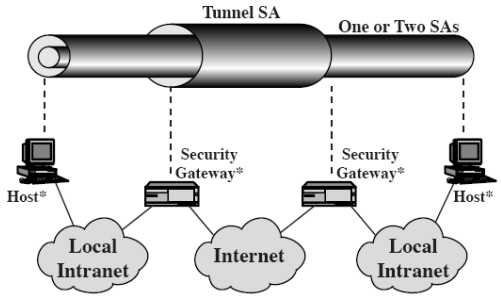
Combineren van SA's (1)

- één SA = één protocol in één modus in één richting
- soms combinatie noodzakelijk
  - AH en ESP alleen of gecombineerd in twee modi
  - beveiligde trafiek op intranet extra beveiligd op internet
- → combinatie hebben eventueel verschillende eindpunten
- van vele mogelijke combinaties zijn slechts enkele zinvol



61

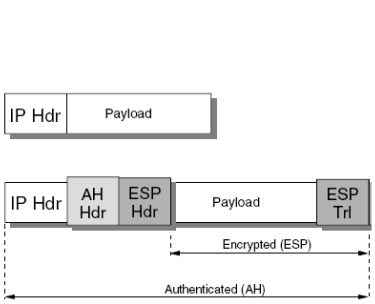
Combineren van SA's (2)

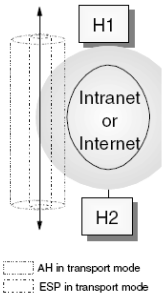
- combinaties worden samengevoegd in SA-bundels
- SA-bundel
  - sequentie van SA's die moeten gebruikt worden na elkaar om gewenst IPSec effect te verkrijgen
  - SA's mogen eindigen in dezelfde of in verschillende eindpunten
- twee benaderingen:
  - *transport adjacency* (nabijheid)
  - *iterated tunneling* (genest)

zie verder …

SA's in transport adjacency (1)

- verschillende protocollen toegepast op zelfde IP-pakket
- "mag uitsluitend gebruikt worden bij authenticatie van geëncrypteerde gegevens" (dixit IPSec-standaard)
- AH na ESP (ESP zonder authenticatie!)
  toegepast in transportmode (geen tunneling)
- meer dan 2 protocols toepassen heeft geen zin (1 pakket)





63

SA's in transport adjacency (2)

- waarom AH en niet ESP alleen?
  - authenticering van AH beslaat meer velden dan die van ESP
- nadeel: meer werk
- waarom niet ESP ná AH ?→ antw 1
  - eerst decryptie bij ontvangst, dan pas authenticatie (meer werk)
- waarom niet ESP ná AH? → antw 2
  - IP-header bevat protocol van IP-payload
    - http://www.iana.org/assignments/protocol-numbers
  - AH authenticeert ook IP-header, incl protocol
  - indien nadien nog ESP toegepast:
    - AH + payload wordt omgeven door ESP-headers
    - IP-protocol moet veranderd worden
    - gegevens voor authenticatie kloppen niet meer

IPSec 64

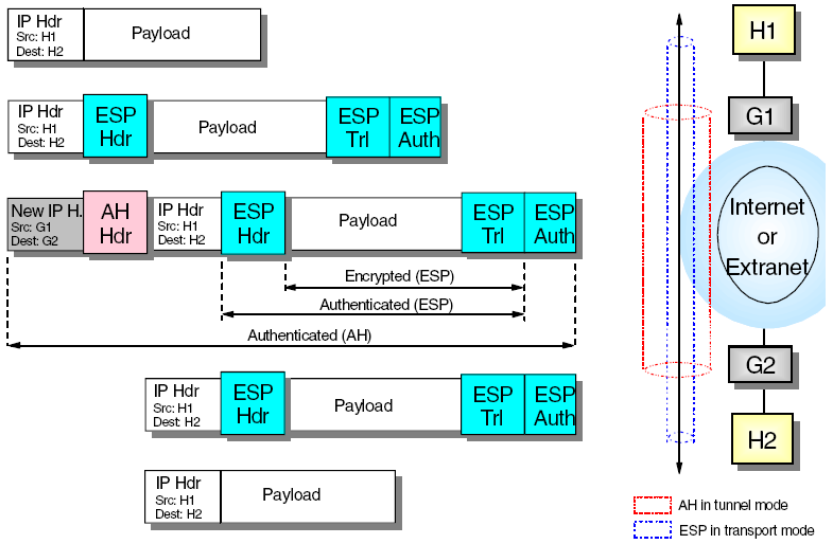
Toch ESP ná AH?

- soms authenticatie vóór encryptie gewenst
  - gegevens van authenticatie kunnen niet gewijzigd worden
  - gegevens van authenticatie kunnen bij boodschap worden opgeslagen
- niet mogelijk bij transport adjacency
- wel bij iterated tunneling

SA's in iterated tunneling

- beide protocollen worden ná elkaar toegepast
  vaak eerst in transportmode, dan in tunnelmode
- na elke toepassing, creatie van nieuw IP-pakket met nieuwe IP-header
- nieuw IP-pakket wordt aangeboden aan volgend protocol
- aantal vernestelde protocols onbeperkt
  (in praktijk meestal 2, maximaal 3, meer onnuttig)
- meestal telkens verschillende eindpunten
- volgende slide: typisch voorbeeld

Voorbeeld iterated tunneling

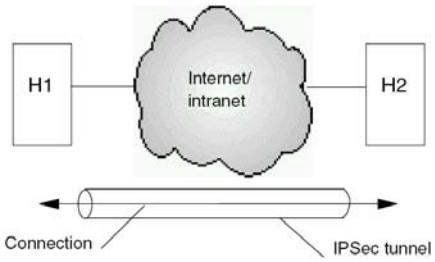


Opmerkingen combinatie SA's

- combinatie van beide benaderingen is mogelijk
- toegepast voor VPN's
- bijkomend protocol moet toegevoegde waarde hebben
- indien SA's met verschillende eindpunten, minstens 1 niveau van tunneling (niet mogelijk met transport)
- bij ontvangst eerst authenticatie dan encryptie behandelen (reden: liever geen foutieve pakketten decrypteren)
  - ➔ bij verzenden dus eerst ESP dan AH
- gebruiken we ESP met authentisering (zie vorige slide)? alleen zinvol als ESP-SA verder reikt dan AH-SA
  - vb.: ESP end-to-end en AH slechts tot gateway
  - vermijdt spoofing-aanval op intranet (is dit wel correct???)
- 4 praktijkvoorbeelden

IPSec 68

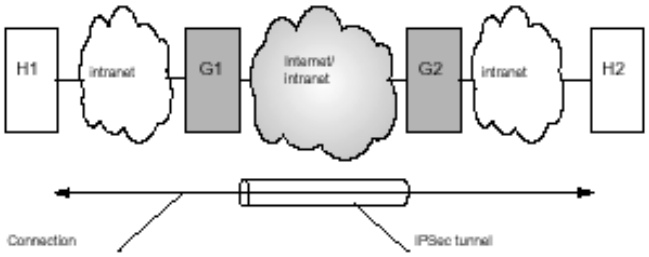
1. End-to-End beveiliging



- 2 host, geen gateways, verbonden via internet
- kunnen beveiligen via ESP, AH of beide
- mogelijke combinaties:
  - transportmode:
    - **AH alleen, ESP alleen, AH ná ESP (transport adjacency)**
  - tunnelmode
    - **AH alleen, ESP alleen**
    - **AH of ESP na transportmode (laat encryptie van de authenticatie toe)**

IPSec 69

2. Basis VPN opbouw



- hosts H1 en H2 voorzien geen IPSec
- gateways G1 en G2 voorzien IPSec in tunnel-mode, AH en/of ESP
- vermits zelfde eindpunten ➔ iterated tunneling heeft geen zin (vb. tunnel ESP, nadien tunnel AH)
- wel bv ESP in tunnel-mode en nadien AH in transport-mode (gecombineerde AH-ESP-tunnel; zie volgende slide)

IPSec 70

Vb. ESP tunnel + AH transport

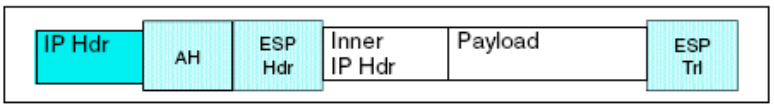
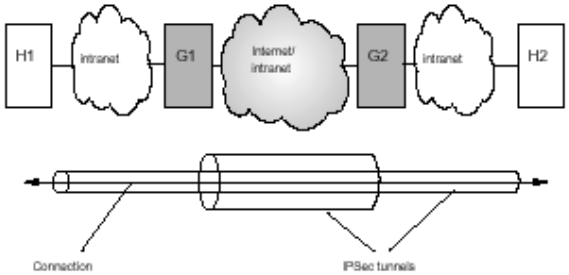


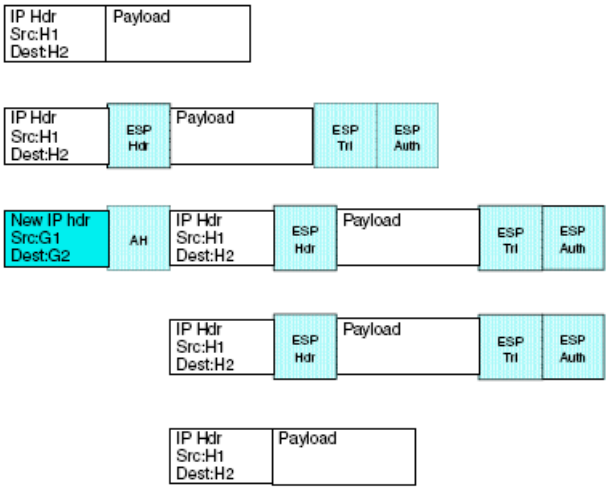
Figure 293. Combined AH-ESP tunnel

3. End-to-End beveiliging + VPN

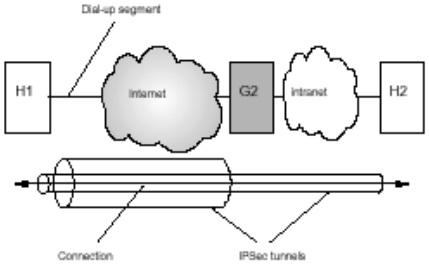


- hosts H1 en H2 voorzien nu wél IPSec
- typische opbouw: (zie figuur volgende dia)
  - G's: AH in tunnelmode
  - H's: ESP in transportmode
- beter: tussen G's AH-ESP tunnel zodat eindadressen niet kunnen worden achterhaald (bundel van 3 SA's → kostelijk)

AH tunnel + ESP transport



4. Remote Access



- host H1 is verbonden met internet via ISP
- tunnelmode tussen H1 en G2 (keuzes zie geval 2)
- tussen H1 en H2: tunnel of transport (zie geval 1)
- typische opbouw (2 gevallen):
  - H1-H2: ESP in transport-mode H1-G2: AH in tunnel-mode
  - H1-H2: ESP in transport-mode H1-G2: AH-ESP tunnel

Sleutelbeheer van IPSec

- Doel:
  - opzetten van SA's
  - bepaling, verdeling en refreshing van sleutels
- kan manueel gebeuren
  - gedaan door systeemadministrator
  - alleen mogelijk voor kleine en statische omgevingen
- gebeurt beter automatisch
- Protocollen (*what's in a name?*):
  - IKE
  - ISAKMP/Oakley
  - SKEME
  - …

Sleutelbeheer van IPSec (1)

- ISAKMP (Internet Security Association and Key Management Protocol, RFC 2408)
  - raamwerk voor beheer van SA's en sleutels
    - onderhandelen, wijzigen, vernietigen
  - bepaalt formaat payload voor uitwisseling van gegevens nodig voor generatie van sleutels en authenticatie
  - bepaalt *niet* sleuteluitwisselingsprotocol
  - gebruikt hiertoe elementen van Oakley en SKEME
- Oakley (RFC 2412)
  - sleuteluitwisselingsprotocol gebaseerd op Diffie-Hellman
  - generisch (geen opgelegde formaten of informatie-uitwisseling)
  - voegt authenticatie toe van de 2 partijen (methode = SKEME)
  - laat verschillende manieren toe ➔ ≠ modes in IKE/ISAKMP

Sleutelbeheer van IPSec (2)

- SKEME, A Versatile Secure Key Exchange Mechanism for Internet
  - artikel van Hugo Krawczyky (IEEE, november 1995)
  - levert o.a. methode van authenticatie:
    - elke peer encrypteert random getal met publieke sleutel
    - plaintext waarden bepalen mee sleutel
- IKE (Internet Key Exchange: RFC 2409)
  - eigenlijke implementatie voor beheer van SA's en sleutels
  - gebruikt ISAKMP, Oakley en SKEME
  - generisch protocol, ook gebruikt in routing protocollen (RIP, OSPF)
- DOI (Domain of Intepretation, RFC 2407)
  - bepaalt hoe IKE moet gebruikt worden om SA's op te zetten voor IPSEC

Sleutelbeheer van IPSec (3)

- Implementaties
  - Photuris (OpenBSD IPSec)
  - SKIP
  - Pluto (FreeS/WAN - OpenS/WAN)
- IKE werkt over UDP, poort 500

IKE overzicht (1)

IPSec-vereisten voor sleutelbeheer ➔ kenmerken IKE:

- alle uitgewisselde info geheim en geauthenticeerd
- gebeurt initieel onbeveiligd
- uitsluitend uitwisseling tussen **geauthenticeerde partijen**
- weerstaat aanvallen
  - denial-of-service
    - unieke cookies (cf. nonces) in boodschappen
    - laten toe snel boodschappen te weigeren
  - man-in-the-middle = boodschappen
    - achterhouden
    - wijzigen
    - terugsturen
    - replay
    - omleiden naar andere ontvangers

IKE overzicht (2)

- voorziet in PFS (perfect forward secrecy)
  dwz. gekraakte sleutel
  - geeft geen bijkomende info over vroegere of nieuwe sleutels
  - kan alleen nuttig gebruikt worden bij gegevens geëncrypteerd met die sleutel
  - heeft geen invloed op andere sleutels en SA's
  - → nieuwe sleutel is totaal onafhankelijk van vorige
  - opgelet: is een optie die dus kan uitgeschakeld worden
- IKE gebruikt 2 fasen
  - eerste bepaalt SA voor sleuteluitwisseling
  - tweede stelt SA's op voor AH of ESP gebruik
- eerste fase zorgt voor authenticatie
  - op basis van "certificaten" (≠ X.509)
  - is IP-adres of user-id/wachtwoord

IKE overzicht (3): 3 schema's

Authenticatie: methode	Authenticatie: hoe?	Voordelen	Nadelen
vooraf gedeelde sleutels Pre-shared Key	berekenen van (H)MAC over uitgewisselde informatie	eenvoudig	• gedeelde sleutel moet vooraf verdeeld zijn (token) • alleen IP-adres als ID
DSS of RSA handtekeningen	handtekenen van berekende hash over uitgewisselde informatie	• andere ID's dan IP-adres • partnercertificering niet noodzakelijk voor onderhandelingen	• vereist behandeling van certificaten
publieke sleutel encryptie (RSA)	hash van nonce geëncrypteerd met private sleutel	• betere beveiliging bovenop DH uitwisseling • ID beveiligd	• publieke sleutels of certificaten vooraf beschikbaar • intensieve berekeningen

IKE overzicht (4)

Twee fasen:

- **fase 1:** opzetten van ISAKMP-SA
  - verwezenlijken van een *master secret* waarvan alle andere sleutels voor de SA's zullen worden afgeleid
  - gebruikt om de ISAKMP-boodschappen te beschermen uit fase 2
    - bepalen van cryptografische protocollen en parameters
    - die zorgen voor authenticatie en encryptie
  - gebeurt 1x per dag of week
- **fase 2:** opzetten van IPSec-SA's
  - stelt SA's op, beveiligd door realisaties van fase 1
  - minder complexe uitwisselingen
  - refresh van sleutels bv. om 2 of 3 minuten

IKE overzicht (5)

- voorziet in gebruik van *Permanent Identifiers*
- oplossing als IP-adres van host vooraf niet gekend is
- permanente identificatie op basis van
  - naam *of*
  - e-mailadres
- gebruik van “certificaten” met link tussen publieke sleutel en permanente identificatie
  - niet noodzakelijk X.509 certificaten
- fase1-boodschappen bevatten dergelijke certificaten
- er kan link gelegd worden met tijdelijk IP-adres van host
  - boodschap maakt immers deel uit van IP-datagram)

IKE fase 1: 3 modes

- 3 modes: main (verplicht), aggressive en base (optioneel)
- Main mode
  - 6 UDP-datagrams worden uitgewisseld
    - **A doet reeks voorstellen aan B, B kiest een voorstel**
    - **sleutelinformatie + nonces worden uitgewisseld**
    - **ID’s v partijen worden uitgewisseld en geauthenticeerd (beveiligd)**
  - eerste uitwisseling gebeurt op basis van IP-adressen
- Aggressive mode (=Quick mode)
  - uitwisseling van slechts 3 pakketten
  - eerste bericht bevat reeds ID’s en DH publieke waarden
  - identiteit niet alleen op basis van IP-adres
  - identiteit is uitgewisseld vóór het bepalen van een gedeelde sleutel
  - kan gebruikt indien initiator A policy kent van B (inbellen nr bedrijf)

IKE fase 1: 3 modes (2)

- Aggressive mode
  - alleen maar voordelen? neen
  - kwetsbaar voor denial of service aanval
    - sturen van veel voorstellen met *spoofed* IP-adressen
    - vergt veel berekeningen voor de ontvanger
  - niet meer aangeraden door de IPSec-werkgroep
- Base mode: 4 datagrammen
  - doel
    - voordelen van aggressive mode
    - nadelen ervan wegwerken
  - er wordt gewacht op 2de datagram van initiator, bewijst zijn bestaan
  - niet alle problemen zijn zo opgelost
    - identiteiten niet beschermd
    - sleutelinformatie slechts uitgewisseld na controle van identiteiten

IKE: modes (1)

Exchange	Note
(a) Base Exchange	
(1) **I → R:** SA; NONCE	Begin ISAKMP-SA negotiation
(2) **R → I:** SA; NONCE	Basic SA agreed upon
(3) **I → R:** KE; ID_I; AUTH	Key generated; Initiator identity verified by responder
(4) **R → I:** KE; ID_R; AUTH	Responder identity verified by initiator; Key generated; SA established
(b) Identity Protection Exchange	
(1) **I → R:** SA	Begin ISAKMP-SA negotiation
(2) **R → I:** SA	Basic SA agreed upon
(3) **I → R:** KE; NONCE	Key generated
(4) **R → I:** KE; NONCE	Key generated
(5)* **I → R:** ID_I; AUTH	Initiator identity verified by responder
(6)* **R → I:** ID_R; AUTH	Responder identity verified by initiator; SA established
(c) Authentication Only Exchange	
(1) **I → R:** SA; NONCE	Begin ISAKMP-SA negotiation
(2) **R → I:** SA; NONCE; ID_R; AUTH	Basic SA agreed upon; Responder identity verified by initiator
(3) **I → R:** ID_I; AUTH	Initiator identity verified by responder; SA established

IKE: modes (2)

(d) Aggressive Exchange	
(1) **I → R:** SA; KE; NONCE; ID_I	Begin ISAKMP-SA negotiation and key exchange
(2) **R → I:** SA; KE; NONCE; ID_R; AUTH	Initiator identity verified by responder; Key generated; Basic SA agreed upon
(3)* **I → R:** AUTH	Responder identity verified by initiator; SA established
(e) Informational Exchange	
(1)* **I → R:** N/D	Error or status notification, or deletion

Notation:
I = initiator
R = responder
* = signifies payload encryption after the ISAKMP header
AUTH = authentication mechanism used

IPSec 87

ISAKMP-boodschap

IP header	UDP header	ISAKMP header	ISAKMP payload	ISAKMP payload	ISAKMP payload	…

- verpakt in UDP-pakket
- header gevolgd door 1 of meerdere payloads
- elke payload bevat een generische header met
  - Next Header of Next Payload
    - = type van volgende payload
  - payload lengte

IPSec 88

ISAKMP-header

- ISAKMP-header
  bevat o.a.
  - Initiator cookie (64 bit pseudorandom getal)
  - Responder cookie (door A op 0 gezet)
  - Exchange type
    - 5 types met verschillend aantal uitgewisselde boodschappen → modes
    - main mode = 2
  - Next Payload
    - geeft soort payload aan direct volgend op header
    - http://www.iana.org/assignments/isakmp-registry
  - Message ID (hier = 0)
  - Flags
  - Length of message

OAKLEY cookie

- pseudorandom getal
  - gegenereerd door elke partij, geëchood in het antwoord
  - door beide partijen te verifiëren
  - 64 bits
- gebruik vermijdt *clogging* of *DOS*-aanvallen
  - systeem wordt overbelast door DH-sleutelberekeningen
  - gebeurt met vervalst bronadres
- door vervalst IP-adres krijgt aanvaller antwoord-cookie niet te zien
- aanvaller kan dus ook de andere peer niet verplichten DH-waarden te berekenen
  - gebeurt pas nadat cookies werden geverifieerd

OAKLEY cookie (2)

- eisen voor een cookie
  - afhankelijk van de twee partijen
    → vermijdt gebruik van willekeurige IP-adressen en poorten
  - generatie en verificatie gebeurt snel (DOS-aanvallen)
  - uitwisselende entiteiten gebruiken ook eigen geheime informatie
    - **cookie moet niet in geheugen bewaard worden**
    - **elke partij kan eigen cookie verifiëren**
- voorbeeld van methode
  - snelle hash (vb. MD5) over
    - **IP-adressen van bron en bestemming**
    - **UDP-poorten van bron en bestemming**
    - **lokaal gegenereerde geheime waarde (nonce)**

IPSec 91

ISAKMP-boodschap

IP header	UDP header	ISAKMP header	ISAKMP payload	ISAKMP payload	ISAKMP payload	…

- verpakt in UDP-pakket
- header gevolgd door 1 of meerdere payloads
- elke payload bevat een generische header met
  - Next Header of Next Payload
    - **= type van volgende payload**
  - payload lengte

IPSec 92

ISAKMP soorten payload

Type	Parameters	Description
Security Association (SA)	Domain of Interpretation, Situation	Used to negotiate security attributes and indicate the DOI and Situation under which negotiation is taking place.
Proposal (P)	Proposal #, Protocol-ID, SPI Size, # of Transforms, SPI	Used during SA negotiation; indicates protocol to be used and number of transforms.
Transform (T)	Transform #, Transform-ID, SA Attributes	Used during SA negotiation; indicates transform and related SA attributes.
Key Exchange (KE)	Key Exchange Data	Supports a variety of key exchange techniques.
Identification (ID)	ID Type, ID Data	Used to exchange identification information.
Certificate (CERT)	Cert Encoding, Certificate Data	Used to transport certificates and other certificate-related information.
Certificate Request (CR)	# Cert Types, Certificate Types, # Cert Auths, Certificate Authorities	Used to request certificates; indicates the types of certificates requested and the acceptable certificate authorities.
Hash (HASH)	Hash Data	Contains data generated by a hash function.
Signature (SIG)	Signature Data	Contains data generated by a digital signa wture function.
Nonce (NONCE)	Nonce Data	Contains a nonce.
Notification (N)	DOI, Protocol-ID, SPI Size, Notify Message Type, SPI, Notification Data	Used to transmit notification data, such as an error condition.
Delete (D)	DOI, Protocol-ID, SPI Size, # of SPIs, SPI (one or more)	Indicates an SA that is no longer valid.

IKE fase 1

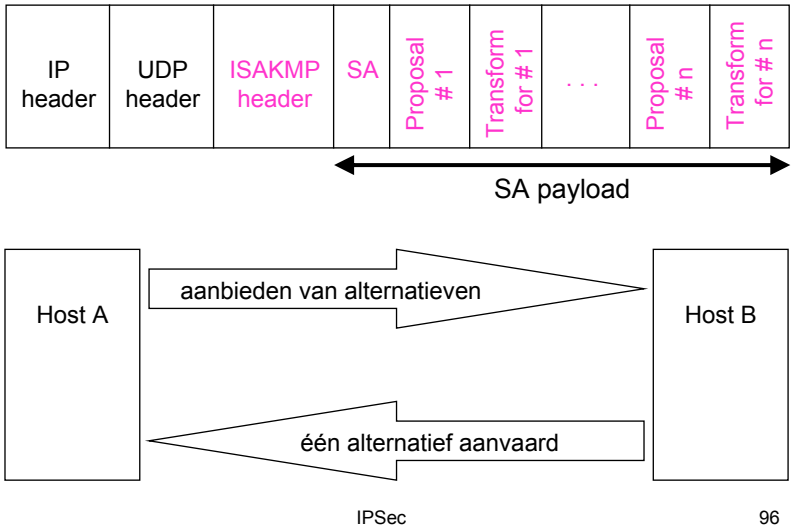
- koude start, geen vroegere sleuteluitwisseling tussen host A en host B
- gebruikt Oakley Main mode (ISAKMP Identity Protect exchange)
- 6 boodschappen:
  - 1&2
    - bepalen eigenschappen van SA's voor ISAKMP
    - onbeveiligd, niet geauthenticeerd
  - 3&4
    - uitwisseling nonces en opstellen master key via Diffie-Hellman
    - onbeveiligd, niet geauthenticeerd
  - 5&6
    - authenticatie van identiteit van beide partijen
    - beveiligd met materiaal bekomen uit 1-4

IPSec 94

IKE fase 1 / boodschap 1

- A stuurt een reeks keuzemogelijkheden naar B in PT
  - Vb.:
    {3DES, MD5}, {BLF, SHA-1, DH-G1}, {CAST, MD5, DH-G1},
    {3DES, SHA-1,DH-G2}
- elk voorstel bevat algoritmen voor
  - conventionele encryptie, berekening van hash, sleuteluitwisseling
  - laatste niet indien pre-shared-key is gebruikt (voorstel 1)
- B kiest vb. {3DES, SHA-1,DH-G2}
- DH-Gx
  - verwijst naar de groepen van DH-algoritme
  - elke groep heeft ander aantal bits voor modulus-priemgetal
- informatie zit in *transform payload* van de boodschap

IKE fase 1 / boodschap 1



IKE fase 1 (2)

- **Boodschap 1**
  - A zendt naar B cleartext als payload van UDP-pakket
  - source en destination IP adressen zijn die van A en B
  - destination port 500 in UDP-header
  - ISAKMP boodschap bevat 1 of meer voorstellen door B in overweging te nemen
    - ISAKMP-header
      (bevat o.a. 2 cookies als id van SA, zie volgende slide)
    - Security Association veld:
      - identificeert implementatiedomein
      - bedoeling is IPSec-uitwisseling ➔ dus DOI is IP
    - Proposal payload (zie verder)
    - Transform payload (zie verder)

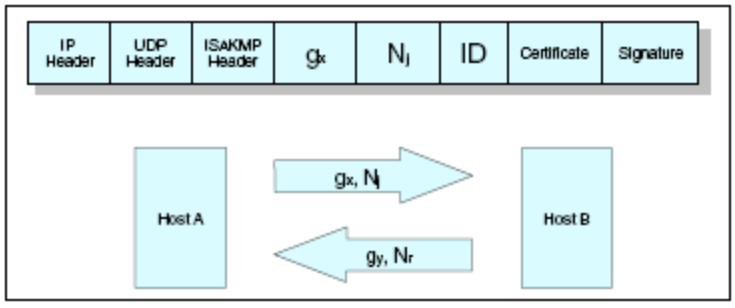
ISAKMP-boodschap (2)

- Proposal payload
  - bevat info over te onderhandelen SA
  - type protocol = PROTO_ISAKMP (hier, kan ook ESP of AH zijn)
  - SPI van responder = 0 (kan nu nog niet bepaald worden)
  - aantal transforms (elke transform in 1 transform-payload)
- Transform payload
  - definieert algemeen de transformatie voor de beveiliging
  - vbn: 3DES voor ESP, SHA1 voor AH
  - hier KEY_OAKLEY
  - relevante attributen
    - methoden voor authenticatie
    - functie om randomgetallen te bepalen
    - encryptiemethode

IKE fase 1: (2)

- Boodschap 2
  - antwoord van B op boodschap 1: B kiest 1 alternatief
  - responder cookie wordt ingevuld
  - source en destination IP adressen zijn die van B en A
  - cookie-A en cookie-B dienen als SPI voor de ISAKMP-SA
  - rest is zelfde: 1 proposal met 1 transform payload
- Opmerkingen:
  - SPI van ISAKMP-SA bepaald
  - SPI evenwel nog niet ingevuld omdat identiteiten van partijen die beweren A en B te zijn, is nog niet geverifieerd
  - eigenschappen (transforms) van ISAKMP-SA zijn vastgelegd
  - nu begint uitwisseling om sleutels te bepalen

IKE fase 1: (3)



- **Boodschap 3** van A naar B in ISAKMP-payload
  - Diffie-Hellman publieke waarde a^x (x is privaat voor A en geheim) (in Key Exchange payload); g_x in figuur
  - nonce N_i (initiator) (in Nonce payload)
  - extra payloads indien authenticatie met certificaat en handtekening (ID, Certificate, Signature) (N_i en ID geëncrypteerd met RSA-pubkey)

IKE fase 1 (4)

- Boodschap 4
  - van B naar A
  - Diffie-Hellman publieke waarde a^y (y is privaat voor B en geheim) (in Key Exchange payload)
  - nonce N_r (responder) (in Nonce payload)
- A & B genereren sleutels
- A en B kennen elk:
  - N_i en N_r
  - x resp. y (private DH-waarde)
  - a^x en a^y (publieke DH-waarde)
  - kunnen a^{xy} berekenen
  - initiator en responder cookies

zie ook volgende dia

IPSec 101

IKE fase 1 (5)

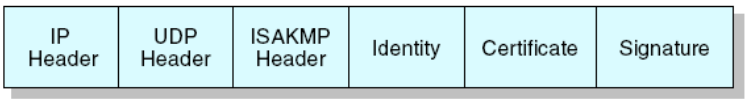
A & B genereren elk hetzelfde stel sleutels

- gebeurt a.d.h.v. **onderhandelde** pseudorandomfctie (**prf**) vb HMAC
- delen door komma gescheiden ≡ concatenatie
- **SKEYID** afhankelijk van toepassing
  - **authenticatie met digitale handtekening: prf(Ni, Nr, a^{xy})**
  - **authenticatie publieke sleutels: prf(hash(Ni,Nr),cookieA, cookieB)**
  - **authenticatie met pre-shared key: prf(presharedkey, Ni, Nr)**
- **SKEYID_x** afhankelijk van toepassing
  - **gebruikt in fase2 om sleutels voor "echte" SA's te bepalen**
    SKEYID_d = prf(SKEYID,a^{xy},cookieA, cookieB,0)
  - **authenticatie van ISAKMP boodschappen**
    SKEYID_a = prf(SKEYID, SKEYID_d,a^{xy},cookieA,cookieB,1)
  - **encrypteren van ISAKMP boodschappen**
    SKEYID_e = prf(SKEYID, SKEYID_a, a^{xy}, cookieA, cookieB,2)

IPSec 102

IKE fase 1 (6)

- A en B hebben identieke sleutels afgeleid
- dienen voor authenticatie en encryptie
- beveiligen de verdere ISAKMP-uitwisselingen
  - in fase 1 stap 5 en 6
  - in fase 2 (=opzetten van SA voor AH of ESP)
  - alleen ISAKMP-payload geëncrypteerd
- tot nu toe zijn identiteiten van beide partijen nog niet geauthenticeerd ten overstaan van mekaar
- gebeurt in boodschap 5:
  - A → B ➔ B authenticeert A



103

IKE fase 1: boodschap 5

Uitwisseling ter authentisering

- **Boodschap 5** van A naar B
  - bevat *Identity*, *Certificate* en *Signature payload*
  - identiteit ID_A
    - **id van certificaat, niet noodzakelijk in relatie tot IP-adres**
  - certificaat:
    - **als aanwezig → B verifieert handtekening CA**
    - **als niet aanwezig → B vraagt dit aan CA, LDAP, secure-DNS, …**
  - handtekening: (SAp is volledige SA ***payload*** uit boodschap1)
    niet berekend op payload van bericht, maar op de gemeenschappelijke gegevens

    HASH_I = prf(SKEYID, a^x, a^y, cookieA, cookieB, SAp , ID_A)
    eventueel getekend door A; te valideren door B

IPSec 104

IKE fase 1: boodschap 6

- **Boodschap 6** van B naar A
  - analoog als boodschap 5, andere volgorde parameters
- Opmerkingen:
  - steeds 6 boodschappen, maar inhoud afhankelijk van methode van authenticatie
  - ISAKMP-payload in boodschappen 5 en 6 gecodeerd met
    - algoritmen uit 1 & 2
    - sleutels uit 3 & 4

    dus geen gebruik van ESP of AH in dit stadium, maar beveiliging op applicatieniveau (ISAKMP payload)
  - Message-id (in ISAKMP-hoofding)

    = 0 in fase1

    enige manier om te weten of het gaat om fase1 of fase2
  - ISAKMP boodschap via UDP → geen garantie op ontvangst

IKE fase 2 (Oakley Quick Mode)

- Oakley Quick Mode = ISAKMP Aggressive Mode
- host **A start onderhandelingsproces** om SA's te bepalen
- **ISAKMP-payload** (niet header) wordt **geëncrypteerd** met onderhandelde methoden uit fase1
- authenticatie d.m.v.
  - hash-functies op materiaal uitgewisseld in fase1 (SKEYID) en fase2
  - fase2 authenticatie is gebaseerd op certificaten op een onrechtstreekse manier :

    gebruikt SKEYID_a uit fase1 dat gecertificeerd is dmv certificaten

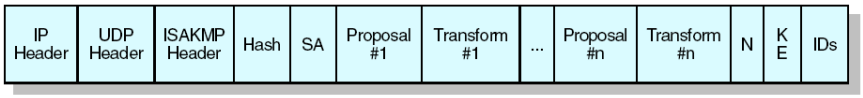
IKE fase 2 (2)

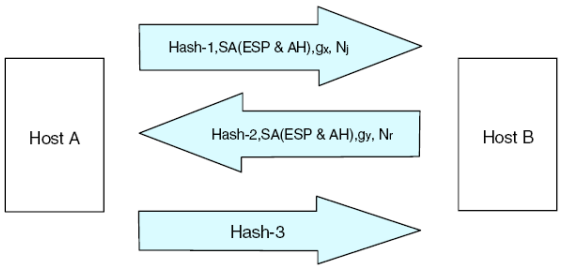
- gebruik van **Oakley Quick mode** in twee modes
  - *zonder* Key Exchange attribuut
    om sleutels te refreshen *zonder* PFS (Perfect Forward Secrecy)
  - *met* Key Exchange attribuut
    om sleutels te refreshen *met* PFS (extra Diffie-Hellman publieke sleuteluitwisseling in boodschappen 1&2)
  - PFS is dus optioneel (raar maar waar)
- **3 boodschappen**

IPSec 107

IKE fase 2

- **3 boodschappen**





IPSec 108

IKE fase 2 (3)

- **Boodschap 1** van A naar B
  - authenticeert A t.o.v. B
  - bevat een nonce
  - stelt meerdere SA's voor aan B
  - bevat A's publieke Diffie-Hellmann waarde
  - geeft aan of A op eigen houtje handelt of als proxy-onderhandelaar voor andere entiteit
  - ISAKMP boodschap bevat 1 of meer voorstellen
    - ISAKMP-header
    - Hash payload
    - Security Association (hier implementatiedomein, dus IP)
    - Proposal + Transform payloads (meerdere)
    - Nonce
    - Key Exchange payload (indien Perfect Forward Secrecy)
    - ID's (eindpunten van deze SA)

IKE fase 2 (4)

- **ISAKMP-header** bevat o.a.
  - Initiator en responder cookies uit fase 1
  - Exchange type (hier Quick mode=4)
  - Message ID gekozen door A (hier ≠ 0)
  - Flags (encryptievlag aan → payload geëncrypteerd)
- **Hash**: HASH_1 = prf(SKEYID_a, M-ID, SA, Ni , KE, IDi , IDr)
- **SA** is de volledige payload van deze boodschap, incl. alle voorstellen
- **KE** is nieuwe gekozen publieke Diffie-Hellman waarde
- Nonce payload nieuw gekozen: Ni
- Proposal+Transform paren
  - genummerd
  - SPI random gekozen door A (wordt toch bepaald door B)
  - Transform bevat protocol (ESP of AH) en algoritme

IKE fase 2 (5)

- **Boodschap 2** van B naar A
  - wordt verstuurd nadat boodschap 1 succesvol werd geauthenticeerd dmv hash_1
  - message-id is zelfde als gekozen door A
  - volgende waarden worden door B aangepast:
    - Hash

      HASH_2 = prf(SKEYID_a, Ni , M-ID, SA, Nr , KE, IDi , IDr)
    - Security Association
      - één gekozen uit aanbod
      - SPI gekozen door B
    - Nonce
    - KE: Diffie-Hellman waarde van B

IPSec 111

IKE fase 2 (6)

Genereren van de sleutels door A en B:

- in de veronderstelling van PFS
- voor gegevens gestuurd door A en ontvangen door B:
  $KEYMAT_{AB}$ = prf(SKEYID_d, a^{xy} , protocol, SPI_B , Ni , Nr)
- voor gegevens gestuurd door B en ontvangen door A:
  $KEYMAT_{BA}$ = prf(SKEYID_d, a^{xy} , protocol, SPI_A , Ni , Nr)
- het kan nodig zijn dat er meerdere sleutels worden gegenereerd voor vb. volgende situaties:
  - genereren van de integrity check value (ICV) van verzonden gegevens
  - valideren van ICV van ontvangen gegevens
  - encrypteren van verzonden gegevens
  - decrypteren van ontvangen gegevens

IPSec 112

IKE fase 2 (7)

- **Boodschap 3** van A naar B
  - bevat alleen ISAKMP header en hash payload
  - HASH_3 = prf(SKEYID_a, 0, M-ID, Ni , Nr)
  - ontvangen hash wordt door B geverifieerd
- eindelijk is nu gegevensuitwisseling mogelijk !!
  - 2 SA's zijn nu opgesteld
  - voor ofwel ESP ofwel AH
- methode laat ook toe een SA-bundel op te stellen
  - verschillende voorstellen voor verschillende SA's in bundel
  - door B SA's gekozen voor de bundel
- onderhandeling kan ook simultaan voor meerdere SA's gebeuren
  - hiertoe wordt Message-ID gebruikt
  - ook opgenomen in berekening van sleutelmateriaal