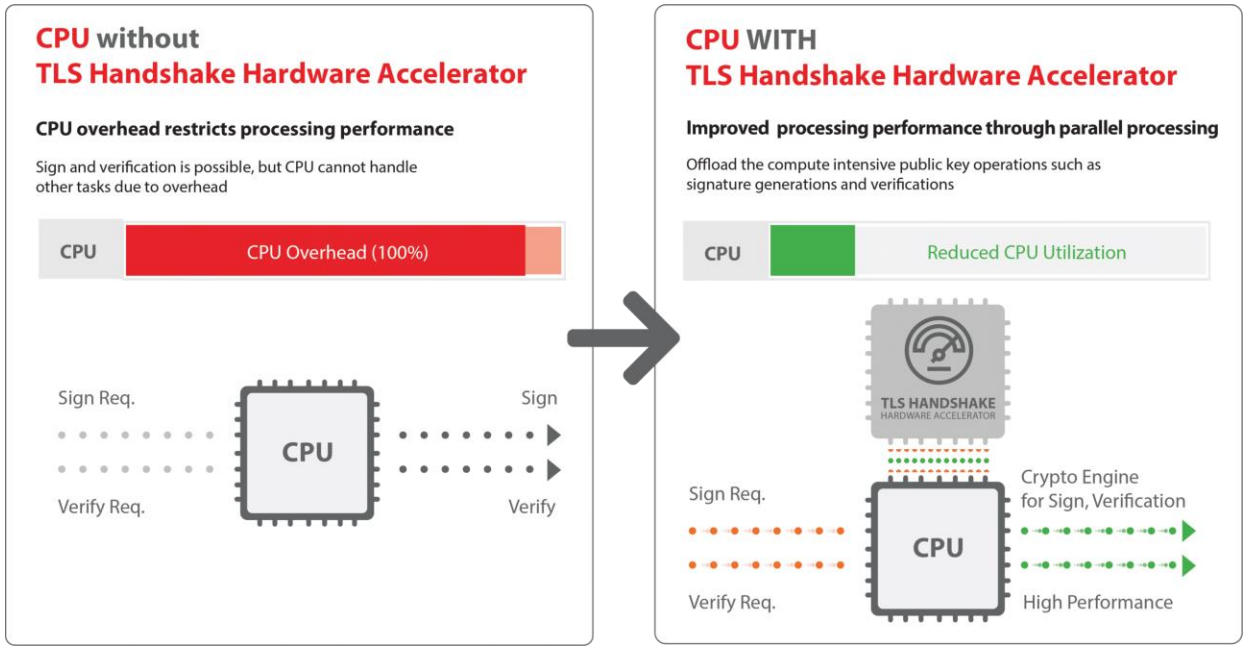
- Um ein Unternehmensnetzwerk über die Standortgrenzen hinweg erweitern zu können, verwendet man kryptografische Protokolle, die den Netzwerkverkehr gegen Mitlesen und Veränderung schützt.
- Man spricht dann auch von einem Virtual Private Network (VPN).
- Auch die sichere Anbindung von mobilen Mitarbeitern oder die sichere Anbindung von Home-Office-Mitarbeitern lässt sich mittels VPNs erreichen.

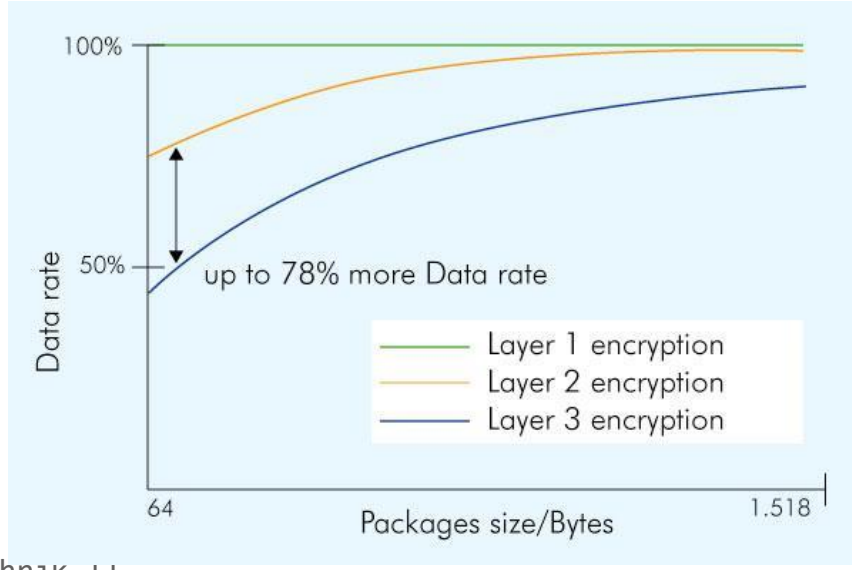
Unter einem VPN (Virtual Private Network) versteht man ein privates virtuelles Netzwerk, das sich physikalisch über öffentliche Kommunikationswege erstreckt.

- "Virtual": VPN-Netzwerk ist rein softwarebasierend und wird über spezielle Tunnel-Netzwerkprotokolle auf bestehenden physikalischen öffentlichen Verbindungen aufgebaut
- "Private": Übertragung erfolgt verschlüsselt und integritätsgesichert über die öffentlichen Verbindungen.

Offloading Layer-3 und Layer-4 Verschlüsselung

- Layer3-Verschlüsselung (IPsec) und höher (TLS, S/MIME) ist in der Betriebssystem-Software implementiert und läuft auf der CPU eines Systems und kann hohe CPU-Last erzeugen. Erzeugen ebenfalls einen Protokolloverhead.
- Netzwerkgeräte besitzen spezielle kryprografische ASICs, die die Verschlüsselung zentral für das Gerät durchführen können.
- Für Serversysteme gibt es Erweiterungskarten oder on-Chip-Prozessoren, die ebenfalls Befehlssätze für krypto-grafische Algorithmen in Hardware umgesetzt haben (AES, SHA-256, ...).
- Netzwerkadapter, die kryptografische Operationen direkt auf der Karte ausführen, um die CPU zu entlasten.
  z.B..: Intel QuickAssist Technologie





Sichere Datenübertragung

- Idee: Alles auf Layer-1 verschlüsseln?
- Problem: Nur Punkt-zu-Punkt Verbindung möglich. Die einzelnen Protokollheader der unterschiedlichen Schichten müssen von Netzwerkgeräten oder Applikationen in Teilen lesbar sein:
  - Switch: MAC-Header
  - Router: IP-Header
  - Firewall: TCP_IP-Header
  - Gateway: Nachrichtenheader
- Lösung: Je nach Anwendungsfall Auswahl eines geeigneten Protokolls.
- Für die sichere Administration von Servern und Netzwerk-geräten verwendet man die Protokolle oberhalb der Transport-Schicht, da so jedes Device und jede Applikation per IP und TCP im Unternehmen/Internet erreichbar ist.

Mögliche Verfahren:

(1) S/MIME : User – 2 - Many User (Layer-7)
Datenverschlüsselung und Signierung

(2) TLS, SSH: App-2-App (Layer-4)
Verschlüsselung und Signierung des Applikations-Payloads.

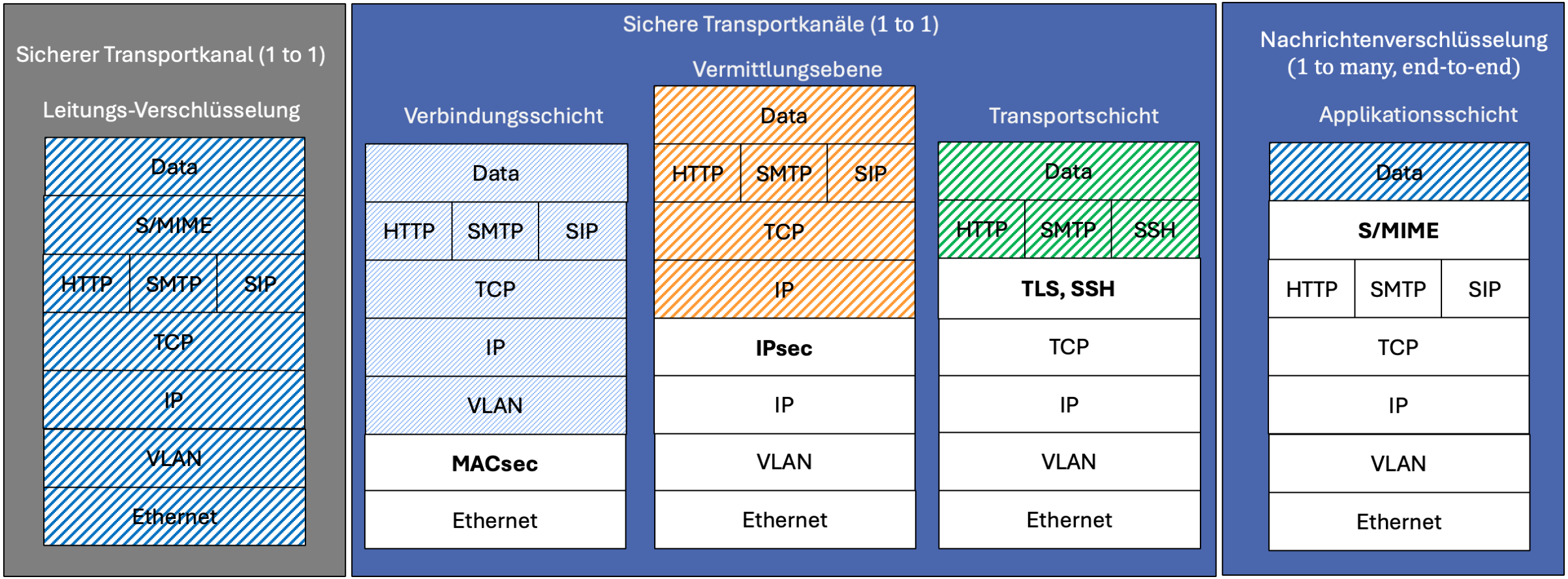
(3) IPsec: Site-2-Site (Layer–3)
Verschlüsselung und Signierung von IP-Paketen oder IP-Payload

(4) MACSec: Port-2-Port (Layer-2)
Verschlüsselung und Signierung pro Port und Endgerät

(5) Phys. Schicht: Leitung (Layer-1)
Verschlüsselung und Signierung der Bits auf einer Leitung (Anbindung ISP)

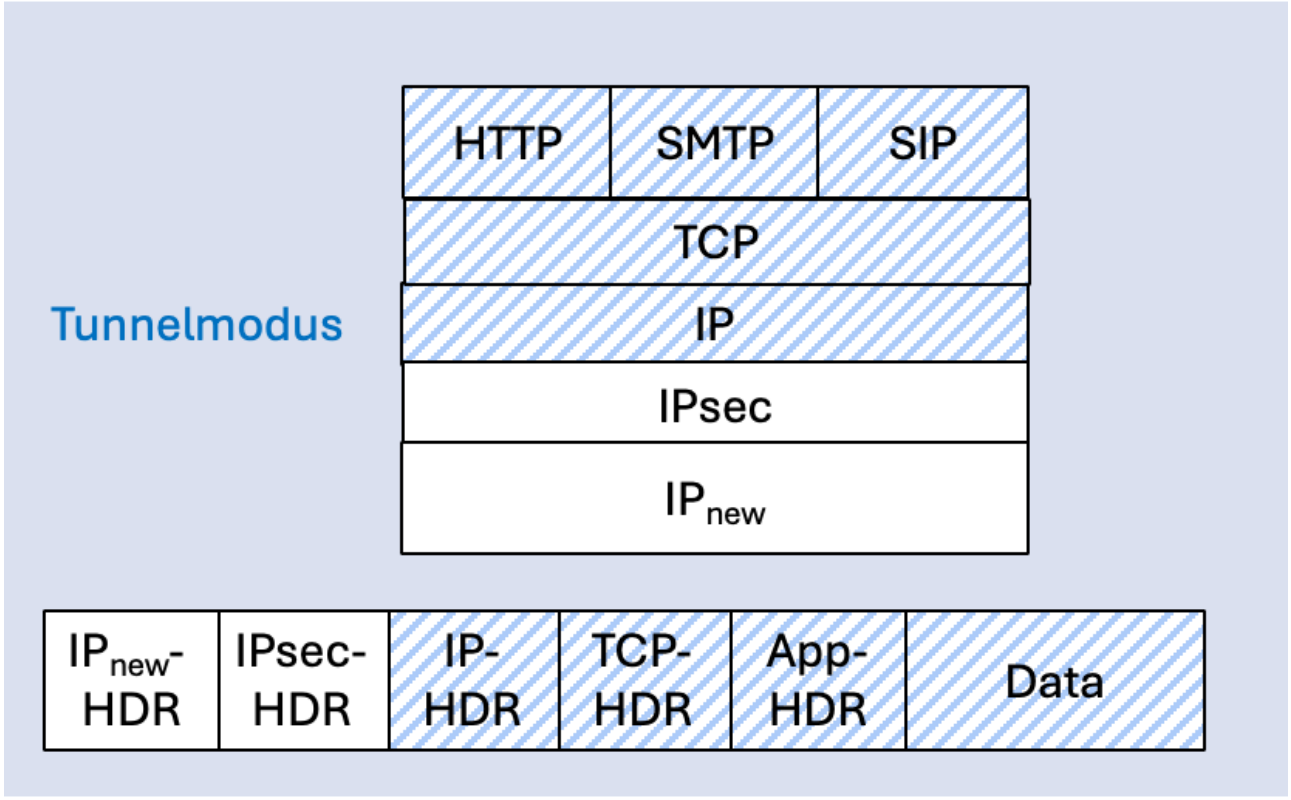
Sichere Transportkanäle

- Die farbigen Felder sind die Protokollschichten die verschlüsselt und integritätsgesichert sind.
- Die weissen Felder stellen den sichtbaren und durch Dritte lesbaren und veränderbaren Bereich dar.



4.2 Sicherer Vermittlungskanal mit IPsec

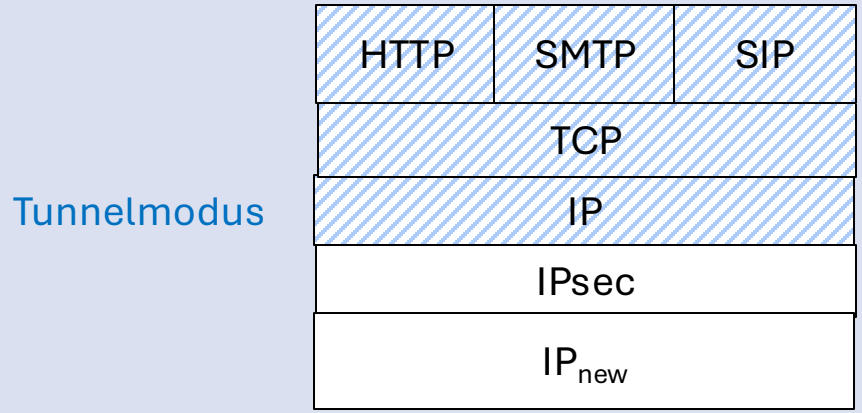
IPsec, EAP, RADIUS

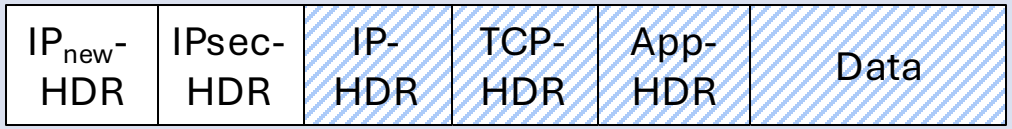


IPsec (RFC 4301)

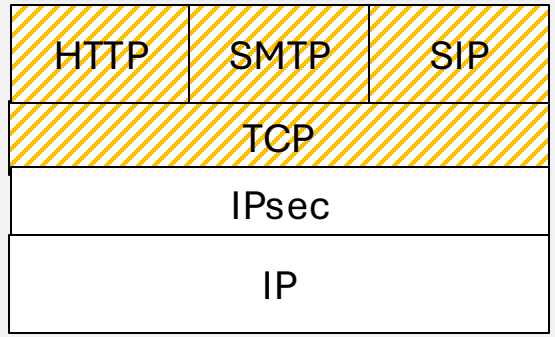
- IPSec (Internet Protocol Security) hat das Ziel den Datenverkehr der Vermittlungsschicht sicher zu gestalten.
- IPsec stellt die folgenden Dienste zur Verfügung
  - Authentifizierung (Pre-Shared-Key, dig. Zertifkate)
  - Integrität der IP-Pakete (HMAC-SHA)
  - Verschlüsselung der Nutzdaten (AES-GCM).
  - Anti-Replay eines IP-Paketes (Sequence-#, Sliding Window).
  - Perfect Forward Secrecy (Diffie-Hellman, ECDSA)
- IPsec kann in 2 Modi betrieben werden:
  - Tunnelmodus: Verschlüsselung und Authentifizierung des IP-Headers und der Nutzdaten (IP-Paket).
  - Transportmodus: Verschlüsselung und Authentifizierung der Nutzdaten (TCP-Segmente).

Tunnelmodus



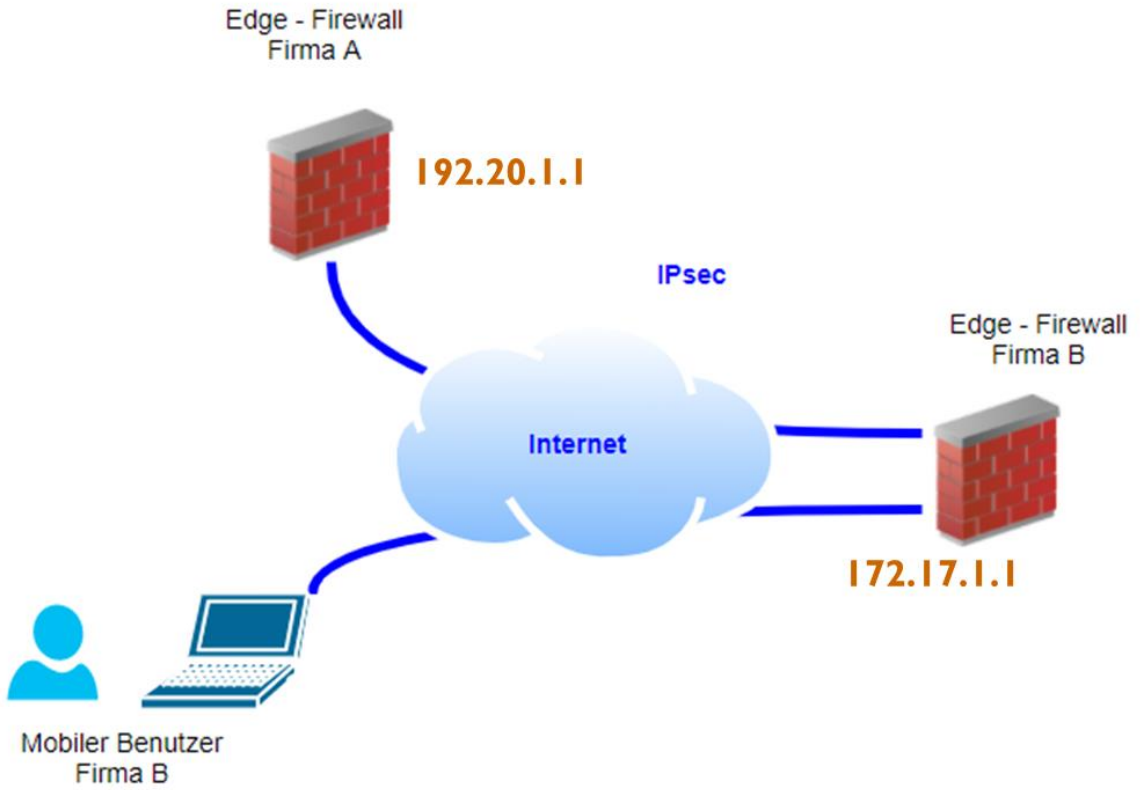


Transport-modus



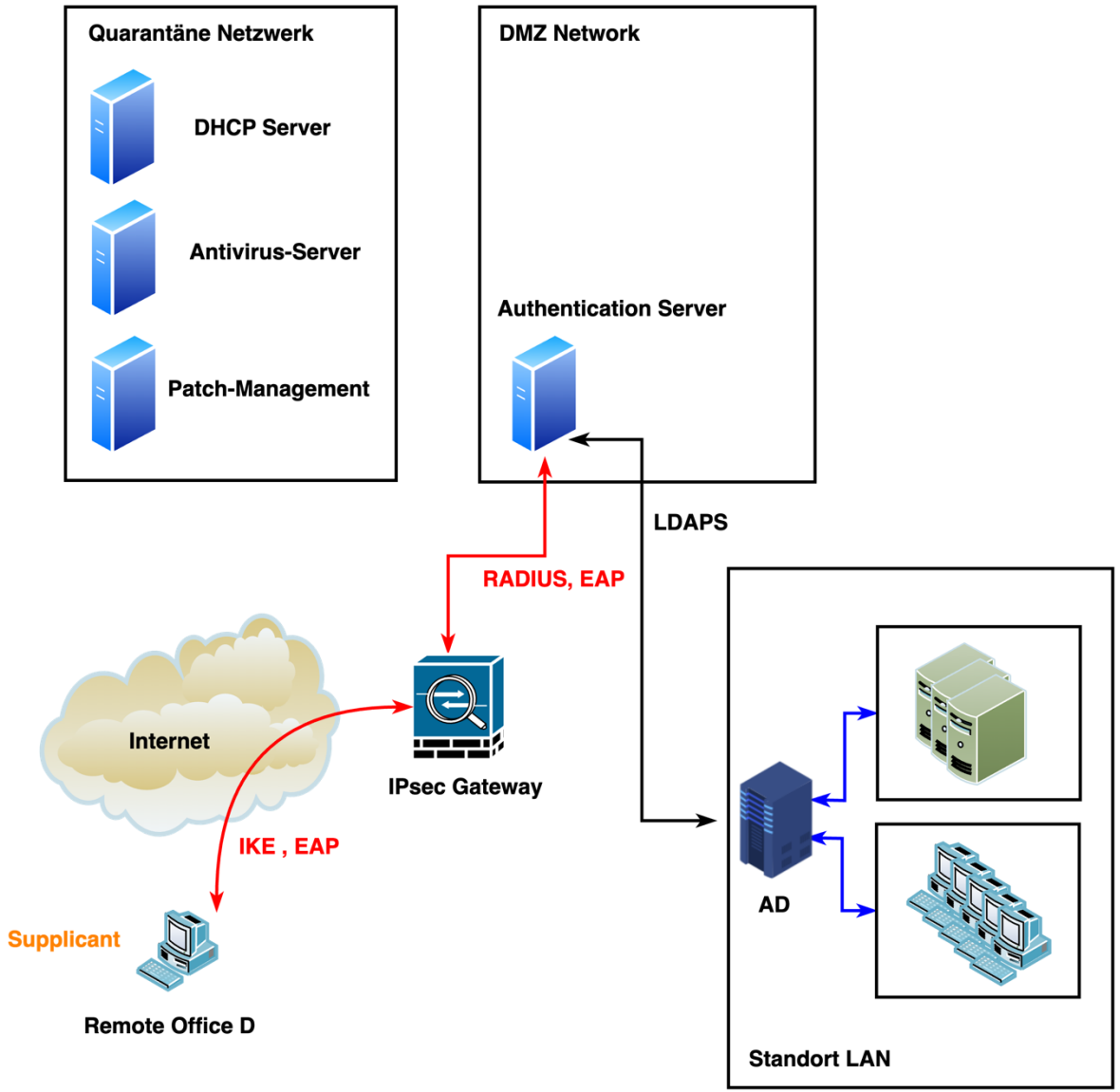
Einsatzszenarien

- IPsec erstellt einen sicheren Kommunikationskanal über das Internet. Man unterscheidet 2 Szenarien
  - Site-2-Site IPsec zwischen den Perimeter-Firewalls oder den Edge-Routern zweier Standorte eines Unternehmens oder zweier Unternehmen (Unternehmen – Zulieferer, Unternehmen – IT-Dienstleister).
  - Client-2-Site IPsec zwischen einem Client mit IPsec-Software (VPN-Client) und der Perimeter-Firewall oder einem Edge–Router.



Beispiel: Authentifizierung via VPN-IPsec-Client

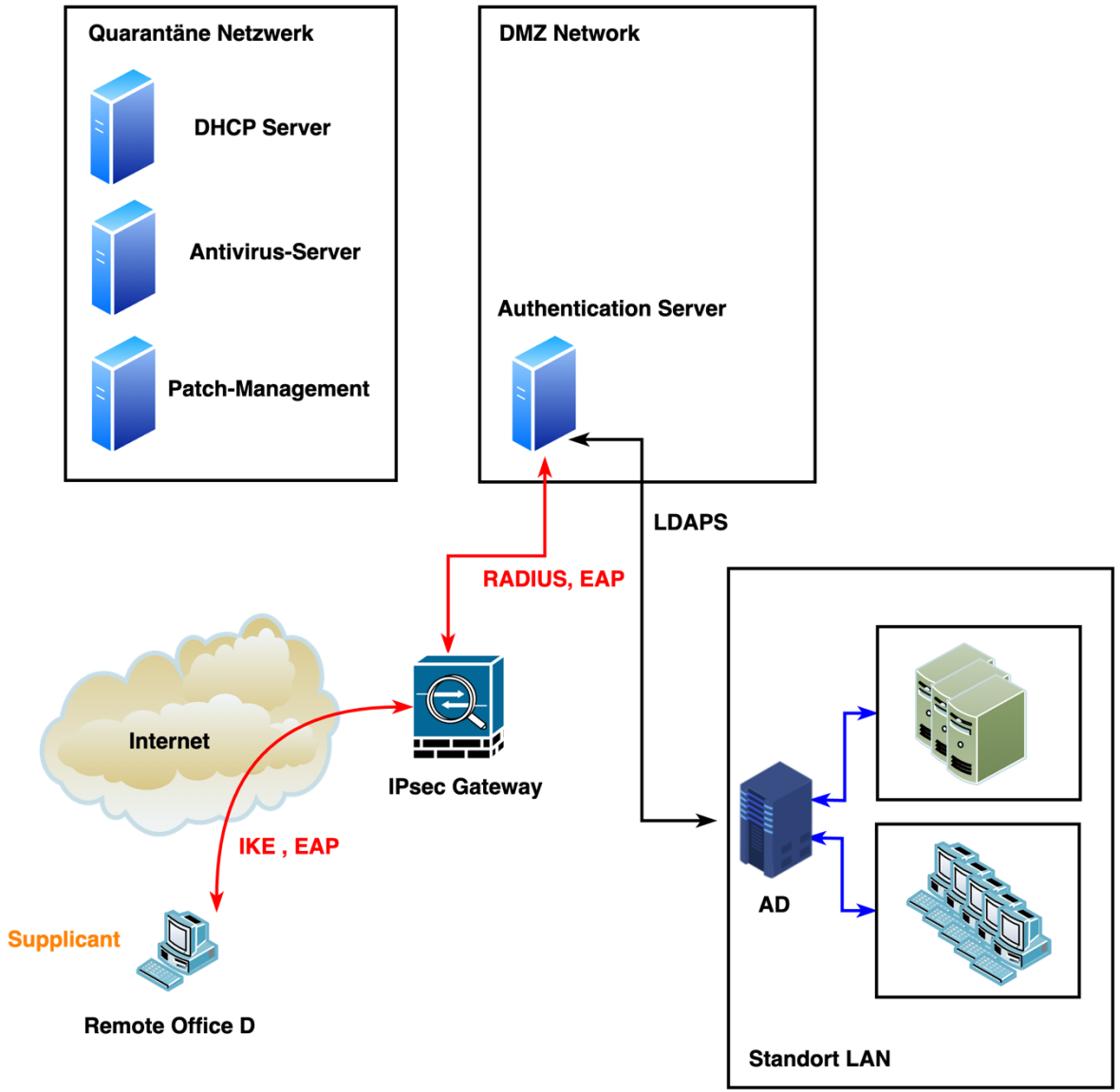
- Verbindet sich ein Mitarbeiter von einer Remote-Lokation (Home-Office, Internet-Kaffee, Zug, ...) mittels eines VPN-IPsec-Clients mit dem IPsec-Gateway des Unternehmens, kann eine sichere Authentifizierung via EAP und RADIUS erfolgen.
- Authentifizierung:
  - Das IPsec-Gateway, fungiert als Authenticator und initiiert den EAP-Prozess indem es den Supplicant auffordert sich zu identifizieren und zu authentifizieren (Name, Passwort oder dig. Zertifikat).
  - Die EAP-Nachrichten werden mittels dem IKE (Internet Key Exchange) Protokoll zwischen Supplicant und Gateway transportiert.
  - Das IPsec-Gateway leitet die EAP-Nachrichten via RADIUS zum Authentifizierungsserver weiter.



Beispiel: Authentifizierung via VPN-IPsec-Client

- Die Authentifizierung erfolgt typischerweise als Zwei-Faktor-Authentifizierung (2FA), d.h. die Zugangsdaten bestehen aus 2 Faktoren beispielsweise einem dig. Zertifikat für den Benutzer/Computer und einem One-Time-Password OTP (Einmalpasswort) für den Benutzer. Das Einmalpasswort kann über eine Authenticator-APP oder ein OTP-Token generiert werden.

- Der Authentifizierungsserver überprüft die Zugangsdaten anhand einer lokalen Benutzerdatenbank oder einem Active Directory Verzeichnisdienst, auf den er mit dem Protokoll LDAPS zugreift.



Beispiel: Authentifizierung via VPN-IPsec-Client

- War die Authentifizierung erfolgreich, sendet der RADIUS-Server mit der RADIUS Access-Accept Nachricht die Autorisierungsinformation für den Supplicant an das VPN-Gateway.
  - Dem Client wird eine private IP-Adresse für die Kommunikation im LAN des Unternehmens zugewiesen.
  - Dem Client wird ein logischer Gruppenname auf dem Gateway zugewiesen werden.
  - Das Gateway kann mittels dem logischen Gruppennamen über ACLs oder über VLANs den Zugriff des Clients auf das Unternehmensnetz autorisieren.

- Das Radius-Attribut Type=26 dient zur Übertragung von herstellerspezifischen Attributen (Vendor-Specific Attribute VSA).
  - Im Beispiel werden dem VPN-Client 2 DNS-Server und ein sogenannter spit-tunnel zugewiesen. Der split-tunnel erlaubt dem Client parallel zur VPN-Verbindung direkt im Internet zu surfen.

RADIUS-Attribute:

Type 64 (Tunnel-Type) : 9 (IPSec)

Type65 (Tunnel-Medium-Type) : 1 (IPv4)

Type 81 (Tunnel-Private-Group-ID) : VPN-Users (Group Name for ASA)

Type 8 (Framed-IP-Address) : 192.168.10.100 (IP-Address for Client)

Type 9 (Framed-IP-Netmask) : 255.255.255.0 (Subnetzmask for Client)

Type 27 (Session-Timeout) : 3600 (Max. Zeitdauer einer Session in s)

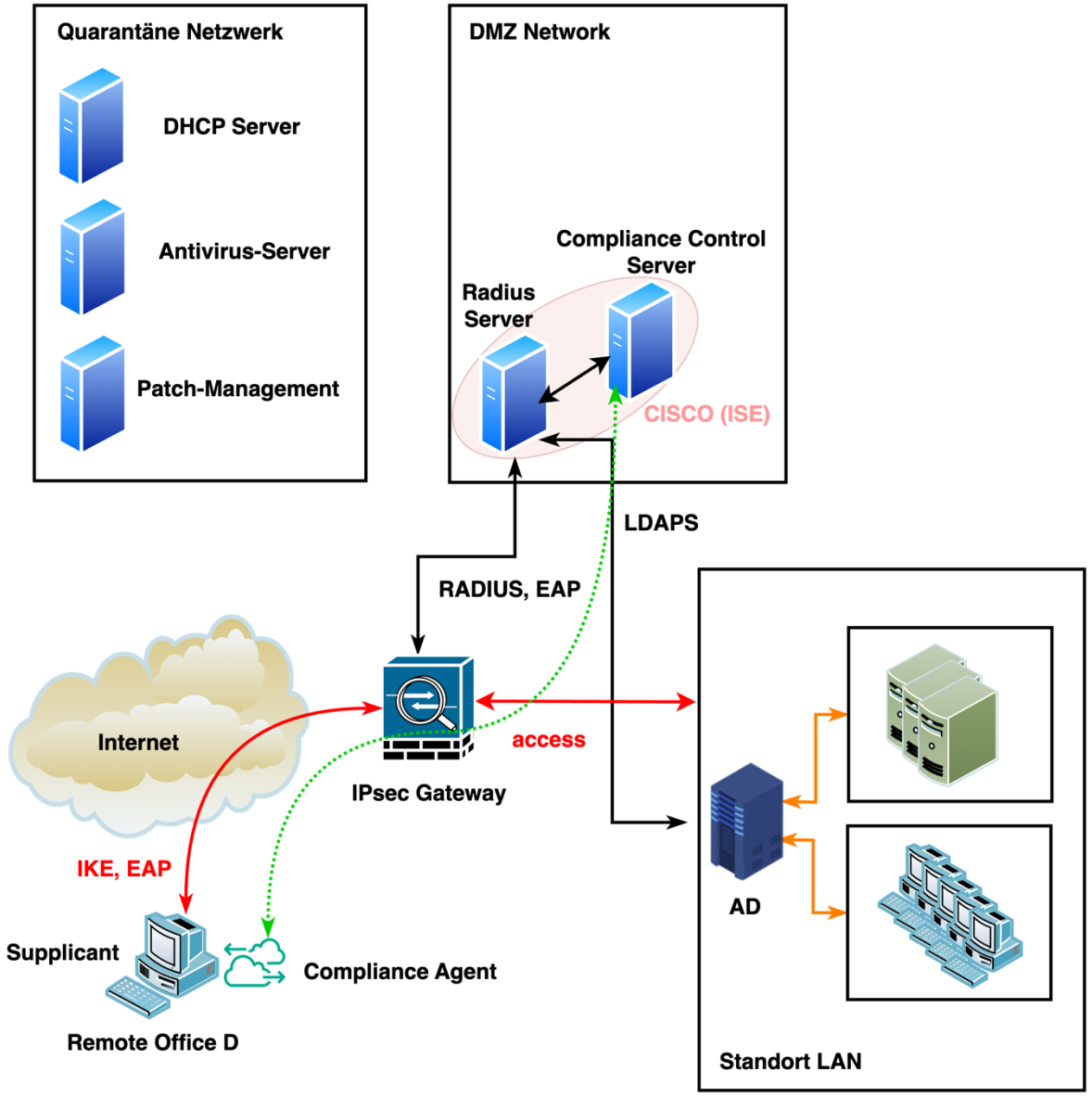
Type28 (Idle-Timeout) : 600 (Max. Inaktivitätszeit einer Session in s)

Type 26 (VSA-Cisco) : "ip:dns-servers=8.8.8.8 1.1.1.1"

type 26 (VSA-Cisco) : "ip:split-tunnel-list=SplitACL"

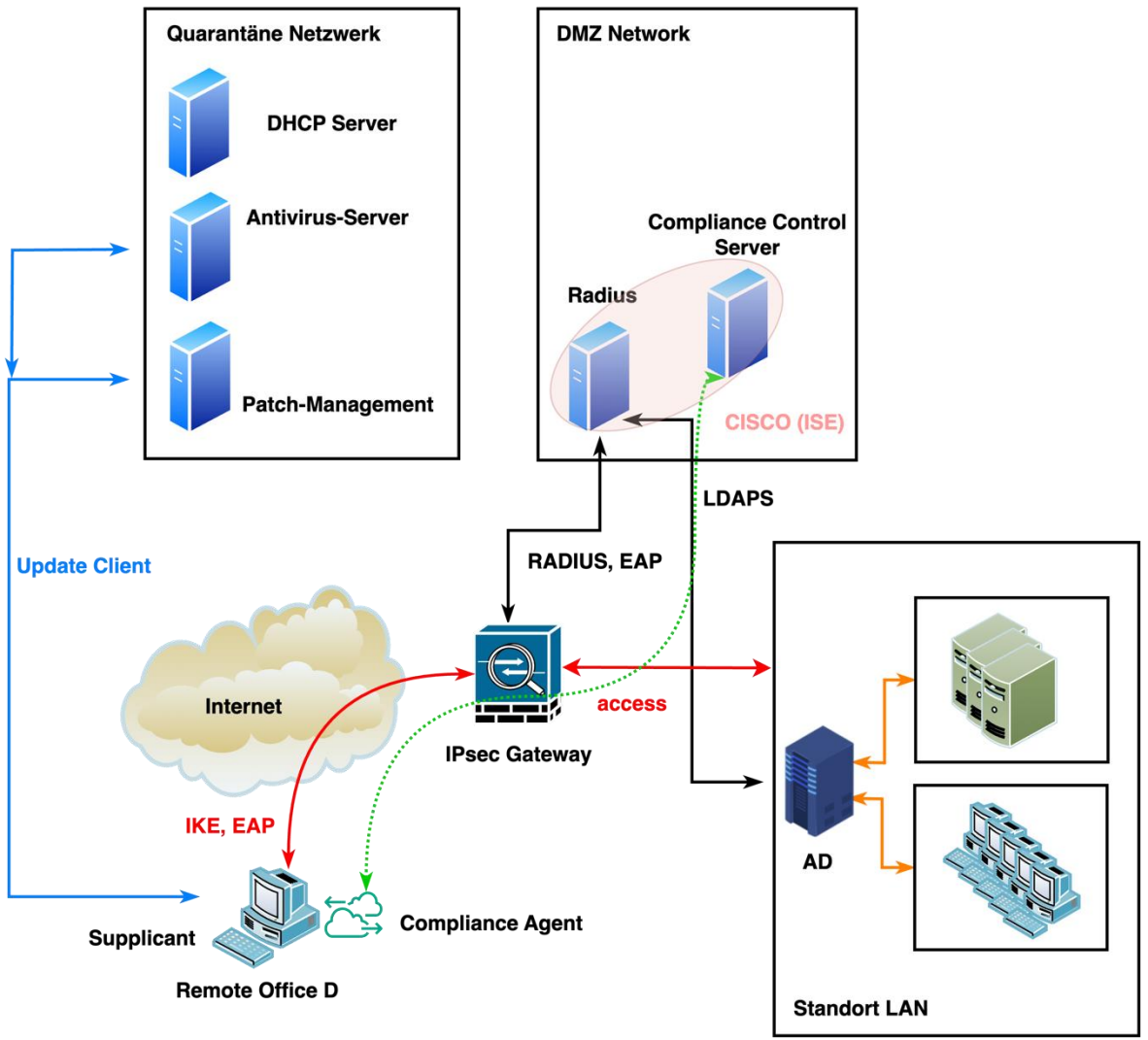
Beispiel: NAC mit Compliance-Überprüfung

- Optionale Compliance-Überprüfung:
- Im Anschluss an die Authentifizierung und vor der Autorisierung kann optional die Compliance (Übereinstimmung mit den Unternehmensrichtlinien) des Supplikants mittels einer zusätzlichen Software (Configuration Mgmt.) geprüft werden.
  - Dazu überträgt der Configuration-Agent auf dem Supplicant über den Authenticator seine aktuelle Konfiguration an einen Configuration-Server (CS).
  - Beispielhaft wird die folgende Information übertragen:
    - OS-Version und Security Patch Status
    - Antivirus-SW-Version und Pattern-Status
    - Status lokale Firewall: aktiviert ja/nein
  - Der Configuration-Server prüft, ob der Client die vordefinierten Sicherheitskriterien erfüllt.



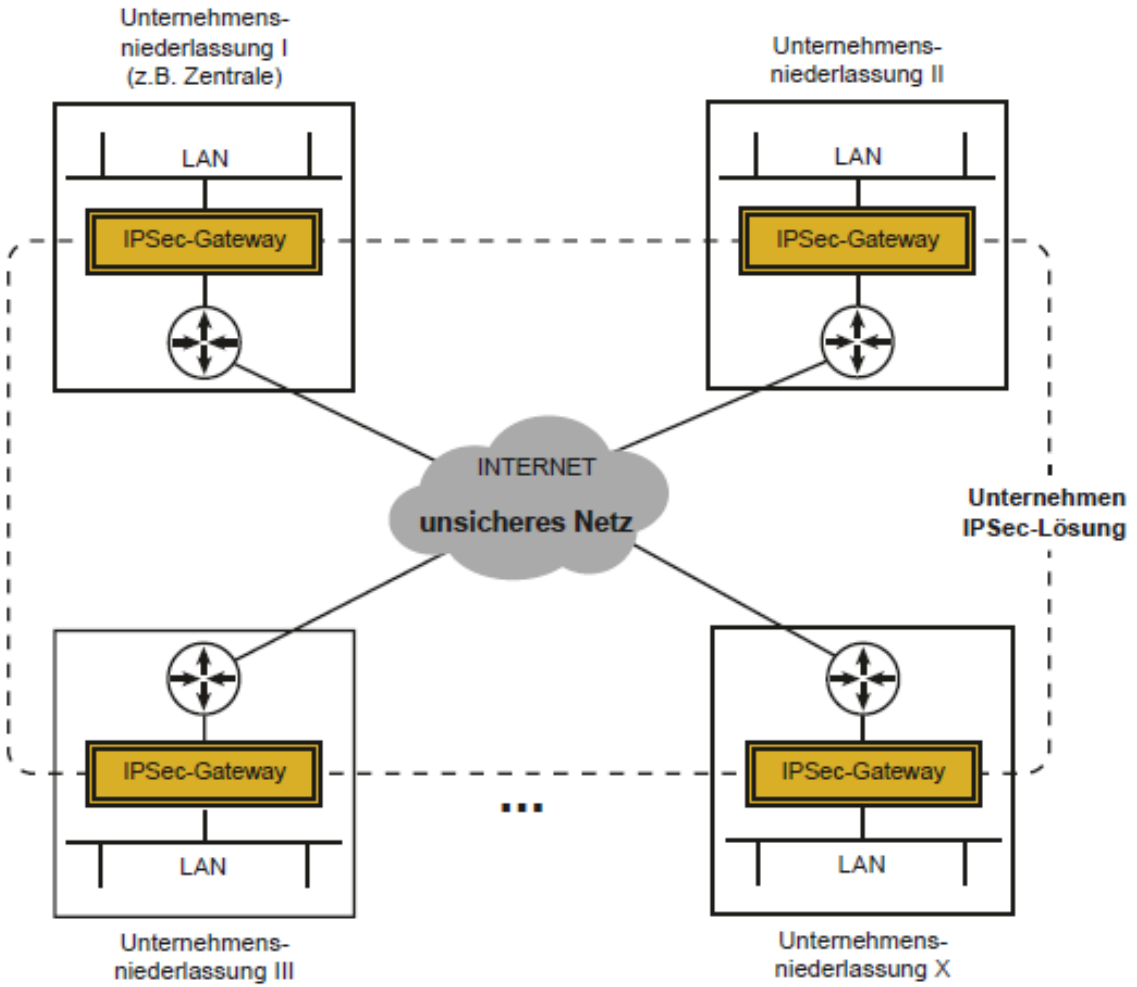
Beispiel: NAC und Compliance-Überprüfung

- Der CS sendet den Konformitätsstatus des Clients an den RADIUS-Server:
  - Bei Konformität und erfolgreicher Authentifizierung sendet der RADIUS-Server die Autorisierungsdaten für den Zugriff auf das Unternehmensnetzwerk.
  - Bei Nichtkonformität sendet der RADIUS-Server an den Authenticator die Autorisierungsdaten für ein Quarantäne Netzwerk.
  - Der Authenticator verschiebt dann den Client in das Quarantäne-Netzwerk und sendet diesem eine Remediation-URL.
  - Der Supplicant kann sich auf die geforderten SW-Stände (Remediation: Sanierung) updaten und muss sich anschließend erneut anmelden.
- Beispiel: Die Cisco Identity Services Engine (ISE) kombiniert einen RADIUS-Server mit einem Configuration Server.

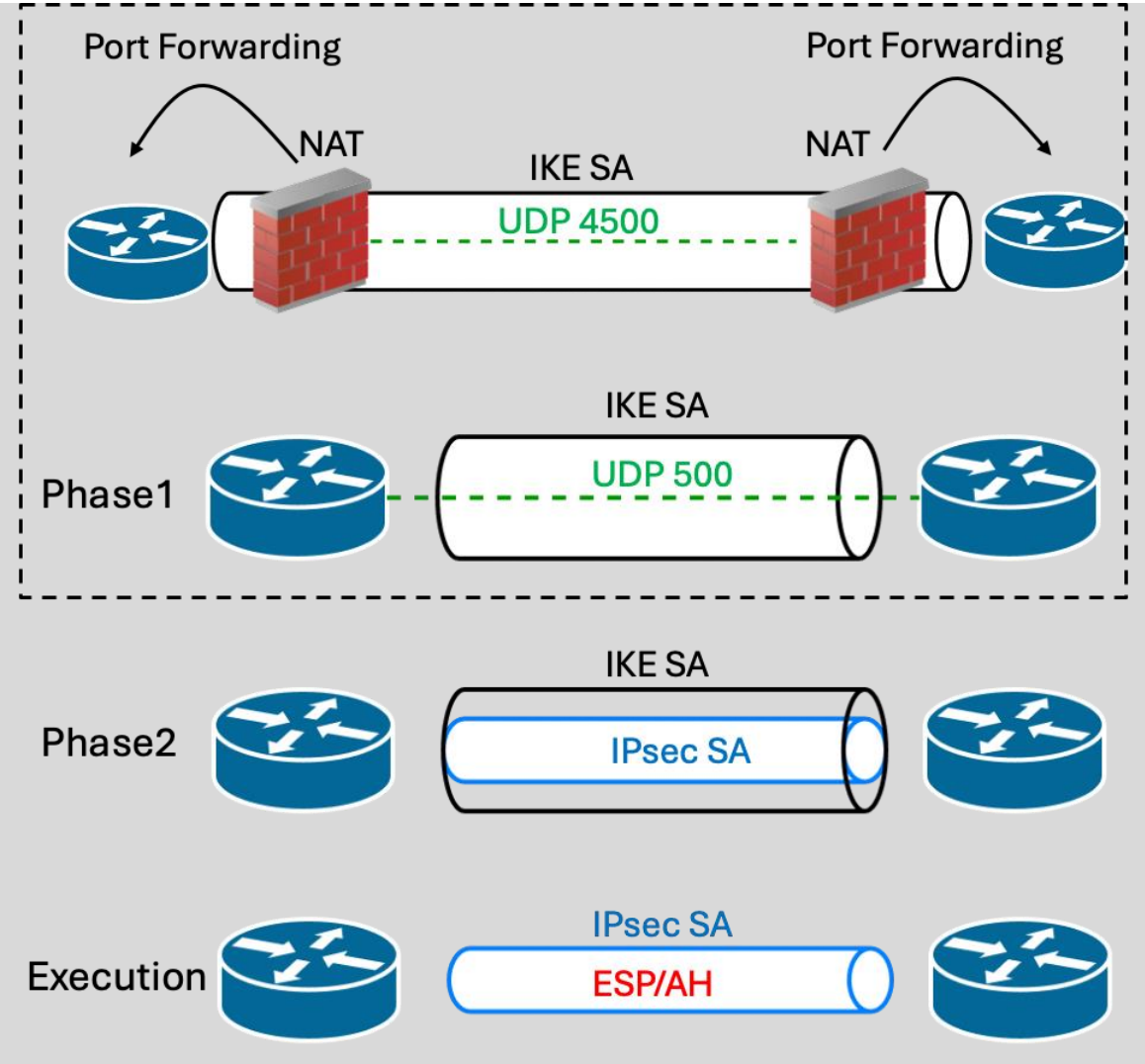


Beispiel: Site-2-Site IPsec

- Die unterschiedlichen LANs eines Unternehmens werden mittels IPsec über öffentliche Kommunikationskanäle transparent vernetzt: Site-2-Site-Kopplung.
- Die Authentifizierung kann mittels einem pre-shared secret erfolgen.
- IPsec arbeitet im Tunnelmodus zwischen den Edge-Firewalls (alternativ den Edge-Router) der einzelnen Niederlassungen.
  - Die äußeren (öffentlichen) IP-Quell- und -Zieladressen identifizieren die Endpunkte des Tunnels (=Firewalls).
  - Die inneren (privaten) IP-Quell- und -Zieladresse identifizieren den Absender und Empfänger des geschützten IP-Datenverkehrs.
- Der öffentliche IP-Header wird vom sendenden IPsec-Gateway hinzugefügt und vom empfangenden Gateway wieder entfernt.

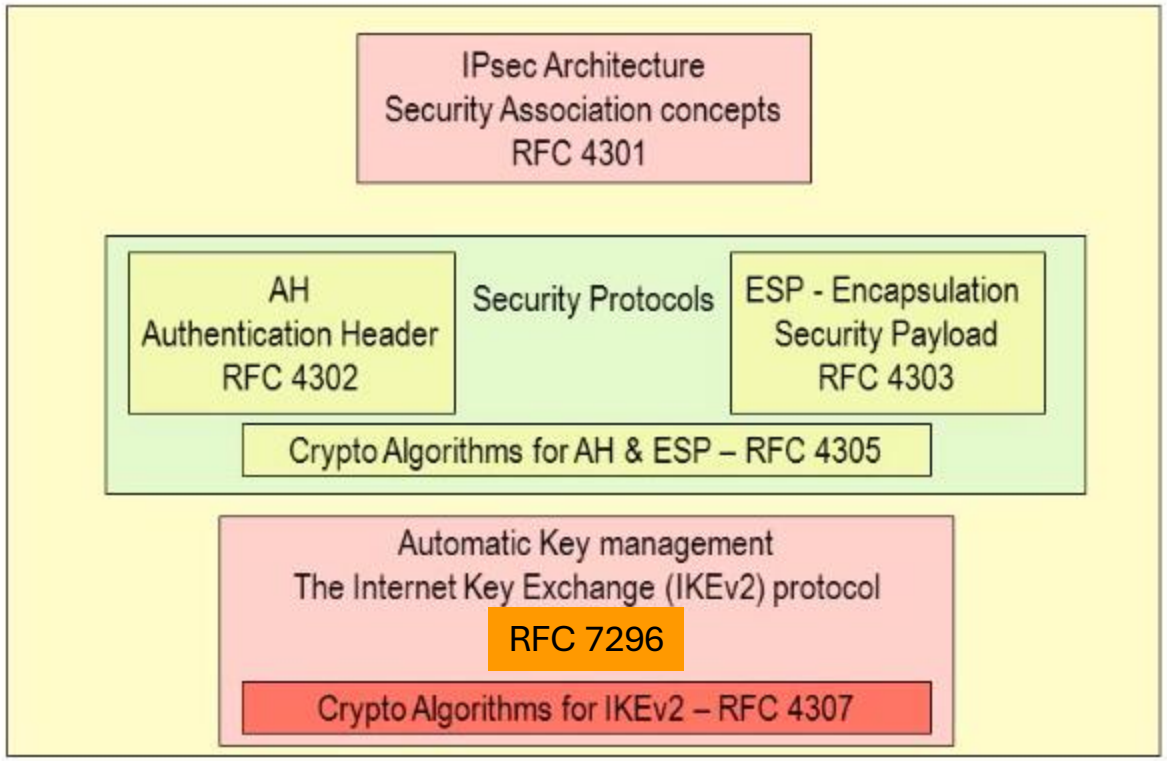


4.3 IPsec Security Protocols (ESP, AH)



IPSec-Spezifikationen

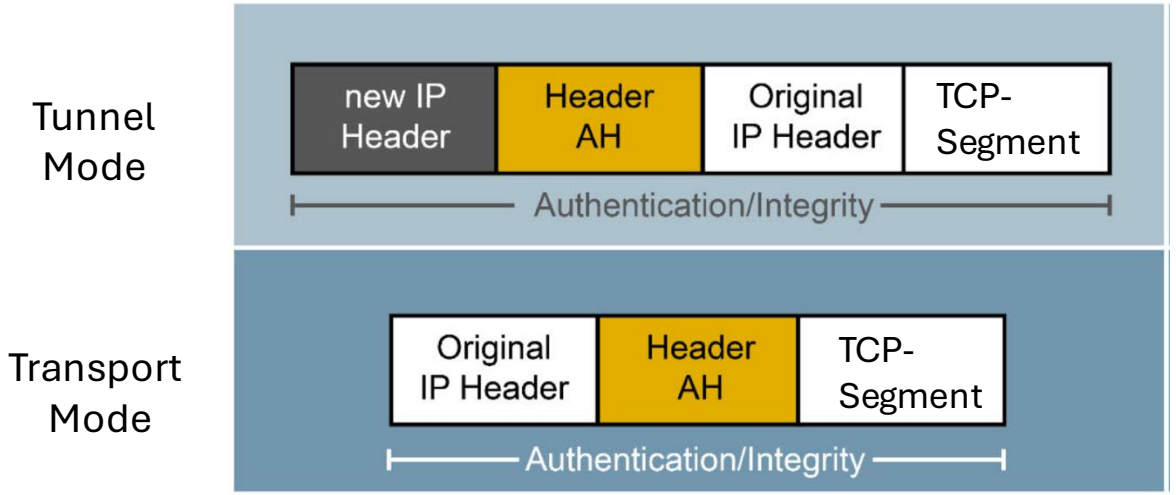
- Aufgrund der Komplexität von IPsec, wird IPsec mittels einer Vielzahl von RFC-Dokumenten beschrieben.
- IPSec verwendet das Internet Key Exchange (IKE) Protokoll, um eine initiale Verbindung (Security Association SA) zwischen den Kommunikationspartner (z. B. VPN-Client und VPN-Server) herzustellen.
  - IKEv2 (Version 2) wurde von der IETF (RFC 7296) standardisiert.
  - IKEv2 authentifiziert die beiden Endpunkte und bestimmt mittels eines sicheren Verfahrens kryptografische Schlüssel.
  - Diese Schlüssel werden anschließend verwendet, um den Datentransfer sicher zu gestalten.
- IPsec verwendet für die verschlüsselte Datenübertragung zwei Sicherheitsprotokolle:

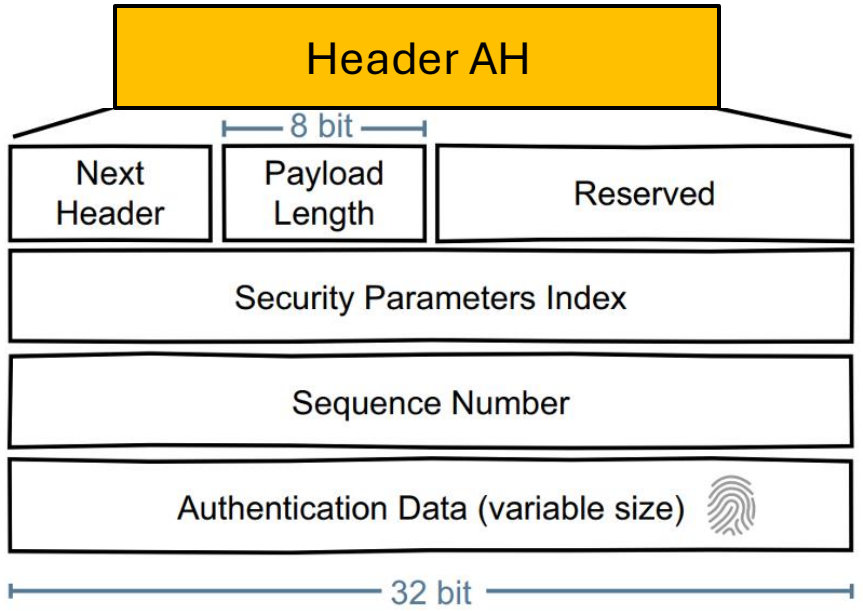


  - AH (Authentification Header Protocol – RFC4302): bietet Authentifizierung für die übermittelten Nachrichten.
  - ESP (Encapsulation Security Payload - RFC4303): bietet Verschlüsselung und Authentifizierung für die übermittelten Nachrichten an.

Authentication Header (AH) Protocol

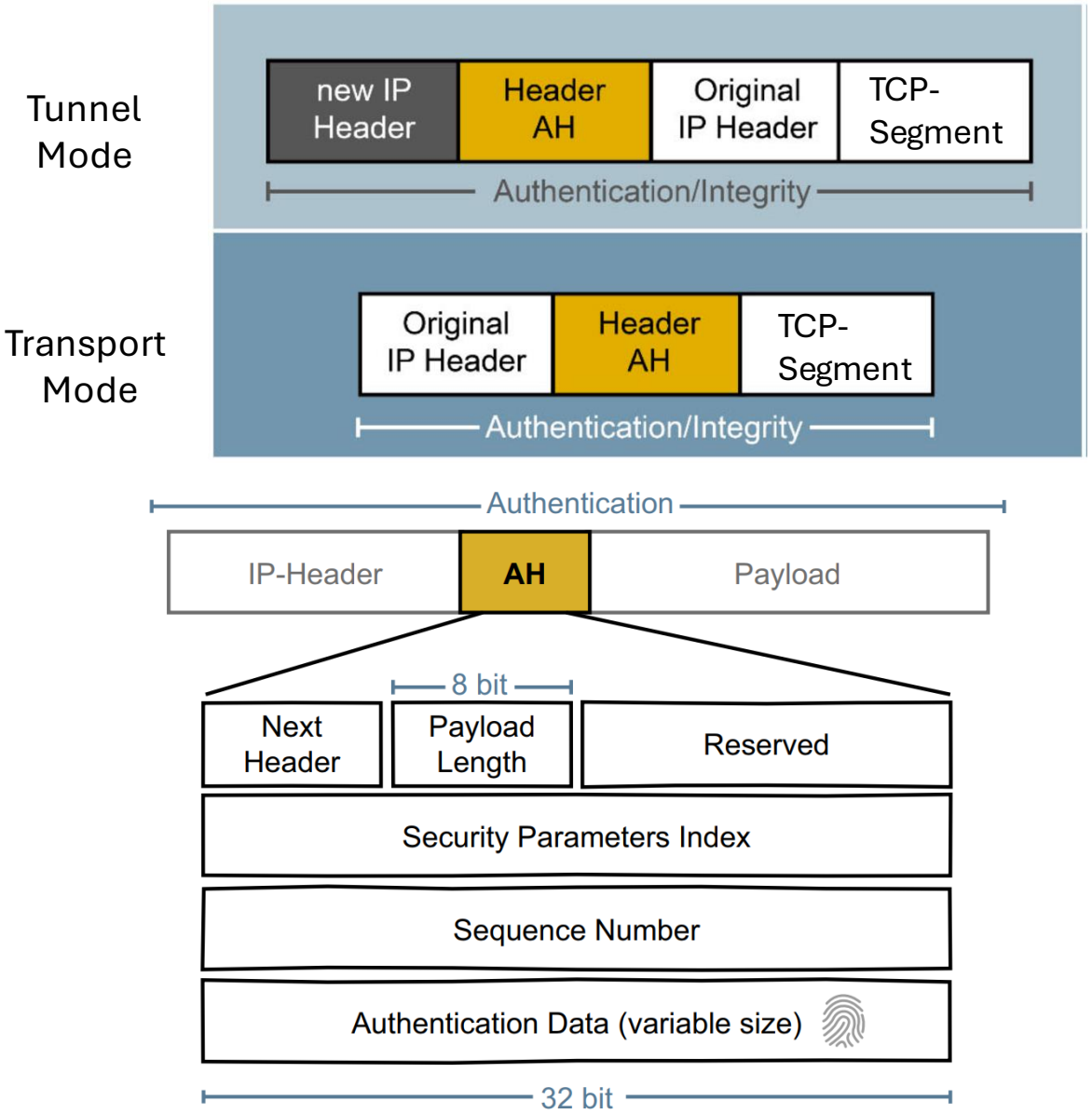
- Das Authentication Header (AH) Protokoll sorgt für die Integrität (Echtheit) der IP-Pakete.
- Dabei wird mit der Hilfe einer HMAC-Funktion über das gesamte IP-Paket (auch den outer IP-Header) ein Hash-Wert berechnet. Die Felder, die während des Transports modifiziert werden, wie Time to Live (TTL), Type of Service (TOS), Flags und Header Checksum, werden bei der HMAC-Berechnung weggelassen.
- AH-Nachrichten besitzen den folgenden Header:
  - Next Header (1B): spezifiziert das Protokoll, das dem AH-Header folgt (4: IPv4, 41: IPv6, 6: TCP, 50: ESP).
  - Payload Length (1B): Länge des AH-Headers in 4Byte Worten minus 2 Byte für (Next Header, Payload Length).
  - Reserved (2B) ist reserviert für zukünftige Funktionen und enthält lauter Nullen.





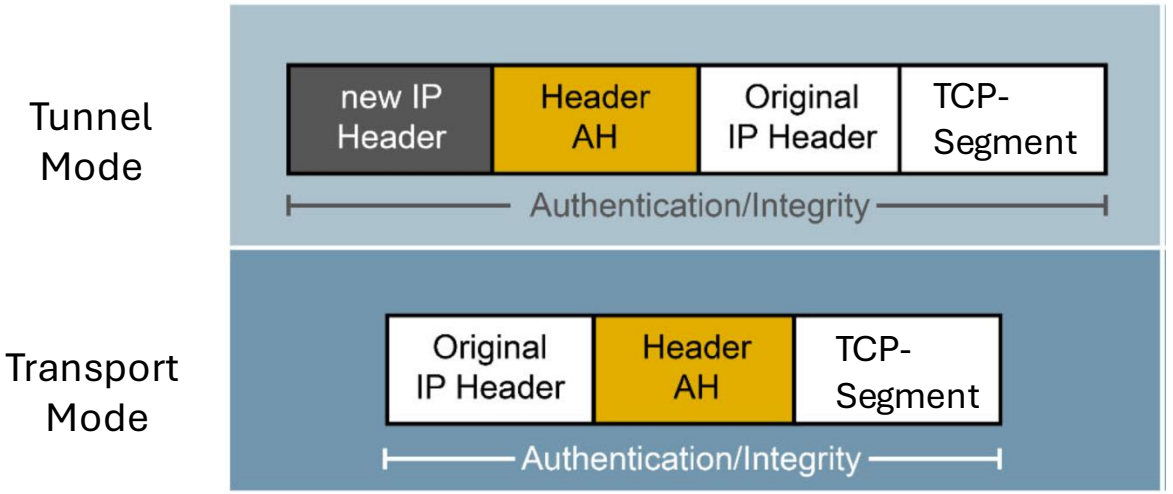
Authentication Header Protocol (AH)

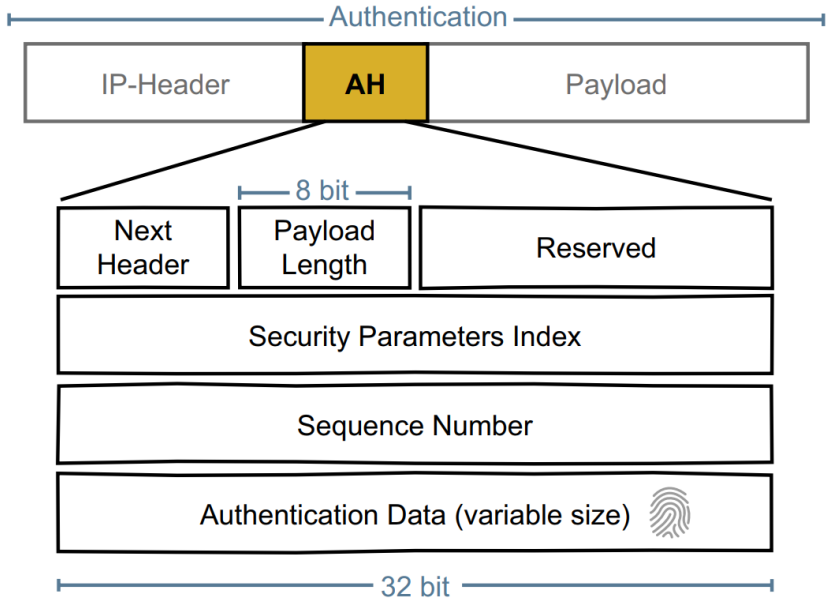
- Security Parameter Index (SPI) ist ein eindeutiger zufälliger 32-Bit Wert zur Kennzeichnung einer Security Association (SA) aus der Sicht eines Endpunktes.
- Sequence Number (SN): 32 Bit-Feld beinhaltet einen eindeutigen Zähler (Anti-Replay-Angriff) pro gesendetes IPsec-Paket und pro SA (und damit pro Senderichtung):
  - 👉 Initial: SN=0
  - 👉 Erstes Paket: SN = 1
  - 👉 Zweites Paket: SN=2
  - ...
  - 👉 Beim Erreichen des Maximalwert $2^{32} - 1$ wird eine neue SA ausgehandelt, um sicherzustellen das jedes Paket einer SA einzigartig ist.



Authentication Header Protocol (AH)

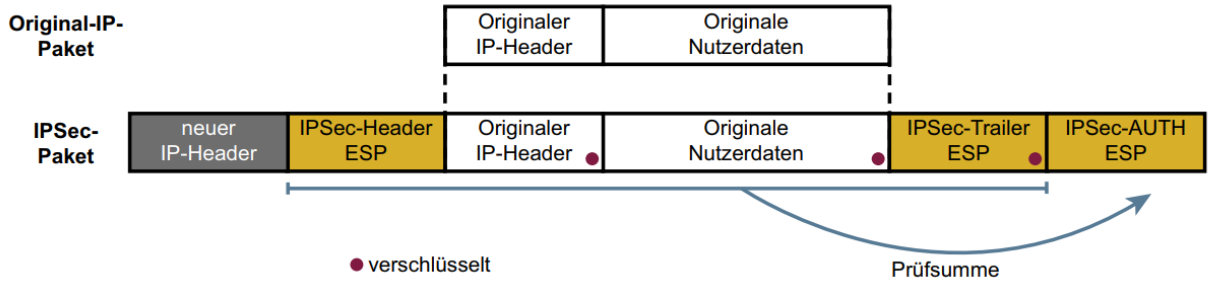
- Authentication Data ist ein Feld, das das Ergebnis der HMAC-Berechnung erhält. Die Länge ist variabel, aber ein ganzzahliges Vielfaches von 32 Bit.
  z.B.: HMAC-SHA-256 erzeugt ein 8x32Bit=256Bit Integrity Check Value (ICV).

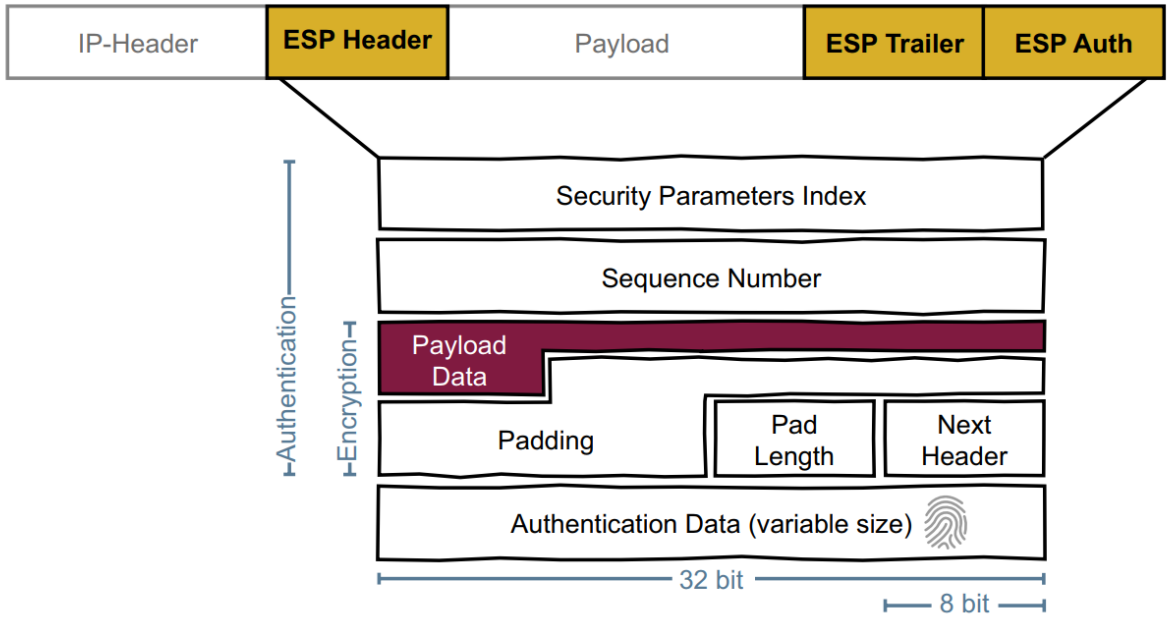




Encapsulated Security Payload (ESP)

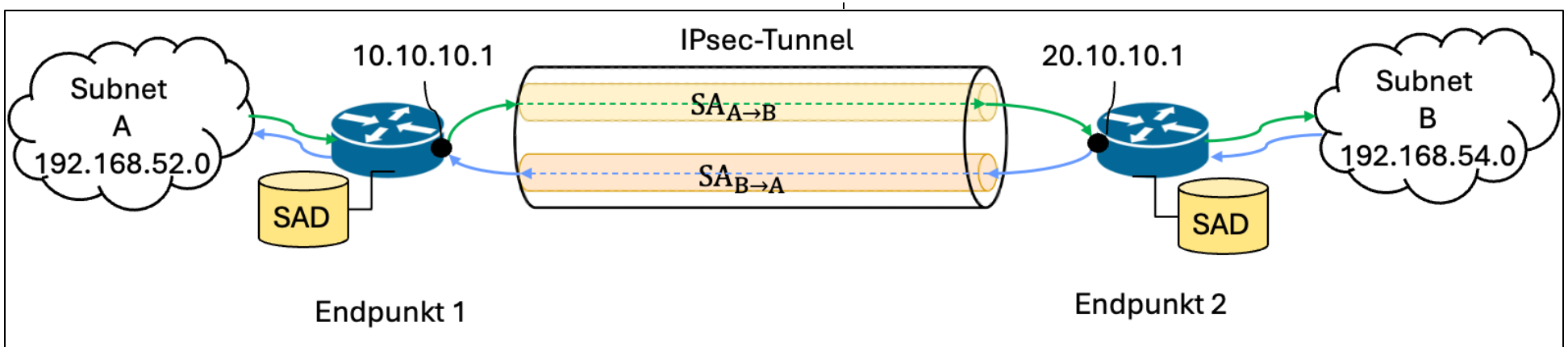
- Das ESP-Protokoll sorgt im Tunnel-Mode für die Authentizität und Verschlüsselung des originalen IP-Headers und der Nutzdaten mit einem symmetrischen Verschlüsselungsverfahren (z.B.: AES-GCM).
  - Die Integrität und Authentizität der IP-Pakete bezieht sich bei ESP nicht auf den „Outer IP-Header“.
  - SPI, SN, Next Header und Authentification Data haben dieselbe Bedeutung und Größe wie beim AH.
  - Padding wird zum Auffüllen des Payloads genutzt (0–255 Byte), falls der Verschlüsselungs-Algorithmus dies erfordert.
  - Pad Length (8Bit) beschreibt die Anzahl an Bytes, die für das Padding verwendet wurden.
- Im Unterschied zu ESP bezieht sich die Authentizität von AH auch auf den "äußeren" IP-Header, sodass die Kombination von AH und ESP zu einer gesteigerten Sicherheit führt.





Security Association

- Im Kontext von IPsec (Internet Protocol Security) ist eine Security Association (SA) eine unidirektionale Verbindung, die zwischen zwei Hosts hergestellt wird, um eine sichere Kommunikation zu ermöglichen.
- Eine SA spezifiziert die Sicherheitsparameter, für die sichere Kommunikation.
- Hauptmerkmale einer SA:
  - Unidirektional: Jede SA sichert den Datenverkehr in eine Richtung. Für eine bidirektionale Kommunikation sind zwei SAs erforderlich.
- Die Sicherheitsparameter einer SA werden in der SA-Datenbank (SAD) pro Endpunkt gespeichert.
  - Jede aktive SA wird in der SAD gespeichert.
  - Bei eingehenden Paketen sucht das System die SPI in der SAD, um die zugeordneten Parameter (z.B.: Schlüssel) zu extrahieren.
  - Bei ausgehenden Paketen wird mit den Daten aus der Security Policy DB eine passende SA in der SAD gesucht
- SAs können manuell konfiguriert werden oder dynamisch mit dem Internet Key Exchange (IKE) Protokoll ausgehandelt werden.



Security Association Datenbank

- Ein SA-Eintrag umfasst die folgenden Parameter:
  - Security Parameters Index (SPI): Eine eindeutige Kennung für die SA.
  - IP-Adressinformationen: Die Quell- und Ziel-IP-Adressen der Gateways, die mit der SA verknüpft sind.
  - IPsec-Protokoll: Gibt an, ob die SA für AH oder ESP ist.
  - Betriebsmodus: Entweder Transportmodus (schützt nur die Nutzlast) oder Tunnelmodus (schützt das gesamte IP-Paket).
  - Lebensdauer der SA: Definiert, wie lange die SA gültig ist, basierend auf einer Zeitdauer oder einem übertragenen Datenvolumen.
  - Verschlüsselungsalgorithmus und -schlüssel: Definiert den Algorithmus (z. B. AES-GCM) und den zugehörigen kryptografischen Schlüssel, der für die Verschlüsselung verwendet wird.
  - Authentifizierungsalgorithmus und -schlüssel: Definiert den Algorithmus (z. B. HMAC-SHA) und den zugehörigen Schlüssel an, der für die Integritätssicherung verwendet wird.

Security Policy Database (SPD)

- Security Policy Database SPD: Jeder Eintrag in der SPD ist eine Richtlinie, die beschreibt, wie mit Paketen verfahren werden soll, die bestimmte Kriterien erfüllen.
  - Kriterien basieren auf den Quell- und Ziel-IP-Adressen der beiden Gateways, und dem zu sichernden Netzwerkverkehr definiert durch Quell- und Ziel-IP-Adressen der Clients, erlaubte Protokolle (z. B. TCP, UDP, ICMP) und erlaubte Portnummern.
- Policies pro Kriterium und Gerät können sein
  - Protect, Bypass oder Discard den IP-Verkehr zwischen 2 Endpunkten.
  - Verwenden von AH, ESP für den Datenverkehr zwischen den Endpunkten.
  - Definition der erlaubten kryptografischen Algorithmen zwischen den Endpunkten.
  - Maximale Lebensdauer (Expiration Time) einer IPsec-SA nach der eine Neuaushandlung des Schlüsselmaterials durchgeführt wird.
  - Größe eines Anti-Replay Windows (default size: 64)

Security Policy Database (SPD)

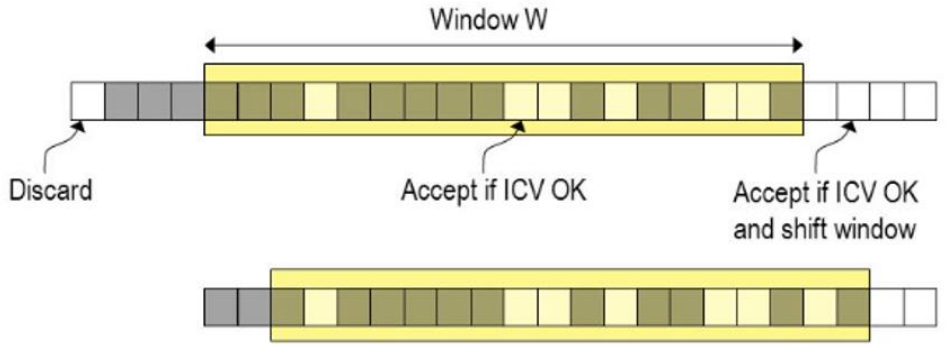
- Die untenstehende Tabelle zeigt beispielhaft eine SPD.
  - Die SPD enthält eine Liste an Richtlinien.
  - Auf den eingehenden und ausgehenden Datenverkehr werden diese Richtlinien angewandt.
  - Diese Regeln können beispielsweise mittels
    `ip xfrm policy` für LINUX, oder
    `crypto map` für Netzwerkgeräte
    definiert werden.

- Anhand des Regelwerkes in der SPD wird entschieden, wie ein Packet verarbeitet wird:
  `DISCARD – verwerfen des Packetes`
  `BYPASS – Paket unverändert durchlassen`
  `PROTECT – mittels ESP schützen`

Richtung	Quell-IP	Ziel-IP	Protokoll	Quell-Port	Ziel-Port	Action
Outbound	192.168.1.0/24	172.16.10.0/24	TCP	Any	ANY	PROTECT:ESP Tunnel
Inbound	172.16.10.0/24	192.168.1.0/24	TCP	ANY	ANY	PROTECT:ESP Tunnel
Outbound	ANY	ANY	ICMP	ANY	ANY	BYPASS
Outbound	192.168.1.0/24	201.10.10.0/24	ANY	ANY	ANY	DISCARD

Anti - Replay - Window

- IPsec Anti-Replay schützt Netzwerke vor Anti-Replay-Angriffen durch Verwendung eines gleitenden Fenstermechanismus namens Anti-Replay-Fenster.
- Das Anti-Replay-Protokoll vergleicht die Sequenznummer jedes empfangenen IPsec-Pakets mit dem aktuellen IPsec-Paketsequenznummernbereich des gleitenden Fensters.
- Wenn die Sequenznummer nicht im aktuellen Sequenz-nummernbereich des Fensters W liegt, wird das Paket als wiederholtes Paket betrachtet und verworfen.



Perfect Forward Secrecy (PFS)

Perfect Forward Secrecy (PFS) (de: "Perfekte vorwärtsgerichtete Geheimhaltung") stellt sicher, dass der Inhalt vergangener verschlüsselter Kommunikationen selbst dann nicht kompromittiert werden kann, wenn der langfristige private Schlüssel eines Kommunikationsteilnehmers in die Hände eines Angreifers gerät.

- PFS wird durch den Einsatz von temporären Sitzungsschlüsseln erreicht.
  - Für jede Sitzung wird mittels eines Schlüsselaustauschverfahren ein neuer, unabhängiger Sitzungsschlüssel generiert wird. Dieser temporäre Sitzungsschlüssel wird auch als ephemerer Schlüssel bezeichnet.
  - Dies verhindert, dass ein Angreifer ältere Kommunikation entschlüsseln kann.
  - Schlüsselaustauschverfahren**:** Bei einem Schlüsselaustauschverfahren generieren beide Kommunikationsparteien (z. B. Client und Server) temporäre Schlüsselpaare, aus denen ein gemeinsamer temporärer Sitzungsschlüssel abgeleitet wird. Typischerweise wird Diffie-Hellman (DH) oder Elliptic Curve Diffie-Hellman (ECDH) verwendet.
  - **Privater Schlüssel der Teilnehmer**: Der langfristige private Schlüssel (z. B. der RSA- oder ECDSA-Schlüssel des Servers) wird nur zur Authentifizierung der Verbindung genutzt, nicht aber zur Generierung der Sitzungsschlüssel.
- Im Falle von IPsec wird PFS erreicht durch
  - Explizite Verwendung von Ephermal Diffie-Hellmann für jede SA in Kombination mit Noncen.
  - Endliche Lebensdauer einer SA, verbunden mit einem Re-Keying.

Überblick: Dynamisches Aushandeln einer IPsec-SA

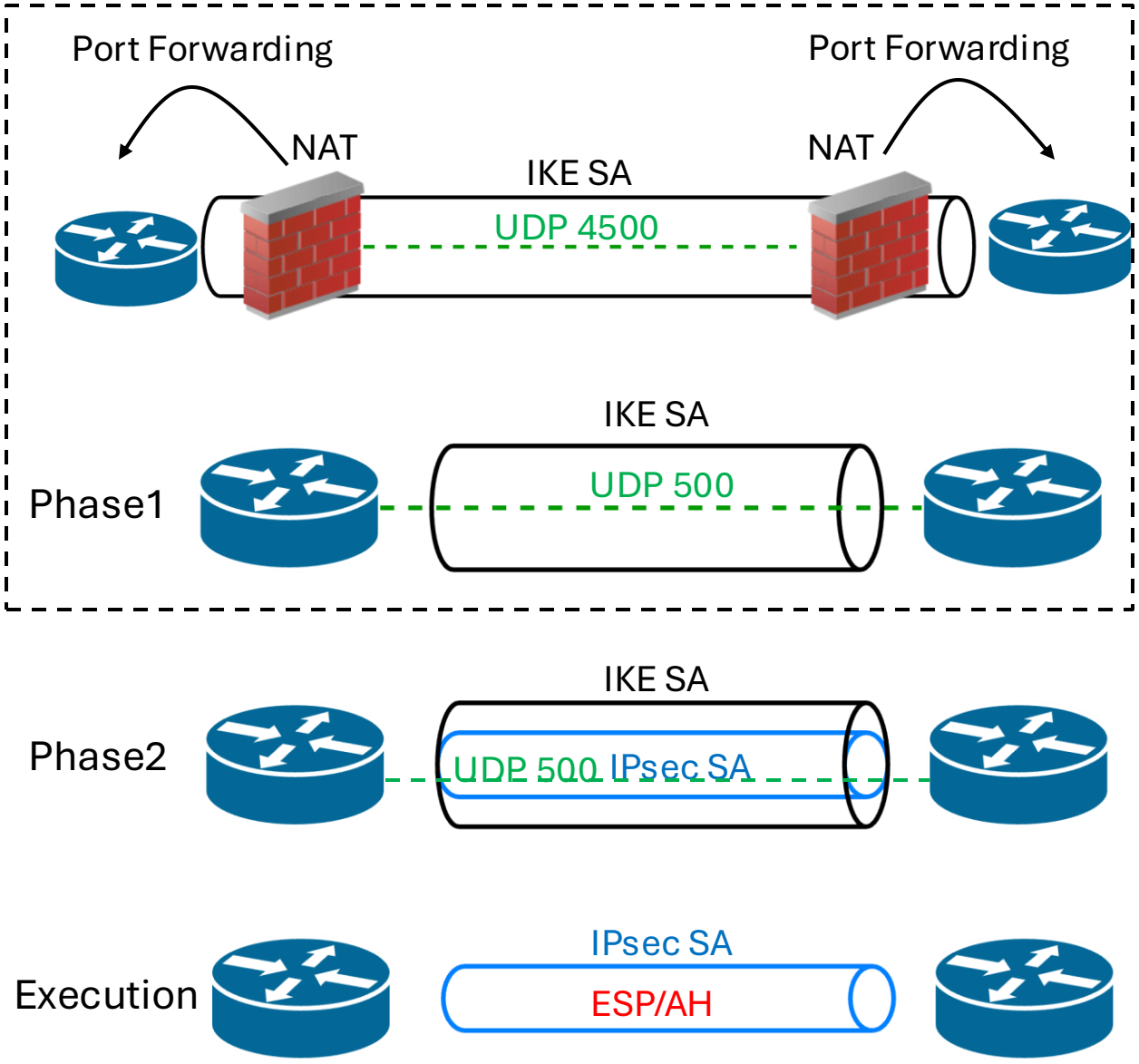
- Dynamisches Aushandeln einer IPsec-SA:
  - IPsec verwendet das Internet Key Exchange (IKE)-Protokoll, um eine IPsec-SA automatisch einzurichten.
  - Das Aushandeln erfolgt in 2 Phasen.
- IKE-SA (Phase 1)
  - Zuerst wird eine IKE_SA aufgebaut, die eine sichere Kommunikationsverbindung zwischen zwei Endpunkten herstellt.
  - Nutzt das Ephermal Diffie-Hellman Verfahren für den sicheren Schlüsselaustausch und legt die Verschlüsselungs- und Integritätsalgorithmen fest.
  - Authentifiziert die beiden Endpunkte mit deren digitalen Zertifikaten.
  - Diese SA bleibt über eine längere Zeit aktiv und wird für die Verwaltung der IPsec-SAs genutzt.
- Ipsec-SA (Phase 2)
  - Nach der IKE SA werden eine oder mehrere IPsec-SAs aufgebaut.
  - Diese IPsec-SAs werden für die eigentliche Datenübertragung verwendet.
  - Jede Richtung (eingehende und ausgehende Kommunikation) hat eine eigene IPsec-SA mit eigener SPI (Security Parameter Index).

Algorithmen: IPsec unterstützt eine Vielzahl kryptografischer Algorithmen.

- Verschlüsselung: AES-GCM
- Authentifizierung: HMAC-SHA3
- Schlüsselaustausch: Diffie-Hellman

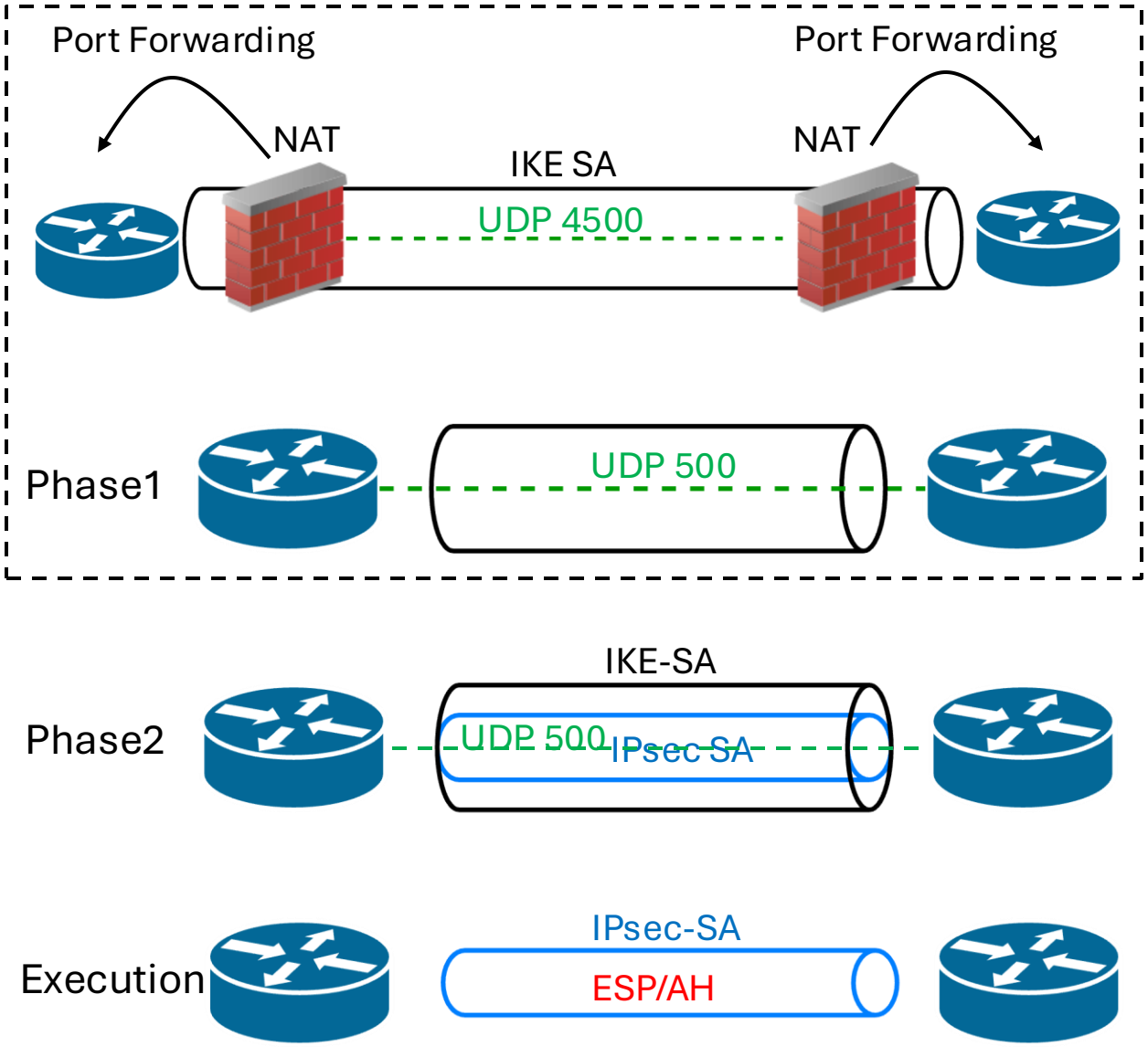
Internet Key Exchange Protocol (IKE)

- IKE verwendet einen zweiphasigen Prozess für die Einrichtung einer IPSec-SA.
- Phase 1: IKE-SA
  - IKE arbeitet auf Layer-5 (Sitzungsschicht) und verwendet zum Aufbau einer IKE-SA das Transportprotokoll UDP mit dem Port 500. Setzen die Endpunkte NAT ein, werden die Nachrichten über einen UDP-Tunnel auf Port 4500 ausgetauscht.
  - Schritt1 - IKE_SA_INIT-Nachrichten: In dieser Phase handeln die IKE-Peers einen gemeinsamen Satz an kryptografischen Protokollen aus und bestimmen mittels Diffie-Hellmann einen gemeinsamen Satz an kryptografischen Schlüsseln.
    Der nachfolgende Nachrichtenverkehr ist dann durch die Schlüssel der IKA_SA_Init Phase geschützt.



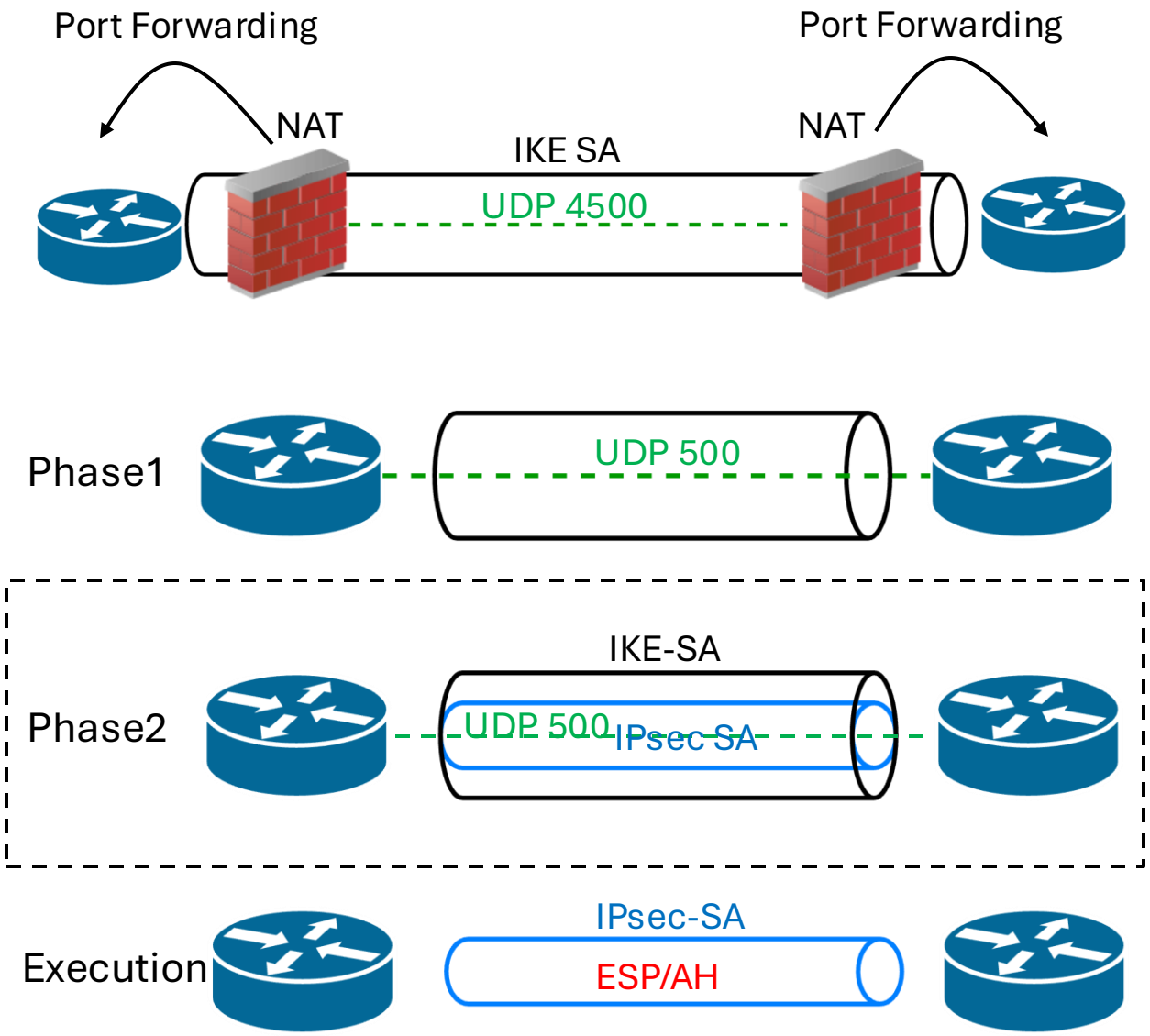
Internet Key Exchange Protocol (IKE)

- Phase 1: IKE-SA
  - Schritt2 - IKE_AUTH - Nachrichten: Die Endpunkte authentifizieren sich gegenseitig (X.509 Zertifikat, Pre-Shared-Key, EAP), und richten einen sicheren und authentifizierten Kommunikationskanal namens IKE-SA ein.
  - Die IKE-SA wird für die sichere Kommunikation der anschließenden Phase2, der Aushandlung der IPsec-SA verwendet.
  - Die IKE-SA bleibt während der Sitzung zwischen 2 Endpunkten bestehen.



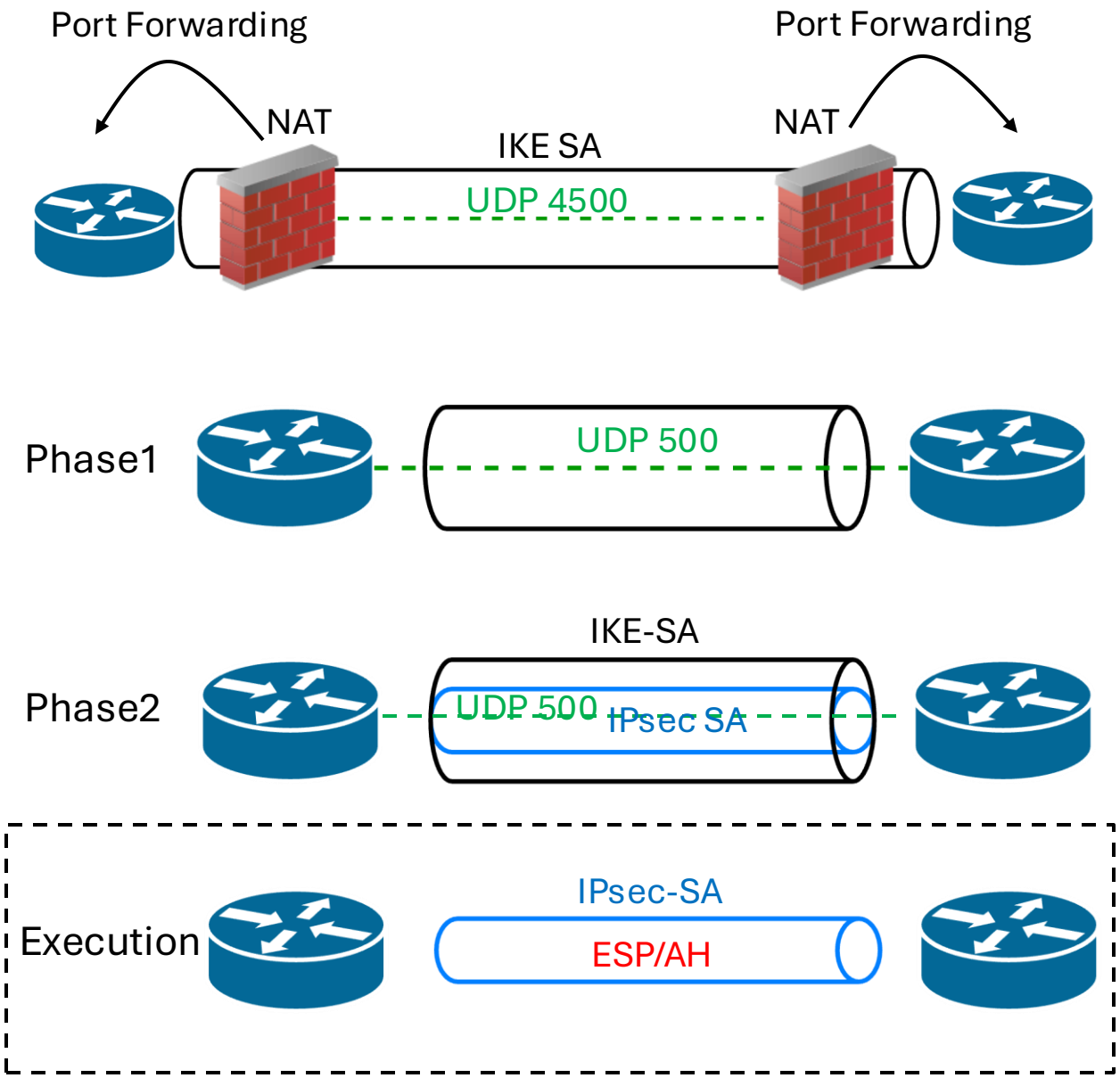
Internet Key Exchange Protocol (IKE)

- Phase 2: IPsec-SA
  - Schritt3 – Create_Child_SA - Nachrichten: Sobald die IKE-SA eingerichtet ist, findet per IKE die Aushandlung einer IPSec-SA auch Child-SA genannt statt. Dabei können pro IKE-SA mehrere IPsec-SA ausgehandelt werden.
  - Pro Child-SA werden deren Sicherheitsparameter (siehe vorne) ausgehandelt und mittels Diffie-Hellmann kryprografische Schlüssel für die Child-SA bestimmt.
  - Eine Authentifizierung wird für die Erzeugung der Child-SA nicht mehr benötigt.



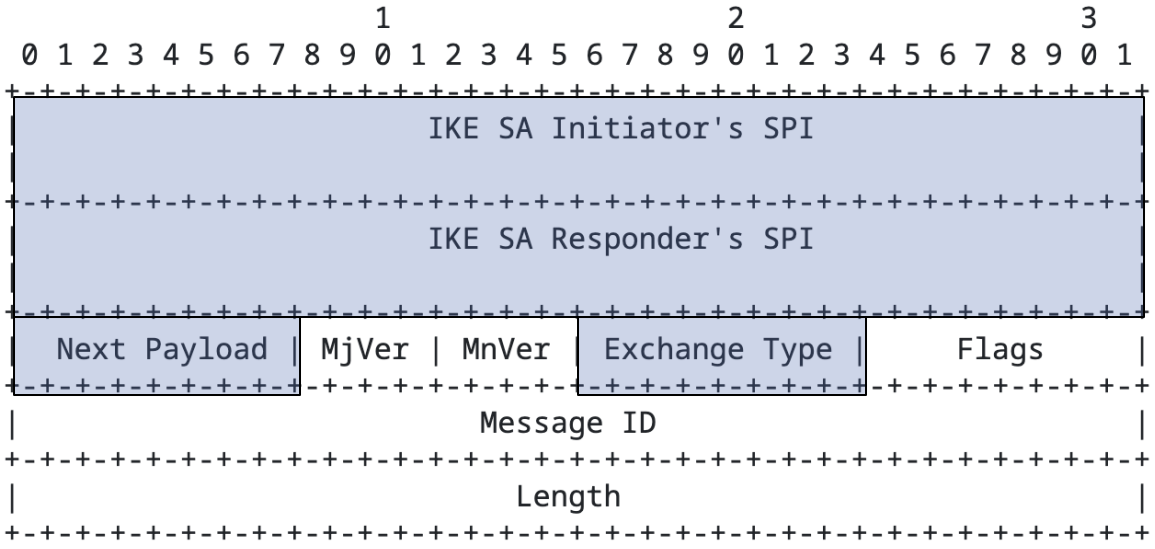
Execution mit AH oder ESP

- Execution: Datentransfer mit AH oder ESP
  - Sobald eine IPsec-SAs eingerichtet ist, kann sie verwendet werden, um den tatsächlichen Benutzerdatenverkehr zwischen den Endpunkten zu schützen.
  - Daten werden mithilfe der vereinbarten SAs sicher zwischen den Peers übertragen. Jedes Paket wird gemäß der IPsec-Richtlinie (AH- oder ESP-, Transport- oder Tunnelmodus) verarbeitet.
  - Die Kommunikation per IPsec findet auf Layer-3 statt.
- Ende einer IPsec-SA:
  - Aushandeln neuer Schlüssel für eine IPsec-SA nach Erreichen der Lebensdauer.
  - Löschen einer IPsec-SA wenn einer der Kommunikationspartner die Kommunikation beendet.



IKEv2 - Header

- Den Aufbau des IKE-Headers zeigt nebenstehende Abbildung.
- Initiator-SPI (64Bit): Wird vom Initiator während des ersten Austauschs festgelegt.
- Responder-SPI (64Bit): Wird vom Responder während der ersten IKE_SA_INIT-Antwort festgelegt.
- Next Payload (8 Bit): Beschreibt den Typ des nächsten Payloads.



- Exchange Type (8Bit): Definiert den Nachrichtentyp

Payload Type	Wert	Beschreibung
SA (Security Association)	33	Definiert Sicherheitsparameter für die SA.
KE (Key Exchange)	34	Austausch von Diffie-Hellman-Parameter.
IDi (Identification Initiator)	35	Identifikation des Initiators. (z.B: IP-Adresse, FQDN)
TSi (Traffic Selector Initiator)	44	Definiert den zu schützenden Datenverkehr des Initiators. (Subnetzwerk: Start IP, End IP, Start TCP Port, End TCP Port)
...	...	...

Exchange Type (IKEv2)	Wert	Beschreibung
IKE_SA_INIT	34	Initialisiert eine IKE SA mit Schlüsselaustausch
IKE_AUTH	35	Authentifiziert beide Endpunkte und erstellt die erste IPsec SA.
CREATE_CHILD_SA	36	Erstellt oder erneuert eine IPsec SA (z. B. nach Ablauf).
INFORMATIONAL	37	Dient für Status- oder Fehlermeldungen (keine SA-Erstellung).

IKEv2 - Header

- Min Version (4Bit) : Das MnVer-Feld (Minimum Version) ist ein 4-Bit-Feld im IKE-Header. Es gibt die niedrigste unterstützte IKE-Version an, die der Sender unterstützt.

```
MnVer = 2 für IKEv2
MnVer = 1 für IKEv1 und IKEv2
```

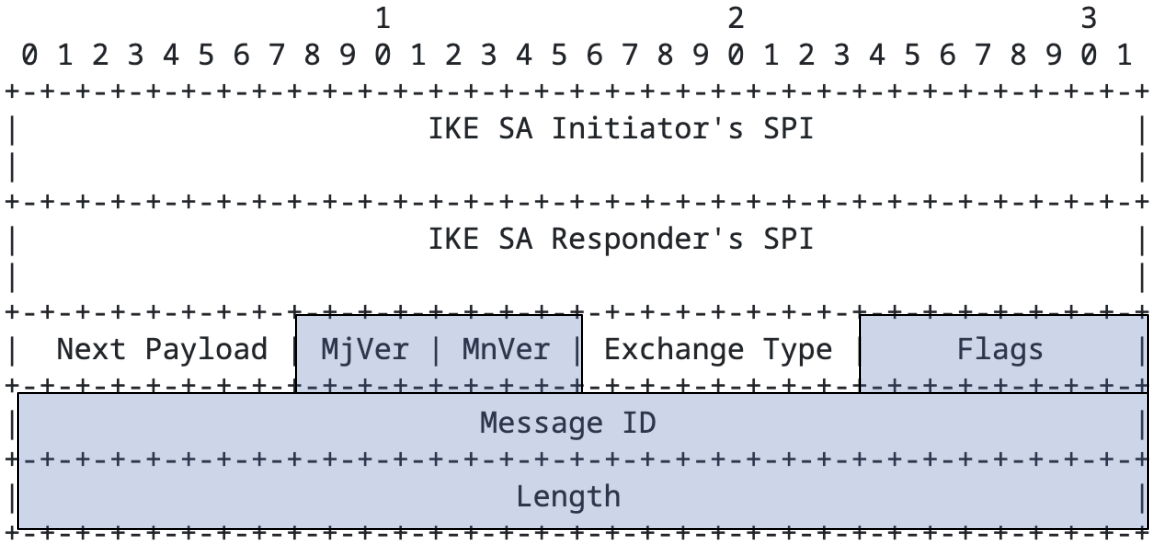
- Major Version (4Bit): Das MjVer-Feld (Major Version) bestimmt die für die Kommunikation verwendete **IKE-Version.**

```
MjVer = 2 für IKEv2
```

- Flags (8Bit): Beschreiben das Verhalten einer Nachricht

```
Bit 3 = 1 (Initiator-Flag I): Nachricht kommt vom Initiator.
Bit 4 = 1 (Version-Flag V): Absender erwartet höhere IKE-Version
Bit 5 = 1 (Responder-Flag R): Nachricht ist eine Antwort.
Bit 1-3, 6-7: reserved (=0)
```

```
+-+-+-+-+-+-+-+-+
|X|X|R|V|I|X|X|X|
+-+-+-+-+-+-+-+-+
```



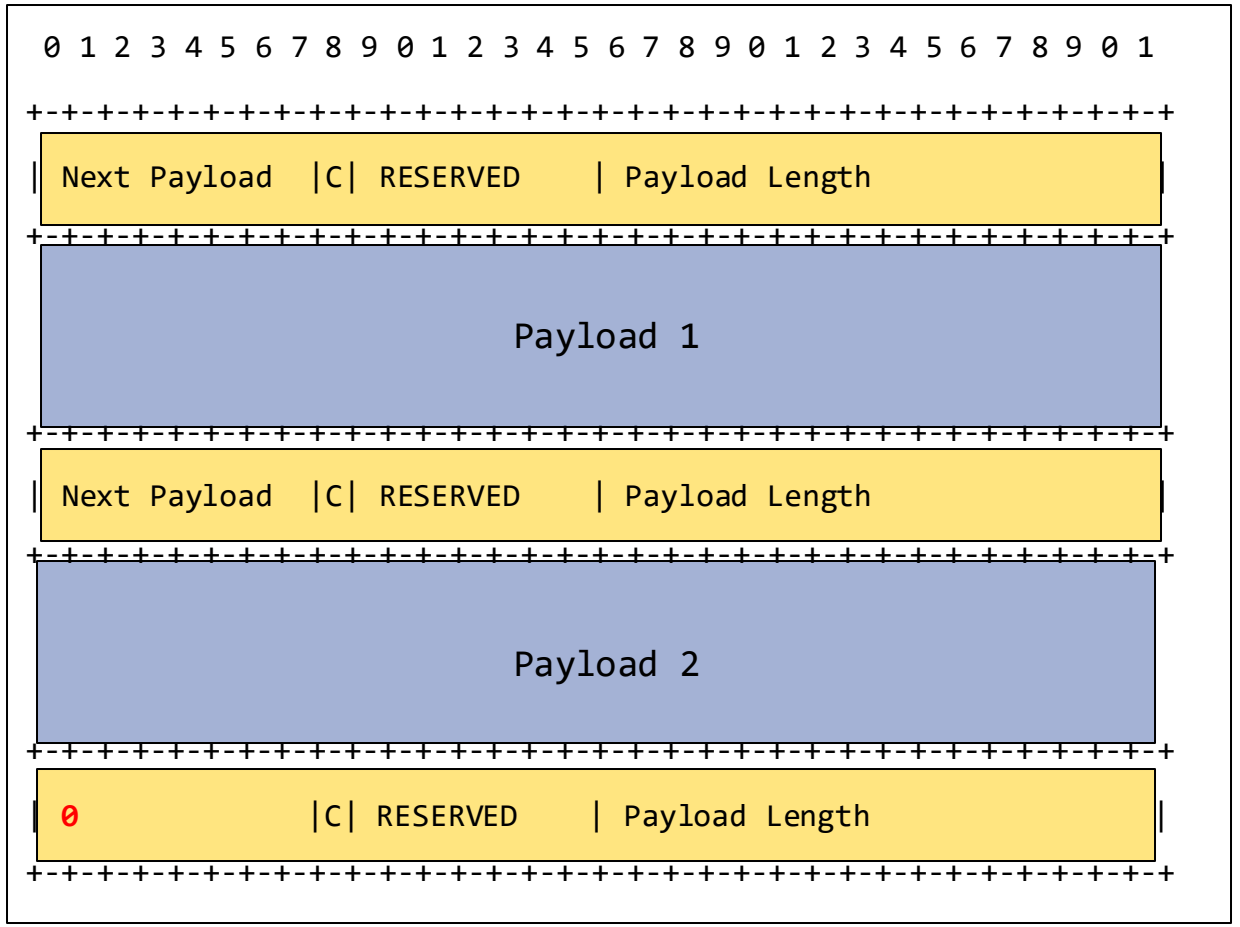
- Message-ID (32Bit): Nachrichten-ID (4 Oktette) – wird für jede gesendete Nachricht um 1 erhöht. Antwort-Nachricht verwendet die empfangene Message-ID.
  - Dient zur Steuerung der erneuten Übertragung verlorener Pakete und zum Abgleichen von Anfragen mit Antworten.
  - Verhinderung von Replay-Angriffen durch die Definition eines Gültigkeitsfensters.
- Length (32Bit): Länge der gesamten Nachricht (Header + Nutzdaten) in Byte.

IKEv2 - Payload

- Eine IKE-Nachricht kann aus mehreren Payloads bestehen.
- Jeder IKE-Payload beginnt mit einem Payload-Header.
- Next Payload (8 Bit) : Kennung für den nächsten Nutzlasttyp analog zum Feld im IKE-Header. Wenn die aktuelle Nutzlast die letzte in der Nachricht ist, ist dieses Feld 0.
- Critical Bit C (1 Bit): Gibt an, ob die Nutzlast für die Verarbeitung kritisch ist.

> C= 1 (kritisch) : Der Empfänger muss die Payload verstehen. Falls nicht, muss er die ganze Nachricht verwerfen und keine Antwort senden.
>
> C= 0 (unkritisch) : Der Empfänger darf diesen Payload ignorieren, der Rest der Nachricht wird verarbeitet.

- Reserved (7Bit) : Reserviert (=0)



- Payload Length (16 Bit): Gibt die Gesamtlänge der zugehörigen Nutzlast einschließlich ihres Headers an.

IKEv2 – Payload: Transformationen

- Im Kontext des IKE-Protokolls (Internet Key Exchange) beschreiben Transformationen die kryptografischen Algorithmen und deren Parameter.
- Transformationen liefern die notwendigen Details zu den kryptografischen Methoden, die zum Sichern der Daten verwendet werden sollen. Dazu gehören Verschlüsselungsalgorithmen, Integritätsalgorithmen und Schlüsselaustauschmechanismen.
- Jede Transformation wird durch eine Transformations-ID identifiziert, die ein numerischer Wert ist, der den verwendeten Algorithmus oder die verwendete Methode darstellt.

- Es gibt im allgemeinen drei Haupttypen von Transformationen:
- Verschlüsselungstransformationen: Definieren den zum Verschlüsseln der Daten verwendeten Algorithmus (z. B. AES, 3DES).
- Integritätstransformationen: Gibt den Algorithmus zur Gewährleistung der Datenintegrität an (z. B. HMAC-SHA1, HMAC-SHA256).
- Key-Exchange Transformation: Gibt die Parameter für Schlüsselaustauschverfahren an (z.B.: Diffie-Hellman-Schlüsselaustausch)

IKEv2 – Payload: Proposals

- Transformationen werden innerhalb von IKE in Form von Proposals organisiert.
- Jedes Proposal kann mehrere Transformationen enthalten.
- Initiator und Responder tauschen beim Erstellen einer SA, in Form von Proposals, welche Algorithmen für Verschlüsselung, Integrität und Schlüsselaustausch verwendet werden sollen.
- Beispiel: Payload für SA-Payload 33
  - Der Initiator sendet einen oder mehrere Proposals für kryptografische Protokolle innerhalb der SA-Nutzlast.
  - Der Responder wählt einen Vorschlag aus (sofern kompatibel) und sendet ihn in der Antwort zurück.

IKE-Header

```
Initiator SPI:   0x1234567890abcdef
Responder SPI:   0x0000000000000000
Next Payload:    33 (Security Association (SA))
Exchange Type:   34 (IKE_SA_INIT)
Flags:           I=1
Message ID:      1
Length:          nn bytes
```

Payload Header

```
Next Payload:    0
Critical Bit:    0
Payload Length: nn bytes
```

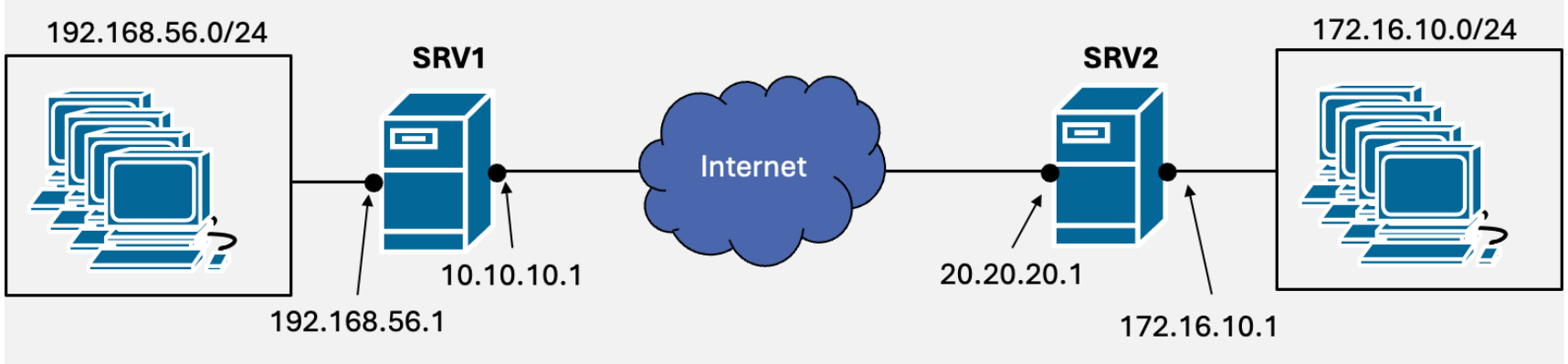
Payload

```
Number of Proposals: 2

Proposal Number : 1

   Protocol ID: ESP

   SPI-Size: 4 bytes

   Number of Transforms: 3

     Transform ID: 12 (Encryption: AES-CBC)

     Transform ID: 2 (Integrity: HMAC-SHA-256)

     Transform ID: 14 (Diffie-Hellman Group: Group 14)
```

Manuelle Konfiguration einer SA

- Manuelle Konfiguration einer SA:
  - Beim „Manual Keying“ werden die notwendigen Schlüssel von einem Administrator manuell generiert.
  - Der Administrator erstellt auf beiden Endpunkten per Command Line Interface eine SA.

- Beispiel: Konfiguration einer SA zwischen den 2 Linux-Gateway mit den IP-Adressen 10.10.10.1 und 20.20.20.1. Die SA verwendet ESP mit Tunnel. Der SA wird eine SPI von 256 zugewiesen.
- Verwenden der ip xfrm state (XFRM)-Schnittstelle des Linux-Kernels.




```
$ ip xfrm state add src 10.10.10.1 dst 20.20.20.1 \        //Gateway for SA
    proto esp spi 0x100 \                                  //IPsec Protocol and SPI
    mode tunnel \                                          //Betriebsmodus: Tunnel
    auth sha256 0xabcdef1234567890abcdef12345... \         //Auth: Algorithmus SHA256 & kryptograf. Schlüssel
    enc aes 0x1234567890abcdef1234567890abcd...            //Enc.: Algorithmus AES & kryptograf. Schlüssel
```

Manuelle Konfiguration einer Policy

- Manuelle Konfiguration einer Policy:
  - Eine Policy für das abgebildete Szenario, bei dem ein IPsec-Tunnel mittels ESP zwischen 2 LINUX-Servern erzeugt wird, der den Nachrichtenverkehr zwischen den Netzwerken (192.165.1.0/24 → 172.16.10.0/24) mit IPsec schützt.

- Ein Template tmpl definiert die Art der zugehörigen SA, die für den angegebenen Datenverkehr genutzt werden muss.
  - Wenn ein Paket eine policy-Regel erfüllt, prüft das System die verfügbaren SAs basierend auf dem Template.
  - Ohne eine zugehörige state-Definition (SA) kann der Verkehr nicht verschlüsselt oder geschützt werden.

```
tmpl src 192.168.1.0/24 dst 172.16.10.0/24 proto esp mode tunnel
```

Erklärung:

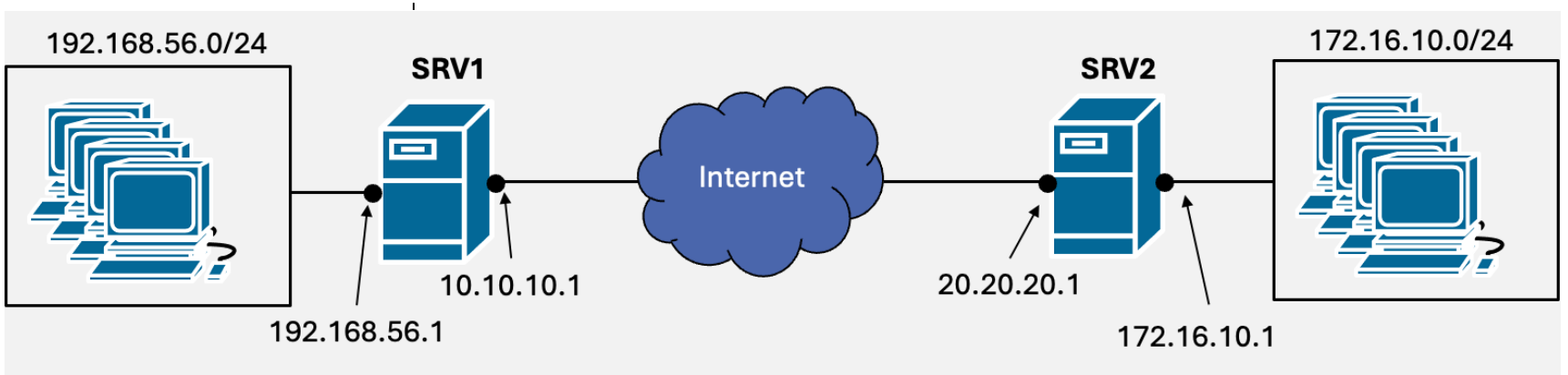
src 10.10.10.1 : Quelle Tunnel-Endpunkt

dst 20.20.20.1/24: Ziel Tunnel-Endpunkt

dir out: Richtung von Quell- zum Netzwerk

tmpl src: Art der SA

```
$ip xfrm policy add src 10.10.10.1 dst 20.20.20.1 \
  dir out tmpl src 192.168.1.0/24 dst 172.16.10.0/24  proto esp mode tunnel
```



Aufgabe 4: IPsec

1. **Grundlagen IPsec**
   a. Erklären Sie die Unterschiede zwischen den verschiedenen IPsec-Protokollen (AH und ESP).
   b. Was ist der Zweck des Security Parameter Index (SPI)?
   c. Beschreiben Sie die Rolle von IKE (Internet Key Exchange) im IPsec-Prozess.
   d. Welche Nachrichtentypen kennt IKE.
   e. Was sind die Unterschiede zwischen Tunnel- und Transportmodus in IPsec?
   f. Welche Authentifizierungsmethoden können mit IKE verwendet werden?
2. **Netzwerk-Sicherheit und Angriffe**
   a. Diskutieren Sie die Vor- und Nachteile von IPsec im Vergleich zu anderen VPN-Technologien (z. B. SSL/TLS).
   b. Analysieren Sie , ob IPsec gegen die folgenden verschiedene Angriffsarten schützt. Begründen Sie ihre Antwort.
      Flooding-Angriff, Man-in-the-Middle-Angriffe, Replay-Angriffe, IP-Spoofing, MAC-Spoofing, Session-HiJacking, Route HiJacking.