

2N01 Einführung in IPsec

Andreas Aurand
Compaq Solution and Service Center

Agenda

- IPsec Architektur
 - Sicherheitsprotokolle
 - AH
 - ESP
 - IPComp
 - Security Associations
 - Schlüsselaustauschverfahren
 - Diffie-Hellman Verfahren
 - ISAKMP und IKE
 - Authentifizierung der Partner
 - über Pre-Shared Keys
 - über RSA-Signaturen
- IPsec und NAT

IPsec

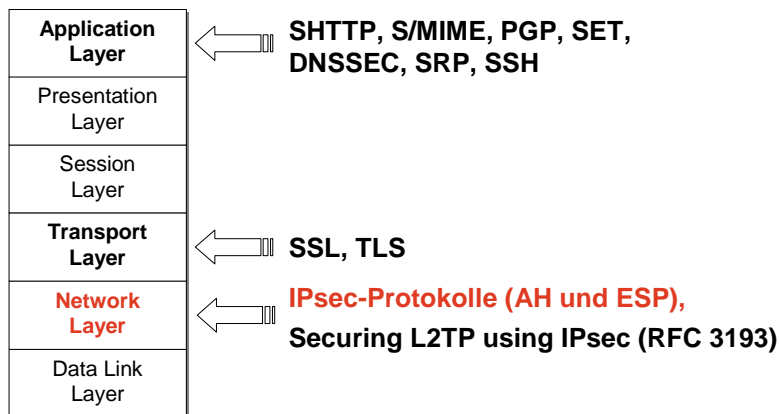
Architektur für die Sicherheit im Internet

Grundziele der Kryptografie

- **Vertraulichkeit (*Confidentiality, Privacy*)**
 - Kein unbefugter Zugriff auf den Inhalt einer Nachricht oder Datei möglich.
- **Integrität (*Integrity*)**
 - Manipulationen an der Nachricht werden entdeckt (z. B. Einfügen, Weglassen, Ersetzung von Teilen).
- **Authentizität (*Authenticity*)**
 - **Identitätsnachweis:** Eine Kommunikationspartei soll einer anderen ihre Identität zweifelsfrei beweisen können.
 - **Herkunftsnachweis:** A soll B beweisen können, daß eine Nachricht von ihm stammt und nicht verändert wurde.
- **Nichtabstreitbarkeit (*Verbindlichkeit, Non-Repudiation*)**
 - **Nichtabstreitbarkeit der Herkunft:** Es soll unmöglich sein, das Absenden einer bestimmten Nachricht nachträglich zu bestreiten.
 - **Nichtabstreitbarkeit des Erhalts:** Es soll unmöglich sein, den Erhalt einer gesendeten Nachricht nachträglich zu bestreiten.

IPsec Architektur

- Services für die Sicherheit in einem IP-Netzwerk auf Ebene der Netzwerkschicht.



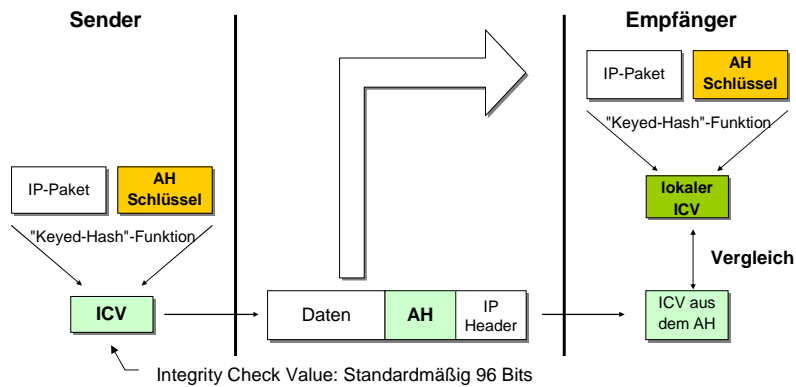
IPsec Komponenten

- Sicherheitsprotokolle**
 - ESP (IP Encapsulating Security Payload)
 - Vertraulichkeit** durch Verschlüsselung der Daten
 - AH (Authentication Header)
 - Integrität** und **Herkunftsnachweis** der Daten
 - IPComp (IP Payload Compression Protocol)
 - Komprimierung der Daten
- Protokolle zur Schlüsselverwaltung**
 - ISAKMP (Internet Security Association and Key Management Protocol)
 - Festlegung der Paketformate
 - IKE (Internet Key Exchange)
 - Automatischer Schlüsselaustausch und **Identitätsnachweis** der Partner
 - Diffie-Hellman Key Agreement Method
 - Algorithmus eines Schlüsselaustauschverfahrens

- IPsec Architektur
 - Sicherheitsprotokolle
 - AH
 - ESP
 - IPComp
 - Security Associations
 - Schlüsselaustauschverfahren
 - Diffie-Hellman Verfahren
 - ISAKMP und IKE
 - Authentifizierung der Partner
- IPsec und NAT

AH - Authentication Header (RFC 2402)

- Schutz der **Integrität** des kompletten IP-Pakets - inklusive IP Header
- Nachweis der **Authentizität** des IP-Pakets - inklusive IP Header
- Schutz vor **Replay-Attacken** (durch Sequenznummer im AH Header)



AH - Authentication Header

- **Hashfunktion** (Prüfsummenfunktion)
 - Transformation einer beliebigen Bitfolge in einen String fester Länge, der Prüfsumme (Fingerabdruck, *Hash Digest*, Finger Print)
 - z.B. **MD5**, **SHA1** oder **RIPEMD**

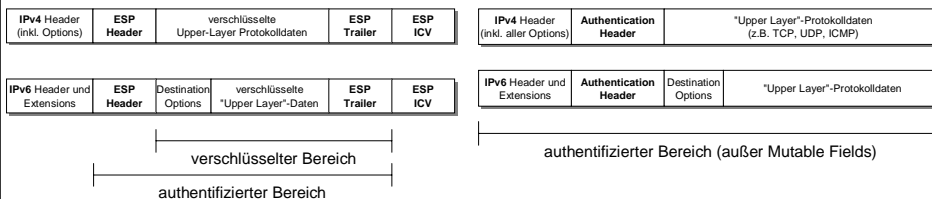
- **Hash-based Message Authentication Code (HMAC)**
 - Aus der Nachricht und dem geheimen Schlüssel erzeugt die Hashfunktion eine Prüfsumme (ICV = Integrity Check Value)
 - z.B. **HMAC-MD5** oder **HMAC-SHA1**
 - Die Prüfsumme wird als zusätzliche Information an die Nachricht angehängen
 - Im Gegensatz zu einer digitalen Signatur wird auf beiden Seiten der gleiche geheime Schlüssel verwendet

ESP - Encapsulating Security Payload (RFC 2406)

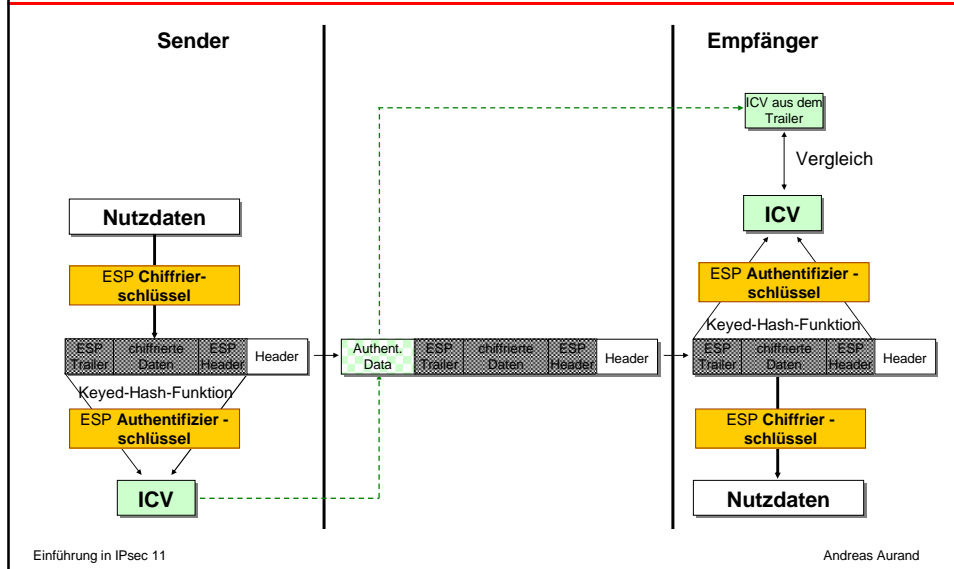
- **Vertraulichkeit** der Nutzdaten und des Datenverkehrs durch symmetrische Verschlüsselung
 - z.B. **DES**, **3DES** oder **AES**

- **Authentizitätsnachweis** und **Integritätsschutz** der Nutzdaten
 - Bei ESP bezieht sich die Integrität und Authentizität **nur auf die Nutzdaten** (inkl. ESP Header) und nicht auf das gesamte Paket
 - Optionale Bestandteile des ESP-Protokolls

- Schutz vor **Replay-Attacken** (durch Sequenznummer im ESP Header)



ESP - Encapsulating Security Payload



IPComp - Payload Compression Protocol (RFC 2393)

- Über IPsec verschlüsselte Daten können auf Data Link Ebene nicht mehr komprimiert werden
 - IP Header Compression oder RTP Header Compression (CRCP) sind wirkungslos
- **IPComp komprimiert die Daten vor der Verschlüsselung**
 - Zusammen mit ESP einzusetzen
 - Unterstützt folgende Algorithmen
 - DEFALTE-Algorithmus RFC 2394
 - LZS-Algorithmus RFC 2395
 - ITU-T V.44 Packet Method RFC 3051
- **RObust Header Compression (ROHC) - RFC 3095**
 - Ermöglicht Komprimierung der Header-Daten bis zum ESP-Header (inklusive)

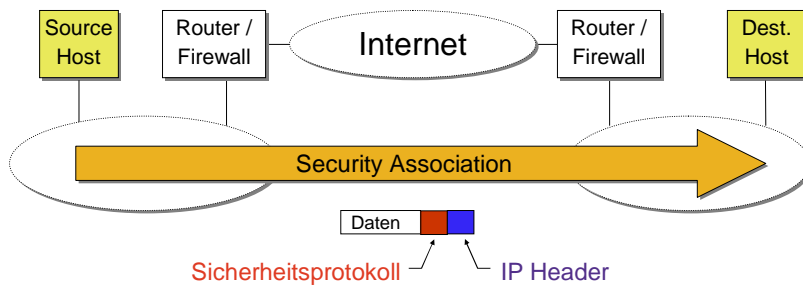
- IPsec Architektur
 - Sicherheitsprotokolle
 - AH
 - ESP
 - IPComp
 - **Security Associations**
 - Schlüsselaustauschverfahren
 - Diffie-Hellman Verfahren
 - ISAKMP und IKE
 - Authentifizierung der Partner
- IPsec und NAT

SA - Security Association

- **Unidirektionale** Festlegung von Services die eine sichere Kommunikation gewährleisten sollen.
 - Für bidirektionale Übertragung, muss immer eine **inbound** und **outbound** SA aufgebaut werden.
 - Gelten für ESP, AH und IPComp als auch ISAKMP
- **Protection Suite**
 - Die Menge aller Services, die innerhalb einer SA von den Sicherheitsprotokollen eingesetzt werden.
 - DES bzw. 3DES für ESP
 - HMAC-MD5 für AH
 - DES, SHA1 und Pre-shared Keys für ISAKMP
- Definition der zu schützenden Daten über **Selektoren**
 - Meistens Source und Destination IP-Adresse (evtl. noch mit Port)
 - Auch als **Identities** oder **Proxy Identities** bezeichnet

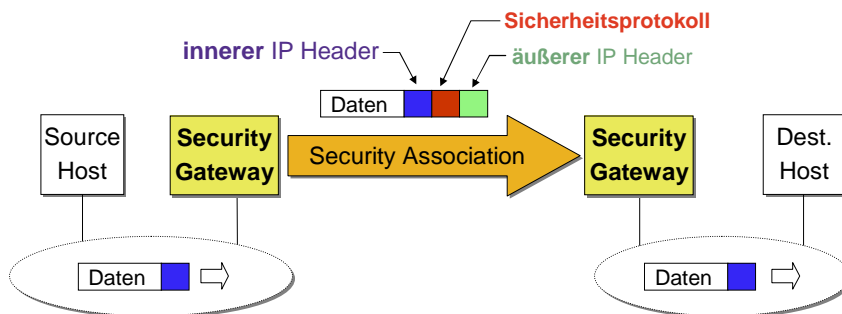
Transport Mode Security Association

- Header des Sicherheitsprotokolls folgt direkt nach dem IP-Header
 - Schutz der direkten Verbindungen zwischen zwei Endsystemen
 - Z.B. L2TP-Verbindung zwischen einem Windows 2000 Client und Server, Telnet-Verbindung zum Router, GRE-Verbindungen



Tunnel Mode Security Association

- Header des Sicherheitsprotokolls folgt nach dem Header der Tunnelverbindung (dem **outer IP Header**) und vor dem Header des ursprünglichen IP-Pakets (dem **inner IP Header**)
 - Handelt sich um einen IP-Tunnel; ermöglicht Aufbau von VPNs
 - Hauptsächlich für Verbindungen zwischen Security-Gateways



Beispiel für IPsec Security Associations

```
# show crypto ipsec sa
local ident (addr/mask/prot/port): (16.204.7.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (192.168.56.0/255.255.255.0/0/0)
current_peer: 16.204.7.91
  PERMIT, flags={origin_is_acl,}
#pkts encaps: 30, #pkts encrypt: 30, #pkts digest 0
#pkts decaps: 30, #pkts decrypt: 30, #pkts verify 0
#pkts compressed: 0, #pkts decompressed: 0
#pkts not compressed: 0, #pkts compr. failed: 0, #pkts decompress failed: 0
#send errors 1, #recv errors 0

inbound esp sas:
  spi: 0x658DB0D9(1703784665)
    transform: esp-des ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 2000, flow_id: 1, crypto map: ETH
    sa timing: remaining key lifetime (k/sec): (4607996/2103)
    IV size: 8 bytes
    replay detection support: N

inbound ah sas:
inbound pcp sas:

outbound esp sas:
  spi: 0x1B11CBD6(454151126)
    transform: esp-des ,
    in use settings ={Tunnel, }
    slot: 0, conn id: 2001, flow_id: 2, crypto map: ETH
    sa timing: remaining key lifetime (k/sec): (4607995/2103)
    IV size: 8 bytes
    replay detection support: N

outbound ah sas:
outbound pcp sas:
```

Proxy Identities / Selektoren

Verwendete Sicherheitsmechanismen

Sicherheitsprotokoll

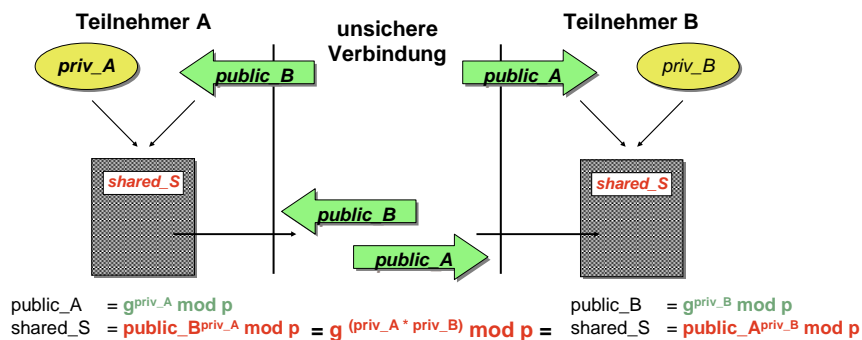
- IPsec Architektur
 - Sicherheitsprotokolle
 - AH
 - ESP
 - IPComp
 - Security Associations
 - Schlüsselaustauschverfahren
 - Diffie-Hellman Verfahren
 - ISAKMP und IKE
 - Authentifizierung der Partner
- IPsec und NAT

Schlüsselaustauschverfahren

- Wie bekomme ich die für ESP und AH benötigten Schlüssel ?
 - **manuelle Konfiguration**
 - aufwendig
 - unflexibel
 - unsicher
 - **automatisches Erzeugen**
 - sehr flexibel
 - Verbindungsaufbau dauert länger

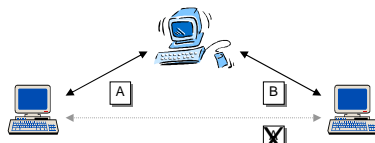
Diffie-Hellman Algorithmus (RFC 2631)

- Phase 1 erzeugt über DH-Algorithmus das Verschlüsselungsmaterial
 - Asymmetrische Verschlüsselung basierend auf diskretem Logarithmusproblem
 - Aus dem Verschlüsselungsmaterial erzeugen die Partner lokal die Sitzungsschlüssel



Diffie-Hellman Algorithmus

- Festlegung der Primzahl und des Generator durch **Diffie-Hellman-Gruppe**
 - Die Werte sind in RFCs (z.B. RFC 2412) definiert
 - **DH-Gruppe 1**: modulare Exponentialgruppe mit einer **768 Bit Primzahl**
 - **DH-Gruppe 2**: modulare Exponentialgruppe mit einer **1024 Bit Primzahl**
 - **DH-Gruppe 5**: modulare Exponentialgruppe mit einer **1536 Bit Primzahl**
 - Generator hat immer den Wert 2
- DH-Algorithmus ist anfällig für **Man-in-the-Middle-Attacken**.
 - Lösung: Anschließende **Authentifizierung** des Partners



Einführung in IPsec 21

Andreas Aurand

- IPsec Architektur
 - Sicherheitsprotokolle
 - AH
 - ESP
 - IPComp
 - Security Associations
 - Schlüsselaustauschverfahren
 - Diffie-Hellman Verfahren
 - **ISAKMP und IKE**
 - Authentifizierung der Partner
- IPsec und NAT

Einführung in IPsec 22

Andreas Aurand

ISAKMP/IKE

▪ ISAKMP (RFC 2408)

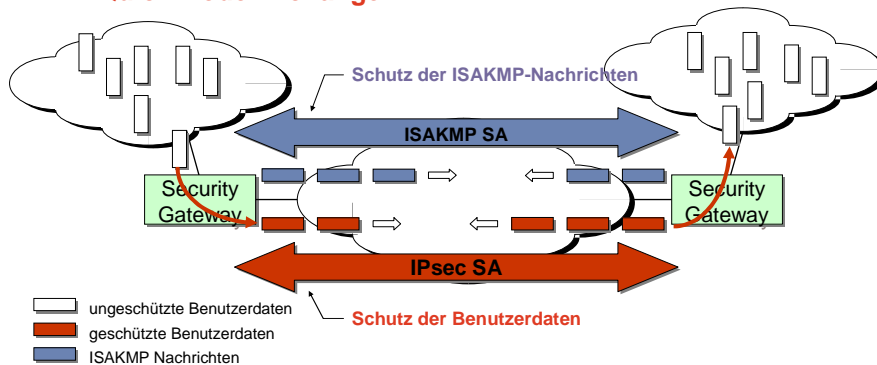
- *Internet Security Association and Key Management Protocol*
- Definition der Prozeduren und Paketformate, die zum Aufbau und zum Management von Security Associations notwendig sind.
- Es kann von allen Protokollen für das Management von SAs eingesetzt werden (z.B. IPsec, TLS oder OSPF).
- Keine Festlegung wie der vertrauliche und authentifizierte Aufbau von SAs und kryptografischen Schlüsseln zu erfolgen hat.

▪ IKE (Internet Key Exchange – RFC 2409)

- Authentifizierter Austausch von kryptografischen Schlüsseln innerhalb der IPsec-Architektur.

ISAKMP Phase 1 und Phase 2

- Phase 1: Schutz der ISAKMP-Nachrichten durch eine ISAKMP (IKE) SA
 - **Main Mode** oder **Aggressive Mode Exchange**
- Phase 2: Schutz der Benutzerdaten durch die IPsec SAs
 - **Quick Mode Exchange**

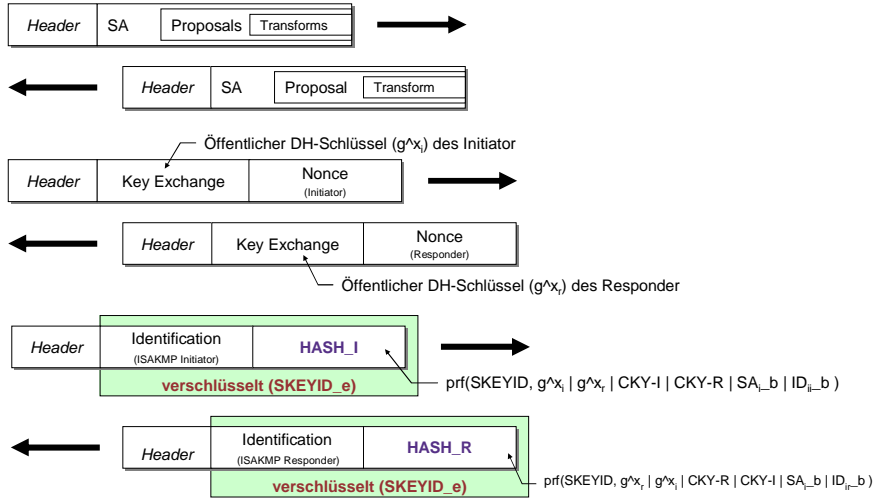


- IPsec Architektur
 - Sicherheitsprotokolle
 - AH
 - ESP
 - IPComp
 - Security Associations
 - Schlüsselaustauschverfahren
 - Diffie-Hellman Verfahren
 - ISAKMP und IKE
 - **Authentifizierung der Partner**
- IPsec und NAT

Authentifizierung der Partner

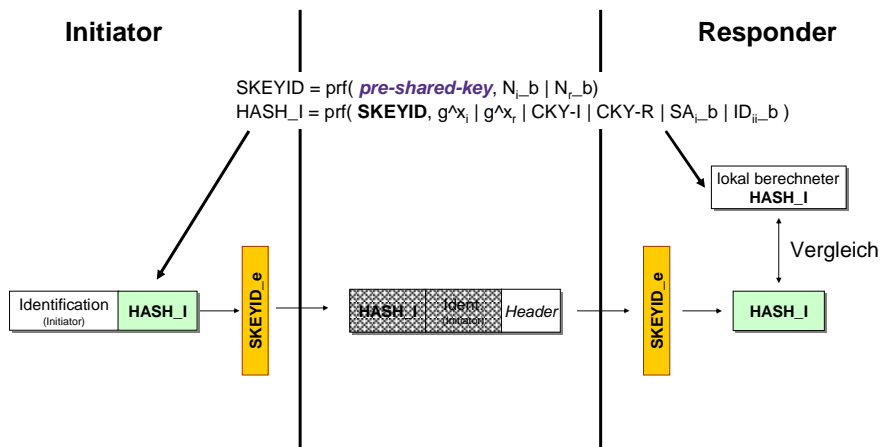
- **Pre-Shared Keys** (Vordefinierte Schlüssel)
 - Manuelle Definition der Authentifizierungsschlüssel
 - Nicht identisch mit dem symmetrischen Sitzungsschlüssel von AH oder ESP
 - Schlechte Skalierbarkeit aber relativ schnell zu implementieren
 - Speichern der Pre-Shared Keys auf einem Radius-Server (z.B. **VPN Client**)
- **RSA-Verschlüsselung**
 - Jedes System muss die öffentlichen RSA-Schlüssel der Partner kennen
 - Keine gute Skalierbarkeit aber sicherer als Pre-Shared Keys
- **RSA-Signaturen**
 - Public-Key-Infrastruktur (**PKI**) notwendig
 - Sehr gute Skalierbarkeit und sehr flexibel
- Proprietäre Lösungen
 - Windows 2000 verwendet z.B. standardmäßig Kerberos

Authentifizierung über Pre-Shared Keys



Authentifizierung über Pre-Shared Keys

- Authentifizierung erfolgt über das Wissen des Pre-Shared Keys



Trace - ISAKMP Protection Suite aushandeln

```

IPSEC(sa_request): ,
(key eng. msg.) src= 16.204.7.97, dest= 16.204.7.91,
src_proxy= 16.204.7.0/255.255.255.0/0/0 (type=1),
dest_proxy= 192.168.56.0/255.255.255.0/0/0 (type=1),
protocol= ESP, transform= esp-des ,
lifedur= 3600s and 4608000kb,
spi= 0x658DB0D9(1703784665), conn_id= 0, keysize= 0, flags= 0x4004
ISAKMP: received ke message (1/1)
ISAKMP: local port 500, remote port 500
ISAKMP (0:1): beginning Main Mode exchange
ISAKMP: Main Mode packet contents (flags 0, len 72):
    SA payload
    PROPOSAL
    TRANSFORM
ISAKMP (0:1): sending packet to 16.204.7.91 (I) MM_NO_STATE.

```

Proxy Identities = Was soll IPsec schützen

Sicherheitsprotokoll und Sicherheitsmechanismen

```

ISAKMP (0:1): received packet from 16.204.7.91 (I) MM_NO_STATE
ISAKMP: Main Mode packet contents (flags 0, len 72):
    SA payload
    PROPOSAL
    TRANSFORM
ISAKMP (0:1): processing SA payload. message ID = 0
ISAKMP (0:1): found peer pre-shared key matching 16.204.7.91
ISAKMP (0:1): Checking ISAKMP transform 1 against priority 5 policy
ISAKMP: encryption DES-CBC
ISAKMP: hash SHA
ISAKMP: default group 1
ISAKMP: auth pre-share
ISAKMP (0:1): atts are acceptable. Next payload is 0

```

Trace - Austausch der DH-Schlüssel

```

CryptoEngine0: generate alg parameter
CRYPTO_ENGINE: Dh phase 1 status: 0
CRYPTO_ENGINE: Dh phase 1 status: 0
ISAKMP (0:1): SA is doing pre-shared key authentication using id type ID_IPV4_ADDR
ISAKMP: Main Mode packet contents (flags 0, len 172):
    KE payload
    NONCE payload
    VENDOR payload
ISAKMP (0:1): sending packet to 16.204.7.91 (I) MM_SA_SETUP

```

Öffentlicher DH-Schlüssel des Initiator

```

ISAKMP (0:1): received packet from 16.204.7.91 (I) MM_SA_SETUP
ISAKMP: Main Mode packet contents (flags 0, len 172):
    KE payload
    NONCE payload
    VENDOR payload
ISAKMP (0:1): processing KE payload. message ID = 0
CryptoEngine0: generate alg parameter
ISAKMP (0:1): processing NONCE payload. message ID = 0
ISAKMP (0:1): found peer pre-shared key matching 16.204.7.91
CryptoEngine0: create ISAKMP SKEYID for conn id 1
ISAKMP (0:1): SKEYID state generated
ISAKMP (0:1): processing vendor id payload
ISAKMP (0:1): speaking to another IOS box!

```

Öffentlicher DH-Schlüssel des Responder

Trace - Authentifizierung des Partners

```

ISAKMP (1): ID payload
next-payload : 8
type         : 1
protocol     : 17
port        : 500
length      : 8
ISAKMP (1): Total payload length: 12
CryptoEngine0: generate hmac context for conn id 1
ISAKMP: Main Mode packet contents (flags 1, len 64):
    ID payload
    HASH payload
ISAKMP (0:1): sending packet to 16.204.7.91 (I) MM_KEY_EXCH

```

```

ISAKMP (0:1): received packet from 16.204.7.91 (I) MM_KEY_EXCH
ISAKMP: Main Mode packet contents (flags 1, len 68):
    ID payload
    HASH payload
ISAKMP (0:1): processing ID payload. message ID = 0
ISAKMP (0:1): processing HASH payload. message ID = 0
CryptoEngine0: generate hmac context for conn id 1
ISAKMP (0:1): SA has been authenticated with 16.204.7.91
CryptoEngine0: clear dh number for conn id 1

```

Sender erzeugt die lokale Prüfsumme und vergleicht sie mit dem empfangenen Hashwert

Trace - IPsec Protection Suite aushandeln

```

ISAKMP (0:1): beginning Quick Mode exchange, M-ID of -1279237069
ISAKMP: Quick Mode packet contents (flags 1, len 156):
    HASH payload
    SA payload
    PROPOSAL
    TRANSFORM
    NONCE payload
    ID payload
    ID payload
ISAKMP (0:1): sending packet to 16.204.7.91 (I) QM_IDLE

```

```

ISAKMP (0:1): received packet from 16.204.7.91 (I) QM_IDLE
ISAKMP: Quick Mode packet contents (flags 1, len 164):
    HASH payload
    SA payload
    PROPOSAL
    TRANSFORM
    NONCE payload
    ID payload
    ID payload
ISAKMP (0:1): processing HASH payload. message ID = -1279237069
ISAKMP (0:1): processing SA payload. message ID = -1279237069

```

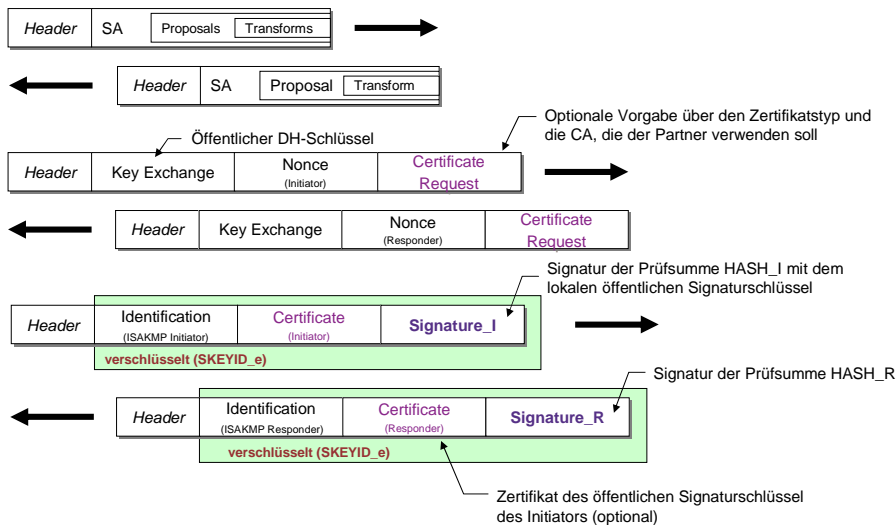
Trace - IPsec Protection Suite aushandeln

```

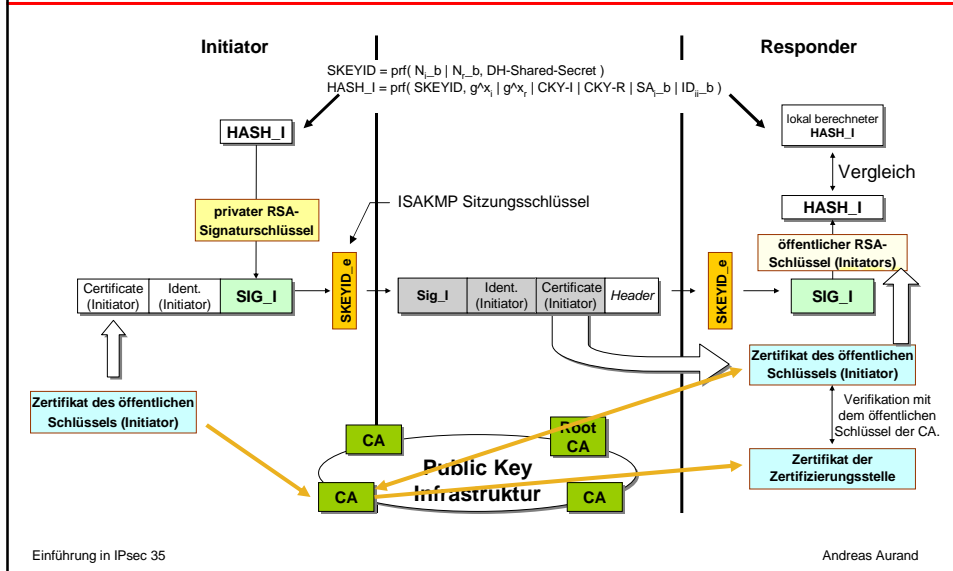
ISAKMP (0:1): Checking IPsec proposal 1
ISAKMP: transform 1, ESP_DES
ISAKMP: attributes in transform:
ISAKMP:   encaps is 1
ISAKMP:   SA life type in seconds
ISAKMP:   SA life duration (basic) of 3600
ISAKMP:   SA life type in kilobytes
ISAKMP:   SA life duration (VPI) of 0x0 0x46 0x50 0x0
ISAKMP (0:1): atts are acceptable.
ISAKMP (0:1): processing NONCE payload. message ID = -1279237069
ISAKMP (0:1): processing ID payload. message ID = -1279237069
ISAKMP (0:1): processing ID payload. message ID = -1279237069
ISAKMP (0:1): Creating IPsec SAs
      inbound SA from 16.204.7.91 to 16.204.7.97 (proxy 192.168.56.0 to 16.204.7.0)
      has spi 0x658DB0D9 and conn_id 2000 and flags 4
      lifetime of 3600 seconds
      lifetime of 4608000 kilobytes
      outbound SA from 16.204.7.97 to 16.204.7.91 (proxy 16.204.7.0 to 192.168.56.0)
      has spi 454151126 and conn_id 2001 and flags 4
      lifetime of 3600 seconds
      lifetime of 4608000 kilobytes

ISAKMP: Quick Mode packet contents (flags 1, len 52):
      HASH payload
ISAKMP (0:1): sending packet to 16.204.7.91 (I) QM_IDLE
    
```

Authentifizierung über RSA-Signaturen



Authentifizierung über RSA-Signaturen



Authentifizierung über RSA-Signaturen

- Überprüfung der Authentizität des Partners
 1. Sender **signiert** den Hashwert mit seinem **privaten** Signaturschlüssel
 2. Empfänger **entschlüsselt** Signatur mit **öffentlichem** Schlüssel des Partners
 3. Empfänger vergleicht lokal berechneten Hashwert mit Wert aus Signatur
- Partner tauschen Zertifikat direkt über ISAKMP-Nachricht aus
 - Zertifikat enthält öffentlichen Signaturschlüssel des Partner
 - Nicht zu verwechseln mit dem erzeugten öffentlichen DH-Schlüssel
- Verifikation des Zertifikats über dessen digitale Signatur
 - Partner benötigen dazu das Zertifikat (und den öffentlichen Schlüssel) der **Certificate Authority (CA)**
 - **Die Sicherheit der IPsec-Verbindung hängt zum großen Teil von der Sicherheit des CA-Zertifikats ab**
 - **Gültigkeit des CA-Zertifikats explizit überprüfen.**
 - Evtl. Überprüfung des Zertifikats gegen die **Widerrufsliste (CRL)** der CA

Trace - Austausch der Zertifikate

```

ISAKMP: Main Mode packet contents (flags 0, len 238):
  KE payload
  NONCE payload
  CERT-REQ payload
  VENDOR payload
ISAKMP (4): sending packet to 172.16.100.2 (I) MM_SA_SETUP

ISAKMP (4): received packet from 172.16.100.2 (I) MM_SA_SETUP
ISAKMP: Main Mode packet contents (flags 0, len 238):
  KE payload
  NONCE payload
  CERT-REQ payload
  VENDOR payload
ISAKMP (0:4): processing KE payload. message ID = 0
ISAKMP (0:4): processing NONCE payload. message ID = 0
CryptoEngine0: calculate pkey hmac for conn id 0
CryptoEngine0: create ISAKMP SKEYID for conn id 4
ISAKMP (0:4): SKEYID state generated
ISAKMP (4): processing CERT_REQ payload. message ID = 0
ISAKMP (4): peer wants a CT_X509_SIGNATURE cert
ISAKMP (4): peer want cert issued by CN = CA FRS-LAB, OU = Andreas Aurand, C = DE
ISAKMP (0:4): processing vendor id payload
ISAKMP (0:4): speaking to another IOS box!

```

Trace - Authentifizierung des Partners

```

ISAKMP (4): ID payload
  next-payload : 6
  type         : 2
  protocol     : 17
  port         : 500
  length       : 20
ISAKMP (4): Total payload length: 24
CryptoEngine0: generate hmac context for conn id 4
Crypto engine 0: RSA encrypt with private key
CryptoEngine0: CRYPTO_RSA_PRIV_ENCRYPT
ISAKMP: Main Mode packet contents (flags 1, len 858):
  ID payload
  CERT payload
  SIG payload
ISAKMP (4): sending packet to 172.16.100.2 (I) MM_KEY_EXCH

ISAKMP (4): received packet from 172.16.100.2 (I) MM_KEY_EXCH
ISAKMP: Main Mode packet contents (flags 1, len 860):
  ID payload
  CERT payload
  SIG payload
ISAKMP (0:4): processing ID payload. message ID = 0
ISAKMP (0:4): processing CERT payload. message ID = 0
ISAKMP (0:4): processing a CT_X509_SIGNATURE cert
CRYPTO_ENGINE: key process suspended and continued
CRYPTO_PKI: Certificate verified, chain status= 1
ISAKMP (0:4): processing SIG payload. message ID = 0
ISAKMP (4): sa->peer.name = , sa->peer_id.id.id.fqdn.fqdn = c2504.frs-lab.de
Crypto engine 0: RSA decrypt with public key
CryptoEngine0: CRYPTO_RSA_PUB_DECRYPT
CryptoEngine0: generate hmac context for conn id 4
ISAKMP (0:4): SA has been authenticated with 172.16.100.2

```

Signatur wird mit dem privaten RSA-Signaturschlüssel chiffriert

Signatur wird mit dem öffentlichen RSA-Signaturschlüssel des Partners dechiffriert

Beispiel für ein X.509v3-Zertifikat

```

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number:
      04:26:fb:f3:00:00:00:00:00:10
    Signature Algorithm: sha1WithRSAEncryption
    Issuer: C=DE, OU=Andreas Aurand, CN=CA FRS-LAB
    Validity
      Not Before: Feb 13 10:10:50 2001 GMT
      Not After : Feb 13 10:20:50 2002 GMT
    Subject: SN=7681045/unstructuredName=c2504.frs-lab.de
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (512 bit)
      Modulus (512 bit):
        00:cc:bc:f5:0d:bb:3b:3a:d5:2f:9d:3b:ce:09:82:
        15:7e:ae:5a:d2:19:d0:12:4c:6a:cf:c7:c3:e2:65:
        cc:12:d2:ee:f2:5d:df:31:fc:4f:70:38:77:96:54:
        37:a2:3b:dc:4e:df:7c:ea:ff:75:0c:93:43:7c:29:
        24:8d:e7:9a:4b
      Exponent: 65537 (0x10001)
    Signature Algorithm: sha1WithRSAEncryption
      39:23:9d:21:5f:5e:90:23:e4:72:6f:a8:0c:af:b2:56:1c:bc:
      b6:28:aa:d4:a9:57:3a:52:86:3b:9f:7a:3f:c1:de:da:c0:24:
      45:3b:73:69:88:38:3f:9d:05:8e:37:67:a0:50:a7:33:44:a9:
      a9:3b:81:7b:4f:49:0e:7b:3d:49
  
```

Öffentlicher RSA-Schlüssel

Digitale Signatur des Zertifikats. Wurde mit dem privaten Schlüssel der CA erzeugt. (DER-kodierter Bit String)

Agenda

- IPsec Architektur
 - Security Associations
 - Sicherheitsprotokolle
 - AH
 - ESP
 - IPComp
 - Schlüsselaustauschverfahren
 - ISAKMP und IKE
 - Diffie-Hellman Verfahren
 - Authentifizierung der Partner
- IPsec und NAT

IPsec und NAT - Problem



- **ISAKMP**
 - UDP Source und Destination Port 500
 - NAT möglich; PAT führt in der Regel zu Problemen
 - ISAKMP-Paket enthält lokale IP-Adresse als *Identity*
 - In der Regel keine Überprüfung der Source Adresse mit dem Identifier
- **AH (IP Protokoll 51)**
 - Prüfsumme über das komplette Paket, inklusive IP Header
 - Kein NAT oder PAT möglich
- **ESP (IP Protokoll 50)**
 - Keine Prüfsumme über den IP Header
 - NAT im Tunnel Mode möglich; evtl. auch PAT, liegt am NAT Device
 - Das NAT Device kann die lokale Source Adressen nicht mehr ändern, da verschlüsselt
 - Im Transport Mode stimmt bei TCP-Verbindungen die TCP Checksum nicht mehr

IPsec und NAT - Lösung

- **Cisco NAT Transparency**
 - ISAKMP und ESP Pakete werden in TCP oder UDP Pakete eingepackt
 - TCP Port Nummer 10000; kann verändert werden
 - UDP Port Nummer wird zwischen den Partnern abgestimmt
 - Cisco VPN Client, Cisco VPN Concentrator, PIX V6.2, IOS V12.2(10)T
- **IETF Draft Standards**
 - IPsec-NAT Compatibility Requirements
 - Negotiation of NAT-Traversal in the IKE
 - UDP Encapsulation of IPsec Packets
 - IPsec over NAT Justification for UDP Encapsulation
- Zusätzliche IPsec-Optionen werden beim Aufbau der ISAKMP SA über das **Vendor ID Feld** ausgehandelt.

Links

- **Cisco Security Whitepaper**
 - <http://www.cisco.com/go/safe>
- **IETF IPsec Working Group**
 - <http://www.ietf.org/html.charters/ipsec-charter.html>
- **IETF Public-Key Infrastructure (X.509) Working Group**
 - <http://www.ietf.org/html.charters/pkix-charter.html>

Fragen ?

